

**DPPL**

**DESKRIPSI PERANCANGAN PERANGKAT LUNAK**

**PROGRAM HOSPITAL MANAGEMENT SYSTEM**

**Dipersiapkan oleh:**

**AHMAD ATTORIQ / 21108012**

**AHMAD WILDAN ROHEDI / 21108014**


**IMMANUEL ANANDA PUTRA HUTAPEA / 21108016**

**Program Studi Sistem Informasi**

**Fakultas Industri Kreatif dan Telematika**

**Universitas Trilogi**

**2023**

	Program Studi Teknik Informatika  Universitas Trilogi	Nomor Dokumen	Halaman
		DPPL- HOSPITAL MNS	1/67

		Revisi		
--	--	--------	--	--

### DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	- Perubahan pada JFrame AddDoctor
<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	Ahmad Wildan Rohedi dan Immanuel Ananda Putra Hutapea							
Diperik sa oleh								
Disetuj ui oleh								

### Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

--	--	--	--

## Daftar Isi

<b>1</b>	<b>Error! Bookmark not defined.</b>	
1.1	Error! Bookmark not defined.	
1.2	9	
1.3	10	
1.4	11	
<b>2</b>	<b>Error! Bookmark not defined.</b>	
2.1	Error! Bookmark not defined.	
2.1.1	Error! Bookmark not defined.	
2.1.2	Error! Bookmark not defined.	
2.1.3	Error! Bookmark not defined.	
2.1.4	Error! Bookmark not defined.	
2.1.5	Error! Bookmark not defined.	
2.1.6	Error! Bookmark not defined.	
2.1.7	Error! Bookmark not defined.	
2.1.8	Error! Bookmark not defined.	
2.2	Error! Bookmark not defined.	
2.2.1	Error! Bookmark not defined.	
2.2.2	Error! Bookmark not defined.	
2.2.3	Error! Bookmark not defined.	
2.2.4	Error! Bookmark not defined.	
2.2.5	Error! Bookmark not defined.	
2.2.6	20	
2.2.7	22	
2.2.8	24	
<b>3</b>	<b>26</b>	
<b>4</b>	<b>27</b>	
4.1	Error! Bookmark not defined.	
4.1.1	28	
4.1.2	28	
4.1.3	29	

4.1.4	29
4.1.5	29
4.1.6	29
4.2	30
5	30
5.1	30
5.1.1	30
5.1.2	31
5.1.3	33
5.1.4	33
5.1.5	34
5.1.6	35
5.1.7	37
5.1.8	39
5.2	42
5.3	43
5.3.1	<b>Error! Bookmark not defined.</b>
5.3.2	43
5.3.3	43
5.3.4	44
5.3.5	44
5.3.6	45
5.3.7	45
5.3.8	46
5.3.9	46
5.3.10	46
5.3.11	47
5.3.12	47
5.3.13	47
5.3.14	48
5.3.15	48
5.3.16	49
5.3.17	49
5.3.18	50
6	51
6.1	52
6.2	53
6.3	<b>Error! Bookmark not defined.</b>
6.4	54
6.5	55
6.6	55
6.7	<b>Error! Bookmark not defined.</b>
6.8	55

6.9	56
6.10	57
6.11	59
6.12	60
6.13	61
6.14	62
6.15	62
6.16	<b>Error! Bookmark not defined.</b>
6.17	63
6.18	64
6.19	65
6.20	66
6.21	67
6.22	<b>Error! Bookmark not defined.</b>
6.23	67
6.24	<b>Error! Bookmark not defined.</b>

## Daftar Gambar

Gambar 2.1 Realisasi Class Diagram : Login.....	11
Gambar 2.2 Realisasi Class Diagram : Pengelolaan Teman dalam Komunitas .....	11
Gambar 2.3 Realisasi Class Diagram : Pengelolaan Anggota dalam Komunitas .....	12
Gambar 2.4 Realisasi Class Diagram : Pengelolaan Data Anggota Komunitas .....	12
Gambar 2.5 Realisasi Class Diagram : Pengelolaan Bulletin Board...	13
Gambar 2.6 Realisasi Class Diagram : Pengelolaan Gambar Favorit...	13
Gambar 2.7 Realisasi Class Diagram : Pengelolaan Testimonial .....	14
Gambar 2.8 Realisasi Class Diagram : Pengelolaan Pesan antar Anggota Komunitas .....	14
Gambar 2.9 Collaboration Diagram : Login.....	15
Gambar 2.10 Collaboration Diagram : Pengelolaan Teman dalam Komunitas - Penambahan Teman.....	15
Gambar 2.11 Collaboration Diagram : Pengelolaan Teman dalam Komunitas - Penerimaan Teman.....	16

Gambar 2.12 Collaboration Diagram : Pencarian Anggota Lain dalam Komunitas .....	16
Gambar 2.13 Collaboration Diagram : Pengelolaan Data Anggota Komunitas - Tambah Anggota.....	17
Gambar 2.14 Collaboration Diagram : Pengelolaan Data Anggota Komunitas - Edit Data Anggota.....	17
Gambar 2.15 Collaboration Diagram : Pengelolaan Data Anggota Komunitas - Display Data Anggota.....	18
Gambar 2.16 Collaboration Diagram : Pengelolaan Bulletin Board - Posting Bulletin.....	18
Gambar 2.17 Collaboration Diagram : Pengelolaan Bulletin Board - Display Bulletin.....	19
Gambar 2.18 Collaboration Diagram : Pengelolaan Gambar Favorit - Upload Gambar.....	19
Gambar 2.19 Collaboration Diagram : Pengelolaan Gambar Favorit - Delete Gambar.....	20
Gambar 2.20 Collaboration Diagram : Pengelolaan Gambar Favorit - Display Gambar.....	20
Gambar 2.21 Collaboration Diagram : Pengelolaan Testimonial - Add Testimonial.....	21
Gambar 2.22 Collaboration Diagram : Pengelolaan Testimonial - Penerimaan Testimonial.....	21
Gambar 2.23 Collaboration Diagram : Pengelolaan Testimonial - Display Testimonial.....	22
Gambar 2.24 Collaboration Diagram : Pengelolaan Pesan antar Anggota Komunitas - Kirim Pesan.....	22
Gambar 2.25 Collaboration Diagram : Pengelolaan Pesan antar Anggota Komunitas - Delete Pesan Masuk.....	23
Gambar 2.26 Collaboration Diagram : Pengelolaan Pesan antar Anggota Komunitas - Delete Pesan Keluar.....	23
Gambar 2.27 Collaboration Diagram : Pengelolaan Pesan antar Anggota Komunitas - Display Pesan Masuk.....	24
Gambar 2.28 Collaboration Diagram : Pengelolaan Pesan antar Anggota Komunitas - Display Pesan Keluar.....	24
Gambar 3 Rancangan Arsitektur SC3.....	25
Gambar 4 Conceptual Data Model.....	29
Gambar 5.1 Sequence Diagram : Login.....	30
Gambar 5.2 Sequence Diagram : Pengelolaan Teman dalam Komunitas - Penambahan Teman.....	30
Gambar 5.3 Sequence Diagram : Pengelolaan Teman dalam Komunitas - Penerimaan Teman.....	31
Gambar 5.4 Sequence Diagram : Pencarian Anggota Lain Dalam Komunitas .....	31
Gambar 5.5 Sequence Diagram : Pengelolaan Data Anggota Komunitas - Tambah Anggota.....	32
Gambar 5.6 Sequence Diagram : Pengelolaan Data Anggota Komunitas - Edit Anggota .....	32
Gambar 5.7 Sequence Diagram : Pengelolaan Data Anggota Komunitas - Display Anggota .....	33
Gambar 5.8 Sequence Diagram : Pengelolaan Bulletin Board - Posting Bulletin.....	33

Gambar 5.9 Sequence Diagram : Pengelolaan Bulletin Board - Display Bulletin.....	34
Gambar 5.10 Sequence Diagram : Pengelolaan Gambar - Upload Gambar .....	34
Gambar 5.11 Sequence Diagram : Pengelolaan Gambar - Delete Gambar .....	35
Gambar 5.12 Sequence Diagram : Pengelolaan Gambar - Display Gambar .....	35
Gambar 5.13 Sequence Diagram : Pengelolaan Testimonial - Add Testimonial.....	36
Gambar 5.14 Sequence Diagram : Pengelolaan Testimonial - Penerimaan Testimonial.....	36
Gambar 5.15 Sequence Diagram : Pengelolaan Testimonial - Display Testimonial.....	37
Gambar 5.16 Sequence Diagram : Pengelolaan Pesan antar Anggota Komunitas - Kirim Pesan.....	37
Gambar 5.18 Sequence Diagram : Pengelolaan Pesan antar Anggota Komunitas - Delete Outbox.....	38
Gambar 5.19 Sequence Diagram : Pengelolaan Pesan antar Anggota Komunitas - Display Inbox.....	39
Gambar 5.20 Sequence Diagram : Pengelolaan Pesan antar Anggota Komunitas - Display Outbox.....	39
Gambar 5.21 Class Diagram.....	40
Gambar 6.1 Rancangan Antarmuka Login.....	54
Gambar 6.1 Rancangan Antarmuka Main Form.....	55
Gambar 6.3 Rancangan Antarmuka Pengelolaan Anggota.....	56
Gambar 6.4 Rancangan Antarmuka Pengelolaan Teman - Tambah Teman...	57
Gambar 6.5 Rancangan Antarmuka Pengelolaan Teman - Penerimaan/Penolakan Teman.....	58
Gambar 6.6 Rancangan Antarmuka Pengelolaan Teman - Pending Friend .....	59
Gambar 6.7 Rancangan Antarmuka Display Teman.....	60
Gambar 6.8 Rancangan Antarmuka Pencarian Anggota - Input Kriteria .....	61
Gambar 6.9 Rancangan Antarmuka Pencarian Anggota - Hasil Pencarian .....	62
Gambar 6.10 Rancangan Antarmuka Pengelolaan Pesan - Kirim Pesan...	63
Gambar 6.11 Rancangan Antarmuka Pengelolaan Pesan - Inbox.....	64
Gambar 6.12 Rancangan Antarmuka Pengelolaan Pesan - Outbox.....	65
Gambar 6.13 Rancangan Antarmuka Display Detail Pesan.....	66
Gambar 6.14 Rancangan Antarmuka Pengelolaan Bulletin Board - Posting Bulletin.....	67
Gambar 6.15 Rancangan Antarmuka Pengelolaan Bulletin Board - Display Bulletin.....	68
Gambar 6.16 Rancangan Antarmuka Pengelolaan Display Detail Bulletin.....	68
Gambar 6.17 Rancangan Antarmuka Pengelolaan Gambar - Upload Gambar .....	69
Gambar 6.18 Rancangan Antarmuka Pengelolaan Gambar - Penerimaan/Penolakan.....	70
Gambar 6.19 Rancangan Antarmuka Pengelolaan Gambar - Display Gambar .....	71



Gambar 6.20 Rancangan Antarmuka Display Detail Gambar.....	72
Gambar 6.21 Rancangan Antarmuka Pengelolaan Testimonial - Add Testimonial.....	73
Gambar 6.22 Rancangan Antarmuka Pengelolaan Testimonial - Display Testimonial.....	74
Gambar 6.23 Rancangan Antarmuka Pengelolaan Testimonial - Penolakan/Penerimaan.....	75
Gambar 6.24 Rancangan Antarmuka Display Detail Testimonial.....	76

## **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 HOSPITAL MANAGEMENT SYSTEM dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan data doctor dan pasien.
2. Menangani pengajuan rawat inap.
3. Menangani pembayaran biaya pengobatan.
4. Menangani pencatatan barang inventaris.
5. Menangani pemengembalian barang inventaris.

Dan berjalan pada lingkungan dengan platform Windows dan dapat berjalan pada semua jenis browser.

## 1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
HOSPITAL MANAGEMENT SYSTEM	HOSPITAL MNS <i>Nama aplikasi yang akan dikembangkan</i>
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.
<b>Server</b>	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

<b>Browser</b>	program aplikasi yang menterjemahkan kode HTML dan merepresentasikan halaman web site.

#### 1.4 Referensi

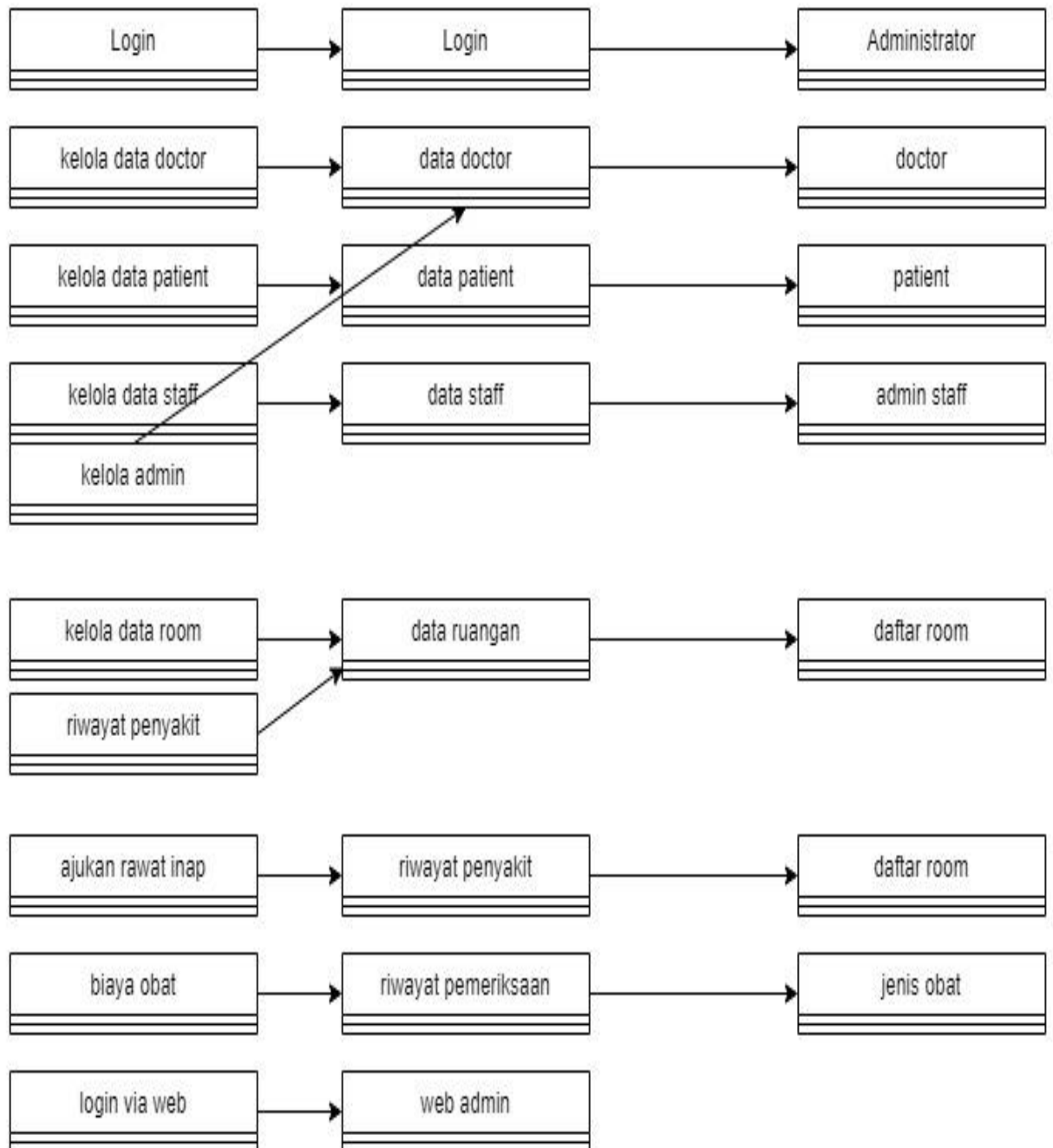
Referensi yang digunakan pada perangkat lunak tersebut adalah:

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. MSDN Library-October 2005, Microsoft, 2005.
5. Permana, Silvester Dian Handy, *Spesifikasi Kebutuhan Perangkat Lunak JUDUL*, Universitas TRILOGI, 2017.

## 2 Perancangan Sistem

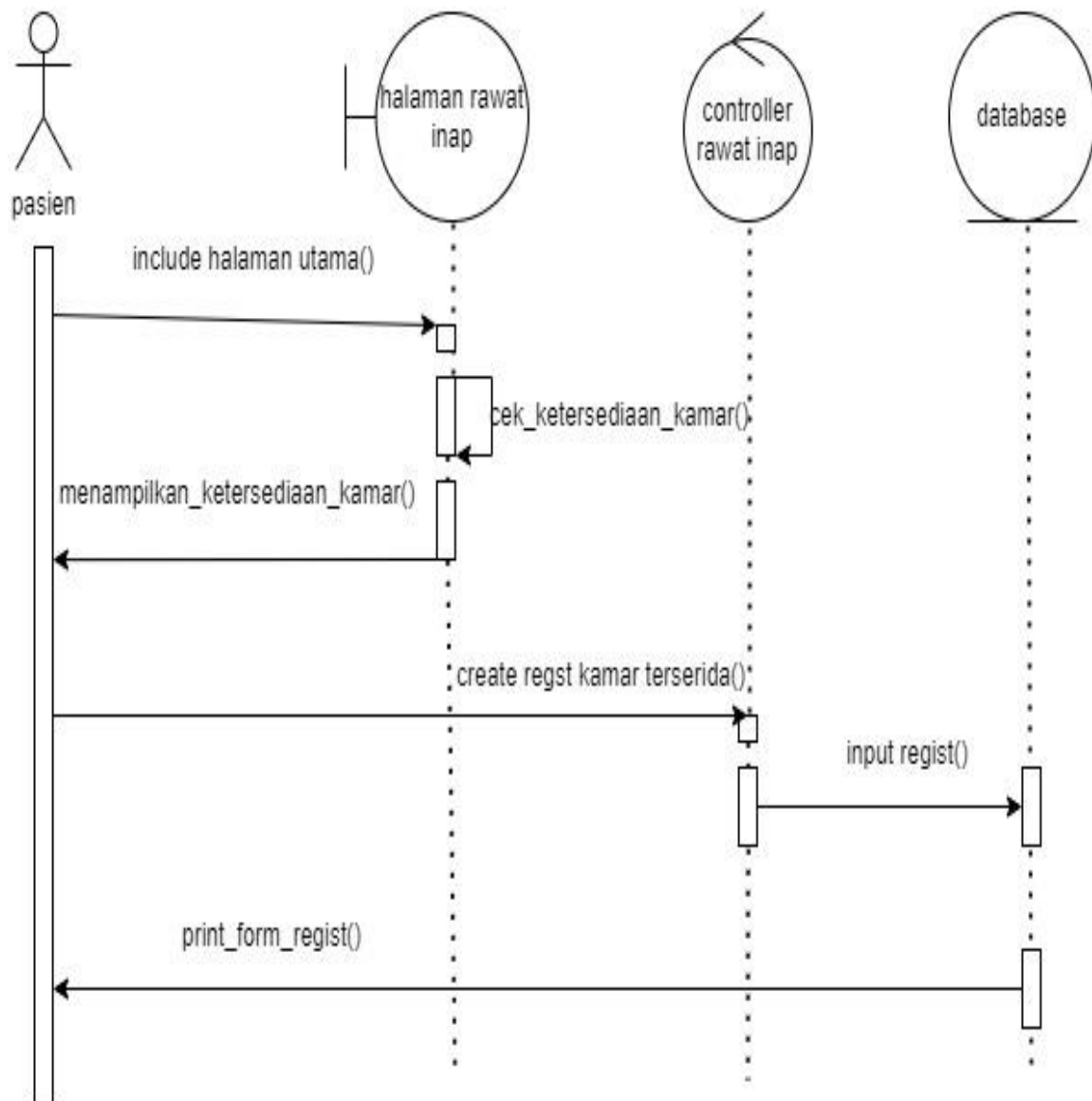
### 2.1 Perancangan Arsitektur



## 2.2 Perancangan Rinci

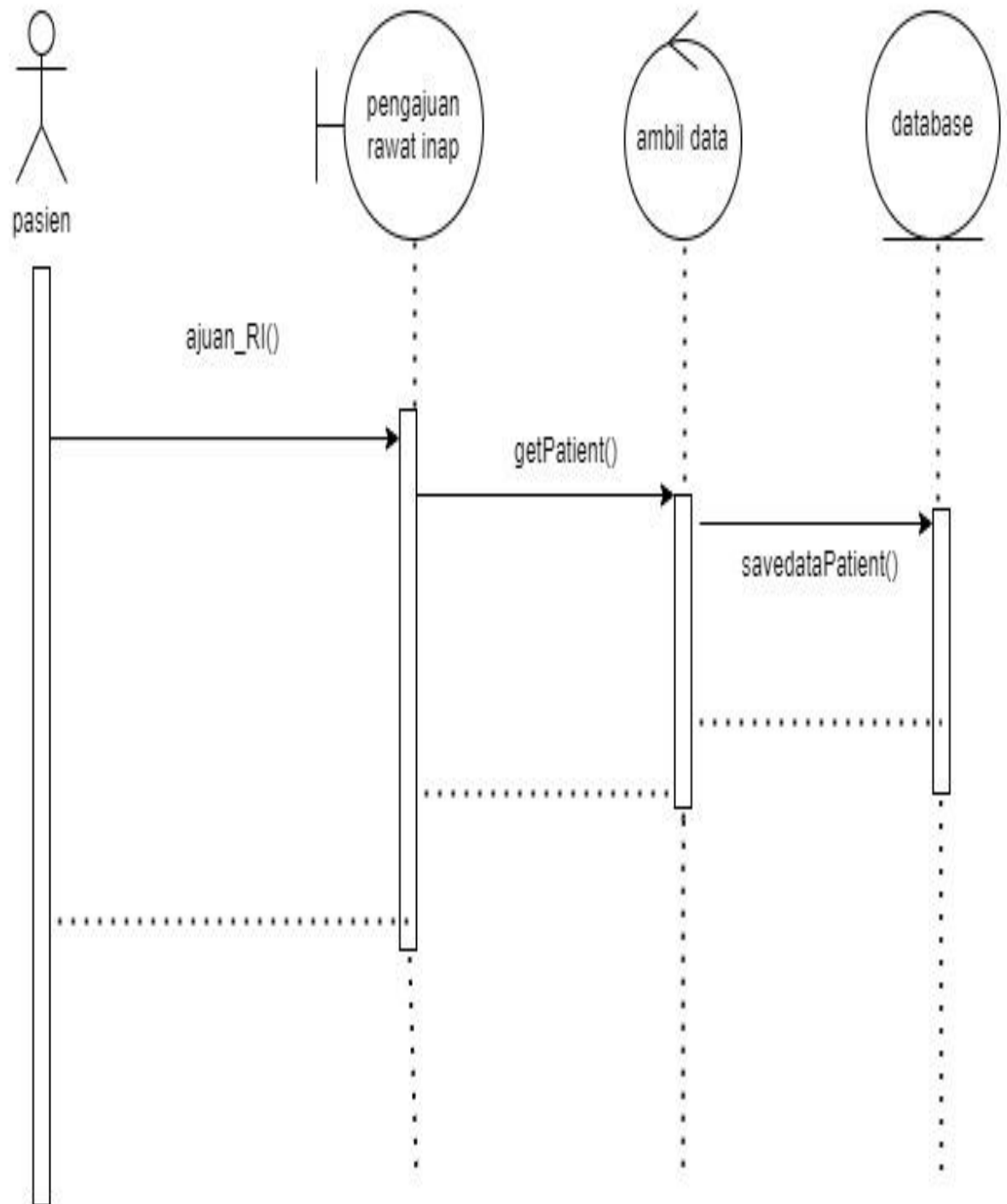
### 2.2.1 Sequence Diagram

#### 2.2.1.1 Pengelolaan Ruang Rawat Inap



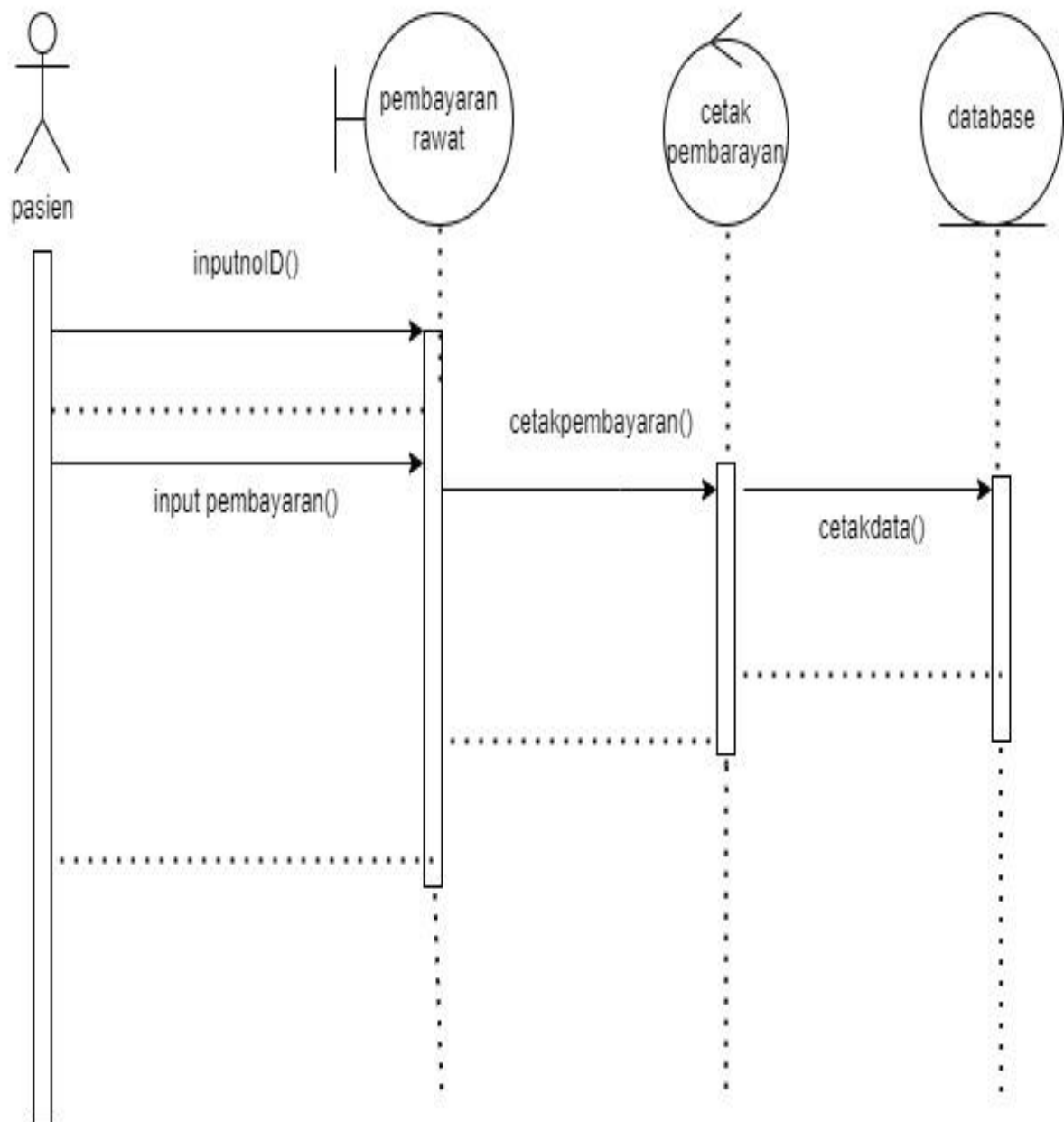
Gambar 2.9 Sequence Diagram : Pengelolaan Data Kamar Rawat Inap

### 2.2.1.2 Pengajuan Rawat Inap



Gambar 2.10 Sequence Diagram : Pengajuan Rawat Inap

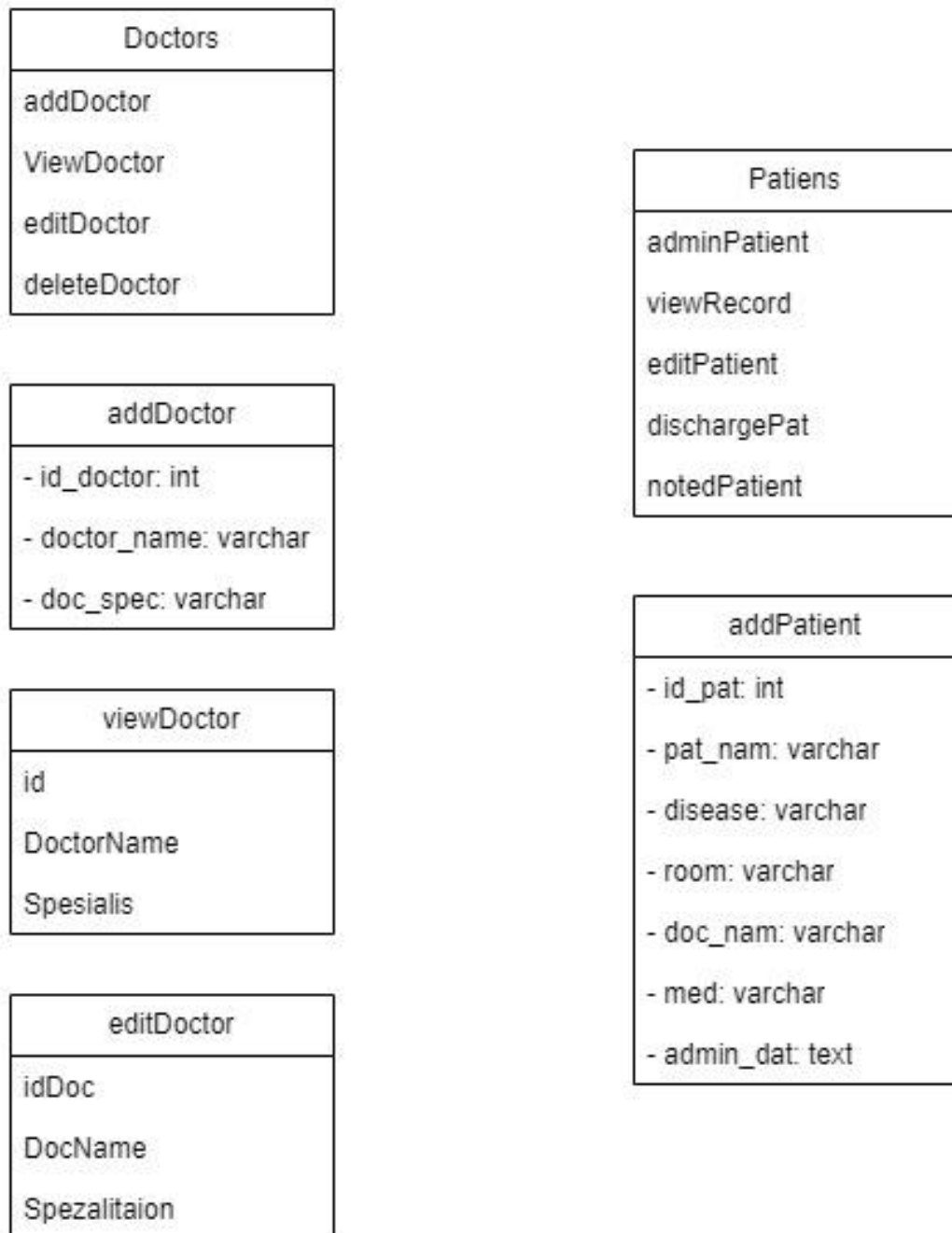
### 2.2.1.3 Pembayaran Pembayaran Rawat Inap



**Gambar 2.11** Sequence Diagram : Pembayaran Rawat Inap



### 2.2.2 Class Diagram



**Gambar 2.13 Collaboration Diagram : Pengelolaan Data Anggota Komunitas - Tambah Anggota**

### 2.2.3 Deskripsi Class

Login (Boundary Class)

- + SetID() Operasi ini digunakan untuk menginputkan ID.
- + SetPass() Operasi ini digunakan untuk menginputkan password.

Kelola Data Doctor (Boundary Class)

- + KelolaDataDoctor() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.
- + AddDataDoctor() Menambah data doctor ke data master doctor
- + EditDataDoctor() Mengedit atau merubah data Doctor yang ada di data master Doctor.
- + ViewDataDoctor() Menampilkan data doctor yang ada di database data master doctor
- + SearchDataDoctor() Mencari data doctor yang ada di data master doctor
- + DeleteDataDoctor() Menghapus data Doctor yang adad di data master doctor

Kelola Data Patient (Boundary Class)

- + KelolaDataPatient() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.
- + AddDataPatient() Menambah data patient ke data patient
- + EditDataPatient() Mengedit atau merubah data patient yang ada di data patient.
- + ViewDataPatient() Menampilkan data patient yang ada di database data patient barang
- + SearchDataPatient() Mencari data patient yang ada di data patient
- + DeleteDataPatient() Menghapus data patient yang ada didata patient

Kelola Admin (Boundary Class)

- + KelolaAdmin() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.
- + AddDataAdmin() Menambah data admin ke data admin
- + EditDataAdmin() Mengedit atau merubah data admin yang ada didata admin.
- + ShowDataAdmin() Menampilkan data admin yang ada di database data admin

- + SearchDataAdmin() Mencari data admin yang ada didata admin
- + DeleteDataAdmin() Menghapus data admin yang ada didata admin.

Ajuan Rawat Inap (Boundary Class)

- + PengajuanRawatInap() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.

Periksa Patient (Boundary Class)

- + PeriksaPatient() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.

Ajuan Pembayaran (Boundary Class)

- + PengajuanPembayaran() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.

Ubah Lokasi Rawat Inap (Boundary Class)

- + PengubahanLokasiRI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.

User Cont (Control Class)

- + CekIDPass() ini digunakan untuk mengecek apakah ID dan password yang dimasukkan itu valid atau tidak.

Data Master Doctor Cont (Control Class)

- + CekInput() ini digunakan untuk mengecek pilihan yang dipilih apakah entri, edit, show, search, atau delete.
- + CekData() ini digunakan untuk mengecek inputan data yang dimasukkan oleh user apakah ada yang salah atau tidak.

Admin (Entity Class)

- + Data(ID, Pass) digunakan untuk mengambil data yang ada berdasarkan ID dan Password yang diinputkan oleh user.

DataAdmin (Entity Class)

- + GetAdmin() digunakan untuk mendapatkan data barang. Ini digunakan untuk menu search.
- + InsertAdmin() digunakan untuk menginputkan data admin yang diinputkan tadi kedalam database master admin.

- + DeleteAdmin() digunakan untuk menghapus data admin dari database master admin.
- + EditAdmin() Digunakan untuk mengedit data admin dari data master admin.

#### Kategori(Entity Class)

- + GetRoom() digunakan untuk mendapatkan data kategori dari room kategori room.
- + InsertRoom () digunakan untuk menginputkan data kategori yang diinputkan tadi kedalam database data kategori room.
- + DeleteRoom () digunakan untuk menghapus data kategori dari database kategori room.
- + EditRoom () Digunakan untuk mengedit data kategori dari data kategori room.

#### Patient(Entity Class)

- + GetPatient () digunakan untuk mendapatkan data patient dari table patient.
- + InsertPatient () digunakan untuk menginputkan data patient yang diinputkan tadi kedalam table patient.
- + DeletePatient () digunakan untuk menghapus data patient dari table patient.
- + EditPatient () Digunakan untuk mengedit data patient dari tabel patient.

#### Patient(Entity Class)

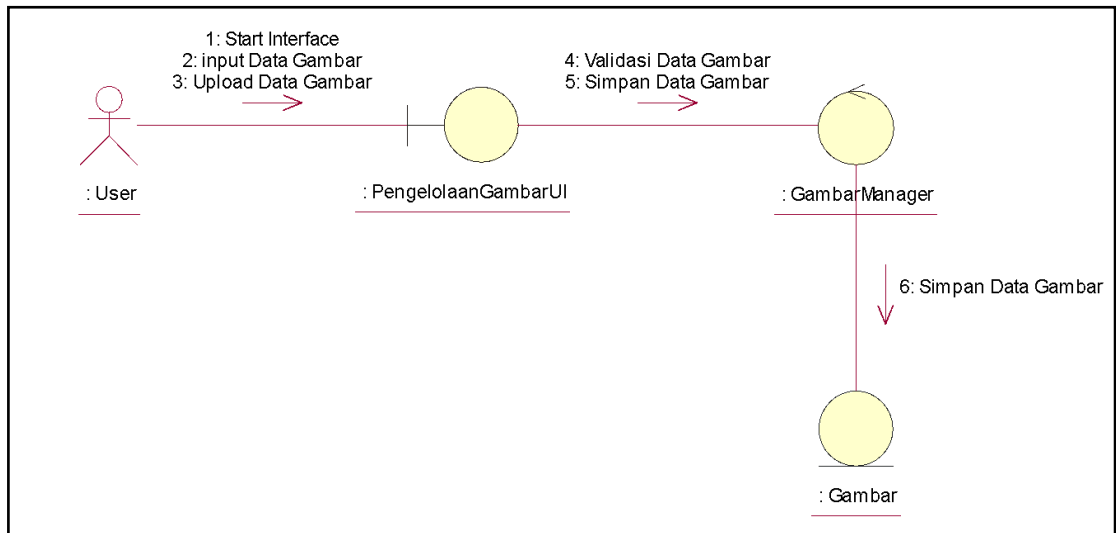
- + GetPatient () digunakan untuk mendapatkan data patient dari database patient.
- + EditCashPatient () Digunakan untuk mengedit data patient yang ada di database case flow.

Mutasi(Entity Class)

- + GetLokasi() Mendapatkan data lokasi dari table mutasi.
- + EditLokasi() Mengedit data lokasi dari table mutasi.
- + GetUnitPemberi() Mendapatkan data unit pemberi pada table mutasi.
- + GetUnitPenerima() Mendapatkan data unit penerima pada table mutasi.
- + GetPenempatanLama() Mendapatkan data tempat barang yang lama atau sebelum mutasi pada table mutasi.
- + GetPenempatanBaru() Mendapatkan data tempat barang dipindahkan atau dimutasi pada table mutasi.

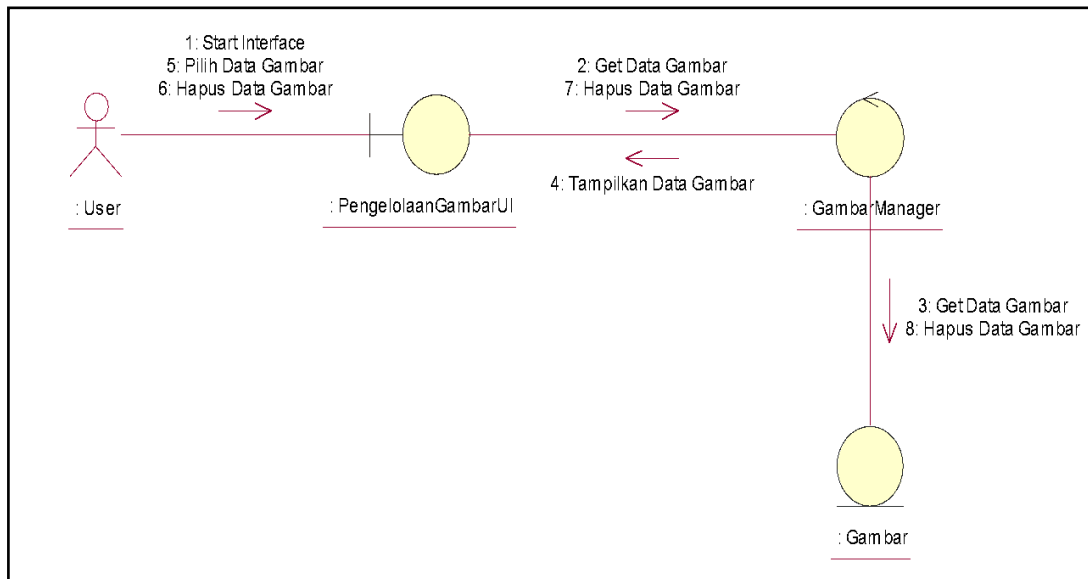
## 2.2.4 Pengelolaan Gambar Favorit

### 2.2.4.1 Upload Gambar



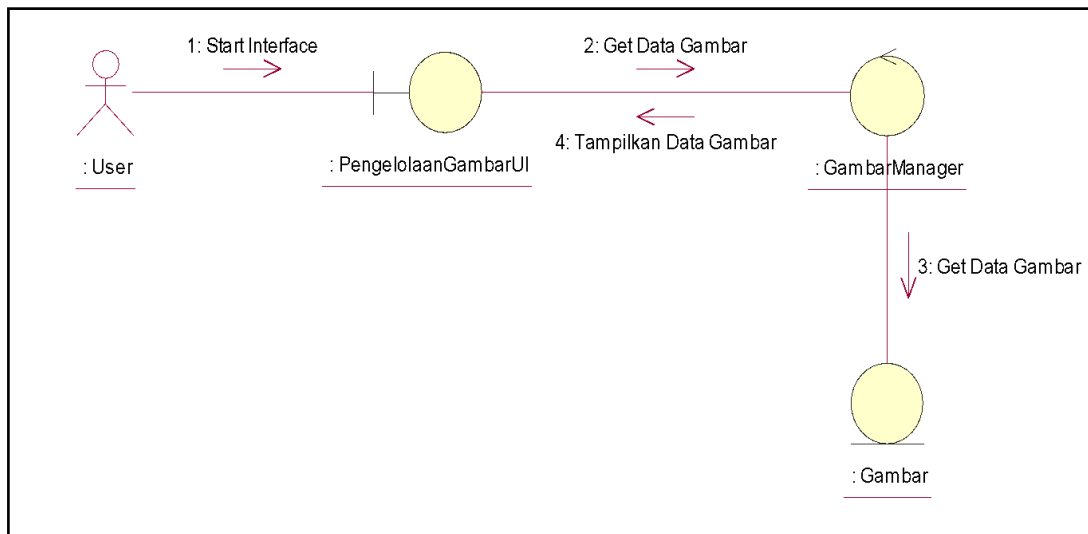
Gambar 2.18 Collaboration Diagram : Pengelolaan Gambar Favorit - Upload Gambar

### 2.2.4.2 Delete Gambar



Gambar 2.19 Collaboration Diagram : Pengelolaan Gambar Favorit - Delete Gambar

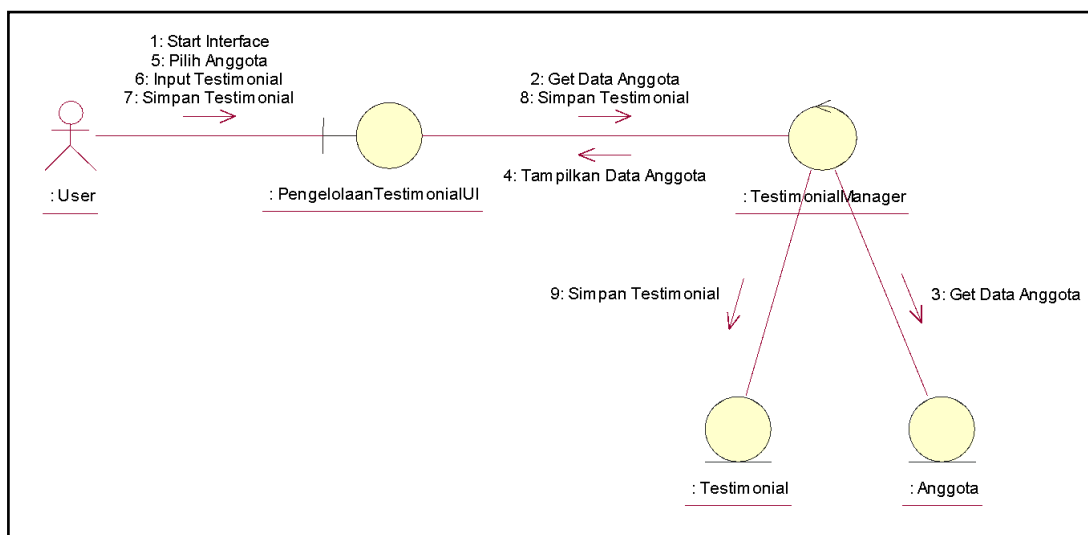
### 2.2.4.3 Display Gambar



Gambar 2.20 Collaboration Diagram : Pengelolaan Gambar Favorit - Display Gambar

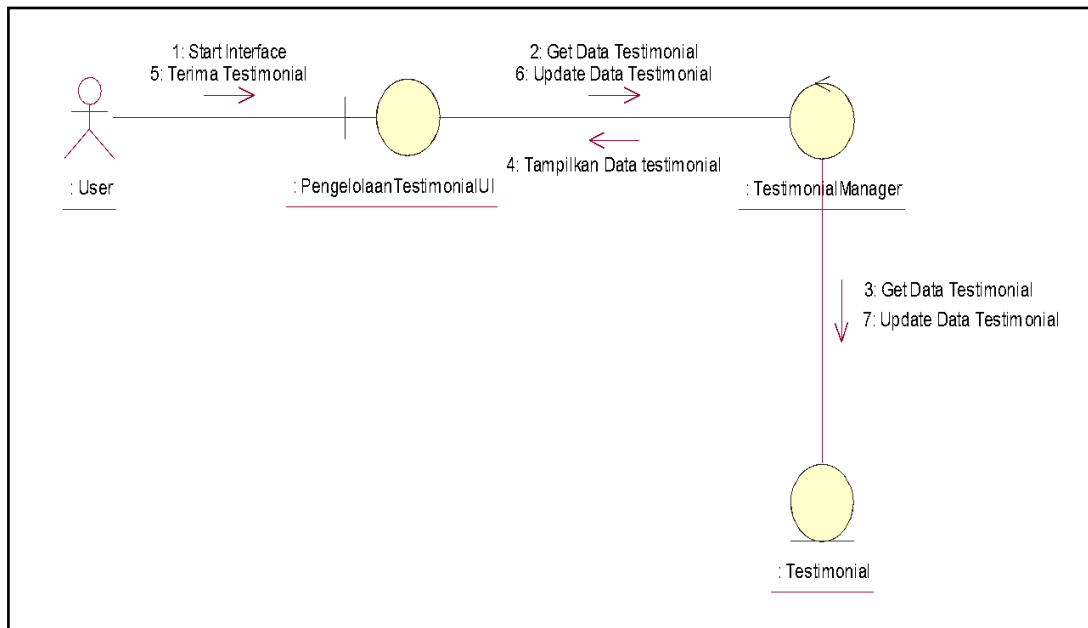
### 2.2.5 Pengelolaan Testimonial

#### 2.2.5.1 Add Testimonial



Gambar 2.21 Collaboration Diagram : Pengelolaan Testimonial - Add Testimonial

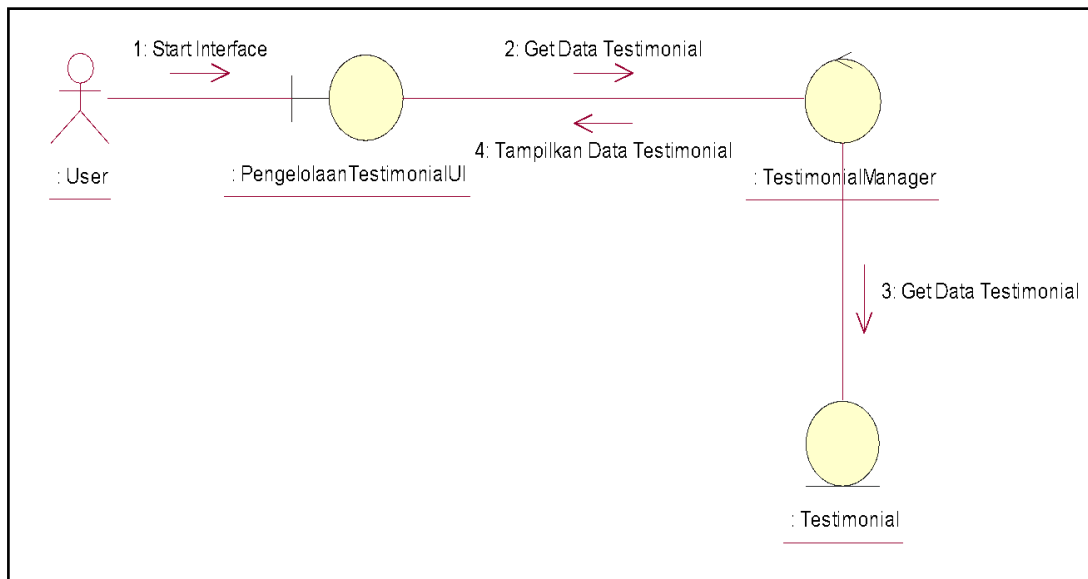
#### 2.2.5.2 Penerimaan Testimonial



**Gambar 2.22 Collaboration Diagram : Pengelolaan Testimonial - Penerimaan Testimonial**



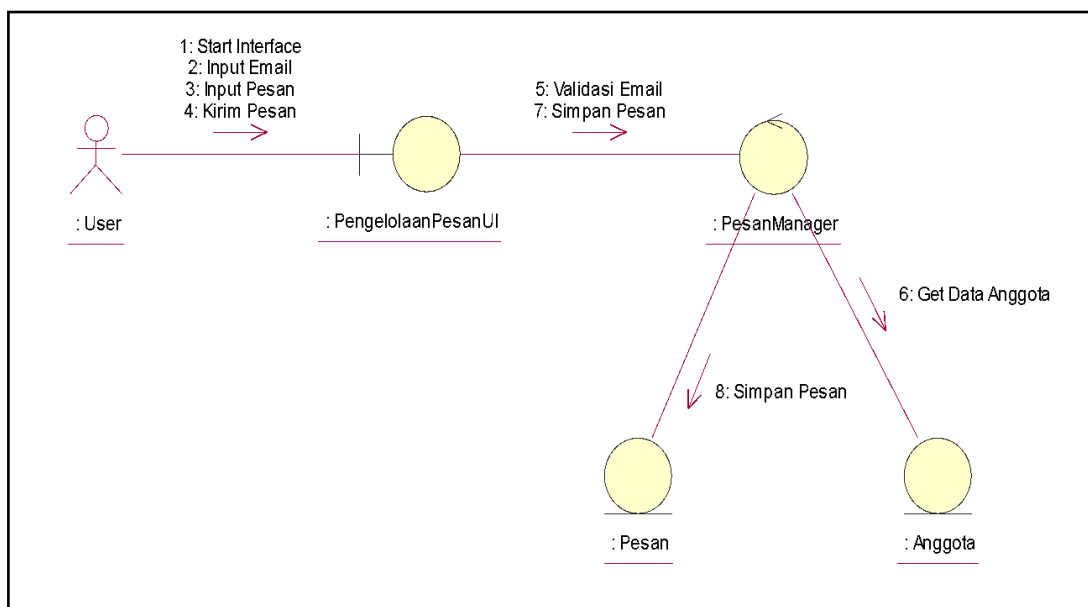
### 2.2.5.3 Display Testimonial



Gambar 2.23 Collaboration Diagram : Pengelolaan Testimonial - Display Testimonial

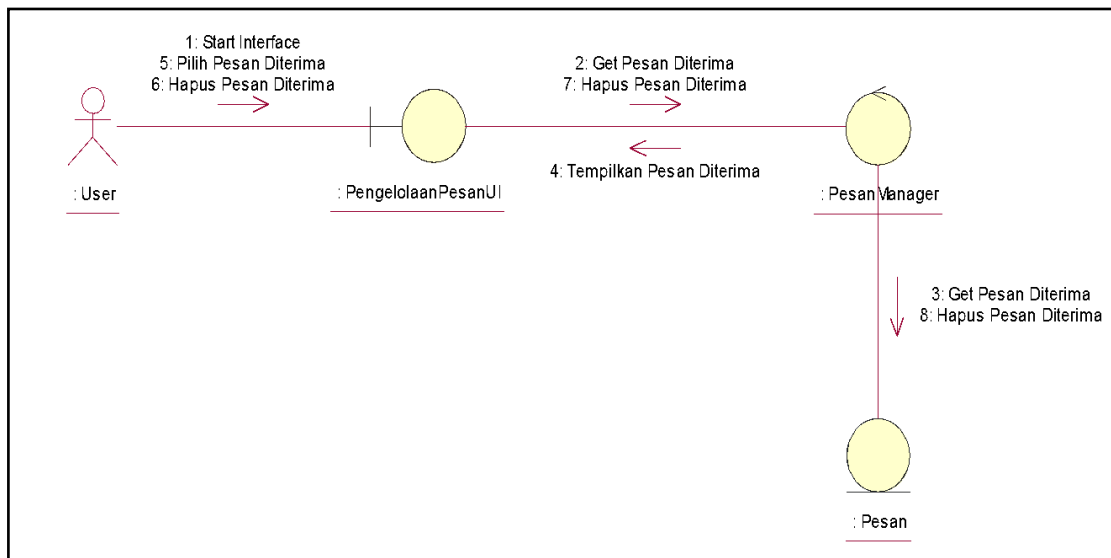
### 2.2.6 Pengelolaan Pesan antar Anggota Komunitas

#### 2.2.6.1 Kirim Pesan



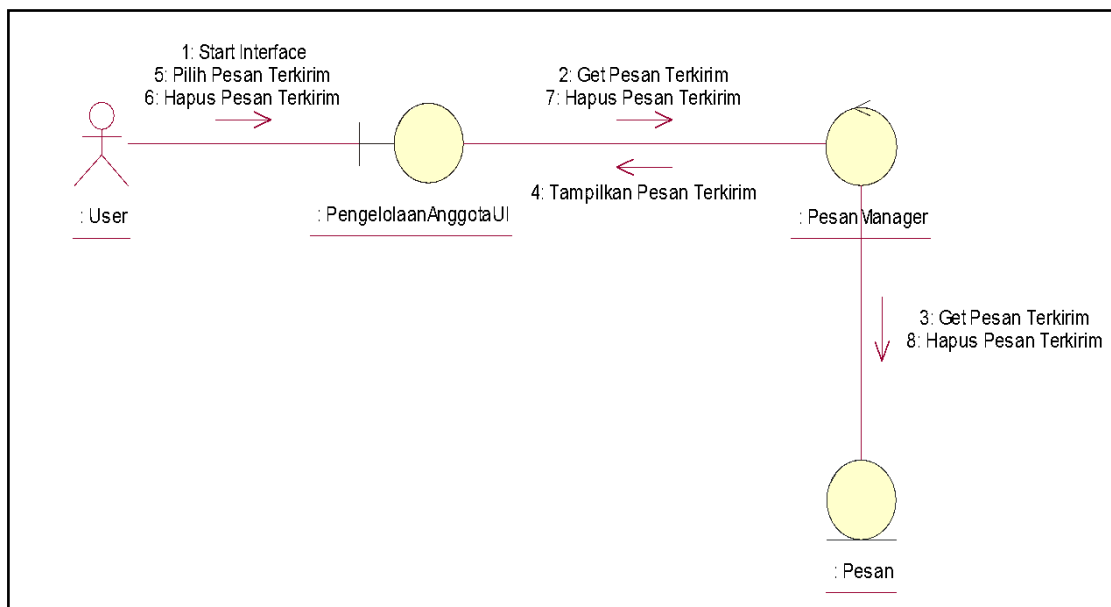
Gambar 2.24 Collaboration Diagram : Pengelolaan Pesan antar Anggota Komunitas - Kirim Pesan

### 2.2.6.2 Delete Pesan Masuk



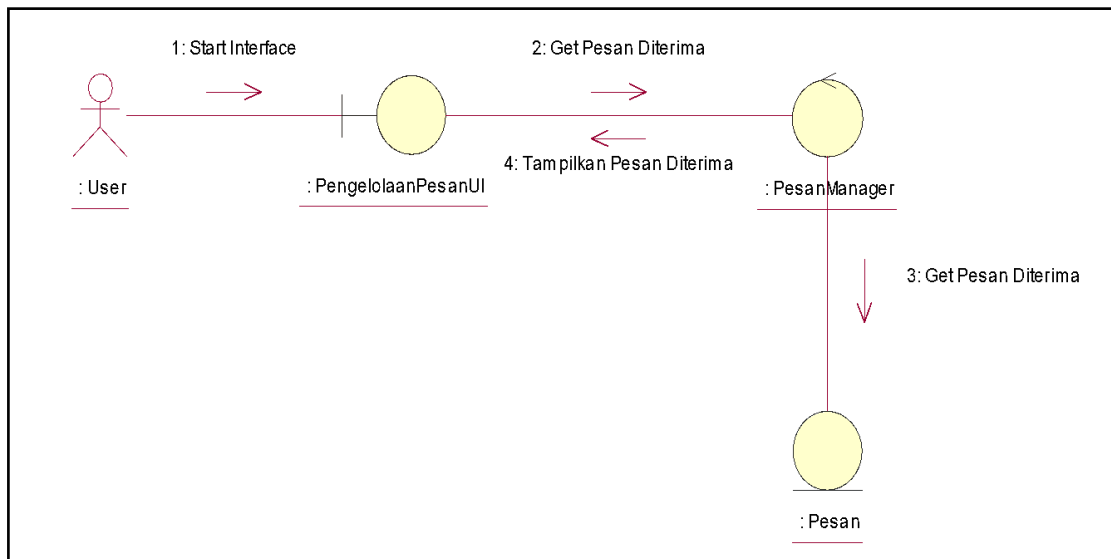
**Gambar 2.25 Collaboration Diagram : Pengelolaan Pesan antar Anggota Komunitas - Delete Pesan Masuk**

### 2.2.6.3 Delete Pesan Keluar



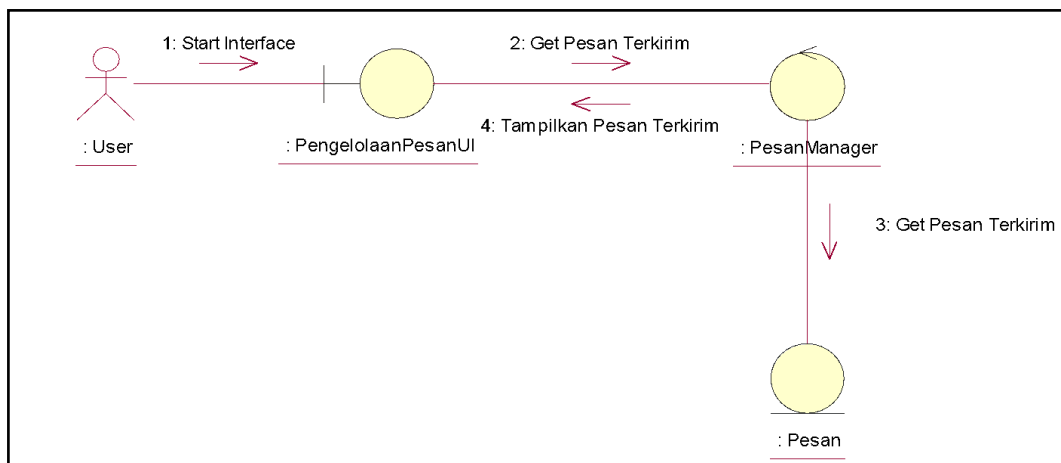
**Gambar 2.26 Collaboration Diagram : Pengelolaan Pesan antar Anggota Komunitas - Delete Pesan Keluar**

#### 2.2.6.4 Display Pesan Masuk



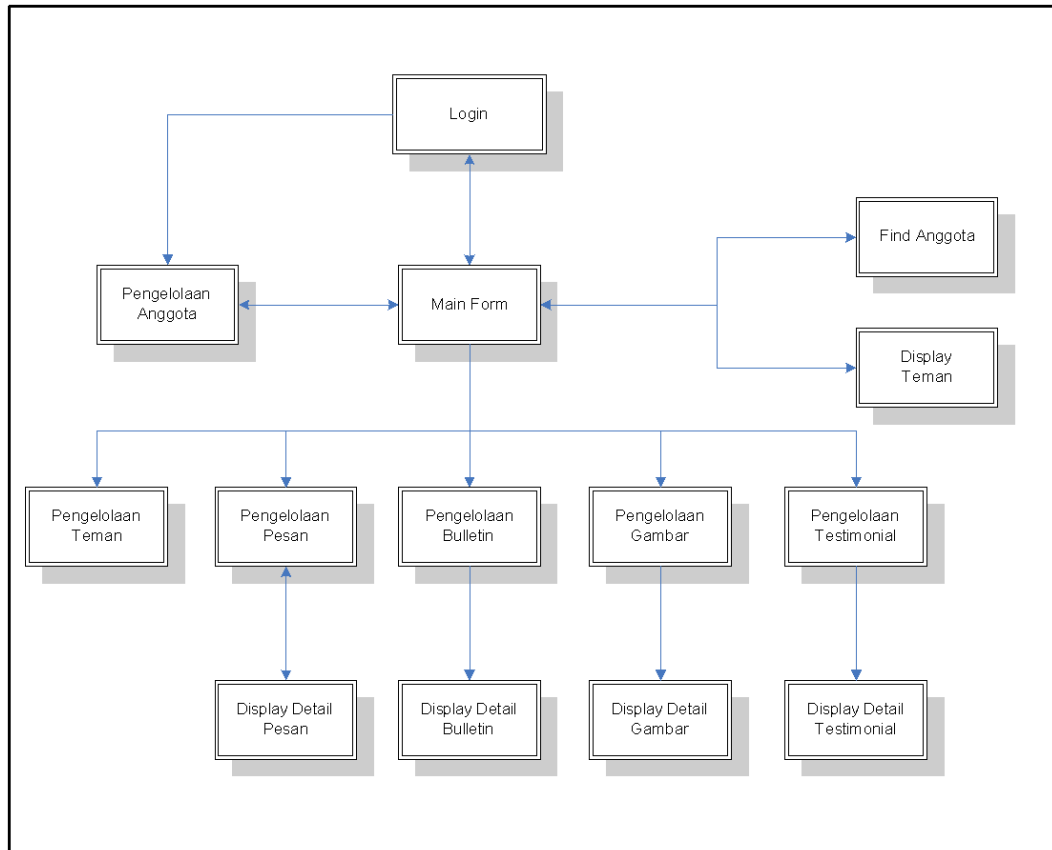
**Gambar 2.27 Collaboration Diagram : Pengelolaan Pesan antar Anggota Komunitas - Display Pesan Masuk**

#### 2.2.6.5 Display Pesan Keluar



**Gambar 2.28 Collaboration Diagram : Pengelolaan Pesan antar Anggota Komunitas - Display Pesan Keluar**

### 3 Rancangan Arsitektur



Gambar 3 Rancangan Arsitektur SC3

## 4 Deskripsi Dekomposisi

### 4.1 Dekomposisi Data

#### 4.1.1 Deskripsi Entitas Data Patient

Nama	Tipe	Panjang	Keterangan
Email	Character	50	Email Patient, Primary key
ID_Patient	Integer	-	ID Gambar, Foreign Key
Password	Character	10	Password Patient
Nama	Character	50	Nama Patient
Jenis_Kelamin		20	Jenis Kelamin Patient
Tempat_Lahir	Character	30	Tempat kelahiran Patient
Tgl_Lahir	Character	Date	Tanggal kelahiran Patient
Tempat_Berada	Character	30	Tempat tinggal patient
Negara	Character	20	Negara dimana patient berada
Kode_Pos	Character	10	Kode pos tempat tinggal patient
Pekerjaan	Character	50	Pekerjaan patient
Sekolah	Character	50	Nama tempat patient bersekolah
Tentang_saya	Character	500	Keterangan pribadi patient

#### 4.1.2 Deskripsi Entitas Data Doctor

Nama	Tipe	Panjang	Keterangan
Email_Doctor	Character	50	Email Doctor, Primary key, Foreign Key
Email_Doctor	Character	50	Email Doctor, Primary key, Foreign Key

Status	Character	10	Status dari teman, Requested, Pending, atau Approved
--------	-----------	----	--

#### 4.1.3 Deskripsi Entitas Data Pesan

Nama	Tipe	Panjang	Keterangan
ID_Pesan	Integer	-	ID Pesan, Primary key
Email_Pengirim	Character	50	Alamat email pengirim pesan, Foreign Key
Email_Penerima	Character	50	Alamat email penerima pesan, Foreign Key
Judul_Pesan	Character	30	Judul dari pesan
Isi_Pesan	Character	500	Isi dari pesan
Tgl_Pesan	Date	-	Tanggal pesan dikirim
Flag_Status	Character	1	Status dari pesan, New, Opened, atau Terkirim

#### 4.1.4 Deskripsi Entitas Data Gambar

Nama	Tipe	Panjang	Keterangan
ID_Gambar	Integer	-	ID Gambar, Primay Key
Email_Pengirim	Character	50	Alamat email pengirim gambar, Foreign Key
Email_Patient	Character	50	Alamat email pemilik gambar, Foreign Key
Nama_Gambar	Character	30	Nama gambar yang diupload
Path_Gambar	Character	50	Path letak file gambar
Tgl_Upload	Date	-	Tanggal gambar diupload
Flag_Status	Character	1	Status dari gambar, Approved atau Pending

#### 4.1.5 Deskripsi Entitas Data Testimonial

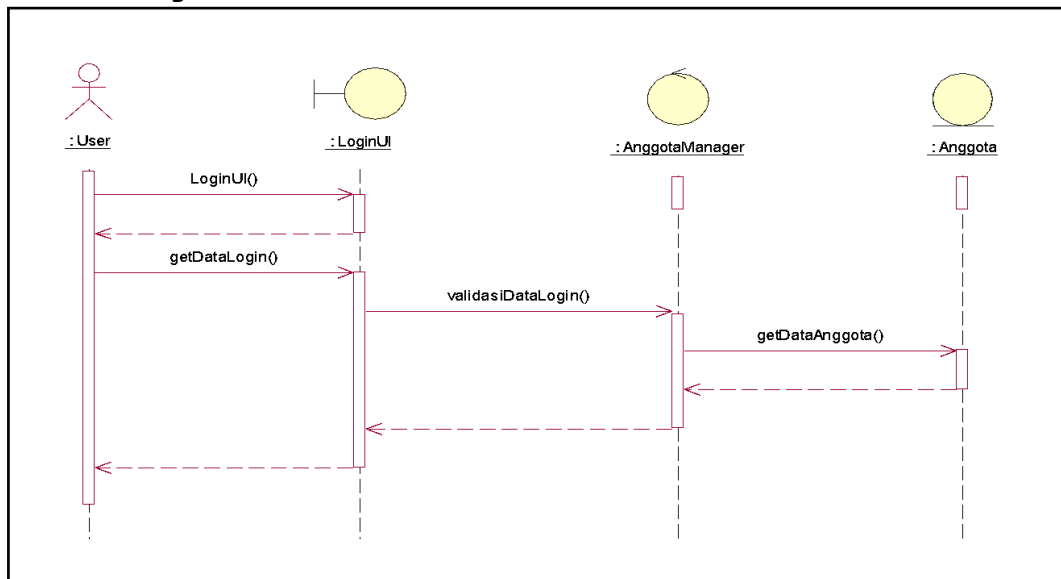
Nama	Tipe	Panjang	Keterangan
ID_Testimonial	Integer	-	ID Testimonial, Primay Key

Email_Pengirim	Character	50	Alamat email pengirim testimonial, Foreign Key
Email_Doctor	Character	50	Alamat email pemilik testimonial, Foreign Key
Isi_Testimonia l	Character	500	Judul dari lagu
Tgl_Kirim	Date	-	Tanggal testimonial dikirim
Flag_Status	Character	1	Status dari testimonial, Pending atau Approved

## 5 Design Model

### 5.1 Sequence Diagram

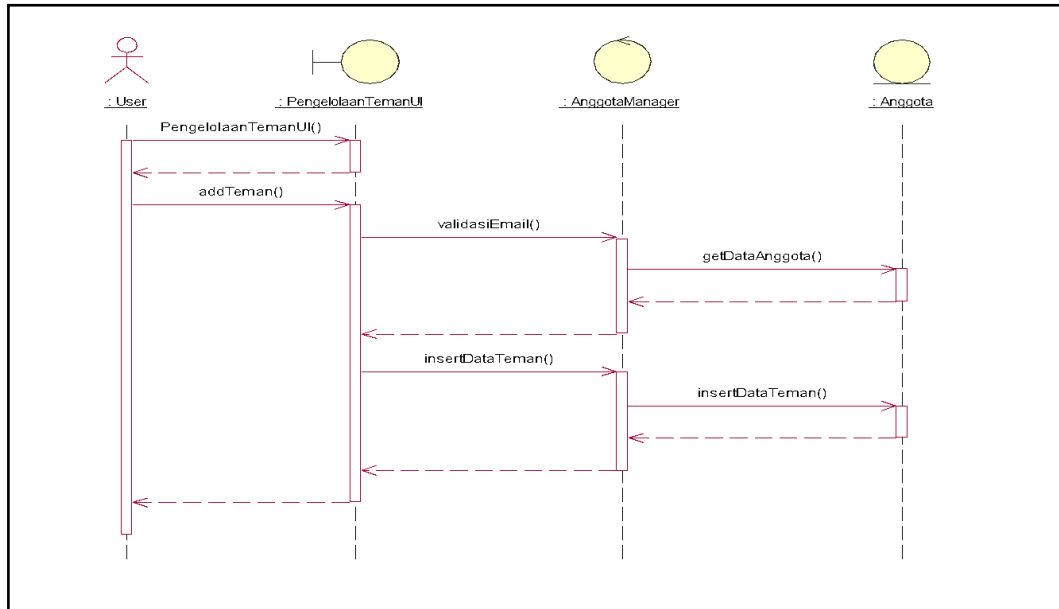
#### 5.1.1 Login



Gambar 5.1 Sequence Diagram : Login

## 5.1.2 Pengelolaan Patient dalam Rumah Sakit

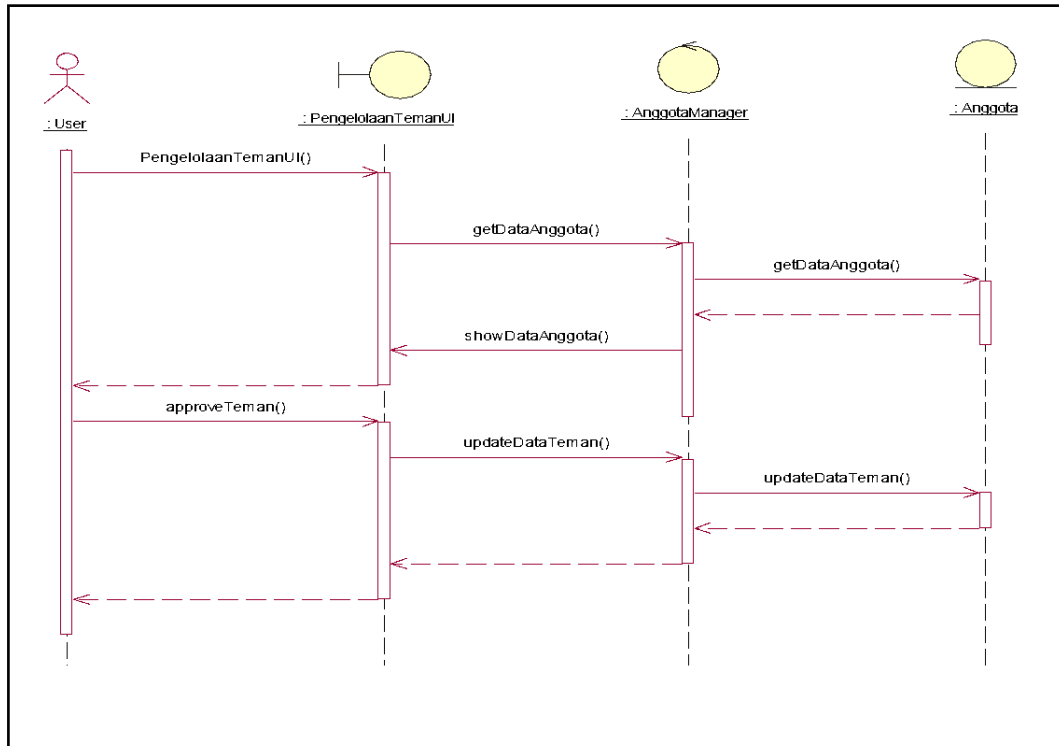
### 5.1.2.1 Penambahan Patient



Gambar 5.2 Sequence Diagram : Pengelolaan Patient dalam rumah sakit

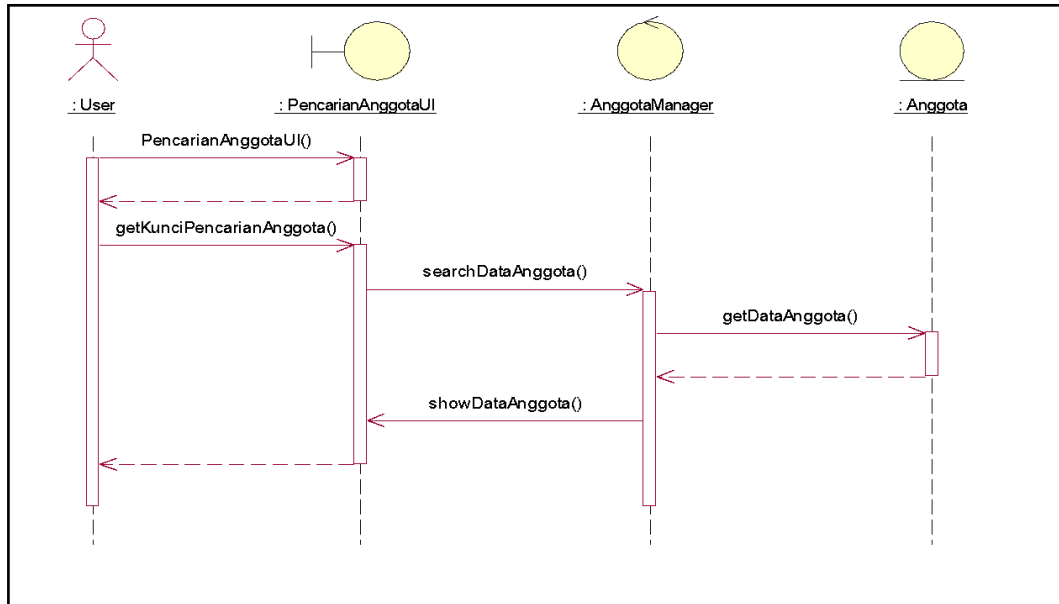


### 5.1.2.2 Penerimaan Patient



**Gambar 5.3 Sequence Diagram : Pengelolaan Patient dalam Rumah Sakit – Penerimaan Patient**

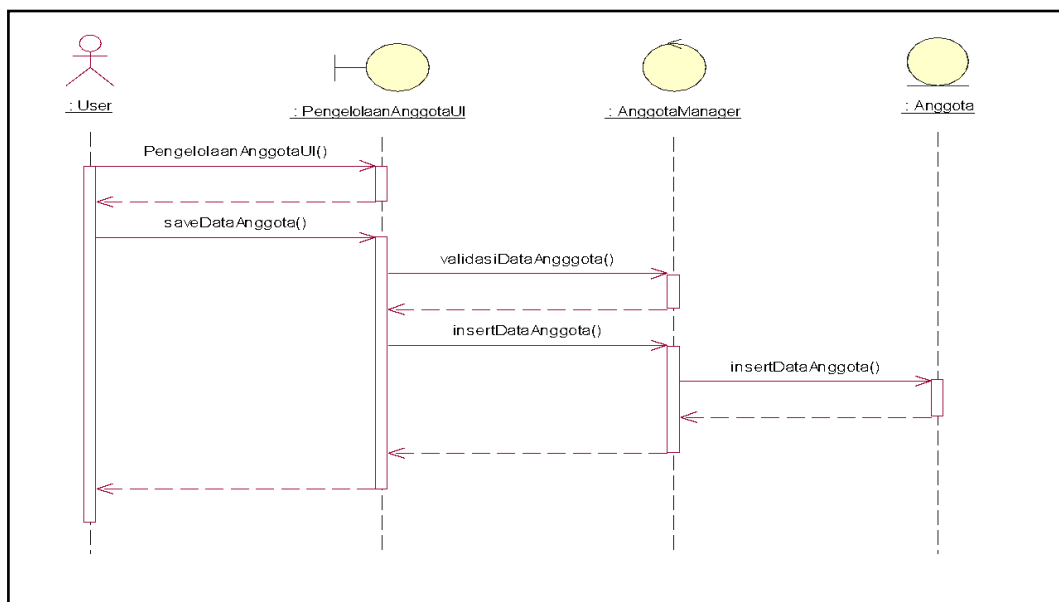
### 5.1.3 Pencarian Patient dalam rumah sakit



Gambar 5.4 Sequence Diagram : Pencarian Patient dalam RS

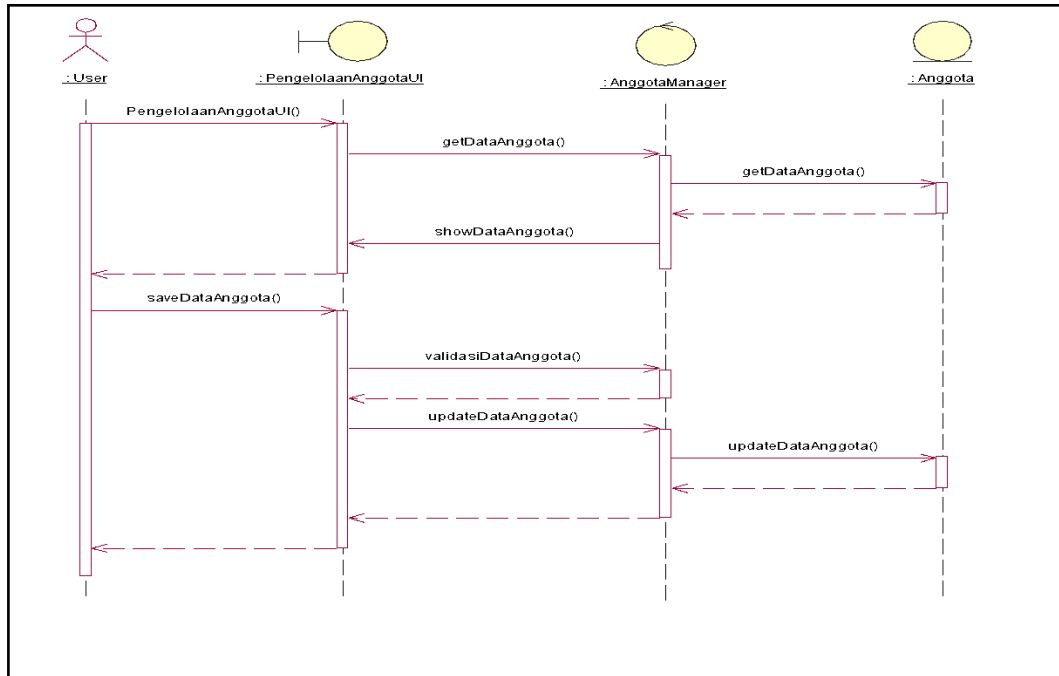
### 5.1.4 Pengelolaan Data Patient pada RS

#### 5.1.4.1 Tambah Anggota



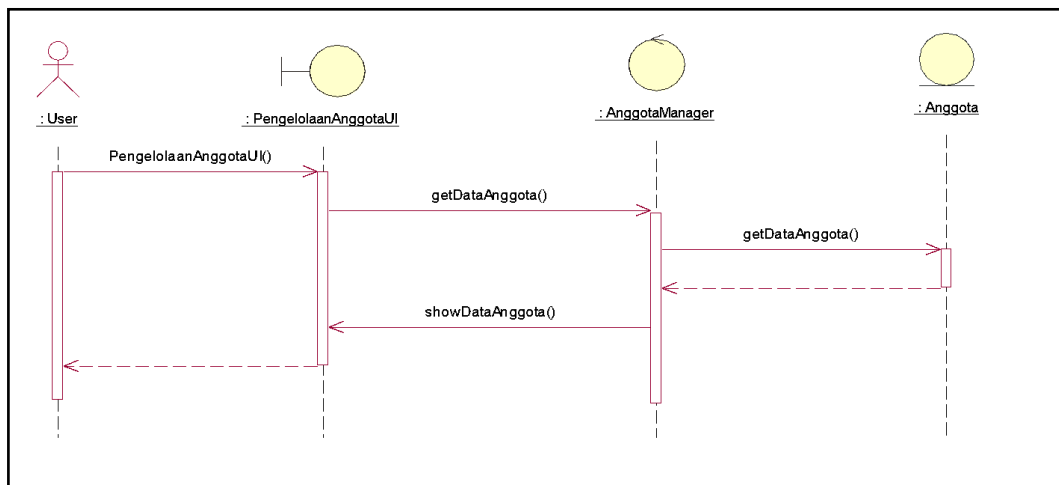
Gambar 5.5 Sequence Diagram : Pengelolaan Data Patient RS - Tambah Patient

#### 5.1.4.2 Edit Patient



Gambar 5.6 Sequence Diagram : Pengelolaan Data Patient RS - Edit Patient

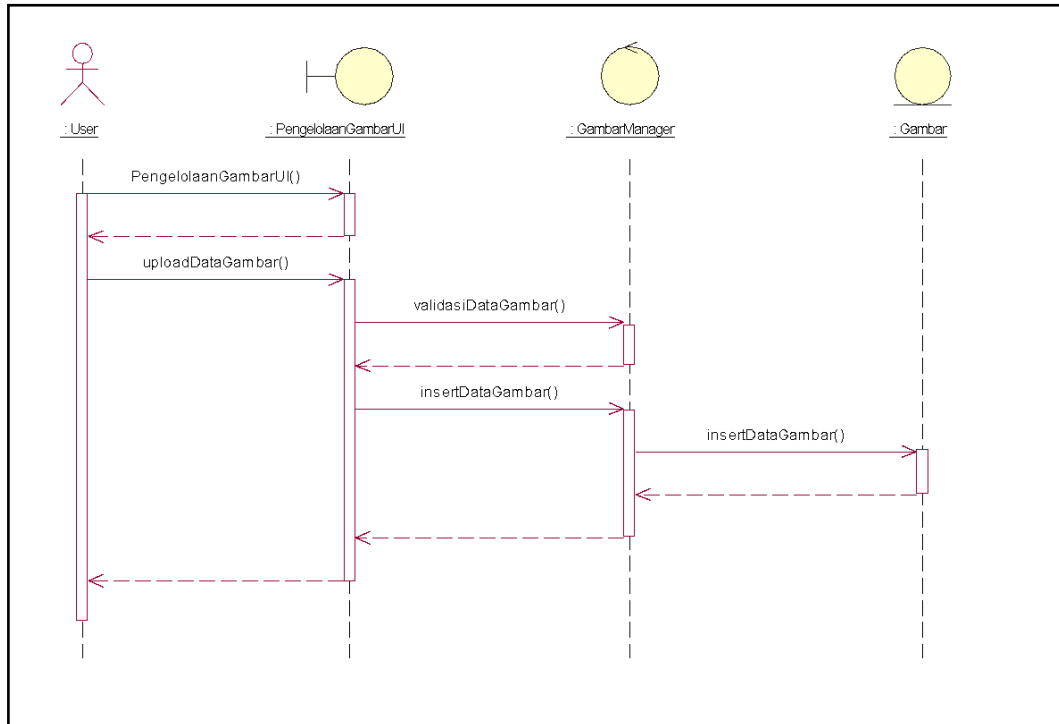
#### 5.1.4.3 Display Patient



Gambar 5.7 Sequence Diagram : Pengelolaan Data Patient RS - Display Patient

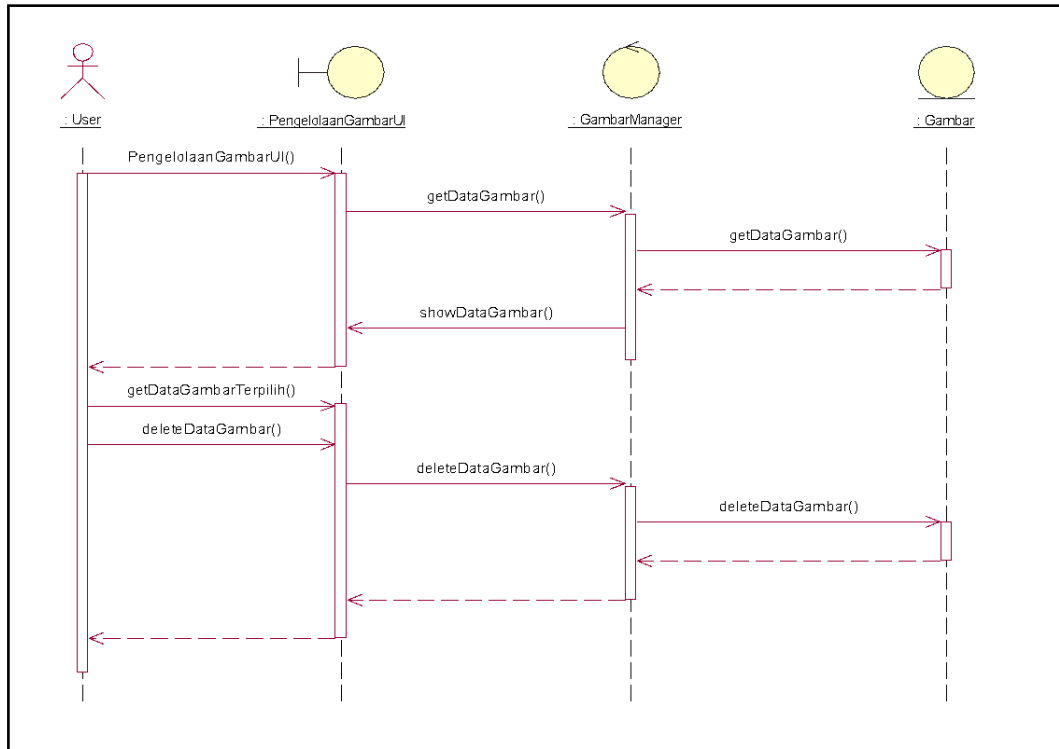
### 5.1.5 Pengelolaan Gambar

#### 5.1.5.1 Upload Gambar



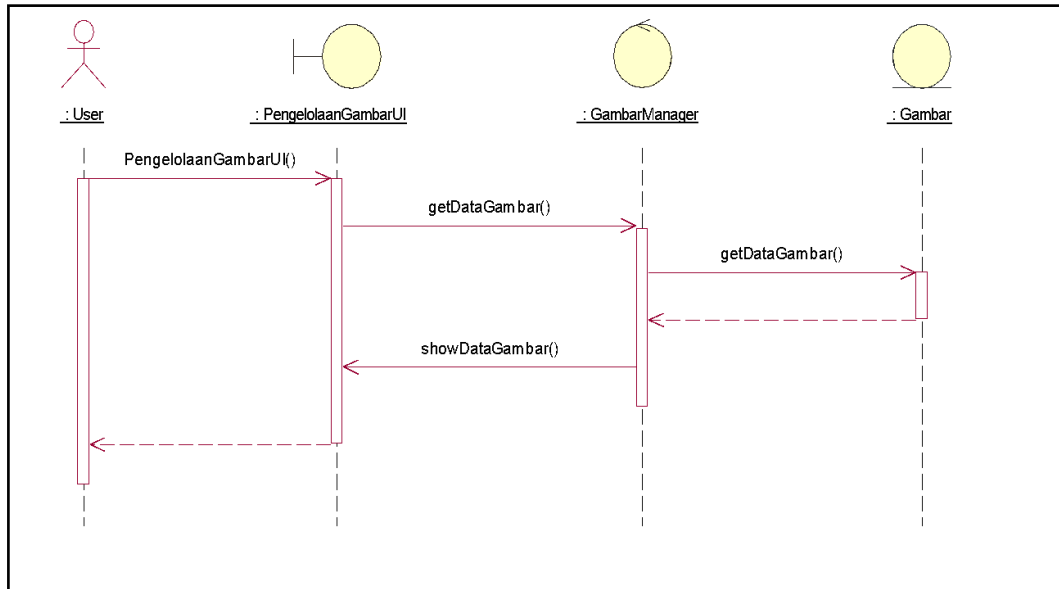
Gambar 5.10 Sequence Diagram : Pengelolaan Gambar - Upload Gambar

### 5.1.5.2 Delete Gambar



Gambar 5.11 Sequence Diagram : Pengelolaan Gambar - Delete Gambar

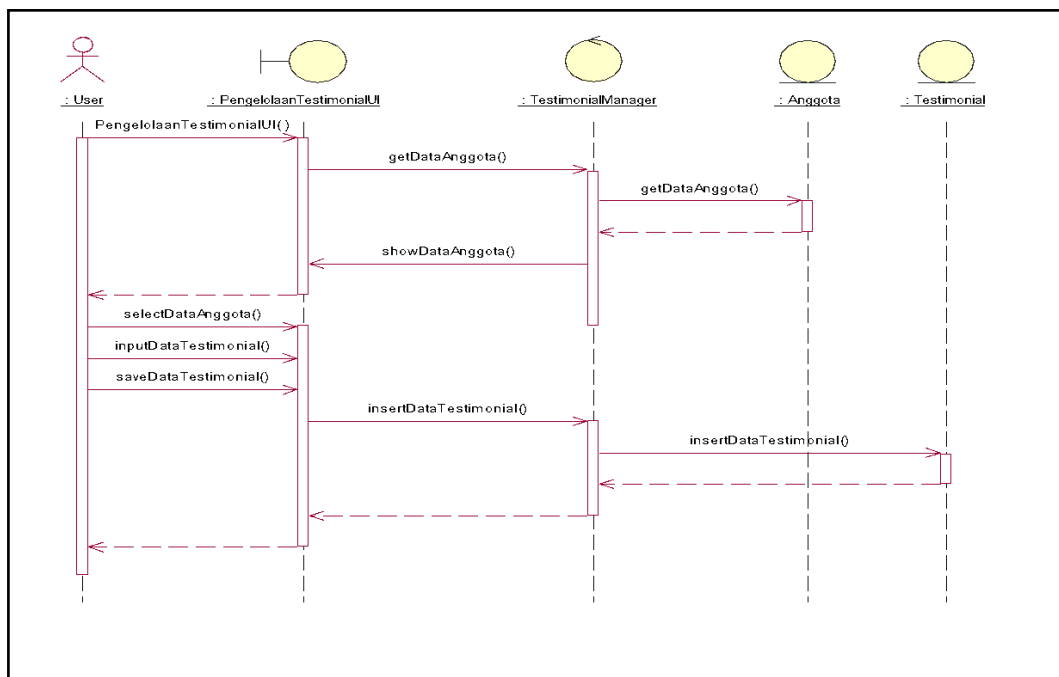
### 5.1.5.3 Display Gambar



Gambar 5.12 Sequence Diagram : Pengelolaan Gambar - Display Gambar

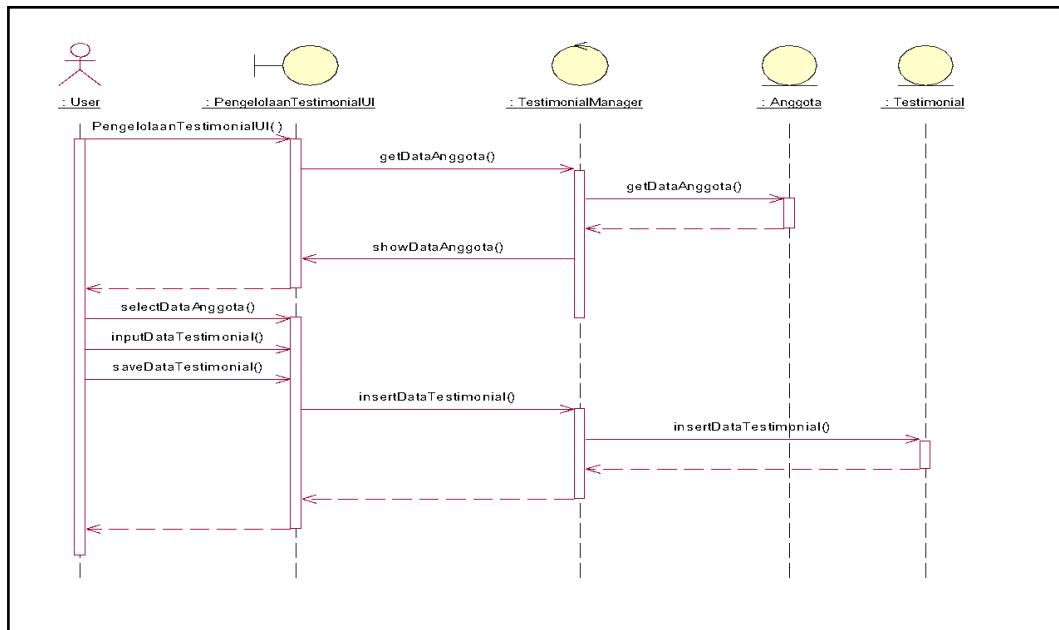
### 5.1.6 Pengelolaan Testimonial

#### 5.1.6.1 Add Testimonial



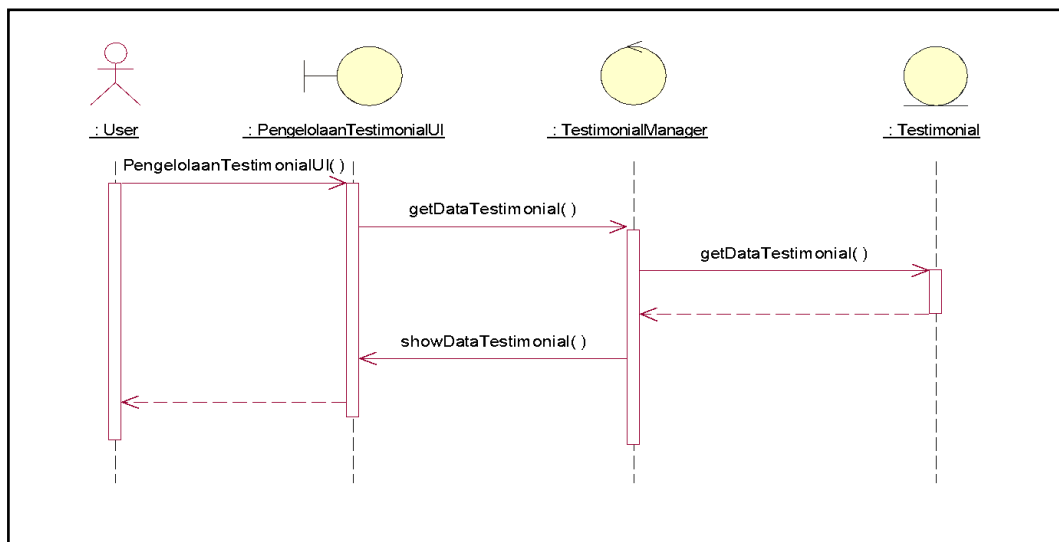
Gambar 5.13 Sequence Diagram : Pengelolaan Testimonial - Add Testimonial

### 5.1.6.2 Penerimaan Testimonial



Gambar 5.14 Sequence Diagram : Pengelolaan Testimonial - Penerimaan Testimonial

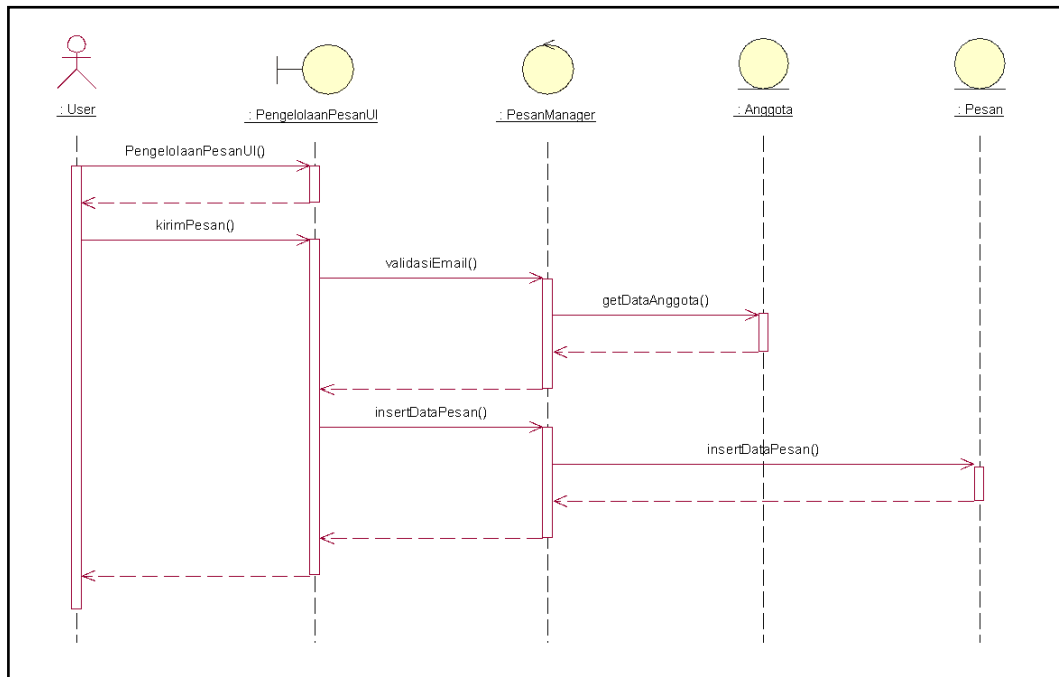
### 5.1.6.3 Display Testimonial



Gambar 5.15 Sequence Diagram : Pengelolaan Testimonial - Display Testimonial

## 5.1.7 Pengelolaan Pesan Antar Doctor RS

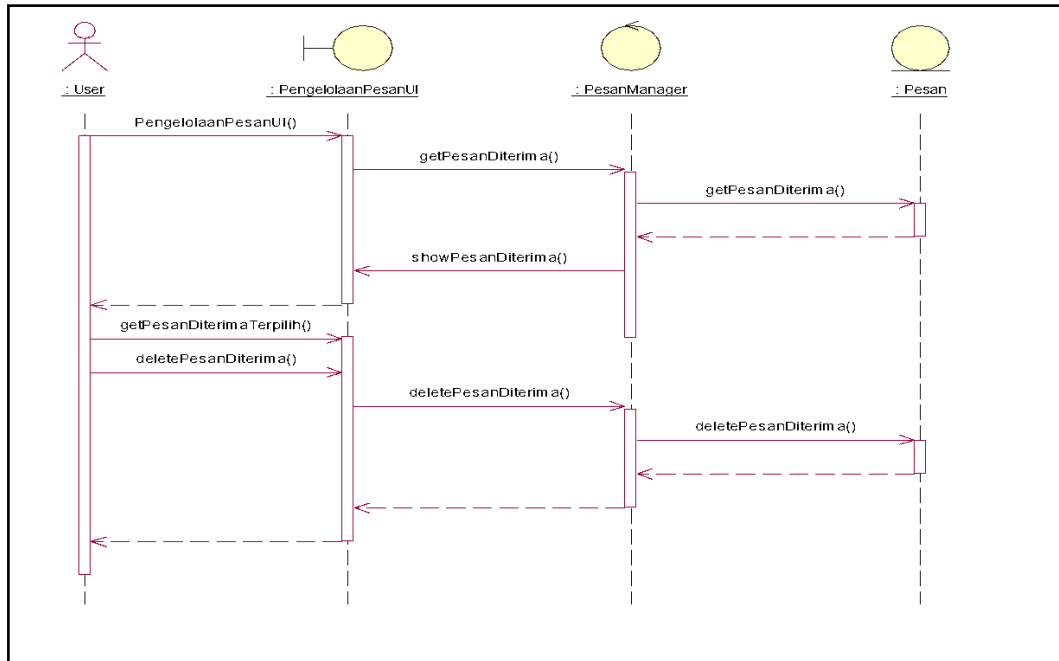
### 5.1.7.1 Kirim Pesan



**Gambar 5.16 Sequence Diagram : Pengelolaan Pesan antar Anggota Komunitas - Kirim Pesan**

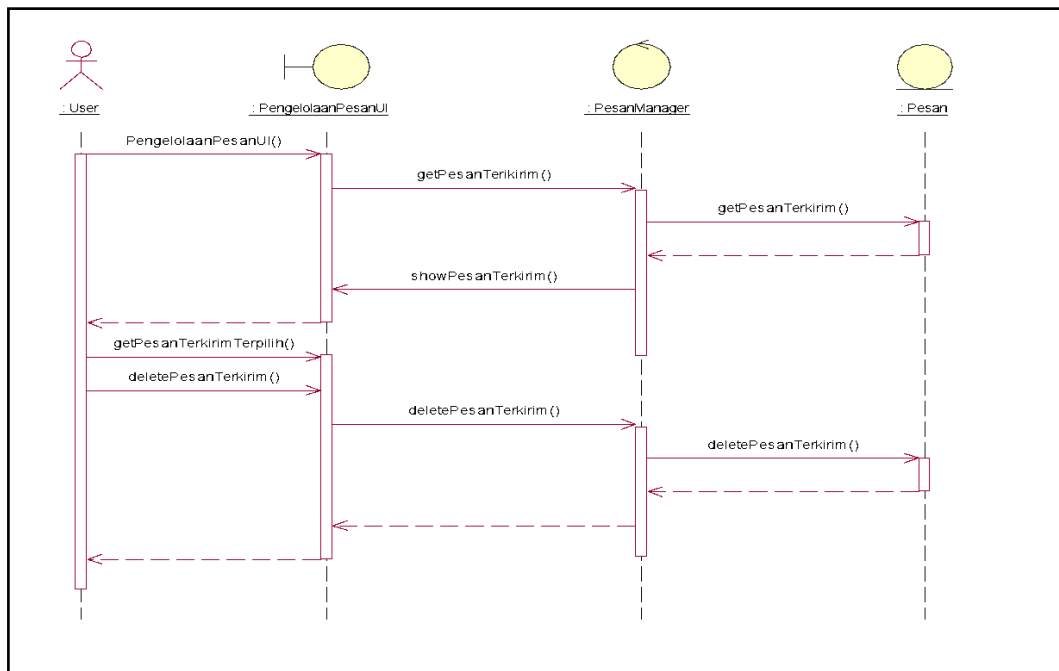


### 5.1.7.2 Delete Inbox



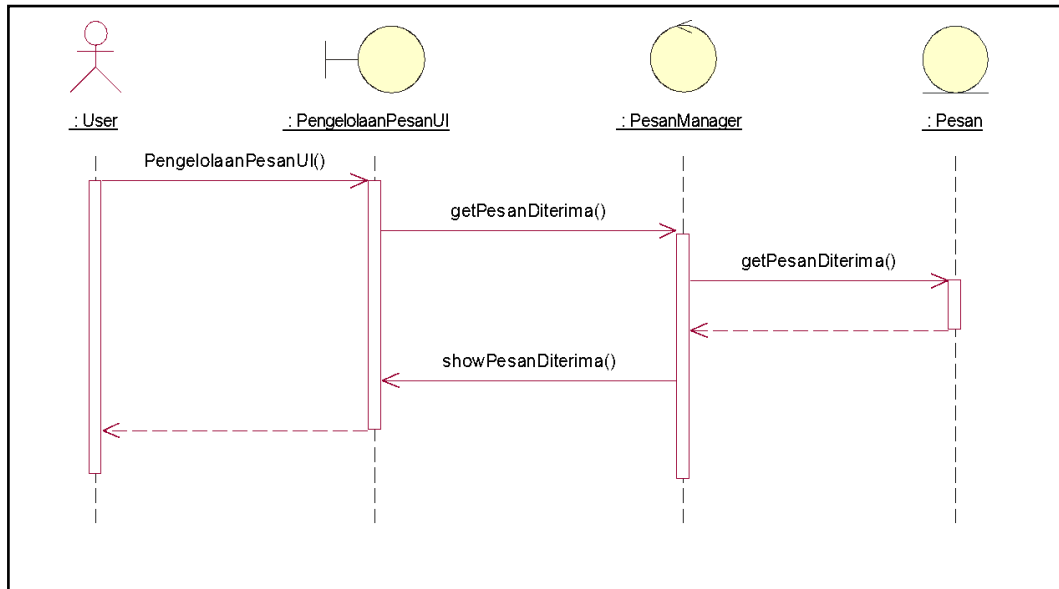
Gambar 5.17 Sequence Diagram : Pengelolaan Pesan antar Doctor RS - Delete Inbox

### 5.1.7.3 Delete Outbox



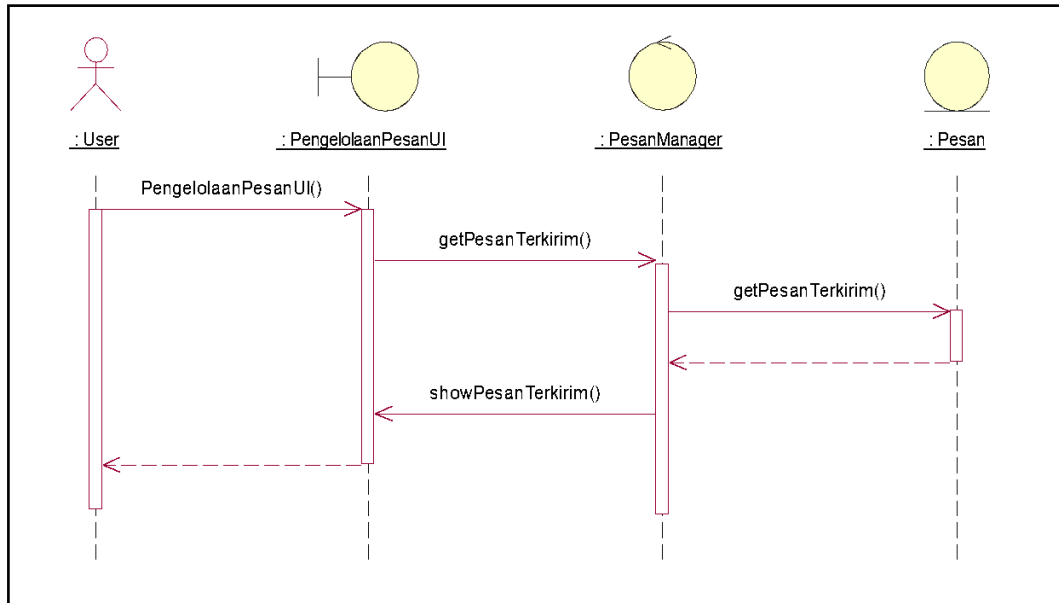
**Gambar 5.18 Sequence Diagram : Pengelolaan Pesan antar Doctor RS - Delete Outbox**

#### 5.1.7.4 Display Inbox



**Gambar 5.19 Sequence Diagram : Pengelolaan Pesan antar Doctor RS - Display Inbox**

### 5.1.7.5 Display Outbox



**Gambar 5.20 Sequence Diagram : Pengelolaan Pesan antar Doctor RS - Display Outbox**

## 5.2 Gambar 5.21 Class Diagram

### 5.3 Class Diagram Specific Descriptions

#### 5.3.1 Specific Design Class LoginHospital MNS

<b>LoginUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+LoginUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDataLogin() : String Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu login id dan password.</pre>	

#### 5.3.2 Specific Design Class PengelolaanDoctorUI

<b>PengelolaanDoctorUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+PengelolaanDoctorUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +showDataDoctor() Operasi ini digunakan untuk menampilkan data doctor yang sudah tersimpan di database. +saveDataDoctor(Doctor) Operasi ini digunakan untuk menyimpan data doctor ke database.</pre>	

#### 5.3.3 Specific Design Class PengelolaanPatientUI

<b>PengelolaanPatientUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+PengelolaanPatientUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</pre>	

+showDataPatient()

Operasi ini digunakan untuk menampilkan data patient yang sudah tersimpan di database.

+addPatient(Anggota)

Operasi ini digunakan untuk menyimpan data patient ke database.

+approvePatient(Patient)

Operasi ini digunakan untuk menerima patient lain menjadi patient

#### 5.3.4 Specific Design Class PencarianDoctorUI

PencarianDoctorUI	<<boundary>>
+PencarianDoctorUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getKunciPencarianDoctor() : String Operasi ini digunakan untuk mengambil kunci yang diinputkan oleh user yang akan digunakan untuk melakukan pencarian doctor.	

#### 5.3.5 Specific Design Class PengelolaanPesanUI

PengelolaanPesanUI	<<boundary>>
+PengelolaanPesanUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+ kirimPesan(Pesan) Operasi ini digunakan untuk mengirim pesan yang telah diinputkan oleh user ke anggota yang dituju.	
+showPesanDiterima() Operasi ini digunakan untuk menampilkan pesan yang diterima yang sudah tersimpan di database.	

```

+showPesanTer kirim()
Operasi ini digunakan untuk menampilkan pesan yang sudah
ter kirim yang tersimpan di database.
+getPesanDiterimaTer pilih() : Pesan
Operasi ini digunakan untuk mengambil data pesan diterima
yang dipilih oleh user.
+getPesanTer kirimTer pilih() : Pesan
Operasi ini digunakan untuk mengambil data pesan ter kirim
yang dipilih oleh user.
+deletePesanDiterima(Pesan)
Operasi ini digunakan untuk menghapus data pesan yang
diterima dari database.
+deletePesanTer kirim(Pesan)
Operasi ini digunakan untuk menghapus data pesan ter kirim
dari database.

```

### 5.3.6 Specific Design Class PengelolaanGambarUI

PengelolaanGambarUI	<<boundary>>
<pre> +PengelolaanGambarUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +showDataGambar() Operasi ini digunakan menampilkan data gambar yang sudah tersimpan di database. +getDataGambarTer pilih() : Gambar Operasi ini digunakan untuk mengambil data gambar yang dipilih oleh user. +uploadDataGambar(Gambar) Operasi ini digunakan untuk menyimpan data gambar yang telah diinputkan oleh user. +deleteDataGambar(Gambar) Operasi ini digunakan untuk menghapus data gambar yang sudah </pre>	

dipilih oleh user.

#### 5.3.7 Specific Design Class PengelolaanTestimonialUI

PengelolaanTestimonialUI	<<boundary>>
<pre>+PengelolaanTestimonialUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +showDataTestimonial() Operasi ini digunakan untuk menampilkan data testimonial milik user. +inputDataTestimonial(Testimonial) Operasi ini digunakan untuk menyimpan data testimonial yang telah diinputkan oleh user. +approveTestimonial(Testimonial) Operasi ini digunakan untuk menerima testimonial yang dikirimkan oleh anggota lain.</pre>	

#### 5.3.8 Specific Design Class PesanManager

PesanManager	<<Control>>
<pre>+PesanManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +validasiEmail() : Boolean Operasi ini digunakan untuk mengecek format email yang diinputkan user. +insertDataPesan(Pesan) Operasi ini digunakan untuk menyimpan data pesan ke database. +getPesanDiterima () : Pesan Operasi ini digunakan untuk mengambil data pesan diterima yang tersimpan di database.</pre>	

```

+getPesanTer kirim () : Pesan
Operasi ini digunakan untuk mengambil data pesan terkirim
yang tersimpan di database.
+deletePesanDiterima(Pesan)
Operasi ini digunakan untuk menghapus data pesan yang
diterima dari database.
+deletePesanTer kirim(Pesan)
Operasi ini digunakan untuk menghapus data pesan terkirim
dari database.

```

### 5.3.9 Specific Design Class GambarManager

GambarManager	<<control>>
<pre> +GambarManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +validasiDataGambar() : boolean Operasi ini digunakan untuk mengecek data gambar yang diinputkan user. +insertDataGambar(Gambar) Operasi ini digunakan untuk meyimpan data gambar ke database. +deleteDataGambar(Gambar) Operasi ini digunakan untuk menghapus data gambar di database. +getDataGambar() : Gambar Operasi ini digunakan untuk mengambil data gambar yang tersimpan di database. </pre>	

### 5.3.10 Specific Design Class TestimonialManager

TestimonialManager	<<control>>
<pre> +TestimonialManager() </pre>	



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

+getDataTestimonial() : Testimonial

Operasi ini digunakan untuk mengambil data testimonial yang tersimpan di database.

+insertDataTestimonial(Testimonial)

Operasi ini digunakan untuk menyimpan data testimonial ke database.

+updateDataTestimonial(Testimonial)

Operasi ini digunakan untuk mengupdate data testimonial di database.

### 5.3.11 Specific Design Class Pesan

Pesan	<<entity>>
<p>-ID_Pesan : Int Atribut ini digunakan untuk menyimpan data id dari pesan</p> <p>-Pengirim : Anggota Atribut ini digunakan untuk menyimpan data pengirim pesan</p> <p>-Penerima : Anggota Atribut ini digunakan untuk menyimpan data penerima pesan</p> <p>-Judul_Pesan : String Atribut ini digunakan untuk menyimpan data judul pesan</p> <p>-Tgl_Pesan : Date Atribut ini digunakan untuk menyimpan data tanggal pengiriman pesan</p> <p>-Flag_Status : String Atribut ini digunakan untuk menyimpan data status dari pesan</p>	
<p>+Pesan () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+insertDataPesan(Pesan) Operasi ini digunakan untuk menyimpan data pesan ke database.</p>	

```

+getPesanDiterima () : Pesan
Operasi ini digunakan untuk mengambil data pesan diterima
yang tersimpan di database.
+getPesanTer kirim () : Pesan
Operasi ini digunakan untuk mengambil data pesan terkirim
yang tersimpan di database.
+deletePesanDiterima(Pesan)
Operasi ini digunakan untuk menghapus data pesan yang
diterima dari database.
+deletePesanTer kirim(Pesan)
Operasi ini digunakan untuk menghapus data pesan terkirim
dari database.

```

### 5.3.12 Specific Design Class Gambar

GambarManager	<<entity>>
<p>-ID_Gambar : Int Atribut ini digunakan untuk menyimpan data id gambar</p> <p>-Uploader : Anggota Atribut ini digunakan untuk menyimpan data pengirim gambar</p> <p>-Owner : Anggota Atribut ini digunakan untuk menyimpan data penerima gambar</p> <p>-Nama_Gambar : String Atribut ini digunakan untuk menyimpan data nama gambar</p> <p>-Path_Gambar : String Atribut ini digunakan untuk menyimpan data path gambar</p> <p>-Tgl_Upload : Date Atribut ini digunakan untuk menyimpan data tanggal pengiriman gambar</p> <p>-Flag_Status : String Atribut ini digunakan untuk menyimpan data status dari gambar</p>	
<p>+Gambar() Default konstruktor, digunakan untuk inisialisasi semua</p>	

attribute dari kelas ini.

+insertDataGambar (Gambar)

Operasi ini digunakan untuk menyimpan data gambar ke database.

+deleteDataGambar (Gambar)

Operasi ini digunakan untuk menghapus data gambar di database.

+getDataGambar() : Gambar

Operasi ini digunakan untuk mengambil data gambar yang tersimpan di database.

### 5.3.13 Specific Design Class Testimonial

Testimonial	<<entity>>
<p>-ID_Testimonial : Int</p> <p>Atribut ini digunakan untuk menyimpan data id testimonial</p> <p>-Uploader : Anggota</p> <p>Atribut ini digunakan untuk menyimpan data pengirim testimonial</p> <p>-Owner : Anggota</p> <p>Atribut ini digunakan untuk menyimpan data penerima testimonial</p> <p>-Isi_Testimonial : String</p> <p>Atribut ini digunakan untuk menyimpan data isi dari testimonial</p> <p>-Tgl_Kirim : Date</p> <p>Atribut ini digunakan untuk menyimpan data tanggal pengiriman testimonial</p> <p>-Flag_Status : String</p> <p>Atribut ini digunakan untuk menyimpan data status dari testimonial</p>	
<p>+Testimonial()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

```
+getDataTestimonial() : Testimonial
```

Operasi ini digunakan untuk mengambil data testimonial yang tersimpan di database.

```
+insertDataTestimonial(Testimonial)
```

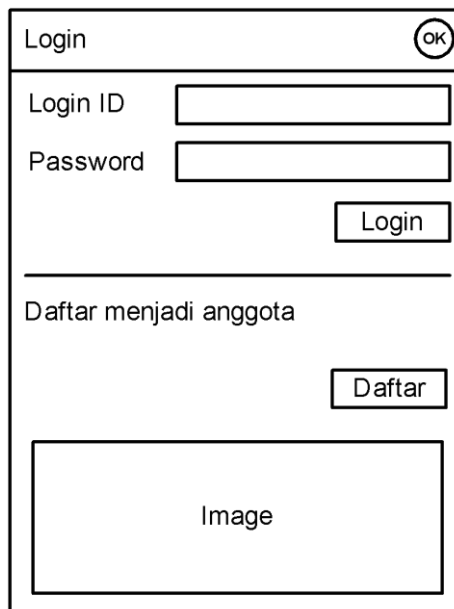
Operasi ini digunakan untuk menyimpan data testimonial ke database.

```
+updateDataTestimonial(Testimonial)
```

Operasi ini digunakan untuk mengupdate data testimonial di database.

## 6 Deskripsi Perancangan Antarmuka

### 6.1 Login



The diagram shows a login window titled "Login" with an "OK" button in the top right corner. Inside the window, there are two input fields: "Login ID" and "Password". To the right of the "Password" field is a "Login" button. Below these fields is a horizontal line, followed by the text "Daftar menjadi anggota". To the right of this text is a "Daftar" button. At the bottom of the window is a large rectangular area labeled "Image".

Gambar 6.1 Rancangan Antarmuka Login

Antarmuka ini digunakan untuk melakukan proses login ke dalam sistem dan melakukan pendaftaran bagi doctor/patient baru. Untuk mendapat akses masuk ke dalam sistem, doctor/patient haru menginputkan login id dan password dengan benar pada textbox yang telah disediakan. Pada saat tombol login ditekan, sistem akan mengecek 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 sistem, sebaliknya jika id dan password salah atau tidak cocok maka akan diberikan pesan peringatan.

Apabila tombol daftar ditekan maka proses akan dilanjutkan ke antarmuka pengelolaan doctor / patient.

## 6.2 Pengelolaan Patient

Data Anggota		
Email	<input type="text"/>	
Password	<input type="text"/>	
Nama	<input type="text"/>	
Jenis Kelamin	<input type="text"/>	▼
Tempat Lahir	<input type="text"/>	
Tgl Lahir	<input type="text"/>	▼
Tempat Tinggal	<input type="text"/>	
Negara	<input type="text"/>	▼
First Page	Second Page	Third Page

Data Anggota		
Kode Pos	<input type="text"/>	
Pekerjaan	<input type="text"/>	
Sekolah	<input type="text"/>	
Buku Favorit	<input type="text"/>	
Film Favorit	<input type="text"/>	
First Page	Second Page	Third Page

Data Anggota		
Music Favorit	<input type="text"/>	
Acara TV Favorit	<input type="text"/>	
Tentang Saya	<input type="text"/>	
		Submit
First Page	Second Page	Third Page

**Gambar 6.3 Rancangan Antarmuka Pengelolaan Patient**

Antarmuka ini digunakan untuk melakukan pengelolaan Patient, termasuk didalamnya tambah patient baru, edit data patient lama, dan display

data anggota. Terdapat tiga buah tab yang berisi data pribadi dari patient atau user, seperti email, password, nama, jenis kelamin, tempat lahir, dll. Pada tab terakhir terdapat sebuah tombol submit yang digunakan untuk menyimpan data patient yang baru atau data patient yang sudah diedit ke database.

### 6.3 Pengelolaan patient - Tambah patient

**Gambar 6.4 Rancangan Antarmuka Pengelolaan Patient - Tambah Patient**

Antarmuka ini merupakan antarmuka yang digunakan untuk melakukan penambahan patient dalam komunitas. User dapat mengirimkan permintaan untuk menjadi patient ke lebih dari satu patient. Caranya dengan menginputkan lebih dari satu email patient, yang akan dijadikan teman, pada textbox yang telah disediakan. Pada saat tombol submit ditekan, data email yang diinputkan akan divalidasi. Apabila data

#### 6.4 Pengelolaan Patient - Pending Friend

**Gambar 6.6 Rancangan Antarmuka Pengelolaan Patient - Pending Friend**



## 6.5 Pencarian Patient – Input Kriteria



Find Anggota	
Nama	<input type="text"/>
Jenis Kelamin	<input type="text" value="▼"/>
Tempat Lahir	<input type="text"/>
Tempat Tinggal	<input type="text"/>
Negara	<input type="text" value="▼"/>
Sekolah	<input type="text"/>
Hobi	<input type="text"/>
<input type="button" value="Search"/>	
Kriteria Pencarian	Hasil Pencarian

**Gambar 6.8 Rancangan Antarmuka Pencarian Patient – Input Kriteria**

Antarmuka ini digunakan untuk melakukan pencarian patient dalam komunitas berdasarkan kriteria yang diinginkan. User dapat mencari patient dalam komunitas berdasarkan nama, jenis kelamin, tempat lahir, negara, dll. Pada saat tombol search ditekan, sistem akan mencari data patient yang sudah tersimpan di database yang sesuai dengan kriteria yang diinputkan. Kemudian hasilnya akan ditampilkan di antarmuka hasil pencarian patient.

## 6.6 Pencarian Patient - Hasil Pencarian

The interface is titled "Find Anggota" with an "OK" button in the top right corner. It features a table with two columns: "Nama" (Name) and "Email". Below the table, there is a "View Detail" button. The table contains four rows of data, with the first row highlighted. A scroll bar is visible on the right side of the table, indicating that there are more results. At the bottom of the interface, there are two tabs: "Kriteria Pencarian" (Search Criteria) and "Hasil Pencarian" (Search Results), with "Hasil Pencarian" being the active tab.

Nama	Email
_____	_____
_____	_____
_____	_____
_____	_____

View Detail

Kriteria Pencarian Hasil Pencarian

Gambar 6.9 Rancangan Antarmuka Pencarian Patient - Hasil Pencarian

Antarmuka ini digunakan untuk menampilkan hasil pencarian patient dalam komunitas sesuai dengan kriteria yang diinputkan sebelumnya. Data patient hasil pencarian tidak ditampilkan secara lengkap. Untuk melihat lebih detail, user dapat memilih salah satu data patient yang diinginkan dari daftar kemudian memilih menu view detail.

## 6.7 Pengelolaan Pesan – Kirim Pesan

The wireframe shows a window titled "Pesan" with a standard "OK" button in the top right corner. The main content area contains three input fields: "To" (recipient), "Subject" (topic), and a large text area for the message body. A "Send" button is positioned to the right of the message body. At the bottom of the window, there are three buttons: "Compose", "Inbox", and "Outbox".

**Gambar 6.10 Rancangan Antarmuka Pengelolaan Pesan – Kirim Pesan**

Antarmuka ini digunakan untuk melakukan pengiriman pesan ke patient lain dalam RS. Terdapat tiga buah textbox, masing-masing untuk menginputkan email anggota tujuan, subject atau judul dari pesan, dan isi dari pesan. Ketika tombol send ditekan, sistem akan melakukan validasi terhadap data yang diinputkan. Apabila validasi sukses, sistem akan menyimpan dua buah data pesan ke database. Satu data pesan sebagai pesan terkirim untuk user sebagai pengirim, dan satu lagi sebagai pesan diterima untuk anggota yang dikirim pesan.

## 6.8 Pengelolaan Pesan - Inbox

The wireframe shows a window titled "Pesan" with an "OK" button in the top right corner. The main content area is a table with three columns: "From", "Subject", and an empty column. The table contains four rows of placeholder text. A "View Detail" button is positioned over the third row. The table has a scroll bar on the right side, indicated by a triangle at the bottom. Below the table are two buttons: "Check All" and "Delete". At the bottom of the window are three buttons: "Compose", "Inbox", and "Outbox".

From	Subject	
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

View Detail

Check All Delete

Compose Inbox Outbox

Gambar 6.11 Rancangan Antarmuka Pengelolaan Pesan - Inbox

Antarmuka ini digunakan untuk menampilkan daftar pesan yang diterima oleh user dari Staff lain (inbox). Untuk melihat detail dari pesan, user dapat memilih pesan yang diinginkan dari daftar dan memilih menu view detail. Tombol check all digunakan untuk menandai semua pesan pada daftar, yang selanjutnya dapat dihapus dengan menekan tombol delete. Pada saat tombol delete di tekan, sistem akan menghapus data pesan yang dipilih atau ditandai tersebut dari database.

## 6.9 Pengelolaan Pesan - Outbox

Pesanan		(OK)
To	Subject	
<input type="text"/>	<input type="text"/>	▲
<input type="text"/>	<input type="text"/>	
<input type="text"/>	<input type="text"/>	
<input type="text"/> View Detail	<input type="text"/>	
		▼

Check All	Delete
-----------	--------

Compose	Inbox	Outbox
---------	-------	--------

**Gambar 6.12 Rancangan Antarmuka Pengelolaan Pesan - Outbox**

Antarmuka ini digunakan untuk menampilkan daftar pesan yang sudah atau pernah dikirim oleh user ke anggota lain (outbox). Untuk melihat detail dari pesan, user dapat memilih pesan yang diinginkan dari daftar dan memilih menu view detail. Tombol check all digunakan untuk menandai semua pesan pada daftar, yang selanjutnya dapat dihapus dengan menekan tombol delete. Pada saat tombol delete di tekan, sistem akan menghapus data pesan yang dipilih atau ditandai tersebut dari database.

#### 6.10 Display Detail Pesan

```
graph TD
    subgraph Window [Detail Pesan]
        direction TB
        Title[Detail Pesan (OK)]
        ToFrom[To / From]
        Subject[Subject]
        Content[ ]
        Forward[Forward]
        Replay[Replay]
    end
```

**Gambar 6.13 Rancangan Antarmuka Display Detail Pesan**

Antarmuka ini digunakan untuk menampilkan detail dari pesan yang sudah dipilih sebelumnya. Data yang ditampilkan adalah pengirim atau penerima

pesan, judul pesan, dan isi pesan. Apabila detail pesan yang ditampilkan adalah detail pesan yang diterima (inbox), user dapat membalas atau melanjutkan pesan tersebut. Untuk membalas pesan, user dapat menekan tombol reply. Sedangkan untuk melanjutkan pesan, user dapat menekan tombol forward. Pada saat tombol reply ditekan, user akan kembali ke antarmuka pengiriman pesan dimana data email tujuan, subject, dan isi pesan sudah terisi secara otomatis. Begitu juga pada saat tombol forward ditekan, user akan kembali ke antarmuka pengiriman pesan dimana data subject dan isi pesan sudah terisi secara otomatis. Apabila detail pesan yang ditampilkan adalah detail pesan yang dikirim (outbox), user tidak dapat membalas atau melanjutkan pesan tersebut.

Antarmuka ini digunakan untuk menampilkan detail dari bulletin yang sudah dipilih sebelumnya. Data yang ditampilkan adalah pengirim bulletin dan isi dari bulletin.

#### 6.11 Pengelolaan Gambar - Upload Gambar

The wireframe shows a window titled "Picture Gallery" with an "OK" button in the top right corner. Inside the window, there is a "Caption" label followed by a text input field. Below this is a "Picture File" label, followed by another text input field and a "Browse" button. A large rectangular area labeled "Image" is positioned below the file input. At the bottom left of the main content area is a "Submit" button. The bottom of the window features a horizontal bar with three buttons: "Upload", "View", and "Approve/Reject".

Gambar 6.17 Rancangan Antarmuka Pengelolaan Gambar - Upload Gambar



Antarmuka ini digunakan untuk mengupload gambar milik user atau mengirim gambar ke teman dari user. User harus menginputkan nama dari gambar dan file gambar yang akan diupload. Untuk mencari file gambar yang akan diupload, user dapat menekan tombol browse. Selanjutnya path gambar beserta gambar yang sudah dipilih akan ditampilkan. Pada saat tombol submit ditekan, sistem akan menyimpan data gambar seperti nama gambar, pengirim, penerima, dan pathnya ke database, selanjutnya sistem juga akan menyimpan file gambar ke server.

## 6.12 Pengelolaan Gambar - Penerimaan/Penolakan

The wireframe shows a window titled "Picture Gallery" with an "OK" button in the top right corner. Inside the window is a table with two columns: "Name" and "From". Below the table is a context menu with three options: "View Detail", "Approve", and "Reject". At the bottom of the window are three buttons: "Upload", "View", and "Approve/Reject".

Name	From

View Detail  
 Approve  
 Reject

Upload
View
Approve/Reject

**Gambar 6.18 Rancangan Antarmuka Pengelolaan Gambar - Penerimaan/Penolakan**

Antarmuka ini digunakan untuk menerima atau menolak gambar yang dikirimkan oleh teman dari user

di dalam komunitas. Sistem akan menampilkan daftar gambar yang dikirimkan oleh teman dari user, dan user dapat menolak atau menerima setiap gambar tersebut. Apabila user menolak salah satu gambar yang dikirimkan dengan memilih menu Reject, maka sistem akan menghapus data gambar tersebut, baik yang berada di server maupun di database. Sebaliknya jika user menerima salah satu gambar yang dikirimkan dengan memilih menu Approve, maka sistem akan mengupdate status dari data gambar tersebut di database, dan memasukkannya ke galeri gambar dari user.

### 6.13 Pengelolaan Gambar - Display Gambar

Name	From

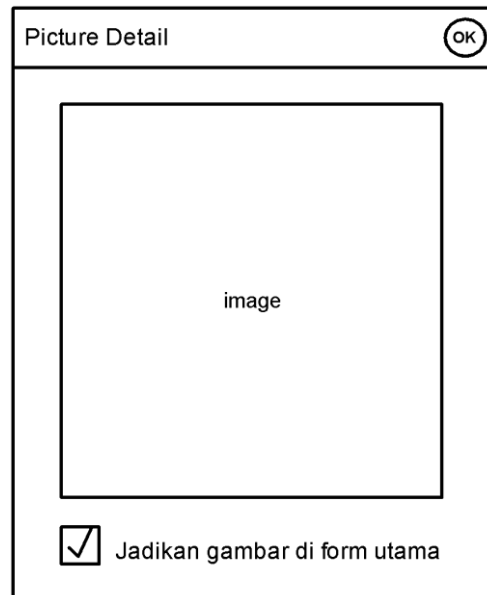
View Detail  
 Delete  
 Download

Upload      View      Approve/Reject

**Gambar 6.19 Rancangan Antarmuka Pengelolaan Gambar - Display Gambar**

Antarmuka ini digunakan untuk menampilkan daftar gambar yang dimiliki oleh user, baik yang diupload oleh user, maupun yang dikirimkan oleh teman dari user. User dapat melihat detail dari setiap gambar dengan memilih menu view detail. Selain itu, user juga dapat menghapus gambar dan mendownload gambar yang diinginkan dari server ke penyimpanan lokal. Apabila user memilih untuk menghapus gambar yang dipilih, sistem akan menghapus data gambar di database, dan file gambar di server. Apabila user memilih untuk mendownload gambar, sistem akan menampilkan antarmuka untuk memilih tempat penyimpanan di penyimpanan lokal, selanjutnya sistem akan menyimpan file gambar yang dipilih ke tempat penyimpanan yang telah ditentukan tersebut.

#### 6.14 Display Detail Gambar



**Gambar 6.20 Rancangan Antarmuka Display Detail Gambar**

Antarmuka ini digunakan untuk menampilkan detail gambar yang telah dipilih sebelumnya. Selain itu user juga dapat menentukan agar gambar yang ditampilkan tersebut menjadi gambar yang akan ditampilkan di Main Form, dengan cara mencentang checkbox di bawah gambar tersebut.