

BAB VI

KESIMPULAN DAN SARAN

VI.1 Kesimpulan

Berdasarkan uraian dan pengujian yang telah dilakukan, kesimpulan dari pembangunan sistem ini adalah sebagai berikut:

1. Sistem Informasi Desa dan Kawasan untuk Perencanaan Pembangunan Desa Berbasis Web telah berhasil dibangun dan dapat menjadi alat pendukung kegiatan pemerintah desa.
2. Sistem Informasi Desa dan Kawasan untuk Perencanaan Pembangunan Desa Berbasis Web telah berhasil dibangun dan berhasil dijalankan untuk mendukung perencanaan pembangunan desa. Fungsi yang dikelola pada aplikasi ini yakni pengelolaan Rencana Jangka Menengah Daerah (RPJMD), Rencana Jangka Menengah Desa (RPJMDes), Rencana Kerja Pemerintah (RKPDDes), Rencana Anggaran Biaya Desa (RABDes), Surat Permintaan Pembayaran (SPP), Laporan Penanggungjawaban (LPJ), dan *reporting*.

VI.2 Saran

Walaupun pembangunan sistem sudah berjalan dengan baik, namun masih terdapat kekurangan yang dilakukan penulis. Maka dari itu, penulis memberikan saran:

Aplikasi ini diharapkan dapat dikembangkan lebih lanjut terkait dengan pengelolaan buku kas umum, buku kas pembantu pajak, dan buku bank desa dapat ditambahkan fungsionalitasnya sehingga pengelolaan keuangan desa dapat berjalan lebih maksimal.

DAFTAR PUSTAKA

Republik Indonesia, Rencana Pembangunan Jangka Menengah Nasional 2014-2019.

-----, Peraturan Menteri Dalam Negeri No 113 Tahun 2014 Tentang Pengelolaan Keuangan Desa.

-----, Peraturan Menteri Dalam Negeri No 114 Tahun 2014 Tentang Pedoman Pembangunan Desa.

-----, Peraturan Pemerintah Nomor 43 Tahun 2014 Tentang Peraturan Pelaksanaan Undang - Undang

-----, Peraturan Pemerintah Nomor 60 Tahun 2014 Tentang Dana Desa

-----, Undang - undang Nomor 6 Tahun 2014 Tentang Desa.

Ananda, E. R., 2014. *PEMBANGUNAN SISTEM INFORMASI DESA BINAAN UNIVERSITAS ATMA JAYA YOGYAKARTA BERBASIS WEB*, Yogyakarta: Universitas Atma Jaya Yogyakarta.

Arisanti, A., 2012. PERANCANGAN SISTEM INFORMASI PENDATAAN PENDUDUK BERBASIS WEB MENGGUNAKAN METODE WATERFALL PADA DESA BOGOREJO KECAMATAN GEDONGTATAAN. *Jurnal Sistem Informasi*, Volume 1, p. 8.

Dwi, P., Bambang Eka, P. & Siska, I., 2013. Sistem Informasi Data Penduduk Pada Desa Bogoharjo Kecamatan Ngadirojo Kabupaten Pacitan. *Indonesian Journal on Networking and Security (IJNS)*, Oktober, Volume 2, pp. 1-7.

Kessa, W., 2015. *Perencanaan Pembangunan Desa*. 1 ed. Jakarta: KEMENTERIAN DESA, PEMBANGUNAN DAERAH TERTINGGAL, DAN TRANSMIGRASI REPUBLIK INDONESIA.

Kurniawan, I. I., 2014. *Pembangunan Sistem Informasi Desa Berbasis Web*, Yogyakarta: Universitas Atma Jaya Yogyakarta.

O'Brein, J. A. & Marakas, G. M., 2007. *Introduction to Information Systems*. 5th ed. New York: McGraw-Hill.

Riyadi, A. S., Retnadi, E. & Supriatna, A. D., 2012. Perancangan Sistem Informasi Berbasis Website Subsistem Guru Di Sekolah Pesantren Persatuan Islam 99 Rancabango. *Jurnal Algoritma Sekolah Tinggi Teknologi Garut*, Volume 09, pp. 1-11.

Sugara, E. P. A., 2011. Sistem Informasi Pencarian Dan Penjualan Barang Berbasis Web Pada Toko Bagus. *Jurnal Teknologi dan Informatika (TEKNOMATIKA)*, Mei, Volume 1, pp. 158-166.

Wahana, K., 2006. *Apa & Bagaimana E-Commerce*. 2 ed. Yogyakarta: Andi Offset.

Warsono, P. T., 2014. *Pembangunan Sistem Informasi Rumah Sakit untuk mendukung Pelayanan Pasien di Rumah Sakit Umum Patmasuri Yogyakarta*, Yogyakarta: Universitas Atma Jaya Yogyakarta.

Wijaya, A. B. M., 2011. *Pembangunan Sistem Informasi Rencana Kerja dan Anggaran (RKA) Kantor Keuangan Universitas Atma Jaya Yogyakarta*, Yogyakarta: Universitas Atma Jaya Yogyakarta.

LAMPIRAN



SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

SISTEM INFORMASI DESA DAN KAWASAN UNTUK PERENCANAAN PEMBANGUNAN (SIDEKa-PPD)

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Lucas Sandy Duta Arga / 110706588

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		SKPL- SIDEKa-PPD		
		Revisi		1/68

Program Studi Teknik Informatika	SKPL –SIDEKa-PPD	1/ 68
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		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	LSDA							
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan	6
1.1	Tujuan	6
1.2	Lingkup Masalah	6
1.3	Definisi, Akronim, dan Singkatan	7
	Server.....	7
	Framework.....	8
	PHP.....	8
	Code Igniter.....	8
1.4	Referensi	8
1.5	Deskripsi Umum	8
2	Deskripsi Kebutuhan	9
2.1	Perspektif Produk	9
2.2	Fungsi Produk	10
2.3	Karakteristik Pengguna	22
2.4	Batasan – Batasan	22
2.5	Asumsi dan Ketergantungan	22
3	Kebutuhan Khusus	22
3.1	Kebutuhan Antarmuka Ekternal	22
3.1.1	Antarmuka Pemakai.....	23
3.1.2	Antarmuka Perangkat Keras.....	23
3.1.3	Antarmuka Perangkat Lunak.....	23
3.1.4	Antarmuka Komunikasi.....	24
3.2	Kebutuhan Fungsionalitas Perangkat Lunak	25
3.2.1	Use Case Diagram.....	25
4	Spesifikasi Rinci Kebutuhan	25
4.1	<i>Use Case Spesification: Login</i>	25
4.2	<i>Use Case Spesification: Mengelola RPJMD</i>	27
4.3	<i>Use Case Spesification: Mengelola RPJMDes</i>	29
4.4	<i>Use Case Spesification: Mengelola Detail RPJMDes</i>	32
4.5	<i>Use Case Spesification: Mencetak Detil RPJMDes</i>	35
4.6	<i>Use Case Spesification: Mengelola RKPDes</i>	36
4.7	<i>Use Case Spesification: Mengelola RABDes</i>	38
4.8	<i>Use Case Spesification: Mengelola RABDes Anggaran</i>	41
4.9	<i>Use Case Spesification: Mengelola SPP</i>	44
4.10	<i>Use Case Spesification: Mengelola Detail SPP</i>	47

4.11	<i>Use Case Spesification: Mengelola LPJ</i>	49
4.12	<i>Use Case Spesification: Mengelola Detail LPJ</i>	52
4.13	<i>Use Case Spesification: Mengelola Periode</i>	54
4.14	<i>Use Case Spesification: Mengelola Tahun Anggaran</i>	57
4.15	<i>Use Case Spesification: Mengelola Bidang</i>	59
4.16	<i>Use Case Spesification: Mengelola Kode Rekening</i>	62
4.17	<i>Use Case Spesification: Mengelola Sumber Dana</i>	64
5	Entity Relationship Diagram (ERD)	68

Daftar Gambar

Gambar 1	Arsitektur Perangkat Lunak SDeKa-PPD.....	10
Gambar 3	Use Case Diagram SDeKa-PPD.....	25
Gambar 4	Entity Relationship Diagram SDeKa-PPD.....	68

1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SIDEKa-PPD (Sistem Informamsi Desa dan Kawasan untuk Perencanaan Pembangunan Desa) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara system dengan system 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-SIDEKa-PPD ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat Lunak ini dikembangkan dengan tujuan:

1. Menangani pengelolaan RPJMD (Rencana Pembangunan Jangka Menengah Daerah).
2. Menangani pengelolaan RPJMDes (Rencana Pembangunan Jangka Menengah Desa).
3. Menangani pengelolaan RKPDes (Rencana Kerja Pemerintah Desa).
4. Menangani pengelolaan RABDes (Rencana Anggaran Belanja Desa).
5. Menangani pengelolaan SPP (Surat Permintaan Pembayaran).
6. Menangani pengelolaan LPJ (Laporan Pertanggungjawaban).
7. Menangani pengelolaan Pustaka Perencanaan meliputi Periode, Tahun Anggaran, Bidang, Kode Rekening, dan Sumber Dana.

Program Studi Teknik Informatika	SKPL –SIDEKa-PPD	6/ 68
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		

Aplikasi ini berjalan pada lingkungan dengan *platform* browser (untuk *web application*).

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-SIDeKa-PPD-XXX	Kode yang merepresentasikan kebutuhan pada SIDeKa-PPD (Sistem Informasi Desa dan Kawasan untuk Perencanaan Pembangunan Desa) dimana XXX merupakan nomor fungsi produk.
SIDeKa-PPD	Perangkat lunak pengelolaan web.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.
RPJMD	Rencana Pembangunan Jangka Menengah Daerah.
RPJMDes	Rencana Pembangunan Jangka Menengah Desa.
RKPDes	Rencana Kerja Pemerintah Desa.
RABDes	Rencana Anggaran Biaya Desa
SPP	Surat Permintaan Pembayaran
LPJ	Laporan Penanggungjawaban.

Framework	Suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah kompleks.
PHP	Hypertext Preprocessor merupakan bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML dan banyak dipakai untuk memprogram situs web dinamis.
Code Igniter	Aplikasi open source yang berupa framework dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Irene Deandra Indarto, *Spesifikasi Kebutuhan Perangkat Lunak (SKPL) SISP*, Universitas Atma Jaya Yogyakarta.

1.5 Deskripsi Umum

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 SDeKa-PPD 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 SDeKa-PPD tersebut.

Program Studi Teknik Informatika	SKPL –SDeKa-PPD	8/ 68
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		

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak SDeKa-PPD yang akan dikembangkan.

2 Deskripsi Kebutuhan

2.1 Perspektif Produk

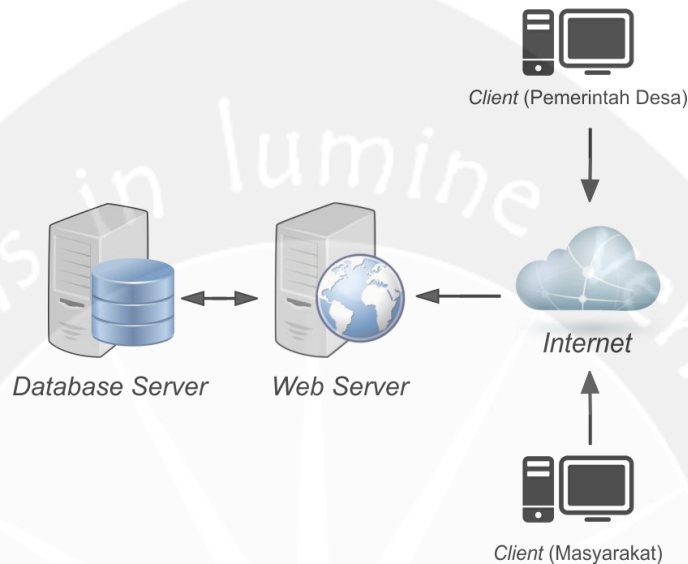
Sistem Informasi Desa dan Kawasan untuk Perencanaan Pembangunan Desa Berbasis Web merupakan perangkat lunak yang dikembangkan untuk membantu pengelolaan perencanaan pembangunan yang ada di desa. Sistem ini menangani perencanaan pembangunan desa meliputi RPJMD (Rencana Pembangunan Jangka Menengah Daerah), RPJMDes (Rencana Pembangunan Jangka Menengah Desa), RKPDes (Rencana Kerja Pemerintah Desa), RABDes (Rencana Anggaran Biaya Desa), SPP (Surat Permintaan Pembayaran), dan LPJ (Laporan Pertanggungjawaban) yang diharapkan dapat membantu pembangunan yang ada di desa. Selain dari segi penggunaan, sistem ini juga dapat digunakan untuk melakukan pengelolaan RPJMD, RPJMDes, RKPDes, RABDes, SPP, LPJ, dan pengelolaan pustaka perencanaan meliputi periode, tahun anggaran, bidang, kode rekening, dan sumber dana.

Perangkat lunak SDeKa-PPD ini berjalan pada platform berbasis web dan dibuat menggunakan bahasa pemrograman PHP dengan framework Codeigniter. Sedangkan untuk lingkungan pemrogramannya menggunakan Notepad++.

Pengguna atau user akan berinteraksi dengan sistem melalui antarmuka *GUI (Graphical User Interface)*. Pada sistem ini arsitektur perangkat lunak yang digunakan berupa client server, dimana data disimpan dalam server. User dapat mengakses data yang ada di server tersebut secara online dengan memanggil web service pada website yang tersedia di dalam web server (web base).

Program Studi Teknik Informatika	SKPL –SDeKa-PPD	9/ 68
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		

Input data yang dimasukkan akan disimpan dalam database, sehingga jika ada pencarian data, maka data yang diinginkan akan dicari ke database server yang selanjutnya dikirimkan ke client yang merequest melalui server.



Gambar 1 Arsitektur Perangkat Lunak SDeKa-PPD

2.2 Fungsi Produk

Fungsi perangkat lunak SDeKa-PPD (Sistem Informasi Perencanaan Pembangunan Desa) adalah sebagai berikut:

1. Fungsi *Login* (SKPL-SDeKa-PPD-001) :

Merupakan fungsi yang digunakan oleh pemerintah desa untuk dapat masuk ke dalam sistem yang akan digunakan.

2. Fungsi *Ubah Password* (SKPL-SDeKa-PPD-002) :

Merupakan fungsi yang digunakan untuk mengubah password.

3. Fungsi *Pengelolaan RPJMD* (SKPL-SDeKa-PPD-003) :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan dari setiap desa untuk mengelola RPJMD (Rencana Pembangunan Jangka Menengah Daerah).

Fungsi pengelolaan RPJMD meliputi:

a. Fungsi *Tambah Data RPJMD* (SKPL-SDeKa-PPD-003-01) .

Program Studi Teknik Informatika	SKPL –SDeKa-PPD	10/ 68
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 menambah data RPJMD.

b. Fungsi *Sunting Data RPJMD* **(SKPL-SIDeKa-PPD-003-02)** .

Merupakan fungsi yang digunakan untuk mengubah data RPJMD.

c. Fungsi *Hapus RPJMD* **(SKPL-SIDeKa-PPD-003-03)** .

Merupakan fungsi yang digunakan untuk menghapus data RPJMD.

d. Fungsi *Cari RPJMD* **(SKPL-SIDeKa-PPD-003-04)** .

Merupakan fungsi yang digunakan untuk mencari data RPJMD berdasarkan atribut yang ada.

e. Fungsi *Tampil Data RPJMD* **(SKPL-SIDeKa-PPD-003-05)** .

Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data RPJMD.

4. Fungsi *Pengelolaan RPJMDes* **(SKPL-SIDeKa-PPD-004)** :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan dari setiap desa untuk mengelola RPJMDes (Rencana Pembangunan Jangka Menengah Desa).

Fungsi pengelolaan RPJMDes meliputi:

a. Fungsi *Tambah Data RPJMDes* **(SKPL-SIDeKa-PPD-004-01)** .

Merupakan fungsi yang digunakan untuk menambah data RPJMDes.

b. Fungsi *Sunting Data RPJMDes* **(SKPL-SIDeKa-PPD-004-02)** .

Merupakan fungsi yang digunakan untuk mengubah data RPJMDes.

c. Fungsi *Hapus RPJMDes* **(SKPL-SIDeKa-PPD-004-03)** .

Merupakan fungsi yang digunakan untuk menghapus data RPJMDes.

d. Fungsi *Cari RPJMDes* **(SKPL-SIDeKa-PPD-004-04)** .

Merupakan fungsi yang digunakan untuk mencari data RPJMDes berdasarkan atribut yang ada.

Program Studi Teknik Informatika	SKPL –SIDeKa-PPD	11/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- e. Fungsi *Tampil Data RPJMDes* (**SKPL-SIDeKa-PPD-004-05**) .
Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data RPJMDes.
- f. Fungsi *Tampil Data RPJMDes Detail* (**SKPL-SIDeKa-PPD-004-06**) .
Merupakan fungsi yang digunakan untuk menampilkan detail data dari setiap RPJMDes yang dipilih.
- g. Fungsi *Tambah Data RPJMDes Detail* (**SKPL-SIDeKa-PPD-004-07**) .
Merupakan fungsi yang digunakan untuk menambahkan detail data dari setiap RPJMDes yang dipilih.
- h. Fungsi *Sunting Data RPJMDes Detail* (**SKPL-SIDeKa-PPD-004-08**) .
Merupakan fungsi yang digunakan untuk mengubah detail data RPJMDes.
- i. Fungsi *Hapus Data RPJMDes Detail* (**SKPL-SIDeKa-PPD-004-09**) .
Merupakan fungsi yang digunakan untuk menghapus detail data RPJMDes.
- j. Fungsi *Cari Data RPJMDes Detail* (**SKPL-SIDeKa-PPD-004-10**) .
Merupakan fungsi yang digunakan untuk mencari detail data dari setiap RPJMDes yang dipilih.
- k. Fungsi *Tampil Data RPJMDes Detail Periode* (**SKPL-SIDeKa-PPD-004-11**) .
Merupakan fungsi yang digunakan untuk menampilkan detail data dari setiap RPJMDes yang dipilih berdasarkan periode yang dipilih.
- l. Fungsi *Cetak Data RPJMDes Detail Periode* (**SKPL-SIDeKa-PPD-004-12**) .
Merupakan fungsi yang digunakan untuk mencetak detail data dari setiap RPJMDes yang dipilih berdasarkan

periode yang dipilih. Hasil cetakan berupa file excel.

5. Fungsi *Pengelolaan RKPDes* **SKPL-SIDeKa-PPD-005**) :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan dari setiap desa untuk mengelola RKPDes (Rencana Kerja Pemerintah Desa). Fungsi pengelolaan RKPDes meliputi:

- a. Fungsi *Tambah Data RKPDes* **(SKPL-SIDeKa-PPD-005-01)** .
Merupakan fungsi yang digunakan untuk menambah data RKPDes.
- b. Fungsi *Sunting Data RKPDes* **(SKPL-SIDeKa-PPD-005-02)** .
Merupakan fungsi yang digunakan untuk mengubah data RKPDes.
- c. Fungsi *Hapus RKPDes* **(SKPL-SIDeKa-PPD-005-03)** .
Merupakan fungsi yang digunakan untuk menghapus data RKPDes.
- d. Fungsi *Cari RKPDes* **(SKPL-SIDeKa-PPD-005-04)** .
Merupakan fungsi yang digunakan untuk mencari data RPJMD berdasarkan atribut yang ada.
- e. Fungsi *Tampil Data RKPDes* **(SKPL-SIDeKa-PPD-005-05)** .
Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data RKPDes.
- f. Fungsi *Pengelolaan Sumber Dana* **(SKPL-SIDeKa-PPD-005-06)** .

Merupakan fungsi yang digunakan untuk mengelola sumber dana yang ada di Rencana Kerja Pemerintah Desa. Sumber Dana ini adalah referensi dari RKPDes. Fungsi Pengelolaan Sumber Dana meliputi:

- a) Fungsi *Tambah Sumber Dana* **(SKPL-SIDeKa-PPD-005-06-01)** .

Merupakan fungsi yang digunakan untuk menambah data Sumber Dana.

Program Studi Teknik Informatika	SKPL –SIDeKa-PPD	13/ 68
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		

b) Fungsi Sunting Sumber Dana (**SKPL-SIDeKa-PPD-005-06-02**) .

Merupakan fungsi yang digunakan untuk mengubah data Sumber Dana.

c) Fungsi Hapus Sumber Dana (**SKPL-SIDeKa-PPD-005-06-03**) .

Merupakan fungsi yang digunakan untuk menghapus data Sumber Dana.

d) Fungsi Cari Sumber Dana (**SKPL-SIDeKa-PPD-005-06-04**) .

Merupakan fungsi yang digunakan untuk mencari data Sumber Dana.

e) Fungsi Tampil Sumber Dana (**SKPL-SIDeKa-PPD-005-06-05**) .

Merupakan fungsi yang digunakan untuk menampilkan semua data Sumber Dana.

6. Fungsi *Pengelolaan RABDes* (**SKPL-SIDeKa-PPD-006**) :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan dari setiap desa untuk mengelola RABDes (Rencana Anggaran Biaya Desa). Fungsi pengelolaan RABDes meliputi:

a. Fungsi *Tambah Data RABDes* (**SKPL-SIDeKa-PPD-006-01**) .

Merupakan fungsi yang digunakan untuk menambah data RABDes.

b. Fungsi *Sunting Data RABDes* (**SKPL-SIDeKa-PPD-006-02**) .

Merupakan fungsi yang digunakan untuk mengubah data RABDes.

c. Fungsi *Hapus RABDes* (**SKPL-SIDeKa-PPD-006-03**) .

Merupakan fungsi yang digunakan untuk menghapus data RABDes.

d. Fungsi *Cari RABDes* (**SKPL-SIDeKa-PPD-006-04**) .

Program Studi Teknik Informatika	SKPL –SIDeKa-PPD	14/ 68
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 mencari data RABDes berdasarkan atribut yang ada.

e. Fungsi *Tampil Data RABDes* **(SKPL-SIDeKa-PPD-006-05)** .

Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data RABDes.

f. Fungsi *Tampil Data RABDes Anggaran* **(SKPL-SIDeKa-PPD-006-06)** .

Merupakan fungsi yang digunakan untuk menampilkan anggaran dari setiap RABDes yang dipilih.

g. Fungsi *Tambah Data RABDes Anggaran* **(SKPL-SIDeKa-PPD-006-07)** .

Merupakan fungsi yang digunakan untuk menambahkan anggaran dari setiap RABDes yang dipilih.

h. Fungsi *Sunting Data RABDes Anggaran* **(SKPL-SIDeKa-PPD-006-08)** .

Merupakan fungsi yang digunakan untuk mengubah anggaran data RABDes.

i. Fungsi *Hapus Data RABDes Anggaran* **(SKPL-SIDeKa-PPD-006-09)** .

Merupakan fungsi yang digunakan untuk menghapus anggaran data RABDes.

j. Fungsi *Cari Data RABDes Anggaran* **(SKPL-SIDeKa-PPD-006-10)** .

Merupakan fungsi yang digunakan untuk mencari anggaran dari setiap RABDes yang dipilih.

k. Fungsi *Cetak Data RABDes* **(SKPL-SIDeKa-PPD-006-11)** .

Merupakan fungsi yang digunakan untuk mencetak data dari setiap RABDes yang dipilih dan menampilkan anggaran RABDes. Hasil cetakan berupa file pdf.

7. Fungsi *Pengelolaan SPP* **(SKPL-SIDeKa-PPD-007)** :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan dari setiap desa untuk mengelola

Program Studi Teknik Informatika	SKPL –SIDeKa-PPD	15/ 68
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		

SPP (Surat Permintaan Pembayaran). Fungsi pengelolaan SPP meliputi:

- a. Fungsi *Tambah Data SPP* (**SKPL-SIDeKa-PPD-007-01**) .
Merupakan fungsi yang digunakan untuk menambah data SPP.
- b. Fungsi *Sunting Data SPP* (**SKPL-SIDeKa-PPD-007-02**) .
Merupakan fungsi yang digunakan untuk mengubah data SPP.
- c. Fungsi *Hapus SPP* (**SKPL-SIDeKa-PPD-007-03**) .
Merupakan fungsi yang digunakan untuk menghapus data SPP.
- d. Fungsi *Cari SPP* (**SKPL-SIDeKa-PPD-007-04**) .
Merupakan fungsi yang digunakan untuk mencari data SPP berdasarkan atribut yang ada.
- e. Fungsi *Tampil Data SPP* (**SKPL-SIDeKa-PPD-007-05**) .
Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data SPP.
- f. Fungsi *Tampil Data SPP Detail* (**SKPL-SIDeKa-PPD-007-06**) .
Merupakan fungsi yang digunakan untuk menampilkan detail data dari setiap SPP yang dipilih.
- g. Fungsi *Tambah Data SPP Detail* (**SKPL-SIDeKa-PPD-007-07**) .
Merupakan fungsi yang digunakan untuk menambahkan detail data dari setiap SPP yang dipilih.
- h. Fungsi *Sunting Data SPP Detail* (**SKPL-SIDeKa-PPD-007-08**) .
Merupakan fungsi yang digunakan untuk mengubah detail data SPP.
- i. Fungsi *Hapus Data SPP Detail* (**SKPL-SIDeKa-PPD-007-09**) .

Program Studi Teknik Informatika	SKPL –SIDeKa-PPD	16/ 68
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 menghapus detail data SPP.

j. Fungsi *Cari Data SPP Detail* **(SKPL-SIDeKa-PPD-007-10)**.

Merupakan fungsi yang digunakan untuk mencari detail data dari setiap SPP yang dipilih.

k. Fungsi *Cetak Data SPP* **(SKPL-SIDeKa-PPD-007-11)**.

Merupakan fungsi yang digunakan untuk mencetak data dari setiap SPP yang dipilih dan menampilkan detail data SPP. Hasil cetakan berupa file pdf.

8. Fungsi *Pengelolaan LPJ* **(SKPL-SIDeKa-PPD-008)** :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan dari setiap desa untuk mengelola LPJ (Laporan Penanggungjawaban). Fungsi pengelolaan LPJ meliputi:

a. Fungsi *Tambah Data LPJ* **(SKPL-SIDeKa-PPD-008-01)**.

Merupakan fungsi yang digunakan untuk menambah data LPJ.

b. Fungsi *Sunting Data LPJ* **(SKPL-SIDeKa-PPD-008-02)**.

Merupakan fungsi yang digunakan untuk mengubah data LPJ.

c. Fungsi *Hapus LPJ* **(SKPL-SIDeKa-PPD-008-03)**.

Merupakan fungsi yang digunakan untuk menghapus data LPJ.

d. Fungsi *Cari LPJ* **(SKPL-SIDeKa-PPD-008-04)**.

Merupakan fungsi yang digunakan untuk mencari data LPJ berdasarkan atribut yang ada.

e. Fungsi *Tampil Data LPJ* **(SKPL-SIDeKa-PPD-008-05)**.

Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data LPJ.

f. Fungsi *Tampil Data LPJ Detail* **(SKPL-SIDeKa-PPD-008-06)**.

Program Studi Teknik Informatika	SKPL –SIDeKa-PPD	17/ 68
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 menampilkan detail data dari setiap LPJ yang dipilih.

- g. Fungsi *Tambah Data LPJ Detail* **(SKPL-SIDeKa-PPD-008-07)** .

Merupakan fungsi yang digunakan untuk menambahkan detail data dari setiap LPJ yang dipilih.

- h. Fungsi *Sunting Data LPJ Detail* **(SKPL-SIDeKa-PPD-008-08)** .

Merupakan fungsi yang digunakan untuk mengubah detail data LPJ.

- i. Fungsi *Hapus Data LPJ Detail* **(SKPL-SIDeKa-PPD-008-09)** .

Merupakan fungsi yang digunakan untuk menghapus detail data LPJ.

- j. Fungsi *Cari Data LPJ Detail* **(SKPL-SIDeKa-PPD-008-10)** .

Merupakan fungsi yang digunakan untuk mencari detail data dari setiap LPJ yang dipilih.

- k. Fungsi *Cetak Data LPJ* **(SKPL-SIDeKa-PPD-008-11)** .

Merupakan fungsi yang digunakan untuk mencetak data dari setiap LPJ yang dipilih dan menampilkan detail data LPJ. Hasil cetakan berupa file pdf.

9. Fungsi Pengelolaan Periode **(SKPL-SIDeKa-PPD-009)** :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan untuk mengelola periode.

Fungsi pengelolaan Periode meliputi:

- a. Fungsi *Tambah Data Periode* **(SKPL-SIDeKa-PPD-009-01)** .

Merupakan fungsi yang digunakan untuk menambah data Periode.

- b. Fungsi *Sunting Data Periode* **(SKPL-SIDeKa-PPD-009-02)** .

Program Studi Teknik Informatika	SKPL –SIDeKa-PPD	18/ 68
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 Periode.

c. Fungsi *Hapus Periode* **(SKPL-SIDeKa-PPD-009-03)** .

Merupakan fungsi yang digunakan untuk menghapus data Periode.

d. Fungsi *Cari Periode* **(SKPL-SIDeKa-PPD-009-04)** .

Merupakan fungsi yang digunakan untuk mencari data Periode berdasarkan atribut yang ada.

e. Fungsi *Tampil Data Periode* **(SKPL-SIDeKa-PPD-009-05)** .

Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data Periode.

10. Fungsi Pengelolaan Tahun Anggaran **(SKPL-SIDeKa-PPD-010)** :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan untuk mengelola tahun anggaran.

Fungsi pengelolaan Tahun Anggaran meliputi:

a. Fungsi *Tambah Data Tahun Anggaran* **(SKPL-SIDeKa-PPD-010-01)** .

Merupakan fungsi yang digunakan untuk menambah data Tahun Anggaran.

b. Fungsi *Sunting Data Tahun Anggaran* **(SKPL-SIDeKa-PPD-010-02)** .

Merupakan fungsi yang digunakan untuk mengubah data Tahun Anggaran.

c. Fungsi *Hapus Tahun Anggaran* **(SKPL-SIDeKa-PPD-010-03)** .

Merupakan fungsi yang digunakan untuk menghapus data Tahun Anggaran.

d. Fungsi *Cari Tahun Anggaran* **(SKPL-SIDeKa-PPD-010-04)** .

Merupakan fungsi yang digunakan untuk mencari data Tahun Anggaran berdasarkan atribut yang ada.

Program Studi Teknik Informatika	SKPL –SIDeKa-PPD	19/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- e. Fungsi *Tampil Data Tahun Anggaran* (**SKPL-SIDeKa-PPD-010-05**) .

Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data Tahun Anggaran.

11. Fungsi Pengelolaan Bidang (**SKPL-SIDeKa-PPD-011**) :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan untuk mengelola bidang.

Fungsi pengelolaan Bidang meliputi:

- a. Fungsi *Tambah Data Bidang* (**SKPL-SIDeKa-PPD-011-01**) .

Merupakan fungsi yang digunakan untuk menambah data Bidang.

- b. Fungsi *Sunting Data Bidang* (**SKPL-SIDeKa-PPD-011-02**) .

Merupakan fungsi yang digunakan untuk mengubah data Bidang.

- c. Fungsi *Hapus Bidang* (**SKPL-SIDeKa-PPD-011-03**) .

Merupakan fungsi yang digunakan untuk menghapus data Bidang.

- d. Fungsi *Cari Bidang* (**SKPL-SIDeKa-PPD-011-04**) .

Merupakan fungsi yang digunakan untuk mencari data Bidang berdasarkan atribut yang ada.

- e. Fungsi *Tampil Data Bidang* (**SKPL-SIDeKa-PPD-011-05**) .

Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data Bidang.

12. Fungsi Pengelolaan Kode Rekening (**SKPL-SIDeKa-PPD-012**) :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan untuk mengelola bidang.

Fungsi pengelolaan Kode Rekening meliputi:

- a. Fungsi *Tambah Data Kode Rekening* (**SKPL-SIDeKa-PPD-012-01**) .

Merupakan fungsi yang digunakan untuk menambah data Kode Rekening.

Program Studi Teknik Informatika	SKPL –SIDeKa-PPD	20/ 68
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		

- b. Fungsi *Sunting Data Kode Rekening* (**SKPL-SIDeKa-PPD-012-02**) .

Merupakan fungsi yang digunakan untuk mengubah data Kode Rekening.

- c. Fungsi *Hapus Kode Rekening* (**SKPL-SIDeKa-PPD-012-03**) .

Merupakan fungsi yang digunakan untuk menghapus data Kode Rekening.

- d. Fungsi *Cari Kode Rekening* (**SKPL-SIDeKa-PPD-012-04**) .

Merupakan fungsi yang digunakan untuk mencari data Kode Rekening berdasarkan atribut yang ada.

- e. Fungsi *Tampil Data Kode Rekening* (**SKPL-SIDeKa-PPD-012-05**) .

Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data Kode Rekening.

13. Fungsi Pengelolaan Sumber Dana (**SKPL-SIDeKa-PPD-013**) :

Merupakan fungsi yang digunakan oleh seorang Perencana Pembangunan untuk mengelola bidang.

Fungsi pengelolaan Sumber Dana meliputi:

- a. Fungsi *Tambah Data Sumber Dana* (**SKPL-SIDeKa-PPD-013-01**) .

Merupakan fungsi yang digunakan untuk menambah data Sumber Dana.

- b. Fungsi *Sunting Data Sumber Dana* (**SKPL-SIDeKa-PPD-013-02**) .

Merupakan fungsi yang digunakan untuk mengubah data Sumber Dana.

- c. Fungsi *Hapus Sumber Dana* (**SKPL-SIDeKa-PPD-013-03**) .

Merupakan fungsi yang digunakan untuk menghapus data Sumber Dana.

- d. Fungsi *Cari Sumber Dana* (**SKPL-SIDeKa-PPD-013-04**) .

Merupakan fungsi yang digunakan untuk mencari data Sumber Dana berdasarkan atribut yang ada.

Program Studi Teknik Informatika	SKPL –SIDeKa-PPD	21/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- e. Fungsi *Tampil Data Sumber Dana* (**SKPL-SIDeKa-PPD-013-05**) .

Merupakan fungsi yang digunakan untuk menampilkan keseluruhan data Sumber Dana.

14. Fungsi Logout (**SKPL-SIDeKa-PPD-014**) :

Merupakan fungsi yang digunakan oleh pemerintah desa untuk keluar dari sistem.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SIDeKa-PPD adalah sebagai berikut :

1. Memahami pengoperasian komputer ataupun laptop dengan sistem operasi yang ada di dalamnya.
2. Mengerti tentang internet dan web.
3. Memahami penggunaan aplikasi web SIDeKa-PPD.

2.4 Batasan - Batasan

Batasan - batasan dalam pengembangan perangkat lunak SIDeKa-PPD adalah sebagai berikut:

1. Kebijakan Umum
Berpedoman pada tujuan dari pengembangan perangkat lunak SIDeKa-PPD.
2. Keterbatasan perangkat keras
Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada perangkat desktop (komputer ataupun laptop) yang menggunakan sistem operasi Windows XP/Vista/7/8/10, Mac OS, ataupun Linux, serta aplikasi web browser apapun.

3 Kebutuhan Khusus

3.1 Kebutuhan Antarmuka Eksternal

Kebutuhan antar muka eksternal pada perangkat lunak SIDeKa-PPD meliputi kebutuhan antarmuka pemakai, antarmuka

Program Studi Teknik Informatika	SKPL -SIDeKa-PPD	22/ 68
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		

perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi.

3.1.1 Antarmuka Pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form.

3.1.2 Antarmuka Perangkat Keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SIDEKa-PPD adalah:

1. Perangkat computer atau laptop.
2. Perangkat Database Server.
3. Perangkat Web Server.

3.1.3 Antarmuka Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak SIDEKa-PPD adalah sebagai berikut:

1. Nama : MySQL
Sumber : Oracle.

Sebagai database management system (DBMS) yang digunakan untuk menyimpan data di sisi server.

2. Nama : Windows XP/Vista/7/8/10
Sumber : Microsoft.

Sebagai sistem operasi yang digunakan.

3. Nama : Apache
Sumber : Apache Software Foundation.

Sebagai web server.

4. Nama : Code Igniter
Sumber : EllisLab, Inc.

Sebagai framework untuk menjalankan aplikasi.

5. Nama : IE/Firefox/Chrome/dll
Sumber : Microsoft/Mozilla/Google/dll

Sebagai web browser untuk membuka sistem web.

Program Studi Teknik Informatika	SKPL –SIDEKa-PPD	23/ 68
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		

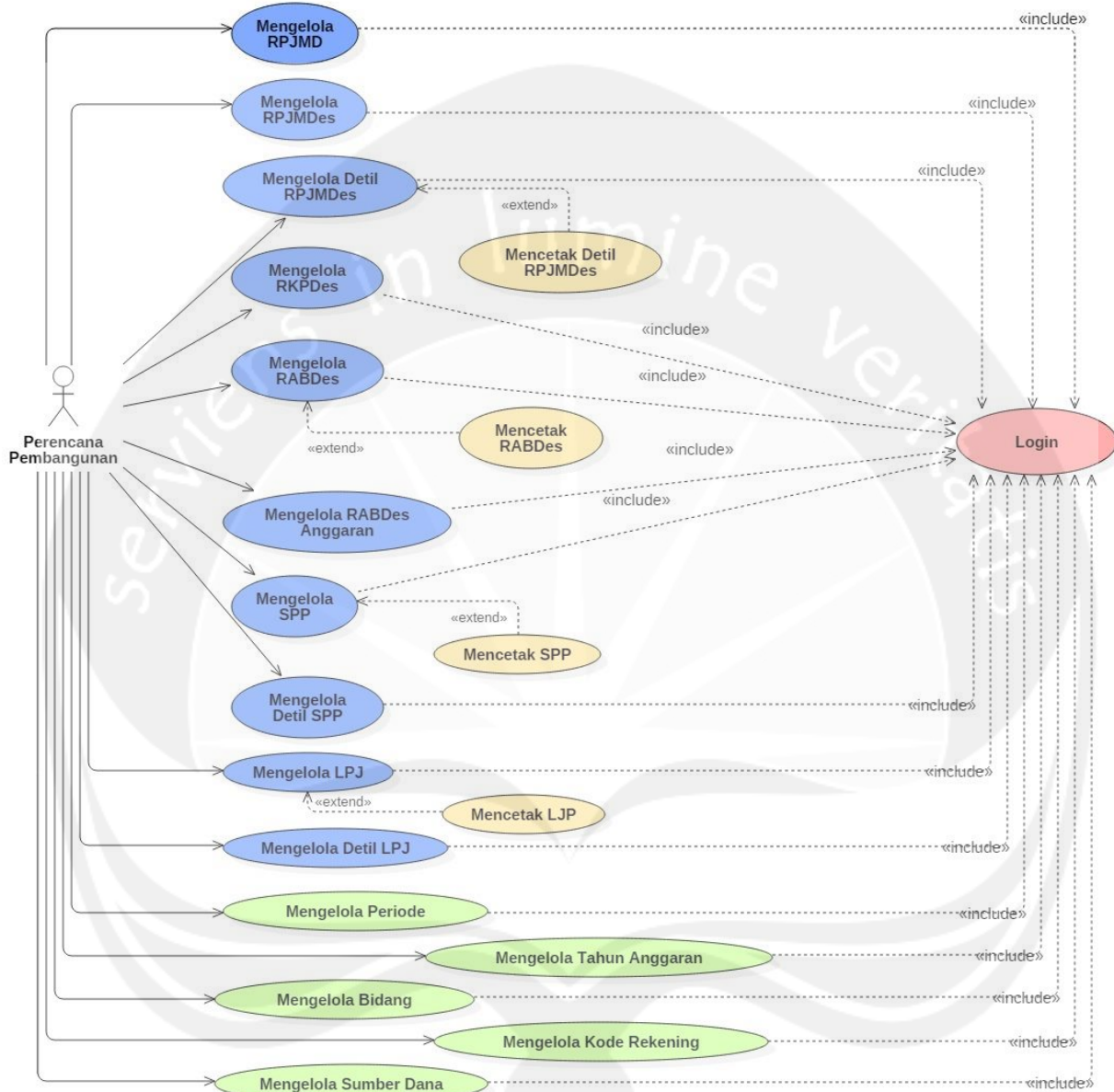
6. Nama : Notepad++
Sumber : Don Ho.
Sebagai aplikasi untuk editor program.

3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak BATRAS menggunakan protocol TCP/IP, sedangkan untuk aplikasi web menggunakan protocol HTTP.

3.2 Kebutuhan Fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram



Gambar 2 Use Case Diagram SIDEKa-PPD

4 Spesifikasi Rinci Kebutuhan

4.1 Use Case Spesification: Login

1. Brief Description

Use case ini digunakan untuk memperoleh akses masuk ke sistem yang digunakan oleh Administrator, Pengelola

Program Studi Teknik Informatika	SKPL –SIDEKa-PPD	25 / 68
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		

Data, Pengelola Asset, Pengelola Peta, Perencana Pembangunan.

2. Primary Aktor

1. Administrator
2. Pengelola Data
3. Pengelola Asset
4. Pengelola Peta
5. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan antarmuka untuk login.
3. Aktor menginputkan username dan password.
4. Sistem melakukan validasi terhadap username dan password yang telah diinputkan.

E-1. Username dan password yang diinputkan actor salah.

5. Sistem memberikan akses masuk ke dalam sistem.
6. Use Case selesai.

5. Alternative Flow

None

6. Error Flow

E-1. Username dan password yang diinputkan actor salah.

1. Sistem memberi peringatan bahwa data yang diinputkan salah.
2. Kembali ke Basic Flow langkah ke-3.

7. Pre-Conditions

1. Aktor telah terdaftar.

8. Post-Conditions

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	26/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem sesuai role yang dimilikinya.

4.2 Use Case Spesification: Mengelola RPJMD

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan RPJMD (Rencana Pembangunan Jangka Menengah Daerah). Aktor dapat melakukan tambah, edit, hapus, dan cari data RPJMD.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan RPJMD.
2. Sistem menampilkan antarmuka pengelolaan RPJMD berupa data table *flexigrid*.
3. Aktor memilih tambah data RPJMD.
 - A-1. Aktor memilih edit data RPJMD.
 - A-2. Aktor memilih hapus data RPJMD.
 - A-3. Aktor memilih cari data RPJMD.
 - A-4. Aktor memilih menambah sub program RPJMD.
4. Sistem akan menampilkan form isian untuk menambahkan RPJMD.
5. Aktor menginputkan data RPJMD.
6. Setelah selesai menginputkan data RPJMD, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data RPJMD yang telah diinputkan
 - E-1. Data RPJMD yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	27/ 68
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		

9. Use Case selesai.

5. Alternative Flow

A-1. Aktor memilih edit data RPJMD.

1. Sistem akan menampilkan form untuk mengubah data RPJMD yang sudah dipilih aktor.
2. Aktor mengubah data RPJMD yang sudah dipilih.
3. Setelah selesai mengubah data RPJMD, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.
4. Sistem melakukan pengecekan data RPJMD yang telah diubah

E-2. Data RPJMD yang diubah aktor salah.

5. Sistem menyimpan data ke database.
6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data RPJMD.

1. Aktor memilih data RPJMD yang ingin dihapus.
2. Sistem melakukan pengecekan data RPJMD yang ingin dihapus.

E-3. Data RPJMD tidak bisa dihapus.

3. Sistem menghapus data yang di database.
4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data RPJMD.

1. Aktor memilih tombol pencarian data RPJMD.
2. Kemudian, aktor mengetik data RPJMD yang ingin dicari berdasarkan atribut yang sudah disediakan.
3. Sistem menampilkan data yang ingin dicari aktor.
4. Berlanjut ke basic flow langkah ke 8.

A-4. Aktor memilih menambah sub program RPJMD.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	28/ 68
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 antarmuka Detail RPJMD berupa tabel list berbentuk *flexigrid* untuk mengelola sub program PRJMD yang sudah dipilih.
2. Kembali ke basic flow langkah 3 dengan berganti menjadi pengelolaan sub program RPJMD.

6. Error Flow

E-1. Data RPJMD yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke basic flow langkah ke 7.

E-2. Data RPJMD yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data RPJMD tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.
2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.

8. Post-Conditions

1. Data di dalam database telah *terupdate*.

4.3 Use Case Spesification: Mengelola RPJMDes

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan RPJMDes (Rencana Pembangunan Jangka Menengah Desa). Aktor dapat melakukan tambah, edit, hapus, dan cari data RPJMDes.

2. Primary Aktor

1. Perencana Pembangunan

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	29/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan RPJMDes.
2. Sistem menampilkan antarmuka pengelolaan RPJMDes berupa data table *flexigrid*.
3. Aktor memilih tambah data RPJMDes.
 - A-1. Aktor memilih edit data RPJMDes.
 - A-2. Aktor memilih hapus data RPJMDes.
 - A-3. Aktor memilih cari data RPJMDes.
 - A-4. Aktor memilih menambah sub program RPJMDes.
4. Sistem akan menampilkan form isian untuk menambahkan RPJMDes.
5. Aktor menginputkan data RPJMDes.
6. Setelah selesai menginputkan data RPJMDes, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data RPJMDes yang telah diinputkan
 - E-1. Data RPJMDes yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.
9. Use Case selesai.

5. Alternative Flow

- A-1. Aktor memilih edit data RPJMDes.
 1. Sistem akan menampilkan form untuk mengubah data RPJMDes yang sudah dipilih aktor.
 2. Aktor mengubah data RPJMDes yang sudah dipilih.
 3. Setelah selesai mengubah data RPJMDes, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	30/ 68
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. Sistem melakukan pengecekan data RPJMDes yang telah diubah

E-2. Data RPJMDes yang diubah aktor salah.

5. Sistem menyimpan data ke database.

6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data RPJMDes.

1. Aktor memilih data RPJMDes yang ingin dihapus.

2. Sistem melakukan pengecekan data RPJMDes yang ingin dihapus.

E-3. Data RPJMDes tidak bisa dihapus.

3. Sistem menghapus data yang di database.

4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data RPJMDes.

1. Aktor memilih tombol pencarian data RPJMDes.

2. Kemudian, aktor mengetik data RPJMDes yang ingin dicari berdasarkan atribut yang sudah disediakan.

3. Sistem menampilkan data yang ingin dicari aktor.

4. Berlanjut ke basic flow langkah ke 8.

A-4. Aktor memilih menambah sub program RPJMDes.

1. Sistem menampilkan antarmuka Detail RPJMDes berupa tabel list berbentuk *flexigrid* untuk mengelola sub program RPJMDes yang sudah dipilih.

2. Kembali ke basic flow langkah 3 dengan berganti menjadi pengelolaan sub program RPJMDes.

6. Error Flow

E-1. Data RPJMD yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.

2. Kembali ke basic flow langkah ke 7.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	31/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

E-2. Data RPJMD yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.

2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data RPJMD tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.

2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.

2. Aktor telah melakukan pengelolaan data RPJMD.

3. Aktor telah memasuki sistem.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.4 Use Case Spesification: Mengelola Detail RPJMDes

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan Detail RPJMDes (Rencana Pembangunan Jangka Menengah Desa). Aktor dapat melakukan tambah, edit, hapus, dan cari data Detail RPJMDes.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan Detail RPJMDes.

2. Sistem menampilkan antarmuka pengelolaan Detail RPJMDes data table berbentuk *flexigrid*.

3. Aktor memilih tambah data RPJMDes.

A-1. Aktor memilih edit data Detail RPJMDes.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	32/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

A-2. Aktor memilih hapus data Detail RPJMDes.

A-3. Aktor memilih cari data Detail RPJMDes.

A-4. Aktor memilih menambah sub program detail RPJMDes.

4. Sistem akan menampilkan form isian untuk menambahkan Detail RPJMDes.

5. Aktor menginputkan data Detail RPJMDes.

6. Setelah selesai menginputkan data Detail RPJMDes, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.

7. Sistem melakukan pengecekan data Detail RPJMDes yang telah diinputkan

E-1. Data Detail RPJMDes salah.

8. Sistem menyimpan data ke database.

9. Use Case selesai.

5. Alternative Flow

A-1. Aktor memilih edit data Detail RPJMDes.

1. Sistem akan menampilkan form untuk mengubah data Detail RPJMDes yang sudah dipilih aktor.

2. Aktor mengubah data Detail RPJMDes yang sudah dipilih.

3. Setelah selesai mengubah data Detail RPJMDes, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.

4. Sistem melakukan pengecekan data Detail RPJMDes yang telah diubah

E-2. Data Detail RPJMDes yang diubah salah.

5. Sistem menyimpan data ke database.

6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data Detail RPJMDes.

1. Aktor memilih data Detail RPJMDes yang ingin dihapus.

2. Sistem melakukan pengecekan data Detail RPJMDes yang ingin dihapus.

E-3. Data Detail RPJMDes tidak bisa dihapus.

3. Sistem menghapus data yang di database.

4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data Detail RPJMDes.

1. Aktor memilih tombol pencarian data Detail RPJMDes.

2. Kemudian, aktor mengetik data Detail RPJMDes yang ingin dicari berdasarkan atribut yang sudah disediakan.

3. Sistem menampilkan data yang ingin dicari aktor.

4. Berlanjut ke basic flow langkah ke 8.

A-4. Aktor memilih menambah sub program Detail RPJMDes.

1. Sistem menampilkan antarmuka Detail RPJMDes berupa tabel list berbentuk *flexigrid* untuk mengelola sub program Detail RPJMDes yang sudah dipilih.

2. Kembali ke basic flow langkah 3 dengan berganti menjadi pengelolaan sub program Detail RPJMDes.

6. Error Flow

E-1. Data Detail RPJMD yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.

2. Kembali ke basic flow langkah ke 7.

E-2. Data Detail RPJMD yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.

2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data Detail RPJMD tidak bisa dihapus.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	34/ 68
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 memberikan pesan peringatan bahwa data yang dihapus tidak bisa.
2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.
3. Aktor telah melakukan pengelolaan data RPJMD.
4. Aktor telah melakukan pengelolaan data RPJMDes.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.5 Use Case Spesification: Mencetak Detil RPJMDes

1. Brief Description

Use Case ini digunakan oleh aktor untuk mencetak Detil RPJMDes.

2. Primary Aktor

2. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih menu Detil RPJMDes.
2. Aktor memilih periode yang akan ditampilkan terlebih dahulu.
3. Sistem akan menampilkan data detil RPJMDes sesuai dengan periode yang dipilih aktor.
4. Aktor memilih aksi cetak Excel.
5. Sistem memberikan cetakan data detil RPJMDes sesuai periode berupa file excel.
6. Use Case selesai.

5. Alternative Flow

None

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	35/ 68
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		

6. Error Flow

None

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.
3. Aktor telah melakukan pengelolaan RPJMDes.
4. Aktor telah melakukan pengelolaan Detil RPJMDes.

8. Post-Conditions

1. Memberikan hasil data Detil RPJMDes berupa file excel.

4.6 Use Case Spesification: Mengelola RKPDes

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan RKPDes (Rencana Kerja Pemerintah Desa). Aktor dapat melakukan tambah, edit, hapus, dan cari data RKPDes.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan RKPDes.
2. Sistem menampilkan antarmuka pengelolaan RKPDes berupa data table *flexigrid*.
3. Aktor memilih tambah data RKPDes.
 - A-1. Aktor memilih edit data RKPDes.
 - A-2. Aktor memilih hapus data RKPDes.
 - A-3. Aktor memilih cari data RKPDes.
4. Sistem akan menampilkan form isian untuk menambahkan RKPDes.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	36/ 68
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. Aktor menginputkan data RKPDes.
6. Setelah selesai menginputkan data RKPDes, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data RKPDes yang telah diinputkan
 - E-1.** Data RKPDes yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.
9. Use Case selesai.

5. Alternative Flow

A-1. Aktor memilih edit data RKPDes.

1. Sistem akan menampilkan form untuk mengubah data RKPDes yang sudah dipilih aktor.
2. Aktor mengubah data RKPDes yang sudah dipilih.
3. Setelah selesai mengubah data RKPDes, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.
4. Sistem melakukan pengecekan data RKPDes yang telah diubah
 - E-2.** Data RKPDes yang diubah aktor salah.
5. Sistem menyimpan data ke database.
6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data RKPDes.

1. Aktor memilih data RKPDes yang ingin dihapus.
2. Sistem melakukan pengecekan data RKPDes yang ingin dihapus.
 - E-3.** Data RKPDes tidak bisa dihapus.
3. Sistem menghapus data yang di database.
4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data RKPDes.

1. Aktor memilih tombol pencarian data RKPDes.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	37/ 68
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. Kemudian, aktor mengetik data RKPDes yang ingin dicari berdasarkan atribut yang sudah disediakan.
3. Sistem menampilkan data yang ingin dicari aktor.
4. Berlanjut ke basic flow langkah ke 8.

6. Error Flow

E-1. Data RKPDes yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke basic flow langkah ke 7.

E-2. Data RKPDes yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data RKPDes tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.
2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.
3. Aktor telah melakukan pengelolaan RPJMDes.
4. Aktor telah melakukan pengelolaan Detil RPJMDes

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.7 Use Case Spesification: Mengelola RABDes

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan RABDes (Rencana Anggaran Biaya Desa). Aktor

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	38/ 68
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		

dapat melakukan tambah, edit, hapus, dan cari data RABDes.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan RABDes.
2. Sistem menampilkan antarmuka pengelolaan RABDes berupa data table *flexigrid*.
3. Aktor memilih tambah data RABDes.
 - A-1. Aktor memilih edit data RABDes.
 - A-2. Aktor memilih hapus data RABDes.
 - A-3. Aktor memilih cari data RABDes.
 - A-4. Aktor memilih cetak RABDes.
4. Sistem akan menampilkan form isian untuk menambahkan RABDes.
5. Aktor menginputkan data RABDes.
6. Setelah selesai menginputkan data RABDes, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data RABDes yang telah diinputkan
 - E-1. Data RABDes yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.
9. Use Case selesai.

5. Alternative Flow

- A-1. Aktor memilih edit data RABDes.
 1. Sistem akan menampilkan form untuk mengubah data RABDes yang sudah dipilih aktor.
 2. Aktor mengubah data RABDes yang sudah dipilih.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	39/ 68
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. Setelah selesai mengubah data RABDes, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.

4. Sistem melakukan pengecekan data RABDes yang telah diubah.

E-2. Data RABDes yang diubah aktor salah.

5. Sistem menyimpan data ke database.

6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data RABDes.

1. Aktor memilih data RABDes yang ingin dihapus.

2. Sistem melakukan pengecekan data RABDes yang ingin dihapus.

E-3. Data RABDes tidak bisa dihapus.

3. Sistem menghapus data yang di database.

4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data RABDes.

1. Aktor memilih tombol pencarian data RABDes.

2. Kemudian, aktor mengetik data RABDes yang ingin dicari berdasarkan atribut yang sudah disediakan.

3. Sistem menampilkan data yang ingin dicari aktor.

4. Berlanjut ke basic flow langkah ke 8.

A-4. Aktor memilih cetak RABdes.

1. Aktor memilih tombol cetak RABDes sesuai data RABDes yang dipilih.

2. Sistem akan menampilkan tampilan berupa *pop-up* hasil cetakan RABDes yang dipilih actor.

3. Aktor memilih tombol simpan untuk mendownload file RABDes yang dipilih berupa file pdf.

4. Berlanjut ke basic flow langkah ke 9.

6. Error Flow

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	40/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

E-1. Data RABDes yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke basic flow langkah ke 7.

E-2. Data RABDes yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data RABDes tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.
2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.8 Use Case Spesification: Mengelola RABDes Anggaran

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan RABDes (Rencana Anggaran Biaya Desa) Anggaran. Aktor dapat melakukan tambah, edit, hapus, dan cari data RABDes Anggaran.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan RABDes Anggaran.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	41/ 68
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 antarmuka pengelolaan RABDes Anggaran sesuai dengan RABDes yang dipilih berupa data table *flexigrid*.
3. Aktor memilih tambah data RABDes Anggaran.
 - A-1.** Aktor memilih edit data RABDes Anggaran.
 - A-2.** Aktor memilih hapus data RABDes Anggaran.
 - A-3.** Aktor memilih cari data RABDes Anggaran.
4. Sistem akan menampilkan form isian untuk menambahkan RABDes Anggaran.
5. Aktor menginputkan data RABDes Anggaran.
6. Setelah selesai menginputkan data RABDes Anggaran, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data RABDes Anggaran yang telah diinputkan
 - E-1.** Data RABDes Anggaran yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.
9. Use Case selesai.

5. Alternative Flow

- A-1.** Aktor memilih edit data RABDes Anggaran.
 1. Sistem akan menampilkan form untuk mengubah data RABDes Anggaran yang sudah dipilih aktor.
 2. Aktor mengubah data RABDes Anggaran yang sudah dipilih.
 3. Setelah selesai mengubah data RABDes Anggaran, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.
 4. Sistem melakukan pengecekan data RABDes Anggaran yang telah diubah.
 - E-2.** Data RABDes Anggaran yang diubah aktor salah.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	42/ 68
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. Sistem menyimpan data ke database.

6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data RABDes Anggaran.

1. Aktor memilih data RABDes Anggaran yang ingin dihapus.

2. Sistem melakukan pengecekan data RABDes Anggaran yang ingin dihapus.

E-3. Data RABDes Anggaran tidak bisa dihapus.

3. Sistem menghapus data yang di database.

4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data RABDes Anggaran.

1. Aktor memilih tombol pencarian data RABDes Anggaran.

2. Kemudian, aktor mengetik data RABDes Anggaran yang ingin dicari berdasarkan atribut yang sudah disediakan.

3. Sistem menampilkan data yang ingin dicari aktor.

4. Berlanjut ke basic flow langkah ke 8.

6. Error Flow

E-1. Data RABDes Anggaran yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.

2. Kembali ke basic flow langkah ke 7.

E-2. Data RABDes Anggaran yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.

2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data RABDes Anggaran tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.

2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.
3. Aktor telah melakukan pengelolaan RABDes.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.9 Use Case Spesification: Mengelola SPP

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan SPP (Surat Permintaan Pembayaran). Aktor dapat melakukan tambah SPP, edit SPP, hapus SPP, dan cari data SPP.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan SPP.
2. Sistem menampilkan antarmuka pengelolaan SPP berupa data table *flexigrid*.
3. Aktor memilih tambah data SPP.
 - A-1. Aktor memilih edit data SPP.
 - A-2. Aktor memilih hapus data SPP.
 - A-3. Aktor memilih cari data SPP.
 - A-4. Aktor memilih cetak SPP.
4. Sistem akan menampilkan form isian untuk menambahkan SPP.
5. Aktor menginputkan data SPP.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	44/ 68
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		

6. Setelah selesai menginputkan data SPP, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.

7. Sistem melakukan pengecekan data SPP yang telah diinputkan

E-1. Data SPP yang diinputkan aktor salah.

8. Sistem menyimpan data ke database.

9. Use Case selesai.

5. Alternative Flow

A-1. Aktor memilih edit data SPP.

1. Sistem akan menampilkan form untuk mengubah data SPP yang sudah dipilih aktor.

2. Aktor mengubah data SPP yang sudah dipilih.

3. Setelah selesai mengubah data SPP, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.

4. Sistem melakukan pengecekan data SPP yang telah diubah.

E-2. Data SPP yang diubah aktor salah.

5. Sistem menyimpan data ke database.

6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data SPP.

1. Aktor memilih data SPP yang ingin dihapus.

2. Sistem melakukan pengecekan data SPP yang ingin dihapus.

E-3. Data SPP tidak bisa dihapus.

3. Sistem menghapus data yang di database.

4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data SPP.

1. Aktor memilih tombol pencarian data SPP.

2. Kemudian, aktor mengetik data SPP yang ingin dicari berdasarkan atribut yang sudah disediakan.
3. Sistem menampilkan data yang ingin dicari aktor.
4. Berlanjut ke basic flow langkah ke 8.

A-4. Aktor memilih cetak SPP.

1. Aktor memilih tombol cetak SPP sesuai data SPP yang dipilih.
2. Sistem akan menampilkan tampilan berupa *pop-up* hasil cetakan SPP yang dipilih aktor.
3. Aktor memilih tombol simpan untuk mendownload file SPP yang dipilih berupa file pdf.
4. Berlanjut ke basic flow langkah ke 9.

6. Error Flow

E-1. Data SPP yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke basic flow langkah ke 7.

E-2. Data SPP yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data SPP tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.
2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.

8. Post-Conditions

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	46/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Data di dalam database telah *terupdate*.

4.10 Use Case Spesification: Mengelola Detail SPP

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan Detail SPP (Surat Permintaan pembayaran). Aktor dapat melakukan tambah detil SPP, edit detil SPP, hapus detil SPP, dan cari data detil SPP.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan detil SPP.
2. Sistem menampilkan antarmuka pengelolaan detil SPP sesuai dengan SPP yang dipilih berupa data table *flexigrid*.
3. Aktor memilih tambah data detil SPP.
 - A-1.** Aktor memilih edit data detil SPP.
 - A-2.** Aktor memilih hapus data detil SPP.
 - A-3.** Aktor memilih cari data detil SPP.
4. Sistem akan menampilkan form isian untuk menambahkan detil SPP.
5. Aktor menginputkan data detil SPP.
6. Setelah selesai menginputkan data detil SPP, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data detil SPP yang telah diinputkan
 - E-1.** Data detil SPP yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.
9. Use Case selesai.

5. Alternative Flow

A-1. Aktor memilih edit data detil SPP.

1. Sistem akan menampilkan form untuk mengubah data detil SPP yang sudah dipilih aktor.
2. Aktor mengubah data detil SPP yang sudah dipilih.
3. Setelah selesai mengubah data detil SPP, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.
4. Sistem melakukan pengecekan data detil SPP yang telah diubah.

E-2. Data detil SPP yang diubah aktor salah.

5. Sistem menyimpan data ke database.
6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data detil SPP.

1. Aktor memilih data detil SPP yang ingin dihapus.
2. Sistem melakukan pengecekan data detil SPP yang ingin dihapus.

E-3. Data detil SPP tidak bisa dihapus.

3. Sistem menghapus data yang di database.
4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data detil SPP.

1. Aktor memilih tombol pencarian data detil SPP.
2. Kemudian, aktor mengetik data detil SPP yang ingin dicari berdasarkan atribut yang sudah disediakan.
3. Sistem menampilkan data yang ingin dicari aktor.
4. Berlanjut ke basic flow langkah ke 8.

6. Error Flow

E-1. Data detil SPP yang diinputkan aktor salah.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	48/ 68
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 memberikan pesan peringatan bahwa data yang diinputkan salah.

2. Kembali ke basic flow langkah ke 7.

E-2. Data detil SPP yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.

2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data detil SPP tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.

2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.

2. Aktor telah memasuki sistem.

3. Aktor telah melakukan pengelolaan SPP.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.11 Use Case Spesification: Mengelola LPJ

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan LPJ (Laporan Penanggungjawaban). Aktor dapat melakukan tambah LPJ, edit LPJ, hapus LPJ, dan cari data LPJ.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan LPJ.

2. Sistem menampilkan antarmuka pengelolaan LPJ berupa data table *flexigrid*.
3. Aktor memilih tambah data LPJ.
 - A-1.** Aktor memilih edit data LPJ.
 - A-2.** Aktor memilih hapus data LPJ.
 - A-3.** Aktor memilih cari data LPJ.
 - A-4.** Aktor memilih cetak LPJ.
4. Sistem akan menampilkan form isian untuk menambahkan LPJ.
5. Aktor menginputkan data LPJ.
6. Setelah selesai menginputkan data LPJ, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data LPJ yang telah diinputkan
 - E-1.** Data LPJ yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.
9. Use Case selesai.

5. Alternative Flow

- A-1.** Aktor memilih edit data LPJ.
 1. Sistem akan menampilkan form untuk mengubah data LPJ yang sudah dipilih aktor.
 2. Aktor mengubah data LPJ yang sudah dipilih.
 3. Setelah selesai mengubah data LPJ, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.
 4. Sistem melakukan pengecekan data LPJ yang telah diubah.
 - E-2.** Data LPJ yang diubah aktor salah.
 5. Sistem menyimpan data ke database.
 6. Berlanjut ke basic flow langkah ke 8.
- A-2.** Aktor memilih hapus data LPJ.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	50/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Aktor memilih data LPJ yang ingin dihapus.
2. Sistem melakukan pengecekan data LPJ yang ingin dihapus.

E-3. Data LPJ tidak bisa dihapus.

3. Sistem menghapus data yang di database.
4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data LPJ.

1. Aktor memilih tombol pencarian data LPJ.
2. Kemudian, aktor mengetik data LPJ yang ingin dicari berdasarkan atribut yang sudah disediakan.
3. Sistem menampilkan data yang ingin dicari aktor.
4. Berlanjut ke basic flow langkah ke 8.

A-4. Aktor memilih cetak LPJ.

1. Aktor memilih tombol cetak LPJ sesuai data LPJ yang dipilih.
2. Sistem akan menampilkan tampilan berupa *pop-up* hasil cetakan LPJ yang dipilih actor.
3. Aktor memilih tombol simpan untuk mendownload file LPJ yang dipilih berupa file pdf.
4. Berlanjut ke basic flow langkah ke 9.

6. Error Flow

E-1. Data LPJ yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke basic flow langkah ke 7.

E-2. Data LPJ yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data LPJ tidak bisa dihapus.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	51/ 68
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 memberikan pesan peringatan bahwa data yang dihapus tidak bisa.
2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.12 Use Case Spesification: Mengelola Detail LPJ

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan Detail LPJ (Laporan Penanggungjawaban). Aktor dapat melakukan tambah detil LPJ, edit detil LPJ, hapus detil LPJ, dan cari data detil LPJ.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan detil LPJ.
2. Sistem menampilkan antarmuka pengelolaan detil LPJ sesuai dengan LPJ yang dipilih berupa data table *flexigrid*.
3. Aktor memilih tambah data detil LPJ.
 - A-1.** Aktor memilih edit data detil LPJ.
 - A-2.** Aktor memilih hapus data detil LPJ.
 - A-3.** Aktor memilih cari data detil LPJ.
4. Sistem akan menampilkan form isian untuk menambahkan detil LPJ.
5. Aktor menginputkan data detil LPJ.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	52/ 68
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		

6. Setelah selesai menginputkan data detail LPJ, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.

7. Sistem melakukan pengecekan data detail LPJ yang telah diinputkan

E-1. Data detail LPJ yang diinputkan aktor salah.

8. Sistem menyimpan data ke database.

9. Use Case selesai.

5. Alternative Flow

A-1. Aktor memilih edit data detail LPJ.

1. Sistem akan menampilkan form untuk mengubah data detail LPJ yang sudah dipilih aktor.

2. Aktor mengubah data detail LPJ yang sudah dipilih.

3. Setelah selesai mengubah data detail LPJ, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.

4. Sistem melakukan pengecekan data detail LPJ yang telah diubah.

E-2. Data detail LPJ yang diubah aktor salah.

5. Sistem menyimpan data ke database.

6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data detail LPJ.

1. Aktor memilih data detail LPJ yang ingin dihapus.

2. Sistem melakukan pengecekan data detail LPJ yang ingin dihapus.

E-3. Data detail LPJ tidak bisa dihapus.

3. Sistem menghapus data yang di database.

4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data detail LPJ.

1. Aktor memilih tombol pencarian data detail LPJ.

2. Kemudian, aktor mengetik data detail LPJ yang ingin dicari berdasarkan atribut yang sudah disediakan.
3. Sistem menampilkan data yang ingin dicari aktor.
4. Berlanjut ke basic flow langkah ke 8.

6. Error Flow

E-1. Data detail LPJ yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke basic flow langkah ke 7.

E-2. Data detail LPJ yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data detail LPJ tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.
2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.
3. Aktor telah melakukan pengelolaan LPJ.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.13 Use Case Spesification: Mengelola Periode

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan Periode. Aktor dapat melakukan tambah periode, edit periode, hapus periode, dan cari data periode.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	54/ 68
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. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan periode.
2. Sistem menampilkan antarmuka pengelolaan periode berupa data table *flexigrid*.
3. Aktor memilih tambah data periode.
 - A-1. Aktor memilih edit data periode.
 - A-2. Aktor memilih hapus data periode.
 - A-3. Aktor memilih cari data periode.
4. Sistem akan menampilkan form isian untuk menambahkan periode.
5. Aktor menginputkan data periode.
6. Setelah selesai menginputkan data periode, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data periode yang telah diinputkan
 - E-1. Data periode yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.
9. Use Case selesai.

5. Alternative Flow

- A-1. Aktor memilih edit data periode.
 1. Sistem akan menampilkan form untuk mengubah data periode yang sudah dipilih aktor.
 2. Aktor mengubah data periode yang sudah dipilih.
 3. Setelah selesai mengubah data periode, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	55/ 68
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. Sistem melakukan pengecekan data periode yang telah diubah

E-2. Data periode yang diubah aktor salah.

5. Sistem menyimpan data ke database.

6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data periode.

1. Aktor memilih data periode yang ingin dihapus.

2. Sistem melakukan pengecekan data periode yang ingin dihapus.

E-3. Data periode tidak bisa dihapus.

3. Sistem menghapus data yang di database.

4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data periode.

1. Aktor memilih tombol pencarian data periode.

2. Kemudian, aktor mengetik data periode yang ingin dicari berdasarkan atribut yang sudah disediakan.

3. Sistem menampilkan data yang ingin dicari aktor.

4. Berlanjut ke basic flow langkah ke 8.

6. Error Flow

E-1. Data periode yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.

2. Kembali ke basic flow langkah ke 7.

E-2. Data periode yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.

2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data periode tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.

2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.14 Use Case Spesification: Mengelola Tahun Anggaran

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan Tahun Anggaran. Aktor dapat melakukan tambah tahun anggaran, edit tahun anggaran, hapus tahun anggaran, dan cari data tahun anggaran.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan tahun anggaran.
2. Sistem menampilkan antarmuka pengelolaan tahun anggaran berupa data table *flexigrid*.
3. Aktor memilih tambah data tahun anggaran.
 - A-1.** Aktor memilih edit data tahun anggaran.
 - A-2.** Aktor memilih hapus data tahun anggaran.
 - A-3.** Aktor memilih cari data tahun anggaran.
4. Sistem akan menampilkan form isian untuk menambahkan tahun anggaran.
5. Aktor menginputkan data tahun anggaran.
6. Setelah selesai menginputkan data tahun anggaran, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.

7. Sistem melakukan pengecekan data tahun anggaran yang telah diinputkan

E-1. Data tahun anggaran yang diinputkan aktor salah.

8. Sistem menyimpan data ke database.

9. Use Case selesai.

5. Alternative Flow

A-1. Aktor memilih edit data tahun anggaran.

1. Sistem akan menampilkan form untuk mengubah data tahun anggaran yang sudah dipilih aktor.

2. Aktor mengubah data tahun anggaran yang sudah dipilih.

3. Setelah selesai mengubah data tahun anggaran, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.

4. Sistem melakukan pengecekan data tahun anggaran yang telah diubah

E-2. Data tahun anggaran yang diubah aktor salah.

5. Sistem menyimpan data ke database.

6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data tahun anggaran.

1. Aktor memilih data tahun anggaran yang ingin dihapus.

2. Sistem melakukan pengecekan data tahun anggaran yang ingin dihapus.

E-3. Data tahun anggaran tidak bisa dihapus.

3. Sistem menghapus data yang di database.

4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data tahun anggaran.

1. Aktor memilih tombol pencarian data tahun anggaran.

2. Kemudian, aktor mengetik data tahun anggaran yang ingin dicari berdasarkan atribut yang sudah disediakan.
3. Sistem menampilkan data yang ingin dicari aktor.
4. Berlanjut ke basic flow langkah ke 8.

6. Error Flow

E-1. Data tahun anggaran yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke basic flow langkah ke 7.

E-2. Data tahun anggaran yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data tahun anggaran tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.
2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.15 Use Case Spesification: Mengelola Bidang

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan Bidang. Aktor dapat melakukan tambah bidang, edit bidang, hapus bidang, dan cari data bidang.

2. Primary Aktor

1. Perencana Pembangunan

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	59/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan bidang.
2. Sistem menampilkan antarmuka pengelolaan bidang berupa data table *flexigrid*.
3. Aktor memilih tambah data bidang.
 - A-1. Aktor memilih edit data bidang.
 - A-2. Aktor memilih hapus data bidang.
 - A-3. Aktor memilih cari data bidang.
4. Sistem akan menampilkan form isian untuk menambahkan bidang.
5. Aktor menginputkan data bidang.
6. Setelah selesai menginputkan data bidang, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data bidang yang telah diinputkan
 - E-1. Data bidang yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.
9. Use Case selesai.

5. Alternative Flow

- A-1. Aktor memilih edit data bidang.
 1. Sistem akan menampilkan form untuk mengubah data bidang yang sudah dipilih aktor.
 2. Aktor mengubah data bidang yang sudah dipilih.
 3. Setelah selesai mengubah data bidang, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.
 4. Sistem melakukan pengecekan data bidang yang telah diubah

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	60/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

E-2. Data bidang yang diubah aktor salah.

5. Sistem menyimpan data ke database.

6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data bidang.

1. Aktor memilih data bidang yang ingin dihapus.

2. Sistem melakukan pengecekan data bidang yang ingin dihapus.

E-3. Data bidang tidak bisa dihapus.

3. Sistem menghapus data yang di database.

4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data bidang.

1. Aktor memilih tombol pencarian data bidang.

2. Kemudian, aktor mengetik data bidang yang ingin dicari berdasarkan atribut yang sudah disediakan.

3. Sistem menampilkan data yang ingin dicari aktor.

4. Berlanjut ke basic flow langkah ke 8.

6. Error Flow

E-1. Data bidang yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.

2. Kembali ke basic flow langkah ke 7.

E-2. Data bidang yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.

2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data bidang tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.

2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	61/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.16 Use Case Spesification: Mengelola Kode Rekening

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan Kode Rekening. Aktor dapat melakukan tambah kode rekening, edit kode rekening, hapus kode rekening, dan cari data kode rekening.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan kode rekening.
2. Sistem menampilkan antarmuka pengelolaan kode rekening berupa data table *flexigrid*.
3. Aktor memilih tambah data kode rekening.
 - A-1. Aktor memilih edit data kode rekening.
 - A-2. Aktor memilih hapus data kode rekening.
 - A-3. Aktor memilih cari data kode rekening.
4. Sistem akan menampilkan form isian untuk menambahkan kode rekening.
5. Aktor menginputkan data kode rekening.
6. Setelah selesai menginputkan data kode rekening, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data kode rekening yang telah diinputkan

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	62/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- E-1.** Data kode rekening yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.
9. Use Case selesai.

5. Alternative Flow

A-1. Aktor memilih edit data kode rekening.

1. Sistem akan menampilkan form untuk mengubah data kode rekening yang sudah dipilih aktor.
2. Aktor mengubah data kode rekening yang sudah dipilih.
3. Setelah selesai mengubah data kode rekening, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.
4. Sistem melakukan pengecekan data kode rekening yang telah diubah

E-2. Data kode rekening yang diubah aktor salah.

5. Sistem menyimpan data ke database.
6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data kode rekening.

1. Aktor memilih data kode rekening yang ingin dihapus.
2. Sistem melakukan pengecekan data kode rekening yang ingin dihapus.

E-3. Data kode rekening tidak bisa dihapus.

3. Sistem menghapus data yang di database.
4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data kode rekening.

1. Aktor memilih tombol pencarian data kode rekening.
2. Kemudian, aktor mengetik data kode rekening yang ingin dicari berdasarkan atribut yang sudah disediakan.

3. Sistem menampilkan data yang ingin dicari aktor.
4. Berlanjut ke basic flow langkah ke 8.

6. Error Flow

E-1. Data kode rekening yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke basic flow langkah ke 7.

E-2. Data kode rekening yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data kode rekening tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.
2. Kembali ke alternative flow A-2 langkah ke 2.

7. Pre-Conditions

1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.

8. Post-Conditions

1. Data di dalam database telah terupdate.

4.17 Use Case Spesification: Mengelola Sumber Dana

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pengelolaan Sumber Dana. Aktor dapat melakukan tambah sumber dana, edit sumber dana, hapus sumber dana, dan cari data sumber dana.

2. Primary Aktor

1. Perencana Pembangunan

3. Supporting Aktor

None

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	64/ 68
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 aktor memilih untuk melakukan pengelolaan sumber dana.
2. Sistem menampilkan antarmuka pengelolaan sumber dana berupa data table *flexigrid*.
3. Aktor memilih tambah data sumber dana.
 - A-1.** Aktor memilih edit data sumber dana.
 - A-2.** Aktor memilih hapus data sumber dana.
 - A-3.** Aktor memilih cari data sumber dana.
4. Sistem akan menampilkan form isian untuk menambahkan sumber dana.
5. Aktor menginputkan data sumber dana.
6. Setelah selesai menginputkan data sumber dana, aktor memilih tombol simpan guna menyimpan data yang sudah diinputkan.
7. Sistem melakukan pengecekan data sumber dana yang telah diinputkan
 - E-1.** Data sumber dana yang diinputkan aktor salah.
8. Sistem menyimpan data ke database.
9. Use Case selesai.

5. Alternative Flow

- A-1.** Aktor memilih edit data sumber dana.
 1. Sistem akan menampilkan form untuk mengubah data sumber dana yang sudah dipilih aktor.
 2. Aktor mengubah data sumber dana yang sudah dipilih.
 3. Setelah selesai mengubah data sumber dana, aktor memilih tombol simpan guna menyimpan data yang sudah diubah.
 4. Sistem melakukan pengecekan data sumber dana yang telah diubah
 - E-2.** Data sumber dana yang diubah aktor salah.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	65/ 68
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. Sistem menyimpan data ke database.
6. Berlanjut ke basic flow langkah ke 8.

A-2. Aktor memilih hapus data sumber dana.

1. Aktor memilih data sumber dana yang ingin dihapus.
2. Sistem melakukan pengecekan data sumber dana yang ingin dihapus.

E-3. Data sumber dana tidak bisa dihapus.

3. Sistem menghapus data yang di database.
4. Berlanjut ke basic flow langkah ke 8.

A-3. Aktor memilih cari data sumber dana.

1. Aktor memilih tombol pencarian data sumber dana.
2. Kemudian, aktor mengetik data sumber dana yang ingin dicari berdasarkan atribut yang sudah disediakan.
3. Sistem menampilkan data yang ingin dicari aktor.
4. Berlanjut ke basic flow langkah ke 8.

6. Error Flow

E-1. Data sumber dana yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke basic flow langkah ke 7.

E-2. Data sumber dana yang diubah aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diubah salah.
2. Kembali ke alternative flow A-1 langkah ke 4.

E-2. Data sumber dana tidak bisa dihapus.

1. Sistem memberikan pesan peringatan bahwa data yang dihapus tidak bisa.
2. Kembali ke alternative flow A-2 langkah ke 2.

Program Studi Teknik Informatika	SKPL –SiDeKa-PPD	66/ 68
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

7. Pre-Conditions

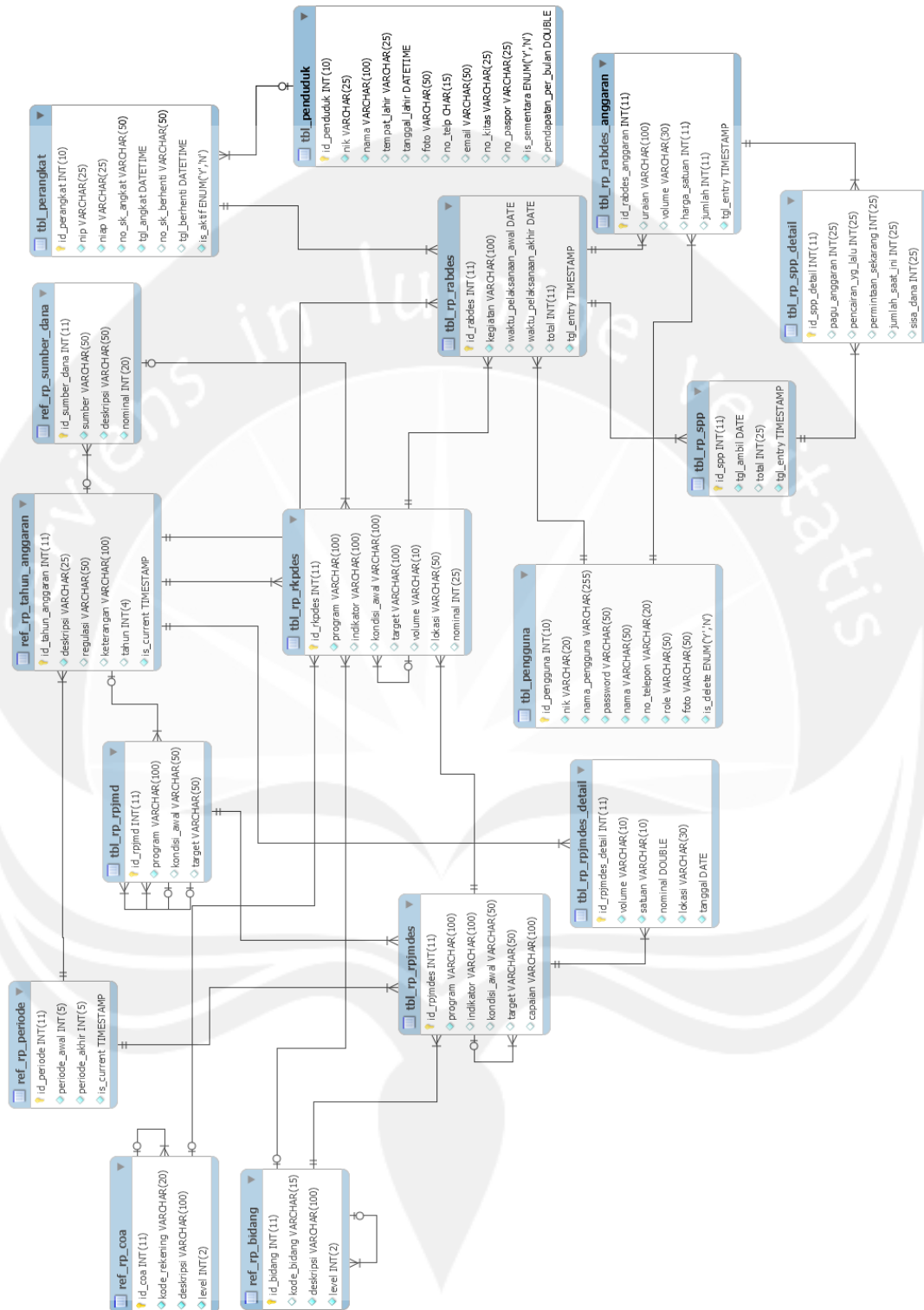
1. Aktor telah melakukan login sebagai perencana pembangunan.
2. Aktor telah memasuki sistem.

8. Post-Conditions

1. Data di dalam database telah terupdate.



5 Entity Relationship Diagram (ERD)



Gambar 3 Entity Relationship Diagram SIDEKa-PPD

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SISTEM INFORMASI DESA DAN KAWASAN UNTUK PERENCANAAN PEMBANGUNAN DESA (SIDEKa-PPD)

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Lucas Sandy Duta Arga / 110706588

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL-SIDEKa-PPD		
		Revisi		1/111

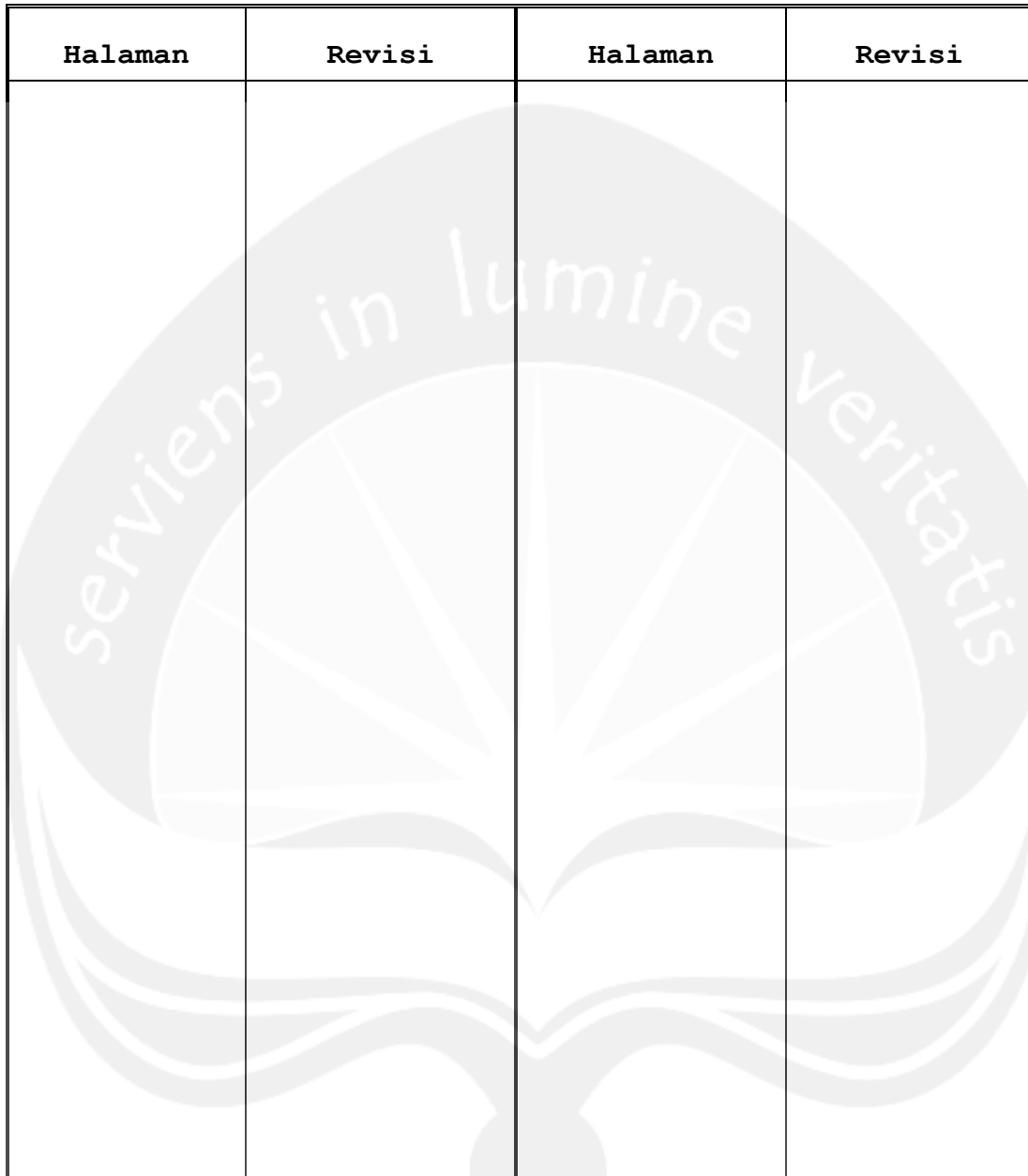
Program Studi Teknik Informatika	DPPL-SIDEKa-PPD	1/111
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	LSDA							
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi
			

DAFTAR ISI

1	Pendahuluan	13
1.1	Tujuan	13
1.2	Lingkup Masalah	13
1.3	Definisi dan Akronim	13
	Code Igniter.....	15
1.4	Referensi	15
2	Analisis Model	15
2.1	Perancangan Arsitektur	15
2.2	Perancangan Rinci	19
2.2.1	Sequence Diagram	19
2.2.1.1	Login	19
2.2.1.2	Ubah Password	20
2.2.1.3	Pengelolaan RPJMD	20
2.2.1.3.1	Tampil Data RPJMD.....	20
2.2.1.3.2	Tambah Data RPJMD.....	20
2.2.1.3.3	Ubah Data RPJMD.....	21
2.2.1.3.4	Hapus Data RPJMD.....	21
2.2.1.3.5	Tampil Sub Program RPJMD.....	22
2.2.1.3.6	Tambah Sub Program RPJMD.....	22
2.2.1.3.7	Hapus Sub Program RPJMD.....	22
2.2.1.4	Pengelolaan RPJMDes	23
2.2.1.4.1	Tampil Data RPJMDes.....	23
2.2.1.4.2	Tambah Data RPJMDes.....	23
2.2.1.4.3	Ubah Data RPJMDes.....	24
2.2.1.4.4	Tampil Sub Program RPJMDes.....	24
2.2.1.4.5	Tambah Sub Program RPJMDes.....	24
2.2.1.4.6	Hapus Sub Program RPJMDes.....	25
2.2.1.4.7	Tampil Data Detil RPJMDes.....	25
2.2.1.4.8	Tambah Data Detil RPJMDes.....	25
2.2.1.4.9	Ubah Data Detil RPJMDes.....	26
2.2.1.4.10	Hapus Data Detil RPJMDes.....	26
2.2.1.5	Pengelolaan Detil RPJMDes	26
2.2.1.5.1	Tampil Detil RPJMDes By Periode.....	26
2.2.1.5.2	Cetak Detil RPJMDes By Periode.....	27
2.2.1.6	Pengelolaan RKPDes	27
2.2.1.6.1	Tampil Data RKPDes.....	27
2.2.1.6.2	Tambah Data RKPDes.....	27

2.2.1.6.3	Ubah Data RKPDes	28
2.2.1.6.4	Hapus Data RKPDes	28
2.2.1.6.5	Tampil Data Sub Program RKPDes	29
2.2.1.6.6	Tambah Data Sub Program RKPDes	29
2.2.1.6.7	Hapus Sub Program RKPDes	29
2.2.1.7	Pengelolaan RABDes	30
2.2.1.7.1	Tampil Data RABDes	30
2.2.1.7.2	Tambah Data RABDes	30
2.2.1.7.3	Ubah Data RABDes	30
2.2.1.7.4	Hapus Data RABDes	31
2.2.1.7.5	Cetak Data RABDes	31
2.2.1.8	Pengelolaan RABDes Anggaran	32
2.2.1.8.1	Tampil Data RABDes Anggaran	32
2.2.1.8.2	Tambah Data RABDes Anggaran	32
2.2.1.8.3	Ubah Data RABDes Anggaran	32
2.2.1.8.4	Hapus Data RABDes Anggaran	32
2.2.1.9	Pengelolaan SPP	33
2.2.1.9.1	Tampil Data SPP	33
2.2.1.9.2	Tambah Data SPP	33
2.2.1.9.3	Ubah Data SPP	34
2.2.1.9.4	Hapus Data SPP	34
2.2.1.9.5	Cetak Data SPP	34
2.2.1.10	Pengelolaan Detil SPP	35
2.2.1.10.1	Tampil Data Detil SPP	35
2.2.1.10.2	Tambah Data Detil SPP	35
2.2.1.10.3	Ubah Data Detil SPP	36
2.2.1.10.4	Hapus Data Detil SPP	36
2.2.1.11	Pengelolaan LPJ	37
2.2.1.11.1	Tampil Data LPJ	37
2.2.1.11.2	Tambah Data LPJ	37
2.2.1.11.3	Ubah Data LPJ	37
2.2.1.11.4	Hapus Data LPJ	38
2.2.1.11.5	Cetak Data LPJ	38
2.2.1.12	Pengelolaan Detil LPJ	39
2.2.1.12.1	Tampil Data Detil LPJ	39
2.2.1.12.2	Tambah Data Detil LPJ	39
2.2.1.12.3	Ubah Data Detil LPJ	40
2.2.1.12.4	Hapus Data Detil LPJ	40

2.2.1.13	Pengelolaan Periode	41
2.2.1.13.1	Tampil Data Periode.....	41
2.2.1.13.2	Tambah Data Periode.....	41
2.2.1.13.3	Ubah Data Periode.....	41
2.2.1.13.4	Hapus Data Periode.....	42
2.2.1.14	Pengelolaan Tahun Anggaran	42
2.2.1.14.1	Tampil Data Tahun Anggaran.....	42
2.2.1.14.2	Tambah Data Tahun Anggaran.....	42
2.2.1.14.3	Ubah Data Tahun Anggaran.....	43
2.2.1.14.4	Hapus Data Tahun Anggaran.....	43
2.2.1.15	Pengelolaan Bidang	44
2.2.1.15.1	Tampil Data Bidang.....	44
2.2.1.15.2	Tambah Data Bidang.....	44
2.2.1.15.3	Ubah Data Bidang.....	44
2.2.1.15.4	Hapus Data Bidang.....	45
2.2.1.16	Pengelolaan Kode Rekening	45
2.2.1.16.1	Tampil Data Kode Rekening.....	45
2.2.1.16.2	Tambah Data Kode Rekening.....	46
2.2.1.16.3	Ubah Data Kode Rekening.....	46
2.2.1.16.4	Hapus Data Kode Rekening.....	47
2.2.1.17	Pengelolaan Sumber Dana	47
2.2.1.17.1	Tampil Data Sumber Dana.....	47
2.2.1.17.2	Tambah Data Sumber Dana.....	47
2.2.1.17.3	Ubah Data Sumber Dana.....	48
2.2.1.17.4	Hapus Data Sumber Dana.....	48
2.2.2	Class Diagram	50
2.2.3	Class Diagram Specific Description	55
2.2.3.1	Specific Design Class v_login	55
2.2.3.2	Specific Design Class v_change_pass	55
2.2.3.3	Specific Design Class rpjmd/v_list	55
2.2.3.4	Specific Design Class rpjmd/v_tambah	55
2.2.3.5	Specific Design Class rpjmd/v_ubah	56
2.2.3.6	Specific Design Class rpjmd/v_tampil_detil_program 56	
2.2.3.7	Specific Design Class rpjmd/v_tambah_sub_program	56
2.2.3.8	Specific Design Class rpjmdes/v_list	57
2.2.3.9	Specific Design Class rpjmdes/v_tambah	57
2.2.3.10	Specific Design Class rpjmdes/v_ubah	57

2.2.3.11	Specific Design Class rpjmdes/ v_tampil_detil_program	58
2.2.3.12	Specific Design Class rpjmdes/v_tambah_sub_program	58
2.2.3.13	Specific Design Class rpjmdes_detail/v_list ...	58
2.2.3.14	Specific Design Class rpjmdes_detail/ v_list_rpjmdes_detail	58
2.2.3.15	Specific Design Class rkpdes/v_list	58
2.2.3.16	Specific Design Class rkpdes/v_tambah	59
2.2.3.17	Specific Design Class rkpdes/v_ubah	59
2.2.3.18	Specific Design Class rkpdes/ v_tampil_detil_program	59
2.2.3.19	Specific Design Class rkpdes/v_tambah_sub_program 60	
2.2.3.20	Specific Design Class rabdes/v_list	60
2.2.3.21	Specific Design Class rabdes/v_tambah	60
2.2.3.22	Specific Design Class rabdes/v_ubah	61
2.2.3.23	Specific Design Class rabdes/ v_tampil_rabdes_anggaran	61
2.2.3.24	Specific Design Class rabdes/ v_tambah_rabdes_anggaran	61
2.2.3.25	Specific Design Class rabdes/ v_ubah_rabdes_anggaran	62
2.2.3.26	Specific Design Class spp/v_list	62
2.2.3.27	Specific Design Class spp/v_tambah	62
2.2.3.28	Specific Design Class spp/v_ubah	62
2.2.3.29	Specific Design Class spp/v_tampil_spp_detail .	63
2.2.3.30	Specific Design Class spp/v_tambah_spp_detail .	63
2.2.3.31	Specific Design Class spp/v_ubah_spp_detail ...	63
2.2.3.32	Specific Design Class lpj/v_list	63
2.2.3.33	Specific Design Class lpj/v_tambah	64
2.2.3.34	Specific Design Class lpj/v_ubah	64
2.2.3.35	Specific Design Class lpj/v_tampil_lpj_detail .	64
2.2.3.36	Specific Design Class lpj/v_tambah_lpj_detail .	65
2.2.3.37	Specific Design Class lpj/v_ubah_lpj_detail ...	65
2.2.3.38	Specific Design Class periode/v_list	65
2.2.3.39	Specific Design Class periode/v_tambah	65
2.2.3.40	Specific Design Class periode/v_ubah	66
2.2.3.41	Specific Design Class tahun_anggaran/v_list ...	66
2.2.3.42	Specific Design Class tahun_anggaran/v_tambah .	66

2.2.3.43	Specific Design Class tahun_anggaran/v_ubah ...	67
2.2.3.44	Specific Design Class bidang/v_list	67
2.2.3.45	Specific Design Class bidang/v_tambah	67
2.2.3.46	Specific Design Class bidang/v_ubah	68
2.2.3.47	Specific Design Class kode_rekening/v_list	68
2.2.3.48	Specific Design Class kode_rekening/v_tambah ..	68
2.2.3.49	Specific Design Class kode_rekening/v_ubah	68
2.2.3.50	Specific Design Class sumber_dana/v_list	69
2.2.3.51	Specific Design Class sumber_dana/v_tambah	69
2.2.3.52	Specific Design Class sumber_dana/v_ubah	69
2.2.3.53	Specific Design Class c_login	70
2.2.3.54	Specific Design Class c_changePass	70
2.2.3.55	Specific Design Class c_rpjmd	70
2.2.3.56	Specific Design Class c_rpjmDES	71
2.2.3.57	Specific Design Class c_rpjmDES_detail	73
2.2.3.58	Specific Design Class c_rkpDES	73
2.2.3.59	Specific Design Class c_rabDES	74
2.2.3.60	Specific Design Class c_spp	76
2.2.3.61	Specific Design Class c_lpj	77
2.2.3.62	Specific Design Class c_periode	79
2.2.3.63	Specific Design Class c_tahun_anggaran	79
2.2.3.64	Specific Design Class c_bidang	80
2.2.3.65	Specific Design Class c_kode_rekening	81
2.2.3.66	Specific Design Class c_sumber_dana	81
2.2.3.67	Specific Design Class m_login	82
2.2.3.68	Specific Design Class m_user	82
2.2.3.69	Specific Design Class m_rpjmd	83
2.2.3.70	Specific Design Class m_rpjmDES	83
2.2.3.71	Specific Design Class m_rpjmDES_detail	84
2.2.3.72	Specific Design Class m_rkpDES	84
2.2.3.73	Specific Design Class m_rabDES	84
2.2.3.74	Specific Design Class m_spp	85
2.2.3.75	Specific Design Class m_lpj	86
2.2.3.76	Specific Design Class m_periode	87
2.2.3.77	Specific Design Class m_tahun_anggaran	88
2.2.3.78	Specific Design Class m_bidang	88
2.2.3.79	Specific Design Class m_kode_rekening	89
2.2.3.80	Specific Design Class m_sumber_dana	89

3	Perancangan Data	90
3.1	Dekomposisi Data	90
3.1.1	Deskripsi Entitas Data ref_rp_periode	90
3.1.2	Deskripsi Entitas Data ref_rp_tahun_anggaran	90
3.1.3	Deskripsi Entitas Data ref_rp_bidang	90
3.1.4	Deskripsi Entitas Data ref_rp_coa	91
3.1.5	Deskripsi Entitas Data ref_rp_sumber_dana	91
3.1.6	Deskripsi Entitas Data tbl_rp_rpjmd	92
3.1.7	Deskripsi Entitas Data tbl_rp_rpjmdes	92
3.1.8	Deskripsi Entitas Data tbl_rp_rpjmdes_detail	93
3.1.9	Deskripsi Entitas Data tbl_rp_rkpdes	94
3.1.10	Deskripsi Entitas Data tbl_rp_rabdes	95
3.1.11	Deskripsi Entitas Data tbl_rp_rabdes_anggaran	95
3.1.12	Deskripsi Entitas Data tbl_rp_spp	96
3.1.13	Deskripsi Entitas Data tbl_rp_spp_detail	96
3.1.14	Deskripsi Entitas Data tbl_rp_lpj	97
3.1.15	Deskripsi Entitas Data tbl_rp_lpj_detail	97
3.2	Physical Data Model	98
4	Perancangan Antarmuka	99
4.1	Antarmuka Login	99
4.2	Antarmuka Ubah Kata Sandi	99
4.3	Antarmuka Tampil Data	100
4.4	Antarmuka Pengelolaan RPJMD	100
4.5	Antarmuka Pengelolaan RPJMDes	101
4.6	Antarmuka Pengelolaan Detil RPJMDes	102
4.7	Antarmuka Pengelolaan RKPDes	102
4.8	Antarmuka Pengelolaan RABDes	103
4.9	Antarmuka Pengelolaan RABDes Anggaran	104
4.10	Antarmuka Pengelolaan SPP	105
4.11	Antarmuka Pengelolaan Detil SPP	106
4.12	Antarmuka Pengelolaan LPJ	106
4.13	Antarmuka Pengelolaan Detil LPJ	107
4.14	Antarmuka Pengelolaan Periode	108
4.15	Antarmuka Pengelolaan Tahun Anggaran	108
4.16	Antarmuka Pengelolaan Bidang	109
4.17	Antarmuka Pengelolaan Kode Rekening	110
4.18	Antarmuka Pengelolaan Sumber Dana	110

Daftar Gambar

Gambar 2.1 Perancangan Arsitektur SDeKa-PPD.....	19
Gambar 2.2 Sequence Diagram: Login.....	19
Gambar 2.3 Sequence Diagram: Ubah Password.....	20
Gambar 2.4 Sequence Diagram: Tampil Data RPJMD.....	20
Gambar 2.5 Sequence Diagram: Tambah Data RPJMD.....	21
Gambar 2.6 Sequence Diagram: Ubah Data RPJMD.....	21
Gambar 2.7 Sequence Diagram: Hapus Data RPJMD.....	22
Gambar 2.8 Sequence Diagram: Tampil Sub Program RPJMD.....	22
Gambar 2.9 Sequence Diagram: Tambah Sub Program RPJMD.....	22
Gambar 2.10 Sequence Diagram: Hapus Sub Program RPJMD.....	23
Gambar 2.11 Sequence Diagram: Tampil Data RPJMDes.....	23
Gambar 2.12 Sequence Diagram: Tambah Data RPJMDes.....	24
Gambar 2.13 Sequence Diagram: Ubah Data RPJMDes.....	24
Gambar 2.14 Sequence Diagram: Tampil Sub Program RPJMDes.....	24
Gambar 2.15 Sequence Diagram: Tambah Sub Program RPJMDes.....	25
Gambar 2.16 Sequence Diagram: Hapus Sub Program RPJMDes.....	25
Gambar 2.17 Sequence Diagram: Tampil Data Detil RPJMDes.....	25
Gambar 2.18 Sequence Diagram: Tambah Data Detil RPJMDes.....	26
Gambar 2.19 Sequence Diagram: Ubah Data Detil RPJMDes.....	26
Gambar 2.20 Sequence Diagram: Hapus Data Detil RPJMDes.....	26
Gambar 2.21 Sequence Diagram: Tampil Detil RPJMDes By Periode..	27
Gambar 2.22 Sequence Diagram: Cetak Detil RPJMDes By Periode..	27
Gambar 2.23 Sequence Diagram: Tampil Data RKPDes.....	27
Gambar 2.24 Sequence Diagram: Tambah Data RKPDes.....	28
Gambar 2.25 Sequence Diagram: Ubah Data RKPDes.....	28
Gambar 2.26 Sequence Diagram: Hapus Data RKPDes.....	29
Gambar 2.27 Sequence Diagram: Tampil Data Sub Program RKPDes..	29
Gambar 2.28 Sequence Diagram: Tambah Data Sub Program RKPDes..	29
Gambar 2.29 Sequence Diagram: Hapus Sub Program RKPDes.....	29
Gambar 2.30 Sequence Diagram: Tampil Data RABDes.....	30
Gambar 2.31 Sequence Diagram: Tambah Data RABDes.....	30
Gambar 2.32 Sequence Diagram: Ubah Data RABDes.....	31
Gambar 2.33 Sequence Diagram: Hapus Data RABDes.....	31
Gambar 2.34 Sequence Diagram: Cetak Data RABDes.....	31

Gambar 2.35	Sequence Diagram: Tampil Data RABDes Anggaran.....	32
Gambar 2.36	Sequence Diagram: Tambah Data RABDes Anggaran.....	32
Gambar 2.37	Sequence Diagram: Ubah Data RABDes Anggaran.....	32
Gambar 2.38	Sequence Diagram: Hapus Data RABDes Anggaran.....	33
Gambar 2.39	Sequence Diagram: Tampil Data SPP.....	33
Gambar 2.40	Sequence Diagram: Tambah Data SPP.....	34
Gambar 2.41	Sequence Diagram: Ubah Data SPP.....	34
Gambar 2.42	Sequence Diagram: Hapus Data SPP.....	34
Gambar 2.43	Sequence Diagram: Cetak Data SPP.....	35
Gambar 2.44	Sequence Diagram: Tampil Data Detil SPP.....	35
Gambar 2.45	Sequence Diagram: Tambah Data Detil SPP.....	36
Gambar 2.46	Sequence Diagram: Ubah Data Detil SPP.....	36
Gambar 2.47	Sequence Diagram: Hapus Data Detil SPP.....	36
Gambar 2.48	Sequence Diagram: Tampil Data LPJ.....	37
Gambar 2.49	Sequence Diagram: Tambah Data LPJ.....	37
Gambar 2.50	Sequence Diagram: Ubah Data LPJ.....	38
Gambar 2.51	Sequence Diagram: Hapus Data LPJ.....	38
Gambar 2.52	Sequence Diagram: Cetak Data LPJ.....	39
Gambar 2.53	Sequence Diagram: Tampil Data Detil LPJ.....	39
Gambar 2.54	Sequence Diagram: Tambah Data Detil LPJ.....	40
Gambar 2.55	Sequence Diagram: Ubah Data Detil LPJ.....	40
Gambar 2.56	Sequence Diagram: Hapus Data Detil LPJ.....	40
Gambar 2.57	Sequence Diagram: Tampil Data Periode.....	41
Gambar 2.58	Sequence Diagram: Tambah Data Periode.....	41
Gambar 2.59	Sequence Diagram: Ubah Data Periode.....	42
Gambar 2.60	Sequence Diagram: Hapus Data Periode.....	42
Gambar 2.61	Sequence Diagram: Tampil Data Tahun Anggaran.....	42
Gambar 2.62	Sequence Diagram: Tambah Data Tahun Anggaran.....	43
Gambar 2.63	Sequence Diagram: Ubah Data Tahun Anggaran.....	43
Gambar 2.64	Sequence Diagram: Hapus Data Tahun Anggaran.....	43
Gambar 2.65	Sequence Diagram: Tampil Data Bidang.....	44
Gambar 2.66	Sequence Diagram: Tambah Data Bidang.....	44
Gambar 2.67	Sequence Diagram: Ubah Data Bidang.....	45
Gambar 2.68	Sequence Diagram: Hapus Data Bidang.....	45
Gambar 2.69	Sequence Diagram: Tampil Data Kode Rekening.....	46

Gambar 2.70 Sequence Diagram: Tambah Data Kode Rekening.....	46
Gambar 2.71 Sequence Diagram: Ubah Data Kode Rekening.....	46
Gambar 2.72 Sequence Diagram: Hapus Data Bidang.....	47
Gambar 2.73 Sequence Diagram: Tampil Data Sumber Dana.....	47
Gambar 2.74 Sequence Diagram: Tambah Data Sumber Dana.....	48
Gambar 2.75 Sequence Diagram: Ubah Data Sumber Dana.....	48
Gambar 2.76 Sequence Diagram: Hapus Data Sumber Dana.....	48
Gambar 2.77 Class Diagram.....	54
Gambar 3.1 Physical Data Model.....	98
Gambar 4.1 Antarmuka Login.....	99
Gambar 4.2 Antarmuka Ubah Kata Sandi.....	99
Gambar 4.3 Antarmuka Tampil Data.....	100
Gambar 4.4 Antarmuka Pengelolaan RPJMD.....	101
Gambar 4.5 Antarmuka Pengelolaan RPJMDes.....	102
Gambar 4.6 Antarmuka Pengelolaan Detil RPJMDes.....	102
Gambar 4.7 Antarmuka Pengelolaan RKPDes.....	103
Gambar 4.8 Antarmuka Pengelolaan RABDes.....	104
Gambar 4.9 Antarmuka Pengelolaan RABDes Anggaran.....	105
Gambar 4.10 Antarmuka Pengelolaan SPP.....	106
Gambar 4.11 Antarmuka Pengelolaan SPP Detail.....	106
Gambar 4.12 Antarmuka Pengelolaan LPJ.....	107
Gambar 4.13 Antarmuka Pengelolaan LPJ Detail.....	107
Gambar 4.14 Antarmuka Pengelolaan Periode.....	108
Gambar 4.15 Antarmuka Tahun Anggaran.....	109
Gambar 4.16 Antarmuka Pengelolaan Bidang.....	109
Gambar 4.17 Antarmuka Pengelolaan Kode Rekening.....	110
Gambar 4.18 Antarmuka Pengelolaan Sumber Dana.....	111

1 Pendahuluan

1.1 Tujuan

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

1.2 Lingkup Masalah

Perangkat Lunak ini dikembangkan dengan tujuan:

1. Menangani pengelolaan RPJMD (Rencana Pembangunan Jangka Menengah Daerah).
2. Menangani pengelolaan RPJMDes (Rencana Pembangunan Jangka Menengah Desa).
3. Menangani pengelolaan RKPDes (Rencana Kerja Pemerintah Desa).
4. Menangani pengelolaan RABDes (Rencana Anggaran Belanja Desa).
5. Menangani pengelolaan SPP (Surat Permintaan Pembayaran).
6. Menangani pengelolaan LPJ (Laporan Pertanggungjawaban).
7. Menangani pengelolaan Pustaka Perencanaan meliputi Periode, Tahun Anggaran, Bidang, Kode Rekening, dan Sumber Dana.

Aplikasi ini berjalan pada lingkungan dengan *platform* browser (untuk *web application*).

1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description

Program Studi Teknik Informatika	DPPL-SIDeKa-PPD	13/ 111
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		

	(SDD) . Merupakan deskripsi dari perancangan produk / perangkat lunak yang dikembangkan.
DPPL-SIDeKa-PPD-XXX	Kode yang merepresentasikan kebutuhan pada SIDeKa-PPD (Sistem Informasi Desa dan Kawasan untuk Perencanaan Pembangunan Desa) dimana XXX merupakan nomor fungsi produk.
SIDeKa-PPD	Perangkat lunak pengelolaan web.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.
RPJMD	Rencana Pembangunan Jangka Menengah Daerah.
RPJMDes	Rencana Pembangunan Jangka Menengah Desa.
RKPDes	Rencana Kerja Pemerintah Desa.
RABDes	Rencana Anggaran Biaya Desa
SPP	Surat Permintaan Pembayaran
LPJ	Laporan Penanggungjawaban.
Framework	Suatu struktur konseptua dasar yang digunakan untuk memecahkan atau menangani suatu masalah kompleks.

PHP	Hypertext Preprocessor merupakan Bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML dan banyak dipakai untuk memprogram situs web dinamis.
Code Igniter	Aplikasi open source yang berupa framework dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP.

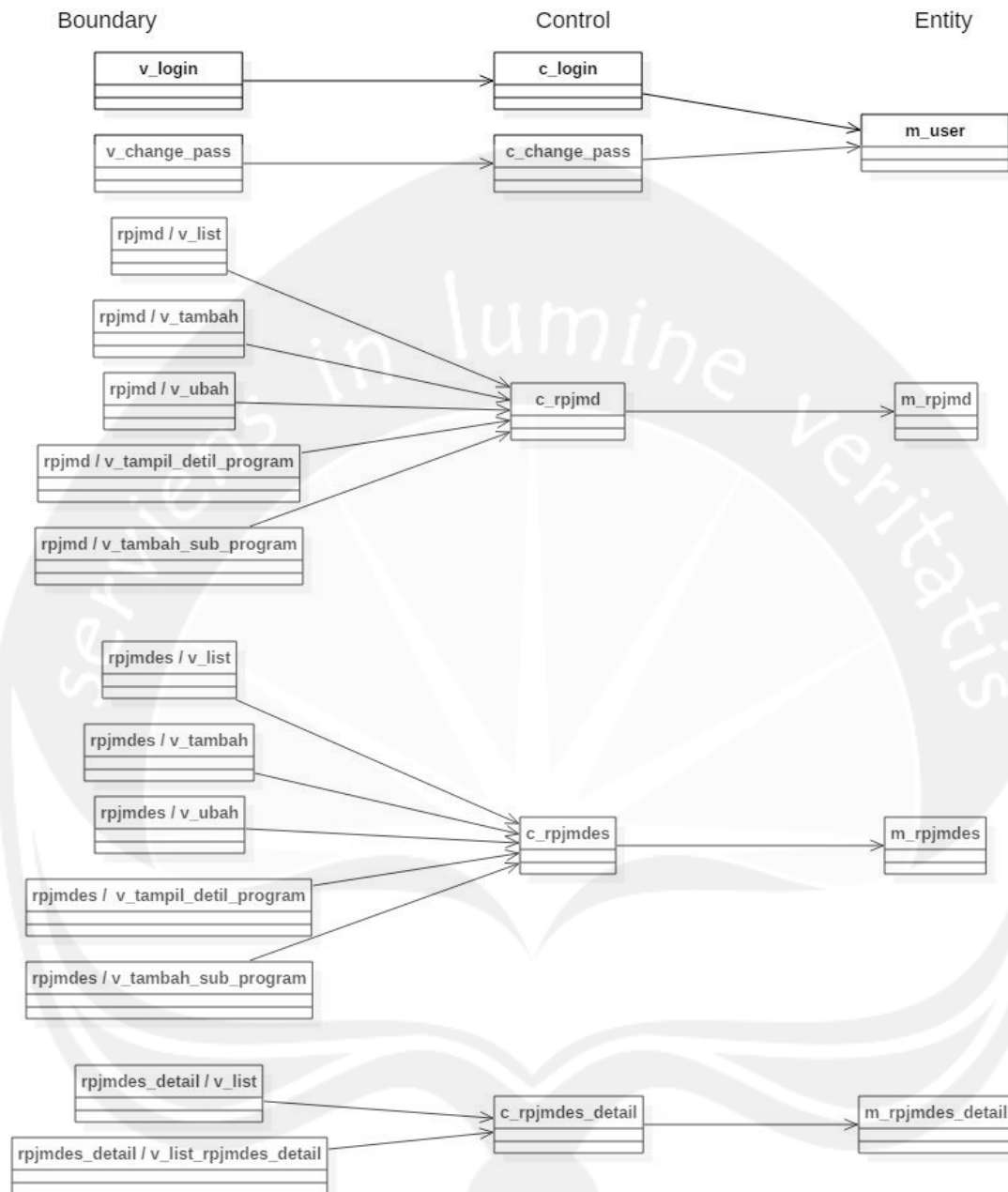
1.4 Referensi

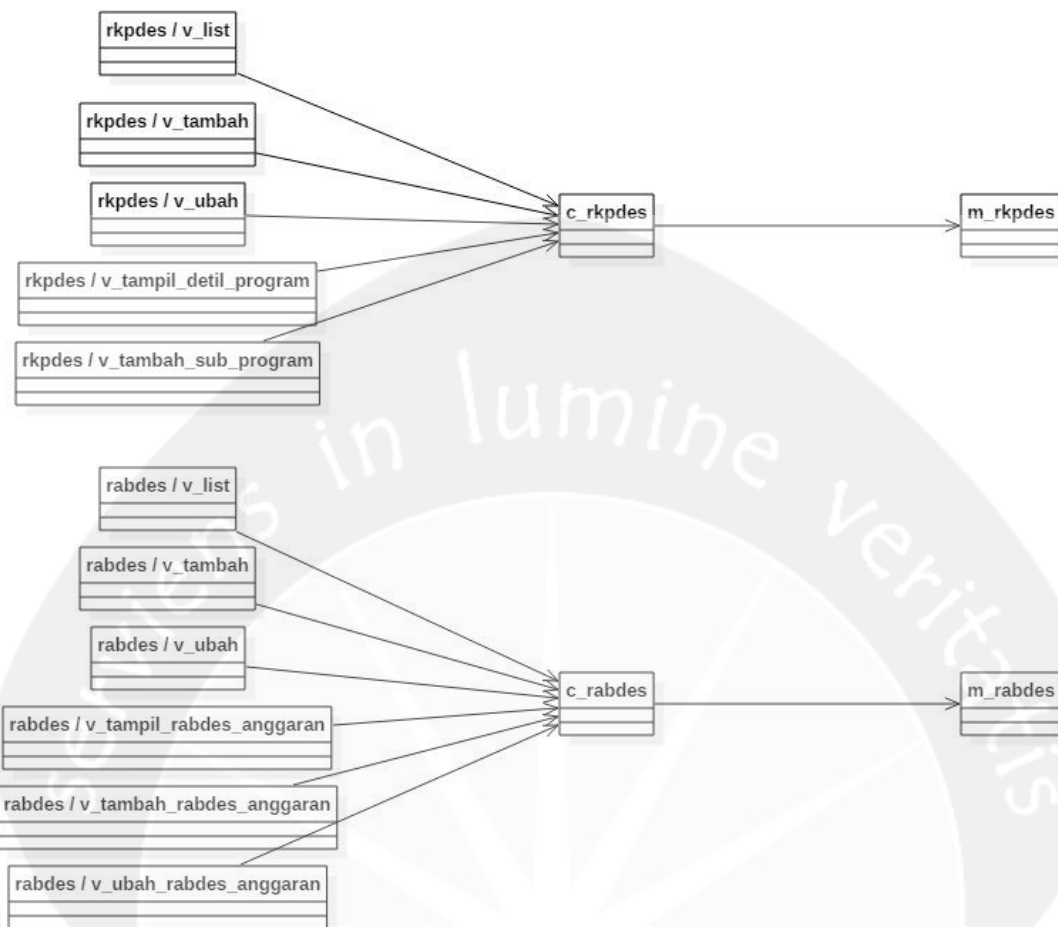
Referensi yang digunakan pada perangkat lunak tersebut adalah:

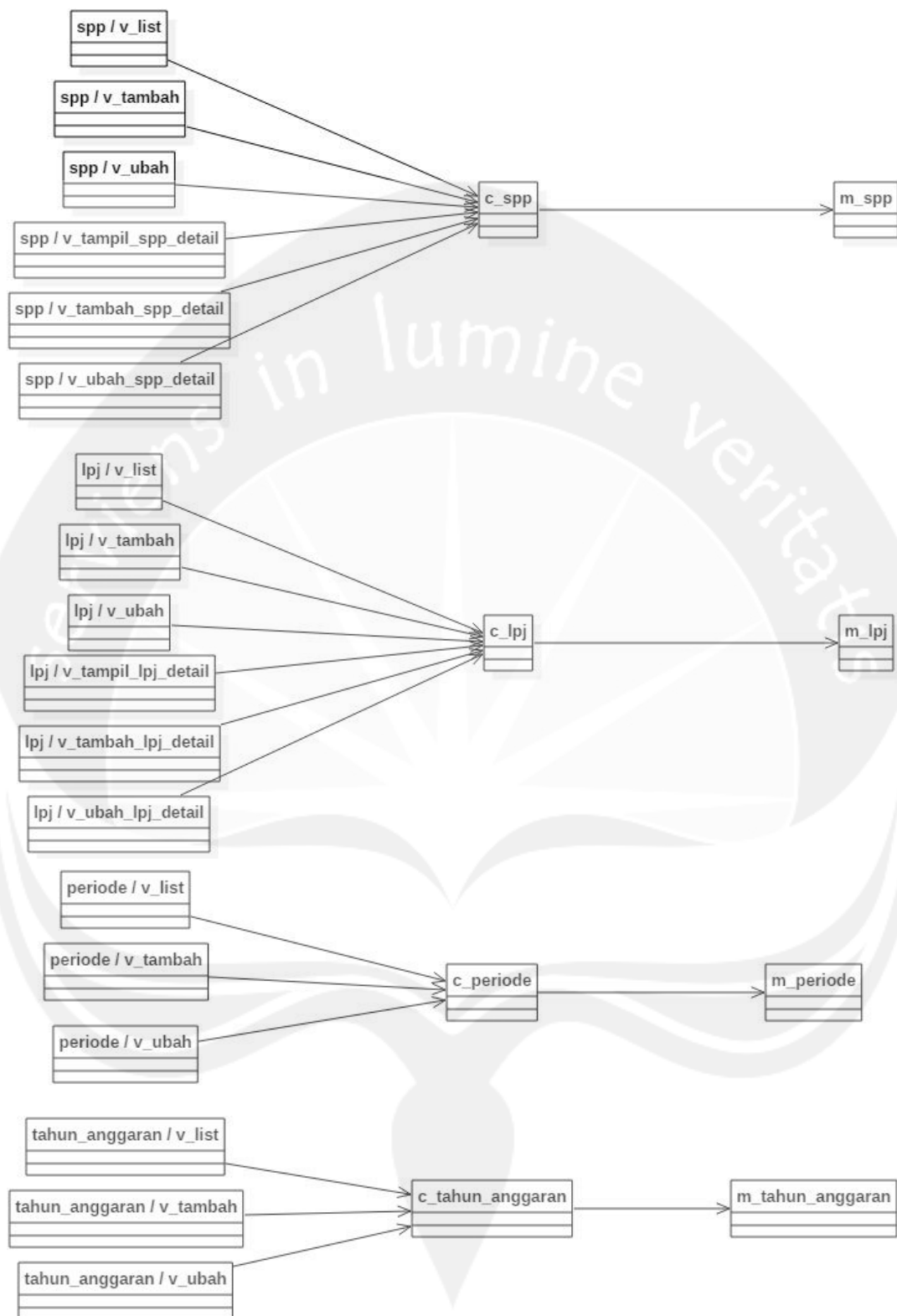
1. Lucas Sandy Duta Arga, *Spesifikasi Kebutuhan Perangkat Lunak (SKPL) SDeKa-PPD*, Universitas Atma Jaya Yogyakarta.
2. Irene Deandra Indarto, *Deskripsi Perancangan Perangkat Lunak (DPPL) SISP*, Universitas Atma Jaya Yogyakarta.

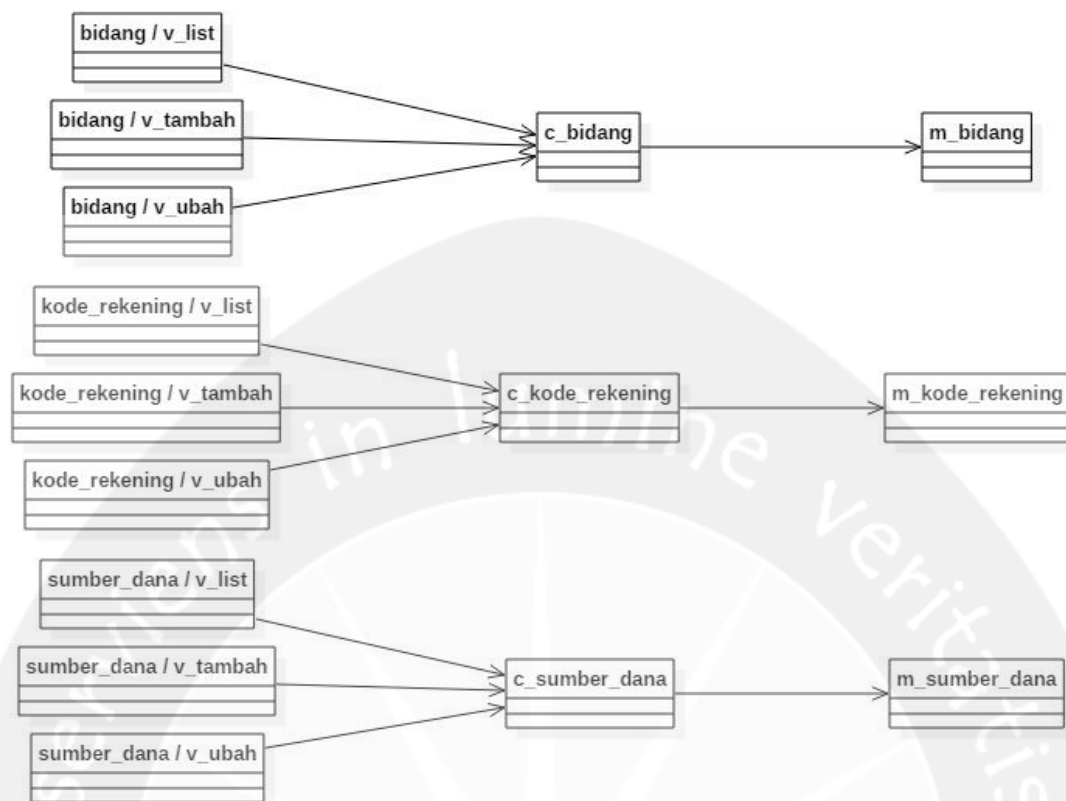
2 Analisis Model

2.1 Perancangan Arsitektur







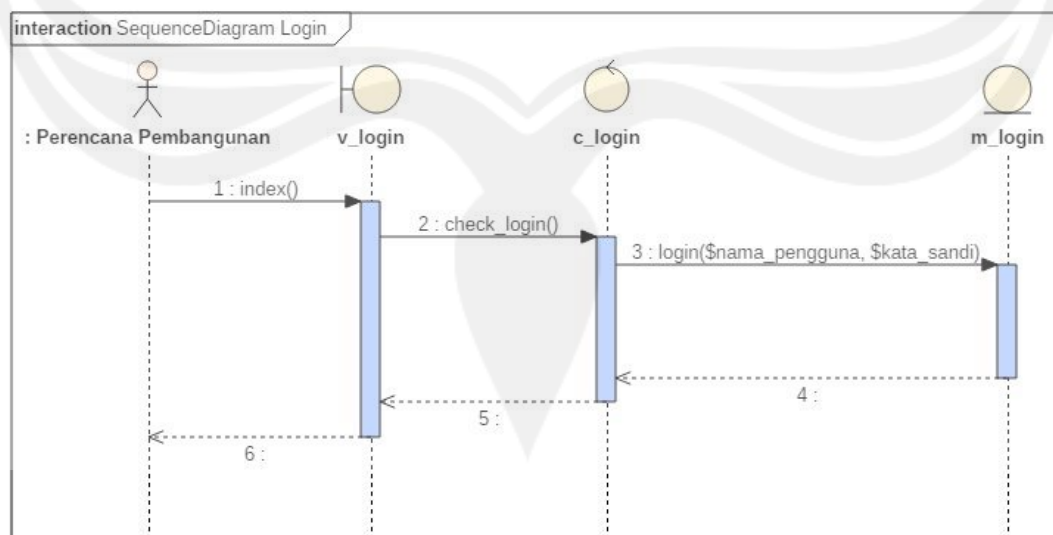


Gambar 2.1 Perancangan Arsitektur SIDEKa-PPD

2.2 Perancangan Rinci

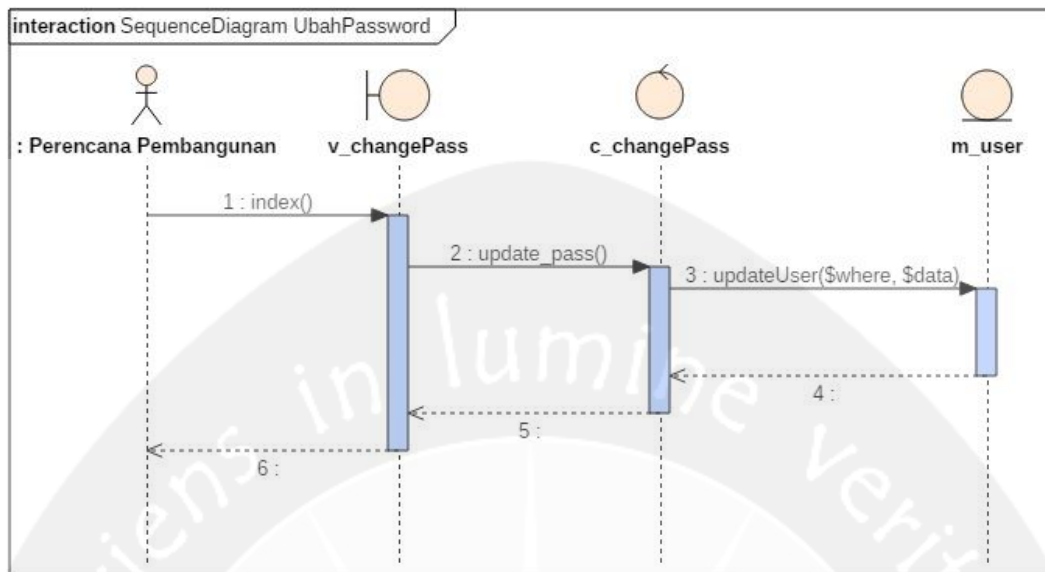
2.2.1 Sequence Diagram

2.2.1.1 Login



Gambar 2.2 Sequence Diagram: Login

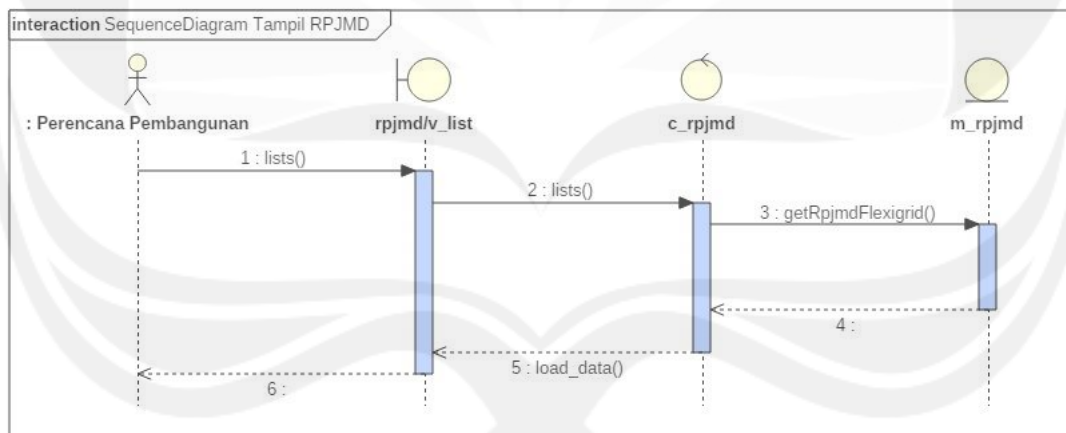
2.2.1.2 Ubah Password



Gambar 2.3 Sequence Diagram: Ubah Password

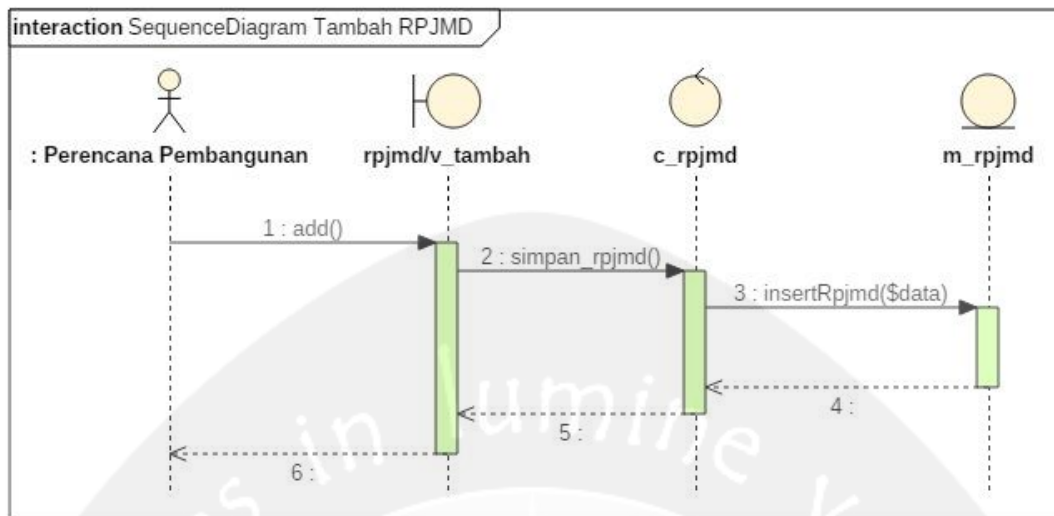
2.2.1.3 Pengelolaan RPJMD

2.2.1.3.1 Tampil Data RPJMD



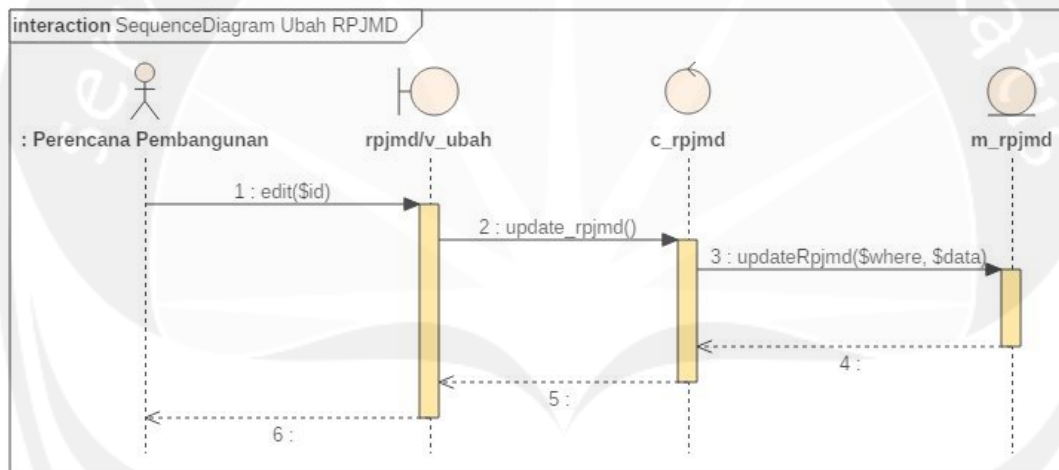
Gambar 2.4 Sequence Diagram: Tampil Data RPJMD

2.2.1.3.2 Tambah Data RPJMD



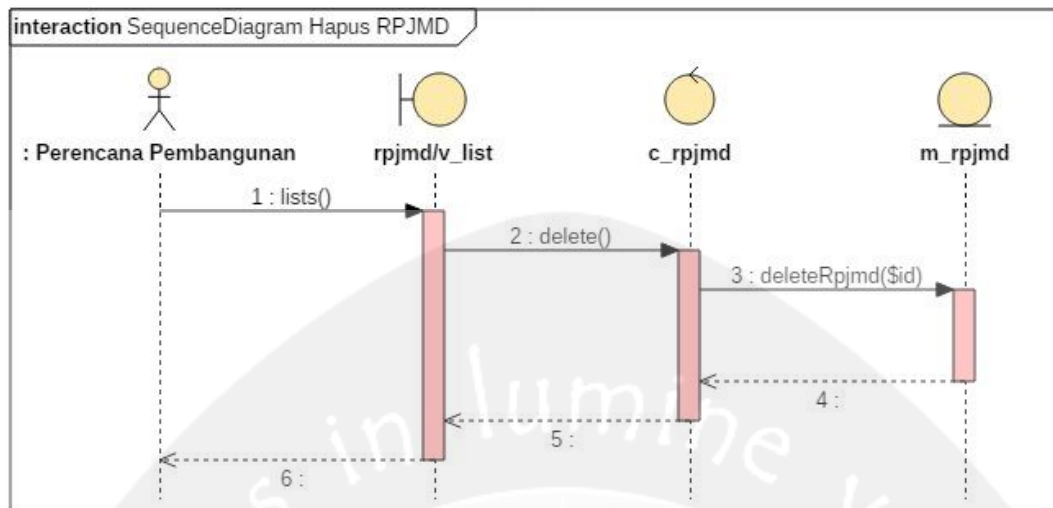
Gambar 2.5 Sequence Diagram: Tambah Data RPJMD

2.2.1.3.3 Ubah Data RPJMD



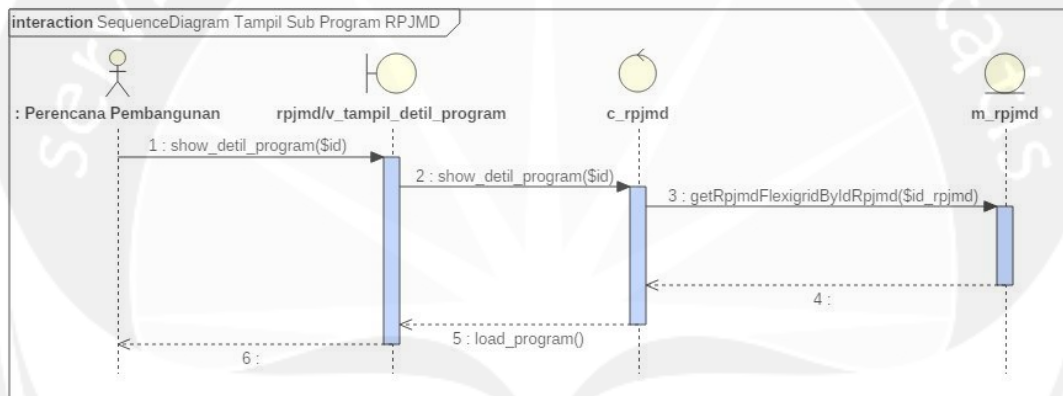
Gambar 2.6 Sequence Diagram: Ubah Data RPJMD

2.2.1.3.4 Hapus Data RPJMD



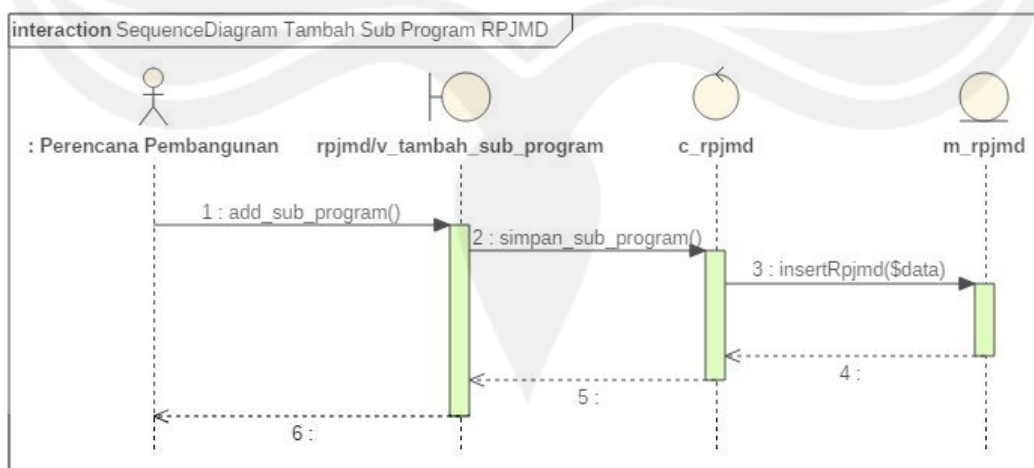
Gambar 2.7 Sequence Diagram: Hapus Data RPJMD

2.2.1.3.5 Tampil Sub Program RPJMD



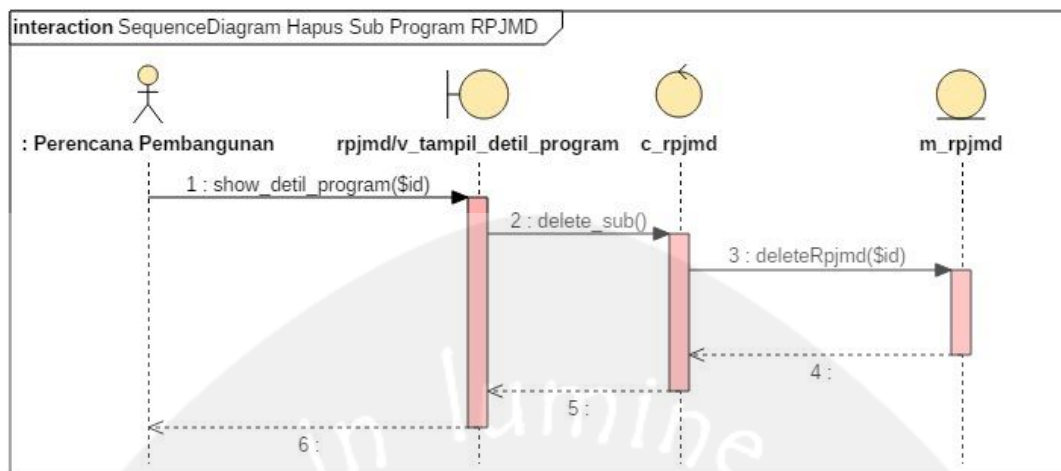
Gambar 2.8 Sequence Diagram: Tampil Sub Program RPJMD

2.2.1.3.6 Tambah Sub Program RPJMD



Gambar 2.9 Sequence Diagram: Tambah Sub Program RPJMD

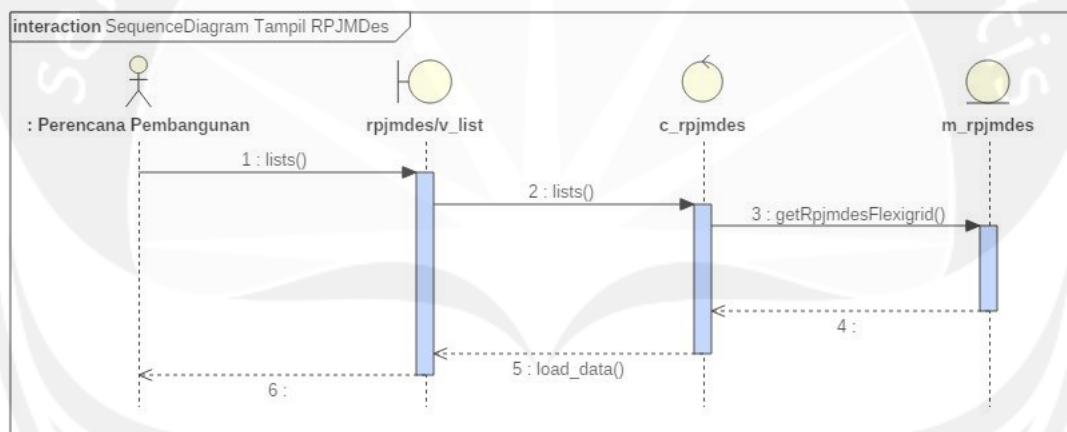
2.2.1.3.7 Hapus Sub Program RPJMD



Gambar 2.10 Sequence Diagram: Hapus Sub Program RPJMD

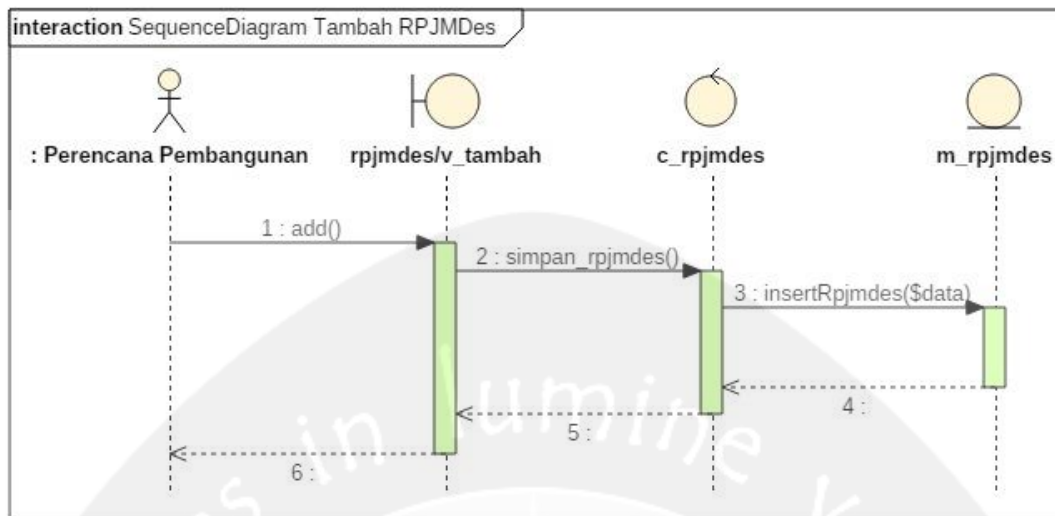
2.2.1.4 Pengelolaan RPJMDes

2.2.1.4.1 Tampil Data RPJMDes



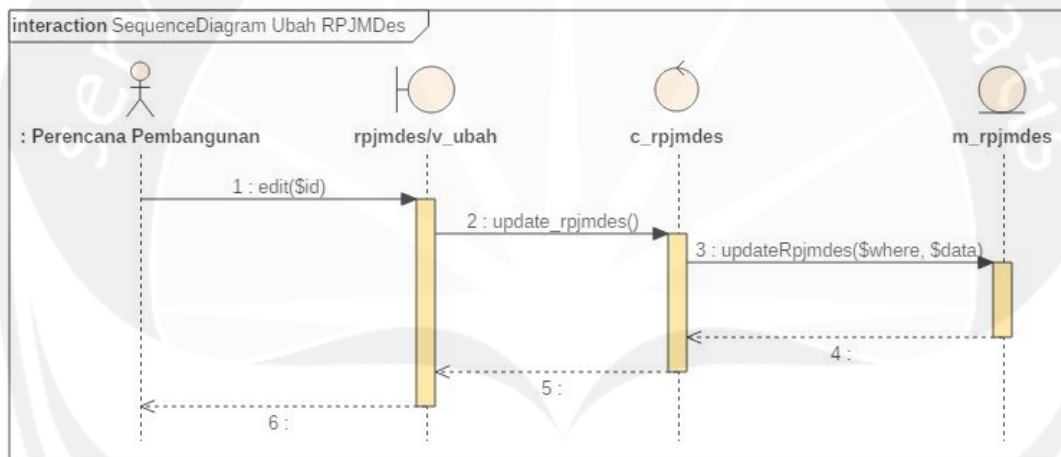
Gambar 2.11 Sequence Diagram: Tampil Data RPJMDes

2.2.1.4.2 Tambah Data RPJMDes



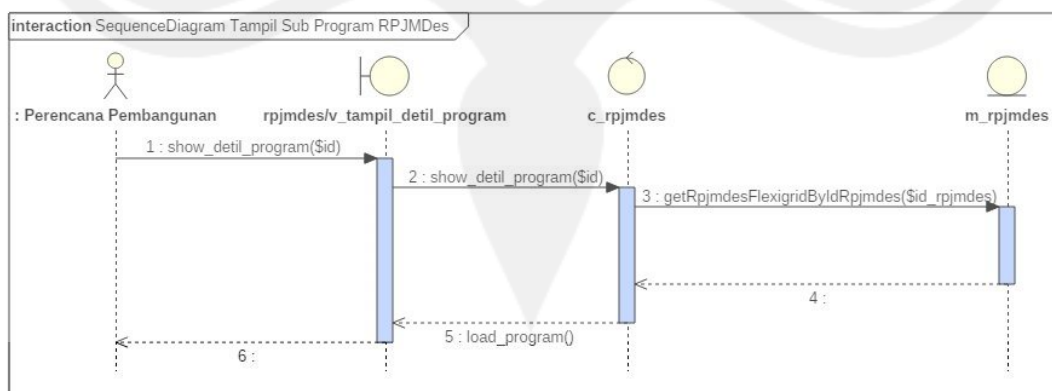
Gambar 2.12 Sequence Diagram: Tambah Data RPJMDes

2.2.1.4.3 Ubah Data RPJMDes



Gambar 2.13 Sequence Diagram: Ubah Data RPJMDes

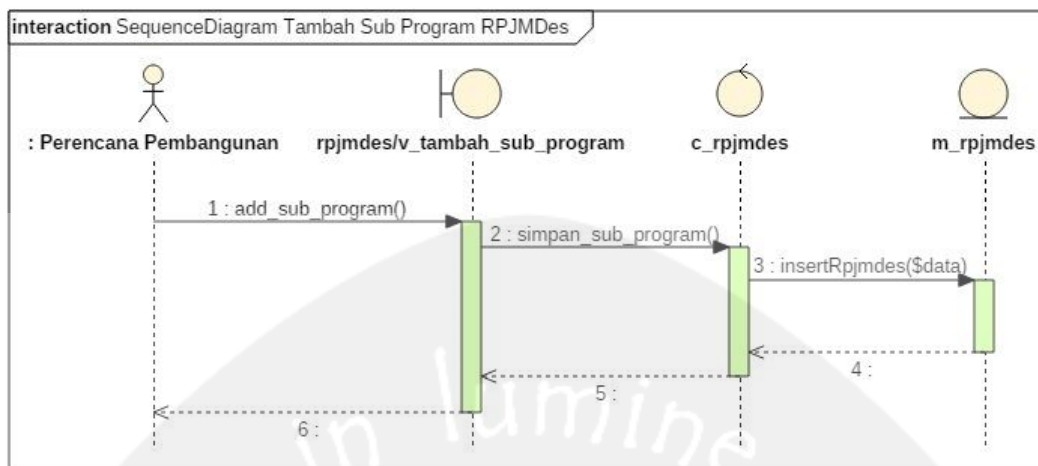
2.2.1.4.4 Tampil Sub Program RPJMDes



Gambar 2.14 Sequence Diagram: Tampil Sub Program RPJMDes

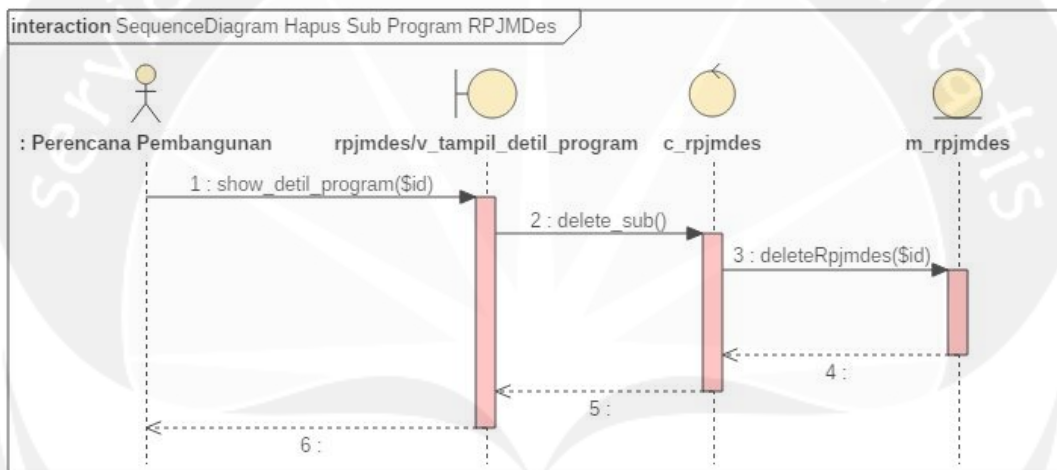
2.2.1.4.5 Tambah Sub Program RPJMDes

Program Studi Teknik Informatika	DPPL–SIDEKa–PPD	24/ 111
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		



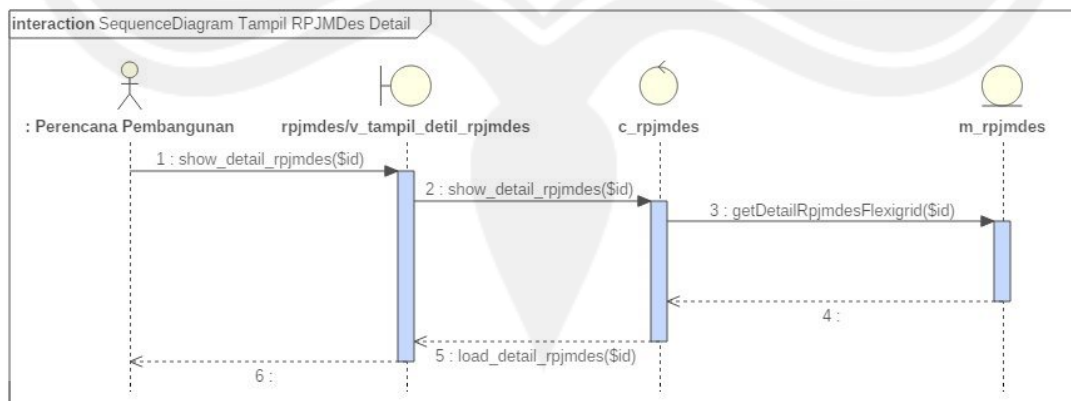
Gambar 2.15 Sequence Diagram: Tambah Sub Program RPJMDes

2.2.1.4.6 Hapus Sub Program RPJMDes



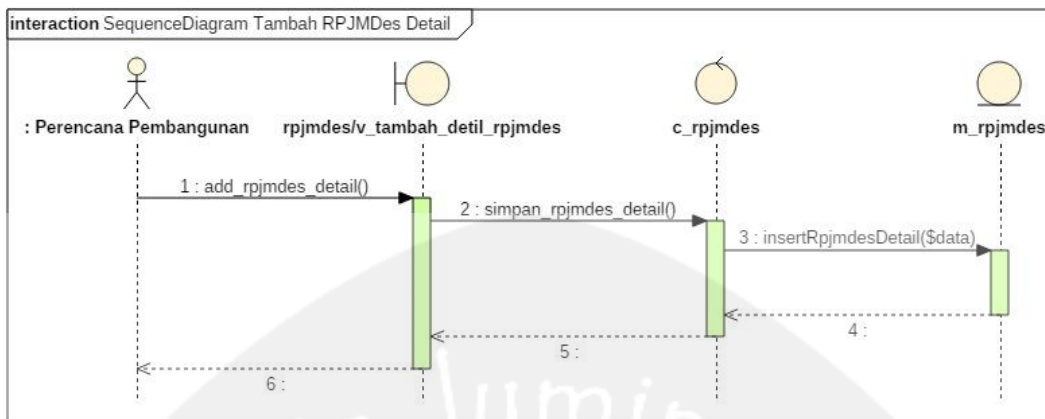
Gambar 2.16 Sequence Diagram: Hapus Sub Program RPJMDes

2.2.1.4.7 Tampil Data Detil RPJMDes



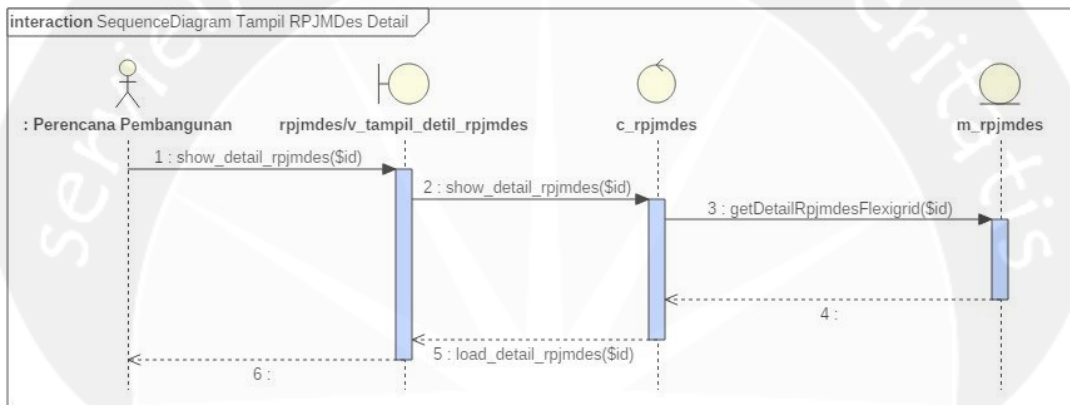
Gambar 2.17 Sequence Diagram: Tampil Data Detil RPJMDes

2.2.1.4.8 Tambah Data Detil RPJMDes



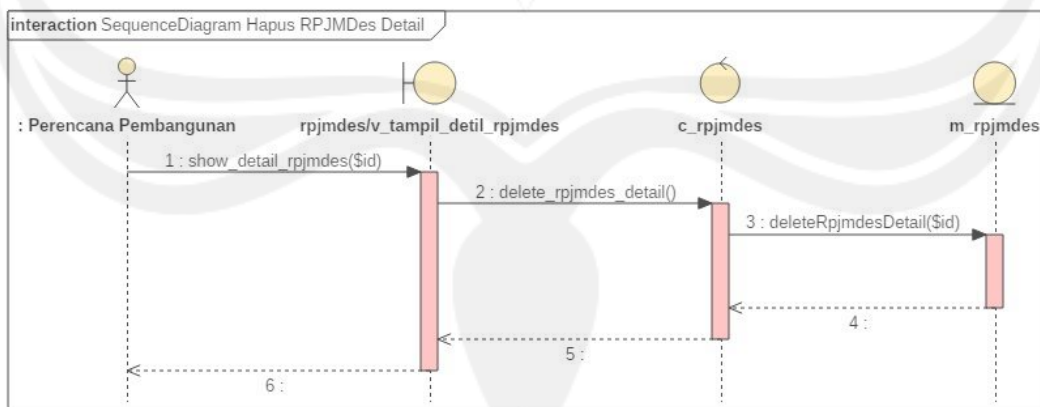
Gambar 2.18 Sequence Diagram: Tambah Data Detil RPJMDes

2.2.1.4.9 Ubah Data Detil RPJMDes



Gambar 2.19 Sequence Diagram: Ubah Data Detil RPJMDes

2.2.1.4.10 Hapus Data Detil RPJMDes

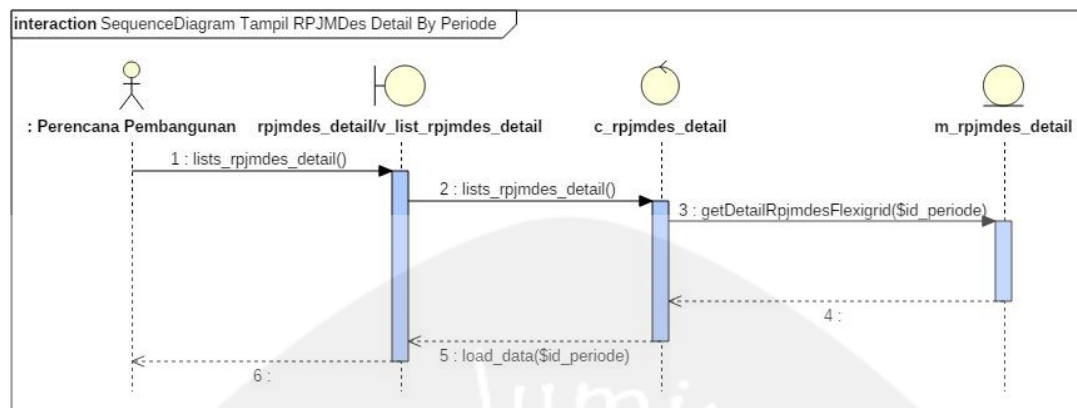


Gambar 2.20 Sequence Diagram: Hapus Data Detil RPJMDes

2.2.1.5 Pengelolaan Detil RPJMDes

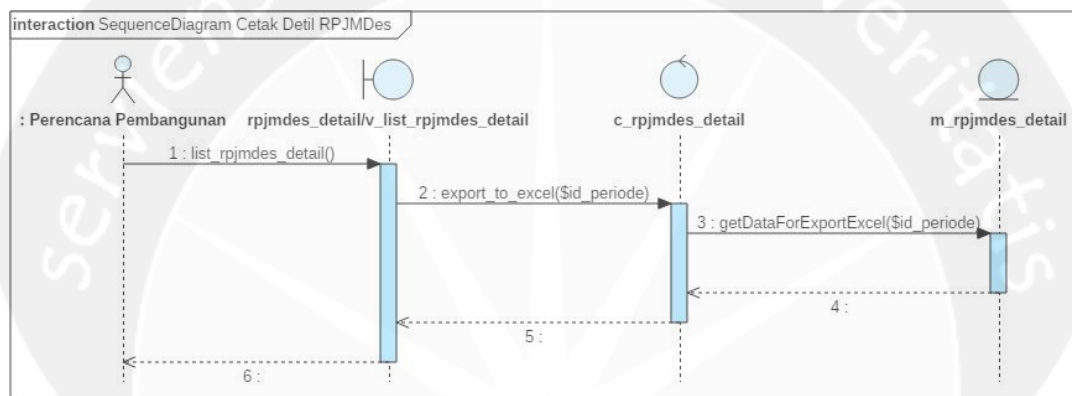
2.2.1.5.1 Tampil Detil RPJMDes By Periode

Program Studi Teknik Informatika	DPPL-SiDeKa-PPD	26/ 111
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		



Gambar 2.21 Sequence Diagram: Tampil Detil RPJMDes By Periode

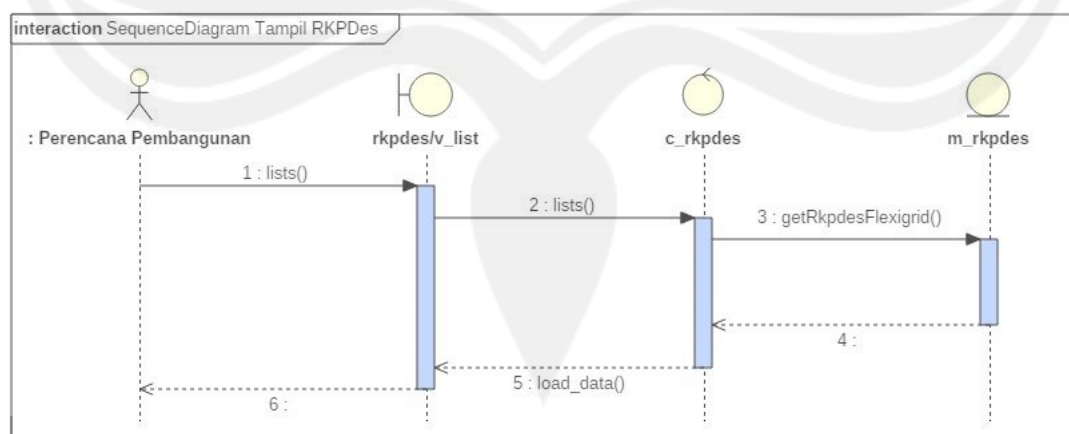
2.2.1.5.2 Cetak Detil RPJMDes By Periode



Gambar 2.22 Sequence Diagram: Cetak Detil RPJMDes By Periode

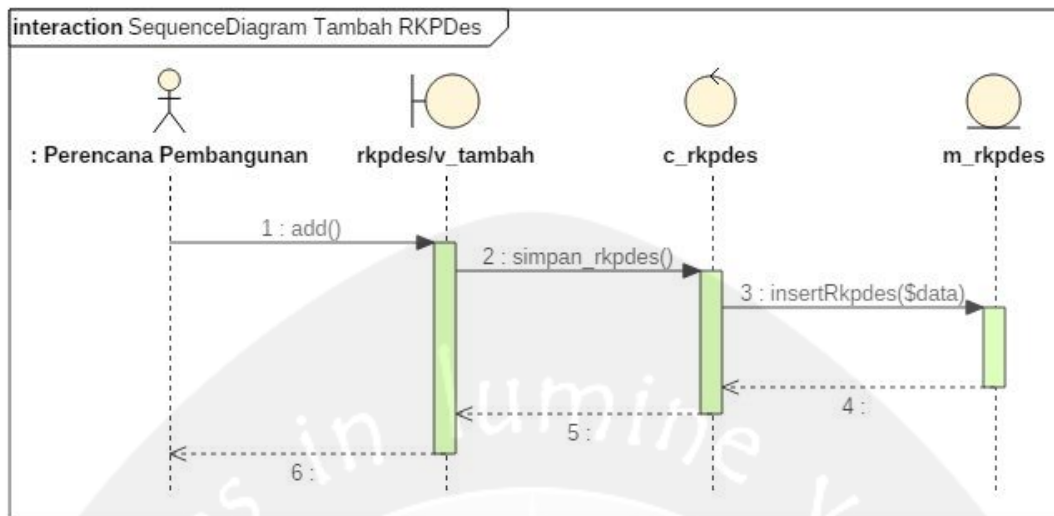
2.2.1.6 Pengelolaan RKPDes

2.2.1.6.1 Tampil Data RKPDes



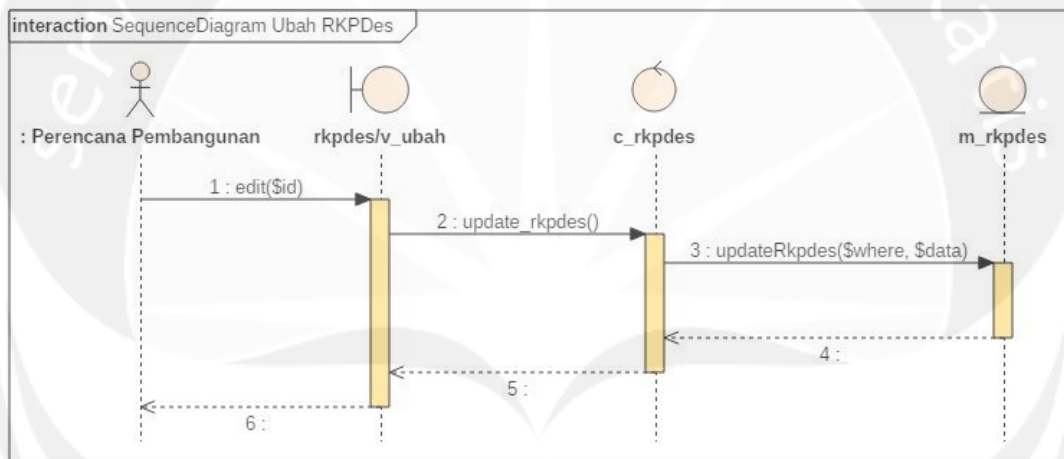
Gambar 2.23 Sequence Diagram: Tampil Data RKPDes

2.2.1.6.2 Tambah Data RKPDes



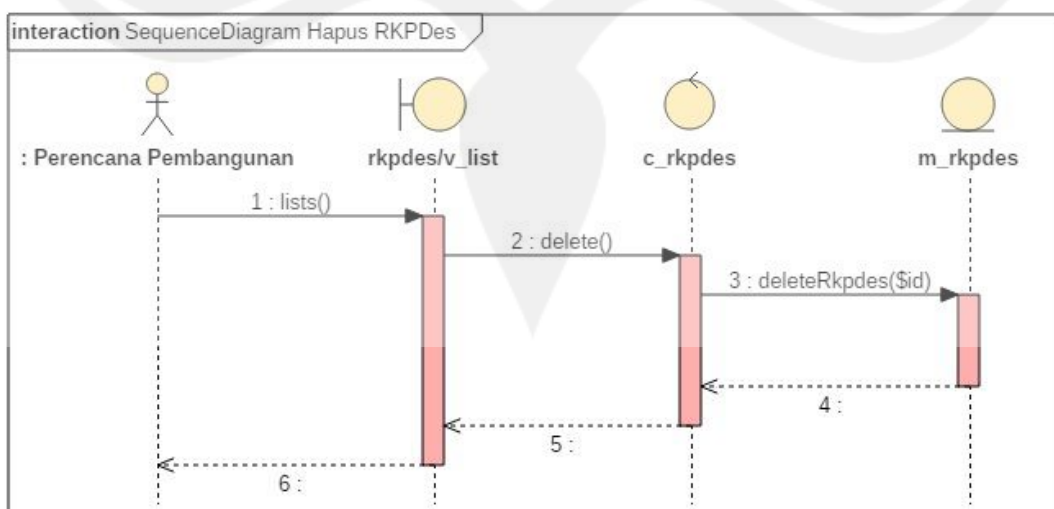
Gambar 2.24 Sequence Diagram: Tambah Data RKPDes

2.2.1.6.3 Ubah Data RKPDes



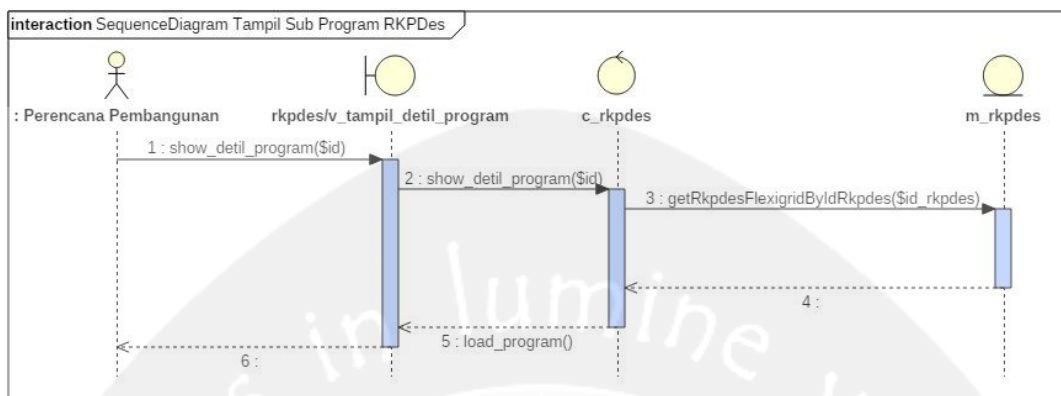
Gambar 2.25 Sequence Diagram: Ubah Data RKPDes

2.2.1.6.4 Hapus Data RKPDes



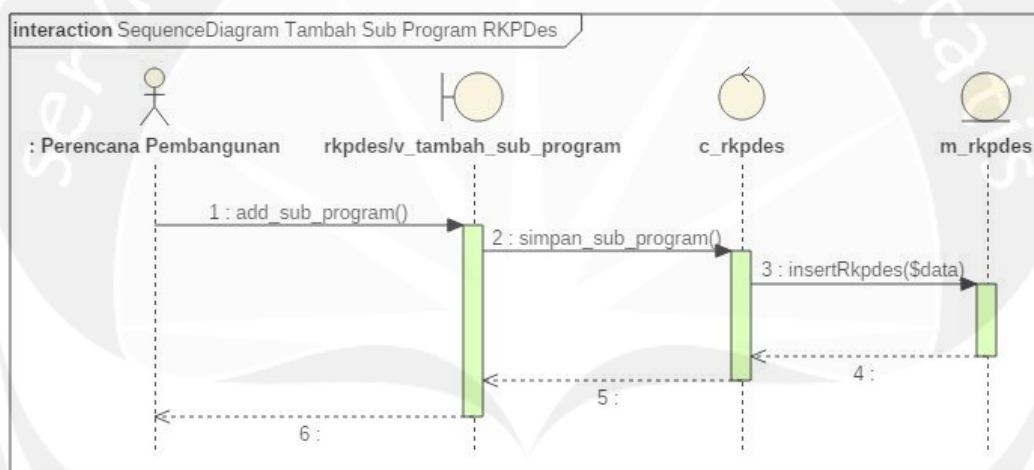
Gambar 2.26 Sequence Diagram: Hapus Data RKPDes

2.2.1.6.5 Tampil Data Sub Program RKPDes



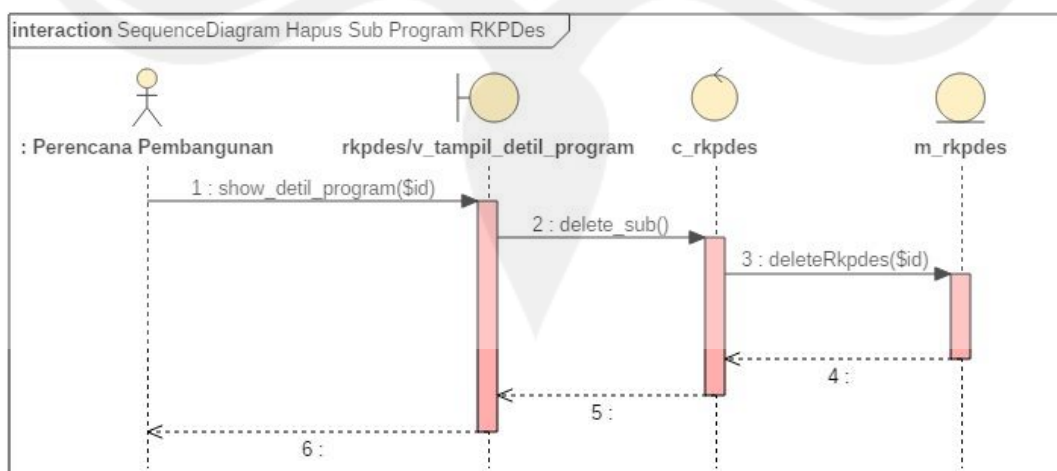
Gambar 2.27 Sequence Diagram: Tampil Data Sub Program RKPDes

2.2.1.6.6 Tambah Data Sub Program RKPDes



Gambar 2.28 Sequence Diagram: Tambah Data Sub Program RKPDes

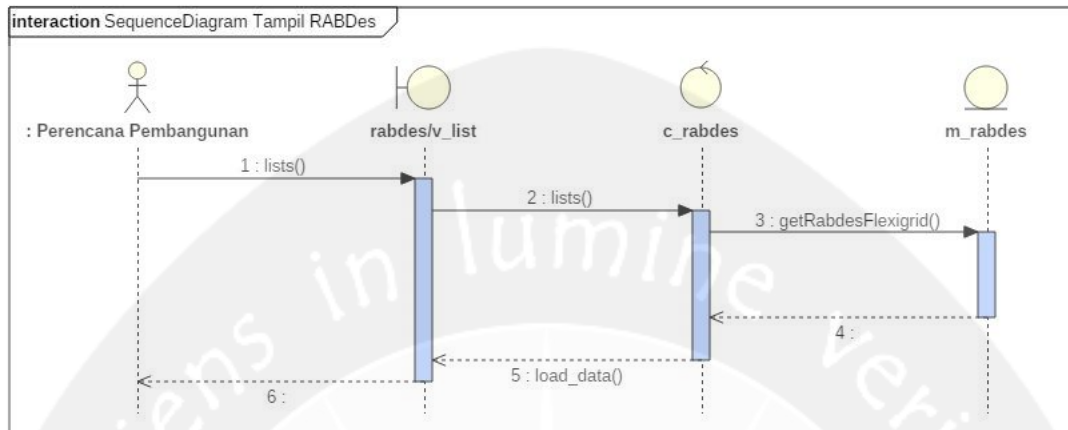
2.2.1.6.7 Hapus Sub Program RKPDes



Gambar 2.29 Sequence Diagram: Hapus Sub Program RKPDes

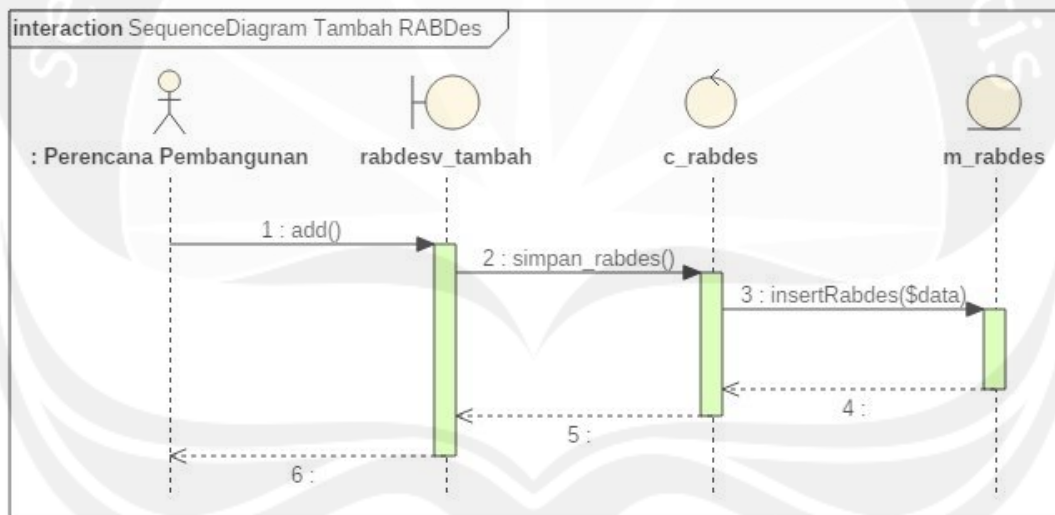
2.2.1.7 Pengelolaan RABDes

2.2.1.7.1 Tampil Data RABDes



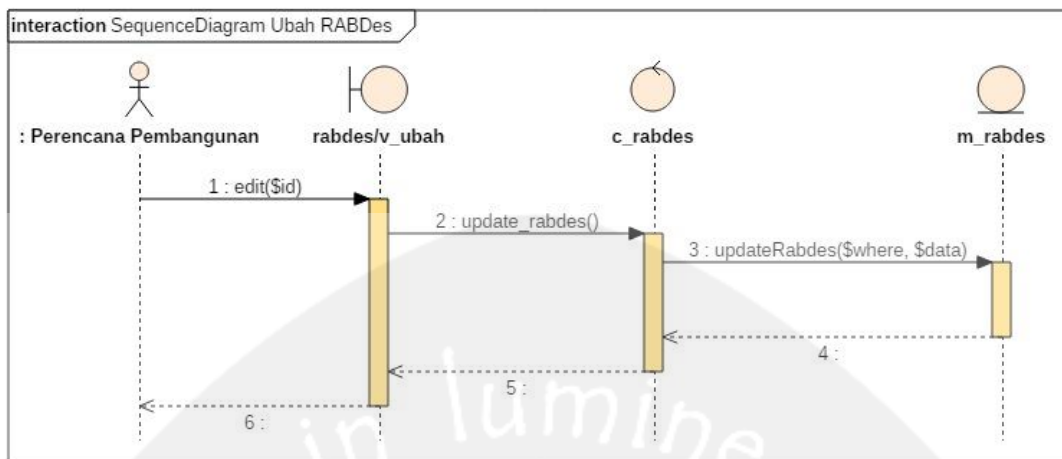
Gambar 2.30 Sequence Diagram: Tampil Data RABDes

2.2.1.7.2 Tambah Data RABDes



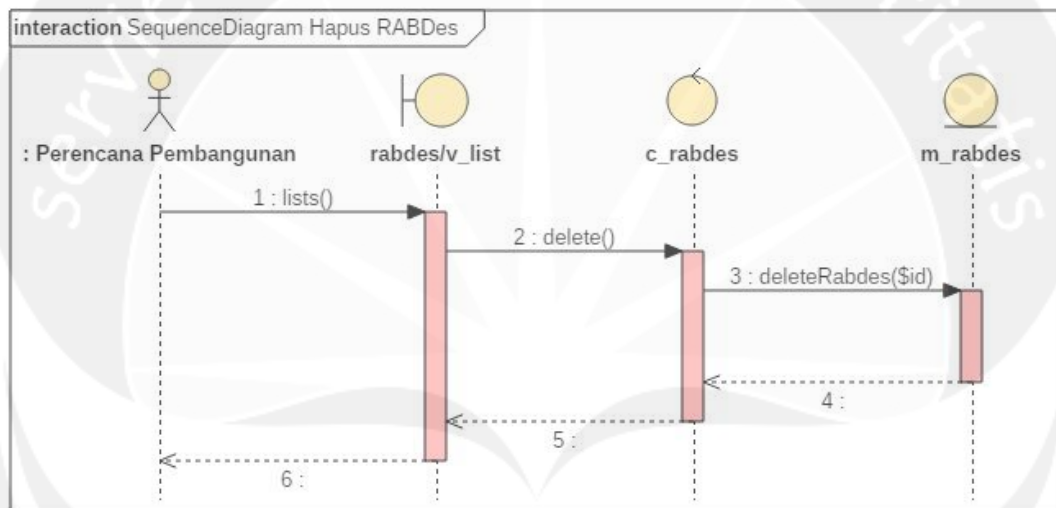
Gambar 2.31 Sequence Diagram: Tambah Data RABDes

2.2.1.7.3 Ubah Data RABDes



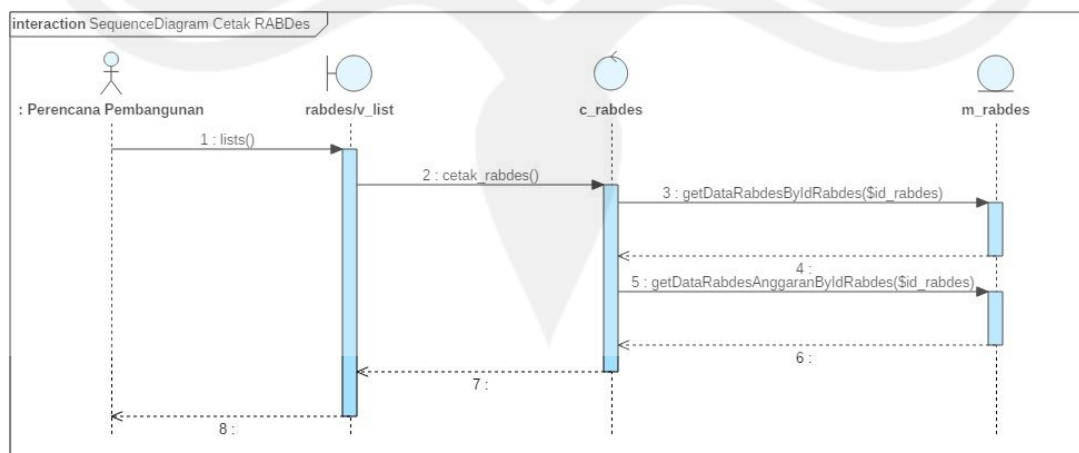
Gambar 2.32 Sequence Diagram: Ubah Data RABDes

2.2.1.7.4 Hapus Data RABDes



Gambar 2.33 Sequence Diagram: Hapus Data RABDes

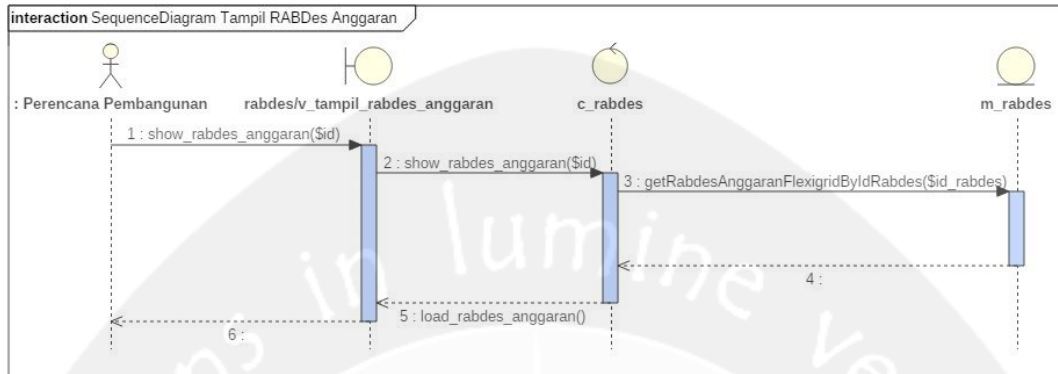
2.2.1.7.5 Cetak Data RABDes



Gambar 2.34 Sequence Diagram: Cetak Data RABDes

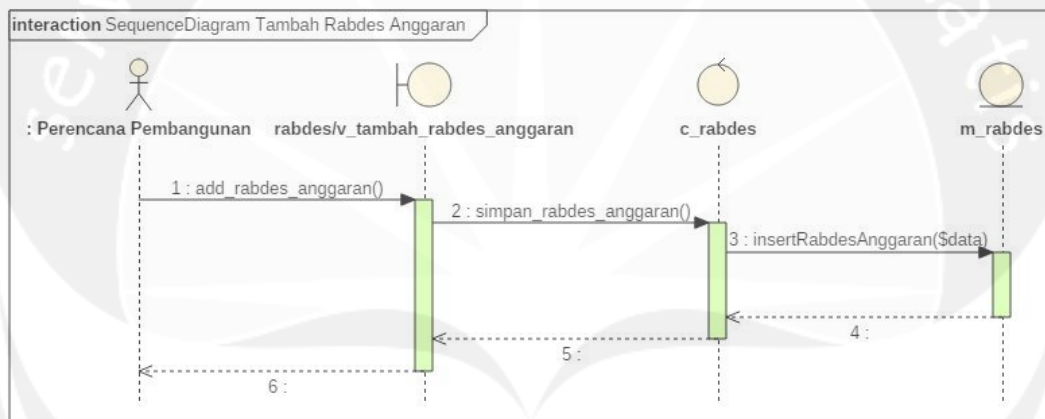
2.2.1.8 Pengelolaan RABDes Anggaran

2.2.1.8.1 Tampil Data RABDes Anggaran



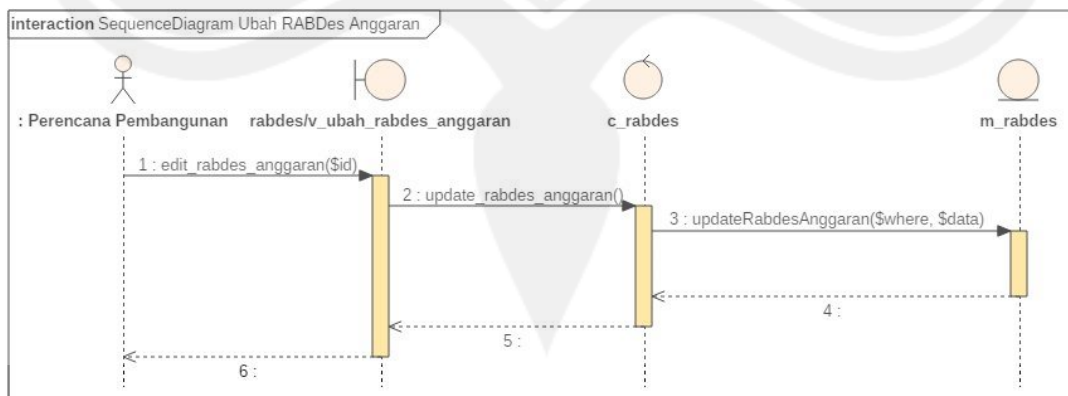
Gambar 2.35 Sequence Diagram: Tampil Data RABDes Anggaran

2.2.1.8.2 Tambah Data RABDes Anggaran



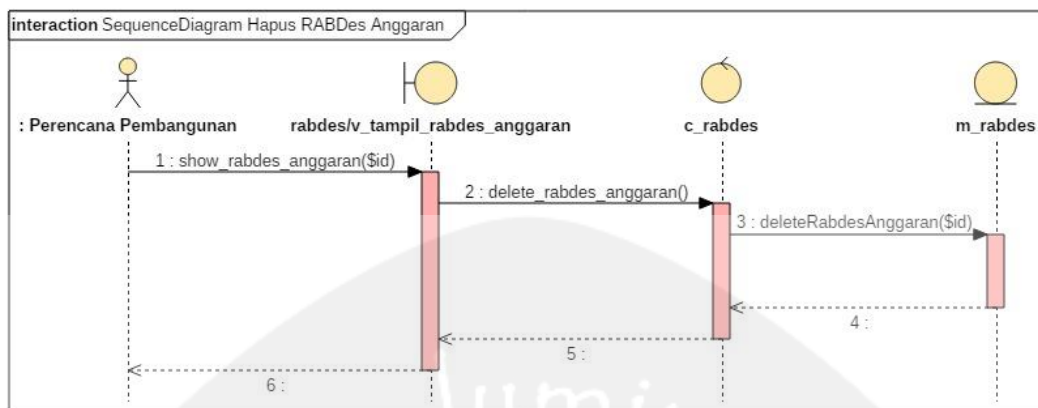
Gambar 2.36 Sequence Diagram: Tambah Data RABDes Anggaran

2.2.1.8.3 Ubah Data RABDes Anggaran



Gambar 2.37 Sequence Diagram: Ubah Data RABDes Anggaran

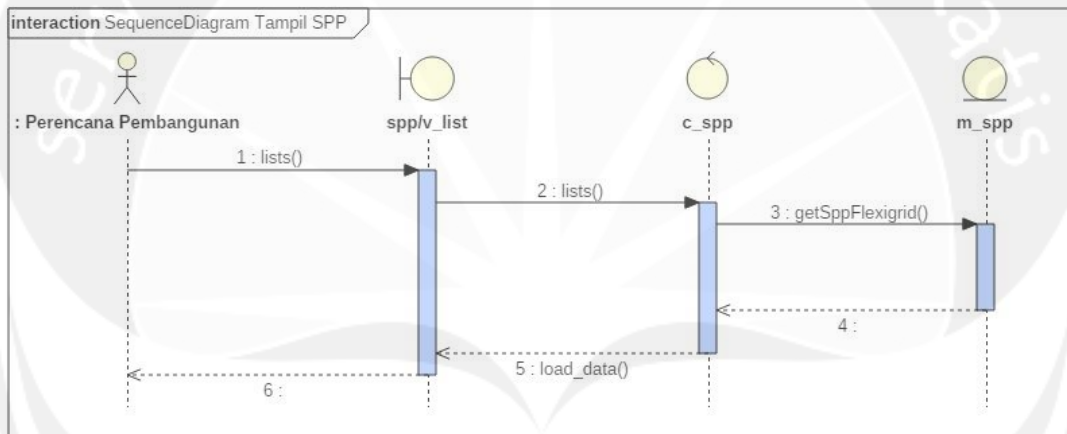
2.2.1.8.4 Hapus Data RABDes Anggaran



Gambar 2.38 Sequence Diagram: Hapus Data RABDes Anggaran

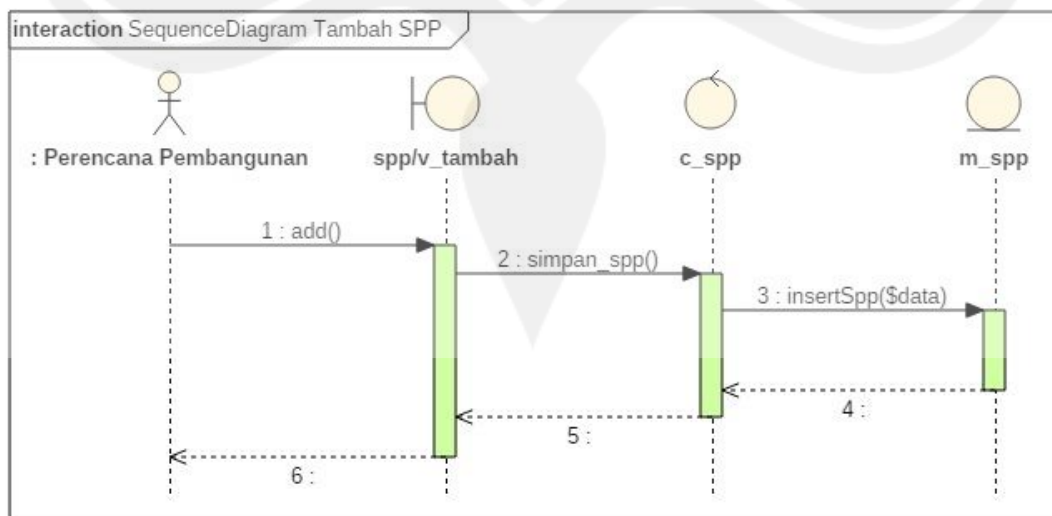
2.2.1.9 Pengelolaan SPP

2.2.1.9.1 Tampil Data SPP



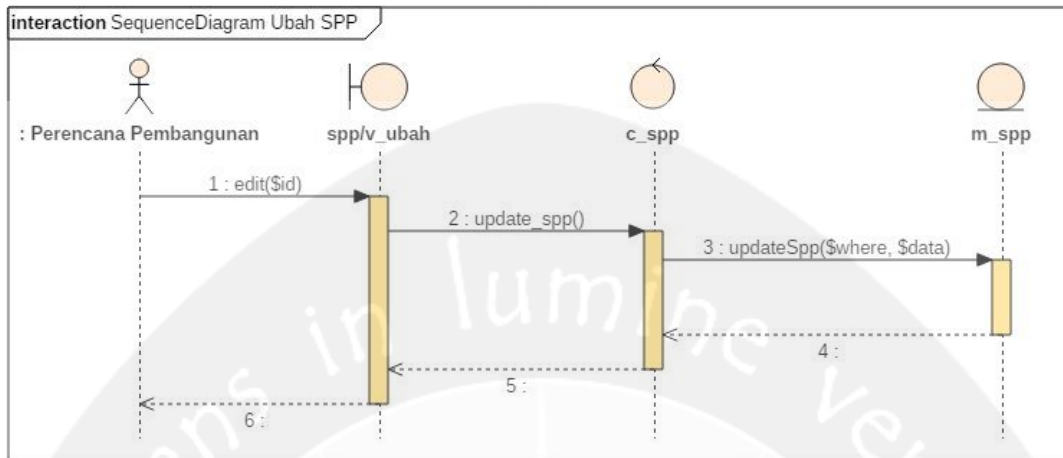
Gambar 2.39 Sequence Diagram: Tampil Data SPP

2.2.1.9.2 Tambah Data SPP



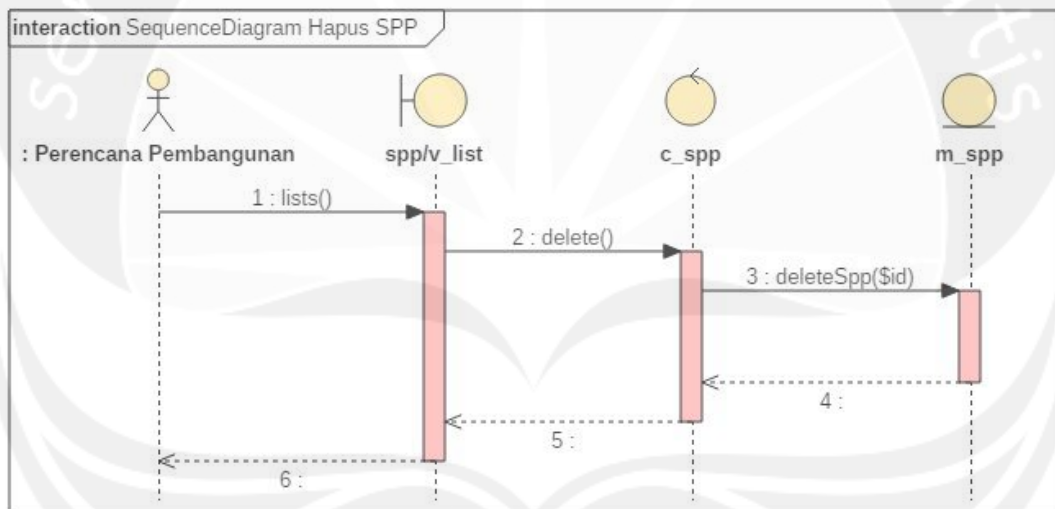
Gambar 2.40 Sequence Diagram: Tambah Data SPP

2.2.1.9.3 Ubah Data SPP



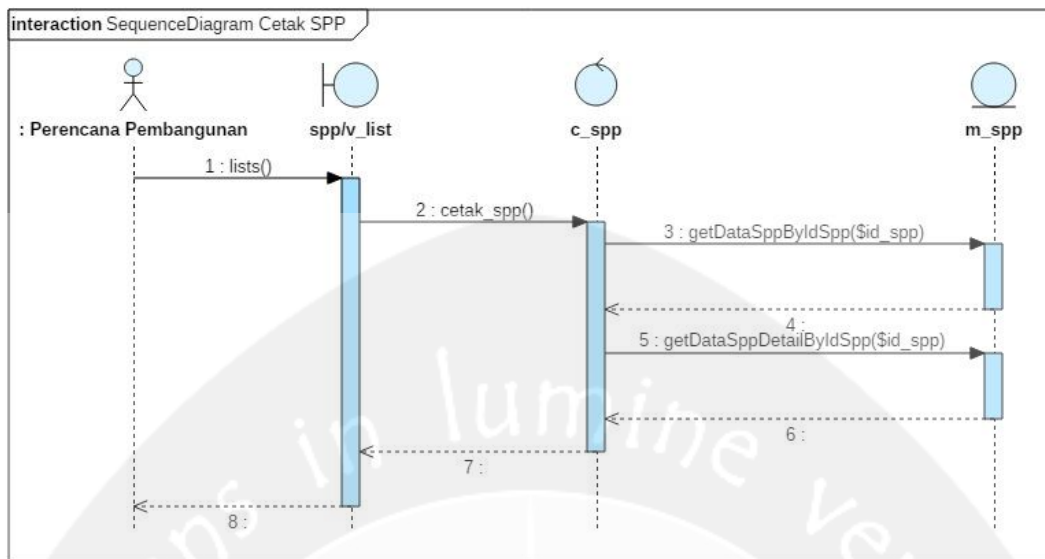
Gambar 2.41 Sequence Diagram: Ubah Data SPP

2.2.1.9.4 Hapus Data SPP



Gambar 2.42 Sequence Diagram: Hapus Data SPP

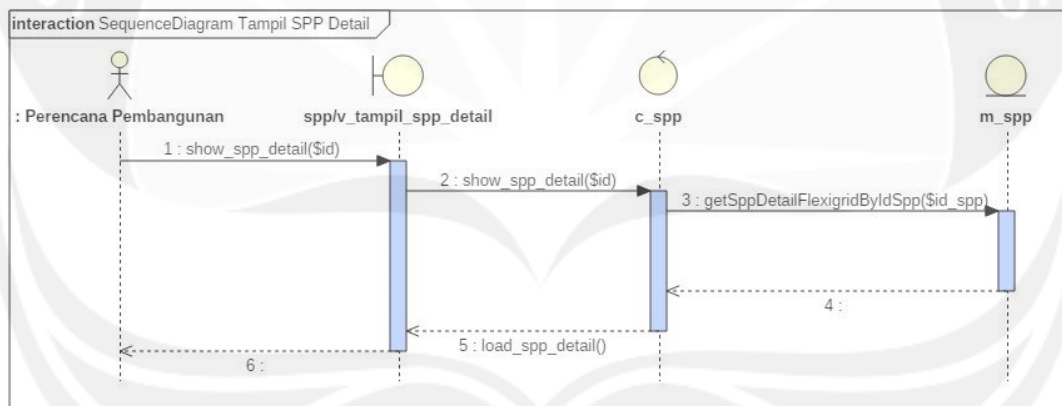
2.2.1.9.5 Cetak Data SPP



Gambar 2.43 Sequence Diagram: Cetak Data SPP

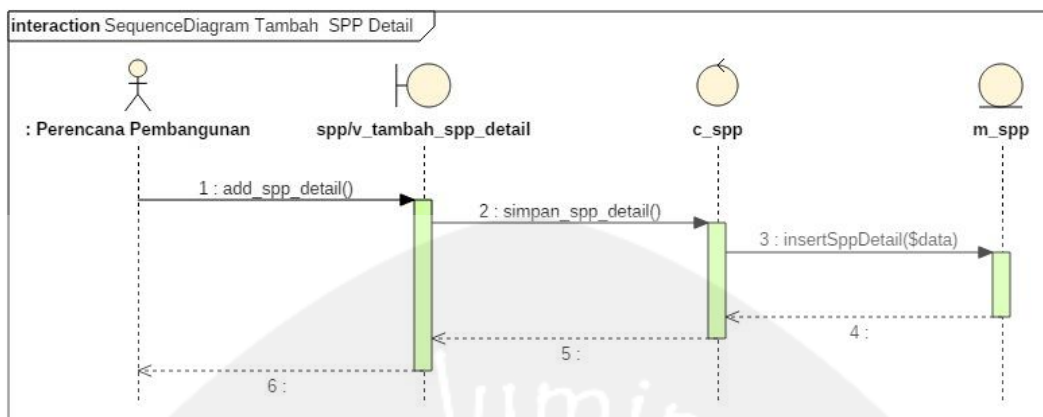
2.2.1.10 Pengelolaan Detil SPP

2.2.1.10.1 Tampil Data Detil SPP



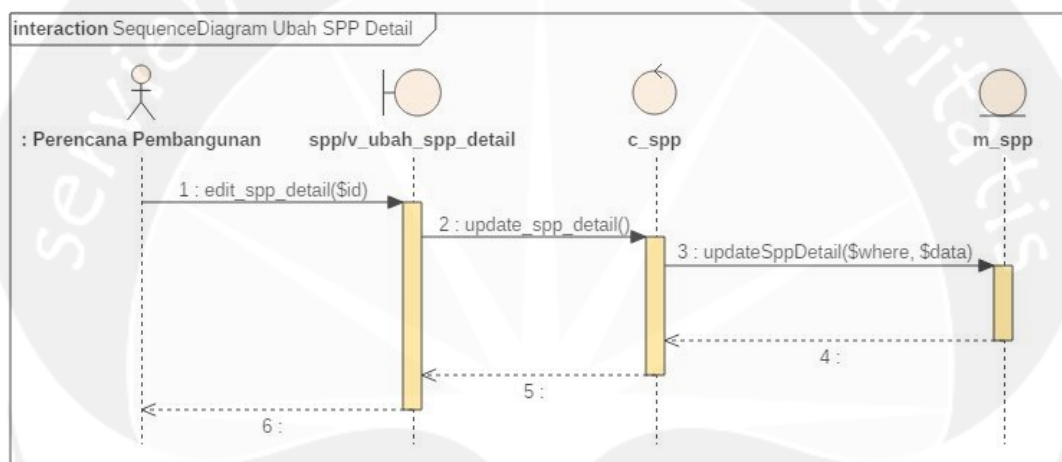
Gambar 2.44 Sequence Diagram: Tampil Data Detil SPP

2.2.1.10.2 Tambah Data Detil SPP



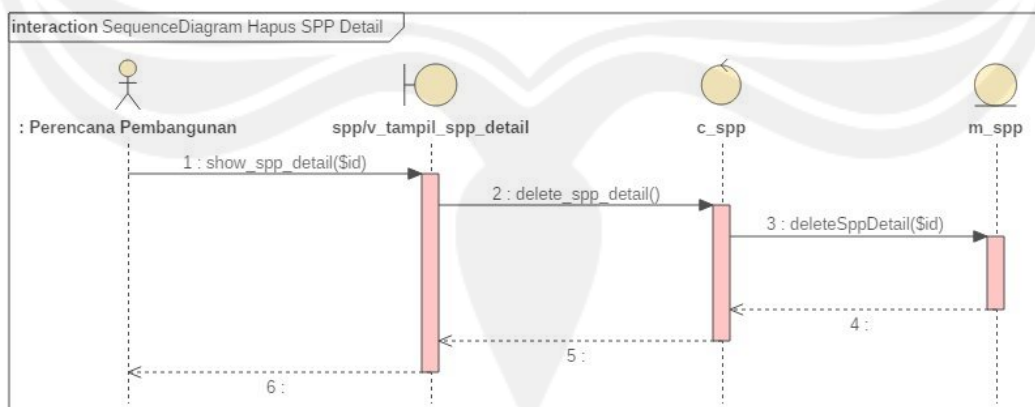
Gambar 2.45 Sequence Diagram: Tambah Data Detil SPP

2.2.1.10.3 Ubah Data Detil SPP



Gambar 2.46 Sequence Diagram: Ubah Data Detil SPP

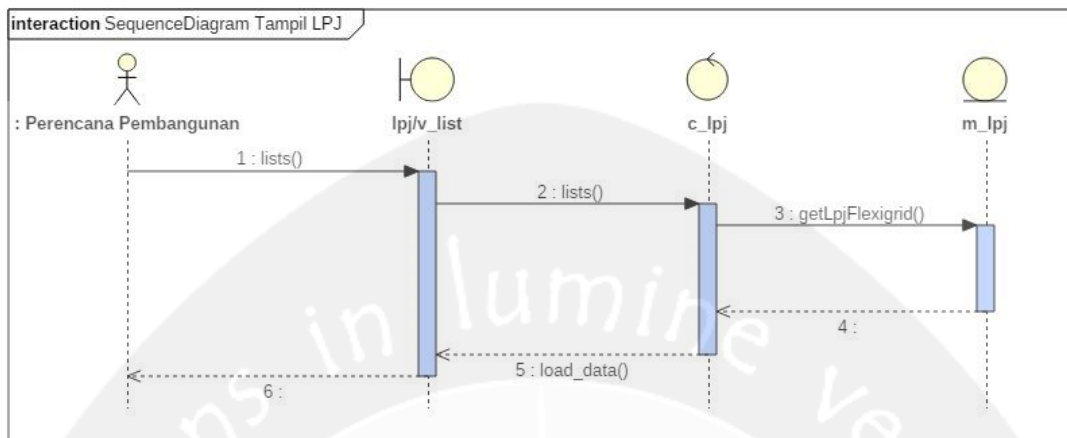
2.2.1.10.4 Hapus Data Detil SPP



Gambar 2.47 Sequence Diagram: Hapus Data Detil SPP

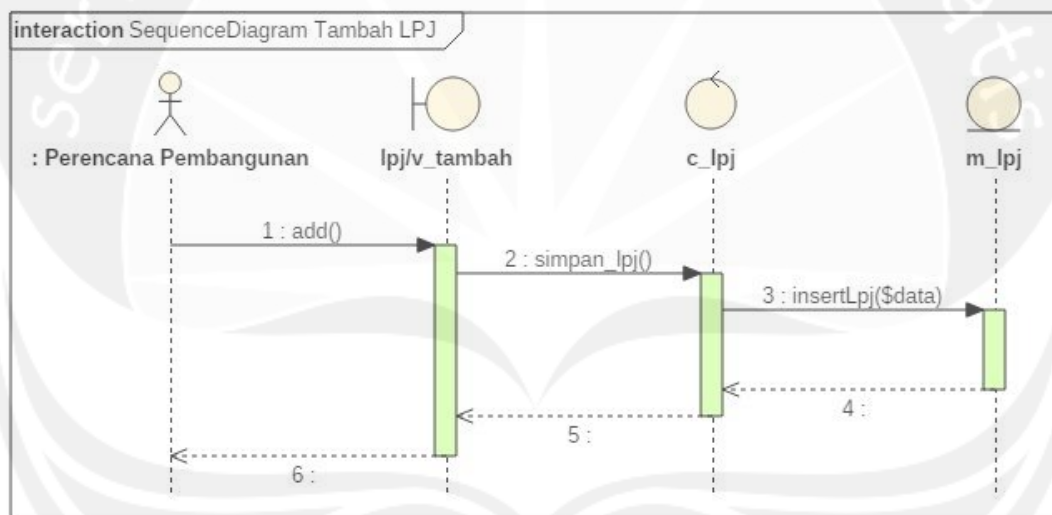
2.2.1.11 Pengelolaan LPJ

2.2.1.11.1 Tampil Data LPJ



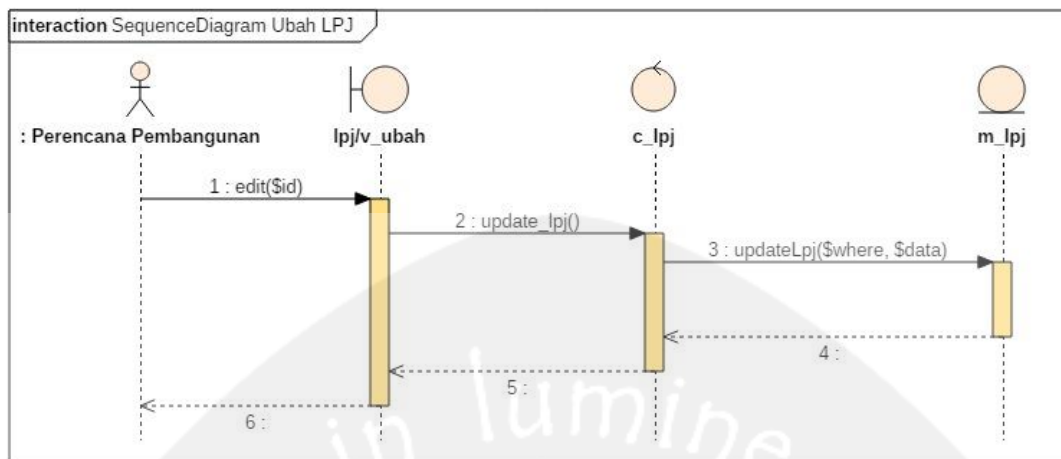
Gambar 2.48 Sequence Diagram: Tampil Data LPJ

2.2.1.11.2 Tambah Data LPJ



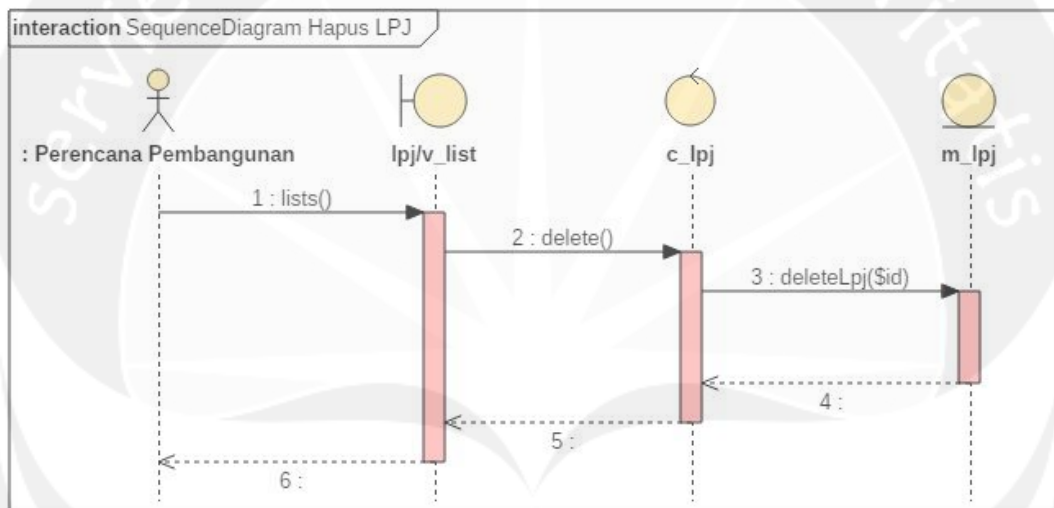
Gambar 2.49 Sequence Diagram: Tambah Data LPJ

2.2.1.11.3 Ubah Data LPJ



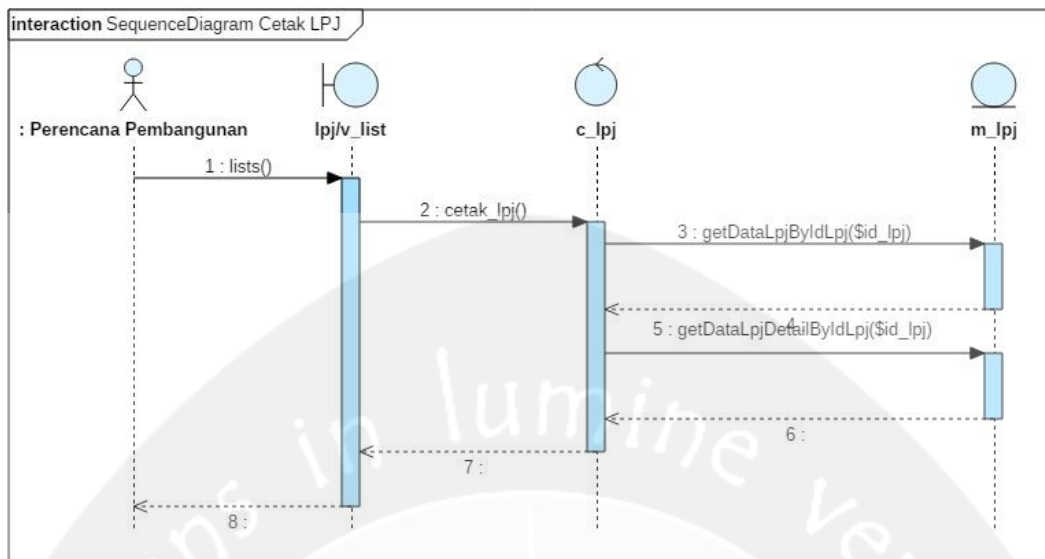
Gambar 2.50 Sequence Diagram: Ubah Data LPJ

2.2.1.11.4 Hapus Data LPJ



Gambar 2.51 Sequence Diagram: Hapus Data LPJ

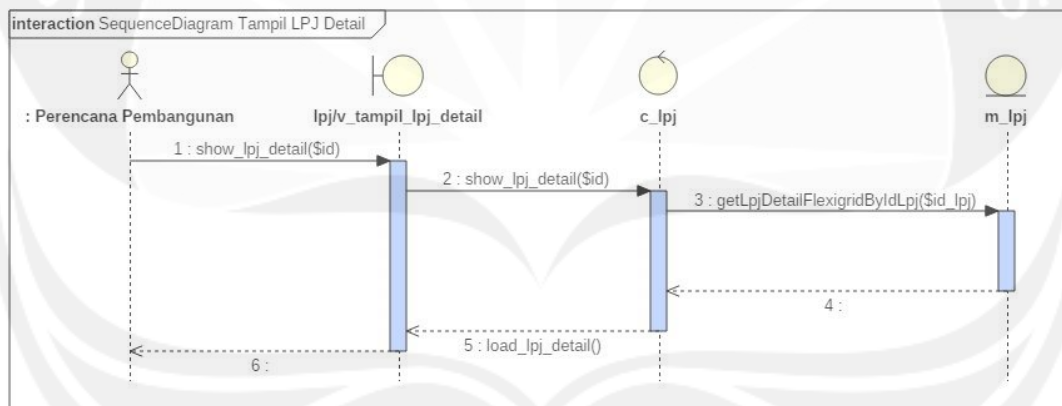
2.2.1.11.5 Cetak Data LPJ



Gambar 2.52 Sequence Diagram: Cetak Data LPJ

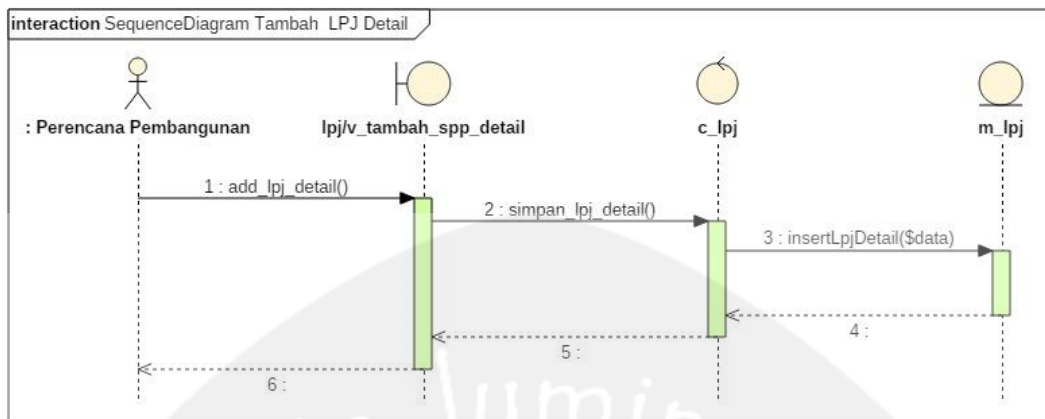
2.2.1.12 Pengelolaan Detil LPJ

2.2.1.12.1 Tampil Data Detil LPJ



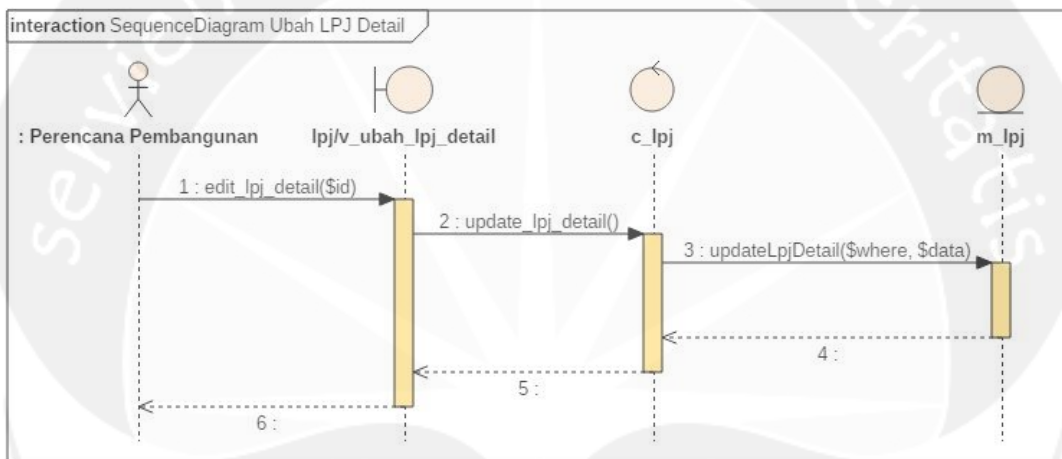
Gambar 2.53 Sequence Diagram: Tampil Data Detil LPJ

2.2.1.12.2 Tambah Data Detil LPJ



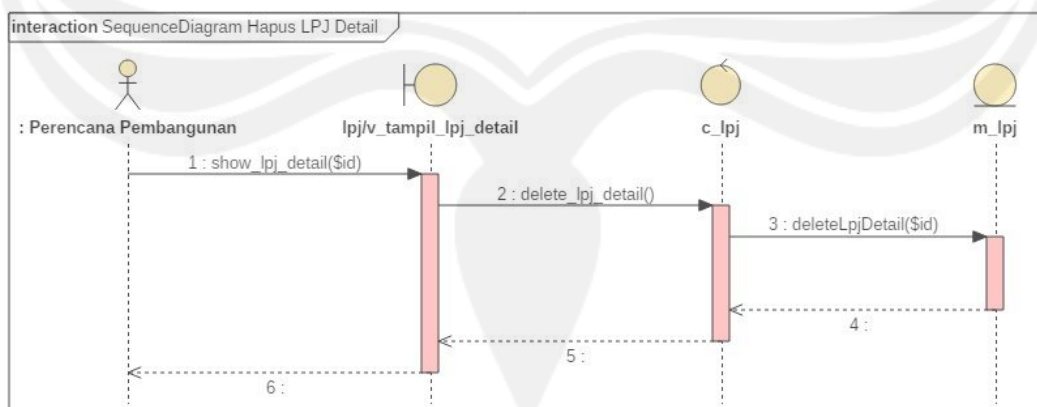
Gambar 2.54 Sequence Diagram: Tambah Data Detil LPJ

2.2.1.12.3 Ubah Data Detil LPJ



Gambar 2.55 Sequence Diagram: Ubah Data Detil LPJ

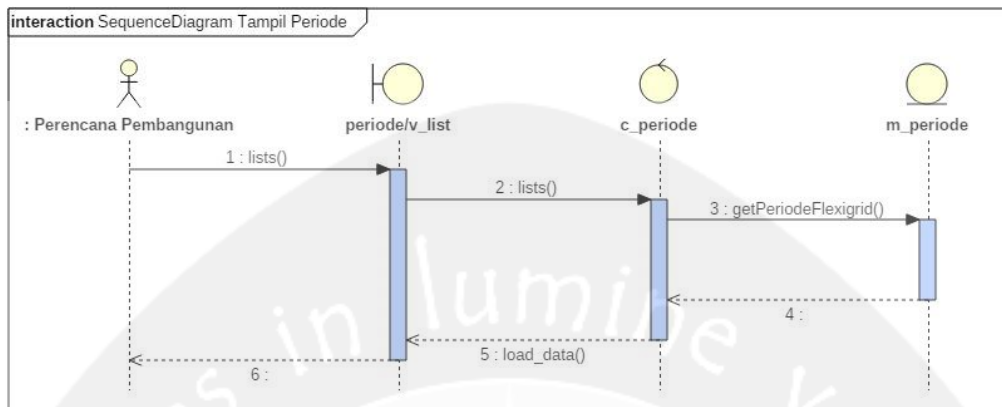
2.2.1.12.4 Hapus Data Detil LPJ



Gambar 2.56 Sequence Diagram: Hapus Data Detil LPJ

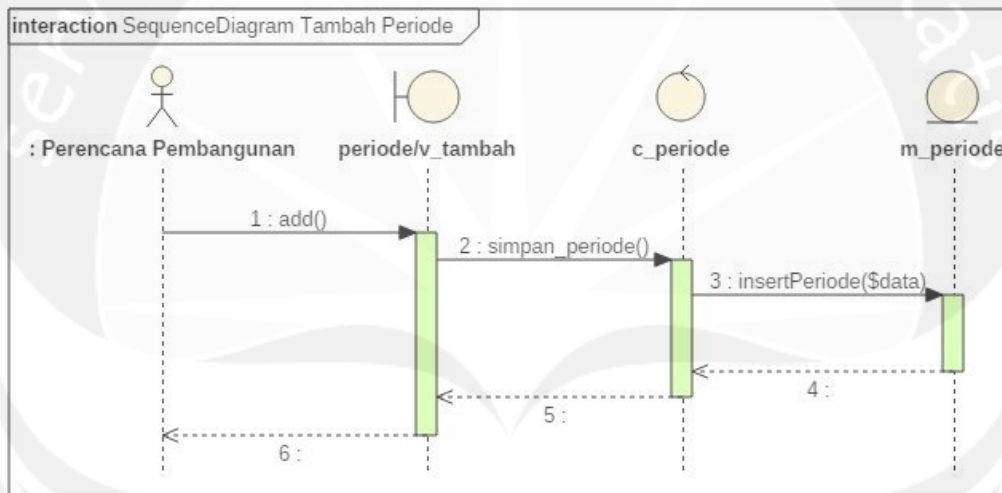
2.2.1.13 Pengelolaan Periode

2.2.1.13.1 Tampil Data Periode



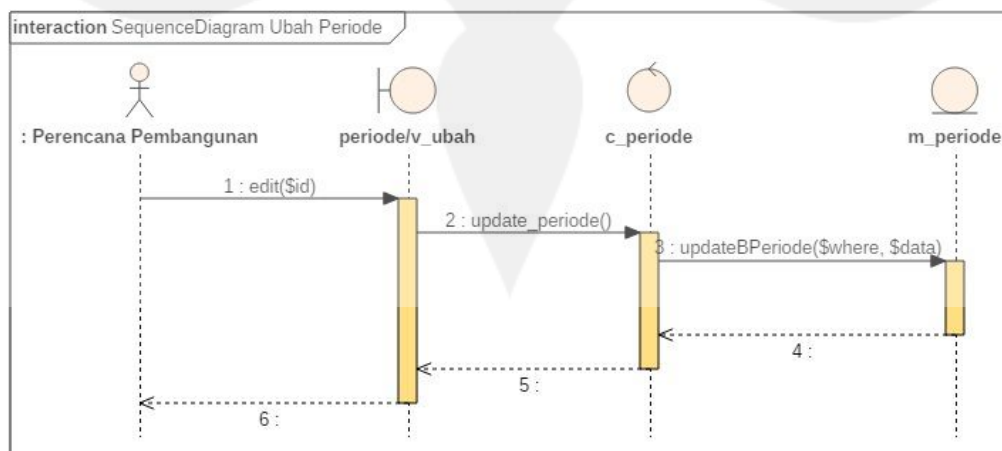
Gambar 2.57 Sequence Diagram: Tampil Data Periode

2.2.1.13.2 Tambah Data Periode



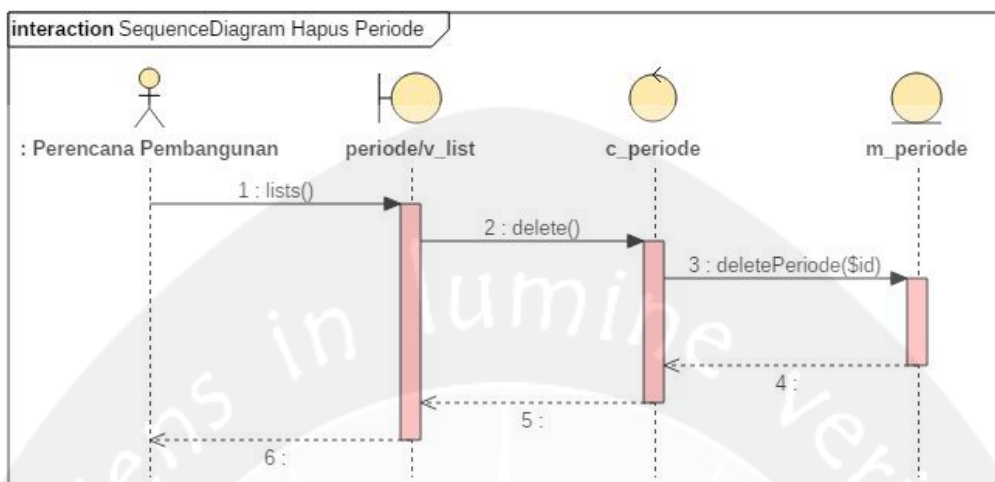
Gambar 2.58 Sequence Diagram: Tambah Data Periode

2.2.1.13.3 Ubah Data Periode



Gambar 2.59 Sequence Diagram: Ubah Data Periode

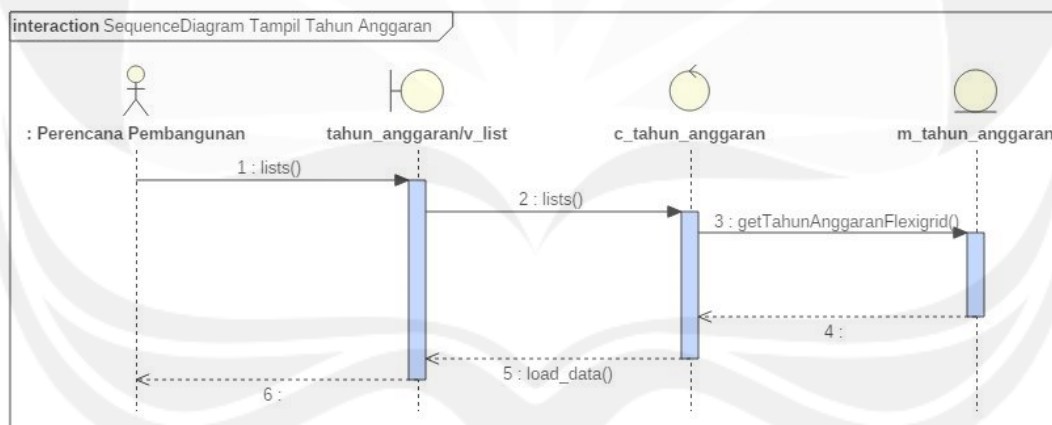
2.2.1.13.4 Hapus Data Periode



Gambar 2.60 Sequence Diagram: Hapus Data Periode

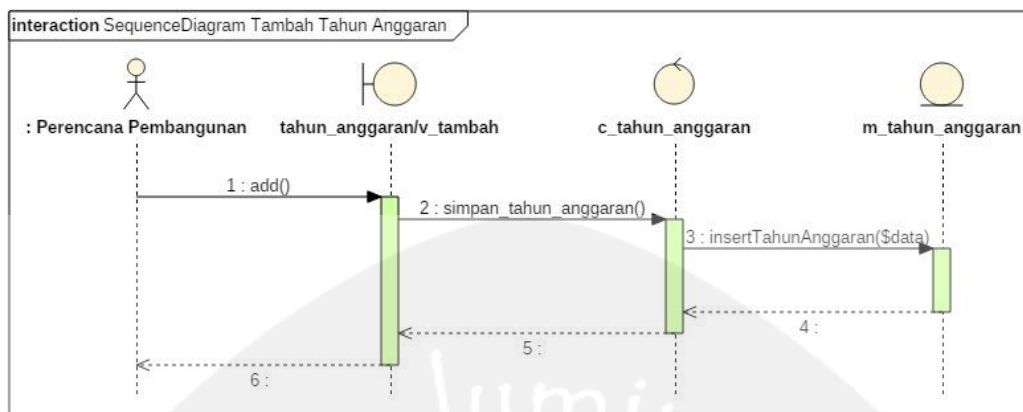
2.2.1.14 Pengelolaan Tahun Anggaran

2.2.1.14.1 Tampil Data Tahun Anggaran



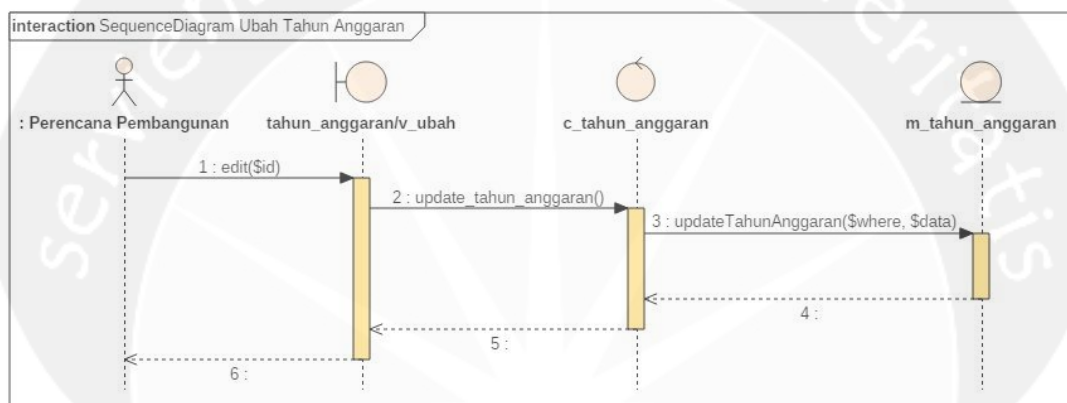
Gambar 2.61 Sequence Diagram: Tampil Data Tahun Anggaran

2.2.1.14.2 Tambah Data Tahun Anggaran



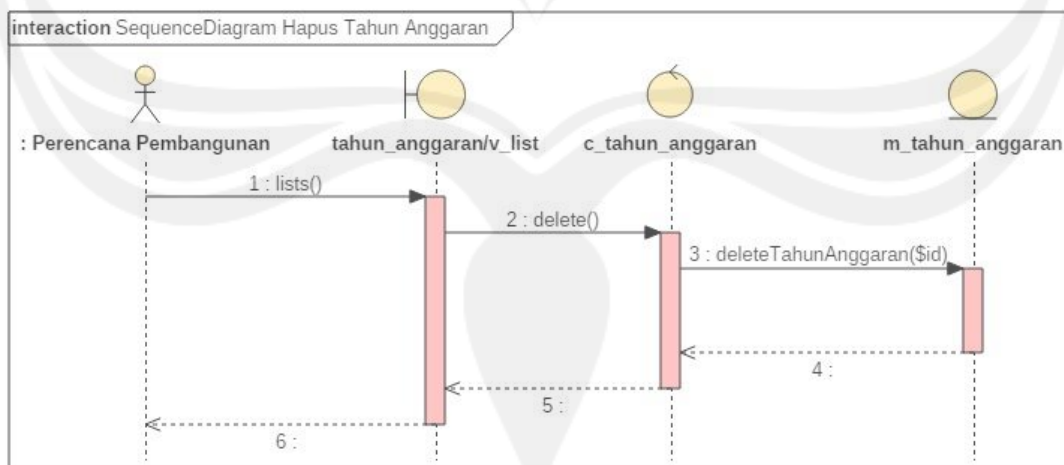
Gambar 2.62 Sequence Diagram: Tambah Data Tahun Anggaran

2.2.1.14.3 Ubah Data Tahun Anggaran



Gambar 2.63 Sequence Diagram: Ubah Data Tahun Anggaran

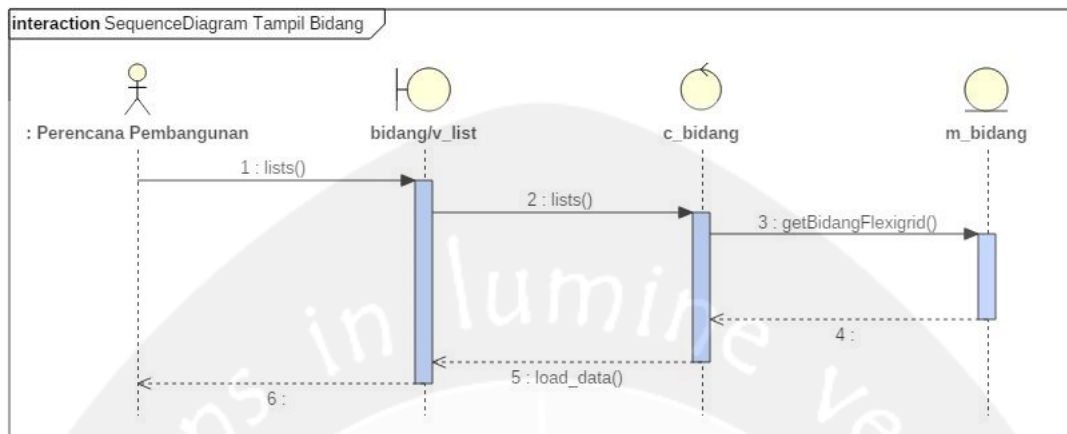
2.2.1.14.4 Hapus Data Tahun Anggaran



Gambar 2.64 Sequence Diagram: Hapus Data Tahun Anggaran

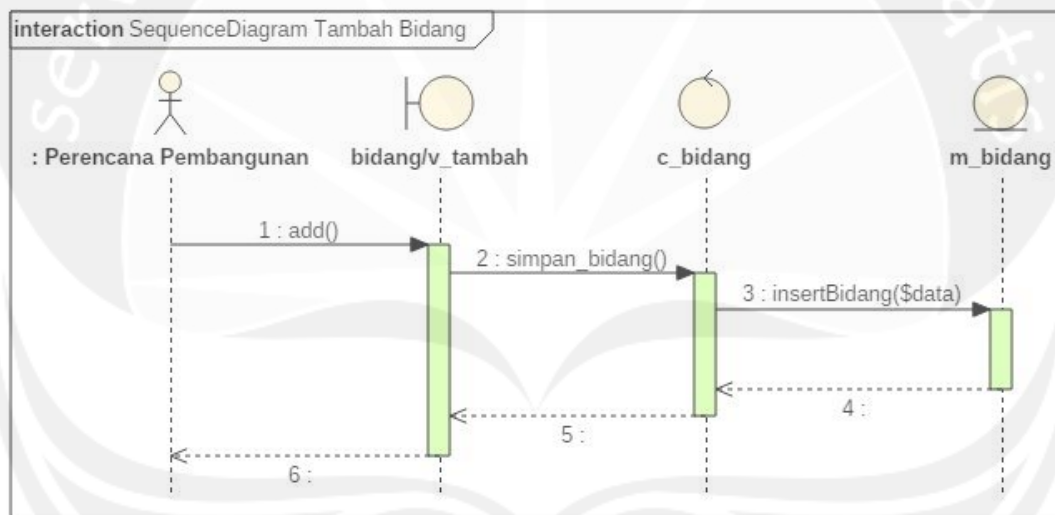
2.2.1.15 Pengelolaan Bidang

2.2.1.15.1 Tampil Data Bidang



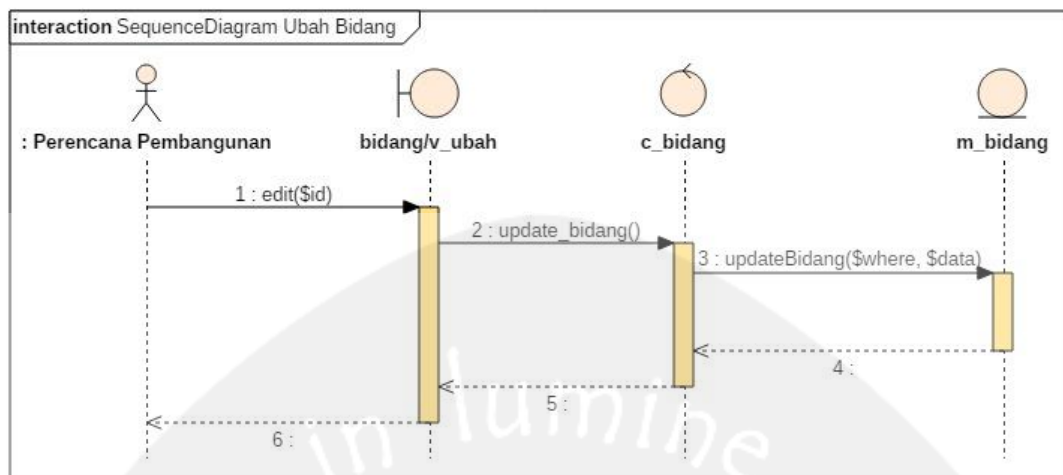
Gambar 2.65 Sequence Diagram: Tampil Data Bidang

2.2.1.15.2 Tambah Data Bidang



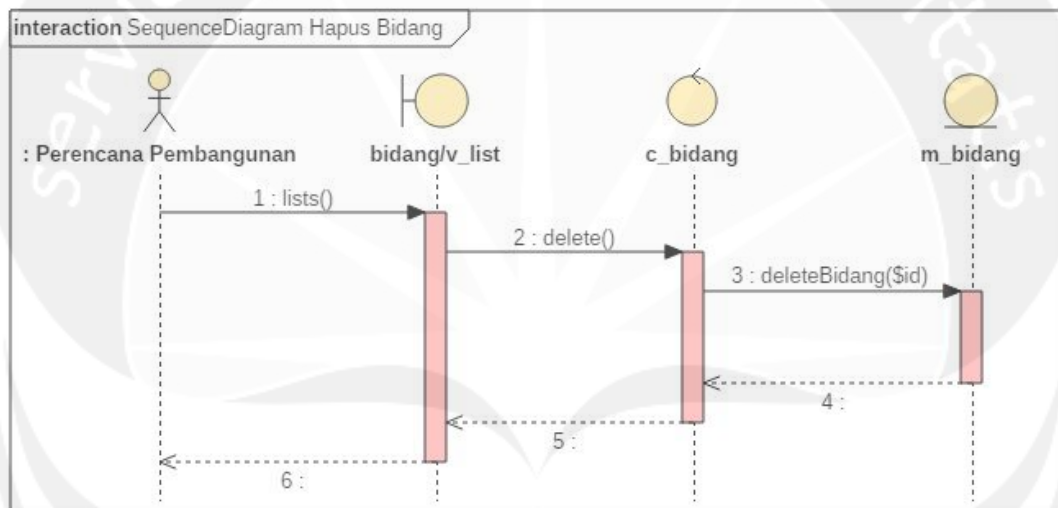
Gambar 2.66 Sequence Diagram: Tambah Data Bidang

2.2.1.15.3 Ubah Data Bidang



Gambar 2.67 Sequence Diagram: Ubah Data Bidang

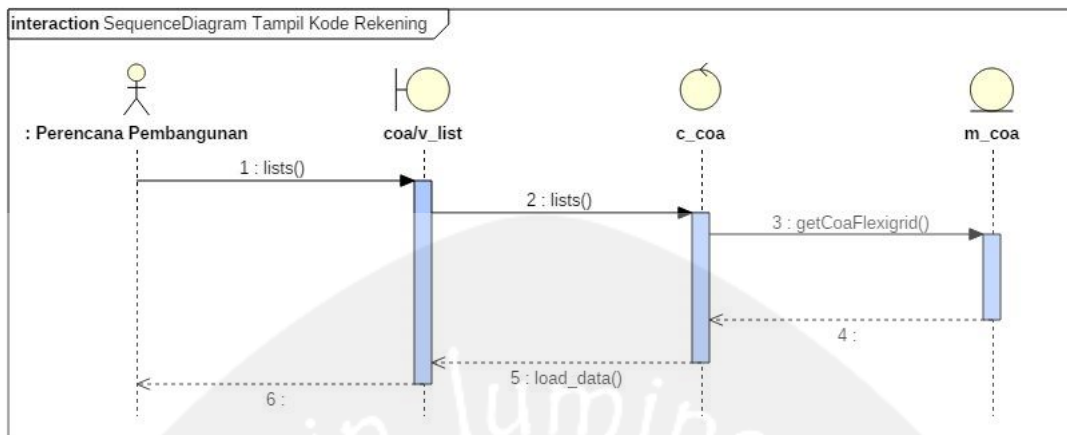
2.2.1.15.4 Hapus Data Bidang



Gambar 2.68 Sequence Diagram: Hapus Data Bidang

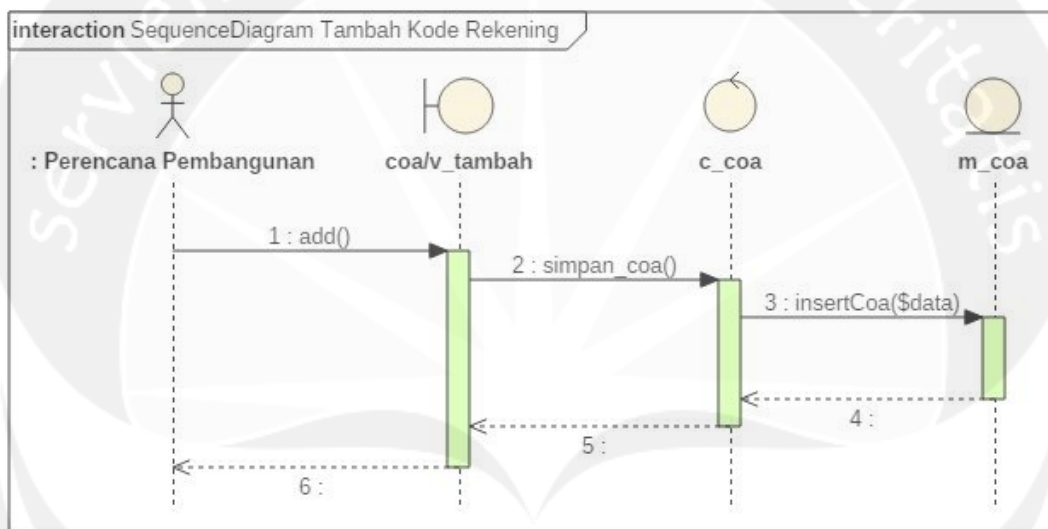
2.2.1.16 Pengelolaan Kode Rekening

2.2.1.16.1 Tampil Data Kode Rekening



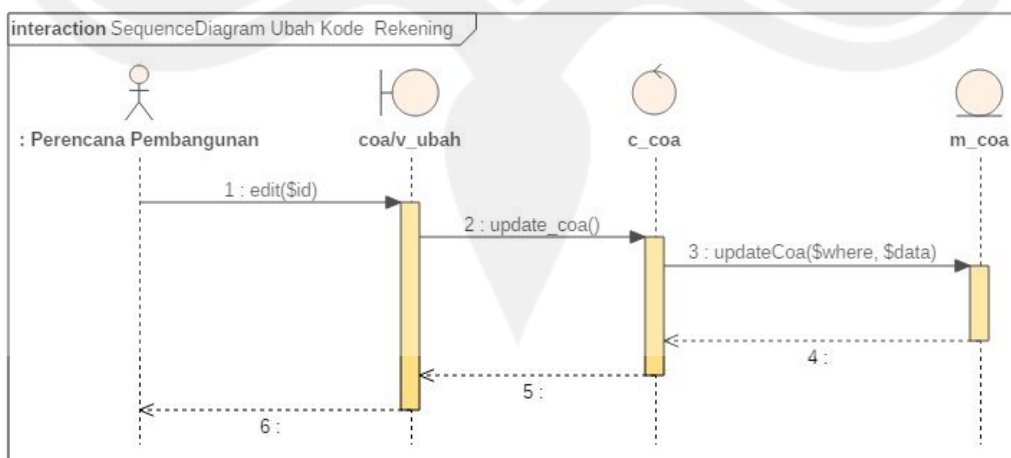
Gambar 2.69 Sequence Diagram: Tampil Data Kode Rekening

2.2.1.16.2 Tambah Data Kode Rekening



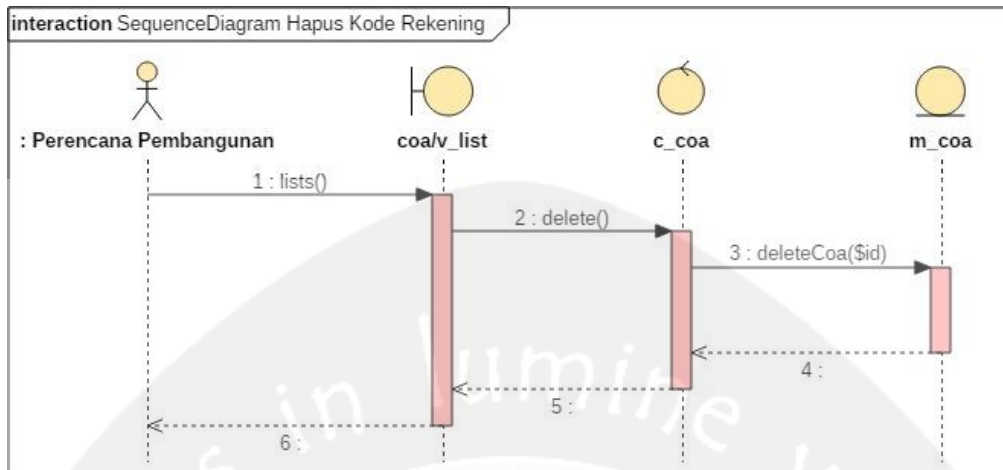
Gambar 2.70 Sequence Diagram: Tambah Data Kode Rekening

2.2.1.16.3 Ubah Data Kode Rekening



Gambar 2.71 Sequence Diagram: Ubah Data Kode Rekening

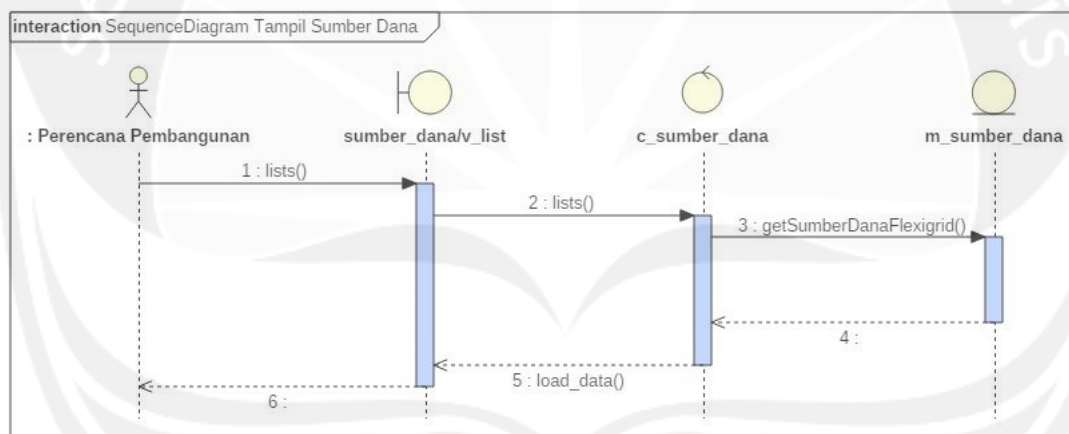
2.2.1.16.4 Hapus Data Kode Rekening



Gambar 2.72 Sequence Diagram: Hapus Data Bidang

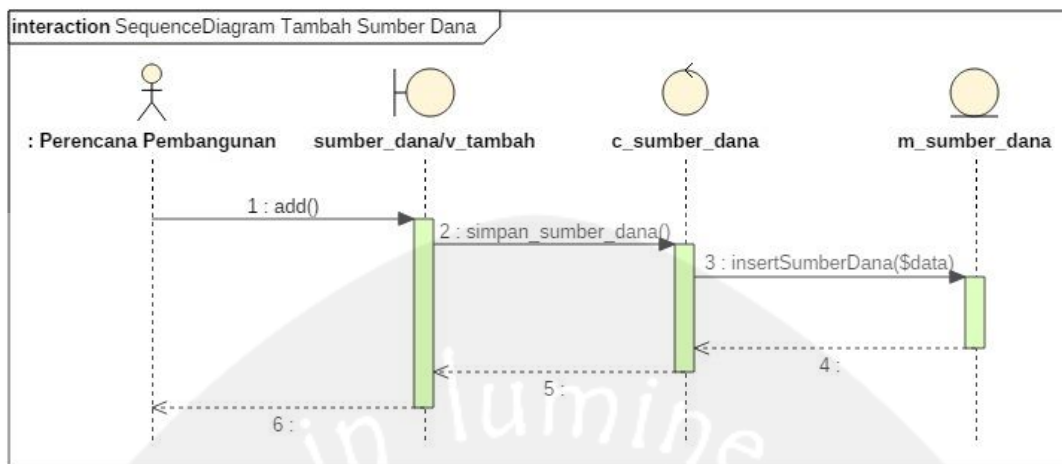
2.2.1.17 Pengelolaan Sumber Dana

2.2.1.17.1 Tampil Data Sumber Dana



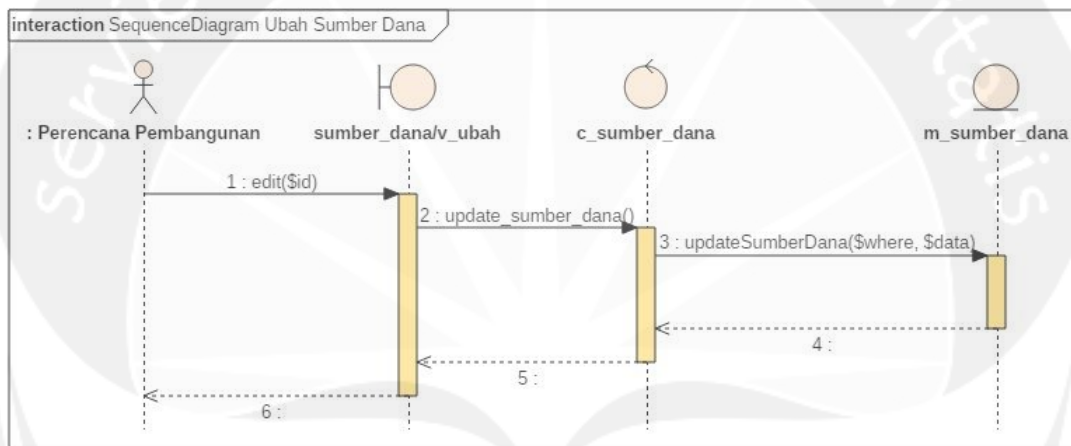
Gambar 2.73 Sequence Diagram: Tampil Data Sumber Dana

2.2.1.17.2 Tambah Data Sumber Dana



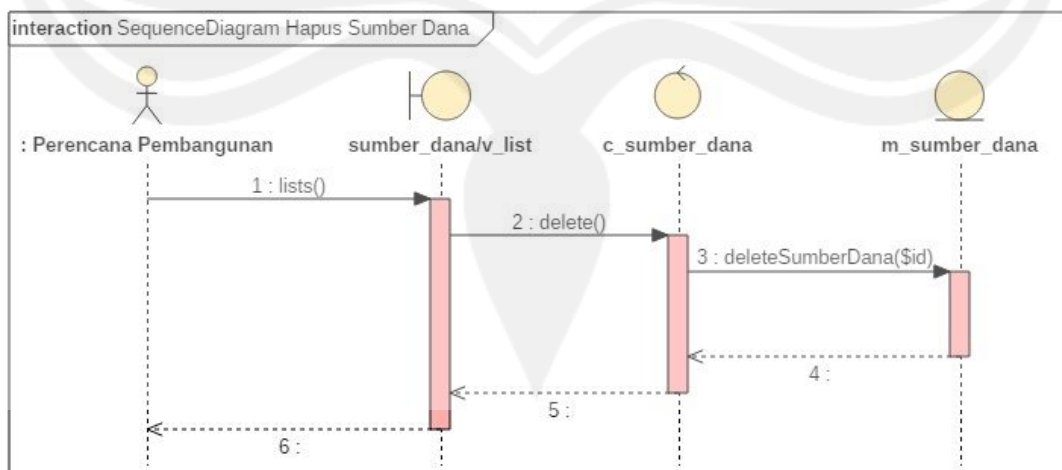
Gambar 2.74 Sequence Diagram: Tambah Data Sumber Dana

2.2.1.17.3 Ubah Data Sumber Dana



Gambar 2.75 Sequence Diagram: Ubah Data Sumber Dana

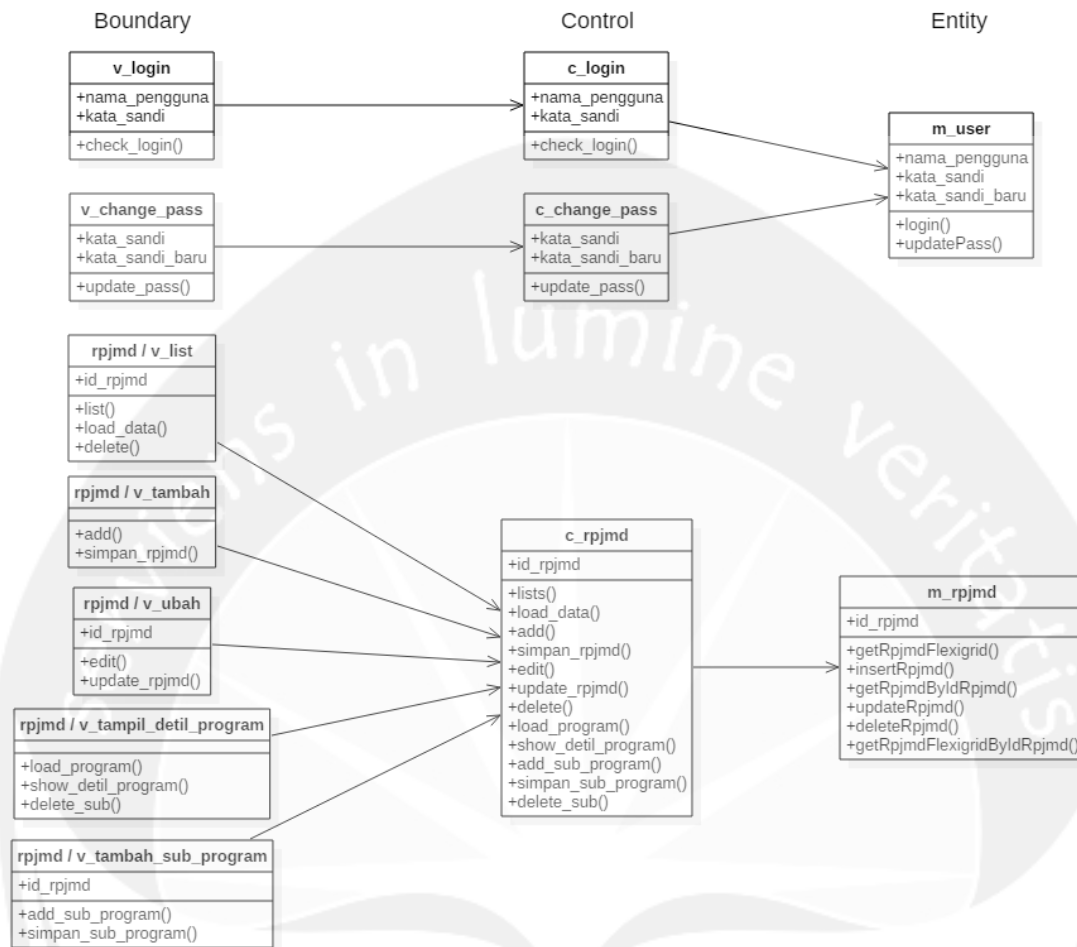
2.2.1.17.4 Hapus Data Sumber Dana

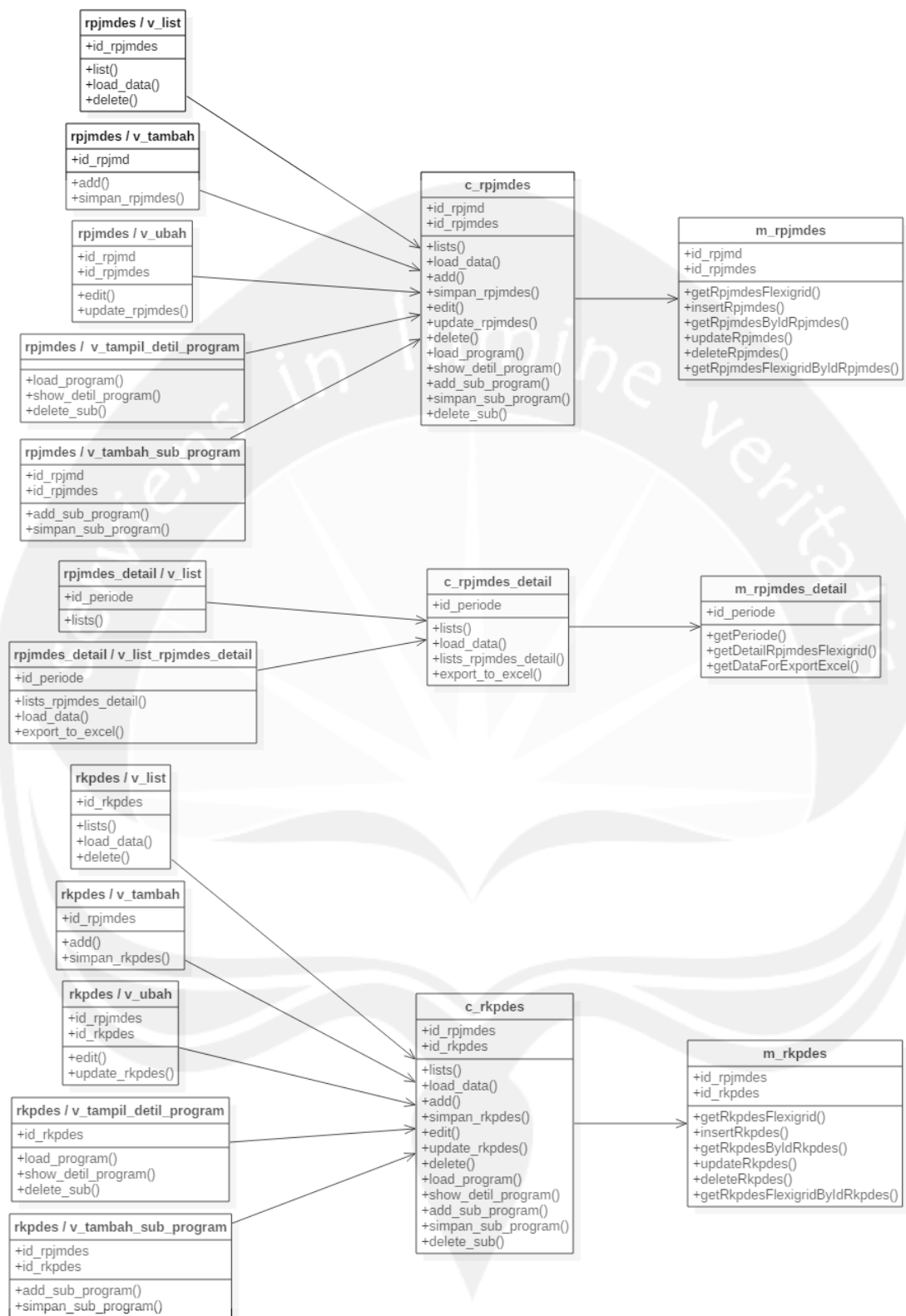


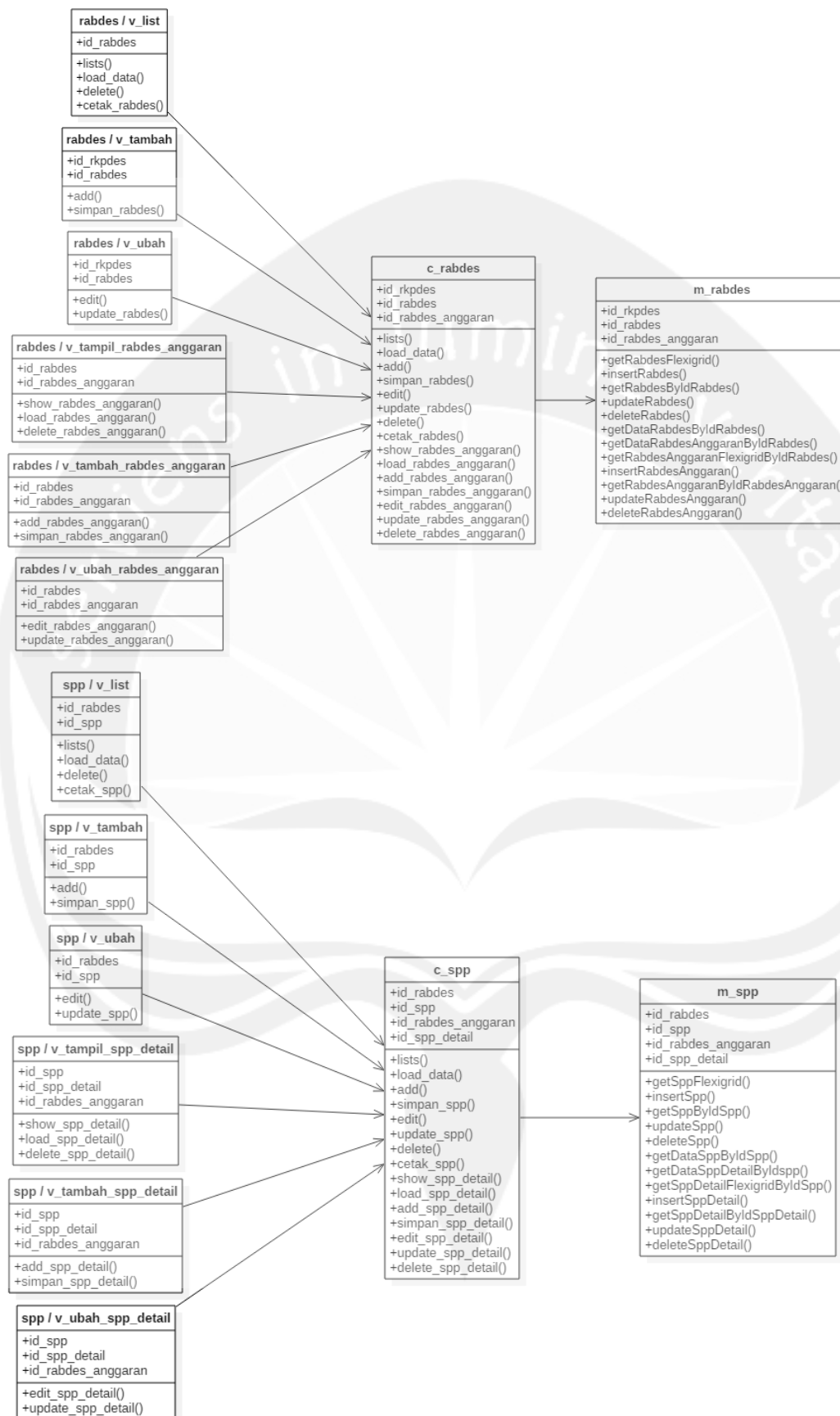
Gambar 2.76 Sequence Diagram: Hapus Data Sumber Dana

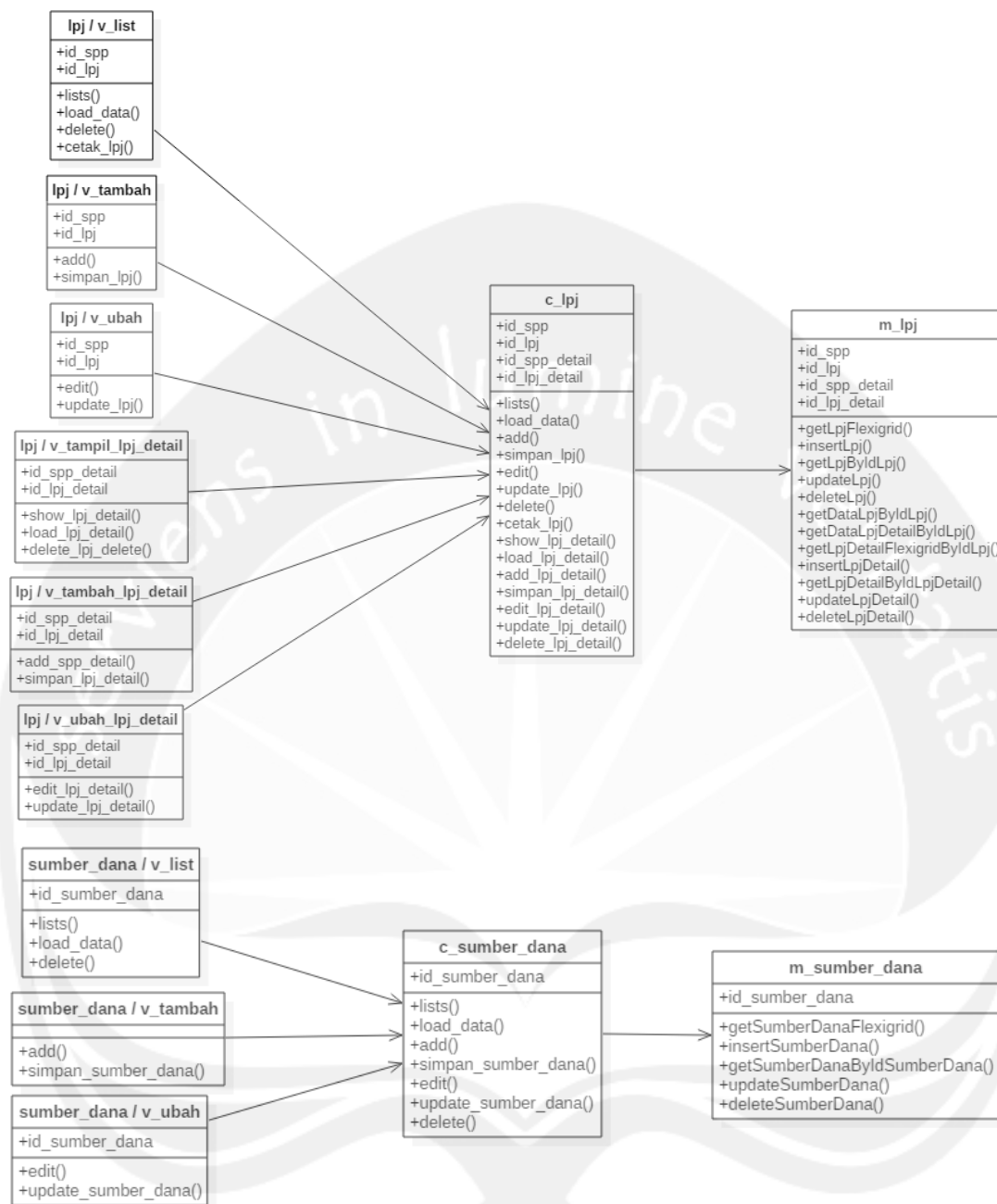


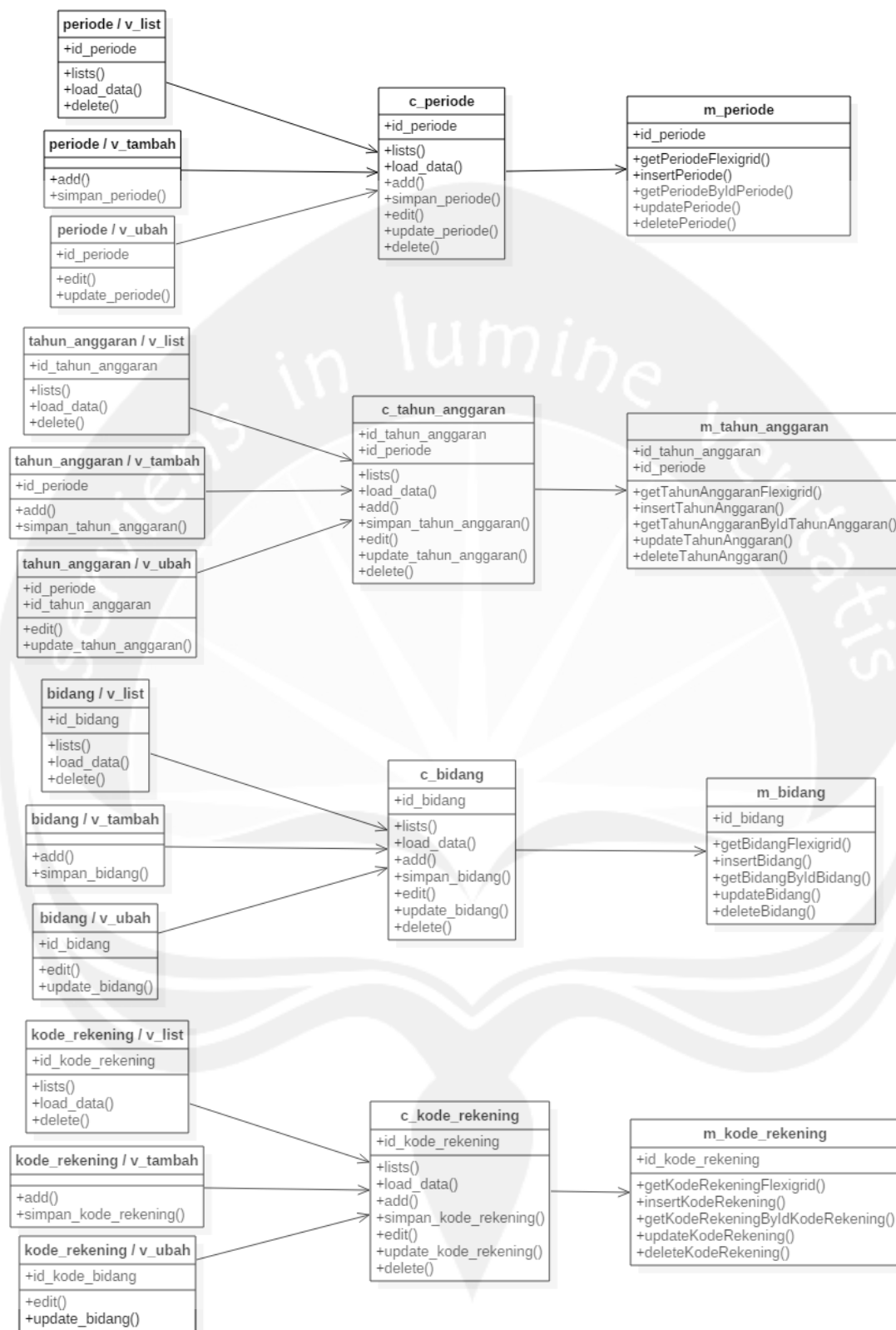
2.2.2 Class Diagram











Gambar 2.77 Class Diagram

2.2.3 Class Diagram Specific Description

2.2.3.1 Specific Design Class v_login

v_login	<<boundary>>
+index() Default konstuktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	

2.2.3.2 Specific Design Class v_change_pass

v_login	<<boundary>>
+index() Default konstuktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +update_pass() Operasi ini digunakan untuk mengubah kata sandi.	

2.2.3.3 Specific Design Class rpjmd/v_list

rpjmd/v_list	<<boundary>>
+lists():Tabel Operasi ini digunakan untuk membentuk data rpjmd beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data rpjmd dengan format table. +delete():void Operasi ini digunakan untuk menghapus data rpjmd secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.4 Specific Design Class rpjmd/v_tambah

rpjmd/v_tambah	<<boundary>>
+add():void	

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data rpjmd baru.

+simpan_rpjmd():void

Operasi ini digunakan untuk menyimpan data rpjmd yang baru dibuat.

2.2.3.5 Specific Design Class rpjmd/v_ubah

c_rpjmd/v_ubah	<<boundary>>
+edit():void Operasi ini digunakan untuk menampilkan form ubah beserta data rpjmd yang spesifik sesuai dengan nilai yang diterima. +update_rpjmd():void Operasi ini digunakan untuk menyimpan data rpjmd yang telah mendapatkan pembaharuan.	

2.2.3.6 Specific Design Class rpjmd/v_tampil_detil_program

rpjmd/v_tampil_detil_program	<<boundary>>
+show_detil_program(\$id):Tabel Operasi ini digunakan untuk membentuk data sub program rpjmd beserta pengaturan format table yang akan ditampilkan. +load_program():void Operasi ini digunakan untuk menampilkan data sub program rpjmd dengan format table. +delete_sub():void Operasi ini digunakan untuk menghapus data sub program rpjmd secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.7 Specific Design Class rpjmd/v_tambah_sub_program

rpjmd/v_tambah_sub_program	<<boundary>>
+add_sub_program():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data sub program rpjmd baru.	

+simpan_sub_program():void

Operasi ini digunakan untuk menyimpan data sub program rpjmd yang baru dibuat.

2.2.3.8 Specific Design Class rpjmdes/v_list

rpjmdes/v_list	<<boundary>>
<pre>+lists():Tabel</pre> Operasi ini digunakan untuk membentuk data rpjmdes beserta pengaturan format table yang akan ditampilkan. <pre>+load_data():void</pre> Operasi ini digunakan untuk menampilkan data rpjmdes dengan format table. <pre>+delete():void</pre> Operasi ini digunakan untuk menghapus data rpjmdes secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.9 Specific Design Class rpjmdes/v_tambah

rpjmdes/v_tambah	<<boundary>>
<pre>+add():void</pre> Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data rpjmdes baru. <pre>+simpan_rpjmdes():void</pre> Operasi ini digunakan untuk menyimpan data rpjmdes yang baru dibuat.	

2.2.3.10 Specific Design Class rpjmdes/v_ubah

c_rpjmdes/v_ubah	<<boundary>>
<pre>+edit():void</pre> Operasi ini digunakan untuk menampilkan form ubah beserta data rpjmdes yang spesifik sesuai dengan nilai yang diterima. <pre>+update_rpjmdes():void</pre> Operasi ini digunakan untuk menyimpan data rpjmdes yang telah mendapatkan pembaharuan.	

2.2.3.11 Specific Design Class rpjmdes/ v_tampil_detil_program

rpjmdes/v_tampil_detil_program	<<boundary>>
<pre>+show_detil_program(\$id):Tabel</pre> Operasi ini digunakan untuk membentuk data sub program rpjmdes beserta pengaturan format table yang akan ditampilkan. <pre>+load_program():void</pre> Operasi ini digunakan untuk menampilkan data sub program rpjmdes dengan format table. <pre>+delete_sub():void</pre> Operasi ini digunakan untuk menghapus data sub program rpjmdes secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.12 Specific Design Class rpjmdes/v_tambah_sub_program

rpjmdes/v_tambah_sub_program	<<boundary>>
<pre>+add_sub_program():void</pre> Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data sub program rpjmdes baru. <pre>+simpan_sub_program():void</pre> Operasi ini digunakan untuk menyimpan data sub program rpjmdes yang baru dibuat.	

2.2.3.13 Specific Design Class rpjmdes_detail/v_list

2.2.3.14 Specific Design Class rpjmdes_detail/ v_list_rpjmdes_detail

2.2.3.15 Specific Design Class rkpdes/v_list

rkpdes/v_list	<<boundary>>
<pre>+lists():Tabel</pre> Operasi ini digunakan untuk membentuk data rkpdes beserta pengaturan format table yang akan ditampilkan. <pre>+load_data():void</pre>	

Operasi ini digunakan untuk menampilkan data rkpdes dengan format table.

+delete():void

Operasi ini digunakan untuk menghapus data rkpdes secara spesifik sesuai dengan nilai yang diterima.

2.2.3.16 Specific Design Class rkpdes/v_tambah

rkpdes/v_tambah	<<boundary>>
<pre>+add():void</pre> Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data rkpdes baru. <pre>+simpan_rkpdes():void</pre> Operasi ini digunakan untuk menyimpan data rkpdes yang baru dibuat.	

2.2.3.17 Specific Design Class rkpdes/v_ubah

c_rkpdes/v_ubah	<<boundary>>
<pre>+edit():void</pre> Operasi ini digunakan untuk menampilkan form ubah beserta data rkpdes yang spesifik sesuai dengan nilai yang diterima. <pre>+update_rkpdes():void</pre> Operasi ini digunakan untuk menyimpan data rkpdes yang telah mendapatkan pembaharuan.	

2.2.3.18 Specific Design Class rkpdes/ v_tampil_detil_program

rkpdes/v_tampil_detil_program	<<boundary>>
<pre>+show_detil_program(\$id):Tabel</pre> Operasi ini digunakan untuk membentuk data sub program rkpdes beserta pengaturan format table yang akan ditampilkan. <pre>+load_program():void</pre> Operasi ini digunakan untuk menampilkan data sub program rkpdes dengan format table. <pre>+delete_sub():void</pre>	

Operasi ini digunakan untuk menghapus data sub program rkpdes secara spesifik sesuai dengan nilai yang diterima.

2.2.3.19 Specific Design Class rkpdes/v_tambah_sub_program

rkpdes/v_tambah_sub_program	<<boundary>>
<pre>+add_sub_program():void</pre> Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data sub program rkpdes baru. <pre>+simpan_sub_program():void</pre> Operasi ini digunakan untuk menyimpan data sub program rkpdes yang baru dibuat.	

2.2.3.20 Specific Design Class rabdes/v_list

rabdes/v_list	<<boundary>>
<pre>+lists():Tabel</pre> Operasi ini digunakan untuk membentuk data rabdes beserta pengaturan format table yang akan ditampilkan. <pre>+load_data():void</pre> Operasi ini digunakan untuk menampilkan data rabdes dengan format table. <pre>+delete():void</pre> Operasi ini digunakan untuk menghapus data rabdes secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.21 Specific Design Class rabdes/v_tambah

rabdes/v_tambah	<<boundary>>
<pre>+add():void</pre> Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data rabdes baru. <pre>+simpan_rabdes():void</pre> Operasi ini digunakan untuk menyimpan data rabdes yang baru dibuat.	

2.2.3.22 Specific Design Class rabdes/v_ubah

rabdes/v_ubah	<<boundary>>
<pre>+edit():void</pre> Operasi ini digunakan untuk menampilkan form ubah beserta data rabdes yang spesifik sesuai dengan nilai yang diterima. <pre>+update_rabdes():void</pre> Operasi ini digunakan untuk menyimpan data rabdes yang telah mendapatkan pembaharuan.	

2.2.3.23 Specific Design Class rabdes/v_tampil_rabdes_anggaran

rabdes/v_tampil_rabdes_anggaran	<<boundary>>
<pre>+show_rabdes_anggaran(\$id):Tabel</pre> Operasi ini digunakan untuk membentuk data rabdes anggaran beserta pengaturan format table yang akan ditampilkan. <pre>+load_rabdes_anggaran():void</pre> Operasi ini digunakan untuk menampilkan data rabdes anggaran dengan format table. <pre>+delete_rabdes_anggaran():void</pre> Operasi ini digunakan untuk menghapus data rabdes anggaran secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.24 Specific Design Class rabdes/v_tambah_rabdes_anggaran

rabdes/v_tambah_rabdes_anggaran	<<boundary>>
<pre>+add_rabdes_anggaran():void</pre> Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data rabdes anggaran baru. <pre>+simpan_rabdes_anggaran():void</pre> Operasi ini digunakan untuk menyimpan data rabdes anggaran yang baru dibuat.	

2.2.3.25 Specific Design Class rabdes/ v_ubah_rabdes_anggaran

rabdes/v_ubah_rabdes_anggaran	<<boundary>>
<pre>+edit():void</pre> Operasi ini digunakan untuk menampilkan form ubah beserta data rabdes yang spesifik sesuai dengan nilai yang diterima. <pre>+update_rabdes():void</pre> Operasi ini digunakan untuk menyimpan data rabdes yang telah mendapatkan pembaharuan.	

2.2.3.26 Specific Design Class spp/v_list

spp/v_list	<<boundary>>
<pre>+lists():Tabel</pre> Operasi ini digunakan untuk membentuk data spp beserta pengaturan format table yang akan ditampilkan. <pre>+load_data():void</pre> Operasi ini digunakan untuk menampilkan data spp dengan format table. <pre>+delete():void</pre> Operasi ini digunakan untuk menghapus data spp secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.27 Specific Design Class spp/v_tambah

spp/v_tambah	<<boundary>>
<pre>+add():void</pre> Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data spp baru. <pre>+simpan_spp():void</pre> Operasi ini digunakan untuk menyimpan data spp yang baru dibuat.	

2.2.3.28 Specific Design Class spp/v_ubah

c_spp/v_ubah	<<boundary>>
---------------------	---------------------------------

Program Studi Teknik Informatika	DPPL-SIDeKa-PPD	62/ 111
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		

+edit():void

Operasi ini digunakan untuk menampilkan form ubah beserta data spp yang spesifik sesuai dengan nilai yang diterima.

+update_spp():void

Operasi ini digunakan untuk menyimpan data spp yang telah mendapatkan pembaharuan.

2.2.3.29 Specific Design Class spp/v_tampil_spp_detail

spp/v_tampil_spp_detail	<<boundary>>
+show_spp_detail(\$id):Tabel	
Operasi ini digunakan untuk membentuk data spp detail beserta pengaturan format table yang akan ditampilkan.	
+load_spp_detail():void	
Operasi ini digunakan untuk menampilkan data spp detail dengan format table.	
+delete_spp_detail():void	
Operasi ini digunakan untuk menghapus data spp detail secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.30 Specific Design Class spp/v_tambah_spp_detail

spp/v_tambah_spp_detail	<<boundary>>
+add_spp_detail():void	
Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data spp detail baru.	
+simpan_spp_detail():void	
Operasi ini digunakan untuk menyimpan data spp detail yang baru dibuat.	

2.2.3.31 Specific Design Class spp/v_ubah_spp_detail

2.2.3.32 Specific Design Class lpj/v_list

lpj/v_list	<<boundary>>

+lists():Tabel
Operasi ini digunakan untuk membentuk data lpj beserta pengaturan format table yang akan ditampilkan.
+load_data():void
Operasi ini digunakan untuk menampilkan data lpj dengan format table.
+delete():void
Operasi ini digunakan untuk menghapus data lpj secara spesifik sesuai dengan nilai yang diterima.

2.2.3.33 Specific Design Class lpj/v_tambah

lpj/v_tambah	<<boundary>>
+add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data lpj baru. +simpan_lpj():void Operasi ini digunakan untuk menyimpan data lpj yang baru dibuat.	

2.2.3.34 Specific Design Class lpj/v_ubah

c_lpj/v_ubah	<<boundary>>
+edit():void Operasi ini digunakan untuk menampilkan form ubah beserta data lpj yang spesifik sesuai dengan nilai yang diterima. +update_lpj():void Operasi ini digunakan untuk menyimpan data lpj yang telah mendapatkan pembaharuan.	

2.2.3.35 Specific Design Class lpj/v_tampil_lpj_detail

lpj/v_tampil_lpj_detail	<<boundary>>
+show_lpj_detail(\$id):Tabel Operasi ini digunakan untuk membentuk data lpj detail beserta pengaturan format table yang akan ditampilkan. +load_lpj_detail():void	

Operasi ini digunakan untuk menampilkan data lpj detail dengan format table.

+delete_lpj_detail():void

Operasi ini digunakan untuk menghapus data lpj detail secara spesifik sesuai dengan nilai yang diterima.

2.2.3.36 Specific Design Class lpj/v_tambah_lpj_detail

lpj/v_tambah_lpj_detail	<<boundary>>
+add_lpj_detail():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data lpj detail baru.	
+simpan_lpj_detail():void Operasi ini digunakan untuk menyimpan data lpj detail yang baru dibuat.	

2.2.3.37 Specific Design Class lpj/v_ubah_lpj_detail

2.2.3.38 Specific Design Class periode/v_list

periode/v_list	<<boundary>>
+lists():Tabel Operasi ini digunakan untuk membentuk data periode beserta pengaturan format table yang akan ditampilkan.	
+load_data():void Operasi ini digunakan untuk menampilkan data periode dengan format table.	
+delete():void Operasi ini digunakan untuk menghapus data periode secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.39 Specific Design Class periode/v_tambah

periode/v_tambah	<<boundary>>
+add():void	

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data periode baru.

+simpan_periode():void

Operasi ini digunakan untuk menyimpan data periode yang baru dibuat.

2.2.3.40 Specific Design Class periode/v_ubah

c_periode/v_ubah	<<boundary>>
+edit():void Operasi ini digunakan untuk menampilkan form ubah beserta data periode yang spesifik sesuai dengan nilai yang diterima. +update_periode():void Operasi ini digunakan untuk menyimpan data periode yang telah mendapatkan pembaharuan.	

2.2.3.41 Specific Design Class tahun_anggaran/v_list

tahun_anggaran/v_list	<<boundary>>
+lists():Tabel Operasi ini digunakan untuk membentuk data tahun anggaran beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data tahun anggaran dengan format table. +delete():void Operasi ini digunakan untuk menghapus data tahun anggaran secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.42 Specific Design Class tahun_anggaran/v_tambah

tahun_anggaran/v_tambah	<<boundary>>
+add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data tahun anggaran baru. +simpan_tahun_anggaran():void	

Operasi ini digunakan untuk menyimpan data tahun anggaran yang baru dibuat.

2.2.3.43 Specific Design Class tahun_anggaran/v_ubah

tahun_anggaran/v_ubah	<<boundary>>
+edit():void Operasi ini digunakan untuk menampilkan form ubah beserta data tahun anggaran yang spesifik sesuai dengan nilai yang diterima. +update_tahun_anggaran():void Operasi ini digunakan untuk menyimpan data tahun anggaran yang telah mendapatkan pembaharuan.	

2.2.3.44 Specific Design Class bidang/v_list

bidang/v_list	<<boundary>>
+lists():Tabel Operasi ini digunakan untuk membentuk data bidang beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data bidang dengan format table. +delete():void Operasi ini digunakan untuk menghapus data bidang secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.45 Specific Design Class bidang/v_tambah

bidang/v_tambah	<<boundary>>
+add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data bidang baru. +simpan_bidang():void Operasi ini digunakan untuk menyimpan data bidang yang baru dibuat.	

2.2.3.46 Specific Design Class bidang/v_ubah

c_bidang/v_ubah	<<boundary>>
+edit():void Operasi ini digunakan untuk menampilkan form ubah beserta data bidang yang spesifik sesuai dengan nilai yang diterima. +update_bidang():void Operasi ini digunakan untuk menyimpan data bidang yang telah mendapatkan pembaharuan.	

2.2.3.47 Specific Design Class kode_rekening/v_list

kode_rekening/v_list	<<boundary>>
+lists():Tabel Operasi ini digunakan untuk membentuk data kode rekening beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data kode rekening dengan format table. +delete():void Operasi ini digunakan untuk menghapus data kode rekening secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.48 Specific Design Class kode_rekening/v_tambah

kode_rekening/v_tambah	<<boundary>>
+add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data kode rekening baru. +simpan_kode_rekening():void Operasi ini digunakan untuk menyimpan data kode rekening yang baru dibuat.	

2.2.3.49 Specific Design Class kode_rekening/v_ubah

kode_rekening/v_ubah	<<boundary>>
+edit():void	

Operasi ini digunakan untuk menampilkan form ubah beserta data kode rekening yang spesifik sesuai dengan nilai yang diterima.

+update_kode_rekening():void

Operasi ini digunakan untuk menyimpan data kode rekening yang telah mendapatkan pembaharuan.

2.2.3.50 Specific Design Class sumber_dana/v_list

sumber_dana/v_list	<<boundary>>
+lists():Tabel Operasi ini digunakan untuk membentuk data sumber dana beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data sumber dana dengan format table. +delete():void Operasi ini digunakan untuk menghapus data sumber dana secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.51 Specific Design Class sumber_dana/v_tambah

sumber_dana/v_tambah	<<boundary>>
+add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data sumber dana baru. +simpan_sumber_dana():void Operasi ini digunakan untuk menyimpan data sumber dana yang baru dibuat.	

2.2.3.52 Specific Design Class sumber_dana/v_ubah

sumber_dana/v_ubah	<<boundary>>
+edit():void	

Operasi ini digunakan untuk menampilkan form ubah beserta data sumber dana yang spesifik sesuai dengan nilai yang diterima.

+update_sumber_dana():void

Operasi ini digunakan untuk menyimpan data sumber dana yang telah mendapatkan pembaharuan.

2.2.3.53 Specific Design Class c_login

c_login	<<control>>
<pre>+index() Default konstuktur, digunakan untuk inisialisasi semua atribut dari kelas ini. +checkLogin() Operasi ini digunakan untuk mengecek apakah nama pengguna dan kata sandi telah terdaftar pada system.</pre>	

2.2.3.54 Specific Design Class c_changePass

c_changePass	<<control>>
<pre>+index() Default konstuktur, digunakan untuk inisialisasi semua atribut dari kelas ini. +update_pass() Operasi ini digunakan untuk mengubah kata sandi.</pre>	

2.2.3.55 Specific Design Class c_rpjmd

c_rpjmd	<<control>>
<pre>+lists():Tabel Operasi ini digunakan untuk membentuk data rpjmd beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data rpjmd dengan format table.</pre>	

+add():void

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data rpjmd baru.

+simpan_rpjmd():void

Operasi ini digunakan untuk menyimpan data rpjmd yang baru dibuat.

+edit(\$id):void

Operasi ini digunakan untuk menampilkan form ubah beserta data rpjmd yang spesifik sesuai dengan nilai yang diterima.

+update_rpjmd():void

Operasi ini digunakan untuk menyimpan data rpjmd yang telah mendapatkan pembaharuan.

+delete():void

Operasi ini digunakan untuk menghapus data rpjmd secara spesifik sesuai dengan nilai yang diterima.

+load_program():void

Operasi ini digunakan untuk membentuk data sub program rpjmd beserta pengaturan format table yang akan ditampilkan.

+show_detil_program():void

Operasi ini digunakan untuk menampilkan data sub program rpjmd dengan format table.

+add_sub_program():void

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data sub program rpjmd baru.

+simpan_sub_program():void

Operasi ini digunakan untuk menyimpan data sub program rpjmd yang baru dibuat.

+delete_sub():void

Operasi ini digunakan untuk menghapus data sub program rpjmd secara spesifik sesuai dengan nilai yang diterima.

2.2.3.56 Specific Design Class c_rpjmdes

c_rpjmdes	<<control>>
+lists():Tabel	

Program Studi Teknik Informatika	DPPL-SiDeKa-PPD	71/ 111
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		

Operasi ini digunakan untuk membentuk data rpjmdes beserta pengaturan format table yang akan ditampilkan.

+load_data():void

Operasi ini digunakan untuk menampilkan data rpjmdes dengan format table.

+add():void

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data rpjmdes baru.

+simpan_rpjmdes():void

Operasi ini digunakan untuk menyimpan data rpjmdes yang baru dibuat.

+edit(\$id):void

Operasi ini digunakan untuk menampilkan form ubah beserta data rpjmdes yang spesifik sesuai dengan nilai yang diterima.

+update_rpjmdes():void

Operasi ini digunakan untuk menyimpan data rpjmdes yang telah mendapatkan pembaharuan.

+delete():void

Operasi ini digunakan untuk menghapus data rpjmdes secara spesifik sesuai dengan nilai yang diterima.

+load_program():void

Operasi ini digunakan untuk membentuk data sub program rpjmdes beserta pengaturan format table yang akan ditampilkan.

+show_detil_program():void

Operasi ini digunakan untuk menampilkan data sub program rpjmdes dengan format table.

+add_sub_program():void

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data sub program rpjmdes baru.

+simpan_sub_program():void

Operasi ini digunakan untuk menyimpan data sub program rpjmdes yang baru dibuat.

+delete_sub():void

Operasi ini digunakan untuk menghapus data sub program rpjmdes secara spesifik sesuai dengan nilai yang diterima.

2.2.3.57 Specific Design Class c_rpjmdes_detail

c_rpjmdes_detail	<<control>>
<pre>+lists():Tabel</pre> Operasi ini digunakan untuk membentuk data detail rpjmdes beserta pengaturan format table yang akan ditampilkan. <pre>+load_data():void</pre> Operasi ini digunakan untuk menampilkan data detail rpjmdes dengan format table. <pre>+lists():Tabel</pre> Operasi ini digunakan untuk membentuk data detail rpjmdes sesuai dengan periode yang dipilih beserta pengaturan format table yang akan ditampilkan. <pre>+export_to_excel():void</pre> Operasi ini digunakan untuk membentuk data detail rpjmdes sesuai dengan periode yang dipilih dengan format excel.	

2.2.3.58 Specific Design Class c_rkpdes

c_rkpdes	<<control>>
<pre>+lists():Tabel</pre> Operasi ini digunakan untuk membentuk data rkpdes beserta pengaturan format table yang akan ditampilkan. <pre>+load_data():void</pre> Operasi ini digunakan untuk menampilkan data rkpdes dengan format table. <pre>+add():void</pre> Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data rkpdes baru. <pre>+simpan_rkpdes():void</pre> Operasi ini digunakan untuk menyimpan data rkpdes yang baru dibuat. <pre>+edit(\$id):void</pre> Operasi ini digunakan untuk menampilkan form ubah beserta data rkpdes yang spesifik sesuai dengan nilai yang diterima.	

Program Studi Teknik Informatika	DPPL-SiDeKa-PPD	73/ 111
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		

+update_rkpdes():void
Operasi ini digunakan untuk menyimpan data rkpdes yang telah mendapatkan pembaharuan.
+delete():void
Operasi ini digunakan untuk menghapus data rkpdes secara spesifik sesuai dengan nilai yang diterima.
+load_program():void
Operasi ini digunakan untuk membentuk data sub program rkpdes beserta pengaturan format table yang akan ditampilkan.
+show_detil_program(\$id):void
Operasi ini digunakan untuk menampilkan data sub program rkpdes dengan format table.
+add_sub_program():void
Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data sub program rkpdes baru.
+simpan_sub_program():void
Operasi ini digunakan untuk menyimpan data sub program rkpdes yang baru dibuat.
+delete_sub():void
Operasi ini digunakan untuk menghapus data sub program rkpdes secara spesifik sesuai dengan nilai yang diterima.

2.2.3.59 Specific Design Class c_rabdes

c_rabdes	<<control>>
+lists():Tabel Operasi ini digunakan untuk membentuk data rabdes beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data rabdes dengan format table. +add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data rabdes baru. +simpan_rabdes():void	

Operasi ini digunakan untuk menyimpan data rabdes yang baru dibuat.

+edit(\$id):void

Operasi ini digunakan untuk menampilkan form ubah beserta data rabdes yang spesifik sesuai dengan nilai yang diterima.

+update_rabdes():void

Operasi ini digunakan untuk menyimpan data rabdes yang telah mendapatkan pembaharuan.

+delete():void

Operasi ini digunakan untuk menghapus data rabdes secara spesifik sesuai dengan nilai yang diterima.

+cetak_rabdes(\$id):void

Operasi ini digunakan untuk mencetak data rabdes dengan format pdf.

+show_rabdes_anggaran(\$id):Table

Operasi ini digunakan untuk membentuk data rabdes anggaran beserta pengaturan format table yang akan ditampilkan.

+load_rabdes_anggaran():void

Operasi ini digunakan untuk menampilkan data rabdes anggaran dengan format table.

+add_rabdes_anggaran():void

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data rabdes anggaran baru.

+simpan_rabdes_anggaran():void

Operasi ini digunakan untuk menyimpan data rabdes anggaran yang baru dibuat.

+edit_rabdes_anggaran():void

Operasi ini digunakan untuk menampilkan form ubah beserta data rabdes anggaran yang spesifik sesuai dengan nilai yang diterima.

+update_rabdes_anggaran(\$id):void

Operasi ini digunakan untuk menyimpan data rabdes anggaran yang telah mendapatkan pembaharuan.

+delete_rabdes_anggaran():void

Operasi ini digunakan untuk menghapus data rabdes anggaran secara spesifik sesuai dengan nilai yang diterima.

2.2.3.60 Specific Design Class c_spp

c_spp	<<control>>
<pre>+lists():Tabel Operasi ini digunakan untuk membentuk data spp beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data spp dengan format table. +add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data spp baru. +simpan_spp():void Operasi ini digunakan untuk menyimpan data spp yang baru dibuat. +edit(\$id):void Operasi ini digunakan untuk menampilkan form ubah beserta data spp yang spesifik sesuai dengan nilai yang diterima. +update_spp():void Operasi ini digunakan untuk menyimpan data spp yang telah mendapatkan pembaharuan. +delete():void Operasi ini digunakan untuk menghapus data spp secara spesifik sesuai dengan nilai yang diterima. +cetak_spp():void Operasi ini digunakan untuk mencetak data spp dengan format pdf. +show_spp_detail(\$id):Table Operasi ini digunakan untuk membentuk data spp detail beserta pengaturan format table yang akan ditampilkan. +load_spp_detail():void</pre>	

Operasi ini digunakan untuk menampilkan data spp detail dengan format table.

+add_spp_detail():void

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data spp detail baru.

+simpan_spp_detail():void

Operasi ini digunakan untuk menyimpan data spp detail yang baru dibuat.

+edit_spp_detail(\$id):void

Operasi ini digunakan untuk menampilkan form ubah beserta data spp detail yang spesifik sesuai dengan nilai yang diterima.

+update_spp_detail():void

Operasi ini digunakan untuk menyimpan data spp detail yang telah mendapatkan pembaharuan.

+delete_spp_detail():void

Operasi ini digunakan untuk menghapus data spp detail secara spesifik sesuai dengan nilai yang diterima.

2.2.3.61 Specific Design Class c_lpj

c_lpj	<<control>>
+lists():Tabel Operasi ini digunakan untuk membentuk data lpj beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data lpj dengan format table. +add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data lpj baru. +simpan_lpj():void Operasi ini digunakan untuk menyimpan data lpj yang baru dibuat. +edit(\$id):void	

Operasi ini digunakan untuk menampilkan form ubah beserta data lpj yang spesifik sesuai dengan nilai yang diterima.

+update_lpj():void

Operasi ini digunakan untuk menyimpan data lpj yang telah mendapatkan pembaharuan.

+delete():void

Operasi ini digunakan untuk menghapus data lpj secara spesifik sesuai dengan nilai yang diterima.

+cetak_lpj():void

Operasi ini digunakan untuk mencetak data lpj kedalam bentuk pdf.

+show_lpj_detail(\$id):Table

Operasi ini digunakan untuk membentuk data lpj detail beserta pengaturan format table yang akan ditampilkan.

+load_lpj_detail():void

Operasi ini digunakan untuk menampilkan data spp detail dengan format table.

+add_lpj_detail():void

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data lpj detail baru.

+simpan_lpj_detail():void

Operasi ini digunakan untuk menyimpan data lpj detail yang baru dibuat.

+edit_lpj_detail(\$id):void

Operasi ini digunakan untuk menampilkan form ubah beserta data lpj detail yang spesifik sesuai dengan nilai yang diterima.

+update_lpj_detail():void

Operasi ini digunakan untuk menyimpan data lpj detail yang telah mendapatkan pembaharuan.

+delete_lpj_detail():void

Operasi ini digunakan untuk menghapus data lpj detail secara spesifik sesuai dengan nilai yang diterima.

2.2.3.62 Specific Design Class c_periode

c_periode	<<control>>
<pre>+lists():Tabel Operasi ini digunakan untuk membentuk data periode beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data periode dengan format table. +add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data periode baru. +simpan_periode():void Operasi ini digunakan untuk menyimpan data periode yang baru dibuat. +edit(\$id):void Operasi ini digunakan untuk menampilkan form ubah beserta data periode yang spesifik sesuai dengan nilai yang diterima. +update_periode():void Operasi ini digunakan untuk menyimpan data periode yang telah mendapatkan pembaharuan. +delete():void Operasi ini digunakan untuk menghapus data periode secara spesifik sesuai dengan nilai yang diterima.</pre>	

2.2.3.63 Specific Design Class c_tahun_anggaran

c_tahun_anggaran	<<control>>
<pre>+lists():Tabel Operasi ini digunakan untuk membentuk data tahun anggaran beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data tahun anggaran dengan format table. +add():void</pre>	

Program Studi Teknik Informatika	DPPL-SiDeKa-PPD	79/ 111
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		

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data tahun anggaran baru.

+simpan_tahun_anggaran():void

Operasi ini digunakan untuk menyimpan data tahun anggaran yang baru dibuat.

+edit(\$id):void

Operasi ini digunakan untuk menampilkan form ubah beserta data tahun anggaran yang spesifik sesuai dengan nilai yang diterima.

+update_tahun_anggaran():void

Operasi ini digunakan untuk menyimpan data tahun anggaran yang telah mendapatkan pembaharuan.

+delete():void

Operasi ini digunakan untuk menghapus data tahun anggaran secara spesifik sesuai dengan nilai yang diterima.

2.2.3.64 Specific Design Class c_bidang

c_bidang	<<control>>
+lists():Tabel Operasi ini digunakan untuk membentuk data bidang beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data bidang dengan format table. +add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data bidang baru. +simpan_bidang():void Operasi ini digunakan untuk menyimpan data bidang yang baru dibuat. +edit(\$id):void Operasi ini digunakan untuk menampilkan form ubah beserta data bidang yang spesifik sesuai dengan nilai yang diterima. +update_bidang():void	

Operasi ini digunakan untuk menyimpan data bidang yang telah mendapatkan pembaharuan.

+delete():void

Operasi ini digunakan untuk menghapus data bidang secara spesifik sesuai dengan nilai yang diterima.

2.2.3.65 Specific Design Class c_kode_rekening

c_kode_rekening	<<control>>
+lists():Tabel Operasi ini digunakan untuk membentuk data kode rekening beserta pengaturan format table yang akan ditampilkan. +load_data():void Operasi ini digunakan untuk menampilkan data kode rekening dengan format table. +add():void Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data kode rekening baru. +simpan_kode_rekening():void Operasi ini digunakan untuk menyimpan data kode rekening yang baru dibuat. +edit(\$id):void Operasi ini digunakan untuk menampilkan form ubah beserta data kode rekening yang spesifik sesuai dengan nilai yang diterima. +update_kode_rekening():void Operasi ini digunakan untuk menyimpan data kode rekening yang telah mendapatkan pembaharuan. +delete():void Operasi ini digunakan untuk menghapus data kode rekening secara spesifik sesuai dengan nilai yang diterima.	

2.2.3.66 Specific Design Class c_sumber_dana

c_sumber_dana	<<control>>

+lists():Tabel

Operasi ini digunakan untuk membentuk data sumber dana beserta pengaturan format table yang akan ditampilkan.

+load_data():void

Operasi ini digunakan untuk menampilkan data sumber dana dengan format table.

+add():void

Operasi ini digunakan untuk menampilkan form tambah yang digunakan untuk membuat data sumber dana baru.

+simpan_sumber_dana():void

Operasi ini digunakan untuk menyimpan data sumber dana yang baru dibuat.

+edit(\$id):void

Operasi ini digunakan untuk menampilkan form ubah beserta data sumber dana yang spesifik sesuai dengan nilai yang diterima.

+update_sumber_dana():void

Operasi ini digunakan untuk menyimpan data sumber dana yang telah mendapatkan pembaharuan.

+delete():void

Operasi ini digunakan untuk menghapus data sumber dana secara spesifik sesuai dengan nilai yang diterima.

2.2.3.67 Specific Design Class m_login

m_login	<<entity>>
+login(\$nama_pengguna, \$kata_sandi):void	
Operasi ini digunakan untuk mengambil dan mengecek data pengguna yang telah terdaftar pada system.	

2.2.3.68 Specific Design Class m_user

m_user	<<entity>>
+updateUser(\$where,\$data):void	

Operasi ini digunakan untuk mengubah data pengguna yang spesifik sesuai dengan nilai yang diterima.

2.2.3.69 Specific Design Class m_rpjmd

m_rpjmd	<<entity>>
<pre> +getRpjmdFlexigrid():Table Operasi ini digunakan untuk menampilkan data rpjmd. +insertRpjmd(\$data):void Operasi ini digunakan untuk menambah data rpjmd baru. +getRpjmdByIdRpjmd(\$id_rpjmd):void Operasi ini digunakan untuk mengambil data rpjmd yang dipergunakan pada saat akan melakukan ubah data rpjmd. +updateRpjmd(\$where,\$data):void Operasi ini digunakan untuk mengubah data rpjmd. +deleteRpjmd(\$id):void Operasi ini digunakan untuk menghapus data rpjmd. +getRpjmdFlexigridByIdRpjmd(\$id_rpjmd):Table Operasi ini digunakan untuk menampilkan data rpjmd berdasarkan data rpjmd yang dipilih. </pre>	

2.2.3.70 Specific Design Class m_rpjmDES

m_rpjmDES	<<entity>>
<pre> +getRpjmDESFlexigrid():Table Operasi ini digunakan untuk menampilkan data rpjmDES. +insertRpjmDES(\$data):void Operasi ini digunakan untuk menambah data rpjmDES baru. +getRpjmDESByIdRpjmDES(\$id_rpjmDES):void Operasi ini digunakan untuk mengambil data rpjmDES yang dipergunakan pada saat akan melakukan ubah data rpjmDES. +updateRpjmDES((\$where,\$data):void Operasi ini digunakan untuk mengubah data rpjmDES. +deleteRpjmDES(\$id):void Operasi ini digunakan untuk menghapus data rpjmDES. </pre>	

```
+getRpjmdesFlexigridByIdRpjmdes($id_rpjmdes):Table
```

Operasi ini digunakan untuk menampilkan data rpjmdes berdasarkan data rpjmdes yang dipilih.

2.2.3.71 Specific Design Class m_rpjmdes_detail

m_rpjmdes_detail	<<entity>>
<pre>+getPeriode():void</pre> Operasi ini digunakan untuk mengambil data periode. <pre>+getDetailRpjmdesFlexigrid(\$id_periode):Table</pre> Operasi ini digunakan untuk mengambil data rpjmdes detail berdasarkan periode yang dipilih. <pre>+getDataForExportExcel():void</pre> Operasi ini digunakan untuk mengambil data rpjmdes guna mencetak data rpjmdes yang dipilih ke dalam bentuk excel.	

2.2.3.72 Specific Design Class m_rkpdes

m_rkpdes	<<entity>>
<pre>+getRkpdesFlexigrid():Table</pre> Operasi ini digunakan untuk menampilkan data rkpdes. <pre>+insertRkpdes(\$data):void</pre> Operasi ini digunakan untuk menambah data rkpdes baru. <pre>+getRkpdesByIdRkpdes(\$id_rkpdes):void</pre> Operasi ini digunakan untuk mengambil data rkpdes yang dipergunakan pada saat akan melakukan ubah data rkpdes. <pre>+updateRkpdes(\$where,\$data):void</pre> Operasi ini digunakan untuk mengubah data rkpdes. <pre>+deleteRkpdes(\$id):void</pre> Operasi ini digunakan untuk menghapus data rkpdes. <pre>+getRkpdesFlexigridByIdRkpdes(\$id_rkpdes):Table</pre> Operasi ini digunakan untuk menampilkan data rkpdes berdasarkan data rkpdes yang dipilih.	

2.2.3.73 Specific Design Class m_rabdes

m_rabdes	<<entity>>
----------	------------

Program Studi Teknik Informatika	DPPL-SiDeKa-PPD	84/ 111
----------------------------------	-----------------	---------

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

<code>+getRabdesFlexigrid():Table</code> Operasi ini digunakan untuk menampilkan data Rabdes. <code>+insertRabdes(\$data):void</code> Operasi ini digunakan untuk menambah data Rabdes baru. <code>+getRabdesByIdRabdes(\$id_Rabdes):void</code> Operasi ini digunakan untuk mengambil data Rabdes yang dipergunakan pada saat akan melakukan ubah data Rabdes. <code>+updateRabdes(\$where,\$data):void</code> Operasi ini digunakan untuk mengubah data Rabdes. <code>+deleteRabdes(\$id):void</code> Operasi ini digunakan untuk menghapus data Rabdes. <code>+getDataRabdesByIdRabdes(\$id_rabdes):void</code> Operasi ini digunakan untuk mengambil data rabdes berdasarkan data yang dipilih. <code>+getDataRabdesAnggaranByIdRabdes(\$id_rabdes):void</code> Operasi ini digunakan untuk mengambil data rabdes anggaran berdasarkan rabdes yang dipilih. <code>+getRabdesAnggaranFlexigridByIdRabdes(\$id_rabdes): void</code> Operasi ini digunakan untuk menampilkan data Rabdes Anggaran berdasarkan dengan rabdes yang dipilih. <code>+insertRabdesAnggaran(\$data):void</code> Operasi ini digunakan untuk menambah data Rabdes anggaran baru. <code>+getRabdesAnggaranByIdRabdesAnggaran(\$id_rabdes_anggaran):void</code> Operasi ini digunakn untuk mengambil data Rabdes Anggaran yang dipergunakan pada saat akan melakukan ubah data rabdes anggaran. <code>+updateRabdesAnggaran(\$where,\$data):void</code> Operasi ini digunakan untuk mengubah data rabdes anggaran. <code>+deleteRabdesAnggaran(\$id):void</code> Operasi ini digunakan untuk menghapus data rabdes anggaran.

2.2.3.74 Specific Design Class m_spp

m_spp	<<entity>>
<code>+getSppFlexigrid():Table</code>	

Program Studi Teknik Informatika	DPPL-SiDeKa-PPD	85/ 111
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		

Operasi ini digunakan untuk menampilkan data Spp.

```
+insertSpp($data):void
```

Operasi ini digunakan untuk menambah data Spp baru.

```
+getSppByIdSpp($id_spp):void
```

Operasi ini digunakan untuk mengambil data Spp yang dipergunakan pada saat akan melakukan ubah data Spp.

```
+updateSpp($where,$data):void
```

Operasi ini digunakan untuk mengubah data Spp.

```
+deleteSpp($id):void
```

Operasi ini digunakan untuk menghapus data Spp.

```
+getDataSppByIdSpp($id_spp):void
```

Operasi ini digunakan untuk mengambil data spp berdasarkan data yang dipilih.

```
+getDataSppDetailByIdSpp($id_spp):void
```

Operasi ini digunakan untuk mengambil data spp detail berdasarkan spp yang dipilih.

```
+getSppDetailFlexigridByIdSpp($id_spp): void
```

Operasi ini digunakan untuk menampilkan data Spp Detail berdasarkan dengan spp yang dipilih.

```
+insertSppDetail($data):void
```

Operasi ini digunakan untuk menambah data Spp detail baru.

```
+getSppDetailByIdSppDetail($id_spp_detail):void
```

Operasi ini digunakn untuk mengambil data Spp Detail yang dipergunakan pada saat akan melakukan ubah data spp detail.

```
+updateSppDetail($where,$data):void
```

Operasi ini digunakan untuk mengubah data spp detail.

```
+deleteSppDetail($id):void
```

Operasi ini digunakan untuk menghapus data spp detail.

2.2.3.75 Specific Design Class m_lpj

m_lpj	<<entity>>
<pre>+getLpjFlexigrid():Table</pre> Operasi ini digunakan untuk menampilkan data Lpj. <pre>+insertLpj(\$data):void</pre>	

Program Studi Teknik Informatika	DPPL-SIDeKa-PPD	86/ 111
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		

Operasi ini digunakan untuk menambah data Lpj baru.
`+getLpjByIdLpj($id_lpj):void`
 Operasi ini digunakan untuk mengambil data Lpj yang dipergunakan pada saat akan melakukan ubah data Lpj.
`+updateLpj($where,$data):void`
 Operasi ini digunakan untuk mengubah data Lpj.
`+deleteLpj($id):void`
 Operasi ini digunakan untuk menghapus data Lpj.
`+getDataLpjByIdLpj($id_lpj):void`
 Operasi ini digunakan untuk mengambil data lpj berdasarkan data yang dipilih.
`+getDataLpjDetailByIdLpj($id_lpj):void`
 Operasi ini digunakan untuk mengambil data lpj detail berdasarkan lpj yang dipilih.
`+getLpjDetailFlexigridByIdLpj($id_lpj): void`
 Operasi ini digunakan untuk menampilkan data Lpj Detail berdasarkan dengan lpj yang dipilih.
`+insertLpjDetail($data):void`
 Operasi ini digunakan untuk menambah data Lpj detail baru.
`+getLpjDetailByIdLpjDetail($id_lpj_detail):void`
 Operasi ini digunakn untuk mengambil data Lpj Detail yang dipergunakan pada saat akan melakukan ubah data lpj detail.
`+updateLpjDetail($where,$data):void`
 Operasi ini digunakan untuk mengubah data lpj detail.
`+deleteLpjDetail($id):void`
 Operasi ini digunakan untuk menghapus data lpj detail.

2.2.3.76 Specific Design Class **m_periode**

m_periode	<<entity>>
<code>+getPeriodeFlexigrid():Table</code> Operasi ini digunakan untuk menampilkan data periode. <code>+insertPeriode(\$data):void</code> Operasi ini digunakan untuk menambah data periode baru. <code>+getPeriodeByIdPeriode(\$id_periode):void</code>	

Operasi ini digunakan untuk mengambil data periode yang dipergunakan pada saat akan melakukan ubah data periode.

+updatePeriode(\$where,\$data):void

Operasi ini digunakan untuk mengubah data periode.

+deletePeriode():void

Operasi ini digunakan untuk menghapus data periode.

2.2.3.77 Specific Design Class m_tahun_anggaran

m_tahun	<<entity>>
<pre>+getTahunAnggaranFlexigrid():Table</pre> Operasi ini digunakan untuk menampilkan data tahun anggaran. <pre>+insertTahunAnggaran(\$data):void</pre> Operasi ini digunakan untuk menambah data tahun anggaran baru. <pre>+getTahunByIdTahunAnggaran(\$id_tahun_anggaran):void</pre> Operasi ini digunakan untuk mengambil data tahun anggaran yang dipergunakan pada saat akan melakukan ubah data tahun anggaran. <pre>+updateTahunAnggaran(\$where,\$data):void</pre> Operasi ini digunakan untuk mengubah data tahun anggaran. <pre>+deleteTahunAnggaran(\$id):void</pre> Operasi ini digunakan untuk menghapus data tahun anggaran.	

2.2.3.78 Specific Design Class m_bidang

m_bidang	<<entity>>
<pre>+getBidangFlexigrid():Table</pre> Operasi ini digunakan untuk menampilkan data bidang. <pre>+insertBidang(\$data):void</pre> Operasi ini digunakan untuk menambah data bidang baru. <pre>+getBidangByIdBidang(\$id_bidang):void</pre> Operasi ini digunakan untuk mengambil data bidang yang dipergunakan pada saat akan melakukan ubah data bidang. <pre>+updateBidang(\$where,\$data):void</pre> Operasi ini digunakan untuk mengubah data bidang.	

```
+deleteBidang($id):void
```

Operasi ini digunakan untuk menghapus data bidang.

2.2.3.79 Specific Design Class **m_kode_rekening**

m_kode_rekening	<<entity>>
<pre>+getKodeRekeningFlexigrid():Table</pre> Operasi ini digunakan untuk menampilkan data kode rekening. <pre>+insertKodeRekening(\$data):void</pre> Operasi ini digunakan untuk menambah data kode rekening baru. <pre>+getKodeByIdKodeRekening(\$id_kode_rekening):void</pre> Operasi ini digunakan untuk mengambil data kode rekening yang dipergunakan pada saat akan melakukan ubah data kode rekening. <pre>+updateKodeRekening(\$where,\$data):void</pre> Operasi ini digunakan untuk mengubah data kode rekening. <pre>+deleteKodeRekening(\$id):void</pre> Operasi ini digunakan untuk menghapus data kode rekening.	

2.2.3.80 Specific Design Class **m_sumber_dana**

m_sumber_dana	<<entity>>
<pre>+getSumberDanaFlexigrid():Table</pre> Operasi ini digunakan untuk menampilkan data sumber dana. <pre>+insertSumberDana(\$data):void</pre> Operasi ini digunakan untuk menambah data sumber dana baru. <pre>+getSumberByIdSumberDana(\$id_sumber_dana):void</pre> Operasi ini digunakan untuk mengambil data sumber dana yang dipergunakan pada saat akan melakukan ubah data sumber dana. <pre>+updateSumberDana(\$where,\$data):void</pre> Operasi ini digunakan untuk mengubah data sumber dana. <pre>+deleteSumberDana(\$id):void</pre> Operasi ini digunakan untuk menghapus data sumber dana.	

3 Perancangan Data

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas Data ref_rp_periode

Nama	Tipe	Panjang	Keterangan
Id_periode	integer	11	Id Periode, primary key
periode_awal	integer	5	Periode awal
periode_akhir	integer	5	Periode akhir
is_current	timestamp	-	Detail data dibuat

3.1.2 Deskripsi Entitas Data ref_rp_tahun_anggaran

Nama	Tipe	Panjang	Keterangan
id_tahun_anggaran	integer	11	Id Tahun Anggaran, primary key
deskripsi	varchar	25	Deskripsi dari tahun anggaran
regulasi	varchar	50	Regulasi tahun anggaran
keterangan	varchar	100	Keterangan tahun anggaran
tahun	int	4	Tahun Anggaran
id_periode	int	11	Id Periode, foreign key dari ref_rp_periode
is_current	timestamp	-	Detail data dibuat

3.1.3 Deskripsi Entitas Data ref_rp_bidang

Nama	Tipe	Panjang	Keterangan
id_bidang	integer	11	Id Bidang, primary key
kode_bidang	varchar	15	Kode Bidang dari setiap bidang
deskripsi	varchar	100	Deskripsi kode bidang
level	integer	2	Level dari bidang, dimana level 1 adalah bidang, 2 adalah urusan, 3 adalah program, 4 adalah kegiatan.

id_parent_bidang	integer	11	Id Parent Bidang, foreign key digunakan untuk rekursif data. Setiap bidang mempunyai sub bidang.
id_top_bidang	integer	11	Id Top Bidang, digunakan untuk menyimpan id top bidang setiap data yang rekursif.

3.1.4 Deskripsi Entitas Data ref_rp_coa

Nama	Tipe	Panjang	Keterangan
id_coa	integer	11	Id Coa, primary key
kode_rekening	varchar	20	Kode rekening dari coa (code of account)
deskripsi	varchar	100	Deskripsi kode rekening
level	integer	2	Level kode rekening
id_parent_coa	integer	11	Id Parent Bidang, foreign key digunakan untuk rekursif data. Setiap kode rekening mempunyai sub kode rekening.
id_top_coa	integer	11	Id Top Coa, digunakan untuk menyimpan id top kode rekening setiap data yang rekursif.

3.1.5 Deskripsi Entitas Data ref_rp_sumber_dana

Nama	Tipe	Panjang	Keterangan
id_sumber_dana	integer	11	Id Sumber Dana, primary key
sumber	varchar	50	Berisi sumber dana didapatkan
deskripsi	varchar	50	Deskripsi sumber dana
nominal	integer	20	Nominal dana dari sumber dana

id_tahun_anggaran	integer	11	Id Tahun Anggaran, foreign key dari ref_rp_tahun_anggaran.
-------------------	---------	----	--

3.1.6 Deskripsi Entitas Data tbl_rp_rpjmd

Nama	Tipe	Panjang	Keterangan
id_rpjmd	integer	11	Id Rpjmd, primary key
program	varchar	100	Program dari setiap RPJMD
kondisi_awal	varchar	50	Kondisi awal rpjmd
target	varchar	50	Target dari rpjmd
id_parent_rpjmd	integer	11	Id Parent Rpjmd, foreign key digunakan untuk rekursif data menyimpan id keluarga dari id rpjmd. Setiap rpjmd mempunyai sub rpjmd.
id_top_rpjmd	integer	11	Id Top Rpjmd, digunakan untuk menyimpan id top Rpjmd / id paling atas setiap data yang rekursif.
id_tahun_anggaran	integer	11	Id Tahun Anggaran, foreign key dari ref_rp_tahun_anggaran.

3.1.7 Deskripsi Entitas Data tbl_rp_rpjmDES

Nama	Tipe	Panjang	Keterangan
id_rpjmDES	integer	11	Id Rpjmdes, primary key
program	varchar	100	Program dari Rpjmdes
indicator	varchar	100	Indikator dari Rpjmdes
kondisi_awal	varchar	50	Kondisi awal Rpjmdes
target	varchar	50	Target dari Rpjmdes
capaian	varchar	100	Capaian dari Rpjmdes
id_parent_rpjmDES	integer	11	Id Parent Rpjmdes, foreign key digunakan

			untuk rekursif data menyimpan id keluarga dari id rpjmdes. Setiap rpjmdes mempunyai sub rpjmdes.
id_top_rpjmdes	integer	11	Id Top Rpjmdes, digunakan untuk menyimpan id top Rpjmdes / id paling atas setiap data yang rekursif.
id_rpjmd	integer	11	Id Rpjmd, foreign key dari tbl_rp_rpjmd
id_periode	integer	11	Id periode, foreign key dari ref_rp_periode
id_bidang	integer	11	Id Bidang, foreign key dari ref_rp_bidang

3.1.8 Deskripsi Entitas Data tbl_rp_rpjmdes_detail

Nama	Tipe	Panjang	Keterangan
id_rpjmdes_detail	integer	11	Id Rpjmdes Detail, primary key
volume	varchar	20	Volume dari rpjmdes detail
satuan	varchar	20	Satuan dari rpjmdes detail
nominal	double	-	Nominal dari rpjmdes detail
lokasi	varchar	30	Lokasi dari rpjmdes detail
tanggal	date	-	Tanggal rpjmdes detail dibuat
id_rpjmdes	integer	11	Id Rpjmdes, foreign key dari tbl_rp_rpjmdes
id_tahun_anggaran	integer	11	Id Tahun Anggaran, foreign key dari ref_tahun_anggaran

3.1.9 Deskripsi Entitas Data tbl_rp_rkpdes

Nama	Tipe	Panjang	Keterangan
id_rkpdes	integer	11	Id Rkpdes, primary key
program	varchar	100	Program dari rkpdes
indicator	varchar	100	Indikator dari rkpdes
kondisi_awal	varchar	100	Kondisi awal dari rkpdes
target	varchar	100	Target dari rkpdes
volume	varchar	10	Volume dari rkpdes
lokasi	varchar	50	Lokasi rkpdes
nominal	integer	25	Nominal dari rkpdes
id_parent_rkpdes	integer	11	Id Parent Rkpdes, foreign key digunakan untuk rekursif data menyimpan id keluarga dari id rkpdes. Setiap rkpdes mempunyai sub rkpdes.
id_top_rkpdes	integer	11	Id Top Rkpdes, digunakan untuk menyimpan id top Rkpdes / id paling atas setiap data yang rekursif.
id_rpjmdes	integer	11	Id Rpjmdes, foreign key dari tbl_rp_rpjmdes
id_tahun_anggaran	integer	11	Id Tahun Anggarn, foreign key dari ref_rp_tahun_anggaran
id_sumber_dana	integer	11	Id Sumber Dana, foreign key dari ref_rp_sumber_dama
id_bidang	integer	11	Id Bidang, foreign key dari ref_rp_bidang
id_coa	integer	11	Id Coa, foreign key dari ref_rp_coa

3.1.10 Deskripsi Entitas Data tbl_rp_rabdes

Nama	Tipe	Panjang	Keterangan
id_rabdes	integer	11	Id Rabdes, primary key
kegiatan	varchar	100	Kegiatan dari Rabdes
waktu_pelaksanaan_awal	date	-	Waktu pelaksanaan awal dari Rabdes
waktu_pelaksanaan_akhir	date	-	Waktu pelaksanaan akhir dari Rabdes
total	integer	11	Total dari Rabdes Anggaran dari setiap Rabdes yang dipilih.
id_tahun_anggaran	integer	11	Id Tahun Anggaran, foreign key dari ref_rp_tahun_anggaran
id_rkpdes	integer	11	Id Rkpdes, foreign key dari tbl_rp_rkpdes
id_pengguna	integer	11	Id Pengguna, foreign key dari tbl_pengguna
nik	integer	25	Nik dari setiap pengguna
id_perangkat	integer	11	Id Perangkat, foreign key dari tbl_perangkat (kepala desa)
nip	integer	25	Nip dari perangkat desa (kepala desa)
tgl_entry	timestamp	-	Digunakan untuk menyimpan kapan data dibuat.

3.1.11 Deskripsi Entitas Data tbl_rp_rabdes_anggaran

Nama	Tipe	Panjang	Keterangan
Id_rabdes_anggaran	integer	11	Id Rabdes Anggaran, primary key

uraian	varchar	100	Uraian dari rabdes anggaran
volume	integer	30	Volume dari rabdes anggaran
harga_satuan	integer	20	Harga satuan dari rabdes anggaran
jumlah	integer	20	Jumlah dari rabdes anggaran (volume * harga satuan)
id_rabdes	integer	11	Id Rabdes, foreign key dari tbl_rp_rabdes
id_pengguna	integer	11	Id Pengguna, foreign key dari tbl_pengguna
tgl_entry	timestamp	-	Digunakan untuk menyimpan kapan data dibuat.

3.1.12 Deskripsi Entitas Data tbl_rp_spp

Nama	Tipe	Panjang	Keterangan
id_spp	integer	11	Id Spp, primary key
tgl_ambil	date	-	Tanggal ambil dari spp
total	integer	25	Total dari spp
id_rabdes	integer	11	Id Rabdes, foreign key ke tbl_rp_rabdes
tgl_entry	timestamp	-	Tanggal entry digunaakn untuk menyimpan kapan data dibuat.

3.1.13 Deskripsi Entitas Data tbl_rp_spp_detail

Nama	Tipe	Panjang	Keterangan
id_spp_detail	integer	11	Id Spp Detail, primary key
pagu_anggaran	integer	25	Pagu anggaran dari spp detail
pencairan_yg_lalu	integer	25	Pencairan yang lalu dari spp detail

permintaan_sekarang	integer	25	Permintaan sekarang dari spp detail
jumlah_saat_ini	integer	25	Jumlah saat ini dari spp detail
sisa_dana	integer	25	Sisa dana dari spp detail
id_spp	integer	11	Id Spp, foreign key ke tbl_rp_spp
id_rabdes_anggaran	integer	11	Id Rabdes Anggaran, foreign key ke tbl_rp_rabdes_anggaran

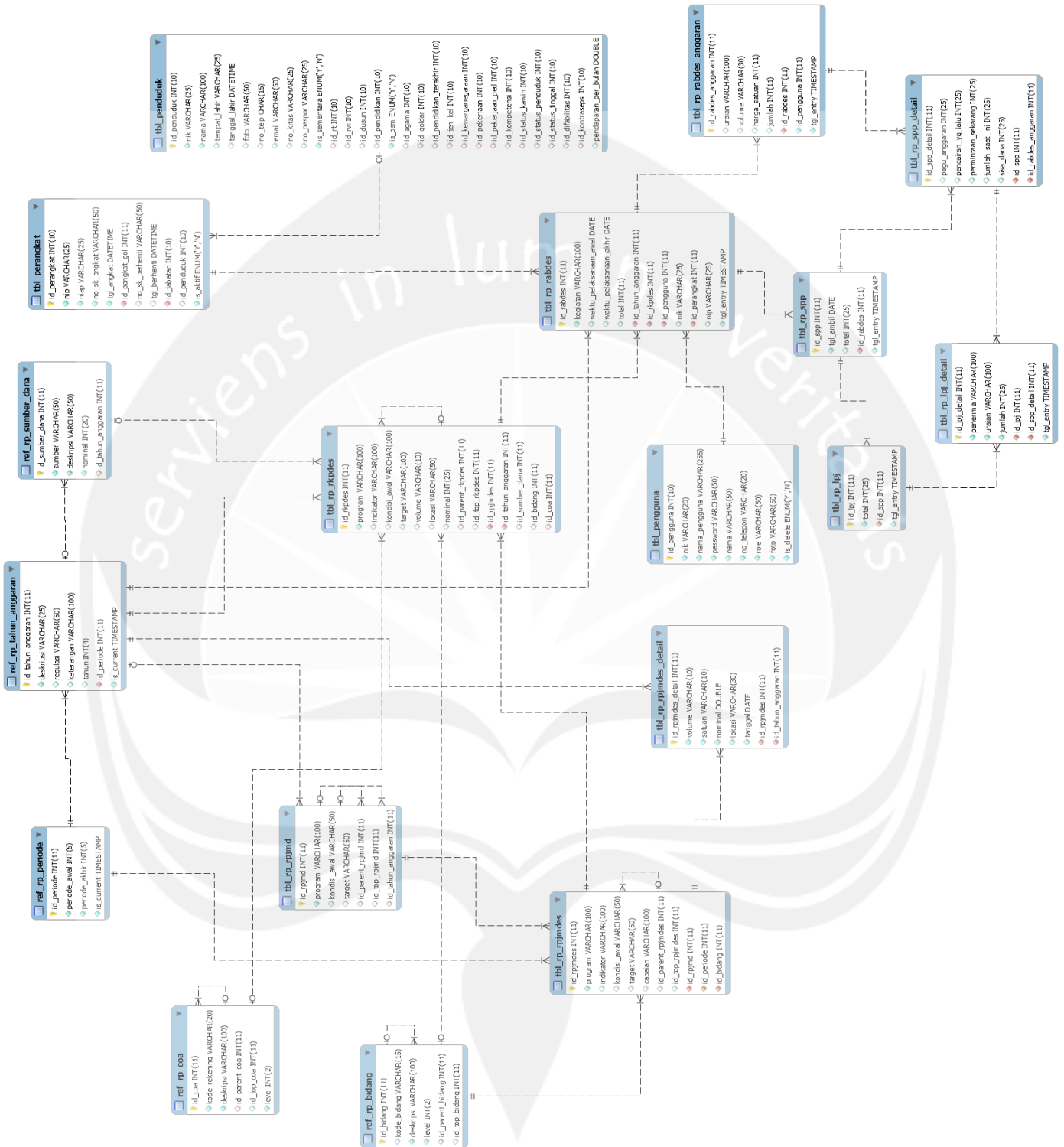
3.1.14 Deskripsi Entitas Data tbl_rp_lpj

Nama	Tipe	Panjang	Keterangan
id_lpj	integer	11	Id Lpj, primary key
total	integer	25	Total dari lpj
id_spp	integer	11	Id Spp, foreign key ke tbl_rp_spp
tgl_entry	timestamp	-	Tanggal entry digunaakn untuk menyimpan kapan data dibuat

3.1.15 Deskripsi Entitas Data tbl_rp_lpj_detail

Nama	Tipe	Panjang	Keterangan
id_lpj_detail	integer	11	Id Lpj Detail, primary key
penerima	varchar	100	Penerima dari lpj detail
uraian	varchar	100	Uraian lpj detail
jumlah	integer	25	Jumlah lpj detail
id_lpj	integer	11	Id Lpj, foreign key dari tbl_rp_lpj
id_spp_detail	integer	11	Id Spp Detail, foreign key dari tbl_rp_spp_detail
tgl_entry	timestamp	-	Tanggal entry digunaakn untuk menyimpan kapan data dibuat

3.2 Physical Data Model

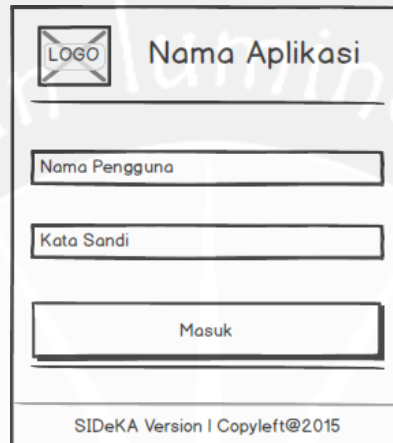


Gambar 3.1 Physical Data Model

4 Perancangan Antarmuka

4.1 Antarmuka Login

Antarmuka ini merupakan antarmuka yang pertama kali muncul ketika aplikasi dijalankan. Antarmuka ini digunakan pengguna untuk mendapatkan hak akses masuk ke dalam sistem. Pengguna mengisi form nama pengguna dan kata sandi. Pengguna mengisi form nama pengguna dan kata sandi.



LOGO Nama Aplikasi

Nama Pengguna

Kata Sandi

Masuk

SIDeKA Version I Copyleft@2015

Gambar 4.1 Antarmuka Login

4.2 Antarmuka Ubah Kata Sandi

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengubah kata sandi. Pengguna mengisi form kata sandi lama, kata sandi baru dan konfirmasi kata sandi baru. Jika data yang diinputkan benar maka sistem akan memperbarui data pengguna.



LOGO Nama Aplikasi

Kata Sandi Lama

Kata Sandi Baru

Konfirmasi Kata Sandi Baru

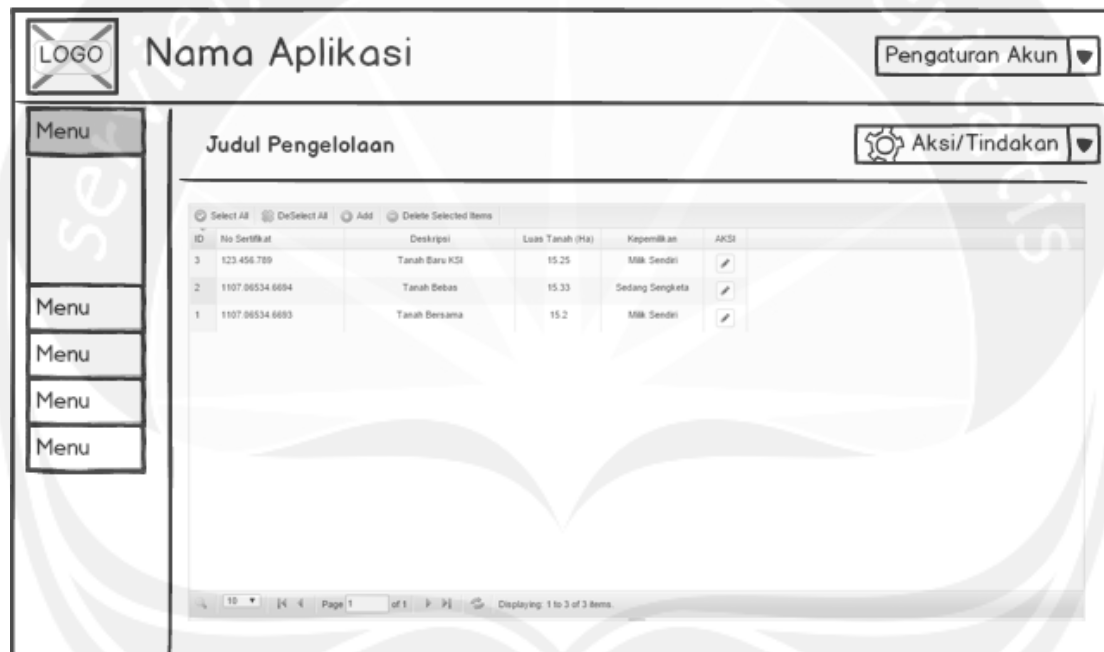
Simpan Kembali

SIDeKA Version I Copyleft@2015

Gambar 4.2 Antarmuka Ubah Kata Sandi

4.3 Antarmuka Tampil Data

Antarmuka ini merupakan antarmuka yang digunakan untuk menampilkan data dengan bentuk table. Antarmuka ini digunakan pada semua antarmuka yang menampilkan data. Pada antarmuka ini disediakan tombol dengan fungsi *select all* digunakan untuk melakukan seleksi semua data, *deselect all* digunakan untuk menghapus seleksi semua data, *add* digunakan untuk menambah data, *delete* digunakan untuk menghapus data, *search* digunakan untuk mencari data berdasarkan nama kolom dan *pagination* digunakan untuk mengakses halaman data.



Gambar 4.3 Antarmuka Tampil Data

4.4 Antarmuka Pengelolaan RPJMD

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data RPJMD dan sub program RPJMD. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data program, kondisi awal, target dan memilih tahun anggaran.

Gambar 4.4 Antarmuka Pengelolaan RPJMD

4.5 Antarmuka Pengelolaan RPJMDes

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data RPJMDes dan sub program RPJMDes. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data periode, bidang, program RPJMDes, program RPJMD, indikator, kondisi awal, target, capaian.

Gambar 4.5 Antarmuka Pengelolaan RPJMDes

4.6 Antarmuka Pengelolaan Detil RPJMDes

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data detil RPJMDes. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data periode, program RPJMDes, tahun anggaran, tanggal, volume, satuan, lokasi dan nominal.

The screenshot shows a web application interface for managing RPJMDes details. The header includes a logo, the application name 'Nama Aplikasi', and a 'Pengaturan Akun' button. A left sidebar contains a 'Menu' button. The main content area is titled 'Judul Pengelolaan' and features a form with the following fields: 'Periode*' (dropdown menu showing '2015 - 2019'), 'Program RPJMDes' (text input showing 'program rpjmdes'), 'Tahun*' (dropdown menu showing '2015'), 'Tanggal' (calendar icon), 'Volume*' (text input), 'Satuan*' (text input), 'Lokasi' (text input), and 'Nominal' (text input showing 'Rp 1000000 = Rp 1.000.000,-'). At the bottom of the form are 'Simpan' and 'Batal' buttons. A watermark 'Universitas Lumine' is visible in the background.

Gambar 4.6 Antarmuka Pengelolaan Detil RPJMDes

4.7 Antarmuka Pengelolaan RKPDes

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data RKPDes dan sub program RKPDes. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data program RPJMDes, tahun anggaran, bidang, program RKPDes, indicator, kondisi awal, target, lokasi, kode rekening, nominal, dan sumber dana.

Gambar 4.7 Antarmuka Pengelolaan RKPDes

4.8 Antarmuka Pengelolaan RABDes

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data RABDes. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data program RKPDes, tahun anggaran, kegiatan RABDes, waktu pelaksanaan awal, dan waktu pelaksanaan akhir.

Gambar 4.8 Antarmuka Pengelolaan RABDes

4.9 Antarmuka Pengelolaan RABDes Anggaran

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data RABDes Anggaran. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data uraian kegiatan, volume, harga satuan, dan jumlah.

Gambar 4.9 Antarmuka Pengelolaan RABDes Anggaran

4.10 Antarmuka Pengelolaan SPP

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data SPP. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data kegiatan RABDes, kegiatan RKPDes, tahun anggaran, dan tanggal ambil.

Gambar 4.10 Antarmuka Pengelolaan SPP

4.11 Antarmuka Pengelolaan Detil SPP

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data SPP Detail. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data permintaan sekarang.

AKSI	URAIAN	PAGU ANGGARAN	PERMINTAAN SEKARANG	JUMLAH SAAT INI	SISA DANA
1	2	3	4	5	6
☐	Batu	600000	Pemintaan Sekarang	Jumlah Saat Ini	297000
☐	Besi	28500000	Pemintaan Sekarang	Jumlah Saat Ini	Sisa Dana

Simpan Batal

Gambar 4.11 Antarmuka Pengelolaan SPP Detail

4.12 Antarmuka Pengelolaan LPJ

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data LPJ. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data kegiatan RABDes, kegiatan RKPDes, tahun anggaran, dan tanggal ambil.

Gambar 4.12 Antarmuka Pengelolaan LPJ

4.13 Antarmuka Pengelolaan Detil LPJ

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data LPJ Detail. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data permintaan sekarang.

AKSI	URAIAN	PAGU ANGGARAN	PERMINTAAN SEKARANG	JUMLAH SAAT INI	SISA DANA
1	2	3	4	5	6
☐	Bata	600000	Pemintaan Sekarang	Jumlah Saat Ini	297000
☐	Besi	28500000	Pemintaan Sekarang	Jumlah Saat Ini	Sisa Dana

Gambar 4.13 Antarmuka Pengelolaan LPJ Detail

4.14 Antarmuka Pengelolaan Periode

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data Periode. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data periode awal dan periode akhir.

The screenshot shows a web application interface for managing periods. It features a header with a logo, the application name, and a user settings dropdown. A sidebar on the left contains a menu button and four additional menu labels. The main workspace is titled 'Judul Pengelolaan' and contains two dropdown menus for 'Periode Awal' and 'Periode Akhir', both currently set to '--Pilih--'. Below these are 'Simpan' and 'Batal' buttons. A 'Aksi/Tindakan' dropdown is located in the top right corner of the main workspace.

Gambar 4.14 Antarmuka Pengelolaan Periode

4.15 Antarmuka Pengelolaan Tahun Anggaran

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data Tahun Anggaran. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data periode, tahun anggaran, deskripsi, regulasi, keterangan.

The screenshot displays a web application interface for managing budget years. At the top, there is a header bar containing a 'LOGO' placeholder, the text 'Nama Aplikasi', and a 'Pengaturan Akun' (Account Settings) button with a dropdown arrow. Below the header, the interface is divided into two main sections. On the left is a vertical 'Menu' sidebar with a 'Menu' button at the top and four additional 'Menu' buttons below it. The main content area on the right is titled 'Judul Pengelolaan' (Management Title) and includes an 'Aksi/Tindakan' (Action/Action) button with a gear icon and a dropdown arrow. The form itself contains several input fields: 'Periode' with a dropdown menu showing '--Pilih--', 'Tahun Anggaran' (Budget Year) with a dropdown menu showing '2015', 'Deskripsi' (Description), 'Regulasi' (Regulation), and 'Keterangan' (Remarks). At the bottom of the form are two buttons: 'Simpan' (Save) and 'Batal' (Cancel).

Gambar 4.15 Antarmuka Tahun Anggaran

4.16 Antarmuka Pengelolaan Bidang

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data Bidang. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data kode bidang, dan bidang.

The screenshot displays a web application interface for managing field data. The layout is similar to the previous interface, with a header bar containing a 'LOGO' placeholder, the text 'Nama Aplikasi', and a 'Pengaturan Akun' (Account Settings) button. The left sidebar contains a 'Menu' button at the top and four additional 'Menu' buttons below it. The main content area is titled 'Judul Pengelolaan' (Management Title) and includes an 'Aksi/Tindakan' (Action/Action) button with a gear icon and a dropdown arrow. The form contains two input fields: 'Kode Bidang*' (Field Code*) and 'Bidang*' (Field*). At the bottom of the form are two buttons: 'Simpan' (Save) and 'Batal' (Cancel).

Gambar 4.16 Antarmuka Pengelolaan Bidang

4.17 Antarmuka Pengelolaan Kode Rekening

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data Kode Rekening. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data kode rekening dan deskripsi.

Gambar 4.17 Antarmuka Pengelolaan Kode Rekening

4.18 Antarmuka Pengelolaan Sumber Dana

Antarmuka ini merupakan antarmuka yang digunakan pengguna untuk mengelola tambah dan ubah data Sumber Dana. Pada antarmuka ini pengguna diminta untuk mengisi form dengan data sumber dana, deskripsi, nominal dan memilih tahun anggaran.

LOGO

Nama Aplikasi

Pengaturan Akun ▼

Menu

Menu

Menu

Menu

Menu

Judul Pengelolaan

Aksi/Tindakan ▼

Sumber Dana*

Sumber Dana

Deskripsi*

Deskripsi Sumber Dana

Nominal

Rp 1000000 = Rp 1.000.000,-

Tahun Anggaran

--Pilih-- ▼

Simpan

Batal

Gambar 4.18 Antarmuka Pengelolaan Sumber Dana