

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 parkir pada Kabupaten Purbalingga dengan nama SIMPARKIR telah berhasil dibangun dan dapat mewujudkan keterbukaan informasi secara optimal terbukti dari hasil pengujian *black box testing*.

6.2. Saran

Beberapa saran dan masukan yang didapatkan penulis terhadap perancangan sistem informasi pengelolaan parkir pada Kabupaten Purbalingga meliputi:

Pengembangan dapat dilakukan pada *platform* lain selain Android.

DAFTAR PUSTAKA

- Ableson. (2009). *Sistem Operasi*. Retrieved 11 29, 2016, from <http://library.binus.ac.id/eColls/eThesisdoc/Bab2HTML/2013100969SIBab2001/page24.html>
- Ana Nur Cahyanti, B. E. (2012). Pembangunan Sistem Informasi Manajemen Puskesmas Pakis Baru Nawangan. *Journal Speed, Sentra Penelitian Engineering dan Edukasi*.
- Badrul Anwar, H. J. (2014). IMPLEMENTASI LOCATION BASED SERVICE BERBASIS ANDROID UNTUK MENGETAHUI POSISI USER. *Jurnal Ilmiah SAINTIKOM*, 2.
- Balachandar, R. A., Aye, W., Avandi, M., Anitha, R., & Kumar, S. A. (2007). In *Web Services Security and E-Business* (p. 166).
- Fauzi, K. (2013). SISTEM INFORMASI DATA KARYAWAN PADA PT. POS INDONESIA (PERSERO) BANDA ACEH BERBASIS WEB. *Jurnal Sarjana Sistem Informasi*.
- Faya Mahdia, F. N. (2013). PEMANFAATAN GOOGLE MAPS API UNTUK PEMBANGUNAN SISTEM INFORMASI MANAJEMEN BANTUAN LOGISTIK PASCA BENCANA ALAM BERBASIS MOBILE WEB. *Jurnal Teknik Informatika*.

- Ibnu Daqiqil Id, M. (2011). *Framework CodeIgniter: sebuah panduan dan best practice*. Pekanbaru.
- Jogiyanto. (2005). *Analisis & Desaaain Sistem Informasi*. Andi Offset, Yogyakarta.
- Kenneth C. Laudon, J. P. (2006). *Management Information Systems*. New Jersey: Pearson.
- Koesuma, A. (2011). Perancangan Sistem Manajemen Basis Data sebagai Penunjang Sistem Informasi. *Thesis S1, UAJY*.
- Ramadhan, D. R. (2015). PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI MOBILE SCHOOL MAPS (MooMaps) BERBASIS MOBILE APPLICATION UNTUK PEMETAAN UNIVERSITAS DI YOGYAKARTA. *Jurnal Sarjana Teknik Informatika*, 3.
- Saputra, A. (2015). Sistem Informasi Persediaan Obat pada APotek Nayla. *Jurnal Sistem Informasi Universitas Bina Darma*.
- Stefan Steiniger, M. N. (2006). *Foundations of Location Based Services*. Retrieved from School of Information Science, University of Zurich, Winterthurerstrasse: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.1844&rep=rep1&type=pdf>
- Sugara, E. P. (2011). SISTEM INFORMASI PENCARIAN DAN PENJUALAN BARANG BERBASIS WEB PADA TOKO BAGUS. *JURNAL TEKNOLOGI DAN INFORMATIKA*, 159.
- Utomo, A. P. (2013). Analisa dan Perancangan Sistem Informasi Parkir di Universitas Muria Kudus. *Jurnal Sistem Informasi*, 1, 18.

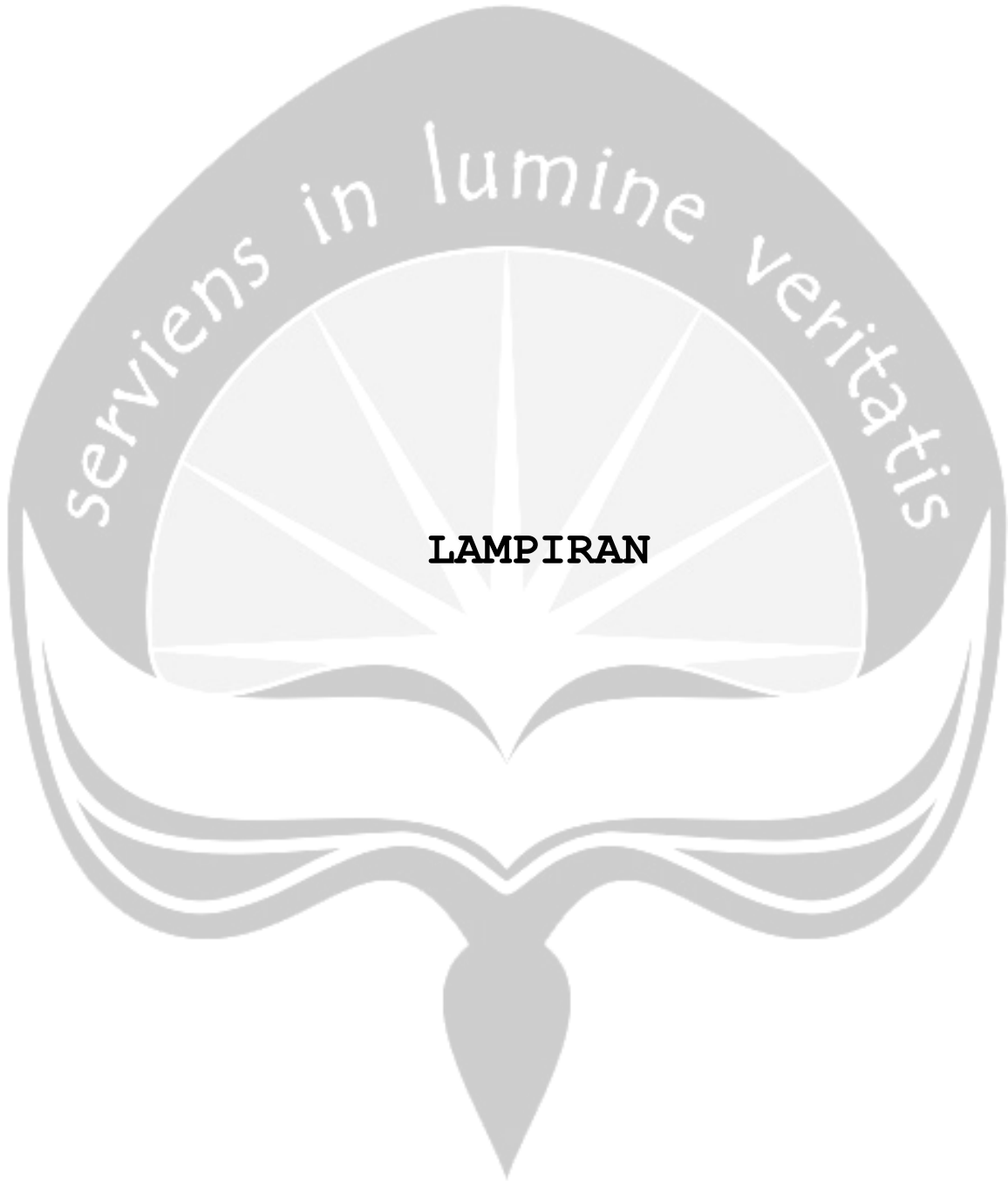
Vebri, V. Y. (2016). Perancangan Sistem Informasi Persediaan di Apotek X. *Thesis S1, UAJY*.

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

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

**PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN
PARKIR BERBASIS WEB DAN MOBILE**

(STUDI KASUS DINAS PERHUBUNGAN KABUPATEN PURBALINGGA)

Untuk:

Universitas Atma Jaya Yogyakarta


Dipersiapkan oleh:

Sukma Stevia Charolline / 130707335

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik	Nomor Dokumen		Halaman
	Informatika	SKPL-SIMPARKIR		1/40
	Fakultas Teknologi Industri	Revisi	A	

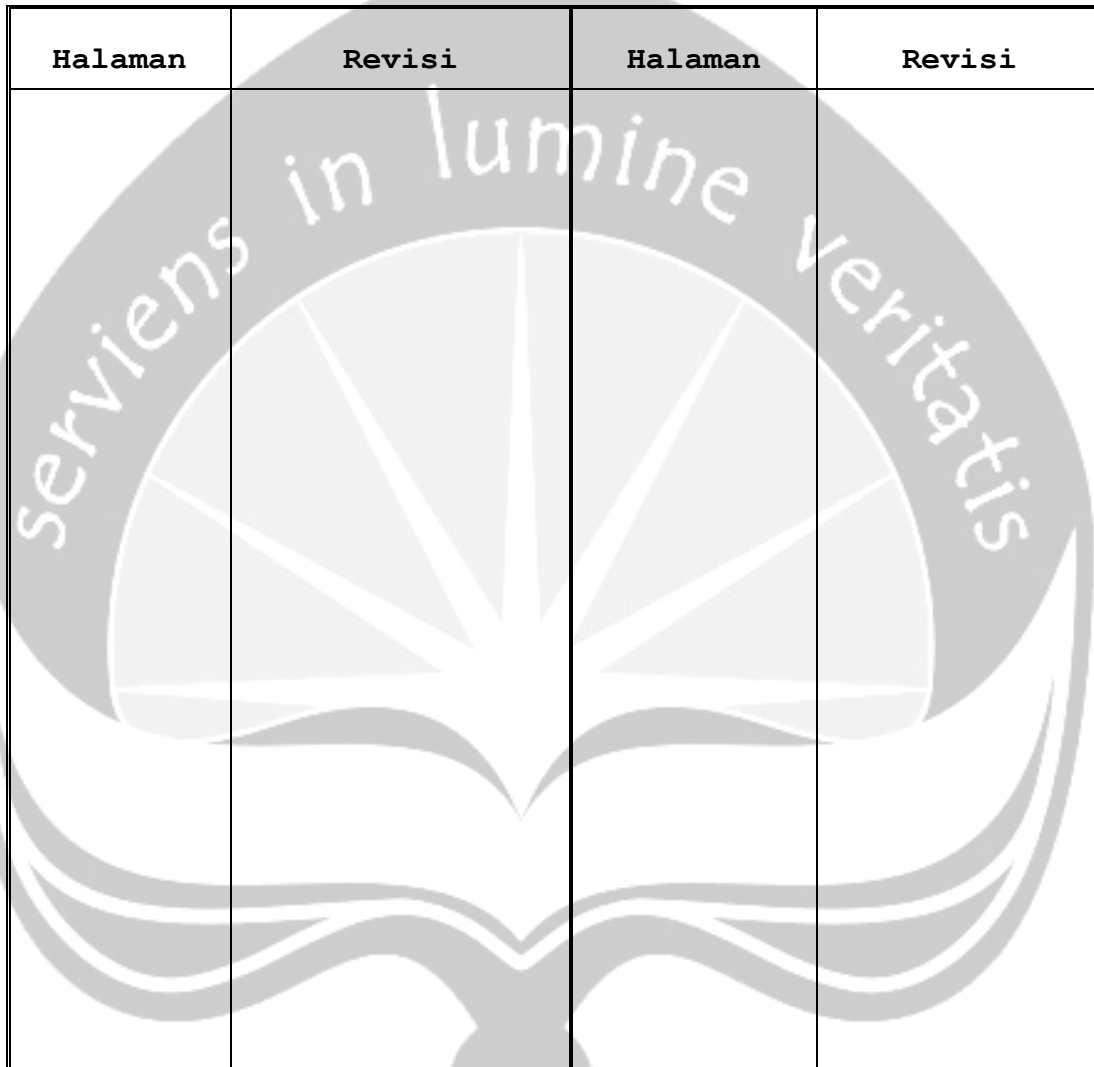
DAFTAR PERUBAHAN

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.....	6
1.1 Tujuan	6
1.2 Lingkup Masalah	6
1.3 Definisi, Akronim dan Singkatan	7
1.4 Referensi	8
1.5 Deskripsi Umum	8
2. Deskripsi Kebutuhan.....	9
2.1 Prespektif Produk	9
2.2 Fungsi Produk	11
2.3 Karakteristik Pengguna	19
2.4 Batasan-batasan	20
2.5 Asumsi dan Ketergantungan	20
3. Kebutuhan Khusus.....	20
3.1 Kebutuhan Antarmuka Eksternal	20
3.2 Kebutuhan Fungsionalitas Perangkat Lunak	22
3.2.1 <i>Use Case Diagram</i> Aplikasi <i>Mobile</i>	22
3.2.2 <i>Use Case Diagram</i> Aplikasi <i>Web</i>	23
4. Spesifikasi Rinci Kebutuhan.....	24
4.1 Spesifikasi Kebutuhan Fungsionalitas	24
4.1.1. Pengelolaan Kategori Pengguna <i>Web</i>	24
4.1.2. Pengelolaan Biaya Parkir <i>Web</i>	25
4.1.3. Pengelolaan Lokasi Parkir <i>Web</i>	27
4.1.4. Pengelolaan Juru Parkir <i>Web</i>	29

4.1.5. Pengelolaan Penugasan Juru Parkir <i>Web</i>	31
4.1.6. Pengelolaan Lokasi Parkir Baru <i>Mobile</i>	34
4.1.7. Pengelolaan Juru Parkir <i>Mobile</i>	35
4.1.8. Pengelolaan Aduan Masyarakat <i>Mobile</i>	36
4.4.9. Pengelolaan Retribusi Parkir <i>Mobile</i>	39
5. <i>Entity Relationship Diagram</i>	40

DAFTAR GAMBAR

Gambar 2.1 Rancangan Arsitektur Perangkat Lunak SIMPARKIR ...	10
Gambar 3.1 <i>Use Case Diagram</i> Aplikasi <i>Mobile</i> SIMPARKIR	22
Gambar 3.2 <i>Use Case Diagram</i> Aplikasi <i>Web</i> SIMPARKIR	23
Gambar 5.1 Perancangan ERD SIMPARKIR	40

1. Pendahuluan

1.1. Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SIMPARKIR (Sistem Informasi Pengelolaan Parkir) 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-SIMPARKIR ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2. Lingkup Masalah

Perangkat Lunak SIMPARKIR dikembangkan dengan tujuan untuk:

1. Pengelolaan pengguna sistem, pengelolaan data admin, pengelolaan biaya parkir, pengelolaan lokasi parkir, pengelolaan juru parkir.
2. Proses transaksi biaya parkir mencakup setoran dan pembayaran sesuai target beserta penentuan denda keterlambatan.
3. Proses penugasan juru parkir mencakup penentuan lokasi tugas, jam tugas, setoran tugas.

4. Proses pendataan asli daerah PAD parkir kabupaten purbalingga.

Aplikasi ini berjalan pada *smartphone* berbasis Android dan *web*.

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-SIMPARKIR-XXX	Kode yang merepresentasikan kebutuhan pada SIMPARKIR (Sistem Informasi Manajemen Parkir) di mana XXX merupakan nomor fungsi produk.
SIMPARKIR	Perangkat lunak pengelolaan <i>web</i> dan <i>mobile</i> .
<i>Database</i>	Kumpulan data yang tersimpan dan diorganisasikan agar dapat diambil dengan mudah secara efisien.
<i>Internet</i>	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> , <i>FTP</i> , dan <i>World Wide Web</i> .
<i>Server</i>	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Android	Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet.
Pengelolaan Parkir	Pengelolaan Parkir merupakan salah satu jenis usaha yang memberikan layanan dalam bentuk jasa pengelolaan dan pelayanan tempat untuk memarkir kendaraan para konsumen.
BPDP	Badan Pengelola Data Parkir
BTP	Badan Teknis Parkir
BLP	Badan Layanan Parkir

1.4. Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Sukma Stevia Charolline / 130707335, 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 SIMPARKIR 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 SIMPARKIR tersebut.

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

2. Deskripsi Kebutuhan

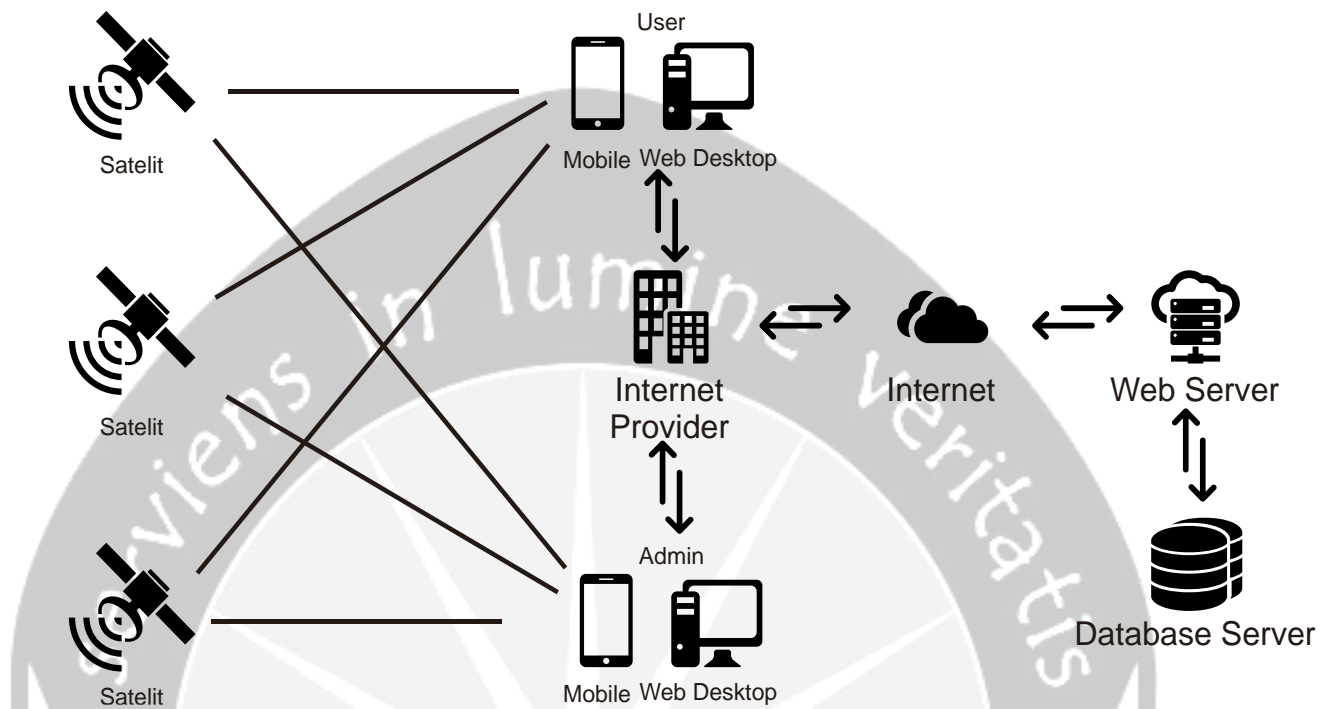
2.1. Perspektif Produk

SIMPARKIR 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 pengelolaan parkir yang terjadi di Dinas Perhubungan Kabupaten Purbalingga 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 masyarakat. Aplikasi ini menangani antara lain: pengelolaan pengguna, pengelolaan data juru parkir, pengelolaan lokasi parkir, pengelolaan retribusi parkir, dan aduan masyarakat. Untuk aplikasi *mobile*, sistem ini dapat digunakan BTP dan masyarakat. Melalui aplikasi *mobile* tersebut BTP melakukan verifikasi lokasi dan masyarakat dapat melihat pelaporan PAD keseluruhan per bulan, lokasi parkir, juru parkir, tarif parkir, daftar juru parkir dan aduan masyarakat.

Perangkat lunak *web* SIMPARKIR 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 Rancangan Arsitektur Perangkat Lunak SIMPARKIR

2.2. Fungsi produk

Fungsi produk perangkat lunak SIMPARKIR adalah sebagai berikut:

APLIKASI WEB

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

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

2. Fungsi *Reset Password* (**SKPL-SIMPARKIR-002**).

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

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

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

Fungsi Pengelolaan Pengguna mencakup:

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

Merupakan fungsi yang digunakan untuk menambahkan data *user*.

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

Merupakan fungsi yang digunakan untuk menampilkan data *user*.

4. Fungsi Pengelolaan Biaya Parkir **(SKPL-SIMPARKIR-004)**.

Merupakan fungsi yang digunakan BPDP untuk mengelola biaya parkir. Fungsi Pengelolaan Biaya Parkir mencakup:

a. Fungsi *Edit* Data Biaya Parkir **(SKPL-SIMPARKIR-004-01)**.

Merupakan fungsi yang digunakan untuk *edit* data parkir.

b. Fungsi *Display* Data Biaya Parkir **(SKPL-SIMPARKIR-004-02)**.

Merupakan fungsi yang digunakan untuk menampilkan data biaya parkir.

5. Fungsi Pengelolaan Lokasi Parkir **(SKPL-SIMPARKIR-005)**.

Merupakan fungsi yang digunakan BTP untuk mengelola lokasi parkir. Fungsi Pengelolaan Lokasi Parkir mencakup:

a. Fungsi *Entri* Lokasi Parkir **(SKPL-SIMPARKIR-005-01)**.

Merupakan fungsi yang digunakan untuk menambahkan data lokasi parkir.

b. Fungsi *Edit* Lokasi Parkir **(SKPL-SIMPARKIR-005-02)**.

Merupakan fungsi yang digunakan untuk mengubah lokasi parkir.

c. Fungsi *Hapus* Lokasi Parkir **(SKPL-SIMPARKIR-005-03)**.

Merupakan fungsi yang digunakan untuk menghapus lokasi parkir.

d. Fungsi *Display* Lokasi Parkir **(SKPL-SIMPARKIR-005-04)**

Merupakan fungsi yang digunakan untuk menampilkan lokasi parkir.

6. Fungsi Pengelolaan Data Pengajuan Calon Juru Parkir **(SKPL-SIMPARKIR-006)**

Merupakan fungsi yang digunakan BLP untuk mengelola data pengajuan calon juru parkir. Fungsi pengelolaan calon juru parkir mencakup:

a. Fungsi *Entri* Data Calon Juru Parkir **(SKPL-SIMPARKIR-006-01)**

Merupakan fungsi yang digunakan untuk menambah data calon juru parkir

- b. Fungsi *Edit* Data Calon Juru Parkir (**SKPL-SIMPARKIR-006-02**).
Merupakan fungsi yang digunakan untuk *edit* data calon juru parkir.
- c. Fungsi *Display* Data Calon Juru Parkir (**SKPL-SIMPARKIR-006-03**).
Merupakan fungsi yang digunakan untuk menampilkan data calon juru parkir.
7. Fungsi Pengelolaan Data Juru Parkir (**SKPL-SIMPARKIR-007**).
Merupakan fungsi yang digunakan BLP untuk mengelola Data Juru Parkir.
- a. Fungsi *Entri* Data Juru Parkir (**SKPL-SIMPARKIR-007-01**).
Merupakan fungsi yang digunakan untuk menambahkan data calon juru parkir.
- b. Fungsi *Edit* Data Juru Parkir (**SKPL-SIMPARKIR-007-02**).
Merupakan fungsi yang digunakan untuk mengubah data juru parkir.
- c. Fungsi *Display* Data Juru Parkir (**SKPL-SIMPARKIR-007-03**).
Merupakan fungsi yang digunakan untuk menampilkan data juru parkir.
8. Fungsi Pengelolaan Data Retribusi Parkir (**SKPL-SIMPARKIR-008**).
Merupakan fungsi yang digunakan BLP untuk mengelola data retribusi Parkir.

- 
- a. Fungsi Bayar Retribusi Parkir **(SKPL-SIMPARKIR-008-01)**
Merupakan fungsi yang digunakan untuk membayar retribusi parkir per bulan.
 - b. Fungsi *Display* Data Retribusi Parkir **(SKPL-SIMPARKIR-008-02)**.
Merupakan fungsi yang digunakan untuk menampilkan data retribusi parkir.
 9. Fungsi Pengelolaan Penugasan Juru Parkir **(SKPL-SIMPARKIR-009)**.
Merupakan fungsi yang digunakan BLP untuk mengelola penugasan juru parkir.
 - a. Fungsi *Entri* Penugasan Juru Parkir **(SKPL-SIMPARKIR-009-01)**.
Merupakan fungsi yang digunakan untuk menambahkan penugasan juru parkir.
 - b. Fungsi *Display* Penugasan Juru Parkir **(SKPL-SIMPARKIR-009-02)**.
Merupakan fungsi yang digunakan untuk menampilkan penugasan juru parkir.
 10. Fungsi Pengelolaan Aduan Masyarakat **(SKPL-SIMPARKIR-010)**.
Merupakan fungsi yang digunakan BLP untuk mengelola Aduan Masyarakat.
 - a. Fungsi *Reply* Aduan Masyarakat **(SKPL-SIMPARKIR-010-01)**
Merupakan fungsi yang digunakan untuk membalas aduan masyarakat.

- b. Fungsi Hapus Aduan Masyarakat (**SKPL-SIMPARKIR-010-02**) .
Merupakan fungsi yang digunakan untuk menghapus aduan masyarakat.
- c. Fungsi *Display* Aduan Masyarakat (**SKPL-SIMPARKIR-010-03**) .
Merupakan fungsi yang digunakan untuk menampilkan aduan masyarakat.
11. Fungsi Layanan Pendaftaran Juru Parkir (**SKPL-SIMPARKIR-011**) .
Merupakan fungsi yang digunakan Calon Juru Parkir untuk melihat informasi mengenai pendaftaran. Fungsi Transaksi Layanan Pendaftaran Juru Parkir mencakup:
Fungsi *Entri* Pendaftaran Juru Parkir (**SKPL-SIMPARKIR-011-01**) .
Merupakan fungsi yang digunakan untuk *entri* data pendaftaran juru parkir
12. Fungsi Layanan Juru Parkir (**SKPL-SIMPARKIR-012**) .
Merupakan fungsi yang digunakan Juru Parkir dan masyarakat untuk melihat informasi juru parkir, penugasan juru parkir dan lokasi parkir. Fungsi Transaksi Layanan Juru Parkir mencakup:
Fungsi *Display* Juru Parkir (**SKPL-SIMPARKIR-012-01**) .
Merupakan fungsi yang digunakan untuk melihat data juru parkir, penugasan juru parkir, dan lokasi parkir.
13. Fungsi Layanan Pendapatan Asli Daerah (**SKPL-SIMPARKIR-013**) .

Merupakan fungsi yang digunakan masyarakat untuk melihat informasi pendapatan asli daerah parkir Kabupaten Purbalingga. Fungsi pendapatan asli daerah parkir mencakup:

Fungsi *Display* Pendapatan Asli Daerah (**SKPL-SIMPARKIR-013-01**).

Merupakan fungsi yang digunakan untuk melihat data pendapatan asli daerah parkir yang diperoleh Dinas Perhubungan Kabupaten Purbalingga.

APLIKASI MOBILE

14. Fungsi *Login* (**SKPL-SIMPARKIR-014**).

Merupakan fungsi yang digunakan oleh BTP untuk dapat masuk dalam aplikasi yang akan digunakan.

15. Fungsi Pengelolaan Lokasi (**SKPL-SIMPARKIR-015**).

Merupakan fungsi yang digunakan oleh BTP untuk mengelola informasi lokasi yang tersedia. Fungsi pengelolaan lokasi mencakup:

a. Fungsi Verifikasi Lokasi Parkir (**SKPL SIMPARKIR-015-01**).

Merupakan fungsi yang digunakan untuk verifikasi lokasi parkir.

b. Fungsi Tampil Lokasi Parkir (**SKPL-SIMPARKIR-015-02**).

Merupakan fungsi yang digunakan untuk menampilkan data lokasi parkir.

16. Fungsi Layanan Peraturan Daerah Parkir (**SKPL-SIMPARKIR-016**).

Merupakan fungsi yang digunakan Masyarakat untuk melihat informasi peraturan daerah mengenai parkir yang tersedia. Fungsi layanan peraturan daerah mencakup:

Fungsi Tampil Peraturan Daerah **(SKPL-SIMPARKIR-016-01)**.

Merupakan fungsi yang digunakan untuk menampilkan peraturan daerah Kabupaten Purbalingga mengenai parkir.

17. Fungsi Layanan Tarif Parkir **(SKPL- SIMPARKIR-017)**.

Merupakan fungsi yang digunakan Masyarakat untuk melihat informasi mengenai tarif parkir yang digunakan pada Kabupaten Purbalingga. Fungsi layanan tarif parkir mencakup:

Fungsi Tampil Tarif Parkir **(SKPL-SIMPARKIR-017-01)**.

Merupakan fungsi yang digunakan untuk menampilkan tarif parkir yang digunakan pada Kabupaten Purbalingga.

18. Fungsi Layanan Pengajuan Juru Parkir **(SKPL-SIMPARKIR-018)**.

Merupakan fungsi yang digunakan Calon Juru Parkir untuk daftar sebagai juru parkir. Fungsi pengajuan juru parkir mencakup:

a. Fungsi Prasyarat Juru Parkir **(SKPL-SIMPARKIR-018-01)**.

Merupakan fungsi yang digunakan untuk menampilkan prasyarat juru parkir.

- b. Fungsi Daftar Juru Parkir (**SKPL SIMPARKIR-018-02**) .
Merupakan fungsi yang digunakan untuk daftar juru parkir.
19. Fungsi Pengelolaan Cari Juru Parkir (**SKPL-SIMPARKIR-019**) .
Merupakan fungsi yang digunakan juru parkir untuk melihat detail informasi miliknya. Fungsi cari juru parkir mencakup:
- a. Fungsi *Entri* No Tugas (**SKPL SIMPARKIR-019-01**) .
Merupakan fungsi yang digunakan untuk melakukan pencarian data juru parkir berdasarkan no tugas.
- b. Fungsi Tampil Juru Parkir (**SKPL-SIMPARKIR-019-02**) .
Merupakan fungsi yang digunakan untuk menampilkan detail data juru parkir.
20. Fungsi Pengelolaan Lokasi Parkir (**SKPL-SIMPARKIR-020**) .
Merupakan fungsi yang digunakan Masyarakat untuk melihat lokasi parkir. Fungsi Pengelolaan Lokasi Parkir mencakup:
Fungsi *Display* Lokasi Parkir (**SKPL SIMPARKIR-020-01**) .
Merupakan fungsi yang digunakan untuk melihat lokasi parkir dan detail juru parkir yang bertugas pada lokasi parkir tersebut.
21. Fungsi Pengelolaan Aduan Masyarakat (**SKPL-SIMPARKIR-021**) .

Merupakan fungsi yang digunakan Masyarakat untuk mengelola aduan masyarakat yang tersedia. Fungsi Aduan Masyarakat mencakup:

a. Fungsi *Entri* Aduan (**SKPL SIMPARKIR-021-01**).

Merupakan fungsi yang digunakan untuk menambah aduan.

b. Fungsi Tampil Aduan (**SKPL-SIMPARKIR-021-02**).

Merupakan fungsi yang digunakan untuk menampilkan aduan.

2.3. Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SIM Parkir adalah sebagai berikut:

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

2.4. Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SIMPARKIR tersebut adalah:

1. Kebijaksanaan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak SIMPARKIR.

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* 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 SIMPARKIR 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 SIMPARKIR adalah:

- Perangkat *Web*.
- Perangkat *Mobile*.
- Perangkat *Database Server*.
- Perangkat *Web Server*.

3. Antarmuka perangkat lunak

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

- a) Nama : Android Studio
Sumber : Google
Sebagai *framework* dalam pembuatan aplikasi *mobile*.
- b) Nama : Android ver 4.0 keatas
Sumber : Android *Open Source Project* (AOSP)
Sebagai sistem operasi untuk *smartphone*.
- c) Nama : Google Chrome/IE/Firefox/Opera/dll
Sumber : berbagai sumber
Sebagai aplikasi web browser untuk membuka sistem *web*.
- d) Nama : Apache
Sumber : Apache Software Foundation.
Sebagai *web server*.
- e) Nama : MySQL
Sumber : MySQL.

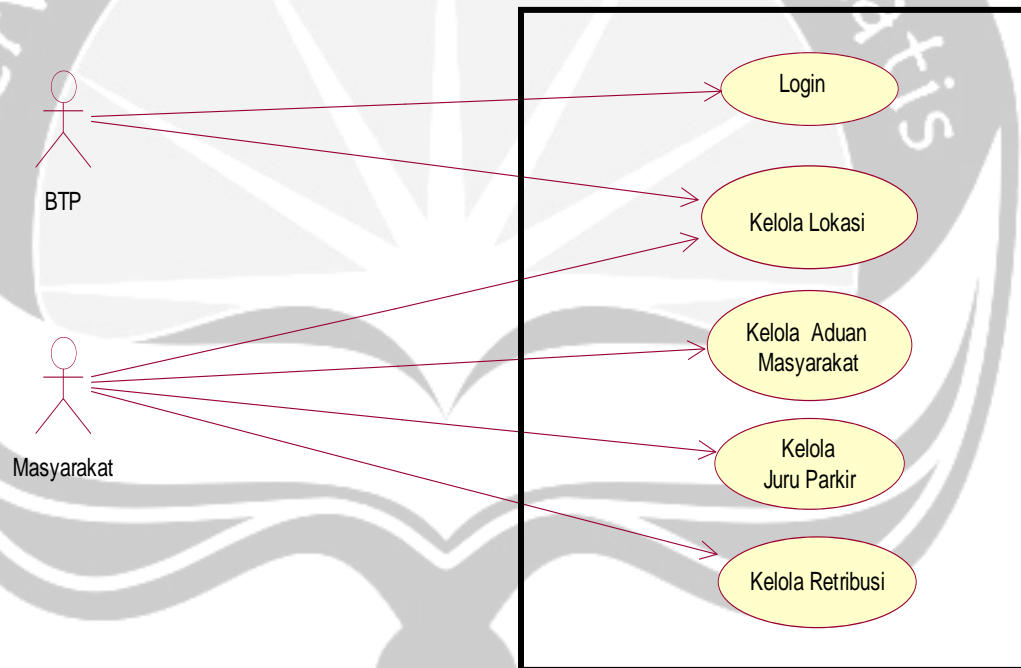
Sebagai Database Management System (DBMS) yang digunakan untuk penyimpanan data di sisi server.

4. Antarmuka komunikasi

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

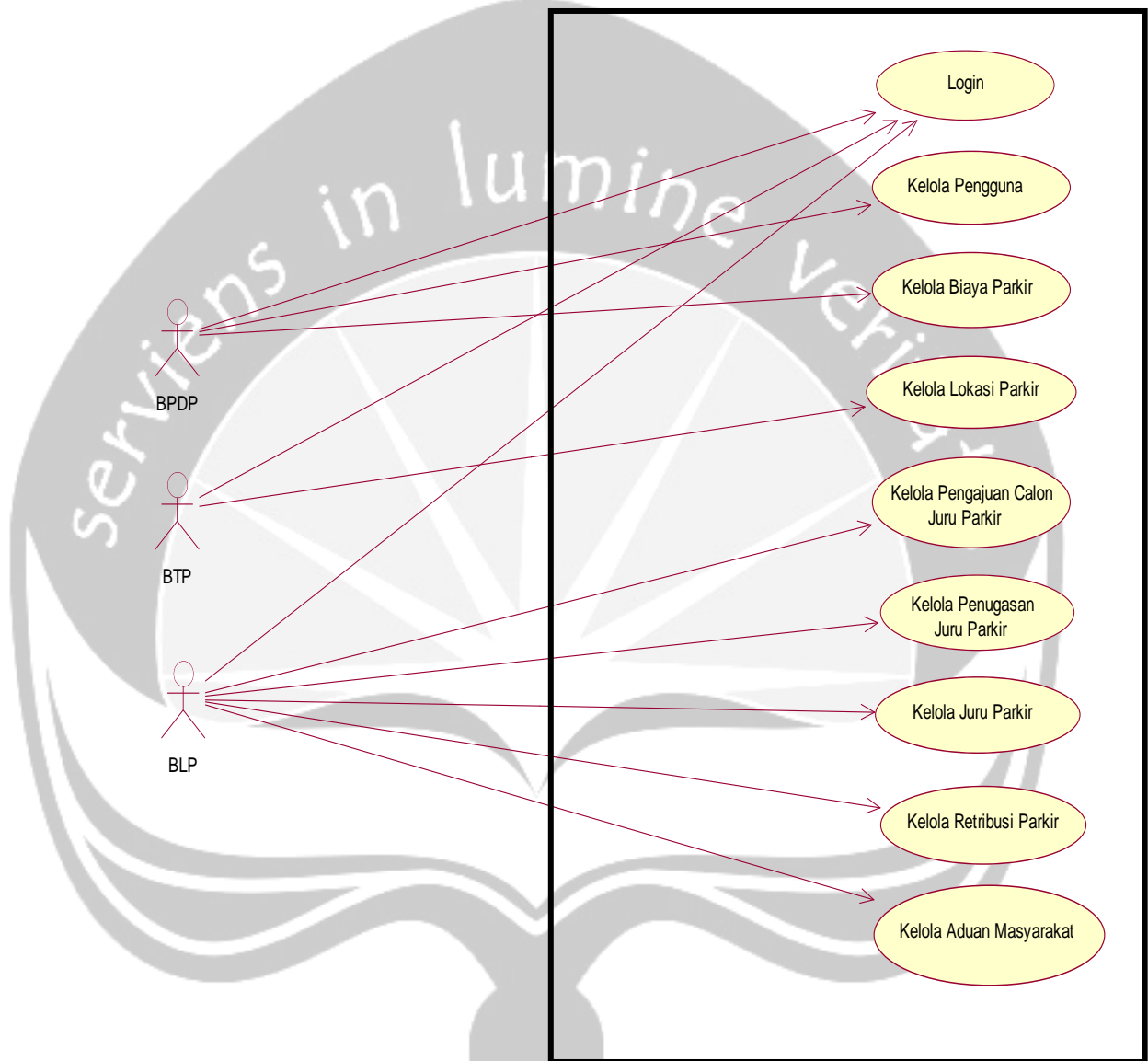
3.2. Kebutuhan Fungsionalitas Perangkat Lunak

3.2.1. Use Case Diagram Aplikasi Mobile



Gambar 3.1. Use Case Diagram Aplikasi Mobile SIMPARKIR

3.2.2. Use Case Diagram Aplikasi Web



Gambar 3.2. Use Case Diagram Aplikasi Web SIMPARKIR

4. Spesifikasi Rinci Kebutuhan

4.1. Spesifikasi Kebutuhan Fungsionalitas

APLIKASI WEB

4.1.1. Use case Specification: Pengelolaan Kategori Pengguna.

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola kategori pengguna. Aktor dapat melakukan *entri new* kategori pengguna dan menampilkan 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 *entri* data baru kategori pengguna.
- e) Aktor memilih untuk melakukan *entri new* data kategori pengguna.
- f) Aktor meminta sistem untuk menyimpan data kategori pengguna yang telah diinputkan.

g) Sistem mengecek data kategori pengguna yang telah diinputkan.

h) Sistem menyimpan data kategori pengguna ke *database*.

i) *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 kategori pengguna di *database* telah terupdate.

4.1.2. Use case Spesification: Pengelolaan Biaya Parkir.

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola biaya parkir. Aktor dapat melakukan *edit* biaya parkir yang sudah ada.

2. *Primary Actor*

BPDP

3. *Supporting Actor*

-

4. *Basic Flow*

- a) Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan biaya parkir.
- b) Sistem menampilkan antarmuka pengelolaan biaya parkir.
- c) Sistem menampilkan data biaya parkir.
- d) Sistem memberikan pilihan untuk melakukan *edit* biaya parkir.
- e) Aktor memilih untuk melakukan *edit* biaya parkir.
- f) Sistem menampilkan pilihan untuk konfirmasi dan batal.
- g) Aktor memilih untuk melakukan konfirmasi.
 - A-1 Aktor memilih untuk melakukan batal menyimpan data biaya parkir.
- h) Sistem mengecek data biaya parkir yang telah dikonfirmasi.
 - E-1 Data biaya parkir yang telah dikonfirmasi salah.
- i) Sistem menyimpan data biaya parkir ke *database*.
- j) *Use Case* selesai.

5. *Alternatif Flow*

A-1 Aktor memilih untuk melakukan batal penyimpanan data biaya parkir.

Berlanjut ke Basic Flow langkah ke-h.

6. *Error Flow*

E-1 Data biaya parkir yang telah dikonfirmasi salah.

a) Sistem memberikan pesan peringatan bahwa data yang dikonfirmasi 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 Biaya Parkir di *database* telah terupdate.

4.1.3. Use case Spesification: Pengelolaan Lokasi Parkir

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola data lokasi parkir. Aktor dapat melakukan *entri* data baru lokasi parkir, *edit* data lokasi parkir, hapus data lokasi parkir, dan menampilkan data lokasi parkir.

2. *Primary Actor*

BTP

3. *Supporting Actor*

-

4. *Basic Flow*

a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan data lokasi parkir.

b) Sistem menampilkan antarmuka pengelolaan data lokasi parkir.

c) Sistem menampilkan data lokasi parkir.

- d) Sistem memberikan pilihan untuk melakukan *entri* data baru lokasi parkir, *edit* data lokasi parkir, hapus data lokasi parkir, dan menampilkan data lokasi parkir.
- e) Aktor memilih untuk melakukan *entri* data baru lokasi parkir.

A-1 Aktor memilih untuk melakukan *edit* data lokasi parkir.

A-2 Aktor memilih untuk melakukan hapus data lokasi parkir.

f) Aktor meminta sistem untuk menyimpan data lokasi parkir yang telah diinputkan.

g) Sistem mengecek data lokasi parkir yang telah diinputkan.

E-1 Data lokasi parkir yang diinputkan aktor salah.

h) Sistem menyimpan data lokasi parkir ke *database*.

i) *Use Case* Selesai.

5. *Alternative Flow*

A-1 Aktor memilih untuk melakukan *edit* data lokasi parkir.

a) Aktor *edit* data lokasi parkir yang sudah ditampilkan.

b) Aktor meminta sistem untuk menyimpan data lokasi parkir yang telah *diedit*.

c) Sistem melakukan pengecekan terhadap data lokasi parkir yang telah *diedit*.

E-2 Data lokasi parkir yang telah *diedit* salah.

d) Berlanjut ke *Basic Flow* langkah ke-h.

6. *Error Flow*

E-1 Data lokasi parkir 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 lokasi parkir yang diedit aktor salah.

- a) Sistem memberikan pesan peringatan bahwa data yang diedit salah.
- b) 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 lokasi parkir di database telah terupdate.

4.1.4. Use case Spesification: Pengelolaan Juru Parkir.

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola data Juru Parkir. Aktor dapat melakukan *entri* data baru Juru Parkir, *edit* data Juru Parkir.

2. *Primary Actor*

BLP

3. *Supporting Actor*

-

4. *Basic Flow*

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

5. *Alternative Flow*

A-1 Aktor memilih untuk melakukan edit Juru Parkir.

- a) Aktor mengedit data Juru Parkir yang sudah ditampilkan.
- b) Aktor meminta sistem untuk menyimpan data Juru Parkir yang telah diedit.

c) Sistem melakukan pengecekan terhadap data Juru Parkir yang telah diedit.

E-2 Data Juru Parkir yang telah diedit salah.

d) Berlanjut ke *Basic Flow* langkah ke-h.

A-2 Aktor memilih untuk melakukan hapus data Juru Parkir.

a) Sistem menampilkan konfirmasi untuk menghapus data.

b) Berlanjut ke *Basic Flow* langkah ke-h.

6. *Error Flow*

E-1 Data Juru Parkir yang diedit aktor salah.

a) Sistem memberikan pesan peringatan bahwa data yang diedit salah.

b) 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 Juru Parkir di database telah terupdate.

4.1.5. Use case Spesification: Pengelolaan Penugasan Juru Parkir.

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola Pengelolaan Penugasan Juru Parkir. Aktor dapat melakukan Verifikasi Data Baru Juru Parkir, edit Verifikasi Data Juru

Parkir, Menampilkan Riwayat Tugas Juru Parkir, Menampilkan Lokasi Parkir yang ada, dan Daftar Penugasan Juru Parkir.

2. *Primary Actor*

BLP

3. *Supporting Actor*

-

4. *Basic Flow*

a) *Use Case* ini dimulai ketika aktor memilih untuk melakukan verifikasi data baru juru parkir.

b) Sistem menampilkan antarmuka pengelolaan data juru parkir.

c) Sistem meminta *user* menginputkan NIK juru parkir.

d) Sistem menampilkan data Juru Parkir.

e) Sistem memberikan peringatan Juru Parkir belum di verifikasi dan *edit* data Juru Parkir.

f) Sistem memberikan pilihan untuk melakukan verifikasi data baru juru parkir, *edit* data juru parkir.

g) Aktor memilih untuk melakukan *edit* data baru juru parkir.

A-1 Aktor memilih untuk melakukan *edit* data juru parkir.

h) Aktor meminta sistem untuk menyimpan data juru parkir yang telah diinputkan.

i) Sistem mengecek data juru parkir yang telah diinputkan.

E-1 Data juru parkir yang diinputkan aktor salah.

j) Sistem menampilkan riwayat tugas juru parkir.

- k) Aktor memilih untuk tambah penugasan.
- l) Sistem menampilkan data lokasi parkir yang tersedia.
- m) Aktor memilih lokasi parkir.
- n) Aktor memilih untuk daftar penugasan pada lokasi yang telah dipilih.
- o) Aktor melakukan *entri* data penugasan.
- p) Sistem menyimpan data juru parkir ke *database*.
- j) *Use Case* selesai.

5. *Alternative Flow*

A-1 Aktor memilih untuk melakukan *edit* data juru parkir.

- a) Aktor *mengedit* data juru parkir yang sudah ditampilkan.
- b) Aktor meminta sistem untuk menyimpan data juru parkir yang telah *diedit*.
- c) Sistem melakukan pengecekan terhadap data juru parkir yang telah *diedit*.

E-2 Data juru parkir yang telah *diedit* salah.

- d) Berlanjut ke *Basic Flow* langkah ke-g.

6. *Error Flow*

E-1 Data juru parkir 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 juru parkir yang *diedit* aktor salah.

a) Sistem memberikan pesan peringatan bahwa data yang *diedit* salah.

b) 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 juru parkir di *database* telah *terupdate*.

APLIKASI MOBILE

4.1.6. *Use case Spesification: Pengelolaan Lokasi Baru.*

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola lokasi. Aktor dapat melakukan verifikasi lokasi dan menampilkan lokasi yang ada.

2. *Primary Actor*

BTP

3. *Supporting Actor*

-

4. *Basic Flow*

a) *Use case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan lokasi.

- b) Sistem menampilkan antarmuka pengelolaan data lokasi.
- c) Sistem menampilkan data lokasi.
- d) Sistem memberikan pilihan untuk melakukan verifikasi data lokasi.
- e) Aktor memilih untuk melakukan verifikasi data lokasi.
- f) Aktor meminta sistem untuk menyimpan data lokasi yang telah diverifikasi.
- g) Sistem mengecek data lokasi yang telah diverifikasi.
- h) Sistem menyimpan data lokasi ke *database*.
- i) *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 lokasi di *database* telah terupdate.

4.1.7. Use case Spesification: Pengelolaan Juru Parkir.

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola aduan

masyarakat. Aktor dapat menampilkan juru parkir dan lokasi penugasan yang ada.

2. *Primary Actor*

Masyarakat

3. *Supporting Actor*

-

4. *Basic Flow*

a) *Use case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan juru parkir.

b) Sistem menampilkan antarmuka pengelolaan data juru parkir.

c) Sistem menampilkan data juru parkir dan lokasi penugasan.

d) *Use Case* selesai.

5. *PreConditions*

a) *Use Case Login* telah dilakukan.

b) Aktor telah memasuki sistem.

4.1.8. Use case Specification: Pengelolaan Aduan Masyarakat.

1. *Brief Description*

Use Case ini digunakan oleh aktor untuk mengelola aduan masyarakat. Aktor dapat melakukan tambah aduan dan menampilkan aduan yang ada.

2. *Primary Actor*

Masyarakat

3. *Supporting Actor*

-

4. *Basic Flow*

- a) *Use case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan aduan masyarakat.
- b) Sistem menampilkan antarmuka pengelolaan data aduan masyarakat.
- c) Sistem menampilkan data aduan masyarakat.
- d) Sistem memberikan pilihan untuk melakukan tambah data baru aduan.
- e) Aktor memilih untuk melakukan tambah data aduan.
- f) Aktor meminta sistem untuk menyimpan data aduan yang telah diinputkan.
- g) Sistem mengecek data aduan yang telah diinputkan.
- h) Sistem menyimpan data aduan ke *database*.
- i) *Use Case* selesai.

5. *PreConditions*

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

6. *PostConditions*

Data aduan di *database* telah terupdate.

4.1.9. Use case Spesification: Pengelolaan Retribusi Parkir.

1. Brief Description

Use Case ini digunakan oleh aktor untuk retribusi. Aktor dapat menampilkan retribusi yang ada.

2. Primary Actor

Masyarakat

3. Supporting Actor

-

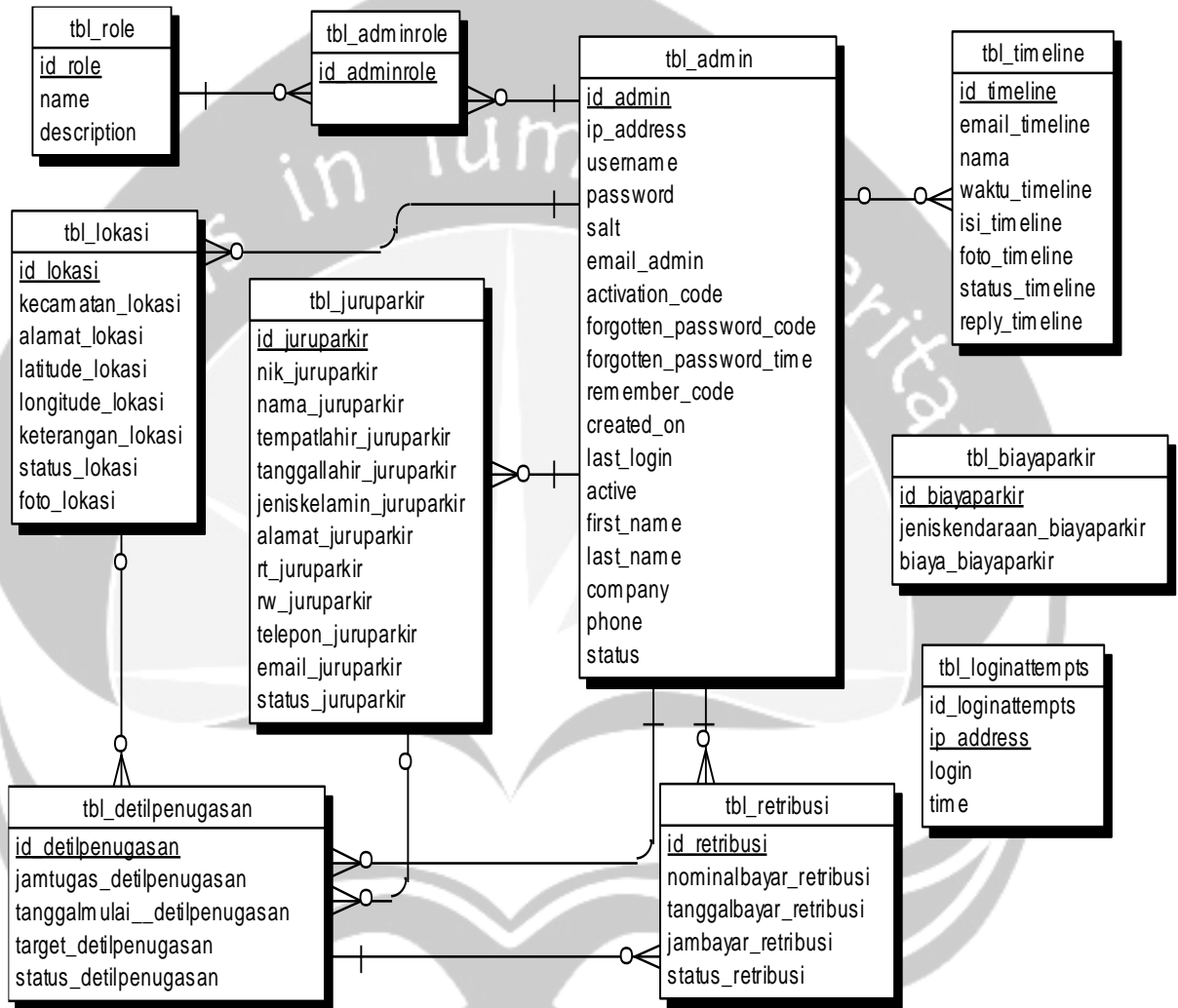
4. Basic Flow

- a) Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan retribusi parkir.
- b) Sistem menampilkan antarmuka pengelolaan data retribusi parkir.
- c) Sistem menampilkan data PAD parkir.
- d) Use Case selesai.

5. PreConditions

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

5. Entity Relationship Diagram



Gambar 5.1 Entity Relationship Diagram SIMPARKIR

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

**PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN PARKIR
BERBASIS WEB DAN MOBILE**

(STUDI KASUS DINAS PERHUBUNGAN KABUPATEN PURBALINGGA)

Untuk:

Universitas Atma Jaya Yogyakarta


Dipersiapkan oleh:

Sukma Stevia Charolline / 130707335

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<i>DPPL-SIMPARKIR</i>		1/67
		<i>Revisi</i>	A	

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	

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

DAFTAR HALAMAN PERUBAHAN

Halaman	Revisi	Halaman	Revisi

DAFTAR ISI

1. Pendahuluan.....	8
1.1 Tujuan	8
1.2 Lingkup Masalah	8
1.3 Definisi, Akronim dan Singkatan	9
1.4 Referensi	10
2. Perancangan Sistem.....	11
2.1 Perancangan Arsitektur	11
2.2 Perancangan Rinci	12
2.2.1 <i>Sequence Diagram</i>	12
2.2.2 <i>Class Diagram</i>	30
2.2.3 <i>Class Diagram Specification</i>	32
3. Perancangan Data.....	50
3.1 Dekomposisi Data	50
3.2 <i>Physical Data Model</i>	55
4. Deskripsi Perancangan Antarmuka.....	56
4.1 Antarmuka Halaman Beranda	56
4.2 Antarmuka Halaman Layanan	56
4.3 Antarmuka Halaman PAD	57
4.4 Antarmuka Halaman Juru Parkir	58
4.5 Antarmuka Halaman Pendaftaran Juru Parkir	59
4.6 Antarmuka Halaman Aplikasi <i>Mobile</i>	59
4.7 Antarmuka Halaman Aduan Masyarakat	60
4.8 Antarmuka Halaman <i>Login Web</i>	60

4.9 Antarmuka Halaman Pengelolaan Admin	61
4.10 Antarmuka Halaman Pengelolaan Biaya Parkir	61
4.11 Antarmuka Halaman Pengelolaan Lokasi Parkir	62
4.12 Antarmuka Halaman Pengelolaan Juru Parkir	62
4.13 Antarmuka Halaman Penugasan Juru Parkir	63
4.14 Antarmuka Halaman <i>Login</i>	64
4.15 Antarmuka Halaman Pengelolaan Lokasi	64
4.16 Antarmuka Halaman Pengelolaan Verifikasi Lokasi	65
4.17 Antarmuka Halaman Beranda	65
4.18 Antarmuka Halaman Juru Parkir	66
4.18 Antarmuka Halaman <i>Timeline</i>	67

DAFTAR GAMBAR

Gambar 2.1 Rancangan Arsitektur SIMPARKIR	11
Gambar 2.2 <i>Sequence Diagram</i> : Login	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 Biaya Parkir	14
Gambar 2.6 <i>Sequence Diagram</i> : Edit Data Biaya Parkir	15
Gambar 2.7 <i>Sequence Diagram</i> : Tambah Data Lokasi Parkir	16
Gambar 2.8 <i>Sequence Diagram</i> : Edit Data Lokasi Parkir	17
Gambar 2.9 <i>Sequence Diagram</i> : Hapus Data Lokasi Parkir	18
Gambar 2.10 <i>Sequence Diagram</i> : Tampil Data Lokasi Parkir	19

Gambar 2.11	<i>Sequence Diagram</i> : Tampil Data Calon Juru Parkir	19
Gambar 2.12	<i>Sequence Diagram</i> : Edit Data Calon Juru Parkir	..20
Gambar 2.13	<i>Sequence Diagram</i> : Tambah Data Juru Parkir21
Gambar 2.14	<i>Sequence Diagram</i> : Edit Data Juru Parkir22
Gambar 2.15	<i>Sequence Diagram</i> : Tampil Juru Parkir23
Gambar 2.16	<i>Sequence Diagram</i> : Tampil Retribusi Parkir23
Gambar 2.17	<i>Sequence Diagram</i> : Tambah Data Retribusi Parkir	.24
Gambar 2.18	<i>Sequence Diagram</i> : Tambah Data Penugasan25
Gambar 2.19	<i>Sequence Diagram</i> : Tampil Penugasan26
Gambar 2.20	<i>Sequence Diagram</i> : Tampil Aduan Masyarakat26
Gambar 2.21	<i>Sequence Diagram</i> : Edit Aduan Masyarakat27
Gambar 2.22	<i>Sequence Diagram</i> : Hapus Aduan Masyarakat28
Gambar 2.23	<i>Sequence Diagram</i> : Tambah Data Pendaftaran29
Gambar 2.24	<i>Sequence Diagram</i> : Tampil Data Pendaftaran30
Gambar 2.25	<i>Class Diagram</i> SIMPARKIR30
Gambar 3.1	<i>Physical Data Model</i>55
Gambar 4.1	Antarmuka Halaman <i>Website</i>56
Gambar 4.2	Antarmuka Halaman <i>Layanan</i>56
Gambar 4.3	Antarmuka Halaman <i>PAD</i>57
Gambar 4.4	Antarmuka Halaman <i>Juru Parkir</i>58
Gambar 4.5	Antarmuka Halaman <i>Pendaftaran Juru Parkir</i>59
Gambar 4.6	Antarmuka Halaman <i>Aplikasi Mobile</i>59
Gambar 4.7	Antarmuka Halaman <i>Aduan Masyarakat</i>60
Gambar 4.8	Antarmuka Halaman <i>Login Web</i>60
Gambar 4.9	Antarmuka Halaman <i>Pengelolaan Admin</i>61

Gambar 4.10	Antarmuka Halaman Pengelolaan Biaya Parkir	61
Gambar 4.11	Antarmuka Halaman Pengelolaan Lokasi Parkir	62
Gambar 4.12	Antarmuka Halaman Pengelolaan Juru Parkir	62
Gambar 4.13	Antarmuka Halaman Pengelolaan Penugasan	63
Gambar 4.14	Antarmuka Halaman <i>Login</i>	64
Gambar 4.15	Antarmuka Halaman Pengelolaan Lokasi	64
Gambar 4.16	Antarmuka Halaman Pengelolaan Verifikasi Lokasi	65
Gambar 4.17	Antarmuka Halaman Beranda	65
Gambar 4.18	Antarmuka Halaman Juru Parkir	66
Gambar 4.19	Antarmuka Halaman <i>Timeline</i>	67



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 SIMPARKIR dikembangkan dengan tujuan untuk:

1. Pengelolaan pengguna sistem, pengelolaan data admin, pengelolaan biaya parkir, pengelolaan lokasi parkir, pengelolaan juru parkir.
2. Proses transaksi biaya parkir mencakup setoran dan pembayaran sesuai target beserta penentuan denda keterlambatan.
3. Proses penugasan juru parkir mencakup penentuan lokasi tugas, jam tugas, setoran tugas.
4. Proses pendatan asli daerah PAD parkir kabupaten purbalingga.

Aplikasi ini berjalan pada smartphone berbasis *mobile* dan *web*.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan:

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga <i>Software Design Description</i> (SDD). Merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
DPPL-SIMPARKIR-XXX	Kode yang merepresentasikan kebutuhan pada SIMPARKIR (Sistem Informasi Manajemen Parkir) di mana XXX merupakan nomor fungsi produk.
SIMPARKIR	Perangkat lunak pengelolaan web <i>desktop</i> dan <i>mobile</i> .
<i>Database</i>	Kumpulan data yang tersimpan dan diorganisasikan agar dapat diambil dengan mudah secara efisien.
<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> .

Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Android	Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet.
Pengelolaan Parkir	Pengelolaan Parkir merupakan salah satu jenis usaha yang memberikan layanan dalam bentuk jasa pengelolaan dan pelayanan tempat untuk memarkir kendaraan para konsumen.
BPDP	Badan Pengelola Data Parkir
BTP	Badan Teknis Parkir
BLP	Badan Layanan Parkir

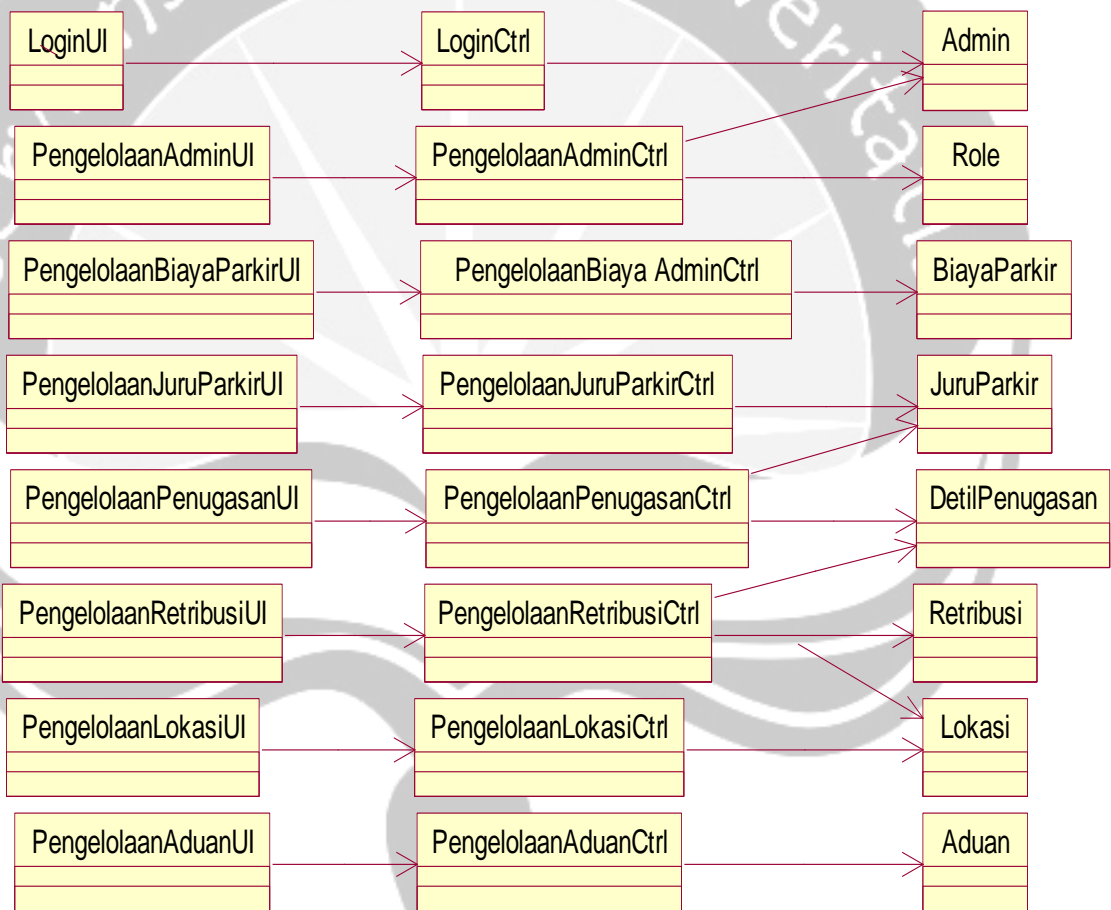
1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Sukma Stevia Charolline / 130707335, Spesifikasi Kebutuhan Perangkat Lunak (SKPL) 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

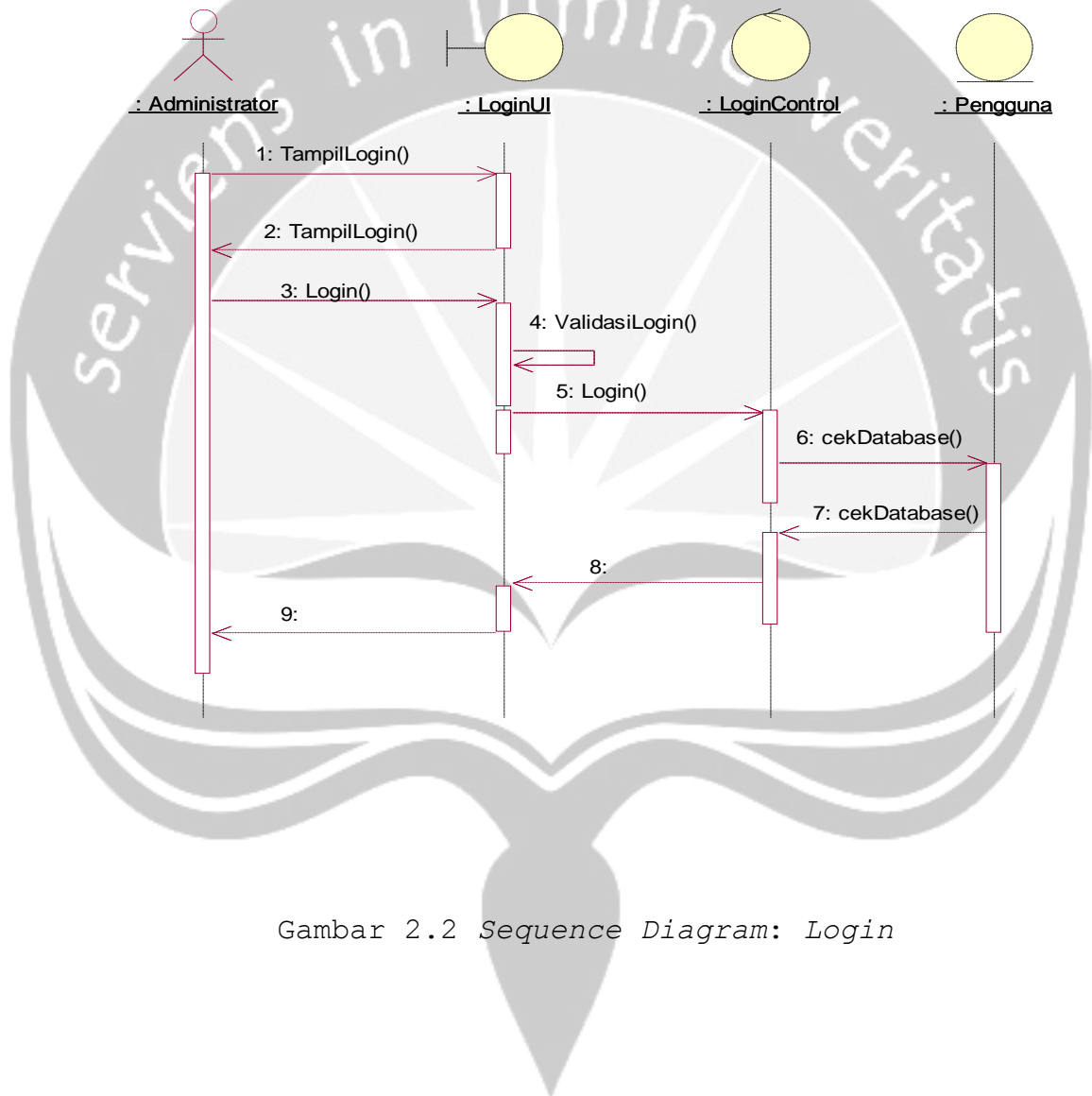


Gambar 2.1 Rancangan Arsitektur SIMPARKIR

2.2 Perancangan Rinci

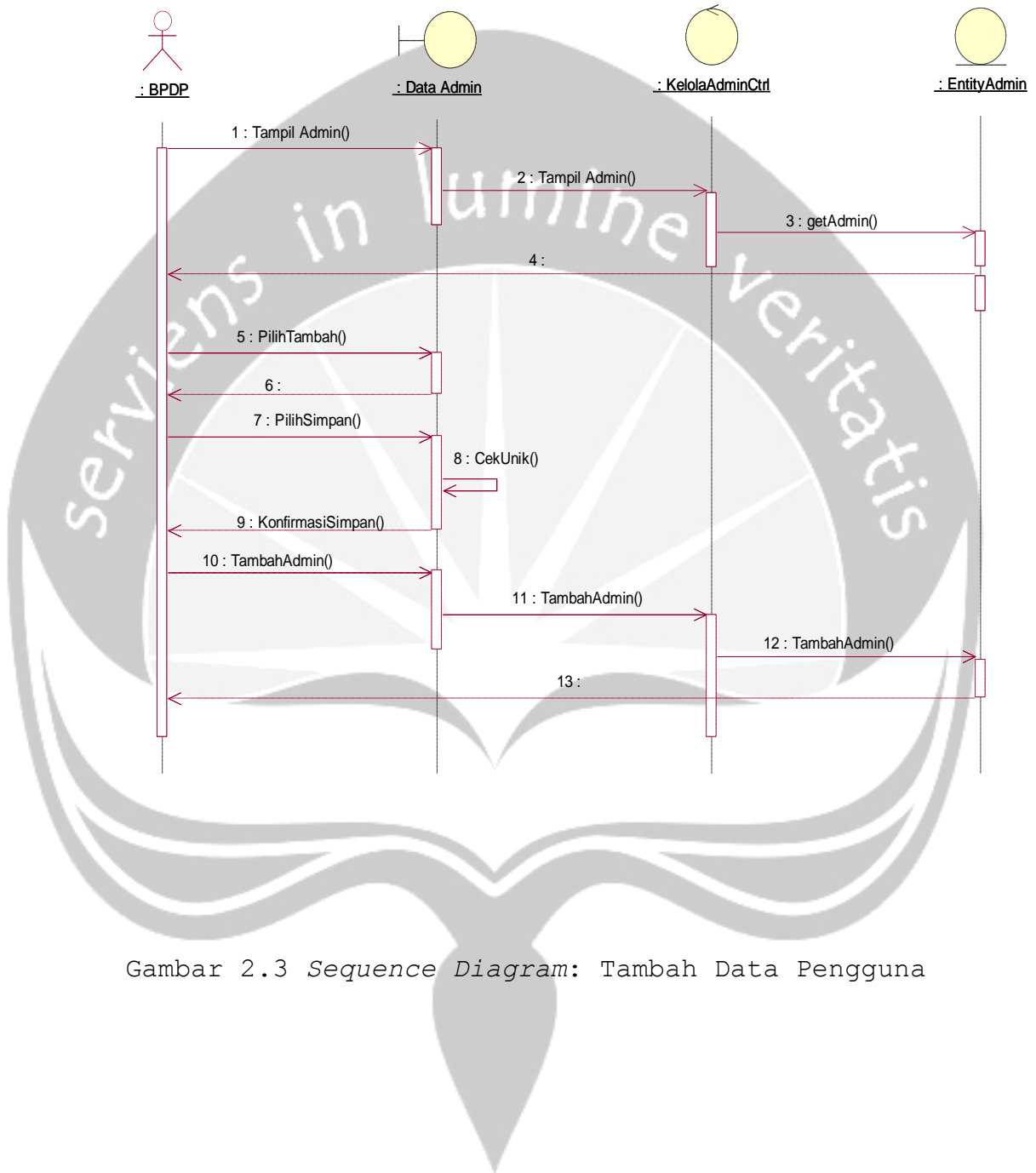
2.2.1. Sequence Diagram

2.2.1.1. Fungsi Login



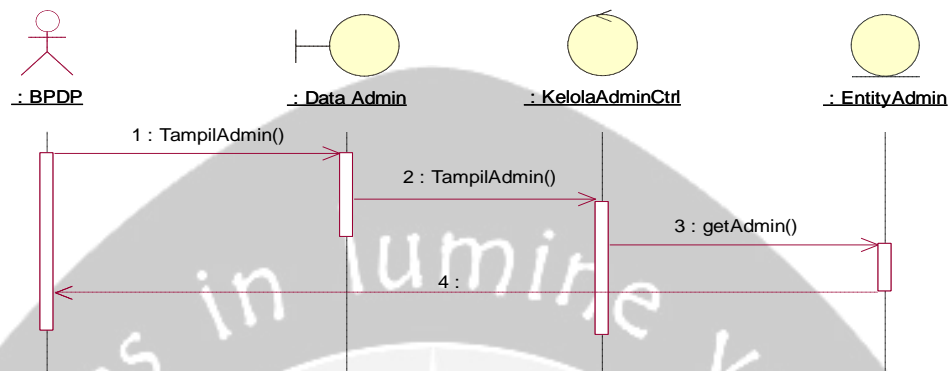
Gambar 2.2 Sequence Diagram: Login

2.2.1.2. Fungsi Tambah Data Pengguna



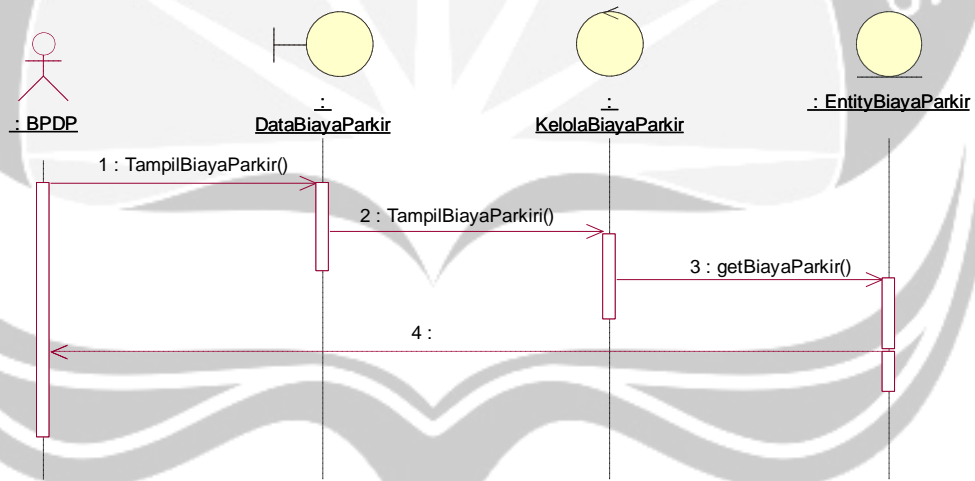
Gambar 2.3 *Sequence Diagram*: Tambah Data Pengguna

2.2.1.3. Fungsi Tampil Data Pengguna



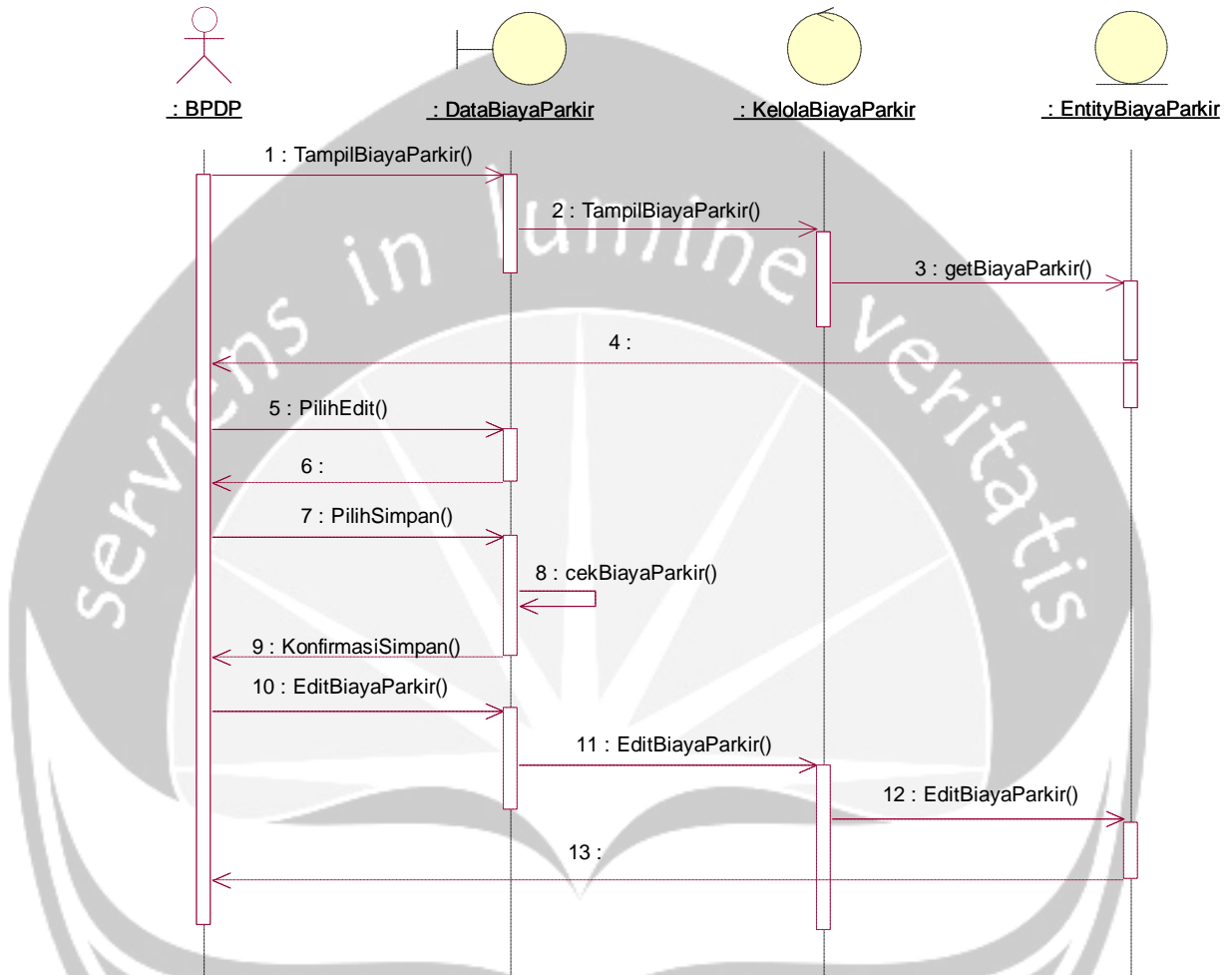
Gambar 2.4 *Sequence Diagram* : Tampil Data Pengguna

2.2.1.4. Fungsi Tampil Data Biaya Parkir



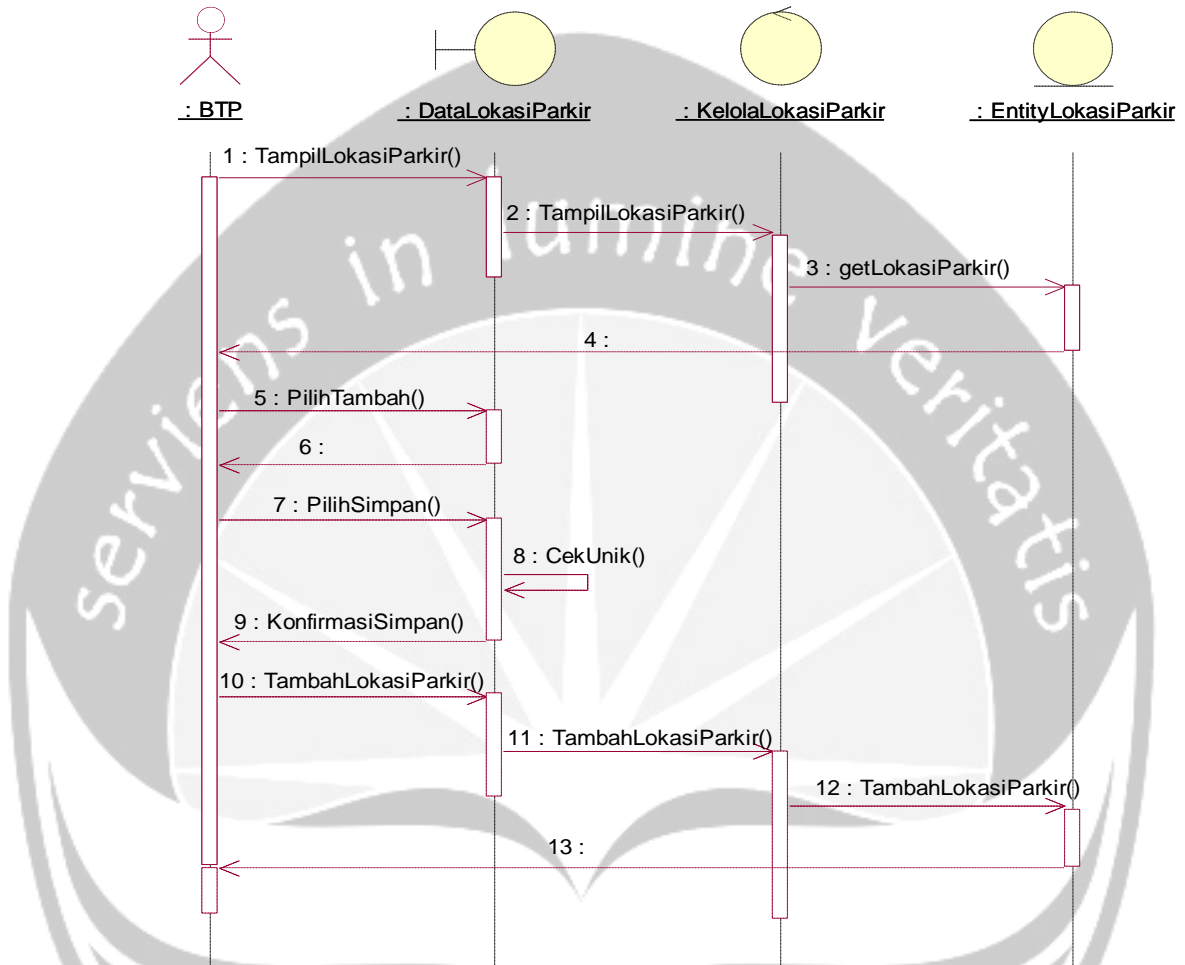
Gambar 2.5 *Sequence Diagram* : Tampil Data Biaya Parkir

2.2.1.5. Fungsi *Edit* Data Biaya Parkir



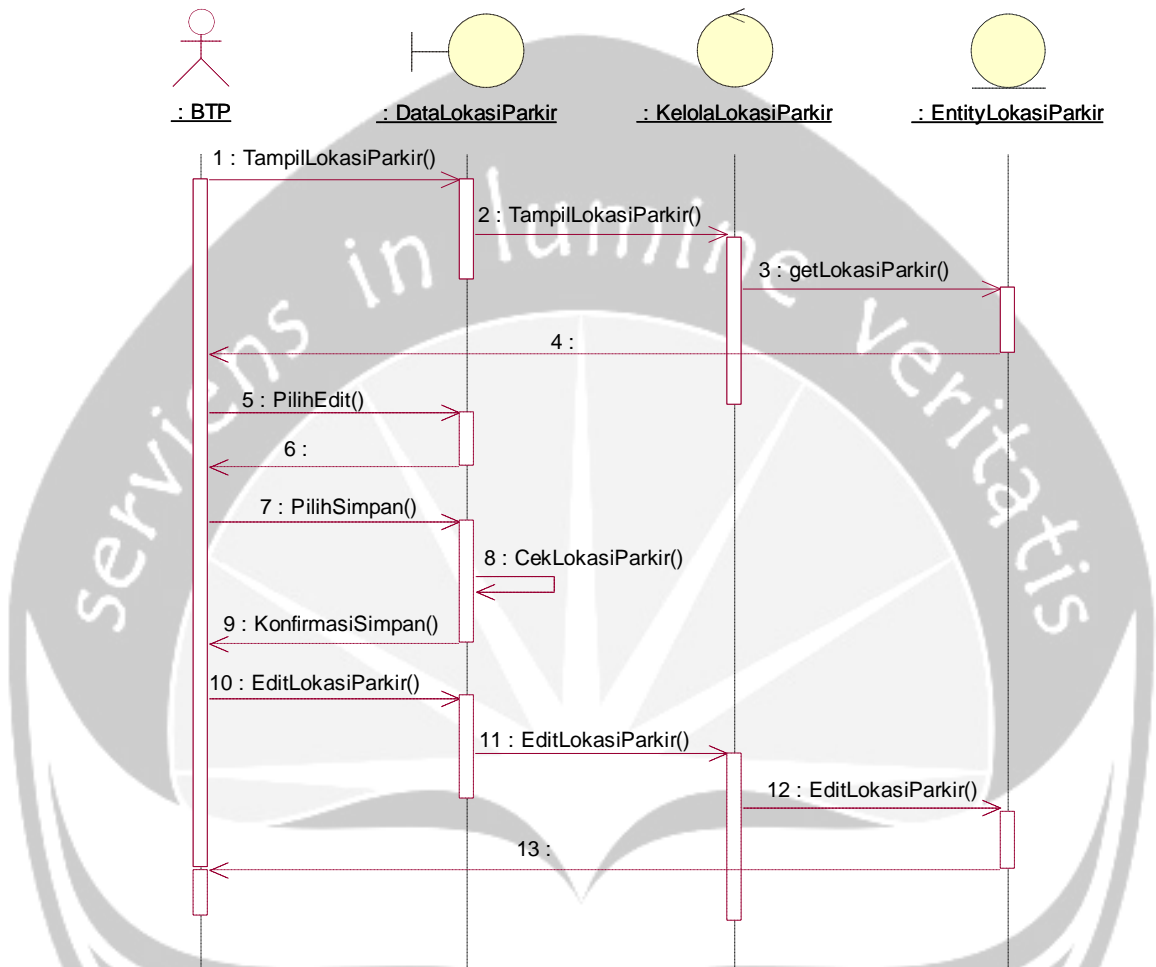
Gambar 2.6 *Sequence Diagram* : Ubah Data Biaya Parkir

2.2.1.6. Fungsi Tambah Data Lokasi Parkir



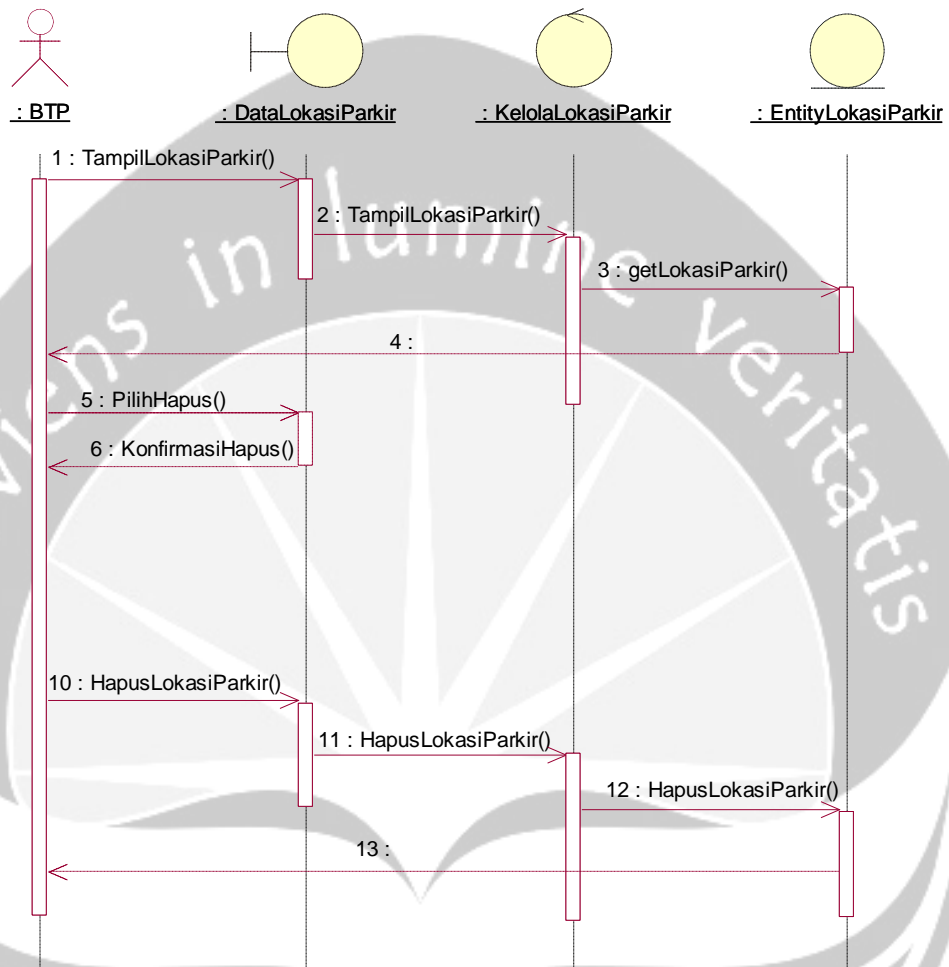
Gambar 2.7 *Sequence Diagram* : Tambah Data Lokasi Parkir

2.2.1.7. Fungsi *Edit* Data Lokasi Parkir



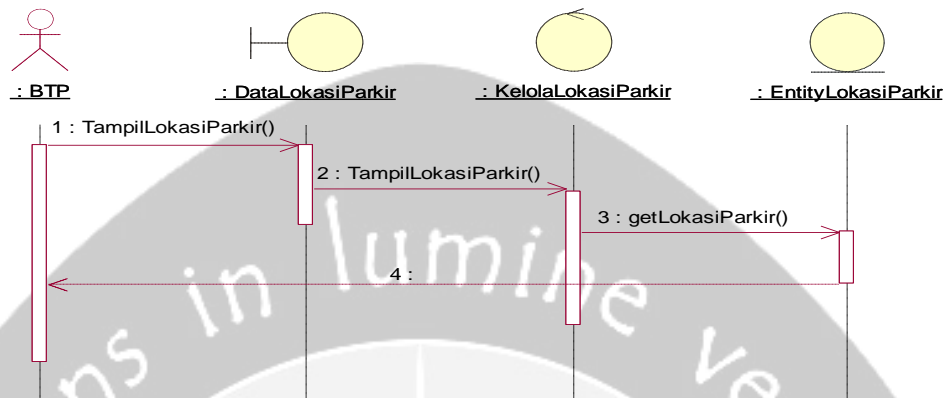
Gambar 2.8 *Sequence Diagram* : Ubah Data Lokasi Parkir

2.2.1.8. Fungsi Hapus Data Lokasi Parkir



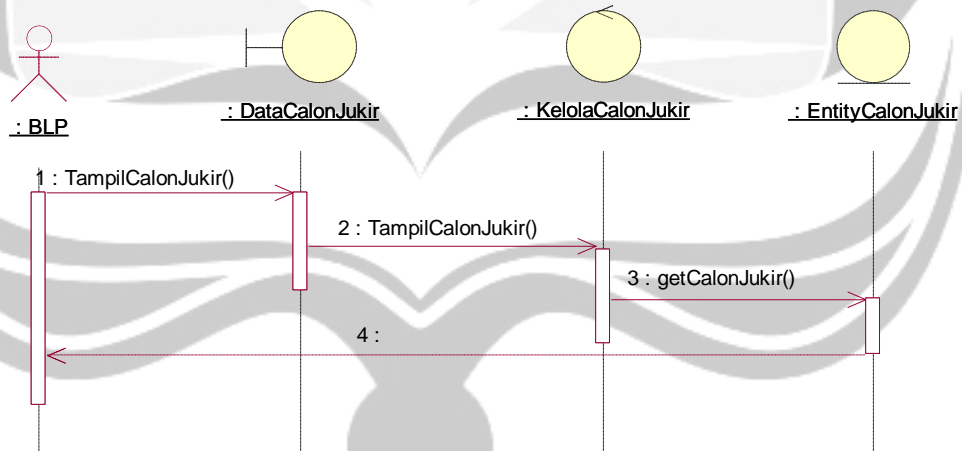
Gambar 2.9 *Sequence Diagram* : Hapus Data Lokasi Parkir

2.2.1.9. Fungsi Tampil Data Lokasi Parkir



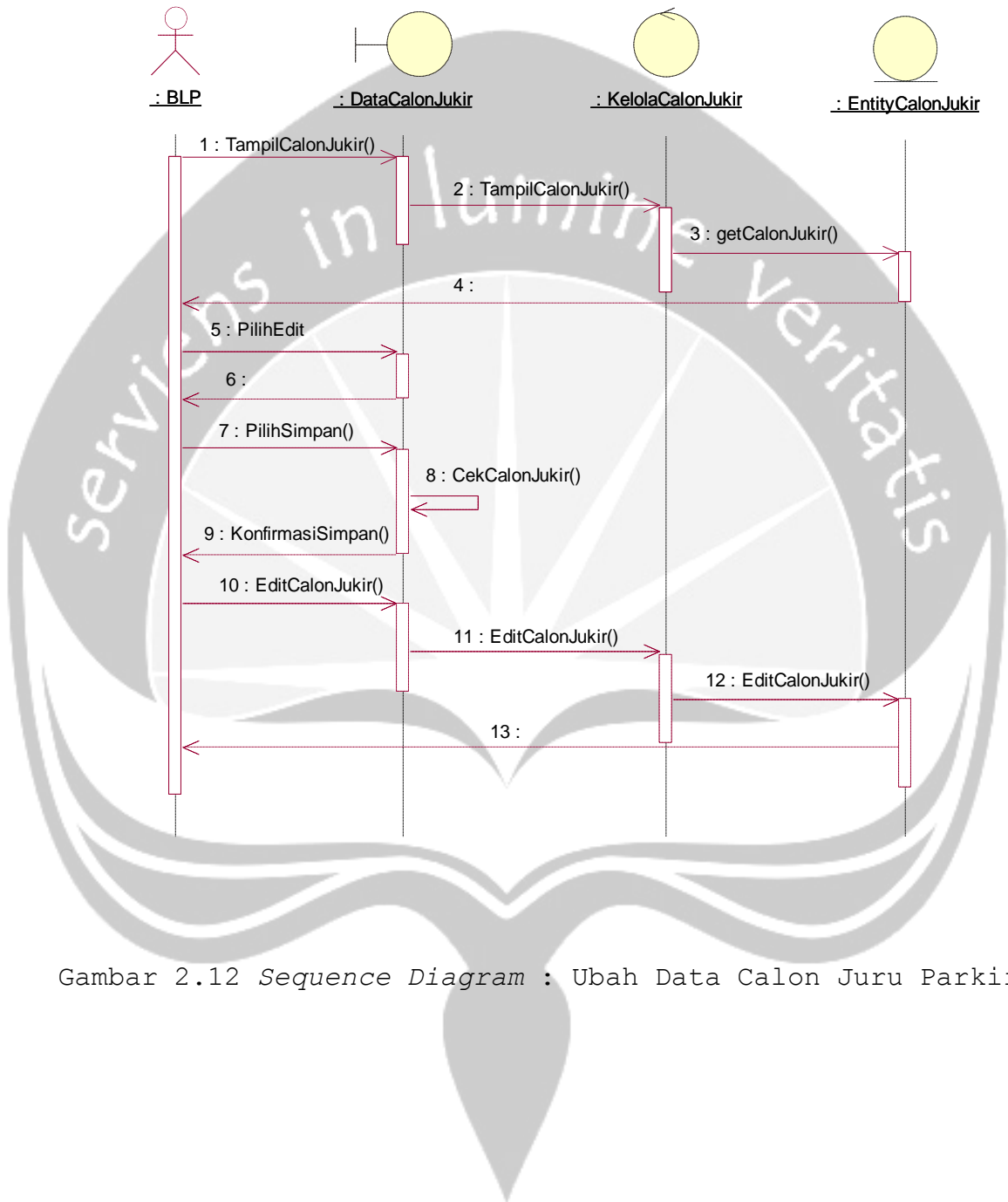
Gambar 2.10 *Sequence Diagram* : Tampil Data Lokasi Parkir

2.2.1.10. Fungsi Tampil Data Calon Juru Parkir



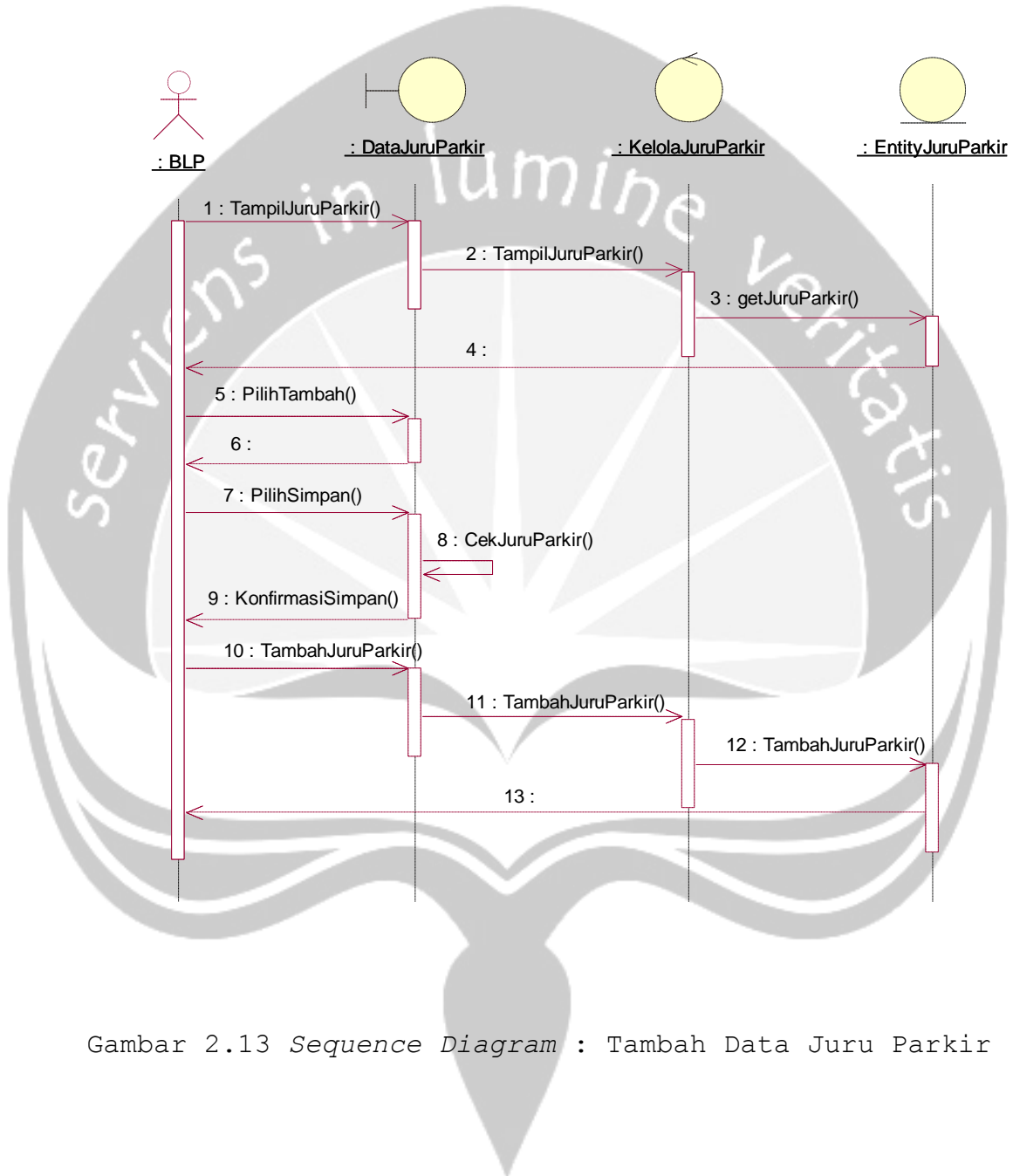
Gambar 2.11 *Sequence Diagram* : Tampil Data Calon Juru Parkir

2.2.1.11. Fungsi *Edit* Data Calon Juru Parkir



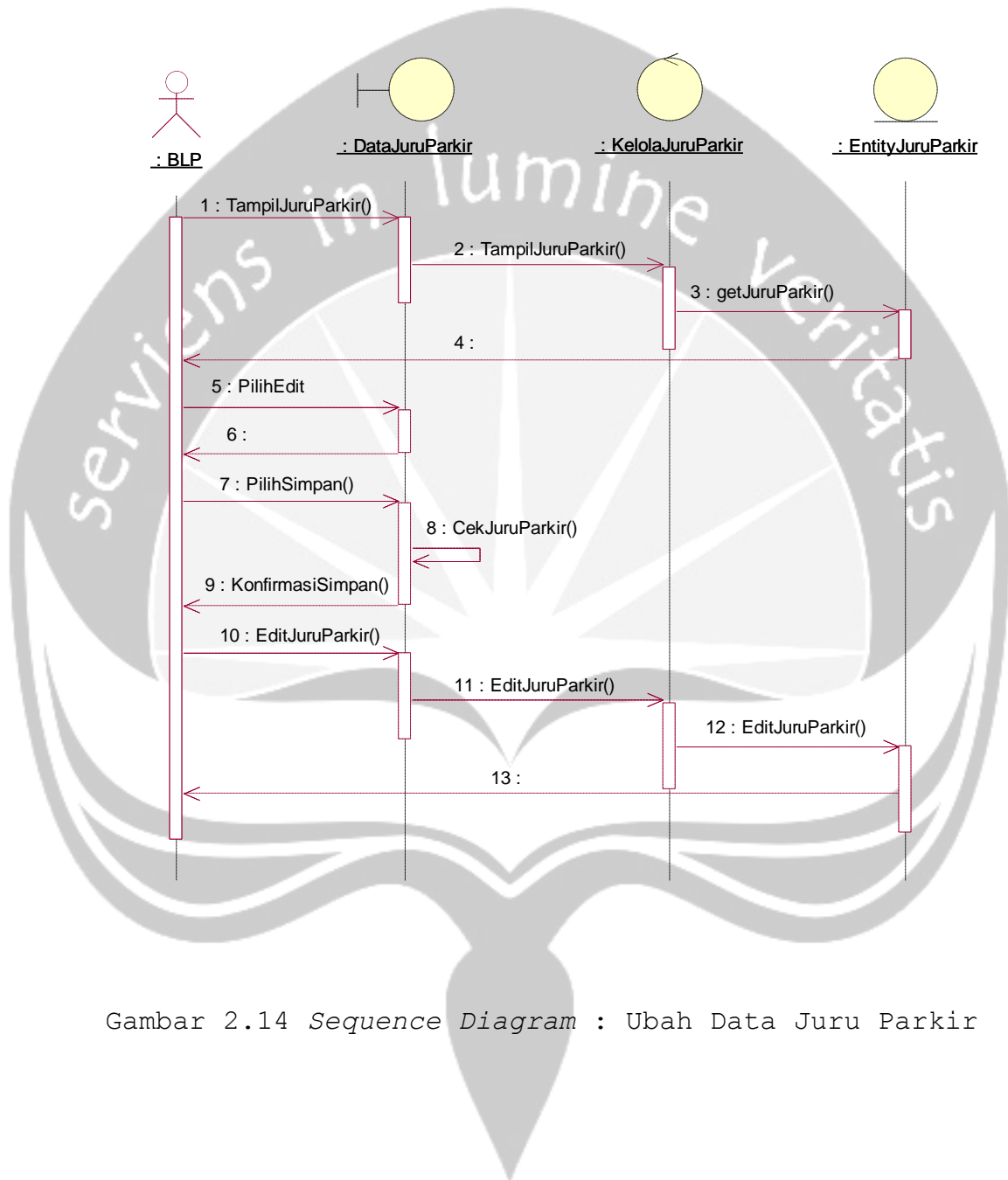
Gambar 2.12 *Sequence Diagram* : Ubah Data Calon Juru Parkir

2.2.1.12. Fungsi Tambah Data Juru Parkir



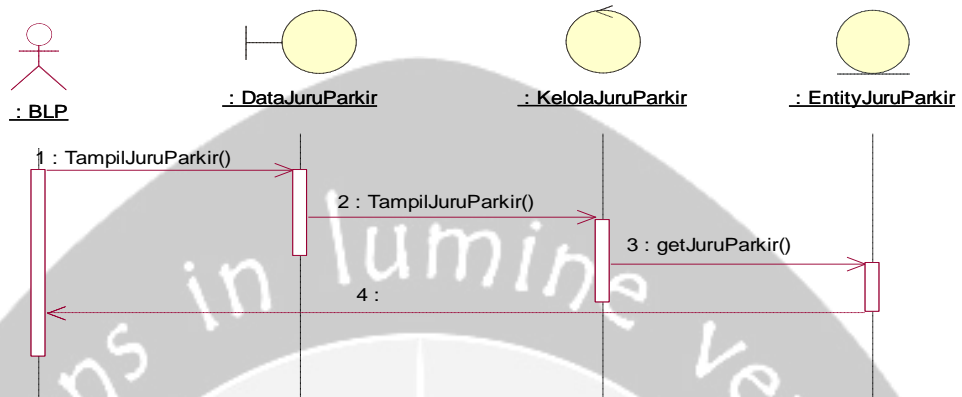
Gambar 2.13 *Sequence Diagram* : Tambah Data Juru Parkir

2.2.1.13. Fungsi *Edit Data Juru Parkir*



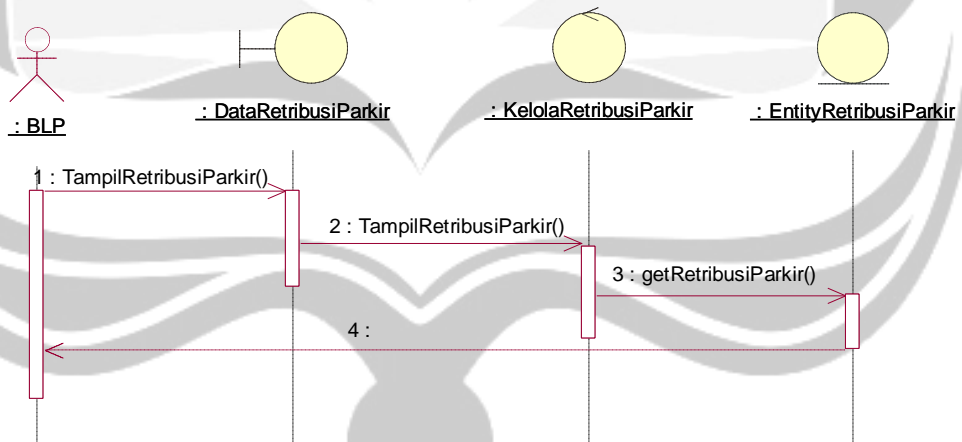
Gambar 2.14 *Sequence Diagram* : Ubah Data Juru Parkir

2.2.1.14. Fungsi *Tampil Data Juru Parkir*



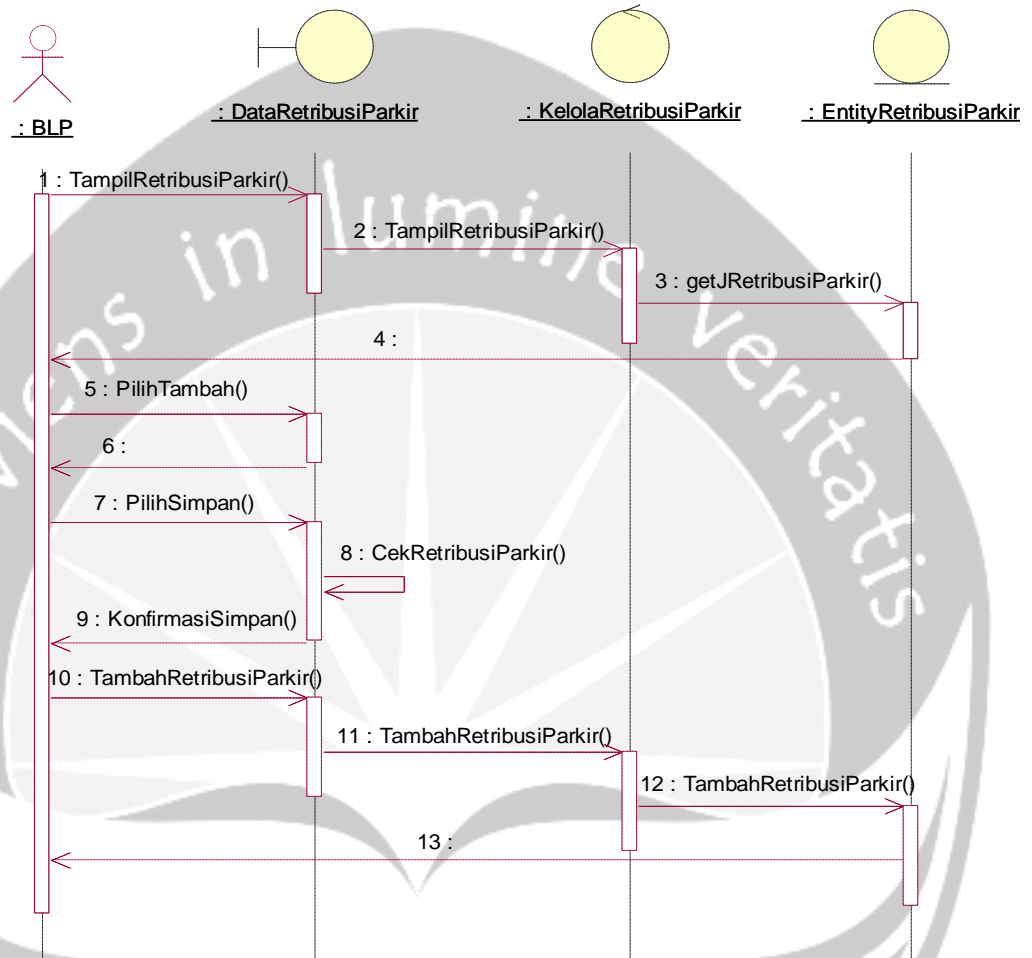
Gambar 2.15 *Sequence Diagram* : Tampil Data Juru Parkir

2.2.1.15. Fungsi Tampil Data Retribusi Parkir



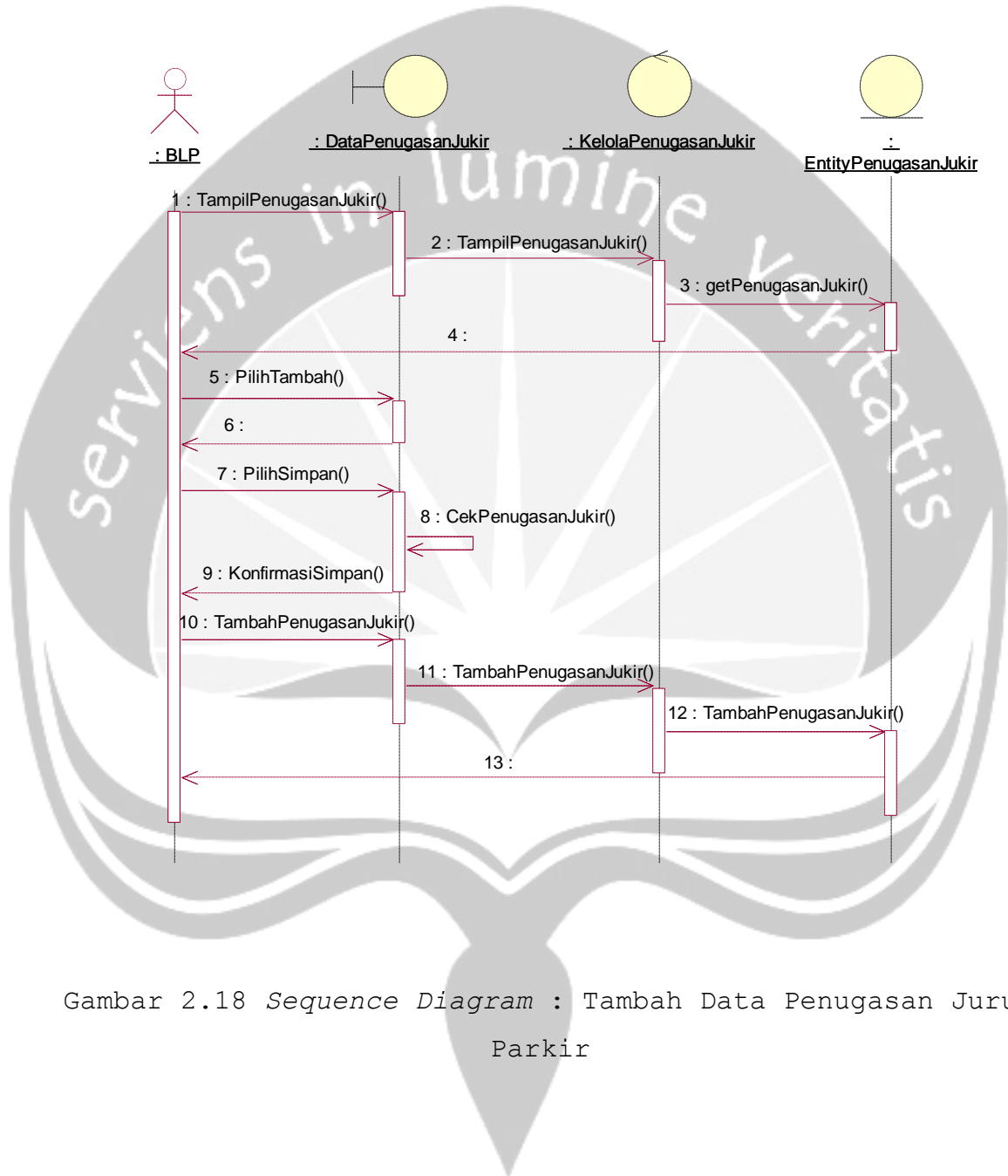
Gambar 2.16 *Sequence Diagram* : Tampil Data Retribusi Parkir

2.2.1.16. Fungsi Tambah Data Retribusi Parkir



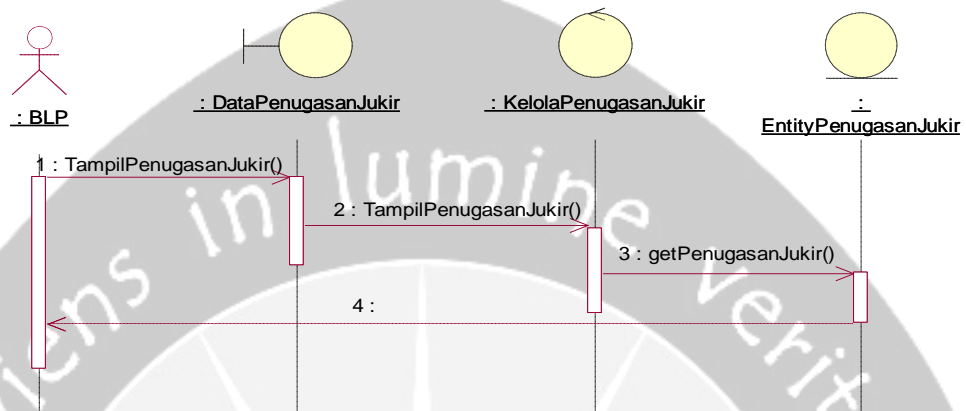
Gambar 2.17 *Sequence Diagram* : Tambah Data Retribusi Parkir

2.2.1.16. Fungsi Tambah Data Penugasan Juru Parkir



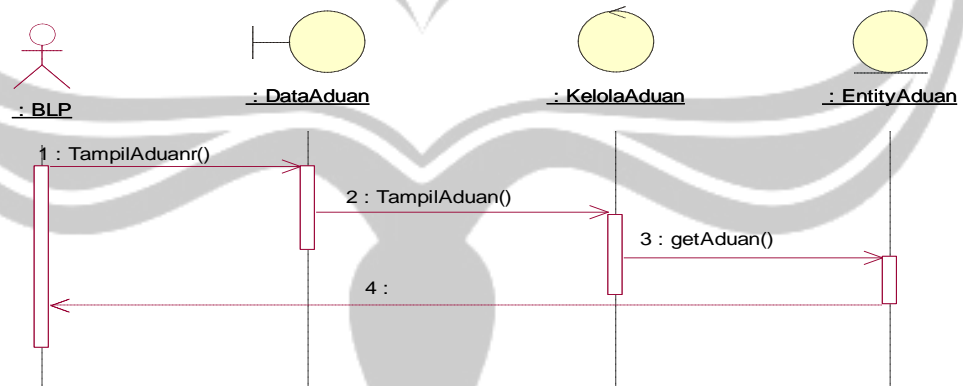
Gambar 2.18 *Sequence Diagram* : Tambah Data Penugasan Juru Parkir

2.2.1.18. Fungsi Tampil Data Penugasan Juru Parkir



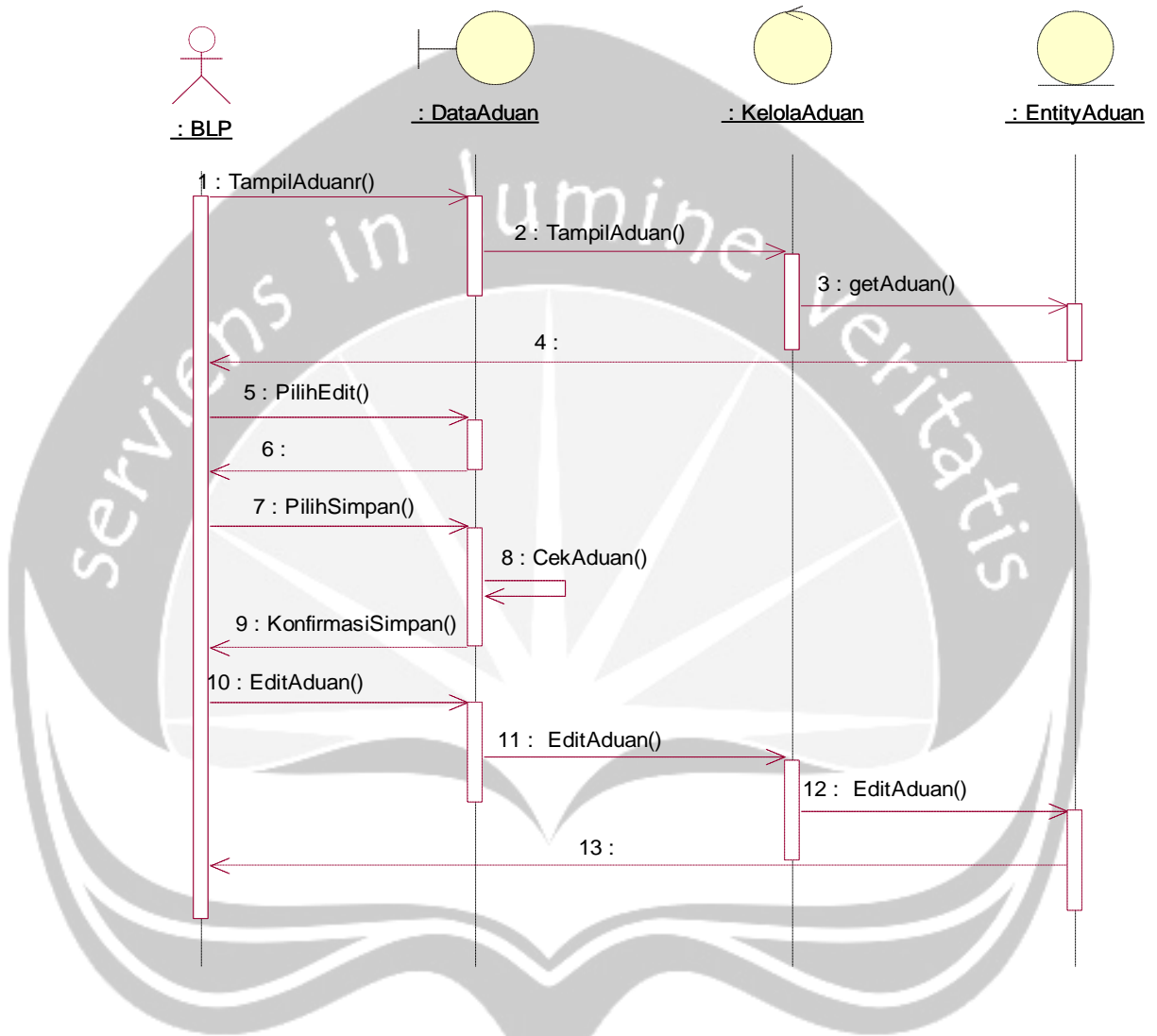
Gambar 2.19 *Sequence Diagram* : Tampil Data Penugasan Juru Parkir

2.2.1.19. Fungsi Tampil Data Aduan Masyarakat



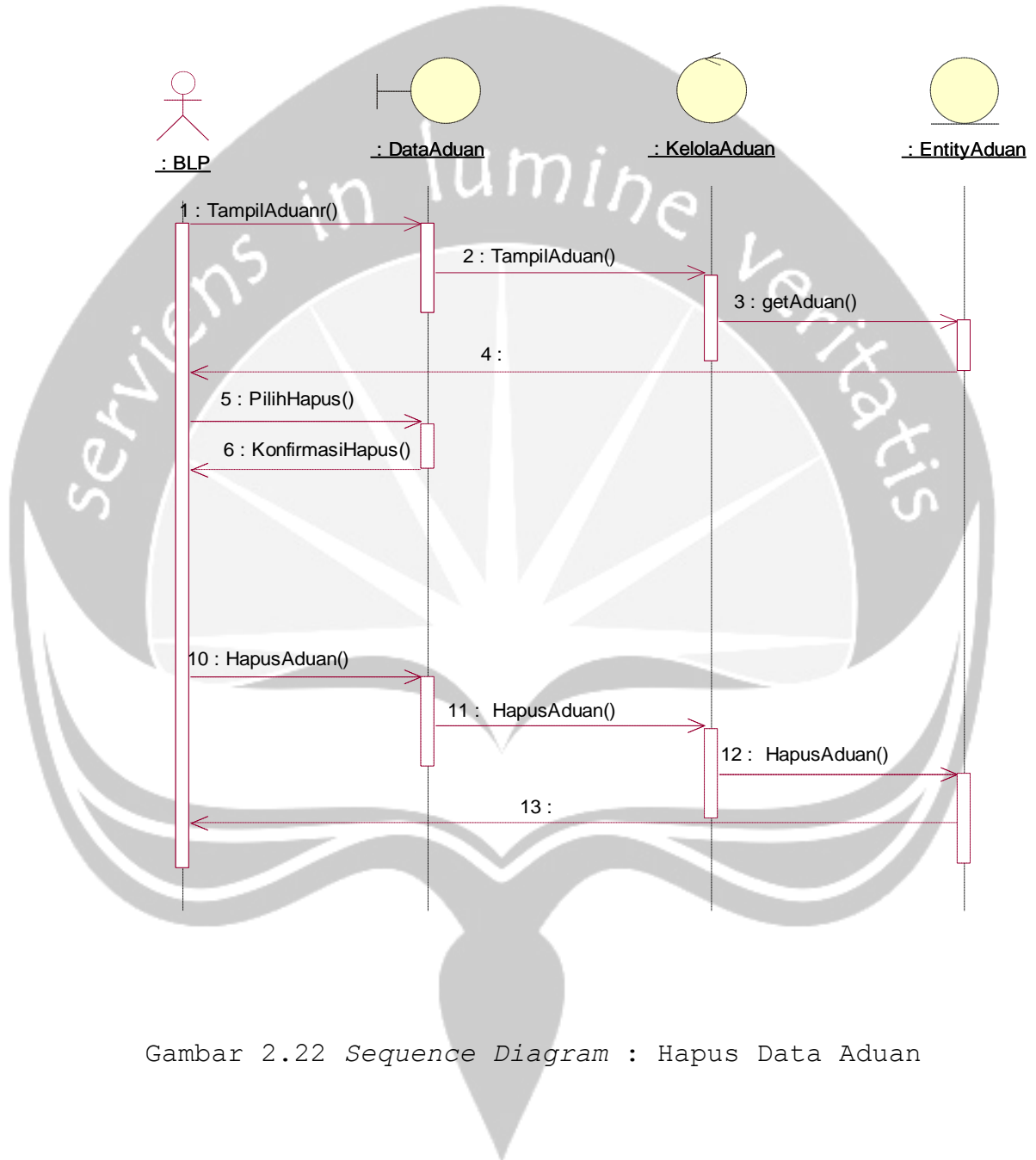
Gambar 2.20 *Sequence Diagram* : Tampil Data Aduan

2.2.1.20. Fungsi *Edit* Data Aduan Masyarakat



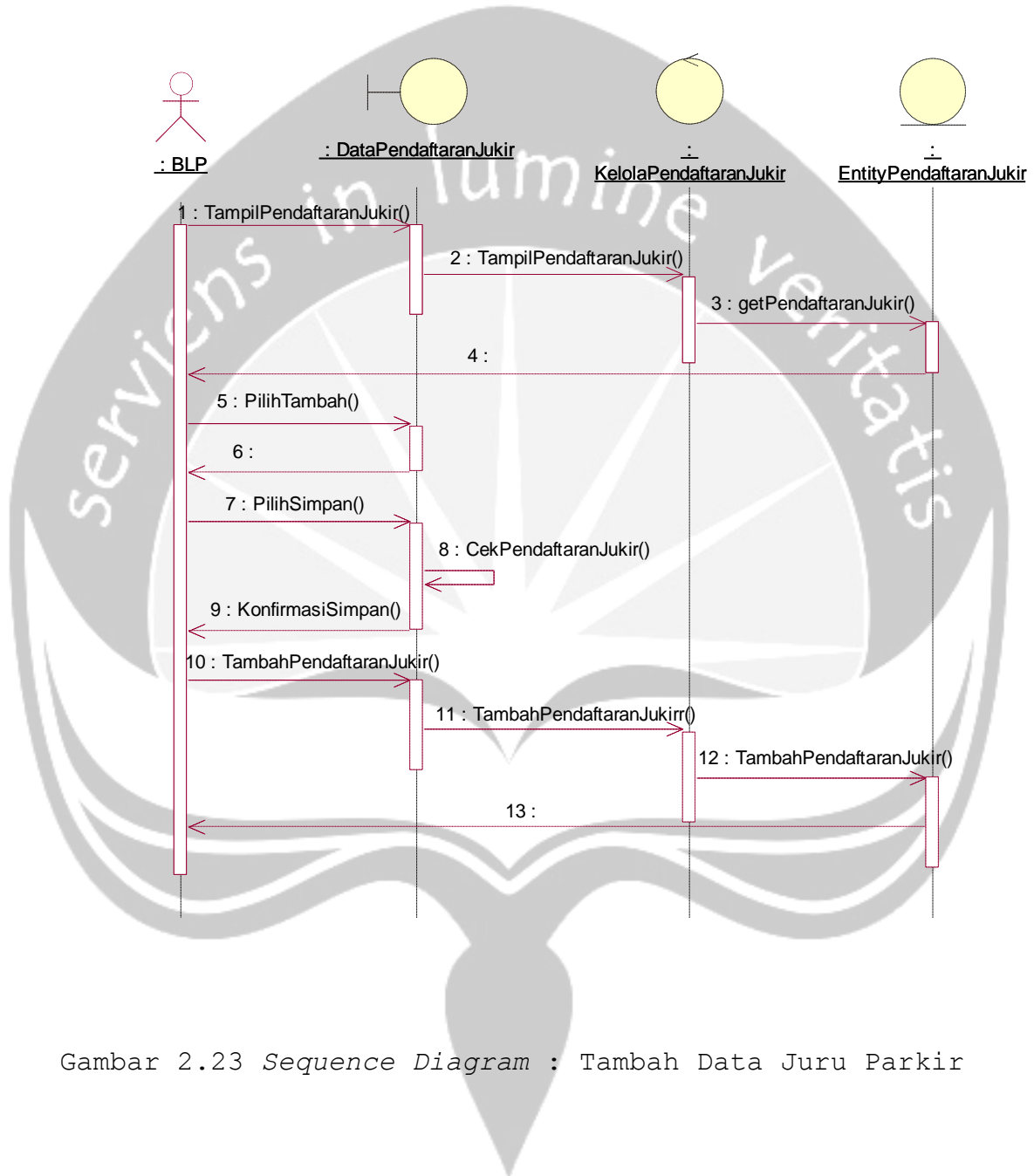
Gambar 2.21 *Sequence Diagram* : Ubah Data Aduan

2.2.1.21. Fungsi Hapus Data Aduan Masyarakat



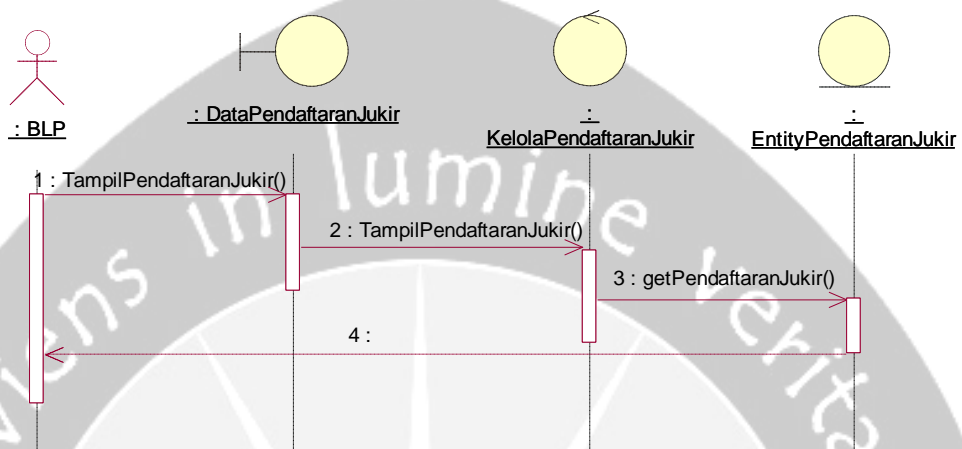
Gambar 2.22 *Sequence Diagram* : Hapus Data Aduan

2.2.1.22. Fungsi Tambah Data Pendaftaran Juru Parkir



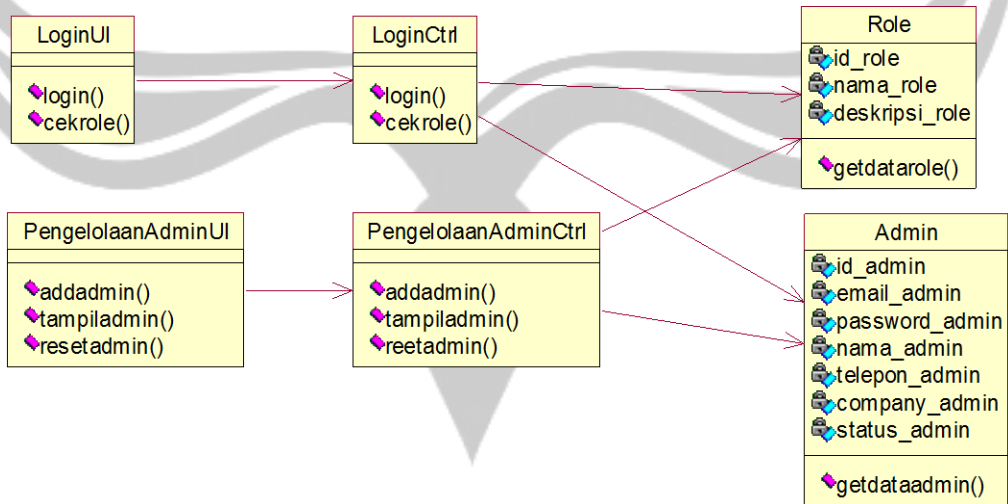
Gambar 2.23 *Sequence Diagram* : Tambah Data Juru Parkir

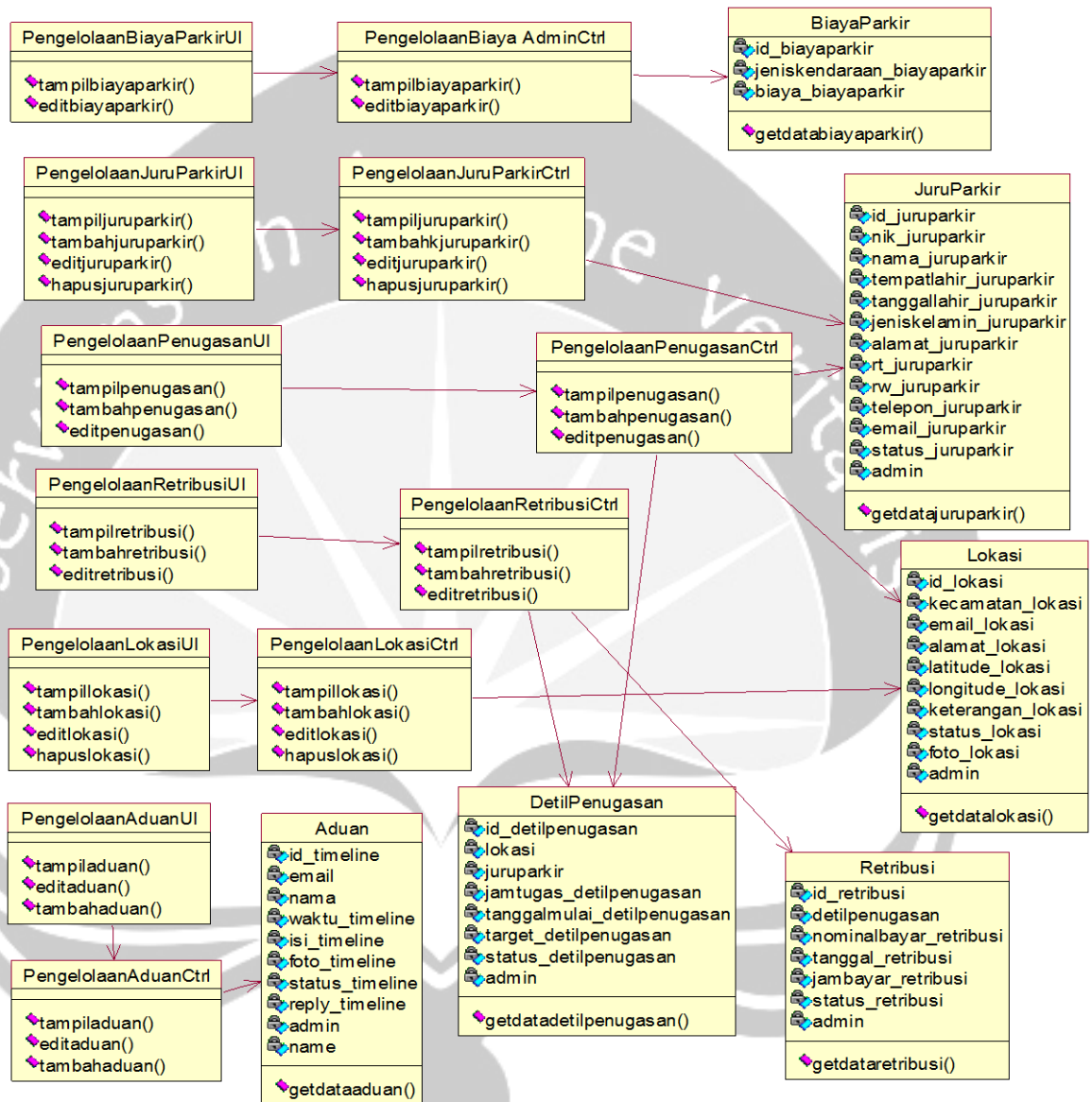
2.2.1.23. Fungsi Tampil Data Pendaftaran Juru Parkir



Gambar 2.24 *Sequence Diagram* : Tampil Data Juru Parkir

2.2.2. *Class Diagram*





2.25 Class Diagram SIMPARKIR

2.2.3. Class Diagram Spesific Description

2.2.3.1. Specific Design Class DataAdminUI

DataAdminUI	<<boundary>>
<pre>+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.</pre>	

2.2.3.2. Specific Design Class PengelolaanBiayaParkirUI

PengelolaanBiayaParkirUI	<<boundary>>
<pre>+Databiaya parkirUI () Default konstruktor untuk menginisialisasi semua atribut dari kelas ini. +Editbiaya parkir() Operasi untuk menerima perintah pengubahan data biaya parkir yang ada di basis data.</pre>	

+TampilDataKuliner()

Operasi untuk menampilkan semua data biaya parkir yang ada di basis data.

2.2.3.3. *Specific Design Class* PengelolaanJuruParkirUI

PengelolaanJuruParkirUI	<<boundary>>
<p>+Datapengelolaanjuruparkir() Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.</p> <p>+Entrijuruparkir() Operasi untuk menerima perintah penambahan data juru parkir.</p> <p>+Editjuruparkir() Operasi untuk menerima perintah pengubahan data juru parkir yang ada di basis data.</p> <p>+Hapusjuruparkir() Operasi untuk menerima perintah penghapusan data juru parkir yang ada di basis data.</p> <p>+Tampiljuruparkir() Operasi untuk menampilkan semua data juru parkir yang ada di basis data.</p>	

2.2.3.4. *Specific Design Class* PengelolaanPenugasanUI

PengelolaanPenugasanUI	<<boundary>>
<pre>+Datapengelolaanpenugasan() Default konstruktor untuk menginisialisasi semua atribut dari kelas ini. +Entridatapenugasan() Operasi untuk menerima perintah penambahan data penugasan. +Editdatapenugasan() Operasi untuk menerima perintah pengubahan data penugasan yang ada di basis data. +Tampilpenugasan() Operasi untuk menampilkan semua data penugasan yang ada di basis data.</pre>	

2.2.3.5. *Specific Design Class* PengelolaanRetribusiUI

PengelolaanRetribusiUI	<<boundary>>
<pre> +Datapengelolaanretribusi() Default konstruktor untuk menginisialisasi semua atribut dari kelas ini. +Entridataretribusi() Operasi untuk menerima perintah penambahan data retribusi. +Editdataretribusi() Operasi untuk menerima perintah pengubahan data retribusi yang ada di basis data. +Tampilretribusi() Operasi untuk menampilkan semua data retribusi yang ada di basis data. </pre>	

2.2.3.6. *Specific Design Class* PengelolaanLokasiUI

PengelolaanLokasiUI	<<boundary>>
<pre> +Datapengelolaanlokasi() Default konstruktor untuk menginisialisasi semua atribut dari kelas ini. </pre>	

+Entrilokasi()

Operasi untuk menerima perintah penambahan data lokasi.

+Editlokasi()

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

+Hapuslokasi()

Operasi untuk menerima perintah penghapusan data lokasi yang ada di basis data.

+Tampillokasi()

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

2.2.3.7. *Specific Design Class* PengelolaanAduanUI

PengelolaanAduanUI	<<boundary>>
+Datapenngelolaanaduan()	
Default konstruktor untuk menginisialisasi semua atribut dari kelas ini.	
+Entriaduan()	
Operasi untuk menerima perintah penambahan data aduan.	
+Editaduan()	

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

+Tampiladuan()

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

2.2.3.8. *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.	

2.2.3.9. Specific Design Class LoginCtrl

DataAdminCtrl	<<control>>
<pre>+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.</pre>	

2.2.3.10. Specific Design Class PengelolaanAdminCtrl

PengelolaanAdminCtrl	<<control>>
<pre>+EntriAdmin() Operasi ini digunakan untuk menambahkan data admin. +Resetdmin() Operasi ini digunakan untuk mengubah data admin. +TampilDataadmin()</pre>	

Operasi ini digunakan untuk menampilkan semua data admin.

2.2.3.11. *Specific Design Class* PengelolaanBiayaParkirCtrl

PengelolaanBiayaParkirCtrl	<<control>>
+Editbiayaparkir() Operasi ini digunakan untuk mengubah data biaya parkir.	
+Tampilbiayaparkir() Operasi ini digunakan untuk menampilkan semua data biaya parkir.	

2.2.3.12. *Specific Design Class* PengelolaanJuruParkirCtrl

PengelolaanJuruParkirCtrl	<<control>>
+Entrijuruparkir() Operasi ini digunakan untuk menambahkan data juru parkir.	
+Editjuruparkir() Operasi ini digunakan untuk mengubah data juru parkir.	
+Hapusjuruparkir()	

Operasi ini digunakan untuk menghapus data juru parkir.
 +Tampiljruparkir()
 Operasi ini digunakan untuk menampilkan semua data juru parkir.

2.2.3.13. Specific Design Class PengelolaanPenugasanCtrl

PengelolaanPenugasanCtrl	<<control>>
+Entripenugasan() Operasi ini digunakan untuk menambahkan data penugasan. +Editpenugasan() Operasi ini digunakan untuk mengubah data penugasan. +Tampilpenugasan() Operasi ini digunakan untuk menampilkan semua data penugasan.	

2.2.3.14. Specific Design Class PengelolaanRetribusiCtrl

PengelolaanRetribusiCtrl	<<control>>
+Entriretribusi() Operasi ini digunakan untuk menambahkan data retribusi.	

<p>+Editretribusi()</p> <p>Operasi ini digunakan untuk mengubah data retribusi.</p> <p>+Tampilretribusi()</p> <p>Operasi ini digunakan untuk menampilkan semua data retribusi.</p>
--

2.2.3.15. *Specific Design Class* PengelolaanLokasiCtrl

PengelolaanLokasiCtrl	<<control>>
<p>+Entrilokasi()</p> <p>Operasi ini digunakan untuk menambahkan data lokasi.</p> <p>+Editlokasi()</p> <p>Operasi ini digunakan untuk mengubah data lokasi</p> <p>+Hapuslokasi()</p> <p>Operasi ini digunakan untuk menghapus data lokasi</p> <p>+Tampillokasi()</p> <p>Operasi ini digunakan untuk menampilkan semua data lokasi.</p>	

2.2.3.16. *Specific Design Class* PengelolaanAduanCtrl

PengelolaanAduanCtrl	<<control>>
-----------------------------	--------------------------------

<p>+Entriaduan() Operasi ini digunakan untuk menambahkan data aduan.</p> <p>+Editaduan() Operasi ini digunakan untuk mengubah data aduan.</p> <p>+Tampiladuan() Operasi ini digunakan untuk menampilkan semua data aduan.</p>

2.2.3.17. *Specific Design Class Role*

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

2.2.3.18. *Specific Design Class Admin*

Role	<<entity>>
<p>-id_admin : integer(11) Atribut ini digunakan untuk menyimpan data id admin.</p> <p>-email_admin : varchar(255) Atribut ini digunakan untuk menyimpan data email admin.</p> <p>-password_admin : varchar(255) Atribut ini digunakan untuk menyimpan data password admin.</p> <p>-nama_admin : varchar(255) Atribut ini digunakan untuk menyimpan data nama admin.</p> <p>-telepon_admin : integer(11) Atribut ini digunakan untuk menyimpan data telepon admin.</p> <p>-company_admin : varchar(255) Atribut ini digunakan untuk menyimpan data company admin.</p> <p>-status_admin : varchar(255) Atribut ini digunakan untuk menyimpan data status admin.</p>	
<p>+getdataadmin() Operasi untuk mengambil data admin.</p>	

2.2.3.19. Specific Design Class Biaya Parkir

Biaya Parkir	<<entity>>
<p>-id_biayaparkir : integer(11) Atribut ini digunakan untuk menyimpan data id biaya parkir.</p> <p>-jeniskendaraan_biayaparkir : varchar(255) Atribut ini digunakan untuk menyimpan data jenis kendaraan biaya parkir.</p> <p>-biaya_biaya parkir : varchar(255) Atribut ini digunakan untuk menyimpan data biaya parkir.</p>	
<p>+getdatabiayaparkir() Operasi untuk mengambil data biaya parkir.</p>	

2.2.3.20. Specific Design Class Juru Parkir

Juru Parkir	<<entity>>
<p>-id_juruparkir : integer(11) Atribut ini digunakan untuk menyimpan data id juru parkir.</p> <p>-nik_juru parkir : integer(11) Atribut ini digunakan untuk menyimpan data nik juru parkir.</p> <p>-nama_juruparkir : varchar(255)</p>	

Atribut ini digunakan untuk menyimpan data nama juru parkir.

-tempatlahir_juruparkir : varchar(255)

Atribut ini digunakan untuk menyimpan data tempat lahir juru parkir.

-tanggalahir_juruparkir : integer(11)

Atribut ini digunakan untuk menyimpan data tanggal juru parkir.

-jeniskelamin_juruparkir : varchar(255)

Atribut ini digunakan untuk menyimpan data jenis kelamin juru parkir.

-alamat_juruparkir : varchar(255)

Atribut ini digunakan untuk menyimpan data alamat juru parkir.

-rt_juruparkir : integer(11)

Atribut ini digunakan untuk menyimpan data rt juru parkir.

-rw_juruparkir : integer(11)

Atribut ini digunakan untuk menyimpan data rw juru parkir.

-telepon_juruparkir : integer(11)

Atribut ini digunakan untuk menyimpan data telepon juru parkir.

-email_juruparkir : varchar(255)

Atribut ini digunakan untuk menyimpan data email juru parkir.

-status_juruparkir : varchar(255)

Atribut ini digunakan untuk menyimpan data status juru parkir.

+getdatajuruparkir()

Operasi untuk mengambil data juru parkir.

2.2.3.21. *Specific Design Class Lokasi*

Lokasi	<<entity>>
<p>-id_lokasi : integer(11)</p> <p>Atribut ini digunakan untuk menyimpan data id lokasi.</p> <p>-kecamatan_lokasi : varchar(255)</p> <p>Atribut ini digunakan untuk menyimpan data lokasi kecamatan.</p> <p>-email_lokasi : varchar(255)</p> <p>Atribut ini digunakan untuk menyimpan data email lokasi.</p> <p>-alamat_lokasi : varchar(255)</p>	

Atribut ini digunakan untuk menyimpan data alamat lokasi.

-latitude_lokasi : double

Atribut ini digunakan untuk menyimpan data latitude lokasi.

-longitude_lokasi : double

Atribut ini digunakan untuk menyimpan data longitude lokasi.

-keterangan_lokasi : varchar(255)

Atribut ini digunakan untuk menyimpan data keterangan lokasi.

-status_lokasi : varchar(255)

Atribut ini digunakan untuk menyimpan data status lokasi.

-foto_lokasi : varchar(255)

Atribut ini digunakan untuk menyimpan data foto lokasi.

+getdata_lokasi()

Operasi untuk mengambil data lokasi.

2.2.3.22. *Specific Design Class Retribusi*

Retribusi	<<entity>>
-id_retribusi : integer(11)	

Atribut ini digunakan untuk menyimpan data id retribusi.

-detilpenugasan : varchar(255)

Atribut ini digunakan untuk menyimpan data detil penugasan.

-nominalbayar_retribusi : integer(11)

Atribut ini digunakan untuk menyimpan data nomil bayar retribusi.

-tanggal_retribusi : date

Atribut ini digunakan untuk menyimpan data tanggal retribusi.

-jam_retribusi : time

Atribut ini digunakan untuk menyimpan data jam retribusi.

-status_retribusi : varchar(255)

Atribut ini digunakan untuk menyimpan data status retribusi.

+getdataretribusi()

Operasi untuk mengambil data retribusi.

2.2.3.23. *Specific Design Class Detil Penugasan*

Detil Penugasan	<<entity>>
-id_detilpenugasan: integer(11)	

Atribut ini digunakan untuk menyimpan data id detil penugasan.

-lokasi : varchar(255)

Atribut ini digunakan untuk menyimpan data lokasi.

-juruparkir : varchar(255)

Atribut ini digunakan untuk menyimpan data juru parkir.

-jamugas_detilpenugasan : time

Atribut ini digunakan untuk menyimpan data jam tugas penugasan.

-tanggalmulai_detilpenugasan : date

Atribut ini digunakan untuk menyimpan data hari mulai penugasan.

-target_detilpenugasan : integer(11)

Atribut ini digunakan untuk menyimpan data target penugasan.

-status_detilpenugasan : varchar(255)

Atribut ini digunakan untuk menyimpan data status penugasan.

+getdetilpenugasan()

Operasi untuk mengambil data penugasan.

2.2.3.24. Specific Design Class Aduan

Aduan	<<entity>>
<p>-id_timeline : integer(11) Atribut ini digunakan untuk menyimpan data id timeline.</p> <p>-email : varchar(255) Atribut ini digunakan untuk menyimpan data email aduan.</p> <p>-nama : varchar(255) Atribut ini digunakan untuk menyimpan data nama aduan.</p> <p>-waktu_timeline : time Atribut ini digunakan untuk menyimpan data waktu aduan.</p> <p>-isi_timeline : varchar(255) Atribut ini digunakan untuk menyimpan data isi aduan.</p> <p>-foto_timeline : varchar(255) Atribut ini digunakan untuk menyimpan data foto timeline.</p> <p>-status_timeline : varchar(255) Atribut ini digunakan untuk menyimpan data status aduan.</p> <p>-reply_timeline : varchar(255) Atribut ini digunakan untuk menyimpan data balas timeline.</p>	
<p>+getdataaduan() Operasi untuk mengambil data aduan.</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 JuruParkir			
Nama	Tipe	Panjang	Keterangan
Id_juruparkir	Integer	11	Id juru parkir, primary key
Nik_juruparkir	Integer	11	NIK juru parkir
Nama_juruparkir	Variable Character	11	Nama juru parkir
Tempatlahir_juruparkir	Variable Character	255	Tempat lahir juru parkir
Tanggallahir_juruparkir	Date		Tanggal lahir juru parkir
Jeniskelamin_juruparkir	Variable Character	255	Jenis kelamin juru parkir
Alamat_juruparkir	Variable Character	255	Alamat juru parkir
RT_juruparkir	Integer	11	RT
RW_juruparkir	Integer	11	RW
Telepon_juruparkir	Integer	11	Telepon juru parkir
Email_juruparkir	Variable Character	255	Email juru parkir
Status_juruparkir	Variable Character	255	Status juru parkir
Admin	Integer	11	Admin yang sedang bertugas, foreign key

3.1.4. Deskripsi Entitas Lokasi			
Nama	Tipe	Panjang	Keterangan

Id_lokasi	Integer	11	Id lokasi parkir, primary key
Kecamatan_lokasi	Variable Character	255	Kecamatan
Alamat_lokasi	Variable Character	255	Alamat lokasi parkir
Latitude_lokasi	Double		Latitude
Longitude_lokasi	Double		Longitude
Keterangan_lokasi	Variable Character	255	Keterangan lokasi parkir
Status_lokasi	Variable Character	255	Status lokasi parkir
Foto_lokasi	Variable Character	255	Foto lokasi parkir
Admin	Integer	11	Admin yang sedang bertugas, foreign key

3.1.5. Deskripsi Eentitas DetilPenugasan			
Nama	Tipe	Panjang	Keterangan
Id_detilpenugasan	Integer	11	Id detil penugasan, primary key
Lokasi_detilpenugasan	Integer	11	Lokasi detil penugasan
Juruparkir_detilpenugasan	Integer	11	Juruparkir
Jamtugas_detilpenugasan	Datetime		Jam tugas pada penugasan

Tanggalmulai_detilpenugasan	Date		Tanggal mulai penugasan
Target_detilpenugasan	Integer	11	Target penugasan
Status_detilpenugasan	Variable Character	255	Status penugasan
Admin	Integer	11	Admin yang sedang bertugas, foreign key

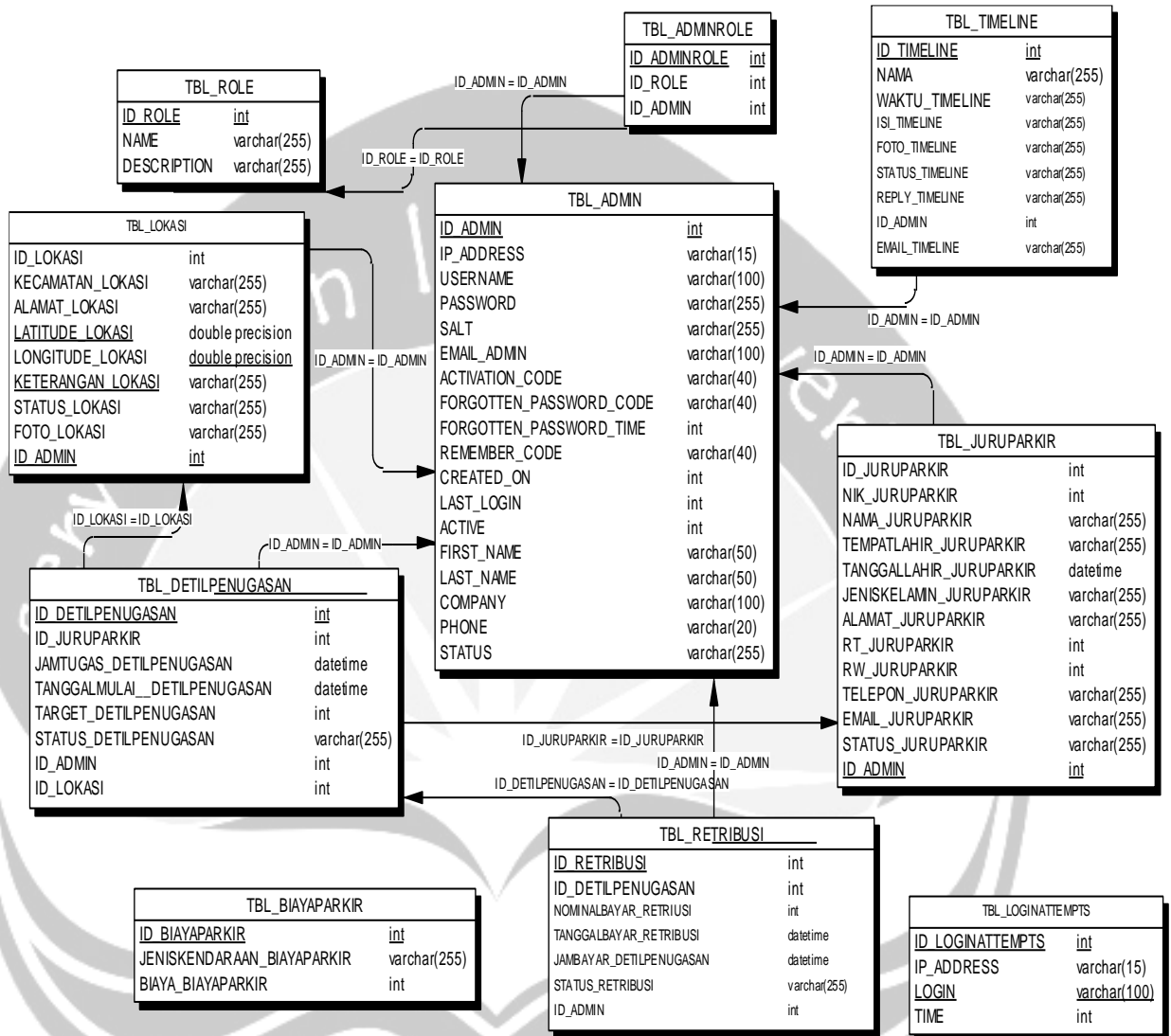
3.1.6. Deskripsi Entitas Retribusi			
Nama	Tipe	Panjang	Keterangan
Id_retriisi	Integer	11	Id reribusi
Detilpenugasan	Integer	11	Detil penugasan, foreign key
Nominalbayar_retriisi	Integer	11	Nominal bayar
Tanggal_retriisi	Date		Tanggal bayar
Jambayar_retriisi	Datetime		Jam bayar
Status_retriisi	Variable Character	255	Status bayar
Admin	Integer	11	Admin yang sedang bertugas, foreign key

3.1.7. Deskripsi Entitas Biayaparkir			
Nama	Tipe	Panjang	Keterangan
Id_biayaparkir	Integer	11	Id biaya parkir, primary key

Jeniskendaraan_ biayaparkir	Variable Character	255	Jenis kendaraan yang parkir
Biaya_ biayaparkir	Integer	11	Biaya parkir

3.1.8. 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

2.3 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 parkir yang disediakan oleh Dinas Perhubungan Kabupaten Purbalingga untuk masyarakat. Pengguna dapat melihat layanan yang tersedia, seperti pendapatan asli daerah, daftar juru parkir, download aplikasi *mobile* SIMPARKIR dan melihat maupun menulis aduan masyarakat.

4.2 Antarmuka Halaman Layanan

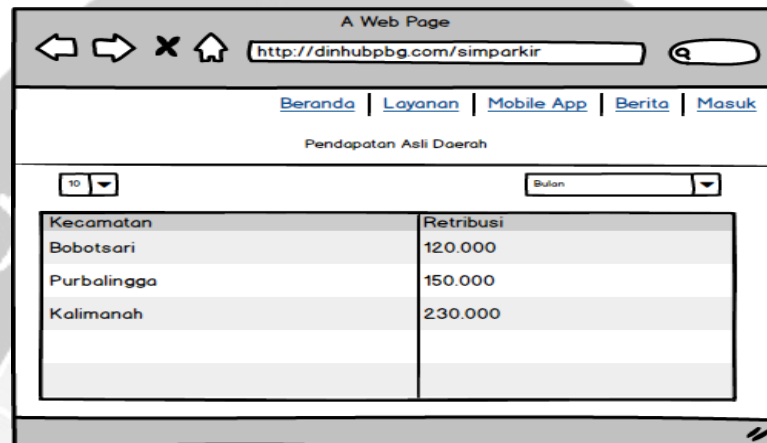


Gambar 4.2 Antarmuka Layanan

Antarmuka 4.2 digunakan untuk menampilkan halaman layanan parkir yang disediakan oleh Dinas Perhubungan Kabupaten Purbalingga untuk masyarakat. Pengguna dapat

melihat layanan yang tersedia, seperti pendapatan asli daerah, juru parkir dan daftar juru parkir.

4.3 Antarmuka Halaman PAD

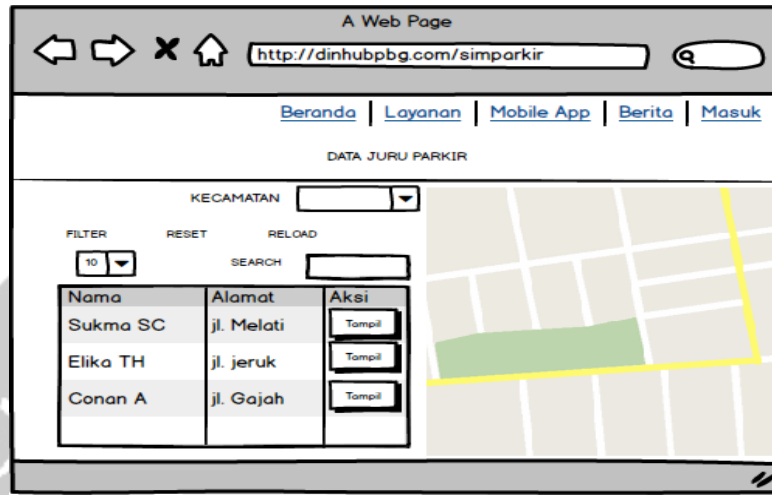


Kecamatan	Retribusi
Bobotsari	120.000
Purbalingga	150.000
Kalimanah	230.000

Gambar 4.3 Antarmuka Halaman PAD

Antarmuka 4.3 digunakan untuk menampilkan halaman PAD yang disediakan oleh Dinas Perhubungan Kabupaten Purbalingga untuk masyarakat. Pengguna dapat melihat pendapatan asli daerah parkir Kabupaten Purbalingga serta pengguna dapat melakukan pencarian PAD berdasarkan bulan.

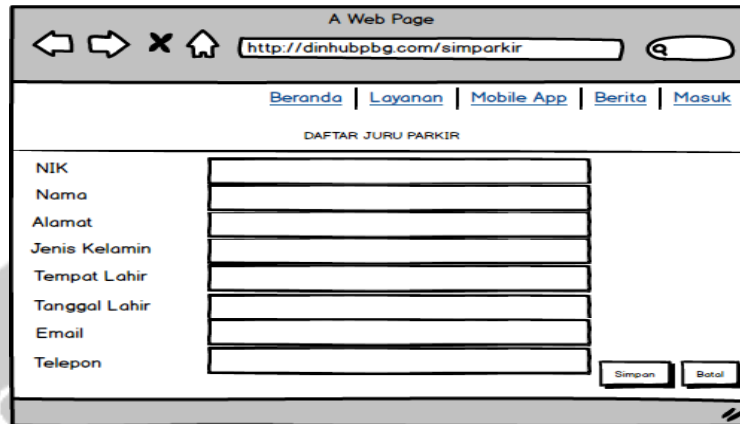
4.4 Antarmuka Halaman Juru Parkir



Gambar 4.4 Antarmuka Halaman Juru Parkir

Antarmuka 4.4 digunakan untuk menampilkan halaman juru parkir yang disediakan oleh Dinas Perhubungan Kabupaten Purbalingga untuk masyarakat. Pengguna dapat melihat data juru parkir yang terdapat di Kabupaten Purbalingga yang sudah terdaftar, pengguna dapat melakukan pencarian data juru parkir per kecamatan yang ada di Kabupaten Purbalingga serta melihat lokasi parkir dari juru parkir pada peta.

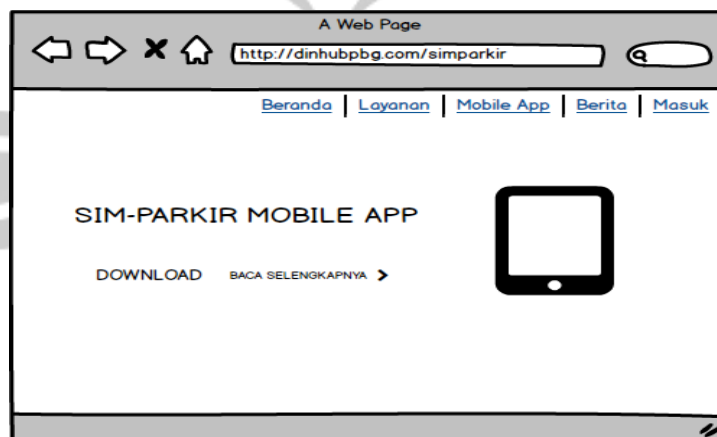
4.5 Antarmuka Halaman Pendaftaran Juru Parkir



Gambar 4.5 Antarmuka Halaman Pendaftaran Juru Parkir

Antarmuka 4.5 digunakan untuk menampilkan halaman pendaftaran juru parkir yang disediakan oleh Dinas Perhubungan Kabupaten Purbalingga untuk masyarakat. Pengguna dapat melakukan pendaftaran sebagai juru parkir dengan mengisi formulir yang sudah disediakan.

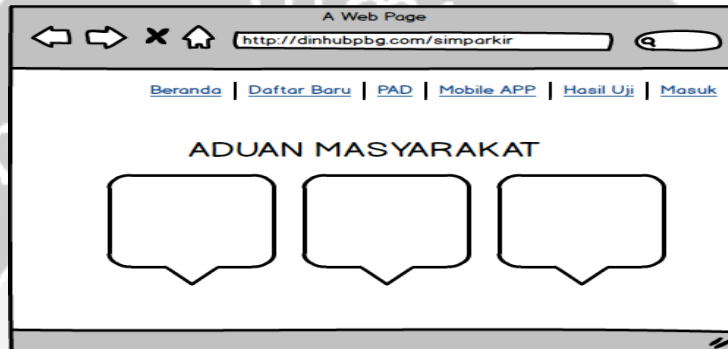
4.6 Antarmuka Halaman Aplikasi Mobile



Gambar 4.6 Antarmuka Aplikasi Mobile

Antarmuka 4.6 digunakan untuk menampilkan halaman aplikasi SIMPARKIR *mobile*. Pengguna dapat melihat informasi terkait SIMPARKIR *mobile*.

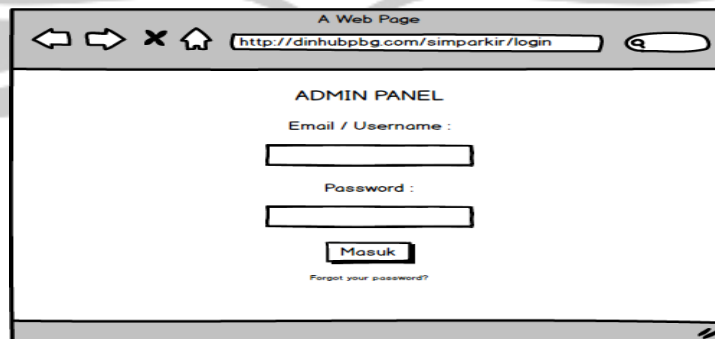
4.7 Antarmuka Halaman Aduan Masyarakat



Gambar 4.7 Antarmuka Halaman Aduan Masyarakat

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

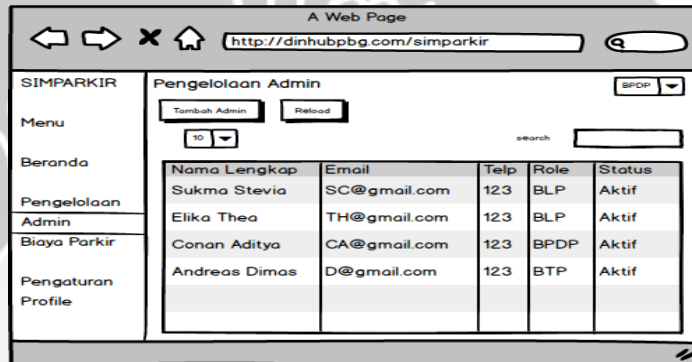
4.8 Antarmuka Halaman Login Admin



Gambar 4.8 Antarmuka Halaman Login Admin

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

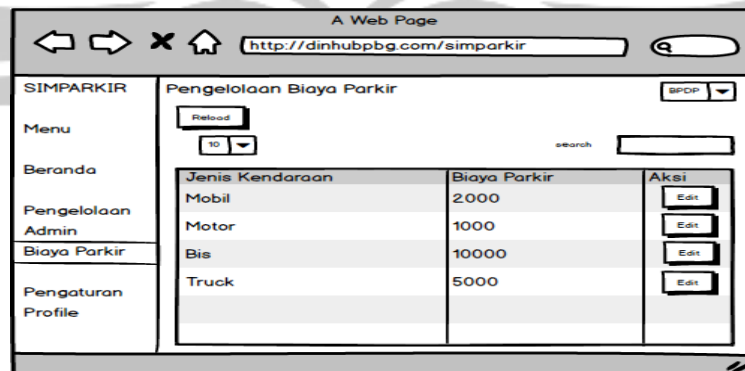
4.9 Antarmuka Halaman Pengelolaan Admin



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.

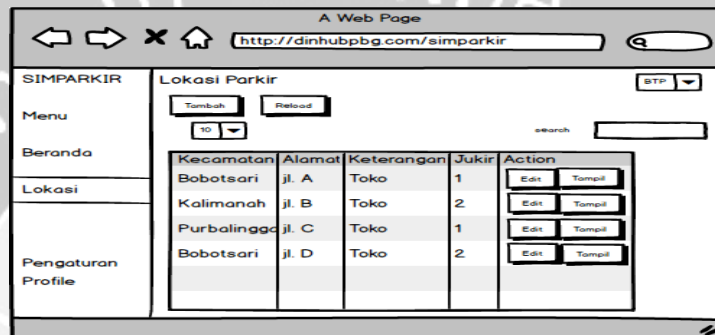
4.10 Antarmuka Halaman Pengelolaan Biaya Parkir



Gambar 4.10 Antarmuka Halaman Pengelolaan Biaya Parkir

Antarmuka 4.10 digunakan untuk menampilkan halaman Pengelolaan biaya parkir. BPPD selaku aktor dapat melihat data parkir, melakukan pencarian data biaya parkir serta melakukan perubahan biaya parkir yang ada.

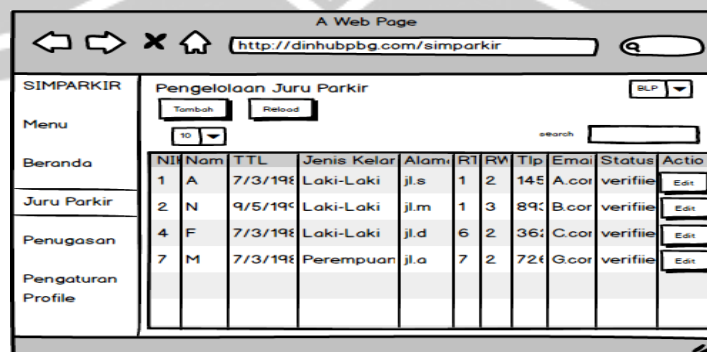
4.11 Antarmuka Halaman Pengelolaan Lokasi Parkir



Gambar 4.11 Antarmuka Halaman Pengelolaan Lokasi Parkir

Antarmuka 4.11 digunakan untuk menampilkan halaman Pengelolaan lokasi parkir. BPPD selaku aktor dapat melihat data lokasi parkir, melakukan pencarian data lokasi parkir serta melakukan perubahan data lokasi parkir yang ada.

4.12 Antarmuka Halaman Pengelolaan Juru Parkir



Gambar 4.12 Antarmuka Halaman Pengelolaan Juru Parkir

Antarmuka 4.12 digunakan untuk menampilkan halaman Pengelolaan juru parkir. BLP selaku aktor dapat menambah atau mendaftarkan calon juru parkir, melihat data juru parkir, melakukan pencarian data juru parkir serta melakukan perubahan data juru parkir yang ada.

4.13 Antarmuka Halaman Pengelolaan Penugasan Juru Parkir

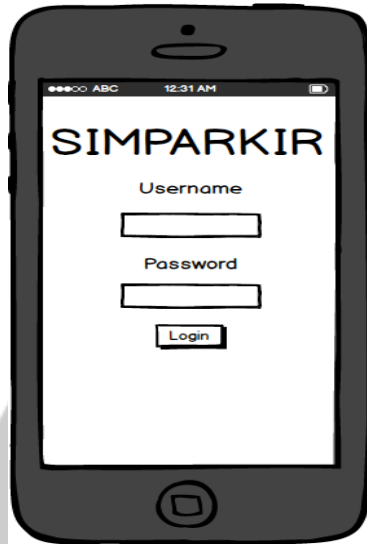
NIK	Nama	TTL	Jenis Kelamin	Alamat	RT	RW	Tip	Email
1	A	7/3/1980	Laki-Laki	jl.s	1	2	145	A.com
2	N	9/5/1990	Laki-Laki	jl.m	1	3	893	B.com

Alamat	Keterangan	Aksi
jl. A	Toko	Pilih
jl. B	Toko	Pilih
jl. M	Toko	Pilih

Gambar 4.13 Antarmuka Halaman Pengelolaan Penugasan Juru Parkir

Antarmuka 4.13 digunakan untuk menampilkan halaman Pengelolaan penugasan juru parkir. BLP selaku aktor dapat melakukan verifikasi terhadap calon juru parkir yang belum terverifikasi dengan mencari NIK calon juru parkir, melakukan perubahan data juru parkir, melihat riwayat penugasan juru parkir, menambah penugasan juru parkir 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 Lokasi*



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

Gambar 4.15 Antarmuka Halaman *Pengelolaan Lokasi*

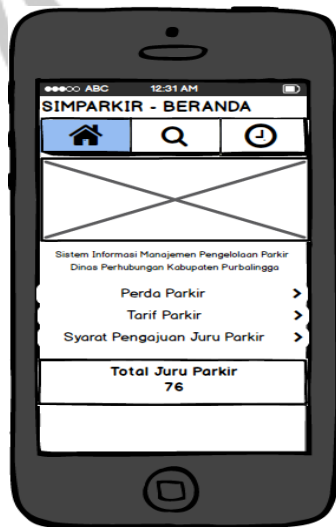
4.16 Antarmuka Halaman Pengelolaan Verifikasi Lokasi



Antarmuka 4.16 digunakan untuk menampilkan halaman Pengelolaan verifikasi lokasi. BTP selaku aktor dapat melakukan proses verifikasi lokasi yang sudah dipilih dari halaman sebelumnya dengan mendatangi lokasi yang sudah terdaftar kemudian mengisi data kecamatan, alamat, keterangan ditempat lokasi yang telah disediakan serta menambahkan dokumentasi foto lokasi parkir kemudian melakukan konfirmasi data yang telah diisi.

Gambar 4.16 Antarmuka Halaman Pengelolaan Verifikasi Lokasi

4.17 Antarmuka Halaman Beranda

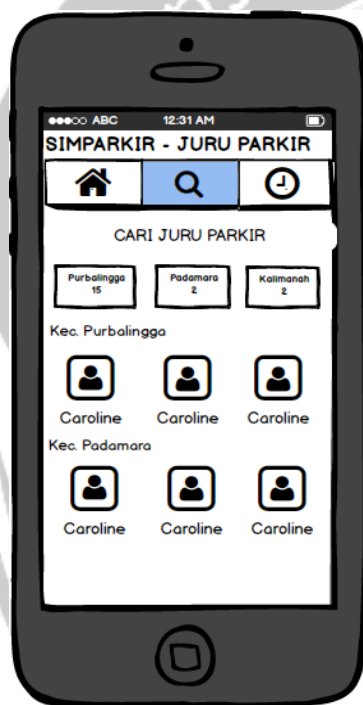


Antarmuka 4.17 digunakan untuk menampilkan halaman utama beranda *mobile* SIMPAKIR. Pengguna dapat melihat layanan yang disediakan Dinas Perhubungan Kabupaten Purbalingga. Pada halaman utama ini pengguna dapat melihat informasi peraturan daerah Kabupaten Purbalingga yang mengatur tentang parkir di Kabupaten Purbalingga, tarif parkir yang digunakan atau yang berlaku

Gambar 4.17 Antarmuka Halaman Mobile

serta pengguna dapat melihat syarat guna mengajukan sebagai juru parkir, daftar juru parkir, dan pelaporann PAD. Pengguna juga dapat melihat jumlah total juru parkir yang terdaftar pada Kabupaten Purbalingga.

4.18 Antarmuka Halaman Juru Parkir



Antarmuka 4.18 digunakan untuk menampilkan halaman juru parkir SIMPARKIR. Pengguna (juru parkir dan masyarakat umum) dapat melihat layanan yang disediakan Dinas Perhubungan Kabupaten Purbalingga mengenai juru parkir. Pada juru parkir ini pengguna (juru parkir) dapat melihat detail informasinya termasuk biaya setoran, dengan menekan tombol cari juru parkir kemudian memasukkan no tugas juru parkir. Masyarakat dapat melihat detail juru parkir dan jumlah juru parkir per kecamatan serta lokasi parkir yang tersedia.

Gambar 4.18 Antarmuka Halaman Juru Parkir

4.19 Antarmuka Halaman *Timeline*



Antarmuka 4.19 digunakan untuk menampilkan halaman *timeline* SIMPARKIR. 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.

Gambar 4.19 Antarmuka Halaman *Timeline*