

**PENGEMBANGAN APLIKASI MOBILE GEOTAGGING
PELAPORAN KONDISI PERTUMBUHAN DAN PRODUKSI
TANAMAN PERKEBUNAN BERBASIS ANDROID
STUDI KASUS: PTPN XII KEBUN PANCURSARI**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Afif Muhammad Rizqiawan

NIM: 125150200111025

UNIVERSITAS BRAWIJAYA



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

PENGEMBANGAN APLIKASI MOBILE GEOTAGGING PELAPORAN KONDISI
PERTUMBUHAN DAN PRODUKSI TANAMAN PERKEBUNAN BERBASIS ANDROID
STUDI KASUS: PTPN XII KEBUN PANCURSARI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Afif Muhammad Rizqiawan
NIM: 125150200111025

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Agustus 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng. Herman Tolle, S.T, M.T
NIP: 19740823 200012 1 001

Ratih Kartika Dewi, S.T., M.Kom
NIK: 201503 890520 2 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 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, 26 Agustus 2016



Afif Muhammad Rizqiwani

NIM: 125150200111025

KATA PENGANTAR

Alhamdulillahirobbilalamin, puji dan syukur kehadiran Allah *Subhanahu Wa Ta'ala*, karena atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul **Pengembangan Aplikasi *Mobile Geotagging* Pelaporan Kondisi Pertumbuhan dan Produksi Tanaman Perkebunan Berbasis Android Studi Kasus: PTPN XII Kebun Pancursari**. Tak lupa, shalawat serta salam penulis haturkan kepada junjungan besar, Nabi Muhammad *Shalallahu Alaihi Wasallam*.

Penyusunan skripsi ini juga tak lepas dari bantuan baik itu berupa bimbingan, kritik, saran, dukungan, motivasi maupun doa banyak pihak. Oleh karena itu, ucapan terima kasih penulis sampaikan kepada :

1. Dr. Eng. Herman Tolle, S.T, M.T selaku dosen pembimbing 1 atas segala bimbingan, waktu yang diluangkan serta kritik dan saran yang telah diberikan kepada penulis.
2. Ratih Kartika Dewi, S.T., M.Kom sebagai dosen pembimbing 2 atas bimbingan dan masukan yang diberikan kepada penulis.
3. Orang tua dan keluarga besar penulis yang tersayang dan tercinta atas segala dukungan, motivasi, doa dan segala pengorbanan baik materil maupun moril.
4. Fitria Dwi Kusumaningsih sebagai rekan seperjuangan dalam penyusunan skripsi.
5. Teman – teman LPM Display dan Serdadu CL yang telah memberikan dukungan lahir dan batin.
6. Seluruh bapak dan ibu dosen Fakultas Ilmu Komputer Universitas Brawijaya Malang atas segala bimbingan serta ilmu yang telah diajarkan kepada penulis.
7. Para pegawai dan staf Program Fakultas Ilmu Komputer Universitas Brawijaya Malang.
8. Dan semua pihak yang tidak bisa disebutkan satu per satu. Terima kasih atas segala bantuannya.

Semoga penulisan laporan skripsi ini bermanfaat bagi pembaca dan untuk pengembangan selanjutnya. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna serta banyak kekurangan disebabkan oleh keterbatasan kemampuan dan pengalaman, dengan kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca.

Malang, 26 Agustus 2016

Afif Muhammad R

ABSTRAK

Afif Muhammad Rizqiwani. 2016. Pengembangan Aplikasi Mobile Geotagging Pelaporan Kondisi Pertumbuhan Dan Produksi Tanaman Perkebunan Berbasis Android Studi Kasus: PTPN XII Kebun Pancursari. Skripsi Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Dr. Eng. Herman Tolle, S.T, M.T dan Ratih Kartika Dewi, S.T., M.Kom

PT. Perkebunan Nusantara XII Kebun Pancursari merupakan perusahaan perseroan yang bergerak pada bidang komoditas perkebunan. Perkebunan ini memiliki luas areal konsesi sebesar 3.031,0740 Ha yang terdiri atas dua bagian kebun yaitu bagian Pancursari dan bagian Sumberjeru. PTPN XII Kebun Pancursari menghasilkan komoditi pokok yaitu cengkih dan karet. Selama 5 tahun terakhir, Kebun Pancursari berhasil memproduksi cengkih dengan rata-rata 103 ton per tahun. Namun dalam menjalankan bisnisnya, semua masih dilakukan secara manual. Hal ini sedikit banyak mempengaruhi jumlah produksi cengkih tiap tahun menjadi tidak stabil. Tidak ada dokumentasi terhadap perlakuan yang sedang diterapkan kepada tanaman tertentu, juga tidak ada sistem pelaporan jika terdapat tanaman yang terkena hama. Akibatnya, tanaman yang terkena hama atau penyakit yang seharusnya membutuhkan pengawasan dan pemeliharaan khusus cenderung terbengkalai. Berdasarkan permasalahan yang telah disebutkan, diperlukan suatu aplikasi mobile geotagging pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan. Memanfaatkan fitur geotagging foto, setiap laporan baik itu laporan hama maupun laporan pertumbuhan tanaman akan disertai informasi lokasi secara otomatis untuk memudahkan pengawasan. Pengembangan aplikasi didasarkan pada analisis kebutuhan dan perancangan agar sistem memenuhi kebutuhan pengguna. Implementasi dibangun dengan pendekatan native memanfaatkan web service untuk mengirim data JSON. Pengujian aplikasi meliputi pengujian validasi dan pengujian usability. Pada pengujian usability didapatkan nilai rata-rata indeks sebesar 80,4 % sehingga aplikasi termasuk pada kategori sangat mudah untuk digunakan.

Kata Kunci: *Geotagging*, Android, PTPN XII, Tanaman, Perkebunan

ABSTRACT

Afif Muhammad Rizqiawan. 2016. Mobile Android Geotagging Application Development For Reporting Plant Growth Condition And Production Case Study: PTPN XII Kebun Pancursari. Informatics Engineering's Thesis, Faculty of Computer Science, Brawijaya University, Malang. Advisor: Dr. Eng. Herman Tolle, S.T, M.T dan Ratih Kartika Dewi, S.T., M.Kom

PT. Perkebunan Nusantara XII Pancursari Gardens is a limited liability company engaged in the field of commodities. This plantation has a concession area of 3031.0740 hectares which consists of two parts, namely the Pancursari gardens and part Sumberjeru. PTPN XII Pancursari gardens produce cloves and basic commodities, namely rubber. Over the last 5 years, Pancursari Garden managed to produce clove with an average of 103 tons per year. But in running the business, all is still done manually. It's a little more influence the amount of production of cloves per year become unstable. There is no documentation of the treatment being applied to certain plants, is also no system of reporting if there are plants affected by pests. As a result, the plants affected by pests or diseases that require monitoring and maintenance should particularly likely to be abandoned. Based on the problems already mentioned, we need a mobile application geotagging reporting growing conditions and tree crop production. Utilizing the photo geotagging feature, any good report that reports of pests and plant growth will be accompanied report location information automatically for easy supervision. Application development based on the needs analysis and design of the system to meet user requirements. Implementation built with native approach utilizing the web service to send JSON data. Testing applications include validation testing and usability testing. In usability testing found the average value of the index amounted to 80.4% so that the applications included in the category of very easy to use.

Keywords: Geotagging, Android, PTPN XI, Plant, Plantation

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah.....	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 PT. Perkebunan Nusantara XII Kebun Pancursari.....	7
2.3 Kondisi Pertumbuhan Tanaman Cengkih	7
2.3.2 Hama dan Penyakit Tanaman Cengkih	8
2.4 Geotagging.....	8
2.5 GPS Android	9
2.6 Android exif.....	9
2.7 Google maps	10
2.8 Pengujian perangkat lunak	11
2.8.1 Pengujian validasi.....	11
2.8.2 Pengujian <i>usability</i>	11
2.8.3 Pengujian <i>performance</i>	12
BAB 3 METODOLOGI	13
3.1 Studi literatur	13
3.2 Analisis Kebutuhan	14

3.3 Perancangan	14
3.4 Implementasi	15
3.5 Pengujian dan analisis hasil	15
3.6 Pengambilan kesimpulan dan saran	15
BAB 4 ANALISIS KEBUTUHAN	16
4.1 Gambaran umum sistem	16
4.2 Identifikasi aktor	18
4.3 Analisis kebutuhan fungsional	18
4.3.1 <i>Use case diagram</i>	19
4.3.2 <i>Use case scenario</i>	19
4.4 Analisis kebutuhan non fungsional	23
BAB 5 PERANCANGAN DAN IMPLEMENTASI	24
5.1 Perancangan aplikasi perangkat bergerak.....	24
5.1.1 Perancangan arsitektur sistem	25
5.1.2 Perancangan <i>activity diagram</i>	25
5.1.3 Perancangan basis data.....	29
5.1.4 Perancangan antarmuka	32
5.2 Implementasi	36
5.2.1 Spesifikasi sistem	36
5.2.2 Batasan implementasi.....	37
5.2.3 Implementasi basis data	37
5.2.4 Implementasi kode program.....	38
5.2.5 Implementasi antarmuka.....	44
BAB 6 PENGUJIAN DAN ANALISIS HASIL	50
6.1 Pengujian	50
6.1.1 Pengujian validasi.....	50
6.1.2 Hasil pengujian validasi	53
6.1.3 Pengujian <i>usability</i>	54
6.1.4 Pengujian <i>performance</i>	58
6.2 Analisis hasil.....	60
6.2.1 Analisis hasil pengujian validasi	60
6.2.2 Analisis hasil pengujian <i>usability</i>	60



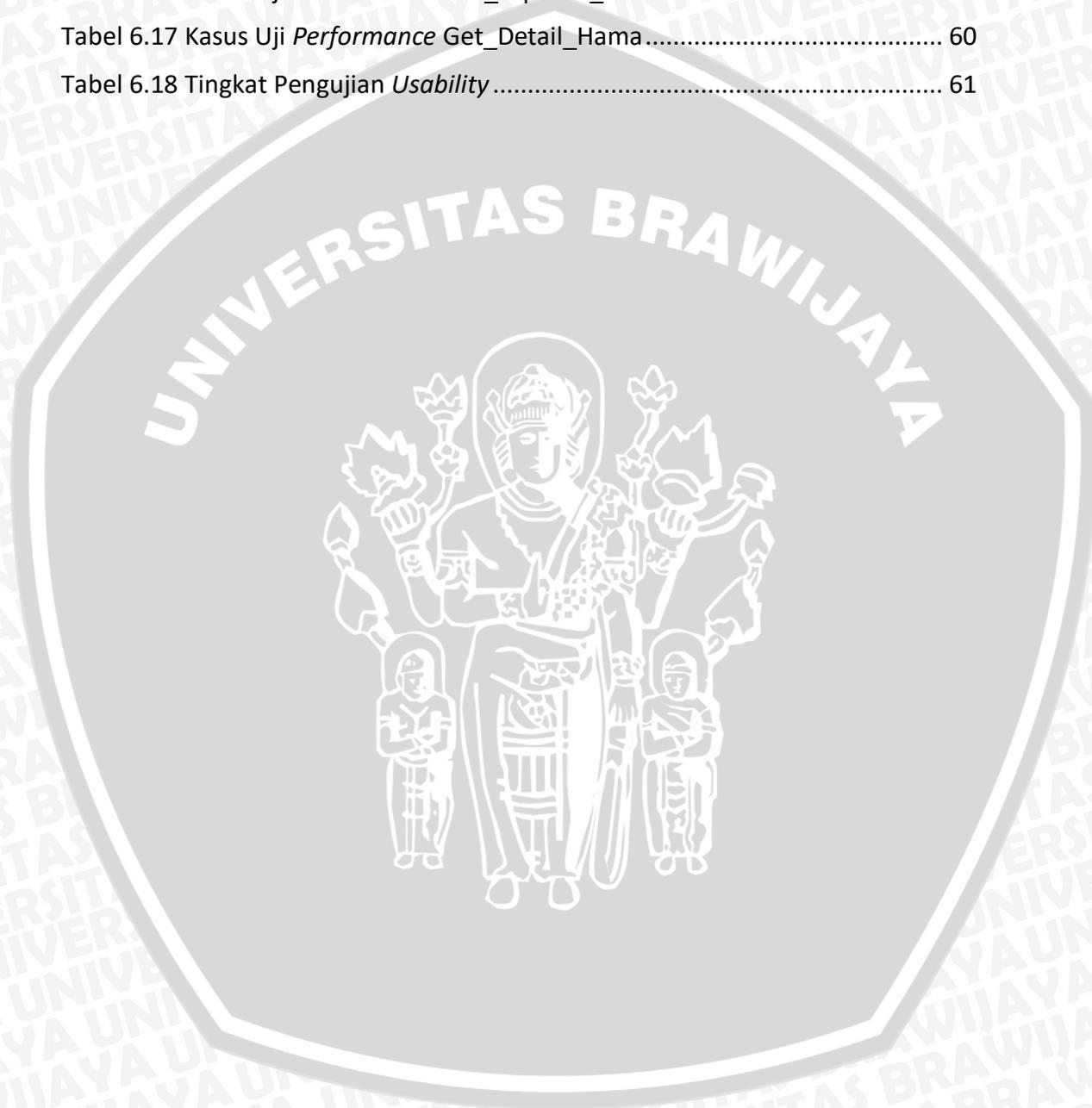
6.2.3 Analisis hasil pengujian <i>performance</i>	61
BAB 7 PENUTUP	62
7.1 Kesimpulan.....	62
7.2 Saran	62
DAFTAR PUSTAKA.....	63



DAFTAR TABEL

Tabel 2.1 Kajian pustaka	5
Tabel 4.1 Identifikasi Aktor	18
Tabel 4.2 Kebutuhan Fungsional.....	18
Tabel 4.3 <i>Use Case Scenario</i> Lihat Peta.....	19
Tabel 4.4 <i>Use Case Scenario</i> Buat Laporan Hama.....	20
Tabel 4.5 <i>Use Case Scenario</i> Buat Laporan Berkala	21
Tabel 4.6 <i>Use Case Scenario</i> Lihat Detail Laporan	21
Tabel 4.7 <i>Use Case Scenario</i> Update Laporan Hama	22
Tabel 4.8 Kebutuhan Non Fungsional	23
Tabel 5.1 Struktur Tabel Pengguna	30
Tabel 5.2 Struktur Tabel Lap_ Berkala	30
Tabel 5.3 Struktur Tabel Lap_Hama.....	31
Tabel 5.4 Struktur Tabel Pemeliharaan.....	31
Tabel 5.5 Spesifikasi Perangkat Keras Komputer	36
Tabel 5.6 Spesifikasi Perangkat Keras <i>Smartphone</i>	36
Tabel 5.7 Spesifikasi Perangkat Lunak Komputer	37
Tabel 5.8 Spesifikasi Perangkat Lunak <i>Smartphone</i>	37
Tabel 5.9 Kode Program Fungsi <code>readGeotagImage()</code>	38
Tabel 5.10 Kode Program Fungsi <code>insertHama()</code>	40
Tabel 5.11 Kode Program <code>displayMarker()</code>	42
Tabel 6.1 Kasus Uji Validasi Buat Lihat Peta	50
Tabel 6.2 Kasus Uji Validasi Buat Laporan Hama	51
Tabel 6.3 Kasus Uji Validasi Buat Laporan Berkala	51
Tabel 6.4 Kasus Uji Lihat Detail Laporan	52
Tabel 6.5 Kasus Uji Validasi Update Laporan Hama	52
Tabel 6.6 Kasus Uji Validasi	53
Tabel 6.7 Pernyataan Parameter <i>Efficiency</i>	54
Tabel 6.8 Pernyataan Parameter <i>Learnability</i>	54
Tabel 6.9 Pernyataan Parameter <i>Memorability</i>	55
Tabel 6.10 Pernyataan Parameter <i>Error</i>	55
Tabel 6.11 Pernyataan Parameter <i>Satisfaction</i>	55

Tabel 6.12 Pemetaan Penilaian Hasil Pengujian <i>Usability</i>	56
Tabel 6.13 Interpretasi Skor Likert.....	57
Tabel 6.14 Hasil Perhitungan Pengujian <i>Usability</i>	57
Tabel 6.15 Kasus Uji <i>Performance</i> Get_Laporan_Hama	59
Tabel 6.16 Kasus Uji <i>Performance</i> Get_Laporan_Berkala	59
Tabel 6.17 Kasus Uji <i>Performance</i> Get_Detail_Hama.....	60
Tabel 6.18 Tingkat Pengujian <i>Usability</i>	61



DAFTAR GAMBAR

Gambar 2.1 Contoh Data Exif Android.....	10
Gambar 3.1 Tahapan Metodologi Penelitian.....	13
Gambar 4.1 Analisis Kebutuhan.....	16
Gambar 4.2 Gambaran Umum Sistem.....	17
Gambar 4.3 <i>Use Case Diagram</i>	19
Gambar 5.1 Tahapan Perancangan dan Implementasi.....	24
Gambar 5.2 Rancangan Arsitektur Aplikasi.....	25
Gambar 5.3 <i>Activity Diagram</i> Lihat Peta.....	26
Gambar 5.4 <i>Activity Diagram</i> Buat Laporan Hama.....	26
Gambar 5.5 <i>Activity Diagram</i> Buat Laporan Berkala.....	27
Gambar 5.6 <i>Activity Diagram</i> Update Laporan Hama.....	28
Gambar 5.7 <i>Activity Diagram</i> Lihat Detail Laporan.....	29
Gambar 5.8 Entity Relationship Diagram.....	29
Gambar 5.9 <i>Screen Flow</i> Aplikasi.....	32
Gambar 5.10 Perancangan Antarmuka Login.....	33
Gambar 5.11 Perancangan Antarmuka Registrasi.....	33
Gambar 5.12 Perancangan Antarmuka Peta.....	34
Gambar 5.13 Perancangan Antarmuka Buat Laporan Hama.....	34
Gambar 5.14 Perancangan Antarmuka Buat Laporan Berkala.....	35
Gambar 5.15 Perancangan Antarmuka Detail Laporan.....	35
Gambar 5.16 Diagram Fisik Basis Data.....	38
Gambar 5.17 Implementasi Antarmuka Login.....	44
Gambar 5.18 Implementasi Antarmuka Registrasi.....	45
Gambar 5.19 Implementasi Antarmuka Buat Laporan Hama.....	46
Gambar 5.20 Implementasi Antarmuka Laporan Berkala.....	47
Gambar 5.21 Implementasi Antarmuka Peta.....	48
Gambar 5.22 Implementasi Antarmuka Detail Laporan.....	49

BAB 1 PENDAHULUAN

1.1 Latar belakang

Indonesia merupakan negara yang kaya akan keragaman flora. Tidak salah bila Indonesia mendapat sebutan negara agraris. Iklimnya sangat cocok untuk ditumbuhi berbagai jenis tanaman. Salah satu tanaman yang dinilai berprospek cerah adalah komoditas perkebunan. Tidak bisa dipungkiri, agribisnis perkebunan memegang peranan penting dalam pembangunan perekonomian di Indonesia. Pusdatin Kementerian Pertanian RI melaporkan bahwa tahun 2010 total volume ekspor sebesar 27,86 juta ton dengan nilai USD 30,70 miliar dan meningkat pada tahun 2012 menjadi 29,82 ton dengan nilai USD 32,48 miliar (Tim Penyusun, 2013). PT. Perkebunan Nusantara XII Kebun Pancursari merupakan perusahaan perseroan yang bergerak pada bidang komoditas perkebunan. Perkebunan ini memiliki luas areal konsesi sebesar 3.031,0740 Ha yang terdiri atas dua bagian kebun yaitu bagian Pancursari dan bagian Sumberjeru. PTPN XII Kebun Pancursari menghasilkan komoditi pokok yaitu cengkeh dan karet. Selama 5 tahun terakhir, Kebun Pancursari berhasil memproduksi cengkeh dengan rata-rata 103 ton per tahun (Tim Penyusun, 2015).

Namun, jumlah produksi tanaman khususnya tanaman cengkeh pada Kebun Pancursari tidak stabil tiap tahunnya. Bahkan berdasarkan data capaian produksi cengkeh Kebun Pancursari pernah mengalami penurunan produksi yang sangat drastis pada tahun 2013. Produksi cengkeh Kebun Pancursari turun menjadi 4, 8 ton setelah di tahun 2012 berhasil memproduksi 21, 3 ton (Tim Penyusun, 2015). Hal ini disebabkan karena kurangnya pengawasan terhadap pertumbuhan dan produksi tanaman yang ada di Kebun Pancursari. Selama ini pengawasan masih dilakukan manual, tidak ada sistem yang memuat data berupa gambar (foto) dan lokasi tanaman yang bisa digunakan oleh pekerja perkebunan untuk saling mengawasi dan melaporkan jika ada tanaman yang mati terkena penyakit atau hama. Akibatnya, tidak jarang tanaman yang mati tersebut lepas dari pengawasan pekerja. Ditambah lagi dengan faktor kurang bertanggung jawabnya pekerja dalam memantau pertumbuhan dan produksi tanaman. Produksi per tahun tanaman cengkeh di Kebun Pancursari menjadi naik turun secara fluktuatif. Imbasnya, diakui sangat mengganggu pendapatan usaha bisnis PTPN XII Kebun Pancursari.

Oleh karena itu, dibutuhkan suatu aplikasi yang dapat membantu PTPN XII Kebun Pancursari dalam memantau kondisi pertumbuhan dan produksi tanaman perkebunan berdasarkan lokasi. Dengan begitu, pekerja yang ada di PTPN XII Kebun Pancursari dapat memanfaatkan aplikasi ini untuk memantau dan melaporkan tanaman yang mati disertai dengan informasi lokasinya. Proses pelaporan memanfaatkan fitur geotagging pada kamera smartphone. *Geotagging* adalah gabungan fitur kamera yang dapat melakukan sinergi langsung dengan fitur GPS (*Global Positioning System*) guna memberikan informasi secara realtime di mana dan bagaimana kondisi sebuah obyek (Mardani, 2014). Dengan aplikasi ini,

diharapkan tanaman yang mati baik terkena hama atau penyakit bisa terpantau dan segera ditangani. Jika sudah tertangani, pekerja bisa memperbarui laporan kepada pimpinan dengan bukti foto tanaman yang sudah diobati memanfaatkan fitur *geotagging*. Aplikasi ini juga bisa dimanfaatkan untuk merekam data pertumbuhan tanaman dalam jangka waktu tertentu sesuai dengan pembagian area pada PTPN XII Kebun Pancursari. Data-data tersebut bisa dijadikan bahan evaluasi, mengingat selama ini belum ada sistem pemantauan dan pelaporan yang cepat dan akurat pada PTPN XII Kebun Pancursari. Lokasi mana yang sering terkena hama atau penyakit ditampilkan dalam bentuk peta persebaran yang bisa diantisipasi lebih awal sehingga produksi tanaman per tahun tetap stabil. Dengan begitu, pendapatan usaha bisnis PTPN XII Kebun Pancursari diharapkan tetap stabil bahkan meningkat tiap tahunnya.

Berdasarkan kebutuhan akan sebuah sistem pemantauan dan pelaporan yang cepat dan akurat pada PTPN XII Kebun Pancursari diperlukan suatu aplikasi *mobile* yang akan dikembangkan dalam skripsi ini. Aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan diharapkan bisa membantu pekerja dalam hal pemantauan dan pelaporan pertumbuhan serta produksi tanaman yang cepat dan akurat. Memanfaatkan fitur *geotagging* pekerja tidak perlu memberikan keterangan lokasi pada foto tanaman yang akan diunggah pada aplikasi secara manual. Aplikasi akan otomatis menambahkan informasi lokasi. Pekerja juga tidak bisa memanipulasi foto lokasi tanaman jika ada laporan tanaman yang terkena masalah. Data foto beserta informasi lokasinya akan ditampilkan ke dalam peta pada aplikasi sesuai areal konsesi PTPN XII Kebun Pancursari. Penerapan aplikasi ini diharapkan dapat mewujudkan pengawasan yang lebih akurat sehingga produksi tanaman per tahun tetap stabil. Pendapatan usaha bisnis PTPN XII Kebun Pancursari bisa stabil bahkan meningkat tiap tahunnya.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, permasalahan yang akan dibahas sebagai berikut:

1. Bagaimana rancangan dan implementasi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android?
2. Bagaimana tingkat *usability* dan performa waktu eksekusi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android sehingga dapat memudahkan pekerja dalam proses pelaporan kondisi tanaman?

1.3 Tujuan

1. Merancang dan mengimplementasikan aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.
2. Menguji aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan ditinjau dari tingkat *usability* dan performa waktu eksekusi.

1.4 Manfaat

Penelitian ini diharapkan bisa memberikan manfaat. Adapun manfaat dari penelitian dari penelitian ini sebagai berikut:

1. Memudahkan pengawasan terhadap tanaman perkebunan yang ada di areal konsesi PTPN XII Kebun Pancursari
2. Membantu memberikan informasi data persebaran area kondisi pertumbuhan tanaman serta produksi tanaman yang bisa dijadikan bahan evaluasi untuk peningkatan produksi
3. Memudahkan dalam mengatur produksi tanaman setiap tahun agar tetap stabil tinggi, tidak naik tinggi dan turun drastis.

1.5 Batasan masalah

Ruang lingkup masalah penelitian dengan batasan sebagai berikut:

1. Identifikasi tanaman yang mati baik itu terkena hama atau penyakit tidak dilakukan oleh kamera *smartphone* melainkan pekerja yang notabene merupakan ahli di bidang pertanian
2. Informasi lokasi pada metada foto untuk layanan *geotagging* berupa *longitude* dan *latitude*.
3. Tanaman yang digunakan sebagai objek penelitian adalah komoditas utama PTPN XII Kebun Pancursari yaitu tanaman cengkeh. Tetapi, diharapkan aplikasi ini bisa diterapkan di berbagai macam tanaman.

1.6 Sistematika pembahasan

Secara garis besar, struktur penulisan penelitian skripsi ini sebagai berikut:

BAB I Pendahuluan

Bab ini membahas tentang latar belakang permasalahan, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan

BAB II Landasan Kepustakaan

Pada bab ini membahas tentang referensi dan konsep teori yang digunakan sebagai dalam penyusunan skripsi.

BAB III Metodologi

Membahas tentang metode yang digunakan dalam penulisan yang terdiri dari studi literatur, perancangan perangkat lunak, implementasi perangkat lunak, pengujian dan analisis.

BAB IV Analisis Kebutuhan

Pada bab ini membahas tentang analisis terhadap kebutuhan sistem, baik fungsional maupun non-fungsional.

BAB V Perancangan dan Implementasi

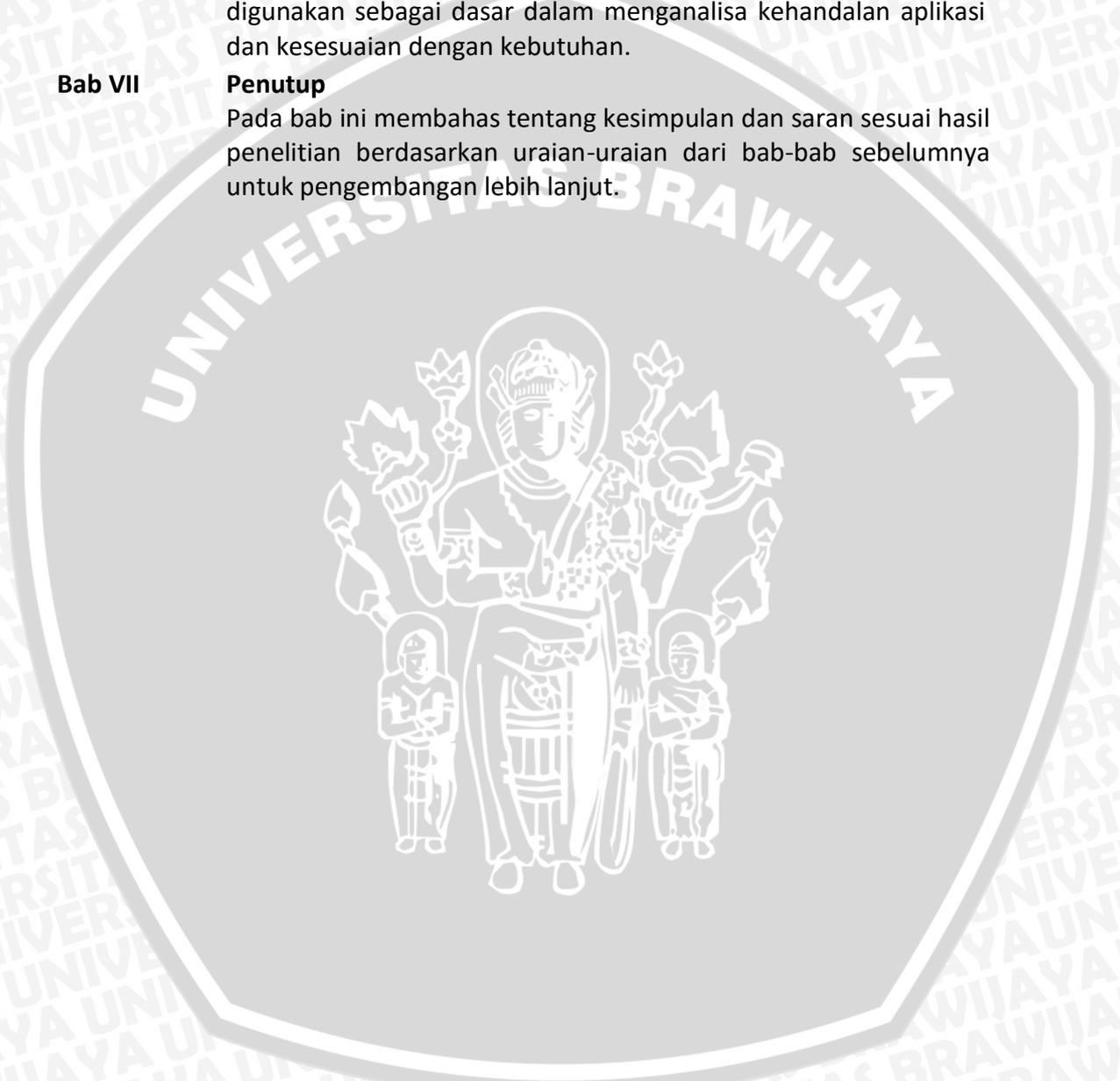
Pada bab ini membahas tentang perancangan dan implementasi *geotagging* dalam membangun aplikasi pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan. Implementasi ini dilakukan sesuai dengan hasil analisa dan perancangan.

BAB VI Pengujian dan Analisis Hasil

Pada bab ini membahas tentang strategi dan teknik pengujian terhadap sistem yang dibangun. Hasil pengujian tersebut dapat digunakan sebagai dasar dalam menganalisa kehandalan aplikasi dan kesesuaian dengan kebutuhan.

Bab VII Penutup

Pada bab ini membahas tentang kesimpulan dan saran sesuai hasil penelitian berdasarkan uraian-uraian dari bab-bab sebelumnya untuk pengembangan lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dibahas mengenai kajian pustaka dari penelitian – penelitian sebelumnya yang terkait dengan penelitian ini dan dasar teori yang menunjang penulisan skripsi ini. Dasar teori digunakan sebagai dasar pembuatan perangkat lunak dan dasar informasi tentang objek penelitian meliputi kondisi tanaman perkebunan yang ada di PTPN XII Kebun Pancursari yaitu tanaman cengkeh. Kondisi tanaman yang dimaksud untuk kebutuhan pelaporan pada aplikasi *geotagging*.

2.1 Kajian Pustaka

Kajian pustaka merupakan *review* penelitian-penelitian sebelumnya yang berkaitan dengan penggunaan *geotagging* dalam aplikasi berbasis Android. Tujuan dari dicantumkannya kajian pustaka adalah sebagai referensi yang dapat membantu proses pengerjaan penelitian ini.

Tabel 2.1 Kajian pustaka

No.	Judul	Penulis	Perbandingan	
			Kajian Pustaka	Skripsi Penulis
1	Rancang Bangun Aplikasi Sosial Geotagging Keindahan Alam Indonesia Pada Sistem Operasi Android	Arik Achmad Efendy	Implementasi <i>geotagging</i> untuk mengumpulkan informasi lokasi keindahan alam Indonesia dari foto yang diunggah masyarakat	Penggunaan <i>geotagging</i> untuk sistem pelaporan kondisi dan lokasi pertumbuhan tanaman dari foto yang diunggah pekerja
2	Rancang Bangun Aplikasi Mobile Social Geotagging	Eka Ariyansyah	Implementasi <i>geotagging</i> untuk promosi keberagaman	Penggunaan <i>geotagging</i> untuk sistem pelaporan

	<p>Keragaman Budaya Indonesia Pada Platform Android</p>		<p>kebudayaan Indonesia berdasarkan foto dan informasi lokasinya</p>	<p>kondisi dan lokasi pertumbuhan tanaman dari foto yang diunggah pekerja</p>
3	<p>Sistem Informasi Geografis Pelaporan Masyarakat (Sigma) Berbasis Foto Geotag</p>	<p>Ary Mardani</p>	<p>Implementasi <i>geotagging</i> untuk pelaporan kerusakan infrastruktur, permasalahan sosial, dan pencemaran yang terjadi di lingkungan masyarakat</p>	<p>Penggunaan <i>geotagging</i> untuk sistem pelaporan kondisi dan lokasi pertumbuhan tanaman dari foto yang diunggah pekerja</p>
4	<p>Perancangan dan Pembangunan Perangkat Lunak Photo Uploader pada Facebook dengan Fitur <i>Geotagging</i></p>	<p>Annisa Nur Sari</p>	<p>Implementasi <i>geotagging</i> untuk pemberian informasi lokasi secara otomatis pada foto yang diunggah ke dalam Facebook</p>	<p>Penggunaan <i>geotagging</i> untuk sistem pelaporan kondisi dan lokasi pertumbuhan tanaman dari foto yang diunggah pekerja</p>

2.2 PT. Perkebunan Nusantara XII Kebun Pancursari

PT. Perkebunan Nusantara XII (Persero) Kebun Pancursari terletak di Kecamatan Sumber Manjing Wetan Kabupaten Malang dengan luas areal konsesi sebesar 3.031,074 Ha. Terdiri atas dua kebun bagian yaitu bagian Pancursari seluas 1.687,6711 Ha dan bagian Sumberjeru seluas 1.343,4029 Ha. Bagian Pancursari terletak di desa Ringinkembar terdiri atas afdeling:

- Pager Gunung (PG) = 266,4600 Ha
- Pancursari Barat (PSB) = 592,1000 Ha
- Pancursari Timur (PST) = 829,1111 Ha.

Bagian Sumberjeru terletak di desa Tegalrejo terdiri atas afdeling:

- Bumirejo (BR) = 377,1621 Ha
- Sumber Manggis = 536,5282 Ha
- Glagah Arum (GA) = 429,7126 Ha.

Kebun Pancursari terdiri atas empat komoditi pokok yaitu cengkih, karet, tebu, aneka kayu dan tanaman semusim. Cengkih dan karet mendominasi komposisi areal dengan presentase sebesar 33% dan 29%. Tetapi, untuk produksi perkebunan yang diunggulkan selama ini adalah tanaman cengkih (Tim Penyusun, 2015).

2.3 Kondisi Pertumbuhan Tanaman Cengkih

Pemeliharaan tanaman cengkih harus dilakukan secara intensif sejak ditanam hingga umur empat tahun. Umur tersebut merupakan masa kritis bagi tanaman cengkih. Pemeliharaan tanaman cengkih meliputi penyulaman dan penyiraman, penyiangan, pemupukan serta penggemburan tanah (Suwarto, et al., 2014) sebagai berikut:

1. Penyulaman dan penyiraman

Penyulaman dilakukan hingga tanaman berumur dua tahun. Sehingga selama dua tahun pertama, tanaman harus selalu diamati. Pengamatan itu meliputi pengecekan jika ada tanaman yang mati harus segera diganti dengan bibit tanaman baru yang baik. Penyiraman pada bibit yang baru ditanam dilakukan pada sore hari setiap 2-3 hari sekali, terutama saat musim kemarau.

2. Penyiangan

Penyiangan perlu dilakukan untuk menghilangkan gulma agar tidak mengganggu pertumbuhan tanaman cengkih. Bila tidak, gulma akan menjadi pesaing tanaman cengkih dalam mendapatkan unsur hara. Biasanya, penyiangan dilakukan pada awal dan akhir musim hujan. Bersamaan dengan penyiangan, perlu dilakukan penggemburan tanah terutama tanah yang padat dan berat.

3. Pemupukan

Pemupukan dilakukan untuk meningkatkan ketersediaan unsur hara pada tanaman. Pupuk kandang diberikan minimal sekali dengan dosis 30 – 60 kg tiap pohon per tahun. Pemberiannya dengan cara membuat alur pupuk melingkar sejauh bentuk kanopi terluar tanaman cengkih. Pupuk lain yang diberikan selain pupuk kandang yaitu pupuk NPK, urea, atau ZA. Dalam setahun, pemupukan ini dilakukan dua kali pada awal musim hujan (September – Oktober) dan awal musim kemarau (Maret – April).

2.3.2 Hama dan Penyakit Tanaman Cengkih

Hama dan penyakit utama yang sering menurunkan produktivitas tanaman cengkih adalah sebagai berikut (Suwarto, et al., 2014):

a. Penghisap daun cengkih (*Helopeltis sp.*)

Serangan hama *Helopeltis sp.* menimbulkan bintik-bintik cokelat pada permukaan daun. Pohon yang rentan terkena serangan biasanya pohon yang merana. Jika serangan berat, daun akan sangat menderita. Untuk mengobatinya dilakukan dengan menyemprotkan insektisida kontak atau mengolesi pangkal batang dengan insektisida sistemik serta memotong ujung-ujung ranting yang mati agar tumbuh tunas baru yang lebih baik.

b. Rayap

Tanaman cengkih yang masih muda dan baru ditanam, tanaman yang kurang sehat dan tanaman di persemaian adalah tanaman yang rentan terkena serangan rayap. Untuk mengantisipasi serangan rayap, dilakukan dengan penyemprotan racun antirayap pada lubang tanam sebelum dilakukan penanaman. Selain insektisida, afval tembakau dari pabrik rokok yang telah dicampur dengan cengkih juga dapat digunakan sebagai pengobatan.

c. Cacar daun

Penyakit cacar daun menimbulkan bercak-bercak menggelembung seperti cacar. Bercak tampak jelas pada daun yang masih muda atau daun yang terserang sejak muda. Serangan yang berat dapat mengakibatkan daun berkerut dan mengeriting. Penyakit ini dapat menyerang tangkai daun, bunga dan buah. Cacar daun disebabkan oleh cendawan *Phylostica sp.*

d. Busuk akar

Kondisi tanah yang berdrainase kurang baik atau dekat dengan sumber air rentan terhadap penyakit busuk akar. Penyakit ini disebabkan oleh cendawan *Pytium rhizoctania* dan *Phytophthora*. Gejala yang ditimbulkan adalah pohon tidak sehat, daunnya jarang dan banyak ranting yang kering. Penyakit busuk akar hanya dapat diatasi dengan pembuatan drainase yang dalam atau tanah digali di sekeliling tanaman.

2.4 Geotagging

Geotagging merupakan proses penambahan informasi geospasial pada berbagai media seperti foto dan video dalam bentuk metadata. Media yang telah mengalami proses geotagging akan memiliki informasi koordinat berupa *longitude*

(bujur), *latitude* (lintang) dan *altitude* (ketinggian). Hal tersebut memungkinkan media dapat diposisikan secara tepat pada peta (Entwistle, 2010). Terdapat tiga cara melakukan *geotagging* pada suatu media sebagai berikut (Nandipati, 2011):

- a. Kamera digital yang dilengkapi dengan GPS.
Perangkat *built-in* GPS saat ini dapat ditemukan pada kamera digital dan kamera pada *smartphone*. Informasi lokasi seperti *longitude* dan *latitude* akan ditambahkan pada foto secara otomatis pada EXIF *header*-nya.
- b. Kamera digital disinkronkan dengan GPS terpisah.
Meski tidak memiliki *built-in* GPS, kamera digital tetap dapat melakukan *geotagging* dengan cara menyinkronkan dengan GPS. Dengan menggunakan perangkat lunak, waktu yang direkam oleh kamera akan disinkronkan dengan waktu yang direkam oleh GPS dan berdasarkan waktu tersebut koordinat yang direkam oleh GPS akan ditambahkan sebagai informasi EXIF pada foto. Faktor terpenting pada cara ini adalah mengatur agar waktu pada kamera dan GPS sama.
- c. *Geocoding* manual.
Geocoding manual adalah proses penambahan informasi lokasi dengan cara menambahkan koordinat atau dengan memilih lokasi pada peta saat melakukan *upload* foto ke server. Tingkat akurasi posisi tergantung pada tools atau GPS *receiver* yang digunakan untuk mendapatkan koordinat.

Pada penelitian ini metode *geotagging* yang digunakan untuk melakukan penambahan informasi lokasi foto adalah *geocoding* manual. Kode program untuk penambahan informasi lokasi otomatis berjalan ketika pengguna memilih opsi menggunakan kamera untuk mengambil foto. Pengguna tidak perlu untuk memilih lokasi. Lokasi yang ditambahkan merupakan lokasi pengguna yang akan disimpan dalam *database* server.

2.5 GPS Android

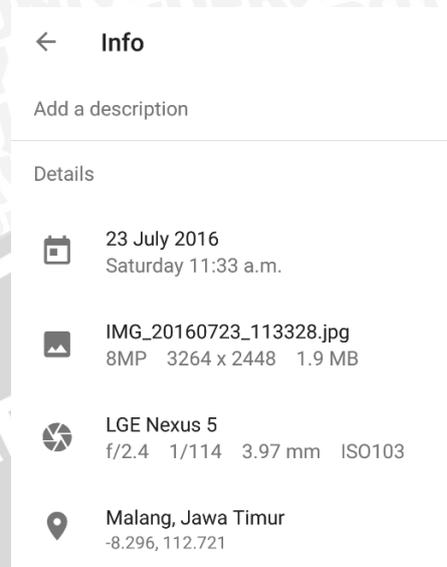
Global Positioning System atau biasa disingkat GPS merupakan sistem navigasi yang digunakan untuk menentukan posisi suatu tempat dengan sinkronisasi sinyal yang didapat dari satelit.

Pada *smartphone* Android yang sudah mendukung GPS, proses penentuan posisi pengguna menggunakan GPS *receiver* yang mengambil informasi lokasi dan akan melakukan perhitungan *triangulation* untuk menentukan lokasi pengguna dengan tepat. GPS *receiver* membandingkan waktu sinyal dikirim dengan waktu sinyal tersebut diterima untuk mengetahui jarak satelit (Holdener, 2011).

2.6 Android exif

Exif adalah format standar untuk berkas gambar, suara, dan tambahan lainnya yang digunakan oleh kamera digital (termasuk *smartphone*), pemindai, dan sistem lainnya yang terkait dengan perekaman gambar dan suara oleh kamera digital. Spesifikasinya menggunakan format berkas yang telah ada dengan label metadata tambahan tertentu (Mardani, 2014).

Exif mendukung Android mulai versi Eclair, `ExifInterface` merupakan *class* yang digunakan untuk menulis dan membaca Exif *tags* pada *file* JPEG (Android Developers, 2016).



Gambar 2.1 Contoh Data Exif Android

2.7 Google maps

Google Maps adalah layanan peta online gratis yang disediakan oleh Google. Google Maps dapat diakses melalui situs <http://maps.google.com> (Sholeh, et al., 2011). Google Maps seperti peta nyata pada umumnya, memberikan informasi geografis berupa nama-nama jalan yang ada di seluruh dunia. Google Maps juga dapat menampilkan nama gedung atau bangunan fasilitas umum seperti ATM dan Pom Bensin (Minarni & Yusdi, 2015).

Pada penelitian ini Google Maps digunakan oleh aplikasi untuk menampilkan sebuah peta yang di dalamnya terdapat foto-foto laporan kondisi tanaman yang ditempatkan berdasarkan lokasi pengambilan setiap foto tersebut. Foto-foto ditampilkan di atas peta sebagai *marker* yang telah dikustomisasi. *Marker* pada Google Maps merupakan identifikasi dari sebuah lokasi. *Marker* dapat ditambahkan ke dalam peta dengan memanggil konstruktor `google.maps.Marker`. Konstruktor tersebut memiliki sebuah objek `Marker options` yang digunakan untuk memberikan spesifikasi awal sebuah *marker* (Google, 2015). Hal-hal yang penting untuk dispesifikasikan saat akan membuat *marker* adalah sebagai berikut (Google, 2015):

- `position` (*required*) dispesifikasikan menggunakan `LatLng` sebagai lokasi awal *marker*.
- `map` (*optional*) dispesifikasikan menggunakan `Map` sebagai sebuah peta yang akan ditempatkan *marker* di dalamnya.

Sebuah marker pada Google Maps secara default dilambangkan dengan ikon berbentuk *pushpin* berwarna merah. Namun ikon tersebut bisa diubah dengan gambar yang kita inginkan. Untuk melakukan kustomisasi ikon marker, perlu ditambahkan parameter *icon* pada konstruktor `google.maps.Marker` (Google, 2015).

2.8 Pengujian perangkat lunak

Pengujian perangkat lunak merupakan kegiatan yang bertujuan untuk mengevaluasi atribut atau kemampuan program. Pengujian juga dapat diartikan sebagai proses menemukan error pada perangkat lunak sebelum dikirim ke pengguna (Rouf, 2012). Pengujian sangat penting dilakukan karena untuk meningkatkan kualitas perangkat lunak.

2.8.1 Pengujian validasi

Pengujian validasi dilakukan untuk menguji kebutuhan fungsional dari aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android, dilakukan dengan menggunakan metode *black box*. *Black box testing* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian *black box* dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dari perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan (Rosa & Shalahuddin, 2011).

2.8.2 Pengujian *usability*

Pengujian *usability* dilakukan untuk mengetahui tingkat kemudahan penggunaan antarmuka aplikasi. Pengukuran *usability* menggunakan 5 kriteria, yaitu:

1. *Learnability*
Mengukur tingkat kemudahan ketika pengguna pertama kali menggunakan suatu produk.
2. *Efficiency*
Mengukur kecepatan mengerjakan tugas tertentu setelah mempelajari produk tersebut.
3. *Memorability*
Seberapa cepat pengguna dapat mengingat langkah-langkah atau proses yang dilakukan untuk mengerjakan tugas tertentu setelah beberapa waktu.
4. *Errors*
Melihat seberapa banyak kesalahan yang dilakukan pengguna, separah apa kesalahan yang dibuat, dan semudah apa mereka mendapatkan penyelesaian.
5. *Satisfaction*
Mengukur tingkat kepuasan dalam menggunakan produk.

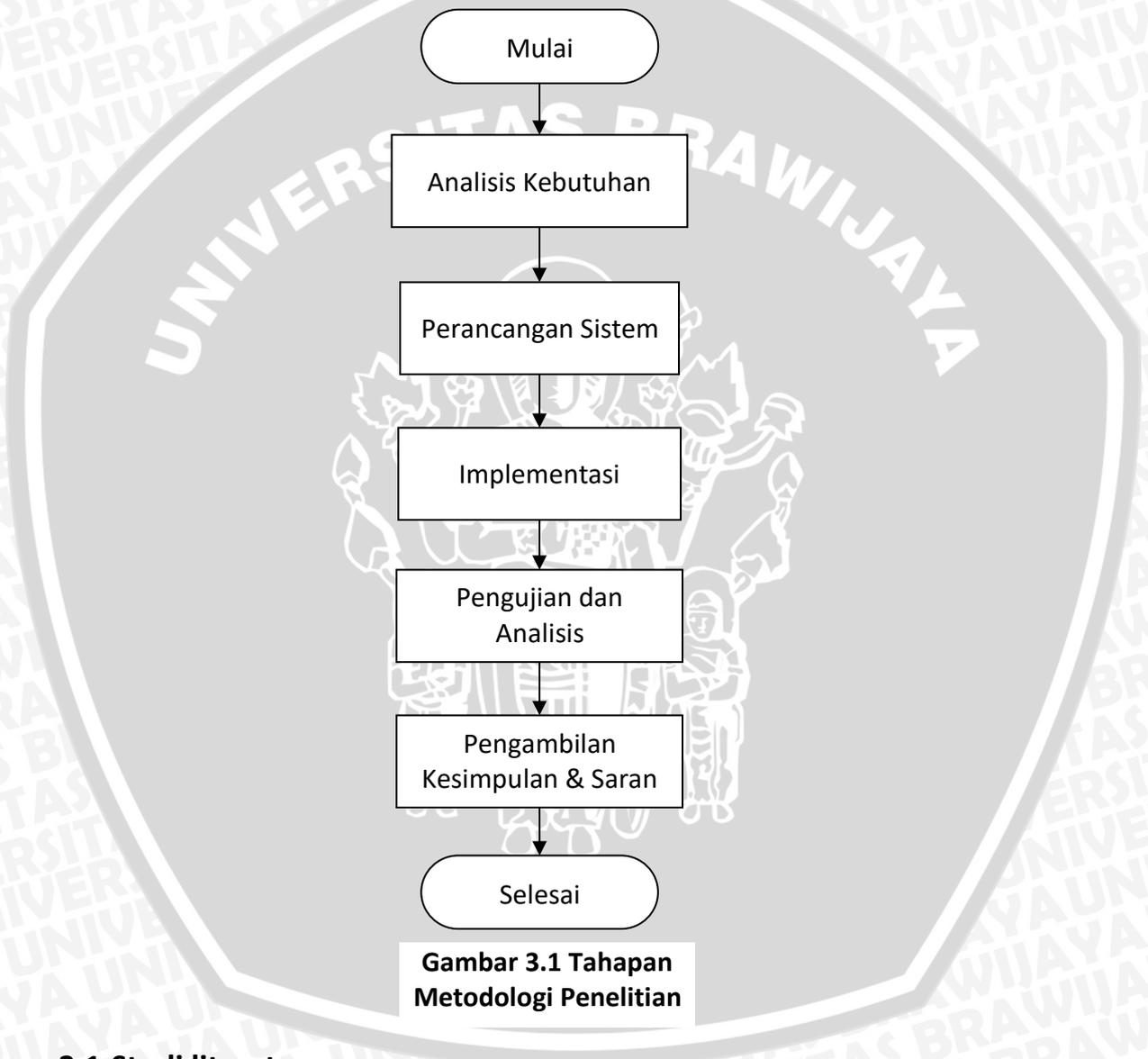
2.8.3 Pengujian *performance*

Pengujian *performance* menekankan pada tingkat performansi aplikasi dalam memuat data atau respon yang diinginkan (Pressman, 2010). Pengukuran performansi didapatkan dari waktu yang dibutuhkan untuk menampilkan data yang diinginkan. Berdasarkan pengujian ini dapat diketahui apakah waktu respon yang diberikan sistem sudah sesuai dengan batasan yang diberikan.



BAB 3 METODOLOGI

Pada bab ini akan dijelaskan tentang langkah-langkah yang dilakukan dalam pengerjaan penelitian. Terdapat lima tahap utama dalam penelitian ini yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis serta pengambilan kesimpulan. Selain itu terdapat proses studi literatur yang dilakukan dari awal hingga akhir penelitian. Gambar 3.1 memperlihatkan diagram alir dari langkah-langkah dalam pengerjaan penelitian ini.



3.1 Studi literatur

Studi literatur merupakan tahapan pencarian literatur yang berguna untuk menunjang penyusunan dasar teori dalam pengembangan aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android. Literatur tersebut diperoleh dari buku, jurnal, dan website terkait. Teori-teori penunjang tersebut meliputi:

- a. PT. Perkebunan Nusantara XII Kebun Pancursari
- b. Kondisi pertumbuhan tanaman cengkih
- c. *Geotagging*
- d. GPS Android
- e. Android exif
- f. Google maps
- g. Pengujian perangkat lunak

3.2 Analisis Kebutuhan

Pada tahap analisis kebutuhan dilakukan identifikasi kebutuhan guna memperoleh kebutuhan-kebutuhan dari aplikasi yang akan dikembangkan. Dalam melakukan identifikasi kebutuhan aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android meliputi beberapa tahapan, yaitu gambaran umum aplikasi yang menjelaskan jalannya aplikasi secara umum. Identifikasi aktor yang bertujuan untuk mengetahui peran dan interaksi aktor terhadap aplikasi. Selanjutnya dijabarkan kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional digambarkan dalam *use case* diagram dan dijelaskan dalam skenario *use case*.

3.3 Perancangan

Tahap perancangan merupakan tahapan yang dilakukan setelah seluruh kebutuhan aplikasi diperoleh dalam proses analisis kebutuhan. Perancangan aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android dijabarkan dalam arsitektur sistem. Dalam perancangan arsitektur sistem dijelaskan perancangan komunikasi data karena aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android merupakan aplikasi yang berbasis client server. Sehingga terdapat komunikasi data berupa *request* dan *response* antara *client* dan *server*.

Langkah selanjutnya adalah perancangan *activity* diagram yang menggambarkan jalannya suatu *use case* sistem dengan aktor berdasarkan *use case scenario*. Setelah perancangan *activity* diagram, langkah selanjutnya adalah perancangan basis data. Perancangan basis data dalam aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android dijabarkan dalam bentuk diagram ERD (*Entity Relationship Diagram*).

Langkah terakhir adalah perancangan antarmuka. Dalam perancangan antarmuka terdapat perancangan *screen flow* yang bertujuan untuk merancang alur navigasi dari tiap halaman yang terdapat dalam aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

3.4 Implementasi

Tahap implementasi mengacu dari hasil perancangan yang telah dilakukan. Implementasi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android dimulai dengan spesifikasi sistem, dimana dalam tahap tersebut memuat penjabaran dari spesifikasi perangkat keras, spesifikasi perangkat lunak dan spesifikasi perangkat bergerak yang digunakan dalam proses implementasian aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android. Kemudian dijelaskan batasan-batasan dalam implementasi aplikasi tersebut.

Langkah selanjutnya adalah implementasi basis data. Implementasi basis data server menggunakan basis data MySQL mengacu pada perancangan *Entity Relationship Diagram*. Tahap selanjutnya adalah implementasi kode program. Aplikasi dibangun dengan pendekatan *native*, sehingga implementasi kode program pada aplikasi *client* menggunakan bahasa pemrograman java, komunikasi data menggunakan format JSON. Tahap terakhir yaitu implementasi antarmuka. Implementasi antarmuka mengacu pada proses perancangan antarmuka.

3.5 Pengujian dan analisis hasil

Tahap pengujian bertujuan untuk mengetahui apakah hasil implementasi sudah sesuai dengan analisis kebutuhan dan perancangan yang telah dijabarkan pada tahap sebelumnya. Dalam pengujian aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android memiliki beberapa pengujian, yaitu pengujian validasi, pengujian *usability* dan pengujian *performance*.

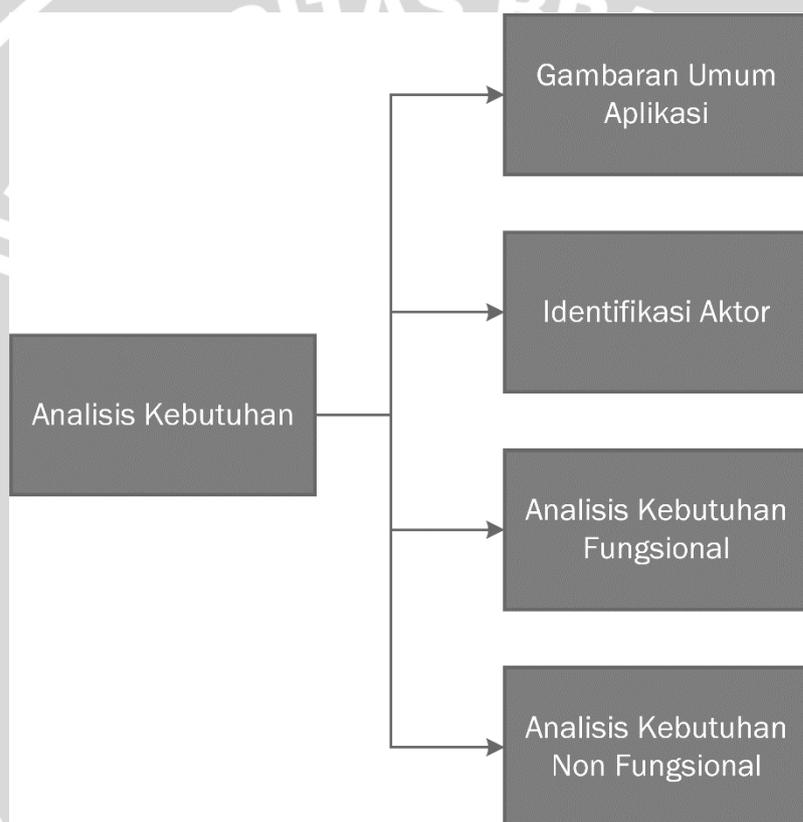
Pengujian validasi menggunakan metode *black-box* untuk menguji kebutuhan fungsional aplikasi. Sedangkan pengujian kebutuhan non fungsional aplikasi menggunakan pengujian *usability* dan pengujian *performance*. Pengujian *usability* dilakukan mengetahui tingkat kemudahan penggunaan aplikasi bagi pengguna. Pengujian *performance* dilakukan dengan mengukur waktu eksekusi untuk menampilkan data yang diinginkan pengguna. Hasil pengujian ini akan dianalisis untuk diambil kesimpulan apakah aplikasi yang telah dibangun sesuai sudah memenuhi kebutuhan atau belum.

3.6 Pengambilan kesimpulan dan saran

Pengambilan kesimpulan dan saran merupakan tahap akhir dari penelitian ini. Kesimpulan diambil berdasarkan hasil dari tahap pengujian dan analisis sistem. Kesimpulan dapat diambil ketika semua kebutuhan yang telah didefinisikan sudah terpenuhi dalam sistem. Kesimpulan juga merupakan jawaban dari rumusan masalah. Sedangkan saran merupakan ide atau pikiran yang diberikan dengan tujuan untuk memperbaiki kesalahan dan melengkapi kekurangan dari penelitian ini. Saran dapat dijadikan pertimbangan dalam pengembangan aplikasi selanjutnya.

BAB 4 ANALISIS KEBUTUHAN

Bab ini berisi tentang tahapan analisis dan perancangan aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan. Sebuah aplikasi yang baik harus dibangun berdasarkan kebutuhan pengguna yang didapat dari proses analisis kebutuhan. Kebutuhan (*requirement*) tersebut akan dikembangkan menjadi fitur-fitur yang ada pada aplikasi. Sehingga aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android ini dapat memenuhi kebutuhan pengguna. Tahapan analisis kebutuhan dimulai dari penjelasan gambaran umum sistem, identifikasi aktor serta penjabaran analisis kebutuhan fungsional dan non fungsional. Gambar 4.1 menunjukkan tahapan analisis kebutuhan aplikasi.



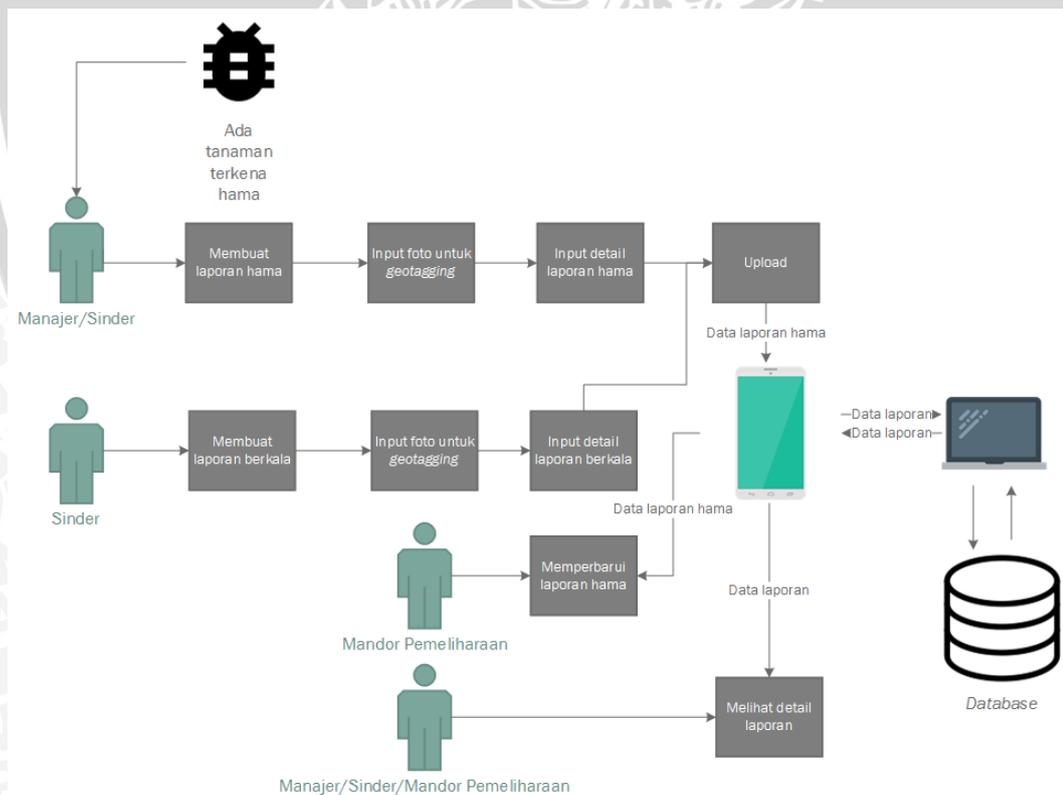
Gambar 4.1 Analisis Kebutuhan

4.1 Gambaran umum sistem

Penelitian ini mengembangkan sebuah aplikasi *mobile* yang memanfaatkan fitur *geotagging* pada kamera perangkat Android. Aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android merupakan aplikasi yang dirancang untuk membantu pelaporan proses pertumbuhan dan produksi tanaman perkebunan. Sesuai dengan bisnis proses yang ada di PTPN XII Pancursari, pengguna yang terlibat dalam aplikasi dibagi menjadi 3, yaitu manajer, sinder dan mandor pemeliharaan. Manajer bertanggung jawab atas seluruh area perkebunan Pancursari. Sinder bertanggung jawab

kepada manajer atas kondisi tanaman dan produksinya, sinder juga bertanggung jawab atas bagian area tertentu di PTPN XII Pancursari. Seperti dijelaskan pada bab 2, areal konsesi PTPN XII Pancursari dibagi menjadi beberapa bagian. Sedangkan mandor pemeliharaan merupakan pengguna yang bertanggung jawab atas tanaman yang terkena hama atau penyakit.

Untuk penerapan bisnis proses ke dalam aplikasi, terdapat fitur untuk membuat laporan hama yang bisa diakses oleh manajer dan sinder ketika ditemukan tanaman yang terkena hama atau penyakit. Setelah itu, mandor pemeliharaan akan mengobati tanaman yang terkena hama atau penyakit. Agar tanaman yang terkena hama atau penyakit dapat terpantau sudah dilakukan pemeliharaan atau belum, terdapat fitur untuk meng-*update* laporan hama yang hanya bisa diakses oleh mandor pemeliharaan. Penerapan pelaporan kondisi pertumbuhan dan produksi tanaman dalam aplikasi terdapat pada fitur untuk membuat laporan berkala yang hanya bisa diakses oleh sinder. Laporan berkala ini digunakan oleh sinder sebagai pertanggung jawaban laporan kondisi dan produksi tanaman kepada manajer. Manajer melihat hasil pelaporan melalui fitur untuk melihat detail laporan, meski begitu semua pengguna dalam aplikasi dapat melihat tiap-tiap detail laporan yang dibuat. Semua laporan yang telah dibuat akan ditampilkan pada fitur untuk melihat peta. Laporan ditampilkan dalam bentuk *marker*. *Geotagging* dalam aplikasi ini diterapkan pada penggunaan foto untuk tiap laporan yang dibuat. Gambar 4.2 menunjukkan gambaran umum sistem.



Gambar 4.2 Gambaran Umum Sistem

4.2 Identifikasi aktor

Tahap identifikasi aktor berisi penjelasan aktor-aktor yang berinteraksi dan menggunakan aplikasi. Tabel 4.1 memuat daftar aktor beserta masing-masing deskripsinya.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Manajer	Manajer adalah pengguna yang memiliki tanggung jawab atas seluruh area perkebunan. Dalam aplikasi manajer dapat membuat laporan hama, untuk laporan berkala manajer hanya bisa melihat detail laporan berkala yang dibuat oleh sinder.
Sinder	Sinder adalah pengguna yang memiliki tanggung jawab atas suatu area dalam perkebunan. Sinder bertanggung jawab melaporkan kondisi tanaman secara berkala. Dalam aplikasi sinder dapat membuat laporan hama dan laporan berkala.
Mandor Pemeliharaan	Mandor pemeliharaan merupakan pengguna yang bertanggung jawab atas tanaman yang terkena hama atau penyakit. Dalam aplikasi, mandor pemeliharaan memiliki fungsi untuk memperbarui laporan hama jika tanaman sudah dilakukan pemeliharaan atau diobati.

4.3 Analisis kebutuhan fungsional

Tahap analisis kebutuhan fungsional dilakukan untuk mengetahui kebutuhan apa saja yang harus dipenuhi oleh aplikasi. Tabel 4.2 memuat daftar kebutuhan fungsional aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

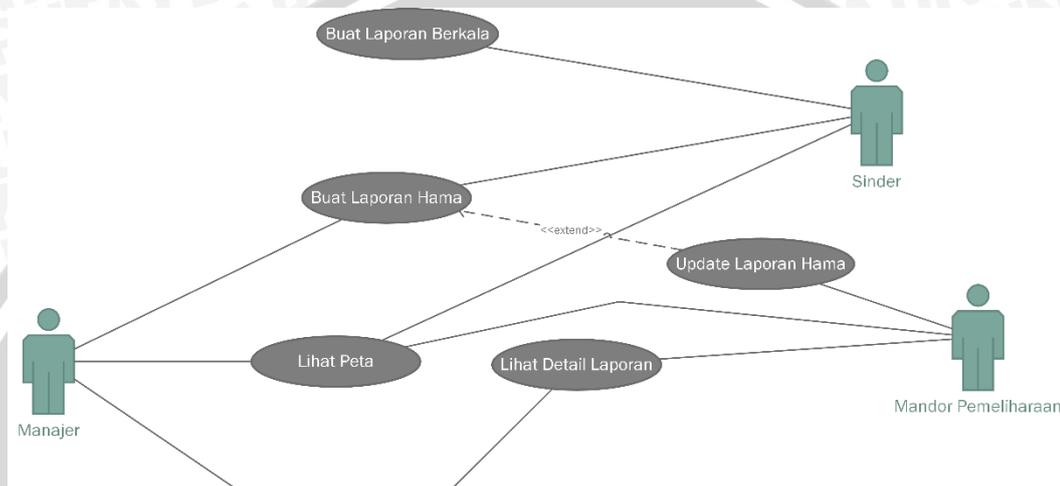
Tabel 4.2 Kebutuhan Fungsional

Kode Fungsi	Kebutuhan Fungsional	Use case
SRS_GEO_001	Aplikasi harus menyediakan halaman beranda. Halaman beranda memuat peta yang akan diisi oleh <i>marker-marker</i> yang jika diklik akan menampilkan detail laporan.	Lihat Peta
SRS_GEO_002	Aplikasi harus menyediakan fitur untuk membuat laporan mengenai tanaman yang terkena penyakit atau hama.	Buat Laporan Hama
SRS_GEO_003	Aplikasi harus menyediakan fitur untuk membuat laporan kondisi tanaman secara berkala.	Buat Laporan Berkala
SRS_GEO_004	Aplikasi harus mampu menampilkan detail laporan yang dipilih oleh pengguna.	Lihat Detail Laporan

SRS_GEO_005	Aplikasi harus menyediakan fitur untuk memperbarui status laporan hama jika tanaman yang bersangkutan sudah diobati.	Update Laporan Hama
-------------	--	---------------------

4.3.1 Use case diagram

Use case diagram menggambarkan kebutuhan-kebutuhan fungsionalitas dalam aplikasi. Gambar 4.3 menampilkan use case aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.



Gambar 4.3 Use Case Diagram

4.3.2 Use case scenario

Use case scenario menjelaskan spesifikasi masing-masing use case pada use case diagram di Gambar 4.2. Tabel 4.3 sampai dengan 4.7 menampilkan use case scenario aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

4.3.2.1 Use case scenario lihat peta [SRS_GEO_001]

Tabel 4.3 menunjukkan use case scenario lihat peta, pada halaman ini akan ditampilkan *marker-marker* laporan yang dibuat oleh pengguna. Untuk memudahkan identifikasi jenis laporan, *marker* diwarnai dengan warna yang berbeda. Warna merah untuk laporan hama, warna hijau untuk laporan hama yang sudah diperbarui (tanaman sudah diobati) dan warna biru untuk laporan berkala.

Tabel 4.3 Use Case Scenario Lihat Peta

Aliran Aktifitas untuk Use Case Lihat Peta	
Tujuan	Aktor dapat melihat peta yang berisi <i>marker-marker</i> laporan kondisi pertumbuhan dan produksi tanaman yang telah dibuat oleh aktor
Aktor	Mandor Pemeliharaan/Sinder/Manajer



Kondisi Awal	Aktor berhasil login ke dalam aplikasi
Alur Utama	<ol style="list-style-type: none"> 1. Aplikasi menampilkan <i>progress dialog</i> sedang memuat data 2. Aplikasi memuat peta berisi <i>marker</i> laporan hama dan berkala yang telah dibuat oleh aktor
Alur Alternatif	2.2 Jika tidak terdapat akses internet, data peta tidak akan muncul secara akurat dan akan menampilkan pesan <i>error</i>
Kondisi Akhir	Peta beserta <i>marker</i> laporan kondisi hama dan laporan berkala

4.3.2.2 Use case scenario buat laporan hama [SRS-GEO-002]

Tabel 4.4 menunjukkan *use case scenario* buat laporan hama, pada halaman ini akan ditampilkan *form* pembuatan laporan hama. Form tersebut termasuk *upload* foto yang diperlukan untuk proses *geotagging*.

Tabel 4.4 Use Case Scenario Buat Laporan Hama

Aliran Aktifitas untuk Use Case Buat Laporan Hama	
Tujuan	Aktor dapat membuat laporan mengenai tanaman yang terkena hama atau penyakit
Aktor	Sinder/Manajer
Kondisi Awal	Aktor berada pada halaman utama
Alur Utama	<ol style="list-style-type: none"> 1. Aktor memilih menu buat laporan hama 2. Aplikasi menampilkan <i>form</i> laporan hama 3. Aktor mengunggah foto tanaman dari mode langsung dari kamera atau galeri 4. Aplikasi menampilkan informasi <i>latitude</i> dan <i>longitude</i> 5. Aktor mengisi <i>form</i> detail laporan hama 6. Aplikasi mengunggah data laporan hama ke <i>database server</i> 7. Aplikasi memuat halaman utama
Alur Alternatif	2.1 Jika data pengisian laporan tidak sesuai, aplikasi akan menampilkan pesan <i>error</i>
Kondisi Akhir	Laporan mengenai tanaman yang terkena hama atau penyakit berhasil dibuat

4.3.2.3 Use case scenario buat laporan berkala [SRS-GEO-003]

Tabel 4.5 menunjukkan *use case scenario* buat laporan berkala, pada halaman ini akan ditampilkan *form* pembuatan laporan berkala. Form tersebut termasuk *upload* foto yang diperlukan untuk proses geotagging.

Tabel 4.5 Use Case Scenario Buat Laporan Berkala

Aliran Aktifitas untuk Use Case Buat Laporan Berkala	
Tujuan	Aktor dapat membuat laporan tanaman berkala
Aktor	Mandor Pemeliharaan
Kondisi Awal	Aktor berada pada halaman utama
Alur Utama	<ol style="list-style-type: none"> 1. Aktor memilih menu buat laporan berkala 2. Aplikasi mengarahkan aktor untuk mengunggah foto 3. Aktor mengunggah foto tanaman dari mode langsung dari kamera atau galeri 4. Aplikasi menampilkan <i>form</i> pengisian pembuatan laporan berkala 5. Aktor mengisi <i>form</i> detail laporan berkala 6. Aplikasi mengunggah data laporan berkala ke <i>database server</i> 7. Aplikasi memuat halaman utama
Alur Alternatif	2.1 Jika data pengisian laporan tidak sesuai, aplikasi akan menampilkan pesan <i>error</i>
Kondisi Akhir	Laporan berkala kondisi pertumbuhan dan produksi tanaman berhasil dibuat

4.3.2.4 Use case scenario lihat detail laporan [SRS-GEO-004]

Tabel 4.7 menunjukkan *use case scenario* lihat detail laporan, pada halaman ini akan ditampilkan detail laporan yang telah dibuat oleh pengguna.

Tabel 4.6 Use Case Scenario Lihat Detail Laporan

Aliran Aktifitas untuk Use Case Lihat Detail Laporan	
Tujuan	Aktor dapat melihat detail laporan yang telah dibuat oleh pengguna
Aktor	Manajer/Sinder/Mandor Pemeliharaan
Kondisi Awal	Aplikasi menampilkan <i>list view</i> laporan yang dibuat oleh pengguna

Alur Utama	<ol style="list-style-type: none"> 1. Aktor memilih laporan yang akan ditampilkan detailnya 2. Aplikasi menampilkan detail laporan
Alur Alternatif	-
Kondisi Akhir	Detail laporan berhasil ditampilkan

4.3.2.5 Use case scenario update laporan hama [SRS-GEO-005]

Tabel 4.6 menunjukkan *use case scenario* update laporan hama, hanya pengguna dengan *role* mandor pemeliharaan yang dapat mengakses halaman ini. *Update* laporan hama dilakukan ketika tanaman sudah diobati atau dilakukan pemeliharaan.

Tabel 4.7 Use Case Scenario Update Laporan Hama

Aliran Aktifitas untuk <i>Use Case</i> Update Laporan Hama	
Tujuan	Aktor dapat memperbarui status tanaman yang sudah diobati
Aktor	Mandor Pemeliharaan
Kondisi Awal	Aktor berada di halaman utama aplikasi
Alur Utama	<ol style="list-style-type: none"> 1. Aktor memilih <i>marker</i> yang menunjukkan laporan tanaman yang terkena hama atau penyakit 2. Aplikasi menampilkan detail laporan tanaman yang terkena hama atau penyakit 3. Aktor mengunggah foto tanaman yang sudah diobati dan memperbarui status tanaman 4. Aplikasi mengunggah data <i>update</i> laporan hama tanaman ke <i>database server</i>
Alur Alternatif	3.1 Jika data <i>update</i> laporan hama tidak sesuai, aplikasi akan menampilkan pesan <i>error</i>
Kondisi Akhir	Laporan hama tanaman berhasil diperbarui

4.4 Analisis kebutuhan non fungsional

Analisis kebutuhan non fungsional digunakan untuk menentukan batasan atau kualitas dari aplikasi. Tabel 4.8 menunjukkan kebutuhan non fungsional aplikasi mobile *geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

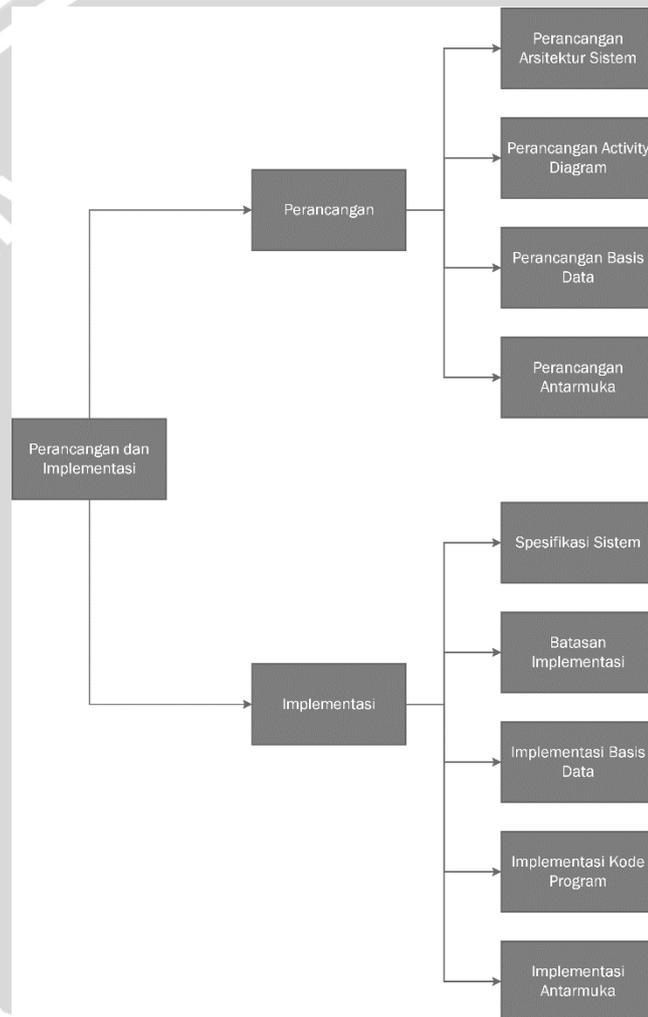
Tabel 4.8 Kebutuhan Non Fungsional

Kode	Parameter	Deskripsi
SRS_GEONF_001	<i>Usability</i>	Aplikasi dapat menyediakan sistem pelaporan yang mudah digunakan oleh pengguna
SRS_GEONF_002	<i>Performance</i>	Aplikasi dapat memuat data dengan waktu eksekusi kurang dari 3 detik



BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini, dijelaskan tentang tahap perancangan dan implementasi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android. Tahapan penjelasan perancangan meliputi perancangan arsitektur sistem, perancangan diagram activity, perancangan basis data dan perancangan antarmuka. Sedangkan tahapan implementasi meliputi spesifikasi sistem, batasan implementasi, implementasi basis data, implementasi kode program dan implementasi antarmuka. Gambar 5.1 menjelaskan tentang tahapan implementasi.

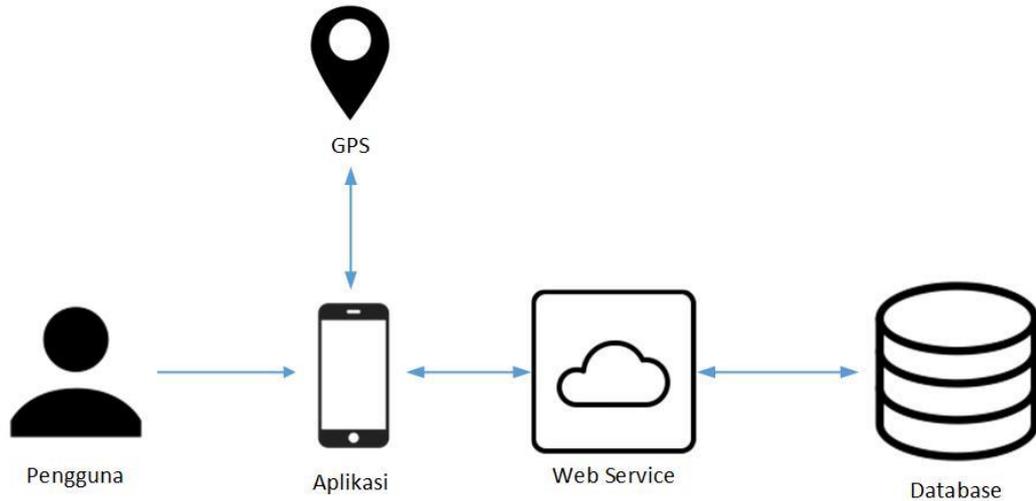


Gambar 5.1 Tahapan Perancangan dan Implementasi

5.1 Perancangan aplikasi perangkat bergerak

Perancangan aplikasi perangkat bergerak terdiri dari beberapa tahapan. Tahapan tersebut meliputi perancangan arsitektur sistem, *activity diagram*, perancangan basis data serta perancangan navigasi dan antarmuka.

5.1.1 Perancangan arsitektur sistem



Gambar 5.2 Rancangan Arsitektur Aplikasi

Gambar 5.2 di atas menjelaskan rancangan arsitektur aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android. Aplikasi melakukan komunikasi ke *database* menggunakan *web service*. Guna mendapatkan informasi lokasi (*geotagging*) tanaman saat pelaporan, aplikasi terhubung dengan satelit GPS melalui perangkat GPS yang ada pada *smartphone*. Data-data pelaporan disimpan di *database*, ditampilkan dalam aplikasi berupa data JSON menggunakan *web service*.

5.1.2 Perancangan *activity diagram*

Activity diagram digunakan untuk memodelkan aktifitas yang terjadi dalam aplikasi. Di bawah ini daftar *activity diagram* aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

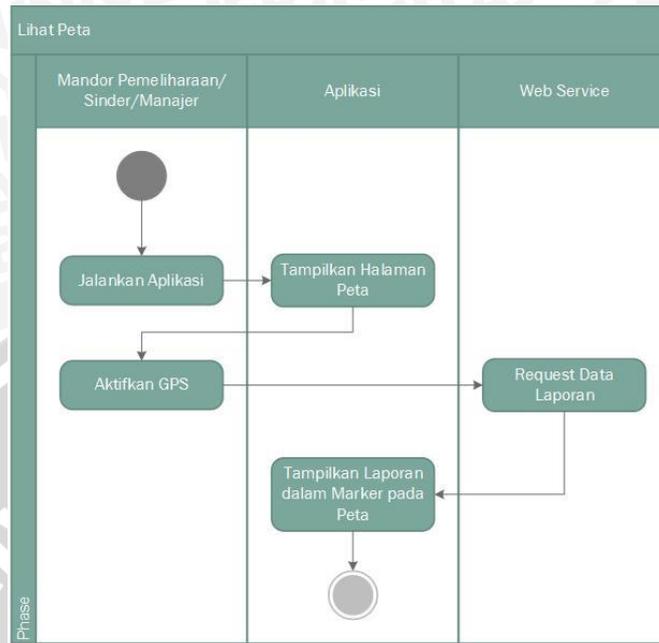
5.1.2.1 Perancangan *activity diagram* lihat peta [SRS_GEO_001]

Gambar 5.3 menunjukkan *Activity Diagram* Lihat Peta, pengguna yang terbagi menjadi Mandor Pemeliharaan, Sinder dan Manajer akan diarahkan langsung pada halaman peta. Halaman ini merupakan halaman utama aplikasi. Pengguna akan diarahkan untuk mengaktifkan GPS, setelah itu aplikasi akan meminta data-data laporan dari *database* melalui *web service*. Data-data tersebut ditampilkan pada peta dalam bentuk marker.

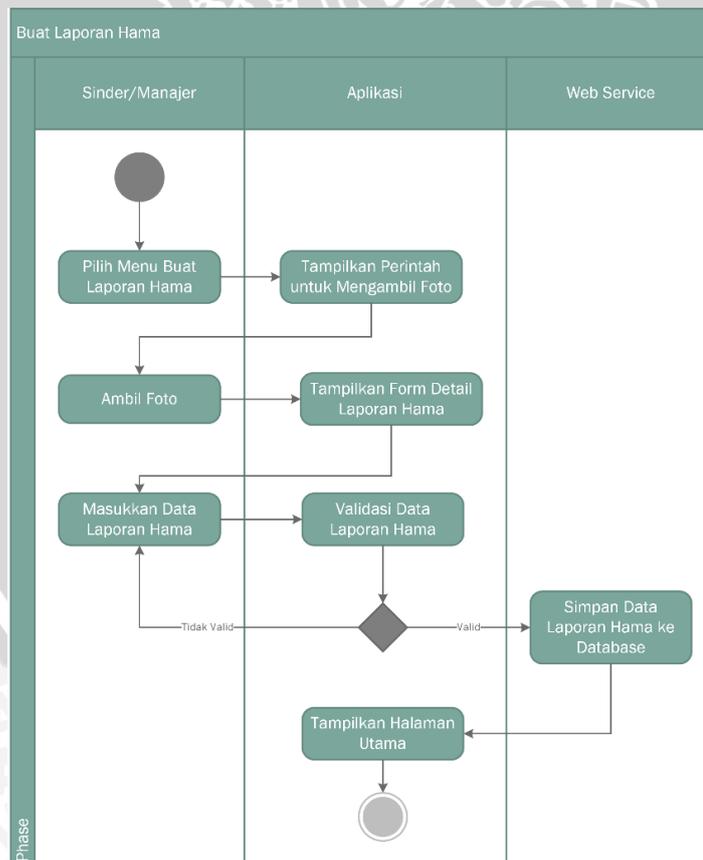
5.1.2.2 Perancangan *activity diagram* buat laporan hama [SRS_GEO_002]

Gambar 5.4 menunjukkan *Activity Diagram* Buat Laporan Hama, berada pada halaman utama aplikasi Sinder atau Manajer memilih menu Buat Laporan Hama. Kemudian aplikasi akan mengarahkan Sinder atau Manajer untuk mengambil foto. Setelah mengambil foto, Sinder atau Manajer diharuskan mengisi detail laporan hama. Data laporan hama akan divalidasi terlebih dahulu sebelum disimpan ke

dalam *database* melalui *web service*. Jika validasi gagal, aplikasi akan menampilkan *error* dan menyuruh untuk melengkapi data.



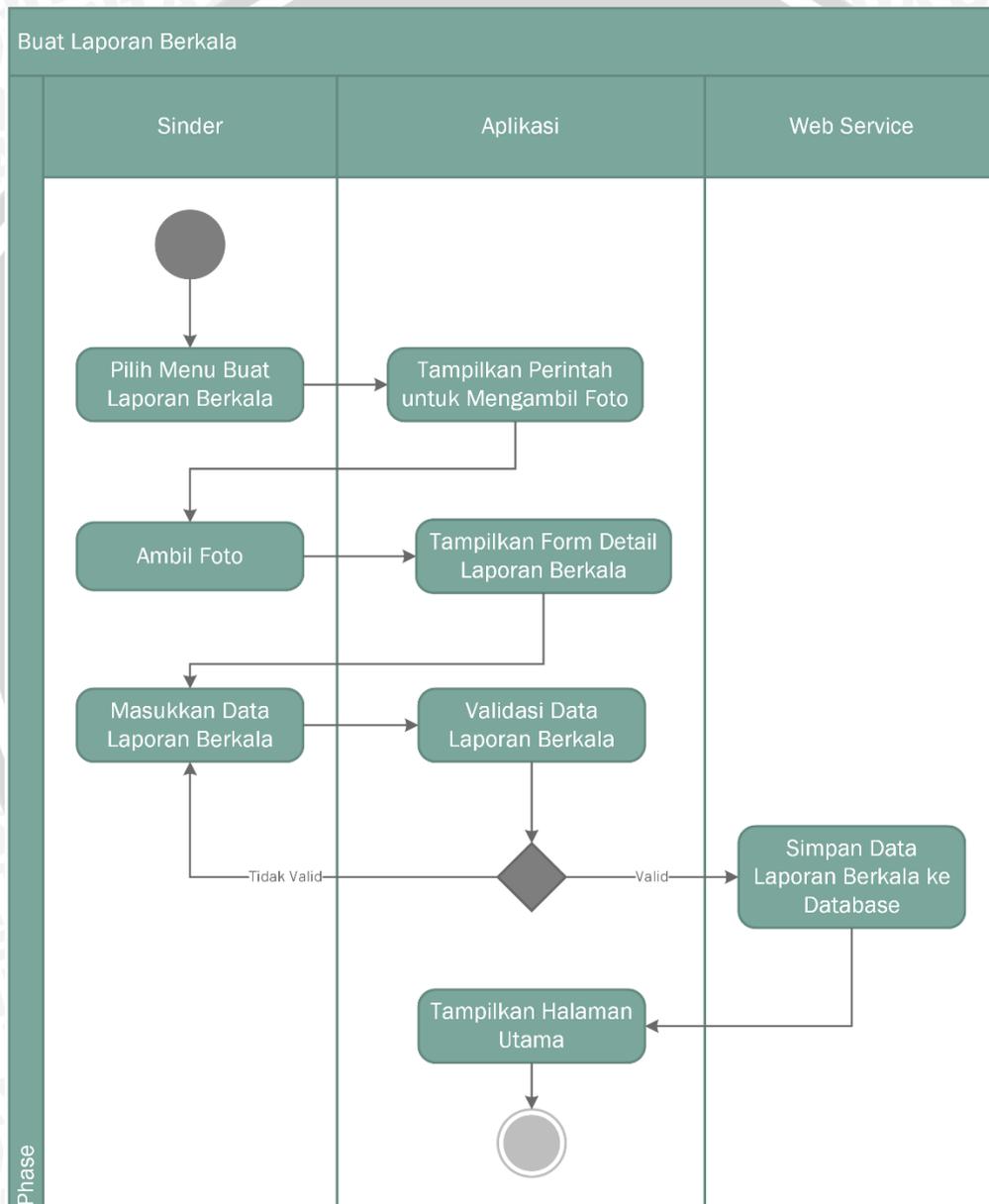
Gambar 5.3 Activity Diagram Lihat Peta



Gambar 5.4 Activity Diagram Buat Laporan Hama

5.1.2.3 Perancangan activity diagram buat laporan berkala [SRS_GEO_003]

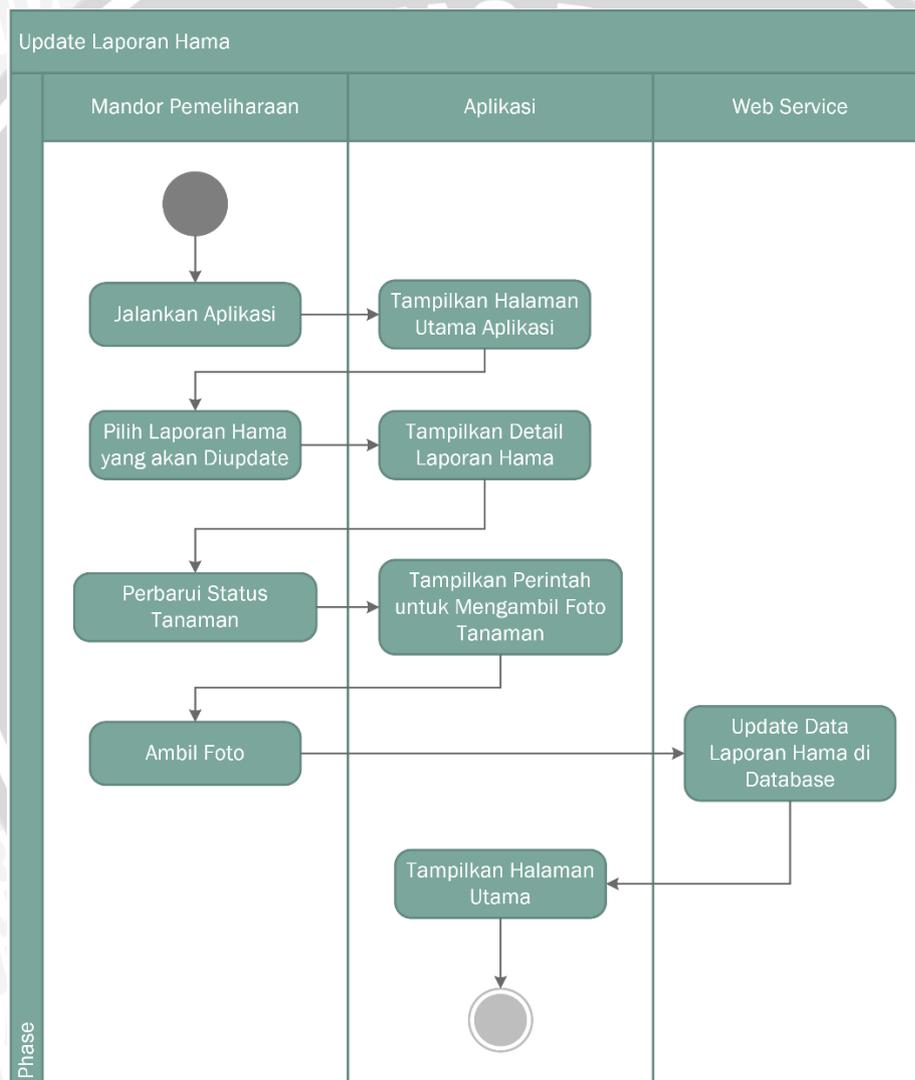
Gambar 5.5 menunjukkan *Activity Diagram* Buat Laporan Berkala, berada pada halaman utama aplikasi Sinder memilih menu Buat Laporan Berkala. Kemudian aplikasi akan mengarahkan Sinder untuk mengambil foto. Setelah mengambil foto, Sinder diharuskan mengisi detail laporan berkala. Data laporan berkala akan divalidasi terlebih dahulu sebelum disimpan ke dalam *database* melalui *web service*. Jika validasi gagal, aplikasi akan menampilkan *error* dan menyuruh untuk melengkapi data.



Gambar 5.5 Activity Diagram Buat Laporan Berkala

5.1.2.4 Perancangan activity diagram update laporan hama [SRS_GEO_005]

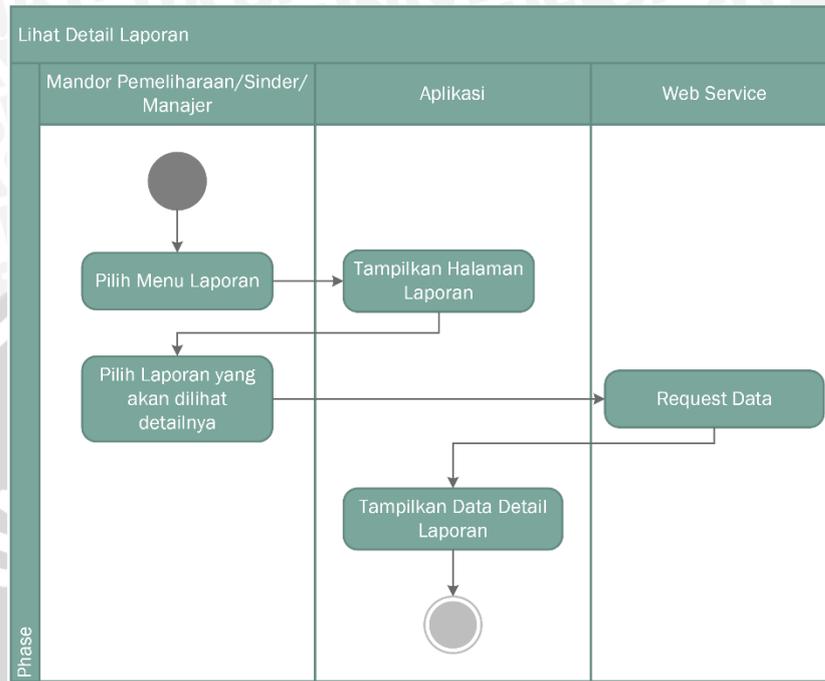
Gambar 5.6 menunjukkan *Activity Diagram* Update Laporan Hama, untuk bisa meng-*update* laporan hama yang masuk menjadi sudah tertangani Mandor Pemeliharaan harus membuka halaman utama aplikasi. Laporan hama ditampilkan dalam peta di halaman utama aplikasi dalam bentuk *marker*. Mandor Pemeliharaan memilih *marker* (laporan) yang akan diperbarui, aplikasi akan menampilkan detail laporan hama. Mandor Pemeliharaan memilih tombol Perbarui Status Tanaman, aplikasi akan mengarahkan untuk mengunggah foto bukti bahwa tanaman telah diobati. Selanjutnya data status tanaman yang terkena hama akan diperbarui di *database* melalui *web service*.



Gambar 5.6 Activity Diagram Update Laporan Hama

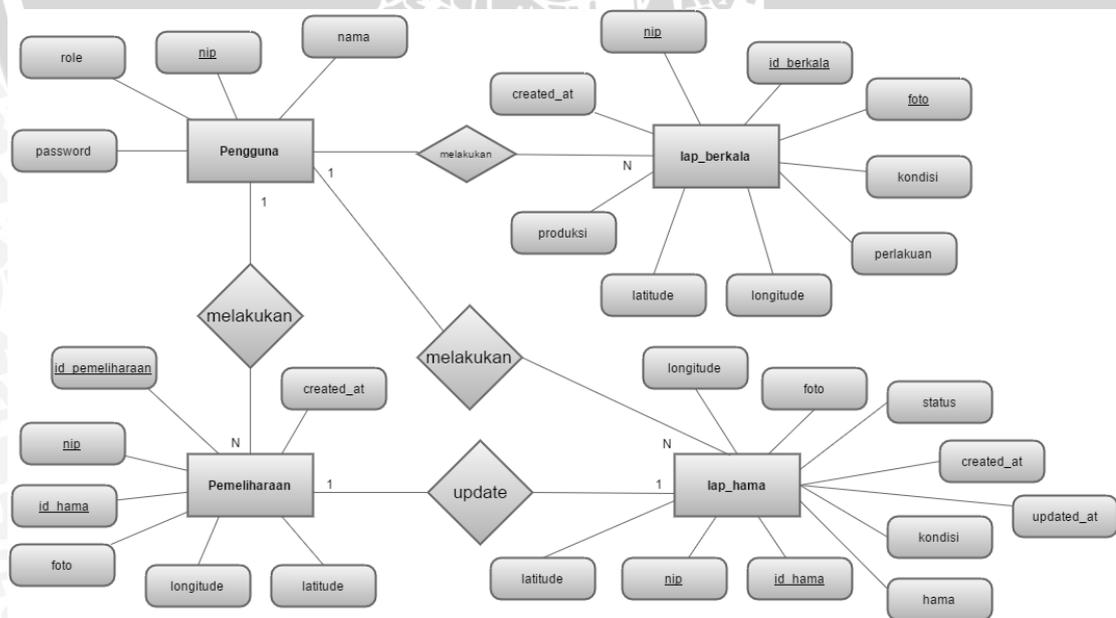
5.1.2.5 Perancangan activity diagram lihat detail laporan [SRS_GEO_004]

Gambar 5.7 menunjukkan *Activity Diagram* Lihat Detail Laporan, Mandor Pemeliharaan/Sinder/Manajer memilih laporan yang akan ditampilkan detailnya. Setelah itu, aplikasi akan menampilkan detail laporan.



Gambar 5.7 Activity Diagram Lihat Detail Laporan

5.1.3 Perancangan basis data



Gambar 5.8 Entity Relationship Diagram

Perancangan basis data menggunakan ERD (*Entity Relationship Diagram*) untuk menjelaskan data apa saja yang disimpan dalam *web server* aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android. ERD dirancang berdasarkan pada *use case* sistem pada bab 4 sub bab 4.3.1. Basis data dalam sistem ini menggunakan MySQL. Terdapat 4 entitas dalam sistem, yaitu pengguna, lap_hama, lap_berkala dan pemeliharaan seperti terlihat pada Gambar 5.8.

Tabel 5.1 menunjukkan entitas *user* yang berisi data pengguna atau aktor yang terlibat dalam aplikasi.

Tabel 5.1 Struktur Tabel Pengguna

No.	Nama Field	Tipe Data	Keterangan
1.	nip	Varchar	Nomor induk pegawai yang digunakan <i>user</i> untuk login juga sebagai <i>primary key</i> entitas pengguna
2.	password	Varchar	Password yang digunakan <i>user</i> untuk login
3.	nama	Varchar	Nama <i>user</i>
4.	role	Varchar	Hak akses <i>user</i> dalam aplikasi yang sesuai dengan jabatan pegawai pada perusahaan

Tabel 5.2 menunjukkan entitas lap_berkala yang berisi data laporan tanaman secara berkala yang dibuat oleh pengguna.

Tabel 5.2 Struktur Tabel Lap_Berkala

No.	Nama Field	Tipe Data	Keterangan
1.	id_berkala	Integer	<i>Primary key</i> untuk entitas lap_berkala
2.	nip	Integer	<i>Foreign key</i> untuk nip yang berasal dari tabel user
3.	foto	Text	Lokasi url file foto tanaman
4.	latitude	Double	Informasi latitude yang diambil dari foto atau posisi pengguna
5.	longitude	Double	Informasi longitude yang diambil dari foto atau posisi pengguna
6.	wilayah	Varchar	Daerah yang mewakili lokasi dari cakupan luas tertentu pada perusahaan
7.	kondisi	Varchar	Kondisi tanaman yang akan dilaporkan
8.	perlakuan	Varchar	Perlakuan yang sedang diterapkan pada tanaman

9.	produksi	Integer	Jumlah prediksi produksi tanaman
10.	created_at	Datetime	Waktu pembuatan laporan

Tabel 5.3 menunjukkan entitas lap_hama yang berisi data laporan hama tanaman yang dibuat oleh pengguna.

Tabel 5.3 Struktur Tabel Lap_Hama

No.	Nama Field	Tipe Data	Keterangan
1.	id_hama	Integer	<i>Primary key</i> untuk entitas lap_hama
2.	nip	Integer	<i>Foreign key</i> untuk nip yang berasal dari tabel user
3.	foto	Text	Lokasi url file foto tanaman
4.	latitude	Double	Informasi latitude yang diambil dari foto atau posisi pengguna
5.	longitude	Double	Informasi longitude yang diambil dari foto atau posisi pengguna
6.	wilayah	Varchar	Daerah yang mewakili lokasi dari cakupan luas tertentu pada perusahaan
7.	hama	Varchar	Jenis hama atau penyakit yang menyerang tanaman
8.	kondisi	Varchar	Kondisi tanaman yang akan dilaporkan
9.	status	Varchar	Status tanaman sudah tertangani atau belum tertangani
10.	created_at	Datetime	Waktu pembuatan laporan
11.	updated_at	Datetime	Waktu ketika tanaman yang terkena hama diobati

Tabel 5.4 menunjukkan entitas pemeliharaan yang berisi data pemeliharaan tanaman yang terkena hama atau penyakit.

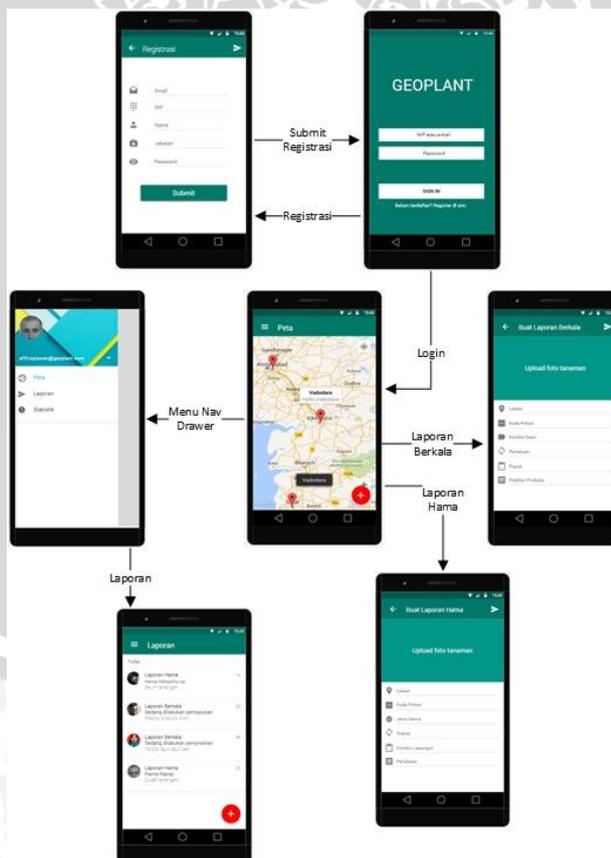
Tabel 5.4 Struktur Tabel Pemeliharaan

No.	Nama Field	Tipe Data	Keterangan
1.	id_pemeliharaan	Integer	<i>Primary key</i> untuk entitas pemeliharaan
2.	nip	Integer	<i>Foreign key</i> untuk nip yang berasal dari tabel user
3.	id_hama	Integer	<i>Foreign key</i> untuk id_hama yang berasal dari tabel lap_hama

4.	foto	Text	Lokasi url file foto tanaman
5.	latitude	Double	Informasi latitude yang diambil dari foto atau posisi pengguna
6.	longitude	Double	Informasi longitude yang diambil dari foto atau posisi pengguna
7.	detail	Varchar	Keterangan pemeliharaan yang dilakukan oleh pengguna dengan <i>role</i> Mandor Pemeliharaan
8.	created_at	Datetime	Waktu pelaksanaan pemeliharaan

5.1.4 Perancangan antarmuka

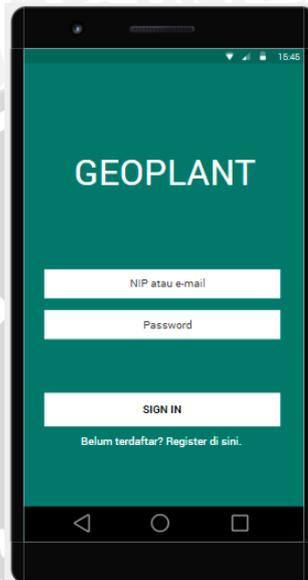
Pada tahap ini dilakukan perancangan antarmuka dan *screen flow* aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android. Antarmuka yang dibuat harus mudah dipahami dan digunakan oleh pengguna. Gambar 5.9 di bawah ini menunjukkan *screen flow* aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android. *Screen flow* merupakan alur perpindahan halaman pada aplikasi ketika pengguna memilih suatu aksi tertentu.



Gambar 5.9 Screen Flow Aplikasi

5.1.4.1 Perancangan antarmuka login

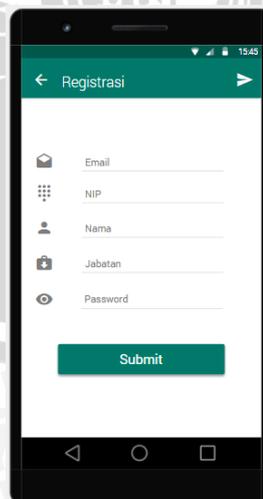
Antarmuka login berisi *form* yang dibutuhkan oleh pengguna agar bisa masuk ke dalam aplikasi. Halaman ini muncul ketika aplikasi belum pernah dibuka sebelumnya atau ketika pengguna melakukan *log out* dan ingin masuk ke dalam aplikasi. Gambar 5.10 menunjukkan perancangan antarmuka halaman login.



Gambar 5.10 Perancangan Antarmuka Login

5.1.4.2 Perancangan antarmuka registrasi

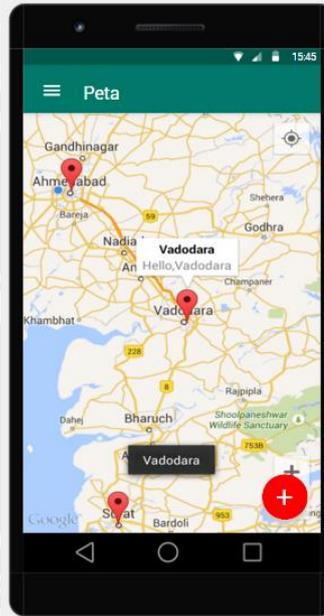
Antarmuka registrasi digunakan ketika pengguna belum terdaftar dalam sistem dan ingin masuk ke dalam aplikasi. Pengguna diminta untuk memasukkan Nomor Induk Pegawai, *e-mail*, nama, jabatan dan *password*. Gambar 5.11 menunjukkan perancangan antarmuka halaman register.



Gambar 5.11 Perancangan Antarmuka Registrasi

5.1.4.3 Perancangan antarmuka peta

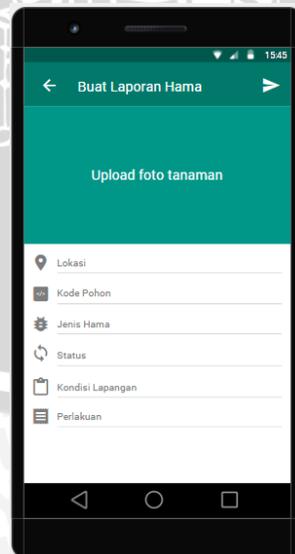
Antarmuka peta menampilkan marker berisi informasi mengenai laporan hama maupun laporan berkala. Setelah melakukan *login*, pengguna diarahkan pada halaman peta. Gambar 5.12 menunjukkan perancangan antarmuka halaman peta.



Gambar 5.12 Perancangan Antarmuka Peta

5.1.4.4 Perancangan antarmuka buat laporan hama

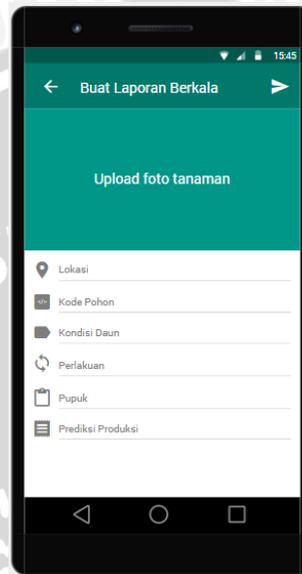
Antarmuka buat laporan hama digunakan untuk menampilkan *form* yang harus diisi oleh pengguna ketika akan melakukan pelaporan hama tanaman di lokasi tertentu. Gambar 5.13 menunjukkan perancangan halaman buat laporan hama.



Gambar 5.13 Perancangan Antarmuka Buat Laporan Hama

5.1.4.5 Perancangan antarmuka buat laporan berkala

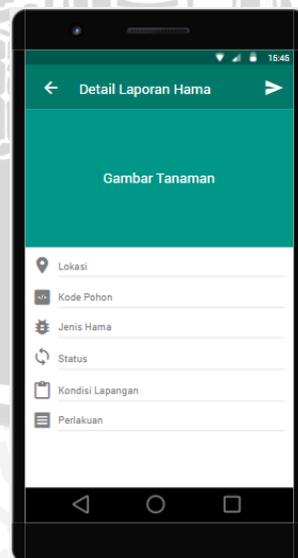
Antarmuka buat laporan berkala digunakan untuk menampilkan *form* yang harus diisi oleh pengguna ketika akan melakukan pelaporan tanaman secara berkala. Pelaporan ini meliputi pemeliharaan tanaman, kondisi tanaman dan prediksi tanaman. Gambar 5.14 menunjukkan perancangan halaman buat laporan berkala.



Gambar 5.14 Perancangan Antarmuka Buat Laporan Berkala

5.1.4.6 Perancangan antarmuka detail laporan

Antarmuka detail laporan digunakan untuk menampilkan detail tiap-tiap laporan yang telah dibuat oleh pengguna. Gambar 5.15 menunjukkan antarmuka detail laporan.



Gambar 5.15 Perancangan Antarmuka Detail Laporan

5.2 Implementasi

Proses implementasi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android. meliputi spesifikasi sistem, batasan implementasi, implementasi basis data, implementasi kode program dan implementasi antarmuka.

5.2.1 Spesifikasi sistem

Dalam upaya mewujudkan implementasi terhadap analisis dan perancangan aplikasi dibutuhkan suatu perangkat keras dan perangkat lunak. Standar dalam proses implementasi dapat diidentifikasi dari perangkat keras dan perangkat lunak yang digunakan.

5.2.1.1 Spesifikasi perangkat keras

Proses implementasi membutuhkan perangkat komputer dan *smartphone*. Tabel 5.5 menjelaskan tentang perangkat keras komputer yang digunakan dalam proses implementasi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

Tabel 5.5 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
<i>System Model</i>	Asus N46VJ
<i>Processor</i>	Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz (4 CPUs), ~2.5GHz
<i>Memory</i>	4096MB RAM
<i>Display</i>	Intel(R) HD Graphics 4000, NVIDIA GeForce GT 635M
<i>Platform</i>	Windows 10

Tabel 5.6 menjelaskan tentang spesifikasi perangkat keras *smartphone* yang digunakan dalam proses implementasi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

Tabel 5.6 Spesifikasi Perangkat Keras Smartphone

Nama Komponen	Spesifikasi
<i>System Model</i>	LG Google Nexus 5
<i>Processor</i>	Qualcomm MSM8974 Snapdragon 800
<i>Memory</i>	2GB RAM
<i>Display</i>	Adreno 330
<i>Platform</i>	Android Marshmallow 6.0.1

5.2.1.2 Spesifikasi perangkat lunak

Tabel 5.7 menjelaskan spesifikasi perangkat lunak pada komputer yang digunakan dalam proses implementasi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

Tabel 5.7 Spesifikasi Perangkat Lunak Komputer

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 10 Pro 64 bit
<i>Programming Language</i>	Java, PHP
<i>Programming Environment</i>	XAMPP 5.6.3
<i>Editor</i>	Android Studio, Atom 1.8.0

Tabel 5.8 menjelaskan spesifikasi perangkat lunak pada *smartphone* yang digunakan dalam proses implementasi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

Tabel 5.8 Spesifikasi Perangkat Lunak Smartphone

Nama Komponen	Spesifikasi
<i>Platform</i>	Android Marshmallow 6.0.1

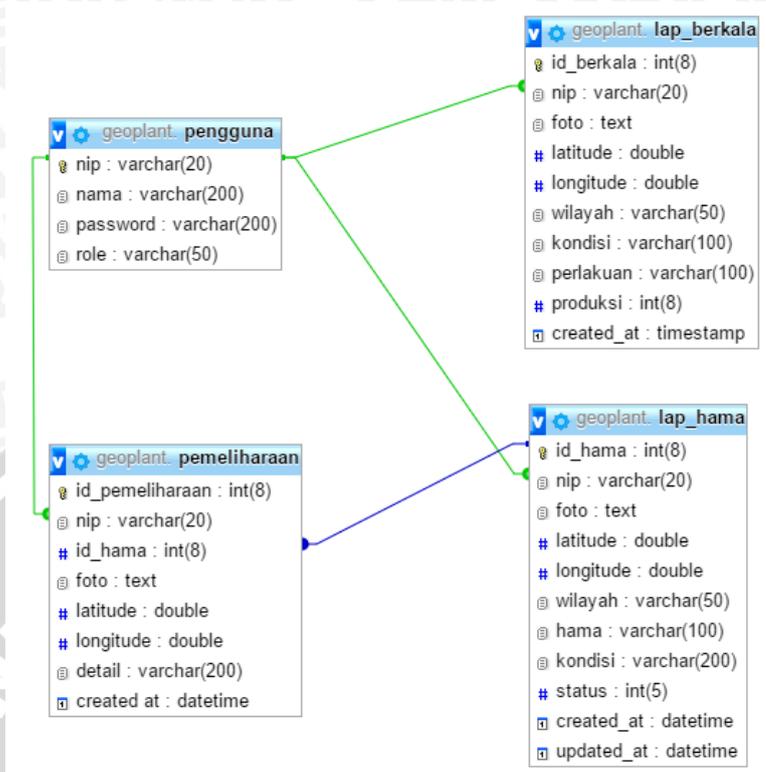
5.2.2 Batasan implementasi

Batasan implementasi diperlukan agar implementasi tidak keluar dari tujuan utama penelitian. Adapun batasan implementasi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android meliputi:

1. Aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android dirancang untuk dapat berjalan pada *smartphone* dengan sistem operasi Android 4.4.2 (SDK 19).
2. Pengembangan aplikasi menggunakan metode pendekatan *native*.
3. Aplikasi hanya dapat berjalan jika terdapat koneksi internet, untuk pencarian lokasi yang akurat GPS pada *smartphone* harus diaktifkan.

5.2.3 Implementasi basis data

Proses implementasi basis data didasarkan pada tahap perancangan *Entity Relationship Diagram* (ERD) pada Gambar 5.8. Basis data yang digunakan adalah basis data MySQL. Gambar 5.16 menunjukkan hasil implementasi basis data aplikasi.



Gambar 5.16 Diagram Fisik Basis Data

5.2.4 Implementasi kode program

Implementasi kode program menjelaskan implementasi algoritma ke dalam kode program dalam bahasa Java. Implementasi kode program disesuaikan dengan kebutuhan sistem yang telah dijelaskan pada bab 4 subbab 4.3 analisis kebutuhan fungsional. Pada tahap ini akan dicantumkan beberapa kode program pada fitur penting aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

5.2.4.1 Implementasi kode program fungsi readGeotagImage()

Fungsi readGeotagImage() digunakan untuk membaca informasi *latitude* dan *longitude* pada foto yang ada di galeri *smartphone*. Kode program ini digunakan ketika pengguna membuat laporan dan memilih opsi pengambilan foto dari galeri. Hasil dari kode program ini adalah informasi *latitude* dan *longitude* dari foto yang ditampilkan pada *form* lokasi di menu laporan hama dan laporan berkala. Tabel 5.9 menunjukkan kode program fungsi readGeotagImage().

Tabel 5.9 Kode Program Fungsi readGeotagImage()

File	helper/Geotagging.java
No	Kode Program
1	public Location readGeotagImage(String imagePath) {
2	Location loc = new Location("");



3	try {
4	ExifInterface exif = new ExifInterface(imagePath);
5	float [] latlong = new float[2] ;
6	if(exif.getLatLong(latlong)){
7	loc.setLatitude(latlong[0]);
8	loc.setLongitude(latlong[1]);
9	}
10	} catch (IOException e) {
11	e.printStackTrace();
12	}
13	return loc;
14	}
15	
16	@SuppressWarnings("NewApi")
17	public static String getRealPathFromURI(Context context, Uri uri) {
18	String filePath = "";
19	String wholeID = DocumentsContract.getDocumentId(uri);
20	
21	// Split at colon, use second item in the array
22	String id = wholeID.split(":")[1];
23	
24	String[] column = { MediaStore.Images.Media.DATA };
25	
26	// where id is equal to
27	String sel = MediaStore.Images.Media._ID + "=?";
28	
29	Cursor cursor = context.getContentResolver().query(MediaStore.Images.Media.EXTERNAL_CON TENT_URI,
30	column, sel, new String[]{ id }, null);
31	
32	int columnIndex = cursor.getColumnIndex(column[0]);
33	
34	if (cursor.moveToFirst()) {

35	<code>filePath = cursor.getString(columnIndex);</code>
36	<code>}</code>
37	<code>cursor.close();</code>
38	<code>return filePath;</code>
39	<code>}</code>

Adapun penjelasan kode program pada Tabel 5.9 fungsi `readGeotagImage()` sebagai berikut:

1. Baris 2: inisiasi objek *location* yang nantinya digunakan untuk memanggil latitude dan longitude.
2. Baris 4: inisiasi kelas `ExifInterface` Android.
3. Baris 5: inisiasi array float untuk menyimpan latitude dan longitude.
4. Baris 6-8: jika objek `exif` dari kelas `ExifInterface` mendapatkan latitude dan longitude, objek `loc` akan menampilkan latitude dan longitude di kelas `CreateLaporan.java`.
5. Baris 17-38: fungsi untuk mendapatkan Uri file foto yang dibutuhkan oleh fungsi `readGeotagImage` untuk mendapatkan latitude dan longitude.

5.2.4.2 Implementasi kode program fungsi `insertHama()`

Fungsi `insertHama()` digunakan untuk mengunggah data laporan hama ke *database*. Kode program ini dieksekusi ketika pengguna menekan *icon send* untuk mengunggah data laporan hama. Tabel 5.10 menunjukkan kode program fungsi `insertHama()`.

Tabel 5.10 Kode Program Fungsi `insertHama()`

File	<code>activity/LaporanHama.java</code>
No	Kode Program
1	<code>private class insertHama extends AsyncTask<Void,Void,String> {</code>
2	<code>ProgressDialog loading;</code>
3	<code>final String image = getStringImage(bitmap);</code>
4	<code>final String latitude = txtLatitude.getText().toString().trim();</code>
5	<code>final String longitude = txtLongitude.getText().toString().trim();</code>
6	<code>final String wilayah = txtWilayah.getText().toString().trim();</code>
7	<code>final String hama = txtHama.getText().toString().trim();</code>
8	<code>final String kondisi = txtKondisi.getText().toString().trim();</code>
9	



10	@Override
11	protected String doInBackground(Void... params) {
12	RequestHandler rh = new RequestHandler();
13	HashMap<String,String> data = new HashMap<>();
14	data.put (IMAGE_VAR, image);
15	data.put (LATITUDE_VAR, latitude);
16	data.put (LONGITUDE_VAR, longitude);
17	data.put (WILAYAH_VAR, wilayah);
18	data.put (HAMA_VAR, hama);
19	data.put (KONDISI_VAR, kondisi);
20	String result = rh.sendPostRequest (INSERT_URL, data);
21	return result;
22	}
23	
24	@Override
25	protected void onPreExecute() {
26	super.onPreExecute();
27	loading = ProgressDialog.show (LaporanHama.this, "Uploading Data", "Please wait..", false, false);
28	}
29	
30	@Override
31	protected void onPostExecute (String s) {
32	super.onPostExecute (s);
33	loading.dismiss ();
34	Toast.makeText (LaporanHama.this, s, Toast.LENGTH_LONG).show ();
35	}
36	}

Adapun penjelasan kode program pada Tabel 5.10 fungsi insertHama() sebagai berikut:

1. Baris 1-8: inisiasi kelas dan variabel yang ada pada *form* laporan hama.
2. Baris 11-21: *method* doInBackground untuk menyimpan nilai variabel pada akses *web service* sebelum disimpan di *database*.



3. Baris 24-28: *method* onPreExecute untuk menampilkan *progress dialog* proses *upload* data ke dalam *database* melalui *web service*.
4. Baris 30-35: *method* onPostExecute untuk menampilkan pesan terkait proses *upload* data.

5.2.4.3 Implementasi kode program fungsi displayMarker()

Fungsi displayMarker() digunakan untuk menampilkan laporan yang telah dibuat oleh pengguna ke dalam peta dalam bentuk *marker*. Kode program ini dieksekusi ketika pengguna memilih menu *map* dalam *navigation drawer*. Tabel 5.11 menunjukkan kode program fungsi displayMarker().

Tabel 5.11 Kode Program displayMarker()

File	fragment/MapFragment.java
No	Kode Program
1	private void displayMarkerHama() {
2	ArrayList<HashMap<String, String>> location = new ArrayList<HashMap<String, String>>();
3	try {
4	JSONObject jsonObject = new JSONObject(JSON_STRING);
5	JSONArray result = jsonObject.getJSONArray(JSON_ARRAY);
6	
7	for (int i = 0; i < result.length(); i++) {
8	JSONObject jo = result.getJSONObject(i);
9	String hama = jo.getString(HAMA_VAR);
10	String latitude = jo.getString(LATITUDE_VAR);
11	String longitude = jo.getString(LONGITUDE_VAR);
12	
13	HashMap<String, String> marker = new HashMap<>();
14	marker.put(HAMA_VAR, hama);
15	marker.put(LATITUDE_VAR, latitude);
16	marker.put(LONGITUDE_VAR, longitude);
17	location.add(marker);
18	}
19	} catch (JSONException e) {
20	e.printStackTrace();
21	}

22	
23	for (int i = 0; i < location.size(); i++) {
24	Latitude = Double.parseDouble(location.get(0).get(LATITUDE_VAR));
25	Longitude = Double.parseDouble(location.get(0).get(LONGITUDE_VAR));
26	LatLng coordinate = new LatLng(Latitude, Longitude);
27	gMap.animateCamera(CameraUpdateFactory.newLatLngZoom(coordinate, 14));
28	Latitude = Double.parseDouble(location.get(i).get(LATITUDE_VAR));
29	Longitude = Double.parseDouble(location.get(i).get(LONGITUDE_VAR));
30	Hama = location.get(i).get(HAMA_VAR);
31	MarkerOptions hamaMarker = new MarkerOptions().position(new LatLng(Latitude, Longitude))
32	.title(Hama);
33	gMap.addMarker(hamaMarker);
34	}
35	}

Adapun penjelasan kode program pada Tabel 5.11 fungsi displayMarker() sebagai berikut:

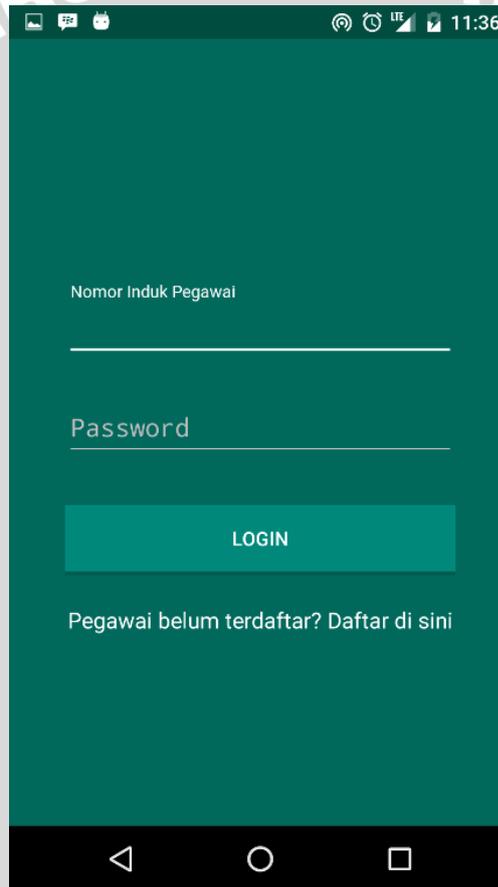
1. Baris 2: inisiasi *array list* untuk menyimpan *array* dari JSON.
2. Baris 4-5: inisiasi *JSONObject* untuk mengambil data *array* JSON.
3. Baris 7-11: perulangan untuk menyimpan *array* JSON ke dalam *string*.
4. Baris 13-17: inisiasi *HashMap* untuk menyimpan *array* JSON yang sudah disimpan dalam *string* ke objek *location*.
5. Baris 23-33: perulangan untuk menampilkan objek *location* yang berisi data *array* JSON dalam bentuk *marker* pada peta.

5.2.5 Implementasi antarmuka

Implementasi antarmuka didasarkan pada tahap perancangan di sub bab sebelumnya. Adapun implementasi antarmuka yang akan dijelaskan meliputi, antarmuka login, antarmuka register, antarmuka peta, antarmuka buat laporan hama, antarmuka buat laporan berkala.

5.2.5.1 Implementasi antarmuka login

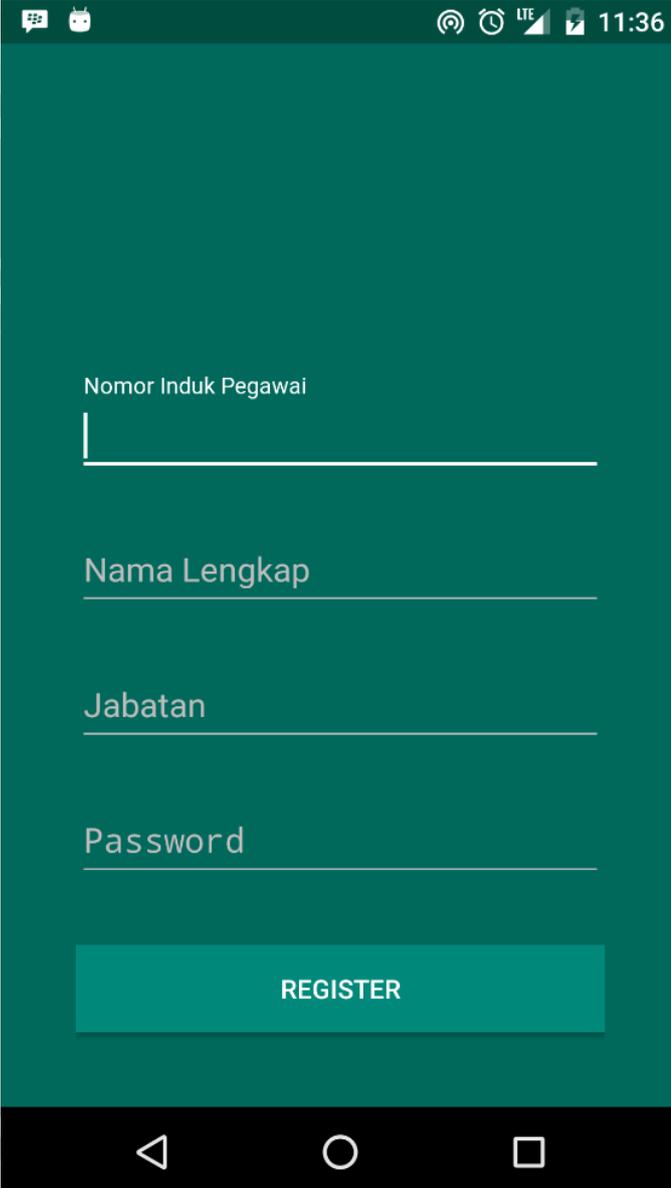
Antarmuka login didasarkan pada perancangan antarmuka pada Gambar 5.10. Antarmuka login berisi *form* yang dibutuhkan oleh pengguna agar bisa masuk ke dalam aplikasi. Antarmuka ini muncul ketika aplikasi belum pernah dibuka sebelumnya atau ketika pengguna melakukan *log out* dan ingin masuk ke dalam aplikasi. Gambar 5.17 menunjukkan implementasi antarmuka login.



Gambar 5.17 Implementasi Antarmuka Login

5.2.5.2 Implementasi antarmuka registrasi

Antarmuka registrasi didasarkan pada perancangan antarmuka registrasi pada Gambar 5.11. Halaman registrasi digunakan ketika pengguna belum terdaftar dalam sistem dan ingin masuk ke dalam aplikasi. Pengguna diminta untuk memasukkan Nomor Induk Pegawai, nama lengkap, jabatan dan *password*. Gambar 5.18 menunjukkan implementasi antarmuka registrasi.

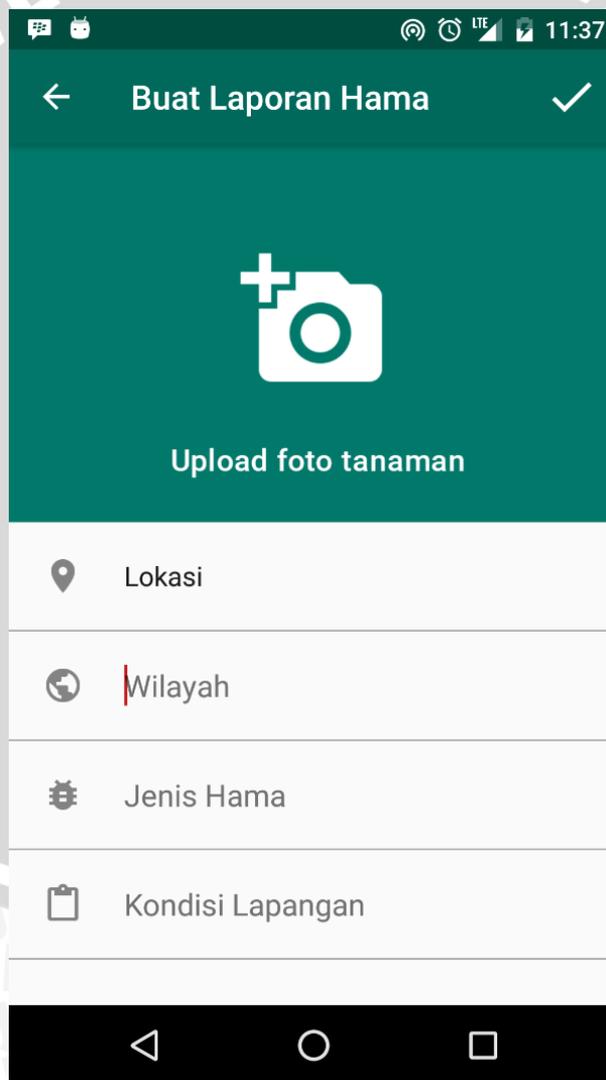


The image shows a mobile application registration screen with a dark teal background. At the top, there is a status bar with icons for messages, a cat, signal strength, LTE, battery, and the time 11:36. Below the status bar, there are four input fields, each with a label and a white underline: 'Nomor Induk Pegawai', 'Nama Lengkap', 'Jabatan', and 'Password'. At the bottom of the form is a large teal button with the text 'REGISTER' in white. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Gambar 5.18 Implementasi Antarmuka Registrasi

5.2.5.3 Implementasi antarmuka buat laporan hama

Antarmuka buat laporan hama didasarkan pada perancangan antarmuka buat laporan hama pada Gambar 5.13. Antarmuka buat laporan hama digunakan untuk menampilkan *form* yang harus diisi oleh pengguna ketika akan melakukan pelaporan hama tanaman di lokasi tertentu. Gambar kamera bertuliskan “*upload foto tanaman*” merupakan *imagebutton* untuk mengunggah foto yang akan diambil informasi geotagging-nya. Hasil pembacaan *geotagging* akan ditampilkan pada *text field* lokasi. *Form* wilayah digunakan untuk memasukkan data area bagian pada PTPN XII Pancursari. *Form* jenis hama digunakan untuk memasukkan jenis hama atau penyakit yang menyerang tanaman. *Form* kondisi lapangan digunakan untuk memasukkan kondisi tanaman yang terkena hama atau penyakit. Berdasarkan observasi penulis di PTPN XII Pancursari, kondisi yang diamati yaitu kondisi daun. Gambar 5.19 menunjukkan implementasi antarmuka laporan hama.



Gambar 5.19 Implementasi Antarmuka Buat Laporan Hama

5.2.5.4 Implementasi antarmuka buat laporan berkala

Antarmuka buat laporan berkala didasarkan pada perancangan antarmuka buat laporan berkala pada Gambar 5.14. Antarmuka buat laporan berkala digunakan untuk menampilkan *form* yang harus diisi oleh pengguna ketika akan melakukan pelaporan tanaman secara berkala. Pelaporan ini meliputi pemeliharaan tanaman, kondisi tanaman dan prediksi produksi tanaman. Gambar kamera bertuliskan “upload foto tanaman” merupakan *imagebutton* untuk mengunggah foto yang akan diambil informasi geotagging-nya. Hasil pembacaan *geotagging* akan ditampilkan pada *text field* lokasi. *Form* wilayah digunakan untuk memasukkan data area bagian pada PTPN XII Pancursari. *Form* kondisi lapangan digunakan untuk memasukkan kondisi tanaman yang terkena hama atau penyakit. Berdasarkan observasi penulis di PTPN XII Pancursari, kondisi yang diamati yaitu kondisi daun. *Form* perlakuan digunakan untuk memasukkan perlakuan yang dilakukan terhadap tanaman, perlakuan ini meliputi pemupukan, penyiangan, panen dan lain lain. Gambar 5.20 menunjukkan implementasi antarmuka laporan berkala.



Gambar 5.20 Implementasi Antarmuka Laporan Berkala

5.2.5.5 Implementasi antarmuka peta

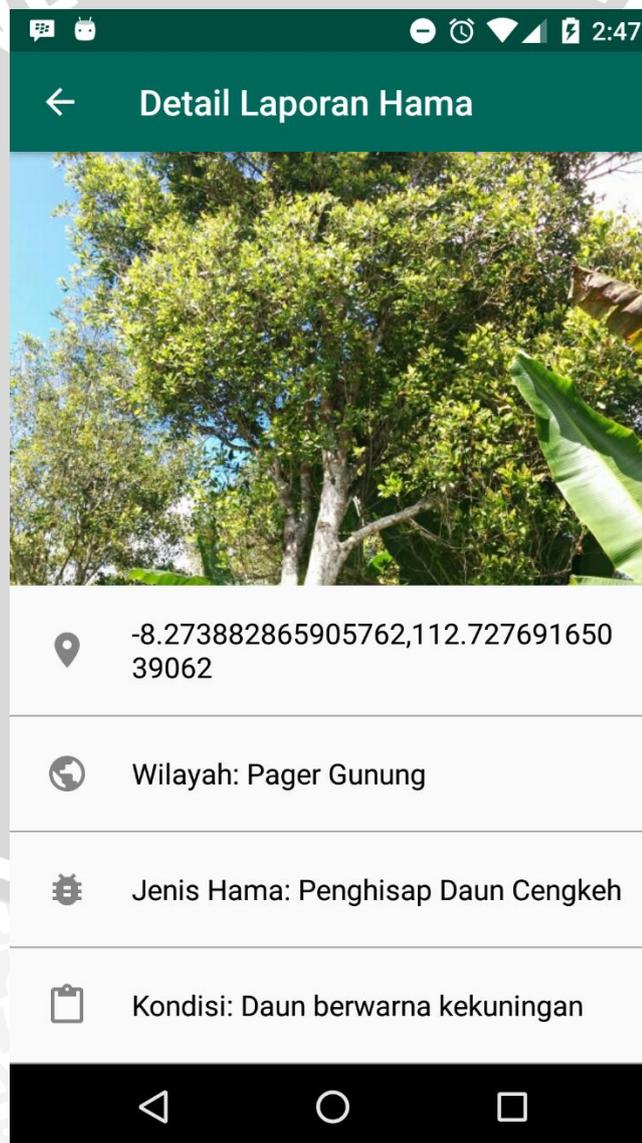
Antarmuka peta didasarkan pada perancangan antarmuka peta pada Gambar 5.12. Antarmuka peta menampilkan marker berisi informasi mengenai laporan hama maupun laporan berkala. Setelah melakukan *login*, pengguna diarahkan pada antarmuka peta. *Marker* yang ada pada halaman peta dibagi menjadi tiga, bila *marker* berwarna merah menandakan ada laporan hama, bila *marker* berwarna hijau menandakan tanaman yang terkena hama sudah diobati atau dilakukan pemeliharaan. *Marker* berwarna biru menandakan ada laporan berkala. Gambar 5.21 menunjukkan implementasi antarmuka peta.



Gambar 5.21 Implementasi Antarmuka Peta

5.2.5.6 Implementasi antarmuka detail laporan

Antarmuka detail laporan didasarkan pada perancangan antarmuka peta pada Gambar 5.15. Antarmuka detail laporan menampilkan detail laporan yang dipilih oleh pengguna. Antarmuka ini berisi detail informasi laporan tanaman yang telah dibuat oleh pengguna. *Imageview* digunakan untuk menampilkan foto tanaman. *Textfield* lokasi digunakan untuk menampilkan informasi *latitude* dan *longitude* yang diambil dari foto pada saat proses *geotagging*. *Textfield* wilayah digunakan untuk menampilkan informasi area bagian PTPN XII Pancursari. *Textfield* jenis hama digunakan untuk menampilkan jenis hama atau penyakit yang menyerang tanaman. *Textfield* kondisi lapangan digunakan untuk menampilkan yang terkena hama atau penyakit. Berdasarkan observasi penulis di PTPN XII Pancursari, kondisi yang diamati yaitu kondisi daun. Gambar 5.22 menunjukkan implementasi antarmuka detail laporan.



Gambar 5.22 Implementasi Antarmuka Detail Laporan

BAB 6 PENGUJIAN DAN ANALISIS HASIL

Pada bab ini dijelaskan mengenai pengujian dan analisis aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android. Pengujian yang dilakukan meliputi pengujian validasi, pengujian *usability* dan pengujian *performance*, sedangkan analisis yang dilakukan meliputi analisis hasil pengujian validasi, analisis hasil pengujian *usability* dan analisis hasil pengujian *performance*.

6.1 Pengujian

Proses pengujian aplikasi dilakukan dengan cara pengujian validasi, pengujian *usability* dan pengujian *performance*.

6.1.1 Pengujian validasi

Pengujian validasi merupakan pengujian dengan teknik *black-box testing*, yaitu menekankan pada kesesuaian antara hasil yang diharapkan dalam kebutuhan dengan hasil yang ditampilkan oleh sistem. Masing-masing kebutuhan dalam sistem dimasukkan dalam kasus uji untuk dilakukan pengujian validasi. Hasil dari proses pengujian validasi akan dianalisis untuk menentukan apakah sistem yang telah dibuat sudah memenuhi kebutuhan. Berikut pengujian validasi aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.

6.1.1.1 Kasus uji validasi lihat peta

Tabel 6.1 menunjukkan kasus uji validasi lihat peta, aplikasi harus dapat menyediakan halaman peta yang menampilkan *marker* berisi laporan hama dan laporan berkala yang dibuat oleh pengguna.

Tabel 6.1 Kasus Uji Validasi Buat Lihat Peta

Nomor Kasus Uji	VAL_GEO_001
Nama Kasus Uji	Lihat Peta
Objek Kasus Uji	Kebutuhan Fungsional (SRS_GEO_001)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan halaman peta yang menampilkan <i>marker</i> berisi laporan hama dan laporan berkala yang dibuat oleh pengguna.
Prosedur Pengujian	1. Menjalankan aplikasi 2. Memilih menu peta
Hasil yang diharapkan	Aplikasi dapat menampilkan informasi laporan hama dan laporan berkala yang dibuat oleh pengguna.

6.1.1.2 Kasus uji validasi buat laporan hama

Tabel 6.2 menunjukkan kasus uji validasi buat laporan hama, aplikasi harus dapat menyediakan *form* untuk membuat laporan hama tanaman pada lokasi tertentu.

Tabel 6.2 Kasus Uji Validasi Buat Laporan Hama

Nomor Kasus Uji	VAL__GEO_002
Nama Kasus Uji	Buat Laporan Hama
Objek Kasus Uji	Kebutuhan Fungsional (SRS_F_002)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan <i>form</i> untuk membuat laporan hama tanaman pada lokasi tertentu.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menjalankan aplikasi 2. Memilih menu buat laporan hama pada <i>floating action button</i> 3. Mengisi data pada <i>form</i> buat laporan hama 4. Menekan tombol submit
Hasil yang diharapkan	Aplikasi dapat melakukan validasi terhadap data laporan hama yang dimasukkan pengguna. Jika data yang dimasukkan benar, aplikasi akan menampilkan pesan "Laporan hama successfully uploaded". Jika data yang dimasukkan salah, aplikasi akan menampilkan pesan <i>error</i> .

6.1.1.3 Kasus uji validasi buat laporan berkala

Tabel 6.3 menunjukkan kasus uji validasi buat laporan berkala, aplikasi harus mampu menyediakan *form* untuk membuat laporan berkala yang berisi informasi kondisi pertumbuhan dan perkiraan produksi tanaman pada lokasi tertentu.

Tabel 6.3 Kasus Uji Validasi Buat Laporan Berkala

Nomor Kasus Uji	VAL_GEO_003
Nama Kasus Uji	Buat Laporan Berkala
Objek Kasus Uji	Kebutuhan Fungsional (SRS_GEO_003)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan <i>form</i> untuk membuat laporan berkala yang berisi informasi kondisi pertumbuhan dan perkiraan produksi tanaman pada lokasi tertentu.

Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menjalankan aplikasi 2. Memilih menu buat laporan berkala pada <i>floating action button</i> 3. Mengisi data pada <i>form</i> buat laporan berkala 4. Menekan tombol submit
Hasil yang diharapkan	Aplikasi dapat melakukan validasi terhadap data laporan hama yang dimasukkan pengguna. Jika data yang dimasukkan benar, aplikasi akan menampilkan pesan "Laporan berkala successfully uploaded". Jika data yang dimasukkan salah, aplikasi akan menampilkan pesan <i>error</i> .

6.1.1.4 Kasus uji validasi lihat detail laporan

Tabel 6.4 menunjukkan kasus uji lihat detail laporan, aplikasi harus mampu menampilkan detail laporan yang dipilih oleh pengguna.

Tabel 6.4 Kasus Uji Lihat Detail Laporan

Nomor Kasus Uji	VAL_GEO_004
Nama Kasus Uji	Lihat Detail Laporan
Objek Kasus Uji	Kebutuhan Fungsional (SRS_GEO_004)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan halaman untuk menampilkan detail laporan yang dibuat oleh pengguna.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menjalankan aplikasi 2. Memilih menu laporan 3. Memilih laporan yang akan dilihat detailnya
Hasil yang diharapkan	Aplikasi dapat menampilkan detail laporan yang dipilih oleh pengguna.

6.1.1.5 Kasus uji validasi update laporan hama

Tabel 6.5 menunjukkan kasus uji validasi update laporan hama, aplikasi harus mampu menyediakan *form* untuk memperbarui laporan hama tanaman yang sudah diobati atau dilakukan pemeliharaan.

Tabel 6.5 Kasus Uji Validasi Update Laporan Hama

Nomor Kasus Uji	VAL_GEO_005
Nama Kasus Uji	Update Laporan Hama
Objek Kasus Uji	Kebutuhan Fungsional (SRS_GEO_005)

Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan <i>form</i> untuk memperbarui laporan hama tanaman yang sudah diobati atau dilakukan pemeliharaan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menjalankan aplikasi 2. Memilih menu laporan 3. Memilih laporan hama, pilih update 4. Mengisi <i>form</i> update laporan hama 5. Memilih tombol update
Hasil yang diharapkan	Aplikasi dapat melakukan validasi terhadap data update laporan hama yang dimasukkan pengguna. Jika data yang dimasukkan benar, aplikasi akan menampilkan pesan "Update laporan hama successfully uploaded". Jika data yang dimasukkan salah, aplikasi akan menampilkan pesan <i>error</i> .

6.1.2 Hasil pengujian validasi

Hasil pengujian validasi berdasarkan kasus uji pada sub bab sebelumnya ditunjukkan pada tabel 6.6 sebagai berikut:

Tabel 6.6 Kasus Uji Validasi

No.	Nomor Kasus Uji	Hasil yang Didapatkan	Status Validitas
1.	VAL_GEO_001	Aplikasi dapat menampilkan informasi laporan hama dan laporan berkala yang dibuat oleh pengguna.	Valid
2.	VAL_GEO_002	Aplikasi dapat melakukan pengambilan foto dan mengunggah foto tersebut ke server dan datanya dapat disimpan pada <i>database</i> . Serta aplikasi dapat menampilkan pesan balasan dari server setelah melakukan unggah foto.	Valid
3.	VAL_GEO_003	Aplikasi dapat melakukan validasi terhadap data laporan hama yang dimasukkan pengguna. Jika data yang dimasukkan benar, aplikasi akan menampilkan pesan "Laporan berkala successfully uploaded". Jika data yang dimasukkan salah, aplikasi akan menampilkan pesan <i>error</i> .	Valid
4.	VAL_GEO_004	Aplikasi dapat menampilkan detail laporan yang dipilih oleh pengguna.	Valid
5.	VAL_GEO_005	Aplikasi dapat melakukan validasi terhadap data update laporan hama yang dimasukkan pengguna. Jika	Valid

	data yang dimasukkan benar, aplikasi akan menampilkan pesan "Update laporan hama successfully uploaded". Jika data yang dimasukkan salah, aplikasi akan menampilkan pesan error.
--	--

6.1.3 Pengujian *usability*

Pengujian *usability* dilakukan untuk mengetahui tingkat kemudahan penggunaan antarmuka aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android.. Pengujian dilakukan dengan membagikan kuisisioner kepada 10 responden. Responden merupakan mahasiswa FILKOM yang telah diedukasi tentang penggunaan aplikasi sebelum pengisian kuisisioner. Metode yang digunakan adalah metode Likert dengan 5 parameter yakni *efficiency*, *learnability*, *memorability*, *error* dan *satisfaction*.

6.1.3.1 Kasus uji

Kasus uji pengujian *usability* meliputi pernyataan-pernyataan yang didefinisikan berdasarkan 5 parameter *usability*. Adapun pernyataan-pernyataannya sebagai berikut:

1. Kecepatan dalam menyelesaikan tugas (*efficiency*)

Parameter *efficiency* digunakan untuk mengukur kecepatan pengguna dalam menjalankan fungsi yang ada pada aplikasi ketika pertama kali menjalankannya. Tabel 6.7 menunjukkan pernyataan berdasarkan parameter *efficiency*.

Tabel 6.7 Pernyataan Parameter *Efficiency*

No	Pernyataan
1.	Proses pembuatan laporan dapat saya jalankan dengan cepat
2.	Halaman peta dapat menampilkan <i>marker</i> yang dibedakan dengan dengan warna sesuai dengan jenis laporan

2. Kemudahan dalam menyelesaikan tugas dasar (*learnability*)

Parameter *learnability* digunakan untuk mengukur tingkat kemudahan penggunaan aplikasi dalam menyelesaikan fungsi-fungsi dasar ketika pengguna pertama kali menggunakan aplikasi. Tabel 6.8 menunjukkan pernyataan berdasarkan parameter *learnability*.

Tabel 6.8 Pernyataan Parameter *Learnability*

No	Pernyataan
1.	Judul tiap halaman sudah jelas dan tepat



2.	Tiap menu sudah diberi nama degan tepat dan mudah dipahami
3.	Informasi yang ditampilkan di tiap halaman sudah jelas

3. Kemudahan dalam mengingat penggunaan aplikasi (*memorability*)

Parameter *memorability* digunakan untuk mengukur tingkat kemudahan dalam mengingat cara menjalankan fungsi dasar ketika pengguna pertama kali menjalankan aplikasi. Tabel 6.9 menunjukkan pernyataan berdasarkan parameter *memorability*.

Tabel 6.9 Pernyataan Parameter Memorability

No	Pernyataan
1.	Aplikasi dapat dioperasikan dengan mudah dan cara penggunaan aplikasi mudah diingat
2.	Ikon-ikon yang terdapat dalam aplikasi mudah dikenali

4. Kemungkinan terjadi kesalahan dan cara mengatasinya (*error*)

Parameter *error* digunakan untuk mengetahui kemungkinan terjadinya *error* atau kesalahan yang dilakukan oleh pengguna dan tingkat kemudahan dalam memahaminya. Tabel 6.10 menunjukkan pernyataan berdasarkan parameter *error*.

Tabel 6.10 Pernyataan Parameter Error

No	Pernyataan
1.	Penyebab kesalahan (<i>error</i>) dapat dengan mudah dipahami dari pesan kesalahan (<i>error</i>) yang muncul

5. Tingkat kepuasan pengguna (*satisfaction*)

Parameter *satisfaction* digunakan untuk mengukur tingkat kepuasan pengguna dalam menggunakan aplikasi. Tabel 6.11 menunjukkan pernyataan berdasarkan parameter *satisfaction*.

Tabel 6.11 Pernyataan Parameter Satisfaction

No	Pernyataan
1.	Saya puas dengan tampilan antarmuka aplikasi
2.	Aplikasi sudah berjalan sesuai dengan yang saya inginkan

6.1.3.2 Hasil pengujian usability

Hasil pengujian *usability* didapatkan berdasarkan penilaian responden melalui kuisioner setelah mencoba aplikasi. Tabel 6.12 menunjukkan pemetaan penilaian hasil pengujian *usability*.

Tabel 6.12 Pemetaan Penilaian Hasil Pengujian Usability

No	Pernyataan	Penilaian				
		STS	TS	N	S	SS
1	Proses pembuatan laporan dapat saya jalankan dengan cepat	0	0	0	7	3
2	Halaman peta dapat menampilkan <i>marker</i> yang dibedakan dengan dengan warna sesuai dengan jenis laporan	0	0	4	4	2
3	Judul tiap halaman sudah jelas dan tepat	0	0	2	5	3
4	Tiap menu sudah diberi nama dengan tepat dan mudah dipahami	0	0	0	7	3
5	Informasi yang ditampilkan di tiap halaman sudah jelas	0	0	3	6	1
6	Aplikasi dapat dioperasikan dengan mudah dan cara penggunaan aplikasi mudah diingat	0	0	0	5	5
7	Ikon-ikon tombol yang terdapat dalam aplikasi mudah dikenali	0	0	2	4	4
8	Penyebab kesalahan dapat mudah dipahami dari pesan kesalahan yang muncul	0	0	3	4	3
9	Saya puas dengan tampilan antarmuka aplikasi	0	0	1	8	1
10	Aplikasi ini sudah berjalan sesuai dengan yang saya inginkan	0	0	0	6	3

Keterangan:

STS : Sangat Tidak Setuju

TS : Tidak Setuju

N : Netral

S : Setuju

SS : Sangat Setuju

Untuk dapat menghitung nilai hasil pengujian *usability*, digunakan perhitungan skala Likert. Nilai interval 20 didapatkan dari pembagian nilai 100 dengan jumlah skor Likert yaitu 5. Skor Likert didapatkan dari pilihan jawaban pada kuisioner dengan skala 1 sampai 5. Tabel 6.13 menunjukkan interpretasi skor Likert.

Tabel 6.13 Interpretasi Skor Likert

Skor Likert	Interpretasi Skor dengan Interval = 20	Pilihan	Status
1	0% - 20%	Sangat Tidak Setuju	Sangat Tidak Mudah
2	20% - 39,99%	Tidak Setuju	Tidak Mudah
3	40% - 59,99%	Netral	Netral
4	60% - 79,99%	Setuju	Mudah
5	80% - 100%	Sangat Setuju	Sangat Mudah

Keterangan:

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

Hasil pengujian berupa total skor didapatkan dari jumlah hasil pengujian dikalikan skor Likert dan indeks yang didapatkan dari presentase dari total skor dibagi dengan total skor maksimum. Tabel 6.14 menunjukkan hasil perhitungan pengujian *usability*.

Tabel 6.14 Hasil Perhitungan Pengujian *Usability*

No	Pernyataan	1	2	3	4	5	Total Skor	Indeks (%)
1	Proses pembuatan laporan dapat saya jalankan dengan cepat	0	0	0	7	3	43	86%
2	Halaman peta dapat menampilkan <i>marker</i> yang dibedakan dengan dengan warna sesuai dengan jenis laporan	0	0	4	4	2	38	76%
Rata-rata Efficiency								81%
3	Judul tiap halaman sudah jelas dan tepat	0	0	2	5	3	41	82%
4	Tiap menu sudah diberi nama dengan tepat dan mudah dipahami	0	0	0	7	3	43	86%
5	Informasi yang ditampilkan di tiap halaman sudah jelas	0	0	3	6	1	38	76%
Rata-rata Learnability								81%
6	Aplikasi mudah dioperasikan dengan mudah dan cara penggunaan aplikasi mudah diingat	0	0	0	5	5	45	90%

7	Ikon-ikon tombol yang terdapat dalam aplikasi mudah dikenali	0	0	2	4	4	42	84%
Rata-rata Memorability								87%
8	Penyebab kesalahan dapat mudah dipahami dari pesan kesalahan yang muncul	0	0	3	4	3	40	80%
Rata-rata Errors								80%
9	Saya puas dengan tampilan antarmuka aplikasi	0	0	1	8	1	40	80%
10	Aplikasi ini sudah berjalan sesuai dengan yang saya inginkan	0	0	0	6	3	39	78%
Rata-rata Satisfaction								79%

Keterangan:

- Total Skor = $\sum STS \times 1 + \sum TS \times 2 + \sum N \times 3 + \sum S \times 4 + \sum SS \times 5$
- Y = Skor Likert tertinggi x jumlah responden
= 5×10
= 50
- Indeks (%) = $\frac{\text{Total Skor}}{Y} \times 100\%$

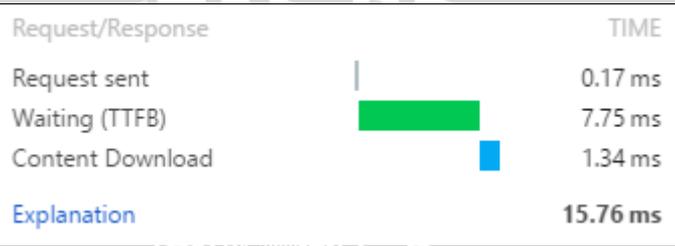
6.1.4 Pengujian *performance*

Pengujian *performance* dilakukan untuk mengetahui tingkat performansi aplikasi dalam memuat data atau respon yang diinginkan. Pengukuran performansi didapatkan dari waktu yang dibutuhkan untuk menampilkan data yang diinginkan. Berdasarkan pengujian ini dapat diketahui apakah waktu respon yang diberikan sistem sudah sesuai dengan batasan yang diberikan. Dalam melakukan pengujian ini, penulis menggunakan *developer tools* pada Google Chrome untuk mengetahui berapa lama waktu yang dibutuhkan oleh *web service* untuk memuat data yang diinginkan oleh pengguna. Data yang dimuat berupa data JSON. Aplikasi *smartphone* terhubung dengan *web service* yang dijalankan di *localhost* melalui jaringan Wi-Fi. Pengujian dilakukan sebanyak 10 kali untuk mendapatkan nilai rata-rata waktu eksekusi. Adapun pengujian *performance* aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android sebagai berikut:

6.1.4.1 Kasus uji performance get_laporan_hama

Tabel 6.15 menunjukkan kasus uji *performance* get_laporan_hama, fungsi ini memuat semua data laporan hama yang telah dibuat oleh pengguna dalam bentuk *marker* pada halaman antarmuka peta.

Tabel 6.15 Kasus Uji Performance Get_Laporan_Hama

Nama fungsi	Get_Laporan_Hama										
Tabel yang terkait	Lap_hama										
Jumlah baris data	6										
Pengujian ke-1	 <table border="1"> <thead> <tr> <th>Request/Response</th> <th>TIME</th> </tr> </thead> <tbody> <tr> <td>Request sent</td> <td>0.17 ms</td> </tr> <tr> <td>Waiting (TTFB)</td> <td>7.75 ms</td> </tr> <tr> <td>Content Download</td> <td>1.34 ms</td> </tr> <tr> <td>Explanation</td> <td>15.76 ms</td> </tr> </tbody> </table>	Request/Response	TIME	Request sent	0.17 ms	Waiting (TTFB)	7.75 ms	Content Download	1.34 ms	Explanation	15.76 ms
Request/Response	TIME										
Request sent	0.17 ms										
Waiting (TTFB)	7.75 ms										
Content Download	1.34 ms										
Explanation	15.76 ms										
Rata-rata waktu yang dibutuhkan	14,043 ms (0,014043 s)										

6.1.4.2 Kasus uji performance get_laporan_berkala

Tabel 6.16 menunjukkan kasus uji *performance* get_laporan_berkala, fungsi ini memuat semua data laporan berkala yang telah dibuat oleh pengguna dalam bentuk *marker* pada halaman antarmuka peta.

Tabel 6.16 Kasus Uji Performance Get_Laporan_Berkala

Nama fungsi	Get_Laporan_Berkala										
Tabel yang terkait	Lap_berkala										
Jumlah baris data	6										
Pengujian ke-1	 <table border="1"> <thead> <tr> <th>Request/Response</th> <th>TIME</th> </tr> </thead> <tbody> <tr> <td>Request sent</td> <td>0.15 ms</td> </tr> <tr> <td>Waiting (TTFB)</td> <td>6.45 ms</td> </tr> <tr> <td>Content Download</td> <td>1.71 ms</td> </tr> <tr> <td>Explanation</td> <td>12.33 ms</td> </tr> </tbody> </table>	Request/Response	TIME	Request sent	0.15 ms	Waiting (TTFB)	6.45 ms	Content Download	1.71 ms	Explanation	12.33 ms
Request/Response	TIME										
Request sent	0.15 ms										
Waiting (TTFB)	6.45 ms										
Content Download	1.71 ms										
Explanation	12.33 ms										

Rata-rata waktu yang dibutuhkan	11,4 ms (0,0114 s)
---------------------------------	--------------------

6.1.4.3 Kasus uji performance `get_detail_hama`

Tabel 6.17 menunjukkan kasus uji *performance* `get_detail_hama`, fungsi ini memuat tiap satu data detail laporan hama yang telah dibuat oleh pengguna.

Tabel 6.17 Kasus Uji Performance Get_Detail_Hama

Nama fungsi	Lihat Detail Laporan										
Tabel yang terkait	Lap_hama										
Jumlah baris data	7										
Pengujian ke-1	<table border="1"> <thead> <tr> <th>Request/Response</th> <th>TIME</th> </tr> </thead> <tbody> <tr> <td>Request sent</td> <td>0.15 ms</td> </tr> <tr> <td>Waiting (TTFB)</td> <td>4.41 ms</td> </tr> <tr> <td>Content Download</td> <td>1.23 ms</td> </tr> <tr> <td>Explanation</td> <td>51.74 ms</td> </tr> </tbody> </table>	Request/Response	TIME	Request sent	0.15 ms	Waiting (TTFB)	4.41 ms	Content Download	1.23 ms	Explanation	51.74 ms
Request/Response	TIME										
Request sent	0.15 ms										
Waiting (TTFB)	4.41 ms										
Content Download	1.23 ms										
Explanation	51.74 ms										
Rata-rata waktu yang dibutuhkan	6,698 ms (0,0066 s)										

6.2 Analisis hasil

Proses analisis hasil bertujuan untuk mengambil kesimpulan terhadap hasil pengujian yang telah dilakukan. Proses analisis hasil meliputi analisis hasil pengujian validasi dan analisis hasil pengujian *usability*.

6.2.1 Analisis hasil pengujian validasi

Proses analisis pengujian validasi dilakukan dengan melihat kesesuaian antara hasil yang diharapkan dalam kebutuhan dengan hasil yang ditampilkan oleh sistem. Berdasarkan hal tersebut, dapat diambil kesimpulan bahwa aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android telah memenuhi kebutuhan dan memberikan hasil yang diharapkan.

6.2.2 Analisis hasil pengujian *usability*

Proses analisis pengujian *usability* dilakukan dengan menghitung nilai rata-rata presentase indeks untuk mengetahui tingkat kemudahan pengguna dalam

menjalankan aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android. Tabel 6.18 menunjukkan tingkat *usability* aplikasi, nilai rata-rata dari tiap parameter *usability* diambil presentase dengan rumus jumlah total rata-rata indeks dibagi jumlah parameter. Dari rumus tersebut didapatkan nilai rata-rata sebesar 80,4%, nilai tersebut masuk ke dalam interval 80-100%. Sehingga dapat diambil kesimpulan bahwa tingkat *usability* pengguna terhadap aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android termasuk dalam tingkat kategori “Sangat Mudah”.

Tabel 6.18 Tingkat Pengujian Usability

Kriteria	Rata-rata Indeks	Status
<i>Efficiency</i>	81%	Sangat Mudah
<i>Learnability</i>	81%	Sangat Mudah
<i>Memorability</i>	87%	Sangat Mudah
<i>Errors</i>	80%	Sangat Mudah
<i>Satisfaction</i>	79%	Mudah

6.2.3 Analisis hasil pengujian *performance*

Proses analisis pengujian *performance* dilakukan dengan membandingkan waktu eksekusi yang dibutuhkan dengan waktu eksekusi yang didefinisikan sesuai dengan kebutuhan non fungsional *performance* (SRS_GEONF_002). Berdasarkan pengujian tiga fungsi yang telah ditentukan, dapat diambil kesimpulan bahwa rata-rata waktu eksekusi yang dibutuhkan yaitu 10,71 *milisecond* atau 0,0107 *second*. Hasil pengujian ini sesuai dengan kebutuhan non fungsional *performance*.

BAB 7 PENUTUP

Pada bab ini dijelaskan mengenai kesimpulan dan saran terkait dengan penelitian yang sudah dilakukan. Sub bab kesimpulan berisi jawaban rumusan masalah yang didefinisikan di awal penelitian. Sub bab saran berisi saran dari penulis untuk pengembangan aplikasi ke depannya.

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Rancangan aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android menghasilkan rancangan arsitektur, rancangan basis data dan rancangan antarmuka. Implementasi dilakukan sesuai dengan hasil perancangan dengan pendekatan *native*.
2. Berdasarkan hasil pengujian validasi, dapat diambil kesimpulan bahwa aplikasi *mobile geotagging* pelaporan kondisi pertumbuhan dan produksi tanaman perkebunan berbasis Android telah memenuhi kebutuhan dan memberikan hasil yang diharapkan. Hasil pengujian *usability* didapatkan nilai presentase sebesar 80,4% dengan kategori "Sangat Mudah". Sedangkan hasil pengujian *performance*, didapatkan rata-rata waktu eksekusi untuk memuat data yaitu 10,71 *milisecond* atau 0,0107 *second*. Hasil ini sesuai dengan kebutuhan non fungsional *performance* yaitu waktu eksekusi kurang dari 3 detik.

7.2 Saran

Berdasarkan penelitian yang telah dilakukan dan hasil pengujian, didapatkan beberapa saran untuk pengembangan aplikasi sebagai berikut:

1. Proses pendeteksian hama dapat dilakukan secara otomatis dengan menambahkan proses pengolahan data citra yang telah diunggah.
2. Penambahan fitur *push notification* kepada pengguna dengan *role* Mandor Pemeliharaan ketika muncul laporan hama.

DAFTAR PUSTAKA

Android Developers, 2016. *Developers.* [Online] Available at: <http://developer.android.com/reference/android/media/ExifInterface.html> [Diakses 18 Februari 2016].

Ariansyah, E., 2016. RANCANG BANGUN APLIKASI SOCIAL GEOTAGGING KERAGAMAN BUDAYA INDONESIA. *Doro Jurnal*, Volume 7.

Efendy, A. A., 2016. RANCANG BANGUN APLIKASI SOSIAL GEOTAGGING KEINDAHAN ALAM INDONESIA PADA SISTEM OPERASI ANDROID. *Doro Jurnal*, Volume 7.

Entwistle, J., 2010. *How can Geotagging be used to Student Activities on Campus?*, s.l.: University of Canterbury.

Google, 2015. *Google Maps JavaScript API.* [Online] Available at: <https://developers.google.com/maps/documentation/javascript/datalayer> [Diakses 18 Februari 2016].

Google, 2015. *Google Maps JavaScript API.* [Online] Available at: <https://developers.google.com/maps/documentation/javascript/layers> [Diakses 18 Februari 2016].

Google, 2015. *Google Maps JavaScript API.* [Online] Available at: <https://developers.google.com/maps/documentation/javascript/markers> [Diakses 18 Februari 2016].

Holdener, A. T., 2011. *HTML5 Geolocation*. 1st penyunt. Sebastopol: O'Reilly Media, Inc..

Mardani, A., 2014. Sistem Informasi Geografis Pelaporan Masyarakat (SIGMA) Berbasis Foto Geotag. *Jurnal Sistem Teknologi Informasi (JustIN)*.

Minarni & Yusdi, Y. F., 2015. Sistem Informasi Geografis Pariwisata Kota Padang Menggunakan Application Programming Interface (API) Google Maps Berbasis Web. *Jurnal TEKNOIF*, pp. 31-37.

Nandipati, A., 2011. *Assessment of metadata associated*, s.l.: Institute for Geoinformatics University of Muenster .

Nielsen, J., 2012. *Nielsen Norman Group.* [Online] Available at: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> [Diakses 18 Februari 2016].

Pressman, R. S., 2010. *Software Engineering: A Practitioner's Approach*. 7th penyunt. New York: McGraw-Hill.

Rosa & Shalahuddin, M., 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Penerbit Modula.

Rouf, A., 2012. Pengujian Perangkat Lunak Dengan Menggunakan Metode White Box Dan Black Box. *HIMSYATECH*, Volume 8, pp. 1-7.

Sholeh, M., Widyastuti, N. & Mashuri, A., 2011. Aplikasi Google Maps API untuk Sistem Informasi Geografis. *Aplikasi Google Maps API*, pp. 97-103.

Suwarto, Octaviany, Y. & Herawati, S., 2014. *Top 15 Tanaman Perkebunan*. 1 penyunt. Jakarta: Penebar Swadaya.

Tim Penyusun, 2013. *Statistik Pertanian 2013*, Jakarta: Pusat Data dan Sistem Informasi Pertanian Kementerian Pertanian RI.

Tim Penyusun, 2015. *Kebun Pancursari*, Malang: PTPN XII Kebun Pancursari.

