



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.

Key-Word: *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, Juni 2017

Ronauli Silva N. Sidabukke

DAFTAR ISI

ABSTRAK	7
ABSTRACT	8
KATA PENGANTAR	9
DAFTAR ISI	11
DAFTAR TABEL	13
DAFTAR GAMBAR	15
I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	2
I.3 Batasan Masalah	2
I.4 Tujuan	2
II LANDASAN TEORI	3
II.1 Lelang	3
II.2 <i>Vendu Reglement</i>	3
II.3 Lelang Daring / Lelang <i>Online</i>	3
II.4 <i>Repository Pattern</i>	4
II.5 <i>Data Growth</i>	5
II.6 <i>Concurrency</i>	5
II.7 NoSQL	6
II.8 Protokol SMTP	6
II.9 Whitelist	6
II.10 JSON <i>Web Token</i>	7
II.11 <i>Service Worker</i>	7
II.12 Laravel	7
II.13 Vue.js / Vue	8
II.14 Node.js	8
II.15 npm / <i>Node Package Manager</i>	9

II.16	Socket.io	9
II.17	PostgreSQL	9
II.18	MongoDB	10
II.19	Redis	10
II.20	SendGrid	11
II.21	Amazon Web Service	11
II.22	<i>Test Script</i>	12
II.23	Laravel Dusk	12
III	ANALISA DAN PERANCANGAN	13
III.1	Analisa	13
III.1.1	<i>Requirement Analysis</i>	13
III.1.2	<i>Technical Analysis</i>	41
III.1.3	Analisa Pribadi Penulis	42
III.1.4	Perancangan Sistem	47
III.2	Perancangan Sistem	47
III.2.1	Identifikasi Komponen Fundamental . . .	47
III.2.2	<i>Technology Options</i>	48
III.2.3	Arsitektur Perangkat Lunak	52
III.2.4	Struktur Penyusun Aplikasi	55
III.2.5	Perancangan <i>Database</i> Transaksional . . .	56
III.2.6	Perancangan <i>Database</i> Non-Transaksional	56
III.2.7	Kamus Data <i>Database</i> Non-Transaksional	56
III.2.8	Kamus Data <i>Database</i> Non-Transaksional	56
IV	IMPLEMENTASI	57
IV.1	Implementasi Perangkat Keras/ <i>Deployment</i> . . .	57
IV.2	Implementasi Perangkat Lunak	58
IV.2.1	Strategi <i>Deployment</i>	58
IV.2.2	Diagram Kelas	63
IV.3	Implementasi Antarmuka / <i>User Interface</i>	63
IV.3.1	Antarmuka Registrasi	63
IV.3.2	Antarmuka Halaman B	63

V	PENGUJIAN DAN EVALUASI	65
V.1	Pengujian	65
V.1.1	Pengujian Fungsionalitas Lelang	65
V.1.2	Pengujian Fungsionalitas Voucher	65
V.1.3	<i>Summary</i> Pengujian	66
V.2	Evaluasi	66
V.2.1	Evaluasi Aspek Kebutuhan	66
V.2.2	Evaluasi Aspek Teknis	66
V.2.3	Evaluasi Aspek <i>Bussiness Engineering</i>	66
V.2.4	Evaluasi Aspek Analisa dan Desain	66
V.2.5	Evaluasi Aspek Implementasi	66
V.2.6	Evaluasi Aspek Pengujian	66
V.2.7	<i>Summary</i> Evaluasi	66
V.2.8	<i>Further Enchancements</i>	66
V.2.9	<i>Further Readings</i>	66
VI	PENUTUP	67
VI.1	Kesimpulan	67
VI.2	Saran	67

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

III.1	Kebutuhan Fungsional Aplikasi Lelang Online . .	19
III.2	Kebutuhan Non-Fungsional Aplikasi Lelang Online	20
III.3	Tabel Kasus Penggunaan	22
III.4	Spesifikasi Kasus Penggunaan Registrasi	23
III.5	Spesifikasi Kasus Penggunaan Login	24
III.6	Spesifikasi Kasus Penggunaan: Konfirmasi Email	25
III.7	Spesifikasi Kasus Penggunaan : Melihat barang yang dilelang	26
III.8	Spesifikasi Kasus Penggunaan : Mencari Barang Lelang	27
III.9	Spesifikasi Kasus Penggunaan : Mencari Barang Lelang	28
III.10	Spesifikasi Kasus Penggunaan : Melihat Riwayat Penawaran Lelang Barang	29
III.11	Spesifikasi Kasus Penggunaan : Mendaftarkan Barang Lelang	31
III.12	Spesifikasi Kasus Penggunaan : Mendaftarkan Barang Lelang	32
III.13	Spesifikasi Kasus Penggunaan : Melihat Barang yang Pernah Didaftarkan	33
III.14	Spesifikasi Kasus Penggunaan : Melihat Riwayat Penawaran Harga	34
III.15	Spesifikasi Kasus Penggunaan : Melihat Review Pengguna	35
III.16	Spesifikasi Kasus Penggunaan : Memberikan Review Pengguna	36
III.17	Spesifikasi Kasus Penggunaan : Melaporkan Barang	37
III.18	Spesifikasi Kasus Penggunaan : Mengirimkan Pesan	38
III.19	Spesifikasi Kasus Penggunaan : Melihat & Membaca Pesan	39
III.20	Spesifikasi Kasus Penggunaan : Mendaftarkan Barang Lelang	40

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

III.1	Fatalitas kegagalan dalam aplikasi Lelang Online, Kepuasan terhadap Perbaikan Pelayanan dan <i>Repeat Purchase Intention</i> setelah Perbaikan Layanan	17
III.2	Kategori Perbaikan terhadap Kegagalan Pelayanan Lelang Online	18
III.3	<i>Essential attributes of good software</i>	42
III.4	Arsitektur dasar yang dibutuhkan untuk membangun aplikasi	48
III.5	<i>Visualisasi arsitektur dan teknologi Final yang diterapkan dalam rancang bangun aplikasi</i>	54
IV.1	Pseudocode Controller untuk Menampilkan Antarmuka A	63
IV.2	Pseudocode Controller untuk Menampilkan Antarmuka B	64

(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.

I.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*) (?, ?).

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 (?, ?).

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*.

I.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 (?, ?)?

I.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.

I.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.

BAB II

LANDASAN TEORI

II.1 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 (?, ?).

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(?, ?).

II.2 *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(?, ?).

II.3 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 (?, ?). 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 (?, ?)

II.4 *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 meredirect request dan validasi request)(?, ?).

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.*" (?, ?).

II.5 *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* (? , ?).

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(? , ?).

II.6 *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* (?, ?)

II.7 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” (?, ?).

II.8 Protokol 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 (?, ?).

II.9 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 (?, ?).

II.10 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) (?, ?).

II.11 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 mencegat dan menangani permintaan jaringan, termasuk mengelola cache respons lewat program(?, ?).

II.12 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-fiendly*
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* (?, ?)

II.13 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*(?, ?).

II.14 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 (?, ?).

II.15 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(?, ?).

II.16 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 (?, ?).

II.17 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 (?, ?)

II.18 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(?, ?).

II.19 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

kelebihannya, 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* (?, ?).

II.20 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*. (?, ?)

II.21 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(?, ?).

Dalam website resminya, disebutkan bahwa AWS dapat membantu aplikasi menjadi lebih cepat, lebih aman, dan

menghemat *costs* dengan *scaling performance* menggunakan teknologi *cloud computing*(?, ?).

II.22 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 (?, ?).

Ada beberapa jenis *script test*:

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

II.23 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 tetap bisa menggunakan driver Selenium yang kompatibel (?, ?).

BAB III

ANALISA DAN PERANCANGAN

III.1 Analisa

Pada subbab ini, penulis akan memaparkan analisa-analisa kebutuhan, teknis dan analisa pribadi serta pengaruhnya dalam perancangan aplikasi pada subbab III.2. Secara garis besar, subbab ini menjelaskan **a)** analisa kebutuhan fungsional, **b)** analisa aspek bisnis, **c)** spesifikasi kebutuhan, dan **d)** rancangan arsitektur dan teknologi yang digunakan. Pada beberapa poin, penulis akan menyertakan rancangan dan pemilihan teknologi yang berdampak dari hasil analisa tersebut.

III.1.1 Requirement Analysis

III.1.1.1 Fungsionalitas Dasar

Aplikasi Lelang Online berbasis Web yang diberi nama **Lelangapa** adalah sebuah *online auction web* yang dibangun berdasarkan paper rujukan utama, dengan fitur-fitur utama sebagai berikut:

1. **Registrasi ke dalam sistem**
2. **Login ke dalam sistem**
3. **Mendaftarkan barang untuk dilelang**
Pengguna dapat mendaftarkan barangnya untuk dijual dan dilelang. Selain itu, pengguna dapat menentukan harga awal dan batas waktu lelang pada saat mendaftarkan barangnya.
4. **Memperbarui barang yang dilelang**
Pengguna dapat memperbarui informasi mengenai barang yang dilelang, seperti nama barang, menambah foto deskripsi barang, atau menambah waktu lelang.
5. **Melihat informasi barang yang dilelang**
Pengguna dapat melihat detail informasi barang yang sedang dilelang – seperti foto barang, riwayat penawaran harga barang lelang, sisa waktu penawaran, deskripsi barang, dsb.

6. **Melihat informasi riwayat lelang** (Siapa saja yang sudah mengajukan penawaran dan harga yang ditawarkan)

7. **Mengajukan penawaran harga untuk barang yang dilelang / Menjadi auctioneer**

Selain menjadi auctioneer, pengguna juga dapat menawarkan harga terhadap barang-barang yang didaftarkan oleh pengguna lain.

8. **Mendapatkan pemberitahuan jika penawaran harga dikalahkan dengan harga lebih tinggi**

Pengguna mendapatkan pemberitahuan jika pengguna sedang mengikuti pelelangan barang, dan ada penawaran harga yang lebih tinggi dari penawaran oleh pengguna tersebut, sehingga pengguna dapat mengikuti perkembangan harga dari barang yang dilelang.

9. **Mengikuti / follow barang yang sedang dilelang dan mendapatkan pemberitahuan jika barang tersebut**

Jika pengguna sedang tidak ingin melelang barang namun ingin tetap mengetahui informasi dari suatu barang lelang, pengguna dapat mengikuti feed/berita dari barang tersebut.

10. **Mendapatkan pemberitahuan jika memenangkan lelang atau tidak.**

Jika pengguna mengajukan penawaran harga terhadap suatu barang, maka pengguna akan mendapatkan pemberitahuan pada saat batas waktu lelang selesai, apakah pengguna tersebut memenangkan proses lelang tersebut atau tidak.

11. **Saling berkirim pesan singkat/chat kepada auctioneer/penawar harga**

Untuk saling bertukar informasi mengenai barang yang sedang dilelang, auctioneer dan penawar harga dapat saling berkirim pesan singkat.

12. **Melihat riwayat penawaran harga lelang**

Pengguna dapat melihat barang riwayat penawaran harga

yang diberikan oleh pengguna tersebut terhadap semua barang yang pernah dia lelang.

13. **Melihat riwayat barang yang dilelang**

Pengguna dapat melihat riwayat barang yang pernah ditawarkan harganya/diberikan penawaran harga.

14. **Memberi review tentang pengguna lain sebagai auctioneer dan atau sebagai penawar harga**

Pengguna dapat memberikan komentar/testimoni berdasarkan pengalaman bertransaksi/penawaran harga dengan pengguna lainnya, baik pengalaman memuaskan ataupun pengalaman buruk.

15. **Melihat review mengenai seorang pengguna**

Selain memberikan review, pengguna dapat melihat review seorang pengguna.

16. **Memblok pengguna sebagai auctioneer**

Auctioneer dapat memblok pengguna agar pengguna tersebut tidak memberikan penawaran harga terhadap barang yang sedang ia lelang. Hal ini bisa saja karena review/testimoni pengguna tersebut buruk atau karena alasan lainnya.

17. **Mencari barang yang dilelang dengan keyword tertentu**

III.1.1.2 Analisa Paper Rujukan

Dengan perkembangan teknologi, perlahan kebiasaan manusia berubah. Termasuk juga dalam perdagangan, dimana transaksi jual beli barang tidak lagi harus melalui tatap muka. Penjualan online saat ini sudah dapat dilakukan lewat berbagai cara, antara lain menggunakan e-commerce, atau posting di social media, atau bisa juga 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.

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 ^a (2.5) ^b	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)

^a Mean.

^b Standard deviation.

Gambar III.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 ^a (32.3) ^b	228 ^a (81.4) ^b	52 ^a (18.6) ^b	7.2 ^c (2.4) ^d	6.6 ^c (2.7) ^d
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)

^a Number.

^b %.

^c Mean.

^d Standard deviation.

Gambar III.2: Kategori Perbaikan terhadap Kegagalan Pelayanan Lelang Online

Maka berdasarkan hasil analisa tersebut, fitur-fitur yang perlu ditambahkan selain daripada fitur dasar aplikasi lelang online 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*.

III.1.1.3 *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.

Dalam sebuah bisnis, *software project* secara murni dilihat sebagai sebuah produk

III.1.1.4 **Spesifikasi Kebutuhan Fungsional**

Berdasarkan deskripsi umum aplikasi, kebutuhan fungsional dari aplikasi dijabarkan di III.1.

Tabel III.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
Dilanjutkan ke halaman selanjutnya		

Tabel III.1 – lanjutan dari halaman sebelumnya

No.	Parameter	Ketersediaan
4	Memajemen Laporan Pengguna	Pengelola: Melihat daftar laporan dari pengguna, memblokir pengguna yang melanggar ketentuan
5	Memajemen Kupon	Pengelola: Membuat kupon, melihat daftar kupon, melihat riwayat penggunaan kupon

III.1.1.5 Spesifikasi Kebutuhan Non-Fungsional

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

Tabel III.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 dilengkapi dengan akses
.. dilanjutkan ke halaman selanjutnya		

Tabel III.2 – lanjutan dari halaman sebelumnya

No.	Parameter	Ketersediaan
4	Portabilitas	Aplikasi dapat diakses pada semua platform selama platform tersebut terinstall web browser

III.1.1.6 *Bussiness Modelling Workflow*

III.1.1.7 Kasus Penggunaan

Kasus penggunaan disini dimaksudkan untuk menurunkan kebutuhan fungsional yang telah dispesifikasikan sebelumnya pada tabel III.1 sebelumnya.

Daftar kasus Penggunaan dapat dilihat pada III.3.

Tabel III.3: Tabel Kasus Penggunaan

ID Kasus Penggunaan	Kasus Penggunaan
KP-01	Manajemen Autentikasi Pengguna
KP-02	Memajemen Transaksi Lelang
KP-03	Manajemen Barang Lelang
KP-04	Manajemen Interaksi & Kepuasan Pengguna
KP-05	Monitoring Proses Lelang
KP-06	Manajemen Voucher

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

III.1.1.8 KP01. Manajemen Autentikasi Pengguna

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

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

Tabel III.4: Spesifikasi Kasus Penggunaan Registrasi

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

Tabel III.5: Spesifikasi Kasus Penggunaan Login

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

Tabel III.6: Spesifikasi Kasus Penggunaan: Konfirmasi Email

III.1.1.9 KP02. Manajemen Transaksi Lelang

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

Kode	UC-02.01
Nama	Melihat daftar barang yang dilelang
Aktor	Pengguna
Deskripsi	Pengguna melihat daftar barang yang sedang dilelang
Tipe	Fungsional
Pre Condition	Sistem belum menampilkan daftar barang yang sedang dilelang
Post Condition	Sistem menampilkan daftar barang yang sedang dilelang
Alur Kejadian Normal	
	<ol style="list-style-type: none"> 1. Pengguna mengklik <i>icon</i> aplikasi di kiri atas halaman 2. Sistem menampilkan halaman depan yang berisi daftar barang yang sedang dilelang <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>
Alur Kejadian Alternatif	
	-

Tabel III.7: Spesifikasi Kasus Penggunaan : Melihat barang yang dilelang

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

Tabel III.8: Spesifikasi Kasus Penggunaan : Mencari Barang Lelang

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

Tabel III.9: Spesifikasi Kasus Penggunaan : Mencari Barang Lelang

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

Tabel III.10: Spesifikasi Kasus Penggunaan : Melihat Riwayat Penawaran Lelang Barang

III.1.1.10 KP03. 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.

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

Tabel III.11: Spesifikasi Kasus Penggunaan : Mendaftarkan Barang Lelang

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

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

Tabel III.13: Spesifikasi Kasus Penggunaan : Melihat Barang yang Pernah Didaftarkan

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

Tabel III.14: Spesifikasi Kasus Penggunaan : Melihat Riwayat Penawaran Harga

III.1.1.11 KP04. Manajemen Interaksi & Kepuasan Pengguna

Pada kasus penggunaan ini, pengguna difasilitasi untuk berinteraksi, memberikan *review* / testimoni terhadap pengguna

lainnya sesuai dengan keinginan.

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

Tabel III.15: Spesifikasi Kasus Penggunaan : Melihat Review Pengguna

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

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

Alur Kejadian Normal

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

Alur Kejadian Alternatif

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

Kode	UC-04.04
Nama	Mengirim Pesan
Aktor	Pengguna
Deskripsi	Pengguna akan mengirimkan pesan kepada pengguna lainnya
Tipe	Fungsional
<i>Precondition</i>	Pesan yang dikirimkan pengguna belum tersimpan pada sistem
<i>Postcondition</i>	Pesan yang dikirimkan pengguna berhasil tersimpan pada sistem

Alur Kejadian Normal

	<ol style="list-style-type: none"> 1. Pengguna mengklik <i>URL</i> pengguna tujuan yang ingin dikirim pesan 2. Sistem menampilkan halaman profil pengguna tujuan 3. Pengguna mengklik tombol "Kirim Pesan" 4. Sistem menampilkan halaman percakapan pengguna terhadap tujuan beserta riwayat percakapan pengguna dengan pengguna tujuan 5. Pengguna memasukkan pesan yang ingin dikirimkan pada <i>field input</i> yang disediakan 6. Setelah selesai, pengguna mengklik tombol 'Kirim' 7. Sistem mengirim kepada koneksi socket 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
--	---

Alur Kejadian Alternatif

	Terjadi masalah teknis sehingga pesan tidak dapat terkirim
	<ol style="list-style-type: none"> 7a. Sistem mendapatkan <i>exception</i> dari koneksi socket, bahwa pesan tidak dapat tersimpan

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

Alur Kejadian Normal

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

Alur Kejadian Alternatif

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

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

Tabel III.20: Spesifikasi Kasus Penggunaan : Mendaftarkan Barang Lelang

III.1.1.12 KP05. *Monitoring* Proses Lelang

Kasus penggunaan ini seluruhnya digunakan oleh *administrator* aplikasi.

III.1.1.13 KP05. Manajemen Voucher

Kasus penggunaan ini seluruhnya digunakan oleh *administrator* aplikasi, ditujukan untuk mempermudah *administrator* dalam memanajemen voucher/kupon yang dibagikan oleh pengguna.

III.1.2 *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 sisi *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* (?, ?, 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 III.3: *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*.

III.1.3 Analisa Pribadi Penulis

III.1.3.1 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*

III.1.3.2 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, *disadvantages*nya adalah ketidakpraktisan untuk selalu *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

III.1.3.3 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 III.2.4, 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 akses *eksternal services*. Tujuannya, agar jika kedepannya terdapat perbaikan fitur/penambahan fitur, lebih mudah melakukan *traceback* terhadap file/kelas yang bertanggungjawab terhadap fitur tersebut.

III.1.3.4 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 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.

III.1.3.5 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.

III.1.3.6 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* tidak cocok dimasukkan kedalam 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 - pada rancang bangun aplikasi ini, DBMS NoSQL, dengan alasan-alasan sebagai berikut.

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

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 (?, ?)

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*.

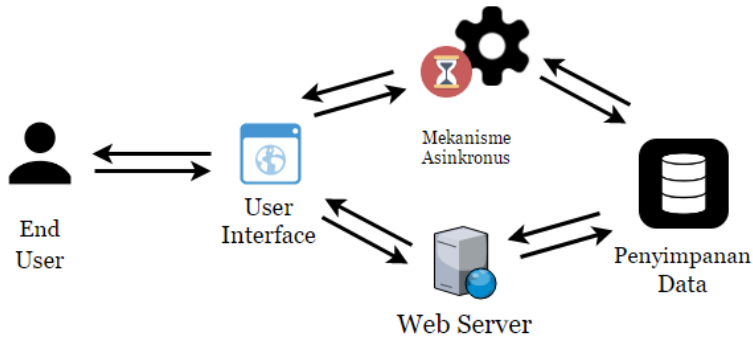
III.1.4 Perancangan Sistem

III.2 Perancangan Sistem

III.2.1 Identifikasi Komponen Fundamental

Berdasarkan Bab Analisa, dapat diidentifikasi dan divisualisasikan (pada gambar III.4) 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 III.4: Arsitektur dasar yang dibutuhkan untuk membangun aplikasi

III.2.2 *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 III.1.2.

III.2.2.1 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

III.2.2.2 PostgreSQL untuk Penyimpanan Data

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

III.2.2.3 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 fiturnya lengkap untuk *data mining*

III.2.2.4 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)

III.2.2.5 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
 3. Pembelajaran bagi penulis
- Opsi lainnya
 1. Disimpan dalam server : Mengurangi performa server karena sifatnya yang memakan *resource* cukup banyak, dan menambah *cost* untuk *upgrade server storage*

III.2.2.6 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

III.2.2.7 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)
 4. Pembelajaran bagi penulis
- 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

III.2.2.8 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)
 4. Pembelajaran bagi penulis
- 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

III.2.2.9 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
 3. Pembelajaran bagi penulis
- 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

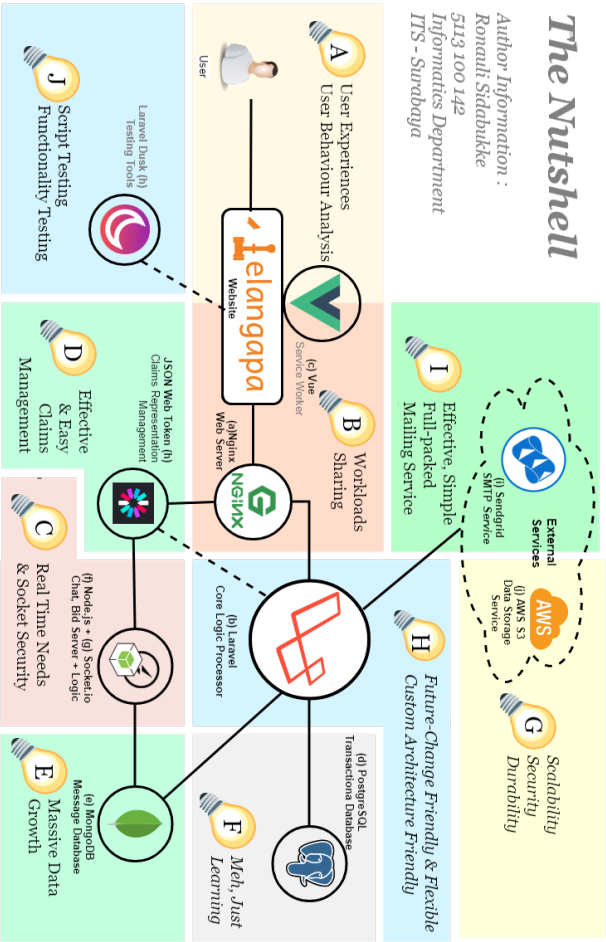
III.2.2.10 LARAVEL DUSK untuk *Functionality Testing Script*

- Kelebihan
 1. *Learning gap* yang kecil karena didesain sefamiliar mungkin dengan Laravel
 2. Pembelajaran untuk penulis
- Opsi lainnya
 1. Selenium : *Learning curve* yang besar
 2. Phantom.js : *Learning curve* yang besar

III.2.3 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 III.1.2 dan subbab III.2.2 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.



Gambar III.5: Visualisasi arsitektur dan teknologi Final yang diterapkan dalam rancang bangun aplikasi

III.2.4 Struktur Penyusun Aplikasi

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 III.1.2.

III.2.5 Perancangan *Database* Transaksional**III.2.5.1 *Conceptual Database Design*****III.2.5.2 *Physical Database Design*****III.2.5.3 *Optimization: Cache Tables*****III.2.6 Perancangan *Database* Non-Transaksional****III.2.6.1 *Social Inboxes Design & Principles*****III.2.6.2 *Physical Database Design*****III.2.7 Kamus Data *Database* Non-Transaksional****III.2.7.1 *Collection* userchats****III.2.7.2 *Collection* joinedrooms****III.2.7.3 *Collection* item_images****III.2.8 Kamus Data *Database* Non-Transaksional****III.2.8.1 *Collection* users****III.2.8.2 *Collection* bids****III.2.8.3 *Collection* categories****III.2.8.4 *Collection* auctioneer_rating****III.2.8.5 *Collection* item_cached****III.2.8.6 *Collection* bid_logs****III.2.8.7 *Collection* bidder_rating**

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*

IV.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

IV.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.

IV.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.

IV.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.

IV.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 di*upgrade*, tidak ada jaminan bahwa *deprecated dependencies* (keadaan dimana sebuah *package* tidak di*support* 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 *dcompile* 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>
```

IV.2.1.3 Instalasi JWT pada Laravel dan Node.js

IV.2.1.4 SENDGRID WHITELISTING

IV.2.2 Diagram Kelas

IV.3 Implementasi Antarmuka / *User Interface*

IV.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 IV.1: Pseudocode Controller untuk Menampilkan Antarmuka A

IV.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 IV.2: Pseudocode Controller untuk Menampilkan Antarmuka B

BAB V

PENGUJIAN DAN EVALUASI

V.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 .

V.1.1 Pengujian Fungsionalitas Lelang

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

V.1.2 Pengujian Fungsionalitas Voucher

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

V.1.3 *Summary* Pengujian

V.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.

V.2.1 Evaluasi Aspek Kebutuhan

V.2.2 Evaluasi Aspek Teknis

V.2.3 Evaluasi Aspek *Bussiness Engineering*

V.2.3.1 Evaluasi Regulasi Terkait

V.2.3.2 *Market Analysis*

V.2.3.3 *Competitors Competitiveness*

V.2.3.4 Evaluasi *Startup Burn Rate*

V.2.4 Evaluasi Aspek Analisa dan Desain

V.2.5 Evaluasi Aspek Implementasi

V.2.6 Evaluasi Aspek Pengujian

V.2.7 *Summary* Evaluasi

V.2.8 *Further Enchancements*

V.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.

VI.1 Kesimpulan

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

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

VI.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 (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.bocahsoenyi.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.nostratech.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.
- 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
- 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/smtp.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* [Business]. 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
- 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/>