

**RANCANG BANGUN APLIKASI LAYANAN INFORMASI TANAH
UNTUK PETUGAS LAPANGAN KANWIL BPN JAWA TIMUR
MENGUNAKAN PLATFORM ANDROID**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Bhawa Jatmiko
NIM: 115060807111079



**INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2015**

PENGESAHAN

RANCANG BANGUN APLIKASI LAYANAN INFORMASI TANAH UNTUK PETUGAS
LAPANGAN KANWIL BPN JAWA TIMUR MENGGUNAKAN PLATFORM ANDROID

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Bhawa Jatmiko

NIM: 115060807111079

Skripsi ini telah diuji dan dinyatakan lulus pada
24 Maret 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Agi Putra Kharisma, S.T., M.T.

NIK: 201304 860430 1 001

Herman Tolle, Dr. Eng., S.T., M.T.

NIK: 19740823 200012 1 001

Mengetahui

Ketua Program Studi NamaProgramStudi

Drs. Marji, M.T

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,

Bhawa Jatmiko

NIM: 115060807111079



KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena hanya dengan rahmat dan karuniaNya, penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun Aplikasi Perangkat Bergerak Informasi Tanah”. Melalui kesempatan ini, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan batuan dan dukungan selama penulisan skripsi ini, diantaranya:

1. Bapak Agi Putra Kharisma, ST., MT. selaku dosen pembimbing I yang telah memberikan masukan dan ilmu serta saran dalam menyelesaikan skripsi ini.
2. Bapak Dr.Eng. Herman Tolle, ST., MT. selaku dosen pembimbing II yang juga memberikan masukan dan ilmu serta saran dalam menyelesaikan skripsi ini.
3. Ayah Bambang Sancoyo dan Ibu Ratna Kartika dan Kaka penulis serta seluruh keluarga besar atas segala doa, nasehat dan dukungan yang diberikan hingga terselesainya skripsi ini.
4. Bapak Ir. Sutрино, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc dan Bapak Edy Santoso, S.Si, M.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2, dan Wakil Ketua 3 Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Informatika/Illmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya.
6. Bapak Denny Sagita Rusdianto, S.Kom, M.Kom selaku dosen pembimbing akademik yang telah memberikan pengarahan selama penulis menempuh pendidikan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
7. Seluruh Civitas Akademik Program Studi Informatika/Illmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi dan selama penyelesaian skripsi ini.
8. Teman-teman unit aktivitas bulu tangkis (UABT) yang telah memberi motivasi, dukungan, semangat, dan doa selama penulis menempuh studi di Fakultas Ilmu Komputer Universitas Brawijaya.
9. Teman-teman *Shine Models Management* yang selalu memberikan dukungan serta semangat dan selalu menghibur penulis dalam menyelesaikan skripsi ini.

10. Teman-teman kuliah diantaranya kelas TIF-H, serta semua teman-teman angkatan 2011, terima kasih atas segala bantuannya selama menjadi mahasiswa serta dukungannya dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan untuk menyempurnakan skripsi ini. Semoga skripsi ini membawa manfaat bagi pihak lain yang menggunakannya.

Malang, 24 Maret 2016

Penulis

bosseywashburn@gmail.com



ABSTRAK

Bhawa Jatmiko.2016. : Rancang Bangun Aplikasi Layanan Informasi Tanah Untuk Petugas Lapangan Kanwil BPN Jawa Timur Menggunakan Platform Android. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Univeristas Brawijaya.

Dosen Pembimbing : Agi Putra Kharisma, ST., MT. dan Dr. Eng Herman Tolle, ST., MT.

BPN (Badan Pertanahan Nasional) merupakan lembaga pemerintahan yang bertugas dalam pembinaan dan pelayanan administrasi umum dibidang pertanahan. Tugas utama dari lembaga BPN adalah mendata setiap bidang tanah dengan akurat dengan menggunakan *GPS Handheld* untuk menentukan lokasi yang tepat, serta dokumen-dokumen pendukung berupa sertifikat tanah, peta tanah, dan gambar ukur. Banyaknya peralatan tersebut berdampak pada lamanya proses pendataan dan survei tanah, serta dapat mempersulit pendataan dan kurang efisien.

Untuk mengatasi permasalahan tersebut, dibutuhkan suatu aplikasi yang dapat menampung data-data yang cukup banyak dan mampu menunjukan lokasi tanah maupun pengguna. Aplikasi informasi tanah ini merupakan aplikasi perangkat bergerak dengan menggunakan kerangka kerja *phonegap*. Langkah-langkah utama pada penelitian ini terdiri dari analisis kebutuhan, perancangan, implementasi, dan pengujian, serta dilakukan pengujian fungsional dan non-fungsional.

Hasil pengujian validasi yang dilakukan menunjukan semua fitur pada sistem telah terpenuhi dan sesuai kebutuhan. Selain itu, Untuk pengujian *non-fungsional* menggunakan metode kuantitatif berupa *quisioner* yang disebarakan ke beberapa *user*, dari hasil penelitian didapat rata-rata 84.132% dengan *range* 80%-100% berdasarkan *learnability*, *efficiency*, *memoriability*, *errors*, dan *satification* sehingga aplikasi ini sangat memuaskan dan dapat berjalan dengan baik.

Kata kunci: GPS, mobile, phonegap, Android.

ABSTRACT

Bhawa Jatmiko.2016. : *Design of Land Information Service application for Field Officers BPN Regional Office of East Java Platform Using Android. Thesis Informatics Engineering Program, Program of Information Technology and Computer Science, the University of Brawijaya.*

Advisor: Agi Putra Kharisma, ST., MT. dan Dr. Eng Herman Tolle, ST., MT.

BPN (National Land Agency) is a government agency in charge of development and public administration services in the field of land. The main task of the agency BPN is the list each plot accurately using GPS Handheld to determine the exact location, as well as supporting documents such as certificates of land, soil maps, and image measurement. The number of such equipment have an impact on the length of the data processing and ground surveys, and can complicate data collection and less efficient.

To overcome these problems, it takes an application that can collect data that is quite a lot and was able to show the location of the land and users. Application information is the land of mobile device applications using the framework phonegap. The major steps in this study consists of requirements analysis, design, implementation, and testing, as well as testing the functional and non-functional.

The results of validation testing performed shows all features of the system have been fulfilled and as needed. In addition, for non-functional testing using quantitative methods such as questionnaires distributed to multiple users, from the result is an average of 84 132% with a range of 80% -100% based learnability, efficiency, memoriability, errors, and so the application is very satisfaction satisfactory and can run well.

Keywords: GPS, mobile, phonegap, Android.

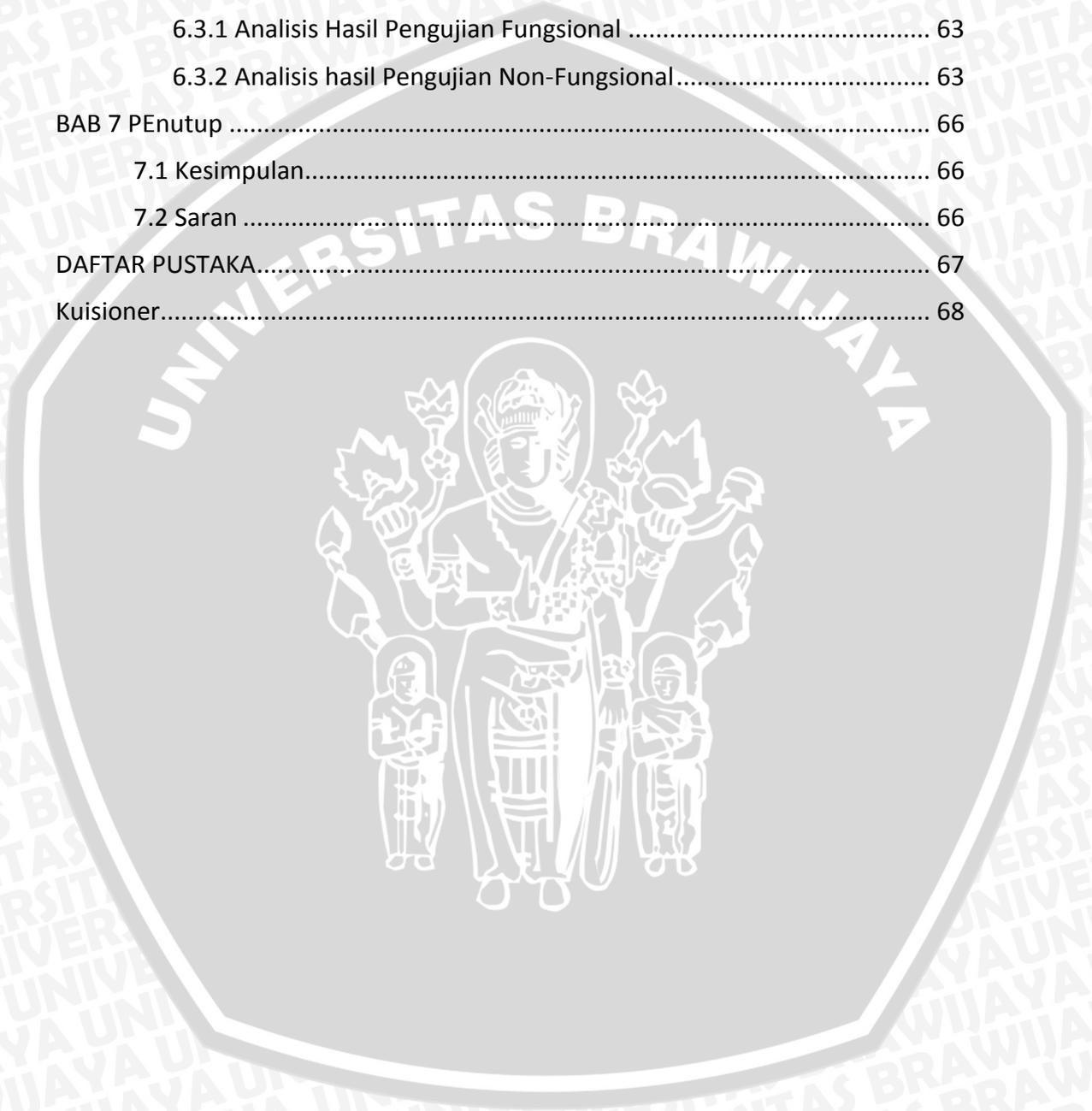
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Petugas Lapangan BPN Kanwil Jawa Timur	5
2.3 GPS Pada <i>Smartphone</i>	5
2.4 <i>Google earth</i>	5
2.5 Android	6
2.6 Phonegap	6
2.7 HTML.....	8
2.8 CSS.....	9
2.9 JavaScript	11
2.10 JQuery UI.....	12
2.11 <i>Web API</i>	12
2.12 JSON	12
2.13 Kohana	15

2.14 KML/KMZ	16
BAB 3 METODOLOGI	17
3.1 Studi Literatur	17
3.2 Analisis Kebutuhan	18
3.3 Perancangan	18
3.4 Implementasi	18
3.5 Pegujian	19
3.6 Pengambilan Kesimpulan dan Saran	19
BAB 4 Rekayasa kebutuhan.....	20
4.1 Analisis Kebutuhan	21
4.1.1 Gambaran Umum Aplikasi	21
4.2 Identifikasi Aktor	22
4.3 Kebutuhan Fungsional	22
4.3.1 Diagram Usecase	23
4.3.2 Skenario <i>Use case</i>	24
4.4 Kebutuhan Non-Fungsional	29
BAB 5 PERANCANGAN DAN IMPLEMENTASI	30
5.1 Perancangan Aplikasi Perangkat Bergerak	31
5.1.1 <i>Activity Diagram</i>	31
5.1.2 Arsitektur Aplikasi	35
5.1.3 Interaksi Antar Objek	35
5.1.4 <i>Class Diagram</i>	39
5.1.5 Basis Data	40
5.1.6 Navigasi dan Antarmuka	41
5.2 Implementasi Aplikasi Perangkat Bergerak	45
5.2.1 Spesifikasi Sistem	45
5.2.2 Implementasi Basis Data	46
5.2.3 Implementasi Kode Aplikasi	47
5.2.4 Implementasi Antarmuka	52
BAB 6 Pengujian dan analisis	57
6.1 Pengujian Fungsional	57
6.1.1 Kasus Uji Fungsional	57



6.1.2 Hasil Pengujian Fungsional.....	60
6.2 Pengujian Non-Fungsional	60
6.2.1 Pengujian <i>Usability</i>	60
6.3 Analisis	63
6.3.1 Analisis Hasil Pengujian Fungsional	63
6.3.2 Analisis hasil Pengujian Non-Fungsional.....	63
BAB 7 PEnutup	66
7.1 Kesimpulan.....	66
7.2 Saran	66
DAFTAR PUSTAKA.....	67
Kuisisioner.....	68



DAFTAR TABEL

Tabel 4.1 Identifikasi Aktor	22
Tabel 4.2 Daftar Kebutuhan Fungsional.....	23
Tabel 4.3 Skenario melihat lokasi <i>user</i>	24
Tabel 4.4 Skenario melihat lokasi <i>user</i>	25
Tabel 4.5 Skenario melihat informasi lokasi	25
Tabel 4.6 Skenario mengolah data.....	26
Tabel 4.7 Skenario menambah KML/KMZ	27
Tabel 4.8 Skenario menghapus KML/KMZ.....	28
Tabel 4.9 Skenario validasi	28
Tabel 4.10 Analisis kebutuhan non- <i>functional</i>	29
Tabel 5.1 Spesifikasi Perangkat Keras Komputer.....	45
Tabel 5.2 Spesifikasi Perangkat Keras Android	45
Tabel 5.3 Spesifikasi Perangkat Lunak pada Komputer	45
Tabel 5.4 Spesifikasi Perangkat Lunak pada Perangkat Bergarak.....	46
Tabel 5.5 Struktur Tabel <i>Devices</i>	46
Tabel 5.6 Struktur Tabel <i>Kml_kmzs</i>	46
Tabel 5.7 Struktur Tabel <i>Users</i>	47
Tabel 6.1 Kasus Uji Fungsional Melihat Lokasi <i>User</i>	58
Tabel 6.2 Kasus Uji Fungsional Melihat Lokasi Denah	58
Tabel 6.3 Kasus Uji Fungsional Lihat Informasi Denah	59
Tabel 6.4 Kasus Uji Fungsional Validasi.....	59
Tabel 6.5 Hasil Pengujian Fungsional.....	60
Tabel 6.6 Isi Kuesioner USE	60
Tabel 6.7 Hasil Pengujian <i>Usability</i>	62
Tabel 6.8 Interpretasi Skor Likert.....	63
Tabel 6.9 Indeks Presentasi Hasil Kuesioner.....	64
Tabel 6.10 Status Pengujian <i>Usability</i> Aplikasi Informasi Tanah	65
Tabel 6.11 Rumus Perhitungan	65

DAFTAR GAMBAR

Gambar 2.1 PhoneGap <i>Application Architecture</i>	7
Gambar 2.2 PhoneGap <i>Application Device Interaction</i>	7
Gambar 2.3 Aplikasi PhoneGap di Berbagai Platform	7
Gambar 2.4 Fitur API PhoneGap di Berbagai Platform	8
Gambar 2.5 <i>Tag HTML Untuk Format Text</i>	9
Gambar 2.6 Objek pada JSON	13
Gambar 2.7 <i>Array</i> pada JSON	14
Gambar 2.8 <i>Value</i> pada JSON	14
Gambar 2.9 <i>String</i> pada JSON	14
Gambar 2.10 Objek pada JSON	15
Gambar 3.1 Diagram Alir Tahapan Penelitian.....	17
Gambar 4.1 Diagram alur kondisi saat ini.....	20
Gambar 4.2 Diagram Pohon Rekayasa Kebutuhan	21
Gambar 4.3 Diagram alur sistem yang ditawarkan.....	22
Gambar 4.4 Diagram <i>Use case</i>	23
Gambar 5.1 Pohon Diagram Perancangan Dan Implementasi	30
Gambar 5.2 <i>Activity</i> diagram melihat lokasi <i>user</i>	31
Gambar 5.3 <i>Activity</i> diagram melihat lokasi denah.....	32
Gambar 5.4 <i>Activity</i> diagram lihat informasi denah	32
Gambar 5.5 <i>Activity</i> diagram mengelola data, menambah, dan Menghapus KML/KMZ.....	33
Gambar 5.6 <i>Activity</i> diagram valiasi.....	34
Gambar 5.7 Arsitektur sistem	35
Gambar 5.8 <i>Sequence</i> diagram melihat lokasi <i>user</i>	36
Gambar 5.9 <i>Sequence</i> diagram melihat lokasi denah.....	36
Gambar 5.10 <i>Sequence</i> diagram lihat informasi denah.....	37
Gambar 5.11 <i>Sequence</i> diagram mengelola data, menambah KML/KMZ, dan Menghapus KML/KMZ.....	38
Gambar 5.12 <i>Sequence</i> diagram validasi	39
Gambar 5.13 <i>Class</i> diagram aplikasi	39

Gambar 5.14 Model ER-Diagram	40
Gambar 5.15 Navigation Map Aplikasi.....	41
Gambar 5.16 <i>Flow Screen</i> Aplikasi	42
Gambar 5.17 Perancangan antarmuka halaman login	42
Gambar 5.18 Perancangan antarmuka halaman dashboard	43
Gambar 5.19 Perancangan antarmuka halaman <i>left side</i> menu	43
Gambar 5.20 Perancangan antarmuka halaman <i>right side</i> menu.....	44
Gambar 5.21 Perancangan antarmuka halaman validasi	44
Gambar 5.22 Implementasi Proses Penampilan Denah pada <i>Smartphone</i>	48
Gambar 5.23 Implementasi Proses Penampilan Denah pada <i>Web Manager</i>	50
Gambar 5.24 Implementasi Proses Tampilan Lokasi <i>User</i>	50
Gambar 5.25 Implementasi Proses Digitasi	51
Gambar 5.26 Implementasi Proses Hitung Luas	51
Gambar 5.27 Halaman <i>login</i>	52
Gambar 5.28 Halaman Informasi Tanah	52
Gambar 5.29 <i>Left side Menu</i>	53
Gambar 5.30 <i>Right side Menu</i>	53
Gambar 5.31 Tampilan Posisi <i>User</i>	54
Gambar 5.32 Tampilan denah.....	54
Gambar 5.33 Informasi data tanah.....	55
Gambar 5.34 Halaman informasi <i>validasi</i>	55
Gambar 5.35 Halaman informasi <i>validasi</i>	56
Gambar 5.36 Halaman informasi <i>validasi</i>	56
Gambar 6.1 Pohon Diagram Pengujian dan Analisis.....	57

DAFTAR LAMPIRAN

Kuisisioner..... 68



BAB 1 PENDAHULUAN

1.1 Latar belakang

BPN (Badan Pertanahan Nasional) adalah lembaga pemerintah non kementerian yang berada dibawah dan bertanggung jawab kepada presiden. Lembaga ini mempunyai visi untuk mewujudkan tanah dan pertanahan untuk sebesar-besar kemakmuran rakyat, serta keadilan dan keberlanjutan sistem kemasyarakatan, kebangsaan dan kenegaraan Republik Indonesia. Selain itu BPN mempunyai tugas yang diantaranya pembinaan dan pelayanan administrasi umum di bidang pertanahan, penyelenggara dan pelaksanaan survei pengukuran dan pemetaan dibidang pertanahan, pelaksanaan pendaftaran tanah dalam rangka menjamin kepastian hukum, pengaturan dan penetapan hak-hak atas tanah, pengelola data informasi di bidang pertanahan. Tugas utama dari lembaga BPN adalah mendata setiap bidang tanah dengan akurat dengan menggunakan GPS *Handheld* untuk menentukan lokasi yang tepat, serta dokumen-dokumen pendukung berupa sertifikat tanah, peta tanah, gambar ukur.

Dalam melakukan tugasnya petugas BPN diharuskan menggunakan peralatan yang banyak. Dengan banyaknya peralatan tersebut dapat menghambat pekerjaan dari lamanya waktu pengukuran dan tidak efisiennya peralatan yang digunakan.

Untuk mendukung kinerja tugas-tugas BPN diperlukan teknologi modern yang dapat digunakan dengan mudah seperti *handphone*, salah satu perangkat teknologi yang telah mengalami evolusi saat ini dan sering digunakan adalah *smartphone*, dimana teknologi *smartphone* dapat mendukung sebuah sistem navigasi atau penentu posisi berbasis satelit yang dikenal dengan GPS (*Global Positioning System*). Seperti yang diketahui GPS berfungsi memberikan posisi dan informasi mengenai waktu secara kontinyu di seluruh dunia tanpa tergantung cuaca. Dengan menggunakan GPS yang ada pada *smartphone* akan memudahkan dan membantu pekerjaan BPN dalam melaksanakan tugas-tugasnya.

Berdasarkan kebutuhan tugas-tugas yang ada di kantor BPN dibutuhkan aplikasi layanan informasi yang efektif dan efisien, serta dapat mempermudah pekerjaan petugas lapangan BPN tersebut dalam mendata dan menentukan lokasi tanah. Aplikasi tersebut harus didukung dengan perangkat GPS yang terdapat pada *smartphone*, untuk memudahkan petugas dalam melakukan pendataan atau pengecekan. Dengan menggunakan aplikasi layanan informasi secara *mobile* diperlukan data-data yang akurat dan teknologi pemrograman terkini seperti JAVA yang dapat diimplementasikan pada perangkat android.

Untuk memenuhi kebutuhan yang telah dipaparkan diatas, diperlukan sebuah *product* berupa aplikasi informasi tanah yang dapat diakses oleh seluruh petugas lapangan BPN dimana saja, dimana aplikasi tersebut dapat menginput data-data sertifikat tanah, peta tanah, dan gambar ukur tanah yang masih berbentuk dokumen. Data tersebut kemudian dikelola secara digital menggunakan teknologi *google earth* yang diupload ke dalam *web manager* sehingga menghasilkan

informasi yang dibutuhkan oleh petugas BPN baik di bagian administrasi ataupun dilapangan dalam bentuk *digital*.

1.2 Rumusan masalah

Dalam pembangunan sebuah aplikasi rumusan masalah mutlak diperlukan sebagai acuan untuk menentukan permasalahan atau langkah-langkah yang akan dilakukan. Adapun rumusan permasalahannya antara lain:

1. Bagaimana rancangan dan implementasi aplikasi layanan informasi tanah pada *smartphone* android yang berguna bagi petugas lapangan sebagai alat dalam pengecekan dan pendataan tanah dan validasi tanah?
2. Bagaimana masukan data sertifikat tanah, peta tanah, dan gambar ukur menggunakan teknologi *google earth* berupa *file* KML/KMZ yang dapat di *upload* menggunakan *web manager* pada aplikasi informasi tanah?
3. Bagaimana *validasi* data yang terdapat pada aplikasi berdasarkan perbandingan data pengukuran di lapangan dan data digital berupa penyajian informasi *validasi*?

1.3 Tujuan

Berdasarkan latar belakang dan permasalahan diatas tujuan dari tugas akhir ini adalah merancang dan membangun suatu aplikasi yang dapat berjalan pada *smartphone*, dengan menggunakan teknologi aplikasi *mobile* dan digunakan oleh petugas lapangan BPN kanwil Jawa Timur dalam pendataan tanah.

1.4 Manfaat

Adapun manfaat dari aplikasi layanan informasi tanah ini antara lain:

1. Membantu petugas lapangan dalam mencatat dan memvalidasi informasi tanah.
2. Mengefisienkan peralatan petugas lapang dalam melakukan identifikasi informasi yang ada dilapangan.
3. Memudahkan pencarian informasi tanah oleh petugas BPN dalam bentuk digital.

1.5 Batasan masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka pada penelitian ini dibatasi beberapa permasalahan antara lain:

1. Data yang digunakan sebagai masukan dalam pemrosesan aplikasi ini berupa data informasi berbentuk *text*.
2. Media yang digunakan untuk mencari informasi lokasi tanah dan koordinat menggunakan *google earth* berbentuk *file* KML/KMZ.
3. Data dari hasil analisis pencarian data di *upload* menggunakan aplikasi *web manager*.
4. Data yang didapat dari hasil analisis di representasikan dalam bentuk penyajian informasi.

5. Teknik pengujian perangkat lunak yang digunakan dalam penelitian ini menggunakan teknik *black box testing* yang dilakukan pada tiap fungsional.
6. Data informasi yang telah dihasilkan sebelumnya dilakukan pengecekan dan sinkronisasi pada aplikasi berupa data informasi *validasi*.
7. Aplikasi yang dikembangkan dapat berjalan dengan baik minimal menggunakan android versi 4.0 (*ice cream sandwich*).

1.6 Sistematika pembahasan

Untuk lebih jelas memahami gambaran dan uraian dari penyusunan skripsi ini maka akan dilakukan pengelompokan materi menjadi beberapa bab dengan sistematika penulisan sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan laporan tugas akhir.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi tentang kajian pustaka yang berasal dari literatur, makalah maupun buku-buku yang memperkuat penulisan tugas akhir ini dan dasar teori yang membahas secara umum mengenai hal-hal teknis yang nantinya diperlukan dalam pengembangan aplikasi mobile penampil komik strip.

BAB 3 METODOLOGI

Bab ini menjelaskan mengenai metode yang digunakan dalam penelitian yang terdiri dari studi literatur, rekayasa kebutuhan, perancangan dan implementasi perangkat lunak, serta pengujian dan analisis.

BAB 4 REKAYASA KEBUTUHAN

Bab ini berisi tentang analisa kebutuhan pada aplikasi mobile penampil komik strip meliputi *use case diagram* dan *use case* skenario.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi tentang perancangan aplikasi yang akan dibuat yang kemudian menjadi landasan implementasi, meliputi penjelasan dan teknik mengenai pembuatan aplikasi perangkat bergerak penampil komik strip.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini berisi tentang teknik atau metode pengujian yang dilakukan pada aplikasi perangkat bergerak penampil komik strip, serta melakukan analisa terhadap hasil pengujian.

BAB 7 PENUTUP

Bab ini berisi kesimpulan dan saran. Kesimpulan yang didapatkan akan disesuaikan dengan permasalahan yang nantinya ditemukan dalam implementasi

dan pengujian, sedangkan saran digunakan sebagai unsur usulan untuk melengkapi aplikasi yang telah dikembangkan.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka penelitian ini membahas penelitian sebelumnya yang meneliti tentang penggunaan metode LBS (*Location Based Services*). Penelitian yang berkaitan dengan metode tersebut dilakukan oleh Juwita Imaniar, Arifin ST. MT., dan Ahmad Subhan Khalilullah (Juwita & Arifin & Khalilullah, 2011).

Penelitian yang dilakukan oleh Juwita Imaniar, Arifin ST. MT., dan Ahmad Subhan Khalilullah pada jurnal yang berjudul “Aplikasi *Location Based Service* untuk Sistem Informasi Publikasi Acara pada *Platform Android*” menjelaskan tentang metode yang efisien untuk membuat sistem informasi pada publikasian acara di kota Surabaya. Secara garis besar jenis Layanan Berbasis Lokasi juga dapat dibagi menjadi dua, yaitu *Pull Service*, dan *Push Service*. *Pull Service* merupakan layanan diberikan berdasarkan permintaan dari pelanggan akan kebutuhan suatu informasi. Jenis layanan ini dapat dianalogikan seperti mengakses suatu *web* pada jaringan internet. *Push Service* merupakan layanan ini diberikan langsung oleh *service provider* tanpa menunggu permintaan dari pelanggan, tentu saja informasi yang diberikan tetap berkaitan dengan kebutuhan pelanggan (Juwita & Arifin & Khalilullah, 2011).

2.2 Petugas Lapangan BPN Kanwil Jawa Timur

BPN (Badan Pertanahan Negara) adalah lembaga non kementerian yang bertugas di bidang pertanahan secara nasional, regional, dan sektoral (BPN, 2015). Dalam menjalankan tugasnya ini BPN melakukan pendataan tanah, pengukuran tanah serta mengeluarkan sertifikat tanah.

Dalam menjalankan fungsinya tersebut petugas lapangan menggunakan GPS *Handheld* dalam menemukan lokasi yang tepat, dan membawa dokumen-dokumen berupa sertifikat tanah, peta tanah dan gambar ukur.

2.3 GPS Pada *Smartphone*

Berkembangnya teknologi juga berpengaruh terhadap dunia komunikasi. *Handphone* yang dulunya hanya digunakan untuk menelpon kini terdapat perangkat-perangkat yang lebih modern, salah satunya GPS. GPS sekarang terdapat pada *handphone* atau yang sekarang disebut *smartphone* digunakan untuk menentukan lokasi dari perangkat tersebut. Seperti tugasnya GPS memberikan informasi lokasi dengan menentukan koordinat X dan Y atau garis bujur dan garis lintang (*longitude / latitude*).

2.4 *Google earth*

Google earth merupakan sebuah program globe virtual yang dibuat oleh *Keyhole, Inc.* Program ini memetakan bumi dari superimposisi gambar yang dikumpulkan dari pemetaan satelit, fotografi udara dan globe GIS 3D. Awalnya

dikenal sebagai *Earth Viewer*, *Google earth* dikembangkan oleh Keyhole, Inc., sebuah perusahaan yang diambil alih oleh Google pada tahun 2004. Produk ini, kemudian diganti namanya menjadi *Google earth* tahun 2005, dan sekarang tersedia untuk komputer pribadi yang menjalankan Microsoft Windows 2000, XP, atau Vista, Mac OS X 10.3.9 dan ke atas, Linux (diluncurkan tanggal 12 Juni 2006) dan FreeBSD. Dengan tambahan untuk peluncuran sebuah klien berbasis *update Keyhole*, Google juga menambah pemetaan dari basis datanya ke perangkat lunak pemetaan berbasis *web*. Peluncuran *Google earth* menyebabkan sebuah peningkatan lebih pada cakupan media mengenai *globe* virtual antara tahun 2005 dan 2006, menarik perhatian publik mengenai teknologi dan aplikasi *geospatial*.

KML (*keyhole markup language*) merupakan sintak kode berbasis XML yang digunakan untuk *generate image* pada *google earth*. Hasil dari file KML akan di *compress* menggunakan file KMZ, sehingga data yang dihasilkan KML yang telah di *compress* ke dalam KMZ akan lebih kecil ukurannya dan memudahkan dalam *upload* data ke *web manager*.

2.5 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux, android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak (Wicaksono, 2013).

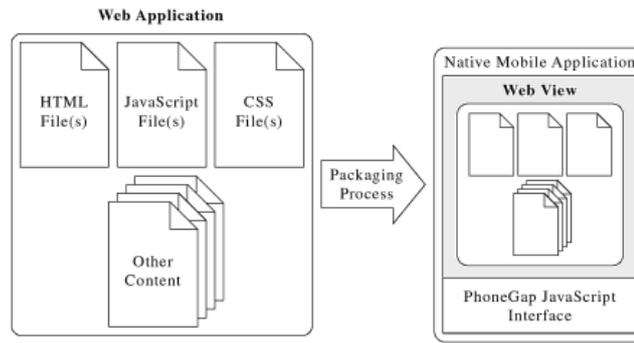
Secara garis besar sistem operasi Android memiliki arsitektur :

- a. *Applications dan widget*
- b. *Applications Frameworks*
- c. *Libraries*
- d. *Android Run Time*
- e. *Linux Kernel*

2.6 Phonegap

PhoneGap adalah *open source framework* untuk membuat *cross-platform native applications* menggunakan teknologi *web* mulai dari HTML, CSS, dan JavaScript. Tipe dari aplikasi ini disebut sebagai *hybrid application*. PhoneGap diciptakan untuk mempermudah *mobile development* (Wargo, 2012).

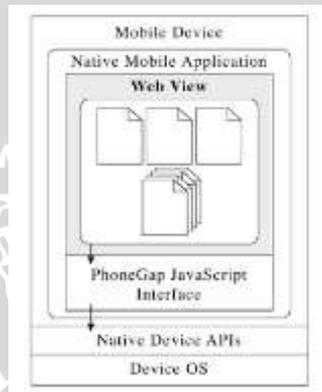
PhoneGap bekerja dengan cara merubah *web application package* menjadi *native application*. Aplikasi yang telah dibuat akan ditampilkan dalam bentuk *web View* yang memungkinkan pengguna untuk melakukan interaksi dengan aplikasi tersebut (Wargo, 2012).



Gambar 2.1 PhoneGap Application Architecture

Sumber: (Wargo, 2012)

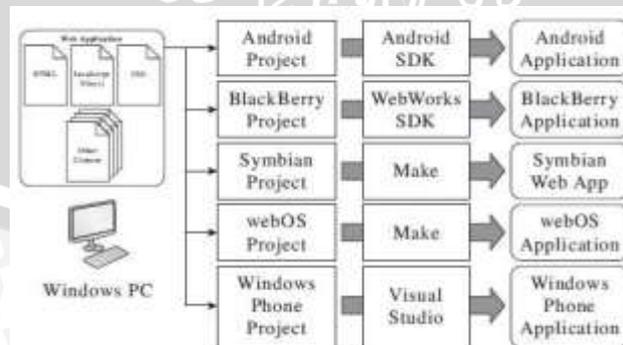
Gambar 2.1 menunjukkan tampilan dari aplikasi yang dibuat oleh PhoneGap dibentuk oleh CSS yang di implementasikan kedalam HTML dan untuk komunikasi data bisa menggunakan XML atau JSON.



Gambar 2.2 PhoneGap Application Device Interaction

Sumber: (Wargo, 2012)

Gambar 2.2 menunjukkan aplikasi yang telah selesai dibuat dengan PhoneGap akan dirubah dengan *Software Development Kit (SDK)* platform yang didukung oleh PhoneGap menjadi *native application*.



Gambar 2.3 Aplikasi PhoneGap di Berbagai Platform

Sumber: (Wargo, 2012)

Gambar 2.3 menunjukkan PhoneGap mendukung platform terkenal antara lain Apple iOS, Google Android, HP/Palm webOS, Microsoft Windows Phone 7, Nokia Symbian, RIM BlackBerry, Samsung Bada (Wargo, 2012).

Dikarenakan fungsi setiap platform berbeda-beda, maka PhoneGap menyediakan *Application Programming Interface* (API) untuk mempermudah pengembang diantaranya adalah *Accelerometer, Camera, Capture, Compass, Connection, Contacts, Device, Events, File, Geolocation, Media, Notification, dan Storage*. API tersebut dibuat agar pengembang dapat mengakses fungsi *native application* melalui javascript dengan *syntax* yang sama di semua platform. Tidak semua platform dapat menggunakan API PhoneGap karena keterbatasan software maupun hardware tiap platform, berikut adalah perbandingan API di masing masing platform.

	Phone OS	Phone OS and more	Android	Symbian OS 2.0	Symbian OS 3.0	Android	Windows Phone 7	Symbian	Bada
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	X	✓	✓	✓	✓
Contacts	✓	✓	✓	✓	✓	✓	✓	✓	✓
File	✓	✓	✓	✓	✓	X	✓	✓	✓
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	X	✓	✓	✓	✓
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
LocalStorage (Web)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓	✓
LocalStorage (Web)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓	✓

Gambar 2.4 Fitur API PhoneGap di Berbagai Platform

Sumber: (Wargo, 2012)

Gambar 2.4 menunjukkan Dalam penyimpanan data PhoneGap menggunakan teknologi *local storage* yang terdapat pada JAVA selain itu juga PhoneGap dapat menggunakan API *storage* untuk mengakses SQLite pada *device* Android (Wargo, 2012).

2.7 HTML

HTML Merupakan suatu *script* dimana kita bisa menampilkan informasi dan daya kreasi kiata lewat internet. HTML memiliki perbedaan dengan dokumen teks biasa. misalnya: perbedaan yang paling mencolok adalah pada dokumen word, banyaknya karakter akan terbatas oleh besarnya kertas. Sedangkan HTML tak memiliki batasan teks (Wulandari, 2012).

HTML adalah Bahasa pemrograman yang fleksibel di mana kita bisa meletakkan skrip dari bahasa pemrograman lain seperti JAVA, Visual Basic, C dan lain-lain. Jika HTML tersebut tidak dapat mendukung suatu perintah pemrograman tertentu. *Browser* tidak akan menampilkan kotak *dialog*; *Syntax Error*; jika terdapat penulisan kode yang keliru pada skrip HTML

sepanjang kode-kode yang kita tuliskan merupakan kode-kode HTML tanpa penambahan kode-kode dari luar seperti java. Oleh karena itu, jika terjadi *syntax error* pada skrip HTML, efek yang paling jelas adalah HTML tersebut tak akan ditampilkan pada halaman jendela *Browser* (Wulandari, 2012).

HTML merupakan sebuah standar yang dimaintain oleh *World Wide Web Consorciium* (W3C). Sebagai standar maka *tag -tag* HTML tentunya akan diterjemahkan sama oleh setiap *browser* yang ada. Hal ini yang menyebabkan aplikasi-aplikasi *web* yang dikunjungi relatif sama tampilannya walaupun dibuka atau diakses menggunakan *browser* yang berbeda-beda seperti Internet Explorer, Firefox, Netscape, Opera dan lain-lain (Tempake, 2007).

Sebuah halaman HTML memiliki struktur seperti berikut ini:

```
<html>
<head>
<title>Title of page</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

HTML sebenarnya file teks biasa yang mengandung *tag -tag* khusus dan disimpan dengan ekstensi *.html* atau *.htm*. Untuk menjalankan *source* di atas dibutuhkan *browser* misalnya Firefox atau Internet Explorer (Tempake, 2007).

Tag HTML	Keterangan
... 	Huruf tebal
<i>... </i>	Huruf miring
<u>... </u>	Garis bawah
^{...}	Super script
_{...}	Sub script
<small>... </small>	Small text

Gambar 2.5 Tag HTML Untuk Format Text

Sumber: (Tempake, 2007)

Pada gambar 2.5 diatas terdapat bebera perintah yang sering digunakan dalam pembuatan program menggunakan bahasa HTML. Sehingga memudahkan *user* membuat *program* berbasis *web* (Tempake, 2007).

2.8 CSS

Cascading Style Sheets (CSS) adalah salah satu bahasa pemrograman desain *web* (*style sheet language*) yang mengontrol *format* tampilan sebuah halaman *web* yang ditulis dengan menggunakan bahasa penanda (*markup language*). Biasanya CSS digunakan untuk mendesain sebuah malam HTML dan XHTML, tetapi

sekarang bahasa pemrograman CSS bisa diaplikasikan untuk segala dokumen XML, termasuk SVG dan XUL (Ginting, 2013).

CSS dibuat untuk memisahkan konteks utama (biasanya dibuat dengan menggunakan bahasa HTML dan sejenisnya) dengan tampilan dokumen yang meliputi *layout*, warna dan *font*. Pemisahan ini dapat meningkatkan daya akses konten pada *web*, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi dari sebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi banyak halaman untuk sebuah *formatting* dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik tables pada *layout* desain *web* (*layout* tanpa tabel) (Ginting, 2013).

Style Sheet adalah sebuah *text* file yang sederhana (dimana berekstensi *.css*), ditulis menurut aturan bahasa yang dipaparkan pada rekomendasi CSS1 atau CSS2. Berikut contoh file CSS:

```
Body { Font-family: Helvetica, sans-Serif;
      Font-size: 12pt; Text-align: justify;
}
h2 {
  Font-family: Arial, sans-serif;
  Font-size: 14pt;
}
code{
  font-family: Courier, sans-serif;
  font-size: 12pt;
}
.note{
  Background-color: #cbcbcb;
}
```

Cara kerja CSS dengan menggunakan dua buah elemen penting untuk pemformatan tampilannya, diantaranya *selektor* dan *Deklarator*. Dua buah elemen ini berfungsi sebagai penentu *format* tampilan dan lainnya menempatkan *format* tampilan tersebut. *Deklarator* berisi beberapa perintah-perintah CSS untuk menentukan *format* dari sebuah elemen pada halaman *web*. Sedangkan *selektor* adalah sebuah perintah lanjut dari *Deklarator* dan berfungsi menempatkan *format* tampilan dari *Deklarator* (Ginting, 2013).

Dalam Cascading *Style Sheets* ada dua cara menghubungkan sebuah dokumen HTML dengan CSS

1. *Selector class*
2. *Selector ID*

3. *Selector Descendant* (turunan)
4. *Selector Link Pseudo Class*
5. *Selector Pseudo elemen*
6. *Selector dynamic pseudo class*
7. *Selector language*
8. *Selector child*
9. *Selector first-child*
10. *Selector adjacent* (berdekatan)

2.9 JavaScript

JavaScript adalah nama implementasi *Netscape Communications Corporation* untuk ECMA Script standar, suatu bahasa yang didasarkan pada konsep pemrograman berbasis prototipe. Bahasa ini terkenal karena penggunaannya di situs *web* (sebagai JavaScript sisi klien) dan juga digunakan untuk menyediakan akses skrip untuk objek yang dibenamkan (*embedded*) di aplikasi lain (Nugroho, 2008).

Walaupun memiliki nama serupa, JavaScript hanya sedikit hubungan dengan bahasa pemrograman Java, dengan kesamaan utamanya adalah penggunaan sintaks C. Secara semantic, JavaScript memiliki lebih banyak kesamaan dengan bahasa pemrograman *Self* [NUR-08]. Skrip JavaScript yang dimasukkan di dalam bahasa HTML ataupun XHTML harus dimasukkan di antara tag `<script>...</script>` (Tempake, 2007).

JavaScript juga dapat disisipkan pada tag `<head>` dari sebuah halaman HTML, sedangkan JavaScript yang disisipkan pada tag `<head>` akan dieksekusi (Tempake, 2007).

```
<html>
<head>
<script type="text/javascript">
function message()
{
alert("Hello Welcome to Javascript!")
}
</script>
</head>
<body >
<input type="button" value="Click Me"
onclick="message()">
```

```
</body>
```

```
</html>
```

2.10 JQuery UI

JQuery merupakan library dari JavaScript yang fokus dalam mengQuery objek JavaScript. *Selector* yang digunakan dalam jQuery biasanya mengakses kelas CSS untuk menangkap DOM dan mengolahnya dengan *method* tertentu.

Beberapa fitur yang menjadi inti dari jQuery adalah (Flanagan, 2011):

1. *Syntax* ekspresif (CSS *Selectors*) untuk menunjuk element pada dokumen.
2. *Query* yang efektif dan efisien dalam menemukan element pada dokumen.
3. Kumpulan *method* yang berguna untuk memanipulasi element yang terpilih.
4. *Succinct idiom* (*method* berantai) untuk membuat *sequence* dari operasi yang dibuat.

2.11 Web API

Web API adalah API (*Application Programing Interace*) yang bekerja diatas teknologi *Web* yang menerima request dari *client* dan memberikan respon yang sesuai melalui *protocol* / aturan tertentu. *Web API* lebih populer dengan nama "*web managers*", karena API tersebut memberikan layanan-layanan "*service*" yang diminta oleh *client*.

2.12 JSON

JSON (*JavaScript Object Notation*) adalah *format* pertukaran data ringan dan sangat mudah untuk dibaca dan ditulis manusia. JSON juga sangat mudah untuk parse ataupun *generate* oleh mesin. JSON berupa *format* teks yang sangat *independen*, tetapi menggunakan ketentuan yang familiar bagi programmer keluarga bahasa C, termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan banyak lainnya. Hal tersebut membuat JSON bahasa pertukaran data yang ideal (Json, 2015).

Pada bahasa pemrograman apapun, terdapat macam-macam tipe *number* untuk berbagai kapasitas dan komplemen, seperti *fixed* atau *float*, *biner* atau desimal. Keberagaman tersebut membuat proses pertukaran data antar bahasa pemrograman yang berbeda menjadi sulit. Pada JSON, *number* hanya direpresentasikan sesuai dengan yang digunakan manusia yakni *sequence of digits* (1, 2, 3, 4 ...). Semua bahasa pemrograman dapat menerima *format number sequence of digits* sehingga pertukaran dapat dilakukan (International, 2013).

Teks JSON merupakan rangkaian dari karakter yang dibentuk dari kode *Unicode* yang sesuai dengan aturan JSON *value*.

Enam karakter struktural:

[U+005B *left square bracket*

{ U+007B	left curly bracket
] U+005D	right square bracket
} U+007D	right curly bracket
: U+003A	colon
, U+002C	comma

Tiga karakter literal:

<i>true</i>	U+0074 U+0072 U+0075 U+0065
<i>false</i>	U+0066 U+0061 U+006c U+0073 U+0065
<i>null</i>	U+006E U+0075 U+006C U+006C

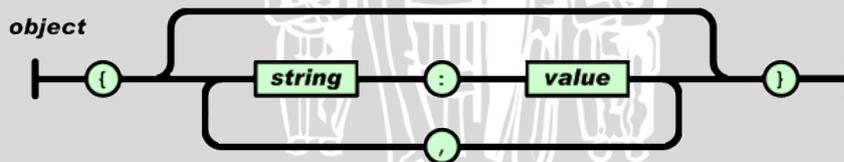
Whitespace yang diperbolehkan sebelum atau setelah karakter - karakter di atas

U+0020	Spasi
U+0009	character tabulation
U+000A	line feed
U+000D	Carriage return

Bentuk - bentuk yang terdapat pada JSON (Json, 2015):

1. Objek

Objek merupakan sekumpulan pasangan *name/value* yang tidak teratur, sebuah objek dimulai dengan tanda awalan kurung kurawal ({) dan diakhiri dengan akhiran kurung kurawal (}). Masing- masing *name* diikuti dengan tanda titik dua (:) dan pasangan *name/value* dipisahkan oleh tanda koma (,) seperti yang terlihat pada gambar 2.6 dibawah ini.



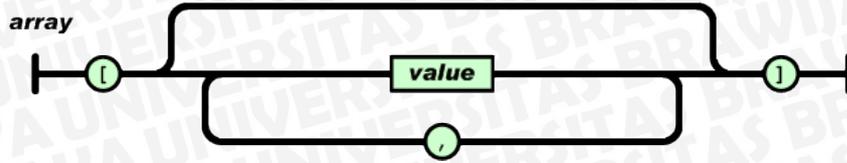
Gambar 2.6 Objek pada JSON

Sumber: (Json, 2015)

2. Array

Array merupakan kumpulan dari *value* yang teratur. *Array* dimulai dengan awalan kurung siku ([) dan diakhiri dengan akhiran kurung siku (]). *Value* dipisahkan oleh koma (,), seperti yang terlihat pada gambar 2.7 dibawah ini.



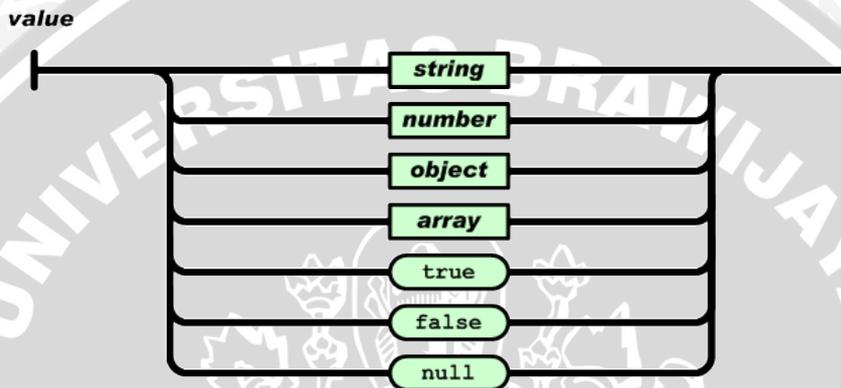


Gambar 2.7 Array pada JSON

Sumber: (Json, 2015)

3. Value

Value dapat berupa *string* di dalam dalam *double quotes* ("), *number*, *true*, *false*, *null*, *object*, atau *Array*, seperti yang terlihat pada gambar 2.8 dibawah ini.

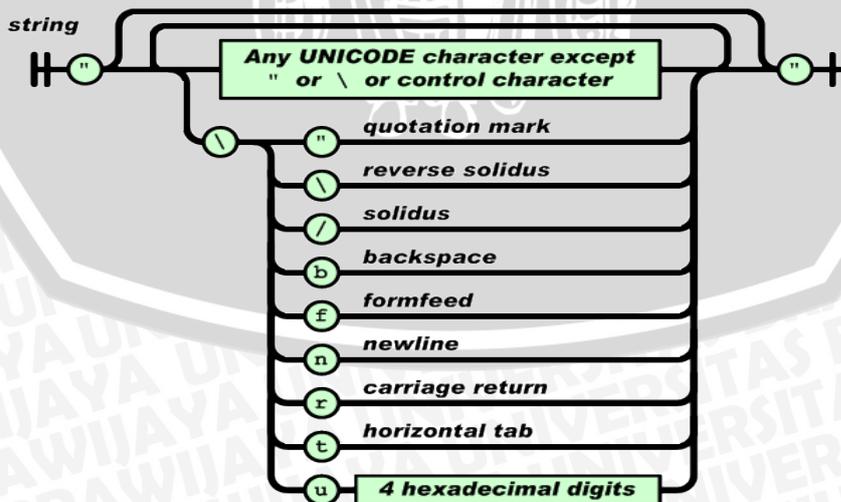


Gambar 2.8 Value pada JSON

Sumber: (Json, 2015)

4. String

String merupakan rangkaian dari nol atau lebih karakter *Unicode* yang dibungkus dengan *double quotes* ("). *String* tunggal dinamakan karakter. *String* hampir sama seperti *string* yang ada pada bahasa C atau Java, seperti yang terlihat pada gambar 2.9 dibawah ini.

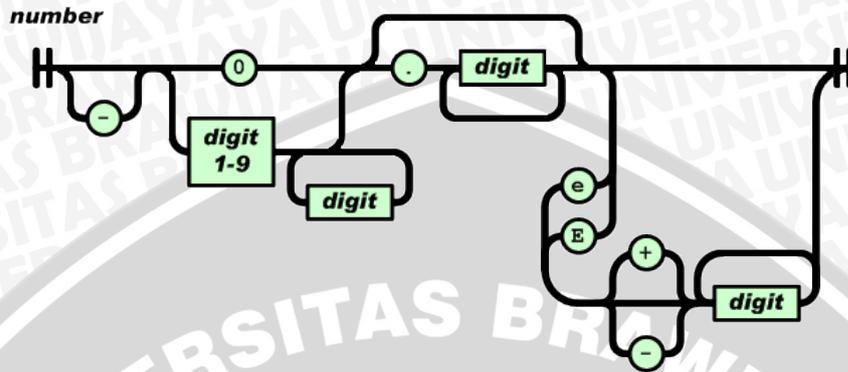


Gambar 2.9 String pada JSON

Sumber: (Json, 2015)

5. Number

Number hampir sama seperti yang ada pada bahasa C dan Java, terkecuali tidak digunakannya oktal dan heksadesimal, seperti yang terlihat pada gambar 2.10 dibawah ini.



Gambar 2.10 Objek pada JSON

Sumber: (Json, 2015)

2.13 Kohana

Kohana merupakan *framework open source* percabangan dari *Code Igniter (CI)*, berorientasi pada objek MVC (*Model, View, Controller*). Kerangka *web* dibangun dibawah *lisensi* BSD menggunakan PHP5 dengan tujuan agar lebih cepat, aman dan kecil.

MVC adalah (Efifa, 2012):

- Model merupakan data pada aplikasi yang beroperasi. Biasanya sebuah *database*.
- *View* berisi kode presentasi seperti HTML, CSS dan JavaScript.
- *Controller* menafsirkan input dari *user* dan mengirimkan kepada model dan / *View*.

Fitur utama pada kohana versi 3.0.3 (Efifa, 2012):

- Sangat aman
- Sangat ringan
- Kurva pembelajaran yang pendek
- Menggunakan pola MVC
- 100% UTF-8 yang kompatibel
- Sangat mudah untuk dikembangkan

jQuery adalah sebuah *Library Javascript* yang cepat dan ringkas yang mana dapat menyederhakan *document traversing, event handling, animating, dan Ajax interactions* untuk pengembangan *web* secara cepat (Efifa, 2012).

2.14 KML/KMZ

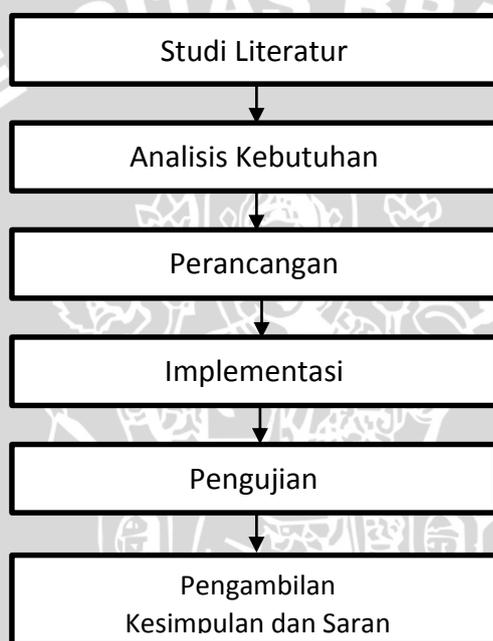
KML (*Keyhole Markup Language*) adalah *format file* yang digunakan untuk menampilkan data geografis dan visualisasi dalam *Earth browser*, seperti Google Earth, Google Maps, dan Google Maps untuk seluler, dengan berbasis internet. Peta yang ditampilkan dapat berbentuk peta 2 dimensi dan 3 dimensi. *File KML* dapat digunakan untuk menentukan lokasi, menambahkan *overlay* gambar, dan mengekspos data yang kaya dengan cara baru (Indrayani, 2013).

File KML sangat sering didistribusikan dalam file KMZ, di mana file zip KML disimpan dengan ekstensi kmz. Isi dari *file KMZ* adalah dokumen *single root KML* (notionally "doc.kml") dan opsional *layer*, gambar, ikon, dan model 3D COLLADA yang direferensikan dalam jaringan file KML. Dokumen root KML merupakan hasil konvensi dengan *file* bernama "doc.kml" yang mana merupakan *file* yang dimuat pada pembukaan. Dengan konvensi dokumen *root KML* pada *direktori root* maka *file* direferensikan dalam subdirektori (misalnya gambar untuk gambar *overlay*) (Indrayani, 2013).



BAB 3 METODOLOGI

Pada bab metodologi penelitian dijelaskan langkah-langkah yang dilakukan pada penelitian ini. Langkah-langkah yang dilakukan adalah studi literatur terhadap teori-teori pendukung dan yang terkait dengan perancangan aplikasi Informasi Tanah. Tahapan selanjutnya analisis Kebutuhan terhadap data-data yang dibutuhkan dalam perancangan aplikasi Informasi Tanah. Setelah itu merancang aplikasi yang akan dibangun berdasarkan analisis kebutuhan. Tahapan berikutnya mengimplementasikan aplikasi Informasi Tanah berdasarkan analisis dan rancangan. Tahapan selanjutnya pengujian aplikasi yang telah dibangun apakah telah sesuai dengan kebutuhan dan rancangan yang dibangun. Tahap akhir ialah melakukan analisis hasil terhadap hasil uji coba.



Gambar 3.1 Diagram Alir Tahapan Penelitian

3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan sebagai pendukung dalam penulisan skripsi ini. Studi literatur diperlukan untuk menambah pengetahuan penulis sehingga dapat meningkatkan kualitas penelitian yang dibuat. Teori pendukung tersebut meliputi:

1. Kajian Pustaka
2. Petugas Lapangan BPN Kanwil Jawa Timur
3. GPS Pada Smartphone
4. Google Earth
5. Android
6. Phonegap

7. HTML
8. CSS
9. JavaScript
10. JQuery UI
11. Wep API
12. JSON
13. Kohana
14. KML/LMZ

3.2 Analisis Kebutuhan

Analisis kebutuhan diperlukan sebagai dasar dalam pembuatan aplikasi. Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan oleh aplikasi kebutuhan diperoleh melalui identifikasi kebutuhan apa saja yang ingin didapatkan pada aplikasi (*requirements*). Kemudian dilakukan pendekatan dengan cara memodelkan *requirement* yang telah didapatkan menggunakan Bahasa pemodelan UML (*Unified Modelling Language*).

Proses yang dilakukan dalam tahap analisis terhadap kebutuhan aplikasi ialah dengan membuat gambaran umum mengenai aplikasi, melakukan identifikasi *actor-actor* yang terlibat di dalam aplikasi, melakukan penjabaran mengenai kebutuhan fungsional dan non-fungsional yang diperlukan dalam pembuatan aplikasi. Penjabaran kebutuhan fungsional yang dimodelkan ke dalam diagram *use case*.

3.3 Perancangan

Perancangan aplikasi dilakukan setelah semua kebutuhan yang diperlukan untuk pembuatan aplikasi terpenuhi. Perancangan aplikasi berdasarkan *Object Oriented Analysis* dan *Object Oriented Design* menggunakan bahasa pemodelan UML (*Unified Modeling Language*). Perancangan dimulai dengan pembuatan arsitektur aplikasi secara keseluruhan. Kemudian membuat *activity* diagram berdasarkan alur aktifitas yang dilakukan pengguna secara bertahap dalam diagram *use case* yang telah dibuat pada tahap analisis kebutuhan. Dilanjutkan dengan membuat diagram *sequence* dan *page flow* tampilan aplikasi berdasarkan diagram *activity*. Selanjutnya, membuat *class* diagram dan merancang basis data menggunakan diagram konseptual entitas untuk mendukung pembuatan aplikasi.

3.4 Implementasi

Implementasi dilakukan dengan mengacu kepada perancangan aplikasi. Implementasi dilakukan dengan menggunakan metode *hybrid* Android yaitu menggabungkan antara *platform web* dengan *native* bahasa android, dalam hal ini menggunakan bahasa Java menggunakan aplikasi Phonegap.

Untuk platform web, menggunakan HTML 5, CSS 3, dan JQuery Mobile, yang dikembangkan menggunakan IDE (*Integrated Development Environment*) Android Studio dan NetBeans IDE, kemudian PHP & MySQL sebagai *web manager* (API) penyedia datanya.

Implementasi *user interface* didasarkan pada perancangan yang sudah dilakukan. Pada tahap akhir, dilakukan simulasi aplikasi yang telah dibuat pada *hardware virtual* yang telah terhubung dengan Android Studio.

3.5 Pegujian

Pengujian perangkat lunak dibutuhkan untuk menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan (*requirement*) yang telah ditentukan. Agar pengujian pada perangkat lunak dapat berjalan dengan baik dan dapat mengetahui kekurangan yang ada pada aplikasi, maka dibutuhkan teknik pengujian perangkat lunak. Teknik pengujian perangkat lunak yang digunakan dalam penelitian ini adalah *teknik black box testing*. Teknik *black box testing* merupakan strategi di mana pengujian pada aplikasi dilakukan pada tiap fungsionalitas secara utuh.

Metode yang dipakai dalam pengujian yang akan dilakukan pada “Rancang Bangun Aplikasi Layanan Informasi Tanah Untuk Petugas Lapangan Kanwil BPN Jawa Timur Menggunakan Platform Android” menggunakan *validation testing*. Pengujian dilakukan untuk mengetahui kesiapan aplikasi untuk diimplementasikan.

Validation testing merupakan salah satu metode yang ada dalam pengujian *black box testing*. Pengujian dengan metode *validation testing* mengacu pada pemeriksaan kepada sekumpulan aktifitas yang berbeda yang dilakukan untuk memastikan bahwa aplikasi telah mengimplementasikan fungsi sesuai dengan *requirements* yang telah didefinisikan sebelumnya.

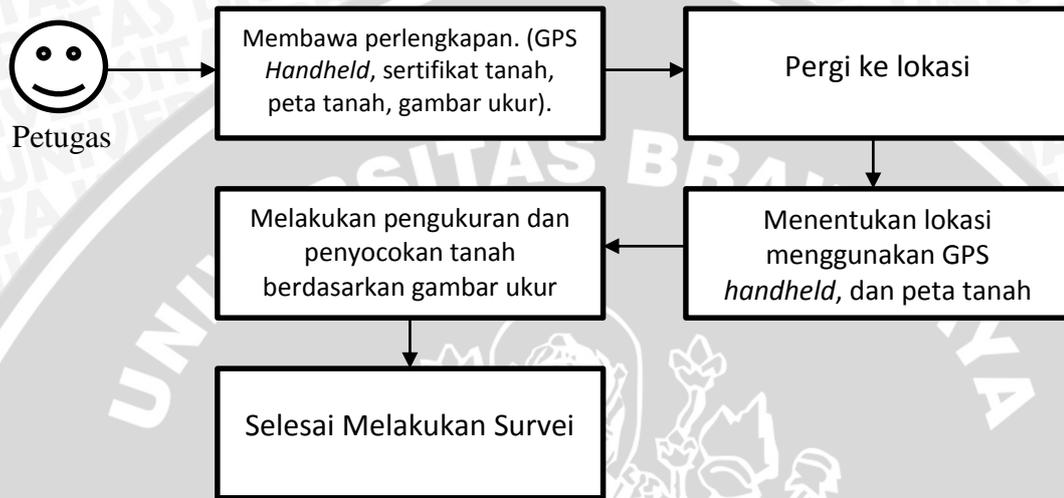
Pada tahap pengujian ini, dilakukan evaluasi setelah pengujian *validation testing* dilakukan yakni melakukan pengecekan apakah sudah sesuai dengan kriteria keberhasilan sehingga menghasilkan data yang valid. Jika hasil pengecekan terhadap pengujian yang telah dilakukan memenuhi kriteria keberhasilan, maka akan dibuat laporan mengenai kegiatan uji aplikasi dari tiap tes yang dilakukan.

3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan selesai dilakukan, mulai dari analisis, perancangan, implementasi serta pengujian. Kesimpulan diambil dari hasil pengujian terhadap aplikasi yang telah dibangun. Tahap terakhir penulisan adalah saran yang dimaksudkan untuk memperbaiki kekurangan yang terjadi dan menyempurnakan penulisan serta mengembangkan penelitian lebih lanjut.

BAB 4 REKAYASA KEBUTUHAN

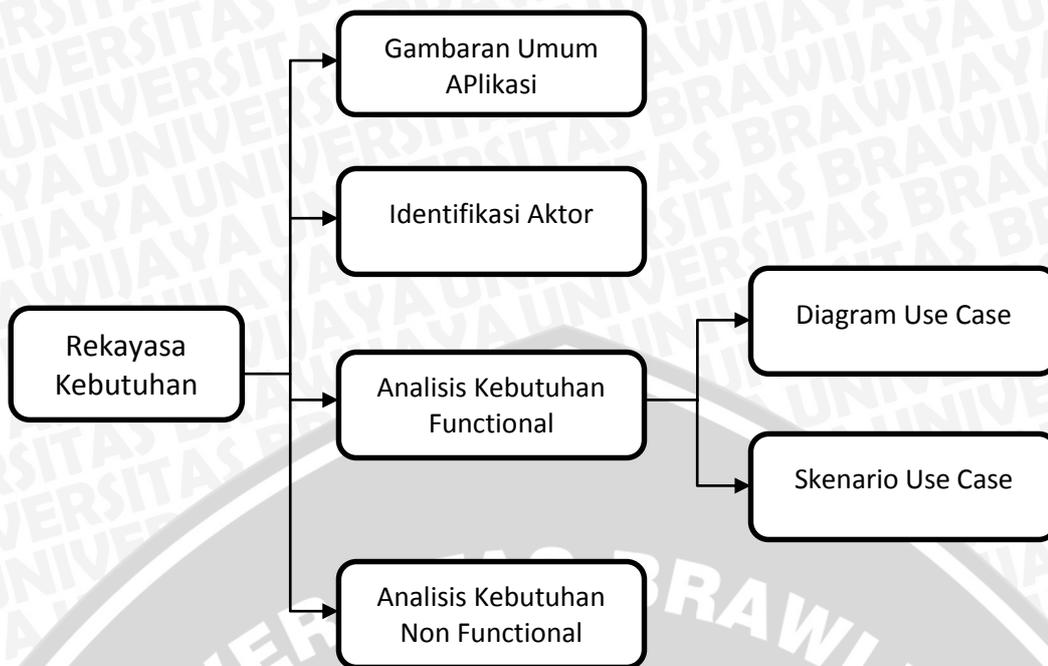
Pada bab ini akan dibahas mengenai analisis kebutuhan dan perancangan perangkat lunak. Saat ini dalam pendataan tanah yang dilakukan BPN masih terbilang manual. Banyaknya dokumentasi berupa sertifikat tanah, peta lokasi, dan gambar ukur, dan serta peralatan berat lainnya seperti *GPS Handheld* yang merupakan perlengkapan wajib bagi petugas ukur BPN saat ini. Gambar 4.1 akan menjelaskan tahapan BPN saat ini.



Gambar 4.1 Diagram alur kondisi saat ini

Gambar 4.1 menjelaskan tentang bagaimana cara kerja petugas lapangan BPN kanwil Jawa Timur saat ini. Pertama petugas lapangan mempersiapkan perlengkapan yang diperlukan seperti *GPS Handheld*, sertifikat tanah, peta ukur, dan gambar ukur. Setelah perlengkapan lengkap petugas lapangan BPN kanwil Jawa Timur pergi ke lokasi dan menentukan lokasi dengan cara menyocokkan denah peta tanah dengan *GPS Handheld*. Setelah ditemukan lokasi yang sesuai kemudian petugas lapangan BPN kanwil Jawa Timur melakukan pengukuran, penyocokan tanah berdasarkan data dan gambar ukur. Setelah sesuai maka kegiatan survei telah berakhir.

Tahapan analisis kebutuhan dan perancangan ini terdiri dari penjabaran gambaran umum aplikasi dan membuat daftar kebutuhan pengguna dengan menganalisa kebutuhan *functional* dan *non-functional*. Pada tahapan perancangan terdiri dari perancang arsitektur sistem, perancangan *activity* diagram, perancangan *sequence* diagram, perancangan *class* diagram, perancangan basis data, perancangan antarmuka aplikasi, seperti yang terlihat pada gambar 4.2.



Gambar 4.2 Diagram Pohon Rekayasa Kebutuhan

4.1 Analisis Kebutuhan

Tahapan analisis kebutuhan diawali dengan penjabaran mengenai gambaran umum aplikasi, kemudian penjabaran mengenai kebutuhan fungsional dan non-fungsional. Analisis kebutuhan diperlukan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem, supaya sesuai dengan kebutuhan pengguna.

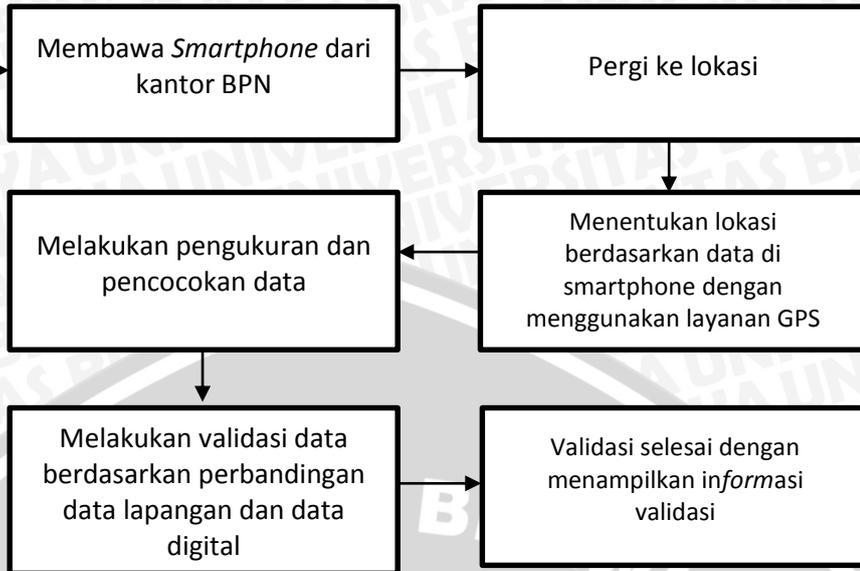
4.1.1 Gambaran Umum Aplikasi

Aplikasi informasi tanah merupakan metode baru bagi petugas BPN kanwil Jawa Timur dalam melakukan survei lapangan. Metode atau cara yang digunakan memerlukan perlengkapan yang tidak sedikit dan prakteknya yang memakan waktu banyak dinilai kurang efisien dalam melakukan survei lapangan. Dalam hal ini pengguna dapat melihat letak posisi lokasi, denah suatu tanah, dan informasi tanah, dan juga petugas dapat mengaktifkan GPS dari perangkat bergerak, sehingga petugas dapat mengetahui posisi yang sebenarnya.

Selain dapat mengetahui informasi suatu bidang tanah, pengguna juga dapat menambah suatu bidang *polygon* yang berisi informasi dengan mengupload KML/KMZ. KML/KMZ yang diupload dapat didapat dengan membuat bidang *polygon* dengan menggunakan aplikasi *Google earth*. Hasil *upload* tersebut kemudian disimpan di *web manager* dan dapat diakses melalui perangkat bergerak tersebut. Gambar 4.3 akan menjelaskan cara kerja aplikasi yang diajukan untuk melakukan survei lapangan BPN kanwil Jawa Timur.



Petugas



Gambar 4.3 Diagram alur sistem yang ditawarkan

Pada gambar 4.3 menjelaskan tentang bagaimana cara kerja sistem yang ditawarkan, dimana petugas BPN dalam pelaksanaan tugas dilapangan cukup membawa *smartphone* dan mengaktifkan GPS yang sudah terinstall pada perangkat tersebut, kemudian petugas dapat menentukan lokasi berdasarkan data di *smartphone* dengan menggunakan layanan GPS tersebut. Setelah menentukan lokasi dilakukan pengukuran data dan pencocokan data, agar dapat dilakukan *validasi* data berdasarkan perbandingan data lapangan dengan data digital. Hasil *validasi* pada aplikasi layanan ini berupa informasi *validasi*.

4.2 Identifikasi Aktor

Dalam tahapan ini akan dilakukan identifikasi *actor* yang akan berinteraksi dengan aplikasi informasi tanah. Pada Table 4.1 berikut ini memperlihatkan aktor yang terlibat.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
User	Pengguna merupakan petugas lapangan BPN

4.3 Kebutuhan Fungsional

Pada tahapan analisis kebutuhan fungsional ini akan didaftar kebutuhan apa saja yang diperlukan dalam membangun aplikasi informasi tanah, yang dispesifikasikan seperti yang dijelaskan pada Tabel 4.2.



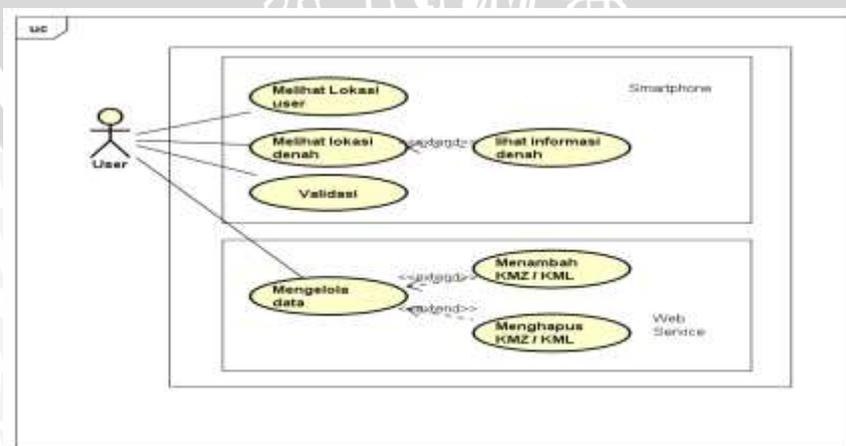
Tabel 4.2 Daftar Kebutuhan Fungsional

No	Kebutuhan Fungsional	Use case
1	Aplikasi harus menyediakan fasilitas lokasi <i>user</i> saat ini.	Melihat lokasi <i>user</i>
2	Aplikasi harus menyediakan fasilitas lokasi denah (<i>polygon</i>) yang tersimpan di <i>database</i>	Melihat lokasi denah
3	Aplikasi harus menyediakan fasilitas melihat informasi <i>detail</i> denah yang tersimpan di <i>database</i>	Melihat informasi lokasi
4	Aplikasi harus menyediakan fasilitas mengelola data yang berada pada <i>database</i>	Mengelola data
5	Aplikasi harus menyediakan fasilitas menambah KML/KMZ yang disimpan di <i>database</i>	Menambah KML/KMZ
6	Aplikasi harus menyediakan fasilitas menghapus KML/KMZ yang berada di <i>database</i>	Menghapus KML/KMZ
7	Aplikasi harus menyediakan fasilitas validasi yang membandingkan data dilapangan dan data digital.	Validasi

Tabel 4.2 merupakan *table* daftar kebutuhan fungsional yang terdiri dari 6 kebutuhan aplikasi yang terdapat pada aplikasi. Selanjutnya setiap kebutuhan aplikasi dimodelkan dalam bentuk diagram *use case*.

4.3.1 Diagram Usecase

Diagram *use case* merupakan salah satu diagram untuk memodelkan perilaku sekumpulan *use case*, *actor* dan relasinya pada sistem. *Use case* merupakan bagian fungsionalitas dari sistem yang diinisiasi oleh *actor*. Gambar 4.4 menunjukkan diagram *use case* aplikasi informasi tanah.



Gambar 4.4 Diagram Use case



Pada Gambar 4.4 menunjukkan diagram *use case* aplikasi informasi tanah yang hanya terdiri dari 1 aktor dan 6 *use case* yang diantaranya adalah melihat lokasi *user*, melihat lokasi denah yang meliputi lihat informasi denah, mengelola data yang meliputi menambah dan menghapus *file* KML/KMZ.

4.3.2 Skenario *Use case*

Use case yang terbentuk pada diagram *use case* kemudian dijabarkan lebih detail pada skenario *use case*. Di dalam skenario *use case* akan berisi uraian nama, aktor, tujuan, dan deskripsi tentang *use case*. Serta terdapat kondisi awal yang harus dipenuhi (*pre-condition*) dan kondisi akhir yang diharapkan (*post-condition*).

4.3.2.1 Skenario melihat lokasi *user*

Skenario melihat lokasi *user* dijelaskan dalam bentuk *user* dimana terdapat nama *use case* yang tersiri dari ID, aktor, tujuan, deskripsi, *pre-condition*, *post-condition*, dan aliran utama serta aliran *alternative*, dapat dilihat pada tabel 4.3 dibawah ini.

Tabel 4.3 Skenario melihat lokasi *user*

Nama <i>use case</i>	Melihat lokasi <i>user</i>
<i>Use case ID</i>	UC-01
Aktor	Pengguna
Tujuan	Melihat lokasi <i>user</i>
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana pengguna dapat melihat lokasi <i>user</i> yang direpresentasikan dalam bentuk pin.
<i>Pre-condition</i>	Pengguna harus membuka aplikasi terlebih dahulu sebelum <i>use case</i> dimulai. Saat pertama kali aplikasi dijalankan akan langsung masuk pada halaman utama aplikasi.
<i>Post-condition</i>	Pengguna dapat melihat posisi <i>user</i> saat ini.
Aliran Utama	
Aksi dari Aktor	Respon dari Aplikasi
1. Pengguna membuka aplikasi 2. <i>User</i> mengaktifkan fitur GPS yang berada pada <i>smartphone</i>	1. Sistem menampilkan koordinat <i>user</i> berdasarkan <i>longitude</i> dan <i>latitude</i> . 2. Menampilkan posisi <i>user</i> dalam bentuk pin.
Aliran Alternatif	
Aksi dari Aktor	Respon dari Aplikasi

4.3.2.2 Skenario melihat lokasi denah

Skenario melihat lokasi denah dijelaskan dalam bentuk *polygon* dimana terdapat nama *use case* yang tersiri dari ID, aktor, tujuan, deskripsi, *pre-condition*, *post-condition*, dan aliran utama serta aliran *alternative*, dapat dilihat pada tabel 4.4 dibawah ini.

Tabel 4.4 Skenario melihat lokasi *user*

Nama <i>use case</i>	Melihat lokasi denah
<i>Use case ID</i>	UC-02
Aktor	Pengguna
Tujuan	Melihat lokasi denah
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana melihat lokasi-lokasi denah yang terdapat dalam <i>database</i> aplikasi yang direpresentasikan dengan <i>polygon</i>
<i>Pre-condition</i>	<i>user</i> memilih denah lokasi mana yang ingin ditampilkan dalam aplikasi
<i>Post-condition</i>	<i>user</i> akan mendapatkan <i>polygon</i> denah lokasi yang ingin ditampilkan
Aliran Utama	
Aksi dari Aktor	Respon dari Aplikasi
<i>User</i> memilih denah yang diinginkan untuk ditampilkan dalam aplikasi.	Sistem akan menampilkan denah yang diinginkan <i>user</i> dalam bentuk <i>polygon</i> .
Aliran Alternatif	
Aksi dari Aktor	Respon dari Aplikasi

4.3.2.3 Skenario melihat informasi lokasi

Pada skenario melihat informasi lokasi memungkinkan *user* mendapatkan informasi yang terkandung di dalam setiap denah yang terdapat di *database* dimana terdapat nama *use case* yang tersiri dari ID, aktor, tujuan, deskripsi, *pre-condition*, *post-condition*, dan aliran utama serta aliran *alternative*, dapat dilihat pada tabel 4.5 dibawah ini.

Tabel 4.5 Skenario melihat informasi lokasi

Nama <i>use case</i>	Melihat informasi lokasi
<i>Use case ID</i>	UC-03
Aktor	Pengguna
Tujuan	Melihat informasi lokasi tanah
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana pengguna dapat melihat informasi dari setiap <i>polygon</i> tanah.

Pre-condition	aplikasi dalam keadaan menampilkan salah satu denah yang terdapat dalam <i>database</i> .
Post-condition	<i>user</i> akan mendapatkan informasi yang terkandung dalam setiap denah lokasi.
Aliran Utama	
Aksi dari Aktor	Respon dari Aplikasi
<ol style="list-style-type: none"> <i>User</i> memilih denah yang diinginkan ditampilkan dalam aplikasi <i>User</i> melihat detail denah 	<ol style="list-style-type: none"> Sistem akan menampilkan denah yang diinginkan <i>user</i> dalam bentuk <i>polygon</i>. Sistem akan menampilkan informasi yang terkandung dalam bidang <i>polygon</i>.
Aliran Alternatif	
Aksi dari Aktor	Respon dari Aplikasi

4.3.2.4 Skenario mengolah data

Skenario mengolah data dijelaskan dalam bentuk informasi berupa *text*, dimana pada tabel skenario mengolah data terdapat nama *use case* yang terdiri dari ID, aktor, tujuan, deskripsi, *pre-condition*, *post-condition*, dan aliran utama serta aliran *alternative*, dapat dilihat pada tabel 4.6 dibawah ini.

Tabel 4.6 Skenario mengolah data

Nama <i>use case</i>	Mengolah data
Use case ID	UC-04
Aktor	Pengguna
Tujuan	Untuk merubah data yang ada dalam bentuk <i>digital</i> .
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana merubah data yang ada, perubahan yang dimaksud seperti menambah <i>polygon</i> , merubah informasi yang terkandung dalam setiap <i>polygon</i> ataupun menghapusnya.
Pre-condition	<i>User</i> membuka halaman <i>web</i> yang digunakan sebagai <i>server</i> dari aplikasi ini.
Post-condition	Data yang terkandung dapat dikelola <i>user</i> .

Aliran Utama	
Aksi dari Aktor	Respon dari Aplikasi
1. <i>User</i> membuka halaman <i>web</i> yang digunakan sebagai <i>server</i> dari aplikasi	<i>Server</i> menampilkan CRUD (<i>Creat, Read, Update, Delete</i>),
2. <i>User</i> melakukan pengolahan data.	
Aliran Alternatif	
Aksi dari Aktor	Respon dari Aplikasi

4.3.2.5 Skenario menambah KML/KMZ

Skenario menambah KML/KMZ memungkinkan *user* untuk menambah data yang ada, didalam tabel skenario menambah KML/KMZ terdapat nama *use case* yang tersiri dari ID, aktor, tujuan, deskripsi, *pre-condition*, *post-condition*, dan aliran utama serta aliran *alternative*, dapat dilihat pada tabel 4.7 dibawah ini.

Tabel 4.7 Skenario menambah KML/KMZ

Nama <i>use case</i>	Menambah KML/KMZ
<i>Use case ID</i>	UC-05
Aktor	Pengguna
Tujuan	Melakukan penambah data pada <i>web manager</i>
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana melakukan penambahan data berbentuk <i>file</i> KML/KMZ pada <i>web manager</i>
<i>Pre-condition</i>	<i>User</i> membuka halaman <i>web</i> yang digunakan sebagai <i>server</i> dari aplikasi ini.
<i>Post-condition</i>	Data yang terkandung dapat dikelola <i>user</i>
Aliran Utama	
Aksi dari Aktor	Respon dari Aplikasi
1. <i>User</i> membuka halaman <i>web</i> yang digunakan sebagai <i>server</i> dari aplikasi	<i>Server</i> menampilkan CRUD (<i>Creat, Read, Update, Delete</i>)
2. <i>User</i> melakukan penambahan data	
Aliran Alternatif	
Aksi dari Aktor	Respon dari Aplikasi

4.3.2.6 Skenario Menghapus KML/KMZ

Skenario menghapus KML/KMZ memungkinkan *user* untuk menghapus data yang ada, didalam tabel skenario menambah KML/KMZ terdapat nama *use case* yang tersiri dari ID, aktor, tujuan, deskripsi, *pre-condition*, *post-condition*, dan aliran utama serta aliran *alternative*, dapat dilihat pada tabel 4.8 dibawah ini.

Tabel 4.8 Skenario menghapus KML/KMZ

Nama <i>use case</i>	Menghapus KML/KMZ
Use case ID	UC-06
Aktor	Pengguna
Tujuan	Melakukan penghapusan data pada <i>web manager</i>
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana melakukan penghapusan data berbentuk <i>file</i> KML/KMZ pada <i>web manager</i>
Pre-condition	<i>User</i> membuka halaman <i>web</i> yang digunakan sebagai <i>server</i> dari aplikasi ini.
Post-condition	Data yang terkandung dapat dikelola <i>user</i>
Aliran Utama	
Aksi dari Aktor	Respon dari Aplikasi
1. <i>User</i> membuka halaman <i>web</i> yang digunakan sebagai <i>server</i> dari aplikasi	<i>Server</i> menampilkan CRUD (<i>Creat, Read, Update, Delete</i>)
2. <i>User</i> melakukan penghapusan data	
Aliran Alternatif	
Aksi dari Aktor	Respon dari Aplikasi

4.3.2.7 Skenario validasi

Skenario validasi memungkinkan *user* untuk melakukan validasi data. Data yang terdapat pada aplikasi akan dicocokkan dengan yang ada dilapangan, didalam tabel validasi terdapat nama *use case* yang tersiri dari ID, aktor, tujuan, deskripsi, *pre-condition*, *post-condition*, dan aliran utama serta aliran *alternative*, dapat dilihat pada tabel 4.8 dibawah ini.

Tabel 4.9 Skenario validasi

Nama <i>use case</i>	Validasi
Use case ID	UC-07
Aktor	Pengguna
Tujuan	Melakukan validasi data yang terdapat pada aplikasi dengan data di lapangan

Deskripsi	<i>Use case</i> ini menjelaskan bagaimana melakukan validasi data
Pre-condition	<i>User</i> memilih denah yang akan divalidasi.
Post-condition	Aplikasi menampilkan informasi validasi.
Aliran Utama	
Aksi dari Aktor	Respon dari Aplikasi
<ol style="list-style-type: none"> 1. <i>User</i> memilih satu denah yang terdapat pada <i>right-side menu</i>. 2. <i>User</i> melakukan pengukuran dengan memasukkan koordinat lokasi. 	<ol style="list-style-type: none"> 1. Aplikasi akan menampilkan polygon denah 2. Aplikasi menerima informasi koordinat dan melakukan penggambaran polygon dan menghitung luas.
Aliran Alternatif	
Aksi dari Aktor	Respon dari Aplikasi

4.4 Kebutuhan Non-Fungsional

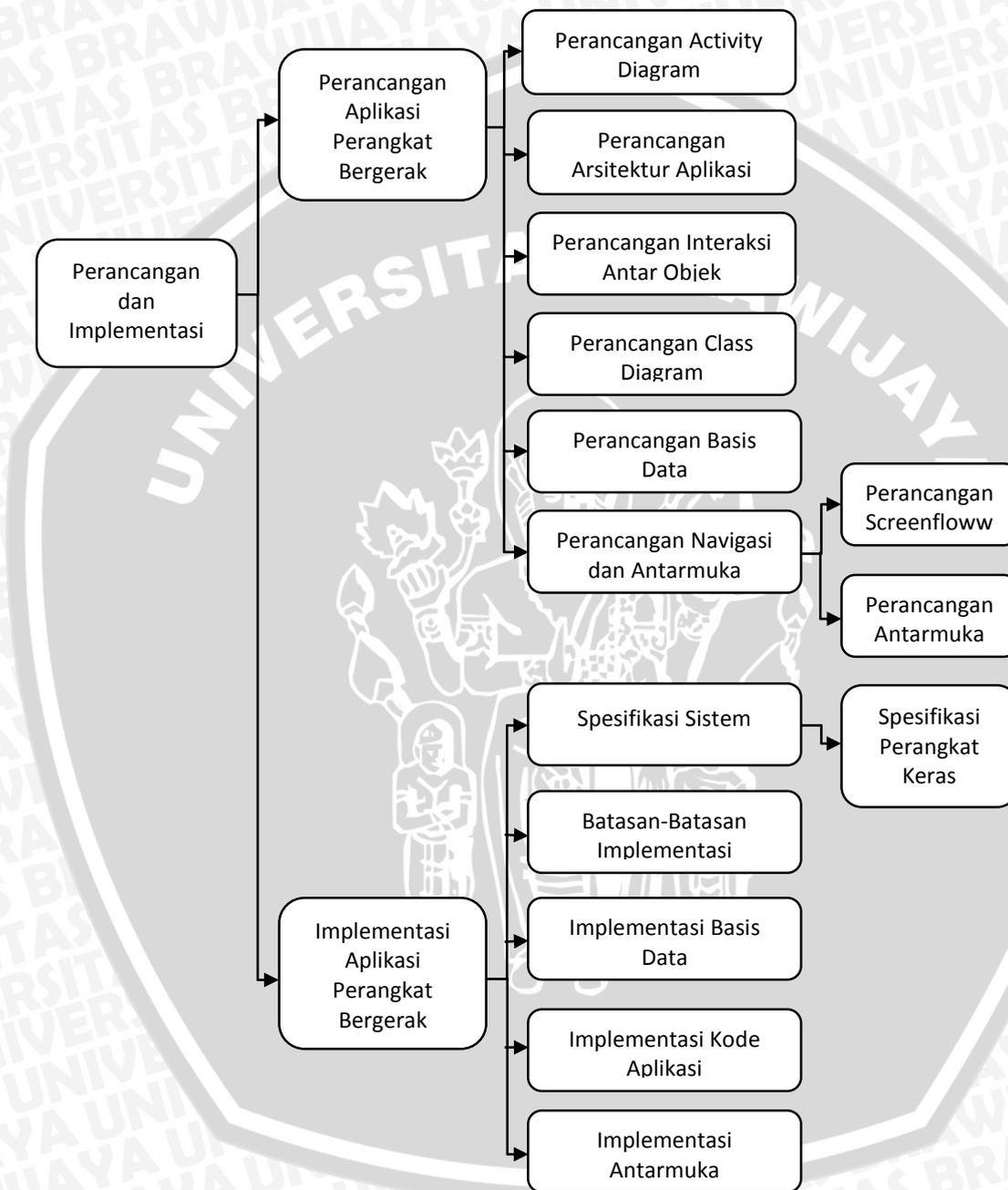
Analisis kebutuhan non fungsional adalah analisis untuk mengetahui spesifikasi yang dibutuhkan oleh sistem, dimana kebutuhan *non-fungsional* dijabarkan pada table 4.10.

Tabel 4.10 Analisis kebutuhan non-functional

Parameter	Deskripsi Kebutuhan
Usability	Aplikasi harus dapat digunakan dengan mudah oleh pengguna, dengan mempelajari fitur-fitur yang disediakan pada aplikasi informasi tanah.
Akses Data	Pada hasil implementasi pengguna dapat melakukan akses data menggunakan teknologi <i>wifi</i> maupun <i>mobile data</i> .

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini menjelaskan tentang perancangan dan implementasi. Pada Gambar 5.1 terdapat struktur dari perancangan dan implementasi dari aplikasi perangkat bergerak informasi tanah.



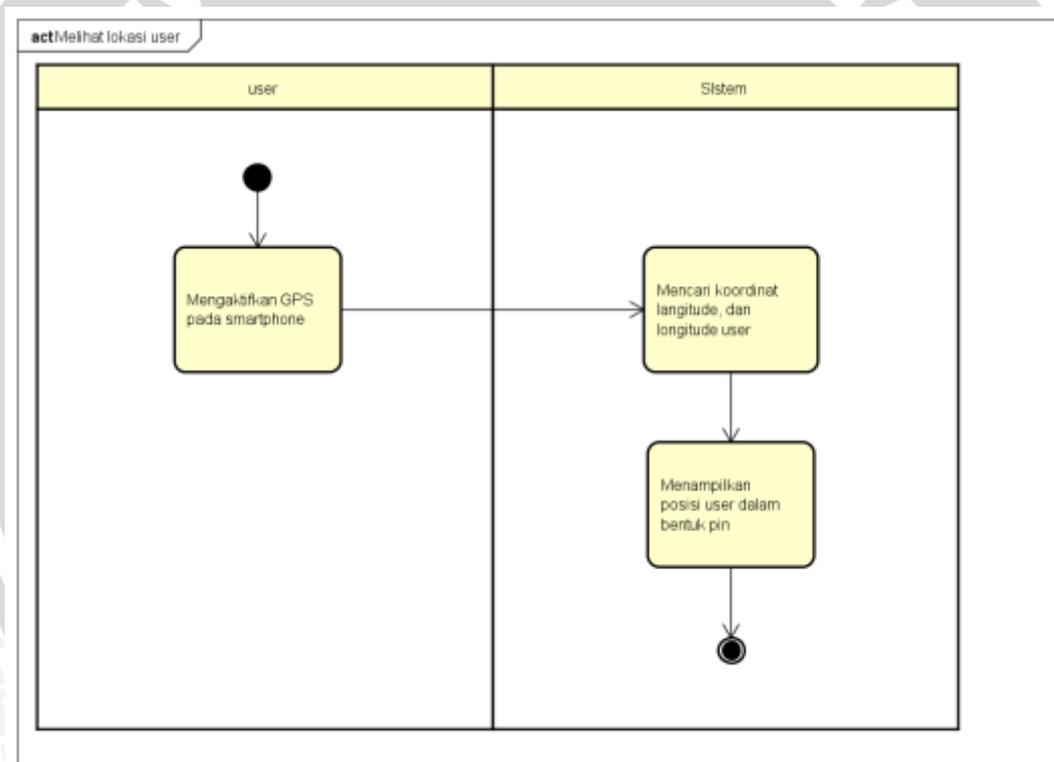
Gambar 5.1 Pohon Diagram Perancangan Dan Implementasi

5.1 Perancangan Aplikasi Perangkat Bergerak

Perancangan aplikasi perangkat bergerak ini terdiri dari beberapa tahapan yaitu perancangan aktivitas, perancangan interaksi antar objek, perancangan kelas, perancangan basis data, perancangan arsitektur aplikasi, perancangan *web manager* dan perancangan antarmuka. Tahapan perancangan arsitektur aplikasi menjelaskan tentang arsitektur aplikasi yang akan dibuat dan dikembangkan. Tahapan perancangan basis data menjelaskan entitas dan relasi masing-masing. Tahapan perancangan *web manager* menjelaskan tentang rancangan *web manager* yang akan dikembangkan dan perancangan antarmuka mendefinisikan tampilan yang di implementasikan.

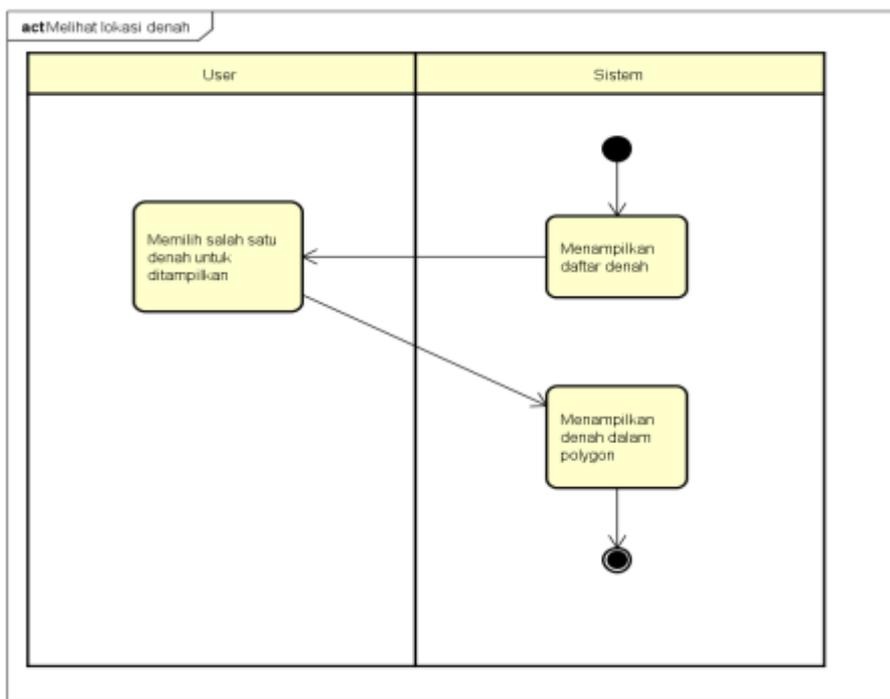
5.1.1 Activity Diagram

Activity diagram merupakan diagram yang digunakan untuk memodelkan interaksi antar aktor yang terjadi dalam *use case*. *Activity diagram* menggambarkan langkah-langkah sistem pada tiap-tiap *use case* secara bertahap.



Gambar 5.2 Activity diagram melihat lokasi user

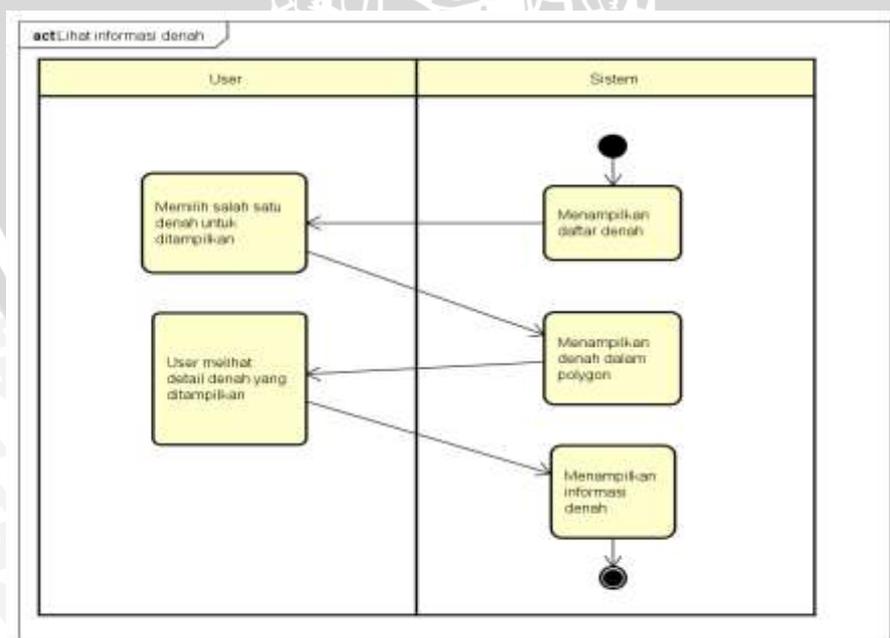
Gambar 5.2 menjelaskan aktifitas yang dilakukan *user* dan sistem dalam *use case* melihat lokasi *user*. Deskripsi *activity diagram* sesuai dengan skenario *use case* melihat lokasi *user*. Alurnya *user* mengaktifkan GPS pada *device* (*smartphone*). Sistem akan mencari posisi lokasi *user* berdasarkan *longitude* dan *latitude*, dan sistem memberikan / menampilkan lokasi *user* dalam bentuk pin.



powered by Astah

Gambar 5.3 Activity diagram melihat lokasi denah

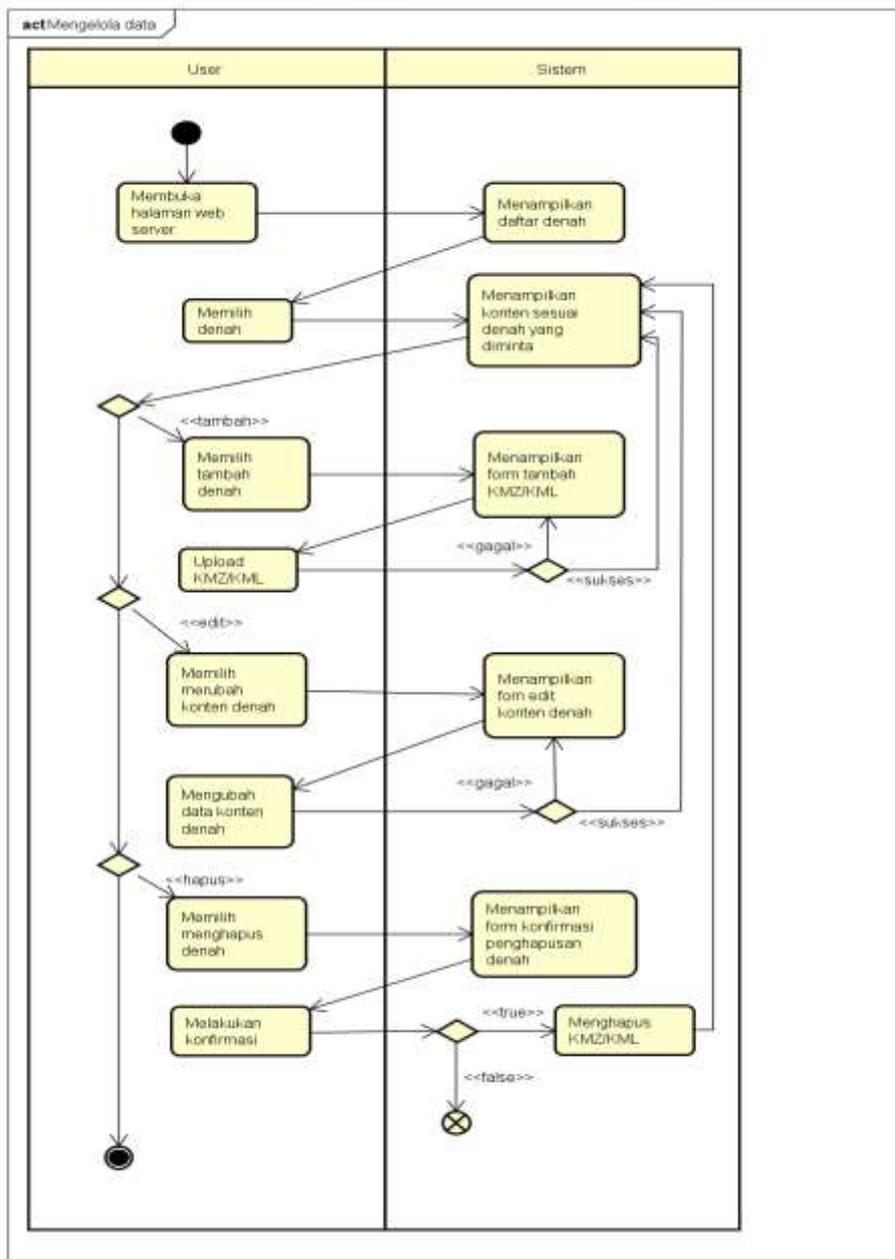
Gambar 5.3 menjelaskan aktifitas yang dilakukan *user* dan sistem dalam *use case* melihat lokasi denah, deskripsi *activity* diagram sesuai dengan skenario *use case* melihat lokasi denah. Alurnya sistem akan menampilkan daftar *list* denah yang terdapat pada sistem, kemudian *user* akan memilih salah satu denah yang ingin ditampilkan. Sistem akan merespon dan menampilkan denah yang diminta dalam bentuk *polygon*.



powered by Astah

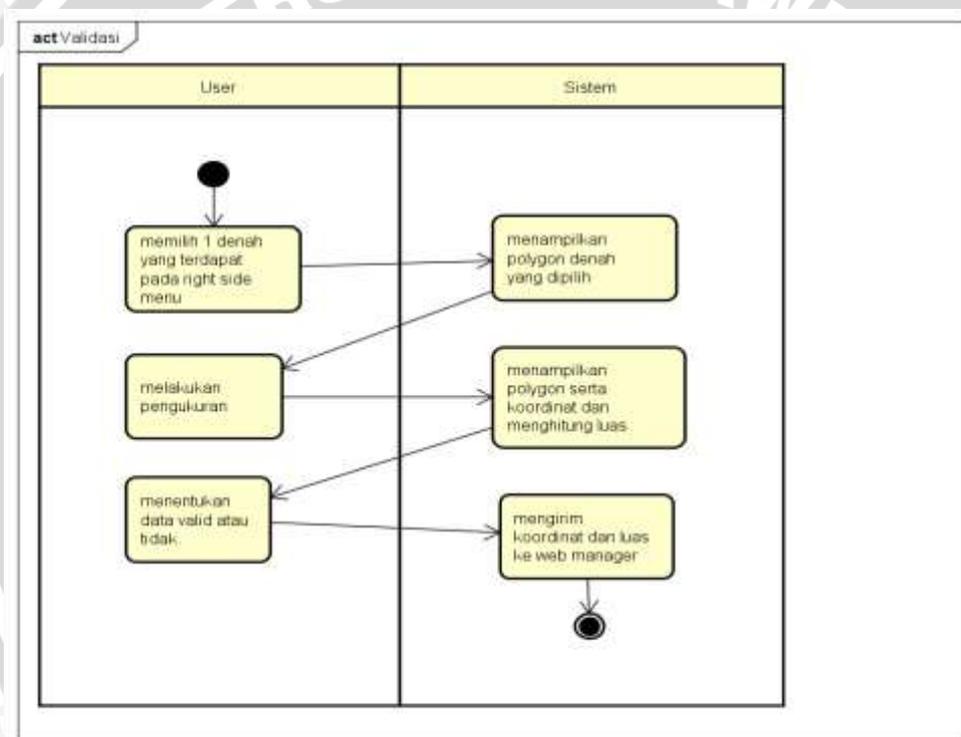
Gambar 5.4 Activity diagram lihat informasi denah

Gambar 5.4 menjelaskan aktifitas yang dilakukan *user* dan sistem dalam *use case* untuk melihat informasi denah, deskripsi *activity* diagram sesuai dengan skenario *use case* melihat informasi denah. Alurnya sama seperti melihat lokasi denah, tetapi ditambahkan detail denah sebagai sumber informasi denah. Sistem akan menampilkan *list* denah yang terdapat pada sistem. kemudian *user* akan memilih salah satu denah yang ingin ditampilkan, sistem akan merespon dan menampilkan denah yang diminta dalam bentuk *polygon*. *User* kemudian memilih detail denah dan sistem menampilkan informasi denah yang diminta.



Gambar 5.5 Activity diagram mengelola data, menambah, dan Menghapus KML/KMZ

Gambar 5.5 menjelaskan aktifitas yang dilakukan *user* dan sistem dalam *use case* mengelola data, deskripsi *activity diagram* sesuai dengan skenario *use case* mengelola data. Alurnya *user* membuka *web manager*, sistem akan menampilkan daftar denah. *User* kemudian memilih pengelolaan data yakni, tambah, *edit*, hapus. Jika *user* memilih tambah, maka tambah yang dimaksud adalah KML/KMZ denah. Sistem akan menampilkan *form* upload file KML/KMZ yang kemudian *user* memilih file KML/KMZ yang diinginkan. Kemudian sistem akan melakukan *validasi* inputan yang dilakukan *user*. Jika *user* memilih *edit* data, maka *edit* yang dimaksud adalah *edit* data konten yang terkandung dalam setiap satu denah, sistem akan menampilkan *form edit*, kemudian *user* melakukan perubahan data konten. Kemudian sistem akan melakukan *validasi* input yang telah dilakukan *user* yang dilanjutkan ke model untuk dieksekusi pada *database*, jika *user* memilih hapus, maka akan muncul *form* konfirmasi terhadap penghapusan KML/KMZ, kemudian denah akan terhapus.



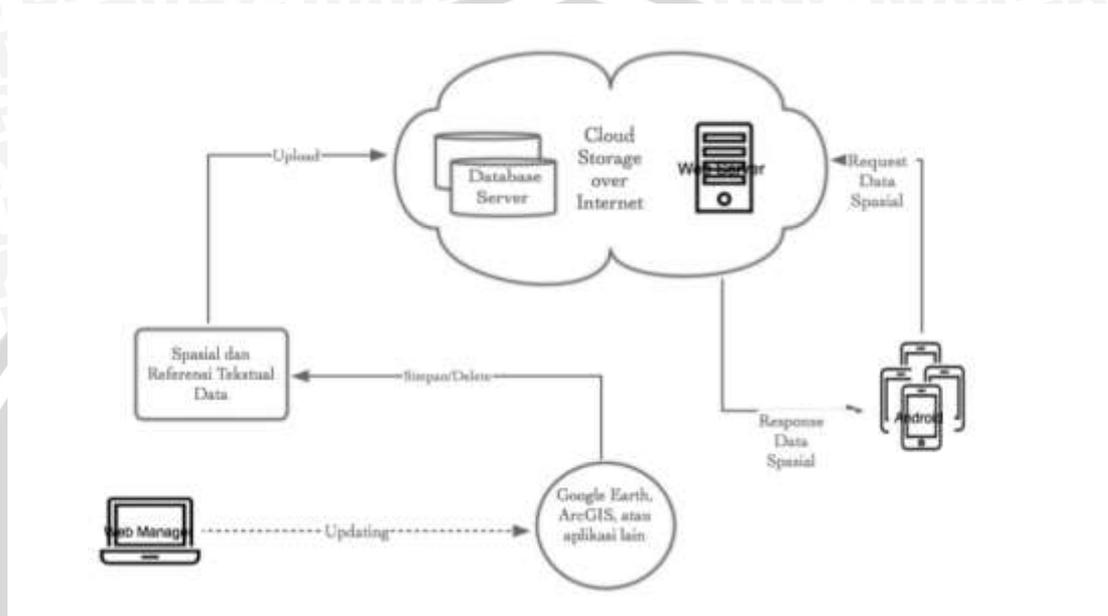
Gambar 5.6 Activity diagram validasi

Gambar 5.6 menjelaskan aktifitas yang dilakukan *user* dan sistem dalam *use case* untuk validasi, deskripsi *activity diagram* sesuai dengan skenario validasi. Alurnya *user* memilih satu denah yang terdapat pada *right-side menu*, sistem akan menampilkan *polygon* denah yang dipilih. Kemudian dilanjutkan dengan *user* melakukan pengecekan dengan melakukan pengukuran terhadap *polygon*. Setelah *user* selesai melakukan pengukuran dengan memasukkan koordinat lokasi, sistem akan melakukan penggambaran *polygon* baru beserta koordinat *longitude* dan *latitude*, dan luas *polygon*. *User* menentukan data tersebut valid atau tidak.

Kemudian sistem akan mengirim gambar *polygon*, koordinat dan perhitungan luas ke *web manager*.

5.1.2 Arsitektur Aplikasi

Perancangan arsitektur sistem ini menjelaskan mengenai gambaran kinerja sistem secara keseluruhan, pada gambar 5.7 berikut menjelaskan tentang rancangan arsitektural sistem yang akan implementasikan.



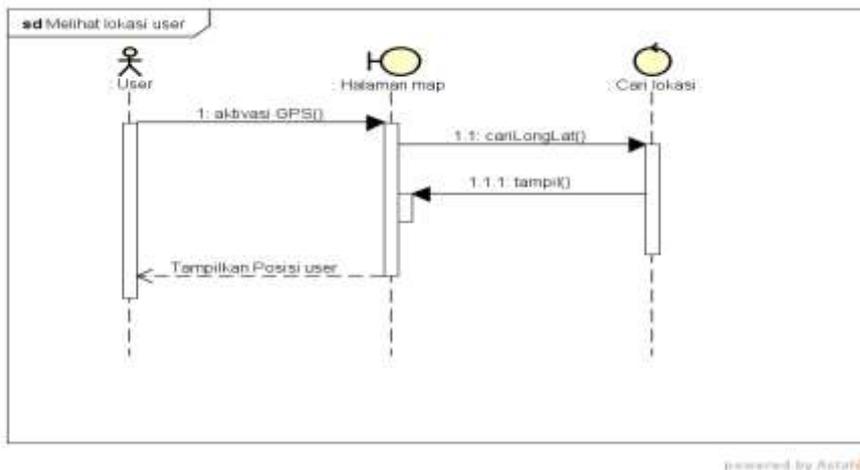
Gambar 5.7 Arsitektur sistem

Gambar 5.7 menggambarkan aplikasi sistem yang diimplementasikan untuk petugas survey lapang, mereka nantinya dapat mengakses data spasial dan referensi tekstual pada *smartphone* android, dimana telah tersimpan data tersebut pada *cloud server*, dikombinasi dengan fitur GPS atau *Location Services* yang tersedia pada *smartphone* Android, maka ini akan lebih mempermudah bagi petugas tersebut untuk mengolah data yang melekat pada tanah *object survey* tersebut.

5.1.3 Interaksi Antar Objek

Sequence diagram merupakan suatu diagram yang menggambarkan interaksi antar objek dalam bentuk urutan pertukaran pesan terhadap waktu dalam sistem, baik sisi *client* maupun *server*. *Sequence* diagram digunakan untuk merepresentasikan skenario kegiatan yang terdapat pada *use case* secara berurutan dan balasan dari tiap pesan antar objek. *Sequence* diagram untuk aplikasi informasi tanah dapat dilihat pada Gambar 5.8 sampai dengan Gambar 5.11 dibawah ini:

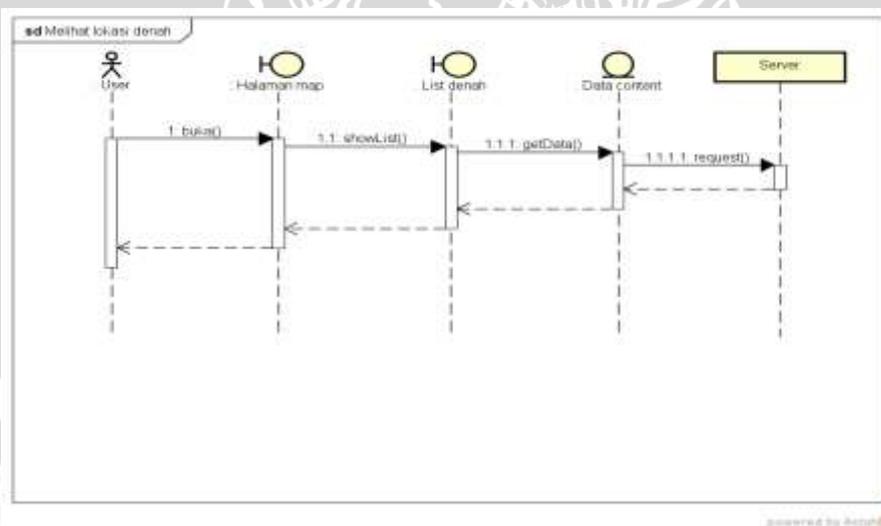
1. Sequence diagram melihat lokasi user.



Gambar 5.8 Sequence diagram melihat lokasi user

Gambar 5.8 merupakan *sequence* diagram melihat lokasi *user* yang akan menjelaskan proses masukan pengguna sampai mendapatkan lokasi *user*. Proses dimulai dengan *user* mengaktifkan GPS pada aplikasi, dan aplikasi akan mencari *longitude* dan *latitude* posisi *user*, dan menampilkannya pada aplikasi.

2. Sequence diagram melihat lokasi denah

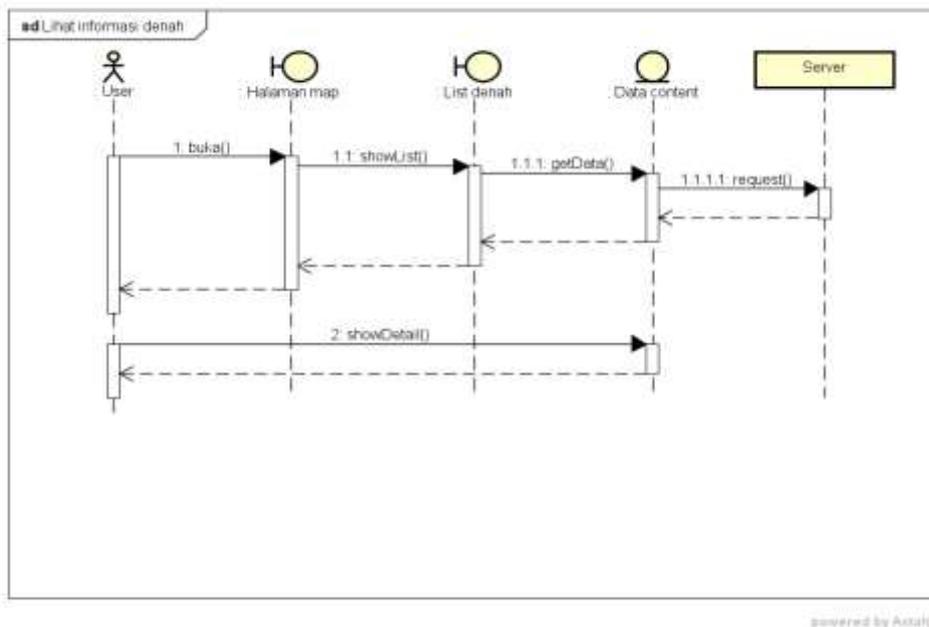


Gambar 5.9 Sequence diagram melihat lokasi denah

Gambar 5.9 merupakan *sequence* diagram melihat lokasi denah yang akan menjelaskan proses masukan pengguna sampai mendapatkan lokasi denah, dimana proses dimulai dengan *user* melihat daftar denah yang ada pada aplikasi, kemudian *user* dapat memilih denah yang terdapat pada aplikasi tersebut, setelah denah terpilih, maka aplikasi akan melakukan *request* ke data server, kemudian *server*, akan merespon permintaan *user* dengan menampilkan denah yang diinginkan tersebut.



3. Sequence diagram lihat informasi denah



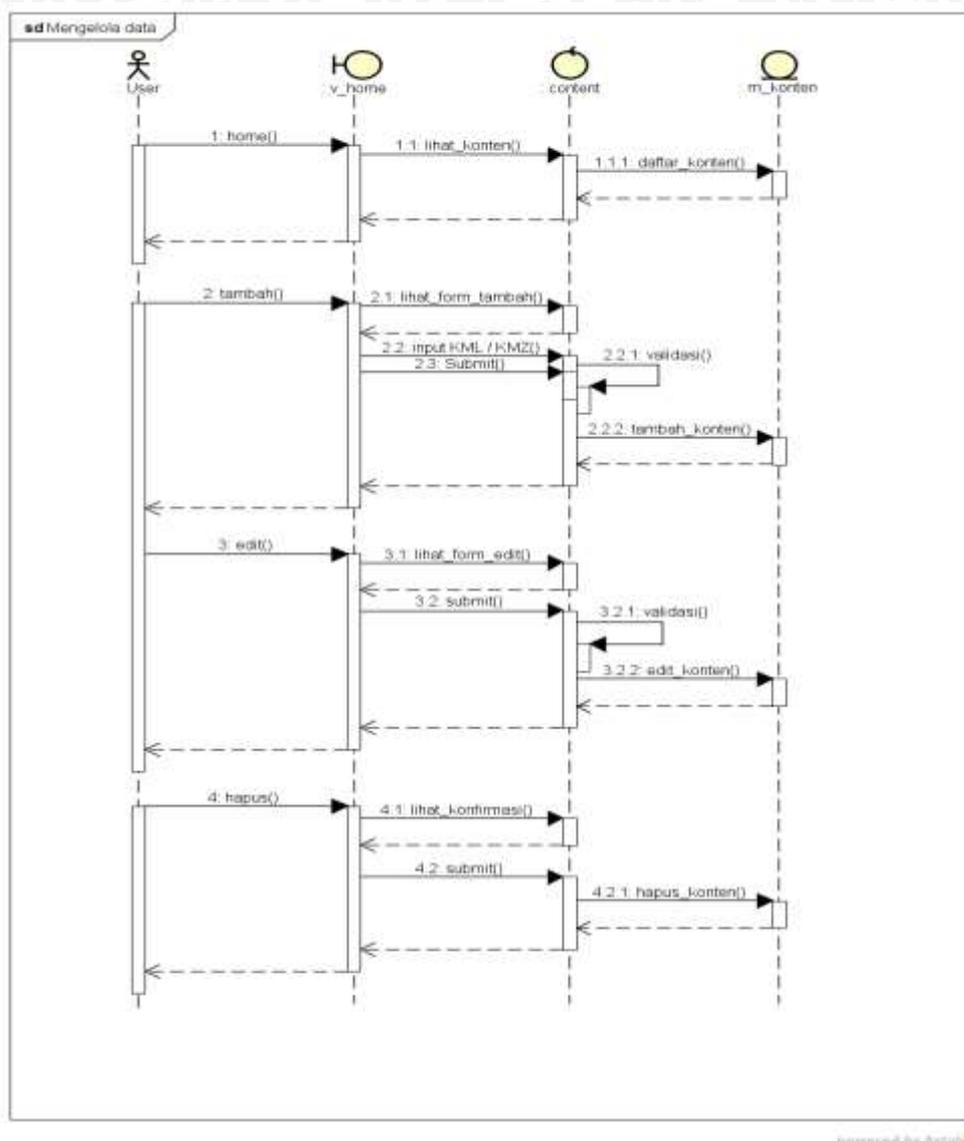
Gambar 5.10 Sequence diagram lihat informasi denah

Permintaan informasi denah akan muncul disaat *sequence* diagram melihat lokasi denah merespon daftar denah yang terdapat pada aplikasi, seperti yang terlihat pada Gambar 5.10 dimana proses *sequence* melihat informasi denah dimulai dengan *user* melihat daftar denah yang ada pada aplikasi, kemudian *server* akan merespon permintaan *user* dengan menampilkan denah yang diinginkan dan *user* melakukan permintaan kembali mengenai informasi denah tersebut, kemudian *server* akan merespon kembali permintaan *user* berupa informasi denah berbentuk *text*.

4. Sequence diagram Mengelola data (Menambah dan Menghapus KML/KMZ)

Sequence diagram mengelola data yang akan menjelaskan proses masukan pengguna sampai dapat menambah, *edit*, dan menghapus *file* KML/KMZ. Proses dimulai dengan *user* membuka halaman *web manager*, kemudian *server* akan menampilkan daftar data denah. *User* dapat melakukan penambahan, perubahan, dan penghapusan pada data denah, penambahan data denah dimulai dengan *user* memilih tambah denah, kemudian *v_home* akan menampilkan *form* untuk melakukan penambahan denah. Setelah *user* mengisi *form* kemudian melakukan submit, *class* konten akan melakukan *validasi* pada *form* yang telah dikirim, kemudian dilanjutkan ke *m_konten* untuk proses penambahan pada *database*. Pengubahan data denah dimulai dengan *user* memilih ubah data denah, kemudian *v_home* akan menampilkan *form* untuk melakukan pengubahan data denah. Penghapusan data denah dimulai dengan admin memilih hapus denah, kemudian akan muncul *form* untuk konfirmasi penghapusan denah, *sequence* mengolah data dapat dilihat pada gambar 5.11 dibawah ini.



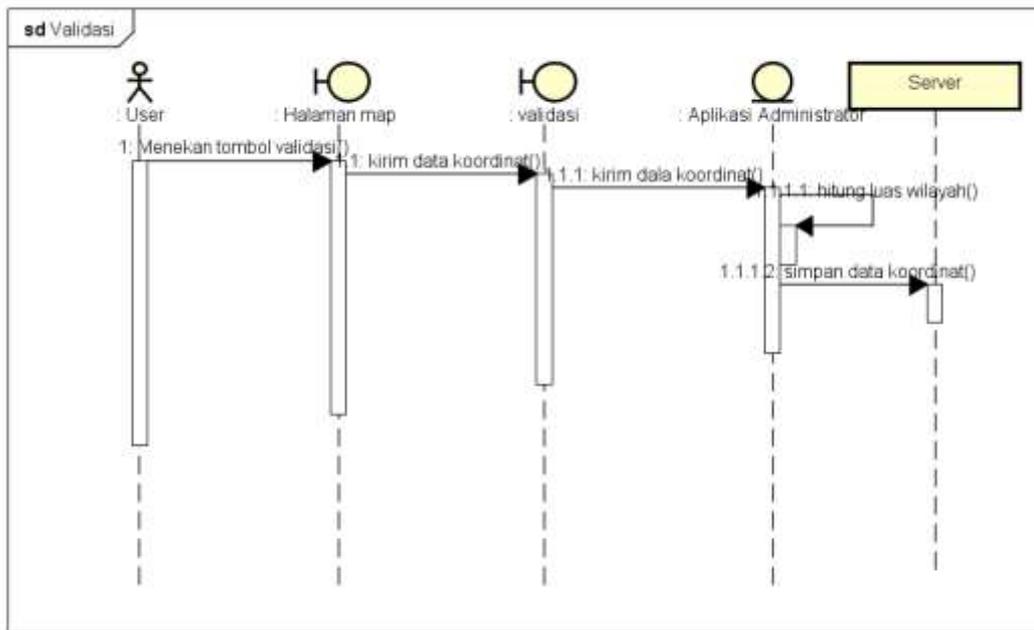


Gambar 5.11 Sequence diagram mengelola data, menambah KML/KMZ, dan Menghapus KML/KMZ

5. Sequence diagram validasi

Gambar 5.12 merupakan sequence diagram validasi yang akan menjelaskan proses validasi data di aplikasi dengan data dilapangan, dimana proses dimulai dengan user menekan tombol validasi dan kemudian sistem akan mengirimkan lokasi koordinat. Aplikasi administrator kemudian akan menghitung luasan wilayah dan data akan disimpan didalam server.



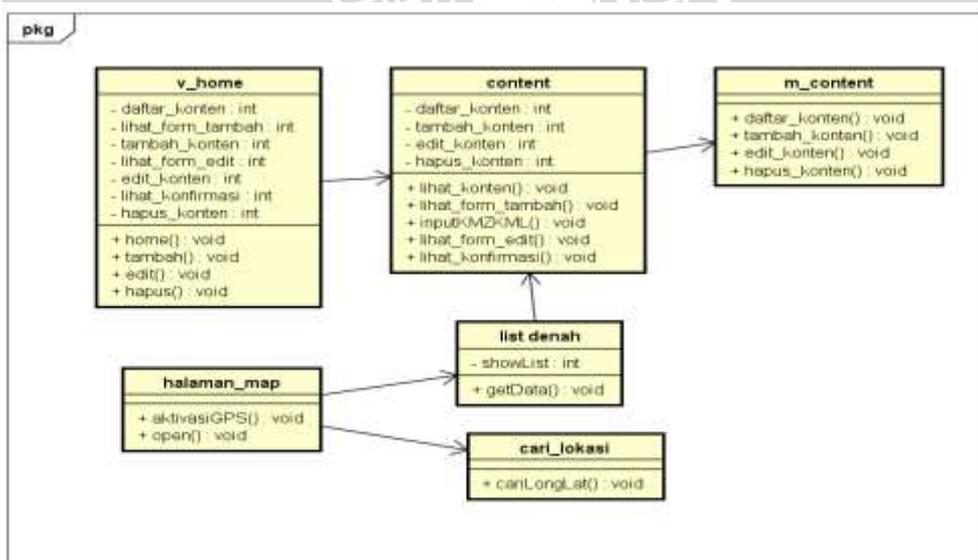


powered by Astah

Gambar 5.12 Sequence diagram validasi

5.1.4 Class Diagram

Perancangan *class* diagram memberikan gambaran mengenai pemodelan elemen-elemen *class* yang membentuk sebuah aplikasi. *Class* dapat diperoleh dari proses menganalisis secara detail terhadap *use case*, *class* diagram merupakan gambaran mengenai sistem beserta aktifitas didalamnya, serta menggambarkan relasi-relasi antar aktifitas yang ada dalam sistem. Gambar 5.11 merupakan perancangan *class* diagram yang akan digunakan dalam pembuatan aplikasi informasi tanah.



powered by Astah

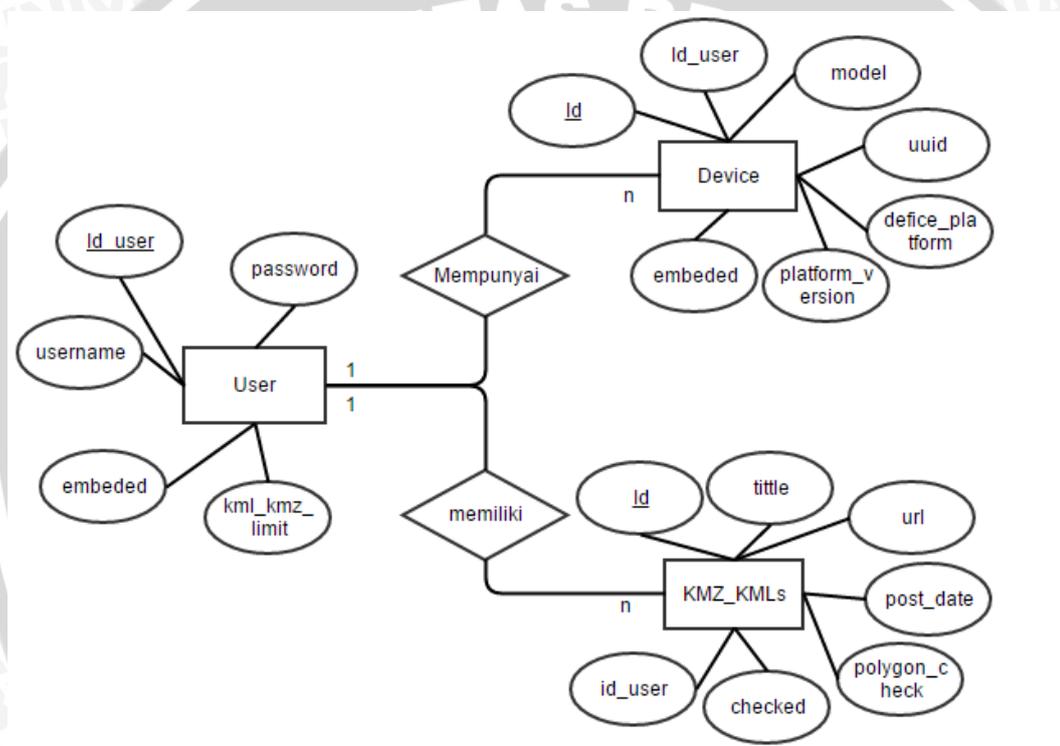
Gambar 5.13 Class diagram aplikasi



Pada Gambar 5.11 diatas menjelaskan tentang *class* yang terdapat pada aplikasi informasi tanah, yang memiliki 6 *class* diagram pada aplikasi ini, yaitu; *class* diagram *v_home* yang berhubungan dengan *content*, dan *class* diagram *content* yang berhubungan dengan *m_content*, serta *class* diagram *m_content*, terdapat juga *class* diagram *halaman_map*, *list_denah*, dan *cari_lokasi* diagram aplikasi.

5.1.5 Basis Data

Basis data memiliki fungsi sebagai tempat untuk penyimpanan data. Pada perancangan ini, sistem basis data yang digunakan untuk mengembangkan aplikasi ini menggunakan model *entity relation* diagram, dan dapat dilihat pada gambar 5.12.



Gambar 5.14 Model ER-Diagram

Gambar 5.14 merupakan model *entity relation* diagram yang merepresentasikan hubungan antara 5 tabel, terdiri dari tabel *devices* untuk menyimpan data *device* secara umum yang mengandung *attribute* *id*, *id_user*, *model device*, *uuid*, *defice_platform* untuk membedakan platform yang digunakan, *platform_version* untuk membedakan versi platform yang digunakan, dan *enabled* untuk melihat status *device*. Tabel *user* untuk menyimpan data *user* yang mengandung *attribute* *id_user*, *username*, *password*, *kml_kmz_limit* untuk menyimpan data kmz kml per *user*, dan *enabled* untuk melihat status *user*. Tabel *kml_kmzs* digunakan untuk menyimpan data *kml_kmzs* yang memiliki *attribute* *id*, *id_user*, *url*, *tittle*, *post_date*, dan *validasi*. Tabel *request_device_licenses* digunakan untuk mendaftarkan *device* baru. Terdapat *attribute* *id*, *id_user*, *uuid*,



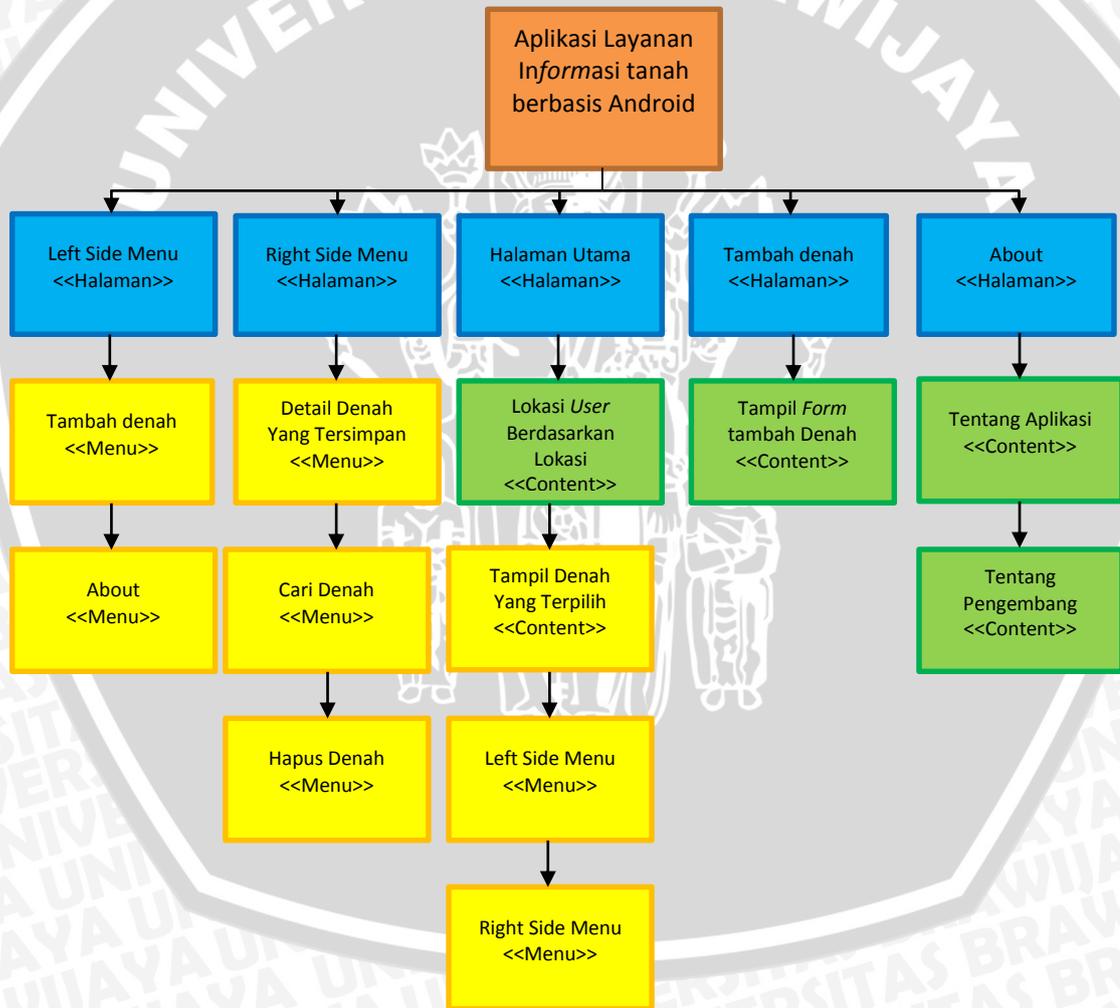
date, dan *has_been_followed_up* untuk status penerimaan tambah device. Dan tabel *notification* terdapat attribute id, id_user, date, status, dan notification.

5.1.6 Navigasi dan Antarmuka

Pada bagian ini akan dijelaskan mengenai perancangan antarmuka aplikasi informasi tanah, yang terdiri dari *screen flow* diagram, dan alur navigasi tiap-tiap halaman, serta tampilan antarmuka dari sistem yang akan dibangun.

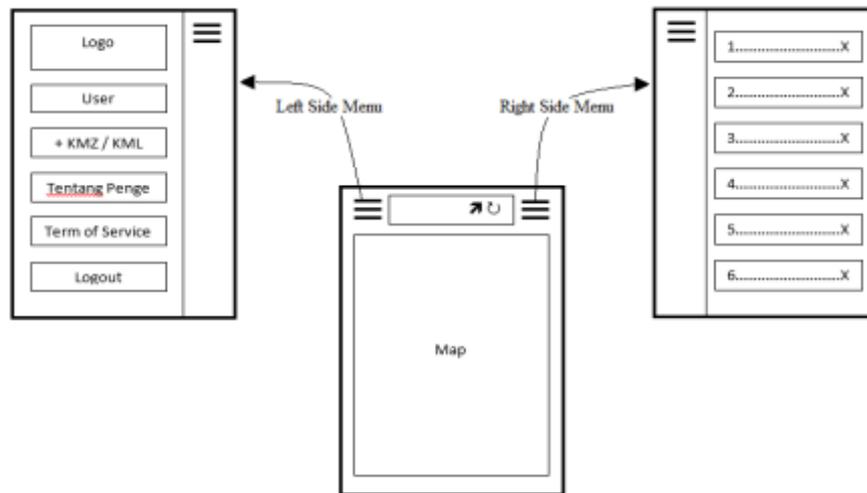
5.1.6.1 Screen Flow

Aplikasi ini akan digunakan oleh pengguna untuk memudahkan dalam pekerjaan pendataan tanah dan mengetahui informasi dari setiap *polygon* tanah. Di dalam aplikasi sendiri terdapat beberapa halaman seperti halaman *login*, halaman utama, left-side menu, dan right-side menu. Berikut ini merupakan peta navigasi aplikasi yang ditunjukkan pada Gambar 5.13 berikut.



Gambar 5.15 Navigation Map Aplikasi

Pada Gambar 5.15 diatas ditunjukkan warna dasar dari navigasi aplikasi, diantaranya biru yang menunjukkan navigasi halaman, warna hijau yang menunjukkan navigasi *content*, dan warna kuning yang menunjukkan navigasi menu.



Gambar 5.16 Flow Screen Aplikasi

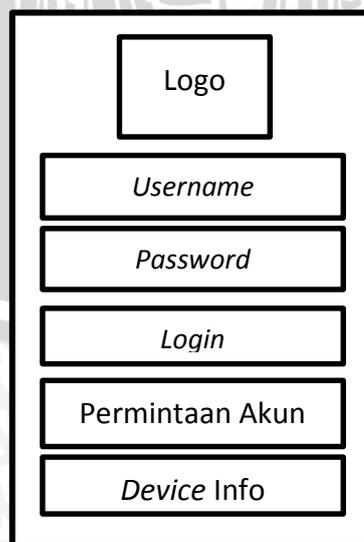
Dari Gambar 5.15 yaitu navigasi *map* dari aplikasi, dapat dipetakan *flow screen* dari tiap-tiap halaman aplikasi yang bisa dilihat pada Gambar 5.16 yang menampilkan aliran tampilan atau *future* yang akan dibangun pada aplikasi ini.

5.1.6.2 Antarmuka

Beberapa gambar yang menunjukkan hasil perancangan antarmuka pengguna aplikasi Informasi Tanah ini ditunjukkan pada Gambar 5.17 sampai dengan Gambar 5.20 berikut ini.

a. Halaman *Login*

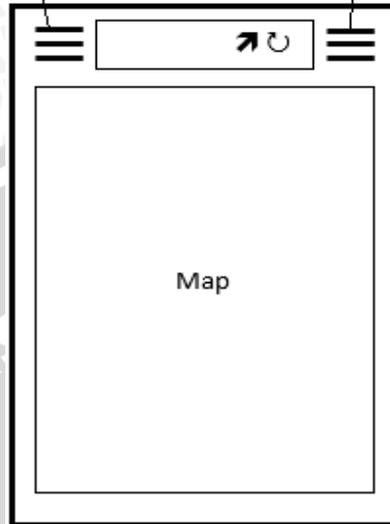
Halaman login merupakan tampilan awal ketika *user* membuka aplikasi informasi tanah, halaman ini mengharuskan *user* telah terdaftar sebagai pengguna aplikasi tanah, jika *user* belum terdaftar sebagai pengguna, maka aplikasi menyediakan *future* permintaan akun, dan dapat dilihat pada Gambar 5.17.



Gambar 5.17 Perancangan antarmuka halaman login

b. Halaman *Dashboard*

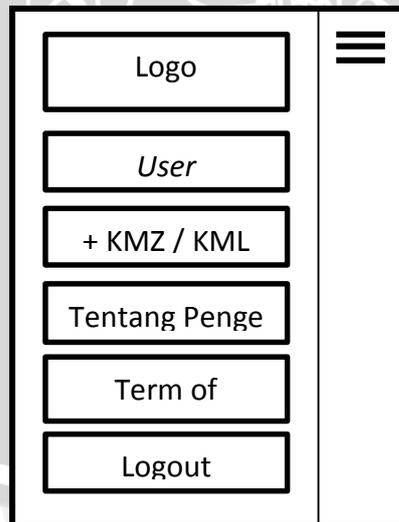
Halaman dashboard yang ada pada Gambar 5.18 merupakan halaman utama yang akan menampilkan informasi lokasi kepemilikan tanah berada dengan menggunakan bantuan aplikasi *google earth*.



Gambar 5.18 Perancangan antarmuka halaman dashboard

c. *Left side Menu*

Left side menu merupakan tampilan menu yang menggunakan *sliding* menu dari kiri ke kanan, selain itu *left side* menu juga dapat diakses dengan memilih *hamburger icon* pada pojok kiri atas, dan dapat dilihat pada Gambar 5.19.

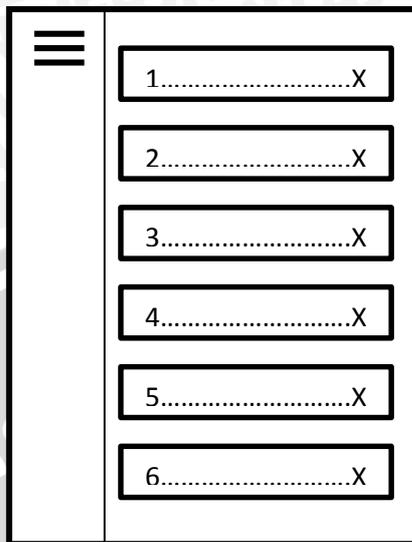


Gambar 5.19 Perancangan antarmuka halaman *left side* menu

d. *Right side Menu*

Right side menu merupakan tampilan menu yang menggunakan *sliding* menu dari kanan ke kiri, selain itu *right side* menu juga dapat diakses

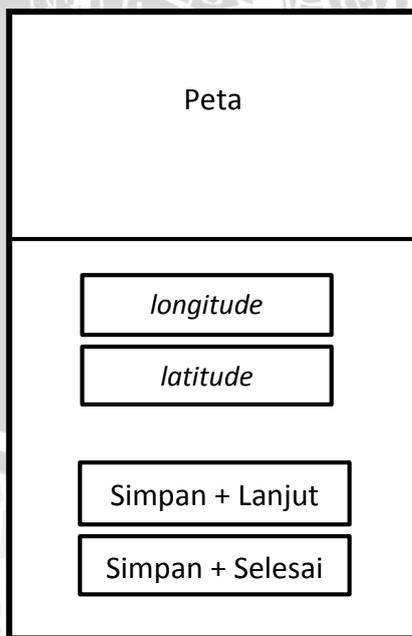
dengan memilih *hamburger icon* pada pojok kanan atas, dan dapat dilihat pada Gambar 5.20.



Gambar 5.20 Perancangan antarmuka halaman *right side* menu

e. Halaman validasi

Halaman validasi merupakan halaman yang digunakan untuk melakukan perhitungan. Dimana akan ditampilkan *longitude* dan *latitudude* sebagai lokasi dari pengguna yang digunakan untuk perhitungan. Terdapat 2 menu, yaitu menu simpan + lanjut apabila pengguna ingin menyimpan posisi dan melanjutkan ke posisi selanjutnya, dan menu simpan + selesai apabila pengguna ingin menyimpan posisi dan melakukan perhitungan, dan dapat dilihat pada gambar 5.21.



Gambar 5.21 Perancangan antarmuka halaman validasi

5.2 Implementasi Aplikasi Perangkat Bergerak

Pada tahapan implementasi perangkat bergerak membahas tentang spesifikasi sistem yang dibutuhkan dalam implementasi aplikasi, berdasarkan rekayasa kebutuhan dan perancangan aplikasi perangkat bergerak yang telah dilakukan. Selain itu juga akan dibahas tentang implementasi basis data, implementasi kode program, dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Rekayasa kebutuhan serta perancangan aplikasi perangkat bergerak informasi tanah akan menjadi acuan dalam implementasi ini, spesifikasi sistem yang diimplementasikan pada aplikasi ini terdiri dari spesifikasi perangkat keras dan perangkat lunak.

Spesifikasi perangkat keras yang digunakan dalam proses implementasi aplikasi perangkat bergerak informasi tanah dapat ditunjukkan pada Tabel 5.1 dan Tabel 5.2.

Tabel 5.1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
<i>System Model</i>	Toshiba Portege R830
<i>Processor</i>	2,80GHz Core i7 2640M
<i>Memory</i>	4GB

Dalam Proses instalasi dan pengujian, perangkat keras bergerak yang digunakan adalah Android dengan spesifikasi dapat ditunjukkan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Keras Android

Nama Komponen	Spesifikasi
<i>System Model</i>	GT-19300
<i>Processor</i>	Samsung Exynos 4 Quad 4412
<i>Memory</i>	1GB
<i>Display</i>	Super AMOLED capacitive touchscreen, 16M colors

Dalam proses pengembangan sistem di perangkat bergerak, aplikasi informasi tanah menggunakan perangkat lunak dengan spesifikasi yang terdapat pada Tabel 5.3 dan Tabel 5.4.

Tabel 5.3 Spesifikasi Perangkat Lunak pada Komputer

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 8 Pro

Bahasa Pemrograman	Java, HTML
IDE (<i>Integrated Development Environment</i>)	Eclipse, NetBeans

Serta menggunakan implementasi dan pengujian dari pengembangan aplikasi ini menggunakan perangkat lunak pada perangkat bergerak dengan spesifikasi pada Tabel 5.4.

Tabel 5.4 Spesifikasi Perangkat Lunak pada Perangkat Bergerak

Nama Komponen	Spesifikasi
<i>Platform</i>	Android 4.3 Jelly Bean

5.2.2 Implementasi Basis Data

Implementasi penyimpanan data pada aplikasi perangkat bergerak informasi tanah menggunakan basis data untuk menyimpan data yang dibutuhkan, implementasi dari basis data pada aplikasi menggunakan pemrograman berbasis MySql, berupa struktur table yang digunakan dalam pembuatan basis data.

Tabel 5.5 Struktur Tabel *Devices*

Column	Type	Null
<i>id (Primary)</i>	int(11)	No
<i>id_user</i>	int(11)	No
Model	varchar(128)	No
Uuid	<i>text</i>	No
<i>device_platform</i>	varchar(128)	No
<i>platform_version</i>	varchar(12)	No
<i>enabled</i>	tinyint(1)	No

Tabel 5.6 Struktur Tabel *Kml_kmzs*

Column	Type	Null	Default
<i>id (Primary)</i>	int(11)	No	
Title	varchar(256)	No	
url	<i>text</i>	No	
post_date	Timestamp	No	CURRENT_TIMESTAMP
Polygon_check	polygon	yes	
Checked	int(1)	No	
Id_user	int(11)	No	

Tabel 5.7 Struktur Tabel *Users*

Column	Type	Null	Default
<i>id (Primary)</i>	int(11)	No	
<i>username</i>	varchar(64)	No	
<i>password</i>	varchar(128)	No	
<i>kml_kmz_limit</i>	int(11)	No	2
<i>enabled</i>	tinyint(1)	No	

5.2.3 Implementasi Kode Aplikasi

Pada penulisan skripsi ini hanya dicantumkan beberapa proses implementasi kode aplikasi, sehingga tidak semua implementasi dari fungsi ditampilkan. Implementasi kode aplikasi yang ditampilkan adalah proses penampilan denah pada *smartphone*, proses penampilan denah pada *web manager*, dan proses penampilan lokasi *user*.

5.2.3.1 Implementasi Prosedur Penampilan Denah pada Smartphone

Implementasi yang digunakan pada penampilan denah pada *device smartphone* akan diitampilkan pada Gambar 5.22.

```

1  $("#add-location-url-button").unbind().click(function() {
2      var e = $("#formAddLocationURL").serialize();
3      $.ajax({
4          type: "POST",
5          url:
6              "http://192.168.1.9/landinformationservices/add_location.ph
7              p",
8          cache: false,
9          data: e,
10         beforeSend: function() {
11             $("#add-location-url-button",
12             $("#formAddLocationURL").attr("disabled", "disabled");
13             $("#add-location-url-
14             button").text("Menyimpan...");
15             $('<div class="loading-overlay">
16             </div>').appendTo(document.body);
17             $.mobile.loading("show", {
18                 text: "Menyimpan KMZ / KML...",
19                 textVisible: true,
20                 theme: "b",
21                 textonly: false,
22                 html: '<p align="center">Menyimpan KMZ /
23                 KML...</p>'
24             });
25             $("#location_name").val("");
26             $("#kml_kmz_url").val("")
27         },
28         success: function(e) {
29             $(".loading-overlay").remove();
30             $.mobile.loading("hide");
31             var t = e.split("|");
32             var n = "";

```

```

27         if (t[0] === "Success") {
28             n = t[1]
29         } else {
30             n = t[1]
31         }
32         $("#popupAddLocationConfirmation-
popup").text(n);
33         $("#popupAddLocationConfirmation").popup("open");
           setTimeout(function() {
34             $("#popupAddLocationConfirmation").popup("close")
           }, 4e3);
           onDeviceReady();
35         $("#add-location-url-
button").text("Simpan");
36         $("#add-location-url-
button").removeAttr("disabled")
37     },
38     error: function(e, t) {
39         $(".loading-overlay").remove();
40         $.mobile.loading("hide");
41         $("#popupAddLocationConfirmation-
popup").text("Terjadi masalah pada koneksi internet.");
42         $("#popupAddLocationConfirmation").popup("open");
           setTimeout(function() {
43             $("#popupAddLocationConfirmation").popup("close")
           }, 4e3);
           $("#add-location-url-
button").text("Simpan");
44         $("#add-location-url-
button").removeAttr("disabled")
45     }
46     });
47     return false
48 }
49 }
50 }
51 }

```

Gambar 5.22 Implementasi Proses Penampilan Denah pada Smartphone

Pada Gambar 5.22, ditampilkan impleemtasi kode program dari proses penampilan denah pada *device smartphone*. Penjelasan kode program diatas merupakan aksi untuk menyimpan data KML/KMZ baru, data KML/KMZ ditambahkan dengan menambahkan URL KML/KMZ yang telah ter-*upload* pada *server* melalui media internet, data dikirim menggunakan metode *Assynchronous process* melalui ajax dengan metode POST.

5.2.3.2 Implementasi Prosedur Penampilan Denah pada Web Manager

Implementasi yang digunakan pada penampilan denah pada *web manager* diitampilkan pada Gambar 5.23.

```

1 $.ajax({
2     url:
  'http://192.168.13.115/landinformationservices/locations.ph
  p',

```

```

3         dataType: 'jsonp',
4         jsonp: 'jsoncallback',
5         data: {id_user: " " .
trim(Session::instance()->get('id')) . " },
6         crossDomain: true,
7         timeout: 20000,
8         success: function(data) {
9             layers_Array.length = 0;
10            var select_location = $("#select-
location");
11            select_location.empty();
12            select_location.append('<option
13            value=""|"></option>');
14
15            $.each(data, function(i, item) {
16                var item_id = item.id;
17                toggleMapStatus[item_id] = 1;
18                layers_Array.push({id: item_id, name:
item.title, url: item.url, zoom: 12});
19
20                select_location.append('<option
value="" + item_id + '|' + item.url + '\>' + item.title +
'</option>');
21                //alert(JSON.stringify(layers_Array));
22                layers = JSON.stringify(layers_Array);
23            });
24            select_location.on('change', function(e) {
25                // var optionSelected =
26                $("#option:selected", this);
27
28                var value = this.value;
29                var splitted_value = value.split("\|");
30
31                if(splitted_value[0] != '' &&
splitted_value[1] != ''){
32                    var idSelected = splitted_value[0];
33                    var urlSelected = splitted_value[1];
34                    //alert(idSelected);
35                    //alert(urlSelected);
36
37                    $("#header-my-location").empty();
38                    $("#header-my-
location").html("<span class='loading-icon'>Mengambil KMZ
/ KML...</span>");
39                    select_location.prop('disabled',
'disabled');
40
41                    toggleGeoXML(idSelected,
urlSelected, 12);
42                }
43            });
44            error: function(request, error) {
45                //alert(request);
46                //alert(error);
47                alert('Terjadi kesalahan mengambil data
KMZ / KML, silakan coba lagi.');
```

```

45         });
46     }

```

Gambar 5.23 Implementasi Proses Penampilan Denah pada Web Manager

Pada Gambar 5.23, ditampilkan implemetansi kode program dari proses penampilan denah pada *web manager*. Penjelasan kode program diatas berfungsi untuk mengambil data KML/KMZ yang telah terdaftar berdasarkan *user* yang login pada *device (smartphone)*. Setelah data berhasil diambil, kemudian data di *populate* pada *List View* menu pada sebelah kanan pada aplikasi.

5.2.3.3 Implementasi Proses Tampilan Lokasi User

Implementasi yang digunakan pada tampilan lokasi *user* ditampilkan pada Gambar 5.24.

```

1  var werror = function(e) {
2      $("#popupFindYourLocationConfirmation-p").text("Gagal
menemukan posisi.");
3      $("#popupFindYourLocationConfirmation").popup("open");
4      setTimeout(function() {
5          $("#popupFindYourLocationConfirmation").popup("close")
6      }, 4e3)
7  };
8  var toggleWatchPosition = function() {
9      if (watchID) {
10         $("#popupFindYourLocationConfirmation-
p").text("Menonaktifkan GPS.");
11         $("#popupFindYourLocationConfirmation").popup("open");
12         setTimeout(function() {
13             $("#popupFindYourLocationConfirmation").popup("close")
14             }, 4e3);
15         clearWatch();

```

Gambar 5.24 Implementasi Proses Tampilan Lokasi User

Pada Gambar 5.24, ditampilkan implemetansi kode program dari proses penampilan denah pada *web manager*, yang menjelaskan proses pengambilan lokasi pengguna aplikasi LIS android.

5.2.3.4 Implementasi proses digitasi

Implementasi yang digunakan pada proses digitasi ditampilkan pada Gambar 5.25.

```

1  google.maps.event.addListener(map, "click", function
2  (event) {
3
4      var lat = event.latLng.lat();
5      var lon = event.latLng.lng();
6
7      placeMarker(event.latLng);
8
9      lpath = polyline.getPath();
10     lpath.push(event.latLng);
11

```

```

12     gpath = polygon.getPath();
13     gpath.push(event.latLng);
14
15     if (start == true) {
16         start = false;
17         $('#koordinat-polygon-
18 checker').val($('#koordinat-polygon-checker').val() + lat +
19 ' ' + lon);
20     } else {
21         $('#koordinat-polygon-
22 checker').val($('#koordinat-polygon-checker').val() + ', ' +
23 lat + ' ' + lon);
24     }
25
26     cleanIncludeExclude($('#koordinat-polygon-
27 checker'));

```

Gambar 5.25 Implementasi Proses Digitasi

Pada Gambar 5.25, ditampilkan implemementasi kode program dari proses digitasi, yang menjelaskan proses pembuatan koordinat untuk proses validasi. Pada proses baris 7 placemarker digunakan untuk pembuatan satu pin lokasi. Dengan variable dari *longitude* dan *latitude*. Proses baris 9-13 merupakan proses gambar *polygon*, gambar didapat dengan penggabungan placemarker yang digunakan. Pada proses baris 15-24 digunakan untuk pemberian koordinat *longitude* dan *latitude* dimana bila placemarker bukan yang pertama, maka ditambahkan koma setelah placemarker sebelumnya. Pada baris proses 26 digunakan untuk menghilangkan duplikasi koordinat.

5.2.3.5 Implementasi proses hitung luas

Implementasi yang digunakan pada proses hitung luas ditampilkan pada Gambar 5.26.

```

1     var save_check = function () {
2         //alert('checkedCounter');
3         if ($('#id_kmz_kml').val() != '' && $('#koordinat-
4 polygon-checker').val() != '' && checkedCounter == 1) {
5             var x =
6 google.maps.geometry.spherical.computeArea(polygon.getPath(
7 ));
8             $('#luas').val('Luas: ' + Number(x).toFixed(2)) + '
9 m2');
10            $('#popupKonfirmSimpanVerifikasi').popup("open");
11            //setTimeout(function () {
12                //
13            $('#popupKonfirmSimpanVerifikasi').popup("close")
14            //}, 4e3);
15        } else {
16            alert('Silakan pilih (check) 1 KML/KMZ.');
```

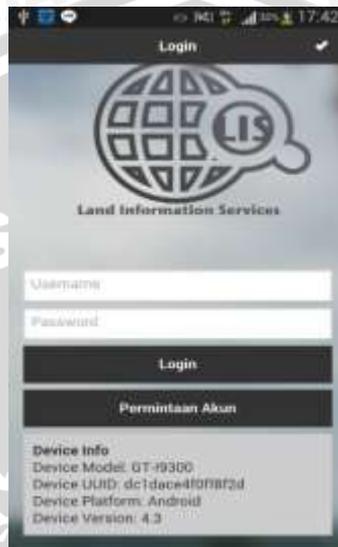
Gambar 5.26 Implementasi Proses Hitung Luas

Pada Gambar 5.26, ditampilkan implemementasi kode program dari proses hitung luas, yang menjelaskan proses penghitungan luas. Dimana proses perhitungan luas ditunjukkan pasda baris 8-9.

5.2.4 Implementasi Antarmuka

Aplikasi sistem yang dibangun akan menjadi prioritas utama dalam hal pengembangan dari sebuah sistem, dimana *user* dengan mudah menggunakan aplikasi tersebut, sehingga tampilan antarmuka tidak membuat pengguna merasa bosan dan memudahkan dalam menggunakan aplikasi.

5.2.4.1 Halaman Utama



Gambar 5.27 Halaman login

Pada Gambar 5.27 merupakan halaman login yang merupakan halaman pertama yang akan muncul ketika *user* mengakses layanan aplikasi, dengan melakukan *login user* menggunakan *username* dan *password*. Tampilan halaman login dibuat dengan menggunakan kode XML.

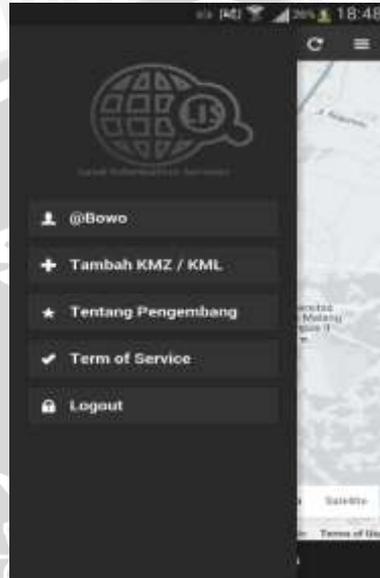
5.2.4.2 Halaman Informasi Tanah



Gambar 5.28 Halaman Informasi Tanah

Setelah melakukan *login user* aplikasi akan menampilkan halaman informasi tanah yang diambil dari layanan *google earth*, dimana tampilan lokasi tersebut diambil dari *library google maps*. Tampilan yang munculkan oleh halaman informasi berdasarkan posisi petugas lapangan tersebut dan terlihat pada Gambar 5.28.

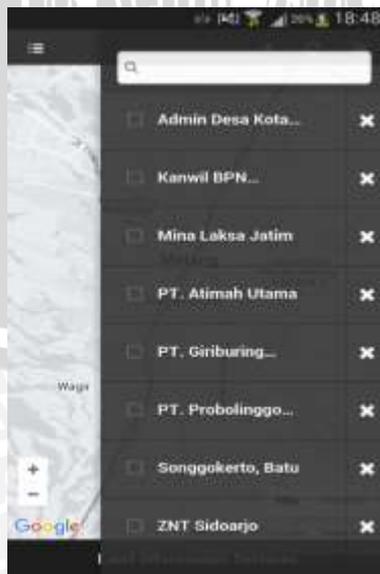
5.2.4.3 Left side Menu



Gambar 5.29 Left side Menu

Pada Gambar 5.29 merupakan halaman *navigasi* apabila *user* melakukan pergeseran (*slide*) dari kiri ke kanan, yang terdiri dari menu *username*, tambah *file KML/KMZ*, tentang pengembang, *term of service*, dan *logout*. Tampilan halaman *left side menu* dibuat dengan menggunakan kode XML.

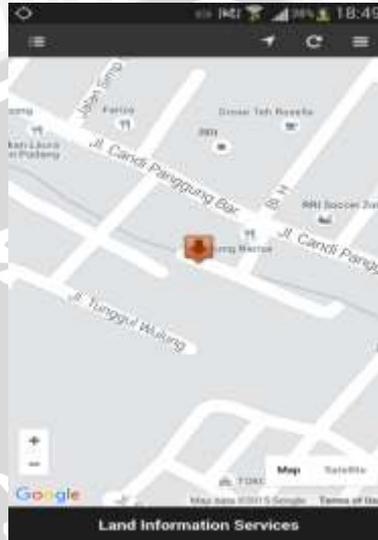
5.2.4.4 Right side Menu



Gambar 5.30 Right side Menu

Pada Gambar 5.30 merupakan halaman *right side menu* ini merupakan halaman apabila *user* melakukan pergeseran (*slide*) dari kanan ke kiri, memuat informasi data kepemilikan tanah yang di *upload* menggunakan *web manager*, dimana data tersebut bersumber dari kantor BPN Kanwil Jawa Timur. Tampilan halaman *right side menu* dibuat dengan menggunakan kode XML.

5.2.4.5 Tampilan Posisi User



Gambar 5.31 Tampilan Posisi User

Pada Gambar 5.31 menampilkan halaman posisi *user* untuk mengidentifikasi dimana *user* berada dengan menggunakan aplikasi *google earth*, memudahkan petugas lapangan BPN dalam melakukan survei lapangan. Dimana tampilan lokasi *user* tersebut diambil dari *library google maps*.

5.2.4.6 Tampilan denah



Gambar 5.32 Tampilan denah

Pada Gambar 5.32 menampilkan tampilan denah yang didapat setelah user memilih salah satu kepemilikan tanah yang berada pada *right side* menu, maka aplikasi akan memunculkan halaman tampilan denah berbentuk *polygon*, informasi denah tanah dapat dilihat pada daerah yang memiliki warna berbeda dari daerah sekitarnya.

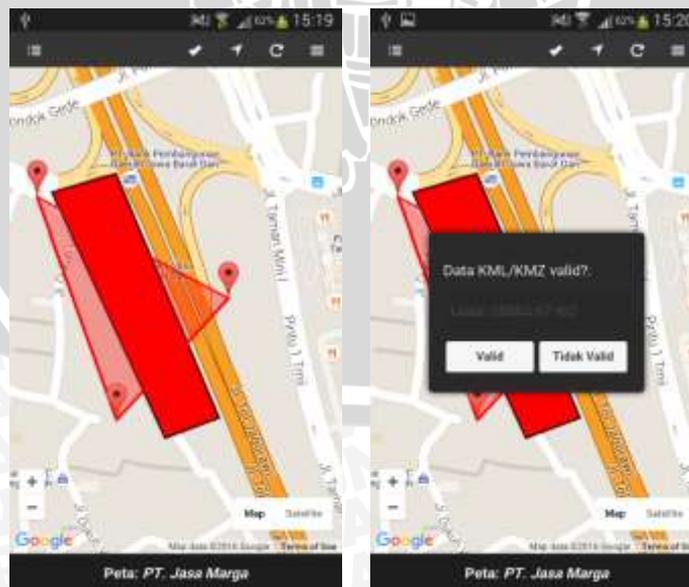
5.2.4.7 Halaman informasi data tanah



Gambar 5.33 Informasi data tanah

Pada gambar 5.33 akan muncul informasi data tanah, saat gambar *polygon* diklik oleh user, informasi data tanah memuat nama perusahaan, alamat perusahaan, email, serta posisi berdasarkan *longitude* dan *latitude*.

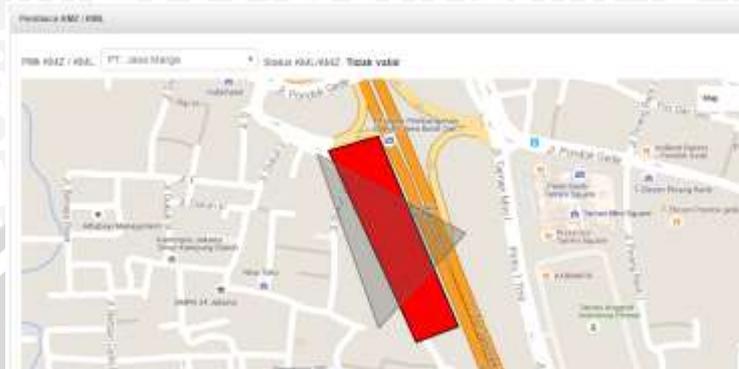
5.2.4.8 Halaman informasi validasi



Gambar 5.34 Halaman informasi validasi

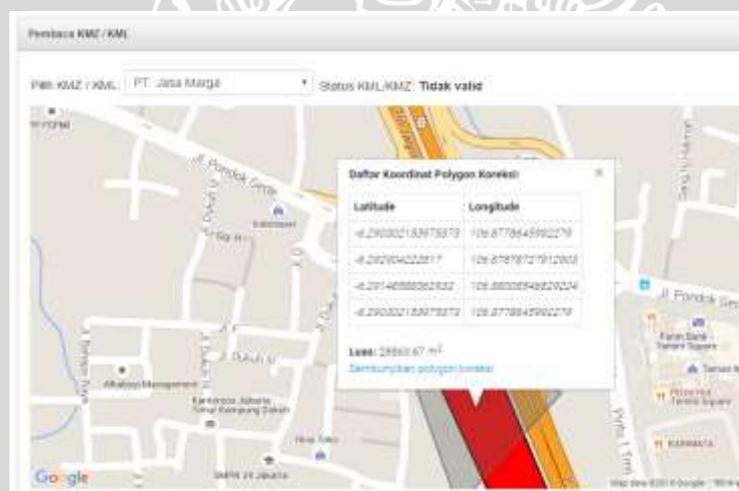
Pada Gambar 5.34 merupakan proses validasi denah yang berfungsi untuk membandingkan data lapangan dengan data digital, akan disinkronisasikan dengan melakukan *validasi* data apakah data tersebut valid atau tidak, seperti yang ditampilkan.

5.2.4.9 Halaman pada web manager setelah validasi



Gambar 5.35 Halaman informasi validasi

Pada Gambar 5.35 merupakan hasil dari validasi yang dilakukan di lapangan oleh petugas BPN, dan telah dilakukan proses pengukuran. Akan ditampilkan *polygon* baru, sebagai koreksi terhadap data digital sebelumnya.

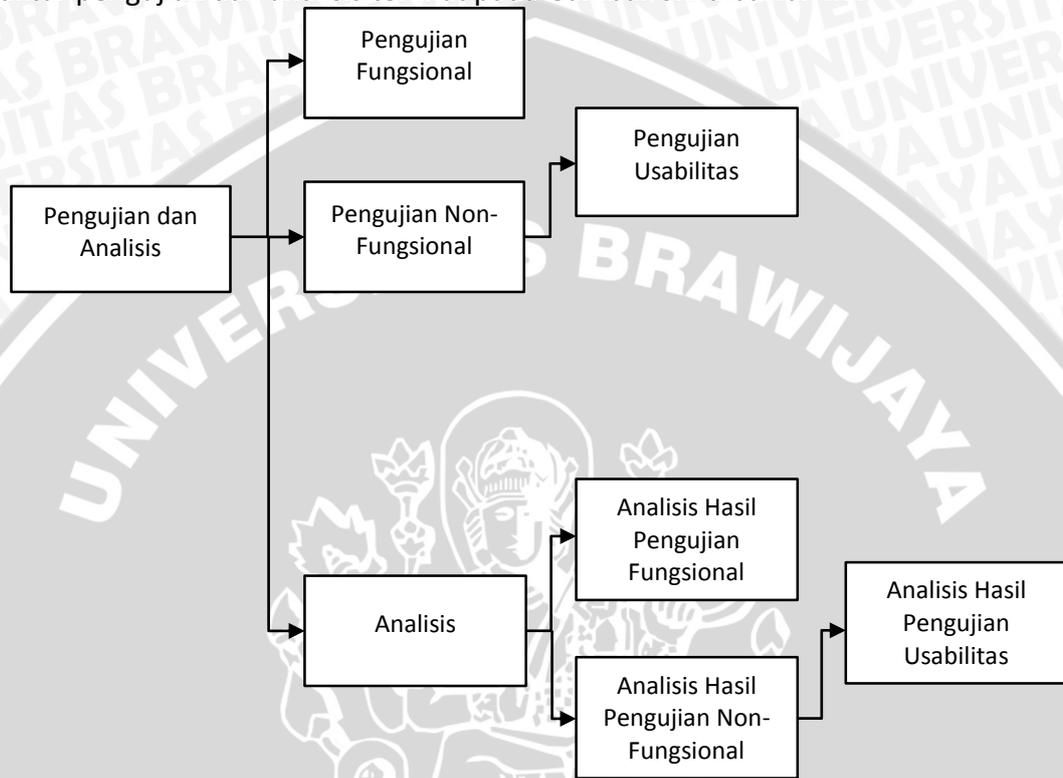


Gambar 5.36 Halaman informasi validasi

Pada Gambar 5.36 merupakan proses validasi yang telah dilakukan oleh petugas BPN di lapangan, menampilkan koordinat *longitude* dan *latitude* setiap titik. Aplikasi pada *web manager* juga akan menampilkan luas denah yang telah dikoreksi.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang proses pengujian dan analisis hasil implementasi dari bab-bab sebelumnya, pengujian dan analisis dilaksanakan dengan tujuan agar aplikasi yang dibangun memiliki kualitas yang lebih baik, struktur pengujian dan analisis terlihat pada Gambar 6.1 dibawah ini.



Gambar 6.1 Pohon Diagram Pengujian dan Analisis

Proses pengujian melalui beberapa jenis pengujian yaitu pengujian fungsional dan pengujian non-fungsional. Pada pengujian fungsional menggunakan teknik pengujian *black-box*. Pada pengujian non-fungsional akan digunakan metode pengujian *usability* dengan teknik kuesioner.

6.1 Pengujian Fungsional

Pengujian fungsional dilakukan untuk mengetahui apakah aplikasi yang dibangun sudah benar dan sesuai dengan yang dibutuhkan. Daftar kebutuhan fungsional telah dirumuskan sebelumnya pada bab rekayasa kebutuhan. Pengujian fungsional menggunakan metode pengujian *black box*, karena lebih ditekankan pada kesesuaian kinerja aplikasi dengan daftar kebutuhan.

6.1.1 Kasus Uji Fungsional

Pada Skripsi ini dilakukan pengujian terhadap aplikasi informasi tanah, yang ditujukan untuk mengetahui kesesuaian antara kebutuhan fungsional dengan fitur

dan kinerja yang terdapat pada aplikasi, pada setiap kebutuhan fungsional akan dilakukan pengujian dengan kasus uji yang berbeda-beda. Adapun kasus uji ditampilkan pada Tabel 6.1 sampai Tabel 6.3.

Tabel 6.1 Kasus Uji Fungsional Melihat Lokasi User

Nomor Kasus Uji	FUN_01
Nama Kasus Uji	Melihat Lokasi <i>user</i>
Objek Uji	Melihat Lokasi <i>user</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fitur untuk melihat lokasi <i>user</i> .
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Membuka Aplikasi 2. Melakukan Login 3. Masuk ke halaman utama 4. Mengaktifkan GPS
Hasil yang Diharapkan	Pengguna dapat melihat pin sebagai <i>symbol</i> posisi lokasi <i>user</i> .

Tabel 6.2 Kasus Uji Fungsional Melihat Lokasi Denah

Nomor Kasus Uji	FUN_02
Nama Kasus Uji	Melihat Lokasi denah
Objek Uji	Melihat Lokasi denah
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fitur untuk melihat lokasi denah.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Membuka Aplikasi 2. Melakukan Login 3. Masuk ke halaman utama 4. Klik tombol menu di kanan aplikasi 5. Memilih salah satu denah yang terdapat dalam <i>list</i> denah
Hasil yang Diharapkan	Pengguna dapat melihat <i>polygon</i> sebagai luasan denah.

Tabel 6.3 Kasus Uji Fungsional Lihat Informasi Denah

Nomor Kasus Uji	FUN_03
Nama Kasus Uji	Lihat Informasi Denah
Objek Uji	Lihat Informasi Denah
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fitur untuk melihat lokasi denah.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Membuka Aplikasi 2. Melakukan Login 3. Masuk ke halaman utama 4. Klik tombol menu di kanan aplikasi 5. Memilih salah satu denah yang terdapat dalam <i>list</i> denah 6. Klik denah
Hasil yang Diharapkan	Pengguna dapat melihat tulisan berupa informasi denah yang dipilih.

Tabel 6.4 Kasus Uji Fungsional Validasi

Nomor Kasus Uji	FUN_04
Nama Kasus Uji	Validasi
Objek Uji	Validasi
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fitur untuk validasi
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Membuka Aplikasi 2. Melakukan Login 3. Masuk ke halaman utama 4. Klik tombol menu di kanan aplikasi 5. Memilih salah satu denah yang terdapat dalam <i>list</i> denah 6. Melakukan digitasi 7. Menentukan valid atau tidak
Hasil yang Diharapkan	Pengguna dapat melihat koordinat, luas, dan informasi validitas.

6.1.2 Hasil Pengujian Fungsional

Berdasarkan pada kasus uji coba yang ditentukan, maka didapatkan hasil pengujian fungsional seperti yang ditampilkan pada Tabel 6.4.

Tabel 6.5 Hasil Pengujian Fungsional

Nomor Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status
FUN_01	Pengguna dapat melihat pin sebagai <i>symbol</i> posisi lokasi <i>user</i> .	Pengguna dapat melihat pin sebagai <i>symbol</i> posisi lokasi <i>user</i> .	Valid
FUN_02	Pengguna dapat melihat <i>polygon</i> sebagai luasan denah.	Pengguna dapat melihat <i>polygon</i> sebagai luasan denah.	Valid
FUN_03	Pengguna dapat melihat tulisan berupa informasi denah yang dipilih.	Pengguna dapat melihat tulisan berupa informasi denah yang dipilih.	Valid
FUN_04	Pengguna dapat melihat koordinat, luas, dan informasi validitas.	Pengguna dapat melihat koordinat, luas, dan informasi validitas.	Valid

6.2 Pengujian Non-Fungsional

Pada pengujian non-fungsional dilakukan pengujian seperti yang telah dijabarkan pada perancangan sebelumnya yaitu *usability*, dan akses data. Pengujian non-fungsional dilakukan bertujuan untuk memastikan aspek non-fungsional dapat berjalan dengan baik.

6.2.1 Pengujian *Usability*

Pengujian *usability* dilakukan dengan menggunakan teknik kuisisioner dan metode skala *likert*, yang digunakan untuk mendapatkan umpan balik dari para responden, pengujian dilakukan dalam bentuk penyebaran kuisisioner memuat 16 pertanyaan diajukan kepada responden petugas lapang BPN Jawa Timur sebanyak 5 orang, dimana ke-5 responden tersebut memiliki latar belakang pengalaman kerja dilapangan minimal 1 tahun. Responden diminta untuk melakukan penilaian dengan memberi tanda silang pada salah satu angkat penilaiin skala *liket* pada tiap pertanyaan. Skala *likert* terdiri dari 5 angka penilaiin, yaitu (1) sangat tidak setuju, (2) tidak setuju, (3) netral, (4) setuju, dan (5) sangat setuju. Pada Tabel 6.5 menunjukkan isi kuisisioner yang ditanyakan.

Tabel 6.6 Isi Kuisisioner USE

Aspek	No	Pertanyaan
<i>Learnability</i>	1.	Apakah informasi teks yang ada pada aplikasi jelas dan mudah dipahami?

	2.	Apakah nama menu yang ada sudah ditulis dengan logis dan mudah dipahami?
	3.	Apakah informasi pada setiap halaman telah memungkinkan pengguna untuk mengambil keputusan?
	4.	Apakah fitur-fitur yang disediakan dapat dengan mudah anda akses?
Efficiency	5.	Apakah pemilihan menu dan <i>list</i> denah dapat anda lakukan dengan cepat?
	6.	Apakah anda dapat dengan mudah melihat informasi denah?
	7.	Apakah anda dapat dengan mudah mengetahui posisi pengguna dalam map aplikasi?
	8.	Apakah aplikasi mudah digunakan?
Memorability	9.	Apakah penamaan nama menu dan judul halaman sesuai dengan ekspektasi yang anda inginkan?
	10.	Apakah struktur menu dan halaman dapat dengan mudah anda ingat?
	11.	Apakah ikon-ikon yang terdapat pada aplikasi dapat mudah anda kenali?
Errors	12.	Apakah pesan kesalahan menginformasikan penyebab kesalahan ?
	13.	Apakah sistem mencegah anda ketika melakukan sebuah kesalahan ?
Satisfaction	14.	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan ?
	15.	Apakah pengalaman dalam penggunaan aplikasi sudah memuaskan anda ?
	16.	Apakah aplikasi membantu dalam kerja yang lebih efektif dan lebih mudah?

6.2.1.1 Hasil Pengujian Usability

Dari hasil pengujian yang telah dikumpulkan, didapatkan nilai hasil pengujian *responden* yang diakumulasikan dalam bentuk poin pada setiap aspek yang diuji, hasil pengujian terlihat pada Tabel 6.6, dimana terdapat 5 poin penilaian, yaitu poin 1 yang berarti sangat tidak setuju, poin 2 yang berarti tidak setuju, poin 3 yang berarti *netral*, poin 4 yang berarti setuju, dan poin 5 yang berarti sangat setuju. Hasil pengujian ini akan dihitung kembali dengan menggunakan skala *likert* pada bagian analisis hasil pengujian *usability*.

Tabel 6.7 Hasil Pengujian *Usability*

No.	Pertanyaan	STS	TS	N	S	SS
1.	Apakah informasi teks yang ada pada aplikasi jelas dan mudah dipahami?	0	0	0	5	0
2.	Apakah nama menu yang ada sudah ditulis dengan logis dan mudah dipahami?	0	0	0	3	2
3.	Apakah informasi pada setiap halaman telah memungkinkan pengguna untuk mengambil keputusan?	0	0	1	3	1
4.	Apakah fitur-fitur yang disediakan dapat dengan mudah anda akses?	0	0	0	4	1
5.	Apakah pemilihan menu dan <i>list</i> denah dapat anda lakukan dengan cepat?	0	0	1	4	0
6.	Apakah anda dapat dengan mudah melihat informasi denah?	0	0	0	2	3
7.	Apakah anda dapat dengan mudah mengetahui posisi pengguna dalam map aplikasi?	0	0	0	4	1
8.	Apakah aplikasi mudah digunakan?	0	0	0	1	4
9.	Apakah penamaan nama menu dan judul halaman sesuai dengan ekspetasi yang anda inginkan?	0	0	0	5	0
10.	Apakah struktur menu dan halaman dapat dengan mudah anda ingat?	0	0	0	3	2
11.	Apakah ikon-ikon yang terdapat pada aplikasi dapat mudah anda kenali?	0	0	0	4	1
12.	Apakah pesan kesalahan menginformasikan penyebab kesalahan ?	0	0	0	4	1
13.	Apakah sistem mencegah anda ketika melakukan sebuah kesalahan ?	0	0	2	2	1

14.	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan ?	0	0	1	2	2
15.	Apakah pengalaman dalam penggunaan aplikasi sudah memuaskan anda ?	0	0	0	5	0
16.	Apakah aplikasi membantu dalam kerja yang lebih efektif dan lebih mudah?	0	0	0	1	4

6.3 Analisis

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian aplikasi ini, proses analisis yang dilakukan adalah analisis pengujian fungsional, dan analisis pengujian non-fungsional antara lain adalah pengujian *usability*.

6.3.1 Analisis Hasil Pengujian Fungsional

Dalam melakukan analisis pengujian fungsional harus melihat kesesuaian antara hasil kinerja dari fitur-fitur yang ada di dalam aplikasi tersebut dengan daftar kebutuhan fungsional. Berdasarkan hasil pengujian fungsional dapat disimpulkan bahwa implementasi dan fungsional aplikasi perangkat bergerak informasi tanah telah memenuhi kebutuhan yang telah dijabarkan pada tahap rekayasa kebutuhan.

6.3.2 Analisis hasil Pengujian Non-Fungsional

Pada hasil pengujian non-fungsional memaparkan analisis dari hasil pengujian *usability* yang telah dilakukan sebelumnya.

6.3.2.1 Analisis Hasil Pengujian Usability

Proses analisis terhadap hasil pengujian *usability* dilakukan dengan menggunakan skala *likert*, interpretasi skor *likert* atau presentase dari setiap skor *likert* ditunjukkan pada Tabel 6.7, dimana hasil perhitungan indeks persentase dari setiap pertanyaan dapat dilihat pada Tabel 6.8 dan hasil pengujian *usability* ditunjukkan pada Tabel 6.9. Untuk melakukan perhitungan total skor dan perhitungan indeks persentase, dilakukan dengan menggunakan rumus perhitungan yang tertera pada Tabel 6.10.

Tabel 6.8 Interpretasi Skor Likert

Skor Likert	Interpretasi Skor Dengan Interval = 20	Pilihan
1	0% – 19,99%	Sangat Tidak Memuaskan
2	20% – 39,99%	Tidak Memuaskan
3	40% – 59,99%	Biasa
4	60% – 79,99%	Memuaskan
5	80% – 100%	Sangat Memuaskan

Tabel 6.9 Indeks Presentasi Hasil Kuesioner

No.	Pertanyaan	STS	TS	N	S	SS	Total Skor	Indeks (%)
Learnability								
1.	Apakah informasi teks yang ada pada aplikasi jelas dan mudah dipahami?	0	0	0	5	0	20	80%
2.	Apakah nama menu yang ada sudah ditulis dengan logis dan mudah dipahami?	0	0	0	3	2	22	88%
3.	Apakah informasi pada setiap halaman telah memungkinkan pengguna untuk mengambil keputusan?	0	0	1	3	1	20	80%
4.	Apakah fitur-fitur yang disediakan dapat dengan mudah anda akses?	0	0	0	4	1	21	84%
Efficiency								
5.	Apakah pemilihan menu dan <i>list</i> denah dapat anda lakukan dengan cepat?	0	0	1	4	0	19	76%
6.	Apakah anda dapat dengan mudah melihat informasi denah?	0	0	0	2	3	23	92%
7.	Apakah anda dapat dengan mudah mengetahui posisi pengguna dalam map aplikasi?	0	0	0	4	1	21	84%
8.	Apakah aplikasi mudah digunakan?	0	0	0	1	4	24	96%
Memorability								
9.	Apakah penamaan nama menu dan judul halaman sesuai dengan ekspektasi yang anda inginkan?	0	0	0	5	0	20	80%
10.	Apakah struktur menu dan halaman dapat dengan mudah anda ingat?	0	0	0	3	2	22	88%

11.	Apakah ikon-ikon yang terdapat pada aplikasi dapat mudah anda kenali?	0	0	0	4	1	21	84%
Errors								
12.	Apakah pesan kesalahan menginformasikan penyebab kesalahan ?	0	0	0	4	1	21	84%
13.	Apakah sistem mencegah anda ketika melakukan sebuah kesalahan ?	0	0	2	2	1	19	76%
Satisfaction								
14.	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan ?	0	0	1	2	2	21	84%
15.	Apakah pengalaman dalam penggunaan aplikasi sudah memuaskan anda ?	0	0	0	5	0	20	80%
16.	Apakah aplikasi membantu dalam kerja yang lebih efektif dan lebih mudah?	0	0	0	1	4	24	96%

Tabel 6.10 Status Pengujian Usability Aplikasi Informasi Tanah

Aspek Penilaian	Rata-Rata Presentase (%)	Status
<i>Learnability</i>	83%	Sangat memuaskan
<i>Efficiency</i>	87%	Sangat memuaskan
<i>Memorability</i>	84%	Sangat memuaskan
<i>Errors</i>	80%	Sangat memuaskan
<i>Satisfaction</i>	86.66%	Sangat memuaskan
Rata-rata	84.132%	Sangat memuaskan

Tabel 6.11 Rumus Perhitungan

Nama Rumus	Rumus Perhitungan
Total Skor	Total Skor = $S_{S1S} \times 1 + S_{TS} \times 2 + S_N \times 3 + S_S \times 4 + S_{SS} \times 5$
Index (%)	Index (%) = $(\text{Total Skor} / Y) \times 100$
Cari nilai Y	$Y = \text{Skor Likert Tertinggi} \times \text{Jumlah Responder}$

Dari hasil pengujian *usability* yang telah dilakukan pada Tabel 6.9 dihasilkan presentase rata-rata sebesar 84.132% dengan status “sangat memuaskan”. Dengan demikian dapat diambil kesimpulan bahwa aplikasi informasi tanah sangat memuaskan bagi pengguna.

BAB 7 PENUTUP

7.1 Kesimpulan

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

1. Hasil perancangan dan implementasi dari aplikasi layanan informasi tanah pada *smartphone* android ini sangat berguna bagi petugas lapangan, dapat merupakan sebagai alat pendukung dalam pengecekan dan pendataan tanah di lapangan.
2. Proses inputan data yang terdapat pada aplikasi berupa data sertifikat tanah, peta tanah, dan gambar ukur menggunakan teknologi *google earth*, dengan menghasilkan *file* berformat KML/KMZ yang dapat di *upload* dengan menggunakan layanan *web manager*.
3. Hasil *validasi* data pada aplikasi informasi tanah ini didapat berdasarkan perbandingan data pengukuran di lapangan dan data digital, data tersebut akan disajikan dalam bentuk informasi *validasi*.
4. Berdasarkan hasil pengujian fungsional dengan metode pengujian *blackbox* menunjukkan bahwa semua fitur atau kebutuhan fungsional telah sesuai dengan yang dibutuhkan. Selain itu, tingkat *usability* dari aplikasi berdasarkan aspek pengujian *usability* yang meliputi aspek *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction* menunjukkan rata-rata sebesar 84.132% dengan status sangat memuaskan. Berdasarkan beberapa hasil pengujian tersebut aplikasi dapat digunakan dengan semestinya.

7.2 Saran

Berikut ini adalah saran-saran yang dapat diberikan untuk pengembangan aplikasi perangkat bergerak informasi tanah agar lebih dapat berkembang, yaitu:

1. Jika memungkinkan, sebaiknya aplikasi ini tidak hanya dapat digunakan pada OS (*Operating System*) Android, akan tetapi dapat di implementasikan pada OS (*Operating System*) perangkat bergerak lainnya, seperti Windows Phone, dan iOS.
2. Pada penelitian selanjutnya disarankan aplikasi ini dapat dikembangkan tidak hanya berupa data simulasi, akan tetapi memuat data yang sebenarnya sehingga memudahkan petugas lapangan BPN dalam melakukan tugasnya.
3. Pada penelitian selanjutnya, dapat dikembangkan dengan ditambahkan fitur validasi *automatic*. Dimana validasi dilakukan langsung ketika sistem telah mendapatkan perbandingan data lapangan dan data digitasi.

DAFTAR PUSTAKA

- Json. (2015). *Introducing JSON*. Retrieved Mei 29, 2015, from <http://www.json.org>
- International, R. (2013). *The JSON Data Interchange Format*. Retrieved Mei 30, 2015, from <http://www.ecma-international.org/publications/files/ECMA>
- Phonegap. (2015). *Introduction to Apache Cordova*. Retrieved Juni 1, 2015, from http://www.slideshare.net/ejlp12/intro-to-apache-cordova?from_action=save&from=fblanding
- Wargo, M.J. (2012). *Phonegap Essenrials: Building Cross-Platform Monile Apps*
- Flanagan, D. (2011). *JQuery Pocket Reference*
- Efifa, S.N. (2012). Penggunaan Framework Kohana dalam Pembuatan Aplikasi Berbasis Web Sederhana
- BPN. (2015) BPN. Retrieved Agustus 13, 2015, from <http://www.bpn.go.id/Tentang-Kami/Sekilas>
- Wicaksono, B. (2013). Sistem Operasi Android Versi 4.0 (ICS: Ice Cream Sandwich) dan Versi 4.1 (Jelly Bean)
- Wulandari, R. (2012). Bahasa HTML
- Ginting, G.N. (2013). Perancangan Aplikasi Pembelajaran Cascading Style Sheets dengan Metode Computer Based Instruction
- Nugroho, N.B (2008). Desain *Web* Menggunakan HTML dan JavaScript.
- Tempake, H.S. (2007). Penggunaan AJAX pada Pengembangan Aplikasi *Web*
- Imaniar, J., & Arifin, & Kalilullah, A.S. (2011). Aplikasi *Location Based Service* untuk Sistem Informasi Publikasi Acara pada *Platform* Android.
- Indrayani, M.A (2013). Format dan Konversi Data GIS.

KUISIONER

Lampiran 1 Lembar Kuesioner Pengujian *Usability*

Kuesioner Penggunaan Aplikasi Mobile Informasi Tanah

Identitas Responden:

Nama :

No Identitas :

(KTP/SIM/NIP)

Email :

No. HP :

Petunjuk Pengisian:

Beri tanda centang (✓) pada masing-masing pertanyaan dengan jawaban yang menurut anda paling sesuai!

1 = Sangat Tidak Setuju

3 = Netral

5 = Sangat Setuju

2 = Tidak Setuju

4 = Setuju

Kuesioner Aplikasi

No.	Pertanyaan	1	2	3	4	5
1.	Apakah informasi teks yang ada pada aplikasi jelas dan mudah dipahami?					
2.	Apakah nama menu yang ada sudah ditulis dengan logis dan mudah dipahami?					
3.	Apakah informasi pada setiap halaman telah memungkinkan pengguna untuk mengambil keputusan?					
4.	Apakah fitur-fitur yang disediakan dapat dengan mudah anda akses?					
5.	Apakah pemilihan menu dan <i>list</i> denah dapat anda lakukan dengan cepat?					

6.	Apakah anda dapat dengan mudah melihat informasi denah?				
7.	Apakah anda dapat dengan mudah mengetahui posisi pengguna dalam map aplikasi?				
8.	Apakah aplikasi mudah digunakan?				
9.	Apakah penamaan nama menu dan judul halaman sesuai dengan ekspektasi yang anda inginkan?				
10.	Apakah struktur menu dan halaman dapat dengan mudah anda ingat?				
11.	Apakah ikon-ikon yang terdapat pada aplikasi dapat mudah anda kenali?				
12.	Apakah pesan kesalahan menginformasikan penyebab kesalahan ?				
13.	Apakah sistem mencegah anda ketika melakukan sebuah kesalahan ?				
14.	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan ?				
15.	Apakah pengalaman dalam penggunaan aplikasi sudah memuaskan anda ?				
16.	Apakah aplikasi membantu dalam kerja yang lebih efektif dan lebih mudah?				

Surabaya,

Kritik:

.....

Saran:

.....

(.....)

