

## **BAB VI**

### **KESIMPULAN & SARAN**

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

Dari pembahasan yang didapat dari bab-bab sebelumnya dapat dan pengujian langsung kepada pengguna dapat ditarik kesimpulan sebagai berikut:

1. Sistem Informasi Umat Gereja (SIMAG) dapat membantu proses pendataan umat di Paroki Santo Petrus dan Paulus Babadan
2. SIMAG dapat menangani keamanan sistem dengan menggunakan SMS Gateway.
3. SIMAG dapat membantu proses pencarian untuk keperluan penelitian dan pengembangan umat gereja di Paroki Santo Petrus dan Paulus Babadan.

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

Saran yang dapat diambil dari proses analisis sampai proses pembuatan SIMAG yaitu:

1. SIMAG diharapkan dapat dikembangkan sehingga dapat melakukan pengelolaan berita dan agenda dari gereja sehingga berita dan agenda gereja dapat diakses oleh umat.
2. SIMAG diharapkan dapat dikembangkan sehingga dapat ditambahkan fitur berupa map sehingga persebaran penduduknya dapat diketahui.

## DAFTAR PUSTAKA

- Amsyah, Zulkifli. 2003. Manajemen Sistem Informasi. Jakarta: PT. Gramedia Pustaka Utama.
- Jogiyanto, Hartono. 2001. Pengenalan Komputer. Yogyakarta: Andi Offset.
- Kristianto, Andi. 2003. *Perancangan Sistem Informasi dan Aplikasinya*. Yogyakarta: Gaya Media.
- Kusuma, Arif Hendra. 2013. Perancangan Sistem Informasi Pelayanan Pemeriksaan Kesehatan Berbasis Web dengan Sistem Pendaftaran Bergerak, *Transient*, (online), Vol.2, No.1, (<http://ejournal-s1.undip.ac.id/index.php/transient> diakses 30 Januari 2014)
- Laura, Lemay, 2001. Desain Grafik dan Halaman Web. Bandung: Informatika.
- Newcomer, Eric.2008. *Understanding Web Services XML, WSDL, SOAP, and UDDI*. USA:Independent Technology Guides.
- Powell, Gavin. 2006. Beginning Database Design. IEEE computer Society. Indiana. Vol 4, pp. 580-584.
- Rizky, Dhanta. 2009. Kamus Istilah Komputer Grafis dan Internet. Surabaya: Indah.
- Romadlan, dkk. 2013. Rancang Bangun Sistem Informasi Umat Gereja Katolik Gembala Yang Baik Surabaya Berbasis Web, *Jurnal JSIKA*, Vol.2, No. 2.
- Saputra, Agus. 2011. Step by Step Membangun Aplikasi SMS dengan php dan MySQL. Jakarta: PT Elex Media Komputindo.
- Soeprpto, Jeffrey. 2011. Pengembangan Sistem Informasi Manajemen Gereja Dengan *Disconnected Database* Gereja Melalui *Web Service*, Tugas Akhir, Jurusan Teknik Informatika, Universitas Atma Jaya Yogyakarta.

- Sutopo, Ariesto Hadi. 2002. Analisis dan Desain Berorientasi Objek. Yogyakarta: J&J Learning.
- Wahid, Fathul. 2002. Kamus Istilah Teknologi Informasi. Yogyakarta: Penerbit Andi.
- Wijayanti, Elisabeth Kurnia. 2011. Pembangunan Sistem Informasi Untuk Pengelolaan Data Warga Gereja dengan Menggunakan *.Net Framework*, Tugas Akhir, Jurusan Teknik Informatika, Universitas Atma Jaya Yogyakarta.
- Wilieyam, Gisela Nina Sevani. 2013. Aplikasi *Reminder* Pengobatan Pasien Berbasis SMS Gateway, Jurnal Informasi, (Online), Vol.7, No.1, (<http://ejournal.undip.ac.id/> diakses 30 Januari 2014).



# SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

SIMAG

(Sistem Informasi Umat Gereja Berbasis WEB)

Untuk :

Univesitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Lucia Desta Tri Utami / 100706160

Program Studi Teknik Informatika  
Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika  Fakultas Teknoloqi	Nomor Dokumen		Halaman
		<i>SKPL-SIMAG</i>		1/33
		Revisi		

**DAFTAR PERUBAHAN**

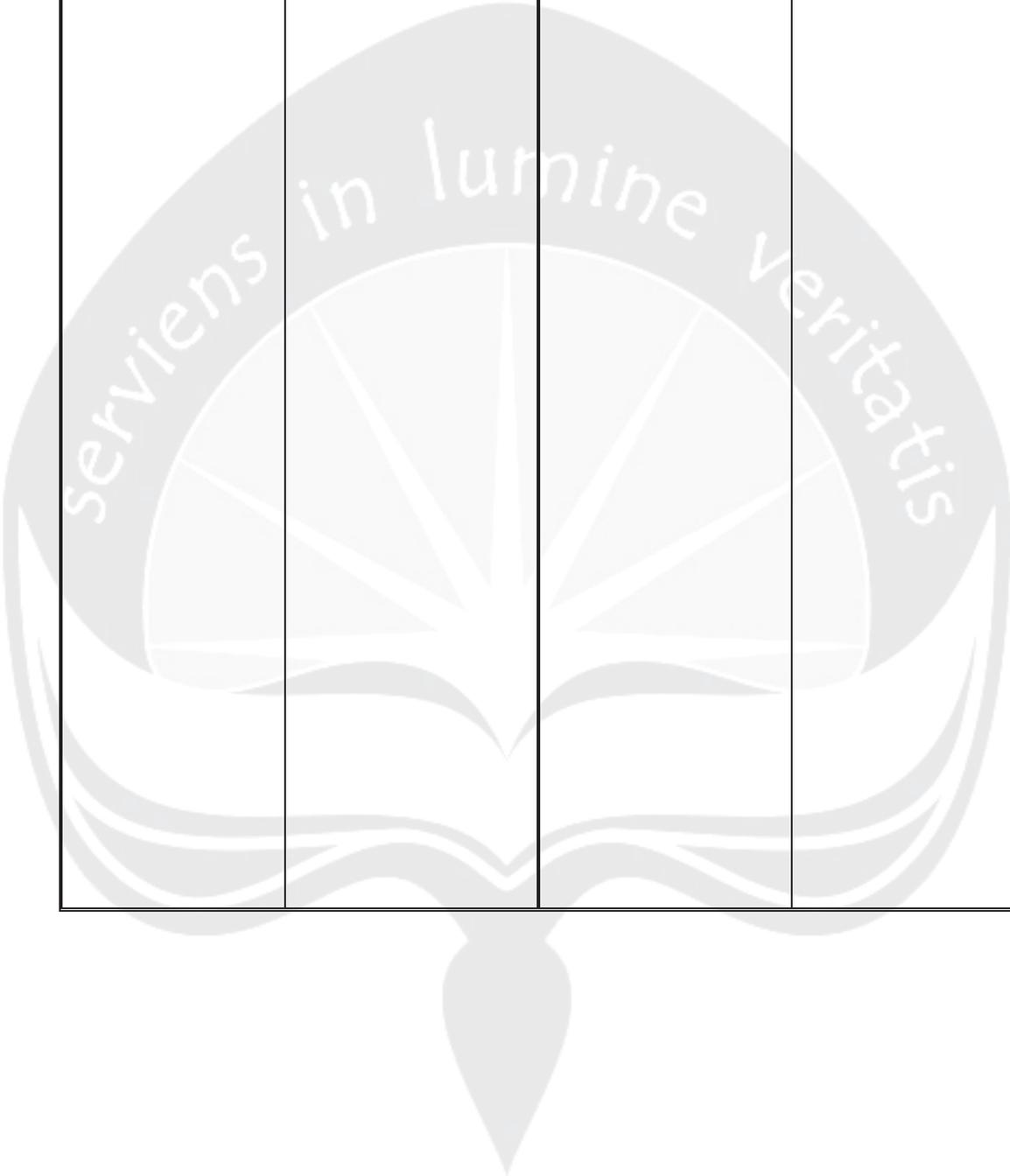
Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

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

**Daftar Halaman Perubahan**

Program Studi Teknik Informatika	SKPL – SIMAG	2/ 33
<p align="center">Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika</p>		

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
1.4	Referensi.....	7
1.5	Deskripsi umum (Overview).....	8
2	Deskripsi Kebutuhan.....	9
2.1	Perspektif produk.....	9
2.2	Fungsi Produk.....	10
2.3	Karakteristik Pengguna.....	13
2.4	Asumsi dan Ketergantungan.....	16
3	Kebutuhan khusus.....	16
3.1	Kebutuhan antarmuka eksternal.....	16
3.2	Kebutuhan fungsionalitas Perangkat Lunak.....	18
4	Spesifikasi Rinci Kebutuhan.....	18
4.1	Spesifikasi Kebutuhan Fungsional.....	18
5	Entity Relationship Diagram (ERD).....	33

## Daftar Gambar

Gambar 1. Arsitektur Perangkat lunak SIMAG.....	10
Gambar 2. Use Case Diagram.....	18
Gambar 3. Entity Relationship Diagram.....	33



## **1 Pendahuluan**

### **1.1 Tujuan**

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SIMAG yaitu Sistem Informasi Pengelolaan Umat Gereja Berbasis WEB untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal, atribut, dan fungsionalitas perangkat lunak. Secara lebih spesifik kebutuhan antar muka eksternal mencakup antarmuka antara sistem dengan sistem lain, perangkat lunak dan perangkat keras, dan pengguna. Sedangkan kebutuhan yang berkenaan dengan atribut menjelaskan fitur-fitur atau layanan-layanan yang disediakan pada sistem. Di dalam SKPL ini didefinisikan pula batasan perancangan perangkat lunak.

### **1.2 Lingkup Masalah**

SIMAG dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan wilayah.
2. Menangani pengelolaan Stasi
3. Menangani pengelolaan lingkungan
4. Menangani pengelolaan kepala keluarga
5. Menangani pengelolaan umat
6. Menangani pencarian
7. Menangani laporan
8. Menangani pencarian umat
9. Menangani keamanan sistem

SIMAG dikembangkan sebagai sebuah aplikasi web, sehingga dapat berjalan pada semua platform sistem operasi yang dilengkapi dengan web browser.

Program Studi Teknik Informatika	SKPL – SIMAG	6/ 33
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-SIMAG-XXX	Kode yang merepresentasikan kebutuhan pada WEB Sistem Informasi Pengelolaan umat dimana XXX merupakan nomor fungsi produk.
SIMAG	Merupakan kependekan dari Sistem Informasi Pengelolaan umat gereja yang merupakan Perangkat lunak berbasis web untuk menangani gereja .
Berbasis web (Web Based)	Menyatakan aplikasi yang dalam penggunaannya pada komputer client dapat diakses mempergunakan web browser tanpa memerlukan proses instalasi terlebih dahulu.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

### 1.4 Referensi

Referensi yang digunakan pada penyusunan dokumen SKPL ini adalah

1. Lucia Desta Tri Utami, *Spesifikasi Kebutuhan Perangkat Lunak CFLSS*, Universitas Atma Jaya Yogyakarta, 2013.

### 1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas lima bagian utama. Kelima bagian tersebut dijelaskan dalam uraian di bawah ini.

Bagian pertama berisi penjelasan mengenai dokumen SKPL ini. Di dalamnya dijelaskan 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 SIMAG yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak SIMAG tersebut.

Bagian ketiga berisi penjelasan tentang kebutuhan khusus perangkat lunak, di dalamnya dideksripsikan kebutuhan antarmuka eksternal dilanjutkan dengan deskripsi kebutuhan fungsionalitas perangkat lunak.

Bagian keempat berisi tentang penjelasan rinci tentang kebutuhan perangkat lunak SIMAG yang akan dikembangkan.

Pada bagian terakhir berisi tentang Entity Relationship Diagram yang menjadi dasar pengembangan perangkat lunak SIMAG.

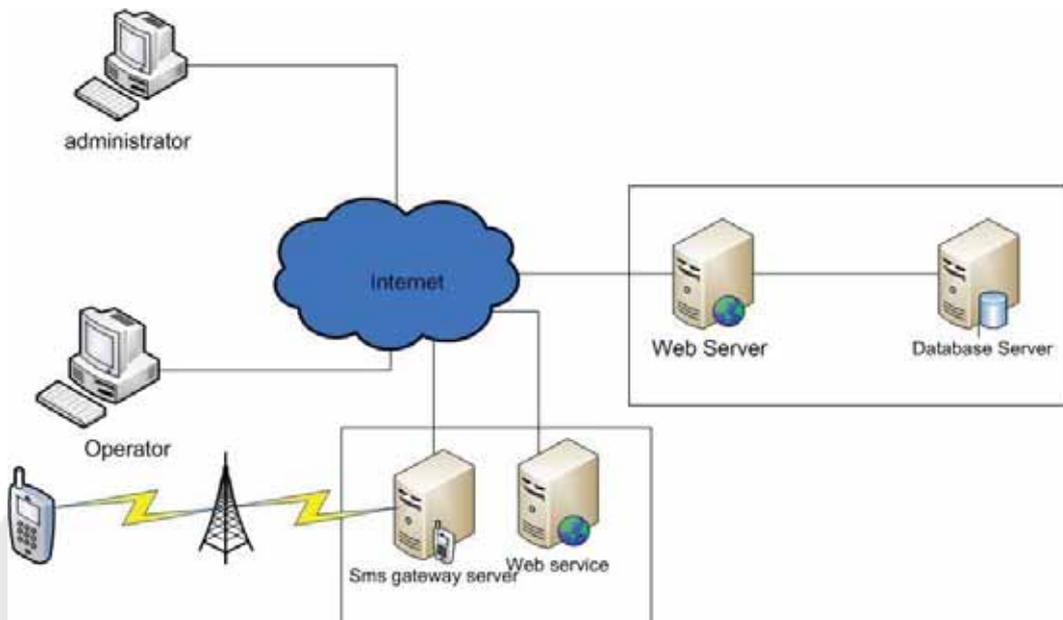
## 2 Deskripsi Kebutuhan

### 2.1 Perspektif produk

Perangkat lunak SIMAG dikembangkan untuk membantu pengelolaan dokumen mengenai pengelolaan umat gereja yang terdiri pengelolaan wilayah, pengelolaan stasi, pengelolaan lingkungan, pengelolaan kepala keluarga, dan pengelolaan umat. Selain itu juga akan menangani pencarian data, laporan, dan keamanan sistem.

Perangkat lunak SIMAG merupakan aplikasi berbasis web yang dikembangkan dengan menggunakan bahasa pemrograman C#. Sedangkan untuk lingkungan pemrogramannya menggunakan Microsoft Visual Studio 2010.

SIMAG dapat berjalan di berbagai platform sistem operasi yang dilengkapi dengan browser. Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (*Graphical User Interface*) yang disediakan. Arsitektur yang digunakan, seperti terlihat pada gambar 1. Dimana semua data disimpan di server. User dapat mengakses data yang ada di server tersebut secara on-line dengan menggunakan web browser untuk kemudian memanggil layanan yang disediakan melalui Web Server.



Gambar 1. Arsitektur Perangkat lunak SIMAG

## 2.2 Fungsi Produk

Fungsi produk perangkat lunak SIMAG adalah sebagai berikut :

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

Merupakan fungsi yang digunakan oleh user untuk masuk dalam sistem. Hak akses untuk masing-masing user ditentukan oleh role yang ada padanya. Sistem dapat menentukan role dari setiap user yang login.

### 2. Fungsi Pengelolaan Wilayah ((**SKPL-SIMAG-002**)).

Merupakan fungsi yang digunakan actor untuk mengelola wilayah yang ada.

Fungsi pengelolaan wilayah meliputi:

#### a. Fungsi *Tambah wilayah Baru* (**SKPL-SIMAG-002-01**).

Merupakan fungsi yang digunakan oleh aktor untuk menambahkan wilayah baru

#### b. Fungsi *Hapus Wilayah* (**SKPL-SIMAG-002-02**).

Merupakan fungsi yang digunakan aktor untuk menghapus data wilayah terpilih.

c. Fungsi Ubah Wilayah (**SKPL-SIMAG-002-03**).

Merupakan fungsi yang digunakan untuk mengubah data wilayah yang terpilih.

d. Fungsi *Tampil Data* (**SKPL-SIMAG-002-04**).

Merupakan fungsi yang digunakan untuk menampilkan seluruh Wilayah yang ada.

e. Fungsi *Cari Data Wilayah* (**SKPL-SIMAG-003-05**).

Merupakan fungsi yang digunakan untuk menampilkan seluruh wilayah yang ada.

3. Fungsi *Pengelolaan Stasi* (**SKPL-SIMAG-003**).

Merupakan fungsi yang digunakan oleh aktor untuk mengelola stasi yang ada.

Fungsi pengelolaan Stasi meliputi:

a. Fungsi *Tambah Stasi Baru* (**SKPL-SIMAG-003-01**).

Merupakan fungsi yang digunakan oleh aktor untuk menambahkan stasi baru

b. Fungsi *Hapus Stasi* (**SKPL-SIMAG-003-02**).

Merupakan fungsi yang digunakan aktor untuk menghapus data stasi terpilih.

c. Fungsi *Ubah Stasi* (**SKPL-SIMAG-003-03**).

Merupakan fungsi yang digunakan untuk mengubah data stasi yang terpilih.

d. Fungsi *Tampil Data* (**SKPL-SIMAG-003-04**).

Merupakan fungsi yang digunakan untuk menampilkan seluruh Stasi yang ada.

Program Studi Teknik Informatika	SKPL – SIMAG	11/ 33
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. Fungsi Cari Data Paroki (**SKPL-SIMAG-003-05**).

Merupakan fungsi yang digunakan untuk menampilkan seluruh Stasi yang ada.

4. Fungsi Pengelolaan Lingkungan (**SKPL-SIMAG-004**).

Merupakan fungsi yang digunakan oleh aktor untuk mengelola lingkungan yang ada.

Fungsi pengelolaan lingkungan meliputi:

a. Fungsi Tambah Lingkungan Baru (**SKPL-SIMAG-004-01**).

Merupakan fungsi yang digunakan oleh aktor untuk menambahkan lingkungan baru

b. Fungsi Hapus Lingkungan (**SKPL-SIMAG-004-02**).

Merupakan fungsi yang digunakan aktor untuk menghapus data lingkungan terpilih.

c. Fungsi Ubah Lingkungan (**SKPL-SIMAG-004-03**).

Merupakan fungsi yang digunakan untuk mengubah data lingkungan yang terpilih.

d. Fungsi Tampil Data (**SKPL-SIMAG-004-04**).

Merupakan fungsi yang digunakan untuk menampilkan seluruh lingkungan yang ada.

e. Fungsi Cari Data Lingkungan (**SKPL-SIMAG-004-05**).

Merupakan fungsi yang digunakan untuk mencari data lingkungan yang ada.

5. Fungsi Pengelolaan Kepala Keluarga (**SKPL-SIMAG-005**).

Program Studi Teknik Informatika	SKPL – SIMAG	12/ 33
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan oleh aktor untuk melakukan pengelolaan kepala keluarga.

Fungsi laporan meliputi :

a. *Fungsi Tambah Kepala Keluarga Baru (SKPL-SIMAG-005-01).*

Merupakan fungsi yang digunakan oleh aktor untuk menambahkan kepala keluarga baru

b. *Fungsi Hapus Kepala Keluarga (SKPL-SIMAG-005-02).*

Merupakan fungsi yang digunakan aktor untuk menghapus data kepala keluarga terpilih.

c. *Fungsi Ubah Kepala Keluarga (SKPL-SIMAG-005-03).*

Merupakan fungsi yang digunakan untuk mengubah data kepala keluarga yang terpilih.

d. *Fungsi Tampil Data (SKPL-SIMAG-005-04).*

Merupakan fungsi yang digunakan untuk menampilkan seluruh kepala keluarga yang ada.

e. *Fungsi Cari Data Kepala Keluarga (SKPL-SIMAG-005-05).*

Merupakan fungsi yang digunakan untuk mencari data kepala keluarga.

6. *Fungsi Pengelolaan Umat (SKPL-SIMAG-006).*

Merupakan fungsi yang digunakan untuk mengelola umat gereja.

Fungsi pengelolaan umat meliputi :

a. *Fungsi Tambah Umat Baru (SKPL-SIMAG-006-01).*

Program Studi Teknik Informatika	SKPL – SIMAG	13/ 33
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan oleh aktor untuk menambahkan umat baru

b. Fungsi Hapus umat (**SKPL-SIMAG-006-02**).

Merupakan fungsi yang digunakan aktor untuk menghapus data umat terpilih.

c. Fungsi Ubah Umat (**SKPL-SIMAG-006-03**).

Merupakan fungsi yang digunakan untuk mengubah data umat yang terpilih.

d. Fungsi Tampil Data (**SKPL-SIMAG-006-04**).

Merupakan fungsi yang digunakan untuk menampilkan seluruh umat yang ada.

e. Fungsi Cari Data Umat (**SKPL-SIMAG-006-05**).

Merupakan fungsi yang digunakan untuk menampilkan seluruh umat yang ada

7. Fungsi Pengelolaan Pengguna(**SKPL-SIMAG-007**)

a. Fungsi Tambah Pengguna(**SKPL-SIMAG-007-01**).

Merupakan fungsi yang digunakan untuk menambah data pengguna.

b. Fungsi Ubah Pengguna(**SKPL-SIMAG-007-02**).

Merupakan fungsi yang digunakan untuk mengubah data pengguna.

c. Fungsi Hapus Pengguna(**SKPL-SIMAG-007-03**).

Merupakan fungsi untuk menghapus data pengguna yang terpilih.

d. Fungsi Tampil Data Pengguna(**SKPL-SIMAG-007-04**).

Merupakan fungsi untuk menghapus data pengguna

e. Fungsi Cari Data pengguna(**SKPL-SIMAG-007-05**).

Merupakan fungsi yang digunakan untuk menampilkan seluruh pengguna yang ada

8. Fungsi Ubah Password(**SKPL-SIMAG-008**).

Program Studi Teknik Informatika	SKPL – SIMAG	14/ 33
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan oleh aktor untuk mengubah password.

9. Fungsi Pencarian (**SKPL-SIMAG-009**).

Merupakan fungsi yang digunakan oleh aktor untuk mencari data umat atau data kepala keluarga.

10. Fungsi kirim SMS(**SKPL-SIMAG-010**)

Merupakan fungsi yang digunakan untuk mengirim SMS ke no handphone pengguna.

11. Fungsi input kode(**SKPL-SIMAG-011**)

Merupakan fungsi yang digunakan untuk memverifikasi kode/pin yang telah dikirim melalui SMS.

12. Fungsi cetak laporan (**SKPL-SIMAG-012**)

a. Cetak kartu kepala keluarga(**SKPL-SIMAG-012-01**)

Merupakan fungsi yang digunakan untuk mencetak kartu kepala keluarga.

b. Cetak laporan jumlah umat berdasarkan pekerjaan(**SKPL-SIMAG-012-02**)

Merupakan fungsi yang digunakan untuk mencetak laporan jumlah umat berdasarkan pekerjaan.

c. Cetak laporan jumlah umat tiap lingkungan(**SKPL-SIMAG-012-03**)

Merupakan fungsi yang digunakan untuk mencetak laporan jumlah umat tiap lingkungan.

### 2.3 Karakteristik Pengguna

Karena aplikasi dikembangkan seragam sebagai aplikasi web, maka karakteristik untuk masing-masing role pengguna adalah sama. Karakteristik pengguna yang dimaksud diuraikan sebagai berikut:

Program Studi Teknik Informatika	SKPL – SIMAG	15/ 33
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. Mengerti pengoperasian web browser.
2. Dapat mempergunakan internet.
3. Memahami pengoperasian perangkat lunak SIMAG.

#### **2.4 Asumsi dan Ketergantungan**

Perangkat lunak SIMAG dikembangkan sebagai aplikasi berbasis web dengan asumsi dan ketergantungan sebagai berikut:

1. Dapat dijalankan pada perangkat komputer non mobile, seperti PC desktop, laptop, maupun netbook yang terhubung dengan internet dan dilengkapi dengan Web Browser.
2. Untuk kenyamanan tampilan, web browser yang direkomendasikan adalah Mozilla Firefox versi 3.5.0 atau lebih tinggi, Google Chrome versi 4.0 atau lebih tinggi.

### **3 Kebutuhan khusus**

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

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

##### **3.1.1 Antarmuka pengguna**

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

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

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SIMAG adalah:

1. Mouse, monitor, keyboard dan cpu.

Program Studi Teknik Informatika	SKPL – SIMAG	16/ 33
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. Infrastruktur jaringan dan internet.

### 3.1.3 Antarmuka perangkat lunak

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

1. Nama : SQL Server 2008  
Sumber : Microsoft  
Sebagai database management sistem (DBMS) yang digunakan untuk penyimpan data di sisi server.
2. Nama : Mozilla Firefox versi 3.5.0 ke atas  
Sumber : Mozilla Foundation  
Sebagai web browser.
3. Nama : Google Chrome versi 4.0 ke atas  
Sumber : Google Incorporation  
Sebagai web browser.
4. Nama : .Net Framework versi 4.0 ke atas  
Sumber : Microsoft Corporation  
sebagai platform, yang merupakan suatu lingkungan terpadu untuk pengembangan dan eksekusi untuk berbagai macam bahasa pemrograman dan kumpulan library untuk bekerja sama membuat dan menjalankan aplikasi berbasis Windows
5. Nama : Web Server IIS  
Sumber : Microsoft Corporation  
Sebagai web server tempat aplikasi SIMAG dijalankan
6. Nama : Zenziva  
Sumber : -  
Sebagai penyedia layanan *SMS Gateway*

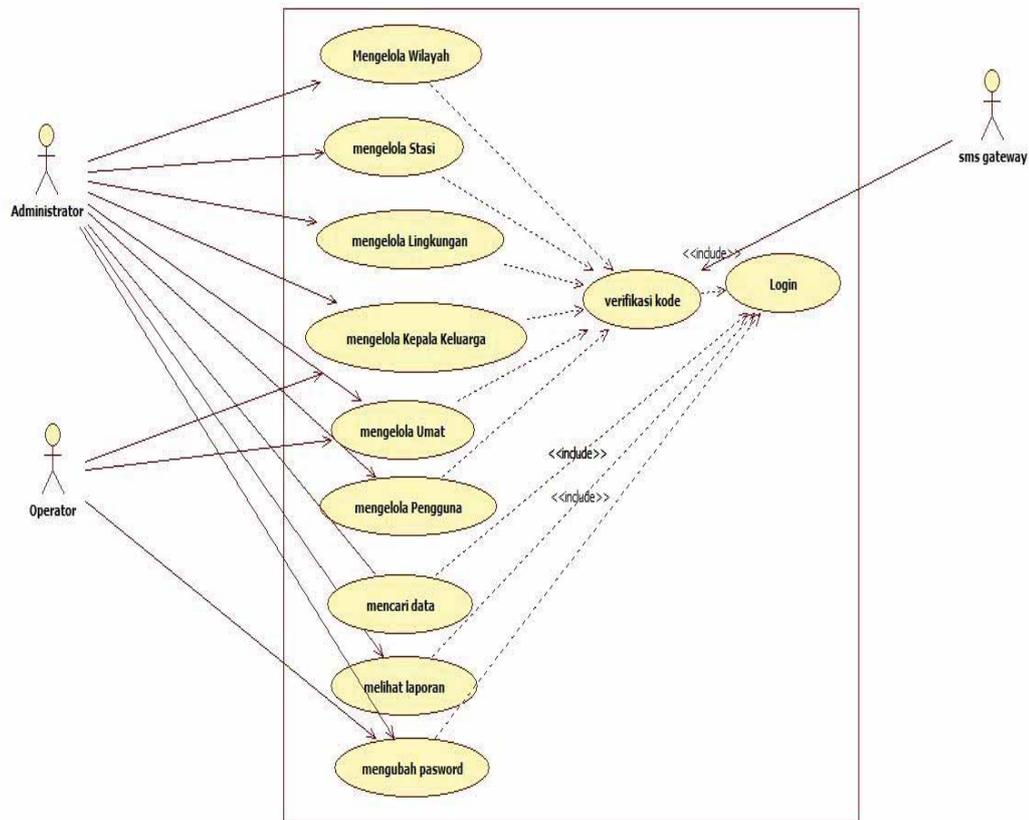
### 3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak SIMAG menggunakan protocol TCP/IP.

Program Studi Teknik Informatika	SKPL – SIMAG	17/ 33
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		

### 3.2 Kebutuhan fungsionalitas Perangkat Lunak

#### 3.2.1 Use Case Diagram



Gambar 2. Use Case Diagram

## 4 Spesifikasi Rinci Kebutuhan

### 4.1 Spesifikasi Kebutuhan Fungsional

#### 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 username dari user dan password yang berupa rangkaian karakter.

##### 2. Primary Aktor

1. User / Petugas

Program Studi Teknik Informatika	SKPL – SIMAG	18/ 33
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. Supporting Aktor

none

### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login
2. Sistem menampilkan antarmuka untuk login
3. Aktor memasukkan username
4. Sistem memeriksa username  
E-1 Username tidak ada
5. Sistem memberikan password melalui sms gateway
6. Aktor memasukkan password
7. Sistem memeriksa password yang diinputkan aktor  
E-2 Password user tidak sesuai
8. Sistem memberikan akses ke aktor
9. Use Case ini selesai

### 5. Alternative Flow

none

### 6. Error Flow

E-1 Username tidak ada.

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

E-2 Password tidak sesuai

1. Sistem menampilkan peringatan password tidak sesuai.
2. Kembali ke Basic Flow langkah ke 6.

### 2. PreConditions

none

### 3. PostConditions

Program Studi Teknik Informatika	SKPL – SIMAG	19/ 33
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem sesuai dengan role yang dimiliki user yang bersangkutan.

#### **4.1.2 Use case Spesification : Pengelolaan Wilayah**

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

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan wilayah.

##### **2. Primary Aktor**

1. User / Petugas

##### **3. Supporting Aktor**

none

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

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan wilayah
2. Sistem menampilkan form pengelolaan wilayah
3. Aktor memilih untuk menambah data wilayah baru
4. Sistem menampilkan form untuk menambah wilayah baru
5. Aktor menginputkan data wilayah
6. Sistem menyimpan data wilayah baru ke dalam database
7. Use Case selesai

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

Setelah basic flow langkah ke 2

A-1 Aktor memilih untuk melakukan ubah data wilayah

1. Sistem menampilkan data wilayah
2. Aktor memilih mengubah data wilayah yang terpilih
3. Sistem menyimpan perubahan data wilayah yang terpilih ke basis data

Program Studi Teknik Informatika	SKPL – SIMAG	20/ 33
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. Use case selesai

Setelah basic flow langkah ke 2

A-2 Aktor memilih untuk menghapus data wilayah

1. Sistem menampilkan data wilayah
2. Aktor memilih menghapus data wilayah yang terpilih
3. Sistem menyimpan perubahan data wilayah yang terpilih ke basis data
4. Use case selesai

Setelah basic flow langkah ke 2

A-3 Aktor memilih untuk mencari data wilayah

1. Sistem menampilkan data wilayah
2. Aktor memilih mencari data wilayah
3. Sistem menyimpan perubahan data wilayah yang terpilih ke basis data
4. Use case selesai

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

none

#### **7. PreConditions**

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem
3. Sistem berjalan dengan benar.

#### **8. PostConditions**

1. Data wilayah telah disimpan dalam database

### **4.1.3 Use case Spesification : Pengelolaan Stasi**

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

Use Case ini digunakan oleh user / petugas untuk melakukan pengelolaan stasi.

#### **2. Primary Aktor**

1. User / petugas

#### **3. Supporting Aktor**

Program Studi Teknik Informatika	SKPL – SIMAG	21/ 33
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		

none

#### 4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan stasi
2. Sistem menampilkan form pengelolaan stasi
3. Aktor memilih untuk menambah data stasi baru
4. Sistem menampilkan form untuk menambah stasi baru
5. Aktor menginputkan data stasi
6. Sistem menyimpan data stasi baru ke dalam database
7. Use Case selesai

#### 5. Alternative Flow

Setelah basic flow langkah ke 2

A-1 Aktor memilih untuk melakukan ubah data stasi

1. Sistem menampilkan data stasi
2. Aktor memilih mengubah data stasi yang terpilih
3. Sistem menyimpan perubahan data stasi yang terpilih ke basis data
4. Use case selesai

Setelah basic flow langkah ke 2

A-2 Aktor memilih untuk menghapus data stasi

1. Sistem menampilkan data stasi
2. Aktor memilih menghapus data stasi yang terpilih
3. Sistem menyimpan perubahan data stasi yang terpilih ke basis data
4. Use case selesai

Setelah basic flow langkah ke 2

A-3 Aktor memilih untuk mencari data stasi

1. Sistem menampilkan data stasi
2. Aktor memilih mencari data stasi

3. Sistem menyimpan perubahan data stasi yang terpilih ke basis data

4. Use case selesai

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

#### **7. PreConditions**

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem
3. Sistem berjalan dengan benar.

#### **8. PostConditions**

1. Data stasi telah disimpan dalam database

### **4.1.4 Use case Spesification : Pengelolaan Lingkungan**

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

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan lingkungan.

#### **2. Primary Aktor**

1. User / Petugas

#### **3. Supporting Aktor**

none

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

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan lingkungan
2. Sistem menampilkan form pengelolaan lingkungan
3. Aktor memilih untuk menambah data lingkungan baru
4. Sistem menampilkan form untuk menambah lingkungan baru
5. Aktor menginputkan data lingkungan
6. Sistem menyimpan data lingkungan baru ke dalam database
7. Use Case selesai

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

Program Studi Teknik Informatika	SKPL – SIMAG	23/ 33
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		

Setelah basic flow langkah ke 2

A-1 Aktor memilih untuk melakukan ubah data lingkungan

1. Sistem menampilkan data lingkungan
2. Aktor memilih mengubah data lingkungan yang terpilih
3. Sistem menyimpan perubahan data lingkungan yang terpilih ke basis data
4. Use case selesai

Setelah basic flow langkah ke 2

A-2 Aktor memilih untuk menghapus data lingkungan

1. Sistem menampilkan data stasi
2. Aktor memilih menghapus data lingkungan yang terpilih
3. Sistem menyimpan perubahan data lingkungan yang terpilih ke basis data
4. Use case selesai

Setelah basic flow langkah ke 2

A-3 Aktor memilih untuk mencari data lingkungan

1. Sistem menampilkan data lingkungan
2. Aktor memilih mencari data lingkungan
3. Sistem menyimpan perubahan data lingkungan yang terpilih ke basis data
4. Use case selesai

## **6. Error Flow**

none

## **7. PreConditions**

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem
3. Sistem berjalan dengan benar.

## **8. PostConditions**

Program Studi Teknik Informatika	SKPL – SIMAG	24/ 33
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. Data lingkungan sudah tersimpan ke database.

#### **4.1.5 Use case Spesification : Pengelolaan Kepala Keluarga**

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

Use case ini digunakan actor untuk melakukan pengelolaan kepala keluarga.

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

1. Aktor

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan kepala keluarga
2. Sistem menampilkan form pengelolaan kepala keluarga
3. Aktor memilih untuk menambah data kepala keluarga baru
4. Sistem menampilkan form untuk menambah kepala keluarga baru
5. Aktor menginputkan data kepala keluarga
6. Sistem menyimpan data kepala keluarga baru ke dalam database
7. Use Case selesai

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

Setelah basic flow langkah ke 2

A-1 Aktor memilih untuk melakukan ubah data kepala keluarga

1. Sistem menampilkan data kepala keluarga
2. Aktor memilih mengubah data kepala keluarga yang terpilih

Program Studi Teknik Informatika	SKPL – SIMAG	25/ 33
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. Sistem menyimpan perubahan data kepala keluarga yang terpilih ke basis data

4. Use case selesai

Setelah basic flow langkah ke 2

A-2 Aktor memilih untuk menghapus data kepala keluarga

1. Sistem menampilkan data kepala keluarga

2. Aktor memilih menghapus data kepala keluarga yang terpilih

3. Sistem menyimpan perubahan data kepala keluarga yang terpilih ke basis data

4. Use case selesai

Setelah basic flow langkah ke 2

A-3 Aktor memilih untuk mencari data kepala keluarga

1. Sistem menampilkan data kepala keluarga

2. Aktor memilih mencari data kepala keluarga

3. Sistem menyimpan perubahan data kepala keluarga yang terpilih ke basis data

4. Use case selesai

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

none

#### **7. PreConditions**

1. Use Case Login sudah dilakukan

2. Aktor telah memasuki sistem

3. Sistem berjalan dengan benar.

#### **8. PostConditions**

1. Data kepala keluarga sudah tersimpan didatabase.

### **4.1.6 Use case Spesification : Pengelolaan Umat**

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

Program Studi Teknik Informatika	SKPL – SIMAG	26/ 33
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Use case ini digunakan actor untuk melakukan pengelolaan umat.

## **2. Primary Actor**

1. Aktor

## **3. Supporting Actor**

none

## **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan umat
2. Sistem menampilkan form pengelolaan umat
3. Aktor memilih untuk menambah data umat baru
4. Sistem menampilkan form untuk menambah umat baru
5. Aktor menginputkan data umat
6. Sistem menyimpan data umat baru ke dalam database
7. Use Case selesai

## **5. Alternative Flow**

Setelah basic flow langkah ke 2

A-1 Aktor memilih untuk melakukan ubah data umat

1. Sistem menampilkan data umat
2. Aktor memilih mengubah data umat yang terpilih
3. Sistem menyimpan perubahan data umat yang terpilih ke basis data
4. Use case selesai

Setelah basic flow langkah ke 2

A-2 Aktor memilih untuk menghapus data umat

1. Sistem menampilkan data umat
2. Aktor memilih menghapus data umat yang terpilih
3. Sistem menyimpan perubahan data umat yang terpilih ke basis data

4. Use case selesai

Setelah basic flow langkah ke 2

A-3 Aktor memilih untuk mencari data umat

1. Sistem menampilkan data umat
2. Aktor memilih mencari data umat
3. Sistem menyimpan perubahan data umat yang terpilih ke basis data
4. Use case selesai

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

none

#### **7. PreConditions**

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem
3. Sistem berjalan dengan benar.

#### **9. PostConditions**

1. Data kepala umat sudah tersimpan didatabase.

### **4.1.7 Use case Spesification : Pengelolaan Pengguna**

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

Use case ini digunakan actor untuk melakukan pengelolaan pengguna.

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

1. Administrator

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan pengguna
2. Sistem menampilkan form pengelolaan pengguna
3. Aktor memilih untuk menambah data pengguna baru

4. Sistem menampilkan form untuk menambah pengguna baru
5. Aktor menginputkan data pengguna
6. Sistem menyimpan data umat baru ke dalam database
7. Use Case selesai

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

Setelah basic flow langkah ke 2

A-1 Aktor memilih untuk melakukan ubah data pengguna

1. Sistem menampilkan data pengguna
2. Aktor memilih mengubah data pengguna yang terpilih
3. Sistem menyimpan perubahan data pengguna yang terpilih ke basis data
4. Use case selesai

Setelah basic flow langkah ke 2

A-2 Aktor memilih untuk menghapus data pengguna

1. Sistem menampilkan data umat
2. Aktor memilih menghapus data umat yang terpilih
3. Sistem menyimpan perubahan data umat yang terpilih ke basis data
4. Use case selesai

Setelah basic flow langkah ke 2

A-3 Aktor memilih untuk mencari data umat

1. Sistem menampilkan data umat
2. Aktor memilih mencari data umat
3. Sistem menyimpan perubahan data umat yang terpilih ke basis data
4. Use case selesai

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

none

Program Studi Teknik Informatika	SKPL – SIMAG	29/ 33
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		

## **7. PreConditions**

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem
3. Sistem berjalan dengan benar.

## **10. PostConditions**

1. Data kepala keluarga sudah tersimpan didatabase.

### **4.1.8 Use case Spesification : Melihat Laporan**

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

Use case ini digunakan actor untuk melakukan melihat laporan.

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

1. Administrator

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

none

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

1. Use Case ini dimulai ketika aktor memilih untuk melihat laporan
2. Sistem menampilkan form pengelolaan laporan
3. Aktor memilih laporan yang ingin ditampilkan
4. Sistem menampilkan form untuk menampilkan laporan
5. Use Case selesai

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

None

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

none

## **7. PreConditions**

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem
3. Sistem berjalan dengan benar.

## **8. PostConditions**

1. Laporan berhasil ditampilkan.

Program Studi Teknik Informatika	SKPL – SIMAG	30/ 33
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

#### 4.1.9 Use case Spesification : Pencarian Data

##### 8. Brief Description

Use case ini digunakan actor untuk melakukan pencarian data umat.

##### 9. Primary Actor

1. Administrator

##### 10. Supporting Actor

none

##### 11. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pencarian data umat
2. Sistem menampilkan form pencarian data umat
3. Aktor memilih kriteria yang diinginkan dan memasukkan value
4. Aktor memilih mencari data umat
5. Sistem menampilkan data umat
6. Use Case selesai

##### 12. Alternative Flow

Setelah basic flow langkah ke 2

A-1 Aktor memilih untuk clear form

1. Sistem menampilkan form pencarian
2. Aktor memilih clear form
3. Form berhasil dikosongkan
4. Use case selesai

Setelah basic flow langkah ke 2

A-2 Aktor memilih untuk mencetak data

1. Sistem menampilkan form pencarian
2. Aktor memilih cetak data
3. Data berhasil dicetak
4. Use case selesai

##### 6. Error Flow

none

Program Studi Teknik Informatika	SKPL – SIMAG	31/ 33
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		

## **7. PreConditions**

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki sistem
3. Sistem berjalan dengan benar.

## **9. PostConditions**

1. Laporan berhasil ditampilkan.





# DPPL

## DESKRIPSI PERANCANGAN PERANGKAT LUNAK

### SIMAG

(Sistem Informasi Umat Gereja Berbasis Web)

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Lucia Desta Tri Utami / 100706160

Program Studi Teknik Informatika  
Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<b>DPPL-SIMAG</b>		1/64
		Revisi		

## 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	LDTU							
Diperiksa oleh	PM							
Disetujui oleh	PM							

### Daftar Halaman Perubahan

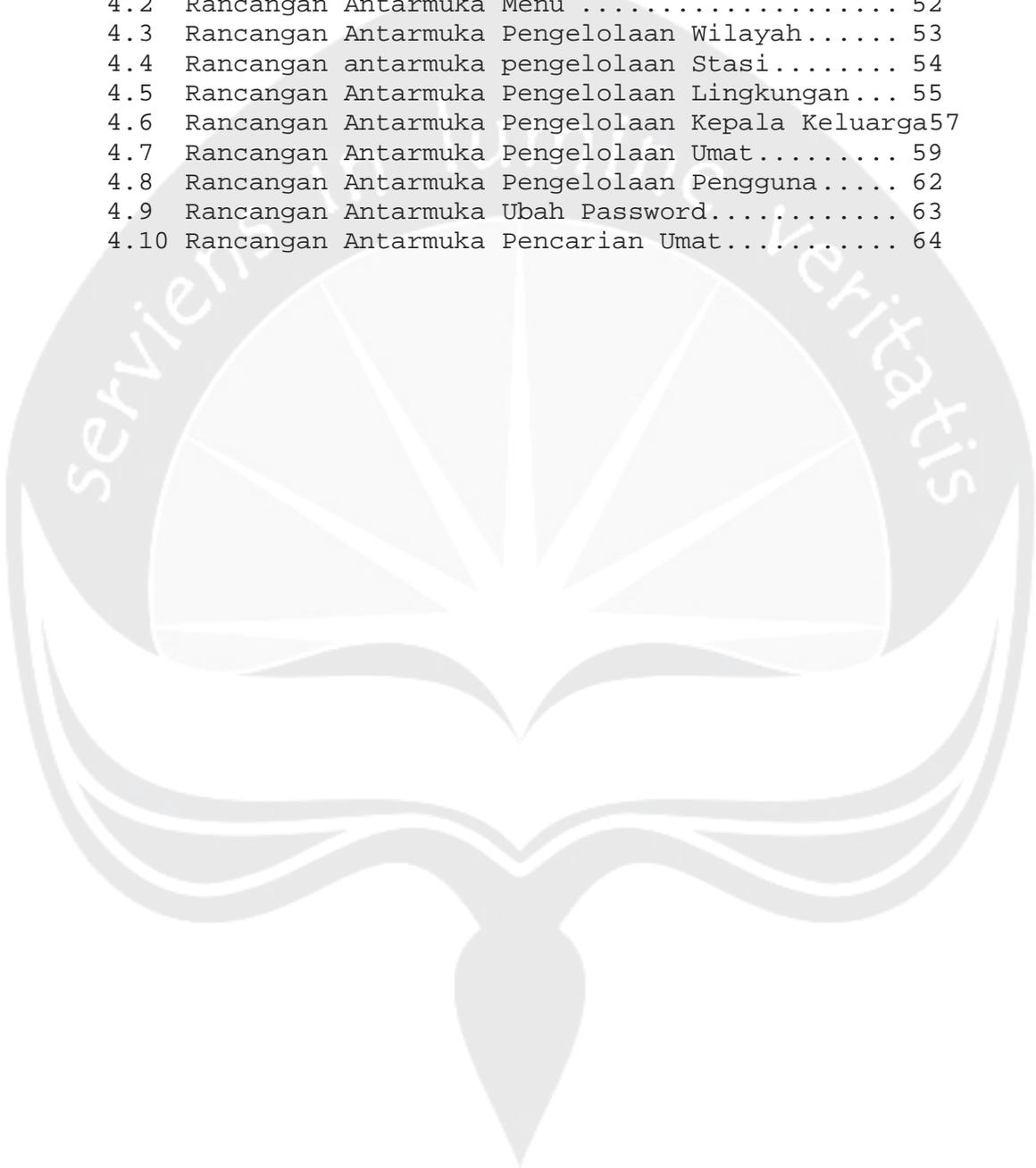
Halaman	Revisi	Halaman	Revisi

## Daftar Isi

1	Pendahuluan .....	7
1.1	<i>Tujuan</i> .....	7
1.2	<i>Ruang Lingkup</i> .....	7
1.3	<i>Definisi dan Akronim</i> .....	7
1.4	<i>Referensi</i> .....	8
2	Analysis Model.....	9
2.1	Perancangan Arsitekur .....	9
2.2	<i>Perancangan Rinci</i> .....	10
2.2.1	Sequence Diagram .....	10
2.2.1.1	Login .....	10
2.2.1.2	Pengelolaan Wilayah.....	11
2.2.1.2.1	Tambah Wilayah Baru.....	11
2.2.1.2.2	Ubah Wilayah.....	11
2.2.1.2.3	Hapus Wilayah.....	12
2.2.1.2.4	Tampil Wilayah.....	12
2.2.1.2.5	Cari Wilayah.....	13
2.2.1.3	Pengelolaan Stasi.....	13
2.2.1.3.1	Tambah Stasi Baru.....	13
2.2.1.3.2	Ubah Stasi.....	14
2.2.1.3.3	Hapus stasi.....	14
2.2.1.3.4	Tampil Stasi.....	15
2.2.1.3.5	Cari Stasi.....	15
2.2.1.4	Pengelolaan Lingkungan.....	16
2.2.1.4.1	Tambah lingkungan.....	16
2.2.1.4.2	Ubah Lingkungan.....	16
2.2.1.4.3	Hapus Lingkungan.....	17
2.2.1.4.4	Tampil Lingkungan.....	17
2.2.1.4.5	Cari Lingkungan.....	18
2.2.1.5	Pengelolaan Kepala Keluarga.....	19
2.2.1.5.1	Tambah kepala Keluarga.....	19
2.2.1.5.2	Ubah Kepala Keluarga.....	20
2.2.1.5.3	Hapus Kepala Keluarga.....	20
2.2.1.5.4	Tampil Kepala Keluarga.....	21
2.2.1.5.5	Cari kepala Keluarga.....	21
2.2.1.6	Pengelolaan Umat.....	22
2.2.1.6.1	Tambah Umat.....	22
2.2.1.6.2	Ubah Umat.....	22
2.2.1.6.3	Hapus Umat.....	23
2.2.1.6.4	Tampil Umat.....	23
2.2.1.6.5	Cari Umat.....	24
2.2.2	Class Diagram .....	25
2.2.3	Class Diagram Specific Descriptions .....	26
2.2.3.1	Specific Design Class Login.....	26
2.2.3.2	Specific Design Class PengelolaanWilayahUI 26	
2.2.3.3	Specific Design Class PengelolaanStasiUI 27	

2.2.3.4	Specific Design Class PengelolaanLingkunganUI	27
2.2.3.5	Specific Design Class PengelolaanKepalaKeluargaUI	28
2.2.3.6	Specific Design Class PengelolaanUmatUI	28
2.2.3.7	Specific Design Class LoginCtrl	29
2.2.3.8	Spesific Design Class WilayahCtrl	30
2.2.3.9	Spesific Design Class StasiCtrl	30
2.2.3.10	Spesific Design Class LingkunganCtrl	31
2.2.3.11	Spesific Design KepalaKeluargaCtrl	32
2.2.3.12	Spesific Design Class UmatCtrl	33
2.2.3.13	Spesific Design Class Pengguna	33
2.2.3.14	Spesific Design Class Wilayah	34
2.2.3.15	Spesific Design Class Stasi	35
2.2.3.16	Spesific Design Class Lingkungan	35
2.2.3.17	Spesific Design Class KepalaKeluarga	36
2.2.3.18	Spesific Design Class Umat	37
3	Perancangan Data	39
3.1	Dekomposisi Data	39
3.1.1	Deskripsi Entitas Data Pengguna	39
3.1.2	Deskripsi Entitas Data Keuskupan	40
3.1.3	Deskripsi Entitas Data Paroki	40
3.1.4	Deskripsi Entitas Data Wilayah	40
3.1.5	Deskripsi Entitas Data Stasi	41
3.1.6	Deskripsi Entitas Data Lingkungan	41
3.1.7	Deskripsi Entitas Data Kepala Keluarga	41
3.1.8	Deskripsi Entitas Data Umat	42
3.1.9	Deskripsi Entitas Data Ref_Kota	45
3.1.10	Deskripsi Entitas Data Ref_Status_ekonomi	45
3.1.11	Deskripsi Entitas Data Ref_Jenis_Kepala_keluarga	45
3.1.12	Deskripsi Entitas Data Ref_Jenis_Kelamin	46
3.1.13	Deskripsi Entitas Data Ref_Suku	46
3.1.14	Deskripsi Entitas Data Ref_Hubungan_Kepala_keluarga	46
3.1.15	Deskripsi Entitas Data Ref_Status_Pendidikan	46
3.1.16	Deskripsi Entitas Data Ref_Pendidikan_Terakhir	47
3.1.17	Deskripsi Entitas Data Ref_Bidang_Studi	47
3.1.18	Deskripsi Entitas Data Ref_Pekerjaan	47
3.1.19	Deskripsi Entitas Data Ref_Status_keluarga	47
3.1.20	Deskripsi Entitas Data Ref_Status_Baptis	47
3.1.21	Deskripsi Entitas Data Ref_Golongan_Darah	47
3.1.22	Deskripsi Entitas Data Ref_Status_Katolik	48
3.1.23	Deskripsi Entitas Data Ref_Jabatan_Paroki	48
3.1.24	Deskripsi Entitas Data Ref_Status_Gereja	48
3.1.25	Deskripsi Entitas Data Ref_Tempat_Tinggal	48
3.1.26	Deskripsi Entitas Data Ref_Status_Sosial	48
3.1.27	Deskripsi Entitas Data Ref_Agama	49
3.1.28	Deskripsi Entitas Data Ref_Status_Perkawinan	49
3.1.29	Deskripsi Entitas Data Ref_Status_Kesehatan	49

3.1.30	Deskripsi Entitas Data Role .....	49
3.2	Physical Data Model .....	50
	Gambar 3.2 : Physical data model.....	50
4	Perancangan Antarmuka.....	51
4.1	Rancangan Antarmuka Login .....	51
4.2	Rancangan Antarmuka Menu .....	52
4.3	Rancangan Antarmuka Pengelolaan Wilayah.....	53
4.4	Rancangan antarmuka pengelolaan Stasi.....	54
4.5	Rancangan Antarmuka Pengelolaan Lingkungan... ..	55
4.6	Rancangan Antarmuka Pengelolaan Kepala Keluarga.....	57
4.7	Rancangan Antarmuka Pengelolaan Umat.....	59
4.8	Rancangan Antarmuka Pengelolaan Pengguna.....	62
4.9	Rancangan Antarmuka Ubah Password.....	63
4.10	Rancangan Antarmuka Pencarian Umat.....	64



# 1 Pendahuluan

## 1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak SIMAG yaitu web pengelolaan umat gereja. Dokumen DPPL ini digunakan oleh pengembang perangkat lunak SIMAG sebagai acuan untuk implementasi pada tahap selanjutnya.

## 1.2 Ruang Lingkup

SIMAG dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan paroki.
2. Menangani pengelolaan wilayah.
3. Menangani pengelolaan stasi.
4. Menangani pengelolaan lingkungan.
5. Menangani pengelolaan kepala keluarga.
6. Menangani pengelolaan umat.
7. Menangani pengelolaan pengguna.
8. Menangani pengelolaan laporan.

SIMAG dikembangkan sebagai sebuah aplikasi web, sehingga dapat berjalan pada semua platform sistem operasi yang dilengkapi dengan web browser.

## 1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
DPPL-SIMAG-XXX	Kode yang merepresentasikan kebutuhan pada WEB Sistem Informasi Pengelolaan umat dimana XXX merupakan nomor fungsi produk.

Program Studi Teknik Informatika	DPPL – SIMAG	7/ 64
----------------------------------	--------------	-------

SIMAG	Merupakan kependekan dari Sistem Informasi Pengelolaan umat gereja yang merupakan Perangkat lunak berbasis web untuk menangani gereja .
Berbasis web (Web Based)	Menyatakan aplikasi yang dalam penggunaannya pada komputer client dapat diakses mempergunakan web browser tanpa memerlukan proses instalasi terlebih dahulu.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

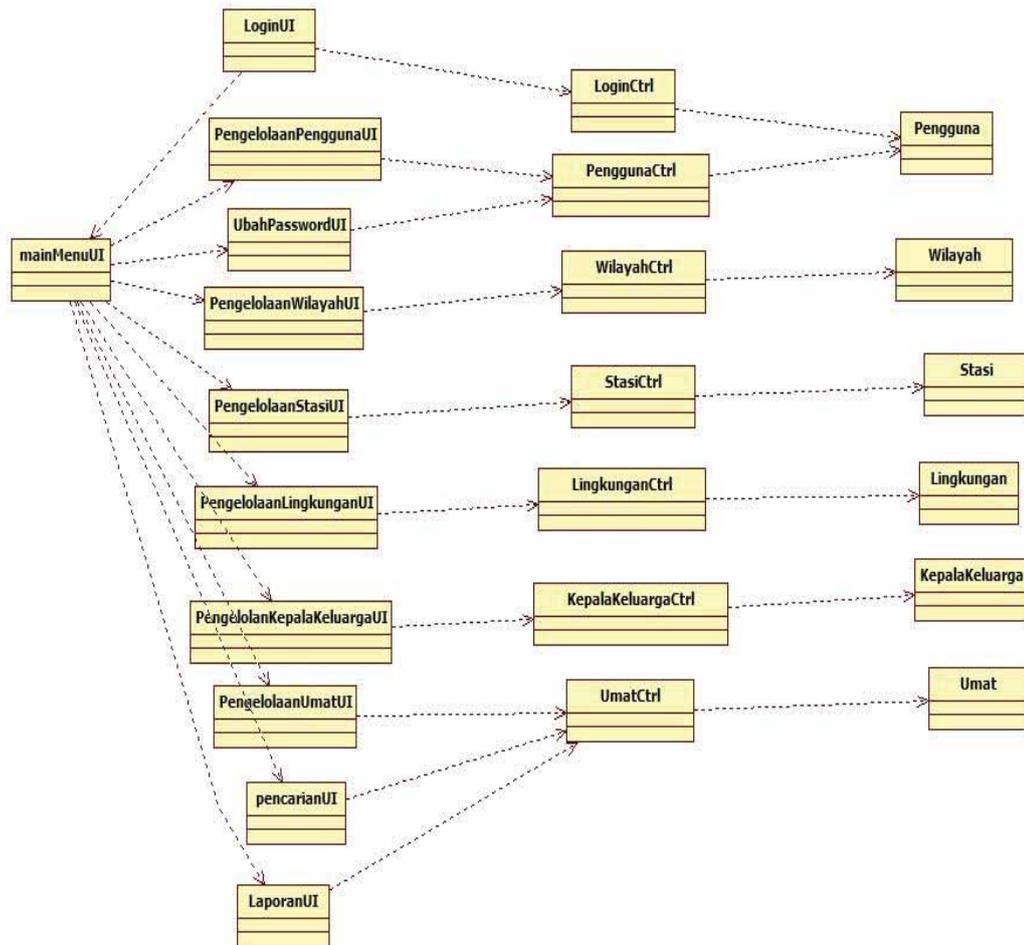
#### 1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Felisia Novita Sari, *Spesifikasi Kebutuhan Perangkat Lunak SeiMah*, Universitas Atma Jaya Yogyakarta, 2013.
2. Lucia Desta Tri Utami. *Deskripsi Perancangan Perangkat Lunak CFLS*, Universitas Atma Jaya Yogyakarta, 2013.

## 2 Analysis Model

### 2.1 Perancangan Arsitektur

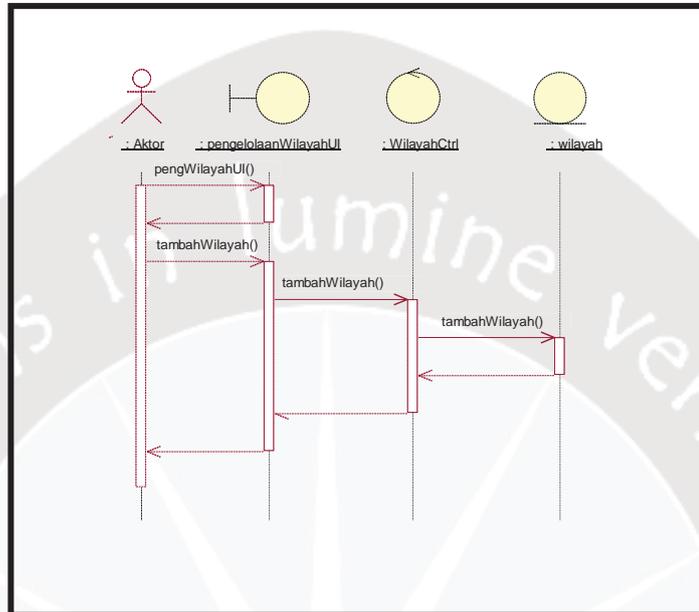


Gambar 2.1 Arsitektur Aplikasi SIMAG



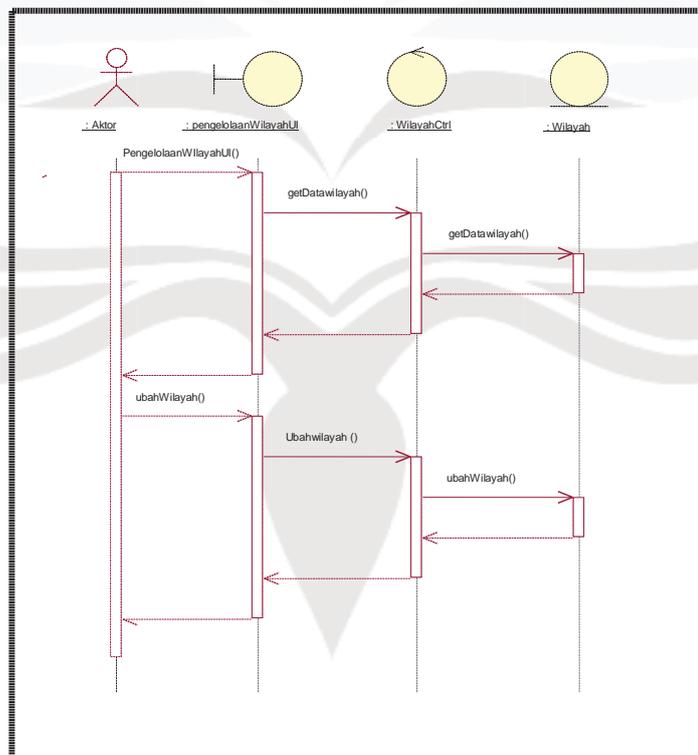
## 2.2.1.2 Pengelolaan Wilayah

### 2.2.1.2.1 Tambah Wilayah Baru



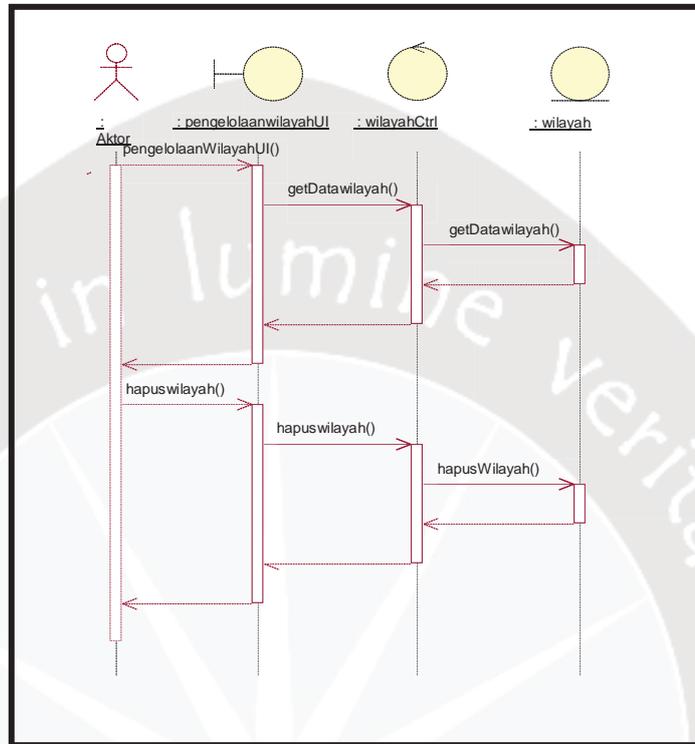
Gambar 2.3 Sequence Diagram : Tambah Wilayah Baru

### 2.2.1.2.2 Ubah Wilayah



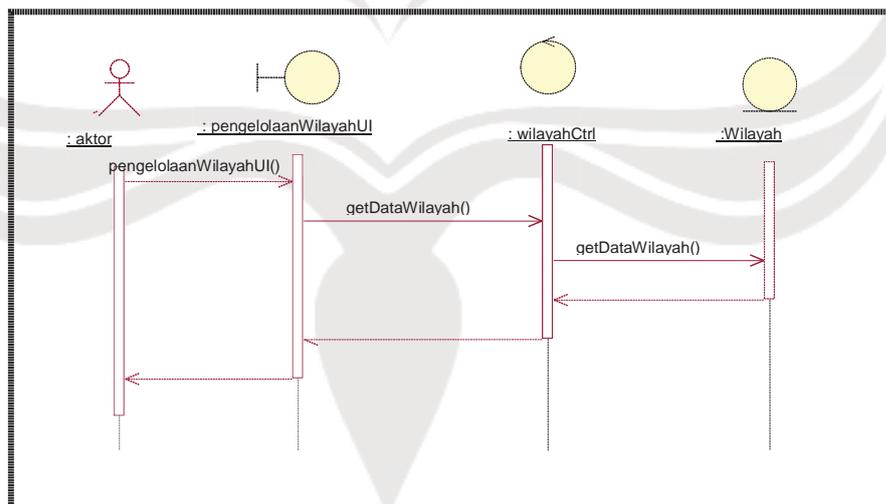
Gambar 2.4 Sequence Diagram : ubah wilayah

### 2.2.1.2.3 Hapus Wilayah



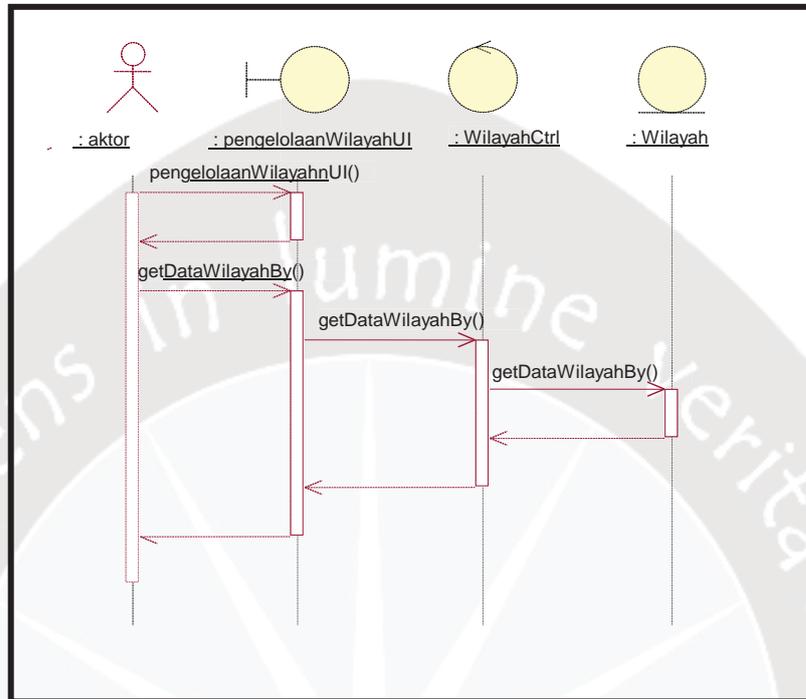
Gambar 2.5 Sequence Diagram : Hapus Wilayah

### 2.2.1.2.4 Tampil Wilayah



Gambar 2.6 Sequence Diagram : Tampil Wilayah

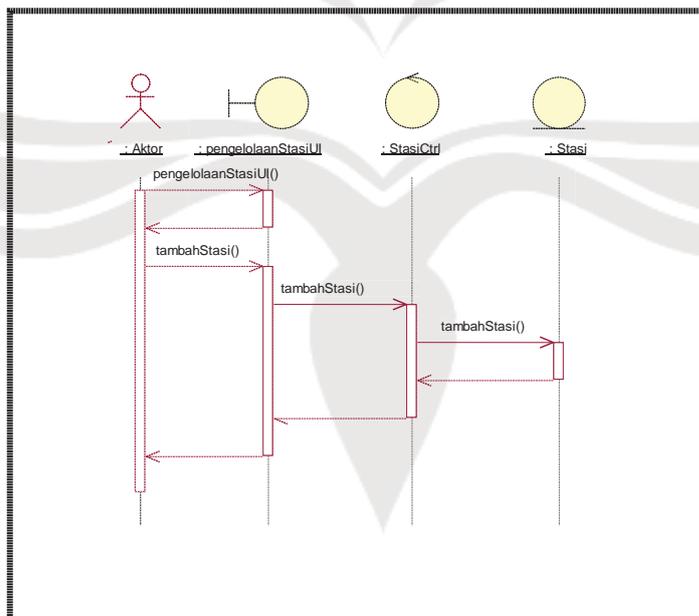
### 2.2.1.2.5 Cari Wilayah



Gambar 2.7 Sequence Diagram : Cari Wilayah

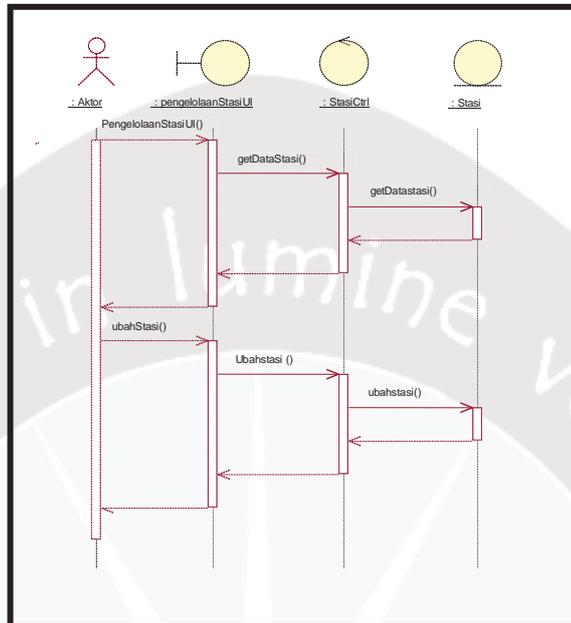
### 2.2.1.3 Pengelolaan Stasi

#### 2.2.1.3.1 Tambah Stasi Baru



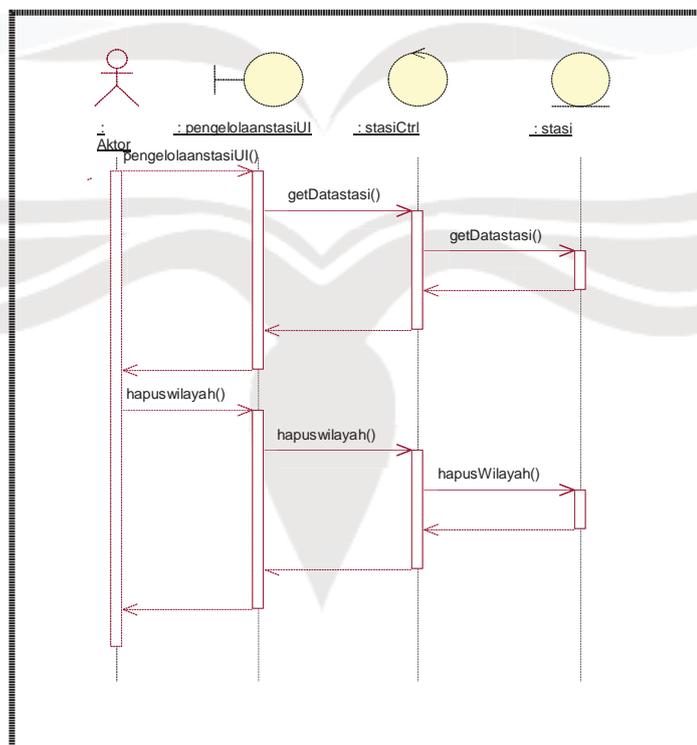
Gambar 2.8 Sequence Diagram : Tambah Stasi Baru

### 2.2.1.3.2 Ubah Stasi



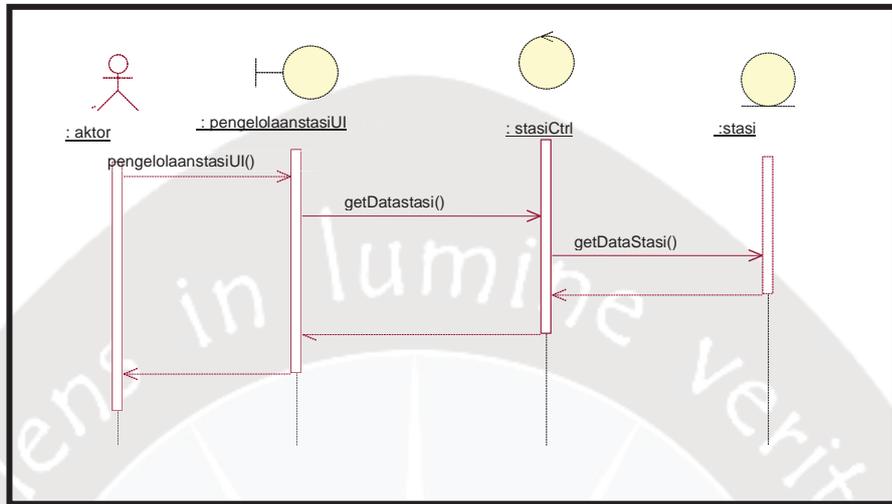
gambar 2.9 Sequence Diagram : Ubah stasi

### 2.2.1.3.3 Hapus stasi



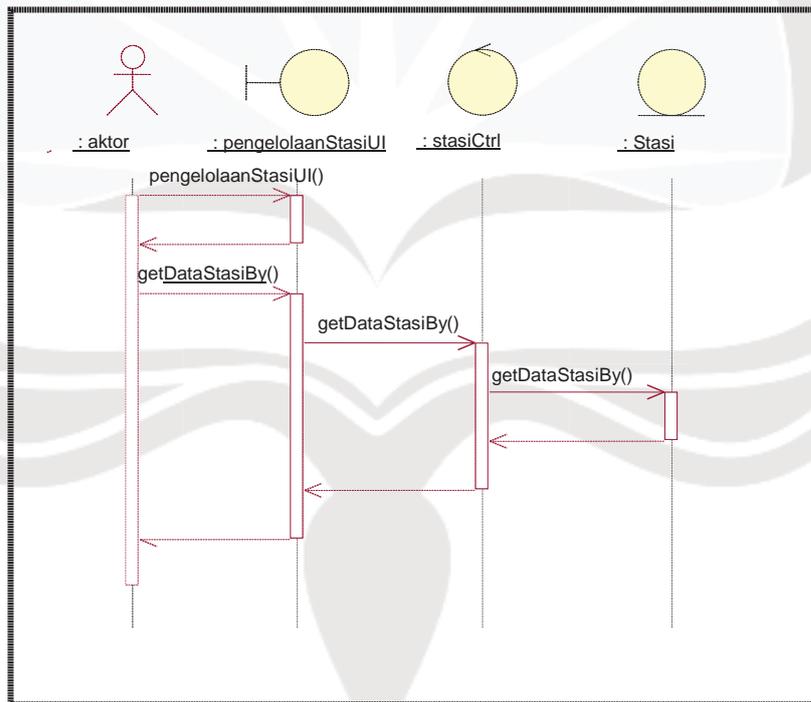
Gambar 2.10 Sequence Diagram : Hapus Stasi

#### 2.2.1.3.4 Tampil Stasi



Gambar 2.11 Sequence Diagram : Tampil Stasi

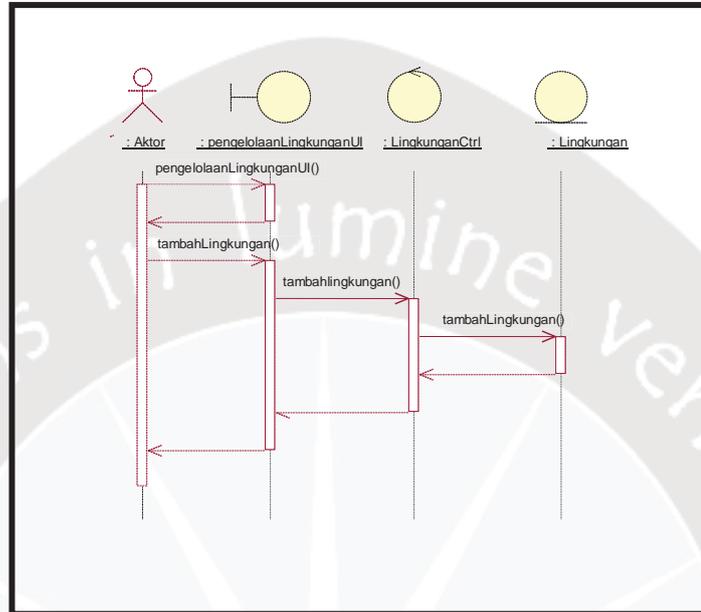
#### 2.2.1.3.5 Cari Stasi



Gambar 2.12 Sequence Diagram : cari stasi

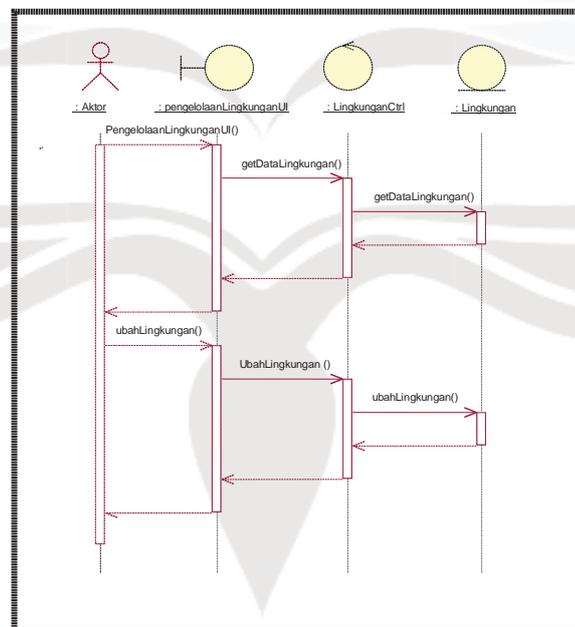
## 2.2.1.4 Pengelolaan Lingkungan

### 2.2.1.4.1 Tambah lingkungan



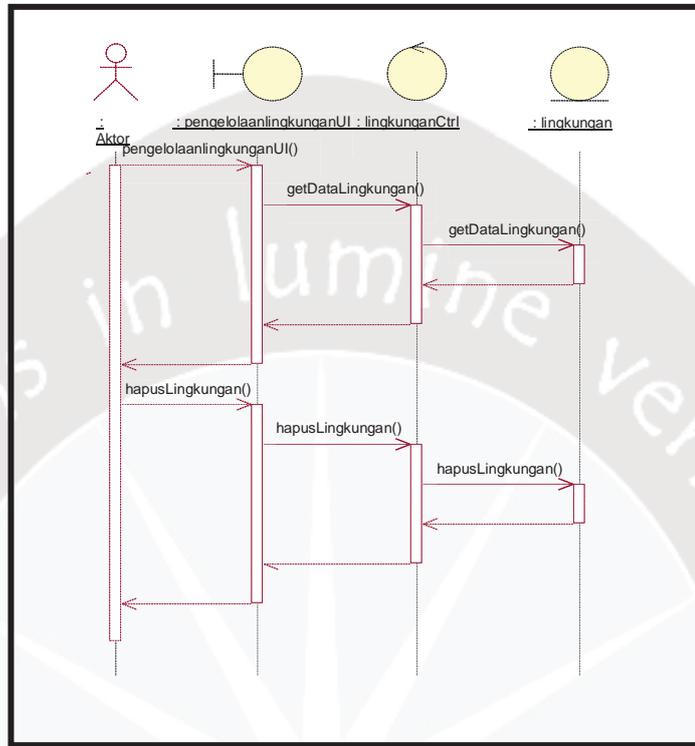
Gambar 2.13 Sequence Diagram : Tambah lingkungan

### 2.2.1.4.2 Ubah Lingkungan



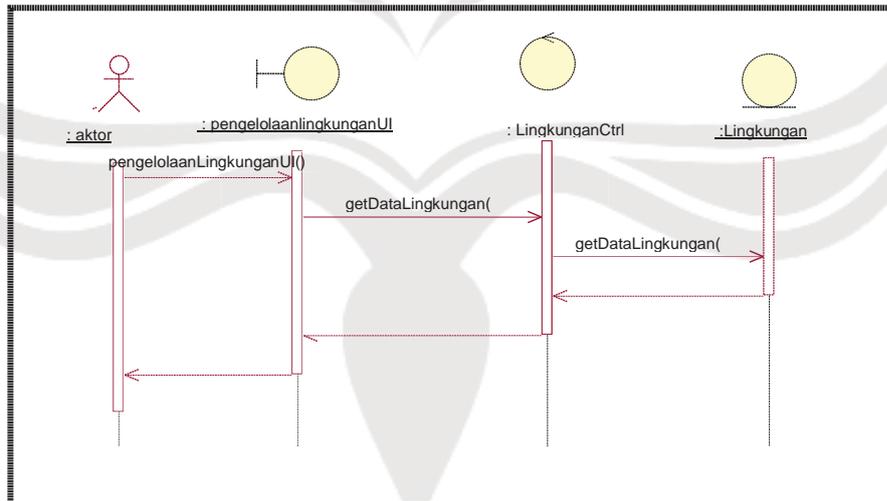
Gambar 2.14 Sequence Diagram : Ubah lingkungan

### 2.2.1.4.3 Hapus Lingkungan



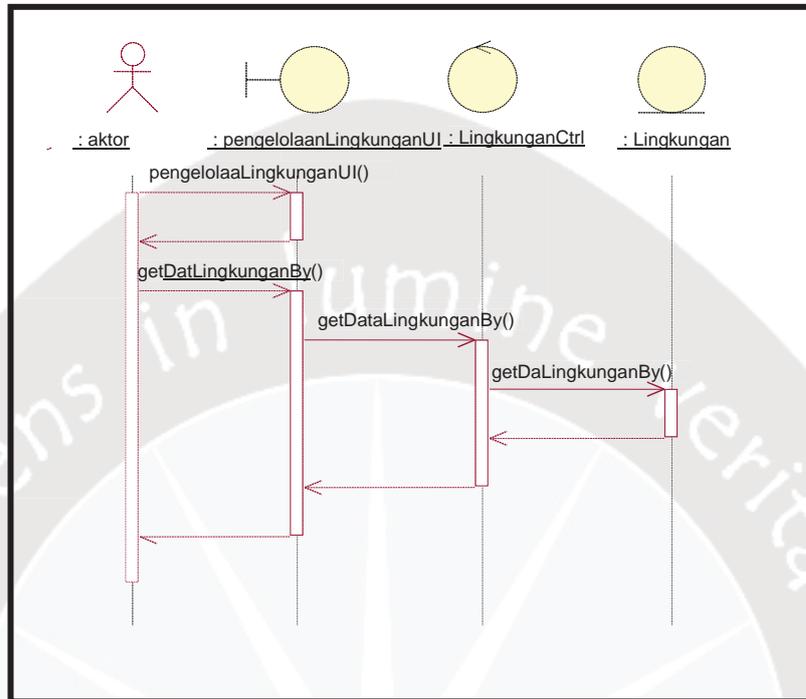
Gambar 2.15 Sequence Diagram : Hapus lingkungan

### 2.2.1.4.4 Tampil Lingkungan



Gambar 2.16 Sequence Diagram : Tampil Lingkungan

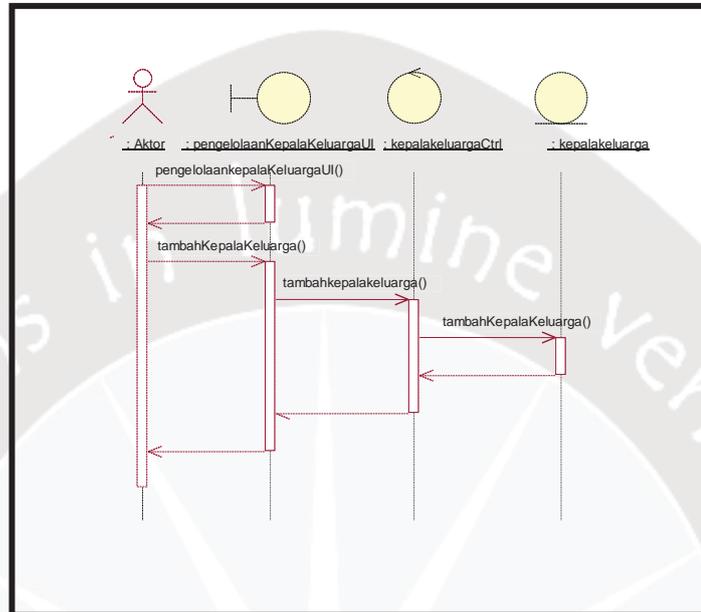
#### 2.2.1.4.5 Cari Lingkungan



Gambar 2.17 Sequence Diagram : Cari Lingkungan

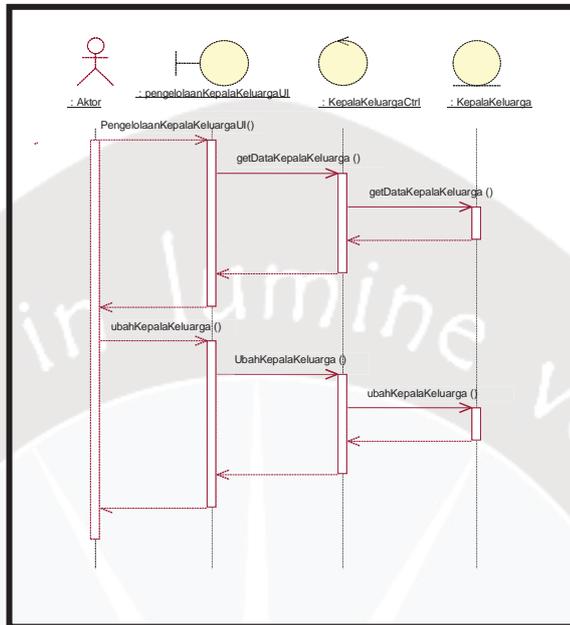
## 2.2.1.5 Pengelolaan Kepala Keluarga

### 2.2.1.5.1 Tambah kepala Keluarga



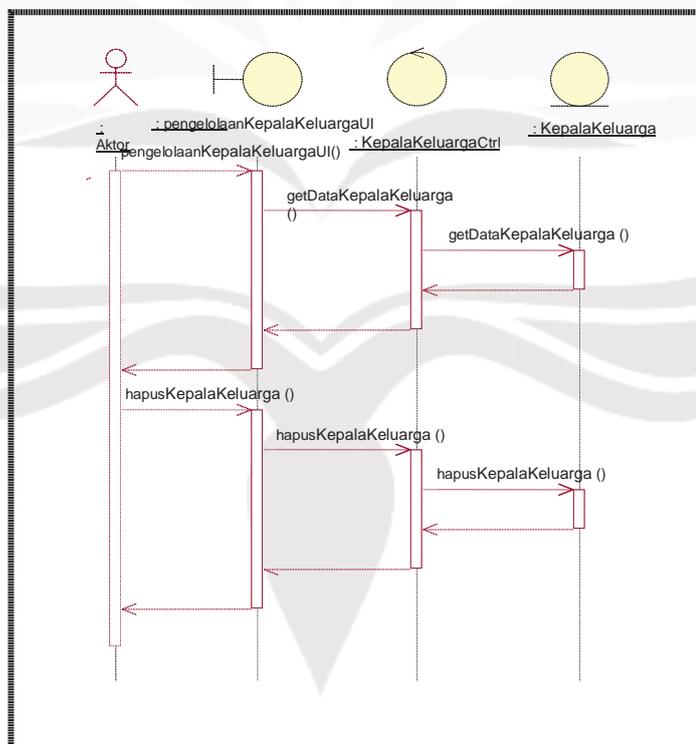
Gambar 2.18 Sequence Diagram : Tambah Kepala Keluarga

### 2.2.1.5.2 Ubah Kepala Keluarga



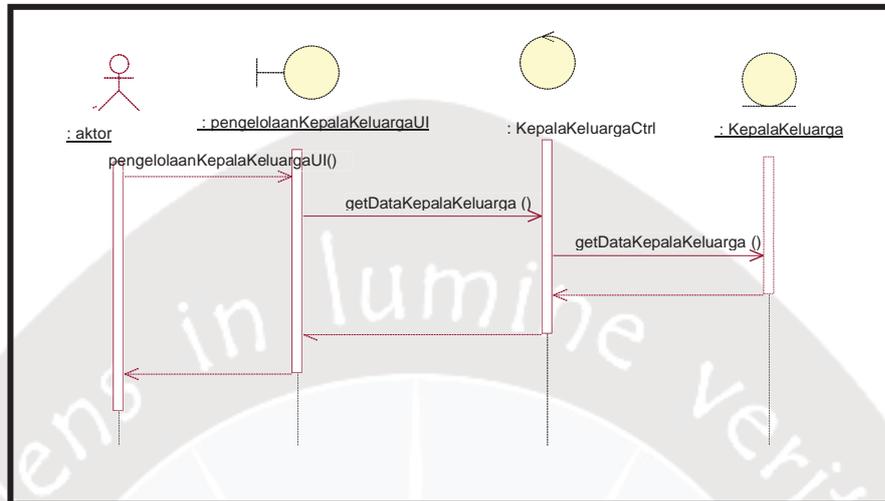
Gambar 2.19 Sequence Diagram : Ubah Kepala Keluarga

### 2.2.1.5.3 Hapus Kepala Keluarga



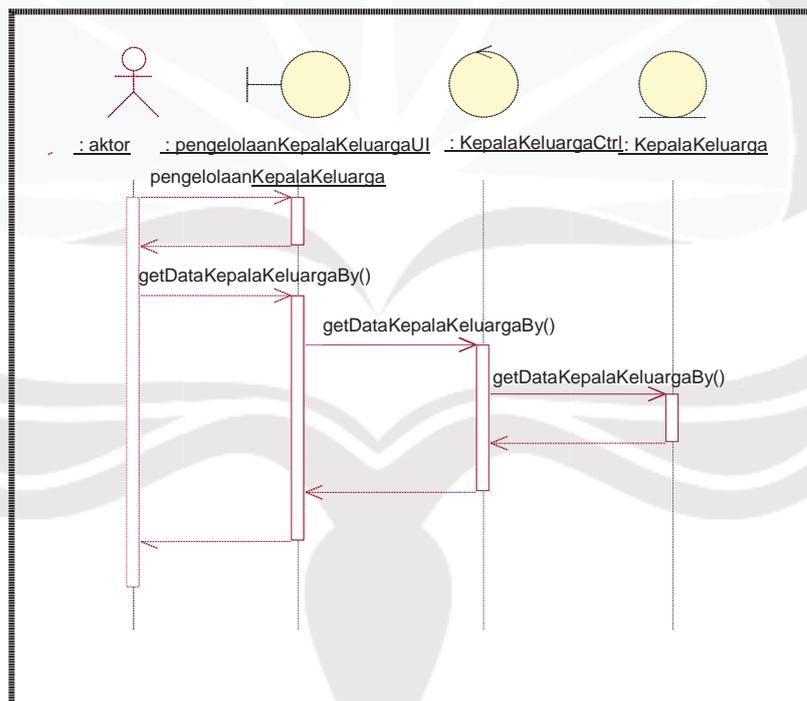
Gambar 2.20 Sequence Diagram : Hapus Kepala Keluarga

#### 2.2.1.5.4 Tampil Kepala Keluarga



Gambar 2.21 Sequence Diagram : Tampil Kepala Keluarga

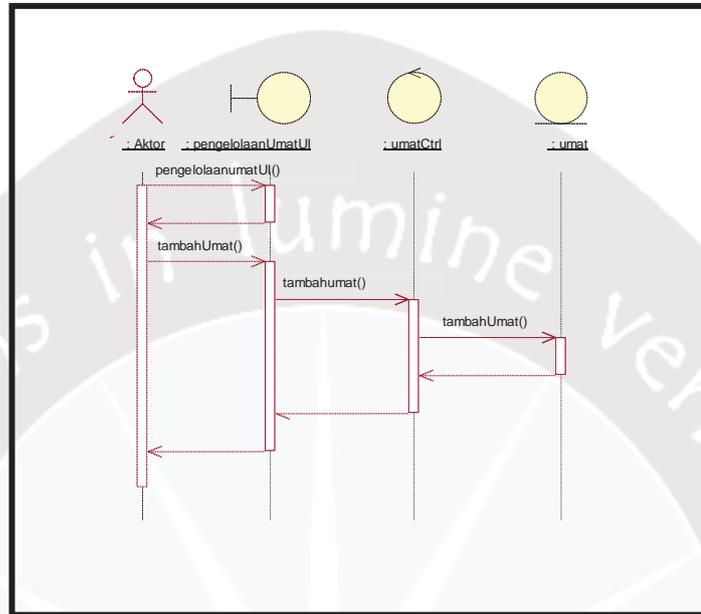
#### 2.2.1.5.5 Cari kepala Keluarga



Gambar 2.22 Sequence Diagram : Tambah Kepala Keluarga

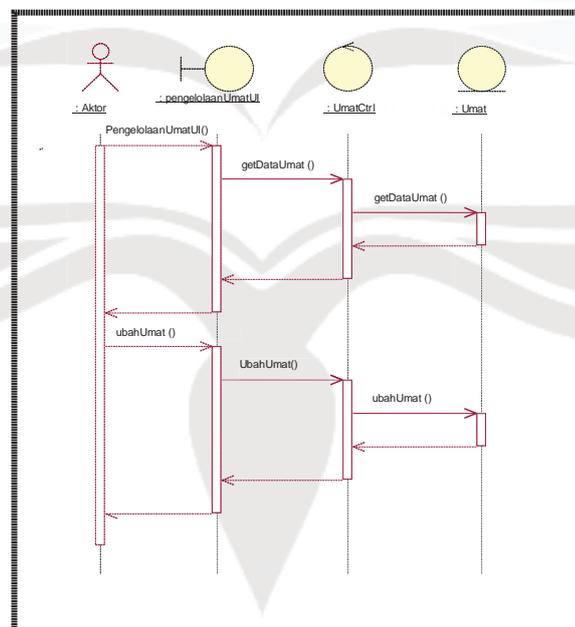
## 2.2.1.6 Pengelolaan Umat

### 2.2.1.6.1 Tambah Umat



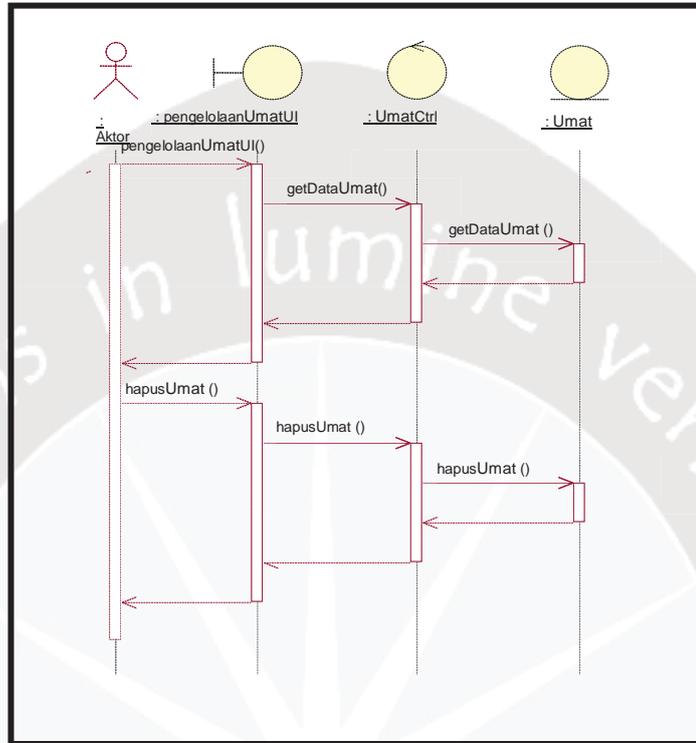
Gambar 2.23 Sequence Diagram : Tambah Umat

### 2.2.1.6.2 Ubah Umat



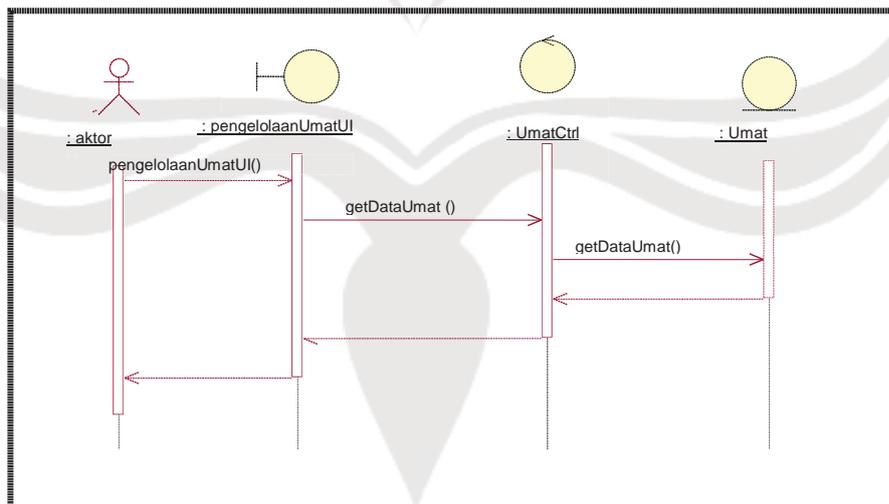
Gambar 2.24 Sequence Diagram : Ubah Umat

### 2.2.1.6.3 Hapus Umat



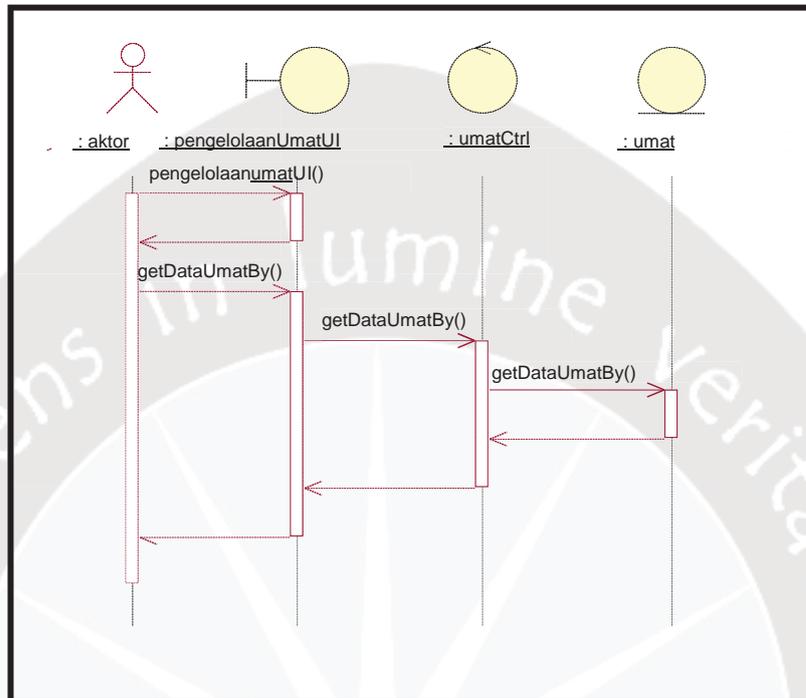
Gambar 2.25 Sequence Diagram : Hapus Umat

### 2.2.1.6.4 Tampil Umat



Gambar 2.26 Sequence Diagram : Tampil Umat

### 2.2.1.6.5 Cari Umat



Gambar 2.27 Sequence Diagram : Cari Umat



## 2.2.3 Class Diagram Specific Descriptions

### 2.2.3.1 Specific Design Class Login

Login	<<boundary>>
+Login() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+getDataLogin() : String Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu username dan password.	
+cekLogin():boolean Operasi ini digunakan untuk memvalidasi data login	

### 2.2.3.2 Specific Design Class PengelolaanWilayahUI

PengelolaanWilayahUI	<<boundary>>
+PengelolaanWilayahUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
+tambahWilayah():void Operasi ini digunakan untuk menambahkan data-data wilayah baru.	
+ubahWilayah():void Operasi ini digunakan untuk melakukan perubahan data wilayah.	
+hapusWilayah():void Operasi ini digunakan untuk menghapus data wilayah.	
+getDataWilayah():void Operasi ini digunakan untuk menampilkan seluruh isi dari data wilayah.	
+GetDataWilayahBy():void Operasi ini digunakan untuk mencari data wilayah berdasarkan	

kata kunci.

### 2.2.3.3 Specific Design Class PengelolaanStasiunUI

PengelolaanStasiunUI	<<boundary>>
<pre>+PengelolaanStasiunUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +tambahStasiun():void Operasi ini digunakan untuk menambahkan data-data stasiun baru. +ubahStasiun():void Operasi ini digunakan untuk melakukan perubahan data stasiun. +hapusStasiun():void Operasi ini digunakan untuk menghapus data stasiun. +getDataStasiun():void Operasi ini digunakan untuk menampilkan seluruh isi dari data stasiun. +GetStasiunBy():void Operasi ini digunakan untuk mencari data stasiun berdasarkan kata kunci.</pre>	

### 2.2.3.4 Specific Design Class PengelolaanLingkunganUI

PengelolaanLingkunganUI	<<boundary>>
<pre>+PengelolaanLingkunganUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +tambahLingkungan():void Operasi ini digunakan untuk menambahkan data-data stasiun baru. +ubahLingkungan():void Operasi ini digunakan untuk melakukan perubahan data stasiun. +hapusLingkungan():void</pre>	

Operasi ini digunakan untuk menghapus data stasi.

```
+getDataLingkungan():void
```

Operasi ini digunakan untuk menampilkan seluruh isi dari data lingkungan.

```
+GetDataLingkunganBy():void
```

Operasi ini digunakan untuk mencari data lingkungan berdasarkan kata kunci.

**2.2.3.5 Specific Design Class PengelolaanKepalaKeluargaUI**

PengelolaanKepalaKeluargaUI	<<Boundary>>
<pre>+PengelolaanKepalaKeluargaUI()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <pre>+tambahKepalaKeluarga():void</pre> <p>Operasi ini digunakan untuk menambahkan data-data kepala keluarga baru.</p> <pre>+ubahKepalaKeluarga():void</pre> <p>Operasi ini digunakan untuk melakukan perubahan data kepala keluarga.</p> <pre>+hapusKepalaKeluarga():void</pre> <p>Operasi ini digunakan untuk menghapus data kepala keluarga.</p> <pre>+getDataKepalaKeluarga():void</pre> <p>Operasi ini digunakan untuk menampilkan seluruh isi dari data kepala keluarga.</p> <pre>+GetDataKepalaKeluargaBy():void</pre> <p>Operasi ini digunakan untuk mencari data kepala keluarga berdasarkan kata kunci.</p>	

**2.2.3.6 Specific Design Class PengelolaanUmatUI**

PengelolaanUmatUI	<<boundary>>
<pre>+PengelolaanKepalaUmatUI()</pre>	

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

+tambahUmat():void  
Operasi ini digunakan untuk menambahkan data-data umat baru.

+ubahUmat():void  
Operasi ini digunakan untuk melakukan perubahan data umat.

+hapusUmat():void  
Operasi ini digunakan untuk menghapus data umat.

+getDataUmat():void  
Operasi ini digunakan untuk menampilkan seluruh isi dari data umat.

+GetDataUmatBy():void  
Operasi ini digunakan untuk mencari data umat berdasarkan kata kunci.

### 2.2.3.7 Specific Design Class LoginCtrl

<b>LoginCtrl</b>	<b>&lt;&lt;control&gt;&gt;</b>
<p>+LoginCtrl() Default konstruktor, digunakan untuk inisialisasi atribut yang berada dikelas ini.</p> <p>+ValidasiLogin() : void Operasi ini digunakan untuk mengecek data yang diinputkan oleh user.</p> <p>+GetDataPengguna() : void Operasi ini digunakan untuk mengambil data pengguna</p>	

### 2.2.3.8 Specific Design Class WilayahCtrl

<b>WilayahCtrl</b>	<b>&lt;&lt;control&gt;&gt;</b>
<p>+WilayahCtrl() Default konstruktor, digunakan untuk inisialisasi atribut yang berada dikelas ini</p> <p>+tambahWilayah():void Operasi ini digunakan untuk menambahkan data-data wilayah baru.</p> <p>+ubahWilayah():void Operasi ini digunakan untuk melakukan perubahan data wilayah.</p> <p>+hapusWilayah():void Operasi ini digunakan untuk menghapus data wilayah.</p> <p>+getDataWilayah(): void Operasi ini digunakan untuk menampilkan seluruh isi dari data wilayah.</p> <p>+GetDataWilayahBy(): void Operasi ini digunakan untuk mencari data wilayah berdasarkan kata kunci.</p>	

### 2.2.3.9 Specific Design Class StasiCtrl

<b>StasiCtrl</b>	<b>&lt;&lt;control&gt;&gt;</b>
------------------	--------------------------------

+StasiCtrl()

Default konstruktor, digunakan untuk inisialisasi atribut yang berada dikelas ini

+tambahStasi():void

Operasi ini digunakan untuk menambahkan data-data stasi baru.

+ubahStasi():void

Operasi ini digunakan untuk melakukan perubahan data stasi.

+hapusStasi():void

Operasi ini digunakan untuk menghapus data stasi.

+getDataStasi(): void

Operasi ini digunakan untuk menampilkan seluruh isi dari data stasi.

+GetDataStasiBy():void

Operasi ini digunakan untuk mencari data stasi berdasarkan kata kunci.

#### 2.2.3.10 Spesific Design Class LingkunganCtrl

LingkunganCtrl

<<control>>

+LingkunganCtrl()

Default konstruktor, digunakan untuk inisialisasi atribut yang berada dikelas ini

+tambahLingkungan():void

Operasi ini digunakan untuk menambahkan data-data stasi baru.

+ubahLingkungan():void

Operasi ini digunakan untuk melakukan perubahan data stasi.

+hapusLingkungan():void

Operasi ini digunakan untuk menghapus data stasi.

+getDataLingkungan(): void

Operasi ini digunakan untuk menampilkan seluruh isi dari data lingkungan.

+GetDataLingkunganBy(): void

Operasi ini digunakan untuk mencari data lingkungan berdasarkan kata kunci.

### 2.2.3.11 Spesific Design KepalaKeluargaCtrl

KepalaKeluargaCtrl	<<control>>
<p>+KepalaKeluargaCtrl() Default konstruktor, digunakan untuk inisialisasi atribut yang berada dikelas ini</p> <p>+tambahKepalaKeluarga():void Operasi ini digunakan untuk menambahkan data-data kepala keluarga baru.</p> <p>+ubahKepalaKeluarga():void Operasi ini digunakan untuk melakukan perubahan data kepala keluarga.</p> <p>+hapusKepalaKeluarga():void Operasi ini digunakan untuk menghapus data kepala keluarga.</p> <p>+getDataKepalaKeluarga(): void Operasi ini digunakan untuk menampilkan seluruh isi dari data kepala keluarga.</p> <p>+GetDataKepalaKeluargaBy(): void Operasi ini digunakan untuk mencari data kepala keluarga berdasarkan kata kunci.</p>	

### 2.2.3.12 Specific Design Class UmatCtrl

<b>UmatCtrl</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre> +umatCtrl() Default konstruktor, digunakan untuk inisialisasi atribut yang berada dikelas ini +tambahUmat():void Operasi ini digunakan untuk menambahkan data-data umat baru. +ubahUmat():void Operasi ini digunakan untuk melakukan pengubahan data umat. +hapusUmat():void Operasi ini digunakan untuk menghapus data umat. +getDataUmat(): void Operasi ini digunakan untuk menampilkan seluruh isi dari data umat. +GetDataUmatBy(): void Operasi ini digunakan untuk mencari data umat berdasarkan kata kunci. </pre>	

### 2.2.3.13 Specific Design Class Pengguna

<b>Pengguna</b>	<b>&lt;&lt;Entity&gt;&gt;</b>
<pre> - Id_pengguna : Integer Atribut ini digunakan untuk menyimpan id dari pengguna - Nama : String Atribut ini digunakan untuk menyimpan nama dari pengguna - Username : String Atribut ini digunakan untuk menyimpan username dari pengguna - Password : String Atribut ini digunakan untuk menyimpan password dari pengguna - No_telepon : String Atribut ini digunakan untuk menyimpan no telepon dari pengguna </pre>	
<pre> +PengelolaanLingkunganUI() </pre>	

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

+tambahLingkungan(): Boolean  
Operasi ini digunakan untuk menambahkan data-data stasi baru.

+ubahLingkungan(): Boolean  
Operasi ini digunakan untuk melakukan perubahan data stasi.

+hapusLingkungan(): Boolean  
Operasi ini digunakan untuk menghapus data stasi.

+getDataLingkungan(): Dataset  
Operasi ini digunakan untuk menampilkan seluruh isi dari data lingkungan.

+GetDataLingkunganBy(): Dataset  
Operasi ini digunakan untuk mencari data lingkungan berdasarkan kata kunci.

#### 2.2.3.14 Spesific Design Class Wilayah

Wilayah	<<Entity>>
<p>- Id_wilayah : Integer atribut ini digunakan untuk menyimpan id wilayah</p> <p>- Kode_wilayah : String Atribut ini digunakan untuk menyimpan kode dari wilayah.</p> <p>- nama_wilayah : String Atribut ini digunakan untuk menyimpan nama sebuah wilayah.</p> <p>- Is_deleted : Boolean Atribut digunakan untuk menyimpan data apakah sudah dihapus atau belum</p> <p>- Id_paroki: Integer atribut ini digunakan untuk menyimpan id dari tabel paroki</p>	
<p>+ Wilayah () digunakan untuk inisialisasi semua atribut dari kelas ini</p>	

### 2.2.3.15 Specific Design Class Stasi

<b>Stasi</b>	<b>&lt;&lt;Entity&gt;&gt;</b>
<p>- Id_stasi : Integer atribut ini digunakan untuk menyimpan id stasi</p> <p>- Kode_ stasi : String Atribut ini digunakan untuk menyimpan kode dari stasi.</p> <p>- nama_ stasi : String Atribut ini digunakan untuk menyimpan nama sebuah stasi.</p> <p>- Is_deleted : Boolean Atribut digunakan untuk menyimpan data apakah sudah dihapus atau belum</p> <p>- Id_paroki: Integer atribut ini digunakan untuk menyimpan id dari tabel paroki</p>	
<p>+Stasi() digunakan untuk inisialisasi semua atribut dari kelas ini.</p>	

### 2.2.3.16 Specific Design Class Lingkungan

<b>Lingkungan</b>	<b>&lt;&lt;Entity&gt;&gt;</b>
<p>- Id_lingkungan : Integer atribut ini digunakan untuk menyimpan id stasi</p> <p>- Kode_lingkungan : String Atribut ini digunakan untuk menyimpan kode dari stasi.</p> <p>- nama_lingkungan: String Atribut ini digunakan untuk menyimpan nama sebuah stasi.</p> <p>- Is_deleted : Boolean Atribut digunakan untuk menyimpan data apakah sudah dihapus atau belum</p> <p>- Id_stasi: Integer atribut ini digunakan untuk menyimpan id dari tabel paroki</p> <p>- Id_wilayah : Integer atribut ini digunakan untuk menyimpan id dari tabel wilayah</p>	
<p>+Lingkungan () digunakan untuk inisialisasi semua atribut dari kelas ini.</p>	

### 2.2.3.17 Spesific Design Class KepalaKeluarga

KepalaKeluarga	<<Entity>>
<p>- Id_kepalakeluarga : Integer atribut ini digunakan untuk menyimpan id kepala keluarga</p> <p>- Kode_kepala_keluarga : String Atribut ini digunakan untuk menyimpan kode dari kepalakeluarga.</p> <p>- nama_kepala_keluarga: String Atribut ini digunakan untuk menyimpan nama sebuah kepala keluarga.</p> <p>- Alamat_ kepala_keluarga : String Atribut ini digunakan untuk menyimpan alamat dari sebuah alamat.</p> <p>- No_telepon Atribut ini digunakan untuk menyimpan nomor telepon.</p> <p>- Tgl_nikah : Date Atribut ini digunakan untuk menyimpan tanggal pernikahan.</p> <p>- Last_update Atribut ini untuk menyimpan tanggal terakhir data diperbaharui.</p> <p>- Is_deleted : Boolean Atribut digunakan untuk menyimpan data apakah sudah dihapus atau belum</p> <p>- Id_lingkungan : Integer Atribut ini digunakan untuk menyimpan id dari lingkungan.</p> <p>- Id_status_ekonomi : Integer Atribut ini digunakan untuk menyimpan id dari status ekonomi.</p> <p>- Id_jenis_kepala_keluarga : Integer Atribut ini digunakan untuk menyimpan data id dari jenis kepala keluarga.</p> <p>- Id_kota_kk : Integer</p>	

<p>Atribut ini digunakan untuk menyimpan id dari kota.</p> <ul style="list-style-type: none"> <li>- Id_kota_nikah : Integer</li> </ul> <p>Atribut ini digunakan untuk menyimpan id dari kota tempat menikah.</p> <ul style="list-style-type: none"> <li>- Id_kota_pindah_dari : Integer</li> </ul> <p>Atribut ini digunakan untuk menyimpan id dari kota tempat pindah</p> <ul style="list-style-type: none"> <li>- Id_kota_pindah_ke : Integer</li> </ul> <p>Atribut ini digunakan untuk menyimpan id dari tempat kota tujuan pindah</p>
<p>+ KepalaKeluarga() digunakan untuk inisialisasi semua atribut dari kelas ini.</p>

### 2.2.3.18 Spesific Design Class Umat

<b>umat</b>	<b>&lt;&lt;Entity&gt;&gt;</b>
<ul style="list-style-type: none"> <li>- Id_umat : Integer</li> </ul> <p>atribut ini digunakan untuk menyimpan id kepala keluarga</p> <ul style="list-style-type: none"> <li>- Kode_umat : String</li> </ul> <p>Atribut ini digunakan untuk menyimpan kode dari kepalakeluarga.</p> <ul style="list-style-type: none"> <li>- nama_umat : String</li> </ul> <p>Atribut ini digunakan untuk menyimpan nama sebuah kepala keluarga.</p> <ul style="list-style-type: none"> <li>- Tgl_lahir: String</li> </ul> <p>Atribut ini digunakan untuk menyimpan alamat dari sebuah alamat.</p> <p>Tgl_baptis</p> <p>Atribut ini digunakan untuk menyimpan nomor telepon.</p> <ul style="list-style-type: none"> <li>- Tgl_krisma: Date</li> </ul> <p>Atribut ini digunakan untuk menyimpan tanggal pernikahan.</p> <p>Lama_tinggal</p> <ul style="list-style-type: none"> <li>- Last_update</li> </ul> <p>Atribut ini untuk menyimpan tanggal terakhir data</p>	

diperbaharui.

- Is\_deleted : Boolean

Atribut digunakan untuk menyimpan data apakah sudah dihapus atau belum

- Id\_kepala\_keluarga : Integer

Atribut ini digunakan untuk menyimpan id dari lingkungan.

- Id\_jenis\_kelamin : Integer

Atribut ini digunakan untuk menyimpan id dari status ekonomi.

- Id\_hubungan\_kk : Integer

Atribut ini digunakan untuk menyimpan data id dari jenis kepala keluarga.

- Id\_suku : Integer

Atribut ini digunakan untuk menyimpan id dari kota.

- Id\_status\_pendidikan : Integer

Atribut ini digunakan untuk menyimpan id dari kota tempat menikah.

- Id\_pendidikan\_terakhir : Integer

Atribut ini digunakan untuk menyimpan id dari kota tempat pindah

- Id\_bidang\_studi : Integer

Atribut ini digunakan untuk menyimpan id dari tempat kota tujuan pindah

- Id\_pekerjaan

Atribut ini digunakan untuk menyimpan id dari pekerjaan

- Id\_golongan\_darah

Atribut ini digunakan untuk menyimpan id dari golongan darah

- Id\_status\_baptis

Atribut ini digunakan untuk menyimpan id dari status baptis

- Id\_status\_kesehatan

Atribut ini digunakan untuk menyimpan id dari status kesehatan

- Id\_status\_katolik

Atribut ini digunakan untuk menyimpan id dari status katolik

- Id\_status\_keluarga  
Atribut ini digunakan untuk menyimpan id dari status keluarga
- Id\_jabatan\_paroki  
Atribut ini digunakan untuk menyimpan id dari jabatan paroki
- Id\_status\_gereja  
Atribut ini digunakan untuk menyimpan id dari status gereja
- Id\_tempat\_tinggal  
Atribut ini digunakan untuk menyimpan id dari tempat tinggal
- Id\_status\_sosial  
Atribut ini digunakan untuk menyimpan id dari status social
- Id\_agama  
Atribut ini digunakan untuk menyimpan id dari agama
- Id\_status\_perkawinan  
Atribut ini digunakan untuk menyimpan id dari status perkawinan
- Id\_kota\_lahir  
Atribut ini digunakan untuk menyimpan id dari kota tempat kelahiran
- Id\_kota\_baptis  
Atribut ini digunakan untuk menyimpan id dari kota tempat baptis
- Id\_kota\_krisma  
Atribut ini digunakan untuk menyimpan id dari kota tempat krisma

+Umat()  
digunakan untuk inisialisasi semua atribut dari kelas ini.

### 3 Perancangan Data

#### 3.1 Dekomposisi Data

##### 3.1.1 Deskripsi Entitas Data Pengguna

Nama	Tipe	Panjang	Keterangan
id_petugas	Integer	-	Id petugas, Primary key

nama_petugas	varchar	100	nama petugas
username	varchar	100	username petugas
password	varchar	100	password petugas
Id_role	Integer	-	Id dari role, foreign key ke tabel role

### 3.1.2 Deskripsi Entitas Data Keuskupan

Nama	Tipe	Panjang	Keterangan
id_keuskupan	Integer	-	Id keuskupan, primary key
Kode_keuskupan	Varchar	50	Kode dari keuskupan
nama_keuskupan	Varchar	100	nama dari sebuah keuskupan
Is_deleted	Boolean	-	Apakah sebuah keuskupan sudah dihapus atau belum

### 3.1.3 Deskripsi Entitas Data Paroki

Nama	Tipe	Panjang	Keterangan
id_paroki	Integer	-	Id paroki, primary key
Kode_paroki	Varchar	50	Kode dari paroki
nama_paroki	Varchar	100	nama dari sebuah paroki
Is_deleted	Boolean	-	Apakah sebuah paroki sudah dihapus atau belum
Paroki_terpilih	Boolean	-	Apakah sebuah paroki sudah dipilih atau tidak
Id_keuskupan	Integer	-	Id dari keuskupan, foreign key ke tabel keuskupan
Id_kota	Integer	-	Id dari kota, foreign key ke tabel ref_kota

### 3.1.4 Deskripsi Entitas Data Wilayah

Nama	Tipe	Panjang	Keterangan
id_wilayah	Integer	-	Id wilayah, primary key
kode_wilayah	Varchar	50	Kode dari wilayah
nama_wilayah	Varchar	100	nama dari sebuah wilayah
is_deleted	Boolean	-	Apakah sebuah wilayah sudah dihapus atau belum

id_paroki	Integer	-	Id dari paroki, foreign key ke tabel paroki
-----------	---------	---	---

### 3.1.5 Deskripsi Entitas Data Stasi

Nama	Tipe	Panjang	Keterangan
id_stasi	Integer	-	Id stasi, primary key
kode_stasi	Varchar	50	Kode dari stasi
nama_stasi	Varchar	100	nama dari sebuah stasi
is_deleted	Boolean	-	Apakah sebuah stasi sudah dihapus atau belum
id_paroki	Integer	-	Id dari paroki, foreign key ke tabel paroki

### 3.1.6 Deskripsi Entitas Data Lingkungan

Nama	Tipe	Panjang	Keterangan
id_lingkungan	Integer	-	Id lingkungan, primary key
kode_lingkungan	Varchar	50	Kode dari stasi
nama_lingkungan	Varchar	100	nama dari sebuah stasi
is_deleted	Boolean	-	Apakah sebuah stasi sudah dihapus atau belum
id_wilayah	Integer	-	Id dari wilayah, foreign key ke tabel wilayah (bisa kosong)
Id_stasi	Integer	-	Id dari stasi, foreign key ke tabel stasi (bisa kosong)

### 3.1.7 Deskripsi Entitas Data Kepala Keluarga

Nama	Tipe	Panjang	Keterangan
id_kepala_keluarga	Integer	-	Id kepala keluarga, primary key
kode_kepala_keluarga	Varchar	50	Kode dari kepala keluarga
nama_kepala_keluarga	Varchar	100	nama dari kepala keluarga
Alamat_kepala_keluarga	Varchar	100	Alamat tempat tinggal kepala keluarga
No_telepon	Varchar	20	No telepon kepala keluarga
Tgl_nikah	Date	-	Tanggal kepala keluarga menikah diperbaharui

Last_update	Date	-	Tanggal kepala keluarga terakhir diperbaharui
Is_deleted	Boolean	-	Apakah seorang kepala keluarga sudah dihapus atau belum
Id_lingkungan	Integer	-	Id dari lingkungan, foreign key ke tabel lingkungan
Id_status_ekonomi	Integer	-	Id dari status ekonomi, foreign key ke tabel ref_status_ekonomi
Id_jenis_kepala_keluarga	Integer	-	Id dari jenis kepala keluarga, foreign key ke tabel ref_jenis_kepala_keluarga
Id_kota_kk	Integer	-	Id dari kota tempat tinggal kepala keluarga, foreign key ke tabel ref_kota
Id_kota_nikah	Integer	-	Id dari kota menikah, foreign key ke tabel lingkungan
Id_kota_pindah_dari	Integer	-	Id dari kota asal, foreign key ke tabel ref_kota
Id_kota_pindah_ke	Integer	-	Id dari kota pindah, foreign key ke tabel ref_kota

### 3.1.8 Deskripsi Entitas Data Umat

Nama	Tipe	Panjang	Keterangan
id_umat	Integer	-	Id umat, primary key
kode_umat	Varchar	50	Kode dari umat
nama_umat	Varchar	100	nama dari umat
Tgl_lahir	Date	-	Tanggal lahir dari umat
Tgl_baptis	Date	-	Tanggal baptis dari umat
Tgl_krisma	Date	-	Tanggal krisma dari

			umat
Lama_tinggal	Integer	-	Lama tinggal dari umat
Last_update	Date	-	Tanggal kepala keluarga terakhir diperbaharui
Is_deleted	Boolean	-	Apakah seorang kepala keluarga sudah dihapus atau belum
Id_kepala_keluarga	Integer	-	Id dari kepala keluarga, foreign key ke tabel kepala keluarga
Id_jenis_kelamin	Integer	-	Id dari jenis kelamin, foreign key ke tabel ref_jenis_kelamin
Id_hubungan_kepala_keluarga	Integer	-	Id dari hubungan kepala keluarga, foreign key ke tabel ref_hubungan_kepala_keluarga
Id_suku	Integer	-	Id dari suku, foreign key ke tabel ref_suku
Id_status_pendidikan	Integer	-	Id dari status pendidikan, foreign key ke ref_status_pendidikan
Id_pendidikan_terakhir	Integer	-	Id dari pendidikan terakhir, foreign key ke tabel ref_pendidikan_terakhir
Id_bidang_studi	Integer	-	Id dari bidang

			studi, foreign key ke tabel ref_bidang_studi
Id_pekerjaan	Integer	-	Id dari pekerjaan, foreign key ke tabel ref_pekerjaan
Id_golongan_darah	Integer	-	Id dari golongan darah, foreign key ke tabel ref_golongan_darah
Id_status_baptis	Integer	-	Id dari status baptis, foreign key ke tabel ref_status_baptis
Id_status_kesehatan	Integer	-	Id dari status kesehatan, foreign key ke tabel ref_status_kesehatan
Id_status_keluarga	Integer	-	Id dari status keluarga, foreign key ke tabel ref_status_keluarga
Id_jabatan_paroki	Integer	-	Id dari jabatan paroki, foreign key ke tabel ref_jabatan_paroki
Id_status_gereja	Integer	-	Id dari status gereja, foreign key ke tabel ref_status_gereja
Id_tempat_tinggal	Integer	-	Id dari tempat tinggal, foreign key ke tabel ref_kota
Id_agama	Integer	-	Id dari agama, foreign key ke

			tabel ref_agama
Id_status_sosial	Integer	-	Id dari status sosial, foreign key ke tabel ref_status_sosial
Id_status_perkawinan	Integer	-	Id dari status perkawinan, foreign key ke tabel ref_status_perkawinan
Id_kota_lahir	Integer	-	Id dari kota tempat kelahiran, foreign key ke tabel ref_kota
Id_kota_baptis	Integer	-	Id dari kota tempat baptis, foreign key ke tabel ref_kota
Id_kota_krisma	Integer	-	Id dari kota tempat krisma, foreign key ke tabel ref_kota

### 3.1.9 Deskripsi Entitas Data Ref\_Kota

Nama	Tipe	Panjang	Keterangan
id_kota	Integer	-	Id kota, primary key
Nama_kota	Varchar	50	nama dari kota

### 3.1.10 Deskripsi Entitas Data Ref\_Status\_ekonomi

Nama	Tipe	Panjang	Keterangan
id_status_ekonomi	Integer	-	Id status ekonomi, primary key
Status_ekonomi	Varchar	50	Nama dari sebuah status ekonomi

### 3.1.11 Deskripsi Entitas Data Ref\_Jenis\_Kepala\_keluarga

Nama	Tipe	Panjang	Keterangan
------	------	---------	------------

id_jenis_kepala_keluarga	Integer	-	Id jenis kepala keluarga, primary key
Jenis_kepala_keluarga	Varchar	50	Nama dari sebuah jenis kepala keluarga

### 3.1.12 Deskripsi Entitas Data Ref\_Jenis\_Kelamin

Nama	Tipe	Panjang	Keterangan
id_jenis_kelamin	Integer	-	Id dari jenis kelamin, primary key
Jensi_kelamin	Varchar	50	Nama dari sebuah jenis kelamin

### 3.1.13 Deskripsi Entitas Data Ref\_Suku

Nama	Tipe	Panjang	Keterangan
id_suku	Integer	-	Id suku, primary key
Suku	Varchar	50	Nama dari sebuah suku

### 3.1.14 Deskripsi Entitas Data Ref\_Hubungan\_Kepala\_keluarga

Nama	Tipe	Panjang	Keterangan
id_hubungan_kepala_keluarga	Integer	-	Id hubungan kepala keluarga, primary key
Hubungan_kepala_keluarga	Varchar	50	Nama dari sebuah hubungan kepala keluarga

### 3.1.15 Deskripsi Entitas Data Ref\_Status\_Pendidikan

Nama	Tipe	Panjang	Keterangan
id_status_pendidikan	Integer	-	Id status pendidikan, primary key
Status_pendidikan	Varchar	50	Nama dari sebuah status pendidikan

### 3.1.16 Deskripsi Entitas Data Ref Pendidikan Terakhir

Nama	Tipe	Panjang	Keterangan
id_pendidikan_terakhir	Integer	-	Id pendidikan terakhir, primary key
Pendidikan_terakhir	Varchar	50	Nama dari pendidikan terakhir

### 3.1.17 Deskripsi Entitas Data Ref Bidang Studi

Nama	Tipe	Panjang	Keterangan
id_bidang_studi	Integer	-	Id bidang studi, primary key
Bidang_studi	Varchar	50	Nama dari sebuah bidang studi

### 3.1.18 Deskripsi Entitas Data Ref Pekerjaan

Nama	Tipe	Panjang	Keterangan
id_pekerjaan	Integer	-	Id pekerjaan, primary key
pekerjaan	Varchar	50	Nama dari sebuah pekerjaan

### 3.1.19 Deskripsi Entitas Data Ref Status keluarga

Nama	Tipe	Panjang	Keterangan
id_status_keluarga	Integer	-	Id status keluarga, primary key
Status_keluarga	Varchar	50	Nama dari sebuah status_keluarga

### 3.1.20 Deskripsi Entitas Data Ref Status Baptis

Nama	Tipe	Panjang	Keterangan
id_status_baptis	Integer	-	Id status baptis, primary key
Status_baptis	Varchar	50	Status baptis dari umat

### 3.1.21 Deskripsi Entitas Data Ref Golongan Darah

Nama	Tipe	Panjang	Keterangan
id_golongan_darah	Integer	-	Id golongan darah,

			primary key
Golongan_darah	Varchar	50	Nama dari sebuah golongan darah

### 3.1.22 Deskripsi Entitas Data Ref\_Status\_Katolik

Nama	Tipe	Panjang	Keterangan
id_status_katolik	Integer	-	Id status katolik, primary key
Status_katolik	Varchar	50	Nama dari sebuah Status katolik umat

### 3.1.23 Deskripsi Entitas Data Ref\_Jabatan\_Paroki

Nama	Tipe	Panjang	Keterangan
id_jabatan_paroki	Integer	-	Id jabatan paroki, primary key
Jabatan_paroki	Varchar	50	Nama jabatan di paroki

### 3.1.24 Deskripsi Entitas Data Ref\_Status\_Gereja

Nama	Tipe	Panjang	Keterangan
id_status_gereja	Integer	-	Id status gereja, primary key
Status_gereja	Varchar	50	Nama sebuah status gereja

### 3.1.25 Deskripsi Entitas Data Ref\_Tempat\_Tinggal

Nama	Tipe	Panjang	Keterangan
id_tempat_tinggal	Integer	-	Id tempat tinggal, primary key
Tempat_tinggal	Varchar	50	Nama tempat tinggal

### 3.1.26 Deskripsi Entitas Data Ref\_Status\_Sosial

Nama	Tipe	Panjang	Keterangan
id_status_sosial	Integer	-	Id status sosial, primary key
Status_sosial	Varchar	50	Nama dari sebuah status sosial

### 3.1.27 Deskripsi Entitas Data Ref\_Agama

Nama	Tipe	Panjang	Keterangan
id_status_baptis	Integer	-	Id status baptis, primary key
Status_baptis	Varchar	50	Status baptis dari umat

### 3.1.28 Deskripsi Entitas Data Ref\_Status\_Perkawinan

Nama	Tipe	Panjang	Keterangan
id_status_perkawinan	Integer	-	Id status perkawinan, primary key
Status_perkawinan	Varchar	50	Nama status perkawinan dari umat

### 3.1.29 Deskripsi Entitas Data Ref\_Status\_Kesehatan

Nama	Tipe	Panjang	Keterangan
id_status_kesehatan	Integer	-	Id status kesehatan, primary key
Status_kesehatan	Varchar	50	Nama dari sebuah status kesehatan

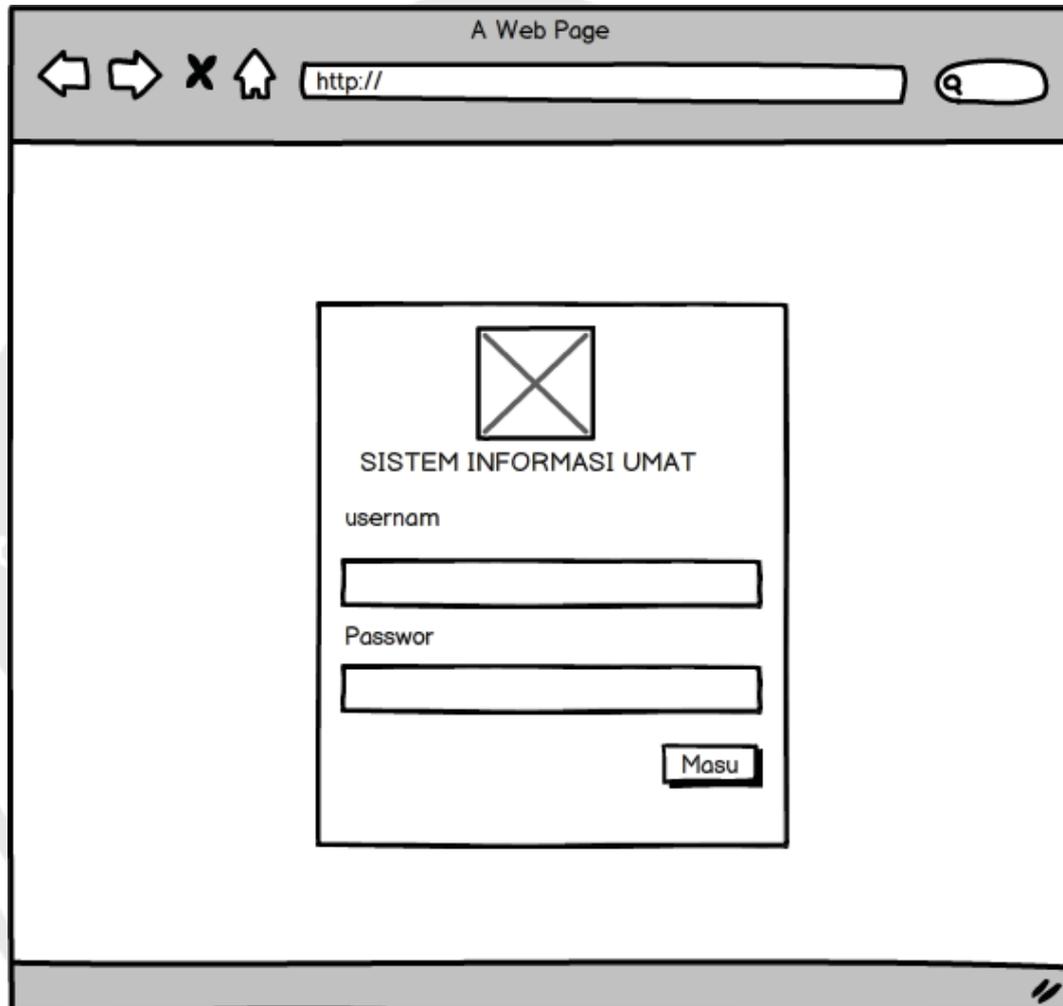
### 3.1.30 Deskripsi Entitas Data Role

Nama	Tipe	Panjang	Keterangan
id_role	Integer	-	Id role, primary key
Role	Varchar	25	Nama dari sebuah status kesehatan



## 4 Perancangan Antarmuka

### 4.1 Rancangan Antarmuka Login

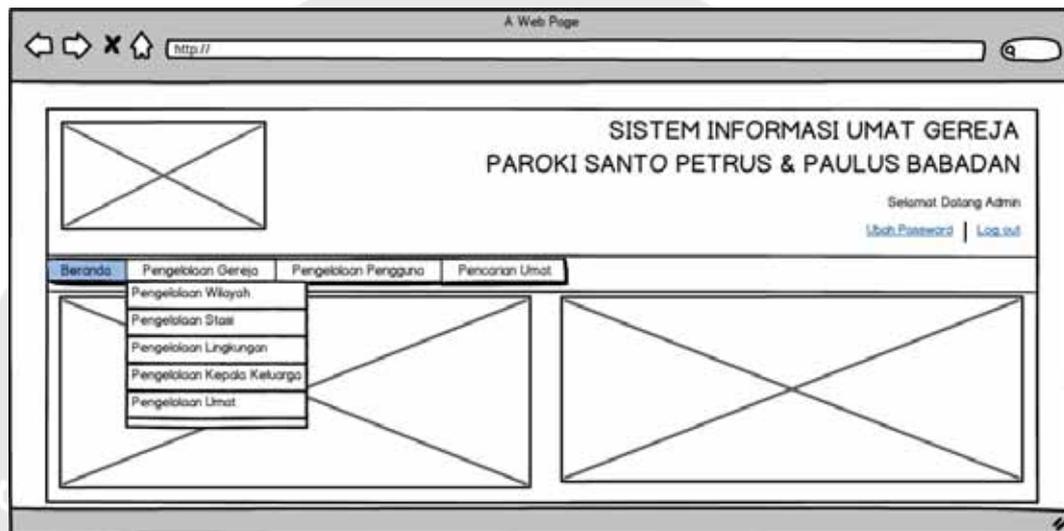


Gambar 4.1 Rancangan Antarmuka Login

Rancangan antarmuka ini digunakan untuk melakukan login ke dalam SIMAG. Untuk mendapat akses ke dalam sistem, user harus memasukkan username dan password pada textbox yang telah disediakan, kemudian menekan tombol Masuk. Saat tombol masuk di tekan maka sistem akan mengecek apakah username dan password sudah sesuai dari database. Bila data yang diinputkan user tidak sesuai maka sistem akan memberikan peringatan.

Bila tombol masuk di klik : *"Select username,password from pengguna where username = txtuser.text and password txtpass.text"*

#### 4.2 Rancangan Antarmuka Menu



**Gambar 4.2 Rancangan Antarmuka Menu**

Rancangan antarmuka seperti yang tertampil pada gambar 4.2 merupakan rancangan antarmuka setelah login masuk ke dalam sistem sebagai admin. Menu Pengelolaan gereja memiliki sub menu pengelolaan wilayah, pengelolaan stasi, pengelolaan lingkungan, pengelolaan kepala keluarga dan pengelolaan umat.

Menu Laporan terdiri dari sub menu laporan jumlah umat berdasarkan pekerjaan, laporan jumlah umat wilayah tiap lingkungan, laporan jumlah umat stasi tiap lingkungan dan laporan kk.

Bila user memilih ubah password maka akan masuk ke antarmuka ubah password, bila user memilih menu log out maka akan kembali ke antarmuka login.

Bila user memilih pengelolaan wilayah maka akan masuk ke antar muka pengelolaan wilayah, bila user memilih pengelolaan stasi maka akan masuk ke antar muka

pengelolaan stasi, bila user memilih pengelolaan lingkungan maka akan masuk ke antarmuka pengelolaan lingkungan, bila user memilih pengelolaan kepala keluarga maka akan masuk ke antarmuka pengelolaan kepala keluarga, bila user memilih pengelolaan umat maka akan masuk ke pengelolaan umat.

Bila user memilih pengelolaan pengguna maka akan masuk ke antarmuka pengelolaan pengguna, bila user memilih pencarian umat maka akan masuk ke pengelolaan pencarian umat.

#### 4.3 Rancangan Antarmuka Pengelolaan Wilayah

The screenshot shows a web browser window titled 'SIMAG'. The browser's address bar is empty. The page content is organized into two main sections. On the left, there is a form titled 'Pengelolaan Wilayah'. It contains four input fields: a dropdown menu for 'Pengelolaan Wilayah', a combobox for 'Nama Paroki', a text input for 'Kode Wilayah', and another text input for 'Nama Wilayah'. Below these fields are two buttons: 'Tambah' and 'Batal'. On the right, there is a search section with a text input labeled 'Cari' and a 'Cari' button. Below the search section is a table with four columns: 'Ubah', 'Hapus', 'Kode Wilayah', and 'Nama Wilayah'. The first two columns contain blue links for 'Ubah' and 'Hapus'.

Gambar 4.3 Rancangan Antarmuka Pengelolaan Wilayah

Rancangan antarmuka seperti tertampil pada gambar 4.3 digunakan oleh user untuk melakukan pengelolaan wilayah. User harus memasukkan data pada field-field yang telah disediakan yaitu nama paroki, kode wilayah dan nama wilayah. Lalu terdapat textbox cari untuk melakukan pencarian data wilayah sesuai dengan kata kunci yang diketikkan.

Di antarmuka ini terdapat 2 tombol untuk melakukan pengelolaan yaitu tambah dan batal. Tambah dilakukan untuk menambahkan data wilayah, batal digunakan untuk membatalkan transaksi. Di data tabel terdapat link ubah dan hapus. Saat link ubah diklik maka data akan tertampil dan tombol tambah menjadi perbaharui data. Lalu link hapus digunakan untuk menghapus data di baris tertentu.

Bila tombol tambah diklik : `insert into wilayah (kode_wilayah, nama_wilayah, is_deleted, id_paroki) values (@kode_wilayah, @nama_wilayah, @is_deleted, @id_paroki)`

Bila tombol perbaharui diklik : `update wilayah set kode_wilayah = @kode_wilayah, nama_wilayah = @nama_wilayah, id_paroki = @id_paroki where id_wilayah = @original_id_wilayah`

Bila tombol link hapus diklik : `update wilayah set is_deleted = true where (id_wilayah = @original_id_wilayah)`

#### 4.4 Rancangan antarmuka pengelolaan Stasi

The screenshot shows a web browser window titled 'SIMAG'. The main content area is titled 'Pengelolaan Stasi'. It features a search bar with a 'Cari' button. Below the search bar, there are three input fields: 'Nama Paroki' (a dropdown menu), 'Kode Stasi', and 'Nama Stasi'. At the bottom of the form area are 'Tambah' and 'Batal' buttons. To the right of the form is a table with four columns: 'Ubah', 'Hapus', 'Kode Stasi', and 'Nama Stasi'. The first two rows of the table have blue links for 'Ubah' and 'Hapus' in the 'Ubah' and 'Hapus' columns respectively. The table has several empty rows below.

Gambar 4.4 Rancangan Antarmuka Pengelolaan stasi

Rancangan antarmuka seperti tertampil pada gambar 4.3 digunakan oleh user untuk melakukan pengelolaan stasi. User

harus memasukkan data pada field-field yang telah disediakan yaitu nama paroki, kode stasi dan nama stasi. Lalu terdapat textbox cari untuk melakukan pencarian data stasi sesuai dengan kata kunci yang diketikkan.

Di antarmuka ini terdapat 2 tombol untuk melakukan pengelolaan yaitu tambah dan batal. Tambah dilakukan untuk manambah data stasi, batal digunakan untuk membatalkan transaksi. Di data tabel terdapat link ubah dan hapus. Saat link ubah diklik maka data akan tertampil dan tombol tambah menjadi perbaharui data. Lalu link hapus digunakan untuk menghapus data di baris tertentu.

Bila tombol tambah diklik : `insert into stasi (id_paroki, kode_stasi, nama_stasi, is_deleted) values (@id_paroki, @kode_stasi, @nama_stasi, false)`

Bila tombol perbaharui data diklik :

`update stasi set id_paroki = @id_paroki, kode_stasi = @kode_stasi, nama_stasi = @nama_stasi where id_stasi = @original_id_stasi`

Bila link hapus diklik : `update stasi set is_deleted = true where (id_stasi= @original_id_stasi)`

#### 4.5 Rancangan Antarmuka Pengelolaan Lingkungan

The screenshot shows a web browser window titled 'SMAAD'. The main content area is titled 'Pengelolaan Lingkungan'. On the left, there is a form with the following fields: 'Wilayah Atasi Stasi' (a dropdown menu), 'Nama' (a text input), 'Kode Lingkungan' (a text input), and 'Nama Lingkungan' (a text input). Below these fields are two buttons: 'Tambah' and 'Batal'. On the right, there is a search section with a 'Cari' label, a text input, and a 'Cari' button. Below the search section is a table with two columns: 'Ubah' and 'Hapus'. The table has several rows, but they are mostly empty or faded. At the top right of the content area, there are two links: 'Data lingkungan di wilayah' and 'Data lingkungan di stasi'.

Gambar 4.5 Rancangan Antarmuka Pengelolaan Lingkungan  
Rancangan antarmuka seperti yang ditampilkan di

gambar 4.5 digunakan oleh user untuk melakukan

pengelolaan stasi. User harus mengisikan data di textfield yang tersedia seperti wilayah atau stasi, nama lingkungan , kode lingkungan dan nama lingkungan. Terdapat pula textbox cari yang digunakan untuk melakukan pencarian data sesuai dengan kata kunci yang diketikkan.

Pada rancangan antarmuka ini terdapat 2 tombol yang digunakan untuk melakukan pengelolaan yaitu tambah dan batal. Tombol tambah digunakan untuk melakukan penambahan data lingkungan dengan mengisikan terlebih dahulu data-data di field yang tersedia. Tombol batal digunakan untuk membatalkan transaksi pengelolaan. Kemudian terdapat link ubah dan hapus di data tabel. Saat link ubah di klik maka data lingkungan akan ditampilkan dan tombol tambah akan berubah menjadi perbaharui data. Lalu saat link delete diklik maka akan menghapus data di baris tertentu.

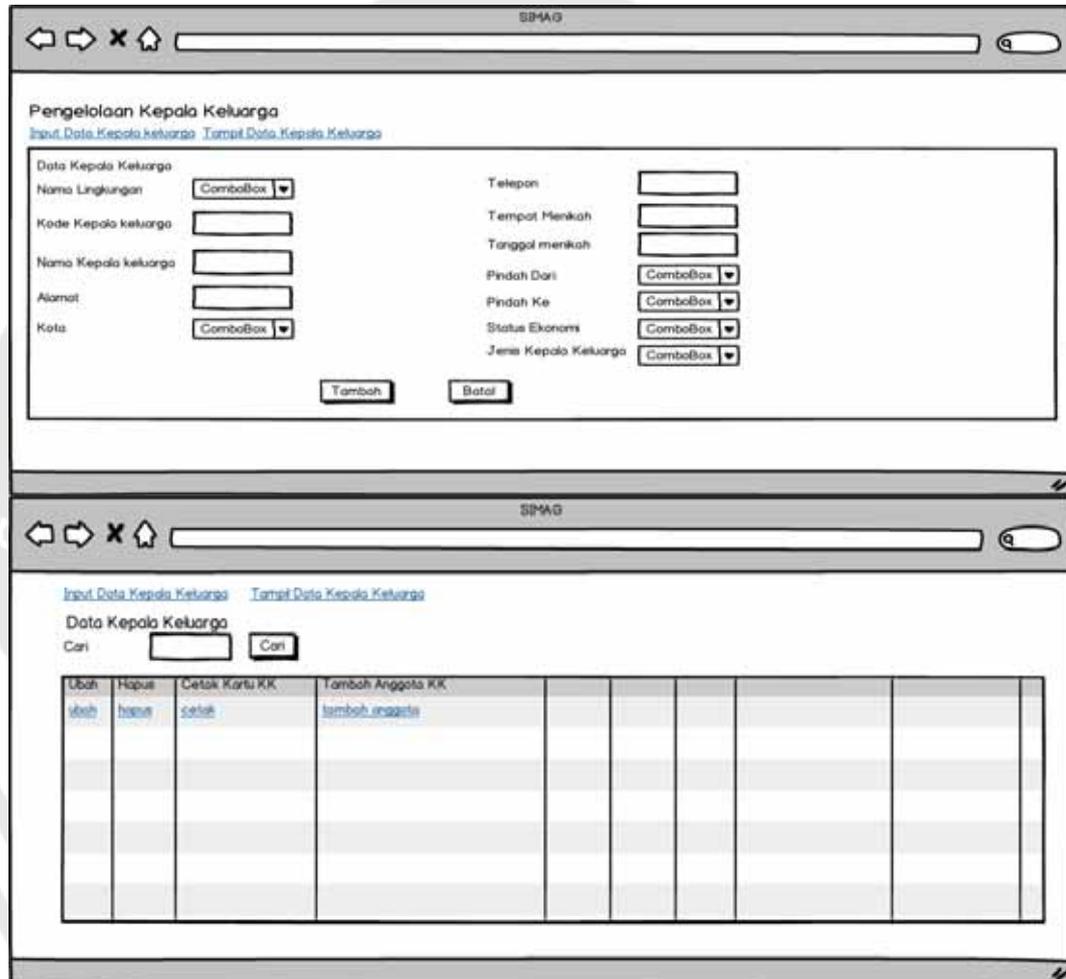
Bila tombol tambah diklik dan radio button wilayah dipilih : `"insert into lingkungan (id_wilayah, kode_lingkungan, nama_lingkungan, is_deleted) values (@id_wilayah, @kode_lingkungan, @nama_lingkungan, @is_deleted)"`

Bila tombol tambah diklik dan radio button stasi dipilih : `"insert into lingkungan ( kode_lingkungan, nama_lingkungan, is_deleted, id_stasi) values (@kode_lingkungan, @nama_lingkungan, @is_deleted, @id_stasi)"`

Bila tombol perbaharui data diklik : `"update lingkungan set id_wilayah = @id_wilayah, kode_lingkungan = @kode_lingkungan, nama_lingkungan =`

@nama\_lingkungan, id\_stasi = null where id\_lingkungan = @original\_id\_lingkungan"

#### 4.6 Rancangan Antarmuka Pengelolaan Kepala Keluarga



Gambar 4.6 Rancangan Antarmuka Pengelolaan Kepala Keluarga

Rancangan antarmuka pada gambar 4.6 ini di atas digunakan oleh user untuk melakukan pengelolaan kepala keluarga. Di antarmuka ini terdapat 2 buah link yaitu input data kepala keluarga dan tampil data kepala keluarga. Saat user memilih input data kepala keluarga maka user bisa melakukan penambahan data dengan mengisi data-data di text field yang telah

disediakan kemudian menekan tombol tambah. Tombol batal digunakan untuk membatalkan transaksi pengelolaan.

Apabila user memilih link tampil data maka akan muncul data kepala keluarga. Di data tabel terdapat link yaitu ubah, hapus, cetak kartu kepala keluarga, dan tambah data kepala keluarga. Saat link ubah dipilih maka data kepala keluarga terpilih akan ditampilkan dan tombol tambah akan menjadi perbaharui data. Lalu bila link hapus terpilih maka akan menghapus data terpilih. Bila link cetak kartu kepala keluarga terpilih maka akan mencetak kartu kepala keluarga. Bila link tambah kepala keluarga terpilih maka akan masuk ke rancangan antarmuka kepala keluarga.

```
Saat tombol tambah diklik: "Insert into
kepala_keluarga (id_lingkungan, kode_kepala_keluarga,
nama_kepala_keluarga, alamat_kepala_keluarga, no_telepon,
tgl_nikah, last_update, is_deleted, id_kota_kk, id_kota_nik
ah, id_kota_pindah_dari, id_kota_pindah_ke, id_jenis_kepal
a_keluarga, id_status_ekonomi) values (@idling,
@kodekk, namakk, @alamat, @tlpn, @tglnikah,
@lastUpdate, @isdel, @idkotakk, @idkotanikah,
@idkotapindahdari, @idkotapindahke, @idjeniskk,
@idstaeko)"
```

saat tombol "perbaharui data diklik:

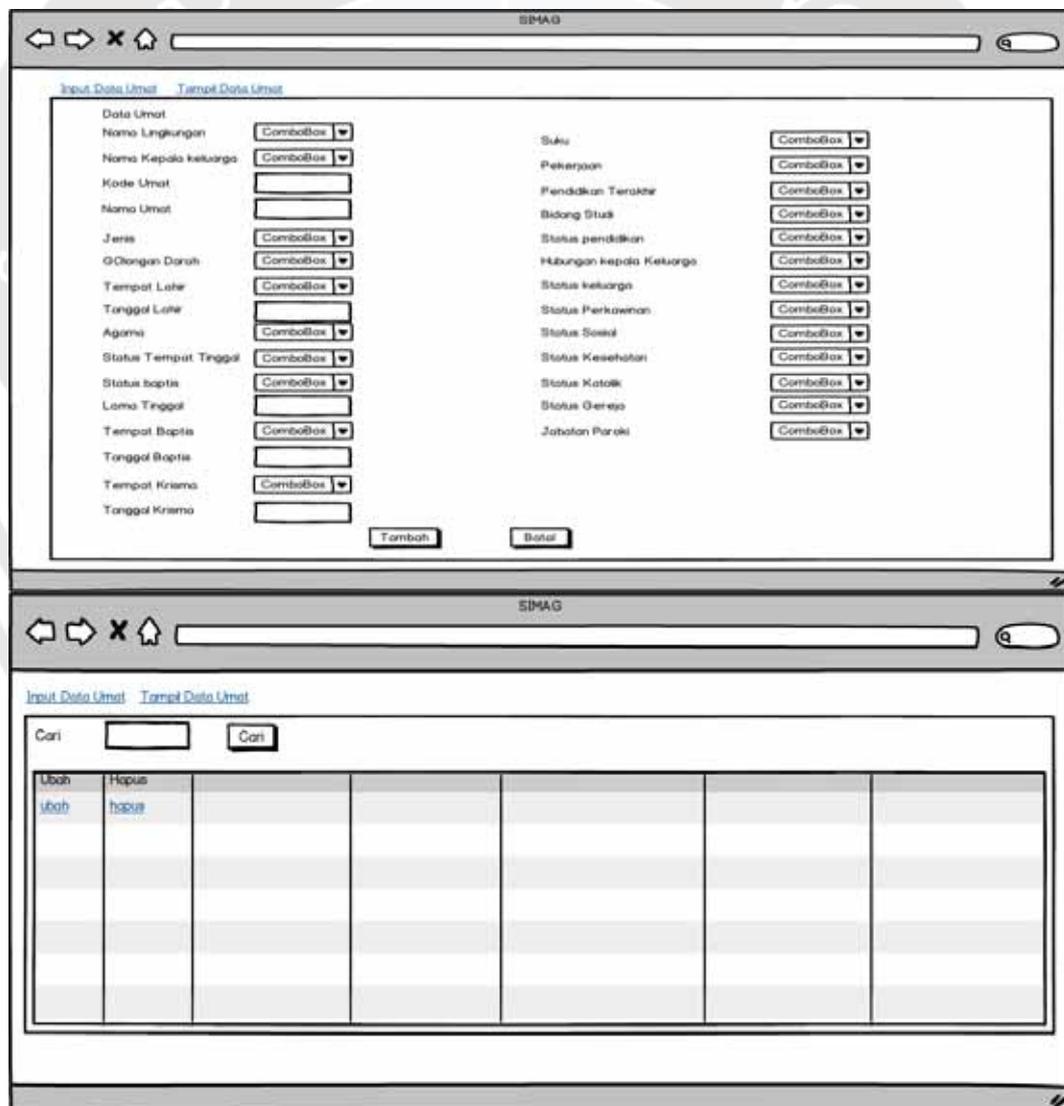
```
"update [kepala_keluarga] set [id_lingkungan]=
@idling, [kode_kepala_keluarga] = @kodekk ,
[nama_kepala_keluarga]=@namakk, [alamat_kepala_keluarga]
=@alamat, [no_telepon] = @tlpn, [tgl_nikah]=@tglnikah,
[last_update]=@lastupdate, [is_deleted]=@isdel,
[id_kota_kk]=@idkotakk, [id_kota_nikah]=@idkotanikah,
[id_kota_pindah_dari]=@idkotapindahdari,
```

Program Studi Teknik Informatika	DPPL – SIMAG	58/ 64
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		

```
[id_kota_pindah_ke]=@idkotapindahke,
[id_jenis_kepala_keluarga]=@idjeniskk,
[id_status_ekonomi]=@idstae
where ([id_kepalakeluarga] = @idkk)"
```

Saat link "hapus" diklik: "update kepala\_keluarga Set is\_deleted = true where (id\_kepalakeluarga = @original\_id\_kepalakeluarga)"

#### 4.7 Rancangan Antarmuka Pengelolaan Umat



Gambar 4.7 Rancangan Antarmuka Pengelolaan Umat

Rancangan antarmuka pada gambar 4.7 ini di atas digunakan oleh user untuk melakukan pengelolaan kepala keluarga. Di antarmuka ini terdapat 2 buah link yaitu input data umat dan tampil data umat. Saat user memilih input data umat maka user bisa melakukan penambahan data dengan mengisi data-data di text field yang telah disediakan kemudian menekan tombol tambah. Tombol batal digunakan untuk membatalkan transaksi pengelolaan.

Apabila user memilih link tampil data maka akan muncul data umat. Di data tabel terdapat link yaitu ubah dan hapus. Saat link ubah dipilih maka data kepala keluarga terpilih akan ditampilkan dan tombol tambah akan menjadi perbaharui data. Lalu bila link hapus terpilih maka akan menghapus data terpilih.

Saat tombol Tambah diklik: *"Insert into Umat (id\_kepalakeuarga,id\_jenis\_kelamin,id\_kota\_lahir,id\_agama,id\_tempat\_tinggal,id\_golongan\_darah,id\_status\_baptis,id\_kota\_baptis,id\_kota\_krisma,id\_hubungan\_kepala\_keluarga,id\_status\_keluarga,id\_suku,id\_pekerjaan,id\_bidang\_studi,id\_pendidikan\_terakhir,id\_status\_pendidikan,id\_status\_perkawinan,id\_status\_sosial,id\_status\_kesehatan,id\_status\_katolik,id\_status\_gereja,id\_jabatan\_paroki,kode\_umat,nama\_umat,lama\_tinggal,is\_deleted,liberbab,tgl\_lahir,tgl\_baptis,tgl\_krisma) values (@idkk, @jk, @idTempatLahir, @idagama, @idStaTmptTinggal, @idGolDarah, @idStaBaptis, @idTmptBaptis, @idTmptKrisma, @idHubKK, @idStaKeluarga, @idSuku, @idpekerjaan, @idbidangSTudi, @idPendidikanTerakhir, @idStaPendidikan, @idStaPerkawinan, @idStaSosial,@idStaKesehatan,*

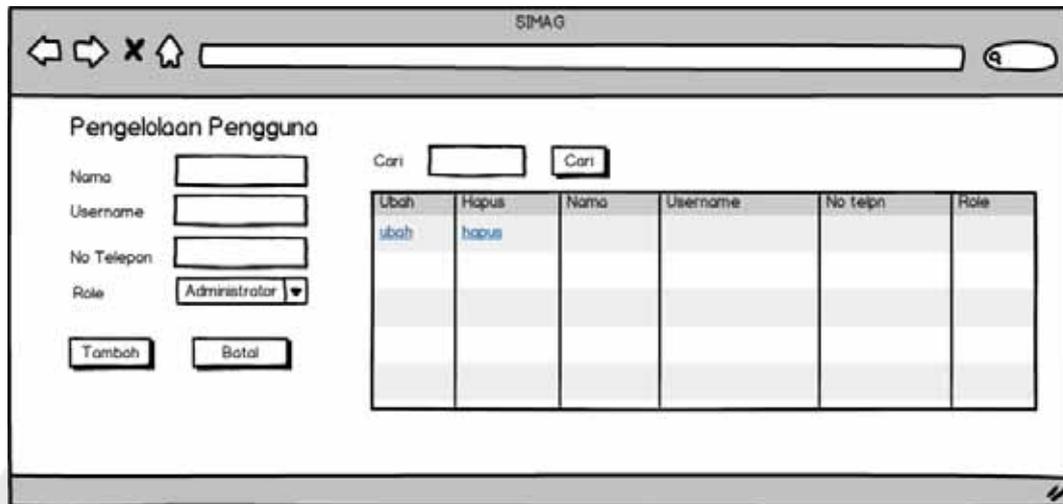
Program Studi Teknik Informatika	DPPL – SIMAG	60/ 64
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

```
@idStaKatolik,@idStaGereja, @idJabatanParoki,@kode,  
@nama, @lamaTinggal, @isdel, @last_update, @tglLahir,  
@tglBaptis, @tglKrisma)"
```

```
Saat tombol "Perbaharui Data" diklik:  
"update Umat set id_kepalakeluarga =@idkk ,  
id_jenis_kelamin=@jkk, id_kota_lahir =  
@idTempatLahir,id_agama = @idagama, id_tempat_tinggal=  
@idStaTmptTinggal,id_golongan_darah=@idGolDarah,id_stat  
us_baptis =@idStaBaptis, id_kota_baptis =  
@idTmptBaptis, id_kota_krisma = @idTmptKrisma,  
id_hubungan_kepala_keluarga = @idHubKK,  
id_status_keluarga = @idStaKeluarga, id_suku = @idSuku,  
id_pekerjaan = @idpekerjaan, id_bidang_studi =  
@idbidangSTudi, id_pendidikan_terakhir =  
@idPendidikanTerakhir, id_status_pendidikan =  
@idStaPendidikan, id_status_perkawinan =  
@idStaPerkawinan, id_status_sosial = @idStaSosial,  
id_status_kesehatan=@idStaKesehatan,id_status_katolik  
=@idStaKatolik, id_status_gereja = @idStaGereja,  
id_jabatan_paroki =@idJabatanParoki, kode_umat =@kode,  
nama_umat =@nama, lama_tinggal = @lamaTinggal, liberbab  
=@last_update, tgl_lahir = @tglLahir, tgl_baptis  
=@tglBaptis, tgl_krisma=@tglKrisma where  
Id_umat=@idUmat"
```

```
Saat link "Hapus" diklik: "update umat set  
is_deleted = true where (id_umat = @original_id_umat)"
```

#### 4.8 Rancangan Antarmuka Pengelolaan Pengguna



Gambar 4.8 Rancangan Antarmuka Pengelolaan Pengguna

Rancangan antarmuka seperti tertampil pada gambar 4.8 digunakan oleh user untuk melakukan pengelolaan pengguna. User harus memasukkan data pada field-field yang telah disediakan yaitu nama pengguna, username, no telepon, dan role. Lalu terdapat textbox cari untuk melakukan pencarian data wilayah sesuai dengan kata kunci yang diketikkan.

Di antarmuka ini terdapat 2 tombol untuk melakukan pengelolaan yaitu tambah dan batal. Tambah dilakukan untuk menambahkan data wilayah, batal digunakan untuk membatalkan transaksi. Di data tabel terdapat link ubah dan hapus. Saat link ubah diklik maka data akan tertampil dan tombol tambah menjadi perbaharui data. Lalu link hapus digunakan untuk menghapus data di baris tertentu.

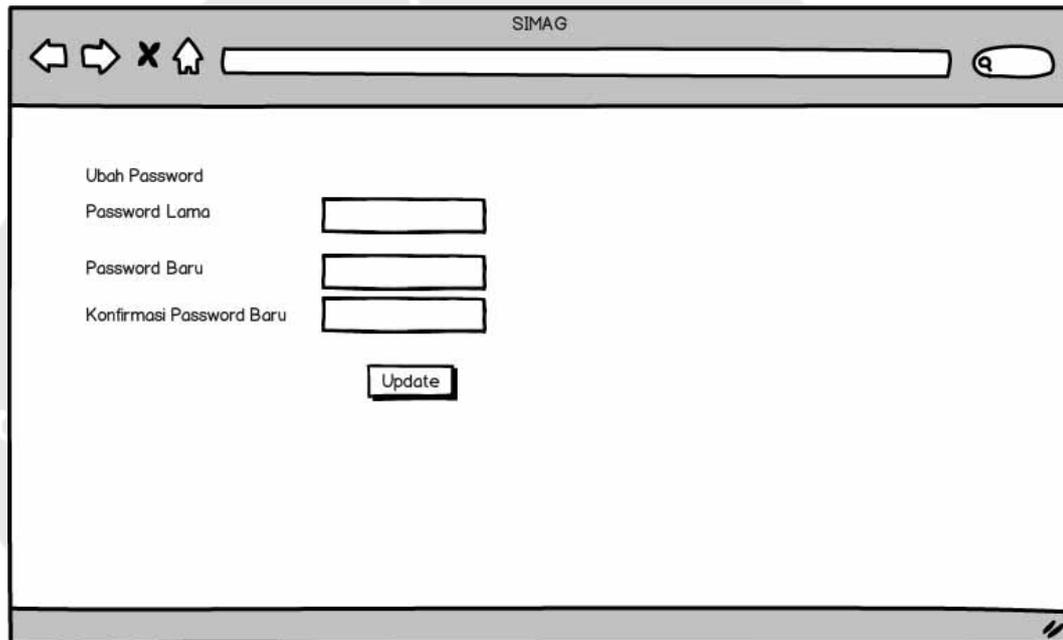
Saat tombol "Tambah" diklik: *"insert into pengguna (nama, username, password, no\_telepon, id\_role) values (@nama, @username, @password, @no\_telepon, @id\_role)"*

Saat Tombol "Perbaharui Data" diklik: *"update pengguna set nama = @nama, username = @username,*

```
no_telepon = @no_telepon, id_role = @role where  
(id_pengguna = @id_pengguna)"
```

Saat link Hapus diklik: "delete from pengguna where  
((id\_pengguna = @id\_pengguna))"

#### 4.9 Rancangan Antarmuka Ubah Password



The image shows a web browser window with the title "SIMAG". The browser's address bar is empty. The main content area displays a form titled "Ubah Password". The form contains three text input fields: "Password Lama", "Password Baru", and "Konfirmasi Password Baru". Below these fields is a button labeled "Update". The browser's navigation icons (back, forward, stop, home) and a search icon are visible in the top left corner of the browser window.

**Gambar 4.9 Rancangan Antarmuka Pengelolaan Pengguna**

Rancangan antarmuka 4.9 merupakan rancangan antarmuka untuk ubah password. Pengguna diminta untuk mengisikan password lama, password baru dan konfirmasi password baru di text box yang telah disediakan. Setelah semua diisi kemudian pengguna menekan tombol "update" untuk menyimpan perubahan di basis data.

Saat tombol "Update" diklik: "update pengguna  
Set password = @password where (id\_pengguna =  
@original\_id\_pengguna)

#### 4.10 Rancangan Antarmuka Pencarian Umat

The screenshot shows a web browser window titled "SIMAG". The address bar is empty. Below the browser window, there is a search form. The form contains three criteria selection sections: "Kriteria pertama" with a dropdown menu showing "Kriteria 1", "Kriteria kedua" with a dropdown menu showing "Kriteria 2", and "kriteria ketiga" with a dropdown menu showing "Kriteria 3". Each criterion section has a "Value" input field. Between the criteria sections are checkboxes for logical operators: "dan" and "atau". Below the criteria sections, there are input fields for "umur" (age) and "sampai" (until), followed by a "Cari" button and a "clear" button. To the right, there are checkboxes for "pilih file" and "pdf", and an "export" button. The main content area below the form is currently empty, showing only horizontal lines representing a table structure.

**Gambar 4.10 Rancangan Antarmuka Pencarian Umat**

Gambar 4.10 merupakan rancangan antarmuka pencarian umat. Pengguna dapat mencari umat berdasarkan dengan kriteria yang ada. Pertama-tama memilih terlebih dahulu kriteria yang diinginkan kemudian mengisi value yang ingin dicari. Untuk mencari umat maka menekan tombol "Cari". Data yang dicari akan tertampil. Data yang ditampilkan dapat diexport ke pdf atau ke excel dengan memilih terlebih dahulu format yang diinginkan kemudian menekan tombol "export"