

BAB VI

PENUTUP

VI.1 Kesimpulan

Berdasarkan pembahasan pada bab-bab sebelumnya dan pengujian langsung pada pengguna, maka dapat ditarik kesimpulan yaitu:

1. Sistem Informasi Manajemen Gereja (SIManja) dan Web Service untuk sinkronisasi kepala keluarga dan umat berhasil dikembangkan.
2. SIManja dapat digunakan untuk mempermudah memprofil umat dengan fasilitas cari yang disediakan.
3. SIManja berhasil melakukan sinkronisasi data kepala keluarga dan umat ke pusat (keuskupan) melalui web service dari paroki-paroki yang tersebar.
4. Web Service SIManja dapat melakukan sinkronisasi yaitu perubahan data yang ada berdasarkan tanggal terakhir diubah dan penambahan data ke pusat bila data belum ada di pusat.

VI.2 Saran

Saran yang dapat diambil dari proses analisis sampai pada pembuatan SIManja adalah sebagai berikut:

1. Sistem diharapkan bisa dikembangkan lebih lanjut dengan menambahkan beberapa fungsionalitas yang dapat membuat sistem menjadi lebih baik, seperti menambahkan pengelolaan baptis, pengelolaan krisma, dan pengelolaan pernikahan.
2. Sistem diharapkan bisa dikembangkan lebih lanjut dengan menambahkan proses sinkronisasi untuk menyamakan data-data selain data umat sehingga tidak terjadi perbedaan di lokal dan di pusat. Data-data yang bisa dilakukan proses sinkronisasi dari pusat ke lokal seperti data paroki, wilayah, stasi, dan lingkungan.

3. Sistem diharapkan bisa menjamin sisi keamanan data dari pihak ke-3 yang tidak berkepentingan, supaya data-data yang diubah menjadi lebih *valid*.



DAFTAR PUSTAKA

- Anindyojati, Himawan. 2007. Pembangunan Perangkat Lunak Manajemen Perkuliahan untuk Mobile Device dengan Arsitektur *Disconnected* menggunakan teknologi .Net, Tugas Akhir, Jurusan Teknik Informatika, Universitas Atma Jaya Yogyakarta.
- Ferdinand, Magaline. 2009. Konsep-konsep Dasar Sistem Informasi.
- Lianto, Ricky. 2010. Pembangunan Aplikasi Simulasi Kredit Berbasis Web Service, Tugas Akhir, Jurusan Teknik Informatika, Universitas Atma Jaya Yogyakarta.
- Powell, Gavin. 2006. *Beginning Database Design*. IEEE Computer Society. Indiana. Vol 4, pp. 580-584.
- Sunardi. Joko, Sutanto, Eko Prihantono. Singgih., 2009. "Rancang Bangun Antarmuka Mikrokontroler Atmega32 Dengan *Multimedia Card*". *Elektronika Instrumentasi*, Jurusan Teknofisika Nuklir, STTN-BATAN.
- Turban Efraim, Volonino Linda. 2010. *Information Technology for Management*. John Wiley & Sons (Asia) Pte Ltd.
- Utama Yadi, S.Kom, M.Kom. 2010. Teknik Pemrograman Web Service PHP Dengan Menggunakan SOAP dan WSDL.
- 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.
- Wijayanti, Ni Made. 2009. Sistem Informasi Traveling Berbasis Internet dengan Teknologi XML (Integrasi Travel, Hotel, dan Perusahaan Penerbangan), Tugas Akhir, Jurusan Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Teknik Komputer Bali.
- Wisanggeni, Adhi Wibowo. 2011. Pembangunan Sistem Informasi E-Travel Dengan Integrasi Web Service. Tugas Akhir,

Jurusan Teknik Informatika, Universitas Atma Jaya
Yogyakarta.

www.codeproject.com

www.codeproject.com/KB/database/Cs_CSV_import_export.aspx

www.codeproject.com/KB/webservices/Transfer_any_files_on_Web.aspx

www.dfpug.de

www.imankatolik.or.id

www.indonesiavatican.it



SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

**Sistem Informasi Manajemen Gereja
(SIManja)**

Untuk:

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Jeffrey Soeprapto / 07 07 05232

**Program Studi Teknik Informatika - Fakultas Teknologi
Industri
Universitas Atma Jaya Yogyakarta**

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		SKPL-SIManja		1/ 31
	Fakultas Teknologi Industri	Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

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

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan.....	5
1.1	Tujuan	5
1.2	Lingkup Masalah	5
1.3	Definisi, Akronim dan Singkatan	5
1.4	Referensi	6
1.5	Deskripsi umum (Overview)	6
2	Deskripsi Kebutuhan.....	7
2.1	Perspektif produk	7
2.2	Fungsi Produk	8
2.3	Karakteristik Pengguna	12
2.4	Batasan-batasan	12
2.5	Asumsi dan Ketergantungan	13
3.	Kebutuhan khusus	13
3.1	Kebutuhan antarmuka eksternal	13
3.2	Kebutuhan fungsionalitas Perangkat Lunak	15
4.	Spesifikasi Rinci Kebutuhan.....	15
4.1	Spesifikasi Kebutuhan Fungsionalitas	15
5.	Entity Relationship Diagram (ERD)	31

1 Pendahuluan

1.1 Tujuan

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

1.2 Lingkup Masalah

Perangkat Lunak SIManja 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 sinkronisasi ke pusat.

Dan berjalan pada lingkungan dengan platform Windows XP.

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-SIManja-XXX	Kode yang merepresentasikan kebutuhan pada SIManja (Sistem Informasi Manajemen Gereja) dimana XXX merupakan nomor

Program Studi Teknik Informatika	SKPL – SIManja	5/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

	fungsi produk.
SIManja	Perangkat lunak yang menangani masalah pengelolaan umat disebuah gereja.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. SKPL dan DPPL Jeffrey Soeprapto
2. Software Development planning SIManja
3. Template SKPL berbasis OO

1.5 Deskripsi umum (Overview)

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

Bagian kedua berisi penjelasan umum tentang perangkat lunak SIManja 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 SIManja tersebut.

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

2 Deskripsi Kebutuhan

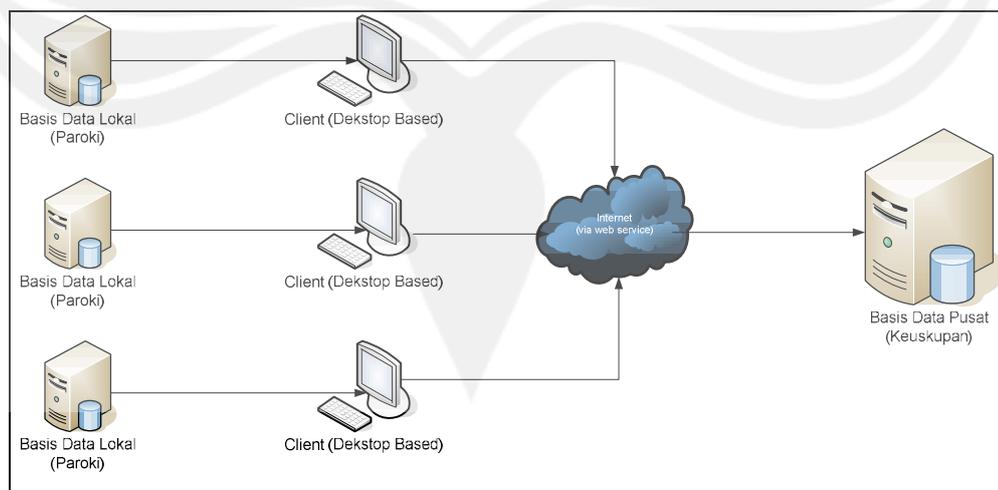
2.1 Perspektif produk

SIManja merupakan perangkat lunak yang dikembangkan untuk membantu menangani pengelolaan paroki, menangani pengelolaan wilayah, menangani pengelolaan stasi, menangani pengelolaan lingkungan, menangani pengelolaan kring, menangani pengelolaan kepala keluarga, dan menangani pengelolaan umat.

Perangkat lunak SIManja ini berjalan pada platform Windows XP untuk desktop. SIManja ini dibuat menggunakan bahasa pemrograman C# dengan lingkungan pemrograman dengan menggunakan Microsoft Visual Studio 2005.

Pengguna dapat berinteraksi melalui antar muka UI (user interface) yang akan dibuat. Pada sistem ini arsitektur yang digunakan berupa client server, di mana semua data tersimpan dalam basis data yang dapat melakukan sinkronisasi ke basis data pusat melalui internet (web service). Pengguna dapat mengakses data-data yang ada melalui desktop.

Inputan data yang telah dimasukkan akan disimpan dalam basis data lokal, maka jika terjadi pengolahan data akan dilakukan melalui basis data lokal tersebut.



Gambar i : Gambar Arsitektur perancangan perangkat lunak

SIManja

Program Studi Teknik Informatika	SKPL – SIManja	7/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2.2 Fungsi Produk

Fungsi produk perangkat lunak SIManja adalah sebagai berikut:

1. Fungsi Login (SKPL-SIManja-001)
Merupakan fungsi yang digunakan oleh pengguna untuk dapat masuk dalam sistem yang akan digunakan.
2. Fungsi Pengelolaan Paroki (SKPL-SIManja-002)
Merupakan fungsi yang digunakan untuk mengelola paroki penerbangan yang mencakup fungsi:
 - a. Fungsi Tambah Data Paroki (SKPL-SIManja-002-01)
Merupakan fungsi yang digunakan untuk menambahkan data paroki yang baru.
 - b. Fungsi Ubah Data Paroki (SKPL-SIManja-002-02)
Merupakan fungsi yang digunakan untuk mengubah data paroki.
 - c. Fungsi Tampil Data Paroki (SKPL-SIManja-002-03)
Merupakan fungsi yang digunakan untuk menampilkan data paroki.
 - d. Fungsi Hapus Data Paroki (SKPL-SIManja-002-04)
Merupakan fungsi yang digunakan untuk menghapus data paroki.
 - e. Fungsi Cari Data Paroki (SKPL-SIManja-002-05)
Merupakan fungsi yang digunakan untuk melakukan pencarian data paroki.
3. Fungsi Pengelolaan Wilayah (SKPL-SIManja-003)
Merupakan fungsi yang digunakan untuk mengelola Wilayah yang mencakup fungsi:
 - a. Fungsi Tambah Wilayah (SKPL-SIManja-003-01)
Merupakan fungsi yang digunakan untuk menambahkan data wilayah yang baru.
 - b. Fungsi Ubah Wilayah (SKPL-SIManja-003-02)
Merupakan fungsi yang digunakan untuk mengubah data wilayah.

Program Studi Teknik Informatika	SKPL – SIManja	8/ 31
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		

- c. Fungsi Tampil Wilayah (SKPL-SIManja-003-03)
Merupakan fungsi yang digunakan untuk menampilkan data wilayah.
 - d. Fungsi Hapus Wilayah (SKPL-SIManja-003-04)
Merupakan fungsi yang digunakan untuk menghapus wilayah.
 - e. Fungsi Cari Wilayah (SKPL-SIManja-003-05)
Merupakan fungsi yang digunakan untuk melakukan pencarian data wilayah.
4. Fungsi Pengelolaan Stasi (SKPL-SIManja-004)
- Merupakan fungsi yang digunakan untuk mengelola Stasi yang mencakup fungsi:
- a. Fungsi Tambah Stasi (SKPL-SIManja-004-01)
Merupakan fungsi yang digunakan untuk menambahkan data stasi yang baru.
 - b. Fungsi Ubah Stasi (SKPL-SIManja-004-02)
Merupakan fungsi yang digunakan untuk mengubah data stasi.
 - c. Fungsi Tampil Stasi (SKPL-SIManja-004-03)
Merupakan fungsi yang digunakan untuk menampilkan data stasi.
 - d. Fungsi Hapus Stasi (SKPL-SIManja-004-04)
Merupakan fungsi yang digunakan untuk menghapus stasi.
 - e. Fungsi Cari Stasi (SKPL-SIManja-004-05)
Merupakan fungsi yang digunakan untuk melakukan pencarian data stasi.
5. Fungsi Pengelolaan Lingkungan (SKPL-SIManja-005)
- Merupakan fungsi yang digunakan untuk mengelola lingkungan yang mencakup fungsi:
- a. Fungsi Tambah Lingkungan (SKPL-SIManja-005-01)
Merupakan fungsi yang digunakan untuk menambahkan data lingkungan yang baru.
 - b. Fungsi Ubah Lingkungan (SKPL-SIManja-005-02)

Program Studi Teknik Informatika	SKPL – SIManja	9/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan untuk mengubah data lingkungan.

- c. Fungsi Tampil Lingkungan (SKPL-SIManja-005-03)
Merupakan fungsi yang digunakan untuk menampilkan data lingkungan.
- d. Fungsi Hapus Lingkungan (SKPL-SIManja-005-04)
Merupakan fungsi yang digunakan untuk menghapus lingkungan.
- e. Fungsi Cari Lingkungan (SKPL-SIManja-005-05)
Merupakan fungsi yang digunakan untuk melakukan pencarian data lingkungan.

6. Fungsi Pengelolaan Kepala Keluarga (SKPL-SIManja-006)
Merupakan fungsi yang digunakan untuk mengelola kepala keluarga yang mencakup fungsi:

- a. Fungsi Tambah Kepala Keluarga (SKPL-SIManja-006-01)
Merupakan fungsi yang digunakan untuk menambahkan data kepala keluarga yang baru.
- b. Fungsi Ubah Kepala Keluarga (SKPL-SIManja-006-02)
Merupakan fungsi yang digunakan untuk mengubah data kepala keluarga.
- c. Fungsi Tampil Kepala Keluarga (SKPL-SIManja-006-03)
Merupakan fungsi yang digunakan untuk menampilkan data kepala keluarga.
- d. Fungsi Hapus Kepala Keluarga (SKPL-SIManja-006-04)
Merupakan fungsi yang digunakan untuk menghapus kepala keluarga.
- e. Fungsi Cari Kepala Keluarga (SKPL-SIManja-006-05)
Merupakan fungsi yang digunakan untuk melakukan pencarian data kepala keluarga.

7. Fungsi Pengelolaan Umat (SKPL-SIManja-007)
Merupakan fungsi yang digunakan untuk mengelola umat yang mencakup fungsi:

- a. Fungsi Tambah Umat (SKPL-SIManja-007-01)

Program Studi Teknik Informatika	SKPL – SIManja	10/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan untuk menambahkan data umat yang baru.

b. Fungsi Ubah Umat (SKPL-SIManja-007-02)

Merupakan fungsi yang digunakan untuk mengubah data umat.

c. Fungsi Tampil Umat (SKPL-SIManja-007-03)

Merupakan fungsi yang digunakan untuk menampilkan data umat.

d. Fungsi Hapus Umat (SKPL-SIManja-007-04)

Merupakan fungsi yang digunakan untuk menghapus umat.

e. Fungsi Cari Umat (SKPL-SIManja-007-05)

Merupakan fungsi yang digunakan untuk melakukan pencarian data umat.

8. Fungsi Pengelolaan Pengguna (SKPL-SIManja-008)

Merupakan fungsi yang digunakan untuk mengelola pengguna yang mencakup fungsi:

a. Fungsi Tambah Pengguna (SKPL-SIManja-008-01)

Merupakan fungsi yang digunakan untuk menambahkan data user yang baru.

b. Fungsi Ubah Pengguna (SKPL-SIManja-008-02)

Merupakan fungsi yang digunakan untuk mengubah data user.

c. Fungsi Tampil Pengguna (SKPL-SIManja-008-03)

Merupakan fungsi yang digunakan untuk menampilkan data user.

9. Fungsi Sinkronasi ke Pusat (SKPL-SIManja-009)

a. Fungsi Simpan ke File .csv (SKPL-SIManja-009-01)

Merupakan fungsi yang digunakan untuk menyimpan data-data kepala keluarga dan umat ke dalam file yang berformat csv.

b. Fungsi Kirim File .csv ke Pusat (SKPL-SIManja-009-02)

Program Studi Teknik Informatika	SKPL – SIManja	11/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan untuk mengirimkan file yang akan di kirim ke basis data pusat, lalu melakukan sinkronasi.

10. Fungsi Pencarian Umat di Pusat (SKPL-SIManja-010)

Merupakan fungsi yang digunakan untuk mencari umat yang ada di dalam basis data pusat.

11. Fungsi Pengelolaan Tabel Refrensi (SKPL-SIManja-011)

a. Fungsi Tambah Data Tabel Refrensi (SKPL-SIManja-011-01)

Merupakan fungsi yang digunakan untuk menambahkan data user yang baru.

b. Fungsi Tampil Data Tabel Refrensi (SKPL-SIManja-011-02)

Merupakan fungsi yang digunakan untuk menampilkan data Tabel Refrensi.

c. Fungsi Cari Keterangan Data Tabel Refrensi (SKPL-SIManja-011-03)

Merupakan fungsi yang digunakan untuk melakukan pencarian data di tabel refrensi.

2.3 Karakteristik Pengguna

Pengguna perangkat lunak SIManja tersebut adalah operator dengan karakteristik sebagai berikut:

1. Memahami pengoperasian komputer secara aktif.
2. Memahami sistem komputer tempat perangkat lunak dijalankan.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan sistem pembuatan perangkat lunak SIManja adalah sebagai berikut:

1. Kebijakan umum
Mengacu pada tujuan pengembangan perangkat lunak SIManja.
2. Keterbatasan perangkat keras

Program Studi Teknik Informatika	SKPL – SIManja	12/ 31
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		

Ditentukan kemudian setelah pengembang mengetahui ketersediaan perangkat keras pada pelanggan.

2.5 Asumsi dan Ketergantungan

Asumsi yang digunakan dalam pengembangan perangkat lunak SIManja yaitu:

- Tersedia komputer Server dengan spesifikasi prosesor 2 GHz, memori primer minimal 512 MB, kapasitas media penyimpanan sekunder server yaitu 40 GB.
- Tersedia Microsoft .NET Framework SDK versi 1.1 atau yang lebih tinggi.
- Tersedia pengendali ODBC atau ODBC driver untuk antarmuka DBMS Microsoft SQL Server 2005.

3. Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

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

3.1.1 Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka tampilan form(GUI). Kemudian pengguna memberikan masukan dengan melalui keyboard dengan cara diketikan pada form yang ada, sesuai dengan yang diminta oleh system perangkat lunak. Keluaran dari perangkat lunak ini juga dalam bentuk tampilan form sesuai dengan perintah yang diberikan.

3.1.2 Antarmuka perangkat keras

Kebutuhan minimum perangkat keras yang dapat digunakan oleh SIManja adalah:

1. PC

Program Studi Teknik Informatika	SKPL – SIManja	13/ 31
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		

PC terdiri dari prosesor 2 GHz, memori primer minimal 512 MB, kapasitas media penyimpanan sekunder server yaitu 40 GB.

2. Keyboard

Perangkat keras ini digunakan untuk mengetikan data yang akan dimasukkan yang berupa karakter atau text.

3. Mouse

Perangkat keras ini digunakan untuk alat bantu yang berkaitan dengan event klik, drag and drop, dan focus.

4. Printer

Perangkat keras ini nantinya digunakan sebagai cetak dokumen-dokumen yang perlu dicetak.

3.1.3 Antarmuka perangkat lunak

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

1. Nama : SQL Server 2005

Sumber : Microsoft

Sebagai basis data management system (DBMS) yang dibutuhkan dalam mengoperasikan perangkat lunak SIManja.

2. Nama : Windows XP/7

Sumber : Microsoft.

Sebagai sistem operasi komputer

3. Nama : .Net Framework

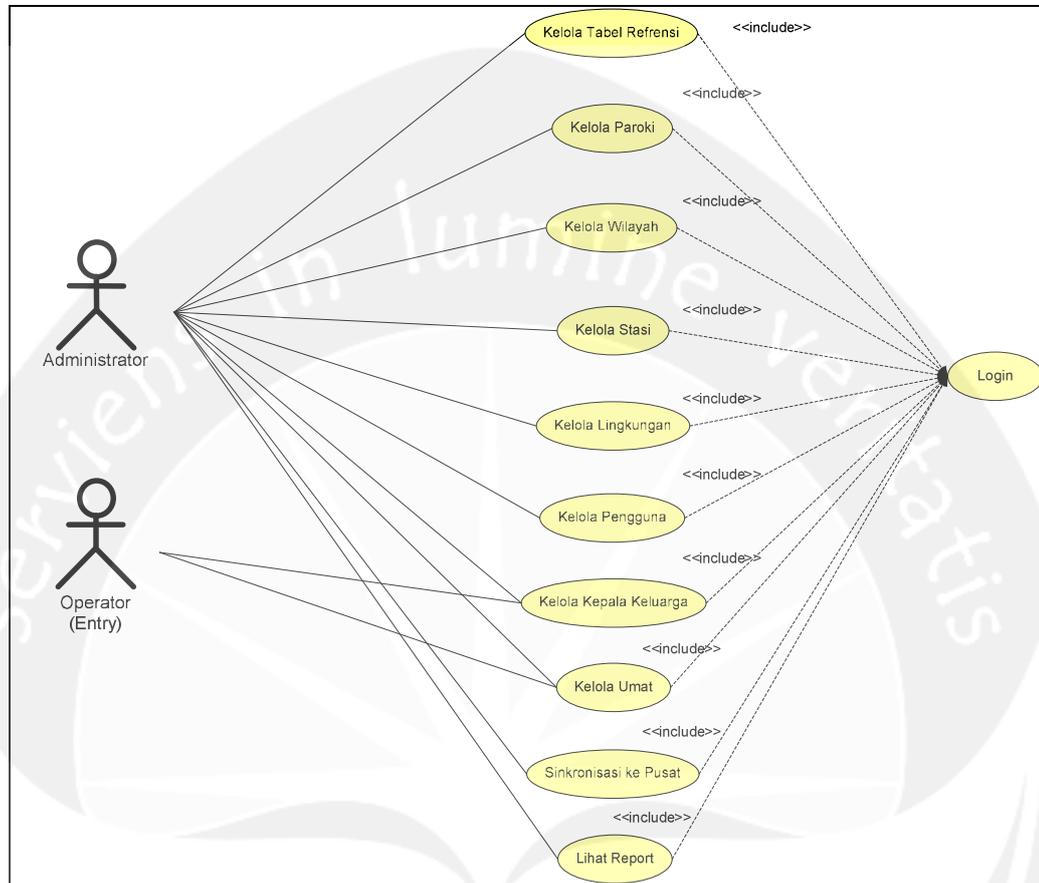
Sumber : Microsoft

Sebagai alat pengembangan Sistem Informasi Manajemen Gereja.

Program Studi Teknik Informatika	SKPL – SIManja	14/ 31
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 ii : Use Case Diagram perancangan SIManja

4. Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

4.1.1 Use case Spesification: Login

1. Brief Description

Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada sebuah id unik yaitu user name dari user dan password yang berupa rangkaian karakter.

2. Primary Actor

1. Administrator
2. Operator (Entry)

3. Supporting Actor

Program Studi Teknik Informatika	SKPL – SIManja	15/ 31
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 login
2. Sistem menampilkan antarmuka untuk login
3. Aktor memasukkan user name dan password
4. Sistem memeriksa user name dan password yang diinputkan aktor
5. Sistem memberikan akses ke aktor
6. Use Case ini selesai

5. Alternative Flow

none

6. Error Flow

Setelah basic flow langkah ke 3:

Jika user name atau password tidak ditemukan maka:

1. Sistem menampilkan peringatan bahwa user name atau password tidak sesuai
2. Kembali ke Basic Flow langkah ke 3

7. PreConditions

none

8. PostConditions

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

4.1.2 Use case Spesification: Pengelolaan Paroki

1. Brief Description

Use Case ini digunakan oleh Administrator untuk mengolah data-data paroki yang ada. Seperti memasukan data baru, mengubah data, menghapus data, menampilkan data, dan mencari data paroki.

2. Primary Actor

Administrator

3. Supporting Actor

none

Program Studi Teknik Informatika	SKPL – SIManja	16/ 31
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. Basic Flow

1. Use Case ini dimulai ketika Administrator menerima perubahan informasi mengenai paroki.
2. Sistem menampilkan form pengelolaan paroki.
3. Administrator akan memasukan perubahan data-data tersebut supaya data di dalam sistem sama dengan keadaan di lapangan.
4. Sistem akan menyimpan data baru tersebut ke dalam basis data.
5. Use Case selesai

5. Alternative Flow

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan perubahan data paroki.

1. Sistem menampilkan data paroki yang sudah tercatat.
2. Administrator mengubah data paroki tertentu yang sudah ditampilkan.
3. Administrator meminta sistem untuk menyimpan data baru yang telah diubah.
4. Sistem melakukan pengecekan terhadap data yang telah diubah.
5. Sistem menyimpan data yang telah diubah ke dalam basis data.
6. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan penghapusan data paroki.

1. Sistem menampilkan data paroki yang sudah tercatat.
2. Administrator memilih data paroki yang akan dihapus dari basis data.
3. Sistem menghapus data yang dipilih dari dalam basis data.
4. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

Program Studi Teknik Informatika	SKPL – SIManja	17/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

*Administrator memilih untuk melakukan pencarian data paroki.

1. Sistem menampilkan form.
2. Administrator memasukkan data paroki yang dicari.
3. Sistem menampilkan data barang tersebut.
4. Kembali ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan tampil data paroki.

1. Sistem menampilkan data paroki yang ada dalam basis data.
2. Berlanjut ke Basic Flow langkah ke 4.

6.Error Flow

none

7.PreConditions

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

8.PostConditions

1. Data-data mengenai paroki telah diubah menjadi data terkini.

4.1.3 Use case Spesification: Pengelolaan Wilayah

1.Brief Description

Use Case ini digunakan oleh Administrator untuk mengolah data-data wilayah yang ada. Seperti memasukan data baru, mengubah data, menghapus data, menampilkan data, dan mencari data wilayah.

2.Primary Actor

Administrator

3.Supporting Actor

none

4.Basic Flow

1. Use Case ini dimulai ketika Administrator menerima perubahan informasi mengenai wilayah.

Program Studi Teknik Informatika	SKPL – SIManja	18/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem menampilkan form pengelolaan wilayah.
3. Administrator akan memasukan perubahan data-data tersebut supaya data di dalam sistem sama dengan keadaan di lapangan.
4. Sistem akan menyimpan data baru tersebut ke dalam basis data.
5. Use Case selesai

5. Alternative Flow

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan perubahan data wilayah.

1. Sistem menampilkan data wilayah yang sudah tercatat.
2. Administrator mengubah data wilayah tertentu yang sudah ditampilkan.
3. Administrator meminta sistem untuk menyimpan data baru yang telah diubah.
4. Sistem melakukan pengecekan terhadap data yang telah diubah.
5. Sistem menyimpan data yang telah diubah ke dalam basis data.
6. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan penghapusan data wilayah.

1. Sistem menampilkan data wilayah yang sudah tercatat.
2. Administrator memilih data wilayah yang akan dihapus dari basis data.
3. Sistem manghapus data yang dipilih dari dalam basis data.
4. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

Program Studi Teknik Informatika	SKPL – SIManja	19/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

*Administrator memilih untuk melakukan pencarian data wilayah.

1. Sistem menampilkan form.
2. Administrator memasukkan data wilayah yang dicari.
3. Sistem menampilkan data barang tersebut.
4. Kembali ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan tampil data wilayah.

1. Sistem menampilkan data wilayah yang ada dalam basis data.
2. Berlanjut ke Basic Flow langkah ke 4.

6.Error Flow

none

7.PreConditions

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

8.PostConditions

1. Data-data mengenai wilayah telah diubah menjadi data terkini.

4.1.4 Use case Spesification: Pengelolaan Stasi

1.Brief Description

Use Case ini digunakan oleh Administrator untuk mengolah data-data stasi yang ada. Seperti memasukan data baru, mengubah data, menghapus data, menampilkan data, dan mencari data stasi.

2.Primary Actor

Administrator

3.Supporting Actor

none

4.Basic Flow

1. Use Case ini dimulai ketika Administrator menerima perubahan informasi mengenai stasi.

Program Studi Teknik Informatika	SKPL – SIManja	20/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem menampilkan form pengelolaan stasi.
3. Administrator akan memasukan perubahan data-data tersebut supaya data di dalam sistem sama dengan keadaan di lapangan.
4. Sistem akan menyimpan data baru tersebut ke dalam basis data.
5. Use Case selesai

5. Alternative Flow

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan pengubahan data stasi.

1. Sistem menampilkan data stasi yang sudah tercatat.
2. Administrator mengubah data stasi tertentu yang sudah ditampilkan.
3. Administrator meminta sistem untuk menyimpan data baru yang telah diubah.
4. Sistem melakukan pengecekan terhadap data yang telah diubah.
5. Sistem menyimpan data yang telah diubah ke dalam basis data.
6. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan penghapusan data stasi.

1. Sistem menampilkan data stasi yang sudah tercatat.
2. Administrator memilih data stasi yang akan dihapus dari basis data.
3. Sistem manghapus data yang dipilih dari dalam basis data.
4. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan pencarian data stasi.

1. Sistem menampilkan form.

Program Studi Teknik Informatika	SKPL – SIManja	21/ 31
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. Administrator memasukkan data stasi yang dicari.
3. Sistem menampilkan data barang tersebut.
4. Kembali ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan tampil data stasi.

1. Sistem menampilkan data stasi yang ada dalam basis data.
2. Berlanjut ke Basic Flow langkah ke 4.

6. Error Flow

none

7. PreConditions

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

8. PostConditions

1. Data-data mengenai stasi telah diubah menjadi data terkini.

4.1.5 Use case Spesification: Pengelolaan Lingkungan

1. Brief Description

Use Case ini digunakan oleh Administrator untuk mengolah data-data lingkungan yang ada. Seperti memasukan data baru, mengubah data, menghapus data, menampilkan data, dan mencari data lingkungan.

2. Primary Actor

Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika Administrator menerima perubahan informasi mengenai lingkungan.
2. Sistem menampilkan form pengelolaan lingkungan.

Program Studi Teknik Informatika	SKPL – SIManja	22/ 31
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. Administrator akan memasukan perubahan data-data tersebut supaya data di dalam sistem sama dengan keadaan di lapangan.
4. Sistem akan menyimpan data baru tersebut ke dalam basis data.
5. Use Case selesai

5. Alternative Flow

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan perubahan data lingkungan.

1. Sistem menampilkan data lingkungan yang sudah tercatat.
2. Administrator mengubah data lingkungan tertentu yang sudah ditampilkan.
3. Administrator meminta sistem untuk menyimpan data baru yang telah diubah.
4. Sistem melakukan pengecekan terhadap data yang telah diubah.
5. Sistem menyimpan data yang telah diubah ke dalam basis data.
6. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan penghapusan data lingkungan.

1. Sistem menampilkan data lingkungan yang sudah tercatat.
2. Administrator memilih data lingkungan yang akan dihapus dari basis data.
3. Sistem menghapus data yang dipilih dari dalam basis data.
4. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan pencarian data lingkungan.

Program Studi Teknik Informatika	SKPL – SIManja	23/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Sistem menampilkan form.
2. Administrator memasukkan data lingkungan yang dicari.
3. Sistem menampilkan data barang tersebut.
4. Kembali ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan tampil data lingkungan.

1. Sistem menampilkan data lingkungan yang ada dalam basis data.
2. Berlanjut ke Basic Flow langkah ke 4.

6.Error Flow

none

7.PreConditions

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

8.PostConditions

1. Data-data mengenai lingkungan telah diubah menjadi data terkini.

4.1.6 Use case Spesification: Pengelolaan Kepala keluarga

1.Brief Description

Use Case ini digunakan oleh Aktor untuk mengolah data-data kepala keluarga yang ada. Seperti memasukan data baru, mengubah data, menghapus data, menampilkan data, dan mencari data kepala keluarga.

2.Primary Actor

1. Administrator
2. Operator (Entry)

3.Supporting Actor

none

4.Basic Flow

1. Use Case ini dimulai ketika Aktor menerima perubahan informasi mengenai kepala keluarga.

Program Studi Teknik Informatika	SKPL – SIManja	24/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem menampilkan form pengolahan kepala keluarga.
3. Aktor akan memasukan perubahan data-data tersebut supaya data di dalam sistem sama dengan keadaan di lapangan.
4. Sistem akan menyimpan data baru tersebut ke dalam basis data.
5. Use Case selesai

5. Alternative Flow

Setelah basic flow langkah ke 2:

*Aktor memilih untuk melakukan perubahan data kepala keluarga.

1. Sistem menampilkan data kepala keluarga yang sudah tercatat.
2. Aktor mengubah data kepala keluarga tertentu yang sudah ditampilkan.
3. Aktor meminta sistem untuk menyimpan data baru yang telah diubah.
4. Sistem melakukan pengecekan terhadap data yang telah diubah.
5. Sistem menyimpan data yang telah diubah ke dalam basis data.
6. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Aktor memilih untuk melakukan penghapusan data kepala keluarga.

1. Sistem menampilkan data kepala keluarga yang sudah tercatat.
2. Aktor memilih data kepala keluarga yang akan dihapus dari basis data.
3. Sistem manghapus data yang dipilih dari dalam basis data.
4. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

Program Studi Teknik Informatika	SKPL – SIManja	25/ 31
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		

*Aktor memilih untuk melakukan pencarian data kepala keluarga.

1. Sistem menampilkan form.
2. Aktor memasukkan data kepala keluarga yang dicari.
3. Sistem menampilkan data barang tersebut.
4. Kembali ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Aktor memilih untuk melakukan tampil data kepala keluarga.

1. Sistem menampilkan data kepala keluarga yang ada dalam basis data.
2. Berlanjut ke Basic Flow langkah ke 4.

6.Error Flow

none

7.PreConditions

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

8.PostConditions

1. Data-data mengenai kepala keluarga telah diubah menjadi data terkini.

4.1.7 Use case Spesification: Pengelolaan Umat

1.Brief Description

Use Case ini digunakan oleh Aktor untuk mengolah data-data umat yang ada. Seperti memasukan data baru, mengubah data, menghapus data, menampilkan data, dan mencari data umat.

2.Primary Actor

1. Administrator
2. Operator (Entry)

3.Supporting Actor

none

4.Basic Flow

Program Studi Teknik Informatika	SKPL – SIManja	26/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Use Case ini dimulai ketika Aktor menerima perubahan informasi mengenai umat.
2. Sistem menampilkan form pengolahan umat.
3. Aktor akan memasukan perubahan data-data tersebut supaya data di dalam sistem sama dengan keadaan di lapangan.
4. Sistem akan menyimpan data baru tersebut ke dalam basis data.
5. Use Case selesai

5. Alternative Flow

Setelah basic flow langkah ke 2:

*Aktor memilih untuk melakukan perubahan data umat.

1. Sistem menampilkan data umat yang sudah tercatat.
2. Aktor mengubah data umat tertentu yang sudah ditampilkan.
3. Aktor meminta sistem untuk menyimpan data baru yang telah diubah.
4. Sistem melakukan pengecekan terhadap data yang telah diubah.
5. Sistem menyimpan data yang telah diubah ke dalam basis data.
6. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Aktor memilih untuk melakukan penghapusan data umat.

1. Sistem menampilkan data umat yang sudah tercatat.
2. Aktor memilih data umat yang akan dihapus dari basis data.
3. Sistem manghapus data yang dipilih dari dalam basis data.
4. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Aktor memilih untuk melakukan pencarian data umat.

1. Sistem menampilkan form.
2. Aktor memasukkan data umat yang dicari.

Program Studi Teknik Informatika	SKPL – SIManja	27/ 31
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 menampilkan data barang tersebut.
4. Kembali ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Aktor memilih untuk melakukan tampil data umat.

1. Sistem menampilkan data umat yang ada dalam basis data.
2. Berlanjut ke Basic Flow langkah ke 4.

6. Error Flow

none

7. PreConditions

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

8. PostConditions

1. Data-data mengenai umat telah diubah menjadi data terkini.

4.1.8 Use case Spesification: Pengelolaan User

1. Brief Description

Use Case ini digunakan oleh Administrator untuk mengolah user yang meliputi penambahan user, mengubah data user, menampilkan data user, dan mencari data user.

2. Primary Actor

Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika ada perubahan informasi mengenai user.
2. Sistem masuk ke form pengelolaan user.
3. Administrator mengolah data-data dalam sistem menjadi data-data user yang sesuai dengan kondisi.
4. Use Case selesai

5. Alternative Flow

Setelah basic flow langkah ke 2:

Program Studi Teknik Informatika	SKPL – SIManja	28/ 31
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

*Administrator memilih untuk melakukan perubahan data user.

1. Sistem menampilkan data user yang sudah tercatat.
2. Administrator mengubah data user tertentu yang sudah ditampilkan.
3. Administrator meminta sistem untuk menyimpan data baru yang telah diubah.
4. Sistem melakukan pengecekan terhadap data yang telah diUbah.
5. Sistem meyimpan data yang telah diUbah ke dalam basis data.
6. Berlanjut ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan pencarian data user.

1. Sistem menampilkan form.
2. Administrator memasukkan data user yang dicari.
3. Sistem menampilkan data user tersebut.
4. Kembali ke Basic Flow langkah ke 4.

Setelah basic flow langkah ke 2:

*Administrator memilih untuk melakukan tampil data user.

1. Sistem menampilkan data user yang ada dalam basis data.
2. Berlanjut ke Basic Flow langkah ke 4.

6. Error Flow

none

7. PreConditions

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

8. PostConditions

1. Data-data user telah dimasukkan ke dalam sistem sehingga data-datanya sesuai dengan keadaan di lapangan.

Program Studi Teknik Informatika	SKPL – SIManja	29/ 31
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: Sinkronisasi ke Pusat

1. Brief Description

Use Case ini digunakan oleh Aktor untuk melakukan sinkronisasi data yang ada di sistem yang akan di kirimkan ke pusat.

2. Primary Actor

1. Operator (Entry)

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika Aktor ingin melakukan sinkronisasi ke data di pusat.
2. Sistem menampilkan form Kirim Data ke Pusat.
3. Aktor akan memasukan file mana yang akan di kirimkan ke pusat.
4. Sistem akan melakukan sinkronisasi ke basis data di pusat.
5. Use Case selesai

5. Alternative Flow

Setelah basic flow langkah ke 2:

*Aktor memilih untuk melakukan penyimpanan ke file.

1. Sistem menampilkan form penyimpanan data berformat .csv.
2. Aktor memilih tempat di mana file akan di simpan.
3. Berlanjut ke Basic Flow langkah ke 5.

6. Error Flow

none

7. PreConditions

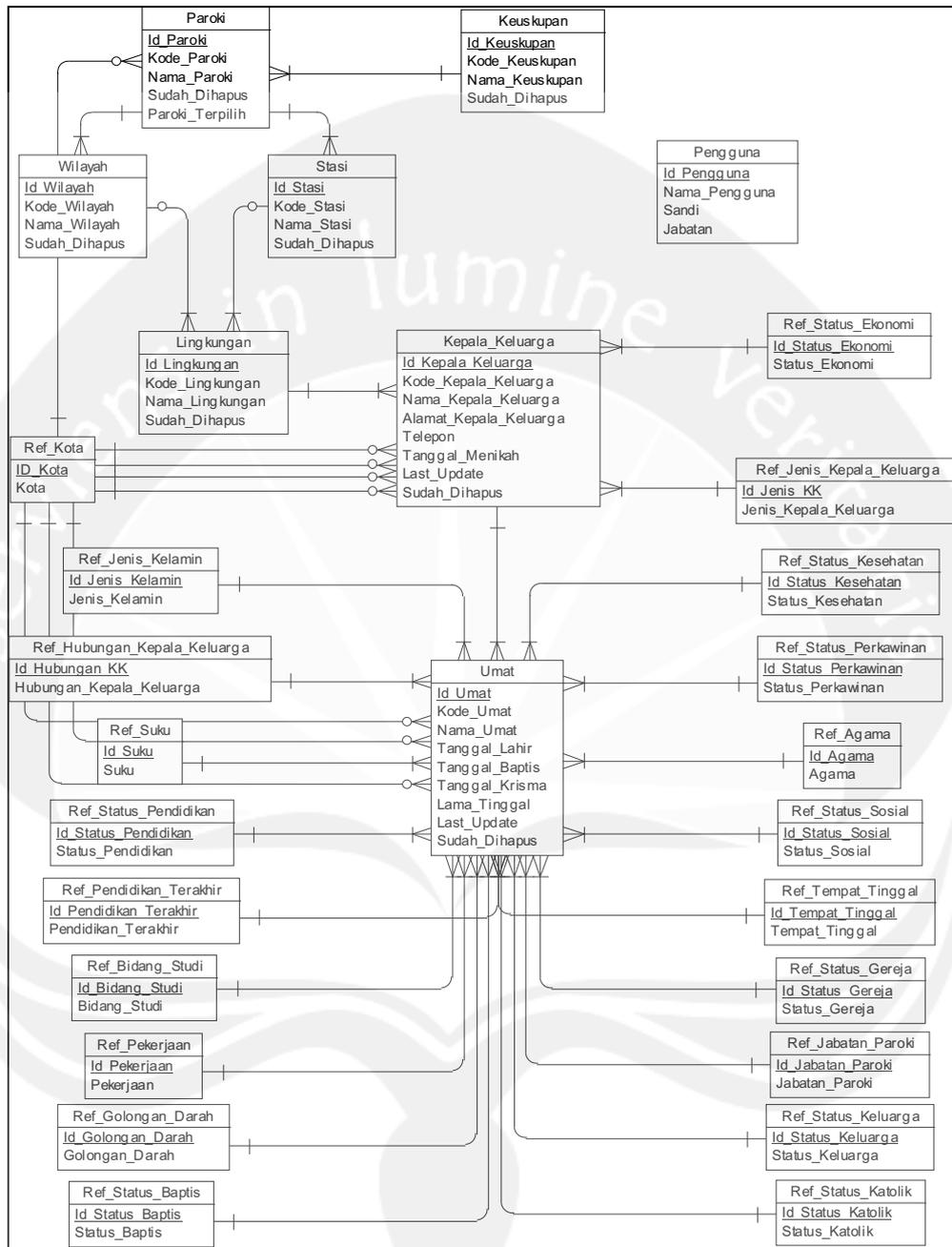
1. Use Case Login telah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

1. Data-data mengenai kepala keluarga dan umat telah disinkronisasi dengan pusat sehingga data pusat menjadi lebih *up to date*.

Program Studi Teknik Informatika	SKPL – SIManja	30/ 31
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		

5. Entity Relationship Diagram (ERD)



Gambar iii : Entity Relationship Diagram perancangan SIManja

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

**Sistem Informasi Manajemen Gereja
(SIManja)**

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Jeffrey Soeprapto / 07 07 05232

**Program Studi Teknik Informatika - Fakultas Teknologi
Industri
Universitas Atma Jaya Yogyakarta**

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL-SIManja		1/ 90
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

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

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1. Pendahuluan.	5
1.1 Tujuan	5
1.2 Lingkup Masalah.	5
1.3 Definisi, Akronim, dan Singkatan	5
1.4 Referensi.	6
2. Rancangan Sistem	6
2.1 Perancangan Arsitektur	6
2.2 Perancangan Rinci	6
2.2.1 Sequence Diagram	7
2.2.2 Kelas Diagram.	38
2.2.3 Spesifikasi Deskripsi Kelas Diagram.	39
3. Deskripsi Dekompisisi	58
3.1 Dekomposisi Data	58
3.2 Conseptual Data Model.	68
3.3 Physical Data Model.	69
4. Deskripsi Rancangan Antarmuka.	71
4.1 Login.	71
4.2 Main Form.	72
4.3 Pengelolaan Paroki	74
4.4 Pengelolaan Wilayah.	75
4.5 Pengelolaan Stasi.	76
4.6 Pengelolaan Lingkungan	77
4.7 Pengelolaan Kepala Keluarga.	78
4.8 Pengelolaan Umat	80
4.9 Pengelolaan Pengguna	82

1. Pendahuluan

1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini merupakan dokumen perancangan perangkat lunak SIManja (Sistem Informasi Manajemen Gereja) untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

1.2 Lingkup Masalah

Perangkat Lunak SIManja 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.

Dan berjalan pada lingkungan dengan platform Windows XP.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Merupakan deskripsi perancangan dari perangkat lunak yang akan dikembangkan.
SIManja	Perangkat lunak yang menangani masalah pengelolaan umat disebuah gereja.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

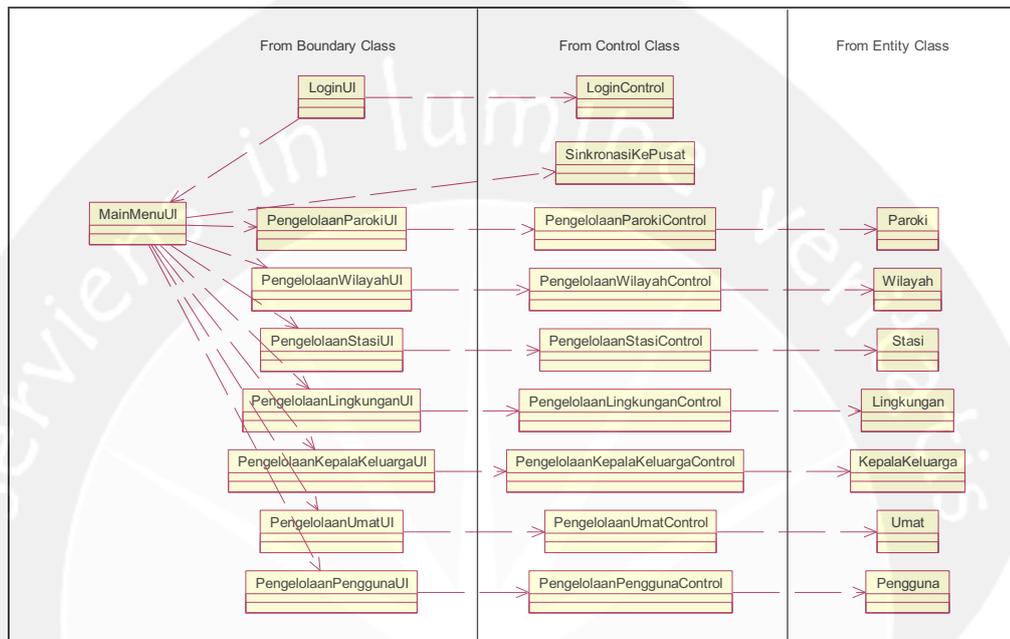
1. SKPL dan DPPL Jeffrey Soeprapto
2. Software Development planning SILogi

Program Studi Teknik Informatika	DPPL-SIManja	5/ 90
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. Template SKPL berbasis OO

2. Rancangan Sistem

2.1 Perancangan Arsitektur

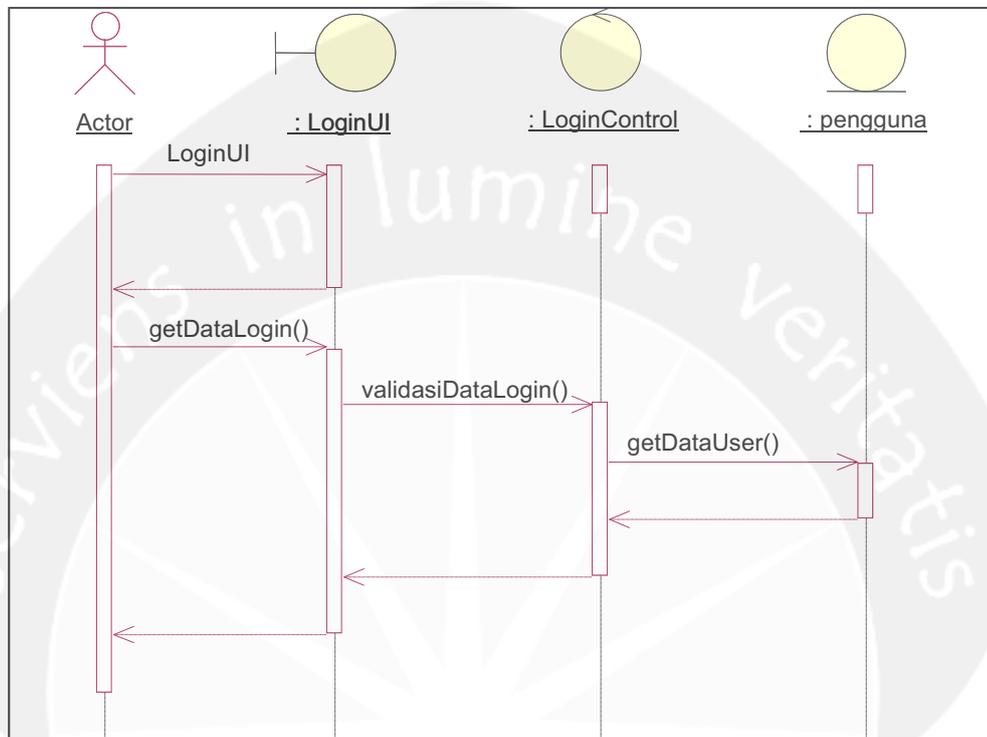


Gambar i: rancangan arsitektur SIManja

2.2 Perancangan Rinci

2.2.1 Sequence Diagram

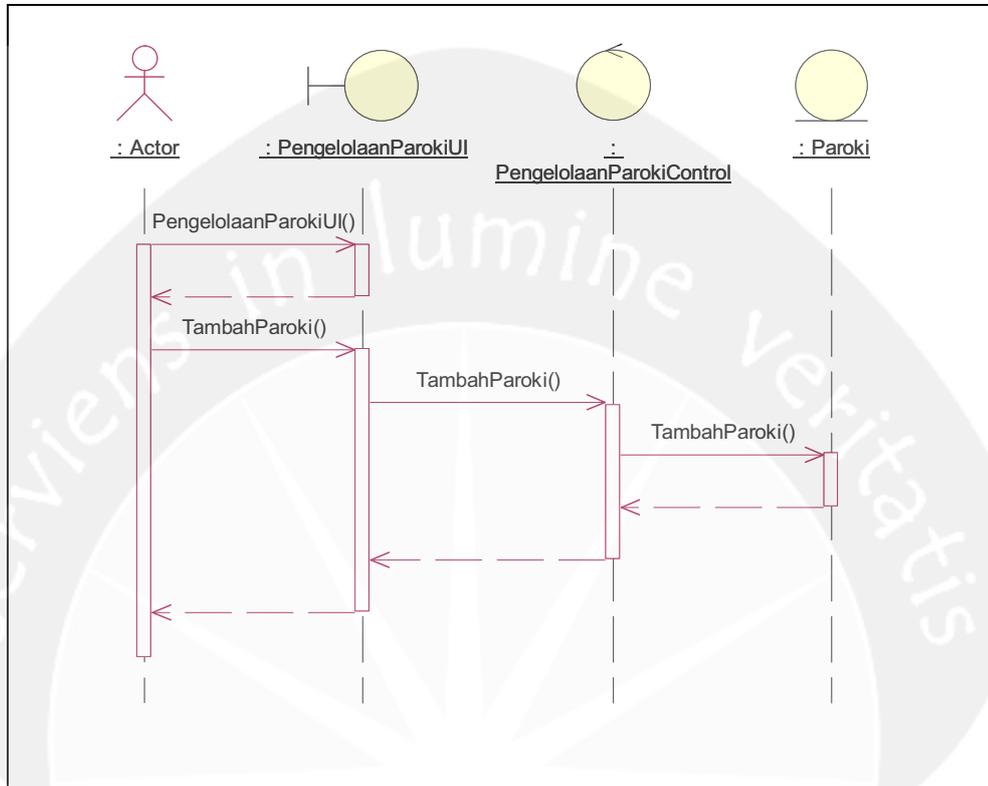
2.2.1.1 Login



Gambar ii: Sequence Diagram Use Case Login

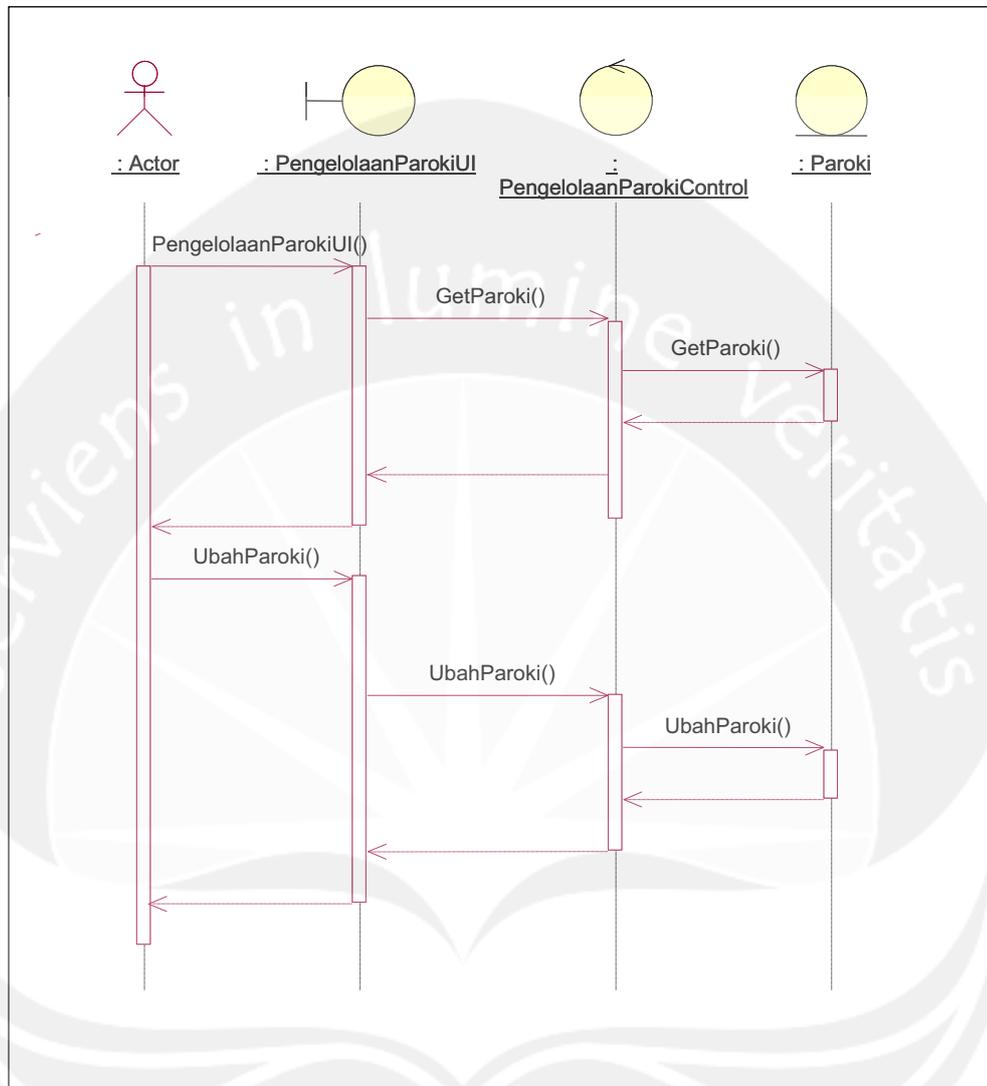
2.2.1.2 Pengelolaan Paroki

2.2.1.2.1 Pengelolaan Paroki (Tambah)



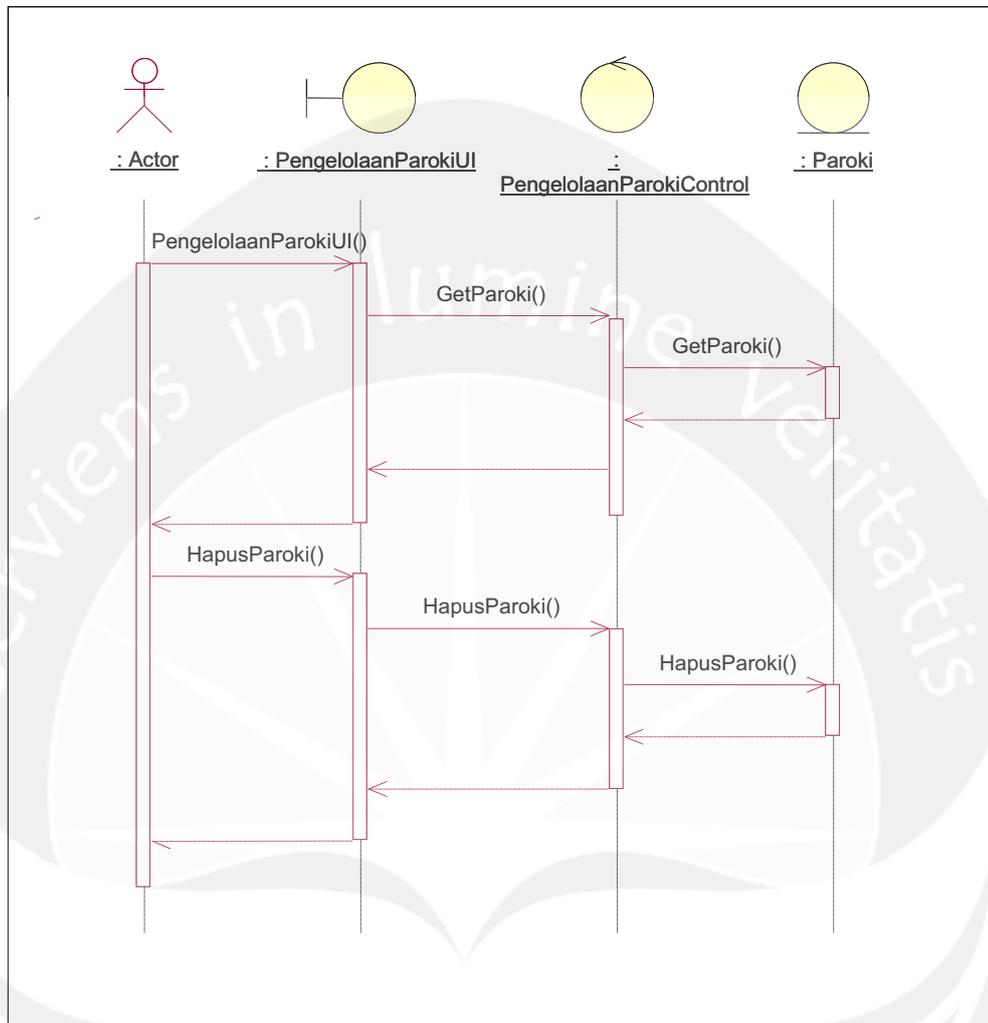
Gambar iii: Sequence Diagram Use Case Pengelolaan Paroki (tambah)

2.2.1.2.2 Pengelolaan Paroki (Ubah)



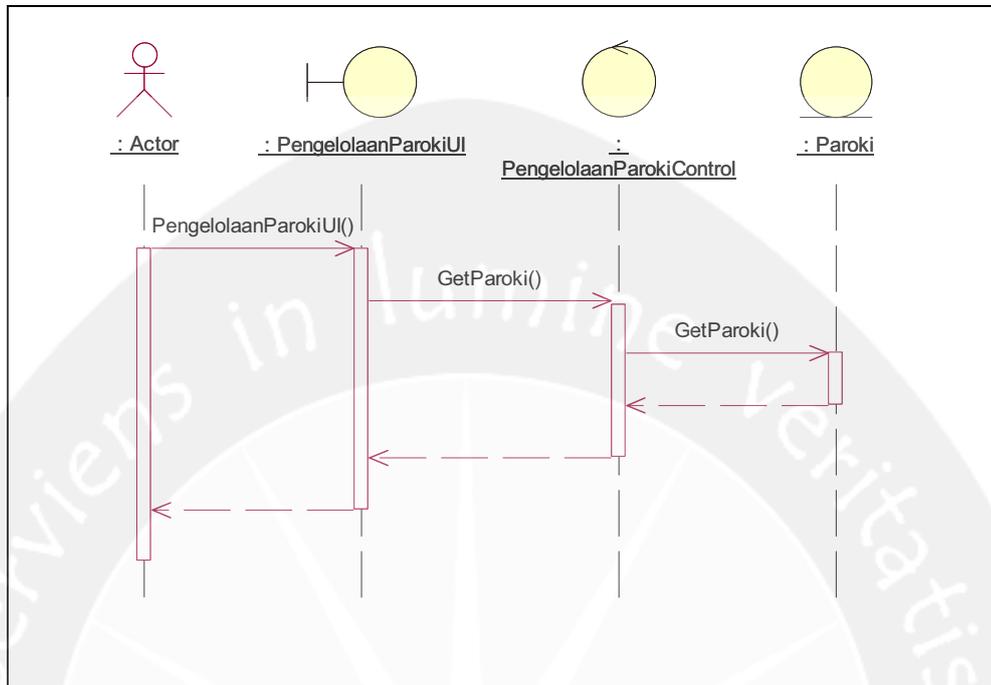
Gambar iv: Sequence Diagram Use Case Pengelolaan Paroki (ubah)

2.2.1.2.3 Pengelolaan Paroki (Hapus)



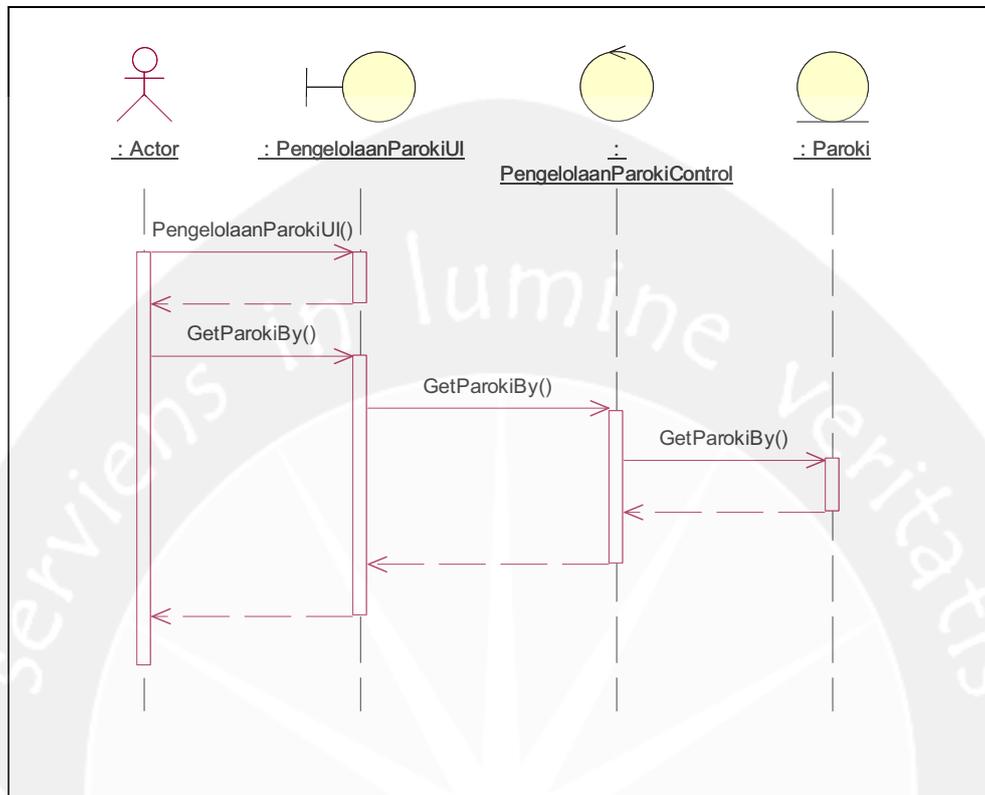
Gambar v: Sequence Diagram Use Case Pengelolaan Paroki (hapus)

2.2.1.2.4 Pengelolaan Paroki (Tampil)



Gambar vi: Sequence Diagram Use Case Pengelolaan Paroki (tampil)

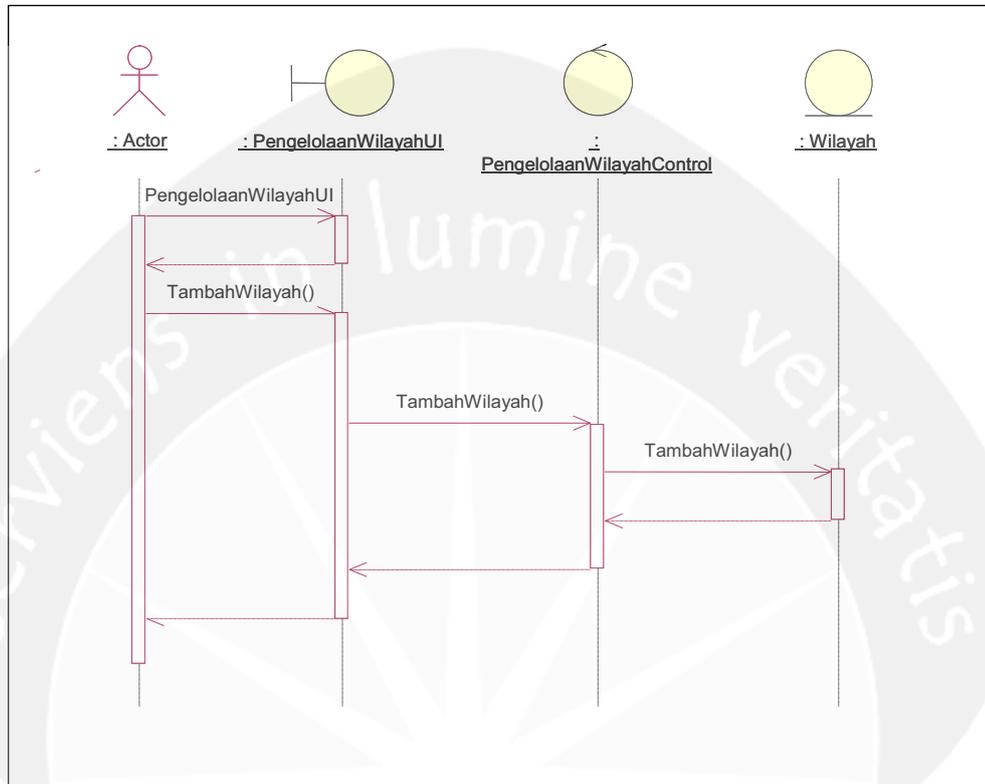
2.2.1.2.5 Pengelolaan Paroki (Cari)



Gambar vii: Sequence Diagram Use Case Pengelolaan Paroki (cari)

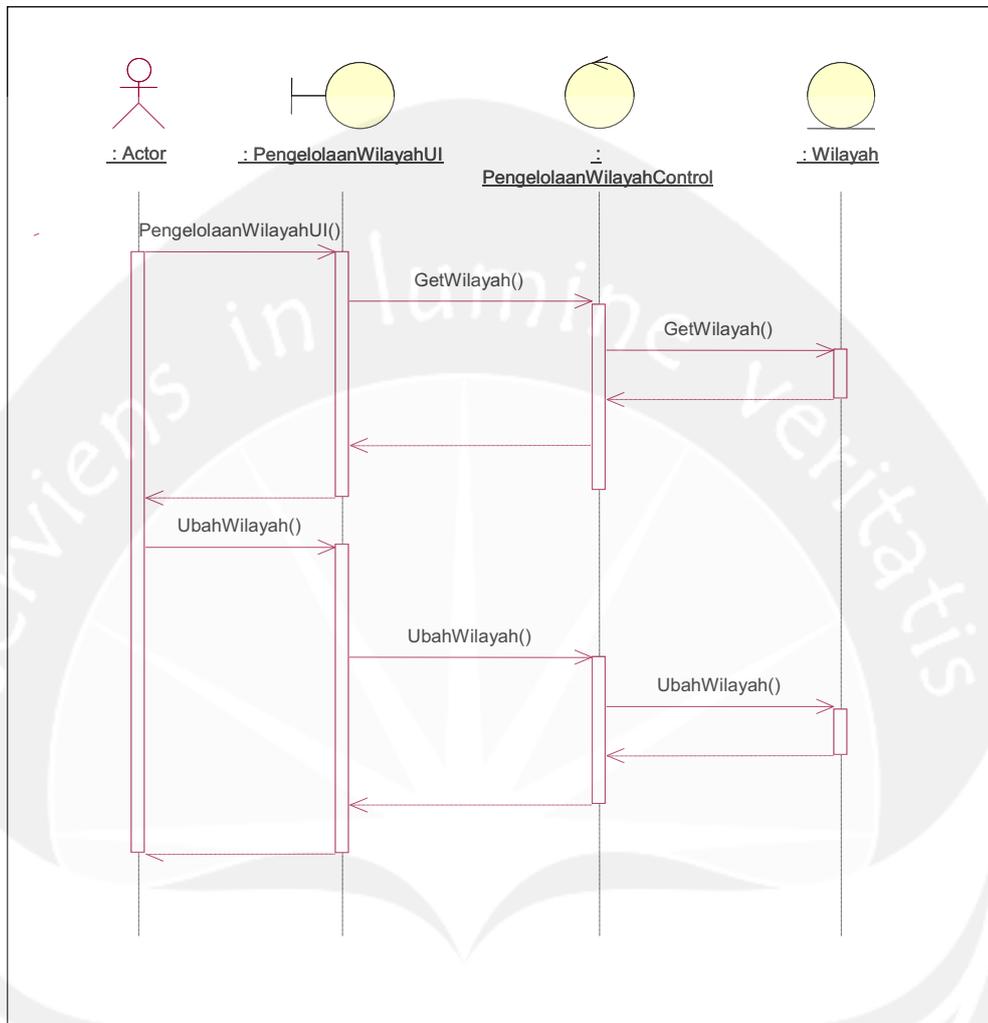
2.2.1.3 Pengelolaan Wilayah

2.2.1.3.1 Pengelolaan Wilayah (Tambah)



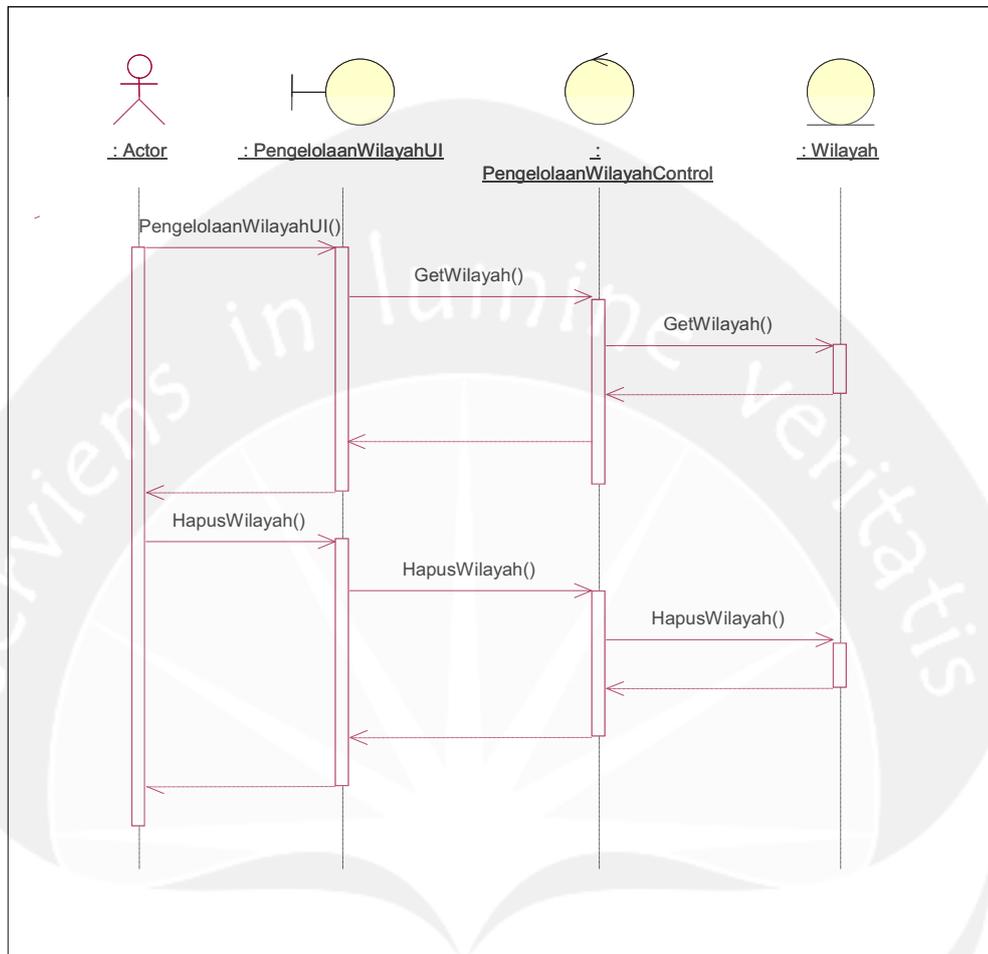
Gambar viii: Sequence Diagram Use Case Pengelolaan Wilayah (tambah)

2.2.1.3.2 Pengelolaan Wilayah (Ubah)



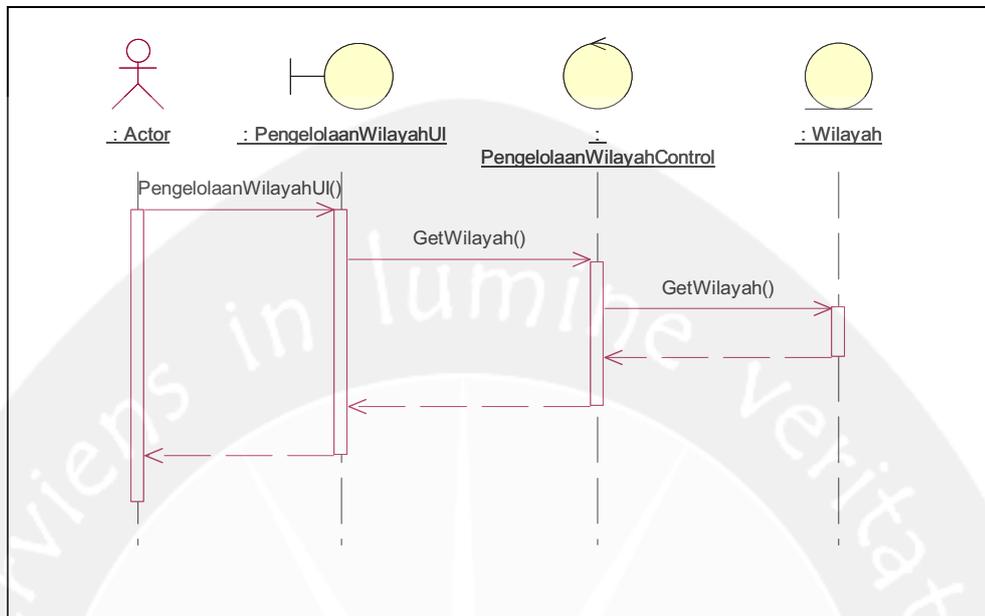
Gambar ix: Sequence Diagram Use Case Pengelolaan Wilayah (ubah)

2.2.1.3.3 Pengelolaan Wilayah (Hapus)



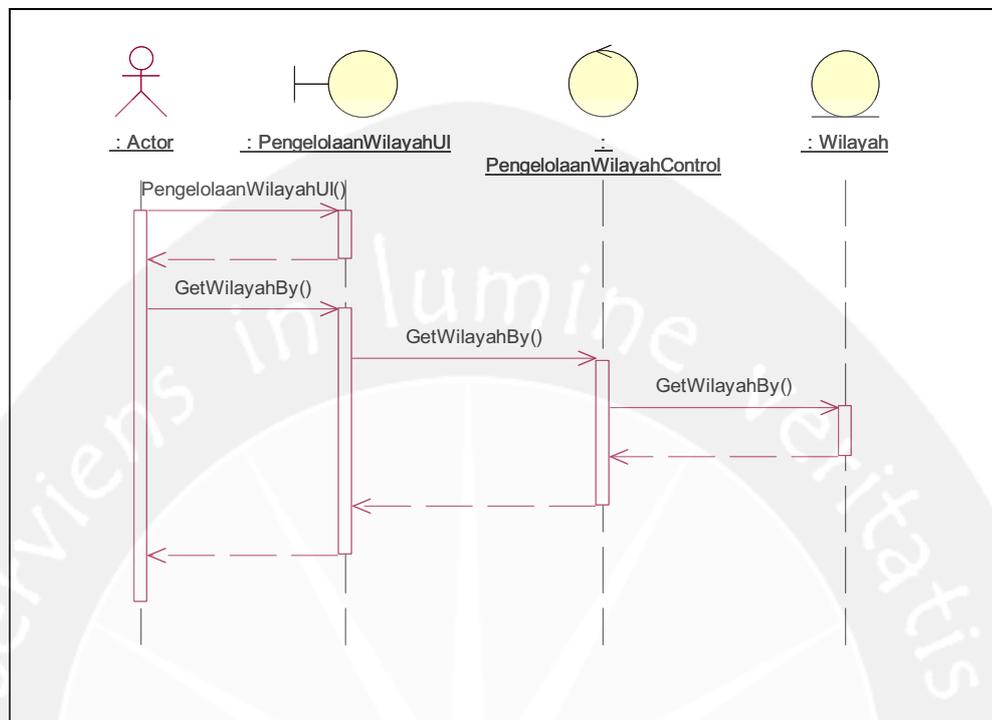
Gambar x: Sequence Diagram Use Case Pengelolaan Wilayah (hapus)

2.2.1.3.4 Pengelolaan Wilayah (Tampil)



Gambar xi: Sequence Diagram Use Case Pengelolaan Wilayah (tampil)

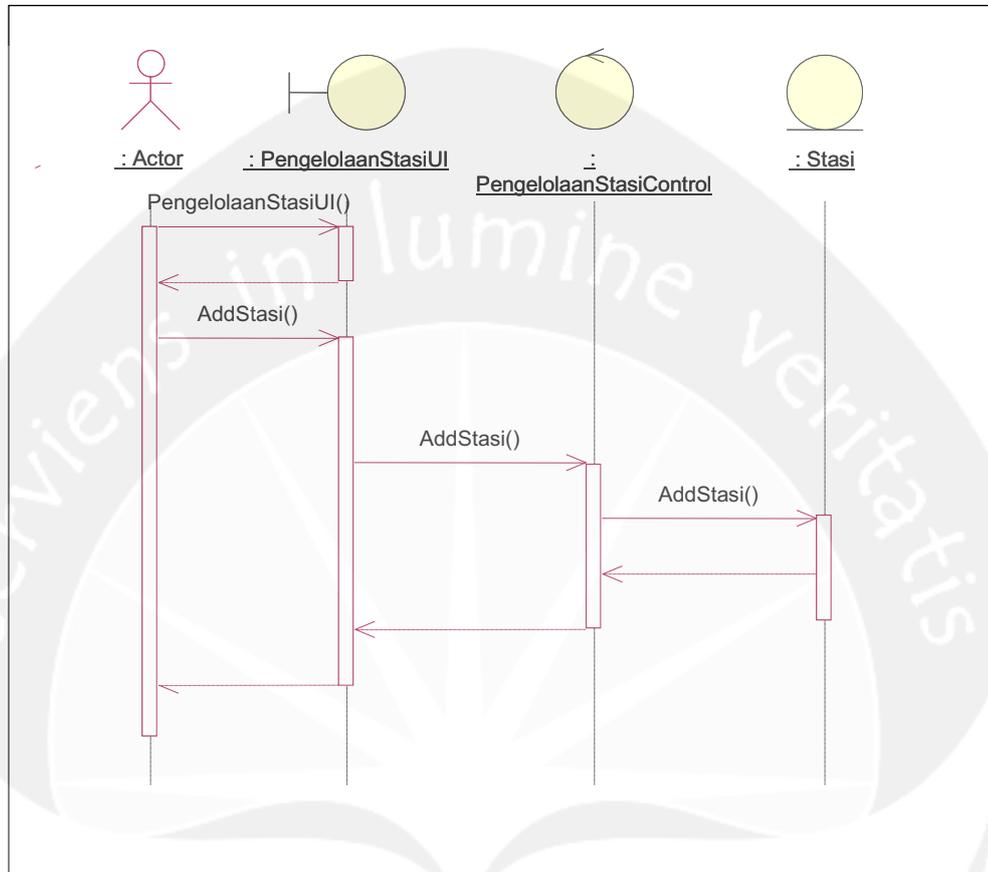
2.2.1.3.5 Pengelolaan Wilayah (Cari)



Gambar xii: Sequence Diagram Use Case Pengelolaan Wilayah (cari)

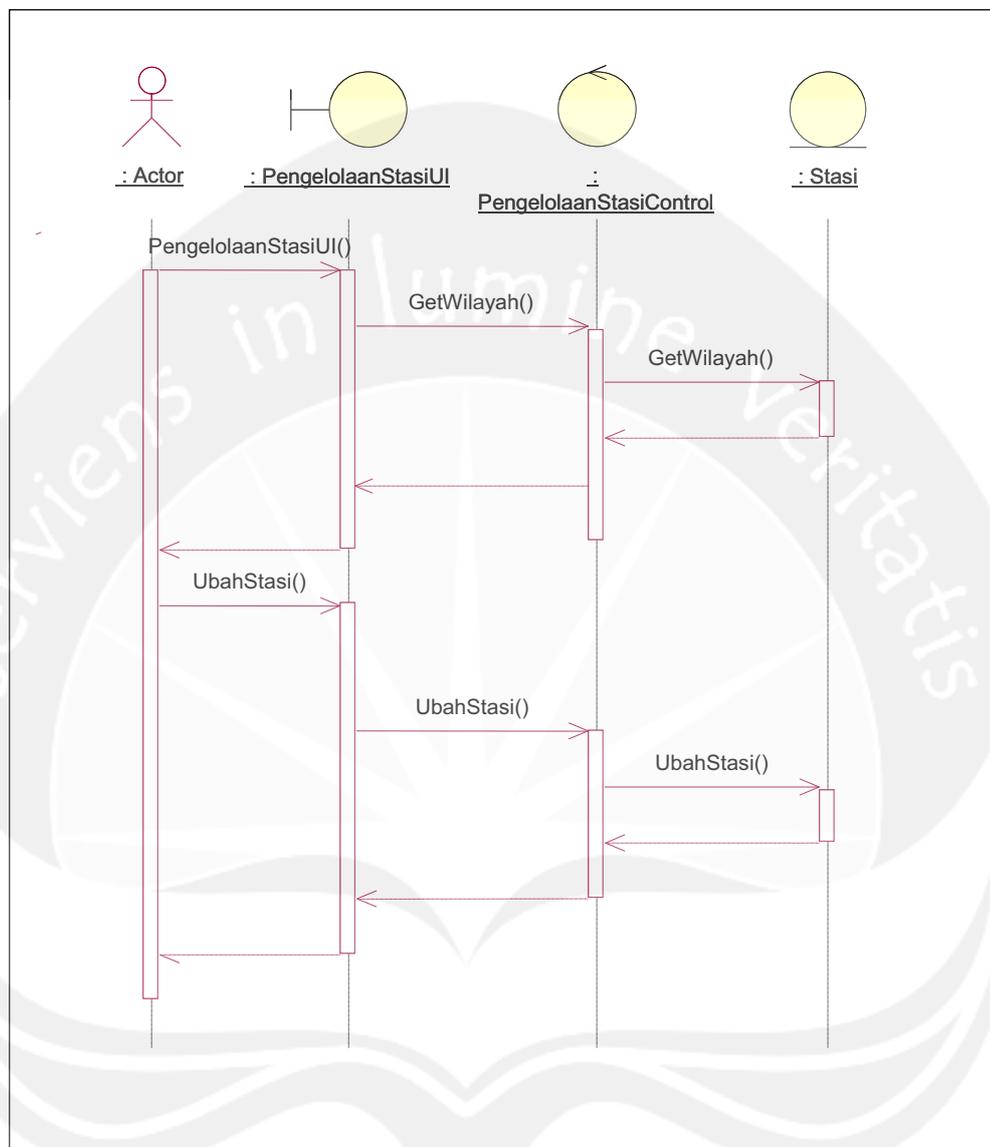
2.2.1.4 Pengelolaan Stasi

2.2.1.4.1 Pengelolaan Stasi (Tambah)



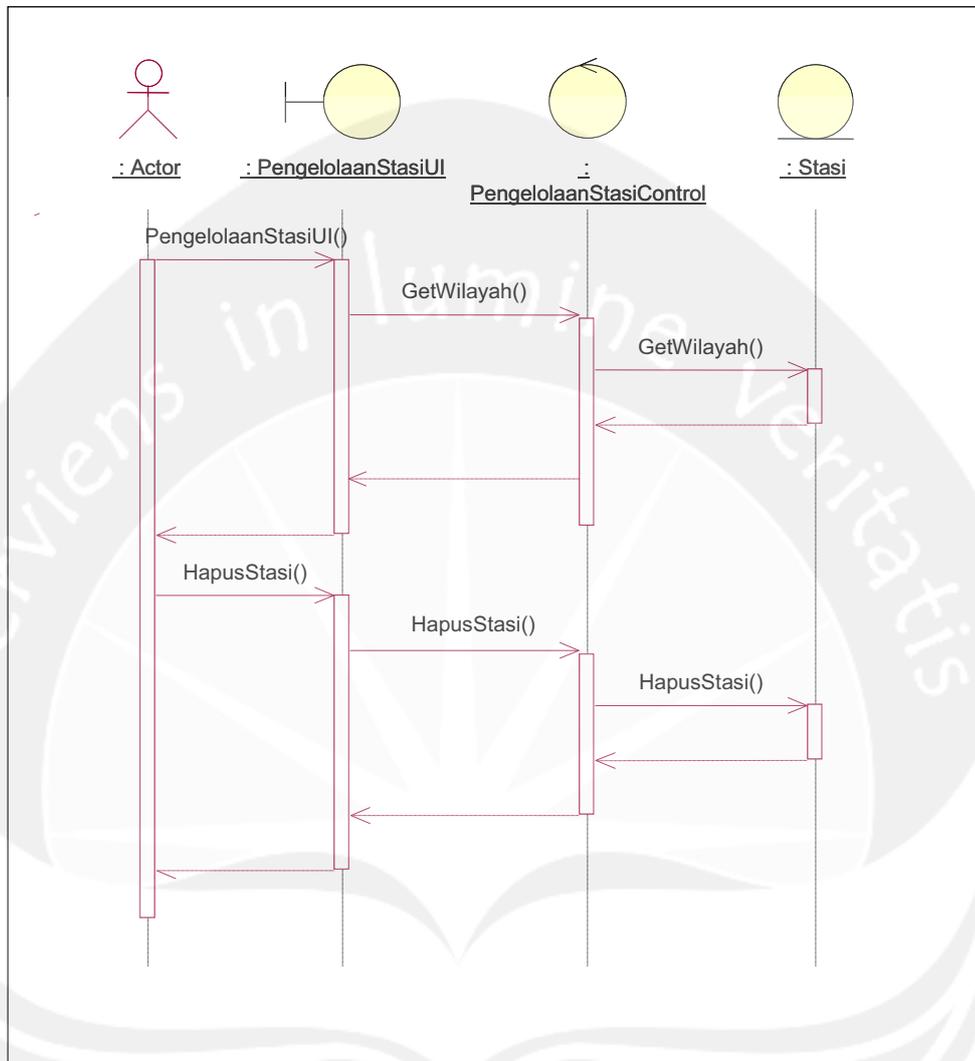
Gambar xiii: Sequence Diagram Use Case Pengelolaan Stasi (tambah)

2.2.1.4.2 Pengelolaan Stasi (Ubah)



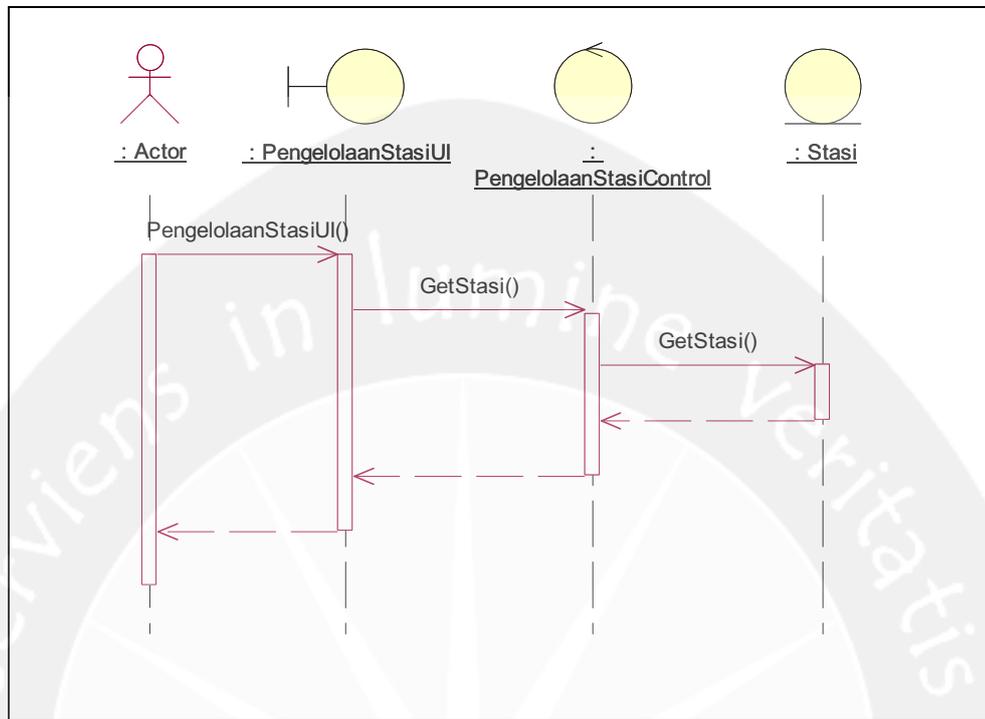
Gambar xiv: Sequence Diagram Use Case Pengelolaan Stasi (ubah)

2.2.1.4.3 Pengelolaan Stasi (Hapus)



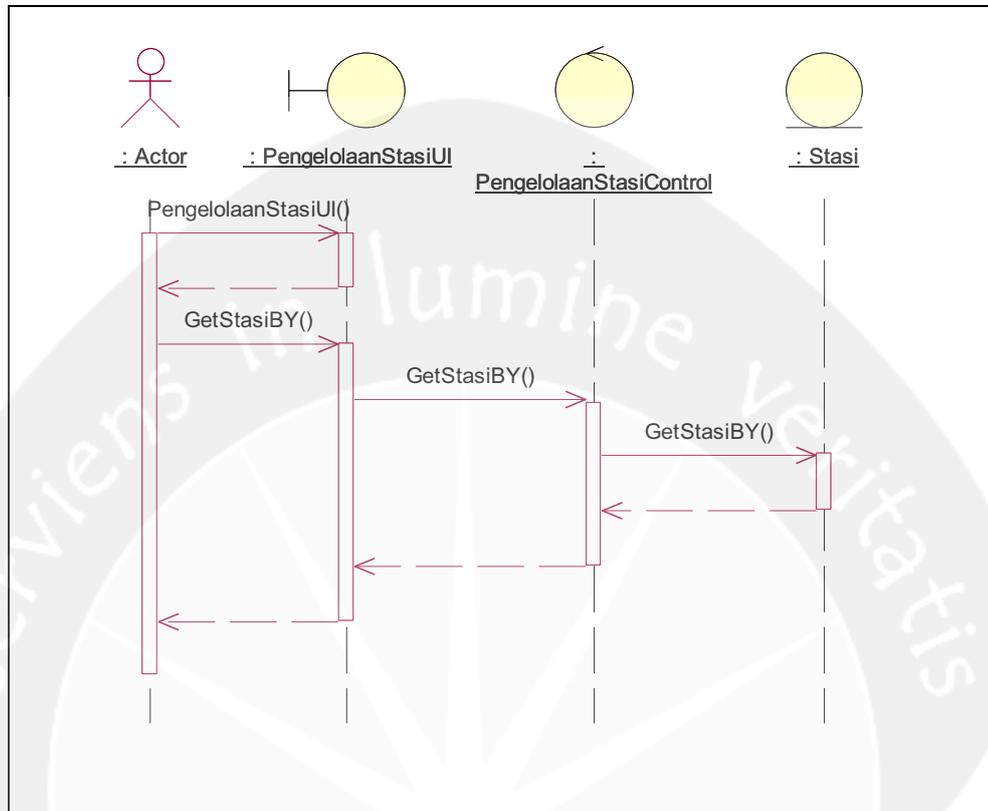
Gambar xv: Sequence Diagram Use Case Pengelolaan Stasi (hapus)

2.2.1.4.4 Pengelolaan Stasi (Tampil)



Gambar xvi: Sequence Diagram Use Case Pengelolaan Stasi (tampil)

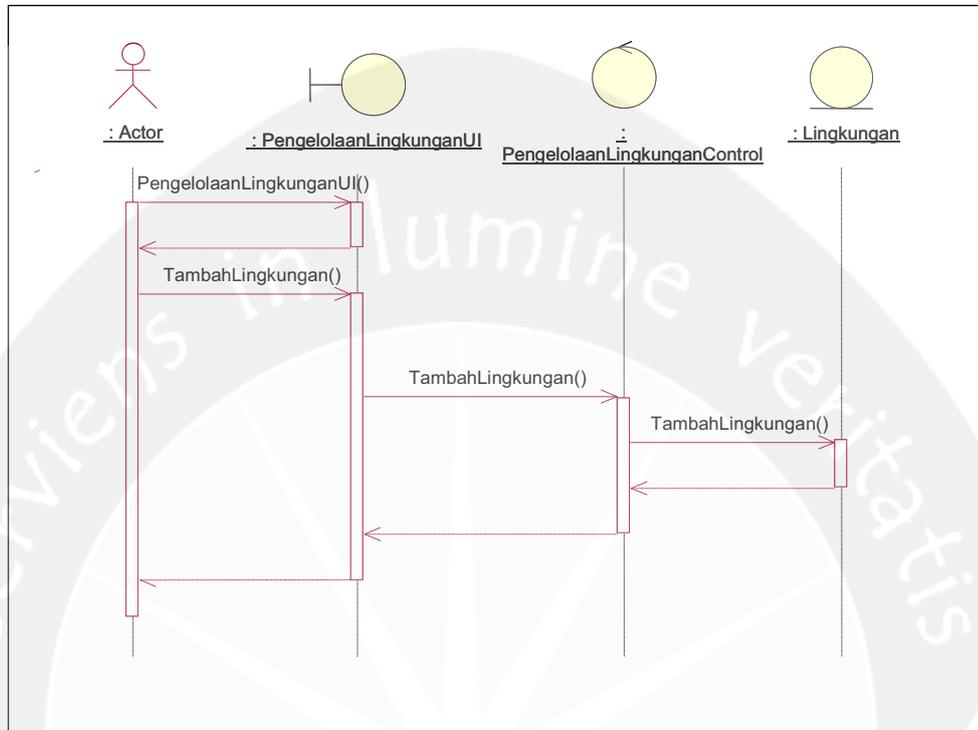
2.2.1.4.5 Pengelolaan Stasi (Cari)



Gambar xvii: Sequence Diagram Use Case Pengelolaan Stasi (cari)

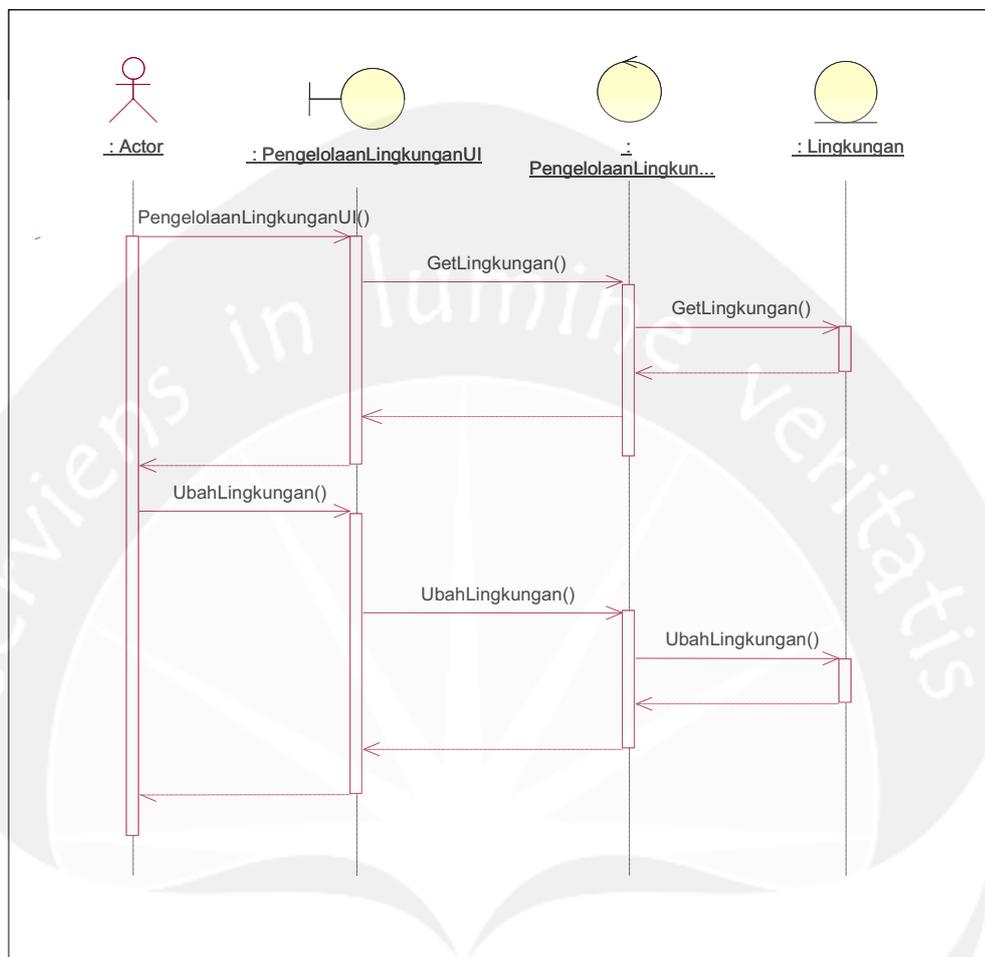
2.2.1.5 Pengelolaan Lingkungan

2.2.1.5.1 Pengelolaan Lingkungan (Tambah)



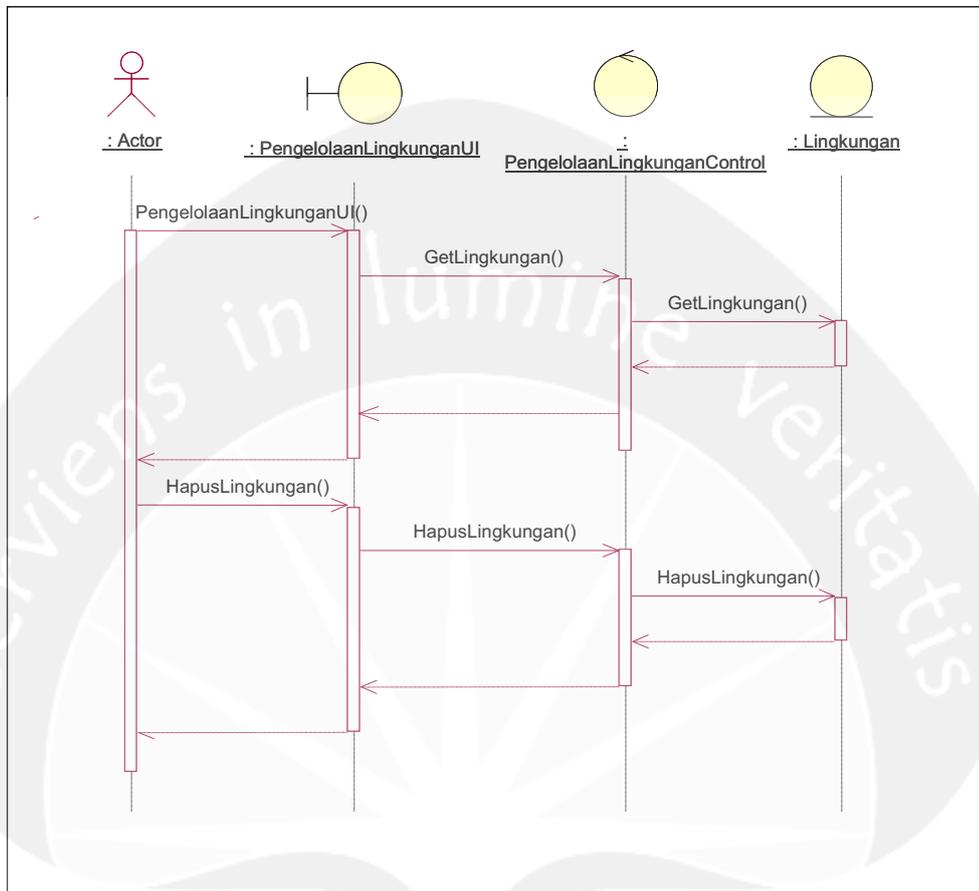
Gambar xviii: Sequence Diagram Use Case Pengelolaan Lingkungan (tambah)

2.2.1.5.2 Pengelolaan Lingkungan (Ubah)



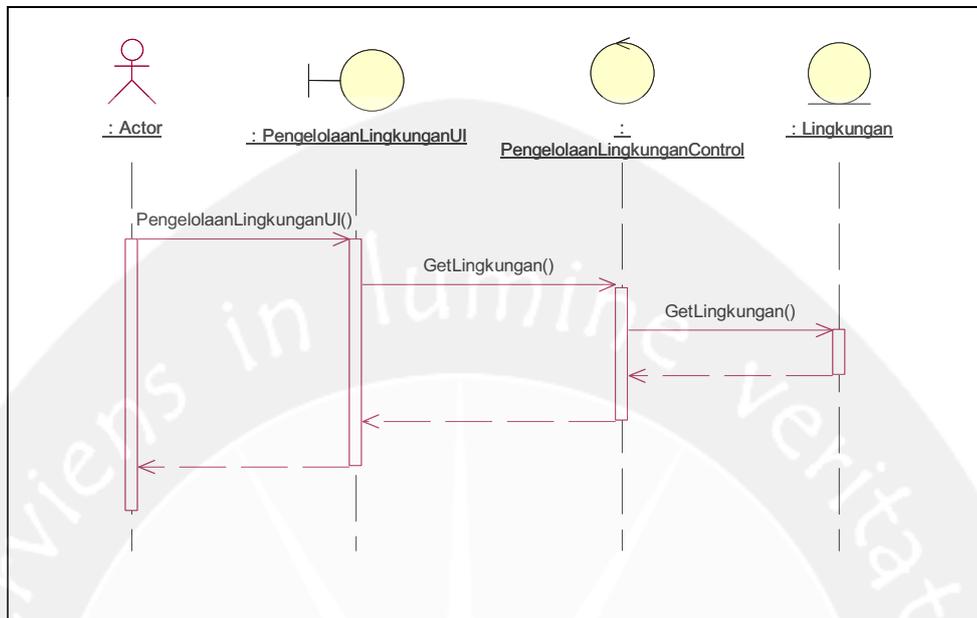
Gambar xix: Sequence Diagram Use Case Pengelolaan Lingkungan (ubah)

2.2.1.5.3 Pengelolaan Lingkungan (Hapus)



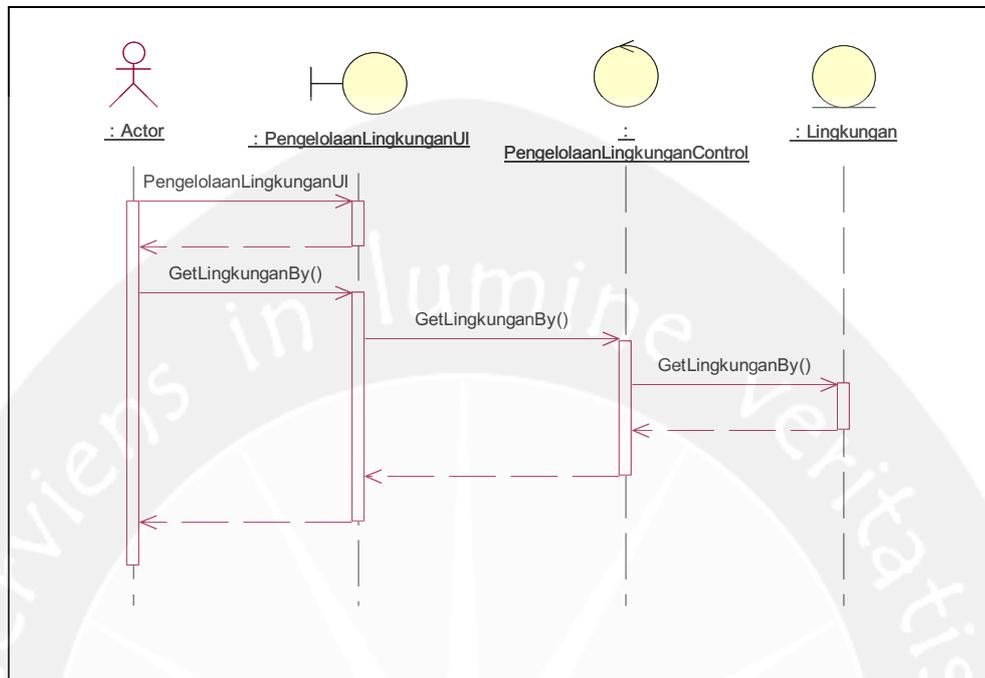
Gambar xx: Sequence Diagram Use Case Pengelolaan Lingkungan (hapus)

2.2.1.5.4 Pengelolaan Lingkungan (Tampil)



Gambar xxi: Sequence Diagram Use Case Pengelolaan Lingkungan (tampil)

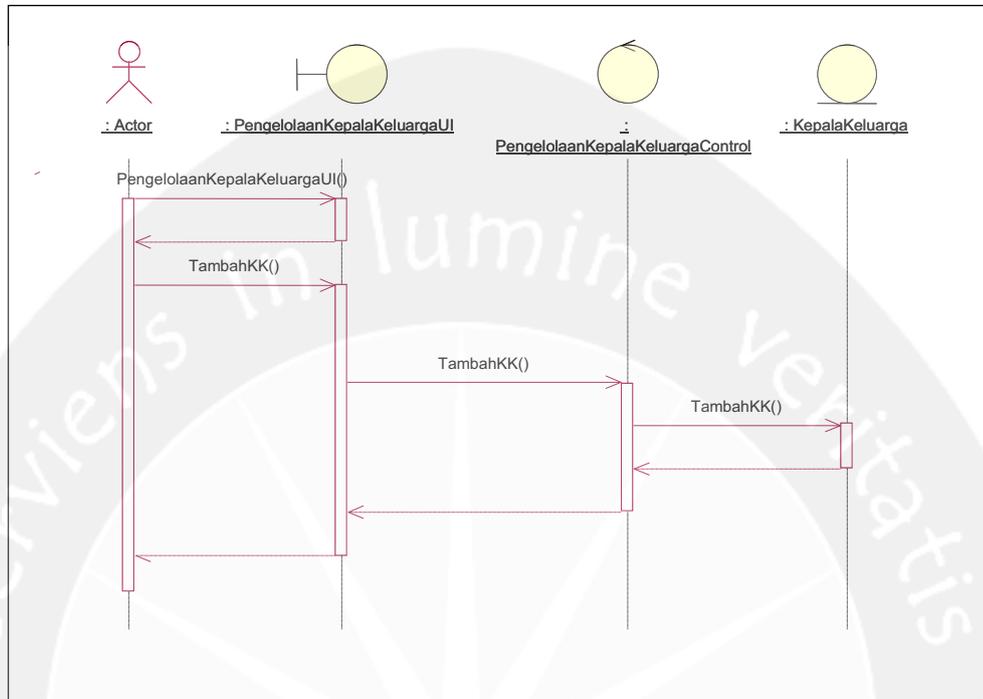
2.2.1.5.5 Pengelolaan Lingkungan (Cari)



Gambar xxii: Sequence Diagram Use Case Pengelolaan Lingkungan (cari)

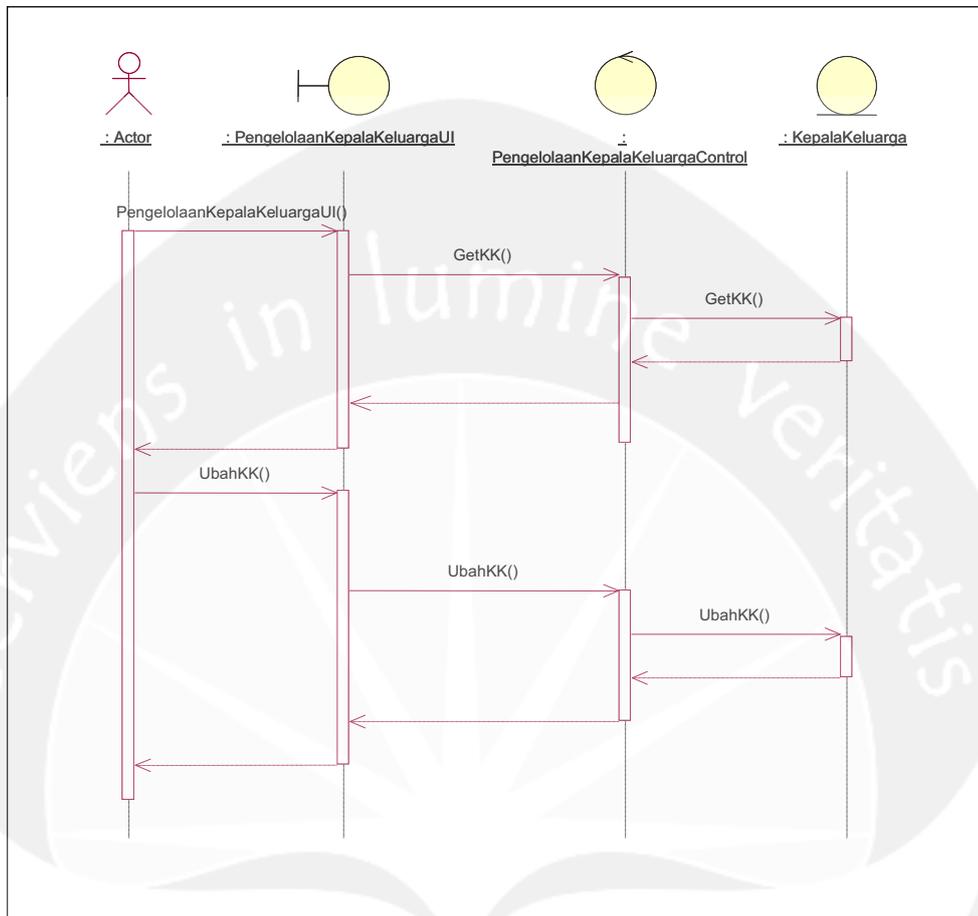
2.2.1.6 Pengelolaan Kepala Keluarga

2.2.1.6.1 Pengelolaan Kepala Keluarga (Tambah)



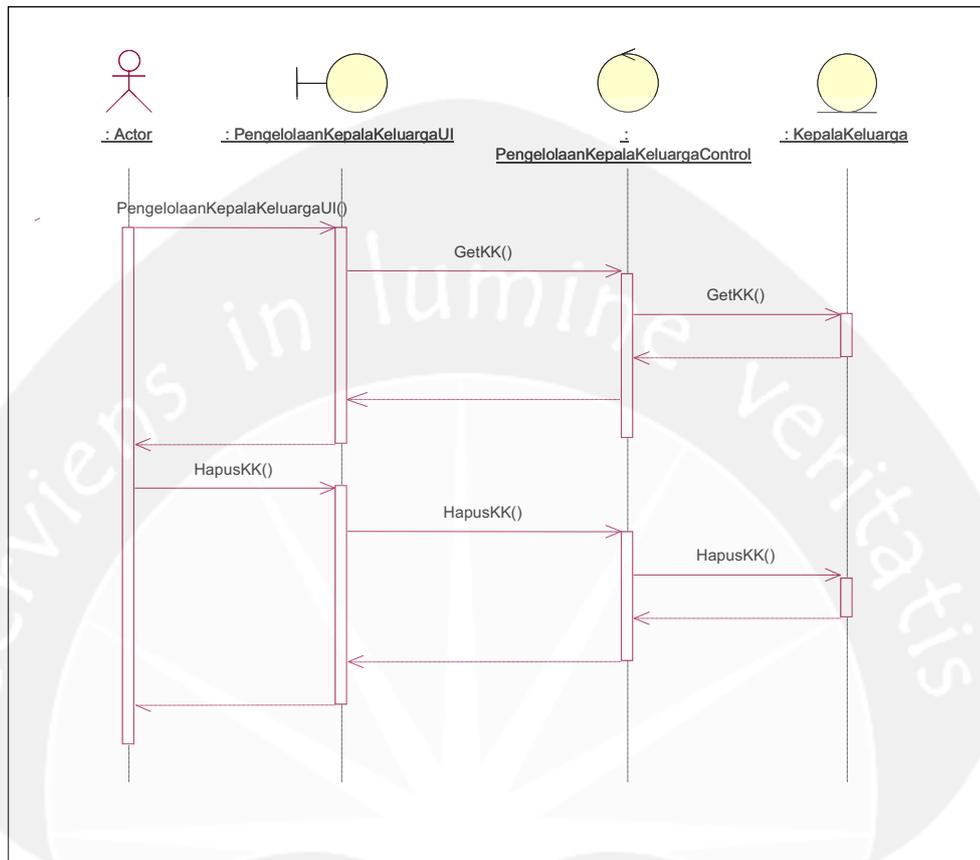
Gambar xxiii: Sequence Diagram Use Case Pengelolaan Kepala Keluarga (tambah)

2.2.1.6.2 Pengelolaan Kepala Keluarga (Ubah)



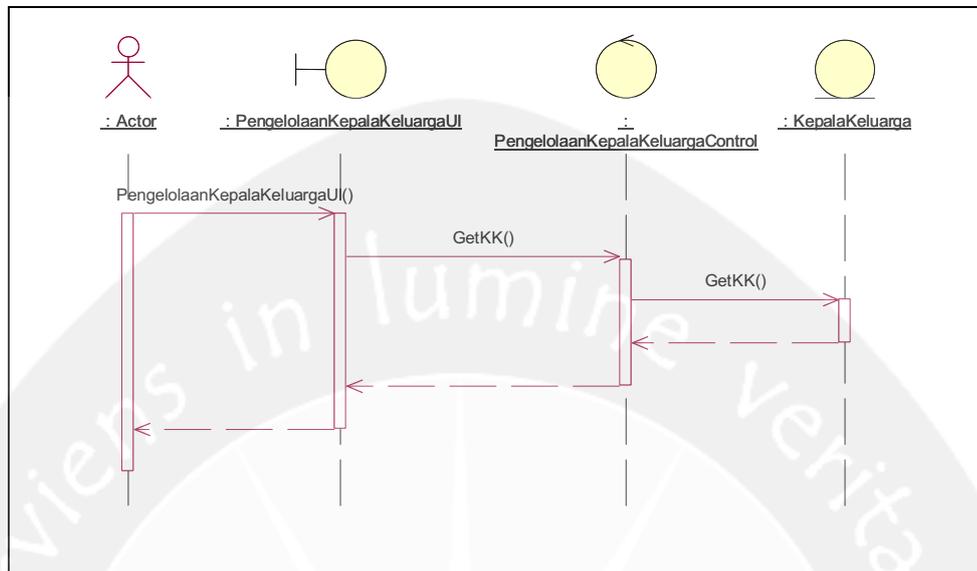
Gambar xxiv: Sequence Diagram Use Case Pengelolaan Kepala Keluarga (ubah)

2.2.1.6.3 Pengelolaan Kepala Keluarga (Hapus)



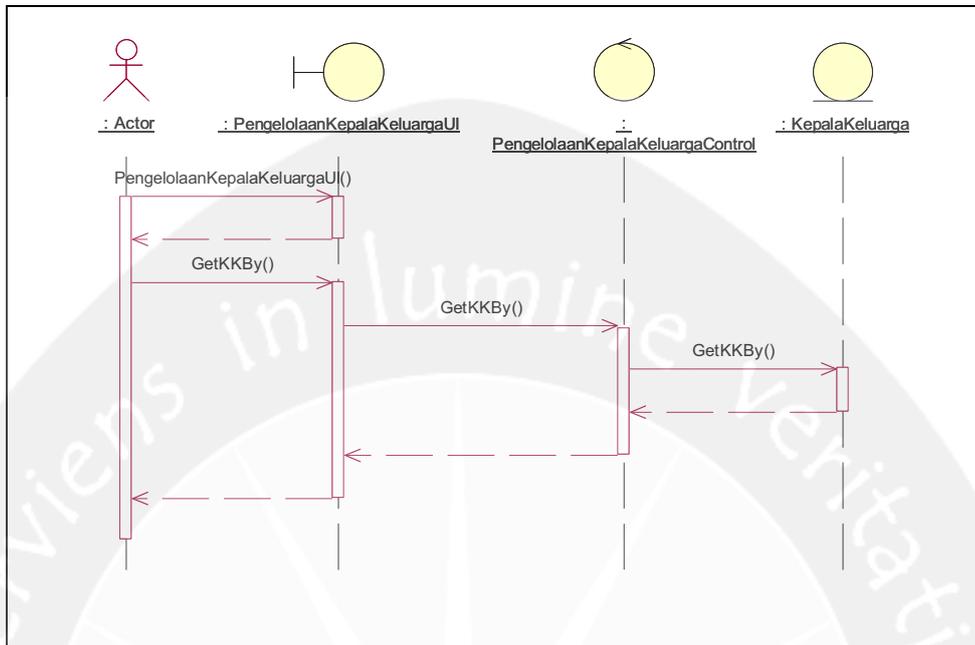
Gambar xxv: Sequence Diagram Use Case Pengelolaan Kepala Keluarga (hapus)

2.2.1.6.4 Pengelolaan Kepala Keluarga (Tampil)



Gambar xxvi: Sequence Diagram Use Case Pengelolaan Kepala Keluarga (tampil)

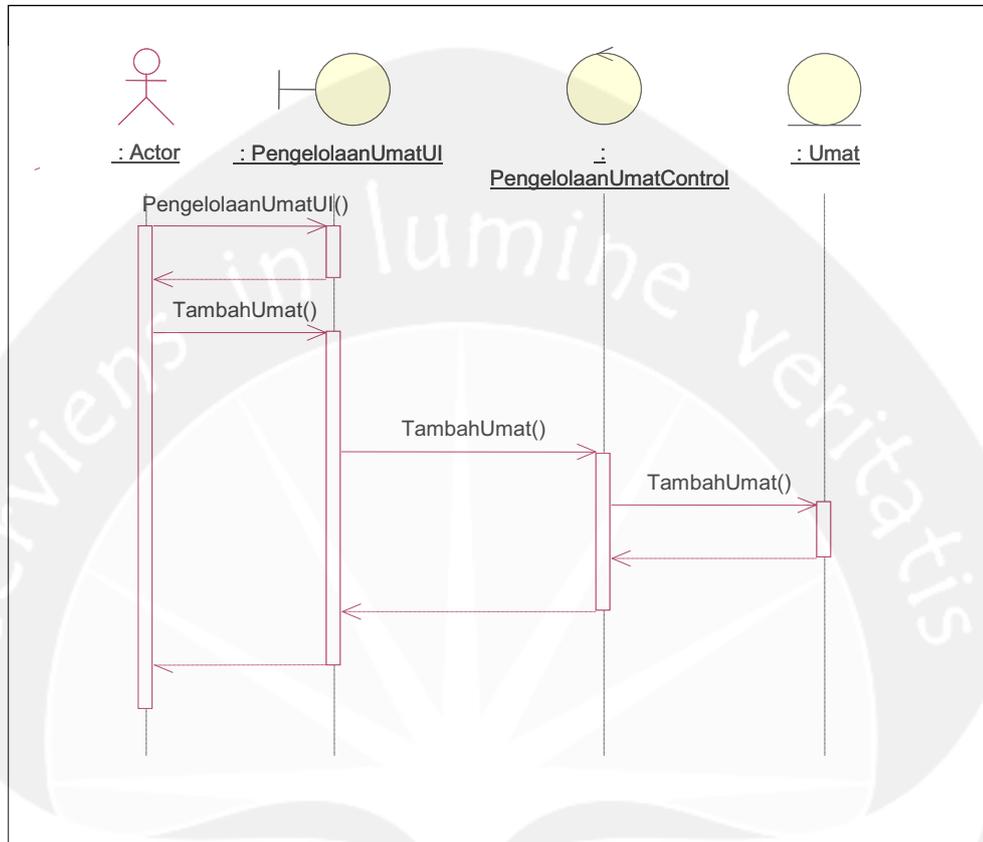
2.2.1.6.5 Pengelolaan Kepala Keluarga (Cari)



Gambar xxvii: Sequence Diagram Use Case Pengelolaan Kepala Keluarga (cari)

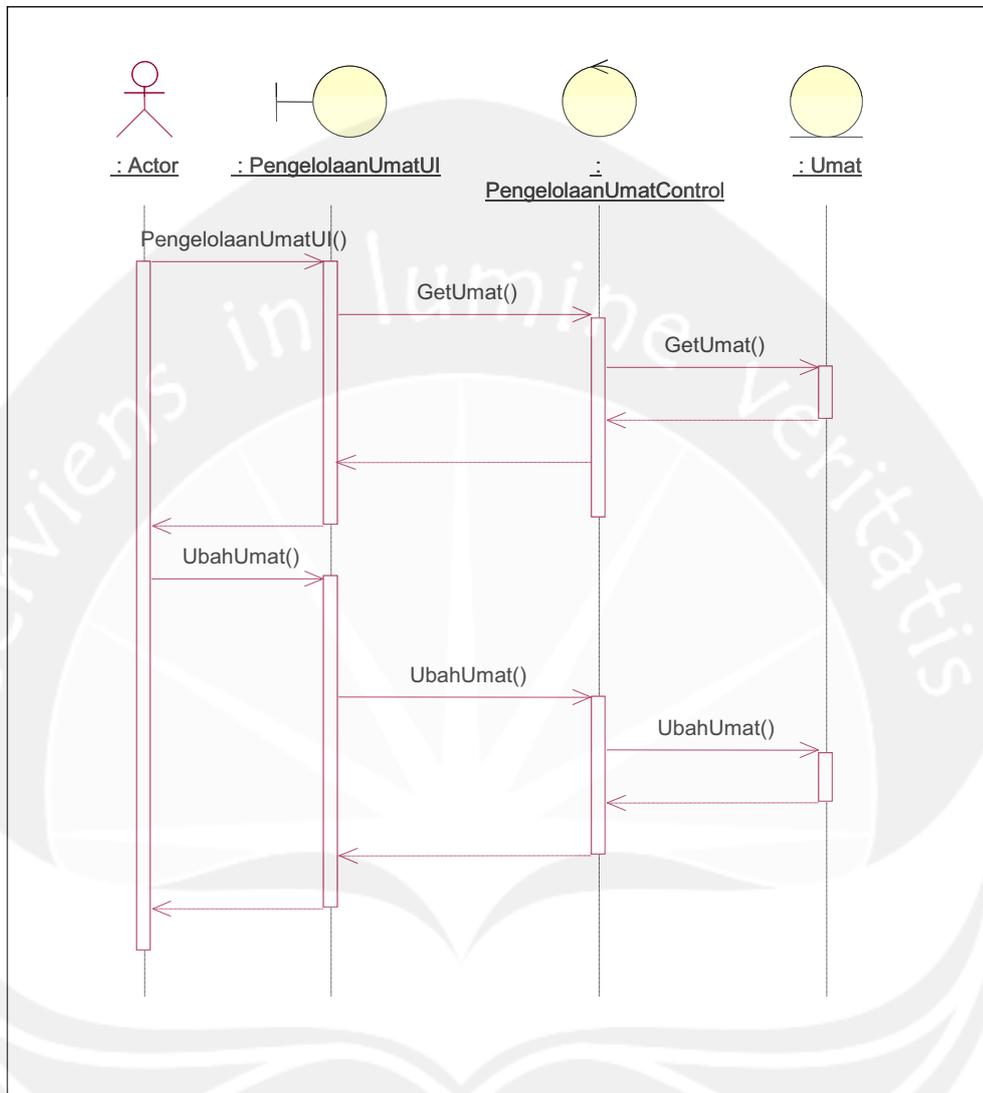
2.2.1.7 Pengelolaan Umat

2.2.1.7.1 Pengelolaan Umat (Tambah)



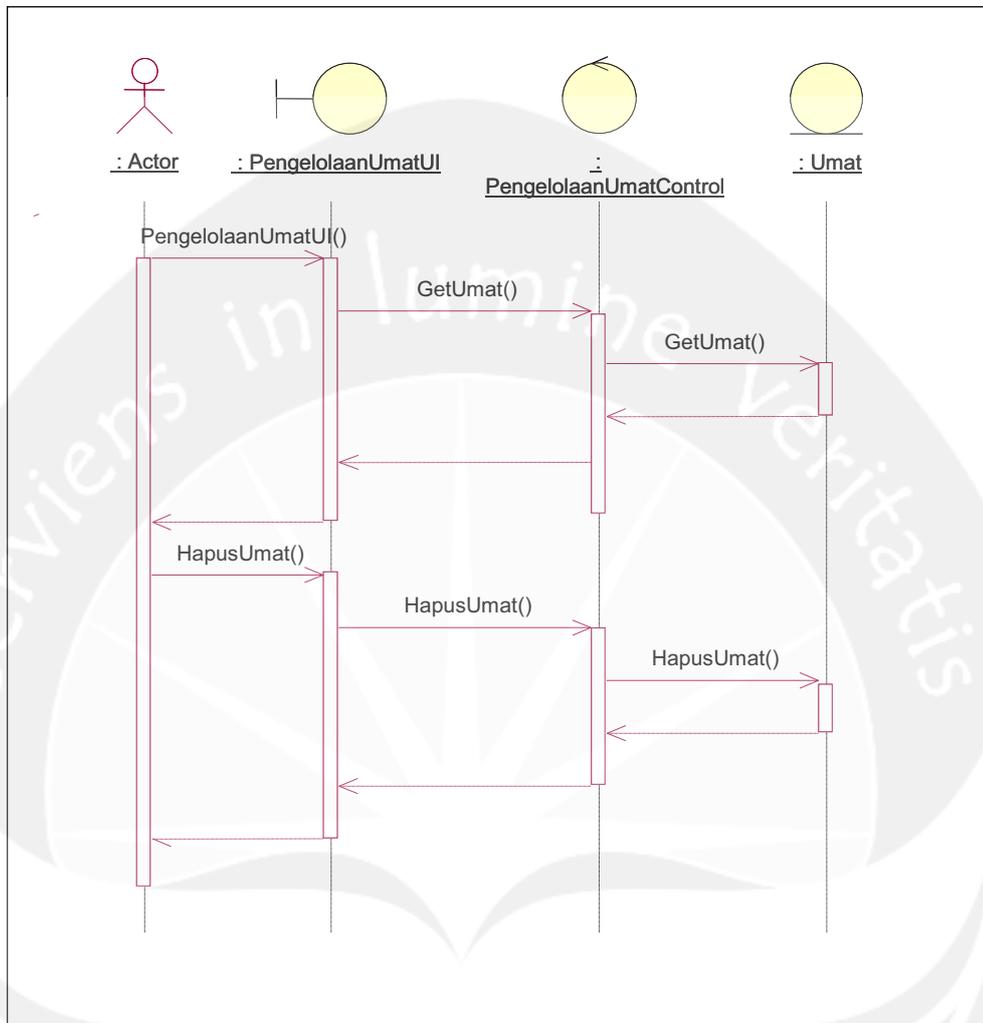
Gambar xxviii: Sequence Diagram Use Case Pengelolaan Umat (tambah)

2.2.1.7.2 Pengelolaan Umat (Ubah)



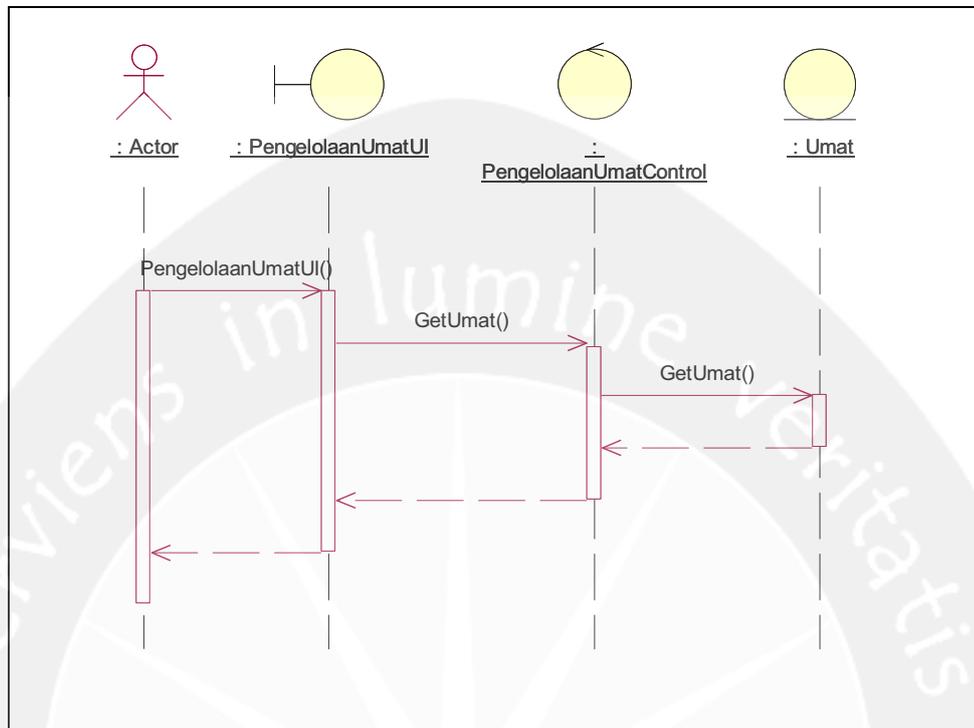
Gambar xxix: Sequence Diagram Use Case Pengelolaan Umat (ubah)

2.2.1.7.3 Pengelolaan Umat (Hapus)



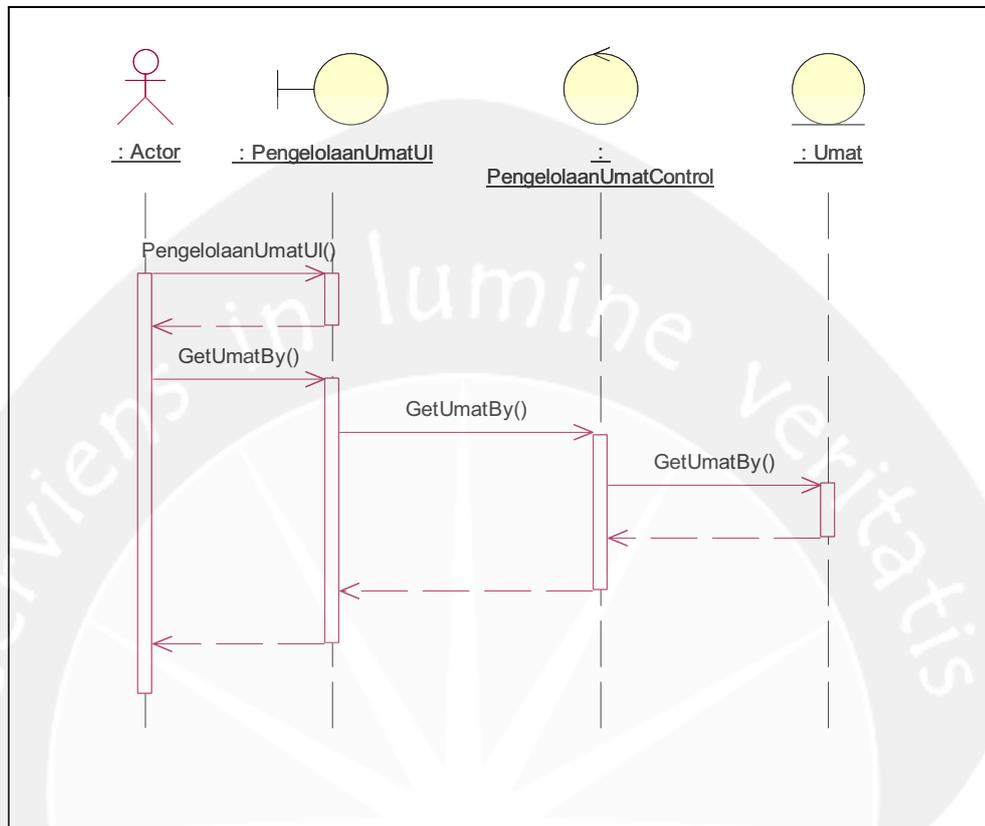
Gambar xxx: Sequence Diagram Use Case Pengelolaan Umat (hapus)

2.2.1.7.4 Pengelolaan Umat (Tampil)



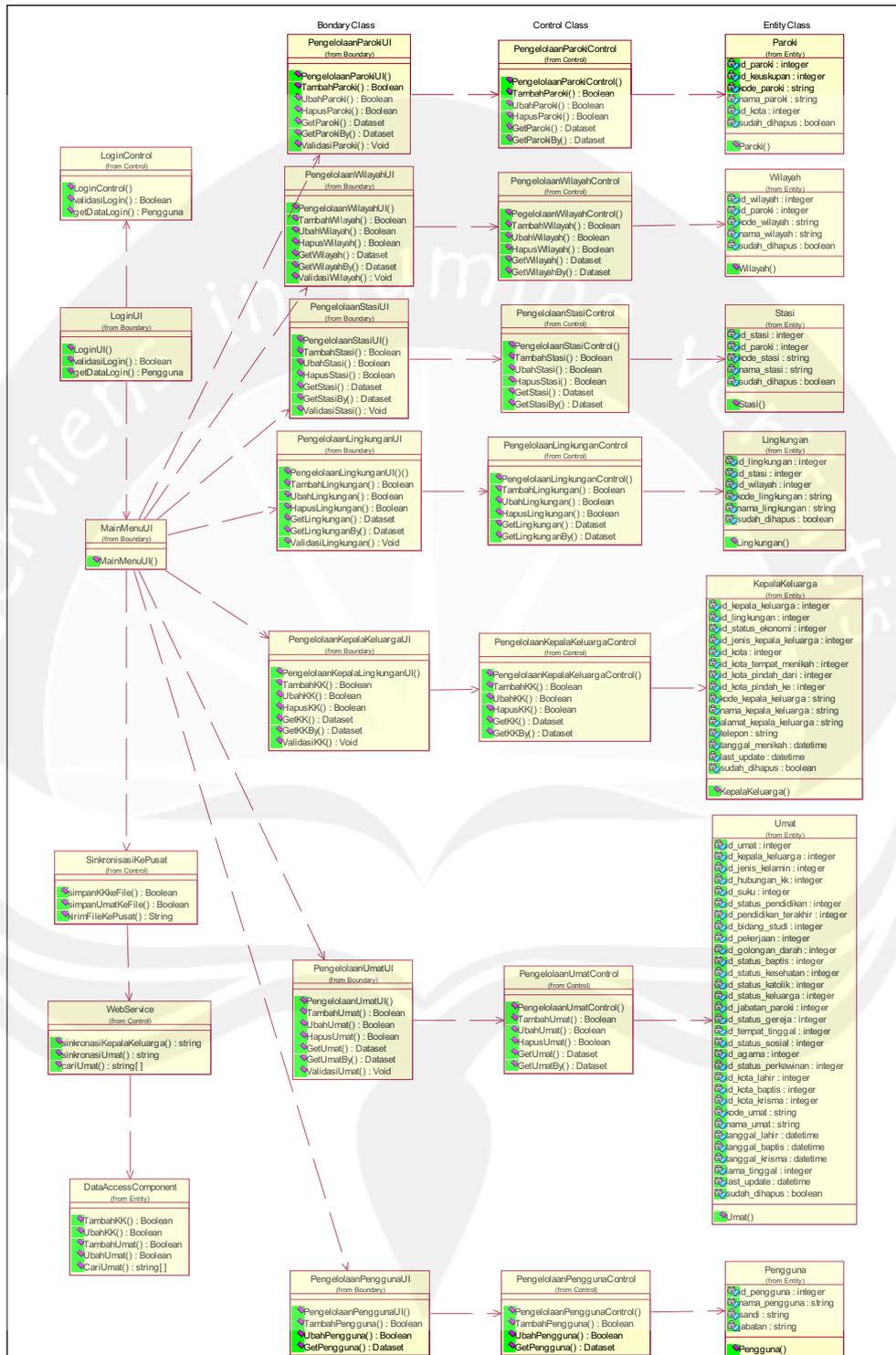
Gambar xxxi: Sequence Diagram Use Case Pengelolaan Umat (tampil)

2.2.1.7.5 Pengelolaan Umat (Cari)



Gambar xxxii: Sequence Diagram Use Case Pengelolaan Umat (cari)

2.2.2 Kelas Diagram



Gambar xxxiii: Kelas Diagram SIManja

2.2.3 Spesifikasi Deskripsi Kelas Diagram

2.2.3.1 Spesifikasi Design Kelas LoginUI

LoginUI	<<boundary>>
<pre>+LoginUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +validasiLogin() : Boolean Operasi ini digunakan untuk mengecek data Login yang diinputkan user ada di dalam basis data atau tidak dan mengecek nama dan sandi. +getDataLogin() : Pengguna Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu login id dan password.</pre>	

2.2.3.2 Spesifikasi Design Kelas MainMenuUI

MainMenuUI	<<boundary>>
<pre>+ MainMenuUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</pre>	

2.2.3.3 Spesifikasi Design Kelas PengelolaanParokiUI

PengelolaanParokiUI	<<boundary>>
<pre>+PengelolaanParokiUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +tambahParoki() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam kelas control. +ubahParoki() : boolean Operasi ini digunakan untuk mengubah data-data paroki yang</pre>	

ada di dalam basis data melalui form ini.

+hapusParoki() : boolean
 Operasi ini digunakan untuk menghapus data-data paroki yang ada di dalam basis data melalui form ini.

+getParoki() : dataset
 Operasi ini digunakan untuk mengambil data paroki-paroki yang ada di dalam basis data yang ditampilkan ke form ini.

+getParokiBy() : dataset
 Operasi ini digunakan untuk mengambil data paroki-paroki yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form ini.

2.2.3.4 Spesifikasi Design Kelas PengelolaanWilayahUI

PengelolaanWilayahUI	<<boundary>>
<p>+PengelolaanWilayahUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+tambahWilayah() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam kelas control.</p> <p>+ubahWilayah() : boolean Operasi ini digunakan untuk mengubah data-data wilayah yang ada di dalam basis data melalui form ini.</p> <p>+hapusWilayah() : boolean Operasi ini digunakan untuk menghapus data-data wilayah yang ada di dalam basis data melalui form ini.</p> <p>+getWilayah() : dataset Operasi ini digunakan untuk mengambil data wilayah-wilayah yang ada di dalam basis data yang ditampilkan ke form ini.</p> <p>+getWilayahBy() : dataset Operasi ini digunakan untuk mengambil data wilayah-wilayah yang ada di dalam basis data dengan berdasarkan kata kunci</p>	

yang dicari yang ditampilkan ke form ini.

2.2.3.5 Spesifikasi Design Kelas PengelolaanStasiunUI

PengelolaanStasiunUI	<<boundary>>
<pre> +PengelolaanStasiunUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +tambahStasiun() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam kelas control. +ubahStasiun() : boolean Operasi ini digunakan untuk mengubah data-data stasiun yang ada di dalam basis data melalui form ini. +hapusStasiun() : boolean Operasi ini digunakan untuk menghapus data-data stasiun yang ada di dalam basis data melalui form ini. +getStasiun() : dataset Operasi ini digunakan untuk mengambil data stasiun-stasiun yang ada di dalam basis data yang ditampilkan ke form ini. +getStasiunBy() : dataset Operasi ini digunakan untuk mengambil data stasiun-stasiun yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form ini. </pre>	

2.2.3.6 Spesifikasi Design Kelas PengelolaanLingkunganUI

PengelolaanLingkunganUI	<<boundary>>
<pre> +PengelolaanLingkunganUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +tambahLingkungan() : boolean Operasi ini digunakan untuk memasukkan data-data yang </pre>	

dimasukkan user dari form ke dalam kelas control.

+ubahLingkungan() : boolean
 Operasi ini digunakan untuk mengubah data-data lingkungan yang ada di dalam basis data melalui form ini.

+hapusLingkungan() : boolean
 Operasi ini digunakan untuk menghapus data-data lingkungan yang ada di dalam basis data melalui form ini.

+getLingkungan() : dataset
 Operasi ini digunakan untuk mengambil data lingkungan-lingkungan yang ada di dalam basis data yang ditampilkan ke form ini.

+getLingkunganBy() : dataset
 Operasi ini digunakan untuk mengambil data lingkungan-lingkungan yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form ini.

2.2.3.7 Spesifikasi Design Kelas PengelolaanKepala keluargaUI

PengelolaanKepala keluargaUI	<<boundary>>
<p>+PengelolaanKepala keluargaUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+tambahKepalaKeluarga() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam kelas control.</p> <p>+ubahKepalaKeluarga() : boolean Operasi ini digunakan untuk mengubah data-data kepala keluarga yang ada di dalam basis data melalui form ini.</p> <p>+hapusKepalaKeluarga() : boolean Operasi ini digunakan untuk menghapus data-data kepala keluarga yang ada di dalam basis data melalui form ini.</p> <p>+getKepalaKeluarga() : dataset</p>	

Operasi ini digunakan untuk mengambil data kepala keluarga-kepala keluarga yang ada di dalam basis data yang ditampilkan ke form ini.

+getKepalaKeluargaBy() : dataset

Operasi ini digunakan untuk mengambil data kepala keluarga-kepala keluarga yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form ini.

2.2.3.8 Spesifikasi Design Kelas PengelolaanUmatUI

PengelolaanUmatUI	<<boundary>>
<p>+PengelolaanUmatUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+tambahUmat() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam kelas control.</p> <p>+ubahUmat() : boolean Operasi ini digunakan untuk mengubah data-data umat yang ada di dalam basis data melalui form ini.</p> <p>+hapusUmat() : boolean Operasi ini digunakan untuk menghapus data-data umat yang ada di dalam basis data melalui form ini.</p> <p>+getUmat() : dataset Operasi ini digunakan untuk mengambil data umat-umat yang ada di dalam basis data yang ditampilkan ke form ini.</p> <p>+getUmatBy() : dataset Operasi ini digunakan untuk mengambil data umat-umat yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form ini.</p>	

2.2.3.9 Spesifikasi Design Kelas PengelolaanPeggunaUI

Program Studi Teknik Informatika	DPPL-SIManja	43/ 90
<p>Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika</p>		

PengelolaanPeggunaUI	<<boundary>>
<pre> +PengelolaanPeggunaUI() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +tambahPegguna() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam kelas control. +ubahPegguna() : boolean Operasi ini digunakan untuk mengubah data-data pegguna yang ada di dalam basis data melalui form ini. +getPegguna() : dataset Operasi ini digunakan untuk mengambil data pegguna- pegguna yang ada di dalam basis data yang ditampilkan ke form ini. +getPeggunaBy() : dataset Operasi ini digunakan untuk mengambil data pegguna- pegguna yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form ini. </pre>	

2.2.3.10 Spesifikasi Design Kelas LoginControl

LoginControl	<<boundary>>
<pre> +LoginControl() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +validasiLogin() : Boolean Operasi ini digunakan untuk mengecek data Login yang diinputkan user ada di dalam basis data atau tidak dan mengecek nama dan sandi. +getDataLogin() : Pengguna Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu login id dan password. </pre>	

Program Studi Teknik Informatika	DPPL-SIManja	44/ 90
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2.2.3.11 Spesifikasi Design Kelas PengelolaanParokiControl

PengelolaanParokiControl	<<control>>
<pre>+PengelolaanParokiControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +tambahParoki() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam basis data. +ubahParoki() : boolean Operasi ini digunakan untuk mengubah data-data paroki yang ada di dalam basis data melalui form. +hapusParoki() : boolean Operasi ini digunakan untuk menghapus data-data paroki yang ada di dalam basis data melalui form. +getParoki() : dataset Operasi ini digunakan untuk mengambil data paroki-paroki yang ada di dalam basis data yang ditampilkan ke form. +getParokiBy() : dataset Operasi ini digunakan untuk mengambil data paroki-paroki yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form.</pre>	

2.2.3.12 Spesifikasi Design Kelas PengelolaanWilayahControl

PengelolaanWilayahControl	<<control>>
<pre>+PengelolaanWilayahControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +tambahWilayah() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam basis data.</pre>	

```

+ubahWilayah() : boolean
Operasi ini digunakan untuk mengubah data-data wilayah yang ada di dalam basis data melalui form.
+hapusWilayah() : boolean
Operasi ini digunakan untuk menghapus data-data wilayah yang ada di dalam basis data melalui form.
+getWilayah() : dataset
Operasi ini digunakan untuk mengambil data wilayah-wilayah yang ada di dalam basis data yang ditampilkan ke form.
+getWilayahBy() : dataset
Operasi ini digunakan untuk mengambil data wilayah-wilayah yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form.

```

2.2.3.13 Spesifikasi Design Kelas PengelolaanStasiControl

PengelolaanStasiControl	<<control>>
<pre> +PengelolaanStasiControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +tambahStasi() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam basis data. +ubahStasi() : boolean Operasi ini digunakan untuk mengubah data-data stasi yang ada di dalam basis data melalui form. +hapusStasi() : boolean Operasi ini digunakan untuk menghapus data-data stasi yang ada di dalam basis data melalui form. +getStasi() : dataset Operasi ini digunakan untuk mengambil data stasi-stasi yang ada di dalam basis data yang ditampilkan ke form. +getStasiBy() : dataset </pre>	

Operasi ini digunakan untuk mengambil data stasi-stasi yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form.

2.2.3.14 Spesifikasi Design Kelas PengelolaanLingkunganControl

PengelolaanLingkunganControl	<<control>>
<p>+PengelolaanLingkunganControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+tambahLingkungan() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam basis data.</p> <p>+ubahLingkungan() : boolean Operasi ini digunakan untuk mengubah data-data lingkungan yang ada di dalam basis data melalui form.</p> <p>+hapusLingkungan() : boolean Operasi ini digunakan untuk menghapus data-data lingkungan yang ada di dalam basis data melalui form.</p> <p>+getLingkungan() : dataset Operasi ini digunakan untuk mengambil data lingkungan-lingkungan yang ada di dalam basis data yang ditampilkan ke form.</p> <p>+getLingkunganBy() : dataset Operasi ini digunakan untuk mengambil data lingkungan-lingkungan yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form.</p>	

2.2.3.15 Spesifikasi Design Kelas PengelolaanKepalaKeluargaControl

PengelolaanKepala KeluargaControl	<<control>>
--	--------------------------------

<p>+PengelolaanKepalaKeluargaControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+tambahKepalaKeluarga() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam basis data.</p> <p>+ubahKepalaKeluarga() : boolean Operasi ini digunakan untuk mengubah data-data kepala keluarga yang ada di dalam basis data melalui form.</p> <p>+hapusKepalaKeluarga() : boolean Operasi ini digunakan untuk menghapus data-data kepala keluarga yang ada di dalam basis data melalui form.</p> <p>+getKepalaKeluarga() : dataset Operasi ini digunakan untuk mengambil data kepala keluarga-kepala keluarga yang ada di dalam basis data yang ditampilkan ke form.</p> <p>+getKepalaKeluargaBy() : dataset Operasi ini digunakan untuk mengambil data kepala keluarga-kepala keluarga yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form.</p>
--

2.2.3.16 Spesifikasi Design Kelas PengelolaanUmatControl

PengelolaanUmatControl	<<control>>
<p>+PengelolaanUmatControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+tambahUmat() : boolean Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam basis data.</p> <p>+ubahUmat() : boolean</p>	

Operasi ini digunakan untuk mengubah data-data umat yang ada di dalam basis data melalui form.

+hapusUmat() : boolean

Operasi ini digunakan untuk menghapus data-data umat yang ada di dalam basis data melalui form.

+getUmat() : dataset

Operasi ini digunakan untuk mengambil data umat-umat yang ada di dalam basis data yang ditampilkan ke form.

+getUmatBy() : dataset

Operasi ini digunakan untuk mengambil data umat-umat yang ada di dalam basis data dengan berdasarkan kata kunci yang dicari yang ditampilkan ke form.

2.2.3.17 Spesifikasi Design Kelas PengelolaanPeggunaControl

PengelolaanPeggunaControl	<<control>>
<p>+PengelolaanPeggunaControl()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+tambahPegguna() : boolean</p> <p>Operasi ini digunakan untuk memasukkan data-data yang dimasukkan user dari form ke dalam basis data.</p> <p>+ubahPegguna() : boolean</p> <p>Operasi ini digunakan untuk mengubah data-data pengguna yang ada di dalam basis data melalui form.</p> <p>+getPegguna() : dataset</p> <p>Operasi ini digunakan untuk mengambil data pengguna-pengguna yang ada di dalam basis data yang ditampilkan ke form.</p> <p>+getPeggunaBy() : dataset</p> <p>Operasi ini digunakan untuk mengambil data pengguna-pengguna yang ada di dalam basis data dengan berdasarkan</p>	

kata kunci yang dicari yang ditampilkan ke form.

2.2.3.18 Spesifikasi Design Kelas Paroki

Paroki	<<entity>>
<p>-id_paroki : integer Atribut ini digunakan untuk menyimpan sementara dari id sebuah paroki yang bertipe integer.</p> <p>-id_keuskupan : integer Atribut ini digunakan untuk menyimpan sementara dari id keuskupan yang bertipe integer.</p> <p>-kode_paroki : string Atribut ini digunakan untuk menyimpan sementara dari kode paroki yang bertipe string.</p> <p>-nama_paroki : string Atribut ini digunakan untuk menyimpan sementara dari nama paroki yang bertipe string.</p> <p>-id_kota : integer Atribut ini digunakan untuk menyimpan sementara dari id kota yang bertipe integer.</p> <p>-sudah_dihapus : boolean Atribut ini digunakan untuk menyimpan sementara apakah sebuah data sudah dihapus atau belum.</p>	
<p>+Paroki() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p>	

2.2.3.19 Spesifikasi Design Kelas Wilayah

Wilayah	<<entity>>
<p>-id_wilayah : integer Atribut ini digunakan untuk menyimpan sementara dari id sebuah wilayah yang bertipe integer.</p> <p>-id_paroki : integer Atribut ini digunakan untuk menyimpan sementara dari id</p>	

Program Studi Teknik Informatika	DPPL-SIManja	50/90
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>paroki yang bertipe integer.</p> <p>-kode_wilayah : string</p> <p>Atribut ini digunakan untuk menyimpan sementara dari kode wilayah yang bertipe string.</p> <p>-nama_wilayah : string</p> <p>Atribut ini digunakan untuk menyimpan sementara dari nama wilayah yang bertipe string.</p> <p>-sudah_dihapus : boolean</p> <p>Atribut ini digunakan untuk menyimpan sementara apakah sebuah data sudah dihapus atau belum.</p>
<p>+Wilayah()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p>

2.2.3.20 Spesifikasi Design Kelas Stasi

Stasi	<<entity>>
<p>-id_stasi : integer</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id sebuah stasi yang bertipe integer.</p> <p>-id_paroki : integer</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id paroki yang bertipe integer.</p> <p>-kode_stasi : string</p> <p>Atribut ini digunakan untuk menyimpan sementara dari kode stasi yang bertipe string.</p> <p>-nama_stasi : string</p> <p>Atribut ini digunakan untuk menyimpan sementara dari nama stasi yang bertipe string.</p> <p>-sudah_dihapus : boolean</p> <p>Atribut ini digunakan untuk menyimpan sementara apakah sebuah data sudah dihapus atau belum.</p>	
<p>+Stasi()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua</p>	

atribut dari kelas ini.

2.2.3.21 Spesifikasi Design Kelas Lingkungan

Lingkungan	<<entity>>
<p>-id_lingkungan : integer Atribut ini digunakan untuk menyimpan sementara dari id sebuah lingkungan yang bertipe integer.</p> <p>-id_wilayah : integer Atribut ini digunakan untuk menyimpan sementara dari id wilayah yang bertipe integer.</p> <p>-id_stasi : integer Atribut ini digunakan untuk menyimpan sementara dari id stasi yang bertipe integer.</p> <p>-kode_lingkungan : string Atribut ini digunakan untuk menyimpan sementara dari kode lingkungan yang bertipe string.</p> <p>-nama_lingkungan : string Atribut ini digunakan untuk menyimpan sementara dari nama lingkungan yang bertipe string.</p> <p>-sudah_dihapus : boolean Atribut ini digunakan untuk menyimpan sementara apakah sebuah data sudah dihapus atau belum.</p>	
<p>+Lingkungan() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p>	

2.2.3.22 Spesifikasi Design Kelas Kepala Keluarga

KepalaKeluarga	<<entity>>
<p>-id_kepala_keluarga : integer Atribut ini digunakan untuk menyimpan sementara dari id sebuah KepalaKeluarga yang bertipe integer.</p> <p>-id_lingkungan : integer Atribut ini digunakan untuk menyimpan sementara dari id</p>	

lingkungan yang bertipe integer.

-id_status_ekonomi : integer

Atribut ini digunakan untuk menyimpan sementara dari id status ekonomi yang bertipe integer.

-id_jenis_kepala_keluarga : integer

Atribut ini digunakan untuk menyimpan sementara dari id jenis kepala keluarga yang bertipe integer.

-id_kota : integer

Atribut ini digunakan untuk menyimpan sementara dari id kota yang bertipe integer.

-id_kota_tempat_menikah : integer

Atribut ini digunakan untuk menyimpan sementara dari id kota tempat menikah yang bertipe integer.

-id_kota_pindah_dari : integer

Atribut ini digunakan untuk menyimpan sementara dari id kota pindahan dari mana keluarga tersebut yang bertipe integer.

-id_kota_pindah_ke : integer

Atribut ini digunakan untuk menyimpan sementara dari id kota pindahan ke mana keluarga tersebut yang bertipe integer.

-kode_kepala_keluarga : string

Atribut ini digunakan untuk menyimpan sementara dari kode kepala keluarga yang bertipe string.

-nama_kepala_keluarga : string

Atribut ini digunakan untuk menyimpan sementara dari nama kepala keluarga yang bertipe string.

-alamat_kepala_keluarga : string

Atribut ini digunakan untuk menyimpan sementara dari alamat kepala keluarga yang bertipe string.

-telepon : string

Atribut ini digunakan untuk menyimpan sementara dari telepon yang bertipe string.

-tanggal_menikah : datetime

<p>Atribut ini digunakan untuk menyimpan sementara dari tanggal menikah kepala keluarga yang bertipe datetime.</p> <p>-alamat_kepala_keluarga : string</p> <p>Atribut ini digunakan untuk menyimpan sementara dari alamat kepala keluarga yang bertipe string.</p> <p>-last_update : datetime</p> <p>Atribut ini digunakan untuk menyimpan sementara dari tanggal terakhir mengubah kepala keluarga yang bertipe datetime.</p> <p>-sudah_dihapus : boolean</p> <p>Atribut ini digunakan untuk menyimpan sementara apakah sebuah data sudah dihapus atau belum.</p>
<p>+KepalaKeluarga()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p>

2.2.3.23 Spesifikasi Design Kelas Umat

Umat	<<entity>>
<p>-id_umat : integer</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id sebuah umat yang bertipe integer.</p> <p>-id_kepala_keluarga : integer</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id kepala keluarga yang bertipe integer.</p> <p>-id_jenis_kelamin : integer</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id jenis kelamin yang bertipe integer.</p> <p>-id_hubungan_kk : integer</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id hubungan kepala keluarga yang bertipe integer.</p> <p>-id_suku : integer</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id suku yang bertipe integer.</p>	

-id_status_pendidikan : integer

Atribut ini digunakan untuk menyimpan sementara dari id status pendidikan yang bertipe integer.

-id_pendidikan_terakhir : integer

Atribut ini digunakan untuk menyimpan sementara dari id pendidikan terakhir yang bertipe integer.

-id_bidang_studi : integer

Atribut ini digunakan untuk menyimpan sementara dari id bidang studi yang bertipe integer.

-id_pekerjaan : integer

Atribut ini digunakan untuk menyimpan sementara dari id pekerjaan yang bertipe integer.

-id_golongan_darah : integer

Atribut ini digunakan untuk menyimpan sementara dari id golongan darah yang bertipe integer.

-id_status_baptis : integer

Atribut ini digunakan untuk menyimpan sementara dari id status baptis yang bertipe integer.

-id_status_kesehatan : integer

Atribut ini digunakan untuk menyimpan sementara dari id status kesehatan yang bertipe integer.

-id_status_katolik : integer

Atribut ini digunakan untuk menyimpan sementara dari id status katolik yang bertipe integer.

-id_status_keluarga : integer

Atribut ini digunakan untuk menyimpan sementara dari id status keluarga yang bertipe integer.

-id_jabatan_paroki : integer

Atribut ini digunakan untuk menyimpan sementara dari id jabatan paroki yang bertipe integer.

-id_status_gereja : integer

Atribut ini digunakan untuk menyimpan sementara dari id status gereja yang bertipe integer.

-id_tempat_tinggal : integer

Atribut ini digunakan untuk menyimpan sementara dari id tempat tinggal yang bertipe integer.

-id_status_sosial : integer

Atribut ini digunakan untuk menyimpan sementara dari id status sosial yang bertipe integer.

-id_agama : integer

Atribut ini digunakan untuk menyimpan sementara dari id agama yang bertipe integer.

-id_status_perkawinan : integer

Atribut ini digunakan untuk menyimpan sementara dari id status perkawinan yang bertipe integer.

-id_kota_lahir : integer

Atribut ini digunakan untuk menyimpan sementara dari id kota tempat lahir yang bertipe integer.

-id_kota_baptis : integer

Atribut ini digunakan untuk menyimpan sementara dari id kota tempat baptis yang bertipe integer.

-id_kota_krisma : integer

Atribut ini digunakan untuk menyimpan sementara dari id kota tempat krisma yang bertipe integer.

-kode_umat : string

Atribut ini digunakan untuk menyimpan sementara dari kode umat yang bertipe string.

-nama_umat : string

Atribut ini digunakan untuk menyimpan sementara dari nama umat yang bertipe string.

-tanggal_lahir : datetime

Atribut ini digunakan untuk menyimpan sementara dari tanggal lahir umat yang bertipe datetime.

-tanggal_baptis : datetime

Atribut ini digunakan untuk menyimpan sementara dari tanggal baptis umat yang bertipe datetime.

-tanggal_krisma : datetime

Atribut ini digunakan untuk menyimpan sementara dari

<p>tanggal krisma umat yang bertipe datetime.</p> <p>-lama_tinggal : integer</p> <p>Atribut ini digunakan untuk menyimpan sementara dari lama tinggal yang bertipe integer.</p> <p>-last_update : datetime</p> <p>Atribut ini digunakan untuk menyimpan sementara dari tanggal terakhir mengubah kepala keluarga yang bertipe datetime.</p> <p>-sudah_dihapus : boolean</p> <p>Atribut ini digunakan untuk menyimpan sementara apakah sebuah data sudah dihapus atau belum.</p>
<p>+Umat()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p>

2.2.3.24 Spesifikasi Design Kelas Pengguna

Pengguna	<<entity>>
<p>-id_pengguna : integer</p> <p>Atribut ini digunakan untuk menyimpan sementara dari id sebuah pengguna yang bertipe integer.</p> <p>-nama_pengguna : string</p> <p>Atribut ini digunakan untuk menyimpan sementara dari nama pengguna yang bertipe string.</p> <p>-sandi : string</p> <p>Atribut ini digunakan untuk menyimpan sementara dari sandi yang bertipe string.</p> <p>-jabatan : string</p> <p>Atribut ini digunakan untuk menyimpan sementara dari jabatan yang bertipe string.</p>	
<p>+Pengguna()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p>	

2.2.3.25 Spesifikasi Design Kelas SinkronisasiKePusat

SinkronisasiKePusat	<<control>>
<pre> +SinkronisasiKePusat() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +simpanKKkeFile() : boolean Operasi ini digunakan untuk menyimpan data kepala keluarga yang ada basis data ke sebuah file berformat cvs. +simpanUmatkeFile() : boolean Operasi ini digunakan untuk menyimpan data umat yang ada basis data ke sebuah file berformat cvs. +kirimFilekePusat() : string Operasi ini digunakan untuk melakukan sinkronisasi data kepala keluarga dan umat yang ada dalam file, ke basis data pusat. </pre>	

2.2.3.26 Spesifikasi Design Kelas Webservice

Webservice	<<control>>
<pre> +WebService() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +sinkronisasiKepalaKeluarga() : string Operasi ini digunakan untuk melakukan sinkronisasi data kepala keluarga yang ada di dalam file yang dikirim. +sinkronisasiUmat() : string Operasi ini digunakan untuk melakukan sinkronisasi data umat yang ada di dalam file yang dikirim. +cariUmat() : string[] Operasi ini diginakan untuk mencari data umat yang ada di basis data pusat. </pre>	

2.2.3.27 Spesifikasi Design Kelas DataAccessComponent

DataAccessComponent	<<control>>
<pre> +DataAccessComponent () Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +TambahKK() : string Operasi ini digunakan untuk melakukan tambah data kepala keluarga di basis data pusat. +UbahKK() : string Operasi ini digunakan untuk melakukan ubah data kepala keluarga di basis data pusat. +TambahUmat() : string Operasi ini digunakan untuk melakukan tambah data umat di basis data pusat. +UbahUmat() : string Operasi ini digunakan untuk melakukan ubah data umat di basis data pusat. +cariUmat() : string[] Operasi ini digunakan untuk mencari data umat yang ada di basis data pusat. </pre>	

3. Deskripsi Dekomposisi

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas Data Pengguna

Nama	Tipe	Panjang	Keterangan
ID Pengguna	Integer	-	Nomor dari seorang pengguna, Primary key
Nama Pengguna	VarChar	50	Nama pengguna dari seorang pengguna
Sandi	VarChar	50	Sandi dari seorang pengguna
Jabatan	VarChar	50	Jabatan di sistem dari seorang pengguna

3.1.2 Deskripsi Entitas Data Keuskupan

Nama	Tipe	Panjang	Keterangan
ID Keuskupan	integer	-	ID dari sebuah keuskupan, Primary key
Kode Keuskupan	VarChar	2	Kode dari sebuah keuskupan
Nama Keuskupan	VarChar	50	Nama dari sebuah keuskupan
Sudah Dihapus	boolean	-	Apakah sebuah keuskupan sudah dihapus atau belum

3.1.3 Deskripsi Entitas Data Paroki

Nama	Tipe	Panjang	Keterangan
ID Paroki	integer	-	ID dari sebuah paroki, Primary key
ID Keuskupan	integer	-	ID Keuskupan dari sebuah paroki, Foreign key ke tabel keuskupan
ID Kota Paroki	integer	-	ID Kota dari sebuah paroki, Foreign key ke tabel kota
Kode Paroki	VarChar	5	Nomor jenis dari sebuah barang, Foreign key ke tabel jenis barang
Nama Paroki	VarChar	50	Nama dari sebuah

Program Studi Teknik Informatika	DPPL-SIManja	60/90
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		

			paroki
Sudah Dihapus	boolean	-	Apakah sebuah paroki sudah dihapus atau belum
Paroki Terpilih	boolean	-	Apakah sebuah paroki merupakan paroki terpilih atau tidak

3.1.4 Deskripsi Entitas Data Wilayah

Nama	Tipe	Panjang	Keterangan
ID Wilayah	integer	-	ID dari sebuah wilayah, Primary key
ID Paroki	integer	-	ID paroki dari sebuah wilayah, Foreign key ke tabel paroki
Kode Wilayah	VarChar	8	Kode dari sebuah wilayah
Nama Wilayah	VarChar	50	Nama dari sebuah wilayah
Sudah Dihapus	boolean	-	Apakah sebuah wilayah sudah dihapus atau belum

3.1.5 Deskripsi Entitas Data Stasi

Nama	Tipe	Panjang	Keterangan
ID Stasi	integer	-	ID dari sebuah stasi, Primary key
ID Paroki	integer	-	ID paroki dari sebuah stasi, Foreign key ke tabel paroki
Kode Stasi	VarChar	8	Kode dari sebuah stasi
Nama Stasi	VarChar	50	Nama dari sebuah stasi
Sudah Dihapus	boolean	-	Apakah sebuah stasi sudah dihapus atau belum

3.1.6 Deskripsi Entitas Data Lingkungan

Nama	Tipe	Panjang	Keterangan
ID Lingkungan	integer	-	ID dari sebuah lingkungan, Primary key
ID Wilayah	integer	-	ID wilayah dari

Program Studi Teknik Informatika	DPPL-SIManja	61/ 90
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		

			sebuah lingkungan, Foreign key ke tabel wilayah (bisa kosong)
ID Stasi	integer	-	ID stasi dari sebuah lingkungan, Foreign key ke tabel stasi (bisa kosong)
Kode Lingkungan	VarChar	11	Kode dari sebuah lingkungan
Nama Lingkungan	VarChar	50	Nama dari sebuah lingkungan
Sudah Dihapus	boolean	-	Apakah sebuah lingkungan sudah dihapus atau belum

3.1.7 Deskripsi Entitas Data Ref Status Ekonomi

Nama	Tipe	Panjang	Keterangan
ID Status Ekonomi	integer	-	ID dari sebuah status ekonomi, Primary key
Status Ekonomi	VarChar	50	Nama dari sebuah status ekonomi

3.1.8 Deskripsi Entitas Data Ref Jenis Kepala Keluarga

Nama	Tipe	Panjang	Keterangan
ID Jenis Kepala Keluarga	integer	-	ID dari sebuah jenis kepala keluarga, Primary key
Jenis Kepala Keluarga	VarChar	50	Nama dari sebuah jenis kepala keluarga

3.1.9 Deskripsi Entitas Data Kepala Keluarga

Nama	Tipe	Panjang	Keterangan
ID Kepala Keluarga	integer	-	ID dari seorang kepala keluarga, Primary key
ID Lingkungan	integer	-	ID lingkungan dari seorang kepala keluarga, Foreign key ke tabel lingkungan
ID Status Ekonomi	integer	-	ID status ekonomi dari seorang kepala

Program Studi Teknik Informatika	DPPL-SIManja	62/ 90
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		

			keluarga, Foreign key ke tabel ref status ekonomi
ID Jenis Kepala Keluarga	integer	-	ID jenis kepala keluarga dari seorang kepala keluarga, Foreign key ke tabel ref jenis kepala keluarga
ID Kota KK	integer	-	ID kota dari seorang kepala keluarga, Foreign key ke tabel kota
ID Kota Tempat Menikah	integer	-	ID kota tempat menikah seorang kepala keluarga, Foreign key ke tabel kota
ID Kota Pindah Dari	integer	-	ID kota keterangan kepala keluarga pindah dari mana, Foreign key ke tabel kota
ID Kota Pindah Ke	integer	-	ID kota keterangan kepala keluarga pindah ke mana, Foreign key ke tabel kota
Nama Kepala Keluarga	VarChar	50	Nama dari seorang kepala keluarga
Alamat Kepala Keluarga	VarChar	100	Alamat dari seorang kepala keluarga
Telepon	VarChar	20	Telepon dari sebuah OP
Tanggal Menikah	Date	-	Tanggal menikah dari seorang kepala keluarga
Last Update	Date	-	Tanggal terakhir diubah dari seorang kepala keluarga
Sudah Dihapus	boolean	-	Apakah seorang kepala keluarga sudah dihapus atau belum

3.1.10 Deskripsi Entitas Data Ref Jenis Kelamin

Nama	Tipe	Panjang	Keterangan
ID Jenis Kelamin	integer	-	ID dari sebuah jenis kelamin, Primary key
Jenis Kelamin	VarChar	20	Nama dari sebuah jenis kelamin

3.1.11 Deskripsi Entitas Data Ref Status Kesehatan

Nama	Tipe	Panjang	Keterangan
ID Status Kesehatan	integer	-	ID dari sebuah status kesehatan, Primary key
Status Kesehatan	VarChar	50	Nama dari sebuah status kesehatan

3.1.12 Deskripsi Entitas Data Ref Hubungan Kepala Keluarga

Nama	Tipe	Panjang	Keterangan
ID Hubungan Kepala Keluarga	integer	-	ID dari sebuah Hubungan Kepala Keluarga, Primary key
Hubungan Kepala Keluarga	VarChar	50	Nama dari sebuah hubungan kepala keluarga

3.1.13 Deskripsi Entitas Data Ref Status Perkawinan

Nama	Tipe	Panjang	Keterangan
ID Status Perkawinan	integer	-	ID dari sebuah status perkawinan, Primary key
Status Perkawinan	VarChar	50	Nama dari sebuah status perkawinan

3.1.14 Deskripsi Entitas Data Ref Suku

Nama	Tipe	Panjang	Keterangan
ID Suku	integer	-	ID dari sebuah suku, Primary key
Suku	VarChar	50	Nama dari sebuah suku

3.1.15 Deskripsi Entitas Data Ref Agama

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

Program Studi Teknik Informatika	DPPL-SIManja	64/ 90
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 Agama	integer	-	ID dari sebuah agama, Primary key
Agama	VarChar	50	Nama dari sebuah agama

3.1.16 Deskripsi Entitas Data Ref Status Pendidikan

Nama	Tipe	Panjang	Keterangan
ID Status Pendidikan	integer	-	ID dari sebuah status pendidikan, Primary key
Status Pendidikan	VarChar	50	Nama dari sebuah status pendidikan

3.1.17 Deskripsi Entitas Data Ref Status Sosial

Nama	Tipe	Panjang	Keterangan
ID Status Sosial	integer	-	ID dari sebuah status sosial, Primary key
Status Sosial	VarChar	50	Nama dari sebuah status sosial

3.1.18 Deskripsi Entitas Data Ref Pendidikan Terakhir

Nama	Tipe	Panjang	Keterangan
ID Pendidikan Terakhir	integer	-	ID dari sebuah pendidikan terakhir, Primary key
Pendidikan terakhir	VarChar	50	Nama dari sebuah pendidikan terakhir

3.1.19 Deskripsi Entitas Data Ref Tempat Tinggal

Nama	Tipe	Panjang	Keterangan
ID Tempat Tinggal	integer	-	ID dari sebuah tempat tinggal, Primary key
Tempat Tinggal	VarChar	50	Nama dari sebuah tempat tinggal

3.1.20 Deskripsi Entitas Data Ref Bidang Studi

Nama	Tipe	Panjang	Keterangan
ID Bidang Studi	integer	-	ID dari sebuah bidang studi, Primary key
Bidang Studi	VarChar	50	Nama dari sebuah

Program Studi Teknik Informatika	DPPL-SIManja	65/ 90
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		

			bidang studi
--	--	--	--------------

3.1.21 Deskripsi Entitas Data Ref Status Gereja

Nama	Tipe	Panjang	Keterangan
ID Status Gereja	integer	-	ID dari sebuah status gereja, Primary key
Status Gereja	VarChar	50	Nama dari sebuah status gereja

3.1.22 Deskripsi Entitas Data Ref Pekerjaan

Nama	Tipe	Panjang	Keterangan
ID Pekerjaan	integer	-	ID dari sebuah pekerjaan, Primary key
Pekerjaan	VarChar	50	Nama dari sebuah pekerjaan

3.1.23 Deskripsi Entitas Data Ref Jabatan Paroki

Nama	Tipe	Panjang	Keterangan
ID Jabatan Paroki	integer	-	ID dari sebuah jabatan paroki, Primary key
Jabatan Paroki	VarChar	50	Nama dari sebuah jabatan paroki

3.1.24 Deskripsi Entitas Data Ref Golongan Darah

Nama	Tipe	Panjang	Keterangan
ID Golongan Darah	integer	-	ID dari sebuah golongan darah, Primary key
Golongan Darah	VarChar	3	Nama dari sebuah golongan darah

3.1.25 Deskripsi Entitas Data Ref Status Keluarga

Nama	Tipe	Panjang	Keterangan
ID Status Keluarga	integer	-	ID dari sebuah status keluarga, Primary key
Status Keluarga	VarChar	50	Nama dari sebuah status keluarga

3.1.26 Deskripsi Entitas Data Ref Status Baptis

Nama	Tipe	Panjang	Keterangan
ID Status Baptis	integer	-	ID dari sebuah status baptis, Primary key
Status Baptis	VarChar	50	Nama dari sebuah status baptis

3.1.27 Deskripsi Entitas Data Ref Status Katolik

Nama	Tipe	Panjang	Keterangan
ID Status Katolik	integer	-	ID dari sebuah status katolik, Primary key
Status Katolik	VarChar	50	Nama dari sebuah status katolik

3.1.28 Deskripsi Entitas Data Umat

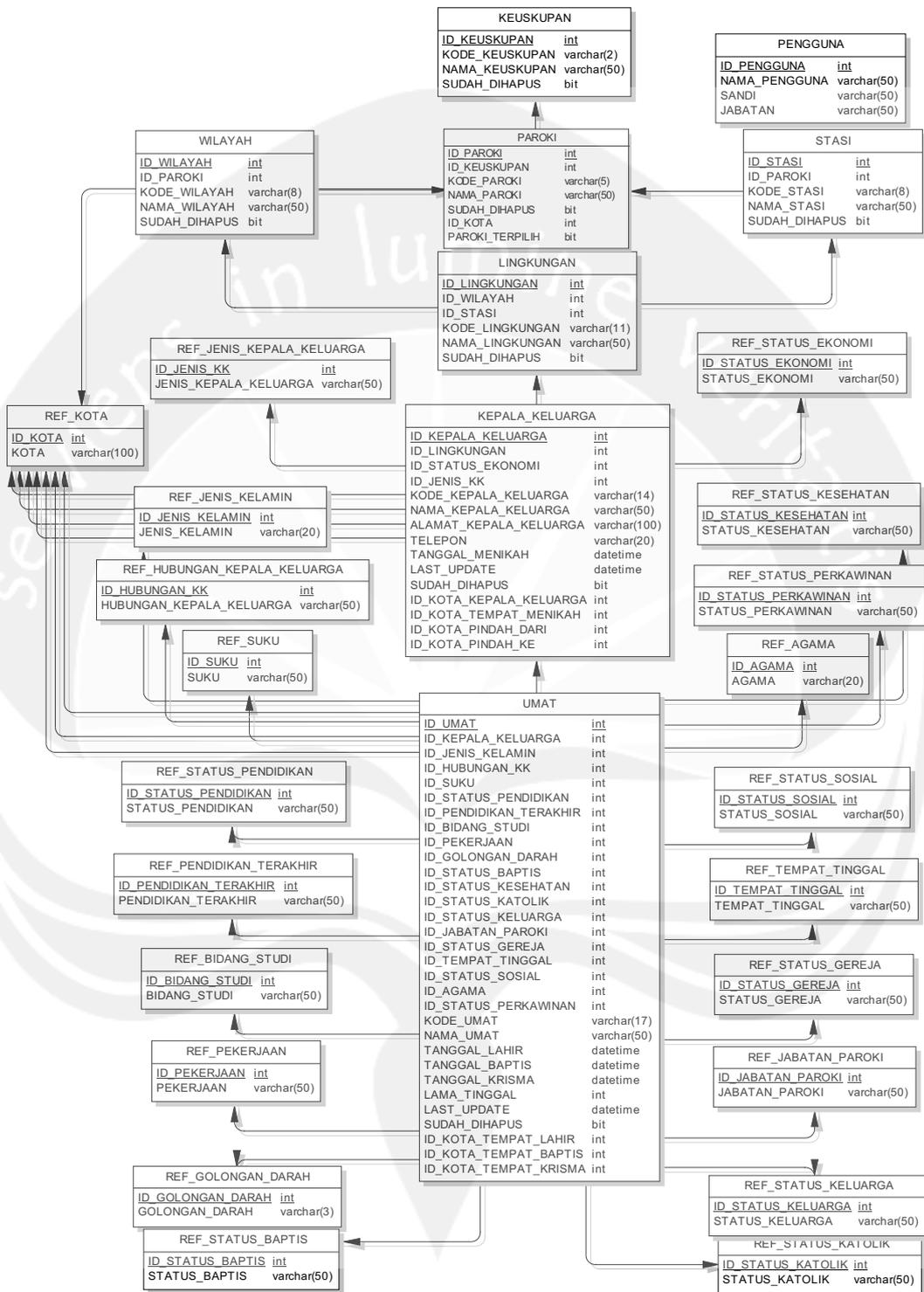
Nama	Tipe	Panjang	Keterangan
ID Umat	integer	-	ID dari seorang umat, Primary key
ID Kepala Keluarga	integer	-	ID kepala keluarga dari seorang umat, Foreign key ke tabel kepala keluarga
ID Jenis Kelamin	integer	-	ID jenis kelamin dari seorang umat, Foreign key ke tabel ref jenis kelamin
ID Hubungan KK	integer	-	ID hubungan kepala keluarga dari seorang umat, Foreign key ke tabel ref hubungan kepala keluarga
ID Suku	integer	-	ID suku dari seorang umat, Foreign key ke tabel ref suku
ID Status Pendidikan	integer	-	ID status pendidikan dari seorang umat, Foreign key ke tabel ref status pendidikan
ID Pendidikan Terakhir	integer	-	ID pendidikan terakhir dari seorang umat,

			Foreign key ke tabel ref pendidikan terakhir
ID Bidang Studi	integer	-	ID bidang studi dari seorang umat, Foreign key ke tabel ref bidang studi
ID Pekerjaan	integer	-	ID pekerjaan dari seorang umat, Foreign key ke tabel ref pekerjaan
ID Golongan Darah	integer	-	ID golongan darah dari seorang umat, Foreign key ke tabel ref golongan darah
ID Status Baptis	integer	-	ID status baptis dari seorang umat, Foreign key ke tabel ref status baptis
ID Status Kesehatan	integer	-	ID status kesehatan dari seorang umat, Foreign key ke tabel ref status kesehatan
ID Status Katolik	integer	-	ID status katolik dari seorang umat, Foreign key ke tabel ref status katolik
ID Status Keluarga	integer	-	ID status keluarga dari seorang umat, Foreign key ke tabel ref jenis kelamin
ID Jabatan Paroki	integer	-	ID jabatan paroki dari seorang umat, Foreign key ke tabel ref jabatan paroki
ID Status Gereja	integer	-	ID status gereja dari seorang umat, Foreign key ke tabel ref status gereja
ID Tempat Tinggal	integer	-	ID tempat tinggal dari seorang umat, Foreign key ke tabel ref tempat tinggal
ID Status Sosial	integer	-	ID status sosial dari seorang umat, Foreign key ke tabel ref status sosial
ID Agama	integer	-	ID agama dari

Program Studi Teknik Informatika	DPPL-SIManja	68/90
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		

			seorang umat, Foreign key ke tabel ref agama
ID Status Perkawinan	integer	-	ID Status Perkawinan dari seorang umat, Foreign key ke tabel ref status perkawinan
ID Kota Tempat Lahir	integer	-	ID kota lahir dari seorang umat, Foreign key ke tabel kota
ID Kota Tempat Baptis	integer	-	ID kota baptis dari seorang umat, Foreign key ke tabel kota
ID Kota Tempat Krisma	integer	-	ID kota krisma dari seorang umat, Foreign key ke tabel kota
Kode Umat	VarChar	17	Kode dari seorang umat
Nama Umat	VarChar	50	Nama dari seorang umat
Tanggal Lahir	Date	-	Tanggal lahir dari seorang umat
Tanggal Baptis	Date	-	Tanggal baptis dari seorang umat
Tanggal Krisma	Date	-	Tanggal krisma dari seorang umat
Lama Tinggal	integer	-	Lama tinggal dari seorang umat
Last Update	Date	-	Tanggal terakhir diubah dari seorang umat
Sudah Dihapus	boolean	-	Apakah seorang umat sudah dihapus atau belum

3.2 Physical Data Model



Gambar xxxiv: Physical Data Model SIManja

4. Deskripsi Rancangan Antarmuka

4.1 Login



The screenshot shows a login window with the following elements:

- Window Title: LoginSIManja_070705232
- Greeting: Selamat Datang Keuskupan Agung Semarang Sistem Manajemen Gereja
- Input Fields: Nama Pengguna : and Sandi :
- Buttons: Login and Exit

Gambar xxxv: Desain Perancangan Form Login SIManja Antarmuka ini digunakan untuk melakukan proses login ke dalam sistem. Untuk mendapat akses masuk ke dalam sistem, pengguna harus menginputkan nama pengguna dan sandi dengan benar pada textbox yang telah disediakan. Pada saat tombol login ditekan, sistem akan mengecek nama pengguna dan sandi yang diinputkan dengan data nama pengguna dan sandi yang telah tersimpan di basis data. Jika data nama pengguna dan sandi benar atau cocok maka user akan masuk ke dalam sistem, sebaliknya jika nama pengguna dan sandi salah atau tidak cocok maka akan diberikan pesan peringatan. Sistem juga akan mengecek jabatan dari pengguna yang login.

Apabila tombol login ditekan maka proses akan dilanjutkan ke antarmuka main form. Bila tombol exit ditekan maka akan keluar dari sistem ini.

Bila tombol "Login" diklik:

```
"select nama_pengguna, sandi from pengguna where  
nama_pengguna = 'nama_pengguna.text' and sandi  
='sandi.text'"
```

Program Studi Teknik Informatika	DPPL-SIManja	71/ 90
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		

4.2 Main Form

Sistem Informasi Logistik	
File	Data
	About
	<ul style="list-style-type: none"> Pengelolaan Paroki Pengelolaan Wilayah Pengelolaan Stasi Pengelolaan Lingkungan Pengelolaan Kepala Keluarga Pengelolaan Umat Pengelolaan Pengguna

Gambar xxxvi: Desain Perancangan Form MainForm SIManja

Antarmuka ini merupakan inti dari sistem yang berisi menu untuk masuk ke antarmuka-antarmuka lainnya. Pada menu file berisi simpan ke file .csv, buka dari file .csv, kirim file .csv ke pusat, Log Out dan keluar dari sistem. Sedangkan menu data berisi sub menu pengelolaan paroki, pengelolaan wilayah, pengelolaan wilayah, pengelolaan stasi, pengelolaan lingkungan, pengelolaan kepala keluarga, pengelolaan umat, dan pengelolaan pengguna. Dan pada menu about berisi sub menu owner untuk mengetahui siapa yang membuat system ini.

Bila kita memilih sub menu log out maka akan kembali ke antarmuka login, sedangkan sub menu keluar akan keluar dari system.

Bila user masuk ke pengelolaan paroki maka akan masuk ke antarmuka pengelolaan paroki, bila user masuk ke pengelolaan wilayah maka akan masuk ke antarmuka pengelolaan wilayah, bila user masuk ke pengelolaan stasi maka akan masuk ke antarmuka pengelolaan stasi, bila user masuk ke pengelolaan lingkungan maka akan masuk ke antarmuka pengelolaan

lingkungan, bila user masuk ke pengelolaan kepala keluarga maka akan masuk ke antar muka pengelolaan kepala keluarga, bila user masuk ke pengelolaan umat maka akan masuk ke antar muka pengelolaan umat, dan bila user masuk ke pengelolaan pengguna maka akan masuk ke antar muka pengelolaan pengguna. Yang bisa masuk ke semua sub menu tersebut adalah user dengan jabatan administrator sedangkan dengan jabatan entry, dia hanya bisa masuk ke sub menu pengelolaan kepala keluarga dan pengelolaan umat.



Program Studi Teknik Informatika	DPPL-SIManja	73/ 90
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		

4.3 Pengelolaan Paroki

Data Paroki		Tampil Paroki		
ID Paroki	<input type="text"/>			
Nama Keuskupan	<input type="text" value="▼"/>			
Kode Paroki	<input type="text"/>			
Nama Paroki	<input type="text"/>			
Kota	<input type="text" value="▼"/>			

Tambah Ubah Hapus Clear

Cari Berdasarkan Nama Kota

Gambar xxxvii: Desain Perancangan Form Pengelolaan Paroki SIManja

Antarmuka ini digunakan untuk melakukan pengelolaan paroki oleh pengguna. Antarmuka ini berisi tempat-tempat untuk mengisi data-data mengenai detail paroki seperti nama keuskupan, nama paroki, dan kota paroki. Dan kolom search untuk mencari data paroki berdasarkan nama kota. Sedangkan pada tampilan sebelah kanan merupakan tabel yang di gunakan untuk menampilkan data-data paroki yang ada dalam basis data.

Antarmuka ini juga ada 4 buah tombol untuk melakukan pengelolaan yaitu tambah untuk menambah paroki dalam basis data, ubah untuk mengubah data paroki dalam basis data, hapus untuk menghapus data paroki yang ada dalam basis data, clean untuk menghapus isi dari textbox dan combobox, dan search untuk mencari data-data yang ada dalam basis data berdasarkan nama kota.

Bila tombol "Tambah" diklik:

```
"insert into paroki (id_paroki, id_keuskupan,
kode_paroki, nama_paroki, id_kota, sudah_dihapus)
```

```
values idParoki.text, namaKeuskupan.value,  
kodeParoki.text, namaParoki.text, kota.value, False"
```

Bila tombol "Ubah" diklik:

```
"update paroki set nama_paroki = namaParoki.text,  
id_kota = kota.value where id_paroki =  
idParoki.text"
```

Bila tombol "Hapus" diklik:

```
"update paroki set sudah_dihapus = true where  
id_paroki = idParoki.text"
```

Bila field "Cari" diketik:

```
"select id_paroki, nama_keuskupan, kode_paroki,  
nama_paroki, kota from paroki join keuskupan on join  
ref_kota where sudah_dihapus = False and kt.kota  
like 'search.text'"
```

Program Studi Teknik Informatika	DPPL-SIManja	75/90
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		

4.4 Pengelolaan Wilayah

Gambar xxxviii: Desain Perancangan Form Pengelolaan Wilayah SIManja

Antarmuka ini digunakan untuk melakukan pengelolaan wilayah oleh pengguna. Antarmuka ini berisi tempat-tempat untuk mengisi data-data mengenai detail wilayah seperti nama paroki dan nama wilayah. Dan kolom search untuk mencari data wilayah berdasarkan nama wilayah. Sedangkan pada tampilan sebelah kanan merupakan tabel yang di gunakan untuk menampilkan data-data wilayah yang ada dalam basis data.

Antarmuka ini juga ada 4 buah tombol untuk melakukan pengelolaan yaitu tambah untuk menambah wilayah dalam basis data, ubah untuk mengubah data wilayah dalam basis data, hapus untuk menghapus data wilayah yang ada dalam basis data, clean untuk menghapus isi dari textbox dan combobox, dan search untuk mencari data-data yang ada dalam basis data berdasarkan nama wilayah.

Bila tombol "Tambah" diklik:

```

insert into wilayah (id_wilayah, id_paroki,
kode_wilayah, nama_wilayah, sudah_dihapus) values
idWilayah.text, kodeWilayah.text, namaParoki.value,
namaWilayah.text, False"

```

Program Studi Teknik Informatika	DPPL-SIManja	76/ 90
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		

Bila tombol "Ubah" diklik:

```
"update wilayah set nama_wilayah = namaWilayah.text  
where id_wilayah = idWilayah.text"
```

Bila tombol "Hapus" diklik:

```
"update wilayah set sudah_dihapus = true where  
id_wilayah = idWilayah.text"
```

Bila field "Cari" diketik:

```
"select id_wilayah, nama_keuskupan, nama_paroki,  
kode_wilayah, nama_wilayah from wilayah join paroki  
join keuskupan where sudah_dihapus = 'False' and  
nama_wilayah like 'search.text'"
```



Program Studi Teknik Informatika	DPPL-SIManja	77/ 90
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		

4.5 Pengelolaan Stasi

Gambar xxxix: Desain Perancangan Form Pengelolaan Stasi SIManja

Antarmuka ini digunakan untuk melakukan pengelolaan stasi oleh pengguna. Antarmuka ini berisi tempat-tempat untuk mengisi data-data mengenai detail stasi seperti nama paroki dan nama stasi. Dan kolom search untuk mencari data stasi berdasarkan nama stasi. Sedangkan pada tampilan sebelah kanan merupakan tabel yang di gunakan untuk menampilkan data-data stasi yang ada dalam basis data.

Antarmuka ini juga ada 4 buah tombol untuk melakukan pengelolaan yaitu tambah untuk menambah stasi dalam basis data, ubah untuk mengubah data stasi dalam basis data, hapus untuk menghapus data stasi yang ada dalam basis data, clean untuk menghapus isi dari textbox dan combobox, dan search untuk mencari data-data yang ada dalam basis data berdasarkan nama stasi.

Bila tombol "Tambah" diklik:

```
"insert into stasi (id_stasi, id_paroki, kode_stasi,
nama_stasi, sudah_dihapus) values idStasi.text,
kodeStasi.text, namaParoki.value, namaStasi.text,
False"
```

Bila tombol "Ubah" diklik:

```
"update stasi set nama_stasi = namaStasi.text where  
id_stasi = idStasi.text"
```

Bila tombol "Hapus" diklik:

```
"update stasi set sudah_dihapus = true where  
id_stasi = idStasi.text"
```

Bila field "Cari" diketik:

```
"select id_stasi, nama_keuskupan, nama_paroki,  
kode_stasi, nama_stasi from stasi join paroki join  
keuskupan where sudah_dihapus = 'False' and  
nama_stasi like 'search.text'"
```



Program Studi Teknik Informatika	DPPL-SIManja	79/ 90
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		

4.6 Pengelolaan Lingkungan

Gambar xl: Desain Perancangan Form Pengelolaan Lingkungan
SIManja

Antarmuka ini digunakan untuk melakukan pengelolaan lingkungan oleh pengguna. Antarmuka ini berisi tempat-tempat untuk mengisi data-data mengenai detail lingkungan seperti nama wilayah atau nama stasi dan nama lingkungan. Dan kolom search untuk mencari data lingkungan berdasarkan nama lingkungan. Sedangkan pada tampilan sebelah kanan merupakan tabel yang di gunakan untuk menampilkan data-data lingkungan yang ada dalam basis data.

Antarmuka ini juga ada 4 buah tombol untuk melakukan pengelolaan yaitu tambah untuk menambah lingkungan dalam basis data, ubah untuk mengubah data lingkungan dalam basis data, hapus untuk menghapus data lingkungan yang ada dalam basis data, clean untuk menghapus isi dari textbox dan combobox, dan search untuk mencari data-data yang ada dalam basis data berdasarkan nama lingkungan.

Bila tombol "Tambah" diklik dan radio button wilayah terpilih:

```
"insert into lingkungan (id_lingkungan, id_wilayah,
kode_lingkungan, nama_lingkungan, sudah_dihapus)
```

```
values idLingkungan.text, kodeLingkungan.text, nama
Wilayah.value, namaLingkungan.text, False"
```

Bila tombol "Tambah" diklik dan radio button stasi terpilih:

```
"insert into lingkungan (id_lingkungan, id_stasi,
kode_lingkungan, nama_lingkungan, sudah_dihapus)
values idLingkungan.text, kodeLingkungan.text, nama
Stasi.value, namaLingkungan.text, False"
```

Bila tombol "Ubah" diklik:

```
"update lingkungan set nama_lingkungan =
namaLingkungan.text where id_stasi = idStasi.text"
```

Bila tombol "Hapus" diklik:

```
"update lingkungan set sudah_dihapus = true where
id_lingkungan = idLingkungan.text"
```

Bila field "Cari" diketik:

```
"select id_lingkungan, nama_wilayah, nama_stasi,
kode_lingkungan, nama_lingkungan from lingkungan
join wilayah join stasi where sudah_dihapus =
'False' and nama_lingkungan like 'search.text'"
```

Program Studi Teknik Informatika	DPPL-SIManja	81/ 90
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		

4.7 Pengelolaan Kepala Keluarga

Gambar xli: Desain Perancangan Form Pengelolaan Kepala Keluarga SIManja

Antarmuka ini berisi 2 buah tab. Tab pertama digunakan untuk melakukan pengelolaan kepala keluarga oleh pengguna. Antarmuka ini berisi tempat-tempat untuk mengisi data-data mengenai detail kepala keluarga seperti nama lingkungan, nama kepala keluarga, alamat kepala keluarga, kota, telepon, tempat menikah, tanggal menikah, keterangan pindah dari, keterangan pindah ke, status ekonomi, dan jenis kepala keluarga. Sedangkan pada tab kedua (sebelah kanan) merupakan tabel yang di gunakan untuk menampilkan data-data kepala keluarga yang ada dalam basis data dan juga terdapat textbox cari berdasarkan nama kepala keluarga, nama lingkungan, nama kota.

Antarmuka ini juga ada 6 buah tombol untuk melakukan pengelolaan yaitu tambah anggota untuk menambah kepala keluarga dalam basis data yang selanjutnya akan masuk ke form pengelolaan umat, tambah kk untuk menambah kepala keluarga dalam basis data, ubah anggota untuk mengubah data anggota dengan kepala keluarga dalam basis data, ubah kepala keluarga untuk mengubah data kepala keluarga dalam basis data, hapus untuk menghapus data kepala keluarga yang ada dalam basis data, clean untuk menghapus isi dari textbox dan

combobox, dan search untuk mencari data-data yang ada dalam basis data berdasarkan nama kepala keluarga, nama lingkungan, dan nama kota.

Bila tombol "Tambah" diklik:

```
"insert into kepala_keluarga (id_kepala_keluarga,
id_lingkungan, kode_kepala_keluarga, nama_kepala_keluarga,
alamat, id_kota_tinggal, telepon, id_kota_nikah, tanggal_nikah,
id_kota_pindahKe, id_kota_pindahDari, id_status_ekonomi, id_jenis_kepala_keluarga,
sudah_dihapus) values id_kepala_keluarga.text,
id_lingkungan.value, kode_kepala_keluarga.text,
nama_kepala_keluarga.text, alamat.text,
id_kota_tinggal.value, telepon.text, id_kota_nikah.value,
tanggal_nikah.value, id_kota_pindahKe.value,
id_kota_pindahDari.value, id_status_ekonomi.value,
id_jenis_kepala_keluarga.value, False"
```

Bila tombol "Ubah" diklik:

```
"update kepala_keluarga set nama_kepala_keluarga=
nama_kepala_keluarga.text, alamat=alamat.text,
id_kota_tinggal= id_kota_tinggal.value, telepon=
telepon.text, id_kota_nikah=id_kota_nikah.value,
tanggal_nikah=tanggal_nikah.value, id_kota_pindahKe=
id_kota_pindahKe.value, id_kota_pindahDari=id_kota_pindahDari.value,
id_status_ekonomi=id_status_ekonomi.value, id_jenis_kepala_keluarga=
id_jenis_kepala_keluarga.value where id_kepala_keluarga =
id_kepala_keluarga.text"
```

Bila tombol "Hapus" diklik:

```
"update kepala_keluarga set sudah_dihapus = true
where id_kepala_keluarga = id_kepala_keluarga.text"
```

Bila field "Cari" diketik:

```
"select id_kepala_keluarga, nama_lingkungan,
kode_kepala_keluarga, nama_kepala_keluarga,
alamat_kepala_keluarga, kota, telepon, kota_nikah,
```

Program Studi Teknik Informatika	DPPL-SIManja	83/ 90
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		

```
tanggal_menikah, kota_pindahDari, kota_pindahKe,
status_ekonomi, jenis_kepala_keluarga , last_update
from kepala_keluarga join lingkungan join
ref_status_ekonomi join ref_jenis_kepala_keluarga
join ref_kota join ref_kota join ref_kota join
ref_kota where sudah_dihapus = 'False' and
nama_kepala_keluarga like 'search.text'"
```



Program Studi Teknik Informatika	DPPL-SIManja	84/ 90
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		

4.8 Pengelolaan Umat

⊗

Olah Umat
Tampil Umat

Data Umat

ID Umat	<input type="text"/>	Suku	<input type="text"/>
Nama Kepala Keluarga	<input type="text"/>	Pekerjaan	<input type="text"/>
Kode Umat	<input type="text"/>	Pendidikan Terakhir	<input type="text"/>
Nama Umat	<input type="text"/>	Bidang Studi	<input type="text"/>
Jenis Kelamin	<input type="text"/>	Status Pendidikan	<input type="text"/>
Tempat Lahir	<input type="text"/>	Hubungan Kepala Keluarga	<input type="text"/>
Tanggal Lahir	dd MMMM yyyy	Status Keluarga	<input type="text"/>
Agama	<input type="text"/>	Status Perkawinan	<input type="text"/>
Status Tempat Tinggal	<input type="text"/>	Status Sosial	<input type="text"/>
Lama Tinggal	<input type="text"/>	Status Kesehatan	<input type="text"/>
Status Baptis	<input type="text"/>	Status Katolik	<input type="text"/>
Tempat Baptis	<input type="text"/>	Status Gereja	<input type="text"/>
Tanggal Baptis	dd MMMM yyyy	Jabatan di Paroki	<input type="text"/>
Tempat Krisma	<input type="text"/>		
Tanggal Krisma	dd MMMM yyyy		

Gambar xlii: Desain Perancangan Form Pengelolaan Umat SIManja

Antarmuka ini berisi 2 buah tab. Tab pertama digunakan untuk melakukan pengelolaan umat oleh pengguna. Antarmuka ini berisi tempat-tempat untuk mengisi data-data mengenai detail umat seperti nama kepala keluarga, nama umat, jenis kelamin, tempat lahir, tanggal lahir, agama, status tempat tinggal, lama tinggal, golongan darah, status baptis, tempat baptis, tanggal baptis, tempat krisma, tanggal krisma, suku, pekerjaan, pendidikan terakhir, bidang studi, status pendidikan, hubungan kepala keluarga, status keluarga, status perkawinan, status sosial, status kesehatan, status katolik, status gereja, jabatan di paroki. Sedangkan pada tab kedua (sebelah kanan) merupakan tabel yang di gunakan untuk menampilkan data-data umat yang ada dalam basis data dan juga terdapat textbox cari berdasarkan nama umat.

Antarmuka ini juga ada 4 buah tombol untuk melakukan pengelolaan yaitu tambah untuk menambah umat dalam basis

data, ubah untuk mengubah data umat dalam basis data, hapus untuk menghapus data umat yang ada dalam basis data, clean untuk menghapus isi dari textbox dan combobox, dan search untuk mencari data-data yang ada dalam basis data berdasarkan nama umat.

Bila tombol "Tambah" diklik:

```
"insert into umat (id_umat, id_kepala_keluarga,
kode_umat, nama_umat, jenis_kelamin, kota_tempat_
lahir, tanggal_lahir, agama, status_tempat_tinggal,
lama_tinggal, golongan_darah, status_baptis, tempat_
baptis, tanggal_baptis, tempat_krisma, tanggal_
krisma, suku, pekerjaan, pendidikan_terakhir,
bidang_studi, status_pendidikan, hubungan_kepala_
keluarga, status_keluarga, status_perkawinan, status_
sosial, status_kesehatan, status_katolik, status_
gereja, jabatan_di_paroki, sudah_dihapus) values
id_umat.text, id_kepala_keluarga.value, kode_umat.
text, nama_umat.text, jenis_kelamin.value, kota_
tempat_lahir.value, tanggal_lahir.value, agama.
value, status_tempat_tinggal.value, lama_tinggal.
value, golongan_darah.value, status_baptis.value,
tempat_baptis.value, tanggal_baptis.value, tempat_
krisma.value, tanggal_krisma.value, suku.value,
pekerjaan.value, pendidikan_terakhir.value, bidang_
studi.value, status_pendidikan.value, hubungan_
kepala_keluarga.value, status_keluarga.value,
status_perkawinan.value, status_sosial.value,
status_kesehatan.value, status_katolik.value,
status_gereja.value, jabatan_di_paroki.value,
False"
```

Bila tombol "Ubah" diklik:

```
"update umat set nama_umat=nama_umat.text,
jenis_kelamin=jenis_kelamin.value, kota_tempat_lahir
=kota_tempat_lahir.value, tanggal_lahir=tanggal_
```

Program Studi Teknik Informatika	DPPL-SIManja	86/ 90
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		

```

lahir.value,      agama=agama.value,      status_tempat_
tinggal=status_tempat_tinggal.value,      lama_tinggal=
lama_tinggal.value,      golongan_darah=golongan_darah
.value,      status_baptis=status_baptis.value,      tempat_
baptis=tempat_baptis.value,      tanggal_baptis=tanggal_
baptis.value,      tempat_krisma=tempat_krisma.value,
tanggal_krisma=tanggal_krisma.value,      suku=suku.
value,      pekerjaan=pekerjaan.value,      pendidikan_
terakhir=pendidikan_terakhir.value,      bidang_studi=
bidang_studi.value,      status_pendidikan=status_
pendidikan.value,      hubungan_kepala_keluarga=hubungan_
kepala_keluarga.value,      status_keluarga=status_
keluarga.value,      status_perkawinan=status_perkawinan.
value,      status_sosial=status_sosial.value,      status_
kesehatan=status_kesehatan.value,      status_katolik=
status_katolik.value,      status_gereja=status_gereja.
value,      jabatan_di_paroki=jabatan_di_paroki.value
where id_umat = id_umat.text"

```

Bila tombol "Hapus" diklik:

```

"update umat set sudah_dihapus = true where id_umat
= id_umat.text"

```

Bila field "Cari" diketik:

```

"select      id_umat,      nama_lingkungan,
nama_kepala_keluarga,      kode_umat,      nama_umat,
jenis_kelamin,      golongan_darah,      agama,      tempat_
tinggal,      status_baptis,      kota,      tanggal_lahir,      kota,
tanggal_baptis,      liberbap,      kota,      tanggal_krisma,
suku,      pekerjaan,      pendidikan_terakhir,      bidang_studi,
sta_pend.status_pendidikan,      hubungan_kepala_
keluarga,      status_keluarga,      status_perkawinan,
status_sosial,      status_kesehatan,      status_katolik,
status_gereja,      jabatan_paroki,      lama_tinggal from
umat join kepala_keluarga join lingkungan ref_kota
join ref_kota join ref_kota join ref_jenis_kelamin

```

Program Studi Teknik Informatika	DPPL-SIManja	87/ 90
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		

```
join ref_golongan_darah join ref_agama join
ref_tempat_tinggal join ref_status_baptis join
ref_suku join ref_pekerjaan join ref_pendidikan_
terakhir join ref_bidang_studi join ref_status_
pendidikan join ref_hubungan_kepala_keluarga join
ref_status_keluarga join ref_status_perkawinan join
ref_status_sosial join ref_status_kesehatan
ref_status_katolik join ref_status_gereja join
ref_jabatan_paroki where u.sudah_dihapus = 'False'
and nama_umat like 'search.text'"
```



Program Studi Teknik Informatika	DPPL-SIManja	88/ 90
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		

4.9 Pengelolaan Pengguna

Gambar xliii: Desain Perancangan Form Pengelolaan Pengguna SIManja

Antarmuka ini digunakan untuk melakukan pengelolaan pengguna. Antarmuka ini berisi tempat-tempat untuk mengisi data-data mengenai detail user seperti nomor pengguna, nama pengguna, sandi, dan jabatan di sistem. Dan kolom search untuk mencari data user berdasarkan nama pengguna. Sedangkan pada tampilan sebelah kanan merupakan tabel yang di gunakan untuk menampilkan data-data user yang ada dalam database.

Antarmuka ini juga ada 3 buah tombol untuk melakukan pengelolaan yaitu tambah untuk menambah user dalam database, ubah untuk mengubah data user dalam database, dan clean untuk menghapus isi dari textbox dan combobox.

Bila tombol "Tambah" diklik:

```
"insert into pengguna (id_pengguna, nama_pengguna, sandi, jabatan) values id_pengguna.text, nama_pengguna.text, sandi.text, jabatan.value"
```

Bila tombol "Ubah" diklik:

```
"update pengguna set nama_pengguna=nama_pengguna.text, sandi=sandi.text, jabatan=jabatan.text where id_pengguna = id_pengguna.text"
```

Program Studi Teknik Informatika	DPPL-SIManja	89/ 90
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		

Bila field "Cari" diketik:

```
"select id_stasi, nama_keuskupan, nama_paroki,  
kode_stasi, nama_stasi from stasi join paroki join  
keuskupan where sudah_dihapus = 'False' and  
nama_stasi like 'search.text'"
```



Program Studi Teknik Informatika	DPPL-SIManja	90/ 90
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		