

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan dari analisis, desain dan implementasi perangkat lunak SIQ, maka dapat ditarik kesimpulan dari Tugas Akhir ini sebagai berikut :

1. Sistem informasi SIQ telah berhasil dibuat untuk dapat melakukan fungsi yang diperlukan dalam pengelolaan prosedur kerja mulai dari pembuatan, *approval*, *rejecting*, *published* dan *searching*.
2. Sistem informasi SIQ membantu proses pengelolaan dan penyampaian informasi suatu dokumen prosedur kerja sehingga lebih efisien serta dalam penyampaian informasi mengenai *history* suatu dokumen QSP.

6.2 Saran

Beberapa saran dan masukan yang dapat disampaikan penulis terhadap pembuatan SIQ ini dimasa yang akan datang adalah :

1. Perangkat lunak diharapkan dapat dikembangkan lebih lanjut dengan penambahan fungsionalitas sehingga sistem menjadi lebih baik seperti misalnya fungsionalitas untuk membuat dokumen selain dokumen QSP contohnya dokumen QSR (*Quality System Record*), *Internal Document*, dll. Sistem juga diharapkan dapat dikembangkan dengan interface yang lebih menarik sehingga user nyaman

dalam menggunakan sistem dan terikat dengan fasilitas yang disediakan.

2. Masih terdapat beberapa perbedaan *business process* yang menjadi dasar dalam pembuatan sistem informasi SIQ, akan lebih baik jika *business process* di masing-masing bagian dapat disetarakan.



DAFTAR PUSTAKA

- Adityatama, Bagus. 2010. *Pengembangan Sistem Informasi Kerja Praktek/Magang Berbasis Web untuk Progrm Studi Teknik Informatika Universitas Atma Jaya Yogyakarta*. Jurusan Teknik Informatika Fakultas Teknologi Industri. Universitas Atma Jaya Yogyakarta. Yogyakarta
- Budiarto. 2008. *Pembangunan Sistem Informasi Pengelolaan Aktivitas Akademik Dosen Universitas Atma Jaya Yogyakarta Berbasis Web dengan Menggunakan Prado Framework*. Jurusan Teknik Informatika Fakultas Teknologi Industri. Universitas Atama Jaya Yogyakarta. Yogyakarta
- Dian, Christina. 2010. *Pembangunan Sistem Informasi Manajemen Dokumen Standard Operating Procedure Berbasis Web (Studi Kasus Chevron Indonesia Company)*. Jurusan Teknik Informatika Fakultas Teknologi Industri. Universitas Atma Jaya Yogyakarta
- Jalulaga, Anggisesa. *Pengembangan Aplikasi CRM (Customer Relationship Management) pada Perusahaan Asuransi dengan Object Relational Database Management System Studi Kasus di PT. Asuransi Allianz Life Indonesia*. Jurusan Teknik Informatika Fakultas Teknologi Industri. Universitas Atma Jaya Yogyakarta. Yogyakarta

- Santoso, Urip. 2008. *Penyusunan Standard Operating Procedure Layanan Universitas Bengkulu*.
<http://uripsantoso.wordpress.com/2008/04/08/penyusunan-standard-operating-procedure-layanan-universitas-bengkulu/> . diakses pada tanggal 12 Juli 2011
- Setiawan, Yeremia., Setiarso, Aris. 2002. *Usulan Perancangan Sistem Informasi Manajemen Pengendalian Mutu Pekerjaan Beton Konvensional di Perusahaan Jasa Konstruksi*. Jurusan Teknik Sipil. Universitas Kristen Petra. Surabaya
- Williamson. 2008. *Generation, Review, Approval and Use of Standard Operating* . Colorado State University. Colorado.
- Suwandhy, Felix. *Perancangan Sistem Standard Operating Procedure (SOP) di Laboratorium Mesin Computer Numerical Control (CNC) Universitas Kristen Petra, Jurusan Teknik Industri, Universitas Kristen Petra, Surabaya*
- Turban et. al., 2009. *Introduction to Information System*. 2nd Edition. John Wiley & Sons.
- United States Environment Protection Agency. 2007. *Guidance for Preparing Standard Operating Procedures(SOP)*. Office of Environmental Information. Washington DC

SKPL


SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK
PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN
QUALITY SYSTEM PROCEDURE UNIVERSITAS ATMA
JAYA YOGYAKARTA

Untuk :
Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:
Herminda Aprisari / 5460

Program Studi Teknik Informatika - Fakultas Teknologi
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		SKPL-SIQ		1/44
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

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

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

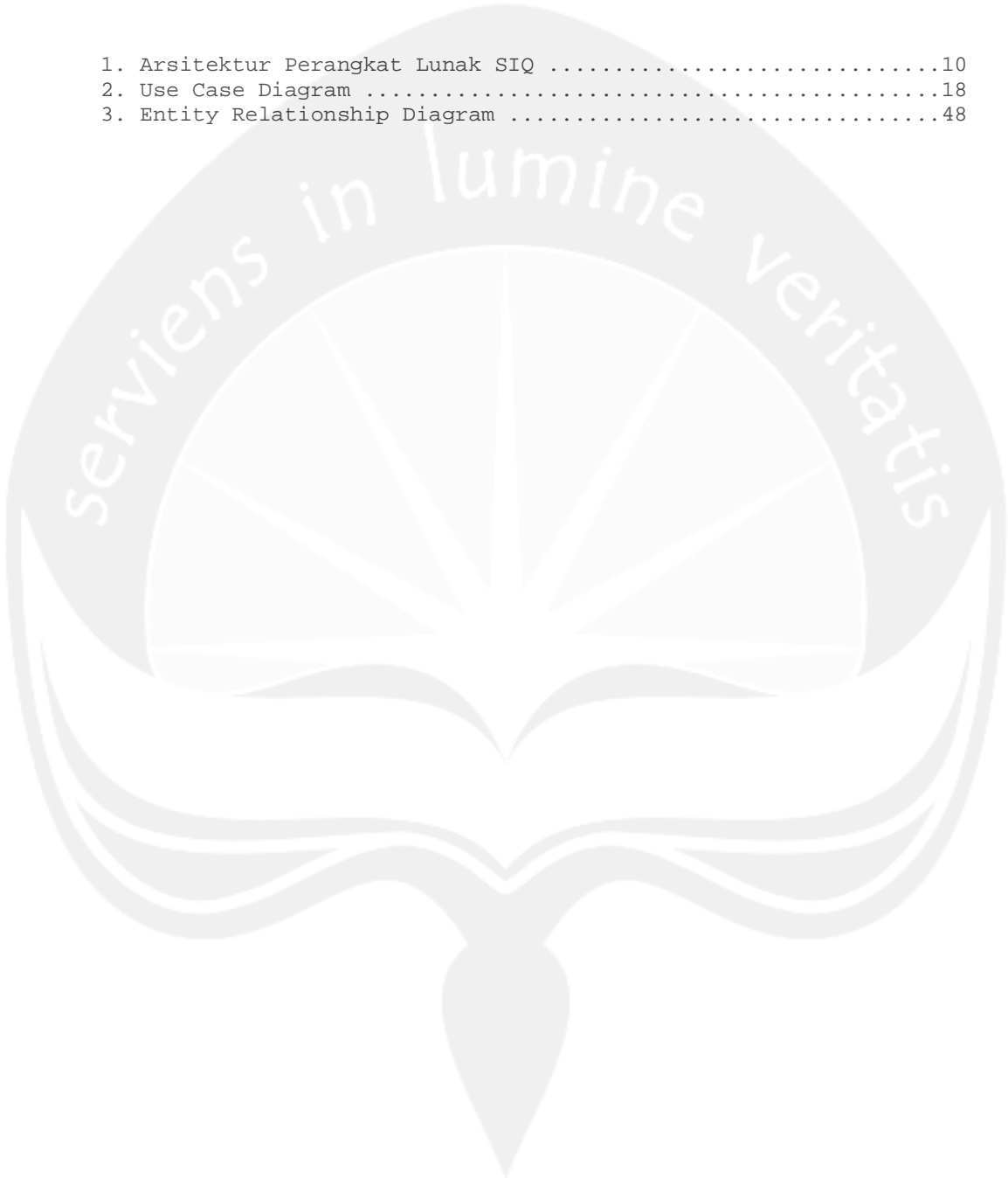


Daftar Isi

1	Pendahuluan.....	6
1.1	Tujuan.....	6
1.2	Lingkup Masalah	6
1.3	Definisi, Akronim dan Singkatan	7
1.4	Referensi.....	8
1.5	Deskripsi umum (Overview)	8
2	Deskripsi Kebutuhan.....	9
2.1	Perspektif produk	9
2.2	Fungsi Produk.....	10
2.3	Karakteristik Pengguna	15
2.4	Batasan-batasan	15
2.5	Asumsi dan Ketergantungan	15
3	Kebutuhan khusus.....	15
3.1	Kebutuhan antarmuka eksternal	15
3.2	Kebutuhan fungsionalitas Perangkat Lunak	17
4	Spesifikasi Rinci Kebutuhan	18
4.1	Spesifikasi Kebutuhan Fungsionalitas	18
5	ERD	44

Daftar Gambar

1. Arsitektur Perangkat Lunak SIQ	10
2. Use Case Diagram	18
3. Entity Relationship Diagram	48



1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan perangkat Lunak (SKPL-SIQ) dalam pengembangan perangkat lunak SIQSP (Pembangunan Sistem Informasi Pengelolaan *Quality System Procedure* Universitas Atma Jaya Yogyakarta) bertujuan mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna), serta mendefinisikan fungsi-fungsi pada perangkat lunak. Dalam SKPL-SIQ ini juga mendefinisikan batasan perancangan perangkat lunak serta karakteristik program yang akan dikembangkan.

1.2 Lingkup Masalah

Perangkat Lunak SIQ dikembangkan dengan tujuan untuk :

1. Membangun aplikasi sistem informasi dokumen manajemen yang dapat menangani segala hal yang berkaitan dengan prosedur kerja.
2. Menangani proses bisnis yang berkaitan dengan pengelolaan QSP seperti *approval*, *searching* dan *published* yang semula masih manual.
3. Menangani proses *approval* dalam assign dokumen.

Dan berjalan pada lingkungan dengan platform Windows.

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-SIQ-XXX	Kode yang merepresentasikan kebutuhan pada SIQ (Pembangunan Sistem Informasi Pengelolaan <i>Quality System Procedure</i> Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta) dimana XXX merupakan nomor fungsi produk.
SIQ	Perangkat lunak yang bertujuan membantu dalam pengelolaan <i>Quality System Procedure</i> (QSP) untuk Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta.
SOP /QSP	<i>Standard Operating Procedure / Quality System Procedure</i> yang merupakan langkah-langkah kerja yang mendeskripsikan prosedur rutin yang digunakan oleh suatu organisasi untuk mencapai suatu tujuan.
<i>Approver</i>	Pengguna sistem informasi yang memiliki role untuk melakukan persetujuan tahap pertama akan data QSP.
<i>Management Representative</i>	Pengguna sistem informasi yang memiliki role paling tinggi dalam melakukan persetujuan tahap kedua akan data QSP yang akan di- <i>published</i> .

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Dian, Christina, *Spesifikasi Kebutuhan Perangkat Lunak*, Jurusan Teknik Informatika - UAJY

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 5 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

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

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak SIQ yang akan dikembangkan, yang mencakup kebutuhan antarmuka eksternal dan kebutuhan fungsionalitas perangkat lunak beserta use case diagram dari kebutuhan fungsionalitas perangkat lunak ini.

Bagian keempat berisi penjelasan mengenai spesifikasi rinci kebutuhan, mencakup spesifikasi kebutuhan fungsionalitas yang terbagi menjadi beberapa use case specification sesuai dengan use case diagram yang ada pada bagian ketiga.

Bagian kelima berisi ERD (Entitas relationship Diagram) yang menggambarkan hubungan antar entitas dalam sistem.

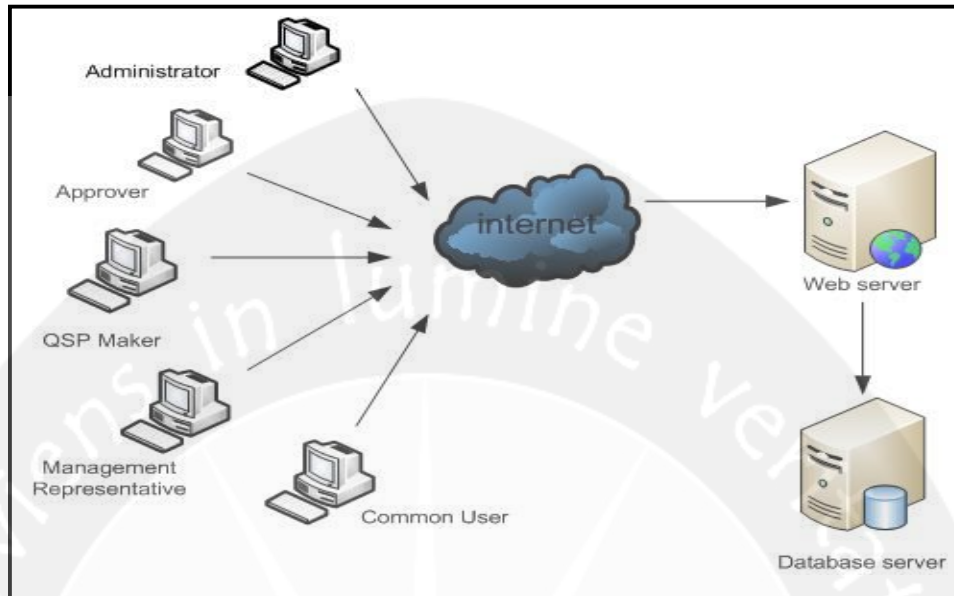
2 Deskripsi Kebutuhan

2.1 Perspektif produk

SIQ merupakan perangkat lunak yang dikembangkan untuk membantu pengelolaan informasi mengenai prosedur kerja. Pengelolaan informasi ini dilakukan sedemikian rupa sehingga user mampu membaca, membuat serta meng-assign dokumen kepada *approver* untuk kemudian disetujui oleh unit yang bersangkutan dan kemudian di-*published*.

Perangkat lunak SIQ ini berjalan pada platform Windows yang terhubung secara online ke intranet, dibuat menggunakan bahasa pemrograman Microsoft Visual C# ASP .NET, sedangkan lingkungan pemrogramannya menggunakan Microsoft Visual Studio .NET 2005.

Dalam sistem ini akan digunakan user interface untuk pengguna berinteraksi dengan sistem, interaksi terjadi antara Web Server dengan Database Server (sebagai media utama penyimpanan data) dan Web Server dengan browser yang terhubung melalui internet. Aplikasi ini akan mendapatkan data inputan dari user, data ini kemudian akan diolah dan disimpan dalam database, pengolahan data tersebut akan menghasilkan data yang dapat diakses oleh semua user dalam sistem.



Gambar 1. Arsitektur Perangkat lunak SIQ

2.2 Fungsi Produk

Fungsi produk perangkat lunak SIQSP akan dibagi berdasarkan user yang masuk kedalam sistem sebagai berikut :

1. Fungsi Login (SKPL-SIQ-001)

Adalah fungsi yang digunakan oleh user sistem (Administrator, QSP Maker, Approver, Management Representative dan Common User) untuk dapat masuk dalam sistem yang akan digunakan.

2. Fungsi Cari QSP (SKPL-SIQ-002)

Adalah fungsi yang digunakan oleh user sistem (Administrator, QSP Maker, Approver, Management Representative dan Common User) untuk melakukan pencarian dokumen QSP.

3. Fungsi *Kelola Unit Kerja* (**SKPL-SIQ-003**)

Merupakan fungsi yang digunakan oleh administrator untuk mengelola unit kerja, fungsi ini meliputi:

a. *Tambah Data Unit Kerja* (**SKPL-SIQ-003-01**)

Merupakan fungsi yang digunakan untuk membuat unit kerja baru.

b. *Ubah Data Unit Kerja* (**SKPL-SIQ-003-02**)

Merupakan fungsi yang digunakan untuk mengubah data unit kerja yang sudah ada.

c. *Hapus Data Unit Kerja* (**SKPL-SIQ-003-03**)

Merupakan fungsi yang digunakan untuk menghapus data unit kerja yang sudah ada.

4. Fungsi *Kelola Type QSP* (**SKPL-SIQ-004**)

Merupakan fungsi yang digunakan oleh administrator untuk mengelola type QSP, fungsi ini meliputi:

a. *Tambah Data Type QSP* (**SKPL-SIQ-004-01**)

Merupakan fungsi yang digunakan untuk membuat type QSP baru.

b. *Ubah Data Type QSP* (**SKPL-SIQ-004-02**)

Merupakan fungsi yang digunakan untuk mengubah data type QSP yang sudah ada.

c. *Hapus Data Type QSP* (**SKPL-SIQ-004-03**)

Merupakan fungsi yang digunakan untuk menghapus data type QSP yang sudah ada.

5. Fungsi *Kelola Status QSP* (**SKPL-SIQ-005**)

Merupakan fungsi yang digunakan oleh administrator untuk mengelola status QSP, fungsi ini meliputi:

a. *Tambah Data Status QSP (SKPL-SIQ-005-01)*

Merupakan fungsi yang digunakan untuk membuat status QSP baru.

b. *Ubah Data Status QSP (SKPL-SIQ-005-02)*

Merupakan fungsi yang digunakan untuk mengubah data status QSP yang sudah ada.

c. *Hapus Data Status QSP (SKPL-SIQ-005-03)*

Merupakan fungsi yang digunakan untuk menghapus data status QSP yang sudah ada.

6. *Fungsi Kelola Jabatan User (SKPL-SIQ-006)*

Merupakan fungsi yang digunakan oleh administrator untuk mengelola data jabatan user, fungsi ini meliputi:

a. *Tambah Data Jabatan User (SKPL-SIQ-006-01)*

Merupakan fungsi yang digunakan untuk membuat status QSP baru.

b. *Ubah Data Jabatan User (SKPL-SIQ-006-02)*

Merupakan fungsi yang digunakan untuk mengubah data status QSP yang sudah ada.

c. *Hapus Data Jabatan User (SKPL-SIQ-006-03)*

Merupakan fungsi yang digunakan untuk menghapus data status QSP yang sudah ada.

7. *Fungsi Kelola Akun User (SKPL-SIQ-007)*

Adalah fungsi yang digunakan oleh administrator untuk mengelola user. Fungsi kelola user meliputi:

a. *Tambah Data Akun User (SKPL-SIQ-007-01)*

Merupakan fungsi yang digunakan untuk membuat user baru.

b. *Ubah Data Akun User (SKPL-SIQ-007-02)*

Merupakan fungsi yang digunakan untuk mengubah data user yang sudah ada.

c. *Hapus Data Akun User (SKPL-SIQ-007-03)*

Merupakan fungsi yang digunakan untuk menghapus data user.

8. *Fungsi Buat QSP (SKPL-SIQ-008)*

Merupakan fungsi yang digunakan oleh Approver, Management Representative dan QSP Maker untuk membuat dokumen atau data QSP yang baru.

9. *Fungsi Kelola Draft QSP (SKPL-SIQ-009)*

Merupakan fungsi yang digunakan oleh Administrator, Approver, dan QSP Maker untuk mengelola draft QSP. Fungsi kelola draft QSP meliputi:

a. *Ubah Draft QSP (SKPL-SIQ-009-01)*

Merupakan fungsi yang digunakan untuk mengubah data draft QSP.

10. *Fungsi Kelola List Approval (SKPL-SIQ-010)*

Merupakan fungsi yang digunakan oleh Approver dan Management Representative untuk mengelola daftar approval QSP. Fungsi ini meliputi:

a. *Approved QSP (SKPL-SIQ-010-01)*

Merupakan fungsi yang digunakan untuk melakukan persetujuan QSP yang kemudian akan dipublish.

b. *Returned QSP (SKPL-SIQ-010-02)*

Merupakan fungsi yang digunakan untuk mengembalikan data QSP kepada pembuat QSP untuk diperbaiki.

c. *Tampil Data QSP (SKPL-SIQ-010-03)*

Merupakan fungsi yang digunakan untuk melihat data QSP.

11. *Fungsi Kelola Published QSP (SKPL-SIQ-011)*

Merupakan fungsi yang digunakan oleh Administrator, Approver, QSP Maker dan Management Representative untuk mengelola QSP yang telah disetujui dan dipublish. Fungsi kelola published QSP meliputi:

a. *Tampil Published QSP (SKPL-SIQ-011-01)*

Merupakan fungsi yang digunakan untuk menampilkan data QSP yang telah dipublish.

b. *Update Status Published QSP (SKPL-SIQ-011-02)*

Merupakan fungsi yang digunakan untuk mengubah status QSP yang sudah dipublish yang sudah tidak digunakan lagi.

c. *Revisi Published QSP (SKPL-SIQ-011-03)*

Merupakan fungsi yang digunakan untuk merevisi QSP yang sudah dipublish.

12. *Fungsi Kelola Returned QSP (SKPL-SIQ-012)*

Merupakan fungsi yang digunakan oleh Approver, QSP Maker dan Management Representative untuk mengelola QSP yang dikembalikan kepada pembuat QSP untuk diperbaiki, fungsi ini meliputi:

a. *Ubah Returned QSP (SKPL-SIQ-012-01)*

Merupakan fungsi yang digunakan untuk mengubah data QSP yang dikembalikan oleh approver.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SIQ ini adalah karyawan Universitas Atma Jaya yang diberi akses untuk masuk ke dalam aplikasi ini dan mengerti mengenai pengelolaan instruksi kerja.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SIQ tersebut adalah:

1. Kebijakan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak SIQ.

2. Keterbatasan perangkat keras

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

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada komputer yang memiliki sistem operasi Windows.

3 Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

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

3.1.1 Antarmuka pemakai

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

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SIQ adalah:

1. Komputer Desktop atau PC

3.1.3 Antarmuka perangkat lunak

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

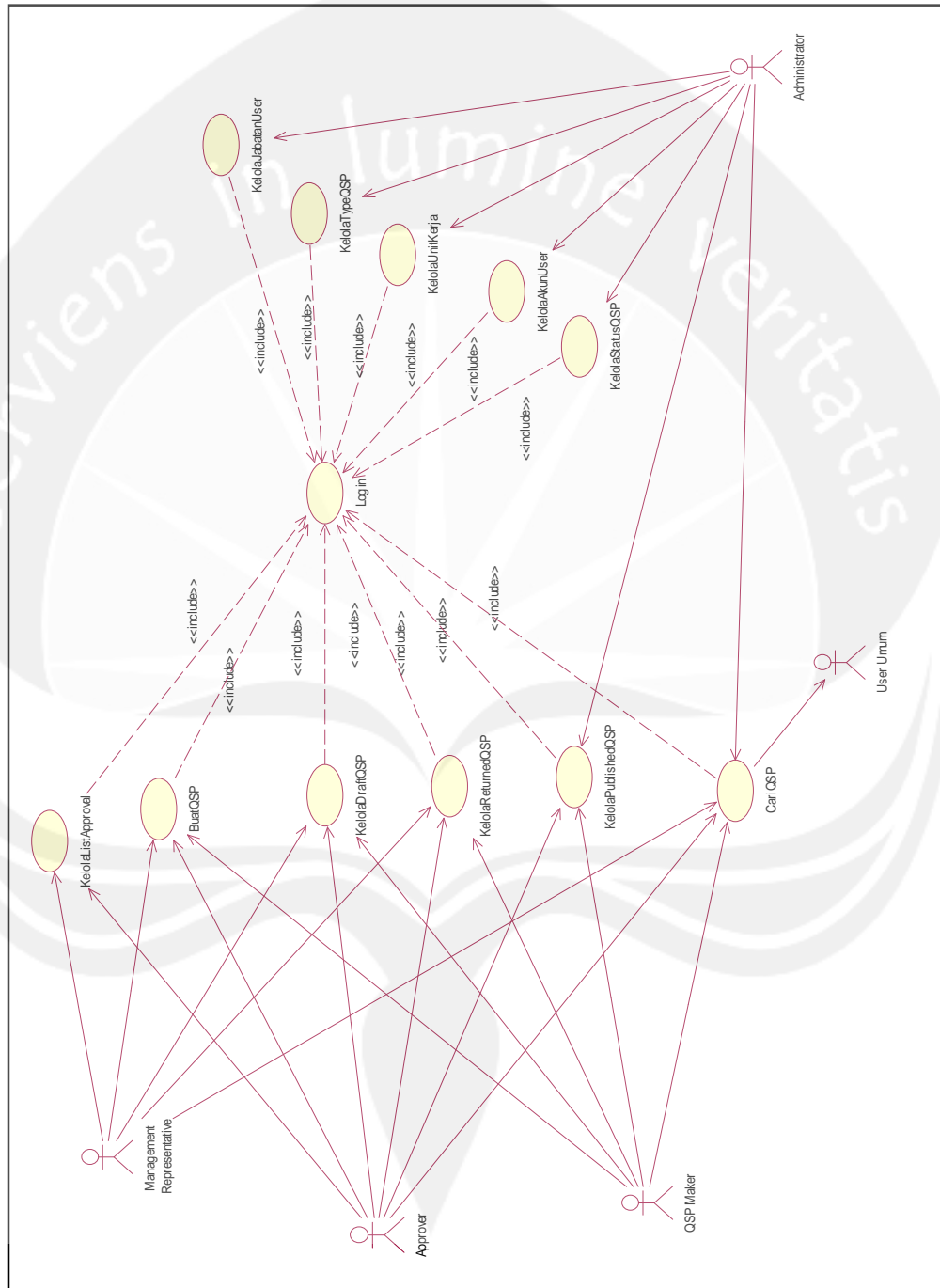
1. Nama : SQL Server 2005
Sumber : Microsoft
Sebagai database management system (DBMS) yang digunakan untuk penyimpan data di sisi server.
2. Nama : Microsoft .NET Framework 2005
Sumber : Microsoft
Sebagai user interface.
3. Nama : Windows 7
Sumber : Microsoft
Sebagai sistem operasi untuk computer desktop
4. Nama : IIS (Internet Information Server)
Sumber : Microsoft
Sebagai web server.
5. Nama : Mozilla Firefox
Sumber : Open Source
Sebagai browser application untuk mengakses halaman web.

3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak SIQ menggunakan protocol TCP/IP.

3.2 Kebutuhan fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram



Gambar 2. Use Case Diagram

4 Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

4.1.1 Use case Spesification : Login

1. Brief Description

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

2. Primary Actor

1. Administrator
2. Approver
3. QSP Maker
4. Management Representative
5. User Umum

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login
2. Sistem menampilkan antarmuka untuk login
3. Aktor memasukkan id dan password
4. Sistem memeriksa id dan password yang diinputkan aktor
E-1 Password atau id user tidak sesuai
5. Sistem memberikan akses ke aktor
6. Use Case ini selesai

5. Alternative Flow

None

6. Error Flow

- E-1 Password atau nama user tidak sesuai
1. Sistem menampilkan peringatan bahwa id user atau password tidak sesuai

2. Kembali ke Basic Flow langkah ke 3

7. PreConditions

None

8. PostConditions

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

4.1.2 Use case Spesification : Cari QSP

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pencarian data QSP dalam sistem dengan memasukkan *keyword* sesuai dengan ketentuan.

2. Primary Actor

1. Administrator
2. Approver
3. QSP Maker
4. Management Representative
5. User Umum

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pencarian data QSP
2. Aktor menginputkan keyword dari data QSP yang akan dicari misalnya nomor QSP
3. Sistem akan mencari data QSP dalam database.
E-1 Sistem tidak menemukan data QSP yang dicari
4. Sistem menampilkan data QSP yang dicari oleh aktor
5. Use Case selesai

5. Alternative Flow

None

6. Error Flow

E-1 Sistem tidak menemukan data QSP yang dicari

1. Sistem akan member peringatan bahwa data tidak ditemukan
2. Kembali ke basic flow langkah 2

7. PreConditions

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

8. PostConditions

1. Data yang dibutuhkan aktor telah ditampilkan

4.1.3 Use case Spesification : Kelola Unit Kerja

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola unit kerja. Aktor dapat melakukan tambah data, ubah data, hapus data atau tampil data.

2. Primary Actor

1. Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan unit kerja data unit kerja
2. Sistem memberikan pilihan untuk melakukan tambah data unit kerja, ubah data unit kerja, hapus data unit kerja atau tampil data unit kerja
3. Aktor memilih untuk melakukan tambah data unit kerja
 - A-1 Aktor memilih untuk melakukan ubah data unit kerja
 - A-2 Aktor memilih untuk melakukan hapus data unit kerja
 - A-3 Aktor memilih untuk melakukan tampil data unit kerja
 - A-4 Aktor memilih untuk melakukan cari data unit kerja

Program Studi Teknik Informatika	SKPL – SIQ	20/ 44
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

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

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan ubah data unit kerja
1. Sistem meminta data unit kerja yang akan diubah
 2. Aktor menginput data unit kerja yang akan diubah berdasarkan keyword yang sudah ditentukan, misalnya ID
 3. Sistem mencari data unit kerja dalam database
 - E-2 Data unit kerja tidak ditemukan
 4. Sistem menampilkan data unit kerja yang akan diubah
 5. Aktor mengubah data unit kerja yang telah ditemukan
 6. Aktor menyimpan data unit kerja yang sudah diubah
 7. Sistem melakukan pengecekan terhadap data unit kerja yang telah diubah
 - E-3 Data unit kerja yang telah diubah salah
 8. Sistem menyimpan data unit kerja yang telah diubah ke database
 9. Berlanjut ke Basic Flow langkah ke 8
- A-2 Aktor memilih untuk melakukan hapus data unit kerja
1. Sistem meminta data unit kerja yang akan dihapus

2. Aktor memasukkan data unit kerja yang akan diubah berdasarkan keyword yang sudah ditentukan, misalnya ID
 3. Sistem mencari data unit kerja dalam database
 - E-2 Data unit kerja tidak ditemukan
 4. Sistem menampilkan data unit kerja yang akan dihapus
 5. Aktor meminta sistem untuk menghapus data unit kerja yang telah ditampilkan
 6. Sistem meminta konfirmasi penghapusan data
 7. Aktor menyetujui penghapusan data unit kerja
 - A-5 Aktor tidak menyetujui penghapusan data unit kerja
 8. Sistem menghapus data unit kerja dari database
 9. Berlanjut ke Basic Flow langkah ke 8
- A-3 Aktor memilih untuk melakukan tampil data unit kerja
1. Sistem menampilkan data atau profile unit kerja
 2. Berlanjut ke Basic Flow langkah ke 8
- A-4 Aktor memilih untuk melakukan cari data unit kerja
1. Sistem akan menampilkan data unit kerja
 2. Aktor menginputkan data unit kerja yang diminta
 3. Sistem akan mencari data unit kerja yang diminta.
 - E-2 Data unit kerja tidak ditemukan
 4. Sistem akan menampilkan data unit kerja yang diminta
 5. Berlanjut ke basic flow langkah 8
- A-5 Aktor tidak menyetujui penghapusan data unit kerja
1. Berlanjut ke Alternative Flow A-2 langkah ke

6. Error Flow

- E-1 Data unit kerja yang diinputkan aktor salah
1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah

2. Kembali ke Basic Flow Langkah ke 4

E-2 Data unit kerja tidak ditemukan

1. Sistem memberikan pesan peringatan bahwa data yang dicari tidak ditemukan

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

E-3 Data unit kerja yang telah diubah salah

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah

2. Kembali ke Alternative Flow A-1 langkah ke 5

7. PreConditions

1. Use Case Login telah dilakukan

2. Aktor telah memasuki sistem

8. PostConditions

1. Data lama unit kerja di database telah *ter-update* dengan data yang baru

4.1.4 Use Case Specification : Kelola Type QSP

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola type QSP. Aktor dapat melakukan tambah data, ubah data, hapus data atau tampil data.

2. Primary Actor

1. Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan type QSP data type QSP

2. Sistem memberikan pilihan untuk melakukan tambah data type QSP, ubah data type QSP, hapus data type QSP atau tampil data type QSP

3. Aktor memilih untuk melakukan tambah data type QSP

- A-1 Aktor memilih untuk melakukan ubah data type QSP
- A-2 Aktor memilih untuk melakukan hapus data type QSP
- A-3 Aktor memilih untuk melakukan tampil data type QSP
- A-4 Aktor memilih untuk melakukan search data type QSP
- 4. Aktor menginputkan data type QSP
- 5. Aktor meminta sistem untuk menyimpan data type QSP yang telah diinputkan
- 6. Sistem mengecek data type QSP yang telah diinputkan
 - E-1 Data type QSP yang diinputkan aktor salah
- 7. Sistem menyimpan data type QSP ke database
- 8. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan ubah data unit group
 - 1. Sistem meminta data type QSP yang akan diubah
 - 2. Aktor menginput data type QSP yang akan diubah berdasarkan keyword yang sudah ditentukan, misalnya ID
 - 3. Sistem mencari data type QSP dalam database
 - E-2 Data type QSP tidak ditemukan
 - 4. Sistem menampilkan data type QSP yang akan diubah
 - 5. Aktor mengubah data type QSP yang telah ditemukan
 - 6. Aktor menyimpan data type QSP yang sudah diubah
 - 7. Sistem melakukan pengecekan terhadap data type QSP yang telah diubah
 - E-3 Data type QSP yang telah diubah salah
 - 8. Sistem meyimpan data type QSP yang telah diubah ke database
 - 9. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan hapus data type QSP

1. Sistem meminta data type QSP yang akan dihapus
2. Aktor memasukkan data type QSP yang akan diubah berdasarkan keyword yang sudah ditentukan, misalnya ID
3. Sistem mencari data type QSP dalam database
E-2 Data type QSP tidak ditemukan
4. Sistem menampilkan data type QSP yang akan dihapus
5. Aktor meminta sistem untuk menghapus data type QSP yang telah ditampilkan
6. Sistem meminta konfirmasi penghapusan data
7. Aktor menyetujui penghapusan data type QSP

A-5 Aktor tidak menyetujui penghapusan data type QSP

8. Sistem menghapus data type QSP dari database
9. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk melakukan tampil data type QSP

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

A-4 Aktor memilih untuk melakukan cari data type QSP

1. Sistem akan menampilkan data type QSP
2. Aktor menginputkan data type QSP yang diminta
3. Sistem akan mencari data type QSP yang diminta.

E-2 Data type QSP tidak ditemukan

4. Sistem akan menampilkan data type QSP yang diminta
5. Berlanjut ke basic flow langkah 8

A-5 Aktor tidak menyetujui penghapusan data type QSP

1. Berlanjut ke Alternative Flow A-2 langkah ke 3

7. Error Flow

E-1 Data type QSP yang diinputkan aktor salah

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah

2. Kembali ke Basic Flow Langkah ke 4

E-2 Data type QSP tidak ditemukan

1. Sistem memberikan pesan peringatan bahwa data yang dicari tidak ditemukan

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

E-3 Data type QSP yang telah diubah salah

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah

2. Kembali ke Alternative Flow A-1 langkah ke 5

8. PreConditions

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

9. PostConditions

1. Data lama type QSP di database telah terupdate dengan data yang baru

4.1.5 Use Case Specification : Kelola Status QSP

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola status QSP dari user yang berinteraksi dengan sistem. Aktor dapat melakukan tambah data, ubah data, hapus data atau tampil data.

2. Primary Actor

1. Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan status QSP data status QSP

2. Sistem memberikan pilihan untuk melakukan tambah data status QSP, ubah data status QSP, hapus data status QSP atau tampil data status QSP

3. Aktor memilih untuk melakukan tambah data status QSP
 - A-1 Aktor memilih untuk melakukan ubah data status QSP
 - A-2 Aktor memilih untuk melakukan hapus data status QSP
 - A-3 Aktor memilih untuk melakukan tampil data status QSP
 - A-4 Aktor memilih untuk melakukan cari data status QSP
4. Aktor menginputkan data status QSP
5. Aktor meminta sistem untuk menyimpan data status QSP yang telah diinputkan
6. Sistem mengecek data status QSP yang telah diinputkan
 - E-1 Data status QSP yang diinputkan aktor salah
7. Sistem menyimpan data status QSP ke database
8. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan ubah data status QSP
 1. Sistem meminta data status QSP yang akan diubah
 2. Aktor menginput data status QSP yang akan diubah berdasarkan keyword yang sudah ditentukan, misalnya ID
 3. Sistem mencari data status QSP dalam database
 - E-2 Data status QSP tidak ditemukan
 4. Sistem menampilkan data status QSP yang akan diubah
 5. Aktor mengubah data status QSP yang telah ditemukan
 6. Aktor menyimpan data status QSP yang sudah diubah
 7. Sistem melakukan pengecekan terhadap data status QSP yang telah diubah

E-3 Data status QSP yang telah diubah salah

8. Sistem menyimpan data status QSP yang telah diubah ke database

9. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan hapus data status QSP

1. Sistem meminta data status QSP yang akan dihapus

2. Aktor memasukkan data status QSP yang akan diubah berdasarkan keyword yang sudah ditentukan, misalnya ID.

3. Sistem mencari data status QSP dalam database

E-2 Data status QSP tidak ditemukan

4. Sistem menampilkan data status QSP yang akan dihapus

5. Aktor meminta sistem untuk menghapus data status QSP yang telah ditampilkan

6. Sistem meminta konfirmasi penghapusan data

7. Aktor menyetujui penghapusan data status QSP

A-5 Aktor tidak menyetujui penghapusan data status QSP

8. Sistem menghapus data status QSP dari database

9. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk melakukan tampil data status QSP

1. Sistem menampilkan data atau profile status QSP

2. Berlanjut ke Basic Flow langkah ke 8

A-4 Aktor memilih untuk melakukan search data status QSP

1. Sistem akan menampilkan data status QSP

2. Aktor menginputkan data status QSP yang diminta

3. Sistem akan mencari data status QSP yang diminta.

E-2 Data status QSP tidak ditemukan

4. Sistem akan menampilkan data status QSP yang diminta
 5. Berlanjut ke basic flow langkah 8
- A-5 Aktor tidak menyetujui penghapusan data status QSP
1. Berlanjut ke Alternative Flow A-2 langkah ke 3

6. Error Flow

E-1 Data status QSP yang diinputkan aktor salah

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

E-2 Data status QSP tidak ditemukan

1. Sistem memberikan pesan peringatan bahwa data yang dicari tidak ditemukan
2. Kembali ke Alternative Flow A-1 atau Alternative Flow A-2 Langkah ke 2

E-3 Data status QSP yang telah diubah salah

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah
2. Kembali ke Alternative Flow A-1 langkah ke 5

7. PreConditions

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

8. PostConditions

1. Data lama status QSP di database telah terupdate dengan data yang baru

4.1.6 Use Case Specification : Kelola Jabatan User

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data jabatan user dari user yang berinteraksi dengan sistem. Aktor dapat melakukan tambah data, ubah data, hapus data atau tampil data.

2. Primary Actor

1. Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data jabatan user data data jabatan user
2. Sistem memberikan pilihan untuk melakukan tambah data data jabatan user, ubah data data jabatan user, hapus data data jabatan user atau tampil data data jabatan user.
3. Aktor memilih untuk melakukan tambah data data jabatan user
 - A-1 Aktor memilih untuk melakukan ubah data data jabatan user
 - A-2 Aktor memilih untuk melakukan hapus data data jabatan user
 - A-3 Aktor memilih untuk melakukan tampil data data jabatan user
 - A-4 Aktor memilih untuk melakukan cari data data jabatan user
4. Aktor menginputkan data data jabatan user
5. Aktor meminta sistem untuk menyimpan data data jabatan user yang telah diinputkan
6. Sistem mengecek data data jabatan user yang telah diinputkan
 - E-1 Data data jabatan user yang diinputkan aktor salah
7. Sistem menyimpan data data jabatan user ke database
8. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan ubah data data jabatan user
 1. Sistem meminta data data jabatan user yang akan diubah

2. Aktor menginput data data jabatan user yang akan diubah berdasarkan keyword yang sudah ditentukan, misalnya ID

3. Sistem mencari data data jabatan user dalam database

E-2 Data data jabatan user tidak ditemukan

4. Sistem menampilkan data data jabatan user yang akan diubah

5. Aktor mengubah data data jabatan user yang telah ditemukan

6. Aktor menyimpan data data jabatan user yang sudah diubah

7. Sistem melakukan pengecekan terhadap data data jabatan user yang telah diubah

E-3 Data data jabatan user yang telah diubah salah

8. Sistem meyimpan data data jabatan user yang telah diubah ke database

9. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan hapus data data jabatan user

1. Sistem meminta data data jabatan user yang akan dihapus

2. Aktor memasukkan data data jabatan user yang akan diubah berdasarkan keyword yang sudah ditentukan, misalnya ID

3. Sistem mencari data data jabatan user dalam database

E-2 Data data jabatan user tidak ditemukan

4. Sistem menampilkan data data jabatan user yang akan dihapus

5. Aktor meminta sistem untuk menghapus data data jabatan user yang telah ditampilkan

6. Sistem meminta konfirmasi penghapusan data

7. Aktor menyetujui penghapusan data data jabatan user

A-5 Aktor tidak menyetujui penghapusan data data jabatan user

8. Sistem menghapus data data jabatan user dari database

9. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk melakukan tampil data data jabatan user

1. Sistem menampilkan data atau profile data jabatan user

2. Berlanjut ke Basic Flow langkah ke 8

A-4 Aktor memilih untuk melakukan search data data jabatan user

1. Sistem akan menampilkan data data jabatan user

2. Aktor menginputkan data data jabatan user yang diminta

3. Sistem akan mencari data data jabatan user yang diminta

E-2 Data data jabatan user tidak ditemukan

4. Sistem akan menampilkan data data jabatan user yang diminta

5. Berlanjut ke basic flow langkah 8

A-5 Aktor tidak menyetujui penghapusan data data jabatan user

1. Berlanjut ke Alternative Flow A-2 langkah ke 3

6. Error Flow

E-1 Data data jabatan user yang diinputkan aktor salah

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah

2. Kembali ke Basic Flow Langkah ke 4

E-2 Data data jabatan user tidak ditemukan

1. Sistem memberikan pesan peringatan bahwa data yang dicari tidak ditemukan

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

E-3 Data data jabatan user yang telah diubah salah

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah

2. Kembali ke Alternative Flow A-1 langkah ke 5

7. PreConditions

1. Use Case Login telah dilakukan

2. Aktor telah memasuki sistem

8. PostConditions

1. Data lama data jabatan user di database telah ter-update dengan data yang baru

4.1.7 Use case Spesification : Kelola Akun User

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data user dari user yang berinteraksi dengan sistem. Aktor dapat melakukan tambah data, ubah data, hapus data atau tampil data.

2. Primary Actor

1. Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data user

2. Sistem memberikan pilihan untuk melakukan tambah data user, ubah data user, hapus data user atau tampil data user

3. Aktor memilih untuk melakukan tambah data user

A-1 Aktor memilih untuk melakukan ubah data user

A-2 Aktor memilih untuk melakukan hapus data user

A-3 Aktor memilih untuk melakukan tampil data user

A-4 Aktor memilih untuk melakukan search data user

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

5. Alternative Flow

A-1 Aktor memilih untuk melakukan ubah data user

1. Sistem meminta data user yang akan diubah
2. Aktor menginput data user yang akan diubah berdasarkan keyword yang sudah ditentukan, misalnya ID
3. Sistem mencari data user dalam database
 - E-2 Data user tidak ditemukan
4. Sistem menampilkan data user yang akan diubah
5. Aktor mengubah data user yang telah ditemukan
6. Aktor menyimpan data user yang sudah diubah
7. Sistem melakukan pengecekan terhadap data user yang telah diubah

E-3 Data user yang telah diubah salah

8. Sistem menyimpan data user yang telah diubah ke database
9. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan hapus data user

1. Sistem meminta data user yang akan dihapus
2. Aktor memasukkan data user yang akan dihapus berdasarkan keyword yang sudah ditentukan, misalnya ID
3. Sistem mencari data user dalam database
 - E-2 Data user tidak ditemukan
4. Sistem menampilkan data user yang akan dihapus

5. Aktor meminta sistem untuk menghapus data user yang telah ditampilkan
6. Sistem meminta konfirmasi penghapusan data
7. Aktor menyetujui penghapusan data user
 - A-5 Aktor tidak menyetujui penghapusan data user
8. Sistem menghapus data user dari database
9. Berlanjut ke Basic Flow langkah ke 8
- A-3 Aktor memilih untuk melakukan tampil data user
 1. Sistem menampilkan data atau profile user
 2. Berlanjut ke Basic Flow langkah ke 8
- A-4 Aktor memilih untuk melakukan search data user
 1. Sistem akan menampilkan data user
 2. Aktor menginputkan data user yang diminta
 3. Sistem akan mencari data user yang diminta.
 - E-2 Data user tidak ditemukan
 4. Sistem akan menampilkan data user yang diminta
 5. Berlanjut ke basic flow langkah 8
- A-5 Aktor tidak menyetujui penghapusan data user
 1. Berlanjut ke Alternative Flow A-2 langkah ke 3

6. Error Flow

- E-1 Data user yang diinputkan aktor salah
 1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah
 2. Kembali ke Basic Flow Langkah ke 4
- E-2 Data user tidak ditemukan
 1. Sistem memberikan pesan peringatan bahwa data yang dicari tidak ditemukan
 2. Kembali ke Alternative Flow A-1 atau Alternative Flow A-2 Langkah ke 2
- E-3 Data user yang telah diubah salah
 1. Sistem memberikan pesan peringatan bahwa data yang diubah salah
 2. Kembali ke Alternative Flow A-1 langkah ke 5

7. PreConditions

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

8. PostConditions

1. Data lama user di database telah terupdate dengan data yang baru

4.1.8 Use case Spesification : Buat QSP

1. Brief Description

Use Case ini memungkinkan aktor untuk membuat data QSP yang baru.

2. Primary Actor

1. QSP Maker
2. Approver
3. Management Representative

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk membuat QSP baru
2. Aktor menginputkan data-data yang diperlukan dalam pembuatan QSP
 - A-1 Aktor mengassign QSP kepada approver
3. Aktor meminta sistem untuk menyimpan data yang telah diinputkan dalam bentuk draft
 - E-1 Aktor tidak lengkap atau ada kesalahan dalam pengisian data QSP baru.
4. Sistem menyimpan data draft QSP kedalam database.
5. Use case selesai.

5. Alternative Flow

- A-1 Aktor mengassign QSP kepada approver
1. Aktor memilih approver yang berhak untuk memberikan persetujuan QSP

2. Aktor memilih untuk mengassign QSP kepada approver yang berwenang
3. Sistem mengirim data QSP kepada approver yang berwenang untuk diperiksa
4. Kembali ke Basic Flow Langkah 5

6. Error Flow

E-1 Aktor tidak lengkap atau ada kesalahan dalam pengisian data QSP

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

7. PreConditions

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

8. PostConditions

1. Data QSP yang baru sudah dibuat dan disimpan dalam database sebagai draft QSP
2. Data QSP yang sudah selesai dibuat sudah terkirim kepada approver

4.1.9 Use case Spesification : Kelola Draft QSP

1. Brief Description

Use Case ini digunakan aktor untuk melakukan pengelolaan draft QSP.

2. Primary Actor

1. Approver
2. QSP Maker
3. Management Representative

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan draft QSP

2. Sistem memberikan pilihan untuk melakukan ubah draft QSP dan tampil draft QSP
3. Aktor memilih untuk melakukan ubah draft QSP
 - A-1 Aktor memilih untuk melakukan tampil draft QSP
4. Aktor memasukkan kata kunci draft QSP yang ingin diubah
5. Sistem mencari draft QSP yang dicari
 - E-1 Draft QSP yang dicari tidak ditemukan
6. Sistem menampilkan draft QSP yang ingin diubah
7. Aktor memasukkan data-data yang ingin diubah
8. Aktor meminta sistem untuk menyimpan data yang sudah diubah
9. Sistem akan menyimpan draft QSP yang sudah diubah kedalam database
10. Use case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan tampil data draft QSP

1. Sistem menampilkan data draft QSP
2. Berlanjut ke Basic Flow langkah ke 10

6. Error Flow

E-1 Draft QSP yang dicari tidak ditemukan

1. Sistem memberikan pesan peringatan bahwa data yang dicari tidak ditemukan
2. Kembali ke basic flow langkah ke 4

7. PreConditions

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

8. PostConditions

1. Data draft QSP dalam database sudah *ter-update*

4.1.10 Use case Specification : Kelola List Approval

1. Brief Description

Use Case ini memungkinkan aktor untuk mengelola approval dokumen QSP.

2. Primary Actor

1. Approver
2. Managemet Representative

3. Supporting Actor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan persetujuan QSP yang telah diassign kepada approver
2. Aktor memilih untuk melakukan approve QSP, ubah QSP atau reject QSP
3. Aktor memilih untuk melakukan approve QSP
 - A-1 Aktor memilih untuk melakukan reject QSP yang telah diassign
 - A-2 Aktor memilih untuk melakukan tampil QSP yang telah diassign
4. Aktor memilih data QSP yang akan diapprove
5. Aktor meminta sistem untuk menyimpan data QSP yang sudah di-approve
6. Sistem menyimpan data QSP kedalam database
7. Use case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan reject QSP yang telah diassign
1. Sistem memilih data QSP yang akan direject
 2. Aktor memasukkan catatan revisi untuk QSP yang direject
 3. Aktor mengirim data QSP yang direject kepada pembuat QSP

4. Berlanjut ke basic flow langkah 7

A-2 Aktor memilih untuk melakukan tampil QSP yang telah diassign

1. Sistem menampilkan data QSP yang sudah diassign

2. Berlanjut ke Basic Flow langkah 7

6. Error Flow

None

7. PreConditions

1. Use Case Login sudah dilakukan

2. Aktor telah memasuki sistem

8. PostConditions

1. Data pada list approval dan QSP yang dipublish telah terupdate

4.1.11 Use case Spesification : Kelola Published QSP

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan QSP yang telah terpublish.

2. Primary Actor

1. Approver

2. QSP Maker

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data QSP yang sudah dipublish

2. Sistem memberikan pilihan untuk melakukan tampil published QSP, update status published QSP dan revisi published QSP

3. Aktor memilih untuk melakukan tampil published QSP

A-1 Aktor memilih untuk melakukan update status published QSP

A-2 Aktor memilih untuk melakukan revisi published QSP

4. Sistem menampilkan data published QSP yang diinginkan oleh aktor

5. Use case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan update status published QSP

1. Sistem meminta aktor memilih data published QSP yang ingin diupdate statusnya

2. Sistem akan melakukan pengupdate-an status data published QSP yang sudah tidak terpakai

3. Berlanjut ke basic flow langkah 5

A-2 Aktor memilih untuk melakukan revisi published QSP

1. Sistem meminta aktor memilih data published QSP yang akan direvisi

2. Aktor melakukan revisi data published QSP

3. Aktor meminta persetujuan untuk QSP yang direvisi

4. Berlanjut ke Basic Flow langkah 5

6. Error Flow

None

7. PreConditions

1. Use Case Login sudah dilakukan

2. Aktor telah memasuki sistem

8. PostConditions

1. Data published QSP dalam database berubah.

4.1.12 Use case Spesification : Kelola Returned QSP

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data QSP yang direject atau dikembalikan oleh approver.

2. Primary Actor

1. QSP Maker
2. Approver
3. Management Representative

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data QSP yang telah dikembalikan oleh approver kepada QSP Maker yang bersangkutan
2. Sistem memberikan pilihan untuk melakukan ubah returned QSP dan tampil returned QSP
3. Aktor memilih untuk melakukan ubah returned QSP
 - A-1 Aktor memilih untuk melakukan tampil returned QSP
4. Sistem menampilkan data returned QSP yang ingin diubah
5. Aktor mengubah data returned QSP
6. Aktor meminta sistem untuk menyimpan data returned QSP yang sudah diubah
7. Sistem menyimpan data dalam database
8. Use case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan tampil data returned QSP
 1. Sistem menampilkan data returned QSP
 2. Berlanjut ke Basic Flow langkah ke 8

5. Error Flow

None

6. PreConditions

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

7. PostConditions

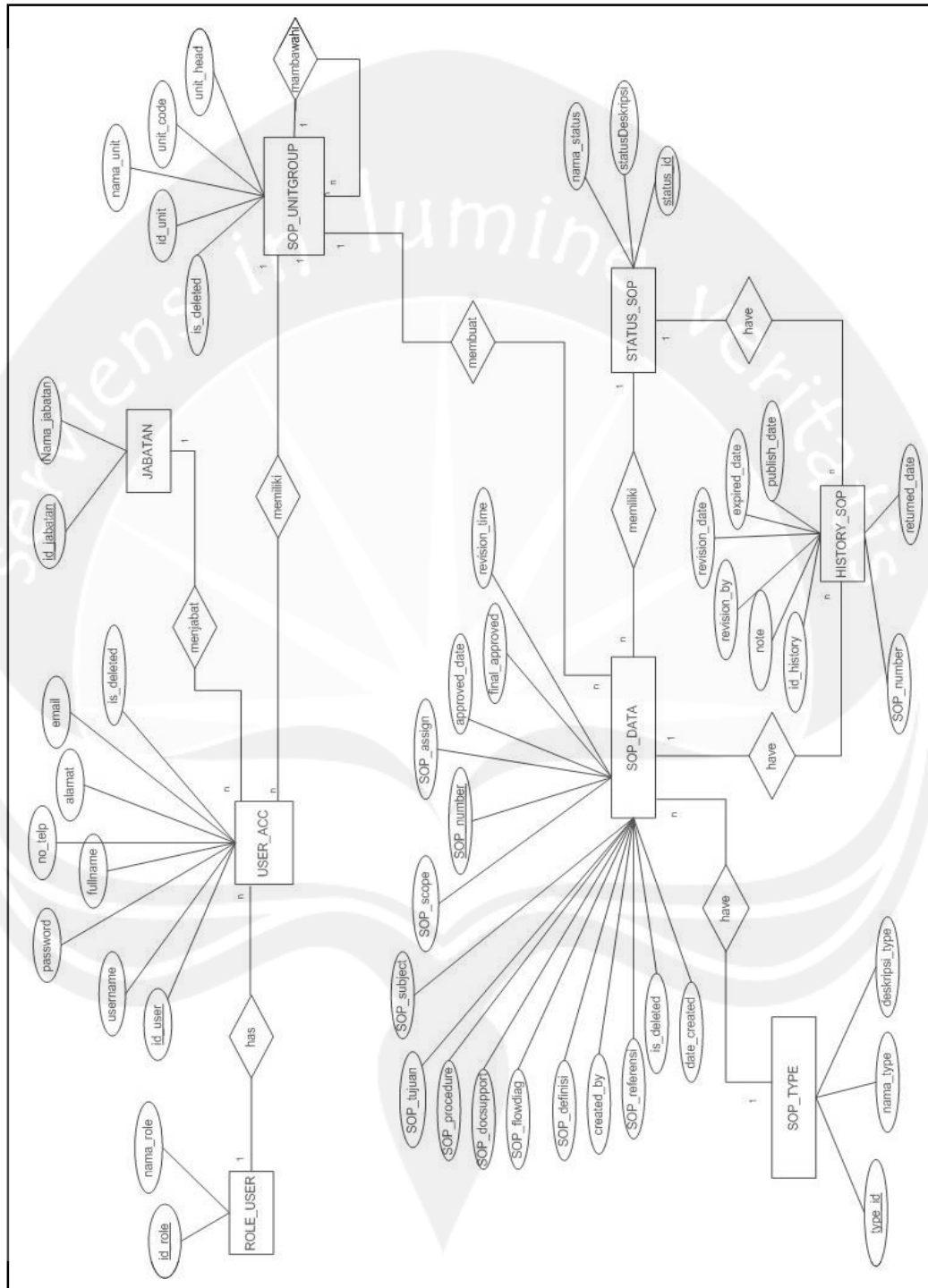
1. Data returned QSP dalam database berubah atau berkurang

8. PostConditions

1. Data returned QSP dalam database berubah.



5 Entity Relationship Diagram (ERD)



DPPL


DESKRIPSI PERANCANGAN PERANGKAT LUNAK
PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN
QUALITY SYSTEM UNIVERSITAS ATMA JAYA
YOGYAKARTA

Untuk :
Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:
Herminda Aprisari / 5460

Program Studi Teknik Informatika - Fakultas Teknologi
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<i>DPPL-SIQ</i>		1/62
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

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

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

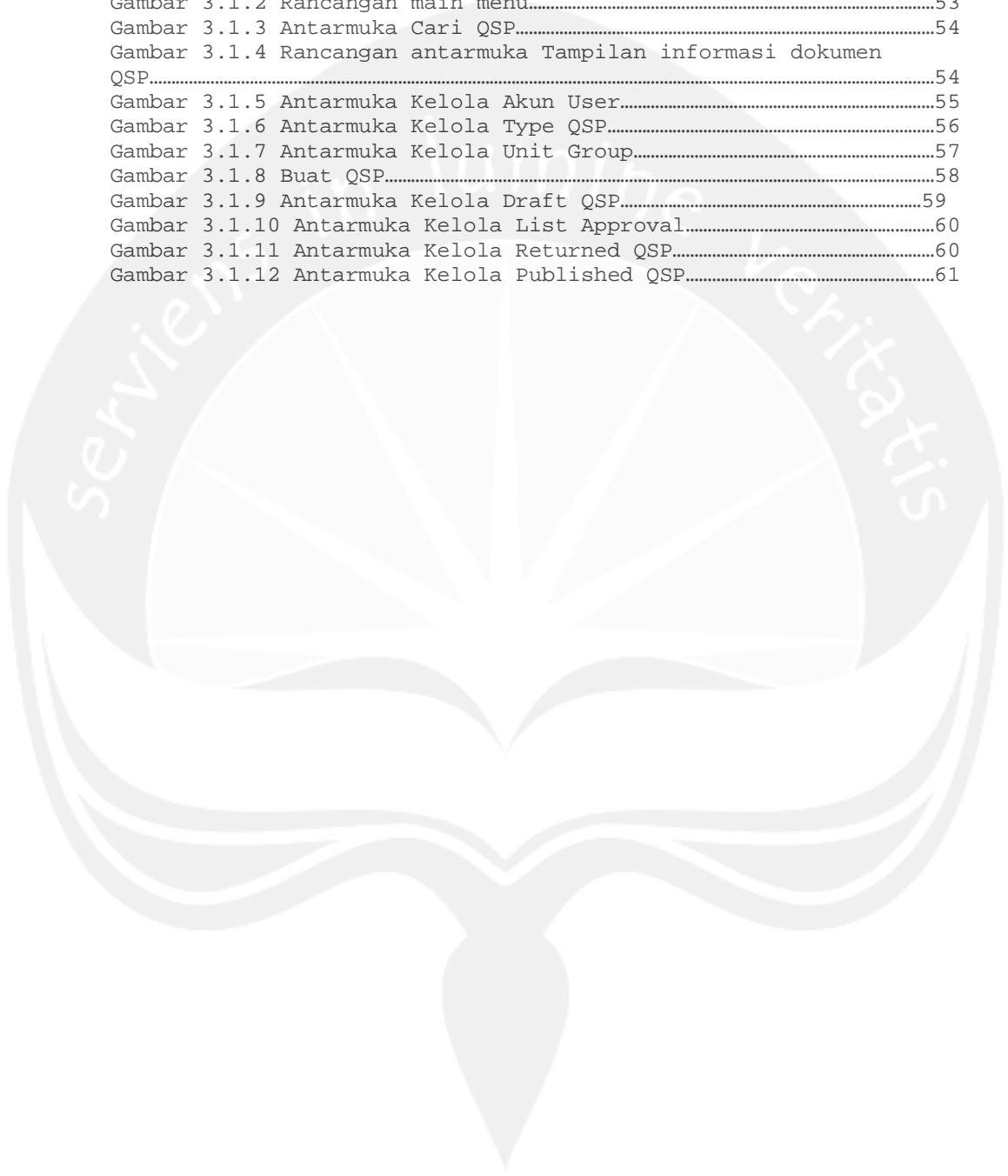
Daftar Isi

1	Pendahuluan.....	7
1.1	Tujuan.....	7
1.2	Ruang Lingkup.....	7
1.3	Definisi dan Akronim.....	7
1.4	Referensi.....	8
2	Perancangan Sistem.....	9
2.1	Perancangan Arsitektur.....	9
2.2	Perancangan Rinci.....	10
2.2.1	Sequence Diagram.....	10
2.2.2	Class Diagram.....	30
2.2.3	Deskripsi Class.....	33
3.	Perancangan Data.....	51
3.1	Dekomposisi Data.....	51
3.2	Physical Data Model.....	55
4	Perancangan Antarmuka.....	56
4.1	Desain.....	56
4.2	Deskripsi.....	56

Daftar Gambar

Gambar 2.1.1 Perancangan Arsitektur.....	9
Gambar 2.2.1 Sequence Diagram - Login.....	10
Gambar 2.2.2 Sequence Diagram - Cari QSP.....	10
Gambar 2.2.3 Sequence Diagram - Kelola Unit Kerja - Tambah Data Unit Kerja.....	11
Gambar 2.2.4 Sequence Diagram - Kelola Unit Kerja - Ubah Data Unit Kerja	11
Gambar 2.2.5 Sequence Diagram - Kelola Unit Kerja - Hapus Data Unit Kerja.....	12
Gambar 2.2.6 Sequence Diagram - Kelola Type QSP - Tambah Data Type QSP.....	13
Gambar 2.2.7 Sequence Diagram - Kelola Type QSP - Ubah Data Type QSP.....	13
Gambar 2.2.8 Sequence Diagram - Kelola Type QSP - Hapus Data Type QSP.....	14
Gambar 2.2.9 Sequence Diagram - Kelola Status QSP - Tambah Data Status QSP.....	14
Gambar 2.2.10 Sequence Diagram - Kelola Status QSP - Ubah Data Status QSP.....	15
Gambar 2.2.11 Sequence Diagram - Kelola Status QSP - Hapus Data Status QSP.....	15
Gambar 2.2.12 Sequence Diagram - Kelola Jabatan User - Tambah Data Jabatan User.....	16
Gambar 2.2.13 Sequence Diagram - Kelola Jabatan User - Ubah Data Jabatan User.....	16
Gambar 2.2.14 Sequence Diagram - Kelola Jabatan User - Hapus Data Jabatan User	17
Gambar 2.2.15 Sequence Diagram - Kelola Akun User - Tambah Data Akun User.....	18
Gambar 2.2.16 Sequence Diagram - Kelola Akun User - Ubah Data Akun User	18
Gambar 2.2.17 Sequence Diagram - Kelola Akun User - Hapus Data Akun User	19
Gambar 2.2.18 Sequence Diagram -Buat QSP.....	20
Gambar 2.2.19 Sequence Diagram - Kelola Draft QSP - Ubah Draft QSP.....	21
Gambar 2.2.20 Sequence Diagram - Kelola List Approval - Approved QSP.....	22
Gambar 2.2.21 Sequence Diagram - Kelola List Approval - Returned QSP	23
Gambar 2.2.22 Sequence Diagram - Kelola List Approval - Tampil Data QSP	24
Gambar 2.2.23 Sequence Diagram - Kelola Returned QSP - Ubah Returned QSP.....	24
Gambar 2.2.24 Sequence Diagram - Kelola Published QSP - Tampil Published QSP.....	25
Gambar 2.2.25 Sequence Diagram - Kelola Published QSP - Update Status Published QSP	26
Gambar 2.2.26 Sequence Diagram - Kelola Published QSP - Revisi Published QSP	27

Gambar 2.2.27 Class Diagram Perangkat Lunak.....	28
Gambar 3.2.1 Physical Data Model.....	52
Gambar 3.1.1 Rancangan antarmuka login.....	53
Gambar 3.1.2 Rancangan main menu.....	53
Gambar 3.1.3 Antarmuka Cari QSP.....	54
Gambar 3.1.4 Rancangan antarmuka Tampilan informasi dokumen QSP.....	54
Gambar 3.1.5 Antarmuka Kelola Akun User.....	55
Gambar 3.1.6 Antarmuka Kelola Type QSP.....	56
Gambar 3.1.7 Antarmuka Kelola Unit Group.....	57
Gambar 3.1.8 Buat QSP.....	58
Gambar 3.1.9 Antarmuka Kelola Draft QSP.....	59
Gambar 3.1.10 Antarmuka Kelola List Approval.....	60
Gambar 3.1.11 Antarmuka Kelola Returned QSP.....	60
Gambar 3.1.12 Antarmuka Kelola Published QSP.....	61



1 Pendahuluan

1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak SIQ yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

1.2 Ruang Lingkup

Perangkat Lunak SIQ dikembangkan dengan tujuan untuk :

1. Membangun aplikasi sistem informasi dokumen manajemen yang dapat menangani segala hal yang berkaitan dengan prosedur kerja.
2. Menangani proses bisnis yang berkaitan dengan pengelolaan QSP seperti *approval*, *searching* dan *published* yang semula masih manual.
3. Menangani proses *approval* dalam *assign* dokumen.

Dan berjalan pada lingkungan dengan platform Windows.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Merupakan deskripsi perancangan dari perangkat lunak yang akan dikembangkan.
SIQ	Perangkat lunak yang bertujuan membantu dalam pengelolaan QSP untuk Fakultas Teknologi Industri Universitas Atma Jaya

	Yogyakarta.
SOP /QSP	<i>Standard Operating Procedure / Quality System Procedure</i> yang merupakan langkah-langkah kerja yang mendeskripsikan prosedur rutin yang digunakan oleh suatu organisasi untuk mencapai suatu tujuan.
<i>Approver</i>	Pengguna sistem informasi yang memiliki role untuk melakukan persetujuan akan data yang akan ditampilkan kedalam sistem.
User/Aktor	Pengguna yang berinteraksi dengan sistem informasi.
<i>User Interface (UI)</i>	Perantara yang disediakan untuk user, misalnya interaksi dari mesin komputer ke layar monitor, sehingga seorang user dapat mengetahui apa yang terjadi pada sistem yang digunakannya.
<i>Update</i>	Memperbaharui atau bisa diartikan mengubah isi data yang lama dengan data yang baru, bisa dikurangi atau juga ditambah.

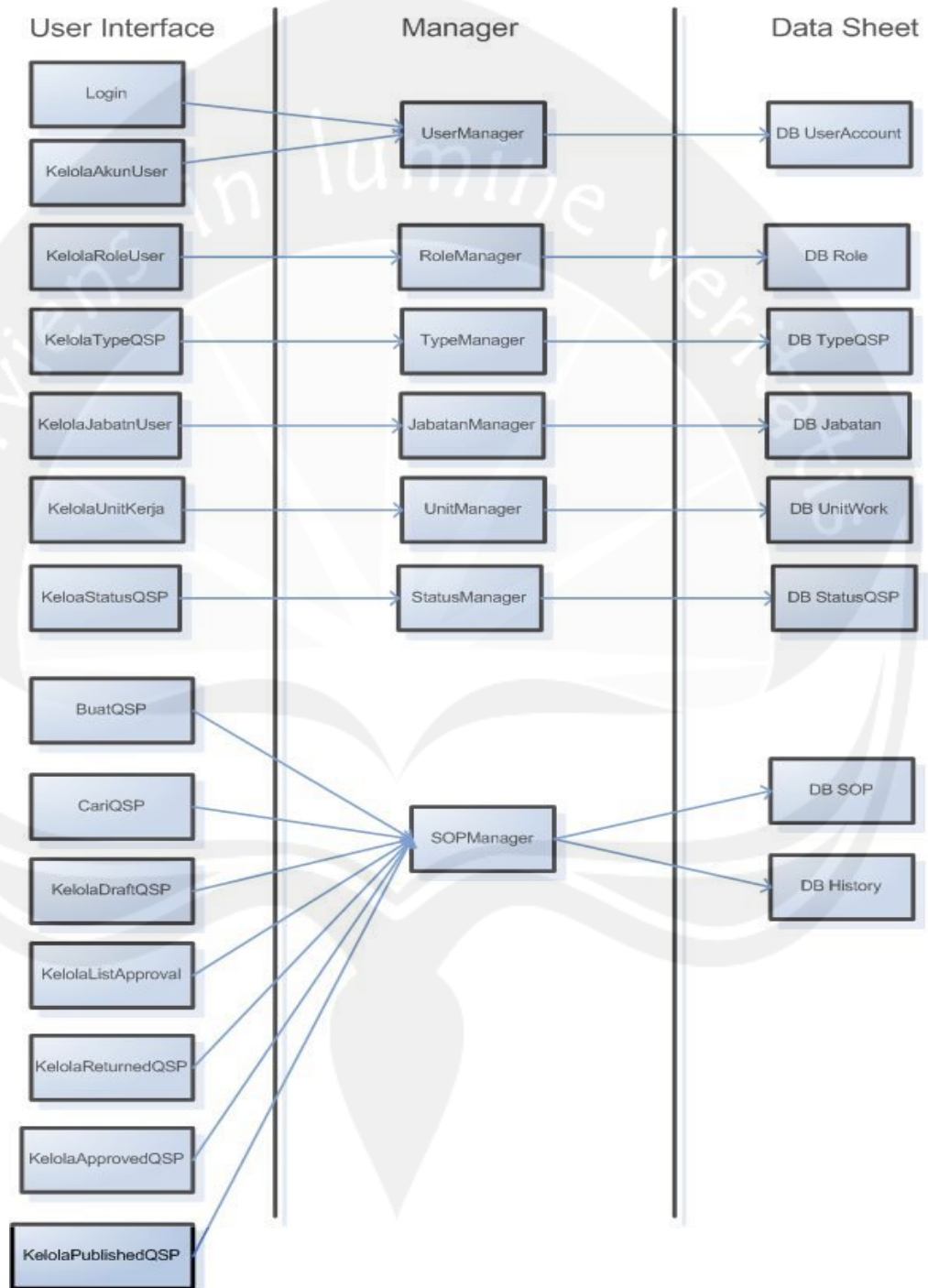
1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Aprisari Herminda, *Spesifikasi Kebutuhan Perangkat Lunak SIQ*, Universitas Atma Jaya Yogyakarta, 2011.
2. Dian, Christina, *Spesifikasi Kebutuhan Perangkat Lunak SISOP*, Universitas Atma Jaya Yogyakarta, 2010.

2. Perancangan Sistem

2.1 Perancangan Arsitektu

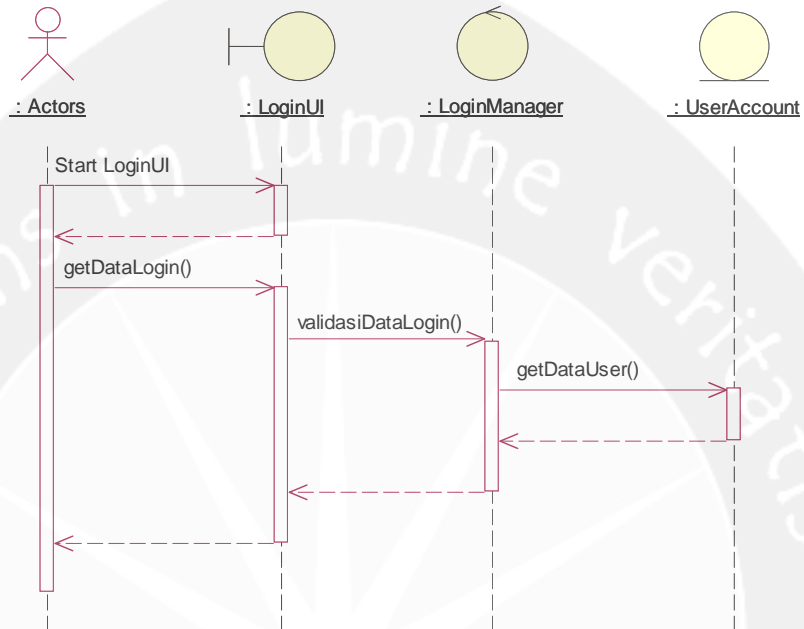


Gambar 2.1.1 Perancangan Arsitektur

2.2 Perancangan Rinci

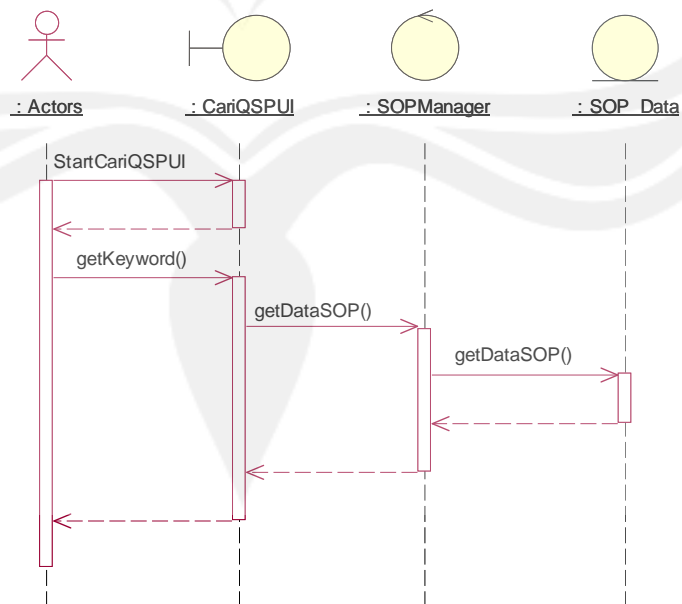
2.2.1 Sequence Diagram

2.2.1.1 Login



Gambar 2.2.1 Sequence Diagram - Login

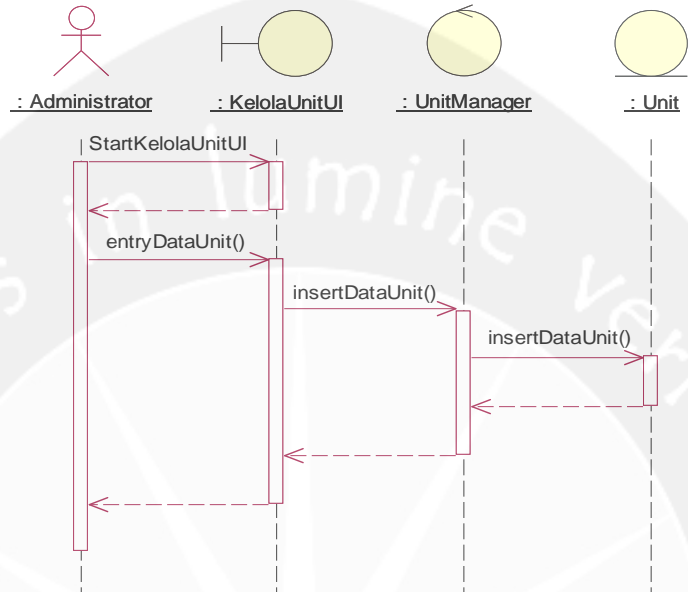
2.2.1.2 Cari QSP



Gambar 2.2.2 Sequence Diagram - Cari QSP

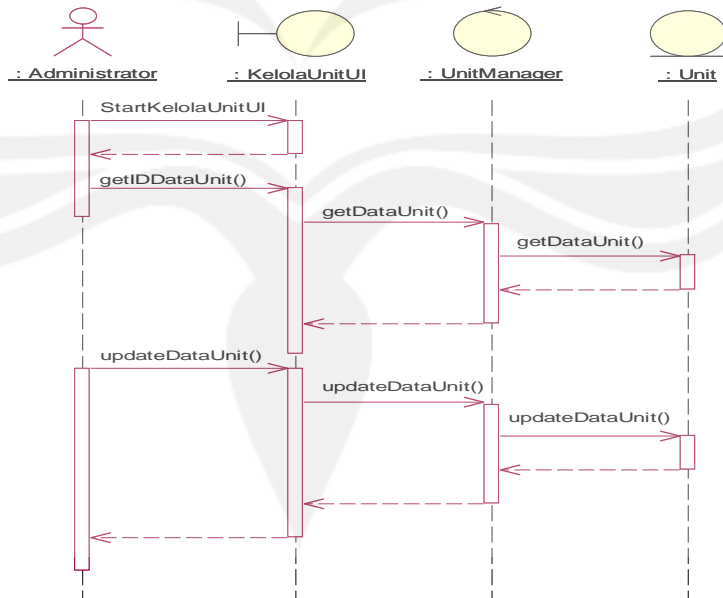
2.2.1.3 Kelola Unit Kerja

2.2.1.3.1 Tambah Data Unit Kerja



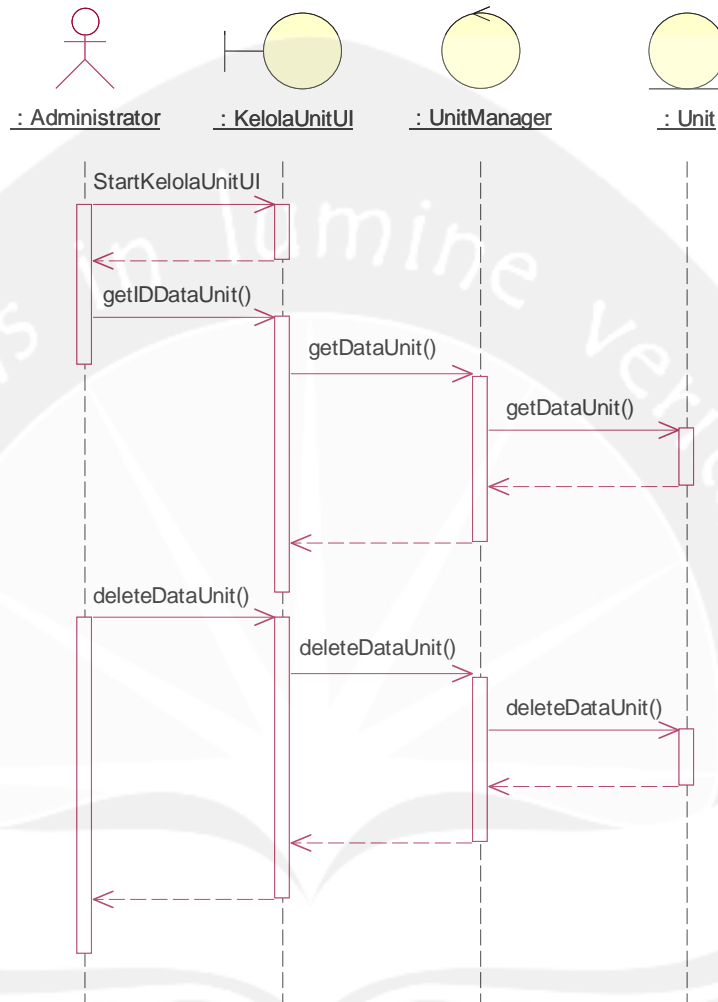
Gambar 2.2.3 Sequence Diagram - Kelola Unit Kerja - Tambah Data Unit Kerja

2.2.1.3.2 Ubah Data Unit Kerja



Gambar 2.2.4 Sequence Diagram - Kelola Unit Kerja - Ubah Data Unit Kerja

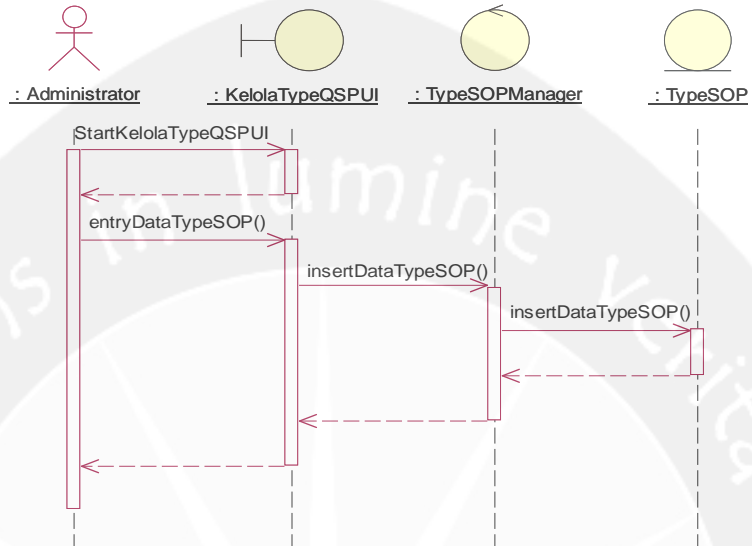
2.2.1.3.3 Hapus Data Unit Kerja



Gambar 2.2.5 Sequence Diagram - Kelola Unit Kerja - Hapus Data Unit Kerja

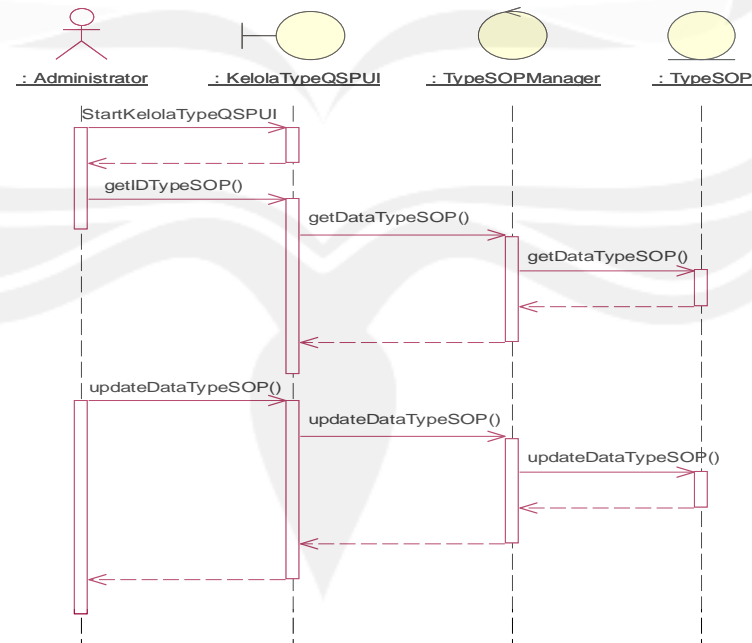
2.2.1.4 Kelola Type QSP

2.2.1.4.1 Tambah Data Type QSP



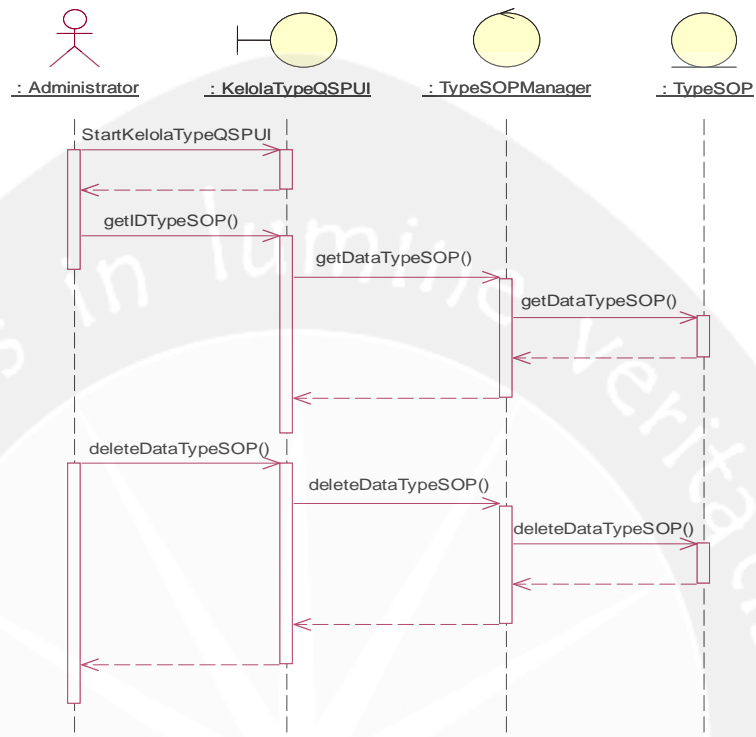
Gambar 2.2.6 Sequence Diagram - Kelola Type QSP - Tambah Data Type QSP

2.2.1.4.2 Ubah Data Type QSP



Gambar 2.2.7 Sequence Diagram - Kelola Type QSP - Ubah Data Type QSP

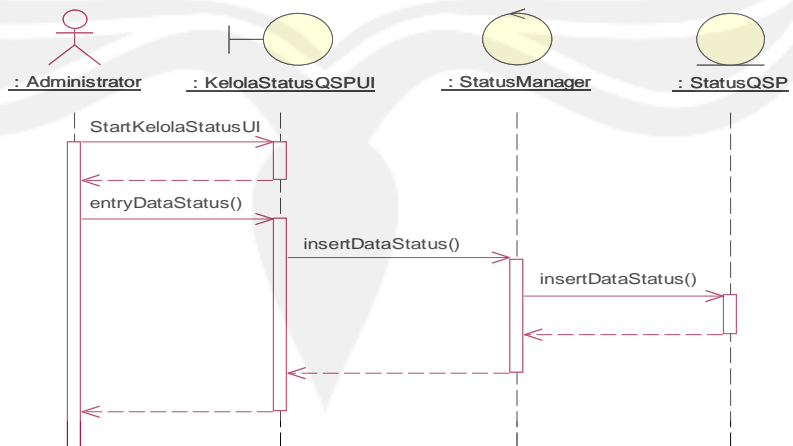
2.2.1.4.3 Hapus Data Type QSP



Gambar 2.2.8 Sequence Diagram - Kelola Type QSP - Hapus Data Type QSP

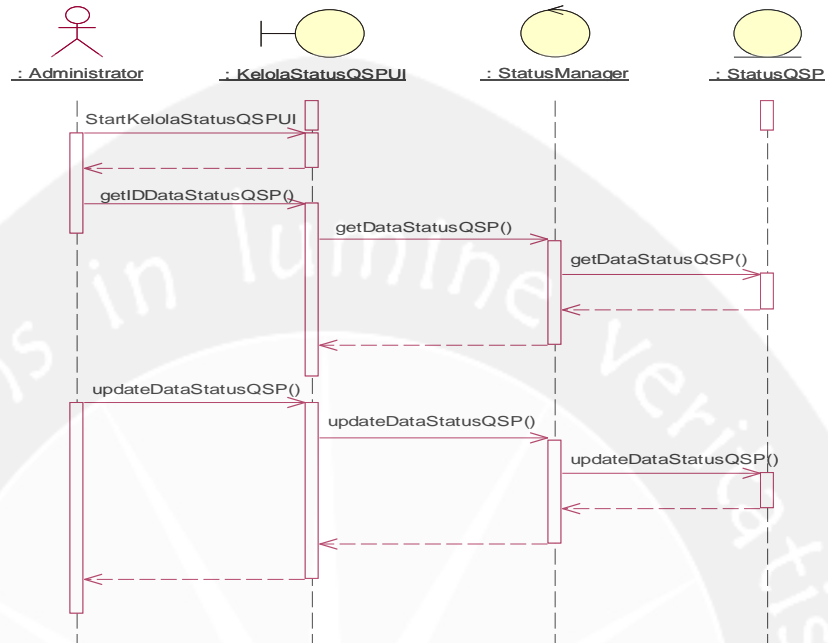
2.2.1.5 Kelola Status QSP

2.2.1.5.1 Tambah Data Status QSP



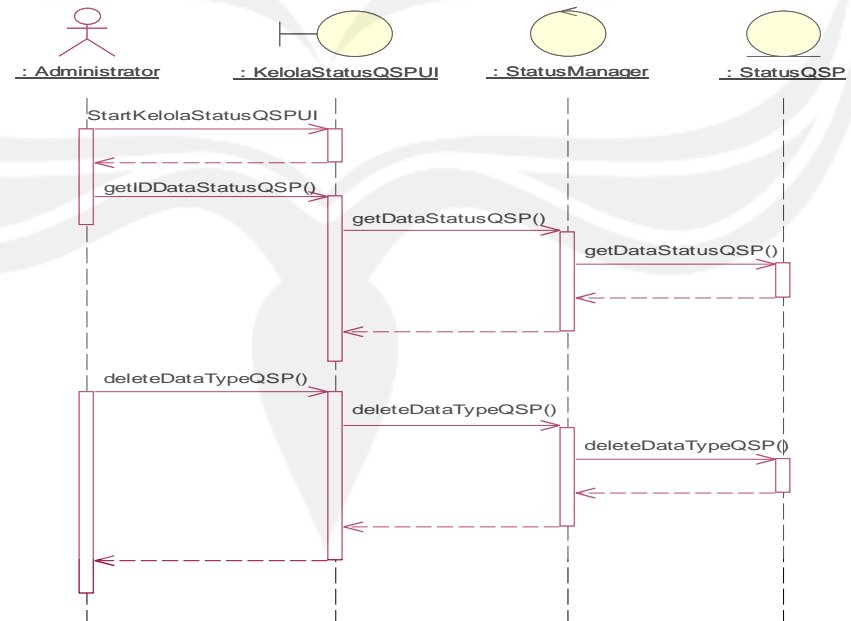
Gambar 2.2.9 Sequence Diagram - Kelola Status QSP - Tambah Data Status QSP

2.2.1.5.2 Ubah Data Status QSP



Gambar 2.2.10 Sequence Diagram - Kelola Status QSP - Ubah Data Status QSP

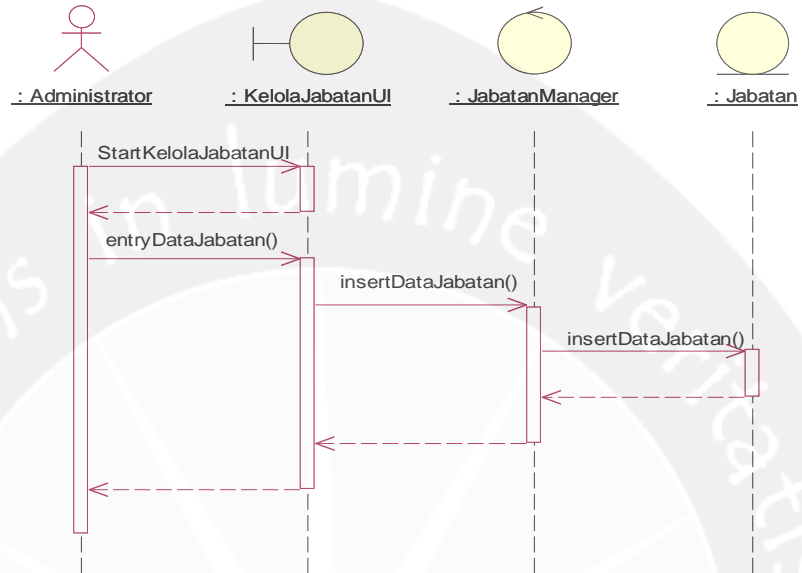
2.2.1.5.3 Hapus Data Status QSP



Gambar 2.2.11 Sequence Diagram - Kelola Status QSP - Hapus Data Status QSP

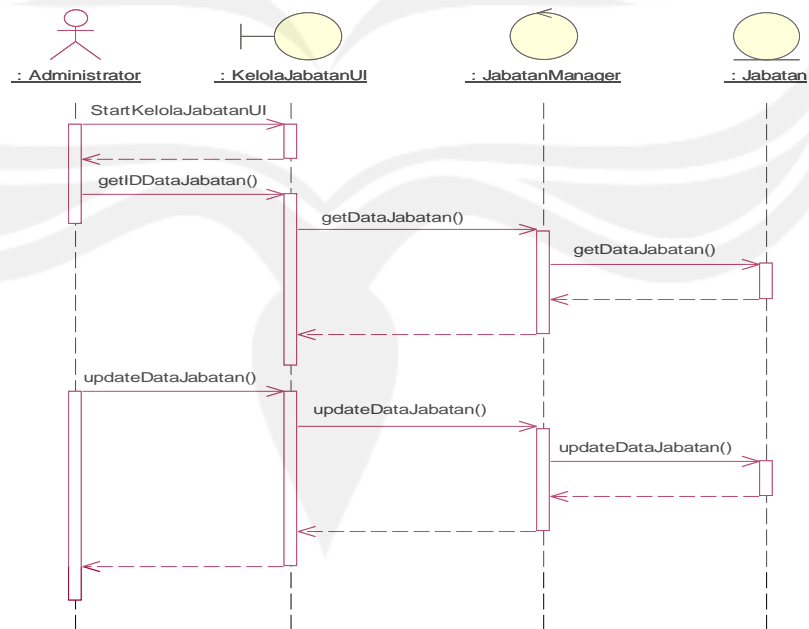
2.2.1.6 Kelola Jabatan User

2.2.1.6.1 Tambah Data Jabatan User



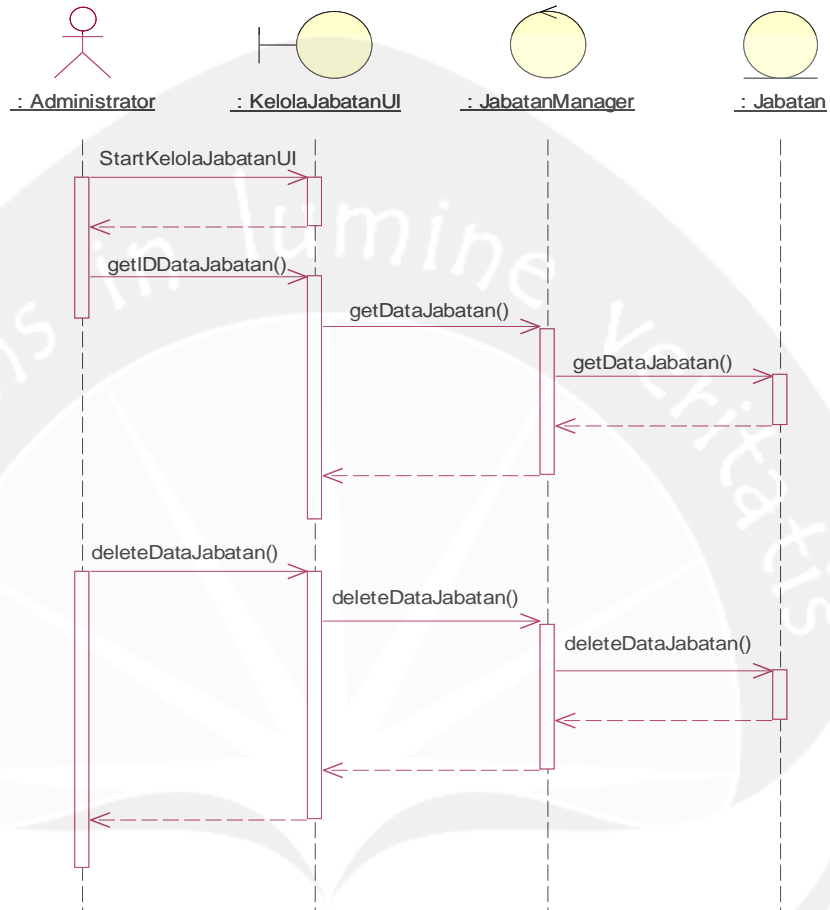
Gambar 2.2.12 Sequence Diagram - Kelola Jabatan User - Tambah Data Jabatan User

2.2.1.6.2 Ubah Data Jabatan User



Gambar 2.2.13 Sequence Diagram - Kelola Jabatan User - Ubah Data Jabatan User

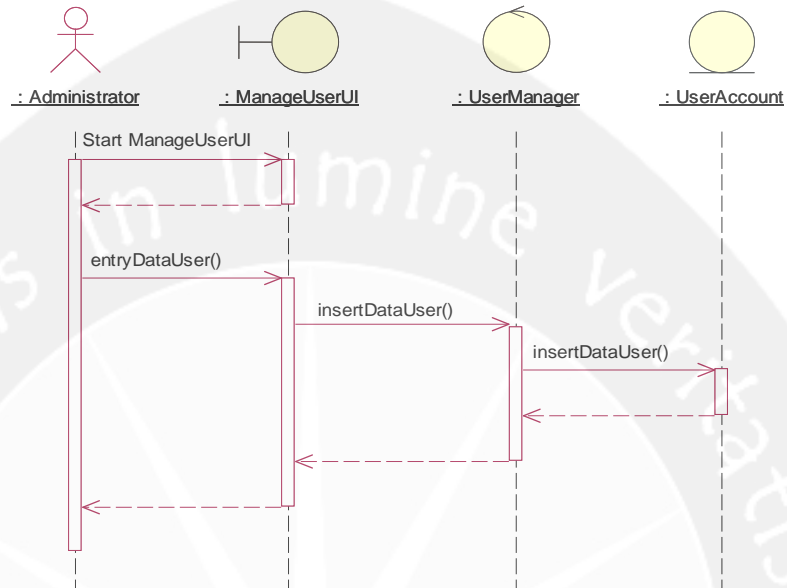
2.2.1.6.3 Hapus Data Jabatan User



Gambar 2.2.14 Sequence Diagram - Kelola Jabatan User - Hapus Data Jabatan User

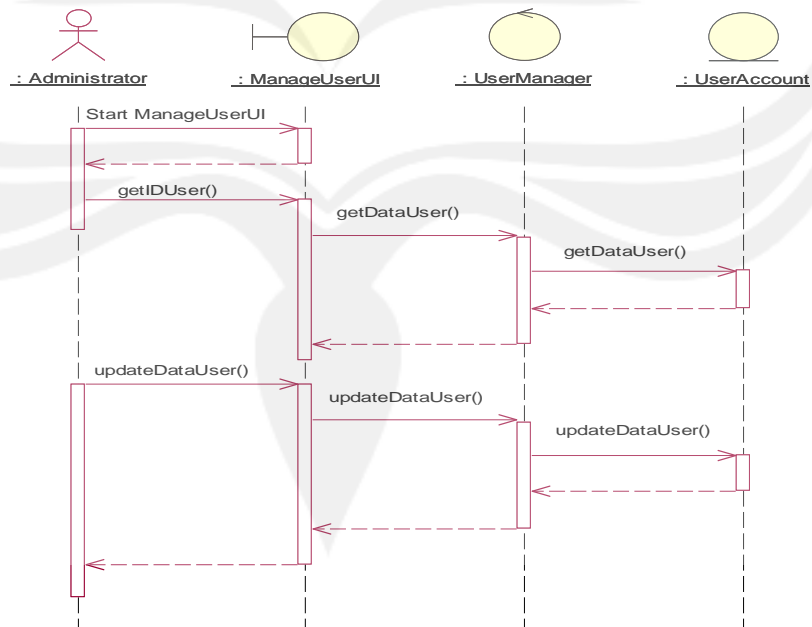
2.2.1.7 Kelola Akun User

2.2.1.7.1 Tambah Data Akun User



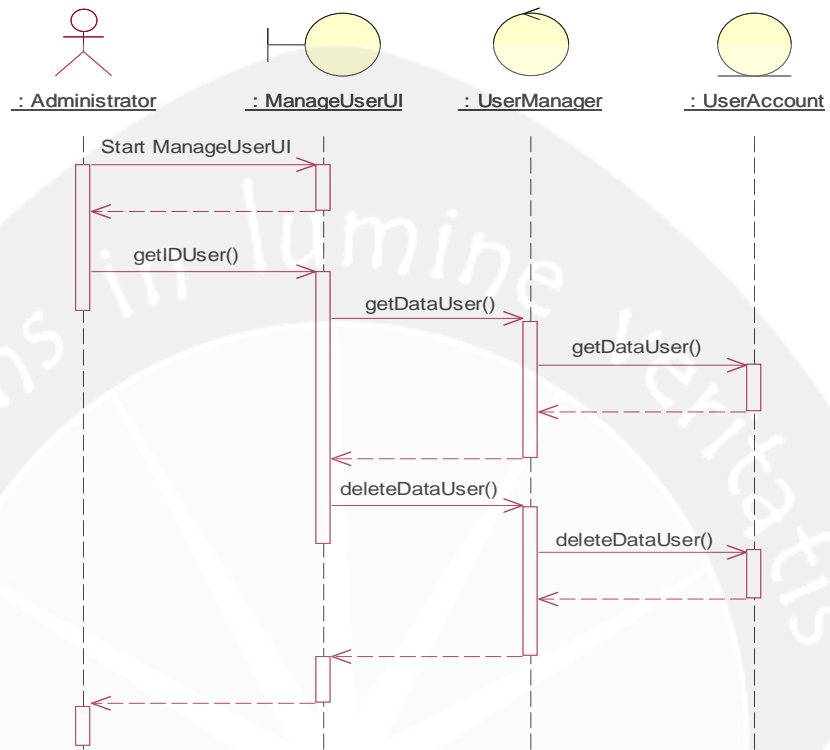
Gambar 2.2.15 Sequence Diagram - Kelola Akun User - Tambah Data Akun User

2.2.1.7.2 Ubah Data Akun User



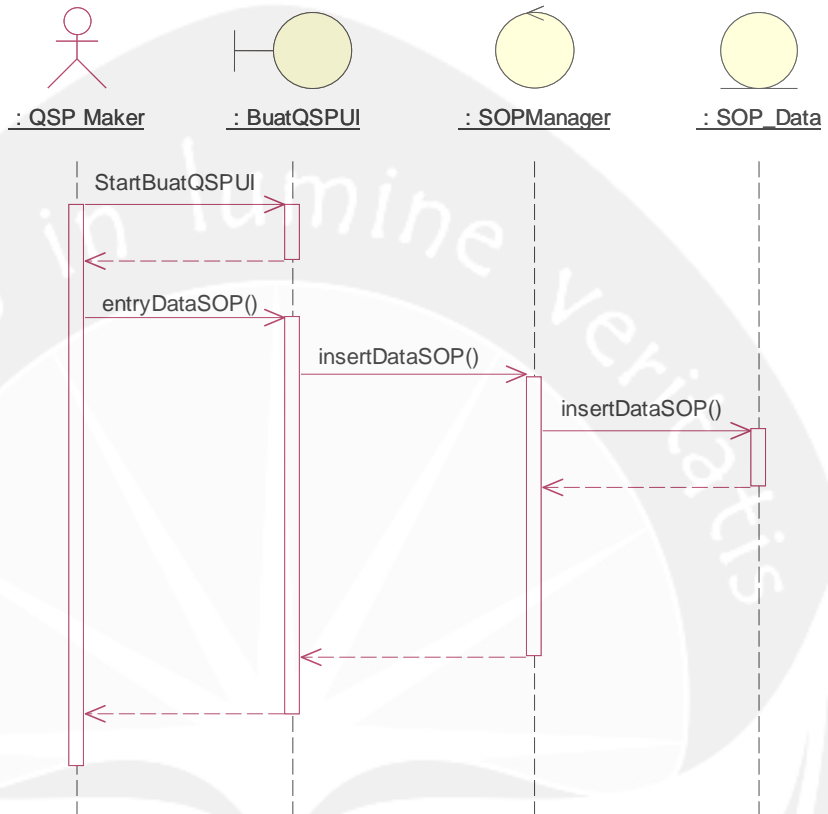
Gambar 2.2.16 Sequence Diagram - Kelola Akun User - Ubah Data Akun User

2.2.1.7.3 Hapus Data AKun User



Gambar 2.2.17 Sequence Diagram - Kelola Akun User - Hapus Data Akun User

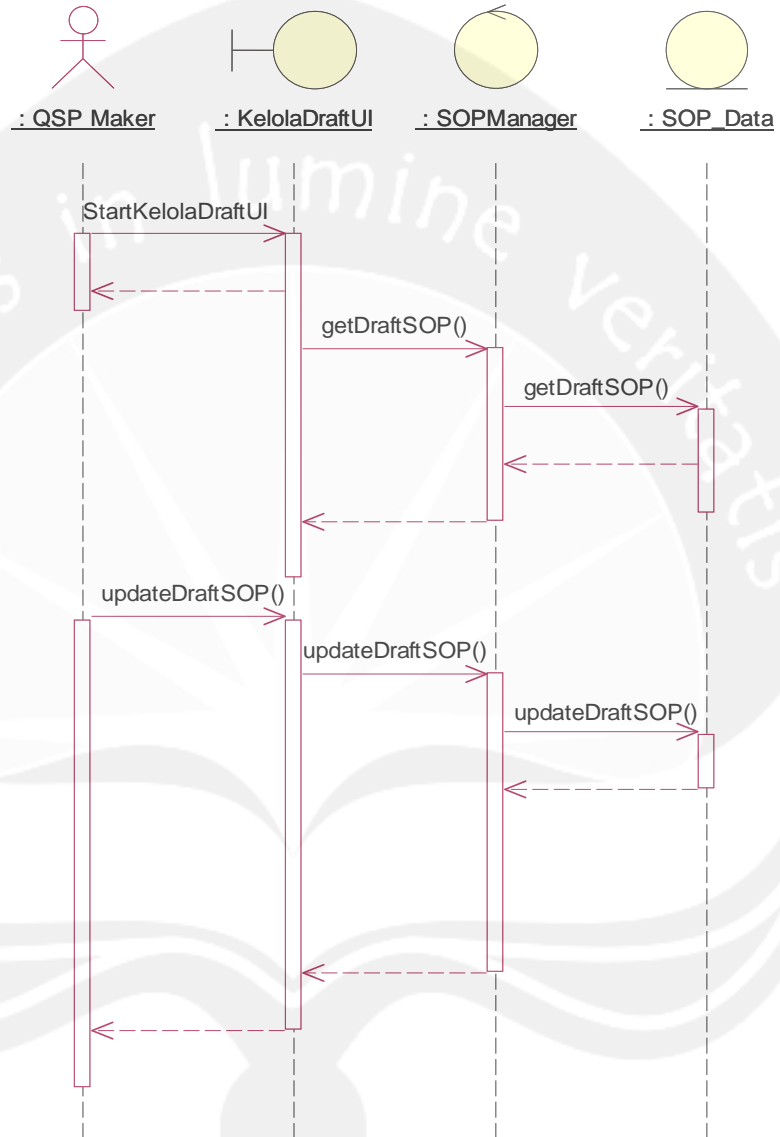
2.2.1.8 Buat QSP



Gambar 2.2.18 Sequence Diagram - Buat QSP

2.2.1.9 Kelola Draft QSP

2.2.1.9.1 Ubah Draft QSP



Gambar 2.2.19 Sequence Diagram - Kelola Draft QSP -
Ubah Draft QSP

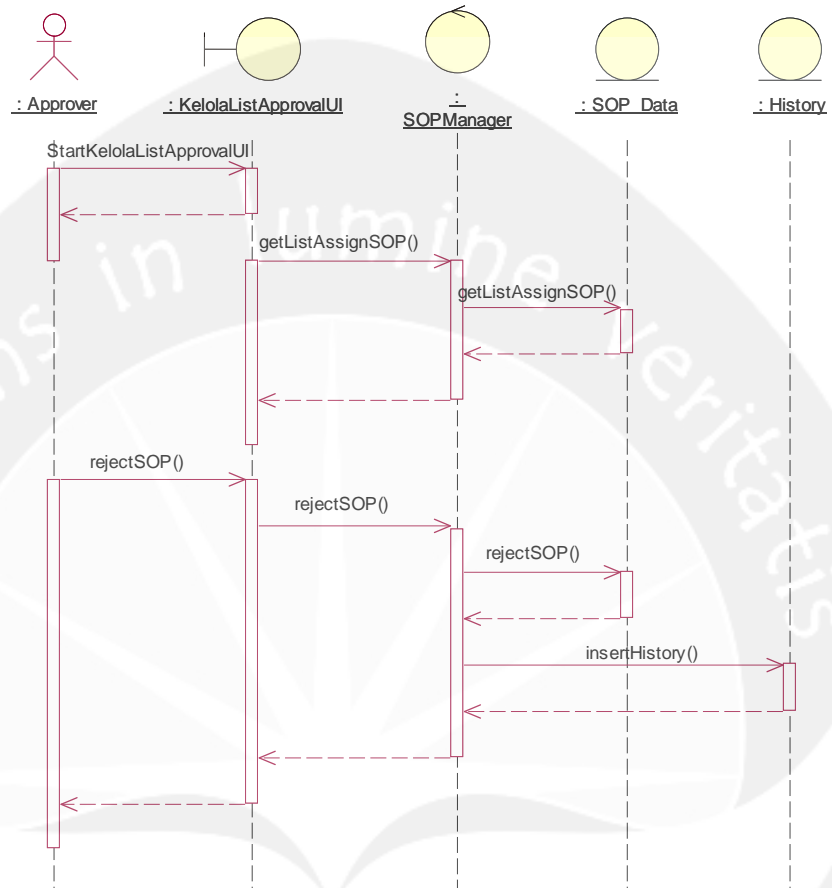
2.2.1.10 Kelola List Approval

2.2.1.10.1 Approved QSP



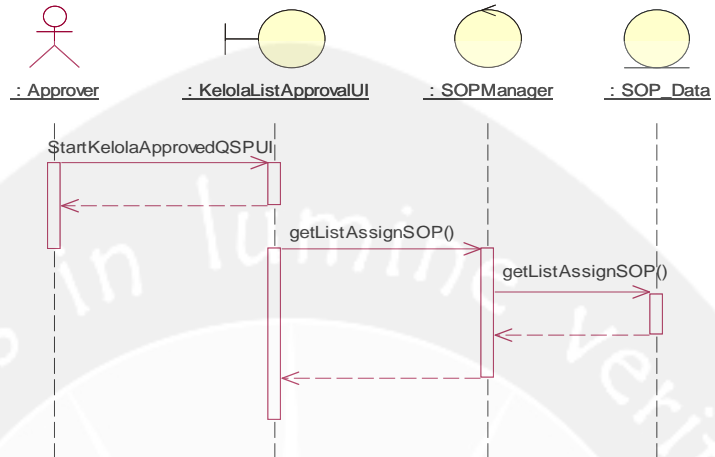
Gambar 2.2.20 Sequence Diagram - Kelola List Approval - Approved QSP

2.2.1.10.2 Returned QSP



Gambar 2.2.21 Sequence Diagram - Kelola List Approval - Returned QSP

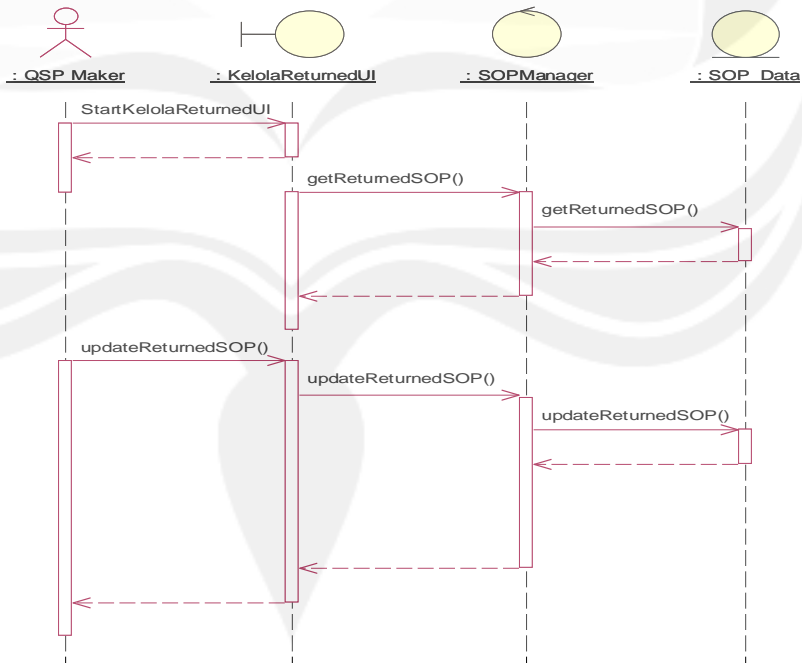
2.2.1.10.3 Tampil Data List Approval QSP



Gambar 2.2.22 Sequence Diagram - Kelola List Approval - Tampil Data List Approval QSP

2.2.1.11 Kelola Returned QSP

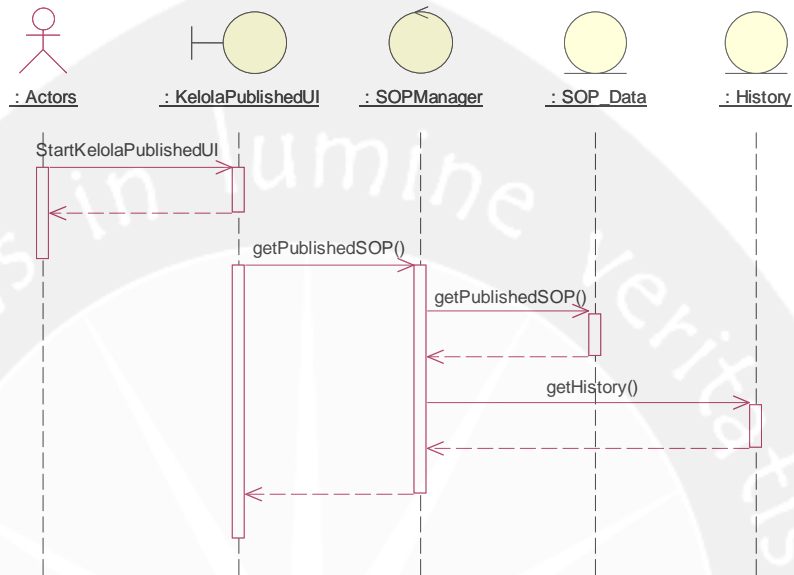
2.2.1.11.1 Ubah Returned QSP



Gambar 2.2.23 Sequence Diagram - Kelola Returned QSP - Ubah Returned QSP

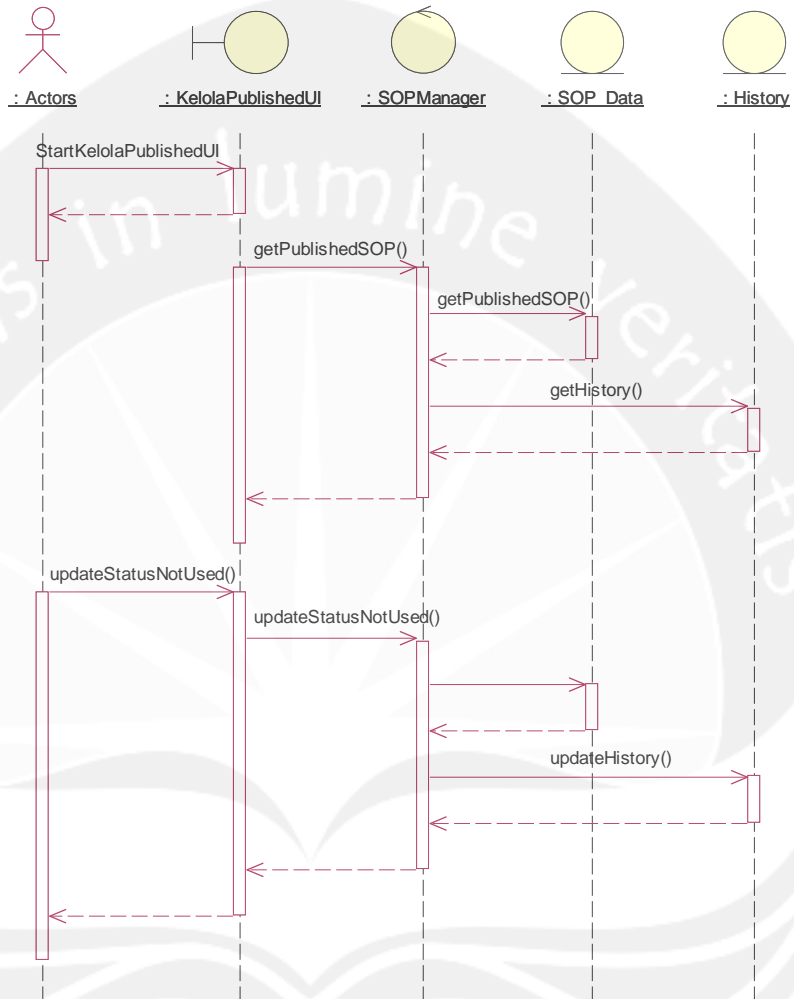
2.2.1.12 Kelola Published QSP

2.2.1.12.1 Tampil Published QSP



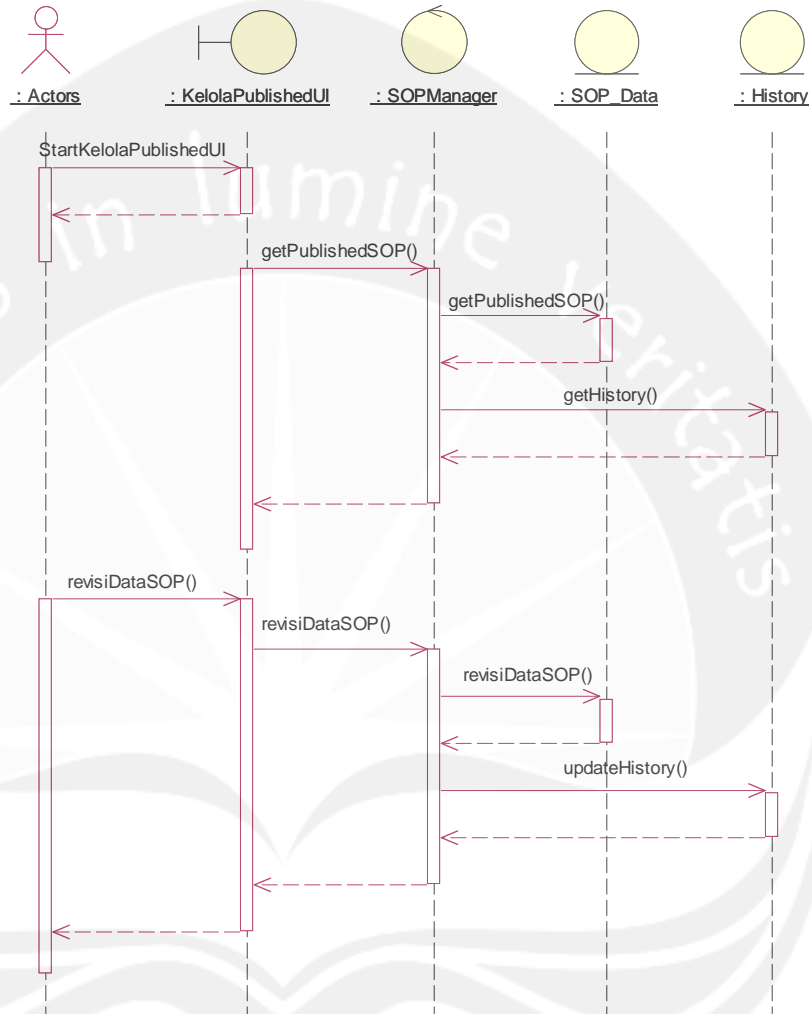
Gambar 2.2.24 Sequence Diagram - Kelola Published QSP - Tampil Published QSP

2.2.1.12.2 Update Status Published QSP



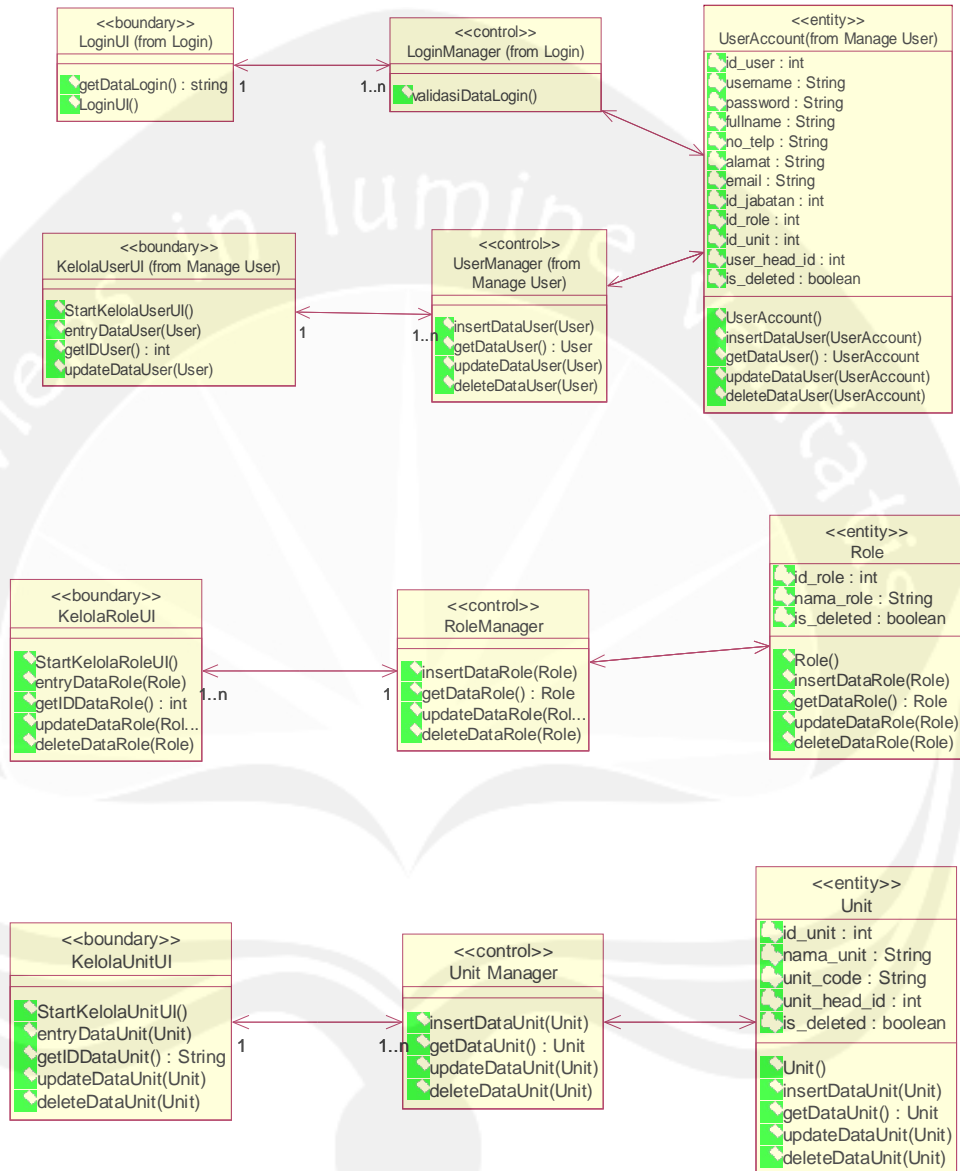
Gambar 2.2.25 Sequence Diagram - Kelola Published QSP - Update Status Published QSP

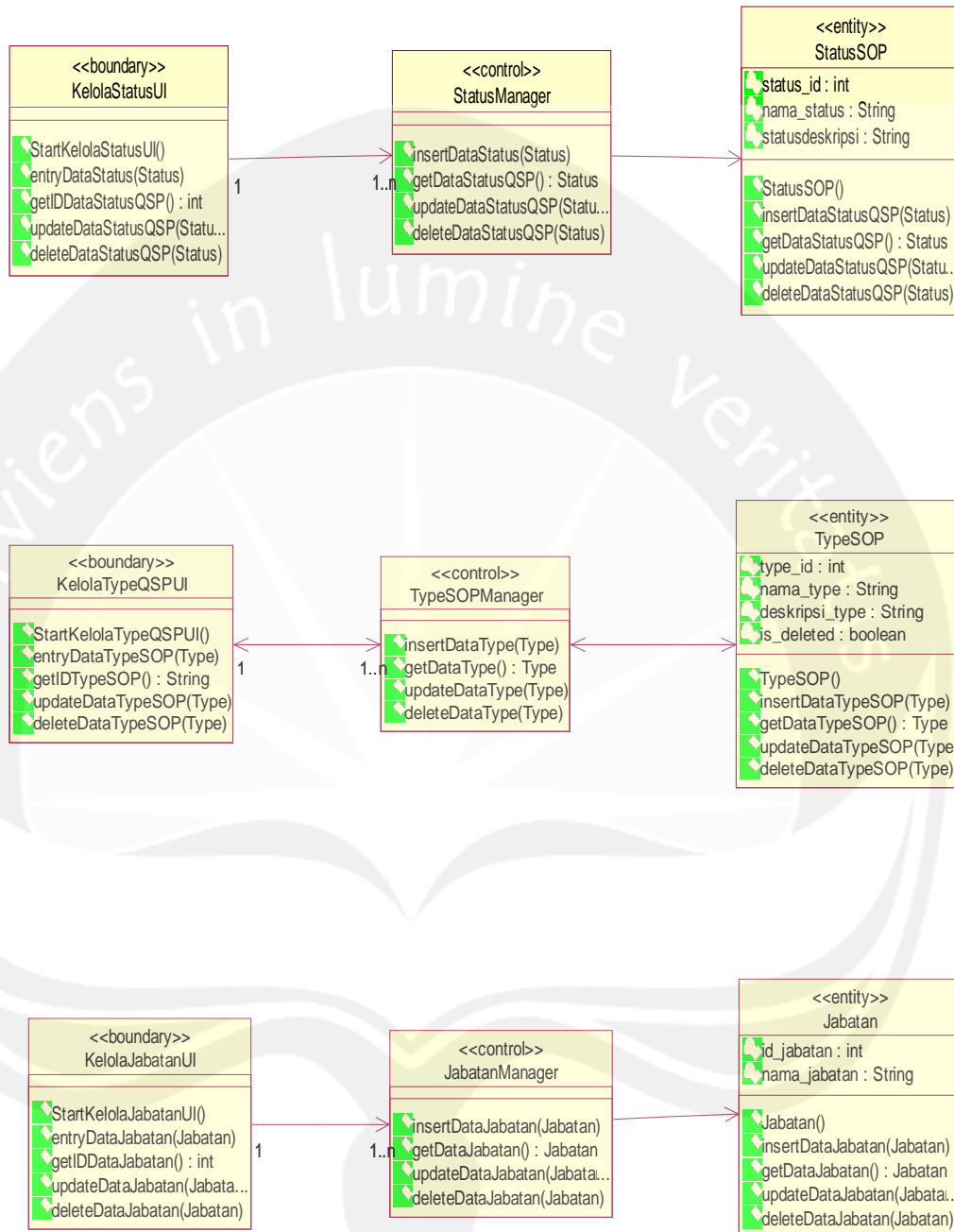
2.2.1.12.3 Revisi Published QSP

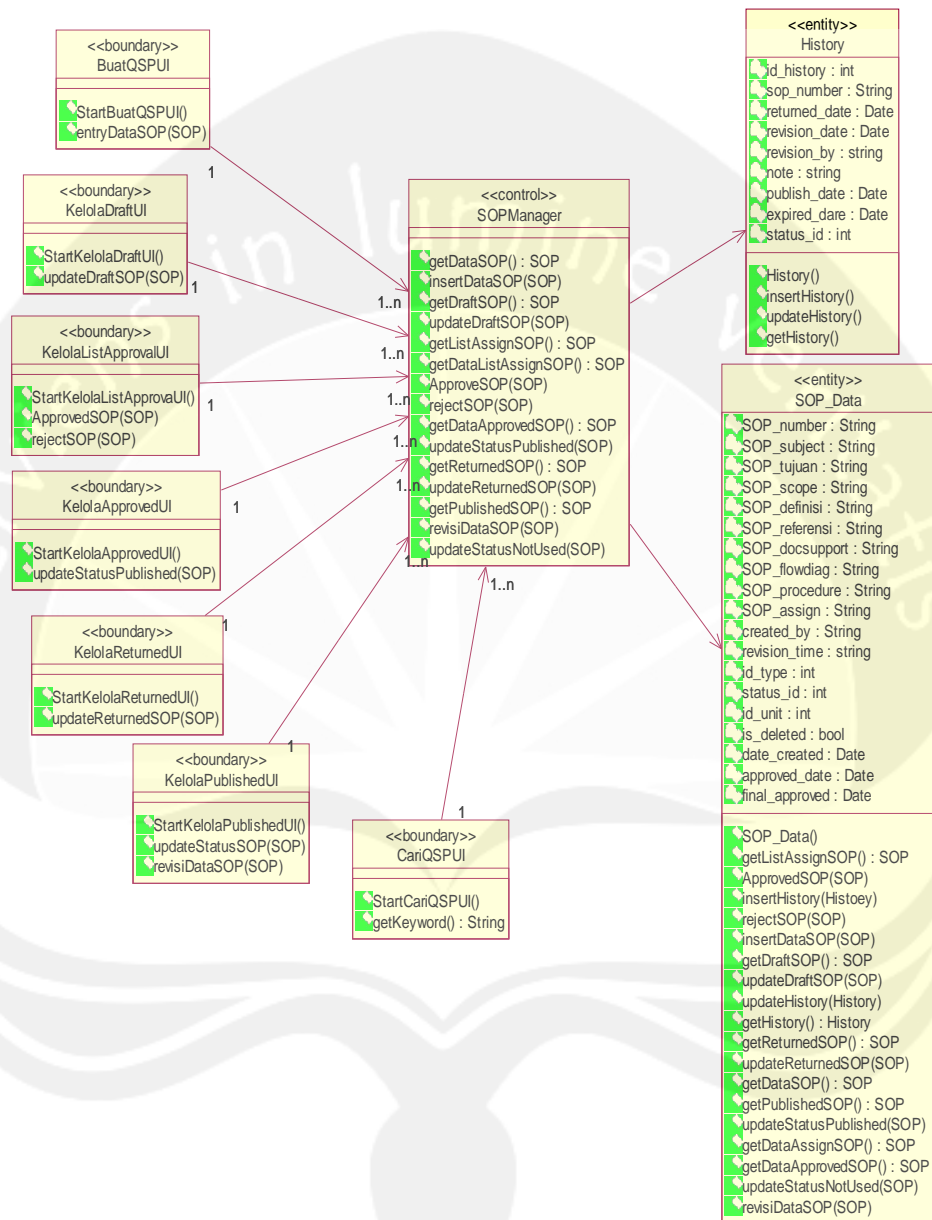


Gambar 2.2.26 Sequence Diagram - Kelola Published QSP -
Revisi Published QSP

2.2.2 Class Diagram







Gambar 2.2.27 Class Diagram Perangkat Lunak SIQ

2.2.3 Deskripsi Class

2.2.3.1 Specific Design Class LoginUI

LoginUI	<<boundary>>
+LoginUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getDataUserLogin() : String Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu login id dan password.	

2.2.3.2 Specific Design Class CariQSPUI

CariQSPUI	<<boundary>>
+ StartCariQSPUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getKeyword():String Operasi ini digunakan untuk mengambil data kata kunci inputan dari user melalui form yang akan digunakan untuk mencari data QSP, kata kunci bisa berupa nomor QSP, subject QSP, dll.	

2.2.3.3 Specific Design Class KelolaUserUI

KelolaUserUI	<<boundary>>
+ StartsKelolaUserUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+entryDataUser(User) Operasi ini digunakan untuk menambahkan data user beserta role-nya ke database	
+getIDUser() : int Operasi ini digunakan untuk mengambil data inputan dari form berupa ID user.	
+updateDataUser(User)	

KelolaUserUI	<<boundary>>
Operasi ini digunakan untuk mengubah data user yang dipilih dalam database.	

2.2.3.4 Specific Design Class KelolaUnitUI

KelolaUnitUI	<<boundary>>
<pre> + StartsKelolaUnitUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +entryDataUnit(Unit) Operasi ini digunakan untuk menginputkan data unit ke database. +getIDDataUnit() : int Operasi ini digunakan untuk mengambil data inputan dari form berupa ID unit. +updateDataUnit(Unit) Operasi ini digunakan untuk mengubah data unit yang sudah dipilih dari databse. +deleteDataUnit(Unit) Operasi ini digunakan untuk menghapus data unit yang dipilih dari dalam database. </pre>	

2.2.3.5 Specific Design Class KelolaTypeQSPUI

KelolaTypeQSPUI	<<boundary>>
<pre> + StartsKelolaTypeQSPUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +entryDataTypeSOP(Type) Operasi ini digunakan untuk menginputkan data type SOP ke database. +getIDTypeSOP():int Operasi ini digunakan untuk mengambil data inputan dari form berupa ID type. +updateDataTypeSOP(Type) </pre>	

KelolaTypeQSPUI	<<boundary>>
<p>Operasi ini digunakan untuk mengubah data type SOP yang sudah dipilih dari database.</p> <p>+deleteDataTypeSOP(Type)</p> <p>Operasi ini digunakan untuk menghapus data type SOP yang dipilih dari dalam database.</p>	

2.2.3.6 Specific Design Class KelolaStatusUI

KelolaStatusUI	<<boundary>>
<p>+ StartsKelolaStatusUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+entryDataStatus(Status)</p> <p>Operasi ini digunakan untuk menginputkan data status ke database.</p> <p>+getIDDataStatus():int</p> <p>Operasi ini digunakan untuk mengambil data inputan dari form berupa ID status.</p> <p>+updateDataStatusQSP(Status)</p> <p>Operasi ini digunakan untuk mengubah data status SQP yang sudah dipilih dari database.</p> <p>+deleteDataStatusQSP(Status)</p> <p>Operasi ini digunakan untuk menghapus data status QSP yang dipilih dari dalam database.</p>	

2.2.3.7 Specific Design Class KelolaJabatanUI

KelolaJabatanUI	<<boundary>>
<p>+ StartsKelolaJabatanUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+entryDataJabatan(Jabatan)</p> <p>Operasi ini digunakan untuk menginputkan data jabatan ke database.</p> <p>+getIDDataJabatan():int</p>	

KelolaJabatanUI	<<boundary>>
<p>Operasi ini digunakan untuk mengambil data inputan dari form berupa ID jabatan.</p> <p>+updateDataJabatan(Jabatan)</p> <p>Operasi ini digunakan untuk mengubah data jabatan yang sudah dipilih dari database.</p> <p>+deleteDataJabatan(Jabatan)</p> <p>Operasi ini digunakan untuk menghapus data jabatan yang dipilih dari dalam database.</p>	

2.2.3.8 Specific Design Class BuatQSPUI

BuatQSPUI	<<boundary>>
<p>+ StartsBuatQSPUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+entryDataSOP(SOP)</p> <p>Operasi ini digunakan untuk menginputkan isi data SOP kedalam database.</p>	

2.2.3.9 Specific Design Class KelolaDraftQSPUI

ManageDraftQSPUI	<<boundary>>
<p>+ StartKelolaDraftQSPUI()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+updateDraftSOP(SOP)</p> <p>Operasi ini digunakan untuk mengubah data SOP dalam database yang masih berupa draft.</p>	

2.2.3.10 Specific Design Class KelolaListApprovalUI

KelolaListApprovalUI	<<boundary>>
<pre>+ StartKelolaListApprovalUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +ApprovedSOP(SOP) Operasi ini digunakan untuk melakukan persetujuan atas SOP yang di-assign kepada approver dan mengganti statusnya menjadi approved. +rejectSOP() Operasi ini digunakan untuk mengubah status SOP menjadi rejected.</pre>	

2.2.3.11 Specific Design Class KelolaPublishedQSPUI

KelolaPublishedQSPUI	<<boundary>>
<pre>+ StartManagePublishedQSPUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +updateStatusNotUsed(SOP) Operasi ini digunakan untuk meng-update status SOP yang telah dipublish jika sudah tidak berlaku lagi +revisiDataSOP(SOP) Operasi ini digunakan untuk merevisi data SOP yang sudah dipublish jika sekiranya perlu diperbaiki</pre>	

2.2.3.12 Specific Design Class KelolaReturnedUI

KelolaReturnedUI	<<boundary>>
<pre>+ StartKelolaReturnedUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +updateReturnedSOP(SOP)</pre>	

KelolaReturnedUI	<<boundary>>
Operasi ini digunakan untuk mengubah data SOP yang dikembalikan oleh approver.	

2.2.3.13 Specific Design Class LoginManager

LoginManager	<<control>>
+validasiDataLogin() : boolean Operasi ini digunakan untuk mengecek apakah data login sudah benar.	

2.2.3.14 Specific Design Class UserManager

UserManager	<<control>>
+insertDataUser(User) Operasi ini digunakan untuk menyimpan data user kedalam database. +getDataUser() : User Operasi ini mengambil untuk mengambil data user dari database. +updateDataUser(User) Operasi ini digunakan untuk mengubah data user yang dipilih dari database. +deleteDataUser(User) Operasi ini digunakan untuk menghapus data user di database.	

2.2.3.15 Specific Design Class TypeSOPManager

TypeSOPManager	<<control>>
+insertDataType(Type) Operasi ini digunakan untuk menyimpan data type SOP kedalam database. +getDataType() : Type Operasi ini mengambil untuk mengambil data type SOP dari database. +updateDataType(Type)	

TypeSOPManager	<<control>>
<p>Operasi ini digunakan untuk mengubah data type SOP yang dipilih dari database.</p> <p>+deleteDataType(Type)</p> <p>Operasi ini digunakan untuk menghapus data type SOP di database.</p>	

2.2.3.16 Specific Design Class StatusManager

StatusManager	<<control>>
<p>+insertDataStatus(Status)</p> <p>Operasi ini digunakan untuk menyimpan data status kedalam database.</p> <p>+getDataStatusQSP() : Status</p> <p>Operasi ini mengambil untuk mengambil data status dari database.</p> <p>+updateDataStatusQSP(Type)</p> <p>Operasi ini digunakan untuk mengubah data status QSP yang dipilih dari database.</p> <p>+deleteDataStatusQSP(Type)</p> <p>Operasi ini digunakan untuk menghapus data status QSP di database.</p>	

2.2.3.17 Specific Design Class JabatanManager

JabatanManager	<<control>>
<p>+insertDataJabatan(Jabatan)</p> <p>Operasi ini digunakan untuk menyimpan data jabatan kedalam database.</p> <p>+getDataJabatan() : Jabatan</p> <p>Operasi ini mengambil untuk mengambil data jabatan dari database.</p> <p>+updateDataJabatan(Jabatan)</p> <p>Operasi ini digunakan untuk mengubah data jabatan yang dipilih dari database.</p> <p>+deleteDataJabatan(Jabatan)</p> <p>Operasi ini digunakan untuk menghapus data jabatan di database.</p>	

2.2.3.18 Specific Design Class UnitManager

UnitManager	<<control>>
<pre>+insertDataUnit(Unit) Operasi ini digunakan untuk menyimpan data unit kedalam database. +getDataUnit() : Unit Operasi ini mengambil untuk mengambil data unit dari database. +updateDataUnit(Unit) Operasi ini digunakan untuk mengubah data unit yang dipilih dari database. +deleteDataUnit(Unit) Operasi ini digunakan untuk menghapus data unit di database.</pre>	

2.2.3.19 Specific Design Class SOPManager

SOPManager	<<control>>
<pre>+getDataSOP(): SOP Operasi ini digunakan untuk mengambil data SOP dari dalam database. +insertDataSOP(SOP) Operasi ini digunakan untuk menyimpan data SOP kedalam database. +getDraftSOP():SOP Operasi ini digunakan untuk mengambil data SOP yang masih berstatus draft dari dalam database. +updateDraftSOP(SOP) Operasi ini digunakan untuk meng-update data SOP yang berupa draft. +getListAssignSOP() : SOP Operasi ini digunakan untuk mengambil data SOP yang dikirim kepada approver untuk disetujui (di-assign). +getDataListAssignSOP() : SOP Operasi ini digunakan untuk mengambil dan menampilkan data SOP yang diassign kepada approver +ApproveSOP(SOP) Operasi ini digunakan untuk menyetujui dokumen SOP oleh approver. +rejectSOP(SOP)</pre>	

SOPManager	<<control>>
<p>Operasi ini digunakan untuk menolak dokumen SOP yang dikirim kepada approver dan akan dikembalikan kepada pembuatnya.</p> <p>+getDataApprovedSOP() : SOP</p> <p>Operasi ini digunakan untuk mengambil data SOP yang sudah disetujui.</p> <p>+updateStatusPublished(SOP)</p> <p>Operasi ini digunakan untuk meng-update status SOP yang sudah disetujui menjadi published saat akan dipublish.</p> <p>+getReturnedSOP() : SOP</p> <p>Operasi ini digunakan untuk mengambil data SOP yang di-return / di-reject oleh approver.</p> <p>+updateReturnedSOP(SOP)</p> <p>Operasi ini digunakan untuk mengubah data SOP yang dikembalikan oleh approver kepada QSP Maker.</p> <p>+getPublishedSOP() : SOP</p> <p>Operasi ini digunakan untuk mengambil data SOP yang telah dipublish</p> <p>+updateStatusNotUsed(SOP)</p> <p>Operasi ini digunakan untuk meng-update status SOP yang telah dipublish jika sudah tidak berlaku lagi</p> <p>+revisiDataSOP(SOP)</p> <p>Operasi ini digunakan untuk merevisi data SOP yang sudah dipublish jika sekiranya perlu diperbaiki</p>	

2.2.3.20 Specific Design Class UserAcc

UserAccount	<<entity>>
<p>-id_user : int Atribut ini menyimpan id user</p> <p>-username: String Atribut ini menyimpan username</p> <p>-password : String Atribut ini menyimpan password</p> <p>-fullname : String Atribut ini menyimpan fullname</p>	

UserAccount	<<entity>>
<p>-no_telp : String Atribut ini menyimpan no telepon</p> <p>-alamat : String Atribut ini menyimpan alamat</p> <p>-email : String Atribut ini menyimpan email</p> <p>-id_jabatan : int Atribut ini menyimpan id jabatan</p> <p>-id_role : int Atribut ini menyimpan id role</p> <p>-id_unit : int Atribut ini menyimpan id unit</p> <p>-user_head_id : int Atribut ini menyimpan id atasan user</p> <p>-is_deleted : Boolean Atribut ini menyimpan status is deleted data user</p>	
<p>+UserAccount() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+insertDataUser(UserAccount) Operasi ini digunakan untuk memasukkan data user account ke dalam database.</p> <p>+getDataUser() : UserAccount Operasi ini digunakan untuk mengambil data user account dari database.</p> <p>+updateDataUser(UserAccount) Operasi ini digunakan untuk meng-update data user account dalam database.</p> <p>+deleteDataUser(UserAccount) Operasi ini digunakan untuk menghapus data user account dari database.</p>	

2.2.3.21 Specific Design Class Unit

Unit	<<entity>>
<p>-id_unit : int Atribut ini menyimpan id unit</p> <p>-nama_unit : String Atribut ini menyimpan nama unit</p> <p>-unit_code : String Atribut ini menyimpan kode nama unit</p> <p>-unit_head_id : int Atribut ini menyimpan id kepala unit</p> <p>-is_deleted : Boolean Atribut ini menyimpan status is deleted data unit</p>	
<p>+Unit() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+insertDataUnit(Unit) Operasi ini digunakan untuk menyimpan data unit ke database.</p> <p>+getDataUnit() : Unit Operasi ini digunakan untuk mengambil data unit dari database.</p> <p>+updateDataUnit(Unit) Operasi ini digunakan untuk mengupdate data unit dalam database.</p> <p>+deleteDataUnit(Unit) Operasi ini digunakan untuk menghapus data unit dari database.</p>	

2.2.3.22 Specific Design Class TypeSOP

TypeSOP	<<entity>>
<p>-type_id : int Atribut ini menyimpan id type</p> <p>-nama_type : String Atribut ini menyimpan nama type</p> <p>-deskripsi_type : String Atribut ini menyimpan deskripsi type</p>	
<p>+TypeSOP() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p>	

TypeSOP	<<entity>>
<pre>+insertDataTypeSOP(Type) Operasi ini digunakan untuk menyimpan data type SOP ke database. +getDataTypeSOP() : Type Operasi ini digunakan untuk mengambil data type SOP dari database. +updateDataTypeSOP(Type) Operasi ini digunakan untuk mengupdate data type SOP dalam database. +deleteDataTypeSOP(Type) Operasi ini digunakan untuk menghapus data type SOP dari database.</pre>	

2.2.3.23 Specific Design Class Status

StatusSOP	<<entity>>
<pre>-status_id : int Atribut ini menyimpan id status -nama_status : String Atribut ini menyimpan nama status -statusdeskripsi : String Atribut ini menyimpan deskripsi status</pre>	
<pre>+StatusSOP() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataStatus(Status) Operasi ini digunakan untuk menyimpan data status QSP ke database. +getDataStatusQSP() : Status Operasi ini digunakan untuk mengambil data status QSP dari database. +updateDataStatusQSP(Status) Operasi ini digunakan untuk mengupdate data status dalam database. +deleteDataStatusQSP(Status) Operasi ini digunakan untuk menghapus data status SOP dari</pre>	

database.

2.2.3.24 Specific Design Class Jabatan

Jabatan	<<entity>>
<pre>-status_id : int Atribut ini menyimpan id status -nama_status : String Atribut ini menyimpan nama status -statusdeskripsi : String Atribut ini menyimpan deskripsi status</pre>	
<pre>+Jabatan() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataJabatan(Jabatan) Operasi ini digunakan untuk menyimpan data jabatan ke database. +getDataJabatan() : Jabatan Operasi ini digunakan untuk mengambil data jabatan dari database. +updateDataJabatan(Jabatan) Operasi ini digunakan untuk mengupdate data jabatan dalam database. +deleteDataJabatan(Jabatan) Operasi ini digunakan untuk menghapus data jabatan dari database.</pre>	

2.2.3.25 Specific Design Class SOP_Data

SOP_Data	<<entity>>
<pre>-SOP_number : String Atribut ini digunakan untuk menyimpan nomor dokumen SOP -SOP_subject : String Atribut ini digunakan untuk menyimpan subject dari SOP -SOP_tujuan : String Atribut ini digunakan untuk menyimpan tujuan SOP -SOP_scope : String Atribut ini digunakan untuk menyimpan scope SOP -SOP_definisi : String Atribut ini digunakan untuk menyimpan definisi SOP -SOP_referensi : String</pre>	

SOP_Data	<<entity>>
<p>Atribut ini digunakan untuk menyimpan referensi SOP</p> <p>-SOP_docsupport : String</p> <p>Atribut ini digunakan untuk menyimpan dokumen penunjang SOP</p> <p>-SOP_flowdiag : String</p> <p>Atribut ini digunakan untuk menyimpan gambar alur proses SOP</p> <p>-SOP_procedure : String</p> <p>Atribut ini digunakan untuk menyimpan gambar prosedur SOP</p> <p>-SOP_assign : String</p> <p>Atribut ini digunakan untuk menyimpan nama approver SOP</p> <p>-created_by : String</p> <p>Atribut ini digunakan untuk menyimpan data waktu SOP dibuat</p> <p>-revision_time : String</p> <p>Atribut ini digunakan untuk menyimpan nilai banyaknya SOP pernah direvisi</p> <p>-id_type : int</p> <p>Atribut ini digunakan untuk menyimpan id type SOP</p> <p>-status_id : int</p> <p>Atribut ini digunakan untuk menyimpan id status SOP</p> <p>-id_unit : int</p> <p>Atribut ini digunakan untuk menyimpan id unit</p> <p>-is_deleted : bool</p> <p>Atribut ini untuk menyimpan status hapus data SOP</p> <p>-date_created : Date</p> <p>Atribut ini untuk menyimpan tanggal dibuatnya SOP</p> <p>-approved_date : Date</p> <p>Atribut ini digunakan untuk menyimpan tanggal disetujuinya SOP oleh Approver</p> <p>-final_approved : Date</p> <p>Atribut ini digunakan untuk menyimpan tanggal disetujuinya SOP oleh Managenet Representative</p>	
<p>+ SOP_Data()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+getListAssignSOP() : SOP</p> <p>Operasi ini digunakan oleh approver untuk mengambil data SOP yang di-assign</p>	

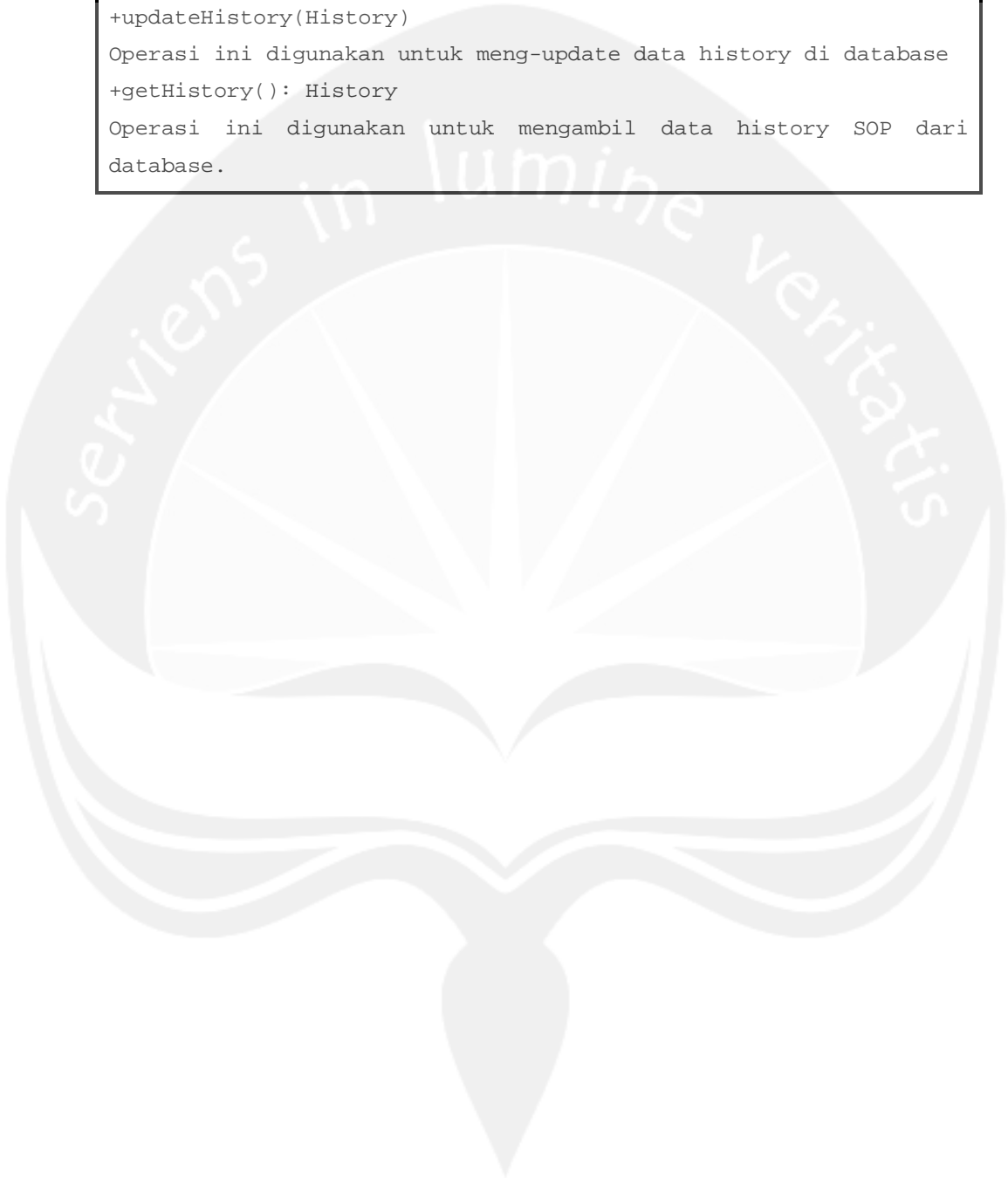
SOP_Data	<<entity>>
<p>+ApprovedSOP(SOP) Operasi ini digunakan untuk melakukan approval dokumen QSP dengan mengubah status SOP</p> <p>+insertHistory(History) Operasi ini digunakan untuk memasukkan history dokumen SOP</p> <p>+rejectQSP(SOP) Operasi ini digunakan untuk mereject dokumen SOP dengan mengubah status SOP</p> <p>+insertDataSOP(SOP) Operasi ini digunakan untuk memasukkan data SOP</p> <p>+getDraftSOP(): SOP Operasi ini digunakan untuk mengambil data SOP yang berstatus draft SOP</p> <p>+updateDraftSOP(SOP) Operasi ini digunakan untuk meng-update data SOP yang masih berupa draft</p> <p>+updateHistory(History) Operasi ini digunakan untuk meng-update history dokumen SOP</p> <p>+getHistory(): History Operasi ini digunakan untuk mengambil history dokumen SOP</p> <p>+getReturnedSOP(): SOP Operasi ini digunakan untuk mengambil data SOP yang direject oleh approver</p> <p>+updateReturnedSOP(SOP) Operasi ini digunakan untuk meng-update data SOP yang direject oleh approver</p> <p>+getDataSOP(): SOP Operasi ini digunakan untuk mengambil data dokumen SOP dari database</p> <p>+getPublishedSOP(): SOP Operasi ini digunakan untuk mengambil data SOP yang telah dipublish</p> <p>+updateStatusPublished(SOP) Operasi ini digunakan untuk meng-update status SOP yang sudah disetujui menjadi published saat akan dipublish.</p> <p>+updateStatusNotUsed(SOP)</p>	

SOP_Data	<<entity>>
<p>Operasi ini digunakan untuk meng-update status SOP yang telah dipublish jika sudah tidak berlaku lagi</p> <p>+getDataListAssignSOP() : SOP</p> <p>Operasi ini digunakan untuk mengambil dan menampilkan data SOP yang diassign kepada approver</p> <p>+getDataApprovedSOP() : SOP</p> <p>Operasi ini digunakan untuk mengambil data SOP yang sudah disetujui.</p>	

2.2.3.26 Specific Design Class History

History	<<entity>>
<p>-id_history : int Atribut ini digunakan untuk menyimpan id history</p> <p>-SOP_number : String Atribut ini digunakan untuk menyimpan nomor SOP</p> <p>-published_date : Date Atribut ini digunakan untuk menyimpan tanggal dipublishnya SOP</p> <p>-returned_date : Date Atribut ini digunakan untuk menyimpan tanggal dikembalikannya SOP</p> <p>-revision_date : Date Atribut ini digunakan untuk menyimpan tanggal revisi SOP</p> <p>-revision_by : String Atribut ini digunakan untuk menyimpan nama perevisi SOP</p> <p>-note : String Atribut ini digunakan untuk menyimpan note revisi SOP</p> <p>-expired_date : Date Atribut ini digunakan untuk menyimpan tanggal tidak berlakunya SOP</p> <p>-status_id : int Atribut ini digunakan untuk menyimpan id status dari history SOP</p>	
<p>+History() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+insertHistory(History) Operasi ini digunakan untuk menyimpan data history SOP ke</p>	

History	<<entity>>
<p>database.</p> <p>+updateHistory(History) Operasi ini digunakan untuk meng-update data history di database</p> <p>+getHistory(): History Operasi ini digunakan untuk mengambil data history SOP dari database.</p>	



3 Perancangan Data

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas Data Role User

Nama	Tipe	Panjang	Keterangan
id_role	Integer	-	Id role, primary key, nilainya digenerate otomatis
nama_role	Varchar	50	Nama role

3.1.2 Deskripsi Entitas Data UserAcc

Nama	Tipe	Panjang	Keterangan
id_user	Integer	-	Id user, primary key
id_role	Integer	-	Id role, foreign key
id_unit	Integer	-	Id unit, foreign key
id_jabatan	Integer	-	Id jabatan, foreign key
user_head_id	Integer	-	Id user atasan, foreign key dari table UserAcc
username	Varchar	50	Username milik user, untuk login
password	Varchar	100	Password user
fullname	Varchar	100	Nam alengkap dari user
no_telp	Varchar	20	No telepon user
alamat	Varchar	100	Alamat user
email	Varchar	50	Email user
is_deleted	Boolean	-	Status is deleted untu data user

3.1.3 Deskripsi Entitas Data Type

Nama	Tipe	Panjang	Keterangan
type_id	Integer	-	Id type SOP, primary key
nama_type	Varchar	100	Nama type SOP
deskripsi_type	Varchar	100	Deskripsi atau keterangan type SOP

3.1.4 Deskripsi Entitas Data Status QSP

Nama	Tipe	Panjang	Keterangan
status_id	Integer	-	Id status SOP, primary key
nama_status	Varchar	50	Nama status SOP
statusdeskripsi	Varchar	100	Deskripsi atau keterangan status SOP

3.1.5 Deskripsi Entitas Data Unit

Nama	Tipe	Panjang	Keterangan
id_unit	Varchar	20	Id unit, primary key
unit_head_id	Varchar	20	Id kepala unit, foreign key, reference ke table Unit
nama_unit	Varchar	100	Nama unit
unit_code	Varchar	20	Kode untuk setiap nama unit
is_deleted	Boolean	-	Status is deleted untuk data unit

3.1.6 Deskripsi Entitas Data SOP_DATA

Nama	Tipe	Panjang	Keterangan
SOP_number	Varchar	50	Nomor dokumen QSP, primary key, nilainya degenerate

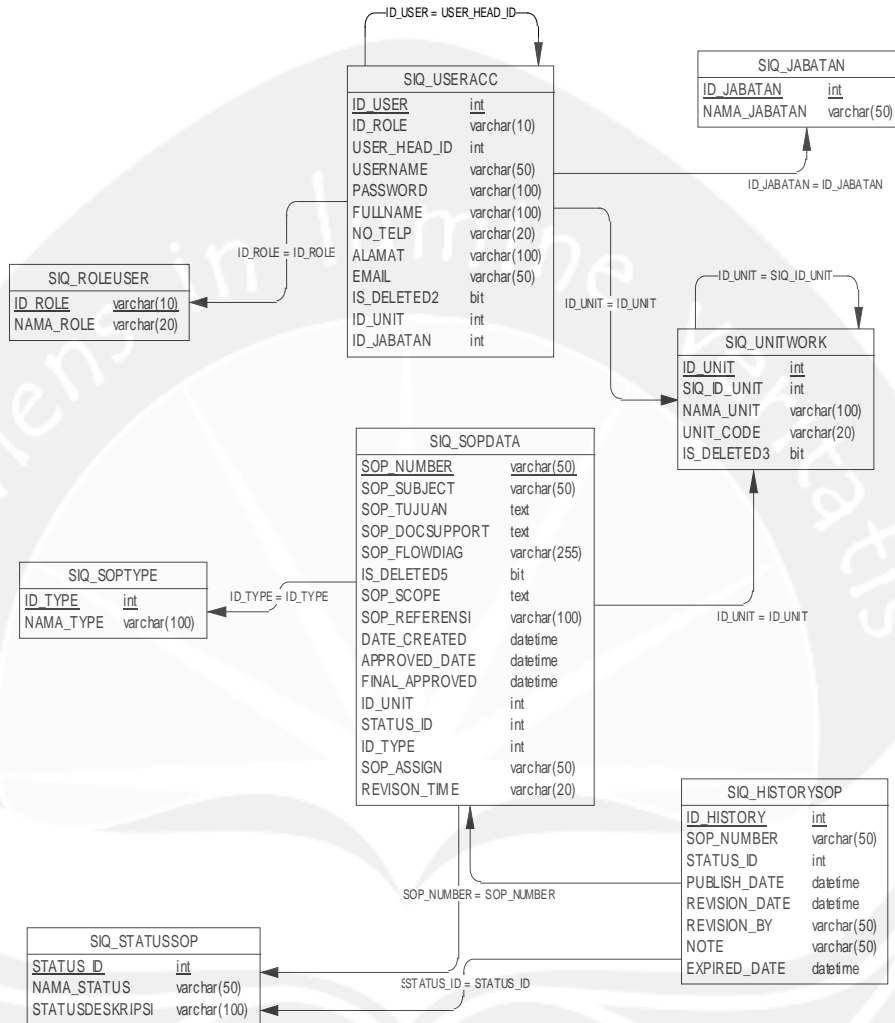
Nama	Tipe	Panjang	Keterangan
			otomatis
id_type	Integer	-	Id type QSP, foreign key
id_unit	Varchar	20	Id unit tempat QSP dibuat, foreign key
SOP_subject	Varchar	50	Subject dari dokumen QSP
SOP_tujuan	Text	-	Tujuan dari dokumen QSP
SOP_scope	Text	-	Lingkup dari dokumen QSP
SOP_definisi	Text	-	Definisi dari dokumen QSP
SOP_referensi	Text	-	Referensi dari dokumen QSP
SOP_docsupport	Text	-	Dokumen penunjang dari QSP
SOP_flowdiag	Varchar	255	Nama file dari gambar alur proses QSP
SOP_procedure	Varchar	255	Nama file dari prosedur QSP
SOP_assign	Varchar	50	Nama approver dari QSP
created_by	Varchar	50	Nama pembuat QSP
revision_time	Varchar	20	Berapa kali QSP pernah direvisi
is_deleted	Boolean	-	Status hapus dari dokumen QSP
date_created	Date	-	Tanggal dibuatnya dokumen QSP
approved_date	Date	-	Tanggal disetujuinya dokumen QSP tingkat approver
final_approved	Date	-	Tanggal

Nama	Tipe	Panjang	Keterangan
			disetujuinya dokumen QSP tingkat management representative

3.1.7 Deskripsi Entitas Data History

Nama	Tipe	Panjang	Keterangan
id_history	Integer	-	Id history QSP, primary key
status_id	Integer	-	Status history dokumen QSP, foreign key
sop_number	String	50	Nomor dokumen QSP, foreign key
returned_date	Date	-	Tanggal QSP dikembalikan
revision_date	Date	-	Tanggal QSP direvisi
revision_by	Varchar	50	Nama perevisi QSP
revision_time	Integer	-	Status berapa kali QSP direvisi
note	Varchar	100	Note revisi untuk QSP
publish_date	Date	-	Tanggal QSP dipublish
expired_date	Date	-	Tanggal tidak berlakunya QSP

3.2 Physical data model



Gambar 3.2.1 Physical Data Model

3. Perancangan Antar Muka

3.1 Login

HEADER	
Welcome to Quality System Procedure Universitas Atma Jaya Yogyakarta	Username <input type="text"/> Password <input type="password"/> <input type="button" value="Login"/>
FOOTER	

Gambar 3.1.1 Rancangan antarmuka login

Pada antarmuka ini, user diminta untuk memasukkan username dan passwordnya untuk menentukan hak akses sesuai role. Sebelumnya data sudah tersimpan dalam data base.

3.2 Main Menu Administrator

HEADER	
Menu Admin Page <ul style="list-style-type: none">• Manage Role• Manage User• Manage• etc	Welcome
FOOTER	

Gambar 3.1.2 Rancangan main menu administrator

Pada antarmuka ini, user, dalam hal ini administrator, akan disuguhkan pilihan menu pengelolaan sesuai hak akses administrator yang ada

dibagian kiri halaman web. Administrator cukup mengklik salah satu link menu yang ingin dipilih.

3.3 Cari QSP

HEADER			
<ul style="list-style-type: none"> Cari QSP 	Cari QSP		
	Cari berdasarkan	<input type="text"/> v	
	Input Keyword	<input type="text"/>	
	Cari		
	Data SOP	Edit	Delete
		Edit	Delete
	Edit	Delete	
	Edit	Delete	
	Edit	Delete	
Tampil Data			
FOOTER			

Gambar 3.1.3 Antarmuka Cari QSP

Antarmuka ini digunakan oleh semua user untuk mencari data QSP. User dapat mencari data QSP berdasarkan nomor QSP, status, dll. Kemudian user memasukkan kata kunci yang diinginkan, dan klik button "Cari" maka akan muncul data QSP yang diinginkan, untuk melihat secara lebih lengkap tekan button "Tampil Data".

HEADER	
<ul style="list-style-type: none"> Search SOP 	Data SOP
	INFO DOKUMEN SOP
FOOTER	

Gambar 3.1.4 Tampilan informasi dokumen QSP

Tampilan dokumen QSP setelah user meng-klik button "Tampil Data" untuk melihat isi dokumen QSP keseluruhan.

3.4 Kelola Akun User

Gambar 3.1.5 Antarmuka Manage User Account

Antarmuka ini digunakan oleh user dalam hal ini administrator untuk mengelola account untuk user. Actor dapat menambahkan data account user dengan mengisi data pada textbox yang sudah tersedia, kemudian klik button "Simpan" untuk menyimpan data. Untuk melakukan perubahan data klik pada link "Edit", sedangkan untuk menghapus data klik pada link "Delete", akan muncul konfirmasi penghapusan data, jika ya tekan yes. Selain menambah, mengubah dan menghapus, actor juga dapat melakukan pencarian data berdasar kata kunci yang diinputkan, lalu tekan button

"Cari", data yang ditampilkan akan difilter berdasarkan pilihan pada bagian "Search by".

3.5 Kelola Type QSP

Gambar 3.1.6 Antarmuka Kelola Type QSP

Antarmuka ini digunakan oleh user dalam hal ini administrator untuk mengelola type QSP. Actor dapat menambahkan data type QSP dengan mengisi data pada textbox yang sudah tersedia, kemudian klik button "Simpan" untuk menyimpan data. Untuk melakukan perubahan data klik pada link "Edit", sedangkan untuk menghapus data klik pada link "Delete", akan muncul konfirmasi penghapusan data, jika ya tekan yes. Selain menambah, mengubah dan menghapus, actor juga dapat melakukan pencarian data berdasar kata kunci yang diinputkan, lalu tekan button "Cari".

3.6 Kelola Unit Group

Gambar 3.1.7 Antarmuka Kelola Unit Group

Antarmuka ini digunakan oleh user dalam hal ini administrator untuk mengelola data unit. Actor dapat menambahkan data unit dengan mengisi data pada textbox yang sudah tersedia, kemudian klik button "Simpan" untuk menyimpan data. Untuk melakukan perubahan data klik pada link "Edit", sedangkan untuk menghapus data klik pada link "Delete", akan muncul konfirmasi penghapusan data, jika ya tekan yes. Selain menambah, mengubah dan menghapus, actor juga dapat melakukan pencarian data berdasar kata kunci yang diinputkan, lalu tekan button "Cari", data yang ditampilkan akan difilter berdasarkan pilihan pada bagian "Search by".

3.7 Buat QSP

HEADER																											
Buat QSP	<table><tr><td>Buat QSP</td><td></td></tr><tr><td>SOP number</td><td><input type="text"/></td></tr><tr><td>Subjetc</td><td><input type="text"/></td></tr><tr><td>Tujuan</td><td><input type="text"/></td></tr><tr><td>Procedure</td><td><input type="text"/> <input type="button" value="Browse"/></td></tr><tr><td>Alur Proses</td><td><input type="text"/> <input type="button" value="Browse"/></td></tr><tr><td>Approver</td><td><input type="text"/> ▼</td></tr><tr><td>Referensi</td><td><input type="text"/></td></tr><tr><td>Lingkup</td><td><input type="text"/></td></tr><tr><td>Created by</td><td><input type="text"/></td></tr><tr><td>Type SOP</td><td><input type="text"/> ▼</td></tr><tr><td>Unit SOP</td><td><input type="text"/></td></tr><tr><td colspan="2"><input type="button" value="Simpan sebagai draft"/> <input type="button" value="Kirim ke Approver"/></td></tr></table>	Buat QSP		SOP number	<input type="text"/>	Subjetc	<input type="text"/>	Tujuan	<input type="text"/>	Procedure	<input type="text"/> <input type="button" value="Browse"/>	Alur Proses	<input type="text"/> <input type="button" value="Browse"/>	Approver	<input type="text"/> ▼	Referensi	<input type="text"/>	Lingkup	<input type="text"/>	Created by	<input type="text"/>	Type SOP	<input type="text"/> ▼	Unit SOP	<input type="text"/>	<input type="button" value="Simpan sebagai draft"/> <input type="button" value="Kirim ke Approver"/>	
Buat QSP																											
SOP number	<input type="text"/>																										
Subjetc	<input type="text"/>																										
Tujuan	<input type="text"/>																										
Procedure	<input type="text"/> <input type="button" value="Browse"/>																										
Alur Proses	<input type="text"/> <input type="button" value="Browse"/>																										
Approver	<input type="text"/> ▼																										
Referensi	<input type="text"/>																										
Lingkup	<input type="text"/>																										
Created by	<input type="text"/>																										
Type SOP	<input type="text"/> ▼																										
Unit SOP	<input type="text"/>																										
<input type="button" value="Simpan sebagai draft"/> <input type="button" value="Kirim ke Approver"/>																											
FOOTER																											

Gambar 3.1.8 Buat QSP

Antarmuka ini digunakan oleh actor untuk menambahkan data QSP yang baru. Actor mengisi field-field yang sudah disediakan seperti QSP number, subject, tujuan, procedure, alur proses, approver, referensi, lingkup, created by, type QSP dan unit QSP. Untuk bagian procedure dan alur proses, actor mengisi dengan cara meng-attach file prosedur, klik pada button "Browse" lalu pilih file yang diinginkan. Setelah semua selesai diisi klik button "Simpan sebagai draft" untuk menyimpan dahulu sebagai draft atau "Kirim ke Approver" jika sudah selesai membuat.

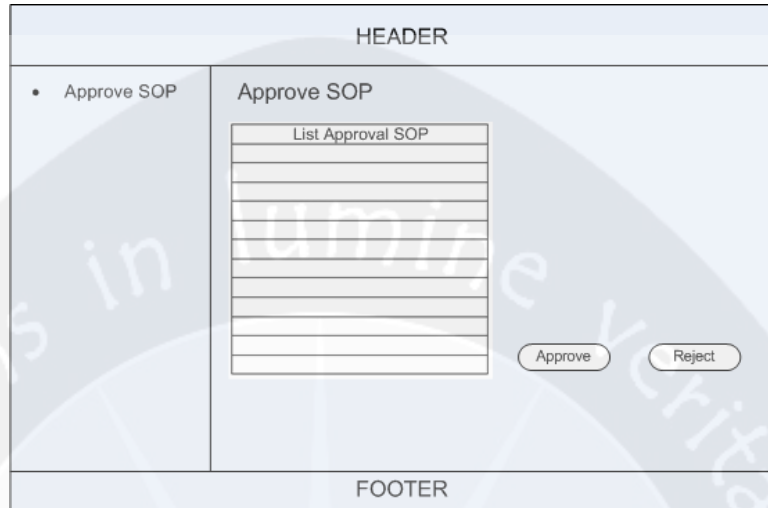
3.8 Kelola Draft QSP

Draft SOP	
	Edit
	Edit
	Edit
	Edit
	Edit

Gambar 3.1.9 Antarmuka Kelola Draft QSP

Antarmuka ini digunakan actor untuk melakukan pengelolaan QSP yang masih berstatus sebagai draft QSP. Actor dapat mencari data dengan memasukkan kata kunci dan pemfilteran data berdasarkan yang diinginkan. Kemudian jika ingin melakukan update, klik pada link "Edit. Untuk melihat data secara keseluruhan tekan button "View Data".

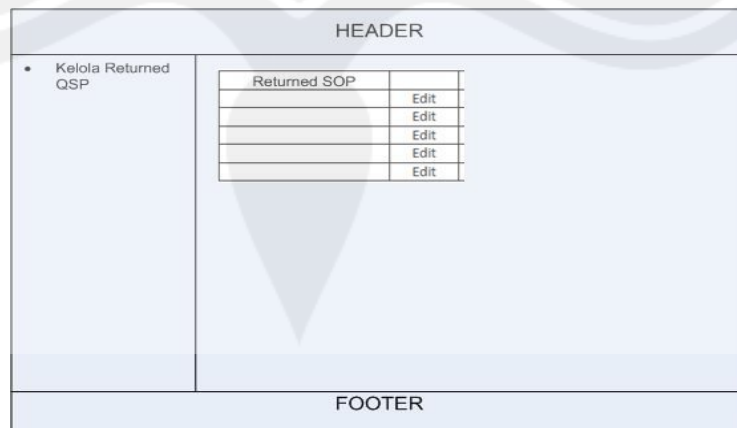
3.9 Kelola List Approval QSP



Gambar 3.1.10 Kelola List Approval

Antarmuka ini digunakan oleh approver untuk mengelola data-data QSP yang akan diapprove. Actor dapat melakukan penolakan ataupun persetujuan untuk dokumen QSP. Untuk melakukan persetujuan actor klik pada button "Approve", sedangkan untuk melakukan penolakan klik pada button "Reject".

3.10 Kelola Returned QSP



Gambar 3.1.11 Kelola Returned QSP

Antarmuka ini digunakan oleh actor untuk mengelola QSP yang dikembalikan oleh approver. Actor dapat mengubah dan menghapus data QSP yang direturned oleh approver. Untuk mengubah data klik pada link "Edit" maka actor akan diarahkan ke halaman lain untuk melakukan perubahan data pada bagian-bagian yang diinginkan. Untuk melakukan penghapusan data, klik pada link "Delete".

3.11 Kelola Published QSP

Published QSP		
	Update Status	Revisi
	Update Status	Revisi
	Update Status	Revisi

Gambar 3.1.12 Kelola Published QSP

Antarmuka ini digunakan untuk mengelola QSP yang telah dipublished. Actor dapat melihat data dan melakukan peng-update-an status untuk QSP yang telah dipublish. Untuk melihat data klik pada button "View Data", sedangkan link "Update Status" digunakan untuk mengubah status QSP yang sudah tidak berlaku lagi. Untuk melakukan revisi QSP maka klik pada link "Revisi"