

## **LAPORAN KKI**

### **LELANG ONLINE SECARA REALTIME DENGAN PROTOKOL WEBSOCKET MENGGUNAKAN SOCKET.IO**

Diajukan untuk memenuhi salah satu syarat mengerjakan dan menempuh ujian  
**KKI**



Disusun Oleh :

Nama : Diky Arga Anggara

NIM : A11.2013.07386

Program Studi : Teknik Informatika-S1

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS DIAN NUSWANTORO  
SEMARANG  
2017**

## **PERSETUJUAN SKRIPSI**

Nama : Diky Arga Anggara  
NIM : A11.2013.07386  
Program Studi : Teknik Informatika  
Fakultas : Ilmu Komputer  
Judul Tugas Akhir : Lelang Online Secara Realtime dengan Protokol  
WebSocket Menggunakan Socket.IO

Tugas Akhir ini telah diperiksa dan disetujui,

Semarang, 11 Agustus 2017

Menyetujui :

Pembimbing

Mengetahui :

Dekan Fakultas Ilmu Komputer

Ajib Susanto M.Kom

NPP. 0686.11.2000.253

Dr. Drs. Abdul Syukur, MM

NPP.0686.11.1992.017

## **PENGESAHAN DEWAN PENGUJI**

Nama : Diky Arga Anggara  
NIM : A11.2013.07386  
Program Studi : Teknik Informatika-S1  
Fakultas : Ilmu Komputer  
Judul Tugas Akhir : LELANG ONLINE SECARA REALTIME DENGAN  
PROTOKOL WEBSOCKET MENGGUNAKAN  
SOCKET.IO

Tugas Akhir ini telah diujikan dan dipertahankan di hadapan Dewan Penguji pada Sidang tugas akhir pada 11 Agustus 2017. Menurut pandangan kami, tugas akhir ini memadai dari segi kualitas maupun kuantitas untuk tujuan penganugerahan gelar Sarjana Komputer (S.Kom).

Semarang, 11 Agustus 2017

Dewan Penguji :

Penguji 1

Penguji 2

Egia Rosi Subhiyakto M.Kom  
NPP. 0686.11.2014.585

Umi Rosyidah S.Kom, M.T  
NPP. 0686.11.2009.372

Ketua Penguji

Danang Wahyu Utomo M.Kom  
NPP. 0686.11.2014.583

## **PERNYATAAN KEASLIAN SKRIPSI**

Sebagai mahasiswa Universitas Dian Nuswantoro, yang bertanda tangan di bawah ini, saya :

Nama : Diky Arga Anggara

NIM : A11.2013. 07386

Menyatakan bahwa karya ilmiah saya yang berjudul :

### **LELANG ONLINE SECARA REALTIME DENGAN PROTOKOL WEBSOCKET MENGGUNAKAN SOCKET.IO**

merupakan karya asli saya (kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya dan perangkat pendukung. Apabila di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, dan disertai dengan bukti-bukti yang cukup, maka saya bersedia untuk dibatalkan gelar saya beserta hak dan kewajiban yang melekat pada gelar tersebut. Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Semarang

Pada tanggal : 11 Agustus 2017

Yang menyatakan

(Diky Arga Anggara)

## **PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS**

Sebagai mahasiswa Universitas Dian Nuswantoro, yang bertanda tangan di bawah ini, saya :

Nama : Diky Arga Anggara

NIM : A11.2013.07386

Demi mengembangkan Ilmu Pengetahuan, menyetujui untuk memberikan kepada Universitas Dian Nuswantoro Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

### **LELANG ONLINE SECARA REALTIME DENGAN PROTOKOL WEBSOCKET MENGGUNAKAN SOCKET.IO**

Beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini Universitas Dian Nuswantoro berhak untuk menyimpan, meng-copy, ulang (memperbanyak), menggunakan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya dan menampilkan/ mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/ pencipta.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Dian Nuswantoro, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Semarang

Pada tanggal : 11 Agustus 2017

Yang menyatakan

(Diky Arga Anggara)

## **UCAPAN TERIMAKASIH**

Alhamdulillah, puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya. Sholawat dan salam kepada Rasulullah Muhammad SAW, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan judul “LELANG ONLINE SECARA REALTIME DENGAN PROTOKOL WEBSOCKET MENGGUNAKAN SOCKET.IO” dengan baik tanpa suatu halangan yang berarti. Tugas Akhir ini disusun untuk memenuhi syarat kelulusan akademik.

Penulis menyadari bahwa tanpa bimbingan, bantuan, dan doa dari berbagai pihak, Tugas Akhir ini tidak akan dapat diselesaikan tepat pada waktunya. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam proses penyusunan Tugas Akhir ini, yaitu kepada :

1. Allah SWT atas kehendak-Nya penulis dapat melaksanakan dan menyelesaikan Tugas Akhir ini.
2. Prof. Dr. Ir. Edi Noersasongko, M.Kom. selaku Rektor Universitas Dian Nuswantoro.
3. Dr. Drs. Abdul Syukur, MM selaku Dekan Fakultas Ilmu Komputer Universitas Dian Nuswantoro.
4. Heru Agus Santoso, Ph.D. selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Dian Nuswantoro.
5. Ajib Susanto M.Kom selaku dosen pembimbing dan akademik yang dengan sabar selalu memberikan arahan, bimbingan dan motivasi dalam penyusunan Tugas Akhir ini.
6. Dosen-dosen pengampu di fakultas Ilmu Komputer Teknik Informatika Universitas Dian Nuswantoro Semarang yang telah memberikan ilmu dan pengalamannya masing-masing,

- sehingga penulis dapat mengimplementasikan ilmu yang telah disampaikan.
7. Kedua Orang Tua yang selalu saya cintai dan banggakan.
  8. Semua pihak yang namanya tidak dapat disebutkan satu persatu.

Akhir kata, penulis ingin menyampaikan bahwa penyusunan Tugas Akhir ini masih sangat jauh dari kesempurnaan. Oleh karena itu, penulis sangat mengharapkan berbagai masukan dari semua pihak, baik berupa saran maupun kritik yang sekiranya bisa memperbaiki kekurangan-kekurangan yang ada dalam Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak. Amin

Semarang, 11 Agustus 2017

Penulis

## RINGKASAN

Lelang telah di akui secara luas sebagai cara yang tepat untuk mengetahui harga pasaran suatu produk. Seiring perkembang zaman, lelang juga di lakukan secara *online*, guna menekan efisiensi waktu dan biaya. Lelang *online* menggunakan metode yang sedikit berbeda dengan lelang konvensional pada umumnya, metode yang kerap dipakai lelang *online* adalah *English auction*, perbedaan utamanya terletak pada penentuan pemenang, yaitu pada waktu berakhirnya lelang, lelang akan selesai tepat ketika batas waktu lelang yang ditentukan berakhir, kemudian secara otomatis peserta lelang dengan tawaran tertinggi yang menang. Karena waktu menjadi penentu utama pada lelang *online*, maka kecepatan bertukar data menjadi sangat penting. Oleh karena dibutuhkan teknologi yang mampu menangani data *realtime* untuk banyak pengguna secara bersamaan dengan efisien. Pada penelitian kali ini penulis mencoba menggunakan standar protokol Websocket untuk memastikan data penawaran lelang diterima semua peserta lelang dengan *realtime*.

Kata Kunci : Lelang Online, Real Time, WebSocket, Test Driven Development

## DAFTAR ISI

PERSETUJUAN SKRIPSI.....	ii
PENGESAHAN DEWAN PENGUJI .....	iii
PERNYATAAN KEASLIAN SKRIPSI .....	iv
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS.....	v
UCAPAN TERIMAKASIH .....	vi
RINGKASAN .....	viii
DAFTAR ISI .....	ix
DAFTAR TABEL .....	xi
DAFTAR GAMBAR .....	xii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Manfaat Kuliah Kerja Industri .....	3
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI .....	6
2.1 Tinjauan Objek Penelitian.....	6
2.2 State of The Art.....	9
2.3 Landasan Teori.....	15
2.4 Deskripsi Tempat KKI .....	24
2.5 Logo dan Makna Tempat KKI .....	25
2.6 Struktur dan Job Deskripsi .....	25
2.7 Visi dan Misi Tempat KKI.....	27
BAB III METODE PENELITIAN .....	31
3.1 Instrumen Penelitian.....	31

3.2	Prosedur Pengambilan Data .....	31
3.3	Teknik Analisis Data .....	29
3.4	Metode.....	30
3.5	Pengujian.....	34
	BAB IV RANCANGAN SISTEM DAN IMPLEMENTASI .....	36
4.1	Perancangan Kebutuhan Sistem .....	36
4.2	Analisis Perancangan Sistem .....	37
4.3	Perancangan Antar Muka Android.....	47
4.4	Perancangan Web Service.....	54
4.5	Implementasi Android.....	55
4.6	Implementasi Web Service.....	58
4.7	Pengujian Aplikasi .....	59
	BAB V KESIMPULAN DAN SARAN .....	61
5.1	Kesimpulan.....	61
5.2	Saran.....	61
	DAFTAR PUSTAKA.....	62

## **DAFTAR TABEL**

Tabel 2. 2 : <i>State of The Art</i> .....	9
Tabel 4. 1 : Tabel Pengujian Kecepatan Bertukar Data Melalui WebSocket .....	56

## DAFTAR GAMBAR

Gambar 2. 1: Data Perbandingan Websocket dengan HTTP Long/Polling.....	6
Gambar 2. 2 : Grafik Perbandingan WebSocket dengan HTTP Poling.....	8
Gambar 2. 3: Alur Pemrosesan pada <i>Websocket</i> .....	18
Gambar 2. 4: Contoh penulisan data menggunakan format JSON .....	20
Gambar 2. 5: Contoh sintaks dan penggunaan Mocha .....	23
Gambar 2. 6: Contoh penggunaan tiga format penulisan tes dengan Chai .....	24
Gambar 2. 7: Logo PT Hactivate Teknologi Indonesia .....	25
Gambar 3. 1: Tangkapan layar wawancara 1 .....	29
Gambar 3. 2: Tangkapan layar wawancara 2 .....	29
Gambar 3. 3: Tangkapan layar wawancara 3 .....	29
Gambar 3. 4: Alur Pengembangan Perangkat Lunak dengan TDD .....	30
Gambar 4. 1: Skema Layanan AWS yang di Gunakan oleh Aplikasi .....	36
Gambar 4. 2: Diagram Use Case .....	38
Gambar 4. 3 Diagram Aktivitas Register .....	41
Gambar 4. 4 Diagram Aktivitas Login .....	42
Gambar 4. 5: Diagram Aktivitas Buat Lelang Baru .....	43
Gambar 4. 6: Diagram Aktivitas Mengikuti Lelang .....	44
Gambar 4. 7: Diagram Aktivitas Logout Aplikasi .....	45
Gambar 4. 8: Diagram Aktivitas Checkout .....	46
Gambar 4. 9: Desain Halaman Register .....	47
Gambar 4. 10: Desain Halaman Login .....	48
Gambar 4. 11: Desain Halaman Beranda Aplikasi .....	49
Gambar 4. 12: Desain Pop up Setelah Klik Buat Lelang .....	49
Gambar 4. 13: Desain Halaman Buat Lelang dari Produk Baru .....	50
Gambar 4. 14: Desain Halaman Buat Lelang dari Lapak .....	51
Gambar 4. 15: Desain Halaman Detail Lelang .....	51
Gambar 4. 16: Desain Riwayat Tawaran Lelang .....	52
Gambar 4. 17: Desain Halaman Checkout Barang .....	53

Gambar 4. 18: Skema Database.....	54
Gambar 4. 19: Antarmuka Implematsi Halaman Register .....	55
Gambar 4. 20: Antarmuka Halaman Detail Lelang .....	55
Gambar 4. 21: Antarmuka Halaman Login .....	56
Gambar 4. 22: Antarmuka Daftar Riwayat Lelang.....	57
Gambar 4. 23: Antarmuka Menu Drawer (Utama).....	57
Gambar 4. 24: Antarmuka Halaman Buat Lelang .....	57
Gambar 4. 25: Antarmuka Menu Opsi Penawaran .....	58
Gambar 4. 26: Dokumentasi API WebService .....	58
Gambar 4. 27: Contoh Unit Test Case Membuat Lelang Baru .....	59

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Pada era digital seperti yang telah terjadi saat ini, segala macam aktivitas manusia telah sangat bergantung dengan kemajuan teknologi. Digitalisasi telah merambah ke segala sektor kehidupan manusia modern ini, termasuk aktivitas dalam melakukan lelang barang.

Pengeritian lelang menurut pasal 1 Vendu Reglement yang hingga saat ini masih berlaku. Menurut pasal tersebut lelang adalah setiap penjualan barang di muka umum dengan cara penawaran harga naik – naik, turun – turun, dan atau tertulis melalui usaha mengumpulkan para peminat atau peserta lelang yang dipimpin oleh pejabat lelang atau Vendemeeste [1].

Lelang secara *online* di Indonesia banyak terjadi pada group atau forum di sosial media, seperti Facebook dan Instargram. Karena sosial media bukan tempat yang sesuai untuk melakukan lelang, kerap terjadi beberapa masalah dalam pelaksannya.

Beberapa masalah dalam lelang *online* di sosial media diantaranya, kasus *bid and run*, saling tidak percaya antara pelelang dan peserta lelang dan rawan terhadap kasus penipuan, karena tidak ada penjembatan diantara kedua belah pihak.

Beberapa situs *online* telah berusaha menyediakan layanan lelang secara *online* ini, namun karena situs-situs tersebut masih baru, membuat peserta lelang tidak percaya untuk mendeposit uangnya sebelum mengikuti lelang. Karena hal tersebut, dibutuhkan kerjasama dengan pihak ketiga yang telah dikenal oleh masyarakat luas, untuk meningkatkan kepercayaan pengguna. Pada penelitian ini penulis berkejasama dengan PT. Bukalapak.com untuk menjadi penjembatan perihal transaksi dan barang.

Lelang secara *online* memiliki sedikit perbedaan dengan lelang yang dilakukan secara konvensional, salah satunya terletak pada penentuan waktu berakhirnya lelang. Lelang *online* akan berakhir jika waktu yang telah ditentukan terpenuhi dan cara menentukan pemenang lelang adalah peserta lelang dengan nominal penawaran tertinggi. Oleh karena itu, peserta lelang *online* biasanya beramai-ramai melakukan penawaran di menit-menit akhir secara bersamaan atau biasa disebut dengan *snipping*, tentu saja hal ini membutuhkan transaksi data antar pengguna secara *realtime*.

Ketika salah satu pengguna melakukan penawaran, semua peserta lelang pun akan mendapatkan notifikasi dan perubahan nominal harga barang saat itu juga. Maka di butuhkan teknologi untuk menangani pertukaran data secara *realtime*, pada penelitian ini penulis mencoba menggunakan protokol Websocket untuk mengatasi masalah pertukaran data tersebut.

Guna mempercepat dan memudahkan implementasi penelitian ini, penulis menggunakan *library* WebSocket bernama Socket.IO yang dipakai disisi *server (backend)* terintegrasi pada Node.js dan sisi *client (front-end)* React Native yang di *compile* menjadi aplikasi Android.

## 1.2 Rumusan Masalah

Dari latar belakang yang telah tertulis, rumusan masalah dalam penelitian ini adalah bagaimana membuat aplikasi lelang *online* secara *realtime* dengan protokol WebSocket.

## 1.3 Batasan Masalah

Dalam penelitian ini terdapat batasan-batasan masalah agar hasil yang didapat tidak menyimpang dari tujuan seperti yang tertulis dalam rumusan masalah dan latar belakang :

1. Perancangan aplikasi lelang *online* ini menggunakan *Socket.IO*.
2. Bahasa pemrograman yang digunakan adalah JavaScript.

## 1.4 Tujuan Penelitian

Membangun aplikasi lelang *online* secara *realtime* dengan protokol websocket dari sisi *server* dan *client* menggunakan *library* JavaScript bernama Socket.IO

## 1.5 Manfaat Penelitian

Penelitian yang dilakukan ini diharapkan dapat bermanfaat bagi pihak terkait, diantaranya adalah :

### 1.5.1 Manfaat Praktis

- a. Menghasilkan sebuah aplikasi yang bisa menjadi penjembatan pelelang dan peserta lelang yang mudah dan cepat.
- b. Menambah wawasan terkait penerapan protokol Websocket yang telah disinggung pada materi perkuliahan pemrograman *mobile*.

### 1.5.2 Bagi Akademik

- a. Digunakan untuk tolak ukur sejauh mana mahasiswa menerima ilmu yang diberikan oleh dosen untuk di implementasikan.
- b. Sebagai refrensi untuk pengembangan pada kasus yang sama oleh mahasiswa yang membutuhkan.

## 1.6 Manfaat Kuliah Kerja Industri

1.6.1 Manfaat yang diperoleh dari Kuliah Kerja Industri bagi Universitas Dian Nuswantoro antara lain :

1. Universitas Dian Nuswantoro dapat menjalin kerjasama bilateral yang saling menguntungkan dengan PT. Hacktivate Teknologi Industri selaku industri yang bergerak di dunia informatika yang berbasis di Jakarta.
2. Menambah kualitas dari lulusan yang dihasilkan karena telah menempuh proses pembelajaran dan pembuatan aplikasi yang digunakan pada dunia nyata selama beberapa waktu di dunia

industri informatika pada PT. Hacktivate Teknologi Indonesia.

3. Universitas Dian Nuswantoro akan lebih dikenal dalam dunia industri informatika, dikarenakan PT. Hacktivate Teknologi Indonesia memiliki kerjasama dengan perusahaan-perusahaan yang bergerak pada bidang yang sama, serta dapat menambah kepercayaan dunia industri dengan mengirimkan mahasiswa yang berkompeten ke PT. Hacktivate Teknologi Indonesia.

1.6.2 Manfaat yang diperoleh bagi PT. Hacktivate Teknologi Indonesia antara lain :

1. Membantu PT. Hacktivate Teknologi Indonesia untuk menyaring dan menemukan sumber daya pengembang aplikasi baru atau *programmer* baru.
2. PT. Hacktivate Teknologi Indonesia akan lebih dikenal dilingkungan universitas melalui mahasiswa yang ditempatkan di perusahaan tersebut.
3. Adanya kritik dan saran yang membangun dari mahasiswa kepada PT. Hacktivate Teknologi Indonesia karena sebelumnya mahasiswa tidak berasal dari perusahaan tersebut dan memiliki potensi untuk mengembangkan apa yang menjadi potensi di perusahaan tersebut.

1.6.3 Manfaat bagi mahasiswa yang melaksanakan KKI pada PT. Hacktivate Teknologi Indonesia antara lain :

1. Mahasiswa mempelajari bahasa pemrograman JavaScript secara intensif selama 12 minggu.
2. Mahasiswa mendapat pengalaman membangun aplikasi dari semua sisi pengembangan atau *full-stack*, meliputi *back-end web service*, *front-end web* dan *mobile app* dengan bahasa pemrograman utama JavaScript.
3. Selama proses pembelajaran dan penggerjaan, mahasiswa dibimbing dan dipantau kemajuannya oleh setidaknya satu instruktur.

4. Mahasiswa mendapatkan kesempatan mempraktikan Test Driven Development saat membuat aplikasi, sehingga dapat meminimalisasi *bugs* pada hasil aplikasi yang dibuat.
5. Mahasiswa lebih dituntut berorientasi pada praktik daripada teori, sehingga diberikan lebih banyak waktu praktik dengan langsung dihadapkan pada problematika atau studi kasus untuk diselesaikan.
6. Mahasiswa dibebaskan menggunakan *library* atau *framework* yang akan digunakan untuk membuat proyek dengan berlandaskan analisa yang telah dilakukan sebelumnya.
7. Mahasiswa berkesempatan berkerja pada PT. Hactivate Teknologi Indonesia atau perusahaan-perusahaan yang sudah berkerja-sama.
8. Membangun karakter dari mahasiswa karena dalam dunia industri berlaku beberapa aturan yang harus dipatuhi, dari hal tersebut kedisiplinan dan tanggung jawab dapat dilatih dan menjadi kebiasaan.
9. Manambah wawasan mahasiswa tentang manajemen proyek dengan metode SCRUM dan Kanban pada industri informatika dewasa ini.

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Objek Penelitian

Victoria Pimentel, Bradford G. Nickerson (2012), komunikasi atau bertukar informasi yang disokong internet dewasa ini membuatnya sangat mudah, namun akan menjadi problema ketika pertukaran data yang dibutuhkan bersifat *real-time*, pada penelitian ini akan menentukan apakah WebSocket lebih cepat dari pada HTTP Polling. Didapatkan hasil sebagai berikut [2].

Table 1. Evaluation results. <sup>b</sup>				
Start	Method	Average latency ( $\mu$ )	Standard deviation ( $\sigma$ )	Ratio ( $r$ )
<b>Edmonton, Canada</b>				
9:04 a.m.	WS <sup>c</sup>	40.3	7.6	1
9:10 a.m.	LP	41.3	0.86	1.02
9:21 a.m.	P	151.3	63.7	3.75
1:04 p.m.	WS	39.7	2.8	1
1:10 p.m.	LP	40.7	2.4	1.02
1:16 p.m.	P	149.9	63.6	3.77
7:01 p.m.	WS	40.5	0.52	1
7:07 p.m.	LP	41.5	0.97	1.02
7:13 p.m.	P	150.4	63.4	3.72
<b>Caracas, Venezuela</b>				
10:25 a.m.	WS	122.9	48.2	1
10:32 a.m.	LP	131.5	66.3	1.07
10:45 a.m.	P	283.5	111.7	2.31
12:00 noon	WS	124.0	51.02	1
12:05 p.m.	LP	121.4	50.8	0.98
12:11 p.m.	P	298.0	114.1	2.40
7:00 p.m.	WS	98.8	1.37	1
7:08 p.m.	LP	106.1	41.2	1.07
7:16 p.m.	P	266.9	100.1	2.70
<b>Lund, Sweden</b>				
10:45 a.m.	WS	91.92	2.74	1
10:53 a.m.	LP	87.5	2.26	0.95
11:00 a.m.	P	241.4	88.4	2.63
3:43 p.m.	WS	95.3	2.08	1
3:51 p.m.	LP	97.3	1.72	1.02
3:58 p.m.	P	253.0	90.1	2.65
1:11 a.m.	WS	75.1	0.42	1
1:17 a.m.	LP	77.6	0.88	1.03
1:23 a.m.	P	225.1	84.8	3.00
<b>Nagaoka, Japan</b>				
11:16 a.m.	WS	153.4	133.3	1
11:22 a.m.	LP	647.0	683.4	4.22
11:28 a.m.	P	584.3	508.8	3.81
12:32 p.m.	WS	163.3	161.9	1
12:45 p.m.	LP	585.7	622.0	3.59

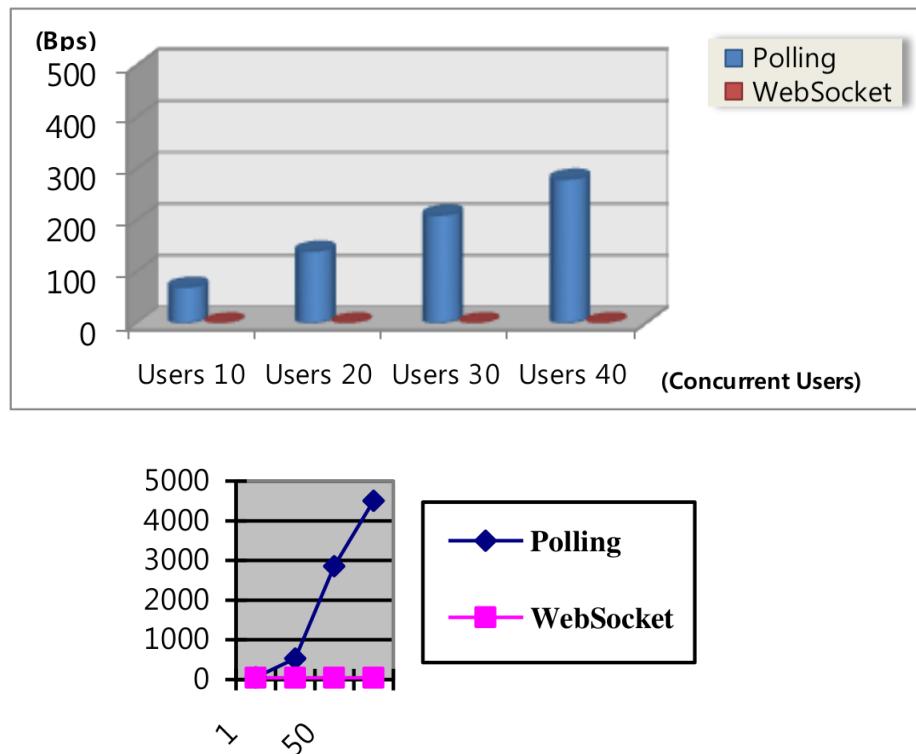
Gambar 2. 1: Data Perbandingan Websocket dengan HTTP Long/Polling

WS = WebSocket, LP = Long Polling, P = Polling

Jiang Feng-Yan (2012), dalam perkembangan sistem informasi, *web tree component* merupakan bagian penting dalam urusan menampilkan data, namun sampai sekarang masih banyak persoalan, mulai dari efisiensi komunikasi antara *server* dan *client*, menampilkan data, *updating rate*, dan lain-lain [3].

Tanapat Anusas-amornkul, Chanapa Silawong (2014), Seiring Websocket menjadi protokol standar HTML5 untuk *real-time* dan merupakan *full-duplex*, namun standar algoritma standar kompresi pada Websocket belum memiliki standar, banyak yang menggunakan Algoritman Deflate, tapi masih banyak algorima yang lain, lalu mana yang terbaik, peneliti ingin menemukan jawabannya pada penitian kali ini dengan membandingkannya melalui beberapa kebutuhan, dengan tipe dokumen yang dikirim, ukuran berkas, berkirim pesan instan dan permaianan *video online* [4].

Jin-tae Park , Hyun-seo Hwang, Jun-soo Yun, Il-young Moon (2014), Perbedaan paling krusial HTML5 dan sebelum HTML5 terletak pada protokol barunya, yaitu Websocket, karena tujuan Websocket adalah menyelesaikan permasalahan dalam hal metode komunikasi dengan segala batasanya, dalam studi ini peneliti ingin menganalisa Websocket tersebut khususnya pada data yang berupa multimedia. Dari hasil percobaan dengan menambah jumlah pengguna secara konkuren atau iterasi, metode Websocket menggunakan jauh lebih sedikit sumber daya dan waktu respon yang lebih cepat ketimbang Ajax [5].



Gambar 2. 2 : Grafik Perbandingan WebSocket dengan HTTP Poling

D. Skvorc, M. Horvat, S. Srbljic (2014), Paradigma *request-response* pada *synchronous* HTTP yang digunakan beberapa dekade ini, padahal web modern ini membutuhkan cara komunikasi yang lebih fleksibel dan *asynchronous* untuk pertukar pesan antar *client* dan *server*, lalu peneliti membandingkan protokol yang menggunakan *asynchronous* yaitu Websocket dengan protokol TCP yang menggunakan konsep *synchronous* [6].

## 2.2 State of The Art

Tabel 2. 1 : *State of The Art*

No	Peneliti	Tahun	Judul	Masalah	Metode	Hasil
1.	Victoria Pimentel, Bradford G. Nickerson	2012	Communicating and Displaying Real-Time Data with WebSockets	Latency adalah masalah yang signifikan dalam aplikasi seperti kontrol jaringan sistem.	HTTP Polling & Long Polling, Websoc ket	Dari percoba membanding kan ketiga teknologi tersebut menggunakan jaringan internet dari berbagai lokasi, Kanada, Venezuela, Sewedia dan Jepang pada rate traffict 4Hz, diketahui dari tabel bahwa WebSocket memiliki mengurangi tingkat latensi yang lebih baik pada sistem berbasis <i>real-</i>

						<i>time.</i>
2.	Jiang Feng-Yan	2012	Application Research of Web Socket Technology on Web Tree Component	Mengalami masalah komunikasi di komponen web, perihal efisiensi, updating rate, dan masalah menampilkan data, dll. Lalu membandingkan solusi terbaik antara teknologi WebSocket dan ajax polling	Websoc ket Tecnology dan Ajax Polling	Setelah dibandingkan dengan beberapa paramater, WebSocket lebih effisien dalam penggunaan sumber daya server, mengurangi node data transmision, menghemat dari sisi penggunaan dana dan penerapan yang relatif sederhana.
3.	Tanapat Anusas-amornkul, Chanapa Silawong	2014	The Study of Compressison Algorithm s for WebSocke	Seiring websocket menjadi protokol standar HTML5 untuk <i>real-</i>	Compression algorith ms, Huffma n, LZ77,	Dari 4 algoritma kompresi yang dibandingkan , masing-masing

			t Protocol	<i>time</i> dan merupakan <i>full-duplex</i> , namun standar algoritma standar kompresi pada WebSocket belum memiliki standar, banyak yang menggunakan Algoritman Deflate, tapi masih banyak algorima yang lain, lalu mana yang terbaik, peneliti ingin menemukan jawabannya pada penitian kali ini.	LZW dan Deflate	memiliki kelebihan pada tipe dokement tertentu, tapi dilihat dari matrik performa secara keseluruhan, algoritma Deflate menunjukan hasil terbaik.
4.	Jin-tae Park ,	2014	Study of HTML5	Perbedaan paling krusial	Metode Websoc	Dari hasil percobaan

	Hyun-seo Hwang, Jun-soo Yun, Il-young Moon	WebSockets for a Multimedia Communication	HTML5 dan sebelum HTML5 terletak pada protokol barunya, yaitu WebSocket, karena tujuan WebSocket adalah menyelesaikan permasalahan dalam hal metode komunikasi dengan segala batasanya, dalam studi ini peneliti ingin menganalisa Websocket tersebut khususnya pada data yang berupa multimedia.	ket dan Metode Polling (Ajax)	dengan menambah jumlah pengguna secara konkuren, metode WebSocket menggunakan jauh lebih sedikit sumber daya dan waktu respon yang lebih cepat ketimbang Ajax.
--	--	---	---	-------------------------------	--

5.	D. Skvorc, M. Horvat, S. Srbljic	2014	Performance Evaluation of Websocket Protocol for Implementation of Full-Duplex Web Streams	Paradigma <i>request-response</i> pada synchronous HTTP yang digunakan beberapa dekade ini, padahal web modern ini membutuhkan cara komunikasi yang lebih <i>flexibel</i> dan <i>asynchronus</i> untuk pertukar pesan antar <i>client</i> dan <i>server</i>	Protokol Websoc ket dan Protoko l TCP	Hasil komperasi kedua teknologi tersebut diukur dari durasi handshake, TCP 0,965 ms sedangkan Websocket 3,533 ms. Saat mengirim pake pesan berukuran kecil Websocket mengalami fluktuasi 150 % sampai 250 %, sedangkan saat mengirim paket pesan yang lebih besar stabil pada kisaran 60 % - 70 %.
----	----------------------------------	------	--	---	---------------------------------------	--

## **Kesimpulan**

Protokol WebsScket telah coba di uji dan diterapkan pada berbagai kepentingan, yang membedakan penelitian ini dengan penelitian-penelitian sebelumnya adalah pada implementasi penggunaannya. Pada penelitian ini aplikasi akan diterapkan pada *mobile* untuk lelang *online* secara *realtime*.

## 2.3 Landasan Teori

### 2.3.1 Lelang

Pengertian lelang menurut pasal 1 Vendu Reglement yang hingga saat ini masih berlaku. Menurut pasal tersebut lelang adalah setiap penjualan barang di muka umum dengan cara penawaran harga naik – naik, turun – turun, dan atau tertulis melalui usaha mengumpulkan para peminat atau peserta lelang yang dipimpin oleh pejabat lelang atau Vendemeester [1].

### 2.3.2 Klasifikasi Lelang

Pasal 1 angka 2 dan 3 Keputusan Menteri Keuangan No. 450/KMK 01/2002, sebagaimana diubah dengan Peraturan Menteri Keuangan No. 40/PMK 07/ 2006 mengklasifikasikan lelang menjadi :

- a. Lelang Eksekusi, jenis lelang ini merupakan penjualan umum untuk melaksanakan atau mengeksekusi putusan atau penetapan Pengadilan atau dokumen yang dipersamakan dengan putusan Pengadilan, seperti hipotik, Hak Tanggungan atau Jaminan Fidusia.
- b. Lelang Non-Eksekusi, jenis lelang ini merupakan penjualan umum di luar pelaksanaan putusan atau penetapan Pengadilan yang terdiri dari :
  - Lelang barang milik/dikuasai negara
  - Lelang sukarela atas barang milik swasta

Lelang yang dimaksud pada penelitian ini adalah lelang non eksekusi.

### 2.3.3 Lelang Online

Lelang yang di selenggarakan di internet. Terdapat banyak variasi format penyelenggaraan lelang *online*. Seperti Descending Dutch Auctions, First-Price Sealed-bid, English Auction, Vickrey Auction dan kombinasi dari dua format. Namun, yang paling sering terkenal adalah English Auction [7].

Lelang online dengan format English Auction diawali dengan mengumukan harga awal, kenaikan penawaran dan waktu berakhirnya lelang. Untuk mengikuti sebuah lelang, penawar harus menawar lebih tinggi dari penawar sebelumnya, pemenang adalah peserta lelang yang melakukan penawaran tertinggi pada waktu yang telah ditentukan [8].

### 2.3.4 WebSocket

Websocket adalah standar baru untuk berkomunikasi *realtime* pada *platform* web dan aplikasi *mobile*, Websocket dirancang untuk diterapkan di peramban dan pada *server* web atau pada *client* dan *server*. Webscoket merupakan protokol yang menyediakan saluran komunikasi jaringan *full-duplex*, yang artinya dapat berkomunikasi dan bertukar data secara dua arah antara *server* dan *client* dalam waktunya yang bersamaan, melalui koneksi TCP tunggal, protokol ini di strandarisasi oleh IETF sebagai RFC 6455 pada tahun 2011 yang sekaligus menjadi standar baru komunikasi pada HTML5 [9].

Keunggulan Websocket :

- Websocket memungkinkan *server* mendorong atau *push* kepada semua klien yang sedang terhubung secara *realtime*.
- Dapat mengurangi trafik atau lalu lintas pada jaringan yang tidak perlu dan latensi dengan menggunakan *full-duplex*.

- Streaming melalui *proxy* dan *firewall*, mendukung komunikasi simultan dari hulu hingga ke hilir.
- Kompetibel dengan pre-WebSocket atau sebelum menggunakan HTML5, sehingga memberi peluang yang luas untuk perlahan bermigrasi ke teknologi WebSocket.

Jenis-jenis aplikasi yang sebaiknya menggunakan WebScoket

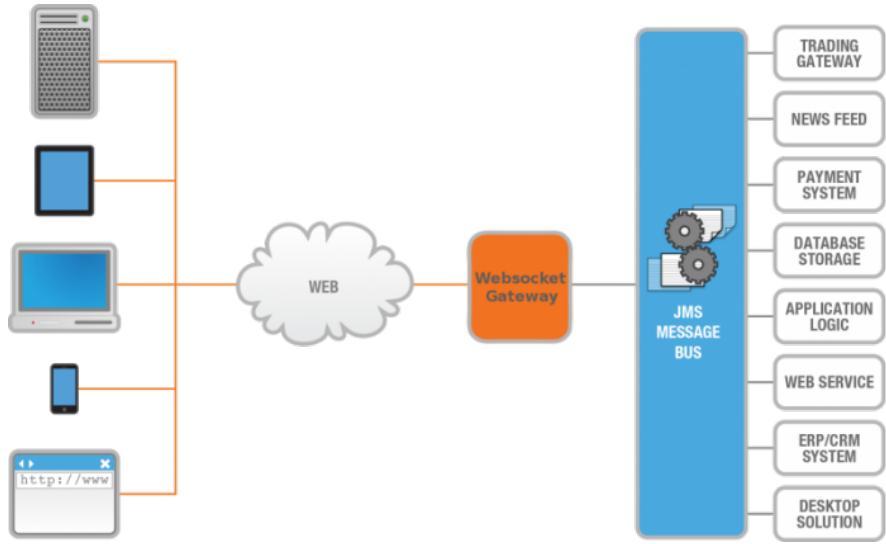
- Aplikasi membutuhkan pertukaran data secara langsung atau *real-time*.
- Aplikasi yang digunakan oleh banyak pengguna atau *multi-user*.
- Aplikasi yang bisa digunakan pada semua *platform modern* saat ini.

Teknologi klien yang mendukung Webscoket

- Pada peramban, WebSocket sebagai standar jalur komunikasi HTML5 sudah di dukung pada hampir seluruh peramban populer sekarang ini, seperti Mozilla Firefox, Google Chrome, Opera, dll
- Pada perangkat bergerak atau mobile, WebSocket kompetibel dengan Android, iOS, Windows Phone hingga Firefox OS

Teknologi server yang mendukung Webscoket

- Bahasa pemrograman disisi *server*, seperti Node.js, Ruby, PHP, .Net, Java, Objective-C, dll. Mendukung WebSocket sebagai protokol komunikasi dan pertukaran data.



Gambar 2. 3: Alur Pemrosesan pada *Websocket*

### 2.3.5 Socket.IO

Socket.IO adalah sebuah *library* yang dibangun di atas Websocket dan memberikan mekanisme-mekanisme untuk mengatasi masalah ketika fungsi yang sama tidak tersedia, maka Socket.IO menyediakan fungsi tersebut [10]. Socket.IO bertujuan membuat aplikasi *realtime* menjadi mungkin digunakan pada setiap peramban dan perangkat *mobile* dengan mudah, menyamarkan perbedaan antar mekanisme transfer data yang ada sekarang, serta menggunakan 100 persen JavaScript.

### 2.3.6 PostgreSQL Database

PostgreSQL adalah sebuah *open source database management system* yang digunakan untuk kepentingan umum serta kaya fitur. PostgreSQL telah dikembangkan lebih dari 20 tahun oleh grup Database Research di University of California dan komunitas open source di seluruh dunia [11].

### 2.3.7 Android

Android adalah sistem operasi perangkat mobile yang berjalan di atas kernel Linux. Dengan menggunakan bahasa pemrograman Java

untuk membuat aplikasi di dalamnya. Android bersifat open source dan gratis untuk digunakan, dipelajari, dikembangkan dan di distribusikan ulang.

#### 2.3.8 Web Service

Web Service adalah suatu mekanisme interaksi yang bertujuan menunjang interoperabilitas interaksi antara beberapa sistem pada suatu jaringan. Implementasi dari *web service* dapat dilakukan pada *platform* apapun dan dapat dibangun dengan berbagai macam bahasa pemrograman [12].

Kelebihan Web Service diantaranya :

- Dapat dibuat dan diakses dengan bahasa pemrograman apapun, sehingga *Language independent*.
- Dapat berkerja lintas *platform*, sehingga tetap dapat berkerja di sistem operasi yang berbeda.
- Dapat digunakan sebagai jembatan ke *database* aplikasi pihak ketiga tanpa perlu mengetahui database apa yang digunakan.

#### 2.3.9 JSON (JavaScript Object Notation)

JSON atau JavaScript Object Notation adalah format pertukaran data yang mudah dibaca oleh manusia dan juga lebih ringan di urai serta digunakan oleh mesin komputer. JSON merupakan bagian dari bahasa pemrograman JavaScript dan edisi ketiga Standard ECMA-262 - Desember 1999, namun dapat digunakan untuk berbagai bahasa pemrograman modern lain [13].

```

1  {
2      "success": true,
3      "status": "OK",
4      "message": "Success load list of auction joined",
5      "user_detail": {
6          "id": 3,
7          "name": "Diky Arga",
8          "username": "dikyarga",
9          "avatarUrl": "https://www.bukalapak.com/images/default_avatar/medium/default.jpg",
10         "auctionsJoinedCount": 5,
11         "wonAuctionsCount": 2
12     },
13     "auctionsJoined": [
14         {
15             "auctionId": 2,
16             "running": true,
17             "isRunning": 1,
18             "time_left": 109090998,
19             "title": "Gundam ukuran asli"
20         },
21         {
22             "auctionId": 3,
23             "running": false,
24             "isRunning": 0,
25             "time_left": 0,
26             "title": "Tamiya tanpa gravitasi"
27         },
28     ]
29 }
```

Gambar 2. 4: Contoh penulisan data menggunakan format JSON

### 2.3.10 Nodejs

Node.js adalah JavaScript *runtime* yang berjalan pada sisi *server*, dibangun di atas Chrome's V8 JavaScript *engine*. Node.js menggunakan model *event-driven, non-blocking* I/O sehingga menjadikan Node.js ringan dan efisien [14].

### 2.3.11 NPM

NPM adalah *package manager* untuk paket atau modul Node.js [15]. NPM secara otomatis terpasang ketika Node.js telah terpasang. Website [www.npmjs.com](http://www.npmjs.com) menyediakan ribuan paket atau modul yang siap di unduh dan digunakan secara gratis, karena bersumber dari paket atau modul yang bersifat *free and open source*. Sehingga orang dapat menggunakan ulang modul yang telah dibuat orang lain, tanpa harus membuatnya dari awal kembali.

Npm saat ini adalah *package manager* terbesar di dunia, dengan lebih dari 350000 modul dengan rata-rata unduhan oleh pengguna setiap bulan hingga 18 milliar kali [16].

#### 2.3.12 Express

Express adalah kerangka kerja atau *framework* Node.js untuk membangun aplikasi web yang mengusung minimalitas dan fleksibelitas [17].

Express.js berjalan diatas Node.js yang berada di sisi server, biasa digunakan untuk menangani *Routing*, Request, Response dan I/O process dengan mudah, sehingga sering dimanfaatkan untuk membuat *backend API*.

#### 2.3.13 React Native

React Native adalah sebuah Standard Development Kit (SDK) yang dapat berjalan pada berbagai platform perangkat mobile seperti Android dan iOS [18].

React Native menggunakan bahasa pemrograman JavaScript berjalan diatas *library* React JS yang di inisiasi oleh Developer Facebook dan kembangan bersama komunitas *open-source*.

#### 2.3.14 Amazon Web Service (AWS)

Amazon Web Services adalah sekumpulan layanan-layanan berbasis Cloud Computing yang di sediakan oleh Amazon sejak tahun 2006 [19].

AWS menawarkan banyak layanan cloud yang dapat digunakan sebagai server berbagai bahasa pemrograman dan satu layanan dengan layanan lainnya dapat saling berintegrasi tanpa harus keluar dari AWS.

Layanan insfrastruktur IT AWS telah digunakan dilebih dari 190 negara didunia dan memiliki server disemua benua.

#### 2.3.15 Test Driven Development

Test Driven Development atau TDD adalah sebuah teknik membangun perangkat lunak secara berkelanjutan dengan menggunakan sekumpulan test yang di spesifikasi di awal sebelum menulis code [20].

### 2.3.16 Mocha (Test Runner)

Mocha adalah testing framework untuk JavaScript yang kayak fitur dan dapat di gunakan pada Node.js serta perambah [21].

Pada penelitian ini fitur dari Mocha yang di gunakan adalah *test runner* atau untuk menjalankan *test* dengan aturan yang di berikan oleh Mocha.

In your editor:

```
var assert = require('assert');
describe('Array', function() {
  describe('#indexOf()', function() {
    it('should return -1 when the value is not present', function() {
      assert.equal(-1, [1,2,3].indexOf(4));
    });
  });
});
```

Back in the terminal:

```
$ ./node_modules/mocha/bin/mocha

Array
#indexOf()
  ✓ should return -1 when the value is not present

1 passing (9ms)
```

Gambar 2. 5: Contoh sintaks dan penggunaan Mocha

### 2.3.17 Chai (Test Assertion)

Chai adalah sebuah *assertion library* untuk Node.js dan *browser* yang dapat dipadukan dengan berbagai *testing framework* JavaScript lainnya [22]. Pada penelitian kali ini Chai dipadukan dengan Mocha.

Ada tiga jenis gaya atau format yang dapat digunakan pada Chai, yaitu: Should, Expect dan Assert seperti pada contoh berikut ini.



Gambar 2. 6: Contoh penggunaan tiga format penulisan tes dengan Chai

## 2.4 Deskripsi Tempat KKI

PT. Hacktivate Teknologi Indonesia adalah lembaga pendidikan *coding bootcamp* intensif yang mencetak *full stack developer* menggunakan bahasa pemrograman JavaScript, meliputi web (*front-end* dan *back-end*) dan *mobile*.

Informasi tentang PT. Hacktivate Teknologi Indonesia :

Alamat : Jl. Sultan Iskandar Muda No.7, RT.5/RW.9, Kby. Lama Sel., Kby. Lama, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12240, Indonesia

Telepon : +62 21 7238756

Email : [admissions@hacktiv8.com](mailto:admissions@hacktiv8.com)

## 2.5 Logo dan Makna Tempat KKI



Gambar 2. 7: Logo PT. Hactivate Teknologi Indonesia

Logo PT Hacktivate Teknologi Indonesia berbentuk serigala yang mencerminkan sifat cerdik atau cerdas dari binatang serigala. Warna *orange* dipilih karena dianggap memiliki arti semangat atau optimis.

## 2.6 Struktur dan Job Deskripsi

### 2.6.1 Tempat KKI

Struktur organisasi pada PT. Hacktivate Teknologi Indonesia [23].

- Managing Director

Bertanggung jawab dalam menetapkan visi misi perusahaan dan mengkomunikasikannya kepada seluruh anggota organisasi. Bertanggung jawab terhadap semua aktivitas dalam perusahaan Hacktiv8, menjamin segala kegiatan bisnis di PT. Hacktivate Teknologi Indonesia bergerak bersinergi dan efektif. Menjadi jembatan antara divisi. Berperan sebagai *main decision maker*, memimpin dan memfasilitasi seluruh suara dalam organisasi.

- Director of Campus Operations

Bertanggung jawab dalam membuat presentasi dan materi pembelajaran untuk kurikulum di perusahaan PT. Hacktivate Teknologi Indonesia, merancang pembelajaran berorientasi proyek untuk para peserta didik, dan membuat metode penilaian untuk mempersiapkan peserta didik agar selanjutnya dapat mengikuti karir program dari perusahaan.

- Head of Admissions

Bertanggung jawab dalam menyusun dan merancang proses administrasi penerimaan peserta murid didik baru mulai dari awal pendaftaran sampai penyaluran kerja ke perusahaan hiring partner.

- Head of Business Operations

Bertanggung jawab dalam seluruh kegiatan operasional bisnis perusahaan, termasuk di dalamnya tanggung jawabnya yaitu memastikan semua sarana yang tersedia dapat memfasilitasi kegiatan yang dijalankan oleh perusahaan.

- Digital Media Producer

Bertanggung jawab untuk pembuatan situs web dan aplikasi seluler, mengawasi produksi media digital, menggabungkan elemen media yang berbeda seperti file *audio*, *video* dan teks sebagai konten web, DVD, presentasi untuk perusahaan.

- Marketing Director

Bertanggung jawab dalam meningkatkan pemasaran program kursus di perusahaan, dimana didalamnya termasuk melakukan riset untuk mengetahui kebutuhan para calon peserta didik, mengembangkan nilai jual program kursus untuk menarik minat para calon peserta didik, merancang strategi pemasaran dan menemukan cara yang tepat untuk menjual program kursus ke pasar.

- Marketing Communications Specialist

Bertanggung jawab untuk membuat strategi promosi dan komunikasi, bertindak sebagai penghubung antara tim internal dan team pihak ketiga (*vendor, media partner*, dan perusahaan *hiring partner*), membuat konten dan manajemen periklanan *digital* termasuk mengelola sosial media untuk mengembangkan aktivitas daring dan meningkatkan kesadaran terhadap program yang ada di PT. Hacktivate Teknologi Indonesia.

- Instructor

Bertanggung jawab untuk mengajar materi JavaScript, memberi latihan, kuis, tugas, ujian dan memantau perkembangan para peserta didik untuk tetap dapat mengikuti materi kurikulum pembelajaran.

- Student

Mengikuti pembelajaran dari para instruktur, mengerjakan tugas dan ujian yang diberikan, dan mengikuti proyek akhir.

## 2.7 Visi dan Misi Tempat KKI

Visi :

Mencetak profesional *full-stack developer* yang siap kerja di industri IT

Misi :

- Memberikan materi terkini, runtut dan lebih menekankan pada praktik daripada teori.
- Menyalurkan lulusan ke industri IT yang telah menjalin kerjasama
- Memberikan pelatihan yang intensif namun tetap memperhatikan sisi jasmani dengan program yoga dan rohani dengan mengadakan program *engineering empathy*

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Instrumen Penelitian**

Untuk menyusun suatu proyek penelitian, diperlukan data-data yang sesuai dengan pokok permasalahan yang dihadapi. Data dikatakan baik ketika data tersebut dapat mewakili objek yang sedang di teliti dan untuk mendapat data yang baik diperlukan serangkaian cara atau metode yang sesuai dengan kebutuhan penelitian.

#### **3.2 Prosedur Pengambilan Data**

Sesuai dengan sumber data dan tujuan penyusunan Laporan Kuliah Kerja Indistri ini serta untuk mendapatkan data yang benar - benar akurat dan relevan, maka dalam pengumpulan data penulis menggunakan beberapa teknik antara lain:

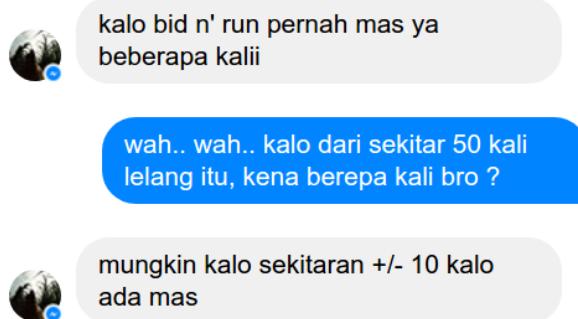
- Observasi

Teknik pengumpulan data dengan melakukan pengamatan langsung pada sistem lelang *online* yang sudah ada saat ini. Beberapa situs lelang *online* yang penulis amati diantaranya adalah ebay.com dan auctions.yahoo.co.jp

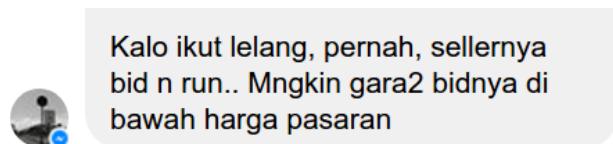
- Wawancara

Penulis melakukan wawancara secara daring dengan pelaku lelang *online* di forum atau group di sosial media guna mengetahui masalah yang sebenarnya dialami oleh calon pengguna sistem lelang *online*.

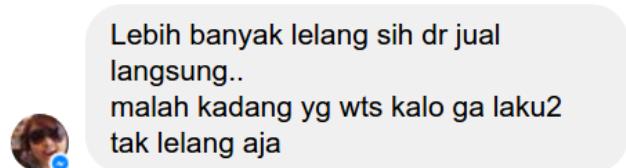
Berikut beberapa tangkapan layar wawancara melalui *chat* di sosial media Facebook :



Gambar 3. 1: Tangkapan layar wawancara 1



Gambar 3. 2: Tangkapan layar wawancara 2



Gambar 3. 3: Tangkapan layar wawancara 3

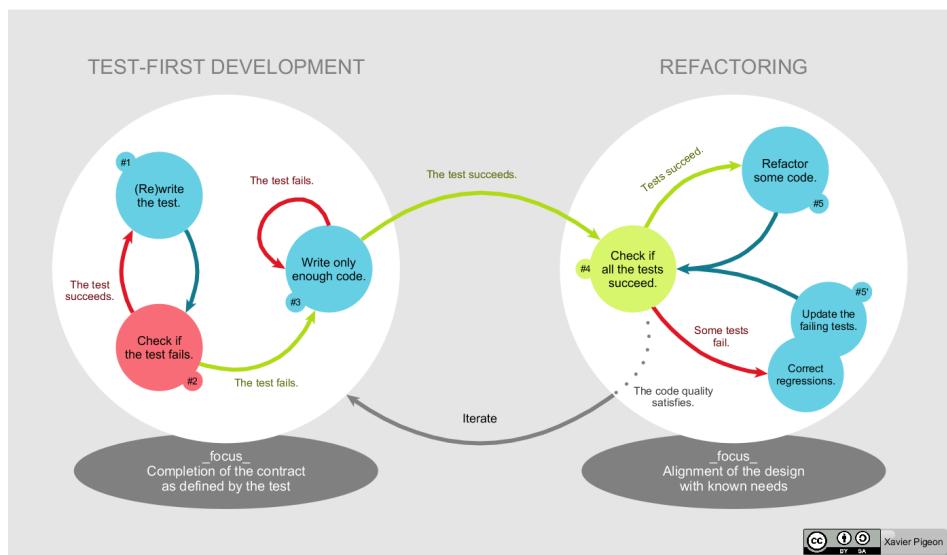
### 3.3 Teknik Analisis Data

Sistem yang akan dibangun merupakan sistem yang memungkinkan pengguna membuat lelang maupun mengikuti lelang yang terintegrasi dengan Bukalapak.com melalui aplikasi Android (*client*). Data yang yang dibutuhkan

disimpan di sisi *server* Amazon Web Service sekaligus sistem Bukalapak.com via *restful API*.

### 3.4 Metode

Dalam penelitian ini menggunakan metode pengembangan Extream Programming, yaitu Test Driven Development (TDD) atau yang sering dikenal juga dengan Unit Test-first Development, yaitu dengan membuat test case sebelum membuat fungsional code. Berikut adalah 5 tahap atau fase dalam menerapkan metode ini :



Gambar 3. 4: Alur Pengembangan Perangkat Lunak dengan TDD

#### 1. Membuat Test

Langkah yang pertama dilakukan adalah membuat *unit test* setiap fitur yang akan dibuat. Untuk membuat *test* dibutuhkan deskripsi fitur yang jelas, sepesifikasi, hingga kebutuhan mulai dari *input* sampai *output* yang di keluarkan.

## 2. Menjalankan Test

Pastikan ketika menjalankan *test*, *test* yang baru saja dibuat gagal atau tidak memenuhi hasil yang di harapkan. Difase ini sering disebut *red-phase*.

## 3. Menulis Code

Menulis kode program pada tahap ini adalah dengan tujuan utama membuat program lulus dari *test*. Di bagian ini mungkin kode masih boleh berantakan atau kurang elegan, karena fokus utama nya adalah lulus *test* dan tidak perlu membuat fungsi tambahan di luar meluluskan *test*. Karena akan di *refactor* pada proses ke lima.

## 4. Jalankan Ulang Test

Pastikan semua *test* lolos, jika belum, lakukan iterasi pada kode hingga semua lolos *test*. Dengan iterasi terus menerus seperti ini *programmer* dapat mudah memantau kinerja aplikasi sekaligus menghindari kesalahan kode setelah menambah fitur baru.

## 5. Refactoring

Di fase inilah kode yang berantakan, duplikat dan tidak efisien di rapikah, dengan adanya iterasi *testing* juga sekaligus memastikan perubahan yang dilakukan menghasilkan hal yang sama sebelum di *refactoring*. Dengan seiring berkembangnya aplikasi kode yang semakin banyak, TDD membantu membuat kode tersebut menjadi lebih mudah di pelihara dan dibaca kembali oleh programmer tersebut, maupun programmer lain. Karena setiap fungsi telah di beri deskripsi, contoh input dan outputnya, sehingga kode tersebut dapat menjelaskan dengan sendirinya.

### 3.4.1 Membuat Test

Sebelum membuat *test case*, penulis menganalisa setiap fitur yang akan dibuat pada aplikasi berdasarkan *use case* dan diagram aktifitas sehingga mengetahui alur setiap fitur berserta data yang dibutuhkan. Terdapat beberapa fitur utama yang harus tersedia pada aplikasi ini, berikut diantaranya :

- Fitur *authentication* meliputi sub-fitur *register*, *login* dan *logout*.
- Fitur lelang meliputi sub-fitur membuat, menghapus, pencarian dan menampilkan lelang.
- Fitur *real-time bidding* meliputi sub-fitur penawaran berdasarkan kelipatan tertentu dan Buy It Now (BIN).

Setiap satu fitur penulis akan membuat sebuah *file* untuk menampung seluruh *test* yang berkaitan terhadap fitur tersebut, dengan ketentuan bahwa setiap sub fitur setidaknya akan dibuat dua kondisi *test* mencakup positif *test* dan negatif *test*, sehingga dapat memastikan bahwa *function* yang dibuat nantinya dapat menangani kondisi ketika semua parameter yang dibutuhkan benar dan dapat mengantisipasi jika parameter yang dikirimkan salah.

Setiap *test case* yang akan dibuat harus meliputi pengecekan *status header*, *property* pada *object* dan kesesuaian data dengan parameter inputan.

Pada penelitian ini, penulis tidak akan langsung membuat semua test case sekaligus, melainkan beriterasi setiap satu fitur telah selesai satu kali *cycle*, kemudian dibuat *test case* fitur lainnya dan disimpan pada sebuah *folder* bernama “test”.

### 3.4.2 Menjalankan Test

Setiap selesai membuat *test case* untuk suatu fitur, *test runner* akan dijalankan sesuai dengan *file* yang akan dicek dengan mengetikan

perintah pada *terminal* “mocha” di ikuti nama file yang akan dieksekusi. Ketika pengcekan pada tahap dua ini, diwajibkan semua *assertion* tidak memenuhi kodisi benar atau dengan kata lain semua *test case* yang baru dibuat gagal. Jika terdapat *test case* yang berhasil, artinya terdapat kesalahan saat membuat *test case* dan harus diperbaiki terlebih dahulu sebelum ketahap selanjutnya.

#### 3.4.3 Menulis Code

Penulisan *code* aplikasi yang dimaksud penulis pada tahap ini adalah akan membuat *function* untuk menangani *request* dan *response* yang berdasarkan *test case* yang telah dibuat, meliputi URL *endpoint*, HTTP *method*, *parameter* dan data yang telah dijalankan sebelumnya.

Setelah dianggap *code* yang dibuat akan dapat memenuhi *test case*, barulah lanjutkan pada proses selanjutnya.

#### 3.4.4 Menjalankan Ulang Test

Proses ini akan dilakukan agar dapat memastikan *code* yang ditulis pada tahapan sebelumnya memenuhi persyaratan *assertion* pada *test case* yang diinginkan. Namun, jika ketika dijalankan nantinya masih belum lolos *test case*, maka perlu dilakukan penulisan code ulang dan akan terus beriterasi dengan tahapan ini hingga semua lolos *test case*.

Setelah semua test case lolos pada setiap fitur, barulah dapat dilanjutkan pada tahapan selanjutnya.

#### 3.4.5 Refactoring

Proses pada tahapan ini sebenarnya tidak wajib dilakukan, karena aplikasi telah berkerja dengan semestinya dan telah diuji coba pada tahapan sebelumnya. Namun, akan lebih baik jika dilakukan *refactoring* agar *code* menjadi lebih rapi dan efisien.

Setiap kali selesai melakukan *refactoring*, akan dijalankan *test* ulang untuk memastikan bahwa proses refactoring ini tidak

merusak *function* yang sebelumnya telah memenuhi semua *test case*.

Jika proses ini membuat aplikasi yang sebelumnya lolos *test-case* menjadi tidak lolos, maka perlu dilakukan iterasi hingga kembali memenuhi semua test-case atau batalkan tahapan *refactoring* ini.

### 3.5 Pengujian

Penulis menggunakan metode Test Driven Development, pengujian dibuat terlebih dahulu sebelum menulis code aplikasi. Sehingga, aplikasi telah teruji secara otomatis setelah selesai di kerjakan.

## **BAB IV**

### **RANCANGAN SISTEM DAN IMPLEMENTASI**

Penelitian akan mengembangkan sebuah sistem aplikasi lelang *online* berbasis *mobile* yang terintegrasi *webservice* dan API BukaLapak.com. Dengan adanya aplikasi ini diharapkan dapat menjadi penjembatan antara pelelang dan peserta lelang yang dapat digunakan secara *realtime* dan dapat menghilangkan kasus *bid & run* yang kerap terjadi.

#### **4.1 Perancangan Kebutuhan Sistem**

Dalam membangun aplikasi ini, peniliti membutuhkan beberapa komponen, yang terdiri dari perangkat lunak dan perangkat keras.

##### **4.1.1 Kebutuhan Perangkat Lunak**

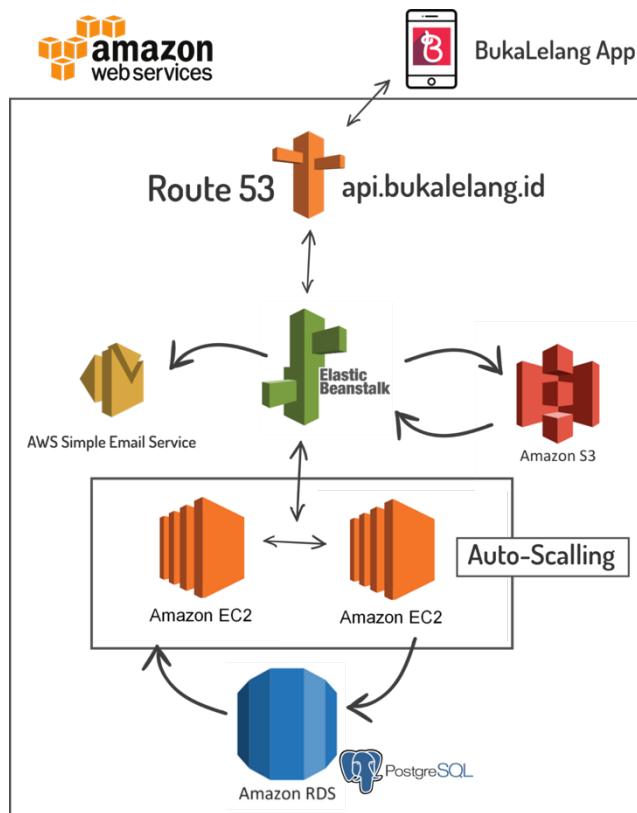
Spesifikasi kebutuhan perangkat lunak yang digunakan dibagi menjadi dua kategori, yaitu *development* yang digunakan membangun aplikasi secara *local* dan *production* yang digunakan di *server*.

- Development
  - Sistem Operasi TeaLinuxOS 8 - 64 Bit
  - Atom Text Editor
  - Android Studio
  - Postgres SQL
  - PgAdmin
  - NodeJS
  - npm
  - Socket.IO
- Production
  - Ubuntu Server 16.04 - 64 Bit
  - PgAdmin
  - NodeJS
  - npm

#### 4.1.2 Kebutuhan Perangkat Keras

Spesifikasi kebutuhan perangkat keras yang digunakan dibagi menjadi dua kategori, yaitu *development* yang di gunakan membangun aplikasi secara *local* dan *production* yang digunakan di *server*.

- Development
  - Intel Core i5
  - RAM 12 GB
- Mobile
  - Xiaomi Redmi 2 Prime
  - RAM 2 GB
  - Prosesor Quad-core 1.2 GHz Cortex-A53
  - GPU Adreno 306
- Server (Amazon Web Service)



Gambar 4. 1: Skema Layanan AWS yang di Gunakan oleh Aplikasi

- Elastic Beanstalk (Load Balancing dan Auto Scalling)
- RDS (Postgres Database)
- Route 53 (Domain Routing)
- S3 (Simple Storage)
- SES (Simple Email Service)
- EC2 Intence

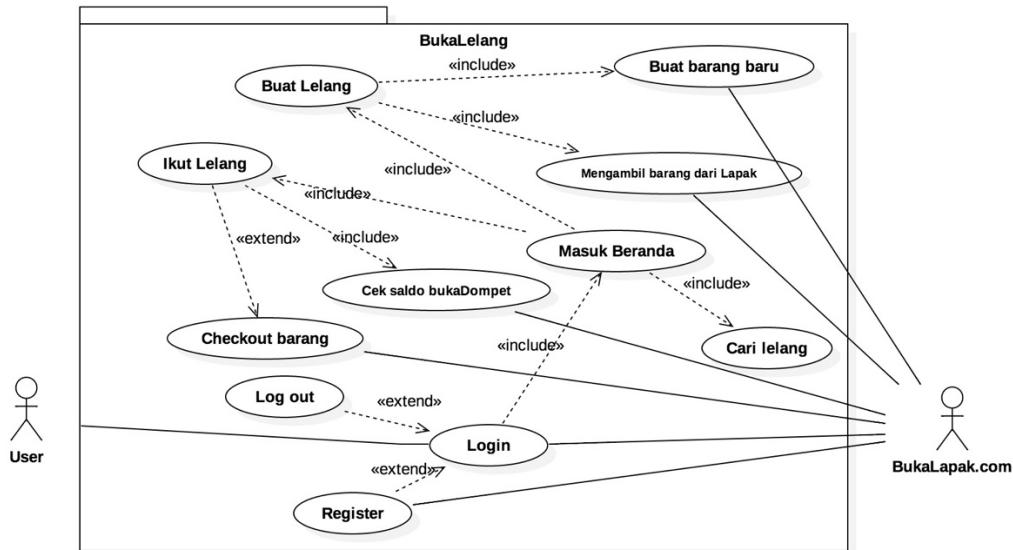
## 4.2 Analisis Perancangan Sistem

### 4.2.1 Identifikasi Aktor

Aktor merupakan pihak yang terlibat dan memiliki akses terhadap sistem aplikasi

- User (Pelelang & Peserta Lelang)
  - Individu yang dapat melakukan *login*, *register*, membuat lelang dan melakukan penawaran pada lelang.
- BukaLapak.com
  - Sebagai sistem penjembatan antara pelelang dan peserta lelang pembayaran dan produk.

#### 4.2.2 Pemodelan *Use Case*



Gambar 4. 2: Diagram Use Case

Berdasarkan use case diatas, maka dapat diketahui bahwa untuk dapat mengikuti lelang dan membuat lelang perlu memiliki akun yang terintegrasi dengan sistem Bukalapak.com. Sehingga ketika user mendaftar akan benar-benar juga akan terdaftar di sistem Bukalapak.com. Begitu pula ketika mengambil barang dari lapak yang sudah ada di Bukalapak.com untuk di lelang, saldo BukaDompet untuk memastikan pengguna memiliki jumlah uang yang cukup untuk melakukan penawaran pada lelang, hingga *checkout* barang dengan memilih jasa kurir pengiriman serta menetukan alamat pengiriman barang ketika pengguna berhasil menangkan lelang.

#### 4.2.3 Use Case Naratif

Use case naratif digunakan sebagai dokumentasi untuk menjelaskan langkah-langkah yang terjadi setiap interaksi dari *use case* yang telah dibuat. Berikut adalah *use case* naratif :

- a. Nama use case : Register
  - Aktor : User dan Bukalapak.com
  - Tujuan : Mendaftar akun
  - Deskripsi : Mendaftar akun untuk dapat membuat lelang sekaligus mengikuti lelang, karena sistem terintegrasi dengan API Bukalapak.com, jadi ketika membuat akun pada sistem, sekaligus membuat akun di Bukalapak.com
- b. Nama use case : Login
  - Aktor : User dan Bukalapak.com
  - Tujuan : Masuk akun ke sistem
  - Deskripsi : Untuk mendapatkan akses kedalam sistem. Karena sistem teringrasi dengan API Bukalapak.com, sehingga *login* dapat menggunakan akun ada di Bukalapak.com.
- c. Nama use case : Log out
  - Aktor : User
  - Tujuan : Keluar dari aplikasi
  - Deskripsi : Ketika pengguna keluar dari sistem, *session user* tersebut akan dihapus.
- d. Nama use case : Masuk Beranda
  - Aktor : User
  - Tujuan : Menampilkan daftar lelang yang sedang berjalan
  - Deskripsi : Setelah masuk kedalam aplikasi, pengguna melihat daftar list lelang yang sedang berjalan dari berbagai kategori, diurutkan berdasarkan waktu lelang yang mendekati waktu berakhirnya.
- e. Nama use case : Membuat lelang
  - Aktor : User dan Bukalapak
  - Tujuan : Membuat lelang baru
  - Deskripsi : Barang yang di lelang dapat dibuat dari produk baru maupun mengambil dari barang di lapak yang sudah ada di Bukalapak.com sebelumnya, sehingga pengguna tidak perlu repot-repot mengisi data dan mengunggah foto produk kembali. Begitu juga

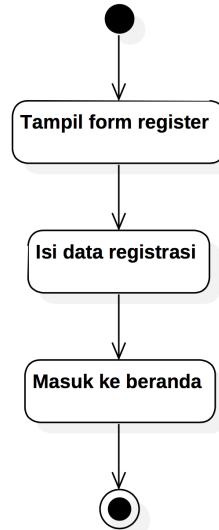
sebaliknya, ketika membuat lelang di aplikasi ini, secara otomatis juga membuat produk di lapak yang ada di Bukalapak.com

- f. Nama use case : Mencari lelang
  - Aktor : User
  - Tujuan : Melakukan pencarian lelang online
  - Deskripsi : Pencarian lelang dapat dilakukan dengan mengetikan judul barang yang di lelang.
- g. Nama use case : Mengikuti lelang
  - Aktor : User
  - Tujuan : Melakukan penawaran pada lelang
  - Deskripsi : Untuk dapat melakukan penawaran pengguna terlebih dahulu harus melakukan top up pada aplikasi Bukalapak.com dan memiliki saldo minimal sama dengan nominal tawaran yang diinginkan.
- h. Nama use case : Checkout
  - Aktor : User dan Bukalapak
  - Tujuan : Melakukan checkout barang setelah menang lelang
  - Deskripsi : Setelah berhasil memenangkan lelang pengguna melakukan *checkout* untuk memilih kurir yang digunakan serta alamat tujuan barang dikirim. Secara otomatis memotong saldo BukaDompet dan melakukan transaksi di Bukalapak.com. Sehingga dapat dilacak setiap proses pengiriman barang hingga sampai ke pemenang lelang tersebut.

#### 4.2.4 Activity Diagram

Activity Diagram dirancang sebagai permodelan urutan aktivitas dan prosedur logika dari aplikasi atau sistem yang akan dibangun. Perancangan ini dilakukan pada masing-masing *use case*.

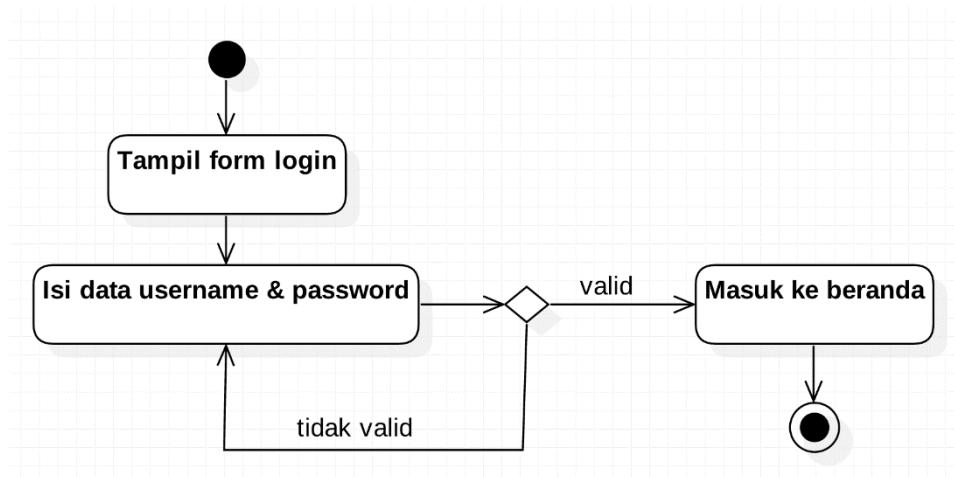
- Diagram Aktivitas Register



Gambar 4. 3 Diagram Aktivitas Register

Setelah berhasil login pengguna langsung masuk ke halaman beranda aplikasi yang berisi daftar lelang, tanpa harus *login* terlebih dahulu.

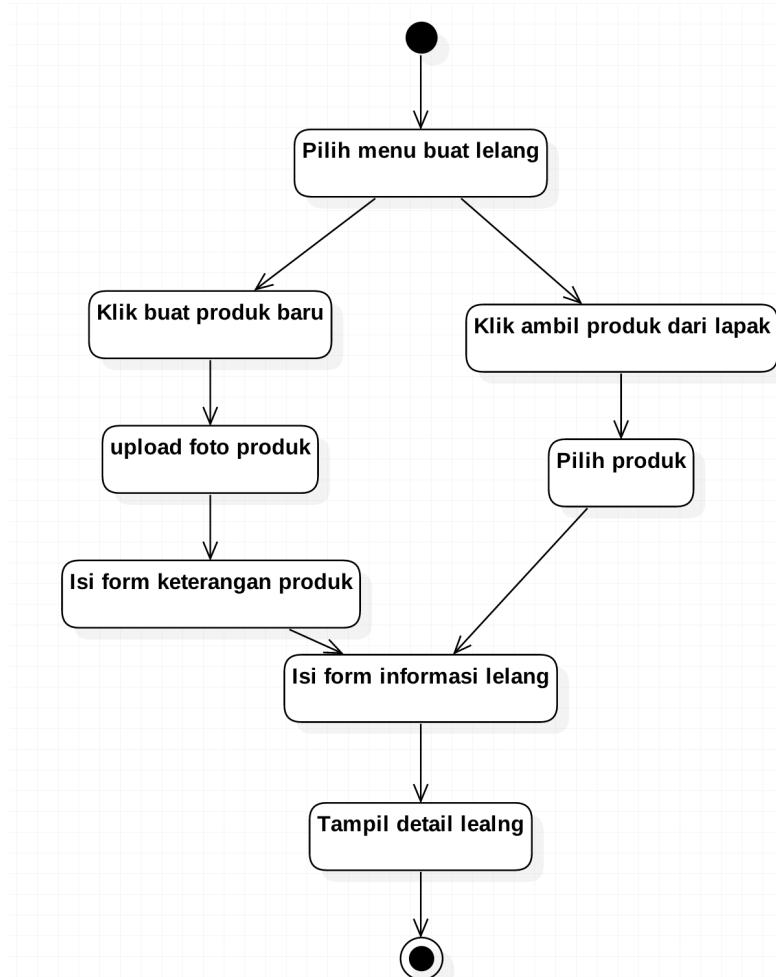
- Diagram Aktivitas Login



Gambar 4. 4 Diagram Aktivitas Login

Setelah memiliki akun di sistem maupun di [Bukalapak.com](#) pengguna dapat memasukan *username* dan *password* di *form login*. Kemudian sistem melakukan validasi dan mengarahkan ke halaman beranda aplikasi beserta memberikan *token* yang harus di sertakan ketika melakukan aktivitas lainnya jika *username* dan *password* tepat.

- Diagram Aktivitas Buat Lelang Baru



Gambar 4. 5: Diagram Aktivitas Buat Lelang Baru

Produk yang akan di lelang dapat di buat baru maupun mengambil dilapak pada Bukalapak.com jika sudah ada. Jika memilih buat lelang dengan produk baru, maka pengguna perlu mengunggah foto produk dan informasi mengenai produk yang akan di lelang, sedangkan jika memilih ambil dari lapak cukup memilih produk yang di lelang. Setelah memilih salah satu dari dua cara tersebut, pengguna mengisi informasi tentang lelangnya. Setelah lelang berhasil dibuat, pengguna akan diarahkan ke laman detail lelang tersebut.

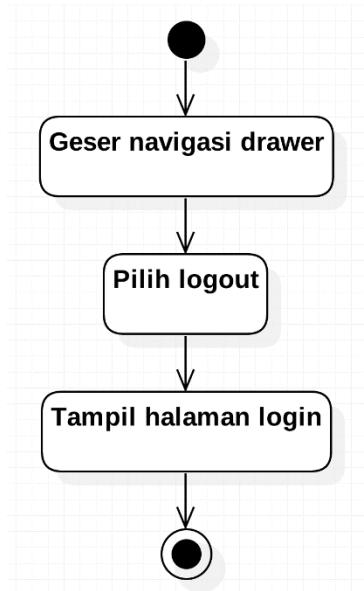
- Diagram Aktivitas Mengikuti Lelang



Gambar 4. 6: Diagram Aktivitas Mengikuti Lelang

Setelah menemukan barang lelang yang sesuai, pengguna dapat melakukan penawaran lebih tinggi dari penawaran sebelumnya atau juga langsung dapat membelinya dengan harga maksimal barang. Kemudian akan tercatat pada riwayat tawaran yang sekaligus memperbaharui data penawaran di semua pengguna aplikasi secara *realtime*.

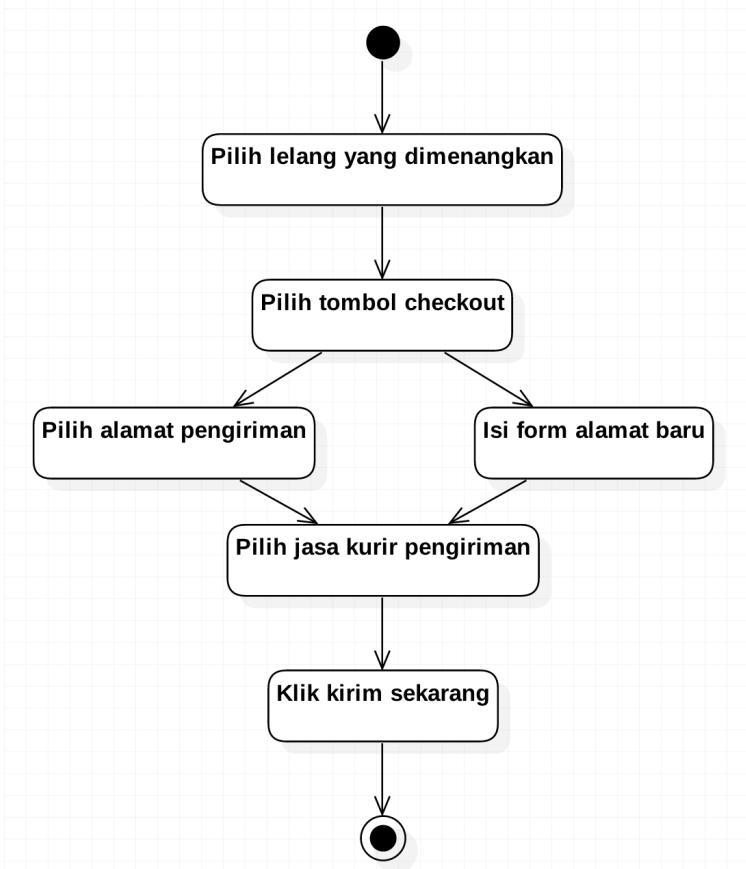
- Diagram Aktivitas Logout Aplikasi



Gambar 4. 7: Diagram Aktivitas Logout Aplikasi

Untuk mengeluarkan akun dari aplikasi, geser navigasi drawer atau tekan menu hamburger, lalu klik logout. Kemudian akan diarahkan ke halaman login.

- Diagram Aktivitas Checkout

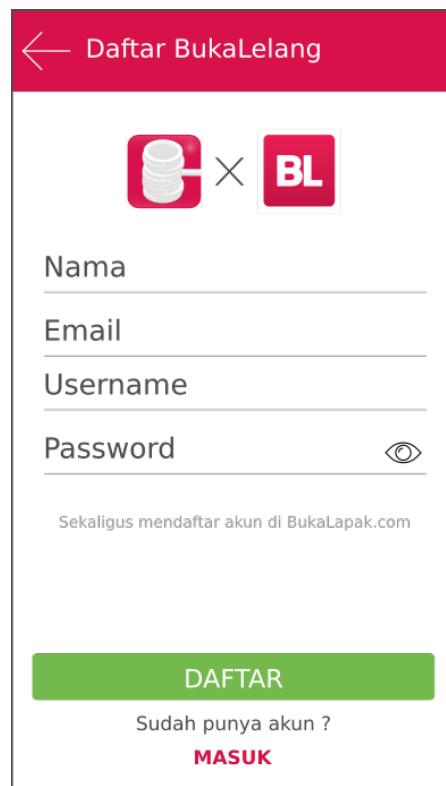


Gambar 4. 8: Diagram Aktivitas Checkout

Setelah memenangkan lelang, pemenang melakukan checkout barang. Jika sudah pernah memasukan alamat di sistem, maka pengguna hanya perlu memilih salah satu alamat dari daftar alamat. Namun, jika belum, pengguna harus mengisi formulir alamat baru. Setelah alamat dipilih sistem menampilkan jasa kurir pengiriman berikut dengan ongkos kirim. Pilih salah satu jasa kurir dan paketnya, klik kirim sekarang. Proses transaksi akan di teruskan di Bukalapak.com, sehingga dapat dilacak pengiriman barang melalui situs atau aplikasi Bukalapak.

### 4.3 Perancangan Antar Muka Android

#### 4.3.1 Desain Halaman Register

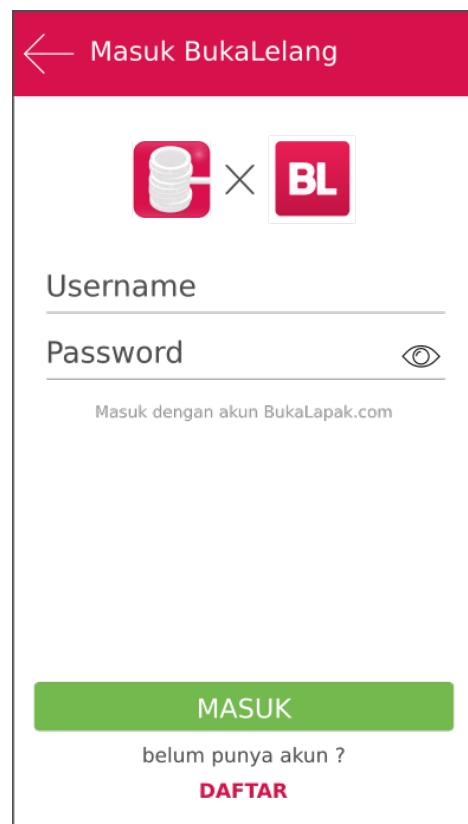


Gambar 4. 9: Desain Halaman Register

Terdapat informasi bahwa ketika membuat akun pada aplikasi ini juga akan terintegrasi dengan akun di Bukalapak.com.

#### 4.3.2 Desain Halaman Login

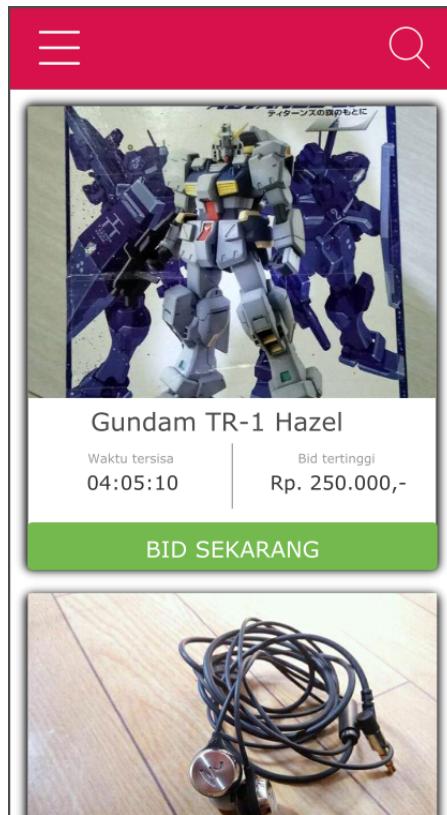
Setelah memiliki akun pada sistem aplikasi ini atau sistem Bukalapak.com, pengguna dapat masuk ke dalam aplikasi. Berikut desain rancangan halaman login.



Gambar 4. 10: Desain Halaman Login

#### 4.3.3 Desain Halaman Beranda Aplikasi

Berikut adalah halaman beranda yang tampil ketika pengguna berhasil masuk ke dalam aplikasi. Terdapat daftar lelang yang sedang berjalan dari berbagai kategori dan di urutkan berdasarkan waktu lelang yang paling mendekati selesai.



Gambar 4. 11: Desain Halaman Beranda Aplikasi

#### 4.3.4 Desain Halaman Membuat Lelang

Ketika pengguna menekan tombol buat lelang, akan diberikan dua opsi seperti berikut.



Gambar 4. 12: Desain Pop up Setelah Klik Buat Lelang

Produk yang akan dilelang dapat di buat baru maupun mengambil di lapak pada Bukalapak.com jika sudah ada. Jika memilih buat lelang dengan

produk baru, maka pengguna perlu mengunggah foto produk dan informasi mengenai produk yang akan dilelang. Seperti pada gambar berikut.

Buat Lelang

Upload Foto Produk

Judul

Pilih kategori

Deskripsi

Kondisi barang

Bekas  Baru

Berat (gram)

Harga dasar lelang

Kelipatan bid

Harga beli langsung

Pilih waktu selsesai lelang

**Pilih waktu**

**BUAT LELANG**

Gambar 4. 13: Desain Halaman Buat Lelang dari Produk Baru

Sedangkan jika memilih ambil dari lapak cukup memilih produk yang dilelang. Setelah memilih salah satu dari dua cara tersebut, pengguna mengisi informasi tentang lelangnya. Seperti pada gambar berikut.



Gambar 4. 14: Desain Halaman Buat Lelang dari Lapak

#### 4.3.5 Desain Halaman Detail Lelang

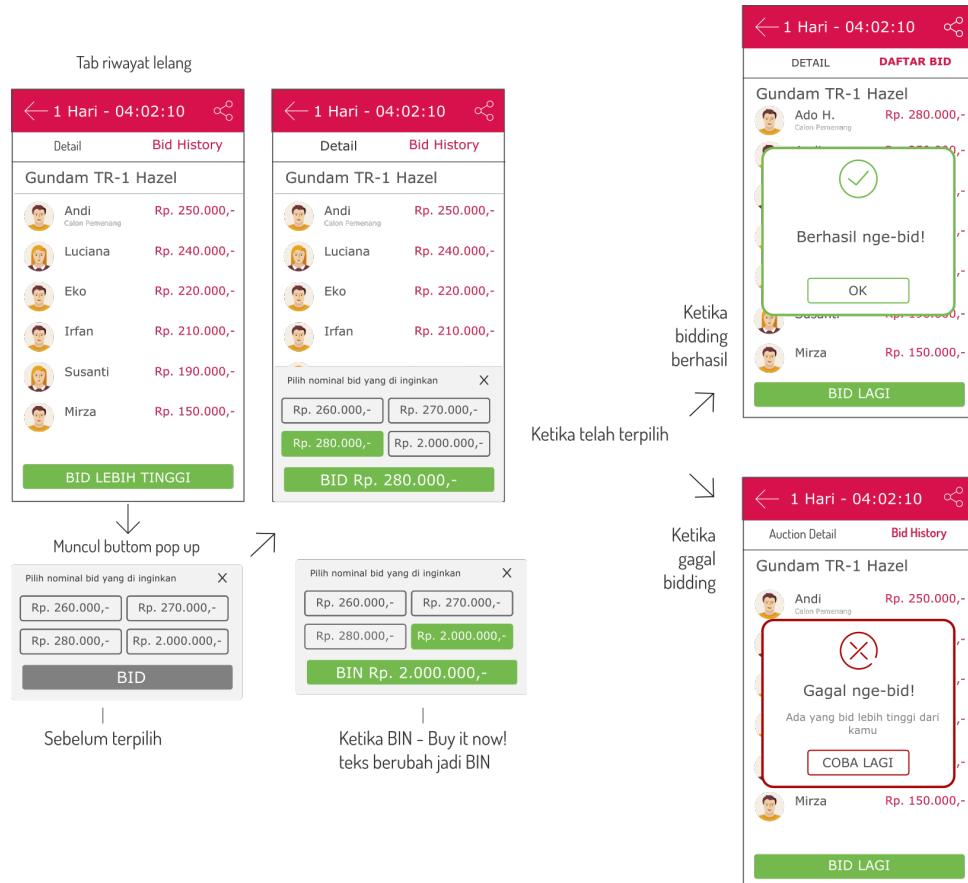
Dalam halaman ini ditampilkan semua informasi tentang barang yang dilelang dan informasi lelang. Seperti berikut ini.



Gambar 4. 15: Desain Halaman Detail Lelang

#### 4.3.6 Desain Halaman Riwayat Tawaran Lelang

Pada tab riwayat lelang tawaran lelang akan tampil daftar peserta lelang berserta nominal yang ditawarkan, diurutkan berdasarkan nominal tertinggi hingga terendah. Berikut gambaran mengajukan penawaran dan tab *history*.



Gambar 4. 16: Desain Riwayat Tawaran Lelang

#### 4.3.7 Desain Halaman Checkout

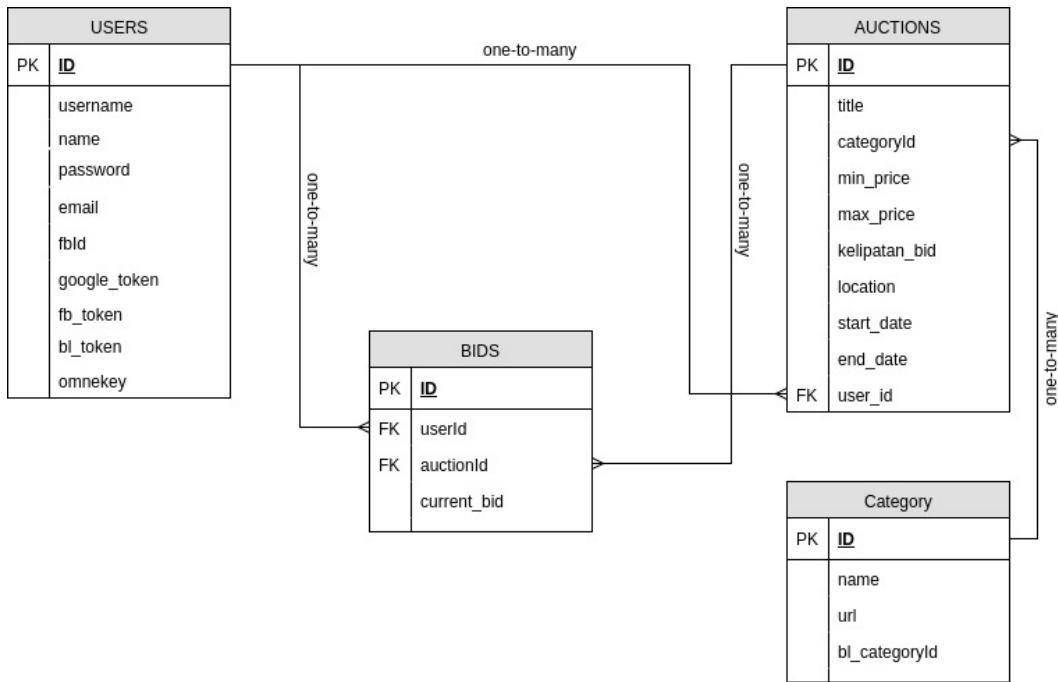
Halaman ini hanya bisa diakses oleh peserta lelang yang berhasil memangkan lelang. Pengguna mengisi alamat tujuan dikirimnya barang dan memilih jasa kurir pengiriman. Gambar rancangan desainnya sebagai berikut.



Gambar 4. 17: Desain Halaman Checkout Barang

#### 4.4 Perancangan Web Service

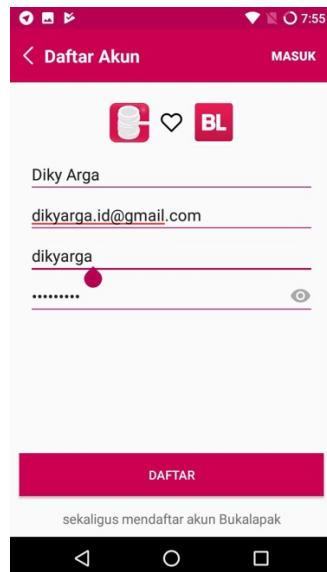
Perancangan web service dimulai dari merancang skema *database* yang akan dibuat, berikut adalah skema database aplikasi ini.



Gambar 4. 18: Skema Database

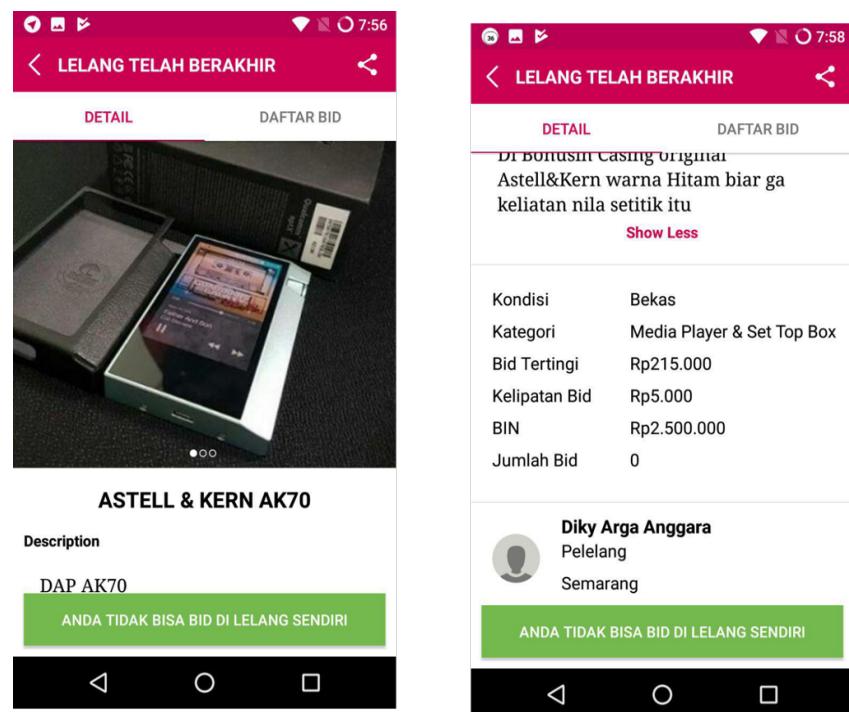
## 4.5 Implementasi Android

### 4.5.1 Implementasi Halaman Register



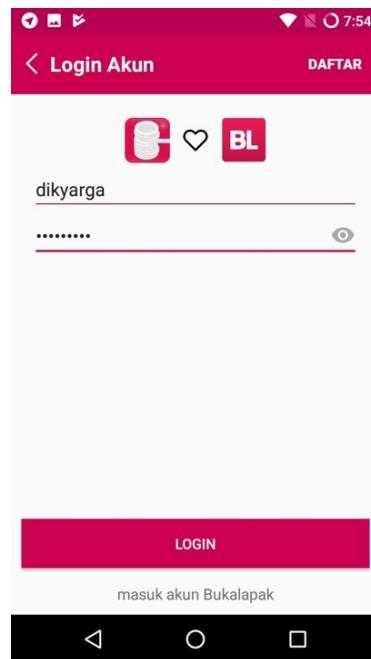
Gambar 4. 19: Antarmuka Implematsi Halaman Register

### 4.5.2 Implementasi Halaman Detail Lelang



Gambar 4. 20: Antarmuka Halaman Detail Lelang

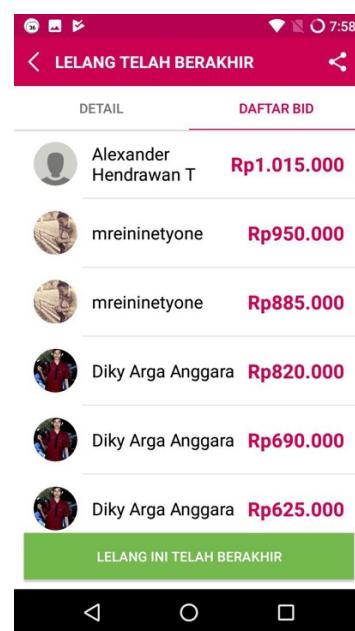
#### 4.5.3 Implementasi Halaman Login



Gambar 4. 21: Antarmuka Halaman Login

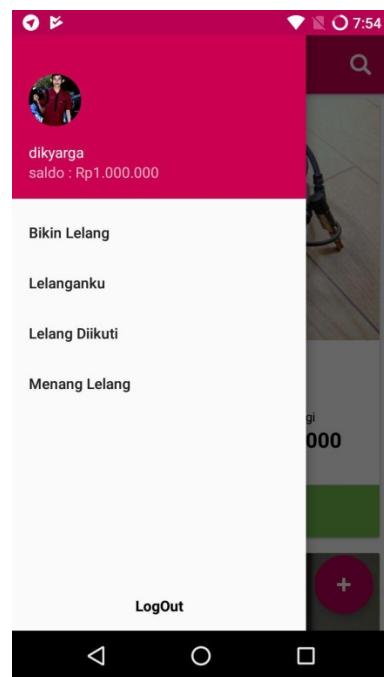
#### 4.5.4 Implementasi Halaman Riwayat Lelang Android

Tangkapan layar daftar riwayat lelang



Gambar 4. 22: Antarmuka Daftar Riwayat Lelang

#### 4.5.5 Implementasi Menu Drawer



Gambar 4. 23: Antarmuka Menu Drawer (Utama)

#### 4.5.6 Implementasi Halaman Buat Lelang

A screenshot of the "Bikin Lelang Baru" (Create New Auction) page. At the top, there are two photo upload boxes, a "UPLOAD PHOTO" button, and a "PUBLISH" button. Below these are fields for "JUDUL" (Title), "Kategori" (Category), and "Deskripsi" (Description). The "Deskripsi" field contains placeholder text "Tulis deskripsi barang". Underneath is a "Kondisi Barang" (Item Condition) section with radio buttons for "Baru" (New) and "Bekas" (Used), where "Bekas" is selected. There is also a "Berat (gr)" (Weight) input field.

Gambar 4. 24: Antarmuka Halaman Buat Lelang

#### 4.5.7 Implementasi Menu Opsi Penawaran



Gambar 4. 25: Antarmuka Menu Opsi Penawaran

### 4.6 Implementasi Web Service

Implementasi pada *web service API* dapat di lihat dari daftar route endpoint berikut.

#### 4.6.1 Integrasi Web Service dengan Android

Untuk memudahkan integrasi antara *web service (server)* dengan aplikasi Android (*client*), maka di butuhkan penjembatan yang menjadi patokan untuk kedua sisi. Maka dibuatlah dokumentasi API yang berisi informasi mengenai nama, *url*, deskripsi, parameter dan respon setiap *endpoint*.

Dokumentasi tersedia *online* dan dapat di melakukan *testing* di web tersebut juga [24].

Field	Type	Description
userId	Integer	userId of user
bukalapakId	Integer	bukalapakId of user
token	String	token of logged in user
title	String	Title of auction, note : Nama barang hanya boleh berupa huruf, angka, spasi dan simbol &.-
categoryId	Integer	category ID
new	Boolean	product is new or second ?

Gambar 4. 26: Dokumentasi API WebService

## 4.7 Pengujian Aplikasi

Pada penelitian ini, penulis menggunakan metode pengembangan Test Driven Development (TDD) sehingga proses pengujian dilakukan secara berulang-ulang dan beriterasi selama proses pengembangan.

Berikut adalah salah satu contoh *test case* yang dibuat sebelum proses pembuatan aplikasi, meliputi pengujian fitur dan *unit test* dengan menggunakan *library* Mocha sebagai *test runner* dan Chai sebagai *assertion*.

```

31  describe('Auction Test', () => {
32    describe('Create Auction', () => {
33      it('Should be return status Success when trying to create new Auction', (done) => {
34        imageUploader.uploadToBukaLapak(
35          process.env.BUKALAPAK_ID,
36          process.env.BUKALAPAK_TOKEN,
37          imagesName).then((responseAfterUpload) => {
38            // console.log('responseAfterUpload : ', responseAfterUpload.id);
39            chai.request(serverHost).post('/auctions').send({
40              userId: '1',
41              bukalapakId: process.env.BUKALAPAK_ID,
42              token: process.env.BUKALAPAK_TOKEN,
43              title: 'Ini cuma testing buat Lelang ' + randomName,
44              categoryId: 2,
45              new: false,
46              weight: 1000,
47              description: randomName + 'Ini cuma contoh deskripsi minimal 30 karakter lho',
48              min_price: min_price,
49              max_price: max_price,
50              kelipatan_bid: 10000,
51              imagesId: responseAfterUpload,
52              end_date: end_date
53            }).end((err, res) => {
54              if (err) {
55                done(err)
56              } else {
57                // console.log('isi res : ', res.body);
58                res.should.have.status(200);
59                res.should.be.json;
60                res.body.success.should.to.equal(true)
61                res.body.should.have.property('id')
62                res.body.should.have.property('category')
63                res.body.should.have.property('categoryId')
64                res.body.should.have.property('title')
65                res.body.should.have.property('slug')
66              }
67            })
68          })
69        })
70      })
71    })
72  })
73})

```

Gambar 4. 27: Contoh Unit Test Case Membuat Lelang Baru

Pengujian kecepatan bertukar data dengan protokol WebSocket pada realtime bidding pada aplikasi BukaLelang :

Tabel 4. 1: Tabel Pengujian Kecepatan Pertukaran Data melalui WebSocket

**Express.js + Socket.IO**

**Uji coba kecepatan kirim dan terima data melalui WebSocket pada aplikasi Bukalelang**

- Rata-rata : 72.6 milisecond

Percobaan	Send	Received	Latency
1	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	23 ms
2	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	33 ms
3	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	40 ms
4	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	42 ms
5	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	52 ms
6	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	54 ms
7	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	62 ms
8	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	64 ms
9	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	68 ms
10	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	75 ms
11	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	88 ms
12	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	89 ms
13	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	91 ms
14	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	92 ms
15	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	94 ms
16	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	95 ms
17	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	96 ms
18	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	97 ms
19	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	98 ms
20	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	Mon Aug 14 2017 22:40:54 GMT+0700 (WIB)	99 ms
Rata-rata			72.6 ms

Dari dua puluh kali pengujian dengan waktu yang bersamaan pada tabel tersebut, rata-rata kecepatan dari waktu kirim data hingga diterima kembali, kemudian dihitung perbedaan nya. Maka, diperoleh hasil rata-rata 72.6 milidetik.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan penelitian ini, maka dapat di simpulkan bahwa aplikasi lelang *online* ini dapat digunakan secara *realtime* dengan protokol baru websocket. Sehingga proses lelang dapat dilakukan dimanapun dan kapanpun, sehingga antar peserta lelang maupun pelelang merasa sedang berada pada waktu dan tempat yang sama.

Sebagai protokol baru websocket berhasil memecahkan masalah-masalah yang terjadi pada komunikasi *client-server realtime*.

Metode pengembangan aplikasi menggunakan Test Driven Development juga berhasil menjadikan aplikasi yang dibuat lebih *testable* atau teruji, sehingga mengurangi kemungkinan munculnya kesalahan sistem saat pengembangan maupun ketika sudah di *deploy*.

#### **5.2 Saran**

Sistem ini tentu tidak lepas dari kekurangan dan kelemahan. Oleh karena itu, demi sistem lelang *online* secara *realtime* ini agar lebih baik penulis menyarankan beberapa hal, diantaranya :

1. Perlu di buat sistem lelang *online* dalam versi *website*, sehingga pengguna dapat menggunakan sistem di berbagai perangkat, seperti PC atau laptop, tanpa harus memasangnya terlebih dahulu.
2. Perlu di tambah fitur diskusi, agar peserta lelang dapat menanyakan lebih detail tentang barang yang di lelang.

## DAFTAR PUSTAKA

- [1] "Peraturan Lelang - Peraturan Penjualan di Muka Umum di Indonesia." Kementrian Keuangan Republik Indonesia, 1908.
- [2] V. Pimentel and B. G. Nickerson, "Communicating and Displaying Real-Time Data with WebSocket," *IEEE Internet Comput.*, vol. 16, no. 4, pp. 45–53, Aug. 2012.
- [3] F. y. Jiang and H. c. Duan, "Application research of WebSocket technology on Web tree component," in *2012 International Symposium on Information Technologies in Medicine and Education*, 2012, vol. 2, pp. 889–892.
- [4] T. Anusas-amornkul and C. Silawong, "The study of compression algorithms for WebSocket protocol," in *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2014, pp. 1–6.
- [5] J. Park, H. Hwang, J. Yun, and I. Moon, "Study of HTML5 WebSocket for a Multimedia Communication," *Int. J. Multimed. Ubiquitous Eng.*, vol. 9, no. 7, pp. 61–72, 2014.
- [6] D. Skvorc, M. Horvat, and S. Srbljic, "Performance evaluation of Websocket protocol for implementation of full-duplex web streams," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, 2014, pp. 1003–1008.
- [7] "Online auction," *Wikipedia*. 28-Jan-2017.
- [8] "English auction," *Wikipedia*. 07-Aug-2017.
- [9] "About HTML5 WebSocket - Powered by Kaazing." [Online]. Available: <https://www.websocket.org/aboutwebsocket.html>. [Accessed: 08-Aug-2017].
- [10] K. Tsonev, *Node.js Blueprints*. Packt Publishing Limited, 2014.
- [11] S. Riggs and H. Krosing, *PostgreSQL 9 Admin Cookbook*. Packt Publishing Limited, 2010.
- [12] E. Sutanta and K. Mustofa, "Kebutuhan Web Service Untuk Sinkronisasi Data Antar Sistem Informasi Dalam E-Gov Di Pemkab Bantul Yogyakarta," *J. Teknol. Inf. Komun. JURTIK*, vol. 1, no. 1, 2012.
- [13] "JSON." [Online]. Available: <http://json.org/>. [Accessed: 08-Aug-2017].
- [14] N. js Foundation, "Node.js," *Node.js*. [Online]. Available: <https://nodejs.org/en/>.
- [15] "npm." [Online]. Available: <https://www.npmjs.com/>.
- [16] "State of the Union: npm," *Linux.com | The source for Linux information*, 13-Jan-2017. [Online]. Available: <https://www.linux.com/news/event/Nodejs/2016/state-union-npm>.
- [17] "Express - Node.js web application framework." [Online]. Available: <https://expressjs.com/>. [Accessed: 08-Aug-2017].
- [18] D. Nahum, *Programming React Native*. Leanpub, 2016.
- [19] "About AWS," *Amazon Web Services, Inc.* [Online]. Available: [//aws.amazon.com/about-aws/](http://aws.amazon.com/about-aws/).
- [20] R. Latorre, "Effects of Developer Experience on Learning and Applying Unit Test-Driven Development," *IEEE Trans. Softw. Eng.*, vol. 40, no. 4, pp. 381–395, Apr. 2014.

- [21] “Mocha - the fun, simple, flexible JavaScript test framework.” [Online]. Available: <https://mochajs.org/>. [Accessed: 08-Aug-2017].
- [22] “Chai.” [Online]. Available: <http://chaijs.com/>. [Accessed: 08-Aug-2017].
- [23] Hacktiv8, “Our Team - Hacktiv8,” *Hacktiv8*. [Online]. Available: <https://hacktiv8.com/team/>.
- [24] Diky Arga, “apiDoc: BukaLelang API Docs - 1.0.0,” *apiDoc: BukaLelang API Docs - 1.0.0*, 2017. [Online]. Available: <http://api.bukalelang.id/docs/>. [Accessed: 09-Aug-2017].