



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - KI141502

## **RANCANG BANGUN APLIKASI WEB LELANG ONLINE (E-AUCTION) BERBASIS KERANGKA KERJA LARAVEL**

**RONAULI SILVA NATALENSIS SIDABUKKE**  
NRP 5113100142

Dosen Pembimbing I  
Rully Soelaiman, S.Kom, M.Kom

Dosen Pembimbing II  
Rizky Januar Akbar, S.Kom., M.Eng

**JURUSAN TEKNIK INFORMATIKA**  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya, 2017

*(Halaman ini sengaja dikosongkan)*

**TUGAS AKHIR - KI141502**

**RANCANG BANGUN APLIKASI WEB LELANG ONLINE  
(E-AUCTION) BERBASIS KERANGKA KERJA LARAVEL**

**RONAULI SILVA NATALENSIS SIDABUKKE**  
**NRP 5113100142**

**Dosen Pembimbing I**  
**Rully Soelaiman, S.Kom, M.Kom**

**Dosen Pembimbing II**  
**Rizky Januar Akbar, S.Kom., M.Eng**

**JURUSAN TEKNIK INFORMATIKA**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya, 2017**

*(Halaman ini sengaja dikosongkan)*

UNDERGRADUATE THESIS - KI141502

**E-AUCTION WEB APPLICATION DESIGN AND  
APPLICATION BASED ON LARAVEL FRAMEWORK**

**RONAULI SILVA NATALENSIS SIDABUKKE**  
**NRP 5113100142**

**Supervisor I**  
**Rully Soelaiman, S.Kom, M.Kom**

**Supervisor II**  
**Rizky Januar Akbar, S.Kom., M.Eng**

**Department of INFORMATICS**  
**Faculty of Information Technology**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya, 2017**

*(Halaman ini sengaja dikosongkan)*

## **LEMBAR PENGESAHAN**

### **RANCANG BANGUN APLIKASI WEB LELANG ONLINE (E-AUCTION) BERBASIS KERANGKA KERJA LARAVEL**

#### **TUGAS AKHIR**

Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Bidang Studi Algoritma Pemrograman  
Program Studi S1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**RONAULI SILVA NATALENSIS SIDABUKKE**  
**NRP: 5113100142**

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Rully Soelaiman, S.Kom, M.Kom

NIP: 197002131994021001

.....

(Pembimbing 1)

Rizky Januar Akbar, S.Kom., M.Eng

NIP: 198701032014041001

.....

(Pembimbing 2)

**SURABAYA**

**Juni 2017**

*(Halaman ini sengaja dikosongkan)*



# **RANCANG BANGUN APLIKASI WEB LELANG ONLINE (E-AUCTION) BERBASIS KERANGKA KERJA LARAVEL**

**Nama** : RONAULI SILVA NATALENSIS  
SIDABUKKE  
**NRP** : 5113100142  
**Jurusan** : Teknik Informatika FTIf  
**Pembimbing I** : Rully Soelaiman, S.Kom, M.Kom  
**Pembimbing II** : Rizky Januar Akbar, S.Kom., M.Eng

## **Abstrak**

*Industri e-commerce berkembang dengan pesat di Indonesia, seiring dengan meningkatnya jumlah pengguna internet dan menjamurnya bisnis online atau sering disebut online shop. Salah satu jenisnya adalah lelang online, yaitu metode jual beli yang mengintegrasikan mekanisme lelang dengan Internet.*

*Dalam interaksi antara pelaku lelang online (penjual dan pembeli) pasti terjadi kegagalan/ketidakpuasan dalam transaksi lelang online. Berangkat sebuah paper yang membahas mengenai analisa kesalahan dan strategi lewat survey terhadap pengguna aplikasi lelang online di Taiwan, penulis membangun aplikasi lelang online yang disertai dengan tambahan fitur maupun saran dari paper tersebut.*

*Tidak hanya berdasarkan paper rujukan, penulis juga menganalisa aplikasi e-commerce yang umum digunakan di Indonesia baik user experience maupun alur transaksi, dan menambahkan beberapa fitur agar lebih sesuai dengan transaksi jual-beli online yang umum di Indonesia. Dengan aplikasi ini, diharapkan kegagalan dalam transaksi online dapat diperbaiki dan membuka peluang lelang online untuk meramaikan industri e-commerce di Indonesia.*

**Kata-Kunci:** lelang online, strategi

## **E-AUCTION WEB APPLICATION DESIGN AND APPLICATION BASED ON LARAVEL FRAMEWORK**

**Name : RONAULI SILVA NATALENSIS  
SIDABUKKE**  
**NRP : 5113100142**  
**Major : Informatics FTIf**  
**Supervisor I : Rully Soelaiman, S.Kom, M.Kom**  
**Supervisor II : Rizky Januar Akbar, S.Kom., M.Eng**

### **Abstract**

*E-commerce industry is growing rapidly in Indonesia, along with the increasing number of internet users and number of online shops is also growing. One of e-commerce type is online auction, a buy and sell method that integrates auction mechanism and the Internet.*

*In the interaction between online auction actors (buyers and sellers), inevitable failure/dissatisfaction of online auction transactions sometimes found. Started by analysing paper about online auction application typologies and strategies through an application's users survey, author want to build online auction application along with additional ideas and suggestions from the paper.*

*Author also analyzed and considering user experience, design and transaction flow local e-commerce platforms that are commonly used in Indonesia, in purpose to make the application suits Indonesian's users better. Furthermore, author hopes that this applications can reduce/prevent the expected failures in online transactions and open up online auction opportunity to enliven the e-commerce industry in Indonesia.*

**Keyword:** *online auction, typologies and strategies*

## KATA PENGANTAR

Puji Syukur kepada Tuhan yang Maha Esa, atas berkatNya penulis dapat menyelesaikan buku berjudul **Rancang Bangun Aplikasi Web Lelang Online (E-Auction) Berbasis Kerangka Kerja Laravel**.

Selain itu, pada kesempatan ini penulis menghaturkan banyak terima kasih kepada pihak-pihak yang tanpa mereka, penulis tidak akan dapat menyelesaikan buku ini dengan baik :

1. **Daddy Jesus** - atas segala berkat, limpahan karunia, kesempatan dan rancangan jalanNya-lah penulis masih diberi nafas kehidupan, waktu, tenaga dan pikiran untuk menyelesaikan buku ini. *Thank you, Big Daddy*.
2. **Papa dan Mama** yang selalu menguatkan, menasehati, dan luar biasa sabar dalam mengingatkan penulis agar tidak lupa menjaga kesehatan dan tidak lupa ke gereja selama masa studi.
3. **Yth Bapak Rully Soelaiman** yang mengajarkan penulis *how to think scientifically* juga bimbingan, nasehat, saran dan memberikan penulis sisi pemikiran dan perspektif lain terhadap setiap masalah.
4. **Yth Bapak Rizky Januar Akbar** sebagai dosen pembimbing yang memberi bimbingan, saran teknis dan administratif, diskusi dan pemecahan masalah dalam pembuatan dan penulisan buku tugas akhir.
5. **Keluarga XL Future Leader Scholarship Camp Batch 5 & KSE ITS** yang telah menyadarkan, memberikan semangat dan inspirasi untuk terus melanjutkan tugas akhir di saat penulis kehilangan semangat.
6. **Keluarga Admin Lab. Pemrograman (2014 - 2017)**, yang telah memberikan penulis banyak pengalaman, pengetahuan dan cerita-cerita untuk dikenang.
7. **Keluarga Pengpro Furions dan HMTc Optimasi 2016** , yang mengajarkan penulis tentang cara berorganisasi, cara berbicara di depan publik, dan banyak lagi.
8. Bang Christo yang sudah jadi inspirasi dan semangat kuliah

penulis sejak tahun pertama masa studi penulis sampai saat ini.

9. **Keluarga Alumni Budi Mulia Siantar-Surabaya angkatan 2013** , teman setia disaat suka maupun duka.

10. Serta semua pihak yang tidak tertulis - yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Oleh karena itu, penulis berharap kritik dan saran dari pembaca sekalian untuk memperbaiki buku ini ke depannya.

Surabaya, Juli 2017

Ronauli Silva N. Sidabukke

## DAFTAR ISI

<b>ABSTRAK</b> .....	<b>7</b>
<b>ABSTRACT</b> .....	<b>8</b>
<b>KATA PENGANTAR</b> .....	<b>9</b>
<b>DAFTAR ISI</b> .....	<b>11</b>
<b>DAFTAR TABEL</b> .....	<b>15</b>
<b>DAFTAR GAMBAR</b> .....	<b>17</b>
<b>DAFTAR KODE SUMBER</b> .....	<b>19</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Metodologi .....	3
1.6 Sistematika Penulisan.....	3
<b>BAB II LANDASAN TEORI</b> .....	<b>5</b>
2.1 Amazon Web Service .....	5
2.2 <i>Concurrency</i> .....	5
2.3 <i>Data Growth</i> .....	6
2.4 <i>JSON Web Token</i> .....	6
2.5 Laravel.....	7
2.6 Laravel Dusk.....	7
2.7 Lelang .....	8
2.8 Lelang Daring / Lelang <i>Online</i> .....	8
2.9 MongoDB .....	9
2.10 Node.js .....	9
2.11 NoSQL .....	9
2.12 npm / <i>Node Package Manager</i> .....	10
2.13 PostgreSQL .....	10
2.14 Redis .....	11
2.15 <i>Repository Pattern</i> .....	11
2.16 SendGrid .....	12
2.17 <i>Service Worker</i> .....	12
2.18 SMTP / <i>Simple Mail Transfer Protocol</i> .....	12
2.19 Socket.io .....	13

2.20	<i>Test Script</i> .....	13
2.21	<i>Vendu Reglement</i> .....	13
2.22	<i>Vue.js / Vue</i> .....	14
2.23	<i>Whitelist</i> .....	14
<b>BAB III ANALISA DAN PERANCANGAN SISTEM.....</b>		<b>15</b>
3.1	Analisa Sistem.....	15
3.1.1	Deskripsi Umum Aplikasi .....	15
3.1.2	Analisa Paper Rujukan .....	16
3.1.3	<i>Bussiness Aspects of Software Engineering</i> .....	19
3.1.4	<i>Technical Analysis</i> .....	21
3.1.5	Analisa Pribadi Penulis.....	22
3.1.6	Spesifikasi Kebutuhan Non-Fungsional.....	28
3.1.7	Tugas dan Hak Akses Aktor .....	30
3.2	Perancangan Sistem .....	32
3.2.1	Spesifikasi Kasus Penggunaan .....	32
3.2.2	Identifikasi Komponen Fundamental .....	55
3.2.3	<i>Technology Options</i> .....	56
3.2.4	Arsitektur Perangkat Lunak .....	60
3.2.5	Struktur Aplikasi.....	63
3.2.6	Perancangan <i>Database</i> Transaksional .....	64
3.2.7	Perancangan <i>Database</i> Non-Transaksional .....	64
3.2.8	Kamus Data <i>Database</i> Transaksional.....	64
3.2.9	Kamus Data <i>Database</i> Non-Transaksional .....	74
3.2.10	Kamus Data .....	75
<b>BAB IV IMPLEMENTASI .....</b>		<b>77</b>
4.1	Implementasi Perangkat Keras/ <i>Deployment</i> .....	77
4.2	Implementasi Perangkat Lunak .....	78
4.2.1	Strategi <i>Deployment</i> .....	78
4.2.2	Diagram Kelas.....	83
4.3	Implementasi Antarmuka / <i>User Interface</i> .....	83
4.3.1	Antarmuka Registrasi.....	83
4.3.2	Antarmuka Halaman B .....	84

<b>BAB V PENGUJIAN DAN EVALUASI.....</b>	<b>85</b>
5.1 Pengujian .....	85
5.1.1 Pengujian Fungsionalitas Lelang .....	85
5.1.2 Pengujian Fungsionalitas Voucher.....	85
5.1.3 <i>Summary</i> Pengujian .....	86
5.2 Evaluasi .....	86
5.2.1 Evaluasi Aspek Kebutuhan .....	86
5.2.2 Evaluasi Aspek Teknis .....	86
5.2.3 Evaluasi Aspek <i>Bussiness Engineering</i> .....	86
5.2.4 Evaluasi Aspek Analisa dan Desain.....	86
5.2.5 Evaluasi Aspek Implementasi .....	86
5.2.6 Evaluasi Aspek Pengujian .....	86
5.2.7 <i>Summary</i> Evaluasi .....	86
5.2.8 <i>Further Enchancements</i> .....	86
5.2.9 <i>Further Readings</i> .....	86
<b>BAB VI PENUTUP.....</b>	<b>87</b>
6.1 Kesimpulan.....	87
6.2 Saran .....	87
Pustaka.....	88
<b>BIODATA PENULIS.....</b>	<b>91</b>

*(Halaman ini sengaja dikosongkan)*



## DAFTAR TABEL

Tabel 3.1 Kebutuhan Fungsional Aplikasi Lelang Online ....	27
Tabel 3.2 Kebutuhan Non-Fungsional Aplikasi Lelang Online .....	28
Tabel 3.3 Identifikasi aktor dalam sistem lelang online .....	30
Tabel 3.4 Detail Tugas dan Hak Akses .....	31
Tabel 3.5 Tabel Kasus Penggunaan.....	33
Tabel 3.6 Tabel Kasus Penggunaan.....	33
Tabel 3.7 Spesifikasi Kasus Penggunaan Registrasi .....	35
Tabel 3.8 Spesifikasi Kasus Penggunaan Login .....	36
Tabel 3.9 Spesifikasi Kasus Penggunaan: Konfirmasi Email	37
Tabel 3.10Spesifikasi Kasus Penggunaan : Melihat barang yang dilelang .....	39
Tabel 3.11Spesifikasi Kasus Penggunaan : Mencari Barang Lelang.....	40
Tabel 3.12Spesifikasi Kasus Penggunaan : Mencari Barang Lelang.....	41
Tabel 3.13Spesifikasi Kasus Penggunaan : Melihat Riwayat Penawaran Lelang Barang .....	42
Tabel 3.14Spesifikasi Kasus Penggunaan : Mendaftarkan Barang Lelang .....	44
Tabel 3.15Spesifikasi Kasus Penggunaan : Mendaftarkan Barang Lelang .....	45
Tabel 3.16Spesifikasi Kasus Penggunaan : Melihat Barang yang Pernah Didaftarkan.....	46
Tabel 3.17Spesifikasi Kasus Penggunaan : Melihat Riwayat Penawaran Harga.....	47
Tabel 3.18Spesifikasi Kasus Penggunaan : Melihat Review Pengguna.....	49
Tabel 3.19Spesifikasi Kasus Penggunaan : Memberikan Review Pengguna.....	50
Tabel 3.20Spesifikasi Kasus Penggunaan : Melaporkan Barang .....	51

Tabel 3.21Spesifikasi Kasus Penggunaan : Mengirimkan Pesan .....	52
Tabel 3.22Spesifikasi Kasus Penggunaan : Melihat & Membaca Pesan .....	53
Tabel 3.23Spesifikasi Kasus Penggunaan : Mendaftarkan Barang Lelang .....	54
Tabel 3.24Spesifikasi Tabel Administrator .....	64
Tabel 3.25Kamus Data Tabel bids .....	65
Tabel 3.26Kamus Data Tabel Categories .....	65
Tabel 3.27Kamus Data Tabel Coupon .....	66
Tabel 3.28Kamus Data Tabel Coupon Usages.....	67
Tabel 3.29Kamus Data Tabel Favorites .....	67
Tabel 3.30Kamus Data Tabel Issues .....	68
Tabel 3.31Kamus Data Tabel Issue Types .....	68
Tabel 3.32Kamus Data Tabel Items .....	68
Tabel 3.33Kamus Data Tabel Rating Auctioneers .....	70
Tabel 3.34Kamus Data Tabel Rating Bidders.....	71
Tabel 3.35Kamus Data Tabel Rating's Logs .....	71
Tabel 3.36Kamus Data Tabel User Verifications.....	72
Tabel 3.37Kamus Data Tabel Issue Types .....	73
Tabel 3.38Kamus Data Tabel Issue Types .....	73
Tabel 3.39Kamus data tabel Userchat .....	74
Tabel 3.40Kamus Data Tabel Chatroom .....	74
Tabel 3.41Kamus data tabel Itemimage .....	75

## DAFTAR GAMBAR

Gambar 3.1	Fatalitas kegagalan dalam aplikasi Lelang Online, Kepuasan terhadap Perbaikan Pelayanan dan <i>Repeat Purchase Intention</i> setelah Perbaikan Layanan .....	17
Gambar 3.2	Kategori Perbaikan terhadap Kegagalan Pelayanan Lelang Online .....	18
Gambar 3.3	Visualisasi aspek bisnis dalam <i>software engineering</i> .....	20
Gambar 3.4	<i>Essential attributes of good software</i> .....	21
Gambar 3.5	Diagram Kasus Penggunaan Aplikasi .....	33
Gambar 3.6	Diagram Kasus Penggunaan Manajemen Autentikasi Pengguna .....	34
Gambar 3.7	Diagram Kasus Penggunaan Manajemen Traksaksi Lelang .....	38
Gambar 3.8	Diagram Kasus Penggunaan Manajemen Barang Lelang .....	43
Gambar 3.9	Diagram Kasus Penggunaan Manajemen Interaksi Antarpengguna .....	48
Gambar 3.10	Diagram Kasus Penggunaan Manajemen Kupon .....	55
Gambar 3.11	Arsitektur dasar yang dibutuhkan untuk membangun aplikasi .....	56
Gambar 3.12	<i>Visualisasi arsitektur dan teknologi Final yang diterapkan dalam rancang bangun aplikasi</i> .....	62
Gambar 3.13	Komponen Penyusun Struktur Aplikasi .....	63
Gambar 4.1	Pseudocode Controller untuk Menampilkan Antarmuka A .....	83
Gambar 4.2	Pseudocode Controller untuk Menampilkan Antarmuka B .....	84

*(Halaman ini sengaja dikosongkan)*

## **DAFTAR KODE SUMBER**

*(Halaman ini sengaja dikosongkan)*

# BAB I

## PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

### 1.1 Latar Belakang

Transaksi jual beli saat ini sudah dapat dilakukan lewat berbagai cara, antara lain menggunakan *e-commerce*, atau lewat *social media*, atau bisa dengan melelang di aplikasi lelang *online*. Sedikit berbeda dengan teknik penjualan di lelang *online*, karena aplikasi ini dapat diakses oleh banyak orang, tentu saja pelelang (*auctioneer*) tidak terbatas pada ruang lelang saja, tapi bisa berasal dari manapun selama mereka mengakses aplikasi tersebut. Lelang *online* ini tentu saja mendatangkan banyak manfaat, selain biaya yang lebih efisien dan hemat, dan juga tidak menguras waktu karena siapapun, kapanpun, dimanapun dapat mengajukan penawaran ataupun melelang barangnya tanpa harus pergi ke instansi tertentu dan melakukan lelang dengan cara konvensional.

Aplikasi serupa telah banyak, namun banyak aspek yang kurang dalam aplikasi tersebut, seperti informasi dari lelang tidak *reliable* (misal: stok barang ternyata sudah habis), alur proses yang tidak jelas sehingga membingungkan pengguna aplikasi, informasi yang kurang jelas, dan produk yang didapatkan ternyata tidak sesuai dengan informasi pada saat produk dilelang (*bad information*) (Kuo, Yen, & Chen, 2016).

Dan dari masalah teknis aplikasi, beberapa sumber menyatakan bahwa ketidakjelasan alur proses yang kurang diperhatikan oleh para developer aplikasi lelang *online* menjadi beberapa alasan yang kuat mengapa lelang *online* masih kurang diminati (*Sistem Lelang Online Masih Bermasalah*, n.d.).

Diharapkan, dengan adanya aplikasi ini, beberapa kelemahan yang masih ada pada aplikasi lelang *online* saat ini dapat diperbaiki, dan juga dapat membantu proses *online* yang ada di Indonesia, dan juga mampu memperbaiki citra aplikasi lelang *online* sehingga mampu meningkatkan minat masyarakat terhadap lelang *online*.

## **1.2 Rumusan Masalah**

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana membangun aplikasi lelang online berbasis web?
2. Bagaimana rancangan arsitektur aplikasi dan fitur yang menganalisa kelemahan aplikasi serupa dan strategi penyelesaian sesuai dengan paper acuan (Kuo et al., 2016)?

## **1.3 Batasan Masalah**

Dari permasalahan yang telah diuraikan di atas, terdapat beberapa batasan masalah pada tugas akhir ini, yaitu:

1. Aplikasi berbasis web dengan bahasa pemrograman PHP.
2. Aplikasi berbasis kerangka kerja Laravel.
3. Basis data yang digunakan adalah PostgreSQL.
4. Aplikasi tidak mencakup proses pembayaran.

## **1.4 Tujuan**

Tujuan dari pengerjaan Tugas Akhir ini adalah:

1. Membangun aplikasi lelang online berbasis web yang lebih kredibel sesuai dengan paper yang dijadikan acuan pada tugas akhir ini.



## 1.5 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

### 1. Studi Literatur & Observasi

Pada tahap ini dilakukan pengumpulan dan penggalian informasi lewat literatur maupun artikel-artikel dari internet, yang diperlukan dalam proses perancangan dan implementasi sistem.

### 2. Analisa dan Perancangan Sistem

Pada tahap ini, dilakukan analisa dan pendefinisian kebutuhan sistem yang digunakan untuk masalah yang dihadapi. Tahapan-tahapannya adalah sebagai berikut:

- a analisa aktor yang terlibat dalam sistem;
- b perancangan model kasus penggunaan;
- c perancangan bisnis proses dalam aplikasi;
- d analisa masalah-masalah yang sering muncul saat penulis membuat aplikasi sebelumnya;
- e perancangan dan desain arsitektur aplikasi; dan
- f perancangan antarmuka aplikasi.

### 3. Implementasi

Tahap ini merupakan implementasi dari rancangan yang telah dibuat pada tahap sebelumnya.

### 4. Pengujian dan Evaluasi

Pada tahap ini, dilakukan pengujian terhadap aplikasi terhadap fungsionalitas dan non-fungsionalitas aplikasi.

## 1.6 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku

Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

### **Bab I Pendahuluan**

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

### **Bab II Dasar Teori**

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

### **Bab III Analisa dan Perancangan Sistem**

Bab ini membahas mengenai analisa dan perancangan aplikasi. Perancangan aplikasi meliputi perancangan data, arsitektur, proses dan struktur program.

### **Bab IV Implementasi**

Bab ini berisi deskripsi lengkap implementasi aplikasi.

### **Bab V Pengujian dan Evaluasi**

Bab ini membahas pengujian untuk menguji apakah aplikasi sudah tepat sasaran pada kebutuhan fungsional dan non-fungsional yang dirumuskan pada tahap Analisa dan Perancangan Sistem (Bab III).

### **Bab VI Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan, dan membahas saran beserta *further enhancements* untuk pengembangan sistem lebih lanjut

#### **Daftar Pustaka**

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir. **Lampiran** Merupakan bab tambahan yang berisi hal-hal terkait yang penting dalam aplikasi ini.

## **BAB II**

### **LANDASAN TEORI**

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir. Juga, teori-teori tersebut ditulis secara berurutan untuk mempermudah penulis mencari teori yang ingin dicari.

#### **2.1 Amazon Web Service**

Amazon Web Services adalah sekumpulan layanan-layanan berbasis Cloud Computing yang di sediakan oleh Amazon sejak tahun 2002. Meskipun salah satu perusahaan raksasa internet ini sering kita kenal untuk membeli buku dan lagu, namun sekarang Amazon telah menambah layanannya dalam hal infrastruktur cloud computing. Amazon Web Services ini menyediakan layanan-layanan nya yang saling terintegrasi dan mudah kustomisasi(Wikipedia, 2016a).

Dalam website resminya, disebutkan bahwa AWS dapat membantu aplikasi menjadi lebih cepat, lebih aman, dan menghemat *costs* dengan *scaling performance* menggunakan teknologi *cloud computing*(Web Services, n.d.).

#### **2.2 Concurrency**

Konkurensi adalah bisa dikatakan sebagai suatu fitur di mana database management system(DBMS) mengijinkan banyak transaksi pada saat bersamaan untuk mengakses data yang sama. Dalam melakukan konkurensi dibutuhkan suatu Concurrency Control Mechanism (CCM) agar transaksi yang dilakukan oleh banyak user pada suatu sistem di dalam waktu yang bersamaan tidak saling “mengganggu” dan tidak menghasilkan inconsistency data.

Tiga masalah umum yang muncul dalam konkurensi adalah sebagai berikut:

1. *Lost Update Problem*

Masalah operasi update yang sukses dari seorang pengguna kemudian ditimpali oleh operasi update dari pengguna lain.

2. *Uncomited dependency problem* (ketergantungan yg tidak sukses/modifikasi sementara)

Masalah terjadi saat suatu transaksi membaca data dari transaksi lain yg belum dicommit.

3. *Inconsistent analysis problem* (*Sistem Basis Data*, 2013)

## 2.3 *Data Growth*

Yang dimaksud dalam *data growth* pada *section* ini adalah seberapa cepat perkembangan jumlah data yang disimpan oleh server. Data tersebut bisa berupa *row* dalam *database* ataupun data *gambar*, *video dll* (insideBIGDATA, 2017).

Sejak tahun 2000s, perkembangan data meningkat pesat dan memunculkan bisnis penyedia *data storage* dan *networking equipment*.

Sebagai ilustrasi, pada buku referensi tercatat bahwa *data equilibrium flow* firma-firma *e-Commerce* pada dekade 1990-2000 meningkat hingga 1.5 *billion gigabytes* setiap tahunnya(D. VanHoose, 2011).

## 2.4 *JSON Web Token*

*JSON Web Token* atau lebih dikenal dengan JWT, merupakan sebuah token berbentuk JSON yang padat-informasi (ukurannya), informasi mandiri untuk ditransmisikan antar pihak yang terkait. Token tersebut ini dapat diverifikasi dan dipercaya karena sudah di-sign secara digital. Token JWT bisa di-sign dengan menggunakan secret (algoritma HMAC) atau pasangan public / private key (algoritma RSA) (JWT, 2016).

## 2.5 Laravel

Laravel adalah *framework* PHP yang dikembangkan pertama kali oleh Taylor Otwell. Walaupun termasuk baru, namun komunitas pengguna laravel sudah berkembang pesat dan mampu menjadi alternatif utama dari sejumlah *framework* besar seperti CodeIgniter dan Yii. Laravel oleh para *developer* disetarakan dengan CodeIgniter dan FuelPHP namun memiliki keunikan tersendiri dari sisi *coding*. Laravel memiliki beberapa keunggulan, diantaranya:

1. Sintaks yang sederhana dan *programmer-friendly*
2. Tersedia *generator* yang canggih dan memudahkan, Artisan CLI
3. Fitur *Schema Builder* untuk berbagai *database*
4. Fitur *Migration* dan *Seeding* untuk berbagai *database*
5. Fitur *Query Builder* yang powerful
6. Eloquent ORM (*Object Relational Mapping*)
7. Fitur pembuatan *package* dan *bundle*
8. *Dependency Injection* (*Apa itu Laravel?*, 2016)

## 2.6 Laravel Dusk

Laravel Dusk adalah sebuah fitur baru yang ditujukan untuk *functional testing*, yang baru diluncurkan dan dinamakan secara *default* pada Laravel versi 5.4. Dalam *site* dokumentasinya, disebutkan bahwa Laravel Dusk menyediakan *browser automation & testing API* yang ekspresif dan mudah digunakan. Secara otomatis, Dusk tidak memerlukan instalasi JDK atau Selenium pada *host*, namun menggunakan *ChromeDriver standalone*, namun juga dapat menggunakan driver Selenium yang kompatibel (Laravel, n.d.).

## 2.7 Lelang

Lelang adalah proses membeli dan menjual barang atau jasa dengan cara menawarkan kepada penawar, menawarkan tawaran harga lebih tinggi, dan kemudian menjual barang kepada penawar harga tertinggi. Dalam teori ekonomi, lelang mengacu pada beberapa mekanisme atau peraturan perdagangan dari pasar modal (BalaiLelang, n.d.).

Lelang menurut sejarahnya berasal dari bahasa Latin *auctio* yang berarti peningkatan harga secara bertahap. Para ahli menemukan bahwa dalam literatur Yunani, lelang telah dikenal 450 tahun sebelum Masehi. Jenis lelang yang populer saat itu antara lain adalah karya seni, tembakau, kuda, budak dan sebagainya (Pratama, 2012).

## 2.8 Lelang Daring / Lelang *Online*

Lelang adalah proses membeli dan menjual barang atau jasa dengan cara menawarkan kepada penawar, menawarkan tawaran harga lebih tinggi, dan kemudian menjual barang kepada penawar harga tertinggi. Dalam teori ekonomi, lelang mengacu pada beberapa mekanisme atau peraturan perdagangan dari pasar modal.

Sementara lelang daring atau lelang melalui internet muncul seiring dengan perkembangan internet. Barang atau jasa yang diperjualbelikan dipasang di situs dan peserta lelang dapat mengikuti acara lelang secara daring. Perusahaan lelang yang berhasil menggunakan sarana internet salah satunya adalah *Ebay*. Di Indonesia, lelang melalui internet (online) sudah dipelopori oleh pemerintah dengan situs lelang online yang dapat diakses melalui website resmi Kemenkeu (Wikipedia, 2016b). Berikut adalah beberapa istilah yang ada dalam lelang online:

1. BID atau *Bidding*, artinya: Menawarkan

2. BIN (*Buy In Now*) artinya: Beli sesuai harga yang telah ditawarkan penjual
3. INC (*Increment*) artinya: Minimum kenaikan *bid* setelah *bid* sebelum nya (*Arti pengertian BID, INC, BIN dalam Bahasa Lelang Website, n.d.*)

## 2.9 MongoDB

MongoDB (dari "humongous") adalah sistem basis data berorientasi dokumen lintas platform. Diklasifikasikan sebagai basis data "NoSQL", MongoDB menghindari struktur basis data relasional tabel berbasis tradisional yang mendukung JSON seperti dokumen dengan skema dinamis (MongoDB menyebutnya sebagai format BSON), membuat integrasi data dalam beberapa jenis aplikasi lebih mudah dan lebih cepat. Dirilis di bawah kombinasi dari GNU Affero General Public License dan Lisensi Apache, MongoDB adalah perangkat lunak bebas dan sumber terbuka (MongoDB, 2017).

## 2.10 Node.js

Node.js adalah platform perangkat lunak pada sisi-server dan aplikasi jaringan. Ditulis dengan bahasa javascript dan bisa dijalankan pada Windows, Mac OS X dan Linux tanpa perubahan kode program. Node.js memiliki pustaka server HTTP sendiri sehingga memungkinkan untuk menjalankan webserver tanpa menggunakan program webserver seperti Apache atau Lighttpd (Node.js (Wikipedia), 2014).

## 2.11 NoSQL

NoSQL adalah istilah yang dikenal dalam teknologi komputasi untuk merujuk kepada kelas yang luas dari sistem

manajemen basis data yang diidentifikasi dengan tidak mematuhi aturan pada model sistem manajemen basis data relasional yang banyak digunakan.

NoSQL tidak dibangun terutama dengan table dan umumnya tidak menggunakan SQL untuk memanipulasi data, sehingga sering ditafsirkan sebagai “tidak hanya SQL” (Wikipedia, n.d.).

## 2.12 npm / *Node Package Manager*

NPM memiliki dua fungsi utama, yaitu sebagai repositori online yang berisi banyak package atau module untuk aplikasi NodeJS dan yang kedua adalah sebuah utilitas baris perintah (command line) yang digunakan untuk menginstal paket-paket yang dibutuhkan dan juga untuk mengelola versi dan ketergantungan package dari NodeJS. Dengan NPM Anda akan mudah mencari, menginstal, uninstall aplikasi atau module/package Node.js (Azurri, 2016).

## 2.13 PostgreSQL

PostgreSQL adalah sebuah produk *database* relasional yang termasuk dalam kategori *free open source software (FOSS)*. PostgreSQL terkenal karena fitur-fitur yang advanced dan pendekatan rancangan modelnya menggunakan paradigma *object-oriented*, sehingga sering dikategorikan sebagai *Object Relational Database Management System (ORDBMS)*. Beberapa fitur PostgreSQL adalah sebagai berikut:

1. *Inheritance*, dimana satu table dapat diturunkan model dan beberapa karakteristik dari table lainnya.
2. *Multi-Version Concurrency Control* dimana user diberi data snapshot ketika suatu perubahan dilakukan sampai commit.
3. *Rules*, dimana suatu *query* DML yang dikirimkan ke server akan mengalami penulisan ulang (*rewrite*). Ini terjadi



sebelum diproses oleh *query planner*.

4. dan berbagai fitur lainnya (*PostgreSQL*, n.d.)

## 2.14 Redis

Redis adalah *open source*, struktur data yang ditempatkan di memori, digunakan sebagai *database*, *cache* dan *message broker*. Redis mendukung struktur data seperti *string*, *sets*, *hash*, *lists* dan *sorted sets*. Sama seperti *cache*, setiap key diisi oleh value. Tapi kelebihanannya, Redis bisa digunakan untuk melakukan operasi dari value tersebut. Cara terbaik untuk memahami redis adalah membuat model aplikasi tanpa memikirkan bagaimana caranya untuk menyimpan data di dalam *database* (Yudana, 2015).

## 2.15 Repository Pattern

*Repository Pattern* adalah sebuah pola dalam struktur *software engineering* yang memisahkan *data management layer* ke dalam sebuah layer tersendiri - yang dihandle oleh sebuah bagian struktur yang disebut *repository*.

Jika menggunakan *pattern* ini, semua kode spesifik yang terkait dengan *persistence logic* dan implementasi akses data berhenti sampai di *repository* (*controller* hanya *redirect request* dan validasi *request*)(*Repository Pattern - Coding4ever's Blog*, 2016).

Dalam sebuah referensi, disebutkan bahwa: "*The repository pattern covers large centralized transaction-oriented databases, the blackboard systems used for some AI applications, and systems with predetermined execution patterns in which different phases add information to a single complex data structures (e.g., compilers). These variants differ chiefly in their control structure.*" (Shaw, 1996).

## 2.16 SendGrid

SendGrid adalah sebuah *customer communication platform* untuk email *marketing* dan transaksional yang berbasis di Denver, Colorado.

SendGrid menyediakan layanan pengiriman email yang berbasis *cloud* kepada pihak bisnis. Layanan yang ditawarkan sangat beragam, mulai dari *shipping notifications*, *friend requests* dan lain lagi.

Selain itu, juga dapat *handling* ISP monitoring, *domain keys*, *feedback loops*, dan juga memberikan report *opened mails*, *unsubscribes*, *bounces* dan *spam reports*. Pada tahun 2012, SendGrid menggaet Twilio dan menambahkan layanan integrasi SMS, Suara dan *push notification*. (Wikipedia, 2017a)

## 2.17 Service Worker

*Service worker* adalah skrip yang dijalankan browser Anda di latar belakang, terpisah dari laman web, yang membuka pintu ke berbagai fitur yang tidak memerlukan laman web atau interaksi pengguna. Saat ini, service worker sudah menyertakan berbagai fitur seperti pemberitahuan push dan sinkronisasi latar belakang. Di masa mendatang, service worker akan mendukung hal-hal lainnya seperti sinkronisasi berkala atau geofencing. Fitur inti yang didiskusikan dalam tutorial adalah kemampuan mencegah dan menangani permintaan jaringan, termasuk mengelola cache respons lewat program (Google Developers, 2017).

## 2.18 SMTP / Simple Mail Transfer Protocol

SMTP adalah suatu protokol yang digunakan untuk mengirimkan pesan e-mail antar server, yang bisa dianalogikan sebagai kantor pos. Ketika kita mengirim sebuah e-mail, komputer kita akan mengarahkan e-mail tersebut ke sebuah

SMTP server, untuk diteruskan ke mail-server tujuan (*SMTP*, n.d.).

## 2.19 Socket.io

Socket.io adalah *library* Javascript untuk aplikasi web yang bersifat *realtime*. Socket.io menjembatani antara komunikasi dua arah antara *web clients* dan *server*. Socket.io terbagi menjadi dua bagian, yaitu *client-side library* yang berjalan di browser client, dan *server-side library* yang menggunakan Node.js. Kedua komponen tersebut mempunyai API yang sama. Seperti Node.js, Socket.io juga bersifat *event-driven*. Socket.IO menggunakan protokol *websocket* dengan *polling* sebagai opsi *fallback*. Meskipun Socket.IO merupakan ‘pembungkus’ untuk soket web, namun ia memiliki banyak fitur, antara lain broadcast ke banyak soket, dan I/O yang asinkronus (*Socket.io (Wikipedia)*, 2016).

## 2.20 Test Script

Test Script dalam dunia *software testing* adalah set instruksi atau sekumpulan baris kode yang akan melakukan *testing* terhadap fungsionalitas sistem dengan target tertentu (, 2016).

Ada beberapa jenis *script test*:

1. *Manual testing*, atau lebih sering disebut *test cases*
2. *Automated Testing*

## 2.21 Vendu Reglement

Lelang dilegalisasi & resmi masuk Indonesia dalam perundang-undangan sejak 1908, yaitu dengan berlakunya *Vendu Reglement*, Stbl. 1908 No. 189 dan *Vendu Instructie*, Stbl 1908 No. 190. *Vendu Reglement* ini berisikan peraturan-peraturan dasar lelang yang berlaku hingga saat ini, dan menjadi dasar hukum penyelenggaraan lelang di Indonesia (BalaiLelang, n.d.).

## 2.22 Vue.js / Vue

Vue adalah sebuah *framework* Javascript yang *progressive* dan bersifat *open-source* untuk membangun *user interface*. Integrasi kedalam project yang menggunakan *library* Javascript lain menjadi lebih mudah dengan Vue karena Vue memang didesain untuk *incrementally adoptable*. Vue juga dapat berfungsi sebagai *web application framewowrk* untuk membangun sebuah *single-page applications*.

Dalam 2016 *Javascript Survey*, Vue mendapatkan 89% untuk kategori *developer satisfaction rating*. Vue mengakumulasikan 98 *stars Github* setiap hari, dan menduduki peringkat ke-10 *Project Github* dengan bintang terbanyak *all the time*(Wikipedia, 2017b).

## 2.23 Whitelist

Whitelist sendiri adalah memindahkan daftar alamat atau domain dari pengirim email dengan harapan agar muncul pada kotak pesan email utama Anda. Sederhananya, hal ini dilakukan untuk menghindari agar pesan email yang dikirimkan oleh pengirim pesan sebenarnya tidak terbaca sebagai spam (Wikipedia, 2017a).

## **BAB III**

### **ANALISA DAN PERANCANGAN SISTEM**

#### **3.1 Analisa Sistem**

Pada subbab ini, penulis akan memaparkan analisa terhadap kebutuhan baik fungsional dan non-fungsional, teknis dan serta pengaruhnya dalam perancangan aplikasi pada subbab 3.2. Subbab Analisa Sistem akan dipaparkan secara sistematis sebagai berikut:

- a. analisa kebutuhan dasar;
- b. analisa aspek bisnis;
- c. analisa kebutuhan fungsional;
- d. analisa kebutuhan non-fungsional; dan
- e. rancangan arsitektur, strktur dan teknologi yang digunakan.

##### **3.1.1 Deskripsi Umum Aplikasi**

*Bid! Bid! Bid!* Istilah tersebut pasti tidak asing bagi para pecinta lelang online. Lelang sendiri adalah proses membeli dan menjual barang atau jasa dengan cara menawarkan kepada penawar, menawarkan tawaran harga lebih tinggi, dan kemudian menjual barang kepada penawar harga tertinggi (Wikipedia.com). Di Indonesia, sistem lelang sudah digunakan sejak jaman Hindia Belanda dimana saat itu sistem lelang pertama kali diperkenalkan di Indonesia dan biasa digunakan lelang terhadap aset-aset pejabat atau pemerintah yang dimutasi pada saat itu (Artikel sejarah lelang, 2011).

Dalam proses bisnisnya sendiri, lelang cukup sederhana, yaitu pengguna dapat melelang dan dapat menjual barang untuk dilelang. Yang berarti, pengguna dapat memanajemen riwayat lelang dan memanajemen barang yang ia daftarkan untuk dilelang. Hal ini akan didefinisikan lebih lanjut dalam subbab 3.1.5.7, yaitu subbab Spesifikasi Kebutuhan Fungsional.

### 3.1.2 Analisa Paper Rujukan

Bercermin terhadap aplikasi *e-commerce* yang telah ada, masalah yang paling sering dialami adalah ketidakpuasan pengguna. Salah satu indikator bahwa suatu perusahaan dikatakan memiliki ketidakpuasan pelanggan adalah karena kegagalan dalam pelayanannya. Seorang pelanggan sangat mungkin memutuskan untuk komplain setelah mengalami ketidakpuasan terhadap layanan suatu perusahaan, dan jika tidak ditangani dengan baik, hal ini bisa berakibat fatal terhadap reputasi dan kepercayaan pengguna terhadap aplikasi tersebut.

Oleh karena itu, sebuah paper mengangkat topik ini khusus dalam bidang aplikasi lelang online, menganalisa kegagalan dan ketidakpuasan pengguna, beserta solusi-solusi yang ditawarkan oleh pengguna aplikasi untuk memperbaiki kegagalan pelayanan tersebut.

Type of service failure	Severity of service failures	Satisfaction with recovery	Repeat purchase intention with recovery
<i>Group 1 service delivery system failures</i>			
Packaging problem	7.1 <sup>a</sup> (2.5) <sup>b</sup>	6.3 (3.1)	5.9 (3.1)
Slow/unavailable service	6.9 (2.5)	4.9 (3.1)	4.4 (3.2)
Product defect	7.5 (2.6)	5.5 (3.4)	4.9 (3.4)
Out of stock	7.2 (2.3)	5.3 (2.9)	5.1 (3.3)
Bad information	7.7 (2.4)	4.2 (3.2)	3.6 (3.1)
Alterations and repairs	8.1 (2.7)	2.2 (2.2)	2.0 (1.9)
Hold disaster	7.8 (1.8)	2.6 (1.7)	3.0 (2.1)
Pricing failure	7.8 (3.0)	5.2 (3.6)	5.7 (3.6)
Policy failure	7.1 (2.1)	3.7 (2.8)	3.2 (2.8)
Subtotal, Group 1	7.2 (2.5)	5.3 (3.2)	4.9 (3.3)
<i>Group 2 Buyer needs and requests</i>			
Gap between expectation and perception	7.9 (2.2)	2.7 (2.6)	2.4 (2.5)
Size variation	6.7 (2.0)	5.9 (3.6)	5.9 (3.6)
Special order or request	8.6 (1.6)	3.5 (3.5)	3.3 (3.1)
Admitted buyer error	4.0 (1.7)	5.7 (2.1)	4.0 (3.5)
Subtotal, Group 2	7.7 (2.2)	3.3 (3.0)	3.0 (3.0)
<i>Group 3 Unprompted and unsolicited seller actions</i>			
Seller attention failures	7.6 (2.4)	3.5 (2.8)	3.3 (2.8)
Seller-created embarrassments	7.7 (2.8)	3.5 (2.7)	3.8 (3.1)
Seller fraud problem	9.4 (1.2)	1.1 (0.4)	1.0 (0.0)
Mischarged	6.8 (2.9)	5.4 (4.0)	5.2 (3.9)
Leak of personal data	9.5 (0.7)	1.0 (0.0)	1.0 (0.0)
Subtotal, Group 3	8.0 (2.4)	3.1 (2.8)	3.0 (2.9)
Total	7.4 (2.5)	4.8 (3.3)	4.4 (3.3)

<sup>a</sup> Mean.

<sup>b</sup> Standard deviation.

**Gambar 3.1** Fatalitas kegagalan dalam aplikasi Lelang Online, Kepuasan terhadap Perbaikan Pelayanan dan *Repeat Purchase Intention* setelah Perbaikan Layanan

Dalam gambar diatas, dijabarkan beberapa jenis kegagalan yang pernah dialami oleh pengguna aplikasi serta fatalitas/pengaruh buruk kegagalan tersebut terhadap kepercayaan pengguna.

Recovery strategy	No. (%)	No. of satisfactory recovery	No. of dissatisfactory recovery	Satisfaction with recovery	Retention with recovery
Correction	280 <sup>a</sup> (32.3) <sup>b</sup>	228 <sup>a</sup> (81.4) <sup>b</sup>	52 <sup>a</sup> (18.6) <sup>b</sup>	7.2 <sup>c</sup> (2.4) <sup>d</sup>	6.6 <sup>c</sup> (2.7) <sup>d</sup>
Correction plus	82 (9.5)	77 (93.9)	5 (6.1)	7.7 (2.1)	7.6 (2.5)
Discount	26 (3.0)	23 (88.5)	3 (11.5)	7.0 (2.2)	6.0 (2.9)
Replacement	14 (1.6)	2 (14.3)	12 (85.7)	3.4 (2.7)	3.5 (2.7)
Store credit	16 (1.8)	9 (56.3)	7 (43.7)	6.1 (2.9)	6.4 (2.7)
Apology	54 (6.2)	20 (37.0)	34 (63.0)	3.7 (2.6)	3.4 (2.6)
Refund	67 (7.7)	42 (62.7)	25 (37.3)	6.4 (2.9)	5.2 (3.1)
Unsatisfactory correction	182 (21.0)	0 (0)	182 (100.0)	1.9 (1.3)	1.8 (1.6)
Failure	39 (4.5)	0 (0)	39 (100)	1.3 (0.9)	1.3 (0.9)
Nothing	107 (12.4)	0 (0)	107 (100)	1.7 (1.2)	1.7 (1.6)
Total	867 (100)	407 (46.9)	460 (53.1)	4.8 (3.3)	4.4 (3.3)

<sup>a</sup> Number.

<sup>b</sup> %.

<sup>c</sup> Mean.

<sup>d</sup> Standard deviation.

**Gambar 3.2** Kategori Perbaikan terhadap Kegagalan Pelayanan Lelang Online

Maka berdasarkan hasil analisa tersebut, fitur-fitur yang ditambahkan selain daripada fitur dasar aplikasi lelang online sebagai *added vaue* adalah sebagai berikut:

1. Fitur chatting, untuk mengurangi kemungkinan *Bad Information* dimana ekspektasi dan persepsi terhadap barang yang dilelang antara pembeli dan penjual tidak sama dan *Special Needs*,
2. Fitur pemberian kupon voucher (*Discount and Correction*)



*Plus*) yang bisa berupa *free shipping* atau *discount*.

### 3.1.3 *Bussiness Aspects of Software Engineering*

Lelang merupakan salah satu metode pertukaran barang dan jasa dengan metode penetapan harga yang berbeda dengan perdagangan. Oleh karena itu, lelang juga termasuk dalam kategori bisnis. Yang menarik adalah, ketika bisnis digabungkan dengan teknologi atau yang sering disebut *e-commerce*, hal yang sekedar pertukaran barang bertransformasi menjadi sebuah sistem interaktif yang kompleks dimana tujuan utamanya adalah menarik pengunjung/pengguna untuk menyelesaikan sebuah transaksi. Hal ini tentu sangat krusial, penting, dan tertantang untuk menyelesaikannya.

Dalam mencapai kesuksesan dan tingkat kompetitif yang tinggi, haruslah menyediakan layanan dengan kesan *user experience (UX)* yang positif bagi para penggunanya. Morville (Laia Bonastre, 2014, p. 27) , dalam studi yang dilakukannya, menyebutkan bahwa UX tercakup dalam 7 aspek esensial, yaitu

- a. *useful*
- b. *usable*
- c. *findable*
- d. *desirable*
- e. *accessible*
- f. *credible*.

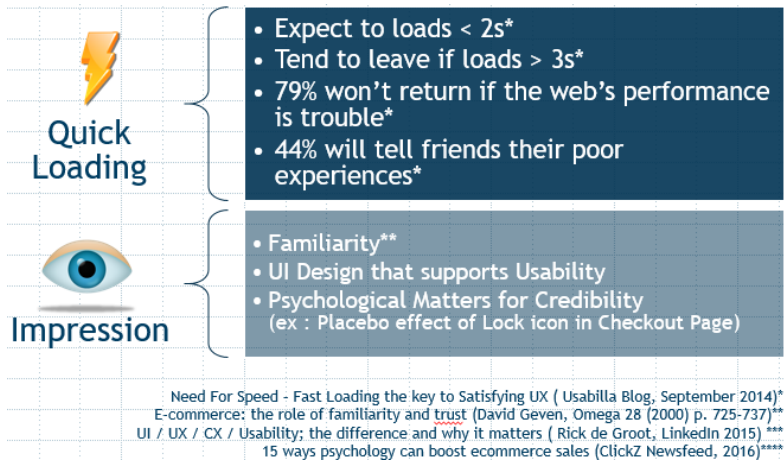
Hasil-hasil temuan penting yang menarik dalam pengaruh *user experience*, adalah sebagai berikut dikutip dari sebuah sumber adalah sebagai berikut:

- a. *User tend to leave if a page loads more than 3 seconds;*
- b. *79% of users won't return if the web's performance and experience is poor;*
- c. *44% of users will tell the poor experiences to their friends.*

Selain dari faktor *user experience* dan *performance*, beberapa hal yang menjadi poin penting dan menarik dalam beberapa studi

yang terkait adalah sebagai berikut:

- a *Familiarity* - yang dapat didefinisikan sebagai tingkat familier atau kesamaan dengan sistem sejenis ternyata dapat membangun *trust* sehingga mensugesti pengguna untuk menyelesaikan transaksi yang dilakukan;
- b *Usability* yang memudahkan pengguna dalam menyelesaikan transaksi; dan
- c Aspek-aspek psikologi seperti pemilihan warna, penggunaan *icon* yang sesuai, seperti *icon* gembok pada halaman pembayaran ternyata dapat mengesankan *security* pada pengguna.



**Gambar 3.3** Visualisasi aspek bisnis dalam *software engineering*

Dari hasil temuan ini, dapat disimpulkan bahwa *user experiences*, *performances*, *usability* dan psikologi memiliki pengaruh besar dalam kesuksesan lelang online dalam menarik hati para penggunanya. Hal ini akan mempengaruhi definisi kebutuhan fungsionalitas yang akan dibahas dalam subbab 3.1.5.7.

### 3.1.4 Technical Analysis

Untuk membuat sebuah aplikasi yang sukses, tentunya banyak sekali aspek yang harus diperhatikan. Selain kualitas aplikasi yang akan dibuat, juga ketahanannya terhadap perubahan karena *e-commerce* adalah sesuatu yang sangat cepat berubah karena kompetitor yang sangat kompetitif dan dorongan teknologi yang membuat efektifitas dan efisiensi menjadi lebih baik.

Dari aspek *software engineering* sendiri, *software engineering* dimaksudkan untuk menunjang/support pengembangan *software* daripada *individual programming*. Hal ini mencakup: **a) evolution** **b) design** **c) supporting program specification** (Sommerville, 2009, p. 5)

Product characteristics	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security	Software dependability includes a range of characteristics including reliability, security, and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilization, etc.
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable, and compatible with other systems that they use.

**Gambar 3.4** *Essential attributes of good software*

Berdasarkan kriteria tersebut, maka setiap poin perlu diperhatikan agar dapat mengembangkan sebuah aplikasi yang tidak hanya sukses, tapi juga bertahan dalam kompetisi. Dalam istilah bisnis, hal ini disebut dengan *risk management & planning*.

### 3.1.5 Analisa Pribadi Penulis

Dalam pemaparan selanjutnya, penulis akan

#### 3.1.5.1 Analisa *User Experience* dari E-Commerce di Indonesia

Selama masa pengerjaan aplikasi, penulis sering menganalisa dan memperhatikan kebiasaan-kebiasaan yang umum di website *e-commerce* di Indonesia. Salah satu yang paling sering dianalisa oleh penulis adalah adalah situs Tokopedia. Dalam pengembangannya, *user interface* aplikasi akan dipengaruhi analisa ini, yang dijabarkan seperti berikut:

1. Halaman yang muncul bukanlah *eagerloading*, tapi *lazy loading*  
Ini adalah solusi cerdas untuk mengakali *delay loading item* yang sudah pasti jumlahnya sangat banyak (maka butuh *query* yang tentunya memakan waktu cukup lama), namun juga memainkan faktor psikologi / *user behaviour* pengguna dengan membiarkan pengguna melihat tahap demi tahap halaman 'diisi'.
2. *User Interface* yang sederhana dan pemilihan warna yang *soft*

#### 3.1.5.2 Analisa Keamanan pada koneksi Soket

Untuk mengakomodasi fitur yang bersifat *realtime*, dibutuhkan koneksi ke soket secara terus menerus. Hal ini tentu dapat menjadi sasaran empuk *security* karena jika tidak diamankan, maka dapat menjadi peluang besar bagi para pihak yang tidak berkepentingan untuk merusak proses bisnis aplikasi.

Namun, jika dalam setiap koneksi soket harus mengirimkan *credentials*, hal ini tentu menjadi tidak praktis dan malah lebih berbahaya karena membiarkan data-data sensitif seperti *password* dan *username* berlalu-lalang di jaringan internet.

Selain itu, *disadvantagenya* adalah ketidakpraktisan untuk selalu meng*query* database setiap kali ada koneksi, tentu saja ini memperlambat kerja *database* dan menambah waktu *delay*. Maka dari itu, penulis mengidentifikasi poin-poin penting berikut :

- Hindari *query* database untuk *autentikasi* yang sifatnya masif
- Menggunakan mekanisme autentikasi yang menggunakan *credentials* karena rentan dengan masalah keamanan
- Mencari metode yang lebih efektif, cepat untuk autentikasi selain

### 3.1.5.3 Analisa *Best Practice* dalam Struktur Perangkat Lunak

Pada dasarnya, Laravel adalah kerangka kerja MVC. Namun, ada banyak fitur yang ada dalam aplikasi Lelang Online ini yang tidak terakomodasi dalam MVC, misal sebagai berikut :

1. Sistem Verifikasi lewat Email - yang berarti aplikasi harus berinteraksi dengan SMTP server
2. Sistem *Generate Token JWT.io*, dimana dalam proses *Generate Token* sama sekali tidak ada database dilibatkan.

Jika fitur-fitur tersebut 'dipaksa' dimuat ke dalam MVC, maka tentu saja strukturnya menjadi ganjil, dan muncul *code smell* berikut :

1. *Large Class*, dimana terdapat satu buah file yang sangat panjang (biasanya merupakan entitas utama, dalam hal ini contohnya barang/item)
2. *Inappropriate Intimacy*, dimana terdapat satu kelas yang menyimpan *logic* yang tidak seharusnya ia simpan
3. *Duplicated Code*

Dari hasil analisa ini, penulis mengidentifikasi strategi-strategi yang akan diterapkan dalam rancangan struktur aplikasi pada subbab 3.2.5, yaitu sebagai berikut:

1. Penggunaan Repository Pattern

Memisahkan antara Data Processing Layer dan View Layer - agar lebih rapi, terstruktur, hal ini juga dapat menghindari *Duplicated Code*.

2. Penambahan Komponen : Service dan Provider

Untuk memisahkan *logic* aplikasi yang terkait dengan *akseseksternal services*. Tujuannya, agar jika kedepannya terdapat perbaikan fitur/penambahan fitur, lebih mudah melakukan *traceback* terhadap file/kelas yang bertanggungjawab terhadap fitur tersebut.

### 3.1.5.4 Analisa Aplikasi Serupa

Selama pengerjaan aplikasi, penulis menganalisa aplikasi serupa. Penulis menemukan aplikasi yang kurang lebih alur bisnis/alur penggunaan aplikasinya serupa yaitu : Carousell. Penulis melihat beberapa kesamaan antara sifat transaksi aplikasi tugas akhir saya dengan aplikasi tersebut, yaitu:

1. Sama-sama tidak mengakomodasi pembayaran
2. Sama-sama tidak adanya kepastian harga (bedanya, pada Carousell yang terjadi adalah *bargaining*

Sehingga dalam alur proses nya, banyak diadaptasi dari Carousell, agar pengguna dapat lebih familiar dan *predictability*nya lebih tinggi jika diadaptasi dari *E-commerce* lainnya yang lebih umum digunakan oleh pengguna.

### 3.1.5.5 Analisa Aplikasi Serupa

Selama pengerjaan aplikasi, penulis menganalisa aplikasi serupa. Penulis menemukan aplikasi yang kurang lebih alur bisnis / alur penggunaan aplikasinya serupa yaitu : Carousell.

Penulis melihat ada beberapa kesamaan antara sifat transaksi aplikasi tugas akhir saya dengan aplikasi tersebut, yaitu :

1. Sama-sama tidak mengakomodasi pembayaran

2. Sama-sama tidak adanya kepastian harga (bedanya, pada Carousell yang terjadi adalah *bargaining* Sehingga dalam alur proses nya, banyak diadaptasi dari Carousell, agar pengguna dapat lebih familiar dan *predictability*nya lebih tinggi jika diadaptasi dari *E-commerce* lainnya yang lebih umum digunakan oleh pengguna.

### 3.1.5.6 Analisa Penyimpanan Data

Untuk penyimpanan data, terdapat 2 jenis data yang sifatnya cukup berbeda, yaitu sebagai berikut:

#### 1. Data transaksional disimpan di DBMS SQL - *Relational*

Data yang sifatnya *transaksional*, seperti data *bidding*, data pengguna, dan lain sebagainya. Untuk data ini, lebih baik jika menggunakan database Postgre, untuk menjaga integritas data dan *integrity checking* juga menjadi lebih baik.

#### 2. Data non-transaksional disimpan di DBMS NoSQL

Data *chatting*, data *joined rooms* kurang tepat jika disimpan dalam database transaksional karena sifat pertambahan datanya yang sangat cepat, masif dan urgensi integritas data tidak terlalu diprioritaskan (dibanding dengan data transaksional pada poin sebelumnya). Oleh karena itu, baiknya data ini disimpan pada database NoSQL dengan alasan-alasan sebagai berikut.

- Banyaknya transaksi *read write*;
- Ketidaksamaan frekuensi *read and write* data semua pengguna;
- Sifat permintaan transaksi yang cepat; dan
- Kemungkinan perubahan struktur atribut pada pesan (misal: *attachments, forwarding, replying*, dll) akan sangat menyulitkan pengembangan selanjutnya jika menggunakan database transaksional yang terpaku

pada skema database yang ditetapkan di awal pengembangan aplikasi.

### 3. **Data citra/gambar menggunakan layanan Pihak Ketiga**

Sekarang telah banyak penyedia jasa *cloud computing* sebagai infrastruktur, seperti Amazon Web Service, Google Cloud Storage Google Alasan-alasan menggunakan AWS sebagai data storage untuk gambar adalah sebagai berikut :

- (a) Skalabilitas aplikasi lebih terjaga.

Dengan memisahkan penyimpanan antara gambar dan server sehingga lebih mudah *maintain* perkembangan aplikasi, dan lebih fokus terhadap pengembangan aplikasi.

- (b) Menyediakan *built-in* keamanan, fleksibel dan efisiensi (Wikipedia, 2016a)

### 4. **Optimasi assets**

Dalam banyak kesempatan, penulis seringkali mendapati bahwa *delay* untuk *loading assets* lebih lama daripada *loading data* dari database. Berikut penulis akan memaparkan hasil analisa berupa penyebab dan *tackling* permasalahan tersebut.

- (a) *Useless assets* yang disertakan dalam halaman : Memisahkan *essentials assets* dan menyertakan *script* yang hanya digunakan oleh halaman tersebut.

- (b) Logika penyusunan script yang tidak efektif dan optimal (misal: ada satu script yang menyertakan file yang tidak diperlukan) : dilakukan *pre-processing* berupa *minifying*, *optimization*, *compiling*, *compression* terhadap *assets*

- (c) Latensi ke server yang cukup tinggi (misal: kecepatan sambungan internet yang rendah) : *Caching*, *upgrading server* agar dapat "lebih terjangkau" secara jaringan, penerapan PWA



(*Progressive Web Apps*) untuk sisi *user experience*.

### 3.1.5.7 Spesifikasi Kebutuhan Fungsional

Berdasarkan deskripsi umum sistem pada subbab 3.1.1, maka dapat disimpulkan bahwa kebutuhan fungsional dari aplikasi lelang online, yang dipaparkan dalam tabel 3.1.

**Tabel 3.1** Kebutuhan Fungsional Aplikasi Lelang Online

No.	Parameter	Ketersediaan
1	Manajemen Akun	Pengguna: Memperbarui dan mengubah informasi pengguna dalam akun
2	Memanajemen penawaran terhadap barang lelang	Pengguna: Menawar barang, mendapat informasi barang, mencari barang yang diinginkan, dan fitur-fitur yang mempermudah pengguna dalam penawaran barang
3	Memanajemen barang yang dilelang	Pengguna: Mendaftarkan barang untuk dilelang, melihat progress kemajuan lelang, membatalkan lelang dari pengguna tertentu
4	Memanajemen Laporan Pengguna	Pengelola: Melihat daftar laporan dari pengguna, memblokir pengguna yang melanggar ketentuan
Dilanjutkan ke halaman selanjutnya		

Tabel 3.1 – lanjutan dari halaman sebelumnya

No.	Parameter	Ketersediaan
5	Memanaajemen Kupon	Pengelola: Membuat kupon, melihat daftar kupon, melihat riwayat penggunaan kupon

### 3.1.6 Spesifikasi Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipeuhi oleh aplikasi ini berhubungan dengan faktor-faktor sebagai berikut:

Tabel 3.2 Kebutuhan Non-Fungsional Aplikasi Lelang Online

No.	Parameter	Ketersediaan
1	Ketersediaan	Aplikasi harus dapat berjalan pada browser, tanpa dibatasi waktu dan tempat selama browser tersebut tersambung ke jaringan internet.
2	Bahasa	Bahasa yang digunakan pada antarmuka merupakan bahasa Indonesia
3	Otorisasi	Setiap pengguna hanya berhak mengakses dan mengubah data yang merupakan milik pengguna tersebut.
3	Performa	Rata-rata waktu akses halaman tidak boleh lebih dari 3 detik.
.. dilanjutkan ke halaman selanjutnya		

Tabel 3.2 – lanjutan dari halaman sebelumnya

No.	Parameter	Ketersediaan
4	Portabilitas	Aplikasi dapat diakses pada semua platform selama platform tersebut terinstall web browser
4	<i>User Experience</i>	Aplikasi memberikan kesan positif terhadap pengalaman penggunaan aplikasi
4	<i>Security</i>	Koneksi terhadap web harus terlindung <i>https</i> .
4	<i>Flexibility</i>	Aplikasi haruslah bersifat fleksibel terhadap perubahan (jika terdapat perubahan fitur di masa depan, tidak harus <i>merefactor</i> atau mengubah struktur program).

### 3.1.7 Tugas dan Hak Akses Aktor

Aplikasi lelang online dapat digunakan sebagai wadah bagi para pecinta lelang online untuk melakukan kegiatan lelang atau bagi para penjual atau pembeli yang ingin menjual atau membeli barang yang sesuai baik dari kualitas maupun harga. Identifikasi aktor dalam sistem lelang online dijelaskan dalam tabel 3.3.

**Tabel 3.3** Identifikasi aktor dalam sistem lelang online

No	Aktor	Keterangan
1	Pengguna	Pengguna yang dimaksud adalah pengguna yang menggunakan fungsionalitas lelang dalam sistem. Pengguna dapat menjadi seorang <i>bidder</i> ataupun seorang <i>auctioneer</i> , dimana pengguna dapat menjual barang untuk dilelang, dan dapat pula melelang barang
2	<i>Administrator</i>	Bertugas melihat dan mengawasi jalannya lelang, melihat laporan keluhan dari pengguna, serta melakukan <i>block</i> pengguna jika terdeteksi adanya tindakan yang tidak sesuai dengan kaidah lelang.

**Tabel 3.4** Detail Tugas dan Hak Akses

<b>Aktor</b>	<b>Tugas</b>	<b>Hak Akses</b>	<b>Kemampuan yang harus dimiliki</b>
Pengguna	Melakukan aktivitas lelang dan aktivitas interaksi antarpengguna	Manajemen barang & aktivitas lelang dan interaksi antarpengguna.	
Dilanjutkan ke halaman selanjutnya			

Tabel 3.4 – lanjutan dari halaman sebelumnya

Aktor	Tugas	Hak Akses	Kemampuan yang harus dimiliki
<i>Administrator</i>	Bertugas melihat dan mengawasi jalannya lelang, melihat laporan keluhan dari pengguna, serta melakukan <i>block</i> pengguna jika terdeteksi adanya tindakan yang tidak sesuai dengan kaidah lelang.	Mengakses halaman-halaman monitoring dari sistem terpisah.	

## 3.2 Perancangan Sistem

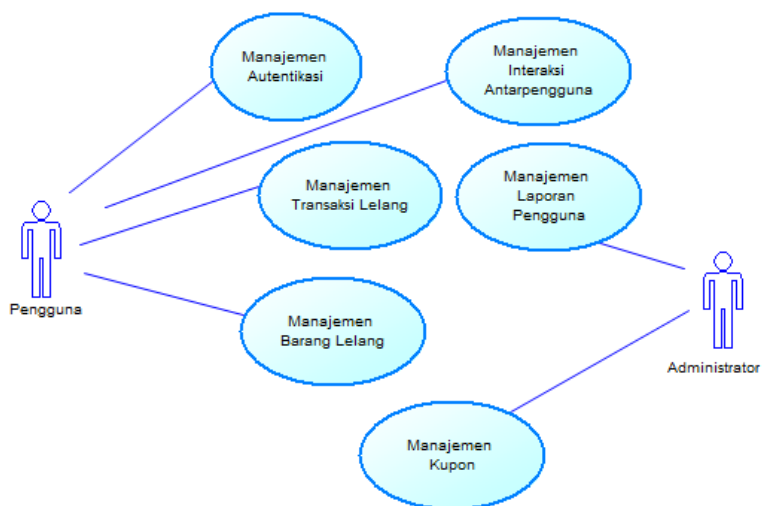
### 3.2.1 Spesifikasi Kasus Penggunaan

Kasus penggunaan disini dimaksudkan untuk menurunkan kebutuhan fungsional yang telah dispesifikasikan sebelumnya pada tabel 3.1 sebelumnya.

Daftar kasus Penggunaan dapat dilihat pada 3.6.

**Tabel 3.5** Tabel Kasus Penggunaan

<b>ID Kasus Penggunaan</b>	<b>Kasus Penggunaan</b>
KP-01	Manajemen Authentikasi Pengguna
KP-02	Memanajemen Transaksi Lelang
KP-03	Manajemen Barang Lelang
KP-04	Manajemen Interaksi Antar Pengguna
KP-05	Monitoring Proses Lelang
KP-06	Manajemen Voucher



**Gambar 3.5** Diagram Kasus Penggunaan Aplikasi

Selanjutnya, akan dijabarkan masing-masing Spesifikasi Kasus Penggunaan untuk semua Kasus Penggunaan yang telah

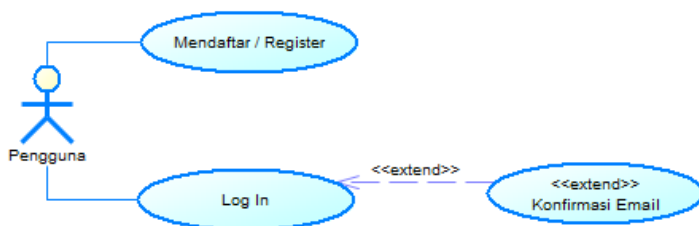
dijabarkan diatas.

### 3.2.1.1 KP01. Manajemen Autentikasi Pengguna

Pada kasus penggunaan ini, pengguna dapat memanajemen autentikasi dan pendaftaran ke dalam sistem.

**Tabel 3.6** Tabel Kasus Penggunaan

<b>ID Kasus Penggunaan</b>	<b>Kasus Penggunaan</b>
KP-01	Manajemen Authentikasi Pengguna
KP-02	Memanajemen Transaksi Lelang
KP-03	Manajemen Barang Lelang
KP-04	Manajemen Interaksi Antar Pengguna
KP-05	Monitoring Proses Lelang
KP-06	Manajemen Voucher



**Gambar 3.6** Diagram Kasus Penggunaan Manajemen Autentikasi Pengguna



<b>Kode</b>	UC-01.01
<b>Nama</b>	<b>Registrasi</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna mendaftar ke dalam akun agar masuk ke dalam sistem
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Pengguna belum memiliki akun di aplikasi
<b><i>Postcondition</i></b>	Pengguna sudah memiliki akun terdaftar di aplikasi
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka Halaman Registrasi</li> <li>2. Sistem menampilkan halaman yang berisi Form Registrasi</li> <li>3. Pengguna mengisi form tersebut</li> <li>4. Setelah selesai mengisi, pengguna mengklik tombol "Registrasi"</li> <li>5. Sistem memvalidasi data yang dimasukkan pengguna</li> <li>6. Jika data valid, sistem <i>redirect</i> ke halaman <i>landing page</i> dalam keadaan sudah terautentikasi &amp; akun berhasil didaftarkan, dan sistem mengirimkan email konfirmasi email ke alamat email yang didaftarkan.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Data yang dimasukkan pengguna tidak valid</b>
	<ol style="list-style-type: none"> <li>5a. Sistem tidak dapat memvalidasi data yang dimasukkan pengguna.</li> <li>5b. Sistem <i>redirect</i> ke halaman form registrasi (langkah 2) dengan <i>error message</i>.</li> </ol>

Tabel 3.7 Spesifikasi Kasus Penggunaan Registrasi

<b>Kode</b>	UC-01.02
<b>Nama</b>	<b><i>Login</i></b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna melakukan <i>login</i> agar dapat masuk ke dalam aplikasi dalam keadaan terautentikasi.
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Pengguna masuk ke dalam aplikasi dalam keadaan belum terautentikasi
<b><i>Postcondition</i></b>	Pengguna masuk ke dalam aplikasi dalam keadaan sudah terautentikasi
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka Halaman Login</li> <li>2. Sistem menampilkan halaman Login</li> <li>3. Pengguna mengisi halaman sesuai <i>credential</i> yang dimiliki</li> <li>4. Setelah selesai mengisi, pengguna mengklik tombol "Login"</li> <li>5. Sistem memverifikasi <i>credential</i> yang diberikan</li> <li>6. Jika data benar, sistem <i>redirect</i> ke halaman <i>landing page</i> dalam keadaan sudah terautentikasi.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Data yang dimasukkan pengguna tidak valid</b>
	<ol style="list-style-type: none"> <li>5a. Sistem tidak dapat memverifikasi <i>credential</i> pengguna.</li> <li>5b. Sistem <i>redirect</i> ke halaman Login (langkah 2) dengan <i>error message</i>.</li> </ol>

**Tabel 3.8** Spesifikasi Kasus Penggunaan Login

<b>Kode</b>	UC-01.03
<b>Nama</b>	<b>Konfirmasi Email</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna melakukan konfirmasi email agar status akun pengguna menjadi teraktivasi
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Status akun pengguna masih belum terverifikasi
<b><i>Postcondition</i></b>	Status akun pengguna masih sudah terverifikasi
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman <i>inbox</i> email pengguna di <i>sistem email service</i> yang mereka gunakan.</li> <li>2. pengguna mencari dan membuka email konfirmasi yang dikirimkan oleh Lelangapa</li> <li>3. Sistem <i>email service</i> pengguna menampilkan isi email konfirmasi, beserta sebuah tombol "Konfirmasi email"</li> <li>4. Pengguna mengklik tombol "Konfirmasi Email"</li> <li>5. Halaman <i>browser</i> akan <i>diredirect</i> ke URL konfirmasi email</li> <li>6. Sistem menampilkan halaman <i>landing page</i> dimana status akun pengguna sudah terverifikasi.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.9** Spesifikasi Kasus Penggunaan: Konfirmasi Email

### 3.2.1.2 KP02. Manajemen Transaksi Lelang



**Gambar 3.7** Diagram Kasus Penggunaan Manajemen Traksaksi Lelang

Pada kasus penggunaan ini, pengguna akan dapat memanajemen transaksi dan penawaran-penawaran yang ia berikan terhadap barang yang terdaftar dalam alikasi.

<b>Kode</b>	UC-02.01
<b>Nama</b>	<b>Melihat daftar barang yang dilelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna melihat daftar barang yang sedang dilelang
<b>Tipe</b>	Fungsional
<b><i>Pre Condition</i></b>	Sistem belum menampilkan daftar barang yang sedang dilelang
<b><i>Post Condition</i></b>	Sistem menampilkan daftar barang yang sedang dilelang
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna mengklik <i>icon</i> aplikasi di kiri atas halaman</li> <li>2. Sistem menampilkan halaman depan yang berisi daftar barang yang sedang dilelang</li> </ol> <p><i>Ket : Pada halaman depan, ditampilkan barang sesuai dengan kategori berdasarkan waktu dan popularitas, seperti Hot Item (barang yang paling ramai transaksi bidnya), Newest Item, dll.</i></p>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.10** Spesifikasi Kasus Penggunaan : Melihat barang yang dilelang

<b>Kode</b>	UC-02.02
<b>Nama</b>	<b>Mencari Barang Lelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin mencari barang lelang dengan kriteria nama tertentu
<b>Tipe</b>	Fungsional
<b>Pre Condition</b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> .
<b>Post Condition</b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> masukan.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan kriteria <i>string</i> pencarian di <i>field</i> masukan di <i>Header Bar</i></li> <li>2. Setelah selesai, pengguna mengklik tombol "Cari"</li> <li>3. Sistem mencari barang terdaftar yang sesuai dengan kriteria masukan pengguna</li> <li>4. Jika ketemu, sistem menampilkan halaman "Hasil Pencarian" beserta barang yang sesuai dengan kriteria pengguna.</li> <li>5. Pengguna lalu mengklik barang yang sesuai dengan keinginan</li> <li>6. Sistem menampilkan detail barang yang sesuai dengan keinginan pengguna</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Tidak ada barang terdaftar dalam sistem yang sesuai dengan kriteria pengguna.</b>
	<ol style="list-style-type: none"> <li>4 a. Sistem tidak dapat menemukan barang yang sesuai</li> <li>4 b. Sistem menampilkan "Hasil Pencarian" namun dengan keterangan "Hasil pencarian kosong"</li> </ol>

**Tabel 3.11** Spesifikasi Kasus Penggunaan : Mencari Barang Lelang

<b>Kode</b>	UC-02.03
<b>Nama</b>	<b>Mencari Barang Lelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin mencari barang lelang dengan kriteria nama tertentu
<b>Tipe</b>	Fungsional
<b>Pre Condition</b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> .
<b>Post Condition</b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> masukan.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan kriteria <i>string</i> pencarian di <i>field</i> masukan di <i>Header Bar</i></li> <li>2. Setelah selesai, pengguna mengklik tombol "Cari"</li> <li>3. Sistem mencari barang terdaftar yang sesuai dengan kriteria masukan pengguna</li> <li>4. Jika ketemu, sistem menampilkan halaman "Hasil Pencarian" beserta barang yang sesuai dengan kriteria pengguna.</li> <li>5. Pengguna lalu mengklik barang yang sesuai dengan keinginan</li> <li>6. Sistem menampilkan detail barang yang sesuai dengan keinginan pengguna</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Tidak ada barang terdaftar dalam sistem yang sesuai dengan kriteria pengguna.</b>
	<ol style="list-style-type: none"> <li>4 a. Sistem tidak dapat menemukan barang yang sesuai</li> <li>4 b. Sistem menampilkan "Hasil Pencarian" namun dengan keterangan "Hasil pencarian kosong"</li> </ol>

**Tabel 3.12** Spesifikasi Kasus Penggunaan : Mencari Barang Lelang

<b>Kode</b>	UC-02.04
<b>Nama</b>	<b>Melihat 3 Riwayat Penawaran Lelang Barang Teratas</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin melihat riwayat penawaran lelang terhadap barang yang ia daftarkan
<b>Tipe</b>	Fungsional
<b><i>Pre Condition</i></b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> .
<b><i>Post Condition</i></b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> masukan.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman "Kelola Barang"</li> <li>2. Pengguna mengklik barang yang ingin dilihat informasi riwayat penawaran lelangnya</li> <li>3. Sistem menampilkan halaman informasi barang tersebut</li> <li>4. Pengguna mengklik tombol "Lihat Penawaran Teratas"</li> <li>5. Sistem menampilkan <i>modal</i> berisi 3 riwayat penawaran lelang teratas barang tersebut.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.13** Spesifikasi Kasus Penggunaan : Melihat Riwayat Penawaran Lelang Barang



### 3.2.1.3 KP03. Manajemen Barang Lelang



**Gambar 3.8** Diagram Kasus Penggunaan Manajemen Barang Lelang

Pada kasus penggunaan ini, pengguna akan dapat memanajemen barang yang ia daftarkan untuk dilelang, dan melihat proses monitoringnya, seperti yang dipaparkan pada penjelasan berikut.

<b>Kode</b>	UC-03.01
<b>Nama</b>	<b>Mendaftarkan Barang Lelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna mendaftarkan barang untuk dilelang di dalam sistem
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Barang yang akan dilelang belum terdaftar dalam sistem
<b>Postcondition</b>	Barang yang akan dilelang sudah terdaftar dalam sistem
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna dalam keadaan terautentikasi, mengklik "Item Anda" -&gt; "Add Items" pada <i>navbar</i> bagian atas halaman.</li> <li>2. Sistem menampilkan halaman yang berisi form pendaftaran barang</li> <li>3. Pengguna mengisi form tersebut sesuai data barang</li> <li>4. Setelah selesai mengisi, pengguna mengklik tombol "Daftar Barang"</li> <li>5. Sistem memvalidasi data yang dimasukkan pengguna</li> <li>6. Jika data valid, sistem <i>redirect</i> ke halaman "Kelola Barang" dalam keadaan barang baru sudah ditambahkan.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Data barang yang dimasukkan pengguna tidak valid</b>
	<ol style="list-style-type: none"> <li>5a. Sistem tidak dapat memvalidasi data yang dimasukkan pengguna.</li> <li>5b. Sistem <i>redirect</i> ke halaman form "Tambah Barang" (langkah 2) dengan <i>error message</i>.</li> </ol>

**Tabel 3.14** Spesifikasi Kasus Penggunaan : Mendaftarkan Barang Lelang

<b>Kode</b>	UC-03.02
<b>Nama</b>	<b>Memperbarui informasi barang yang dilelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna memperbarui informasi barang yang sebelumnya sudah terdaftar di dalam sistem
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Informasi barang belum diperbarui.
<b><i>Postcondition</i></b>	Informasi barang sudah diperbarui.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna dalam keadaan terautentikasi, mengklik "Item Anda" -&gt; "Manage Items" pada <i>navbar</i> bagian atas halaman.</li> <li>2. Sistem menampilkan halaman yang berisi daftar barang yang didaftarkan pengguna.</li> <li>3. Pengguna mengklik barang yang ingin diperbarui informasinya</li> <li>4. Sistem menampilkan halaman <i>form</i> "Perbarui barang".</li> <li>5. Pengguna mengisi informasi pembaruan barang di dalam form tersebut.</li> <li>6. Setelah selesai, pengguna mengklik tombol "Simpan Pembaruan".</li> <li>7. Sistem memvalidasi data (termasuk file gambar) yang dimasukkan pengguna</li> <li>8. Jika data valid, sistem <i>redirect</i> ke halaman "Kelola Barang" dalam keadaan barang baru sudah ditambahkan.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Data barang yang dimasukkan pengguna tidak valid</b>
	<ol style="list-style-type: none"> <li>7a. Sistem tidak dapat memvalidasi data yang dimasukkan pengguna.</li> <li>7b. Sistem <i>redirect</i> ke halaman "Perbarui Barang" (langkah 4) dengan <i>error message</i>.</li> </ol>
	<b>Gambar yang dimasukkan pengguna tidak</b>

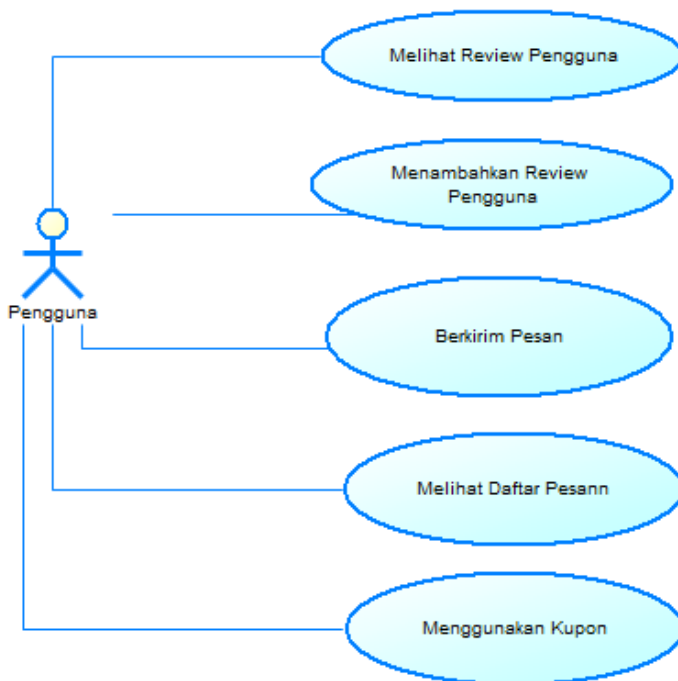
<b>Kode</b>	UC-03.03
<b>Nama</b>	<b>Melihat Daftar Barang yang Pernah Dilelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna hendak melihat daftar semua barang yang pernah didaftarkan untuk dilelang di dalam sistem.
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Informasi daftar barang belum ditampilkan.
<b><i>Postcondition</i></b>	Informasi daftar barang sudah ditampilkan.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna dalam keadaan terautentikasi, mengklik "Item Anda" -&gt; "Manage Items" pada <i>navbar</i> bagian atas halaman.</li> <li>2. Sistem menampilkan halaman yang berisi daftar barang yang didaftarkan pengguna.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.16** Spesifikasi Kasus Penggunaan : Melihat Barang yang Pernah Didaftarkan

<b>Kode</b>	UC-03.04
<b>Nama</b>	<b>Melihat Detail Riwayat Penawaran Harga</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna hendak melihat daftar semua barang yang pernah didaftarkan dalam sistem.
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Informasi daftar barang belum ditampilkan.
<b>Postcondition</b>	Informasi daftar barang sudah ditampilkan.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna dalam keadaan terautentikasi, mengklik "Item Anda" -&gt; "Manage Items" pada <i>navbar</i> bagian atas halaman.</li> <li>2. Sistem menampilkan halaman yang berisi daftar barang yang didaftarkan pengguna.</li> <li>3. Pengguna mengklik barang yang ingin dilihat daftar penawaran harganya</li> <li>4. Sistem menampilkan halaman detail informasi barang</li> <li>5. Pengguna mengklik tombol "Lihat Riwayat Penawaran"</li> <li>6. Sistem menampilkan halaman berisi daftar riwayat penawaran.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.17** Spesifikasi Kasus Penggunaan : Melihat Riwayat Penawaran Harga

### 3.2.1.4 KP04. Manajemen Interaksi Antarpengguna



**Gambar 3.9** Diagram Kasus Penggunaan Manajemen Interaksi Antarpengguna

Pada kasus penggunaan ini, pengguna difasilitasi untuk berinteraksi, memberikan *review* / testimoni terhadap pengguna lainnya sesuai dengan keinginan.

<b>Kode</b>	UC-04.01
<b>Nama</b>	<b>Melihat Review Pengguna</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin melihat review pada pengguna tertentu
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Review pengguna belum ditampilkan
<b><i>Postcondition</i></b>	Review pengguna berhasil ditampilkan
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna mengklik <i>link</i> profil pengguna</li> <li>2. Sistem menampilkan halaman profil pengguna</li> <li>3. Pengguna dapat melihat <i>review</i> pengguna di bagian kiri bawah beserta rata-rata <i>rating</i> yang diberikan.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.18** Spesifikasi Kasus Penggunaan : Melihat Review Pengguna

<b>Kode</b>	UC-04.02
<b>Nama</b>	<b>Menambahkan Review Pengguna</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin menambahkan review dari <i>transaksi</i> yang pernah dilakukan.
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Review dari pengguna belum tercatat/tersimpan dalam sistem
<b>Postcondition</b>	Review dari pengguna berhasil tercatat dalam sistem

#### **Alur Kejadian Normal**

1. Pengguna mengklik halaman 'Riwayat Transaksi'
2. Sistem menampilkan halaman Riwayat Transaksi yang pernah dilakukan pengguna
3. Pengguna mengklik *tab* jenis transaksi yang pernah dilakukan (Beli atau Lelang)
4. Sistem menampilkan riwayat transaksi sesuai dengan jenis transaksi yang dipilih pengguna
5. Pengguna mengklik transaksi yang ingin diberikan *review*
6. Sistem mengecek apakah *review* sudah pernah diberikan sebelumnya
7. Sistem menampilkan *modal* berisi *field input* jumlah *rating*
8. Pengguna mengisi *field* tersebut sesuai jumlah *rating* yang ingin diberikan
9. Setelah selesai, pengguna klik 'Next'
10. Sistem menampilkan *modal* kedua, berisikan *field input* untuk deskripsi *review*
11. Pengguna mengisi *field input* sesuai dengan deskripsi yang ingin diberikan
12. Setelah selesai, pengguna mengklik tombol 'Simpan Review'
13. Sistem memvalidasi masukan dari pengguna
14. Jika tervalidasi, sistem menampilkan modal berisi informasi sukses menyimpan review



<b>Kode</b>	UC-04.03
<b>Nama</b>	<b>Melaporkan Barang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin melaporkan barang yang dianggap melanggar aturan/tidak pantas diperjualbelikan
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Laporan dari pengguna belum tersimpan dalam sistem
<b><i>Postcondition</i></b>	Laporan dari pengguna berhasil tersimpan dalam sistem

#### **Alur Kejadian Normal**

	<ol style="list-style-type: none"> <li>1. Pengguna mengklik barang yang ingin dilaporkan</li> <li>2. Sistem menampilkan halaman informasi barang</li> <li>3. Pengguna mengklik tombol "Laporkan Barang"</li> <li>4. Sistem menampilkan <i>modal</i> berisi <i>input field</i> laporan</li> <li>5. Pengguna mengisi <i>fields</i> tersebut sesuai dengan konten laporan yang ingin disampaikan</li> <li>6. Setelah selesai, pengguna mengklik tombol "Laporkan"</li> <li>7. Sistem mengecek dan memvalidasi masukan pengguna</li> <li>8. Jika valid, sistem akan menampilkan <i>modal</i> Sukses Menyimpan Laporan</li> <li>9. Sistem <i>redirect</i> pengguna kembali ke halaman di 2</li> </ol>
--	--

#### **Alur Kejadian Alternatif**

	<b>Data masukan laporan pengguna tidak valid</b>
	<ol style="list-style-type: none"> <li>13a. Sistem mendeteksi masukan pengguna tidak valid.</li> <li>13c. Sistem menampilkan kembali modal di poin 6 beserta dengan <i>error message</i></li> </ol>

<b>Kode</b>	UC-04.04
<b>Nama</b>	<b>Mengirim Pesan</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna akan mengirimkan pesan kepada pengguna lainnya
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Pesan yang dikirimkan pengguna belum tersimpan pada sistem
<b><i>Postcondition</i></b>	Pesan yang dikirimkan pengguna berhasil tersimpan pada sistem
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna mengklik <i>URL</i> pengguna tujuan yang ingin dikirim pesan</li> <li>2. Sistem menampilkan halaman profil pengguna tujuan</li> <li>3. Pengguna mengklik tombol "Kirim Pesan"</li> <li>4. Sistem menampilkan halaman percakapan pengguna terhadap tujuan beserta riwayat percakapan pengguna dengan pengguna tujuan</li> <li>5. Pengguna memasukkan pesan yang ingin dikirimkan pada <i>field input</i> yang disediakan</li> <li>6. Setelah selesai, pengguna mengklik tombol 'Kirim'</li> <li>7. Sistem mengirim kepada koneksi socket</li> <li>8. Jika proses pengiriman kepada socket berhasil dan tidak ada gangguan, sistem kembali menampilkan halaman pengguna dengan informasi pesan yang sudah terkirim muncul di riwayat percakapan pengguna dengan pengguna tujuan</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Terjadi masalah teknis sehingga pesan tidak dapat terkirim</b>
	<ol style="list-style-type: none"> <li>7a. Sistem mendapatkan <i>exception</i> dari koneksi socket, bahwa pesan tidak dapat tersimpan</li> </ol>

<b>Kode</b>	UC-04.05
<b>Nama</b>	<b>Melihat dan Membaca Pesan</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin melihat daftar percakapan/ daftar perpesanan yang pernah dilakukan pengguna
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Daftar percakapan/ daftar perpesanan belum ditampilkan
<b>Postcondition</b>	Daftar percakapan/ daftar perpesanan berhasil ditampilkan

#### **Alur Kejadian Normal**

	<ol style="list-style-type: none"> <li>1. Pengguna mengklik tombol 'Conversations' di <i>navbar</i> aplikasi</li> <li>2. Sistem menampilkan halaman daftar percakapan pengguna</li> <li>3. Sistem memanggil fungsi AJAX untuk meminta data-data percakapan terakhir pengguna</li> <li>4. Balasan dari fungsi AJAX di <i>load</i> oleh <i>browser</i> untuk selanjutnya diparse ke dalam HTML</li> <li>5. Sistem menampilkan daftar percakapan pengguna</li> <li>6. Pengguna mengklik percakapan yang ingin dilihat/dibaca</li> <li>7. Sistem menampilkan etail percakapan pengguna dengan pengguna tujuan</li> </ol>
--	--

#### **Alur Kejadian Alternatif**

	<b>Koneksi terhadap soket <i>chat</i> tidak dapat dibangun/bermasalah</b>
	<ol style="list-style-type: none"> <li>7a. Sistem mendapatkan <i>exception</i> dari fungsi AJAX, bahwa daftar perpesanan/percakapan tidak dapat di<i>load</i></li> <li>7b. Sistem menampilkan kembali halaman pada poin 5, dengan <i>modal</i> berisikan <i>error message</i></li> </ol>

<b>Kode</b>	UC-04.06
<b>Nama</b>	<b>Memasukkan Kupon pada Transaksi</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin menggunakan kupon/voucher yang ia miliki untuk pada sebuah transaksi
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Pengguna belum berhasil mensubmit kode kupon ke dalam transaksi barang
<b>Postcondition</b>	Pengguna berhasil mensubmit kode kupon ke dalam transaksi barang
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman 'Riwayat Transaksi Lelang'</li> <li>2. Sistem menampilkan halaman Riwayat Transaksi Lelang pengguna</li> <li>3. Pengguna mengklik tombol 'Masukkan Kupon' pada transaksi yang diinginkan</li> <li>4. Sistem mengecek permintaan penggunaan kupon</li> <li>5. Jika permintaan dapat diverifikasi dan valid, sistem menampilkan <i>modal</i> berisi <i>input field</i> kupon</li> <li>6. Pengguna memasukkan kupon yang ingin dimasukkan, lalu mengklik tombol 'Submit'</li> <li>7. Sistem memvalidasi kupon voucher dan status barang</li> <li>8. Jika valid, sistem menerapkan penggunaan kupon ke dalam transaksi barang</li> <li>9. Sistem lalu menampilkan <i>modal</i> yang berisi informasi sukses penggunaan kupon pada transaksi</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.23** Spesifikasi Kasus Penggunaan : Mendaftarkan Barang Lelang

### 3.2.1.5 KP05. Manajemen Voucher



**Gambar 3.10** Diagram Kasus Penggunaan Manajemen Kupon

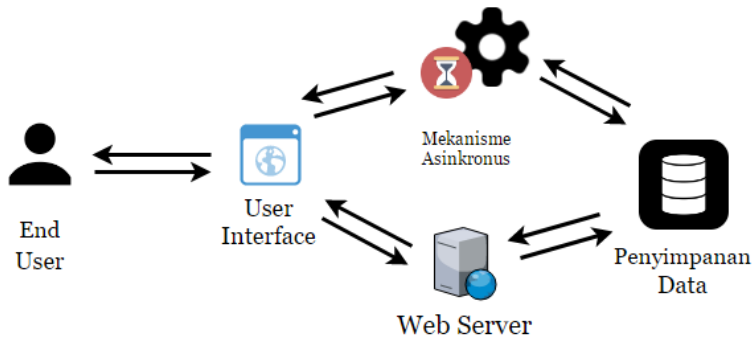
Kasus penggunaan ini seluruhnya digunakan oleh *administrator* aplikasi dan dilakukan di sistem terpisah. Kasus penggunaan ditujukan untuk mempermudah *administrator* dalam memanajemen voucher/kupon yang dibagikan oleh pengguna.

### 3.2.2 Identifikasi Komponen Fundamental

Berdasarkan Bab Analisa, dapat diidentifikasi dan divisualisasikan (pada gambar 3.11) komponen-komponen penting dalam pembuatan aplikasi, yaitu sebagai berikut:

1. Web Server
2. Mekanisme penyimpanan data (*database* dan *data storage*)
3. *User Interface* sebagai media terhadap *end-user*
4. Mekanisme Asinkronus untuk mengakomodasi fitur *realtime*

5. Verifikasi, Autentikasi dan Otorisasi (Keamanan) dalam implementasi program
6. Implementasi *User Behavior* dan HCI



**Gambar 3.11** Arsitektur dasar yang dibutuhkan untuk membangun aplikasi

### 3.2.3 *Technology Options*

Pada subbab sebelumnya, penulis sudah memaparkan arsitektur dasar yang dibutuhkan dalam rancang bangun aplikasi. Terkait dengan arsitektur dasar tersebut, banyak pilihan teknologi yang dapat mengimplementasikan arsitektur tersebut. Dalam pemaparan selanjutnya, akan dijelaskan alasan penulis *best practice* dalam pemilihan teknologi yang digunakan, didasarkan pada *best practices* dan pengalaman-pengalaman penulis. Keterkaitan dengan aspek-aspek yang dijelaskan sebelumnya pada subbab 3.1.4.

#### 3.2.3.1 NGINX sebagai Web Server

- Kelebihan
  1. Konfigurasi yang lebih *friendly* dan terstruktur
  2. Ketersediaan fitur yang lengkap & krusial (*reverse proxy*, memungkinkan skalabilitas & *load balancing*)

3. *Learning-gap* yang kecil terhadap pengalaman penulis/sudah familiar
- Opsi lainnya
  1. Apache2: Fiturnya kurang lengkap
  2. Node.js: *Learning-gap* yang besar bagi penulis/belum familiar
  3. Python: Belum Familiar, dan perlu eksplorasi fitur lebih dalam

### 3.2.3.2 POSTGRESQL untuk Penyimpanan Data

- Kelebihan
  1. *Learning gap* yang kecil
  2. Stabil karena telah digunakan dan dikembangkan oleh banyak *developer* selama bertahun-tahun
- Opsi lainnya
  1. SQL Server: Instalasi yang kompleks, penggunaan *resource* yang cukup besar

### 3.2.3.3 MONGODB untuk Penyimpanan Data Nontansaksional

- Kelebihan
  1. *Learning curve* yang mudah/sintaksnya kurang lebih sama dengan sintaks *database* transaksional pada umumnya
  2. Performa yang cepat karena menggunakan BSON
  3. Fitur yang lengkap untuk *sustainability* aplikasi seperti (Replikasi, Sharding, dll)
  4. *Handling* terhadap data yang sangat besar yang cukup bagus, cocok untuk data yang masif seperti *chatting*.
- Opsi lainnya
  1. Redis: Cepat, namun penyimpanan dilakukan di RAM sehingga lebih cocok untuk penyimpanan *auth session*,

- bukan untuk penyimpanan data yang sifatnya masif
2. Cassandra: *Learning-gap* yang besar, namun fiturinya lengkap untuk *data mining*

#### 3.2.3.4 CDN sebagai *Assets Sources*

- Kelebihan
  1. Akses cepat karena besar kemungkinan asset tersebut telah *dicache* sebelumnya dalam browser pengguna
  2. Mengurangi *bandwith* server
  3. Telah dioptimasi oleh pengembang masing-masing asset.
- Opsi lainnya
  1. Disimpan dalam server: Mengurangi *bandwith* server (*cost* meningkat)

#### 3.2.3.5 AWS S3 untuk *Content Growth Scalability*

- Kelebihan
  1. *Benefit* yang sangat *krusial*: keamanan, skalabilitas, *availability* - karena sudah dihandle langsung oleh pengembang *cloud computing* yang ahli di bidangnya
  2. Perkembangan jumlah konten yang akan disimpan (gambar barang yang diupload pengguna) tentunya bersifat sangat masif, sehingga tidak mungkin disimpan dalam server
- Opsi lainnya
  1. Disimpan dalam server: Mengurangi performa server karena sifatnya yang memakan *resource* cukup banyak, dan menambah *cost* untuk *upgrade server storage*

#### 3.2.3.6 SENDGRID untuk SMTP Relay

- Kelebihan



1. Konfigurasi yang mudah
  2. Dokumentasi yang cukup lengkap dan mudah ditemukan
  3. Fitur yang lengkap
  4. Adanya *free storage* dari akun Github Student Pack penulis
- Opsi lainnya
    1. MailChimp: Dokumentasi kurang lengkap, tidak ada *free storage* untuk akun penulis

### 3.2.3.7 VUE.JS untuk *Workloads Sharing*

- Kelebihan
  1. *Learning-gap* relatif kecil dibandingkan *Javascript tools* lainnya, karena didesain khusus untuk Laravel
  2. Adanya program utilitas (webpack) yang membuat performa Vue.js jauh lebih cepat
  3. Logika aplikasi dapat *diobfuscate* dengan webpack (*embedded* dalam Laravel)
- Opsi lainnya
  1. React: *Learning gap* dan *learning curve* yang sangat besar untuk penulis
  2. jQuery: tidak efektif karena *code smells* yang ditimbulkan cukup banyak

### 3.2.3.8 SOCKET.IO untuk Mekanisme Asinkronus

- Kelebihan
  1. *Learning-gap* relatif kecil dibandingkan *Javascript tools* lainnya, karena didesain khusus untuk Laravel
  2. Adanya program utilitas (webpack) yang membuat performa Vue.js jauh lebih cepat
  3. Logika aplikasi dapat *diobfuscate* dengan webpack (*embedded* dalam Laravel)

- Opsi lainnya
  1. React: *Learning gap* dan *learning curve* yang sangat besar untuk penulis
  2. jQuery: tidak efektif karena *code smells* yang ditimbulkan cukup banyak

### 3.2.3.9 JWT untuk Keamanan Soket

- Kelebihan
  1. *Library support* yang lengkap untuk komponen-komponen lainnya
  2. Efektif dan efisien karena tidak ada *query* ke database untuk autentikasi
- Opsi lainnya
  1. *Query* ke *database* secara konvensional: Sifat koneksi soket yang masif akan sangat memberatkan *database* jika setiap kali ada koneksi baru, harus melakukan *query database* sehingga tidak efektif
  2. *Session caching* dengan Redis: *Learning gap* yang besar

### 3.2.3.10 LARAVEL DUSK untuk *Functionality Testing Script*

- Kelebihan
  1. *Learning gap* yang kecil karena didesain sefamiliarmungkin dengan Laravel
- Opsi lainnya
  1. Selenium: *Learning curve* yang besar
  2. Phantom.js: *Learning curve* yang besar

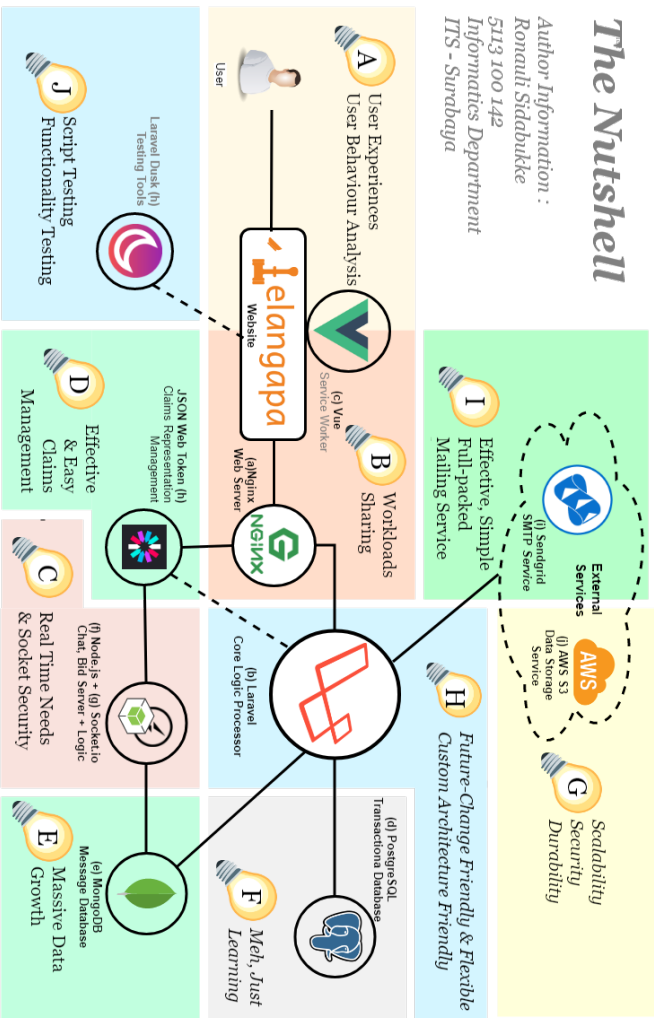
## 3.2.4 Arsitektur Perangkat Lunak

Sifat rancang bangun aplikasi ini yang bersifat *incremental* terhadap hasil analisa ilmiah dan pengalaman penulis sendiri. Sifat *incremental* tersebut berdampak perubahan arsitektur dan

pemilihan teknologi yang digunakan selama masa pengerjaan aplikasi. Subbab ini akan merangkum subbab 3.1.4 dan subbab 3.2.3 dalam bentuk sebuah diagram visualisasi. Rangkuman tersebut berupa visualisasi arsitektur digabungkan dengan teknologi yang digunakan. Selain itu, penulis juga menambahkan visualisasi warna untuk mempermudah pemahaman pembaca.

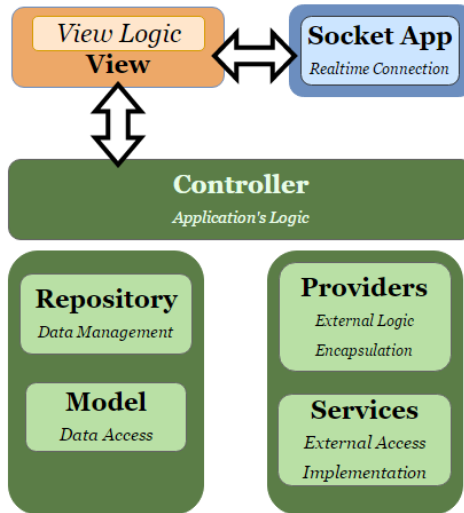
# The Nutshell

Author Information :  
 Ronauli Sidabukke  
 5113 100 142  
 Informatics Department  
 ITS - Surabaya



**Gambar 3.12** Visualisasi arsitektur dan teknologi Final yang diterapkan dalam rancang bangun aplikasi

### 3.2.5 Struktur Aplikasi



Gambar 3.13 Komponen Penyusun Struktur Aplikasi

- 3.2.5.1 **Komponen *View***
- 3.2.5.2 **Komponen *View Logic***
- 3.2.5.3 **Komponen *Controller***
- 3.2.5.4 **Komponen *Repository***
- 3.2.5.5 **Komponen *Model***
- 3.2.5.6 **Komponen *Socket App***
- 3.2.5.7 **Komponen *Providers***
- 3.2.5.8 **Komponen *Services***

### 3.2.6 Perancangan *Database Transaksional*

#### 3.2.6.1 *Conceptual Database Design*

#### 3.2.6.2 *Physical Database Design*

#### 3.2.6.3 *Optimization: Cache Tables*

### 3.2.7 Perancangan *Database Non-Transaksional*

#### 3.2.7.1 *Social Inboxes Design & Principles*

#### 3.2.7.2 *Physical Database Design*

### 3.2.8 Kamus Data *Database Transaksional*

#### 3.2.8.1 Kamus data tabel *administrator*

**Tabel 3.24** Spesifikasi Tabel Administrator

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK , Auto increment
2	name	varchar(255)	Nama
3	username	varchar(255)	Username yang digunakan
4	password	varchar(255)	Password
5	created_at	timestamp	Timestamp default laravel
6	updated_at	timestamp	Timestamp default laravel
7	email	varchar(255)	Email

#### 3.2.8.2 Kamus data tabel *administrator*

**Tabel 3.25** Kamus Data Tabel bids

No	Nama Atribut	Tipe Data	Keterangan
1	id	bigint	PK , Auto increment
2	id_user	int	FK ID user yang memberikan penawaran
3	id_item	bigint	FK ID barang yang ditawarkan
4	price_bid	bigint	Harga yang ditawarkan
5	win_status	bool	Status apakah bid ini memenangkan lelang
6	bid_timestamp	timestamp	Timestamp

### 3.2.8.3 Kamus data tabel *categories*

**Tabel 3.26** Kamus Data Tabel Categories

No	Nama Atribut	Tipe Data	Keterangan
1	id	bigint	PK, AutoIncrement
2	name	int	Nama Kategori
3	description	bigint	Deskripsi Kategori

### 3.2.8.4 Kamus data tabel *coupon*

**Tabel 3.27** Kamus Data Tabel Coupon

No	Nama Atribut	Tipe Data	Keterangan
1	coupon_id	int	PK, AutoIncr
2	<u>customerfacing</u> coupon	varchar(15)	String voucher yang diberikan kepada pengguna
3	activedate	timestamp	Tanggal aktif voucher
4	expireddate	timestamp	Tanggal nonaktif voucher
5	terms	varchar(255)	Deskripsi aturan penggunaan voucher
6	usages	bigint	Banyak penggunaan voucher
7	limit_usages	bigint	Batas penggunaan voucher
8	free_shipping	bool	Status apakah voucher berupa kupon <i>free shipping</i> atau tidak
9	price	float	Persentase diskon yang diberikan

### 3.2.8.5 Kamus data tabel *coupon\_usages*



**Tabel 3.28** Kamus Data Tabel Coupon Usages

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, Autoincrement
2	id_user	bigint	FK ID User yang menggunakan voucher
3	id_item	bigint	FK ID Item
4	id_coupon	int	FK ID Coupon yang digunakan
5	price_after	bigint	FK Harga setelah penggunaan voucher

### 3.2.8.6 Kamus data tabel *favorites*

**Tabel 3.29** Kamus Data Tabel Favorites

No	Nama Atribut	Tipe Data	Keterangan
1	id	bigint	PK, Autoincrement
2	id_user	int	FK ID User
3	id_item_favorite	bigint	FK ID Item
4	remove_status	bool	
5	time_listed	timestamp	timestamp

### 3.2.8.7 Kamus data tabel *issues*

**Tabel 3.30** Kamus Data Tabel Issues

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, Autoincrement
2	id_issue_type	int	FK ID tipe issue yang dilaporkan
3	id_user	int	FK ID user yang melaporkan
4	id_issued_object	int	FK ID item yang dilaporkan
5	description	text	Deskripsi Laporan

### 3.2.8.8 Kamus data tabel *issue\_types*

**Tabel 3.31** Kamus Data Tabel Issue Types

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, Autoincrement
2	name	varchar(50)	Nama tipe laporan
3	priority	int	Prioritas laporan

### 3.2.8.9 Kamus data tabel *items*

**Tabel 3.32** Kamus Data Tabel Items

No	Nama Atribut	Tipe Data	Keterangan
1	id	bigint	PK, Autoincrement
2	id_user	int	FK ID User

Tabel 3.32 – lanjutan dari halaman sebelumnya

No	Nama Atribut	Tipe Data	Keterangan
3	name	varchar(255)	Nama barang
4	description	text	Deskripsi barang
5	starting_price	bigint	Harga awal lelang
6	expected_price	bigint	Harga akhir yang diinginkan
7	start_time	timestamp	Awal waktu lelang
8	end_time	timestamp	Akhir waktu lelang
9	id_category	int	FK - ID kategori barang
10	<u>winner_chosen_status</u>	bool	Status terpilih pemenang lelang
11	id_user_winner	int	FK - ID user pemenang lelang
12	bid_price	bigint	Harga penawaran lelang tertinggi pada barang tersebut
13	<u>end_time_after_extended</u>	timestamp	Waktu akhir extended
14	status_extend	int	Status extend waktu lelang barang
15	cancel_status	bool	Status cancel lelang barang

Tabel 3.32 – lanjutan dari halaman sebelumnya

No	Nama Atribut	Tipe Data	Keterangan
16	id_coupon_usage	int	FK - ID kupon yang digunakan pada barang tersebut

### 3.2.8.10 Kamus data tabel *rating\_auctioneers*

Tabel 3.33 Kamus Data Tabel Rating Auctioneers

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, Autoincrement
2	<u>id_user</u> _auctioneer	varchar(50)	FK ID User Penjual
3	id_user_rater	int	FK ID User pemberi review
4	<u>nama_user</u> _auctioneer	int	Nama user penjual
5	nama_user_rater	int	Nama user pemberi rating
6	rate	bigint	Jumlah rating yang diberikan
7	id_item	int	FK ID barang yang dirating
8	rate_message	text	Deskripsi Rate

### 3.2.8.11 Kamus data tabel *ratingbidders*

**Tabel 3.34** Kamus Data Tabel Rating Bidders

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, Autoincrement
2	id_user_bidder	varchar(50)	FK ID User Pelelang
3	id_user_rater	int	FK ID User pemberi review
4	nama_user_bidder	int	Nama user pelelang
5	nama_user_rater	int	Nama user pemberi rating
6	rate	bigint	Jumlah rating yang diberikan
7	id_item	int	FK ID barang yang dirating
8	rate_message	text	Deskripsi Rate

### 3.2.8.12 Kamus data tabel *ratinglogs*

**Tabel 3.35** Kamus Data Tabel Rating's Logs

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, Autoincrement
2	id_user_winner	varchar(50)	FK ID User pemenang lelang
3	id_user_auctioneer	int	FK ID User Penjual
4	id_item	int	FK ID barang yang dirating

Tabel 3.35 – lanjutan dari halaman sebelumnya

No	Nama Atribut	Tipe Data	Keterangan
5	<u>status_rating</u> <u>_from_winner</u>	int	Flag apakah transaksi sudah dirating oleh pemenang lelang
6	<u>status_rating</u> <u>_from_auctioneer</u>	bigint	Flag apakah transaksi sudah dirating oleh penjual barang
7	id_rating_bidder	int	FK ID rating di tabel ratingauctioneers
8	<u>id_rating</u> <u>_auctioneer</u>	text	FK ID rating di tabel ratingbidders

### 3.2.8.13 Kamus data tabel *userverifications*

Tabel 3.36 Kamus Data Tabel User Verifications

No	Nama Atribut	Tipe Data	Keterangan
1	id_user	int	FK ID User
2	created_at	timestamp	Timestamp
3	token	varchar(255)	Token random yang digenerate sistem

### 3.2.8.14 Kamus data tabel *userissues*

**Tabel 3.37** Kamus Data Tabel Issue Types

<b>No</b>	<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Keterangan</b>
1	id	int	PK, Autoincrement
2	id_user	int	FK - ID user yang melaporkan
3	id_issued_user	int	FK - ID User yang dilaporkan
4	description	text	Deskripsi laporan yang diberikan

### 3.2.8.15 Kamus data tabel *users*

**Tabel 3.38** Kamus Data Tabel Issue Types

<b>No</b>	<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Keterangan</b>
1	id	int	PK, Autoincrement
2	name	varchar(255)	Nama
3	email	varchar(50)	Email
4	password	varchar(255)	Password
5	address	text	Alamat
6	phone	varchar(50)	No. HP
7	created_at	timestamp	Timestamp
8	updated_at	timestamp	timestamp
9	username	varchar(20)	Username

Tabel 3.38 – lanjutan dari halaman sebelumnya

No	Nama Atribut	Tipe Data	Keterangan
10	verification	timestamp	Tanggal user terverifikasi - jika belum diverifikasi, isinya NULL
11	is_banned	timestamp	Status banned pengguna

### 3.2.9 Kamus Data *Database* Non-Transaksional

#### 3.2.9.1 Kamus data *collection* userchats

Tabel 3.39 Kamus data tabel Userchat

No	Nama Atribut	Tipe Data	Keterangan
1	id	-	Hash generated key
2	room	varchar	Room chat
3	sender	int	FK ID user pengirim sender
4	msg	int	FK ID user receiver
5	sent	ISODate	timestamp pesan diterima d

#### 3.2.9.2 Kamus data *collection* chatroom

Tabel 3.40 Kamus Data Tabel Chatroom

No	Nama Atribut	Tipe Data	Keterangan
1	id	-	Hash generated
2	room	varchar	Room chat



Tabel 3.40 – lanjutan dari halaman sebelumnya

No	Nama Atribut	Tipe Data	Keterangan
3	id_user_1	int	FK ID user pengirim sender
4	id_user_2	int	FK ID user receiver
5	created_at	ISODate	timestamp pesan pertama
6	updated_at	ISODate	timestamp pesan terakhir

### 3.2.9.3 Kamus data *collection* itemimages

Tabel 3.41 Kamus data tabel Itemimage

No	Nama Atribut	Tipe Data	Keterangan
1	id	-	PK - Generated key
2	item_id	int	FK - ID barang
3	url	int	Url gambar

### 3.2.10 Kamus Data

*(Halaman ini sengaja dikosongkan)*

## BAB IV

### IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi aplikasi, dibagi menjadi empat subbab yaitu :

1. Implementasi Perangkat Keras / *Deployment*
2. Implementasi Perangkat Lunak
3. Implementasi Antarmuka / *User Interface*

#### 4.1 Implementasi Perangkat Keras/ *Deployment*

Aplikasi dideploy secara *online*, dalam sebuah *Virtual Private Server* yang dihost oleh *Digital Ocean*. Spesifikasi VPS yang digunakan adalah sebagai berikut:

1. **Hardware**
  - (a) CPU: Intel(R) Xeon(R) CPU E5-2630L v2 @ 2.40GHz
  - (b) Operating System:
  - (c) RAM: 512MB
  - (d) Storage Space: 20GB
2. **Operating System**
  - (a) Architecture: 64bit
  - (b) Kernel Version: Linux 4.4.0-75-generic x86 64
  - (c) OS Version: Ubuntu 16.04.2 LTS Xenial
3. **Networking Stats**
  - (a) Tersambung ke Internet: Ya
  - (b) IP Publik: Ya
  - (c) Alamat IP Publik (IPv4): 188.166.179.2
  - (d) *Average Download Speed*: 1371 Mbit/s
  - (e) *Average Upload Speed*: 860.12 Mbit/s
  - (f) DNS: Google
4. **Domain Stats**
  - (a) HTTPS Support: Yes
  - (b) SSL Certificate issued by: Avast
  - (c) Domain: <https://Lelangapa.com>

- (d) Testing-purpose subdomain:  
<https://testing.lelangapa.com>  
 (e) Domain issued by: Namecheap

## 4.2 Implementasi Perangkat Lunak

Pada subbab ini, penulis akan memaparkan mengenai spesifikasi dan pemasangan perangkat lunak yang dibutuhkan dalam rancang bangun aplikasi lelang online ini.

### 4.2.1 Strategi *Deployment*

Pada saat instalasi dan konfigurasi beberapa komponen perangkat lunak, tidak tidak keseluruhan prosesnya berjalan dengan baik dalam sekali percobaan. Berikut paparan beberapa kesulitan dan penyelesaiannya.

#### 4.2.1.1 NGINX

##### 1. CORS (*Cross Origin Resource Sharing*)

CORS adalah sebuah mekanisme yang memungkinkan sebuah *website* menggunakan *resources* (seperti skrip Javascript, *fonts*, dll) untuk diakses dari sumber lain selain *domain originnya*.

Secara teknis, CORS mendefinisikan *protokol/cara* browser dan server untuk berinteraksi otorisasi permintaan *resource* dari domain lain, dan juga lebih aman karena *developer* dapat mengontrol otorisasi tersebut (daripada mengizinkan semua permintaan).

#### ***The Problem***

Masalah koneksi ke server lelang (yang berjalan pada domain yang sama, namun port yang berbeda) tidak dapat tersambung karena *error* berikut :

```
1 No 'Access-Control-Allow-Origin' header
    ↳ is present on the requested
    ↳ resource.
```

### ***Insight***

Selama 2 minggu kurang lebih penulis mencoba cara yang ditemukan penulis dalam situs stackoverflow untuk mengkonfigurasi *server* lelang agar otorisasi CORS dapat dilakukan, namun tidak ada hasil.

Masalah ini terselesaikan setelah menggunakan fitur *reverse proxy* dari Nginx.

### ***Solution***

Strateginya adalah sebagai berikut :

- (a) Server Lelang berjalan pada localhost port 3000.
- (b) Kita mengatur sebuah subdomain khusus – misalkan A.domain.com
- (c) Dalam NGINX , kita konfigurasi agar semua *requests* menuju A.domain.com ke aplikasi localhost yang kita maksud di poin 1a
- (d) Selain meneruskan *requests*, NGINX juga akan meneruskan *reply* dari server lelang tersebut kepada *client/origin* yang meminta *request* tersebut.

#### **4.2.1.2 VUE.JS**

##### **1. *Package Dependencies***

Pada versi terbaru Laravel (5.4\*), Laravel secara *default* menyertakan *package* Laravel Mix - yaitu fitur untuk *compiling assets* dengan Webpack, dengan hasil akhir *compiled assets* (terutama *script* Javascript) yang eksekusinya jauh lebih cepat, karena menggunakan V8 – sebuah *engine* Javascript yang telah dioptimasi yang

bersifat *just-in-time* (JIT) yang memproduksi *machine code* dari sebuah *script* Javascript lalu dieksekusi.

### ***Main Problem***

Masalah muncul saat versi Laravel yang digunakan untuk membangun aplikasi adalah versi (5.3) – dan jika Laravelnya *diupgrade*, tidak ada jaminan bahwa *deprecated dependencies* (keadaan dimana sebuah *package* tidak *disupport* oleh versi terbaru) – yang berarti harus *refactoring code* yang pasti memakan waktu lama.

### ***Insights***

Penulis menganalisa perbedaan mendasar *package.json* antara Laravel 5.3 dan 5.4 adalah sebagai berikut:

- (a) Basis : Perubahan basis yang awalnya Gulp menjadi Webpack
- (b) *Dependencies* : Webpack ternyata menggunakan beberapa plugin tambahan yang tidak diakomodasi dalam *package.json* di versi 5.3
- (c) *Run Script* : Terdapat beberapa perubahan signifikan terhadap *run script alias* di versi 5.4 - dibandingkan pada versi 5.3.
- (d) *Additional Files* : Terdapat beberapa file konfigurasi tambahan agar proses kompilasi aset dapat berjalan dengan baik.

### ***Solution***

Penulis lalu mengoreksi dan *update package.json* dengan pendekatan *trial and error*, dan bisa terselesaikan dengan script berikut :

```
1 { "private": true,
```

```

2   "scripts": {
3     "_comment" : "Lists of running
                      ↪ npm commands defined here"
4   },
5   "devDependencies": {
6     "axios": "^0.15.3",
7     "bootstrap-sass": "^3.3.7",
8     "cross-env": "^3.2.3",
9     "jquery": "^3.1.1",
10    "laravel-mix": "0.*",
11    "lodash": "^4.17.4",
12    "vue": "^2.1.10"
13  },
14  "dependencies": {
15    "vue-resource": "^1.3.1"
16  } }

```

## 2. *Dependencies Optimization Problem*

Setelah menulis beberapa *script* Vue, penulis menyadari bahwa setiap *script* Vue ternyata mempunyai *dependencies* yang sama, yaitu axios, Promise, toastr dan vue. Setiap file Vue *include* sebuah *script* yang berisi:

```

1   window.axios = require('axios');
2   window.toastr = require('toastr');
3   window. = require('toastr');
4   require('vue-resource');

```

Hal ini mengakibatkan semua file vue yang *compile* ukurannya cukup besar ( 400kb), padahal sebenarnya di dalam setiap file tersebut sebenarnya ada yang sama. Hal

ini tentu tidak efektif, karna sebenarnya hal-hal yang sama tersebut bisa dipisahkan, dan dijadikan *cache* sehingga *loading* halaman bisa jauh lebih cepat.

### ***Insight & Solution***

Setelah penulis menanyakan dan mendiskusikan di forum Slack, beberapa pengguna Vue menyarankan untuk *compile* keseluruhan *dependencies* yang digunakan kedalam satu file terpisah, dan hanya menulis logika Vue untuk setiap file Vue.

Isi file `webpack.mix.js` (file yang *dicompile* oleh Webpack) menjadi seperti berikut.

```

1  /* dependencies all compiled into one
   ↪ single file */
2  mix.js('scripts/dependencies.js', '
   ↪ public/js');
3
4  /* dependencies all compiled into one
   ↪ single file */
5  mix.js('scripts/favorites.js', 'public/
   ↪ js');
6  mix.js('scripts/other_vue_script.js', '
   ↪ public/js');
```

Dan di HTML, untuk *including script* dituliskan seperti berikut:

```

1  <script src="dependencies.js" ></script>
2  <script src="custom_page_script.js" >
   ↪ </script>
```



#### 4.2.1.3 *Library* JWT pada Laravel dan Node.js

#### 4.2.1.4 *Whitelisting* pada SENDGRID

### 4.2.2 Diagram Kelas

## 4.3 Implementasi Antarmuka / *User Interface*

### 4.3.1 Antarmuka Registrasi

Penjelasan otorisasi terhadap antarmuka A, link yang tersedia dalam antarmuka A, dan penjelasan *exception* jika terjadi masalah baik otorisasi ataupun autentikasi saat mengakses antarmuka ini.

```

1: initializeSwarm()
2: initializeLeadersArchive()
3: generation = 0
4: while generation < maxGenerations do
5:     computeSpeed()
6:     updatePosition()
7:     mutation() // Turbulence
8:     evaluation()
9:     updateLeadersArchive()
10:    updateParticlesMemory()
11:    generation ++
12: end while
13: returnLeadersArchive()

```

**Gambar 4.1** Pseudocode Controller untuk Menampilkan Antarmuka A

### 4.3.2 Antarmuka Halaman B

Penjelasan otorisasi terhadap antarmuka B, link yang tersedia dalam antarmuka B, dan penjelasan *exception* jika terjadi masalah baik otorisasi ataupun autentikasi saat mengakses antarmuka ini.

```
1: initializeSwarm()  
2: initializeLeadersArchive()  
3: generation = 0  
4: while generation < maxGenerations do  
5:     computeSpeed()  
6:     updatePosition()  
7:     mutation() // Turbulence  
8:     evaluation()  
9:     updateLeadersArchive()  
10:    updateParticlesMemory()  
11:    generation ++  
12: end while  
13: returnLeadersArchive()
```

**Gambar 4.2** Pseudocode Controller untuk Menampilkan Antarmuka B

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

#### **5.1 Pengujian**

Pada subbab ini, penulis akan memaparkan pengujian terhadap aplikasi. Pengujian yang dilakukan adalah pengujian fungsionalitas, dimana penulis menggunakan tool Laravel Dusk sebagai *testing code* untuk menguji fungsionalitas aplikasi.

Dikarenakan keterbatasan waktu, dan atas saran dari pembimbing, penulis tidak menuliskan *testing script* untuk keseluruhan fungsionalitas yang sudah pasti teruji, seperti Login (sudah menggunakan *facade* Laravel), transaksi CRUD dll.

Pada pemaparan ini, penulis mengidentifikasikan fungsionalitas utama dalam aplikasi lelang ini adalah sebagai berikut :

1. Pengujian Fungsionalitas Lelang
  - (a) Pengujian Penawaran Lelang
2. Pengujian Fungsionalitas Voucher
  - (a) Pengujian Penggunaan Voucher

Pada bagian ini juga, penulis menuliskan *summary* pengujian fungsionalitas ini pada .

##### **5.1.1 Pengujian Fungsionalitas Lelang**

Pada pengujian ini, terdapat beberapa skenario pengujian yang dipaparkan dalam tabel berikut :

##### **5.1.2 Pengujian Fungsionalitas Voucher**

Pada pengujian ini, terdapat beberapa skenario pengujian yang dipaparkan dalam tabel berikut :

### **5.1.3 Summary Pengujian**

## **5.2 Evaluasi**

Pada subbab ini, penulis akan memaparkan hasil analisa terhadap aplikasi, perspektif non-IT terhadap pengerjaan maupun lingkup pekerjaan dari aplikasi Lelang Online ini.

### **5.2.1 Evaluasi Aspek Kebutuhan**

### **5.2.2 Evaluasi Aspek Teknis**

### **5.2.3 Evaluasi Aspek *Bussiness Engineering***

#### **5.2.3.1 Evaluasi Regulasi Terkait**

#### **5.2.3.2 *Market Analysis***

#### **5.2.3.3 *Competitors Competitiveness***

#### **5.2.3.4 Evaluasi *Startup Burn Rate***

### **5.2.4 Evaluasi Aspek Analisa dan Desain**

### **5.2.5 Evaluasi Aspek Implementasi**

### **5.2.6 Evaluasi Aspek Pengujian**

### **5.2.7 *Summary Evaluasi***

### **5.2.8 *Further Enhancements***

### **5.2.9 *Further Readings***

## **BAB VI**

### **PENUTUP**

Bab ini membahas kesimpulan yang dapat diambil dari tujuan pembuatan sistem dan hubungannya dengan hasil uji coba dan evaluasi yang telah dilakukan. Selain itu, terdapat beberapa saran yang bisa dijadikan acuan untuk melakukan pengembangan dan penelitian lebih lanjut.

#### **6.1 Kesimpulan**

Dari proses perancangan, implementasi dan pengujian terhadap sistem, dapat diambil beberapa kesimpulan berikut:

1. Kesimpulan 1
2. Kesimpulan 2
3. Kesimpulan 3

#### **6.2 Saran**

Berikut beberapa saran yang diberikan untuk pengembangan lebih lanjut:

- Menggunakan mekanisme *Queue* sebagai *countermeasure* dari masalah *occurence* (di Laravel sudah ada disediakan *base classnya* sendiri).
- Mengikutsertakan pihak yang menguasai/spesialisasi di bidang hukum untuk menetapkan peraturan-peraturan terkait
- Saran 2
- Saran 3

## Pustaka

- W. (2016, January). *Test Script* [Online Encyclopedia]. Retrieved from [https://en.wikipedia.org/wiki/Test\\_script](https://en.wikipedia.org/wiki/Test_script) (translated by author)
- Apa itu Laravel?* (2016). Retrieved from <http://emka.web.id/tutorial/tutorial-laravel/2013/tutorial-laravel-1-apa-itu-laravel/>
- Arti pengertian BID, INC, BIN dalam Bahasa Lelang Website.* (n.d.). Retrieved from <http://www.bocahsoeniyi.com/2015/02/arti-pengertian-bid-inc-bin-dalam-lelang-website.html>
- Azurri, A. (2016, October). *Node Package Manager* [Blog]. Retrieved from <http://www.azuwir.com/2016/10/nodejs-node-package-manager-npm/>
- BalaiLelang. (n.d.). *Sejarah Lelang di Indonesia*. Retrieved from <https://www.balailelang.co.id/index.php/sejarah-lelang/sejarah-lelang-di-indonesia>
- D. VanHoose, D. (2011). *E-Commerce Economics* (2nd ed.). Routledge. Retrieved from <https://books.google.co.id/books?id=doKsAgAAQBAJ&hl=id>
- Google Developers, W. F. (2017, April). *Service Worker : Pengantar*. Google Developers. Retrieved from <https://developers.google.com/web/fundamentals/getting-started/primers/service-workers?hl=id>
- insideBIGDATA, E. T. (2017, February). *Exponential Growth of Data* [Blog & News]. Retrieved from <https://insidebigdata.com/2017/02/16/the-exponential-growth-of-data/>
- JWT* [Blog]. (2016, June). Retrieved from <http://blog.nostrattech.com/2016/06/jwt-json-web-token.html>

- Kuo, Y.-F., Yen, S.-T., & Chen, L.-H. (2016, October). Online auction service failures in Taiwan: Typologies and recovery strategies. *Science Direct*, 10, 183–193.
- Laia Bonastre, T. G. (2014). A set of heuristics for user experience evaluation in e-commerce websites. *ACHI 2014 : The Seventh International Conference on Advances in Computer-Human Interactions*, 27–34.
- Laravel. (n.d.). *Browser Testing* [Documentation]. Retrieved from <https://laravel.com/docs/5.4/dusk>
- MongoDB [Online Encyclopedia]. (2017, January). Retrieved from <https://id.wikipedia.org/wiki/MongoDB>
- Node.js (Wikipedia). (2014, November). Retrieved from <https://id.wikipedia.org/wiki/Node.js>
- PostgreSQL. (n.d.). Retrieved from <http://www.belajarsql.com/postgresql>
- Pratama, R. (2012, February). *Lelang* [Blog]. Retrieved from <http://raypratama.blogspot.co.id/2012/02/lelang.html>
- Repository Pattern - Coding4ever's Blog* [Educational Blog]. (2016, February). Retrieved from <http://coding4ever.net/blog/2016/02/22/repository-pattern/>
- Shaw, M. (1996). Some Patterns for Software Architectures. *Pattern Languages of Program Design*, 2, 255–269. Retrieved from [https://sophia.javeriana.edu.co/~cbustaca/docencia/DEAS-2015-03/documentos/1996\\_Shaw\\_Some%20Patterns%20for%20Software%20Architectures.pdf](https://sophia.javeriana.edu.co/~cbustaca/docencia/DEAS-2015-03/documentos/1996_Shaw_Some%20Patterns%20for%20Software%20Architectures.pdf)
- Sistem Basis Data* [Blog]. (2013, March). Retrieved from <https://rezapratama8.wordpress.com/2013/05/19/sistem-basis-data/>
- Sistem Lelang Online Masih Bermasalah*. (n.d.). Retrieved from <http://www.saidabdullah.info/index.php/issues/394-sistem-lelang-online-masih>

-bermasalah

*SMTP*. (n.d.). Retrieved from <http://techscape.co.id/hosting/smtpt.ts>

*Socket.io (Wikipedia)*. (2016, January). Retrieved from <https://en.wikipedia.org/wiki/Socket.IO>

Sommerville, I. (2009). *Software Engineering* (9th ed.). United States of America: Addison-Wesley. All rights reserved.

Web Services, A. (n.d.). *Amazon Benefits* [Bussiness]. Retrieved from <https://aws.amazon.com/application-hosting/benefits/>

Wikipedia. (n.d.). *NoSQL* [Online Encyclopedia]. Retrieved from <https://id.wikipedia.org/wiki/NoSQL>

Wikipedia. (2016a, January). *Amazon Web Services* [Wikipedia]. Retrieved from [https://id.wikipedia.org/wiki/Amazon\\_Web\\_Services](https://id.wikipedia.org/wiki/Amazon_Web_Services)

Wikipedia. (2016b, January). *Lelang*. Retrieved from <https://id.wikipedia.org/wiki/Lelang>

Wikipedia. (2017a, March). *SendGrid* [Online Encyclopedia]. Retrieved from <https://en.wikipedia.org/wiki/SendGrid>

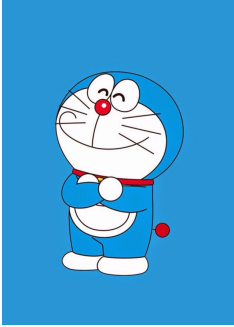
Wikipedia. (2017b, May). *Vue.js* [Online Encyclopedia]. Retrieved from <https://en.wikipedia.org/wiki/Vue.js> (translated to Indonesia)

Yudana. (2015, September). *Redis Lebih Dari Sekedar Cache*. Retrieved from <http://www.yudana.id/redis-lebih-dari-sekedar-cache/>



## BIODATA PENULIS

### Ronauli



**Silva NS**, seorang yang lahir & besar di Siantar Medan - Sumatera Utara, sangat suka belajar. Diberi amanah untuk menjadi *administrator* Laboratorium Pemrograman di tahun 2015, penulis belajar banyak mengenai administrasi *server*, rancang bangun aplikasi terutama di bidang web. Selain itu, beberapa *project* yang diambil penulis mengenai rancang bangun aplikasi yang baik dan buruk yang mengajarkan

penulis cara memperbaiki, menangkal dan & mengoptimasinya. Selain itu, penulis juga banyak belajar *softskills* saat menjabat sebagai Sekretaris Departemen HMTTC ITS 2015/2016 (Kabinet Optimasi) dan lewat pelatihan-pelatihan yang diberikan donatur beasiswa saat penulis masih diamanahi sebagai beswan Karya Salemba Empat 2014-2016.

Motto penulis yaitu "*Always go for the extra miles*", membawa penulis mengambil topik tugas akhir ini, dimana penulis dapat menerapkan perbaikan, optimasi dan pelajaran yang penulis petik dari *project-project* sebelumnya, dengan bimbingan dosen-dosen pembimbing penulis yang baik hati. Dalam pendalaman topik tugas akhir ini juga, penulis banyak belajar dan menjadi sangat tertarik mendalami *bussiness engineering*, *user experiences and usability*, dan *data scientist & engineering*.

Dengan segala kerendahan hati, ilmu penulis masihlah setitik dibandingkan susu sebelanga. Penulis sangat mengharapkan diskusi, ajaran dan bantuan dalam memperbaiki diri. Apabila pembaca berkenan, penulis dapat dihubungi melalui *email* ke [ronayumik@gmail.com](mailto:ronayumik@gmail.com), atau *Whatsapp* ke nomor +62821-6066-5568.