

BAB VI

PENUTUP

Pada bab penutup ini akan dijabarkan mengenai kesimpulan dan saran dari pembangunan sistem, daftar pustaka yang digunakan penulis dalam menyusun laporan ini, dan lampiran yang berisi SKPL (Spesifikasi Kebutuhan Perangkat Lunak) dan DPPL (Dokumen Perancangan Perangkat Lunak) dari perancangan pembangunan sistem.

6.1 Kesimpulan

Berdasarkan analisis, desain, dan implementasi perangkat lunak dan pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan dari Tugas Akhir ini yaitu Aplikasi *mobile marketplace* yang dapat membentuk seperti mall virtual telah berhasil dibangun dengan menggunakan *framework* Android dan CodeIgniter.

6.2 Saran

Saran yang dapat diambil dari proses analisis sampai pada pembuatan Tanlomall di tugas akhir ini adalah sebagai berikut:

1. Menambahkan fungsionalitas pada aplikasi *mobile* untuk laporan-laporan pendapatan dan keuangan yang lebih baik lagi.
2. Menambahkan fungsionalitas pada aplikasi seperti saldo, sehingga pengguna mengambil dana hasil penjualannya dengan nominal tertentu.
3. Menambahkan fungsionalitas pada aplikasi *web* sebagai *front-end* untuk *website*-nya agar pelanggan juga dapat mengaksesnya melalui *web browser* di laptop atau komputer.

DAFTAR PUSTAKA

- Anindito, S. D. A., 2012. *Pembangunan Aplikasi Android untuk Layanan Pesan Antar Barang atau Jasa Berbasis Lokasi dan Teknologi Push Notification*, Yogyakarta: Universitas Atma Jaya Yogyakarta.
- Arvi, A., Satoto, K. I. & Kridalukmana, R., 2013. Perancangan Aplikasi Toko Online XO-Licious Berbasis Mobile Web pada Sistem Operasi Android. *Jurnal Teknologi dan Sistem Komputer*, I(4), pp. 1-11.
- Fatmanto, A., 2013. *Pembangunan Aplikasi Forum Jual Beli Berbasis Mobile*, Yogyakarta: Universitas Atma Jaya Yogyakarta.
- Gantini, T. & Kurniawan, D., 2008. Website E-Commerce Toko Baros Elektronik Cimahi. *Jurnal Sistem Informasi*, III(2), pp. 101 - 111.
- Hartmann, B., 2001. *Strategic Design of B2B e-Marketplace Business Models*, Hamburg: Diplomica Verlag GmbH.
- Hasanuddin, Aryanto & Siregar, D. O., 2012. Perancangan Toko Online Produk Lokal Khas Nusantara dengan Metode Modern Retail. *Jurnal Fasilkom*, I(2), pp. 1-5.
- Laudon, K. C. & Laudon, J. P., 2006. *Management Information Systems*. 9th Editon Pearson penyunt. New Jersey: Prentice Hall.
- Lee, H. L. & Whang, S., 2001. *E-Business and Supply Chain Integration*. California, SGSCMF-W2-2001, pp. 1-29.
- Mariana, N., Priyambodo, A. & Arianto, T., 2011. E-Commerce Sentra Kluster IKM Sepatu di Kota Semarang sebagai Media Promosi dan Komunitas Online. *Jurnal Teknologi Informasi DINAMIK*, XVI(1), pp. 50-55.

- Marthasari, G. I., Aminudin & Munarko, Y., 2010. *Implementasi Web Service untuk Mendukung Interoperabilitas pada Aplikasi E-Commerce*. Surabaya, Electronic Engineering Polytechnic Institute of Surabaya, pp. 117 - 121.
- Meida, A. & Jauhari, J., 2011. Aplikasi M-Commerce untuk Penjualan Buku pada Suatu Toko Buku berbasis WAP [Wireless Application Protocol] Service dengan Media Handphone. *Jurnal Sistem Informasi*, III(2), pp. 14-39.
- Mitchell, L. J., 2013. *PHP Web Services*. 2nd penyunt. California: O'Reilly.
- Nuryana, I. K. D. & Putra, S. M., 2012. Rancang Bangun Aplikasi M-Commerce Berbasis Android sebagai Media Informasi dan Pemesanan Online pada Distro [Pakaian]. *Jurnal Manajemen Informatika*, I(1), pp. 25-32.
- Pratama, A. B., 2005. Implementasi E-Government dalam Penyelenggaraan Pemerintahan di Era Globalisasi. *Dialogue JIAKP*, II(3), pp. 939-960.
- Putra, P. K. & Santoso, P., 2013. Aplikasi Toko Tas Online Berbasis Android. *Jurnal Dimensi Teknik Elektro*, I(1), pp. 12-17.
- Ratnasingam, P., 2007. A Risk-Control Framework for e-Marketplace Participation: The Findings of Seven Cases. *Information Management & Computer Security*, XV(2), pp. 149 - 166.
- Risti, O. & Riasti, B. K., 2013. *Pembuatan E-Commerce pada Toko Velg Mobil Limited Edition Berbasis Opencart*. Semarang, Seminar Riset Unggulan Nasional Informatika dan Komputer, p. 33036.

- Sahney, S., 2008. Critical Success Factors in Online Retail - An Application of Quality Function Deployment and Interpretive Structural Modeling. *International Journal of Business and Information*, III(1), pp. 144 - 163.
- Sariana, K., 2010. *Perancangan Sistem Informasi Berbasis Web pada Kantor BPN (Badan Pertanahan Nasional) Sleman*, Yogyakarta: Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM.
- Suhariono, A. S., Andjarwirawan, J. & Noertjahyana, A., 2013. Pembuatan Webportal untuk Jual dan Beli Customer to Customer. *Jurnal Infra*, I(2), pp. 243 - 248.
- Supriyadi, Hudiono, R. & Sinatra, L., 2013. Rancang Bangun Sistem Jejaring Klaster Berbasis web dengan Pendekatan Model E-Commerce: Marketplace Concentrator. *Jurnal Teknologi Informasi*, X(1), pp. 71-83.
- W3C, 2007. *Simple Object Access Protocol (SOAP)*. [Online]
Available at: <http://www.w3.org/TR/soap12>
[Diakses 30 Maret 2015].
- Wagh, K. & Thool, R., 2012. A Comparative Study of SOAP VS Rest Web Services Provisioning Techniques for Mobile Host. *Journal of Information Engineering and Applications*, II(5), pp. 12-17.
- Wibowo, A. T., 2013. Pembuatan Aplikasi E-Commerce Pusat Oleh-Oleh Khas Pacitan pada Toko Sari Rasa Pacitan. *Indonesian Journal on Networking and Security*, II(4), pp. 62-67.

- Wiradinata, T., 2013. Factors Influencing Nascent Entrepreneurs in an e-Marketplace. *International Journal of Business and Commerce*, III(4), pp. 35-61.
- Witono, T. & Hendrayana, F., 2011. Pembuatan Aplikasi Web Jual Beli dan Lelang Online. *Jurnal Sistem Informasi*, VI(1), pp. 101 - 111.
- Zhang, H. & Okoroafo, S. C., 2014. An e-Commerce Key Success Factors Framework for Chinese SME Exporters. *International Journal of Economics and Finance*, VI(1), pp. 129 - 137.
- Zwass, V., 1996. Electronic Commerce: Structures and Issues. *International Journal of Electronic Commerce*, I(1), pp. 3-23.

LAMPIRAN



DPPL

**DESKRIPSI PERANCANGAN PERANGKAT
LUNAK**

TanioMall

[Pembangunan Aplikasi *Mobile Marketplace*]

Untuk :

Tugas Akhir (Skripsi)

Dipersiapkan oleh:

Yohanes Marius Wahyu D.P. / 110706510

**Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Atma Jaya Yogyakarta**

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<i>DPPL-TanioMall</i>		1/ 82
		Revisi		

DAFTAR PERUBAHAN

REVISI	DESKRIPSI
A	
B	
C	
D	
E	
F	

INDEKSTGL	-	A	B	C	D	E	F
DITULIS OLEH	YMW						
DIPERIKSA OLEH							
DISETUJUI OLEH							

DAFTAR HALAMAN PERUBAHAN

Halaman	Revisi	Halaman	Revisi

DAFTAR ISI

1. Pendahuluan	10
1.1 Tujuan	10
1.2 Lingkup Masalah	10
1.3 Definisi, Akronim dan Singkatan.....	11
1.4 Referensi	11
2 Perancangan Sistem	12
2.1 Perancangan Asitektur	12
2.2 Perancangan Rinci	14
2.2.1 Sign In <i>Mobile App</i>	14
2.2.2 Sign Up <i>Mobile App</i>	14
2.2.3 Home <i>Mobile App</i>	15
2.2.4 Pengelolaan Akun Member <i>Mobile App</i>	15
2.2.5 Register Toko <i>Mobile App</i>	16
2.2.6 Pengelolaan Toko <i>Mobile App</i>	17
2.2.7 Pengelolaan Produk <i>Mobile App</i>	17
2.2.8 List Produk <i>Mobile App</i>	19
2.2.9 Detail Produk <i>Mobile App</i>	19
2.2.10 Transaksi Pemesanan <i>Mobile App</i>	20
2.2.11 Transaksi Pembayaran <i>Mobile App</i>	21
2.2.12 Negosiasi Harga <i>Mobile App</i>	22
2.2.13 Chatting <i>Mobile App</i>	22
2.2.14 Sign In Admin <i>Web</i>	23
2.2.15 Home Admin <i>Web</i>	23
2.2.16 Pengelolaan Akun Admin <i>Web</i>	24
2.2.17 Peringatan Toko <i>Web</i>	24
2.2.18 Peringatan Produk <i>Web</i>	25
2.2.19 Pengelolaan Kategori Produk <i>Web</i>	25
2.2.20 List Produk <i>Web</i>	27
2.2.21 Detail Produk <i>Web</i>	28
2.2.22 Pengawasan Transaksi Toko <i>Web</i>	28

2.2.23	Pengawasan Detil Transaksi Toko <i>Web</i>	29
2.2.24	Class Diagram <i>Mobile App</i>	30
2.2.25	Class Diagram <i>Web</i>	31
2.2.26	Class Diagram <i>Specific Descriptions</i>	32
3	PERANCANGAN DATA	52
3.1	Dekomposisi Data	52
3.1.1	Deskripsi Entitas Member	52
3.1.2	Deskripsi Entitas Toko	52
3.1.3	Deskripsi Entitas Kategori	53
3.1.4	Deskripsi Entitas SubKategori.....	53
3.1.5	Deskripsi Entitas Item	54
3.1.6	Deskripsi Entitas Testimonial.....	54
3.1.7	Deskripsi Entitas Message	55
3.1.8	Deskripsi Entitas Favorit	55
3.1.9	Deskripsi Entitas Nego	56
3.1.10	Deskripsi Entitas Transaksi	56
3.1.11	Deskripsi Entitas DetilTransaksi.....	57
3.2	Physical Data Model	58
4	Perancangan Antarmuka	59
4.1	Antarmuka Halaman Home	59
4.2	Antarmuka Pencarian	62
4.3	Antarmuka Kategori Barang	63
4.4	Antarmuka List Barang	64
4.5	Antarmuka Halaman Detil Barang.....	67
4.6	Antarmuka Halaman List Pesan Nego.....	69
4.7	Antarmuka Halaman Tambah Barang.....	71
4.8	Antarmuka Halaman Edit Barang.....	72
4.9	Antarmuka Halaman List Edit Barang Toko.....	73
4.10	Antarmuka Halaman Profil Member	74
4.11	Antarmuka Halaman Edit Profil Member	75
4.12	Antarmuka Halaman Keranjang Belanja	77
4.13	Antarmuka Halaman Beranda Web	78
4.14	Antarmuka Halaman Pesan	79
4.15	Antarmuka Halaman Detil Pesan	80

4.16	Antarmuka Halaman Bukti Pembayaran	81
4.17	Antarmuka Halaman Pengawasan Barang	82

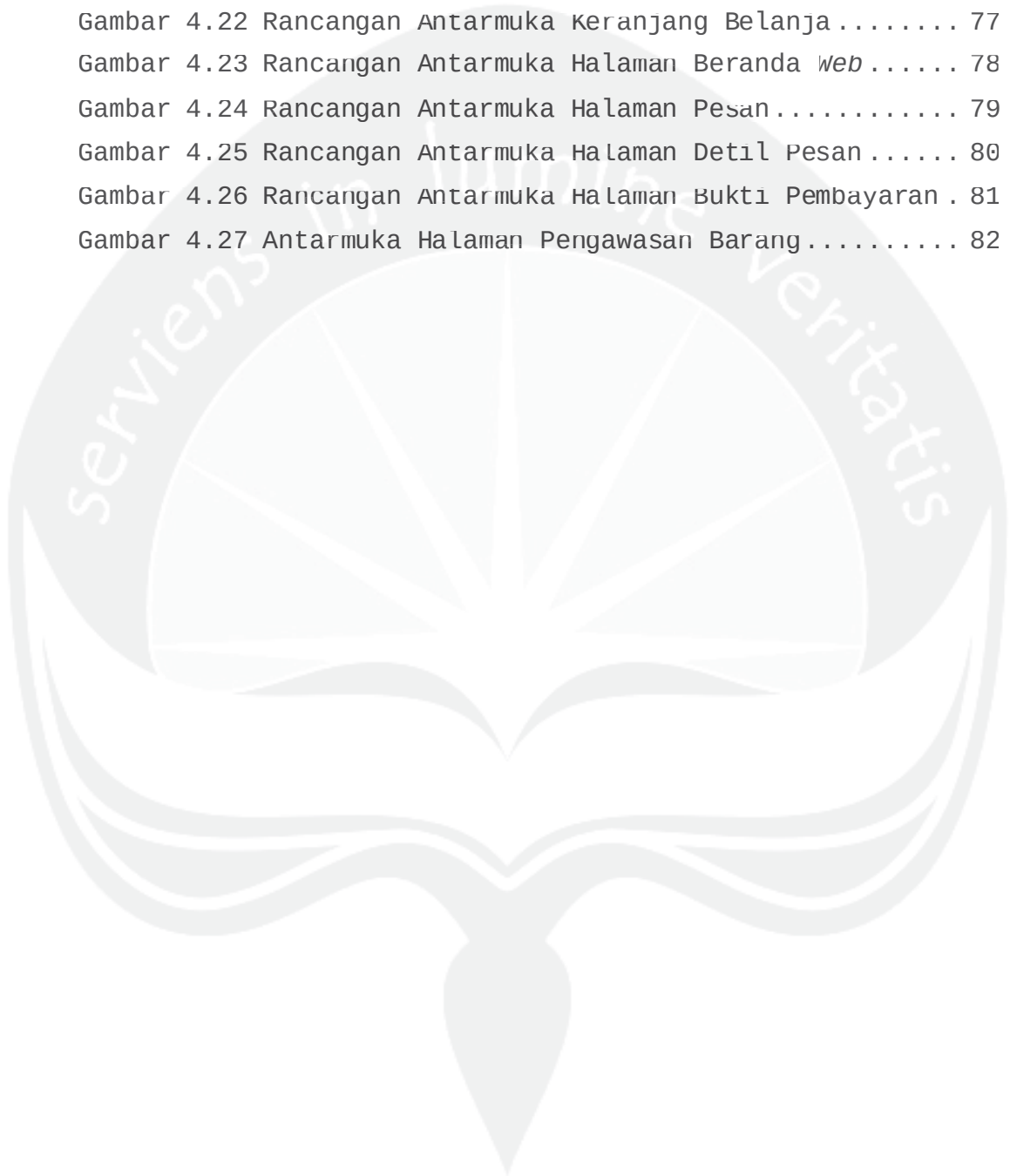


DAFTAR GAMBAR

Gambar 2.1 Perancangan Arsitektur Kelas <i>Mobile</i> TanioMall ..	12
Gambar 2.2 Perancangan Arsitektur Kelas <i>Web App</i> TanioMall ..	13
Gambar 2.3 <i>Sequence Diagram Sign In Mobile App</i>	14
Gambar 2.4 <i>Sequence Diagram Sign Up Mobile App</i>	14
Gambar 2.5 <i>Sequence Diagram Home Mobile App</i>	15
Gambar 2.6 <i>Sequence Diagram Pengelolaan Akun Member Mobile</i>	16
Gambar 2.7 <i>Sequence Diagram Register Toko Mobile App</i>	16
Gambar 2.8 <i>Sequence Diagram Pengelolaan Toko Mobile App</i> ..	17
Gambar 2.9 <i>Sequence Diagram Tambah Data Produk Mobile App</i>	17
Gambar 2.10 <i>Sequence Diagram Edit Data Produk Mobile App</i> ..	18
Gambar 2. 11 <i>Sequence Diagram Hapus Data Produk Mobile App</i>	18
Gambar 2. 12 <i>Sequence Diagram List Produk Mobile App</i>	19
Gambar 2.13 <i>Sequence Diagram Detail Produk Mobile App</i>	19
Gambar 2.14 <i>Sequence Diagram Tambah Transaksi Pemesanan Mobile App</i>	20
Gambar 2.15 <i>Sequence Diagram Batal Transaksi Pemesanan Mobile App</i>	21
Gambar 2.16 <i>Sequence Diagram Transaksi Pembayaran Mobile App</i>	21
Gambar 2.17 <i>Sequence Diagram Negosiasi Harga Mobile App</i> ..	22
Gambar 2.18 <i>Sequence Diagram Chatting Mobile App</i>	22
Gambar 2.19 <i>Sequence Diagram Sign In Web</i>	23
Gambar 2.20 <i>Sequence Diagram Home Web</i>	23
Gambar 2.21 <i>Sequence Diagram Pengelolaan Akun Admin Web</i> ..	24
Gambar 2.22 <i>Sequence Diagram Register Toko Web</i>	24
Gambar 2.23 <i>Sequence Diagram Pengelolaan Toko Web</i>	25
Gambar 2.24 <i>Sequence Diagram Tampil Data Kategori Produk Web</i>	25
Gambar 2.25 <i>Sequence Diagram Tambah Data Kategori Produk Web</i>	26
Gambar 2.26 <i>Sequence Diagram Edit Data Kategori Produk Web</i>	26

Gambar 2.27 <i>Sequence Diagram</i> Hapus Data Kategori Produk <i>Web</i>	27
Gambar 2.28 <i>Sequence Diagram</i> List Produk <i>Web</i>	27
Gambar 2.29 <i>Sequence Diagram</i> Detail Produk <i>Web</i>	28
Gambar 2. 30 <i>Sequence Diagram</i> Pengawasan Transaksi Toko <i>Web</i>	28
Gambar 2.31 <i>Sequence Diagram</i> Pengawasan Detil Transaksi <i>Web</i>	29
Gambar 2.32 <i>Class Diagram</i> TanioMall <i>Mobile App</i>	30
Gambar 2.33 <i>Class Diagram</i> TanioMall <i>Web</i>	31
Gambar 3.1 Physical Data Model TanioMall	58
Gambar 4. 1 Rancangan Antarmuka Halaman Home	59
Gambar 4. 2 Rancangan Antarmuka Halaman Menu Pengunjung ..	60
Gambar 4.3 Rancangan Antarmuka Halaman Menu <i>Member</i>	61
Gambar 4.4 Rancangan Antarmuka Halaman Pencarian	62
Gambar 4.5 Rancangan Antarmuka Halaman Kategori Barang ...	63
Gambar 4.6 Rancangan Antarmuka Halaman List Barang	64
Gambar 4.7 Rancangan Antarmuka Halaman Filter List Barang	65
Gambar 4.8 Rancangan Antarmuka Pengurutan List Barang	66
Gambar 4.9 Rancangan Antarmuka Detil Barang Nego	67
Gambar 4.10 Rancangan Antarmuka Halaman Detil Barang Tanpa Nego	67
Gambar 4.11 Rancangan Antarmuka Halaman Detil Barang Milik <i>Member</i>	68
Gambar 4.12 Rancangan Antarmuka Halaman Pesan Nego	69
Gambar 4.13 Antarmuka Halaman Detil Pesan Nego	70
Gambar 4.14 Rancangan Antarmuka Halaman Tambang Barang oleh <i>Member</i>	71
Gambar 4.15 Rancangan Antarmuka Halaman Edit Barang oleh <i>Member</i>	72
Gambar 4.16 Rancangan Antarmuka List Barang Toko	73
Gambar 4.17 Rancangan Antarmuka Profil <i>Member</i>	74
Gambar 4.18 Rancangan Antarmuka <i>Edit</i> Profil Diri <i>Member</i> ..	75
Gambar 4.19 Rancangan Antarmuka <i>Edit</i> Alamat <i>Member</i>	75

Gambar 4.20 Rancangan Antarmuka <i>Edit Akun Fasapay</i> oleh <i>Member</i>	76
Gambar 4.21 Rancangan Antarmuka <i>Edit Password Member</i>	76
Gambar 4.22 Rancangan Antarmuka keranjang Belanja	77
Gambar 4.23 Rancangan Antarmuka Halaman Beranda <i>Web</i>	78
Gambar 4.24 Rancangan Antarmuka Halaman Pesan	79
Gambar 4.25 Rancangan Antarmuka Halaman Detil Pesan	80
Gambar 4.26 Rancangan Antarmuka Halaman Bukti Pembayaran .	81
Gambar 4.27 Antarmuka Halaman Pengawasan Barang	82



1. Pendahuluan

1.1 Tujuan

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

1.2 Lingkup Masalah

Perangkat Lunak TanioMall dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan pengguna.
2. Menangani pengelolaan toko.
3. Menangani pengelolaan transaksi penjualan.
4. Menangani pengelolaan transaksi pembayaran.
5. Menangani pengelolaan kategori dan sub kategori produk oleh administrator.
6. Menangani pengelolaan produk.
7. Menangani pengelolaan negosiasi harga produk.
8. Menangani pengelolaan testimonial.
9. Menangani proses komunikasi antar pengguna dengan menggunakan fitur *chatting*.

Perangkat lunak ini berjalan pada semua lingkungan *platform* dengan *browser* yang mendukung penggunaan *javascript* (*web*) dan *smartphone* dengan *platform Android*.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan:

Keyword/Phrase	Definisi
DPPL	Merupakan deskripsi perancangan perangkat lunak yang akan dikembangkan.
TanioMall	Aplikasi <i>mobile marketplace</i> .
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World wide web.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Database	Merupakan tempat penyimpanan data.
Hapus semu	Sebenarnya tidak menghapus data, hanya membuat data tidak ditampilkan.

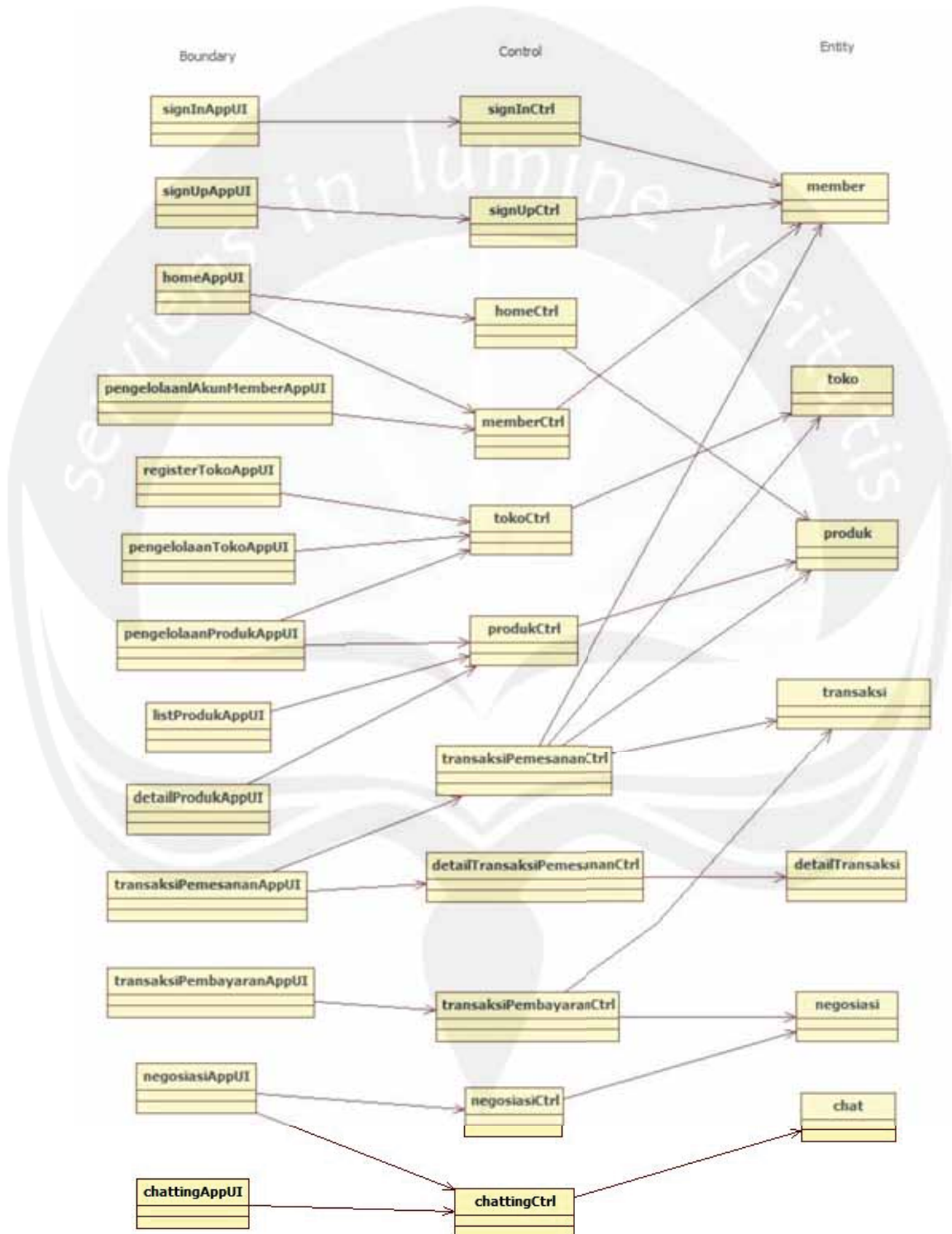
1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

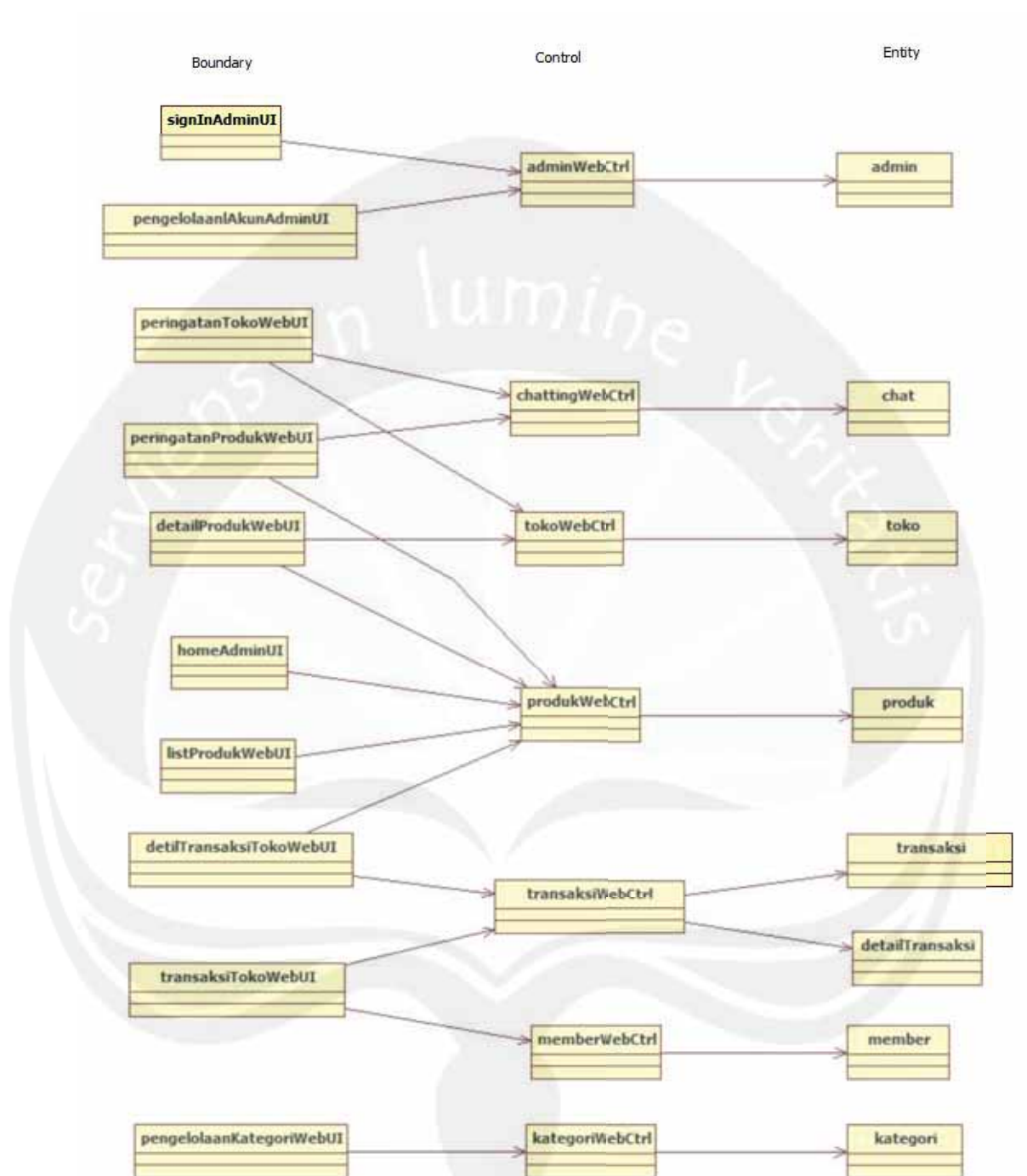
1. Stefanus Diptya Adeshya Anindito, *Pembangunan Aplikasi Android untuk Layanan Pesan Antar Barang atau Jasa Berbasis Lokasi dan Teknologi Push Notification*. 2012.
2. Septiana Rahayu Dewi, *Pembangunan Online Marketplace untuk Pengusaha Mikro di PASTY*. 2014.

2 Perancangan Sistem

2.1 Perancangan Asitektur



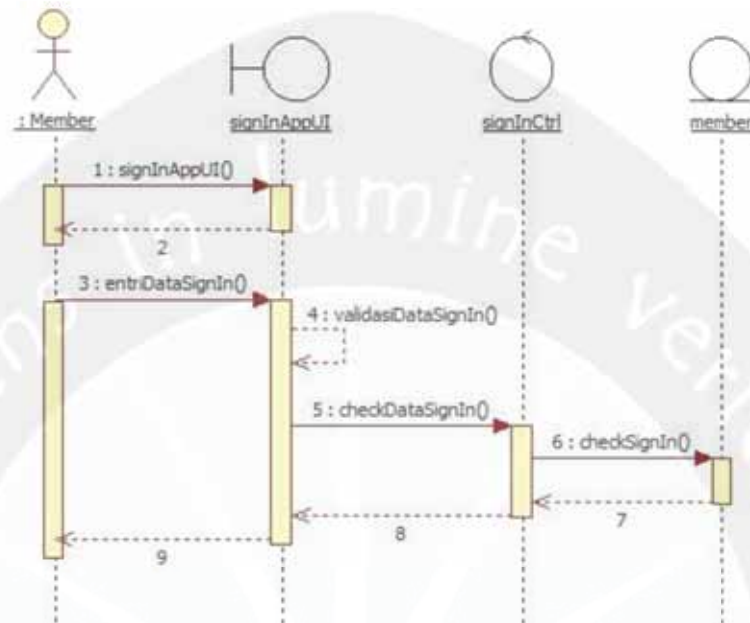
Gambar 2.1 Perancangan Arsitektur Kelas *Mobile App* TanioMall



Gambar 2.2 Perancangan Arsitektur Kelas *Web App* TanioMall

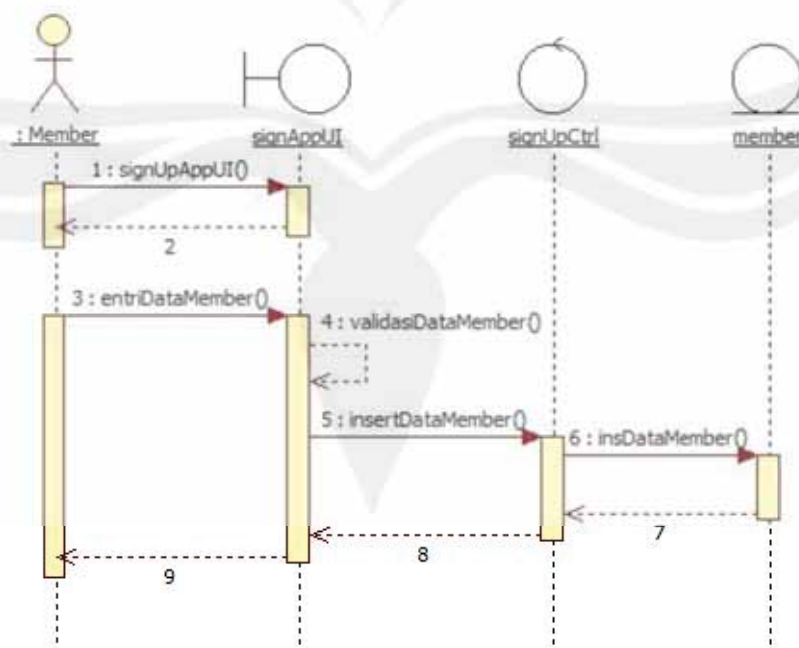
2.2 Perancangan Rinci

2.2.1 Sign In Mobile App



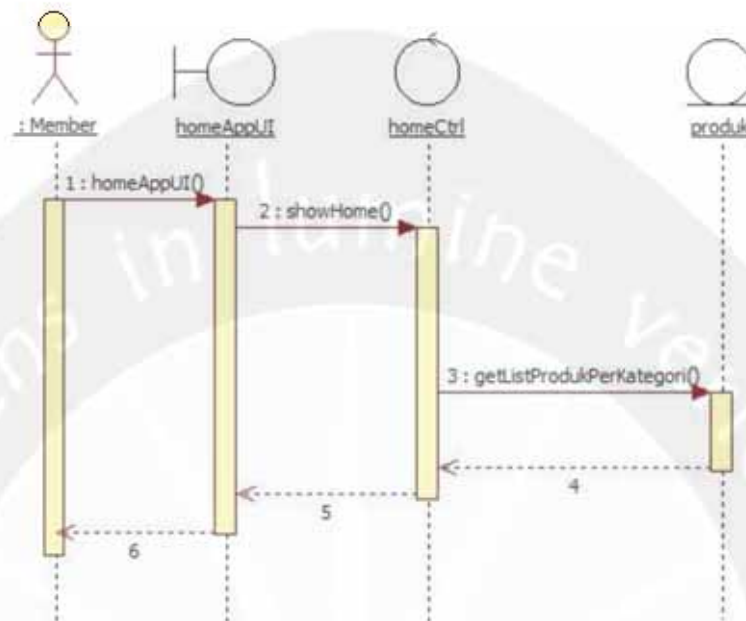
Gambar 2.3 Sequence Diagram Sign In Mobile App

2.2.2 Sign Up Mobile App



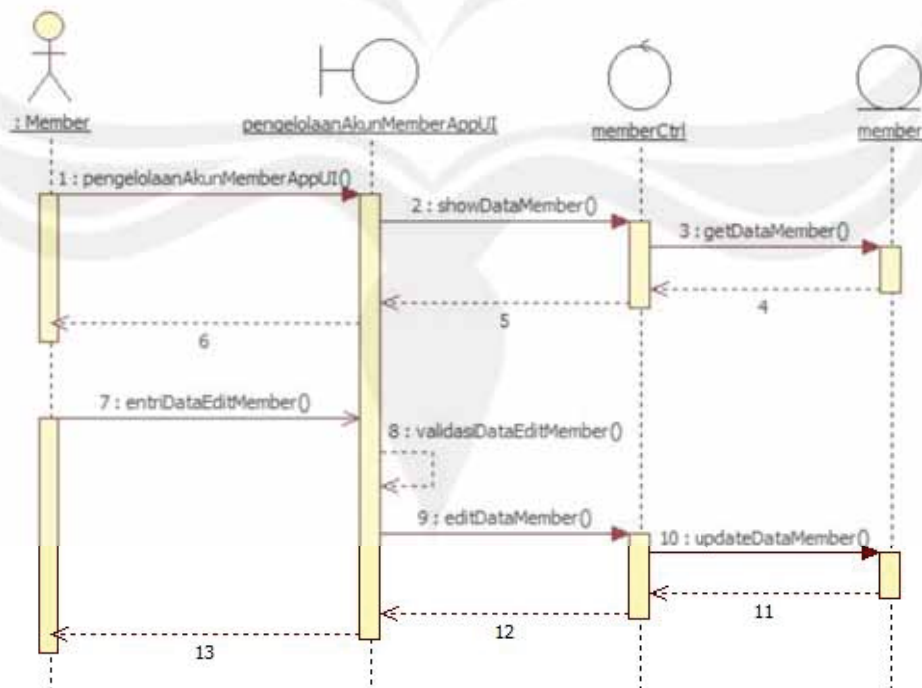
Gambar 2.4 Sequence Diagram Sign Up Mobile App

2.2.3 Home Mobile App



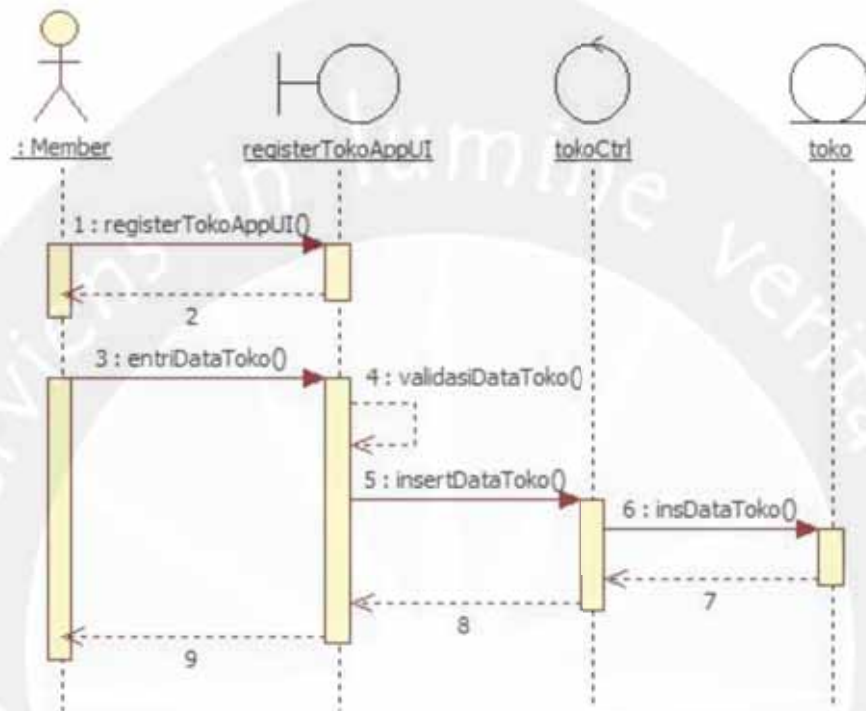
Gambar 2.5 Sequence Diagram Home Mobile App

2.2.4 Pengelolaan Akun Member Mobile App



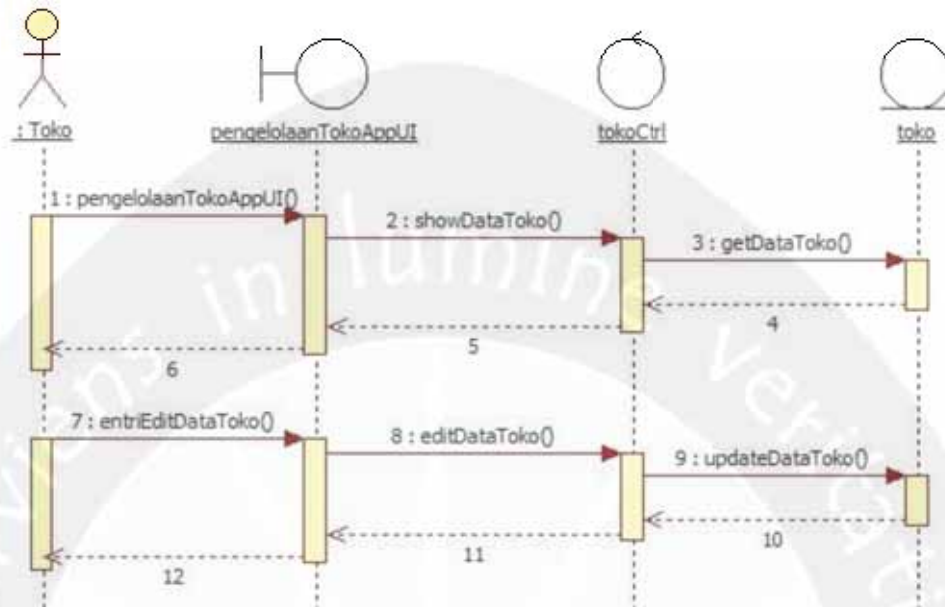
Gambar 2.6 Sequence Diagram Pengelolaan Akun Member Mobile App

2.2.5 Register Toko Mobile App



Gambar 2.7 Sequence Diagram Register Toko Mobile App

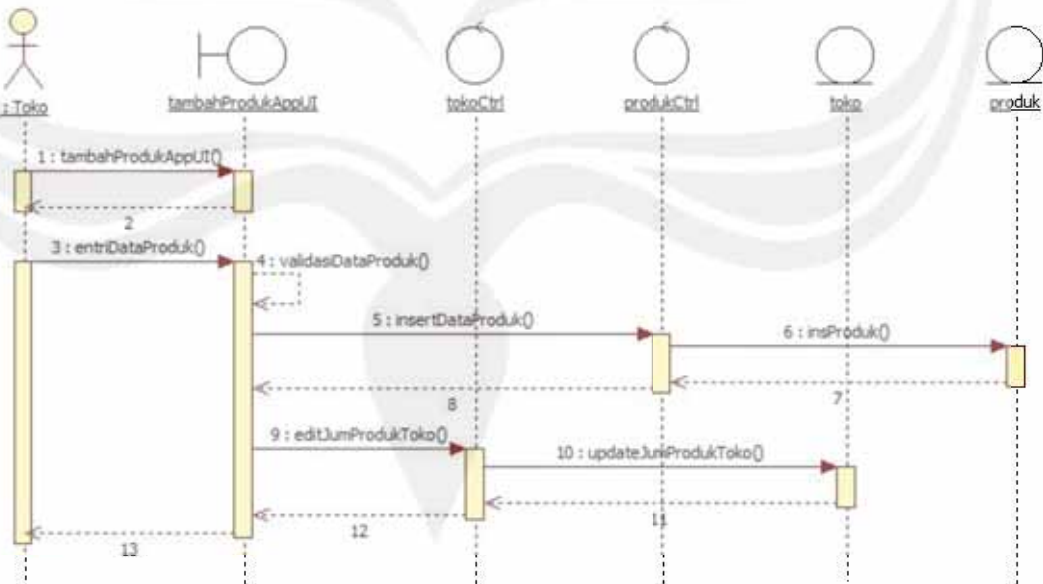
2.2.6 Pengelolaan Toko *Mobile App*



Gambar 2.8 Sequence Diagram Pengelolaan Toko *Mobile App*

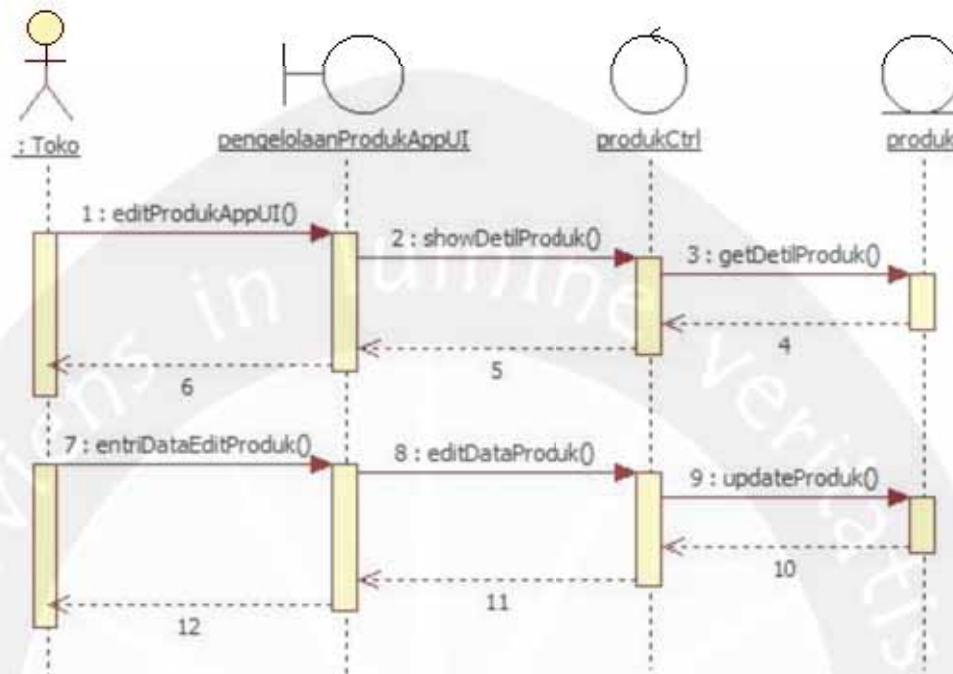
2.2.7 Pengelolaan Produk *Mobile App*

2.2.7.1 Tambah Data Produk



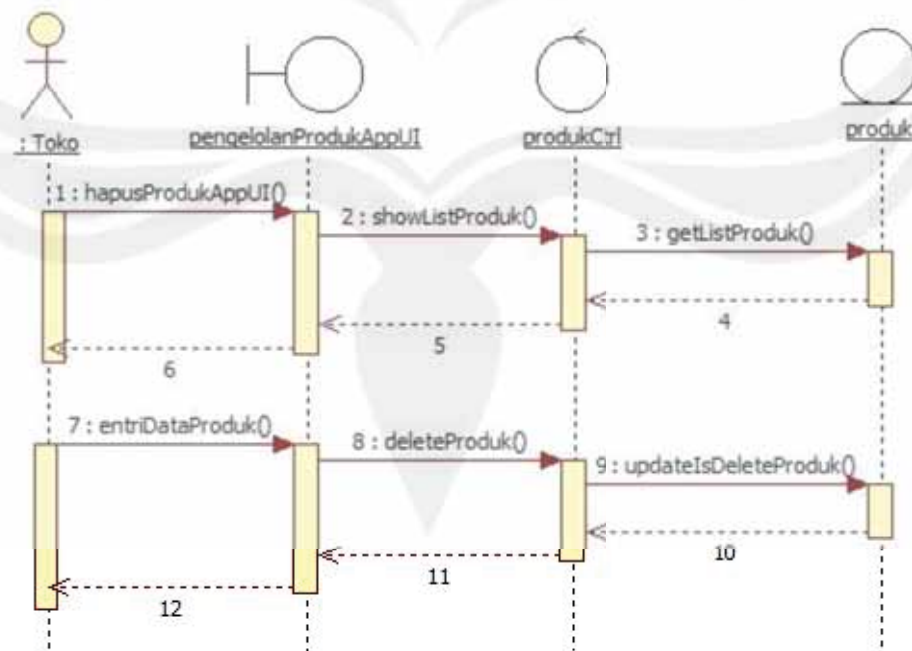
Gambar 2.9 Sequence Diagram Tambah Data Produk *Mobile App*

2.2.7.2 Edit Data Produk



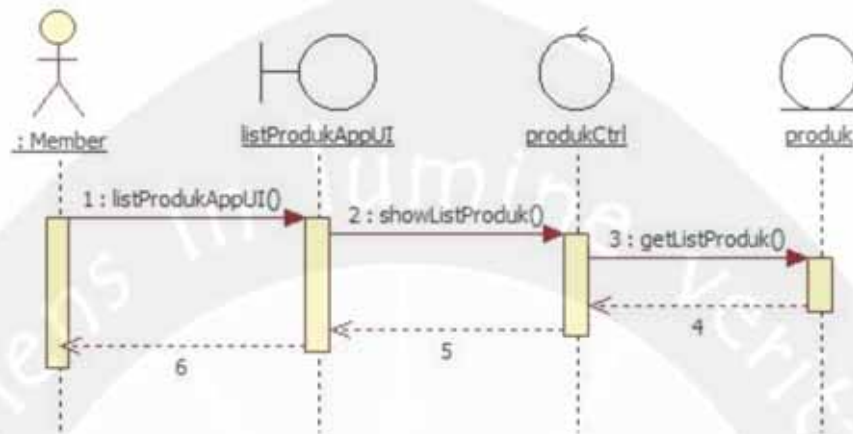
Gambar 2.10 Sequence Diagram Edit Data Produk Mobile App

2.2.7.3 Hapus Data Produk



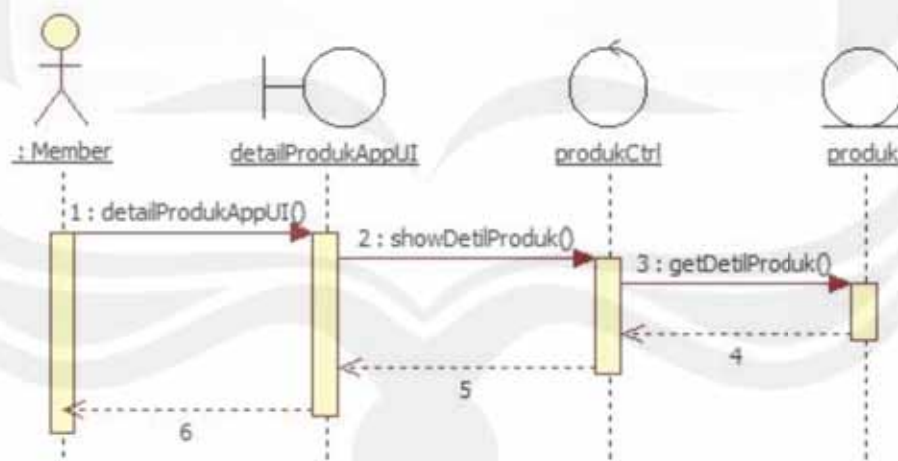
Gambar 2. 11 Sequence Diagram Hapus Data Produk Mobile App

2.2.8 List Produk *Mobile App*



Gambar 2. 12 Sequence Diagram List Produk *Mobile App*

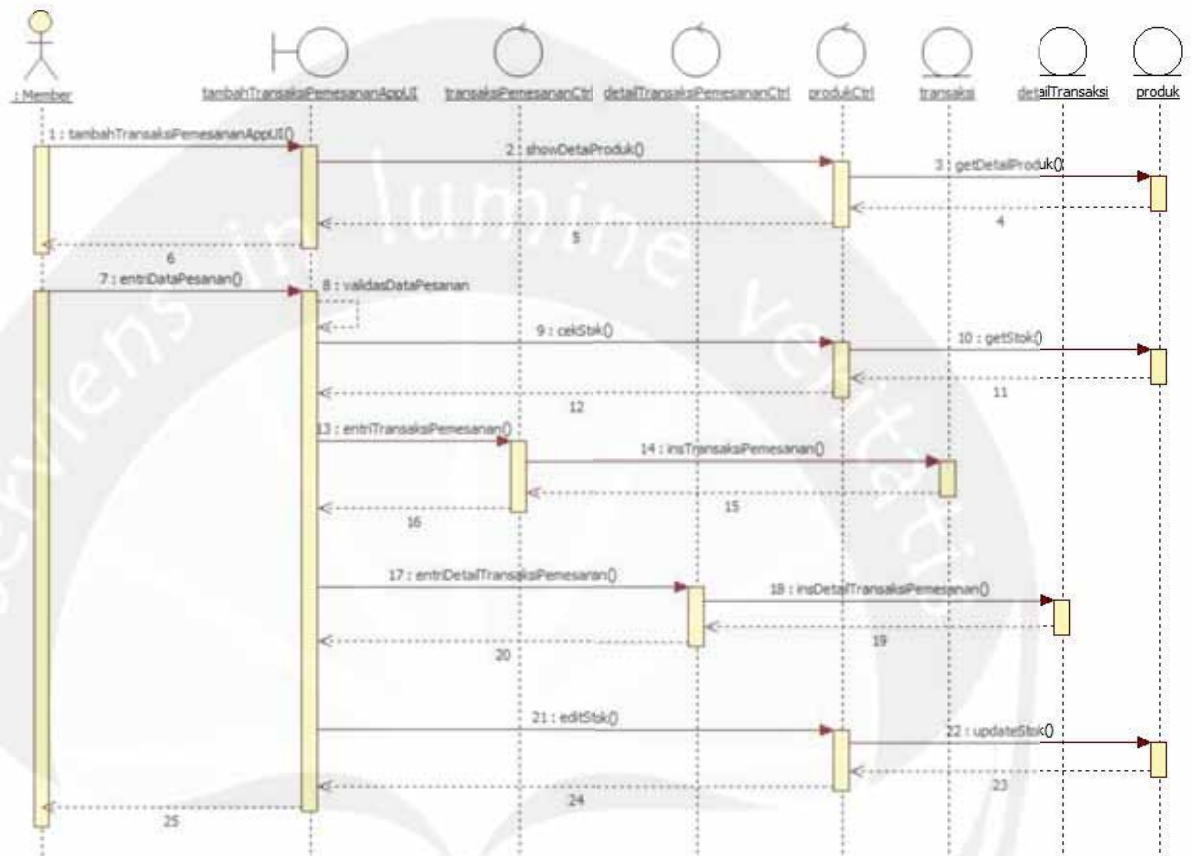
2.2.9 Detail Produk *Mobile App*



Gambar 2.13 Sequence Diagram Detail Produk *Mobile App*

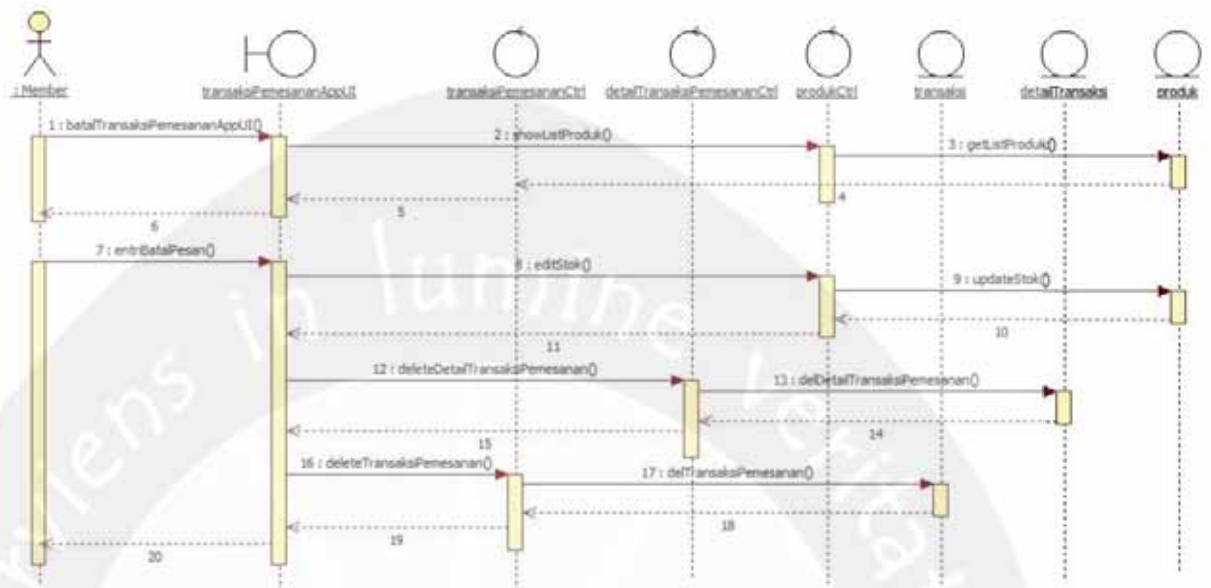
2.2.10 Transaksi Pemesanan *Mobile App*

2.2.10.1 Tambah Transaksi Pemesanan



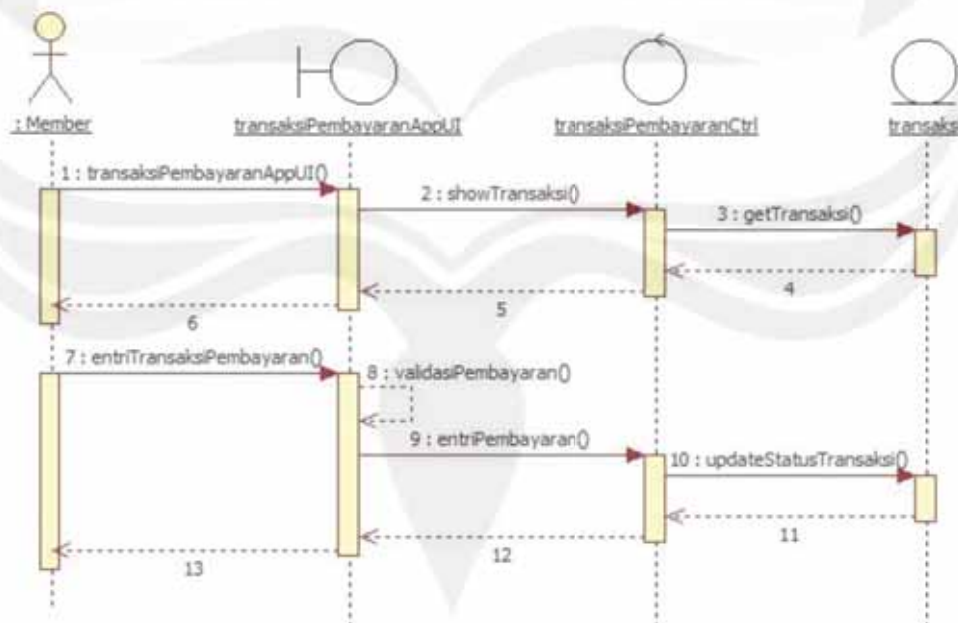
Gambar 2.14 Sequence Diagram Tambah Transaksi Pemesanan
Mobile App

2.2.10.2 Batal Transaksi Pemesanan



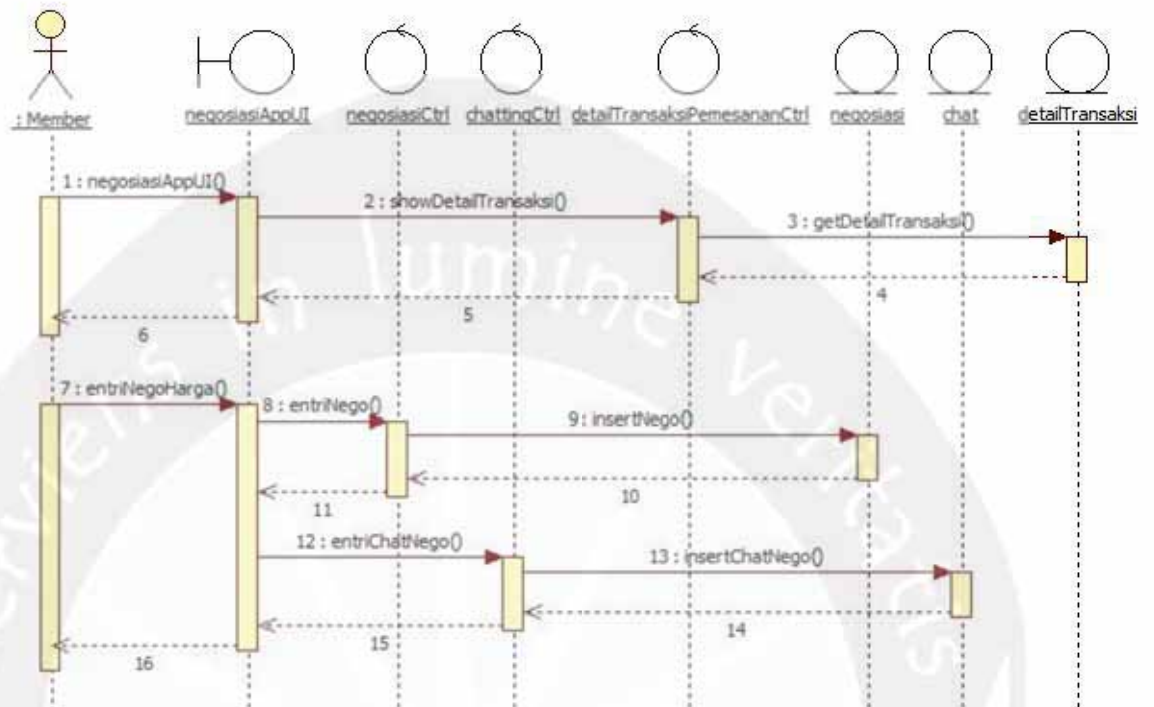
Gambar 2.15 Sequence Diagram Batal Transaksi Pemesanan Mobile App

2.2.11 Transaksi Pembayaran Mobile App



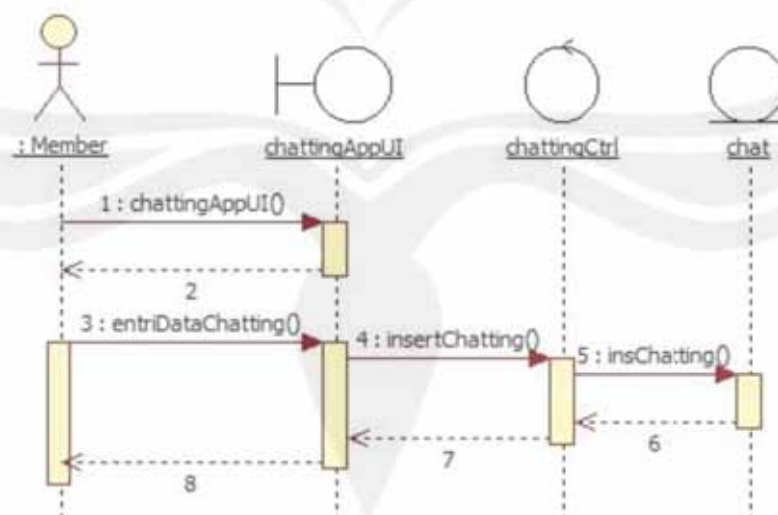
Gambar 2.16 Sequence Diagram Transaksi Pembayaran Mobile App

2.2.12 Negosiasi Harga Mobile App



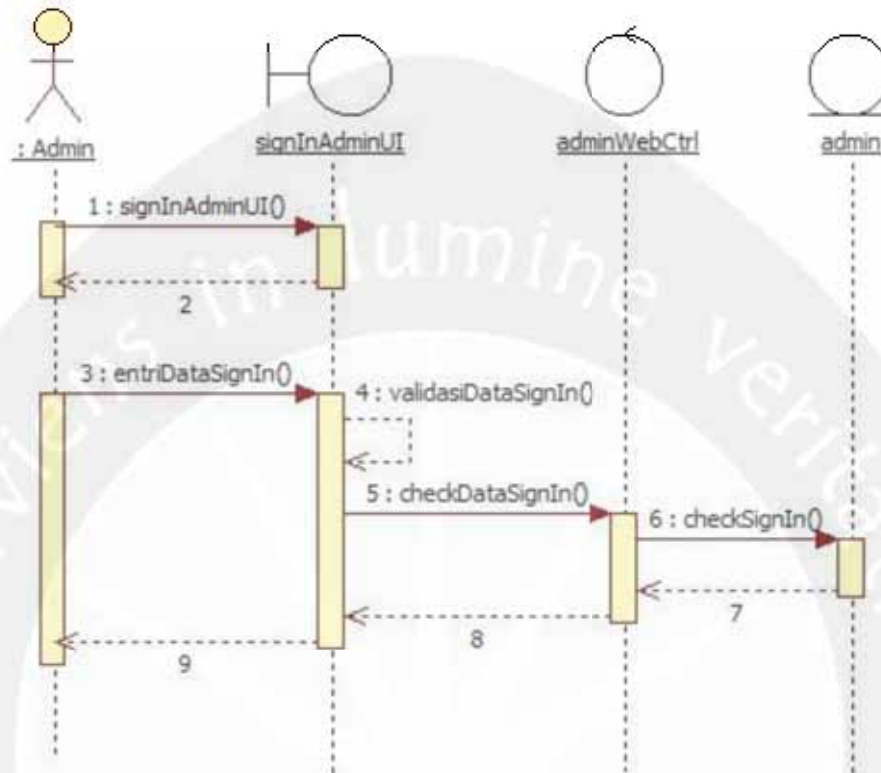
Gambar 2.17 Sequence Diagram Negosiasi Harga Mobile App

2.2.13 Chatting Mobile App



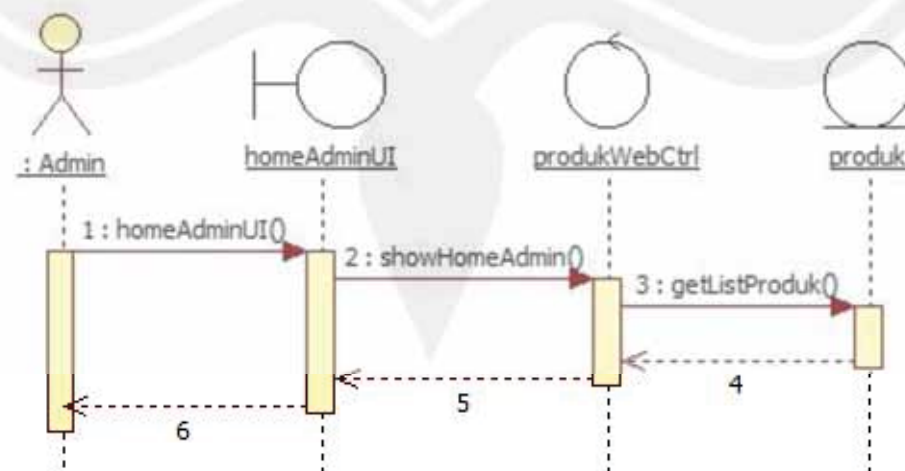
Gambar 2.18 Sequence Diagram Chatting Mobile App

2.2.14 Sign In Admin Web



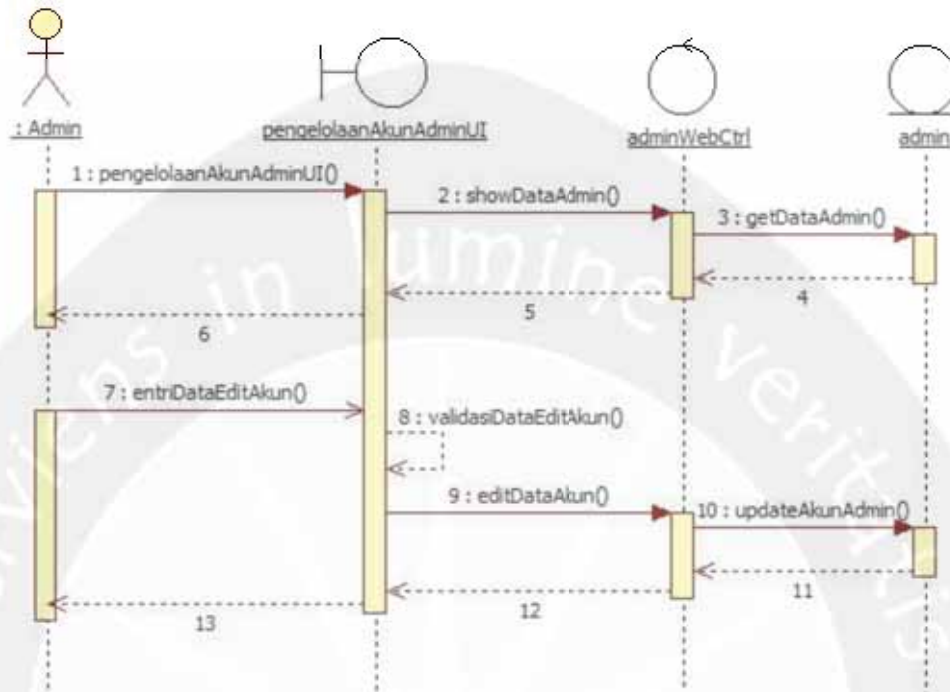
Gambar 2.19 Sequence Diagram Sign In Web

2.2.15 Home Admin Web



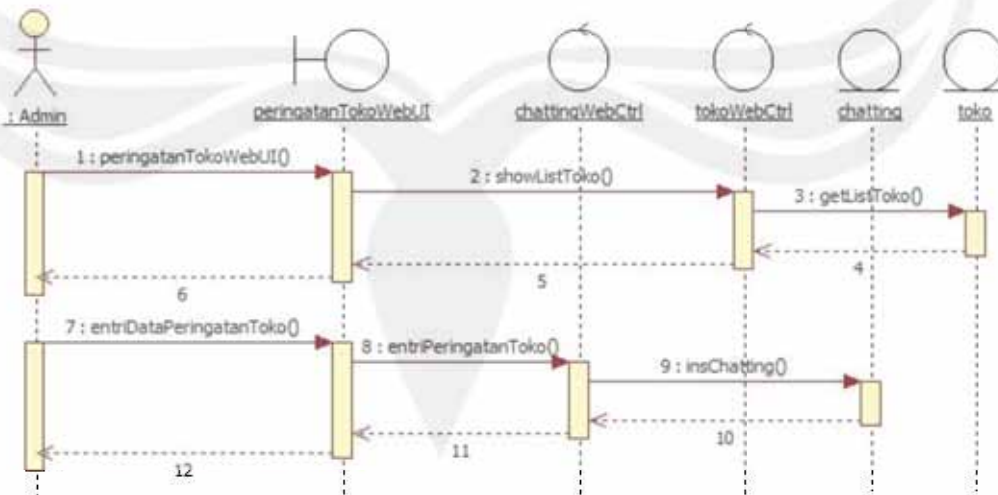
Gambar 2.20 Sequence Diagram Home Web

2.2.16 Pengelolaan Akun Admin Web



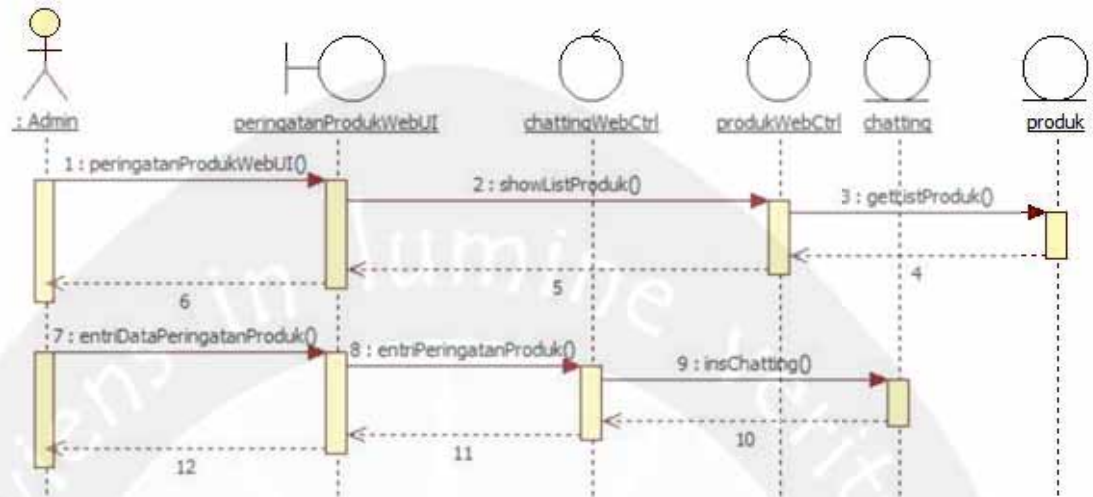
Gambar 2.21 Sequence Diagram Pengelolaan Akun Admin web

2.2.17 Peringatan Toko Web



Gambar 2.22 Sequence Diagram Register Toko Web

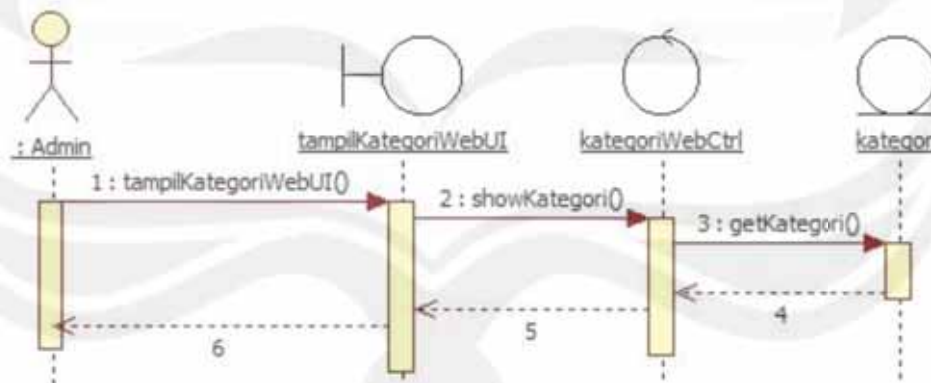
2.2.18 Peringatan Produk Web



Gambar 2.23 Sequence Diagram Pengelolaan Toko Web

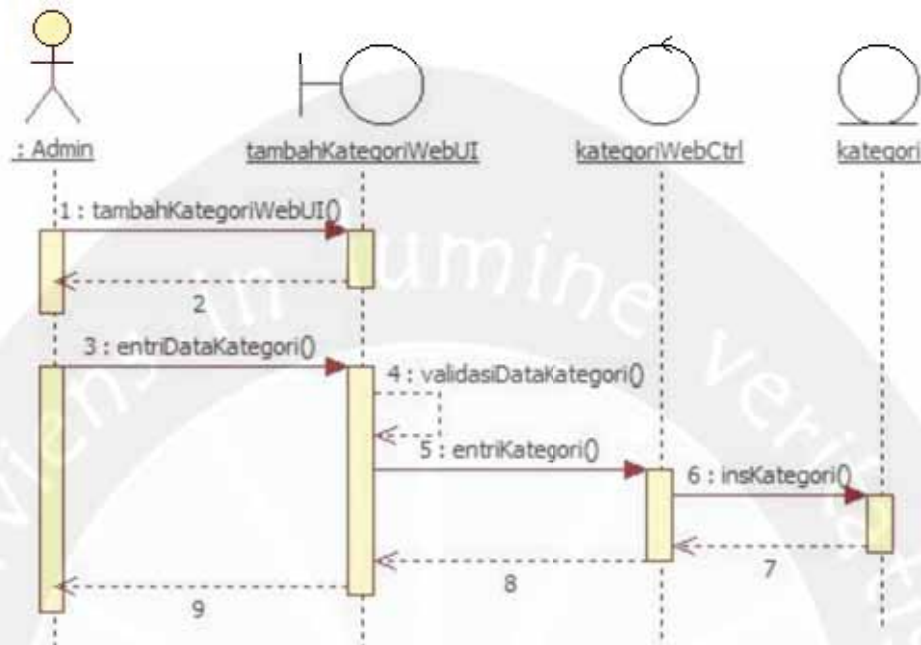
2.2.19 Pengelolaan Kategori Produk Web

2.2.19.1 Tampil Data Kategori Produk



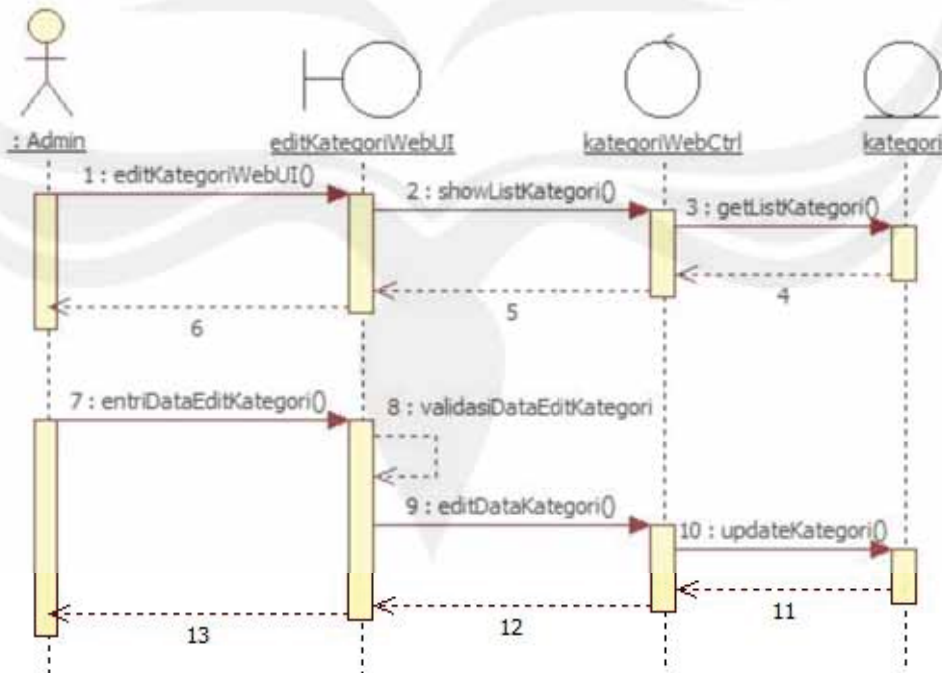
Gambar 2.24 Sequence Diagram Tampil Data Kategori Produk Web

2.2.19.2 Tambah Data Kategori Produk



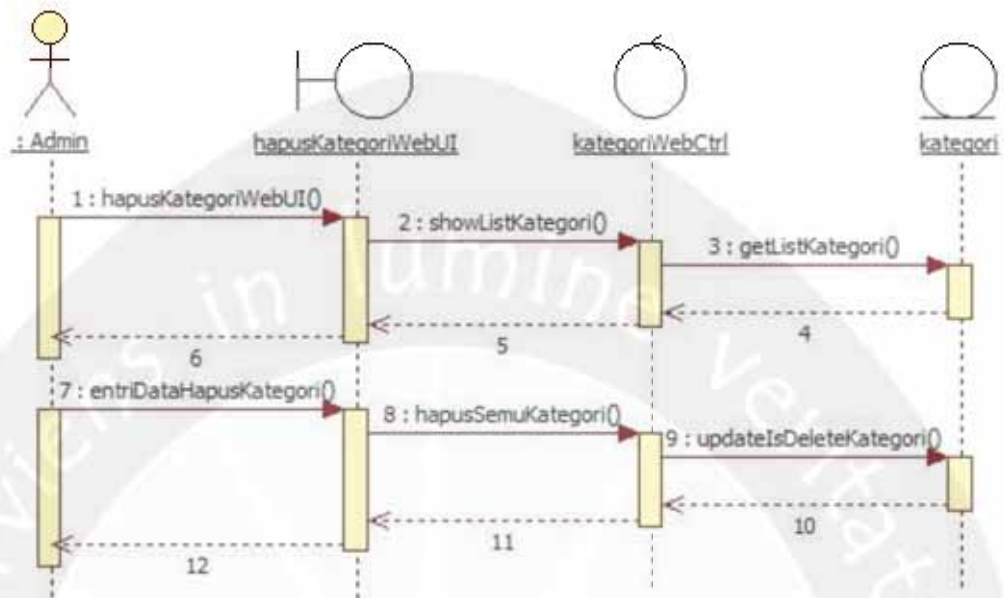
Gambar 2.25 Sequence Diagram Tambah Data Kategori Produk Web

2.2.19.3 Edit Data Kategori Produk



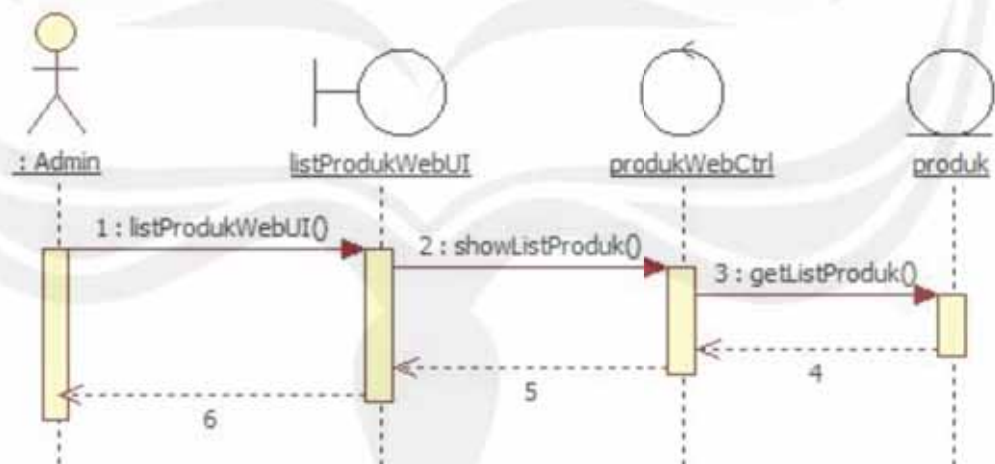
Gambar 2.26 Sequence Diagram Edit Data Kategori Produk Web

2.2.19.4 Hapus Data Kategori Produk



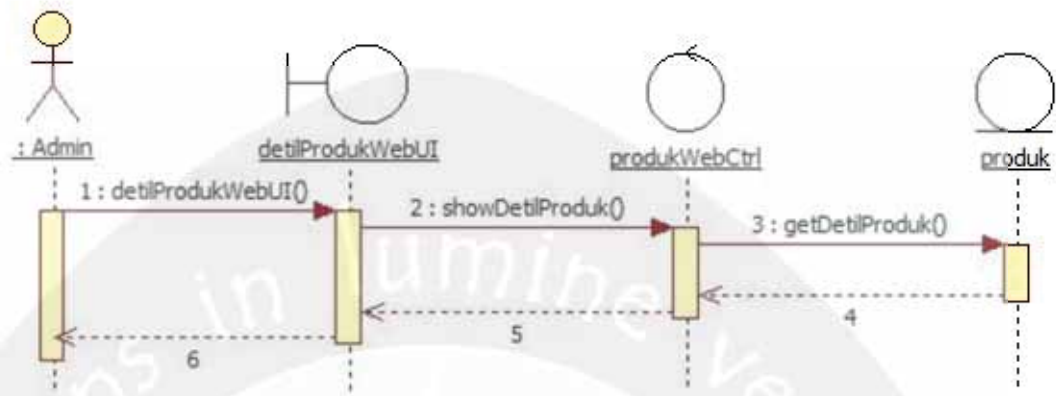
Gambar 2.27 Sequence Diagram Hapus Data Kategori Produk web

2.2.20 List Produk Web



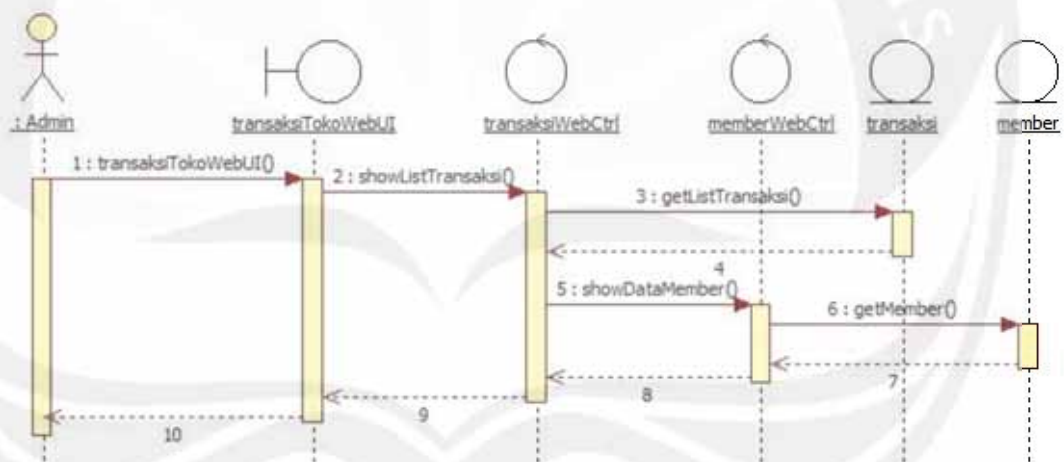
Gambar 2.28 Sequence Diagram List Produk Web

2.2.21 Detail Produk Web



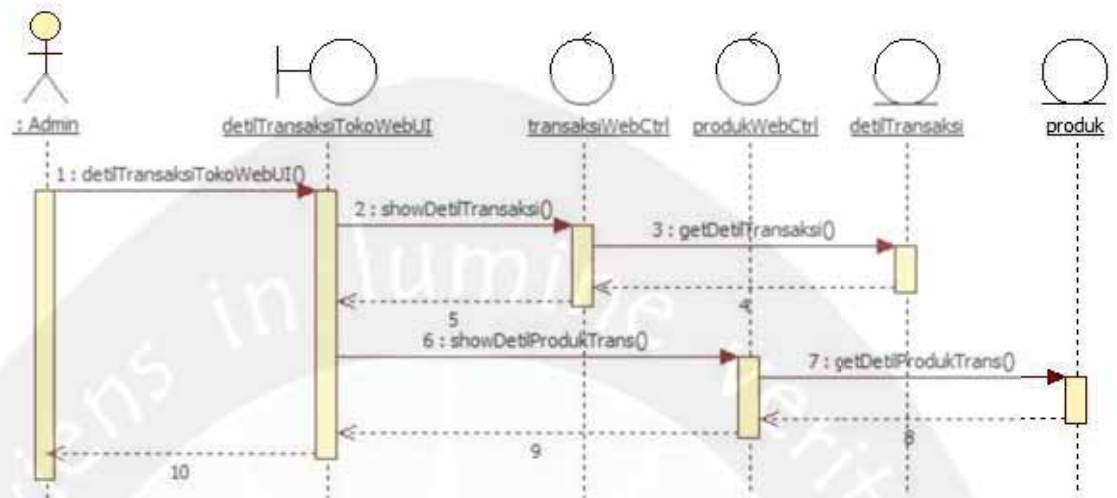
Gambar 2.29 Sequence Diagram Detail Produk Web

2.2.22 Pengawasan Transaksi Toko Web



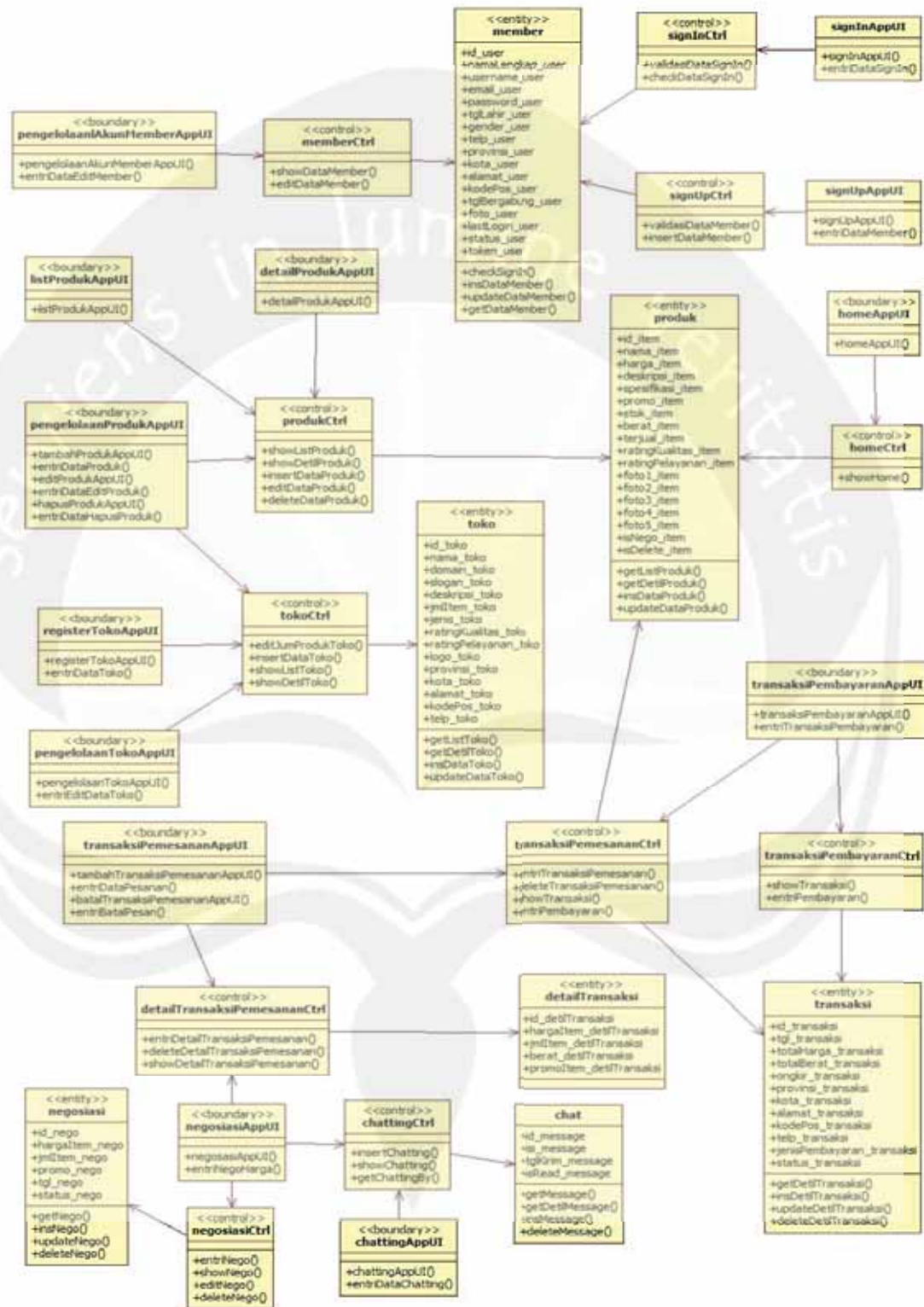
Gambar 2. 30 Sequence Diagram Pengawasan Transaksi Toko Web

2.2.23 Pengawasan Detil Transaksi Toko Web



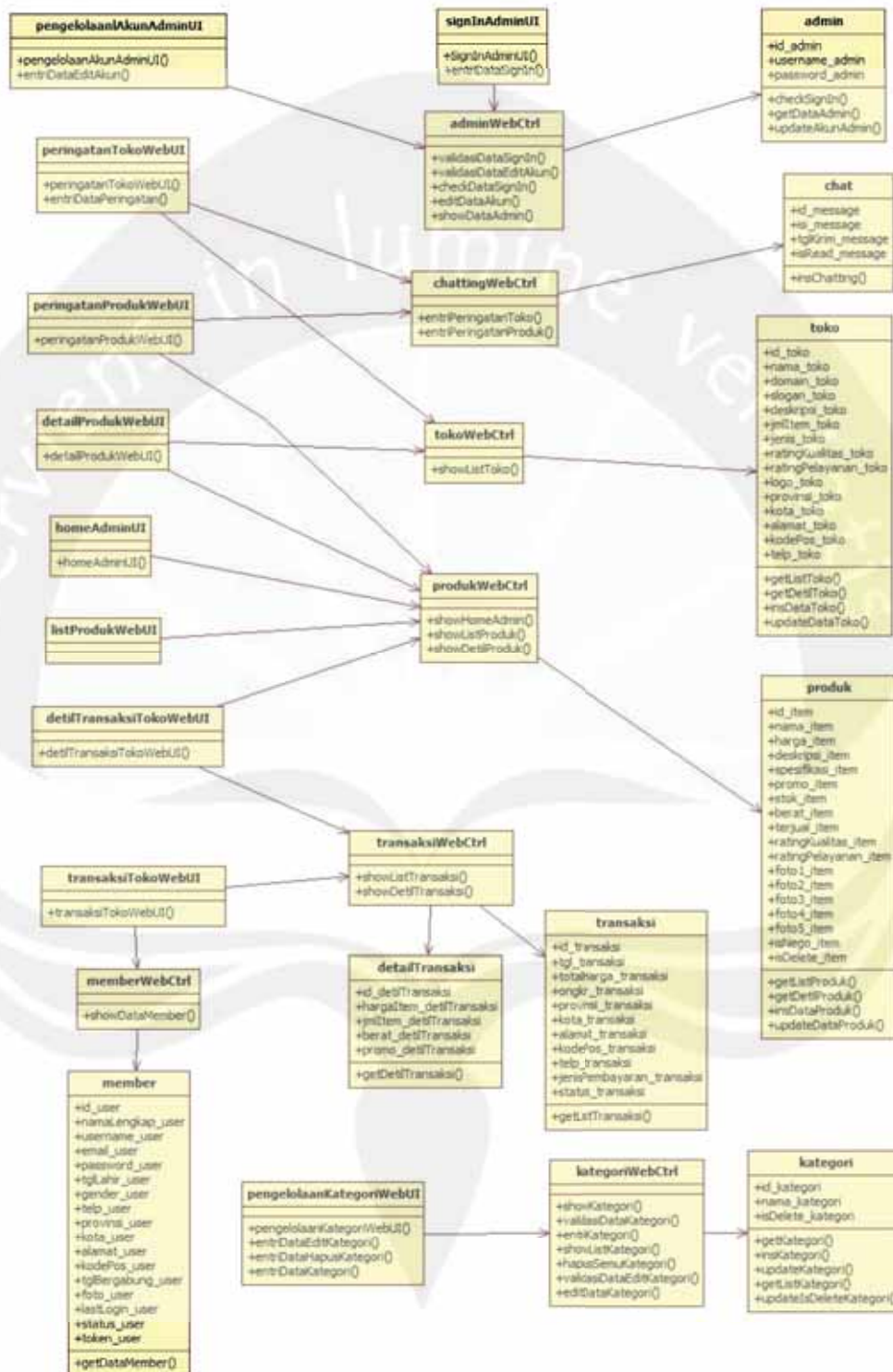
Gambar 2.31 Sequence Diagram Pengawasan Detil Transaksi Web

2.2.24 Class Diagram Mobile App



Gambar 2.32 Class Diagram TanioMall Mobile App

2.2.25 Class Diagram Web



Gambar 2.33 Class Diagram TanioMall Web

2.2.26 Class Diagram Specific Descriptions

2.2.26.1 Specific Design Class SignInAppUI

signInAppUI	<<boundary>>
<pre>+signInAppUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +entriDataSignIn() : Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu username dan password.</pre>	

2.2.26.2 Specific Design Class SignUpAppUI

signUpAppUI	<<boundary>>
<pre>+signUpAppUI() Digunakan untuk memanggil UI untuk merubah password +entriDataMember() digunakan untuk mengambil data member yang ada di form yang telah diisikan oleh member.</pre>	

2.2.26.3 Specific Design Class HomeAppUI

homeAppUI	<<boundary>>
<pre>+homeAppUI() sebagai default konstruktor dimana semua atribut pada class ini di inisialisasi.</pre>	

2.2.26.4 Specific Design Class PengelolaanAkunMemberAppUI

pengelolaanAkunMemberAppUI	<<boundary>>

+pengelolaanAkunMemberAppUI() sebagai default konstruktor dimana semua atribut pada class ini di inisialisasi.

+entriDataEditMember() digunakan untuk mengambil data member yang ada di form yang telah diisikan oleh member.

2.2.26.5 Specific Design Class RegisterTokoAppUI

registerTokoAppUI	<<boundary>>
+registerTokoAppUI[] Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +entriDataToko()digunakan untuk mengambil data toko dari form yang diinputkan oleh member.	

2.2.26.6 Specific Design Class Penge lo laanTokoAppUI

penge lo laanTokoAppUI	<<boundary>>
+pengelolaanTokoAppUI[] Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +entriEditDataToko()digunakan untuk mengambil data edit toko dari form yang diinputkan oleh member.	

2.2.26.7 Specific Design Class Penge lo laanProdukAppUI

penge lo laanProdukAppUI	<<boundary>>

+tambahProdukAppUI()

Default konstruktor, digunakan untuk inisialisasi attribute tambah dari kelas ini.

+entriDataProduk()digunakan untuk mengambil data produk dari form yang diinputkan oleh member.

+editProdukAppUI[]

Default konstruktor, digunakan untuk inisialisasi attribute edit dari kelas ini.

+editDataProduk[] digunakan untuk mengedit data produk yang ada di database.

+hapusProdukAppUI[]

Default konstruktor, digunakan untuk inisialisasi attribute hapus dari kelas ini.

+hapusJasa() digunakan untuk menghapus semu data jasa yang ada di database.

2.2.26.8 Specific Design Class ListProdukAppUI

ListProdukAppUI	<<boundary>>
+ListProdukAppUI[] Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	

2.2.26.9 Specific Design Class DetailProdukAppUI

detailProdukAppUI	<<boundary>>
+detailProdukAppUI[] Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	

2.2.26.10 Specific Design Class TransaksiPemesananAppUI

transaksiPemesananAppUI	<<boundary>>
<pre>+tambahTransaksiPemesananAppUI()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+entriDataPesanan()</pre> digunakan untuk mengambil data pesanan dari form yang diinputkan oleh member. <pre>+batalTransaksiPemesananAppUI()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+entriBatalPesanan()</pre> digunakan untuk mengambil data pembatalan pesanan dari form yang diinputkan oleh member.	

2.2.26.11 Specific Design Class TransaksiPembayaranAppUI

transaksiPembayaranAppUI	<<boundary>>
<pre>+transaksiPembayaranAppUI()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+entriTransaksiPembayaran()</pre> digunakan untuk mengambil data pembayaran dari form yang diinputkan oleh member.	

2.2.26.12 Specific Design Class NegosiasiAppUI

negosiasiAppUI	<<boundary>>
<pre>+negosiasiAppUI()</pre> Default konstruktor, digunakan untuk inisialisasi semua	

attribute dari kelas ini.

+entriNegoHarga() digunakan untuk mengambil data negosiasi harga dari form yang diinputkan oleh member.

2.2.26.13 Specific Design Class ChattingAppUI

chattingAppUI	<<boundary>>
<pre>+chattingAppUI {}</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+entriDataChatting {}</pre> digunakan untuk mengambil data chatting dari form yang diinputkan oleh member.	

2.2.26.14 Specific Design Class SignInAdminUI

signInAdminUI	<<boundary>>
<pre>+signInAdminUI {}</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+entriDataSignIn() :</pre> Operasi ini digunakan untuk mengambil data login yang diinputkan oleh admin, yaitu username dan password.	

2.2.26.15 Specific Design Class HomeAdminUI

homeAdminUI	<<boundary>>
<pre>+homeAdminUI {}</pre> sebagai default konstruktor dimana semua atribut pada class ini di inisialisasi.	

2.2.26.16 Specific Design Class PengelolaanAkunAdminUI

pengelolaanAkunAdminUI	<<boundary>>
+pengelolaanAkunAdminUI() sebagai default konstruktor dimana semua atribut pada class ini di inisialisasi. +entryDataEditMember() digunakan untuk mengambil data member yang ada di form yang telah diisikan oleh member.	

2.2.26.17 Specific Design Class PengelolaanKategoriProdukWebUI

pengelolaanKategoriProdukWebUI	<<boundary>>
+tambahProdukWebUI[] Default konstruktor, digunakan untuk inisialisasi attribute tambah dari kelas ini. +entryDataKategoriProduk() digunakan untuk mengambil data kategori dari form yang diinputkan oleh admin. +entryDataEditKategoriProduk() digunakan untuk mengedit data kategori yang ada di database. +entryDataHapusKategoriProduk() digunakan untuk menghapus semua data kategori yang ada di database.	

2.2.26.18 Specific Design Class ListProdukWebUI

listProdukWebUI	<<boundary>>
+listProdukWebUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	

2.2.26.19 Specific Design Class DetailProdukWebUI

detailProdukWebUI	<<boundary>>
<pre>+detailProdukWebUI{} Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</pre>	

2.2.26.20 Specific Design Class TransaksiTokoWebUI

transaksiTokoUI	<<boundary>>
<pre>+transaksiTokoWebUI{} Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</pre>	

2.2.26.21 Specific Design Class DetilTransaksiTokoWebUI

detilTransaksiTokoWebUI	<<boundary>>
<pre>+detilTransaksiTokoWebUI{} Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</pre>	

2.2.26.22 Specific Design Class PeringatanTokoWebUI

peringatanTokoWebUI	<<boundary>>
<pre>+peringatanTokoWebUI{} Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +entriPeringatanToko()digunakan untuk mengambil data peringatan dari form yang diinputkan oleh admin.</pre>	

2.2.26.1 Specific Design Class PeringatanProdukWebUI

peringatanProdukWebUI	<<boundary>>
<pre>+peringatanProdukWebUI()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+entriPeringatanProduk()</pre> digunakan untuk mengambil data peringatan dari form yang diinputkan oleh admin.	

2.2.26.2 Specific Design Class AdminCtrl()

memberCtrl	<<control>>
<pre>+memberCtrl()</pre> default konstruktor yang menginisialisasi semua atribut pada class ini. <pre>+validasiDataSignIn()</pre> operasi ini digunakan untuk mengecek format data Sign In yang diberikan oleh user lewat form. <pre>+checkDataSignIn()</pre> operasi ini digunakan untuk mencocokkan data Sign In yang diberikan oleh user lewat form, dengan data yang ada di database. <pre>+validasiDataEditAkun()</pre> operasi ini digunakan untuk mengecek format data edit akun yang diberikan oleh user lewat form. <pre>+showDataAdmin()</pre> operasi yang digunakan untuk menampilkan semua data admin. <pre>+editDataAkun()</pre> operasi ini digunakan untuk mengedit data admin yang ada di database.	

2.2.26.3 Specific Design Class TokoWebCtrl()

tokoWebCtrl	<<control>>
+tokoWebCtrl() default konstruktor yang menginisialisasi semua atribut pada class ini +showListToko() operasi ini digunakan untuk menampilkan semua data toko dari database.	

2.2.26.4 Specific Design Class ProdukWebCtrl()

produkWebCtrl	<<control>>
+produkWebCtrl() default konstruktor yang digunakan untuk inisialisasi semua atribut pada kelas ini +showHomeAdmin() operasi ini digunakan untuk menampilkan semua data produk yang atribut is_deleted = False dari database. +showListProduk() operasi ini digunakan untuk menampilkan semua data produk yang atribut is_deleted = False dari database. +showDetailProduk() operasi ini digunakan untuk menampilkan semua data detail produk yang atribut is_deleted = False dari database.	

2.2.26.5 Specific Design Class TransaksiWebCtrl()

transaksiWebCtrl	<<control>>
+transaksiWebCtrl() default konstruktor yang digunakan untuk inisialisasi semua atribut pada kelas ini +showListTransaksi() operasi ini digunakan untuk menampilkan semua data transaksi yang atribut	

status_transaksi = Pesan dari database.
 +showDEtilTransaksi() operasi ini digunakan untuk menampilkan data detil transaksi yang atribut status_transaksi – Pesan dari database.

2.2.26.6 Specific Design Class ChattingWebCtrl[]

ChattingWebCtrl	<<control>>
+chattingWebCtrl() default konstruktor, digunakan untuk meng inisialisasi semua atribut yang ada di kelas ini +entriPeringatanToko() operasi ini digunakan untuk memasukkan data peringatan toko dari inputan user ke dalam database. +entriPeringatanProduk() operasi ini digunakan untuk memasukkan data peringatan produk dari inputan user ke dalam database.	

2.2.26.7 Specific Design Class Member

Member	<<entity>>
-id user: integer, digunakan untuk menyimpan data numerik identitas user. -namaLengkap_user: string, digunakan untuk menyimpan data nama lengkap user. -username_user: string, digunakan untuk menyimpan data username user. -email_user: string, digunakan untuk menyimpan data email user. -password_user: string, digunakan untuk menyimpan data password user. -tglLahir_user: string, digunakan untuk menyimpan data tanggal lahir user.	

-gender_user: integer, digunakan untuk menyimpan data gender user.

-telp_user: string, digunakan untuk menyimpan data telepon user.

-provinsi_user: string, digunakan untuk menyimpan data provinsi user.

-kota_user: string, digunakan untuk menyimpan data kota user.

-alamat_user: string, digunakan untuk menyimpan data alamat user.

-kodePos_user: string, digunakan untuk menyimpan data kode pos user.

-tglBergabung_user: date, digunakan untuk menyimpan data tanggal bergabung user.

-foto_user: string, digunakan untuk menyimpan data foto user.

-lastLogin_user: date, digunakan untuk menyimpan data terakhir login user.

-status_user: integer, digunakan untuk menyimpan data status aktif user.

-token_user: string, digunakan untuk menyimpan data token user.

+member() default konstruktor untuk menginisialisasikan seluruh variable di kelas ini

+checkSignIn() operasi ini digunakan untuk mengecek data username dan password yang diinputkan user melalui form dengan data member pada database.

+getDataMember() operasi ini digunakan untuk mengambil semua data member dari database

+insDataMember() operasi ini digunakan untuk mengentrikan data member ke dalam database.

+updateDataMember() operasi ini digunakan untuk mengedit data member yang ada di database

2.2.26.8 Specific Design Class Toko

Toko	<<entity>>
-id toko: int ,digunakan untuk menyimpan data numerik identitas user -nama_toko: string, digunakan untuk menyimpan data nama toko. -domain_toko: string, digunakan untuk menyimpan data domain toko. -slogan_toko: string , digunakan untuk menyimpan data slogan toko -deskripsi_toko: string , digunakan untuk menyimpan data deskripsi toko -jmlItem_toko: integer , digunakan untuk menyimpan data jumlah item toko -jenis_toko: integer , digunakan untuk menyimpan data jenis toko -ratingKualitas_toko: float , digunakan untuk menyimpan data rating kualitas toko -ratingPelayanan_toko: float , digunakan untuk menyimpan data rating pelayanan petugas -logo_toko: string , digunakan untuk menyimpan data logo toko -provinsi_toko: string , digunakan untuk menyimpan data provinsi toko -kota_toko: string , digunakan untuk menyimpan data kota toko -alamat_toko: string , digunakan untuk menyimpan data	

alamat toko

-kodePos_toko: string , digunakan untuk menyimpan data kode pos toko

-telp_toko: string , digunakan untuk menyimpan data telepon toko

+toko() default konstruktor, digunakan untuk menginisialisasi semua atribut pada kelas ini.

+getListToko() operasi ini digunakan untuk mengambil semua data toko yang ada di database.

+getDetailToko() operasi ini digunakan untuk mengambil data toko berdasarkan id_toko.

+insDataToko() operasi ini digunakan untuk menginputkan data toko kedalam data base.

+updateDataToko () operasi ini digunakan untuk mengedit data toko yang ada di database

2.2.26.9 Specific Design Class Produk

produk	<<Entity>>
-id item: integer, digunakan untuk menyimpan data nomor identitas dari item. -nama item: string, digunakan untuk menyimpan data nama item. -harga item: decimal, digunakan untuk menyimpan data harga item. -deskripsi item: string, digunakan untuk menyimpan data deskripsi item. -spesifikasi item: string, digunakan untuk menyimpan data spesifikasi item. -promo_item: decimal, digunakan untuk menyimpan data harga promo item. -stok_item: integer, digunakan untuk menyimpan data	

stok item.

-berat_item: integer, digunakan untuk menyimpan data berat item.

-terjual_item: integer, digunakan untuk menyimpan data terjual item.

-ratingKualitas_item: integer, digunakan untuk menyimpan data rating kualitas item.

-ratingPelayanan_item: integer, digunakan untuk menyimpan data rating pelayanan item.

-foto1_item: integer, digunakan untuk menyimpan data foto1 item.

-foto2_item: integer, digunakan untuk menyimpan data foto2 item.

-foto3_item: integer, digunakan untuk menyimpan data foto3 item.

-foto4_item: integer, digunakan untuk menyimpan data foto4 item.

-foto5_item: integer, digunakan untuk menyimpan data foto5 item.

-isNegotiation_item: integer, digunakan untuk menyimpan status negosiasi harga data item.

-isDelete_item: byte, digunakan untuk menyimpan status hapus semua data item.

+produk() default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada di kelas ini.

+getListProduk() operasi ini digunakan untuk mengambil semua data item dari database. Kecuali yang isDelete_item = True

+getDetailProduk() operasi ini digunakan untuk mengambil data detail item dari database. Kecuali yang isDelete_item = True

+insDataProduk() operasi ini digunakan untuk memasukkan data item ke dalam database.

+updateDataProduk() operasi ini digunakan untuk mengedit data item yang ada di database

+deleteDataProduk() operasi ini digunakan untuk menghapus semua data item yang ada di database.

2.2.26.10 Specific Design Class Transaksi

transaksi	<<Entity>>
-id transaksi: integer, digunakan untuk menyimpan data nomor identitas transaksi. -tgl_transaksi: date, digunakan untuk menyimpan data tanggal transaksi. -totalHarga_transaksi: decimal, digunakan untuk menyimpan data total harga transaksi. -totalBerat_transaksi: float, digunakan untuk menyimpan data harga transaksi. -ongkir_transaksi: decimal, digunakan untuk menyimpan data harga ongkos kirim transaksi. -provinsi_transaksi: string, digunakan untuk menyimpan data provinsi transaksi. -kota_transaksi: string, digunakan untuk menyimpan data kota transaksi. -alamat_transaksi: string, digunakan untuk menyimpan data alamat transaksi. -kodePos_transaksi: string, digunakan untuk menyimpan data kode pos transaksi. -telp_transaksi: string, digunakan untuk menyimpan data telepon transaksi. -jenisPembayaran_transaksi: integer, digunakan untuk menyimpan data jenis pembayaran transaksi.	

-status_transaksi: integer, digunakan untuk menyimpan status transaksi data transaksi.

+transaksi() default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada dikelas ini.

+getDetilTransaksi() operasi ini digunakan untuk mengambil semua data transaksi dari database. Kecuali yang status_transaksi = Lunas

+insDataTransaksi() operasi ini digunakan untuk memasukkan data transaksi ke dalam database.

+updateDataTransaksi() operasi ini digunakan untuk mengedit data transaksi yang ada di database

+deleteDataTransaksi() operasi ini digunakan untuk menghapus data transaksi yang ada di database.

2.2.26.11 Specific Design Class DetilTransaksi

detilTransaksi	<<Entity>>
-id_detilTransaksi: integer, digunakan untuk menyimpan data nomor identitas detil transaksi. -hargaItem detilTransaksi: decimal, digunakan untuk menyimpan data harga item pada detil transaksi. -jmlItem_detilTransaksi: integer, digunakan untuk menyimpan data jumlah item pada detil transaksi. -berat detilTransaksi: float, digunakan untuk menyimpan data total berat item pada detil transaksi. -promoItem detilTransaksi: decimal, digunakan untuk menyimpan data harga promo pada detil transaksi.	
+detilTransaksi() default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada dikelas ini. +getDataDetilTransaksi() operasi ini digunakan untuk mengambil data detil transaksi dari database. Kecuali	

yang status_transaksi = Lunas

+insDataDetilTransaksi() operasi ini digunakan untuk memasukkan data detil transaksi ke dalam database.

+updateDataDetilTransaksi() operasi ini digunakan untuk mengedit data detil transaksi yang ada di database

+deleteDataDetilTransaksi() operasi ini digunakan untuk menghapus data detil transaksi yang ada di database.

2.2.26.12 Specific Design Class Negosiasi

negosiasi	<<Entity>>
-id_nego: integer, digunakan untuk menyimpan data nomor identitas negosiasi. -hargaitem_nego: decimal, digunakan untuk menyimpan data harga item negosiasi. -jmlItem_nego: integer, digunakan untuk menyimpan data jumlah item negosiasi. -promo_nego: decimal, digunakan untuk menyimpan data promo harga negosiasi. -tgl_nego: date, digunakan untuk menyimpan data tanggal negosiasi. -status_nego: integer, digunakan untuk menyimpan data status negosiasi.	
+negosiasi{} defaultkonstruktor, digunakan untuk menginisialisasi semua atribut yang ada dikelas ini. +getNego{} operasi ini digunakan untuk mengambil seluruh data negosiasi dari table negosiasi. +insNego {} operasi ini dilakukan untuk memasukkan data rincian negosiasi ke table negosiasi. +updateNego() operasi ini dilakukan untuk mengedit data rincian negosiasi yang ada di table negosiasi.	

+deleteNego() operasi ini dilakukan untuk menghapus data negosiasi berdasarkan nomer negosiasi.

2.2.26.13 Specific Design Class Chatting

chatting	<<Entity>>
-id message: string, digunakan untuk menyimpan data karakter identitas pesan. -isi_message: string, digunakan untuk menyimpan data isi pesan. -tglKirim_message: date, digunakan untuk menyimpan data tanggal kirim pesan. -isRead_message: integer, digunakan untuk menyimpan status terbaca pesan.	
+chatting() default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada pada kelas ini. +getMessage() operasi yang digunakan untuk mengambil seluruh data pesan yang ada di table message. +getDetilMessage() operasi yang digunakan untuk mengambil seluruh data pesan yang ada di table message. +insMessage() operasi yang digunakan untuk menyimpan data pesan. +deleteMessage() operasi yang digunakan untuk menghapus data pesan yang ada di table message.	

2.2.26.1 Specific Design Class Admin

chatting	<<Entity>>
-id_admin: integer, digunakan untuk menyimpan data karakter identitas admin. -username_admin: string, digunakan untuk menyimpan data username.	

-password_admin: string, digunakan untuk menyimpan data kata sandi admin.

+admin() default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada pada kelas ini.

+checkSignIn() operasi yang digunakan untuk mengecek data sign in yang ada di table admin.

+getDataMessage() operasi yang digunakan untuk mengambil seluruh data admin yang ada di table admin.

+updateAkunAdmin() operasi yang digunakan untuk mengedit data admin yang ada di table admin.

2.2.26.2 Specific Design Class Kategori

chatting	<<Entity>>
----------	------------

-id_kategori: integer, digunakan untuk menyimpan data karakter identitas kategori.

-nama_kategori: string, digunakan untuk menyimpan data kategori.

-isDelete_kategori: integer, digunakan untuk menyimpan status hapus semu kategori.

+kategori() default konstruktor, digunakan untuk menginisialisasikan semua atribut yang ada pada kelas ini.

+getKategori() operasi yang digunakan untuk mengambil data detail kategori yang ada di table kategori.

+getListKategori() operasi yang digunakan untuk mengambil seluruh data kategori yang ada di table kategori.

+insKategori() operasi yang digunakan untuk menyimpan data kategori yang ada di table kategori.

+updateKategori() operasi yang digunakan untuk mengedit

data kategori yang ada di table kategori.
+updateIsDeleteKategori() operasi yang digunakan untuk
mengedit status hapus semu kategori.



3 PERANCANGAN DATA

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas Member

Nama	Tipe	Panjang	Keterangan
ID_USER	int		Kode dari member , primary key
NAMALENGKAP_USER	varchar	50	Nama lengkap dari member
USERNAME_USER	varchar	20	Nama sandi dari member
EMAIL_USER	varchar	50	Email dari member
PASSWORD_USER	varchar	20	Kata sandi dari member
TGLLAHIR_USER	date		Tanggal lahir dari member
GENDER_USER	int		Jenis kelamin dari member
TELP_USER	varchar	20	Telepon dari member
PROVINSI_USER	varchar	30	Provinsi dari member
KOTA_USER	varchar	30	Kota dari member
ALAMAT_USER	text		Alamat dari member
KODEPOS_USER	varchar	10	Kode Pos dari member
TGLBERGABUNG_USER	date		Tanggal bergabung member
FOTO_USER	varchar	100	Foto member
LASTLOGIN_USER	datetime		Tanggal terakhir member login
STATUS_USER	int		Status aktif member
TOKEN_USER	varchar	30	Token dari member

3.1.2 Deskripsi Entitas Toko

Nama	Tipe	Panjang	Keterangan
ID_TOKO	int		Kode dari toko, primary key
ID_USER	int		Kode dari member, foreign key
NAMA_TOKO	varchar	50	Nama dari toko

DOMAIN_TOKO	varchar	50	Domain dari toko
SLOGAN_TOKO	text		Slogan dari toko
DESKRIPSI_TOKO	text		Deskripsi dari toko
JMLITEM_TOKO	int		Jumlah item yang dimiliki toko
JENIS_TOKO	int		Jenis dari toko
RATINGPELAYANAN_TOKO	float		Rating pelayanan dari toko
RATINGKUALITAS_TOKO	float		Rating kualitas dari toko
LOGO_TOKO	varchar	100	Logo dari toko
PROVINSI_TOKO	varchar	30	Provinsi dari toko
KOTA_TOKO	varchar	30	Kota dari toko
ALAMAT_TOKO	text		Alamat dari toko
KODEPOS_TOKO	varchar	20	Kode pos dari toko
TELP_TOKO	varchar	20	Telepon dari toko

3.1.3 Deskripsi Entitas Kategori

Nama	Tipe	Panjang	Keterangan
ID_KATEGORI	int		Kode dari kategori, primary key
NAMA_KATEGORI	varchar	30	Nama dari kategori
FOTO_KATEGORI	varchar	100	Foto dari kategori

3.1.4 Deskripsi Entitas SubKategori

Nama	Tipe	Panjang	Keterangan
ID_SUBKATEGORI	int		Kode dari sub kategori, primary key
ID_KATEGORI	int		Kode dari kategori, foreign key
NAMA_SUBKATEGORI	varchar	30	Nama kategori
FOTO_SUBKATEGORI	varchar	100	Foto dari kategori

3.1.5 Deskripsi Entitas Item

Nama	Tipe	Panjang	Keterangan
ID_ITEM	int		Kode dari jenis jasa, primary key
ID_SUBKATEGORI	int		Kode dari sub kategori, foreign key
ID_TOKO	int		Kode dari toko, foreign key
NAMA_ITEM	varchar	30	Nama dari item
HARGA_ITEM	decimal	18,0	Harga dari item
DESKRIPSI_ITEM	text		Deskripsi dari item
SPEKIFIKASI_ITEM	text		Spesifikasi dari item
PROMO_ITEM	decimal	18,0	Promo dari item
STOK_ITEM	int		Stok dari item
DILIHAT_ITEM	int		Jumlah dilihat dari item
TERJUAL_ITEM	int		Jumlah terjual dari item
RATING_ITEM	float		Rating dari item
FOTO1_ITEM	varchar	100	Foto pertama dari item
FOTO2_ITEM	varchar	100	Foto kedua dari item
FOTO3_ITEM	varchar	100	Foto ketiga dari item
FOTO4_ITEM	varchar	100	Foto keempat dari item
FOTO5_ITEM	varchar	100	Foto kelima dari item
ISNEGO_ITEM	int		Status dapat melakukan nego dari item
ISDELETE_ITEM	int		Status hapus semu dari item

3.1.6 Deskripsi Entitas Testimonial

Nama	Tipe	Panjang	Keterangan
ID_TESTIMONIAL	int		Kode dari testimonial

			perbaikan, primary key
ID_DETILTRANSAKSI	int		Kode dari detil transaksi, foreign key
JENIS_TESTIMONIAL	int		Jenis dari testimonial
ISI_TESTIMONIAL	text		Isi dari testimonial
TGLKIRIM_TESTIMONIAL	datetime		Tanggal kirim dari testimonial
RATING_KEPERCAYAAN	float		Rating kepercayaan dari toko
RATING_KUALITAS	float		Rating kualitas dari item
RATING_PELAYANAN	float		Rating pelayanan dari toko

3.1.7 Deskripsi Entitas Message

Nama	Tipe	Panjang	Keterangan
ID_MESSAGE	int		Kode pesan, foreignkey, primary key
ID_USER	int		Kode user pengirim, foreignkey
ID_USERPENERIMA	int		Kode user penerima, foreignkey
ISI_MESSAGE	text		Isi dari pesan
TGLKIRIM_MESSAGE	datetime		Tanggal dan jam kirim pesan
ISREAD_MESSAGE	int		Status terbaca pesan

3.1.8 Deskripsi Entitas Favorit

Nama	Tipe	Panjang	Keterangan
ID_ITEM	int		Kode item pada favorit, foreignkey, primarykey
ID_USER	int		Kode user pada favorit, foreignkey, primary key

3.1.9 Deskripsi Entitas Nego

Nama	Tipe	Panjang	Keterangan
ID_NEGO	int		Kode nego, primary key
ID_USER	int		Kode user, foreignkey
ID_ITEM	int		Kode item, foreignkey
HARGAITEM_NEGO	decimal	18,0	Harga item pada nego
JMLITEM_NEGO	int		Jumlah item pada nego
PROMO_NEGO	decimal	18,0	Harga promo pada nego
TGL_NEGO	datetime		Tanggal dan jam dari nego

3.1.10 Deskripsi Entitas Transaksi

Nama	Tipe	Panjang	Keterangan
ID_TRANSAKSI	int		Kode nego, primary key
ID_USER	int		Kode user, foreignkey
ID_TOKO	int		Kode toko, foreignkey
TGL_TRANSAKSI	datetime		Tanggal dan jam dari transaksi
TOTALHARGA_TRANSAKSI	decimal	18,0	Total harga dari transaksi
TOTALBERAT_TRANSAKSI	int		Total berat item dari transaksi
ONGKIR_TRANSAKSI	decimal	18,0	Ongkos kirim dari transaksi
PROVINSI_TRANSAKSI	varchar	30	Provinsi dari transaksi
KOTA_TRANSAKSI	varchar	30	Kota dari transaksi
ALAMAT_TRANSAKSI	text		Alamat dari transaksi
KODEPOS_TRANSAKSI	varchar	20	Kode pos

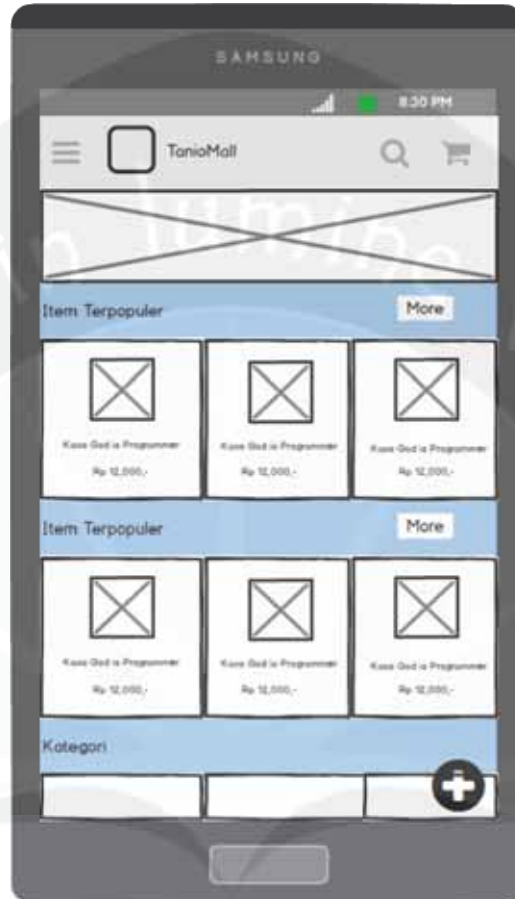
			dari transaksi
TELP_TRANSAKSI	varchar	20	Telepon dari transaksi
JENISPEMBARAN_TRANSAKSI	int		Jenis pembayaran dari transaksi
STATUS_TRANSAKSI	int		Status dari transaksi

3.1.11 Deskripsi Entitas DetilTransaksi

Nama	Tipe	Panjang	Keterangan
ID_DETILTRANSAKSI	int		Kode nego, primary key
ID_TRANSAKSI	int		Kode transaksi, foreignkey
ID_ITEM	int		Kode item, foreignkey
HARGAITEM_DETILTRANSAKSI	decimal	18,0	Harga item pada detil transaksi
JMLITEM_DETILTRANSAKSI	int		Jumlah item pada detil transaksi
PROMO_DETILTRANSAKSI	decimal	18,0	Harga promo pada detil transaksi
BERAT_DETILTRANSAKSI	float		Berat total item dari detil transaksi

4 Perancangan Antarmuka

4.1 Antarmuka Halaman *Home*



Gambar 4. 1 Rancangan Antarmuka Halaman Home

Rancangan Antarmuka gambar 4.1 adalah rancangan antarmuka ketika pengguna melihat halaman utama atau *home* pada aplikasi *mobile*. Pengguna dapat melihat barang-barang terpopuler, barang-barang terbaru, dan kategori barang yang tersedia. Selain itu pengguna dapat melihat menu yang terdapat pada aplikasi. Menu aplikasi terdiri dari 2 jenis yaitu menu untuk pengunjung dan menu untuk pengguna yang telah mendaftar sebagai *member*. Berikut rancangan antarmuka menu untuk pengunjung seperti yang ditunjukkan pada gambar 4.2.



Gambar 4. 2 Rancangan Antarmuka Halaman Menu Pengunjung

Pada gambar 4.2 menunjukkan rancangan antarmuka menu untuk pengunjung. Menu yang disediakan bagi pengunjung yaitu *home* untuk menuju ke halaman *home*, kategori barang untuk melihat daftar kategori barang, *sign in* untuk melakukan *sign in* apabila pengunjung telah mendaftar sebagai *member* aplikasi, dan *sign up* untuk mendaftar sebagai *member* pada aplikasi TanioMall.



Gambar 4.3 Rancangan Antarmuka Halaman Menu *Member*

Pada gambar 4.3 menunjukkan rancangan antarmuka menu untuk *member*. Menu untuk *member* ini terdapat menu home yang digunakan untuk menuju ke halaman *home*, kategori barang yang digunakan untuk melihat kategori barang, profil yang digunakan untuk melihat detail profil *member*, toko yang digunakan untuk melihat detail toko *member*, pesan yang digunakan untuk melihat pesan singkat yang masuk, keranjang belanja untuk melihat daftar belanja yang dimiliki oleh *member*, nego yang digunakan untuk melihat negosiasi pembelian dan penjualan yang dimiliki oleh *member*, selanjutnya menu transaksi yang digunakan untuk melihat dan mengelola transaksi pembelian dan penjualan yang dimiliki oleh *member*, dan menu *logout*.

4.2 Antarmuka Pencarian



Gambar 4.4 Rancangan Antarmuka Halaman Pencarian

Rancangan Antarmuka gambar 4.4 adalah rancangan antarmuka ketika pengguna ingin melakukan pencarian barang. Pengguna diminta untuk mengisi kata pencariannya, kategori barang, lokasi toko, harga nego, dan harga minimum dan maksimumnya. Setelah mengisi dan menekan tombol cari maka aplikasi akan menampilkan barang-barang yang sesuai dengan kriteria pencarian yang telah didefinisikan oleh pengguna.

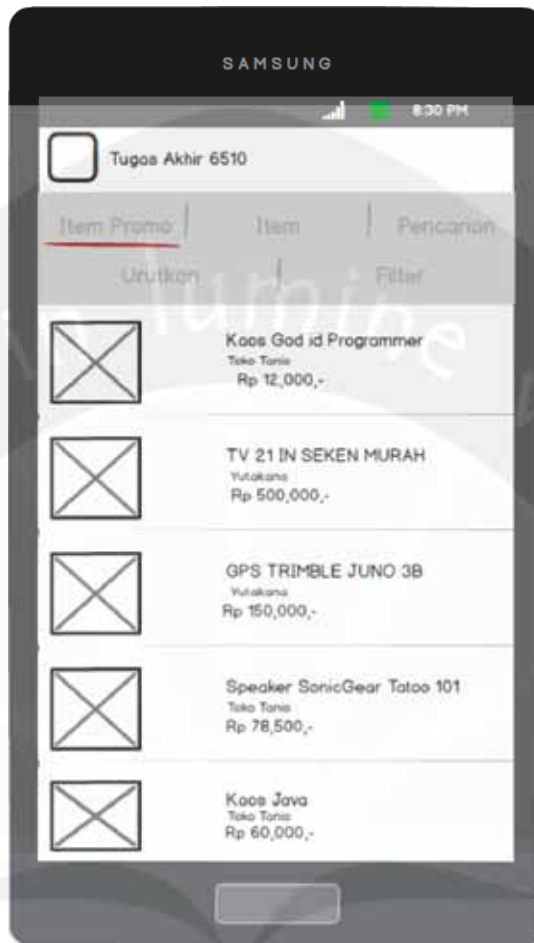
4.3 Antarmuka Kategori Barang



Gambar 4.5 Rancangan Antarmuka Halaman Kategori Barang

Rancangan Antarmuka gambar 4.5 merupakan antarmuka halaman kategori barang yang terdapat pada aplikasi TanioMall. Pengguna dapat melihat kategori barang dan dapat memilih kategori barang untuk melihat barang-barang sesuai dengan kategori barang yang dipilih oleh pengguna.

4.4 Antarmuka List Barang



Gambar 4.6 Rancangan Antarmuka Halaman List Barang

Rancangan Antarmuka gambar 4.6 merupakan antarmuka halaman list barang. Pada halaman ini pengguna dapat melihat daftar barang, dapat mengurutkan, dapat memfilter, dan dapat melakukan pencarian barang sesuai dengan kriteria barang yang dicari. Ketika suatu barang dipilih maka akan menampilkan detail barang yang dipilih.



Gambar 4.7 Rancangan Antarmuka Halaman Filter List Barang

Pada gambar 4.7 menunjukkan rancangan antarmuka filtrasi pencarian barang berdasarkan kategori barang, lokasi toko, barang memiliki harga nego, harga minimum dan maksimum. Setelah pengguna menekan tombol OK maka akan tampil barang sesuai dengan kriteria filtrasi pencarian barang.



Gambar 4.8 Rancangan Antarmuka Pengurutan List Barang

Pada gambar 4.8 menunjukkan rancangan antarmuka pengurutan list barang dengan kategori list barang terbaru, termurah, dan termahal. Setelah memilih jenis pengurutan list barang maka list barang akan terurut sesuai dengan jenis pengurutan yang dipilih oleh pengguna aplikasi TanioMall.

4.5 Antarmuka Halaman Detil Barang



Gambar 4.9 Rancangan Antarmuka Detil Barang Nego



Gambar 4.10 Rancangan Antarmuka Halaman Detil Barang Tanpa Nego

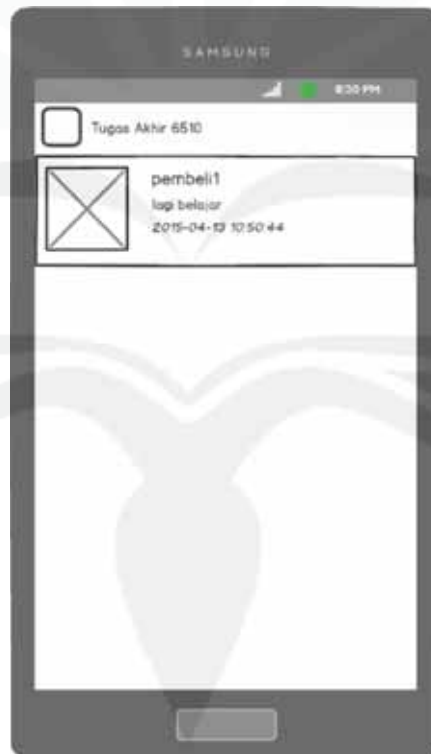


**Gambar 4.11 Rancangan Antarmuka Halaman Detil Barang Milik
Member**

Halaman detil barang memiliki 3 jenis, yaitu halaman detil barang untuk barang yang dapat melakukan negosiasi harga, halaman detil barang untuk barang yang tidak dapat melakukan negosiasi harga, dan halaman detil barang untuk barang milik toko sendiri. Pada rancangan antarmuka halaman detil barang seperti yang ditunjukkan pada gambar 4.9, 4.10, dan 4.11 masing-masing halaman menampilkan foto barang, nama barang, harga, jumlah dilihat, jumlah terjual, jumlah ulasan dari *member* yang telah membeli barang tersebut, *rating* kualitas barang dan pelayanan toko, deskripsi barang, dan spesifikasi barang. Dari ketiga rancangan antarmuka tersebut yang membedakan adalah tombol pengelolaan yang dimiliki oleh masing-masing antarmuka, pada rancangan antarmuka detil barang untuk barang yang dapat

melakukan negosiasi barang seperti pada gambar 4.9 memiliki tombol beli untuk membeli barang dan tombol nego yang digunakan untuk mengajukan harga negosiasi, sedangkan pada rancangan antarmuka halaman detail barang untuk barang yang tidak dapat melakukan negosiasi harga seperti pada gambar 4.10 hanya terdapat tombol beli untuk membeli barang. Yang membedakan pada rancangan antarmuka detail barang untuk barang milik *member* itu sendiri seperti pada gambar 4.11 hanya terdapat tombol edit yang digunakan untuk mengedit detail barang yang dimiliki.

4.6 Antarmuka Halaman *List Pesan Nego*



Gambar 4.12 Rancangan Antarmuka Halaman Pesan Nego

Pada gambar 4.12 menunjukkan rancangan antarmuka list pesan nego. Halaman tersebut digunakan untuk melihat history pesan nego kepada pembeli maupun

penjual. Semua pesan yang masuk maupun keluar akan dicatat di halaman ini. Jika kita pilih salah satu pesan dari *list* maka akan muncul halaman untuk *chat* seperti gambar 4.13.



Gambar 4.13 Antarmuka Halaman Detil Pesan Nego

Rancangan antarmuka pada gambar 4.13 menunjukkan halaman untuk *chat*. Pembeli maupun penjual dapat bertukar pesan satu sama lain dengan cara mengetikkan pesannya ke dalam textbox yang sudah tersedia, apabila telah selesai klik tombol *send* untuk mengirim pesan. Jika ada pesan masuk, maka secara otomatis akan muncul pop up atau pemberitahuan pesan masuk di *smartphone*.

4.7 Antarmuka Halaman Tambah Barang

The screenshot shows a mobile application interface for adding a new item. The screen is titled "Tambah Barang" and displays a form with the following fields:

- Foto Barang:** A section for uploading photos, showing "0 / 5" photos and a note "Minimal 1 (satu) foto barang".
- Detail Barang:** A section for item details, including:
 - Nama Item:** A text input field.
 - Harga Item:** A text input field with a "Rp." prefix.
 - Harga Promo:** A text input field with a "Rp." prefix.
 - Baru:** A toggle switch.
 - Dua Naga:** A toggle switch.
 - Berat:** A text input field with a "Perkiraan Berat" label and a "gram" unit.
 - Jumlah Stok:** A text input field.

Gambar 4.14 Rancangan Antarmuka Halaman Tambah Barang oleh Member

Rancangan antarmuka pada gambar 4.14 menunjukkan halaman untuk tambah barang toko. Penjual dapat menambahkan barang yang akan dijual. Penjual harus mengupload minimal satu foto barang yang akan dijual, kemudian mengisi informasi tentang barang yang akan dijual pada form yang telah disediakan. Setelah informasi terisi dengan lengkap, kemudian klik simpan. Barang berhasil ditambahkan dalam toko.

4.8 Antarmuka Halaman Edit Barang



**Gambar 4.15 Rancangan Antarmuka Halaman Edit Barang oleh
Member**

Rancangan antarmuka pada gambar 4.15 menunjukkan halaman untuk *edit* barang toko. Penjual dapat *mengedit* informasi maupun menambah foto barang yang akan dijual. Penjual harus *upload* minimal satu foto barang yang akan dijual, kemudian mengisi informasi tentang barang yang akan dijual pada *form* yang telah disediakan. Setelah informasi terisi dengan lengkap, kemudian klik *simpan*. Barang berhasil ditambahkan dalam toko.

4.9 Antarmuka Halaman List *Edit* Barang Toko



Gambar 4.16 Rancangan Antarmuka List Barang Toko

Rancangan antarmuka pada gambar 4.16 menunjukkan halaman untuk list barang toko yang akan di *edit*. Pilih dahulu barang toko yang akan di *edit*, kemudian *edit* informasi atau tambah foto barang.

4.10 Antarmuka Halaman Profil *Member*



Gambar 4.17 Rancangan Antarmuka Profil *Member*

Rancangan antarmuka pada gambar 4.17 menunjukkan halaman profil member yang berisi data diri serta detail toko dari member.

4.11 Antarmuka Halaman Edit Profil *Member*



Gambar 4.18 Rancangan Antarmuka *Edit Profil Diri Member*



Gambar 4.19 Rancangan Antarmuka *Edit Alamat Member*



Gambar 4.20 Rancangan Antarmuka *Edit Akun Fasapay* oleh *Member*



Gambar 4.21 Rancangan Antarmuka *Edit Password Member*

Edit profil terdiri dari 4 macam informasi yang akan di *edit* yaitu Profil diri, alamat, akun fasapay, dan password. Untuk masuk ke dalam menu edit, pilih tombol pengaturan saat halaman *view profile* dibuka.

Pada rancangan antarmuka pada gambar 4.18 menunjukkan halaman untuk *edit* data diri yang terdiri dari informasi serta foto. Kemudian untuk gambar 4.19 menunjukkan halaman *edit* alamat. Pada gambar 4.20 menunjukkan halaman untuk *edit* akun *Fasapay*. Dan yang terakhir adalah gambar 4.21 yang menunjukkan halaman *edit password*. Pada tiap halaman *edit* yang telah dijelaskan di atas, jika sudah selesai *mengedit* untuk menyimpan pilih tombol *simpan*, dan data sudah *terupdate* dengan data yang baru.

4.12 Antarmuka Halaman Keranjang Belanja



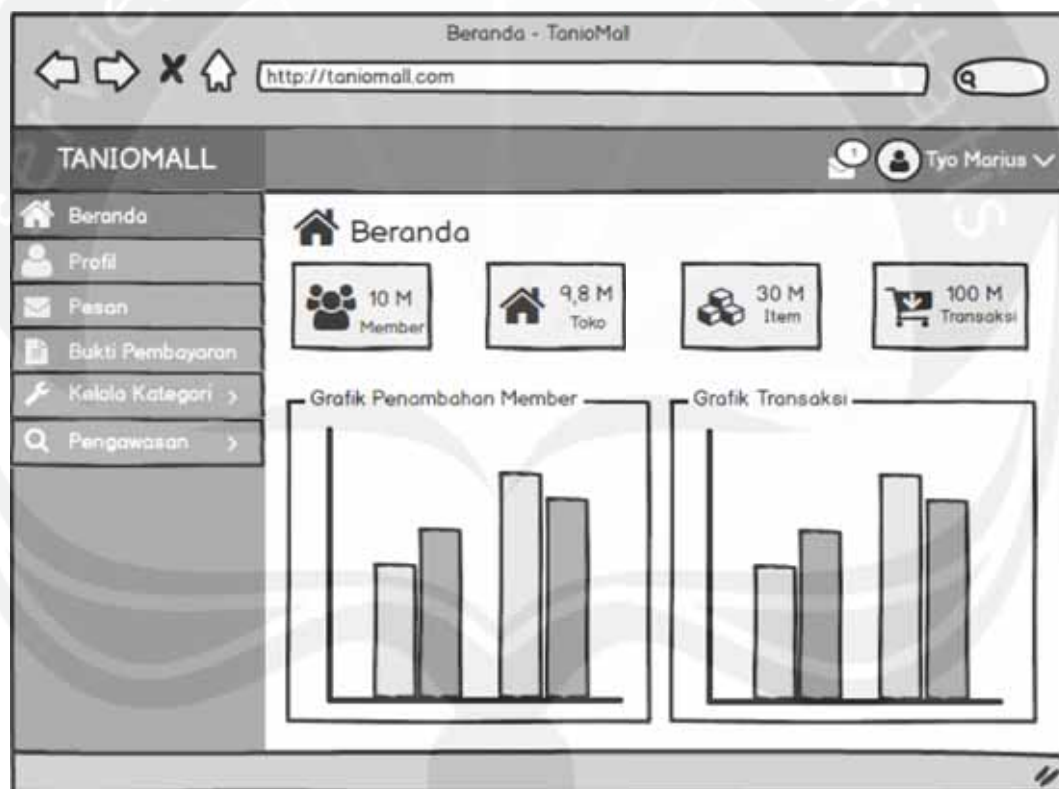
Gambar 4.22 Rancangan Antarmuka Keranjang Belanja

Rancangan antarmuka pada gambar 4.22 menunjukkan halaman untuk *edit* barang toko. Penjual dapat *mengedit* informasi maupun menambah foto barang yang akan dijual. Penjual harus *upload* minimal satu foto barang yang

akan dijual, kemudian mengisikan informasi tentang barang yang akan dijual pada *form* yang telah disediakan. Setelah informasi terisi dengan lengkap, kemudian klik simpan. Barang berhasil ditambahkan dalam toko.

Sedangkan untuk rancangan antarmuka perangkat lunak TanioMall untuk aplikasi *web*, sebagai berikut:

4.13 Antarmuka Halaman Beranda *Web*



Gambar 4.23 Rancangan Antarmuka Halaman Beranda *Web*

Pada gambar 4.23 menunjukkan rancangan antar muka halaman beranda pada aplikasi *web*. Pada halaman ini admin dapat melihat jumlah member, jumlah toko, jumlah item, jumlah transaksi, grafik penambahan member per bulan, dan grafik jumlah transaksi yang terjadi per bulan secara *real time*.

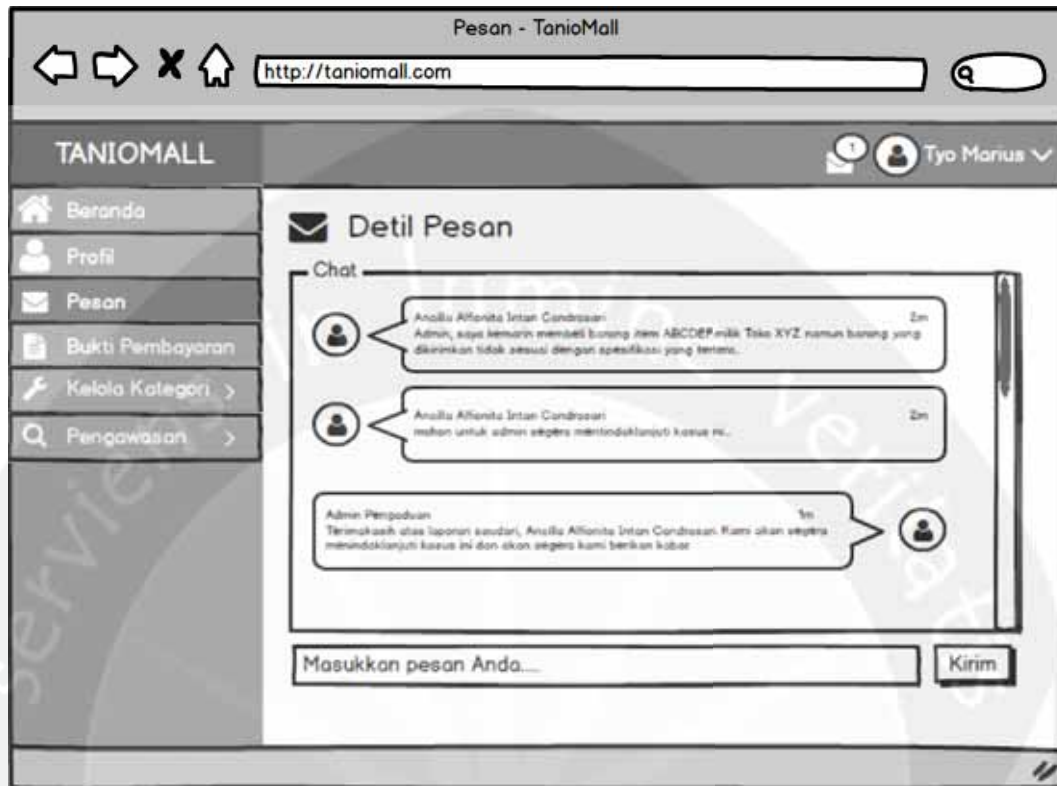
4.14 Antarmuka Halaman Pesan



Gambar 4.24 Rancangan Antarmuka Halaman Pesan

Pada gambar 4.24 menunjukkan rancangan antarmuka halaman pesan, pada halaman ini admin dapat melihat daftar pesan atau laporan-laporan dari *member* tentang ketidaknyamanan *member* dalam bertransaksi, seperti barang yang dikirim oleh penjual tidak sesuai dengan yang ditampilkan pada halaman detail barang, atau terjadi keganjalan pada transaksi barang yang dilakukan oleh *member*.

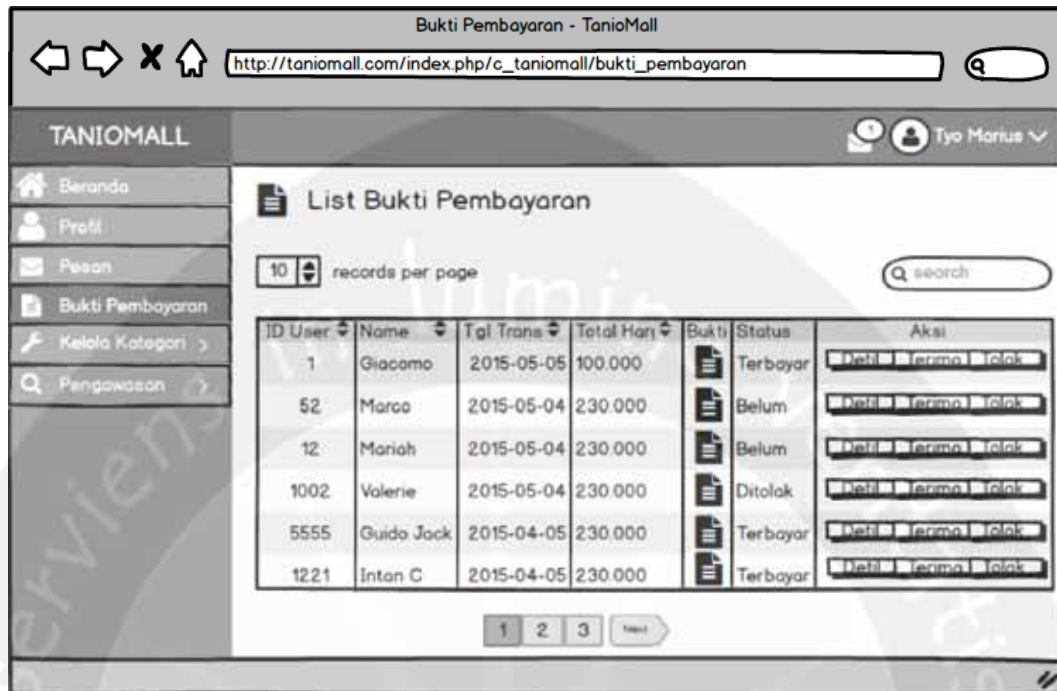
4.15 Antarmuka Halaman Detil Pesan



Gambar 4.25 Rancangan Antarmuka Halaman Detil Pesan

Pada gambar 4.25 menunjukkan rancangan antarmuka halaman detil pesan. Pada halaman ini admin dapat melihat detil pesan dan dapat saling berinteraksi atau *chatting* dengan *member* yang memiliki keluhan mengenai transaksi yang dilakukan oleh *member*.

4.16 Antarmuka Halaman Bukti Pembayaran



Gambar 4.26 Rancangan Antarmuka Halaman Bukti Pembayaran

Pada gambar 4.26 menunjukkan rancangan antarmuka halaman daftar bukti pembayaran. Pada halaman ini admin dapat melihat daftar bukti-bukti pembayaran yang telah di-upload oleh member dan dapat melihat detail bukti pembayaran, selain itu admin juga dapat menerima atau menolak bukti pembayaran tersebut.

4.17 Antarmuka Halaman Pengawasan Barang



Gambar 4.27 Antarmuka Halaman Pengawasan Barang

Pada gambar 4.27 menunjukkan rancangan antarmuka halaman pengawasan barang. Pada halaman ini admin dapat melihat daftar barang-barang yang dimiliki oleh toko-toko yang terdaftar dan dapat melihat detail barang dan komentar atau ulasan yang bersifat negatif maupun positif sehingga admin dapat memberikan suatu peringatan terhadap toko apabila memang suatu barang memiliki keganjalan seperti banyak pembeli yang kecewa terhadap barang tersebut sesuai dengan ulasan-ulasan yang diberikan oleh *member* yang telah membeli barang tersebut.

SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

TanioMall

[Pembangunan Aplikasi *Mobile Marketplace*]

Untuk :

Tugas Akhir (Skripsi)

Dipersiapkan oleh:

Yohanes Marius Wahyu Dwi Prasetyo / 6510

Program Studi Teknik Informatika – Fakultas Teknologi
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi	Nomor Dokumen		Halaman
		SKPL - TanioMall		1/65
		REVISI		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

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

Daftar Halaman Perubahan

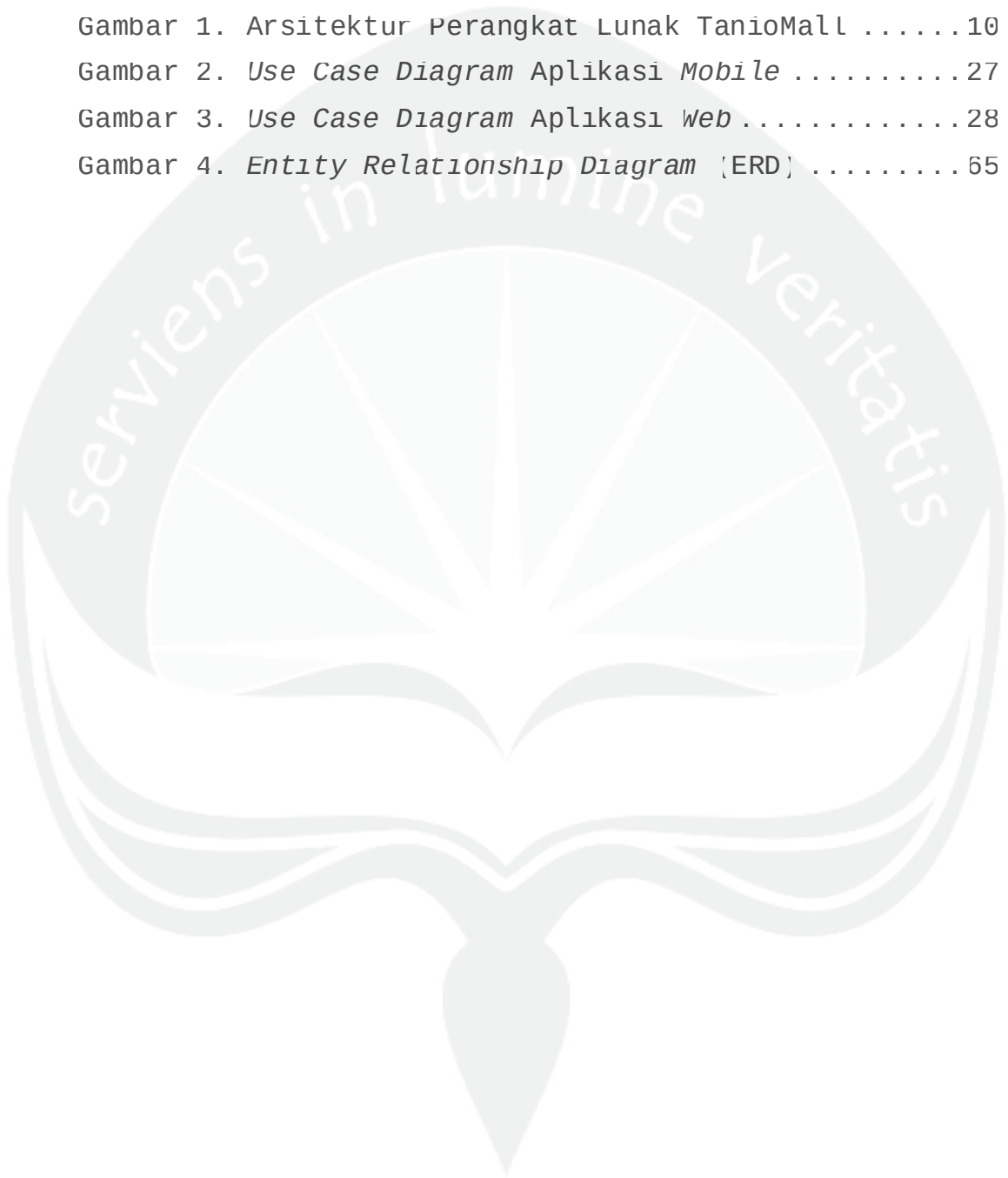
Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan.....	6
1.1	Tujuan.....	6
1.2	Lingkup Masalah.....	6
1.3	Definisi, Akronim dan Singkatan.....	7
1.4	Referensi.....	7
1.5	Deskripsi umum (Overview).....	8
2	Deskripsi Kebutuhan.....	8
2.1	Perspektif produk.....	8
2.2	Fungsi Produk.....	10
2.3	Karakteristik Pengguna.....	22
2.4	Batasan-batasan.....	23
2.5	Asumsi dan Ketergantungan.....	23
3	Kebutuhan Khusus.....	23
3.1	Kebutuhan antarmuka eksternal.....	23
3.2	Kebutuhan fungsionalitas Perangkat Lunak.....	27
4	Spesifikasi Rinci Kebutuhan.....	28
4.1	Spesifikasi Kebutuhan Fungsionalitas.....	28
5	<i>Entity Relationship Diagram [ERD]</i>	65

Daftar Gambar

Gambar 1. Arsitektur Perangkat Lunak TanioMall	10
Gambar 2. <i>Use Case Diagram</i> Aplikasi <i>Mobile</i>	27
Gambar 3. <i>Use Case Diagram</i> Aplikasi <i>Web</i>	28
Gambar 4. <i>Entity Relationship Diagram</i> [ERD]	65



1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak TanioMall (Aplikasi *Mobile Marketplace*) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-TanioMall ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat Lunak TanioMall dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan pengguna.
2. Menangani pengelolaan toko.
3. Menangani pengelolaan transaksi penjualan.
4. Menangani pengelolaan transaksi pembayaran.
5. Menangani pengelolaan kategori dan sub kategori produk oleh administrator.
6. Menangani pengelolaan produk.
7. Menangani pengelolaan negosiasi harga produk.
8. Menangani pengelolaan testimonial.
9. Menangani proses komunikasi antar pengguna dengan menggunakan fitur *chatting*.

Perangkat lunak ini berjalan pada semua lingkungan *platform* dengan *browser* yang mendukung penggunaan *javascript* (*web*) dan *smartphone* dengan *platform Android*.

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-TanioMall-XXX	Kode yang merepresentasikan kebutuhan pada TanioMall di mana XXX merupakan nomor fungsi produk.
TanioMall	Aplikasi <i>mobile marketplace</i> .
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Database	Merupakan tempat penyimpanan data.
Hapus semu	Sebenarnya tidak menghapus data, hanya membuat data tidak ditampilkan.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Stefanus Diptya Adeshya Anindito, *Pembangunan Aplikasi Android untuk Layanan Pesan Antar*

Barang atau Jasa Berbasis Lokasi dan Teknologi Push Notification. 2012.

2. Septiana Rahayu Dewi, *Pembangunan Online Marketplace untuk Pengusaha Mikro di PASTY. 2014.*

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 TanioMall 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 TanioMall tersebut.

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

2 Deskripsi Kebutuhan

2.1 Perspektif produk

TanioMall merupakan perangkat lunak *mobile* yang dikembangkan untuk membantu pengguna dalam melakukan penjualan barang atau produk. Sistem ini menangani

penjualan produk, pemesanan produk, penanganan produk, pengelolaan toko, dan pengelolaan sistem oleh administrator TanioMall. Sistem ini juga menangani pembayaran pemesanan produk dengan menggunakan fitur API FasaPay. Sistem ini sendiri terbagi menjadi dua jenis, aplikasi *mobile* dan aplikasi *web*.

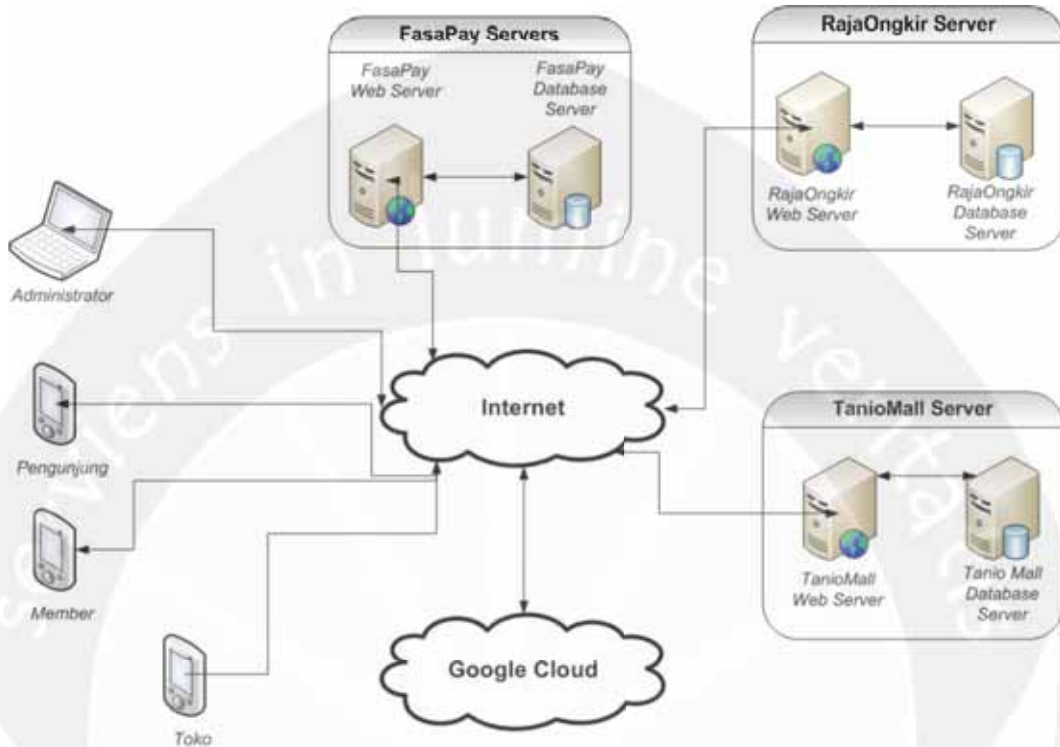
Pada aplikasi *mobile*-nya, perangkat lunak ini berjalan pada *platform Android* yang dibuat menggunakan bahasa pemrograman Java dengan mengacu pada MySQL sebagai DBMS (*Database Management System*), dengan Eclipse Juno sebagai lingkungan pemrogramannya.

Untuk aplikasi web, perangkat lunak TanioMall ini berjalan pada *framework CodeIgniter* yang dibuat menggunakan bahasa pemrograman PHP dengan mengacu pada MySQL sebagai DBMS (*Database Management System*), dengan Sublime Text 3.0 dan Notepad++ sebagai lingkungan pemrogramannya.

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (*Graphical User Interface*). Pada sistem ini, seperti pada gambar 1, arsitektur perangkat lunak yang digunakan berupa *client server*, di mana semua data disimpan di *server*.

Seluruh data yang berhubungan dengan sistem ini disimpan dalam *database*, sehingga untuk menghubungkan aplikasi *mobile* dengan *database* diperlukan sebuah API (*Application Programming Interface*) agar data hanya dapat diakses dengan autentifikasi yang khusus. Selain itu, sistem ini juga melibatkan sistem lainnya seperti FasaPay yang digunakan untuk transaksi pembayaran, dan RajaOngkir yang digunakan untuk mendapatkan tarif jasa pengiriman. Dalam melibatkan sistem lain tersebut

digunakan API untuk menghubungkan sistem FasaPay dan RajaOngkir dengan sistem TanioMall.



Gambar 1. Arsitektur Perangkat Lunak TanioMall

2.2 Fungsi Produk

Fungsi produk perangkat lunak TanioMall adalah sebagai berikut :

2.2.1 Aplikasi Mobile

1. Fungsi *Mendaftar Akun Member* [SKPL-TanioMall-001] .

Merupakan fungsi yang digunakan untuk menambah data pengguna.

2. Fungsi *Tampil Produk* [SKPL-TanioMall-002] .

Merupakan fungsi yang digunakan oleh semua pengunjung, member, atau toko untuk melihat data produk.

3. Fungsi *Cari Produk* (**SKPL-TanioMall-003**).

Merupakan fungsi yang digunakan oleh semua pengunjung, member, atau toko untuk mencari data produk.

4. Fungsi *Tampil Toko* (**SKPL-TanioMall-004**).

Merupakan fungsi yang digunakan oleh semua pengunjung, member, atau toko untuk melihat data toko.

5. Fungsi *Cari Toko* (**SKPL-TanioMall-005**).

Merupakan fungsi yang digunakan oleh semua pengunjung, member, atau toko untuk mencari data toko.

6. Fungsi *Pengelolaan Data Transaksi Pemesanan Produk* (**SKPL-TanioMall-006**).

Merupakan fungsi yang digunakan oleh semua *member* atau toko dalam melakukan pemesanan suatu produk yang dijual oleh suatu toko.

Fungsi *Pengelolaan Data Transaksi Pemesanan Produk* meliputi:

a. Fungsi *Entri Data Transaksi Pemesanan Produk* (**SKPL-TanioMall-006-01**).

Merupakan fungsi yang digunakan untuk penambahan data transaksi pemesanan produk.

b. Fungsi *Edit Data Transaksi Pemesanan Produk* (**SKPL-TanioMall-006-02**).

Merupakan fungsi yang digunakan untuk mengubah/ mengganti data transaksi pemesanan produk.

c. Fungsi *Hapus Data Transaksi Pemesanan Produk* (**SKPL-TanioMall-006-03**).

Merupakan fungsi yang digunakan untuk menghapus data pemesanan produk.

d. Fungsi *Display Data Pemesanan Negosiasi Produk* (**SKPL-TanioMall-006-04**).

Merupakan fungsi yang digunakan untuk menampilkan data transaksi pemesanan produk.

e. Fungsi *Cari Data Transaksi Pemesanan Produk* (**SKPL-TanioMall-006-05**).

Merupakan fungsi yang digunakan untuk mencari data transaksi pemesanan produk berdasarkan nama produk.

f. Fungsi *Mengupdate Status Transaksi Pemesanan Produk* (**SKPL-TanioMall-006-06**).

Merupakan fungsi yang digunakan untuk memperbarui status pemesanan.

7. Fungsi *Pengelolaan Data Detail Transaksi Pemesanan Produk* (**SKPL-TanioMall-007**).

Merupakan fungsi yang digunakan oleh semua *member* untuk mengelola data detail pemesanan produk.

Fungsi *Pengelolaan Data Transaksi Pemesanan Produk* meliputi:

a. Fungsi *Entri Data Detail Transaksi Pemesanan Produk* (**SKPL-TanioMall-007-01**).

Merupakan fungsi yang digunakan untuk menambahkan data detail transaksi pemesanan produk.

b. Fungsi *Edit Data Transaksi Pemesanan Produk* (**SKPL-TanioMall-007-02**).

Merupakan fungsi yang digunakan untuk mengubah/ mengganti data detail transaksi pemesanan produk.

c. Fungsi *Display Data Detail Transaksi Pemesanan Produk* (**SKPL-TanioMall-007-03**).

Merupakan fungsi yang digunakan untuk menampilkan data detail transaksi pemesanan produk.

d. Fungsi *Cari Data Detail Transaksi Pemesanan Produk* (**SKPL-TanioMall-007-04**).

Merupakan fungsi yang digunakan untuk mencari data detail transaksi pemesanan produk berdasarkan nomor plat mobil.

8. Fungsi *Pengelolaan Transaksi Pembayaran* (**SKPL-TanioMall-008**).

Merupakan fungsi yang dipergunakan oleh semua *member* untuk mengelola data pembayaran yang dilakukan.

9. Fungsi *Login* (**SKPL-TanioMall-009**).

Merupakan fungsi yang digunakan oleh *member* dan toko untuk dapat masuk ke dalam sistem dan memperoleh hak akses sesuai dengan *role* yang dimiliki pengguna.

10. Fungsi *Edit Password* (**SKPL-TanioMall-010**).

Merupakan fungsi yang digunakan untuk mengubah/mengganti password dari pengguna.

11. Fungsi *Pengelolaan Data Member* (**SKPL-TanioMall-011**).

Merupakan fungsi yang digunakan semua *member* untuk mengelola data member.

Fungsi *Pengelolaan Data Member* mencakup :

a. Fungsi *Edit Data Pengguna* (**SKPL-TanioMall-011-01**).

Merupakan fungsi yang digunakan untuk mengubah data pengguna.

b. Fungsi *Display Data Pengguna* (**SKPL-TanioMall-011-02**).

Merupakan fungsi yang digunakan untuk menampilkan data pengguna.

c. Fungsi *Cari Data Pengguna* (**SKPL-TanioMall-011-03**).

Merupakan fungsi yang digunakan untuk mencari data toko berdasarkan nama.

12. Fungsi *Mendattar Toko* (**SKPL-TanioMall-012**).

Merupakan fungsi yang digunakan oleh *member* untuk membuat sebuah toko baru.

13. Fungsi *Pengelolaan Data Transaksi Negosiasi Produk* (**SKPL-TanioMall-013**).

Merupakan fungsi yang digunakan oleh semua *member* dalam melakukan negosiasi harga suatu produk yang dijual oleh suatu toko.

Fungsi *Pengelolaan Data Transaksi Negosiasi Produk* meliputi:

a. Fungsi *Entri Data Transaksi Negosiasi Produk* (**SKPL-TanioMall-013-01**).

Merupakan fungsi yang digunakan untuk penambahan data transaksi negosiasi produk.

b. Fungsi *Edit Data Transaksi Negosiasi Produk* (**SKPL-TanioMall-013-02**).

Merupakan fungsi yang digunakan untuk mengubah/mengganti data transaksi negosiasi produk.

c. Fungsi *Display Data Transaksi Negosiasi Produk* (**SKPL-TanioMall-013-03**).

Merupakan fungsi yang digunakan untuk menampilkan data transaksi negosiasi produk.

d. Fungsi *Cari Data Transaksi Negosiasi Produk* (**SKPL-TanioMall-013-04**).

Merupakan fungsi yang digunakan untuk mencari data transaksi negosiasi produk berdasarkan nama produk.

14. Fungsi *Pengelolaan Data Testimonial* (**SKPL-TanioMall-014**).

Merupakan fungsi yang digunakan oleh semua *member* atau toko untuk mengelola data testimonial.

Fungsi *Pengelolaan Data Testimonial* meliputi:

a. Fungsi *Entri Data Testimonial* (**SKPL-TanioMall-014-01**).

Merupakan fungsi yang digunakan untuk menambah data testimonial.

b. Fungsi *Edit Data Testimonial* (SKPL-TanioMall-014-02).

Merupakan fungsi yang digunakan untuk mengganti/mengubah data testimonial.

c. Fungsi *Hapus Data Testimonial* (SKPL-TanioMall-014-03).

Merupakan fungsi yang digunakan untuk menghapus semua data testimonial.

d. Fungsi *Display Data Testimonial* (SKPL-TanioMall-014-04).

Merupakan fungsi yang digunakan untuk menampilkan data testimonial.

15. Fungsi *Pengelolaan Data Favorit* (SKPL-TanioMall-015).

Merupakan fungsi yang digunakan oleh semua *member* atau toko untuk mengelola data favorit produk.

Fungsi *Pengelolaan Data Favorit* meliputi:

a. Fungsi *Entri Data Favorit* (SKPL-TanioMall-015-01).

Merupakan fungsi yang digunakan untuk menambah data favorit produk.

b. Fungsi *Hapus Data Favorit* (SKPL-TanioMall-015-02).

Merupakan fungsi yang digunakan untuk menghapus data favorit produk.

c. Fungsi *Display Data Favorit* (**SKPL-TanioMall-015-03**).

Merupakan fungsi yang digunakan untuk menampilkan data favorit produk.

d. Fungsi *Cari Data Favorit* (**SKPL-TanioMall-015-04**).

Merupakan fungsi yang digunakan untuk mencari data favorit produk berdasarkan nama produk.

16. Fungsi *Pengelolaan Data Pesan* (**SKPL-TanioMall-016**).

Merupakan fungsi yang digunakan oleh semua *member* atau toko untuk mengelola data pesan.

Fungsi *Pengelolaan Data Pesan* meliputi:

a. Fungsi *Entri Data Pesan* (**SKPL-TanioMall-016-01**).

Merupakan fungsi yang digunakan untuk menambah data pesan.

b. Fungsi *Hapus Data Pesan* (**SKPL-TanioMall-016-02**).

Merupakan fungsi yang digunakan untuk menghapus data pesan.

c. Fungsi *Display Data Pesan* (**SKPL-TanioMall-016-03**).

Merupakan fungsi yang digunakan untuk menampilkan data pesan.

d. Fungsi *Cari Data Pesan* (**SKPL-TanioMall-016-04**).

Merupakan fungsi yang digunakan untuk mencari data pesan berdasarkan nama *member* pengirim.

17. Fungsi *Pengelolaan Data Toko* (SKPL-TanioMall-017).

Merupakan fungsi yang digunakan semua *member* untuk mengelola data toko.

Fungsi *Pengelolaan Data Toko* mencakup :

a. Fungsi *Edit Data Toko* (SKPL-TanioMall-017-01).

Merupakan fungsi yang digunakan untuk mengubah data toko.

b. Fungsi *Hapus Data Toko* (SKPL-TanioMall-017-02).

Merupakan fungsi yang digunakan untuk menghapus semua data toko.

c. Fungsi *Display Data Toko* (SKPL-TanioMall-017-03).

Merupakan fungsi yang digunakan untuk menampilkan data toko.

d. Fungsi *Cari Data Toko* (SKPL-TanioMall-017-04).

Merupakan fungsi yang digunakan untuk mencari data toko berdasarkan nama.

18. Fungsi *Pengelolaan Data Produk* (SKPL-TanioMall-018).

Merupakan fungsi yang digunakan semua toko untuk mengelola data produk toko.

Fungsi *Pengelolaan Data Produk* meliputi:

a. Fungsi Generate ID Produk (**SKPL-TanioMall-018-01**).

Merupakan fungsi yang digunakan untuk men-generate ID Produk agar unik.

b. Fungsi *Entri Data Produk* (**SKPL-TanioMall-018-02**).

Merupakan fungsi yang digunakan untuk menambah data produk toko.

c. Fungsi *Edit Data Produk* (**SKPL-TanioMall-018-03**).

Merupakan fungsi untuk mengubah atau mengganti data produk toko.

d. Fungsi *Hapus Data Produk* (**SKPL-TanioMall-018-04**).

Merupakan fungsi untuk menghapus semua data produk toko.

e. Fungsi *Display Data Produk* (**SKPL-TanioMall-018-05**).

merupakan fungsi yang digunakan untuk menampilkan data produk toko yang ada.

f. Fungsi *Cari Data Produk* (**SKPL-TanioMall-018-06**).

Merupakan fungsi yang digunakan untuk mencari data produk berdasarkan nama produk pengirim.

19. Fungsi *Pengelolaan Data Transaksi Penjualan* (**SKPL-TanioMall-019**).

Merupakan fungsi yang digunakan semua toko untuk mengelola data transaksi penjualan.

Fungsi *Pengelolaan Data Produk* meliputi:

a. Fungsi *Edit Data Transaksi Penjualan* (**SKPL-TanioMall-019-01**).

Merupakan fungsi untuk mengubah atau mengganti data transaksi penjualan.

b. Fungsi *Hapus Data Transaksi Penjualan* (**SKPL-TanioMall-019-02**).

Merupakan fungsi untuk menghapus semua data transaksi penjualan.

c. Fungsi *Display Data Transaksi Penjualan* (**SKPL-TanioMall-019-03**).

Merupakan fungsi yang digunakan untuk menampilkan data transaksi penjualan yang ada.

d. Fungsi *Cari Data Transaksi Penjualan* (**SKPL-TanioMall-019-04**).

Merupakan fungsi yang digunakan untuk mencari data transaksi penjualan.

2.2.2 Aplikasi Web

20. Fungsi *Login Web* (**SKPL-TanioMall-020**).

Merupakan fungsi yang digunakan oleh administrator untuk masuk ke dalam *web*.

21. Fungsi *Pengelolaan Data Admin* (**SKPL-TanioMall-021**).

Merupakan fungsi yang digunakan admin untuk mengelola data admin.

Fungsi *Pengelolaan Data Admin* mencakup :

a. Fungsi *Edit Data Admin* (**SKPL-TanioMall-021-01**).

Merupakan fungsi yang digunakan untuk mengubah data admin.

b. Fungsi *Display Data Admin* (**SKPL-TanioMall-021-02**).

Merupakan fungsi yang digunakan untuk menampilkan data admin.

22. Fungsi *Pengelolaan Data Kategori* (**SKPL-TanioMall-022**).

Merupakan fungsi yang digunakan oleh administrator untuk mengelola data kategori produk.

Fungsi *Pengelolaan Data Kategori* meliputi:

a. Fungsi *Entri Data Kategori* (**SKPL-TanioMall-022-01**).

Merupakan fungsi yang digunakan untuk menambah data kategori produk.

b. Fungsi *Edit Data Kategori* (**SKPL-TanioMall-022-02**).

Merupakan fungsi yang digunakan untuk mengganti/mengubah data kategori produk.

c. Fungsi *Hapus Data Kategori* (**SKPL-TanioMall-022-03**).

Merupakan fungsi yang digunakan untuk menghapus semua data kategori produk.

d. Fungsi *Display Data Kategori* (**SKPL-TanioMall-022-04**).

Merupakan fungsi yang digunakan untuk menampilkan data kategori produk.

e. Fungsi *Cari Data Kategori* (SKPL-TanioMall-022-05).

Merupakan fungsi yang digunakan untuk mencari data kategori produk berdasarkan nama kategori.

23. Fungsi *Pengawasan Data Produk* (SKPL-TanioMall-023).

Merupakan fungsi yang digunakan semua admin untuk mengawasi data produk.

24. Fungsi *Pengawasan Data Transaksi Toko* (SKPL-TanioMall-024).

Merupakan fungsi yang digunakan semua *member* untuk mengelola data toko.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak TanioMall adalah sebagai berikut :

1. Memahami pengoperasian komputer.
2. Mengerti pengoperasian *smartphone platform Android*.
3. Memahami penggunaan perangkat lunak TanioMall.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak TanioMall tersebut adalah :

1. Kebijakan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak TanioMall.

2. Keterbatasan perangkat keras

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

2.5 Asumsi dan Ketergantungan

Aplikasi *mobile* TanioMall dapat dijalankan pada perangkat *smartphone* yang menggunakan sistem operasi Android, dan untuk *web application* dapat berjalan disemua lingkungan perangkat lunak dengan menggunakan *browser* yang mendukung *javascript*.

3 Kebutuhan Khusus

3.1 Kebutuhan antarmuka eksternal

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

3.1.1 Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk halaman-halaman *web* maupun *activity-activity* pada aplikasi *moile*.

Antarmuka pada halaman *web* dapat berisi *form-form*, tabel, dan grafik, sedangkan pada aplikasi *mobile* dapat berisi *list view*, *grid view*, dan juga *form-form*.

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak TanioMall adalah:

1. Perangkat dekstop.

Komponen	Minimal	Dianjurkan
Processor	Pentium III-class, 1GHz	Pentium 4 1,7 GHz
RAM	256 MB	512 MB
Hard Disk	1 GB	10 GB
Monitor	800x600 256 colors High Color 16-bit	1024x 768 256 colors High Color 32-bit
Mouse	Mouse Microsoft atau alat penunjuk yang compatible dengan OS	Mouse Microsoft atau alat penunjuk yang compatible dengan OS
Keyboard	Keyboard Microsoft atau alat pengetik yang compatible dengan OS	Keyboard Microsoft atau alat pengetik yang compatible dengan OS
Printer	Printer yang Compatible dengan OS	Printer yang Compatible dengan OS

2. Perangkat Database Server.

Komponen	Minimal	Dianjurkan
Memory	512 MB	1GB

Processor Speed	X86 1.0 GHZ x64 1.4 GHZ	2.0 GHZ
Processor type	x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support x86 Processor: Pentium III-compatible	Intel Pentium IV with EM64T support
RAM	2GB	4GB

3. Perangkat Web Server.

CPU	2 core (3GHz or higher)	
RAM	4GB	
Disks	Array 1:	
	OS, software, and websites	3 x 146GB, SCSI, RAID 5 (hardware impl.)
Disk Partitioning	Array 1:	
	C: 40GB	for OS and software
	D: remaining space [250GB]	for customer data
NICS	FrontNet, BackNet	

4. Perangkat *mobile*, *smartphone* berbasis Android.

3.1.3 Antarmuka perangkat lunak

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

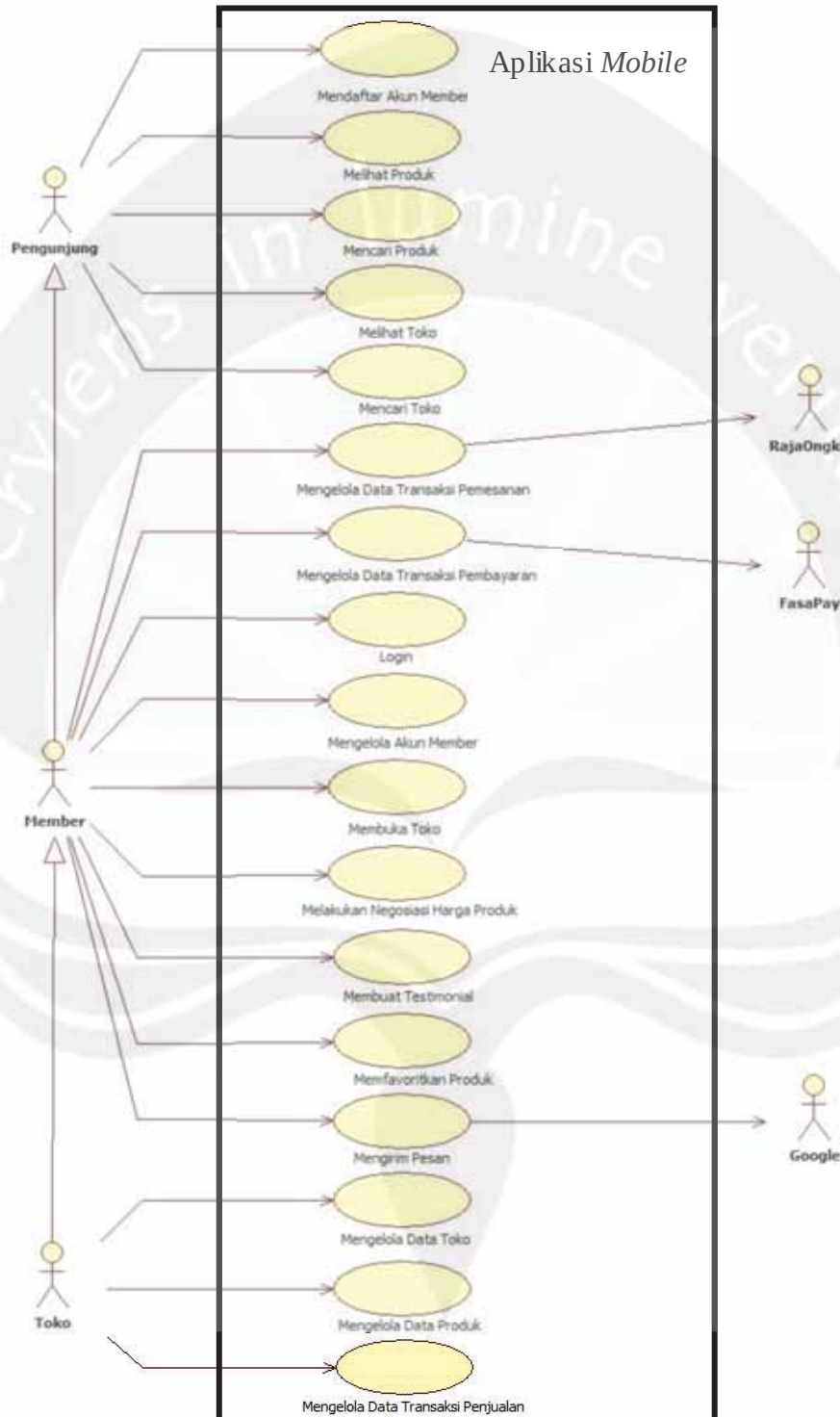
1. Nama : Android
Sumber : Google
Sebagai *platform environment* pada perangkat *mobile*.
2. Nama : MySQL
Sumber : *Open Source*
Sebagai *database management system* (DBMS) yang digunakan untuk penyimpanan data di sisi *server*.
3. Nama : IE/Firefox/Chrome/Opera/dll
Sumber : berbagai sumber
Sebagai aplikasi *web browser* untuk membuka sistem *web*.
4. Nama : Apache
Sumber : *Open Source*
Sebagai *web server*.

3.1.4 Antarmuka Komunikasi

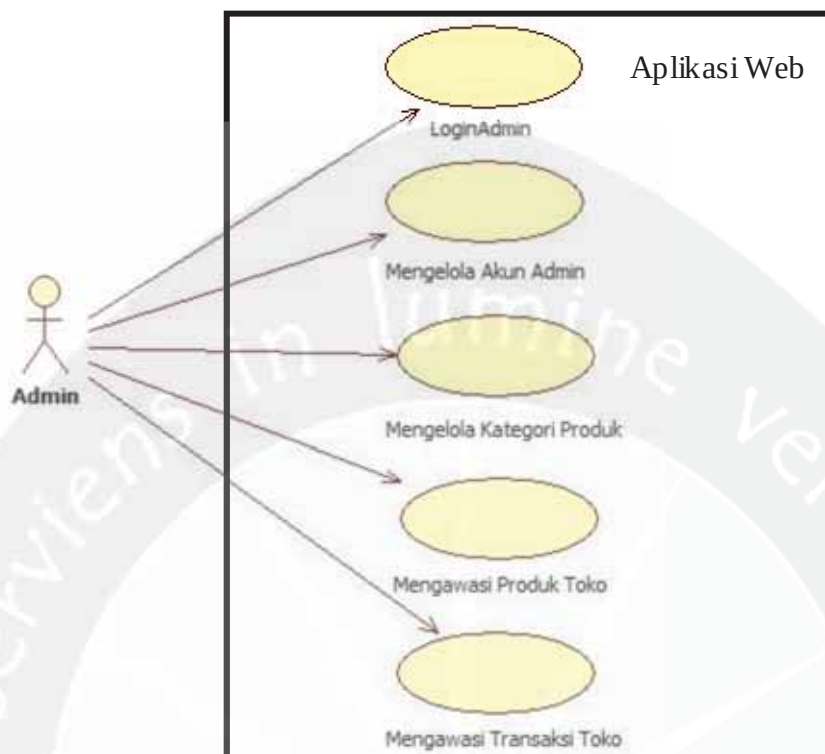
Antarmuka komunikasi perangkat lunak TanioMall menggunakan menggunakan *protocol* HTTP.

3.2 Kebutuhan Fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram



Gambar 2. Use Case Diagram Aplikasi Mobile



Gambar 3. Use Case Diagram Aplikasi Web

4 Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

4.1.1 Use case Specification : *Mendattar Akun Member*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mendaftar ke aplikasi.

2. Primary Actor

Pengunjung

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pendaftaran.
2. Sistem menampilkan *form* data pendaftaran dan meminta nama, *email*, alamat, nomor telepon, *username*, *password*, dan konfirmasi *password*.
3. Aktor memasukkan nama, *email*, alamat, nomor telepon, *username*, *password*, dan konfirmasi *password*.
4. Aktor mengklik tombol Daftar untuk menyimpan data pengguna yang telah diinputkan.
5. Sistem memeriksa nama, *email*, alamat, nomor telepon, *username*, *password*, dan konfirmasi *password* yang diinputkan oleh aktor.
 - E-1 Username telah terdaftar
 - E-2 Email tidak sesuai format
 - E-3 Email telah terdaftar
 - E-4 Password tidak sama
6. Sistem menyimpan data pengguna ke *database*.
7. Sistem menampilkan informasi bahwa proses pendaftaran telah berhasil.
8. Use case selesai.

5. Alternative Flow

none

6. Error Flow

- E-1 Username telah terdaftar
 1. Sistem menampilkan peringatan bahwa username diinputkan telah terdaftar.
 2. Kembali ke Basic Flow langkah ke 3.
- E-2 Email tidak sesuai format

1. Sistem memberikan pesan peringatan bahwa email yang diinputkan tidak sesuai format.
2. Kembali ke Basic Flow langkah ke 3.

E-3 Email telah terdaftar

1. Sistem memberikan pesan peringatan bahwa email yang diinputkan telah terdaftar.
2. Kembali ke Basic Flow langkah ke 3.

E-4 Password tidak sama

1. Sistem memberikan pesan peringatan bahwa password yang diinputkan tidak sesuai dengan password konfirmasi.
2. Kembali ke Basic Flow langkah ke 3.

7. PreConditions

none

8. PostConditions

Aktor telah terdaftar ke dalam sistem.

4.1.2 Use case Spesification : *Melihat Produk*

1. Brief Description

Use Case ini digunakan oleh aktor untuk melihat produk.

2. Primary Actor

1. *Member*
2. Pengunjung
3. Toko

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melihat produk.

Program Studi Teknik Informatika	SKPL – TaniaMall	30/65
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem menampilkan produk.
3. Use case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

Aktor telah memasuki sistem

8. PostConditions

Aktor dapat melihat produk yang dimiliki sistem.

4.1.3 Use case Spesification : Mencari Produk

1. Brief Description

Use Case ini digunakan oleh aktor untuk mencari produk.

2. Primary Actor

1. *Member*
2. Pengunjung
3. Toko

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mencari produk.
 2. Sistem menampilkan *form* pencarian.
 3. Aktor memasukkan kata pencarian pada *form* pencarian.
 4. Sistem melakukan pencarian produk berdasarkan kata pencarian yang dimasukkan oleh aktor.
- E-1 Produk yang dicari tidak ditemukan

5. Sistem menampilkan produk yang dicari oleh aktor.

6. Use case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Produk yang dicari tidak ditemukan

1. Sistem menampilkan peringatan bahwa produk yang dicari tidak ditemukan.

2. Kembali ke Basic Flow langkah ke 2.

7. PreConditions

Aktor telah memasuki sistem

8. PostConditions

Aktor menemukan produk yang dicari.

4.1.4 Use case Spesification : *Melihat Toko*

1. Brief Description

Use Case ini digunakan oleh aktor untuk melihat toko.

2. Primary Actor

1. *Member*

2. Pengunjung

3. Toko

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melihat toko.

2. Sistem menampilkan toko.

3. Use case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

Aktor telah memasuki sistem

8. PostConditions

Aktor dapat melihat toko yang dimiliki sistem.

4.1.5 Use case Specification : Mencari Toko

1. Brief Description

Use Case ini digunakan oleh aktor untuk mencari toko.

2. Primary Actor

1. Member
2. Pengunjung
3. Toko

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mencari toko.
2. Sistem menampilkan *form* pencarian.
3. Aktor memasukkan kata pencarian pada *form* pencarian.
4. Sistem melakukan pencarian toko berdasarkan kata pencarian yang dimasukkan oleh aktor.
E-1 Toko yang dicari tidak ditemukan
5. Sistem menampilkan produk yang dicari oleh aktor.
6. Use case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Produk yang dicari tidak ditemukan

1. Sistem menampilkan peringatan bahwa toko yang dicari tidak ditemukan.
2. Kembali ke Basic Flow langkah ke 2.

7. PreConditions

Aktor telah memasuki sistem

8. PostConditions

Aktor menemukan toko yang dicari.

4.1.6 Use case Spesification : *Mengelola Data Transaksi Pemesanan*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola atau memesan suatu produk toko.

2. Primary Actor

Member

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data transaksi pemesanan.
2. Sistem menampilkan produk-produk yang dimiliki oleh berbagai toko, menampilkan antarmuka untuk batal pemesanan suatu produk.
3. Aktor memilih suatu produk yang akan dipesan.
A-1 Aktor membatalkan pemesanan suatu produk.
A-2 Aktor mengubah data pemesanan suatu produk.

4. Sistem menampilkan data detail produk yang dipilih.
5. Aktor memasukkan jumlah pesanan.
6. Sistem mengecek jumlah pesanan dengan stok yang tersedia.

E-1 Stok tidak mencukupi

7. Sistem memberikan informasi bahwa stok mencukupi dan meminta konfirmasi untuk pemesanan.
8. Aktor memberikan konfirmasi pemesanan dengan mengklik tombol Pesan.
9. Sistem menyimpan data transaksi pemesanan satu produk ke database
10. Use case selesai

5. Alternative Flow

A-1 Aktor membatalkan pesanan suatu produk

1. Sistem menampilkan data detail transaksi pemesanan sementara.
2. Aktor memilih produk yang akan dihapus dari daftar pemesanan sementara.
3. Sistem meminta konfirmasi penghapusan data pemesanan suatu produk.
4. Aktor mengkonfirmasi penghapusan data pemesanan suatu produk
5. Sistem menghapus data detail transaksi pemesanan suatu produk dari database.
6. Berlanjut ke Basic Flow langkah ke 10

A-2 Aktor mengubah data detail pemesanan suatu produk.

1. Sistem menampilkan data detail transaksi pemesanan sementara.
2. Aktor memilih untuk mengubah data detail transaksi pemesanan suatu produk.
3. Sistem menampilkan detail pemesanan suatu produk.
4. Aktor mengubah data detail pemesanan suatu produk.
5. Sistem mengecek jumlah pesanan dengan stok yang tersedia.
E-1 Stok tidak mencukupi
6. Sistem meminta konfirmasi pengubahan data detail pemesanan suatu produk.
7. Aktor mengkonfirmasi ingin mengubah data detail transaksi pemesanan suatu produk
6. Sistem menyimpan data detail transaksi pemesanan yang diubah ke *database*
7. Berlanjut ke Basic Flow langkah 10

6. Error Flow

E-1 Stok tidak mencukupi

1. Sistem menampilkan pesan peringatan bahwa jumlah stok tidak mencukupi.
2. Kembali ke Basic Flow Langkah ke 5

7. PreConditions

1. Use case login telah dilakukan
2. Aktor telah memasuki system

8. PostConditions

Aktor telah berhasil memesan suatu produk dan transaksi pemesanan telah tercatat pada *database*.

Program Studi Teknik Informatika	SKPL – Taniomall	36/65
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika- UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4.1.7 Use case Spesification : Mengelola Data Transaksi Pembayaran

1. Brief Description

Use Case ini digunakan aktor untuk mengelola data transaksi pembayaran.

2. Primary Actor

Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengelola data transaksi pembayaran.
2. Sistem menampilkan daftar data transaksi pemesanan.
3. Aktor memilih suatu data transaksi pemesanan untuk dibayarkan.
4. Sistem menampilkan data transaksi pembayaran suatu transaksi pemesanan.
5. Aktor menekan tombol Bayar untuk melakukan pembayaran suatu transaksi pemesanan.
6. Sistem menampilkan form transaksi pembayaran.
7. Aktor memasukkan data akun pembayaran.
8. Sistem melakukan pengecekan data transaksi pembayaran.

E-1 Saldo tidak mencukupi

9. Sistem mengubah status pembayaran dan menampilkan informasi bahwa pembayaran telah berhasil dilakukan.

10. Use Case selesai

5. Alternative Flow

none

6. Error Flow

E-1 Saldo tidak mencukupi

1. Sistem menampilkan pesan peringatan bahwa saldo tidak mencukupi untuk membayar transaksi pemesanan.
2. Kembali ke Basic Flow Langkah ke 6

7. PreConditions

Aktor telah memasuki sistem

8. PostConditions

1. Aktor telah berhasil melakukan pembayaran suatu transaksi pemesanan.
2. Status transaksi pemesanan berhasil diperbarui.

4.1.8 Use case Spesification : *Login*

1. Brief Description

Use Case ini digunakan oleh aktor untuk login ke dalam aplikasi.

2. Primary Actor

Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan form login dan meminta inputan *username* dan *password* aktor.
3. Aktor memasukkan *username* dan *password*-nya dan meminta autentifikasi.
4. Sistem mengecek *username* dan *password* yang diinputkan

E-1 *Username* dan/atau *password* salah.

5. Sistem menampilkan halaman utama.
6. Use case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 *username* dan/atau *password* yang diinputkan aktor tidak ada pada *database* atau salah.

1. Sistem menampilkan peringatan bahwa data yang diinputkan tidak ada pada *database* atau salah.
2. Kembali ke Basic Flow langkah ke 2

7. PreConditions

Aktor telah memasuki sistem

8. PostConditions

Aktor berhasil login.

4.1.9 Use case Spesification : *Mengelola Akun Member*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data akun member yang terdiri dari fungsi mengubah foto profil, mengubah data akun, dan mengubah password.

2. Primary Actor

Member

3. Supporting Actor

none

4. Basic Flow

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

2. Sistem menampilkan data akun sementara dan memberikan antarmuka untuk mengubah foto profil, data akun, dan mengubah password.
3. Aktor memilih untuk mengubah foto profil
 - A-1 Aktor memilih untuk mengubah data akun
 - A-2 Aktor memilih untuk mengubah password
4. Sistem meminta file foto.
5. Aktor menginputkan foto.
6. Aktor menekan tombol Simpan untuk menyimpan perubahan.
7. Sistem mengecek file foto yang diinputkan.
 - E-1 File foto tidak sesuai kriteria
8. Sistem menyimpan file foto yang diinputkan
9. Sistem memberikan informasi bahwa pembaruan berhasil disimpan dan diperbarui.
10. Use case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk mengubah data akun
 1. Sistem meminta data nama, email, tanggal lahir, alamat, dan nomor telepon.
 2. Aktor memasukkan data nama, email, tanggal lahir, alamat, dan nomor telepon.
 3. Aktor menekan tombol Simpan untuk menyimpan perubahan.
 4. Sistem mengecek data yang diinputkan.
 - E-2 Email telah terdaftar
 5. Sistem menyimpan data akun yang telah diperbarui ke *database*.
 6. Berlanjut ke Basic Flow langkah ke 9
- A-2 Aktor memilih untuk mengubah password

1. Sistem meminta password lama, password baru, dan konfirmasi password baru.
2. Aktor memasukkan password lama, password baru, dan konfirmasi password baru.
3. Aktor menekan tombol Simpan untuk menyimpan perubahan.
4. Sistem mengecek data password diinputkan.
 E-3 Password lama salah
 E-4 Password baru dan konfirmasi password baru salah
5. Sistem menyimpan password yang telah diperbarui ke *database*.
6. Berlanjut ke Basic Flow langkah ke 9

6. Error Flow

E-1 File Foto tidak sesuai kriteria

1. Sistem memberikan pesan peringatan bahwa foto tidak sesuai dengan kriteria.
2. Kembali ke Basic Flow Langkah ke 4

E-2 Email telah terdaftar

1. Sistem memberikan pesan peringatan bahwa email telah terdaftar.
2. Kembali ke Alternative Flow A-1 Langkah ke 2

E-3 Password lama salah

1. Sistem memberikan pesan peringatan bahwa password lama salah.
2. Kembali ke Alternative Flow A-2 Langkah ke 2

E-4 Password baru dan konfirmasi password baru salah

1. Sistem memberikan pesan peringatan bahwa password baru dan konfirmasi password baru tidak sama.

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

7. PreConditions

1. Use case Login telah dilakukan
1. Aktor telah memasuki sistem

8. PostConditions

Data akun di *database* telah diperbarui.

4.1.10 Use case Spesification : Membuka Toko

1. Brief Description

Use Case ini digunakan oleh aktor untuk mendaftar toko ke sistem.

2. Primary Actor

Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pendaftaran toko.
2. Sistem menampilkan *form* data pendaftaran dan meminta nama toko, alamat toko, nomor telepon toko, slogan toko, deskripsi toko, dan logo toko.
3. Aktor memasukkan nama toko, alamat toko, nomor telepon toko, slogan toko, deskripsi toko, dan logo toko.
4. Aktor mengklik tombol Daftar untuk menyimpan data toko yang telah diinputkan.

5. Sistem memeriksa nama toko, alamat toko, nomor telepon toko, slogan toko, deskripsi toko, dan logo toko yang diinputkan oleh aktor.

E-1 Nama toko telah terdaftar

E-2 Foto tidak sesuai kriteria

6. Sistem menyimpan data toko ke *database*.

7. Sistem menampilkan informasi bahwa proses pendaftaran telah berhasil.

8. Use case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Nama toko telah terdaftar

1. Sistem menampilkan peringatan bahwa username diinputkan telah terdaftar.

2. Kembali ke Basic Flow langkah ke 3.

E-2 Foto tidak sesuai kriteria

1. Sistem memberikan pesan peringatan bahwa foto yang diinputkan tidak sesuai kriteria.

2. Kembali ke Basic Flow langkah ke 3.

7. PreConditions

1. Use case login telah dilakukan

2. Aktor telah memasuki sistem.

5. PostConditions

Toko milik aktor telah terdaftar ke dalam sistem.

4.1.11 Use case Spesification : Melakukan Negosiasi Harga Produk

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan negosiasi harga suatu produk toko.

2. Primary Actor

Member

3. Supporting Actor

Toko

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan negosiasi harga produk.
2. Sistem menampilkan produk-produk yang dimiliki oleh berbagai toko.
3. Aktor memilih suatu produk yang akan dinego.
4. Sistem menampilkan *form* negosiasi harga produk dan jumlah produk yang ingin dinego.
5. Aktor memasukkan harga negosiasi produk.
6. Sistem mengecek jumlah produk yang ingin dinego.
E-1 Jumlah produk melebihi jumlah stok produk.
7. Sistem menyimpan data negosiasi suatu produk ke database.
8. Use case selesai

5. Alternative Flow

none

6. Error Flow

E-1 Jumlah produk melebihi jumlah stok produk

1. Sistem menampilkan peringatan bahwa jumlah produk melebihi jumlah stok produk.

2. Kembali ke Basic Flow langkah ke 4.

7. PreConditions

1. Use case login telah dilakukan

2. Aktor telah memasuki sistem

8. PostConditions

Data negosiasi harga produk telah diinputkan ke dalam database.

4.1.12 Use case Spesification : Membuat Testimonial

1. Brief Description

Use Case ini digunakan oleh aktor untuk membuat testimonial tentang toko dan produk yang telah dibeli.

2. Primary Actor

Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk memberikan testimonial terhadap produk atau toko.

2. Sistem menampilkan produk-produk yang telah dibeli oleh Aktor dan belum pernah diberikan testimonial terhadap produk atau toko tersebut.

3. Aktor memilih suatu produk untuk diberikan testimonial

4. Sistem menampilkan *form* testimonial.

5. Aktor memasukkan testimonial.

6. Sistem menyimpan data testimonial ke dalam *database*.

7. Use case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Produk yang dicari tidak ditemukan

1. Sistem menampilkan peringatan bahwa produk yang dicari tidak ditemukan.

2. Kembali ke Basic Flow langkah ke 2.

7. PreConditions

1. Use case login telah dilakukan.

2. Aktor telah memasuki sistem.

3. Aktor pernah melakukan transaksi pemesanan.

8. PostConditions

Aktor telah memberikan testimonial terhadap suatu produk atau toko.

4.1.13 Use case Spesification : Memfavoritkan Produk

1. Brief Description

Use Case ini digunakan oleh aktor untuk menjadikan suatu produk sebagai produk favoritnya.

2. Primary Actor

Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk memfavoritkan suatu produk.

2. Sistem menampilkan list produk.

3. Aktor memilih suatu produk untuk dijadikan sebagai produk favoritnya.
4. Sistem menyimpan data favorit produk ke dalam *database*.
5. Use case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use case login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

Aktor telah membuat suatu produk sebagai produk favoritnya.

4.1.14 Use case Spesification : Mengirim pesan

1. Brief Description

Use Case ini digunakan oleh aktor untuk saling mengirim pesan antar *member*.

2. Primary Actor

Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengirim atau melihat pesan.
2. Sistem menampilkan pesan-pesan yang ada.
3. Aktor membuat pesan baru.
A-1 Aktor membalas pesan yang masuk.
4. Sistem menampilkan *form* pengiriman pesan baru.

5. Aktor mengisi username Aktor lainnya dan mengisi isi pesan.
6. Aktor menekan tombol Kirim untuk mengirim pesan.
7. Sistem menyimpan data pesan ke *database*.
8. Use case selesai.

5. Alternative Flow

A-1 Aktor membalas pesan masuk

1. Aktor memilih suatu pesan dari Aktor lain.
2. Sistem menampilkan detail pesan.
3. Aktor mengisi isi pesan balasan.
4. Berlanjut ke Basic Flow langkah ke 6.

6. Error Flow

none

7. PreConditions

1. Use case login telah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

Data pesan telah diinputkan ke dalam *database*.

4.1.15 Use case Specification : Mengelola Data Toko

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data toko *member* yang terdiri dari fungsi mengubah logo toko dan mengubah data toko.

2. Primary Actor

Toko

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data toko.
2. Sistem menampilkan data toko sementara dan memberikan antarmuka untuk mengubah logo dan data toko.
3. Aktor memilih untuk mengubah logo toko
A-1 Aktor memilih untuk mengubah data toko
4. Sistem meminta file logo.
5. Aktor menginputkan logo.
6. Aktor menekan tombol Simpan untuk menyimpan perubahan.
7. Sistem mengecek file logo yang diinputkan.
E-1 File logo tidak sesuai kriteria
8. Sistem menyimpan file logo yang diinputkan
9. Sistem memberikan informasi bahwa pembaruan berhasil disimpan dan diperbarui.
10. Use case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk mengubah data toko
1. Sistem meminta data nama toko, alamat toko, nomor telepon toko, slogan toko, dan deskripsi toko.
 2. Aktor memasukkan data nama toko, alamat toko, nomor telepon toko, slogan toko, dan deskripsi toko.
 3. Aktor menekan tombol Simpan untuk menyimpan perubahan.
 4. Sistem mengecek data yang diinputkan.
E-2 Nama toko telah terdaftar

5. Sistem menyimpan data akun yang telah diperbarui ke *database*.

6. Berlanjut ke Basic Flow langkah ke 9

6. Error Flow

E-1 File logo tidak sesuai kriteria

1. Sistem memberikan pesan peringatan bahwa logo tidak sesuai dengan kriteria.
2. Kembali ke Basic Flow Langkah ke 4

E-2 Nama toko telah terdaftar

1. Sistem memberikan pesan peringatan bahwa email telah terdaftar.
2. Kembali ke Alternative Flow A-1 Langkah ke 2

7. PreConditions

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

8. PostConditions

1. Data toko di *database* telah diperbarui.

4.1.16 Use case Specification : Mengelola Data Produk

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data produk yang terdiri dari fungsi tampil, tambah, edit, hapus, dan cari data produk.

2. Primary Actor

Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data produk.
2. Sistem menampilkan data produk, memberikan antarmuka untuk penambahan dan pencarian data produk.
3. Aktor melakukan penambahan data produk
 - A-1 Aktor memilih untuk mengubah data produk
 - A-2 Aktor memilih untuk menghapus data produk
 - A-3 Aktor memilih untuk mencari data produk
4. Aktor memasukkan data produk.
5. Aktor mengklik tombol Tambah untuk menyimpan data produk yang telah diinputkan.
6. Sistem memeriksa data produk.
 - E-1 Data produk yang diinputkan kurang lengkap atau salah.
7. Sistem menyimpan data produk ke *database*.
8. Sistem menampilkan informasi bahwa proses penambahan telah berhasil.
9. Use case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk mengubah data produk
 1. Aktor memilih data produk yang akan diedit.
 2. Sistem menampilkan *form* ubah data produk.
 3. Aktor memasukkan data produk.
 4. Aktor menekan tombol Simpan untuk menyimpan perubahan.
 5. Sistem mengecek data produk yang diinputkan.

E-1 Data produk yang diinputkan kurang lengkap atau salah.

6. Sistem menyimpan data produk yang telah diperbarui ke *database*.

7. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk menghapus data produk

1. Aktor memilih data produk yang akan dihapus.

2. Aktor menekan tombol Hapus untuk menghapus data produk.

3. Sistem meminta konfirmasi penghapusan.

4. Sistem memperbarui data hapus semua produk yang telah dipilih pada *database*.

5. Berlanjut ke Basic Flow langkah ke 8

A-1 Aktor memilih untuk mencari data produk

1. Sistem menampilkan *form* pencarian data produk.

2. Aktor memasukkan kriteria data produk yang dicari.

3. Aktor menekan tombol Cari untuk mencari data produk.

4. Sistem mencari dan menampilkan data produk yang dicari Aktor.

5. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Data produk yang diinputkan kurang lengkap atau salah

1. Sistem menampilkan peringatan bahwa data yang diinputkan kurang lengkap atau salah.

2. Kembali ke Basic Flow langkah ke 3.

7. PreConditions

1. Use case login telah dilakukan
2. Aktor telah memasuki sistem.

8. PostConditions

Produk milik toko telah terdaftar ke dalam *database*.

4.1.17 Use case Specification : *Mengelola Data Transaksi Penjualan*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data transaksi penjualan suatu produk toko.

2. Primary Actor

Toko

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data transaksi penjualan.
2. Sistem menampilkan data transaksi penjualan.
3. Aktor memilih data transaksi penjualan.
4. Sistem menampilkan data detail transaksi penjualan dan pembayaran.
5. Aktor mengubah status transaksi.
6. Sistem menyimpan data transaksi penjualan suatu produk ke *database*
7. Use case selesai

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use case login telah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

Data detail transaksi penjualan di database telah terupdate

4.1.18 Use case Spesification : Login Admin

1. Brief Description

Use Case ini digunakan oleh aktor untuk login ke dalam aplikasi.

2. Primary Actor

Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan form login dan meminta inputan *username* dan *password* aktor.
3. Aktor memasukkan *username* dan *password*-nya dan meminta autentifikasi.
4. Sistem mengecek *username* dan *password* yang diinputkan
E-1 *Username* dan/atau *password* salah.
5. Sistem menampilkan halaman utama.
6. Use case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 *Username* dan/atau *password* yang diinputkan aktor tidak ada pada *database* atau salah.

1. Sistem menampilkan peringatan bahwa data yang diinputkan tidak ada pada *database* atau salah.
2. Kembali ke Basic Flow langkah ke 2

7. PreConditions

Aktor telah memasuki sistem

8. PostConditions

Aktor berhasil login.

4.1.19 Use case Spesification : *Mengelola Akun Admin*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data akun admin yang terdiri dari fungsi mengubah foto profil, mengubah data akun, dan mengubah password.

2. Primary Actor

Admin

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data akun.
2. Sistem menampilkan data akun sementara dan memberikan antarmuka untuk mengubah foto profil, data akun, dan mengubah password.
3. Aktor memilih untuk mengubah foto profil
 - A-1 Aktor memilih untuk mengubah data akun
 - A-2 Aktor memilih untuk mengubah password

4. Sistem meminta file foto.
5. Aktor menginputkan foto.
6. Aktor menekan tombol Simpan untuk menyimpan perubahan.
7. Sistem mengecek file foto yang diinputkan.
E-1 File foto tidak sesuai kriteria
8. Sistem menyimpan file foto yang diinputkan
9. Sistem memberikan informasi bahwa pembaruan berhasil disimpan dan diperbarui.
10. Use case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk mengubah data akun
 7. Sistem meminta data nama, email, tanggal lahir, alamat, dan nomor telepon.
 8. Aktor memasukkan data nama, email, tanggal lahir, alamat, dan nomor telepon.
 9. Aktor menekan tombol Simpan untuk menyimpan perubahan.
 10. Sistem mengecek data yang diinputkan.
E-2 Email telah terdaftar
 11. Sistem menyimpan data akun yang telah diperbarui ke *database*.
 12. Berlanjut ke Basic Flow langkah ke 9
- A-2 Aktor memilih untuk mengubah password
 7. Sistem meminta password lama, password baru, dan konfirmasi password baru.
 8. Aktor memasukkan password lama, password baru, dan konfirmasi password baru.
 9. Aktor menekan tombol Simpan untuk menyimpan perubahan.

10. Sistem mengecek data password diinputkan.

E-3 Password lama salah

E-4 Password baru dan konfirmasi password baru salah

11. Sistem menyimpan password yang telah diperbarui ke *database*.

12. Berlanjut ke Basic Flow langkah ke 9

6. Error Flow

E-1 File Foto tidak sesuai kriteria

1. Sistem memberikan pesan peringatan bahwa foto tidak sesuai dengan kriteria.
2. Kembali ke Basic Flow Langkah ke 4

E-2 Email telah terdaftar

1. Sistem memberikan pesan peringatan bahwa email telah terdaftar.
2. Kembali ke Alternative Flow A-1 Langkah ke 2

E-3 Password lama salah

1. Sistem memberikan pesan peringatan bahwa password lama salah.
2. Kembali ke Alternative Flow A-2 Langkah ke 2

E-4 Password baru dan konfirmasi password baru salah

1. Sistem memberikan pesan peringatan bahwa password baru dan konfirmasi password baru tidak sama.
2. Kembali ke Alternative Flow A-2 Langkah ke 2

7. PreConditions

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

8. PostConditions

Data akun di *database* telah diperbarui.

4.1.20 Use case Spesification : *Menjela Katego* *Produk*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data kategori produk.

2. Primary Actor

Admin

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data kategori produk.
2. Sistem menampilkan data kategori produk, memberikan antarmuka untuk penambahan dan pencarian data kategori produk.
3. Aktor melakukan penambahan data kateogori produk
 - A-1 Aktor memilih untuk mengubah data kategori produk
 - A-2 Aktor memilih untuk menghapus data kategori produk
 - A-3 Aktor memilih untuk mencari data kategori produk
4. Aktor memasukkan data kategori produk.
5. Aktor mengklik tombol Tambah untuk menyimpan data kategori produk yang telah diinputkan.

6. Sistem memeriksa data kategori produk.

E-1 Data kategori produk yang diinputkan kurang lengkap atau salah.

7. Sistem menyimpan data kategori produk ke *database*.

8. Sistem menampilkan informasi bahwa proses penambahan kategori produk telah berhasil.

9. Use case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk mengubah data kategori produk

1. Aktor memilih data kategori produk yang akan diedit.

2. Sistem menampilkan *form* ubah data kategori produk.

3. Aktor memasukkan data kategori produk.

4. Aktor menekan tombol Simpan untuk menyimpan perubahan.

5. Sistem mengecek data kategori produk yang diinputkan.

E-1 Data produk yang diinputkan kurang lengkap atau salah atau telah ada.

6. Sistem menyimpan data kategori produk yang telah diperbarui ke *database*.

7. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk menghapus data kategori produk

1. Sistem menampilkan data kategori produk.

2. Aktor memilih data kategori produk.

3. Aktor menekan tombol Hapus untuk menghapus data kategori produk.

4. Sistem meminta konfirmasi penghapusan.
5. Sistem memperbarui data hapus semu kategori produk yang telah dipilih pada *database*.
6. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk mencari data kategori produk

1. Sistem menampilkan *form* pencarian data kategori produk.
2. Aktor memasukkan kriteria data kategori produk yang dicari.
3. Aktor menekan tombol Cari untuk mencari data kategori produk.
4. Sistem mencari dan menampilkan data kategori produk yang dicari Aktor.
5. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Data produk yang diinputkan kurang lengkap atau salah atau telah ada

1. Sistem menampilkan peringatan bahwa data yang diinputkan kurang lengkap atau salah atau telah ada pada *database*.
2. Kembali ke Basic Flow langkah ke 3.

7. PreConditions

1. Use case login telah dilakukan
2. Aktor telah memasuki sistem.

8. PostConditions

Pengelolaan kategori produk telah sukses dilakukan dan tersimpan ke dalam *database*.

4.1.21 Use case Spesification : Mengawasi Produk Toko

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengawasi produk toko termasuk untuk memberikan peringatan kepada pemilik toko tentang produk yang dimiliki.

2. Primary Actor

Admin

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengawasi produk toko.
2. Sistem menampilkan data produk dan pencarian data produk.
3. Aktor melakukan pencarian data produk
 - A-1 Aktor memilih untuk memberikan peringatan kepada toko tentang suatu produk yang dimiliki toko tersebut.
4. Sistem menampilkan *form* pencarian produk.
5. Aktor memasukkan nama produk yang dicari.
6. Aktor mengklik tombol Cari untuk mencari data produk yang telah diinputkan.
7. Sistem mencari data produk yang dicari Aktor.
 - E-1 Data produk tidak ditemukan.
8. Sistem menampilkan data produk yang sesuai dengan kata pencarian yang diinputkan oleh Aktor.
9. Use case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk memberikan peringatan kepada toko tentang suatu produk yang dimiliki toko tersebut

1. Aktor memilih data kategori produk yang akan diberikan peringatan.
2. Sistem menampilkan *form* peringatan data produk.
3. Aktor memasukkan data peringatan suatu produk toko.
4. Aktor menekan tombol Kirim untuk mengirim peringatan kepada toko pemilik produk.
5. Sistem mengirimkan email peringatan kepada pemilik toko.
6. Berlanjut ke Basic Flow langkah ke 9

6. Error Flow

E-1 Data produk tidak ditemukan.

1. Sistem menampilkan peringatan bahwa data produk yang dicari tidak ditemukan.
2. Kembali ke Basic Flow langkah ke 4.

7. PreConditions

1. Use case login telah dilakukan
2. Aktor telah memasuki sistem.

8. PostConditions

Pengelolaan kategori produk telah sukses dilakukan dan tersimpan ke dalam *database*.

4.1.22 Use case Spesification : Mengawasi Transaksi Toko

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengawasi transaksi toko termasuk untuk memberikan

Program Studi Teknik Informatika	SKPL – TaniaMall	62/65
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika- UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

peringatan kepada pemilik toko tentang transaksi penjualan yang dimiliki toko tersebut.

2. Primary Actor

Admin

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengawasi transaksi toko.
2. Sistem menampilkan data transaksi toko dan pencarian data transaksi toko.
3. Aktor melakukan pencarian data transaksi toko.
A-1 Aktor memilih untuk memberikan peringatan kepada toko tentang suatu transaksi yang dimiliki toko tersebut.
4. Sistem menampilkan *form* pencarian produk.
5. Aktor memasukkan nama produk yang dicari.
6. Aktor mengklik tombol Cari untuk mencari data produk yang telah diinputkan.
7. Sistem mencari data produk yang dicari Aktor.
E-1 Data produk tidak ditemukan.
8. Sistem menampilkan data produk yang sesuai dengan kata pencarian yang diinputkan oleh Aktor.
9. Use case selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk memberikan peringatan kepada toko tentang suatu produk yang dimiliki toko tersebut
7. Aktor memilih data kategori produk yang akan diberikan peringatan.

8. Sistem menampilkan *form* peringatan data produk.
9. Aktor memasukkan data peringatan suatu produk toko.
10. Aktor menekan tombol Kirim untuk mengirim peringatan kepada toko pemilik produk.
11. Sistem mengirimkan email peringatan kepada pemilik toko.
12. Berlanjut ke Basic Flow langkah ke 9

6. Error Flow

E-1 Data produk tidak ditemukan.

3. Sistem menampilkan peringatan bahwa data produk yang dicari tidak ditemukan.
4. Kembali ke Basic Flow langkah ke 4.

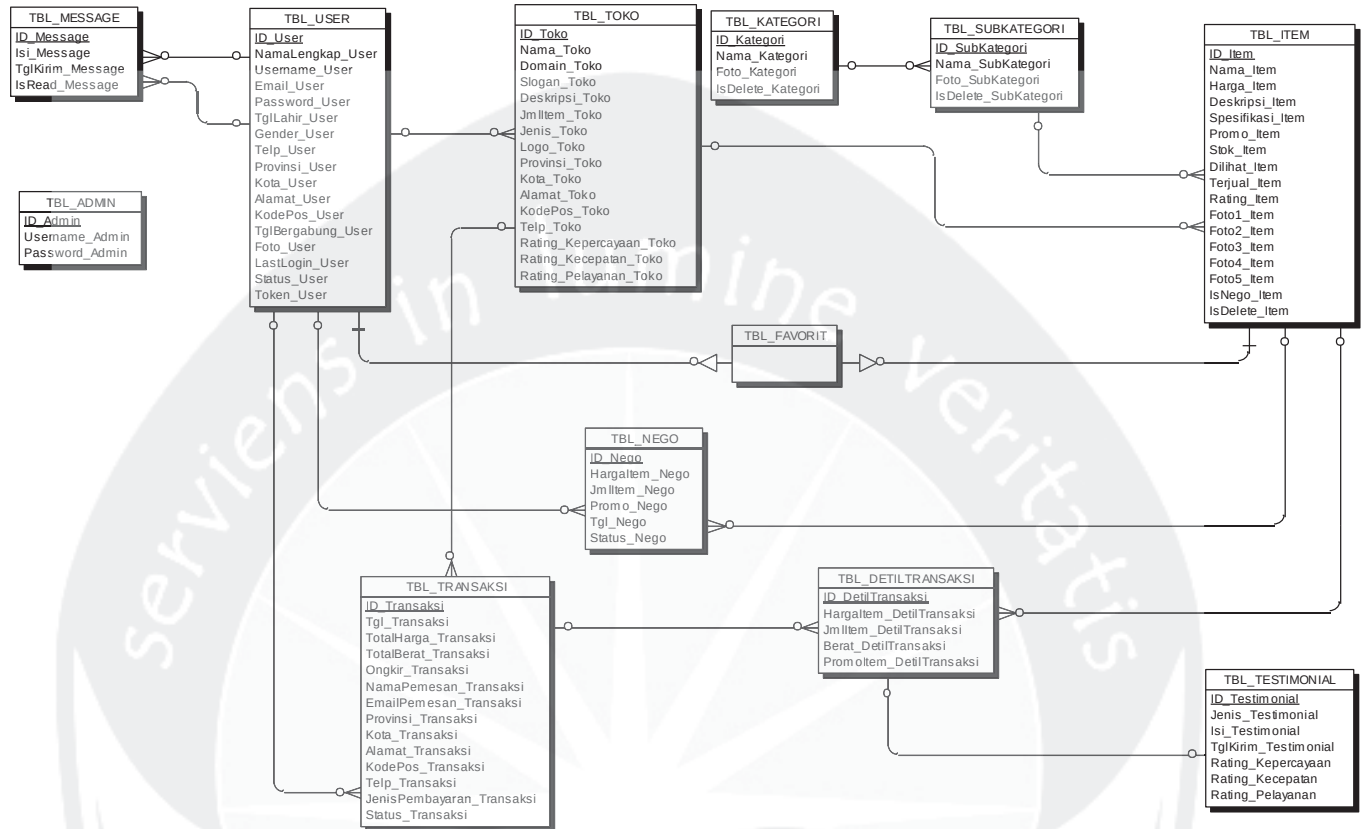
7. PreConditions

1. Use case login telah dilakukan
2. Aktor telah memasuki sistem.

8. PostConditions

Pengelolaan kategori produk telah sukses dilakukan dan tersimpan ke dalam *database*.

5 Entity Relationship Diagram (ERD)



Gambar 4. Entity Relationship Diagram (ERD)