

## **BAB VI**

### **PENUTUP**

#### **6.1 Kesimpulan**

Berdasarkan analisis, desain dan implementasi aplikasi FORJUBEL dan pembahasan pada bab-bab sebelumnya, maka kesimpulan yang diperoleh adalah:

1. Tugas Akhir ini yaitu Aplikasi Forum Jual Beli berhasil dibangun dengan menggunakan framework *Android* dan *CodeIgniter*.
2. Aplikasi FORJUBEL dapat berjalan dengan baik pada handphone *Samsung Galaxy Ace S5380* dengan sistem operasi *Android* versi 2.3. Hubungan *device* dengan server aplikasi dapat berjalan dengan baik yang dimana *device* akan langsung mengakses server dari API FORJUBEL.

#### **6.2 Saran**

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

1. Menambahkan fungsionalitas pada web, agar fungsionalitas FORJUBEL tidak hanya sebatas pengelolaan report tetapi bisa juga menjadi sebuah Forum tersendiri pada web.

## DAFTAR PUSTAKA

- Basuki, A. p. (2010). *Membangun Web Berbasis PHP dengan framework Codeigniter*. Yogyakarta : Lokomedia.
- Greenspan, Jay, and Bulger,Brad, *MySQL/PHP Database Application*, M&T Books,Foster City CA USA, 2001.
- Gunadi, Kartika dan Christian Donny, 2004, *Pengembangan Aplikasi Jual dan Cari Properti pada Website www.idaman.com*.
- Hashimi, Sayed Y., et al., 2010, *Pro Android 2*, Apress, New York.
- Hendrayana, Ferry (dkk), 2011, *Pembuatan Aplikasi Web Jual Beli dan Lelang Online*.
- Lubis, J. (2001). *Internet User Behaviour*. McMillan Publishing.
- Manes, A.T., 2001, *Introduction to Web-services*.
- Rickyanto Isak, *Dasar Pemrograman Berorientasi Objek dengan Java 2 ( JDK 1.4)*, Andi Yogyakarta, Yogyakarta, 2003.
- Sasmita, G, M., Arya (dkk), 2009, *Sistem Transaksi Penjualan Kain Gringsing Berbasis Mobile(Mobile Commerce)*.
- Saylor, Michael,2012, *The Mobile Wave: How Mobile Intelligence Will Change Everything*, Perseus Books/Vanguard Press. page 304.
- Simon, Darwin (dkk), 2011, *Pengembangan Aplikasi Supermarket Online pada Platform Android dan iOS*.
- Tjiptono, F dan Diana, A, *Mengenal E-BUSINESS*. Yogyakarta: Andi 2001.
- Wijaya, Anthony, Januar (dkk), 2011, *Aplikasi Promosi dan Pencarian Rumah Tinggal Berbasis Android*.



**LAMPIRAN**



# SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

(FORJUBEL)

Aplikasi Forum Jual Beli berbasis Mobile

Untuk :


Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Aloysius Herry F. / 09.07.06044

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<b>SKPL-FORJUBEL</b>		1/24
		Revisi		

## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	

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	Fungsionalitas Produk .....	9
3.1.1	Mobile Aplikasi .....	9
3.1.1.1	Fungsi Login ( <b>SKPL-FORJUBEL-001</b> ) .....	9
3.1.1.2	Fungsi Registrasi ( <b>SKPL-FORJUBEL-002</b> ) .....	10
3.1.1.3	Fungsi Pengelolaan Data Akun ( <b>SKPL-FORJUBEL-003</b> ) ..	10
3.1.1.4	Fungsi Pengelolaan Data Thread ( <b>SKPL-FORJUBEL-004</b> )	11
3.1.2	Web Aplikasi .....	12
3.1.2.1	Fungsi Login ( <b>SKPL-FORJUBEL-005</b> ) .....	12
3.1.2.2	Fungsi Update Status Thread Order ( <b>SKPL-FORJUBEL-006</b> )	12
3.1.2.3	Fungsi Update Status Thread Transfer ( <b>SKPL-FORJUBEL-007</b> ) .....	12
3.1.2.4	Fungsi Tampil Data Laporan ( <b>SKPL-FORJUBEL-008</b> ) ...	12
3.1.2.5	Fungsi Hapus Thread ( <b>SKPL-FORJUBEL-009</b> ) .....	12
2.3	Karakteristik Pengguna .....	12
2.4	Batasan-batasan .....	12
2.5	Asumsi dan Ketergantungan .....	13
3	Ketentuan Khusus.....	13
3.1	Kebutuhan antarmuka eksternal .....	13
3.1.1	Antarmuka pemakai .....	13
3.1.2	Antarmuka perangkat keras .....	13
3.1.3	Antarmuka perangkat lunak .....	13
3.1.4	Antarmuka Komunikasi .....	15
3.2	Kebutuhan fungsionalitas Perangkat Lunak .....	15
3.2.1	Use Case Diagram .....	15
4	Spesifikasi Rinci Kebutuhan.....	15
4.1	Spesifikasi Kebutuhan Fungsionalitas .....	15
4.1.1	Use case Spesification : Login .....	15
4.1.2	Use case Spesification : Mengelola Data Akun .....	16
4.1.3	Use case Spesification : Mengelola Data Thread ...	18
5	Entity Relationship Diagram (ERD).....	24



## Daftar Gambar

Gambar 1.1	Arsitektur Perangkat lunak FORJUBEL .....	9
Gambar 3.1	Use Case Diagram .....	15
Gambar 5.1	Entity Relationship Diagram .....	24



## 1 Pendahuluan

### 1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak FORJUBEL (*Forum Jual Beli*) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara aplikasi dengan aplikasi lain perangkat lunak dan perangkat keras, dan pengguna) performansi (kemampuan perangkat lunak dari segi kecepatan, tempat penyimpanan yang dibutuhkan, serta keakuratan), dan atribut (*feature-feature* tambahan yang dimiliki aplikasi), serta mendefinisikan fungsi perangkat lunak. SKPL-FORJUBEL ini juga mendefinisikan batasan perancangan perangkat lunak.

### 1.2 Lingkup Masalah

Perangkat Lunak FORJUBEL dikembangkan dengan tujuan untuk :

1. Menangani pengelolaan data user.
2. Menangani pengelolaan pencatatan data thread dan status thread.
3. Menangani pencarian data informasi thread.
4. Menangani proses transaksi konsumen dengan konsumen.

### 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-FORJUBEL-XXX	Kode yang merepresentasikan kebutuhan pada FORJUBEL ( <i>Courier Service and Delivery' Service</i> ) dimana XXX merupakan nomor fungsi produk.
FORJUBEL	Perangkat lunak yang menyediakan jasa penjualan dan pembelian barang.
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.
Role	Hak akses yang dimiliki user untuk menggunakan aplikasi.

### 1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. <http://developer.android.com>, Android, 2012.
2. Wibisono, Aryo, *Spesifikasi Kebutuhan Perangkat Lunak SIPengAP*, 2009, PT.(Persero) Angkasa Pura I, Yogyakarta.

### 1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 (tiga) bagian utama. Bagian utama berisi penjelasan mengenai

Program Studi Teknik Informatika	SKPL – FORJUBEL	7/ 24
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		

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 FORJUBEL 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 FORJUBEL tersebut.

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

## **2 Deskripsi Kebutuhan**

### **2.1 Perspektif produk**

FORJUBEL merupakan perangkat lunak yang dikembangkan untuk membantu dalam mengelola proses penjualan dan pembelian thread. Aplikasi ini menangani pengelolaan data tarif, thread, dan konsumen. Aplikasi ini memudahkan customer untuk melihat dan mencari (*tracking*) data thread. Selain itu, customer juga dapat mengetahui status thread.

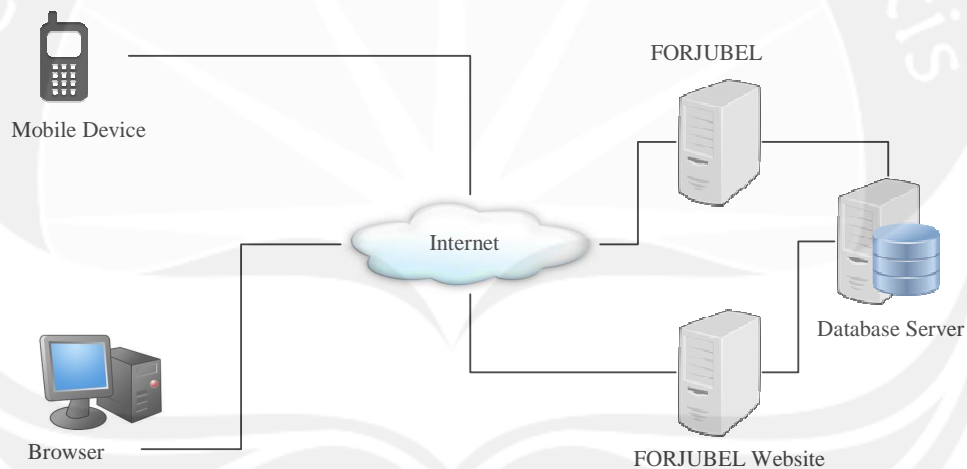
Perangkat lunak FORJUBEL ini berjalan pada mobile device, dan dibuat menggunakan bahasa pemrograman Java. Sedangkan untuk lingkungan pemrogramannya menggunakan Eclipse Juno.

Pengguna akan berinteraksi dengan aplikasi melalui antarmuka GUI (Graphical User Interface). Pada aplikasi

Program Studi Teknik Informatika	SKPL – FORJUBEL	8/ 24
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		

ini, seperti terlihat pada gambar 1, arsitektur perangkat lunak yang digunakan berupa client server, di mana semua data disimpan di server. User dapat mengakses data yang ada 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, maka data yang diinginkan akan dicari ke database server yang selanjutnya dikirimkan ke client yang merequest melalui web server.



Gambar 1.1 Arsitektur Perangkat lunak FORJUBEL

## 2.2 Fungsionalitas Produk

Fungsi produk perangkat lunak FORJUBEL adalah sebagai berikut :

### 3.1.1 Mobile Aplikasi

#### 3.1.1.1 Fungsi Login (SKPL-FORJUBEL-001)

Fungsi yang digunakan oleh semua pengguna untuk bisa masuk kedalam aplikasi mobile.

Program Studi Teknik Informatika	SKPL - FORJUBEL	9/ 24
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		

3.1.1.2 Fungsi Registrasi (**SKPL-FORJUBEL-002**)

Fungsi yang digunakan pengguna untuk mendaftarkan diri pada aplikasi FORJUBEL.

3.1.1.3 Fungsi Pengelolaan Data Akun (**SKPL-FORJUBEL-003**)

Fungsi yang digunakan oleh semua user yang telah mendaftarkan diri untuk mengelola data akun.

3.1 Fungsi Ubah Password (**SKPL-FORJUBEL-003-01**)

Fungsi yang digunakan user untuk mengubah password.

3.2 Fungsi Ubah Akun (**SKPL-FORJUBEL-003-02**)

Fungsi yang digunakan user untuk mengubah data-data diri.

3.3 Fungsi Tampil Bookmark Thread (**SKPL-FORJUBEL-003-03**)

Fungsi yang digunakan untuk mencari data thread.

3.4 Fungsi Delete Bookmark Thread (**SKPL-FORJUBEL-003-04**)

Fungsi yang digunakan user untuk menghapus thread yang di bookmark.

3.5 Fungsi Tampil Thread User (**SKPL-FORJUBEL-003-05**)

Fungsi yang digunakan user untuk melihat thread yang dibuat beserta status transaksi.

3.6 Fungsi Upload Bukti Pembayaran (**SKPL-FORJUBEL-003-06**)

Fungsi yang digunakan user untuk mengirimkan bukti pembayaran ke web server.

3.7 Fungsi Update Status Barang sedang Dikirim (**SKPL-FORJUBEL-003-07**)

Fungsi yang digunakan user untuk mengupdate status jika barang telah dikirim.

3.8 Fungsi Update Status Barang Sampai (**SKPL-FORJUBEL-003-08**)

Fungsi yang digunakan user untuk mengupdate status jika barang telah sampai.

3.9 Fungsi Batal Pesanan Barang (**SKPL-FORJUBEL-003-09**)

Fungsi yang digunakan user untuk membatalkan pemesanan barang.

3.1.1.4 Fungsi Pengelolaan Data Thread (**SKPL-FORJUBEL-004**)

4.1 Fungsi Entri Data Thread (**SKPL-FORJUBEL-004-01**)

Fungsi yang digunakan untuk menambah data thread kedalam sistem.

4.2 Fungsi Edit Data Thread (**SKPL-FORJUBEL-004-02**)

Fungsi yang digunakan untuk mengubah data thread.

4.3 Fungsi Hapus Data Thread (**SKPL-FORJUBEL-004-03**)

Fungsi yang digunakan untuk menghapus data thread.

4.4 Fungsi Tampil Data Thread (**SKPL-FORJUBEL-004-04**)

Fungsi yang digunakan untuk menghapus data thread.

4.5 Fungsi Cari Data Thread (**SKPL-FORJUBEL-004-05**)

Fungsi yang digunakan untuk mencari data thread.

4.6 Fungsi Bookmark Thread (**SKPL-FORJUBEL-004-06**)

Fungsi yang digunakan user untuk menandai sebuah thread.

4.7 Fungsi Pesan Barang Thread (**SKPL-FORJUBEL-004-07**)

Fungsi yang digunakan user untuk memesan barang pada sebuah thread.

### 3.1.2 Web Aplikasi

#### 3.1.2.1 Fungsi Login (SKPL-FORJUBEL-005)

Fungsi yang digunakan oleh admin untuk bias masuk kedalam sistem.

#### 3.1.2.2 Fungsi Update Status Thread Order (SKPL-FORJUBEL-006)

Fungsi yang digunakan admin untuk memperbarui status thread order menjadi transfer.

#### 3.1.2.3 Fungsi Update Status Thread Transfer (SKPL-FORJUBEL-007)

Fungsi yang digunakan admin untuk memperbarui status thread transfer thread menjadi sold.

#### 3.1.2.4 Fungsi Tampil Data Laporan (SKPL-FORJUBEL-008)

Fungsi yang digunakan admin untuk melihat daftar penjualan tertentu berdasarkan bulan.

#### 3.1.2.5 Fungsi Hapus Thread (SKPL-FORJUBEL-009)

Fungsi yang digunakan admin untuk menghapus thread pengguna.

## 2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak FORJUBEL adalah sebagai berikut :

1. Memahami pengoperasian Perangkat Mobile.
2. Memahami penggunaan Aplikasi Jual Beli (FORJUBEL).

## 2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak FORJUBEL tersebut adalah :

Program Studi Teknik Informatika	SKPL – FORJUBEL	12/ 24
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. Kebijakan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak FORJUBEL.

### 2. Keterbatasan perangkat keras

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

## 2.5 Asumsi dan Ketergantungan

Aplikasi ini dapat dijalankan pada perangkat Mobile SmartPhone yang menggunakan sistem operasi versi Android 2.3.

## 3 Ketentuan Khusus

### 3.1 Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak FORJUBEL 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 FORJUBEL adalah:

1. Perangkat Mobile Android.

#### 3.1.3 Antarmuka perangkat lunak

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

1. Nama : Java

Sumber : Oracle Corporation

Sebagai bahasa pemrograman mobile yang digunakan untuk membuat FORJUBEL.

2. Nama : PHP

Sumber : The PHP Group

Sebagai bahasa pemrograman web service yang digunakan untuk membuat FORJUBEL.

3. Nama : Eclipse

Sumber : Free and Open Source Community

Sebagai tool perancangan yang dibutuhkan untuk membuat FORJUBEL.

4. Nama : Notepad++

Sumber : Don Ho

Sebagai tool perancangan yang dibutuhkan untuk membuat FORJUBEL.

5. Nama : Windows 7

Sumber : Microsoft

Sebagai aplikasi operasi yang digunakan dalam FORJUBEL.

6. Nama : PHP

Sumber : The PHP Group

Sebagai web browser yang digunakan dalam FORJUBEL.

7. Nama : Android SDK

Sumber : Google Inc.

Sebagai library dalam perancangan aplikasi FORJUBEL.

8. Nama : MySQL

Sumber : MySQL

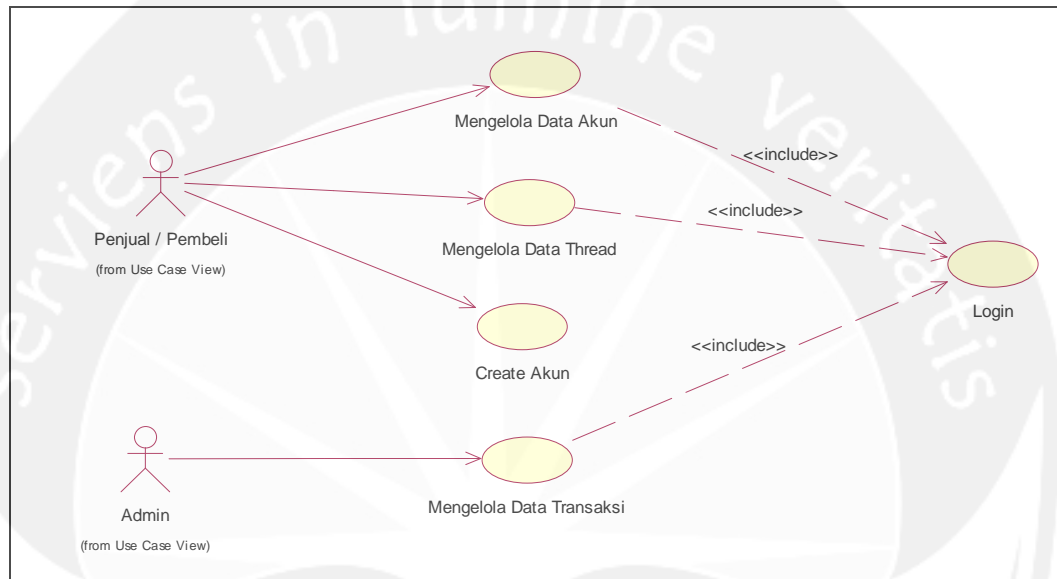
Sebagai database management system (DBMS) yang digunakan untuk penyimpanan data di sisi server.

### 3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak FORJUBEL menggunakan protocol HTTP.

## 3.2 Kebutuhan fungsionalitas Perangkat Lunak

### 3.2.1 Use Case Diagram



Gambar 3.1 Use Case Diagram

## 4 Spesifikasi Rinci Kebutuhan

### 4.1 Spesifikasi Kebutuhan Fungsionalitas

#### 4.1.1 Use case Specification : Login

##### 1. Brief Description

Use Case ini digunakan oleh aktor untuk memperoleh akses ke mobile aplikasi. Login didasarkan pada sebuah id unik yaitu nama dari user dan password yang berupa rangkaian karakter.

##### 2. Primary Actor

1. Penjual / Pembeli

##### 3. Supporting Actor

1. Admin

##### 4. Basic Flow

Program Studi Teknik Informatika	SKPL – FORJUBEL	15/ 24
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 login
2. Aplikasi menampilkan antarmuka untuk login
3. Aktor memasukkan id dan password
4. Aplikasi memeriksa id dan password yang diinputkan aktor
  - E-1 Password atau id user tidak sesuai
5. Aplikasi memberikan akses ke aktor
6. Use Case ini selesai

**5. Alternative Flow**

None

**6. Error Flow**

E-1 Password atau nama user tidak sesuai

1. Aplikasi menampilkan peringatan bahwa id user atau password tidak sesuai
2. Kembali ke Basic Flow langkah ke 3

**7. PreConditions**

None

**8. PostConditions**

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

**4.1.2 Use case Spesification : Mengelola Data Akun**

**1. Brief Description**

Use Case ini digunakan oleh aktor untuk mengubah data akun dan password.

**2. Primary Actor**

1. Penjual / Pembeli

**3. Supporting Actor**

1. Admin

**4. Basic Flow**

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

2. Aplikasi menampilkan antarmuka untuk mengubah data akun

A-1 Aktor memilih mengubah password

A-2 Aktor memilih mengupload bukti pembayaran barang

A-3 Aktor memilih mengupdate status pengiriman barang

A-4 Aktor memilih mengupdate status barang yang telah sampai

A-5 Aktor memilih membatalkan pesanan yang telah dibuat

3. Aktor memasukkan data diri yang baru

4. Aplikasi memeriksa data yang diinputkan aktor

E-1 Data tidak sesuai

5. Aplikasi mengupdate data akun terbaru

6. Use Case ini selesai

#### 5. Alternative Flow

A-1 Aktor memilih untuk mengubah password

1. Aktor memasukkan password lama

2. Aktor memasukkan password baru

3. Aplikasi mengecek password lama user

E-2 Password lama tidak sesuai

4. Aplikasi mengupdate password user yang baru

5. use case selesai

A-2 Aktor memilih mengupload bukti pembayaran barang

1. Aktor memilih thread yang status bukti pembayarannya belum dikirim

2. Aktor menekan tombol send

3. Aplikasi mengirimkan bukti pembayaran dan mengupdate status thread

4. use case selesai

A-3 Aktor memilih mengupdate status pengiriman barang

1. Aktor memilih thread yang status threadnya transferred

2. Aktor menekan tombol update
3. Aplikasi mengupdate status thread
4. use case selesai

A-4 Aktor memilih mengupdate status barang yang telah sampai

1. Aktor memilih thread yang barangnya telah sampai
2. Aktor menekan tombol update
3. Aplikasi mengupdate status thread
4. use case selesai

A-5 Aktor memilih membatalkan pesanan yang telah dibuat

1. Aktor memilih thread yang yang ingin dibatalkan
2. Aktor menekan tombol cancel order
3. Aplikasi mengupdate status thread
4. use case selesai

## 6. Error Flow

E-1 Data tidak sesuai

1. Aplikasi menampilkan peringatan bahwa data yang diinputkan tidak sesuai
2. Kembali ke Basic Flow langkah ke 3

E-2 Password lama tidak sesuai

1. Aplikasi menampilkan peringatan bahwa password lama tidak sesuai
2. Kembali ke Alternative Flow ke 1

## 7. PreConditions

None

## 8. PostConditions

1. Aktor berhasil memperbarui data akun atau password.

### 4.1.3 Use case Spesification : Mengelola Data Thread

#### 1. Brief Description

Use Case ini digunakan oleh user untuk mengelola data thread. Aktor dapat melakukan entri data thread, tampil data thread, cari data thread, bookmark thread atau order thread.

## 2. Primary Actor

1. Penjual / Pembeli

## 3. Supporting Actor

1. Admin

## 4. Basic Flow

1. Use Case ini dimulai ketika Aktor memilih untuk melakukan pengelolaan data thread.
2. Aplikasi menampilkan form pengisian thread.
3. Aktor memilih untuk melakukan entri data thread
  - A-1 Aktor memilih untuk melakukan pencarian data thread
  - A-2 Aktor memilih untuk melakukan bookmark thread
  - A-3 Aktor memilih untuk melakukan order thread
4. Aktor menginputkan data thread
5. Aktor meminta aplikasi untuk menyimpan data thread baru yang telah diinputkan
  - E-1 Format data inputan tidak sesuai
6. Aplikasi menyimpan data thread yang terbaru ke database
7. Use Case selesai

## 5. Alternative Flow

- A-1 Aktor memilih untuk melakukan cari data thread
  1. Aktor menginputkan nama thread, memilih kategori atau memilih lokasi dari data thread yang ingin dicari
  2. Aplikasi menampilkan data thread yang dicari
    - E-2 Data thread yang dicari tidak ditemukan
  3. Kembali ke basic flow langkah 7
- A-2 Aktor memilih untuk melakukan bookmark thread
  1. Aktor memilih thread yang ingin di bookmark
  2. Thread berhasil di bookmark
    - E-3 Thread gagal di bookmark
  3. Aplikasi menyimpan hasil bookmark thread
  4. Kembali ke basic flow langkah 7

A-3 Aktor memilih untuk melakukan order thread

1. Aktor memilih thread yang ingin dipesan
2. Thread berhasil di pesan

E-4 Thread gagal di order

3. Aplikasi menyimpan status order user
4. Kembali ke basic flow langkah 7

#### 6. Error Flow

E-1 Format data inputan tidak sesuai

1. Aplikasi memberikan pesan peringatan bahwa format data yang diinputkan salah
2. Kembali ke Basic Flow Langkah ke 4

E-2 Data thread yang dicari tidak ditemukan

1. Aplikasi memberikan pesan informasi bahwa data thread yang dicari tidak ditemukan.
2. Kembali ke Alternatif Flow A-1 langkah ke 1

E-3 Thread gagal di bookmark

1. Aplikasi memberikan peringatan bahwa thread telah di bookmark
2. Kembali ke Alternatif Flow A-2 langkah 4

E-4 Thread gagal di order

1. Aplikasi memberikan peringatan bahwa thread telah di order
2. Kembali ke Alternatif Flow A-3 langkah 4

#### 7. PreConditions

1. Use Case Login telah dilakukan
2. Aktor telah memasuki aplikasi sebagai User

#### 8. PostConditions

1. Data thread di database telah ter-update

#### 4.1.4 Use case Spesification : Login

##### 1. Brief Description

Use Case ini digunakan oleh aktor untuk memperoleh akses ke web aplikasi. Login didasarkan pada sebuah id



unik yaitu nama dari user dan password yang berupa rangkaian karakter.

## **2. Primary Actor**

1. Admin

## **3. Supporting Actor**

None

## **4. Basic Flow**

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

E-1 Password atau id user tidak sesuai

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

## **5. Alternative Flow**

None

## **6. Error Flow**

E-1 Password atau nama user tidak sesuai

1. Aplikasi menampilkan peringatan bahwa id user atau password tidak sesuai
2. Kembali ke Basic Flow langkah ke 3

## **7. PreConditions**

None

## **8. PostConditions**

1. Aktor memasuki web aplikasi dan dapat menggunakan fungsi-fungsi pada aplikasi.

### **4.1.5 Use case Spesification : Mengelola Data Status Transaksi**

#### **1. Brief Description**

Program Studi Teknik Informatika	SKPL – FORJUBEL	21/ 24
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 data transaksi. Aktor dapat mengubah status transaksi dari sebuah thread dan dapat melihat laporan bulanan.

## **2. Primary Actor**

1. Admin

## **3. Supporting Actor**

None

## **4. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk mengupdate status transaksi order menjadi transfer.
2. Aplikasi menampilkan data thread dengan harga yang terbaru.
3. Aktor memilih untuk mengupdate status transaksi order
  - A-1 Aktor memilih untuk melakukan update status transaksi transfer menjadi sold
  - A-2 Aktor memilih untuk melihat laporan
  - A-3 Aktor memilih untuk tidak melakukan transaksi
4. Aktor mengupdate status thread
  - E-1 User belum login kedalam aplikasi
5. Aplikasi mengupdate data status thread terbaru ke database
6. Use case selesai

## **5. Alternative Flow**

- A-1 Aktor memilih untuk melakukan update status transaksi transfer menjadi sold
  1. Aktor memilih daftar thread yang akan diupdate
  2. Aktor mengupdate status thread
  3. Berlanjut ke Basic Flow langkah ke 4
- A-2 Aktor memilih untuk melihat laporan
  1. Actor memasukan range tanggal dari thread yang ingin dicari.
  2. Sistem mengeksekusi perintah user
    - E-2 Thread tidak ditemukan
  3. Berlanjut ke Basic Flow langkah 6

A-3 Aktor memilih untuk tidak melakukan transaksi

1. Berlanjut ke Basic Flow langkah ke 6

#### **6. Error Flow**

E-1 User belum login kedalam aplikasi

1. Aplikasi memberikan pesan peringatan bahwa user harus login terlebih dahulu
2. Kembali ke Basic Flow Langkah ke 3

E-2 Thread tidak ditemukan

1. Aplikasi memberikan pesan peringatan bahwa thread yang dicari tidak ada
2. Kembali ke Alternatif Flow Langkah ke 1

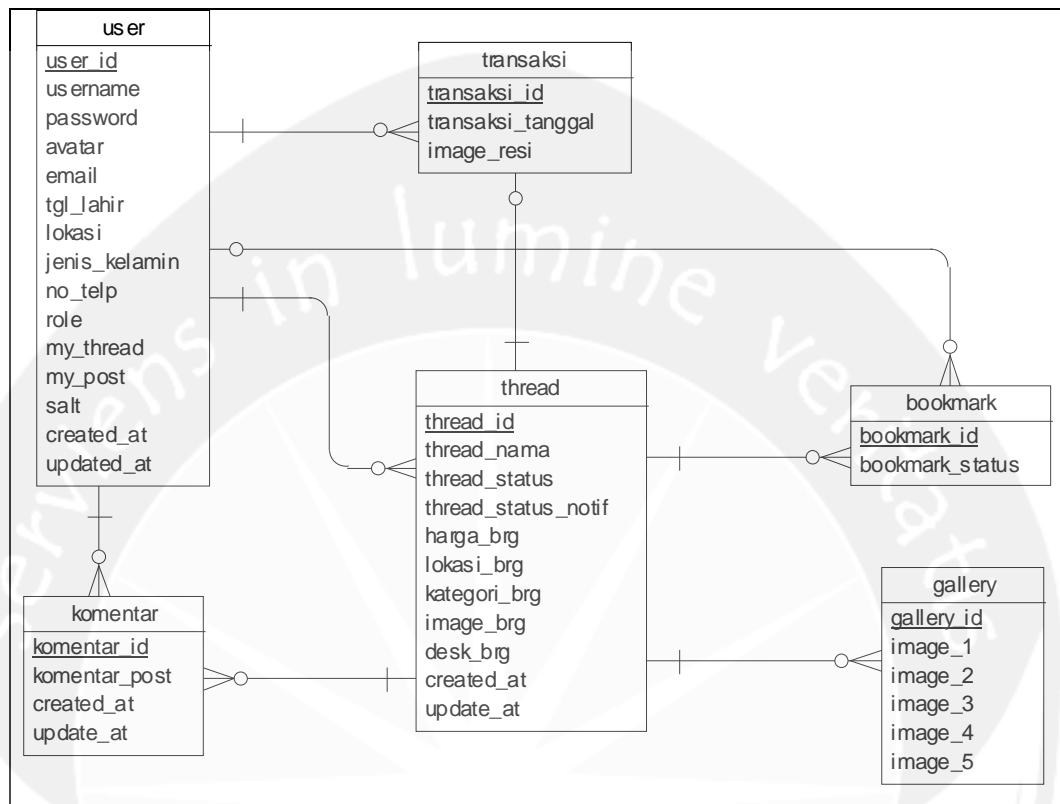
#### **7. PreConditions**

1. Use Case Login sudah dilakukan
2. Aktor telah memasuki aplikasi sebagai admin

#### **8. PostConditions**

1. Data status transaksi di database telah ter-update
2. Admin dapat melihat laporan

## 5 Entity Relationship Diagram (ERD)



Gambar 5.1 Entity Relationship Diagram

**DPPL**

**DESKRIPSI PERANCANGAN PERANGKAT LUNAK**

**(FORJUBEL)**

**Aplikasi Forum Jual Beli berbasis Mobile**

**Untuk :**


**Universitas Atma Jaya Yogyakarta**

**Dipersiapkan oleh:**

**Aloysius Herry F. / 09.07.06044**

**Program Studi Teknik Informatika - Fakultas Teknologi  
Industri**

**Universitas Atma Jaya Yogyakarta**

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<b><i>DPPL-FORJUBEL</i></b>		1/54
		Revisi		

## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	

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	Ruang Lingkup.....	6
1.3	Definisi dan Akronim.....	6
1.4	Referensi.....	7
2	Perancangan Sistem .....	8
2.1	Perancangan Arsitektur.....	8
2.1.1	Mobile.....	8
2.1.2	Web.....	10
	Web Service.....	10
2.2	Perancangan Rinci.....	11
2.2.1	Sequence Diagram.....	11
2.2.2	Class Diagram.....	22
2.2.3	Spesifikasi Deskripsi Kelas Diagram.....	24
3	Perancangan Data .....	42
3.1	Dekomposisi Data.....	42
3.1.1	Deskripsi Entitas Data Tabel User.....	42
3.1.2	Deskripsi Entitas Data Tabel Thread.....	43
3.1.3	Deskripsi Entitas Data Tabel Komentar.....	43
3.1.4	Deskripsi Entitas Data Tabel Gallery.....	43
3.1.5	Deskripsi Entitas Data Tabel Transaksi.....	44
3.1.6	Deskripsi Entitas Data Tabel Bookmark.....	44
3.2	Physical Data Model.....	45
4	Perancangan Antar Muka .....	46
4.1	Sketsa UI dan Deskripsinya.....	46
4.1.1	Antarmuka Login.....	46
4.1.2	Antarmuka Pendaftaran User.....	47
4.1.3	Antarmuka Edit Password.....	47
4.1.4	Antarmuka Edit Account User.....	48
4.1.5	Antarmuka Buat Thread.....	49
4.1.6	Antarmuka Edit Thread.....	49
4.1.7	Antarmuka Pencarian Thread.....	50
4.1.8	Antarmuka Tampil Thread.....	51
4.1.9	Antarmuka Tampil Detail My Thread.....	51
4.1.10	Antarmuka Tampil Detail Thread.....	52
4.1.11	Antarmuka Login Web.....	53
4.1.12	Antarmuka Tampil Thread Order.....	53
4.1.13	Antarmuka Tampil Thread Transfer.....	54
4.1.14	Antarmuka Tampil Report Thread.....	54
4.1.15	Antarmuka Delete Thread.....	55



## Daftar Gambar

Gambar 2.1 Rancangan Arsitektur FORJUBEL Mobile App .....	9
Gambar 2.2 Rancangan Arsitektur FORJUBEL Web App.....	10
Gambar 2.3 Rancangan Arsitektur FORJUBEL Web Service.....	10
Gambar 2.4 Sequence Diagram : Login.....	11
Gambar 2.5 Sequence Diagram : Pendaftaran .....	11
Gambar 2.6 Sequence Diagram : Edit Data Akun .....	12
Gambar 2.7 Sequence Diagram : Edit Password.....	12
Gambar 2.8 Sequence Diagram : Display Bookmark User .....	13
Gambar 2.9 Sequence Diagram : Display Thread Order User.....	13
Gambar 3.10 Sequence Diagram : Display Thread User.....	14
Gambar 3.11 Sequence Diagram : Entri Data Thread.....	14
Gambar 3.12 Sequence Diagram : Search Data Thread .....	15
Gambar 3.13 Sequence Diagram : Bookmark Thread .....	15
Gambar 3.14 Sequence Diagram : Order Thread.....	16
Gambar 3.15 Sequence Diagram : Entri Data Comment Thread....	16
Gambar 3.16 Sequence Diagram : Display Data Thread.....	17
Gambar 3.17 Sequence Diagram : Login Web.....	17
Gambar 3.18 Sequence Diagram : Update Status Transaksi Order .....	18
Gambar 3.19 Sequence Diagram : Cancel Order .....	19
Gambar 3.20 Sequence Diagram : Upload Receipt.....	19
Gambar 3.21 Sequence Diagram : Update Status Transaksi Transfer .....	20
Gambar 3.22 Sequence Diagram : Display Report Thread Sold..	21
Gambar 3.23 Mobile Class Diagram.....	23
Gambar 3.24 Web Class Diagram.....	23
Gambar 3.25 Web Service Class Diagram.....	24
Gambar 3.26 Rancangan Arsitektur : Physical Data Model .....	45
Gambar 4.1 Rancangan Antarmuka Login .....	46
Gambar 4.2 Rancangan Antarmuka Pendaftaran User .....	47
Gambar 4.3 Rancangan Antarmuka Edit Password .....	47
Gambar 4.4 Rancangan Antarmuka Edit Account Password .....	48
Gambar 4.5 Rancangan Antarmuka Buat Thread .....	49
Gambar 4.6 Rancangan Antarmuka Edit Thread .....	49
Gambar 4.7 Rancangan Antarmuka Pencarian Thread .....	50
Gambar 4.8 Rancangan Antarmuka Tampil Thread .....	51
Gambar 4.9 Rancangan Antarmuka Tampil Detail My Thread ...	51
Gambar 4.10 Rancangan Antarmuka Tampil Detail Thread .....	52
Gambar 4.11 Rancangan Antarmuka Login Web .....	53
Gambar 4.12 Antarmuka Tampil Thread Order .....	53
Gambar 4.13 Antarmuka Tampil Thread Transfer .....	54
Gambar 4.14 Antarmuka Tampil Report Thread .....	54
Gambar 4.15 Antarmuka Delete Thread .....	55

## 1 Pendahuluan

### 1.1 Tujuan

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

### 1.2 Ruang Lingkup

Perangkat Lunak FORJUBEL dikembangkan dengan tujuan :

1. Menangani pengelolaan data akun.
2. Menangani pengelolaan data thread.
3. Menangani pengelolaan data status transaksi.
4. Menangani penampilan data laporan berdasarkan bulan.

Dan berjalan pada lingkungan dengan platform mobile android.

### 1.3 Definisi dan Akronim

Beberapa definisi, akronim, singkatan yang digunakan dalam DPPL FORJUBEL ini antara lain adalah sebagai berikut:

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

	jasa penjualan dan pembelian barang.
GUI	<i>Graphical User Interface</i> , tampilan antarmuka program bagi pengguna.
Database	Kumpulan data terkait yang diorganisasikan dalam struktur tertentu dan dapat diakses dengan cepat.
Combobox	Control yang digunakan untuk menampilkan daftar pilihan menu sehingga pengguna tinggal memilih informasi yang sudah disediakan.

#### 1.4 Referensi

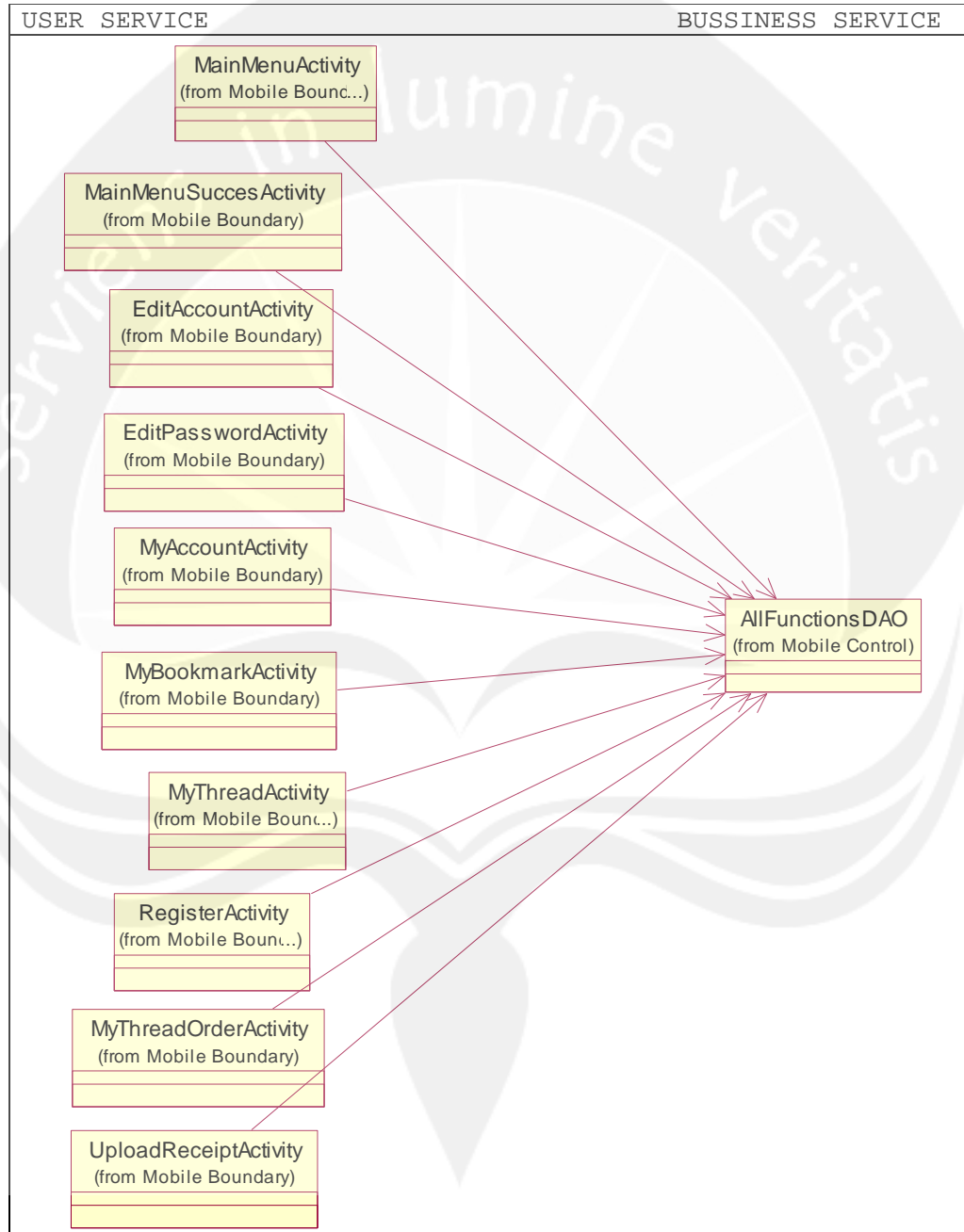
Referensi yang digunakan pada perangkat lunak tersebut adalah:

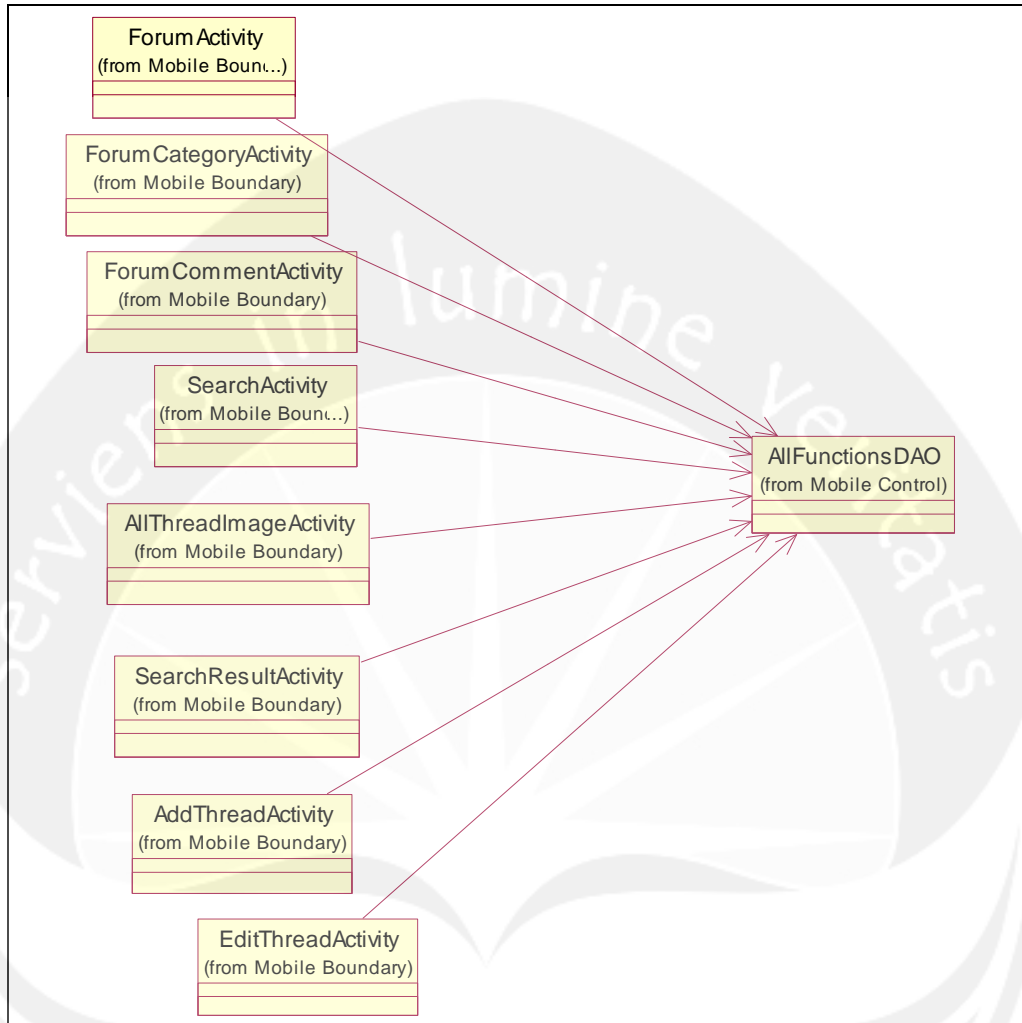
1. <http://developer.android.com>, Android, 2012.
2. Wibisono, Aryo, *Spesifikasi Kebutuhan Perangkat Lunak SIPengAP*, 2009, PT.(Persero) Angkasa Pura I, Yogyakarta.

## 2 Perancangan Sistem

### 2.1 Perancangan Arsitektur

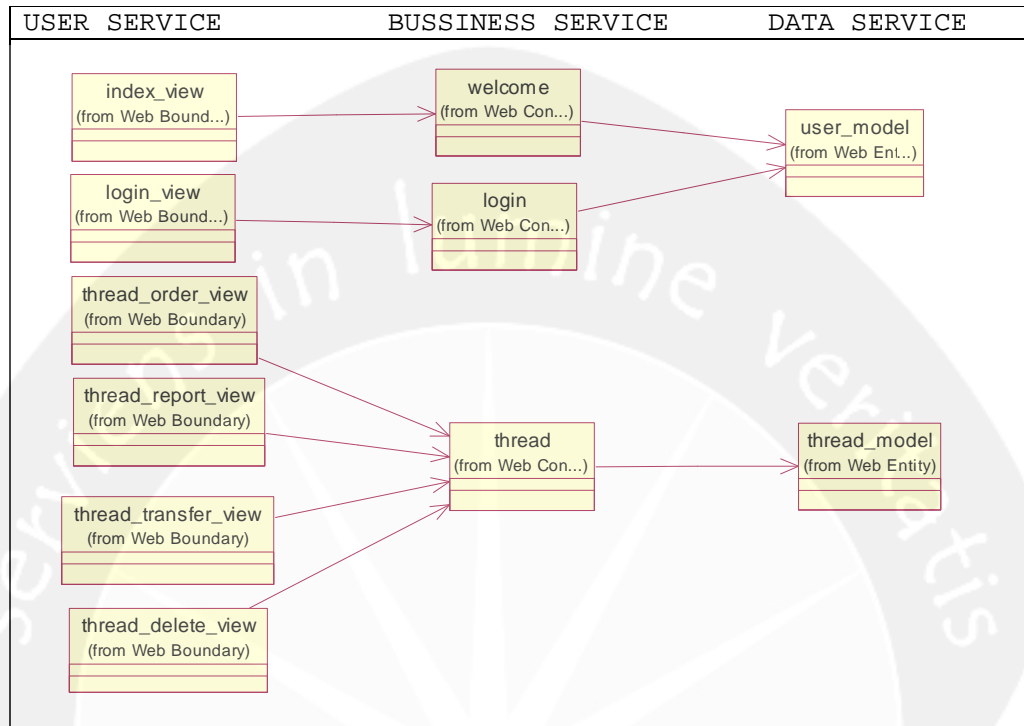
#### 2.1.1 Mobile





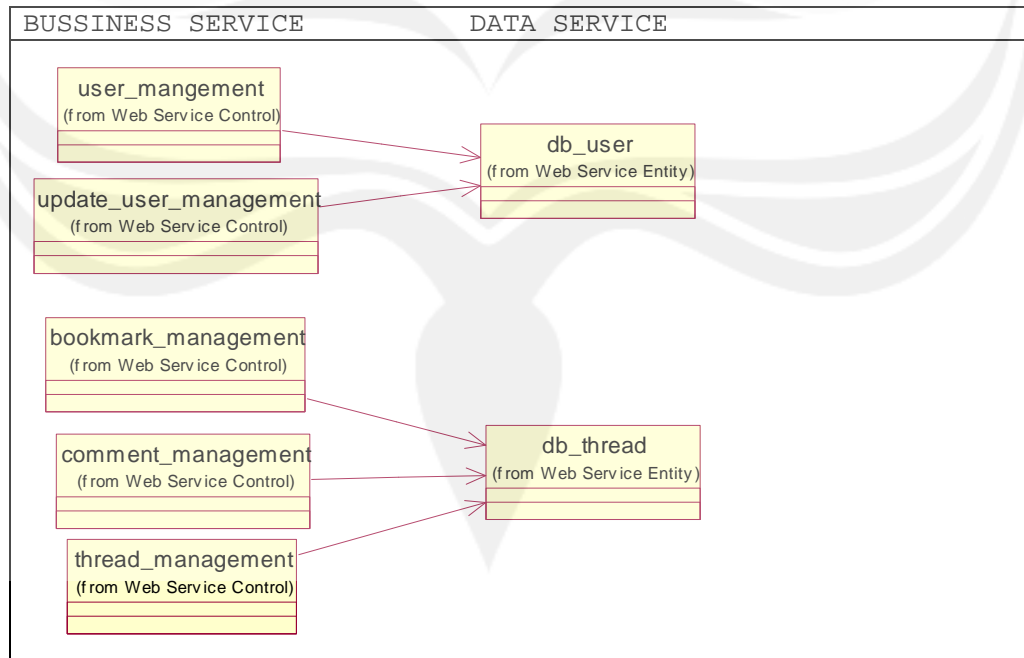
Gambar 2.1 Rancangan Arsitektur FORJUBEL Mobile App

### 2.1.2 Web



Gambar 2.2 Rancangan Arsitektur FORJUBEL Web App

### Web Service



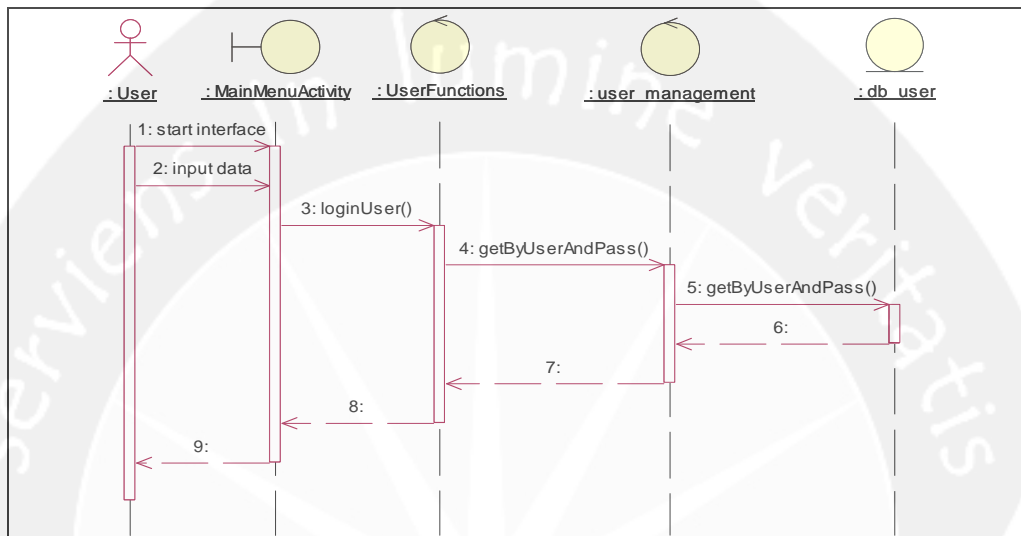
Gambar 2.3 Rancangan Arsitektur FORJUBEL Web Service

## 2.2 Perancangan Rinci

### 2.2.1 Sequence Diagram

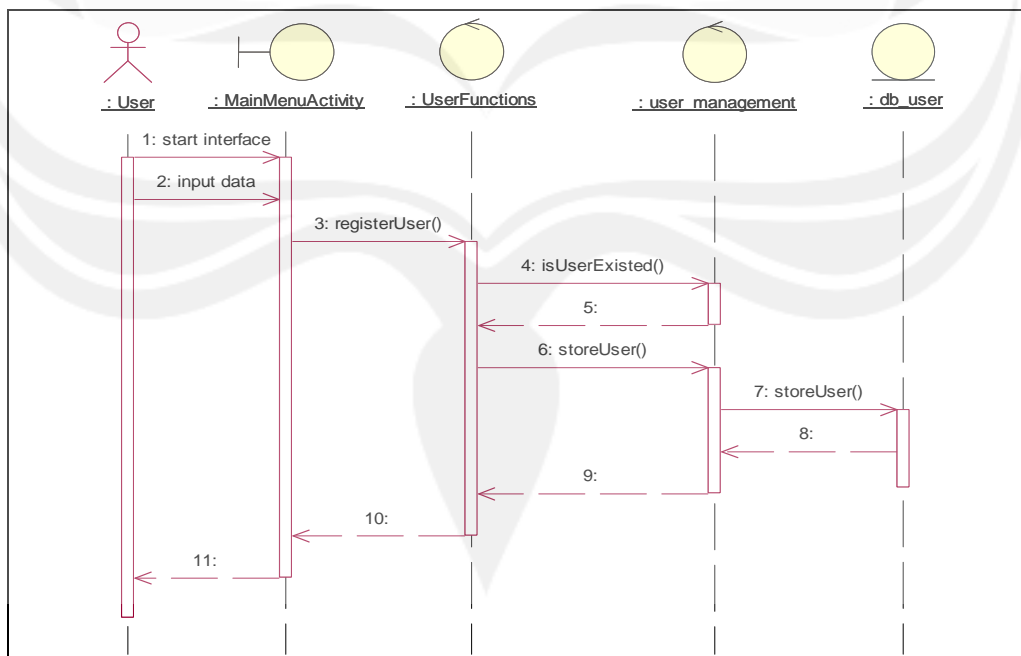
#### 2.2.1.1 Autentikasi (Pengguna aplikasi FORJUBEL)

##### 2.2.1.1.1 Login Mobile



Gambar 2.4 Sequence Diagram : Login

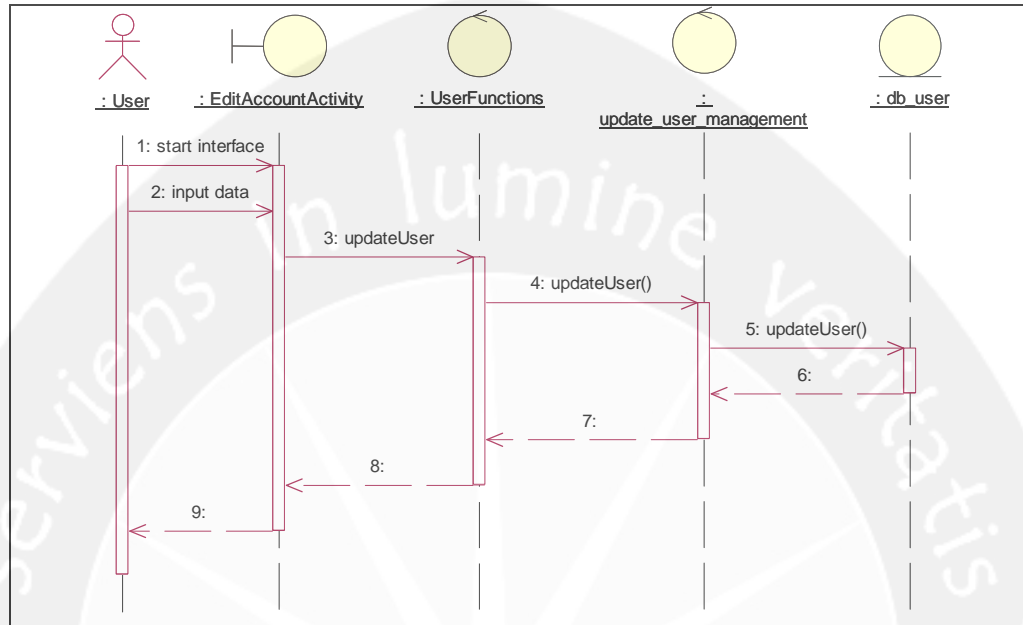
##### 2.2.1.1.2 Pendaftaran



Gambar 2.5 Sequence Diagram : Pendaftaran

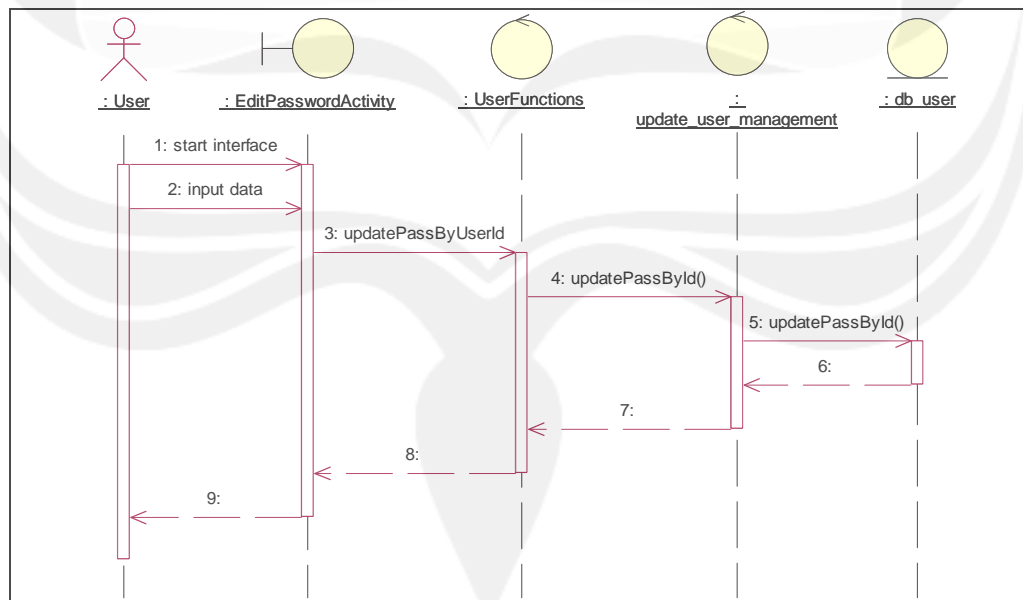
## 2.2.1.2 Pengelolaan Data Akun

### 2.2.1.2.1 Edit Data Akun



Gambar 2.6 Sequence Diagram : Edit Data Akun

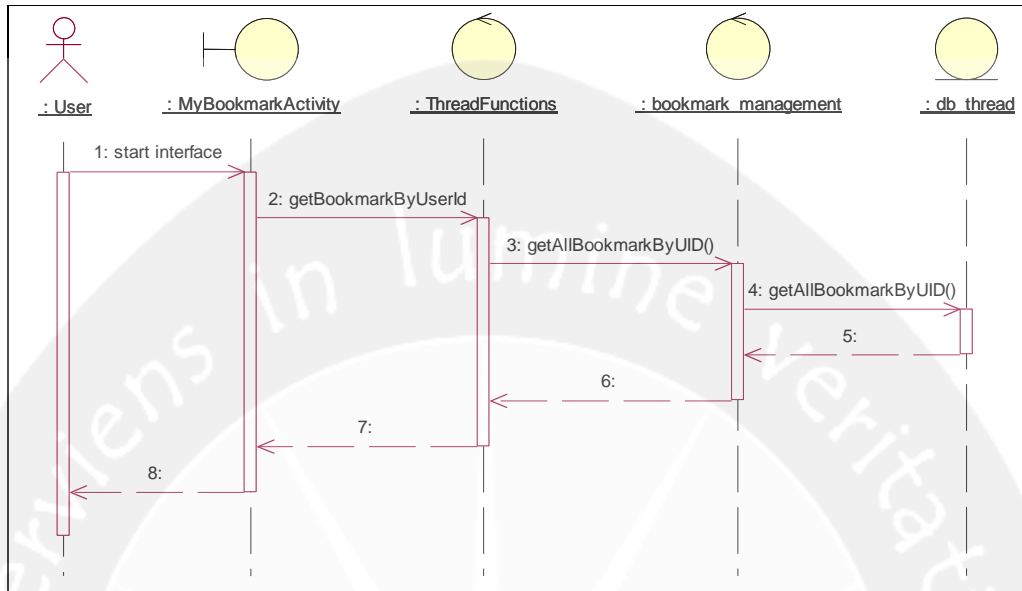
### 2.2.1.2.2 Edit Password



Gambar 2.7 Sequence Diagram : Edit Password

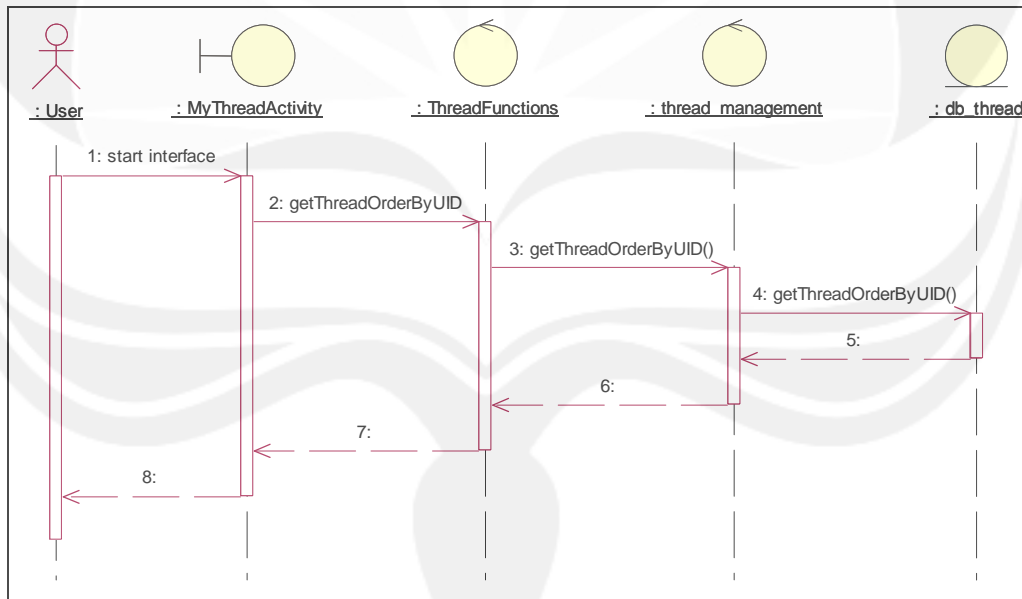


### 2.2.1.2.3 Display Bookmark User



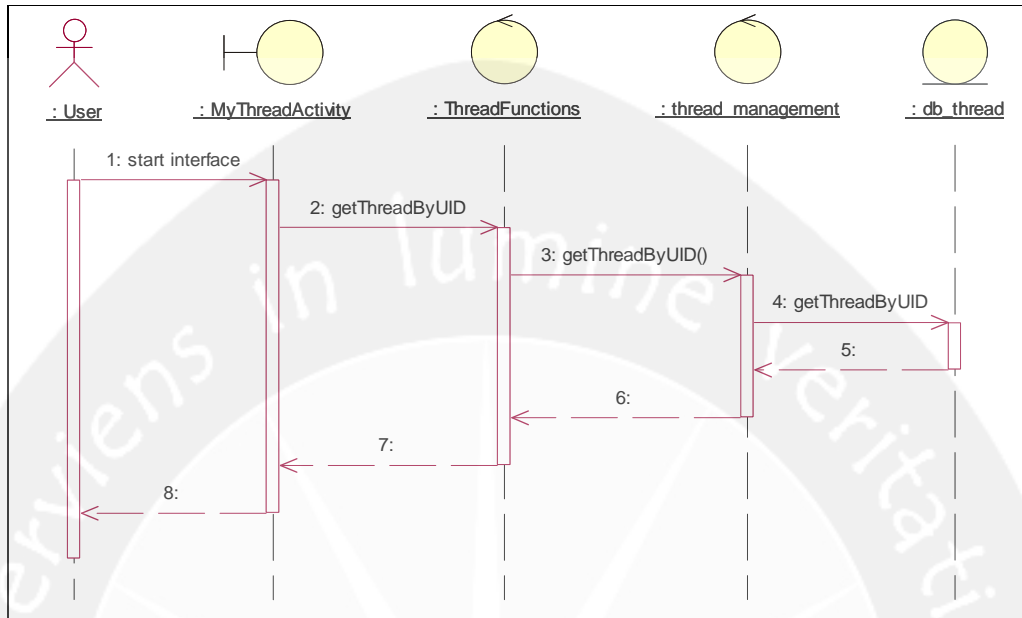
Gambar 2.8 Sequence Diagram : Display Bookmark User

### 2.2.1.2.4 Display Thread Order User



Gambar 2.9 Sequence Diagram : Display Thread Order User

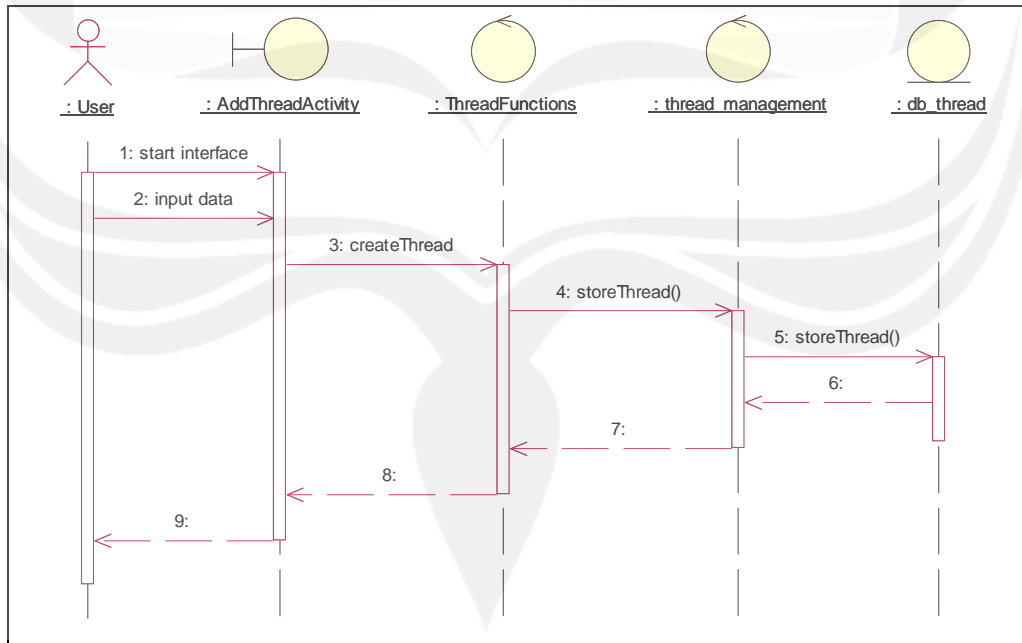
### 2.2.1.2.5 Display Thread User



Gambar 3.10 Sequence Diagram : Display Thread User

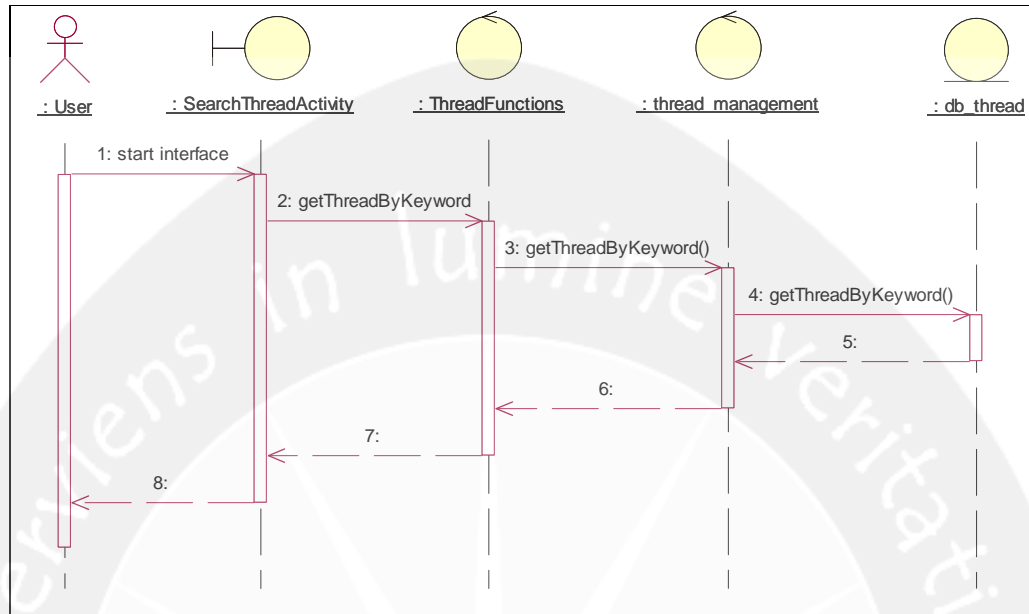
### 2.2.1.3 Pengelolaan Data Thread

#### 2.2.1.3.1 Entri Data Thread



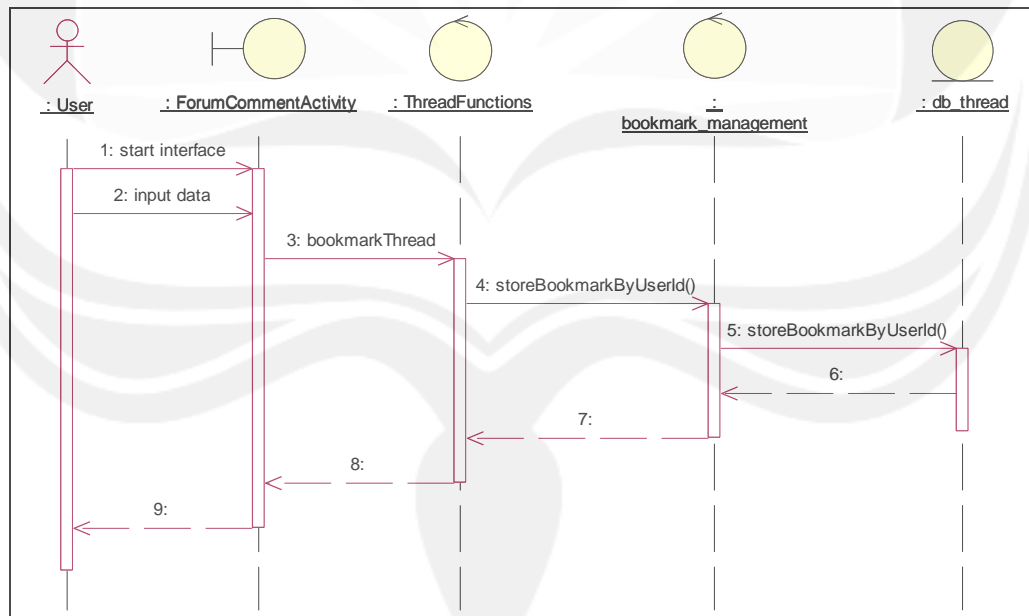
Gambar 3.11 Sequence Diagram : Entri Data Thread

### 2.2.1.3.2 Search Data Thread



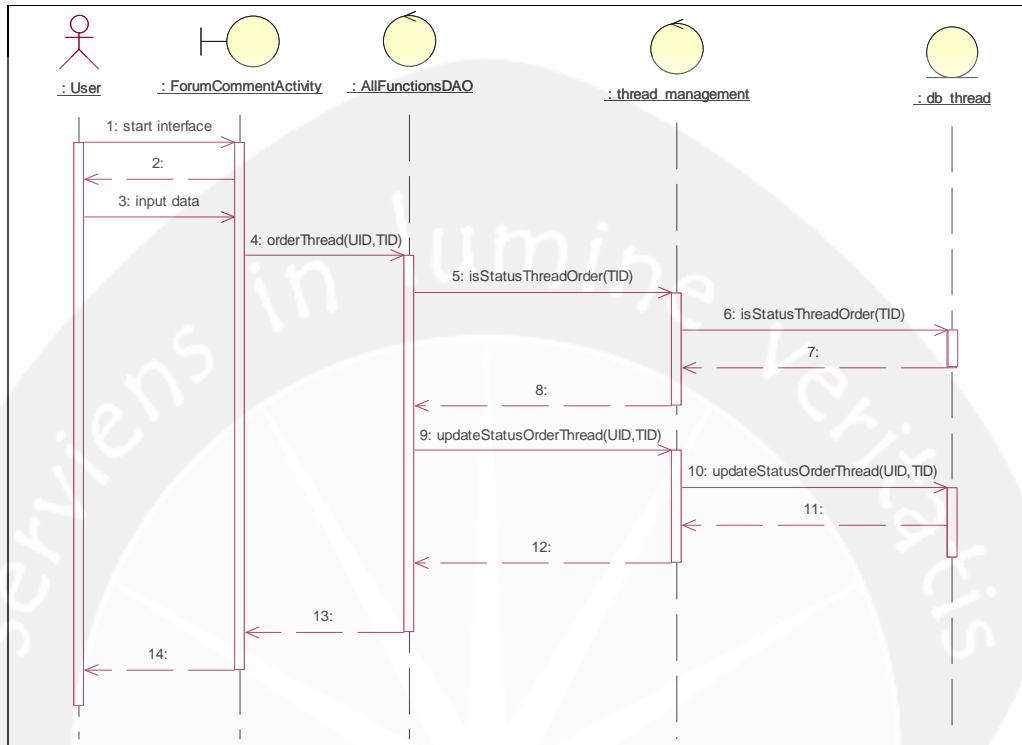
Gambar 3.12 Sequence Diagram : Search Data Thread

### 2.2.1.3.3 Bookmark Thread



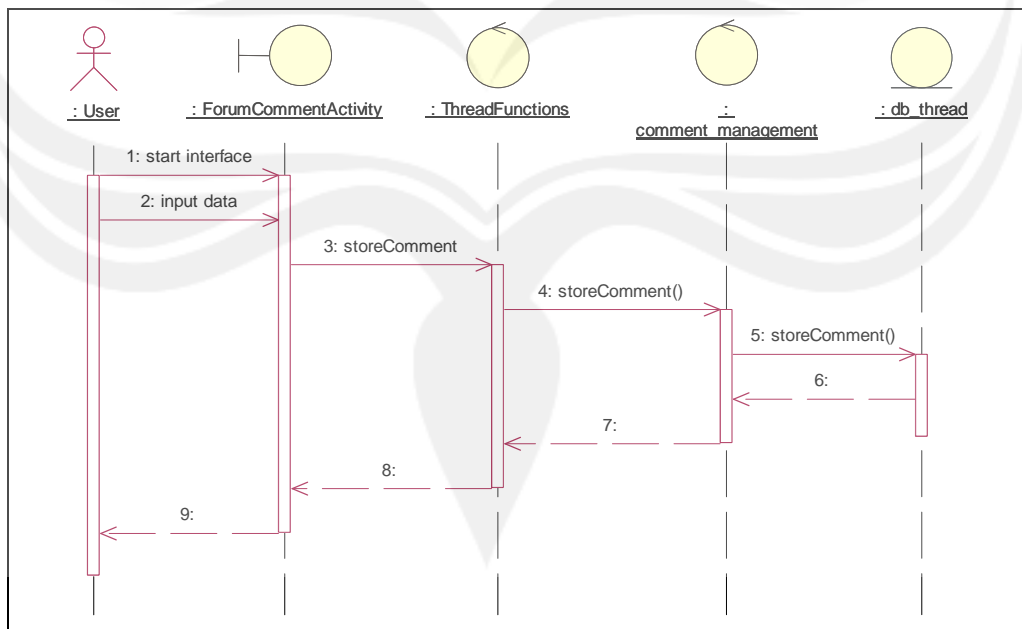
Gambar 3.13 Sequence Diagram : Bookmark Thread

### 2.2.1.3.4 Order Thread



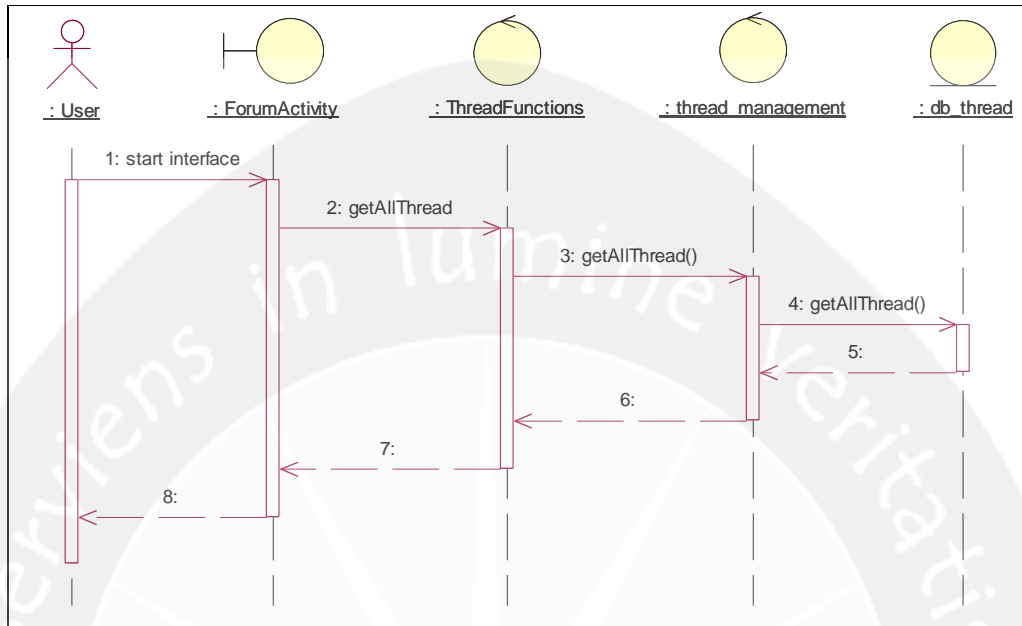
Gambar 3.14 Sequence Diagram : Order Thread

### 2.2.1.3.5 Entri Data Comment Thread



Gambar 3.15 Sequence Diagram : Entri Data Comment Thread

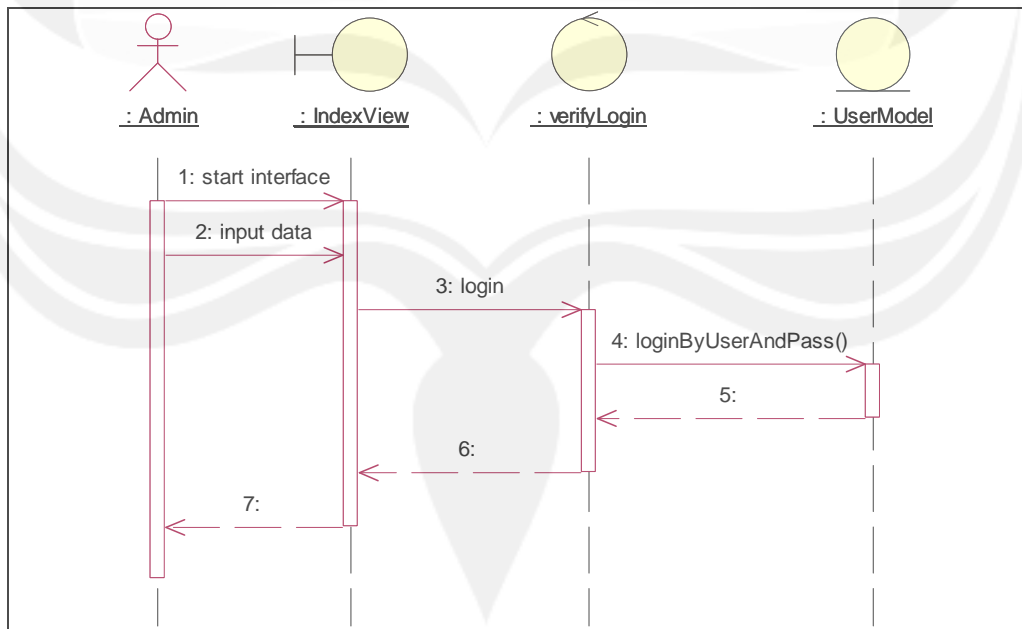
### 2.2.1.3.6 Display Data Thread



Gambar 3.16 Sequence Diagram : Display Data Thread

### 2.2.1.4 Autentikasi (Admin dari FORJUBEL)

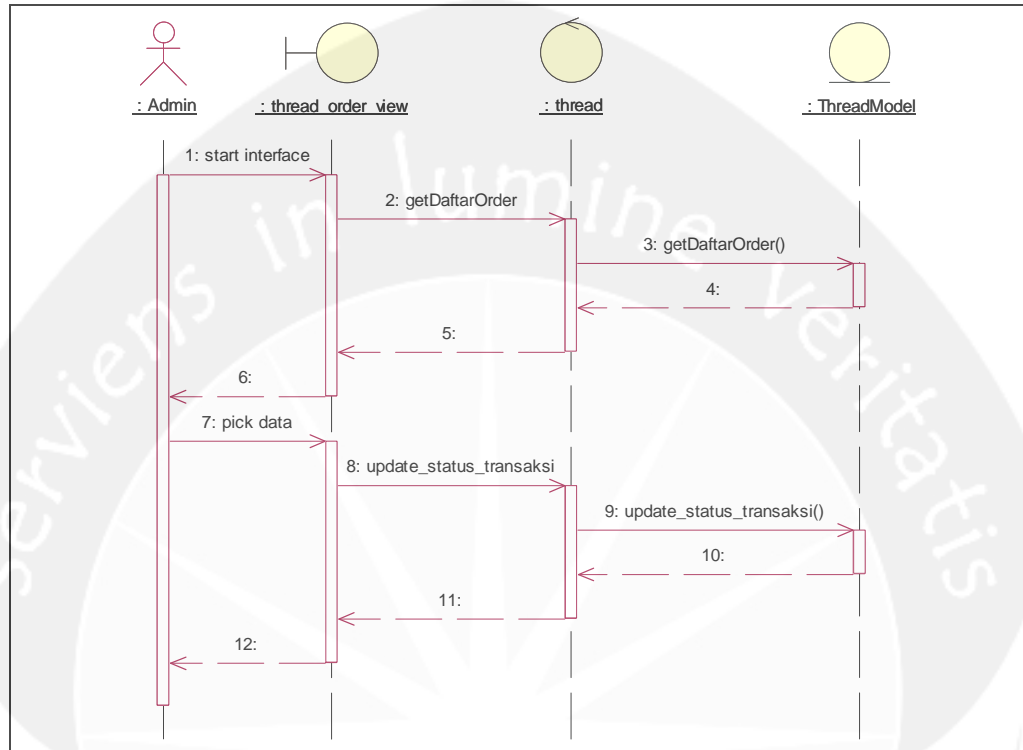
#### 2.2.1.4.1 Login Web



Gambar 3.17 Sequence Diagram : Login Web

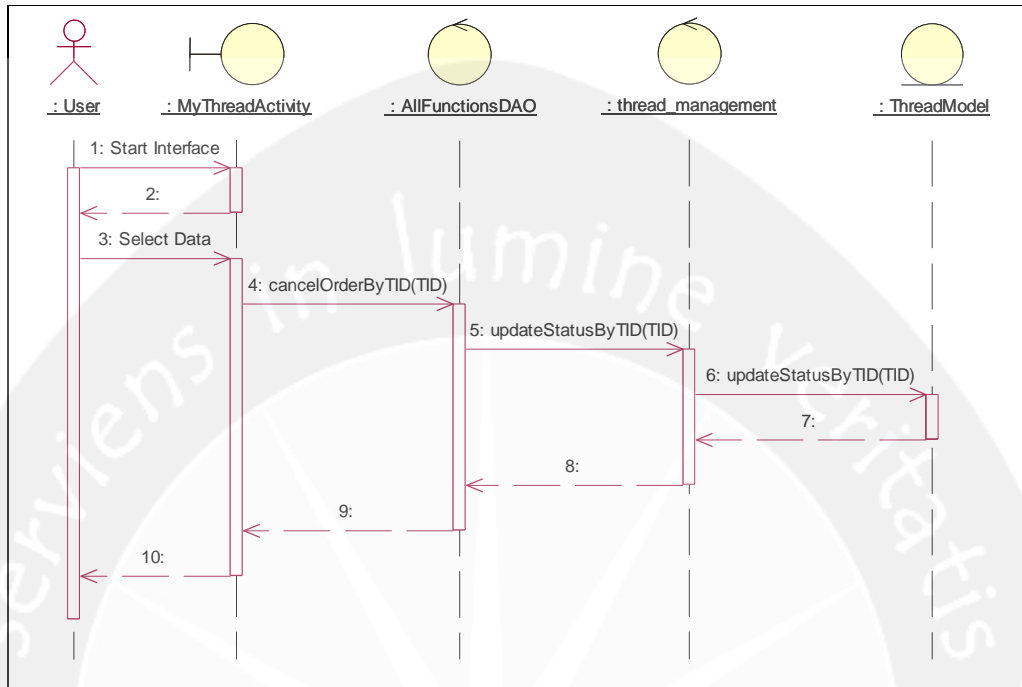
## 2.2.1.5 Pengelolaan Data Status Transaksi Thread

### 2.2.1.5.1 Update Status Transaksi Order



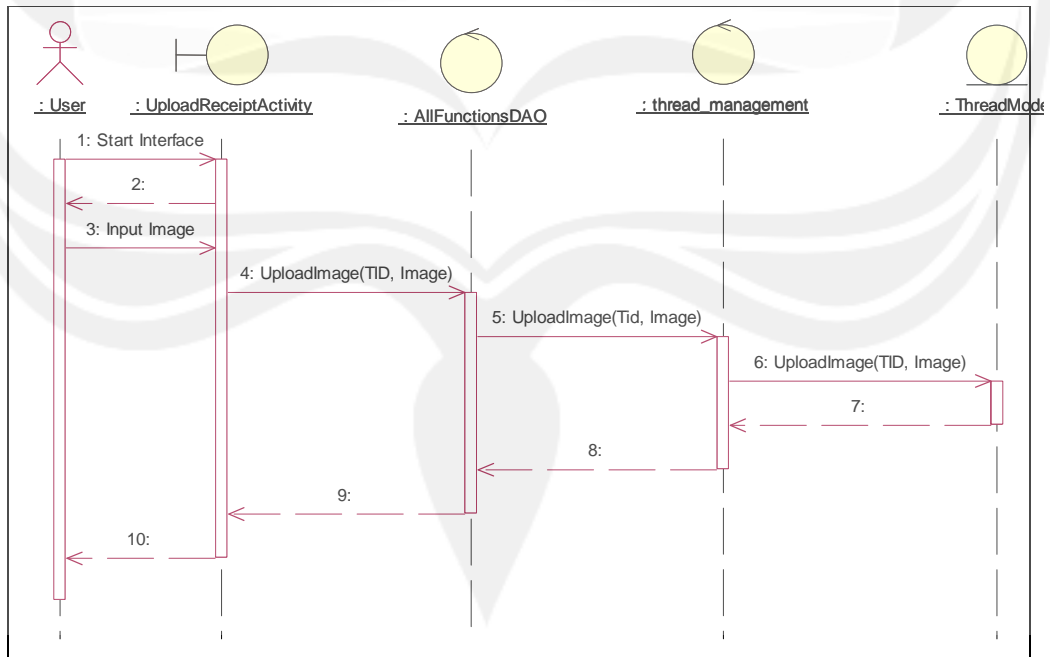
Gambar 3.18 Sequence Diagram : Update Status Transaksi Order

### 2.2.1.5.2 Upload Receipt



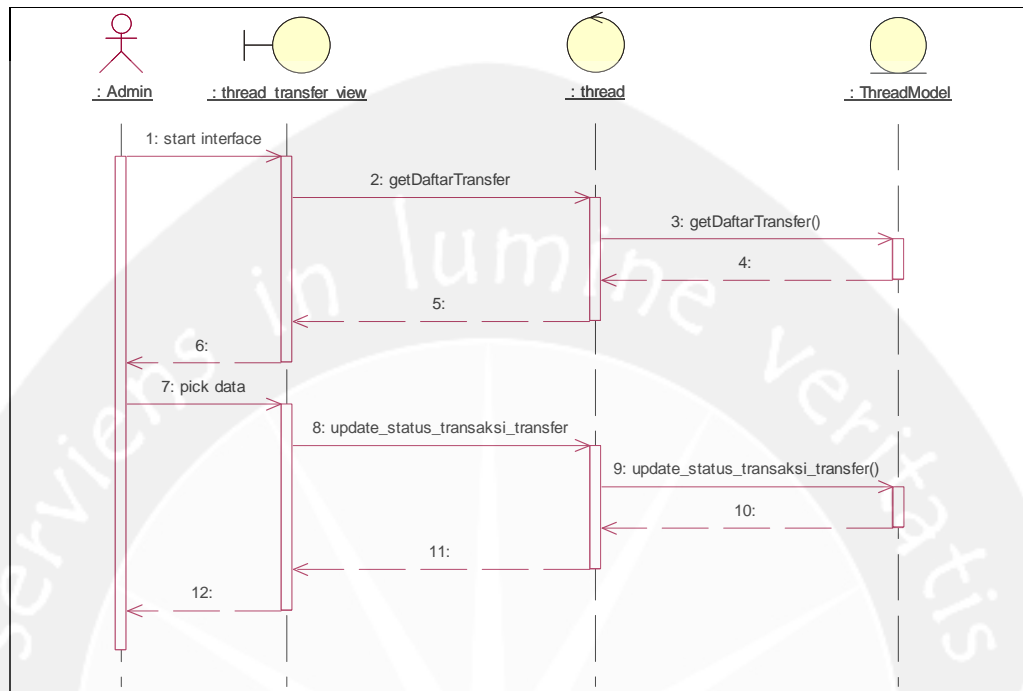
Gambar 3.19 Sequence Diagram : Cancel Order

### 2.2.1.5.3 Upload Receipt



Gambar 3.20 Sequence Diagram : Upload Receipt

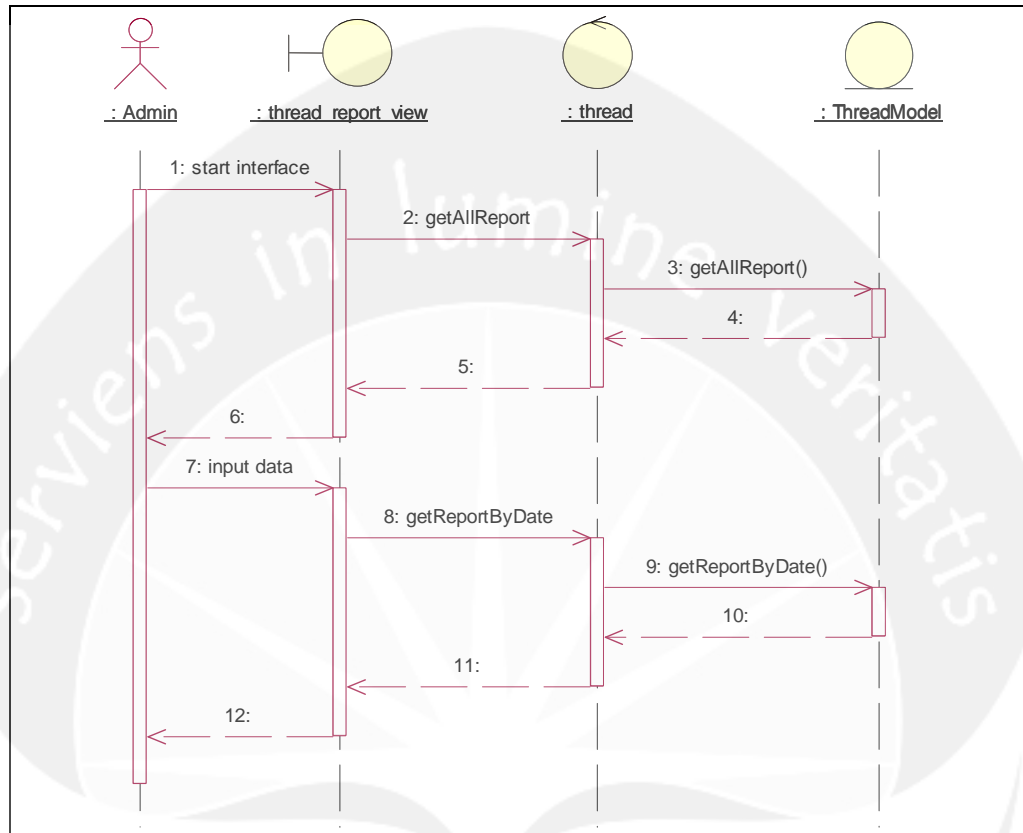
### 2.2.1.5.4 Update Status Transaksi Transfer



Gambar 3.21 Sequence Diagram : Update Status Transaksi Transfer



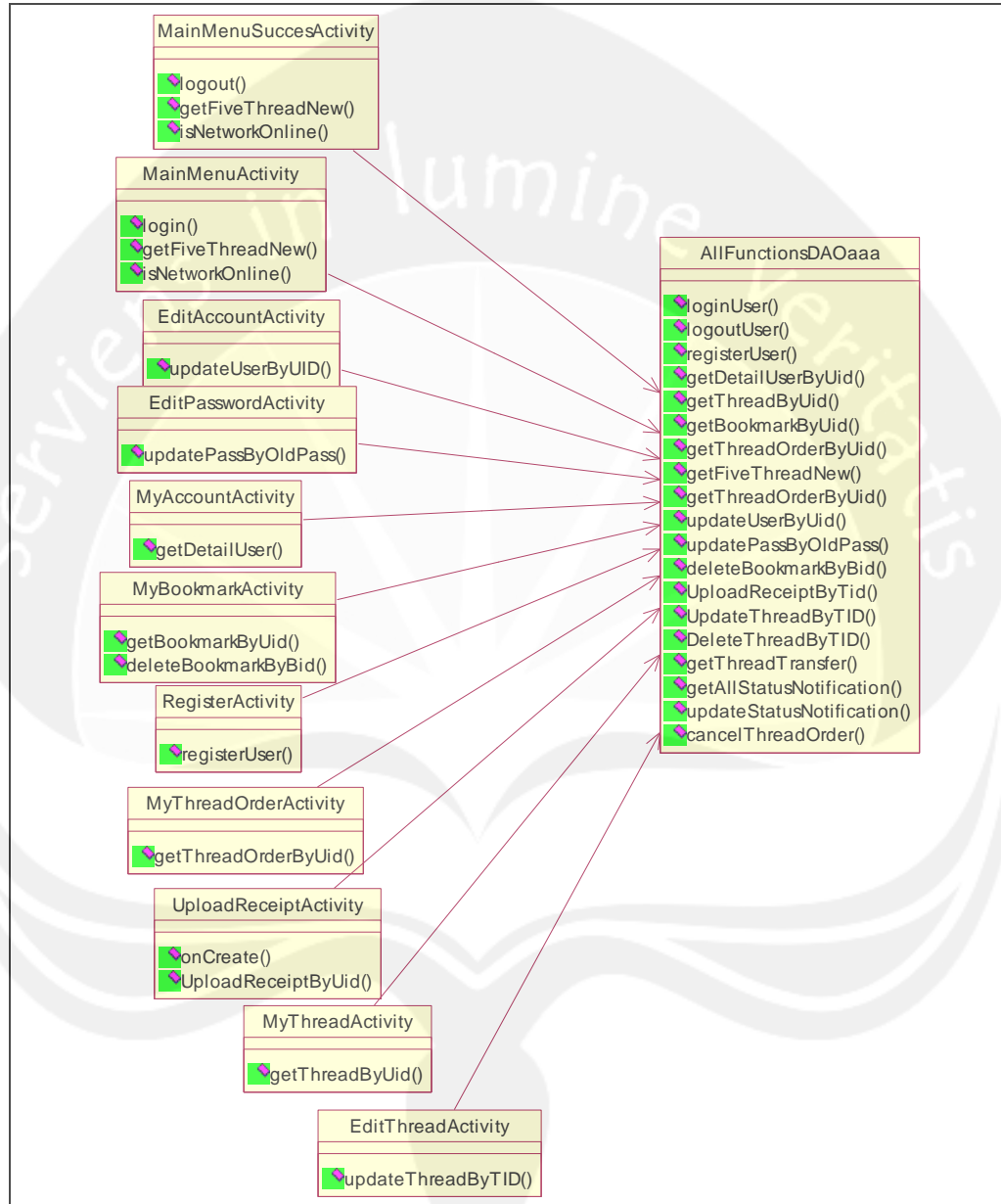
### 2.2.1.5.5 Display Report Thread Delivered

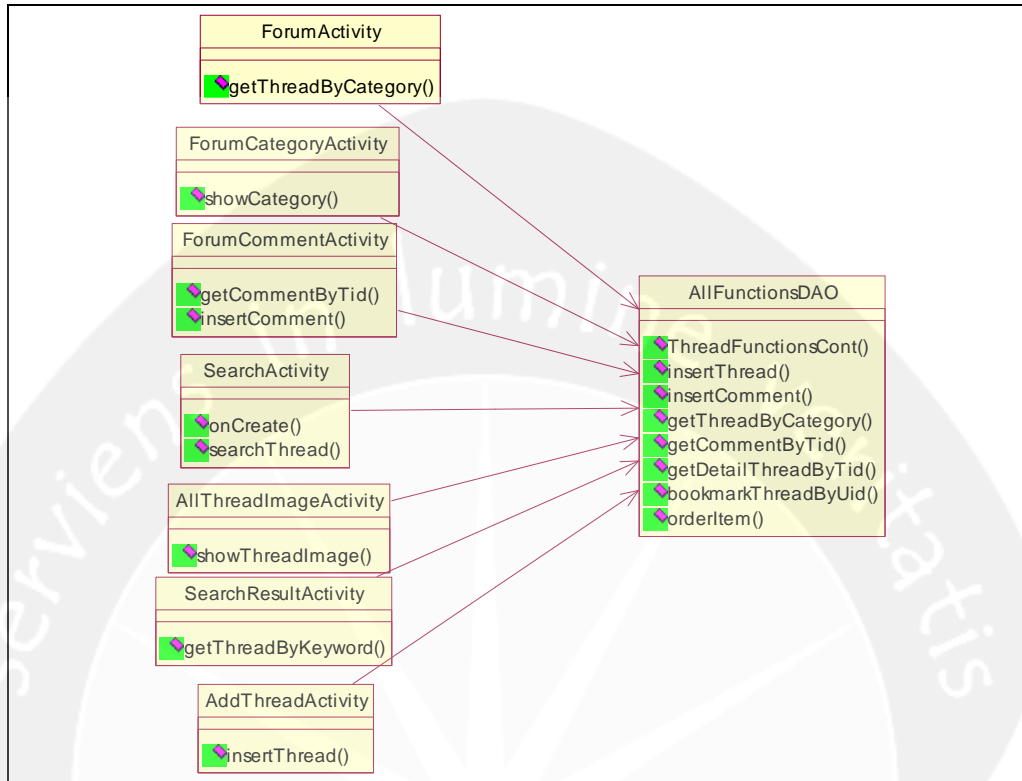


Gambar 3.22 Sequence Diagram : Display Report Thread Sold

## 2.2.2 Class Diagram

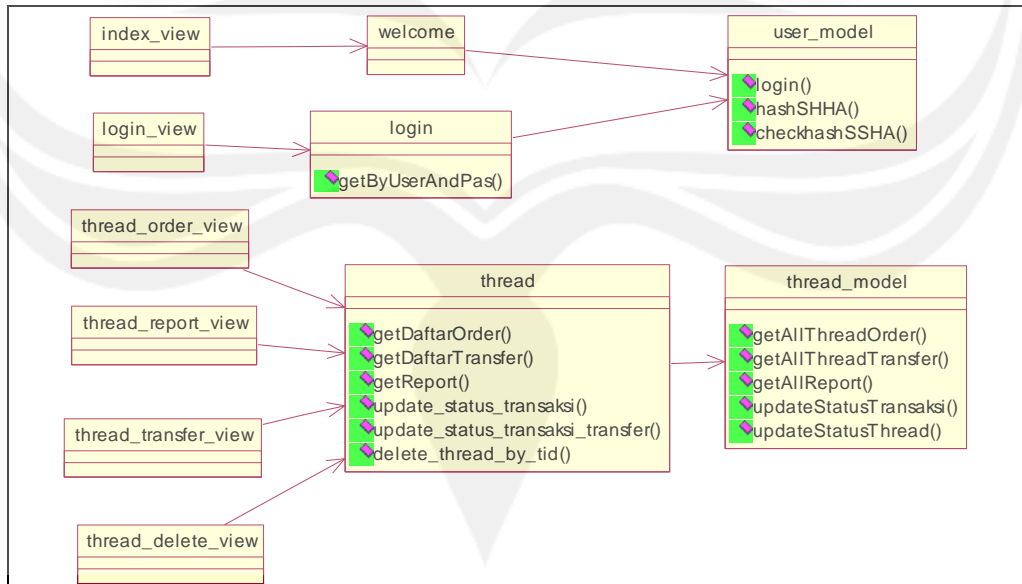
### 2.2.2.1 Mobile Diagram





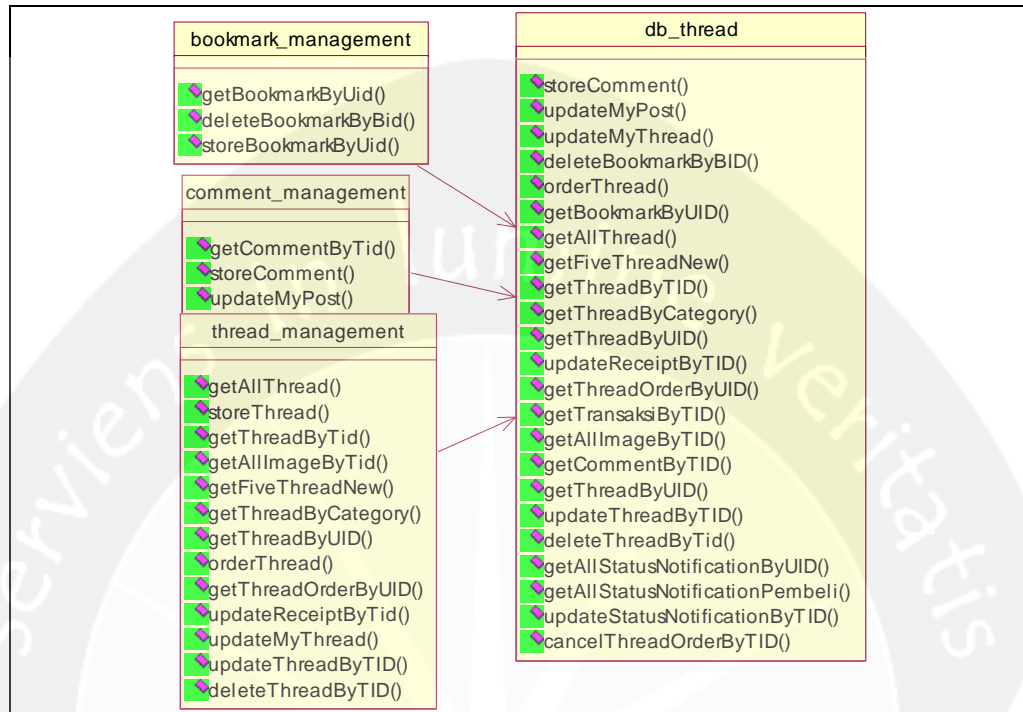
Gambar 3.23 Mobile Class Diagram

### 2.2.2.2 Web Diagram



Gambar 3.24 Web Class Diagram

### 2.2.2.3 Web Service Diagram



Gambar 3.25 Web Service Class Diagram

### 2.2.3 Spesifikasi Deskripsi Kelas Diagram

#### 2.2.3.1 Spesifikasi Design Kelas Login Mobile

<b>MainActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<p>+MainMenuActivity() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+loginUser(username,password) Operasi ini digunakan untuk membandingkan username dan password pada data server. Mengembalikan nilai true jika sama dan false jika tidak sama</p> <p>+get5ThreadNew() Operasi ini digunakan untuk mengambil lima thread terbaru yang telah dibuat.</p> <p>+isNetworkOnline()</p>	

Operasi ini digunakan untuk mengecek jika jaringan tersedia  
 +cekUserLogin()  
 Operasi ini digunakan untuk mengecek apakah user menggunakan fungsi keep me login saat login. Jika ya maka user akan langsung masuk kedalam sistem.  
 +getStatusThreadNotificationByUID()  
 Operasi ini digunakan untuk mengambil semua status thread notification berdasarkan user id.  
 +getStatusThreadNotificationPembeli()  
 Operasi ini digunakan untuk mengambil semua status thread notifikasi pembeli berdasarkan user id.

#### 2.2.3.2 Spesifikasi Design Kelas RegisterActivity

<b>RegisterActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+RegisterActivity() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +insertDataUser() Operasi ini digunakan untuk menginputkan data user kedalam database.	

#### 2.2.3.3 Spesifikasi Design Kelas MyAccountActivity

<b>MyAccountActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
+MyAccountActivity() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +getDetailUserByUid() Operasi ini digunakan untuk mengambil detail user dari database.	

#### 2.2.3.4 Spesifikasi Design Kelas EditPasswordActivity

<b>EditPasswordActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<b>+EditPasswordActivity()</b> Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
<b>+updatePasswordByOldPassword()</b> Operasi ini digunakan untuk mengubah password user.	

#### 2.2.3.5 Spesifikasi Design Kelas EditAccountActivity

<b>EditAccountActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<b>+EditAccountActivity()</b> Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
<b>+updateUserAccountByUid()</b> Operasi ini digunakan untuk mengubah data akun user di dalam database.	

#### 2.2.3.6 Spesifikasi Design Kelas MyThreadActivity

<b>MyThreadActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<b>+MyThreadActivity()</b> Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.	
<b>+getThreadByUid()</b> Operasi ini digunakan untuk mengambil data thread berdasarkan user id di dalam database.	
<b>+cancelThreadOrderByTID()</b> Operasi ini digunakan untuk membatalkan thread order berdasarkan thread id.	

### 2.2.3.7 Spesifikasi Design Kelas MyThreadOrderActivity

<b>MyThreadOrderActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+MyThreadOrderActivity() Default konstruktor, digunakan untuk inialisasi semua atribut dari kelas ini. +getThreadOrderById() Operasi ini digunakan untuk mengambil data thread pesanan user berdasarkan user id di dalam database.</pre>	

### 2.2.3.8 Spesifikasi Design Kelas MyBookmarkActivity

<b>MyBookmarkActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+MyBookmarkActivity() Default konstruktor, digunakan untuk inialisasi semua atribut dari kelas ini. +getBookmarkById() Operasi ini digunakan untuk mengambil data bookmark berdasarkan user id di dalam database. +deleteBookmarkById() Operasi ini digunakan untuk menghapus data bookmark user berdasarkan user id dan bookmark id.</pre>	

### 2.2.3.9 Spesifikasi Design Kelas AddThreadActivity

<b>AddThreadActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+AddThreadActivity() Default konstruktor, digunakan untuk inialisasi semua atribut dari kelas ini. +AddDataThread() Operasi ini digunakan untuk memasukkan data thread.</pre>	

### 2.2.3.10 Spesifikasi Design Kelas ThreadActivity

<b>ThreadActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<p>+ThreadActivity() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+getThreadByCategory() Operasi ini digunakan untuk mengambil semua data thread dari database berdasarkan kategori.</p>	

### 2.2.3.11 Spesifikasi Design Kelas DetailThreadActivity

<b>DetailThreadActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<p>+DetailThreadActivity() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <p>+getDetailThreadByTid() Operasi ini digunakan untuk mengambil detail thread dari database berdasarkan thread id.</p> <p>+getCommentByTid() Operasi ini digunakan untuk mengambil komentar thread dari database berdasarkan thread id.</p> <p>+orderThread(user_id) Operasi ini digunakan untuk memesan barang pada suatu thread.</p> <p>+bookmarkThread(user_id) Operasi ini digunakan untuk menandai suatu thread.</p>	

### 2.2.3.12 Spesifikasi Design Kelas SearchActivity

<b>SearchResultActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<p>+SearchResultActivity()</p>	



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

```
+insertSearchKeyword(title, category, price, location)
```

Operasi ini digunakan untuk menginputkan kata kunci pencarian.

#### 2.2.3.13 Spesifikasi Design Kelas SearchResultActivity

<b>SearchResultActivity</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+SearchResultActivity()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <pre>+getThreadByKeyword(title, category, price, location)</pre> <p>Operasi ini digunakan untuk mencari data thread tertentu didalam database berdasarkan keyword pencarian.</p>	

#### 2.2.3.14 Spesifikasi Design Kelas Index

<b>Index</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+Index()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini.</p> <pre>+verifyUser(username,password)</pre> <p>Operasi ini digunakan untuk membandingkan username dan password yang dimasukkan oleh user dengan hasil enkripsi. Akan mengembalikan nilai true jika sama dan false jika tidak sama.</p>	

#### 2.2.3.15 Spesifikasi Design Kelas thread\_order\_view

<b>thread_order_view</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+thread_order_view()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua</p>	

atribut dari kelas ini.

+getDaftarOrder()  
Operasi ini digunakan untuk menampilkan semua data thread yang statusnya order.

+update\_status\_transaksi()  
Operasi ini digunakan untuk mengupdate semua data status order.

#### 2.2.3.16 Spesifikasi Design Kelas thread\_transfer\_view

<b>thread_transfer_view</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+thread_transfer_view() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +getDaftarTransfer() Operasi ini digunakan untuk menampilkan semua data thread yang statusnya transfer. +update_status_transaksi_transfer() Operasi ini digunakan untuk mengupdate semua data status transfer.</pre>	

#### 2.2.3.17 Spesifikasi Design Kelas thread\_report\_view

<b>thread_report_view</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+thread_report_view() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +getReport() Operasi ini digunakan untuk menampilkan semua thread. +getReportByDate() Operasi ini digunakan untuk menampilkan semua data thread tertentu sesuai dengan inputan bulan.</pre>	

### 2.2.3.18 Spesifikasi Design Kelas thread\_delete\_view

<b>thread_delete_view</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+thread_delete_view() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +getAllThreadNew() Operasi ini digunakan untuk menampilkan semua thread yang statusnya new. +deleteThreadByTID() Operasi ini digunakan untuk menghapus data thread berdasarkan thread id.</pre>	

### 2.2.3.19 Spesifikasi Design Kelas AllFunctionsDAO

<b>AllFunctionsDAO</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+ AllFunctionsDAO() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +loginUser(name, password) Operasi ini digunakan untuk membandingkan username dan password di dalam database server. Akan mengembalikan nilai true jika sama dan false jika tidak sama. +registerUser(input data) Operasi ini digunakan untuk menginputkan data user kedalam database server +getDetailUserByUid(uid) Operasi ini digunakan untuk mengambil data-data user berdasarkan user id dari database. +updateUser(uid) Operasi ini digunakan untuk memperbarui data akun user</pre>	

berdasarkan user id didalam database.

+updatePassByOldPass()

Operasi ini digunakan untuk memperbarui password user di dalam database.

+EditThreadByTID()

Operasi ini digunakan untuk mengubah data thread di dalam database server.

+deleteThreadByTID()

Operasi ini digunakan untuk menghapus data thread di dalam database server.

+getThreadByUid()

Operasi ini digunakan untuk mengambil thread berdasarkan user id di dalam database server.

+getBookmarkByUid()

Operasi ini digunakan untuk mengambil data bookmark berdasarkan user id di dalam database.

+logoutUser(uid)

Operasi digunakan untuk menghilangkan session dari user yang login.

+isUserLoggedIn(uid)

Operasi digunakan untuk menyimpan session dari user yang login.

#### 2.2.3.20 Spesifikasi Design Kelas ThreadFunctions

<b>ThreadFunctions</b>	<b>&lt;&lt;control&gt;&gt;</b>
+ThreadFunctions() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+insertThread() Operasi ini digunakan untuk menginputkan data thread	

kedalam database server.

+getAllThread()

Operasi ini digunakan untuk mengambil semua thread di dalam database server.

+getThreadByCategory()

Operasi ini digunakan untuk mengambil thread berdasarkan kategori di dalam database server.

+getThreadByTid()

Operasi ini digunakan untuk mengambil thread berdasarkan thread id di dalam database server.

+getThreadByKeyword()

Operasi ini digunakan untuk mengambil thread berdasarkan kata kunci yang diinputkan user.

+insertComment()

Operasi ini digunakan untuk menginputkan data komentar kedalam database server.

+getCommentByTid()

Operasi ini digunakan untuk mengambil data comment berdasarkan thread id di dalam database.

+bookmarkThread()

Operasi ini digunakan untuk menyimpan data bookmark kedalam database.

+deleteBookmark()

Operasi ini digunakan untuk menghapus data bookmark didalam database.

+orderItem()

Operasi ini digunakan untuk memesan barang pada thread.

#### 2.2.3.21 Spesifikasi Design Kelas user\_management

<b>user_management</b>	<b>&lt;&lt;control&gt;&gt;</b>

```
+user_management()
```

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

```
+getByUserAndPass()
```

Operasi ini digunakan untuk membandingkan username dan password. Akan mengembalikan nilai true jika sama dan false jika tidak sama

```
+storeUser()
```

Operasi ini digunakan untuk menginputkan data user kedalam database

```
+isUserExisted()
```

Operasi ini digunakan untuk mengecek jika username sudah ada didalam database

```
+getUserDetailByUID()
```

Operasi ini digunakan untuk mengambil detail dari user didalam database berdasarkan user id.

#### 2.2.3.22 Spesifikasi Design Kelas update\_user\_management

<b>update_user_management</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+update_user_management()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <pre>+updateUser()</pre> <p>Operasi ini digunakan untuk memperbarui data user didalam database</p> <pre>+updatePassByUID()</pre> <p>Operasi ini digunakan untuk memperbarui password user didalam database berdasarkan user id.</p>	

#### 2.2.3.23 Spesifikasi Design Kelas thread\_management

<b>thread_management</b>	<b>&lt;&lt;control&gt;&gt;</b>	
Program Studi Teknik Informatika	DPPL – FORJUBEL	34/ 55
<p>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</p>		

+thread\_management()

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

+getAllThread()

Operasi ini digunakan untuk mengambil semua thread didalam database

+storeThread(input data)

Operasi ini digunakan untuk menginputkan thread kedalam database.

+updateMyThread()

Operasi ini digunakan untuk menambahkan jumlah thread user jika user berhasil memasukkan thread.

+getThreadByTID(thread\_id) :bool

Operasi ini digunakan untuk mengambil thread berdasarkan thread id didalam database

+getAllThreadImageByTID(thread\_id) : bool

Operasi ini digunakan untuk mengambil semua image didalam database berdasarkan thread id.

+getStatusThreadNotificationByUID()

Operasi ini digunakan untuk mengambil semua status thread notification berdasarkan user id.

+getStatusThreadNotificationPembeli()

Operasi ini digunakan untuk mengambil semua status thread notifikasi pembeli berdasarkan user id.

+updateStatusNotif(String tid, String status)

Operasi ini digunakan untuk mengupdate status notifikasi berdasarkan thread id

+cancelThreadOrderByTID()

Operasi ini digunakan untuk membatalkan thread order berdasarkan thread id.

#### 2.2.3.24 Spesifikasi Design Kelas `comment_management`

<code>comment_management</code>	<code>&lt;&lt;control&gt;&gt;</code>
<pre>+comment_management() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +storeComment() Operasi ini digunakan untuk menginputkan komentar kedalam database +getCommentByTID() Operasi ini digunakan untuk mengambil komentar sesuai thread id didalam database +updateMyPost() Operasi ini digunakan untuk menambahkan jumlah post jika user berhasil memasukkan komentar. +orderThread() Operasi ini digunakan untuk menginputkan order pada status transaksi didalam database.</pre>	

#### 2.2.3.25 Spesifikasi Design Kelas `bookmark_management`

<code>bookmark_management</code>	<code>&lt;&lt;control&gt;&gt;</code>
<pre>+bookmark_management() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +storeBookmarkByUID(user_id) :bool Operasi ini digunakan untuk menginputkan bookmark kedalam database. +deleteBookmarkByBID(bookmark_id) :bool Operasi ini digunakan untuk menghapus bookmark didalam database.</pre>	



```
+getAllBookmark() :bool
```

Operasi ini digunakan untuk mengambil semua bookmark didalam database.

#### 2.2.3.26 Spesifikasi Design Kelas verifylogin

<b>Verifylogin</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+verifylogin() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +login(user_id) :bool Operasi ini digunakan untuk membandingkan username dan password. Akan mengembalikan nilai true jika sama dan false jika tidak sama.</pre>	

#### 2.2.3.27 Spesifikasi Design Kelas thread

<b>Thread</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+thread() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDaftarOrder() Operasi ini digunakan untuk mengambil data thread dengan status order dari database. +getDaftarTransfer() Operasi ini digunakan untuk mengambil data thread dengan status transfer dari database. +getReport() Operasi ini digunakan untuk mengambil data thread dengan dari database berdasarkan inputan bulan. +updateStatusTransaksi()</pre>	

Operasi ini digunakan untuk memperbarui status transaksi order menjadi transfer.

```
+updateStatusTransaksiTransfer()
```

Operasi ini digunakan untuk memperbarui status transaksi transfer menjadi sold.

```
+logout()
```

Operasi ini digunakan untuk menghilangkan session dari user.

### 2.2.3.28 Spesifikasi Design Kelas DB\_User

<b>DB_User</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<pre>+DB_User()</pre> <p>Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <pre>+getByUserAndPass()</pre> <p>Operasi ini digunakan untuk membandingkan username dan password. Akan mengembalikan nilai true jika sama dan false jika tidak sama</p> <pre>+storeUser()</pre> <p>Operasi ini digunakan untuk menginputkan data user kedalam database</p> <pre>+isUserExisted()</pre> <p>Operasi ini digunakna untuk mengecek jika username sudah ada didalam database.</p> <pre>+getUserDetailByUID()</pre> <p>Operasi ini digunakan untuk mengambil detail dari user didalam database berdasarkan user id.</p> <pre>+updateUser()</pre> <p>Operasi ini digunakan untuk memperbarui data user didalam database</p>	

+updatePassByUID()

Operasi ini digunakan untuk memperbarui password user didalam database berdasarkan user id.

+hashSSHA()

Operasi ini digunakan untuk mengenkripsi password user.

#### 2.2.3.29 Spesifikasi Design Kelas DB\_Thread

DB_Thread	<<entity>>
<p>+DB_Thread() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</p> <p>+getAllThread() Operasi ini digunakan untuk mengambil semua thread didalam database</p> <p>+storeThread(input data) Operasi ini digunakan untuk menginputkan thread kedalam database.</p> <p>+editThreadByTID() Operasi ini digunakan untuk mengubah data thread di dalam database.</p> <p>+deleteThreadByTID() Operasi ini digunakan untuk menghapus data thread di dalam database.</p> <p>+updateMyThread() Operasi ini digunakan untuk menambahkan jumlah thread user jika user berhasil memasukkan thread.</p> <p>+getThreadByTID(thread_id) :bool Operasi ini digunakan untuk mengambil thread berdasarkan thread id didalam database</p> <p>+getAllThreadImageByTID(thread_id) : bool</p>	

Operasi ini digunakan untuk mengambil semua image didalam database berdasarkan thread id.

+storeComment()

Operasi ini digunakan untuk menginputkan komentar kedalam database

+getCommentByTID()

Operasi ini digunakan untuk mengambil komentar sesuai thread id didalam database

+updateMyPost()

Operasi ini digunakan untuk menambahkan jumlah post jika user berhasil memasukkan komentar.

+orderThread()

Operasi ini digunakan untuk menginputkan order pada status transaksi didalam database.

+storeBookmarkByUID(user\_id) :bool

Operasi ini digunakan untuk menginputkan bookmark kedalam database.

+deleteBookmarkByBID(bookmark\_id) :bool

Operasi ini digunakan untuk menghapus bookmark didalam database.

+getAllBookmark() :bool

Operasi ini digunakan untuk mengambil semua bookmark didalam database.

+getStatusThreadNotificationByUID()

Operasi ini digunakan untuk mengambil semua status thread notification berdasarkan user id.

+getStatusThreadNotificationPembeli()

Operasi ini digunakan untuk mengambil semua status thread notifikasi pembeli berdasarkan user id.

+updateStatusNotif(String tid, String status)

Operasi ini digunakan untuk mengupdate status

notifikasi berdasarkan thread id

+cancelThreadOrderByTID()

Operasi ini digunakan untuk membatalkan thread order berdasarkan thread id.

#### 2.2.3.30 Spesifikasi Design Kelas user\_model

<b>user_model</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<b>+user_model()</b> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
<b>+login()</b> Operasi ini digunakan untuk membandingkan username dan password. Akan mengembalikan nilai true jika sama dan false jika tidak sama.	
<b>+hassSSHA()</b> Operasi ini digunakan untuk mendekripsi password dari user.	

#### 2.2.3.31 Spesifikasi Design Kelas thread\_model

<b>thread_model</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<b>+thread_model()</b> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
<b>+getAllThreadOrder()</b> Operasi ini digunakan untuk mengambil data thread dengan status order dari database.	
<b>+getAllThreadTransfer()</b> Operasi ini digunakan untuk mengambil data thread dengan status transfer dari database.	

```
+getAllThreadReport()
```

Operasi ini digunakan untuk mengambil data thread dengan dari database berdasarkan inputan bulan.

```
+updateStatusTransaksi()
```

Operasi ini digunakan untuk memperbarui status transaksi order menjadi transfer.

```
+updateStatusTransaksiTransfer()
```

Operasi ini digunakan untuk memperbarui status transaksi transfer menjadi sold.

### 3 Perancangan Data

#### 3.1 Dekomposisi Data

##### 3.1.1 Deskripsi Entitas Data Tabel User

Nama	Tipe	Panjang	Keterangan
USER_ID	int	11	User id, primary key
USERNAME	varchar	100	Nama Login pengguna untuk dapat masuk ke dalam sistem
PASSWORD	varchar	100	Password pengguna
AVATAR	varchar	100	Foto pengguna
EMAIL	varchar	100	Email pengguna
TGL_LAHIR	varchar	100	Tanggal lahir pengguna
LOKASI	varchar	100	Lokasi pengguna
JENIS_KELAMIN	varchar	100	Jenis kelamin pengguna
JAWABAN	varchar	100	Jawaban dari pertanyaan
NO_TELP	varchar	100	Nomor telepon pengguna
ROLE	varchar	100	Aturan yang diberikan system untuk membatasi akses pengguna
MY_THREAD	int	11	Jumlah thread yang dibuat
MY_POST	int	11	Jumlah komentar yang dibuat
SALT	Varchar	10	Kode yang diberikan untuk menambah pengecekan login user
CREATED_AT	datetime	-	Tanggal dan waktu pada saat user dibuat
UPDATE_AT	datetime	-	Tanggal dan waktu pada saat user mengupdate

**data akun**

**3.1.2 Deskripsi Entitas Data Tabel Thread**

Nama	Tipe	Panjang	Keterangan
THREAD_ID	int	11	Thread id, primary key
USER_ID	int	11	User id, foreign key
TRANSAKSI_ID	int	11	Transaksi id, foreign key
THREAD_NAMA	varchar	100	Nama atau judul thread
THREAD_STATUS	varchar	100	Status thread
THREAD_STATUS_NOTIF	varchar	100	Status thread notifikasi
HARGA_BRG	varchar	100	Harga barang
LOKASI_BRG	varchar	100	Lokasi barang
KATEGORI_BRG	varchar	100	Kategori barang
IMAGE_BRG	varchar	200	Foto barang
DESK_BRG	varchar	1000	Deskripsi barang
CREATED_AT	datetime	-	Tanggal dan waktu pada saat thread dibuat
UPDATE_AT	datetime	-	Tanggal dan waktu pada saat thread diupdate

**3.1.3 Deskripsi Entitas Data Tabel Komentar**

Nama	Tipe	Panjang	Keterangan
KOMENTAR_ID	int	11	Komentar id, primary key
USER_ID	int	11	User id, foreign key
THREAD_ID	int	11	Thread id, foreign key
KOMENTAR_POST	varchar	100	Komentar user pada sebuah thread
CREATED_AT	datetime		Tanggal dan waktu pada saat komentar dibuat
UPDATED_AT	datetime		Tanggal dan waktu pada saat komentar diupdate

**3.1.4 Deskripsi Entitas Data Tabel Gallery**

Nama	Tipe	Panjang	Keterangan
GALLERY_ID	int	11	Gallery id, primary key

<b>THREAD_ID</b>	int	11	Thread id, foreign key
<b>IMAGE_1</b>	varchar	100	Foto barang 1
<b>IMAGE_2</b>	varchar	100	Foto barang 2
<b>IMAGE_3</b>	varchar	100	Foto barang 3
<b>IMAGE_4</b>	varchar	100	Foto barang 4
<b>IMAGE_5</b>	varchar	100	Foto barang 5

### 3.1.5 Deskripsi Entitas Data Tabel Transaksi

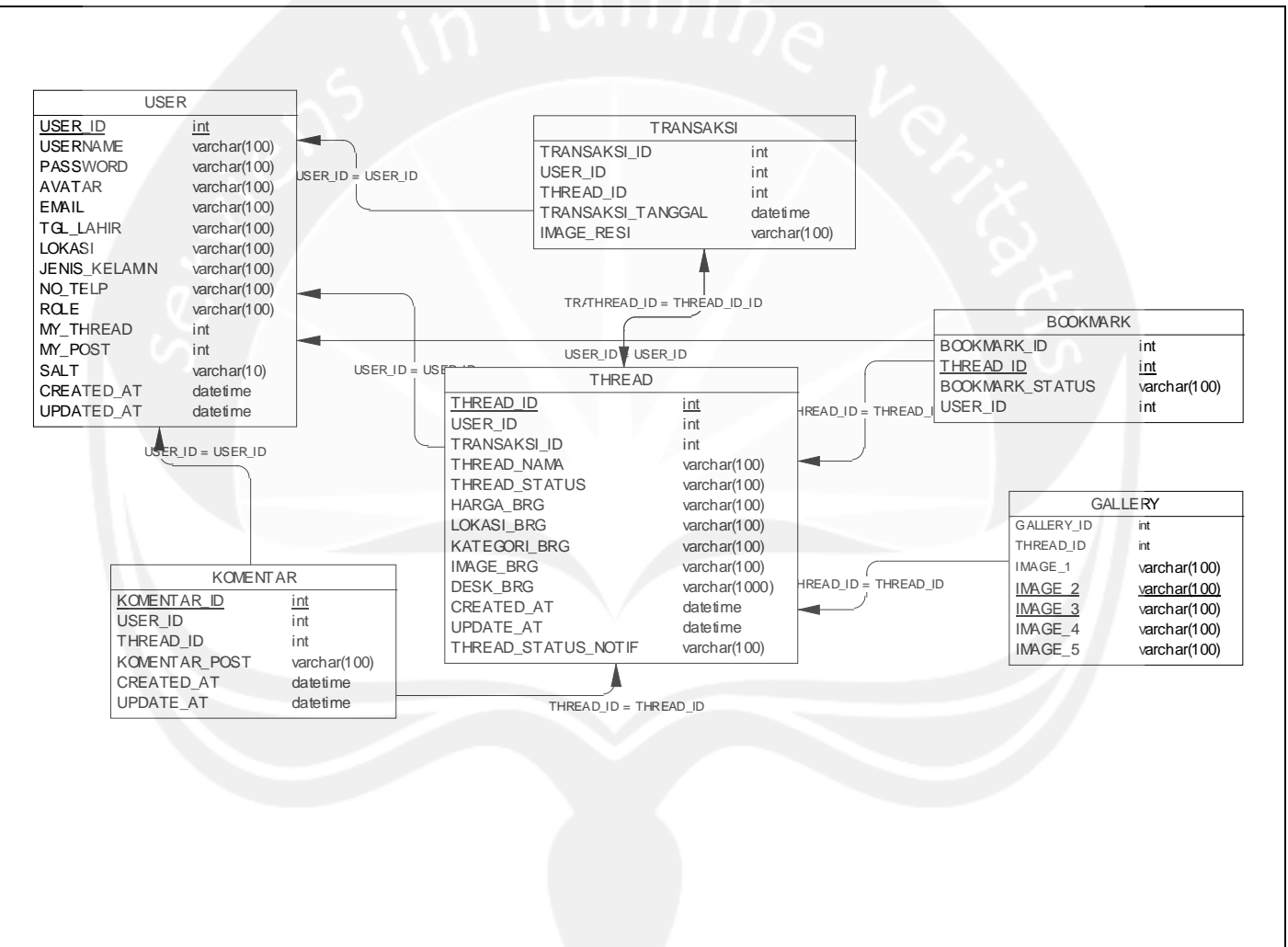
Nama	Tipe	Panjang	Keterangan
<b>TRANSAKSI_ID</b>	int	11	Transaksi id, primary key
<b>USER_ID</b>	int	11	User id, foreign key
<b>THREAD_ID</b>	int	11	Thread id, foreign key
<b>TRANSAKSI_TANGGAL</b>	datetime	-	Tanggal transaksi dibuat
<b>IMAGE_RESI</b>	varchar	100	Gambar resi transaksi

### 3.1.6 Deskripsi Entitas Data Tabel Bookmark

Nama	Tipe	Panjang	Keterangan
<b>BOOKMARK_ID</b>	int	11	Bookmark id, primary key
<b>THREAD_ID</b>	int	11	Thread id, foreign key
<b>BOOKMARK_STATUS</b>	varchar	100	Status untuk menandai sebuah thread yang sedang di bookmark
<b>USER_ID</b>	int	11	User id, foreign key



### 3.2 Physical Data Model



Gambar 3.26 Rancangan Arsitektur : Physical Data Model

## 4 Perancangan Antar Muka

### 4.1 Sketsa UI dan Deskripsinya

#### 4.1.1 Antarmuka Login



Gambar 4.1 Rancangan Antarmuka Login

Antarmuka ini digunakan agar user dapat masuk kedalam sistem FORJUBEL. User dapat memasukkan username dan password, lalu menekan tombol sign in. Jika username atau password salah maka aplikasi akan memberikan pesan peringatan, jika benar maka user akan masuk kedalam menu utama. Jika user belum memiliki account maka user dapat mendaftarkan diri dengan cara menekan label register me.

#### 4.1.2 Antarmuka Pendaftaran User



Gambar 4.2 Rancangan Antarmuka Pendaftaran User

Antarmuka ini digunakan untuk mendaftarkan diri untuk menjadi pengguna aplikasi FORJUBEL, juga dapat menggunakan fungsi-fungsi tertentu misal buat thread, order, dll. User memasukkan data pada text box, lalu menekan tombol register. Jika data tidak sesuai maka aplikasi akan memberikan pesan peringatan, jika benar maka user berhasil mendaftarkan diri.

#### 4.1.3 Antarmuka Edit Password



Gambar 4.3 Rancangan Antarmuka Edit Password

Antarmuka ini digunakan untuk mengganti password. User menginputkan old password, new password dan re password, lalu menekan tombol update. Apabila old password dan new password valid, maka password user

telah terganti dengan sempurna, namun bila tidak valid, maka sistem akan meminta user memasukkan kembali old password dan new password.

#### 4.1.4 Antarmuka Edit Account User



Gambar 4.4 Rancangan Antarmuka Edit Account Password

Antarmuka ini digunakan untuk mengubah data diri user. User menginputkan data diri, lalu menekan tombol save. Apabila inputan sesuai, maka data akun akan terganti dengan sempurna, namun bila tidak valid, maka sistem akan meminta user untuk memasukkan kembali data diri.

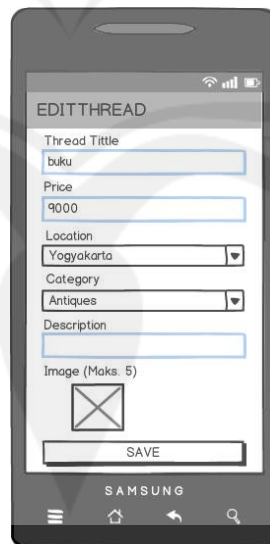
#### 4.1.5 Antarmuka Buat Thread



Gambar 4.5 Rancangan Antarmuka Buat Thread

Antarmuka ini digunakan untuk memasukkan data thread baru. User menginputkan data thread yang akan dibuat, kemudian user menekan tombol create. Apabila data inputan valid, maka thread baru berhasil dibuat, namun bila tidak valid, maka sistem akan meminta user memasukkan kembali data thread yang valid.

#### 4.1.6 Antarmuka Edit Thread



Gambar 4.6 Rancangan Antarmuka Edit Thread

Antarmuka ini digunakan untuk mengubah data thread. User memasukkan data thread yang baru,

kemudian user menekan tombol save. Apabila data inputan valid, maka thread berhasil diubah, namun bila tidak valid, maka sistem akan meminta user memasukkan kembali data thread yang valid.

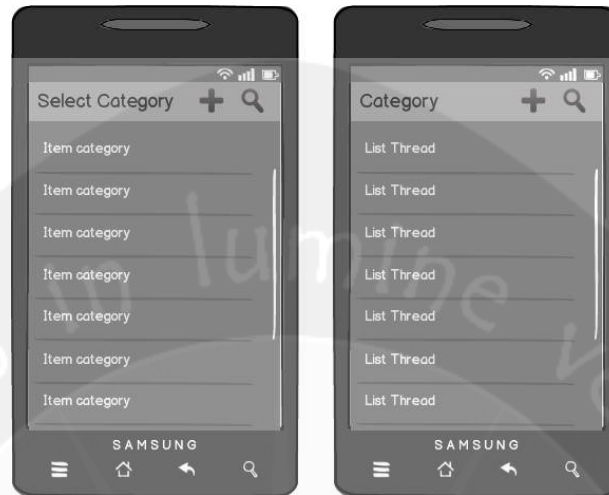
#### 4.1.7 Antarmuka Pencarian Thread



Gambar 4.7 Rancangan Antarmuka Pencarian Thread

Antarmuka ini digunakan untuk mencari thread berdasarkan keyword yang diberikan user. User menginputkan keyword pada kolom yang disediakan, kemudian user menekan tombol search jika data thread yang dicari ada maka akan tampil daftar dari thread yang dicari. Jika tidak ada maka tidak akan menampilkan apa-apa.

#### 4.1.8 Antarmuka Tampil Thread



Gambar 4.8 Rancangan Antarmuka Tampil Thread

Antarmuka ini digunakan untuk menampilkan thread yang mau dilihat oleh user. Pertama user akan dihadapkan dengan tampilan select category dimana user akan memilih satu diantara category. Setelah user memilih salah satu category maka sistem akan menampilkan data thread sesuai category yang user pilih.

#### 4.1.9 Antarmuka Tampil Detail My Thread

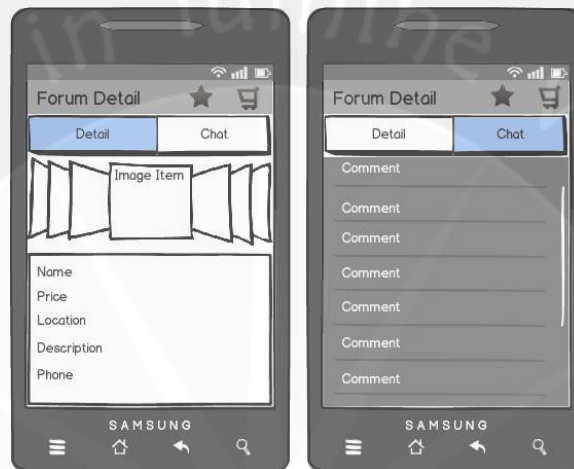


Gambar 4.9 Rancangan Antarmuka Tampil Detail My Thread

Antarmuka ini digunakan untuk menampilkan detail dari thread yang dibuat oleh user dan thread yang di order user. Pada tampilan ini user dapat memilih akan melihat thread atau thread order. Jika user memilih

thread maka daftar thread yang telah dibuat user akan muncul beserta status transaksi. Jika user memilih thread order maka user akan melihat daftar thread yang dipesannya dimana data yang ditampilkan berarti user belum melakukan konfirmasi pengiriman gambar resi transaksi.

#### 4.1.10 Antarmuka Tampil Detail Thread

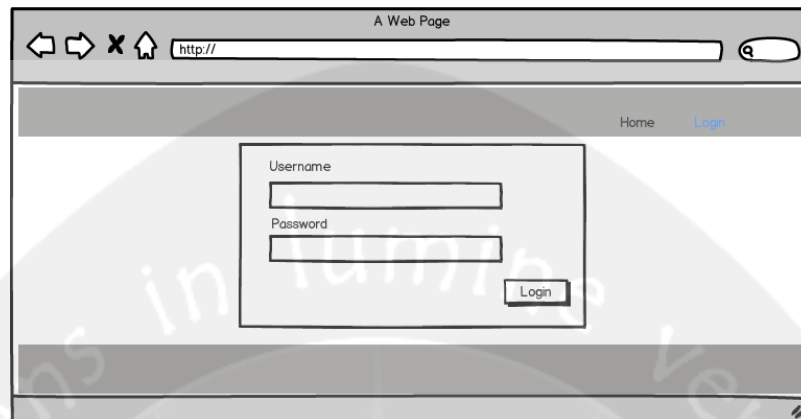


Gambar 4.10 Rancangan Antarmuka Tampil Detail Thread

Antarmuka ini digunakan untuk menampilkan detail dari thread serta komentar yang diberikan pada thread tersebut. Pada tampilan ini user dapat memilih akan melihat detail atau chat. Jika user memilih detail maka detail thread akan muncul dan jika user memilih chat maka daftar komentar akan muncul. Pada tampilan ini user juga dapat menyimpan thread ini ke menu bookmark dengan cara menekan gambar bintang, jika user belum menyimpan thread ini maka proses penyimpanan berhasil, tetapi jika user sudah menyimpan thread ini maka sistem akan memberikan pesan thread sudah di simpan. User juga dapat memesan barang yang dijual pada thread ini dengan cara menekan gambar keranjang, jika barang belum dipesan maka sistem akan memberikan pesan peringatan bahwa pemesanan berhasil jika telah dipesan maka sistem akan memberikan pesan peringatan barang telah dipesan.



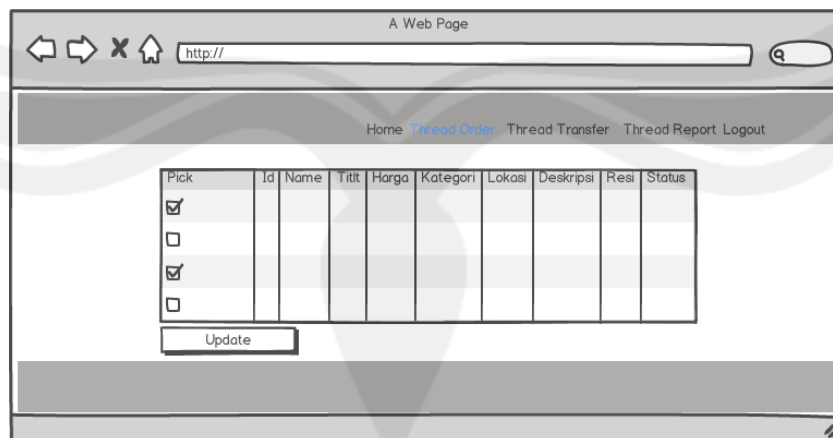
#### 4.1.11 Antarmuka Login Web



Gambar 4.11 Rancangan Antarmuka Login Web

Antarmuka ini digunakan untuk memasukkan username dan password agar pengguna dapat masuk ke dalam sistem web FORJUBEL. User memasukkan username dan password, lalu menekan tombol 'LOGIN'. Apabila username atau password valid, maka user akan masuk ke halaman utama, namun bila tidak valid, maka sistem akan meminta untuk memasukkan username dan password yang valid.

#### 4.1.12 Antarmuka Tampil Thread Order

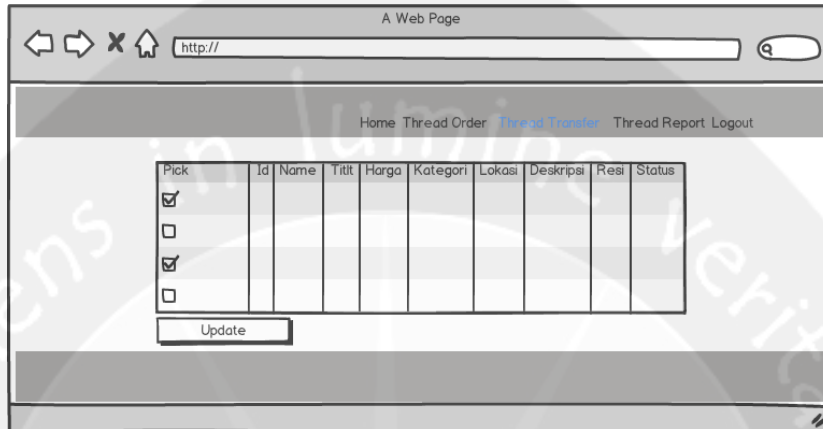


Gambar 4.12 Antarmuka Tampil Thread Order

Antarmuka ini digunakan untuk menampilkan data thread yang statusnya order dan mengupdate status threadnya menjadi transfer. Admin memilih beberapa

data thread kemudian menekan tombol update maka thread akan terupdate.

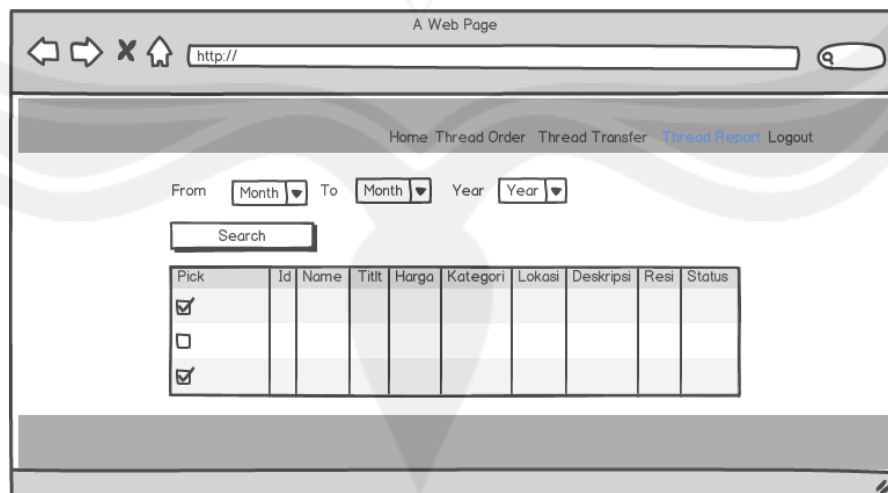
#### 4.1.13 Antarmuka Tampil Thread Transfer



Gambar 4.13 Antarmuka Tampil Thread Transfer

Antarmuka ini digunakan untuk menampilkan data thread yang statusnya transfer dan mengupdate status threadnya menjadi sold. Admin memilih beberapa data thread kemudian menekan tombol update maka thread akan terupdate.

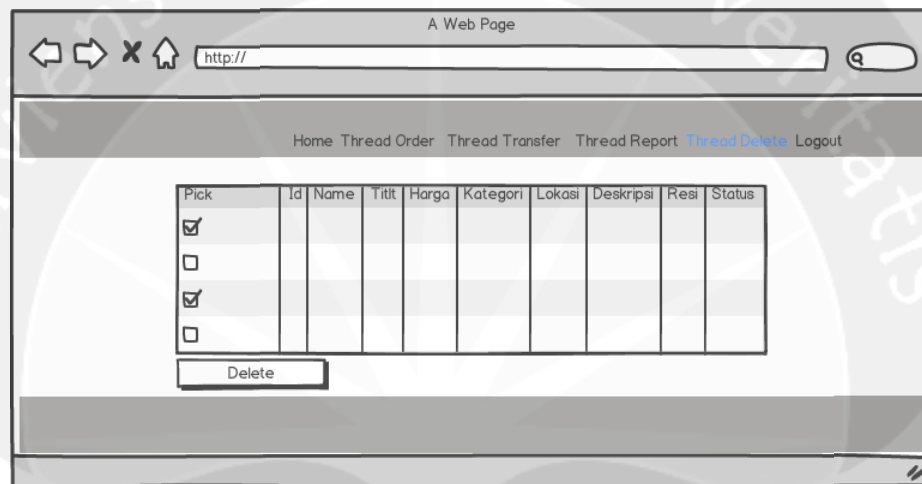
#### 4.1.14 Antarmuka Tampil Report Thread



Gambar 4. 14 Antarmuka Tampil Report Thread

Antarmuka ini digunakan oleh admin untuk mengecek laporan berdasarkan inputan bulan dan tahun. Admin memilih combobox dari bulan ke bulan, memilih tahun, kemudian menekan tombol search. Jika data yang dicari ada didalam database maka laporan thread akan muncul, jika tidak maka sistem menampilkan pesan thread yang dicari tidak ada.

#### 4.1.15 Antarmuka Delete Thread



Gambar 4.15 Antarmuka Delete Thread

Antarmuka ini digunakan oleh admin untuk menghapus data thread berdasarkan thread id. Admin memilih beberapa data yang ingin dihapus dengan cara mencentang checkbox yang tersedia, kemudian menekan tombol delete. Kemudian sistem akan memberikan peringatan apakah data ingin benar-benar dihapus. Jika ya maka data akan terhapus dari database, jika tidak maka data tidak berhasil dihapus.