

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan dari pembahasan-pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan dari Tugas Akhir ini :

1. Forum Jual Beli (LAPAK) yang dibangun telah berhasil dibuat dan berjalan dengan baik untuk dapat melakukan fungsi yang memudahkan promosi barang dengan baik.
2. Forum Jual Beli (LAPAK) yang dibangun telah diintegrasikan *google map* dan jejaring sosial *twitter*.
3. Forum Jual Beli (LAPAK) dapat dipahami pengguna dengan mudah karena tampilan dibuat dengan sederhana.

6.2 Saran

Beberapa saran dan masukan yang dapat disampaikan penulis terhadap pembuatan Forum Jual Beli ini dimasa yang akan datang adalah :

1. Sistem diharapkan bisa dikembangkan lebih lanjut dengan memaksimalkan integrasi dengan jejaring *social* yang ada seperti memposting pada halaman *facebook* sehingga promosi dengan situs jejaring sosial dapat digunakan lebih maksimal untuk mempromosikan barang dalam bentuk *Thread* dalam sistem.

DAFTAR PUSTAKA

- Adriana, Dian ., 2003, Pengenalan Program *E-commerce* dengan PHP dan MySql, http://learning.unla.ac.id/ft/praktikum/sim_tutorial/web%20dan%20internet/tutorial%20pengenalan%20ecommerce%20dg%20php%20%26%20mysql.pdf, diakses tanggal 3 September 2012.
- Anonim, 2008, *Kamus Besar Bahasa Indonesia*, Balai Pustaka, Jakarta.
- Deptan, 2004, <http://www.deptan.go.id/pusdatin/admin/RB/Programming/Materi%20PHP.pdf>, diakses tanggal 3 September 2012
- EllisLab, 2010, *CodeIgniter User Guide Version 1.7*, <http://codeigniter.com/>, diakses tanggal 14 November 2011.
- G.M. Arya Sasmita, P. A. Bayupati, dan Gerlan A. Manu, 2009. Sistem Transaksi Penjualan Kain Gringsing Berbasis Mobile (*Mobile Commerce*). Staff Pengajar Teknik Elektro, Fakultas Teknik, Universitas Udayana, Vol.8 No.1, Hal.91-96.
- Garofakalas, John., Stefanis, Vassilios., 2007. Using RSS Feeds For Effective Mobile Web Browsing. *Univ Acces InfSoc*, Vol.6 No.DOI 10.1007/s10209-007-0086-8, Hal.249-257.
- Handayani, Sisti, 2010. *Laris Manis Jual Beli Lewat Kaskus*, Mediakom, Yogyakarta.
- Hill, 2009, <http://www.xtremepapers.com/SAT/McGraw-Hill%27s%20SAT%2C%202009.pdf>, diakses tanggal 3 September 2012

- Kadir, Abdul, 2009. *Konsep & Tuntunan Praktis Basis Data*, Andi, Yogyakarta.
- P Farley, M Capp, 2005. *Mobile Web Service*. *BT Technology Journal*, Vol.23 No.2, Hal.202-213, 11hal.
- Rozi, I. F., 2008, *MVC - Model, View, Controller*, <http://ifrozi.wordpress.com/2008/01/09/mvc-model-view-controller/>, diakses tanggal 3 September 2012.
- Santoso, Gatot., Susanto, Adhi., Wardani, M. Budi, 2009. *Perancangan Konten M-learning Dengan Sistem Live Multimedia Berbasis Selular*. *Jurnal Teknologi*, Vol.2 No.2, Hal.124-129.
- Suryanegara, 2009, <http://digilib.itttelkom.ac.id>, diakses tanggal 3 September 2012
- Suyanto, Asep, 2010, http://jurnalkomputer.com/attachments/005_BASIS%20DATA%20DAN%20DBMS.pdf, diakses tanggal 3 September 2012.
- Maffeis, S., 2005, *Client-server Computing*, media.wiley.com/assets/152/06/computer.pdf, diakses tanggal 3 September 2012.
- Boss, R. W., 2006, *Client-server Technology*, *American Library Association*, pp.1-4.
- Deacon, J., 2009, *Model-View-Controller (MVC) Architecture*, <http://www.jdl.co.uk/briefings/index.html#mvc>, diakses tanggal 3 September 2012.
- Morgan, E., 2004, *Client-server Model of Computing*, <http://infomotions.com/musings/waves/clientservercomputing.html>, diakses tanggal 3 September 2012.
- Rahayu, S., 2008, *Sekapur Sirih Tentang Framework CodeIgniter*, *Open Knowledge and Education*, 1-7.

Sofwan, A., 2007, *Belajar PHP dengan Framework CodeIgniter*, ilmukomputer.org/wp-content/uploads/2010/05/belajar-php-dengan-framework-code-igniter.pdf, diakses tanggal 3 September 2012.

Steiniger, Stefan., Neun, Moritz., Edwardes, . Alistair, 2006, *Foundations of Location Based Services*, https://docs.google.com/viewer?a=v&q=cache:AiYgQ_vrPfmJ:www.spatial.cs.umn.edu/Courses/Fall11/8715/papers/IM7_steiniger.pdf+&hl=en&gl=id&pid=bl&srcid=ADGEESjGmA9YyVmKyFgoswkdTdEemByDhp-0tVjXhvIhekCDq-deXpQ2HR3SzV7oKA9LBNA6hBmj_lSZkIs3BIkGue1fiyh4qskFljrzENlT6daNmhhzr2snXqhFwAHOInNlwSf0QlLC&sig=AHIEtbQiwlv2k8T0zAuaSMFjGhyWKBoQiw, diakses pada tanggal 24 Oktober 2012.

<https://developers.google.com/maps/documentation/javascript/tutorial>, diakses tanggal 3 September 2012.

<http://www.kaskus.co.id>, diakses pada tanggal 24 Oktober 2012.

<http://jquerymobile.com/demos/1.2.0/docs/about/intro.html>, diakses tanggal 24 Oktober 2012.

www.techsource.ala.org, diakses pada tanggal 24 Oktober 2012.

SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

LAPAK

(PEMBANGUNAN FORUM JUAL BELI DENGAN MENGGUNAKAN
MOBILE-WEB BERBASIS LOKASI)

Dipersiapkan oleh:

Yohanes Adven Stefanus / 08 07 05507

Program Studi Teknik Informatika - Fakultas Teknologi
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		SKPL-LAPAK		1/32
		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

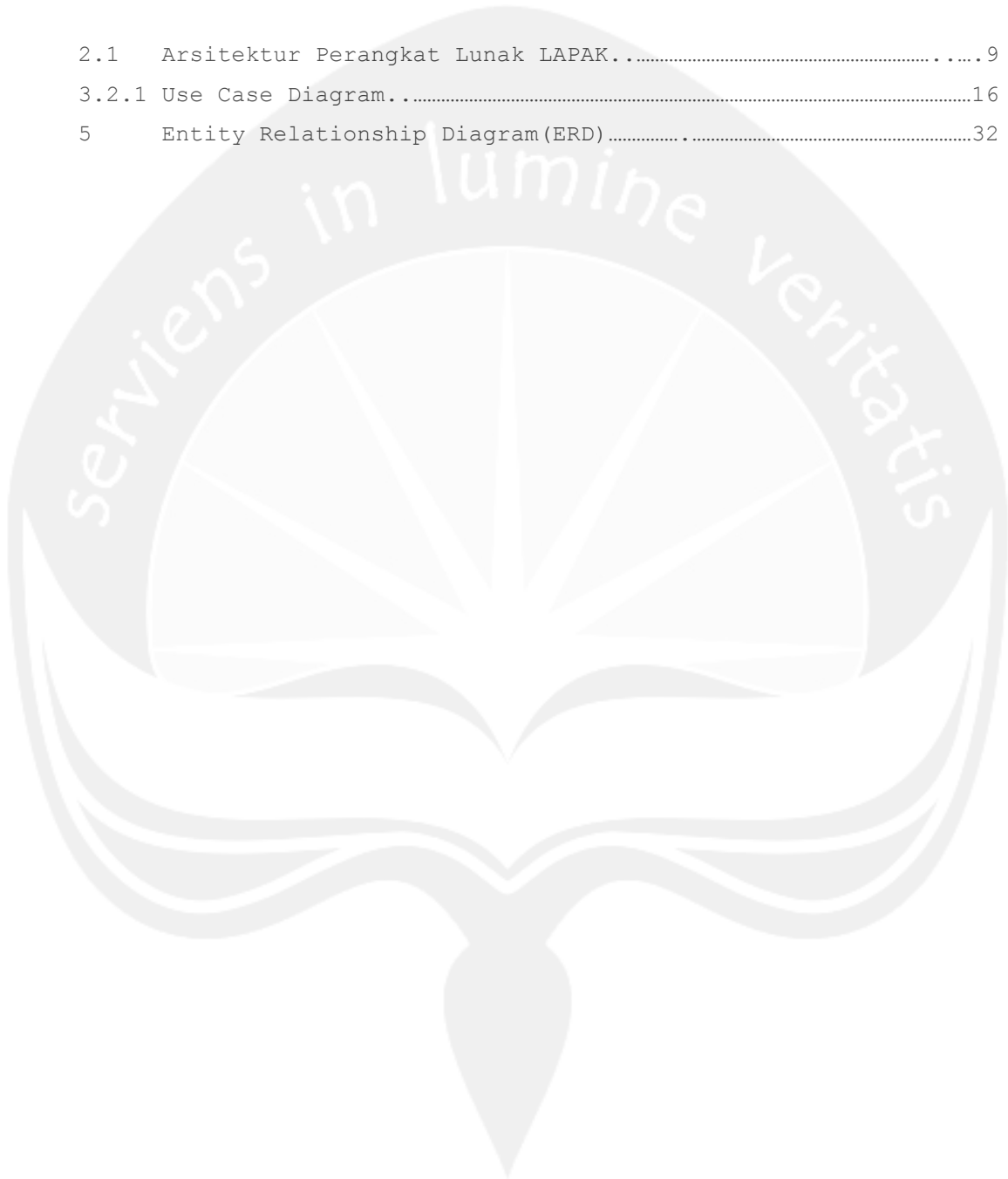
Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan	6
1.1	Tujuan	6
1.2	Lingkup Masalah	6
1.3	Definisi, Akronim dan Singkatan	7
1.4	Referensi	7
1.5	Deskripsi umum (Overview)	7
2	Deskripsi Kebutuhan	8
2.1	Perspektif produk	8
2.2	Fungsi Produk	10
2.3	Karakteristik Pengguna	13
2.4	Batasan-batasan	13
2.5	Asumsi dan Ketergantungan	13
3	Kebutuhan khusus	14
3.1	Kebutuhan antarmuka eksternal	14
3.2	Kebutuhan fungsionalitas Perangkat Lunak	16
4	Spesifikasi Rinci Kebutuhan	17
4.1	Spesifikasi Kebutuhan Fungsionalitas	17
5	Entity Relationship Diagram (ERD)	32

Daftar Gambar

2.1	Arsitektur Perangkat Lunak LAPAK.....	9
3.2.1	Use Case Diagram.....	16
5	Entity Relationship Diagram (ERD).....	32



1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak LAPAK untuk mendefinisikan kebutuhan perangkat lunak yang akan dikembangkan meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna), tempat penyimpanan yang dibutuhkan, serta keakuratan), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-LAPAK ini juga mendefinisikan batasan perancangan perangkat lunak.

Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya.

1.2 Lingkup Masalah

Perangkat Lunak LAPAK dikembangkan dengan tujuan untuk :

1. Menangani login pengguna.
2. Menangani pendaftaran member.
3. Menangani *view thread*.
4. Menangani *create thread*.
5. Menangani *edit thread*.
6. Menangani ubah status *thread*.
7. Menangani *cari thread*.
8. Menangani *view comment*.
9. Menangani *post comment*.
10. Menangani lihat lokasi.
11. Menangani pengelolaan *thread*.

Program Studi Teknik Informatika	SKPL – LAPAK	6/ 32
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		

12. Menangani pengelolaan kategori barang.

13. Menangani pengelolaan user.

14. Menangani pengelolaan *comment*.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-LAPAK-XXX	Kode yang merepresentasikan kebutuhan pada LAPAK (Berdagang di Awan) dimana XXX merupakan nomor fungsi produk.
LAPAK	Perangkat lunak untuk membuat <i>thread</i> pada <i>mobile-web</i> .
<i>Thread</i>	Tulisan atau buah pikiran seseorang dalam forum online.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Adven Stefanus, Yohanes, Spesifikasi Kebutuhan Perangkat Lunak (SKPL), Jurusan Teknik Informatika UAJY.
2. Upton, David, *CodeIgniter for Rapid PHP Application Development*, Packt Publishing, 2007.

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 2 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL,

Program Studi Teknik Informatika	SKPL – LAPAK	7/ 32
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		

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 LAPAK 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 LAPAK tersebut.

2 Deskripsi Kebutuhan

2.1 Perspektif produk

LAPAK merupakan perangkat lunak yang dikembangkan untuk membantu menjual barang dalam bentuk membuat *thread* untuk media promosi. Sistem ini dapat menangani login pengguna, pendaftaran member, *view thread*, *create thread*, *edit thread*, ubah status *thread*, cari *thread*, *view comment*, *post comment*, pengelolaan *thread*, pengelolaan kategori barang, pengelolaan user, dan pengelolaan *comment*. Selain itu sistem ini juga menangani pencarian lokasi penjual.

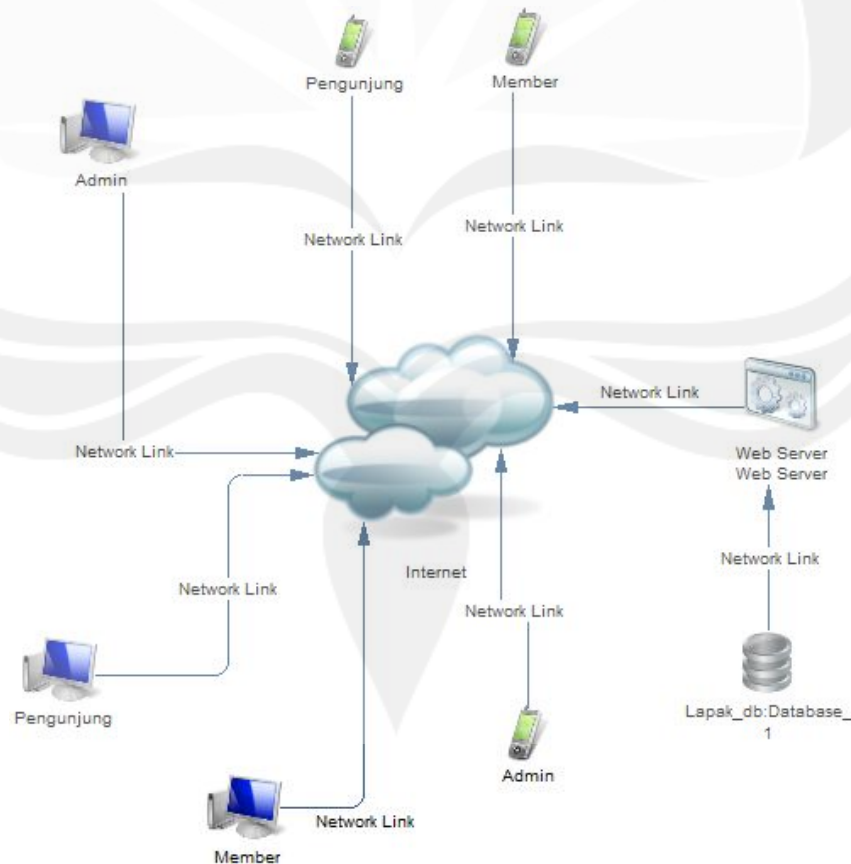
Perangkat lunak ini dapat dijalankan pada perangkat *mobile* dan dibuat menggunakan bahasa pemrograman PHP dengan menggunakan *framework* CodeIgniter 2.1.1. Sedangkan untuk lingkungan pemrogramannya menggunakan NetBeans IDE 7.0.1.

Perangkat lunak ini dapat diakses oleh client pada lingkungan sistem operasi apapun yang mempunyai aplikasi penjelajah situs serta mendukung *JQuery Mobile* (web browser) untuk web-user.

Program Studi Teknik Informatika	SKPL – LAPAK	8/ 32
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		

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. Ada 1 karakteristik end-user dalam sistem ini, yaitu: web-user. Web-user dapat mengakses data yang ada di server tersebut secara *online* membuka web site yang tersedia di web server.

Inputan data yang dimasukkan akan disimpan dalam database server, sehingga jika ada pencarian data, maka data yang diinginkan akan dicari ke database server yang selanjutnya dikirimkan ke client yang merequest melalui atau tanpa melalui web server tergantung dari karakteristik user-nya.



Gambar 2.1. Arsitektur Perangkat lunak LAPAK

2.2 Fungsi Produk

Fungsi produk perangkat lunak LAPAK adalah sebagai berikut :

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

Merupakan fungsi yang digunakan oleh seluruh pengguna sistem (administrator dan member) untuk dapat masuk dalam LAPAK.

2. Fungsi *Pendaftaran* (**SKPL-LAPAK-002**).

Merupakan fungsi yang digunakan untuk menambahkan data pengunjung ke dalam basis data sehingga pengunjung tersebut memiliki akun sebagai member dan dapat melakukan pembuatan *thread*.

3. Fungsi *View Thread* (**SKPL-LAPAK-003**).

Merupakan fungsi yang digunakan oleh pengguna untuk melihat *thread*.

4. Fungsi *Create Thread* (**SKPL-LAPAK-004**).

Merupakan fungsi yang digunakan oleh pengguna untuk membuat *thread*.

5. Fungsi *Edit Thread* (**SKPL-LAPAK-005**).

Merupakan fungsi yang digunakan oleh pengguna untuk mengubah data *thread*.

6. Fungsi *Ubah Status Thread* (**SKPL-LAPAK-006**).

Merupakan fungsi yang digunakan oleh pengguna untuk mengubah Status *thread*.

7. Fungsi *Cari Thread* (**SKPL-LAPAK-007**).

Merupakan fungsi yang digunakan oleh pengguna sistem untuk mencari suatu barang atau mencari *thread* yang diinginkan.

8. Fungsi *View Comment* (**SKPL-LAPAK-008**).

Merupakan fungsi yang digunakan untuk melihat *comment* dari pengguna lain.

9. Fungsi *Post Comment* (**SKPL-LAPAK-009**).

Merupakan fungsi yang digunakan oleh pengguna untuk mem-*posting comment*.

10. Fungsi Lihat Lokasi (**SKPL-LAPAK-010**).

Merupakan fungsi yang digunakan oleh pengguna untuk melihat lokasi penjual.

11. Fungsi Pengelolaan *Thread* (**SKPL-LAPAK-011**)

Merupakan fungsi yang digunakan untuk mengelola pembuatan.

Fungsi Pengelolaan *Thread* meliputi :

a. Fungsi *View Thread* (**SKPL-LAPAK-011-01**)

Merupakan fungsi yang digunakan administrator untuk menampilkan data *thread*.

b. Fungsi *Delete Thread* (**SKPL-LAPAK-011-02**)

Merupakan fungsi yang digunakan administrator untuk menghapus data *thread*.

12. Fungsi Pengelolaan Kategori Barang (**SKPL-LAPAK-012**).

Merupakan fungsi yang digunakan untuk mengelola kategori barang.

Fungsi Pengelolaan kategori barang mencakup :

Program Studi Teknik Informatika	SKPL – LAPAK	11/ 32
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. Fungsi *View* Kategori Barang (**SKPL-LAPAK-012-01**)

Merupakan fungsi yang digunakan untuk menampilkan data kategori barang

b. Fungsi *Add* Kategori Barang (**SKPL-LAPAK-012-02**)

Merupakan fungsi yang digunakan untuk menambahkan data kategori barang.

c. Fungsi *Edit* Kategori Barang (**SKPL-LAPAK-012-03**)

Merupakan fungsi yang digunakan untuk mengubah data kategori barang.

13. Fungsi Pengelolaan User (**SKPL-LAPAK-013**).

Merupakan fungsi yang digunakan untuk mengelola user.

Fungsi Pengelolaan user mencakup :

a. Fungsi *View User* (**SKPL-LAPAK-013-01**)

Merupakan fungsi yang digunakan untuk menampilkan data user.

b. Fungsi *Delete User* (**SKPL-LAPAK-013-02**)

Merupakan fungsi yang digunakan untuk menghapus data user.

14. Fungsi Pengelolaan *Comment* (**SKPL-LAPAK-014**).

Merupakan fungsi yang digunakan untuk mengelola *comment*.

Fungsi Pengelolaan *comment* mencakup :

a. Fungsi *View Comment* (**SKPL-LAPAK-014-01**)

Merupakan fungsi yang digunakan untuk menampilkan *comment*.

b. Fungsi *Delete Comment* (**SKPL-LAPAK-014-02**)

Merupakan fungsi yang digunakan untuk menghapus *comment*.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak LAPAK adalah sebagai berikut :

1. Member

- Memahami pengoperasian komputer, dan perangkat mobile.
- Mengerti tentang internet.
- Memahami pengoperasian web *browser*.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak LAPAK tersebut adalah :

1. Kebijaksanaan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak LAPAK.

2. Keterbatasan perangkat keras

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

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada perangkat komputer dan perangkat *mobile* (PC, Laptop, tablet-PC, *handphone*, dll) yang menggunakan sistem operasi Windows

Program Studi Teknik Informatika	SKPL – LAPAK	13/ 32
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		

XP, windows 7 untuk desktop-user dan segala sistem operasi untuk web-user dan segala perangkat *mobile* yang mendukung *JQuery Mobile*.

3 Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak LAPAK 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 LAPAK adalah:

1. Perangkat komputer dan perangkat *mobile* (yang mendukung *JQuery Mobile*).

3.1.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak LAPAK di sisi server adalah sebagai berikut :

1. Nama : MySQL

Sumber : MySQL

Sebagai database management system (DBMS) yang digunakan untuk penyimpan data di sisi server.

2. Nama : Apache

Program Studi Teknik Informatika	SKPL – LAPAK	14/ 32
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		

Sumber : Apache.

Sebagai web server.

3. Nama : Codeigniter

Sumber : Codeigniter.

Sebagai framework untuk membangun aplikasi.

Sedangkan perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak LAPAK di sisi client adalah sebagai berikut :

1. Nama : Microsoft Windows 95/98/NT/2000/XP, Android OS.

Sebagai sistem operasi untuk web-user.

2. Nama : Microsoft Internet Explorer, Netscape Navigator, Opera, Safari, Mozilla Firefox, Konqueror, Nautilus, Lynx, dll.

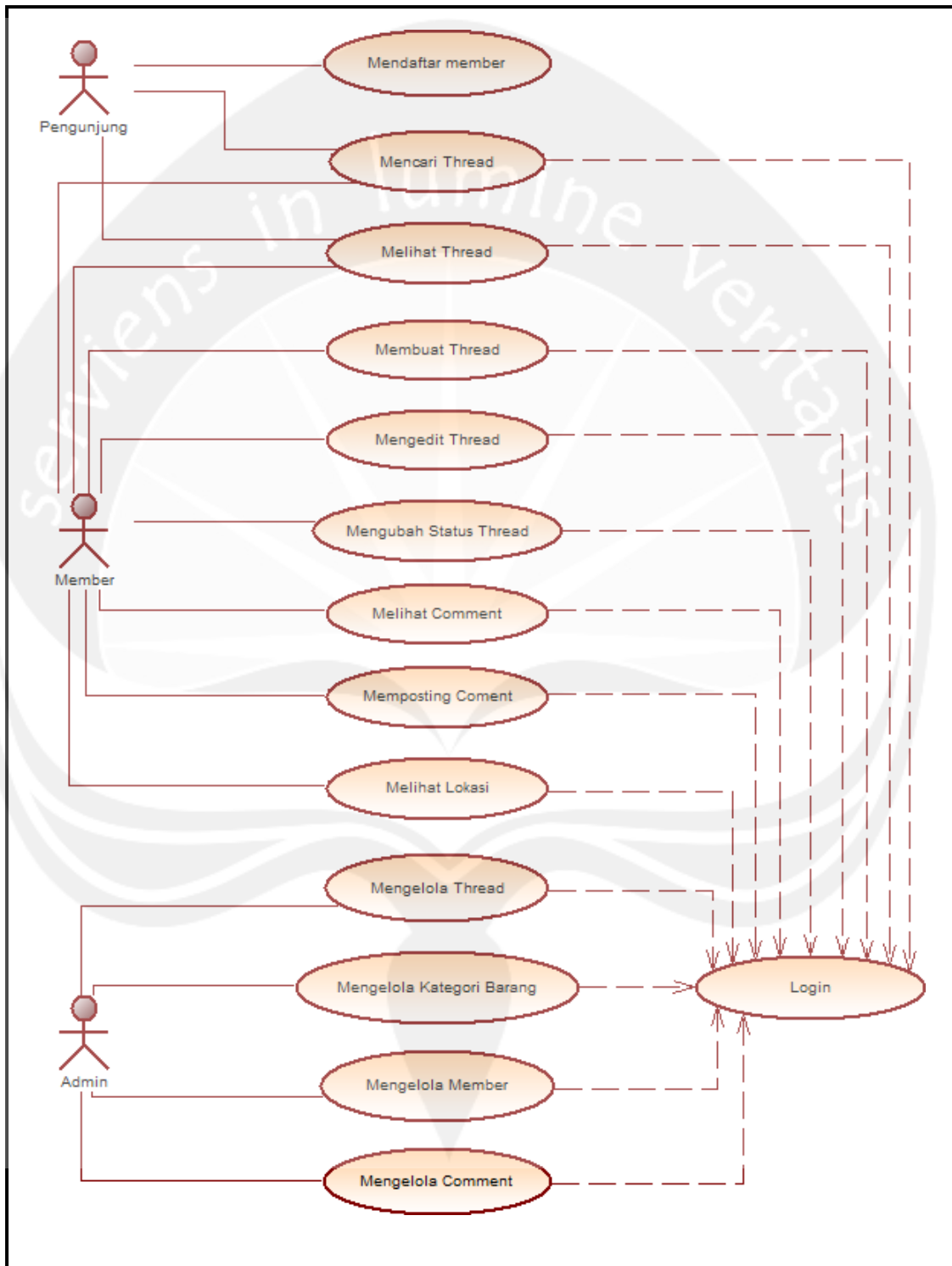
Sebagai penjelajah situs (web browser) berbasis *mobile*.

3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak LAPAK terhubung secara client-server dalam lingkup jaringan Internet.

3.2 Kebutuhan fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram



Gambar 3.2.1. Use Case Diagram

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 sistem.

2. Primary Actor

1. Administrator
2. Member

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 username dan password.
4. Sistem memeriksa username dan password yang diinputkan aktor.

E-1 Password dan username user tidak sesuai.

5. Sistem memberikan akses ke aktor.
6. Use Case ini selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Password dan username user tidak sesuai

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

7. PreConditions

1. Sistem berjalan dengan baik dan lancar.

Program Studi Teknik Informatika	SKPL – LAPAK	17/ 32
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 memiliki koneksi internet sehingga dapat terhubung dengan sistem.
3. Sistem berjalan dengan baik dan lancar.
4. Aktor telah terdaftar pada sistem.
5. Aktor telah membuka halaman antarmuka login.

8. PostConditions

1. Aktor dapat memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem.

4.1.2 Use case Spesification: Mendaftar

1. Brief Description

Use Case ini digunakan oleh aktor untuk melakukan pendaftaran pengguna untuk bergabung ke dalam sistem.

2. Primary Actor

2. Pengunjung

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pendaftaran.
2. Aktor memilih untuk melakukan pendaftaran member.
3. Sistem menampilkan antarmuka untuk melakukan pendaftaran member.
4. Aktor memasukkan username, password, email, jenis kelamin, no telepon, alamat lengkap, latitude, dan longitude.
5. Sistem memeriksa data yang diinputkan aktor valid atau tidak.

E-1 Data yang diinputkan tidak valid.

Program Studi Teknik Informatika	SKPL – LAPAK	18/ 32
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 menampilkan antarmuka yang menunjukkan bahwa pendaftaran telah berhasil.

7. Use Case ini selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Data yang diinputkan tidak valid.

1. Sistem menampilkan peringatan bahwa data yang diinputkan tidak valid.

2. Kembali ke Basic Flow langkah ke-2.

7. PreConditions

1. Sistem berjalan dengan baik dan lancar.

2. Aktor telah membuka halaman antarmuka pendaftaran member.

8. PostCondition

1. Aktor telah terdaftar sebagai member dari sistem.

2. Aktor telah memiliki akun untuk memperoleh akses ke sistem.

4.1.3 Use case Spesification: Mencari Thread

1. Brief Description

Use Case ini digunakan oleh aktor untuk mencari *thread*.

2. Primary Actor

1. Member

2. Pengunjung

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pencarian *thread*.

Program Studi Teknik Informatika	SKPL – LAPAK	19/ 32
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 memasukan barang yang dicari
 - E-1 Jenis barang tidak ditemukan.
3. Sistem menampilkan hasil pencarian.
4. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

E-1

1. Jenis barang tidak ditemukan.
2. Kembali ke basic Flow 2.

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostCondition

1. Aktor melihat *thread* yang diinginkan.

4.1.4 Use case Spesification: Melihat Thread

1. Brief Description

Use Case ini digunakan oleh aktor untuk melihat *thread*. Aktor dapat melakukan *view*.

2. Primary Actor

1. Member
2. Pengunjung

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melihat *thread*.
2. Sistem menampilkan data *thread*.
3. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

Program Studi Teknik Informatika	SKPL – LAPAK	20/ 32
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

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostCondition

1. Aktor melihat *thread* yang diinginkan.

4.1.5 Use case Spesification: Membuat Thread

1. Brief Description

Use Case ini digunakan oleh aktor untuk membuat *thread*.

2. Primary Actor

1. Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk membuat *thread*.
2. Sistem menampilkan halaman antarmuka untuk membuat *thread*.
3. Aktor menginputkan data *thread*.
4. Aktor meminta sistem untuk menyimpan data yang telah diinputkan.
5. Sistem melakukan pengecekan terhadap data yang telah diinputkan.
E-1 Data *thread* yang diinputkan tidak valid.
6. Sistem menambahkan data *thread* ke dalam database dan menampilkan antarmuka yang menunjukkan bahwa pembuatan *thread* telah berhasil.
7. Use Case selesai.

5. Alternative Flow

none

Program Studi Teknik Informatika	SKPL – LAPAK	21/ 32
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. Error Flow

E-1 Data *thread* yang diinputkan tidak valid

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan tidak valid.
2. Berlanjut ke Basic Flow langkah ke-3.

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostCondition

1. Data dalam database telah ter-*update*.

4.1.6 Use case Spesification: Mengedit *Thread*

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengedit *thread*.

2. Primary Actor

1. Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengedit *thread*.
2. Sistem menampilkan halaman antarmuka untuk mengedit *thread*.
3. Aktor menginputkan data *thread*.
4. Aktor meminta sistem untuk menyimpan data yang telah diinputkan.
5. Sistem melakukan pengecekan terhadap data yang telah diinputkan.
E-1 Data *thread* yang diinputkan tidak valid.
6. Sistem meng-*update* data *thread* dalam database dan menampilkan antarmuka yang menunjukkan bahwa *update thread* telah berhasil.

Program Studi Teknik Informatika	SKPL – LAPAK	22/ 32
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. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Data *thread* yang diinputkan tidak valid

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan tidak valid.
2. Berlanjut ke Basic Flow langkah ke-3.

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostCondition

1. Data dalam database telah *ter-update*.

4.1.7 Use case Spesification: Mengubah Status

Thread

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengubah status *thread*.

2. Primary Actor

1. Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengubah status *thread*.
2. Sistem menampilkan halaman antarmuka untuk mengubah status *thread*.
3. Aktor meminta sistem untuk mengubah status *thread*.
4. Sistem meng-*update* data *thread* dalam database dan menampilkan antarmuka yang menunjukkan bahwa ubah status *thread* telah berhasil.

Program Studi Teknik Informatika	SKPL – LAPAK	23/ 32
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		

5. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostCondition

1. Data dalam database telah ter-update.

4.1.8 Use case Spesification: Melihat Comment

1. Brief Description

Use Case ini digunakan oleh aktor untuk melihat *comment*.

2. Primary Actor

1. Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melihat *comment*.

2. Sistem menampilkan halaman antarmuka *view comment*.

3. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostCondition

1. Aktor melihat *comment* yang diinginkan.

4.1.9 Use case Spesification: Memposting Comment

1. Brief Description

Use Case ini digunakan oleh aktor untuk *memposting comment*.

2. Primary Actor

1. Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk *memposting comment*.
2. Sistem menampilkan halaman antarmuka untuk *memposting comment*.
3. Aktor menginputkan *comment*.
4. Sistem menambahkan *comment* ke dalam database dan menampilkan antarmuka yang menunjukkan bahwa *posting comment* telah berhasil.
5. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostCondition

1. Data *comment* dalam database telah *ter-update*.
2. Sistem menampilkan *comment* pada halaman antarmuka sistem.

4.1.10 Use case Spesification: Melihat Lokasi

1. Brief Description

Program Studi Teknik Informatika	SKPL – LAPAK	25/ 32
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 melihat lokasi.

2. Primary Actor

1. Member

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melihat lokasi.
2. Sistem menampilkan halaman antarmuka untuk melihat lokasi.
3. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostCondition

1. Sistem menampilkan lokasi yang diinginkan aktor.

4.1.11 Use case Spesification: Mengelola Thread

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola *thread*. Aktor dapat melakukan *view*, *add*, dan *delete*.

2. Primary Actor

1. Admin

3. Supporting Actor

none

4. Basic Flow

Program Studi Teknik Informatika	SKPL – LAPAK	26/ 32
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 ini dimulai ketika aktor memilih untuk melakukan pengelolaan *thread*.
2. Sistem memberikan pilihan untuk melakukan *view*, *create*, *edit*, dan *delete*.
3. Aktor memilih untuk melakukan *view*.
 - A-1 Aktor memilih untuk melakukan *create*.
 - A-2 Aktor memilih untuk melakukan *delete*.
4. Sistem menampilkan data *thread*.
5. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan *create*.

1. Sistem menampilkan halaman antarmuka untuk menambah *thread*.
2. Aktor menginputkan data *thread*.
3. Aktor meminta sistem untuk menyimpan data yang telah diinputkan.
4. Sistem melakukan pengecekan terhadap data yang telah diinputkan.
 - E-1 Data *thread* yang diinputkan tidak valid.
5. Sistem menambahkan *thread* ke dalam database dan menampilkan antarmuka yang menunjukkan bahwa penambahan *thread* telah berhasil.
6. Berlanjut ke Basic Flow langkah ke-5.

A-2 Aktor memilih untuk melakukan *delete*.

1. Sistem menampilkan *thread* yang akan dihapus.
2. Aktor meminta sistem untuk menghapus *thread*.
3. Sistem mengupdate database dan menampilkan antarmuka yang menunjukkan bahwa penghapusan *thread* telah berhasil.
4. Berlanjut ke Basic Flow langkah ke-5.

Program Studi Teknik Informatika	SKPL – LAPAK	27/ 32
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. Error Flow

E-1 Data *thread* yang telah diinputkan tidak valid.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan tidak valid.
2. Kembali ke Alternative Flow Langkah ke-2.

7. PreConditions

1. Use Case Login telah dilakukan.
3. Aktor telah memasuki sistem.

8. PostCondition

1. Data *thread* di database telah *ter-update*.

4.1.12 Use case Spesification: Mengelola Kategori Barang

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola kategori barang. Aktor dapat melakukan *view*, *create*, dan *delete*.

2. Primary Actor

1. Admin

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan kategori barang.
2. Sistem memberikan pilihan untuk melakukan *view*, *create*, dan *delete*.
3. Aktor memilih untuk melakukan *view*.
A-1 Aktor memilih untuk melakukan *create*.
A-2 Aktor memilih untuk melakukan *delete*.
4. Sistem meng-*update* database kategori barang.
5. Use Case selesai.

5. Alternative Flow

Program Studi Teknik Informatika	SKPL – LAPAK	28/ 32
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 *create*.

1. Sistem menampilkan halaman antarmuka untuk menambah kategori barang.
2. Aktor menginputkan kategori barang.
3. Aktor meminta sistem untuk menyimpan data yang telah diinputkan.
4. Sistem melakukan pengecekan terhadap data yang telah diinputkan.

E-1 Data kategori yang diinputkan tidak valid.

5. Sistem menambahkan kategori ke dalam database dan menampilkan antarmuka yang menunjukkan bahwa penambahan kategori telah berhasil.
6. Berlanjut ke Basic Flow langkah ke-5.

A-2 Aktor memilih untuk melakukan *delete*.

1. Sistem menampilkan kategori barang yang akan dihapus.
2. Aktor meminta sistem untuk menghapus kategori barang.
3. Sistem mengupdate database dan menampilkan antarmuka yang menunjukkan bahwa penghapusan kategori barang telah berhasil.
4. Berlanjut ke Basic Flow langkah ke-5.

6. Error Flow

E-1 Data kategori yang telah diinputkan tidak valid.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan tidak valid.
2. Kembali ke Alternative Flow 1 Langkah ke-2.

7. PreConditions

Program Studi Teknik Informatika	SKPL – LAPAK	29/ 32
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.
2. Aktor telah memasuki sistem.

8. PostCondition

1. Data kategori barang di database telah *ter-update*.

4.1.13 Use case Spesification: Mengelola Member

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola member. Aktor dapat melakukan *view* dan *delete*.

2. Primary Actor

1. Admin

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan kelola member.
2. Sistem akan menampilkan data member dan memberikan pilihan melakukan *delete*.
3. Aktor memilih untuk melihat data member.
A-1 Aktor memilih untuk melakukan *delete*.
4. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan *delete*.

1. Sistem menampilkan data member yang akan dihapus.
2. Aktor meminta sistem untuk menghapus data member.

3. Sistem mengupdate database dan menampilkan antarmuka yang menunjukkan bahwa penghapusan member telah berhasil.

4. Berlanjut ke Basic Flow langkah ke-4.

6. Error Flow

none

7. PreConditions

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

8. PostCondition

1. Data member di database telah ter-update.

4.1.14 Use case Spesification: Mengelola Comment

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola *comment*. Aktor dapat melakukan *view* dan *delete*.

2. Primary Actor

1. Admin

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan kelola *comment*.
2. Sistem akan menampilkan data *comment* dan memberikan pilihan melakukan *delete*.
3. Aktor memilih untuk melihat *comment*.
A-1 Aktor memilih untuk melakukan *delete*.
4. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan *delete*.

Program Studi Teknik Informatika	SKPL – LAPAK	31/ 32
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 *comment* yang akan dihapus.
2. Aktor meminta sistem untuk menghapus *comment*.
3. Sistem mengupdate database dan menampilkan antarmuka yang menunjukkan bahwa penghapusan *comment* telah berhasil.
4. Berlanjut ke Basic Flow langkah ke-4.

6. Error Flow

none

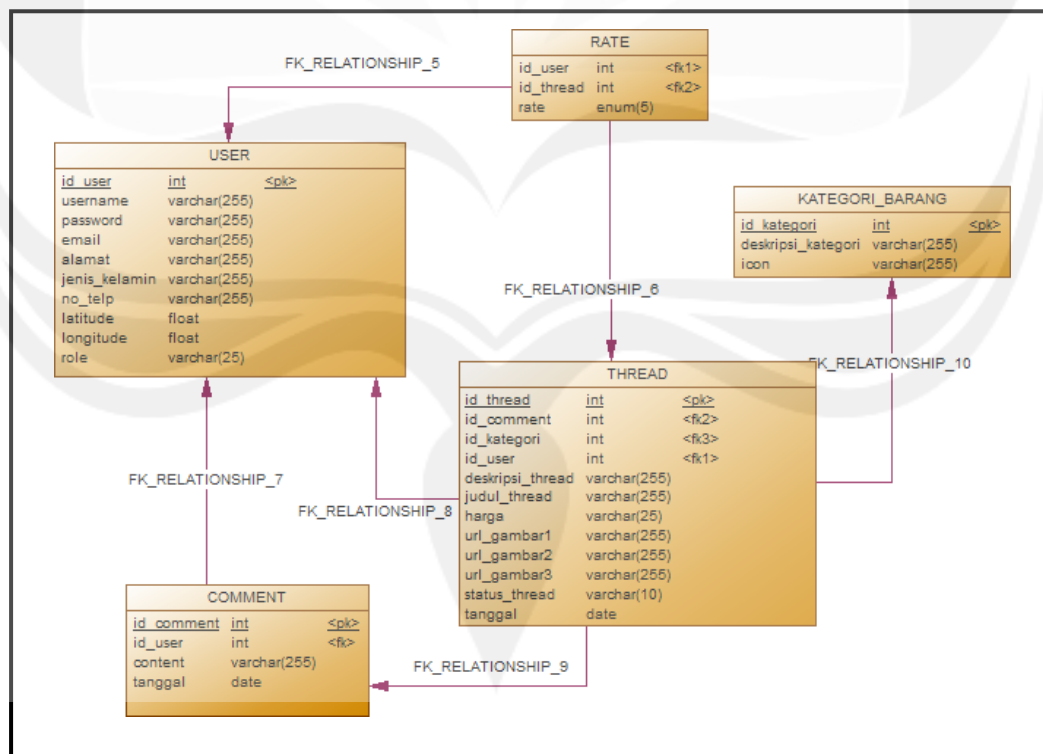
7. PreConditions

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

8. PostCondition

1. Data member di database telah ter-update.

5 Entity Relationship Diagram (ERD)



Gambar 5. Entity Relationship Diagram

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

LAPAK


(PEMBANGUNAN FORUM JUAL BELI DENGAN
MENGUNAKAN *MOBILE-WEB* BERBASIS LOKASI)

Dipersiapkan oleh:

Yohanes Adven Stefanus / 08 07 05507

Program Studi Teknik Informatika - Fakultas Teknologi
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL-LAPAK		1/38
		Revisi		

Program Studi Teknik Informatika	DPPL-SIMPONI UAJY	1/38
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		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G

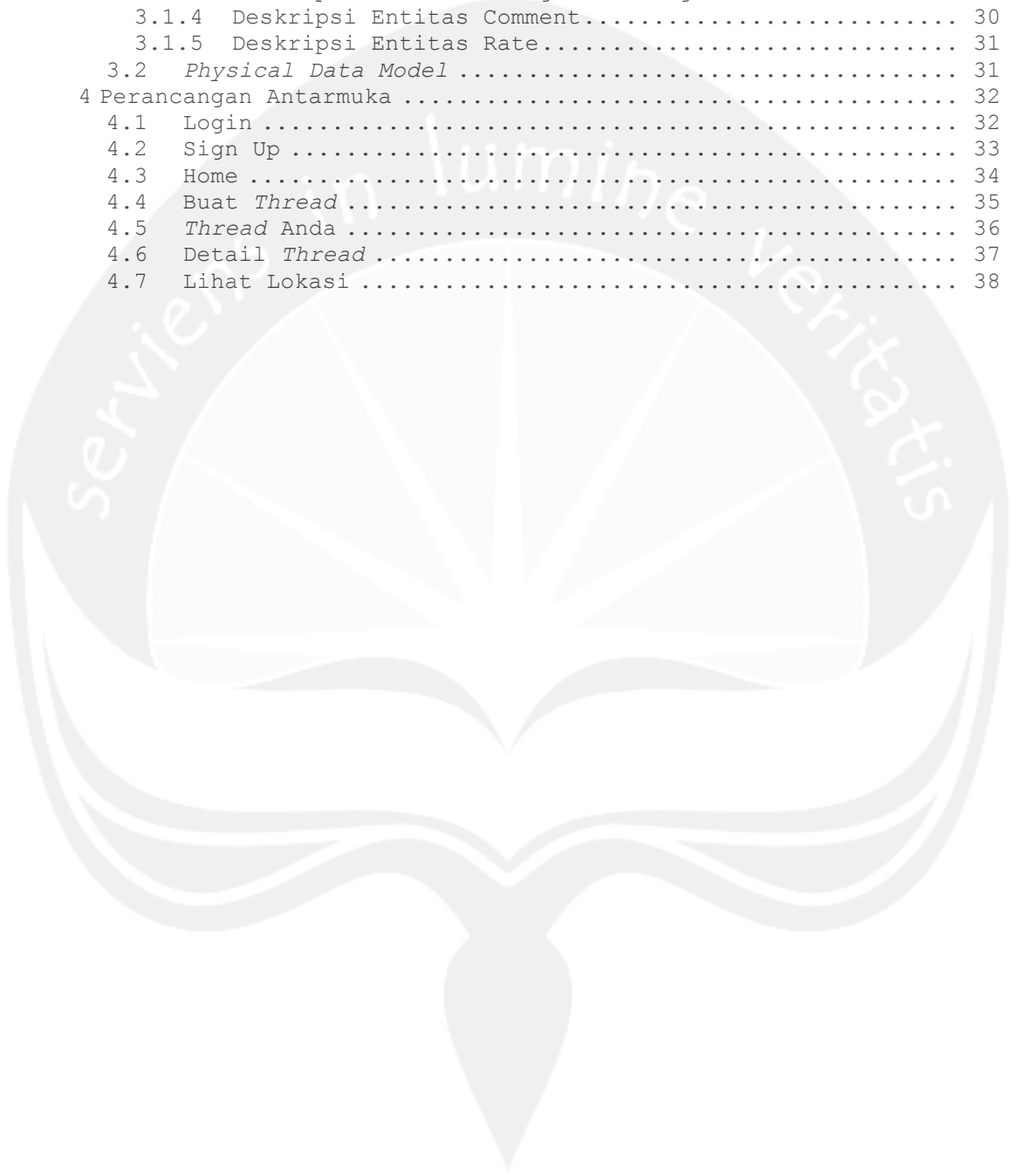
Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan	6
1.1	Tujuan	6
1.2	Ruang Lingkup	7
1.3	Definisi dan Akronim	7
1.4	Referensi	8
2	Perancangan Sistem	9
2.1	Perancangan Arsiteksur	9
2.2	Perancangan Rinci	10
2.2.1	Sequence Diagram.....	10
2.2.1.1	Login	10
2.2.1.2	Daftar	10
2.2.1.3	View Thread	11
2.2.1.4	Create Thread	11
2.2.1.5	Edit Thread	12
2.2.1.6	Ubah Status Thread	12
2.2.1.7	Search Thread	13
2.2.1.8	View Comment	13
2.2.1.9	Post Comment	14
2.2.1.10	Mengelola User	15
2.2.1.11	Mengelola Kategori	16
2.2.1.12	Mengelola Comment	17
2.2.1.13	Mengelola Thread	18
2.2.2	Class Diagram.....	19
2.2.3	Class Diagram Specific Descriptions.....	20
2.2.3.1	Specific Design Class admin_view	20
2.2.3.3	Specific Design Class kelolaKategori_view	20
2.2.3.4	Specific Design Class kelolaThread_view	20
2.2.3.5	Specific Design Class kelolaComment_view	20
2.2.3.6	Specific Design Class kelolaUser_view	20
2.2.3.7	Specific Design Class landing_view	20
2.2.3.8	Specific Design Class showThread_view	21
2.2.3.9	Specific Design Class detailThread_view	21
2.2.3.10	Specific Design Class resultSearch_view	21
2.2.3.11	Specific Design Class login_view	21
2.2.3.12	Specific Design Class buatThread_view	21
2.2.3.13	Specific Design Class editThread_view	21
2.2.3.14	Specific Design Class threadAnda_view	22
2.2.3.15	Specific Design Class signup_view	22
2.2.3.16	Specific Design Class lihatLokasi_view	22
2.2.3.17	Specific Design Class admin_control	22
2.2.3.18	Specific Design Class landing_control	23
2.2.3.19	Specific Design Class login_control	24
2.2.3.20	Specific Design Class buatThread_control	24
2.2.3.21	Specific Design Class threadAnda_control	25
2.2.3.22	Specific Design Class signup_control	25
2.2.3.23	Specific Design Class lihatLokasi_control	26
2.2.3.24	Specific Design Class admin_model	26
2.2.3.25	Specific Design Class landing_model	27
2.2.3.26	Specific Design Class login_model	27
2.2.3.27	Specific Design Class buat_thread_model	28
2.2.3.28	Specific Design Class thread_anda_model	29
2.2.3.29	Specific Design Class signup_model	29

3	Perancangan Data	30
3.1	Dekomposisi Data	30
3.1.1	Deskripsi Entitas User.....	30
3.1.2	Deskripsi Entitas Thread.....	30
3.1.3	Deskripsi Entitas Kategori Barang.....	30
3.1.4	Deskripsi Entitas Comment.....	30
3.1.5	Deskripsi Entitas Rate.....	31
3.2	<i>Physical Data Model</i>	31
4	Perancangan Antarmuka	32
4.1	Login	32
4.2	Sign Up	33
4.3	Home	34
4.4	Buat <i>Thread</i>	35
4.5	<i>Thread</i> Anda	36
4.6	Detail <i>Thread</i>	37
4.7	Lihat Lokasi	38



Daftar Gambar

Gambar 2.1. Perancangan Arsitektur.....	9
Gambar 2.2.1.1. Sequence Diagram : Login.....	10
Gambar 2.2.1.2. Sequence Diagram : <i>Datfar</i>	10
Gambar 2.2.1.3. Sequence Diagram : <i>View Thread</i>	11
Gambar 2.2.1.4. Sequence Diagram : <i>Create Thread</i>	11
Gambar 2.2.1.5. Sequence Diagram : <i>Edit Thread</i>	12
Gambar 2.2.1.6. Sequence Diagram : <i>Ubah Status Thread</i>	12
Gambar 2.2.1.7. Sequence Diagram : <i>Search Thread</i>	13
Gambar 2.2.1.8. Sequence Diagram : <i>View Comment</i>	13
Gambar 2.2.1.9. Sequence Diagram : <i>Post Comment</i>	14
Gambar 2.2.1.10.1. Sequence Diagram : <i>View User</i>	15
Gambar 2.2.1.10.2. Sequence Diagram : <i>Delete User</i>	15
Gambar 2.2.1.11.1. Sequence Diagram : <i>View Kategori</i>	16
Gambar 2.2.1.11.2. Sequence Diagram : <i>Add Kategori</i>	16
Gambar 2.2.1.12.1. Sequence Diagram : <i>View Comment</i>	17
Gambar 2.2.1.12.2. Sequence Diagram : <i>Delete Comment</i>	17
Gambar 2.2.1.13.1. Sequence Diagram : <i>View Thread</i>	18
Gambar 2.2.1.13.2. Sequence Diagram : <i>Delete Thread</i>	18
Gambar 2.2.2. Class Diagram.....	19
Gambar 3.2. Physical Data Model.....	31
Gambar 4.1. Rancangan Antarmuka Login.....	32
Gambar 4.2. Rancangan Sign Up.....	33
Gambar 4.3. Rancangan Antarmuka Home.....	34
Gambar 4.4. Rancangan Antarmuka <i>Buat Thread</i>	35
Gambar 4.5. Rancangan Antarmuka <i>Thread Anda</i>	36
Gambar 4.6. Rancangan Antarmuka <i>Thread Anda</i>	37
Gambar 4.7. Rancangan Antarmuka <i>Lihat Lokasi</i>	38

1 Pendahuluan

1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL

Program Studi Teknik Informatika	DPPL-LAPAK	6/ 38
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		

tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

1.2 Ruang Lingkup

Perangkat Lunak LAPAK dikembangkan dengan tujuan untuk :

1. Menangani login pengguna.
2. Menangani pendaftaran member.
3. Menangani *view thread*.
4. Menangani *create thread*.
5. Menangani *edit thread*.
6. Menangani ubah status *thread*.
7. Menangani cari *thread*.
8. Menangani *view comment*.
9. Menangani *post comment*.
10. Menangani lihat lokasi.
11. Menangani pengelolaan *thread*.
12. Menangani pengelolaan kategori barang.
13. Menangani pengelolaan user.
14. Menangani pengelolaan *comment*.

Dan berjalan pada perangkat komputer dan perangkat *mobile* (PC, Laptop, tablet-PC, *handphone*, dll) yang menggunakan sistem operasi Windows XP, windows 7 untuk desktop-user dan segala sistem operasi untuk web-user dan segala perangkat *mobile* yang mendukung *JQuery Mobile*.

1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak

Program Studi Teknik Informatika	DPPL-LAPAK	7/38
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		

	disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
LAPAK	Perangkat lunak untuk membuat <i>thread</i> pada <i>mobile-web</i> .
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented System Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
2. Adven Stefanus, Yohanes, *Spesifikasi Kebutuhan Perangkat Lunak LAPAK*, Universitas Atma Jaya Yogyakarta, 2012.

2 Perancangan Sistem

2.1 Perancangan Arsitektur

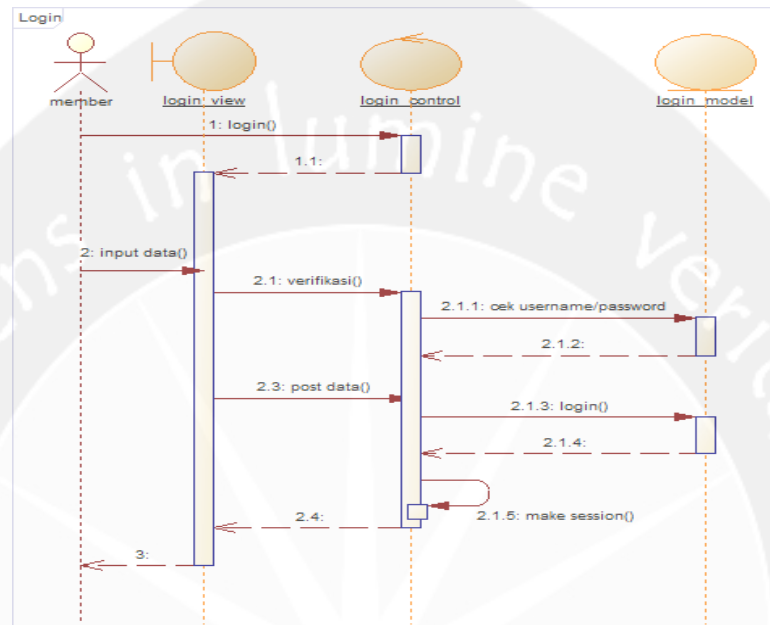


Gambar 2.1. Perancangan Arsitektur

2.2 Perancangan Rinci

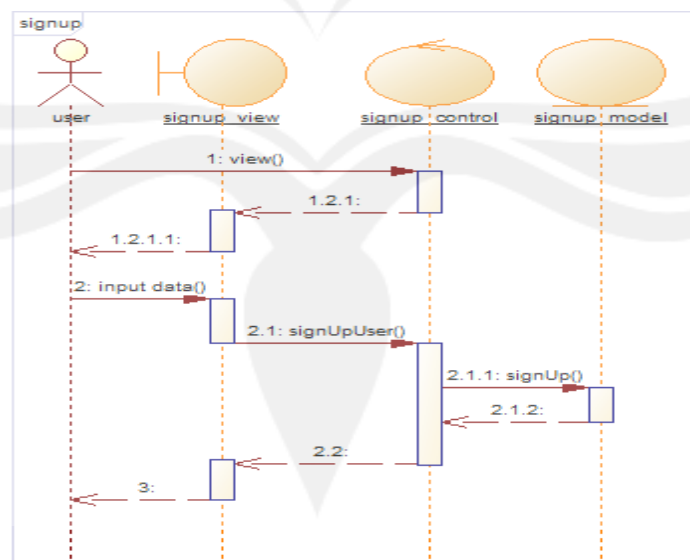
2.2.1 Sequence Diagram

2.2.1.1 Login



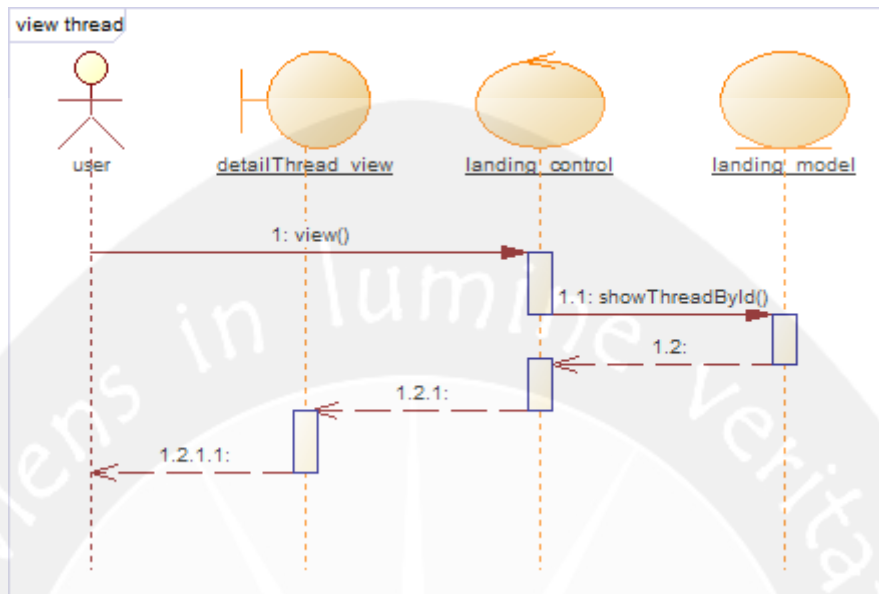
Gambar 2.2.1.1. Sequence Diagram : Login

2.2.1.2 Daftar



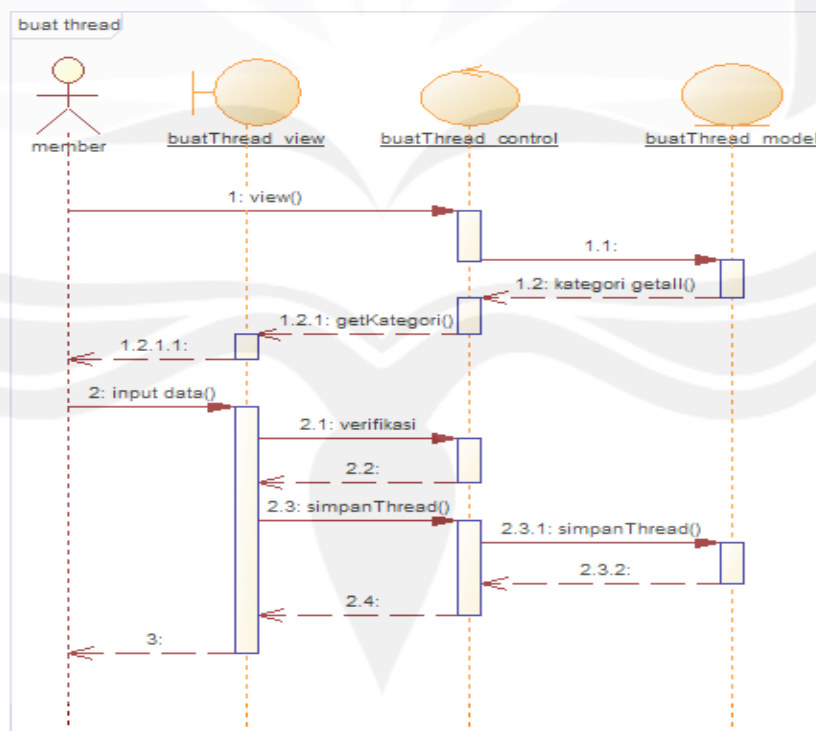
Gambar 2.2.1.2. Sequence Diagram : Daftar

2.2.1.3 View Thread



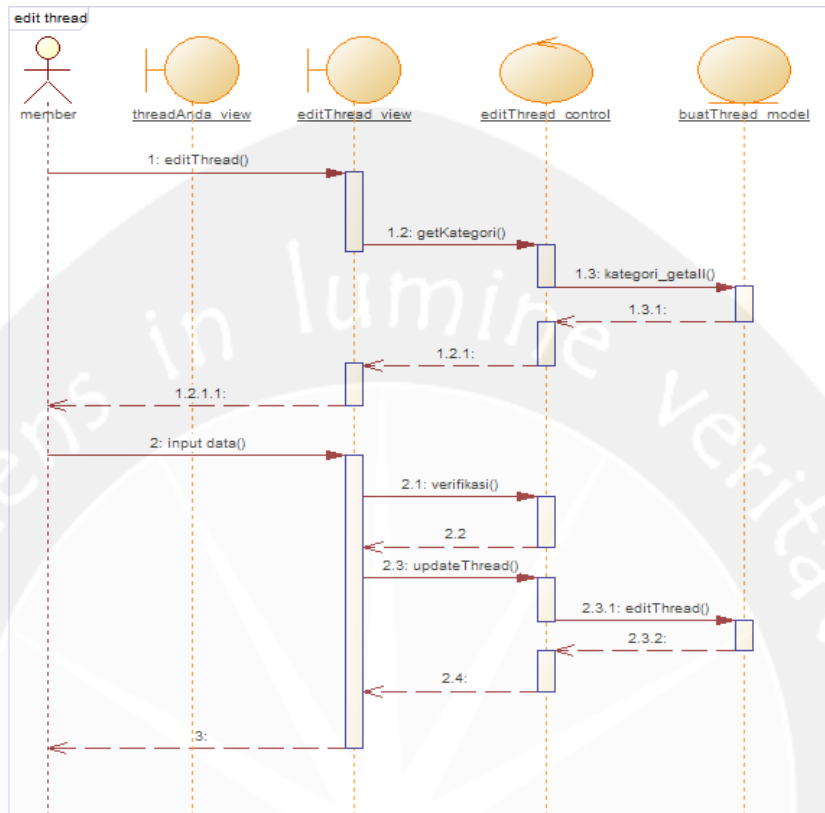
Gambar 2.2.1.3. Sequence Diagram : View Thread

2.2.1.4 Create Thread



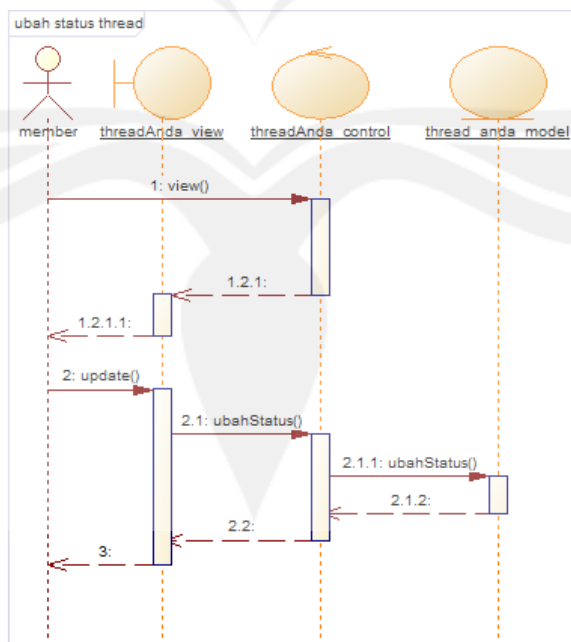
Gambar 2.2.1.4. Sequence Diagram : Create Thread

2.2.1.5 *Edit Thread*



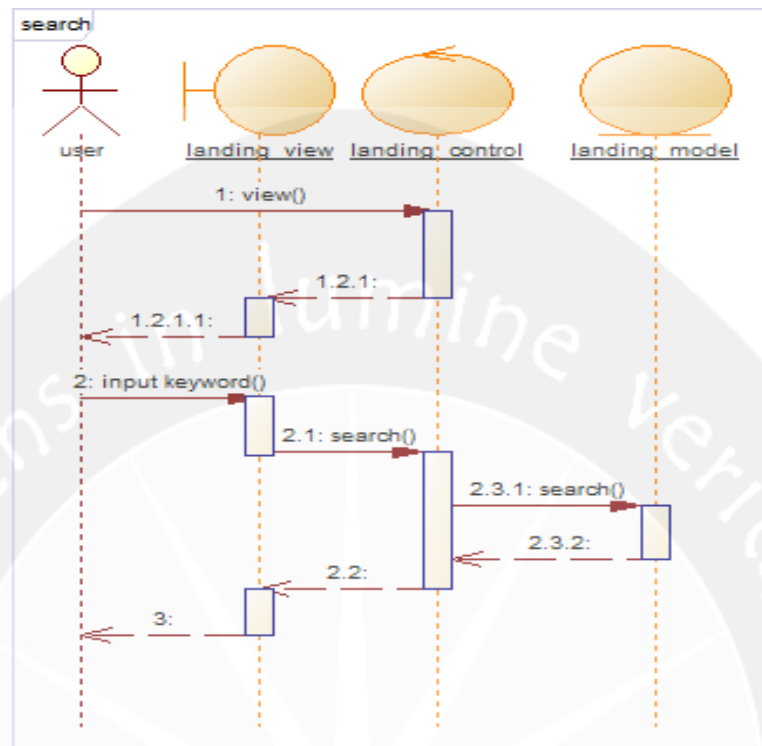
Gambar 2.2.1.5. Sequence Diagram : *Edit Thread*

2.2.1.6 *Ubah Status Thread*



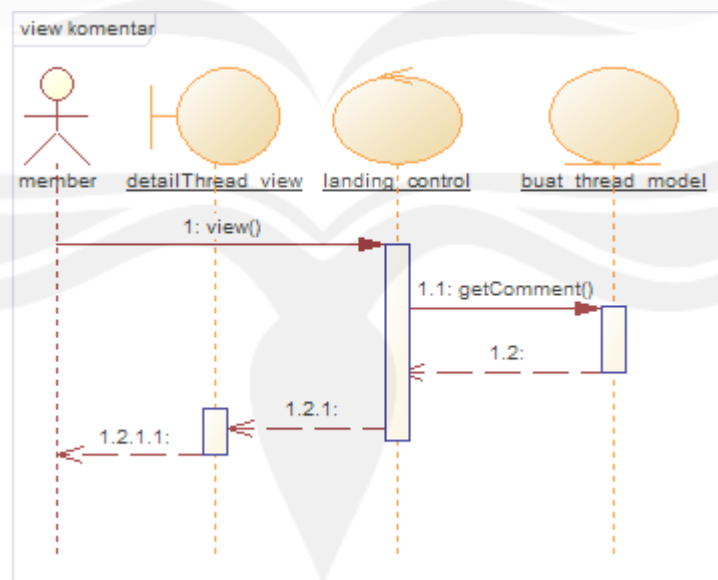
Gambar 2.2.1.6. Sequence Diagram : *Ubah Status Thread*

2.2.1.7 Search Thread



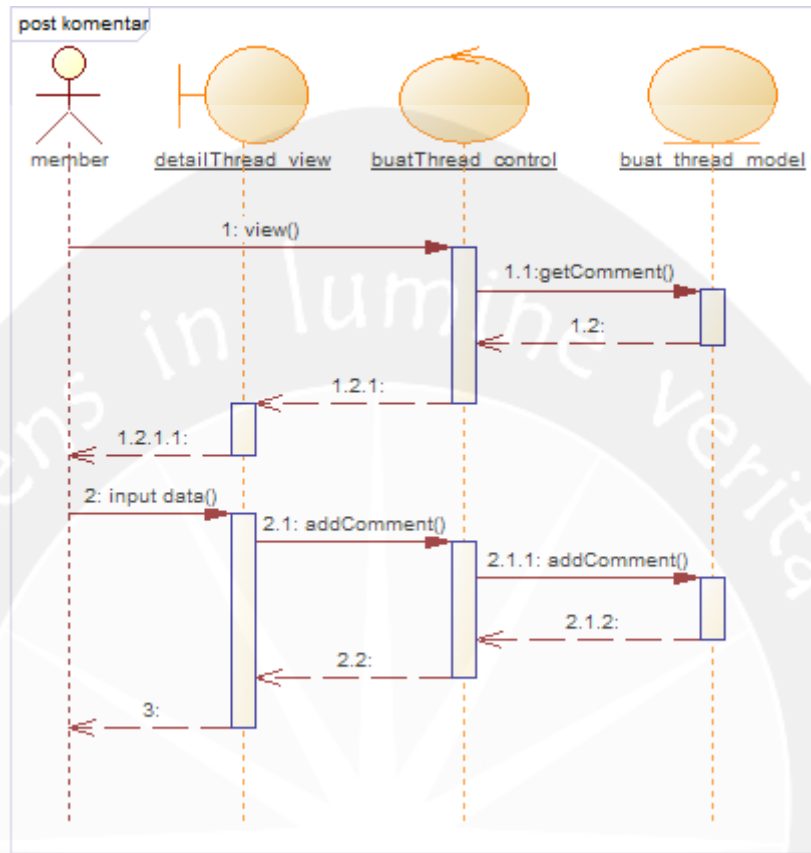
Gambar 2.2.1.7. Sequence Diagram : *Search Thread*

2.2.1.8 View Comment



Gambar 2.2.1.8. Sequence Diagram : *View Comment*

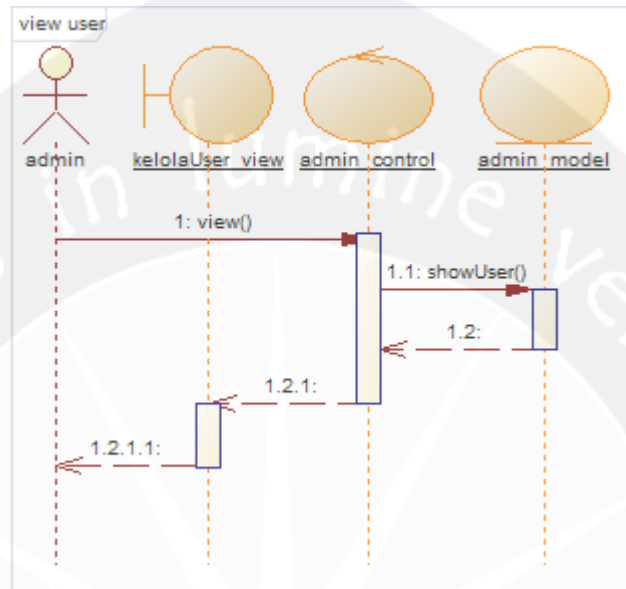
2.2.1.9 Post Comment



Gambar 2.2.1.9. Sequence Diagram : Post Comment

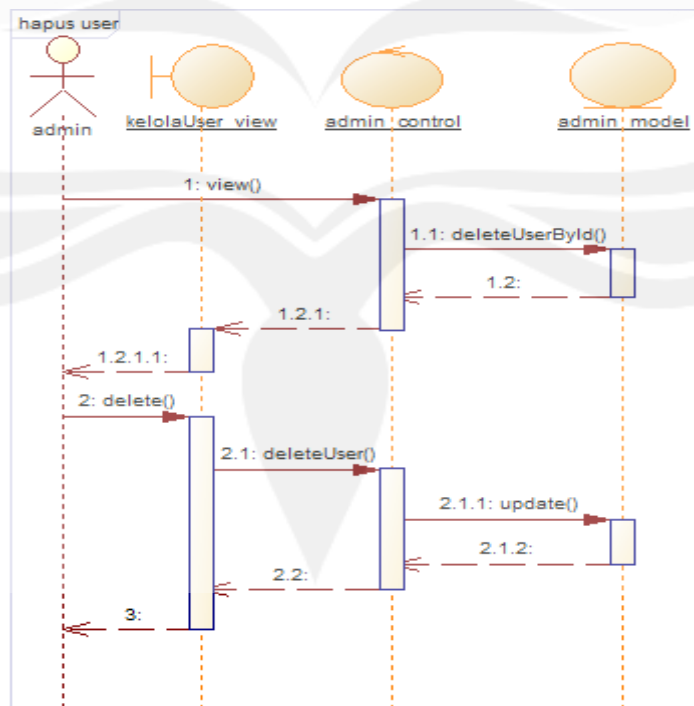
2.2.1.10 Mengelola User

2.2.1.10.1 View User



Gambar 2.2.1.10.1. Sequence Diagram : View User

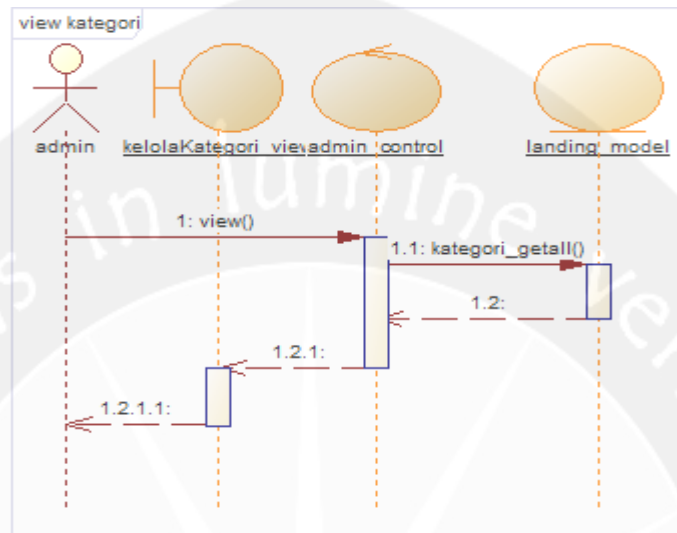
2.2.1.10.2 Delete User



Gambar 2.2.1.10.2. Sequence Diagram : Delete User

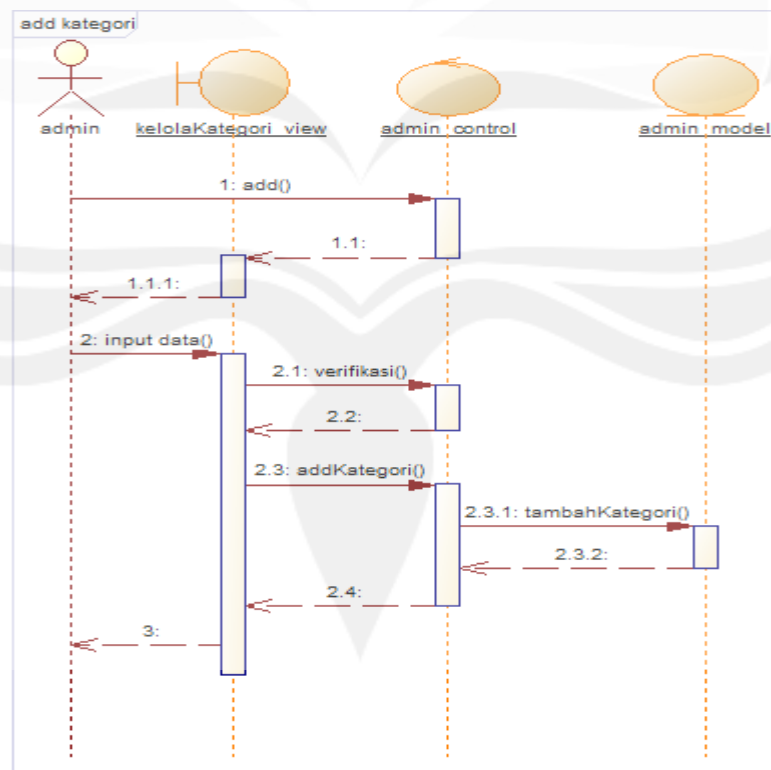
2.2.1.11 Mengelola Kategori

2.2.1.11.1 View Kategori



Gambar 2.2.1.11.1. Sequence Diagram : View Kategori

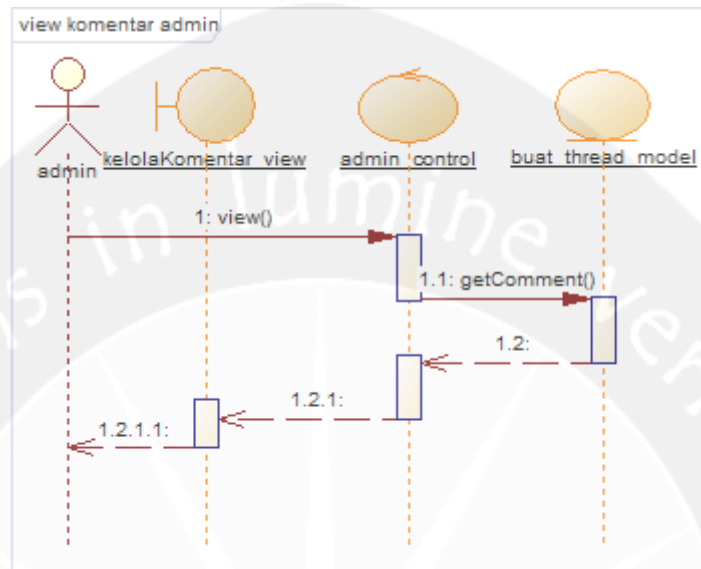
2.2.1.11.2 Add Kategori



Gambar 2.2.1.11.2. Sequence Diagram : Add Kategori

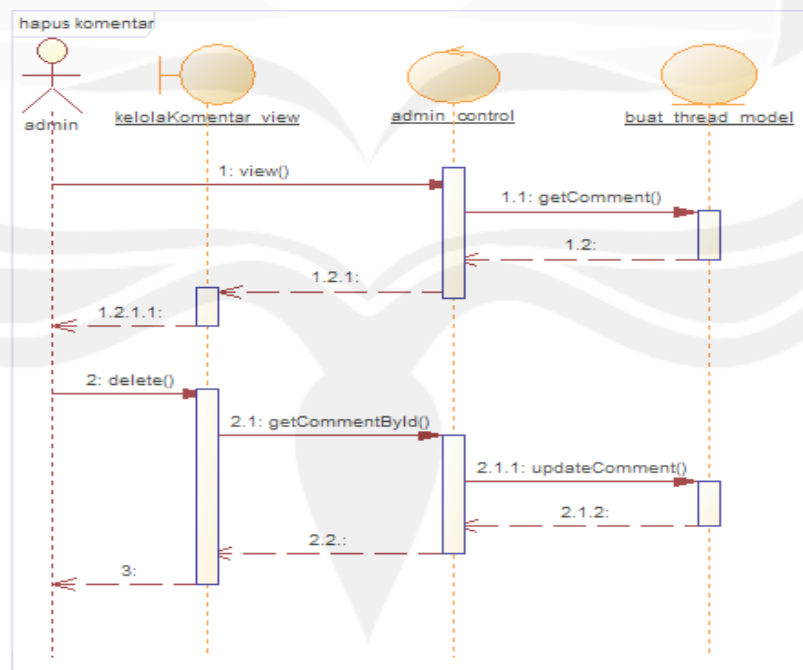
2.2.1.12 Mengelola Comment

2.2.1.12.1 View Comment



Gambar 2.2.1.12.1. Sequence Diagram : View Comment

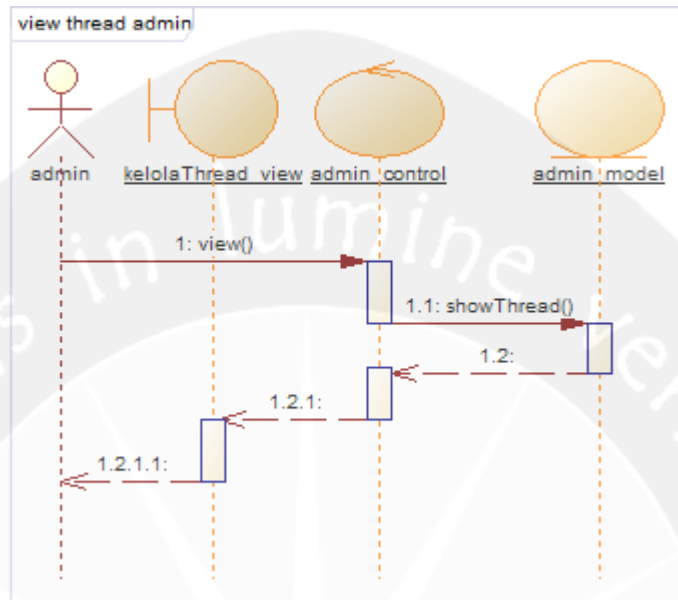
2.2.1.12.2 Delete Comment



Gambar 2.2.1.12.2. Sequence Diagram : Delete Comment

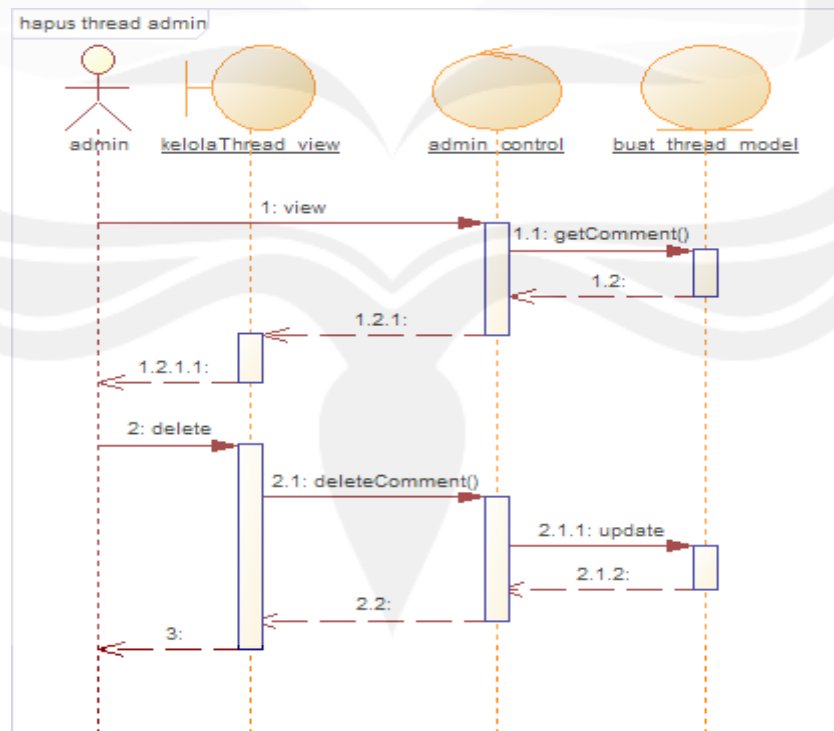
2.2.1.13 Mengelola Thread

2.2.1.13.1 View Thread



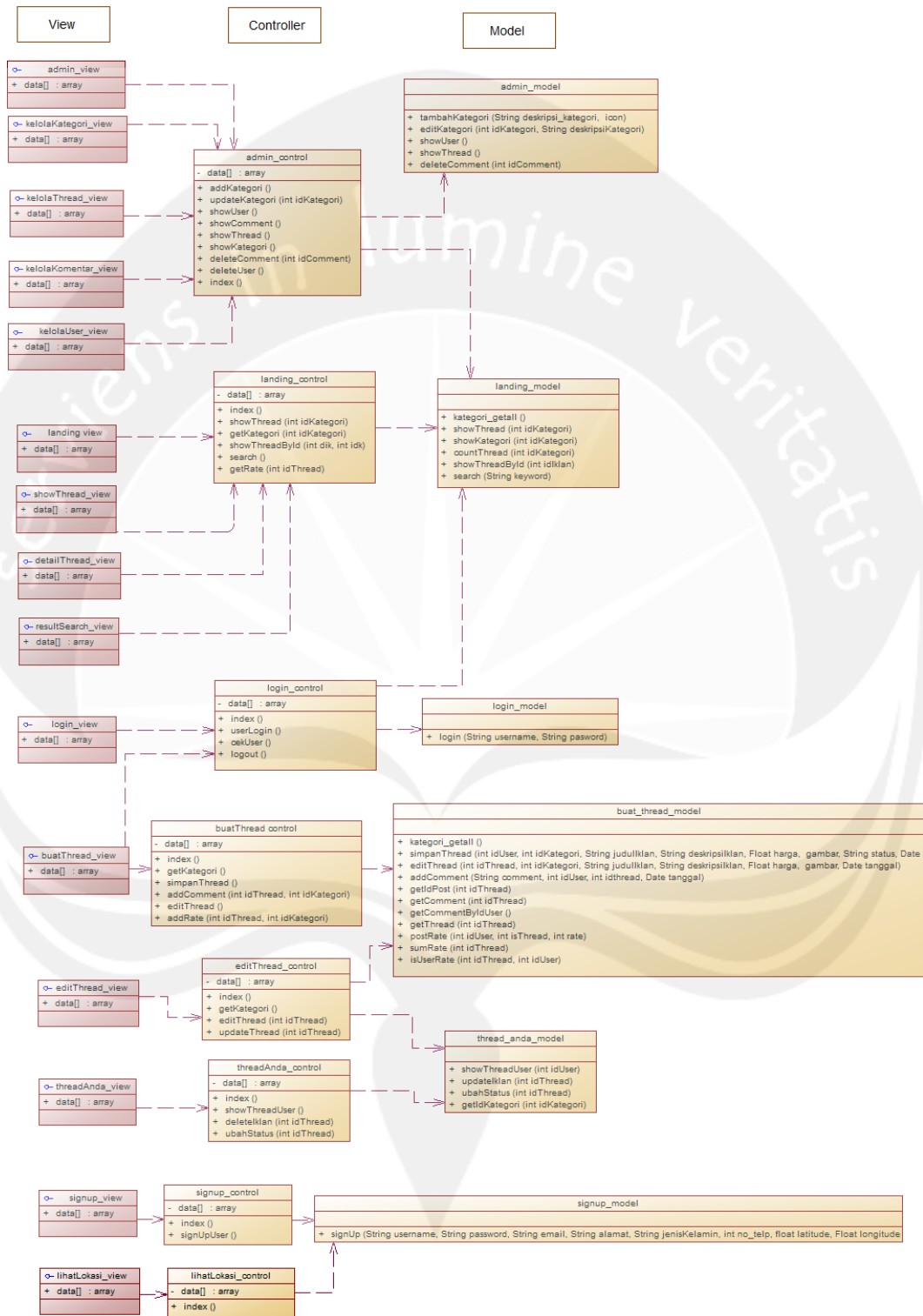
Gambar 2.2.1.13.1. Sequence Diagram : View Thread

2.2.1.13.2 Delete Thread



Gambar 2.2.1.13.2. Sequence Diagram : Delete Thread

2.2.2 Class Diagram



Gambar 2.2.2. Class Diagram

2.2.3 Class Diagram Specific Descriptions

2.2.3.1 Specific Design Class `admin_view`

<code>admin_view</code>	<code><<boundary>></code>
<code>data[]: array</code> Atribut ini digunakan untuk menampung data dari <i>control</i> untuk ditampilkan kepada pengguna	

2.2.3.3 Specific Design Class `kelolaKategori_view`

<code>kelolaKategori_view</code>	<code><<boundary>></code>
<code>data[]: array</code> Atribut ini digunakan untuk menampung data dari <i>control</i> untuk ditampilkan kepada pengguna	

2.2.3.4 Specific Design Class `kelolaThread_view`

<code>kelolaThread_view</code>	<code><<boundary>></code>
<code>data[]: array</code> Atribut ini digunakan untuk menampung data dari <i>control</i> untuk ditampilkan kepada pengguna	

2.2.3.5 Specific Design Class `kelolaComment_view`

<code>kelolaComment_view</code>	<code><<boundary>></code>
<code>data[]: array</code> Atribut ini digunakan untuk menampung data dari <i>control</i> untuk ditampilkan kepada pengguna	

2.2.3.6 Specific Design Class `kelolaUser_view`

<code>kelolaUser_view</code>	<code><<boundary>></code>
<code>data[]: array</code> Atribut ini digunakan untuk menampung data dari <i>control</i> untuk ditampilkan kepada pengguna	

2.2.3.7 Specific Design Class `landing_view`

<code>landing_view</code>	<code><<boundary>></code>
<code>data[]: array</code>	

Atribut ini digunakan untuk menampung data dari *control* untuk ditampilkan kepada pengguna

2.2.3.8 Specific Design Class showThread_view

showThread_view	<<boundary>>
data[]: array Atribut ini digunakan untuk menampung data dari <i>control</i> untuk ditampilkan kepada pengguna	

2.2.3.9 Specific Design Class detailThread_view

detailThread_view	<<boundary>>
data[]: array Atribut ini digunakan untuk menampung data dari <i>control</i> untuk ditampilkan kepada pengguna	

2.2.3.10 Specific Design Class resultSearch_view

resultSearch_view	<<boundary>>
data[]: array Atribut ini digunakan untuk menampung data dari <i>control</i> untuk ditampilkan kepada pengguna	

2.2.3.11 Specific Design Class login_view

Login_view	<<boundary>>
data[]: array Atribut ini digunakan untuk menampung data dari <i>control</i> untuk ditampilkan kepada pengguna	

2.2.3.12 Specific Design Class buatThread_view

buatThread_view	<<boundary>>
data[]: array Atribut ini digunakan untuk menampung data dari <i>control</i> untuk ditampilkan kepada pengguna	

2.2.3.13 Specific Design Class editThread_view

editThread_view	<<boundary>>
------------------------	---------------------------------

data[]: array

Atribut ini digunakan untuk menampung data dari *control* untuk ditampilkan kepada pengguna

2.2.3.14 Specific Design Class threadAnda_view

threadAnda_view

<<boundary>>

data[]: array

Atribut ini digunakan untuk menampung data dari *control* untuk ditampilkan kepada pengguna

2.2.3.15 Specific Design Class signup_view

Signup_view

<<boundary>>

data[]: array

Atribut ini digunakan untuk menampung data dari *control* untuk ditampilkan kepada pengguna

2.2.3.16 Specific Design Class lihatLokasi_view

lihatLokasi_view

<<boundary>>

data[]: array

Atribut ini digunakan untuk menampung data dari *control* untuk ditampilkan kepada pengguna

2.2.3.17 Specific Design Class admin_control

Admin_control

<<control>>

data[]: array

Atribut ini digunakan untuk menampung data dari *model* untuk dilempar di *view*

index()

Fungsi ini digunakan untuk menampilkan halaman admin

addKategori()

Fungsi ini digunakan untuk menambah kategori barang

updateKategori(int idKategori)

Fungsi ini digunakan untuk mengedit kategori barang

showUser ()

Fungsi ini digunakan untuk menampilkan list user

showComment ()

Fungsi ini digunakan untuk menampilkan list *comment*

showThread ()

Fungsi ini digunakan untuk menampilkan list *thread*

showKategori ()

Fungsi ini digunakan untuk menampilkan list kategori barang

deleteComment(int idComment)

Fungsi ini digunakan untuk menghapus *comment* user

deleteUser ()

Fungsi ini digunakan untuk menghapus user

2.2.3.18 Specific Design Class landing_control**landing_control**

<<control>>

data[]: array

Atribut ini digunakan untuk menampung data dari *model* untuk dilempar di *view*

Index ()

Fungsi ini digunakan untuk menampilkan list kategori barang

showThread(int idKategori)

Fungsi ini digunakan untuk menampilkan list *thread* berdasar kategori barang

getKategori(int idKategori)

Fungsi ini digunakan untuk menampilkan deskripsi kategori

showThreadById(int dik,int idk)

Fungsi ini digunakan untuk menampilkan detail *thread* berdasarkan berdasarkan id *thread*

search ()

Fungsi ini digunakan untuk mencari *thread*

getRate ()

Fungsi ini digunakan untuk menampung nilai rating

2.2.3.19 Specific Design Class login_control

login_control	<<control>>
data[]: array	
Atribut ini digunakan untuk menampung data dari <i>model</i> untuk dilempar di <i>view</i>	
index ()	
Fungsi ini digunakan untuk menampilkan halaman antarmuka login	
userLogin ()	
Fungsi ini digunakan untuk mencocokkan username dan password yang diinputkan user dengan data yang ada pada database	
cekUser ()	
Fungsi ini digunakan untuk mengecek apakah user sudah login atau belum	
logout ()	
Fungsi ini digunakan untuk menghapus session dan cookie	

2.2.3.20 Specific Design Class buatThread_control

buatThread_control	<<control>>
data[]: array	
Atribut ini digunakan untuk menampung data dari <i>model</i> untuk dilempar di <i>view</i>	
index ()	
Fungsi ini digunakan untuk menampilkan list kategori barang	
getKategori ()	

Fungsi ini digunakan untuk menampung id dan deskripsi kategori barang yang diambil dari *database*

simpanThread()

Fungsi ini digunakan untuk menyimpan *thread*

addComment(int idThread, int idKategori)

Fungsi ini digunakan untuk menambahkan *comment* pada *thread*

editThread()

Fungsi ini digunakan untuk menampung data sebelum di-*update*

addRate(int idThread, int idKategori)

Fungsi ini digunakan untuk menampung rating dari user

2.2.3.21 Specific Design Class threadAnda_control

threadAnda_control	<<control>>
data[]: array	
Atribut ini digunakan untuk menampung data dari <i>model</i> untuk dilempar di <i>view</i>	
index()	
Fungsi ini digunakan untuk menampilkan halaman antarmuka <i>threadAnda</i>	
showThreadUser()	
Fungsi ini digunakan untuk menampilkan <i>thread</i> yang dimiliki oleh member	
deleteIklan()	
Fungsi ini digunakan untuk menghapus <i>thread</i> member	
ubahStatus()	
Fungsi ini digunakan untuk merubah status <i>thread</i>	

2.2.3.22 Specific Design Class signup_control

Forum	<<control>>
data[]: array	

Atribut ini digunakan untuk menampung data dari *model* untuk dilempar di *view*

index()

Fungsi ini digunakan untuk menampilkan halaman antarmuka *signup*

signUpUser()

Fungsi ini digunakan untuk menyimpan data dari *user*

validation()

Fungsi ini digunakan untuk mengecek apakah data yang diinputkan valid atau tidak

2.2.3.23 Specific Design Class lihatLokasi_control

lihatLokasi_control	<<control>>
----------------------------	--------------------------------

data[]: array

Atribut ini digunakan untuk menampung data dari *model* untuk dilempar di *view*

index()

Fungsi ini digunakan untuk menampilkan halaman antarmuka lihat lokasi

2.2.3.24 Specific Design Class admin_model

admin_model	<<entity>>
--------------------	-------------------------------

tambahKategori(String deskripsi_kategori, icon)

Fungsi ini digunakan untuk menambahkan data kategori barang berdasarkan deskripsi kategori

editKategori(int idKategori, String deskripsiKategori)

Fungsi ini digunakan untuk mengubah data kategori barang di basis data

showUser()

Fungsi ini digunakan untuk mengambil data member dari basis data

showThread()

Fungsi ini digunakan untuk mengambil data thread dari basis data

deleteComment(int idComment)

Fungsi ini digunakan untuk menghapus *comment* dari basis data

2.2.3.25 Specific Design Class landing_model

landing_model	<<entity>>
kategori_getall()	
Fungsi ini digunakan untuk mengambil data kategori barang dari basis data	
showThread(int idKategori)	
Fungsi ini digunakan untuk mengambil data <i>thread</i> dari basis data berdasarkan id kategori	
showKategori(int idKategori)	
Fungsi ini digunakan untuk mengambil data kategori barang dari basis data berdasar id kategori	
countThread(int idKategori)	
Fungsi ini digunakan untuk mengambil beberapa data kategori barang dari basis data berdasar id kategori	
showThreadById(int idIklan)	
Fungsi ini digunakan untuk mengambil data <i>thread</i> dari basis data berdasar id thread	
search(String keyword)	
Fungsi ini digunakan untuk mengambil <i>thread</i> dari basis data berdasarkan parameter yang diminta	

2.2.3.26 Specific Design Class login_model

login_model	<<entity>>
login(String username, String password)	
Fungsi ini digunakan untuk mengecek data yang diinputkan user apakah sama dengan yang ada di basis	

data atau tidak

2.2.3.27 Specific Design Class `buat_thread_model`

<code>Buat_thread_model</code>	<code><<entity>></code>
<code>kategori_getall()</code> Fungsi ini digunakan untuk mengambil data kategori barang dari basis data	
<code>simpanThread(int idUser, int idKategori, String judulIklan, String deskripsiIklan, Float harga, gambar, String status, date tanggal)</code> Fungsi ini digunakan untuk menambah tulisan di basis data	
<code>editThread(int idThread, int idKategori, String judulIklan, String deskripsiIklan, Float harga, gambar, String status, date tanggal)</code> Fungsi ini digunakan untuk menambah tulisan di basis data	
<code>addComment(String comment, int idUser, int idThread, date tanggal)</code> Fungsi ini digunakan untuk menambahkan <i>comment</i> di basis data	
<code>getIdPost(int idThread)</code> Fungsi ini digunakan untuk mengambil id <i>thread</i> dari basis data	
<code>getComment(int idThread)</code> Fungsi ini digunakan untuk mengambil <i>comment</i> dari basis data berdasar id <i>thread</i>	
<code>getCommentByIdUser()</code> Fungsi ini digunakan untuk mengambil <i>comment</i> dari basis data	
<code>getThread(int idThread)</code>	

Fungsi ini digunakan untuk mengambil *thread* dari basis data berdasar id *thread*

postRate(int idUser, int idThread, int rate)

Fungsi ini digunakan untuk menambahkan rate pada basis data

sumRate(int idThread)

Fungsi ini digunakan untuk menghitung jumlah rating

isUserRate(int idThread, int idUser)

Fungsi ini digunakan untuk mengecek apakah user sudah memberi rating pada suatu *thread*

2.2.3.28 Specific Design Class *thread_anda_model*

Thread_anda_model	<<entity>>
showThreadUser(int idUser)	
Fungsi ini digunakan untuk mengambil data <i>thread</i> dari basis data berdasar id user	
updateIklan(int idThread)	
Fungsi ini digunakan untuk mengubah data <i>thread</i> di basis data berdasarkan id <i>thread</i>	
ubahStatus(int idThread)	
Fungsi ini digunakan untuk mengubah status <i>thread</i> di basis data berdasarkan id <i>thread</i>	
getIdKategori(int idKategori)	
Fungsi ini digunakan untuk mengambil id kategori dari basis data	

2.2.3.29 Specific Design Class *signup_model*

signup_model	<<entity>>
signUp(String username, String password, String email, String alamat, String jenisKelamin, int no_telp, Float latitude, Float longitude, String role)	
Fungsi ini digunakan untuk menambah user ke basis data	

3 Perancangan Data

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas User

Nama	Tipe	Panjang	Keterangan
ID_USER	Integer	11	Id user, <i>Primary key</i>
USERNAME	Varchar	255	Nama dari user
PASSWORD	Varchar	255	Password user
EMAIL	Varchar	255	Email user
ALAMAT	Varchar	255	Alamat user
JENIS_KELAMIN	Varchar	255	Jenis kelamin user
NO_TELP	Varchar	255	Nomor telpon user
LATITUDE	Float	-	Koordinat user
LONGITUDE	Float	-	Koordinat user
ROLE	Varchar	22	Role user

3.1.2 Deskripsi Entitas Thread

Nama	Tipe	Panjang	Keterangan
ID_THREAD	Integer	11	Id thread, <i>Primary key</i>
ID_USER	Integer	11	Id user, <i>Foreign key</i>
ID_COMMENT	Integer	11	Id comment, <i>Foreign key</i>
ID_KATEGORI	Integer	11	Id kategori, <i>Foreign key</i>
DESKRIPSI_THREAD	Varchar	255	Deskripsi thread
JUDUL_THREAD	Varchar	255	Judul thread
HARGA	Varchar	255	Harga barang
URL_GAMBAR1	Varchar	255	Link gambar barang
URL_GAMBAR2	Varchar	255	Link gambar barang
URL_GAMBAR3	Varchar	255	Link gambar barang
STATUS_THREAD	Varchar	255	Status thread
TANGGAL	Datetime	-	Tanggal pembuatan thread

3.1.3 Deskripsi Entitas Kategori Barang

Nama	Tipe	Panjang	Keterangan
ID_KATEGORI	Integer	11	Id kategori barang, <i>Primary key</i>
DESKRIPSI_KATEGORI	Varchar	255	Deskripsi kategori
ICON	Varchar	255	Gambar/ikon dari kategori barang

3.1.4 Deskripsi Entitas Comment

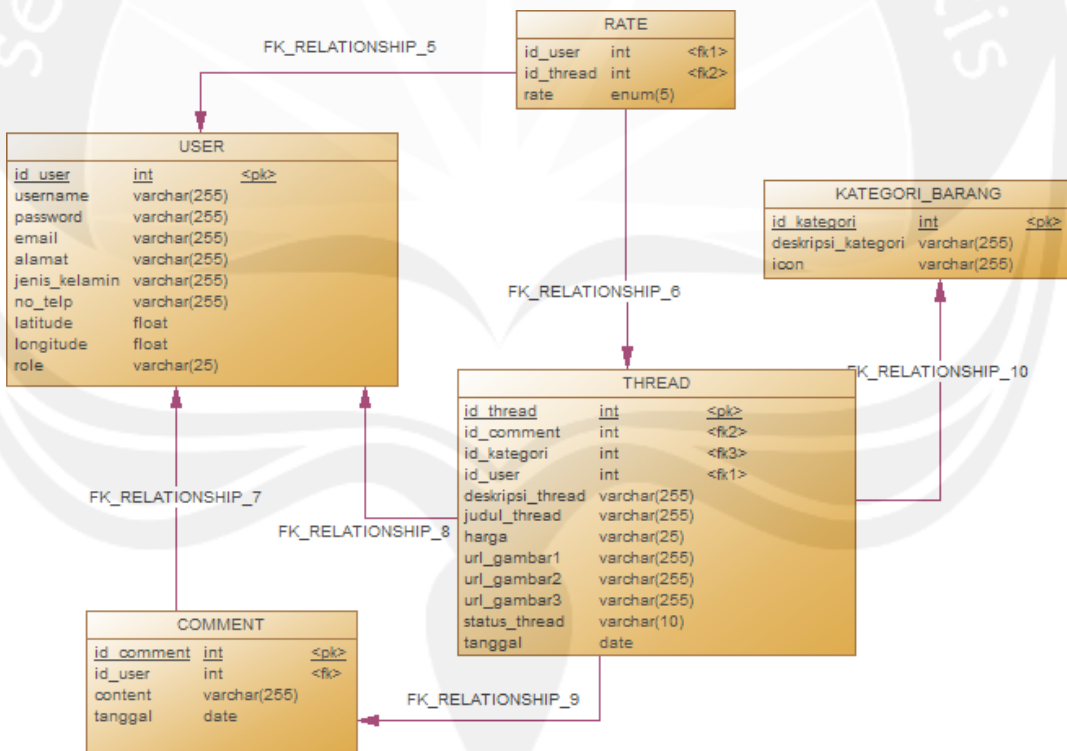
Nama	Tipe	Panjang	Keterangan
Program Studi Teknik Informatika		DPPL-LAPAK	30/38
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika			

ID_COMMENT	Integer	11	Id comment, Primary key
ID_USER	Integer	11	Id user, Foreign key
ID_THREAD	Integer	11	Id thread
CONTENT	Varchar	255	Isi comment
TANGGAL	Datetime	-	Tanggal dibuat comment

3.1.5 Deskripsi Entitas Rate

Nama	Tipe	Panjang	Keterangan
ID_USER	Integer	255	Id user, Foreign key
ID_THREAD	Integer	255	Id thread, Foreign key
RATING	Enum	5	Jumlah rating

3.2 Physical Data Model



Gambar 3.2. Physical Data Model

4 Perancangan Antarmuka

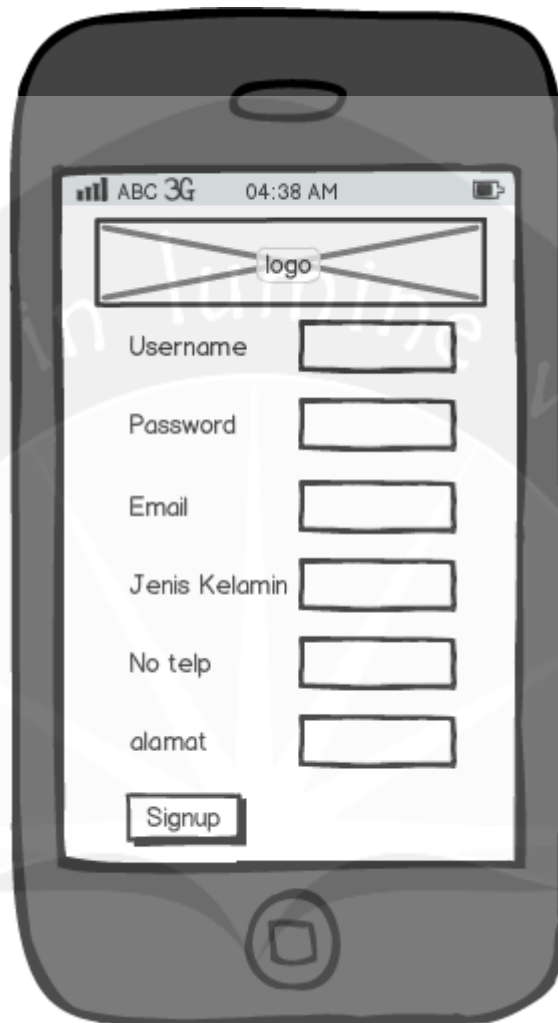
4.1 Login



Gambar 4.1. Rancangan Antarmuka Login

Antarmuka gambar 4.1 digunakan untuk mengotentikasi pengguna yang akan memasuki sistem. user dapat login dengan memasukkan username dan password di form yang disediakan, kemudian menekan tombol login di bawahnya.

4.2 Sign Up



The image shows a mobile phone screen with a sign-up form. The status bar at the top displays 'ABC 3G' and '04:38 AM'. Below the status bar is a logo placeholder. The form consists of the following fields and a button:

- Username:
- Password:
- Email:
- Jenis Kelamin:
- No telp:
- alamat:
- Signup:

Gambar 4.2. Rancangan Sign Up

Antarmuka gambar 4.2 digunakan mendaftarkan user menjadi member. Untuk mendaftar, pengguna harus mengisi username, password, email, jenis kelamin, nomor telpon, dan alamat pada form yang telah disediakan kemudian menekan tombol *signup*.

4.3 Home



Gambar 4.3. Rancangan Antarmuka Home

Antarmuka gambar 4.3 digunakan sebagai halaman muka LAPAK. Halaman ini akan menampilkan list kategori barang.

4.4 Buat Thread



Gambar 4.4. Rancangan Antarmuka Buat Thread

Antarmuka gambar 4.4 digunakan untuk membuat *thread*. Pengguna dapat mengedit *thread*-nya pada menu *thread* anda.

4.5 Thread Anda



Gambar 4.5. Rancangan Antarmuka Thread Anda

Antarmuka gambar 4.5 digunakan untuk melihat dan mengedit *thread*. Pengguna dapat mengedit, menghapus, dan mengubah status *thread*-nya.

4.6 Detail Thread



Gambar 4.6. Rancangan Antarmuka Thread Anda

Antarmuka gambar 4.6 digunakan untuk melihat *thread* dan untuk user yang sudah terdaftar menjadi member dapat memberi *comment* dan me-rating *thread*, serta dapat melihat lokasi penjual.

4.7 Lihat Lokasi



Gambar 4.7. Rancangan Antarmuka Lihat Lokasi

Antarmuka gambar 4.7 digunakan untuk melihat lokasi penjual, selain itu user juga dapat melihat rute jalan yang dapat dilalui untuk menuju lokasi penjual.