

Bab VI

Penutup

Pada bab IV dijabarkan kesimpulan dan saran yang didapatkan selama proses pembangunan aplikasi tugas akhir.

6.1. Kesimpulan

Berdasarkan pembahasan yang telah dilakukan dari bab-bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

Pembangunan aplikasi mobile unit donor darah berhasil dibangun terbukti dari hasil pengujian terhadap pengguna.

6.2. Saran

Berdasarkan proses pembangunan yang telah dilakukan hingga proses pengujian, penulis mendapat beberapa saran untuk pengembangan lebih terhadap aplikasi PMI yang telah dibuat sebagai berikut:

1. Pengembangan *multi platform*.
2. Sistem pelaporan di *backend* bisa dikembangkan lebih banyak.

DAFTAR PUSTAKA

Android Authority, T., 2017. *Android Authority*. [Online] Available at: <http://www.androidauthority.com/what-is-android-328076/>

[Accessed 18 Juni 2017].

Bhardwaj , S., Chouhan , P., Sharma , P. & Sharma , R., 2013. Android Operating Systems. *International Journal of Engineering Technology & Management Research* , 1(1), p. 147.

Bharthan , A. & Bharathan , D., 2014. RelationalJSON, An Enriched Method to Store and Query JSON Records. *International Journal of Computer Applications*, 98(7), p. 3.

DepKes, 2016. *Kementrian Kesehatan Republik Indonesia*. [Online]

Available at: <http://www.depkes.go.id/article/print/16060300001/ketersediaan-darah-ditentukan-partisipasi-masyarakat-menjadi-donor.html>

[Accessed 26 7 2017].

Developer, A., n.d. *Android Studio*. [Online] Available at:

<https://developer.android.com/studio/intro/index.html?hl=id>

[Accessed 18 6 2017].

eMarketer, 2014. *eMarketer*. [Online] Available at: <https://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694>

[Accessed 19 Juni 2017].

Gatot, A. B., 2016. *Permasalahan Darah di PMI Kota Yogyakarta* [Interview] (7 11 2016).

Hamzah, 2015. Analisa dan Rancangan Sistem Informasi Ketersediaan Donor Darah Hidup. *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(2), p. 15.

Jain, A., Nirmal, A., Sapre, N. & Mone, S., 2016. Online Blood Bank Management System using Android. *International Journal of Innovative Studies in Sciences and Engineering Technology*, 2(2), p. 55.

John, D. & M. S., . R., 2013. RESTDoc: Describe, Discover and Compose RESTful Semantic Web Services using Annotated Documentations. *International Journal of Web & Semantic Technology (IJWest)*, 4(1), p. 37.

Lunawat, N. M. et al., 2016. Blood and Organ For Patient Using Android Application. *IJRET: International Journal of Research in Engineering and Technology*, 05(05), p. 312.

Mumbaikar, S. & Padiya, P., 2013. Web Services Based On SOAP and REST Principles. *International Journal of Scientific and Research Publications*, 3(5), p. 1.

Oxford, n.d. *English Oxford Living Dictionaries*. [Online] Available at: <https://en.oxforddictionaries.com/definition/smartphone> [Accessed 19 6 2017].

Rachman, A. F., 2015. *detik.com*. [Online] Available at: <https://inet.detik.com/consumer/d-3054169/android-kuasai-asia-tenggara-di-indonesia-paling-juara> [Accessed 7 11 2016].

Snigdha, et al., 2015. Android Blood Bank. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(11), p. 86.

Yamamoto, F., 2014. Review:ABO blood group system—ABH oligosaccharide antigens, anti-A and anti-B,A and B. *Immunoematology*, 20(1), p. 3.

SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK
PEMBANGUNAN APLIKASI MOBILE UNIT TRANSFUSI DARAH
PMI KOTA YOGYAKARTA

Untuk :

Universitas Atma Jaya Yogyakarta


Dipersiapkan Oleh :

Nicky Apriliani/ 13 07 07209

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ATMA JAYA YOGYAKARTA

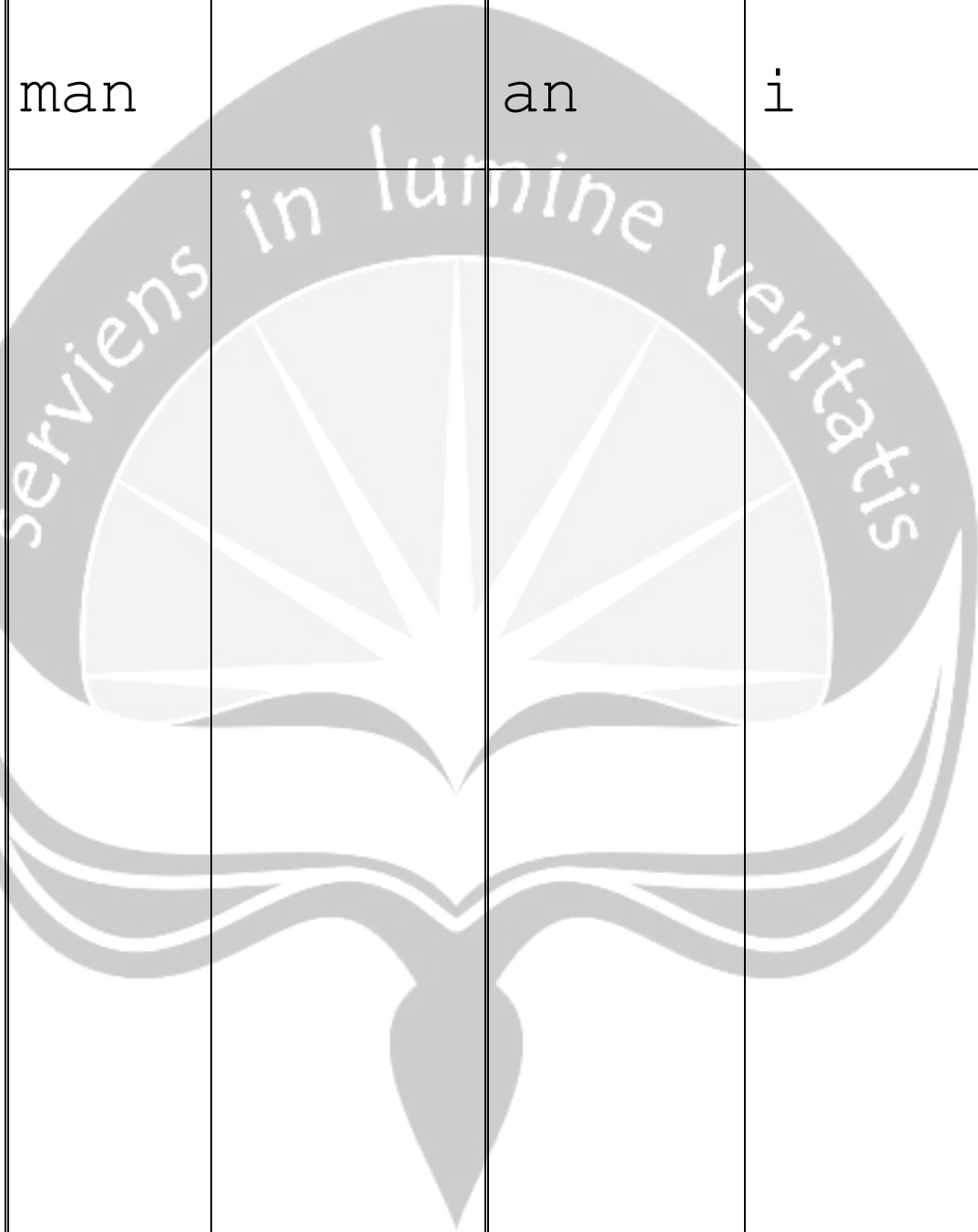
	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<i>SKPL</i> —		1/40
		Revisi		

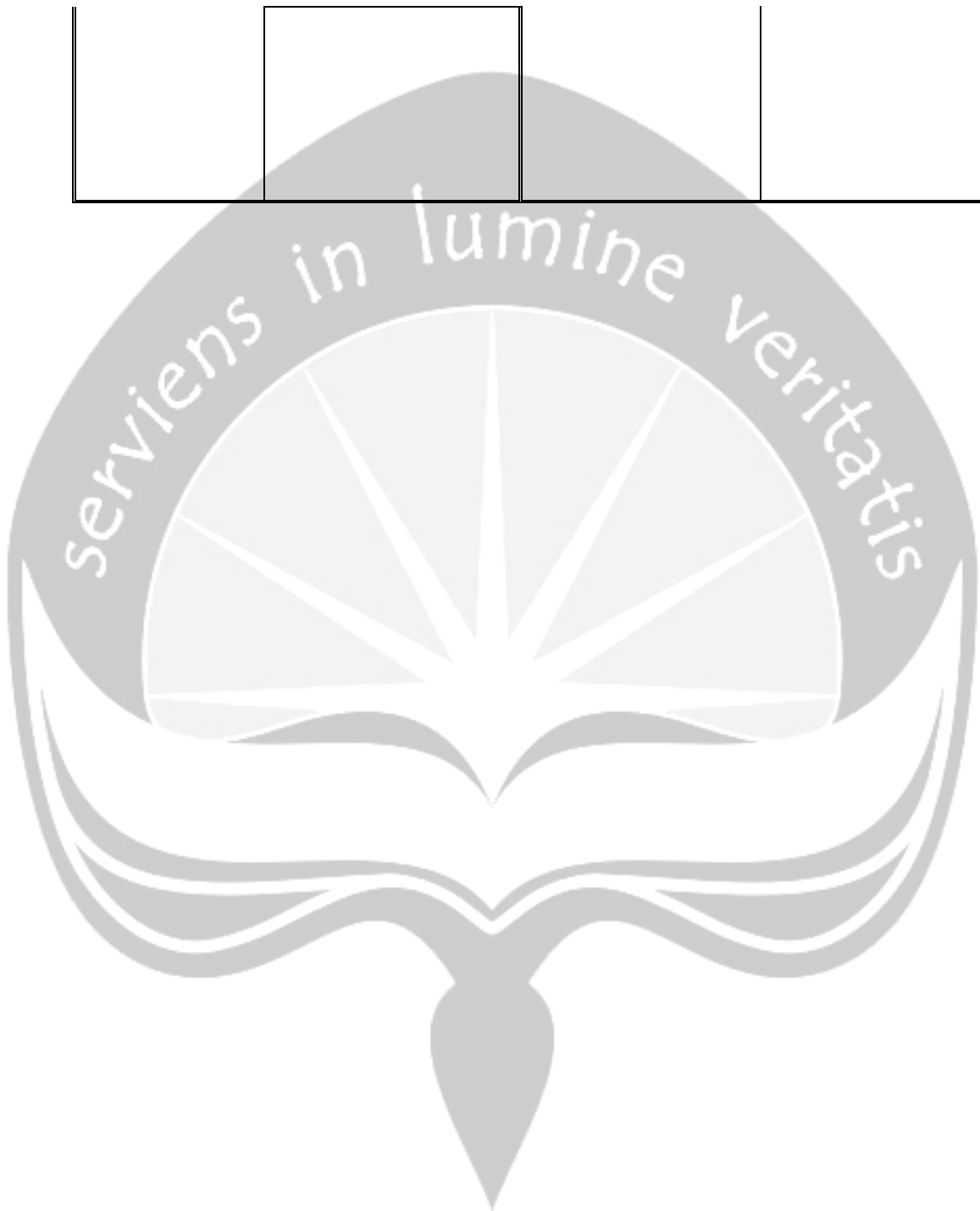
DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX	-	A	B	C	D	E	F	G
TGL								
Ditulis oleh	NA							
Diperiksa oleh	WIL/AGB							
Disetujui oleh								

Daftar Halaman Perubahan

Hala man	Revisi	Halam an	Revis i
			



Daftar Isi

DAFTAR PERUBAHAN	128
Daftar Halaman Perubahan	128
Daftar Isi	131
Daftar Gambar	133
Bab 1	134
Pendahuluan	134
1.1. Tujuan	134
1.2. Lingkup Masalah	134
1.3. Definisi, Akronim Dan Singkatan	135
1.4. Referensi	135
1.5. Deskripsi Umum	136
Bab II	137
Deskripsi Kebutuhan	137
2.1. Perspektif Produk	137
2.2. Fungsi Produk	138
2.3. Karakteristik Pengguna	144
2.4. Batasan - Batasan	144
2.5. Asumsi dan Ketergantungan	144
Bab III	146
Kebutuhan Khusus	146
3.1. Kebutuhan Antarmuka Eksternal	146

3.2. Kebutuhan Fungsionalitas Perangkat Lunak	147
Bab IV	150
Spesifikasi Rinci Kebutuhan	150
4.1. Spesifikasi Kebutuhan Fungsionalitas <i>Frontend</i> ..	150
4.2. Spesifikasi Kebutuhan Fungsionalitas <i>Backend</i> ..	160
Bab V	169
Entity Relationship Diagram (ERD)	169



Daftar Gambar

Gambar 2.1. Arsitektur Perangkat Lunak.....	138
Gambar 3.1. Use Case Diagram Aplikasi Mobile	148
Gambar 3.2. Use Case Diagram <i>Web Backend</i>	149
Gambar 5.1. Gambar ERD.....	169



Bab 1

Pendahuluan

1.1. Tujuan

Dokumen spesifikasi perangkat lunak (SKPL) merupakan dokumen spesifikasi kebutuhan perangkat lunak aplikasi mobile unit donor darah pmi kota Yogyakarta. Untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal dan atribut, serta mendefinisikan fungsi perangkat lunak. SKPL-PMI juga mendefinisikan batasan perancangan perangkat lunak.

1.2. Lingkup Masalah

Perangkat lunak dikembangkan dengan tujuan sebagai berikut:

- i. Aplikasi *mobile* dapat menjadi solusi bagi masyarakat dalam menerima informasi yang akurat mengenai data-data darah yang tersedia di PMI.
- ii. Aplikasi *mobile* dapat menjadi solusi dalam memberikan informasi kepada masyarakat mengenai kegiatan - kegiatan dari PMI.
- iii. Informasi dalam bentuk notifikasi kepada pendonor ketika stok darah PMI sudah mencapai batas minimum dapat menjadi solusi dalam mencegah kekosongan stok darah di PMI.
- iv. Data Riwayat donor dapat menjadi solusi bagi pengguna dalam mengetahui waktu terakhir donor darah dilakukan.

1.3. Definisi, Akronim Dan Singkatan

Berikut disajikan table berupa daftar definisi singkatan yang terdapat pada SKPL-PMI.

Kata kunci atau Akronim	Definisi
SKPL	Merupakan sebuah dokumen spesifikasi kebutuhan dari perangkat lunak yang akan dibangun.
SKPL-PMI-XXX	Kode yang merepresentasikan kebutuhan pada aplikasi dimana XXX merupakan nomor fungsi produk
Mobile	Sebuah lingkungan yang akan digunakan untuk menjalankan aplikasi.

1.4. Referensi

Referensi yang digunakan pada perangkat lunak adalah:

- i. Aloysius Editiyan / 120706957, *Spesifikasi Kebutuhan Perangkat Lunak (SKPL) SIBEVI*, Universitas Atma Jaya Yogyakarta.
- ii. Nicky Apriliani / 130707209, *Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ILUSI*, Universitas Atma Jaya Yogyakarta.

1.5. Deskripsi Umum

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

Bagian kedua berisi penjelasan umum tentang perangkat lunak yang akan dibangun. Bagian kedua berisi tentang perspektif produk yang akan dibangun, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pembangunan aplikasi.

Bagian ketiga berisi penjelasan secara rinci mengenai kebutuhan perangkat lunak yang akan dikembangkan.

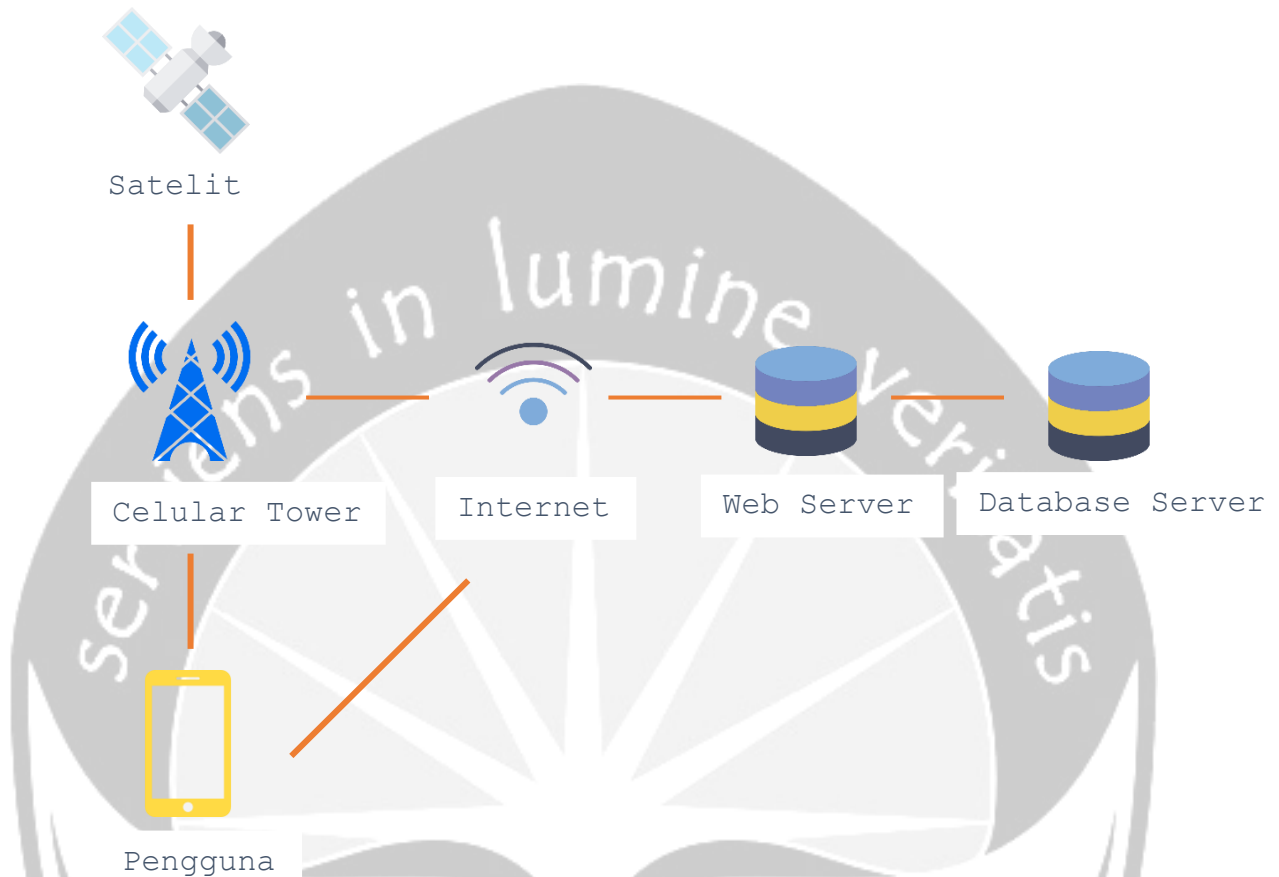
Bab II

Deskripsi Kebutuhan

2.1. Perspektif Produk

Aplikasi unit donor darah dikembangkan dengan tujuan memepermudah masyarakat untuk mendapatkan informasi - informasi mengenai pmi. Informasi yang disediakan berupa informasi stok darah, informasi kegiatan - kegiatan pmi dan informasi donor pengguna. Aplikasi menyediakan fitur notifikasi ketika stok darah tertentu sudah berada di batas minimum. Fitur ini digunakan untuk menjaga agar stok darah pmi akan selalu tersedia. Aplikasi menyediakan data donor pengguna dana akan memberikan notifikasi ketika pengguna sudah bisa melakukan donor. Pengguna juga bisa melihat jumlah poin yang di dapatkan dari donor dan membeli merchandise pmi. Selain itu, aplikasi memiliki fitur tambahan seperti tukar poin dengan diskon dari toko - toko yang bekerja sama dengan pmi. Serta, toko online yang akan memudahkan pengguna dalam membeli merchandise pmi.

Aplikasi dikembangkan pada perangkat smartphone dengan sistem operasi android. Aplikasi dikembangkan menggunakan tools Android Studio dengan bahasa pemrograman Java, serta XML dalam pembuatan layout. Aplikasi menggunakan web service sebagai pengelola data dari database ke aplikasi. Webservice dibuat menggunakan framework CodeIgniter dengan bahasa pemrograman PHP.



Gambar 2.1. Arsitektur Perangkat Lunak

Pada gambar 2.1. arsitektur perangkat lunak yang digunakan adalah *client server*. Inputan data dari pengguna akan disimpan ke server menggunakan koneksi internet. Pada saat pengaksesan data, *client* mengirimkan permintaan (*request*) ke server untuk diproses. Hasil pemrosesan data dikirim kembali ke client dengan bentuk data JSON.

2.2. Fungsi Produk

Berikut adalah fungsi produk aplikasi:

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

Fungsi yang digunakan oleh user untuk dapat masuk ke dalam sistem.

2. Fungsi Daftar Akun (**SKPL-PMI-002**)

Fungsi yang digunakan oleh pengguna untuk membuat akun baru.

3. Fungsi Kelola Akun (**SKPL-PMI-003**)

Fungsi digunakan pengguna untuk melengkapi data-data pengguna.

4. Fungsi Tampil Daftar Konten Acara (**SKPL-PMI-004**)

Fungsi digunakan untuk menampilkan daftar acara. Fungsi ini dijalankan setelah pengguna login dan ketika menu acara dipilih.

5. Fungsi Tampil Detil Konten Acara (**SKPL-PMI-005**)

Fungsi digunakan untuk menampilkan detil acara. Fungsi ini dijalankan setelah salah satu acara pada daftar acara dipilih.

6. Fungsi Tampil Daftar Kategori dan Stok Darah (**SKPL-PMI-006**)

Fungsi digunakan untuk menampilkan data stok darah berdasarkan kategori darah. Fungsi dijalankan ketika menu darah dipilih.

7. Fungsi Tampil Daftar Konten Partner PMI (**SKPL-PMI-007**)

Fungsi digunakan untuk menampilkan daftar konten partner PMI. Fungsi dijalankan ketika pengguna memilih menu voucher.

8. Fungsi Tampil Detil Konten Partner PMI (**SKPL-PMI-008**)

Fungsi digunakan untuk menampilkan detail partner PMi. Fungsi dijalankan ketika pengguna memilih salah satu data partner dari daftar partner PMI.

9. Fungsi Tampil Daftar Barang Usaha Dana PMI **(SKPL-PMI-009)**

Fungsi digunakan untuk menampilkan daftar barang dari usaha dana PMi. Fungsi dijalankan ketika pengguna memilih menu belanja.

10. Fungsi Tampil Detail Barang Usaha Dana PMI **(SKPL-PMI-010)**

Fungsi digunakan untuk menampilkan detail barang yang dipilih pengguna. Fungsi dijalankan setelah pengguna memilih salah satu barang di daftar barang.

11. Fungsi Kelola Keranjang Belanja **(SKPL-PMI-011)**

Fungsi yang digunakan untuk mengelola data keranjang belanja pengguna.

Fungsi pengelolaan data mencakup:

a. Fungsi Entri Data **(SKPL-PMI-011-01)**

Fungsi yang digunakan untuk menambahkan data barang belanja.

b. Fungsi Hapus Data **(SKPL-PMI-011-02)**

Fungsi yang digunakan untuk menghapus data barang belanja.

12. Fungsi Pesan Barang **(SKPL-PMI-012)**

Fungsi dijalankan ketika pengguna melakukan pembelian barang di menu toko.

13. Fungsi Tampil Transaksi **(SKPL-PMI-013)**

Fungsi digunakan untuk menampilkan data transaksi pengguna baik yang sedang berjalan maupun yang sudah selesai.

14. Fungsi Unggah Gambar Bukti Pembayaran **(SKPL-PMI-014)**

Fungsi digunakan pengguna untuk mengunggah gambar bukti pembayaran.

15. Fungsi Notifikasi Stok Darah **(SKPL-PMI-015)**

Fungsi ini dibuat oleh sistem ketika stok darah berada pada batas minimum.

16. Fungsi Notifikasi Status Donor **(SKPL-PMI-016)**

Fungsi dibuat oleh sistem ketika status donor pengguna berubah menjadi 'sudah bisa donor'.

Aplikasi WEB

17. Fungsi Login **(SKPL-PMI-017)**

Fungsi digunakan oleh admin untuk masuk ke dalam web.

18. Fungsi Kelola Akun **(SKPL-PMI-018)**

Fungsi digunakan oleh admin untuk mengubah data pmi.

19. Fungsi Kelola Data Acara **(SKPL-PMI-019)**

Fungsi digunakan untuk mengelola data acara. Fungsi kelola data acara mencakup:

a. Fungsi Tampil Data Acara **(SKPL-PMI-019-01)**

Fungsi digunakan untuk menampilkan daftar data acara.

b. Fungsi Tambah Data Acara **(SKPL-PMI-019-02)**

Fungsi digunakan untuk menambah data acara.

c. Fungsi Ubah Data Acara **(SKPL-PMI-019-03)**

Fungsi digunakan untuk mengubah data acara.

d. Fungsi Hapus Data Acara **(SKPL-PMI-019-04)**

Fungsi digunakan untuk menghapus data acara.

20. Fungsi Kelola Data Darah **(SKPL-PMI-020)**

Fungsi digunakan untuk mengelola data darah. Fungsi kelola data darah mencakup:

a. Fungsi Tampil Data Kategori Darah **(SKPL-PMI-020-01)**

Fungsi digunakan untuk menampilkan data kategori darah.

b. Fungsi Tambah Data Kategori Darah **(SKPL-PMI-020-02)**

Fungsi digunakan untuk menambah data kategori darah.

c. Fungsi Ubah Data Kategori Darah **(SKPL-PMI-020-03)**

Fungsi digunakan untuk mengubah data kategori darah.

d. Fungsi Hapus Data Kategori Darah **(SKPL-PMI-020-04)**

Fungsi digunakan untuk menghapus data kategori darah.

e. Fungsi Tampil Data Darah **(SKPL-PMI-020-05)**

Fungsi digunakan untuk menampilkan data darah.

f. Fungsi Tambah Data Darah **(SKPL-PMI-020-06)**

Fungsi digunakan untuk menampilkan data darah.

g. Fungsi Ubah Data Darah **(SKPL-PMI-020-07)**

Fungsi digunakan untuk menampilkan data darah.

h. Fungsi Hapus Data Darah **(SKPL-PMI-020-08)**

Fungsi digunakan untuk menampilkan data darah.

21. Fungsi Kelola Data Partner (SKPL-PMI-021)

Fungsi digunakan untuk mengelola data partner.

Fungsi kelola data partner mencakup:

a. Fungsi Tampil Data Partner **(SKPL-PMI-021-01)**

Fungsi digunakan untuk menampilkan daftar data partner.

b. Fungsi Tambah Data Partner **(SKPL-PMI-021-02)**

Fungsi digunakan untuk menambah data partner.

c. Fungsi Ubah Data Partner **(SKPL-PMI-021-03)**

Fungsi digunakan untuk mengubah data partner.

d. Fungsi Hapus Data Partner **(SKPL-PMI-021-04)**

Fungsi digunakan untuk menghapus data partner.

22. Fungsi Kelola Data Barang (SKPL-PMI-022)

Fungsi digunakan untuk mengelola data barang. Fungsi kelola data barang mencakup:

a. Fungsi Tampil Data Barang **(SKPL-PMI-022-01)**

Fungsi digunakan untuk menampilkan daftar data barang.

b. Fungsi Tambah Data Barang **(SKPL-PMI-022-02)**

Fungsi digunakan untuk menambah data barang.

c. Fungsi Ubah Data Barang **(SKPL-PMI-022-03)**

Fungsi digunakan untuk mengubah data barang.

d. Fungsi Hapus Data Barang **(SKPL-PMI-022-04)**

Fungsi digunakan untuk menghapus data barang.

23. Fungsi Kelola Data Transaksi (SKPL-PMI-023)

Fungsi ini digunakan untuk mengelola transaksi.

Fungsi kelola data transaksi mencakup:

a. Fungsi Tampil Daftar Transaksi **(SKPL-PMI-023-01)**

Fungsi digunakan untuk menampilkan daftar transaksi.

b. Fungsi Tampil Detil Transaksi **(SKPL-PMI-023-02)**

Fungsi digunakan untuk menampilkan detil transaksi. Fungsi dijalankan setelah admin menekan tombol melihat detail.

c. Fungsi Ubah Status Transaksi **(SKPL-PMI-023-03)**

Fungsi digunakan untuk mengubah status transaksi.

2.3. Karakteristik Pengguna

Karakteristik dari pengguna aplikasi ICan adalah sebagai berikut:

- i. Memahami pengoperasian sistem operasi Android versi 4.1 keatas.
- ii. Mengerti penggunaan aplikasi unit donor darah pmi.

2.4. Batasan - Batasan

Batasan-batasan dalam pengembangan aplikasi ICan ini adalah:

- i. Kebijakan Umum
Berpedoman pada tujuan dari pembangunan aplikasi.
- ii. Keterbatasan Perangkat Keras
Dapat diketahui kemudian setelah aplikasi ini berjalan.

2.5. Asumsi dan Ketergantungan

Aplikasi ini dapat berjalan pada perangkat mobile Android baik dalam bentuk ponsel pintar maupun tablet. Sistem operasi yang bisa digunakan adalah Android versi 4.2 atau dikenal dengan *jelly bean* (API 17) keatas.



Bab III

Kebutuhan Khusus

3.1. Kebutuhan Antarmuka Eksternal

Kebutuhan antarmuka eksternal pada aplikasi meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras dan antarmuka perangkat lunak.

3.1.1. Antarmuka Pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form.

3.1.2. Antarmuka Perangkat Keras

Antarmuka perangkat keras yang digunakan adalah:

- i. Perangkat ponsel pintar Android
- ii. Perangkat Database Server
- iii. Perangkat Web Server

3.1.3. Antarmuka Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan aplikasi adalah sebagai berikut:

- i. Nama : Sistem Operasi Android versi 4.2
Sumber: Google
Sebagai sistem operasi untuk menjalankan aplikasi.
- ii. Nama : MySql
Sumber: GPL
Sebagai database management system (DBMS) yang digunakan untuk menyimpan data.
- iii. Nama : Web Browser
Sumber: Berbagai sumber

Sebagai aplikasi untuk pengelolaan web service dan backend.

iv. Nama : Android Studio

Sumber: Google

Sebagai alat yang digunakan untuk menulis kode program ketika membangun aplikasi.

v. Nama : Sublime Text

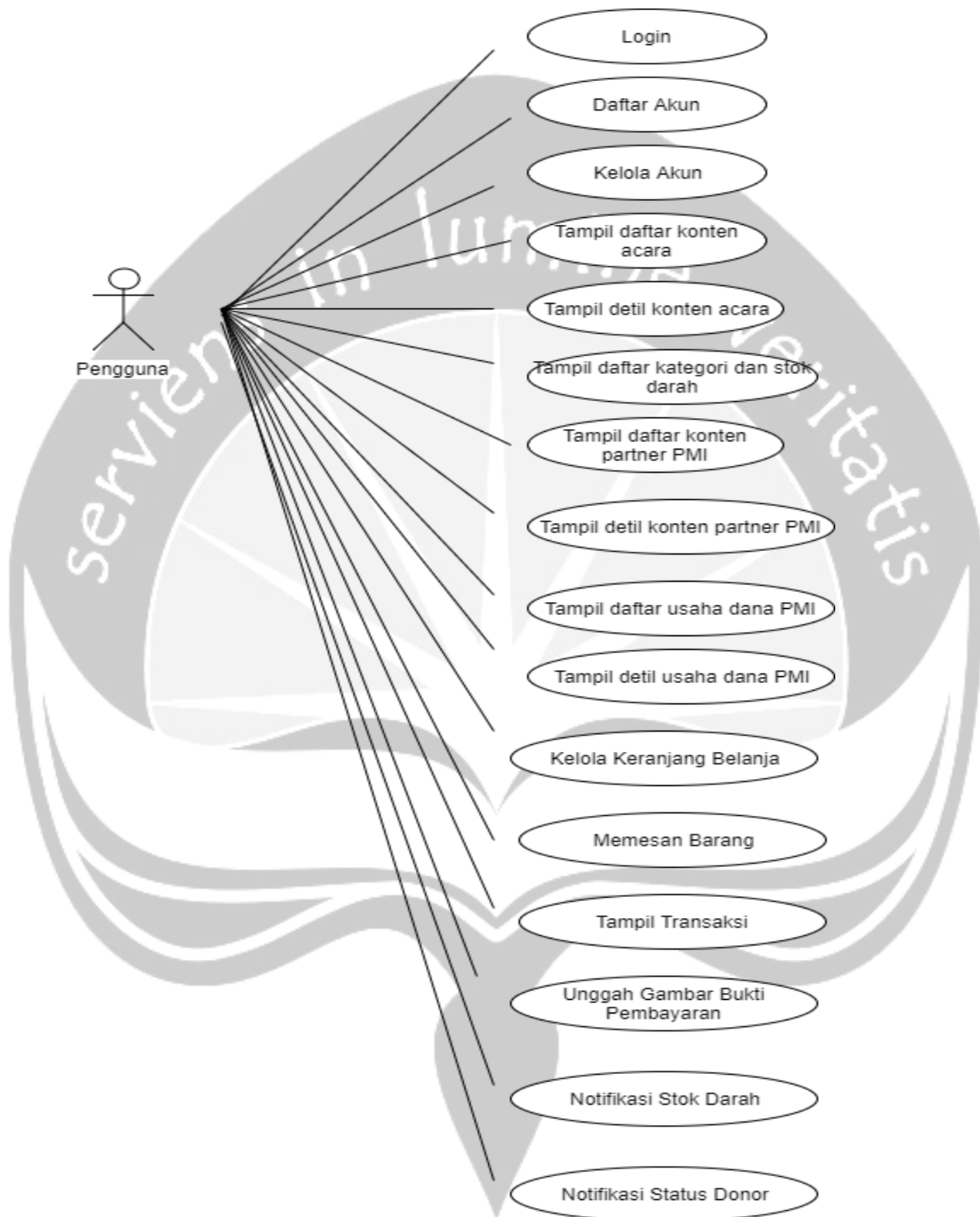
Sumber: Jon Skinner

Sebagai aplikasi pembangunan *web service* dan *backend web*.

3.2. Kebutuhan Fungsionalitas Perangkat Lunak

3.2.1. Diagram Use Case Aplikasi

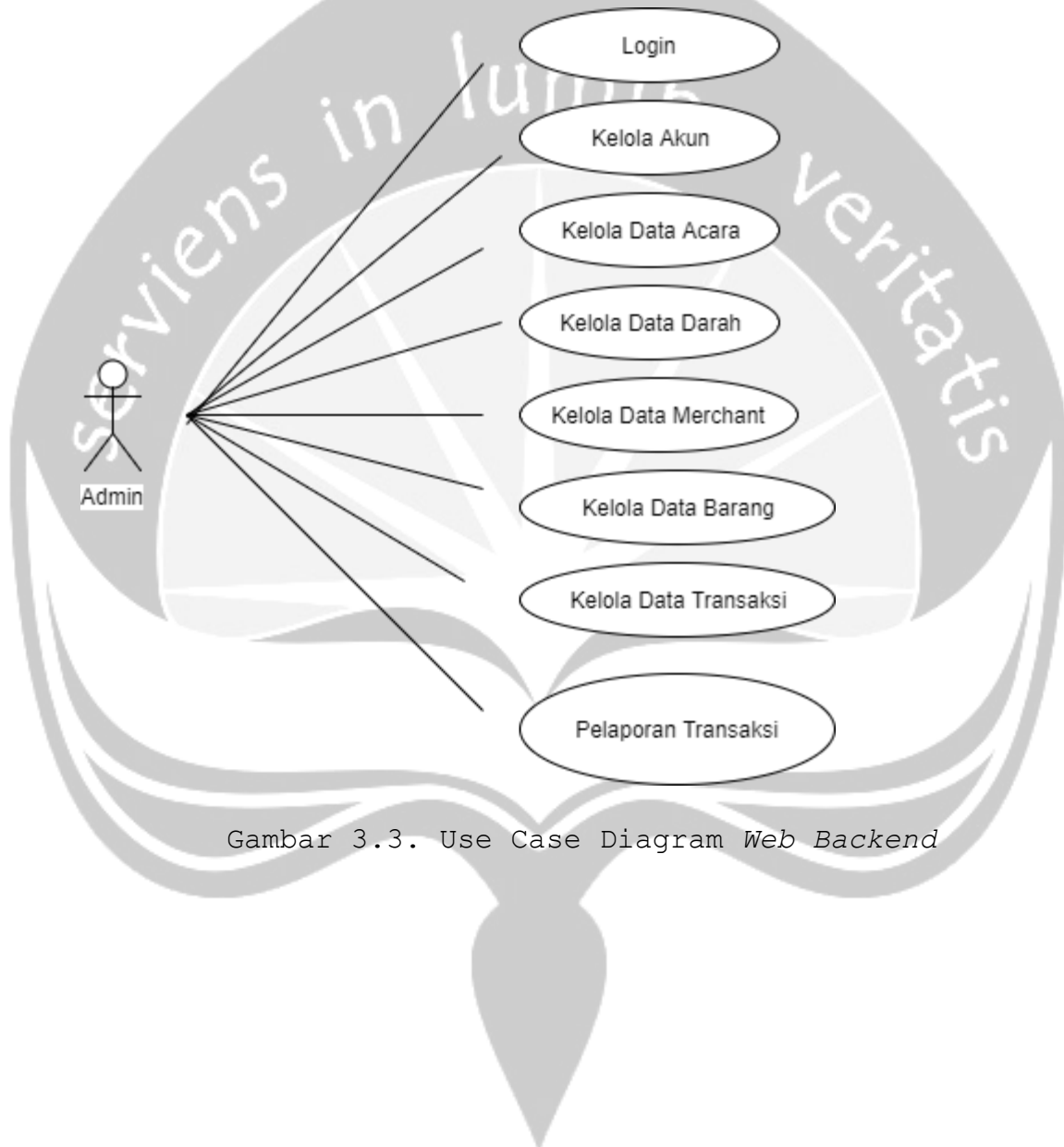
Berikut merupakan diagram use case dari aplikasi.



Gambar 3.2. Use Case Diagram Aplikasi Mobile

3.2.2. Diagram Use Case *Backend*

Berikut merupakan diagram use case dari aplikasi *backend*.



Gambar 3.3. Use Case Diagram *Web Backend*

Bab IV

Spesifikasi Rinci Kebutuhan

4.1. Spesifikasi Kebutuhan Fungsionalitas *Frontend*

4.1.1. Use case specification : Login

1. Brief Description

Use case ini digunakan aktor untuk masuk ke dalam aplikasi.

2. Actor

Pengguna

3. Basic Flow

- a. Use case dimulai ketika aktor membuka aplikasi
- b. Sistem akan menampilkan form login.
- c. Aktor memasukkan username dan password
- d. Aktor menekan tombol login.

4. Pre Condition

- a. Aktor sudah memiliki akun.

5. Post Condition

Aktor masuk ke aplikasi.

4.1.2. Use case Spesifikasi : Daftar Akun

1. Brief Description

Use case digunakan aktor untuk membuat akun baru.

2. Actor

Pengguna

3. Basic Flow

- a. Use case dimulai ketika aktor menekan tombol daftar.

- b. Sistem akan menampilkan form registrasi.
 - c. Aktor mengisi data - data yang diminta.
 - d. Aktor menekan tombol daftar.
4. Pre Condition
Aktor belum memiliki akun
 5. Post Condition
Aktor memiliki akun.

4.1.3. Use case Spesification : Kelola Akun

1. Brief Description
Use case digunakan untuk mengelola akun pengguna.
2. Actor
Pengguna
3. Basic Flow
 - a. Use case dimulai ketika aktor akan menekan tombol ubah data diri di menu profile.
 - b. Sistem menampilkan form yang berisi data diri aktor.
 - c. Aktor mengisi form dan menekan tombol ubah data.
4. Pre Condition
Aktor sudah memiliki akun.
5. Post Condition
Data aktor sudah terupade di database.

4.1.4. Use case Spesification : Tampil Daftar Konten Acara

1. Brief Description

Use case digunakan untuk menampilkan daftar data acara.

2. Actor

Pengguna

3. Basic Flow

- a. Use case dimulai ketika pengguna sudah login.
- b. Sistem menampilkan data daftar konten acara.

4. Pre Condition

- a. Use case login sudah dilakukan.
- b. Aktor telah memasuki sistem.

5. Post Condition

Sistem menampilkan data daftar konten acara.

4.1.5. Use case Spesification : Tampil Detil Konten Acara

1. Brief Description

Use case ini digunakan untuk menampilkan detil konten acara.

2. Actor

Pengguna

3. Basic Flow

- a. Use case dimulai ketika aktor memilih salah satu konten dari daftar konten acara.
- b. Sistem menampilkan detil konten acara.

4. Pre Condition

- a. Use case login sudah dilakukan.
- b. Aktor sudah memasuki sistem
5. Post Condition
Sistem menampilkan detil konten acara.

4.1.6. Use case Spesification : Tampil Daftar Kategori dan Stok Darah

1. Brief Description
Use case digunakan untuk menampilkan data daftar kategori dan stok darah.
2. Actor
Pengguna
3. Basic Flow
 - a. Use case dimulai ketika pengguna memilih menu darah.
 - b. Sistem menampilkan data kategori darah dan stok darah.
4. Pre Condition
 - a. Use case login sudah dilakukan.
 - b. Aktor telah memasuki sistem.
5. Post Condition
Sistem menampilkan data daftar kategori darah dan stok darah.

4.1.7. Use case Spesification : Tampil Daftar Konten Partner PMI

1. Brief Description

Use case digunakan untuk menampilkan daftar konten partner pmi.

2. Actor

Pengguna

3. Basic Flow

a. Use case dimulai ketika aktor memilih menu voucher.

b. Sistem menampilkan daftar konten partner pmi.

4. Pre Condition

a. Use case login sudah dilakukan.

b. Aktor telah memasuki sistem.

5. Post Condition

Daftar konten partner pmi ditampilkan.

4.1.8. Use case Spesification : Tampil Detil Konten Partner PMI

1. Brief Description

Use case digunakan untuk menampilkan detil konten partner pmi.

2. Actor

Pengguna

3. Basic Flow

a. Use case dimulai ketika pengguna memilih salah satu data dari daftar partner.

b. Sistem menampilkan detil data partner pmi.

4. Pre Condition

a. Use case login sudah dilakukan.

- b. Aktor telah memasuki sistem.
- 5. Post Condition
Data detil partner ditampilkan.

4.1.9. Use case Spesification : Tampil Daftar Barang Usaha Dana PMI

1. Brief Description
Use case digunakan untuk menampilkan daftar barang dari usaha dana PMI.
2. Actor
Pengguna
3. Basic Flow
 - a. Use case dimulai ketika aktor memilih menu toko.
 - b. Sistem menampilkan daftar barang usaha dana.
4. Pre Condition
 - a. Use case login sudah dilakukan.
 - b. Aktor telah memasuki sistem.
5. Post Condition
Daftar barang ditampilkan.

4.1.10. Use case Spesification : Tampil Detil Barang Usaha Dana PMI

1. Brief Description
Use case digunakan untuk menampilkan detil barang.
2. Actor
Pengguna
3. Basic Flow

- a. Use case dimulai ketika aktor memilih salah satu barang dari daftar barang.
- b. Sistem menampilkan detail barang yang dipilih aktor.
- 4. Pre Condition
 - a. Use case login sudah dilakukan.
 - b. Aktor telah memasuki sistem.
- 5. Post Condition
Detail barang ditampilkan.

4.1.11. Use case Spesification : Kelola Keranjang Belanjaan

- 1. Brief Description
Use case digunakan untuk mengelola keranjang belanja pengguna.
- 2. Actor
Pengguna
- 3. Basic Flow
 - a. Use case dimulai ketika aktor memilih untuk mengelola daftar belanja.
 - b. Sistem menampilkan daftar belanja aktor.
 - A-1 Aktor memilih untuk menambah barang
 - A-2 Aktor memilih untuk menghapus barang.
- 4. Alternative Flow
A-1 Aktor memilih menambah barang ke keranjang belanja.
 - a. Aktor membuka detail barang yang ingin dibeli dari daftar barang.

b. Aktor menekan tombol tambah ke keranjang belanja.
 c. Sistem menambahkan data barang ke daftar belanja.
 A-2 Aktor memilih menghapus barang dari keranjang belanja

- a. Aktor membuka daftar keranjang belanja.
- b. Aktor menekan gambar tempat sampah di samping daftar barang yang ingin dihapus.
- c. Data barang dihapus.

5. Pre Condition

- a. Use case login sudah dilakukan.
- b. Aktor telah memasuki sistem.

6. Post Condition

Daftar keranjang belanja ditampilkan

4.1.12. Use case Spesification : Memesan Barang

1. Brief Description

Use case ini digunakan untuk melakukan pemesanan barang dari keranjang belanja.

2. Actor

Pengguna

3. Basic Flow

- a. Use case dimulai ketika aktor menekan tombol pesan di keranjang belanja.
- b. Sistem menampilkan konfirmasi pemesanan.
- c. Aktor menekan tombol proses.

4. Pre Condition

- a. Use case login sudah dilakukan.
- b. Aktor telah memasuki sistem.

- c. Aktor telah memiliki barang di keranjang belanja
- 5. Post Condition
Data transaksi terinput ke database.

4.1.13. Use case Spesification : Tampil Transaksi

1. Brief Description
Use case ini digunakan untuk menampilkan data transaksi pengguna.
2. Actor
Pengguna
3. Basic Flow
 - a. Pengguna memilih data transaksi yang akan dilihat.
 - b. Sistem menampilkan data rincian transaksi pengguna.
4. Pre Condition
 - a. Use case login sudah dilakukan.
 - b. Aktor telah memasuki sistem.
 - c. Pengguna sudah melakukan transaksi pemesanan barang.
5. Post Condition
Sistem menampilkan data transaksi.

4.1.14. Use case Spesification : Unggah Gambar Bukti Pembayaran

1. Brief Description
Use case ini digunakan untuk mengunggah gambar bukti pembayaran.
2. Actor
Pengguna

3. Basic Flow
 - a. Use case dimulai ketika ada transaksi yang berstatus belum mengunggah bukti pembayaran.
 - b. Aktor memilih gambar yang akan diunggah.
 - c. Sistem mengunggah gambar.
4. Pre Condition
 - a. Use case login sudah dilakukan.
 - b. Aktor telah memasuki sitem.
 - c. Pengguna sudah melakukan transaksi pemesanan barang.
5. Post Condition

Status transaksi berubah menjadi bukti telah di unggah.

4.1.15. Use case Spesification : Notifikasi Stok Darah

1. Brief Description

Use case digunakan oleh sistem untuk menampilkan notifikasi ketika jumlah darah berada pada batas minimum.
2. Actor

Sistem
3. Basic Flow
 - a. Use case dimulai ketika admin mengubah jumlah stok darah.
 - b. Sistem akan mengecek apakah jumlah stok darah lebih kecil dari jumlah minimum.
 - c. Sistem menampilkan notifikasi.
4. Pre Condition

Data darah sudah diupdate.
5. Post Condition

Sistem akan menampilkan notifikasi mengenai kondisi stok darah.

4.1.16. Use case Spesification : Notifikasi Status Donor

1. Brief Description

Use case digunakan oleh sistem untuk menampilkan notifikasi jika status donor berubah menjadi bisa donor.

2. Actor

Sistem

3. Basic Flow

a. Use case dimulai ketika pengguna melakukan donor dan status donor menjadi sudah donor.

b. Sistem akan mengecek apakah tanggal sekarang sudah 3 bulan dari tanggal terakhir donor.

c. Sistem menampilkan notifikasi

4. Pre Condition

Data terakhir donor sudah terupdate.

5. Post Condition

Sistem akan menampilkan notifikasi tentang perubahan status.

4.2. Spesifikasi Kebutuhan Fungsionalitas Backend

4.2.1. Use case Spesification: Login

1. Brief Description

Use case ini digunakan aktor untuk masuk ke dalam web.

2. Actor

Admin

3. Basic Flow
 - a. Use case dimulai ketika *actor* membuka web
 - b. Sistem akan menampilkan form login.
 - c. Aktor memasukan username dan password
 - d. Aktor menekan tombol login.
4. Pre Condition
 - a. Aktor sudah memiliki akun.
5. Post Condition
Aktor masuk ke web.

4.2.2. Use case Spesification: Kelola Akun

1. Brief Description
Use case ini digunakan aktor untuk mengelola akun PMI.
2. Actor
Admin
3. Basic Flow
 - a. Use case dimulai ketika aktor memilih menu profile.
 - b. Sistem menampilkan data-data yang bisa di ubah.
 - c. Aktor mengubah data.
 - d. Sistem menyimpan perubahan data ke database.
4. Pre Condition
Aktor sudah masuk ke dalam sistem.
5. Post Condition
Data terupdate di database.

4.2.3. Use case Spesification: **Kelola Data Acara**

1. Brief Description

Use case digunakan untuk mengelola data acara.

2. Actor

Admin

3. Basic Flow

a. Use case dimulai ketika aktor memilih untuk mengelola daftar acara.

b. Sistem menampilkan daftar data acara.

A-1 Aktor memilih untuk menambah data.

A-2 Aktor memilih untuk mengubah data.

A-3 Aktor memilih untuk menghapus data.

4. Alternative Flow

A-1 Aktor memilih untuk menambah data.

a. Aktor menekan tombol tambah data acara.

b. Sistem menampilkan form tambah data acara.

c. Aktor menginputkan data acara.

d. Sistem menginput data di database.

A-2 Aktor memilih untuk mengubah data.

a. Aktor menekan tombol ubah data acara.

b. Sistem menampilkan form ubah data acara.

c. Aktor mengubah data acara.

d. Sistem mengupdate data di database

A-3 Aktor memilih untuk menghapus data

a. Aktor menekan tombol hapus data acara.

b. Sistem menampilkan konfirmasi hapus data acara.

c. Aktor mengonfirmasi penghapusan data.

- d. Sistem mengupdate data di database.
- 5. Pre Condition
 - a. Use case login sudah dilakukan.
 - b. Aktor telah memasuki sistem.
- 6. Post Condition
 - Data acara telah diupdate

4.2.4. Use case Spesification: Kelola Data Darah

1. Brief Description
 - Use case digunakan untuk mengelola data darah.
2. Actor
 - Admin
3. Basic Flow
 - a. Use case dimulai ketika aktor memilih untuk mengelola data darah.
 - b. Sistem menampilkan daftar data darah.
 - A-1 Aktor memilih untuk menambah data kategori darah.
 - A-2 Aktor memilih untuk mengubah data kategori darah.
 - A-3 Aktor memilih untuk menghapus data kategori darah.
 - A-4 Aktor memilih untuk menambah data darah.
 - A-5 Aktor memilih untuk mengubah data darah.
 - A-6 Aktor memilih untuk menghapus data darah.
4. Alternative Flow
 - A-1 Aktor memilih untuk menambah data kategori darah.
 - a. Aktor menekan tombol tambah data kategori darah.

- b. Sistem menampilkan form tambah data kategori darah.
- c. Aktor menginputkan data kategori darah.
- d. Sistem menginput data di database.

A-2 Aktor memilih untuk mengubah data kategori darah.

- a. Aktor menekan tombol ubah data kategori darah.
- b. Sistem menampilkan data kategori darah.
- c. Aktor menginputkan data kategori darah.
- d. Sistem mengupdate data di database.

A-3 Aktor memilih untuk menghapus data kategori darah

- a. Aktor menekan tombol hapus data kategori darah.
- b. Sistem menampilkan konfirmasi hapus data kategori darah.
- c. Aktor mengonfirmasi penghapusan data.
- d. Sistem mengupdate data di database.

A-4 Aktor memilih untuk menambah data darah.

- a. Aktor menekan tombol tambah data darah.
- b. Sistem menampilkan form tambah data darah.
- c. Aktor menginputkan data darah.
- d. Sistem menginput data di database.

A-5 Aktor memilih untuk mengubah data darah.

- a. Aktor menekan tombol ubah data darah.
- b. Sistem menampilkan data darah.
- c. Aktor menginputkan data darah.
- d. Sistem mengupdate data di database.

A-6 Aktor memilih untuk menghapus data darah.

- a. Aktor menekan tombol hapus data darah.

- b. Sistem menampilkan konfirmasi hapus data kategori darah.
 - c. Aktor mengonfirmasi penghapusan data.
 - d. Sistem mengupdate data di database.
5. Pre Condition
- a. Use case login sudah dilakukan.
 - b. Aktor telah memasuki sistem.
6. Post Condition
- Database telah diupdate.

4.2.5. Use case Spesification: Kelola Data Partner

1. Brief Description
- Use case digunakan untuk mengelola data partner.
2. Actor
- Admin
3. Basic Flow
- a. Use case dimulai ketika aktor memilih untuk mengelola data partner.
 - b. Sistem menampilkan daftar data partner.
- A-1 Aktor memilih untuk menambah data.
- A-2 Aktor memilih untuk mengubah data.
- A-3 Aktor memilih untuk menghapus data.
4. Alternative Flow
- A-1 Aktor memilih untuk menambah data.
- a. Aktor menekan tombol tambah data partner.
 - b. Sistem menampilkan form tambah data partner.
 - c. Aktor menginputkan data partner.
 - d. Sistem menginput data di database.
- A-2 Aktor memilih untuk mengubah data.

- a. Aktor menekan tombol ubah data partner.
- b. Sistem menampilkan form ubah data partner.
- c. Aktor mengubah data partner.
- d. Sistem mengupdate data di database

A-3 Aktor memilih untuk menghapus data

- a. Aktor menekan tombol hapus data partner.
- b. Sistem menampilkan konfirmasi hapus data partner.
- c. Aktor mengonfirmasi penghapusan data.
- d. Sistem mengupdate data di database.

5. Pre Condition

- a. Use case login sudah dilakukan.
- b. Aktor telah memasuki sistem.

6. Post Condition

Data partner telah diupdate

4.2.6. Use case Spesification: Kelola Data Barang

1. Brief Description

Use case digunakan untuk mengelola data barang.

2. Actor

Admin

3. Basic Flow

a. Use case dimulai ketika aktor memilih untuk mengelola daftar barang.

b. Sistem menampilkan daftar data barang.

A-1 Aktor memilih untuk menambah data.

A-2 Aktor memilih untuk mengubah data.

A-3 Aktor memilih untuk menghapus data.

4. Alternative Flow

A-1 Aktor memilih untuk menambah data.

- a. Aktor menekan tombol tambah data barang.
- b. Sistem menampilkan form tambah data barang.
- c. Aktor menginputkan data barang.
- d. Sistem menginput data di database.

A-2 Aktor memilih untuk mengubah data.

- a. Aktor menekan tombol ubah data barang.
- b. Sistem menampilkan form ubah data barang.
- c. Aktor mengubah data barang.
- d. Sistem mengupdate data di database

A-3 Aktor memilih untuk menghapus data

- a. Aktor menekan tombol hapus data barang.
- b. Sistem menampilkan konfirmasi hapus data barang.
- c. Aktor mengonfirmasi penghapusan data.
- d. Sistem mengupdate data di database.

5. Pre Condition

- a. Use case login sudah dilakukan.
- b. Aktor telah memasuki sistem.

6. Post Condition

Data barang telah diupdate

4.2.7. Use case Spesification: Kelola Data Transaksi

1. Brief Description

Use case digunakan untuk mengelola data transaksi.

2. Actor

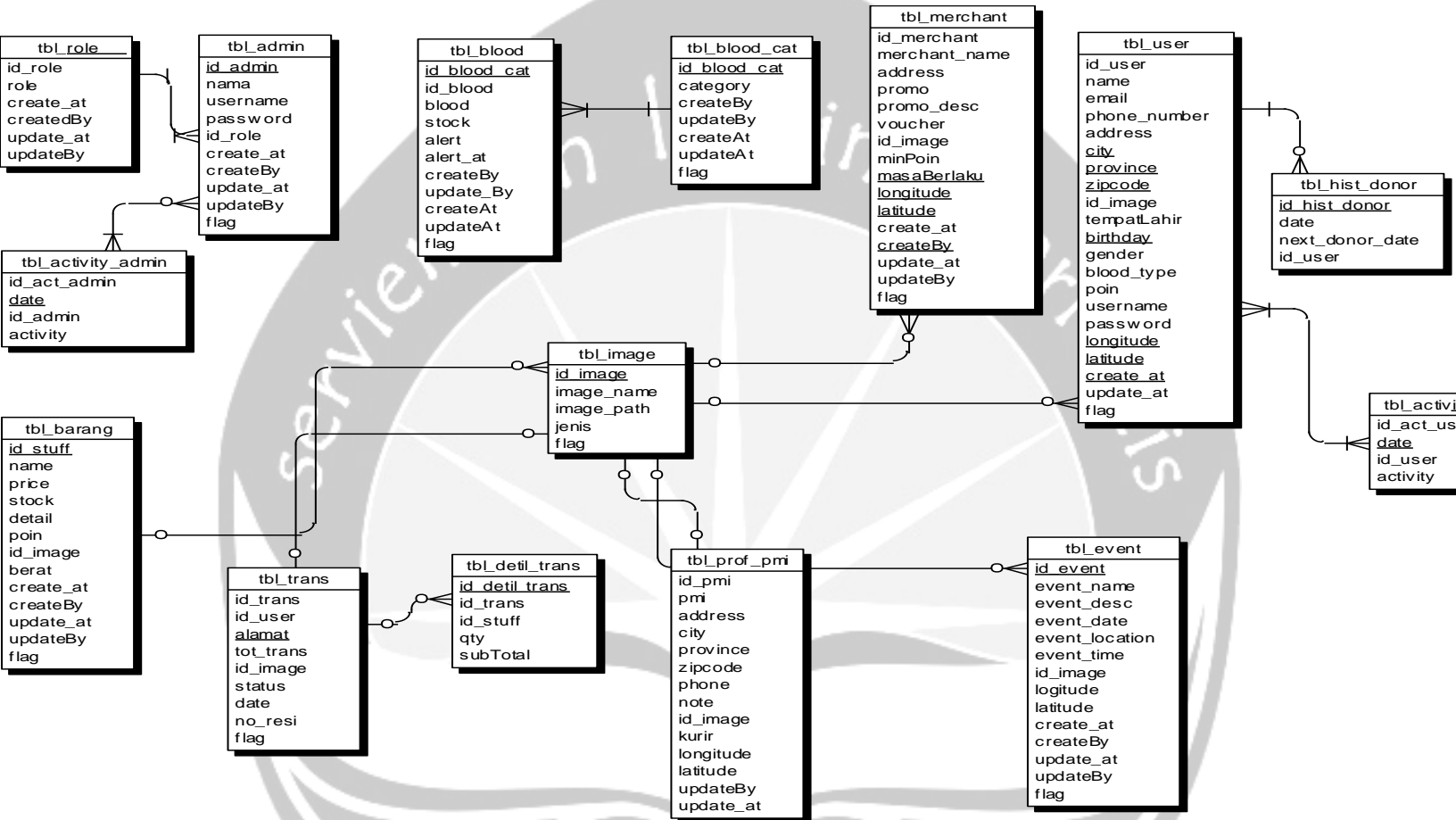
Admin

3. Basic Flow

- a. Use case dimulai ketika aktor memilih untuk mengelola daftar transaksi.
 - b. Sistem menampilkan daftar data acara.
 - A-1 Aktor memilih untuk melihat detail data transaksi.
 - A-2 Aktor memilih untuk mengubah status transaksi.
4. Alternative Flow
- A-1 Aktor memilih untuk melihat detail data transaksi.
 - a. Aktor menekan tombol lihat detail
 - b. Sistem menampilkan detail transaksi
 - A-2 Aktor memilih untuk mengubah status data transaksi.
 - a. Aktor menekan tombol ubah status
 - b. Sistem menampilkan form
 - c. Aktor memilih status baru untuk transaksi tersebut
5. Pre Condition
- a. Use case login sudah dilakukan.
 - b. Aktor telah memasuki sistem.
6. Post Condition
- Database transaksi telah diupdate.

Bab V

Entity Relationship Diagram (ERD)



Gambar 5.56. Gambar ERD

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK
PEMBANGUNAN APLIKASI MOBILE UNIT TRANSFUSI DARAH
PMI KOTA YOGYAKARTA

Untuk :

Universitas Atma Jaya Yogyakarta


Dipersiapkan Oleh :

Nicky Apriliani/ 13 07 07209

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ATMA JAYA YOGYAKARTA

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL		1/99
		Revisi		

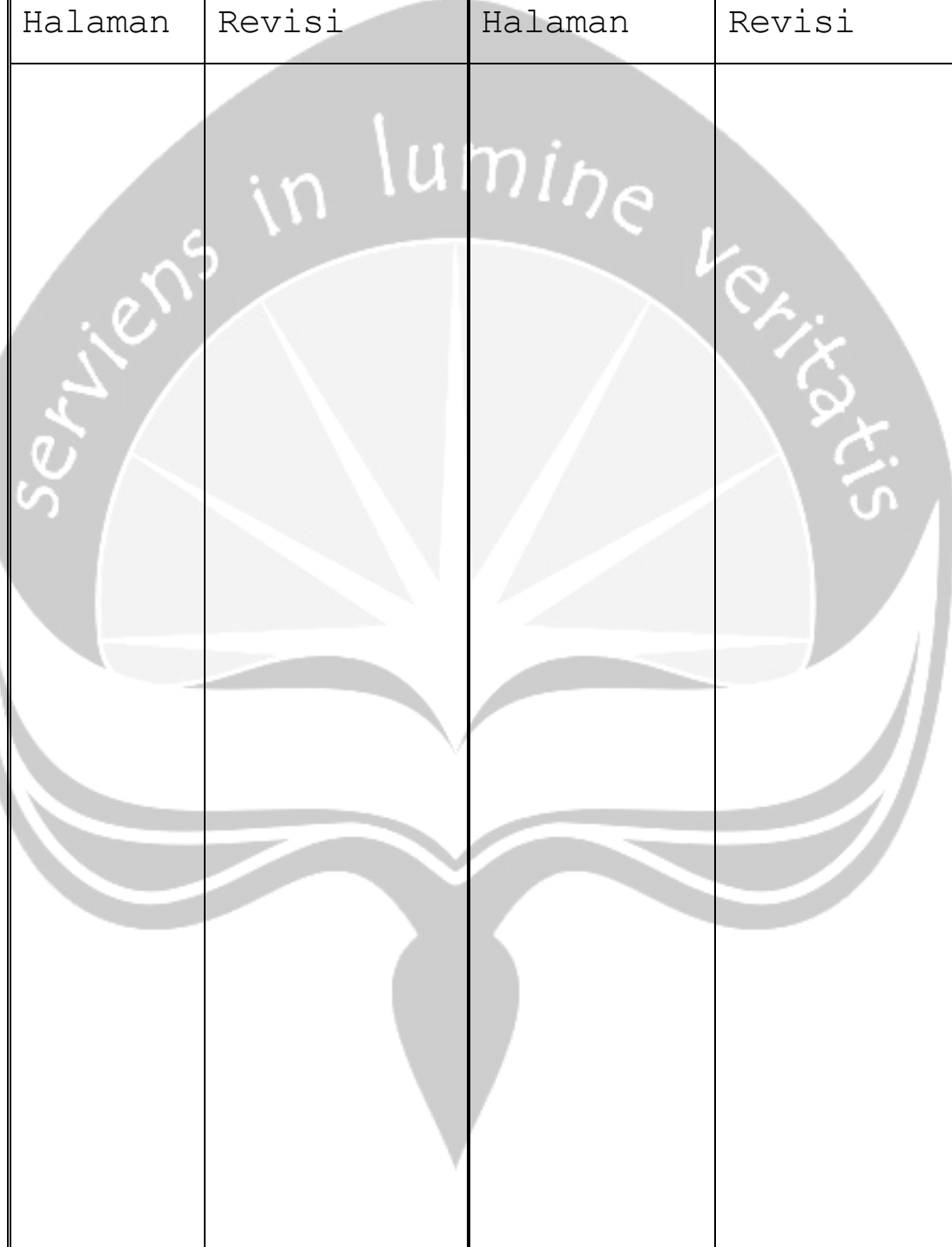
DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX	-	A	B	C	D	E	F	G
TGL								
Ditulis oleh	NA							
Diperiksa oleh	WIL/AGB							
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi



Daftar Isi

DAFTAR PERUBAHAN	171
Daftar Halaman Perubahan	172
Daftar Isi	173
Bab I	176
Pendahuluan	176
1.1. Tujuan	176
1.2. Lingkup Masalah	176
1.3. Definisi, Akronim, dan Singkatan	176
1.4. Referensi	177
1.5. Deskripsi Umum	177
Bab II	179
Perancangan Aplikasi	179
2.1. Perancangan Arsitektur	180
2.2. Perancangan Rinci	181
2.2.1. Sequence Diagram	181
2.2.2. Class Diagram	204
Bab III	222
Perancangan Data	222
3.1. Dekomposisi Data	222
3.1.1. Deskripsi Entitas Tabel Role	222
3.1.2. Deskripsi Entitas Tabel Admin	222
3.1.3. Deskripsi Entitas Tabel User	223
3.1.4. Deskripsi Entitas Tabel Hist Donor	223
3.1.5. Deskripsi Entitas Tabel Image	224
3.1.6. Deskripsi Entitas Tabel Voucher	224
3.1.7. Deskripsi Entitas Tabel Prof_PMI	224
3.1.8. Deskripsi Entitas Tabel Event	225

3.1.9. Deskripsi Entitas Tabel Blood_Cat.....	226
3.1.10. Deskripsi Entitas Tabel Blood.....	226
3.1.11. Deskripsi Entitas Tabel Merchant.....	227
3.1.12. Deskripsi Entitas Tabel Barang.....	227
3.1.13. Deskripsi Entitas Tabel Trans.....	228
3.1.14. Deskripsi Entitas Tabel Detil_Trans.....	228
3.2. Physical Data Model.....	230
Bab IV.....	231
Deskripsi Perancangan Antarmuka.....	231
4.1. Login.....	232
4.2. Signup.....	232
4.3. Daftar Konten Acara PMI.....	232
4.4. Deskripsi Konten Acara PMI.....	233
4.5. Daftar Darah PMI.....	234
4.6. Daftar Partner PMI (Voucher).....	234
4.7. Deskripsi Partner PMI (Voucher).....	235
4.8. Daftar Barang Usaha Dana PMI.....	235
4.9. Deskripsi Barang Usaha Dana PMI.....	236
4.10. Daftar Keranjang Belanja Pengguna.....	236
4.11. Pesan Barang.....	237
4.12. Daftar Transaksi Pengguna.....	237
4.13. Deskripsi Transaksi Pengguna.....	238
4.14. Data Pengguna.....	238
4.15. Ubah Data Pengguna.....	239
4.16. Login Web.....	240
4.17. Kelola Akun.....	240
4.18. Tampil Daftar Acara.....	241
4.19. Tambah Data Acara.....	241
4.20. Ubah Data Acara.....	242

4.21. Hapus Data Acara	242
4.22. Tampil Daftar Kategori Darah	243
4.23. Tambah Data Kategori Darah	243
4.24. Ubah Data Kategori Darah	244
4.25. Hapus Data Kategori Darah	244
4.26. Tampil Daftar Darah	245
4.27. Tambah Data Darah	245
4.28. Ubah Data Darah	246
4.29. Hapus Data Darah	246
4.30. Tampil Daftar Partner	247
4.31. Tambah Data Partner	247
4.32. Ubah Data Partner	248
4.33. Hapus Data Partner	248
4.34. Tampil Daftar Barang	249
4.35. Tambah Data Barang	249
4.36. Ubah Data Barang	250
4.37. Hapus Data Barang	250
4.38. Tampil Daftar Transaksi	251
4.39. Tampil Detil Transaksi	251
4.40. Ubah Status Transaksi	252

Bab I

Pendahuluan

1.1. Tujuan

Dokumen deskripsi perancangan perangkat lunak (DPPL) ini bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dibangun. Dokumen akan digunakan oleh pengembang aplikasi sebagai acuan untuk implementasi pada tahap berikutnya.

1.2. Lingkup Masalah

Perangkat lunak dikembangkan dengan tujuan:

- i. Aplikasi *mobile* dapat menjadi solusi bagi masyarakat dalam menerima informasi yang akurat mengenai data-data darah yang tersedia di PMI.
- ii. Aplikasi *mobile* dapat menjadi solusi dalam memberikan informasi kepada masyarakat mengenai kegiatan - kegiatan dari PMI.
- iii. Informasi dalam bentuk notifikasi kepada pendonor ketika stok darah PMI sudah mencapai batas minimum dapat menjadi solusi dalam mencegah kekosongan stok darah di PMI.
- iv. Data Riwayat donor dapat menjadi solusi bagi pengguna dalam mengetahui waktu terakhir donor darah dilakukan.

1.3. Definisi, Akronim, dan Singkatan

Berikut disajikan table berupa daftar definisi, akronim dan singkatan yang terdapat pada DPPL.

Kata kunci atau Akronim	Definisi
DPPL	Deskripsi perancangan perangkat lunak yang disebut juga dengan Software Design Description (SDD). Merupakan deskripsi dari perancangan produk yang akan dibangun.
DPPL-PMI-XXX	Kode yang merepresentasikan kebutuhan pada aplikasi dimana XXX merupakan nomor fungsi produk
Mobile	Sebuah lingkungan yang akan digunakan untuk menjalankan aplikasi.

1.4. Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

- i. Aloysius Editiyan / 120706957, *Deskripsi Perancangan Perangkat Lunak (DPPL) SIBEVI*, Universitas Atma Jaya Yogyakarta.
- ii. Nicky Apriliani / 130707209, *Deskripsi Perancangan Perangkat Lunak (DPPL) ILUSI*, Universitas Atma Jaya Yogyakarta.

1.5. Deskripsi Umum

Secara umum dokumen DPPL terbagi atas 4 bagian utama. Bagian pertama berisi penjelasan mengenai dokumen DPPL yang mencakup tujuan pembuatan DPPL, ruang lingkup masalah dalam

pengembangan perangkat lunak, definisi, referensi dan deskripsi umum tentang dokumen DPPL ini.

Bagian kedua berisi tentang perancangan aplikasi. Terdapat 2 buah sub bagian, pertama perancangan arsitektur dan yang kedua adalah perancangan rinci. Pada perancangan rinci terdapat sequence diagram dari aplikasi.

Bagian ketiga berisi perancangan data. Perancangan data memiliki 2 bagian yaitu dekomposisi data dan physical data model.

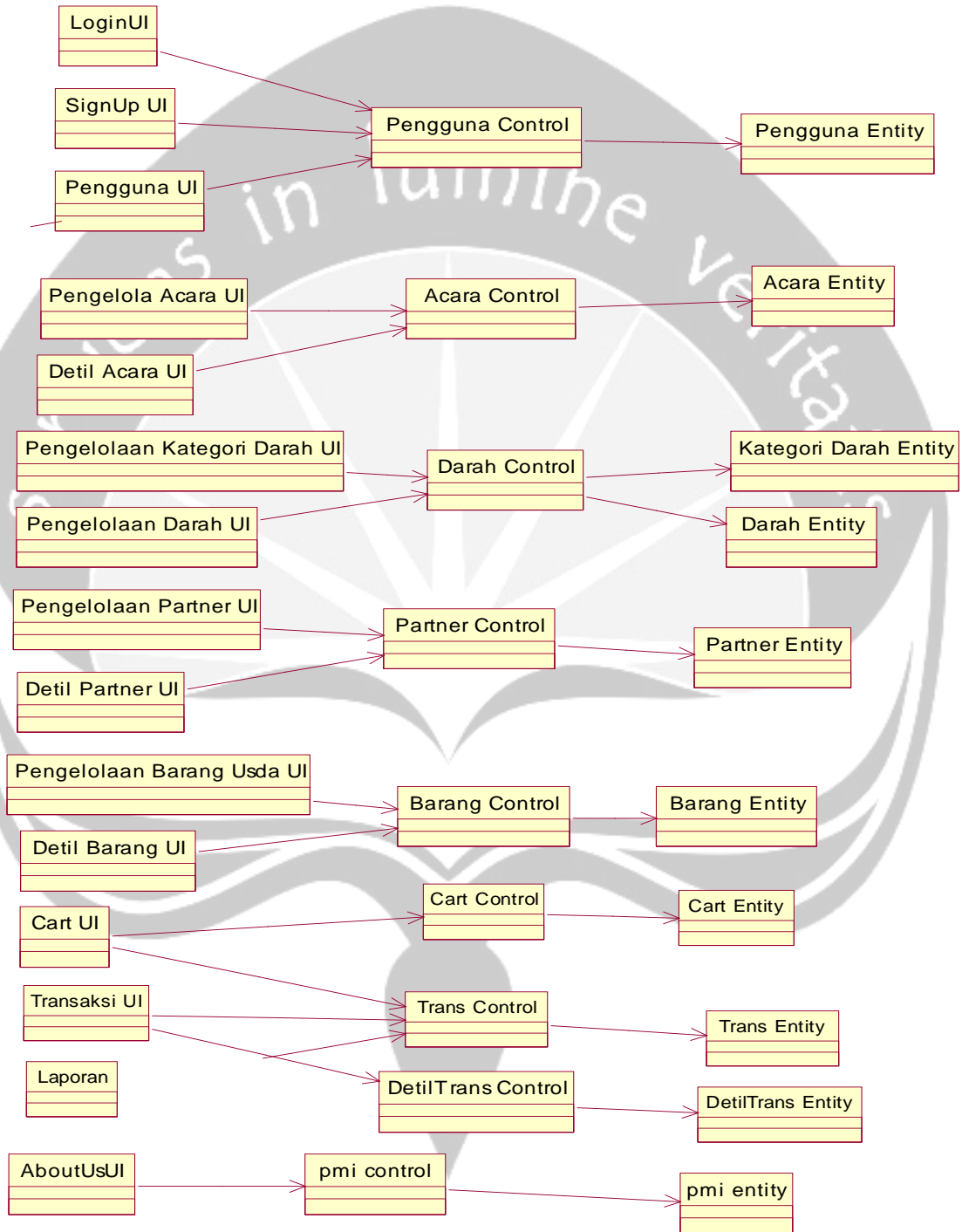
Bagian keempat berisi tentang deskripsi perancangan antarmuka. Bagian ini berisi mockup dari aplikasi yang akan dibangun.

Bab II

Perancangan Aplikasi



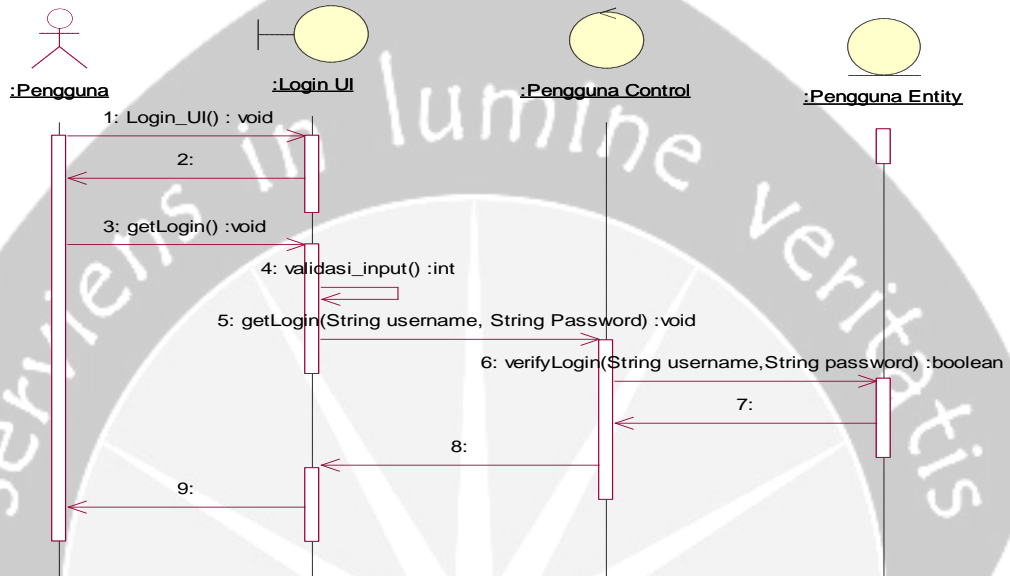
2.1. Perancangan Arsitektur



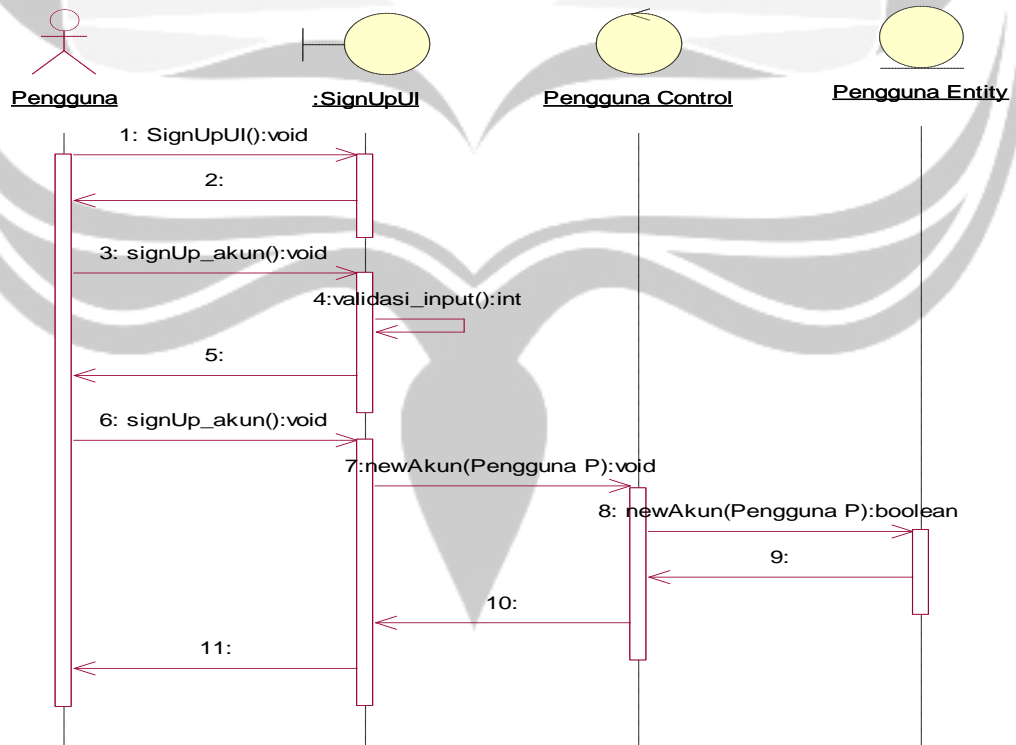
2.2. Perancangan Rinci

2.2.1. Sequence Diagram

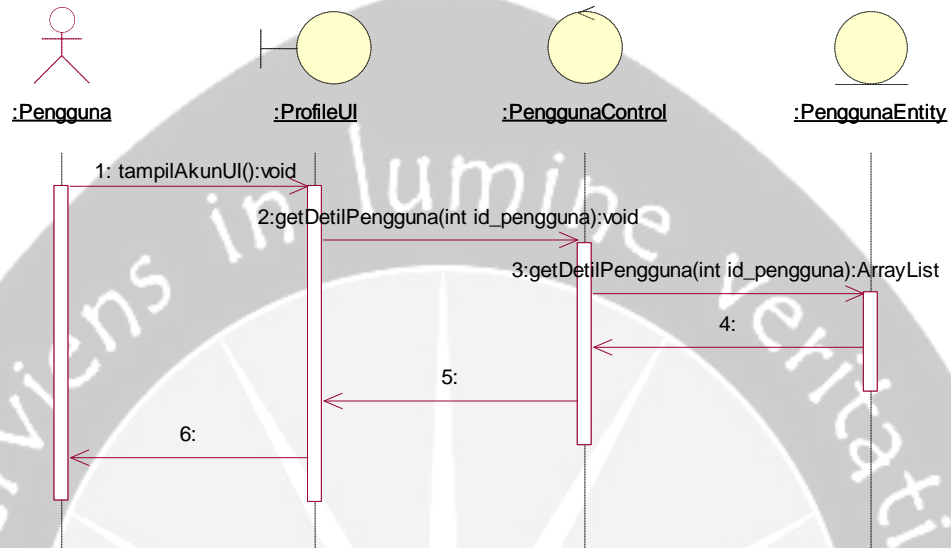
2.2.1.1. Fungsi Login



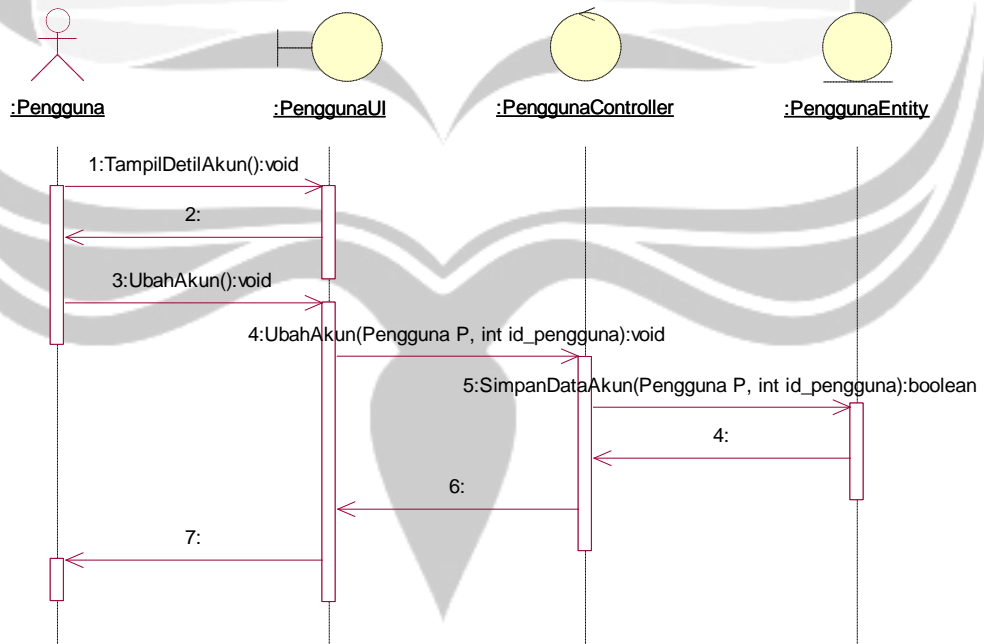
2.2.1.2. Fungsi SignUp



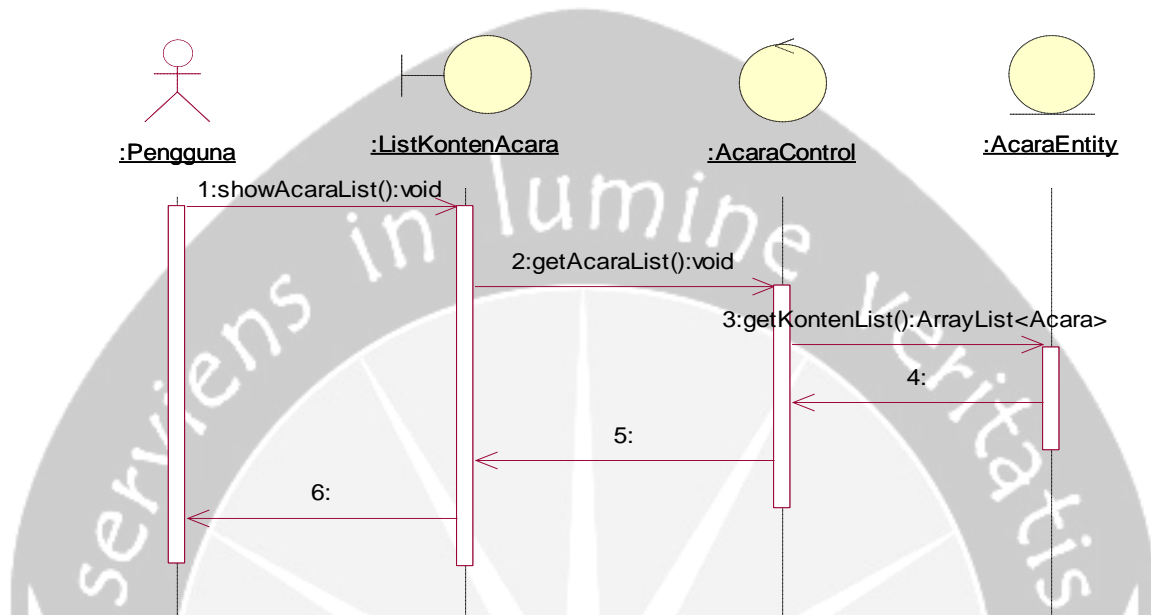
2.2.1.3. Fungsi Tampil Akun



2.2.1.4. Fungsi Ubah Akun

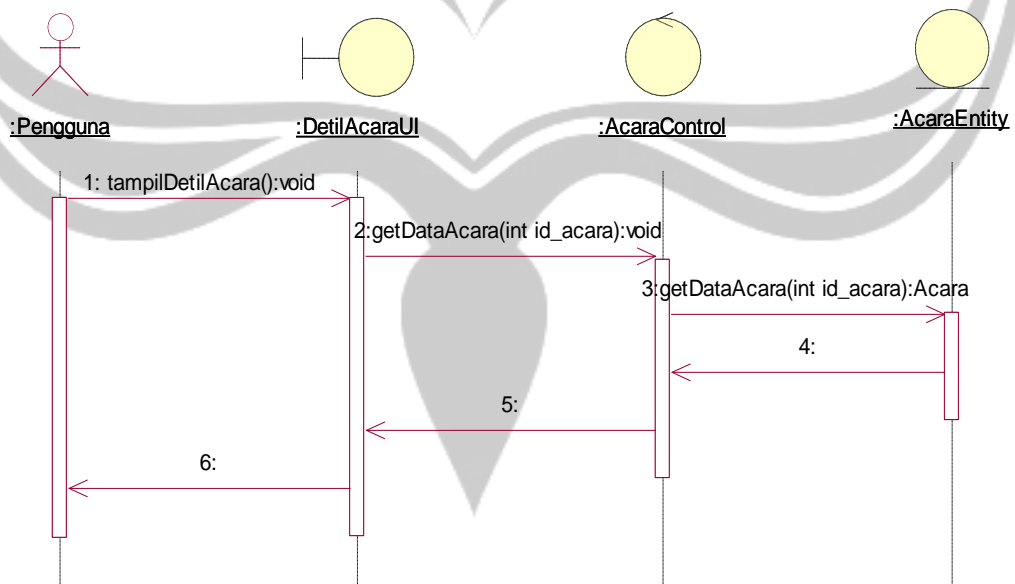


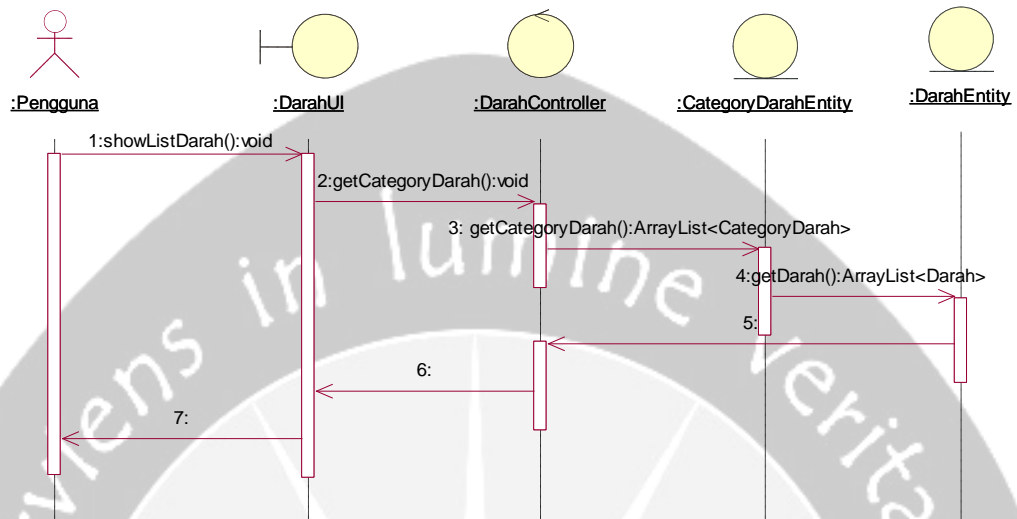
2.2.1.5. Fungsi Tampil Daftar Data Acara



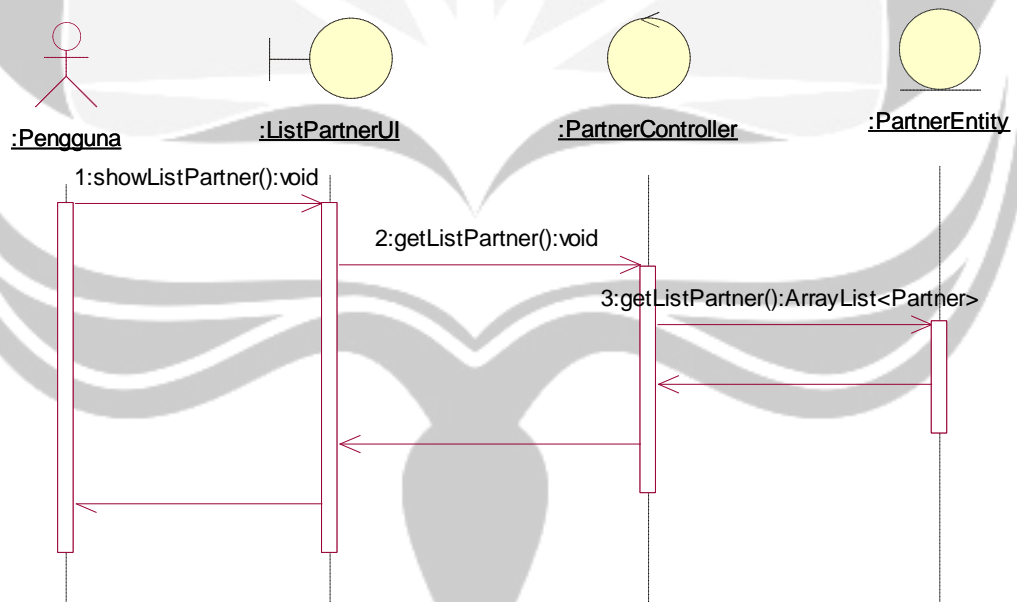
2.2.1.6. Fungsi Tampil Detil Acara

2.2.1.7. Fungsi Tampil Daftar Data Darah

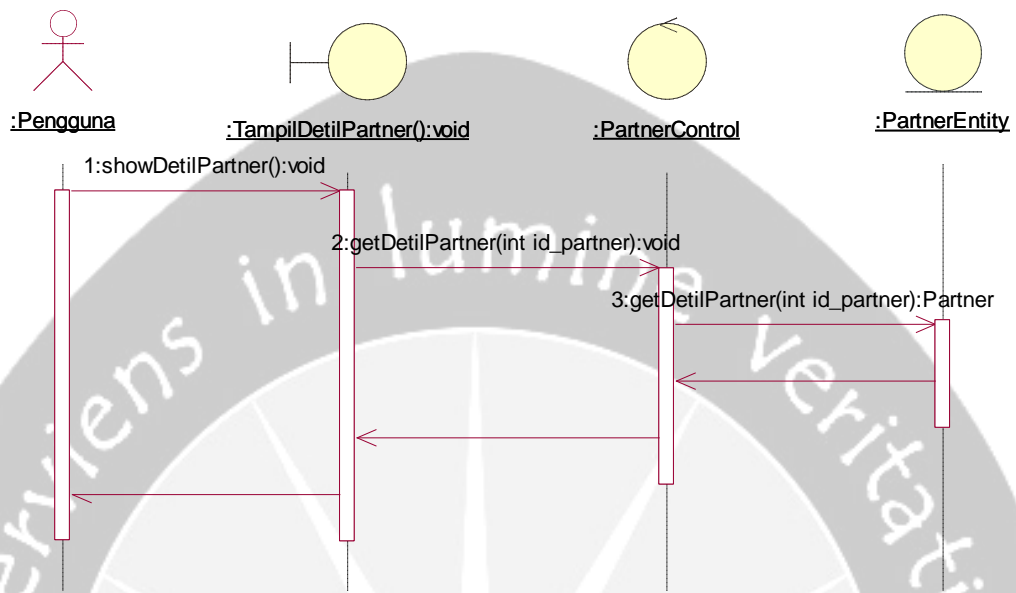




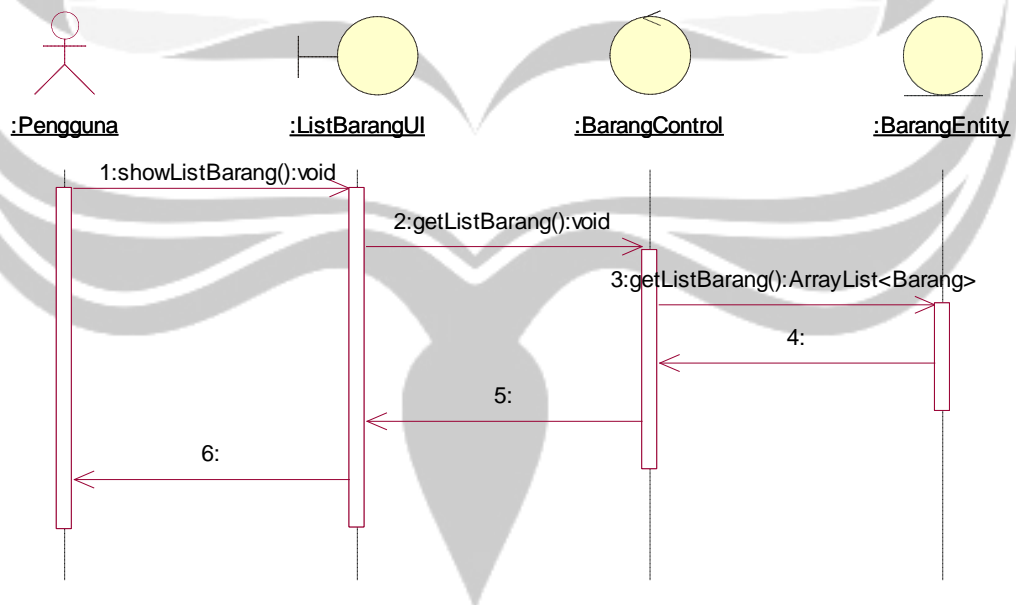
2.2.1.8. Fungsi Tampil List Data Partner



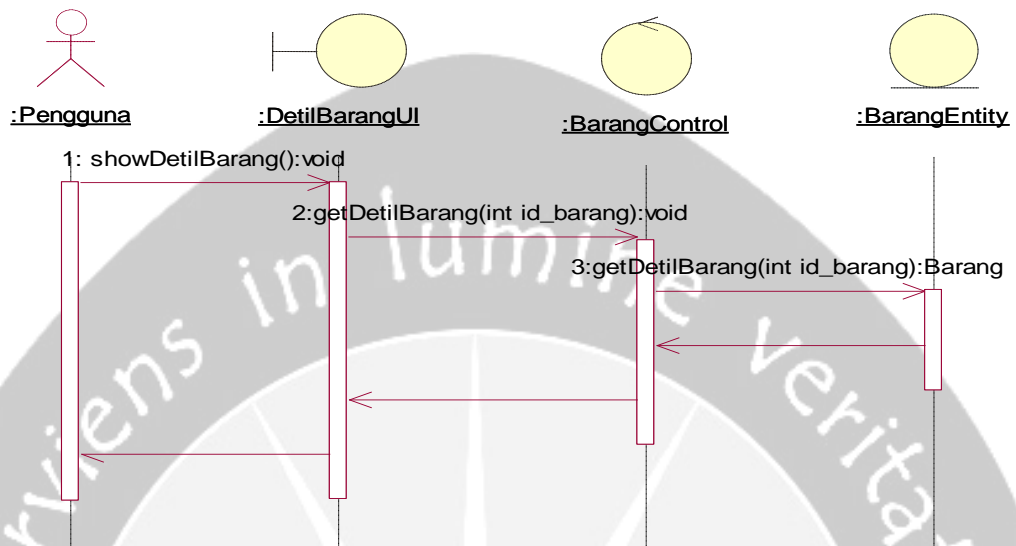
2.2.1.9. Fungsi Tampil Detil Partner



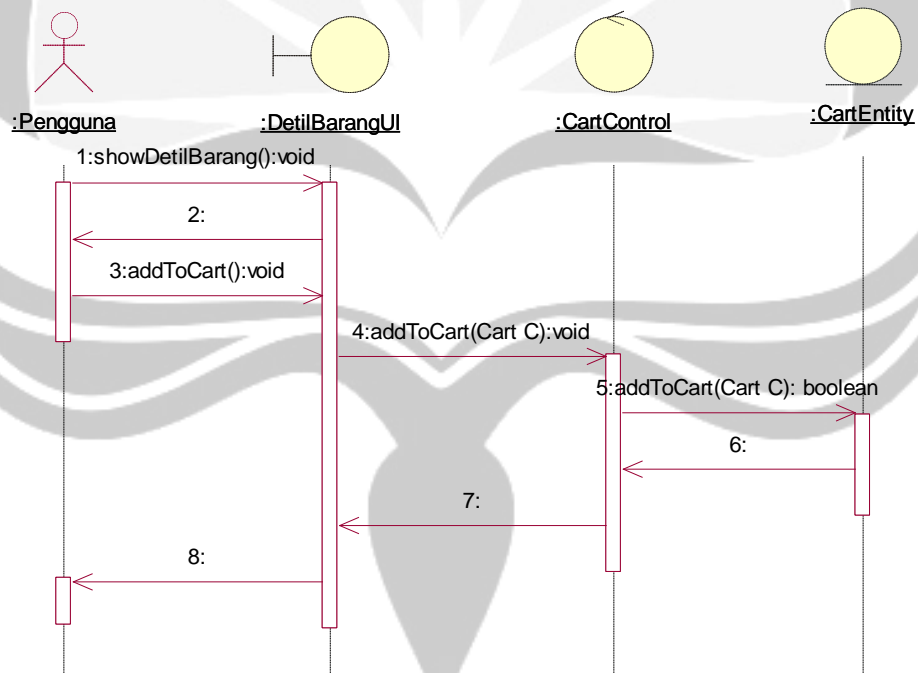
2.2.1.10. Fungsi Tampil Daftar Data Barang



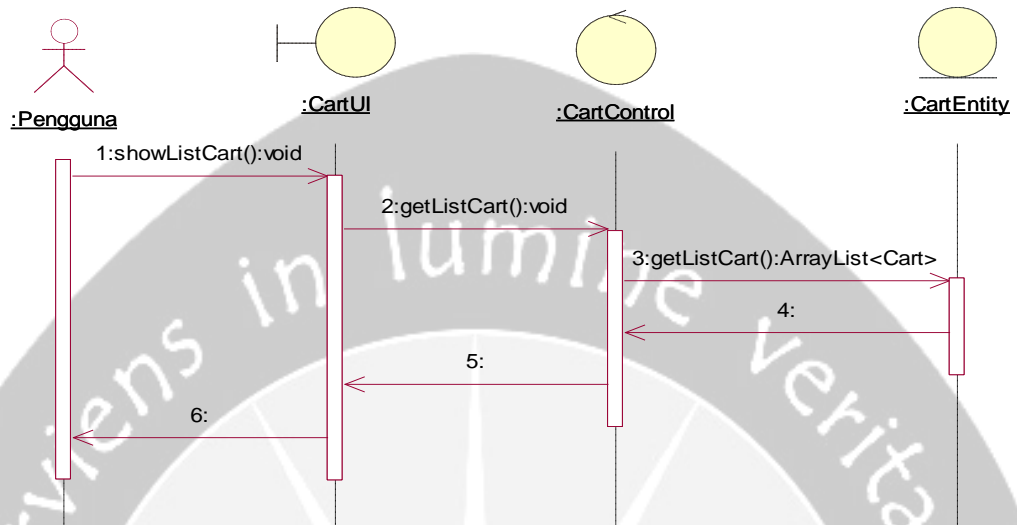
2.2.1.11. Fungsi Tampil Detil Barang



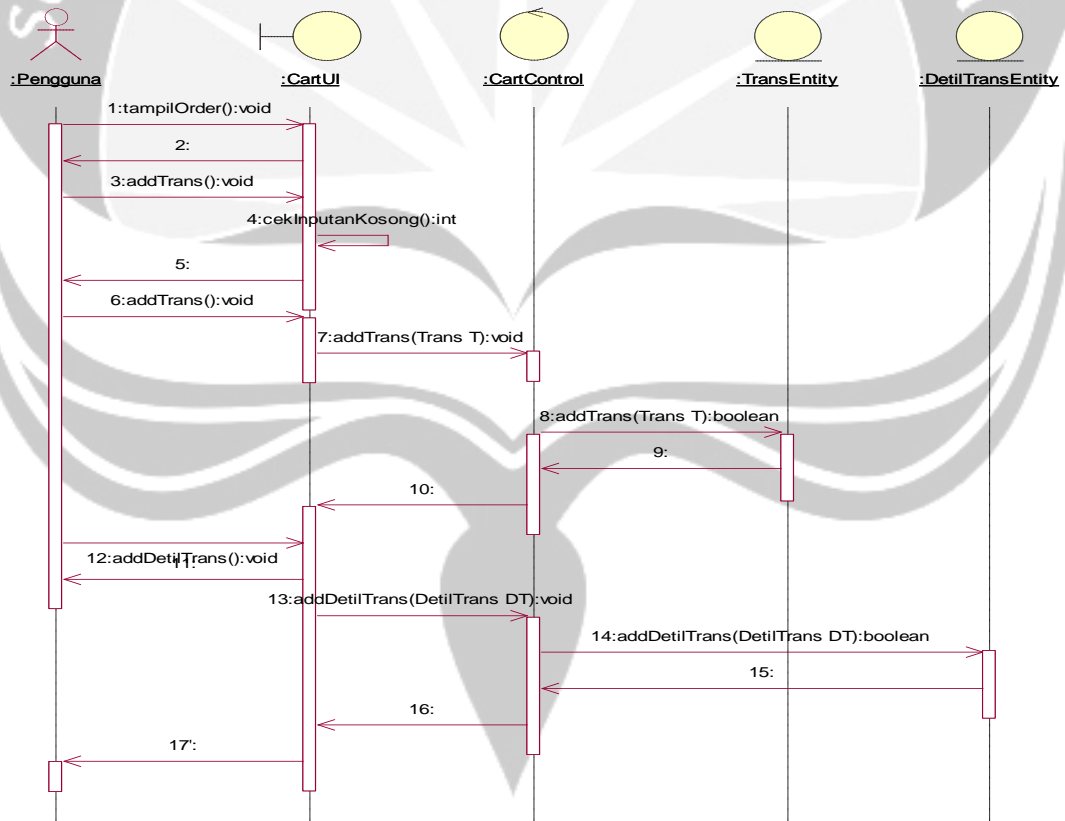
2.2.1.12. Fungsi Tambah ke Keranjang Belanja



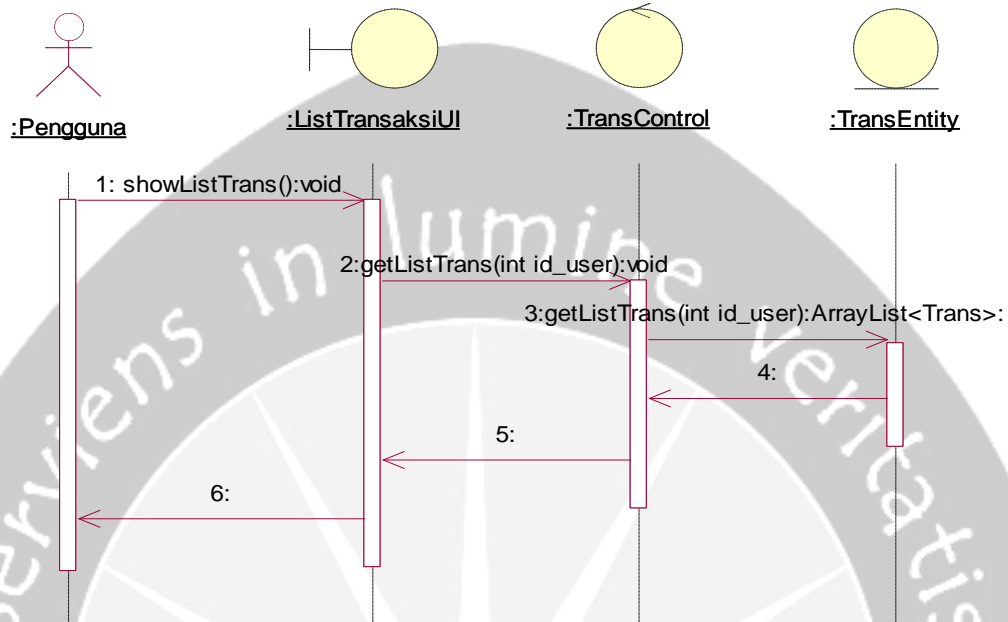
2.2.1.13. Fungsi Tampil Keranjang Belanja



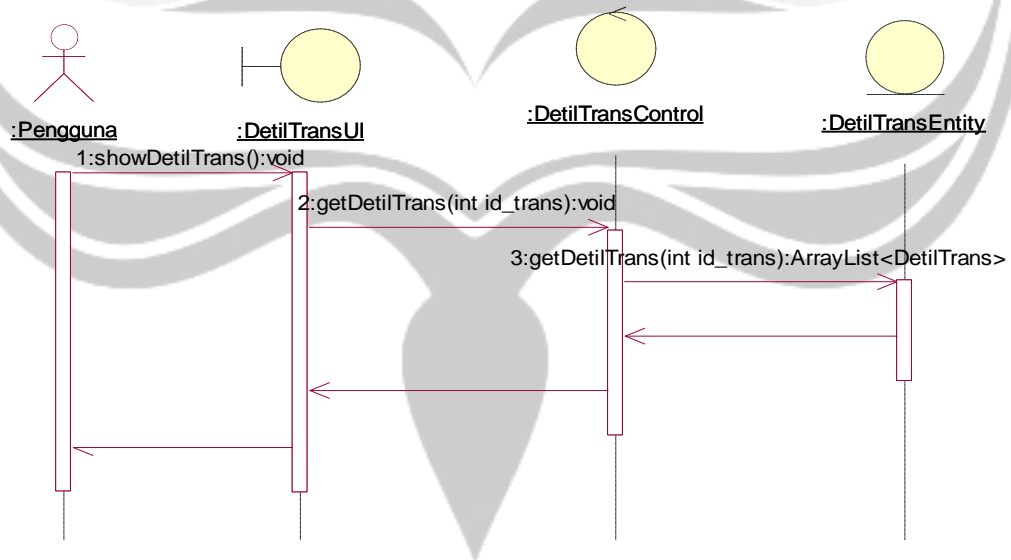
2.2.1.14. Fungsi Pesan Barang



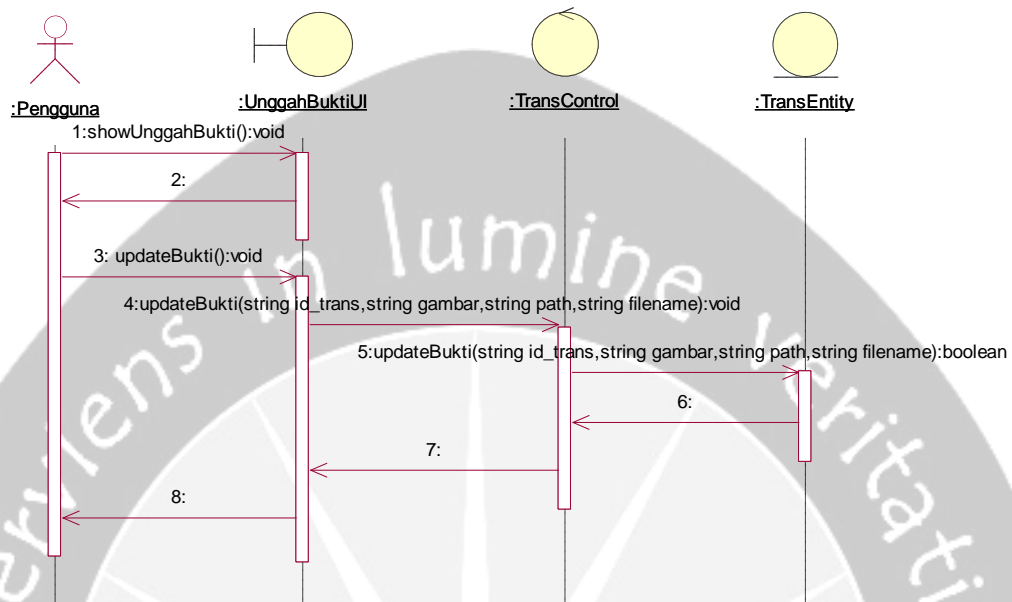
2.2.1.15. Fungsi Tampil Transaksi



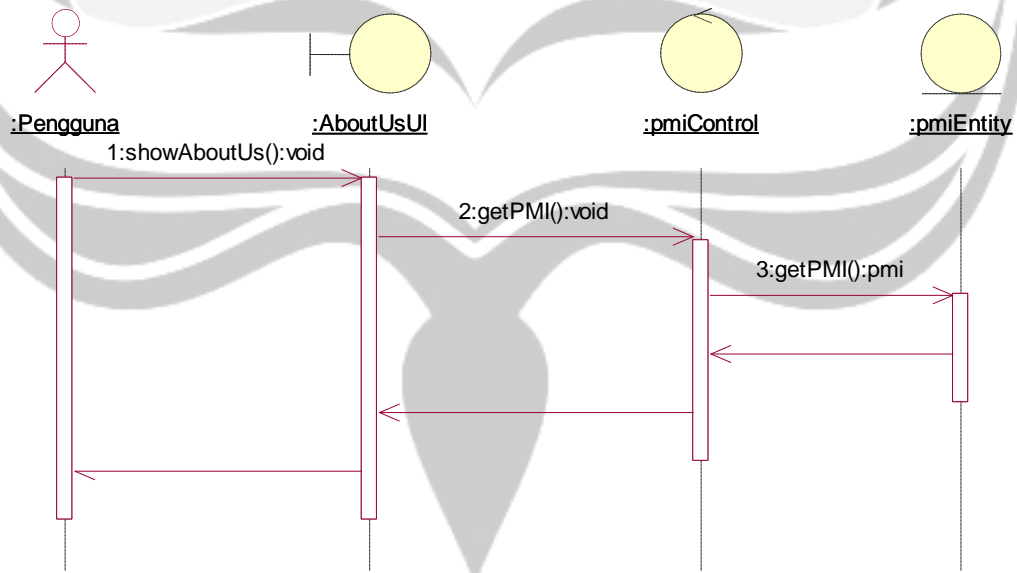
2.2.1.16. Fungsi Tampil Detil Transaksi



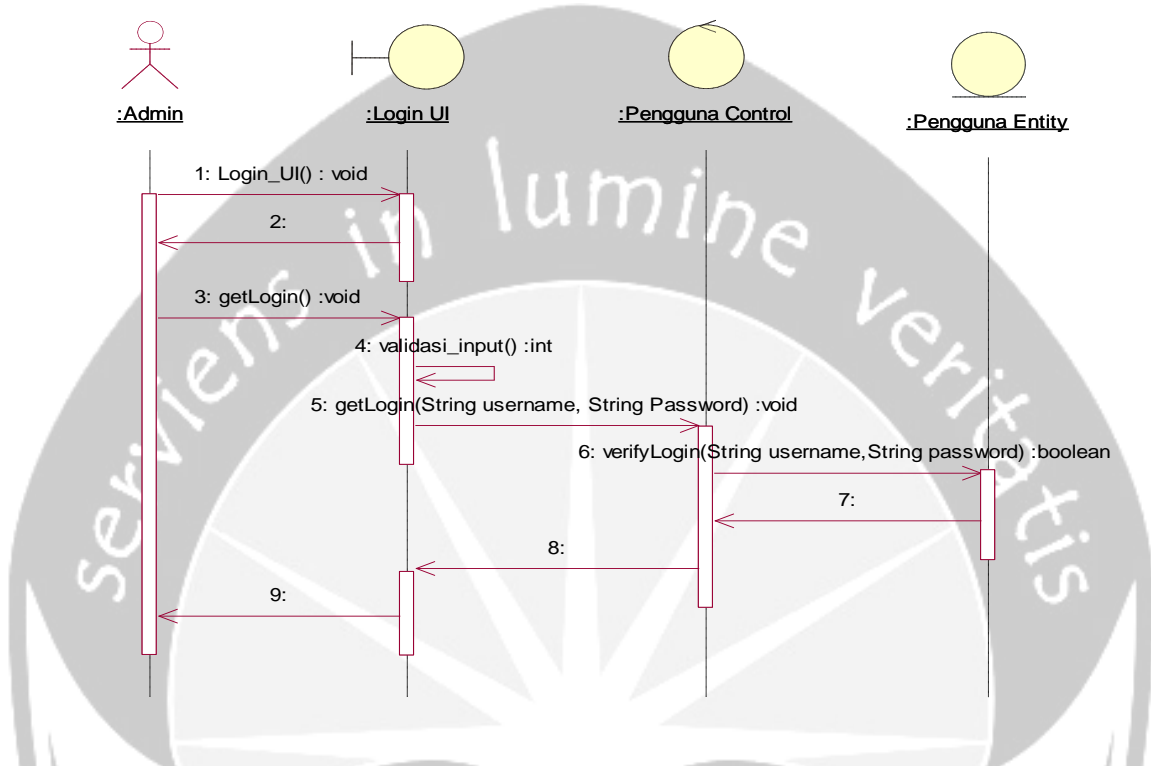
2.2.1.17. Fungsi Unggah Bukti Pembayaran



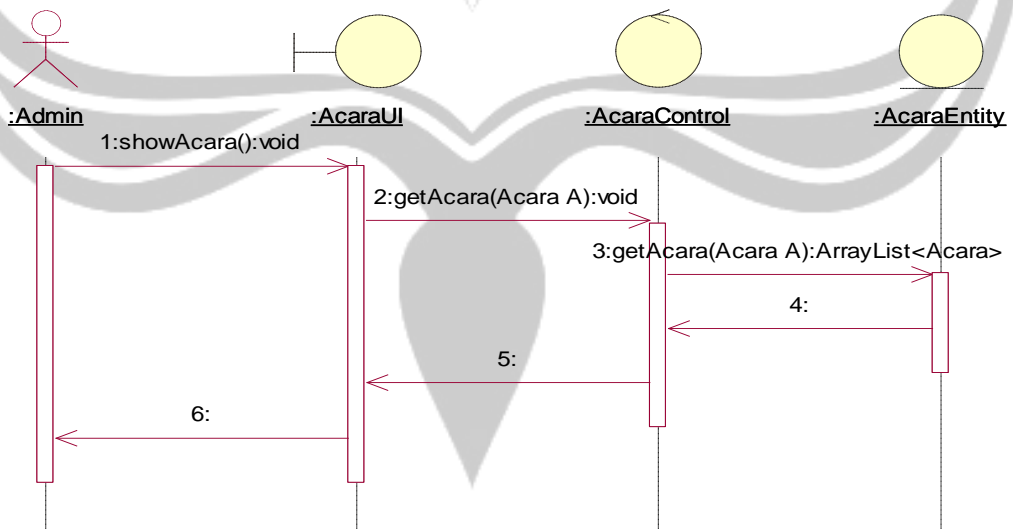
2.2.1.18. Fungsi Tampil Data PMI



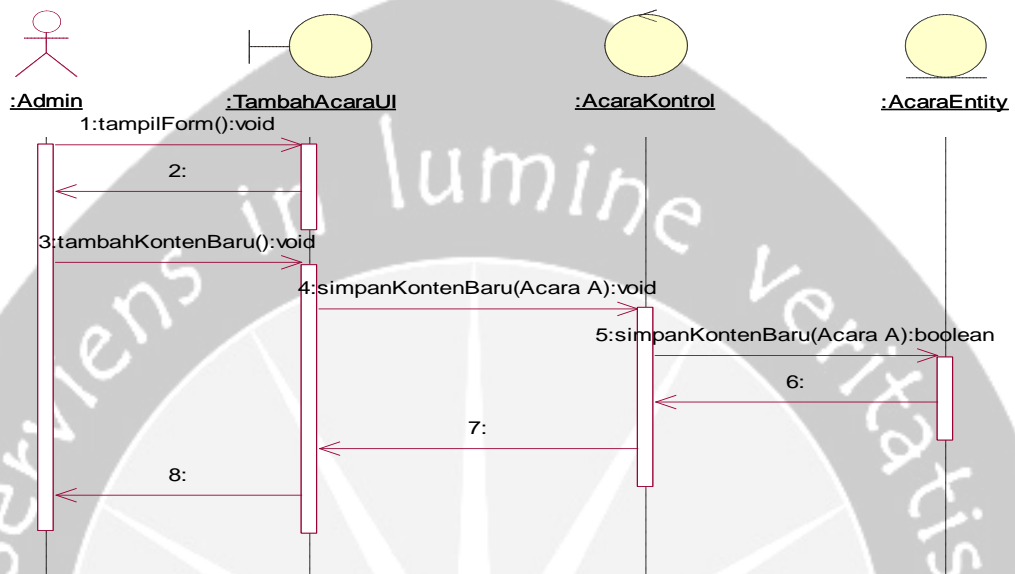
2.2.1.19. Fungsi Web Login



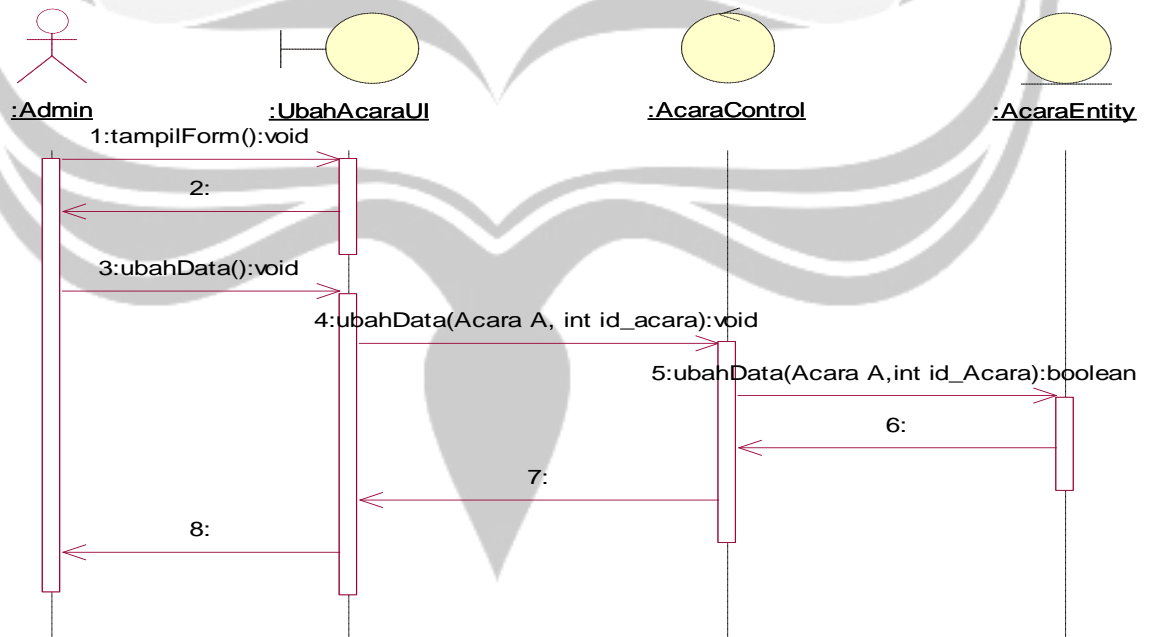
2.2.1.20. Fungsi Tampil Acara



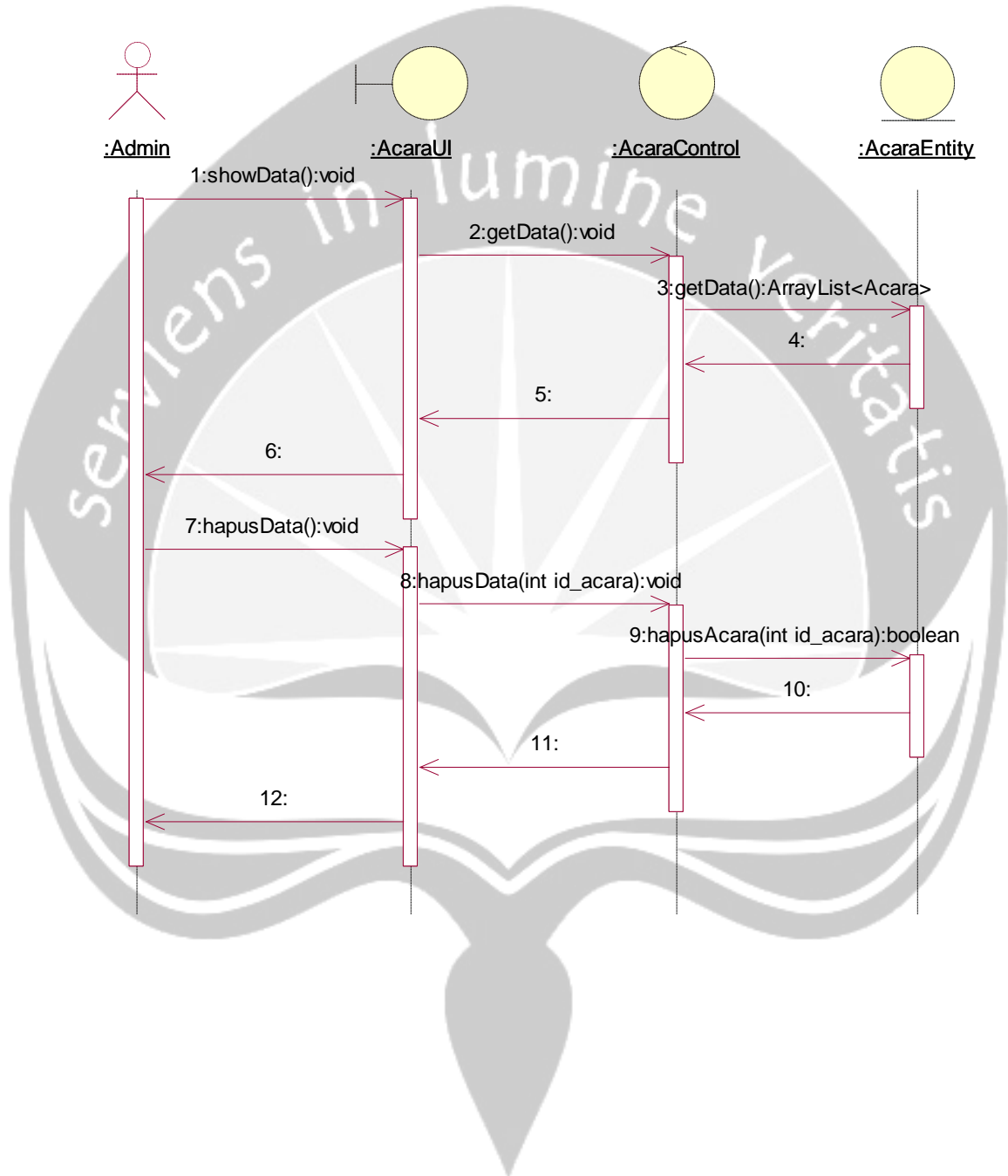
2.2.1.21. Fungsi Tambah Acara



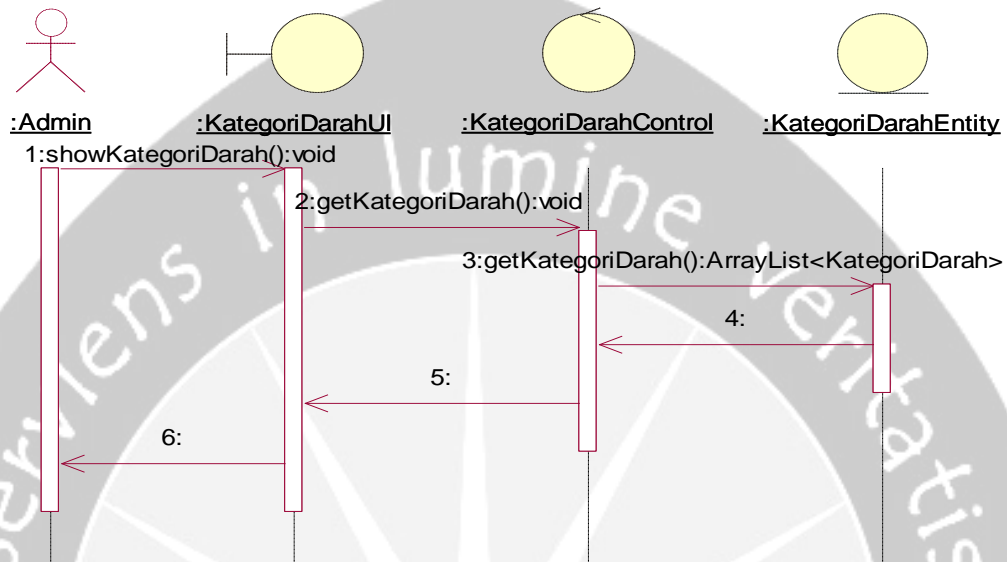
2.2.1.22. Fungsi Ubah Acara



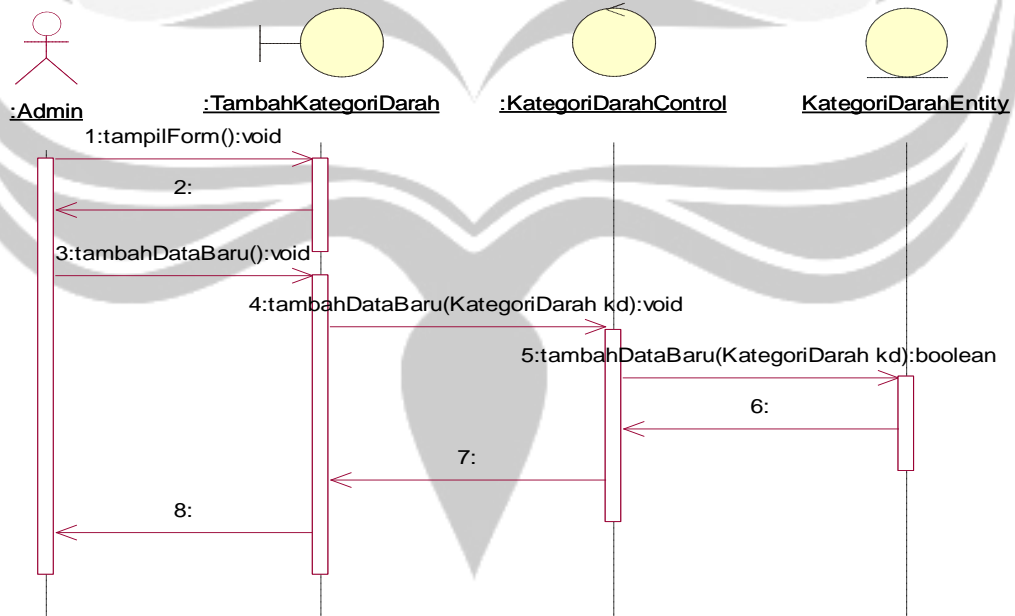
2.2.1.23. Fungsi Hapus Acara



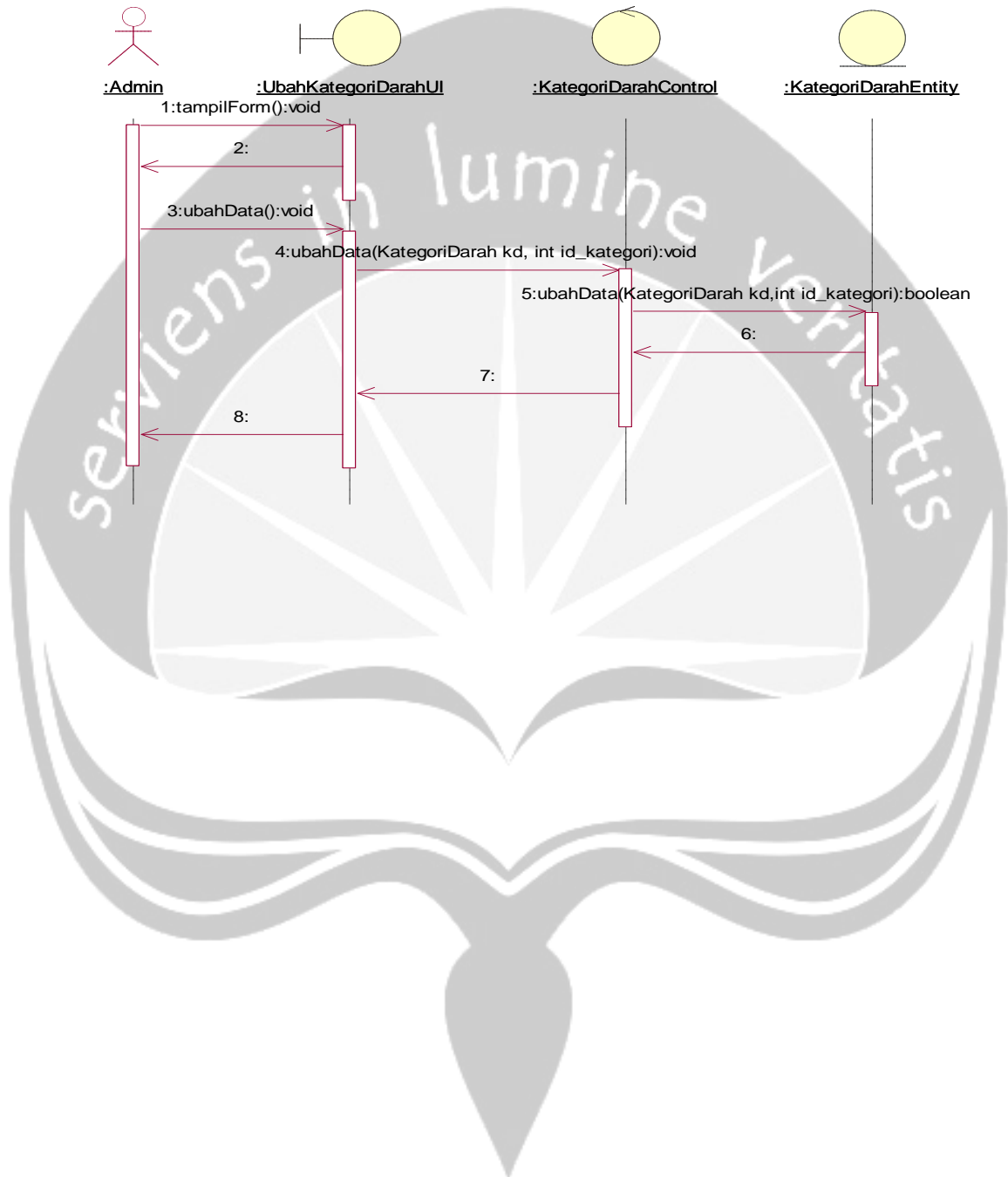
2.2.1.24. Fungsi Tampil Kategori Darah



2.2.1.25. Fungsi Tambah Kategori Darah

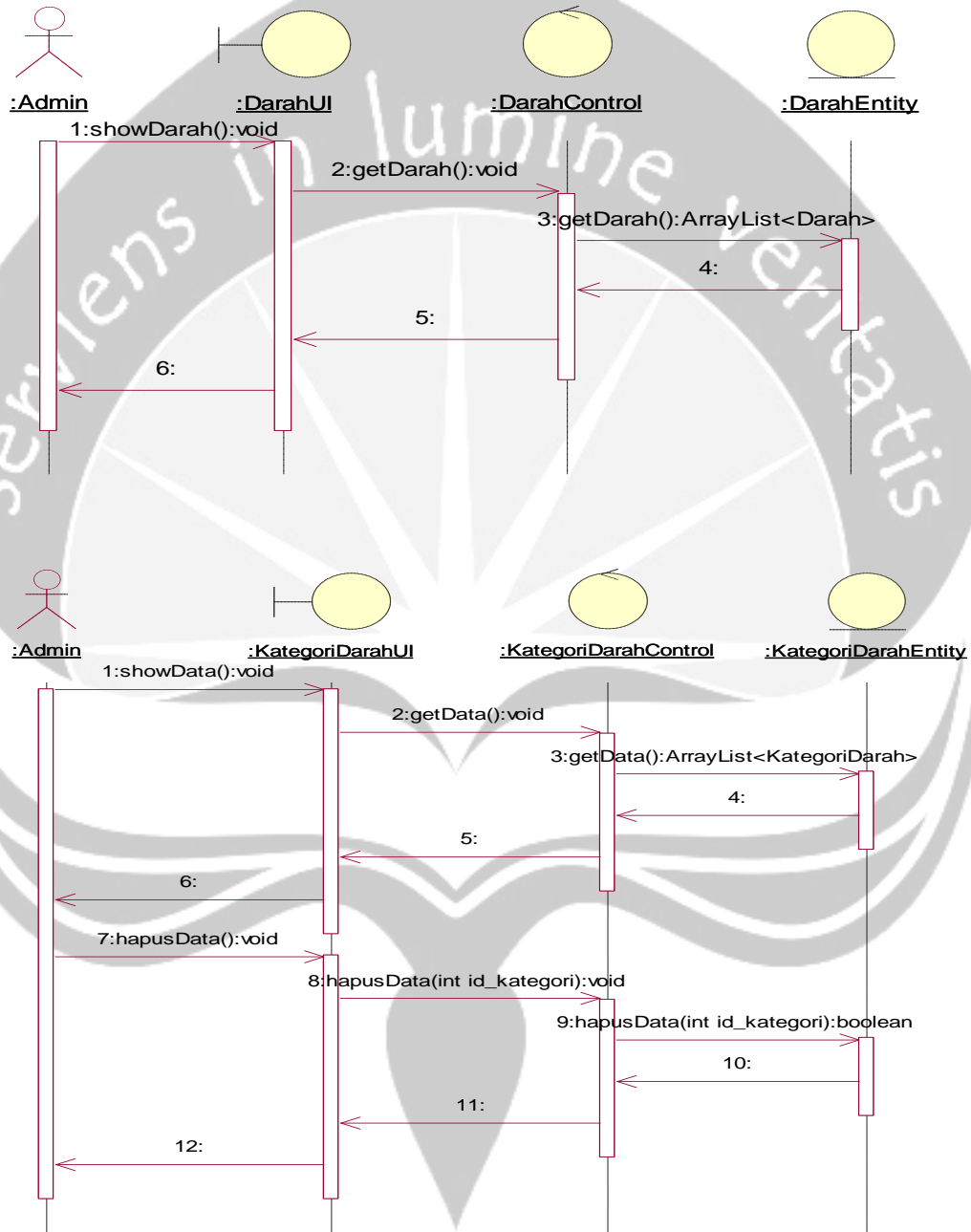


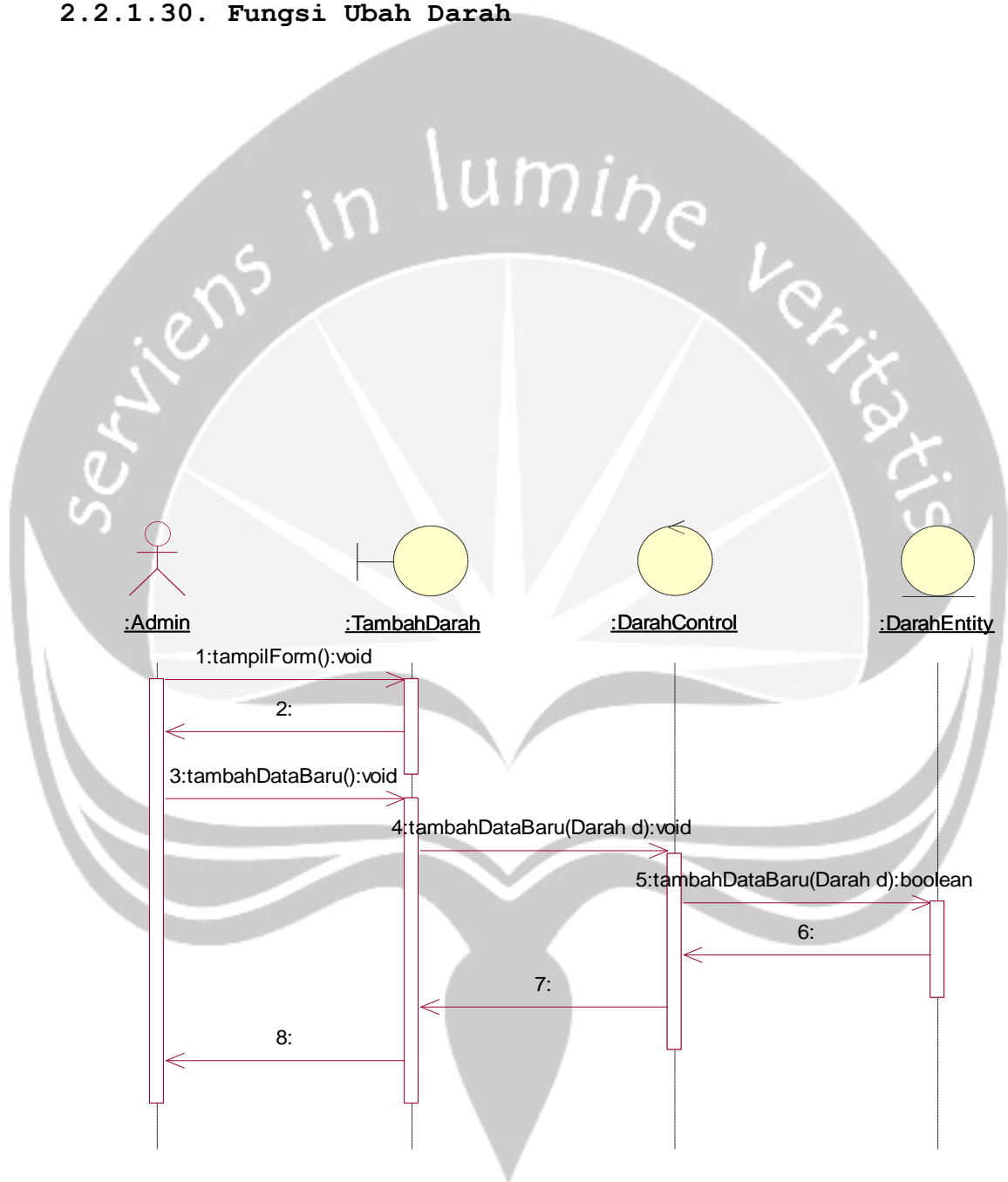
2.2.1.26. Fungsi Ubah Kategori Darah

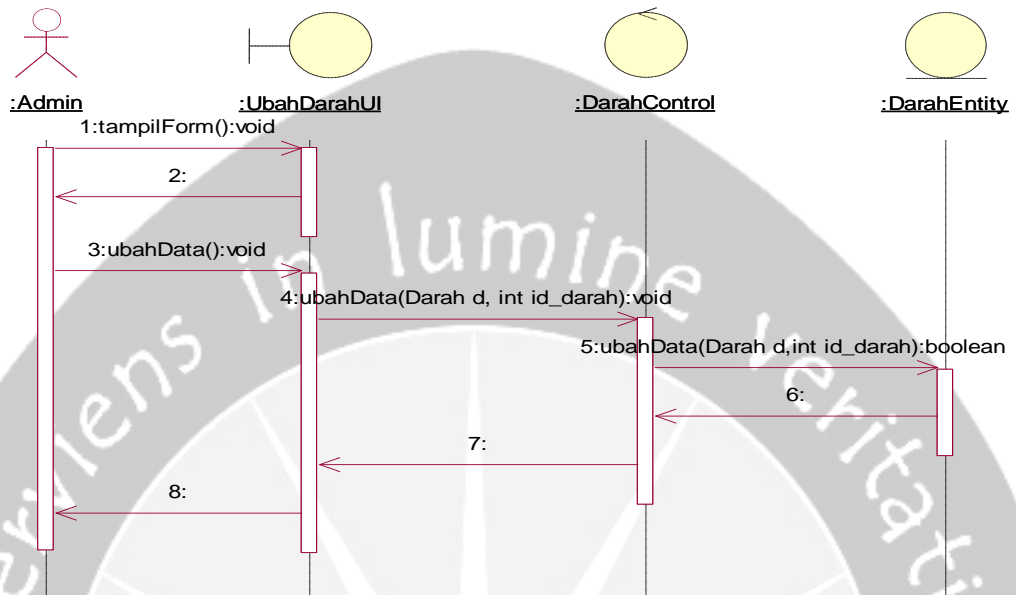


2.2.1.27. Fungsi Hapus Kategori Darah

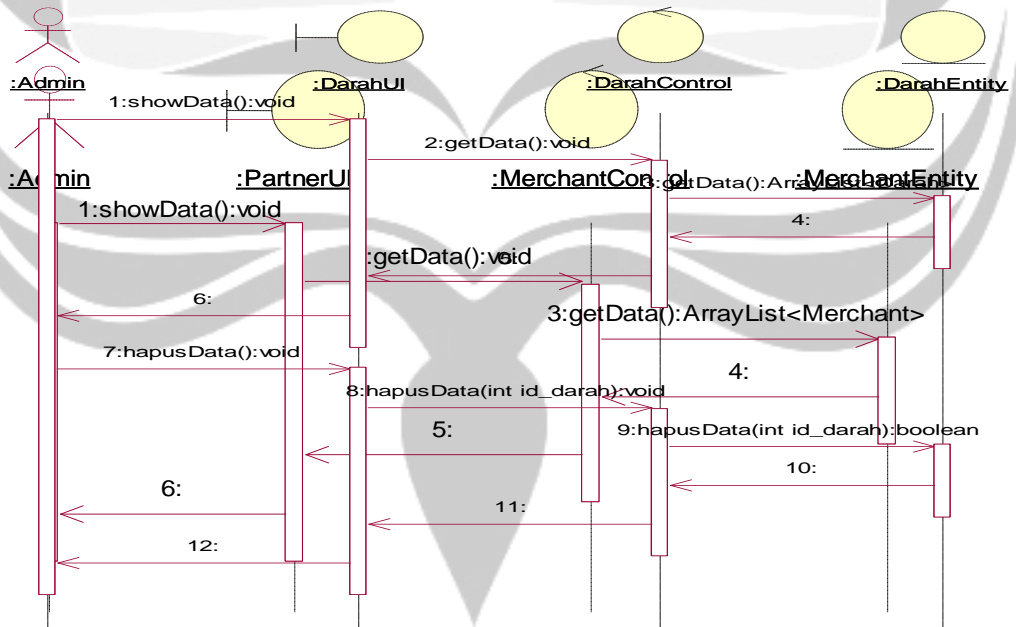
2.2.1.28. Fungsi Tampil Darah



2.2.1.29. Fungsi Tambah Darah**2.2.1.30. Fungsi Ubah Darah**

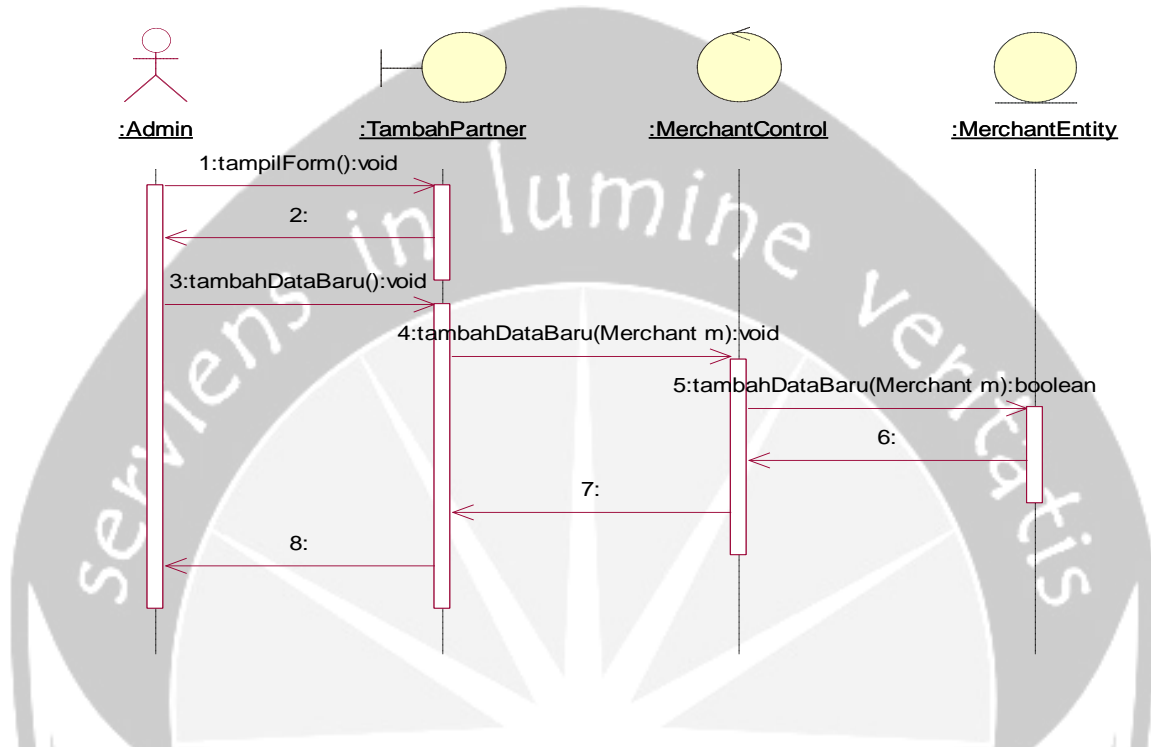


2.2.1.31. Fungsi Hapus Darah

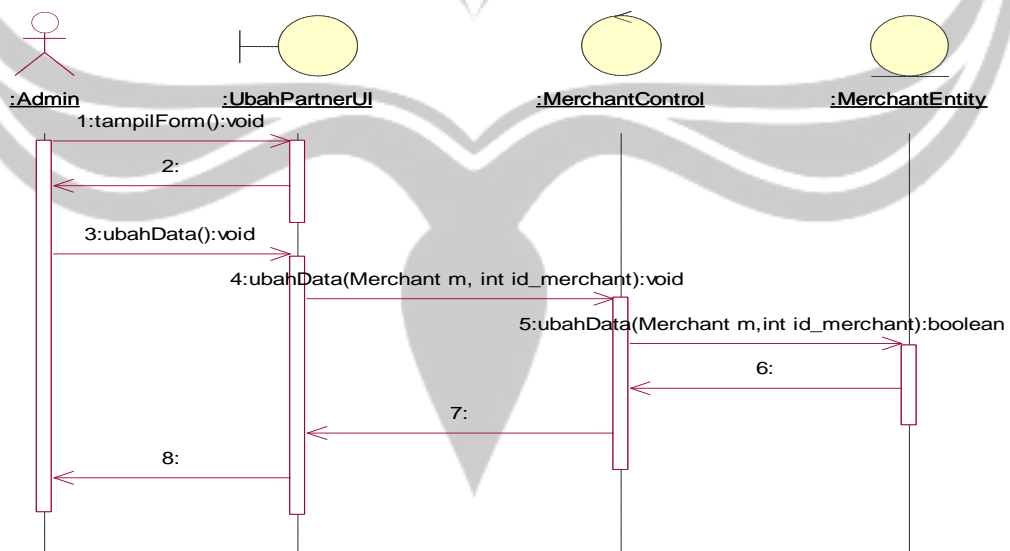


2.2.1.32. Fungsi Tampil Partner

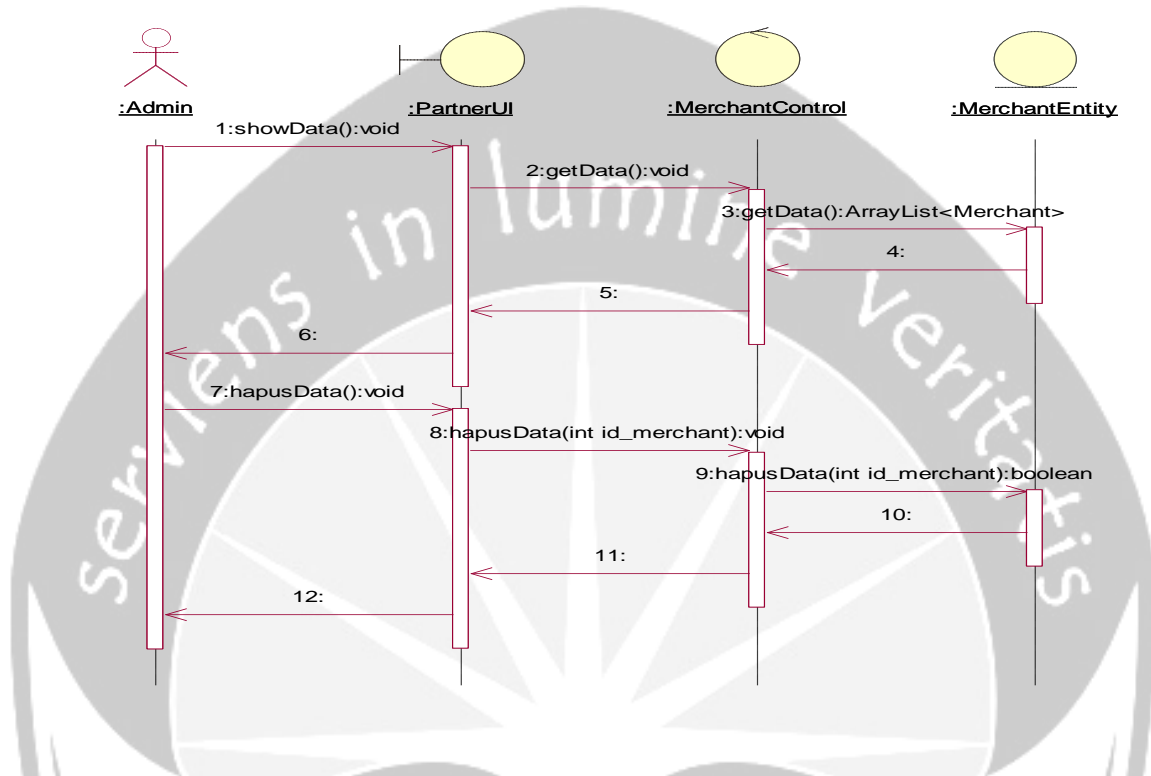
2.2.1.33. Fungsi Tambah Partner



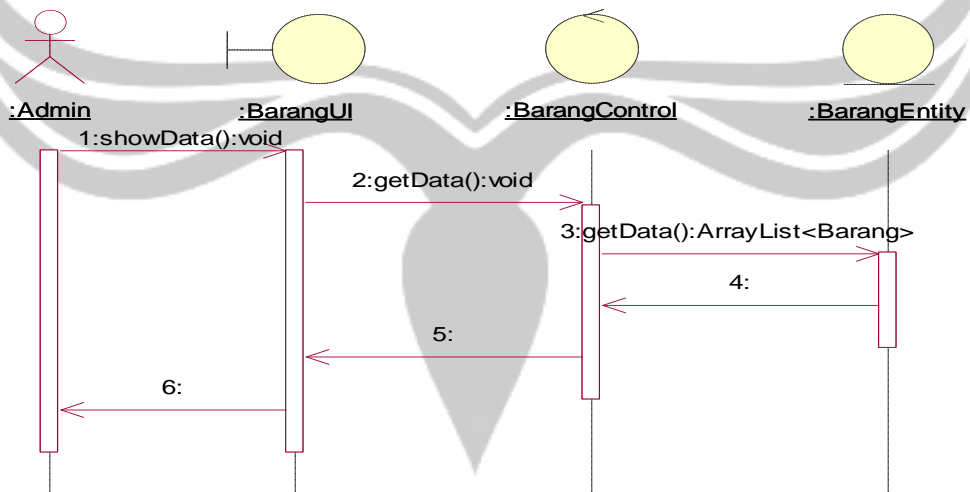
2.2.1.34. Fungsi Ubah Partner



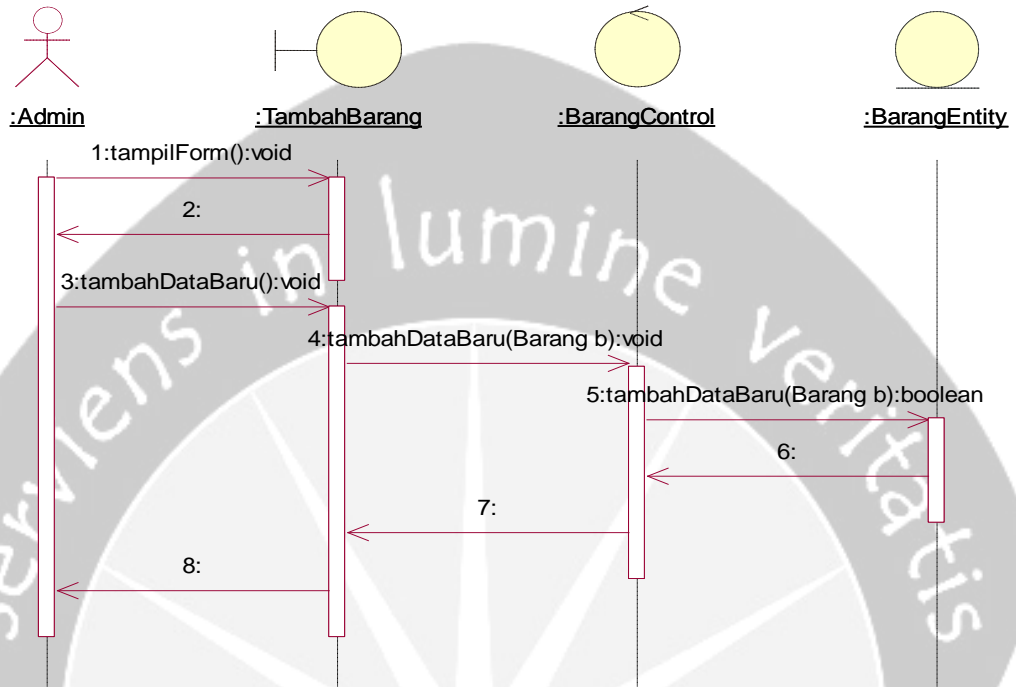
2.2.1.35. Fungsi Hapus Partner



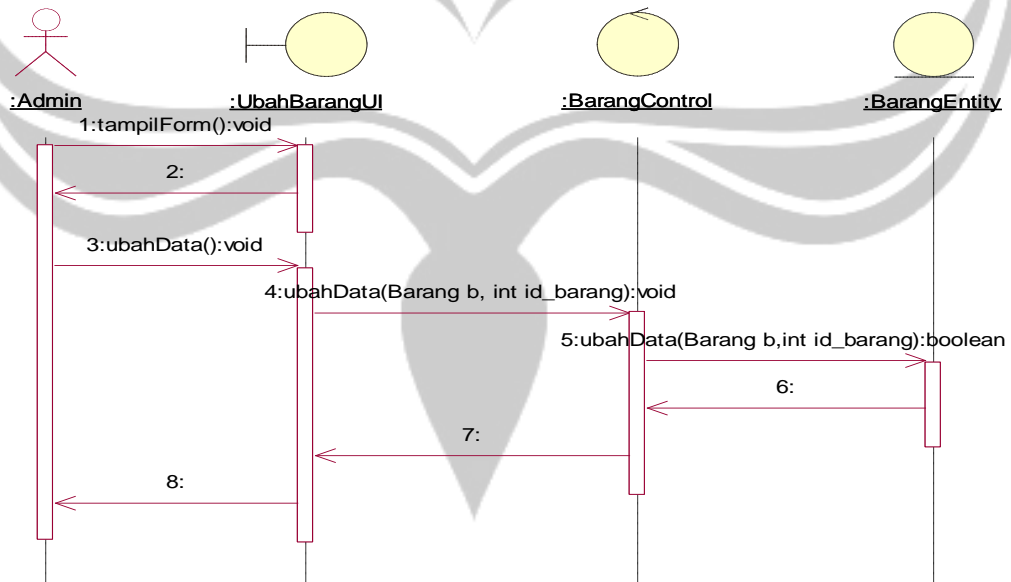
2.2.1.36. Fungsi Tampil Barang



2.2.1.37. Fungsi Tambah Barang

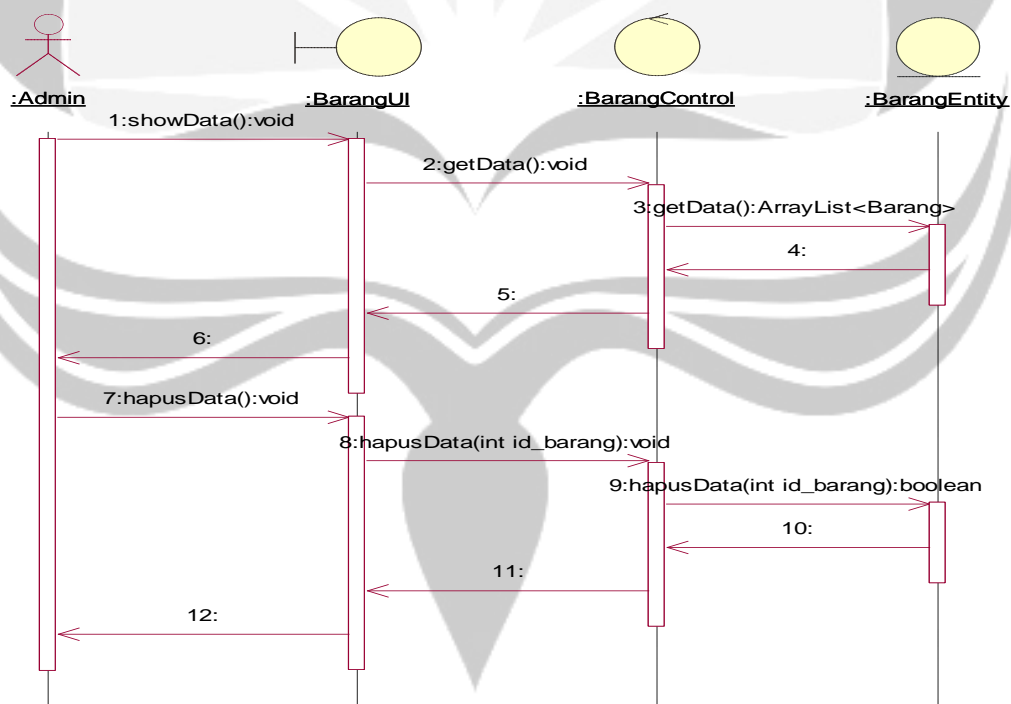


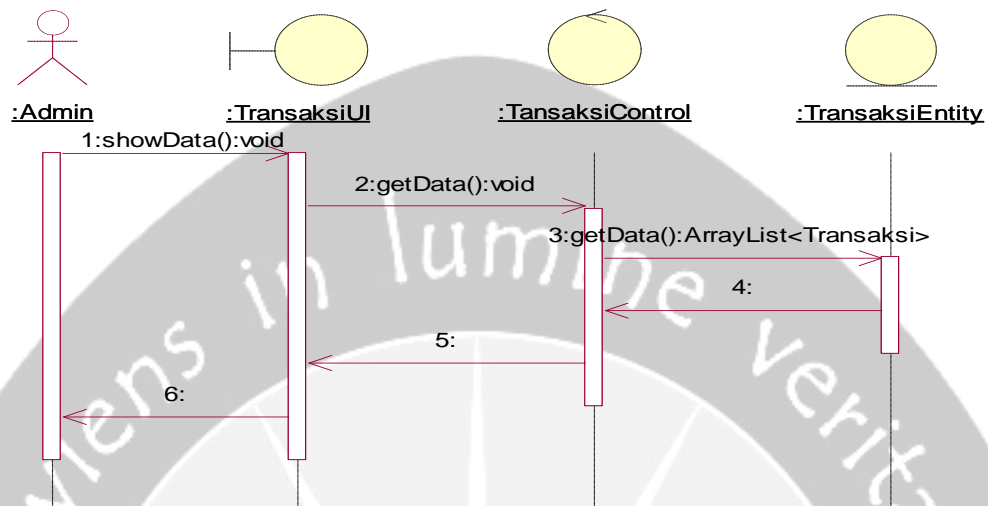
2.2.1.38. Fungsi Ubah Barang



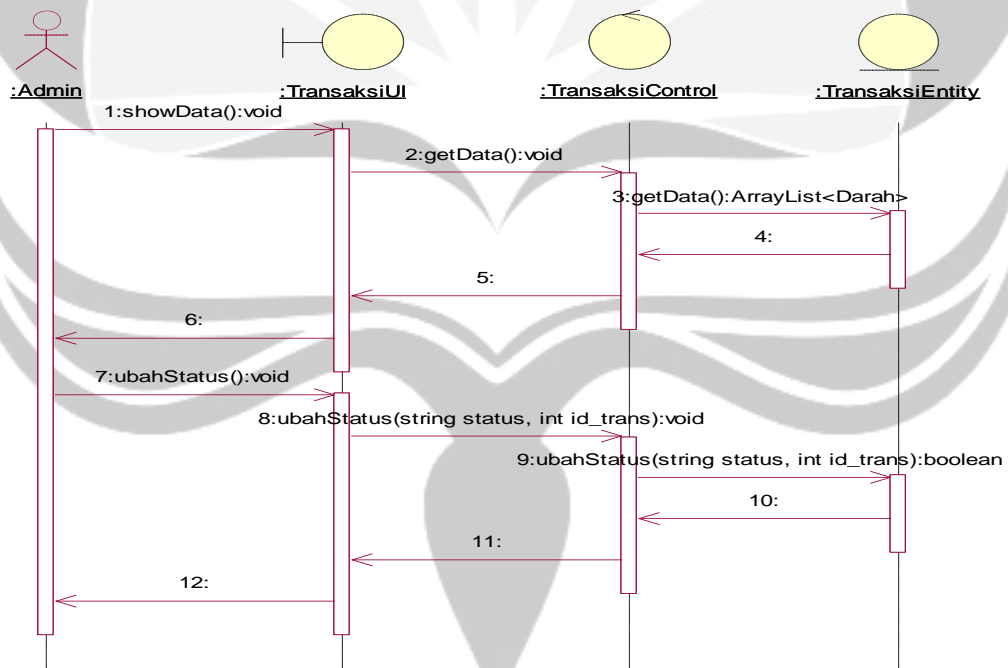
2.2.1.39. Fungsi Hapus Barang

2.2.1.40. Fungsi Tampil Transaksi

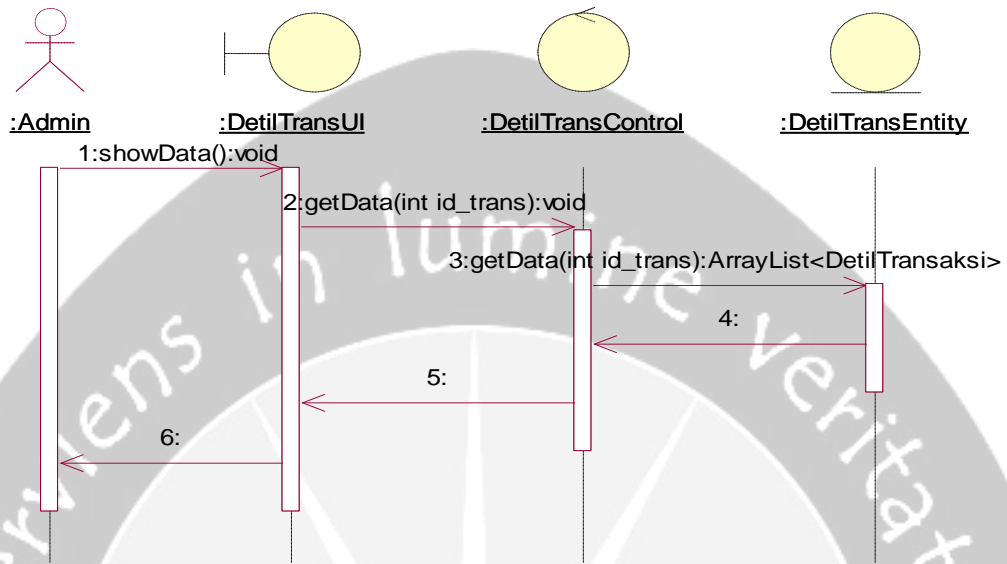




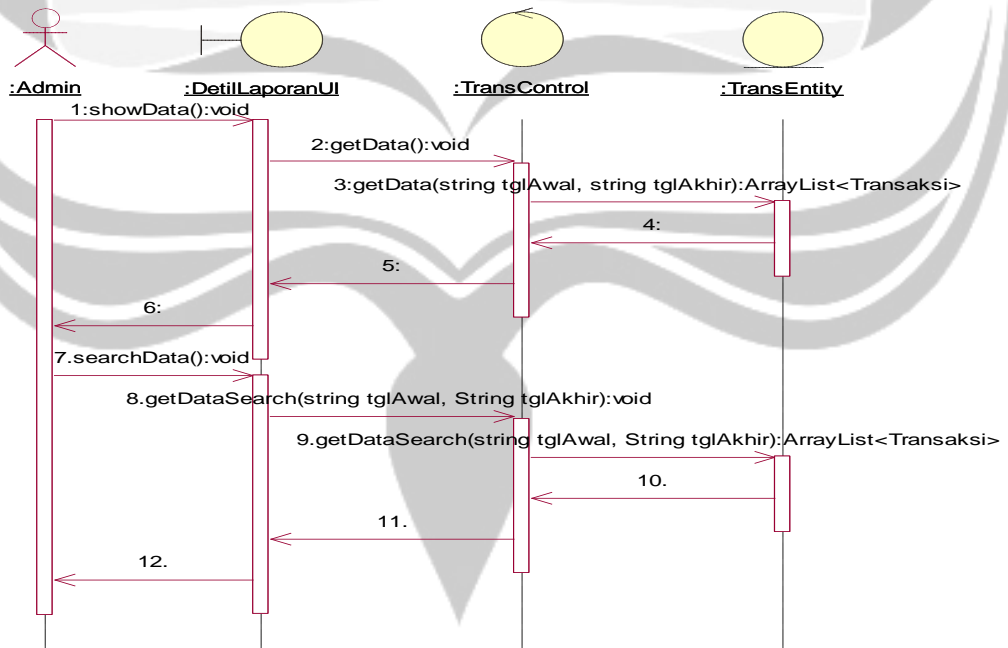
2.2.1.41. Fungsi Ubah Status Transaksi



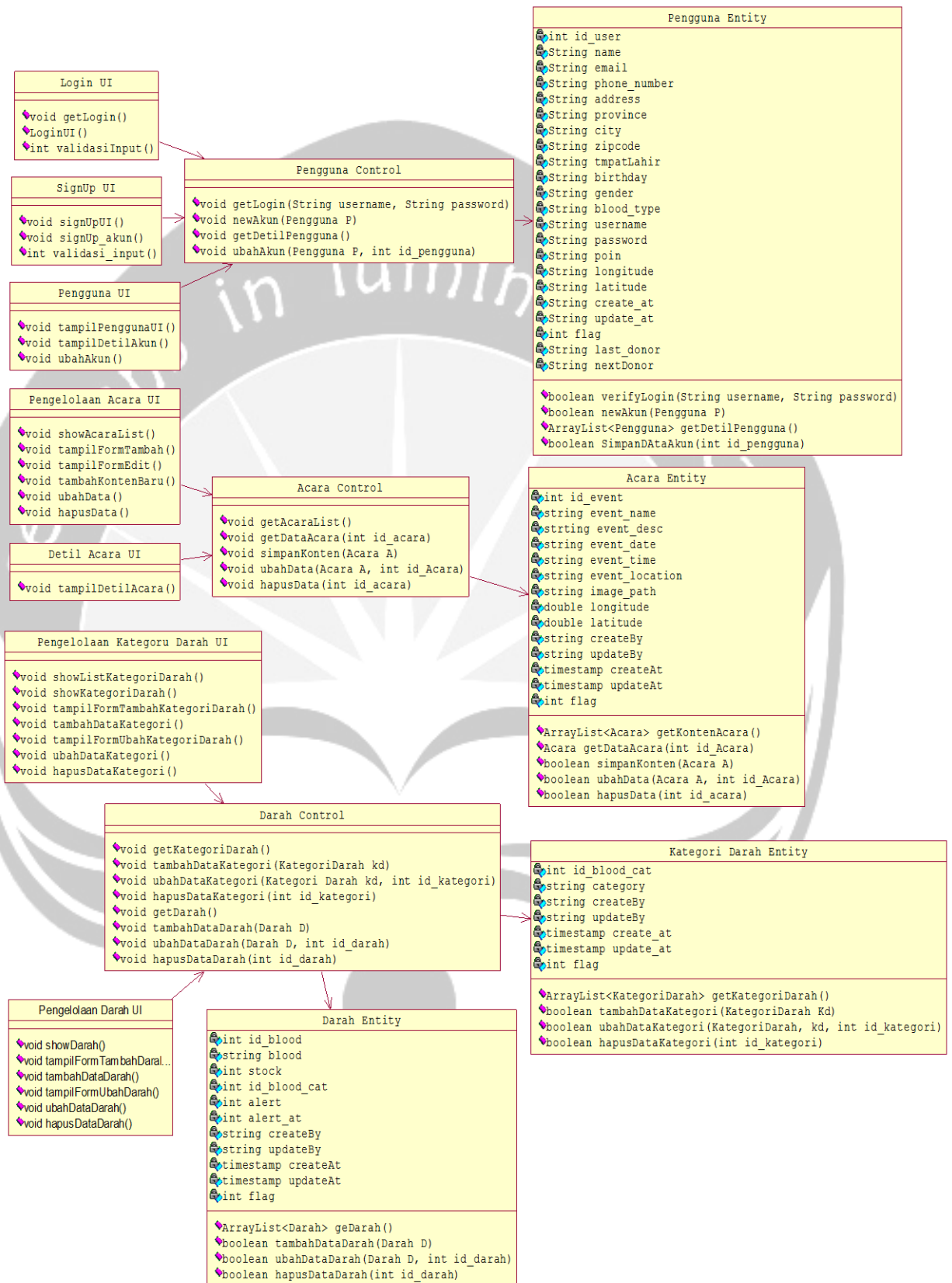
2.2.1.42. Fungsi Tampil Detil Transaksi

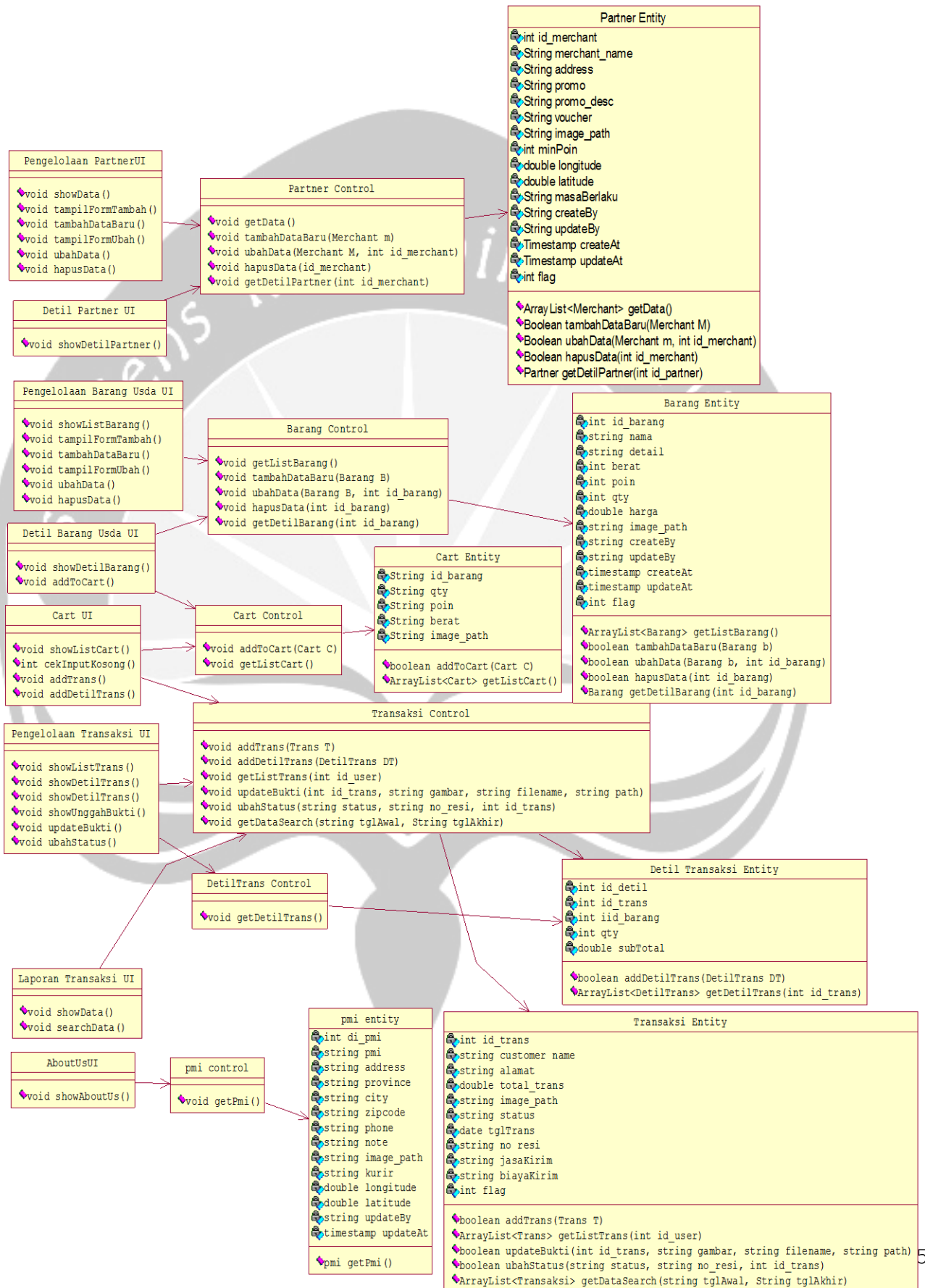


2.2.1.43. Laporan



2.2.2. Class Diagram





2.2.3. Class Diagram Spesific Description

2.2.3.1. Spesific Design Class Login UI

Login UI	<<boundary>>
LoginUI() Default Konstruktor Void getLogin() Prosedur yang digunakan untuk masuk ke dalam sistem aplikasi Int validasiInput() Fungsi yang digunakan untuk melakukan pengecekan apakah data yang diinputkan sudah benar atau tidak	

2.2.3.2. Spesific Design Class SignUp UI

SignUp UI	<<boundary>>
SignUpUI() Default konstruktor Void signUpAkun() Prosedur yang digunakan untuk melakukan pendaftaran akun Int validasi_input() Fungsi yang digunakan untuk mengecek apakah inputan sudah benar atau belum	

2.2.3.3. Spesific Design Class Pengguna UI

Pengguna UI	<<boundary>>
tampilPenggunaUI() Default konstruktor Void tampilDetilAkun() Prosedur yang digunakan untuk menampilkan detil akun pengguna yang sedang login Void ubahAkun() Prosedur yang digunakan untuk merubah data akun pengguna yang sedang login	

2.2.3.4. Spesific Design Class Kelola Acara UI

Pengelolaan Acara UI	<<boundary>>

```

AcaraUI()
Default Konstruktor
Void showAcaraList()
Prosedur yang digunakan untuk menampilkan daftar acara
Void tampilFormTambah()
Prosedur yang digunakan untuk membuka form tambah data
acara
Void tampilFormEdit()
Prosedur yang digunakan untuk membuka form ubah data
acara
Void tambahKontenBaru()
Prosedur yang digunakan untuk menambah data acara baru
Void ubahData()
Prosedur yang digunakan untuk mengubah data acara
Void hapusData()
Prosedur yang digunakan untuk menghapus data acara dengan
mengubah status menjadi 0.

```

2.2.3.5. Specific Design Class Detil Acara UI

Detil Acara UI	<<boundary>>
<pre> DetilAcaraUI() Default konstruktor Void tampilDetilAcara() Prosedur yang digunakan untuk menampilkan detil acara yang dipilih pengguna. </pre>	

2.2.3.6. Specific Design Class Kelola Kategori Darah UI

Kelola Darah UI	<<boundary>>
<pre> KelolaKategoriDarahUI() Default Konstruktor Void showListKategoriDarah() Prosedur yang digunakan untuk menampilkan daftar data kategori darah Void tampilFormTambahKategoriDarah() Prosedur yang digunakan untuk membuka form tambah data kategori darah Void tambahDataKategori() Prosedur yang digunakan untuk menambah data kategori Void tampilFormUbahDataKategori() </pre>	

Prosedur yang digunakan untuk membuka form ubah data kategori darah
 Void hapusDataKategori()
 Prosedur yang digunakan untuk menghapus data kategori darah dengan mengubah nilai keaktifan menjadi 0

2.2.3.7. Specific Design Class Kelola Darah UI

Kelola Darah UI	<<boundary>>
DarahUI() Default konstruktor Void showDarah() Prosedur digunakan untuk menampilkan data darah Void tampilFormTambahDarah() Prosedur digunakan untuk menampilkan form tambah data darah Void tambahDataDarah() Prosedur digunakan untuk menambah data darah Void tampilFormUbahDataDarah() Prosedur digunakan untuk mengubah data darah Void hapusDataDarah() Prosedur digunakan untuk menghapus data darah dengan mengubah nilai keaktifan menjadi 0	

2.2.3.8. Specific Design Class Kelola Partner UI

Kelola Partner UI	<<boundary>>
KelolaPartnerUI() Default Konstruktor Void showData() Prosedur yang digunakan untuk menampilkan daftar data partner Void tampilFormTambahData() Prosedur yang digunakan untuk membuka form tambah data Void tampilFormUbah() Prosedur yang digunakan untuk membuka form ubah data Void tambahDataBaru() Prosedur yang digunakan untuk menambah data partner ke database Void ubahData() Prosedur yang digunakan untuk mengubah data	

Void hapusData() Prosedur yang digunakan untuk menghapus data dengan mengubah nilai status menjadi 0
--

2.2.3.9. Specific Design Class Detil Partner UI

Detil Partner UI	<<boundary>>
DetilPartnerUI() Default Konstruktor Void showDetilPartner() Prosedur yang digunakan untuk menampilkan data detil partner.	

2.2.3.10. Specific Design Class Kelola Barang UI

Kelola Barang UI	<<boundary>>
kelolaBarangUI() Default Konstruktor Void showListBarang() Prosedur yang digunakan untuk menampilkan daftar barang Void tampilFormTambah() Prosedur yang digunakan untuk membuka form tambah barang Void tambahDataBaru() Prosedur yang digunakan untuk menambah data barang Void tampilFormUbah() Prosedur yang digunakan untuk membuka form ubah data Void ubahData() Prosedur yang digunakan untuk mengubah data barang Void hapusData() Prosedur yang digunakan untuk menghapus data barang dengan mengubah nilai status menjadi 0	

2.2.3.11. Specific Design Class Detil Barang UI

Detil Barang UI	<<boundary>>
DetilBarangUI() Default Konstruktor Void showDetilBarang() Prosedur yang digunakan untuk menampilkan detil barang	

Void addToCart() Prosedur yang digunakan untuk menambahkan data barang ke dalam keranjang pengguna

2.2.3.12. Specific Design Class Cart UI

Cart UI	<<boundary>>
CartUI() Default Konstruktor Void showListCart() Prosedur yang digunakan untuk menampilkan daftar list barang yang sudah diinputkan ke dalam keranjang pengguna Int cekInputKosong Fungsi yang digunakan untuk mengecek apakah inputan kosong atau tidak Void addTrans Prosedur yang digunakan untuk menambah data transaksi Void addDetilTrans Prosedur yang digunakan untuk menambah data detil transaksi	

2.2.3.13. Specific Design Class Kelola Transaksi UI

Transaksi UI	<<boundary>>
CartUI() Default Konstruktor Void showListCart() Prosedur yang digunakan untuk menampilkan daftar list barang yang sudah diinputkan ke dalam keranjang pengguna Int cekInputKosong Fungsi yang digunakan untuk mengecek apakah inputan kosong atau tidak Void addTrans Prosedur yang digunakan untuk menambah data transaksi Void addDetilTrans Prosedur yang digunakan untuk menambah data detil transaksi TransaksiUI() Default Konstruktor Void showListTrans()	

Prosedur yang digunakan untuk menampilkan daftar transaksi
 Void showDetilTrans()
 Prosedur yang digunakan untuk menampilkan data detil transaksi
 Void showUnggahBukti()
 Prosedur yang digunakan untuk mengunggah gambar bukti transaksi
 Void updateBukti()
 Prosedur yang digunakan untuk membuka halaman mengunggah bukti transaksi
 Void ubahStatus()
 Prosedur yang digunakan untuk mengubah status transaksi

2.2.3.14. Specific Design Class Laporan Transaksi UI

Laporan Transaksi UI	<<boundary>>
LaporanUI() Default Konstruktor Void showData() Prosedur yang digunakan untuk menampilkan semua data transaksi Void searchData() Prosedur yang digunakan untuk menampilkan data transaksi berdasarkan tanggal dari inputan admin	

2.2.3.15. Specific Design Class Pengguna Control

Pengguna Control	<<control>>
Void getLogin(String username, String password) Prosedur yang digunakan untuk melakukan <i>passing variable</i> username dan password dari boundary menuju entity Void newAkun(Pengguna P) Prosedur yang digunakan untuk menambahkan pengguna dengan melakukan <i>passing variable</i> P ke entity Void getDetilPengguna(Pengguna P, int id_pengguna) Prosedur yang digunakan untuk mendapatkan detil data pengguna yang sedang login Void ubahAkun(int id pengguna)	

Prosedur yang digunakan untuk mengubah data pengguna yang sedang login dengan *passing variable* Pengguna sebagai *variable* yang diubah dan *id_pengguna* sebagai *variable* penunjuk data yang akan diubah

2.2.3.16. Spesific Design Class Acara Control

Acara Control	<<boundary>>
Void getAcaraList() Prosedur yang digunakan untuk menadapatkan daftar acara Void getDataAcara(int id_acara) Prosedur yang digunakan untuk mendapatkan data detil acara yang dipilih Void simpanKonten(Acara A) Prosedur yang digunakan untuk menyimpan data acara baru Void ubahData(Acara A, int id_acara) Prosedur yang digunakan untuk mengubah data acara Void hapusData(int id_acara) Prosedur yang digunakan untuk menghapus data acara yang dipilih	

2.2.3.17. Spesific Design Class Darah Control

Darah Control	<<control>>
Void getKategoriDarah() Void tambahDataKategori(KategoriDarah kd) Void ubahDataKategori(KategoriDarah kd, int id_kategori) Void hapusDataKategori(int id_kategori) Void getDarah() Void tambahDataDarah(Darah D) Void ubahDataDarah(Darah D, int id_darah) Void hapusDataDarah(int id_darah)	

2.2.3.18. Spesific Design Class Partner Control

Partner Control	<<control>>
Void getData() Prosedur yang digunakan untuk menampilkan daftar partner Void tambahDataBaru(Merchant m) Prosedur yang digunakan untuk menambah data partner	

Void ubahData(Merchant m, int id_merchant) Prosedur yang digunakan untuk mengubah data partner yang dipilih Void hapusData(int id_merchant) Prosedur yang digunakan untuk menghapus data partner yang dipilih Void getDetilPartner(int id_merchant) Prosedur yang digunakan untuk menampilkan data detil partner yang dipilih
--

2.2.3.19. Specific Design Class Barang Control

Barang Control	<<control>>
Void getListBarang() Prosedur yang digunakan untuk menampilkan daftar barang Void tambahDataBaru(Barang B) Prosedur yang digunakan untuk menambah data barang baru Void ubahData(Barang B, int id_barang) Prosedur yang digunakan untuk mengubah data barang yang dipilih Void hapusData(int id_barang) Prosedur yang digunakan untuk menghapus data barang yang dipilih Void getDetilBarang(int id_barang) Prosedur yang digunakan untuk menghapus data barang yang dipilih	

2.2.3.20. Specific Design Class Cart Control

Cart Control	<<control>>
Void addToCart(Cart C) Prosedur yang digunakan untuk menambah data barang ke keranjang belanja pengguna Void getListCart() Prosedur yang digunakan untuk menampilkan data barang di keranjang belanja	

2.2.3.21. Specific Design Class Transaksi Control

Transaksi Control	<<control>>

```

Void addTrans(Trans T)
Prosedur yang digunakan untuk menambah data transaksi
Void addDetilTrans(DetilTrans DT)
Prosedur yang digunakan untuk menambah data detil
transaksi
Void getListTrans(int id_user)
Prosedur yang digunakan untuk menampilkan daftar
transaksi
Void updateBukti(int id_trans, string gambar, string
filename, string path)
Prosedur yang digunakan untuk mengunggah data gambar
bukti transaksi
Void ubahStatus(string status, string no_resi, int
id_trans)
Prosedur yang digunakan untuk mengubah status transaksi
Void getDataSearch(string tglAwal, string tgl_akhir)
Prosedur yang digunakan untuk menampilkan data transaksi
berdasarkan tanggal

```

2.2.3.22. Specific Design Class DetilTrans Control

Detil Trans Control	<<control>>
<pre> Void getDetilTrans(int id_trans) Prosedur yang digunakan untuk menampilkan detil transaksi dari transaksi yang dipilih </pre>	

2.2.3.23. Specific Design Class Pmi Control

Pmi Control	<<control>>
<pre> Void getPMI() Prosedur yang digunakan untuk menampilkan data pmi </pre>	

2.2.3.24. Specific Design Class Pengguna Entity

Pengguna Entity	<<entity>>
<pre> Int id_user : digunakan untuk menyimpan id_ akun String name : digunakan untuk menyimpan nama akun String email : digunakan untuk menyimpan email akun String phone_number : digunakan untuk menyimpan no telepon akun String address : digunakan untuk menyimpan alamat akun </pre>	

String province : digunakan untuk menyimpan provinsi akun
 String city : digunakan untuk menyimpan kota akun
 String zipcode : digunakan untuk menyimpan kode pos akun
 String tmpatLahir : digunakan untuk menyimpan tempat lahir akun
 String birthday : digunakan untuk menyimpan hari lahir akun
 String gender : digunakan untuk menyimpan jenis kelamin akun
 String blood_type : digunakan untuk menyimpan golongan darah akun
 String username : digunakan untuk menyimpan username akun
 String password : digunakan untuk menyimpan password akun
 String poin : digunakan untuk menyimpan poin akun
 String longitude : digunakan untuk menyimpan longitude akun
 String latitude : digunakan untuk menyimpan latitude akun
 String last_donor : digunakan untuk menyimpan tanggal terakhir donor akun
 String nextDonor : digunakan untuk menyimpan tanggal bisa melakukan donor
 String create_at : digunakan untuk menyimpan data tanggal akun dibuat
 String update_at : digunakan untuk menyimpan data tanggal akun diubah
 Int flag : digunakan untuk menyimpan status keaktifan akun

Boolean verifyLogin(String username, String password)
 Digunakan untuk melakukan verifikasi username dan password apakah data ditemukan di database
 Boolean newAkun(Pengguna P)
 Fungsi untuk menambahkan akun baru pada tabel user
 ArrayList<Pengguna> getDetilPengguna()
 Fungsi untuk mendapatkan data detil pengguna
 Boolean simanDataAkun(Pengguna P, int id_pengguna)
 Fungsi untuk menyimpan data akun yang sudah diubah

2.2.3.25. Spesific Design Class Acara Entity

Acara Entity	<<entity>>
Int id_event : digunakan untuk menyimpan id acara	
String event_name : digunakan untuk menyimpan nama acara	

```

String event_desc : digunakan untuk menyimpan deskripsi
acara
String event_date : digunakan untuk menyimpan tanggal
acara
String event_time : digunakan untuk menyimpan waktu acara
String event_location : digunakan untuk menyimpan lokasi
acara
String image_path : digunakan untuk menyimpan link gambar
acara
Double longitude : digunakan untuk menyimpan longitude
Double latitude : digunakan untuk menyimpan latitude
String createBy : digunakan untuk menyimpan tanggal data
dibuat
String updateBy : digunakan untuk menyimpan tanggal data
diubah
Timestamp createAt : digunakan untuk menyimpan waktu data
dibuat
Timestamp updateAt : digunakan untuk menyimpan waktu data
diubah
Int flag : digunakan untuk menyimpan status keaktifan
data

ArrayList<Acara> getKontenAcara()
Fungsi digunakan untuk mendapatkan daftar data Acara
Acara getDataAcara(int id_Acara)
Fungsi digunakan untuk mendapatkan detil data acara
Boolean simpanKonten(Acara A)
Fungsi yang digunakan untuk menyimpan data acara baru
Boolean ubahData(Acara A, int id_Acara)
Fungsi yang digunakan untuk mengubah data acara yang
dipilih
Boolean hapusData(int id_acara)
Fungsi yang digunakan untuk menghapus data acara yang
dipilih

```

2.2.3.26. Spesific Design Class Kategori Darah Entity

KategoriDarah Entity	<<entity>>
<pre> Int id_blood_cat : digunakan untuk menyimpan id kategori darah String category : digunakan untuk menyimpan nama kategori String createBy : digunakan untuk menyimpan tanggal data dibuat </pre>	

String updateBy : digunakan untuk menyimpan tanggal data diubah
 Timestamp create_at : digunakan untuk menyimpan waktu data dibuat
 Timestamp update_at : digunakan untuk menyimpan waktu data diubah
 Int flag : digunakan untuk menyimpan status keaktifan data

ArrayList<KategoriDarah> getKategoriDarah()
 Fungsi digunakan untuk mendapatkan data kategori darah
 Boolean tambahKategori(KategoriDarah kd)
 Fungsi yang digunakan untuk menyimpan data kategori darah
 Boolean ubahDataKategori(KategoriDarah kd, int id_kategori)
 Fungsi yang digunakan untuk menyimpan data kategori darah yang sudah diubah
 Boolean hapusDataKategori(int id_kategori)
 Fungsi yang digunakan untuk menghapus data kategori darah yang dipilih

2.2.3.27. Specific Design Class Darah Entity

Darah Entity	<<entity>>
<p>Int id_blood : digunakan untuk menyimpan id darah String blood : digunakan untuk menyimpan nama darah Int stock : digunakan untuk menyimpan stok darah Int id_blood_cat : digunakan untuk menyimpan id kategori darah Int alert : digunakan untuk menyimpan status notifikasi pada data tertentu dinyalakan atau tidak Int alert_at : digunakan untuk menyimpan data alert pada stok di angka tertentu String createBy : digunakan untuk menyimpan tanggal data dibuat String updateBy : digunakan untuk menyimpan tanggal data diubah String createAt : digunakan untuk menyimpan waktu data dibuat String updateAt : digunakan untuk menyimpan waktu data diubah Int flag : digunakan untuk menyimpan status keaktifan data</p>	
<p>ArrayList<Darah> getDarah()</p>	

Fungsi yang digunakan untuk mendapatkan data darah
 Boolean tambahDataDarah(Darah D)
 Fungsi yang digunakan untuk menyimpan data darah baru
 Boolean ubahDataDarah(Darah kd, int id_darah)
 Fungsi yang digunakan untuk menyimpan data yang sudah diubah
 Boolean hapusDataDarah(int id_darah)
 Fungsi yang digunakan untuk menghapus data yang dipilih

2.2.3.28. Specific Design Class Partner Entity

Partner Entity	<<entity>>
Int id_merchant	: digunakan untuk menyimpan data id partner
String merchant_name	: digunakan untuk menyimpan data nama partner
String address	: digunakan untuk menyimpan data alamat partner
String promo	: digunakan untuk menyimpan data promo partner
String promo_desc	: digunakan untuk menyimpan data deskripsi promo
String voucher	: digunakan untuk menyimpan data kode voucher
String image_path	: digunakan untuk menyimpan data link gambar
Int minPoin	: digunakan untuk menyimpan data poin
Double longitude	: digunakan untuk menyimpan data longitude
Double latitude	: digunakan untuk menyimpan data latitude
String masaBerlaku	: digunakan untuk menyimpan data masa berlaku promo
String createBy	: digunakan untuk menyimpan tanggal data dibuat
String updateBy	: digunakan untuk menyimpan tanggal data diubah
Timestamp createAt	: digunakan untuk menyimpan waktu data dibuat
Timestamp updateAt	: digunakan untuk menyimpan waktu data diubah
Int flag	: digunakan untuk menyimpan status keaktifan data
ArrayList<Merchant> getData()	

Fungsi yang digunakan untuk mendapatkan data partner
 Boolean tambahDataBaru(Merchant M)
 Fungsi yang digunakan untuk menambah data partner baru
 Boolean ubahData(Merchant m, int id_merchant)
 Fungsi yang digunakan untuk menyimpan data partner yang telah diubah
 Boolean hapusData(int id_merchant)
 Fungsi yang digunakan untuk menghapus data partner yang dipilih
 Partner getDetilPartner(int id_partner)
 Fungsi yang digunakan untuk mendapatkan detil data partner yang dipilih

2.2.3.29. Specific Design Class Barang Entity

Barang Entity	<<entity>>
Int id_barang : digunakan untuk menyimpan id barang String nama : digunakan untuk menyimpan nama barang String detail : digunakan untuk menyimpan detail barang Int berat : digunakan untuk menyimpan berat barang Int poin : digunakan untuk menyimpan poin Int qty: digunakan untuk menyimpan banyak barang Double harga : digunakan untuk menyimpan harga barang String image_path : digunakan untuk menyimpan link gambar barang String createBy : digunakan untuk menyimpan tanggal data dibuat Timestamp createAt : digunakan untuk menyimpan waktu data dibuat String updateBy : digunakan untuk menyimpan tanggal data diubah Timestamp updateAt : digunakan untuk menyimpan waktu data diubah Int flag : digunakan untuk menyimpan status keaktifan data	
ArrayList<Barang> getListBarang() Fungsi digunakan untuk mendapatkan data barang Boolean tambahDataBaru(Barang B) Fungsi yang digunakan untuk menyimpan data barang baru Boolean ubahData(Barang B, int id_barang) Fungsi digunakan untuk menyimpan data barang yang sudah diubah Boolean hapusData(int id_barang)	

Fungsi digunakan untuk menghapus data yang dipilih
 Barang getDetilBarang(ind id_barang)
 Fungsi digunakan untuk mendapatkan data detil barang yang
 dipilih

2.2.3.30. Specific Design Class Transaksi Entity

Transaksi Entity	<<entity>>
<p>Int id_trans : digunakan untuk menyimpan id transaksi String customer_name : digunakan untuk menyimpan nama pembeli String alamat : digunakan untuk menyimpan alamat pengiriman String total_trans : digunakan untuk menyimpan total transaksi String image_path : digunakan untuk menyimpan link gambar bukti transaksi String status : digunakan untuk menyimpan data status transaksi Date tglTrans : digunakan untuk menyimpan tanggal transaksi String noResi : digunakan untuk menyimpan no resi String jasaKirim : digunakan untuk menyimpan jasa pengiriman String biayaKirim : digunakan untuk menyimpan biaya kirim Int flag : digunakan untuk menyimpan status keaktifan transaksi</p>	
<p>Boolean addTrans(Trans T) Fungsi digunakan untuk menyimpan data transaksi baru ArrayList<Trans> getListTrans(int id_user) Fungsi digunakan untuk mendapatkan data transaksi Boolean updateBukti(int id_trans, string gambar, string filename, string path) Fungsi digunakan untuk menyimpan data yang sudah diubah Boolean ubahStatus(String status, string noResi, int id_trans) Fungsi digunakan untuk menyimpan data status yang sudah diubah ArrayList<Transaksi> getDataSearch(String tglAwal, String tglAkhir) Fungsi digunakan untuk mendapatkan data transaksi dengan kondisi tanggal diantara tglAwal dan tglAkhir</p>	

2.2.3.31. Spesific Design Class Detil Transaksi Entity

Detil Transaksi Entity	<<entity>>
Int id_detil : digunakan untuk menyimpan id detil transaksi	
Int id_trans : digunakan untuk menyimpan id transaksi	
Int id_barang : digunakan untuk menyimpan id barang	
Int qty : digunakan untuk menyimpan jumlah barang	
Double subtotal : digunakan untuk menyimpan subtotal	
Boolean addDetilTrans(DetilTrans DT)	
Fungsi digunakan untuk menyimpan data baru	
ArrayList<DetilTrans>getDetilTrans(int id_trans)	
Fungsi yang digunakan untuk mendapatkan data detil transaksi	

2.2.3.32. Spesific Design Class PMI Entity

PMI Entity	<<entity>>
Int id_pmi : digunakan untuk menyimpan id_pmi	
String pmi : digunakan untuk menyimpan nama pmi	
String address : digunakan untuk menyimpan alamat pmi	
String province : digunakan untuk menyimpan provinsi pmi	
String city : digunakan untuk menyimpan kota pmi	
String zipcode : digunakan untuk menyimpan kodepos pmi	
String phone : digunakan untuk menyimpan no telepon pmi	
String note : digunakan untuk menyimpan info pmi	
String image_path : digunakan untuk menyimpan link gambar pmi	
String kurir : digunakan untuk menyimpan harga pengiriman via kurir	
String longitude : digunakan untuk menyimpan longitude	
String latitude : digunakan untuk menyimpan latitude	
String updateBy : digunakan untuk menyimpan data tanggal data diubah	
Timestamp updateAt : digunakan untuk menyimpan data jam data diubah	
Pmi getPmi()	
Prosedur digunakan untuk mendapatkan data pmi	

Bab III

Perancangan Data

3.1. Dekomposisi Data

3.1.1. Deskripsi Entitas Tabel Role

Nama	Tipe	Panjang	Keterangan
id_role	Integer	11	Primary Key
role	String	50	
created	Timestamp		
createdBy	String	50	
updated	Timestamp		
updatedBy	String	50	
flag	integer	1	

3.1.2. Deskripsi Entitas Tabel Admin

Nama	Tipe	Panjang	Keterangan
id_admin	Integer	11	Primary Key
Id_role	Integer	1	Foreign Key
nama	String	100	
username	String	50	
password	Text		
create_at	Timestap		
createdBy	String	50	
updated_at	String		
updatedBy	timestamp	50	
flag	integer	1	

3.1.3. Deskripsi Entitas Tabel User

Nama	Tipe	Panjang	Keterangan
Id_user	Integer	11	Primary Key
name	String	100	
Email	String	100	
Phone_number	String	100	
Address	Text		
City	String	100	
Province	String	100	
Zipcode	String	100	
tempatLahir	String	100	
Birthday	Date		
gender	String	20	
Blood_type	String	10	
Poin	Integer	11	
Username	String	50	
Password	Text		
Longitude	Double	20	
Latitude	Double	20	
Create_at	Timestamp		
Update_at	Timestamp	50	
flag	integer	1	

3.1.4. Deskripsi Entitas Tabel Hist Donor

Nama	Tipe	Panjang	Keterangan
Id_hist_donor	Integer	11	Primary Key
Date	date		
Next_donor_date	date		

Id_user	integer	11	
---------	---------	----	--

3.1.5. Deskripsi Entitas Tabel Image

Nama	Tipe	Panjang	Keterangan
Id_image	Integer	11	Primary Key
Image_name	String	100	
Image_path	String	100	
jenis	String	100	
flag	integer	1	

3.1.6. Deskripsi Entitas Tabel Voucher

Nama	Tipe	Panjang	Keterangan
Id_voucher	Integer	11	Primary Key
Voucher	String	50	
poin	Integer	11	

3.1.7. Deskripsi Entitas Tabel Prof_PMI

Nama	Tipe	Panjang	Keterangan
Id_pmi	Integer	11	Primary Key
Pmi	String	50	
Address	string	100	
City	string	100	
Province	String	100	
zipcode	string	100	
Phone	string	100	
Note	Text		

Nama	Tipe	Panjang	Keterangan
Id_image	integer	11	Foreign key
Kurir	double	20	
Longitude	double	20	
Latitude	Double	20	
Create_at	Timestamp		
createdBy	String	50	
Update_at	Timestamp		
updatedBy	String	50	
flag	Integer	1	

3.1.8. Deskripsi Entitas Tabel Event

Nama	Tipe	Panjang	Keterangan
Id_event	Integer	11	Primary Key
Event_name	String	50	
Event_desc	Text		
Event_date	datetime	100	
Event_loc	string	100	
Event_time	Datetime		
Id_image	integer	11	Foreign Key
Longitude	double	20	
latitude	double	20	
Create_at	Timestamp		
createdBy	String	50	
Update_at	Timestamp		
updatedBy	String	50	

Nama	Tipe	Panjang	Keterangan
flag	integer	1	

3.1.9. Deskripsi Entitas Tabel Blood_Cat

Nama	Tipe	Panjang	Keterangan
Id_blood_cat	Integer	11	Primary Key
category	String	50	
Create_at	Timestamp		
createdBy	String	50	
Update_at	Timestamp		
updatedBy	String	50	
flag	Integer	1	

3.1.10. Deskripsi Entitas Tabel Blood

Nama	Tipe	Panjang	Keterangan
Id_blood	Integer	11	Primary Key
Id_blood_cat	integer	11	Foreign Key
Blood	String	20	
Stock	integer	11	
Alert	integer	1	
Alert_at	integer	11	
Create_at	Timestamp		
createdBy	String	50	
Update_at	Timestamp		
updatedBy	String	50	

Flag	Integer	1	
------	---------	---	--

3.1.11. Deskripsi Entitas Tabel Merchant

Nama	Tipe	Panjang	Keterangan
Id_merchant	Integer	11	Primary Key
Merchant_name	String	50	
Address	string	100	
Promo	string	100	
Promo_desc	Text		
Voucher	String	8	
Id_image	Integer	11	
minPoin	Integer	11	
masaBerlaku	Date		
Longitude	Double	20	
latitude	Double	20	
Create_at	Timestamp		
createdBy	String	50	
Update_at	Timestamp		
updatedBy	String	50	
flag	Integer	1	

3.1.12. Deskripsi Entitas Tabel Barang

Nama	Tipe	Panjang	Keterangan
Id_stuff	Integer	11	Primary Key
Name	String	50	
Price	Double	20	
Stock	Integer	11	
Berat	Float	11	

Nama	Tipe	Panjang	Keterangan
Detail	Text		
Poin	Integer	11	
Id_image	Integer	11	Foreign Key
Create_at	Timestamp		
createdBy	String	50	
Update_at	Timestamp		
updatedBy	String	50	
flag	Integer	1	

3.1.13 Deskripsi Entitas Tabel Trans

Nama	Tipe	Panjang	Keterangan
Id_trans	Integer	11	Primary Key
Id_user	integer	11	Foreign Key
Id_image	Integer	11	Foreign Key
Alamat	String	100	
Tot_trans	Double	20	
Status	String	50	
date	Date		
No_resi	String	20	
flag	Integer	1	

3.1.14. Deskripsi Entitas Tabel Detil_Trans

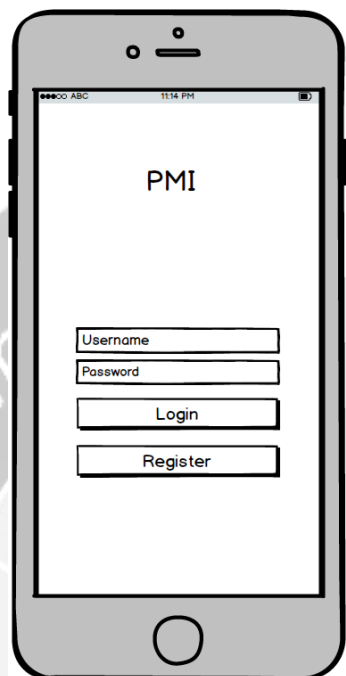
Nama	Tipe	Panjang	Keterangan
id_detil_trans	Integer	11	Primary Key
id_trans	integer	11	Foreign key
Id_stuff	Integer	11	Foreign key

Qty	Integer	11	
Subtotal	Float	11	



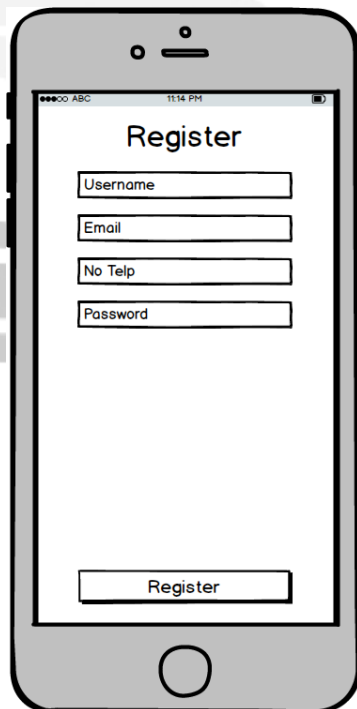
Bab IV**Deskripsi Perancangan Antarmuka**

4.1. Login



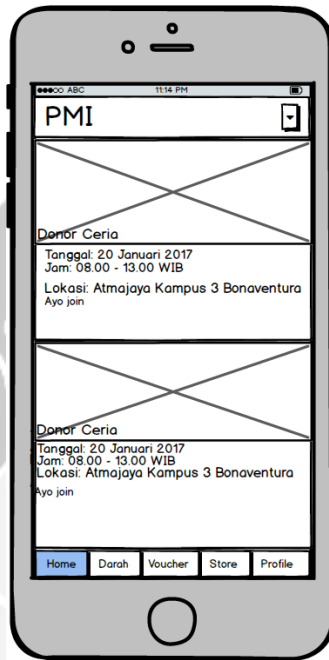
A smartphone mockup displaying the login screen for PMI. The screen shows the title "PMI" at the top. Below the title are two input fields: "Username" and "Password". Underneath these fields are two buttons: "Login" and "Register". The background of the screen is white, and the phone's status bar at the top shows "ABC" and "11:14 PM".

4.2. Signup



A smartphone mockup displaying the register screen. The screen shows the title "Register" at the top. Below the title are four input fields: "Username", "Email", "No Telp", and "Password". At the bottom of the screen is a "Register" button. The background of the screen is white, and the phone's status bar at the top shows "ABC" and "11:14 PM".

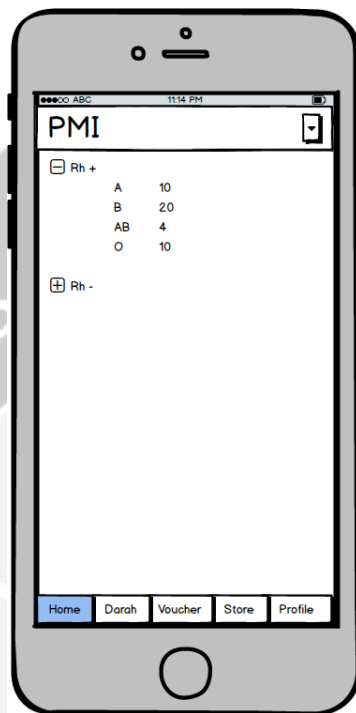
4.3. Daftar Konten Acara PMI



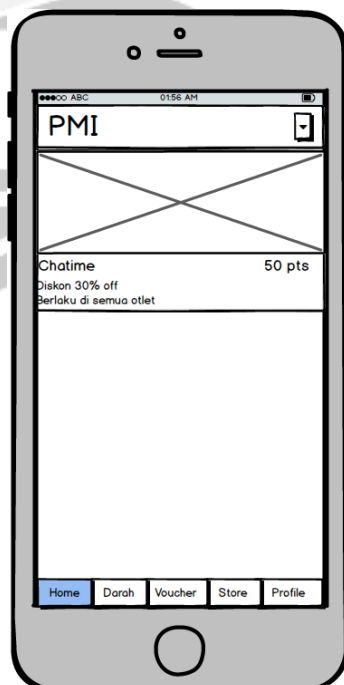
4.4. Deskripsi Konten Acara PMI



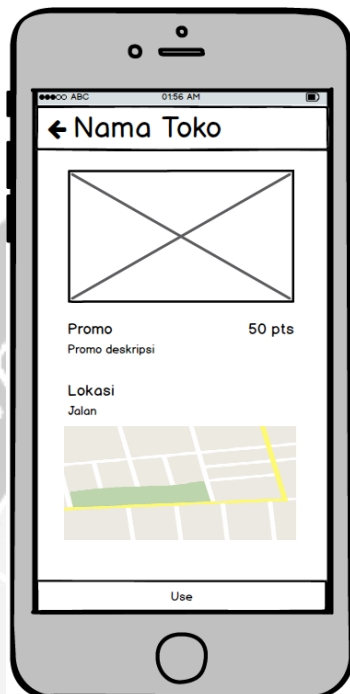
4.5. Daftar Darah PMI



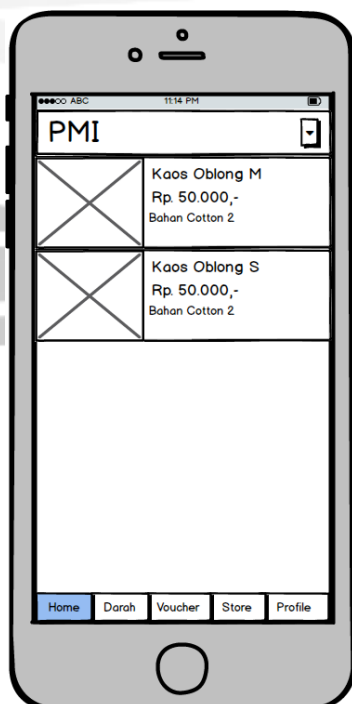
4.6. Daftar Partner PMI (Voucher)



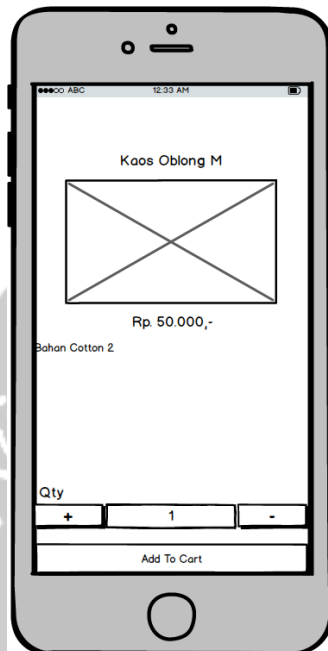
4.7. Deskripsi Partner PMI (Voucher)



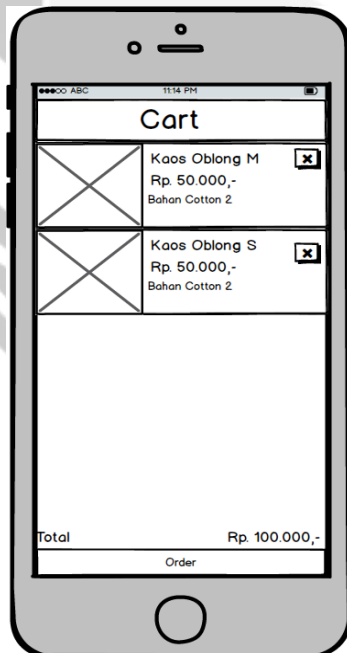
4.8. Daftar Barang Usaha Dana PMI



4.9. Deskripsi Barang Usaha Dana PMI



4.10. Daftar Keranjang Belanja Pengguna



4.11. Pesan Barang

ABC 12:40 AM

Alamat Saya
 Alamat Lain

Alamat
Provinsi
Kota

Jasa Pengiriman
JNE
TIKI
KURIR

Ongkos Kirim 15000
Total 200000

Pesan

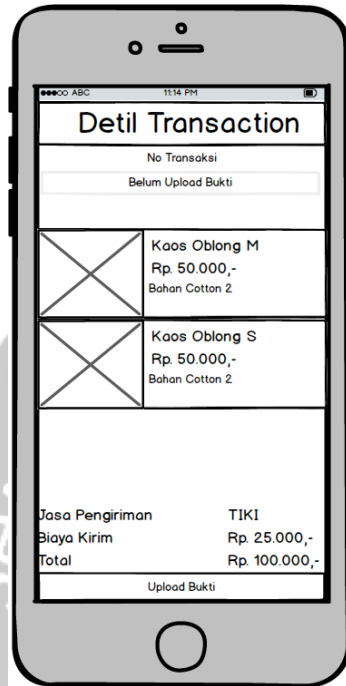
4.12. Daftar Transaksi Pengguna

ABC 02:14 AM

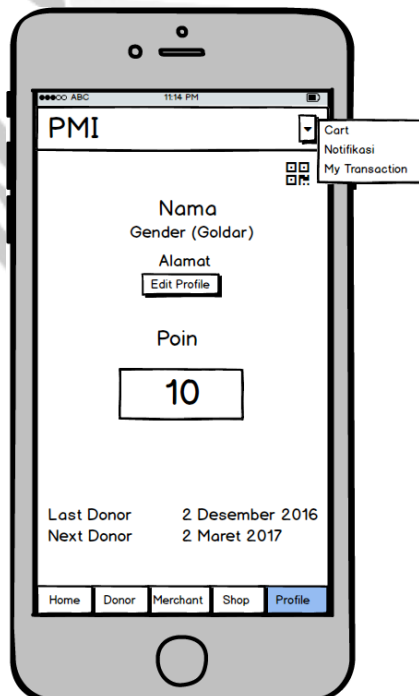
Transaksi

Di Proses	Selesai
<input checked="" type="checkbox"/>	<input type="checkbox"/>
No Transaksi	
Total	
Jumlah Barang: 2	
Status: Belum Upload	

4.13. Deskripsi Transaksi Pengguna



4.14. Data Pengguna



4.15. Ubah Data Pengguna

Ubah Data Diri

Data Diri

Nama

Email

No Telp

Tempat Lahir

Tgl Lahir

Jenis Kelamin

Golongan Darah

Alamat

Alamat

Provinsi

Kota

Simpan

4.16. Login Web

A Web Page

http://

Username

Password

Login

4.17. Kelola Akun

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Edit Profile

Username Role

Save

Nama PMI

Alamat

No Telp

Info PMI

Biaya Kurir

Gambar Upload

4.18. Tampil Daftar Acara

Manage Event

Username Role

Add

No	Event	Date	Time	Detail	Image	Action
1	Donor Ceria	23 Desember 2016	08.00-17.00			Edit Hapus
2	Donor	23 Januari 2017	10.00-15.00	Ayo Join		Edit Hapus

4.19. Tambah Data Acara

Manage Event - Tamt

Username Role

Save Cancel

Event Name

Event Date

Time

Event Location

Details

Image

Upload Image

4.20. Ubah Data Acara

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Manage Event - Edit

Username Role

Save Cancel

Event Name: Donor Ceria

Event Date: 23 /12 /2016

Time: 08.00 - 17.00

Event Location: Jl. Jendral Sudirman no 11

Details: Ayo Join

Image: Upload Image

4.21. Hapus Data Acara

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Manage Event

Username Role

Add

No	Event	Date	Time	Detail	Image	Action
1	Donor Ceria					Edit Hapus
2	Donor					Edit Hapus

Hapus Data Acara

Apakah anda yakin akan menghapus data?

Ya Tidak

4.22. Tampil Daftar Kategori Darah

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
 - Blood Category
 - Blood
- Manage Merchant
- Manage Store
 - Store
 - Transaction

Manage Darah - Kategori Darah

Username Role

Add

No ^	Kategori	Action
1	Plasma	Edit Hapus

4.23. Tambah Data Kategori Darah

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
 - Blood Category
 - Blood
- Manage Merchant
- Manage Store
 - Store
 - Transaction

Manage Darah - Kategori Darah

Username Role

Add

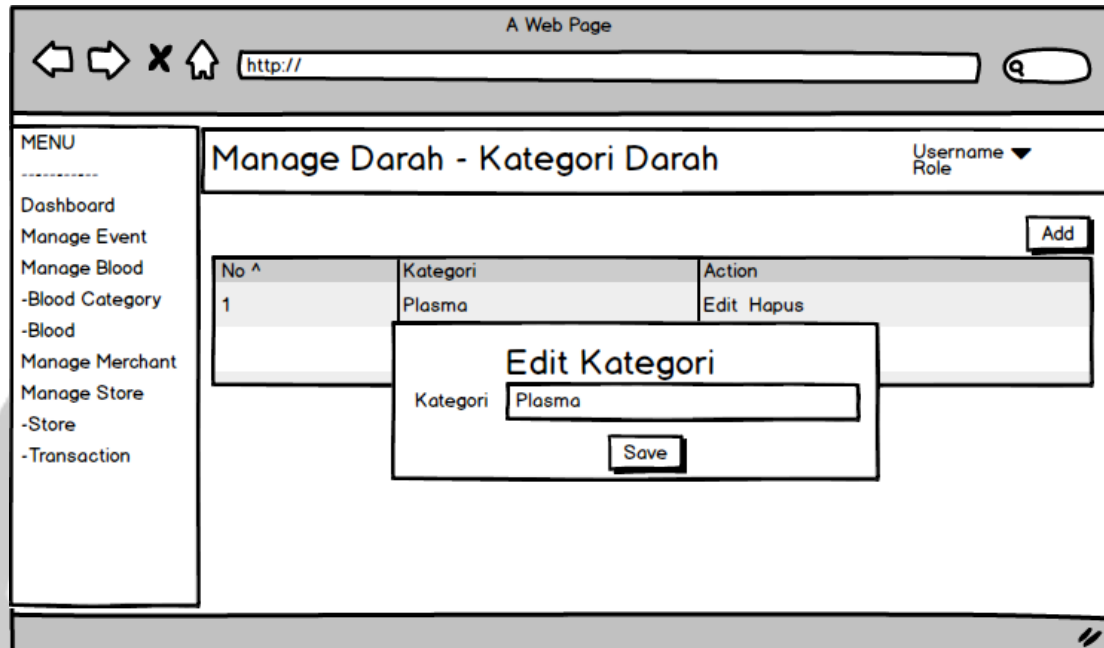
No ^	Kategori	Action
1	Plasma	Edit Hapus

Tambah Kategori

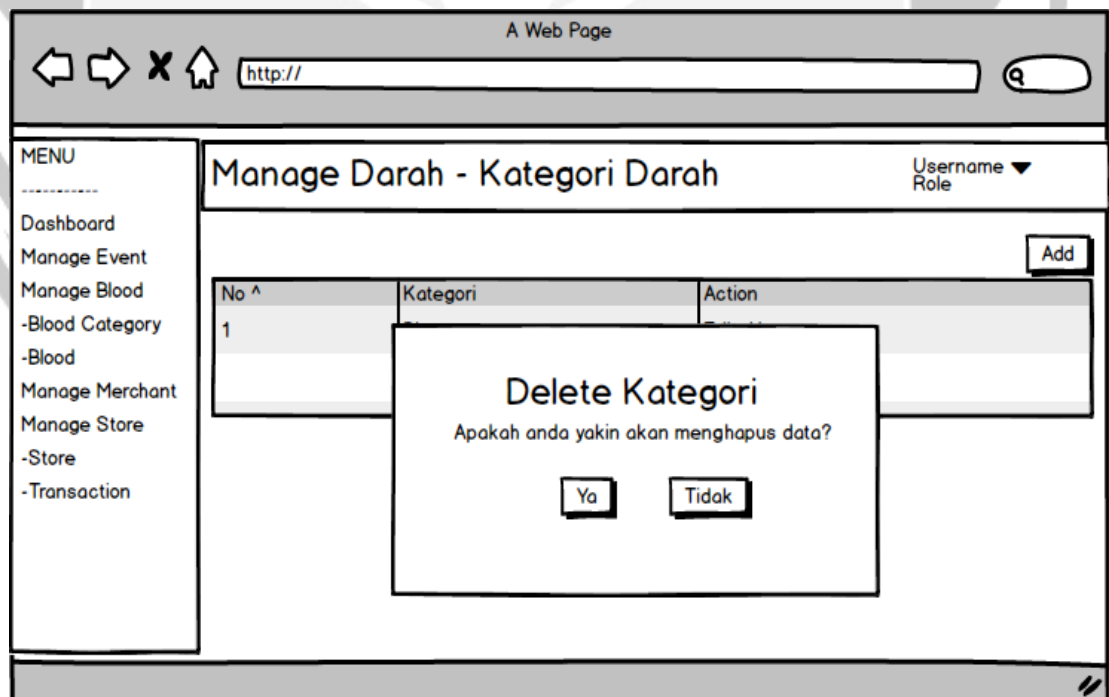
Kategori

Save

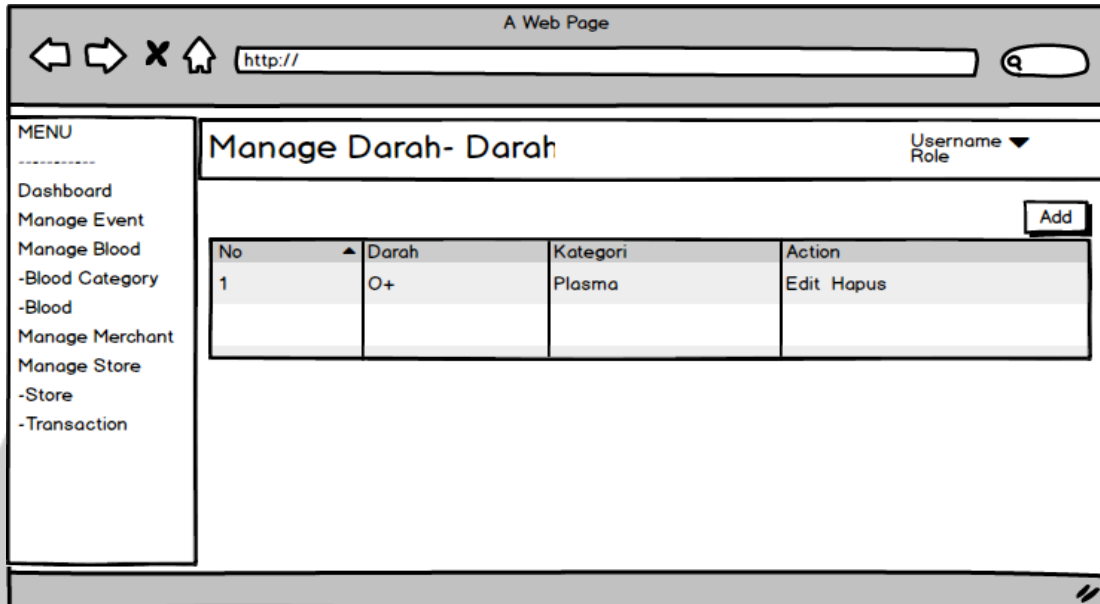
4.24. Ubah Data Kategori Darah



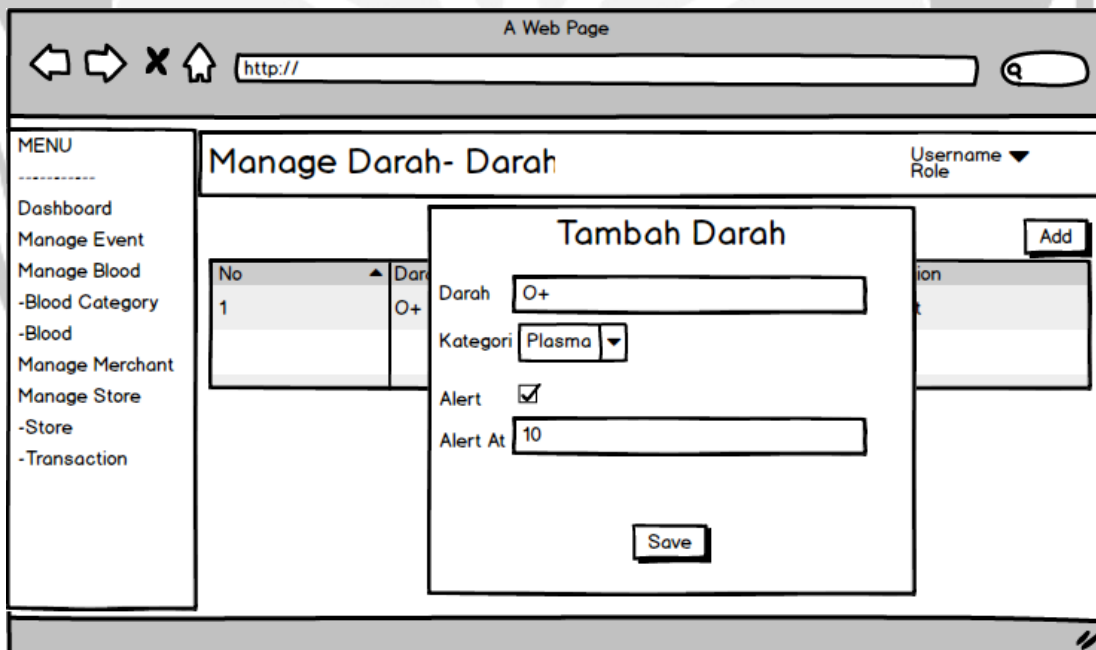
4.25. Hapus Data Kategori Darah



4.26. Tampil Daftar Darah



4.27. Tambah Data Darah



4.28. Ubah Data Darah

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Manage Darah- Darah

Username Role

Ubah Darah

Add

No	Darah
1	O+

Darah:

Kategori:

Alert:

Alert At:

Save

4.29. Hapus Data Darah

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Manage Darah- Darah

Username Role

Add

No	Darah	Kategori	Action
1	O+	Plasma	us

Hapus Data Darah

Apakah anda yakin akan menghapus data?

Ya Tidak

4.30. Tampil Daftar Partner

A Web Page

http://

MENU

Dashboard

Manage Event

Manage Blood

-Blood Category

-Blood

Manage Merchant

Manage Store

-Store

-Transaction

Manage Merchant

Username Role

Add

No	Merchant Name	Address	Promo	Image	Voucher	Aksi
1	Chatime	Jl. Babarsari no. xx	20% off		123EV429	Edit Hapus
2	Megurame Udon	Ambarukmo Plaza It.3	10 % off		EA42d134	Edit Hapus

4.31. Tambah Data Partner

A Web Page

http://

MENU

Dashboard

Manage Event

Manage Blood

-Blood Category

-Blood

Manage Merchant

Manage Store

-Store

-Transaction

Manage Merchant - Add

Username Role

Save Cancel

Merchant Name

Address

Promo

Voucher

Image

123EV429

Upload Image

4.32. Ubah Data Partner

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Manage Merchant - Edit

Username Role

Save Cancel

Merchant Name

Address

Promo

Voucher

Image

123EV429

Upload Image

4.33. Hapus Data Partner

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Manage Merchant

Username Role

Add

No	Merchant Name	Address	Promo	Voucher	Aksi
1	Chatime			EV429	Edit Hapus
2	Megurame Ud			2dl34	Edit Hapus

Hapus Data Merch

Apakah anda yakin akan menghapus data?

Ya Tidak

4.34. Tampil Daftar Barang

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Manage Store - Store

Username ▼
Role

Add

No	Stuff	Detail	Berat	Stok	Poin	Image	Price	Action
1	Clothes	Bahan Terbaik	15gr	10	10		Rp.50.000	Edit Hapus

4.35. Tambah Data Barang

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Manage Store - Add Stuff

Username ▼
Role

Save Cancel

Stuff Name

Price

Stok

Berat

Poin

Details

Image

4.36. Ubah Data Barang

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Manage Store - Edit Stuff

Username Role

Save Cancel

Stuff Name

Price

Stok

Berat

Poin

Details

Image Upload Image

4.37. Hapus Data Barang

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
- Blood Category
- Blood
- Manage Merchant
- Manage Store
- Store
- Transaction

Manage Store - Store

Username Role

Add

No	Stuff	Detail	Berat	Stok	Poin	Image	Price	Action
1	Clothes	B...	000	Edit Hapus

Hapus Data Baran!

Apakah anda yakin akan menghapus data?

Ya Tidak

4.38. Tampil Daftar Transaksi

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
 - Blood Category
 - Blood
- Manage Merchant
- Manage Store
 - Store
 - Transaction

Manage Store - Transaction Username Role

No	Name	Address	Telephone	Total	Status	
1	AAA	Jl. Babarsari no. xx	08731xxxx	123EV429	Paid	see detail ubah status
2	BBB	Ambarukmo Plaza It.3	0815xxxx	EA42dl34	UnPaid	see detail ubah status

4.39. Tampil Detil Transaksi

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
 - Blood Category
 - Blood
- Manage Merchant
- Manage Store
 - Store
 - Transaction

Transaction Detail - AAA Username Role

No	Stuff	Qty	Notes
1	Clothes	2	Size M & L
2	HandBand	1	

4.40. Ubah Status Transaksi

A Web Page

http://

MENU

- Dashboard
- Manage Event
- Manage Blood
 - Blood Category
 - Blood
- Manage Merchant
- Manage Store
 - Store
 - Transaction

Manage Store - Transaction

Username ▼
Role

No ▲	Name	Addr
1	AAA	Jl. B
2	BBB	Amb

Ubah Status

Status ▼

No resi

Detail ubah status

Detail ubah status