

## **BAB VI**

### **KESIMPULAN DAN SARAN**

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

Setelah aplikasi PWGCI ini selesai diimplementasikan maka dapat ditarik beberapa kesimpulan, yaitu:

1. Perangkat Lunak PWGCI telah berhasil dibangun sebagai *website* gereja St. Yohanes Rasul Somohitan dengan *Framework CodeIgniter*.
2. Perangkat Lunak PWGCI telah berhasil dibangun sebagai aplikasi *web* untuk pengelolaan profil, kebaktian, petugas kebaktian, lingkungan, kumpulan, pengumuman dan surat pembaca.
3. Dari hasil pengujian dapat diketahui bahwa fungsi-fungsi yang disediakan oleh perangkat lunak PWGCI berjalan dengan benar dan sesuai dengan yang diharapkan.

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

Penulis ingin memberikan beberapa saran untuk pengembangan lebih lanjut perangkat lunak PWGCI ini:

1. Penulis menyarankan agar sistem bisa dikembangkan lebih lanjut dengan fungsionalitas yang baik dan komplek, seperti pengelolaan pendaftaran baptis, pendaftaran krisma, sensus jemaat dan penanganan lupa *password*.
2. Penulis menyarankan agar sistem ini mempunyai hak akses tertentu.
3. Penulis menyarankan agar sistem dapat dijalankan menggunakan *mobile device*

## DAFTAR PUSTAKA

Hartono, Jogiyanto, 2005, *Analisis & Desain: Sistem Informasi*, Andi, Yogyakarta.

Prasetyo, Herlambang, 2009, *CodeIgniter Lightweight PHP Framework*, <http://herlambangprasetyo.net/2009/02/CodeIgniter-lightweight-php-framework>, diakses pada tanggal 15 Oktober 2010.

Rizky, Dhanta, 2009, *Kamus Istilah Komputer Grafis dan Internet*, Indah, Surabaya.

Saputra, Agus, 2011, *Trik Kolaborasi CodeIgniter dan JQuery*, Lokomedia, Yogyakarta.

Kristanto, Andi, 2003, *Perancangan Sistem Informasi dan Aplikasinya*, Gaya Media, Klaten.

Laura, Lemay, 2001, *Desain Grafik dan Halaman Web*, Informatika, Bandung.

Supono, 2010, *CodeIgniter Framework PHP*, [supono.wordpress.com/2010/04/15/CodeIgniter-framework-php/](http://supono.wordpress.com/2010/04/15/CodeIgniter-framework-php/), diakses pada tanggal 5 Maret 2012.

Upton, David, 2007, *CodeIgniter for Rapid PHP Application Development*, Wiley Publishing, Inc., Indianapolis.

Whitten, Jeffery, 2004, *Metode Desain dan Analisis Sistem*, Andi, Yogyakarta.



# DPPL

## DESKRIPSI PERANCANGAN PERANGKAT LUNAK PWGCI

### PEMBANGUNAN WEBSITE GEREJA SOMOHITAN MENGUGUNAKAN FRAMEWORK CODEIGNITER

Untuk:

Universitas Atmajaya Yogyakarta

Dipersiapkan oleh:

Yulianus Andri Ardiyanto / 4980

Program Studi Teknik Informatika - Fakultas Teknologi  
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		<b>DPPL-PWGCI</b>		1/77
	Fakultas Teknologi Industri			

## DAFTAR PERUBAHAN

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

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

## Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

## Daftar Isi

1	Pendahuluan.....	5
1.1	Tujuan.....	5
1.2	Ruang Lingkup.....	5
1.3	Definisi dan Akronim.....	6
1.4	Referensi.....	6
2	Perancangan Sistem.....	7
2.1	Perancangan Arsitektur.....	7
2.2	Perancangan Rinci.....	8
2.2.2	Class Diagrams.....	32
2.2.3	Spesifikasi Deskripsi Class Diagram.....	33
3	Deskripsi Dekomposisi.....	49
3.1	Dekomposisi Data.....	49
3.1.1	Deskripsi Entitas Data User.....	49
3.1.2	Deskripsi Entitas Data Jemaat.....	49
3.1.3	Deskripsi Entitas Data Kebaktian.....	49
3.1.4	Deskripsi Entitas Data ref_petugas.....	50
3.1.5	Deskripsi Entitas Data tr_petugas.....	50
3.1.6	Deskripsi Entitas Data Pengumuman.....	51
3.1.7	Deskripsi Entitas Data Profil.....	51
3.1.8	Deskripsi Entitas Data Kumpulan.....	51
3.1.9	Deskripsi Entitas Data Lingkungan.....	51
3.1.10	Deskripsi Entitas Data Komentar.....	52
4.	Perancangan Antarmuka.....	53
4.1.	Sketsa UI dan Deskripsinya.....	53

## **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 PWGCI dikembangkan dengan tujuan untuk:

1. Mengelola kebaktian.
2. Mengelola petugas kebaktian.
3. Mengelola jemaat.
4. Mengelola pengumuman.
5. Mengelola profil.
6. Mengelola kumpulan.
7. Mengelola lingkungan.
8. Mengelola surat pembaca.

Dan berjalan pada lingkungan dengan platform Windows

### 1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga <i>Software Design Description</i> (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
PWGCI	Pembangunan Web Gereja Somohitan Menggunakan Framework CodeIgniter.

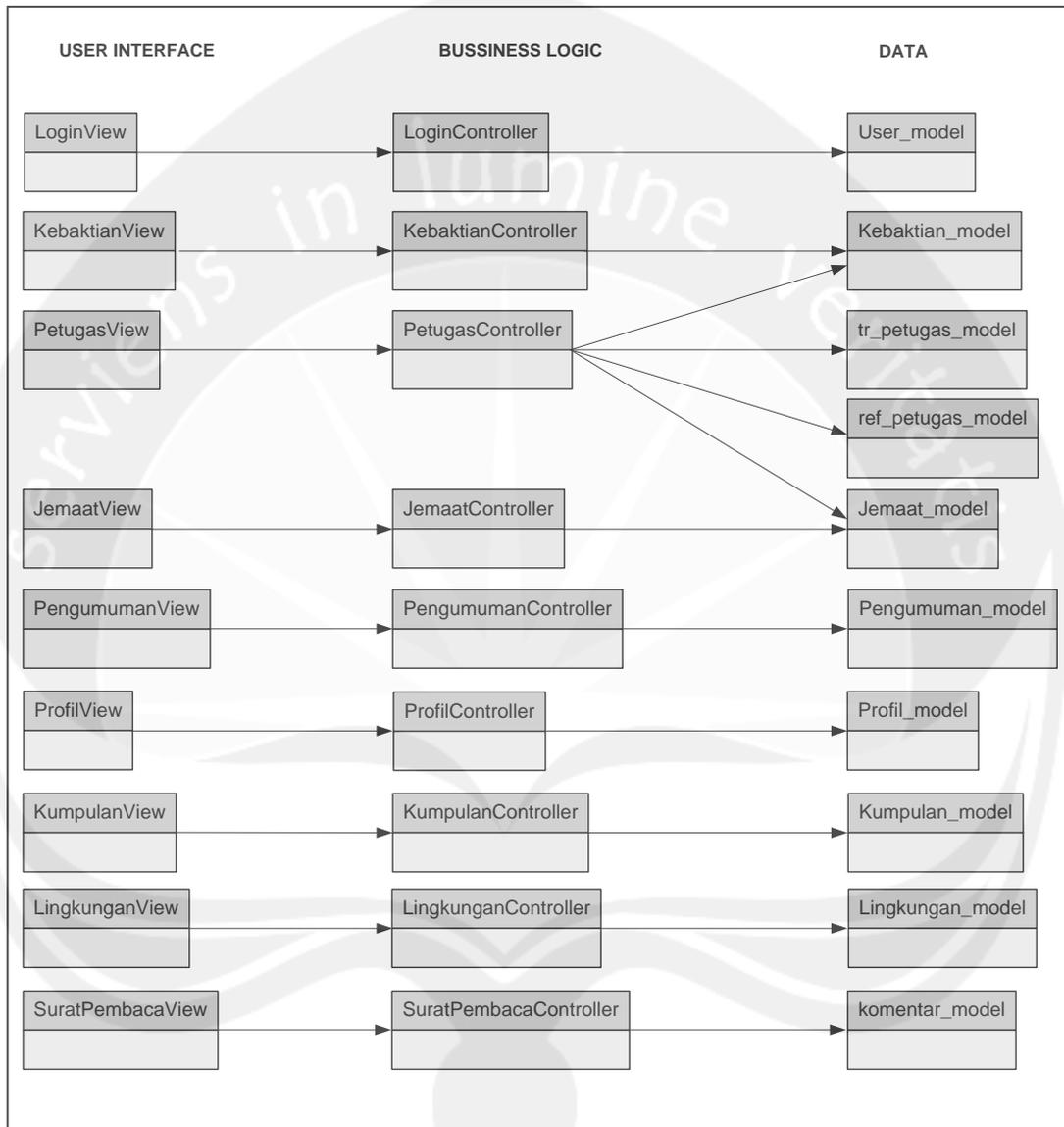
### 1.4 Referensi

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. SKPL Pembangunan Web Gereja Somohitan Menggunakan Codeigniter.

## 2 Perancangan Sistem

### 2.1 Perancangan Arsitektur

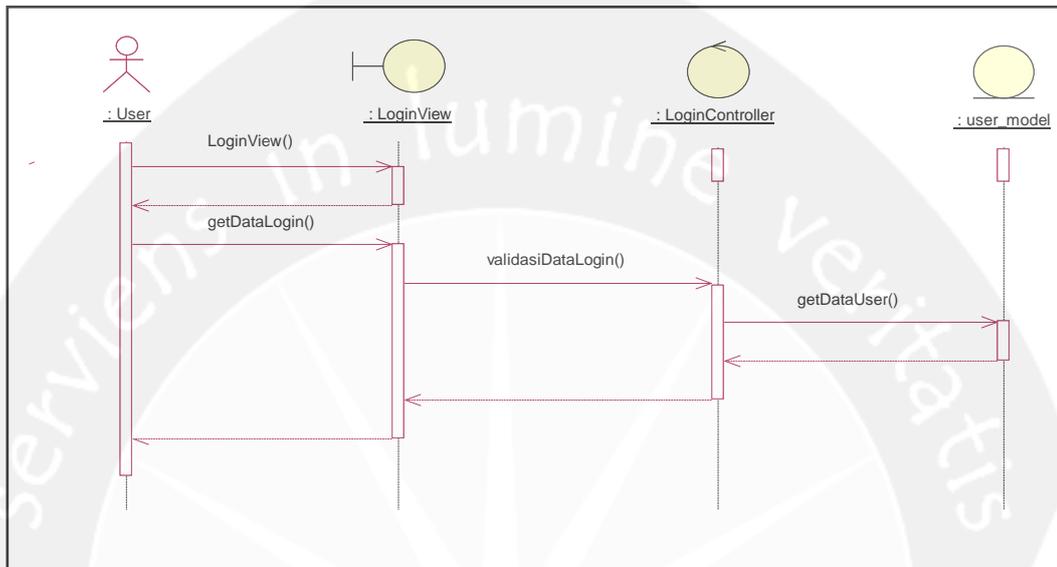


Gambar 2.1 Rancangan Arsitektur Aplikasi PWGCI

## 2.2 Perancangan Rinci

### 2.2.1 Sequence Diagram

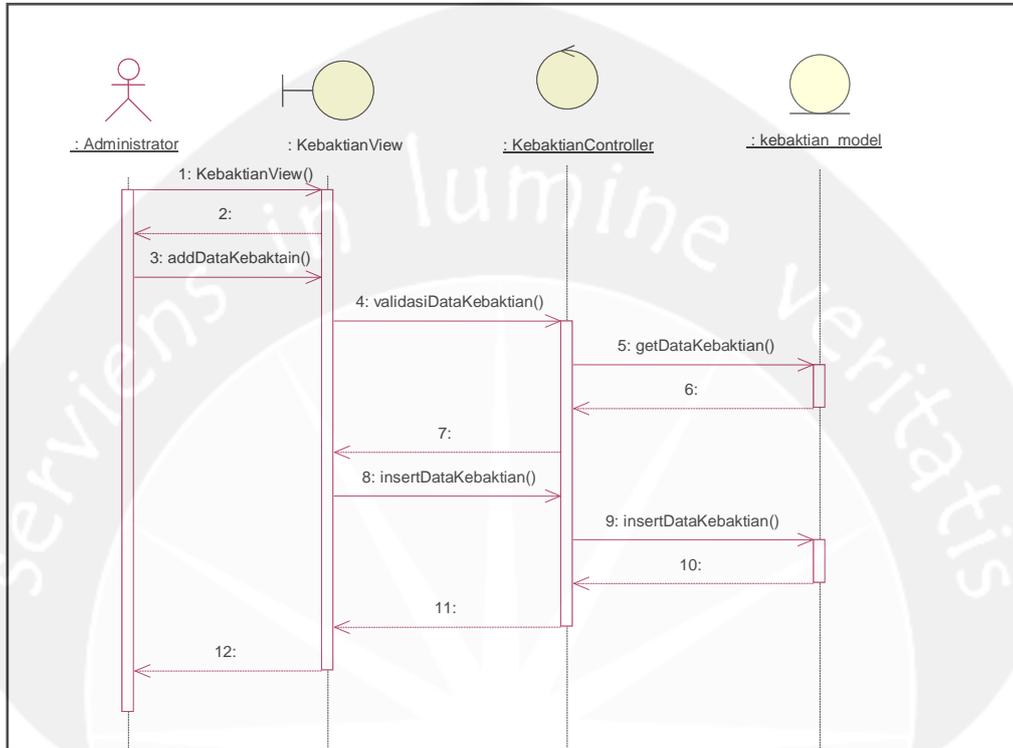
#### 2.2.1.1 Login User



Gambar 2.2 Sequence Diagram: Login User

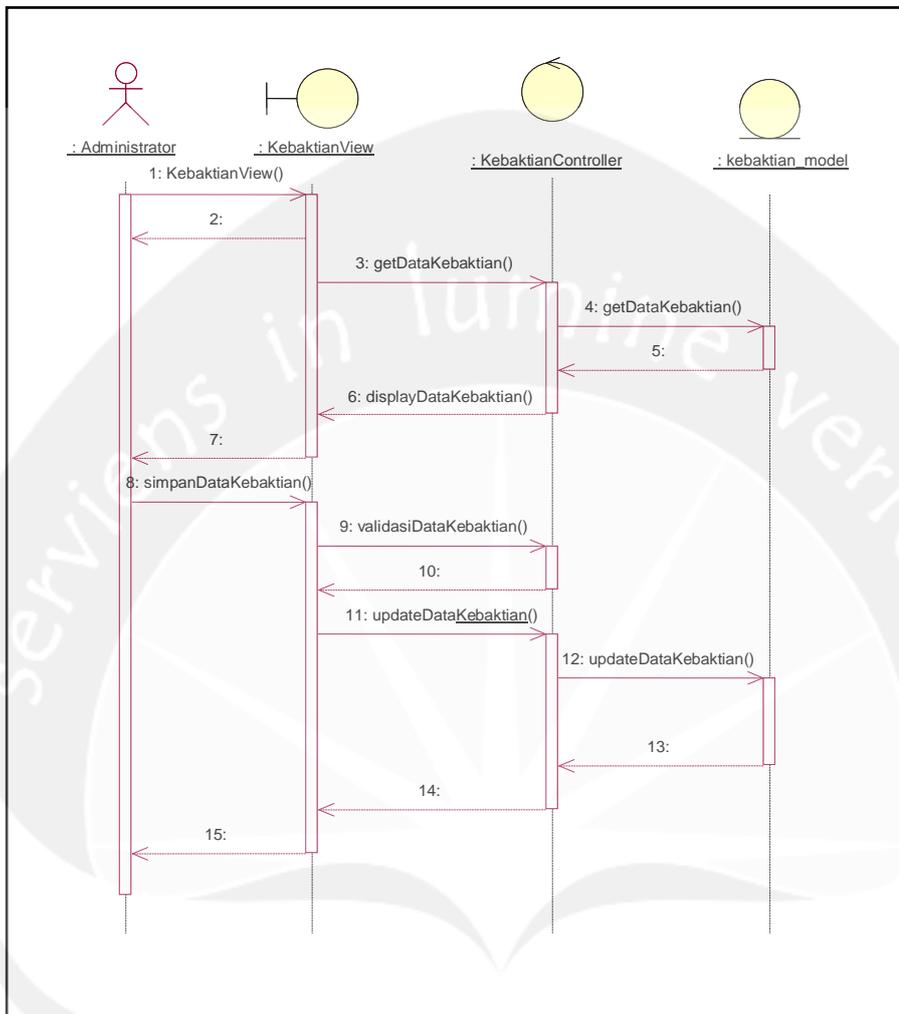
## 2.2.1.2 Mengelola Kebaktian

### 2.2.1.2.1 Entry Data Kebaktian



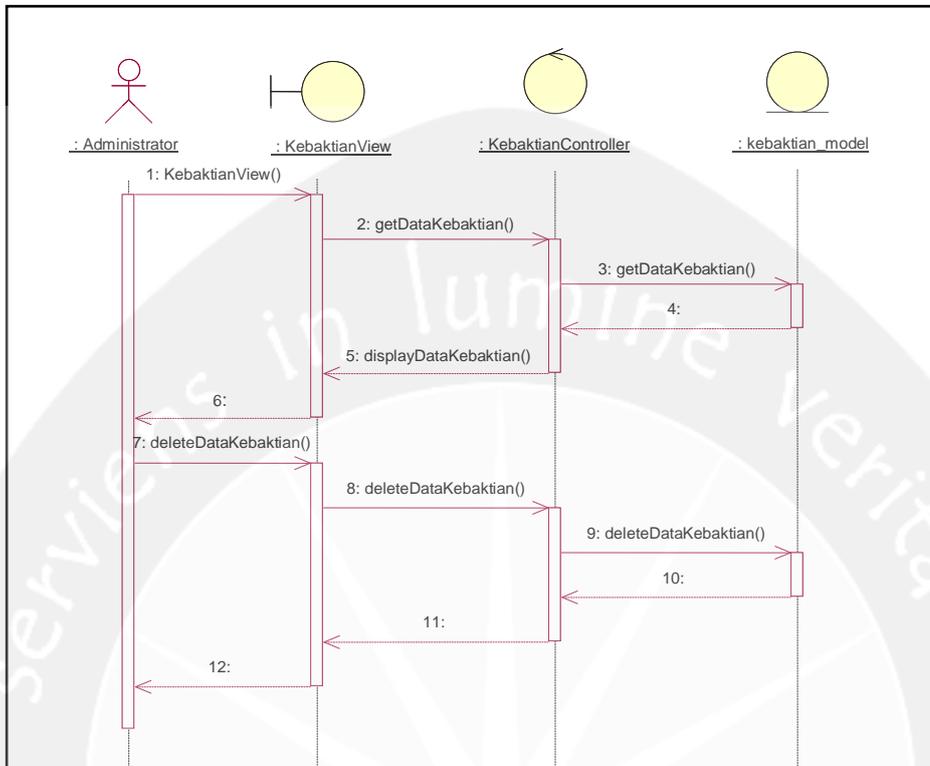
Gambar 2.3 Sequence Diagram : Mengelola Kebaktian - Entry Data Kebaktian

### 2.2.1.2.2 Edit Data Kebaktian



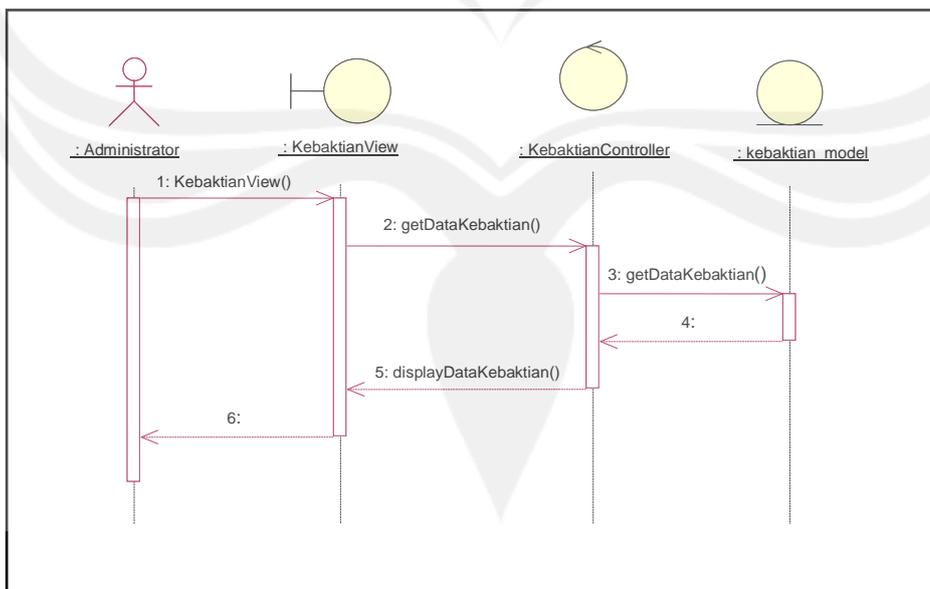
Gambar 2.4 Sequence Diagram : Mengelola Kebaktian – Edit Data Kebaktian

### 2.2.1.2.3 Hapus Data Kebaktian



Gambar 2.5 Sequence Diagram : Mengelola Kebaktian - Hapus Data Kebaktian

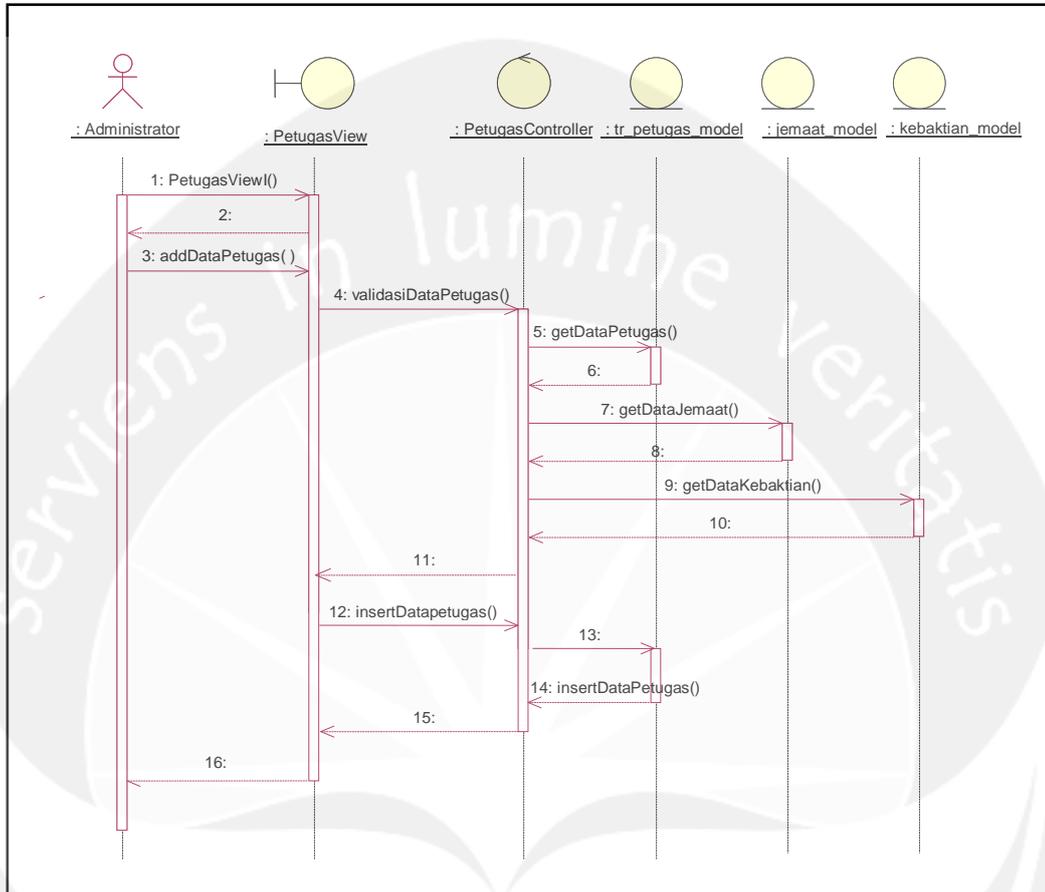
### 2.2.1.2.4 Display Data Kebaktian



Gambar 2.6 Sequence Diagram : Mengelola Kebaktian - Display Data Kebaktian

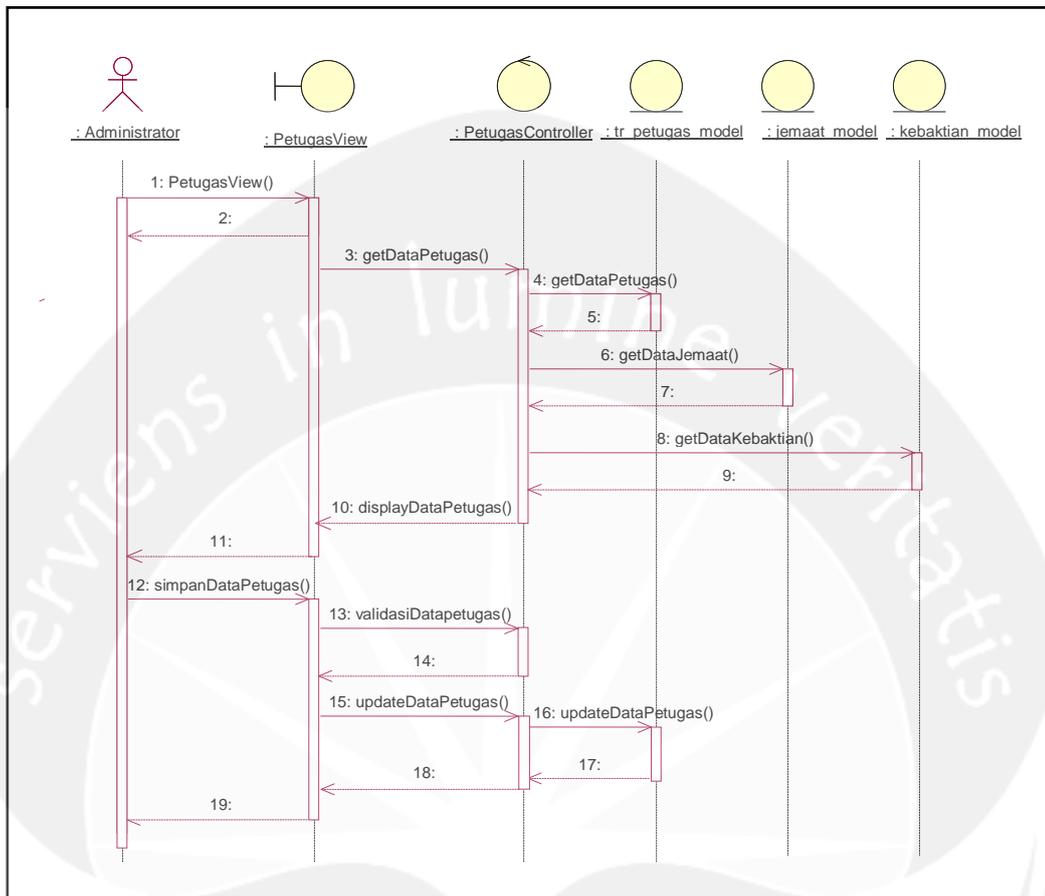
### 2.2.1.3 Mengelola Petugas Kebaktian

#### 2.2.1.3.1 Entry Data Petugas



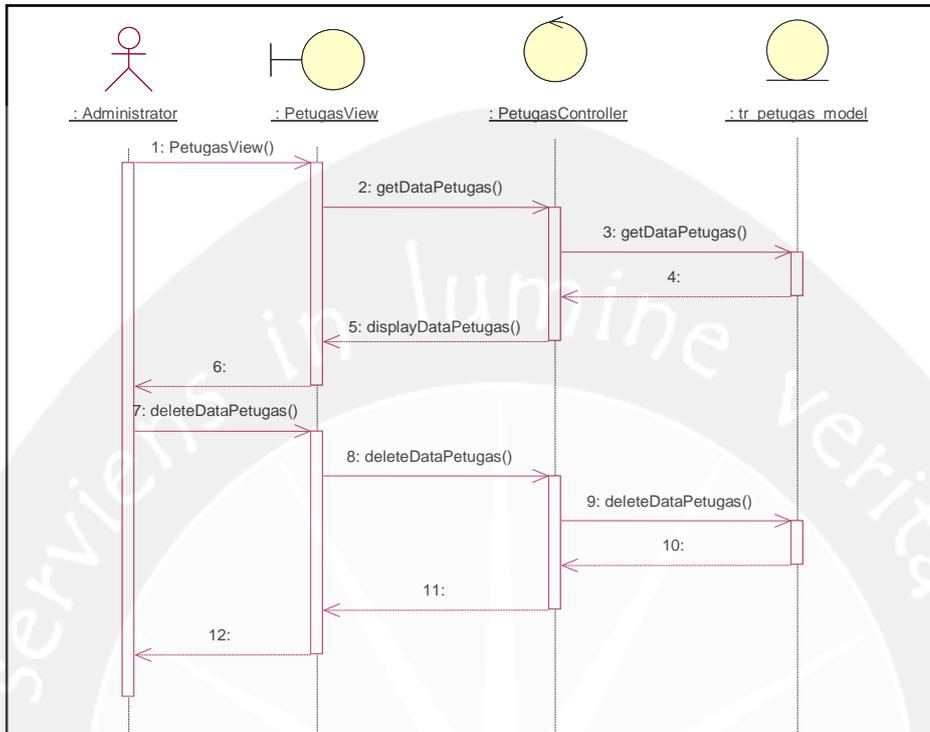
Gambar 2.7 Sequence Diagram: Mengelola Petugas Kebaktian - Entry Data Petugas

### 2.2.1.3.2 Edit Data Petugas



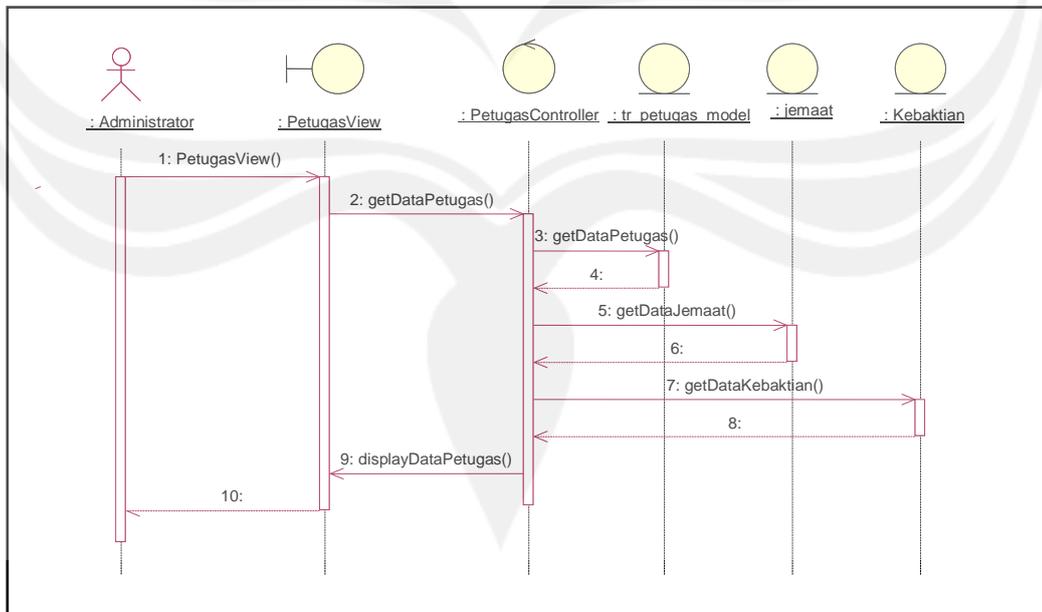
Gambar 2.8 Sequence Diagram: Mengelola Petugas Kebaktian – Edit Data Petugas

### 2.2.1.3.3 Delete Data Petugas



Gambar 2.9 Sequence Diagram: Mengelola Petugas Kebaktian – Delete Data Petugas

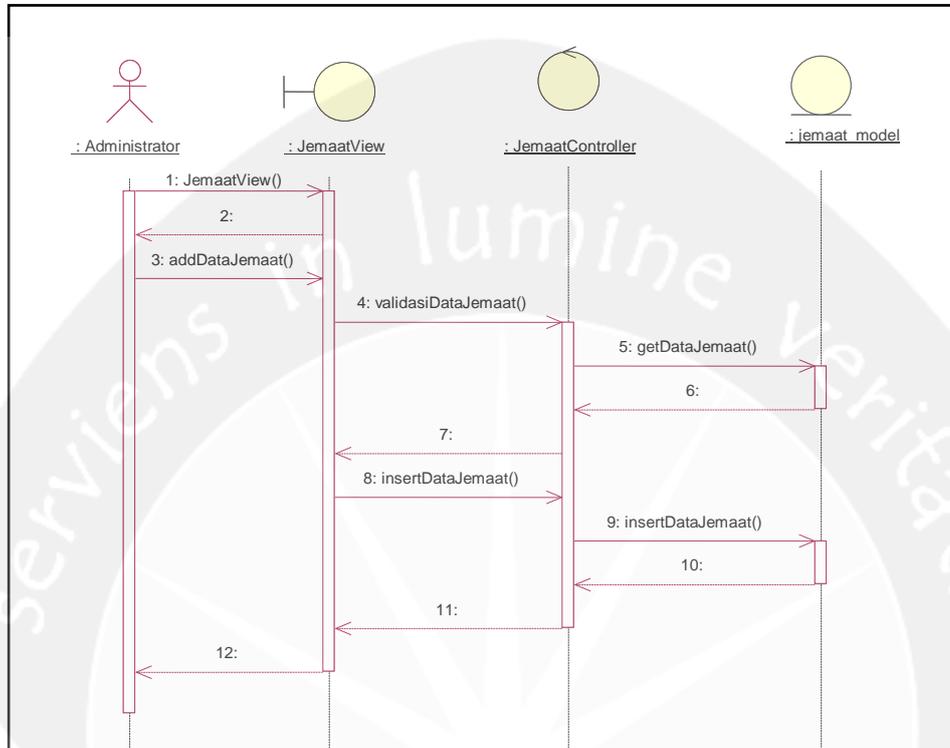
### 2.2.1.3.4 Display Data Petugas



Gambar 2.10 Sequence Diagram: Mengelola Petugas Kebaktian – Display Data Petugas

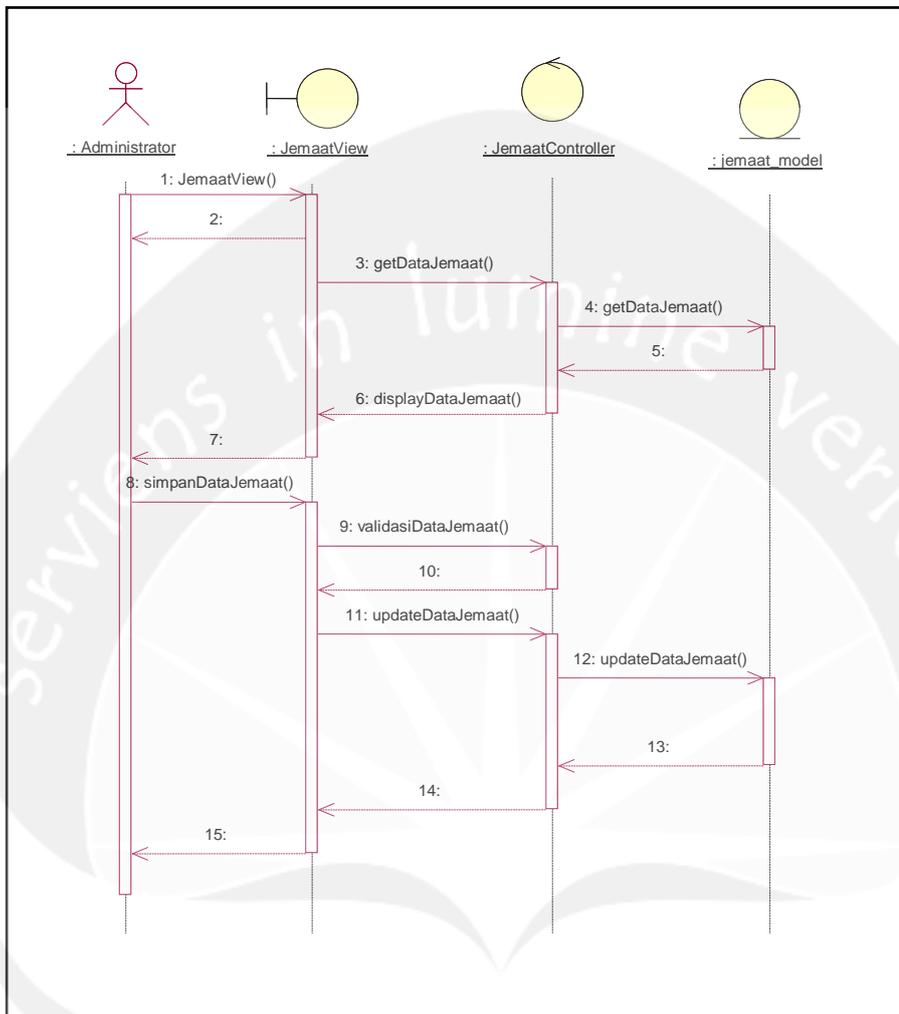
## 2.2.1.4 Mengelola Jemaat

### 2.2.1.4.1 Entry Data Jemaat



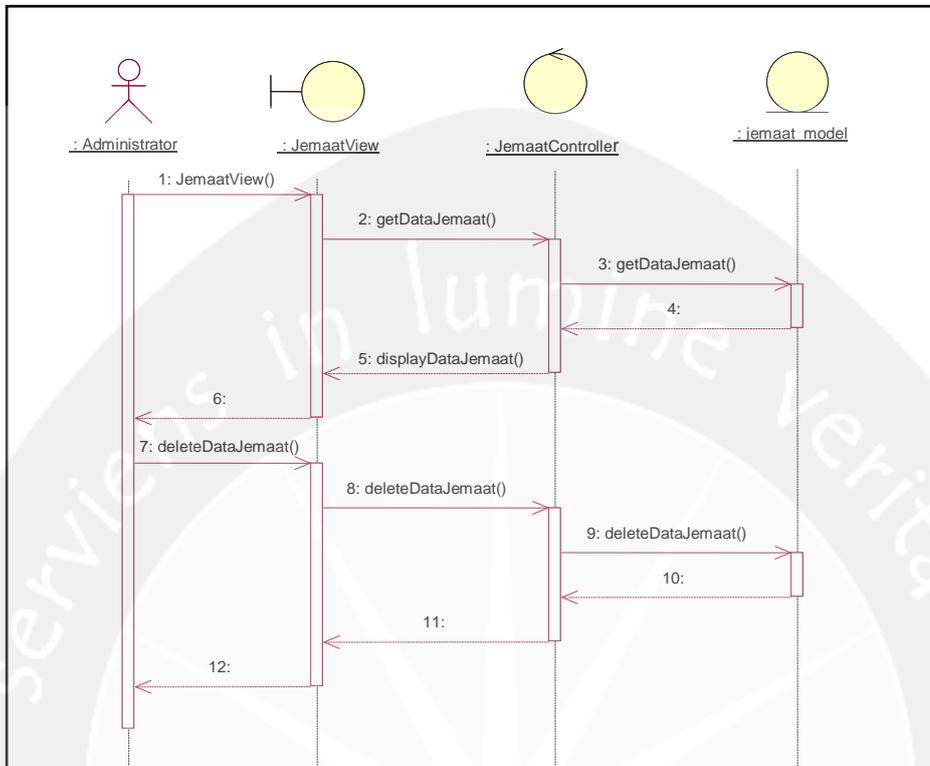
Gambar 2.15 Sequence Diagram: Mengelola Jemaat - Entry Data Jemaat

### 2.2.1.4.2 Edit Data Jemaat



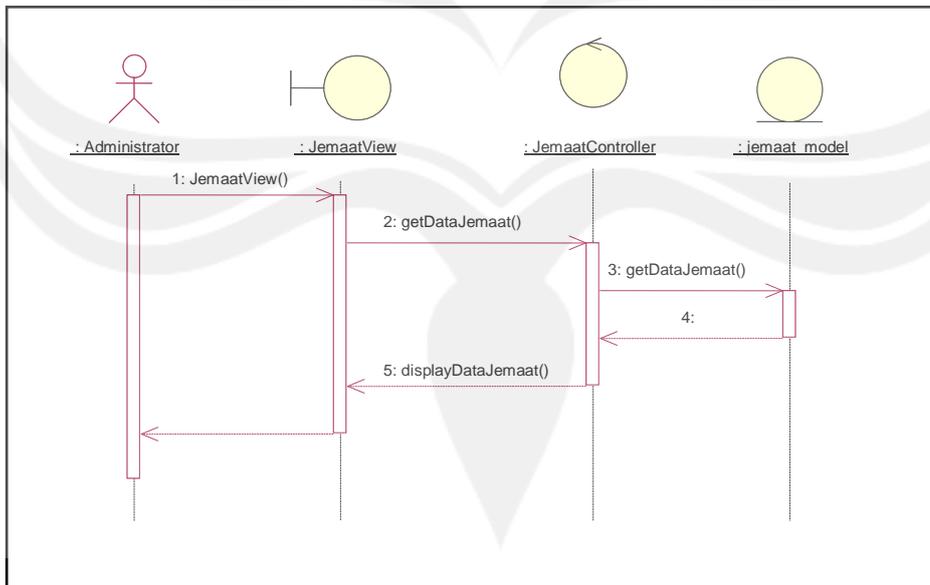
Gambar 2.16 Sequence Diagram: Mengelola Jemaat - Edit Data Jemaat

### 2.2.1.4.3 Hapus Data Jemaat



Gambar 2.17 Sequence Diagram: Mengelola Jemaat - Hapus Data Jemaat

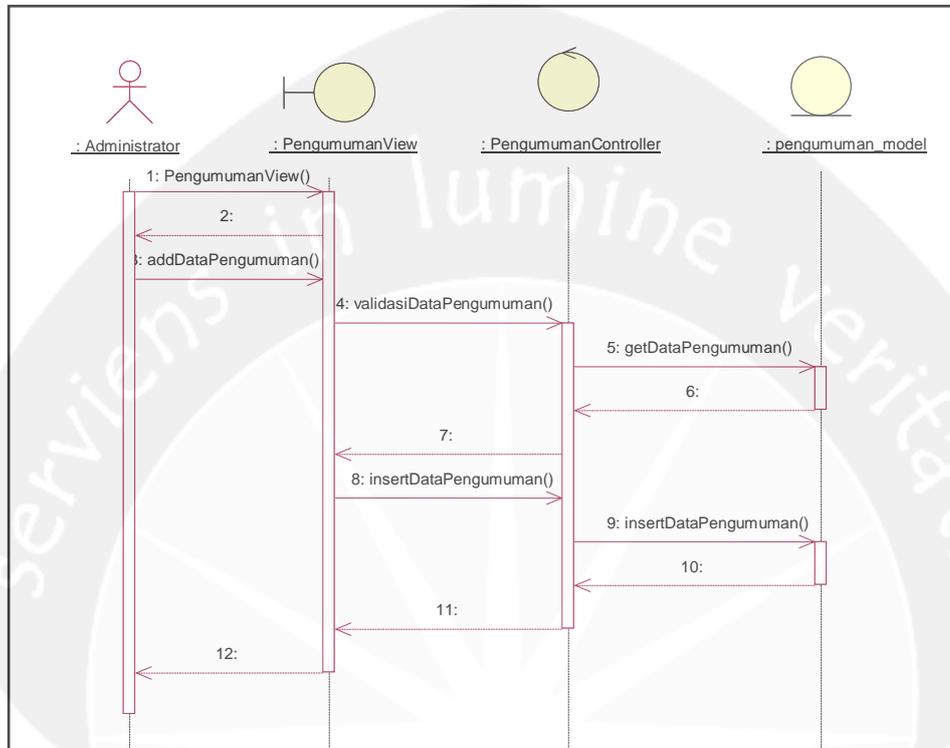
### 2.2.1.4.4 Display Data Jemaat



Gambar 2.18 Sequence Diagram: Mengelola Jemaat - Display Data Jemaat

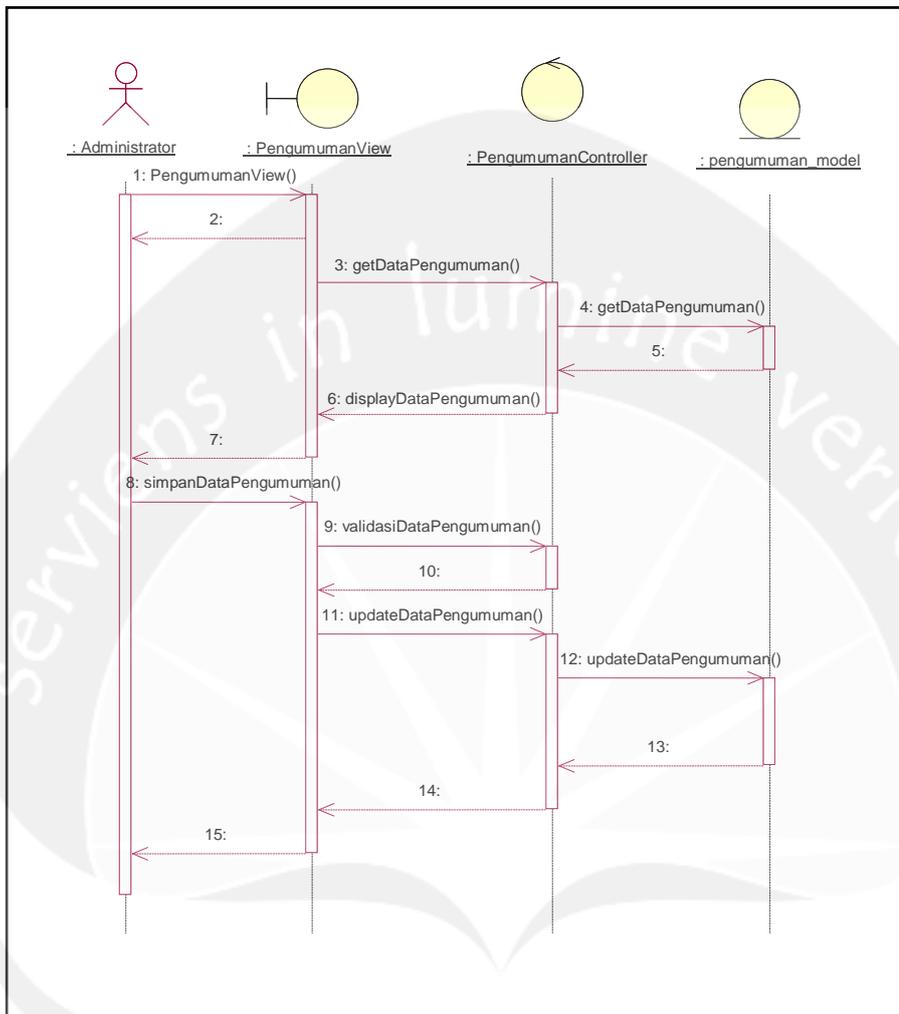
## 2.2.1.5 Mengelola Pengumuman

### 2.2.1.5.1 Entry Data Pengumuman



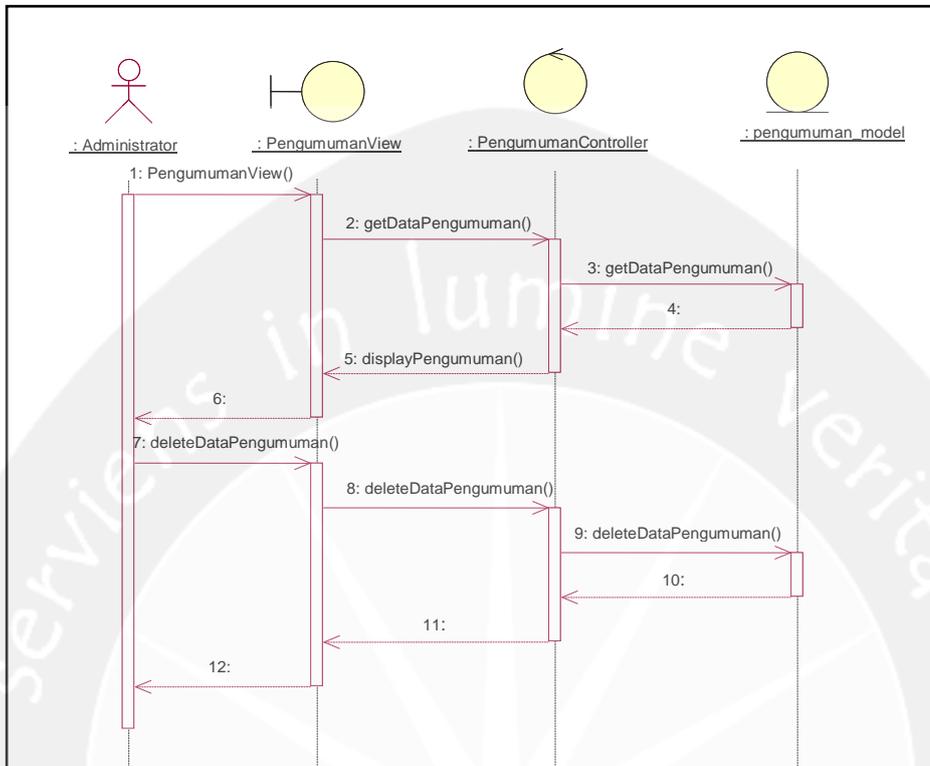
Gambar 2.19 Sequence Diagram: Mengelola Pengumuman - Entry Data Pengumuman

### 2.2.1.5.2 Edit Data Pengumuman



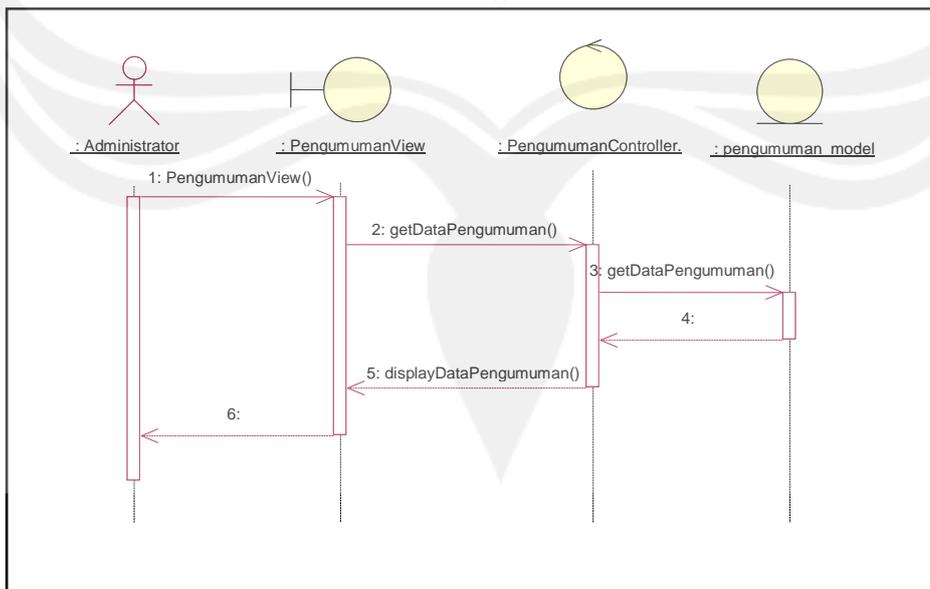
Gambar 2.20 Sequence Diagram: Mengelola Pengumuman - Edit Data Pengumuman

### 2.2.1.5.3 Delete Data Pengumuman



Gambar 2.21 Sequence Diagram: Mengelola Pengumuman - Delete Data Pengumuman

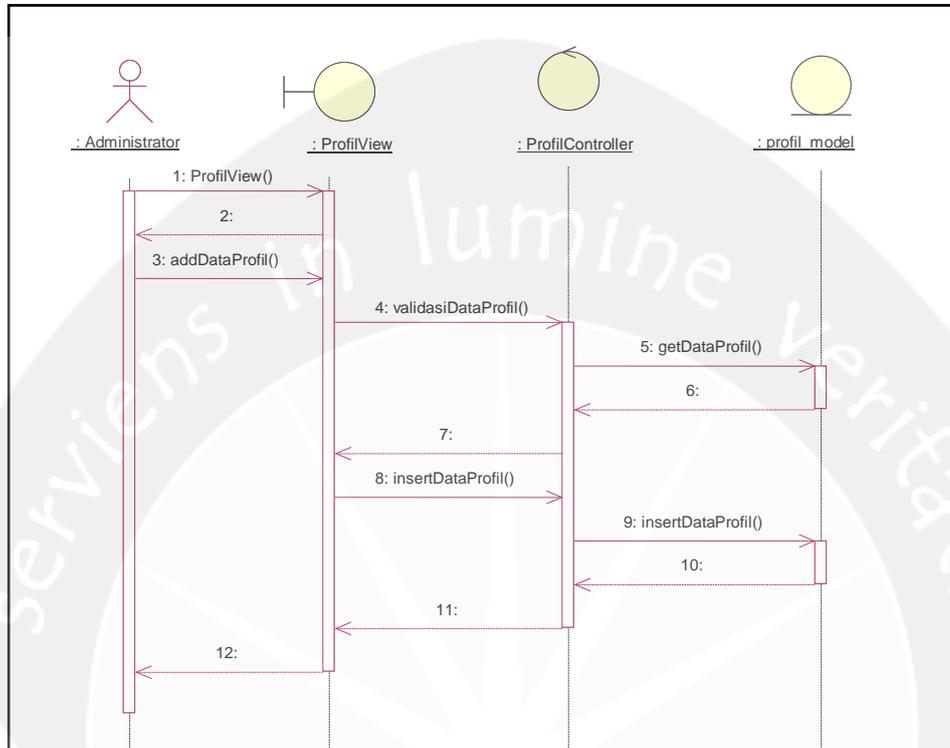
### 2.2.1.5.4 Display Data Pengumuman



Gambar 2.22 Sequence Diagram: Mengelola Pengumuman - Display Data Pengumuman

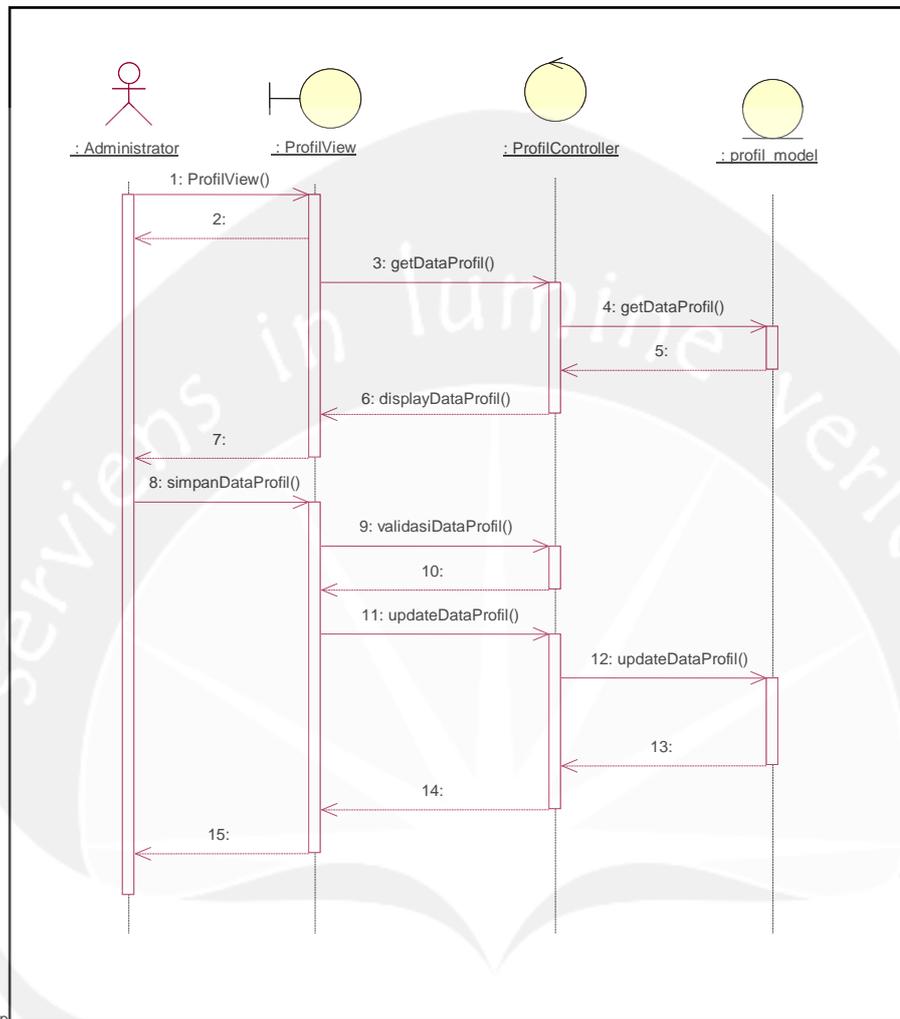
## 2.2.1.6 Mengelola Profil

### 2.2.1.6.1 Entry Data Profil



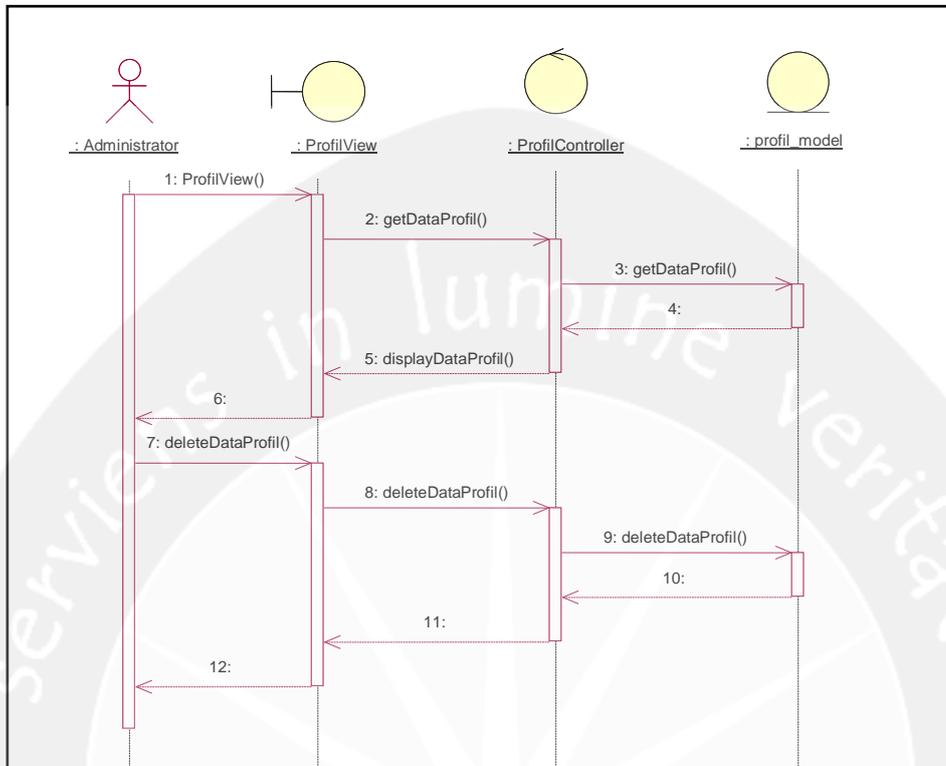
Gambar 2.23 Sequence Diagram: Mengelola Profil - Entry Data Profil

### 2.2.1.6.2 Edit Data Profil



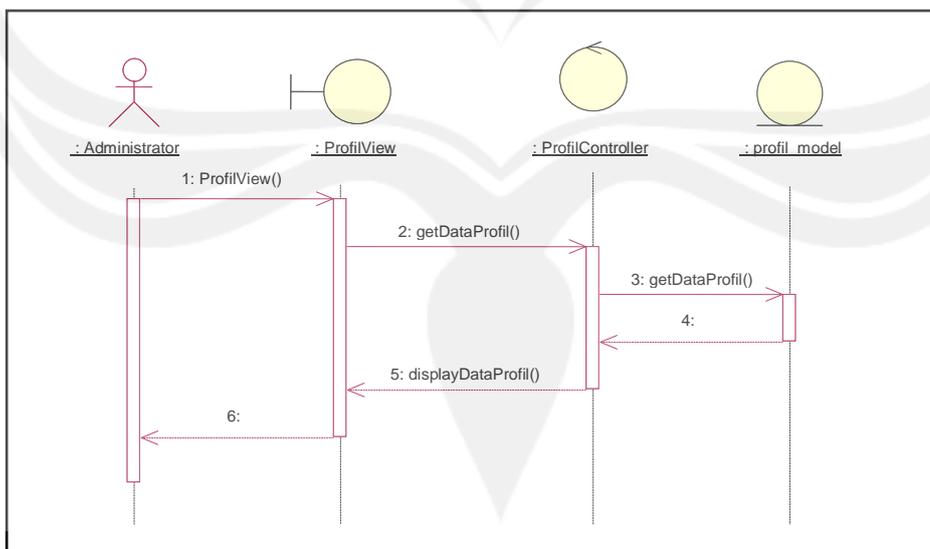
Gambar 2.24 Sequence Diagram: Mengelola Profil – Edit Data Profil

### 2.2.1.6.3 Delete Data Profil



Gambar 2.25 Sequence Diagram: Mengelola Profil - Delete Data Profil

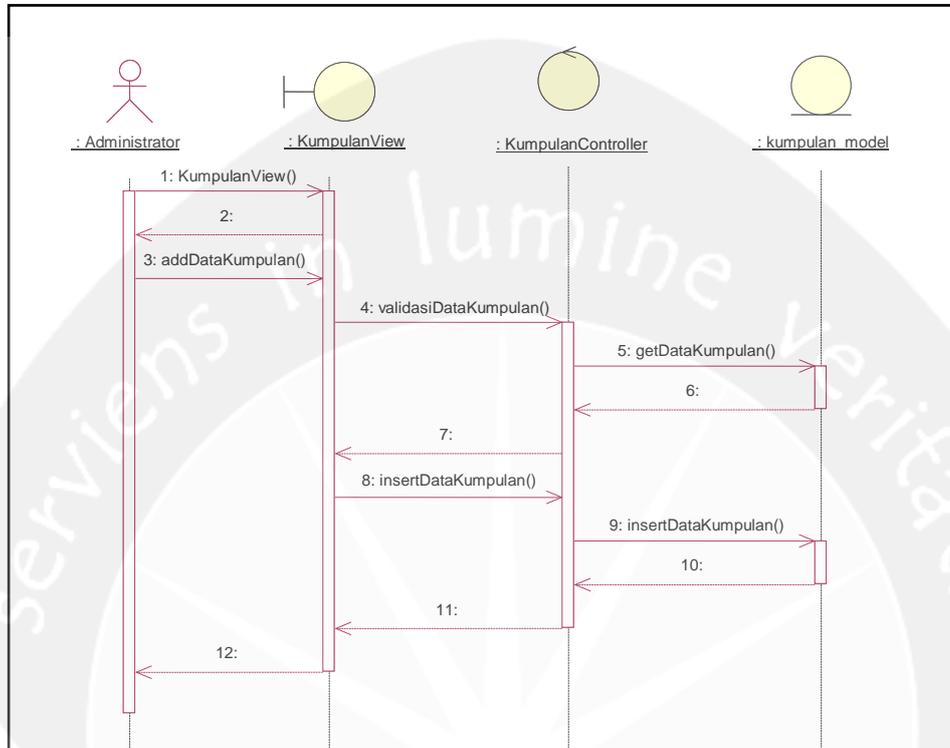
### 2.2.1.6.4 Display Data Profil



Gambar 2.26 Sequence Diagram: Mengelola Profil - Display Data Profil

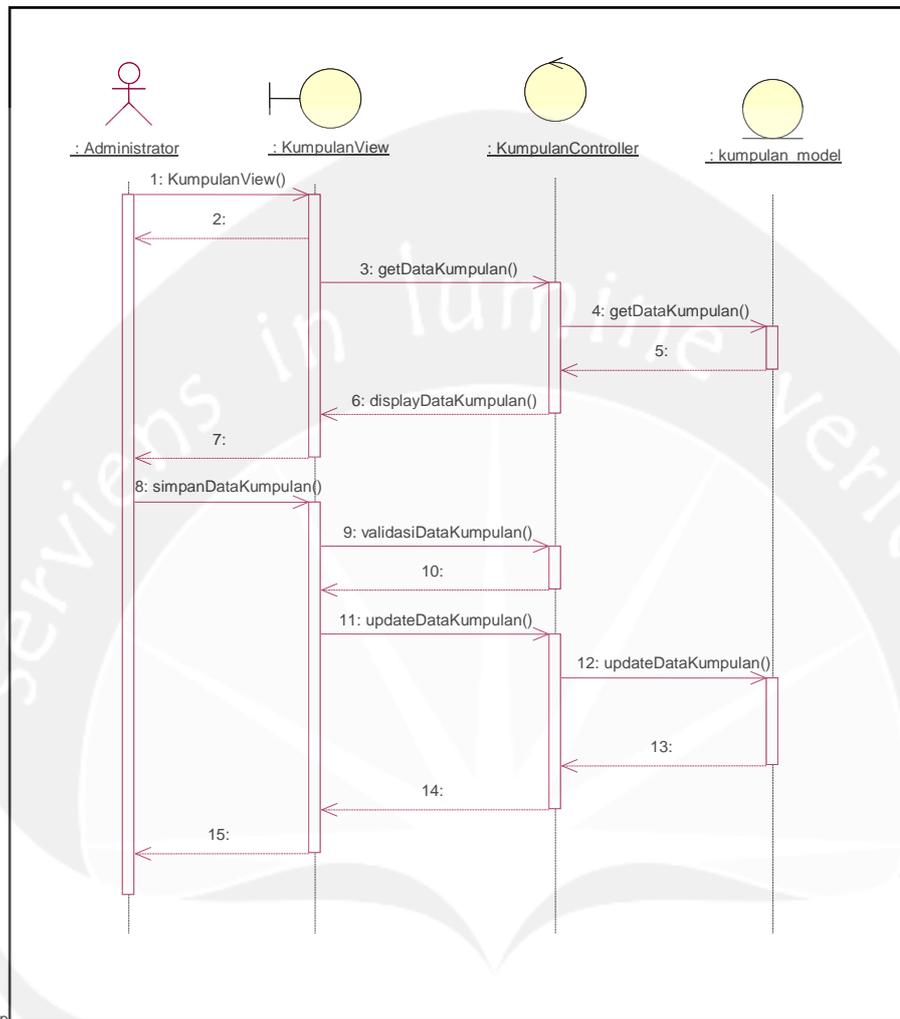
## 2.2.1.7 Mengelola Kumpulan

### 2.2.1.7.1 Entry Data Kumpulan



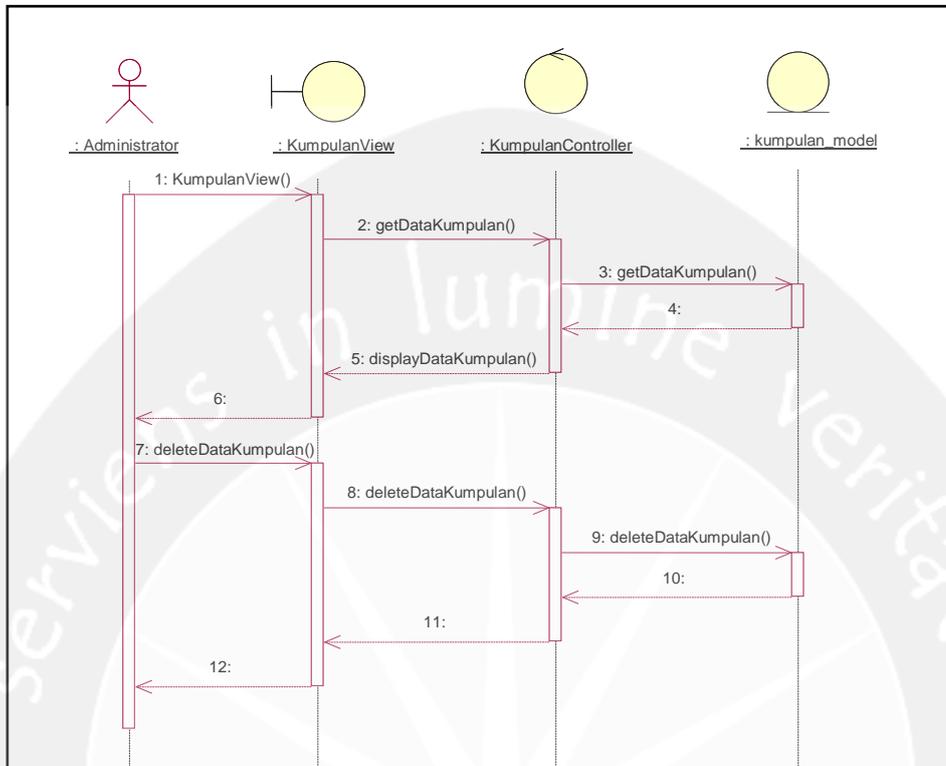
Gambar 2.27 Sequence Diagram: Mengelola Kumpulan - Entry Data Kumpulan

### 2.2.1.7.2 Edit Data Kumpulan



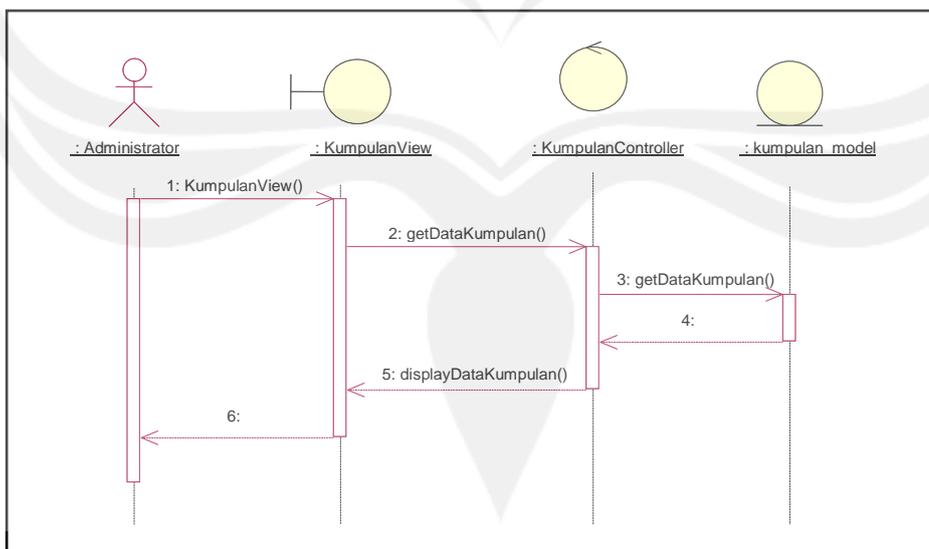
Gambar 2.28 Sequence Diagram: Mengelola Kumpulan – Edit Data Kumpulan

### 2.2.1.7.3 Delete Data Kumpulan



Gambar 2.29 Sequence Diagram: Mengelola Kumpulan - Delete Data Kumpulan

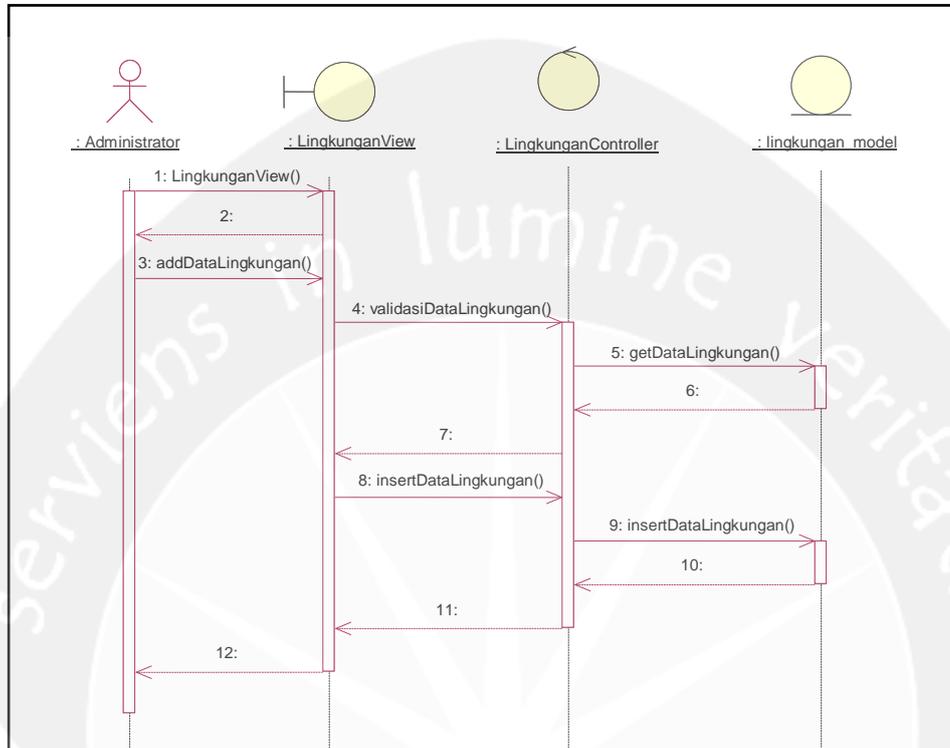
### 2.2.1.7.4 Display Data Kumpulan



Gambar 2.30 Sequence Diagram: Mengelola Kumpulan - Display Data Kumpulan

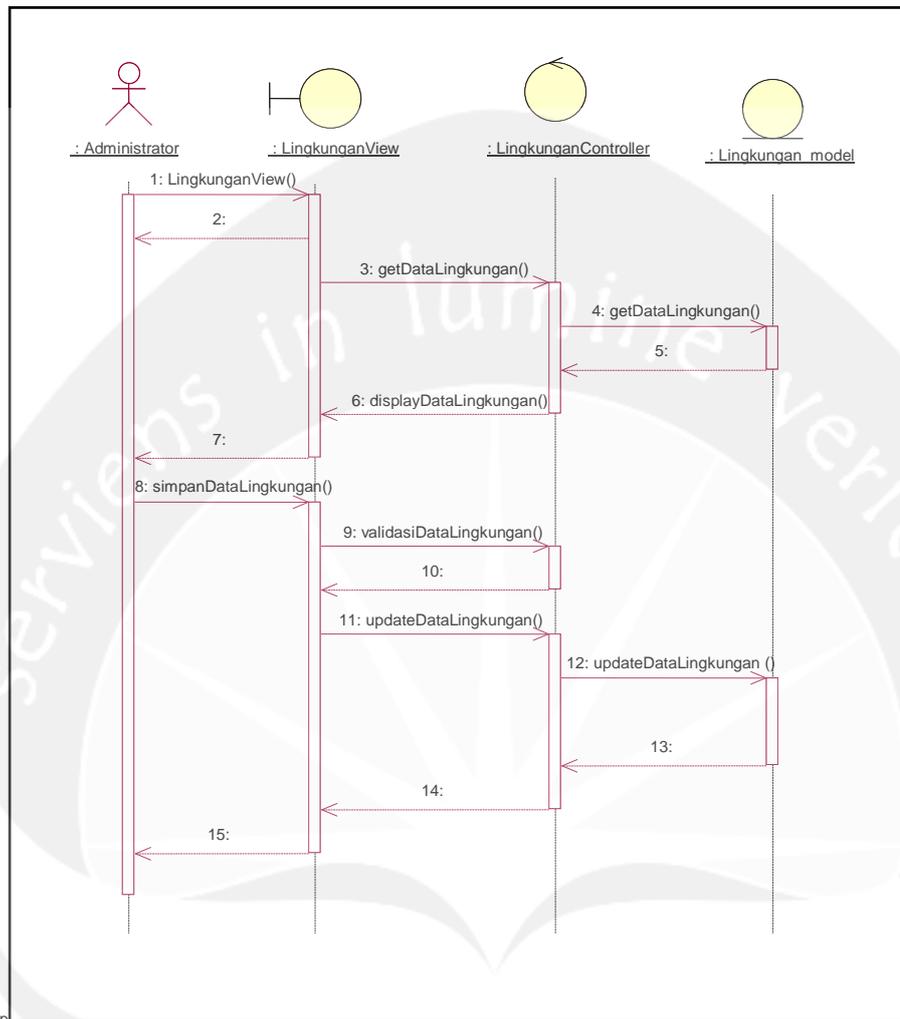
## 2.2.1.8 Mengelola Lingkungan

### 2.2.1.8.1 Entry Data Lingkungan



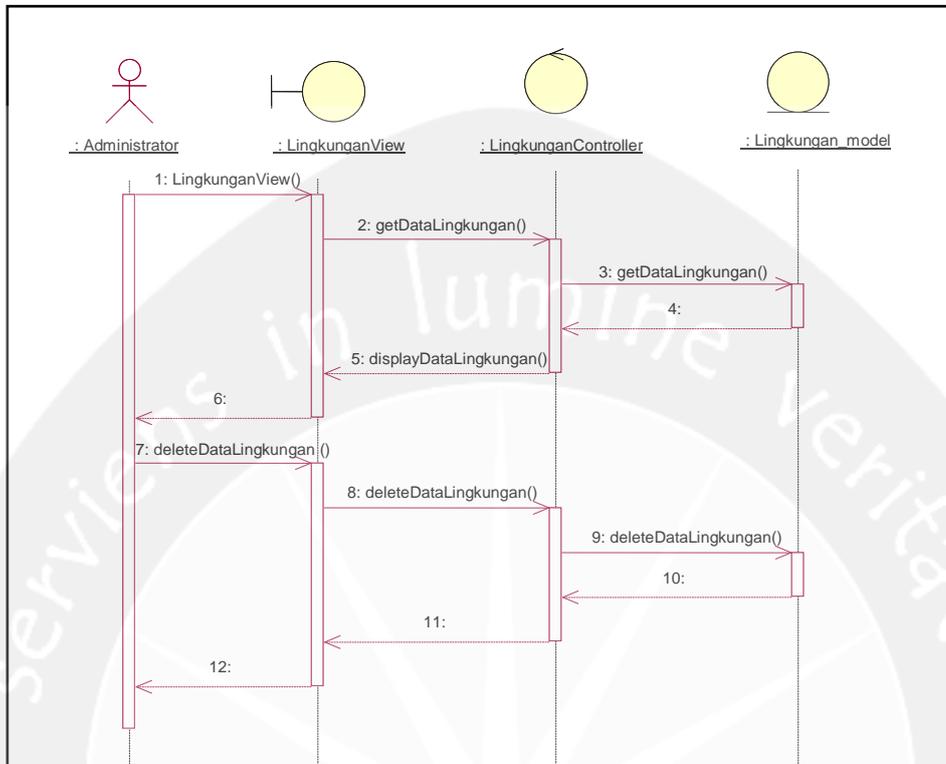
Gambar 2.31 Sequence Diagram: Mengelola Lingkungan - Entry Data Lingkungan

### 2.2.1.8.2 Edit Data Lingkungan



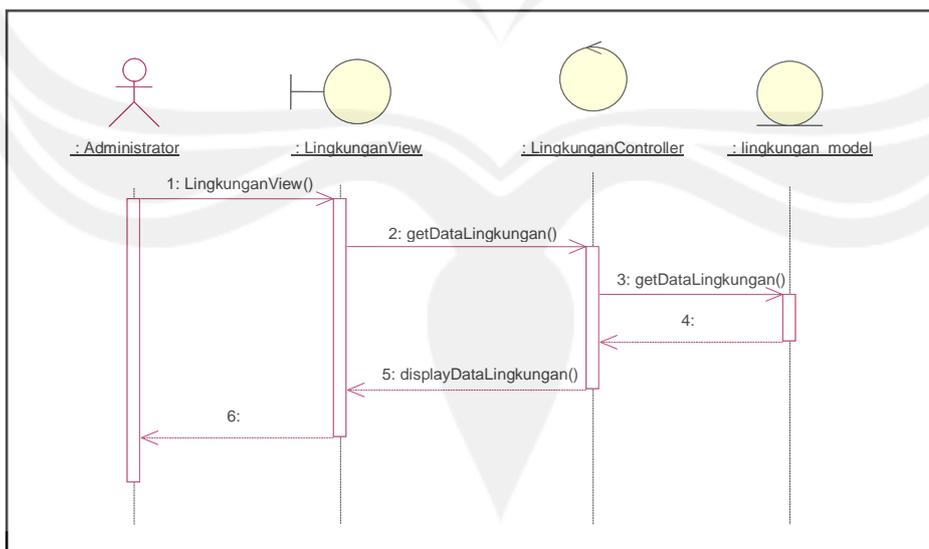
Gambar 2.32 Sequence Diagram: Mengelola Lingkungan - Edit Data Lingkungan

### 2.2.1.8.3 Delete Data Lingkungan



Gambar 2.33 Sequence Diagram: Mengelola Lingkungan - Delete Data Lingkungan

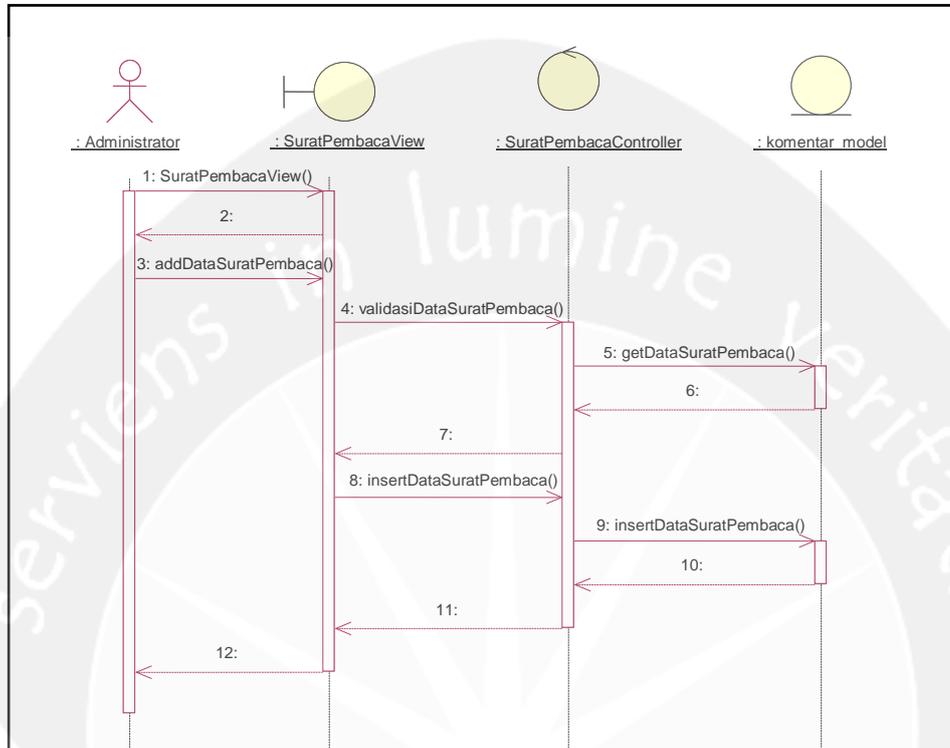
### 2.2.1.8.4 Display Data Lingkungan



Gambar 2.34 Sequence Diagram: Mengelola Lingkungan - Display Data Lingkungan

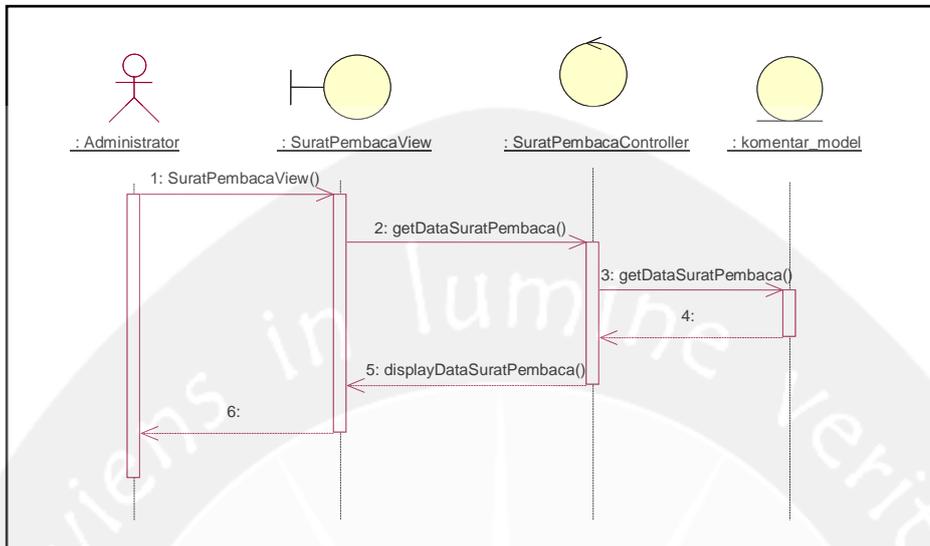
## 2.2.1.9 Mengelola Surat Pembaca

### 2.2.1.9.1 Entry Data Surat Pembaca



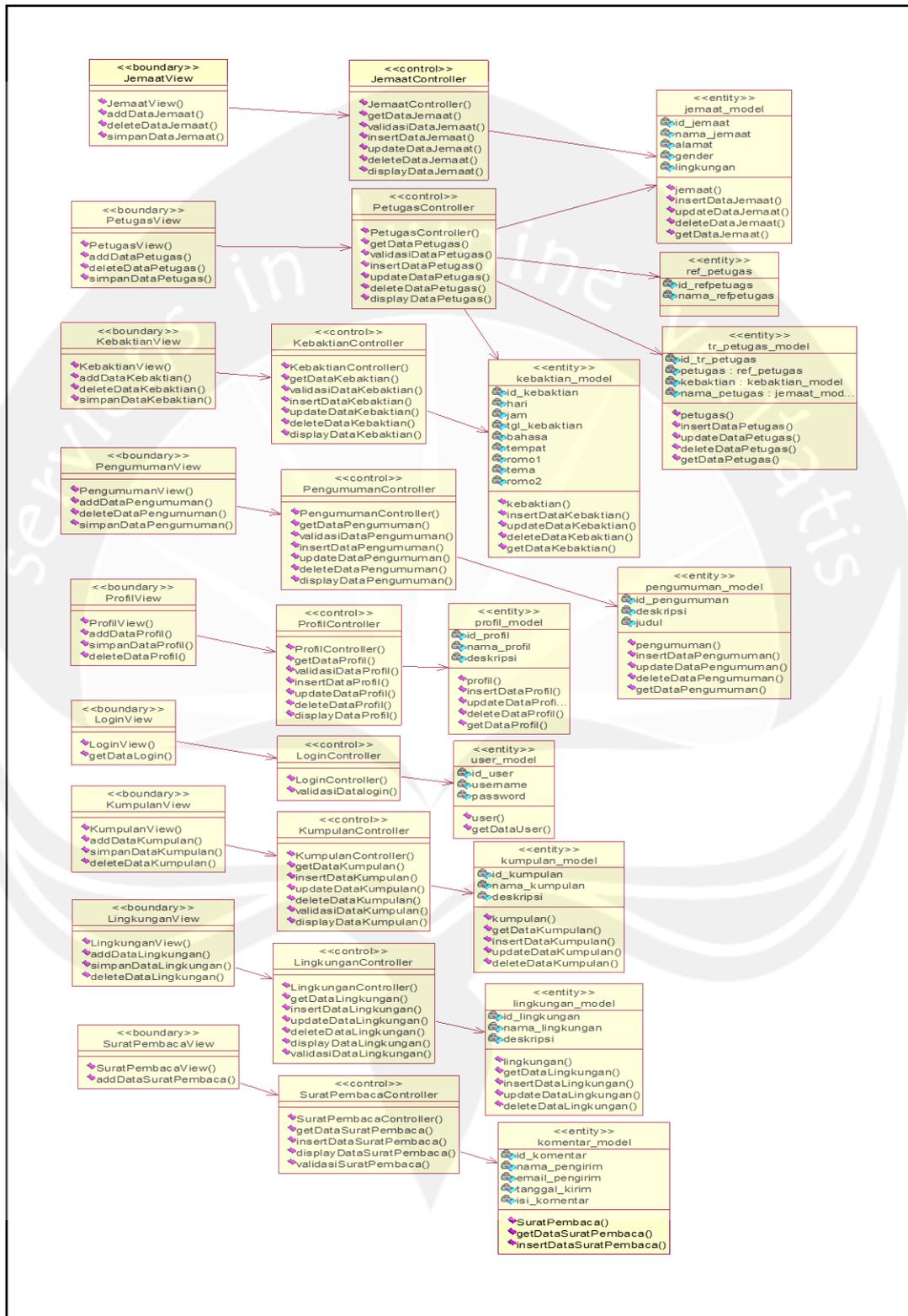
Gambar 2.35 Sequence Diagram: Mengelola Surat pembaca - Entry Data Surat Pembaca

### 2.2.1.9.2 Display Data Surat Pembaca



Gambar 2.36 Sequence Diagram: Mengelola Surat Pembaca – Display Data Surat Pembaca

## 2.2.2 Class Diagrams



Gambar 2.35 Class Diagram

### 2.2.3 Spesifikasi Deskripsi Class Diagram

#### 2.2.3.1 Spesifikasi Desain Kelas LoginView

<b>LoginView</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+loginView() 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.	

#### 2.2.3.2 Spesifikasi Desain Kelas JemaatView

<b>JemaatView</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+JemaatView() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+addDataJemaat() Operasi ini digunakan untuk menjalankan form action entry data jemaat pada controller.	
+deleteDataJemaat() Operasi ini digunakan untuk menjalankan form action delete data jemaat pada controller.	
+simpanDataJemaat() Operasi ini digunakan untuk menjalankan form action edit data jemaat pada controller.	

#### 2.2.3.3 Spesifikasi Desain Kelas PetugasView

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

+addDataPetugas()

Operasi ini digunakan untuk menjalankan form action entry Petugas pada controller.

+simpanDataPetugas()

Operasi ini digunakan untuk menjalankan form action edit petugas pada controller.

+deleteDataPetugas()

Operasi ini digunakan untuk menjalankan form action delete Petugas pada controller.

#### 2.2.3.4 Spesifikasi Desain Kelas KebaktianView

<b>KebaktianView</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+KebaktianView() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+addDataKebaktian() Operasi ini digunakan untuk menjalankan form action entry kebaktian pada controller.	
+simpanDataKebaktian() Operasi ini digunakan untuk menjalankan form action edit kebaktian pada controller.	
+deleteDataKebaktian() Operasi ini digunakan untuk menjalankan form action delete kebaktian pada controller.	

#### 2.2.3.5 Spesifikasi Desain Kelas PengumumanView

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

+addDataPengumuman()

Operasi ini digunakan untuk menjalankan form action entry pengumuman pada controller.

+simpanDataPengumuman()

Operasi ini digunakan untuk menjalankan form action edit pengumuman pada controller.

+deleteDataPengumuman()

Operasi ini digunakan untuk menjalankan form action delete pengumuman pada controller.

### 2.2.3.6 Spesifikasi Desain Kelas ProfilView

<b>ProfilView</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+ProfilView() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+addDataProfil() Operasi ini digunakan untuk menjalankan form action entry profil pada controller.	
+simpanDataProfil () Operasi ini digunakan untuk menjalankan form action edit profil pada controller.	
+deleteDataProfil() Operasi ini digunakan untuk menjalankan form action delete profil pada controller.	

### 2.2.3.7 Spesifikasi Desain Kelas KumpulanView

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

+addDataKumpulan()

Operasi ini digunakan untuk menjalankan form action entry kumpulan pada controller.

+simpanDataKumpulan()

Operasi ini digunakan untuk menjalankan form action edit kumpulan pada controller.

+deleteDataKumpulan()

Operasi ini digunakan untuk menjalankan form action delete kumpulan pada controller.

### 2.2.3.8 Spesifikasi Desain Kelas LingkunganView

<b>LingkunganView</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+LingkunganView() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+addDataLingkungan() Operasi ini digunakan untuk menjalankan form action entry lingkungan pada controller.	
+simpanDataLingkungan() Operasi ini digunakan untuk menjalankan form action edit lingkungan pada controller.	
+deleteDataLingkungan() Operasi ini digunakan untuk menjalankan form action delete lingkungan pada controller.	

### 2.2.3.9 Spesifikasi Desain Kelas SuratPembacaView

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

```
+addDataSuratPembaca()
```

Operasi ini digunakan untuk menyimpan data surat pembaca.

#### 2.2.3.10 Spesifikasi Desain Kelas Login Controller

<b>LoginController</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+LoginController() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +validasiUser() Operasi ini digunakan untuk mengecek data user apakah valid atau tidak.</pre>	

#### 2.2.3.11 Spesifikasi Desain Kelas Jemaat Controller

<b>JemaatController</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<pre>+JemaatController() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataJemaat() Operasi ini digunakan untuk menginputkan data jemaat. +updateDataJemaat() Operasi ini digunakan untuk mengubah data jemaat. +deleteDataJemaat() Operasi ini digunakan untuk menghapus data jemaat. +displayDataJemaat() Operasi ini digunakan untuk menampilkan data jemaat. +getDataJemaat() Operasi ini digunakan untuk mengambil data jemaat. +validasiDataJemaat() Operasi ini digunakan untuk mengecek data jemaat apakah valid atau tidak.</pre>	

### 2.2.3.12 Spesifikasi Desain Kelas PetugasController

PetugasController	<<Control>>
<pre> +PetugasController() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataPetugas() Operasi ini digunakan untuk menginputkan data petugas. +updateDataPetugas() Operasi ini digunakan untuk mengubah data petugas. +deleteDataPetugas() Operasi ini digunakan untuk menghapus data petugas. +displayDataPetugas() Operasi ini digunakan untuk menampilkan data petugas. +getDataPetugas() Operasi ini digunakan untuk mengambil data petugas. +validasiDataPetugas() Operasi ini digunakan untuk mengecek data petugas apakah valid atau tidak. </pre>	

### 2.2.3.13 Spesifikasi Desain Kelas KebaktianController

KebaktianController	<<Control>>
<pre> +KebaktianController() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataKebaktian() Operasi ini digunakan untuk menginputkan data kebaktian. +updateDataKebaktian() Operasi ini digunakan untuk mengubah data kebaktian. +deleteDataKebaktian() </pre>	

Operasi ini digunakan untuk menghapus data kebaktian.  
+displayDataKebaktian()  
Operasi ini digunakan untuk menampilkan data kebaktian.  
+getDataKebaktian()  
Operasi ini digunakan untuk mengambil data kebaktian.  
+validasiDataKebaktian()  
Operasi ini digunakan untuk mengecek data kebaktian apakah valid atau tidak.

**2.2.3.14 Spesifikasi Desain Kelas PengumumanController**

<b>PengumumanController</b>	<b>&lt;&lt;Control&gt;&gt;</b>
+PengumumanController() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataPengumuman() Operasi ini digunakan untuk menginputkan data pengumuman. +updateDataPengumuman() Operasi ini digunakan untuk mengubah data pengumuman. +deleteDataPengumuman() Operasi ini digunakan untuk menghapus data pengumuman. +displayDataPengumuman() Operasi ini digunakan untuk menampilkan data pengumuman. +getDataPengumuman() Operasi ini digunakan untuk mengambil data pengumuman. +validasiDataPengumuman() Operasi ini digunakan untuk mengecek data pengumuman apakah valid atau tidak.	

**2.2.3.15 Spesifikasi Desain Kelas ProfilController**

<b>ProfilController</b>	<b>&lt;&lt;Control&gt;&gt;</b>
+ProfilController()	

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

+insertDataProfil()  
Operasi ini digunakan untuk menginputkan data profil.

+updateDataProfil()  
Operasi ini digunakan untuk mengubah data profil.

+deleteDataProfil()  
Operasi ini digunakan untuk menghapus data profil.

+displayDataProfil()  
Operasi ini digunakan untuk menampilkan data profil.

+getDataProfil()  
Operasi ini digunakan untuk mengambil data profil.

+validasiDataProfil()  
Operasi ini digunakan untuk mengecek data profil apakah valid atau tidak.

### 2.2.3.16 Spesifikasi Desain Kelas KumpulanController

<b>KumpulanController</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<p>+KumpulanController() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+insertDataKumpulan() Operasi ini digunakan untuk menginputkan data kumpulan.</p> <p>+updateDataKumpulan() Operasi ini digunakan untuk mengubah data kumpulan.</p> <p>+deleteDataKumpulan() Operasi ini digunakan untuk menghapus data kumpulan.</p> <p>+displayDataKumpulan() Operasi ini digunakan untuk menampilkan data kumpulan.</p> <p>+getDataKumpulan() Operasi ini digunakan untuk mengambil data kumpulan.</p>	

+validasiDataKumpulan()

Operasi ini digunakan untuk mengecek data kumpulan apakah valid atau tidak.

### 2.2.3.17 Spesifikasi Desain Kelas LingkunganController

LingkunganController	<<Control>>
<pre>+LingkunganController() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataLingkungan() Operasi ini digunakan untuk menginputkan data lingkungan. +updateDataLingkungan() Operasi ini digunakan untuk mengubah data lingkungan. +deleteDataLingkungan() Operasi ini digunakan untuk menghapus data lingkungan. +displayDataLingkungan() Operasi ini digunakan untuk menampilkan data lingkungan. +getDataLingkungan() Operasi ini digunakan untuk mengambil data lingkungan. +validasiDataLingkungan() Operasi ini digunakan untuk mengecek data lingkungan apakah valid atau tidak.</pre>	

### 2.2.3.18 Spesifikasi Desain Kelas SuratPembacaController

SuratPembacaController	<<Control>>
<pre>+SuratPembacaController() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataSuratPembaca() Operasi ini digunakan untuk menginputkan data surat pembaca.</pre>	

```
+displayDataSuratPembaca()
Operasi ini digunakan untuk menampilkan data surat pembaca.
+getDataSuratPembaca()
Operasi ini digunakan untuk mengambil data surat pembaca.
+validasiDataSuratPembaca()
Operasi ini digunakan untuk mengecek data surat pembaca
apakah valid atau tidak.
```

### 2.2.3.19 Spesifikasi Desain Kelas User Model

<b>UserModel</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<pre>-id_user: Integer Atribut ini digunakan untuk menyimpan id user. -username: String Atribut ini digunakan untuk menyimpan data nama user. -password: String Atribut ini digunakan untuk menyimpan data password.</pre>	
<pre>+user() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDataUser() Atribut ini digunakan untuk mengambil data user.</pre>	

### 2.2.3.20 Spesifikasi Desain Kelas Jemaat Model

<b>JemaatModel</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<pre>-id_jemaat : Integer Atribut ini digunakan untuk menyimpan data id jemaat. - nama_jemaat : String -alamat: String Atribut ini digunakan untuk menyimpan data tanggal krisma. - gender : String</pre>	

Atribut ini digunakan untuk menyimpan data gender.

- lingkungan: String

Atribut ini digunakan untuk menyimpan data lingkungan.

+jemaat()

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

+insertDataJemaat()

Operasi ini digunakan untuk memasukkan data jemaat ke dalam database.

+deleteDataJemaat()

Operasi ini digunakan oleh user untuk menghapus data jemaat yang ada didalam database.

+updateDataJemaat()

Operasi ini digunakan untuk mengupdate data jemaat yang ada di dalam database.

+getDataJemaat()

Operasi ini digunakan untuk mengambil data jemaat.

#### 2.2.3.21 Spesifikasi Desain Kelas tr\_petugas\_model

<b>tr_petugas_model</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<p>-id_tr_petugas: Integer Atribut ini digunakan untuk menyimpan data id_tr_petugas</p> <p>-id_kebaktian: Integer Atribut ini digunakan untuk menyimpan data id_kebaktian</p> <p>-id_jemaat: Integer Atribut ini digunakan untuk menyimpan data id_jemaat</p> <p>-id_refpetugas: Integer Atribut ini digunakan untuk menyimpan data id_refpertugas</p>	
<p>+petugas() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+insertDataPetugas() </p>	

Operasi ini digunakan untuk memasukkan data petugas ke dalam database.

+deleteDataPetugas()  
Operasi ini digunakan oleh user untuk menghapus data petugas yang ada didalam database.

+updateDataPetugas()  
Operasi ini digunakan untuk mengupdate data petugas yang ada di dalam database.

+getDataPetugas()  
Operasi ini digunakan untuk mengambil data petugas.

### 2.2.3.22 Spesifikasi Desain Kelas kebaktian\_model

Kebaktian_model	<<entity>>
<p>-id_kebaktian: Integer Atribut ini digunakan untuk menyimpan data id kebaktian.</p> <p>-hari: String Atribut ini digunakan untuk menyimpan data hari.</p> <p>-jam: Integer Atribut ini digunakan untuk menyimpan data jam.</p> <p>-bahasa: String Atribut ini digunakan untuk menyimpan data bahasa.</p> <p>-tgl_kebaktian: Datetime Atribut ini digunakan untuk menyimpan data tanggal kebaktian</p> <p>-tempat: String Atribut ini digunakan untuk menyimpan data tempat kebaktian</p> <p>-romol: String Atribut ini digunakan untuk menyimpan data nama romo 1</p> <p>-romo2: String Atribut ini digunakan untuk menyimpan data nama romo 2</p> <p>-tema: String Atribut ini digunakan untuk menyimpan data tema</p>	
<p>+kebaktian()</p>	

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

+insertDataKebaktian()  
Operasi ini digunakan untuk memasukkan data kebaktian ke dalam database.

+deleteDataKebaktian()  
Operasi ini digunakan oleh user untuk menghapus data kebaktian yang ada didalam database.

+updateDataKebaktian()  
Operasi ini digunakan untuk mengupdate data kebaktian yang ada di dalam database.

+getDataKebaktian()  
Operasi ini digunakan untuk mengambil data kebaktian.

### 2.2.3.23 Spesifikasi Desain Kelas pengumuman\_model

<b>Pengumuman_model</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<p>-id_pengumuman: Integer Atribut ini digunakan untuk menyimpan data id pengumuman</p> <p>-judul: String Atribut ini digunakan untuk menyimpan data judul pengumuman</p> <p>-deskripsi: Text Atribut ini digunakan untuk menyimpan data deskripsi.</p>	
<p>+pengumuman() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+insertDataPenguman() Operasi ini digunakan untuk memasukkan data pengumuman ke dalam database.</p> <p>+deleteDataPengumuman() Operasi ini digunakan oleh user untuk menghapus data pengumuman yang ada didalam database.</p> <p>+updateDataPengumuman()</p>	

Operasi ini digunakan untuk mengupdate data pengumuman yang ada di dalam database.

```
+getDataPengumuman()
```

Operasi ini digunakan untuk mengambil data pengumuman.

#### 2.2.3.24 Spesifikasi Desain Kelas profil\_model

<b>profil_model</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<p>-id_profil: Integer Atribut ini digunakan untuk menyimpan data id pengumuman</p> <p>-nama_profil: String Atribut ini digunakan untuk menyimpan data nama profil</p> <p>-deskripsi: Text Atribut ini digunakan untuk menyimpan data deskripsi.</p>	
<pre>+profil()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <pre>+insertDataProfil()</pre> <p>Operasi ini digunakan untuk memasukkan data profil ke dalam database.</p> <pre>+deleteDataProfil()</pre> <p>Operasi ini digunakan oleh user untuk menghapus data profil yang ada didalam database.</p> <pre>+updateDataProfil()</pre> <p>Operasi ini digunakan untuk mengupdate data profil yang ada di dalam database.</p> <pre>+getDataProfil()</pre> <p>Operasi ini digunakan untuk mengambil data profil.</p>	

#### 2.2.3.25 Spesifikasi Desain Kelas ref\_petugas

<b>ref_petugas</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<p>-id_refpetugas: Integer</p>	

Atribut ini digunakan untuk menyimpan data id\_refpetugas.  
 -nama\_refpetugas: String  
 Atribut ini digunakan untuk menyimpan data nama\_refpetugas.

### 2.2.3.26 Spesifikasi Desain Kelas kumpulan\_model

<b>kumpulan_model</b>	<b>&lt;&lt;entity&gt;&gt;</b>
-id_kumpulan: Integer Atribut ini digunakan untuk menyimpan data id kumpulan -nama_kumpulan: String Atribut ini digunakan untuk menyimpan data nama kumpulan -deskripsi: Text Atribut ini digunakan untuk menyimpan data deskripsi.	
+kumpulan() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataKumpulan() Operasi ini digunakan untuk memasukkan data kumpulan ke dalam database. +deleteDataKumpulan() Operasi ini digunakan oleh user untuk menghapus data kumpulan yang ada didalam database. +updateDataKumpulan() Operasi ini digunakan untuk mengupdate data kumpulan yang ada di dalam database. +getDataKumpulan() Operasi ini digunakan untuk mengambil data kumpulan.	

### 2.2.3.27 Spesifikasi Desain Kelas lingkungan\_model

<b>lingkungan_model</b>	<b>&lt;&lt;entity&gt;&gt;</b>
-id_lingkungan: Integer Atribut ini digunakan untuk menyimpan data id lingkungan	

-nama\_lingkungan: String

Atribut ini digunakan untuk menyimpan data nama lingkungan

-deskripsi: Text

Atribut ini digunakan untuk menyimpan data deskripsi.

+lingkungan()

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

+insertDataLingkungan()

Operasi ini digunakan untuk memasukkan data lingkungan ke dalam database.

+deleteDataLingkungan()

Operasi ini digunakan oleh user untuk menghapus data lingkungan yang ada didalam database.

+updateDataLingkungan()

Operasi ini digunakan untuk mengupdate data lingkungan yang ada di dalam database.

+getDataLingkungan()

Operasi ini digunakan untuk mengambil data lingkungan.

### 2.2.3.28 Spesifikasi Desain Kelas komentar\_model

<b>komentar_model</b>	<b>&lt;&lt;entity&gt;&gt;</b>
-----------------------	-------------------------------

-id\_komentar: Integer

Atribut ini digunakan untuk menyimpan data id komentar

-nama\_pengirim: String

Atribut ini digunakan untuk menyimpan data nama pengirim

-email\_pengirim: String

Atribut ini digunakan untuk menyimpan data email pengirim

-tanggal\_kirim: date

Atribut ini digunakan untuk menyimpan data tanggal kirim

-isi\_komentar: text

Atribut ini digunakan untuk menyimpan data isi komentar

+suratPembaca()

Default konstruktor, digunakan untuk inisialisasi semua

attribute dari kelas ini.

```
+insertDataSuratPembaca()
```

Operasi ini digunakan untuk memasukkan data surat pembaca ke dalam database.

```
+getDataSuratPembaca()
```

Operasi ini digunakan untuk mengambil data surat pembaca.

### 3 Deskripsi Dekomposisi

#### 3.1 Dekomposisi Data

##### 3.1.1 Deskripsi Entitas Data User

Nama	Tipe	Panjang	Keterangan
id_user	integer	-	Id_user, primary key
username	Varchar	20	Username dari user
password	Varchar	20	Password dari user

##### 3.1.2 Deskripsi Entitas Data Jemaat

Nama	Tipe	Panjang	Keterangan
id_jemaat	Integer	-	id_jemaat, Primary key
nama_jemaat	Varchar	50	Nama jemaat dari jemaat
alamat	Varchar	50	Alamat dari jemaat
gender	Varchar	10	Gender dari jemaat
lingkungan	Varchar	15	Lingkungan dari jemaat

##### 3.1.3 Deskripsi Entitas Data Kebaktian

Nama	Tipe	Panjang	Keterangan
id_kebaktian	Integer	-	id_kebaktian dari kebaktian, primary key

hari	Varchr	10	Hari dari kebaktian
tgl_kebaktian	Date	-	Tanggal kebaktian dari kebaktian
jam	Time	-	Jam dari kebaktian
romol	varchar	25	Romo 1 dari kebaktian
Romo2	Varchar	25	Romo 2 dari kebaktian
tempat	Varchar	10	Tempat dari kebaktian
bahasa	Varchar	10	Bahasa dari kebaktian
tema	Varchar	10	Tema dari kebaktian

#### 3.1.4 Deskripsi Entitas Data ref\_petugas

Nama	Tipe	Panjang	Keterangan
id_refpetugas	Integer	-	id_refpetugas dari ref_petugas, primary key
nama_refpetugas	Varchar	25	Nama dari ref_petugas

#### 3.1.5 Deskripsi Entitas Data tr\_petugas

Nama	Tipe	Panjang	Keterangan
id_tr_petugas	Integer	-	id_tr_petugas dari tr_petugas, primary key
id_kebaktian	Integer	-	id kebaktian dari kebaktian, foreign key
id_jemaat	Integer	-	id jemaat dari jemaat, foreign key
id_refpetugas	Integer	-	id_refpetugas dari ref_petugas, foreign key

### 3.1.6 Deskripsi Entitas Data Pengumuman

Nama	Tipe	Panjang	Keterangan
id_pengumuman	Integer	-	id_pengumuman dari pengumuman, primary key
judul	Varchar	50	Judul dari pengumuman
deskripsi	Text	-	Deskripsi dari pengumuman

### 3.1.7 Deskripsi Entitas Data Profil

Nama	Tipe	Panjang	Keterangan
id_profil	Integer	-	id_profil dari profil, primary key
nama_profil	Varchar	15	Nama profil dari profil
deskripsi	Text	-	Deskripsi dari profil

### 3.1.8 Deskripsi Entitas Data Kumpulan

Nama	Tipe	Panjang	Keterangan
id_kumpulan	Integer	-	id_kumpulan dari kumpulan, primary key
nama_kumpulan	Varchar	15	Nama kumpulan dari kumpulan
deskripsi	Text	-	Deskripsi dari kumpulan

### 3.1.9 Deskripsi Entitas Data Lingkungan

Nama	Tipe	Panjang	Keterangan
id_lingkungan	Integer	-	id_lingkungan dari lingkungan, primary key

nama_lingkungan	Varchar	25	Nama lingkungan dari lingkungan
deskripsi	Text	-	Deskripsi dari lingkungan

### 3.1.10 Deskripsi Entitas Data Komentar

Nama	Tipe	Panjang	Keterangan
id_komentar	Integer	-	id_komentar dari komentar, primary key
nama_pengirim	Varchar	25	Nama pengirim dari komentar
email_pengirim	Varchar	50	Email pengirim dari komentar
tanggal_kirim	date	-	Tanggal kirim dari komentar
isi_komentar	Text	-	Isi komentar dari komentar

## 4. Perancangan Antarmuka

### 4.1. Sketsa UI dan Deskripsinya

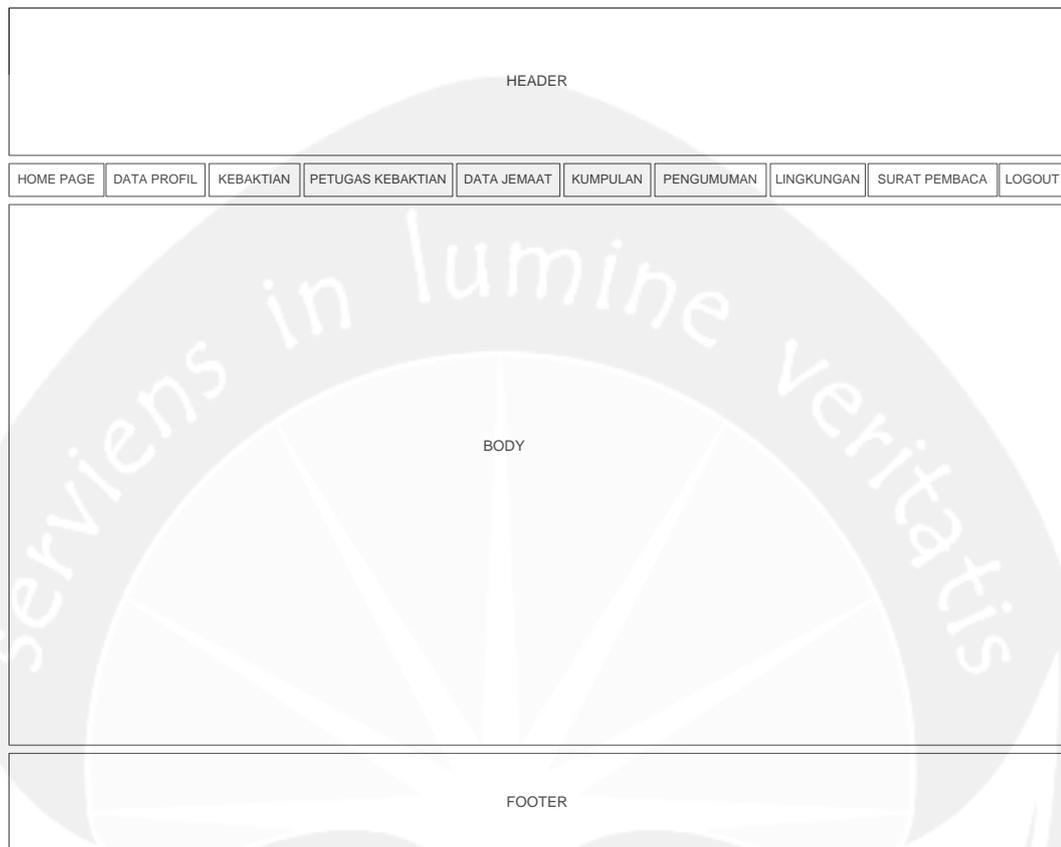
#### 4.1.1. Login Admin

The image shows a wireframe for a web login page. At the top, there is a rectangular box labeled "HEADER". Below the header, the main content area contains a login form. The form has two input fields: "USERNAME" and "PASSWORD", each with a corresponding text box. Below these fields is a "LOGIN" button. The entire design is overlaid on a large, faint watermark of a university logo with the motto "serviens in lumine veritatis".

Gambar 4.1 Rancangan Antarmuka Login Web

Antarmuka ini merupakan antarmuka login web yang digunakan oleh admin untuk dapat mengakses admin web form PWGCI. Pada antarmuka login web ini, meminta user (admin) untuk memasukkan username dan passwordnya.

#### 4.1.2. Main Web Admin



Gambar 4.2 Rancangan Antarmuka Main Web Admin

Antarmuka ini merupakan antarmuka main web admin setelah user (admin) melakukan login.

### 4.1.3. Mengelola Jemaat

HEADER																	
DATA PROFIL	KEBAKTIAN	PETUGAS KEBAKTIAN	DATA JEMAAT	KUMPULAN	PENGUMUMAN	LINGKUNGAN	LOGOUT										
<div style="display: flex; justify-content: space-between; align-items: center;"> <span>ADD</span> <span>DATA JEMAAT</span> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 5%;">NO</th> <th style="width: 25%;">NAMA</th> <th style="width: 25%;">ALAMAT</th> <th style="width: 10%;">GENDER</th> <th style="width: 15%;">LINGKUNGAN</th> <th style="width: 20%;">ACTION</th> </tr> </thead> <tbody> <tr> <td style="height: 150px;"></td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center; vertical-align: top;">                     EDIT    DELETE                 </td> </tr> </tbody> </table>						NO	NAMA	ALAMAT	GENDER	LINGKUNGAN	ACTION						EDIT    DELETE
NO	NAMA	ALAMAT	GENDER	LINGKUNGAN	ACTION												
					EDIT    DELETE												
FOOTER																	

Gambar 4.3 Rancangan Antarmuka mengelola jemaat

Antarmuka ini digunakan administrator untuk mengelola data jemaat. Dalam antarmuka ini terdapat display data jemaat, pada kanan form display jemaat terdapat form action, yang berisi action untuk mengedit dan mendelete data jemaat. Jika administrator memilih action delete pada form action, maka sistem akan menghapus data jemaat di dalam database. Tetapi jika memilih action edit maka akan muncul form untuk melakukan perubahan data jemaat.

Diatas form display data jemaat, ada button action add. Jika button action add diklik maka akan muncul form untuk menambahkan data jemaat di database.

HEADER																	
DATA PROFIL	KEBAKTIAN	PETUGAS KEBAKTIAN	DATA JEMAAT	KUMPULAN	PENGUMUMAN	LINGKUNGAN	LOGOUT										
ADD JEMAAT																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">NAMA</td> <td style="padding: 5px;"><input type="text"/></td> </tr> <tr> <td style="padding: 5px;">GENDER</td> <td style="padding: 5px;"><input type="text" value="▼"/></td> </tr> <tr> <td style="padding: 5px;">ALAMAT</td> <td style="padding: 5px;"><input type="text"/></td> </tr> <tr> <td style="padding: 5px;">LINGKUNGAN</td> <td style="padding: 5px;"><input type="text" value="▼"/></td> </tr> <tr> <td colspan="2" style="text-align: center; padding: 5px;"><input type="button" value="ADD"/></td> </tr> </table>								NAMA	<input type="text"/>	GENDER	<input type="text" value="▼"/>	ALAMAT	<input type="text"/>	LINGKUNGAN	<input type="text" value="▼"/>	<input type="button" value="ADD"/>	
NAMA	<input type="text"/>																
GENDER	<input type="text" value="▼"/>																
ALAMAT	<input type="text"/>																
LINGKUNGAN	<input type="text" value="▼"/>																
<input type="button" value="ADD"/>																	
FOOTER																	

**Gambar 4.4 Rancangan Antarmuka Mengelola Jemaat - Add Data Jemaat**

Pada antarmuka Add Data Jemaat ini terdapat textbox nama , combobox gender, textbox alamat dan combobox lingkungan. Administrator memasukan data jemaat pada textbox nama, combobox gender, textbox alamat dan combobox lingkungan. Jika tombol add diklik maka data jemaat akan disimpan ke database.

HEADER							
DATA PROFIL	KEBAKTIAN	PETUGAS KEBAKTIAN	DATA JEMAAT	KUMPULAN	PENGUMUMAN	LINGKUNGAN	LOGOUT
EDIT JEMAAT							
NAMA	<input type="text"/>						
GENDER	<input style="width: 100%;" type="text"/>						
ALAMAT	<input type="text"/>						
LINGKUNGAN	<input style="width: 100%;" type="text"/>						
<input type="button" value="EDIT"/>							
FOOTER							

**Gambar 4.5 Rancangan Antarmuka Mengelola Jemaat - Edit Data Jemaat**

Pada antarmuka Edit Data Jemaat ini terdapat textbox nama, combobox gender, textbox alamat dan combobox lingkungan. Administrator memasukan data jemaat pada textbox nama, combobox gender, textbox alamat dan combobox lingkungan. Jika tombol edit diklik maka data jemaat yang baru akan tersimpan ke database.

#### 4.1.4. Mengelola Kebaktian

The interface is divided into several sections:

- HEADER:** A large empty rectangular box at the top.
- Navigation Menu:** A row of buttons including 'DATA PROFIL', 'KEBAKTIAN', 'PETUGAS KEBAKTIAN', 'DATA JEMAAT', 'KUMPULAN', 'PENGUMUMAN', and 'LOGOUT'.
- Main Content Area:**
  - JADWAL KEBAKTIAN:** A title above a table.
  - ADD:** A button to the left of the table.
  - Table:** A table with columns: NO, HARI, JAM, TANGGAL, ROMO, TEMA, TEMPAT, and ACTION. The ACTION column contains 'EDIT' and 'DELETE' links.
- FOOTER:** A large empty rectangular box at the bottom.

Gambar 4.6 Rancangan Antarmuka mengelola kebaktian

Antarmuka ini digunakan administrator untuk mengelola data kebaktian. Dalam antarmuka ini terdapat display data kebaktian, pada kanan form display kebaktian terdapat form action, yang berisi action untuk mengedit dan mendelete data kebaktian. Jika administrator memilih action delete pada form action, maka sistem akan menghapus data kebaktian di dalam database. Tetapi jika memilih action edit maka akan muncul form untuk melakukan perubahan data kebaktian.

Diatas form display data kebaktian, ada button action add. Jika button action add diklik maka akan muncul form untuk menambahkan data kebaktian di database.

HEADER

DATA PROFIL
KEBAKTIAN
PETUGAS KEBAKTIAN
DATA JEMAAT
KUMPULAN
PENGUMUMAN
LOGOUT

ADD KEBAKTIAN

HARI	<input type="text"/>
TANGGAL	<input type="text"/>
JAM	<input type="text"/>
TEMPAT	<input style="border-bottom: 1px solid black; border-top: 1px solid black; border-right: 1px solid black; border-left: 1px solid black; width: 100%;" type="text"/>
TEMA	<input style="border-bottom: 1px solid black; border-top: 1px solid black; border-right: 1px solid black; border-left: 1px solid black; width: 100%;" type="text"/>
ROMO 1	<input style="border-bottom: 1px solid black; border-top: 1px solid black; border-right: 1px solid black; border-left: 1px solid black; width: 100%;" type="text"/>
ROMO 2	<input style="border-bottom: 1px solid black; border-top: 1px solid black; border-right: 1px solid black; border-left: 1px solid black; width: 100%;" type="text"/>
BAHASA	<input style="border-bottom: 1px solid black; border-top: 1px solid black; border-right: 1px solid black; border-left: 1px solid black; width: 100%;" type="text"/>

FOOTER

**Gambar 4.7 Rancangan Antarmuka Mengelola Kebaktian - Add Data Kebaktian**

Pada antarmuka Add Data Kebaktian ini terdapat textbox hari , textbox tanggal, textbobox jam, combobox tempat, combobox tema, combobox romo 1, combobox romo 2 dan combobox bahasa. Administrator memasukan data jemaat pada textbox hari , textbox tanggal, textbobox jam, combobox tempat, combobox tema, combobox romo 1, combobox romo 2 dan combobox bahasa. Jika tombol add diklik maka data kebaktian akan disimpan ke database.

HEADER						
DATA PROFIL	KEBAKTIAN	PETUGAS KEBAKTIAN	DATA JEMAAT	KUMPULAN	PENGUMUMAN	LOGOUT
EDIT KEBAKTIAN						
HARI	<input type="text"/>					
TANGGAL	<input type="text"/>					
JAM	<input type="text"/>					
TEMPAT	<input style="border-bottom: 1px solid black; border-right: 1px solid black; border-left: 1px solid black; border-top: 1px solid black; width: 100%;" type="text"/>					
TEMA	<input style="border-bottom: 1px solid black; border-right: 1px solid black; border-left: 1px solid black; border-top: 1px solid black; width: 100%;" type="text"/>					
ROMO 1	<input style="border-bottom: 1px solid black; border-right: 1px solid black; border-left: 1px solid black; border-top: 1px solid black; width: 100%;" type="text"/>					
ROMO 2	<input style="border-bottom: 1px solid black; border-right: 1px solid black; border-left: 1px solid black; border-top: 1px solid black; width: 100%;" type="text"/>					
BAHASA	<input style="border-bottom: 1px solid black; border-right: 1px solid black; border-left: 1px solid black; border-top: 1px solid black; width: 100%;" type="text"/>					
<input type="button" value="EDIT"/>						
FOOTER						

**Gambar 4.8 Rancangan Antarmuka Mengelola Kebaktian - Edit Data Kebaktian**

Pada antarmuka Edit Data Kebaktian ini terdapat textbox hari , textbox tanggal, textbobox jam, combobox tempat, combobox tema, combobox romo 1, combobox romo 2 dan combobox bahasa. Administrator memasukan data kebaktian pada textbox hari , textbox tanggal, textbobox jam, combobox tempat, combobox tema, combobox romo 1, combobox romo 2 dan combobox bahasa. Jika tombol edit diklik maka data kebaktian yang baru akan tersimpan ke database.

#### 4.1.5. Mengelola Petugas Kebaktian

HEADER

DATA PROFIL
KEBAKTIAN
PETUGAS KEBAKTIAN
DATA JEMAAT
KUMPULAN
PENGUMUMAN
LOGOUT

ADD
DATA PETUGAS KEBAKTIAN

NO	PRODIAKON	LEKTOR	PA	KOLEKTE	KOOR	MISA	ACTION
							EDIT DELETE

FOOTER

Gambar 4.9 Rancangan Antarmuka Mengelola Data Petugas Kebaktian

Antarmuka ini digunakan administrator untuk mengelola data petugas kebaktian. Dalam antarmuka ini terdapat display data petugas kebaktian, pada kanan form display data petugas kebaktian terdapat form action, yang berisi action untuk mengedit dan mendelete data petugas kebaktian. Jika administrator memilih action delete pada form action, maka sistem akan menghapus data petugas kebaktian di dalam database. Tetapi jika memilih action edit maka akan muncul form untuk melakukan perubahan data petugas kebaktian.

Diatas form display data petugas kebaktian, ada button action add. Jika button action add diklik maka akan muncul form untuk menambahkan data petugas kebaktian di database.

HEADER																		
DATA PROFIL	KEBAKTIAN	PETUGAS KEBAKTIAN	DATA JEMAAT	KUMPULAN	PENGUMUMAN	LOGOUT												
ADD PETUGAS KEBAKTIAN																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">NAMA PETUGAS</td> <td style="padding: 5px;"><input type="text"/></td> <td style="padding: 5px; text-align: right;">▼</td> </tr> <tr> <td style="padding: 5px;">PETUGAS</td> <td style="padding: 5px;"><input type="text"/></td> <td style="padding: 5px; text-align: right;">▼</td> </tr> <tr> <td style="padding: 5px;">KEBAKTIAN</td> <td style="padding: 5px;"><input type="text"/></td> <td style="padding: 5px; text-align: right;">▼</td> </tr> <tr> <td colspan="3" style="padding: 5px; text-align: center;"> <input type="button" value="ADD"/> </td> </tr> </table>							NAMA PETUGAS	<input type="text"/>	▼	PETUGAS	<input type="text"/>	▼	KEBAKTIAN	<input type="text"/>	▼	<input type="button" value="ADD"/>		
NAMA PETUGAS	<input type="text"/>	▼																
PETUGAS	<input type="text"/>	▼																
KEBAKTIAN	<input type="text"/>	▼																
<input type="button" value="ADD"/>																		
FOOTER																		

**Gambar 4.10 Rancangan Antarmuka Mengelola Data Petugas Kebaktian - Add Data Petugas Kebaktian**

Pada antarmuka Add Data Petugas Kebaktian ini terdapat combobox nama petugas, combobox petugas dan combobox kebaktian. Administrator memasukan data petugas kebaktian pada combobox nama petugas, combobox petugas dan combobox kebaktian. Jika tombol add diklik maka data petugas kebaktian akan disimpan ke database.

HEADER						
DATA PROFIL	KEBAKTIAN	PETUGAS KEBAKTIAN	DATA JEMAAT	KUMPULAN	PENGUMUMAN	LOGOUT
EDIT PETUGAS KEBAKTIAN						
PRODIAKON	<input type="text"/>					▼
LEKTOR	<input type="text"/>					▼
PA	<input type="text"/>					▼
<input type="button" value="EDIT"/>						
FOOTER						

**Gambar 4.11 Rancangan Antarmuka Mengelola Data Petugas Kebaktian - Edit Data Petugas Kebaktian**

Pada antarmuka Edit Data Petugas Kebaktian ini terdapat combobox prodiakon, combobox lektor, combobox putra altar. Administrator memasukan data petugas kebaktian pada combobox prodiakon, combobox lektor, combobox putra altar. Jika tombol edit diklik maka data petugas kebaktian yang diedit akan disimpan ke database.

#### 4.1.6. Mengelola Profil

The interface is divided into several sections:

- HEADER:** A large empty rectangular box at the top.
- Navigation Menu:** A horizontal row of buttons: DATA PROFIL, KEBAKTIAN, PETUGAS KEBAKTIAN, DATA JEMAAT, KUMPULAN, PENGUMUMAN, and LOGOUT.
- Main Content Area:**
  - ADD:** A button located above the table.
  - PROFIL:** A table with the following structure:

NO	NAMA	DESKRIPSI	ACTION
			EDIT DELETE
- FOOTER:** A large empty rectangular box at the bottom.

Gambar 4.12 Rancangan Antarmuka Mengelola Profil

Antarmuka ini digunakan administrator untuk mengelola profil. Dalam antarmuka ini terdapat display profil, pada kanan form display profil terdapat form action, yang berisi action untuk mengedit dan mendelete profil. Jika administrator memilih action delete pada form action, maka sistem akan menghapus profil di dalam database. Tetapi jika memilih action edit maka akan muncul form untuk melakukan perubahan profil.

Diatas form display profil, ada button action add. Jika button action add diklik maka akan muncul form untuk menambahkan profil di database.

The image shows a web application interface for managing profiles. It consists of several sections:

- HEADER:** A rectangular box at the top of the page.
- Navigation Menu:** A horizontal row of buttons below the header, including "DATA PROFIL", "KEBAKTIAN", "PETUGAS KEBAKTIAN", "DATA JEMAAT", "KUMPULAN", "PENGUMUMAN", and "LOGOUT".
- ADD PROFIL:** The main content area, containing:
  - A label "NAMA" next to a text input field.
  - A label "DESKRIPSI" next to a larger text area.
  - An "ADD" button located below the text area.
- FOOTER:** A rectangular box at the bottom of the page.

Gambar 4.13 Rancangan Antarmuka Mengelola Profil - Add Data Profil

Pada antarmuka Add data profil ini terdapat textbox nama dan textarea deskripsi. Administrator memasukan data profil pada textbox nama dan textarea deskripsi. Jika tombol add diklik maka data profil akan disimpan ke database.

HEADER

EDIT PROFIL

NAMA

DESKRIPSI

FOOTER

**Gambar 4.14 Rancangan Antarmuka Mengelola Profil - Edit Data Profil**

Pada antarmuka edit data profil ini terdapat textbox nama dan textarea deskripsi. Administrator memasukan data profil pada textbox nama dan textarea deskripsi. Jika tombol edit diklik maka data profil yang diedit akan disimpan ke database.

#### 4.1.7. Mengelola Kumpulan

The interface consists of several sections:

- HEADER:** A rectangular box at the top.
- Navigation Menu:** A row of buttons: DATA PROFIL, KEBAKTIAN, PETUGAS KEBAKTIAN, DATA JEMAAT, **KUMPULAN** (highlighted), PENGUMUMAN, and LOGOUT.
- Main Content Area:**
  - An **ADD** button on the left.
  - A **KUMPULAN** title above a table.
  - Table:** A table with 4 columns: NO, NAMA, DESKRIPSI, and ACTION. The ACTION column contains 'EDIT' and 'DELETE' links.
- FOOTER:** A rectangular box at the bottom.

Gambar 4.15 Rancangan Antarmuka Mengelola Kumpulan

Antarmuka ini digunakan administrator untuk mengelola kumpulan. Dalam antarmuka ini terdapat display kumpulan, pada kanan form display kumpulan terdapat form action, yang berisi action untuk mengedit dan mendelete kumpulan. Jika administrator memilih action delete pada form action, maka sistem akan menghapus kumpulan di dalam database. Tetapi jika memilih action edit maka akan muncul form untuk melakukan perubahan kumpulan.

Diatas form display kumpulan, ada button action add. Jika button action add diklik maka akan muncul form untuk menambahkan kumpulan di database.

The image shows a web application interface for managing groups. At the top is a 'HEADER' section. Below it is a navigation menu with buttons for 'DATA PROFIL', 'KEBAKTIAN', 'PETUGAS KEBAKTIAN', 'DATA JEMAAT', 'KUMPULAN', 'PENGUMUMAN', and 'LOGOUT'. The main content area is titled 'ADD KUMPULAN' and contains a form with two input fields: 'NAMA' (Name) and 'DESKRIPSI' (Description). Below the 'DESKRIPSI' field is an 'ADD' button. At the bottom of the page is a 'FOOTER' section.

**Gambar 4.16 Rancangan Antarmuka Mengelola Kumpulan - Add Data Kumpulan**

Pada antarmuka add data kumpulan ini terdapat textbox nama dan textarea deskripsi. Administrator memasukan data kumpulan pada textbox nama dan textarea deskripsi. Jika tombol add diklik maka data kumpulan akan disimpan ke database.

HEADER						
DATA PROFIL	KEBAKTIAN	PETUGAS KEBAKTIAN	DATA JEMAAT	KUMPULAN	PENGUMUMAN	LOGOUT
EDIT KUMPULAN						
NAMA	<input style="width: 100%;" type="text"/>					
DESKRIPSI	<div style="border: 1px solid black; height: 100px;"></div>					
<input style="width: 100px;" type="button" value="EDIT"/>						
FOOTER						

**Gambar 4.17 Rancangan Antarmuka Mengelola Kumpulan - Edit Data Kumpulan**

Pada antarmuka edit data kumpulan ini terdapat textbox nama dan textarea deskripsi. Administrator memasukan data kumpulan pada textbox nama dan textarea deskripsi. Jika tombol edit diklik maka data kumpulan yang diedit akan disimpan ke database.

#### 4.1.8. Mengelola Pengumuman

HEADER

DATA PROFIL KEBAKTIAN PETUGAS KEBAKTIAN DATA JEMAAT KUMPULAN PENGUMUMAN LOGOUT

ADD PENGUMUMAN

NO	JUDUL	DESKRIPSI	ACTION
			EDIT DELETE

FOOTER

Gambar 4.18 Rancangan Antarmuka Mengelola Pengumuman

Antarmuka ini digunakan administrator untuk mengelola pengumuman. Dalam antarmuka ini terdapat display data pengumuman, pada kanan form display data pengumuman terdapat form action, yang berisi action untuk mengedit dan mendelete pengumuman. Jika administrator memilih action delete pada form action, maka sistem akan menghapus data pengumuman di dalam database. Tetapi jika memilih action edit maka akan muncul form untuk melakukan perubahan data pengumuman.

Diatas form display pengumuman, ada button action add. Jika button action add diklik maka akan muncul form untuk menambahkan pengumuman di database.

The image shows a web application interface for managing announcements. At the top is a 'HEADER' section. Below it is a navigation menu with buttons for 'DATA PROFIL', 'KEBAKTIAN', 'PETUGAS KEBAKTIAN', 'DATA JEMAAT', 'KUMPULAN', 'PENGUMUMAN', and 'LOGOUT'. The main content area is titled 'ADD PENGUMUMAN' and contains a form with two input fields: 'JUDUL' (a text box) and 'DESKRIPSI' (a text area). Below these fields is an 'ADD' button. At the bottom of the page is a 'FOOTER' section.

Gambar 4.19 Rancangan Antarmuka Mengelola Pengumuman - Add Data Pengumuman

Pada antarmuka Add pengumuman ini terdapat textbox judul dan textarea deskripsi. Administrator memasukan data pengumuman pada textbox judul dan textarea deskripsi. Jika tombol add diklik maka data pengumuman akan disimpan ke database.

HEADER

EDIT PENGUMUMAN

JUDUL

DESKRIPSI

FOOTER

**Gambar 4.20 Rancangan Antarmuka Mengelola Pengumuman - Edit Data Pengumuman**

Pada antarmuka edit data pengumuman ini terdapat textbox judul dan textarea deskripsi. Administrator memasukan data pengumuman pada textbox judul dan textarea deskripsi. Jika tombol edit diklik maka data pengumuman yang diedit akan disimpan ke database.

#### 4.1.9. Mengelola Lingkungan

HEADER							
DATA PROFIL	KEBAKTIAN	PETUGAS KEBAKTIAN	DATA JEMAAT	KUMPULAN	PENGUMUMAN	LINGKUNGAN	LOGOUT
ADD				DATA LINGKUNGAN			
NO	NAMA LINGKUNGAN	DESKRIPSI	ACTION				
			EDIT	DELETE			
FOOTER							

Gambar 4.21 Rancangan Antarmuka Mengelola Lingkungan

Antarmuka ini digunakan administrator untuk mengelola lingkungan. Dalam antarmuka ini terdapat display data lingkungan, pada kanan form display data lingkungan terdapat form action, yang berisi action untuk mengedit dan mendelete lingkungan. Jika administrator memilih action delete pada form action, maka sistem akan menghapus data lingkungan di dalam database. Tetapi jika memilih action edit maka akan muncul form untuk melakukan perubahan data lingkungan.

Diatas form display lingkungan, ada button action add. Jika button action add diklik maka akan muncul form untuk menambahkan data lingkungan di database.

HEADER													
DATA PROFIL	KEBAKTIAN	PETUGAS KEBAKTIAN	DATA JEMAAT	KUMPULAN	PENGUMUMAN	LINGKUNGAN	LOGOUT						
ADD LINGKUNGAN													
<table style="width: 100%; border: 1px solid black;"> <tr> <td style="width: 20%; padding: 5px;">NAMA LINGKUNGAN</td> <td style="border: 1px solid black; width: 80%; height: 20px;"></td> </tr> <tr> <td style="padding: 5px;">DESKRIPSI</td> <td style="border: 1px solid black; height: 100px;"></td> </tr> <tr> <td colspan="2" style="text-align: center; padding: 5px;"> <input type="button" value="ADD"/> </td> </tr> </table>								NAMA LINGKUNGAN		DESKRIPSI		<input type="button" value="ADD"/>	
NAMA LINGKUNGAN													
DESKRIPSI													
<input type="button" value="ADD"/>													
FOOTER													

**Gambar 4.22 Rancangan Antarmuka Mengelola Lingkungan - Add Data Lingkungan**

Pada antarmuka edit data lingkungan ini terdapat textbox nama lingkungan dan textarea deskripsi. Administrator memasukan data lingkungan pada textarea nama lingkungan dan textarea deskripsi. Jika tombol edit diklik maka data lingkungan yang diedit akan disimpan ke database.

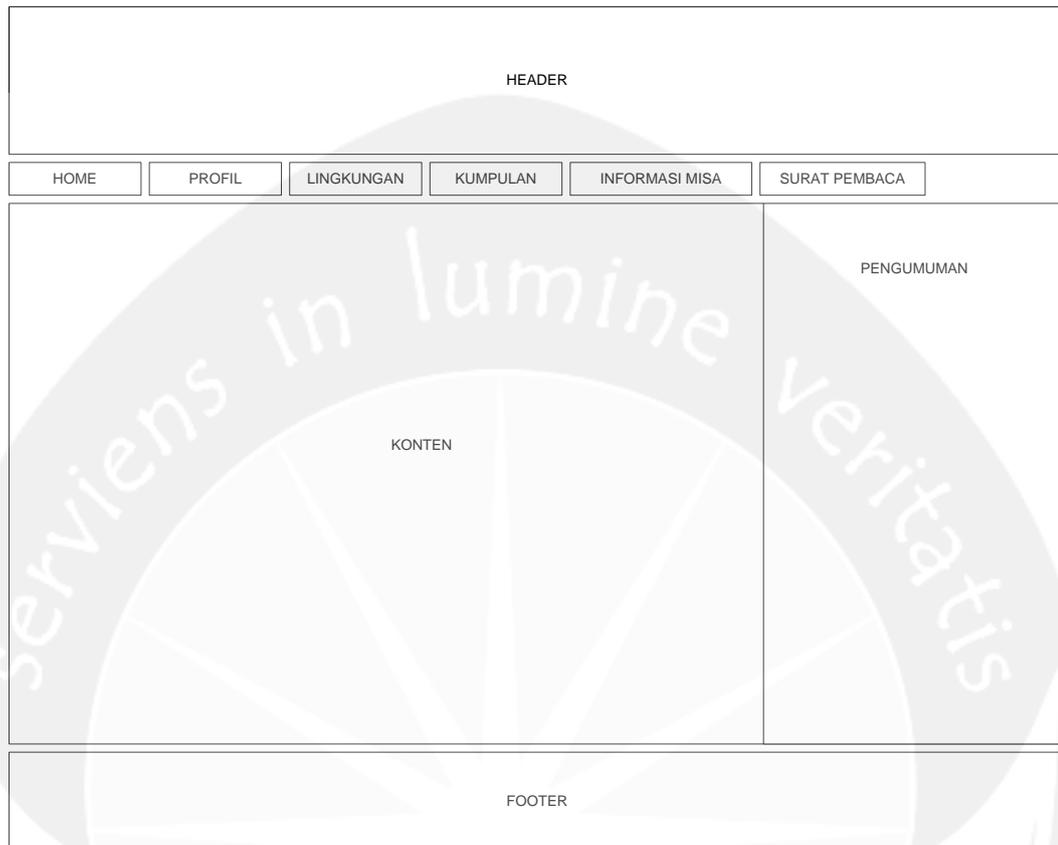
HEADER													
DATA PROFIL	KEBAKTIAN	PETUGAS KEBAKTIAN	DATA JEMAAT	KUMPULAN	PENGUMUMAN	LINGKUNGAN	LOGOUT						
EDIT LINGKUNGAN													
<table style="width: 100%; border: 1px solid black;"> <tr> <td style="width: 20%; padding: 5px;">NAMA LINGKUNGAN</td> <td style="border: 1px solid black; width: 80%; height: 20px;"></td> </tr> <tr> <td style="padding: 5px;">DESKRIPSI</td> <td style="border: 1px solid black; height: 100px;"></td> </tr> <tr> <td colspan="2" style="text-align: center; padding: 10px;"> <input type="button" value="EDIT"/> </td> </tr> </table>								NAMA LINGKUNGAN		DESKRIPSI		<input type="button" value="EDIT"/>	
NAMA LINGKUNGAN													
DESKRIPSI													
<input type="button" value="EDIT"/>													
FOOTER													

**Gambar 4.23 Rancangan Antarmuka Mengelola Lingkungan - Edit Data Lingkungan**

Pada antarmuka edit data lingkungan ini terdapat textbox nama lingkungan dan textarea deskripsi. Administrator memasukan data lingkungan pada textarea nama lingkungan dan textarea deskripsi. Jika tombol edit diklik maka data lingkungan yang diedit akan disimpan ke database.



#### 4.1.11. Main Web Umum



Gambar 4.25 Rancangan Antarmuka Main Web Umum

Antarmuka ini merupakan antarmuka untuk menampilkan main web untuk pengunjung website gereja st. Yohanes rasul somohitan.

# SKPL

**SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK**

**PEMBANGUNAN WEBSITE GEREJA SOMOHITAN  
MENGUNAKAN FRAMEWORK CODEIGNITER**

**PWGCI**

**Untuk :**

**Universitas Atma Jaya Yogyakarta**

**Dipersiapkan oleh:**

**Yulianus Andri Ardiyanto / 4980**

**Program Studi Teknik Informatika - Fakultas Teknologi  
Industri**

**Universitas Atma Jaya Yogyakarta**

	Program Studi Teknik Informatika  Fakultas Teknologi Industri	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<b>SKPL-PWGCI</b>		1/35
		<b>Revisi</b>		

## DAFTAR PERUBAHAN

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

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

### Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

## Daftar Isi

1	Pendahuluan.....	6
1.1	Tujuan.....	6
1.2	Lingkup Masalah.....	6
1.3	Definisi, Akronim dan Singkatan.....	7
Server.....		7
1.4	Referensi.....	7
1.5	Deskripsi umum (Overview).....	8
2	Deskripsi Kebutuhan.....	8
2.1	Perspektif produk.....	8
2.2	Fungsi Produk.....	9
2.3	Batasan-batasan.....	14
2.4	Asumsi dan Ketergantungan.....	15
3	Kebutuhan khusus.....	15
3.1	Kebutuhan antarmuka eksternal.....	15
3.2	Kebutuhan fungsionalitas Perangkat Lunak.....	17
4	Spesifikasi Rinci Kebutuhan.....	18
4.1	Spesifikasi Kebutuhan Fungsionalitas.....	18
5	Entity Relationship Diagram (ERD).....	35

## Daftar Gambar

1. Arsitektur Sistem PWGCI.....	9
2. Use Case Diagram .....	17



## 1 Pendahuluan

### 1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak PWGCI (Pembangunan Web Gereja Somohitan Menggunakan Framework CodeIgniter) 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-PWGCI ini juga mendefinisikan batasan perancangan perangkat lunak.

### 1.2 Lingkup Masalah

Perangkat Lunak PWGCI dikembangkan dengan tujuan untuk :

1. Mengelola kebaktian.
2. Mengelola petugas kebaktian.
3. Mengelola jemaat.
4. Mengelola pengumuman.
5. Mengelola profil.
6. Mengelola kumpulan.
7. Mengelola lingkungan.
8. Mengelola surat pembaca.

Dan berjalan pada lingkungan dengan platform Windows XP.

Program Studi Teknik Informatika	SKPL – PWGCI	6/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

### 1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-PWGCI-XXX	Kode yang merepresentasikan kebutuhan pada PWGCI (Pembangunan Web Gereja Somohitan Menggunakan Framework CodeIgniter) dimana XXX merupakan nomor fungsi produk.
PWGCI	Pembangunan Web Gereja Dengan Framework CodeIgniter.
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.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

### 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.

Program Studi Teknik Informatika	SKPL – PWGCI	7/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

## **1.5 Deskripsi umum (Overview)**

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

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

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak PWGCI yang akan dikembangkan.

## **2 Deskripsi Kebutuhan**

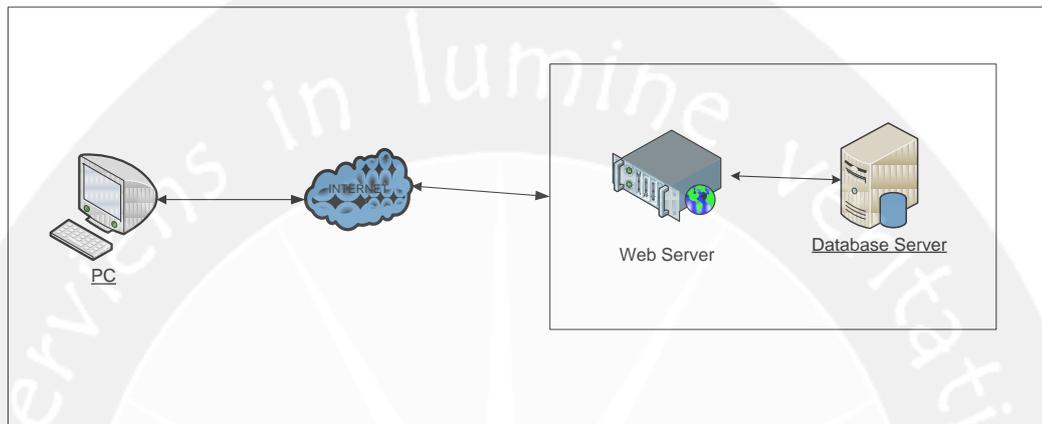
### **2.1 Perspektif produk**

Perangkat lunak PWGCI ini berjalan pada platform Windows XP, dan dibuat menggunakan bahasa pemrograman PHP dengan framework CodeIgniter. Sedangkan untuk lingkungan pemrogramannya menggunakan NetBeans IDE.

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (Graphical User Interface). Pada sistem ini, seperti terlihat pada gambar 1, arsitektur perangkat tersebut dengan menggunakan web aplikasi (application web).

Program Studi Teknik Informatika	SKPL – PWGCI	8/ 35
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		

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 station yang merequest melalui web application.



Gambar 1. Arsitektur Sistem PWGCI

## 2.2 Fungsi Produk

Fungsi produk perangkat lunak PWGCI adalah sebagai berikut :

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

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

### 2. Fungsi *Mengelola Kebaktian* (SKPL-PWGCI-002) .

Merupakan fungsi yang digunakan untuk mengelola kebaktian.

Fungsi *Mengelola Kebaktian* mencakup :

#### a. Fungsi *Entry Kebaktian* (SKPL-PWGCI-002-01) .

Merupakan fungsi yang digunakan untuk menambahkan data kebaktian yang baru.

b. Fungsi *Edit Kebaktian* (**SKPL-PWGCI-002-02**).  
Merupakan fungsi yang digunakan untuk mengubah data kebaktian.

c. Fungsi *Display Kebaktian* (**SKPL-PWGCI-002-03**).  
Merupakan fungsi yang digunakan untuk menampilkan data kebaktian.

d. Fungsi *Delete Kebaktian* (**SKPL-PWGCI-002-04**).  
Merupakan fungsi yang digunakan untuk menghapus data kebaktian.

3. Fungsi *Mengelola Petugas Kebaktian* (**SKPL-PWGCI-003**).

Merupakan fungsi yang digunakan untuk mengelola petugas kebaktian.

Fungsi mengelola petugas kebaktian meliputi:

a. Fungsi *Entry Petugas* (**SKPL-PWGCI-003-01**).

Merupakan fungsi yang digunakan untuk menambah data petugas kebaktian.

b. Fungsi *Edit Petugas* (**SKPL-PWGCI-003-02**).

Merupakan fungsi yang digunakan untuk mengubah data petugas kebaktian.

c. Fungsi *Display Petugas* (**SKPL-PWGCI-003-03**).

Merupakan fungsi yang digunakan untuk data menampilkan petugas kebaktian.

d. Fungsi *Delete Petugas* (**SKPL-PWGCI-003-04**).

Merupakan fungsi yang digunakan untuk menghapus data petugas kebaktian.

4. Fungsi *Mengelola Jemaat* (**SKPL-PWGCI-004**).

Merupakan fungsi yang digunakan untuk mengelola jemaat.

Fungsi mengelola jemaat meliputi:

a. Fungsi *Entry Data Jemaat* (**SKPL-PWGCI-004-01**).

Merupakan fungsi yang digunakan untuk menambah data jemaat.

b. Fungsi *Edit Jemaat* (**SKPL-PWGCI-004-02**).

Merupakan fungsi yang digunakan untuk mengubah data jemaat.

c. Fungsi *Display Jemaat* (**SKPL-PWGCI-004-03**).

Merupakan fungsi yang digunakan untuk menampilkan data jemaat.

d. Fungsi *Delete Jemaat* (**SKPL-PWGCI-004-04**).

Merupakan fungsi yang digunakan untuk menghapus data jemaat.

5. Fungsi *Mengelola Pengumuman* (**SKPL-PWGCI-005**)

Merupakan fungsi yang digunakan untuk mengelola pengumuman.

Fungsi *Mengelola Pengumuman* meliputi :

a. Fungsi *Entry Pengumuman* (**SKPL-PWGCI-005-01**).

Merupakan fungsi yang digunakan untuk menambahkan data pengumuman baru.

b. Fungsi *Edit Pengumuman* (**SKPL-PWGCI-005-02**).

Merupakan fungsi yang digunakan untuk mengubah data pengumuman.

c. Fungsi *Display Pengumuman* (**SKPL-PWGCI-005-03**).

Merupakan fungsi yang digunakan untuk menampilkan data pengumuman.

d. Fungsi *Delete Pengumuman* (**SKPL-PWGCI-005-04**).

Merupakan fungsi yang digunakan untuk menghapus data pengumuman.

6. Fungsi *Mengelola Profil* (**SKPL-PWGCI-006**)

Merupakan fungsi yang digunakan untuk mengelola profil.

Fungsi *Mengelola Profil* meliputi :

a. Fungsi *Entry Profil* (**SKPL-PWGCI-006-01**).

Merupakan fungsi yang digunakan untuk memasukkan data profil.

b. Fungsi *Edit Profil* (**SKPL-PWGCI-006-02**).

Merupakan fungsi yang digunakan untuk mengubah data profil.

c. Fungsi *Delete Profil* (**SKPL-PWGCI-006-03**).

Merupakan fungsi yang digunakan untuk menghapus data profil.

d. Fungsi *Display Profil* (**SKPL-PWGCI-006-04**).

Merupakan fungsi yang digunakan untuk menampilkan data profil.

7. Fungsi *Mengelola Kumpulan* (**SKPL-PWGCI-007**)

Merupakan fungsi yang digunakan untuk mengelola kumpulan.

Program Studi Teknik Informatika	SKPL – PWGCI	12/ 35
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		

Fungsi Mengelola Kumpulan meliputi :

a. Fungsi *Entry Kumpulan* (**SKPL-PWGCI-007-01**).

Merupakan fungsi yang digunakan untuk memasukkan data kumpulan.

b. Fungsi *Edit Kumpulan* (**SKPL-PWGCI-007-02**).

Merupakan fungsi yang digunakan untuk mengubah data kumpulan.

c. Fungsi *Delete Kumpulan* (**SKPL-PWGCI-007-03**).

Merupakan fungsi yang digunakan untuk menghapus data kumpulan.

d. Fungsi *Display Kumpulan* (**SKPL-PWGCI-007-04**).

Merupakan fungsi yang digunakan untuk menampilkan data kumpulan.

8. Fungsi *Mengelola Lingkungan* (**SKPL-PWGCI-008**)

Merupakan fungsi yang digunakan untuk mengelola data lingkungan.

Fungsi Mengelola Lingkungan meliputi :

a. Fungsi *Entry Lingkungan* (**SKPL-PWGCI-008-01**).

Merupakan fungsi yang digunakan untuk memasukkan data lingkungan.

b. Fungsi *Edit Lingkungan* (**SKPL-PWGCI-008-02**).

Merupakan fungsi yang digunakan untuk mengubah data lingkungan.

c. Fungsi *Delete Lingkungan* (**SKPL-PWGCI-008-03**).

Merupakan fungsi yang digunakan untuk menghapus data lingkungan.

Program Studi Teknik Informatika	SKPL – PWGCI	13/ 35
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		

d. Fungsi *Display Lingkungan* (**SKPL-PWGCI-008-04**).

Merupakan fungsi yang digunakan untuk menampilkan data lingkungan.

9. Fungsi *Mengelola Surat Pembaca* (**SKPL-PWGCI-009**)

Merupakan fungsi yang digunakan untuk mengelola data komentar.

Fungsi *Mengelola komentar* meliputi :

a. Fungsi *Entry Surat Pembaca* (**SKPL-PWGCI-009-01**).

Merupakan fungsi yang digunakan untuk memasukkan data lingkungan surat pembaca.

b. Fungsi *Display Surat Pembaca* (**SKPL-PWGCI-009-02**).

Merupakan fungsi yang digunakan untuk mengubah data surat pembaca.

Karakteristik dari pengguna perangkat lunak PWGCI adalah sebagai berikut :

1. Memahami pengoperasian Personal Computer (PC).
2. Mengerti tentang internet.

### **2.3 Batasan-batasan**

Batasan-batasan dalam pengembangan perangkat lunak PWGCI tersebut adalah :

1. Kebijakan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak PWGCI.

Program Studi Teknik Informatika	SKPL – PWGCI	14/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

## 2. Keterbatasan perangkat keras

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

### 2.4 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada perangkat PC yang menggunakan system operasi Windows XP.

## 3 Kebutuhan khusus

### 3.1 Kebutuhan antarmuka eksternal

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

#### 3.1.1 Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form-form.

#### 3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak PWGCI adalah:

1. Keyboard
2. Mouse
3. CPU (Central Processing Unit)
4. Monitor

### 3.1.3 Antarmuka perangkat lunak

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

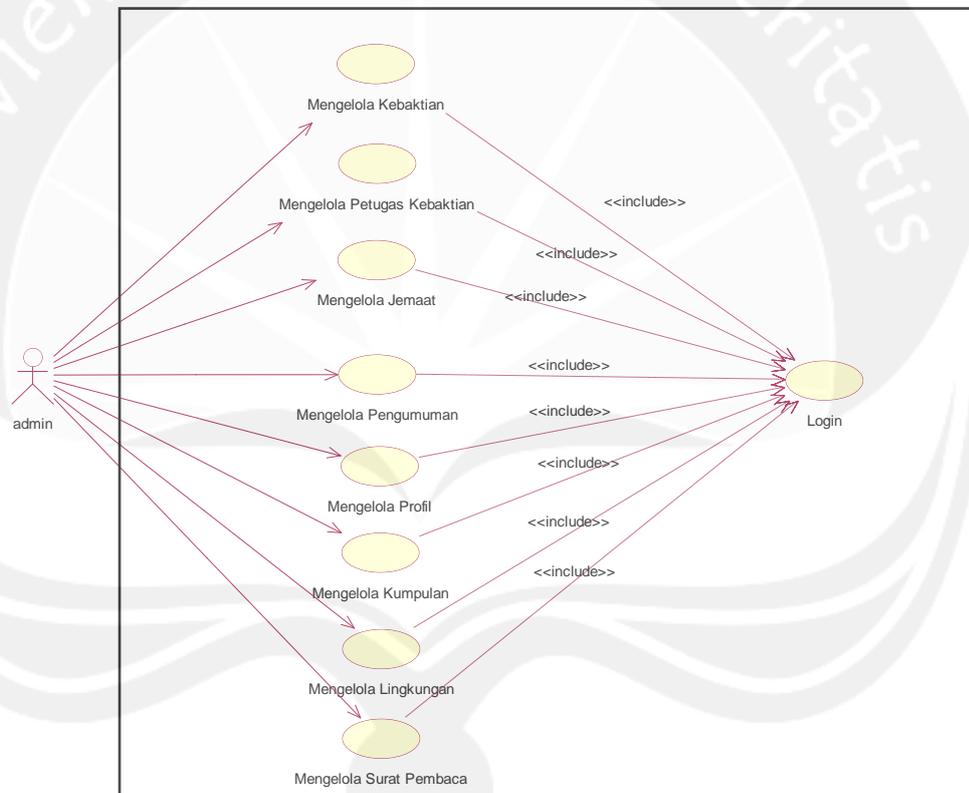
1. Nama : Windows  
Nomor Versi : Windows 2000/XP/Vista/7  
Sumber : Microsoft  
Sebagai sistem operasi untuk perangkat PC
2. Nama : Apache  
Nomor Versi : 2.0  
Sumber : Open Source (XAMPP)  
Sebagai web server.
3. Nama : Internet Explorer  
Nomor Versi : 6.0  
Sumber : Microsoft  
Sebagai web browser
4. Nama : MySQL  
Nomor versi : 5.0  
Sumber : Open Source  
Sebagai database management system (DBMS) yang digunakan untuk penyimpanan data di sisi server.
5. Nama : CodeIgniter 2.1.0  
Sumber : Open Source (EllisLab Inc)  
Sebagai framework pembangunan aplikasi berbasis php.
6. Nama : PHP 5.2.6  
Sumber : PHP  
Sebagai bahasa pemrograman pengembangan perangkat lunak pwgci

### 3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak PWGCI menggunakan protocol HTTP.

## 3.2 Kebutuhan fungsionalitas Perangkat Lunak

### 3.2.1 Use Case Diagram



Gambar 2. 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 dari user dan password yang berupa rangkaian karakter.

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

1. Admin

##### **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 sesuai

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 sesuai

1. Sistem menampilkan peringatan bahwa id user atau password tidak sesuai
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 : Mengelola Kebaktian

#### 1. Brief Description

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

#### 2. Primary Actor

1. Admin

#### 3. Supporting Actor

none

#### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengelola kebaktian.
2. Sistem memberikan pilihan untuk melakukan entry data kebaktian, edit data kebaktian, display data kebaktian, atau delete data kebaktian.
3. Aktor memilih untuk melakukan entry data kebaktian
  - A-1 Aktor memilih untuk melakukan edit data kebaktian
  - A-2 Aktor memilih untuk melakukan display data kebaktian
  - A-3 Aktor memilih untuk melakukan delete data kebaktian
4. Aktor menginputkan data kebaktian
5. Aktor meminta sistem untuk menyimpan data kebaktian yang telah diinputkan
6. Sistem mengecek data kebaktian yang telah diinputkan
  - E-1 Data kebaktian yang diinputkan aktor salah
7. Sistem menyimpan data kebaktian ke database

Program Studi Teknik Informatika	SKPL – PWGCI	19/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

8. Use Case selesai

### 5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit data kebaktian

1. Sistem menampilkan data kebaktian
2. Aktor mengedit data kebaktian yang sudah ditampilkan
3. Aktor meminta sistem untuk menyimpan data kebaktian yang telah diedit
4. Sistem melakukan pengecekan terhadap data kebaktian yang telah diedit

E-2 Data kebaktian yang telah diedit salah

5. Sistem menyimpan data kebaktian yang telah diedit ke database
6. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan display data kebaktian

1. Sistem menampilkan data kebaktian
2. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk melakukan delete data kebaktian

1. Sistem menampilkan data kebaktian
2. Aktor menghapus data kebaktian yang sudah ditampilkan
3. Aktor meminta system untuk menyimpan data kebaktian yang telah dihapus
4. Berlanjut ke Basic Flow langkah ke 8

### 6. Error Flow

E-1 Data kebaktian 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 kebaktian yang diedit aktor salah

1. Sistem memberikan pesan peringatan bahwa data yang diedit salah

Program Studi Teknik Informatika	SKPL – PWGCI	20/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

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

#### **7. PreConditions**

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

#### **8. PostConditions**

1. Data kebaktian di database telah terupdate

#### **4.1.3 Use case Spesification : Mengelola petugas kebaktian**

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

Use Case ini memungkinkan aktor untuk mengelola petugas kebaktian. Aktor dapat melakukan entry data petugas kebaktian, edit data petugas kebaktian, display data petugas kebaktian atau delete data petugas kebaktian.

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

1. Admin

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk mengelola petugas kebaktian.
2. Sistem memberikan pilihan untuk melakukan entry data petugas kebaktian, edit data petugas kebaktian, display data petugas kebaktian, atau delete data petugas kebaktian.
3. Aktor memilih untuk melakukan entry data petugas kebaktian
  - A-1 Aktor memilih untuk melakukan edit data petugas kebaktian
  - A-2 Aktor memilih untuk melakukan display data petugas kebaktian
  - A-3 Aktor memilih untuk melakukan delete data petugas kebaktian

Program Studi Teknik Informatika	SKPL – PWGCI	21/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Aktor menginputkan data petugas kebaktian
5. Aktor meminta sistem untuk menyimpan data petugas kebaktian yang telah diinputkan
6. Sistem mengecek data petugas kebaktian yang telah diinputkan  
E-1 Data petugas kebaktian yang diinputkan aktor salah
7. Sistem menyimpan data petugas kebaktian ke database
8. Use Case selesai

#### 5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit data petugas kebaktian

1. Sistem menampilkan data petugas kebaktian
2. Aktor memilih data petugas kebaktian yang akan di edit
3. Aktor meminta sistem untuk menyimpan data petugas kebaktian yang telah diedit
4. Sistem melakukan pengecekan terhadap data petugas kebaktian yang telah di edit

E-2 Data petugas yang telah di edit salah

5. Sistem mengupdate data petugas kebaktian di database
6. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan delete data petugas kebaktian

1. Sistem menampilkan data petugas kebaktian
2. Aktor memilih data petugas kebaktian yang akan di delete
3. Sistem menghapus data petugas kebaktian yang di delete oleh aktor
4. Sistem mengupdate data petugas kebaktian di database

5. Berlanjut ke Basic Flow langkah ke 8

Program Studi Teknik Informatika	SKPL – PWGCI	22/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

A-3 Aktor memilih untuk display data petugas kebaktian

1. Sistem menampilkan data petugas kebaktian dari database
2. Berlanjut ke Basic Flow langkah ke 8

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

E-1 Data petugas kebaktian yang diinputkan salah

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

E-2 Data petugas kebaktian yang diedit salah

1. Sistem memberikan peringatan bahwa data petugas kebaktian yang di edit salah
2. Kembali ke Alternative Flow A-1 Langkah ke 2

#### **7. PreConditions**

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

#### **8. PostConditions**

1. Data petugas kebaktian terupdate

### **4.1.4 Use case Spesification : Mengelola jemaat**

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

Use Case ini memungkinkan aktor untuk mengelola jemaat. Aktor dapat melakukan Entry Data jemaat, Edit Data jemaat, Delete Data jemaat, Display Data jemaat.

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

1. Admin

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk mengelola jemaat.

Program Studi Teknik Informatika	SKPL – PWGCI	23/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem memberikan pilihan untuk melakukan Entry Data jemaat, Edit Data jemaat, Delete Data jemaat, Search Data jemaat, Display Data jemaat .
3. Aktor memilih untuk melakukan Entry Data jemaat
  - A-1 Aktor memilih untuk edit data jemaat
  - A-2 Aktor Memilih untuk delete data jemaat
  - A-3 Aktor Memilih untuk display data jemaat
4. Aktor menginputkan data jemaat
5. Aktor meminta sistem untuk menyimpan data jemaat
6. Sistem mengecek data pasien yang telah diinputkan
  - E-1 Data jemaat yang diinputkan aktor salah
7. Sistem menyimpan data jemaat ke database
8. Use Case selesai

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

- A-1 Aktor memilih untuk mengedit data jemaat
  1. Sistem menampilkan data jemaat
  2. Aktor mengedit data jemaat yang akan dipilih
  3. Aktor meminta sistem untuk menyimpan data jemaat yang telah di edit
  4. Sistem melakukan pengecekan terhadap data jemaat yang telah di edit
    - E-2 Data jemaat yang telah di edit salah
  5. Sistem mengupdate data jemaat di database
  6. Berlanjut ke Basic Flow langkah ke 8
- A-2 Aktor memilih untuk melakukan delete data jemaat
  1. Sistem menampilkan data jemaat
  2. Aktor memilih data jemaat yang akan di delete
  3. Sistem menghapus data jemaat yang di delete oleh aktor
  4. Sistem mengupdate data jemaat di database
  5. Berlanjut ke Basic Flow langkah ke 8
- A-3 Aktor memilih untuk melakukan display data jemaat
  1. Sistem menampilkan data jemaat dari database
  2. Berlanjut ke Basic Flow langkah ke 8

## 6. Error Flow

E-1 Data jemaat yang diinputkan aktor salah

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

E-2 Data jemaat yang telah di edit salah

1. Sistem memberikan peringatan bahwa data jemaat yang telah di edit salah
2. Kembali ke Alternative Flow 1 langkah ke 2

## 7. PreConditions

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

## 8. PostConditions

1. Data jemaat terupdate

### 4.1.5 Use case Spesification : Mengelola pengumuman

#### 1. Brief Description

Use Case ini memungkinkan aktor untuk mengelola pengumuman. Aktor dapat melakukan entry data pengumuman, edit data pengumuman, display data pengumuman, delete data pengumuman.

#### 2. Primary Actor

1. Admin

#### 3. Supporting Actor

none

#### 4. Basic Flow

1. Use Case ini dimulai ketika Aktor memilih untuk mengelola pengumuman
2. Sistem memberikan pilihan untuk melakukan Entry Data pengumuman, Edit Data pengumuman, Delete Data pengumuman, Display Data pengumuman.
3. Aktor memilih untuk melakukan Entry Data Pengumuman

Program Studi Teknik Informatika	SKPL – PWGCI	25/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- A-1 Aktor Memilih untuk edit data pengumuman
- A-2 Aktor Memilih untuk delete pengumuman
- A-3 Aktor Memilih untuk display pengumuman
- 4. Aktor menginputkan data pengumuman
- 5. Aktor meminta sistem untuk menyimpan data pengumuman
- 6. Sistem mengecek data pengumuman yang telah diinputkan
  - E-1 Data pengumuman yang diinputkan aktor salah
- 7. Sistem menyimpan data pengumuman ke database
- 8. Use Case selesai

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

- A-1 Aktor memilih untuk melakukan edit data pengumuman
  - 1. Sistem menampilkan data pengumuman
  - 2. Aktor mengedit data pengumuman yang dipilih
  - 3. Aktor meminta sistem untuk menyimpan data pengumuman yang telah di edit
  - 4. Sistem melakukan pengecekan terhadap data pengumuman yang telah di edit
    - E-2 Data pengumuman yang telah di edit salah
  - 5. Sistem mengupdate data pengumuman di database
  - 6. Berlanjut ke Basic Flow langkah ke 8
- A-2 Aktor memilih untuk melakukan delete data pengumuman
  - 1. Sistem menampilkan data pengumuman dari database
  - 2. Aktor memilih data pengumuman yang akan di delete
  - 3. Sistem menghapus data pengumuman yang dipilih aktor
  - 4. Sistem mengupdate data pengumuman di database
  - 5. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk melakukan display data pengumuman

1. Sistem menampilkan data pengumuman dari database
2. Berlanjut ke Basic Flow langkah ke 8

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

E-1 Data pengumuman yang diinputkan aktor salah

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

E-2 Data pengumuman yang telah di edit salah

1. Sistem memberikan peringatan bahwa data pengumuman yang telah di edit salah
2. Kembali ke Alternative Flow 1 langkah ke 2

#### **7. PreConditions**

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

#### **8. PostConditions**

1. Data pengumuman terupdate

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

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

Use Case ini digunakan oleh aktor untuk mengelola profil. Aktor data melakukan entry data profil, edit data profil, delete data profil dan display data profil

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

1. Admin

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk mengelola profil

Program Studi Teknik Informatika	SKPL – PWGCI	27/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem memberikan pilihan untuk melakukan entry data profil, edit data profil, delete data profil, display data profil
3. Aktor memilih untuk melakukan entry data profil
  - A-1 Aktor memilih untuk edit data profil
  - A-2 Aktor memilih untuk delete data profil
  - A-3 Aktor memilih untuk display data profil
4. Aktor menginputkan data profil
5. Aktor meminta sistem untuk menyimpan data profil
6. Sistem mengecek data profil yang sudah diinputkan
  - E-1 Data profil yang diinputkan aktor salah
7. Sistem menyimpan data profil ke database
8. Use case selesai

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

- A-1 Aktor memilih untuk melakukan edit data profil
1. Sistem menampilkan data profil
  2. Aktor mengedit data profil yang dipilih
  3. Aktor meminta sistem untuk menyimpan data profil yang telah di edit
  4. Sistem melakukan pengecekan terhadap data profil yang telah di edit
    - E-2 Data profil yang telah di edit salah
  5. Sistem mengupdate data profil di database
  6. Berlanjut ke Basic Flow langkah ke 8
- A-2 Aktor memilih untuk melakukan delete data profil
1. Sistem menampilkan data profil dari database
  2. Aktor memilih data profil yang akan di delete
  3. Sistem menghapus data profil yang dipilih aktor
  4. Sistem mengupdate data profil di database
  5. Berlanjut ke Basic Flow langkah ke 8
- A-3 Aktor memilih untuk melakukan display data profil
1. Sistem menampilkan data profil dari database
  2. Berlanjut ke Basic Flow langkah ke 8

## 6. Error Flow

E-1 Data profil yang diinputkan aktor salah

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

E-2 Data profil yang telah di edit salah

1. Sistem memberikan peringatan bahwa data profil yang telah di edit salah
2. Kembali ke Alternative Flow 1 langkah ke 2

## 7. PreConditions

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

## 8. PostConditions

1. Data profil terupdate

### 4.1.7 Use case Spesification : Mengelola Kumpulan

#### 1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola kumpulan. Aktor data melakukan entry data kumpulan, edit data kumpulan, delete data kumpulan dan display data kumpulan

#### 2. Primary Actor

1. Admin

#### 3. Supporting Actor

none

#### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengelola kumpulan
2. Sistem memberikan pilihan untuk melakukan entry data kumpulan, edit data kumpulan, delete data kumpulan, display data kumpulan
3. Aktor memilih untuk melakukan entry data kumpulan  
A-1 Aktor memilih untuk edit data kumpulan

Program Studi Teknik Informatika	SKPL – PWGCI	29/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- A-2 Aktor memilih untuk delete data kumpulan
- A-3 Aktor memilih untuk display data kumpulan
- 4. Aktor menginputkan data kumpulan
- 5. Aktor meminta sistem untuk menyimpan data kumpulan
- 6. Sistem mengecek data kumpulan yang sudah diinputkan
- E-1 Data kumpulan yang diinputkan aktor salah
- 7. Sistem menyimpan data kumpulan ke database
- 8. Use case selesai

#### 5. Alternative Flow

- A-1 Aktor memilih untuk melakukan edit data kumpulan
  - 1. Sistem menampilkan data kumpulan
  - 2. Aktor mengedit data kumpulan yang dipilih
  - 3. Aktor meminta sistem untuk menyimpan data kumpulan yang telah di edit
  - 4. Sistem melakukan pengecekan terhadap data kumpulan yang telah di edit
  - E-2 Data kumpulan yang telah di edit salah
  - 5. Sistem mengupdate data kumpulan di database
  - 6. Berlanjut ke Basic Flow langkah ke 8
- A-2 Aktor memilih untuk melakukan delete data kumpulan
  - 1. Sistem menampilkan data kumpulan dari database
  - 2. Aktor memilih data kumpulan yang akan didelete
  - 3. Sistem menghapus data kumpulan yang dipilih aktor
  - 4. Sistem mengupdate data kumpulan di database
  - 5. Berlanjut ke Basic Flow langkah ke 8
- A-3 Aktor memilih untuk melakukan display data kumpulan
  - 1. Sistem menampilkan data kumpulan dari database
  - 2. Berlanjut ke Basic Flow langkah ke 8

#### 6. Error Flow

Program Studi Teknik Informatika	SKPL – PWGCI	30/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- E-1 Data kumpulan yang diinputkan aktor salah
1. Sistem memberikan peringatan bahwa data kumpulan yang diinputkan salah
  2. Kembali ke Basic Flow langkah ke 4
- E-2 Data kumpulan yang telah di edit salah
1. Sistem memberikan peringatan bahwa data kumpulan yang telah di edit salah
  2. Kembali ke Alternative Flow 1 langkah ke 2

#### **7. PreConditions**

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

#### **8. PostConditions**

1. Data kumpulan terupdate

### **4.1.8 Use case Spesification : Mengelola lingkungan**

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

Use Case ini digunakan oleh aktor untuk mengelola lingkungan. Aktor data melakukan entry data lingkungan, edit data lingkungan, delete data lingkungan dan display data lingkungan

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

1. Admin

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk mengelola lingkungan
  2. Sistem memberikan pilihan untuk melakukan entry data lingkungan, edit data lingkungan, delete data lingkungan, display data lingkungan
  3. Aktor memilih untuk melakukan entry data lingkungan
- A-1 Aktor memilih untuk edit data lingkungan

Program Studi Teknik Informatika	SKPL – PWGCI	31/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- A-2 Aktor memilih untuk delete data lingkungan
- A-3 Aktor memilih untuk display data lingkungan
- 4. Aktor menginputkan data lingkungan
- 5. Aktor meminta sistem untuk menyimpan data lingkungan
- 6. Sistem mengecek data lingkungan yang sudah diinputkan
- E-1 Data lingkungan yang diinputkan aktor salah
- 7. Sistem menyimpan data lingkungan ke database
- 8. Use case selesai

#### 5. Alternative Flow

- A-1 Aktor memilih untuk melakukan edit data lingkungan
  - 1. Sistem menampilkan data lingkungan
  - 2. Aktor mengedit data lingkungan yang dipilih
  - 3. Aktor meminta sistem untuk menyimpan data lingkungan yang telah di edit
  - 4. Sistem melakukan pengecekan terhadap data lingkungan yang telah di edit
  - E-2 Data lingkungan yang telah di edit salah
  - 5. Sistem mengupdate data lingkungan di database
  - 6. Berlanjut ke Basic Flow langkah ke 8
- A-2 Aktor memilih untuk melakukan delete data lingkungan
  - 1. Sistem menampilkan data lingkungan dari database
  - 2. Aktor memilih data lingkungan yang akan didelete
  - 3. Sistem menghapus data lingkungan yang dipilih aktor
  - 4. Sistem mengupdate data lingkungan di database
  - 5. Berlanjut ke Basic Flow langkah ke 8
- A-3 Aktor memilih untuk melakukan display data lingkungan

1. Sistem menampilkan data lingkungan dari database
2. Berlanjut ke Basic Flow langkah ke 8

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

E-1 Data lingkungan yang diinputkan aktor salah

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

E-2 Data lingkungan yang telah di edit salah

1. Sistem memberikan peringatan bahwa data lingkungan yang telah di edit salah
2. Kembali ke Alternative Flow 1 langkah ke 2

#### **7. PreConditions**

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

#### **8. PostConditions**

1. Data lingkungan terupdate

### **4.1.9 Use case Spesification : Mengelola Surat Pembaca**

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

Use Case ini memungkinkan aktor untuk mengelola surat pembaca.

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

1. Administrator

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data surat pembaca
2. Sistem memberikan pilihan untuk melakukan entry data surat pembaca atau display data surat pembaca.

Program Studi Teknik Informatika	SKPL – PWGCI	33/ 35
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Aktor memilih untuk melakukan entry data surat pembaca

A-1 Aktor Memilih untuk melakukan display data surat pembaca

4. Aktor menginputkan data surat pembaca

5. Aktor meminta sistem untuk menyimpan data surat pembaca yang telah diinputkan

6. Sistem mengecek data surat pembaca yang telah diinputkan

E-1 Data surat pembaca yang diinputkan aktor salah

7. Sistem menyimpan data surat pembaca ke database

8. Use Case selesai

#### **7. Alternative Flow**

A-1 Aktor memilih untuk melakukan display data surat pembaca

1. Sistem menampilkan data surat pembaca

2. Berlanjut ke Basic Flow langkah ke 8

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

E-1 Data surat pembaca yang diinputkan aktor salah

1. Sistem memberikan peringatan bahwa data surat pembaca 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 surat pembaca telah terupdate

## 5 Entity Relationship Diagram (ERD)

