

**RANCANG BANGUN APLIKASI *MOBILE* JUAL BELI PRODUK
PERTANIAN MENGGUNAKAN PENDEKATAN *LOCATION*
BASED SERVICES (LBS)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Fakhry Ikhsan Firdaus

NIM: 115090600111028



PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

RANCANG BANGUN APLIKASI *MOBILE* JUAL BELI PRODUK PERTANIAN
MENGUNAKAN PENDEKATAN *LOCATION BASED SERVICES* (LBS)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Fakhry Ikhsan Firdaus
NIM: 115090600111028

Skripsi ini telah diuji dan dinyatakan lulus pada
20 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng Herman Tolle, ST., MT
NIP: 19740823 200012 1 001

Himawat Aryadita, ST., M.Sc.
NIP: 19801018 200801 1 003

Mengetahui
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Mardji, MT.
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 21 Januari 2016



Fakhry Ikhsan Firdaus

NIM: 115090600111028

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “**Rancang Bangun Aplikasi Mobile Jual Beli Produk Pertanian Menggunakan Pendekatan Location Based Services (LBS)**”.

Penyusunan skripsi ini juga tak lepas dari bantuan semua pihak yang telah memberikan semangat, doa, bimbingan kritik, serta saran. Maka dari itu penulis menyampaikan ucapan terimakasih kepada:

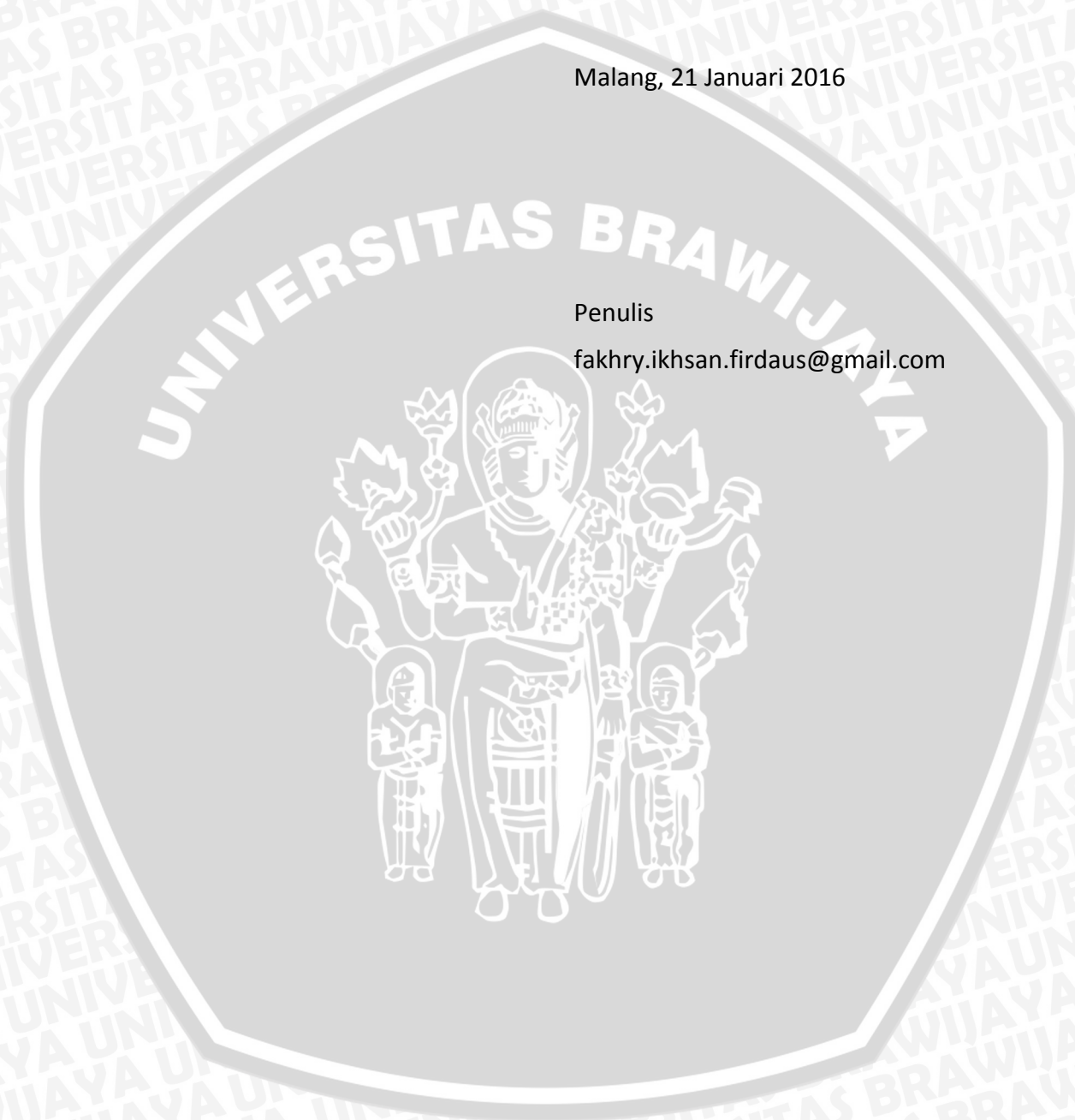
1. Bapak Dr.Eng. Herman Tolle, ST., MT selaku dosen pembimbing I yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Himawat Aryadita, ST., M.Sc. selaku dosen pembimbing II yang telah banyak memberikan ilmu dan saran untuk laporan skripsi ini.
3. Drs. Marji. M.T. dan Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Informatika/Illmu Komputer Universitas Brawijaya.
4. Seluruh dosen Informatika/Illmu Komputer Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
5. Kedua orang tua penulis, Bapak Sumadi dan Ibu Siti Kunaidah yang telah memberi motivasi, kasih sayang serta dukungan moril dan materil. Seluruh keluarga besar yang telah memberikan semangat dari awal sampai akhir pengerjaan skripsi ini.
6. Seluruh Civitas BPTIK dan teman-teman *Student Employee* yang telah memberikan saran dan motivasi kepada penulis.
7. Seluruh Civitas Akademik Informatika/Illmu Komputer Universitas Brawijaya yang telah banyak memberi dukungan dan bantuan selama penulis menempuh studi di Informatika/Illmu Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.
8. Seluruh teman-teman Ilkom 2011 khususnya Ilkom-B 2011 dan Tim CREVION; Singgih, Yusuf, Ami, yang selalu memberi dukungan, motivasi, kritik, dan saran.
9. Rivaldi Rizalul Akhsan dan Dwi Retno Palupi yang selalu memberi dukungan kepada penulis.
10. Teman Seperjuangan; Fauzi, Pipit, Irul, Darmawan, Singgih, Gigih, Luthfi, Budi, Ifan, Wisnu, Tiwi,.
11. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Dengan segala kerendahan hati, penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Akhirnya semoga skripsi ini dapat bermanfaat dan berguna bagi pembaca terutama mahasiswa Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

Malang, 21 Januari 2016

Penulis

fakhry.ikhsan.firdaus@gmail.com



ABSTRAK

Indonesia merupakan negara yang sebagian besar penduduknya bermata pencaharian sebagai petani. Penduduk di Indonesia mayoritas tinggal di daerah pedesaan, dimana sebagian besar masih menggantungkan hidupnya pada sektor pertanian. Namun masih banyak petani yang hidup dibawah garis kemiskinan. Hal ini menyebabkan penurunan jumlah petani, banyak petani beralih ke pekerjaan lain yang lebih menjanjikan. Penurunan jumlah ini disebabkan karena kurangnya kesejahteraan petani. Rendahnya kesejahteraan petani ini disebabkan oleh banyak faktor, diantaranya petani kesulitan untuk memasarkan produk pertanian secara lebih luas.

Location Based Services (LBS) atau layanan berbasis lokasi merupakan sebuah layanan informasi yang dapat diakses dengan perangkat bergerak melalui jaringan dan mampu menampilkan posisi secara geografis keberadaan perangkat bergerak tersebut. Dengan memanfaatkan teknologi LBS, akses petani untuk memasarkan produk pertaniannya secara lebih luas menjadi lebih mudah. Selain itu, dengan teknologi LBS para pembeli juga dapat mencari bahan pangan yang telah dipasarkan oleh petani sesuai dengan lokasi petani. Aplikasi ini dikembangkan pada perangkat bergerak untuk lebih mudah digunakan.

Pengujian yang dilakukan ialah pengujian validasi dan *usability*. Berdasarkan pengujian validasi, keseluruhan kebutuhan fungsional sistem rekomendasi dapat berjalan dengan baik dan sesuai dengan kebutuhan yang telah dirancang. Sedangkan berdasarkan pengujian *usability* diperoleh tingkat *usability* dari aplikasi berdasarkan pengujian *usability* yaitu *usefulness*, *ease of use*, *ease of learning*, dan *satisfaction* menunjukkan rata-rata sebesar 87,6 %, sehingga dapat disimpulkan bahwa aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) dapat dengan mudah digunakan.

Kata Kunci: Pertanian, Jual Beli, Perangkat bergerak, *Location Based Services*

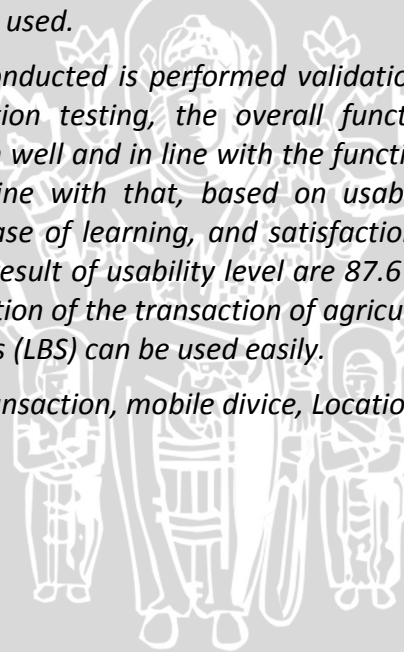
ABSTRACT

Indonesia is a country that the most of the population are farmers. In Indonesia, the most of population are living in rural areas, which are dependent on the agricultural sector. But there are still many farmers who live under the poverty line. It causes a decrease in the number of farmers. Therefore many farmers move to another job. This number of decrease is due to lack of farmers' welfare. The low welfare of farmers is caused by many factors, such as farmers cannot sell their product widely.

Location Based Services (LBS) is an information service that can be accessed by mobile devices through the network and it is capable to display the geographical position from the mobile device. By utilizing LBS technology, farmers are able to sell their agricultural products easier. In addition, the LBS technology is able to help buyers to find out the agricultural products that have been marketed by the farmers according to the location of the farmers. This application is developed on mobile devices, so it is easier to be used.

The testing which is conducted is performed validation testing and usability testing. Based on validation testing, the overall functional requirements of transaction system can run well and in line with the functional requirements that have been designed. In line with that, based on usability testing which are usefulness, ease of use, ease of learning, and satisfaction, the level of usability showed that the average result of usability level are 87.6 %. Therefore, it can be concluded that the application of the transaction of agricultural products by using the Location Based Services (LBS) can be used easily.

Key words: agricultural. transaction, mobile device, Location Based Services



DAFTAR ISI

RANCANG BANGUN APLIKASI <i>MOBILE</i> JUAL BELI PRODUK PERTANIAN MENGUNAKAN PENDEKATAN <i>LOCATION BASED SERVICES (LBS)</i>	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR PERSAMAAN.....	xv
DAFTAR <i>SOURCE CODE</i>	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 TINJAUAN PUSTAKA.....	5
2.1 Kajian pustaka	5
2.2 Landasan teori	6
2.2.1 Pertanian.....	6
2.2.2 Sistem informasi.....	6
2.2.3 <i>Location Based Services (LBS)</i>	7
2.2.4 <i>Web Service</i>	7
2.2.5 <i>Javascript Object Notation (JSON)</i>	8
2.2.5.1 Tipe data JSON	9
2.2.5.2 Contoh data JSON	9
2.2.6 <i>Hybrid Application</i>	10

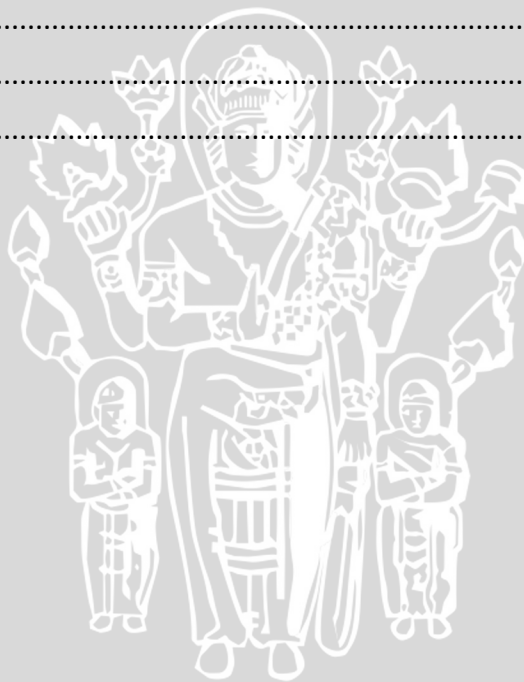
2.2.7 Apache Cordova	11
2.2.8 AngularJS.....	11
2.2.9 Ionic Framework.....	12
2.2.10 NgCordova.....	12
2.2.11 Google Maps API.....	13
2.2.12 UML.....	13
2.2.12.1 Use Case Diagram	14
2.2.12.2 Activity Diagram	14
2.2.12.3 Class Diagram	14
2.2.12.4 Sequence Diagram	14
2.2.13 Pengujian perangkat lunak.....	15
2.2.13.1 Pengujian validasi.....	15
2.2.13.2 Pengujian usability.....	15
BAB 3 METODOLOGI PENELITIAN	18
3.1 Studi literatur	18
3.2 Pengumpulan data.....	19
3.3 Analisis kebutuhan sistem	19
3.4 Perancangan aplikasi	20
3.5 Implementasi sistem.....	20
3.6 Pengujian dan analisis.....	20
3.7 Kesimpulan dan saran.....	20
BAB 4 PERANCANGAN.....	21
4.1 Analisis kebutuhan.....	21
4.1.1 Gambaran umum sistem.....	22
4.1.1.1 Deskripsi sistem	22
4.1.1.2 Lingkungan sistem.....	22
4.1.1.3 Bisnis proses sistem	23
4.1.2 Identifikasi aktor	24
4.1.3 Analisis kebutuhan fungsional	24
4.1.3.1 Use Case diagram.....	27
4.1.3.2 Skenario Use Case	27
4.1.4 Analisis kebutuhan non fungsional.....	32



4.2 Perancangan sistem	33
4.2.1 Perancangan UML	33
4.2.1.1 Perancangan Activity Diagram	33
4.2.1.2 Perancangan Class Diagram	37
4.2.1.3 Perancangan Sequence Diagram	37
4.2.2 Perancangan arsitektur sistem	40
4.2.3 Perancangan basis data	41
4.2.4 Perancangan antarmuka	44
4.2.5 Perancangan komunikasi data	48
BAB 5 IMPLEMENTASI	49
5.1 Spesifikasi sistem	49
5.1.1 Spesifikasi perangkat keras	49
5.1.2 Spesifikasi perangkat lunak	50
5.2 Batasan implementasi	50
5.3 Implementasi <i>class</i> dan <i>interface</i> pada file program	51
5.4 Implementasi kode program	51
5.4.1 Implementasi kode program untuk proses jual produk pertanian	52
5.4.2 Implementasi kode program untuk melihat produk dijual (<i>map</i>)	54
5.4.3 Implementasi kode program untuk mengajukan penawaran	57
5.4.4 Implementasi kode program untuk memilih penawar	58
5.4.5 Implementasi kode program untuk melaporkan pengguna	59
5.5 Implementasi Antarmuka Pengguna	60
5.5.1 Implementasi Halaman <i>Splash Screen</i>	60
5.5.2 Implementasi halaman jual produk pertanian	61
5.5.3 Implementasi halaman melihat produk dijual (<i>map</i>)	62
5.5.4 Implementasi halaman mengajukan penawaran	62
5.5.5 Implementasi antarmuka memilih penawar	63
5.5.6 Implementasi halaman melaporkan pengguna	63
BAB 6 PENGUJIAN DAN ANALISIS	64
6.1 Pengujian	64



6.1.1 Pengujian validasi.....	64
6.1.1.1 Kasus uji menjual produk pertanian	64
6.1.1.2 Kasus uji melihat produk dijual (map)	65
6.1.1.3 Kasus uji mengajukan penawaran	66
6.1.1.4 Kasus uji memilih penawar	68
6.1.1.5 Kasus uji melaporkan pengguna	70
6.1.2 Pengujian <i>usability</i>	71
6.2 Analisis	73
6.2.1 Analisis hasil pengujian validasi	73
6.2.2 Analisis hasil pengujian <i>usability</i>	74
BAB 7 PENUTUP	78
7.1 Kesimpulan.....	78
7.2 Saran	78
DAFTAR PUSTAKA.....	79



DAFTAR TABEL

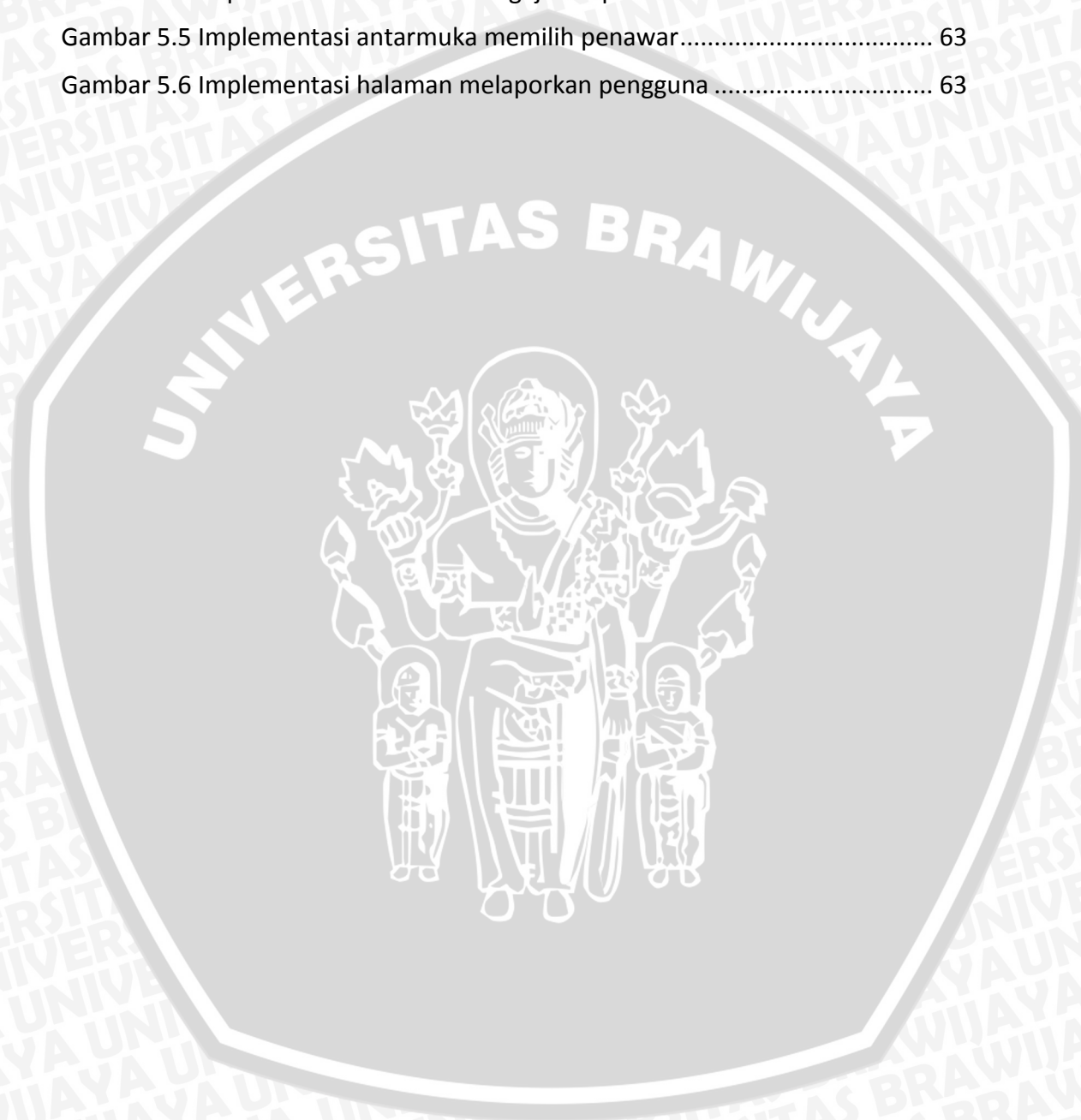
Tabel 2.1 Penilaian jawaban kuantitatif.....	17
Tabel 4.1 Identifikasi aktor.....	24
Tabel 4.2 Daftar kebutuhan fungsional.....	24
Tabel 4.3 Skenario <i>Use Case</i> jual produk.....	29
Tabel 4.4 Skenario <i>Use Case</i> lihat produk dijual (map)	29
Tabel 4.5 Skenario <i>Use Case</i> ajukan penawaran.....	30
Tabel 4.6 Skenario <i>Use Case</i> memilih penawar	31
Tabel 4.7 Skenario <i>Use Case</i> laporkan pengguna	32
Tabel 4.8 Detail atribut tiap basis data	42
Tabel 4.14 Perancangan komunikasi data	48
Tabel 5.1 Spesifikasi perangkat keras komputer	49
Tabel 5.2 Spesifikasi perangkat keras <i>smartphone</i> Android.....	49
Tabel 5.3 Spesifikasi perangkat lunak komputer	50
Tabel 5.4 Spesifikasi perangkat lunak <i>smartphone</i> Android.....	50
Tabel 6.1 Kasus uji menjual produk pertanian.....	64
Tabel 6.2 Kasus uji melihat produk dijual	66
Tabel 6.3 Kasus uji mengajukan penawaran.....	67
Tabel 6.4 Kasus uji memilih penawar.....	68
Tabel 6.5 Kasus uji melaporkan pengguna.....	70
Tabel 6.6 Hasil pengujian <i>usability</i>	71
Tabel 6.7 Interpretasi skor Likert	74
Tabel 6.8 Hasil perhitungan pengujian <i>usability</i>	74
Tabel 6.8 Hasil status pengujian <i>usability</i>	77

DAFTAR GAMBAR

Gambar 2.1 Contoh Data JSON	10
Gambar 2.2 Logo AngularJS	12
Gambar 2.3 Logo <i>Ionic Framework</i>	12
Gambar 2.4 Logo <i>NgCordova</i>	13
Gambar 3.1 Diagram Blok Metodologi Penelitian	18
Gambar 4.1 Diagram perancangan	21
Gambar 4.2 Bisnis proses sistem	23
Gambar 4.3 Diagram <i>Use Case</i>	28
Gambar 4.4 <i>Activity Diagram</i> proses menjual produk pertanian.....	34
Gambar 4.5 <i>Activity Diagram</i> proses melihat produk yang dijual oleh pengguna lain	35
Gambar 4.6 <i>Activity Diagram</i> proses mengajukan penawaran	35
Gambar 4.7 <i>Activity Diagram</i> proses memilih penawar	36
Gambar 4.8 <i>Activity Diagram</i> proses melaporkan pengguna	36
Gambar 4.9 Diagram <i>Class</i> Sistem	37
Gambar 4.10 <i>Sequence Diagram</i> pada skenario jual produk pertanian.....	38
Gambar 4.11 <i>Sequence Diagram</i> pada skenario melihat produk yang dijual oleh pengguna lain.....	39
Gambar 4.12 <i>Sequence Diagram</i> pada skenario mengajukan penawaran.....	39
Gambar 4.13 <i>Sequence Diagram</i> pada skenario memilih penawar	39
Gambar 4.14 <i>Sequence Diagram</i> pada skenario melaporkan pengguna	40
Gambar 4.15 Arsitektur aplikasi jual beli produk pertanian menggunakan LBS ..	41
Gambar 4.16 ERD sistem.....	42
Gambar 4.17 <i>Screen Flow</i> Sistem <i>Client</i>	44
Gambar 4.18 Antarmuka halaman <i>Splash Screen</i>	45
Gambar 4.19 Antarmuka halaman jual produk pertanian.....	45
Gambar 4.20 Antarmuka halaman lihat produk dijual (<i>map</i>).....	46
Gambar 4.21 Antarmuka halaman mengajukan penawaran.....	46
Gambar 4.22 Antarmuka halaman memilih penawar	47
Gambar 4.23 Antarmuka halaman melaporkan pengguna	47

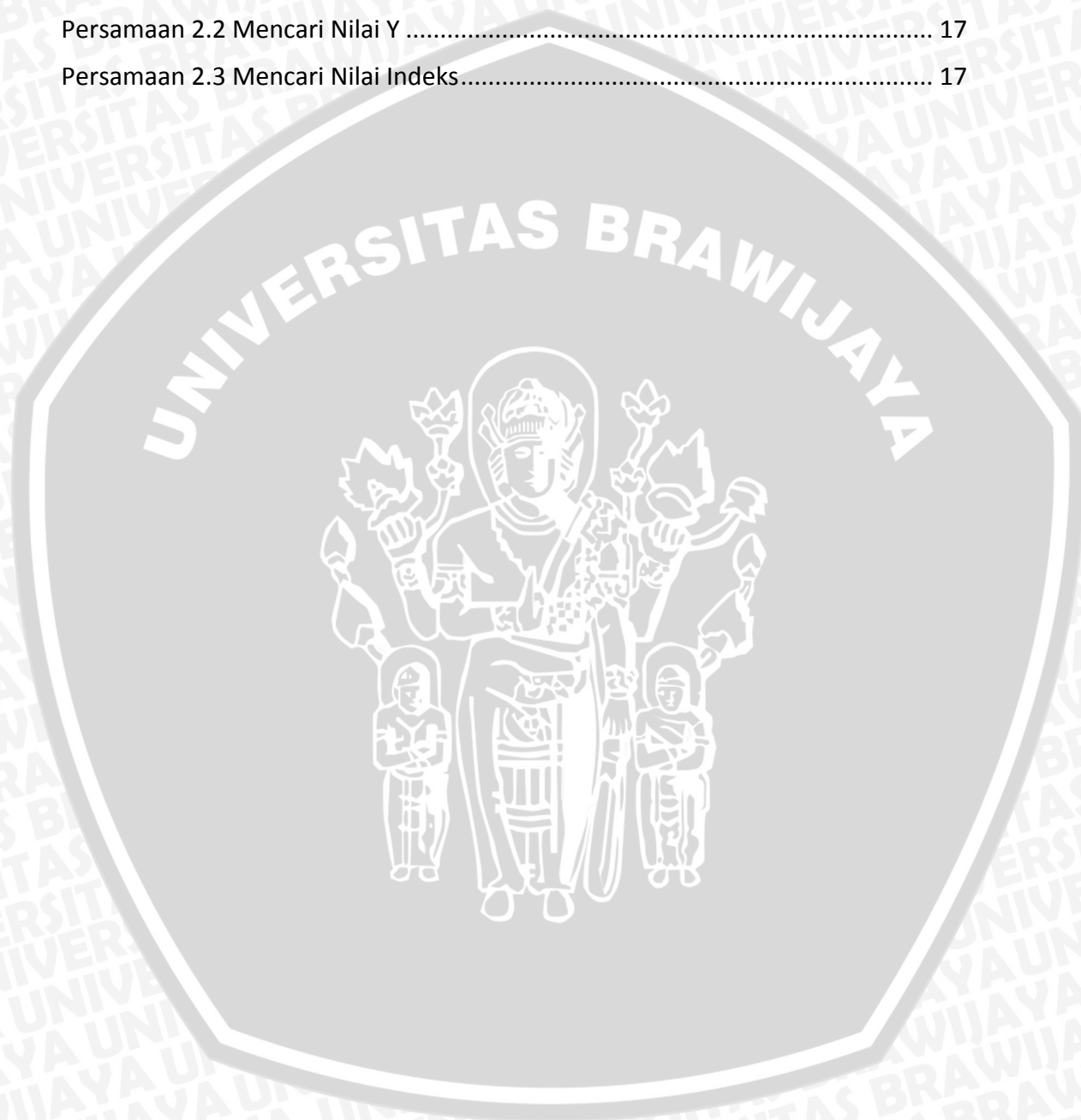


Gambar 5.1 Implementasi halaman <i>Splash Screen</i>	61
Gambar 5.2 Implementasi halaman jual produk dijual	61
Gambar 5.3 Implementasi halaman melihat produk dijual (<i>map</i>)	62
Gambar 5.4 Implementasi halaman mengajukan penawaran.....	62
Gambar 5.5 Implementasi antarmuka memilih penawar.....	63
Gambar 5.6 Implementasi halaman melaporkan pengguna	63



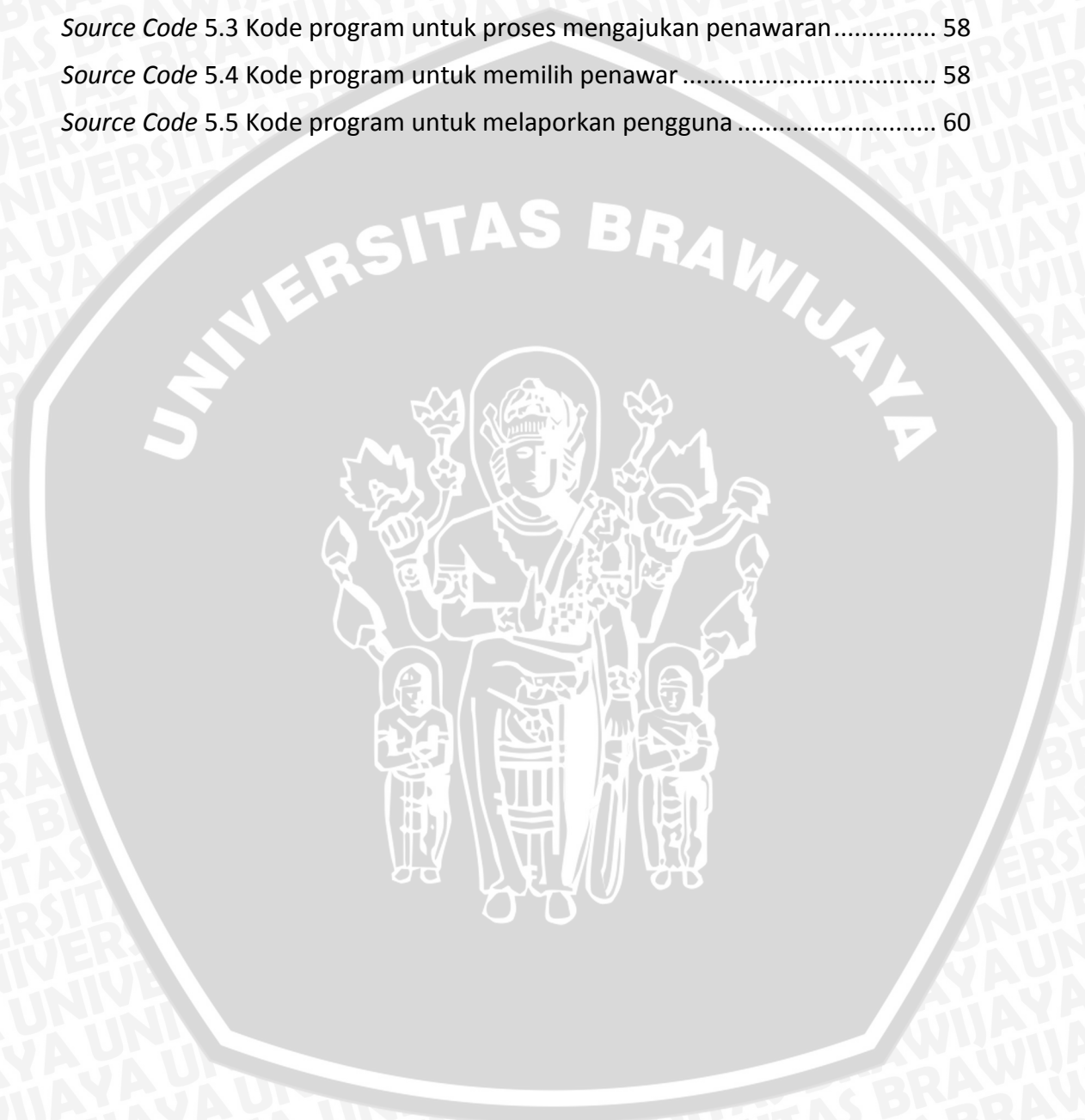
DAFTAR PERSAMAAN

Persamaan 2.1 Mencari Nilai TotalSkor.....	17
Persamaan 2.2 Mencari Nilai Y	17
Persamaan 2.3 Mencari Nilai Indeks.....	17



DAFTAR SOURCE CODE

Source Code 5.1 Kode program untuk proses jual produk pertanian.....	53
Source Code 5.2 Kode program untuk melihat produk dijual (<i>map</i>).....	57
Source Code 5.3 Kode program untuk proses mengajukan penawaran.....	58
Source Code 5.4 Kode program untuk memilih penawar.....	58
Source Code 5.5 Kode program untuk melaporkan pengguna.....	60



BAB 1 PENDAHULUAN

Bab ini membahas mengenai latar belakang dilakukannya penelitian beserta rumusan masalah, tujuan, manfaat, batasan masalah sistematika penulisan. Latar belakang penelitian didapat dari masalah yang ada dan solusi yang ditawarkan. Dari latar belakang tersebut dapat dirumuskan masalah yang akan menjadi hasil kesimpulan penelitian ini.

1.1 Latar belakang

Indonesia merupakan negara yang sebagian besar penduduknya bermata pencaharian sebagai petani. Kurang lebih 60 persen penduduk di Indonesia tinggal di daerah pedesaan, dimana sebagian besar masih menggantungkan hidupnya pada sektor pertanian [BPS - 02]. Wilayah Indonesia yang di anugerahi sumberdaya yang melimpah dan tanah subur yang sangat mendukung masyarakatnya untuk mengembangkan usaha di bidang pertanian. Sektor pertanian juga memberikan peranan besar dalam kehidupan khususnya masyarakat Indonesia [KPT - 11], mayoritas hasil pangan berasal dari sektor pertanian, untuk menjaga agar pertanian selalu menghasilkan produk yang baik maka pengelolaannya pun harus dengan cara yang baik pula. Keberhasilan dalam pengelolaan ini tidak lepas dari peran petani. Menurut data Sensus Pertanian tahun 2013 (ST2013) yang dilakukan oleh Badan Pusat Statistik (BPS), jumlah rumah tangga usaha pertanian di Indonesia tercatat sebanyak 26,14 juta rumah tangga, menurun sebesar 16,32 persen dari hasil Sensus Pertanian 2003 (ST2003) yang tercatat sebanyak 31,23 juta rumah tangga [BPS - 02].

Penurunan jumlah ini disebabkan karena kurangnya kesejahteraan petani. Rendahnya kesejahteraan petani ini disebabkan oleh banyak faktor, diantaranya rendahnya nilai tambah produk yang dinikmati oleh petani. banyak petani beralih ke pekerjaan lain yang lebih menjanjikan. Menurut Badan Pusat Statistik sebagian besar dari para pekerja di sektor pertanian hidup dibawah garis kemiskinan [BPS – 04]. Salah satu penyebabnya adalah petani tidak dapat menjual produk pertaniannya dengan harga yang tinggi, petani kesulitan untuk menjual produk pertaniannya secara langsung ke konsumen, petani lebih banyak menjual produk pertanian kepada tengkulak yang biasanya membeli hasil pertanian dengan harga yang murah untuk dijual kembali dengan harga yang tinggi. Hal ini mengakibatkan pendapatan petani menjadi rendah dan jauh dari kata sejahtera.

Banyak masyarakat di dunia secara langsung atau tidak langsung bergantung pada sektor pertanian untuk mata pencaharian utama. Selain membangun kemampuan produksi petani, meningkatkan akses petani ke pasar merupakan elemen kunci dalam strategi untuk mempromosikan pembangunan pedesaan dan peningkatan taraf hidup petani. [ELI – 11].

Sesuai Pedoman Reformasi Perencanaan dan Penganggaran, maka sebagai salah satu unit kerja Eselon I di Kementerian Pertanian, Direktorat Jenderal

Pengolahan dan Pemasaran Hasil Pertanian memiliki satu program yang mendukung Kementerian Pertanian dalam pencapaian tujuan dan sasaran yang telah ditetapkan, yaitu “Program Peningkatan Nilai Tambah, Daya Saing, Industri Hilir, Pemasaran dan Ekspor Hasil Pertanian”. Program tersebut dijabarkan dalam kegiatan – kegiatan yang sesuai dengan tugas fungsi Eselon II di dalamnya meliputi kegiatan: (1) Pengembangan Usaha dan Investasi, (2) Pengembangan Pengolahan Hasil Pertanian, (3) Pengembangan Mutu dan Standardisasi Pertanian, (4) Pengembangan Pemasaran Domestik, (5) Pengembangan Pemasaran Internasional, (6) Dukungan Manajemen dan Teknis lainnya pada Direktorat Jenderal Pengolahan dan Pemasaran Hasil Pertanian [KPT – 11].

Location Based Services (LBS) atau layanan berbasis lokasi merupakan sebuah layanan informasi yang dapat diakses dengan perangkat bergerak melalui jaringan dan mampu menampilkan posisi secara geografis keberadaan perangkat bergerak tersebut, teknologi ini memungkinkan pengguna dapat memperoleh informasi lokasi sesuai dengan kebutuhan [ANW-14]. Dengan memanfaatkan teknologi LBS, akses petani untuk memasarkan produk pertaniannya secara lebih luas menjadi lebih mudah. Selain itu, dengan teknologi LBS para pembeli juga dapat mencari bahan pangan yang telah dipasarkan oleh petani sesuai dengan lokasi petani.

Berdasarkan latar belakang tersebut, penulis mengangkat judul skripsi “Rancang Bangun Aplikasi *Mobile* Jual Beli Produk Pertanian Menggunakan Pendekatan *Location Based Services* (LBS)”. Aplikasi ini bertujuan untuk membantu petani untuk memasarkan produk pertanian secara lebih luas dalam menjangkau konsumen secara langsung, Aplikasi ini nantinya juga dapat memudahkan pembeli dalam mencari hasil pertanian sesuai dengan kebutuhan dan lokasi terdekat.

1.2 Rumusan masalah

Berdasarkan uraian latar belakang, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana rancangan dan implementasi aplikasi *mobile* jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) yang dapat membantu petani untuk memasarkan produk pertanian secara lebih luas dalam menjangkau konsumen secara langsung serta memudahkan pembeli dalam mencari hasil pertanian.
2. Bagaimana tingkat *usability* aplikasi *mobile* jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS).

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah terciptanya sebuah aplikasi *mobile* yang dapat membantu mempermudah petani dalam proses pemasaran produk pertanian secara lebih luas dan secara langsung ke pembeli dengan menggunakan

pendekatan *Location Based Services* (LBS) serta memudahkan pembeli dalam mencari hasil pertanian.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat umum antara lain :

1. Aplikasi ini diharapkan dapat mempermudah proses jual beli antara petani dan konsumen.
2. Aplikasi ini diharapkan dapat mempermudah petani dalam memasarkan produk pertanian secara lebih luas.
3. Aplikasi ini diharapkan dapat mempermudah pembeli dalam mencari hasil pertanian yang dibutuhkan.

1.5 Batasan masalah

Agar permasalahan yang dirumuskan lebih terfokus dan tidak terjadi pelebaran topik, maka penelitian tugas akhir ini dibatasi dalam hal :

1. Data yang digunakan dalam pembangunan aplikasi ini adalah data dari Kabupaten Jember.
2. Implementasi aplikasi berbasis *Hybrid App* dengan menggunakan *Ionic Framework, AngularJS, NgCordova* dan *Apache Cordova*.
3. Sistem ini terdiri dari aplikasi *front end* yang berbasis *mobile* dan *back end* yang berbasis *website*, namun pembahasan kajian pada penelitian ini berfokus pada aplikasi *front end* yang berbasis *mobile*.
4. Aplikasi *front end* diimplementasikan pada perangkat bergerak berbasis Android.

1.6 Sistematika pembahasan

Sistematika penulisan dalam penyusunan skripsi ini adalah sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, serta sistematika penulisan.

BAB II Tinjauan Pustaka

Menguraikan tentang kajian pustaka serta teori penunjang yang mendasari proses perancangan dan implementasi pada penelitian ini.

BAB III Metodologi Penelitian

Membahas mengenai langkah-langkah kerja yang dilakukan dalam penelitian ini yang terdiri dari studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, serta pengujian dan analisis.

BAB IV Analisis dan Perancangan

Membahas analisis kebutuhan dan perancangan *user interface* untuk pengembangan perangkat lunak.

BAB V Implementasi

Membahas tentang implementasi *Location Based Services* (LBS) pada studi kasus aplikasi jual beli produk pertanian.

BAB VI Pengujian dan Analisis

Membahas hasil pengujian dan analisis terhadap sistem yang telah dibangun menggunakan teknologi *Location Based Services* (LBS).

BAB VII Penutup

Memuat kesimpulan yang diperoleh dari pembuatan sistem dan pengujian perangkat lunak yang dikembangkan dengan menggunakan teknologi *Location Based Services* (LBS) serta saran-saran untuk pengembangan lebih lanjut

BAB 2 TINJAUAN PUSTAKA

Bab tinjauan pustaka terdiri dari kajian pustaka dan dasar teori. Kajian pustaka membahas penelitian yang telah ada dan yang diusulkan. Dasar teori membahas landasan teori yang diperlukan untuk menyusun penelitian yang diusulkan. Dasar teori yang menjadi landasan dalam penelitian ini meliputi pengertian pertanian dan sistem informasi, *Location Based Services (LBS)*, *Web Service*, *Javascript Object Notation (JSON)*, *Hybrid Application*, *Apache Cordova*, *AngularJS*, *Ionic Framework*, *NgCordova*, Google Maps API, konsep dasar *Unified Modelling Language* yang dipakai pada rekayasa perangkat lunak, dan strategi pengujian perangkat lunak.

2.1 Kajian pustaka

Kajian pustaka yang akan dibahas adalah penelitian terdahulu dengan topik yang menjadi dasar dari penelitian yang diajukan penulis. Penelitian terdahulu yang berjudul “Rancang Bangun Sistem Informasi Jual Beli Bahan Pangan dan Pertanian Berbasis *Web*” dilakukan oleh Rendra Pranadipa pada tahun 2014, mahasiswa Universitas Brawijaya Malang. Penelitian tersebut membahas mengenai perancangan dan pembuatan aplikasi sistem informasi jual beli bahan pangan dan pertanian berbasis web yang ditujukan untuk petani dalam memasarkan produk pertaniannya secara lebih luas dan pembeli dapat melihat produk – produk pertanian melalui *website* yang telah ada.

Aplikasi dibangun berbasis *web base responsive*. User diharuskan membuat akun terlebih dahulu agar dapat menggunakan fitur – fitur yang ada secara lengkap, seperti menjual bahan - bahan pangan dan pertanian. Ketika mendaftar menjadi member, user harus mempunyai nomor registrasi yang diberikan oleh admin, hal ini dimaksudkan agar user yang menggunakan aplikasi merupakan seorang petani asli. Nomor registrasi diberikan kepada user yang benar – benar memiliki syarat sebagai seorang petani.

Berdasarkan pengujian yang dilakukan menggunakan metode pengujian validasi dan *usability* didapatkan presentasi yang positif baik. Dari hasil pengujian validasi, fitur yang terdapat pada aplikasi dapat berjalan dengan baik sesuai perancangan awal dengan presentasi 100%. Dan untuk pengujian *usability* didapatkan hasil yang cukup memuaskan dimana semua pertanyaan yang diajukan memiliki nilai lebih dari 50% [REN-14].

Penelitian selanjutnya adalah penelitian dari Courtney McTavish dan Suresh Sankaranarayanan dengan judul “*Intelligent Agent Based Hotel Search & Booking System*” membahas tentang pembelian dan penjualan barang dan jasa melalui perangkat genggam nirkabel seperti telepon selular (*mobile commerce*). *M-commerce (mobile commerce)* menyediakan banyak layanan seperti *Mobile Ticketing*, *Mobile banking*, layanan *mobile* berbasis lokasi. Hal ini merupakan kesempatan yang luar biasa untuk memungkinkan perangkat *mobile*, sebagai perangkat universal untuk aplikasi *commerce*. Dalam studi kasus penelitian ini,

pengguna dapat memilih hotel terbaik di lokasi terdekat, dengan fasilitas modern, lingkungan yang bersih dan harga yang terjangkau. Hal ini dapat memakan waktu dan lebih mahal ketika melakukan pencarian sendiri maupun menggunakan tenaga orang lain. Sistem menggunakan *intelligent agent* (agen cerdas) untuk melakukan kegiatan serupa pencarian dan pemesanan yang dapat meningkatkan kecepatan pencarian dan mengurani biaya secara signifikan [COU-10].

Implementasi aplikasi *mobile* jual beli produk pertanian dengan menggunakan pendekatan *Location Based Services* (LBS) diharapkan dapat menghasilkan sebuah aplikasi *mobile* yang dapat membantu mempermudah petani dalam proses pemasaran produk pertanian secara lebih luas dan secara langsung ke konsumen, sehingga hasil atau keuntungan yang didapat oleh petani menjadi lebih banyak.

2.2 Landasan teori

Landasan teori yang digunakan dalam penelitian ini berasal dari buku yang mendukung dan jurnal – jurnal penelitian sebelumnya. Teori yang digunakan sebagai referensi tidak diambil secara keseluruhan dari jurnal. Hanya teori – teori yang dijadikan sebagai dasar penelitian inilah yang dikutip pada penelitian ini. Landasan – landasan teori yang diambil yaitu berkaitan dengan pertanian, sistem informasi, *Location Based Services* (LBS), *Web service*, *JSON*, *Hybrid Application*, *Apache Cordova*, *AngularJS*, *Ionic Framework*, *NgCordova*, *Google Maps API*, *UML* dan pengujian perangkat lunak.

2.2.1 Pertanian

Definisi Pertanian menurut Kamus Besar Bahasa Indonesia (KBBI) adalah usaha pembibitan, pengolahan, pemeliharaan, dsb (pada tanaman dan hewan). Sebagai negara agraris, proporsi terbesar penduduk Indonesia berada pada sektor pertanian. Menurut BPS (2013) hasil sementara Sensus Pertanian 2013 menunjukkan jumlah rumah tangga pertanian mencapai 26.126,2 ribu rumah tangga atau 42,7 persen dari rumah tangga total nasional. Jumlah rumah tangga tani tahun 2013 tersebut mengalami penurunan sebesar 5.043,9 ribu rumah tangga dibandingkan kondisi tahun 2003. Sementara itu dari aspek ketenagakerjaan, pada tahun 2011 jumlah tenaga kerja di sektor pertanian berjumlah 39,3 juta jiwa atau 35,9 persen total tenaga kerja. Meskipun mengalami penurunan, jumlah rumah tangga tani dan jumlah serapan tenaga kerja sektor pertanian tersebut masih cukup dominan. Aktivitas sektor pertanian sebagian besar dilakukan di wilayah perdesaan dan didominasi oleh petani dengan kegiatan utama usahatani budidaya (*on farm*). Pada kondisi demikian maka perhatian pembangunan untuk peningkatan pendapatan petani menjadi sangat relevan dan strategis [MUC - 13].

2.2.2 Sistem informasi

Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat

untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pengguna [KAD – 03].

Sistem informasi merupakan suatu sistem yang saling berkaitan dan berintegrasi satu sama lain dan bertujuan menyediakan informasi untuk mendukung operasi, manajemen dan fungsi pengambilan keputusan dalam suatu organisasi.

Kegiatan di Sistem Informasi mencakup [KAD – 03] :

1. Input, menggambarkan suatu kegiatan untuk menyediakan data untuk diproses.
2. Proses, menggambarkan bagaimana suatu data di proses untuk menghasilkan suatu informasi yang bernilai tambah.
3. Output, suatu kegiatan untuk menghasilkan laporan dari proses diatas tersebut.
4. Penyimpanan, suatu kegiatan untuk memelihara dan menyimpan data.
5. Kontrol, ialah suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

2.2.3 Location Based Services (LBS)

Location Based Services (LBS) atau layanan berbasis lokasi adalah istilah umum yang digunakan untuk menentukan lokasi perangkat bergerak. LBS merupakan suatu layanan yang bereaksi aktif terhadap perubahan entitas posisi sehingga mampu mendeteksi letak objek dan memberikan layanan sesuai dengan letak objek yang telah diketahui tersebut [ANW-14]. LBS pada penelitian ini digunakan untuk memperlihatkan lokasi user saat ini dan juga menampilkan lokasi – lokasi produk pertanian yang dijual oleh petani melalui aplikasi ini. Terdapat tiga unsur utama LBS yaitu [ANW-14]:

1. *Location Manager* (API Maps), menyediakan perangkat bagi sumber atau *source* untuk LBS, *Application Programming Interface* (API) Maps menyediakan fasilitas untuk menampilkan atau memanipulasi peta.
2. *Location Providers* (API Location), menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat.
3. API Location berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. Lokasi, perpindahan, serta kedekatan dengan lokasi tertentu dapat ditentukan melalui *Location Manager*.

2.2.4 Web Service

Web service adalah aplikasi basis data (*database*), perangkat lunak (*software*) atau bagian dari perangkat lunak yang dapat diakses secara *remote* oleh berbagai perangkat dengan sebuah perantara tertentu. Secara umum, *web service* dapat diidentifikasi dengan menggunakan URL seperti *website* pada umumnya. Namun yang membedakan *web service* dengan *website* pada umumnya adalah interaksi yang diberikan oleh *web service*. Berbeda dengan URL *website*, URL *web service*

hanya mengandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berfungsi untuk membangun sebuah fungsi-fungsi tertentu aplikasi. *Web service* dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah *database* disimpan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di *platform* apa sebuah data tersebut dikonsumsi. *Web service* mampu menunjang interoperabilitas, sehingga mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada [GOD – 13].

Website pada umumnya digunakan untuk melakukan *request* dan *response* yang dilakukan antara *client* dengan *server*. Sebagai contoh, seorang pengguna layanan *web* tertentu mengetikkan alamat url *website* untuk membentuk sebuah *request*. Maka *request* akan sampai pada *server*, kemudian diolah dan disajikan dalam bentuk sebuah *response*. Dengan singkat kata terjadilah hubungan *client-server* secara sederhana. Sedangkan pada *web service* hubungan antara *client* dan *server* tidak terjadi secara langsung. Hubungan antara *client* dan *server* dijematani oleh data *web service* dalam format tertentu. Sehingga akses terhadap *database* tidak akan ditangani secara langsung oleh *server*, melainkan melalui perantara yang disebut *web service* [GOD – 13]. Pada penelitian ini menggunakan *web service* sebagai media pertukaran informasi data antara *server* dan *client*.

2.2.5 Javascript Object Notation (JSON)

Proses pertukaran data antara *server* dan *client* pada aplikasi jual beli produk pertanian menggunakan *Javascript Object Notation (JSON)* yang merupakan format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia. Bisa juga diartikan dengan format sederhana untuk melakukan pertukaran data antara *client* dan *server* [KSI-08]. Selain itu, data berformat JSON mudah diuraikan dan diolah oleh *client*. JSON merupakan bagian dari bahasa pemrograman *Javascript*, standar ECMA-262 Edisi ke-3 Desember 1999. JSON merupakan format teks berbahasa independen, tetapi menggunakan konversi yang akrab bagi *programmer* dari keluarga besar bahasa C, seperti C, C++, C#, Java, *Javascript*, Perl, Python dan sebagainya. Hal tersebut membuat JSON menjadi bahasa pertukaran data yang ideal [JSN-14].

JSON dibentuk dari 2 struktur, yaitu [KSI-08] :

1. Kumpulan pasangan nama/nilai. Dalam berbagai bahasa, hal ini direalisasikan sebagai sebuah objek, rekaman, struktur, kamus, tabel hash, daftar berkunci, atau array asosiatif.
2. Daftar nilai berurutan. Pada kebanyakan bahasa, hal ini direalisasikan sebagai sebuah array, vector atau urutan.

Struktur data JSON adalah struktur data universal. Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data tersebut dalam bentuk yang sama maupun berlainan. Hal tersebut dapat dikatakan demikian karena format data mudah dipertukarkan dengan bahasa pemrograman yang berdasarkan pada struktur data tersebut [JSN-14].

2.2.5.1 Tipe data JSON

Pada JSON, data dapat berupa string, angka, objek atau larik. Berikut ini merupakan tipe data JSON [KSI-08] :

- Number*, dapat berupa angka, integer, float.
- String*, dapat berupa teks atau karakter *unicode* dalam petik ganda dengan *Backslash escapes*.
- Boolean, tipe data boolean merepresentasikan nilai benar dan salah.
- Null*, nilai *null* merepresentasikan bahwa variabel tidak memiliki nilai.
- Object*, kumpulan padangan nilai yang dipisahkan dengan koma “,” dan dimulai dengan kurung kurawal buka “{”, serta diakhiri dengan kurung kurawal tutup “}”.
- Array*, urutan nilai yang berurutan yang dipisahkan dengan “,” dan dimulai dengan kurung siku buka “[”, serta diakhiri dengan kurung siku tutup “]”.

Pada bentuk inti, JSON merepresentasikan 4 tipe data primitif seperti string, angka, boolean, null, dan 2 tipe terstruktur, yaitu objek dan larik. Tipe data tersebut menggunakan kode escape *Javascript* standar dan menambahkan sebuah *backslash* “\” sebelum menuliskan karakter berikut [KSI-08]:

- “ (tanda petik)
- B (*backspace*)
- n (baris baru)
- f (*form feed*)
- r (*carriage return*)
- t (tab horisontal)
- u (ditambah 4 digit untuk sebuah karakter *Unicode*)
- \ (*backslash*)
- / (*forward slash*)

2.2.5.2 Contoh data JSON

Contoh data berformat JSON yang akan dijelaskan dalam Gambar 2.1

```
{
  "Details":
  {
    "FirstName": "Suchita",
    "LastName": "Jain",
    "Address":
```



```
{
  "StreetAddress": "43 4th Street",
  "Country": "India",
  "State": "New Delhi",
  "PostalCode": 110002
},
"PhoneNumber": 0112345689
}
```

Gambar 2.1 Contoh Data JSON

Sumber: [KSI-08]

Gambar 2.1 menjelaskan bahwa Details adalah objek tertinggi dan semua data lainnya adalah anggota dari objek tersebut. FirstName, LastName, dan Address adalah Kolom string dan PostalCode dan PhoneNumber adalah kolom angka. Pasangan nama dan nilai dipisah dengan koma, misalnya "FirstName": "Suchita" adalah salah satu pasangan nama/ nilai dan pasangan nama/nilai lainnya yaitu, "LastName": "Jain", maka koma akan memisahkan 2 pasangan tersebut. Selain itu, titik dua ":" digunakan antara nama dan nilai [KSI-08].

2.2.6 Hybrid Application

Teknik pengembangan aplikasi jual beli produk pertanian ini menggunakan teknik *hybrid*. Aplikasi *hybrid* dibangun dengan menggunakan bahasa HTML, CSS, dan Javascript. Aplikasi *hybrid* memanfaatkan WebKit untuk membuat konten web dan proses kode javascript. Aplikasi *Hybrid*, seperti aplikasi *native* pada umumnya, berjalan dalam lingkungan *native process*. Aplikasi *hybrid* juga memanfaatkan *bridge layer* yang dapat memungkinkan javascript untuk mengakses banyak kemampuan spesifik perangkat dan native API yang umumnya tidak dapat diakses dari web browser [GOK-13].

Keuntungan menggunakan aplikasi *hybrid* dibanding *native* sebagai berikut [GOK-13]:

1. *Cost of maintenance*

Biaya pemeliharaan biasanya lebih rendah jika dibanding dengan *native app* karena *developer* tidak perlu menulis ulang semua kode aplikasi ke bahasa native pada masing – masing device target.

2. *Hybrid app* merupakan metode yang lebih banyak digunakan di masa yang akan datang.

Melihat ke arah kemajuan masa dengan dan yang akan datang dalam teknologi OS mobile, orang dapat dengan mudah mengatakan bahwa *hybrid app* adalah langkah pengembangan yang lebih banyak digunakan di masa depan, dengan banyaknya OS smartpone seperti Windows Phone, IOS, google

mengumumkan untuk akhirnya menggabungkan Chromium OS dan Android, Tizen OS, dan Firefox OS.

3. Akses ke kemampuan bawaan perangkat.

Aplikasi *hybrid* memiliki perbedaan dengan aplikasi berbasis web, yaitu dapat mengakses API asli dan kemampuan bawaan platform yang tidak tersedia jika melalui lingkungan browser web. Aplikasi *hybrid* dapat mengakses seperti penyimpanan offline, serta akses ke kontak, GPS dan media lainnya pada perangkat mobile.

4. Dapat diakses secara offline

Meskipun dibangun dengan html dan javascript, aplikasi *hybrid* seperti aplikasi *native* pada umumnya yaitu dapat diakses lokal pada device meskipun device offline (tidak terkoneksi ke jaringan).

2.2.7 Apache Cordova

Apache Cordova adalah sebuah set API device yang memungkinkan pengembang aplikasi mobile untuk mengakses fungsi perangkat asli seperti kamera atau *accelerometer* dengan menggunakan javascript. Dengan mengkombinasikan framework UI seperti jQuery Mobile, ionic framework, Dojo Mobile atau Sencha Touch, hal ini memungkinkan aplikasi smartphone untuk dikembangkan dengan menggunakan HTML, CSS, dan Javascript.

Ketika menggunakan API Cordova sebuah aplikasi dapat dibangun tanpa kode asli (Java, Objective-C dan lain lain) dari pengembang aplikasi. Sebaliknya teknologi web yang digunakan untuk membangun sebuah aplikasi, dan aplikasi tersebut dijalankan di lokal (tidak pada *server remote* http).

Aplikasi jual beli produk pertanian dibuat dengan menggunakan Cordova sebagai pembungkus kode – kode HTML, CSS, dan JavaScript dengan menggunakan SDK platform agar dapat berjalan secara lokal di device mobile tanpa melalui *server remote* http. Cordova menyediakan satu set library Javascript yang dapat dipanggil dan digunakan dalam pengembangan aplikasi. Cordova tersedia untuk platform seperti iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada dan Symbian[[COR-15](#)].

2.2.8 AngularJS

Aplikasi jual beli produk pertanian dikembangkan dengan menggunakan *AngularJS*. *AngularJS* merupakan sebuah *framework* yang dibuat untuk pemrograman JavaScript. Framework *AngularJS* dibuat langsung oleh Google dan dikembangkan oleh komunitas pengembang dan perusahaan – perusahaan lainnya. Pemilihan *AngularJS* dikarenakan framework *AngularJS* dikembangkan untuk memperluas sintaks HTML agar penulisan kode lebih jelas dan ringkas, sehingga akan lebih mempermudah dalam proses pembuatan aplikasi yang dinamis [[ANG – 15](#)]. Logo *AngularJS* ditunjukkan pada gambar 2.2



Gambar 2.2 Logo AngularJS

Sumber: [ANG – 15]

2.2.9 Ionic Framework

Aplikasi jual beli produk pertanian menggunakan *ionic framework* sebagai kerangka pembuatan UI aplikasi. *Ionic* merupakan *framework* aplikasi mobile berbasis HTML5 yang berfokus pada pengembangan aplikasi *mobile hybrid*. Tidak seperti *responsive framework*, *ionic* dilengkapi dengan elemen UI *mobile* yang sangat mirip dengan gaya *native* ketika menggunakan SDK asli dari iOS atau Android. *Ionic* perlu membutuhkan *cordova* untuk membungkus kode – kode HTML pada *ionic* [ION - 15].

Aplikasi yang dibuat dengan *ionic* merupakan sebuah halaman web yang berjalan di sebuah shell aplikasi asli. Hal ini berarti dalam proses pembuatan aplikasi dapat menggunakan HTML, CSS dan JavaScript, namun agar dapat mengakses hingga *native layer* maka harus dibungkus menggunakan *cordova*. *Ionic* juga menggunakan *AngularJS* untuk menjalankan fungsi – fungsi dari *framework* [ION - 15]. Logo *ionic* ditunjukkan pada gambar 2.3



Gambar 2.3 Logo Ionic Framework

Sumber: [ION - 15]

2.2.10 NgCordova

Pengembangan aplikasi jual beli produk pertanian ini menggunakan *NgCordova*, sebuah *extension* yang bertujuan untuk mempermudah pembuatan aplikasi dalam mengakses fitur asli pada *device*. Dengan menggunakan *NgCordova*, pengaksesan *hardware native* akan lebih mudah, hanya perlu menginstall plugin sesuai dengan yang dibutuhkan dan melakukan pemanggilan fungsi untuk mengakses *hardware native*. *NgCordova* telah mendukung 55 plugin untuk membantu mengakses *hardware native device* [NGC - 15]. Logo *NgCordova* ditunjukkan pada gambar 2.4



Gambar 2.4 Logo NgCordova

Sumber: [NGC - 15]

NgCordova merupakan penggabungan dari *cordova* dan *AngularJS*. Aplikasi jual beli produk pertanian menggunakan beberapa plugin, diantaranya *Geolocation*, *Camera* dan *FileTransfer*. Plugin *Geolocation* digunakan untuk mengakses *hardware* GPS *device* agar aplikasi dapat mengetahui lokasi user. Sedangkan plugin *Camera* digunakan untuk mengakses *hardware* Kamera *device*. Plugin *FileTransfer* digunakan untuk proses upload data hasil foto ke server.

2.2.11 Google Maps API

Google Maps adalah fitur yang disediakan oleh google, dan bukan standar android. Fitur Google Maps ini dapat digunakan secara embeded dalam beberapa bahasa pemrograman termasuk Android melalui sebuah jembatan aplikasi yang disebut API (Application Programming Interface). Google menyediakan API untuk pengembang mengntegrasikan aplikasinya dengan fitur Google Maps. Dengan menggunakan Google Maps API, peta google akan dapat diintegrasikan dengan berbagai platform pemrograman, sehingga peta google dapat tampil di halaman aplikasi yang dibuat [ADA - 12].

2.2.12 UML

Bahasa pemodelan yang digunakan dalam proses perancangan aplikasi jual beli produk pertanian adalah UML. UML merupakan bahasa pemodelan yang bertujuan untuk melakukan spesifikasi, visualisasi, konstruksi dan dokumentasi artefak dari aplikasi jual beli produk pertanian. UML digunakan untuk mempermudah dalam memahami, merancang, menelusuri, mengatur, memelihara, dan mengontrol informasi mengenai sistem yang akan dibangun. UML menangkap informasi mengenai struktur statis dan perilaku dinamis dari suatu sistem. Struktur statis menentukan jenis objek yang penting untuk sistem dan implementasinya, serta hubungan antar objek. Perilaku dinamis mendefinisikan sejarah objek dari waktu ke waktu dan komunikasi antar objek untuk mencapai tujuan [RUB - 99]. UML digambarkan dengan menggunakan berbagai macam diagram yang masing-masing memiliki fungsi dan makna tersendiri. Pada rancangan aplikasi jual beli produk pertanian digunakan 5 diagram yakni, *use case diagram*, *activity diagram*, *class diagram*, *sequence diagram* dan *entity relationship diagram*.



2.2.12.1 Use Case Diagram

Use case diagram digunakan untuk menggambarkan fungsionalitas yang dibutuhkan dari aplikasi jual beli produk pertanian. Diagram ini yang ditekankan adalah “apa” yang dapat diperbuat sistem beserta aktornya, bukan “bagaimana” sistem berjalan. Sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* membantu dalam menyusun *requirement* aplikasi jual beli produk pertanian ini dan merancang *test case* untuk semua fitur yang ada pada sistem dalam sebuah skenario *use case* [DHA - 03].

2.2.12.2 Activity Diagram

Activity Diagram digunakan untuk menggambarkan alur aktivitas dalam aplikasi jual beli produk pertanian yang dirancang, bagaimana aktivitas berawal, kondisi yang mungkin terjadi, dan bagaimana aktivitas tersebut berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Diagram *activity* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi dilakukan *trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan perilaku internal sebuah sistem (interaksi antar subsistem), tetapi lebih menggambarkan berbagai proses dan jalur aktivitas dari level atas secara umum [DHA - 03].

2.2.12.3 Class Diagram

Aplikasi jual beli produk pertanian merupakan sistem yang berorientasi objek baik pada sisi *client* maupun *server*, oleh sebab itu pada proses perancangan diperlukan *Class diagram* untuk memodelkan tampilan desain statis dari suatu sistem. *Class diagram* menunjukkan satu set *class*, *interface*, dan kolaborasi beserta hubungan antar *class*. *Class* adalah cetakan dari sekumpulan objek yang berbagi atribut, operasi, hubungan, dan semantik yang sama [BOC - 98]. *Class* dapat berupa implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class* [DHA - 03].

2.2.12.4 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan interaksi antar objek di dalam dan di sekitar aplikasi jual beli produk pertanian (termasuk pengguna, *display*, dan *database*) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan [DHA - 03]. Diagram *sequence* menampilkan interaksi sebagai grafik dua dimensi. Dimensi vertikal menunjukkan sumbu waktu. Dimensi horizontal menunjukkan peran *classifier* yang mewakili objek individu dalam berkolaborasi. Setiap peran *classifier* diwakili oleh kolom vertikal. Selama waktu

untuk suatu objek masih ada, maka suatu peran ditunjukkan dengan garis putus-putus. Selama waktu aktivasi prosedur pada objek yang aktif, maka digambarkan sebagai garis ganda [RUB - 99].

2.2.13 Pengujian perangkat lunak

Penelitian rancang bangun aplikasi jual beli produk pertanian dengan menggunakan pendekatan *Location Based Services* (LBS) perlu dilakukan pengujian sistem yang telah dirancang dan diimplementasikan dengan tujuan untuk menentukan kesalahan dan segala kemungkinan yang akan menimbulkan kesalahan sesuai dengan spesifikasi aplikasi yang telah ditentukan. Berdasarkan standart IEEE, pengujian perangkat lunak meliputi pengertian aktivitas yang dilakukan untuk mengevaluasi kualitas produk dan untuk mengembangkannya dengan mengidentifikasi kelemahan dan permasalahan yang terjadi [SIM - 10].

Strategi untuk pengujian perangkat lunak mengintegrasikan metode desain kasus uji perangkat lunak kedalam serangkaian langkah yang disusun dengan baik dan hasilnya adalah konstruksi perangkat bergerak yang berhasil. Strategi pengujian dapat dilakukan melalui pengujian tingkat rendah yaitu pengujian yang dilakukan pada kode program untuk membuktikan bahwa segmen kode sumber yang kecil telah diimplementasikan dengan tepat. Selain itu terdapat pengujian tingkat tinggi yang memvalidasi fungsi-fungsi sistem mayor yang tidak sesuai dengan kebutuhan pengguna [PRE - 10].

Pada aplikasi jual beli produk pertanian ini dilakukan pengujian dengan menggunakan pengujian validasi untuk memvalidasi fungsi – fungsi aplikasi mayor yang tidak sesuai dengan kebutuhan pengguna, serta dilakukan pengujian *usability* untuk mengukur kepuasan akan kemudahan penggunaan dan kebermanfaatannya yang diberikan oleh sistem. Berikut merupakan penjelasan dari pengujian validasi dan pengujian *usability*.

2.2.13.1 Pengujian validasi

Pengujian validasi aplikasi jual beli produk pertanian dicapai melalui sederetan pengujian *blackbox* yang menampilkan kesesuaian sistem dengan persyaratan kebutuhan yang telah didefinisikan pada *use case diagram* dan skenario *use case*. Pengujian menguraikan kelas – kelas pengujian yang akan digunakan untuk mengungkap kesalahan ketika terjadi penyesuaian sistem dengan persyaratan. Pengujian dan skenario uji keduanya didesain untuk memastikan apakah semua persyaratan fungsional terpenuhi pada sistem, semua persyaratan kinerja tercapai, dan dokumentasi dibuat dengan benar [PRE - 10].

2.2.13.2 Pengujian *usability*

Pada penelitian ini, pengujian *usability* dilakukan untuk mengkaji dan menilai seberapa mudah penggunaan aplikasi yang dibuat. Artinya kata "*usability*" mengacu pada metode untuk melakukan improvisasi terhadap *ease-of use* selama proses perancangan.

Pengujian *usability* mencakup 5 komponen yaitu [NIE - 12]:

1. *Learnability*

Semudah apa pengguna dapat mempelajari penggunaan produk tersebut untuk pertama kali.

2. *Efficiency*

Secepat apa pengguna dapat melakukan tugasnya.

3. *Memorability*

Sejauh mana pengguna dapat mengingat langkah-langkah atau proses yang dilakukan untuk mencapai tujuannya.

4. *Errors*

Sebanyak apa, sejauh mana dan semudah apa bagi pengguna jika berhubungan pada eror yang ada.

5. *Satisfaction*

Bagaimana tanggapan pengguna terhadap rancangan produk secara keseluruhan.

➤ **Kuesioner USE**

Dalam penelitian ini, acuan kuesioner yang dipakai adalah kuesioner USE. Kuesioner USE merupakan media kuesioner untuk mengukur *usability* dengan memakai 3 parameter yaitu kegunaan (*usefulness*), kepuasan (*satisfaction*) dan kemudahan penggunaan (*ease of use*). *Ease of use* merupakan sebuah parameter yang dibagi menjadi 2 faktor yaitu kemudahan dalam penggunaan (*ease of use*) dan kemudahan dalam mempelajari aplikasi (*ease of learning*) [AEL - 12].

Contoh beberapa pertanyaan dalam kuesioner USE adalah sebagai berikut:

1. *Usefulness*

- Aplikasi ini dalam pengerjaannya memenuhi ekspektasi saya.
- Aplikasi ini membuat saya menjadi lebih produktif.
- Aplikasi ini sangat berguna.

2. *Ease of Use*

- Aplikasi ini mudah digunakan.
- Aplikasi ini *user-friendly*.
- Aplikasi ini fleksibel.

3. *Ease of Learning*

- Aplikasi ini dapat dengan mudah dan cepat saya pelajari.
- Aplikasi ini mudak diingat dalam penggunaannya.

4. *Satisfaction*

- Aplikasi ini menyenangkan untuk digunakan.
- Saya merasa saya harus memiliki aplikasi ini.

➤ **Skala Likert**

Skala Likert dalam penelitian ini digunakan sebagai acuan penilaian dan analisis hasil dalam melakukan survei untuk keperluan pengujian *usability*. Metode Likert

merupakan metode penskalaan pernyataan sikap yang menggunakan distribusi respons sebagai dasar penentuan nilai skalanya. Nilai dari skala Likert tergantung kebutuhan. Skala Likert menjabarkan variabel yang diukur menjadi indikator variabel. Indikator tersebut kemudian dijadikan sebagai titik tolak untuk menyusun butir-butir instrumen yang dapat berupa pernyataan atau pernyataan [RIS - 12].

Jawaban setiap pertanyaan atau pernyataan yang menggunakan skala Likert mempunyai gradasi dari sangat positif sampai sangat negatif yang dapat berupa kata-kata sebagai contoh:

- a. Sangat setuju
- b. Setuju
- c. Ragu-ragu
- d. Tidak setuju
- e. Sangat tidak setuju

Untuk keperluan analisis kuantitatif, maka jawaban tersebut dapat diberi skor yang ditunjukkan pada Tabel 2.1.

Tabel 2.1 Penilaian jawaban kuantitatif

Jawaban	Skor
Sangat setuju	5
Setuju	4
Netral	3
Tidak setuju	2
Sangat tidak setuju	1

Perhitungan skala Likert diterapkan untuk mendapatkan indeks presentase pengujian *usability*. Persamaan 2.1 digunakan untuk menghitung Total Skor, Persaman 2.2 digunakan untuk menghitung nilai Y dan Persamaan 2.3 digunakan untuk menghitung indeks presentase.

$$TotalSkor = (nilaiSTS \times 1) + (nilaiTS \times 2) + (nilaiRG \times 3) + (nilaiST \times 4) + (nilaiSS \times 5) \quad (2-1)$$

Persamaan 2.1 Mencari Nilai TotalSkor

$$Y = SkorLikerTertinggi \times JumlahResponden \quad (2-2)$$

Persamaan 2.2 Mencari Nilai Y

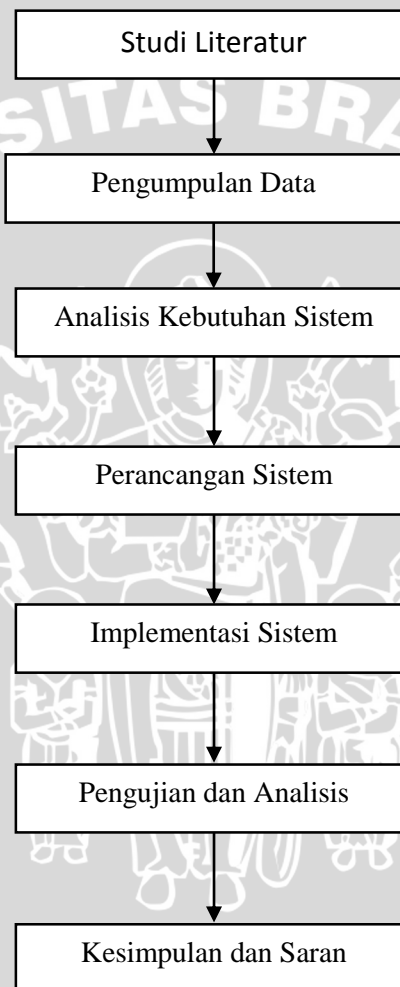
$$Index(\%) = (TotalSkor / Y) \times 100\% \quad (2-3)$$

Persamaan 2.3 Mencari Nilai Indeks



BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metodologi yang digunakan dalam pembuatan aplikasi *mobile* jual beli produk pertanian menggunakan pendekatan *location based services*. Metodologi penelitian yang dilakukan dalam penelitian ini melalui beberapa tahapan yaitu studi literatur, pengumpulan data, analisis kebutuhan dan perancangan, implementasi sistem, pengujian dan kesimpulan. Tahapan – tahapan dalam penelitian tersebut dapat diilustrasikan dengan diagram blok metodologi penelitian seperti pada Gambar 3.1.



Gambar 3.1 Diagram Blok Metodologi Penelitian

3.1 Studi literatur

Studi literatur dilakukan untuk memahamai secara mendalam hal-hal yang berkaitan dengan penelitian, baik dari segi konsep, dasar teori, maupun hal-hal yang membantu dalam proses perancangan, implementasi, dan pengujian sistem.

Sumber-sumber yang digunakan pada penelitian ini sebagai bahan pembelajaran telah dilampirkan pada bagian daftar pustaka. Sumber pembelajaran pada penelitian ini didapatkan melalui buku tulis maupun media internet dalam bentuk jurnal ilmiah, maupun tugas akhir. Adapun yang dijadikan bahan studi literatur untuk mendukung penelitian ini antara lain:

- a. Pertanian
- b. Sistem Informasi
- c. *Location Based Services* (LBS)
- d. *Web Service*
- e. *Javascript Object Notation* (JSON)
- f. *Hybrid Aplication*
- g. *Apache Cordova*
- h. *AngularJS*
- i. *Ionic Framework*
- j. *NgCordova*
- k. Google Maps API
- l. UML
- m. Pengujian Perangkat Lunak

3.2 Pengumpulan data

Pengumpulan data dilakukan untuk mendapatkan data terkait produk pertanian dan transaksi jual beli antar petani dan pembeli yang berkaitan dengan pembuatan aplikasi *mobile* jual beli produk pertanian. Pengumpulan data pada penelitian ini dilakukan dengan teknik penelitan lapangan. Penelitian lapangan atau observasi bertujuan untuk mengetahui secara langsung lingkungan penelitian dan objek yang diteliti sehingga dapat melengkapi data guna pengembangan aplikasi *mobile* jual beli produk pertanian. Observasi dilakukan di Kabupaten Jember untuk mengetahui data terkait penelitian. Hal ini dikarenakan wilayah Jawa Timur khususnya Kabupaten Jember mayoritas penduduk bermatapencaharian sebagai petani.

3.3 Analisis kebutuhan sistem

Pada tahap ini dilakukan analisis kebutuhan sistem yang akan diimplementasikan pada perangkat lunak. Analisis sistem ini menunjukkan alur proses, kebutuhan, dan fitur apa saja pada aplikasi. Langkah – langkah dalam analisis kebutuhan tersebut adalah (1) identifikasi aktor, (2) analisis data, (3) identifikasi kebutuhan, (4) pembuatan *use case diagram*, (5) pembuatan *use case scenario*.

3.4 Perancangan aplikasi

Perancangan sistem dilakukan sebagai panduan sebelum melakukan implementasi. Perancangan didasarkan pada kebutuhan yang sudah di data pada analisis kebutuhan. Perancangan yang dilakukan dalam penelitian ini antara lain perancangan sistem penggunaan aplikasi mobile jual beli produk pertanian yang akan digambarkan menggunakan *activity diagram*, perancangan hubungan antar objek yang akan digambarkan menggunakan *sequence diagram*, perancangan *class* pada sistem yang akan digambarkan dengan *class diagram*, perancangan penyimpanan dan hubungan data dalam tabel yang digambarkan dengan *entity relationship diagram*, dan perancangan antarmuka pengguna untuk aplikasi *client*.

3.5 Implementasi sistem

Pada tahapan ini dilakukan pembuatan program dan merepresentasikan hasil desain ke dalam pemrograman berdasarkan sistem yang telah dirancang sebelumnya. Implementasi aplikasi antara lain:

1. Pembuatan aplikasi *client* menggunakan pemrograman HTML5, CSS dan javascript dengan menggunakan *Ionic Framework*.
2. Pembuatan *backend* sistem berupa aplikasi *server* dengan menggunakan bahasa pemrograman PHP, HTML, dan javascript dengan memanfaatkan *framework* panada.
3. Menggunakan MySQL sebagai *database*.

3.6 Pengujian dan analisis

Pada tahap pengujian dan analisis digunakan untuk mengetahui seberapa berhasil aplikasi dibangun sesuai dengan perancangan aplikasi yang telah dilakukan sebelumnya. Item – item yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian yang dilakukan meliputi pengujian dengan metode *blackbox* menggunakan metode pengujian validasi sistem dan juga pengujian dengan menggunakan metode pengujian *usability*.

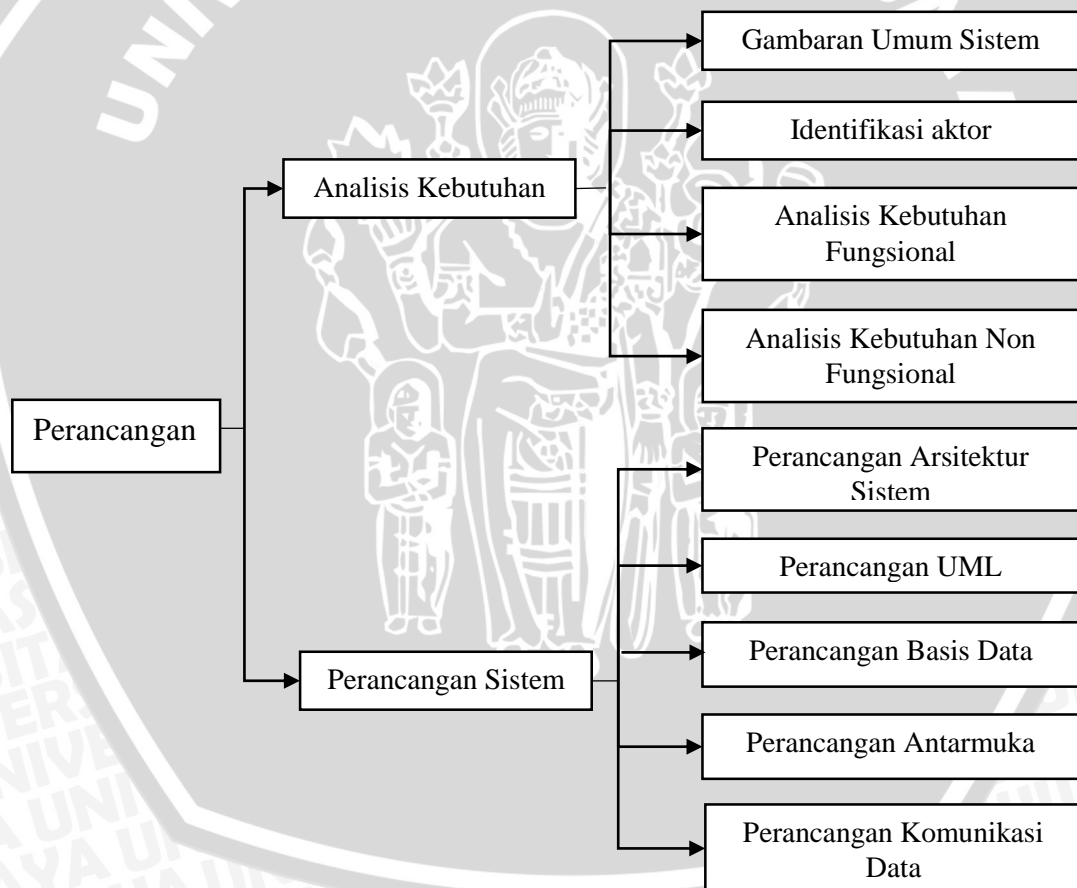
Setelah melakukan pengujian terhadap aplikasi, selanjutnya adalah menganalisis hasil dari pengujian. Analisis dilakukan untuk mengetahui hasil dari pengujian perangkat lunak sehingga akan didapatkan kesimpulan dari pengembangan perangkat lunak yang dilakukan.

3.7 Kesimpulan dan saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis setelah pembuatan aplikasi *mobile* jual beli produk pertanian menggunakan pendekatan *Location Based Services*. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan – kesalahan yang terjadi serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

BAB 4 PERANCANGAN

Pada bab ini akan membahas tentang perancangan dari Aplikasi Mobile Jual Beli Produk Pertanian menggunakan pendekatan *Location Based Services* (LBS) yang akan dibangun. Sistem dibangun dalam dua sisi, dimana terdapat aplikasi *front end* berbasis aplikasi mobile yang digunakan oleh pengguna (petani maupun pembeli), dan aplikasi *back end* berbasis aplikasi web base yang digunakan oleh admin. Pada bab ini akan lebih difokuskan pada pembahasan aplikasi dari sisi *client*. Perancangan yang dilakukan meliputi dua tahap. Tahap pertama adalah analisis sistem yang berisi tentang gambaran umum sistem dan kebutuhan sistem. Selanjutnya tahap kedua adalah bagian perancangan aplikasi. Pada perancangan aplikasi akan membahas lebih detail tentang perancangan aplikasi tersebut berdasarkan analisis kebutuhan yang telah dilakukan sebelumnya. Diagram perancangan aplikasi *mobile* jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram perancangan

4.1 Analisis kebutuhan

Proses analisis kebutuhan mengacu pada gambaran umum perangkat lunak jual beli produk pertanian menggunakan *location based services* dan hasil

pengumpulan, pemahaman, dan penetapan kebutuhan – kebutuhan yang ingin didapatkan oleh pengguna. Proses analisis kebutuhan diawali dengan penjabaran gambaran umum aplikasi jual beli produk pertanian, identifikasi aktor yang terlibat, analisis data, penjelasan tentang daftar kebutuhan dan kemudian memodelkannya ke dalam diagram *use case*. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan – kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

4.1.1 Gambaran umum sistem

Gambaran umum sistem jual beli produk pertanian menggunakan pendekatan *location based services* berbasis *mobile* terdiri dari dua bagian, yaitu deskripsi umum sistem jual beli produk pertanian dan lingkungan sistem jual beli produk pertanian.

4.1.1.1 Deskripsi sistem

Sistem jual beli produk pertanian ini merupakan sistem yang dirancang untuk perangkat bergerak pada sisi *client*. Sistem ini dikembangkan untuk membantu mempermudah petani dalam memasarkan hasil pertanian secara lebih luas dan langsung ke pembeli. Sistem ini juga sebagai sarana untuk memudahkan pembeli dalam mencari hasil pertanian yang diinginkan berdasarkan lokasi. Terdapat aktor admin yang bertugas untuk mengolah dan memverifikasi data yang ada pada sisi *server* yaitu data yang telah diinputkan oleh petani melalui aplikasi *client*.

Dengan menggunakan media internet maka pemasaran dan jangkauan pembeli pun akan semakin luas. Petani dapat memasarkan produk pertaniannya dengan mengunggah data ke server melalui aplikasi, sehingga calon pembeli dapat melihat hasil – hasil pertanian yang dipasarkan oleh petani melalui aplikasi.

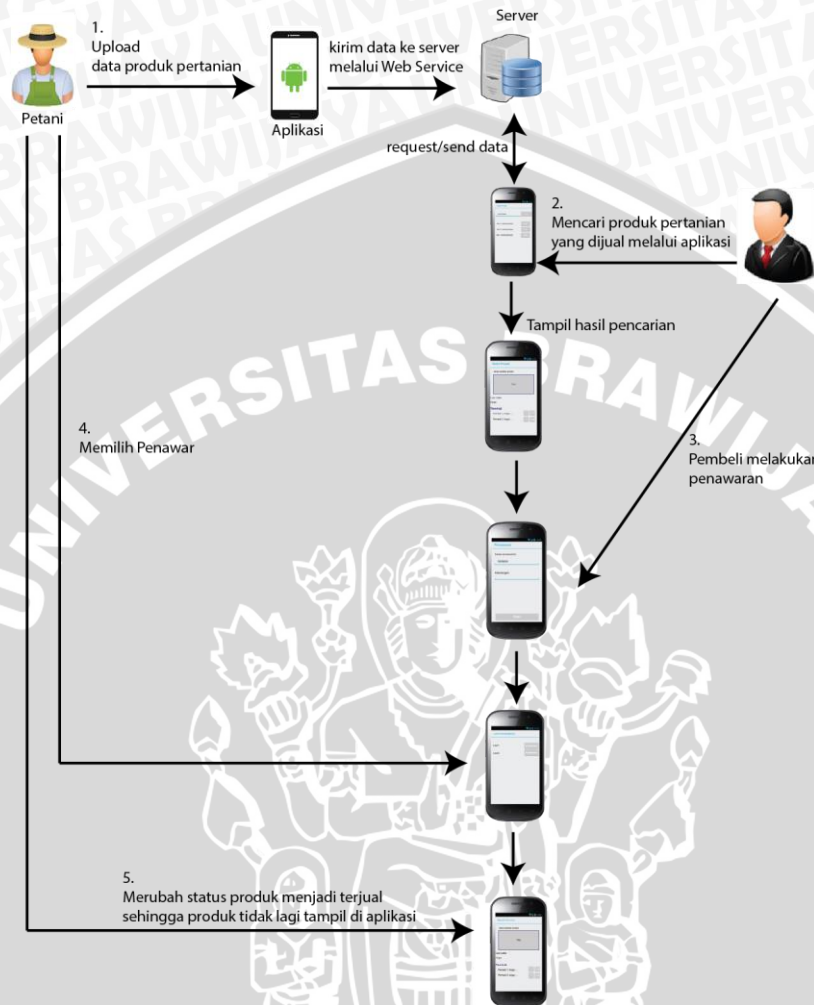
Bagi para pembeli yang ingin mencari hasil pertanian dapat menggunakan aplikasi jual beli produk pertanian untuk mencari hasil pertanian yang diinginkan, aplikasi akan menampilkan hasil pertanian yang dijual oleh petani yang berada di sekitar lokasi pembeli. Dengan menampilkan hasil pertanian berdasarkan lokasi maka akan lebih mempermudah pembeli dalam mencari hasil pertanian yang diinginkan.

4.1.1.2 Lingkungan sistem

Aplikasi yang dibangun pada sistem ini terbagi menjadi dua, yaitu aplikasi pada sisi *client* menggunakan perangkat mobile dengan memanfaatkan fungsionalitas dari *mobile*. Aplikasi *client* dibangun dengan berbasis *Hybrid* dengan menggunakan *Ionic Framework*, *NgCordova* dan *Apache Cordova*. Sedangkan aplikasi Administrator berbasis *website* yang digunakan untuk manajemen data yang diinputkan oleh pengguna serta berfungsi juga sebagai server untuk penyedia layanan data dari aplikasi client.

4.1.1.3 Bisnis proses sistem

Bisnis proses sistem jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Bisnis proses sistem

Dalam Gambar 4.2 menjelaskan bahwa langkah pertama dalam melakukan proses jual beli produk pertanian menggunakan aplikasi adalah petani melakukan upload data produk pertanian yang akan dijual melalui aplikasi *mobile*, selanjutnya aplikasi *mobile* akan mengirim data ke *server* untuk disimpan. Langkah selanjutnya adalah Pembeli atau konsumen melakukan pencarian produk pertanian yang akan dibeli melalui aplikasi. Setelah menemukan produk pertanian yang dicari, pembeli dapat melakukan penawaran kepada petani dengan memasukkan harga yang diinginkan beserta keterangan penawaran melalui aplikasi. Petani memilih penawaran yang sesuai dengan kriteria petani. Pada sistem ini proses transaksi keuangan berada diluar sistem, sehingga petani maupun pembeli dapat melakukan telepon melalui fitur yang telah disediakan aplikasi. Setelah proses transaksi jual beli selesai, petani merubah status produknya menjadi terjual sehingga produk tidak lagi tampil di aplikasi.

4.1.2 Identifikasi aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor – aktor yang akan berinteraksi dengan sistem jual beli produk pertanian. Pada tabel 4.1 menunjukkan aktor – aktor yang berhubungan dengan sistem.

Tabel 4.1 Identifikasi aktor

Aktor	Deskripsi
Pengguna	Pengguna merupakan aktor yang menggunakan aplikasi <i>client</i> , dan tidak memiliki akses ke server. Pengguna dapat bertindak sebagai petani sekaligus juga sebagai pembeli. Pengguna yang sudah terdaftar dan telah terverifikasi dapat menggunakan aplikasi <i>client</i> .
Admin	Admin merupakan orang yang memiliki wewenang penuh dalam sistem untuk menambah, mengubah, dan menghapus data. Admin juga memiliki tugas untuk memverifikasi pendaftaran pengguna baru.

4.1.3 Analisis kebutuhan fungsional

Daftar kebutuhan terdiri dari kebutuhan fungsional dan non fungsional. Pada daftar kebutuhan fungsional akan dispesifikasikan yaitu spesifikasi kebutuhan fungsional dari Petani, pembeli, dan admin. Daftar kebutuhan fungsional akan lebih dijabarkan menggunakan *use case diagram* dan skenario *use case*. Spesifikasi kebutuhan fungsional ditunjukkan pada Tabel 4.2.

Tabel 4.2 Daftar kebutuhan fungsional

Nomor	Kebutuhan	Use Case	Jenis
KF-001	Aplikasi mampu menjalankan mekanisme pendaftaran anggota baru	<i>Sign up</i>	Pengguna
KF-002	Aplikasi mampu menampilkan seluruh produk yang dijual melalui visualisasi <i>map</i> (peta)	Lihat produk dijual (<i>map</i>)	Pengguna
KF-003	Aplikasi mampu menampilkan hasil pencarian produk pertanian yang dijual berdasarkan kategori hasil pertanian	Pencarian Produk dijual	Pengguna

Tabel 4.2 Daftar kebutuhan fungsional (lanjutan)

Nomor	Kebutuhan	Use Case	Jenis
KF-004	Aplikasi dapat menampilkan detail produk yang dipilih oleh pengguna terkait foto dan keterangan dari hasil pertanian yang dipilih.	Lihat detail produk dijual	Pengguna
KF-005	Aplikasi mampu menampilkan form jual produk yang harus diisi oleh pengguna serta proses foto produk pertanian yang akan dijual dan dapat mengirim data, mengambil koordinat lokasi pengguna dan menyimpan data ke server	Jual Produk	Pengguna
KF-006	Aplikasi mampu menelpon petani sesuai dengan nomor yang bersangkutan	Telepon pengguna	Pengguna
KF-007	Aplikasi dapat menampilkan form penawaran yang harus diisi pengguna dan mengirim data ke server untuk disimpan	Ajukan Penawaran	Pengguna
KF-008	Aplikasi dapat menampilkan daftar penawar dan pengguna dapat memilih salah satu penawar.	Memilih Penawar	Pengguna

Tabel 4.2 Daftar kebutuhan fungsional (lanjutan)

Nomor	Kebutuhan	Use Case	Jenis
KF-009	Aplikasi dapat menampilkan halaman profil pengguna lain sesuai dengan yang dipilih	Lihat profil pengguna lain	Pengguna
KF-010	Aplikasi dapat menampilkan form pelaporan petani dan mengirim data petani yang dilaporkan oleh pengguna ke server	Laporkan Pengguna	Pengguna
KF-011	Aplikasi dapat menampilkan halaman profil pribadi sesuai dengan pengguna yang login di Aplikasi	Lihat Profil Pribadi	Pengguna
KF-012	Aplikasi dapat menampilkan halaman manajemen produk	Manajemen Produk	Pengguna
KF-013	Aplikasi dapat menampilkan daftar produk yang telah dijual oleh pengguna.	Melihat daftar produk pengguna	Pengguna
KF-014	Aplikasi dapat menampilkan daftar penawaran yang telah dibuat oleh pengguna	Melihat daftar penawaran pengguna	Pengguna
KF-015	Aplikasi dapat menampilkan seluruh daftar tawaran dari produk yang dijual oleh pengguna	Melihat daftar tawaran produk pengguna	Pengguna

Tabel 4.2 Daftar kebutuhan fungsional (lanjutan)

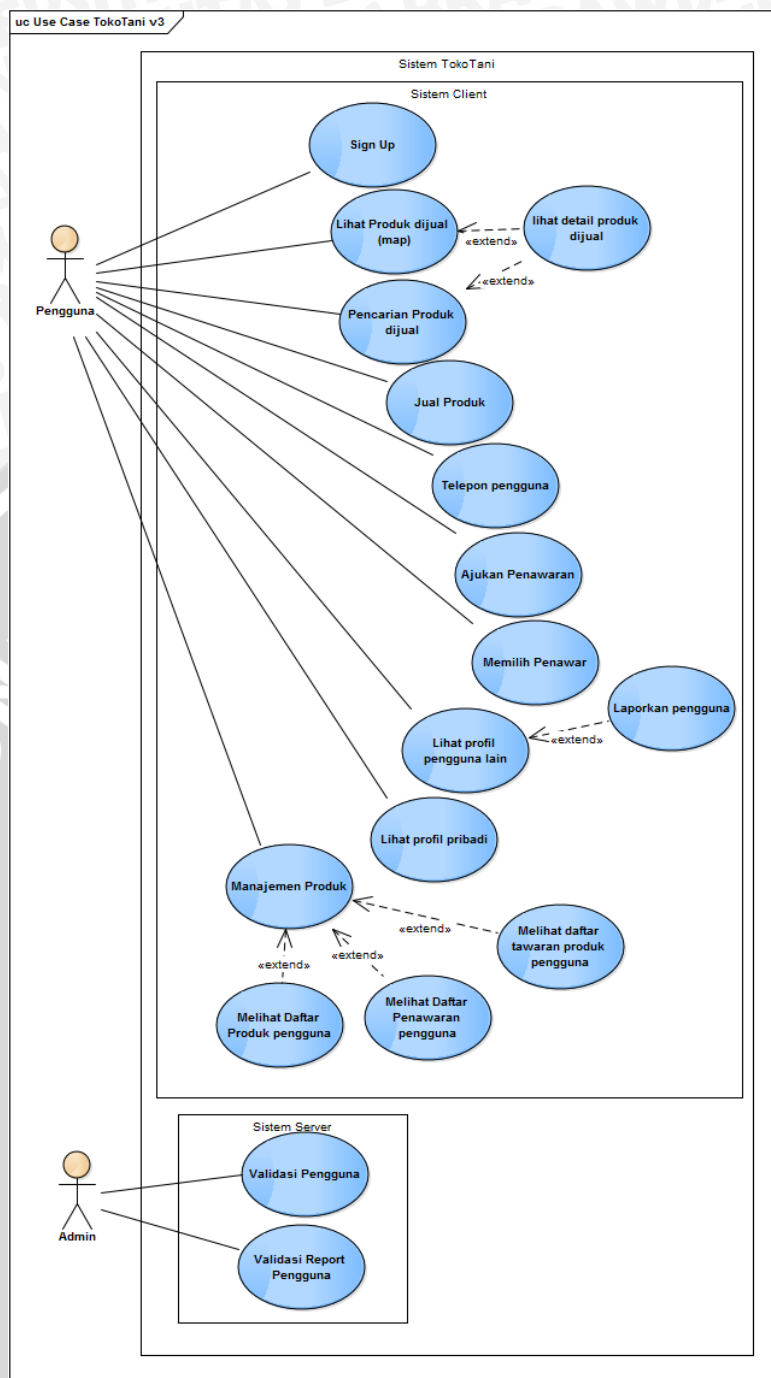
Nomor	Kebutuhan	Use Case	Jenis
KF-016	Sistem <i>server</i> dapat memvalidasi pendaftar dari pengguna yang melakukan pendaftaran melalui aplikasi <i>client</i>	Validasi Pengguna	Admin
KF-017	Sistem <i>server</i> dapat melakukan validasi <i>report</i> / laporan yang telah dikirimkan oleh pengguna melalui aplikasi <i>client</i>	Validasi Report Pengguna	Admin

4.1.3.1 Use Case diagram

Pada sistem ini pemodelan kebutuhan fungsional dimodelkan menggunakan *use case* diagram untuk memodelkan aspek perilaku sistem. Terdapat 2 aktor pada sistem ini yaitu Pengguna dan Admin. Aktor Pengguna hanya dapat mengakses aplikasi pada sisi *client* yang berbasis *mobile*, sedangkan Admin dapat mengakses aplikasi pada sisi *server* yang berbasis Web. Diagram *use case* untuk aplikasi jual beli produk pertanian ditunjukkan dalam Gambar 4.3.

4.1.3.2 Skenario Use Case

Skenario *use case* digunakan untuk menjelaskan masing – masing *use case* yang terdapat pada Gambar 4.3. skenario *use case* menguraikan mengenai nama *use case*, aktor yang berhubungan dengan *use case* tersebut, tujuan dari *use case*, deskripsi global tentang *use case*, kondisi awal yang harus dipenuhi dan kondisi akhir yang diharapkan setelah berjalannya fungsional *use case*. Selain itu juga akan diberikan ulasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang diberikan oleh aktor. Berikut skenario *use case* untuk sistem jual beli produk pertanian.



Gambar 4.3 Diagram Use Case

1. Skenario Use Case jual produk

Tabel 4.3 menjelaskan bagaimana proses untuk menambahkan data produk yang akan dijual. Tahap awal yang dilakukan adalah Pengguna masuk ke menu kelola produk, kemudian memilih tombol Tambah produk. Selanjutnya aplikasi akan menampilkan form isian detail produk yang akan dijual. Pengguna diharuskan mengisi form dengan lengkap, mencakup jenis produk pertanian, luas lahan

tanam, jenis bibit, harga, lokasi, foto. Setelah semua form terisi, pengguna menekan tombol Simpan untuk menyimpan data produk pertanian ke server.

Tabel 4.3 Skenario Use Case jual produk

Nomor Use Case	KF-005
Nama Use Case	Jual produk
Prasyarat Konteks	- Pengguna telah melakukan pemasangan aplikasi pada perangkat Android - Pengguna sudah login di aplikasi.
Tujuan dalam Konteks	Pengguna dapat menjual produk pertanian ke aplikasi sehingga dapat dipasarkan lebih luas.
Prakondisi	- Aplikasi telah terpasang pada perangkat <i>smartphone</i>
Kondisi Akhir Sukses	- Aplikasi menampilkan daftar produk pertanian yang telah dijual oleh pengguna yang bersangkutan.
Kondisi Akhir Gagal	Pengguna menerima peringatan gagal apabila proses gagal.
Aktor	Pengguna
Trigger	Pengguna masuk ke menu Jual Produk.
Alur Utama	1. Pengguna mengisi data sesuai dengan form yang tersedia di halaman jual produk.
	2. Pengguna mengambil gambar produk pertanian yang akan dijual dengan menekan tombol "Ambil gambar"
	3. Pengguna menekan tombol kirim.
	4. aplikasi menyimpan data ke server.

2. Skenario Use Case lihat produk dijual (map)

Tabel 4.4 menjelaskan bagaimana proses untuk melihat produk dijual melalui aplikasi. Proses melihat produk dapat dijalankan oleh pengguna, hal ini bertujuan agar semua pengguna dapat melihat seluruh produk pertanian yang dijual baik yang berada di sekitarnya. Tahap awal yang dilakukan adalah pengguna memilih menu Lihat Produk dijual, selanjutnya aplikasi menampilkan map beserta marker lokasi – lokasi produk pertanian yang dijual di sekitar pengguna.

Tabel 4.4 Skenario Use Case lihat produk dijual (map)

Nomor Use Case	KF-002
Nama Use Case	Lihat Produk dijual (map)
Prasyarat Konteks	- Pengguna telah melakukan pemasangan aplikasi pada perangkat Android - Pengguna telah melakukan login ke aplikasi.
Tujuan dalam Konteks	Pengguna dapat melihat seluruh produk pertanian yang dijual melalui aplikasi.
Prakondisi	- Aplikasi telah terpasang pada perangkat <i>smartphone</i>
Kondisi Akhir Sukses	- Aplikasi menampilkan <i>map</i> dengan daftar seluruh produk pertanian yang dijual melalui aplikasi.

	- Jika tidak ada produk pertanian yang dijual maka aplikasi menampilkan map dan tidak ada marker produk pertanian.
Kondisi Akhir Gagal	Pengguna menerima peringatan gagal apabila proses permintaan data ke server gagal.
Aktor	Pengguna
Trigger	Pengguna memilih menu Lihat produk dijual pada aplikasi.
Alur Utama	1. Pengguna memilih menu Lihat produk dijual
	2. Aplikasi menampilkan map beserta marker di seluruh lokasi produk pertanian
	3. Jika tidak terdapat produk pertanian yang dijual maka Aplikasi akan menampilkan map dan tidak ada marker produk pertanian.

3. Skenario Use Case ajukan penawaran

Tabel 4.5 menjelaskan bagaimana proses pengguna untuk melakukan penawaran ke produk pertanian yang dijual pengguna lain. Tahap awal yang dilakukan adalah Pengguna memilih salah satu produk hasil pertanian. Selanjutnya pembeli menekan tombol tawar yang ada dibagian bawah, sistem akan menampilkan form penawaran yang berisi harga yang diajukan dan keterangan. Pengguna harus mengisi form tersebut, setelah semua diisi pengguna menekan tombol simpan.

Tabel 4.5 Skenario Use Case ajukan penawaran

Nomor Use Case	KF-007
Nama Use Case	Ajukan penawaran

Tabel 4.5 Skenario Use Case ajukan penawaran (lanjutan)

Prasyarat Konteks	- Pengguna telah melakukan pemasangan aplikasi pada perangkat Android - Pengguna telah melakukan login ke aplikasi.
Tujuan dalam Konteks	Pengguna dapat melakukan penawaran pada produk hasil pertanian yang dijual Pengguna lain melalui aplikasi.
Prakondisi	Aplikasi telah terpasang pada perangkat <i>smartphone</i> , pembeli sudah melakukan login ke aplikasi.
Kondisi Akhir Sukses	- Aplikasi menampilkan pemberitahuan sukses yang menandakan data telah berhasil disimpan ke server.
Kondisi Akhir Gagal	Pengguna menerima peringatan gagal apabila proses gagal.
Aktor	Pengguna
Trigger	Pengguna memilih produk dan masuk ke tampilan detail produk.
Alur Utama	1. Pengguna menekan tombol tawar

	2. aplikasi menampilkan form penawaran.
	3. Pengguna mengisi form secara lengkap, yang berisi harga yang ditawarkan dan keterangan.
	4. Pengguna menekan tombol simpan.

4. Skenario Use Case memilih penawar

Tabel 4.6 menjelaskan bagaimana proses untuk memilih dan mengkonfirmasi penawar yang diinginkan pengguna. Tahap awal yang dilakukan adalah petani memilih produk dan sistem akan menampilkan detail produk dan list penawar produk yang bersangkutan sesuai dengan yang dipilih. Pengguna dapat memilih salah satu penawar yang sesuai kemudian menekan tombol konfirmasi disebelah nama penawar.

Tabel 4.6 Skenario Use Case memilih penawar

Nomor Use Case	KF-008
Nama Use Case	Memilih Penawar
Prasyarat Konteks	- Pengguna telah melakukan pemasangan aplikasi pada perangkat Android
Tujuan dalam Konteks	Pengguna dapat memilih dan mengkonfirmasi pembeli yang melakukan penawaran terhadap produk pertanian yang dijualnya.
Prakondisi	Aplikasi telah terpasang pada perangkat <i>smartphone</i> , petani sudah melakukan login ke aplikasi.
Kondisi Akhir Sukses	- Aplikasi menampilkan pemberitahuan sukses yang menandakan data telah berhasil disimpan ke server.
Kondisi Akhir Gagal	Pengguna menerima peringatan gagal apabila proses gagal.

Tabel 4.6 Skenario Use Case memilih penawar (lanjutan)

Aktor	Pengguna
Trigger	Pengguna memilih produk dan masuk ke tampilan detail produk.
Alur Utama	1. Pembeli memilih penawar yang sesuai dengan yang diinginkan pada list penawar.
	2. Petani menekan tombol Konfirmasi (logo centang) disebelah nama penawar.

5. Skenario Use Case laporkan pengguna

Tabel 4.7 menjelaskan bagaimana proses untuk melaporkan pengguna. Fitur ini dapat dimanfaatkan apabila ada Pengguna yang dianggap menyimpang dari tujuan aplikasi, sehingga dengan melaporkan ke sistem nantinya admin akan memverifikasi setiap request yang masuk, jika disetujui oleh admin maka Pengguna yang dilaporkan tidak dapat masuk ke sistem. Tahap awal yang

dilakukan adalah Pengguna memilih salah satu nama Pengguna yang terdaftar, dan masuk ke profil Pengguna. Selanjutnya menekan tombol Laporkan untuk mengirim data *report* ke server.

Tabel 4.7 Skenario Use Case laporkan pengguna

Nomor Use Case	KF-010
Nama Use Case	Laporkan Pengguna
Prasyarat Konteks	- Pengguna telah melakukan pemasangan aplikasi pada perangkat Android
Tujuan dalam Konteks	Pengguna dapat melaporkan Pengguna yang dianggap menyimpang dari tujuan aplikasi.
Prakondisi	Aplikasi telah terpasang pada perangkat <i>smartphone</i>
Kondisi Akhir Sukses	- Pengguna yang dilaporkan oleh pengguna lain dan telah disetujui oleh admin tidak akan bisa login ke sistem dan tidak bisa menggunakan aplikasi.
Kondisi Akhir Gagal	Pengguna menerima peringatan gagal apabila proses gagal.
Aktor	Pengguna
Trigger	Pengguna menekan salah satu nama Pengguna lain dan masuk ke halaman profil Pengguna tersebut.
Alur Utama	1. Pengguna menekan salah satu nama Pengguna yang dipilih
	2. aplikasi menampilkan detail profil Pengguna yang dipilih
	3. Pengguna menekan tombol Laporkan
	4. Aplikasi menampilkan halaman Pelaporan, Pengguna mengisi detail laporan. Setelah selesai, pengguna menekan tombol kirim, sehingga aplikasi akan mengirim data ke server untuk disimpan

4.1.4 Analisis kebutuhan non fungsional

Analisis kebutuhan non fungsional adalah analisis untuk mengetahui spesifikasi yang dibutuhkan oleh sistem. Sistem jual beli ini lebih menekankan pada bagaimana rancangan dan implementasi aplikasi jual beli produk pertanian menggunakan pendekatan *location based services* dapat memberikan kemudahan dan manfaat penggunaan dari sistem terhadap penjualan dan pembelian produk pertanian. Pengujian yang digunakan untuk mengukur kepuasan tersebut ialah pengujian *usability* yang diharapkan dari hasil pengujian tersebut dapat melebihi target. Jadi analisis kebutuhan non fungsional dari sistem rekomendasi ini ialah *usability* dengan target penilaian lebih dari (\geq) 80, sebab syarat diterimanya sebuah sistem dari hasil pengujian *usability* menggunakan skor SUS ialah rentang nilai antara 68-100.

4.2 Perancangan sistem

Proses perancangan sistem dibagi menjadi empat bagian, yaitu perancangan UML, perancangan arsitektur sistem, perancangan basis data, dan perancangan antar muka. Perancangan sistem pada penelitian ini menggunakan pendekatan berorientasi objek yang direpresentasikan dengan menggunakan UML.

4.2.1 Perancangan UML

4.2.1.1 Perancangan *Activity Diagram*

Activity diagram digunakan untuk memodelkan aktifitas antara pengguna dan sistem yang berjalan berdasarkan pada skenario *use case* yang telah dibuat. Berikut ialah salah satu *activity diagram* pada sistem jual beli produk pertanian.

1. *Activity Diagram* proses menjual produk pertanian

Gambar 4.4 menunjukkan aktifitas – aktifitas yang terjadi di dalam proses menjual produk pertanian. Aktifitas terbagi menjadi dua berdasarkan aktor yang terlibat pada proses tersebut. Aktifitas ini dapat dijalankan jika pengguna sudah melakukan login dan telah terdaftar di sistem sesuai dengan skenario *use case*, aktifitas dimulai dengan pengguna memilih menu jual produk. Aplikasi akan menampilkan halaman jual produk pertanian dan aplikasi menampilkan form tambah produk yang harus diisi oleh petani. Setelah selesai mengisi tekan tombol simpan untuk menyimpan data.

2. *Activity Diagram* proses melihat produk yang dijual oleh pengguna lain

Gambar 4.5 menunjukkan aktifitas – aktifitas yang terjadi di dalam proses melihat produk yang dijual di aplikasi. Aktifitas terbagi menjadi dua berdasarkan aktor yang terlibat pada proses tersebut. sesuai dengan skenario *use case*, aktifitas dimulai dengan pengguna memilih menu Lihat Produk. Sistem akan *request* data ke server, Jika data yang dicari ditemukan maka aplikasi akan menampilkan map beserta lokasi – lokasi produk yang dijual oleh pengguna, namun jika data tidak ditemukan maka aplikasi akan menampilkan map tanpa marker lokasi produk pertanian yang dijual pengguna.

3. *Activity Diagram* proses mengajukan penawaran

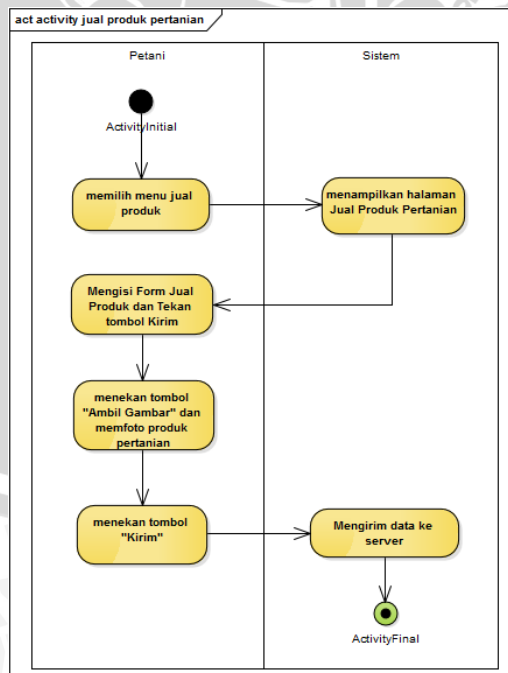
Gambar 4.6 menunjukkan aktifitas - aktifitas yang terjadi di dalam proses tawar yang dilakukan oleh pengguna. Aktifitas terbagi menjadi dua berdasarkan aktor yang terlibat pada proses tersebut. Sesuai dengan skenario *use case*, aktifitas dimulai dengan pembeli menekan produk hasil pertanian yang diinginkan sehingga sistem akan menampilkan detail produk sesuai dengan yang dipilih. Selanjutnya pengguna menekan tombol Tawar. Kemudian aplikasi akan menampilkan form penawaran yang harus diisi oleh pengguna, mencakup harga yang ditawarkan dan keterangan. Setelah form diisi pengguna menekan tombol Kirim agar aplikasi mengirim data ke server.

4. Activity Diagram proses memilih penawar

Gambar 4.7 menunjukkan aktifitas – aktifitas yang terjadi di dalam proses memilih penawar yang dilakukan oleh pengguna. Aktifitas terbagi menjadi dua berdasarkan aktor yang terlibat pada proses tersebut. Sesuai dengan skenario *use case*, aktifitas dimulai dengan pengguna menekan produk hasil pertanian yang pernah dijual oleh yang bersangkutan sehingga sistem akan menampilkan detail produk sesuai dengan yang dipilih beserta list penawaran yang telah menawar produk tersebut. Pengguna dapat memilih penawaran yang sesuai berdasarkan harga yang ditawarkan dan juga keterangan. Selanjutnya pengguna dapat menekan tombol konfirmasi yang berada di sebelah nama penawaran untuk memilih dan mengkonfirmasi penawaran yang bersangkutan, sehingga aplikasi akan mengirim data ke *server*.

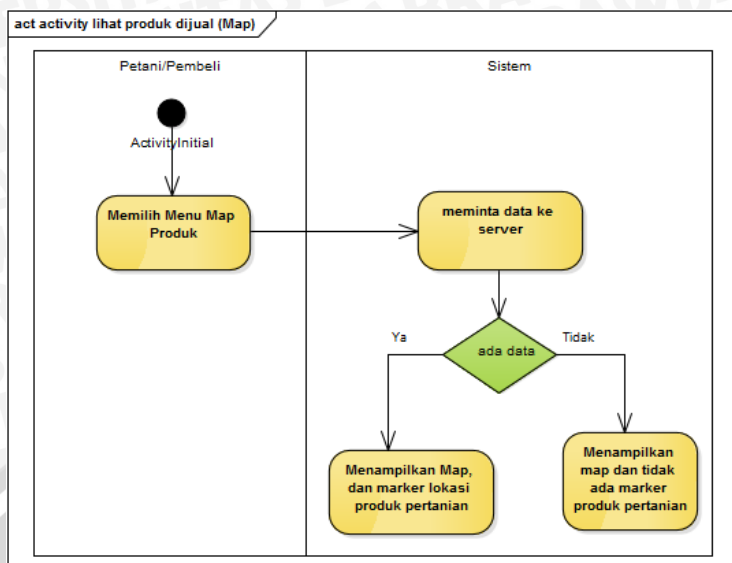
5. Activity Diagram proses melaporkan pengguna

Gambar 4.8 menunjukkan aktifitas – aktifitas yang terjadi di dalam proses melaporkan pengguna, jika pengguna menyalahi ketentuan atau tujuan dari aplikasi ini maka pengguna lain dapat melaporkannya dan nantinya Admin akan memverifikasi setiap laporan yang masuk. Aktifitas terbagi menjadi dua berdasarkan aktor yang terlibat pada proses tersebut. Sesuai dengan skenario *use case*, aktifitas dimulai dengan pengguna menekan salah satu nama pengguna yang menyalahi ketentuan dan terdaftar dalam sistem ini sehingga nantinya aplikasi akan menampilkan detail profil pengguna yang dipilih. Selanjutnya pengguna menekan tombol Laporkan, dan masuk ke halaman Pelaporan. Pengguna mengisi detail laporan, kemudian menekan tombol Kirim dan aplikasi akan mengirim data ke server yang nantinya akan diverifikasi oleh admin.

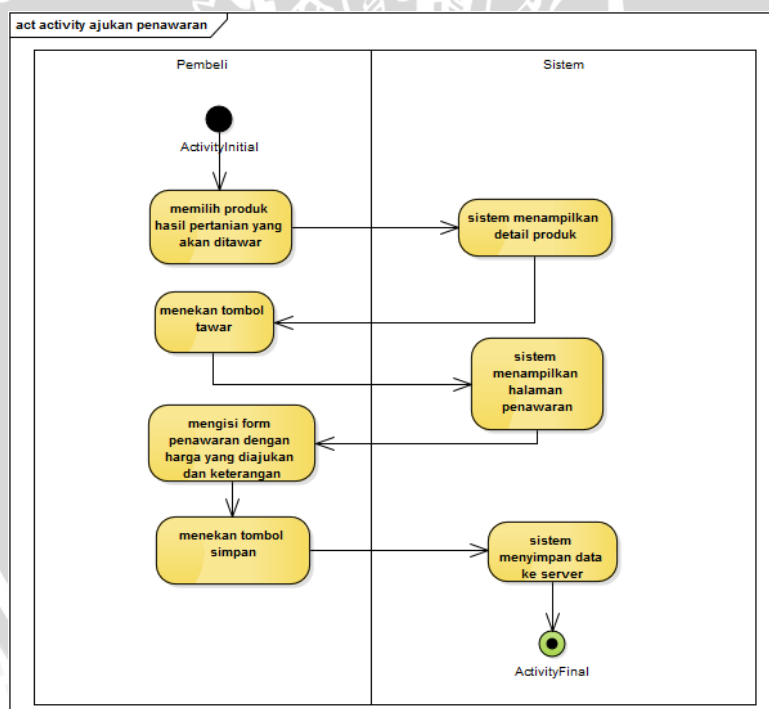


Gambar 4.4 Activity Diagram proses menjual produk pertanian

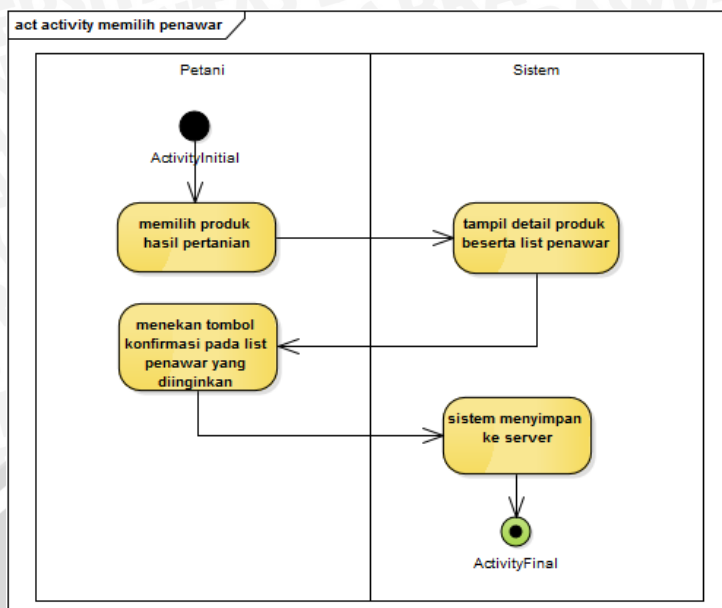




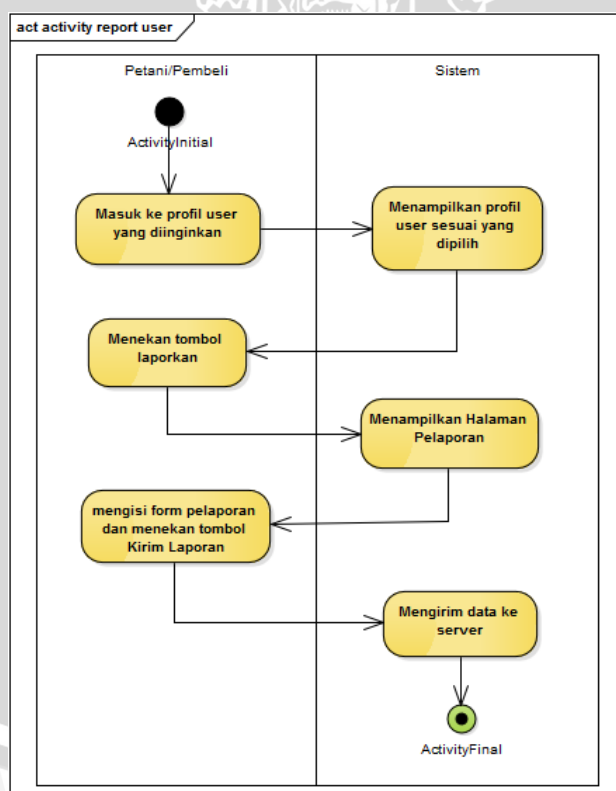
Gambar 4.5 Activity Diagram proses melihat produk yang dijual oleh pengguna lain



Gambar 4.6 Activity Diagram proses mengajukan penawaran



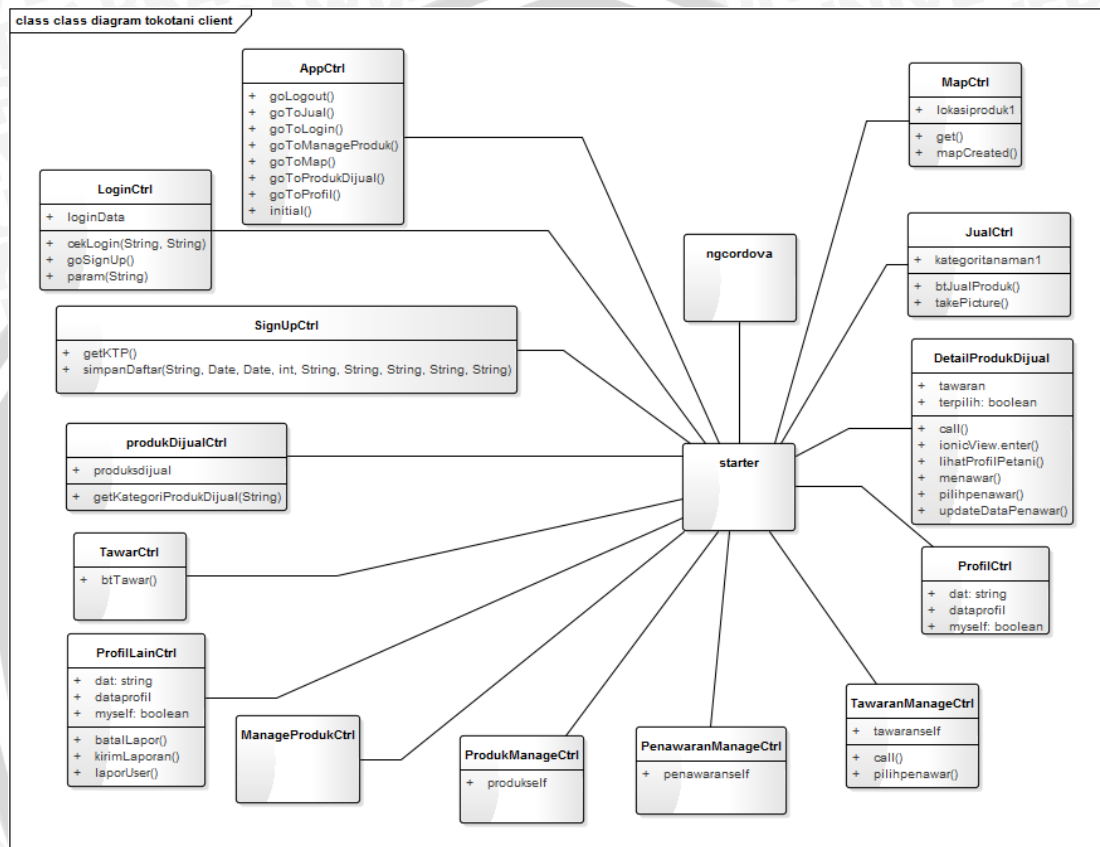
Gambar 4.7 Activity Diagram proses memilih penawaran



Gambar 4.8 Activity Diagram proses melaporkan pengguna

4.2.1.2 Perancangan *Class Diagram*

Class diagram memberikan gambaran pemodelan elemen-elemen *class* yang membentuk sebuah perangkat bergerak. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungannya satu sama lain seperti pewarisan, asosiasi dan lain-lain yang membentuk secara detail terhadap *use case* yang dimodelkan. Gambar 4.9 menunjukkan diagram *class* dari aplikasi yang dibuat.



Gambar 4.9 Diagram *Class System*

4.2.1.3 Perancangan *Sequence Diagram*

Pada diagram *sequence* akan memodelkan bagaimana hubungan antar objek dalam aplikasi berdasarkan waktu dan menerapkan fungsi-fungsi yang ada dalam objek yang terjadi di setiap skenario *use case*. Berikut merupakan diagram-diagram *sequence* pada aplikasi jual beli produk pertanian berdasarkan dari skenario *use case* yang telah dibuat sebelumnya.

1. *Sequence Diagram* pada skenario jual produk pertanian

Gambar 4.10 merupakan diagram *sequence* untuk proses menjual produk pertanian. Diagram ini menggambarkan interaksi ketika pengguna menjual produk pertanian melalui aplikasi dan selanjutnya aplikasi mengirim data ke server untuk disimpan.

2. Sequence Diagram pada skenario melihat produk yang dijual oleh pengguna lain

Gambar 4.11 merupakan diagram *sequence* untuk proses melihat produk yang dijual oleh pengguna lain. Diagram ini menggambarkan interaksi ketika pengguna melihat produk yang dijual oleh pengguna lain dalam tampilan map sesuai lokasi dari pengguna tersebut.

3. Sequence Diagram pada skenario mengajukan penawaran

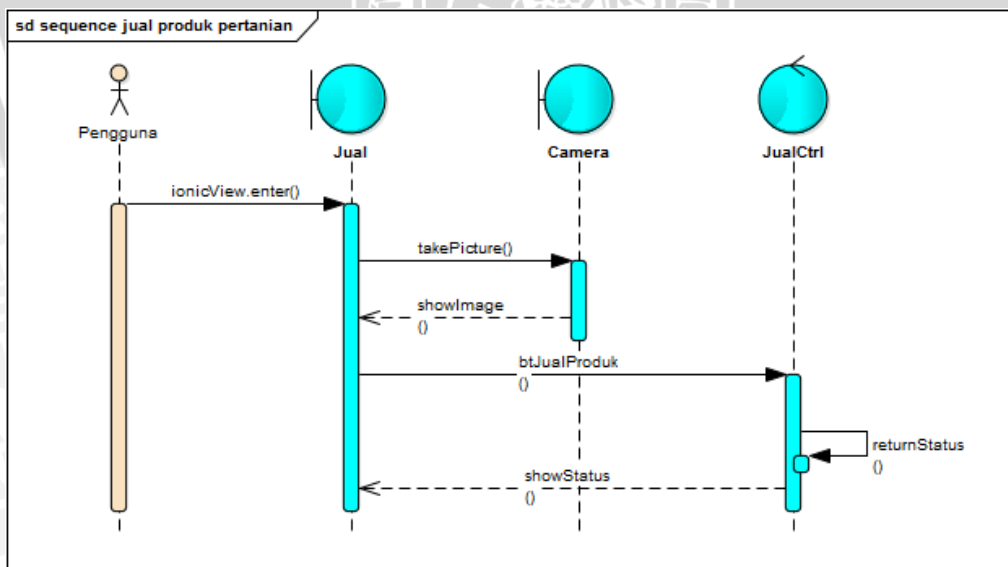
Gambar 4.12 merupakan diagram *sequence* untuk proses mengajukan penawaran pada sebuah produk pertanian yang dijual di aplikasi. Diagram ini menggambarkan interaksi ketika pengguna mengajukan penawaran terhadap produk pertanian yang dijual di aplikasi.

4. Sequence Diagram pada skenario memilih penawar

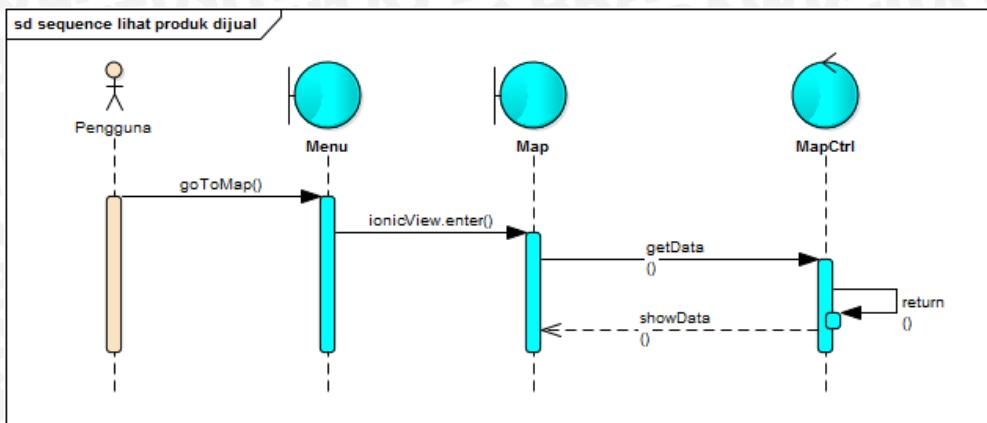
Gambar 4.13 merupakan diagram *sequence* untuk proses memilih penawar yang telah mengajukan penawaran terhadap produk pertanian milik pengguna yang bersangkutan. Diagram ini menggambarkan interaksi ketika pengguna memilih penawar.

5. Sequence Diagram pada skenario melaporkan pengguna

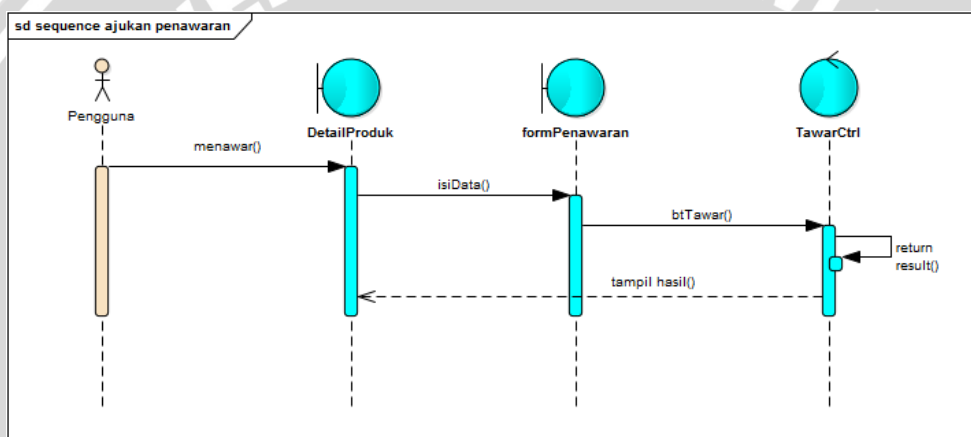
Gambar 4.14 merupakan diagram *sequence* untuk proses melaporkan pengguna yang telah menyalahi ketentuan atau tujuan dari aplikasi ini. Pengguna lain dapat melaporkannya dan nantinya Admin akan memverifikasi setiap laporan yang masuk. Diagram ini menggambarkan interaksi ketika pengguna melaporkan pengguna yang telah menyalahi ketentuan aplikasi.



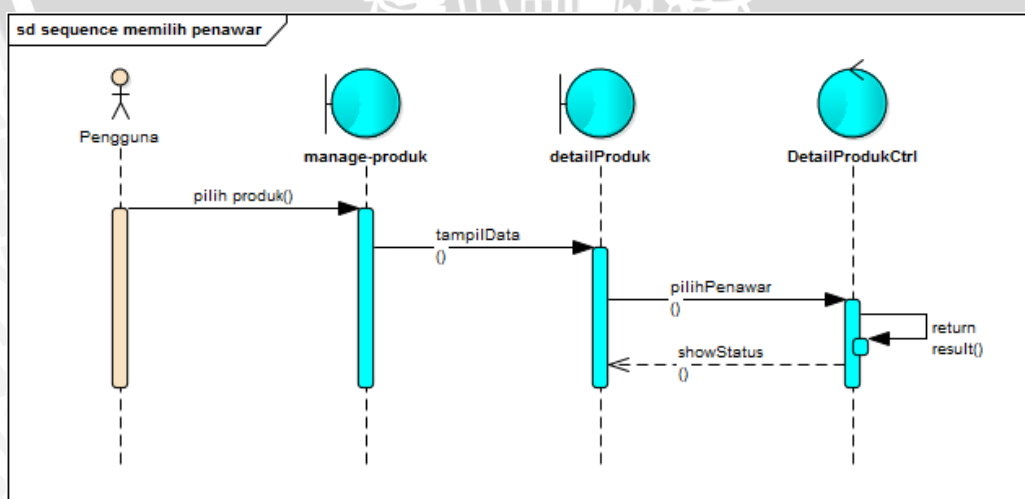
Gambar 4.10 Sequence Diagram pada skenario jual produk pertanian



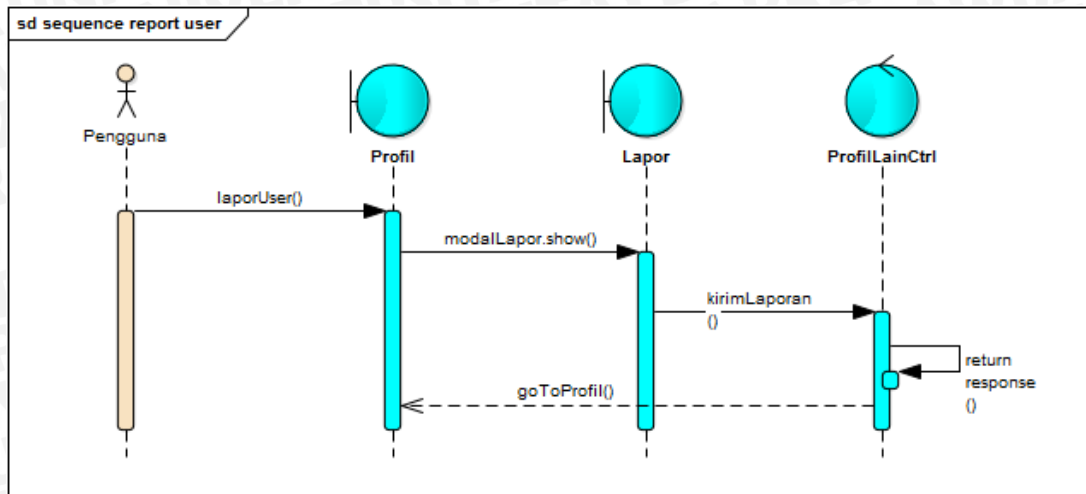
Gambar 4.11 Sequence Diagram pada skenario melihat produk yang dijual oleh pengguna lain



Gambar 4.12 Sequence Diagram pada skenario mengajukan penawaran



Gambar 4.13 Sequence Diagram pada skenario memilih penawar



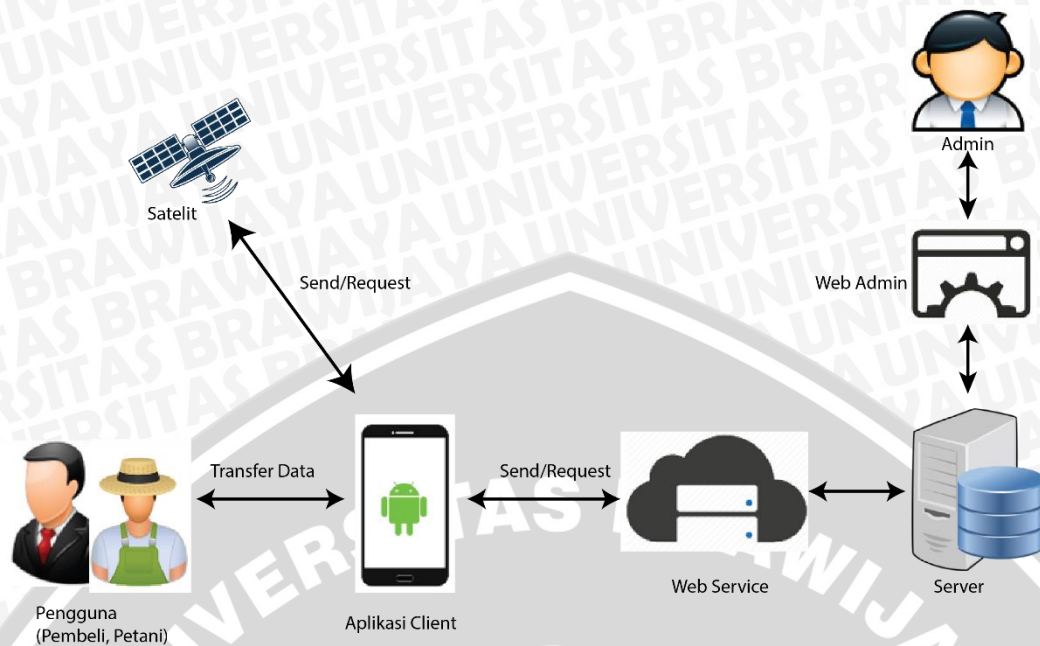
Gambar 4.14 Sequence Diagram pada skenario melaporkan pengguna

4.2.2 Perancangan arsitektur sistem

Sistem jual beli produk pertanian dibangun dengan menggunakan konsep *client* dan *server*. Perancangan arsitektur sistem dibuat untuk menjelaskan bagaimana arsitektur sistem jual beli produk pertanian akan diimplementasikan. Sistem ini diberi nama “TokoTani”. Perancangan arsitektur sistem jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) ditunjukkan pada Gambar 4.15.

Sistem *client* dibuat dengan bahasa HTML, CSS, dan Javascript dan dibangun dengan menggunakan *Ionic Framework* dan *apache cordova*, dengan memanfaatkan *AngularJS* yang dikembangkan oleh Google dan digabung dengan *NgCordova*. *NgCordova* digunakan agar aplikasi dapat mengakses *hardware native* dari perangkat bergerak, beberapa *plugin* dari *NgCordova* yang digunakan dalam penelitian ini adalah *plugin geolocation* dan *plugin camera*. Versi *Ionic Framework* yang digunakan untuk pengembangan aplikasi *client* adalah versi 1.6.4, sedangkan versi *apache cordova* yang digunakan adalah versi 5.2.0. *NgCordova* yang digunakan adalah versi 1.17. Sedangkan untuk sistem *server* dibuat dengan menggunakan bahasa pemrograman PHP, HTML, CSS, dan Javascript dengan menggunakan *Framework Panada*.

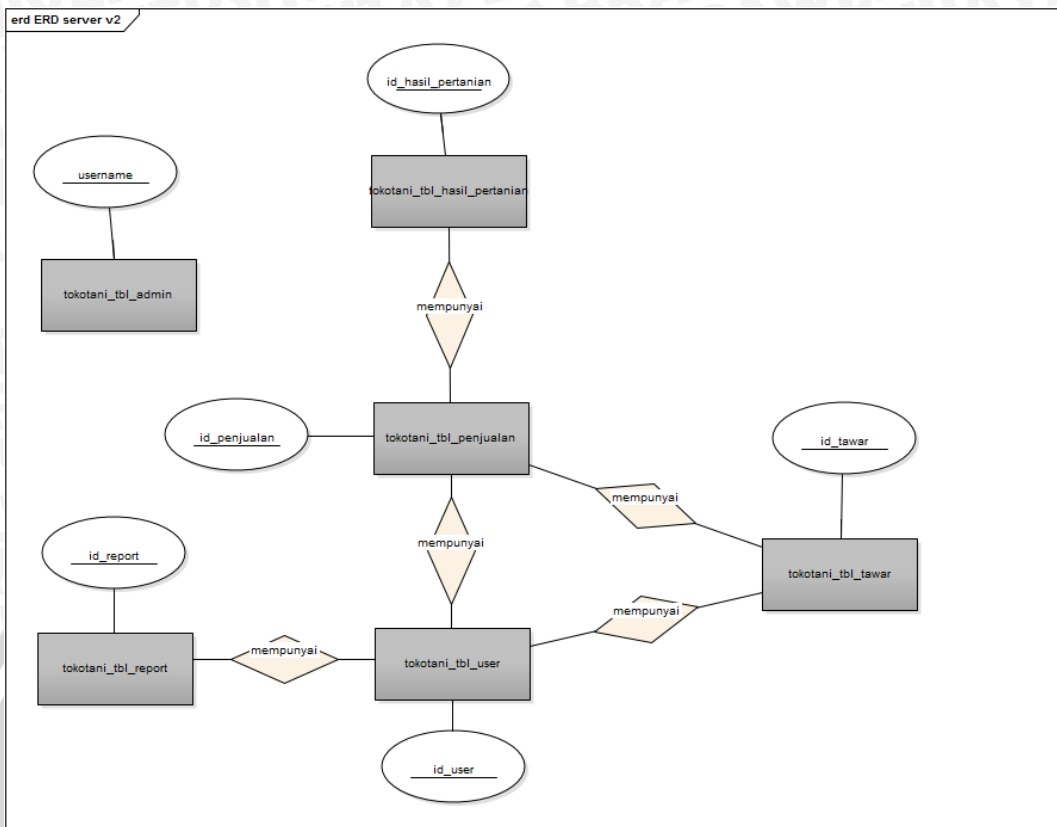
Pengguna hanya dapat mengakses aplikasi *client*, sedangkan sistem *server* hanya dapat diakses oleh admin. Komunikasi antara aplikasi *client* dan *server* dilakukan dengan menggunakan *Web Service*. Proses pertukaran data yang terjadi antara aplikasi *client* dan *server* menggunakan format JSON.



Gambar 4.15 Arsitektur aplikasi jual beli produk pertanian menggunakan LBS

4.2.3 Perancangan basis data

Perancangan basis data diperlukan sebagai gambaran bagaimana data yang diperlukan akan disimpan. Pada penelitian ini perancangan basis data direpresentasikan dalam bentuk ERD (*Entity Relationship Diagram*). ERD menunjukkan hubungan yang terjadi di antara objek (*entitas*) yang terlibat dalam suatu *database*. ERD berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan beberapa atribut yang mempresentasikan seluruh fakta yang ditinjau dari keadaan yang nyata. Pada perancangan basis data sistem ini terdapat 6 (enam) buah tabel yaitu Tabel tokotani_tbl_admin, Tabel tokotani_tbl_hasil_pertanian, Tabel tokotani_tbl_penjualan, Tabel tokotani_tbl_report, Tabel tokotani_tbl_tawar dan Tabel tokotani_tbl_user. Tabel tokotani_tbl_admin digunakan untuk menyimpan data admin yang dapat mengakses sistem server. Tabel tokotani_tbl_hasil_penjualan digunakan untuk menyimpan jenis – jenis hasil pertanian yang nantinya dipakai di tabel tokotani_tbl_penjualan. Tabel tokotani_tbl_penjualan digunakan untuk menyimpan atribut – atribut dari hasil pertanian yang dijual oleh petani. Tabel tokotani_tbl_report digunakan untuk menyimpan data pelaporan yang dikirim oleh pengguna aplikasi client. Tabel tokotani_tbl_tawar digunakan untuk menyimpan data penawaran terhadap produk pertanian yang dijual. Tabel tokotani_tbl_user digunakan untuk menyimpan data user yang dapat mengakses aplikasi client. ERD sistem jual beli produk pertanian dapat dilihat pada Gambar 4.16. Detail atribut tiap basis data dijelaskan pada Tabel 4.8.



Gambar 4.16 ERD sistem

Tabel 4.8 Detail atribut tiap basis data

Nama Tabel	Atribut	Tipe, Length
Tokotani_tbl_admin	Username (PK)	Varchar, 20
	password	Varchar, 100
	nama	Varchar, 30
Tokotani_tbl_hasil_pertanian	Id_hasil_pertanian(PK)	Varchar, 30
	keterangan	Varchar, 30
	link	Varchar,50
Tokotani_tbl_penjualan	Id_penjualan(PK)	Varchar, 30
	Id_user	Varchar, 30
	Id_hasil_pertanian	Varchar, 30
	varietas	Varchar, 30
	Luas_lahan	Varchar, 20
	harga	Int

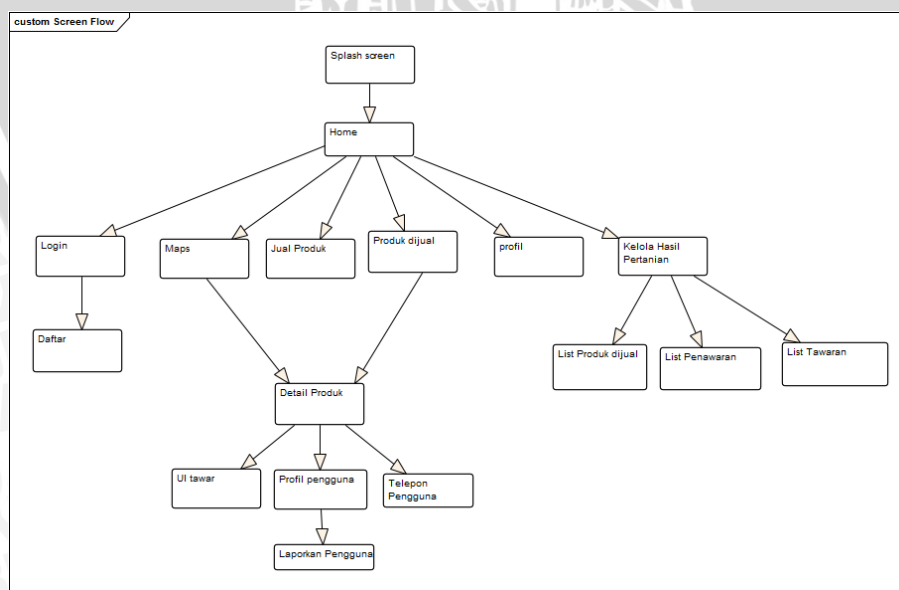
Tabel 4.8 Detail atribut tiap basis data (lanjutan)

Nama Tabel	Atribut	Tipe, Length
	jumlah	Int
	deskripsi	Varchar, 100
	Tgl_tanam	Date
	Tgl_panen	Date
	latitude	Varchar,50
	longitude	Varchar, 50
	foto	Varchar, 255
	status	Int
	Tanggal	Datetime
Tokotani_tbl_report	Id_report (PK)	Varchar, 30
	Id_pelapor	Varchar, 30
	Id_dilapor	Varchar, 30
	keterangan	Varchar, 300
	Tgl_lapor	Datetime
Tokotani_tbl_tawar	Id_tawar (PK)	Varchar, 30
	Id_user	Varchar, 30
	Id_penjualan	Varchar, 30
	harga	Int
	keterangan	Varchar, 255
	Status_pilih	Int
	Tgl_tawar	datetime
Tokotani_tbl_user	Id_user (PK)	Varchar, 30
	nama	Varchar, 30
	alamat	Varchar, 50
	lokasi	Varchar, 30
	telp	Varchar, 13
	email	Varchar, 50
	Tempat_lahir	Varchar, 30
	Tanggal_lahir	Date

Nama Tabel	Atribut	Tipe, Length
	username	Varchar, 20
	password	Varchar, 30
	url_ktp	Varchar, 255
	Is_aktif	Varchar, 11
	Is_baru	Int
	Tgl_daftar	Datetime

4.2.4 Perancangan antarmuka

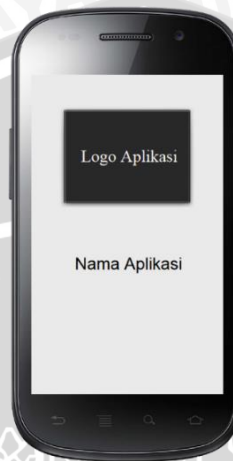
Perancangan *screen flow* ini digunakan untuk mempermudah dalam menjelaskan alur proses antarmuka dari sistem jual beli produk pertanian pada sisi *client*. Antar muka sistem jual beli produk pertanian ini akan menampilkan halaman *splash screen* terlebih dahulu untuk beberapa detik, selanjutnya sistem akan mengecek apakah user telah login ke sistem atau belum, jika user belum login maka sistem akan menampilkan halaman form login, jika user sudah login maka sistem akan menampilkan halaman Profil user sesuai dengan *id* login. Selanjutnya user dapat memilih semua menu yang ada di *sidebar*, diantaranya menu maps yang akan menampilkan peta beserta lokasi hasil pertanian yang dijual, jual produk yang akan menampilkan form untuk menjual hasil pertanian, *list* produk yang dijual yang akan menampilkan daftar hasil – hasil pertanian yang dijual oleh petani, *list* hasil pertanian user yang akan menampilkan daftar seluruh hasil pertanian yang pernah dijual oleh user (petani). Perancangan *screen flow* dari sistem jual beli produk pertanian ditunjukkan pada Gambar 4.17.



Gambar 4.17 Screen Flow Sistem Client

1. Halaman *Splash Screen*

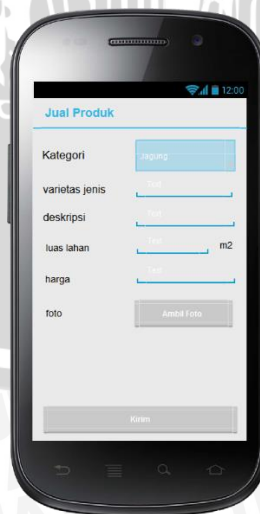
Halaman *splash screen* merupakan halaman yang tampil pertama kali saat pengguna menjalankan aplikasi jual beli produk pertanian. Pada halaman ini terdapat logo dan nama aplikasi, *splash screen* akan tertutup secara otomatis dalam beberapa detik. Antarmuka halaman *splash screen* ditunjukkan pada Gambar 4.18.



Gambar 4.18 Antarmuka halaman *Splash Screen*

2. Halaman jual produk pertanian

Halaman jual produk pertanian berfungsi agar pengguna dapat menjual hasil pertanian melalui aplikasi. Dalam antarmuka ini terdapat form yang harus diisi oleh pengguna diantaranya kategori tanaman, jenis varietas tanaman, deskripsi singkat mengenai tanaman, luas media tanam, harga yang diinginkan, tanggal tanam, perkiraan masa panen dan foto tanaman. Antarmuka halaman jual produk pertanian ditunjukkan pada Gambar 4.19.



Gambar 4.19 Antarmuka halaman jual produk pertanian

3. Halaman lihat produk dijual (*map*)

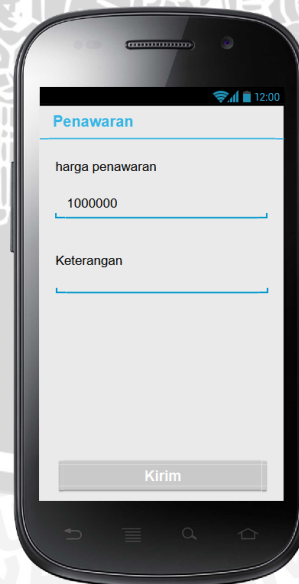
Halaman lihat produk dijual berfungsi untuk menampilkan peta (*maps*) yang menunjukkan lokasi hasil pertanian yang dijual. Peta ditampilkan dengan menggunakan google maps API. Lokasi hasil pertanian yang dijual akan ditandai dengan menggunakan marker. Antarmuka halaman *Maps* ditunjukkan pada Gambar 4.20.



Gambar 4.20 Antarmuka halaman lihat produk dijual (*map*)

4. Halaman mengajukan penawaran

Halaman mengajukan penawaran digunakan oleh calon pembeli untuk menawar hasil pertanian yang dijual oleh petani. Antarmuka halaman mengajukan penawaran ditunjukkan pada Gambar 4.21.



Gambar 4.21 Antarmuka halaman mengajukan penawaran

5. Halaman memilih penawar

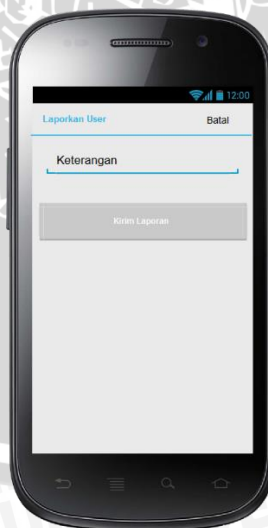
Halaman memilih penawar sama dengan Halaman detail produk pertanian, karena terdapat dalam satu tampilan halaman. Antarmuka halaman memilih penawar ditunjukkan pada Gambar 4.22.



Gambar 4.22 Antarmuka halaman memilih penawar

6. Halaman melaporkan pengguna

Halaman melaporkan pengguna digunakan untuk melaporkan pengguna yang telah menyalahi ketentuan atau tujuan dari aplikasi ini. Pengguna menekan tombol laporkan maka aplikasi akan menuju ke halaman pelaporan. Antarmuka halaman pelaporan ditunjukkan pada Gambar 4.23.



Gambar 4.23 Antarmuka halaman melaporkan pengguna

4.2.5 Perancangan komunikasi data

Perancangan komunikasi data antara aplikasi *client* dan *server* menggunakan *web service*. Proses pengiriman data dari aplikasi *client* ke *server* menggunakan format data *string* dan *file*, yang nantinya akan diterima oleh aplikasi *server* untuk disimpan di sisi *server*. Proses pengiriman data dari aplikasi *server* ke *client* menggunakan format data JSON, sehingga aplikasi *client* harus *memparsing* terlebih dahulu data yang diterima dari *server*. Tabel 4.14 menjelaskan salah satu contoh format dari antara aplikasi *client* dan *server*.

Tabel 4.14 Perancangan komunikasi data

No	Fitur Terkait	Format Data
1	Pengiriman data penjualan hasil pertanian (proses pengiriman dari aplikasi client ke server)	Kategori = " " Varietas = " " Deskripsi = " " luasLahan = " " tglTanam = " " tglPanen = " " harga = " " Lat = " " Lon = " " Foto = " "
2	Melihat hasil produk pertanian (proses pengiriman data dari server ke aplikasi client)	{ "id_produk" : " " "kategori" : " " "deskripsi" : " " "luas_lahan" : " " "tgl_tanam" : " " "tgl_panen" : " " "harga" : " " "latititude" : " " "longitude" : " " "foto" : " " }

BAB 5 IMPLEMENTASI

Pada bab ini membahas mengenai implementasi pembuatan aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Service* (LBS) berdasarkan hasil yang didapat dari analisis kebutuhan dan proses perancangan perangkat lunak. Pembahasan implementasi terdiri dari penjelasan tentang spesifikasi lingkungan implementasi, batasan-batasan implementasi, implementasi class, implementasi kode program, dan implementasi antarmuka aplikasi.

5.1 Spesifikasi sistem

Hasil dari tahap analisis kebutuhan dan perancangan sistem menjadi dasar untuk dilakukan implementasi menjadi aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Service* (LBS) agar dapat berfungsi sesuai kebutuhan. implementasi sistem bekerja pada lingkungan perangkat keras dan perangkat lunak. Perangkat keras yang digunakan pada lingkungan *client* adalah perangkat bergerak (*smartphone*).

5.1.1 Spesifikasi perangkat keras

Dalam pengembangan aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Service* (LBS) menggunakan sebuah komputer dengan spesifikasi yang dijelaskan pada Tabel 5.1

Tabel 5.1 Spesifikasi perangkat keras komputer

Nama Komponen	Spesifikasi
Model Sistem	Laptop ASUS A43SD
Prosesor	Intel® Core™ i3 2310M CPU @2.10Ghz
Memori (HDD)	700 GB
Memori (RAM)	4 GB
Kartu Grafik	NVIDIA Geforce 610M 2 GB

Proses instalasi dan pengujian perangkat yang digunakan menggunakan *smartphone* Android dengan spesifikasi perangkat keras yang ditunjukkan pada Tabel 5.2

Tabel 5.2 Spesifikasi perangkat keras *smartphone* Android

Nama Komponen	Spesifikasi
Model Sistem	ASUS Zenfone 5
Prosesor	Intel® Atom™ Multi-Core Z2560 Processor 1.6Hz Hyper-Threading Technology

Tabel 5.2 Spesifikasi perangkat keras *smartphone* Android (lanjutan)

Nama Komponen	Spesifikasi
Memori Internal	16 GB
Memori (RAM)	2 GB
Layar	5 Inchi, HD 1280x720, IPS
Kamera	8 Megapiksel
WLAN	802.11 b/g/n

5.1.2 Spesifikasi perangkat lunak

Dalam pengembangan aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Service* (LBS) menggunakan sebuah komputer dengan spesifikasi perangkat lunak yang dijelaskan pada Tabel 5.3

Tabel 5.3 Spesifikasi perangkat lunak komputer

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 8.1 Pro © 2013 64-Bit
Bahasa Pemrograman	HTML, CSS, Javascript
<i>Software Development Kit</i>	Java SE Development Kit 8 (64-Bit)
<i>Programming Environment</i>	Java Runtime Environment 8
Android SDK	API 16
Editor	Aptana Studio
<i>Browser</i>	Google Chrome Version 47.0.2526.80 m

Proses instalasi dan pengujian perangkat dilakukan pada perangkat bergerak *smartphone* Android dengan spesifikasi perangkat lunak yang ditunjukkan pada Tabel 5.4

Tabel 5.4 Spesifikasi perangkat lunak *smartphone* Android

Nama Komponen	Spesifikasi
Sistem Operasi	Android Versi 4.4.2 (Kitkat)

5.2 Batasan implementasi

Beberapa batasan dalam mengimplementasikan sistem adalah sebagai berikut:

1. Tampilan antarmuka pengguna aplikasi *server* hanya berupa *prototype*

2. Implementasi lebih difokuskan pada aplikasi *mobile*.

5.3 Implementasi *class* dan *interface* pada file program

Setiap *class* yang telah dirancang pada proses perancangan direalisasikan pada masing-masing *controller*. Tabel 5.5 menjelaskan mengenai pasangan antar *class* pada tiap *controller* pada file *controller.js* yang digunakan untuk proses implementasi pada aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS).

Tabel 5.5 Implementasi *class* pada tiap *controller*

No	Nama <i>Class</i>	Nama <i>Controller</i>	Nama <i>view</i>
1	AppCtrl	AppCtrl	Menu.html
2	LoginCtrl	LoginCtrl	Login.html
3	SignUpCtrl	SignUpCtrl	Signup.html
4	MapCtrl	MapCtrl	Map.html
5	JualCtrl	JualCtrl	jualProduk.html
6	produkDijualCtrl	produkDijualCtrl	produkDijual.html
7	DetailProdukDijualCtrl	DetailProdukDijualCtrl	detail- produkDijual.html
8	TawarCtrl	TawarCtrl	tawar.html
9	ProfilCtrl	ProfilCtrl	profil.html
10	ProfilLainCtrl	ProfilLainCtrl	profil-lain.html
11	ManageProdukCtrl	ManageProdukCtrl	manageProduk.html
12	ProdukManageCtrl	ProdukManageCtrl	manage- produk.html
13	PenawaranManageCtrl	PenawaranManageCtrl	manage- penawaran.html
14	TawaranManageCtrl	TawaranManageCtrl	manage- tawaran.html

5.4 Implementasi kode program

Aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) memiliki beberapa proses atau *method* yang terdapat pada tiap *class*. Proses yang akan dicantumkan dalam penelitian ini merupakan proses utama yang dijalankan pada aplikasi seperti proses jual produk pertanian, melihat hasil pertanian yang dijual berdasarkan lokasi menggunakan map, proses tawar produk pertanian, proses memilih penawar, dan proses melaporkan pengguna.

5.4.1 Implementasi kode program untuk proses jual produk pertanian

Pada aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) terdapat proses untuk melakukan penjualan produk. Input dari pengguna tentang detail produk yang akan dijual akan di kirim ke server. Kode program untuk proses jual produk ditunjukkan pada *Source Code* 5.1

```
1 .controller('JualCtrl', function($scope, $ionicLoading,
2 $cordovaCamera, $cordovaFileTransfer, $cordovaGeolocation,
3 kategori_hasil_pertanian) {
4     $scope.$on('$ionicView.enter', function(){
5         $scope.kategoriTanaman1 =
6         kategori_hasil_pertanian.get();
7         document.addEventListener('deviceready', function
8         () {
9             $scope.takePicture = function() {
10                 var options = {
11                     quality : 100,
12                     destinationType :
13                     Camera.DestinationType.FILE_URI,
14                     sourceType :
15                     Camera.PictureSourceType.CAMERA,
16                     allowEdit : false,
17                     encodingType: Camera.EncodingType.JPEG,
18                     targetWidth: 300,
19                     targetHeight: 300,
20                     popoverOptions: CameraPopoverOptions,
21                     saveToPhotoAlbum: true
22                 };
23
24                 $cordovaCamera.getPicture(options).then(function(imageData) {
25                     $scope.imgURI = imageData;
26                 }, function(err) {
27                     // An error occured. Show a message to the
28                     user
29                     alert('error');
30                 });
31             };
32         }, false);
33
34         $scope.btJualProduk = function(kategoriTanaman,
35         varietas, luasLahan, hargaJual, jumlah, deskripsi, tglTanam,
36         jmlHariPanen){
37             var posOptions = {timeout: 30000,
38             enableHighAccuracy: false};
39             $ionicLoading.show();
40             $cordovaGeolocation
41             .getCurrentPosition(posOptions)
42             .then(function (position) {
43                 var lat = position.coords.latitude;
44                 var lon = position.coords.longitude;
45                 alert("lat " +lat);
46                 alert("tgl "+tglTanam);
47                 var datel = new
48                 Date(tglTanam.getFullYear(), tglTanam.getMonth(),
49                 tglTanam.getDate()+1);
50                 $scope.tgl_tanam = datel;
51                 alert("th "+datel.getFullYear()+" bln "+
52                 (datel.getMonth()+ " tgl "+datel.getDate());
53                 var tglPanen = new
54                 Date(datel.getFullYear(), datel.getMonth(),
55                 datel.getDate()+jmlHariPanen);
56                 $scope.tgl panen = tglPanen;
```



```

58         alert("2 th "+tglPanen.getFullYear()+" bln
59 "+ ( tglPanen.getMonth()+ " tgl "+tglPanen.getDate());
60         var url1 =
61 "http://crevion.net/tokotani/index.php/Home/simpan_penjualan";
62         var user_profil =
63 JSON.parse(localStorage.getItem('user-profil'));
64         var targetPath = $scope.imgURI;
65         var filename =
66 targetPath.split("/").pop();
67         var formatFile =
68 targetPath.split(".").pop();
69         var options = {
70         fileKey: "file",
71         fileName: "jual."+formatFile,
72         chunkedMode: false,
73         mimeType: "image/"+formatFile,
74         params : {'lat':lat, 'lon':lon,
75 'idUser': user_profil[0].id_user, 'id_hasil_pertanian':
76 kategoriTanaman, 'varietas': varietas, 'luasLahan': luasLahan,
77 'hargaJual': hargaJual, 'jumlah': jumlah, 'deskripsi':
78 deskripsi, 'tglTanam': $scope.tgl_tanam, 'tglPanen':
79 $scope.tgl_panen}
80         };
81         $cordovaFileTransfer.upload(url1,
82 targetPath, options)
83         .then(function(result) {
84         console.log("SUCCESS: " +
85 JSON.stringify(result.response));
86         alert("sucess");
87
88 alert(JSON.stringify(result.response));
89         $state.go('app.manageProduk.Produk');
90
91         }, function(err) {
92         console.log("ERROR: " +
93 JSON.stringify(err));
94         alert(JSON.stringify(err));
95         }, function (progress) {
96
97         });
98         $ionicLoading.hide();
99         }, function(err) {
100         // error
101         $ionicLoading.hide();
102         alert("gagal mengambil posisi");
103         });
104     };
105 });
106 })
107

```

Source Code 5.1 Kode program untuk proses jual produk pertanian

Penjelasan *Source Code 5.1*, pada baris 9 – 32 merupakan method untuk mengambil gambar dengan menjalankan kamera. Proses menjalankan kamera *native device* dilakukan dengan memanfaatkan *plugin camera* dari *ngcordova*. Pada baris 35 – 105 merupakan method *btJualProduk*, method tersebut digunakan untuk mengirim data dari aplikasi ke *server*. Pada method *btJualProduk* terdapat kode untuk mengambil lokasi pengguna saat mengirim data produk dijual ke *server*, ditunjukkan pada baris 41 – 104. Aplikasi *client* memanfaatkan GPS pada *smartphone* untuk mendapatkan lokasi pengguna dengan menggunakan *plugin geolocation* dari *ngcordova*.

5.4.2 Implementasi kode program untuk melihat produk dijual (*map*)

Pada aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) terdapat proses untuk melihat produk yang dijual oleh pengguna lain. Lokasi – lokasi produk pertanian yang dijual pengguna akan ditampilkan dalam tampilan *map*. Kode program untuk proses melihat produk dijual ditunjukkan pada *Source Code 5.2*

```

1  .controller('MapCtrl', function ($scope, $ionicLoading,
2  $ionicPopup, $state, $cordovaGeolocation, $http) {
3      $scope.lokasiproduk1=[];
4
5  $http.get('http://crevion.net/tokotani/index.php/Home/get_lokasi_p
6  roduk_all/')
7      .success(function (response) {
8          for (var i=0; i < response.length; i++) {
9              $scope.lokasiproduk1[i] = {
10                 id : response[i].id_penjualan,
11                 latitude : response[i].latitude,
12                 longitude :
13 response[i].longitude,
14                 jenis : response[i].keterangan,
15                 varietas : response[i].varietas
16             };
17         };
18
19         console.log($scope.lokasiproduk1);
20         var posOptions = {timeout: 30000,
21 enableHighAccuracy: false};
22         $cordovaGeolocation
23             .getCurrentPosition(posOptions)
24             .then(function (position) {
25                 var lat = position.coords.latitude;
26                 var long = position.coords.longitude;
27                 var currentLocation = new
28 google.maps.LatLng(position.coords.latitude,
29 position.coords.longitude);
30                 var pinColor = "FFFF00";
31                 var pinImage = new
32 google.maps.MarkerImage("http://chart.apis.google.com/chart?chst=d
33 _map_pin_letter&chld=%E2%80%A2|" + pinColor,
34                 new google.maps.Size(21, 34),
35                 new google.maps.Point(0, 0),
36                 new google.maps.Point(10, 34));
37                 var pinShadow = new
38 google.maps.MarkerImage("http://chart.apis.google.com/chart?chst=d
39 _map_pin_shadow",
40                 new google.maps.Size(40, 37),
41                 new google.maps.Point(0, 0),
42                 new google.maps.Point(12, 35));
43                 var marker = new google.maps.Marker({
44 map: $scope.map,
45 position: currentLocation,
46 icon: pinImage,
47 shadow: pinShadow
48             });
49
50 marker.setAnimation(google.maps.Animation.BOUNCE);
51                 google.maps.event.addListener(marker,
52 'click', function () {
53

```

```
54     $scope.infowindow.setContent('Lokasi
55     Anda');
56     $scope.infowindow.open($scope.map,
57     this);
58         });
59     $scope.map.setCenter(currentLocation);
60     $scope.map.setZoom(30);
61
62     $scope.map.setMapTypeId(google.maps.MapTypeId.ROADMAP);
63
64     var requestRs = {
65         location: currentLocation,
66         radius: 1000,
67         keyword: 'rumah sakit'
68     };
69     var requestPuskesmas = {
70         location: currentLocation,
71         radius: 1000,
72         keyword: 'puskesmas'
73     };
74     $scope.directionsDisplay = new
75     google.maps.DirectionsRenderer({suppressMarkers: true});
76     $scope.directionsService = new
77     google.maps.DirectionsService();
78     $scope.infowindow = new
79     google.maps.InfoWindow();
80
81     var pinColor = "FE7569";
82     var pinImage = new
83     google.maps.MarkerImage("http://chart.apis.google.com/chart?chst=d
84     _map_pin_letter&chld=%E2%80%A2|" + pinColor,
85         new google.maps.Size(21, 34),
86         new google.maps.Point(0, 0),
87         new google.maps.Point(10, 34));
88     var pinShadow = new
89     google.maps.MarkerImage("http://chart.apis.google.com/chart?chst=d
90     _map_pin_shadow",
91         new google.maps.Size(40, 37),
92         new google.maps.Point(0, 0),
93         new google.maps.Point(12, 35));
94     var markers = [];
95
96     for (var i = 0; i <
97     $scope.lokasiproduk1.length; i++) {
98         var lokasi = new
99     google.maps.LatLng($scope.lokasiproduk1[i].latitude,
100     $scope.lokasiproduk1[i].longitude);
101         markers[i] = new google.maps.Marker({
102             map: $scope.map,
103             position: lokasi,
104             icon: pinImage, //isi dengan
105     iconnya ex: images/beachflag.png
106             shadow: pinShadow
107         });
108
109     console.log($scope.lokasiproduk1[i].varietas);
110
111     google.maps.event.addListener(markers[i], 'click', function () {
112         if($scope.infowindow !== void
113     0){
114             $scope.infowindow.close();
115         }
116         var infoWindowOptions =
117     {content: $scope.lokasiproduk1[i].varietas};
118     }
```



```

119         $scope.infowindow = new
120 google.maps.InfoWindow(infoWindowOptions);
121         $scope.infowindow.open($scope.map,
122 markers[i]);
123     });
124 }
125 function createMarker(place) {
126     console.log(place);
127     var placeLoc = place.geometry.location;
128     var marker = new google.maps.Marker({
129         map: $scope.map,
130         position: place.geometry.location,
131         icon: pinImage,
132         shadow: pinShadow
133     });
134
135     google.maps.event.addListener(marker,
136 'click', function () {
137
138 $scope.infowindow.setContent(place.name + '<br />' +
139 place.vicinity);
140         $scope.infowindow.open($scope.map,
141 this);
142
143 $scope.createDirection(currentLocation, placeLoc)
144     });
145 }
146
147 $scope.createDirection = function (awal,
148 tujuan) {
149     $scope.request = {
150         origin: awal,
151         destination: tujuan,
152         travelMode:
153 google.maps.TravelMode.DRIVING
154     };
155
156 $scope.directionsService.route($scope.request, function (response,
157 status) {
158     if (status ==
159 google.maps.DirectionsStatus.OK) {
160         $scope.resDir = response;
161
162 $scope.directionsDisplay.setDirections(response);
163     }
164 });
165
166 $scope.goPanel = function () {
167     $scope.isSmall = !$scope.isSmall;
168     $scope.map = map;
169 };
170 $ionicLoading.hide();
171 }, function(err) {
172     // error
173     $ionicLoading.hide();
174     $ionicPopup.alert({
175         title: "Tidak Bisa Mengakses",
176         content: "Pastikan lokasi Anda aktif :
177 " + error.message
178     })
179     .then(function (result) {
180         $state.go('app.profil');
181     });
182 });
183 }).error(function (err) {

```



```

183         res = 'error : ' + err;
184     });
185     console.log($scope.lokasiproduk1);
186     $scope.mapCreated = function (map) {
187         $scope.map = map;
188         if (!$scope.map) {
189             return;
190         }
191     };
192
193     });
194 })

```

Source Code 5.2 Kode program untuk melihat produk dijual (*map*)

Penjelasan *Source Code 5.2*, pada baris 5 – 6 merupakan *script* untuk meminta data lokasi produk pertanian dari *server*, jika aplikasi berhasil meminta data ke *server* maka akan menjalankan method *success*, namun jika aplikasi gagal meminta data maka akan menjalankan method *error*. Pada method *success*, baris 8 – 17 digunakan untuk inialisasi data dari *server* kedalam variabel array. Pada baris 22 – 23 digunakan untuk mengambil posisi pengguna saat ini dengan memanfaatkan *plugin cordova geolocation*, jika proses pengambilan posisi berhasil maka menjalankan *script* baris 24 – 169, yang mana pada baris 25 – 42 terdapat inialisasi variabel yang akan digunakan untuk proses menampilkan *marker*. Pada baris 43 – 50 digunakan untuk menampilkan *marker* posisi pengguna. Pada baris 95 – 123 digunakan untuk menampilkan posisi produk pertanian yang dijual oleh pengguna.

5.4.3 Implementasi kode program untuk mengajukan penawaran

Pada aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services (LBS)* terdapat proses untuk mengajukan penawaran terhadap produk yang dijual oleh pengguna lain. Kode program untuk proses mengajukan penawaran ditunjukkan pada *Source Code 5.3*

```

1     .controller('TawarCtrl', function ($scope, $http, $rootScope) {
2         $scope.btTawar = function(harga, ket){
3             var user =
4             JSON.parse(localStorage.getItem('user-profil'));
5             $http({
6                 method: 'POST',
7                 url:
8                 "http://crevion.net/tokotani/index.php/home/add_tawaran",
9                 data: {
10                  userId : user[0].id_user,
11                  idPenjualan: $rootScope.id_penjualan,
12                  harga : harga,
13                  keterangan : ket},
14                 headers: {'Content-Type': 'application/x-www-
15                 form-urlencoded'},
16                 transformRequest: function(obj) {
17                     var str = [];
18                     for(var p in obj)
19                         str.push(encodeURIComponent(p) + "=" +
20                     encodeURIComponent(obj[p]));
21                     return str.join("&");
22                 },
23             })
24             .success(function (response) {

```

```

25         if(!response){
26             alert('error');
27         }else{
28             alert('sukses');
29             $scope.$emit('todo:updateDataPenawar');
30         }
31     }
32
33     });
34 };
35 })

```

Source Code 5.3 Kode program untuk proses mengajukan penawaran

Penjelasan *Source Code 5.3*, terdapat fungsi *btTawar* yang akan dijalankan jika tombol Kirim pada halaman tawar produk ditekan. Pada baris 3-4 digunakan untuk inisialisasi data pengguna yang login dari *localStorage*. Pada baris 5 – 23 digunakan untuk mengirimkan data tawaran ke *server*, jika proses pengiriman dan penyimpanan data ke *server* berhasil maka akan menjalankan *script* baris 28 – 29, jika gagal menyimpan maka akan menjalankan *script* baris 26.

5.4.4 Implementasi kode program untuk memilih penawar

Pada aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) terdapat proses untuk memilih penawar. Fitur ini hanya bisa dijalankan oleh pengguna (petani) yang menjual/mengirim produk pertanian yang bersangkutan ke aplikasi. Kode program untuk proses memilih penawar ditunjukkan pada *Source Code 5.4*

```

1     $scope.pilihpenawar = function(id){
2         $http({
3             method: 'POST',
4             url:
5             "http://crevion.net/tokotani/index.php/home/pilih_penawar",
6             data: {
7                 idtawar : id
8             },
9             headers: {'Content-Type':
10            'application/x-www-form-urlencoded'},
11            transformRequest: function(obj)
12            {
13                var str = [];
14                for(var p in obj)
15
16                str.push(encodeURIComponent(p) + "=" +
17                encodeURIComponent(obj[p]));
18
19                return str.join("&");
20            },
21            .success(function (response) {
22                if(!response){
23                    alert('error');
24                }else{
25                    alert('sukses');
26
27                    $scope.$emit('todo:updateDataPenawar');
28                }
29            });
30        });
31    };

```

Source Code 5.4 Kode program untuk memilih penawar

Penjelasan *Source Code* 5.4, terdapat fungsi pilihpenawar yang akan dijalankan jika tombol konfirmasi penawar yang ada pada halaman detail produk pertanian ditekan. Pada baris 2 – 20 digunakan untuk mengirim data ke *server* dan nantinya *server* akan menyimpan data ke *database*. Jika proses pengiriman dan penyimpanan data ke *database* berhasil maka akan muncul pemberitahuan bahwa proses berhasil, namun jika gagal maka akan muncul pemberitahuan bahwa proses gagal.

5.4.5 Implementasi kode program untuk melaporkan pengguna

Pada aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) terdapat proses untuk proses melaporkan pengguna aplikasi yang melakukan pelanggaran aturan. Kode program untuk proses melaporkan pengguna ditunjukkan pada *Source Code* 5.5

```

1      $ionicModal.fromTemplateUrl('templates/modalLapor.html', {
2          scope: $scope
3      }).then(function(modal) {
4          $scope.modalLapor = modal;
5      });
6
7      $scope.laporUser = function(idDilapor){
8          $scope.idlapor = idDilapor;
9          $scope.modalLapor.show();
10     };
11
12     $scope.batalLapor = function(){
13         $scope.modalLapor.hide();
14     };
15
16     $scope.kirimLaporan = function(keterangan){
17         $ionicLoading.show({
18             template: 'Loading...'
19         });
20         $http({
21             method: 'POST',
22             url:
23             "http://crevion.net/tokotani/index.php/home/lapor_user",
24             timeout: 60000,
25             data: {
26                 id_pelapor :
27                 $scope.dat[0].id_user,
28                 id_dilapor : $scope.idlapor,
29                 keterangan : keterangan
30             },
31             headers: {'Content-Type':
32             'application/x-www-form-urlencoded'},
33             transformRequest: function(obj) {
34                 var str = [];
35                 for(var p in obj)
36                     str.push(encodeURIComponent(p) +
37                 "=" + encodeURIComponent(obj[p]));
38                 return str.join("&");
39             },
40             // data: {user_id:
41             $scope.loginData.username, pass: $scope.loginData.password}
42         })
43         .success(function (response) {
44             $ionicLoading.hide();
45             if(!response){
46                 alert('error');

```



```

47     }else{
48         alert('Berhasil, data akan
49         diverifikasi oleh Admin');
50         $state.go('app.profil');
51     }
52
53     })
54     .error(function() {
55         $ionicLoading.hide();
56         alert('Gagal Melaporkan, Periksa
57         koneksi anda');
58     });
59     };

```

Source Code 5.5 Kode program untuk melaporkan pengguna

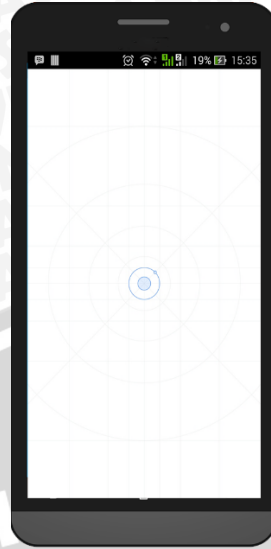
Penjelasan *Source Code 5.5*, pada baris 1 – 5 dilakukan inisialisasi *modal* untuk halaman pelaporan. Pada baris 7 – 10 merupakan fungsi untuk menampilkan halaman *modal* pelaporan. Pada baris 12 – 14 merupakan fungsi untuk menutup halaman *modal* pelaporan. Pada baris 16 – 59 merupakan fungsi kirimLaporan untuk mengirim data ke *server*. Jika *server* telah berhasil menyimpan data ke *database*, maka akan menjalankan *script* baris 48–51, jika gagal menyimpan maka akan menjalankan *script* baris 46. Jika aplikasi gagal mengirimkan data ke *server* maka akan menjalankan *script* baris 55 – 57.

5.5 Implementasi Antarmuka Pengguna

Pada tahap ini dijelaskan mengenai berbagai implementasi antarmuka pengguna yang telah diimplementasikan berdasarkan dari perancangan antarmuka yang telah dilakukan. Beberapa diantaranya yaitu implementasi halaman jual produk pertanian, implementasi halaman melihat produk dijual (*map*), implementasi halaman mengajukan penawaran, implementasi halaman memilih penawar, dan implementasi halaman melaporkan pengguna.

5.5.1 Implementasi Halaman *Splash Screen*

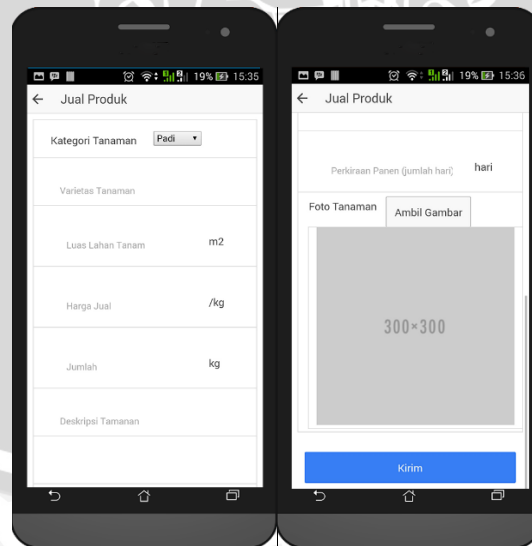
Halaman *splash screen* merupakan tampilan awal ketika pengguna membuka sistem. *Splash screen* pada aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) berisi logo aplikasi berwarna putih. Halaman *splash screen* akan tampil dalam 5 detik, dan selanjutnya akan otomatis tertutup. Implementasi antarmuka halaman *splash screen* ditunjukkan pada Gambar 5.1.



Gambar 5.1 Implementasi halaman *Splash Screen*

5.5.2 Implementasi halaman jual produk pertanian

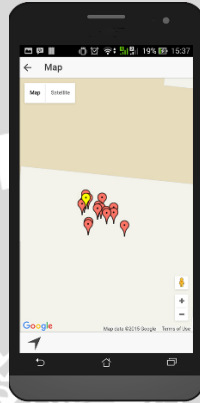
Halaman jual produk merupakan halaman yang digunakan pengguna untuk mengirim data produk pertanian untuk dijual melalui aplikasi. Terdapat beberapa *textfield* yang harus diisi oleh pengguna ketika akan menjual produk pertanian melalui aplikasi diantaranya kategori tanaman, varietas tanaman, luas lahan tanam, harga jual, jumlah, deskripsi tanaman, tanggal tanam, perkiraan panen, dan foto tanaman. Terdapat juga tombol Ambil Gambar yang berfungsi menjalankan *camera* pada *smartphone*. Implementasi antarmuka halaman jual produk pertanian ditunjukkan pada Gambar 5.2.



Gambar 5.2 Implementasi halaman jual produk dijual

5.5.3 Implementasi halaman melihat produk dijual (*map*)

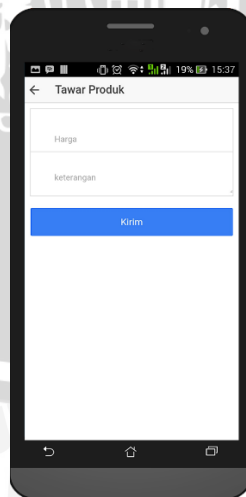
Halaman melihat produk dijual digunakan untuk menampilkan lokasi – lokasi produk yang dijual dalam peta (*map*) Google. Dalam halaman ini ditampilkan lokasi pengguna saat ini dan produk dijual yang berada disekitar pengguna. Lokasi pengguna saat ini akan ditunjukkan oleh *marker* berwarna kuning, sedangkan lokasi produk pertanian yang dijual ditunjukkan dengan *marker* berwarna merah. Implementasi antarmuka halaman melihat produk dijual ditunjukkan pada Gambar 5.3.



Gambar 5.3 Implementasi halaman melihat produk dijual (*map*)

5.5.4 Implementasi halaman mengajukan penawaran

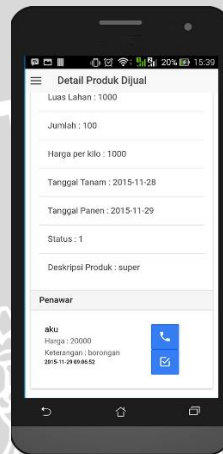
Halaman mengajukan penawaran digunakan oleh pengguna untuk menawar produk pertanian yang dijual oleh pengguna lain. Pada halaman penawaran terdapat *form* yang harus diisi oleh pengguna, yaitu harga yang diajukan dan keterangan penawaran, terdapat pula tombol Kirim untuk mengirim data yang telah diinputkan pengguna ke *server*. Implementasi antarmuka halaman mengajukan penawaran ditunjukkan pada Gambar 5.4.



Gambar 5.4 Implementasi halaman mengajukan penawaran

5.5.5 Implementasi antarmuka memilih penawar

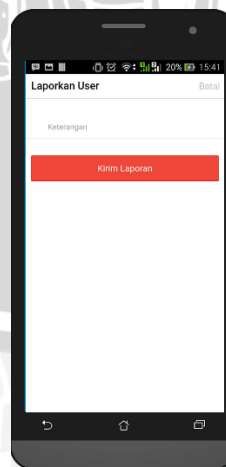
Antarmuka memilih penawar digunakan oleh pengguna (petani) untuk memilih penawar yang sesuai. Antarmuka memilih penawar hanya dapat diakses jika pengguna melihat detail produk pertanian yang telah diunggah oleh pengguna yang sama ke aplikasi, jika pengguna melihat detail produk pertanian milik pengguna lain maka antarmuka memilih penawar tidak dapat diakses. Proses memilih penawar dapat dilakukan dengan menekan tombol konfirmasi (tombol warna biru dengan ikon v). Implementasi antarmuka memilih penawar ditunjukkan pada Gambar 5.5.



Gambar 5.5 Implementasi antarmuka memilih penawar

5.5.6 Implementasi halaman melaporkan pengguna

Halaman melaporkan pengguna digunakan oleh pengguna untuk melaporkan pengguna yang melakukan pelanggaran aturan. Halaman melaporkan pengguna hanya dapat diakses jika pengguna melihat profil pengguna lain. Pada halaman melaporkan pengguna terdapat *field* keterangan yang harus diisi pengguna yang akan lapor. Implementasi melaporkan pengguna ditunjukkan pada Gambar 5.6.



Gambar 5.6 Implementasi halaman melaporkan pengguna

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian dan analisis perangkat bergerak aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) yang telah diimplementasikan sebelumnya. Proses pengujian dilakukan melalui dua jenis pengujian yaitu pengujian validasi dan pengujian *usability*.

6.1 Pengujian

Pengujian bertujuan khusus untuk menemukan kesalahan dan segala kemungkinan yang akan menimbulkan kesalahan sesuai dengan spesifikasi perangkat bergerak yang telah ditentukan sebelum sistem diserahkan kepada *client*. Pengujian menunjukkan bahwa fungsi perangkat lunak bekerja sesuai spesifikasi dan bahwa persyaratan kinerja telah dipenuhi. Pada aplikasi yang dibuat dalam penelitian ini dilakukan pengujian berupa pengujian validasi dan pengujian *usability*.

6.1.1 Pengujian validasi

Pengujian validasi digunakan untuk mengetahui apakah aplikasi yang dibangun sudah sesuai dengan kebutuhan. Daftar kebutuhan yang telah dirumuskan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *black-box* karena tidak diperlukan konsentrasi terhadap alur jalannya kode program dan lebih ditekankan untuk menemukan kesalahan antara kinerja aplikasi dengan daftar kebutuhan. Kasus uji yang dilakukan terhadap aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) ditunjukkan pada Tabel 6.1 sampai Tabel 6.5.




6.1.1.1 Kasus uji menjual produk pertanian

Proses menjual produk pertanian merupakan proses yang digunakan untuk mengirim data produk pertanian ke server, dan selanjutnya server akan menyimpan data yang dikirim melalui aplikasi client ke dalam database. Aplikasi harus dapat mengirim data ke server dan nantinya server harus dapat menyimpan data ke database. Proses ini harus diuji untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam mengirim data ke server dan disimpan ke database server. Kasus uji untuk menjalankan proses jual produk pertanian ditunjukkan pada Tabel 6.1.

Tabel 6.1 Kasus uji menjual produk pertanian

Nama Kasus Uji	Kasus Uji Menjual Produk Pertanian
Objek Uji	KF-005

Tabel 6.1 Kasus uji menjual produk pertanian (lanjutan)

Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam mengirim data ke server dan disimpan ke database server.
Kondisi Awal	Aplikasi telah masuk ke halaman Jual Produk
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Aplikasi telah menampilkan halaman Jual Produk. 2. Pengguna mengisi <i>form</i> yang tersedia. 3. Pengguna menekan tombol Ambil Gambar, dan aplikasi menjalankan kamera. 4. Pengguna mengambil gambar produk pertanian yang akan di unggah. 5. Pengguna menekan tombol Kirim.
Hasil yang diharapkan	Aplikasi dapat mengirim data ke server dan data disimpan ke database server.
Hasil	<ol style="list-style-type: none"> 1. Aplikasi menampilkan halaman jual produk  2. Aplikasi menampilkan pesan sukses dan data berhasil disimpan ke database server.  
Status Validasi	Valid

6.1.1.2 Kasus uji melihat produk dijual (*map*)

Proses melihat produk dijual dalam bentuk visualisasi *map* merupakan proses untuk melihat produk pertanian yang dijual melalui aplikasi berdasarkan lokasi tiap produk pertanian. Seluruh lokasi pertanian ditampilkan dalam bentuk visual *map*. Aplikasi harus dapat mendapatkan dan menampilkan lokasi pengguna dan lokasi produk pertanian yang dijual melalui sistem. Halaman ini harus diuji untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dimana



aplikasi dapat menampilkan lokasi pengguna dan lokasi produk pertanian disekitar pengguna. Kasus uji untuk melihat produk dijual ditunjukkan pada Tabel 6.2.



Tabel 6.2 Kasus uji melihat produk dijual


Nama Kasus Uji	Kasus Uji Melihat Produk Dijual
Objek Uji	KF-002
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dimana aplikasi dapat menampilkan lokasi pengguna dan lokasi produk pertanian disekitar pengguna.
Kondisi Awal	Aplikasi telah melakukan login
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan menu <i>Map</i> yang berada pada <i>side menu</i> 2. Aplikasi masuk ke halaman <i>Map</i>.
Hasil yang diharapkan	Aplikasi dapat menampilkan lokasi pengguna dengan tanda <i>marker</i> kuning dan lokasi produk pertanian dengan tanda <i>marker</i> merah.
Hasil	<ol style="list-style-type: none"> 1. Menampilkan halaman <i>Map</i> dan dapat menampilkan lokasi pengguna dengan menggunakan <i>marker</i> kuning, dan lokasi produk pertanian dengan menggunakan <i>marker</i> merah. 
Status Validasi	Valid

6.1.1.3 Kasus uji mengajukan penawaran

Proses mengajukan penawaran merupakan proses dimana pengguna dapat mengirimkan penawaran terhadap produk pertanian yang dipilih. Pengguna hanya bisa menawar produk pertanian yang bukan miliknya sendiri. Fitur mengajukan penawaran juga harus dapat mengirim data penawaran ke *server* dan menyimpan ke *database*. Halaman ini harus diuji untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dimana pengguna dapat menawar sebuah produk pertanian kemudian aplikasi mengirim data ke *server* dan menyimpan ke *database server*. Kasus uji untuk mengajukan penawaran ditunjukkan pada Tabel 6.3.

Tabel 6.3 Kasus uji mengajukan penawaran

Nama Kasus Uji	Kasus Uji Mengajukan Penawaran
Objek Uji	KF-007
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dimana pengguna dapat menawar sebuah produk pertanian kemudian aplikasi mengirim data ke server dan menyimpan ke database server.
Kondisi Awal	Aplikasi telah masuk ke halaman detail produk pertanian yang dijual.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka halaman detail produk pertanian miliknya sendiri 2. Pengguna membuka halaman detail produk pertanian yang bukan milik sendiri. 3. Aplikasi menampilkan halaman tawar. 4. Pengguna mengisi <i>form</i> yang tersedia 5. Pengguna menekan tombol kirim, aplikasi mengirim data ke server dan menyimpan ke database.
Hasil yang diharapkan	Aplikasi dapat mengirim data ke server dan menyimpan data ke database.
Hasil	<ol style="list-style-type: none"> 1. Tombol tawar hanya muncul ketika membuka detail produk pertanian yang bukan milik sendiri. -melihat detail produk pertanian milik sendiri  -melihat detail produk pertanian yang bukan milik sendiri  2. Aplikasi berhasil mengirim data ke server dan berhasil menyimpan data ke database.

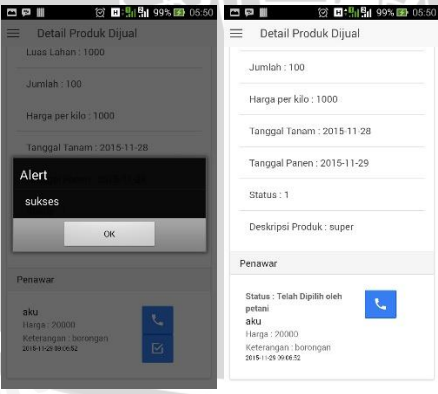
	
Status Validasi	Valid

6.1.1.4 Kasus uji memilih penawar

Proses memilih penawar merupakan proses dimana pengguna dapat memilih penawar yang telah menawar produk pertaniannya. Halaman ini harus diuji untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dimana pengguna dapat memilih penawar yang sesuai dengan kriteria kemudian aplikasi mengirim data ke *server* dan menyimpan ke *database server*. Kasus uji untuk memilih penawar ditunjukkan pada Tabel 6.4.

Tabel 6.4 Kasus uji memilih penawar

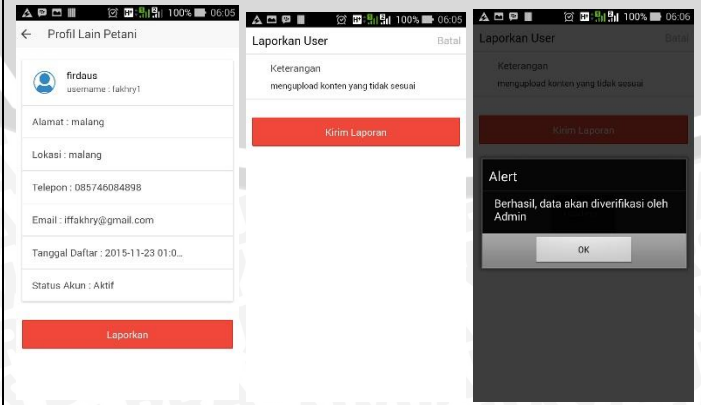
Nama Kasus Uji	Kasus Uji Memilih Penawar
Objek Uji	KF-008
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dimana pengguna dapat memilih penawar yang sesuai dengan kriteria kemudian aplikasi dapat mengirim data ke server dan menyimpan ke database server.
Kondisi Awal	Aplikasi telah masuk ke halaman detail produk pertanian yang dijual.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka halaman detail produk pertanian miliknya sendiri 2. Pengguna membuka halaman detail produk pertanian yang bukan milik sendiri. 3. Aplikasi memilih tombol konfirmasi penawar di salah satu penawar produk pertanian miliknya.
Hasil yang diharapkan	Pengguna dapat memilih penawar dan Aplikasi dapat mengirim data penawar yang dipilih ke server dan menyimpan data ke database.
Hasil	<ol style="list-style-type: none"> 1. Tombol tawar hanya muncul ketika membuka detail produk pertanian yang bukan milik sendiri.

	<p>-melihat detail produk pertanian yang bukan milik sendiri</p>  <p>-melihat detail produk pertanian milik sendiri</p>  <p>2. Pengguna dapat memilih penawar dan Aplikasi berhasil mengirim data ke server dan berhasil menyimpan data ke database.</p>  <p>Status Validasi</p> <p>Valid</p>
--	---

6.1.1.5 Kasus uji melaporkan pengguna

Proses melaporkan pengguna merupakan proses dimana pengguna dapat melaporkan pengguna lain yang melanggar aturan atau tidak sesuai dengan tujuan aplikasi. Data pelaporan yang dikirimkan oleh pengguna akan disimpan ke *database*, yang selanjutnya akan dianalisis oleh Admin. Halaman ini harus diuji untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dimana pengguna dapat melaporkan pengguna aplikasi kemudian aplikasi mengirim data pelaporan ke *server* dan menyimpan ke *database server*. Kasus uji untuk melaporkan pengguna ditunjukkan pada Tabel 6.5.

Tabel 6.5 Kasus uji melaporkan pengguna

Nama Kasus Uji	Kasus Uji Melaporkan Pengguna
Objek Uji	KF-010
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dimana pengguna dapat melaporkan pengguna aplikasi kemudian aplikasi mengirim data pelaporan ke server dan menyimpan ke database server.
Kondisi Awal	Aplikasi telah masuk ke profil pengguna yang akan dilaporkan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan tombol Laporkan pada profil orang yang akan dilaporkan. 2. Masuk ke halaman pelaporan. 3. Pengguna mengisi keterangan pelaporan dan menekan tombol kirim. 4. Menampilkan status pelaporan.
Hasil yang diharapkan	Pengguna dapat melaporkan pengguna yang melanggar aturan.
Hasil	<ol style="list-style-type: none"> 1. Pelaporan dari pengguna berhasil dikirim ke server dan berhasil disimpan ke database. 

Status Validasi	Valid
-----------------	-------

6.1.2 Pengujian *usability*

Pengujian *usability* digunakan untuk mengkaji seberapa puas dan mudah penggunaan aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) oleh pengguna baik dari kalangan petani maupun masyarakat umum. Pengujian ini dilakukan dengan cara menyebarkan kuisisioner kepada calon pengguna yakni petani dan masyarakat umum sebanyak 10 responden secara acak, dengan kriteria responden 5(lima) petani dan 5(lima) masyarakat umum sebagai konsumen. Jenis kuesioner yang digunakan mengacu pada kuesioner USE. Pertanyaan yang diajukan dalam kuesioner ini mencakup daya guna aplikasi, kemudahan penggunaan, kemudahan dalam mempelajari aplikasi dan kepuasan pengguna terhadap aplikasi. Hasil dari pengujian *usability* ditunjukkan pada Tabel 6.6.

Tabel 6.6 Hasil pengujian *usability*

No	Pertanyaan	SS	S	KS	TS	STS	Jumlah
<i>Usefulness</i>							
1	Aplikasi ini membantu saya menjadi lebih efektif memasarkan dan mencari produk pertanian	6	4	0	0	0	10
2	Penggunaan aplikasi ini membuat saya menjadi lebih produktif dan dapat mengerjakan pekerjaan lain.	5	3	2	0	0	10
3	Aplikasi ini sangat berguna bagi saya.	5	4	1	0	0	10
4	Aplikasi ini memberikan saya kontrol lebih dalam memasarkan dan mencari produk pertanian yang saya inginkan.	4	4	2	0	0	10
5	Aplikasi ini membuat saya lebih mudah dalam melakukan penjualan produk pertanian maupun pencarian produk pertanian.	5	5	0	0	0	10
6	Dengan aplikasi ini, saya dapat menghemat waktu untuk	6	4	0	0	0	10

	memasarkan dan mencari produk pertanian.						
7	Aplikasi ini memenuhi kebutuhan saya untuk memasarkan dan mencari produk pertanian.	4	6	0	0	0	10
8	Aplikasi ini sesuai dengan yang saya harapkan.	4	5	1	0	0	10
<i>Ease of use</i>							
9	Aplikasi ini mudah digunakan.	3	5	1	1	0	10
10	Aplikasi ini dapat digunakan dengan sederhana.	3	5	2	0	0	10
11	Aplikasi ini dapat digunakan oleh semua kalangan masyarakat.	3	4	3	0	0	10
12	Aplikasi ini memerlukan langkah-langkah sederhana untuk dapat mencapai apa yang saya inginkan.	3	6	1	0	0	10
13	Aplikasi ini dapat digunakan secara fleksibel.	4	6	0	0	0	10
14	Tidak membutuhkan banyak usaha untuk menggunakan aplikasi ini.	4	4	2	0	0	10
15	Saya dapat menggunakan aplikasi ini tanpa instruksi tertulis	5	5	0	0	0	10
16	Saya tidak menemukan inkonsistensi (ketidakesuaian) selama menggunakan aplikasi ini.	4	5	1	0	0	10
17	Baik pengguna reguler maupun yang bukan reguler, akan menyukai aplikasi ini.	4	5	1	0	0	10
18	Saya dapat memperbaiki kesalahan penggunaan aplikasi dengan cepat dan mudah.	3	6	1	0	0	10
19	Saya dapat menggunakan aplikasi ini dengan praktis setiap hari.	6	4	0	0	0	10

Ease of Learning							
20	Saya dapat belajar menggunakan aplikasi ini dengan cepat.	5	5	0	0	0	10
21	Saya mudah mengingat bagaimana menggunakan aplikasi ini.	6	4	0	0	0	10
22	Aplikasi ini dapat saya gunakan dengan mudah.	5	4	1	0	0	10
23	Saya dengan cepat menguasai aplikasi ini.	4	6	0	0	0	10
Satisfaction							
24	Saya puas dengan aplikasi ini.	3	5	2	0	0	10
25	Saya akan merekomendasikan aplikasi ini kepada orang lain.	4	6	0	0	0	10
26	Aplikasi ini menyenangkan untuk digunakan.	4	5	0	1	0	10
27	Aplikasi ini bekerja seperti yang saya inginkan.	4	5	1	0	0	10
28	Aplikasi ini sangat bagus.	4	5	1	0	0	10
29	Saya ingin memiliki aplikasi ini.	5	5	0	0	0	10
30	Aplikasi ini nyaman digunakan.	6	4	0	0	0	10

Keterangan:

STS : Sangat Tidak Setuju S : Setuju
 TS : Tidak Setuju SS : Sangat Setuju
 N : Netral

6.2 Analisis

Analisis terdiri dari analisis pengujian validasi dan analisis pengujian *usability*. Analisis dilakukan untuk mempermudah dalam mendapatkan kesimpulan dari tugas akhir rancang bangun aplikasi *mobile* jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS).

6.2.1 Analisis hasil pengujian validasi

Analisis terhadap hasil pengujian validasi dilakukan dengan melihat kesesuaian antara hasil yang diharapkan dengan hasil dari aplikasi. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) telah memenuhi analisis dan perancangan sistem.

6.2.2 Analisis hasil pengujian *usability*

Setelah dilakukan pengujian *usability* kepada 10 responden, hasil yang didapatkan adalah berupa akumulasi poin pada tiap aspek yang ditunjukkan pada Terdapat 5 poin, yaitu poin 1 yang berarti sangat tidak setuju, point 2 yang berarti tidak setuju, poin 3 yang berarti netral, poin 4 yang berarti setuju, dan poin 5 yang berarti sangat setuju sesuai dengan Tabel 2.1. Aplikasi dinyatakan memenuhi indeks dari pengujian *usability* jika rata-rata presentase yang dihasilkan dari setiap kriteria lebih dari 60% sesuai pada interpretasi skor yang ditunjukkan pada Tabel 6.7. Hasil perhitungan dari pengujian *usability* ditunjukkan pada Tabel 6.8. Hasil status pengujian *usability* ditunjukkan pada Tabel 6.9.

Tabel 6.7 Interpretasi skor Likert

Skor Likert	Interpretasi skor dengan interval = 20	Pilihan
1	0% - 19,99%	Sangat tidak setuju
2	20% - 39,99%	Tidak setuju
3	40% - 59,99%	Netral
4	60% - 79,99%	Setuju
5	80% - 100%	Sangat setuju

Keterangan:

Interval = 20 didapatkan dari hasil pembagian jumlah skor Likert dengan nilai 100.

Interpretasi skor Likert diterapkan untuk mendapatkan indeks presentase pengujian *usability* dari 30 pertanyaan yang ditunjukkan pada Tabel 6.6. Dari Persamaan 2.1, Persamaan 2.2 dan Persamaan 2.3 maka didapatkan nilai indeks presentase skor Likert untuk masing – masing pertanyaan berikut merupakan contoh perhitungan indeks presentase skor Likert untuk pertanyaan nomor 1.

$$\begin{aligned} \text{Total Skor} &= \sum \text{STS} \times 1 + \sum \text{TS} \times 2 + \sum \text{N} \times 3 + \sum \text{S} \times 4 + \sum \text{SS} \times 5 \\ &= (0 \times 1) + (0 \times 2) + (0 \times 3) + (4 \times 4) + (6 \times 5) \\ &= 46 \end{aligned}$$

$$\begin{aligned} Y &= \text{Skor Likert tertinggi} \times \text{jumlah responden} \\ &= 5 \times 10 \\ &= 50 \end{aligned}$$

$$\begin{aligned} \text{Indeks (\%)} &= \frac{\text{Total Skor}}{Y} \times 100\% \\ &= (46 / 50) \times 100\% \\ &= 92\% \end{aligned}$$

Tabel 6.8 Hasil perhitungan pengujian *usability*

No	Pertanyaan	SS	S	KS	TS	STS	Total Skor	Indeks (%)
<i>Usefulness</i>								
1	Aplikasi ini membantu saya menjadi lebih efektif	6	4	0	0	0	46	92

	memasarkan dan mencari produk pertanian							
2	Penggunaan aplikasi ini membuat saya menjadi lebih produktif dan dapat mengerjakan pekerjaan lain.	5	3	2	0	0	43	86
3	Aplikasi ini sangat berguna bagi saya.	5	4	1	0	0	44	88
4	Aplikasi ini memberikan saya kontrol lebih dalam memasarkan dan mencari produk pertanian yang saya inginkan.	4	4	2	0	0	42	84
5	Aplikasi ini membuat saya lebih mudah dalam melakukan penjualan produk pertanian maupun pencarian produk pertanian.	5	5	0	0	0	45	90
6	Dengan aplikasi ini, saya dapat menghemat waktu untuk memasarkan dan mencari produk pertanian.	6	4	0	0	0	46	92
7	Aplikasi ini memenuhi kebutuhan saya untuk memasarkan dan mencari produk pertanian.	4	6	0	0	0	44	88
8	Aplikasi ini sesuai dengan yang saya harapkan.	4	5	1	0	0	43	86
Rata – rata indeks <i>usefulness</i>								88,25%
<i>Ease of use</i>								
9	Aplikasi ini mudah digunakan.	3	5	1	1	0	40	80
10	Aplikasi ini dapat digunakan dengan sederhana.	3	5	2	0	0	41	82
11	Aplikasi ini dapat digunakan oleh semua kalangan masyarakat.	3	4	3	0	0	40	80
12	Aplikasi ini memerlukan langkah-langkah sederhana untuk dapat mencapai apa yang saya inginkan.	3	6	1	0	0	42	84

13	Aplikasi ini dapat digunakan secara fleksibel.	4	6	0	0	0	44	88
14	Tidak membutuhkan banyak usaha untuk menggunakan aplikasi ini.	4	4	2	0	0	42	84
15	Saya dapat menggunakan aplikasi ini tanpa instruksi tertulis	5	5	0	0	0	45	90
16	Saya tidak menemukan inkonsistensi (ketidakesesuaian) selama menggunakan aplikasi ini.	4	5	1	0	0	43	86
17	Baik pengguna reguler maupun yang bukan reguler, akan menyukai aplikasi ini.	4	5	1	0	0	43	86
18	Saya dapat memperbaiki kesalahan penggunaan aplikasi dengan cepat dan mudah.	3	6	1	0	0	42	84
19	Saya dapat menggunakan aplikasi ini dengan praktis setiap hari.	6	4	0	0	0	46	92
Rata – rata <i>Easy of use</i>								85,09%
<i>Ease of Learning</i>								
20	Saya dapat belajar menggunakan aplikasi ini dengan cepat.	5	5	0	0	0	45	90
21	Saya mudah mengingat bagaimana menggunakan aplikasi ini.	6	4	0	0	0	46	92
22	Aplikasi ini dapat saya gunakan dengan mudah.	5	4	1	0	0	44	88
23	Saya dengan cepat menguasai aplikasi ini.	4	6	0	0	0	44	88
Rata – rata <i>Easy of Learning</i>								89,5%
<i>Satisfaction</i>								
24	Saya puas dengan aplikasi ini.	3	5	2	0	0	41	82
25	Saya akan merekomendasikan aplikasi ini kepada orang lain.	4	6	0	0	0	44	88

26	Aplikasi ini menyenangkan untuk digunakan.	4	5	0	1	0	42	84
27	Aplikasi ini bekerja seperti yang saya inginkan.	4	5	1	0	0	43	86
28	Aplikasi ini sangat bagus.	4	5	1	0	0	43	86
29	Saya ingin memiliki aplikasi ini.	5	5	0	0	0	45	90
30	Aplikasi ini nyaman digunakan.	6	4	0	0	0	46	92
Rata – rata <i>Satisfaction</i>								86,85%

Tabel 6.8 Hasil status pengujian *usability*

Aspek Penilaian	Rata-Rata Presentase (%)	Status
<i>Usefulness</i>	88.25	Sangat setuju
<i>Ease of use</i>	85.09	Sangat setuju
<i>Ease of learning</i>	89.5	Sangat setuju
<i>Satisfaction</i>	86.85	Sangat setuju

Perhitungan presentase rata-rata dari semua kriteria
 = jumlah total presentase (%) / jumlah kriteria
 = $(88,25 + 85,09 + 89,5 + 86,85) / 4$
 = $350,26 / 4$
 = 87.6 %

Dari hasil pengujian *usability* yang dilakukan menunjukka bahwa aplikasi memenuhi empat kriteria yang ada dengan rata – rata presentase semua kriteria sebesar 87.6 %, sehingga menunjukkan bahwa aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) memenuhi kriteria *usability*.

BAB 7 PENUTUP

Bab ini berisi kesimpulan dari penelitian dan saran bagi pembaca. Kesimpulan berisi analisis hasil pengujian untuk menjawab rumusan masalah yang telah dijelaskan pada tahap pendahuluan. Saran ditujukan bagi pembaca yang ingin melanjutkan penelitian ke tahap yang lebih baik.

7.1 Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi, dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Perancangan dan implementasi aplikasi jual beli produk pertanian menggunakan pendekatan *Location Based Services* (LBS) telah dibuat sesuai dengan spesifikasi kebutuhan yang telah dianalisis.
2. Berdasarkan hasil pengujian validasi pada aplikasi menunjukkan nilai dengan prosentasi 100%. Sistem sudah memenuhi spesifikasi kebutuhan yang telah dianalisis. Sedangkan tingkat *usability* dari aplikas berdasarkan aspek pengujian *usability* yaitu *usefulness*, *ease of use*, *ease of learning*, dan *satisfaction* menunjukkan rata-rata sebesar 87,6 % , yang berarti aplikasi jual beli produk pertanian ini telah terbukti dapat mempermudah petani untuk memasarkan produk pertanian dan proses pencarian produk pertanian oleh konsumen.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi produk pertanian menggunakan pendekatan *Location Based Services* (LBS) selanjutnya antara lain:

1. Untuk pengembangan lebih lanjut, aplikasi ini dapat dikembangkan dengan menambahkan fitur pembayaran secara terpusat, bukan antar pengguna.
2. Pada pengembangan lebih lanjut perlu ditambahkan fitur simpan data penjualan secara *offline*. Hal ini bertujuan untuk mempermudah dalam proses penyimpanan produk pertanian yang akan dijual.

DAFTAR PUSTAKA

- [ADA-12] Adam, Eriq Muhammad dan Widhy Hayuhardika. 2012. Pemrograman Perangkat Mobile – BAB 13 Google Maps API. Teknik Informatika-PTIHK.
- [AEL - 12] Aelani, Khoirida dan Falahah. 2012. Pengukuran *Usability* Sistem Menggunakan *USE Questionnaire* (Studi Kasus Aplikasi Perwalian *Oline* STMIK “AMIKBANDUNG”). Yogyakarta: SNATI 2012. Diakses dari <http://journal.uui.ac.id/index.php/Snati/article/viewFile/2913/2676>. Tanggal akses 10 Desember 2015.
- [ANG – 15] AngularJS. “*Introduction*”. Diakses pada tanggal 7 September 2015. <https://docs.angularjs.org/guide/introduction>
- [ANW-14] Anwar, Badrul, Hendra Jaya dan Putra Indra Kusuma. 2014. Implementasi *Location Based Service* berbasis Android untuk Mengetahui Posisi User. Medan : Jurnal Ilmiah Saintikom (ISSN: 1978-6603)
- [BOC – 98] Booch, Rumbaugh dan Jacobson. 1998. *The Unified Modelling Language User Guide*. Canada : Addison Wesley Longman, Inc.
- [BPS – 02] BPS. 2002. Leaflet Nilai Tukar Petani. Akses dari http://www.bps.go.id/brs_file/leaflet-ntp-23okt02.pdf . Tanggal Akses 16 Oktober 2014
- [BPS – 04] BPS. 2004. Data dan Informasi Kemiskinan Tahun 2004. Buku 2: Kabupaten. Badan Pusat Statistik, Jakarta Indonesia.
- [COR-15] Apache Cordova, “*About Apache Cordova*”. Diakses pada tanggal 10 Februari 2015. <http://cordova.apache.org/>
- [COU-10] McTavish, Courtney dan Suresh Sankaranarayanan. 2010. *Intelligent Agent based Hotel Search & Booking System*. Kingston: IEEE.
- [DHA – 03] Dharwiyanti, Sri. 2003. Pengantar *Unified Modelling Language (UML)*. Akses dari <http://mirror.unej.ac.id/iso/dokumen/ikc/yanti-uml.pdf>. Tanggal Akses 15 November 2014
- [ELI – 11] Fischer, Elisabeth dan Martin Qaim. 2011. *Linking Smallholders to Markets: Determinants and Impacts of Farmers Collective Action in Kenya*. Jerman: Elsevier. doi: 10.1016/j.worlddev.2011.11.018
- [GOD – 13] Godfrey Nolan, David Truxall, Onur Cinar. 2013. *Android Best Practices*. Addison-Wesley. 167-168.
- [GOK-13] Gok, Nizamettin dan Nitin Khanna. 2013. *Building Hybrid Android Apps with Java and Javascript*. Amerika. O’Reilly Media Inc.

- [ION - 15] Ionic Framework. "Ionic Framework". Diakses pada tanggal 7 September 2015. <http://ionicframework.com/docs/guide/preface.html>
- [JSN-14] JSON. "Introducing JSON". Diakses pada tanggal 22 April 2014. www.json.org.
- [KAD - 03] Kadir, Abdul. 2003. Pengenalan Sistem Informasi. Yogyakarta: Andi.
- [KPT - 11] Kementrian Pertanian. 2011 Rencana Strategis 2010 - 2014 Direktorat Jenderal Pengolahan dan Pemasaran Hasil Pertanian Revisi 1. Jakarta.
- [KSI-08] Kogent Solutions Inc. 2008. "Ajax Black Book, New Edition". Penerbit Dreamtech Press. New Delhi.
- [MUC - 13] Rachmat, Muchjidin. 2013. Nilai Tukar Petani : Konsep, Pengukuran dan Relevansinya sebagai Indikator Kesejahteraan Petani. Bogor. Pusat Sosial Ekonomi dan Kebijakan Pertanian.
http://pse.litbang.pertanian.go.id/ind/pdf/files/FAE31-2_b.pdf
diakses pada tanggal 8 September 2015
- [NGC - 15] NgCordova. "NgCordova". Diakses pada tanggal 7 September 2015. <http://ngcordova.com>
- [NIE - 12] Nielsen, Jakob. 2012. "Usability 101 : Introduction". Akses dari: <http://www.useit.com/alertbox/20030825.html>. Tanggal akses 10 Desember 2015.
- [PRE - 01] Pressman, R.S. 2001. *Software Engineering : A Practitioner's Approach (Fifth Edition)*. New York : McGraw-Hill Book Co.
- [PRE - 10] Pressman, Roger S. 2010. *"Software Engineering: a practitioner's approach, 7th edition"*. Mc Graw I lill.
- [REN - 14] Pranadipa, Rendra. 2014. Rancang Bangun Sistem Informasi Jual Beli Bahan Pangan Dan Pertanian Berbasis Web. Malang: PTIIK.
- [RIS - 12] Risnita. 2012. Pengembangan Skala Model Likert. Edu-Bio Vol 3, p. 88. Diakses dari <http://download.portalgaruda.org/article.php?article=252693&val=6813&title=Pengembangan%20Skala%20Model%20Likert>. Tanggal akses 10 Desember 2015.
- [RUB - 99] Rumbaugh, Jacobson dan Booch. 1999. *The Unified Modelling Language Reference Manual*. Canada : Addison Wesley Longman, Inc.
- [SAF - 12] Safaat, Nazruddin. 2012. Pemrograman Aplikasi *Mobile Smartphone* dan *Tablet PC* berbasis Android. Informatika, Bandung.
- [SIM - 10] Simarmata, Janner. 2010. "Rekayasa Perangkat Lunak". Yogyakarta: Andi

- [SUR – 06] Suryatmo, F. dan Dedy Rusmadi. 2006. Pengetahuan Dasar Komputer. Semarang : Rineka Cipta.
- [WAM – 12] Komputer, Wahana. 2012. Membuat Aplikasi Android untuk *Tablet* dan *Handphone*. Jakarta : PT Elex Media Komputindo.
- [WMS – 10] Komputer, Wahana. 2010. Panduan Belajar MySQL Database Server. Jakarta : Media Kita.

