

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini merupakan bab penutup, akan diberikan kesimpulan dan saran yang didapatkan selama pembuatan Tugas Akhir untuk pengembangan sistem lebih lanjut.

6.1. Kesimpulan

Berdasarkan analisis, desain, implementasi perangkat lunak dan pembahasan sebelumnya, maka dapat ditarik kesimpulan:

Pembangunan sistem informasi pengelolaan pengujian kendaraan bermotor Kabupaten Purbalingga dengan nama SIPPKB telah berhasil dibangun terbukti dari hasil pengujian terhadap pengguna.

6.2. Saran

Beberapa saran dan masukan yang didapatkan penulis terhadap perancangan sistem informasi pengelolaan pengujian kendaraan bermotor Kabupaten Purbalingga meliputi:

Pengembangan dapat menambahkan fitur pelaporan dan notifikasi.

DAFTAR PUSTAKA

- Abdullah, D. (2015). PERANCANGAN SISTEM INFORMASI PENGELOLAAN RETRIBUSI PENGUJIAN KENDARAAN BERMOTOR KABUPATEN ACEH UTARA. *Jurnal Teknik Informatika*.
- Ableson. (2009). *Sistem Operasi*. Retrieved 11 29, 2016, from <http://library.binus.ac.id/eColls/eThesisdoc/Bab2HTML/2013100969SIBab2001/page24.html>
- Alkautsar, A. R., Raudah, & Welda. (2008). SISTEM INFORMASI MANAJEMEN PROYEK PADA PT. ANUGRAH PERTIWI KONTRINDO PALEMBANG. *Jurnal STMIK-MDP*, 2.
- Balachandar, R. A., Aye, W., Avandi, M., Anitha, R., & Kumar, S. A. (2007). In *Web Services Security and E-Business* (p. 166).
- Daqiqil, I. (2011). *Framework CodeIgniter*. Pekanbaru.
- Jogiyanto, H. M. (2004). Analisis dan Desain.
- Kristanto, A. (2003). Perancangan Sistem Informasi dan Aplikasinya.
- Loudon, K. C., & Jane, P. (2007). Sistem Informasi Manajemen.
- Mohammad Rosul, Y. I. (2014). SISTEM INFORMASI PENGUJIAN KENDARAAN BERMOTOR DINAS PERHUBUNGAN KOMUNIKASI DAN INFORMATIKA KABUPATEN KUDUS. *Jurnal Sistem Informasi*.

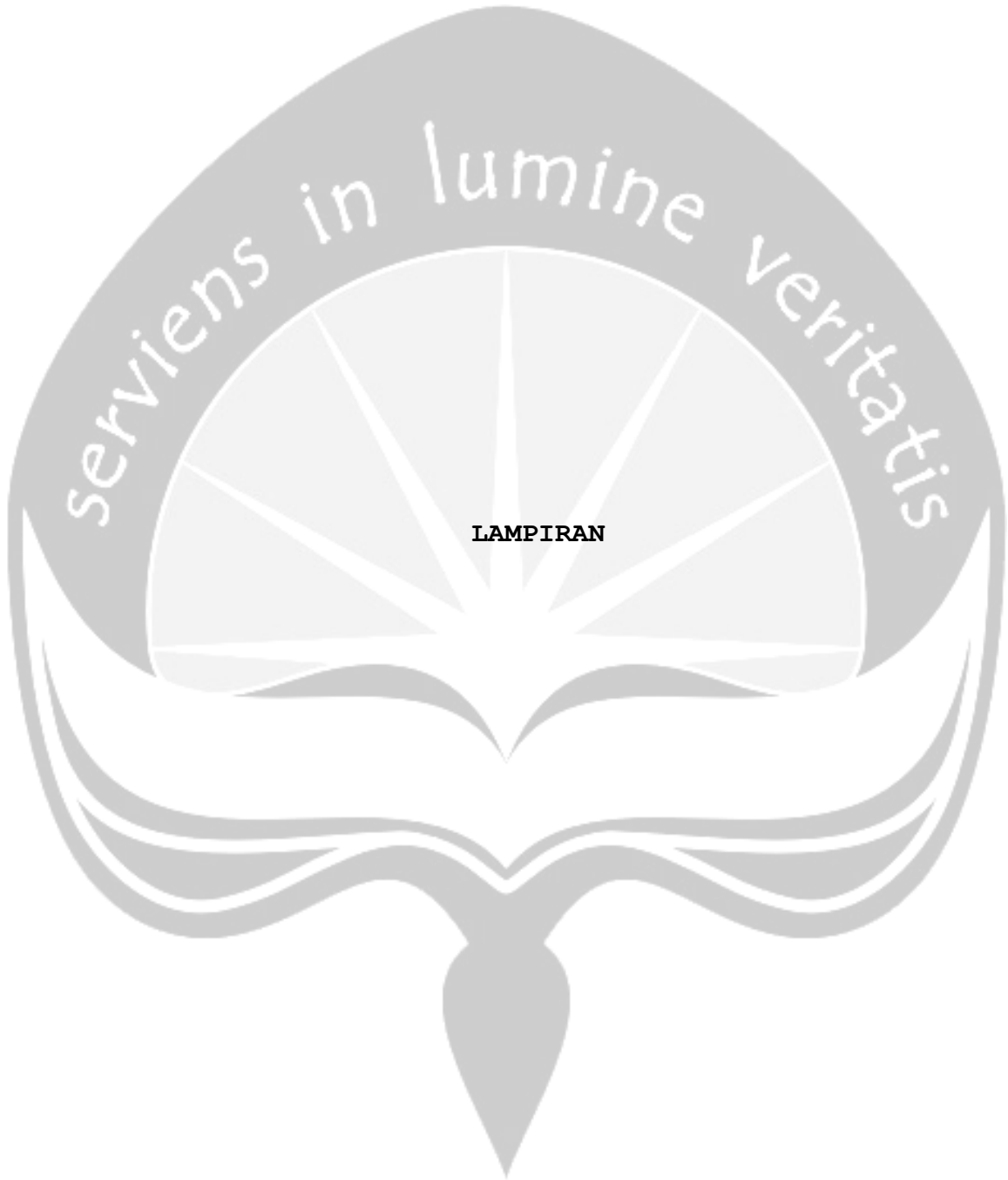
Samsinar, L. S. (2014). DESAIN SISTEM INFORMASI PENGUJIAN KENDARAAN BERMOTOR PADA UPT. PKB KENDARAAN KHUSUS CILINCING. *Jurnal Sistem Informasi*.

Sutabri, T. (2004). Analisa Sistem Informasi.

Wahyono, T. (2005). *Komputer Pemrograman Web Dinamis dengan PHP5*. Jakarta: Elex Media Komputindo.

Wardana, L. A. (2016). Perancangan Antarmuka Aplikasi Mobile Konseling pada Gereja Katolik dengan Metode User Centered Design and Wireframe. *Thesis S2, UAJY*.

Welling, T. (2005). MySQL. <http://library.binus.ac.id/eColls/eThesisdoc/Bab2HTML/2013100969SIBab2001/page25.html>.



SKPL

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

**PEMBANGUNAN SISTEM INFORMASI
PENGELOLAAN PENGUJIAN KENDARAAN BERMOTOR
STUDI KASUS DINAS PERHUBUNGAN KABUPATEN
PURBALINGGA**

Untuk:

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Andreas Dimas Setyoko / 130707538

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<i>SKPL-SIPPKB</i>		1/44
		<i>Revisi</i>	A	

DAFTAR PERUBAHAN

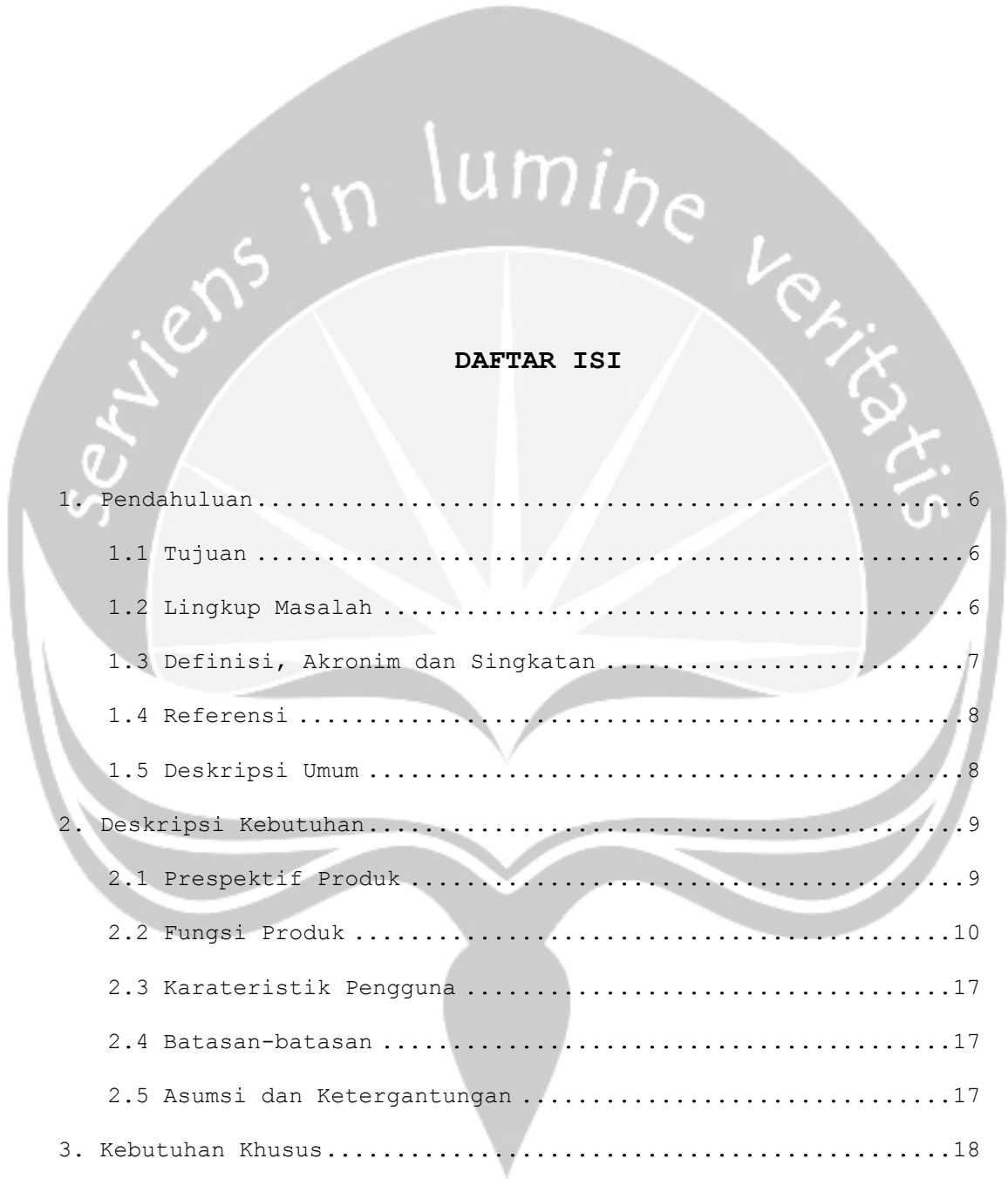
Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

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

Disetujui oleh								
-------------------	--	--	--	--	--	--	--	--

DAFTAR HALAMAN PERUBAHAN

Halaman	Revisi	Halaman	Revisi



DAFTAR ISI

1. Pendahuluan.....6

 1.1 Tujuan6

 1.2 Lingkup Masalah6

 1.3 Definisi, Akronim dan Singkatan7

 1.4 Referensi8

 1.5 Deskripsi Umum8

2. Deskripsi Kebutuhan.....9

 2.1 Prespektif Produk9

 2.2 Fungsi Produk10

 2.3 Karakteristik Pengguna17

 2.4 Batasan-batasan17

 2.5 Asumsi dan Ketergantungan17

3. Kebutuhan Khusus.....18

 3.1 Kebutuhan Antarmuka *Eksternal*18

 3.2 Kebutuhan Fungsionalitas Perangkat Lunak19

3.2.1 Use Case Diagram Aplikasi <i>Mobile</i>	19
3.2.2 Use Case Diagram Aplikasi <i>Web</i>	20
4. Spesifikasi Rinci Kebutuhan.....	21
4.1 Spesifikasi Kebutuhan Fungsionalitas	21
4.1.1. Pengelolaan Kategori Pengguna	21
4.1.2. Pengelolaan Tarif Uji	23
4.1.3. Pengelolaan Calon Pemilik	24
4.1.4. Pengelolaan Calon Kendaraan Uji	25
4.1.5. Pengelolaan Pemilik	27
4.1.6. Pengelolaan Kendaraan Uji	29
4.1.7. Pengelolaan Data Aduan	31
4.1.8. Pengelolaan Data Retribusi	32
4.1.9. Pengelolaan Detil Uji Laik Jalan	34
4.1.10. Pengelolaan Detil Uji Laik Jalan <i>Mobile</i>	37
4.1.11. Pengelolaan Pemilik <i>Mobile</i>	40
4.1.12. Pengelolaan Kendaraan Uji <i>Mobile</i>	41
4.1.13. Pengelolaan Data Aduan <i>Mobile</i>	41
4.1.14. Pengelolaan Data Tarif Uji <i>Mobile</i>	42
5. <i>Entity Relationship Diagram</i>	44

DAFTAR GAMBAR

Gambar 2.1 Rancangan Arsitektur Perangkat Lunak SIPPKB	10
Gambar 3.1 Use Case Diagram Aplikasi <i>Mobile</i> SIPPKB	19

Gambar 3.2 Use Case Diagram Aplikasi *Web* SIPPKB20

Gambar 5.1 Perancangan ERD SIPPKB44



1. Pendahuluan

1.1. Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SIPPKB (Sistem Informasi Pengelolaan Pengujian Kendaraan Bermotor) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka *eksternal* (antarmuka antara sistem dengan sistem lain perangkat lunak, perangkat keras, dan pengguna) dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-SIPPKB ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2. Lingkup Masalah

Perangkat Lunak SIPPKB dikembangkan dengan tujuan untuk:

1. Pengelolaan pengguna sistem, pengelolaan data kendaraan pada masing-masing kategori kendaraan uji dan pengelolaan fasilitas layanan tambahan (pembekuan kendaraan, perubahan bentuk kendaraan, peringatan keterlambatan, dan perubahan jenis uji)
2. Pengelolaan tarif uji, pengelolaan retribusi, pengelolaan calon pemilik dan pemilik, pengelolaan kendaraan bermotor, pengelolaan pengujian, pengelolaan teknis dan aduan masyarakat.
3. Proses transaksi layanan pengujian kendaraan bermotor mencakup jadwal tanggal pengujian dan pembayaran biaya uji kendaraan beserta penentuan denda keterlambatan uji kendaraan.

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-SIPPKB-XXX	Kode yang merepresentasikan kebutuhan pada SIPPKB (Sistem Informasi Pengelolaan Pengujian Kendaraan Bermotor) dimana XXX merupakan nomor fungsi produk.
SIPPKB	Perangkat lunak pengelolaan <i>web</i> dan <i>mobile</i> .
<i>Internet</i>	<i>Internet</i> merupakan istilah umum yang dipakai untuk menunjuk <i>Network global</i> yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk <i>e-mail</i> , <i>FTP</i> , dan <i>World Wide Web</i> .
<i>Server</i>	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Pengujian kendaraan bermotor	Pengujian kendaraan bermotor disebut juga uji kir adalah serangkaian kegiatan menguji dan/atau memeriksa bagian-bagian kendaraan bermotor,

	kereta gandengan, kereta tempelan dan kendaraan khusus dalam rangka pemenuhan terhadap persyaratan teknis dan laik jalan.
--	---

BPDP	Bagian Pengelolaan Data Pengujian
BTP	Bagian Teknis Pengujian
BLP	Bagian Layanan Pengujian

1.4. Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Andreas Dimas Setyoko / 130707538, Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ILUSI, Universitas Atma Jaya Yogyakarta.
2. Soal Projek Pembangunan Perangkat Lunak Profesional (P3LP) semester ganjil, Universitas Atma Jaya Yogyakarta.

1.5. Deskripsi umum (Overview)

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

Bagian kedua berisi penjelasan umum tentang perangkat lunak SIPPKB 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 SIPPKB tersebut. Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak SIPPKB yang akan dikembangkan.

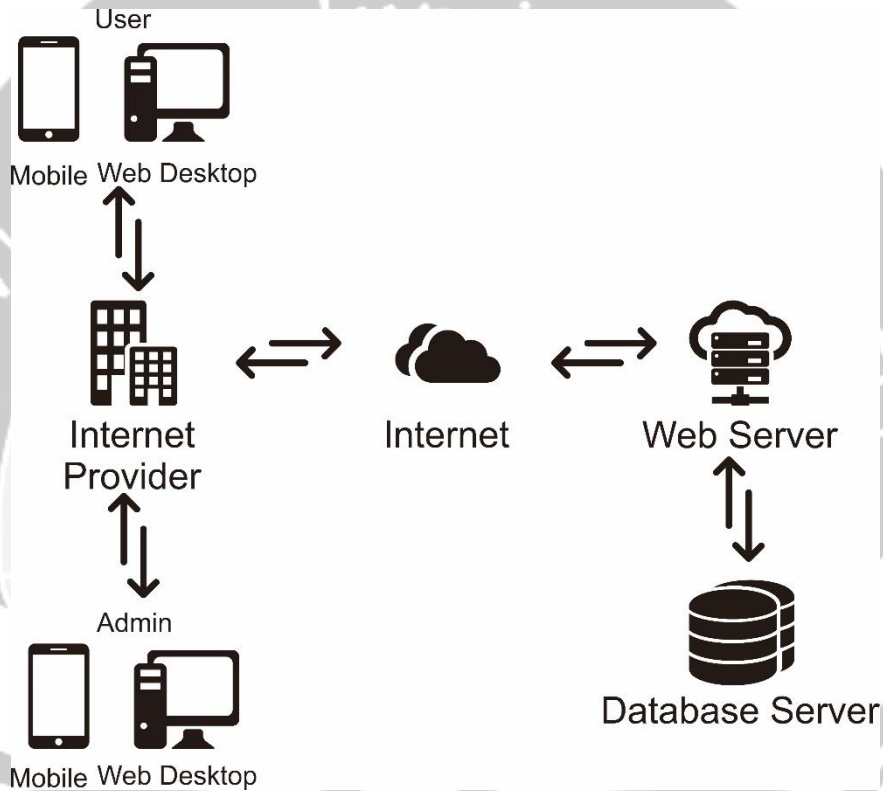
2. Deskripsi Kebutuhan

2.1. Perspektif produk

SIPPKB merupakan perangkat lunak yang dikembangkan untuk membantu pengelolaan sistem informasi pada *web* dan *mobile*. Sistem ini membantu Dinas Perhubungan Kabupaten Purbalingga untuk membangun sebuah sistem informasi yang dapat menangani semua proses pengujian kendaraan bermotor yang terjadi di dinas tersebut. Sistem ini dapat dipergunakan oleh Dinas Perhubungan Kabupaten Purbalingga sendiri maupun dipergunakan oleh masyarakat. Sistem ini sendiri dibagi menjadi dua jenis, aplikasi *web* dan aplikasi *mobile*.

Pada aplikasi *web* digunakan oleh pihak *internal* dinas seperti BPDP, BTP, dan BLP maupun pihak *eksternal* seperti pemilik kendaraan uji dan masyarakat. Aplikasi ini menangani antara lain: pengelolaan pengguna, pengelolaan data kendaraan berdasarkan kategori kendaraan uji, pengelolaan data dimensi kendaraan, pengelolaan transaksi pengujian, dan pengelolaan transaksi pembayaran. Untuk aplikasi *mobile*, sistem ini dapat digunakan masyarakat. Melalui aplikasi *mobile* tersebut masyarakat dapat melihat jadwal, biaya, dan denda pengujian kendaraan berdasarkan nomor kendaraan ataupun nomor uji kendaraan.

Perangkat lunak web SIPPKB ini berjalan pada *platform Windows* apapun, untuk web bisa berjalan pada web browser Google Chrome, untuk aplikasi *mobile* berjalan pada *platform android jelly bean*.



Gambar 2.1. Perancangan Arsitektur Perangkat Lunak SIM-PKB

2.2. Fungsi produk

Fungsi produk perangkat lunak SIPPKB adalah sebagai berikut:

APLIKASI DEKSTOP

1. Fungsi *Login* (SKPL-SIPPKB-001) .

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

2. Fungsi *Change Password* (SKPL-SIPPKB-002) .

Merupakan fungsi yang digunakan oleh semua *user internal* untuk mengubah *password user*.

3. Fungsi Pengelolaan Pengguna **(SKPL-SIPPKB-003)**.

Merupakan fungsi yang digunakan BPDP untuk mengelola data *user*.

Fungsi Pengelolaan Pengguna mencakup:

a. Fungsi *Entri* Pengguna **(SKPL-SIPPKB-003-01)**.

Merupakan fungsi yang digunakan untuk menambahkan data *user*.

b. Fungsi *Display* Pengguna **(SKPL-SIPPKB-003-02)**

Merupakan fungsi yang digunakan untuk menampilkan data *user*.

4. Fungsi Pengelolaan Tarif Uji **(SKPL-SIPPKB-004)**.

Merupakan fungsi yang digunakan BPDP untuk mengelola Tarif Uji. Fungsi Pengelolaan Tarif Uji mencakup:

a. Fungsi *Edit* Data Tarif Uji **(SKPL-SIPPKB-004-01)**.

Merupakan fungsi yang digunakan untuk mengubah data tarif uji

b. Fungsi Tampil Data Tarif Uji **(SKPL-SIPPKB-004-02)**.

Merupakan fungsi yang digunakan untuk menampilkan data tarif uji

5. Fungsi Pengelolaan Calon Pemilik **(SKPL-SIPPKB-005)**.

Merupakan fungsi yang digunakan BLP untuk mengelola data Calon Pemilik. Fungsi Pengelolaan Calon Pemilik mencakup:

a. Fungsi *Edit* Data Calon Pemilik **(SKPL-SIPPKB-005-01)**.

Merupakan fungsi yang digunakan untuk mengubah data calon pemilik

b. Fungsi Tampil Data Calon Pemilik **(SKPL-SIPPKB-005-02)**.

Merupakan fungsi yang digunakan untuk menampilkan data calon pemilik

6. Fungsi Pengelolaan Calon Pemilik Kendaraan Uji (**SKPL-SIPPKB-006**) .

Merupakan fungsi yang digunakan BLP untuk mengelola calon pemilik kendaraan uji. Fungsi Pengelolaan Calon Pemilik Kendaraan Uji mencakup:

a. Fungsi *Edit* Data Calon Pemilik Kendaraan Uji (**SKPL-SIPPKB-006-01**) .

Merupakan fungsi yang digunakan untuk mengubah data calon pemilik kendaraan uji

b. Fungsi Tampil Data Calon Pemilik Kendaraan Uji (**SKPL-SIPPKB-006-02**) .

Merupakan fungsi yang digunakan untuk menampilkan data calon pemilik kendaraan uji

7. Fungsi Pengelolaan Pemilik (**SKPL-SIPPKB-007**) .

Merupakan fungsi yang digunakan BLP untuk mengelola data pemilik. Fungsi Pengelolaan Pemilik mencakup:

a. Fungsi *Entri* Data Pemilik (**SKPL SIPPKB-007-01**) .

Merupakan fungsi yang digunakan untuk menambahkan data pemilik

b. Fungsi *Edit* Data Pemilik (**SKPL-SIPPKB-007-02**) .

Merupakan fungsi yang digunakan untuk mengubah data pemilik

c. Fungsi Tampil Data Pemilik (**SKPL-SIPPKB-007-03**) .

Merupakan fungsi yang digunakan untuk menampilkan data pemilik

8. Fungsi Pengelolaan Kendaraan Uji (**SKPL-SIPPKB-008**) .

Merupakan fungsi yang digunakan BLP untuk mengelola kendaraan uji. Fungsi Pengelolaan Kendaraan Uji mencakup:

a. Fungsi *Entri* Data Kendaraan Uji (**SKPL SIPPKB-008-01**).

Merupakan fungsi yang digunakan untuk menambahkan data kendaraan uji

b. Fungsi *Edit* Data Kendaraan Uji (**SKPL-SIPPKB-008-02**).

Merupakan fungsi yang digunakan untuk mengubah data kendaraan uji

c. Fungsi Tampil Data Kendaraan Uji (**SKPL-SIPPKB-008-03**).

Merupakan fungsi yang digunakan untuk menampilkan data kendaraan uji

9. Fungsi Pengelolaan Data Aduan (**SKPL-SIPPKB-009**).

Merupakan fungsi yang digunakan BLP untuk mengelola data aduan. Fungsi Pengelolaan Data Aduan mencakup:

a. Fungsi *Edit* Data Aduan (**SKPL-SIPPKB-009-01**).

Merupakan fungsi yang digunakan untuk mengubah data aduan

b. Fungsi Hapus Data Aduan (**SKPL-SIPPKB-009-02**).

Merupakan fungsi yang digunakan untuk menghapus data aduan

c. Fungsi Tampil Data Aduan (**SKPL-SIPPKB-009-03**).

Merupakan fungsi yang digunakan untuk menampilkan data aduan

10. Fungsi Pengelolaan Data Retribusi (**SKPL-SIPPKB-010**).

Merupakan fungsi yang digunakan BLP untuk mengelola data retribusi. Fungsi Pengelolaan Data Retribusi mencakup:

a. Fungsi *Entri* Data Retribusi (**SKPL SIPPKB-010-01**).

Merupakan fungsi yang digunakan untuk menambahkan data retribusi

b. Fungsi Tampil Data Retribusi (**SKPL-SIPPKB-010-02**).

Merupakan fungsi yang digunakan untuk menampilkan data retribusi

11. Fungsi Pengelolaan Pengujian (**SKPL-SIPPKB-011**).

Merupakan fungsi yang digunakan BLP untuk mengelola pengujian. Fungsi Pengelolaan Pengujian mencakup:

a. Fungsi *Entri* Data Pengujian (**SKPL SIPPKB-004-01**).

Merupakan fungsi yang digunakan untuk menambahkan data pengujian

b. Fungsi Tampil Data Pengujian (**SKPL-SIPPKB-004-02**).

Merupakan fungsi yang digunakan untuk menampilkan data pengujian

12. Fungsi Pengelolaan Detil Uji Laik Jalan (**SKPL-SIPPKB-012**).

Merupakan fungsi yang digunakan BLP untuk mengelola Detil Uji Laik Jalan. Fungsi Pengelolaan Detil Uji Laik Jalan mencakup:

a. Fungsi *Entri* Data Detil Uji Laik Jalan (**SKPL SIPPKB-012-01**).

Merupakan fungsi yang digunakan untuk menambahkan data Detil Uji Laik Jalan

b. Fungsi Tampil Data Detil Uji Laik Jalan (**SKPL-SIPPKB-012-04**).

Merupakan fungsi yang digunakan untuk menampilkan data Detil Uji Laik Jalan

APLIKASI MOBILE

1. Fungsi Pengelolaan Detil Uji Laik Jalan (SKPL-SIPPKB-013) .

Merupakan fungsi yang digunakan BTP untuk mengelola Detil Uji Laik Jalan. Fungsi Pengelolaan Detil Uji Laik Jalan mencakup:

a. Fungsi *Edit* Data Detil Uji Laik Jalan (SKPL-SIPPKB-013-01) .

Merupakan fungsi yang digunakan untuk mengubah data Detil Uji Laik Jalan

b. Fungsi Tampil Data Detil Uji Laik Jalan (SKPL-SIPPKB-013-02) .

Merupakan fungsi yang digunakan untuk menampilkan data Detil Uji Laik Jalan

2. Fungsi Pengelolaan Pemilik (SKPL-SIPPKB-015) .

Merupakan fungsi yang digunakan Masyarakat untuk melihat pengelolaan pemilik. Fungsi Pengelolaan Pemilik mencakup:

Fungsi Tampil Data Pemilik (SKPL SIPPKB-015-01) .

Merupakan fungsi yang digunakan untuk menambahkan data Pemilik

3. Fungsi Pengelolaan Kendaraan Uji (SKPL-SIPPKB-016) .

Merupakan fungsi yang digunakan Masyarakat melihat pengelolaan kendaraan uji. Fungsi Pengelolaan Kendaraan Uji mencakup:

Fungsi Tampil Data Kendaraan Uji (SKPL SIPPKB-016-01) .

Merupakan fungsi yang digunakan untuk menambahkan data kendaraan uji

4. Fungsi Pengelolaan Data Aduan (SKPL-SIPPKB-017) .

Merupakan fungsi yang digunakan Masyarakat melihat pengelolaan data aduan. Fungsi Pengelolaan Data Aduan mencakup:

Fungsi Tampil Data Aduan (SKPL SIPPKB-017-01) .

Merupakan fungsi yang digunakan untuk menambahkan data aduan

5. Fungsi Pengelolaan Data Retribusi **(SKPL-SIPPKB-018)** .

Merupakan fungsi yang digunakan Masyarakat melihat pengelolaan data retribusi. Fungsi Pengelolaan Data Retribusi mencakup:

a. Fungsi Tampil Data Retribusi **(SKPL SIPPKB-018-01)** .

Merupakan fungsi yang digunakan untuk menambahkan data retribusi

6. Fungsi Pengelolaan Data Tarif Uji **(SKPL-SIPPKB-019)** .

Merupakan fungsi yang digunakan Masyarakat melihat pengelolaan data tarif uji. Fungsi Pengelolaan Data Tarif Uji mencakup:

Fungsi Tampil Data Tarif Uji **(SKPL SIPPKB-019-01)** .

Merupakan fungsi yang digunakan untuk menambahkan data tarif uji

2.3. Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SIPPKB adalah sebagai berikut:

1. Mengerti tentang *internet* dan *web*
2. Mengerti tentang sistem operasi android
3. Memahami penggunaan aplikasi SIPPKB.

2.4. Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SIPPKB tersebut adalah:

1. Kebijakan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak SIPPKB.

2. Keterbatasan perangkat keras

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

2.5. Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada *web* yang menggunakan sistem operasi Windows apapun, dengan aplikasi *web browser* Google Chrome serta aplikasi *mobile* berbasis android.

3. Kebutuhan Khusus

3.1. Kebutuhan antarmuka eksternal

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

1. Antarmuka pemakai

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

2. Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SIPPKB adalah:

- Perangkat *web*.
- Perangkat *Database Server*.
- Perangkat *Web Server*.

3. Antarmuka perangkat lunak

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

1. Nama : Android Studio
Sumber : Google

Sebagai *framework* dalam pembuatan aplikasi *mobile*.

2. Nama : Android ver 4.0 keatas
Sumber : Android Open Source Project
(AOSP)

Sebagai sistem operasi untuk *smartphone*.

3. Nama : Google Chrome/Firefox
Sumber : berbagai *web server*

Sebagai *web server* untuk membuka sistem *web*.

4. Nama : Apache
Sumber : Apache Software Foundation.

Sebagai *web server*.

5. Nama : MySQL
Sumber : MySQL.

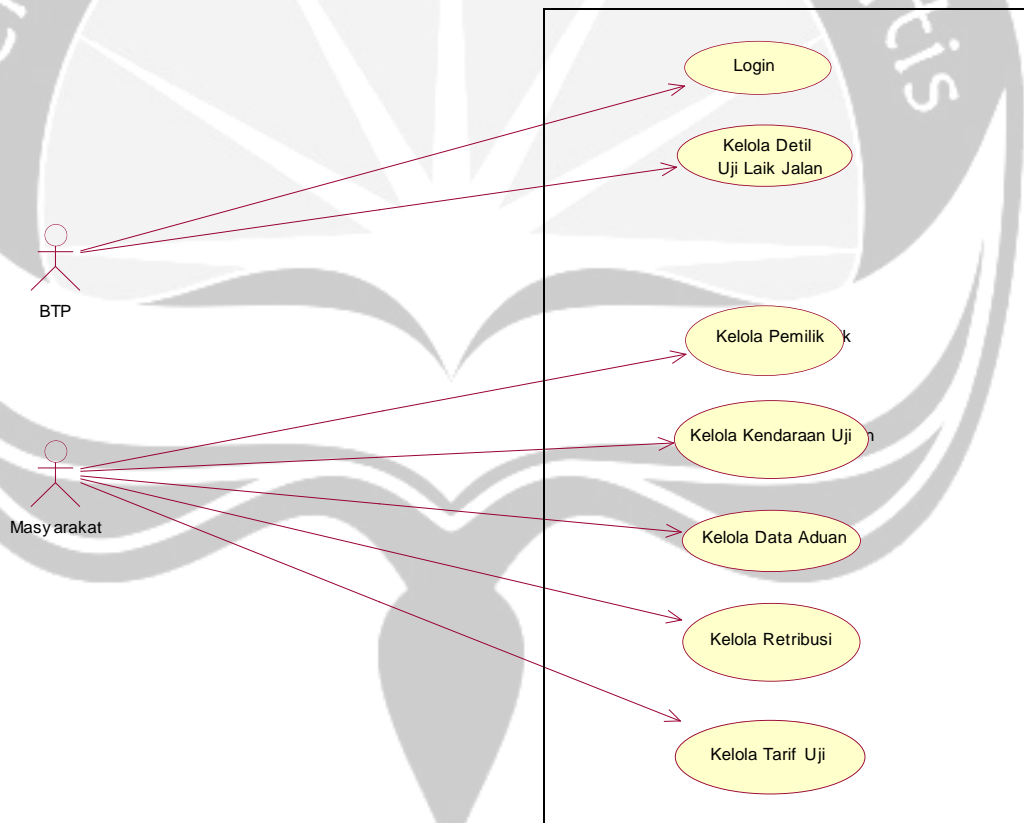
Sebagai *Database Management System* (DBMS) yang digunakan untuk penyimpanan data di sisi server.

4. Antarmuka komunikasi

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

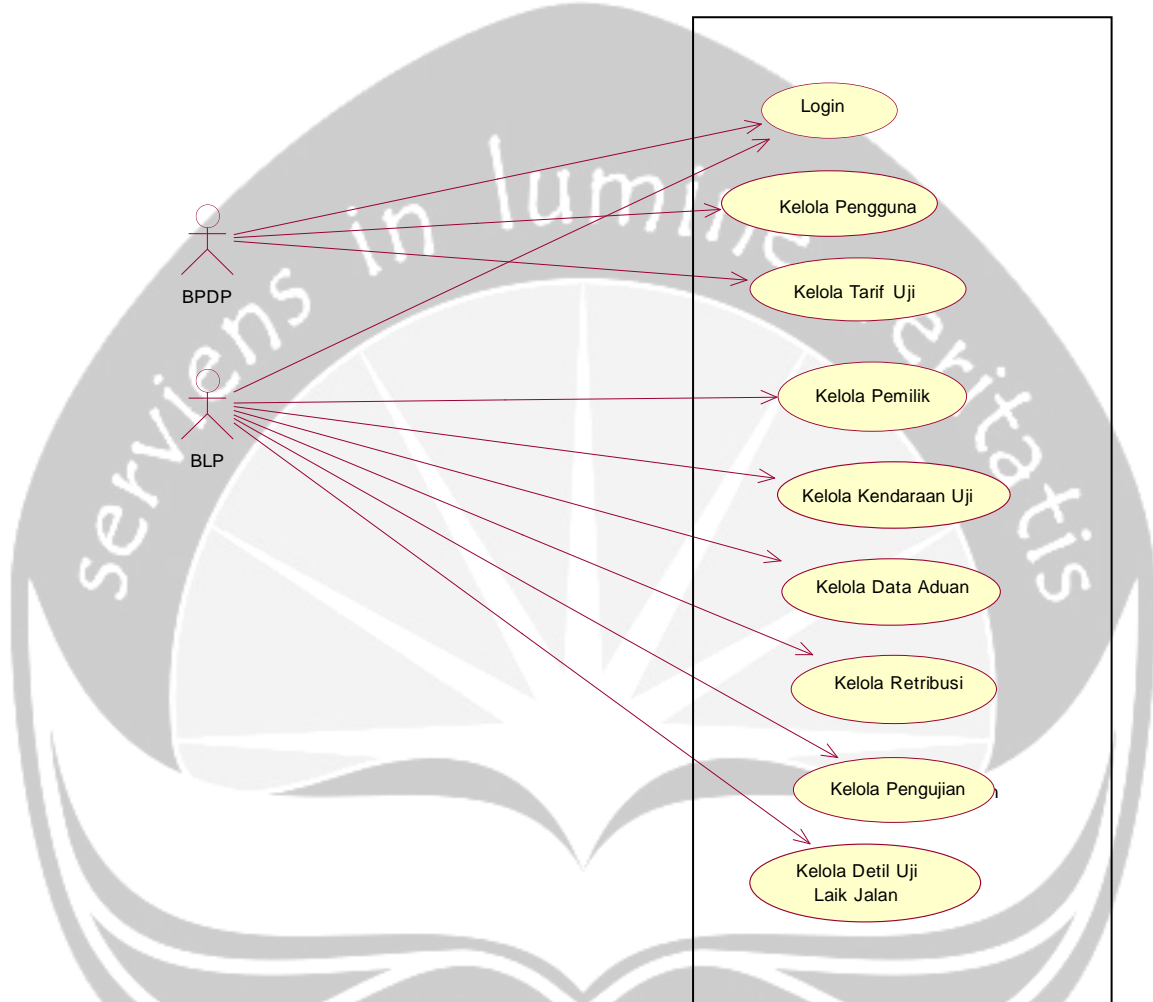
3.2. Kebutuhan Fungsionalitas Perangkat Lunak

3.2.1. Use Case Diagram Aplikasi Mobile



Gambar 3.1 Perancangan *Use Case Diagram* Aplikasi Mobile SIM-PKB

3.2.2. Use Case Diagram Aplikasi Web



Gambar 3.2 Perancangan Use Case Diagram Aplikasi Web SIM-PKB

4. Spesifikasi Rinci Kebutuhan

4.1. Spesifikasi Kebutuhan Fungsionalitas

APLIKASI *WEB*

4.1.1. Use case Spesification: Pengelolaan Kategori Pengguna

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola kategori pengguna. Aktor dapat melakukan *entry new* kategori pengguna, *edit* data kategori pengguna yang ada.

2. Primary Actor

BPDP

3. Supporting Actor

-

4. Basic Flow

a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan data kategori pengguna.

b) Sistem menampilkan antarmuka pengelolaan data kategori pengguna.

c) Sistem menampilkan data kategori pengguna.

d) Sistem memberikan pilihan untuk melakukan *entry* data baru kategori pengguna, *edit* data kategori pengguna.

e) Aktor memilih untuk melakukan *entry new* data kategori pengguna.

A-1 Aktor memilih untuk melakukan *edit* data kategori pengguna.

f) Aktor meminta sistem untuk menyimpan data kategori pengguna yang telah diinputkan

g) Sistem mengecek data kategori pengguna yang telah dimasukkan.

E-1 Data kategori pengguna yang diinputkan aktor salah.

- h) Sistem menyimpan data kategori pengguna ke *database*.
- i) *Use Case* selesai.

5. *Alternative Flow*

- a) A-1 Aktor memilih untuk melakukan *edit* data kategori pengguna.
- b) Aktor mengedit data kategori pengguna yang sudah ditampilkan.
- c) Aktor meminta sistem untuk menyimpan data kategori pengguna yang telah diubah.
- d) Sistem melakukan pengecekan terhadap data pengguna yang telah diubah.
E-2 Data kategori pengguna yang telah diedit salah.
- e) Berlanjut ke *Basic Flow* langkah ke-h.

6. *Error Flow*

- a) E-1 Data kategori pengguna yang diinputkan aktor salah
- b) Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
- c) Kembali ke *Basic Flow* Langkah ke-f.
E-2 Data kategori pengguna yang diinputkan aktor salah.
- d) Sistem memberikan pesan peringatan bahwa data yang diubah salah.
- e) Kembali ke *Alternative Flow* A-1 Langkah ke-2.

7. *PreConditions*

- a) *Use Case Login* telah dilakukan.
- b) Aktor telah memasuki sistem.

8. *PostConditions*

Data kategori pengguna di *database* telah *terupdate*.

4.1.2. Use case Spesification: Pengelolaan Tarif Uji

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola tarif uji. Aktor dapat melakukan *edit* data tarif uji dan menampilkan tarif uji yang ada.

2. *Primary Actor*

BPDP

3. *Supporting Actor*

-

4. *Basic Flow*

- a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan data tarif uji.
- b) Sistem menampilkan antarmuka pengelolaan data tarif uji.
- c) Sistem menampilkan data tarif uji.
- d) Sistem memberikan pilihan untuk melakukan *edit* data tarif uji.
- e) Aktor memilih untuk melakukan *edit* data tarif uji.
A-1 Aktor memilih untuk melakukan batal *edit* data tarif uji
- f) Aktor meminta sistem untuk menyimpan data tarif uji yang telah diinputkan
- g) Sistem mengecek data tarif uji yang telah diinputkan.
E-1 Data tarif uji yang diinputkan aktor salah.
- h) Sistem menyimpan data tarif uji ke *database*.
- i) *Use Case* selesai.

5. *Alternative Flow*

- a) A-1 Aktor memilih untuk melakukan batal *edit* data tarif uji.
Aktor batal mengubah data tarif uji yang sudah ditampilkan.
- b) Berlanjut ke *Basic Flow* langkah ke-h.

6. *Error Flow*

- a) E-1 Data tarif uji yang diinputkan aktor salah
Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
- b) Kembali ke *Basic Flow* Langkah ke-f.

7. *PreConditions*

- a) *Use Case Login* telah dilakukan.
- b) Aktor telah memasuki sistem.

8. *PostConditions*

Data tarif uji di *database* telah *terupdate*.

4.1.3. Use case Spesification: Pengelolaan Calon Pemilik

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola Calon Pemilik. Aktor dapat melakukan *edit* data Calon Pemilik dan menampilkan Calon Pemilik yang ada.

2. *Primary Actor*

BLP

3. *Supporting Actor*

-

4. *Basic Flow*

- a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Calon Pemilik.
- b) Sistem menampilkan antarmuka pengelolaan data Calon Pemilik.
- c) Sistem menampilkan data Calon Pemilik.
- d) Sistem memberikan pilihan untuk melakukan *edit* data Calon Pemilik.
- e) Aktor memilih untuk melakukan *edit* data Calon Pemilik.
 - A-1 Aktor memilih untuk melakukan batal *edit* data Calon Pemilik.

- f) Aktor meminta sistem untuk menyimpan data Calon Pemilik yang telah diinputkan
- g) Sistem mengecek data Calon Pemilik yang telah diinputkan.
E-1 Data Calon Pemilik yang diinputkan aktor salah.
- h) Sistem menyimpan data Calon Pemilik ke *database*.
- i) *Use Case* selesai.
4. *Alternative Flow*
- a) A-1 Aktor memilih untuk melakukan *batal edit* data Calon Pemilik
Aktor *batal mengedit* data Calon Pemilik yang sudah ditampilkan.
Aktor meminta sistem untuk menyimpan data Calon Pemilik yang telah diubah.
5. *Error Flow*
- a) E-1 Data Calon Pemilik yang diinputkan aktor salah
Sistem memberikan pesan peringatan bahwa data yang dimasukkan salah.
- b) Kembali ke *Basic Flow* Langkah ke-f.
6. *PreConditions*
- a) *Use Case* Login telah dilakukan.
- b) Aktor telah memasuki sistem.
7. *PostConditions*
- Data Calon Pemilik di *database* telah *terupdate*.

4.1.4. Use case Spesification: Pengelolaan Calon Kendaraan Uji

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola calon kendaraan uji. Aktor dapat melakukan *edit* data calon kendaraan uji dan menampilkan calon kendaraan uji yang ada.

2. *Primary Actor*

BLP

3. *Supporting Actor*

-

4. *Basic Flow*

a) Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data calon kendaraan uji.

b) Sistem menampilkan antarmuka pengelolaan data calon kendaraan uji.

c) Sistem menampilkan data calon kendaraan uji.

d) Sistem memberikan pilihan untuk melakukan *edit* data calon kendaraan uji.

e) Aktor memilih untuk melakukan *edit* data calon kendaraan uji.

A-1 Aktor memilih untuk melakukan batal *edit* data calon kendaraan uji.

f) Aktor meminta sistem untuk menyimpan data calon kendaraan uji yang telah diinputkan

g) Sistem mengecek data calon kendaraan uji yang telah diinputkan.

E-1 Data calon kendaraan uji yang diinputkan aktor salah.

h) Sistem menyimpan data calon kendaraan uji ke *database*.

i) *Use Case* selesai.

8. *Alternative Flow*

a) A-1 Aktor memilih untuk melakukan batal *edit* data calon kendaraan uji.

Aktor batal mengubah data calon kendaraan uji yang sudah ditampilkan.

Aktor meminta sistem untuk menyimpan data merk kendaraan uji yang telah batal diubah.

Berlanjut ke *Basic Flow* langkah ke-h.

9. *Error Flow*

- a) E-1 Data calon kendaraan uji yang diinputkan aktor salah
Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
 - b) Kembali ke *Basic Flow* Langkah ke-f.
10. *PreConditions*
- a) *Use Case Login* telah dilakukan.
 - b) Aktor telah memasuki sistem.
11. *PostConditions*
- Data calon kendaraan uji di *database* telah *terupdate*.

4.1.5. Use case Spesification: Pengelolaan Pemilik

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola Pemilik. Aktor dapat melakukan *entry new* data Pemilik, *edit* data Pemilik dan menampilkan data Pemilik yang ada.

2. *Primary Actor*

BLP

3. *Supporting Actor*

-

4. *Basic Flow*

- a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Pemilik.
- b) Sistem menampilkan antarmuka pengelolaan data Pemilik.
- c) Sistem menampilkan data Pemilik.
- d) Sistem memberikan pilihan untuk melakukan *entry* data baru Pemilik, *edit* data Pemilik.
- e) Aktor memilih untuk melakukan *entry new* data Pemilik.
A-1 Aktor memilih untuk melakukan *edit* data Pemilik.

- f) Aktor meminta sistem untuk menyimpan data Pemilik yang telah diinputkan
 - g) Sistem mengecek data Pemilik yang telah dimasukkan.
E-1 Data Pemilik yang diinputkan aktor salah.
 - h) Sistem menyimpan data Pemilik ke *database*.
 - i) *Use Case* selesai.
5. *Alternative Flow*
- a) A-1 Aktor memilih untuk melakukan *edit* data Pemilik.
 - b) Aktor mengedit data Pemilik yang sudah ditampilkan.
 - c) Aktor meminta sistem untuk menyimpan data Pemilik yang telah diubah.
 - d) Sistem melakukan pengecekan terhadap data Pemilik yang telah diubah.
E-2 Data Pemilik yang telah diubah salah.
 - e) Berlanjut ke *Basic Flow* langkah ke-h.
6. *Error Flow*
- E-1 Data Pemilik yang diinputkan aktor salah
- a) Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
 - b) Kembali ke *Basic Flow* Langkah ke-f.
- E-2 Data Pemilik yang diinputkan aktor salah.
- c) Sistem memberikan pesan peringatan bahwa data yang diubah salah.
 - d) Kembali ke *Alternative Flow* A-1 Langkah ke-2.
7. *PreConditions*
- a) *Use Case Login* telah dilakukan.
 - b) Aktor telah memasuki sistem.
8. *PostConditions*
- Data Pemilik di *database* telah *terupdate*.

4.1.6. Use case Spesification: Pengelolaan Kendaraan Uji

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola kendaraan uji. Aktor dapat melakukan *entry new* kendaraan uji, *edit* data kendaraan uji dan menampilkan kendaraan uji yang ada.

2. *Primary Actor*

BLP

3. *Supporting Actor*

-

4. *Basic Flow*

a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan data kendaraan uji.

b) Sistem menampilkan antarmuka pengelolaan data kendaraan uji.

c) Sistem menampilkan data kendaraan uji.

d) Sistem memberikan pilihan untuk melakukan *entry* data baru kendaraan uji, *edit* data kendaraan uji.

e) Aktor memilih untuk melakukan *entry new* data kendaraan uji.

A-1 Aktor memilih untuk melakukan *edit* data kendaraan uji.

f) Aktor meminta sistem untuk menyimpan data kendaraan uji yang telah diinputkan

g) Sistem mengecek data kendaraan uji yang telah dimasukkan.

E-1 Data kendaraan uji yang diinputkan aktor salah.

h) Sistem menyimpan data kendaraan uji ke *database*.

i) *Use Case* selesai.

5. *Alternative Flow*

- a) A-1 Aktor memilih untuk melakukan *edit* data kendaraan uji.
 - b) Aktor mengubah data kendaraan uji yang sudah ditampilkan.
 - c) Aktor meminta sistem untuk menyimpan data kendaraan uji yang telah diubah.
 - d) Sistem melakukan pengecekan terhadap data kendaraan uji yang telah diubah.
E-2 Data kendaraan uji yang telah diubah salah.
 - e) Berlanjut ke *Basic Flow* langkah ke-h.
 - f) A-2 Aktor memilih untuk melakukan hapus data kendaraan uji.
Sistem menampilkan konfirmasi untuk menghapus data
 - h) Berlanjut ke *Basic Flow* langkah ke-h.
4. *Error Flow*
- a) E-1 Data kendaraan uji yang diinputkan aktor salah
Sistem memberikan pesan peringatan bahwa data yang dimasukkan salah.
Kembali ke *Basic Flow* Langkah ke-f.
 - b) E-2 Data kendaraan uji yang diinputkan aktor salah.
Sistem memberikan pesan peringatan bahwa data yang diubah salah.
 - c) Kembali ke *Alternative Flow* A-1 Langkah ke-2.
5. *PreConditions*
- a) *Use Case Login* telah dilakukan.
 - b) Aktor telah memasuki sistem.
6. *PostConditions*
- Data kendaraan uji di *database* telah *terupdate*.

4.1.7. Use case Spesification: Pengelolaan Data Aduan

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola Data Aduan. Aktor dapat melakukan *edit* data Data Aduan, hapus data Data Aduan, dan menampilkan Data Aduan yang ada.

2. *Primary Actor*

BLP

3. *Supporting Actor*

-

4. *Basic Flow*

- a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan Data Aduan.
- b) Sistem menampilkan antarmuka pengelolaan Data Aduan.
- c) Sistem menampilkan Data Aduan.
- d) Sistem memberikan pilihan untuk melakukan *edit* Data Aduan, hapus Data Aduan.
- e) Aktor memilih untuk melakukan *edit* Data Aduan.
A-1 Aktor memilih untuk melakukan batal *edit* Data Aduan.
A-2 Aktor memilih untuk melakukan hapus Data Aduan.
- f) Aktor meminta sistem untuk menyimpan Data Aduan yang telah diubah
- g) Sistem mengecek Data Aduan yang telah diubah.
E-1 Data Aduan yang diubah aktor salah.
- h) Sistem menyimpan Data Aduan ke *database*.
- i) *Use Case* selesai.

7. *Alternative Flow*

- a) A-1 Aktor memilih untuk melakukan batal *edit* Data Aduan.
Aktor batal mengedit Data Aduan yang sudah ditampilkan.
Aktor meminta sistem untuk menyimpan Data Aduan yang telah batal diubah.

- Berlanjut ke *Basic Flow* langkah ke-h.
- b) A-2 Aktor memilih untuk melakukan hapus Data Aduan. Sistem menampilkan konfirmasi untuk menghapus data
 - c) Berlanjut ke *Basic Flow* langkah ke-h.
8. *Error Flow*
- a) E-1 Data Aduan yang diedit aktor salah Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
 - b) Kembali ke *Basic Flow* Langkah ke-f.
9. *PreConditions*
- a) *Use Case Login* telah dilakukan.
 - b) Aktor telah memasuki sistem.
10. *PostConditions*
- Data Aduan di *database* telah *terupdate*.

4.1.8. Use case Spesification: Pengelolaan Data Retribusi

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola data Retribusi. Aktor dapat melakukan tambah data baru Retribusi dan menampilkan data Retribusi yang ada.

2. *Primary Actor*

BLP

3. *Supporting Actor*

-

4. *Basic Flow*

- a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Retribusi.
- b) Sistem menampilkan antarmuka pengelolaan data Retribusi.
- c) Sistem menampilkan data Retribusi.
- d) Sistem memberikan pilihan untuk melakukan tambah data baru data Retribusi.

- e) Aktor memilih untuk melakukan tambah data Retribusi.
 - f) Aktor meminta sistem untuk menyimpan data kendaraan uji yang telah diinputkan
 - g) Sistem mengecek data Retribusi yang telah diinputkan.
E-1 Data Retribusi yang diinputkan aktor salah.
 - h) Sistem menyimpan data Retribusi ke *database*.
 - i) *Use Case* selesai.
11. *Alternative Flow*
-
12. *Error Flow*
- E-1 Data Retribusi yang diinputkan aktor salah
- a) Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
 - b) Kembali ke Basic Flow Langkah ke-e.
13. *PreConditions*
- a) *Use Case Login* telah dilakukan.
 - b) Aktor telah memasuki sistem.
14. *PostConditions*
- Data Retribusi di *database* telah *terupdate*.

4.1.9. Use case Spesification: Pengelolaan Detil Uji Laik Jalan

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola Detil Uji Laik Jalan. Aktor dapat melakukan edit data biaya uji dan menampilkan data Detil Uji Laik Jalan yang ada.

2. *Primary Actor*

BTP

3. *Supporting Actor*

-

4. *Basic Flow*

- a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Detil Uji Laik Jalan.

- b) Sistem menampilkan antarmuka pengelolaan data Detil Uji Laik Jalan.
- c) Sistem menampilkan data Detil Uji Laik Jalan.
- d) Sistem memberikan pilihan untuk melakukan *edit* data Detil Uji Laik Jalan.
- e) Aktor memilih untuk melakukan *edit* data Detil Uji Laik Jalan.
 - A-1 Aktor memilih untuk melakukan *batal edit* data Detil Uji Laik Jalan.
- f) Aktor meminta sistem untuk menyimpan data Detil Uji Laik Jalan yang telah diinputkan
- g) Sistem mengecek data Detil Uji Laik Jalan yang telah diinputkan.
 - E-1 Data Detil Uji Laik Jalan yang diubah aktor salah.
- h) Sistem menyimpan Detil Uji Laik Jalan ke *database*.
- i) *Use Case* selesai.

5. *Alternative Flow*

- a) A-1 Aktor memilih untuk melakukan *batal edit* data Detil Uji Laik Jalan.
 - Aktor *batal mengedit* data Detil Uji Laik Jalan yang sudah ditampilkan.
 - Aktor meminta sistem untuk menyimpan data Detil Uji Laik Jalan yang telah *batal* diubah.
- b) Berlanjut ke *Basic Flow* langkah ke-h.

6. *Error Flow*

- E-1 Data Detil Uji Laik Jalan yang diubah aktor salah
- a) Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
- b) Kembali ke *Basic Flow* Langkah ke-f.

7. *PreConditions*

- a) *Use Case Login* telah dilakukan.
- b) Aktor telah memasuki sistem.

8. *PostConditions*

Data Detil Uji Laik Jalan di database telah terupdate.

4.1.10. Use case Spesification: Pengelolaan Detil Uji Teknis

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola Detil Uji Teknis. Aktor dapat melakukan *edit* data Detil Uji Teknis dan menampilkan Detil Uji Teknis yang ada.

2. *Primary Actor*

BTP

3. *Supporting Actor*

-

4. *Basic Flow*

- a) *Use Case* ini dimulai ketika actor memilih untuk melakukan pengelolaan Detil Uji Teknis.
- b) Sistem menampilkan antarmuka pengelolaan Detil Uji Teknis.
- c) Sistem menampilkan Detil Uji Teknis.
- d) Sistem memberikan pilihan untuk melakukan edit Detil Uji Teknis dan tampil Detil Uji Teknis.
- e) Aktor memilih untuk melakukan tampil Detil Uji Teknis.

A-1 Aktor memilih untuk melakukan *edit* Detil Uji Teknis.

- g) Aktor meminta sistem untuk menyimpan data Detil Uji Teknis yang telah diubah.
- h) Sistem mengecek data Detil Uji Teknis yang telah diubah.

E-1 Data Detil Uji Teknis yang diedit aktor salah.

- i) Sistem menyimpan Detil Uji Teknis ke *database*.
- j) *Use Case* selesai.
- 5. *Alternative Flow*
 - a) A-1 Aktor memilih untuk melakukan batal edit Detil Uji Teknis.
Sistem menampilkan konfirmasi untuk batal edit data
 - b) Berlanjut ke *Basic Flow* langkah ke-h.
- 6. *Error Flow*
 - a) E-1 Data Detil Uji Teknis yang diedit aktor salah
Sistem memberikan pesan peringatan bahwa data yang dimasukkan salah.
 - b) Kembali ke *Basic Flow* Langkah ke-f.
- 7. *PreConditions*
 - a) *Use Case Login* telah dilakukan.
 - b) Aktor telah memasuki sistem.
- 8. *PostConditions*
Data Detil Uji Teknis di *database* telah terupdate.

4.1.11. Use case Spesification: Pengelolaan Detil Uji Laik Jalan

- 1. *Brief Description*
Use Case ini digunakan oleh actor untuk mengelola Detil Uji Laik Jalan. Aktor dapat melakukan edit Detil Uji Laik Jalan dan menampilkan Detil Uji Laik Jalan.
- 2. *Primary Actor*
BTP
- 3. *Supporting Actor*
-
- 4. *Basic Flow*
 - a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan Detil Uji Laik Jalan.

- b) Sistem menampilkan antarmuka pengelolaan Detil Uji Laik Jalan.
- c) Sistem menampilkan Detil Uji Laik Jalan.
- d) Sistem memberikan pilihan untuk melakukan *edit* Detil Uji Laik Jalan.
- e) Aktor memilih untuk melakukan *edit* data Detil Uji Laik Jalan.
- A-1 Aktor memilih untuk melakukan *batal edit* data Detil Uji Laik Jalan.
- f) Aktor meminta sistem untuk menyimpan data Detil Uji Laik Jalan yang telah *batal* diubah
- g) Sistem mengecek data Detil Uji Laik Jalan yang telah *batal* diubah.
- E-1 Data Detil Uji Laik Jalan yang diinputkan aktor salah.
- h) Sistem menyimpan data Detil Uji Laik Jalan ke *database*.
- i) Use Case selesai.

5. *Alternative Flow*

- a) A-1 Aktor memilih untuk melakukan *batal edit* data Detil Uji Laik Jalan.
Aktor melakukan *batal edit* data Detil Uji Laik Jalan yang sudah ditampilkan.
Aktor meminta sistem untuk menyimpan data Detil Uji Laik Jalan yang telah dibatalkan.
- b) Berlanjut ke *Basic Flow* langkah ke-h.

6. *Error Flow*

- a) E-1 Data Detil Uji Laik Jalan yang diedit aktor salah
Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
- b) Kembali ke *Basic Flow* Langkah ke-f.

7. *PreConditions*

- a) *Use Case Login* telah dilakukan.
 - b) Aktor telah memasuki sistem.
8. *PostConditions*
Data Detil Uji Laik Jalan di database telah terupdate.

4.1.12. Use case Spesification: Pengelolaan Detil Uji Teknis

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola Detil Uji Teknis. Aktor dapat melakukan *edit* Detil Uji Teknis dan menampilkan Detil Uji Teknis.

2. *Primary Actor*

BTP

3. *Supporting Actor*

-

3. *Basic Flow*

- a) *Use Case* ini dimulai ketika actor memilih untuk melakukan pengelolaan Detil Uji Teknis.
- b) Sistem menampilkan antarmuka pengelolaan Detil Uji Teknis.
- c) Sistem menampilkan Detil Uji Teknis.
- d) Sistem memberikan pilihan untuk melakukan *edit* Detil Uji Teknis.
- e) Aktor memilih untuk melakukan *edit* data Detil Uji Teknis.
 - A-1 Aktor memilih untuk melakukan batal *edit* data Detil Uji Teknis.
- f) Aktor meminta sistem untuk menyimpan data Detil Uji Teknis yang telah batal diubah
- g) Sistem mengecek data Detil Uji Teknis yang telah batal diubah.
 - E-1 Data Detil Uji Teknis yang diinputkan aktor salah.

- h) Sistem menyimpan data Detil Uji Teknis ke *database*.
 - i) *Use Case* selesai.
4. *Alternative Flow*
- a) A-1 Aktor memilih untuk melakukan batal *edit* data Detil Uji Teknis.
Aktor melakukan batal *edit* data Detil Uji Teknis yang sudah ditampilkan.
Aktor meminta sistem untuk menyimpan data Detil Uji Teknis yang telah dibatalkan.
 - b) Berlanjut ke *Basic Flow* langkah ke-h.
5. *Error Flow*
- a) E-1 Data Detil Uji Teknis yang diedit aktor salah Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
 - b) Kembali ke *Basic Flow* Langkah ke-f.
6. *PreConditions*
- a) *Use Case Login* telah dilakukan.
 - b) Aktor telah memasuki sistem.
7. *PostConditions*
- Data Detil Uji Teknis di *database* telah *terupdate*.

4.1.13. Use case Spesification: Pengelolaan Pemilik

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola Pemilik. Aktor dapat melakukan menampilkan data Pemilik yang ada.

2. *Primary Actor*

MASYARAKAT

3. *Supporting Actor*

-

4. *Basic Flow*

- a) Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data Pemilik.
 - b) Sistem menampilkan antarmuka pengelolaan data Pemilik.
 - c) Sistem menampilkan data Pemilik.
 - d) Use Case selesai.
5. *Alternative Flow*
 -
 6. *Error Flow*
 -
 7. *PreConditions*
 - e) Use Case Login telah dilakukan.
 - f) Aktor telah memasuki sistem.
 8. *PostConditions*

Data Pemilik berhasil ditampilkan.

4.1.14. Use case Specification: Pengelolaan Kendaraan Uji

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola kendaraan uji. Aktor dapat melakukan menampilkan kendaraan uji yang ada.
2. *Primary Actor*

MASYARAKAT
3. *Supporting Actor*
 -
4. *Basic Flow*
 - g) Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data kendaraan uji.
 - h) Sistem menampilkan antarmuka pengelolaan data kendaraan uji.
 - i) Sistem menampilkan data kendaraan uji.
 - j) Use Case selesai.
5. *Alternative Flow*

-
- 6. *Error Flow*
 -
- 7. *PreConditions*
 - a) *Use Case Login* telah dilakukan.
 - b) Aktor telah memasuki sistem.
- 8. *PostConditions*

Data Kendaraan Uji berhasil ditampilkan.

4.1.15. Use case Spesification: Pengelolaan Data Aduan

- 1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola Data Aduan. Aktor dapat melakukan menampilkan Data Aduan yang ada.
- 2. *Primary Actor*

MASYARAKAT
- 3. *Supporting Actor*
 -
- 4. *Basic Flow*
 - a) *Use Case* ini dimulai ketika actor memilih untuk melakukan pengelolaan Data Aduan.
 - b) Sistem menampilkan antarmuka pengelolaan Data Aduan.
 - c) Sistem menampilkan Data Aduan.
 - d) *Use Case* selesai.
- 5. *Alternative Flow*
 -
- 6. *Error Flow*
 -
- 7. *PreConditions*
 - e) *Use Case Login* telah dilakukan.
 - f) Aktor telah memasuki sistem.
- 8. *PostConditions*

Data Retribusi berhasil ditampilkan.

4.1.16. Use case Spesification: Pengelolaan Data Tarif Uji

1. *Brief Description*

Use Case ini digunakan oleh actor untuk mengelola data Retribusi. Aktor dapat menampilkan data Tarif Uji yang ada.

2. *Primary Actor*

MASYARAKAT

3. *Supporting Actor*

-

4. *Basic Flow*

g) *Use Case* ini dimulai ketika actor memilih untuk melakukan pengelolaan data Tarif Uji.

h) Sistem menampilkan antarmuka pengelolaan data Tarif Uji.

i) Sistem menampilkan data Tarif Uji.

j) *Use Case* selesai.

5. *Alternative Flow*

-

6. *Error Flow*

-

7. *PreConditions*

a) *Use Case Login* telah dilakukan.

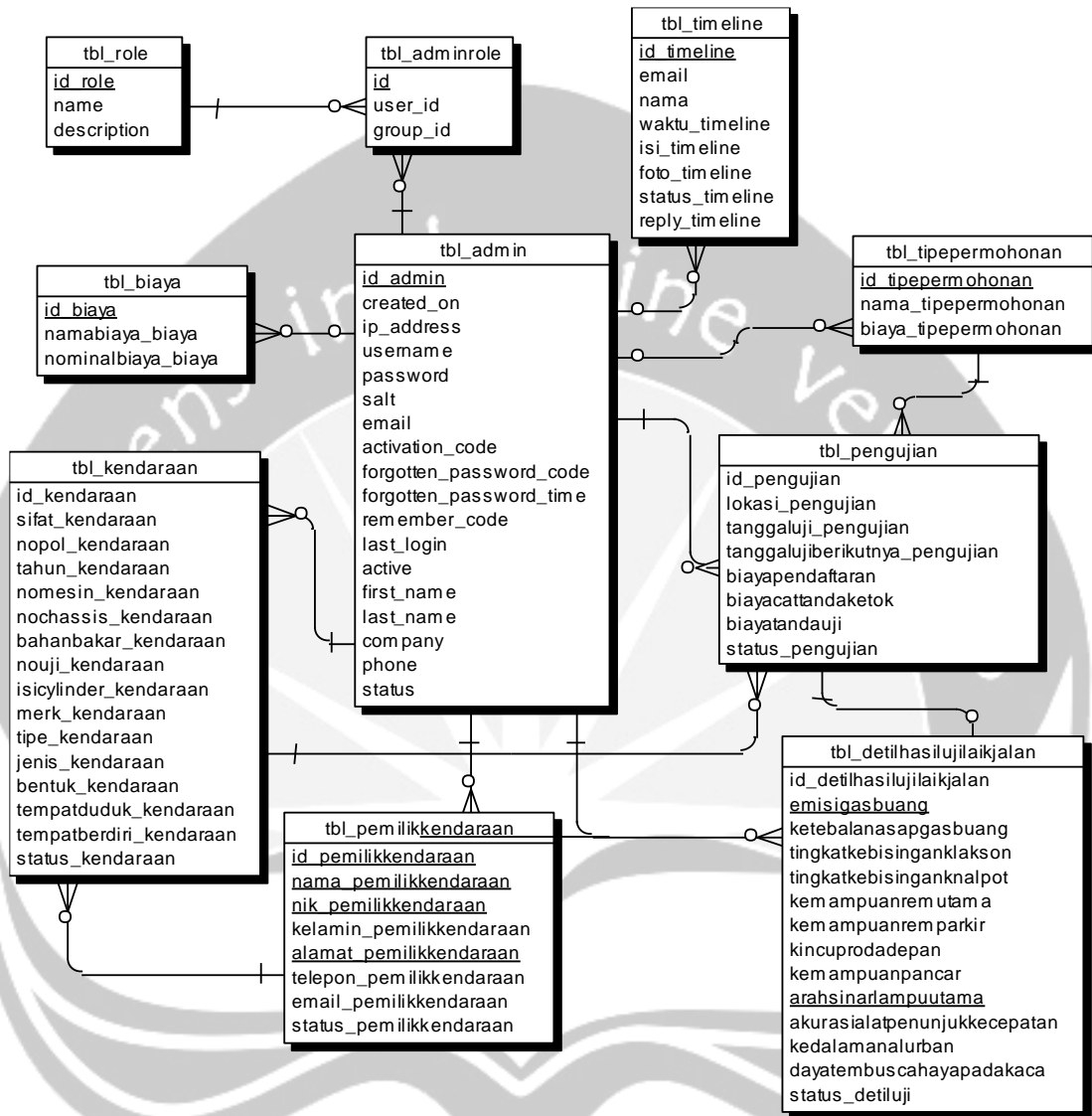
b) Aktor telah memasuki sistem.

8. *PostConditions*

Data Tarif Uji telah ditampilkan.



5. *Entity Relationship Diagram*



Gambar 5.1 Perancangan *Entity Relationship Diagram* SIPPKB

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

**PEMBANGUNAN SISTEM INFORMASI
PENGELOLAAN PENGUJIAN KENDARAAN BERMOTOR
STUDI KASUS DINAS PERHUBUNGAN KABUPATEN
PURBALINGGA**

Untuk:

Universitas Atma Jaya Yogyakarta


Dipersiapkan oleh:

Andreas Dimas Setyoko / 130707538

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<i>DPPL-SIPPKB</i>		<i>1/73</i>
		<i>Revisi</i>	<i>A</i>	

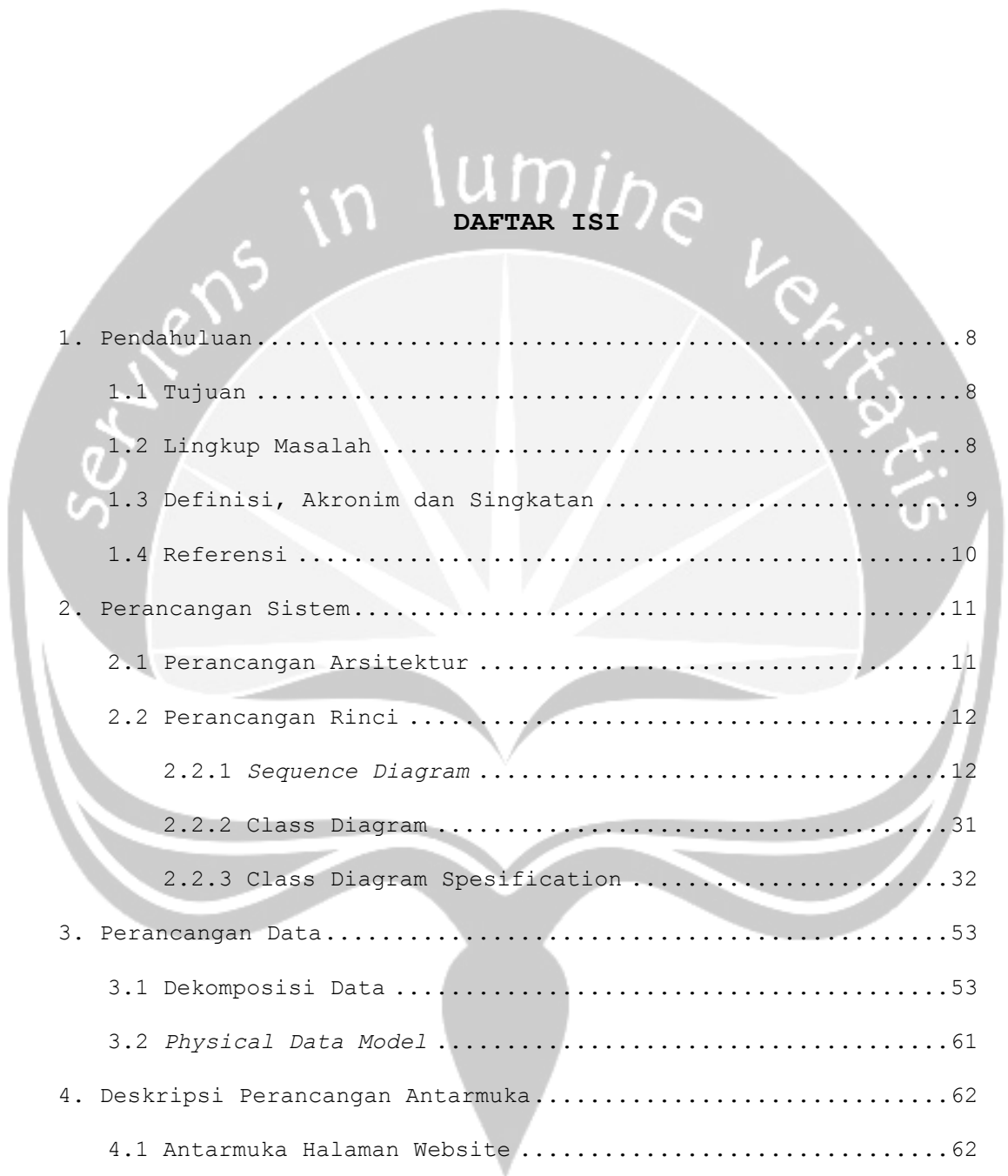
DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	

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

DAFTAR HALAMAN PERUBAHAN

Halaman	Revisi	Halaman	Revisi



DAFTAR ISI


- 1. Pendahuluan.....8
 - 1.1 Tujuan8
 - 1.2 Lingkup Masalah8
 - 1.3 Definisi, Akronim dan Singkatan9
 - 1.4 Referensi10
- 2. Perancangan Sistem.....11
 - 2.1 Perancangan Arsitektur11
 - 2.2 Perancangan Rinci12
 - 2.2.1 *Sequence Diagram*12
 - 2.2.2 *Class Diagram*31
 - 2.2.3 *Class Diagram Spesification*32
- 3. Perancangan Data.....53
 - 3.1 Dekomposisi Data53
 - 3.2 *Physical Data Model*61
- 4. Deskripsi Perancangan Antarmuka.....62
 - 4.1 Antarmuka Halaman Website62
 - 4.2 Antarmuka Halaman Layanan62
 - 4.3 Antarmuka Halaman Daftar Baru63

4.4	Antarmuka Halaman PAD	64
4.5	Antarmuka Halaman Hasil Uji Kendaraan	64
4.6	Antarmuka Halaman Aplikasi Mobile	65
4.7	Antarmuka Halaman Aduan Masyarakat	65
4.8	Antarmuka Halaman <i>Login Web</i>	66
4.9	Antarmuka Halaman Pengelolaan Admin	66
4.10	Antarmuka Halaman Pengelolaan Tipe Permohonan	67
4.11	Antarmuka Halaman Pengelolaan Pemilik Kendaraan	67
4.4	Antarmuka Halaman Pengelolaan Kendaraan	68
4.5	Antarmuka Halaman Pengelolaan Daftar Uji	69
4.8	Antarmuka Halaman <i>Login Mobile</i>	69
4.9	Antarmuka Halaman Pengelolaan Kendaraan	70
4.10	Antarmuka Halaman Verifikasi Kendaraan	70
4.11	Antarmuka Halaman <i>Mobile</i>	71
4.12	Antarmuka Halaman Hasil Uji	72
4.13	Antarmuka Halaman <i>Timeline</i>	73

DAFTAR GAMBAR

Gambar 2.1	Rancangan Arsitektur SIPPKB	11
Gambar 2.2	<i>Sequence Diagram</i> : <i>Login</i>	12
Gambar 2.3	<i>Sequence Diagram</i> : Tambah Data Pengguna	13
Gambar 2.4	<i>Sequence Diagram</i> : Tampil Data Pengguna	14
Gambar 2.5	<i>Sequence Diagram</i> : Tampil Data Tarif Uji	14

Gambar 2.6	<i>Sequence Diagram</i> : Edit Data Tarif Uji	15
Gambar 2.7	<i>Sequence Diagram</i> : Edit Data Detil Uji Laik Jalan	16
Gambar 2.8	<i>Sequence Diagram</i> : Tampil Data Uji Laik Jalan	17
Gambar 2.9	<i>Sequence Diagram</i> : Tampil Data Calon Pemilik	17
Gambar 2.10	<i>Sequence Diagram</i> : Edit Data Calon Pemilik	18
Gambar 2.11	<i>Sequence Diagram</i> : Tambah Data Calon Kendaraan ..	19
Gambar 2.12	<i>Sequence Diagram</i> : Tampil Data Calon Kendaraan ..	20
Gambar 2.13	<i>Sequence Diagram</i> : Tampil Data Pemilik	20
Gambar 2.14	<i>Sequence Diagram</i> : Tambah Data Pemilik	21
Gambar 2.15	<i>Sequence Diagram</i> : Edit Data Pemilik	22
Gambar 2.16	<i>Sequence Diagram</i> : Tambah Data Kendaraan Uji	23
Gambar 2.17	<i>Sequence Diagram</i> : Edit Data Kendaraan Uji	24
Gambar 2.18	<i>Sequence Diagram</i> : Tampil Data Kendaraan Uji	25
Gambar 2.19	<i>Sequence Diagram</i> : Tampil Data Aduan	25
Gambar 2.20	<i>Sequence Diagram</i> : Edit Data Aduan	26
Gambar 2.21	<i>Sequence Diagram</i> : Hapus Data Aduan	27
Gambar 2.22	<i>Sequence Diagram</i> : Tambah Data Retribusi	28
Gambar 2.23	<i>Sequence Diagram</i> : Tampil Data Retribusi	29
Gambar 2.24	<i>Sequence Diagram</i> : Tampil Data Pengujian	29
Gambar 2.25	<i>Sequence Diagram</i> : Tambah Data Pengujian	30
Gambar 2.25	<i>Class Diagram</i> SIPPKB	31
Gambar 3.1	<i>Physical Data Model</i>	61
Gambar 4.1	Antarmuka Halaman Beranda	62
Gambar 4.2	Antarmuka Halaman Layanan	62
Gambar 4.3	Antarmuka Halaman Daftar Baru	63



Gambar 4.4 Antarmuka Halaman PAD	64
Gambar 4.5 Antarmuka Halaman Hasil Uji Kendaraan	64
Gambar 4.6 Antarmuka Halaman Aplikasi <i>Mobile</i>	65
Gambar 4.7 Antarmuka Halaman Aduan Masyarakat	65
Gambar 4.8 Antarmuka Halaman Login Web	66
Gambar 4.9 Antarmuka Halaman Pengelolaan Admin	66
Gambar 4.10 Antarmuka Halaman Pengelolaan Tipe Permohonan	67
Gambar 4.11 Antarmuka Halaman Pengelolaan Pemilik Kendaraan ..	67
Gambar 4.12 Antarmuka Halaman Pengelolaan Kendaraan	68
Gambar 4.13 Antarmuka Halaman Pengelolaan Daftar Uji	69
Gambar 4.14 Antarmuka Halaman Login <i>Mobile</i>	69
Gambar 4.15 Antarmuka Halaman Pengelolaan Kendaraan	70
Gambar 4.16 Antarmuka Halaman Pengelolaan Verifikasi	70
Gambar 4.17 Antarmuka Halaman <i>Mobile</i>	71
Gambar 4.18 Antarmuka Halaman Hasil Uji	72
Gambar 4.19 Antarmuka Halaman <i>Timeline</i>	73

1. Pendahuluan

1.1. Tujuan

Dokumen Perancangan Perangkat Lunak (DPPL) ini merupakan dokumen perancangan perangkat lunak SIPPKB (Sistem Informasi Pengelolaan Pengujian Kendaraan Bermotor) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak, perangkat keras, dan pengguna) dan atribut (feature-feature tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-SIPPKB ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2. Lingkup Masalah

Perangkat Lunak SIPPKB dikembangkan dengan tujuan untuk :

1. Pengelolaan pengguna sistem, pengelolaan data kendaraan pada masing-masing kategori kendaraan uji dan pengelolaan fasilitas layanan tambahan (pembekuan kendaraan, perubahan bentuk kendaraan, peringatan keterlambatan, dan perubahan jenis uji)
2. Proses transaksi pengujian kendaraan bermotor.
3. Proses transaksi layanan pengujian kendaraan bermotor mencakup jadwal tanggal pengujian dan pembayaran biaya uji kendaraan beserta penentuan denda keterlambatan uji kendaraan.
4. Pemantauan laporan pengelolaan pengujian kendaraan bermotor oleh kepala dinas dan kepala bidang angkutan dinas perhubungan kabupaten purbalingga.

1.3. Definisi, Akronim, dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Merupakan perancangan dari perangkat lunak yang akan dikembangkan.
SIPPKB	Perangkat lunak pengelolaan <i>web</i> dan <i>mobile</i> .
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network global</i> yang terdiri dari komputer dan layanan servis dengan

	sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk <i>e-mail</i> , FTP, dan <i>World Wide Web</i> .
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Pengujian kendaraan bermotor	Pengujian kendaraan bermotor disebut juga uji kir adalah serangkaian kegiatan menguji dan/atau memeriksa bagian-bagian kendaraan bermotor, kereta gandengan, kereta tempelan dan kendaraan khusus dalam rangka pemenuhan terhadap persyaratan teknis dan laik jalan.
BPDP	Bagian Pengelolaan Data Pengujian yang memiliki tugas untuk melakukan pengelolaan data pendukung kegiatan pengujian kendaraan bermotor.
BTP	Badan Teknis Pengujian
BLP	Badan Layanan Pengujian

1.4. Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

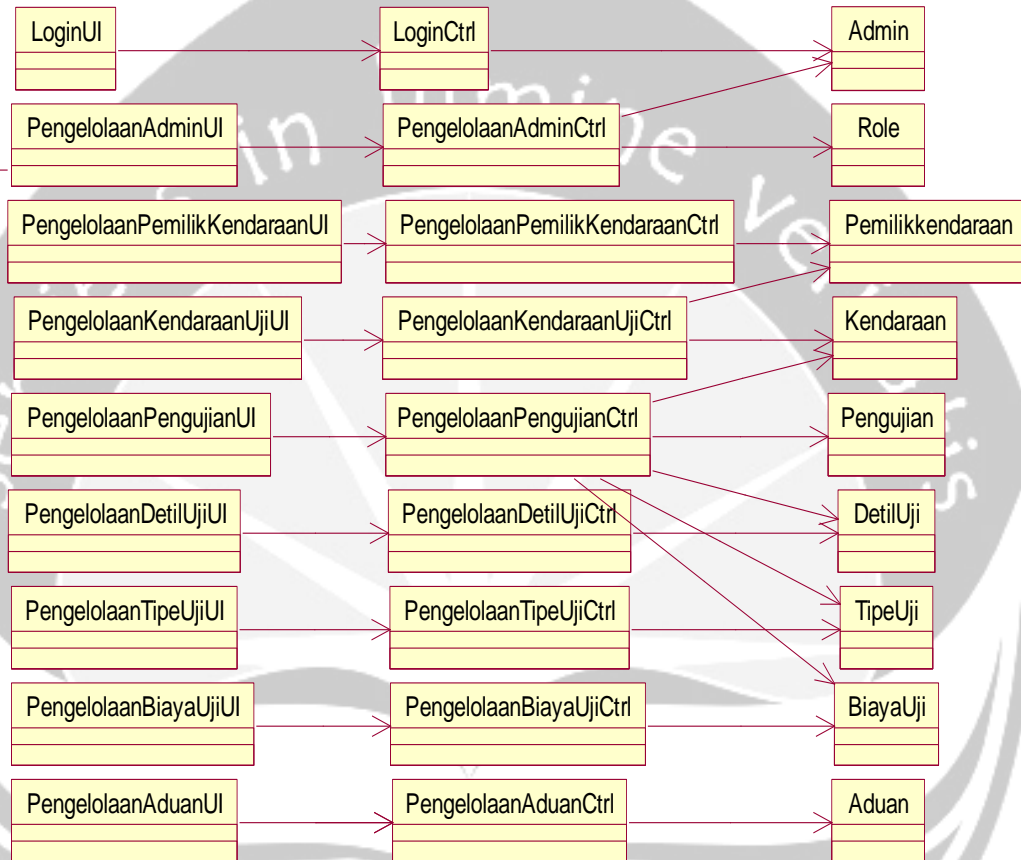
1. Andreas Dimas Setyoko / 130707538, Dokumen Perancangan Perangkat Lunak (DPPL) ILUSI, Universitas Atma Jaya Yogyakarta.
2. Soal Projek Pembangunan Perangkat Lunak Profesional (P3LP) semester ganjil, Universitas Atma Jaya Yogyakarta.



2. Perancangan Sistem

2.1. Perancangan

Arsitektur

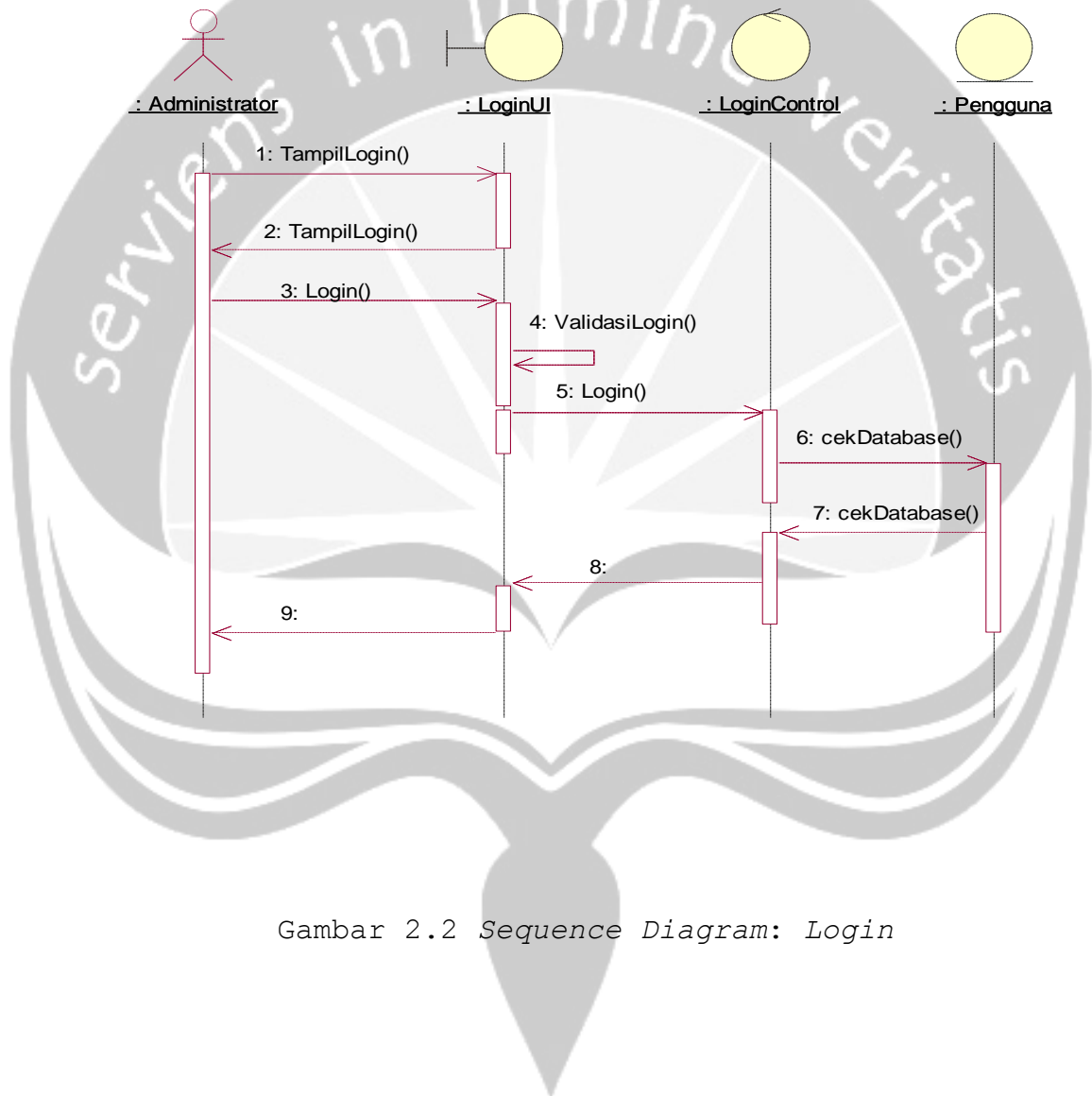


Gambar 2.1 Rancangan Arsitektur SIPPKB

2.2 Perancangan Rinci

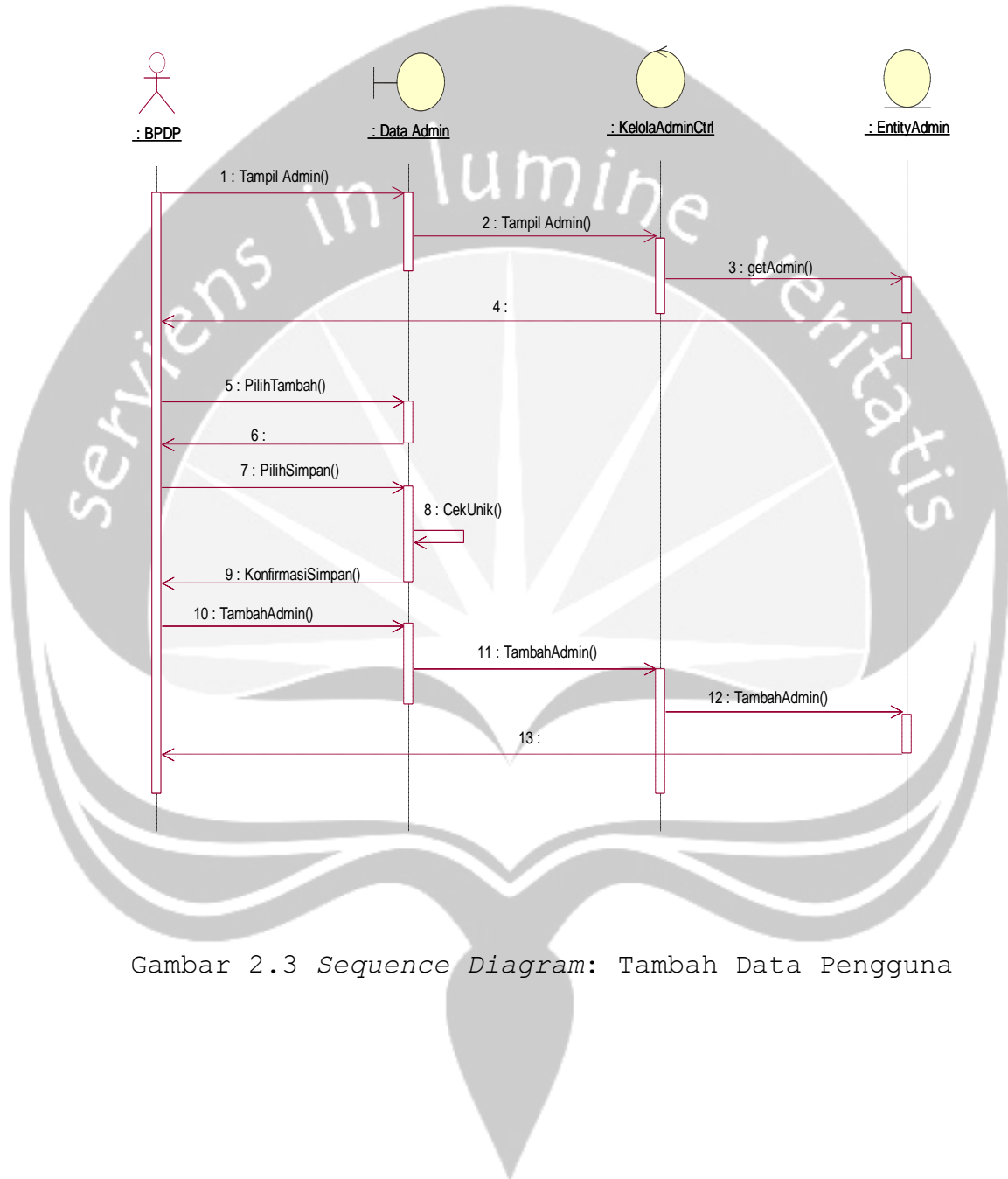
2.2.1. Sequence Diagram

2.2.1.1. Fungsi Login



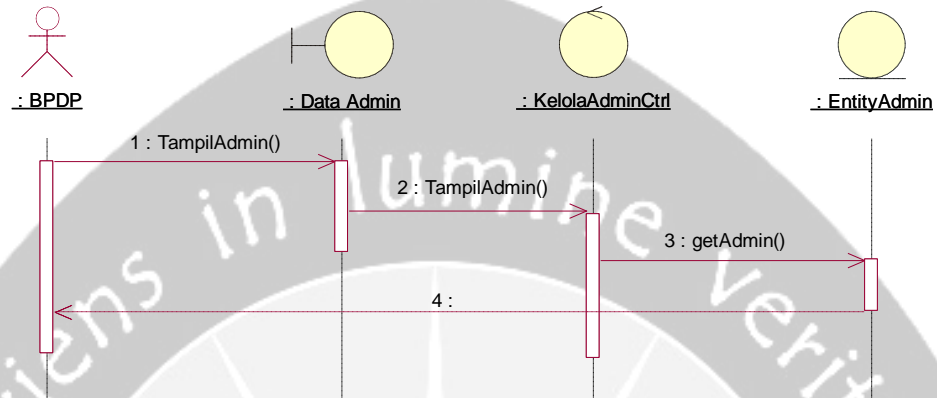
Gambar 2.2 Sequence Diagram: Login

2.2.1.2 Fungsi Tambah Data Pengguna



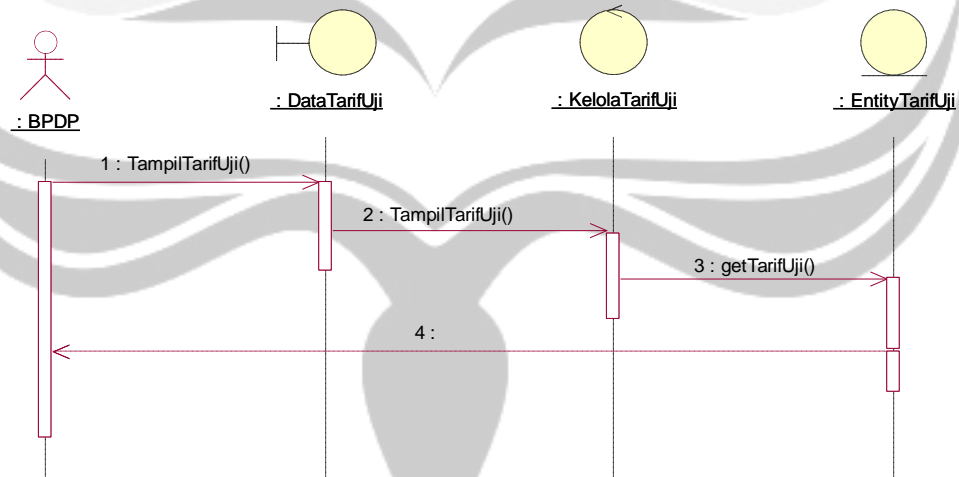
Gambar 2.3 Sequence Diagram: Tambah Data Pengguna

2.2.1.3. Fungsi Tampil Data Pengguna



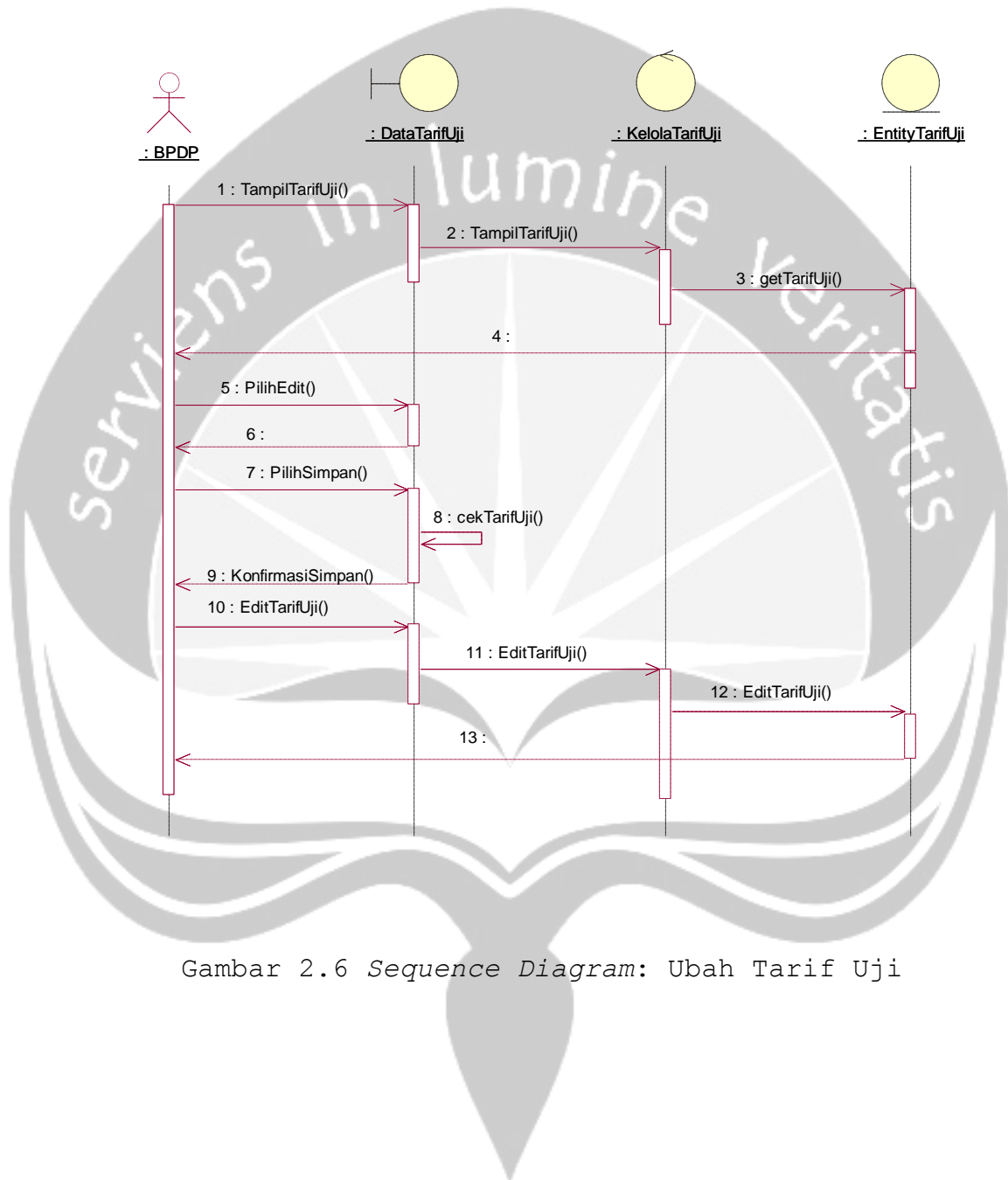
Gambar 2.4 *Sequence Diagram*: Tampil Data Pengguna

2.2.1.3. Fungsi Tampil Data Tarif Uji



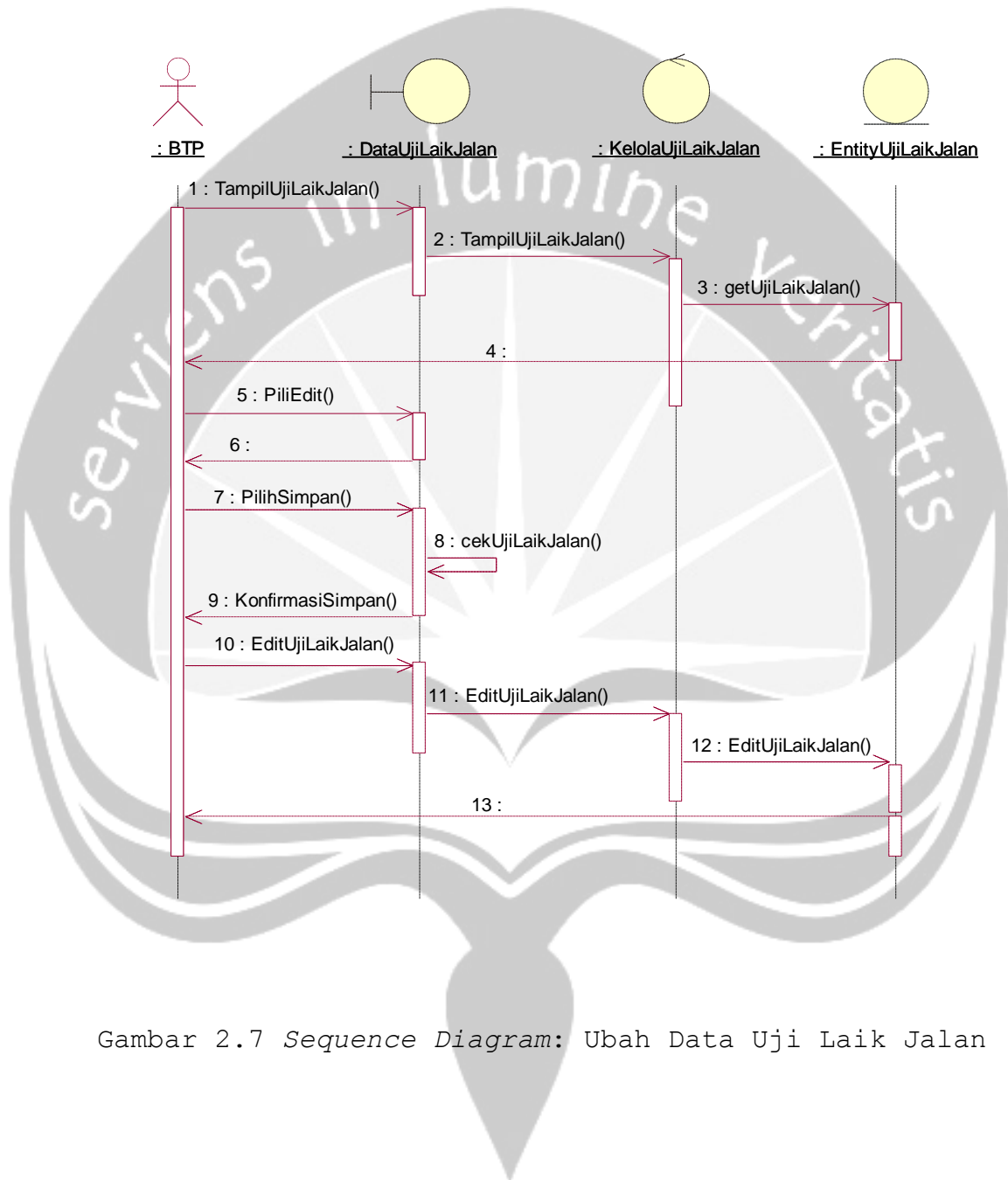
Gambar 2.5 *Sequence Diagram*: Tampil Data Tarif Uji

2.2.1.4. Fungsi *Edit Data Tarif Uji*



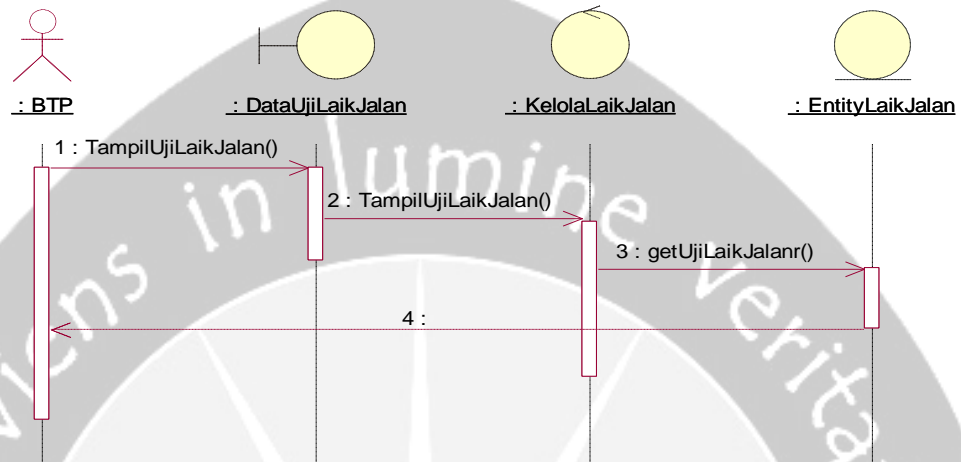
Gambar 2.6 *Sequence Diagram*: Ubah Tarif Uji

2.2.1.5. Fungsi *Edit* Data Detil Uji Laik Jalan



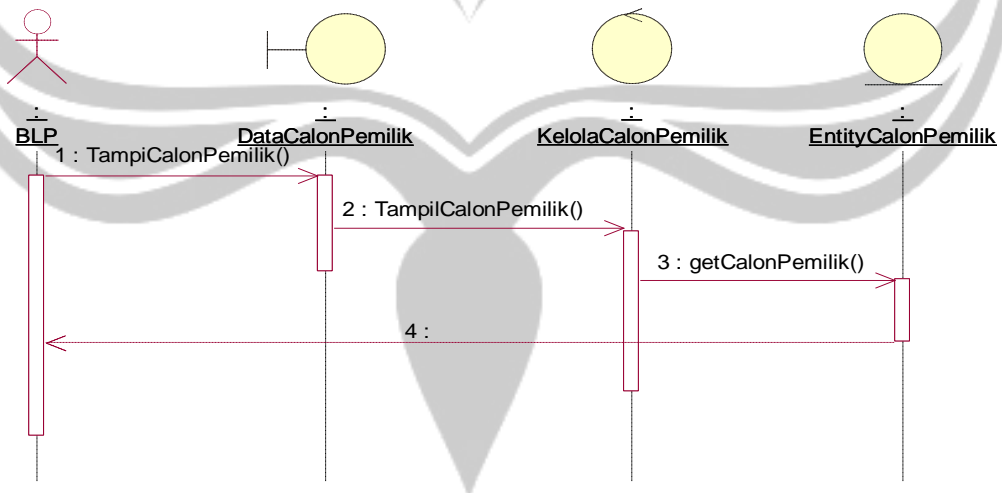
Gambar 2.7 *Sequence Diagram*: Ubah Data Uji Laik Jalan

2.2.1.6. Fungsi Tampil Data Detil Uji Laik Jalan



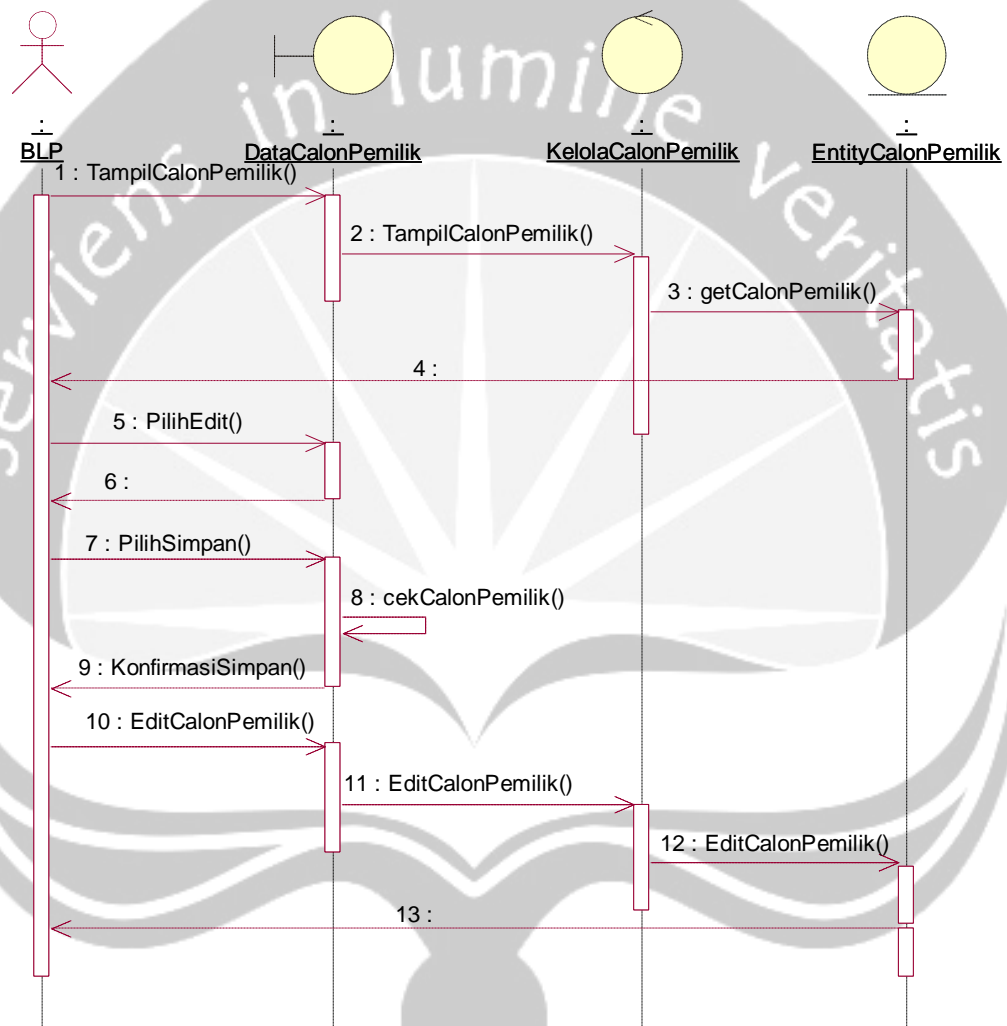
Gambar 2.8 Sequence Diagram: Tampil Data Uji Laik Jalan

2.2.1.7. Fungsi Tampil Data Calon Pemilik Kendaraan



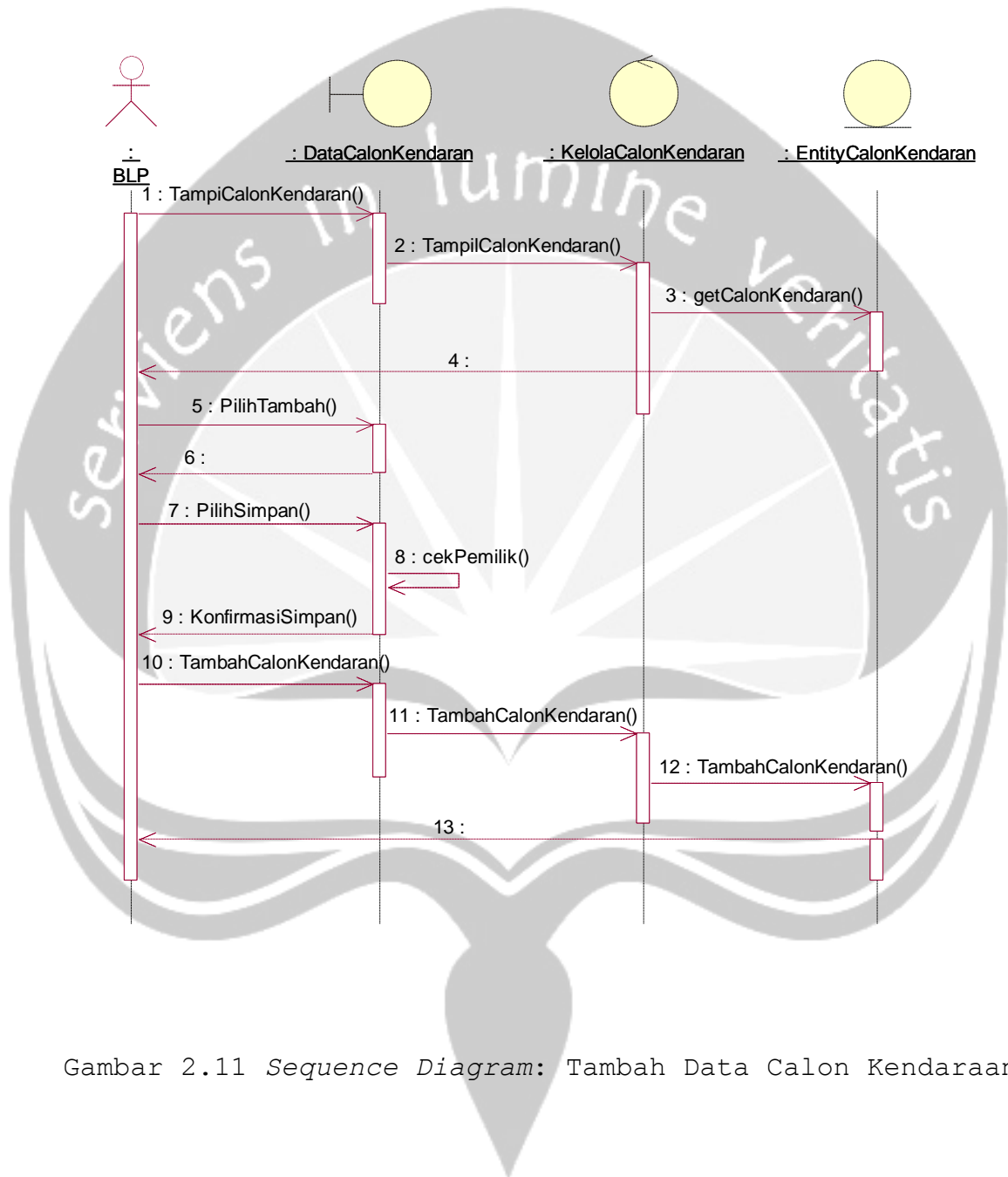
Gambar 2.9 Sequence Diagram: Tampil Data Calon Pemilik

2.2.1.8. Fungsi *Edit* Data Calon Pemilik Kendaraan



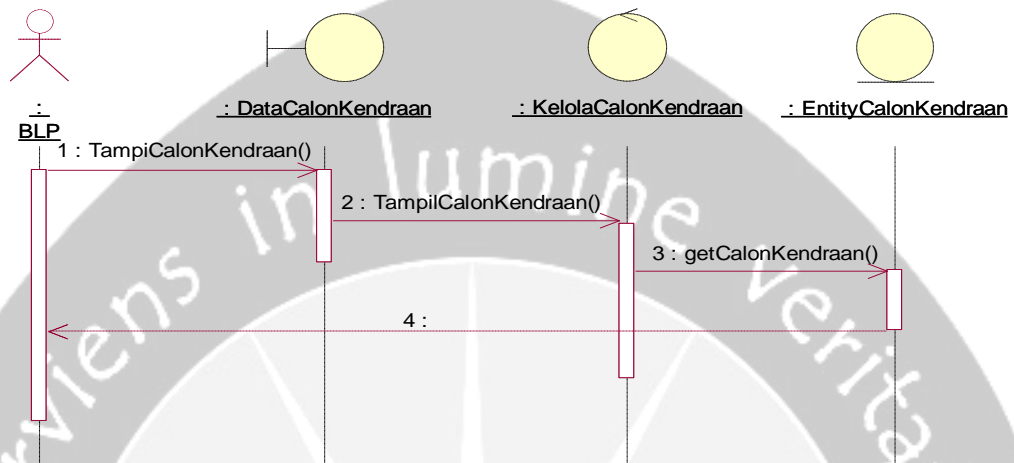
Gambar 2.10 *Sequence Diagram*: Ubah Data Calon Pemilik

2.2.1.9. Fungsi Tambah Data Calon Kendaraan Uji



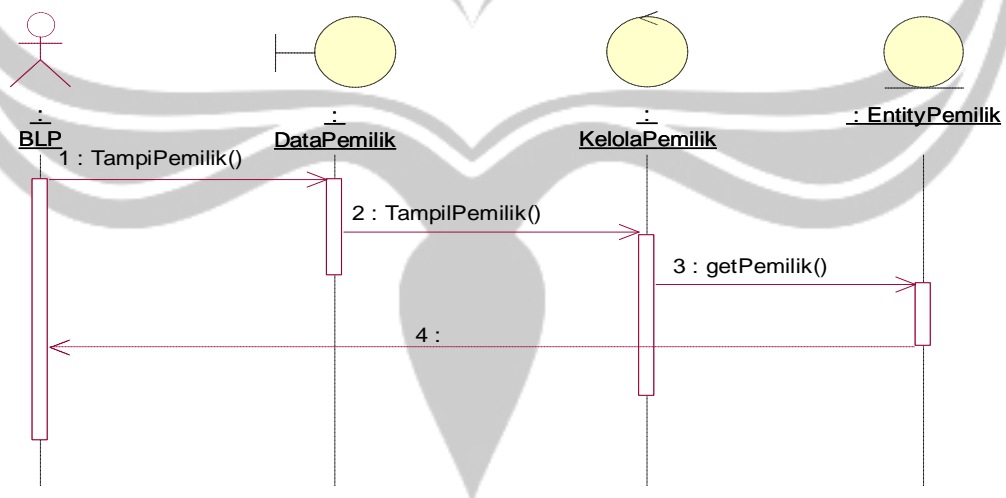
Gambar 2.11 *Sequence Diagram*: Tambah Data Calon Kendaraan

2.2.1.10. Fungsi Tampil Data Calon Kendaraan Uji



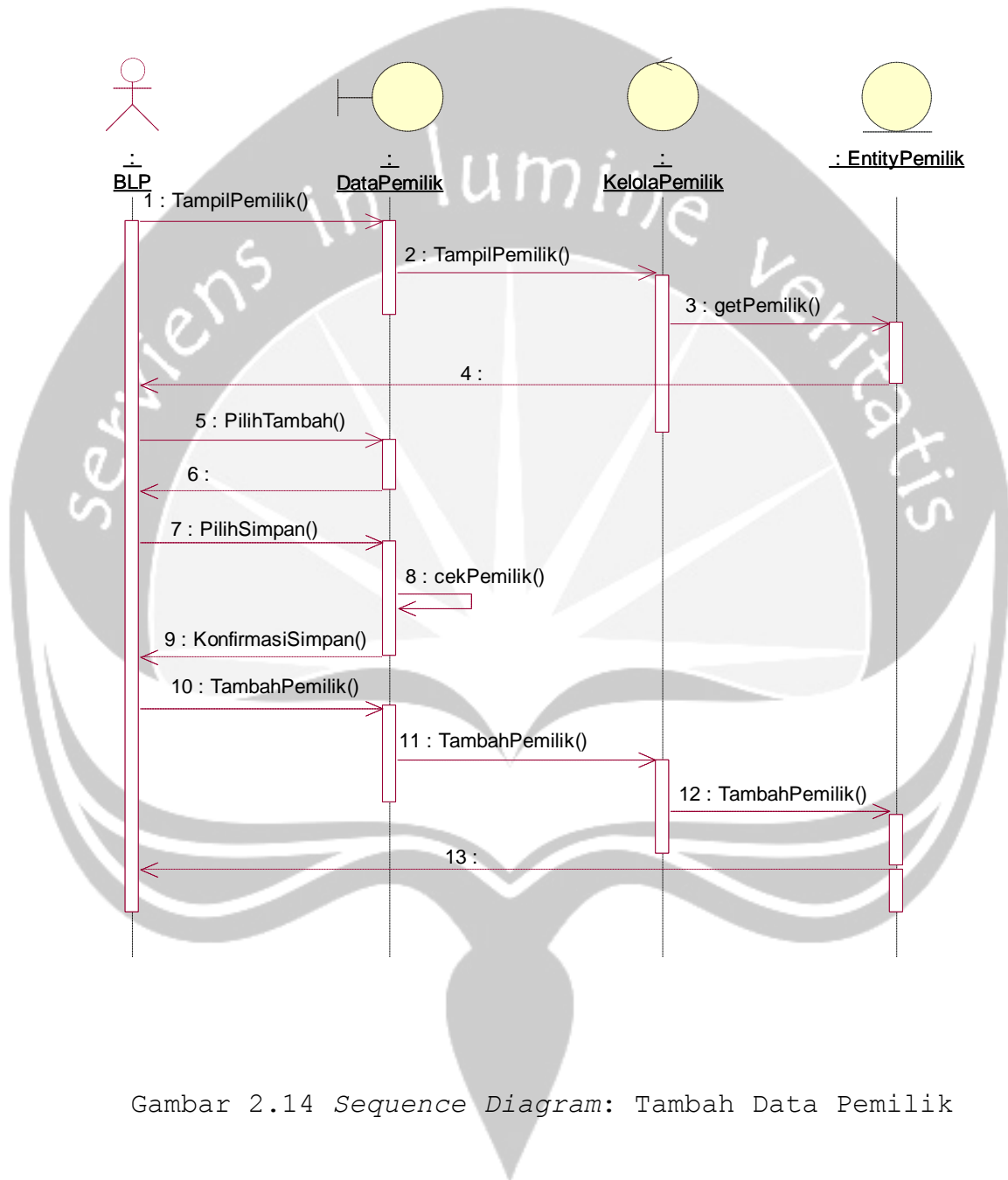
Gambar 2.12 *Sequence Diagram*: Tampil Data Calon Kendaraan

2.2.1.11. Fungsi Tampil Data Pemilik



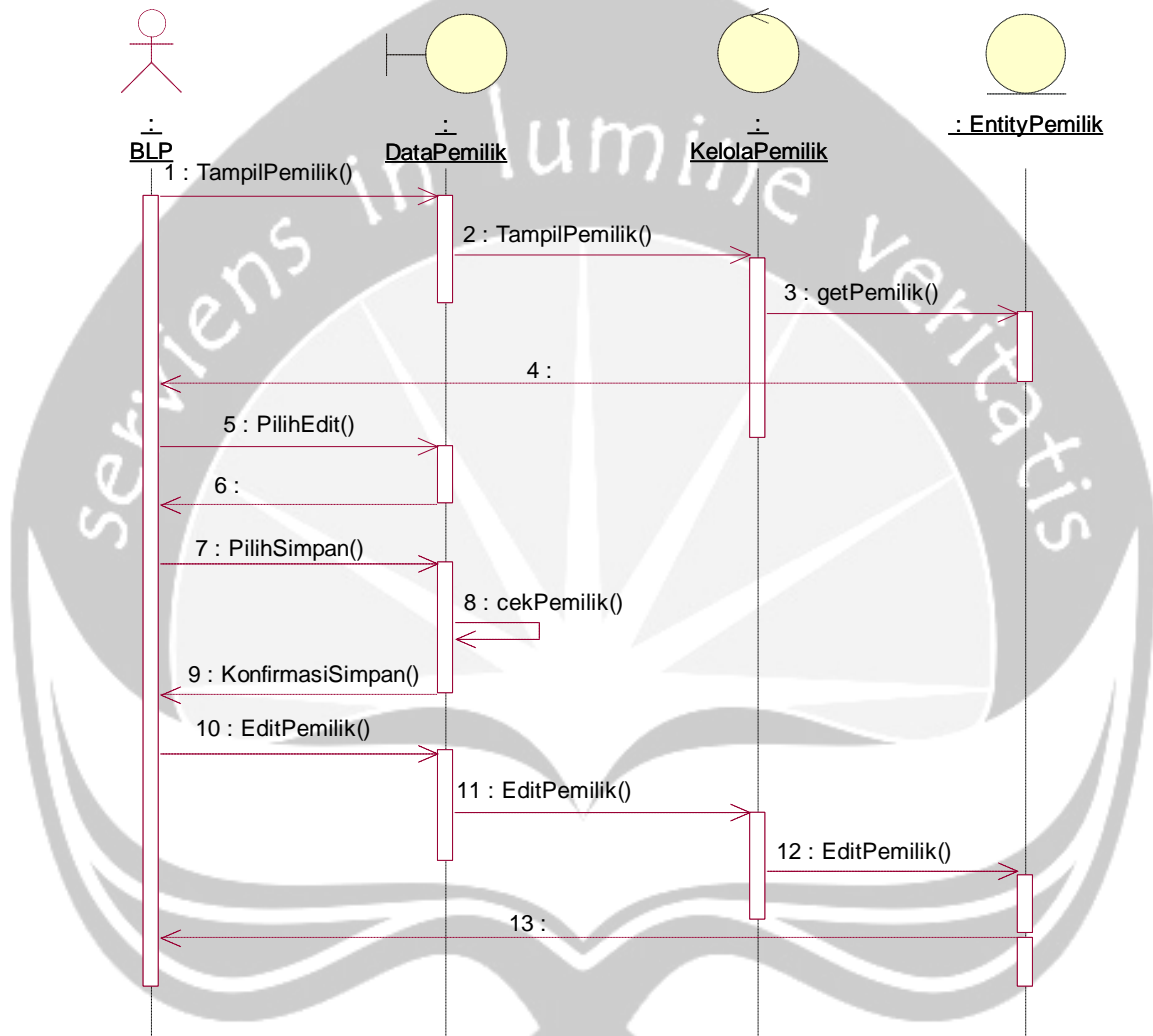
Gambar 2.13 *Sequence Diagram*: Tampil Data Calon Kendaraan

2.2.1.12. Fungsi Tambah Data Pemilik



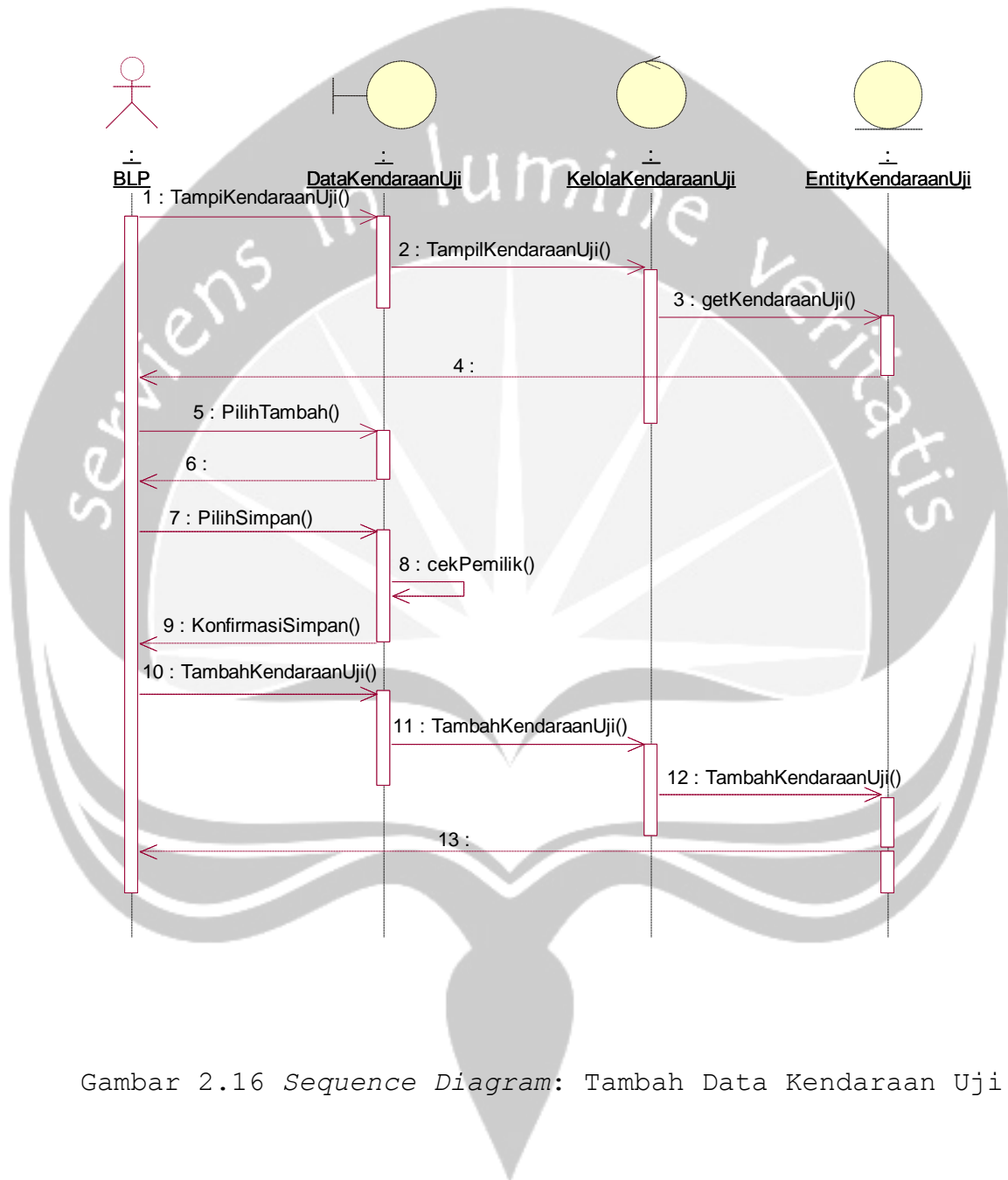
Gambar 2.14 Sequence Diagram: Tambah Data Pemilik

2.2.1.13. Fungsi *Edit Data Pemilik*



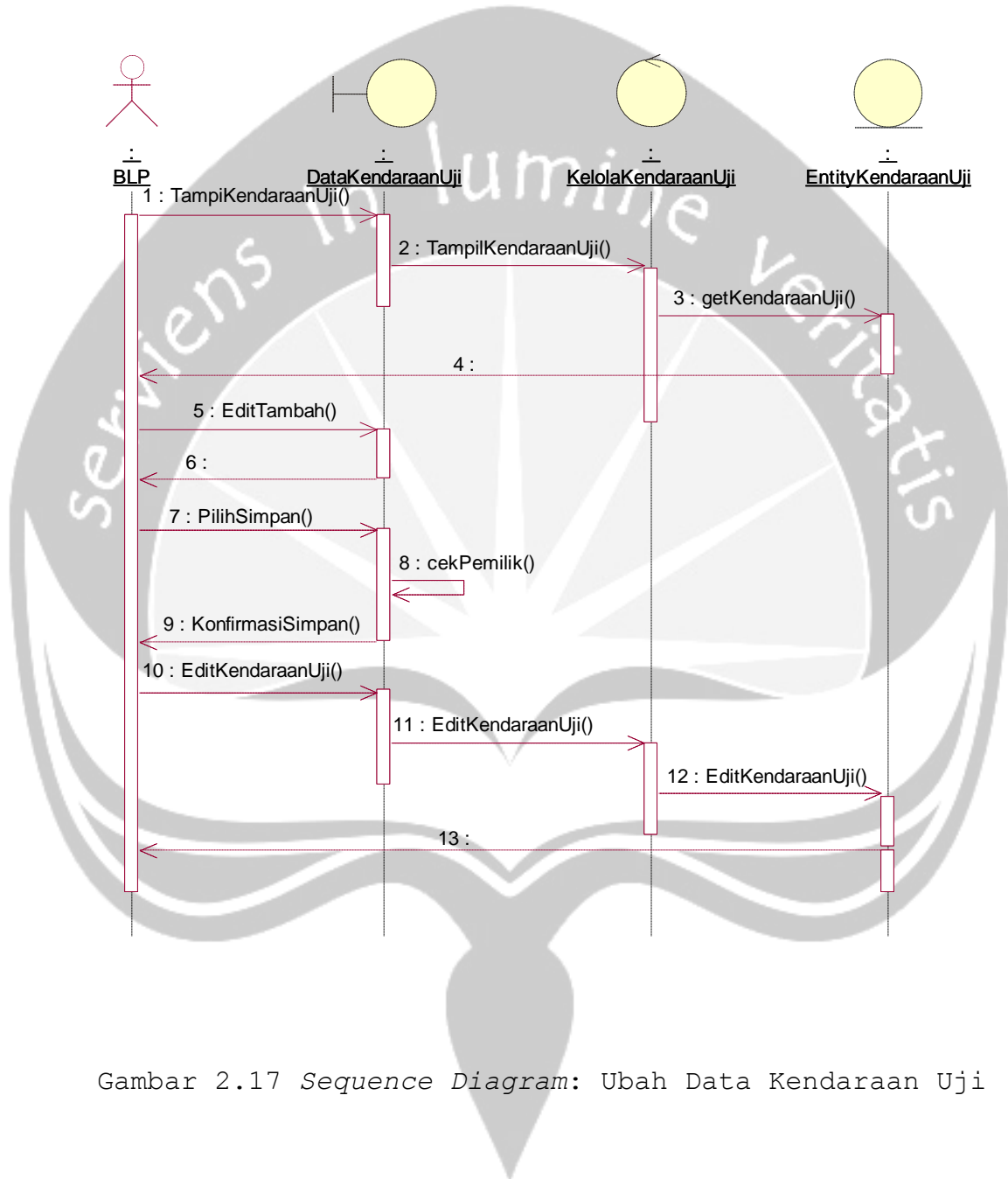
Gambar 2.15 *Sequence Diagram*: Ubah Data Pemilik

2.2.1.14. Fungsi Tambah Data Kendaraan Uji



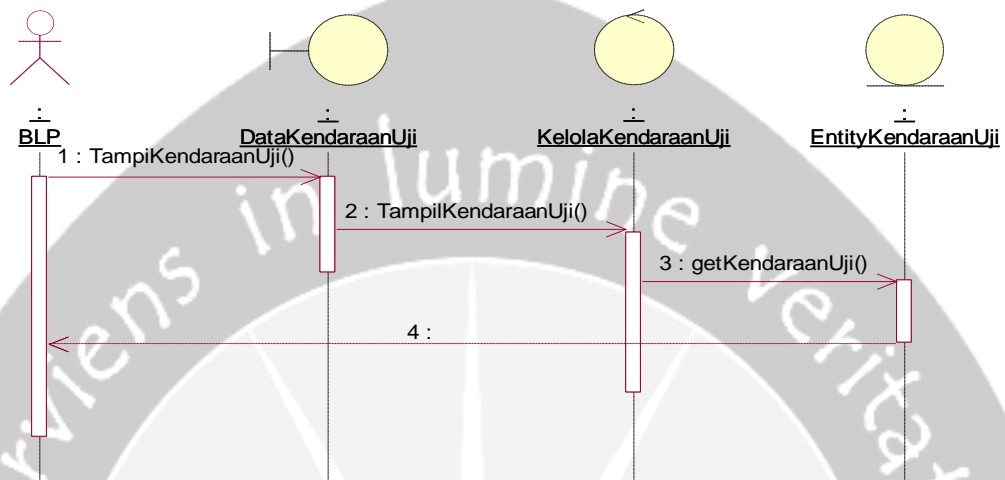
Gambar 2.16 *Sequence Diagram*: Tambah Data Kendaraan Uji

2.2.1.15. Fungsi *Edit Data Kendaraan Uji*



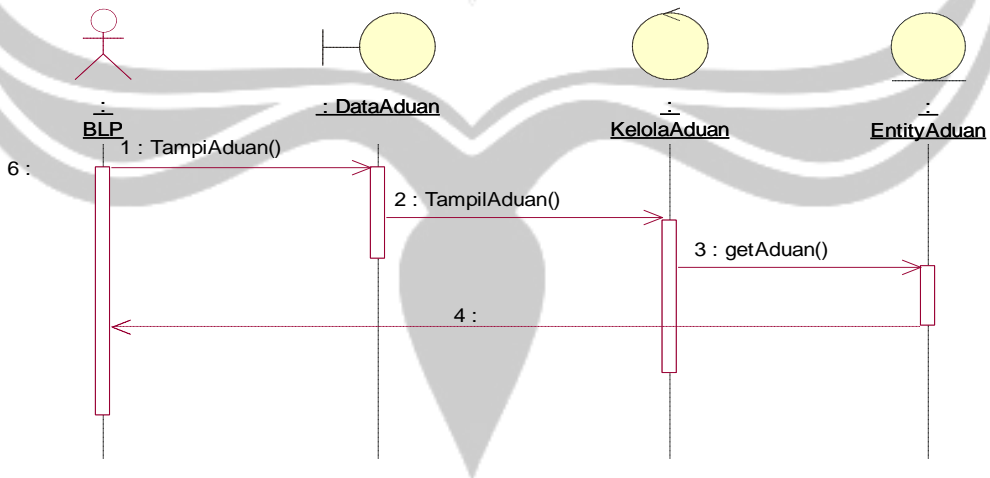
Gambar 2.17 *Sequence Diagram*: Ubah Data Kendaraan Uji

2.2.1.16. Fungsi Tampil Data Kendaraan Uji



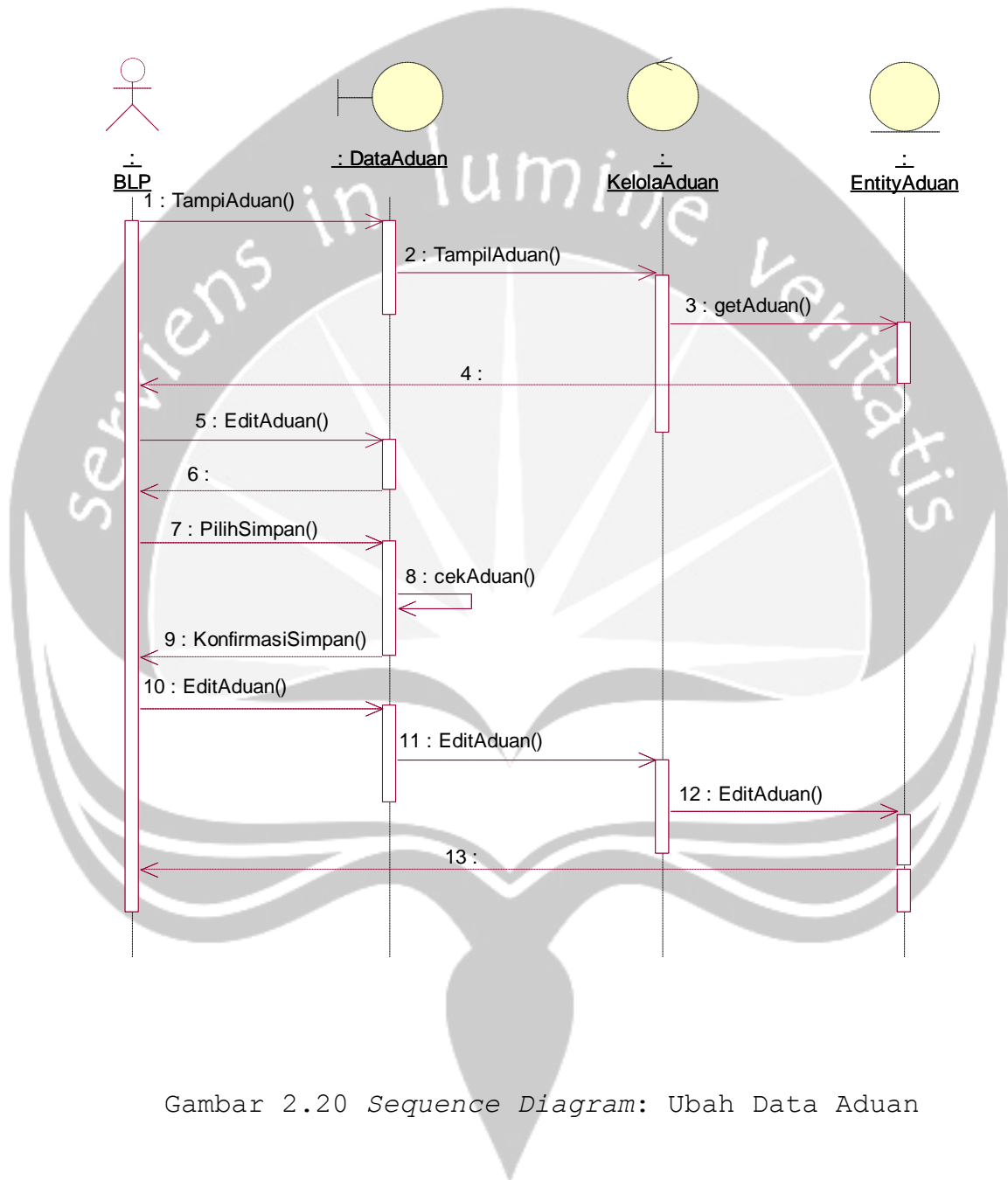
Gambar 2.18 *Sequence Diagram*: Tampil Data Kendaraan Uji

2.2.1.17. Tampil Data Aduan



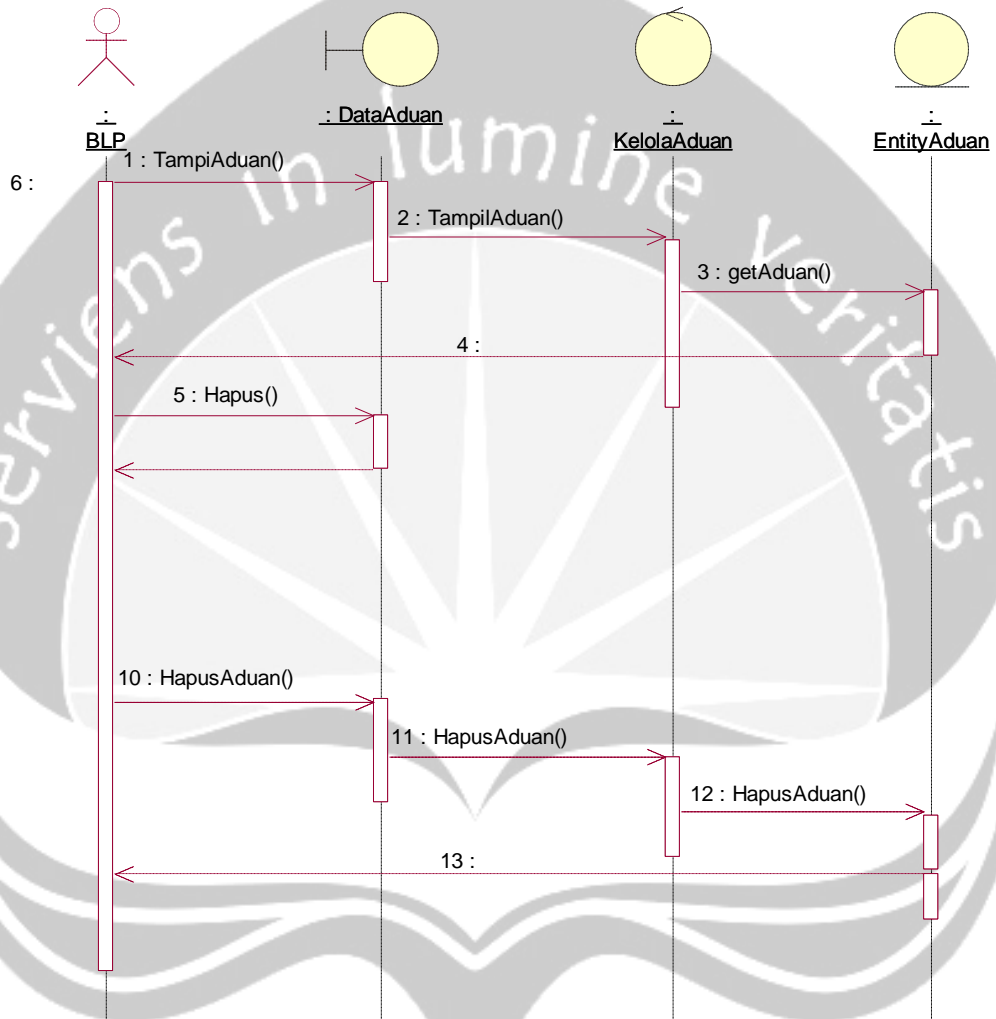
Gambar 2.19 *Sequence Diagram*: Tampil Data Aduan

2.2.1.18. Fungsi *Edit Data Aduan*



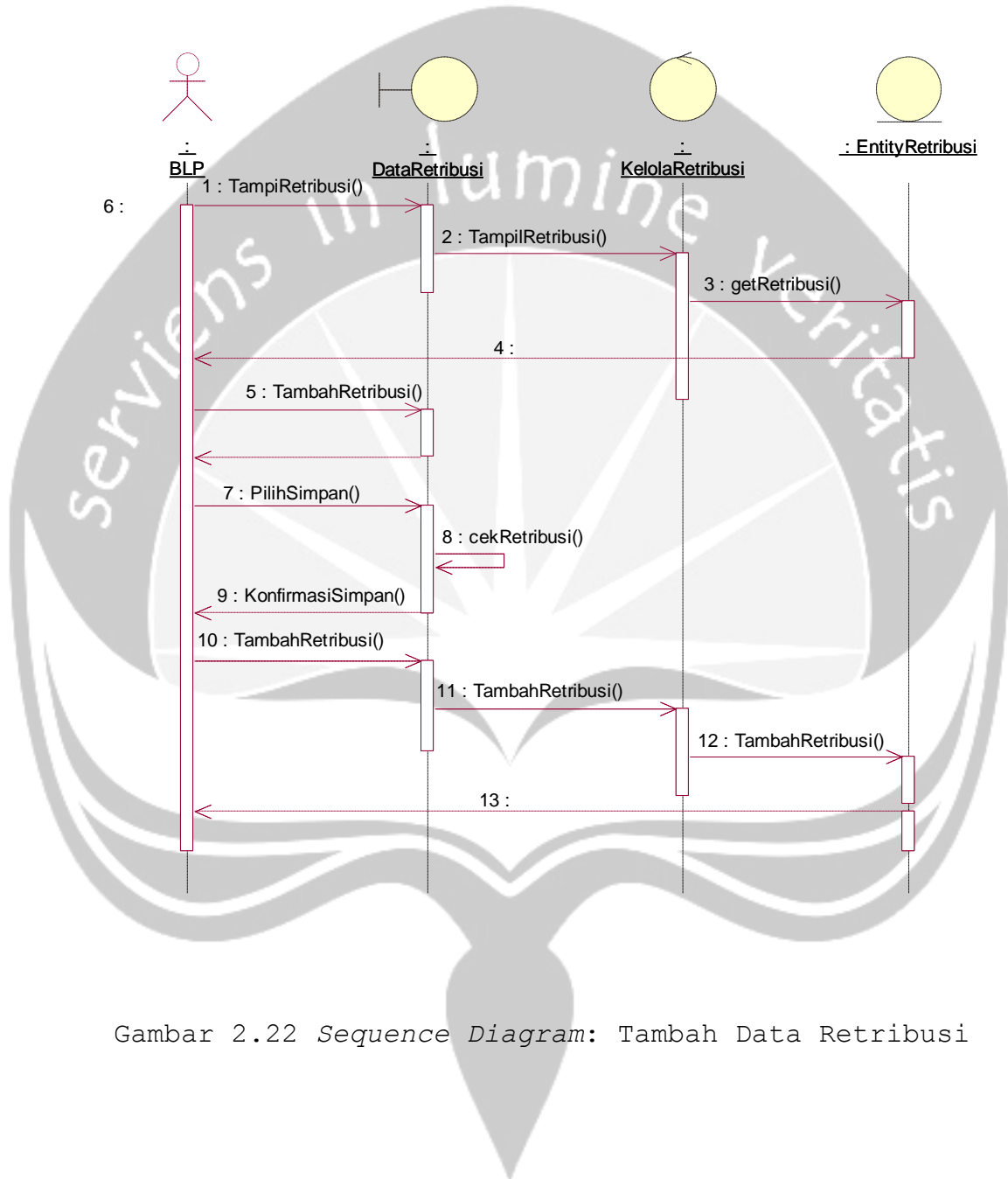
Gambar 2.20 *Sequence Diagram*: Ubah Data Aduan

2.2.1.19. Fungsi Hapus Data Aduan



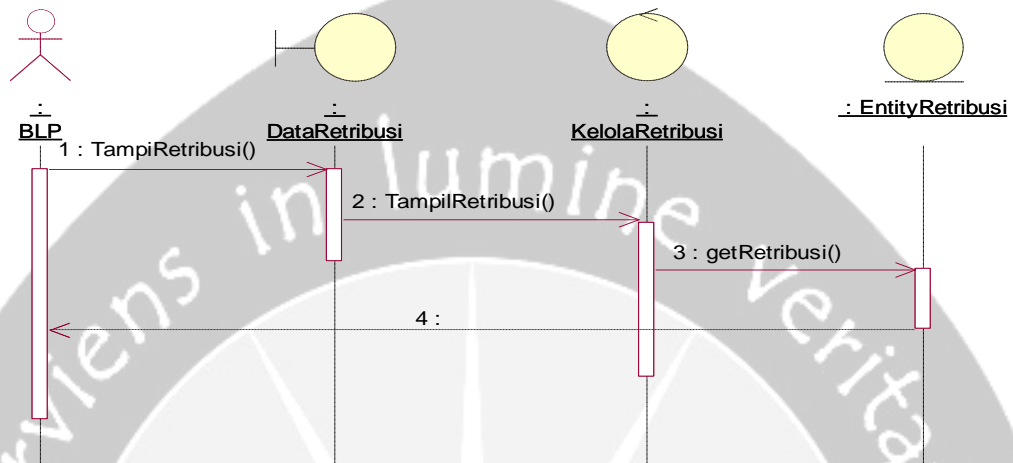
Gambar 2.21 *Sequence Diagram*: Hapus Data Aduan

2.2.1.20. Fungsi Tambah Data Retribusi



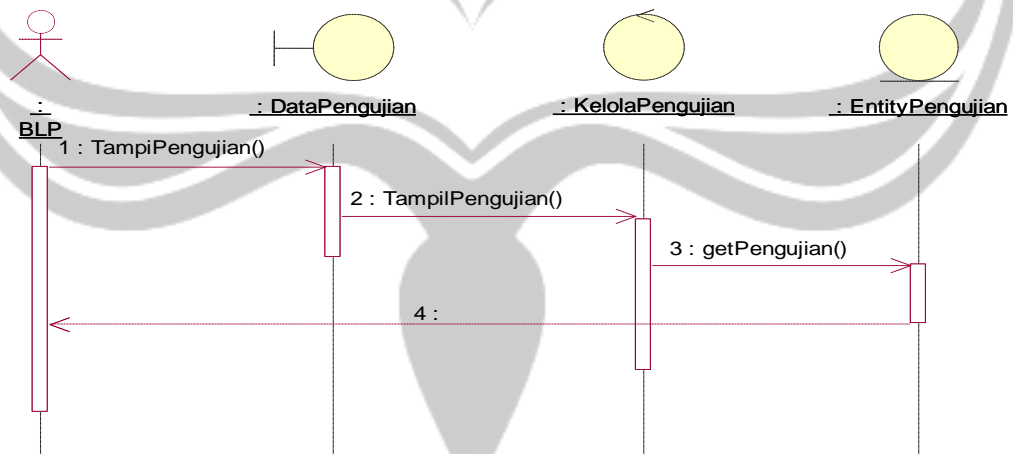
Gambar 2.22 Sequence Diagram: Tambah Data Retribusi

2.2.1.21. Fungsi Tampil Data Retribusi



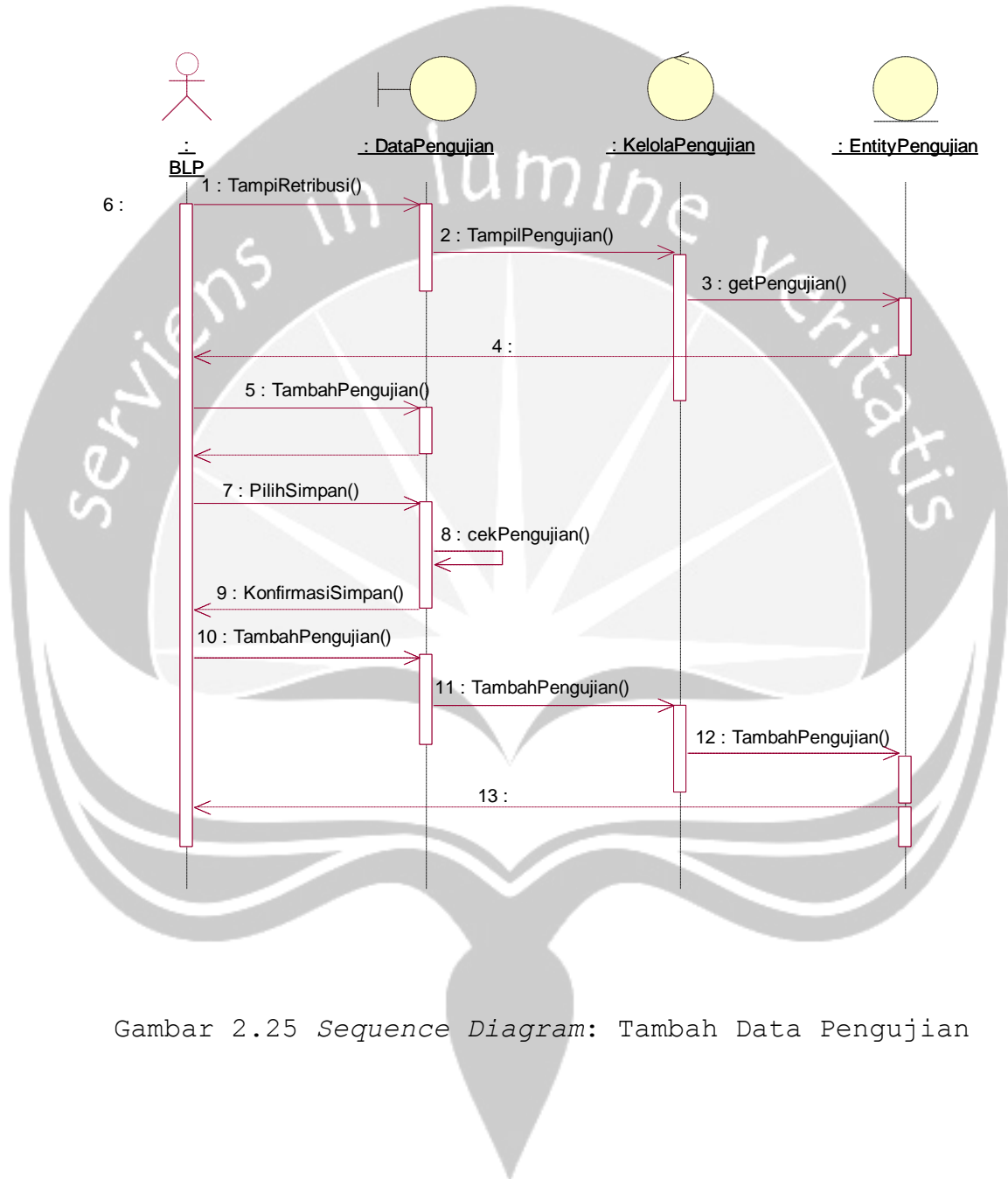
Gambar 2.23 Sequence Diagram: Tampil Data Retribusi

2.2.1.22. Tampil Data Pengujian



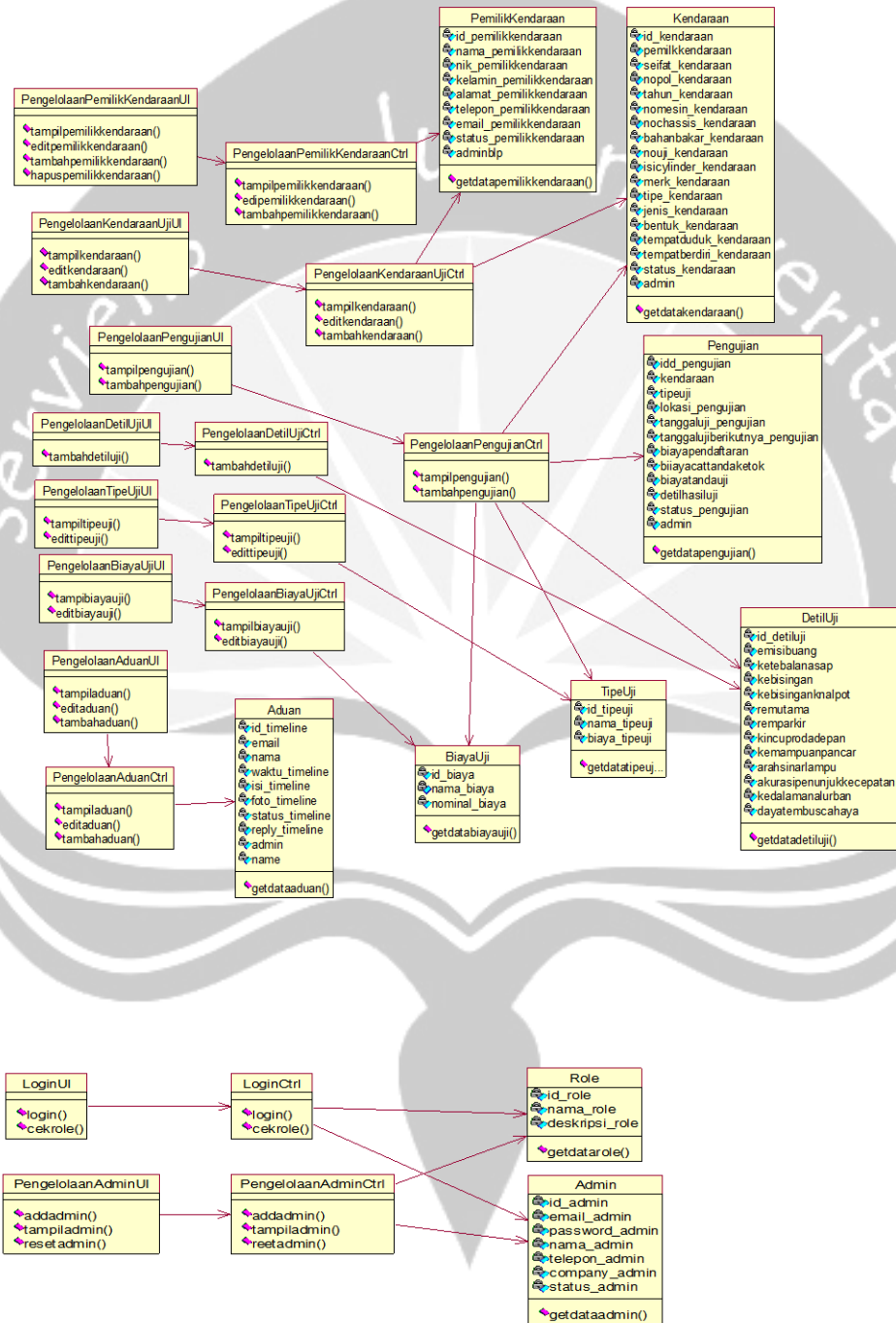
Gambar 2.24 Sequence Diagram: Tampil Data Pengujian

2.2.1.23. Tambah Data Pengujian



Gambar 2.25 *Sequence Diagram*: Tambah Data Pengujian

2.2.2 Class Diagram



Gambar 2.27 Rancangan Class Diagram SIPPKB

2.2.3 Class Diagram Spesific Description

2.2.3.1 Specific Design Class DataAdminUI

DataAdminUI	<<boundary>>
+addAdminUI() Digunakan untuk menambah data pengguna	
+TampilAdmin() Operasi untuk menampilkan semua data pengguna yang ada di basis data.	
+ResetAdmin() Operasi untuk menerima perintah pengubahan data pengguna yang ada di basis data.	

2.2.3.2. Specific Design Class

PengelolaanPemilikKendaraanUI

PengelolaanPemilikKendaraanUI	<<boundary>>
+DatapemilikkednaraanUI()	

Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.

+Tambahpemilikkendaraan()

Operasi untuk menerima perintah penambahan data pemilik kendaraan yang ada di basis data.

+Editpemilikkendaraan()

Operasi untuk menerima perintah pengubahan data pemilik kendaraan yang ada di basis data.

+Tampilpemilikkendaraan()

Operasi untuk menampilkan semua data pemilik kendaraan yang ada di basis data.

2.2.3.3. *Specific Design Class* PengelolaanKendaraanUjiUI

PengelolaanKendaraanUjiUI	<<boundary>>
---------------------------	--------------

+Datapengelolaankendaraanuji()

Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.

+Entrikendaraanuji()

Operasi untuk menerima perintah penambahan data kendaraanuji.

+Editkendaraanuji()

Operasi untuk menerima perintah pengubahan data kendaraan uji yang ada di basis data.

+Tampilkendaraanuji()

Operasi untuk menampilkan semua data kendaraan uji yang ada di basis data.

2.2.3.4. Specific Design Class PengelolaanPengujianUI

PengelolaanPengujianUI	<<boundary>>
+Datapengelolaanpengujian()	
Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.	
+Entridatapengujian()	
Operasi untuk menerima perintah penambahan data pengujian.	
+Tampilpengujian()	
Operasi untuk menampilkan semua data pengujian yang ada di basis data.	

2.2.3.5. Specific Design Class PengelolaaDetilUjiUI

PengelolaaDetilUjiUI	<<boundary>>
<pre>+Datapengelolaandetiluji() Default konstruktor untuk menginisialisasi semua atribut dari kelas ini. +Entridetiluji() Operasi untuk menerima perintah penambahan data detil uji. +Tampildetiluji() Operasi untuk menampilkan semua data detil uji yang ada di basis data.</pre>	

2.2.3.6. *Specific Design Class* PengelolaanTipeUjiUI

PengelolaanTipeUjiUI	<<boundary>>
<pre>+Datapengelolaantipeuji() Default konstruktor untuk menginisialisasi semua atribut dari kelas ini. +Edittipeuji() Operasi untuk menerima perintah pengubahan data tipe uji yang ada di basis data.</pre>	

<p>+Tampillokasi()</p> <p>Operasi untuk menampilkan semua data lokasi yang ada di basis data.</p>

2.2.3.7. Specific Design Class PengelolaanBiayaUjiUI

PengelolaanBiayaUjiUI	<<boundary>>
<p>+Datapengelolaanbiayauji()</p> <p>Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.</p> <p>+Editbiayauji()</p> <p>Operasi untuk menerima perintah perubahan data biaya uji yang ada di basis data.</p> <p>+Tampilbiayauji()</p> <p>Operasi untuk menampilkan semua data biaya uji yang ada di basis data.</p>	

2.2.3.8. Specific Design Class PengelolaanAduanUI

PengelolaanBiayaUjiUI	<<boundary>>
<p>+Datapengelolaanaduan()</p>	

Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.

+Tambahaduan()

Operasi untuk menerima perintah menambah data aduan yang ada di basis data.

+Editaduan()

Operasi untuk menerima perintah pengubahan data aduan yang ada di basis data.

+Tampiladuan()

Operasi untuk menampilkan semua data aduan yang ada di basis data.

2.2.3.9. Specific Design Class LoginUI

LoginUI

<<boundary>>

+Datalogin()

Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.

+login()

Operasi untuk menerima perintah masuk ke sistem.

+cekrole()

Operasi untuk menerima perintah pengecekan data pengguna yang ada di basis data.

2.2.3.10. *Specific Design Class DataAdminCtrl*

DataAdminCtrl	<<control>>
+addAdminUI () Digunakan untuk menambah data pengguna	
+TampilAdmin () Operasi untuk menampilkan semua data pengguna yang ada di basis data.	
+ResetAdmin () Operasi untuk menerima perintah pengubahan data pengguna yang ada di basis data.	

2.2.3.11. *Specific Design Class*

PengelolaanPemilikKendaraanCtrl

PengelolaanPemilikKendaraanCtrl	<<control>>
+DatapemilikkednaraanUI ()	

Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.

+Tambahpemilikkendaraan()

Operasi untuk menerima perintah penambahan data pemilik kendaraan yang ada di basis data.

+Editpemilikkendaraan()

Operasi untuk menerima perintah pengubahan data pemilik kendaraan yang ada di basis data.

+Tampilpemilikkendaraan()

Operasi untuk menampilkan semua data pemilik kendaraan yang ada di basis data.

2.2.3.12. *Specific Design Class*

PengelolaanKendaraanUjiCtrl

PengelolaanKendaraanUjiCtrl	<<control>>
+Datapengelolaankendaraanuji() Default konstruktor untuk menginisialisasi semua atribut dari kelas ini. +Entrikendaraanuji() Operasi untuk menerima perintah penambahan data kendaraanuji.	

+Editkendaraanuji()

Operasi untuk menerima perintah pengubahan data kendaraan uji yang ada di basis data.

+Tampilkendaraanuji()

Operasi untuk menampilkan semua data kendaraan uji yang ada di basis data.

2.2.3.13. *Specific Design Class* PengelolaanPenguujianCtrl

PengelolaanPenguujianCtrl	<<control>>
---------------------------	-------------

+Datapengelolaanpenguujian()

Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.

+Entridatapenguujian()

Operasi untuk menerima perintah penambahan data penguujian.

+Tampilpenguujian()

Operasi untuk menampilkan semua data penguujian yang ada di basis data.

2.2.3.14. *Specific Design Class* PengelolaaDetilUjiCtrl

PengelolaaDetilUjiCtrl	<<control>>
<pre>+Datapengelolaandetiluji() Default konstruktor untuk menginisialisasi semua atribut dari kelas ini. +Entridetiluji() Operasi untuk menerima perintah penambahan data detil uji. +Tampildetiluji() Operasi untuk menampilkan semua data detil uji yang ada di basis data.</pre>	

2.2.3.15. *Specific Design Class* PengelolaantipeUjiCtrl

PengelolaantipeUjiCtrl	<<control>>
<pre>+Datapengelolaantipeuji() Default konstruktor untuk menginisialisasi semua atribut dari kelas ini. +Edittipeuji() Operasi untuk menerima perintah pengubahan data tipe uji yang ada di basis data.</pre>	

<p>+Tampillokasi()</p> <p>Operasi untuk menampilkan semua data lokasi yang ada di basis data.</p>

2.2.3.16. Specific Design Class PengelolaanBiayaUjiCtrl

PengelolaanBiayaUjiCtrl	<<control>>
<p>+Datapengelolaanbiayauji()</p> <p>Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.</p> <p>+Editbiayauji()</p> <p>Operasi untuk menerima perintah perubahan data biaya uji yang ada di basis data.</p> <p>+Tampilbiayauji()</p> <p>Operasi untuk menampilkan semua data biaya uji yang ada di basis data.</p>	

2.2.3.17. Specific Design Class PengelolaanAduanCtrl

PengelolaanBiayaUjiCtrl	<<control>>
<p>+Datapengelolaanaduan()</p>	

Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.

+Tambahaduan()

Operasi untuk menerima perintah menambah data aduan yang ada di basis data.

+Editaduan()

Operasi untuk menerima perintah pengubahan data aduan yang ada di basis data.

+Tampiladuan()

Operasi untuk menampilkan semua data aduan yang ada di basis data.

2.2.3.18. *Specific Design Class LoginCtrl*

<code>LoginCtrl</code>	<code><<control>></code>
<p>+Datalogin()</p> <p>Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.</p> <p>+login()</p> <p>Operasi untuk menerima perintah masuk ke sistem.</p>	

+cekrole()

Operasi untuk menerima perintah pengecekan data pengguna yang ada di basis data.

2.2.3.19. *Specific Design Class Role*

Role	<<entity>>
-id_role : integer(11) Atribut ini digunakan untuk menyimpan data id role.	
-nama_role : varchar(255) Atribut ini digunakan untuk menyimpan data nama role.	
-deskripsi_role : varchar(255) Atribut ini digunakan untuk menyimpan deskripsi role.	
+getrole() Operasi untuk mengambil data role.	

2.2.3.20. *Specific Design Class Admin*

Admin	<<entity>>
-id_admin : integer(11) Atribut ini digunakan untuk menyimpan data id admin.	
-email_admin : varchar(255)	

Atribut ini digunakan untuk menyimpan data email admin.

-password_admin : varchar(255)

Atribut ini digunakan untuk menyimpan data password admin.

-nama_admin : varchar(255)

Atribut ini digunakan untuk menyimpan data nama admin.

-telepon_admin : integer(11)

Atribut ini digunakan untuk menyimpan data telepon admin.

-company_admin : varchar(255)

Atribut ini digunakan untuk menyimpan data company admin.

-status_admin : varchar(255)

Atribut ini digunakan untuk menyimpan data status admin.

+getdataadmin()

Operasi untuk mengambil data admin.

2.2.3.21. *Specific Design Class* PemilikKendaraan

PemilikKendaraan	<<entity>>
------------------	------------

-id_pemilikkendaraan : integer(11)

Atribut ini digunakan untuk menyimpan data id pemilik kendaraan.

-nama_pemilikkendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data nama pemilik kendaraan.

-nik_pemilikkendaraan : integer(11)

Atribut ini digunakan untuk menyimpan data nik pemilik kendaraan.

-kelamin_pemilikkendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data kelamin pemilik kendaraan.

-alamat_pemilikkendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data alamat pemilik kendaraan.

-telepon_pemilikkendaraan : integer(11)

Atribut ini digunakan untuk menyimpan data telepon pemilik kendaraan.

-email_pemilikkendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data email pemilik kendaraan.

-status_pemilikkendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data status pemilik kendaraan.

-adminblp: integer(11)

Atribut ini digunakan untuk menyimpan data adminblp pemilik kendaraan.

+getdatapemilikkendaraan()

Operasi untuk mengambil data pemilik kendaraan.

2.2.3.22. *Specific Design Class* Kendaraan

Kendaraan

<<entity>>

-id_kendaraan : integer(11)

Atribut ini digunakan untuk menyimpan data id kendaraan.

-pemilikkendaraan : integer(11)

Atribut ini digunakan untuk menyimpan data pemilikkendaraan kendaraan.

-sifat_kendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data sifat kendaraan.

-nopol_kendaraan : integer(11)

Atribut ini digunakan untuk menyimpan data nopol kendaraan.

-tahun_kendaraan : integer(11)

Atribut ini digunakan untuk menyimpan data tahun kendaraan.

-nomesin_kendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data nomesin kendaraan.

-nochassis_kendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data nochassis kendaraan.

-bahanbakar_kendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data bahanbakar kendaraan.

-isicylinder_kendaraan : integer(11)

Atribut ini digunakan untuk menyimpan data isicylinder kendaraan.

-merk_kendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data merk kendaraan.

-tipe_kendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data tipe kendaraan.

-jenis_kendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data jenis kendaraan.

-bentuk_kendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data bentuk kendaraan.

-tempatduduk_kendaraan : integer(11)

Atribut ini digunakan untuk menyimpan data tempatduduk kendaraan.

-tempatberdiri_kendaraan : integer(11)

Atribut ini digunakan untuk menyimpan data tempatberdiri kendaraan.

-status_kendaraan : varchar(255)

Atribut ini digunakan untuk menyimpan data status kendaraan.

-admin : integer(11)

Atribut ini digunakan untuk menyimpan data admin kendaraan.

+getdata kendaraan ()

Operasi untuk mengambil data kendaraan.

2.2.3.23. *Specific Design Class DetilUji*

DetilUji	<<entity>>
----------	------------

-id_detiluji : integer(11)

Atribut ini digunakan untuk menyimpan data id detiluji.

-emisibuang: varchar(255)

Atribut ini digunakan untuk menyimpan data emisibuang detiluji.

-ketebalanasap: varchar(255)

Atribut ini digunakan untuk menyimpan data ketebalanasap detiluji.

-kebisinganklakson: varchar(255)

Atribut ini digunakan untuk menyimpan data kebisinganklakson detiluji.

-kebisinganknalpot: varchar(255)

Atribut ini digunakan untuk menyimpan data kebisinganknalpot detiluji.

-remutama: varchar(255)

Atribut ini digunakan untuk menyimpan data remutama detiluji.

-remparkir: varchar(255)

Atribut ini digunakan untuk menyimpan data remparkir detiluji.

-kincuprodadepan: varchar(255)

<p>Atribut ini digunakan untuk menyimpan data kincuprodadepan detiluji.</p> <p>-kemampuanpancar: varchar(255)</p> <p>Atribut ini digunakan untuk menyimpan data kemampuanpancar detiluji.</p> <p>-arahsinarlampu: varchar(255)</p> <p>Atribut ini digunakan untuk menyimpan data arahsinarlampu detiluji.</p> <p>-akurasipenunjukkecepatan: varchar(255)</p> <p>Atribut ini digunakan untuk menyimpan data akurasipenunjukkecepatan detiluji.</p> <p>-kedalamanalurban: varchar(255)</p> <p>Atribut ini digunakan untuk menyimpan data kedalamanalurban detiluji.</p> <p>-dayatembuscahaya: varchar(255)</p> <p>Atribut ini digunakan untuk menyimpan data dayatembuscahaya detiluji.</p>
<p>+getdatadetiluji()</p> <p>Operasi untuk mengambil data detiluji.</p>

2.2.3.24. Specific Design Class Pengujian

Pengujian	<<entity>>
-----------	------------

-id_pengujian: integer(11)

Atribut ini digunakan untuk menyimpan data id pengujian.

-kendaraan: integer(11)

Atribut ini digunakan untuk menyimpan data kendaraan pengujian.

-tipeuji: varchar(255)

Atribut ini digunakan untuk menyimpan data tipeuji pengujian.

-lokasi_pengujian: varchar(255)

Atribut ini digunakan untuk menyimpan data lokasi pengujian.

-tanggaluji_pengujian: date

Atribut ini digunakan untuk menyimpan data tanggaluji pengujian.

-tanggalujiberikutnya_pengujian: date

Atribut ini digunakan untuk menyimpan data tanggalujiberikutnya pengujian.

-biayapendaftaran: integer(11)

Atribut ini digunakan untuk menyimpan data biayapendaftaran pengujian.

-biyacattandaketok: integer(11)

Atribut ini digunakan untuk menyimpan data biayacattandaketok pengujian.

-biayatandauji: integer(11)

Atribut ini digunakan untuk menyimpan data biayatandauji pengujian.

-detilhasiluji: integer(11)

Atribut ini digunakan untuk menyimpan data detilhasiluji pengujian.

-status_pengujian: varchar(255)

Atribut ini digunakan untuk menyimpan data status pengujian.

-admin: integer(11)

Atribut ini digunakan untuk menyimpan data admin pengujian.

+getdata pengujian ()

Operasi untuk mengambil data pengujian.

2.2.3.25. Specific Design Class TipeUji

TipeUji	<<entity>>
---------	------------

<pre>-id_tipeuji : integer(11)</pre> <p>Atribut ini digunakan untuk menyimpan data id tipeuji.</p> <pre>-nama_ tipeuji: varchar(255)</pre> <p>Atribut ini digunakan untuk menyimpan data nama tipeuji.</p> <pre>-biaya_ tipeuji: integer(11)</pre> <p>Atribut ini digunakan untuk menyimpan data biaya tipeuji.</p>
<pre>+getdatatipeuji()</pre> <p>Operasi untuk mengambil data tipeuji.</p>

2.2.3.26. *Specific Design Class BiayaUji*

BiayaUji	<<entity>>
<pre>-id_biayauji : integer(11)</pre> <p>Atribut ini digunakan untuk menyimpan data id biayauji.</p> <pre>-nama_ biayauji : varchar(255)</pre> <p>Atribut ini digunakan untuk menyimpan data id biayauji.</p> <pre>-nominal_ biayauji : integer(11)</pre> <p>Atribut ini digunakan untuk menyimpan data id biayauji.</p>	
<pre>+getdatabiayauji ()</pre> <p>Operasi untuk mengambil data biayauji.</p>	

3. Perancangan Data

3.1 Dekomposisi Data

3.1.1. Deskripsi Entitas Role			
Nama	Tipe	Panjang	Keterangan
Id_role	Integer	11	Id role, primary key
Nama_role	Variable Character	255	Nama role
Deskripsi_role	Variable Character	255	Deskripsi role

3.1.2. Deskripsi Entitas Admin			
Nama	Tipe	Panjang	Keterangan
Id_admin	Integer	11	Id admin, primary key
Email_admin	Variable Character	255	Email admin
Password_admin	Variable Character	255	Password admin
Nama_admin	Variable Character	255	Nama admin
Telepon_admin	Integer	11	Telepon admin
Company_admin	Variable Character	255	Company admin
Status_admin	Variable Character	255	Status admin

3.1.3. Deskripsi Entitas PemilikKendaraan			
Nama	Tipe	Panjang	Keterangan
Id_pemilikkendaraan	Integer	11	Id pemilik kendaraan, primary key
Nama_pemilikkendaraan	Variable Character	255	Nama pemilik kendaraan
Nik_pemilikkendaraan	Integer	11	NIK pemilik kendaraan
Kelamin_pemilikkendaraan	Variable Character	255	Kelamin pemilik kendaraan
Alamat_pemilikkendaraan	Variable Character	255	Alamat pemilik kendaraan
Telepon_pemilikkendaraan	Integer	11	Telepon pemilik kendaraan
Email_pemilikkendaraan	Variable Character	255	Email pemilik kendaraan
Status_pemilikkendaraan	Variable Character	255	Status pemilik kendaraan

Admin	Integer	11	Admin yang bertugas, foreign key
-------	---------	----	----------------------------------

3.1.4. Deskripsi Entitas Kendaraan			
Nama	Tipe	Panjang	Keterangan
Id_kendaraan	Integer	11	Id kendaraan, primary key
Pemilikkendaraan	Integer	11	Pemilik kendaraan, foreign key
Sifat_kendaraan	Variable character	255	Sifat kendaraan
Nopol_kendaraan	Integer	11	Nopol kendaraan
Tahun_kendaraan	Integer	11	Tahun kendaraan
Nomesin_kendaraan	Variable character	255	Nomesin kendaraan

Nochassis_kendaraan	Variable character	255	Nochassis kendaraan
Bahanbakar_kendaraan	Variable character	255	Bahan bakar kendaraan
Isicylinder_kendaraan	Integer	11	Isi cylinder kendaraan
Merk_kendaraan	Variable character	255	Merk kendaraan
Tipe_kendaraan	Variable character	255	Tipe kendaraan
Jenis_kendaraan	Variable character	255	Jenis kendaraan
Bentuk_kendaraan	Variable character	255	Bentuk kendaraan
Tempatduduk_kendaraan	Integer	11	Tempat duduk kendaraan
Tempatberdiri_kendaraan	Integer	11	Tempat berdiri kendaraan
Status_kendaraan	Variable character	255	Status kendaraan
Admin	Integer	11	Admin yang bertugas

3.1.5. Deskripsi Eentitas Pengujian

Nama	Tipe	Panjang	Keterangan
Id_pengujian	Integer	11	Id pengujian
Kendaraan	Integer	11	Kendaraan yang di uji, foreign key
Tipeuji	Integer	11	Tipe uji, foreign key
Lokasi_pengujian	Variable Character	255	Lokasi pengujian
Tanggaluji_pengujian	Variable Character	255	Tanggal uji pengujian
Tanggalujiberikutnya_pengujian	Variable Character	255	Tanggal uji berikutnya
Biayapendaftaran	Integer	11	Biaya pendaftaran uji
Biayacattandaketok	Integer	11	Biaya cat tanda ketok

Biayatandauji	Integer	11	Biaya tanda uji
Detilhasiluji	Integer	11	Detil hasil pengujian , foreign key
Status_pengujian	Variable Character	255	Status pengujian
Admin	Integer	11	Admin yang bertugas

3.1.6. Deskripsi Entitas DetilUji			
Nama	Tipe	Panjang	Keterangan
Id	Integer	11	Id_detiluji
Emisigas	Variable Character	255	Tingkat emisi gas
Ketebalanasap	Variable Character	255	Tingkat ketebalan asap
Kebisinganklakson	Variable Character	255	Tingkat kebisingan klakson
Kebisinganknalpot	Variable Character	255	Tingkat kebisingan knalpot

Remutama	Variable Character	255	Tingkat kemampuan rem utama
Remparkir	Variable Character	255	Tingkat kemampuan rem parker
Kincuprodadepan	Variable Character	255	Jarak kincup depan
Kemampuanpancar	Variable Character	255	Tingkat kemampuan pancar
Arahsinarlampu	Variable Character	255	Arah sinar lampu utama
Akurasipenunjukkecepatan	Variable Character	255	Tingkat akurasi penunjuk kecepatan
Kedalamanalurban	Variable Character	255	Tingkat kedalaman alur ban
Dayatembuscahaya	Variable Character	255	Tingkat daya tembus cahaya pada kaca
Foto_detiluji	Variable Character	255	Foto pengujian

Status_detiluji	Variable Character	255	Status detil uii
Admin	Integer	11	Admin yang sedang bertugas, foreign key

3.1.7. Deskripsi Entitas TipeUji

Nama	Tipe	Panjang	Keterangan
Id_tipeuji	Integer	11	Id tipe uji, primary key
Nama_tipeuji	Variable Character	255	Nama tipe uji
Biaya_tipeuji	Integer	11	Biaya pendaftaran uji

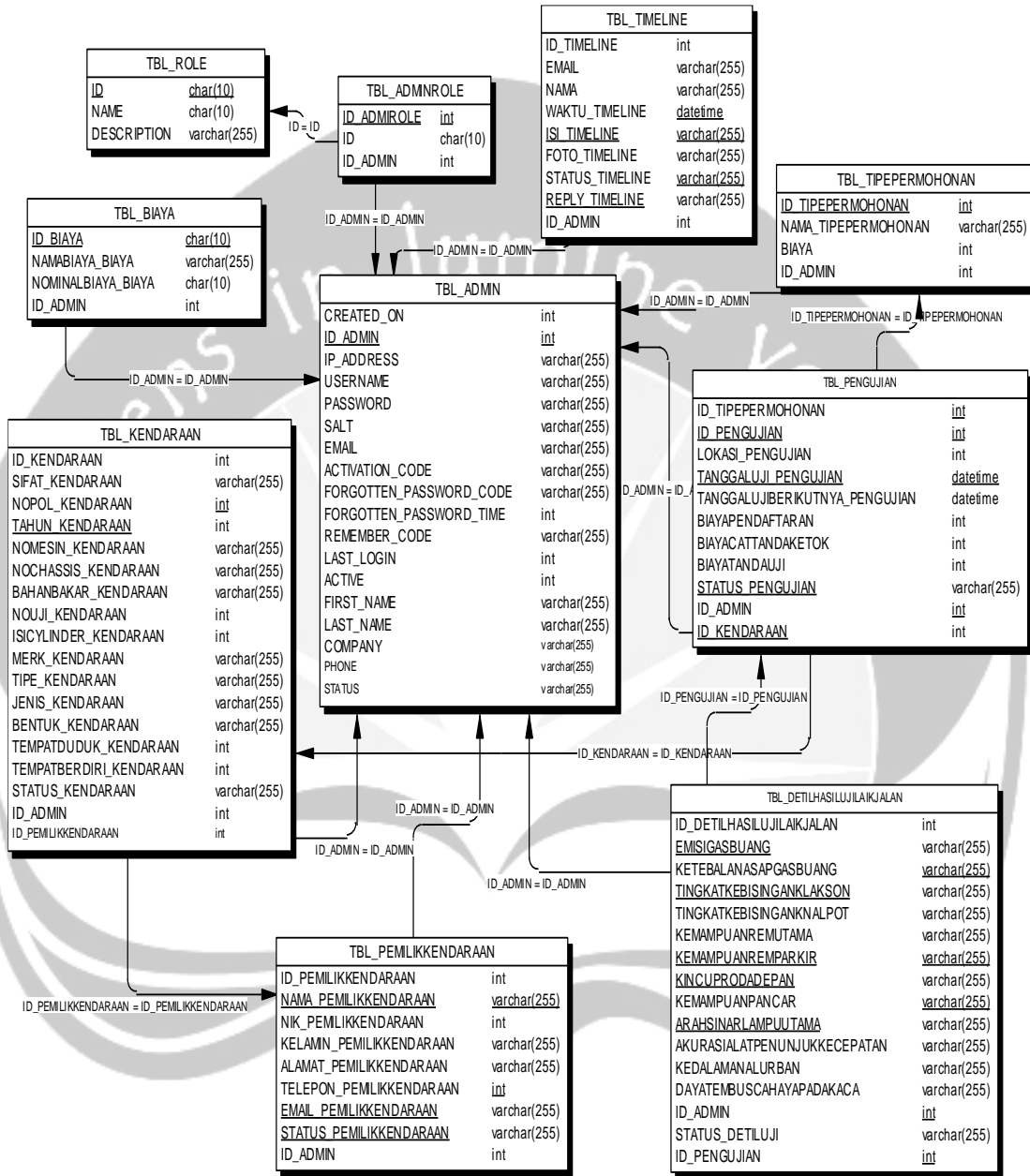
3.1.8. Deskripsi Entitas BiayaUji

Nama	Tipe	Panjang	Keterangan
Id_biaya	Integer	11	Id biaya uji
Nama_biaya	Variable Character	255	Nama-nama biaya dalam pengujian
Nominal_biaya	Integer	11	Nominal biaya pengujian

3.1.9. Deskripsi Entitas Aduan			
Nama	Tipe	Panjang	Keterangan
Id_aduan	Integer	11	Id aduan, primary key
Email	Variable Character	255	Email masyarakat
Nama	Variable Character	255	Nama masyarakat
Waktu	Timespan		Waktu ketika aduan dikirimkan
Isi_aduan	Variable Character	255	Isi aduan
Foto_aduan	Variable Character	255	Foto lampiran dalam aduan
Status_aduan	Variable Character	255	Status aduan
Balasan	Variable Character	255	Balasan terhadap aduan masyarakat
Admin	Integer	11	Admin yang sedang bertugas, foreign key

3.2 Physical Data Model





Gambar 3.1 Physical Data Model

4. Deskripsi Perancangan Antarmuka

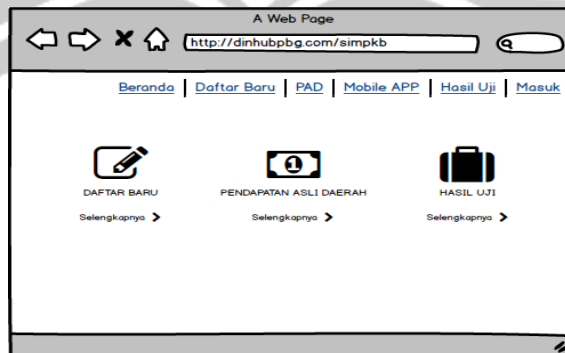
4.1 Antarmuka Halaman Beranda



Gambar 4.1 Antarmuka Beranda

Antarmuka 4.1 digunakan untuk menampilkan halaman awal layanan pengujian kendaraan bermotor yang disediakan oleh Dinas Perhubungan Kabupaten Purbalingga untuk masyarakat. Pengguna dapat melakukan daftar baru kendaraan uji, menampilkan PAD, download aplikasi *mobile* SIPPKB, melihat hasil uji kendaraan.

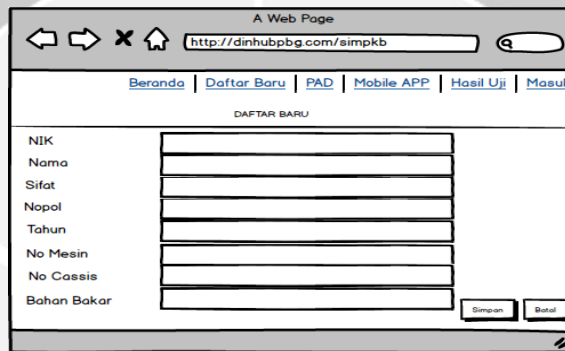
4.2 Antarmuka Halaman Layanan



Gambar 4.2 Antarmuka Layanan

Antarmuka 4.2 digunakan untuk menampilkan halaman layanan pengujian kendaraan bermotor yang disediakan oleh Dinas Perhubungan Kabupaten Purbalingga untuk masyarakat. Pengguna dapat melihat layanan yang tersedia, seperti daftar baru kendaraan uji, melihat pendapatan asli daerah, melihat hasil uji kendaraan.

4.3 Antarmuka Halaman Daftar Baru



The image shows a web browser window with the address bar displaying 'http://dinhubpbg.com/simpkb'. The page has a navigation menu with links: Beranda, Daftar Baru, PAD, Mobile APP, Hasil Uji, and Masuk. Below the menu is a form titled 'DAFTAR BARU' with the following fields:

NIK	<input type="text"/>
Nama	<input type="text"/>
Sifat	<input type="text"/>
Nopol	<input type="text"/>
Tahun	<input type="text"/>
No Mesin	<input type="text"/>
No Cassis	<input type="text"/>
Bahan Bakar	<input type="text"/>

At the bottom right of the form are two buttons: 'Simpan' and 'Batal'.

Gambar 4.3 Antarmuka Halaman Daftar Baru

Antarmuka 4.3 digunakan untuk menampilkan halaman pendaftaran uji kendaraan bermotor disediakan oleh Dinas Perhubungan Kabupaten Purbalingga untuk masyarakat. Pengguna dapat melihat melakukan pendaftaran dengan mengisi formulir yang sudah disediakan.

4.4 Antarmuka Halaman PAD

Kecamatan	Retribusi
Bobotsari	120.000
Purbalingga	150.000
Kalimanah	230.000

Gambar 4.4 Antarmuka Halaman PAD

Antarmuka 4.4 digunakan untuk menampilkan halaman PAD yang disediakan oleh Dinas Perhubungan Kabupaten Purbalingga untuk masyarakat. Pengguna dapat melihat data pendapatan asli daerah di Kabupaten Purbalingga, pengguna juga dapat melakukan pencarian PAD per bulan melalui filter yang ada.

4.5 Antarmuka Halaman Hasil Uji Kendaraan

Nopol	Tahun	No Mesin	No Cassis	Merk	Tipe	Status
B1212	2008	83923	0982	Toyota	G7	Verified
G786	2011	981832	2122	Daihatsu	87	Verified
B1212	2008	83923	0982	Toyota	G7	Verified
R6712	2014	89137	2878	Mitsubishi	K8	Not Verified

Gambar 4.5 Antarmuka Halaman Hasil Uji Kendaraan

Antarmuka 4.5 digunakan untuk menampilkan halaman hasil uji kendaraan yang disediakan oleh Dinas Perhubungan Kabupaten Purbalingga untuk masyarakat. Pengguna dapat melihat data hasil uji kendaraan yang ada.

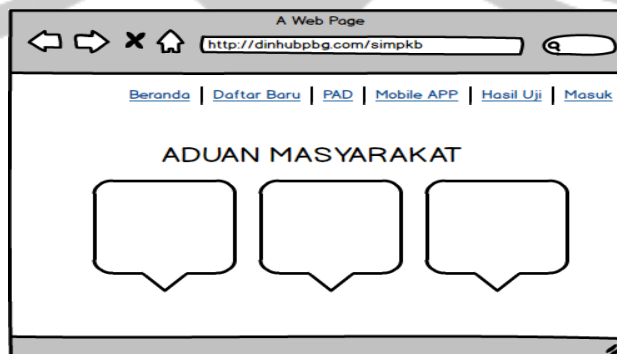
4.6 Antarmuka Halaman *Mobile* Aplikasi



Gambar 4.6 Antarmuka Aplikasi *Mobile*

Antarmuka 4.6 digunakan untuk menampilkan halaman aplikasi SIPPKB *mobile*. Pengguna dapat melihat informasi terkait SIPPKB *mobile* serta mengunduh aplikasi SIPPKB *mobile*.

4.7 Antarmuka Halaman Aduan Masyarakat



Gambar 4.7 Antarmuka Halaman Aduan Masyarakat

Antarmuka 4.7 digunakan untuk menampilkan halaman aduan masyarakat. Pengguna dapat melihat aduan masyarakat terkait layanan pengujian kendaraan bermotor di Kabupaten Purbalingga serta pengguna dapat menuliskan aduan masyarakat yang dimiliki.

4.8 Antarmuka Halaman Login Admin

Gambar 4.8 Antarmuka Halaman Login Admin

Antarmuka 4.8 digunakan untuk login, data yang dimasukkan adalah data email atau username dan password pengguna.

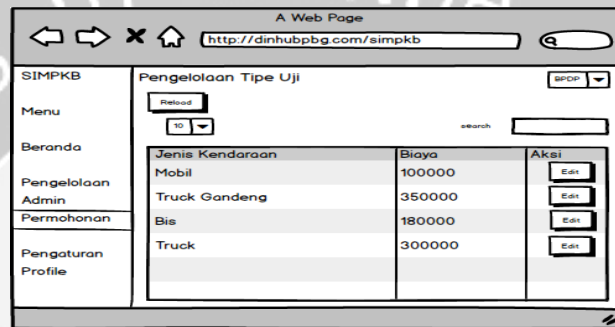
4.9 Antarmuka Halaman Pengelolaan Admin

Nama Lengkap	Email	Telp	Role	Status
Sukma Stevia	SC@gmail.com	123	BLP	Aktif
Elika Thea	TH@gmail.com	123	BLP	Aktif
Juming	J@gmail.com	123	BPDP	Aktif
Andreas Dimas	D@gmail.com	123	BTP	Aktif

Gambar 4.9 Antarmuka Halaman Pengelolaan Admin

Antarmuka 4.9 digunakan untuk menampilkan halaman Pengelolaan admin. BPDP selaku aktor dapat melihat data admin, melakukan pencarian admin yang ada serta menambahkan admin baru.

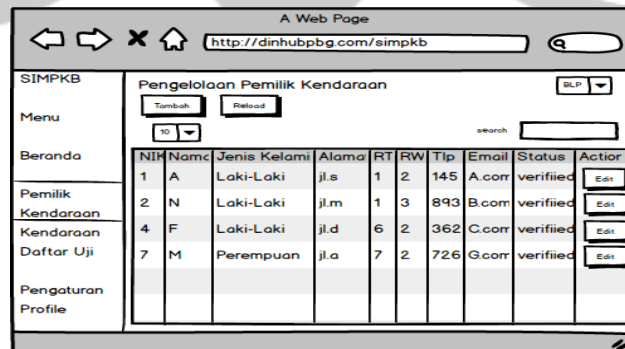
4.10 Antarmuka Halaman Pengelolaan Tipe Permohonan



Gambar 4.10 Antarmuka Halaman Pengelolaan Tipe Permohonan

Antarmuka 4.10 digunakan untuk menampilkan halaman Pengelolaan tipe permohonan kendaraan. BPDP selaku aktor dapat melihat data jenis kendaraan dan biaya untuk permohonan uji kendaraan bermotor serta melakukan perubahan biaya uji yang ada.

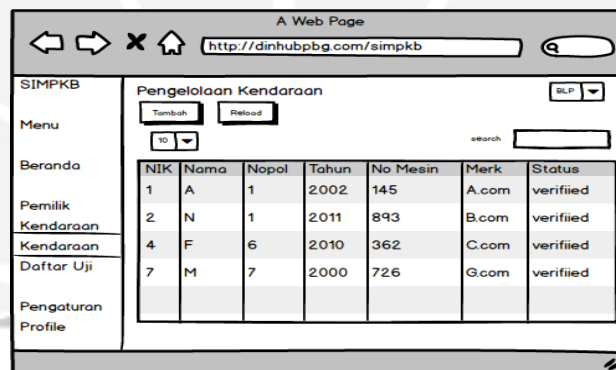
4.11 Antarmuka Halaman Pengelolaan Pemilik Kendaraan



Gambar 4.11 Antarmuka Halaman Pengelolaan Pemilik Kendaraan

Antarmuka 4.11 digunakan untuk menampilkan halaman Pengelolaan pemilik kendaraan. BLP selaku aktor dapat melihat data pemilik kendaraan, melakukan pendaftaran atau penambahan calon pemilik kendaraan, melihat data pemilik kendaraan serta melakukan pencarian serta melakukan perubahan data data pemilik kendaraan yang ada.

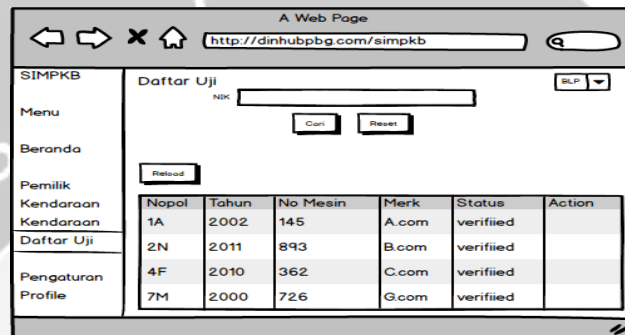
4.12 Antarmuka Halaman Pengelolaan Kendaraan



Gambar 4.12 Antarmuka Halaman Pengelolaan Kendaraan

Antarmuka 4.12 digunakan untuk menampilkan halaman Pengelolaan kendaraan. BLP selaku aktor dapat melakukan penambahan atau mendaftarkan calon kendaraan uji, melihat data kendaraan uji, melakukan pencarian data kendaraan uji yang ada.

4.13 Antarmuka Halaman Pengelolaan Daftar Uji



Gambar 4.13 Antarmuka Halaman Pengelolaan Daftar Uji

Antarmuka 4.13 digunakan untuk menampilkan halaman Pengelolaan daftar uji. BLP selaku aktor dapat melakukan verifikasi terhadap calon daftar uji yang belum terverifikasi dengan mencari NIK calon daftar uji yang sudah mendaftar, melakukan perubahan data daftar uji, melihat data daftar uji yang ada.

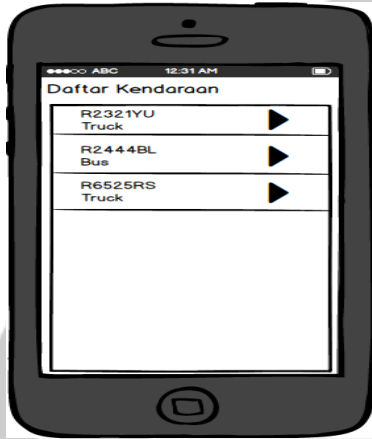
4.14 Antarmuka Halaman Login



Antarmuka 4.14 digunakan untuk login, data yang dimasukkan adalah data *username* dan *password* BTP.

Gambar 4.14 Antarmuka Halaman *Login*

4.15 Antarmuka Halaman Pengelolaan Kendaraan



Antarmuka 4.15 digunakan untuk menampilkan halaman Pengelolaan lokasi. BTP selaku aktor dapat melihat data daftar kendaraan yang tersedia yang akan diverifikasi.

Gambar 4.15 Antarmuka Halaman Pengelolaan Kendaraan

4.16 Antarmuka Halaman Pengelolaan Verifikasi Kendaraan



Antarmuka 4.16 digunakan untuk menampilkan halaman Pengelolaan verifikasi kendaraan. BTP selaku aktor dapat melakukan proses verifikasi kendaraan yang sudah dipilih dari halaman sebelumnya kemudian mengisi data yang telah disediakan. Kemudian melakukan konfirmasi data yang telah

diisi.

Gambar 4.16 Antarmuka Halaman Pengelolaan Verifikasi Kendaraan

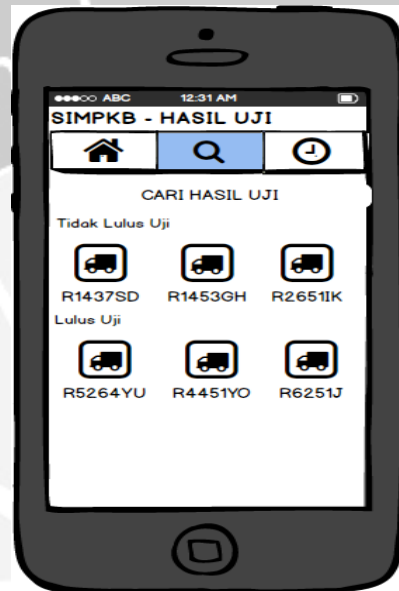
4.17 Antarmuka Halaman *Mobile*



Gambar 4.17 Antarmuka Halaman *Mobile*

Antarmuka 4.17 digunakan untuk menampilkan halaman utama *mobile* SIPPKB. Pengguna dapat melihat layanan yang disediakan Dinas Perhubungan Kabupaten Purbalingga. Pada halaman utama ini pengguna dapat melihat informasi seputar uji kendaraan bermotor, kendaraan wajib uji, syarat untuk mengajukan uji kendaraan, serta detil pengujian pada kendaraan yang akan dilakukan pengujian. Pengguna juga dapat melihat jumlah total kendaraan uji, tidak lulus dan lulus uji.

4.18 Antarmuka Halaman Hasil Uji



Gambar 4.18 Antarmuka Halaman Hasil Uji

Antarmuka 4.18 digunakan untuk menampilkan halaman hasil uji SIPPKB. Pengguna (pemilik kendaraan uji dan masyarakat umum) dapat melihat layanan yang disediakan Dinas Perhubungan Kabupaten Purbalingga mengenai hasil uji kendaraan. Pada hasil uji ini pengguna (pemilik kendaraan) dapat melihat detail informasinya termasuk biaya uji kendaraan, dengan menekan tombol cari hasil uji kemudian memasukkan no uji kendaraan. Masyarakat dapat melihat detail kendaraan uji yang tidak lulus maupun lulus uji yang tersedia.

4.19 Antarmuka Halaman *Timeline*



Gambar 4.19 Antarmuka Halaman *Timeline*

Antarmuka 4.19 digunakan untuk menampilkan halaman *timeline* SIPPKB Pengguna (masyarakat umum) dapat melihat layanan yang disediakan Dinas Perhubungan Kabupaten Purbalingga mengenai aduan masyarakat. Pengguna dapat melihat aduan masyarakat yang tersedia serta dapat menambahkan aduan.