

BAB VI

PENUTUP

Pada bab penutup ini akan diberikan kesimpulan dan saran yang didapatkan selama pembuatan Tugas Akhir.

6.1. Kesimpulan

Setelah situs SOS ini selesai diimplementasikan, dapat ditarik beberapa kesimpulan sebagai berikut:

- a. Perangkat lunak SOS telah berhasil dibangun untuk mempermudah pengguna dalam melakukan permintaan dan penawaran barang.
- b. Google Map API telah berhasil diimplementasikan untuk menampilkan lokasi pengguna.
- c. Dari hasil pengujian dapat diketahui bahwa fungsi-fungsi yang disediakan oleh perangkat lunak SOS berjalan dengan benar dan sesuai dengan yang diharapkan

6.2. Saran

Dari proses analisis, perancangan, implementasi hingga pengujian sistem pada pembuatan Tugas Akhir, penulis mendapatkan beberapa saran untuk pengembangan lebih lanjut perangkat lunak ShareEvent ini, yaitu:

- a. Mengubah logo dan tampilan dari situs SOS agar pengguna mudah memahami bahwa situs SOS merupakan situs jual beli.
- b. Menambahkan informasi tanggal pendaftaran user dan memberikan badge secara otomatis kepada pengguna untuk mempermudah pengguna mencari lawan transaksi yang dapat dipercaya.

- c. Menambahkan fungsi pembayaran online dan verifikasi barang ketika barang sudah sampai pada pembeli.
- d. Memperbaiki tampilan situs menjadi lebih rapi agar lebih nyaman digunakan.



DAFTAR PUSTAKA

- Broulik, B., 2011, *Pro jQuery Mobile*, Apress.
- Chow, C., Mokbel, M. F., dan Liu, X., 2011, *Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments*, *GeoInformatica*.
- Dhar, S., dan Varshney, U., 2011, *Challenges and Business Models for Mobile Location-based Services and Advertising*, *Communications of The Acm*.
- Han, B., dan Windsor, J., 2011, *User's Willingness To Pay On Social Network Sites*, *The Journal of Computer Information*.
- Isaías, P., Miranda, P., dan Pífano, S., 2009, *Critical Success Factors for Web 2.0 - A Reference Framework*, *Online Communities*.
- Juniar, P. R., 2010. *Pembangunan Aplikasi Sosial Networking Berbasis Lokasi*, Universitas Atma Jaya Yogyakarta, Yogyakarta.
- Kim, Y., Choi, S. M., dan Sohn, D., 2008, *Understanding Motives For Using Social Network Site*, *American Academy of Advertising*.
- Meyer, R. L., dan Mollod, J. P., 2008, *Social Network Site Litigation*, *The IP Litigator : Devoted to Intellectual Property Litigation and Enforcement*.
- Nabizadeh, T., Zahra, T. G., dan Nabizadeh, M., 2012, *Representing a Frame for Creating Electronic Trust and Developing E-Commerce and Its Impact on Developing Direct Export*, *Interdisciplinary Journal of Contemporary Research In Business*.
- Petrova, K., dan Wang, B., 2011, *Location-based services deployment and demand: a roadmap model*, *Electronic Commerce Research*.

- Saputra, A., 2011, *Trik Kolaborasi Codeigniter & jQuery*. Penerbit Lokopedia. Yogyakarta.
- Sledgianowski, D., dan Kulviwat, S., 2009, *Using Social Network Sites: The Effects of Playfulness, Critical Mass and Trust In a Hedonic Context*, The Journal of Computer Information Systems.
- Stephen, A. T., dan Toubia, O., 2010, *Deriving Value from Social Commerce Networks*. Journal of Marketing Research.
- Sunaryo, M. O. A., 2012, *Pembangunan Aplikasi Jejaring Sosial untuk Berbagi dan Mengingat Acara Berbasis Android*, Universitas Atma Jaya Yogyakarta, Yogyakarta.
- Yanuar, S., 2010, *Pembangunan Aplikasi Katalog Belanja Berbasis Lokasi Memanfaatkan J2ME dan Google Maps API*, Universitas Atma Jaya Yogyakarta, Yogyakarta.
- Zaki, A., 2010, *Keliling Dunia dengan Google Earth + Google Maps*, CV Andi Offset, Yogyakarta.
- _____, 2012, *CodeIgniter at a Glance*, http://codeigniter.com/user_guide/overview/at_a_glance.html, diakses pada tanggal 11 Juni 2012 pukul 15.16.
- _____, 2012, *Welcome to CodeIgniter!*, <http://codeigniter.com/>, diakses pada tanggal 11 Juni 2012 pukul 15.07.
- _____, 2012, *Model-View-Controller*, http://codeigniter.com/user_guide/overview/mvc.html, diakses pada tanggal 11 Juni 2012 pukul 15.25.
- _____, 2012, *Design and Architectural Goals*, http://codeigniter.com/user_guide/overview/

goals.html, diakses pada tanggal 11 Juni 2012 pukul 15.29.

_____, 2012, *FAQ*,
<https://developers.google.com/maps/faq?hl=id#whatis>, diakses pada tanggal 11 Juni 2012 pukul 14.27.

_____, 2012, *Static Maps API V2 Developer Guide*,
<https://developers.google.com/maps/documentation/staticmaps/?hl=id>, diakses pada tanggal 11 Juni 2012 pukul 14.30.

_____, 2012, *The Solution for Maps Applications for both the Desktop and Mobile Devices*,
<https://developers.google.com/maps/documentation/javascript/?hl=id>, diakses pada tanggal 11 Juni 2012 pukul 14.29.

_____, 2012, *Welcome to Google Maps*,
<http://support.google.com/maps/bin/answer.py?hl=en&topic=1687350&answer=144352>, diakses pada tanggal 11 Juni 2012 pukul 14.15.

Sebastian, I., 2012, 2012, *Transaksi Online Shopping di Indonesia USD 4,1 miliar*,
<http://inet.detik.com/read/2012/02/29/082349/1854134/319/2012-transaksi-online-shopping-di-indonesia-usd-41-miliar>, diakses pada tanggal 21 April 2012 pukul 22.10.

LAMPIRAN

SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

SOS4HELP

(Social Online Shop)

Dipersiapkan oleh:

Lilyantika / 5948

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		SKPL-SOS		1/50
	Fakultas Teknologi Industri	Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperik sa oleh								
Disetuj ui 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.....	6
1.4	Referensi.....	7
1.5	Deskripsi umum (Overview).....	7
2	Deskripsi Kebutuhan.....	8
2.1	Perspektif produk.....	8
2.2	Fungsi Produk.....	9
2.3	Karakteristik Pengguna.....	18
2.4	Batasan-batasan.....	18
2.5	Asumsi dan Ketergantungan.....	19
3	Kebutuhan khusus.....	19
3.1	Kebutuhan antarmuka eksternal.....	19
3.2	Kebutuhan fungsionalitas Perangkat Lunak.....	21
4	Spesifikasi Rinci Kebutuhan.....	23
4.1	Spesifikasi Kebutuhan Fungsionalitas.....	23
5	Entity Relationship Diagram (ERD).....	50

Daftar Gambar

Gambar 1	Arsitektur Perangkat Lunak SOS.....	9
Gambar 2	Use Case Diagram Situs SOS.....	22
Gambar 3	Entity Relationship Diagram.....	50



1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SOS (Social Online Shop) 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-SOS ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat Lunak SOS dikembangkan dengan tujuan untuk :

1. Menyediakan sarana bagi pengguna untuk menyebarkan informasi mengenai produk yang dijual atau dibutuhkan kepada pengguna lainnya.
2. Mempermudah pengguna untuk mencari pembeli atau penjual yang tepat dari sisi jarak maupun kepercayaan.
3. Mempererat hubungan antar pengguna dengan saling membantu untuk memenuhi kebutuhan antara satu pengguna dengan pengguna lainnya.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari

Program Studi Teknik Informatika	SKPL – SPP	6/50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

	perangkat lunak yang akan dikembangkan.
SKPL-SOS-XXX	Kode yang merepresentasikan kebutuhan pada SOS (Social Online Shop) dimana XXX merupakan nomor fungsi produk.
SOS	Social Online Shop.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Lilyantika, *Spesifikasi Kebutuhan Perangkat Lunak SPP*, Universitas Atma Jaya Yogyakarta, 2012.

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 SOS 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 SOS tersebut.

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

2 Deskripsi Kebutuhan

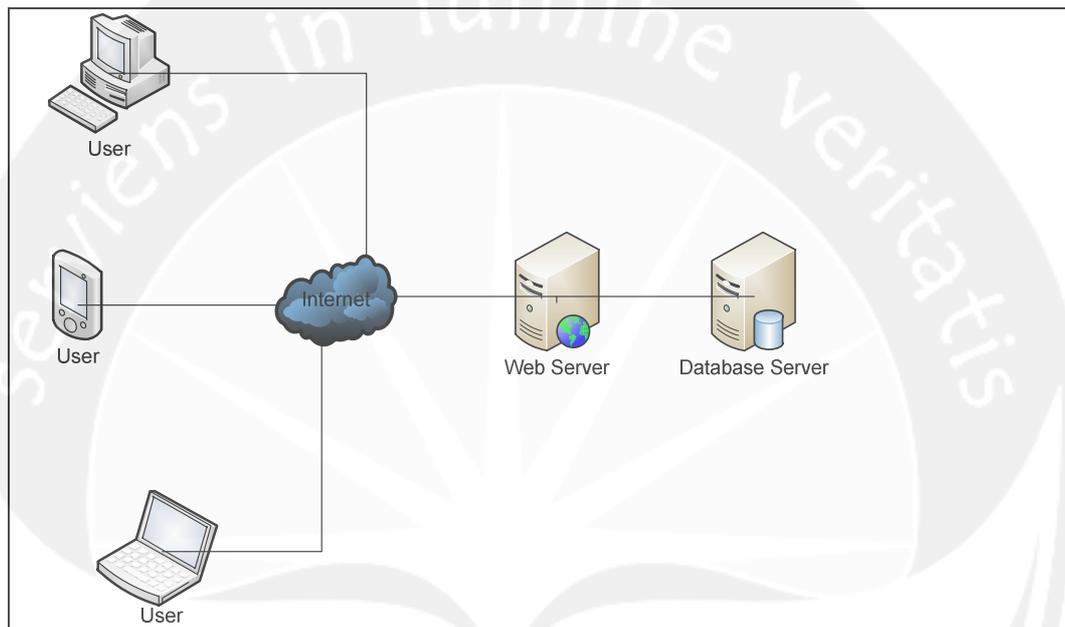
2.1 Perspektif produk

SOS merupakan perangkat lunak yang dikembangkan untuk mempermudah pengguna web agar dapat saling bertukar informasi mengenai produk yang ingin dibeli atau produk yang sedang dijual. Melalui situs ini, pengguna web dapat membagikan produk yang sedang dijual atau produk yang ingin dibeli, mengelola informasi lokasi pengguna untuk mempermudah pencarian penjual atau pembeli terdekat, mencari produk yang sedang dijual atau diinginkan oleh teman atau seluruh pengguna sos, dan berbagai kegiatan lainnya yang mendukung proses penawaran, permintaan dan pencarian produk oleh pengguna.

Perangkat lunak SOS ini berjalan pada browser yang mendukung kemampuan javascript dan ajax, dan dibuat menggunakan bahasa pemrograman PHP. Sedangkan untuk lingkungan pemrogramannya menggunakan framework Codeigniter dengan DBMS (Data Base Management System) yang digunakan yaitu Mysql.

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (Graphical User Interface). Pada sistem ini, seperti terlihat pada gambar 1, arsitektur perangkat lunak yang digunakan berupa client server, di mana semua data disimpan di server. Pengguna dapat mengakses data yang terdapat di server tersebut secara online dengan memanggil web service pada web site yang tersedia di web server.

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



Gambar 1 Arsitektur Perangkat Lunak SOS

2.2 Fungsi Produk

Fungsi produk perangkat lunak SOS adalah sebagai berikut :

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

Merupakan fungsi yang digunakan oleh pengguna situs SOS untuk masuk ke dalam sistem sesuai dengan status masing-masing.

2. Fungsi *Register* (**SKPL-SOS-002**).

Merupakan fungsi yang digunakan oleh pengguna untuk mendaftarkan diri sebagai pengguna dari situs SOS.

3. Fungsi *Manage Users* (**SKPL-SOS-003**).

Merupakan fungsi yang digunakan untuk mengelola data pengguna. Fungsi ini hanya dapat diakses oleh Administrator.

Fungsi *Manage Users* mencakup :

- a. Fungsi *Show Users* (**SKPL-SOS-003-01**). Merupakan fungsi yang digunakan untuk menampilkan semua pengguna situs SOS.
- b. Fungsi *Show Detail User* (**SKPL-SOS-003-02**). Merupakan fungsi yang digunakan untuk menampilkan informasi rinci dari pengguna situs SOS.
- c. Fungsi *Delete User* (**SKPL-SOS-003-03**). Merupakan fungsi yang digunakan untuk menghapus pengguna situs SOS.

4. Fungsi *Manage Categories* (**SKPL-SOS-004**).

Merupakan fungsi yang digunakan untuk mengelola data kategori produk. Fungsi ini hanya dapat diakses oleh Administrator.

Fungsi *Manage Categories* mencakup :

- a. Fungsi *Insert Category* (**SKPL-SOS-004-01**). Merupakan fungsi yang digunakan untuk menambahkan kategori baru.

Program Studi Teknik Informatika	SKPL – SPP	10/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

b. Fungsi *Edit Category* (**SKPL-SOS-004-02**).
Merupakan fungsi yang digunakan untuk mengubah kategori yang telah ada.

c. Fungsi *Delete Category* (**SKPL-SOS-004-03**).
Merupakan fungsi yang digunakan untuk menghapus kategori yang telah ada.

d. Fungsi *Show Category* (**SKPL-SOS-004-04**).
Merupakan fungsi yang digunakan untuk menampilkan semua kategori yang telah ada.

5. Fungsi *Manage Profile* (**SKPL-SOS-005**).

Merupakan fungsi yang digunakan untuk mengelola profile pengguna. Fungsi ini dapat diakses oleh semua pengguna situs SOS.

Fungsi *Manage Profile* mencakup :

a. Fungsi *Change Username* (**SKPL-SOS-005-01**).
Merupakan fungsi yang digunakan untuk mengubah username yang digunakan untuk login.

b. Fungsi *Change Password* (**SKPL-SOS-005-02**).
Merupakan fungsi yang digunakan untuk mengubah password yang digunakan untuk login.

c. Fungsi *Change Profile Picture* (**SKPL-SOS-005-03**).
Merupakan fungsi yang digunakan untuk mengubah foto profil yang sedang digunakan.

d. Fungsi *Edit Profile* (**SKPL-SOS-005-04**).
Merupakan fungsi yang digunakan untuk mengubah data profil pengguna tersebut.

e. Fungsi *Edit Setting* (**SKPL-SOS-005-05**).
Merupakan fungsi yang digunakan untuk mengubah data setting pengguna tersebut.

f. Fungsi *Show Profile* (**SKPL-SOS-005-06**).
Merupakan fungsi yang digunakan untuk menampilkan data profil pengguna tersebut.

6. Fungsi *Manage Friends* (**SKPL-SOS-006**).

Merupakan fungsi yang digunakan untuk mengelola pertemanan antar pengguna. Fungsi ini dapat diakses oleh semua pengguna situs SOS.

Fungsi *Manage Friends* mencakup :

a. Fungsi *Insert Friend* (**SKPL-SOS-006-01**).
Merupakan fungsi yang digunakan untuk mengirimkan permintaan pertemanan kepada pengguna lain.

b. Fungsi *Accept Friend* (**SKPL-SOS-006-02**).
Merupakan fungsi yang digunakan untuk menerima tawaran pertemanan dari pengguna lain.

c. Fungsi *Reject Friend* (**SKPL-SOS-006-03**).
Merupakan fungsi yang digunakan untuk menolak pertemanan dari pengguna lain.

d. Fungsi *Delete Friend* (**SKPL-SOS-006-04**).
Merupakan fungsi yang digunakan untuk menghapus hubungan pertemanan dengan pengguna lain.

e. Fungsi *Show Friend* (**SKPL-SOS-006-05**). Merupakan fungsi yang digunakan untuk menampilkan semua teman yang dimiliki pengguna.

7. Fungsi *Manage Products* (**SKPL-SOS-007**).

Program Studi Teknik Informatika	SKPL – SPP	12/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan untuk mengelola produk yang diinginkan atau yang dijual. Fungsi ini dapat diakses oleh semua pengguna situs SOS

Fungsi *Manage Products* mencakup :

- a. Fungsi *Insert Product* (**SKPL-SOS-007-01**).
Merupakan fungsi yang digunakan untuk menambahkan produk yang baru.
- b. Fungsi *Edit Product* (**SKPL-SOS-007-02**).
Merupakan fungsi yang digunakan untuk mengubah data produk yang ada.
- c. Fungsi *Delete Product* (**SKPL-SOS-007-03**).
Merupakan fungsi yang digunakan untuk menghapus produk yang ada.
- d. Fungsi *Show Products* (**SKPL-SOS-007-04**).
Merupakan fungsi yang digunakan untuk menampilkan semua produk yang dimiliki oleh pengguna.
- e. Fungsi *Show Detail Product* (**SKPL-SOS-007-05**).
Merupakan fungsi yang digunakan untuk menampilkan detail produk yang dimiliki pengguna.
- f. Fungsi *Insert Photo Product* (**SKPL-SOS-007-06**).
Merupakan fungsi yang digunakan untuk menambahkan gambar atau foto pada produk.
- g. Fungsi *Delete Photo Product* (**SKPL-SOS-007-07**).
Merupakan fungsi yang digunakan untuk menghapus gambar atau foto pada produk.

h. Fungsi *Promote Product* (**SKPL-SOS-007-08**).
Merupakan fungsi yang digunakan untuk mempromosikan produk yang dimiliki.

8. Fungsi *Manage Places* (**SKPL-SOS-008**).

Merupakan fungsi yang digunakan untuk mengelola tempat. Fungsi ini dapat diakses oleh semua pengguna situs SOS.

Fungsi *Manage Places* mencakup :

- a. Fungsi *Insert Place* (**SKPL-SOS-008-01**).
Merupakan fungsi yang digunakan untuk menambahkan tempat yang baru.
- b. Fungsi *Delete Place* (**SKPL-SOS-008-02**).
Merupakan fungsi yang digunakan untuk menghapus tempat yang telah ada.
- c. Fungsi *Show Place* (**SKPL-SOS-008-03**). Merupakan fungsi yang digunakan untuk menampilkan semua tempat yang dimiliki.

9. Fungsi *Manage Testimony* (**SKPL-SOS-009**).

Merupakan fungsi yang digunakan untuk mengelola testimony. Fungsi ini dapat diakses oleh semua pengguna situs SOS

Fungsi *Manage Testimony* mencakup :

- a. Fungsi *Insert Testimony* (**SKPL-SOS-009-01**).
Merupakan fungsi yang digunakan untuk menambahkan testimonial ke halaman profil teman.

b. Fungsi *Edit Testimony* (**SKPL-SOS-009-02**).
Merupakan fungsi yang digunakan untuk mengedit testimonial yang telah dibuat.

c. Fungsi *Delete Testimony* (**SKPL-SOS-009-03**).
Merupakan fungsi yang digunakan untuk menghapus testimonial yang telah dibuat.

d. Fungsi *Show Testimony* (**SKPL-SOS-009-04**).
Merupakan fungsi yang digunakan untuk menampilkan semua testimonial yang ditujukan kepada pengguna.

10. Fungsi *Manage Message* (**SKPL-SOS-010**).

Merupakan fungsi yang digunakan untuk mengelola pesan. Fungsi ini dapat diakses oleh semua pengguna situs SOS

Fungsi *Manage Message* mencakup :

a. Fungsi *Insert Message* (**SKPL-SOS-010-01**).
Merupakan fungsi yang digunakan untuk menambahkan pesan kepada teman.

b. Fungsi *Insert Message Content* (**SKPL-SOS-010-02**).
Merupakan fungsi yang digunakan untuk menambahkan isi pesan kepada teman.

c. Fungsi *Show Message* (**SKPL-SOS-010-03**).
Merupakan fungsi yang digunakan untuk menampilkan semua pesan yang dimiliki.

d. Fungsi *Show Message Contents* (**SKPL-SOS-010-04**).
Merupakan fungsi yang digunakan untuk menampilkan isi dari pesan yang dimiliki.

11. Fungsi *Manage Appointment* (**SKPL-SOS-011**).

Program Studi Teknik Informatika	SKPL – SPP	15/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan untuk mengelola perjanjian transaksi antar pengguna. Fungsi ini dapat diakses oleh semua pengguna situs SOS

Fungsi *Manage Appointment* mencakup :

- a. Fungsi *Insert Appointment* (**SKPL-SOS-011-01**). Merupakan fungsi yang digunakan untuk menambahkan perjanjian oleh pihak yang menawarkan penjualan atau pembelian.
- b. Fungsi *Accept Appointment* (**SKPL-SOS-011-02**). Merupakan fungsi yang digunakan untuk menerima tawaran penjualan atau pembelian oleh pemilik produk.
- c. Fungsi *Reject Appointment* (**SKPL-SOS-011-03**). Merupakan fungsi yang digunakan untuk menolak tawaran penjualan atau pembelian oleh pemilik produk.
- d. Fungsi *Send Product or Money* (**SKPL-SOS-011-04**). Merupakan fungsi yang digunakan untuk mengubah status perjanjian menjadi terkirim yang menandakan bahwa barang atau uang telah dikirimkan oleh pihak yang mengirim.
- e. Fungsi *Accept Product or Money* (**SKPL-SOS-011-05**). Merupakan fungsi yang digunakan untuk mengubah status perjanjian menjadi diterima yang menandakan bahwa barang atau uang telah dikirimkan oleh pihak yang mengirim telah diterima.
- f. Fungsi *Edit Appointment* (**SKPL-SOS-011-06**). Merupakan fungsi yang digunakan untuk mengubah

Program Studi Teknik Informatika	SKPL – SPP	16/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

data dari perjanjian yang telah dibuat. Fungsi ini hanya dapat diakses oleh pihak yang membuat perjanjian selama perjanjian belum diterima.

g. Fungsi *Delete Appointment* (**SKPL-SOS-011-07**). Merupakan fungsi yang digunakan untuk menghapus atau membatalkan perjanjian. Apabila perjanjian telah diterima, maka dibutuhkan pembatalan dari kedua belah pihak apabila pembatalan diinginkan.

h. Fungsi *Rate Appointment* (**SKPL-SOS-011-08**). Merupakan fungsi yang digunakan untuk *merating* appointment yang telah dilakukan.

12. Fungsi *Manage Comment* (SKPL-SOS-012).

Merupakan fungsi yang digunakan untuk mengelola laporan. Fungsi ini dapat diakses oleh semua pengguna situs SOS

Fungsi *Manage Comment* mencakup :

a. Fungsi *Insert Comment* (**SKPL-SOS-012-01**). Merupakan fungsi yang digunakan untuk menambahkan komentar ke produk.

b. Fungsi *Delete Comment* (**SKPL-SOS-012-02**). Merupakan fungsi yang digunakan untuk menghapus komentar yang telah ditambahkan ke produk. Penghapusan komentar hanya dapat dilakukan oleh pengguna yang memberikan komentar tersebut.

c. Fungsi *Show Comment* (**SKPL-SOS-012-03**). Merupakan fungsi yang digunakan untuk

menampilkan semua komentar yang terdapat dalam produk tersebut.

13. Fungsi Show Notification (SKPL-SOS-013).

Merupakan fungsi yang digunakan untuk menampilkan semua pemberitahuan yang dimiliki oleh pengguna. Fungsi ini dapat diakses oleh semua pengguna situs SOS.

14. Fungsi Write Contact Us (SKPL-SOS-014).

Merupakan fungsi yang digunakan untuk menuliskan pesan kepada admin. Fungsi ini hanya dapat diakses oleh pengguna yang bukan termasuk administrator

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SOS adalah sebagai berikut :

1. Memahami cara pengoperasian aplikasi SOS sesuai dengan statusnya.
2. Dapat menggunakan internet untuk mengakses aplikasi web SOS.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SOS tersebut adalah :

1. Kebijaksanaan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak SOS.

2. Keterbatasan perangkat keras

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

Program Studi Teknik Informatika	SKPL – SPP	18/ 50
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		

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada komputer(PC, laptop) dan perangkat mobile (Smart Phone, Pocket PC, PDA, dll) yang memiliki browser.

3 Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak SOS 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 form-form.

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SOS adalah:

1. Mouse, digunakan untuk mengenali input yang dilakukan oleh pengguna yang berkaitan dengan event click.
2. Keyboard, digunakan untuk mengenali input yang dilakukan oleh pengguna untuk menginputkan data berupa karakter, teks, ataupun menu pull down.
3. Monitor, digunakan untuk menampilkan aplikasi SOS kepada pengguna.
4. Perangkat Mobile yang memiliki web browser

3.1.3 Antarmuka perangkat lunak

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

1. Nama : MySql
Sumber : Microsoft
Sebagai database management system (DBMS) yang digunakan untuk menyimpan data di sisi server.
2. Nama : CodeIgniter 2.0.3
Sumber : CodeIgniter
Sebagai framework yang dibutuhkan untuk membuat SOS versi Web.
3. Nama : jQuery Mobile 1.2.0
Sumber : jQuery
Sebagai framework yang dibutuhkan untuk membuat SOS versi Mobile.
4. Nama : Google API v.3
Sumber : Google Inc
Sebagai API yang digunakan untuk mengakses Google Map.
5. Nama : Windows 7 32 bits
Sumber : Microsoft
Sebagai sistem operasi yang digunakan untuk menjalankan browser.
6. Nama : Apache
Sumber : Apache Group
Sebagai web server untuk SOS.
7. Nama : Google Chrome
Sumber : Google Inc

Sebagai browser standar untuk menjalankan perangkat web.

8. Nama : Opera Mobile Emulator 12.0

Sumber : Opera

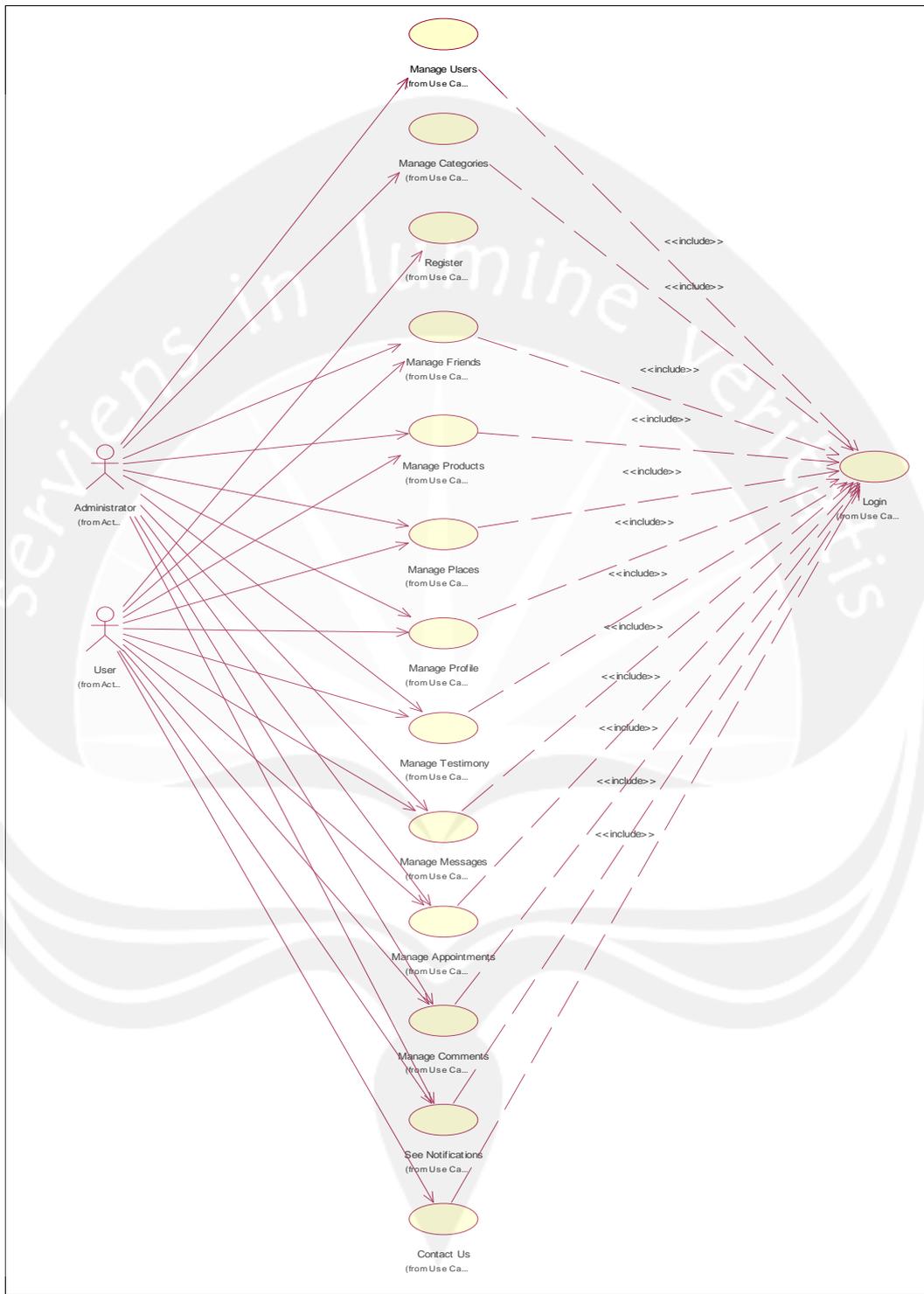
Sebagai browser mobile untuk menjalankan perangkat web.

3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi yang digunakan dalam SOS menggunakan protocol HTTP.

3.2 Kebutuhan fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram



Gambar 2 Use Case Diagram Situs SOS

4 Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

4.1.1 Use case Spesification : *Login*

1. Brief Description

Use Case ini digunakan oleh aktor untuk memperoleh akses ke system sesuai dengan status yang dimiliki. Login didasarkan pada sebuah id unik yaitu username dan password yang berupa rangkaian karakter.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login
2. Sistem menampilkan antarmuka untuk login
3. Aktor memasukkan id dan password
4. Sistem memeriksa id dan password yang diinputkan aktor

E-1 Password atau username tidak sesuai

5. Sistem memberikan akses ke aktor sesuai dengan status yang dimiliki aktor tersebut
6. Use Case ini selesai

5. Alternative Flow

none

6. Error Flow

E-1 Password atau username tidak sesuai

1. Sistem menampilkan peringatan bahwa username atau password tidak sesuai
2. Kembali ke Basic Flow langkah ke 3

7. PreConditions

none

Program Studi Teknik Informatika	SKPL – SPP	23/ 50
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		

8. PostConditions

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

4.1.2 Use case Spesification : Register

1. Brief Description

Use Case ini digunakan oleh aktor untuk mendaftarkan diri sebagai pengguna di situs SOS.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan register
2. Sistem menampilkan antarmuka untuk register
3. Aktor memasukkan data yang dimiliki
4. Sistem memeriksa data yang diinputkan aktor
 - E-1 Data yang diinputkan memiliki format yang salah atau tidak lengkap
5. Sistem menambahkan data user yang baru ke database
6. Use Case selesai

5. Alternative Flow

none

6. Error Flow

E-1 Data yang diinputkan memiliki format yang salah atau tidak lengkap

1. Sistem menampilkan peringatan bahwa data yang diinputkan memiliki format yang salah atau tidak lengkap
2. Kembali ke Basic Flow langkah ke 3

7. PreConditions

Program Studi Teknik Informatika	SKPL – SPP	24/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Use Case Login telah dilakukan

8. PostConditions

1. Aktor telah terdaftar dalam database dan dapat melakukan fungsi login dengan username dan password yang diisikan.

4.1.3 Use case Spesification : Manage Users

1. Brief Description

Use Case ini digunakan oleh administrator untuk mengelola data pengguna. Aktor dapat melakukan show users, show detail user, atau delete user.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan manage users
2. Sistem memberikan pilihan untuk melakukan show users, show detail user, atau delete user
3. Aktor memilih untuk melakukan show user
 - A-1 Aktor memilih untuk melakukan show detail user
 - A-2 Aktor memilih untuk melakukan delete user
4. Sistem menampilkan semua user yang ada di database
5. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan show detail user
1. Sistem menampilkan semua user yang ada di database
 2. Aktor memilih user yang akan ditampilkan secara detil

Program Studi Teknik Informatika	SKPL – SPP	25/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Sistem menampilkan detail user yang dipilih oleh aktor
 4. Berlanjut ke Basic Flow langkah ke 5
- A-2 Aktor memilih untuk melakukan delete user
1. Sistem menampilkan semua user yang ada di database
 2. Aktor memilih user yang akan di delete
 3. Aktor meminta sistem untuk mendelete user yang dipilih
 4. Sistem mendelete user yang dipilih oleh aktor
 5. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan
2. Aktor telah memasuki sistem sebagai Administrator

8. PostConditions

1. Data user di database telah terupdate

4.1.4 Use case Spesification : *Manage Categories*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data kategori. Aktor dapat melakukan insert category, edit category, delete category, atau show category.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan manage categories
2. Sistem memberikan pilihan untuk melakukan insert category, edit category, delete category, atau show category

Program Studi Teknik Informatika	SKPL – SPP	26/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Aktor memilih untuk melakukan insert category
 - A-1 Aktor memilih untuk melakukan edit category
 - A-2 Aktor memilih untuk melakukan delete category
 - A-3 Aktor memilih untuk melakukan show category
4. Aktor menginputkan data kategori baru
5. Aktor meminta sistem untuk menyimpan data kategori yang telah diinputkan
6. Sistem mengecek data kategori yang telah diinputkan
 - E-1 Data kategori yang diinputkan aktor tidak lengkap
 - E-2 Data kategori yang diinputkan aktor sudah ada di database
7. Sistem menyimpan data kategori ke database
8. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan edit category
 1. Sistem menampilkan data kategori
 2. Aktor mengedit data kategori yang sudah ditampilkan
 3. Aktor meminta sistem untuk menyimpan data kategori yang telah diedit
 4. Sistem melakukan pengecekan terhadap data kategori yang telah diedit
 - E-3 Data kategori tidak lengkap
 - E-4 Data kategori yang diedit telah ada di database
 5. Sistem menyimpan data perabot yang telah diedit ke database
 6. Berlanjut ke Basic Flow langkah ke 8
- A-2 Aktor memilih untuk melakukan delete category

Program Studi Teknik Informatika	SKPL – SPP	27/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Sistem menampilkan data kategori
 2. Aktor memilih kategori yang akan dihapus
 3. Aktor meminta sistem untuk menghapus kategori yang dipilih
 4. Sistem menghapus kategori yang dipilih oleh aktor
 5. Berlanjut ke Basic Flow langkah ke 8
- A-3 Aktor memilih untuk melakukan show category
1. Sistem menampilkan data kategori
 2. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

- E-1 Data kategori yang diinputkan aktor salah atau tidak lengkap
1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah atau tidak lengkap
 2. Kembali ke Basic Flow Langkah ke 4
- E-2 Data kategori yang diinputkan aktor sudah ada di database
1. Sistem memberikan pesan peringatan bahwa data yang diinputkan sudah ada di database
 2. Kembali ke Basic Flow Langkah ke 4
- E-3 Data kategori tidak lengkap
1. Sistem memberikan pesan peringatan bahwa data yang diedit tidak lengkap
 2. Kembali ke Alternative Flow A-1 Langkah ke 2
- E-4 Data kategori yang diinputkan aktor sudah ada di database
1. Sistem memberikan pesan peringatan bahwa data yang diinputkan sudah ada di database
 2. Kembali ke Basic Flow Langkah ke 4

7. PreConditions

1. Use Case Login telah dilakukan
2. Aktor telah memasuki sistem sebagai Administrator

Program Studi Teknik Informatika	SKPL – SPP	28/ 50
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		

8. PostConditions

1. Data kategori di database telah terupdate

4.1.5 Use case Spesification : *Manage Profile*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data profil yang dimiliki. Aktor dapat melakukan change username, change password, change profile picture atau show profile.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan manage profile
2. Sistem memberikan pilihan untuk melakukan change username, change password, change profile picture, edit profile, edit setting atau show profile
3. Aktor memilih untuk melakukan change username
 - A-1 Aktor memilih untuk melakukan change password
 - A-2 Aktor memilih untuk melakukan change profile picture
 - A-3 Aktor memilih untuk melakukan edit profile
 - A-4 Aktor memilih untuk melakukan edit setting
 - A-5 Aktor memilih untuk melakukan show profile
4. Aktor menginputkan username baru
5. Aktor meminta sistem untuk menyimpan username yang telah diinputkan
6. Sistem mengecek username yang telah diinputkan
 - E-1 Username tidak diisi oleh aktor

Program Studi Teknik Informatika	SKPL – SPP	29/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

E-2 Username yang diinputkan aktor sudah ada di database

7. Sistem menyimpan perubahan username ke database
8. Use Case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan change password

1. Aktor menginputkan password lama dan password baru
2. Aktor meminta sistem untuk menyimpan password yang telah diedit
3. Sistem melakukan pengecekan terhadap password lama dan password baru

E-3 Password lama yang diisikan salah

E-4 Password baru tidak diisikan

4. Sistem menyimpan perubahan password ke database
5. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan change profile picture

1. Aktor memilih untuk mengupload gambar sebagai profile picture

A-5 Aktor memilih file gambar yang telah diupload sebelumnya sebagai profile picture

2. Aktor memilih file gambar yang akan di upload
3. Aktor meminta sistem untuk mengupload file gambar yang dipilih
4. Sistem mengupload file gambar yang dipilih oleh aktor
5. Sistem mengubah profile picture aktor dengan file gambar yang baru diupload
6. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk melakukan edit profile

1. Sistem menampilkan data profile yang dimiliki aktor

Program Studi Teknik Informatika	SKPL – SPP	30/ 50
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. Aktor mengedit data profile yang dimiliki
3. Aktor meminta sistem untuk menyimpan data profile yang telah diedit
4. Sistem melakukan pengecekan terhadap data profile yang telah diedit

E-5 Data profile yang diisikan tidak lengkap

5. Sistem menyimpan data profile yang telah diedit ke database
6. Berlanjut ke Basic Flow langkah ke 8

A-4 Aktor memilih untuk melakukan edit setting

1. Sistem menampilkan data setting yang dimiliki aktor
2. Aktor mengedit data setting yang dimiliki
3. Aktor meminta sistem untuk menyimpan data setting yang telah diedit
4. Sistem menyimpan data setting yang telah diedit ke database

5. Berlanjut ke Basic Flow langkah ke 8

A-5 Aktor memilih untuk melakukan show profile

1. Sistem menampilkan data profile yang dimiliki aktor
2. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Username tidak diisi oleh aktor

1. Sistem memberikan pesan peringatan bahwa username tidak diisi oleh aktor
2. Kembali ke Basic Flow Langkah ke 4

E-2 Username yang diinputkan aktor sudah ada di database

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan sudah ada di database
2. Kembali ke Basic Flow Langkah ke 4

E-3 Password lama yang diisikan salah

Program Studi Teknik Informatika	SKPL – SPP	31/50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Sistem memberikan pesan peringatan bahwa password lama yang diisikan salah
 2. Kembali ke Alternative Flow A-1 Langkah ke 1
- E-4 Password baru tidak diisikan
1. Sistem memberikan pesan peringatan bahwa password baru tidak diisikan
 2. Kembali ke Alternative Flow A-1 Langkah ke 1
- E-5 Data profile yang diisikan tidak lengkap
1. Sistem memberikan pesan peringatan bahwa data profile yang diisikan tidak lengkap
 2. Kembali ke Alternative Flow A-1 Langkah ke 1

7. PreConditions

1. Use Case Login telah dilakukan

8. PostConditions

1. Data profile di database telah terupdate

4.1.6 Use case Spesification : *Manage Friends*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data teman yang dimiliki. Aktor dapat melakukan insert friend, accept friend, atau reject friend.

2. Primary Actor

1. Bagian Pembelian

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data teman
2. Sistem memberikan pilihan untuk melakukan insert friend, accept friend, reject friend, atau delete friend
3. Aktor memilih untuk melakukan insert friend

Program Studi Teknik Informatika	SKPL – SPP	32/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- A-1 Aktor memilih untuk melakukan accept friend
- A-2 Aktor memilih untuk melakukan reject friend
- A-3 Aktor memilih untuk melakukan delete friend
- 4. Aktor menginputkan nama teman yang akan ditambahkan
- 5. Aktor meminta sistem untuk mencari nama teman yang telah diinputkan
- 6. Sistem mencari nama teman yang telah diinputkan
 - E-1 Nama teman yang diinputkan aktor tidak terdapat di database
- 7. Sistem menampilkan data-data pengguna dengan nama sesuai dengan inputan aktor
- 8. Aktor memilih teman yang akan ditambahkan
- 9. Sistem menyimpan data teman yang ditambahkan ke dalam database
- 10. Sistem mengirimkan notifikasi ke tabel notifikasi teman yang akan ditambahkan aktor
- 11. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan accept friend
 - 1. Sistem menampilkan notifikasi teman yang mengirimkan permintaan pertemanan
 - 2. Aktor memilih untuk menerima permintaan pertemanan
 - 3. Sistem menyimpan data teman yang ditambahkan ke dalam database
 - 4. Sistem mengirimkan notifikasi ke tabel notifikasi teman yang diterima aktor
 - 5. Berlanjut ke Basic Flow langkah ke 11
- A-2 Aktor memilih untuk melakukan reject friend

1. Sistem menampilkan notifikasi teman yang mengirimkan permintaan pertemanan
 2. Aktor memilih untuk menolak permintaan pertemanan
 3. Sistem menyimpan data teman yang ditambahkan ke dalam database
 4. Sistem mengirimkan notifikasi ke tabel notifikasi teman yang diterima aktor
 5. Berlanjut ke Basic Flow langkah ke 11
- A-3 Aktor memilih untuk melakukan delete friend
1. Sistem menampilkan data teman yang dipilih
 2. Aktor memilih untuk mendelete pertemanan yang telah dilakukan
 3. Sistem menghapus hubungan pertemanan yang terdapat di database
 4. Berlanjut ke Basic Flow langkah ke 11

6. Error Flow

- E-1 Nama teman yang diinputkan aktor tidak terdapat di database
1. Sistem tidak menampilkan data teman yang diminta oleh aktor
 2. Kembali ke Basic Flow Langkah ke 4

7. PreConditions

1. Use Case Login telah dilakukan

8. PostConditions

1. Data friendship dan notification di database telah terupdate

4.1.7 Use case Spesification : *Manage Products*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data produk yang dimiliki. Aktor dapat melakukan insert product, edit product, delete product, show products,

Program Studi Teknik Informatika	SKPL – SPP	34/ 50
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		

show detail product, insert photo product, delete photo product, atau promote product.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan *Manage Products*
2. Sistem memberikan pilihan untuk melakukan insert product, edit product, delete product, show products, show detail product, insert photo product, delete photo product, atau promote product
3. Aktor memilih untuk melakukan insert product
 - A-1 Aktor memilih untuk melakukan edit product
 - A-2 Aktor memilih untuk melakukan delete product
 - A-3 Aktor memilih untuk melakukan show products
 - A-4 Aktor memilih untuk melakukan show detail product
 - A-5 Aktor memilih untuk melakukan insert photo product
 - A-6 Aktor memilih untuk melakukan delete photo product
 - A-7 Aktor memilih untuk melakukan promote product
4. Aktor menginputkan data produk
5. Aktor meminta sistem untuk menyimpan data produk yang telah diinputkan
6. Sistem mengecek data produk yang telah diinputkan
 - E-1 Data produk yang diinputkan aktor salah atau tidak lengkap

Program Studi Teknik Informatika	SKPL – SPP	35/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

7. Sistem menyimpan data produk ke database
8. Use Case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit product

1. Sistem menampilkan data produk yang dimiliki aktor
2. Aktor mengedit data produk yang sudah ditampilkan
3. Aktor meminta sistem untuk menyimpan data produk yang telah diedit
4. Sistem melakukan pengecekan terhadap data produk yang telah diedit

E-2 Data produk yang telah diedit salah atau tidak lengkap

5. Sistem menyimpan data produk yang telah diedit ke database
6. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan delete product

1. Sistem menampilkan data produk
2. Aktor memilih produk yang akan dihapus
3. Aktor meminta sistem untuk menghapus produk yang dipilih
4. Sistem menghapus produk yang dipilih oleh aktor
5. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk melakukan show products

1. Sistem menampilkan data produk yang dimiliki aktor
2. Berlanjut ke Basic Flow langkah ke 8

A-4 Aktor memilih untuk melakukan show detail product

1. Sistem menampilkan data produk yang dimiliki aktor
2. Aktor memilih produk yang ingin ditampilkan detailnya

Program Studi Teknik Informatika	SKPL – SPP	36/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Berlanjut ke Basic Flow langkah ke 8

A-5 Aktor memilih untuk melakukan insert photo product

1. Aktor memilih file gambar yang akan di upload
2. Aktor meminta sistem untuk mengupload file gambar yang dipilih
3. Sistem mengupload file gambar yang dipilih oleh aktor
4. Berlanjut ke Basic Flow langkah ke 8

A-6 Aktor memilih untuk melakukan delete photo product

1. Aktor memilih gambar yang akan di delete
2. Aktor meminta sistem untuk mendelete file gambar yang dipilih
3. Sistem mendelete file gambar yang dipilih oleh aktor
4. Berlanjut ke Basic Flow langkah ke 8

A-7 Aktor memilih untuk melakukan promote product

1. Sistem menampilkan halaman untuk menambahkan promote product
2. Aktor menginputkan teks yang akan digunakan untuk mempromosikan produk
3. Aktor meminta sistem untuk menyimpan teks yang telah diinputkan
4. Sistem melakukan pengecekan terhadap data promote yang telah diinputkan

E-3 Data promote tidak diisi

5. Sistem menyimpan teks yang diinputkan oleh aktor
6. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Data produk yang diinputkan aktor salah atau tidak lengkap

Program Studi Teknik Informatika	SKPL – SPP	37/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah atau tidak lengkap
 2. Kembali ke Basic Flow Langkah ke 4
- E-2 Data produk yang telah diedit salah atau tidak lengkap
1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah atau tidak lengkap
 2. Kembali ke Alternative Flow A-1 Langkah ke 2
- E-3 Data promote tidak diisi
1. Sistem memberikan pesan peringatan bahwa data yang belum diisi
 2. Kembali ke Alternative Flow A-7 Langkah ke 2

7. PreConditions

1. Use Case Login telah dilakukan

8. PostConditions

1. Data product di database telah terupdate

4.1.8 Use case Spesification : *Manage Places*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data tempat yang dimilikinya. Aktor dapat melakukan insert place, delete place, atau show place.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan *Manage Places*
2. Sistem memberikan pilihan untuk melakukan insert place, delete place, atau show place
3. Aktor memilih untuk melakukan insert place
 - A-1 Aktor memilih untuk melakukan delete place
 - A-2 Aktor memilih untuk melakukan show place

Program Studi Teknik Informatika	SKPL – SPP	38/ 50
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. Aktor menginputkan data tempat yang baru
5. Aktor meminta sistem untuk menyimpan data tempat yang telah diinputkan
6. Sistem mengecek data user yang telah diinputkan
E-1 Data tempat yang diinputkan aktor salah atau tidak lengkap
7. Sistem menyimpan data user ke database
8. Use Case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan delete place

1. Sistem menampilkan data tempat yang ada
2. Aktor memilih tempat yang akan dihapus
3. Aktor meminta sistem untuk menghapus data tempat yang dipilih
4. Sistem menghapus data tempat yang telah dipilih
5. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan show place

1. Sistem menampilkan data tempat yang ada
2. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Data user yang diinputkan aktor salah atau tidak lengkap

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah atau tidak lengkap
2. Kembali ke Basic Flow Langkah ke 4

7. PreConditions

1. Use Case Login telah dilakukan

8. PostConditions

1. Data place di database telah terupdate

4.1.9 Use case Spesification : *Manage Testimony*

1. Brief Description

Program Studi Teknik Informatika	SKPL – SPP	39/ 50
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		

Use Case ini digunakan oleh aktor untuk mengelola testimonial. Aktor dapat melakukan insert testimony, edit testimony, delete testimony, atau show testimony.

2. Primary Actor

1. Shopkeeper

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan *Manage Testimony*
2. Sistem memberikan pilihan untuk melakukan insert testimony, edit testimony, delete testimony, atau show testimony
3. Aktor memilih untuk melakukan insert testimony
 - A-1 Aktor memilih untuk melakukan edit testimony
 - A-2 Aktor memilih untuk melakukan delete testimony
 - A-3 Aktor memilih untuk melakukan show testimony
4. Aktor menginputkan testimony kepada user
5. Sistem mengecek testimony yang telah diinputkan
 - E-1 Testimony tidak diisi
6. Aktor meminta sistem untuk testimony yang telah diinputkan
7. Sistem menyimpan testimony ke database
8. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan edit testimony
1. Sistem menampilkan testimony
 2. Aktor mengedit testimony yang sudah ditampilkan
 3. Aktor meminta sistem untuk menyimpan testimony yang telah diedit

Program Studi Teknik Informatika	SKPL – SPP	40/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Sistem mengecek testimony yang telah diinputkan

E-2 Testimony tidak diisi

5. Sistem menyimpan testimony yang telah diedit ke database

6. Berlanjut ke Basic Flow langkah ke 8

A-2 Aktor memilih untuk melakukan delete testimony

1. Sistem menampilkan testimony

2. Aktor memilih testimony yang akan dihapus

3. Aktor meminta sistem untuk menghapus testimony yang dipilih

4. Sistem menghapus testimony yang dipilih oleh aktor

5. Berlanjut ke Basic Flow langkah ke 8

A-3 Aktor memilih untuk melakukan show testimony

1. Sistem menampilkan semua testimony yang dimiliki aktor

2. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

E-1 Testimony tidak diisi

1. Sistem memberikan pesan peringatan bahwa testimony belum diisi

2. Kembali ke Basic Flow Langkah ke 4

E-1 Testimony tidak diisi

1. Sistem memberikan pesan peringatan bahwa testimony belum diisi

2. Kembali ke Alternative Flow A-1 Langkah ke 2

7. PreConditions

1. Use Case Login telah dilakukan

2. Aktor telah memasuki sistem sebagai Shopkeeper

8. PostConditions

1. Gambar perabot di database telah terupdate

Program Studi Teknik Informatika	SKPL – SPP	41/ 50
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		

4.1.10 Use case Spesification : *Manage Messages*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola pesan. Aktor dapat melakukan insert message, show message atau show message contents.

2. Primary Actor

1. Kasir

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan *Manage Messages*
2. Sistem memberikan pilihan untuk melakukan insert message, show messages, atau show message content
3. Aktor memilih untuk melakukan insert message
 - A-1 Aktor memilih untuk melakukan show messages
 - A-2 Aktor memilih untuk melakukan show message content
4. Aktor menginputkan pesan
5. Aktor meminta sistem untuk menyimpan pesan yang telah diinputkan
6. Sistem mengecek pesan yang telah diinputkan
 - E-1 Data pesan yang diinputkan aktor salah atau tidak lengkap
7. Sistem menyimpan data pesan ke database
8. Sistem mengirim notifikasi ke teman yang ditujukan pesan
9. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan show message
1. Sistem menampilkan semua pesan yang dimiliki aktor
 2. Berlanjut ke Basic Flow langkah ke 9

Program Studi Teknik Informatika	SKPL – SPP	42/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

A-2 Aktor memilih untuk melakukan show message content

1. Sistem menampilkan semua pesan yang dimiliki aktor
2. Aktor memilih pesan yang ingin ditampilkan detilnya
3. Berlanjut ke Basic Flow langkah ke 9

6. Error Flow

E-1 Data pesan yang diinputkan aktor salah atau tidak lengkap

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah atau tidak lengkap
2. Kembali ke Basic Flow Langkah ke 4

7. PreConditions

1. Use Case Login telah dilakukan

8. PostConditions

1. Data pesan di database telah terupdate

4.1.11 Use case Spesification : *Manage Appointments*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola perjanjian yang dimiliki. Aktor dapat melakukan insert appointment, accept appointment, reject appointment, send product or money, accept product or money, edit appointment, delete appointment, atau rate appointment.

2. Primary Actor

1. Bagian Penjualan

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan *Manage Appointments*
2. Sistem memberikan pilihan untuk melakukan insert appointment, accept appointment, reject

Program Studi Teknik Informatika	SKPL – SPP	43/ 50
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		

appointment, send product or money, accept product or money, edit appointment, delete appointment, atau rate appointment

3. Aktor memilih untuk melakukan insert appointment
 - A-1 Aktor memilih untuk melakukan accept appointment
 - A-2 Aktor memilih untuk melakukan reject appointment
 - A-3 Aktor memilih untuk melakukan send product or money
 - A-4 Aktor memilih untuk melakukan accept product or money
 - A-5 Aktor memilih untuk melakukan edit appointment
 - A-6 Aktor memilih untuk melakukan delete appointment
 - A-7 Aktor memilih untuk melakukan rate appointment
4. Aktor menginputkan data appointment
5. Aktor meminta sistem untuk menyimpan data appointment
6. Sistem mengecek data penjualan partai yang telah diinputkan
 - E-1 Data appointment yang diinputkan aktor salah atau tidak lengkap
7. Sistem menyimpan data appointment ke database
8. Sistem mengirimkan notifikasi kepada teman yang dikirimkan appointment
9. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan accept appointment
1. Sistem menampilkan appointment yang dimiliki
 2. Aktor memilih untuk menerima appointment yang ditawarkan

Program Studi Teknik Informatika	SKPL – SPP	44/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Sistem menyimpan perubahan appointment di database
 4. Sistem mengirimkan notifikasi kepada pengirim appointment bahwa appointment telah diterima
 5. Berlanjut ke Basic Flow langkah ke 9
- A-2 Aktor memilih untuk melakukan accept appointment
1. Sistem menampilkan appointment yang dimiliki
 2. Aktor memilih untuk menolak appointment yang ditawarkan
 3. Sistem menyimpan perubahan appointment di database
 4. Sistem mengirimkan notifikasi kepada pengirim appointment bahwa appointment telah ditolak
 5. Berlanjut ke Basic Flow langkah ke 9
- A-3 Aktor memilih untuk melakukan send product or money
1. Sistem menampilkan appointment yang dimiliki
 2. Aktor memilih untuk memberitahukan bahwa produk atau uang telah dikirimkan
 3. Sistem menyimpan perubahan appointment di database
 4. Sistem mengirimkan notifikasi kepada pengirim appointment bahwa produk atau uang telah dikirimkan
 5. Berlanjut ke Basic Flow langkah ke 9
- A-4 Aktor memilih untuk melakukan accept product or money
1. Sistem menampilkan appointment yang dimiliki
 2. Aktor memilih untuk memberitahukan bahwa produk atau uang dikirim telah diterima
 3. Sistem menyimpan perubahan appointment di database

4. Sistem mengirimkan notifikasi kepada pengirim appointment bahwa produk atau uang dikirim telah diterima
 5. Berlanjut ke Basic Flow langkah ke 9
- A-5 Aktor memilih untuk melakukan edit appointment
1. Sistem menampilkan appointment yang dimiliki
 2. Aktor mengedit appointment yang sudah ditampilkan
 3. Aktor meminta sistem untuk menyimpan appointment yang telah diedit
 4. Sistem mengecek data appointment yang telah diedit
 - E-2 Data appointment yang diedit aktor salah atau tidak lengkap
 5. Sistem menyimpan appointment yang telah diedit ke database
 6. Berlanjut ke Basic Flow langkah ke 9
- A-6 Aktor memilih untuk melakukan delete appointment
1. Sistem menampilkan appointment yang dimiliki
 2. Aktor memilih appointment yang akan di delete
 3. Aktor meminta sistem untuk mendelete appointment yang dipilih
 4. Sistem mendelete appointment yang dipilih oleh aktor
 5. Sistem mengirimkan notifikasi kepada pengirim appointment bahwa appointment akan di delete
 6. Berlanjut ke Basic Flow langkah ke 9
- A-7 Aktor memilih untuk melakukan *rate* appointment
1. Sistem menampilkan appointment yang dimiliki
 2. Aktor memilih appointment yang akan *rate*
 3. Aktor meminta *rating* yang akan diberikan
 4. Sistem menyimpan *rating* yang telah diberikan aktor
 5. Berlanjut ke Basic Flow langkah ke 9

6. Error Flow

E-1 Data appointment yang diinputkan aktor salah atau tidak lengkap

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah atau tidak lengkap
2. Kembali ke Basic Flow Langkah ke 4

E-2 Data appointment yang diedit aktor salah atau tidak lengkap

1. Sistem memberikan pesan peringatan bahwa data yang diedit salah atau tidak lengkap
2. Kembali ke Alternative Flow A-5 Langkah ke 2

7. PreConditions

1. Use Case Login telah dilakukan

8. PostConditions

1. Data appointment dan notification di database telah terupdate

4.1.12 Use case Spesification : *Manage Comment*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola komentar pada produk. Aktor dapat melakukan insert comment, delete comment, atau show comment.

2. Primary Actor

1. Bagian Penjualan

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan *Manage Comment*
2. Sistem memberikan pilihan untuk melakukan insert comment, delete comment, atau show comment
3. Aktor memilih untuk melakukan insert comment
A-1 Aktor memilih untuk melakukan delete comment

Program Studi Teknik Informatika	SKPL – SPP	47/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

- A-2 Aktor memilih untuk melakukan show comment
- 4. Aktor menginputkan komentar
- 5. Aktor meminta sistem untuk menyimpan komentar
- 6. Sistem mengecek komentar yang telah diinputkan
 - E-1 Komentar tidak diisi
- 7. Sistem menyimpan komentar ke database
- 8. Use Case selesai

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan delete comment
 - 1. Sistem menampilkan data komentar
 - 2. Aktor memilih komentar yang akan dihapus
 - 3. Sistem menghapus komentar yang telah dipilih
 - 4. Berlanjut ke Basic Flow langkah ke 8
- A-2 Aktor memilih untuk melakukan show comment
 - 1. Sistem menampilkan semua komentar pada produk
 - 2. Berlanjut ke Basic Flow langkah ke 8

6. Error Flow

- E-1 Komentar tidak diisi
 - 1. Sistem memberikan pesan peringatan bahwa data komentar belum diisi
 - 2. Kembali ke Basic Flow Langkah ke 4

7. PreConditions

- 1. Use Case Login telah dilakukan

8. PostConditions

- 1. Data comment di database telah terupdate

4.1.13 Use case Spesification : Show Notification

1. Brief Description

Use Case ini digunakan oleh aktor untuk menampilkan semua pemberitahuan yang dimiliki.

2. Primary Actor

- 1. Manager

3. Supporting Actor

Program Studi Teknik Informatika	SKPL – SPP	48/ 50
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		

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan Show Notification
2. Sistem menampilkan semua notification yang ada
3. Use Case selesai

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan

8. PostConditions

1. Data notification di database telah terupdate

4.1.14 Use case Spesification : Write Contact Us

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengirim pesan kepada administrator.

2. Primary Actor

1. Pengguna selain administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih menu contact us
2. Sistem menampilkan halaman contact us
3. Aktor menginputkan data contact us
4. Aktor meminta sistem untuk menyimpan data yang telah diinputkan
5. Sistem mengecek data penjualan partai yang telah diinputkan

E-1 Data appointment yang diinputkan aktor tidak lengkap

Program Studi Teknik Informatika	SKPL – SPP	49/ 50
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

6. Sistem menyimpan data yang telah diinputkan user

7. Use Case selesai

5. Alternative Flow

none

6. Error Flow

E-1 Data contact us yang diinputkan aktor tidak lengkap

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan tidak lengkap

2. Kembali ke Basic Flow Langkah ke 3

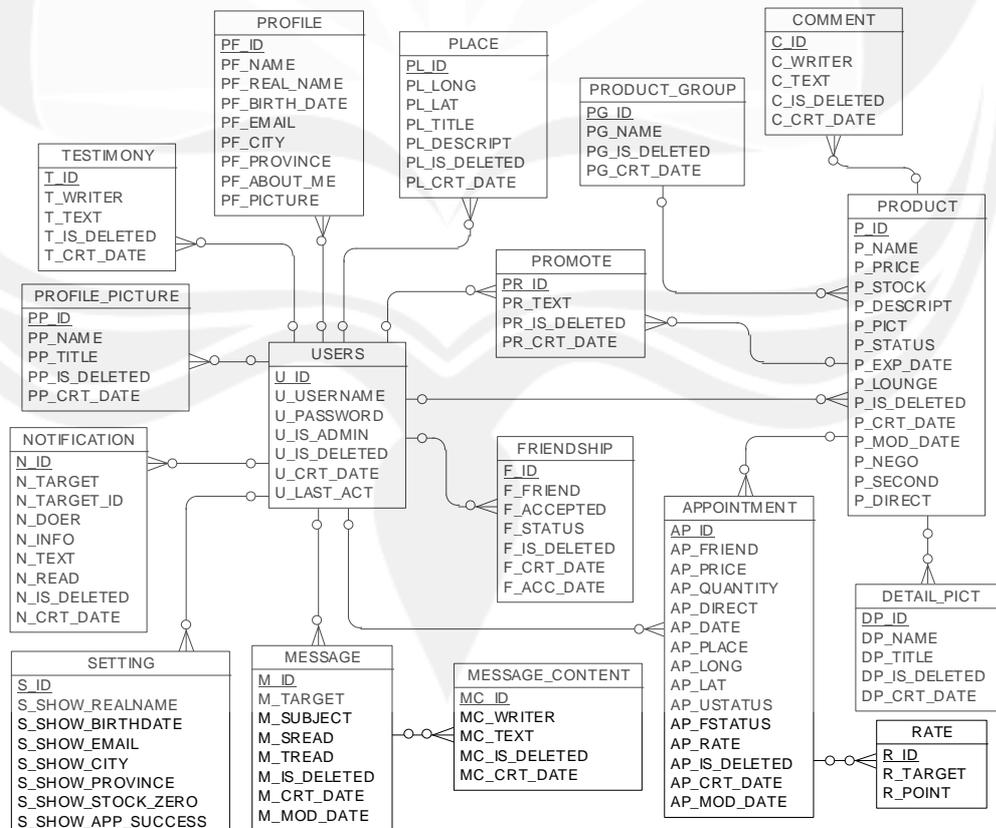
7. PreConditions

1. Use Case Login telah dilakukan

8. PostConditions

1. Data message di database telah terupdate

5 Entity Relationship Diagram (ERD)



Gambar 3 Entity Relationship Diagram

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SOS

(Social Online Shop)

Dipersiapkan oleh:

Lilyantika / 5948

Program Studi Teknik Informatika - Fakultas
Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL-ATK		1/152
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	

INDEX TGL	A	B	c	D	E	F	G
Ditulis oleh							
Diperik sa oleh							
Disetuj ui oleh							

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan	9
1.1	Tujuan	9
1.2	Ruang Lingkup	9
1.3	Definisi dan Akronim	9
1.4	Referensi	10
2	Perancangan Sistem	11
2.1	Perancangan Arsitektur	11
2.2	Perancangan Rinci	14
2.2.1	Sequence Diagram.....	14
2.2.1.1	Login	14
2.2.1.2	Register	15
2.2.1.3	Change Password	16
2.2.1.4	Change Username	17
2.2.1.5	Change Profile Picture	18
2.2.1.6	Edit Profile	19
2.2.1.7	Edit Setting	20
2.2.1.8	Upload Profile Picture	21
2.2.1.9	Show Notification	21
2.2.1.10	Pengelolaan Testimony	22
2.2.1.10.1	Insert Testimony.....	22
2.2.1.10.2	Edit Testimony.....	22
2.2.1.10.3	Delete Testimony.....	23
2.2.1.11	Pengelolaan User	24
2.2.1.11.1	Search User By Profile Name.....	24
2.2.1.11.2	Search User By Realname.....	24
2.2.1.11.3	Search User By Username.....	25
2.2.1.11.4	Show Detail User.....	26
2.2.1.11.5	Delete User.....	26
2.2.1.12	Pengelolaan Category	27
2.2.1.12.1	Insert Category.....	27
2.2.1.12.2	Edit Category.....	27
2.2.1.12.3	Delete Category.....	28
2.2.1.13	Pengelolaan Product	29
2.2.1.13.1	Show Product.....	29
2.2.1.13.2	Insert Product.....	30
2.2.1.13.3	Edit Product.....	31
2.2.1.13.4	Delete Product.....	31
2.2.1.13.5	Detail Product.....	32
2.2.1.14	Pengelolaan Comment	33
2.2.1.14.1	Insert Comment.....	33
2.2.1.14.2	Delete Comment.....	33
2.2.1.15	Go to Showroom	34
2.2.1.16	Go to Lounge	35
2.2.1.17	Insert Promote	36
2.2.1.18	Pengelolaan Product Picture	36
2.2.1.18.1	Insert Product Picture.....	36
2.2.1.18.2	Change Product Picture.....	37
2.2.1.18.3	Delete Product Picture.....	38
2.2.1.19	Pengelolaan Friendship	39
2.2.1.19.1	Insert Friendship.....	39
2.2.1.19.2	Accept Friendship.....	39
2.2.1.19.3	Reject Friendship.....	40
2.2.1.19.4	Delete Friendship.....	41

2.2.1.19.5	Show Friend.....	42
2.2.1.20	Contact Us	43
2.2.1.21	Pengelolaan Message	43
2.2.1.21.1	List Message.....	43
2.2.1.21.2	Insert Message.....	44
2.2.1.21.3	Detail Message.....	45
2.2.1.22	Pengelolaan Place	45
2.2.1.22.1	Insert Place.....	45
2.2.1.22.2	Delete Place.....	46
2.2.1.22.3	Show Place.....	47
2.2.1.23	Pengelolaan Appointment	48
2.2.1.23.1	Insert Appointment.....	48
2.2.1.23.2	Insert Time or Place Appointment.....	49
2.2.1.23.3	Edit Appointment.....	49
2.2.1.23.4	Approve Appointment.....	50
2.2.1.23.5	Reject Appointment.....	51
2.2.1.23.6	Send Product Or Money.....	52
2.2.1.23.7	Accept Product Or Money.....	53
2.2.1.23.8	Cancel Appointment.....	54
2.2.1.23.9	Do not Appointment.....	55
2.2.1.23.10	Show Appointment.....	56
2.2.1.23.11	Detail Appointment.....	57
2.2.1.23.12	Rate Appointment.....	57
2.2.2	Class Diagram.....	59
2.2.3	Spesifikasi Kelas Diagram.....	61
2.2.3.1	Spesifikasi Design Kelas Homepage	61
2.2.3.2	Spesifikasi Design Kelas User_manage	61
2.2.3.3	Spesifikasi Design Kelas User_detail	61
2.2.3.4	Spesifikasi Design Kelas User_home	62
2.2.3.5	Spesifikasi Design Kelas Profile_home	62
2.2.3.6	Spesifikasi Design Kelas Profile_edit	62
2.2.3.7	Spesifikasi Design Kelas Profile_editpics	62
2.2.3.8	Spesifikasi Design Kelas Notification_home	63
2.2.3.9	Spesifikasi Design Kelas Testimony_edit	63
2.2.3.10	Spesifikasi Design Kelas Setting_edit	63
2.2.3.11	Spesifikasi Design Kelas Category_home	64
2.2.3.12	Spesifikasi Design Kelas Category_edit	64
2.2.3.13	Spesifikasi Design Kelas Friendship_home	64
2.2.3.14	Spesifikasi Design Kelas Friendship_showfriend	64
2.2.3.15	Spesifikasi Design Kelas Product_home	65
2.2.3.16	Spesifikasi Design Kelas Product_insert	65
2.2.3.17	Spesifikasi Design Kelas Product_edit	65
2.2.3.18	Spesifikasi Design Kelas Product_detail	65
2.2.3.19	Spesifikasi Design Kelas Product_insertpic	66
2.2.3.20	Spesifikasi Design Kelas Lounge	66
2.2.3.21	Spesifikasi Design Kelas Showroom	66
2.2.3.22	Spesifikasi Design Kelas Promote_right	66
2.2.3.23	Spesifikasi Design Kelas Promote_insert	67
2.2.3.24	Spesifikasi Design Kelas Place_home	67
2.2.3.25	Spesifikasi Design Kelas Place_insert	67
2.2.3.26	Spesifikasi Design Kelas Message_list	67
2.2.3.27	Spesifikasi Design Kelas Message_insert	67
2.2.3.28	Spesifikasi Design Kelas Message_detail	68
2.2.3.29	Spesifikasi Design Kelas User_contactus	68
2.2.3.30	Spesifikasi Design Kelas Appointment_home	68
2.2.3.31	Spesifikasi Design Kelas Appointment_insert	68

2.2.3.32	Spesifikasi Design Kelas	Appointment_edit	69
2.2.3.33	Spesifikasi Design Kelas	Appointment_detail	69
2.2.3.34	Spesifikasi Design Kelas	Rate_insert	69
2.2.3.35	Spesifikasi Design Kelas	Home	69
2.2.3.36	Spesifikasi Design Kelas	Users	70
2.2.3.37	Spesifikasi Design Kelas	Category	72
2.2.3.38	Spesifikasi Design Kelas	Friendship	73
2.2.3.39	Spesifikasi Design Kelas	Product	74
2.2.3.40	Spesifikasi Design Kelas	Place	76
2.2.3.41	Spesifikasi Design Kelas	Message	77
2.2.3.42	Spesifikasi Design Kelas	Appointment	78
2.2.3.43	Spesifikasi Design Kelas	Users_model	79
2.2.3.44	Spesifikasi Design Kelas	Notification_model	82
2.2.3.45	Spesifikasi Design Kelas	Testimony_model	84
2.2.3.46	Spesifikasi Design Kelas	Category_model	85
2.2.3.47	Spesifikasi Design Kelas	Friendship_model	86
2.2.3.48	Spesifikasi Design Kelas	Product_model	88
2.2.3.49	Spesifikasi Design Kelas	Comment_model	92
2.2.3.50	Spesifikasi Design Kelas	Promote_model	92
2.2.3.51	Spesifikasi Design Kelas	Place_model	93
2.2.3.52	Spesifikasi Design Kelas	Message_model	94
2.2.3.53	Spesifikasi Design Kelas	Appointment_model	96
3	Deskripsi Dekomposisi		101
3.1	Dekomposisi Data		101
3.1.1	Deskripsi Entitas Tabel	Users	101
3.1.2	Deskripsi Entitas Tabel	Profile	102
3.1.3	Deskripsi Entitas Tabel	Profile_Picture	102
3.1.4	Deskripsi Entitas Tabel	Product_Group	103
3.1.5	Deskripsi Entitas Tabel	Product	103
3.1.6	Deskripsi Entitas Tabel	Detail_Pict	104
3.1.7	Deskripsi Entitas Tabel	Place	105
3.1.8	Deskripsi Entitas Tabel	Friendship	105
3.1.9	Deskripsi Entitas Tabel	Message	106
3.1.10	Deskripsi Entitas Tabel	Message_Content	107
3.1.11	Deskripsi Entitas Tabel	Comment	107
3.1.12	Deskripsi Entitas Tabel	Promote	108
3.1.13	Deskripsi Entitas Tabel	Appointment	108
3.1.14	Deskripsi Entitas Tabel	Testimony	110
3.1.15	Deskripsi Entitas Tabel	Notification	110
3.1.16	Deskripsi Entitas Tabel	Rate	111
3.1.17	Deskripsi Entitas Tabel	Setting	111
3.2	Physical Data Model		113
4	Deskripsi Perancangan AntarMuka		114
4.1	Login		114
4.2	Register		115
4.3	Main Menu		116
4.4	Profile Menu		117
4.5	Link and Promote		118
4.6	Manage Users		119
4.7	User home		121
4.8	Profile		122
4.9	Edit Profile		124
4.10	Change Username		125
4.11	Change Password		126
4.12	Edit Profile Picture		127
4.13	Edit Setting		128

4.14	Notification	129
4.15	Category	129
4.16	Friendship	131
4.17	Show Friend	132
4.18	Product	133
4.19	Insert Product	134
4.20	Edit Product	135
4.21	Detail Product	136
4.22	Lounge	138
4.23	Showroom	140
4.24	Insert Promote	141
4.25	Place	142
4.26	Insert Place	143
4.27	List Message	144
4.28	Detail Message	145
4.29	Contact Us	146
4.30	List Appointment	147
4.31	Insert Appointment	148
4.32	Insert Time or Place Appointment	150
4.33	Edit Appointment	151
4.34	Rate Appointment	152

Daftar Gambar

Gambar 3.1	Physical Data Model.....	113
Gambar 4.1	Rancangan Antarmuka Login.....	114
Gambar 4.2	Rancangan Antarmuka Register.....	115
Gambar 4.3	Rancangan Antarmuka Menu Utama.....	116
Gambar 4.4	Rancangan Antarmuka Profile Menu.....	117
Gambar 4.5	Rancangan Antarmuka Link dan Promote.....	118
Gambar 4.6	Rancangan Antarmuka Manage User.....	119
Gambar 4.7	Rancangan Antarmuka Detail User.....	120
Gambar 4.8	Rancangan Antarmuka User Home.....	121
Gambar 4.9	Rancangan Antarmuka Profile.....	122
Gambar 4.10	Rancangan Antarmuka Insert Message.....	123
Gambar 4.11	Rancangan Antarmuka Edit Testimony.....	123
Gambar 4.12	Rancangan Antarmuka Edit Profil.....	124
Gambar 4.13	Rancangan Antarmuka Change Username.....	125
Gambar 4.14	Rancangan Antarmuka Change Password.....	126
Gambar 4.15	Rancangan Antarmuka Edit Profile Picture.....	127
Gambar 4.16	Rancangan Antarmuka Edit Setting.....	128
Gambar 4.17	Rancangan Antarmuka Notification.....	129
Gambar 4.18	Rancangan Antarmuka Category.....	130
Gambar 4.19	Rancangan Antarmuka Edit Category.....	130
Gambar 4.20	Rancangan Antarmuka Friendship.....	131
Gambar 4.21	Rancangan Antarmuka Show Friend.....	132
Gambar 4.22	Rancangan Antarmuka Product.....	133
Gambar 4.23	Rancangan Antarmuka Insert Product.....	134
Gambar 4.24	Rancangan Antarmuka Edit Product.....	135
Gambar 4.25	Rancangan Antarmuka Detail Product.....	136
Gambar 4.26	Rancangan Antarmuka Insert Picture.....	137
Gambar 4.27	Rancangan Antarmuka Lounge.....	138
Gambar 4.28	Rancangan Antarmuka Showroom.....	140
Gambar 4.29	Rancangan Antarmuka Promote.....	141
Gambar 4.30	Rancangan Antarmuka Place.....	142
Gambar 4.31	Rancangan Antarmuka Insert Place.....	143
Gambar 4.32	Rancangan Antarmuka List Message.....	144
Gambar 4.33	Rancangan Antarmuka Detail Message.....	145
Gambar 4.34	Rancangan Antarmuka Contact Us.....	146
Gambar 4.35	Rancangan Antarmuka List Appointment.....	147
Gambar 4.36	Rancangan Antarmuka Detail Appointment.....	148
Gambar 4.37	Rancangan Antarmuka Insert Appointment.....	149
Gambar 4.38	Rancangan Antarmuka Insert Time or Place Appointment	150
Gambar 4.39	Rancangan Antarmuka Edit Appointment.....	151
Gambar 4.40	Rancangan Antarmuka Rate Appointment.....	152

1 Pendahuluan

1.1 Tujuan

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

1.2 Ruang Lingkup

Perangkat Lunak SOS dikembangkan dengan tujuan untuk :

1. Menyediakan sarana bagi pengguna untuk menyebarkan informasi mengenai produk yang dijual atau dibutuhkan kepada pengguna lainnya.
2. Mempermudah pengguna untuk mencari pembeli atau penjual yang tepat dari sisi jarak maupun kepercayaan.
3. Mempererat hubungan antar pengguna dengan saling membantu untuk memenuhi kebutuhan antara satu pengguna dengan pengguna lainnya.

1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

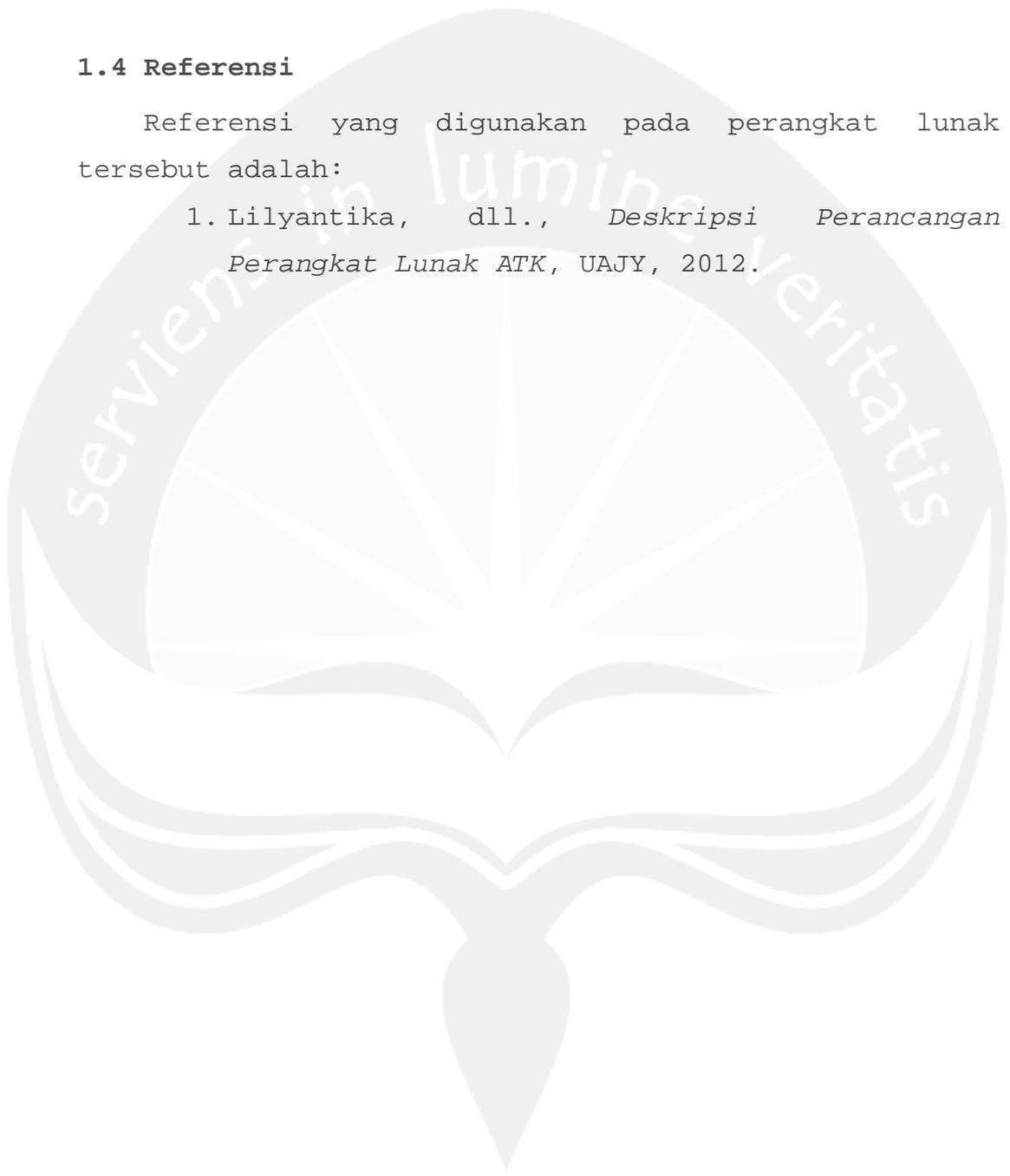
Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.

SOS	Perangkat lunak pengelolaan penjualan perabot.
-----	--

1.4 Referensi

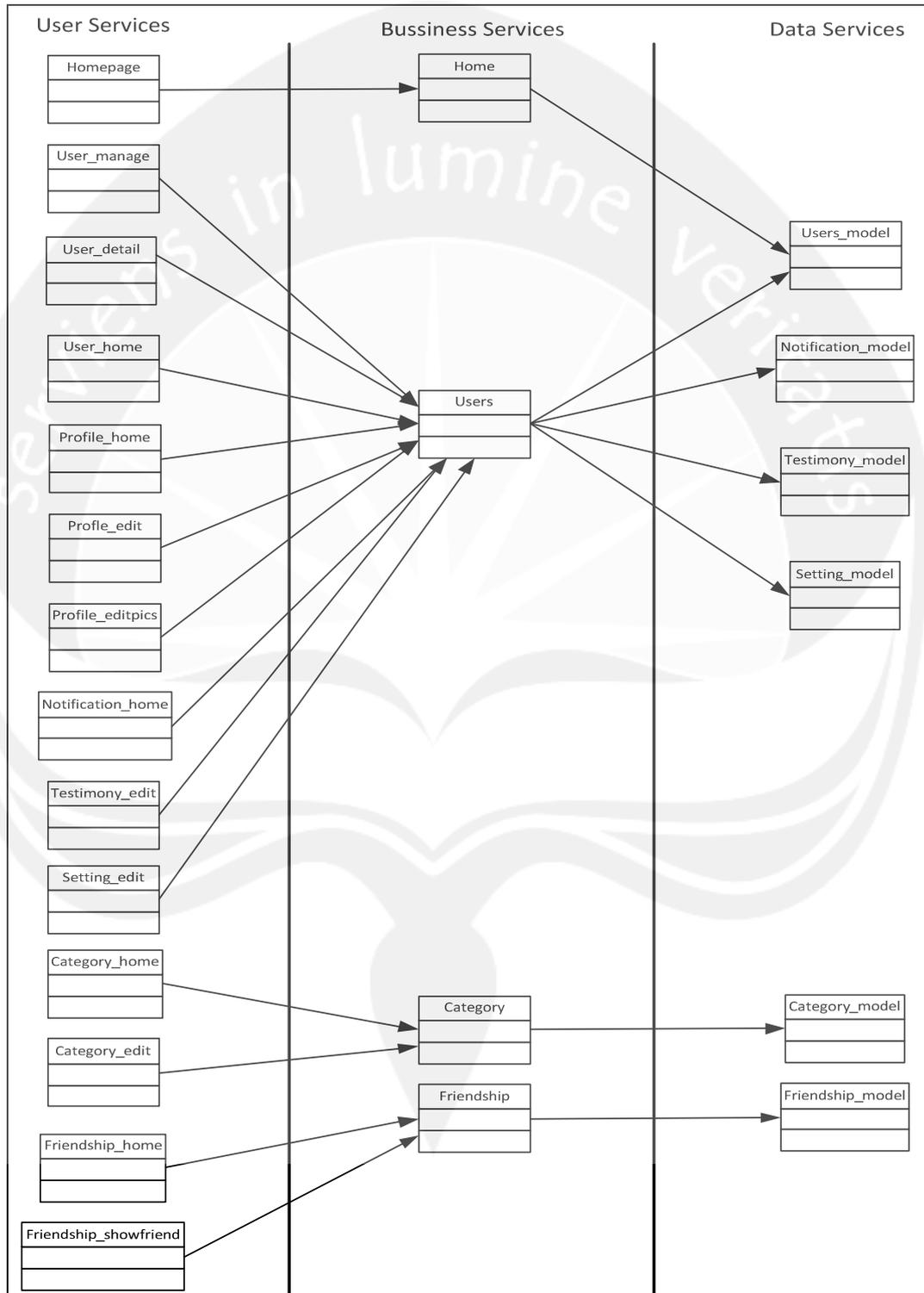
Referensi yang digunakan pada perangkat lunak tersebut adalah:

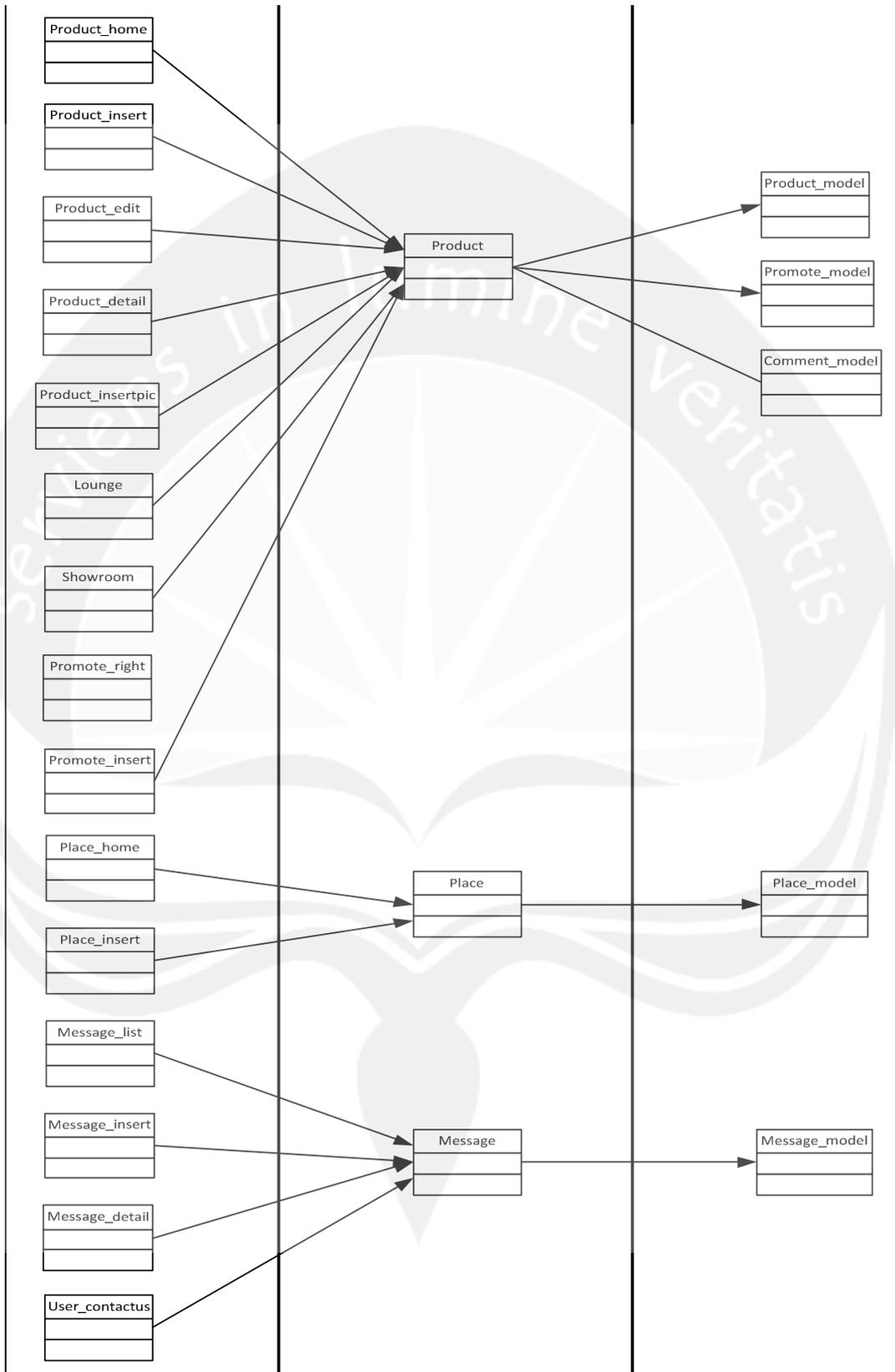
1. Lilyantika, dll., *Deskripsi Perancangan Perangkat Lunak ATK*, UAJY, 2012.

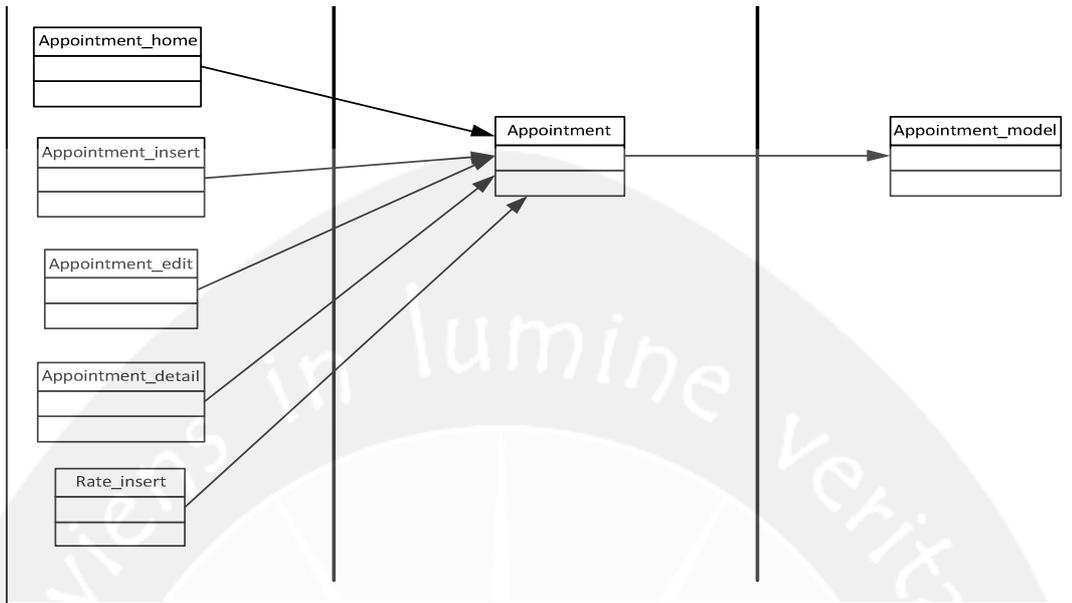


2 Perancangan Sistem

2.1 Perancangan Arsitektur





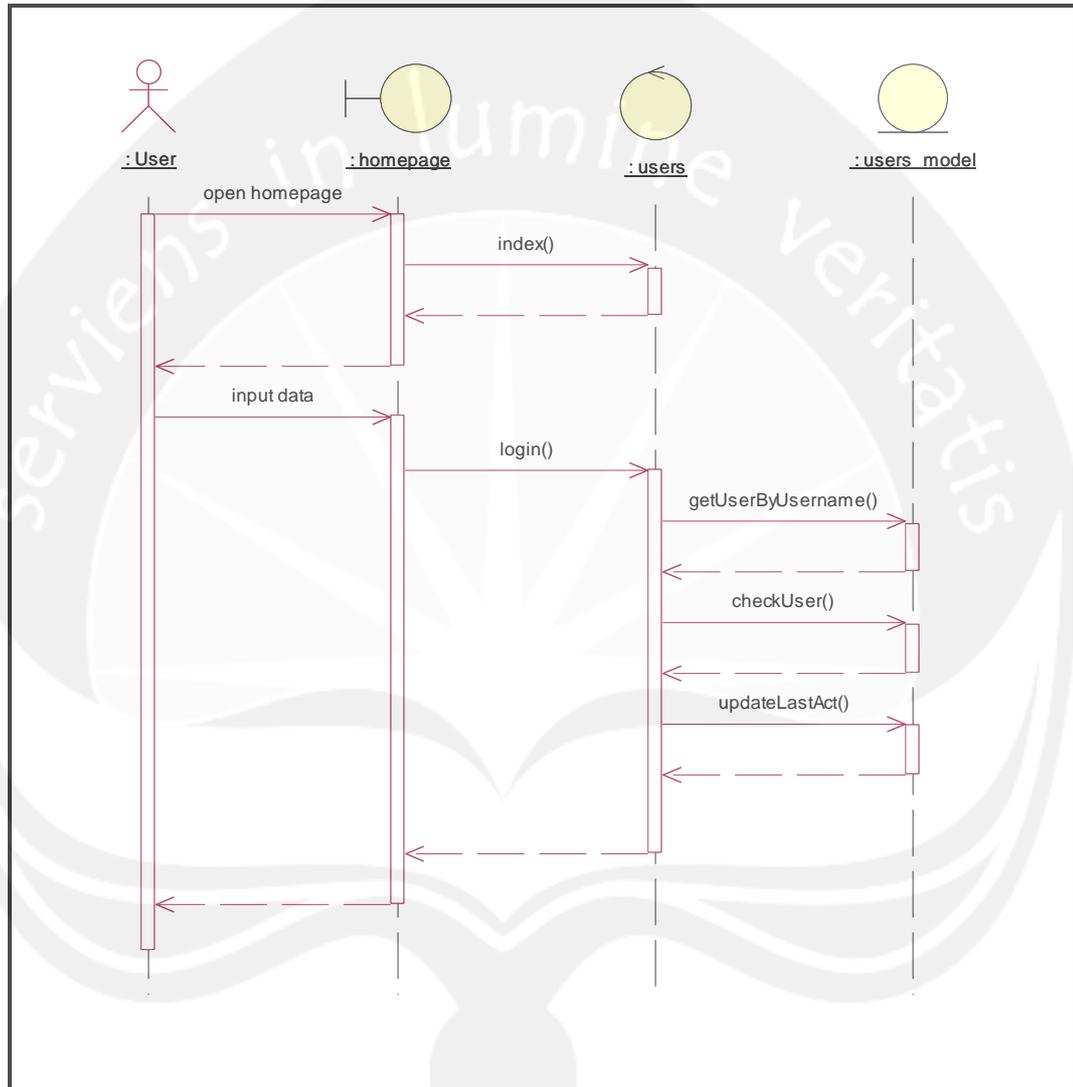


Gambar 2.1 Rancangan Arsitektur SOS

2.2 Perancangan Rinci

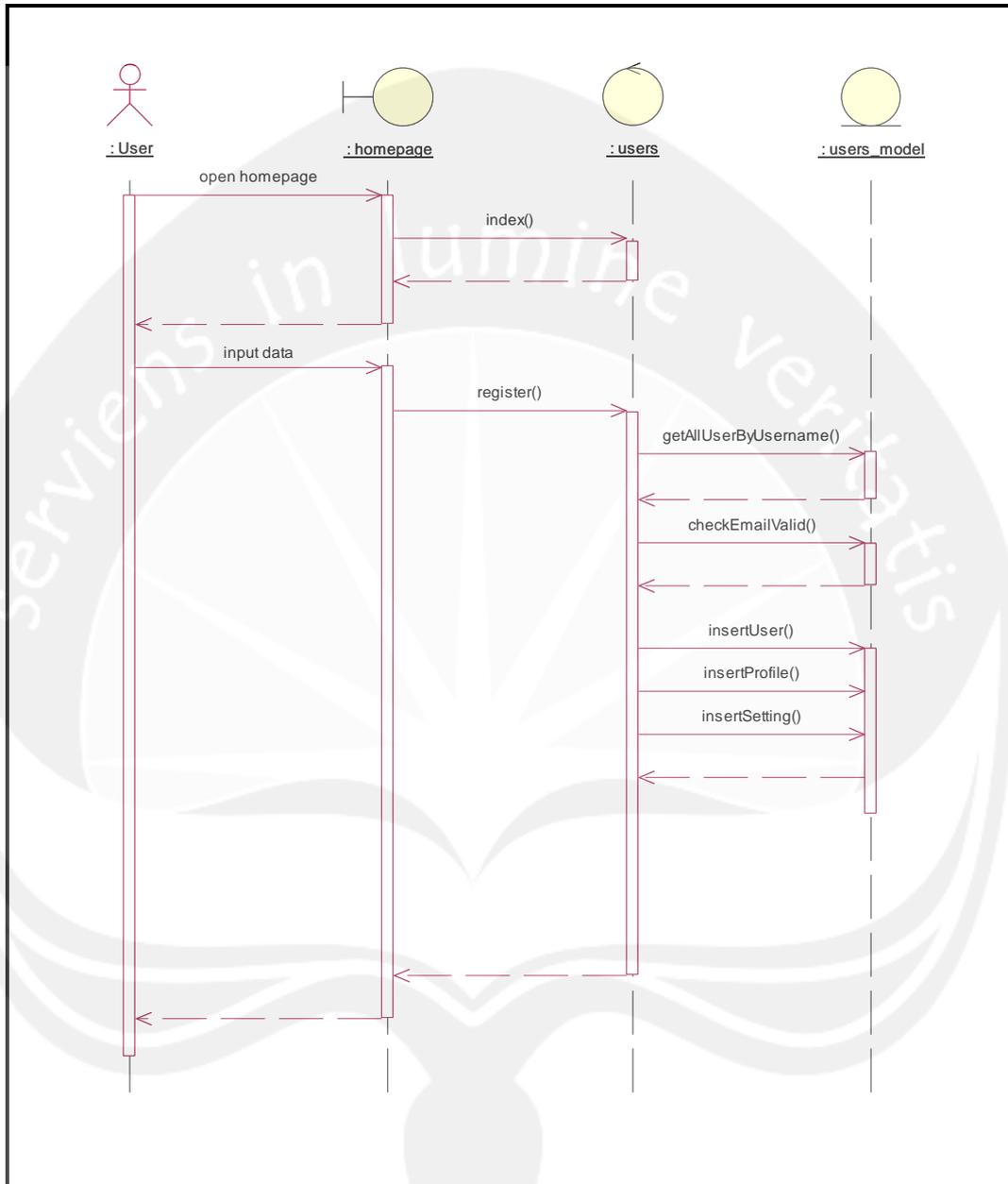
2.2.1 Sequence Diagram

2.2.1.1 Login



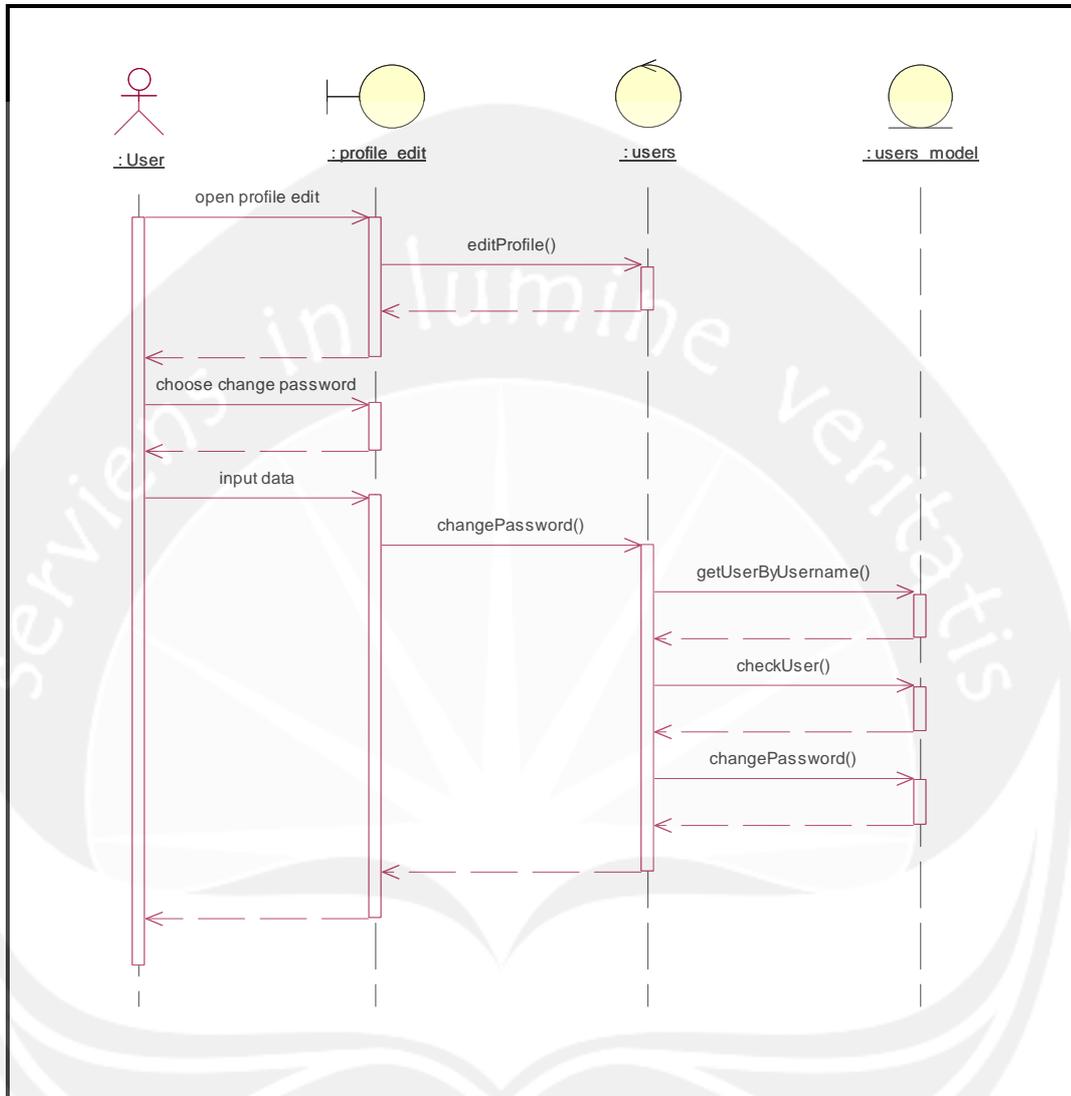
Gambar 2.2 Sequence Diagram : Login

2.2.1.2 Register



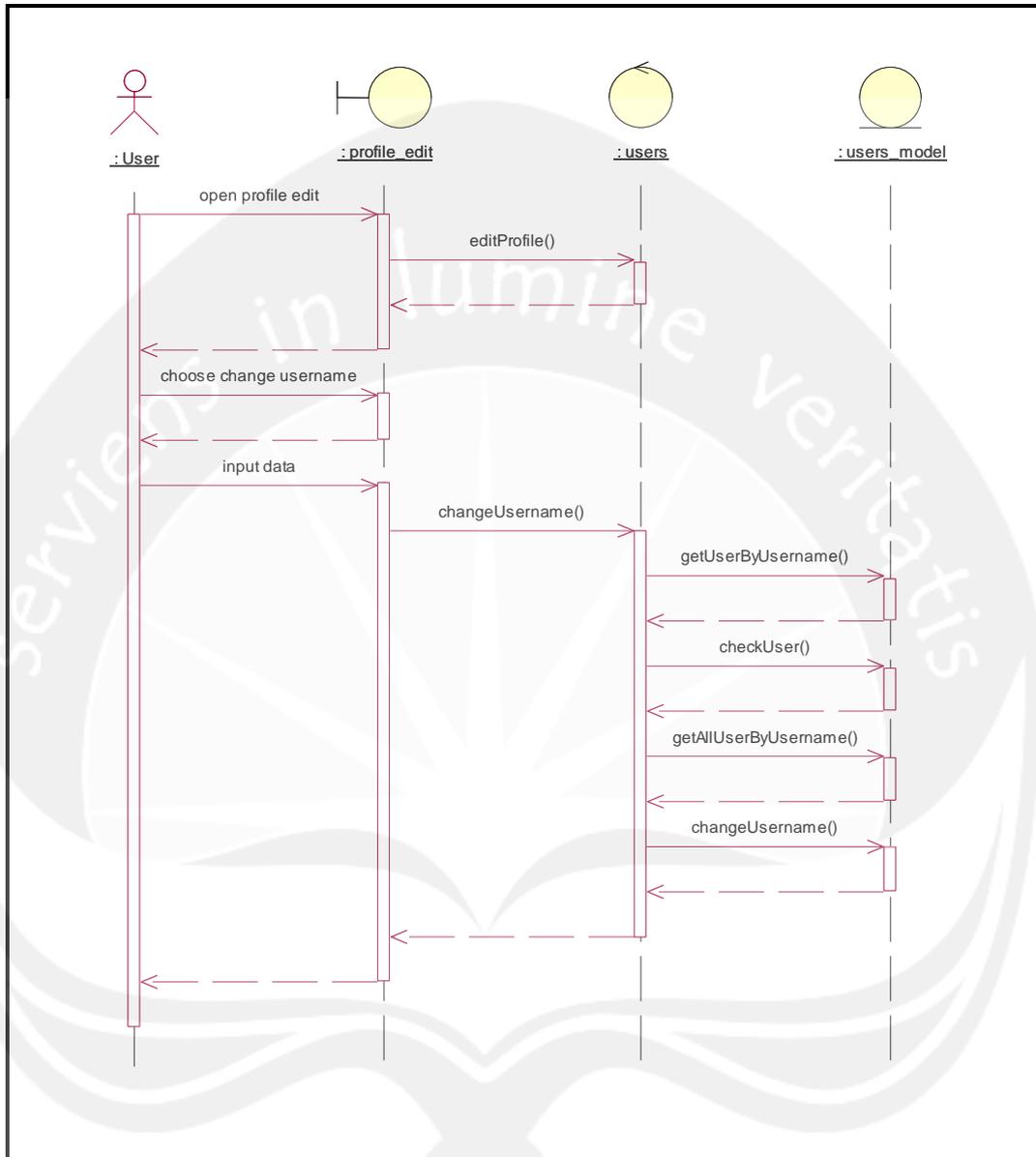
Gambar 2.3 Sequence Diagram : Register

2.2.1.3 Change Password



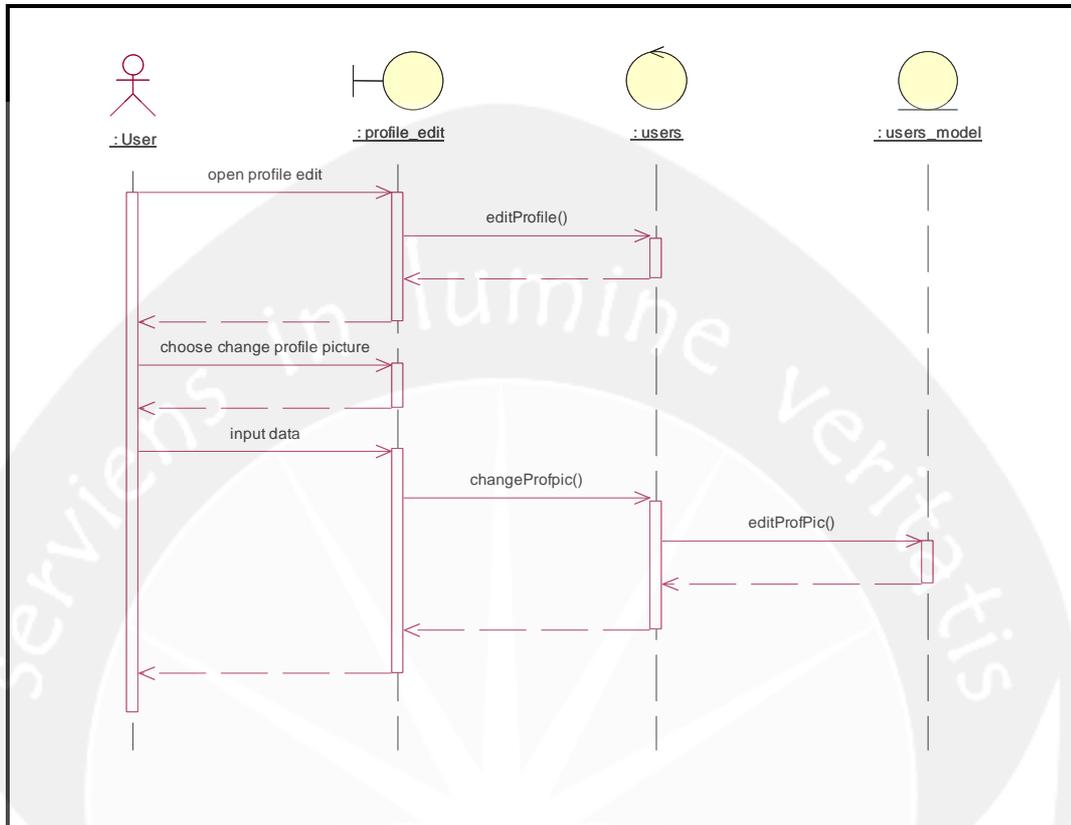
Gambar 2.4 Sequence Diagram : Change Password

2.2.1.4 Change Username



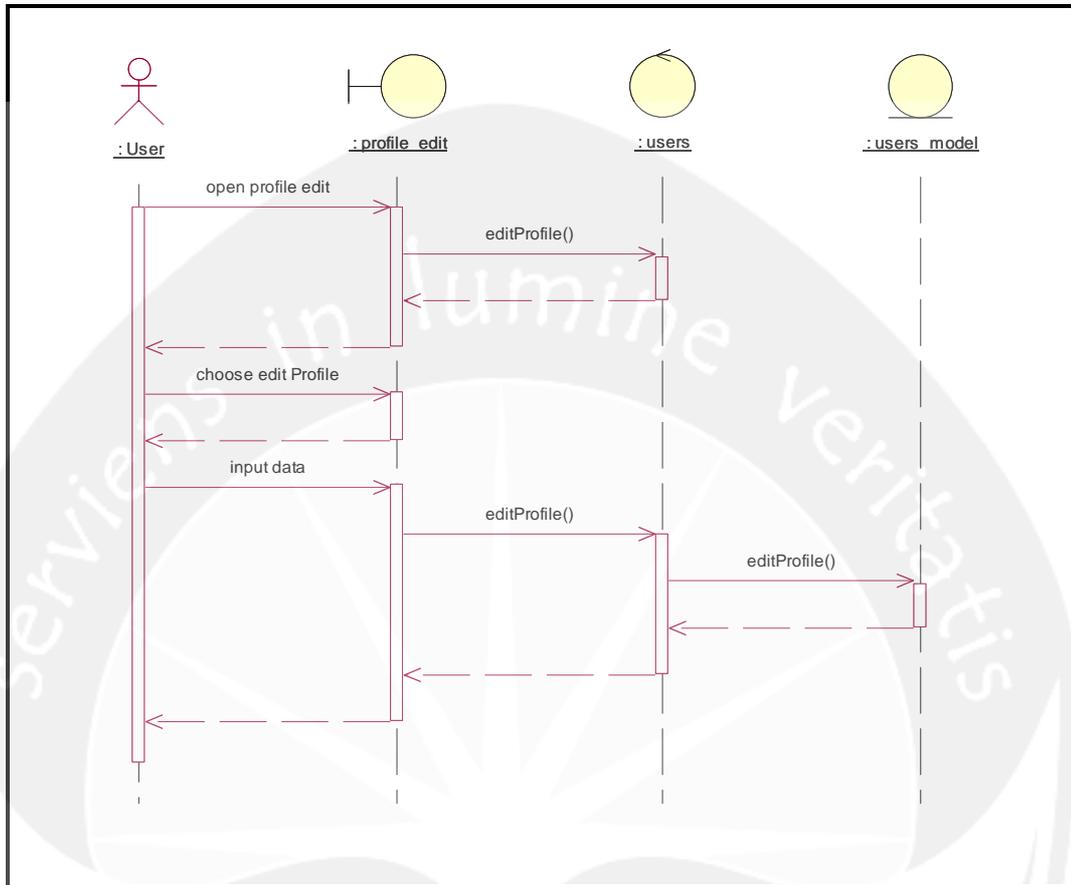
Gambar 2.5 Sequence Diagram : Change Password

2.2.1.5 Change Profile Picture



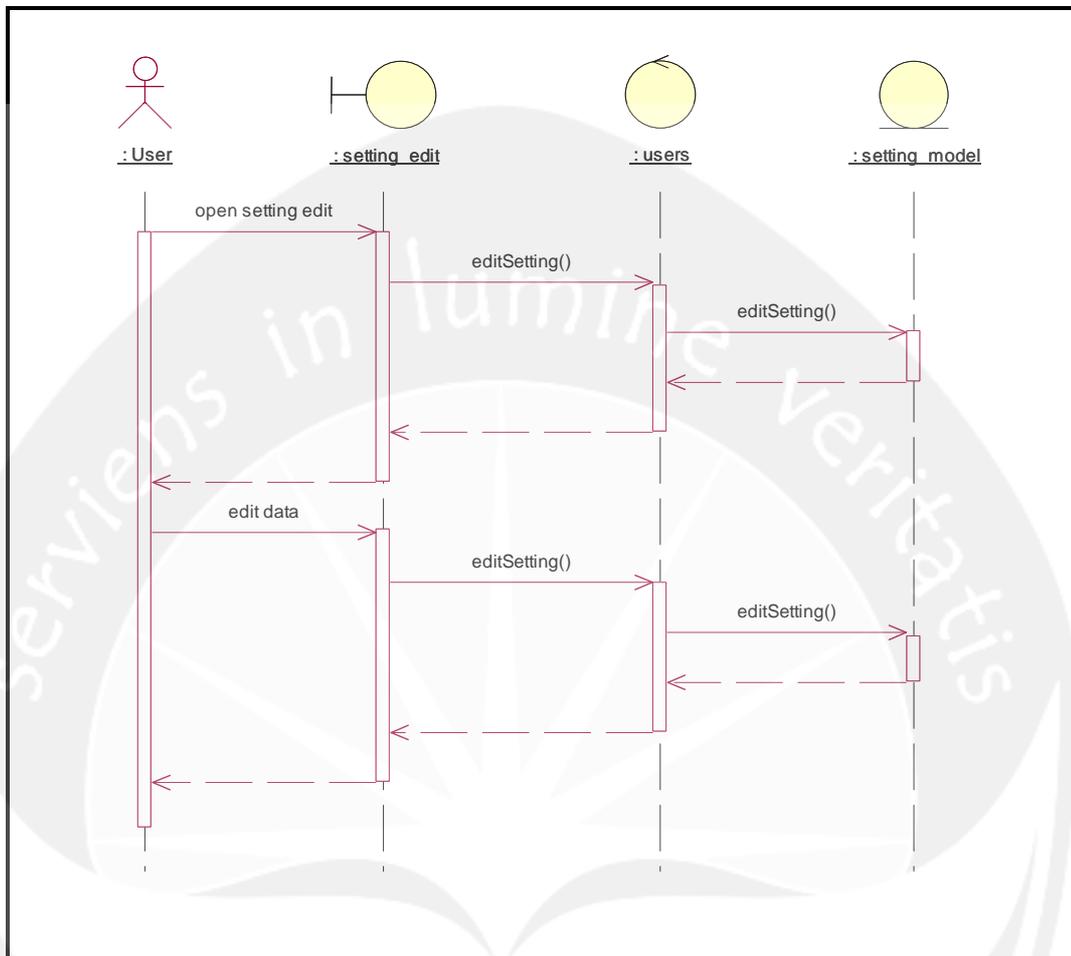
Gambar 2.6 Sequence Diagram : Change Password

2.2.1.6 Edit Profile



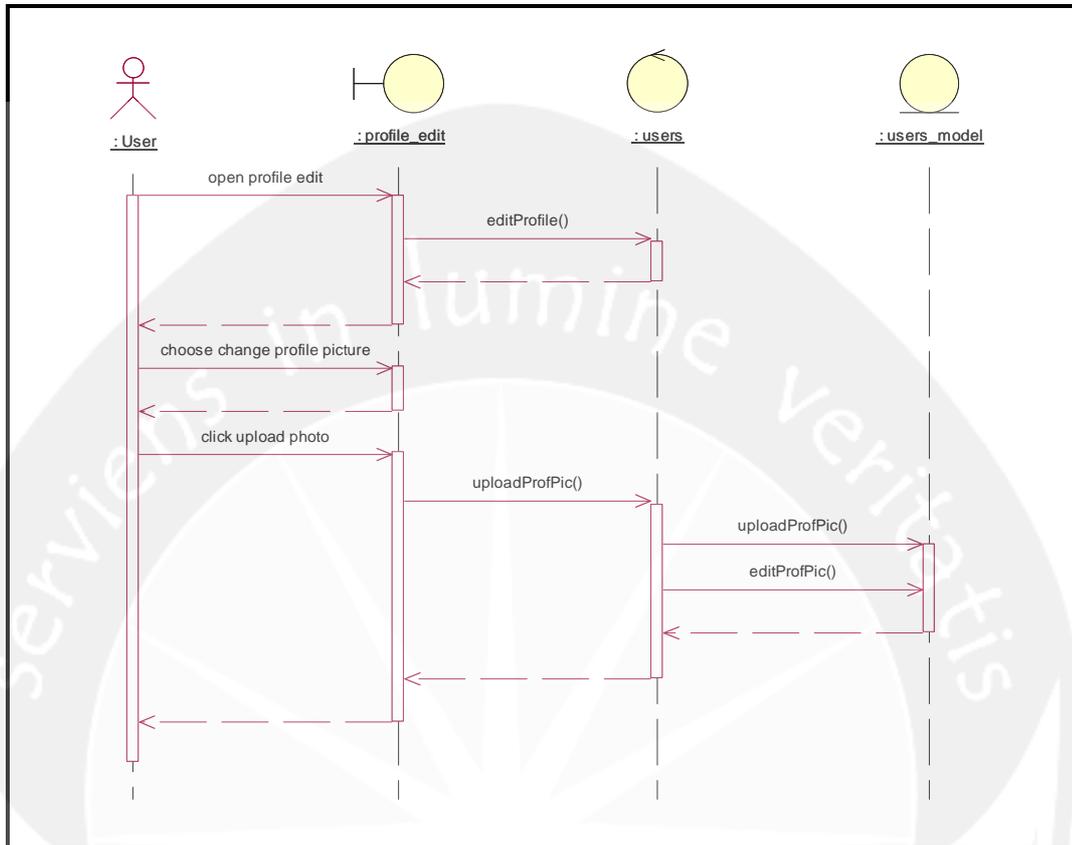
Gambar 2.7 Sequence Diagram : Change Password

2.2.1.7 Edit Setting



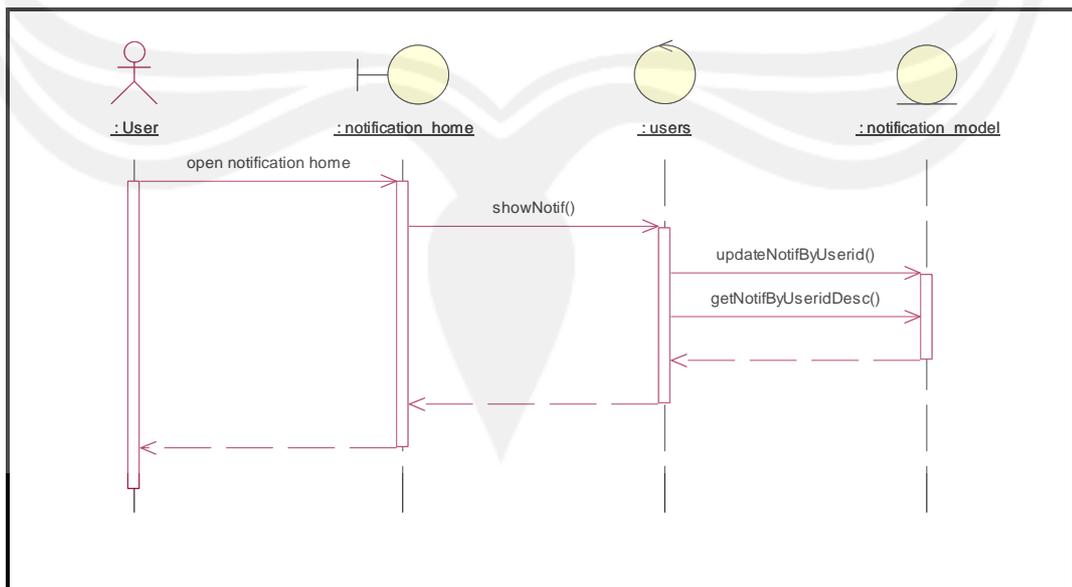
Gambar 2.8 Sequence Diagram : Login

2.2.1.8 Upload Profile Picture



Gambar 2.9 Sequence Diagram : Login

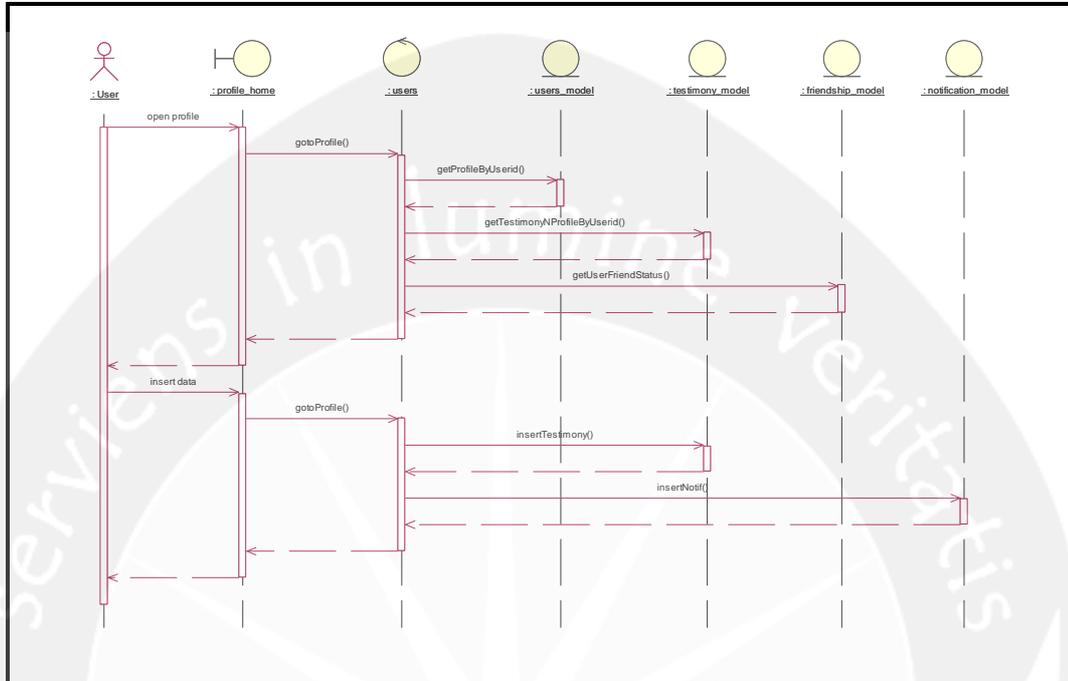
2.2.1.9 Show Notification



Sequence Diagram : Change Password

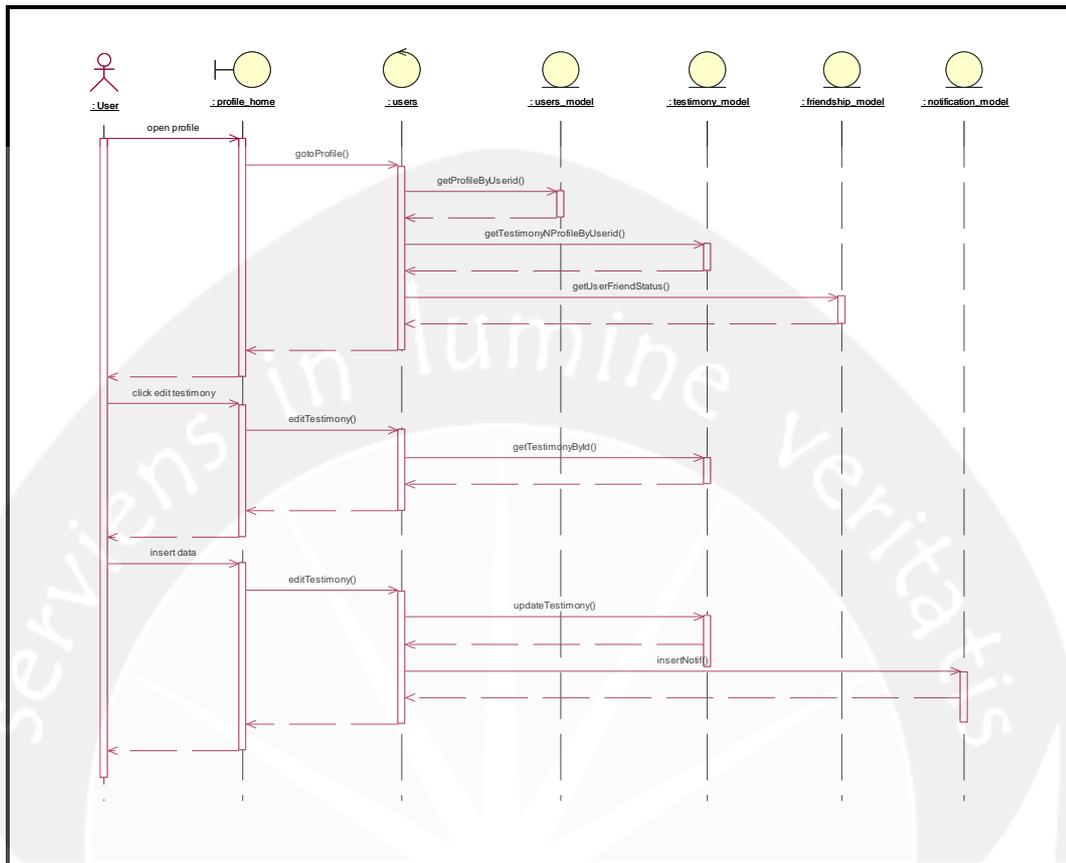
2.2.1.10 Pengelolaan Testimony

2.2.1.10.1 Insert Testimony



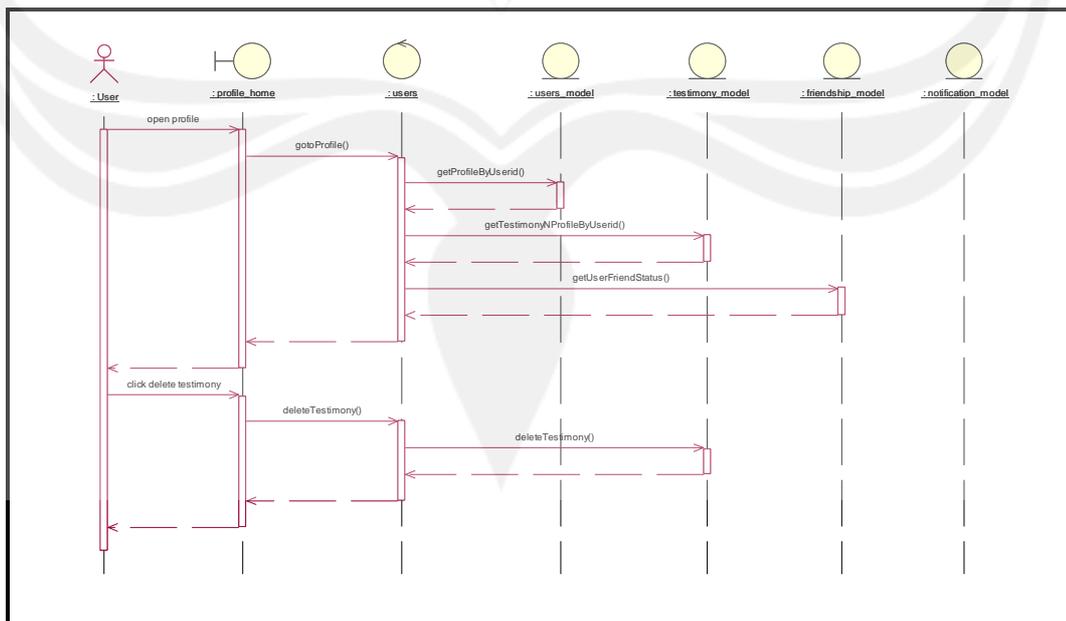
Gambar 2.10 Sequence Diagram : Login

2.2.1.10.2 Edit Testimony



Gambar 2.11 Sequence Diagram : Login

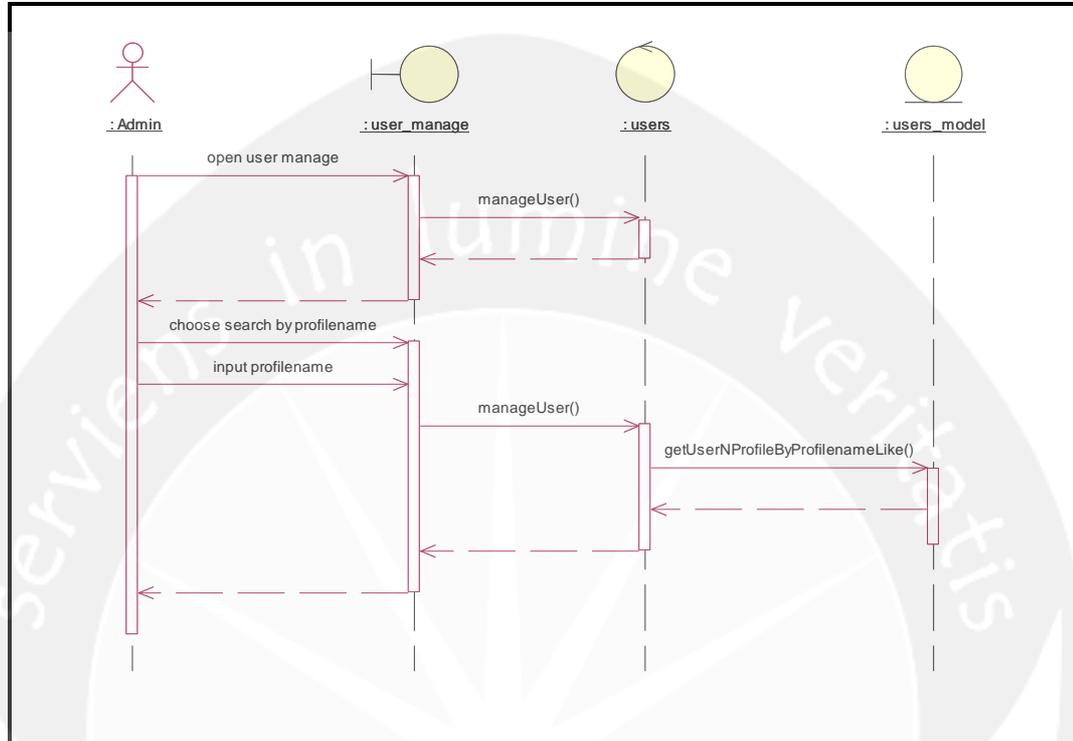
2.2.1.10.3 Delete Testimony



Gambar 2.12 Sequence Diagram : Login

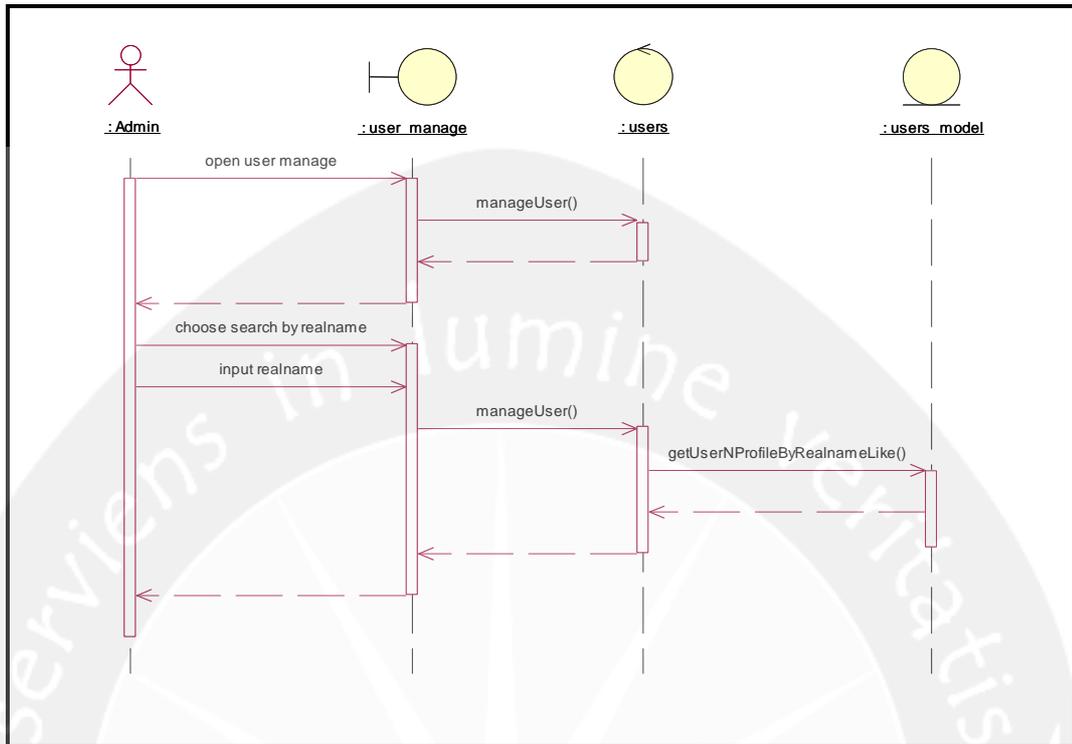
2.2.1.11 Pengelolaan User

2.2.1.11.1 Search User By Profile Name



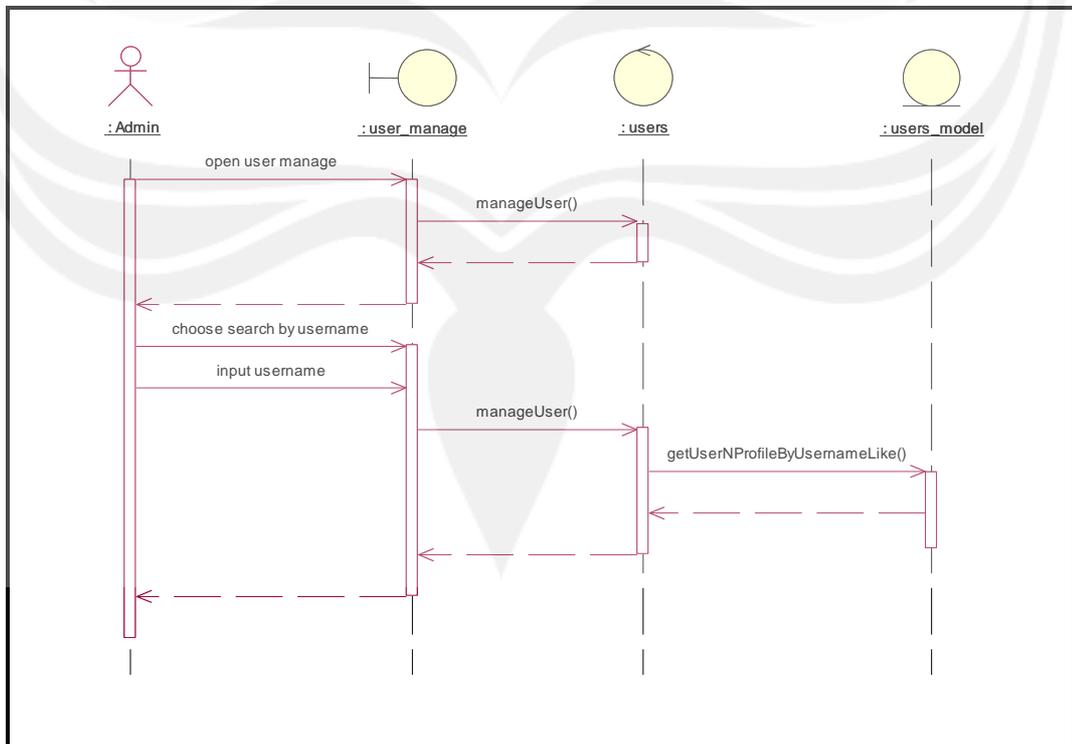
Gambar 2.13 Sequence Diagram : Login

2.2.1.11.2 Search User By Realname



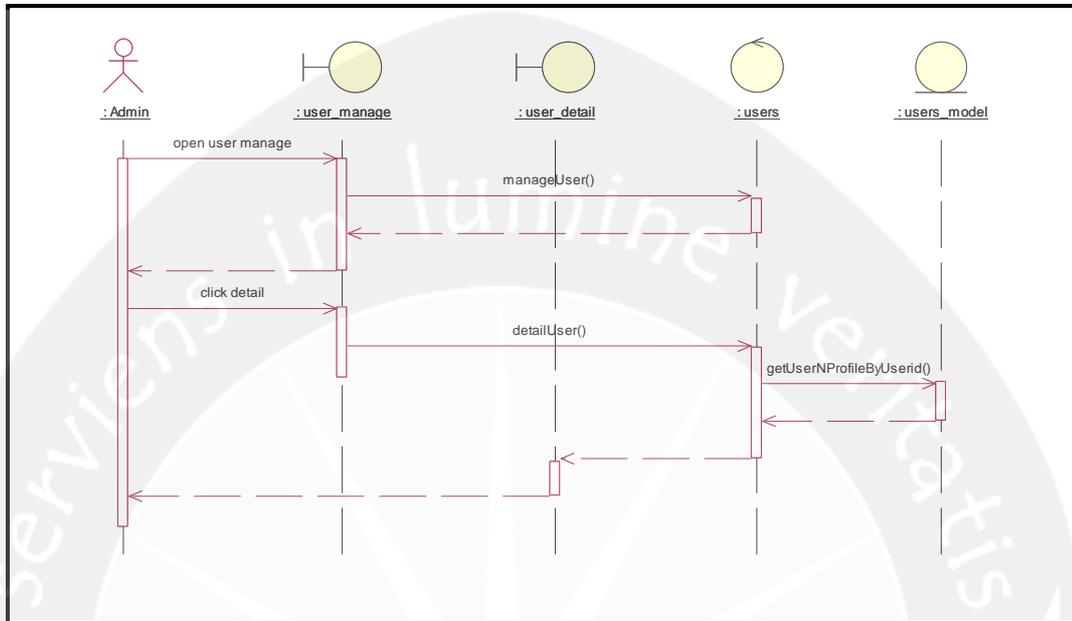
Gambar 2.14 Sequence Diagram : Login

2.2.1.11.3 Search User By Username



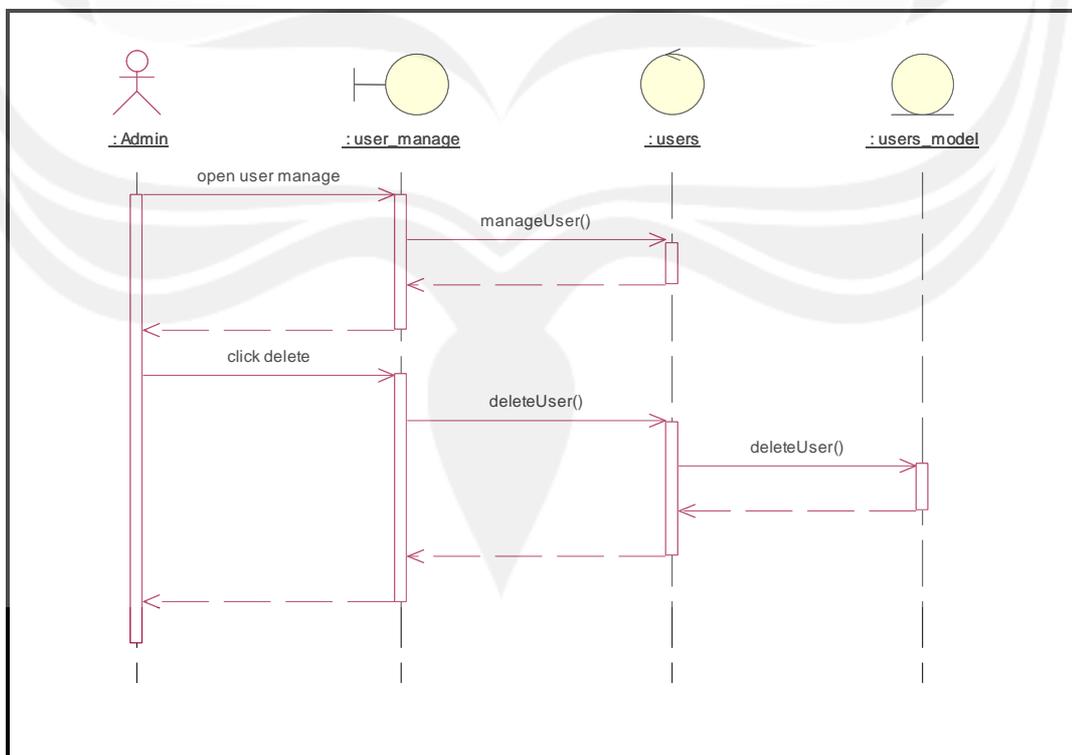
Gambar 2.15 Sequence Diagram : Login

2.2.1.11.4 Show Detail User



Gambar 2.16 Sequence Diagram : Login

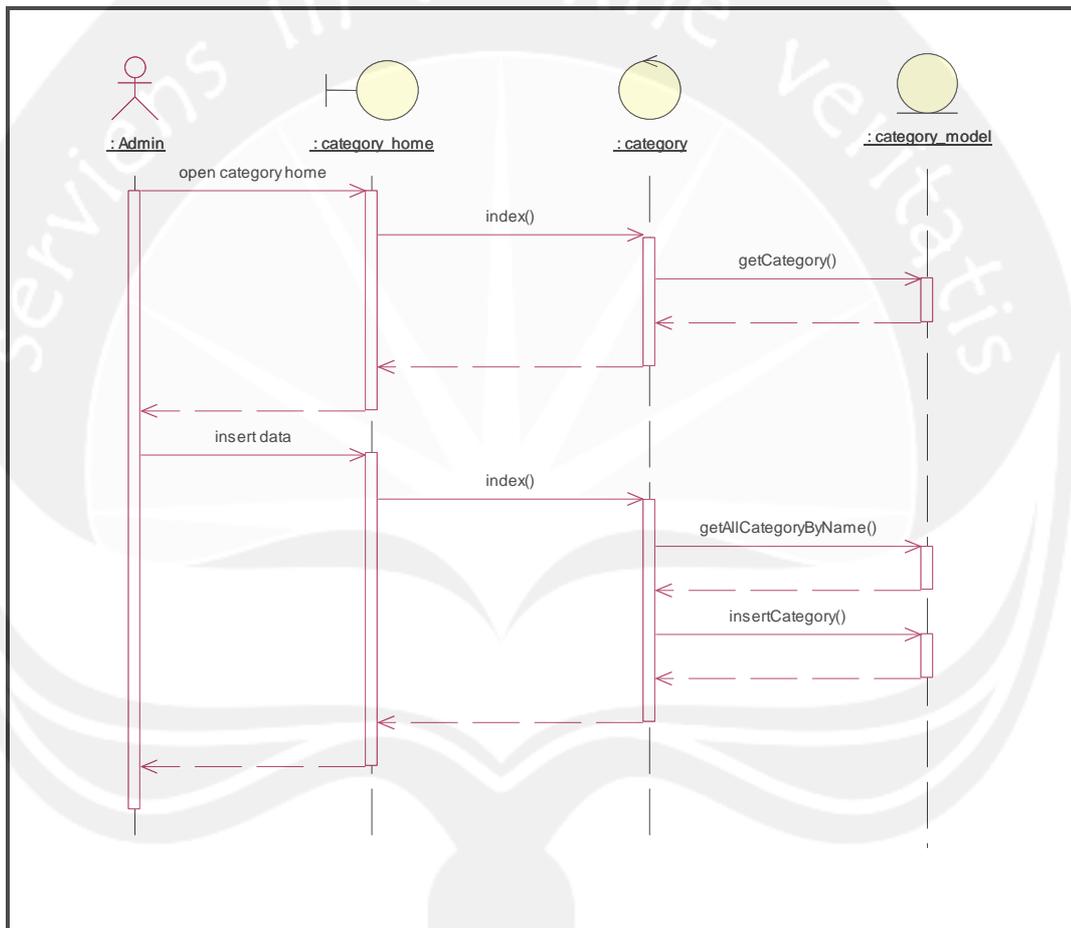
2.2.1.11.5 Delete User



Gambar 2.17 Sequence Diagram : Login

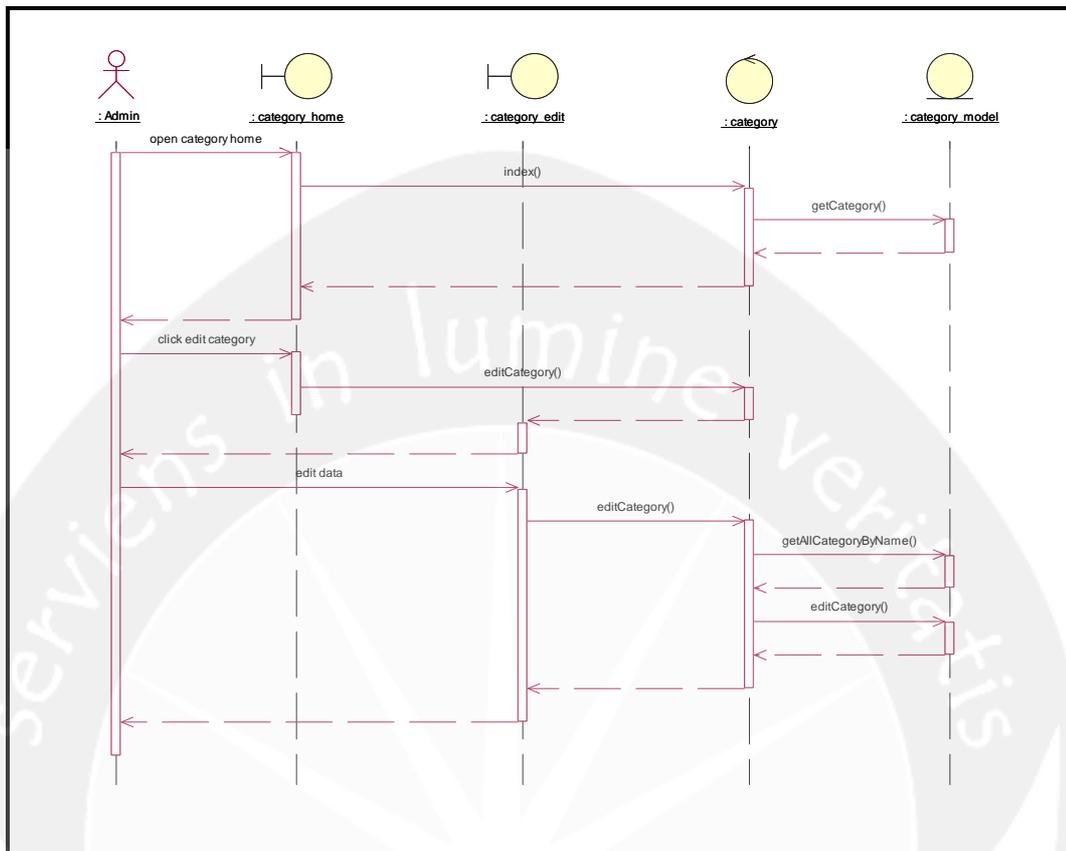
2.2.1.12 Pengelolaan Category

2.2.1.12.1 Insert Category



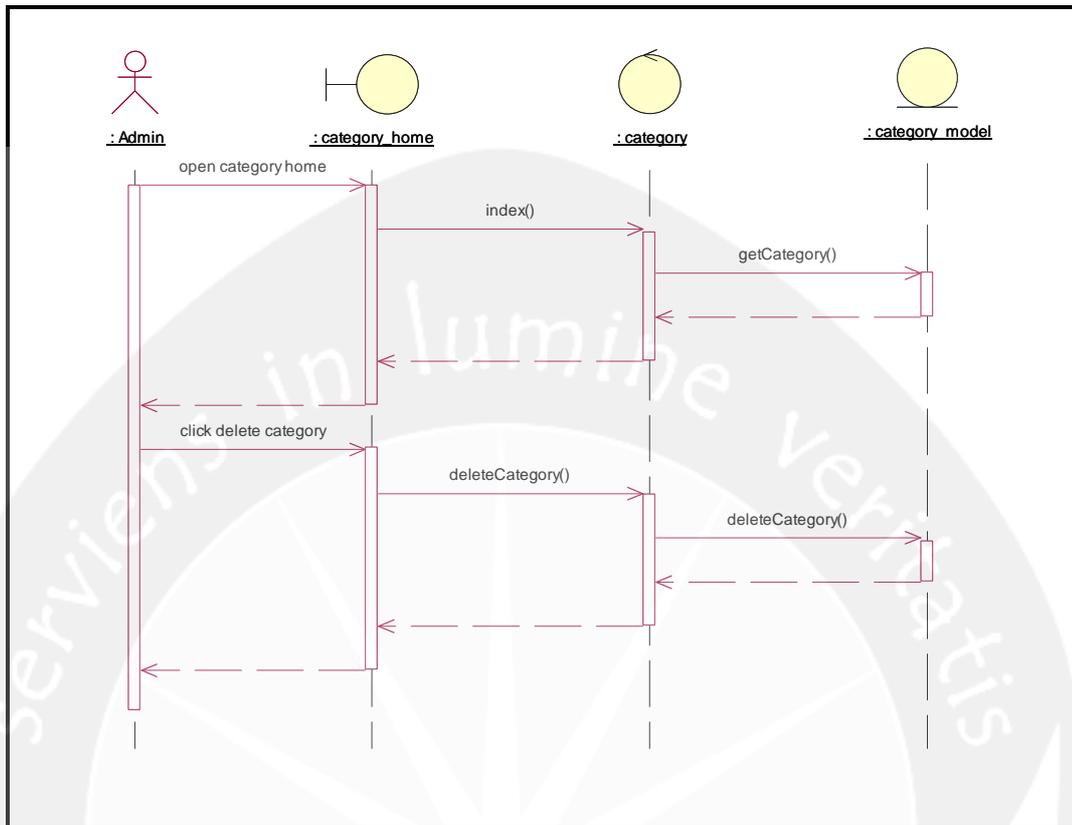
Gambar 2.18 Sequence Diagram : Login

2.2.1.12.2 Edit Category



Gambar 2.19 Sequence Diagram : Login

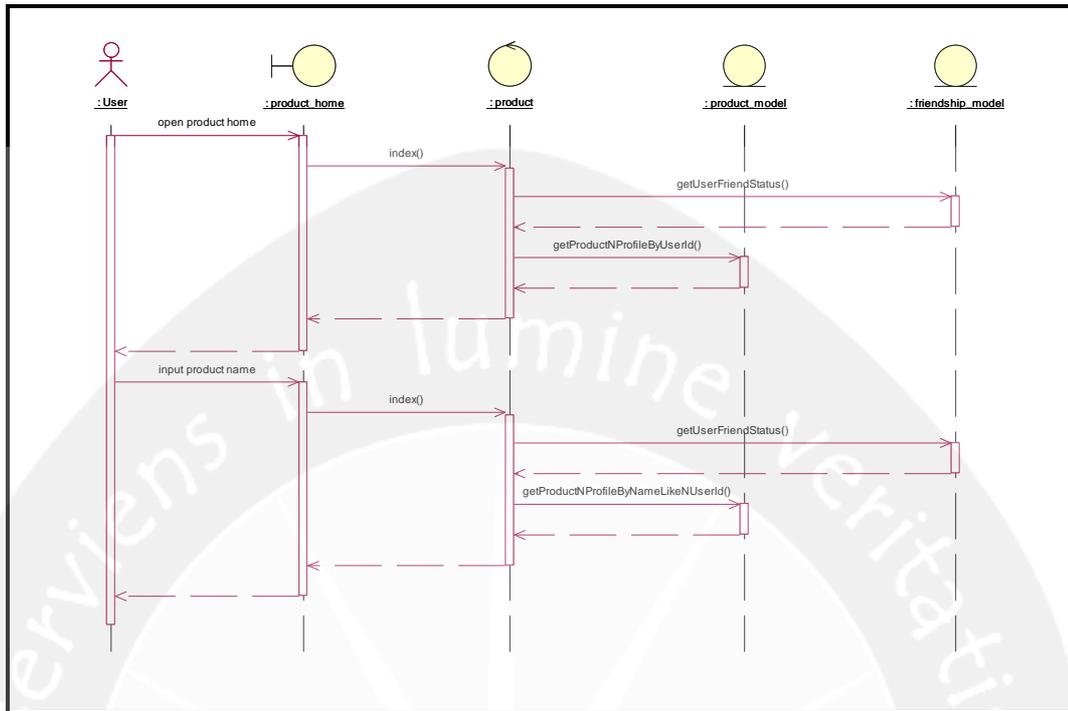
2.2.1.12.3 Delete Category



Gambar 2.20 Sequence Diagram : Login

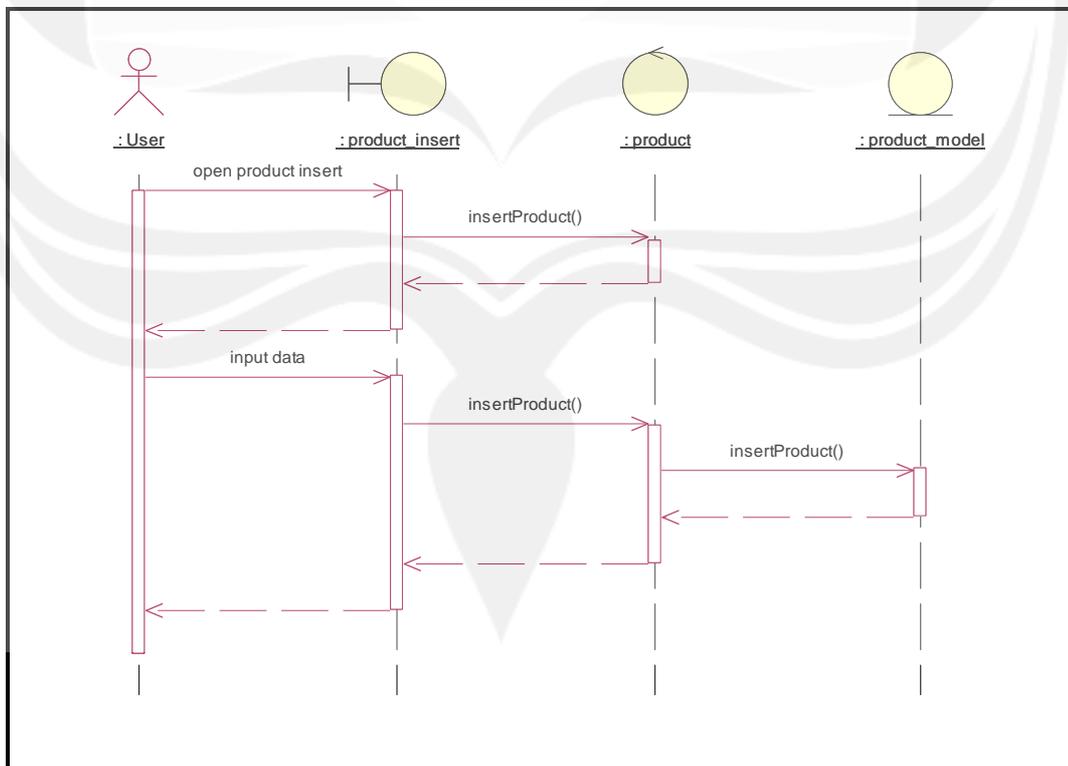
2.2.1.13 Pengelolaan Product

2.2.1.13.1 Show Product



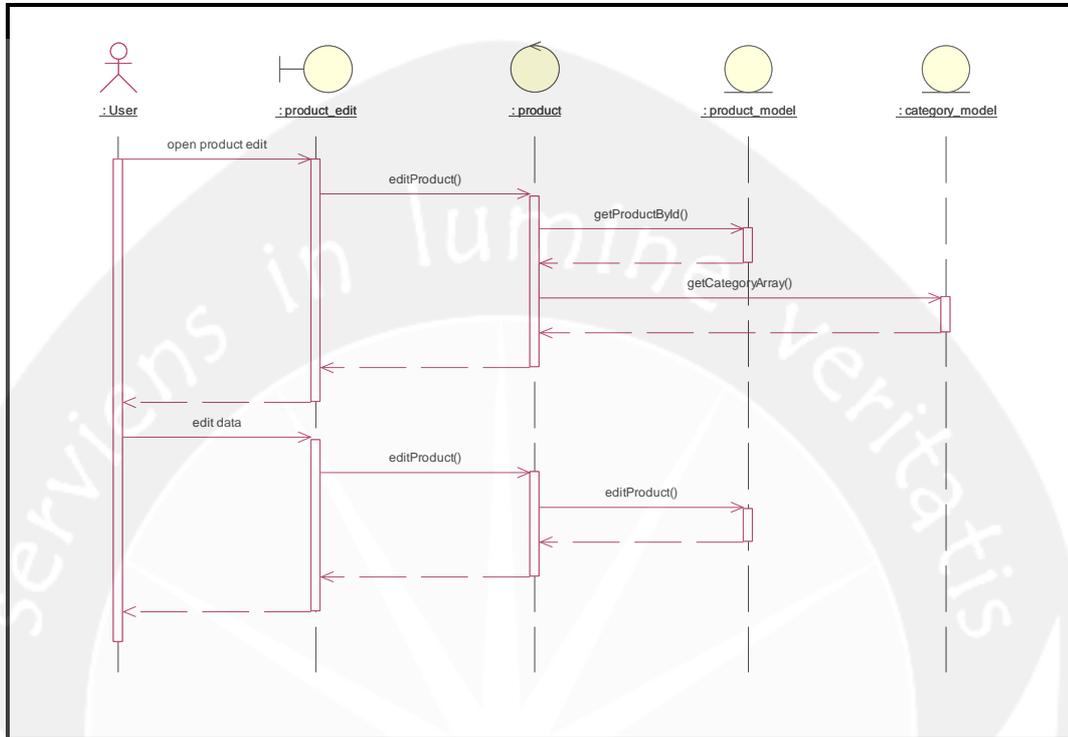
Gambar 2.21 Sequence Diagram : Login

2.2.1.13.2 Insert Product



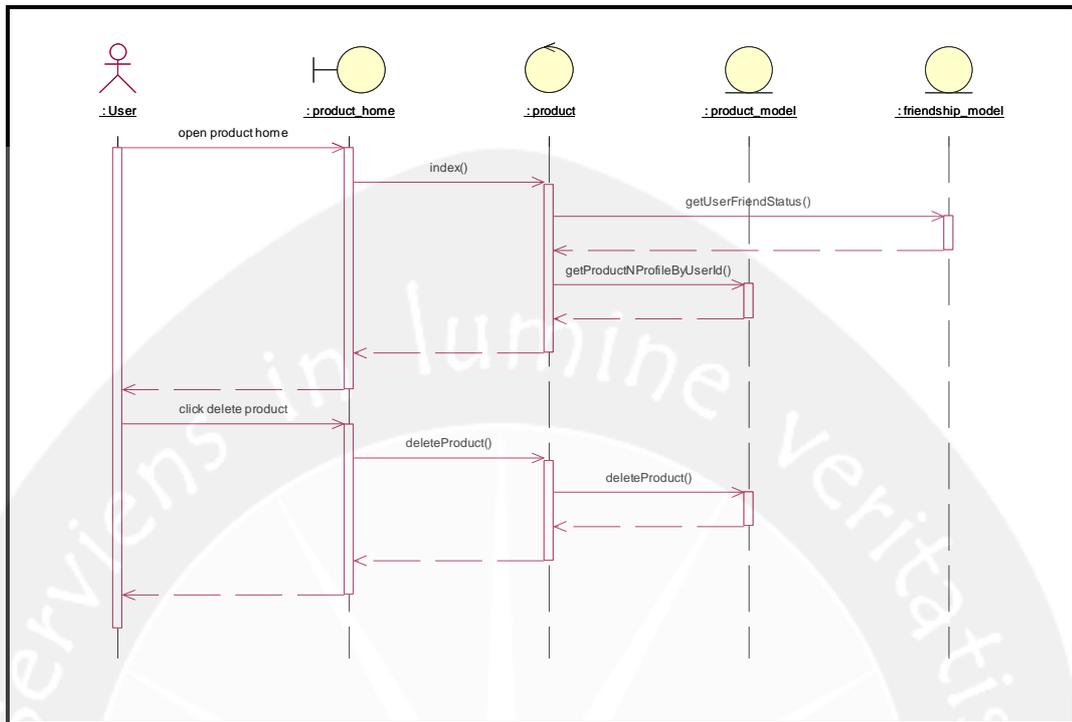
Gambar 2.22 Sequence Diagram : Login

2.2.1.13.3 Edit Product



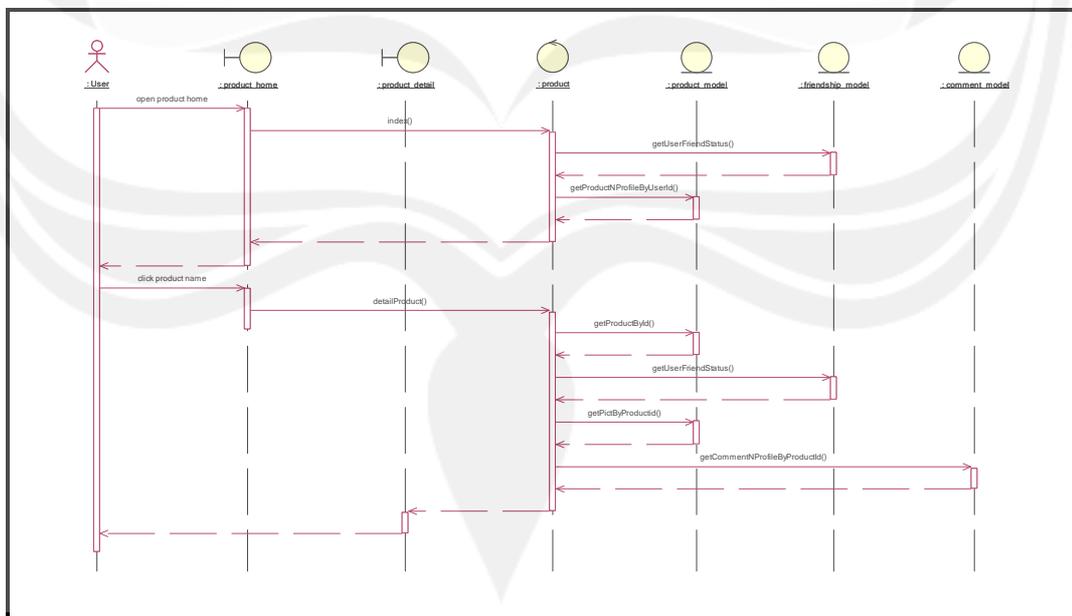
Gambar 2.23 Sequence Diagram : Login

2.2.1.13.4 Delete Product



Gambar 2.24 Sequence Diagram : Login

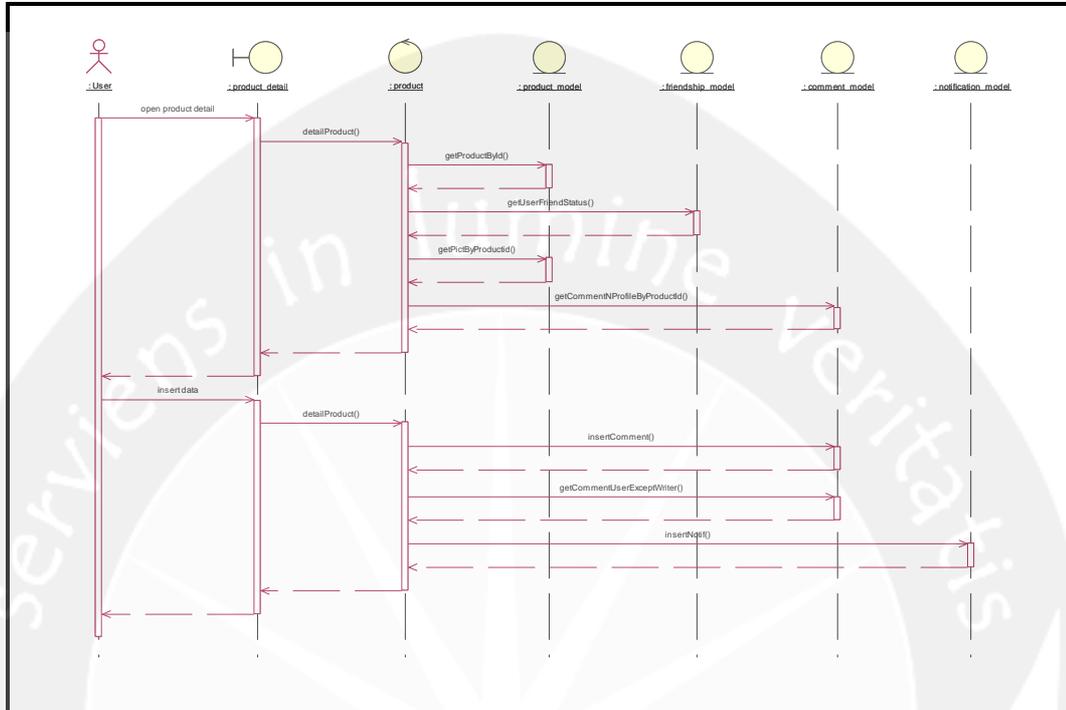
2.2.1.13.5 Detail Product



Gambar 2.25 Sequence Diagram : Login

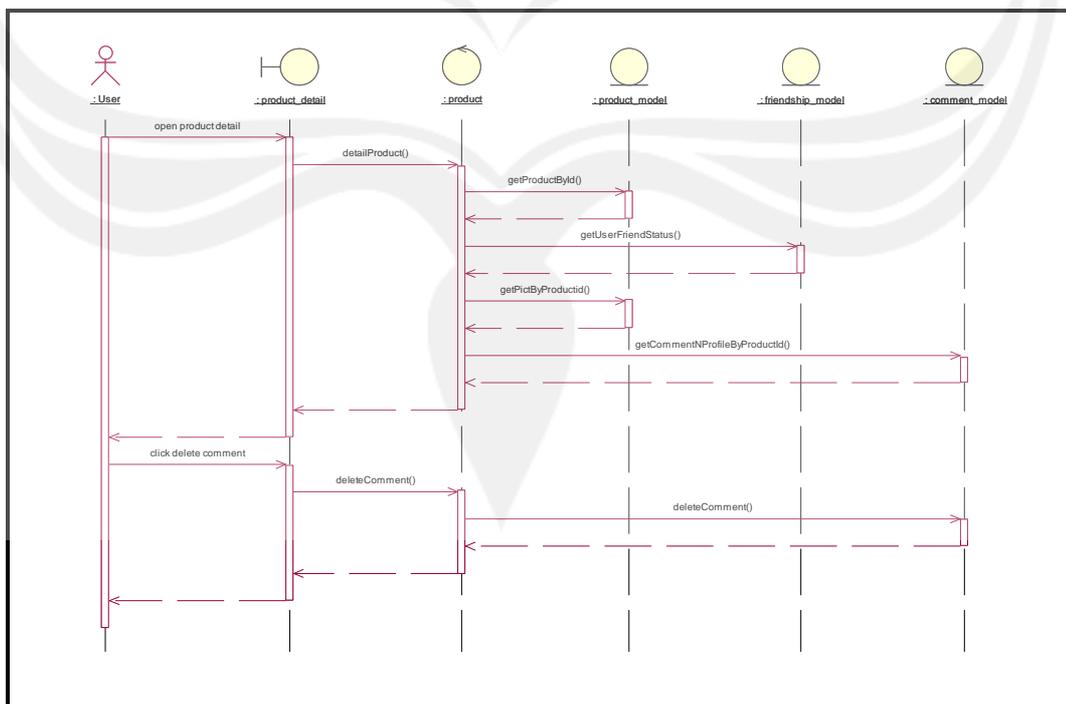
2.2.1.14 Pengelolaan Comment

2.2.1.14.1 Insert Comment



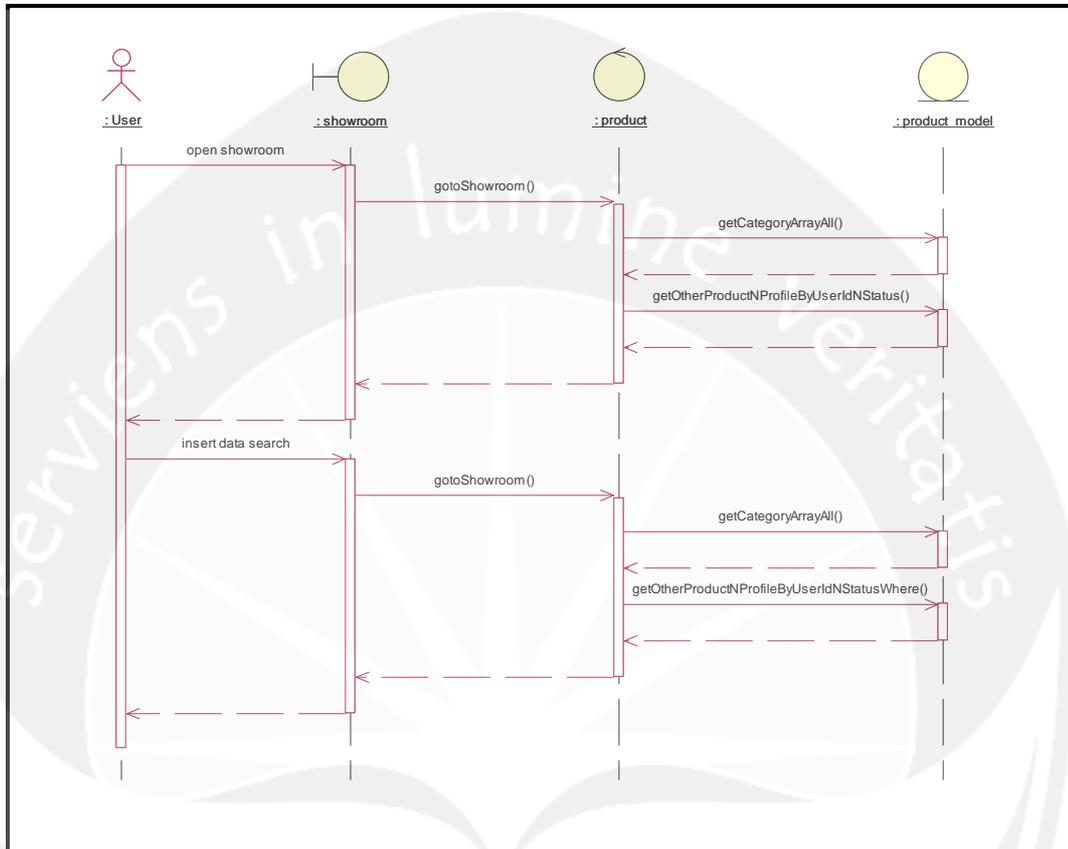
Gambar 2.26 Sequence Diagram : Login

2.2.1.14.2 Delete Comment



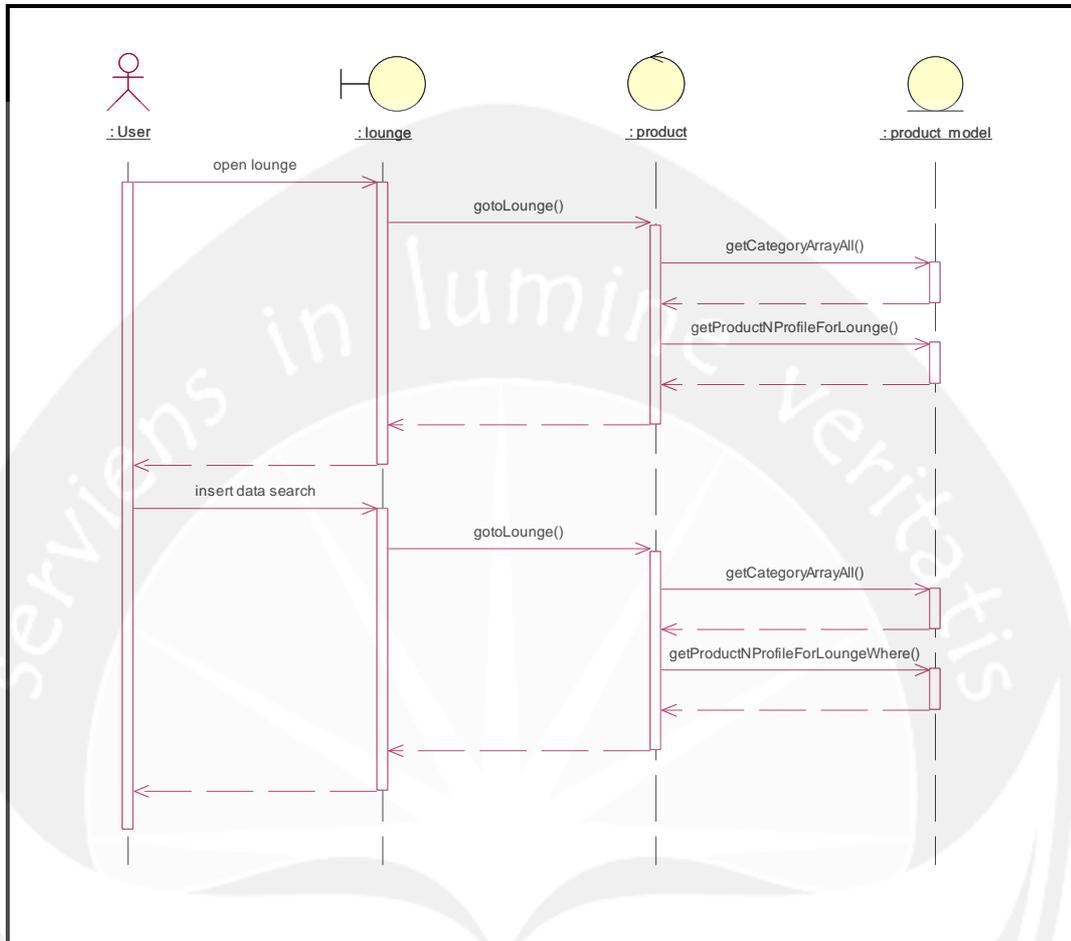
Gambar 2.27 Sequence Diagram : Login

2.2.1.15 Go to Showroom



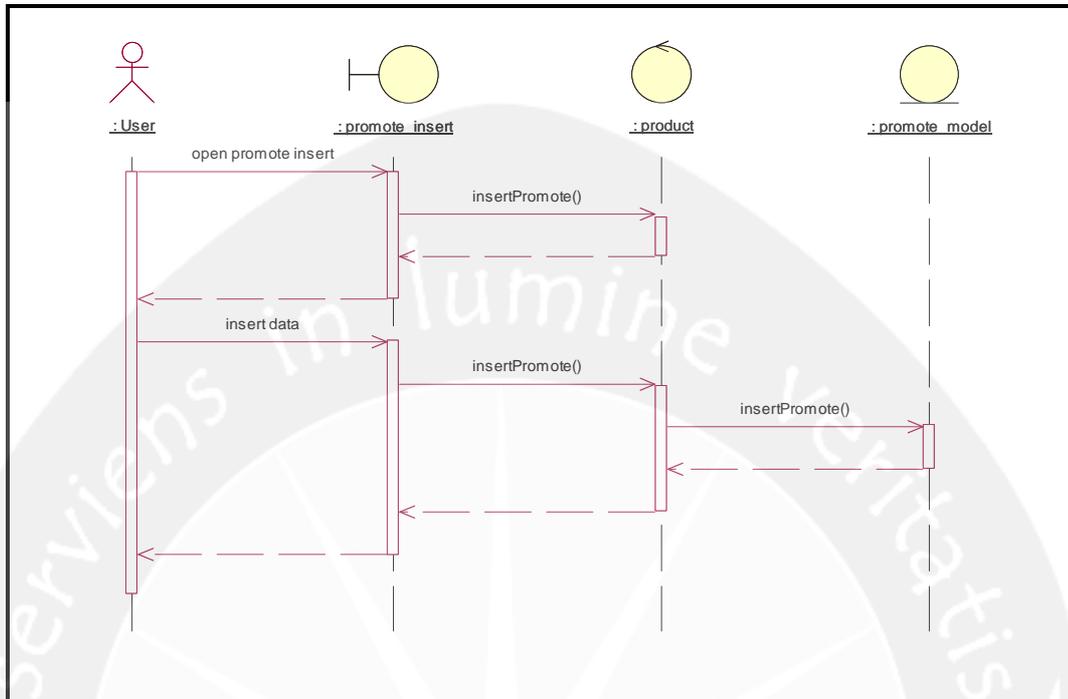
Gambar 2.28 Sequence Diagram : Login

2.2.1.16 Go to Lounge



Gambar 2.29 Sequence Diagram : Login

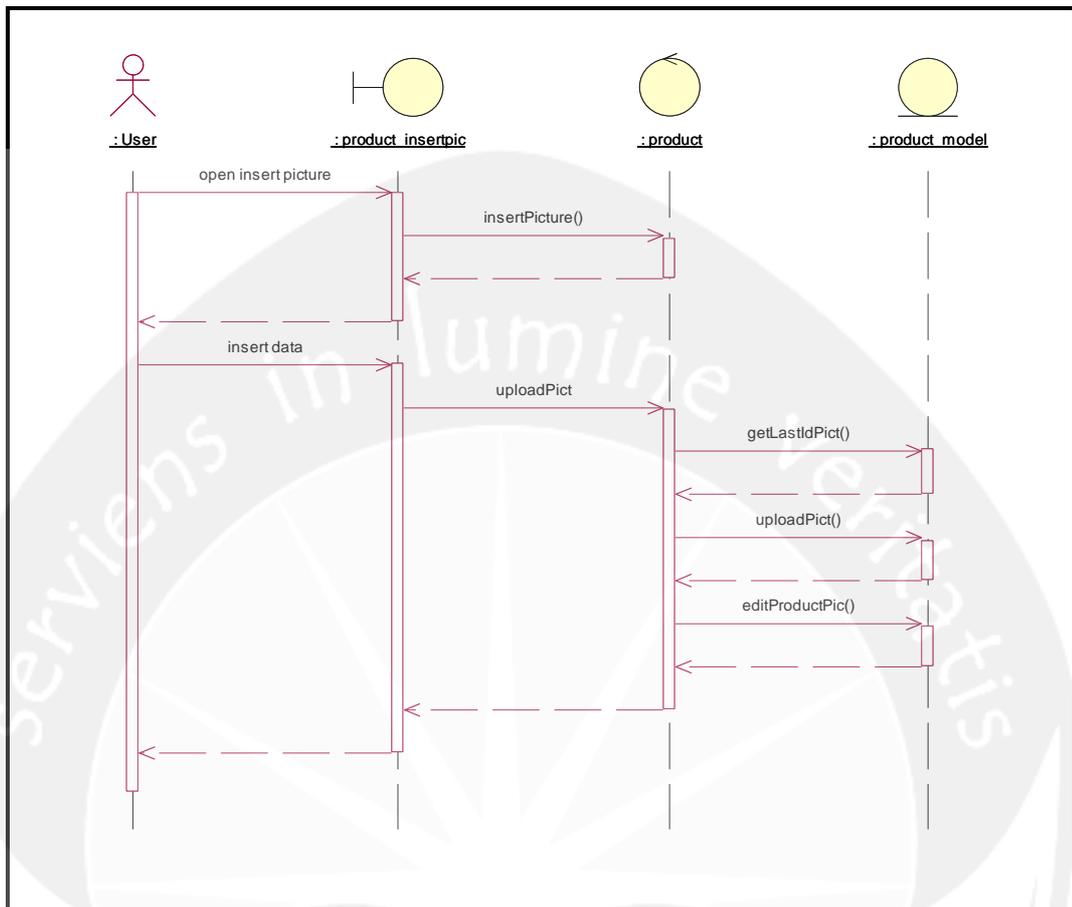
2.2.1.17 Insert Promote



Gambar 2.30 Sequence Diagram : Login

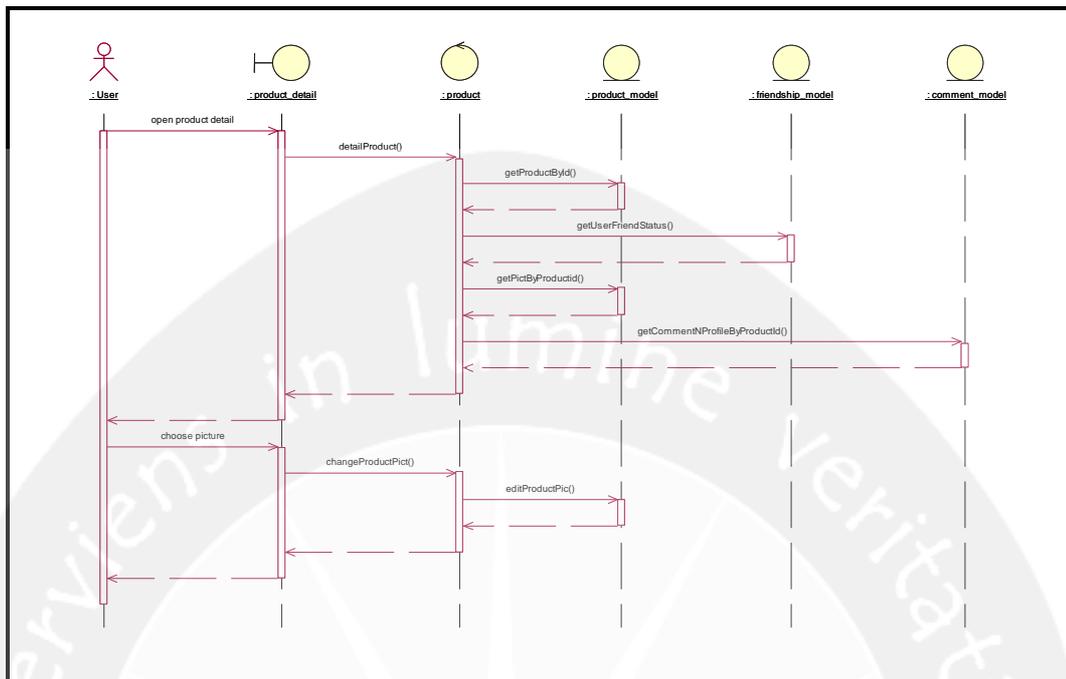
2.2.1.18 Pengelolaan Product Picture

2.2.1.18.1 Insert Product Picture



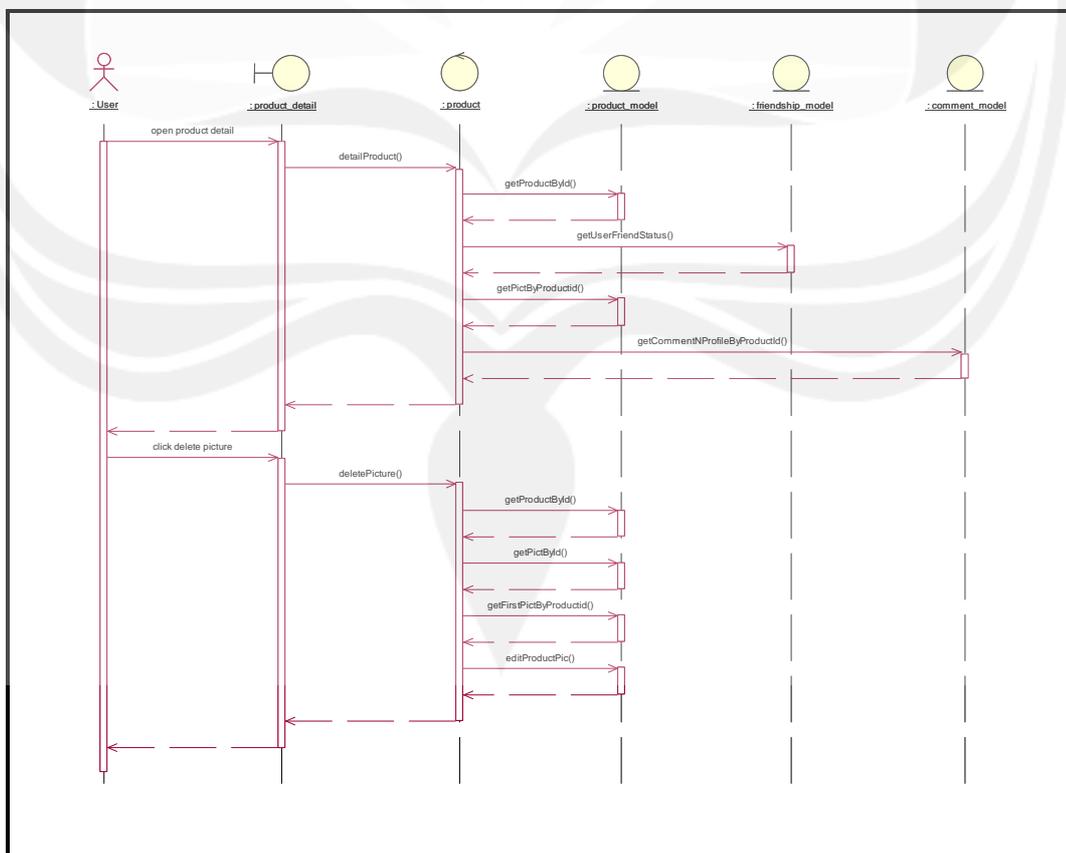
Gambar 2.31 Sequence Diagram : Login

2.2.1.18.2 Change Product Picture



Gambar 2.32 Sequence Diagram : Login

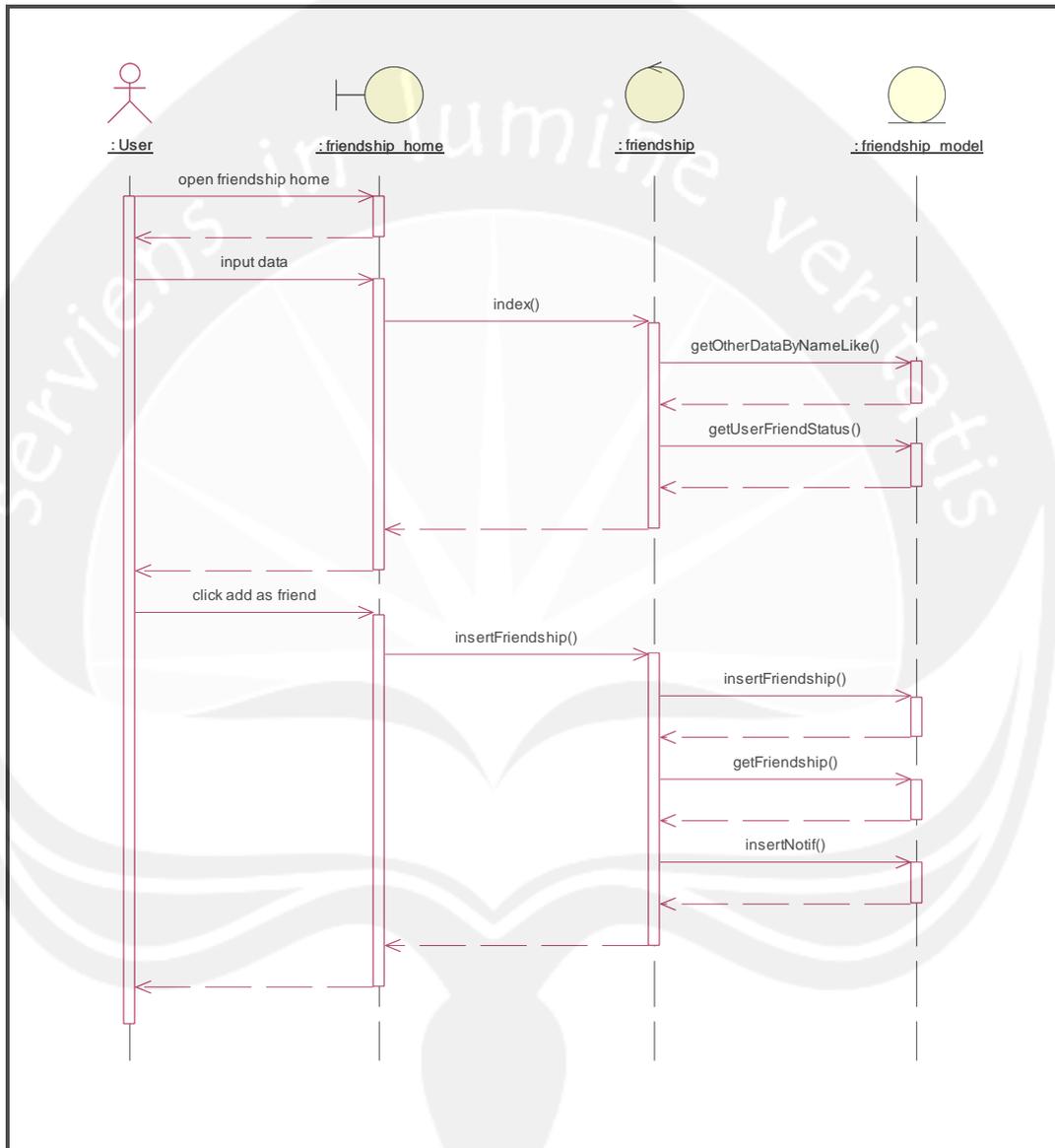
2.2.1.18.3 Delete Product Picture



Gambar 2.33 Sequence Diagram : Login

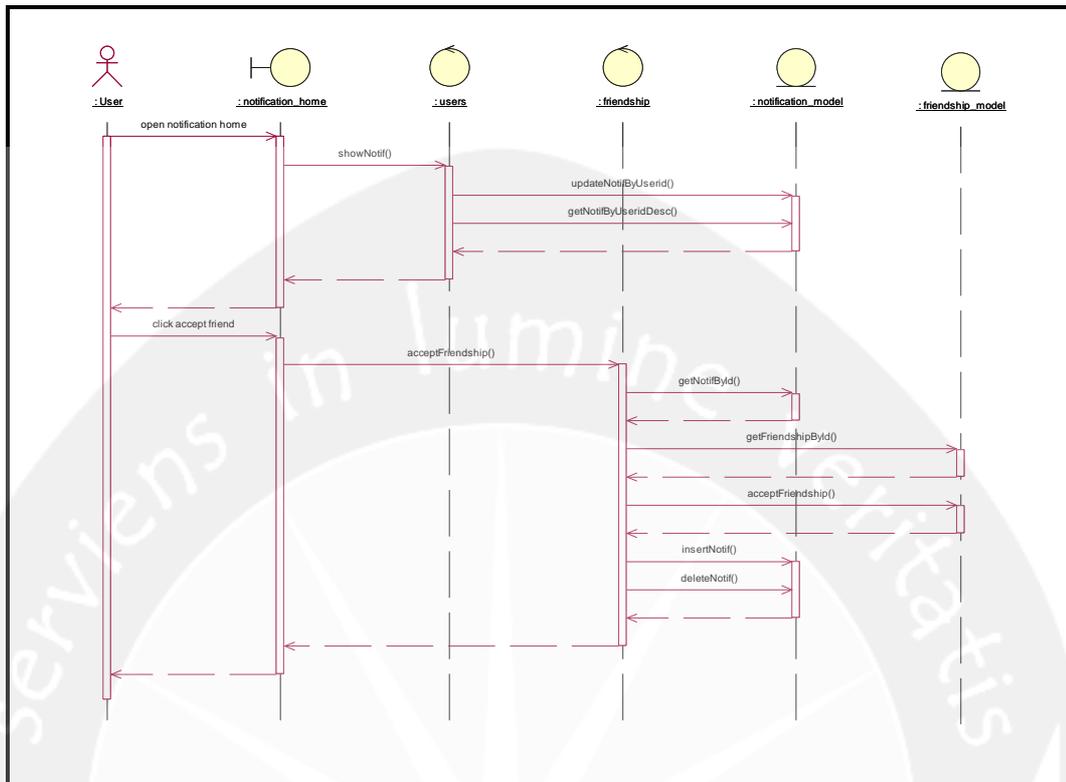
2.2.1.19 Pengelolaan Friendship

2.2.1.19.1 Insert Friendship



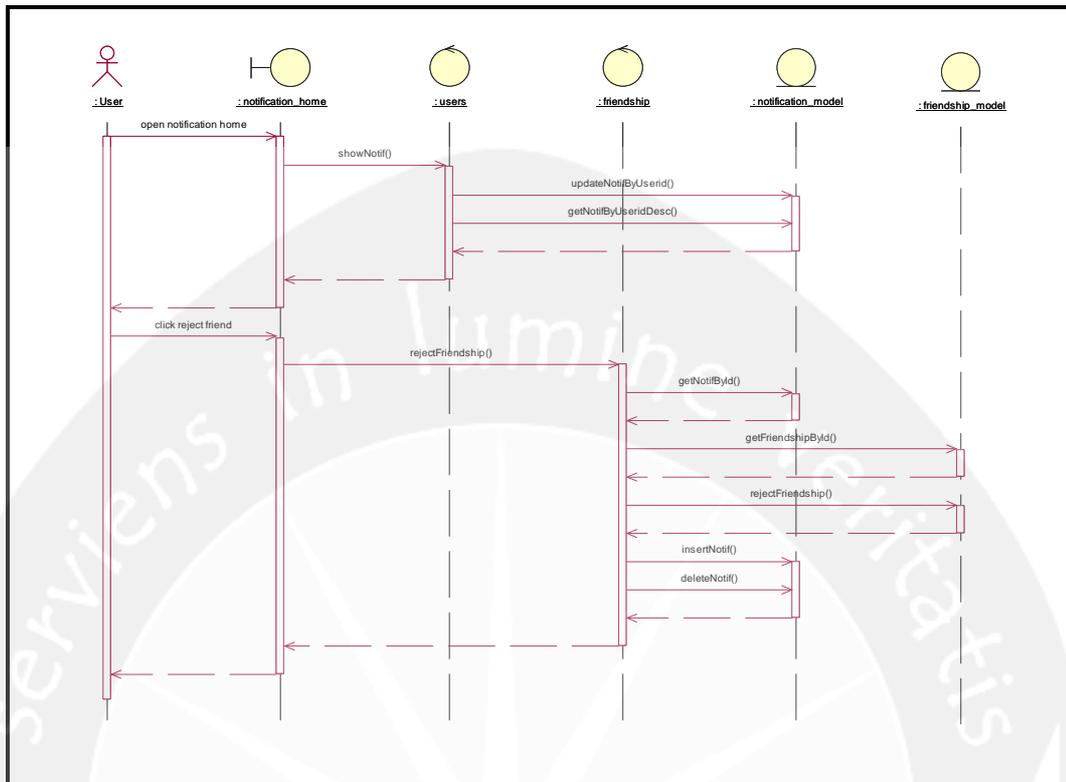
Gambar 2.34 Sequence Diagram : Login

2.2.1.19.2 Accept Friendship



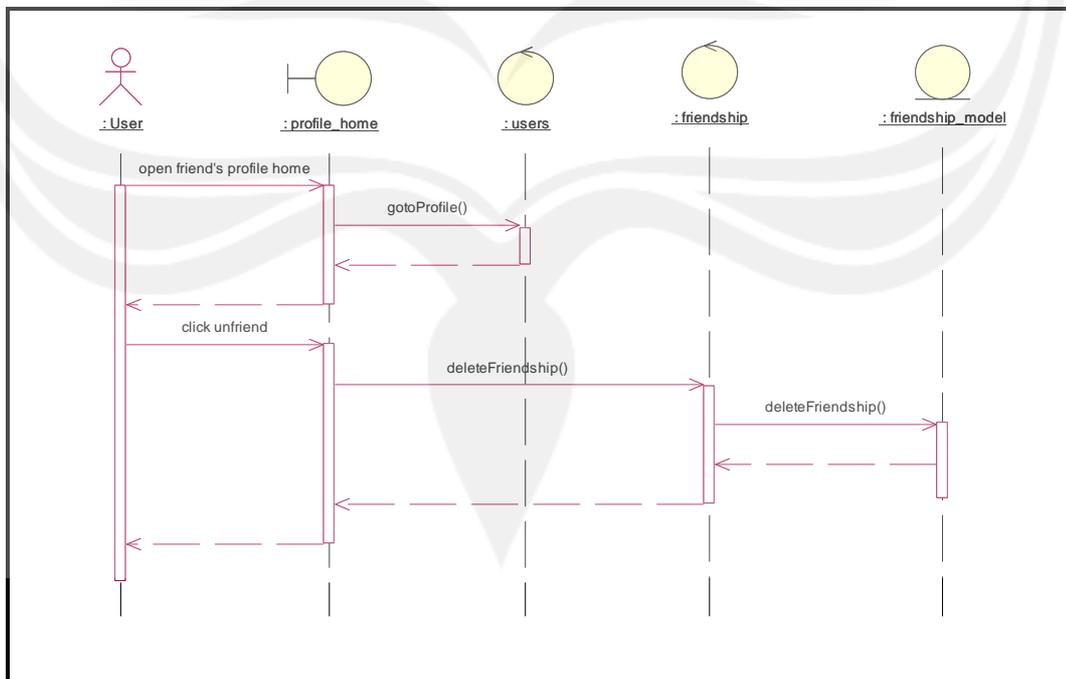
Gambar 2.35 Sequence Diagram : Login

2.2.1.19.3 Reject Friendship



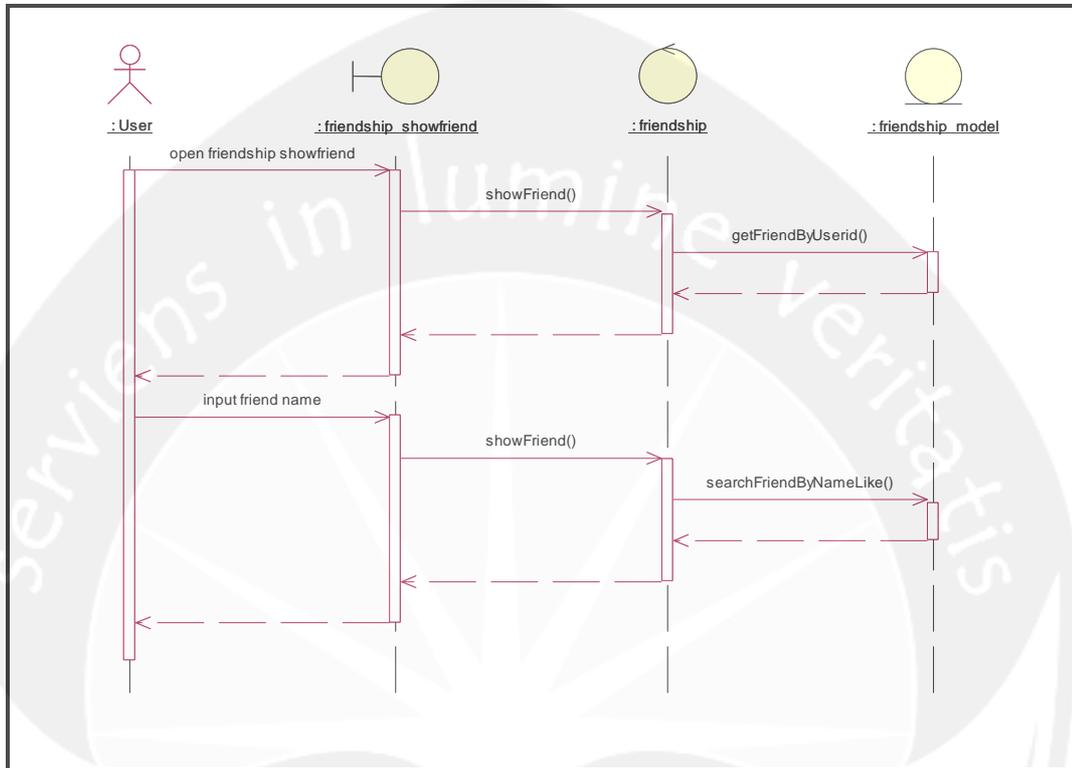
Gambar 2.36 Sequence Diagram : Login

2.2.1.19.4 Delete Friendship



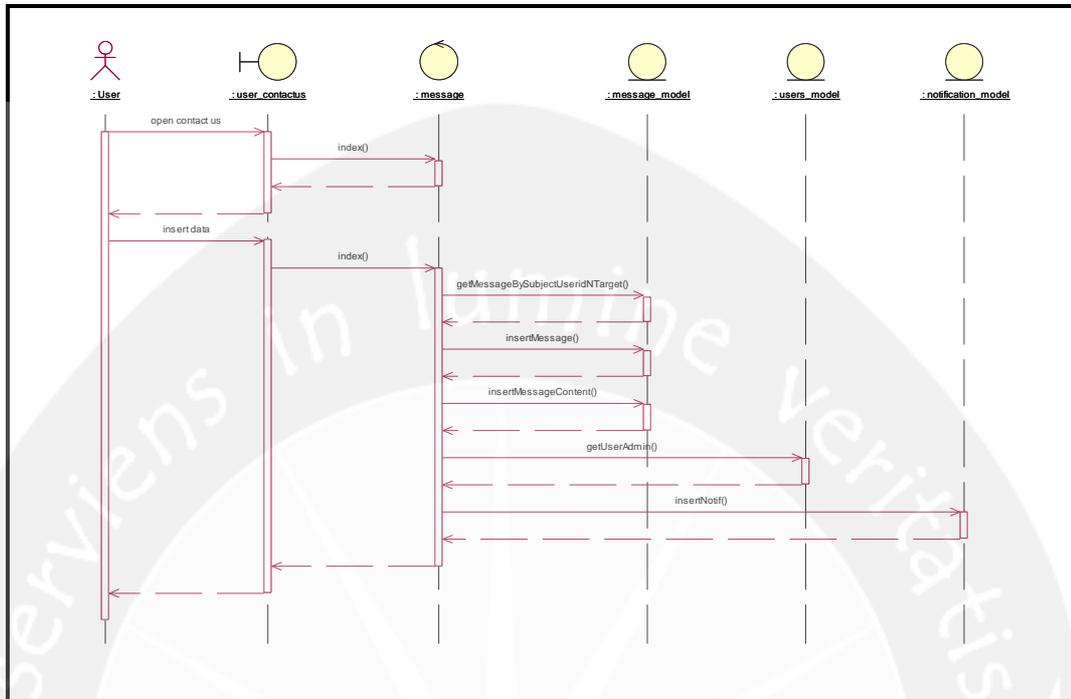
Gambar 2.37 Sequence Diagram : Login

2.2.1.19.5 Show Friend



Gambar 2.38 Sequence Diagram : Login

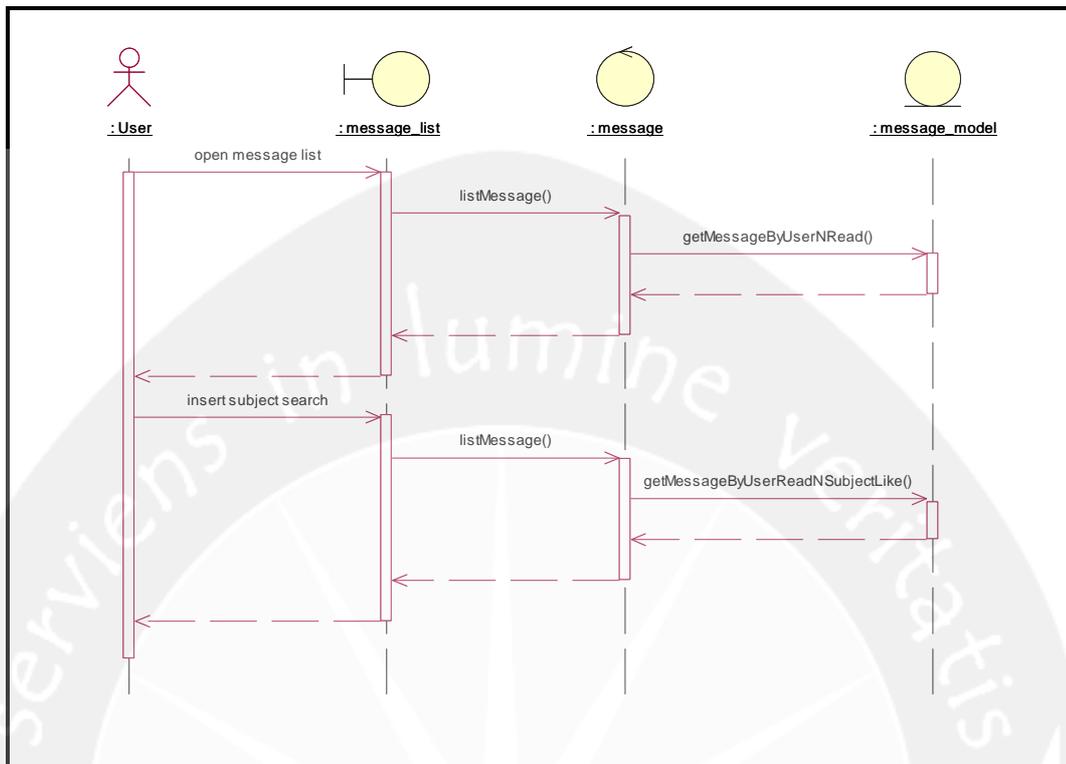
2.2.1.20 Contact Us



Gambar 2.39 Sequence Diagram : Login

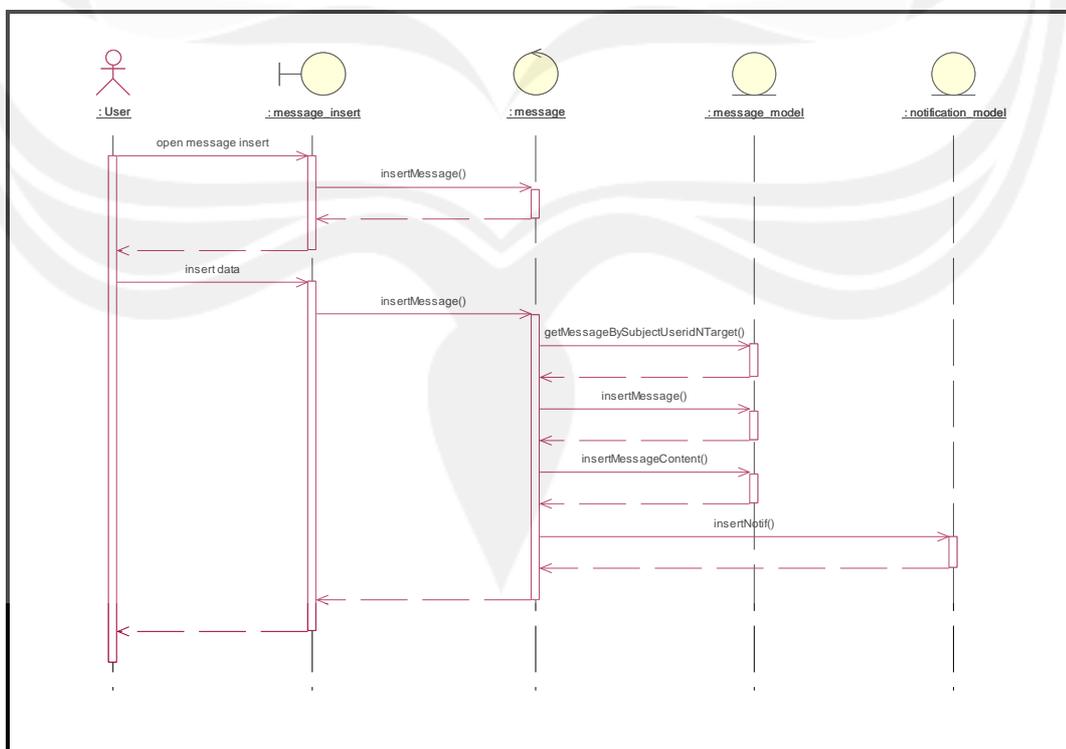
2.2.1.21 Pengelolaan Message

2.2.1.21.1 List Message



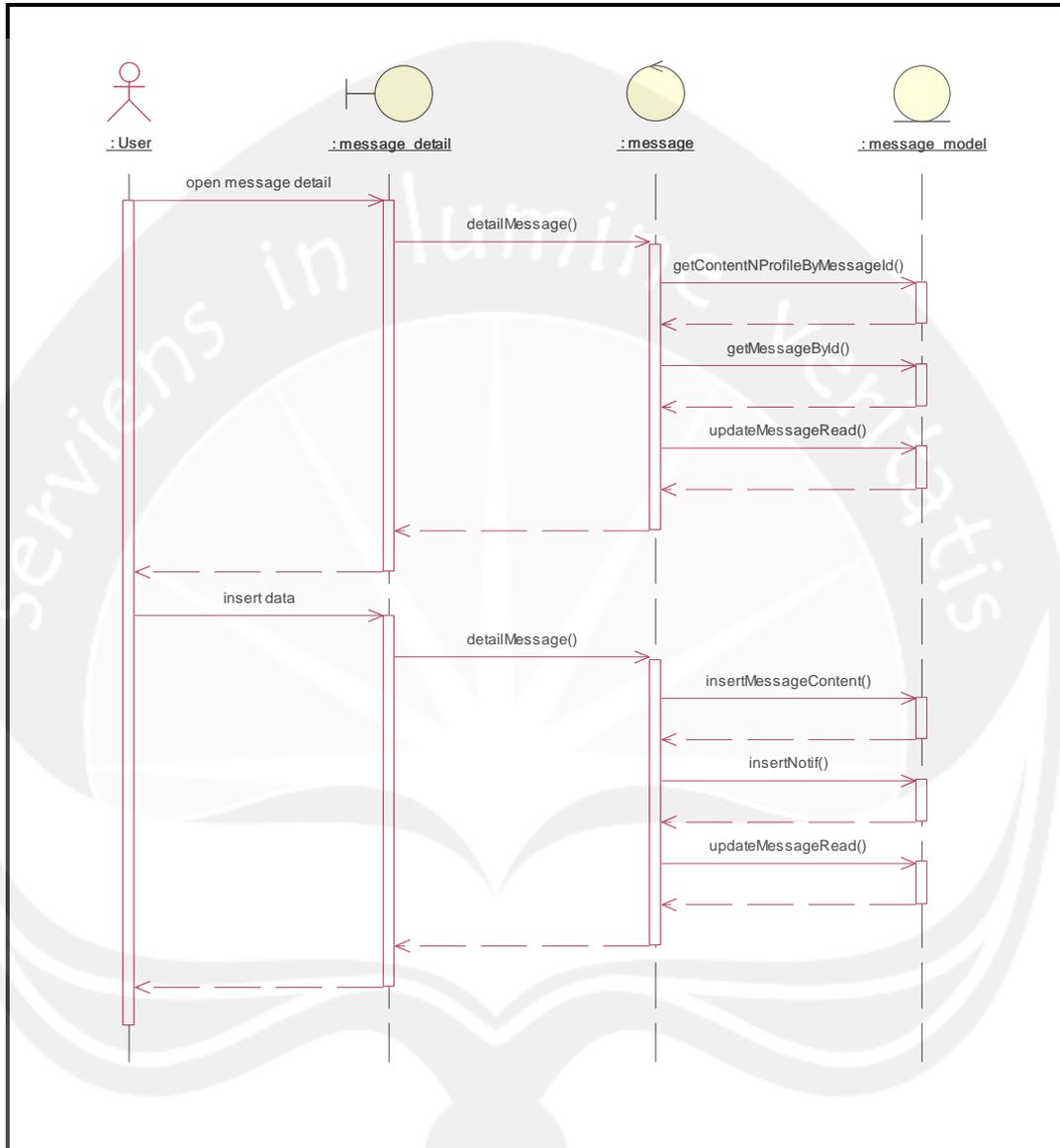
Gambar 2.40 Sequence Diagram : Login

2.2.1.21.2 Insert Message



Gambar 2.41 Sequence Diagram : Login

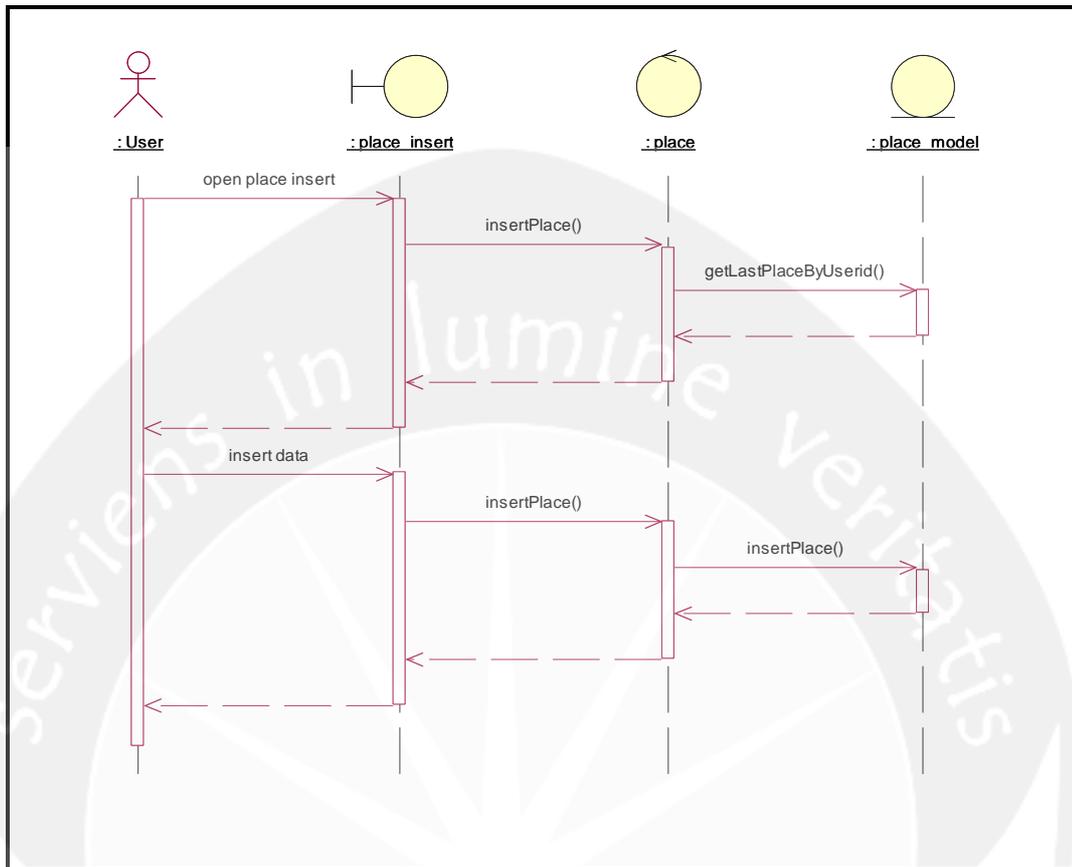
2.2.1.21.3 Detail Message



Gambar 2.42 Sequence Diagram : Login

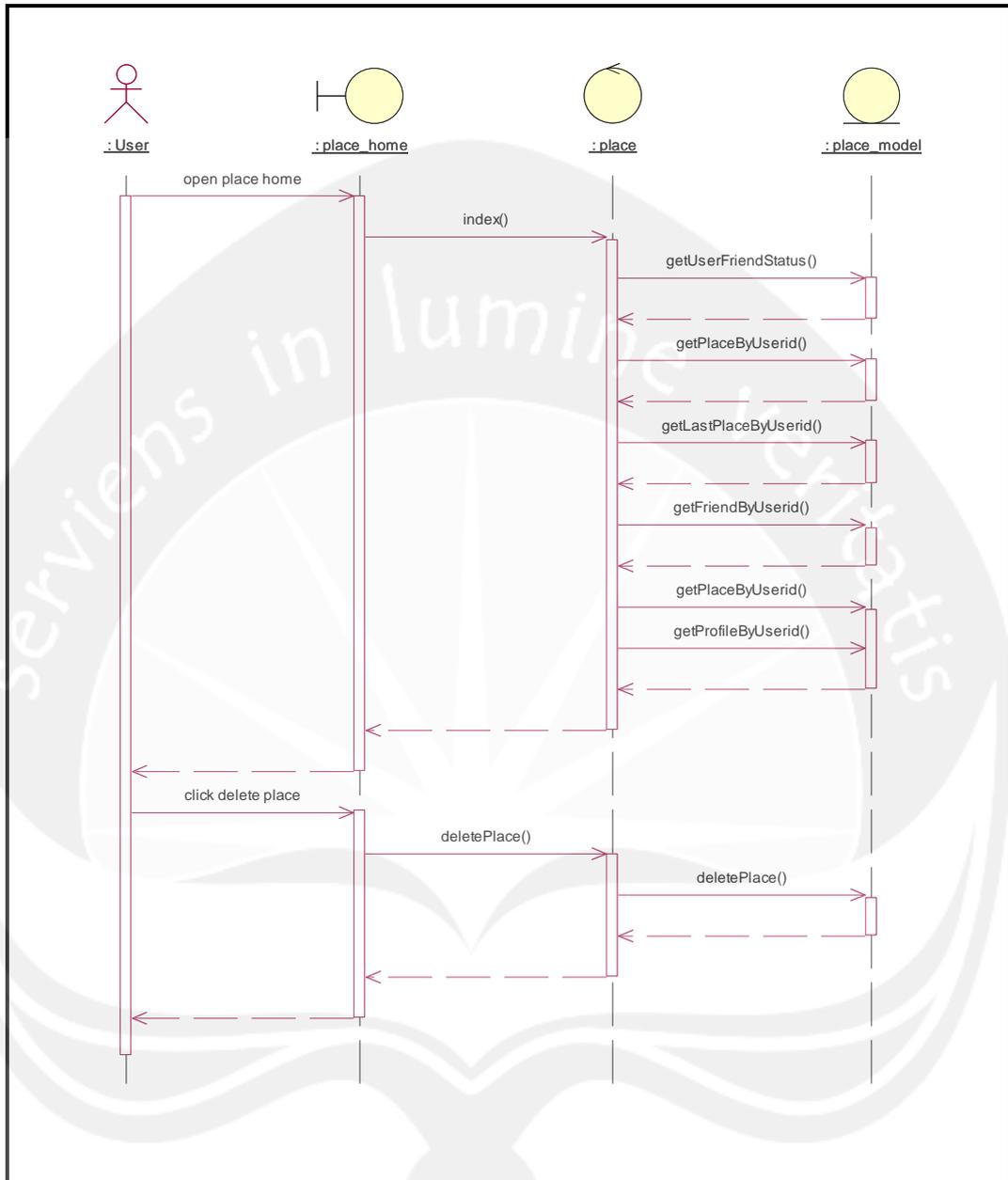
2.2.1.22 Pengelolaan Place

2.2.1.22.1 Insert Place



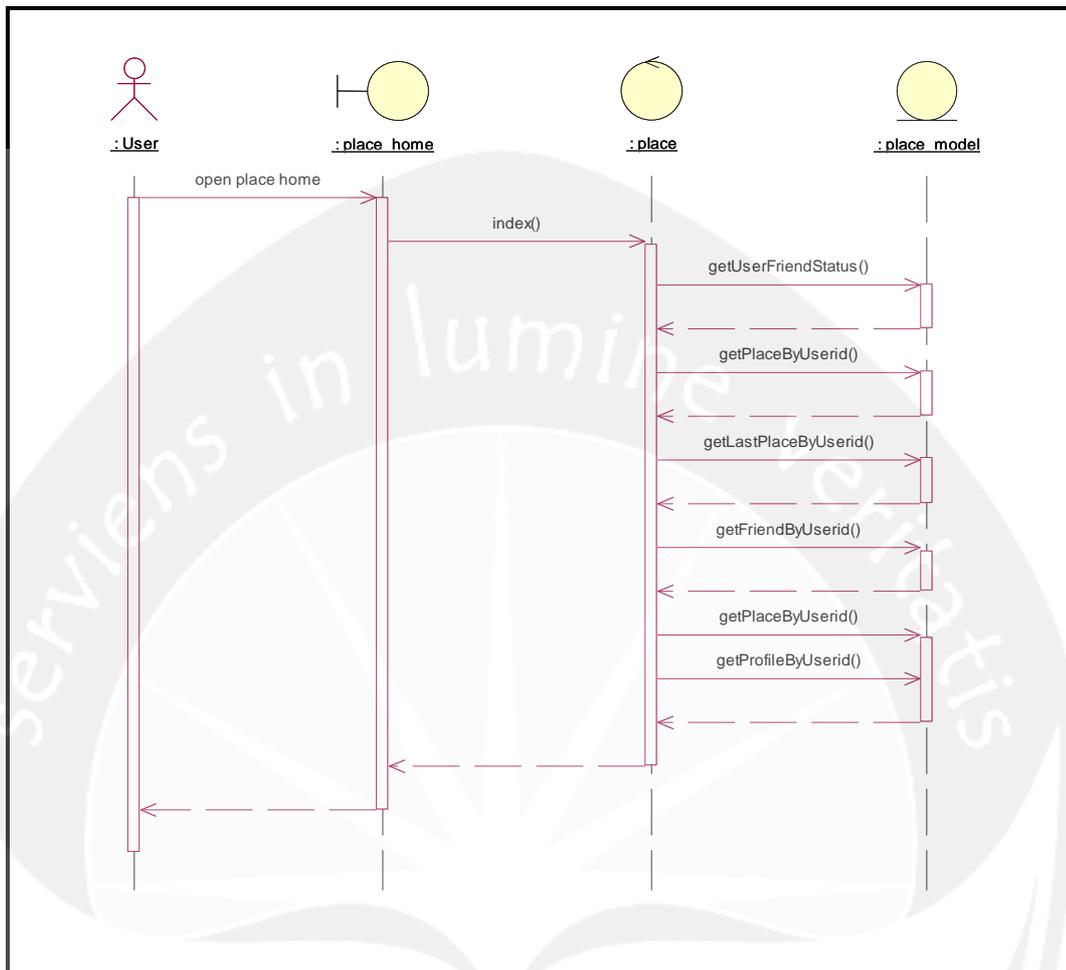
Gambar 2.43 Sequence Diagram : Login

2.2.1.22.2 Delete Place



Gambar 2.44 Sequence Diagram : Login

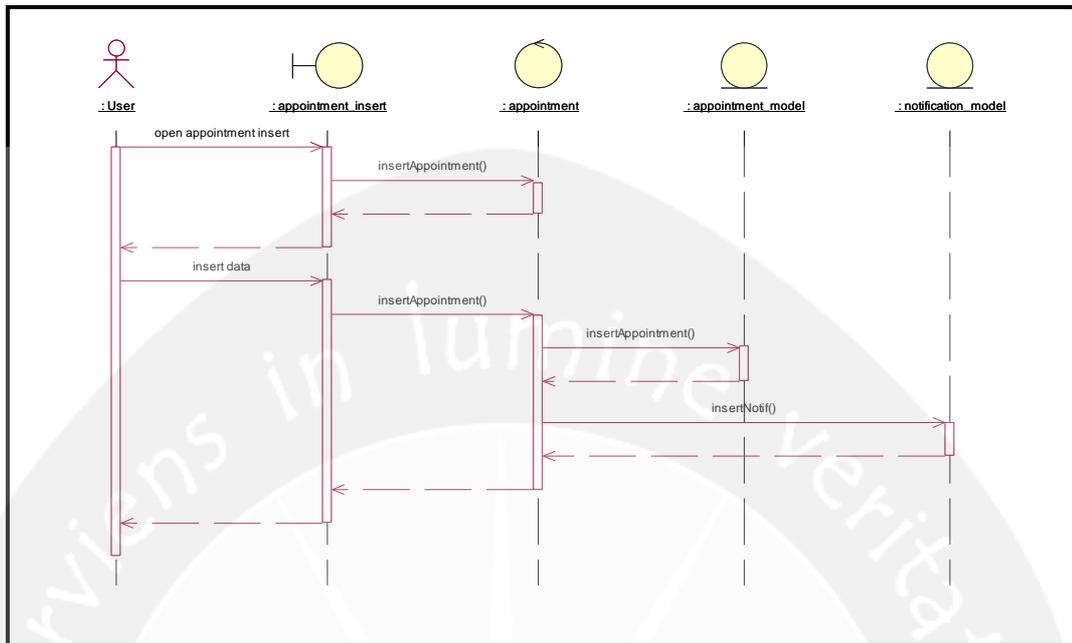
2.2.1.22.3 Show Place



Gambar 2.45 Sequence Diagram : Login

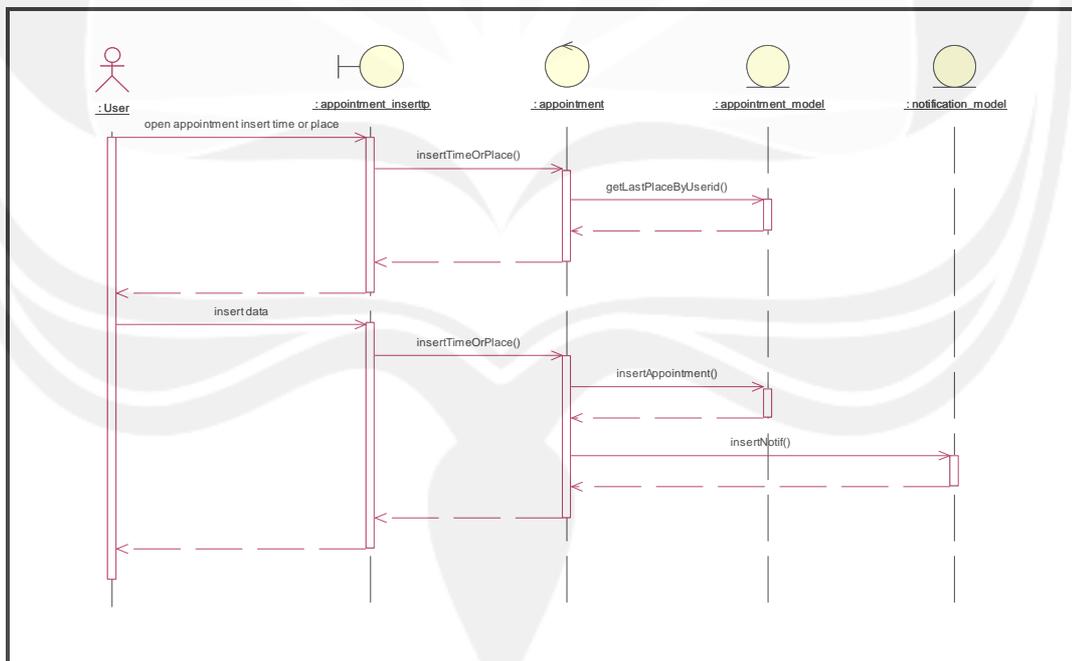
2.2.1.23 *Pengelolaan Appointment*

2.2.1.23.1 *Insert Appointment*



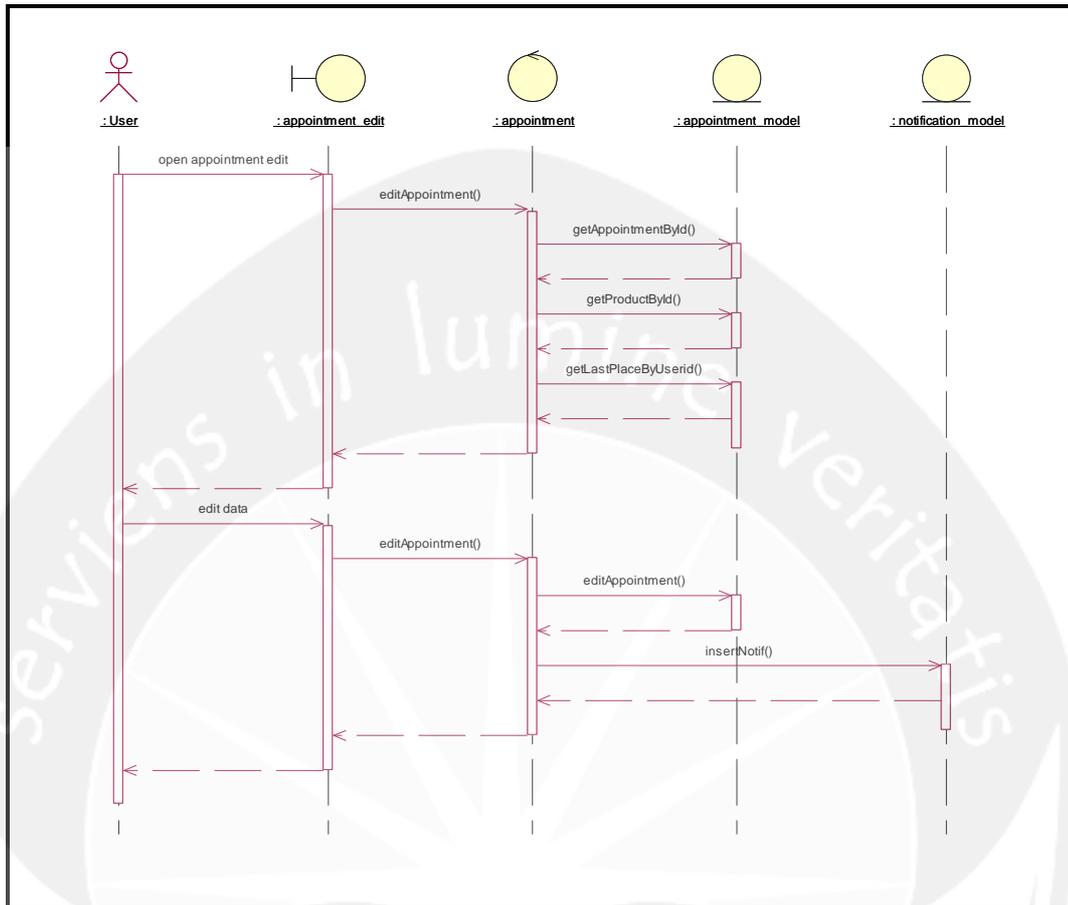
Gambar 2.46 Sequence Diagram : Login

2.2.1.23.2 Insert Time or Place Appointment



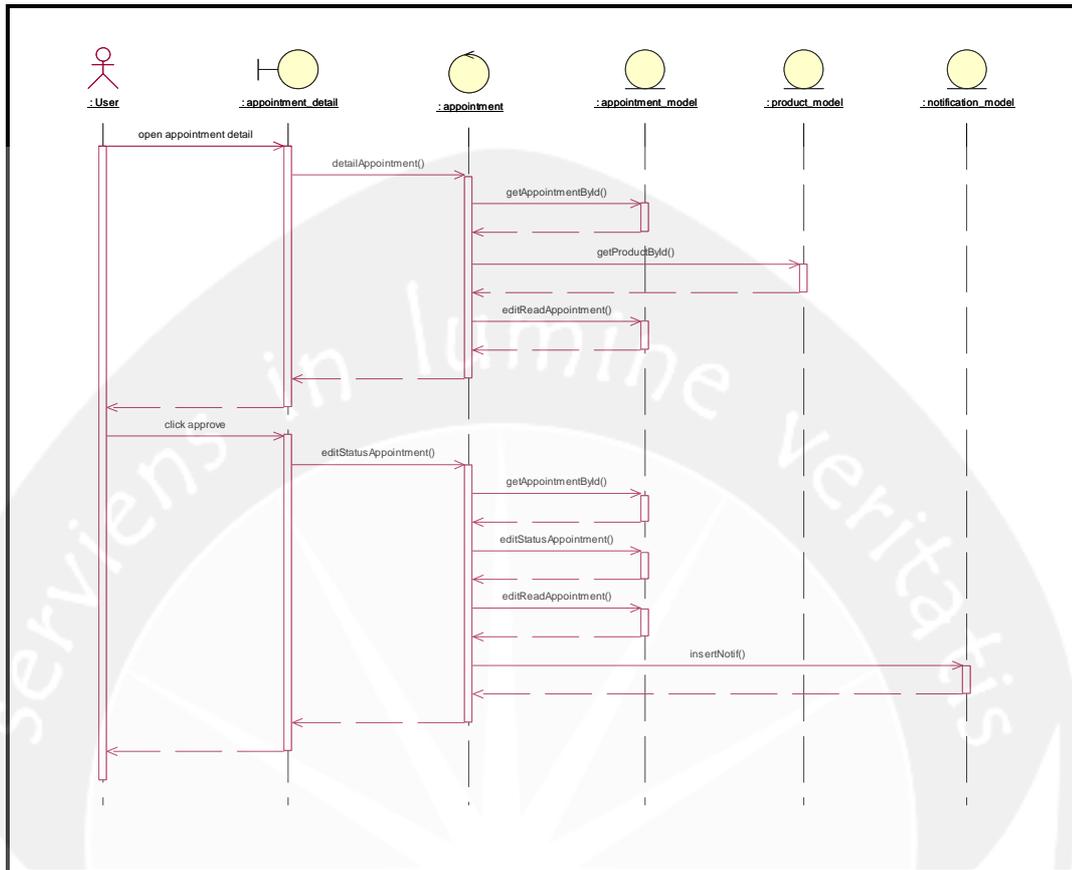
Gambar 2.47 Sequence Diagram : Login

2.2.1.23.3 Edit Appointment



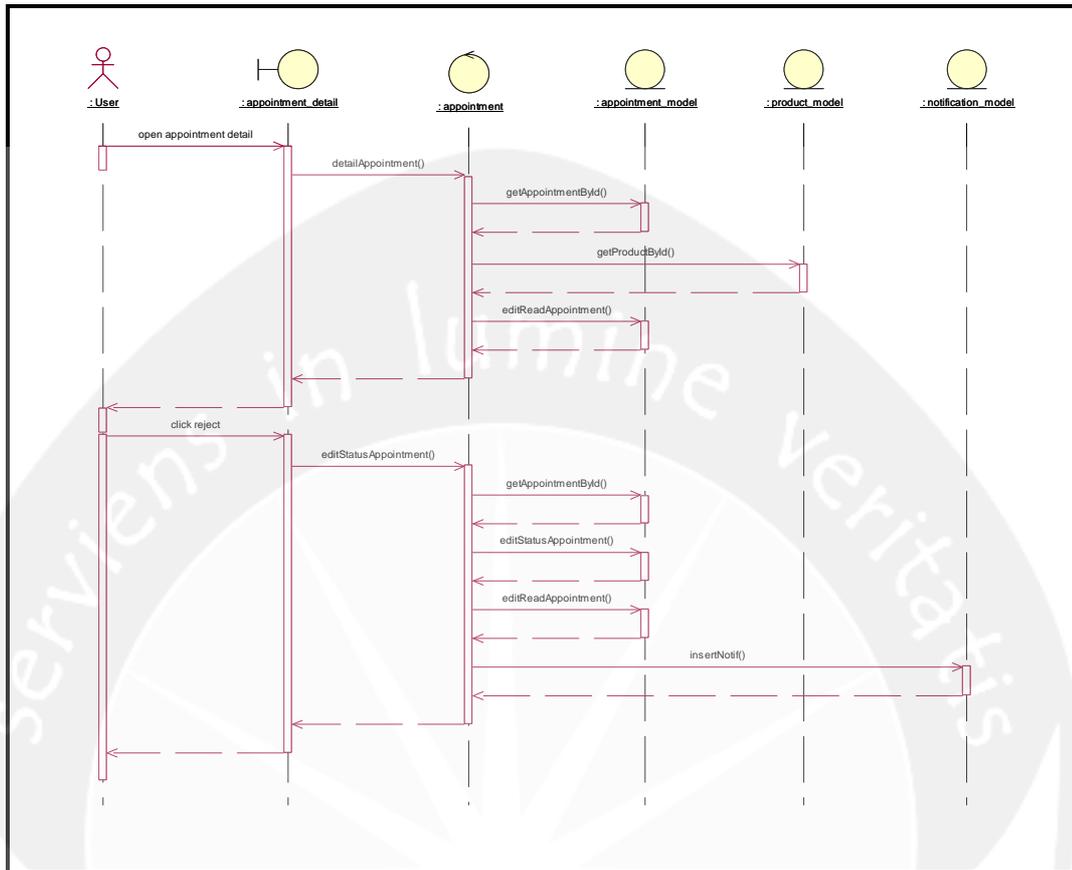
Gambar 2.48 Sequence Diagram : Login

2.2.1.23.4 Approve Appointment



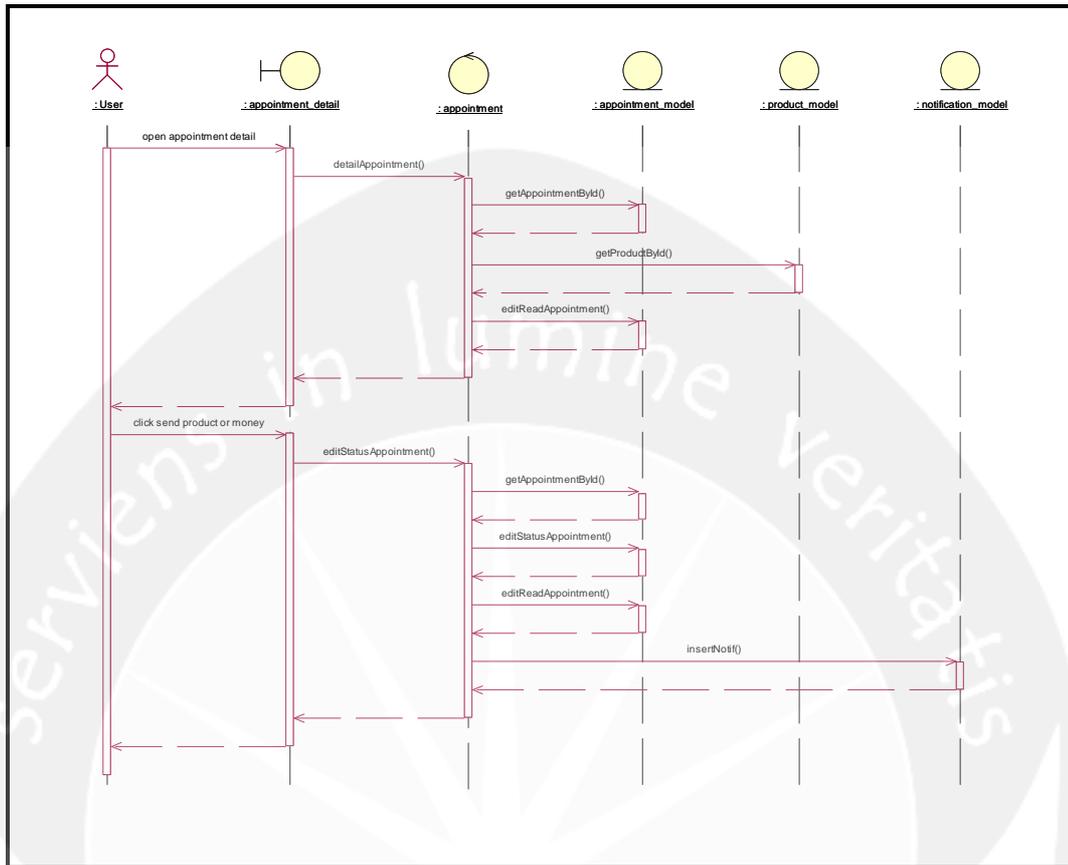
Gambar 2.49 Sequence Diagram : Login

2.2.1.23.5 Reject Appointment



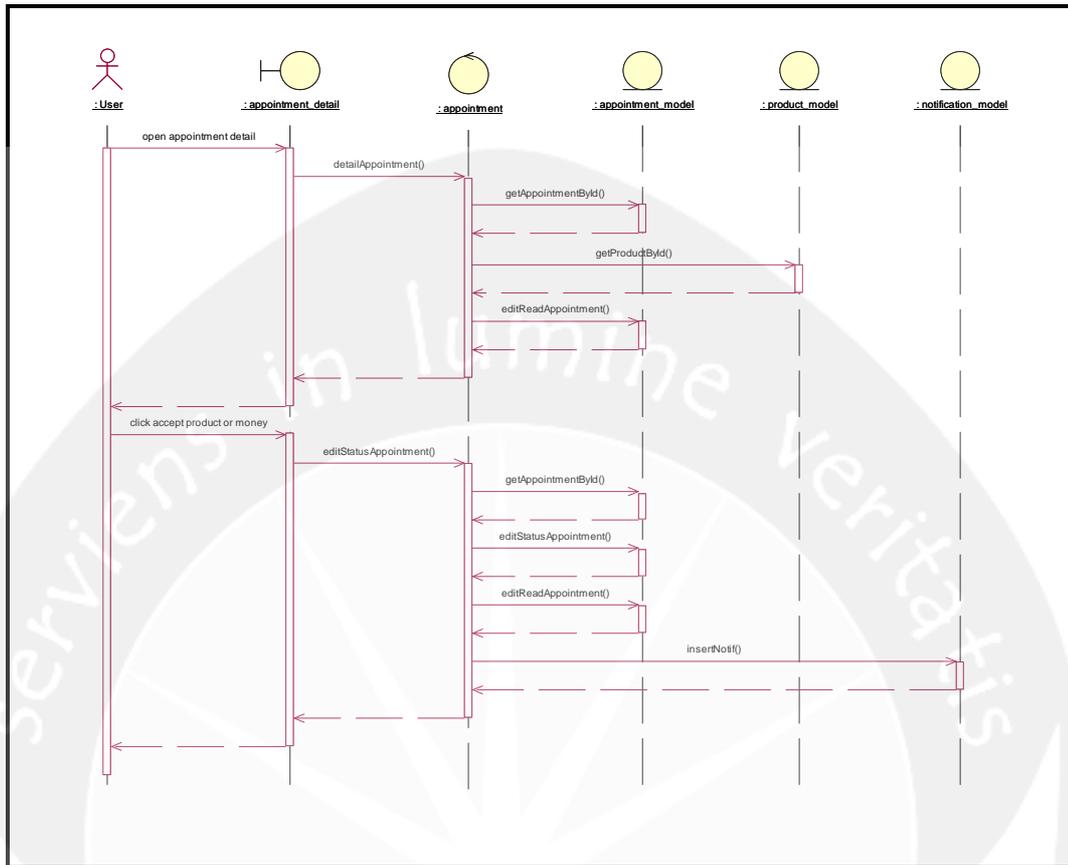
Gambar 2.50 Sequence Diagram : Login

2.2.1.23.6 Send Product Or Money



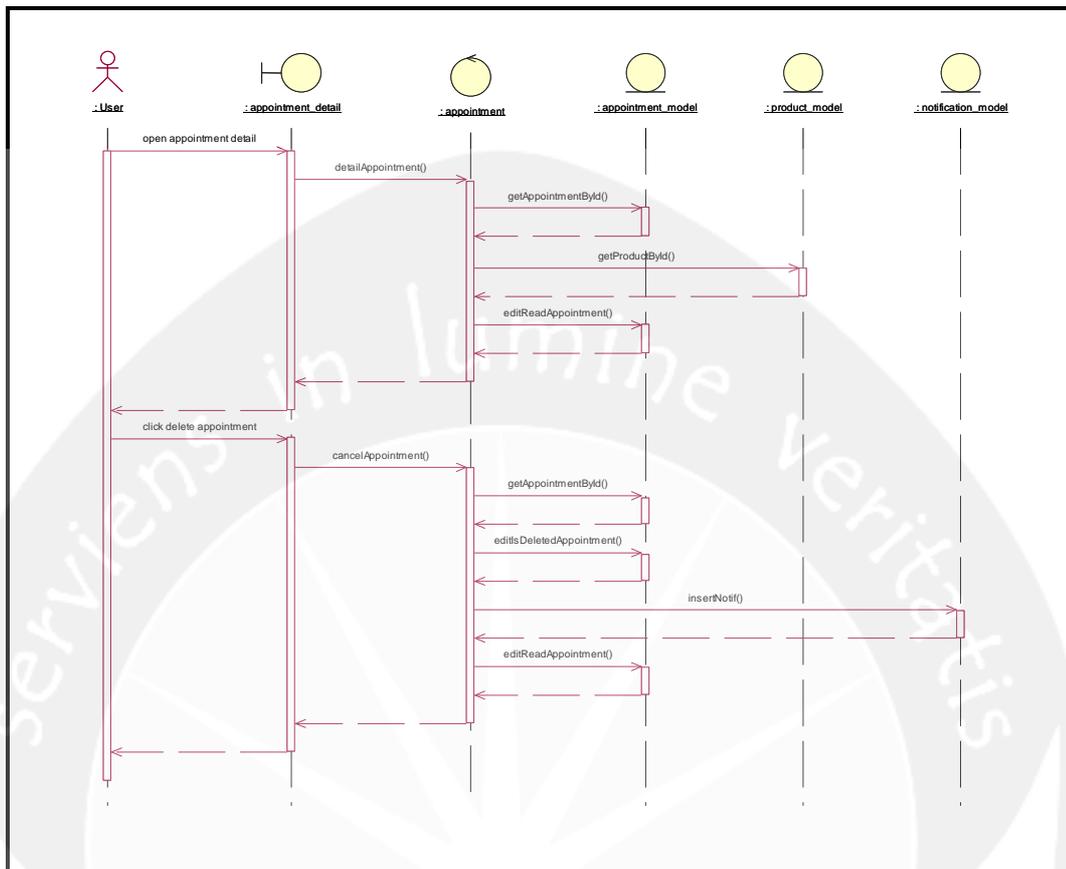
Gambar 2.51 Sequence Diagram : Login

2.2.1.23.7 Accept Product Or Money



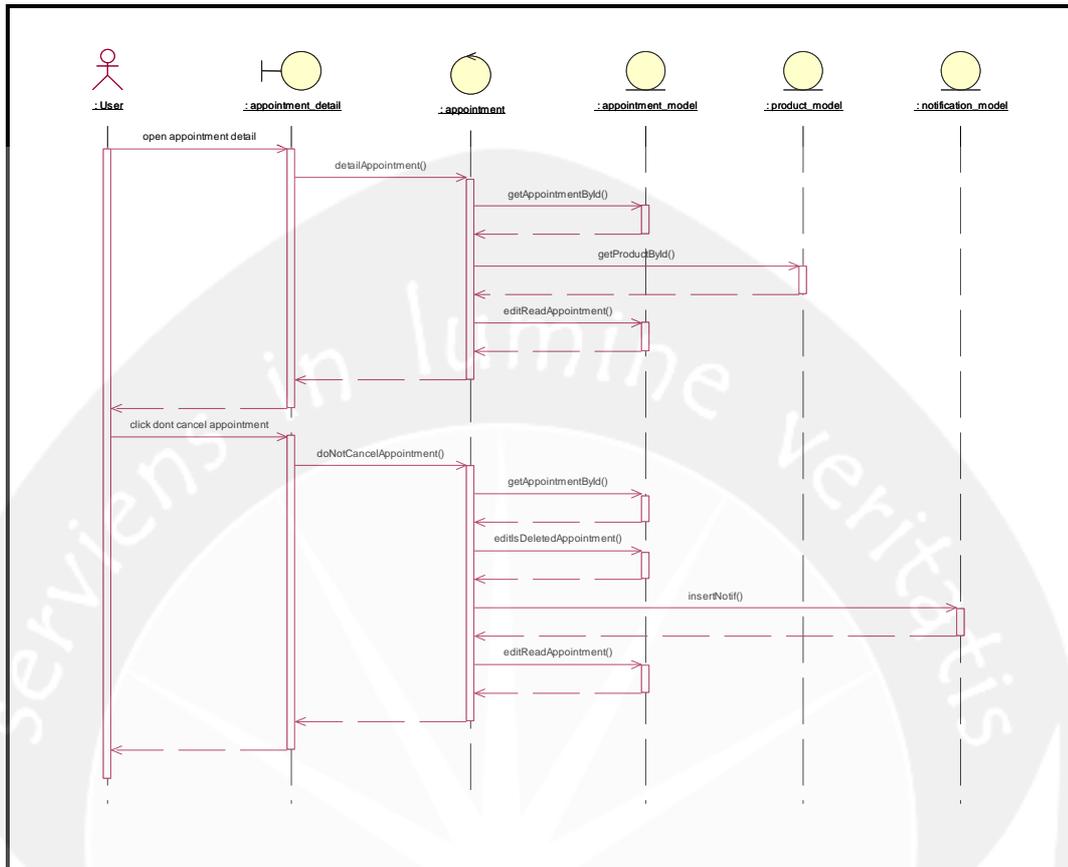
Gambar 2.52 Sequence Diagram : Login

2.2.1.23.8 Cancel Appointment



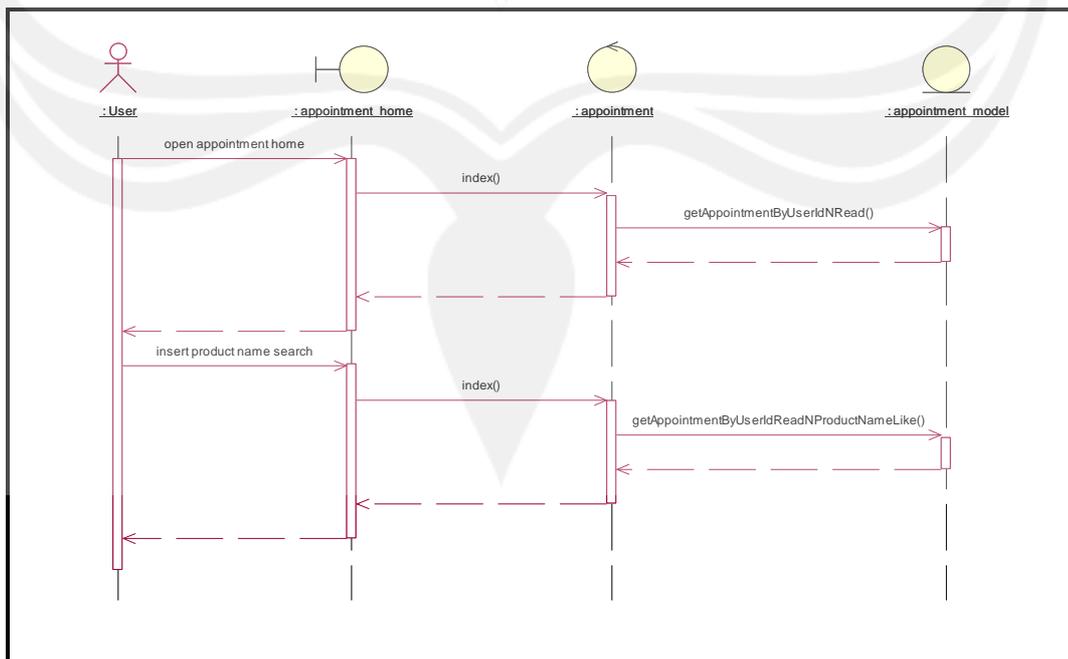
Gambar 2.53 Sequence Diagram : Login

2.2.1.23.9 Do not Appointment



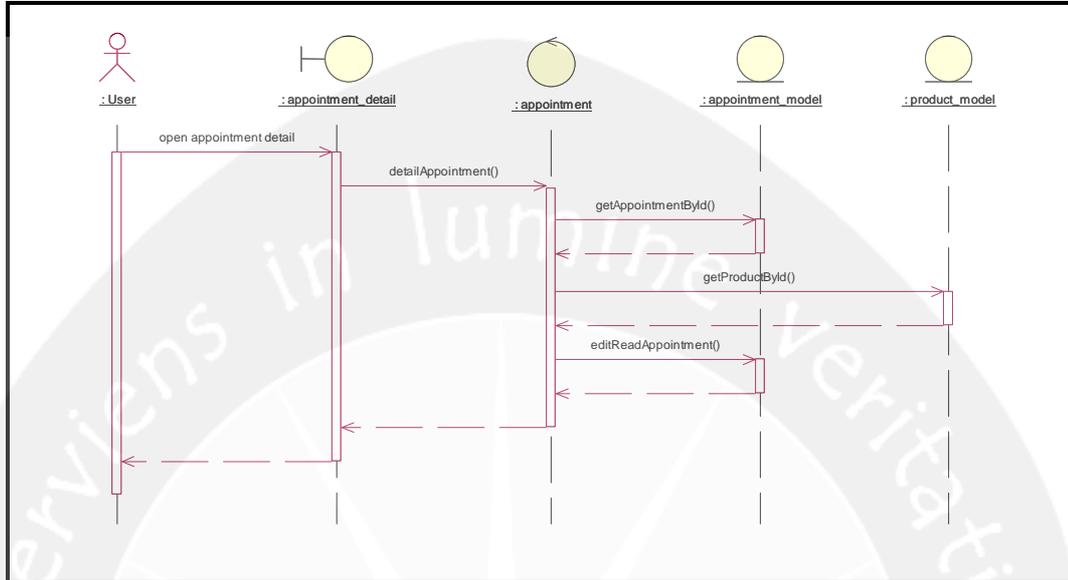
Gambar 2.54 Sequence Diagram : Login

2.2.1.23.10 Show Appointment



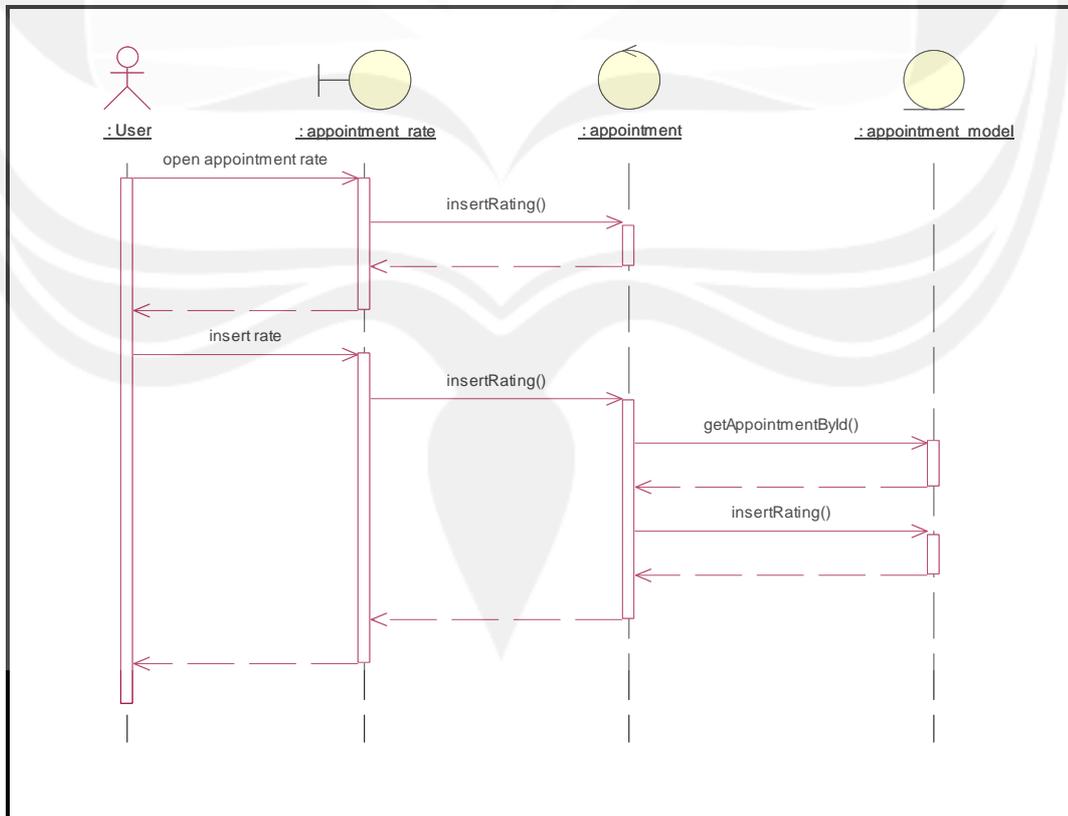
Gambar 2.55 Sequence Diagram : Login

2.2.1.23.11 Detail Appointment



Gambar 2.56 Sequence Diagram : Login

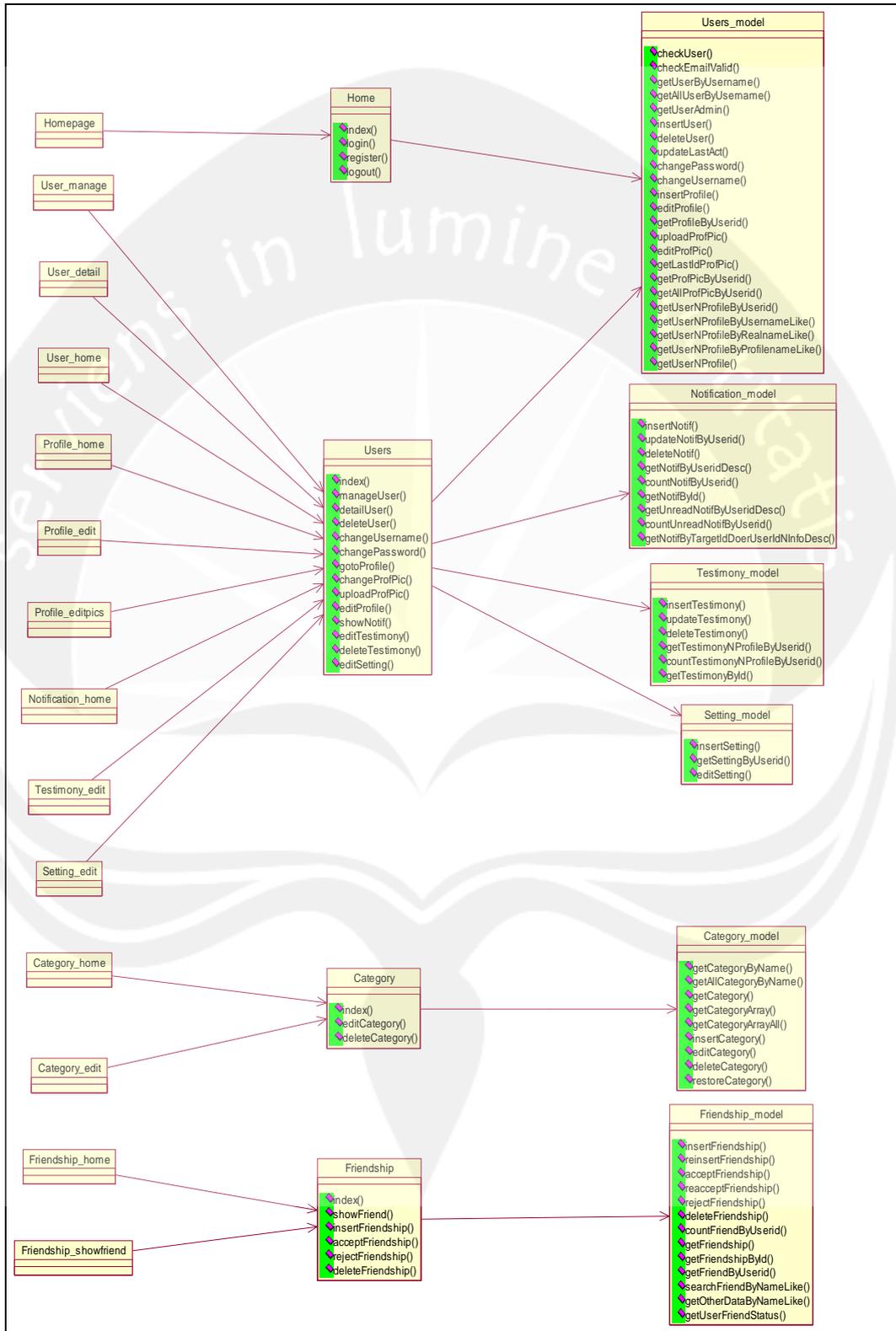
2.2.1.23.12 Rate Appointment

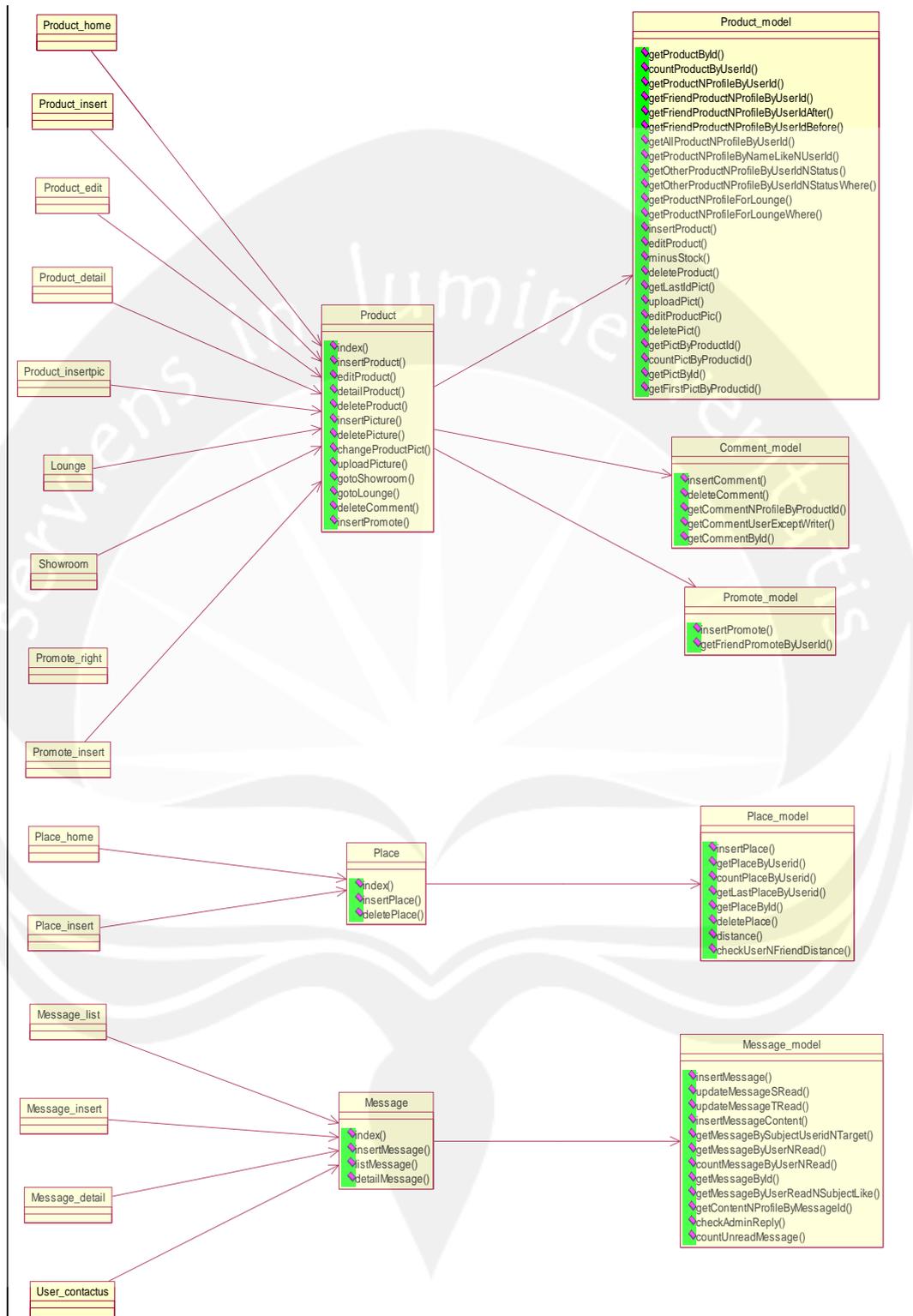


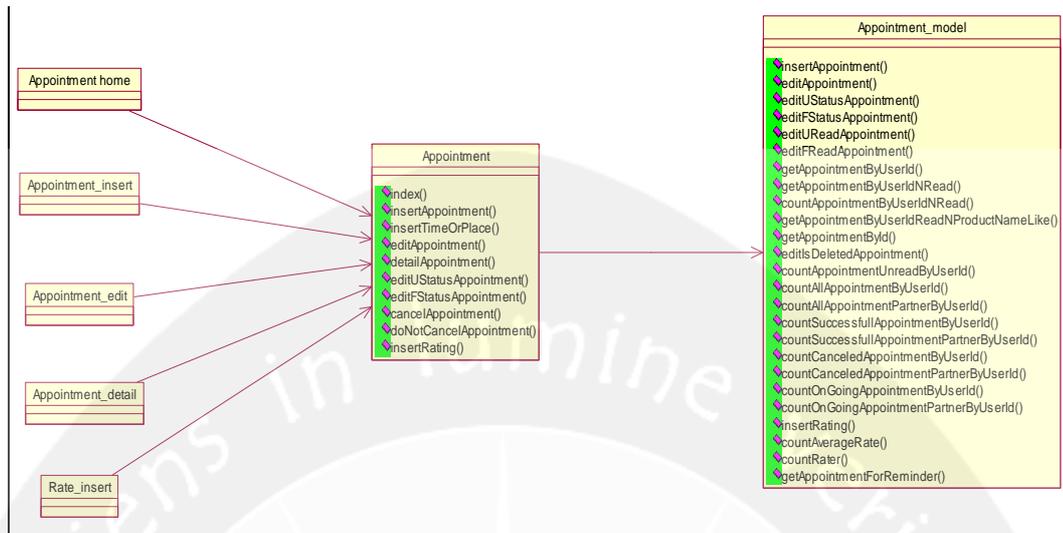
Gambar 2.57 Sequence Diagram : Login



2.2.2 Class Diagram







Gambar 2.58 Class Diagram

2.2.3 Spesifikasi Kelas Diagram

2.2.3.1 Spesifikasi Design Kelas Homepage

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

2.2.3.2 Spesifikasi Design Kelas User_manage

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

2.2.3.3 Spesifikasi Design Kelas User_detail

User_detail	<<boundary>>

+User_detail()

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

2.2.3.4 Spesifikasi Design Kelas User_home

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

2.2.3.5 Spesifikasi Design Kelas Profile_home

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

2.2.3.6 Spesifikasi Design Kelas Profile_edit

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

2.2.3.7 Spesifikasi Design Kelas Profile_editpics

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

2.2.3.8 Spesifikasi Design Kelas Notification_home

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

2.2.3.9 Spesifikasi Design Kelas Testimony_edit

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

2.2.3.10 Spesifikasi Design Kelas Setting_edit

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

2.2.3.11 Spesifikasi Design Kelas Category_home

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

2.2.3.12 Spesifikasi Design Kelas Category_edit

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

2.2.3.13 Spesifikasi Design Kelas Friendship_home

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

2.2.3.14 Spesifikasi Design Kelas Friendship_showfriend

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

2.2.3.15 Spesifikasi Design Kelas Product_home

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

2.2.3.16 Spesifikasi Design Kelas Product_insert

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

2.2.3.17 Spesifikasi Design Kelas Product_edit

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

2.2.3.18 Spesifikasi Design Kelas Product_detail

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

2.2.3.19 Spesifikasi Design Kelas Product_insertpic

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

2.2.3.20 Spesifikasi Design Kelas Lounge

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

2.2.3.21 Spesifikasi Design Kelas Showroom

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

2.2.3.22 Spesifikasi Design Kelas Promote_right

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

2.2.3.23 Spesifikasi Design Kelas Promote_insert

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

2.2.3.24 Spesifikasi Design Kelas Place_home

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

2.2.3.25 Spesifikasi Design Kelas Place_insert

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

2.2.3.26 Spesifikasi Design Kelas Message_list

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

2.2.3.27 Spesifikasi Design Kelas Message_insert

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

2.2.3.28 Spesifikasi Design Kelas Message_detail

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

2.2.3.29 Spesifikasi Design Kelas User_contactus

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

2.2.3.30 Spesifikasi Design Kelas Appointment_home

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

2.2.3.31 Spesifikasi Design Kelas Appointment_insert

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

2.2.3.32 Spesifikasi Design Kelas Appointment_edit

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

2.2.3.33 Spesifikasi Design Kelas Appointment_detail

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

2.2.3.34 Spesifikasi Design Kelas Rate_insert

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

2.2.3.35 Spesifikasi Design Kelas Home

Home	<<control>>
-------------	--------------------------------

+__construct()

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

+index() : void

Operasi ini digunakan untuk menampilkan halaman login atau register apabila pengguna belum login.

+login() : void

Operasi ini digunakan untuk mengambil data login yang diinputkan oleh pengguna, yaitu username dan password kemudian di cek kecocokannya dengan data user di database.

+register() : void

Operasi ini digunakan untuk mendaftarkan pengguna ke situs SOS dengan mengambil data register yang diinputkan oleh pengguna, yaitu username, password, realname, profilename, city, province, email, dan birthdate yang kemudian akan disimpan di database.

+logout() : void

Operasi ini digunakan untuk mengakhiri session yang sedang berjalan.

2.2.3.36 Spesifikasi Design Kelas Users

Users	<<control>>
+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+index() : void Operasi ini digunakan untuk menampilkan semua produk	

yang dimiliki oleh pengguna dan pengguna lainnya yang telah berteman dengan pengguna.

+manageUser() : void

Operasi ini digunakan untuk menampilkan semua user yang ada di database. Operasi ini memungkinkan pengguna mencari user situs SOS berdasarkan username, realname, atau profile name

+detailUser(u_id) : void

Operasi ini digunakan untuk menampilkan detil pengguna berdasarkan u_id.

+deleteUser(u_id) : void

Operasi ini digunakan untuk menghapus pengguna berdasarkan u_id.

+changeUsername() : void

Operasi ini digunakan untuk mengubah username menjadi username yang baru sesuai dengan inputan user.

+changePassword() : void

Operasi ini digunakan untuk mengubah password menjadi password yang baru sesuai dengan inputan user.

+gotoProfile(u_id) : void

Operasi ini digunakan untuk mengarahkan tampilan ke halaman profile_home. Operasi ini juga dapat digunakan untuk mengambil data testimony yang diinputkan user yaitu text dan menyimpannya ke database.

+changeProfPic(pp_name) : void

Operasi ini digunakan untuk mengubah profile picture pengguna yang ada di database menjadi pp_name.

+uploadProfPic() : void

Operasi ini digunakan untuk mengupload gambar ke server kemudian mengubah profile picture pengguna yang ada di database menjadi gambar yang baru diupload.

+editProfile() : void

Operasi ini digunakan untuk mengubah data Profile yaitu realname, profilename, city, province, email, birthdate, dan aboutme yang ada didatabase menjadi sesuai dengan inputan user.

+showNotif() : void

Operasi ini digunakan untuk menampilkan semua notifikasi yang dimiliki pengguna yang terdapat di database.

+editTestimony(t_id) : void

Operasi ini digunakan untuk mengubah data testimony yaitu text yang ada di database sesuai dengan t_id.

+deleteTestimony(t_id) : void

Operasi ini digunakan untuk menghapus testimony yang ada di database berdasarkan t_id.

+editSetting() : void

Operasi ini digunakan untuk mengubah data Setting yaitu show realname, show birthdate, show email, show city, show province, show zero stock, dan show successful appointment yang ada didatabase menjadi sesuai dengan inputan user.

2.2.3.37 Spesifikasi Design Kelas Category

Category	<<control>>
+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+index() : void Operasi ini digunakan untuk menampilkan semua category	

yang terdapat didatabase. Operasi ini juga dapat digunakan untuk mengambil data category yang diinputkan user yaitu category kemudian menyimpannya ke database.

+editCategory(category) : void

Operasi ini digunakan untuk mengubah data category yaitu nama category menjadi sesuai dengan inputan user berdasarkan category.

+deleteCategory(category) : void

Operasi ini digunakan untuk menghapus category berdasarkan category.

2.2.3.38 Spesifikasi Design Kelas Friendship

Friendship	<<control>>
<p>+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+index() : void Operasi ini digunakan untuk melakukan pencarian pengguna situs SOS berdasarkan profile name yang diinputkan user.</p> <p>+showFriend(u_id) : void Operasi ini digunakan untuk menampilkan semua teman yang dimiliki oleh user yang memiliki u_id yang sama dengan u_id.</p> <p>+insertFriendship(friend_id) : void Operasi ini digunakan untuk menambahkan hubungan pertemanan antara pengguna dengan pengguna yang memiliki u_id yang sama dengan friend_id. Pada operasi ini juga dilakukan penambahan notifikasi kepada</p>	

pengguna yang memiliki u_id yang sama dengan friend_id bahwa terdapat tawaran pertemanan yang diberikan kepadanya

+ acceptFriendship (notif_id) : void

Operasi ini digunakan untuk menerima tawaran pertemanan yang permintaannya terdapat pada data di notification dengan n_id yang sama dengan notif_id. Pada operasi ini juga dilakukan penambahan notifikasi kepada pengirim tawaran bahwa tawaran pertemanan telah diterima dan notifikasi tawaran yang telah dijawab akan dihapus.

+ rejectFriendship (notif_id) : void

Operasi ini digunakan untuk menolak tawaran pertemanan yang permintaannya terdapat pada data di notification dengan n_id yang sama dengan notif_id. Pada operasi ini juga dilakukan penambahan notifikasi kepada pengirim tawaran bahwa tawaran pertemanan telah ditolak dan notifikasi tawaran yang telah dijawab akan dihapus.

+ deleteFriendship (friend_id) : void

Operasi ini digunakan untuk menghapus pertemanan dengan user yang memiliki u_id sama dengan friend_id.

2.2.3.39 Spesifikasi Design Kelas Product

Product	<<control>>
<p>+__construct()</p> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+index(u_id) : void</p> <p>Operasi ini digunakan untuk menampilkan semua product yang dimiliki oleh pengguna yang memiliki u_id yang</p>	

sama dengan u_id.

+ insertProduct() : void

Operasi ini digunakan untuk mengambil data yang diinputkan user yaitu name, price, nego, second, stock, descript, status, direct, expire, lounge, dan pg_id kemudian memasukkannya ke database.

+ editProduct(p_id,from) : void

Operasi ini digunakan untuk mengubah data product yaitu name, price, nego, second, stock, descript, status, direct, expire, lounge, dan pg_id berdasarkan p_id kemudian mengarahkan halaman sesuai dengan from.

+ detailProduct(p_id) : void

Operasi ini digunakan untuk menampilkan detil dari product berdasarkan p_id. Operasi ini juga digunakan untuk menambahkan komentar yang diinputkan oleh user.

+ deleteProduct(p_id) : void

Operasi ini digunakan untuk menghapus product berdasarkan p_id.

+ insertPicture(p_id) : void

Operasi ini digunakan untuk menampilkan halaman untuk menambahkan gambar ke product yang memiliki p_id yang sama dengan p_id .

+ deletePicture(dp_id,p_id) : void

Operasi ini digunakan untuk menghapus gambar pada product berdasarkan dp_id dan p_id.

+ changeProductPict(p_id,pict) : void

Operasi ini digunakan untuk mengganti gambar utama pada product yang memiliki p_id yang sama dengan p_id menjadi gambar dengan nama pict.

+ uploadPicture(p_id) : void

Operasi ini digunakan untuk mengupload gambar ke server

dan mengubah gambar utama pada product yang memiliki p_id yang sama dengan p_id.

+ gotoShowroom() : void

Operasi ini digunakan untuk semua product yang dimiliki oleh pengguna yang telah berteman dengan user.

+ gotoLounge() : void

Operasi ini digunakan untuk menampilkan semua product yang memiliki lounge yang sama dengan 'y'.

+ deleteComment(p_id,c_id) : void

Operasi ini digunakan untuk menghapus komentar yang memiliki c_id yang sama dengan c_id kemudian mengarahkan halaman kembali pada halaman detail product dengan p_id yang sama dengan p_id.

+ insertPromote(p_id) : void

Operasi ini digunakan untuk menampilkan semua group barang yang ada di database berdasarkan c_id.

2.2.3.40 Spesifikasi Design Kelas Place

Place	<<control>>
<p>+ __construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+ index(u_id,lat,long) : void Operasi ini digunakan untuk menampilkan lokasi yang dimiliki oleh pengguna yang memiliki u_id sama dengan u_id. Apabila lat dan long tidak diisi dengan default, maka lokasi yang akan ditampilkan akan mengikuti nilai lat dan long.</p> <p>+ insertPlace() : void</p>	

Operasi ini digunakan untuk mengambil data yang diinputkan oleh user yaitu latitude, longitude, descript dan title kemudian memasukkannya ke database.

+ deletePlace(pl_id) : void

Operasi ini digunakan untuk menghapus lokasi yang dimiliki oleh user berdasarkan pl_id.

2.2.3.41 Spesifikasi Design Kelas Message

Message	<<control>>
<p>+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+index() : void Operasi ini digunakan untuk menampilkan halaman contact us dan mengambil inputan user yaitu subject dan text kemudian memasukkannya ke database. Setelah itu ditambahkan notifikasi ke user yang memiliki is_admin sama dengan 'y' bahwa terdapat pesan yang merupakan contact us</p> <p>+insertMessage(target_id) : void Operasi ini digunakan untuk mengambil inputan user yaitu subject dan text kemudian memasukkannya ke database. Setelah itu ditambahkan notifikasi ke user yang memiliki u_id sama dengan target_id bahwa terdapat pesan dari pengguna</p> <p>+listMessage() : void Operasi ini digunakan untuk menampilkan semua pesan yang dimiliki oleh pengguna.</p> <p>+detailMessage(m_id) : void</p>	

Operasi ini digunakan untuk menampilkan detail dari pesan yang memiliki m_id sama dengan m_id. Operasi ini juga dapat digunakan untuk menambahkan isi pesan dengan text yang diinputkan oleh user kemudian menambahkan notifikasi kepada tujuan pesan bahwa terdapat pesan baru yang dikirimkan.

2.2.3.42 Spesifikasi Design Kelas Appointment

Appointment	<<control>>
<p>+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+index() : void Operasi ini digunakan untuk menampilkan semua appointment yang dimiliki oleh pengguna.</p> <p>+insertAppointment(p_id) : void Operasi ini digunakan untuk mengambil data yang diinputkan user yaitu price, qty, dan direct dan memasukkannya ke database. Setelah itu ditambahkan notifikasi kepada pemilik product yang memiliki p_id sama dengan p_id bahwa terdapat appointment yang dikirimkan.</p> <p>+insertTimeOrPlace() : void Operasi ini digunakan untuk menambahkan informasi berupa waktu dan tempat pada appointment. Setelah itu ditambahkan notifikasi kepada pemilik product yang memiliki p_id sama dengan p_id bahwa terdapat appointment yang dikirimkan.</p> <p>+editAppointment(ap_id) : void Operasi ini digunakan untuk mengubah data appointment</p>	

yang memiliki `ap_id` seperti `ap_id` dengan inputan user yaitu `price`, `qty`, `types`, `date`, `time`, `lat`, `long`, dan `place`. Setelah itu ditambahkan notifikasi kepada pemilik product yang memiliki `p_id` sama dengan `p_id` bahwa appointment yang dikirimkan telah diubah.

+detailAppointment(`ap_id`) : void

Operasi ini digunakan untuk menampilkan detail dari appointment yang memiliki `ap_id` sama dengan `ap_id`.

+editUStatusAppointment(`ap_id`,`status`) : void

Operasi ini digunakan untuk mengubah status pengguna yang memiliki `u_id` sama dengan `u_id` pada appointment tersebut dengan status.

+editFStatusAppointment(`ap_id`,`status`) : void

Operasi ini digunakan untuk mengubah status pengguna yang memiliki `u_id` sama dengan `ap_friendid` pada appointment tersebut dengan status.

+cancelAppointment(`ap_id`) : void

Operasi ini digunakan untuk mengubah status `is_delete` pada appointment dengan `ap_id` yang sama dengan `ap_id`.

+doNotCancelAppointment(`ap_id`) : void

Operasi ini digunakan untuk mengubah kembali status `is_delete` pada appointment dengan `ap_id` yang sama dengan `ap_id` menjadi `'n'`.

+insertRating(`ap_id`) : void

Operasi ini digunakan untuk menambahkan rating lawan transaksi dari user yang terdapat pada appointment yang memiliki `ap_id` sama dengan `ap_id`.

2.2.3.43 Spesifikasi Design Kelas `Users_model`

<code>Users_model</code>	<code><<entity>></code>

Program Studi Teknik Informatika	DPPL – SPP	79/ 152
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		

+__construct()

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

+checkUser(un,pw) : bool

Operasi ini digunakan untuk mengecek apakah user dengan username yang sama dengan un memiliki password yang sama dengan pw atau tidak.

+checkEmailValid(email) : bool

Operasi ini digunakan untuk mengecek apakah sudah terdapat user yang memiliki email yang sama dengan email atau tidak.

+getUserByUsername(username) : row

Operasi ini digunakan untuk mengambil data user yang memiliki username yang sama dengan username dan is_deleted sama dengan 'n'.

+getAllUserByUsername(username) : row

Operasi ini digunakan untuk mengambil data semua user yang memiliki username sama dengan username.

+getUserAdmin() : result

Operasi ini digunakan untuk mengambil semua user yang memiliki is_admin sama dengan 'y'.

+insertUser(un,pw) : integer

Operasi ini digunakan untuk menambahkan user dengan username sama dengan un dan password sama dengan pw ke tabel users di database.

+deleteUser(u_id) : void

Operasi ini digunakan untuk menghapus user yang memiliki u_id sama dengan u_id.

+updateLastAct(u_id) : void

Operasi ini digunakan untuk mengubah u_last_act yang dimiliki oleh user yang memiliki u_id yang sama seperti

u_id dengan waktu saat ini.

+changePassword(u_id,newpassword) : void

Operasi ini digunakan untuk mengganti password user yang memiliki u_id sama seperti u_id dengan newpassword.

+changeUsername(u_id,newusername) : bool

Operasi ini digunakan untuk mengganti username user yang memiliki u_id sama seperti u_id dengan newusername.

+insertProfile(u_id,name,realname,birthdate,email,city,province) : void

Operasi ini digunakan untuk menambahkan data u_id, name, realname, birthdate, email, city, dan province ke tabel profile di database.

+editProfile(u_id,name,realname,birthdate,email,city,province,aboutme) : void

Operasi ini digunakan untuk mengubah data name, realname, birthdate, email, city, province dan aboutme pada tabel profile yang memiliki u_id sama dengan u_id.

+getProfileByUserid(u_id) : row

Operasi ini digunakan untuk mengambil data profile yang memiliki u_id sama dengan u_id.

+uploadProfPic(u_id,pp_name,filename) : void

Operasi ini digunakan untuk menambahkan data u_id, pp_name, dan filename ke tabel profile_picture di database.

+editProfPic(u_id,pp_name) : void

Operasi ini digunakan untuk mengganti pf_picture pada profile yang memiliki u_id yang sama seperti u_id dengan pp_name.

+getLastIdProfPic() : row

Operasi ini digunakan untuk mengambil data profile_picture yang memiliki pp_id terakhir.

+getProfPicByUserid(u_id) : result

Operasi ini digunakan untuk mengambil semua data di profile_picture yang memiliki u_id sama dengan u_id dan is_deleted sama dengan 'n'.

+getAllProfPicByUserid(u_id) : result

Operasi ini digunakan untuk mengambil semua data di profile_picture yang memiliki u_id sama dengan u_id.

+getUserNProfileByUserid(u_id) : row

Operasi ini digunakan untuk mengambil data di profile dan user yang memiliki u_id sama dengan u_id.

+getUserNProfileByUsernameLike(username) : result

Operasi ini digunakan untuk mengambil data di profile dan user yang memiliki username sama dengan username.

+getUserNProfileByRealnameLike(realname) : result

Operasi ini digunakan untuk mengambil data di profile dan user yang memiliki realname sama dengan realname.

+getUserNProfileByProfileNameLike(profilename) : result

Operasi ini digunakan untuk mengambil data di profile dan user yang memiliki name sama dengan profilename.

+getUserNProfile() : result

Operasi ini digunakan untuk mengambil semua data di user dan profile.

+insertSetting(u_id) : void

Operasi ini digunakan untuk menambahkan data berupa u_id ke tabel setting di database.

2.2.3.44 Spesifikasi Design Kelas Notification_model

Notification_model	<<entity>>
--------------------	------------

Program Studi Teknik Informatika	DPPL – SPP	82/ 152
----------------------------------	------------	---------

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

+__construct()

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

+insertNotif(u_id,target,target_id,doer,info) : void

Operasi ini digunakan untuk menambahkan data berupa u_id, target, target_id, doer, dan info ke tabel notification di database.

+updateNotifByUserid(u_id) : void

Operasi ini digunakan untuk mengubah is_read pada notifikasi yang memiliki u_id sama dengan u_id.

+deleteNotif(n_id) : void

Operasi ini digunakan untuk menghapus notifikasi yang memiliki n_id sama dengan n_id.

+getNotifByUseridDesc(u_id,limit) : result

Operasi ini digunakan untuk mengambil data notifikasi yang memiliki u_id sama dengan u_id secara descending sejumlah limit.

+CountNotifByUserid(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah notifikasi yang memiliki u_id sama dengan u_id.

+getNotifById(n_id) : row

Operasi ini digunakan untuk mengambil data notifikasi yang memiliki n_id sama dengan n_id.

+getUnreadNotifByUseridDesc(u_id) : result

Operasi ini digunakan untuk mengambil semua data notifikasi yang memiliki u_id sama dengan u_id dan memiliki is_read sama dengan 'n' secara descending.

+countUnreadNotifByUserid(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah notifikasi yang memiliki u_id sama dengan u_id dan

is_read sama dengan 'n'.

+getNotifByTargetIdDoerUserIdNInfoDesc(target, doer, user, info): row

Operasi ini digunakan untuk mengambil data notifikasi yang memiliki n_target yang sama dengan target, n_doer yang sama dengan doer, u_id sama dengan user, dan n_info sama dengan info.

2.2.3.45 Spesifikasi Design Kelas Testimony_model

Testimony_model	<<entity>>
+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+insertTestimony(to,from,text) : integer Operasi ini digunakan untuk menambahkan data berupa to, from, dan text ke tabel testimony di database	
+updateTestimony(t_id,text) : void Operasi ini digunakan untuk mengubah data text di testimony yang memiliki t_id sama dengan t_id.	
+deleteTestimony(t_id) : void Operasi ini digunakan untuk menghapus testimony yang memiliki t_id sama dengan t_id.	
+getTestimonyNProfileByUserid(u_id,limit) : result Operasi ini digunakan untuk mengambil data di tabel testimony dan profile yang memiliki u_id sama dengan u_id sejumlah limit.	
+countTestimonyNProfileByUserid(u_id) : integer Operasi ini digunakan untuk menghitung jumlah testimony dan profile yang memiliki u_id sama dengan u_id.	

+ getTestimonyById(t_id) : row

Operasi ini digunakan untuk mengambil data testimony yang memiliki t_id sama dengan t_id.

2.2.3.46 Spesifikasi Design Kelas Category_model

Category_model	<<entity>>
+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getAllCategoryByName(category) : row Operasi ini digunakan untuk mengambil semua data dari tabel productgroup yang memiliki pg_name sama dengan category	
+getCategory() : result Operasi ini digunakan untuk mengambil semua data category di tabel product_group pada database.	
+getCategoryArray() : array Operasi ini digunakan untuk mengambil semua data category di tabel product_group dan dikembalikan dalam bentuk array dengan menambahkan data bernama 'other'.	
+GetCategoryArrayAll() : array Operasi ini digunakan untuk mengambil semua data category di tabel product_group dan dikembalikan dalam bentuk array dengan menambahkan data bernama 'other' dan 'all category'.	
+insertCategory(category) : void Operasi ini digunakan untuk menambahkan data berupa category ke tabel product_group di database.	
+editCategory(category,newcategory) : void	

Operasi ini digunakan untuk mengubah data di tabel product group yang memiliki pg_name sama dengan category menjadi newcategory.

+ deleteCategory(category) : void

Operasi ini digunakan untuk menghapus category yang memiliki pg_name yang sama dengan category.

+ restoreCategory(category) : void

Operasi ini digunakan untuk mengubah is_deleted pada product_group yang memiliki pg_name sama dengan category menjadi 'n'.

2.2.3.47 Spesifikasi Design Kelas Friendship_model

Friendship_model	<<entity>>
<p>+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+insertFriendship(u_id,friend_id) :void Default konstruktor, digunakan untuk menambahkan data berupa u_id dan friend_id ke dalam tabel friendship di database.</p> <p>+reinsertFriendship(u_id,friend_id) : void Operasi ini digunakan untuk mengubah f_accepted menjadi null pada friendship yang memiliki u_id sama dengan u_id dan f_friend sama dengan friend_id.</p> <p>+acceptFriendship(u_id,friend_id,friendship_id) : void Operasi ini digunakan untuk menambahkan data berupa u_id dan friend_id ke tabel friendship di database dan mengubah f_accepted menjadi 'y' dan f_acc_date menjadi</p>	

waktu saat ini pada friendship yang memiliki f_id sama dengan friendship_id.

+reacceptFriendship(u_id,friend_id,friendship_id) : void

Operasi ini digunakan untuk mengubah f_accepted menjadi 'y', f_status menjadi 'acceptor', f_is_deleted menjadi 'n', f_crt_date dan f_acc date menjadi tanggal hari ini pada friendship yang memiliki u_id sama dengan u_id dan f_friend sama dengan friend_id serta mengubah f_accepted menjadi 'y' dan f_acc_date menjadi waktu saat ini pada friendship yang memiliki f_id sama dengan friendship_id.

+rejectFriendship(friendship_id) : void

Operasi ini digunakan untuk mengubah f_accepted menjadi 'n' pada friendship yang memiliki f_id sama dengan friendship_id.

+deleteFriendship(u_id,friend_id) : void

Operasi ini digunakan untuk manghapus hubungan friendship antara user yang memiliki u_id sama dengan u_id dan user yang memiliki u_id sama dengan friend_id.

+countFriendByUserid(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah teman yang memiliki u_id sama dengan u_id.

+getFriendship(u_id,friend_id) : row

Operasi ini digunakan untuk mengambil data friendship yang memiliki u_id sama dengan u_id dan f_friend sama dengan friend_id.

+getFriendshipById(friendship_id) : row

Operasi ini digunakan untuk menampilkan semua friendship yang memiliki f_id sama dengan friendship_id.

+ getFriendByUserid(u_id) : result

Operasi ini digunakan untuk menampilkan semua friendship yang memiliki u_id sama dengan u_id.

+ searchFriendByNameLike(profilename) : result

Operasi ini digunakan untuk menampilkan semua friendship yang user dengan u_id sama dengan f_friend memiliki pf_name sama dengan profilename.

+ getOtherDataByNameLike(profilename) : result

Operasi ini digunakan untuk menampilkan semua user yang memiliki pf_name sama dengan profilename.

+ getUserFriendStatus(u_id,friend_id) : string

Operasi ini digunakan untuk menampilkan hubungan friendship antara user yang memiliki u_id sama dengan u_id dan user yang memiliki u_id sama dengan friend_id.

2.2.3.48 Spesifikasi Design Kelas Product_model

Product_model	<<entity>>
+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getProductById(p_id) : row Operasi ini digunakan untuk mengambil data product yang memiliki p_id sama dengan p_id	
+ countProductByUserId(u_id) : integer Operasi ini digunakan untuk menghitung jumlah data product yang memiliki u_id sama dengan u_id.	
+ getProductNProfileByUserId(u_id) : result Operasi ini digunakan untuk mengambil data dari tabel product dan profile yang memiliki u_id sama dengan	

u_id.

+ getFriendProductNProfileByUserId(u_id) : result

Operasi ini digunakan untuk mengambil data dari tabel product dan profile yang dimiliki oleh user yang telah berteman dengan user yang memiliki u_id sama dengan u_id.

+ getFriendProductNProfileByUserIdAfter(u_id,firstid) : result

Operasi ini digunakan untuk mengambil data dari tabel product dan profile yang dimiliki oleh user yang telah berteman dengan user yang memiliki u_id sama dengan u_id dan p_id lebih kecil dari firstid.

+ getFriendProductNProfileByUserIdBefore(u_id,lasted, from) : result

Operasi ini digunakan untuk mengambil data dari tabel product dan profile yang dimiliki oleh user yang telah berteman dengan user yang memiliki u_id sama dengan u_id dan p_id lebih besar dari lasted.

+ getAllProductNProfileByUserId(u_id) : result

Operasi ini digunakan untuk mengambil data dari tabel product dan profile yang memiliki u_id sama seperti u_id.

+ getProductNProfileByNameLikeNUserId(u_id,name) : result

Operasi ini digunakan untuk mengambil data dari tabel product dan profile yang memiliki u_id sama seperti u_id dan p_name sama dengan name.

+ getOtherProductNProfileByUserIdNStatus(u_id,status) : result

Operasi ini digunakan untuk mengambil semua product dan profile yang dimiliki oleh user yang telah berteman

dengan user yang memiliki u_id sama dengan u_id, dan dengan p_status sama dengan status.

```
+getOtherProductNProfileByUserIdNStatusWhere(u_id, status, name, choose, minimal, maximal, second, direct, city, province) : result
```

Operasi ini digunakan untuk mengambil semua product dan profile yang dimiliki oleh user yang telah berteman dengan user yang memiliki u_id sama dengan u_id, dan sesuai dengan status, name, choose, minimal, maximal, second, direct, city dan province yang diinputkan.

```
+getProductNProfileForLounge(status) : result
```

Operasi ini digunakan untuk mengambil semua product yang memiliki p_lounge sama dengan 'y' dengan p_status sama dengan status.

```
+getProductNProfileForLoungeWhere(status, name, choose, minimal, maximal, second, direct, city, province) : result
```

Operasi ini digunakan untuk mengambil semua product yang memiliki p_lounge sama dengan 'y' dengan p_status sama dengan status dan sesuai dengan status, name, choose, minimal, maximal, second, direct, city dan province yang diinputkan.

```
+insertProduct(u_id, name, price, nego, stock, descript, status, direct, expire, lounge, pg_id, second) : integer
```

Operasi ini digunakan untuk menambahkan data yaitu u_id, name, price, nego, stock, descript, status, direct, expire, lounge, pg_id dan second ke tabel product di database.

```
+editProduct(p_id, name, price, nego, stock, descript, status, direct, expire, lounge, pg_id, second) : void
```

Operasi ini digunakan untuk mengubah data product yang memiliki p_id sama dengan p_id dengan name, price,

nego, stock, descript, status, direct, expire, lounge, pg_id, dan second.

+minusStock(p_id, minus) : void

Operasi ini digunakan untuk mengurangi p_stock yang dimiliki oleh product yang memiliki p_id sama dengan p_id sejumlah minus.

+deleteProduct(p_id) : void

Operasi ini digunakan untuk menghapus product yang memiliki p_id sama dengan p_id.

+getLastIdPict() : row

Operasi ini digunakan untuk mengambil data dari detail_pict yang memiliki id paling besar.

+uploadPict(p_id, dp_name, descript) : void

Operasi ini digunakan untuk menambahkan data berupa p_id, dp_name, dan descript ke tabel detail_pict di database.

+editProductPic(p_id, dp_name) : void

Operasi ini digunakan untuk mengubah p_pict pada product yang memiliki p_id sama seperti p_id dengan dp_name.

+deletePict(dp_id) : void

Operasi ini digunakan untuk menghapus data pada detail_pict yang memiliki dp_id sama dengan dp_id.

+getPictByProductId(p_id) : result

Operasi ini digunakan untuk mengambil semua data pada tabel detail_pict yang memiliki p_id sama dengan p_id.

+countPictByProductId(p_id) : integer

Operasi ini digunakan untuk menghitung jumlah data pada tabel detail_pict yang memiliki p_id sama dengan p_id.

+getPictById(dp_id) : row

Operasi ini digunakan untuk mengambil data detail_pict

yang memiliki dp_id sama dengan dp_id.

+getFirstPictByProductid(p_id) : row

Operasi ini digunakan untuk mengambil data detail_pict yang memiliki dp_id paling kecil, dp_is_deleted sama dengan 'n' dan p_id sama dengan p_id.

2.2.3.49 Spesifikasi Design Kelas Comment_model

Comment_model	<<entity>>
+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+insertComment(u_id,p_id,text) : integer Operasi ini digunakan untuk menambahkan data berupa u_id, p_id, dan text ke tabel comment di database.	
+deleteComment(c_id) : void Operasi ini digunakan untuk menghapus comment di database yang memiliki c_id sama dengan c_id.	
+getCommentNProfileByProductId(p_id) : result Operasi ini digunakan untuk mengambil comment dan profile yang memiliki p_id sama dengan p_id.	
+getCommentUserExceptWriter(writer,p_id) : result Operasi ini digunakan untuk mengambil semua data comment yang memiliki p_id sama dengan p_id dan c_writer tidak sama dengan writer.	
+getCommentById(c_id) : row Operasi ini digunakan untuk mengambil data comment yang memiliki c_id sama dengan c_id.	

2.2.3.50 Spesifikasi Design Kelas Promote_model

Promote_model	<<entity>>
----------------------	-------------------------------

Program Studi Teknik Informatika	DPPL – SPP	92/ 152
----------------------------------	------------	---------

+__construct()

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

+insertPromote(u_id,p_id,text) : integer

Operasi ini digunakan untuk menambahkan data berupa u_id, p_id, dan text pada tabel promote di database.

+getFriendPromoteByUserId(u_id) : result

Operasi ini digunakan untuk mengambil semua data promote yang memiliki u_id sama dengan f_friend dari user yang memiliki u_id sama dengan u_id.

2.2.3.51 Spesifikasi Design Kelas Place_model

Place_model	<<entity>>
--------------------	-------------------------------

+__construct()

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

+insertPlace(u_id,long,lat,title,descript) : void

Operasi ini digunakan untuk menambakan data berupa u_id, long, lat, title, dan descript ke tabel place di database.

+getPlaceByUserid(u_id) : result

Operasi ini digunakan untuk mengambil data place yang memiliki u_id sama dengan u_id.

+countPlaceByUserid(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah place yang memiliki u_id sama dengan u_id.

+getLastPlaceByUserid(u_id) : row

Operasi ini digunakan untuk mengambil data place yang memiliki u_id sama dengan u_id dan pl_id paling besar.

+getPlaceById(pl_id) : row

Operasi ini digunakan untuk mengambil data place yang memiliki pl_id sama dengan pl_id.

+deletePlace(pl_id) : void

Operasi ini digunakan untuk menghapus data place berdasarkan pl_id.

+distance(lat1,lng1,lat2,lng2,miles) : decimal

Operasi ini digunakan untuk menghitung jarak antara lat1 dan lng1 dengan lat2 dan lng2. Apabila miles bernilai true maka operasi ini akan mengembalikan hitungan dalam satuan miles, jika bernilai false maka operasi ini akan mengembalikan dalam satuan km

+checkUserNFriendDistance(u_id,f_id,distance) : bool

Operasi ini digunakan untuk mengecek apakah terdapat tempat yang dimiliki oleh user dengan u_id sama dengan u_id dan user dengan u_id sama dengan f_id yang berjarak kurang dari distance.

2.2.3.52 Spesifikasi Design Kelas Message_model

Message_model	<<entity>>
+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+insertMessage(subject,creator,target) : integer Operasi ini digunakan untuk menambahkan data berupa subject, creator, dan target ke tabel message di database.	
+updateMessageSRead(m_id,read) : void Operasi ini digunakan untuk mengubah m_sread menjadi	

read pada message yang memiliki m_id sama dengan m_id.

+ updateMessageTread(m_id,read) : void

Operasi ini digunakan untuk mengubah m_tread menjadi read pada message yang memiliki m_id sama dengan m_id.

+ insertMessageContent(m_id,writer,text) : void

Operasi ini digunakan untuk menambah data m_id, writer, dan text ke tabel message_content di database.

+ getMessageBySubjectUseridNTarget(subject,u_id,target) : row

Operasi ini digunakan untuk mengambil data message yang memiliki m_subject, u_id, dan m_target sama seperti subject, u_id dan target.

+ getMessageByUserNRead(u_id,status,read,limit) : result

Operasi ini digunakan untuk mengambil data message yang memiliki u_id dan m_read sama dengan u_id dan read sesuai dengan status yang dimiliki sebanyak limit.

+ countMessageByUserNRead(u_id,status,read) : integer

Operasi ini digunakan untuk menghitung jumlah message yang memiliki u_id sama dengan u_id, dan m_read sama dengan read sesuai dengan status yang dimiliki.

+ getMessageById(m_id) : row

Operasi ini digunakan untuk mengambil data message yang memiliki m_id sama dengan m_id.

+ getMessageByUserReadNSubjectLike(u_id,subject,status,read) : result

Operasi ini digunakan untuk mengambil data message yang memiliki u_id, m_subject dan m_read sama dengan u_id, subject dan read sesuai dengan status yang dimiliki.

+ getContentNProfileByMessageId(m_id) : result

Operasi ini digunakan untuk mengambil message_content dan profile yang memiliki m_id.

+ checkAdminReply(u_id,m_id) : bool

Operasi ini digunakan untuk mengecek apakah admin telah membalas message yang memiliki m_id sama dengan m_id yang dikirim oleh user dengan u_id sama dengan u_id.

+ countUnreadMessage(u_id,is_admin) : integer

Operasi ini digunakan untuk menghitung jumlah pesan yang belum di baca oleh user dengan u_id sama dengan u_id sesuai dengan is_admin yang dimiliki.

2.2.3.53 Spesifikasi Design Kelas Appointment_model

Appointment_model	<<entity>>
+__construct() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+insertAppointment(u_id,p_id,friend,price,qty,direct, date,place,long,lat) : integer Operasi ini digunakan untuk menambahkan data berupa u_id, p_id, friend, price, qty, direct, date, place, long dan lat ke tabel appointment di database.	
+editAppointment(ap_id,u_id,p_id,friend,price,qty, direct,date,place,long,lat) : void Operasi ini digunakan untuk mengubah data appointment menjadi u_id, p_id, friend, price, qty, direct, date, place, long dan lat di tabel appointment berdasarkan ap_id.	
+ editUStatusAppointment(ap_id,status) : void Operasi ini digunakan untuk mengubah ap_ustatus di appointment yang memiliki ap_id sama dengan ap_id menjadi status.	

+ editFStatusAppointment(ap_id,status) : void

Operasi ini digunakan untuk mengubah ap_fstatus di appointment yang memiliki ap_id sama dengan ap_id menjadi status.

+ editUReadAppointment(ap_id,read) : void

Operasi ini digunakan untuk mengubah ap_uread di appointment yang memiliki ap_id sama dengan ap_id menjadi read.

+ editFReadAppointment(ap_id,read) : void

Operasi ini digunakan untuk mengubah ap_fread di appointment yang memiliki ap_id sama dengan ap_id menjadi read.

+ getAppointmentByUserId(u_id) : result

Operasi ini digunakan untuk mengambil data appointment yang memiliki u_id sama seperti u_id.

+ getAppointmentByUserIdNRead(u_id,read,limit) : result

Operasi ini digunakan untuk mengambil data appointment yang memiliki u_id dan ap_read sama seperti u_id dan read sejumlah limit.

+countAppointmentByUserIdNRead(u_id,read) : integer

Operasi ini digunakan untuk menghitung jumlah appointment yang memiliki u_id dan ap_read sama seperti u_id dan read.

**+getAppointmentByUserIdReadNProductNameLike(u_id,
p_name,read) : result**

Operasi ini digunakan untuk mengambil data appointment yang memiliki u_id dan ap_read sama dengan u_id dan read, serta memiliki p_id dengan p_name sama .

+ getAppointmentById(ap_id) : row

Operasi ini digunakan untuk mengambil data appointment berdasarkan ap_id.

+ editIsDeletedAppointment(ap_id,deleted) : void

Operasi ini digunakan untuk mengubah ap_is_deleted pada tabel appointment menjadi sama dengan deleted berdasarkan ap_id.

+countAppointmentUnreadByUserId(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah data appointment yang memiliki ap_read sama dengan 'n' dan u_id sama dengan u_id atau ap_friend sama dengan u_id.

+countAllAppointmentByUserId(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah data appointment yang memiliki u_id sama dengan u_id atau ap_friend sama dengan u_id.

+countAllAppointmentPartnerByUserId(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah data appointment yang memiliki u_id sama dengan u_id atau ap_friend sama dengan u_id dengan u_id dari lawan transaksi yang unik.

+countSuccessfullAppointmentByUserId(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah data appointment yang memiliki ap_ustatus sama dengan 'accepted', ap_fstatus sama dengan 'accepted' dan u_id atau ap_friend sama dengan u_id.

+countSuccessfullAppointmentPartnerByUserId(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah data appointment yang memiliki ap_ustatus sama dengan 'accepted', ap_fstatus sama dengan 'accepted' dan u_id atau ap_friend sama dengan u_id dengan u_id dari lawan transaksi yang unik.

+countCanceledAppointmentByUserId(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah data

appointment yang memiliki `ap_is_deleted` sama dengan 'y' dan `u_id` atau `ap_friend` sama dengan `u_id`.

+countCanceledAppointmentPartnerByUserId(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah data appointment yang memiliki `ap_is_deleted` sama dengan 'y' dan `u_id` atau `ap_friend` sama dengan `u_id` dengan `u_id` dari lawan transaksi yang unik.

+countOnGoingAppointmentByUserId(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah data appointment yang memiliki `ap_is_deleted` tidak sama dengan 'y', `ap_ustatus` tidak sama dengan 'accepted', `ap_fstatus` tidak sama dengan 'accepted' dan id atau `ap_friend` sama dengan `u_id`.

+countOnGoingAppointmentPartnerByUserId(u_id) : integer

Operasi ini digunakan untuk menghitung jumlah data appointment yang memiliki `ap_is_deleted` tidak sama dengan 'y', `ap_ustatus` tidak sama dengan 'accepted', `ap_fstatus` tidak sama dengan 'accepted' dan id atau `ap_friend` sama dengan `u_id` dengan `u_id` dari lawan transaksi yang unik.

+insertRating(ap_id) : void

Operasi ini digunakan untuk menambahkan data berupa `ap_id`, `r_target`, dan `r_point` ke tabel rate di database.

+countAverateRate(u_id) : float

Operasi ini digunakan untuk menghitung rata-rata `r_point` yang dimiliki oleh user dengan `u_id` sama dengan `u_id`.

+ getAppointmentForReminder(u_id) : result

Operasi ini digunakan untuk mengambil data appointment yang memiliki `u_id` atau `ap_friend` sama dengan `u_id`,

ap_date tidak sama dengan null, ap_date lebih dari atau sama dengan hari ini, ap_fstatus tidak sama dengan null, ap_fstatus tidak sama dengan 'rejected', dan ap_ustatus serta ap_fstatus tidak sama dengan accepted yang diurutkan berdasarkan ap_date secara ascending.



3 Deskripsi Dekomposisi

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas Tabel Users

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
U_ID	int(11)	ID Users, auto increment, primary key
U_USERNAME	varchar(25)	Username
U_PASSWORD	varchar(25)	Password
U_IS_ADMIN	varchar(1)	Penanda User merupakan admin atau bukan
U_IS_DELETED	varchar(1)	Penanda User telah di delete atau belum
U_CRT_DATE	datetime	Tanggal pendaftaran user
U_LAST_ACT	datetime	Tanggal terakhir kali user melakukan login

3.1.2 Deskripsi Entitas Tabel Profile

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
PF_ID	int(11)	ID Profile, auto increment, primary key
U_ID	int(11)	ID dari user yang memiliki profile, foreign key
PF_NAME	varchar(35)	Nama yang ditampilkan pada situs
PF_REAL_NAME	varchar(35)	Nama sebenarnya dari user
PF_BIRTH_DATE	datetime	Tanggal lahir user
PF_EMAIL	varchar(35)	Email yang dimiliki user
PF_CITY	varchar(25)	Kota yang ditinggali user
PF_PROVINCE	varchar(25)	Provinsi dari kota yang ditinggali user
PF_ABOUT_ME	text	Tentang user
PF_PICTURE	varchar(15)	Nama gambar yang digunakan sebagai foto profil

3.1.3 Deskripsi Entitas Tabel Profile_Picture

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
PP_ID	int(11)	ID profile_picture, auto increment, primary key
U_ID	int(11)	ID dari user yang memiliki profile_picture, foreign key
PP_NAME	varchar(25)	Nama profile picture
PP_TITLE	varchar(25)	Judul dari profile picture yang akan ditampilkan pada situs

PP_IS_DELETED	varchar(1)	Penanda profile_picture telah di delete atau belum
PP_CRT_DATE	datetime	Tanggal upload profile picture

3.1.4 Deskripsi Entitas Tabel Product_Group

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
PG_ID	int(11)	ID product_group, auto increment, primary key
PG_NAME	varchar(25)	Nama dari product_group
PG_IS_DELETED	varchar(1)	Penanda product_group telah di delete atau belum
PG_CRT_DATE	datetime	Tanggal ditambahkan product_group tersebut

3.1.5 Deskripsi Entitas Tabel Product

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
P_ID	int(11)	ID product, auto increment, primary key
U_ID	int(11)	ID dari user yang memiliki product, foreign key
P_NAME	varchar(25)	Nama product
P_PRICE	decimal(15)	Harga product
P_STOCK	int	Jumlah stock product yang dimiliki user
P_DESCRIPT	text	Deskripsi produk
P_PICT	varchar(25)	Nama gambar yang digunakan product

P_STATUS	varchar(10)	Status product penanda bahwa product tersebut dicari atau dijual
P_EXP_DATE	datetime	Tanggal berakhirnya penawaran product
P_LOUNGE	varchar(1)	Penanda bahwa product akan ditampilkan pada lounge atau tidak
P_IS_DELETED	varchar(1)	Penanda product telah di delete atau belum
P_CRT_DATE	datetime	Tanggal ditambahkan product tersebut
P_MOD_DATE	datetime	Tanggal diubahnya data pada product tersebut
P_NEGO	varchar(1)	Penanda bahwa harga product diperbolehkan untuk nego atau tidak
PG_ID	int(11)	ID dari product_group yang dimiliki product, foreign key
P_SECOND	varchar(1)	Penanda bahwa product merupakan barang second atau bukan
P_DIRECT	varchar(1)	Penanda bahwa product hanya dijual secara langsung atau boleh melalui pengiriman atau bank

3.1.6 Deskripsi Entitas Tabel Detail_Pict

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
DP_ID	int(11)	ID detail_pict, auto increment, primary key
P_ID	int(11)	ID dari product yang memiliki detail_pict, foreign key
DP_NAME	varchar(25)	Nama dari gambar detail_picture

DP_TITLE	varchar(25)	Judul dari gambar detail_picture yang akan ditampilkan pada situs
DP_IS_DELETED	varchar(1)	Penanda bahwa product telah didelete atau belum
DP_CRT_DATE	datetime	Tanggal ditambahkan detail_picture tersebut

3.1.7 Deskripsi Entitas Tabel Place

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
PL_ID	int(11)	ID place, auto increment, primary key
U_ID	int(11)	ID dari user yang memiliki place, foreign key
PL_LONG	decimal(15, 10)	Longitude dari place tersebut
PL_LAT	decimal(15, 10)	Latitude dari place tersebut
PL_TITLE	varchar(25)	Judul place tersebut
PL_DESCRIPTOR	varchar(50)	Deskripsi dari place tersebut
PL_IS_DELETED	varchar(1)	Penanda bahwa place telah didelete atau belum
PL_CRT_DATE	datetime	Tanggal ditambahkan place tersebut

3.1.8 Deskripsi Entitas Tabel Friendship

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
F_ID	int(11)	ID friendship, auto increment, primary key

U_ID	int(11)	ID dari user yang menawarkan friendship, foreign key
F_FRIEND	int(11)	ID dari user yang ditawarkan friendship
F_ACCEPTED	varchar(1)	Penanda bahwa pertemanan yang ditawarkan telah diterima, tidak diterima atau belum diterima
F_STATUS	varchar(15)	Penanda bahwa user yang memiliki u_id merupakan pemberi tawaran atau penerima tawaran
F_IS_DELETED	varchar(1)	Penanda bahwa pertemanan telah dihapus atau belum
F_CRT_DATE	datetime	Tanggal ditambahkan friendship tersebut
F_ACC_DATE	datetime	Tanggal diterimanya friendship tersebut

3.1.9 Deskripsi Entitas Tabel Message

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
M_ID	int(11)	ID message, auto increment, primary key
U_ID	int(11)	ID dari user yang memiliki message, foreign key
M_TARGET	int(11)	ID dari user yang merupakan target dari pengiriman pesan. Apabila M_TARGET sama dengan 0, message tersebut artinya ditujukan kepada admin sebagai contact us
M_SUBJECT	varchar(25)	Subject dari message
M_SREAD	varchar(1)	Penanda bahwa sender atau user yang memiliki u_id telah membaca pesan

		atau belum
M_TREAD	varchar(1)	Penanda bahwa sender atau user yang merupakan m_target telah membaca pesan atau belum
M_IS_DELETED	varchar(1)	Penanda bahwa message telah dihapus atau belum
M_CRT_DATE	datetime	Tanggal ditambahkan message tersebut
M_MOD_DATE	datetime	Tanggal ditambahkan message_content dengan m_id sama dengan message tersebut

3.1.10 Deskripsi Entitas Tabel Message_Content

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
MC_ID	int(11)	ID message_content, auto increment, primary key
M_ID	int(11)	ID dari message yang dimiliki message_content, foreign key
MC_WRITER	int(11)	ID dari user yang menambahkan message_content tersebut
MC_TEXT	text	Text dari message_content
MC_IS_DELETED	varchar(1)	Penanda bahwa message_content telah dihapus atau belum
MC_CRT_DATE	datetime	Tanggal ditambahkan message_content tersebut

3.1.11 Deskripsi Entitas Tabel Comment

NAMA ELEMEN DATA	TIPE DATA	NARASI
C_ID	int(11)	ID comment, auto increment,

		primary key
P_ID	int(11)	ID dari product yang memiliki comment, foreign key
C_WRITER	int(11)	ID dari user yang menambahkan comment
C_TEXT	text	Text comment
C_IS_DELETED	varchar(1)	Penanda bahwa comment telah didelete atau belum
C_CRT_DATE	datetime	Tanggal ditambahkan comment tersebut

3.1.12 Deskripsi Entitas Tabel Promote

NAMA ELEMEN DATA	TIPE DATA	NARASI
PR_ID	int(11)	ID promote, auto increment, primary key
U_ID	int(11)	ID dari user yang membuat promote, foreign key
P_ID	int(11)	ID dari product yang di-promote, foreign key
PR_TEXT	varchar(150)	Text promote
PR_IS_DELETED	varchar(1)	Penanda bahwa promote telah didelete atau belum
PR_CRT_DATE	datetime	Tanggal ditambahkan promote tersebut

3.1.13 Deskripsi Entitas Tabel Appointment

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
------------------	-----------	-----------

AP_ID	int(11)	ID appointment, auto increment, primary key
P_ID	int(11)	ID dari product yang memiliki appointment, foreign key
U_ID	int(11)	ID dari user yang menawarkan appointment, foreign key
AP_FRIEND	int(11)	ID dari user yang ditawarkan appointment
AP_PRICE	int(11)	Harga barang yang ditawarkan oleh user
AP_QUANTITY	int(11)	Jumlah barang yang ditawarkan oleh user
AP_DIRECT	varchar(1)	Penanda bahwa transaksi akan dilakukan secara langsung atau tidak
AP_DATE	datetime	Tanggal dilakukan transaksi jual beli
AP_PLACE	varchar(50)	Tempat dilakukan transaksi jual beli
AP_LONG	decimal(15, 10)	Longitude dari tempat dilakukan transaksi jual beli
AP_LAT	decimal(15, 10)	Latitude dari tempat dilakukan transaksi jual beli
AP_USTATUS	varchar(15)	Status yang menandakan keadaan transaksi di pihak penawar transaksi
AP_FSTATUS	varchar(15)	Status yang menandakan keadaan transaksi di pihak penerima tawaran transaksi
AP_RATE	Varchar(1)	Penanda bahwa transaksi telah dirate atau belum
AP_IS_DELETED	varchar(1)	Penanda bahwa appointment telah didelete atau belum
AP_CRT_DATE	datetime	Tanggal ditawarkan appointment

		tersebut
AP_MOD_DATE	datetime	Tanggal terakhir dilakukan perubahan pada appointment tersebut

3.1.14 Deskripsi Entitas Tabel Testimony

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
T_ID	int(11)	ID testimony, auto increment, primary key
U_ID	int(11)	ID dari user yang memiliki testimony, foreign key
T_WRITER	int(11)	ID dari user yang menulis testimony
T_TEXT	text	Text testimony
T_IS_DELETED	varchar(1)	Penanda testimony telah di delete atau tidak
T_CRT_DATE	datetime	Tanggal ditambahkan testimony tersebut

3.1.15 Deskripsi Entitas Tabel Notification

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
N_ID	int(11)	ID notification, auto increment, primary key
U_ID	int(11)	ID dari user yang memiliki notification, foreign key
N_TARGET	varchar(15)	Tabel yang di-notifikasikan
N_TARGET_ID	int(11)	ID dari data tabel yang di-notifikasikan
N_DOER	int(11)	ID dari user yang menyebabkan atau

		menambahkan notifikasi
N_INFO	varchar(15)	Informasi tambahan dari notifikasi
N_READ	varchar(1)	Penanda notifikasi telah di baca atau belum
N_IS_DELETED	varchar(1)	Penanda notifikasi telah di delete atau belum
N_CRT_DATE	datetime	Tanggal dibuatnya notifikasi
N_TEXT	varchar(300)	Text yang akan ditampilkan pada notifikasi

3.1.16 Deskripsi Entitas Tabel Rate

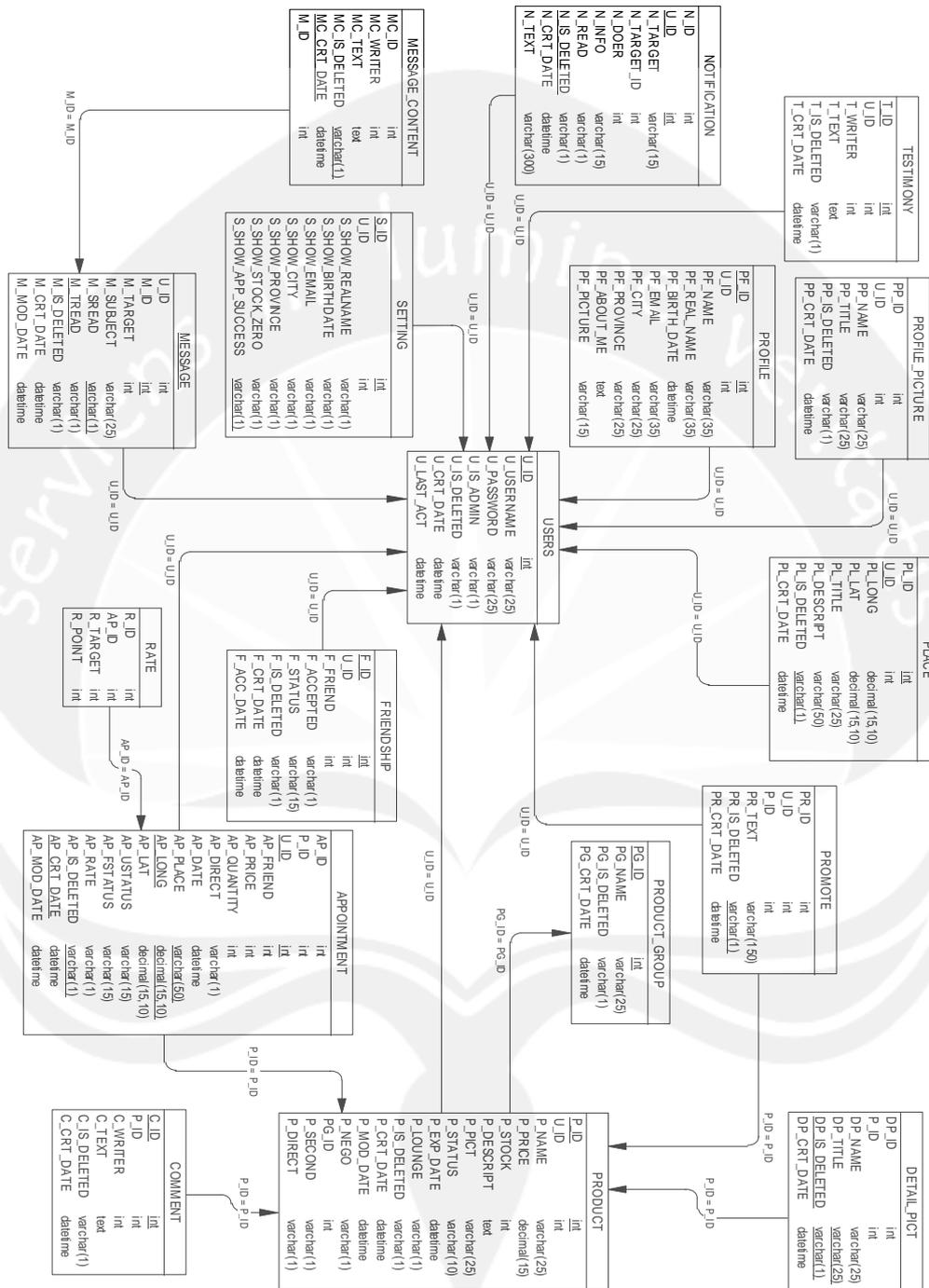
NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
R_ID	int(11)	ID notification, auto increment, primary key
AP_ID	int(11)	ID dari appointment yang di <i>rate</i> , foreign key
R_TARGET	int(11)	ID dari user yang menjadi target <i>rate</i>
R_POINT	int(11)	Poin yang diberikan untuk hasil transaksi dengan user yang menjadi target <i>rate</i>

3.1.17 Deskripsi Entitas Tabel Setting

NAMA ELEMEN DATA	TIPE DATA	DESKRIPSI
S_ID	int(11)	ID setting, auto increment, primary key
U_ID	int(11)	ID dari user yang memiliki setting, foreign key
S_SHOW_REALNAME	varchar(1)	Penanda realname akan ditampilkan pada profil user atau

		tidak
S_SHOW_BIRTHDATE	varchar(1)	Penanda birthdate akan ditampilkan pada profil user atau tidak
S_SHOW_EMAIL	varchar(1)	Penanda email akan ditampilkan pada profil user atau tidak
S_SHOW_CITY	varchar(1)	Penanda city akan ditampilkan pada profil user atau tidak
S_SHOW_PROVINCE	varchar(1)	Penanda province akan ditampilkan pada profil user atau tidak
S_SHOW_STOCK_ZERO	varchar(1)	Penanda stock barang yang berjumlah nol akan ditampilkan pada profil user atau tidak
S_SHOW_APP_SUCCESS	varchar(1)	Penanda appointment yang telah sukses akan ditampilkan pada list appointment atau tidak

3.2 Physical Data Model



Gambar 3.1 Physical Data Model

4 Deskripsi Perancangan Antarmuka

4.1 Login



Gambar 4.1 Rancangan Antarmuka Login

Antarmuka Gambar 4.1 merupakan antarmuka login yang digunakan untuk melakukan proses login ke dalam situs. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka login seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka login seperti pada gambar (a). Untuk mendapatkan akses masuk ke dalam situs, user harus menginputkan username dan password dengan benar pada textbox yang telah disediakan. Setelah tombol login ditekan, sistem akan mengecek username dan password yang diinputkan dengan data username dan password yang telah tersimpan di database. Apabila data username dan password benar atau cocok maka user akan masuk ke dalam sistem, sebaliknya jika username dan password salah atau tidak cocok maka akan diberikan pesan peringatan. Pada antarmuka ini, terdapat tombol Register yang

digunakan untuk menghubungkan user ke antarmuka Register.

4.2 Register



Gambar 4.2 Rancangan Antarmuka Register

Antarmuka Gambar 4.2 merupakan antarmuka Register yang digunakan untuk mendaftarkan diri ke situs SOS. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka register seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka register seperti pada gambar (a). Untuk melakukan pendaftaran, user harus memasukkan username yang unik, password dan konfirmasi password, real name atau nama asli user, profile name atau nama profil yang digunakan, city atau kota tempat tinggal user, province atau provinsi tempat tinggal user, email yang digunakan user, dan birth date atau tanggal lahir user kemudian menekan tombol Register. Setelah tombol register ditekan, maka sistem akan memeriksa data-data

Program Studi Teknik Informatika	DPPL – SPP	115/ 152
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		

yang dimasukkan user. Apabila semua data valid, maka user akan terdaftar di situs ini.

4.3 Main Menu

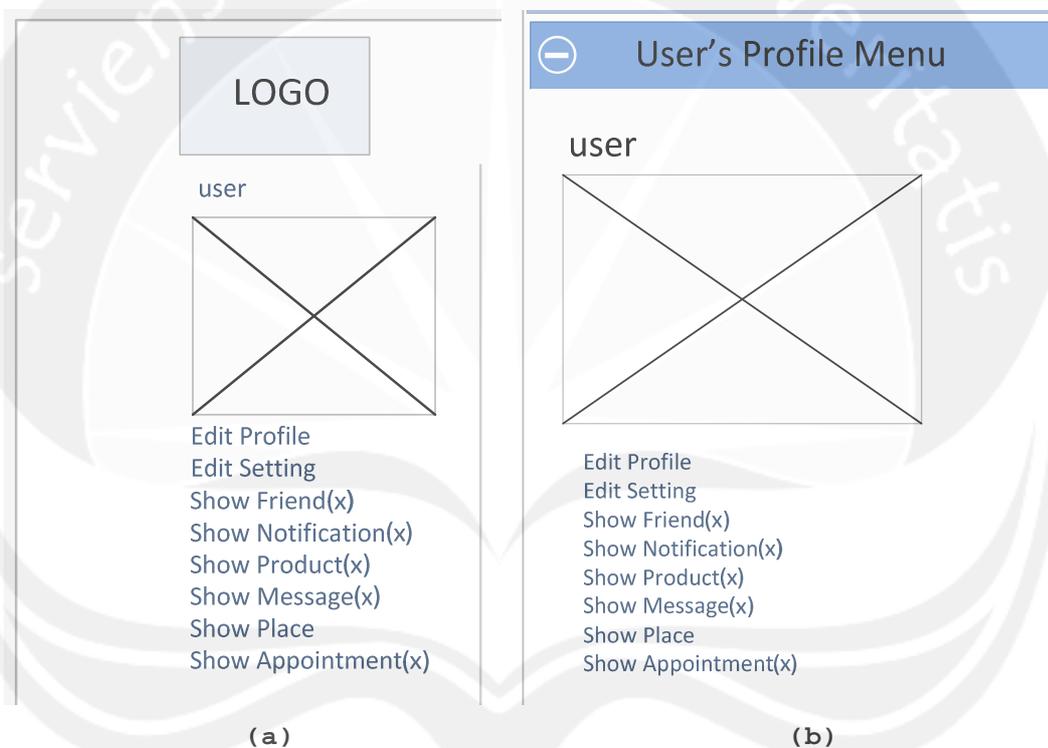


Gambar 4.3 Rancangan Antarmuka Menu Utama

Antarmuka Gambar 4.3 merupakan antarmuka Menu utama yang digunakan sebagai link untuk ke menu-menu utama. Apabila user yang merupakan admin melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka menu utama seperti pada gambar (c). Apabila user yang merupakan admin melakukan akses dengan perangkat yang bukan mobile maka akan tampil antarmuka menu utama seperti pada gambar (a). Apabila user yang bukan merupakan admin melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka menu utama seperti pada gambar (d). Apabila user yang bukan merupakan admin melakukan akses dengan perangkat yang bukan mobile maka akan tampil antarmuka menu utama seperti pada gambar (b). Link "home" akan mengarahkan user ke halaman utama, link profile akan mengarahkan user ke profile user, link contact us akan mengarahkan user ke halaman contact us,

dan link log out akan mengakhiri session yang telah dibangun user. Apabila user yang masuk merupakan admin, maka akan link contact us akan menghilang dan muncul dua menu yaitu manage user dan manage category. Link manage user akan mengarahkan user ke halaman manage user dan manage category akan mengarahkan user ke halaman manage category.

4.4 Profile Menu

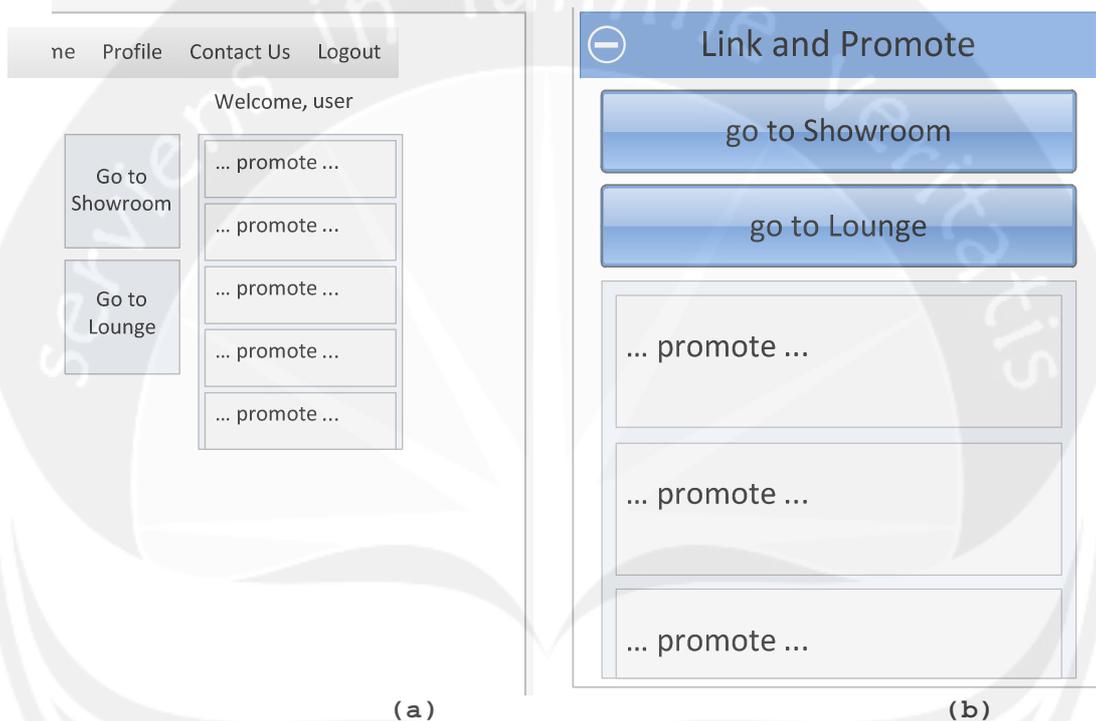


Gambar 4.4 Rancangan Antarmuka Profile Menu

Antarmuka Gambar 4.4 merupakan antarmuka menu profil yang digunakan sebagai link ke menu-menu yang dapat diakses user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka register seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka register seperti pada gambar (a). Apabila user memasuki halaman user lain

yang merupakan teman dari user, maka menu yang akan ditampilkan hanya show friend, show product dan show place. Apabila user memasuki halaman user lain yang bukan merupakan teman dari user, maka menu yang akan ditampilkan hanya show friend.

4.5 Link and Promote



Gambar 4.5 Rancangan Antarmuka Link dan Promote

Antarmuka Gambar 4.5 merupakan antarmuka untuk menampilkan link ke showroom atau lounge serta promosi yang dilakukan user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka link dan promosi seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka link dan promosi seperti pada gambar (a). Pada tampilan bukan mobile, link ke showroom atau lounge tidak selalu ada

di setiap halaman namun, link ke showroom atau lounge selalu ada pada tampilan mobile.

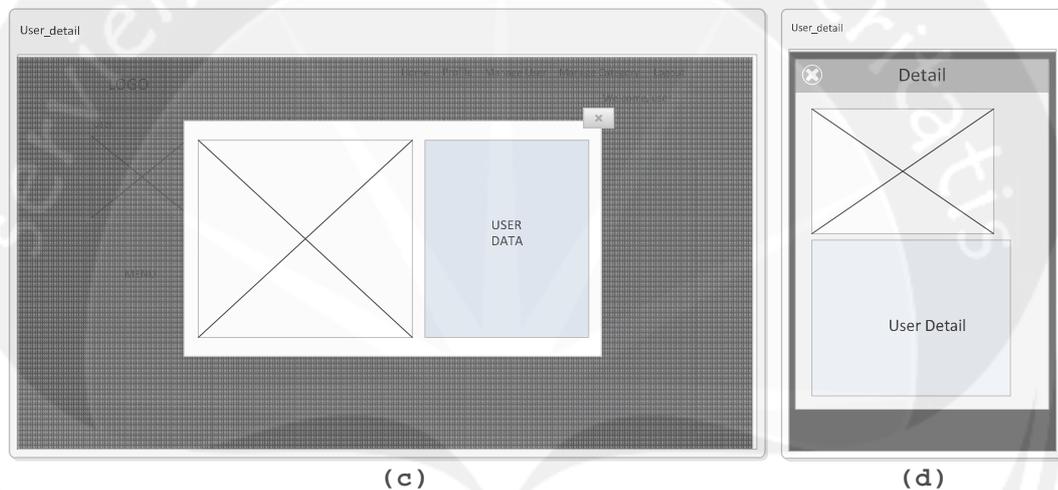
4.6 Manage Users



Gambar 4.6 Rancangan Antarmuka Manage User

Antarmuka Gambar 4.6 digunakan untuk melakukan pengelolaan user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka manage user seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka link dan promosi seperti pada gambar (a). Antarmuka ini hanya dapat dimasuki oleh user yang merupakan administrator. Pada antarmuka ini, user dapat melakukan pencarian user berdasarkan username, pencarian user berdasarkan real name, pencarian user berdasarkan profile name, menampilkan kembali semua user, menampilkan detail user yang dipilih, atau mendelete user yang telah dipilih. Untuk melakukan pencarian user, dapat dilakukan dengan cara memilih pencarian akan dilakukan berdasarkan apa dari

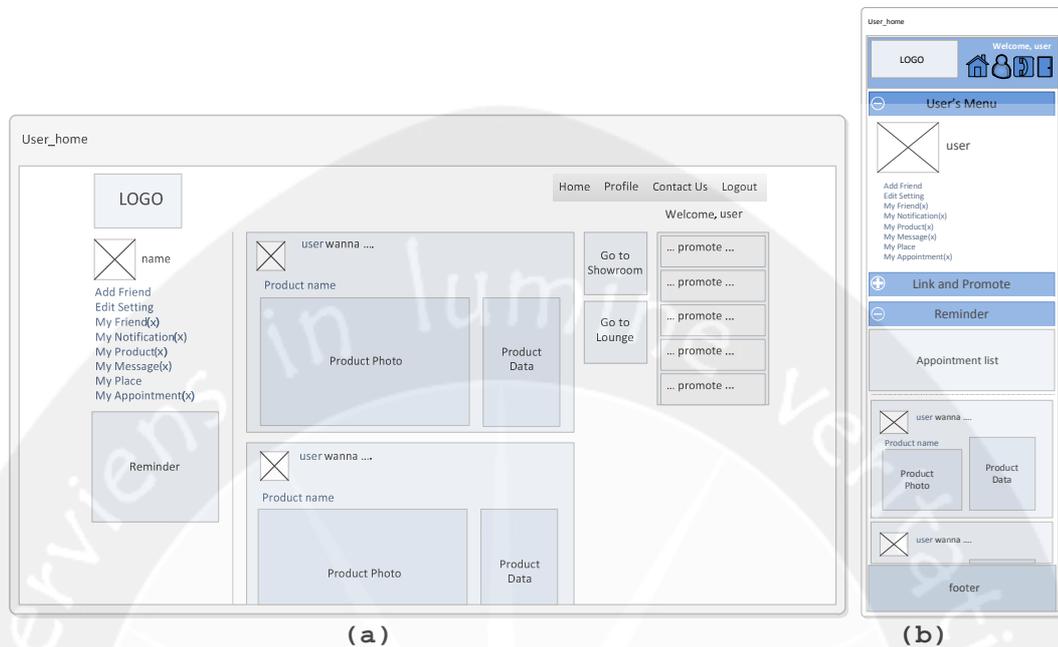
radio button, kemudian mengetikkan nama user yang dicari pada textbox dan menekan tombol Search User. Untuk menampilkan kembali semua user dapat dilakukan dengan cara menekan tombol Show All User. Untuk menampilkan detail user dapat dilakukan dengan cara menekan link detail pada user table. Setelah menekan link detail maka akan muncul pop up seperti pada Gambar 4.7 yang menampilkan foto profil dan data dari user yang di klik.



Gambar 4.7 Rancangan Antarmuka Detail User

Untuk mendelete user dapat dilakukan dengan cara menekan link delete pada user table.

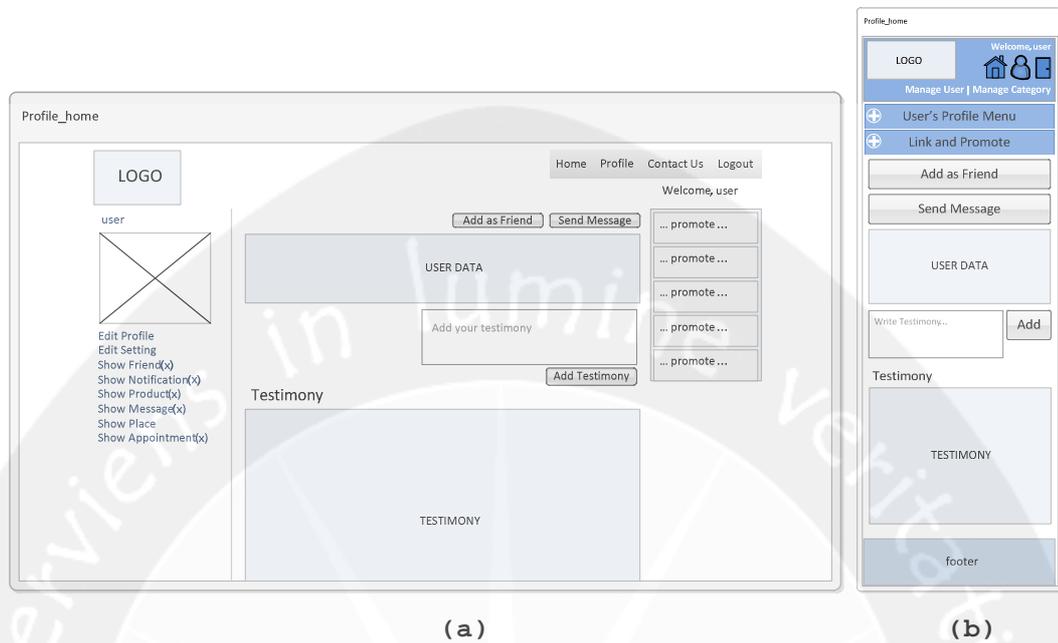
4.7 User home



Gambar 4.8 Rancangan Antarmuka User Home

Antarmuka Gambar 4.8 merupakan antarmuka awal yang ditampilkan setelah user melakukan login atau apabila user menekan link "home". Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka user home seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka user home seperti pada gambar (a). Pada antarmuka ini ditampilkan semua product yang dimiliki oleh semua user yang telah berteman dengan user berdasarkan waktu ditambahkan product tersebut.

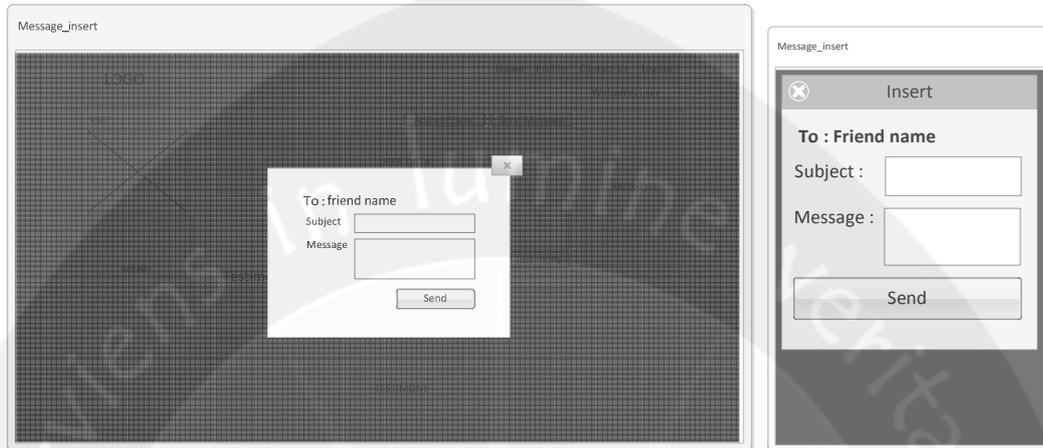
4.8 Profile



Gambar 4.9 Rancangan Antarmuka Profile

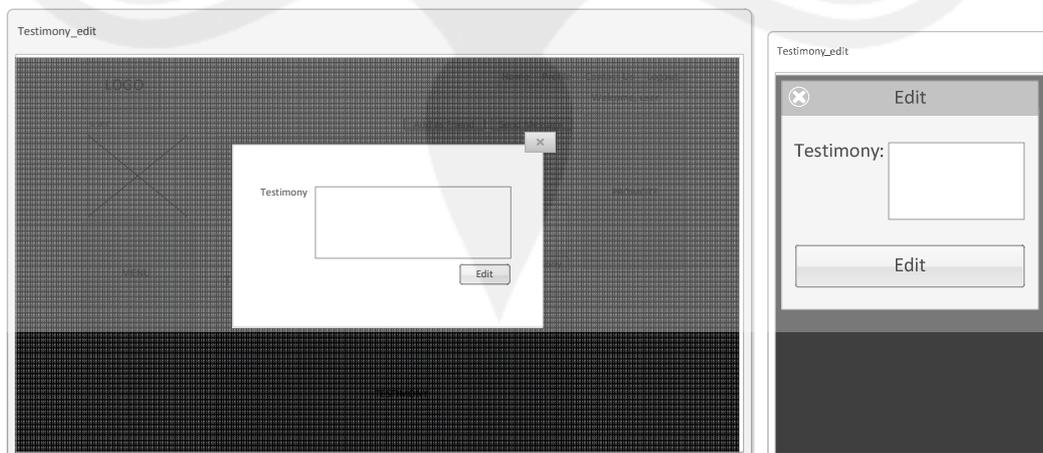
Antarmuka Gambar 4.9 merupakan antarmuka yang menampilkan profil atau data-data pengguna. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka profile seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka profile seperti pada gambar (a). Pada antarmuka ini terdapat tombol Add as Friend yang akan muncul apa bila user membuka profil dari user lain yang belum berteman dengan user, Send Message yang akan muncul apabila user membuka profil dari user lain yang sudah berteman dengan user, textbox testimony dan tombol testimony yang digunakan untuk menambahkan testimony ke halaman profil user, dan testimony-testimony yang telah ada. Apabila user menekan tombol add as friend, maka permintaan pertemanan akan dikirimkan kepada pemilik profil tersebut. Apabila user menekan tombol send

message, maka akan muncul pop up seperti gambar 4.10 yang digunakan untuk menulis dan mengirimkan pesan kepada pemilik profil tersebut.



Gambar 4.10 Rancangan Antarmuka Insert Message

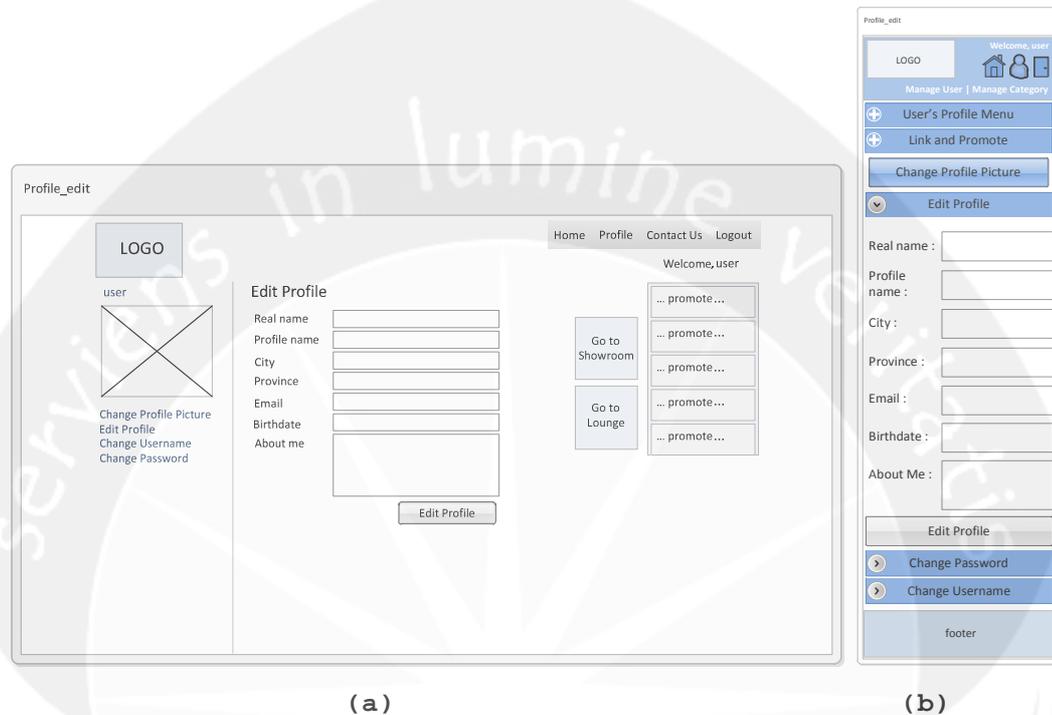
Apabila user menambahkan testimony dengan mengisi textbox dan menekan tombol Add testimony maka testimony yang ditambahkan user akan dimunculkan di bagian testimony. Testimony-testimony yang ditambahkan dapat diedit dan di delete oleh user yang menambahkannya dengan cara menekan gambar edit sehingga muncul pop up seperti pada gambar 4.11. Kemudian user dapat mengedit testimony yang terdapat pada textbox dan menekan tombol edit sehingga testimony tersebut terupdate.



Gambar 4.11 Rancangan Antarmuka Edit Testimony

Untuk menghapus testimony, user dapat menekan gambar delete yang terdapat di samping testimony.

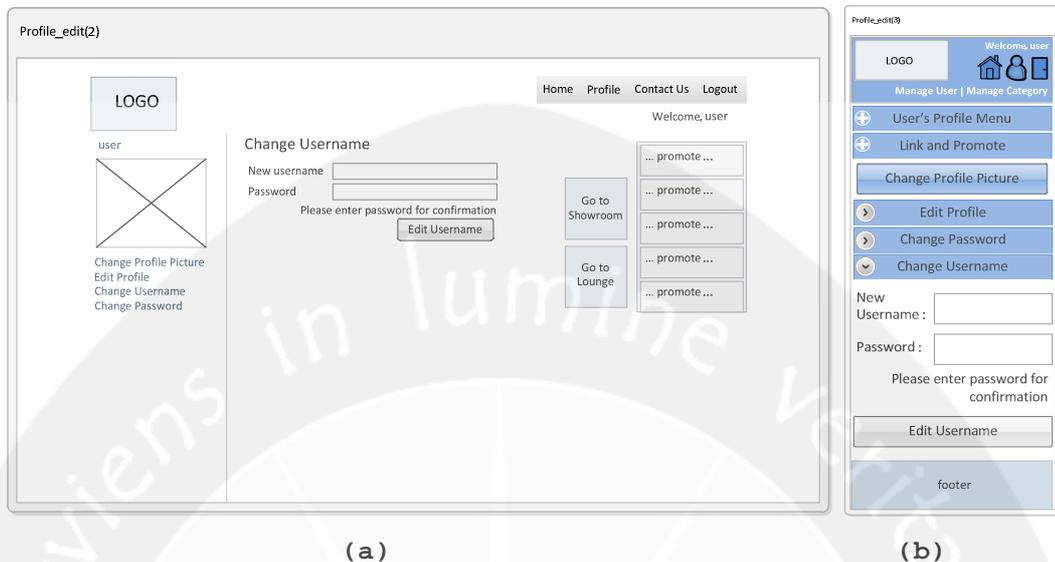
4.9 Edit Profile



Gambar 4.12 Rancangan Antarmuka Edit Profil

Antarmuka Gambar 4.12 merupakan antarmuka yang digunakan untuk mengedit profil user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka edit profil seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka edit profil seperti pada gambar (a). Untuk melakukan edit profil, user mengubah data profil yang telah di tampilkan kemudian menekan tombol edit profile. Apabila data-data yang dimasukkan valid, maka profil yang dimiliki oleh user tersebut akan teredit.

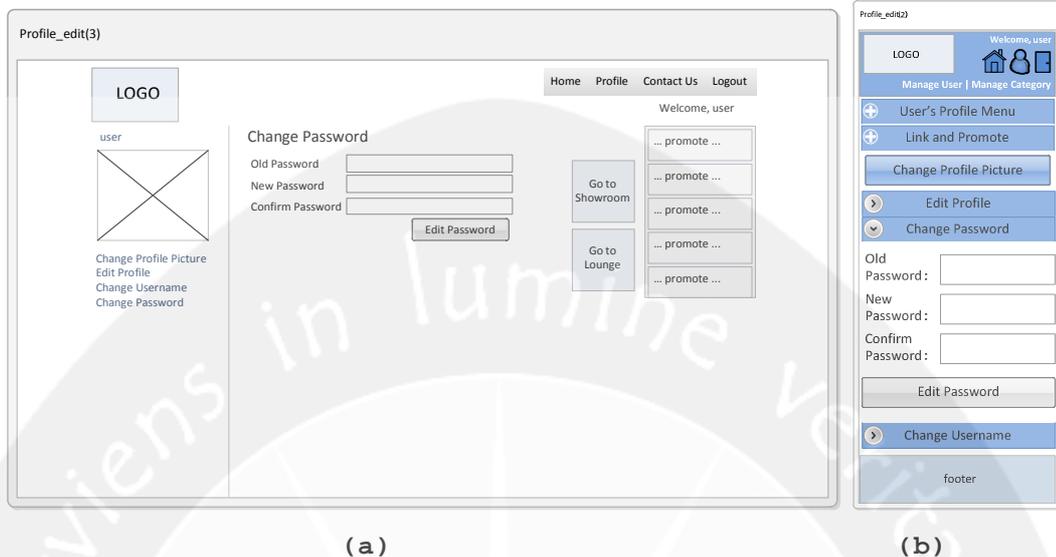
4.10 Change Username



Gambar 4.13 Rancangan Antarmuka Change Username

Antarmuka Gambar 4.13 merupakan antarmuka yang digunakan untuk mengubah username yang dimiliki user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka change username seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka change username seperti pada gambar (a). Untuk mengubah username, user dapat mengisi username yang baru serta password yang dimiliki saat ini untuk mengkonfirmasi bahwa yang mengubah username adalah user yang bersangkutan. Kemudian menekan tombol edit username dan username akan diubah sesuai dengan inputan.

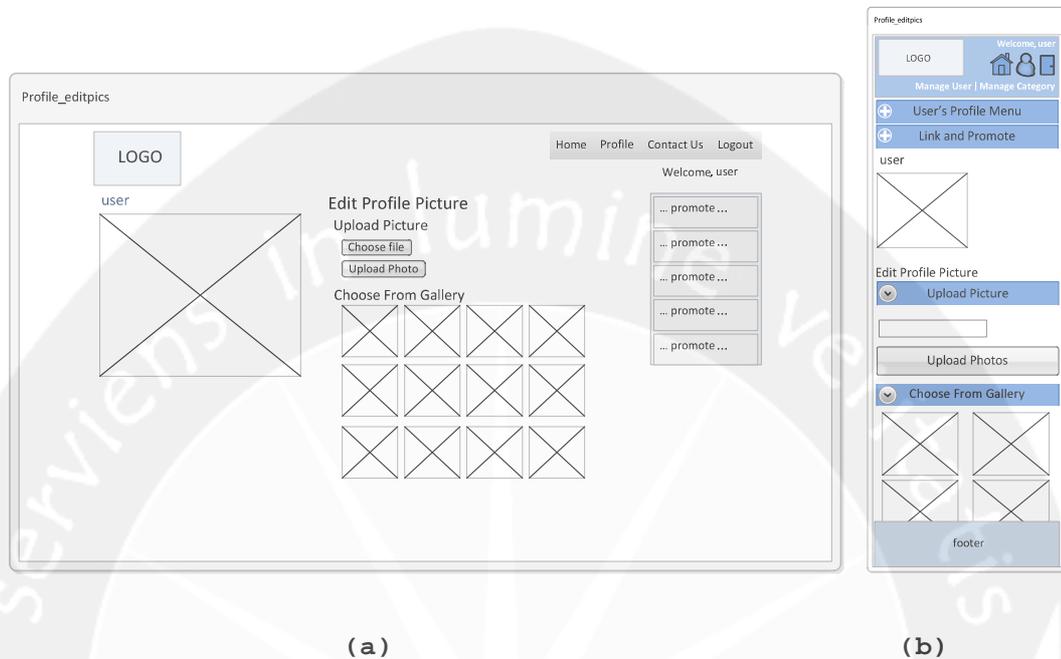
4.11 Change Password



Gambar 4.14 Rancangan Antarmuka Change Password

Antarmuka Gambar 4.14 merupakan antarmuka yang digunakan untuk mengubah password yang dimiliki user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka change password seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka change password seperti pada gambar (a). Untuk mengubah password, user dapat mengisi password yang dimiliki, password yang baru serta konfirmasi password yang baru. Setelah itu user dapat menekan tombol edit password sehingga password yang dimiliki user terupdate.

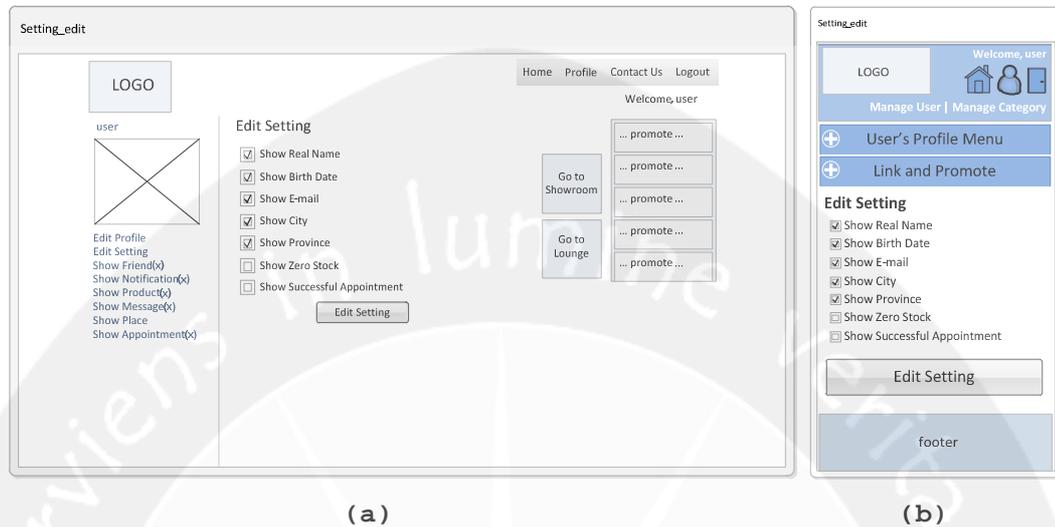
4.12 Edit Profile Picture



Gambar 4.15 Rancangan Antarmuka Edit Profile Picture

Antarmuka Gambar 4.15 merupakan antarmuka yang digunakan untuk mengubah foto profil yang dimiliki oleh user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka edit profile picture seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka edit profile picture seperti pada gambar (a). User dapat mengubah foto profil yang dimiliki menjadi foto yang sudah pernah di upload dengan cara memilih salah satu foto dari bagian "Choose From Gallery" atau mengupload gambar baru dengan menekan tombol Choose File dan memunculkan file dialog untuk memilih file gambar yang akan diupload, kemudian menekan tombol upload photo.

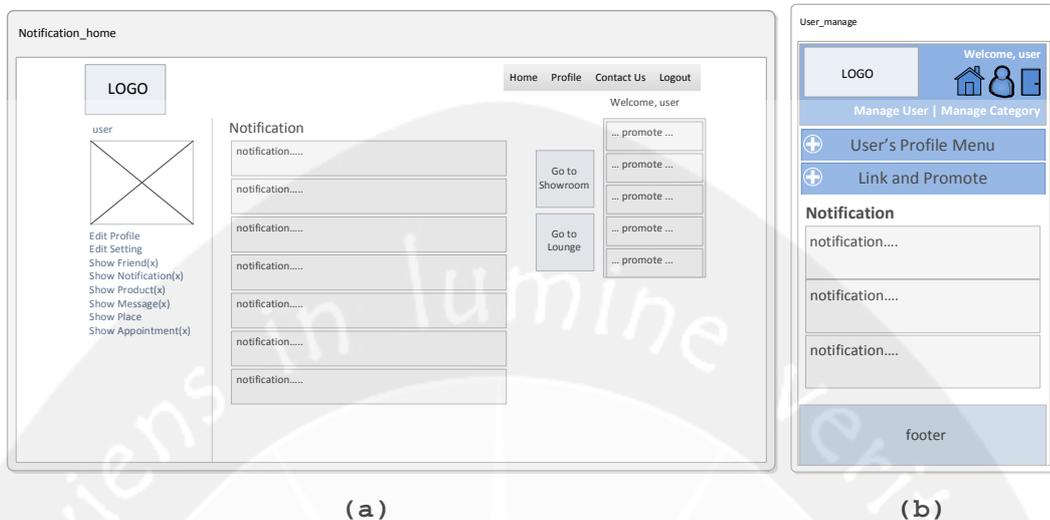
4.13 Edit Setting



Gambar 4.16 Rancangan Antarmuka Edit Setting

Antarmuka Gambar 4.16 merupakan antarmuka yang digunakan untuk mengubah setting yang dimiliki oleh user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka edit setting seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka edit setting seperti pada gambar (a). Untuk melakukan edit setting, user mengubah data setting yang telah ditampilkan kemudian menekan tombol edit setting.

4.14 Notification



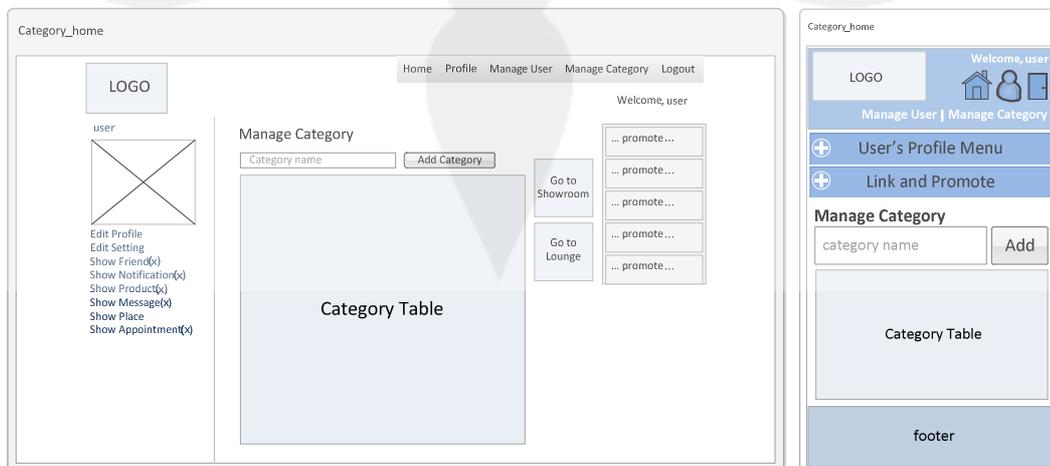
(a)

(b)

Gambar 4.17 Rancangan Antarmuka Notification

Antarmuka Gambar 4.17 merupakan antarmuka yang digunakan untuk menampilkan semua notifikasi yang dimiliki oleh user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka notifikasi seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka notifikasi seperti pada gambar (a).

4.15 Category

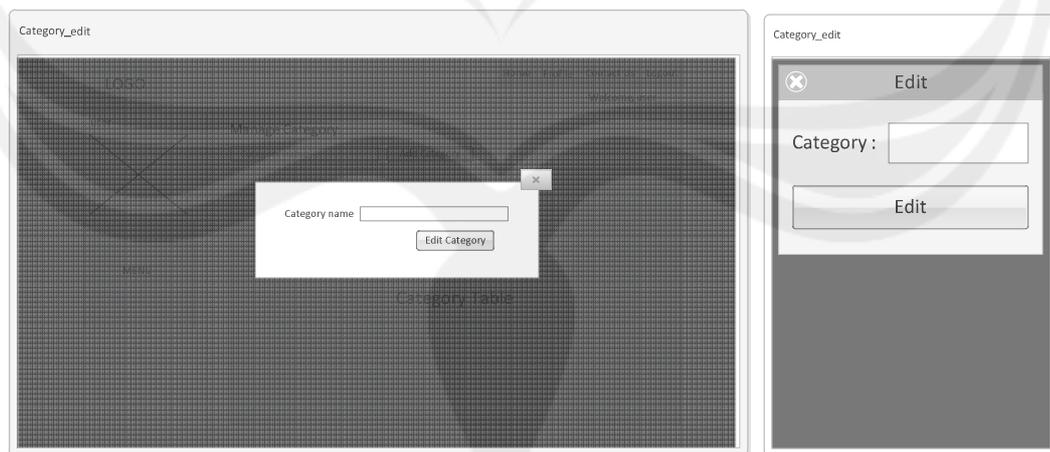


(a)

(b)

Gambar 4.18 Rancangan Antarmuka Category

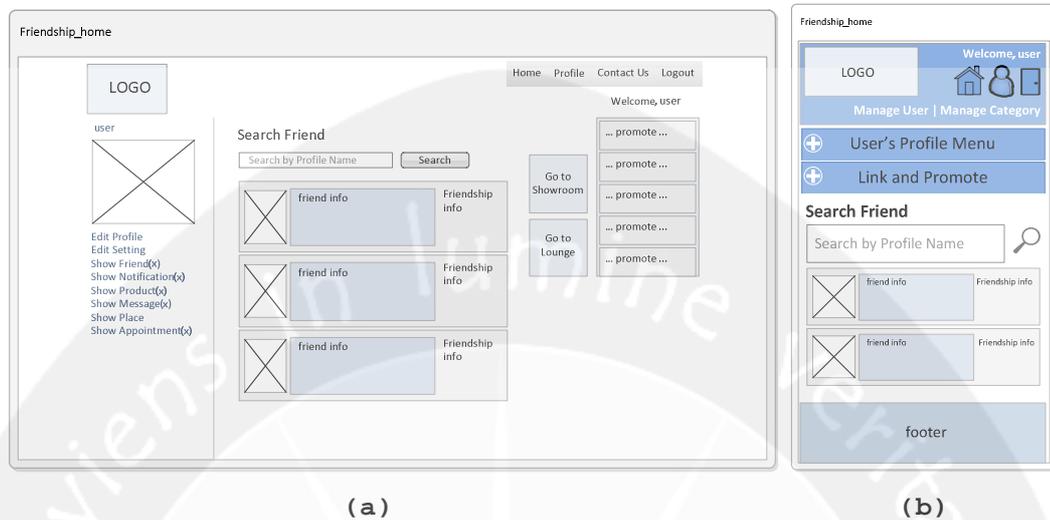
Antarmuka Gambar 4.18 merupakan antarmuka yang digunakan untuk mengelola kategori. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka manage category seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka manage category seperti pada gambar (a). Pada antarmuka ini, user dapat menambah kategori, mengedit kategori, atau mendelete kategori. Untuk menambahkan kategori, user dapat mengisikan nama kategori di textbox kemudian menekan tombol add category. Untuk mengedit kategori, user dapat menekan link update pada category table sehingga akan muncul pop up seperti gambar 4.19, kemudian user dapat mengedit nama kategori dan menekan tombol edit category sehingga category akan teredit.



Gambar 4.19 Rancangan Antarmuka Edit Category

Untuk mendelete kategori, user dapat menekan tombol delete yang terdapat pada category table.

4.16 Friendship

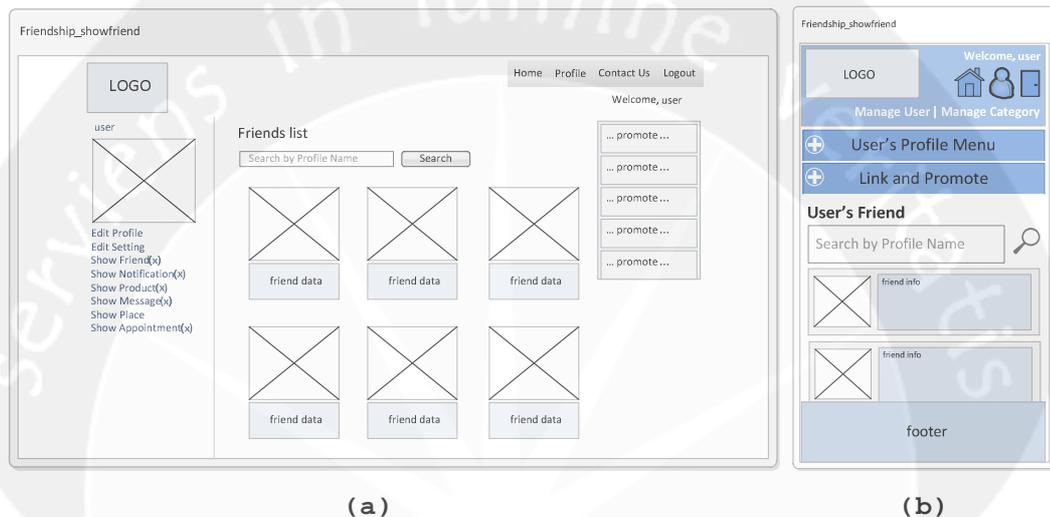


Gambar 4.20 Rancangan Antarmuka Friendship

Antarmuka Gambar 4.20 merupakan antarmuka yang digunakan untuk mencari teman. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka friendship seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka friendship seperti pada gambar (a). Untuk mencari teman dapat dilakukan dengan cara mengetikkan nama profil teman di textbox search friend kemudian menekan tombol search sehingga sistem akan mencari data pengguna situs yang memiliki nama yang mengandung kata yang diketikkan oleh user. Pada hasil pencarian akan ditampilkan foto profil user, informasi user dan informasi hubungan antar user. Apabila user belum menjadi teman dengan user yang dicari, maka user dapat menekan tombol Add as Friend pada bagian friendship info. Apabila user telah menawarkan pertemanan maka akan muncul tulisan "waiting..." Apabila user ditawarkan permintaan pertemanan oleh user yang dicari, maka user dapat

memilih untuk menerima tawaran pertemanan dengan menekan tombol accept atau menolak tawaran pertemanan dengan menekan tombol reject. Apabila user telah berteman dengan user yang dicari, maka pada friendship info akan tertulis "Friend".

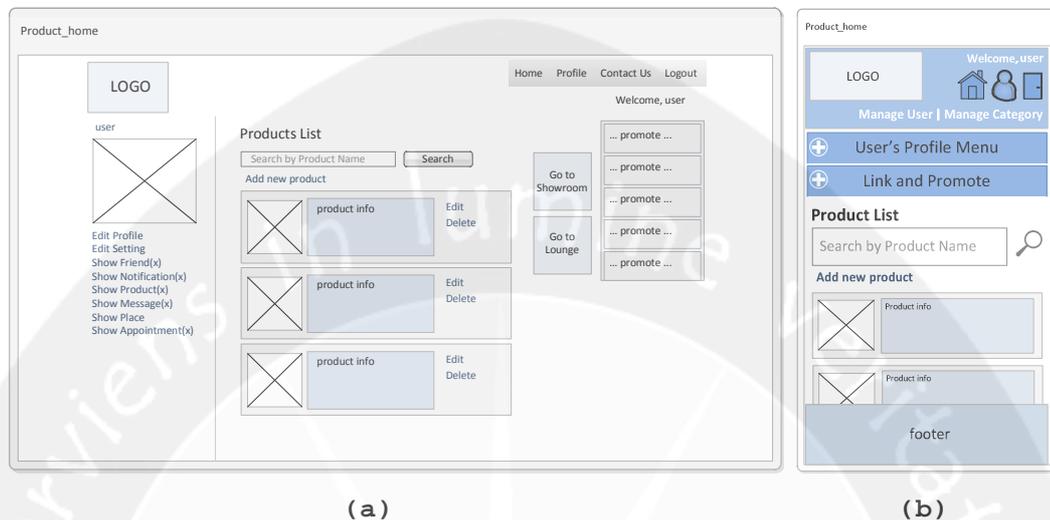
4.17 Show Friend



Gambar 4.21 Rancangan Antarmuka Show Friend

Antarmuka Gambar 4.21 merupakan antarmuka yang digunakan untuk menampilkan semua teman yang user miliki. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka show friend seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka show friend seperti pada gambar (a). Pada antarmuka ini user dapat mencari teman berdasarkan profile. Untuk mencari teman, user dapat mengetikkan nama teman yang dicari pada textbox dan menekan tombol Search. Kemudian sistem akan mencari teman yang dimiliki user yang memiliki profil name yang mengandung kata yang di cari user dan menampilkan foto profil serta data teman yang ada.

4.18 Product

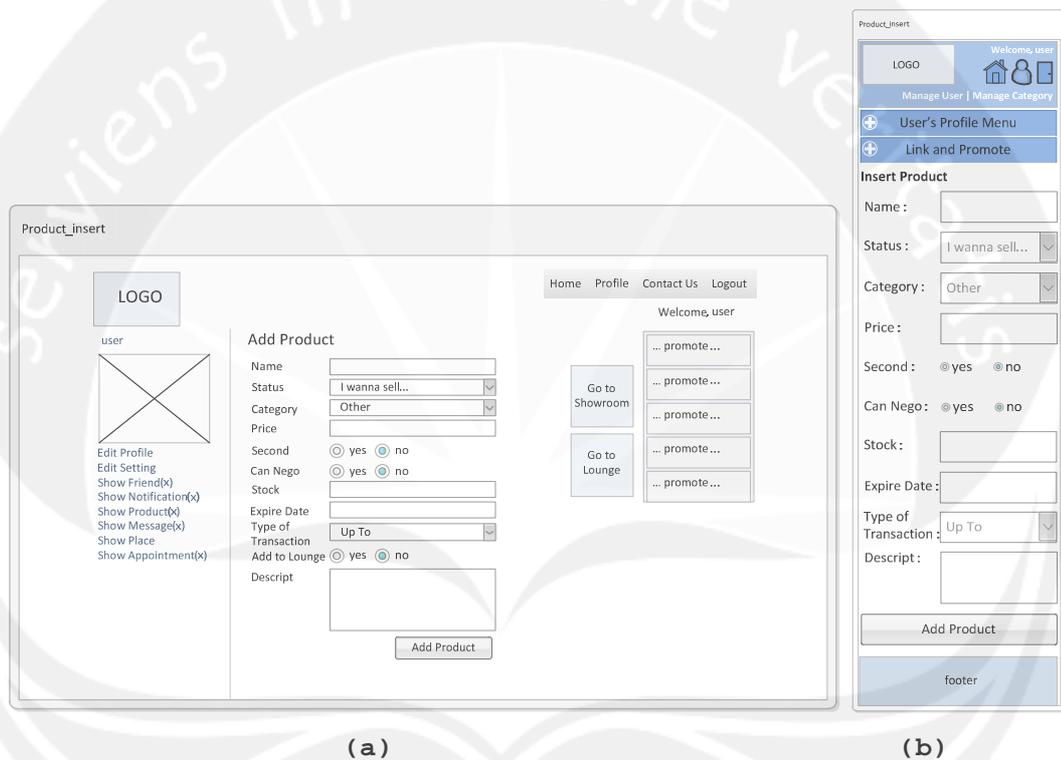


Gambar 4.22 Rancangan Antarmuka Product

Antarmuka Gambar 4.22 merupakan antarmuka yang digunakan untuk menampilkan semua produk yang dimiliki oleh user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka product seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka product seperti pada gambar (a). Pada antar muka ini user dapat melakukan pencarian produk berdasarkan nama produk, penambahan produk, edit produk, dan delete produk. Untuk melakukan pencarian produk, user dapat mengetikkan nama produk pada textbox kemudian menekan tombol search sehingga sistem akan melakukan pencarian terhadap produk-produk yang memiliki nama seperti inputan user dan menampilkan foto dan info produk. Untuk melakukan penambahan produk, user dapat mengeklik link add new product sehingga akan dihubungkan ke halaman tambah produk. Untuk mengedit produk dapat dilakukan dengan cara mengeklik link edit

product sehingga dihubungkan ke halaman edit produk. Untuk menghapus produk dapat dilakukan dengan cara mengklik link delete product sehingga sistem akan menghapus produk dari database.

4.19 Insert Product

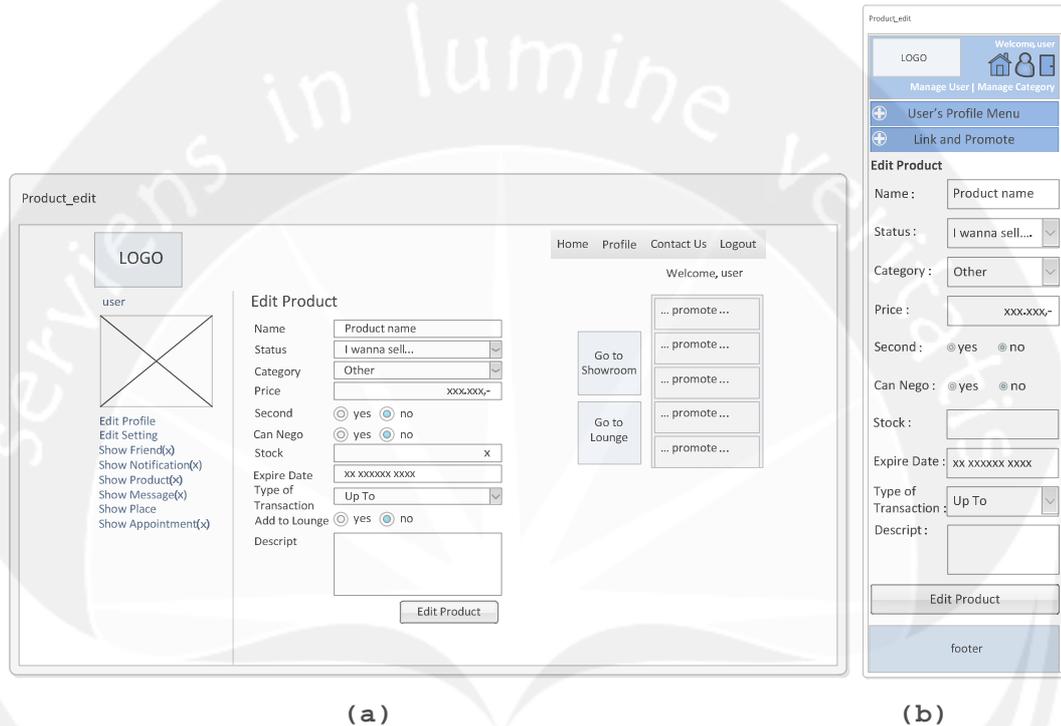


Gambar 4.23 Rancangan Antarmuka Insert Product

Antarmuka Gambar 4.23 merupakan antarmuka yang digunakan untuk menambahkan produk yang dimiliki user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka insert product seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka insert product seperti pada gambar (a). Pada antarmuka ini, user memasukkan data-data yang diminta kemudian menekan tombol add product. Apabila

semua data yang dimasukkan user valid, maka data product akan disimpan di database.

4.20 Edit Product

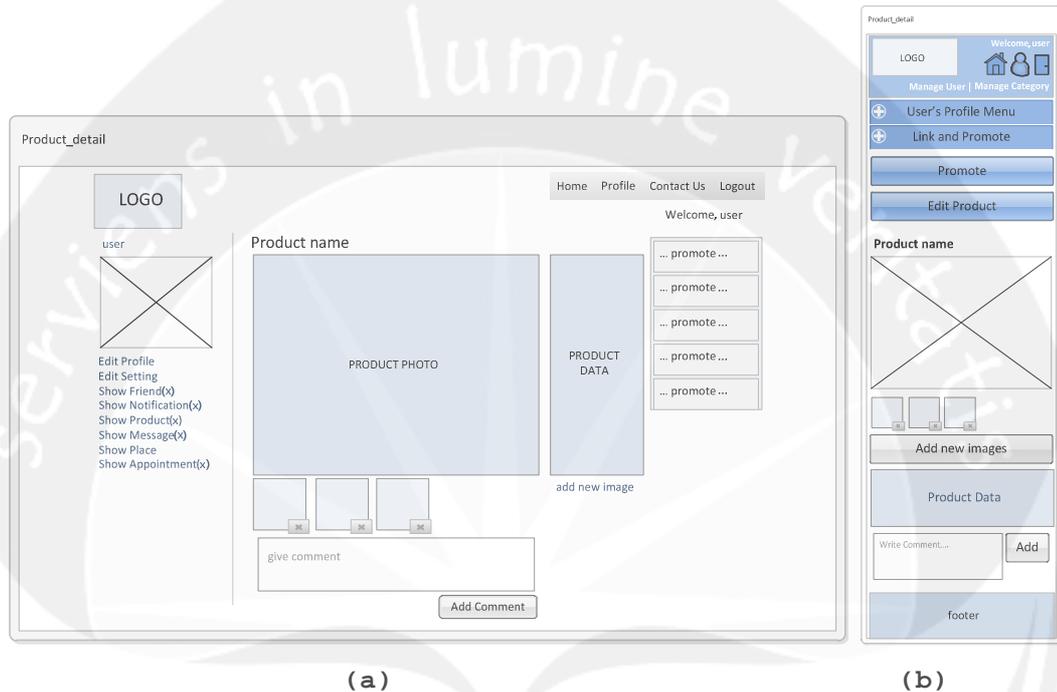


Gambar 4.24 Rancangan Antarmuka Edit Product

Antarmuka Gambar 4.24 merupakan antarmuka yang digunakan untuk mengedit data produk. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka edit product seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka edit product seperti pada gambar (a). Pada antar muka ini data produk akan ditampilkan sehingga user dapat mengedit data produk dan meminta sistem menyimpan perubahan tersebut dengan menekan tombol edit product.

Apabila data produk yang diedit oleh user valid maka sistem akan menyimpan perubahan tersebut di database.

4.21 Detail Product



Gambar 4.25 Rancangan Antarmuka Detail Product

Antarmuka Gambar 4.25 merupakan antarmuka yang digunakan untuk menampilkan detail dari produk yang dimiliki oleh user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka detail product seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka detail product seperti pada gambar (a). Pada antarmuka ini, user dapat menambahkan gambar baru, menghapus gambar produk, menambahkan komentar, mengedit komentar dan menghapus komentar. Untuk menambahkan gambar baru, user dapat menekan link add new image sehingga muncul pop up

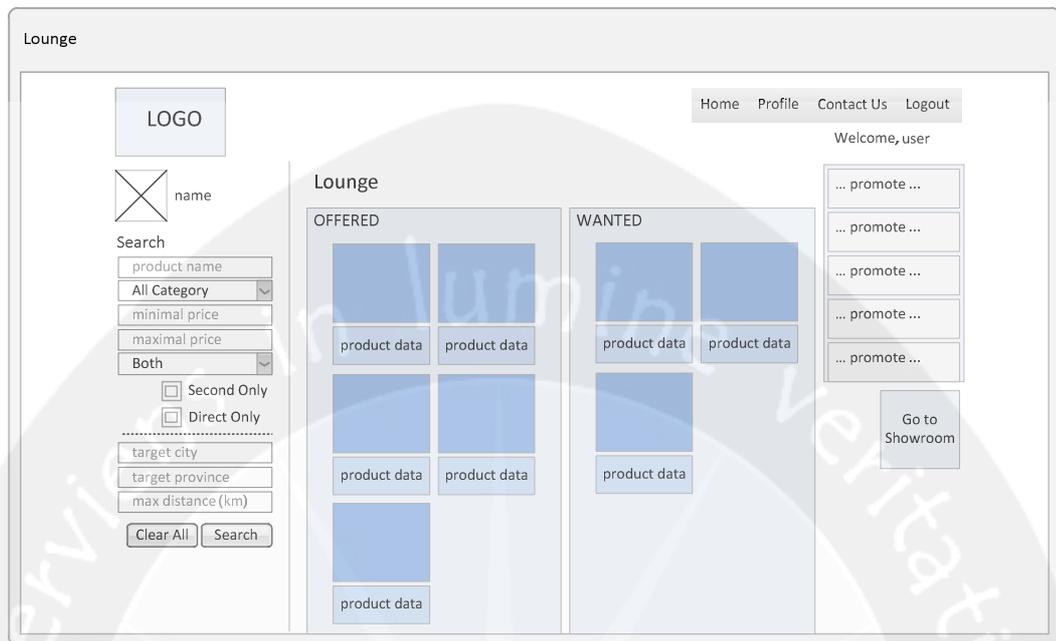
seperti pada gambar 4.26. Kemudian user dapat menekan tombol choose file untuk memunculkan file dialog dan memilih file gambar yang ingin diupload, setelah itu memberikan deskripsi foto dan menekan tombol upload photo untuk menambahkan foto pada produk.



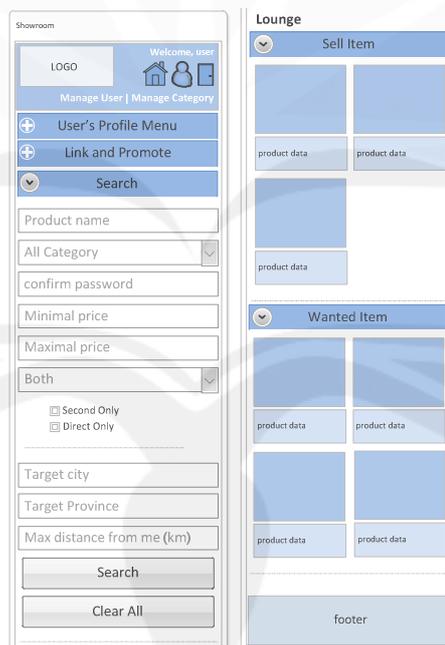
Gambar 4.26 Rancangan Antarmuka Insert Picture

Untuk menghapus foto produk dapat dilakukan dengan cara menekan gambar hapus yang terdapat di sebelah kanan bawah foto produk. Untuk menambahkan komentar dapat dilakukan dengan cara mengetikkan komentar pada textarea kemudian mengeklik tombol add comment. Untuk delete komentar hanya dapat dilakukan oleh user yang membuat komentar yaitu dengan cara mengeklik gambar hapus yang terdapat di sebelah kanan komentar.

4.22 Lounge



(a)



(b)

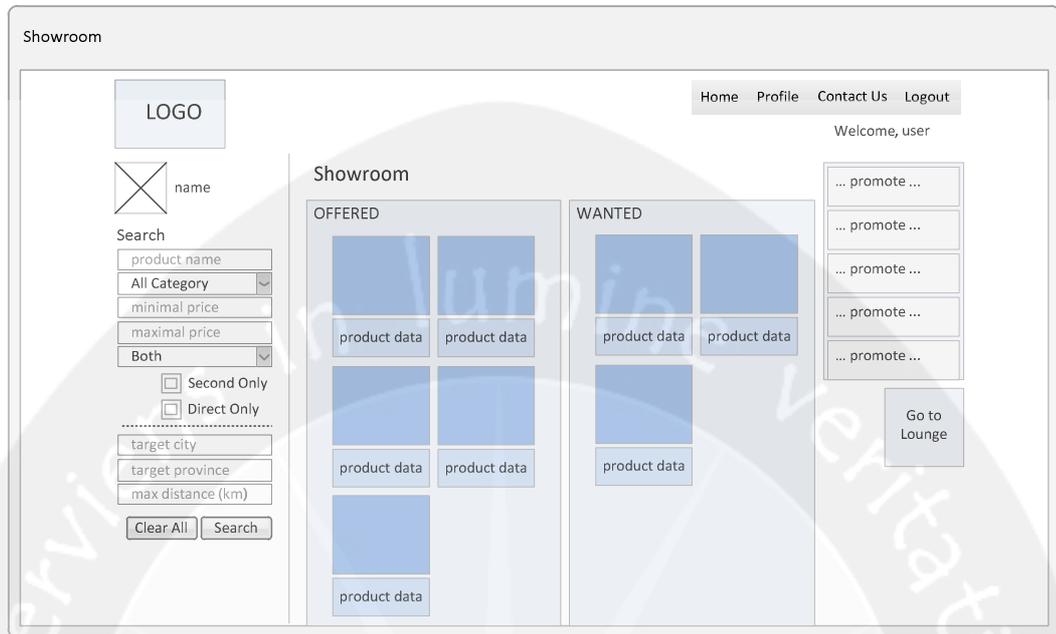
Gambar 4.27 Rancangan Antarmuka Lounge

Antarmuka Gambar 4.27 digunakan untuk melakukan menampilkan semua produk yang memiliki lounge sama dengan y. Apabila user melakukan akses ke situs dengan

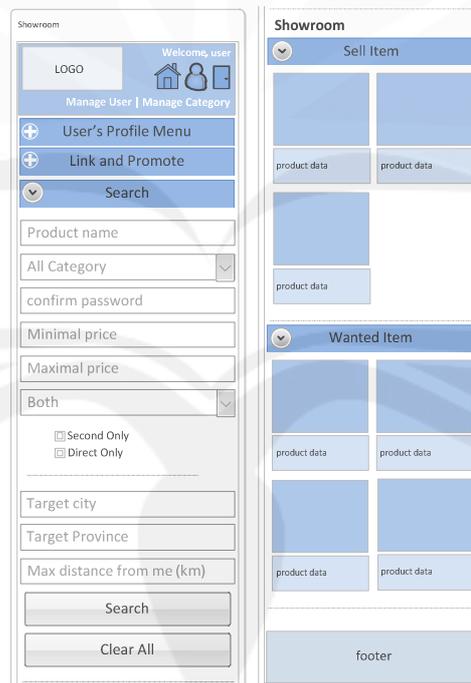
menggunakan perangkat mobile maka akan tampil antarmuka lounge seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka lounge seperti pada gambar (a). Pada Lounge di tampilan web, produk yang ditampilkan dipisahkan berdasarkan statusnya yaitu sebelah kiri merupakan produk yang ingin dijual dan sebelah kanan merupakan produk yang ingin dibeli user. Pada bagian sebelah kiri merupakan bagian search yang memiliki textbox, combobox, dan check box untuk memasukkan data produk yang ingin dicari. Sedangkan pada tampilan mobile, produk yang ingin dijual berada di bagian atas dan produk yang ingin dibeli user berada di bagian atas. Pada bagian atas sebelum tampilan produk merupakan bagian search. Untuk mencari produk, user dapat mengisikan data produk pada bagian search kemudian menekan tombol search. Untuk menghapus semua data yang terisi dapat digunakan tombol clear all.

Program Studi Teknik Informatika	DPPL – SPP	139/ 152
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		

4.23 Showroom



(a)



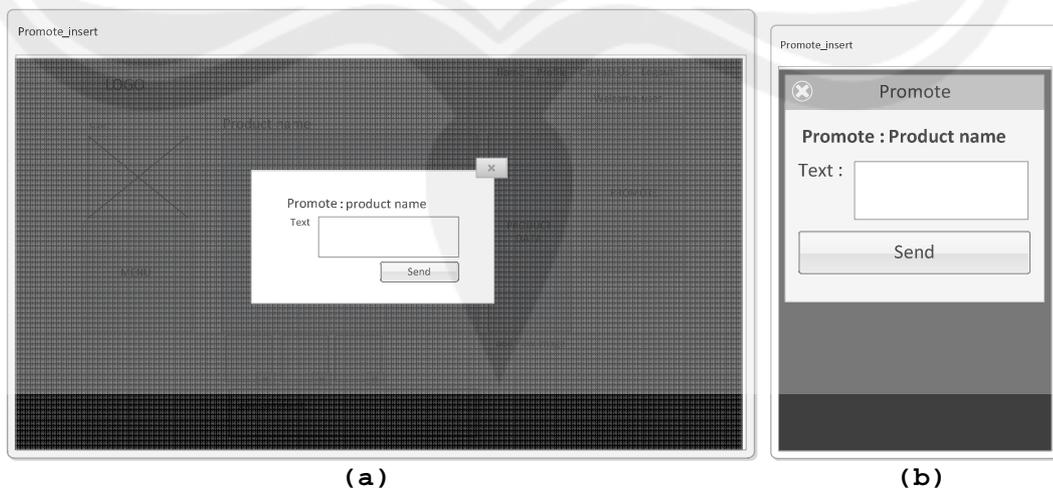
(b)

Gambar 4.28 Rancangan Antarmuka Showroom

Antarmuka Gambar 4.28 digunakan untuk melakukan menampilkan semua produk yang dimiliki oleh user yang

telah berteman dengan user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka showroom seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka showroom seperti pada gambar (a). Pada Showroom di tampilan web, produk yang ditampilkan dipisahkan berdasarkan statusnya yaitu sebelah kiri merupakan produk yang ingin dijual dan sebelah kanan merupakan produk yang ingin dibeli user. Pada bagian sebelah kiri merupakan bagian search yang memiliki textbox, combobox, dan check box untuk memasukkan data produk yang ingin dicari. Sedangkan pada tampilan mobile, produk yang ingin dijual berada di bagian atas dan produk yang ingin dibeli user berada di bagian atas. Pada bagian atas sebelum tampilan produk merupakan bagian search. Untuk mencari produk, user dapat mengisikan data produk pada bagian search kemudian menekan tombol search. Untuk menghapus semua data yang terisi dapat digunakan tombol clear all.

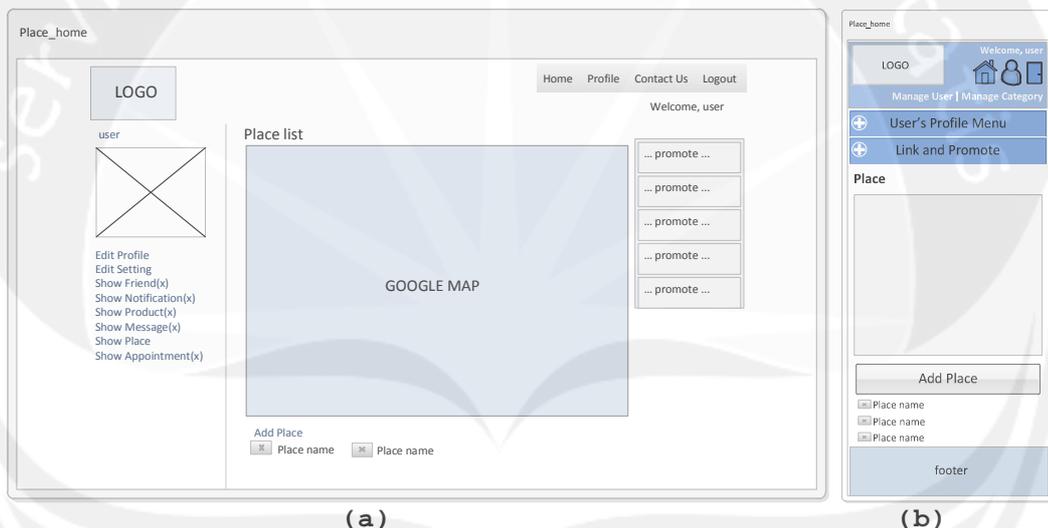
4.24 Insert Promote



Gambar 4.29 Rancangan Antarmuka Promote

Antarmuka Gambar 4.29 merupakan pop up yang digunakan untuk menambahkan promote. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka insert promote seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka insert promote seperti pada gambar (a). Untuk menambahkan promote, dilakukan dengan cara mengetikkan text promote dan menekan tombol Send.

4.25 Place



Gambar 4.30 Rancangan Antarmuka Place

Antarmuka Gambar 4.30 merupakan antarmuka untuk menampilkan tempat yang dimiliki oleh user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka place seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka place seperti pada gambar (a). Pada antar muka ini, user dapat melihat tempat yang dimiliki di bagian google map, menambahkan tempat, atau menghapus tempat. Untuk melihat tempat yang dimiliki, user dapat menekan

Program Studi Teknik Informatika	DPPL – SPP	142/ 152
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		

link Place name sehingga pusat google map akan diarahkan pada lokasi tempat itu berada. Untuk menambahkan tempat, user dapat menekan link Add Place sehingga akan dihubungkan ke halaman insert place. Untuk menghapus tempat, user dapat menekan gambar hapus yang terdapat di sebelah kiri Place name yang ingin dihapus.

4.26 Insert Place

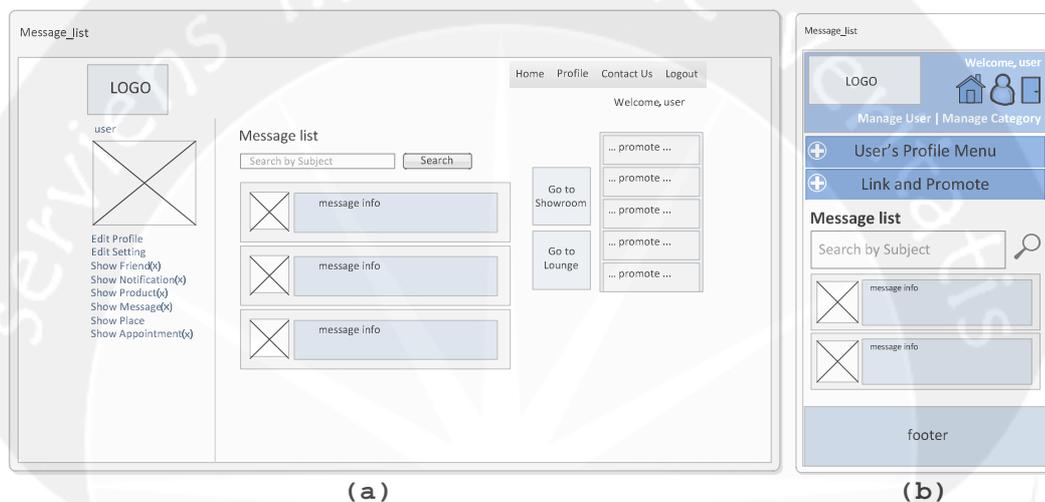


Gambar 4.31 Rancangan Antarmuka Insert Place

Antarmuka Gambar 4.31 merupakan antarmuka yang digunakan untuk menambahkan tempat yang dimiliki user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka insert place seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka insert place seperti pada gambar (a). Untuk menginputkan longitude dan latitude pada tampilan web, user dapat mengarahkan point pada google map ke tempat yang ingin ditambahkan, sedangkan untuk tampilan mobile, user dapat mengeklik google map pada bagian

yang ingin ditambahkan. Kemudian user dapat mengisikan judul tempat dan deskripsi tempat di textbox dan textarea, setelah itu menekan tombol add place untuk meminta sistem menyimpan data tempat tersebut. Apabila data valid maka sistem akan menyimpan data tempat tersebut ke database.

4.27 List Message

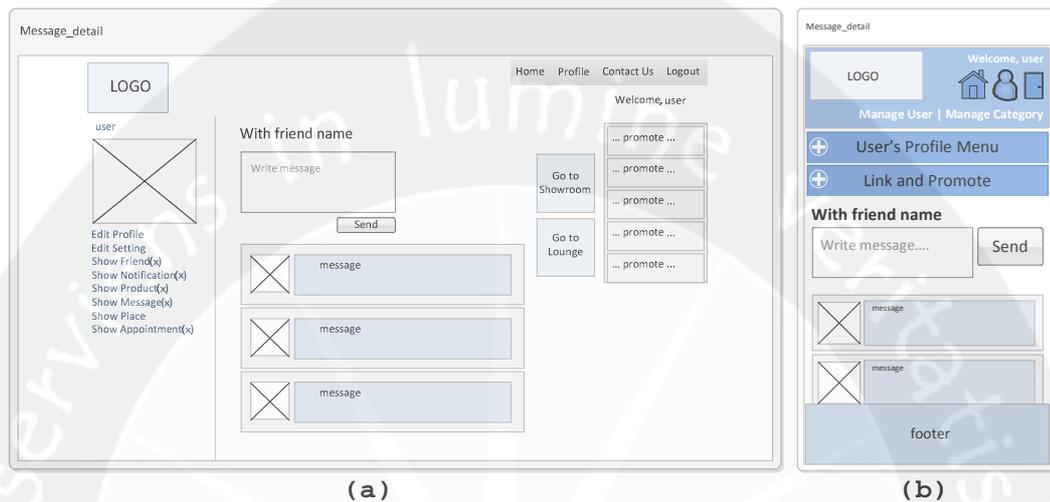


Gambar 4.32 Rancangan Antarmuka List Message

Antarmuka Gambar 4.32 merupakan antarmuka yang digunakan untuk menampilkan semua pesan yang dimiliki oleh user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka list message seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka list message seperti pada gambar (a). Pada antarmuka ini terdapat fungsi search yang digunakan untuk mencari pesan berdasarkan subject. Untuk melakukan pencarian pesan dapat dilakukan dengan cara mengetikkan subjek dari pesan yang di cari pada textbox, kemudian menekan tombol search. Setelah itu sistem akan mencari pesan yang memiliki subject yang

mengandung inputan dari user dan menampilkannya dalam bentuk foto profil user yang berkirim-kiriman pesan dengan user serta informasi pesan.

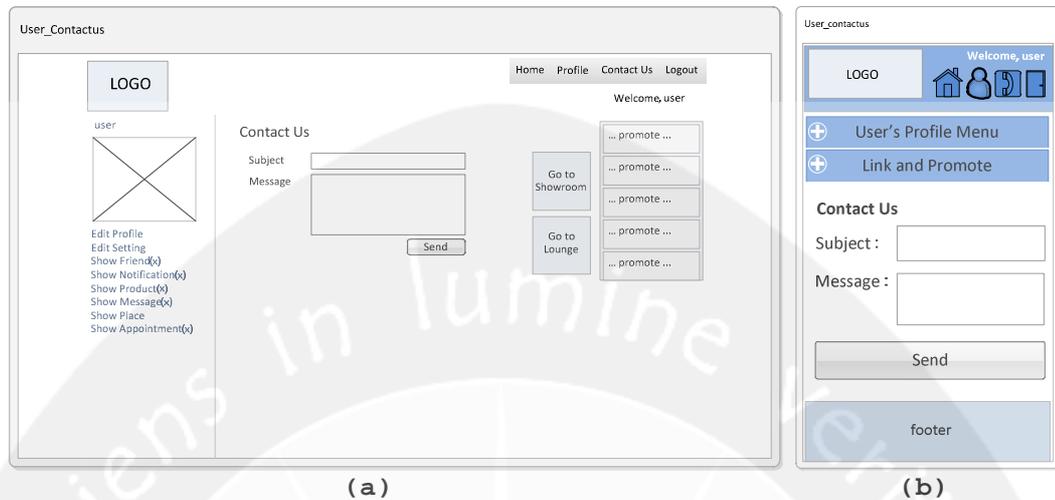
4.28 Detail Message



Gambar 4.33 Rancangan Antarmuka Detail Message

Antarmuka Gambar 4.33 merupakan antarmuka yang digunakan untuk menampilkan detail pesan yang dimiliki. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka detail message seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka detail message seperti pada gambar (a). Pada antarmuka ini dapat dilakukan pengiriman pesan dengan cara mengetikkan pesan pada textarea setelah itu menekan tombol send. Apabila pesan terkirim maka pesan akan ditampilkan pada bagian bawah situs.

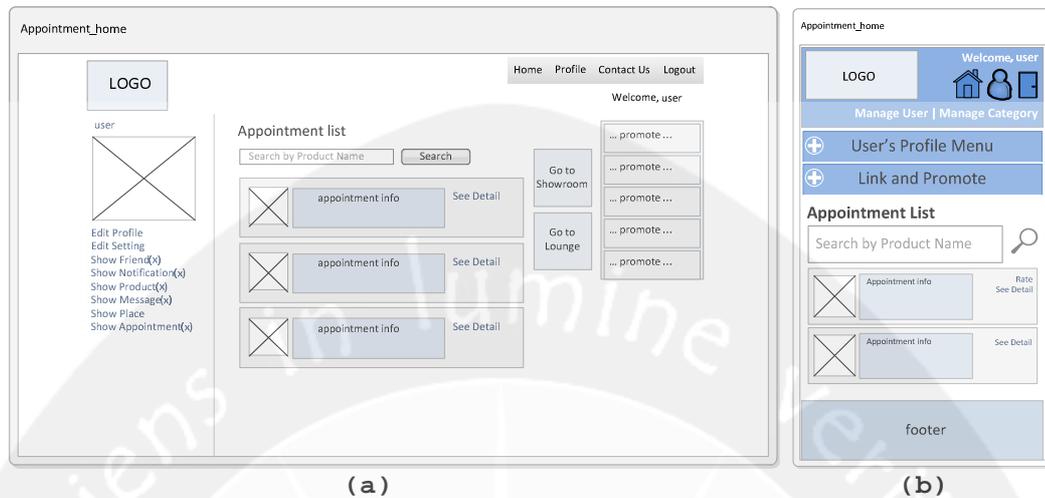
4.29 Contact Us



Gambar 4.34 Rancangan Antarmuka Contact Us

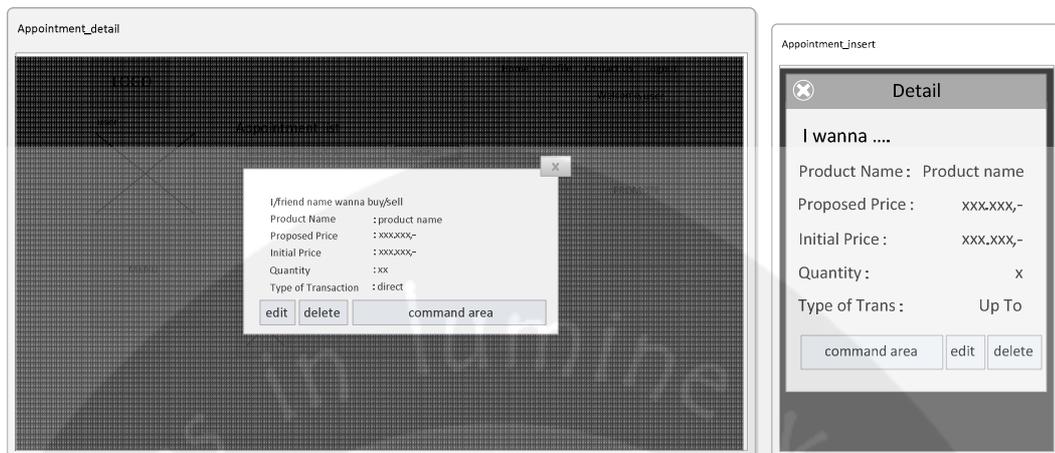
Antarmuka Gambar 4.34 merupakan antarmuka yang digunakan oleh user untuk mengirimkan pesan ke administrator situs. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka contact us seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka contact us seperti pada gambar (a). Untuk mengirimkan pesan dapat dilakukan dengan cara mengisi subject dan pesan pada textbox dan textarea kemudian menekan tombol send untuk mengirimkan pesan tersebut.

4.30 List Appointment



Gambar 4.35 Rancangan Antarmuka List Appointment

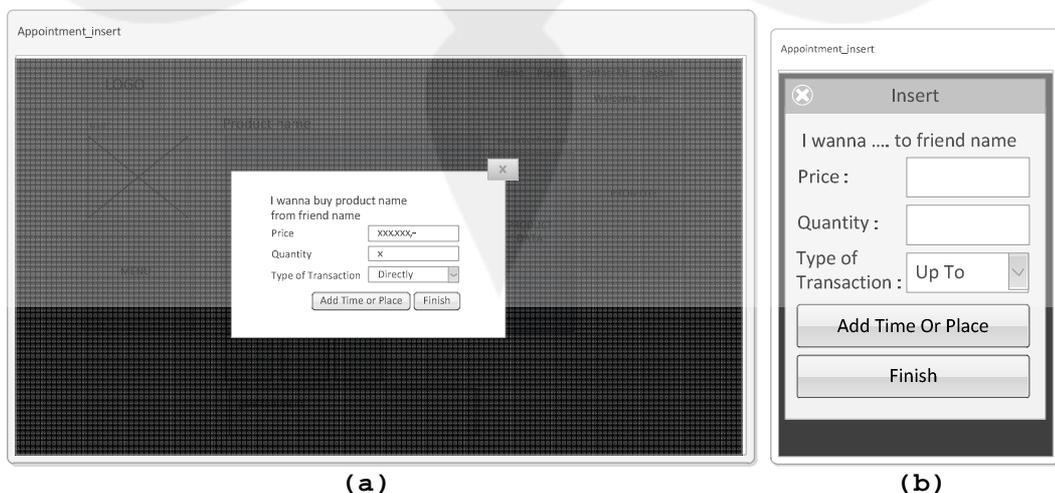
Antarmuka Gambar 4.35 merupakan antarmuka yang digunakan untuk menampilkan semua appointment yang dimiliki oleh user. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka list appointment seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka list appointment seperti pada gambar (a). Pada antarmuka ini terdapat fungsi untuk mencari appointment yang dimiliki berdasarkan nama produk dengan cara mengetikkan nama produk dari appointment yang dicari pada textbox kemudian menekan tombol search. Kemudian sistem akan mencari dan menampilkan semua appointment yang memiliki nama produk yang mengandung kata yang diinputkan. Untuk melihat detail appointment dapat dilakukan dengan cara mengeklik link See Detail sehingga akan ditampilkan pop up seperti pada gambar 4.36.



Gambar 4.36 Rancangan Antarmuka Detail Appointment

Kemudian user dapat melakukan edit appointment apabila user merupakan orang yang menawarkan appointment dan appointment tersebut belum diterima atau telah ditolak dengan menekan gambar edit, user dapat melakukan delete appointment untuk meminta pembatalan appointment dengan menekan gambar delete, menerima atau menolak appointment apabila appointment tersebut merupakan permintaan dari user lain, menyatakan bahwa user sudah mengirim barang atau uang kepada user lawan, atau sudah menerima barang atau uang dari user lawan.

4.31 Insert Appointment



Gambar 4.37 Rancangan Antarmuka Insert Appointment

Antarmuka Gambar 4.37 merupakan pop up yang digunakan untuk menambahkan appointment. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka insert appointment seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka insert appointment seperti pada gambar (a). Apabila user ingin menambahkan appointment, user dapat mengisi data price, quantity dan type of transaction kemudian menekan tombol finish untuk meminta sistem mengirimkan appointment pada target. Kemudian sistem akan mengirimkan appointment jika data yang diinputkan valid. Apabila user ingin menambahkan waktu atau tempat pada appointment, user dapat menekan tombol Add Time or Place yang akan menghubungkan user ke halaman Insert time or place dengan membawa data yang telah diisikan user.

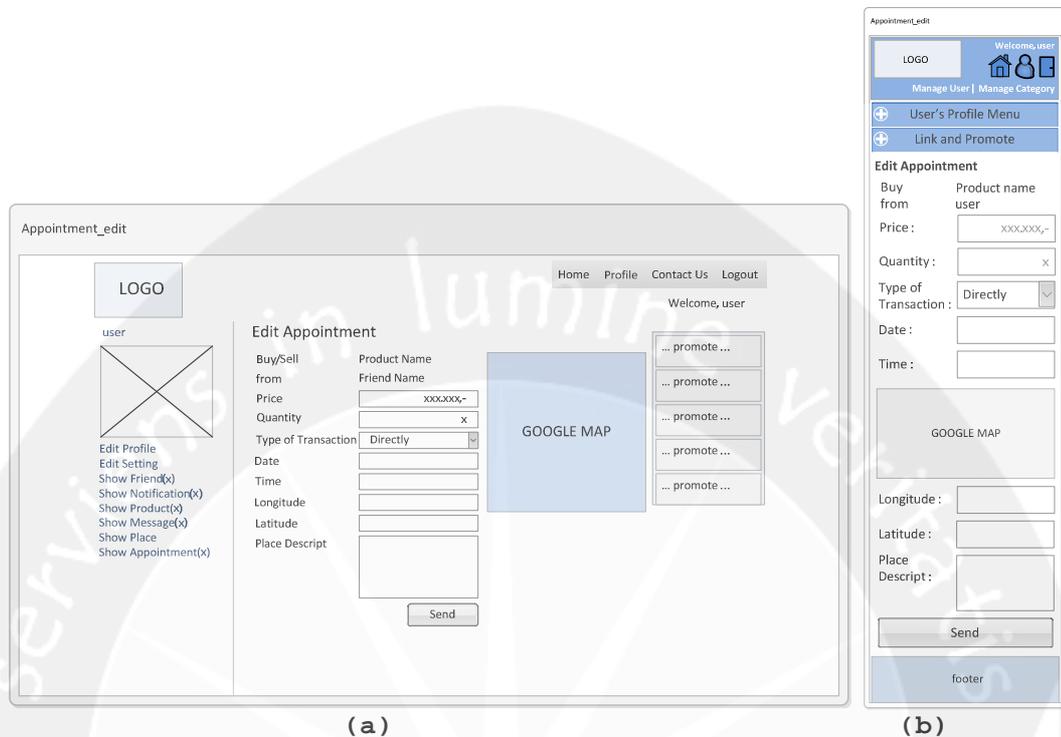
4.32 Insert Time or Place Appointment



Gambar 4.38 Rancangan Antarmuka Insert Time or Place Appointment

Antarmuka Gambar 4.38 merupakan antarmuka yang digunakan untuk menambahkan appointment secara lengkap. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka insert time or place appointment seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka insert time or place appointment seperti pada gambar (a). Untuk menambahkan appointment, user dapat mengisi data-data kemudian menekan tombol send untuk meminta sistem mengirimkan appointment yang telah dibuat. Apabila semua data yang diisikan valid, maka sistem akan mengirimkan appointment tersebut.

4.33 Edit Appointment



Gambar 4.39 Rancangan Antarmuka Edit Appointment

Antarmuka Gambar 4.39 merupakan antarmuka yang digunakan untuk mengedit appointment. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka edit appointment seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka edit appointment seperti pada gambar (a). Pada antarmuka ini akan ditampilkan semua data dari appointment yang akan diedit sehingga user dapat mengubah data tersebut dan menekan tombol send untuk meminta sistem menyimpan perubahan yang telah dibuat. Apabila semua data yang telah diedit valid, maka sistem akan menyimpan perubahan yang telah dibuat.

4.34 Rate Appointment



Gambar 4.40 Rancangan Antarmuka Rate Appointment

Antarmuka Gambar 4.40 merupakan antarmuka yang digunakan untuk melakukan rate pada appointment yang telah selesai dilakukan. Apabila user melakukan akses ke situs dengan menggunakan perangkat mobile maka akan tampil antarmuka rate appointment seperti pada gambar (b). Apabila akses dilakukan dengan perangkat yang bukan mobile maka akan tampil antarmuka rate appointment seperti pada gambar (a). Pada antarmuka ini user dapat memilih rating yang akan diberikan untuk appointment yang telah dilakukan. Setelah user memilih rating dan menekan tombol submit, maka data rate tersebut akan tersimpan di database