

**RANCANG BANGUN APLIKASI PEMETAAN LAHAN HUTAN
MILIK PERUM PERHUTANI KPH MALANG BKPH
SUMBERMANJING MENGGUNAKAN SMARTPHONE
ANDROID**

SKRIPSI

Diajukan Untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana Komputer



Disusun Oleh :

Bagas Dimas Permadi

NIM.0910680094

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
MALANG
2014**

LEMBAR PERSETUJUAN

RANCANG BANGUN APLIKASI PEMETAAN LAHAN HUTAN MILIK PERUM PERHUTANI KPH MALANG BKPH SUMBERMANJING MENGUNAKAN SMARTPHONE ANDROID

SKRIPSI

Diajukan Untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana Komputer



Disusun Oleh :

Bagas Dimas Permadi

NIM.0910680094

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng Herman Tolle, ST., MT.

NIP.19740823200012 1 001

Wibisono Sukmo Wardhono, ST.,MT

NIK. 820404 06 1 1 0091

repository.ub.ac

LEMBAR PENGESAHAN
RANCANG BANGUN APLIKASI PEMETAAN LAHAN HUTAN MILIK
PERUM PERHUTANI KPH MALANG BKPH SUMBERMANJING
MENGUNAKAN SMARTPHONE ANDROID

SKRIPSI
KONSENTRASI REKAYASA PERANGKAT LUNAK

Diajukan Untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Komputer

Disusun Oleh :
Bagas Dimas Permadi
NIM. 0910680094

Skrripsi ini telah diujikan dan dinyatakan lulus pada
Tanggal 8 Juli 2014

Penguji I

Penguji II

Denny Sagita Rusdianto, S.Kom., M.Kom.
NIK. 851124 06 1 1 0250

Issa Arwani, S.Kom., M.Sc.
NIP. 198309222012121003

Penguji III

Fajar Pradana, S.ST., M.Eng.
NIK. 87112116110371

Mengetahui
Ketua Program Studi Teknik Informatika/Illmu Komputer

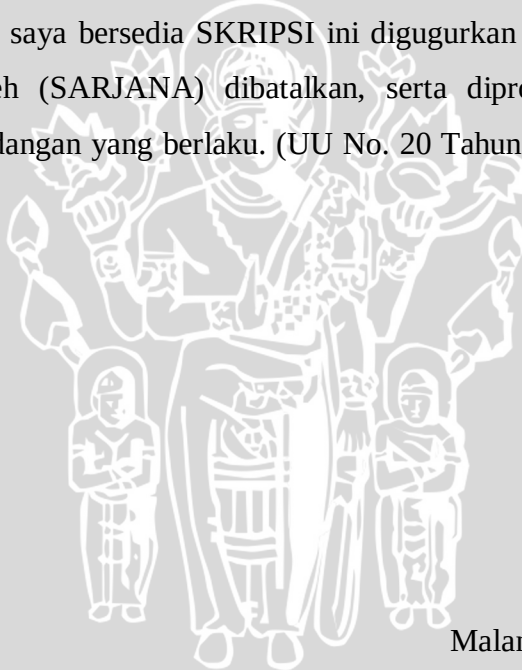
Drs. Marji, M.Si.
NIP. 19670801 199203 1 001

UNIVERSITAS BRAWIJAYA

PERNYATAAN ORISINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan 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, 19 Juni 2014

Mahasiswa,

Bagas Dimas Permadi

NIM.0910680094

KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih dan Penyayang. Segala puji bagi Allah SWT karena atas rahmat dan hidayahNya-lah penulis dapat menyelesaikan Skripsi yang berjudul “Rancang Bangun Aplikasi Pemetaan Lahan Hutan milik Perum PERHUTANI KPH Malang BKPH Sumbermanjing Menggunakan Smartphone Android”. Shalawat dan salam atas junjungan besar kita Nabi Muhammad S.A.W. beserta keluarga dan para sahabat sekalian. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih penulis yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan – bantuan baik lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :

1. Ibunda Dra. Sri Sukanti, Ayahanda Drs. Sumarsono, M.Pd, dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti – hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
2. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Bapak Drs. Marji, M.T, dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
4. Bapak Dr. Eng Herman Tolle, ST., MTdanWibisono Sukmo Wardhono, ST.,MT selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.

5. Bapak Suprpto, ST., MTselaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
6. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
7. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
8. Melina Rubaeah, S.Si sekeluarga yang selalu memberikan motivasi, dukungan, semangat, dan doa selama penulis menempuh dan menyelesaikan studi di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
9. Sahabat-sahabatku Hafidz Rozaq Nur J., S.Kom, Hoyi Ndadak Aji, S.Kom, Iqbal Lutfillah, S.Kom, Angie Pramudhita, S.Kom, Himawan Aditya, S.Kom dan teman – teman kontrakan bahagia yang telah memberi motivasi, dukungan, semangat, dan doa selama penulis menempuh dan menyelesaikan studi di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
10. Teman-teman Angkatan 2009, 2010, 2011, dan 2012 Teknik Informatika, terimakasih atas segala bantuannya selama menempuh studi di Program Teknologi Informasi dan Ilmu Universitas Brawijaya.
11. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun yang tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 19 Juni 2014

Penulis

ABSTRAK

Bagas Dimas P. 2014. : Rancang Bangun Aplikasi Pemetaan Lahan Hutan milik Perum PERHUTANI KPH Malang BKPH Sumbermanjing Menggunakan Smarthphone Android. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing : Dr. Eng Herman Tolle, ST., MT dan Wibisono Sukmo Wardhono, ST.,MT.

Sistem pendataan dan pemetaan lahan yang dilakukan oleh Perum Perhutani KPH Malang BKPH Sumber Manjing masih menggunakan cara yang manual. Yaitu dengan mencatat hasil pemetaan pada sebuah buku RPKH (Rencana Pengaturan Kelestarian Hutan) lalu memetakan pada sebuah peta yang telah disediakan. Cara tersebut memiliki resiko kerusakan data yang cukup besar, untuk itu diperlukan suatu sistem baru yang dapat melakukan pekerjaan tersebut dengan cepat dan efisien, yaitu perangkat lunak Pemetaan Lahan Hutan berbasis Android (*Assisted GPS*).

Perangkat lunak Pemetaan Lahan Hutan berbasis Android terbagi menjadi dua subsistem, yaitu subsistem aplikasi user dan subsistem aplikasi administrator. Perangkat lunak Pemetaan Lahan Hutan berbasis Android dikembangkan dengan metode CBSE (*Component-Based Software Engineering*) karena dirasa dapat memudahkan dalam perancangan perangkat lunak. Sebab, CBSE menggunakan komponen perangkat lunak yang sudah ada dan bersifat dapat digunakan kembali. Perancangan perangkat lunak dimodelkan dengan UML (*Unified Modelling Language*) dan diimplementasikan dengan bahasa pemrograman *Java*. Pengujian perangkat lunak Pemetaan Lahan Hutan berbasis Android dilakukan dengan menggunakan *black-box testing* serta pengujian performa melalui analisis perbandingan antara perangkat GPS dengan tiga *Assisted GPS* yang berbeda.

Dari hasil pengujian yang diperoleh dapat disimpulkan bahwa ketepatan antara perangkat GPS dengan tiga *Assisted GPS* yang berbeda berjarak kurang lebih 3-5 meter dan memiliki selisih luas sebesar 200,207 meter². Perangkat *Assisted GPS* pada *Smartphone* Sony Experia L memiliki akurasi paling bagus dibandingkan dua perangkat lainnya. Selain itu perangkat GPS mempunyai banyak kelemahan yaitu harga perangkat yang mahal dan membutuhkan waktu yang lama pada saat penentuan lokasi.

Kata Kunci : Perangkat lunak Pemetaan Lahan Hutan, CBSE, *Assisted GPS*.

ABSTRACT

Bagas Dimas Permadi. 2014. : *Design of PERHUTANI's Android-Based Forest Land Mapping Application.*

Advisor : Dr. Eng Herman Tolle, ST., MT and Wibisono Sukmo Wardhono, ST.,MT.

Data collection system and land mapping by Perum Perhutani KPH Malang BKP Sumber Manjing still using manual method, by recording note the map results of the mapping in a RPKH book (Plannning for Forest Sustainability) and then, charted on a map that has been provided. This way it has a large risk of data recording, that is why we purpose a new system that can do the task quickly and efficiently, named Forest Land Mapping software based on Android (Assisted GPS).

Android-Based Forest Land Mapping software is divided into two subsystems: user's application and administrator's application. Android-Based Forest Land Mapping software developed by the method of CBSE (Component-Based Software Engineering) because it can be easy to developed. Therefore, CBSE using software components that already exist and reusable. Software design modeled by UML (Unified Modelling Language) and implemented with the Java programming language. Android-Based Forest Land Mapping software testing is performed using black-box testing and performance testing through comparative analysis between a GPS device with three different Assisted GPS.

The test results obtained can be concluded that the accuracy of the GPS devices with three different Assisted GPS is approximately 3-5 meters and has a difference area of 200,207 meters². Assisted GPS devices in Smartphones Sony Experia L has the best accuracy compared the other devices. In addition to the GPS device has many weaknesses that the price is expensive, and takes a long time when deciding on the location.

Keywords : Forest Land Mapping software, CBSE, Assisted GPS.

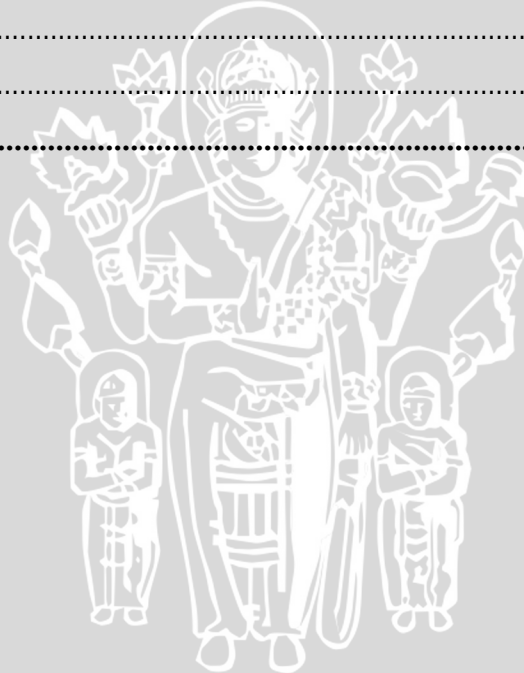
DAFTAR ISI

KATA PENGANTAR.....	i
ABSTRAK.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xii
BAB IPENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	4
1.6 Sistematika Pembahasan.....	4
BAB IITINJAUAN PUSTAKA.....	6
2.1 Hutan Produksi.....	6
2.1.1 Tanaman Jati.....	6
2.1.2 Tanaman Sengon.....	7
2.1.3 Tanaman Mahoni.....	7
2.1.4 Tanaman Tumpangsari.....	8
2.2 Perangkat Lunak.....	8
2.2.1 Produk generik.....	9
2.2.2 Produk pesanan.....	9
2.3 Rekayasa Perangkat Lunak.....	9
2.4 <i>Component-Based Software Engineering</i>	10
2.5 <i>Unified Modelling Language</i>	15
2.5.1 <i>UseCase Diagram</i>	15
2.5.2 <i>Class Diagram</i>	18
2.5.3 <i>Activity Diagram</i>	21
2.6 Pengujian Perangkat Lunak.....	22

2.6.1	Teknik Pengujian <i>Black-Box Testing</i>	23
2.6.2	Strategi Pengujian	23
2.6.2.1	Pengujian Validasi	24
2.6.2.2	Pengujian Performa	24
2.7	Bahasa Pemrograman.....	25
2.7.1	Bahasa Pemrograman HTML dan CSS.....	25
2.7.2	Bahasa Pemrograman PHP	25
2.7.3	Bahasa Pemrograman Java	26
2.8	Basis Data.....	26
2.8.1	MySQL.....	26
2.8.2	SQLite	28
2.9	Google Maps Api.....	29
2.10	Sistem Operasi Android	29
2.11	GPS	30
2.11.1	<i>Tracking</i> Sistem	30
2.11.2	<i>Asissted-GlobalPositioning</i> Sistem (A-GPS)	31
2.11.3	Sumber Kesalahan Perhitungan Posisi GPS (Global Positioning Sistem).....	32
BAB IIIMETODE PENELITIAN		34
BAB IVPERANCANGAN		38
4.1	Analisis Kebutuhan.....	39
4.1.1	Gambaran Umum Perangkat Lunak.....	39
4.1.2	Deskripsi Perangkat Lunak.....	39
4.1.3	Cara Penggunaan Perangkat Lunak.....	41
4.1.4	Identifikasi Aktor	44
4.1.5	Analisa <i>Use Case</i>	45
4.1.5.1	Diagram <i>Use Case</i> Pemetaan Lahan Hutan Berbasis Android.....	47
4.1.6	Analisis Data.....	66
4.2	Perancangan Perangkat Lunak.....	67
4.2.1	Perancangan Basis Data	67
4.2.2	Pemodelan Kelas.....	71

4.2.2.1	Pemodelan Kelas Komponen <i>Helper</i>	72
4.2.2.2	Pemodelan Kelas Komponen <i>Inti</i>	75
4.2.3	Perancangan <i>Sequences Diagram</i>	81
4.2.4	Perancangan Antarmuka	88
4.1.4.1	Perancangan Antarmuka Aplikasi <i>User</i>	88
4.1.4.2	Perancangan Antarmuka Aplikasi <i>Administrator</i>	94
BAB VIMPLEMENTASI		98
5.1	Spesifikasi Sistem	99
5.1.1	Spesifikasi Perangkat Keras	99
5.1.2	Spesifikasi Perangkat Lunak	99
5.2	Batasan-Batasan Implementasi	100
5.3	Implementasi Basis Data	100
5.4	Implementasi <i>Class/Interface</i> dan <i>Layout</i>	101
5.5	Implementasi dengan Komponen	103
5.4.1	Implementasi <i>Login</i>	103
5.4.2	Implementasi Membuat Informasi Data Lokasi	104
5.4.3	Implementasi Menambah Detail Tumpangsari	105
5.4.4	Implementasi Menambah Detail Lokasi	106
5.4.5	Implementasi Mengubah Informasi Data Lokasi	107
5.4.6	Implementasi Mengirim Informasi Data Lokasi	108
5.4.7	Implementasi Membuat Waktu Tebang Tanaman Pokok	109
5.4.8	Implementasi Membuat Waktu Tebang Tanaman Tumpangsari	110
5.4.9	Implementasi Membuat Luas Wilayah	110
5.4.10	Implementasi Logout	111
5.6	Implementasi Antarmuka	112
5.5.1	Implementasi Antarmuka Aplikasi <i>User</i>	112
5.5.2	Implementasi Antarmuka Aplikasi <i>Administrator</i>	118
BAB VIPENGUJIAN DAN ANALISIS		122
6.1	Pengujian	123
6.1.1	Pengujian Validasi	123

6.1.1.1 Pengujian Validasi Aplikasi <i>User</i>	123
6.1.1.2 Pengujian Validasi Aplikasi <i>Administrator</i>	125
6.1.2 Pengujian Performa.....	127
6.1.2.1 Gambar Hasil Perbandingan Jarak Titik Koordinat Dalam Peta	134
6.1.2.2 Gambar Hasil Perbandingan Luas Bidang Dalam Peta	136
6.2 Analisis	137
6.2.1 Analisis Metode Pengembangan Perangkat Lunak.....	137
6.2.2 Analisis Hasil Pengujian Validasi.....	138
6.2.3 Analisis Hasil Pengujian Performa	138
BAB VIIPENUTUP	139
7.1 Kesimpulan.....	139
7.2 Saran	140
DAFTAR PUSTAKA.....	141



DAFTARGAMBAR

Gambar 2.1 <i>Software process</i> dari CBSE	11
Gambar 2.2 Contoh <i>use case diagram</i>	16
Gambar 2.3 Contoh sebuah <i>class</i>	18
Gambar 2.4 Contoh <i>package</i>	19
Gambar 2.5 Contoh <i>class diagram</i>	20
Gambar 2.6 Contoh <i>activity diagram</i>	22
Gambar 2.7 Contoh SQLite	28
Gambar 2.8 <i>Tracking</i> Sistem	31
Gambar 2.9 Rambatan gelombang dari lapisan ionosfer.....	32
Gambar 2.10 Terjadinya benturan sinyal satelit pada obyek.....	33
Gambar 3.1 Metode Penelitian CBSE (<i>Component Based Software Engineering</i>)...	34
Gambar 4.1 Diagram Blok Perancangan	38
Gambar 4.2 Urutan cara penggunaan perangkat lunak	41
Gambar 4.3 Diagram Alur Tambah Data Koordinat Lokasi	43
Gambar 4.4 Diagram <i>use case</i>	48
Gambar 4.5 Diagram aktivitas login	50
Gambar 4.6 Diagram aktivitas Menambah Informasi Data Lokasi di <i>Database</i>	51
Gambar 4.7 Diagram aktivitas Mengubah Informasi Data Lokasi di <i>Database</i>	53
Gambar 4.8 Diagram aktivitas Mengirim Informasi Data Lokasi di <i>Database</i>	54
Gambar 4.9 Diagram aktivitas Menambah Informasi Detail Lokasi	56
Gambar 4.10 Diagram aktivitas Menambah Informasi Detail Tumpangsari	58
Gambar 4.11 Diagram aktivitas Logout	59
Gambar 4.12 Diagram aktivitas <i>Login administrator</i>	61
Gambar 4.13 Diagram aktivitas Membuat Akun Klien	62
Gambar 4.14 Diagram aktivitas Melihat Informasi Data Klien.....	63
Gambar 4.15 Diagram aktivitas MelihatSeluruhInformasi Data Lokasi.....	64
Gambar 4.16 Diagram aktivitas <i>Logout Admin</i>	66
Gambar 4.17 Diagram <i>EntityRelationship</i>	68
Gambar 4.18 Pemodelan MVC Kelas Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	71
Gambar 4.19 Kelas DataBaseHelper Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android.....	72
Gambar 4.20 Kelas GPSServicesPerangkat Lunak Pemetaan Lahan Hutan Berbasis Androidip	73
Gambar 4.21 Kelas HttpClient Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	74

Gambar 4.22 Kelas HttpClient Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	75
Gambar 4.23 Kelas HomeActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	76
Gambar 4.24 Kelas BuatLokasiActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	77
Gambar 4.25 Kelas DetailLokasiActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	78
Gambar 4.26 Kelas DetailTumpangsariActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android.....	79
Gambar 4.27 Kelas InfoDataActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	80
Gambar 4.28 Kelas LoginActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	81
Gambar 4.29 Diagram blok perancangan <i>sequences diagram</i>	82
Gambar 4.30 <i>Sequence diagram</i> detail lokasi Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android.....	83
Gambar 4.31 <i>Sequence diagram</i> mengirim data lokasi Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	84
Gambar 4.32 <i>Sequence diagram</i> Penentuan Luas wilayah Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	85
Gambar 4.33 <i>Sequence diagram</i> penentuan waktu tebang tanaman pokok Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	86
Gambar 4.34 <i>Sequence diagram</i> penentuan waktu tebang tanaman tumpangsari Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android	87
Gambar 4.35 <i>Sitemap</i> Antarmuka Aplikasi User.....	88
Gambar 4.36 Halaman Login Pada Aplikasi User.....	89
Gambar 4.37 Halaman Utama Pada Aplikasi User.....	90
Gambar 4.38 Halaman Pengisian <i>form</i> tambah lokasi Pada Aplikasi User	91
Gambar 4.39 Halaman pembuatan detail lokasi	92
Gambar 4.40 Halaman pembuatan detail tumpangsari	93
Gambar 4.41 Halaman informasi data lokasi.....	94
Gambar 4.42 <i>Sitemap</i> Antarmuka Aplikasi Administrator	95
Gambar 4.43 Rancangan Antarmuka Halaman Home	95
Gambar 4.44 Rancangan Antarmuka Halaman Data Lokasi.....	96
Gambar 4.45 Rancangan Antarmuka Halaman Data Klien.....	97
Gambar 5.1 Diagram Blok Implementasi.....	98
Gambar 5.2 Diagram <i>Entity relationship</i> Aplikasi Pemetaan Lahan Hutan.....	101

Gambar 5.3 Implementasi Cuplikan <i>Code Login</i>	104
Gambar 5.4 Implementasi Cuplikan <i>Code Membuat Informasi Data Lokasi</i>	105
Gambar 5.5 Implementasi Cuplikan <i>Code Menambah Detail Tumpangsari</i>	105
Gambar 5.6 Implementasi Cuplikan <i>Code Menambah Detail Lokasi</i>	106
Gambar 5.7 Implementasi Cuplikan <i>Code Mengubah Informasi Data Lokasi</i>	107
Gambar 5.8 Implementasi Cuplikan <i>Code Mengirim Informasi Data Lokasi</i>	108
Gambar 5.9 Implementasi Cuplikan <i>Code Membuat Waktu Tebang Tanaman Pokok</i>	109
Gambar 5.10 Implementasi Cuplikan <i>Code Membuat Waktu Tebang Tanaman</i> <i>Tumpangsari</i>	110
Gambar 5.11 Implementasi Cuplikan <i>Code Membuat Luas Wilayah</i>	111
Gambar 5.12 Implementasi Cuplikan <i>Code Logout</i>	112
Gambar 5.13 Antarmuka Login	113
Gambar 5.14 Antarmuka Halaman <i>Home</i>	114
Gambar 5.15 Antarmuka Membuat Informasi Data Lokasi	115
Gambar 5.16 Antarmuka Detail Tumpangsari.....	116
Gambar 5.17 Antarmuka Detail Lokasi.....	117
Gambar 5.18 Antarmuka Informasi Data Lokasi.....	117
Gambar 5.19 Antarmuka Halaman Data Lokasi.....	118
Gambar 5.20 Antarmuka Halaman Data Klien.....	119
Gambar 5.21 Antarmuka Halaman Buat Klien.....	120
Gambar 5.22 Antarmuka Halaman Lihat Peta.....	121
Gambar 6.1 Diagram Alir Pengujian dan Analisis.....	122
Gambar 6.2 GPS GARMIN 60	128
Gambar 6.3 Sony Experia Tipo	129
Gambar 6.4 Sony Experia L	130
Gambar 6.5 Lenovo A3000-H.....	130
Gambar 6.6 Contoh selisih jarak perangkat GPS dan GPS pada <i>Smartphone</i> Sony <i>Experia Tipo</i>	134
Gambar 6.7 Contoh selisih jarak perangkat GPS dan GPS pada <i>Smartphone</i> Sony <i>Experia L</i>	135
Gambar 6.8 Contoh selisih jarak perangkat GPS dan GPS pada <i>Smartphone</i> Lenovo <i>A3000-H</i>	135
Gambar 6.9 Luas bidang dalam peta pada pengukuran menggunakan GPS <i>Smartphone</i> Sony Experia LL	136
Gambar 6.10 Luas bidang dalam peta pada pengukuran menggunakan GPS	136

DAFTAR TABEL

Tabel2.1 Keterangan simbol - simbol <i>use case diagram</i>	16
Tabel2.2 Keterangan simbol - simbol <i>class diagram</i>	20
Tabel4.1 Identifikasi aktor.....	45
Tabel4.2 Spesifikasi kebutuhan fungsional <i>user</i>	45
Tabel4.3 Spesifikasi kebutuhan fungsional <i>administrator</i>	46
Tabel4.4 <i>Use case</i> Menunjukkan <i>Login</i> Klien	48
Tabel4.5 <i>Use case</i> Menambah Informasi Data Lokasi di <i>Database</i>	50
Tabel4.6 <i>Use case</i> Mengubah Informasi Data Lokasi di <i>Database</i>	52
Tabel4.7 <i>Use case</i> Mengirim Informasi Data Lokasi di <i>Database</i>	53
Tabel4.8 <i>Use case</i> Menambah Informasi Detail Lokasi	54
Tabel4.9 <i>Use case</i> Menambah Informasi Detail Tumpangsari	56
Tabel4.10 <i>Use case Logout</i> Klien	58
Tabel4.11 <i>Use case</i> Menunjukkan <i>Login Administrator</i>	59
Tabel4.12 <i>Use Case</i> Membuat Akun Klien.....	61
Tabel4.13 <i>Use Case</i> Melihat Informasi Data Klien.....	62
Tabel4.14 <i>Use Case</i> Melihat Seluruh Informasi Data Lokasi.....	63
Tabel4.15 <i>Use Case Logout</i> Admin	65
Tabel4.16 Struktur tabel admin	68
Tabel4.17 Struktur tabel klien	69
Tabel4.18 Struktur tabel lokasi.....	69
Tabel4.19 Struktur tabel detail_lokasi	70
Tabel5.1 Spesifikasi Perangkat Keras	99
Tabel5.2 Spesifikasi Perangkat Lunak	99
Tabel5.3 Implementasi <i>class</i> pada kode program *.java.....	102
Tabel5.4 Implementasi <i>Layout</i>	103
Tabel6.1 Pengujian Validasi Aplikasi <i>User</i>	123
Tabel6.2 Pengujian Validasi Aplikasi <i>Administrator</i>	125
Tabel6.3 Spesifikasi perangkat keras GPS.....	127
Tabel6.4 Spesifikasi perangkat lunak <i>Smartphone</i> Android Sony Experia Tipo.....	128
Tabel6.5 Spesifikasi perangkat lunak <i>Smartphone</i> Android Sony Experia L.....	129
Tabel6.6 Spesifikasi perangkat lunak <i>Smartphone</i> Android Lenovo A3000-H	130
Tabel6.7 Perbandingan alat GPS dan Assited GPS Sony Experia Tipo.....	131
Tabel6.8 Perbandingan alat GPS dan Assited GPS Sony Experia L.....	132
Tabel6.9 Perbandingan alat GPS dan Assited GPS Lenovo A3000-H.....	132
Tabel6.10 Perbandingan Luas Wilayah menggunakan alat GPS dan Assited GPS..	133

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di dalam kantor Perum Perhutani KPH (Kesatuan Pengelolaan Hutan) Malang BKPH (Bagian Kesatuan Pengelolaan Hutan) Sumber Manjing, sistem pendataan dan pemetaan lahan yang dilakukan masih menggunakan cara yang manual. Yaitu dengan mencatat hasil pemetaan pada sebuah buku. Sistem yang dilakukan saat ini, apabila ada tindakan reboisasi atau penebangan maka petugas terkait akan membuat data koordinat lahan dan keterangan kegiatan pada buku RPKH (Rencana Pengaturan Kelestarian Hutan) lalu memetakan pada sebuah peta yang telah disediakan. Buku RPKH tersebut juga berfungsi sebagai tempat untuk data warga sekitar yang ingin memanfaatkan lahan Perum Perhutani yang biasa disebut dengan tumpangsari.

Apabila terjadi sesuatu seperti kerusakan dan kehilangan buku RPKH, maka seluruh data tentang pemakaian lahan dan pemetaan lahan akan hilang. Oleh karena itu dibutuhkan suatu sistem untuk menanggulangi resiko tersebut dengan menggunakan sistem informasi geografis. Sehingga data – data yang tadinya hanya bisa dilihat di buku dapat dilihat melalui web secara *real time*.

Di dalam sistem ini, dibutuhkan sebuah perangkat *mobile* yang berfungsi sebagai alat untuk mengirimkan data lahan hutan ke kantor Perum Perhutani. Android merupakan platform yang dipilih oleh penulis untuk mendata lahan hutan yang akan dipetakan. Android digunakan karena pengembangan aplikasi pada platform ini bersifat *open source*. Di Indonesia, pengguna Android sudah menembus angka psikologis 100 juta pengguna [MOB-12]. Metode pengembangan perangkat lunak yang akan digunakan adalah metode *Component-Based Software Engineering* (CBSE) karena dirasa dapat memudahkan dalam perancangan perangkat lunak. Sebab, menggunakan komponen perangkat lunak yang sudah ada dan bersifat dapat digunakan kembali [PRE-01:721].

Berdasarkan latar belakang tersebut penulis mengambil judul “Rancang Bangun Aplikasi Pemetaan Lahan Hutan milik Perum Perhutani KPH Malang BKPH Sumbermanjing Menggunakan Smartphone Android”. Sebuah aplikasi GPS yang ada dalam Smartphone bersistem operasi Android yang mempunyai fungsi sama dengan alat GPS yaitu untuk menentukan letak koordinat yang akan dituju, dengan harapan perangkat lunak ini nantinya dapat membantu dan mempermudah pegawai atau *user* Perhutani KPH Malang BKPH Sumbermanjing dalam usahanya untuk melakukan pendataan dan pemetaan lahan.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang diangkat pada bagian latar belakang, maka rumusan masalah dikhususkan pada :

1. Bagaimana merancang perangkat lunak aplikasi Pemetaan Lahan Hutan milik Perum Perhutani KPH Malang BKPH Sumbermanjing?
2. Bagaimana teknis implementasi pemrograman dan pengembangan perangkat lunak aplikasi Pemetaan Lahan Hutan milik Perum Perhutani KPH Malang BKPH Sumbermanjing?
3. Bagaimana proses dan hasil pengujian aplikasi Pemetaan Lahan Hutan milik Perum Perhutani KPH Malang BKPH Sumbermanjing?

1.3 Batasan Masalah

Agar diperoleh hasil pembahasan yang sesuai dengan apa yang diharapkan, maka perlu diberikan batasan masalah pada pengembangan perangkat lunak ini, yaitu :

1. Pembahasan difokuskan pada rekayasa perangkat lunak dari pembuatan perangkat lunak *GPS* berbasis Android untuk pendataan dan pemetaan lahan.
2. Sumber data yang digunakan adalah sumber data yang diperoleh dari kantor Perum Perhutani KPH Malang BKPH Sumbermanjing.

3. IDE (*Integrated Development Environment*) yang dipakai dalam pengembangan perangkat lunak GPS berbasis Android untuk pendataan dan pemetaan lahan adalah Eclipse, Emulator Android, dan Dreamweaver.
4. GPS berbasis Android untuk pendataan dan pemetaan lahan menggunakan perangkat mobile berbasis Android minimal versi 4.0 ICS (*Ice Cream Sandwich*).
5. Pengembangan perangkat lunak dilakukan dengan bahasa pemrograman Java, PHP, HTML, dan CSS.
6. Versi platform JDK (*Java Development Kit*) yang dipakai adalah JDK 7 Update 21.
7. DBMS (*Database Management System*) yang digunakan dalam pengembangan perangkat lunak ini adalah MySQL dan SQLite.
8. Performa penguncian posisi koordinat *user* tergantung performa GPS dari perangkat.

1.4 Tujuan

Sesuai dengan latar belakang dan rumusan masalah, tujuan dari pengembangan perangkat lunak ini adalah sebagai berikut:

1. Mewujudkan perancangan dan rekayasa perangkat lunak dari aplikasi Pemetaan Lahan Hutan milik Perum Perhutani KPH Malang BKPH Sumbermanjing.
2. Mewujudkan teknis implementasi pemrograman dan pengembangan perangkat lunak dari aplikasi Pemetaan Lahan Hutan milik Perum Perhutani KPH Malang BKPH Sumbermanjing.
3. Memperoleh skenario, proses dan hasil pengujian perangkat lunak dari aplikasi Pemetaan Lahan Hutan milik Perum Perhutani KPH Malang BKPH Sumbermanjing.

1.5 Manfaat

Manfaat yang nantinya dapat diambil dari pengembangan perangkat lunak ini adalah sebagai berikut :

a. Bagi penulis

1. Menerapkan ilmu yang telah didapatkan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
2. Mendapatkan pemahaman lebih lanjut dalam pengembangan perangkat lunak dari aplikasi Pemetaan Lahan Hutan milik Perum Perhutani KPH Malang BKPH Sumbermanjing.
3. Sebagai salah satu referensi belajar bagi mahasiswa untuk meningkatkan pengetahuannya dalam bidang perancangan khususnya perancangan aplikasi perangkat lunak.

b. Bagi pengguna

1. Menyediakan aplikasi yang dapat digunakan untuk pendataan dan pemetaan lahan dengan menggunakan teknologi *GPS* berbasis *Android* pada Perum Perhutani KPH Malang BKPH Sumbermanjing.
2. Untuk menekan biaya dan mempermudah tugas/pekerjaan pegawai atau *user* Perum Perhutani KPH Malang BKPH Sumbermanjing.

1.6 Sistematika Pembahasan

Sistematika pembahasan ditunjukkan untuk memberikan gambaran dan uraian dari penulisan skripsi ini secara garis besar yang meliputi beberapa bab, sebagai berikut :

Bab I : Pendahuluan

Menguraikan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika pembahasan.

Bab II : Kajian Pustaka dan Dasar Teori

Menguraikan tentang dasar teori dan referensi yang mendasari pengembangan perangkat lunak GPS berbasis Android untuk pendataan dan pemetaan lahan.

Bab III : Metode Penelitian dan Perancangan

Menguraikan dan membahas langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri dari studi literature, implementasi, perancangan, analisis kebutuhan dan pengujian perangkat lunak GPS berbasis Android untuk pendataan dan pemetaan lahan sesuai dengan dasar teori yang ada.

Bab IV : Implementasi

Membahas implementasi dari perangkat lunak GPS Berbasis Android untuk pendataan dan pemetaan lahan sesuai dengan perancangan perangkat lunak yang telah dibuat.

Bab V : Pengujian dan Analisis

Memuat hasil pengujian dan analisis terhadap perangkat lunak yang telah direalisasikan.

Bab VI : Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran – saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

Pada bab ini berisi pembahasan tentang teori dasar yang berhubungan dengan pengembangan perangkat lunak yang dilakukan. Teori dasar yang akan dibahas pada bab ini yaitu penjelasan tentang konsep dasar perangkat lunak, pengembangan dan rekayasa perangkat lunak, konsep *Component-Based Software Engineering*, konsep dasar *Unified Modelling Language* yang dipakai pada rekayasa perangkat lunak, konsep dasar teknik pengujian perangkat lunak, penjelasan tentang *Web Service*, penjelasan tentang bahasa pemrograman HTML dan CSS, penjelasan tentang bahasa pemrograman PHP, dan penjelasan tentang bahasa pemrograman Java. Konsep dan implementasi basis data, penjelasan tentang MySQL, penjelasan tentang SQLite, penjelasan tentang Google Maps API, pengenalan tentang sistem operasi Android, teknologi GPS dan A-GPS yang digunakan.

2.1 Hutan Produksi

Hutan produksi adalah kawasan yang diperuntukkan guna produksi hasil hutan untuk memenuhi keperluan masyarakat pada umumnya dan khususnya untuk pembangunan industri dan ekspor. Hutan ini biasanya terletak di dalam batas-batas suatu HPH (Hak Pengusahaan Hutan) dan dikelola untuk menghasilkan kayu. Dengan pengelolaan yang baik, tingkat penebangan diimbangi dengan penanaman dan pertumbuhan ulang sehingga hutan terus menghasilkan kayu secara lestari. Secara praktis, hutan-hutan di kawasan HPH sering dibalak secara berlebihan dan kadang ditebang habis [DEP-00].

2.1.1 Tanaman Jati

Hutan Jati sebagian besar tumbuh di pulau Jawa, yang dikelola oleh Perum Perhutani seluas 2,6 juta ha yang terdiri dari 54 KPH (Kesatuan Pemangku Hutan). Tanaman Jati tersebut umumnya dapat dipanen apabila sudah berumur 20 tahun lebih

karena tanaman tersebut dirasa sudah cukup tua untuk di produksi. Produksi hutan jati yang dikelola Perum Perhutani rata-rata 800 ribu m³/tahun. Sebagian besar (85%) dijual dalam bentuk log, sedangkan sisanya digunakan untuk memenuhi kebutuhan bahan baku industri milik Perum Perhutani dan Industri Mitra Kerja Sama Pengelolaan Mitra (Mitra KSP) Perhutani dengan swasta [AMI-01].

2.1.2 Tanaman Sengon

Tanaman Sengon menurut Warisho adalah tanaman dengan masa produksi tahunan, artinya kita perlu menunggu setidaknya 5-8 tahun sebelum tanaman sengon memberikan hasil”. Adapula pengertian menurut Frans Wanggai, tanaman sengon adalah tanaman yang umurnya berumur antara empat sampai enam tahun sudah dapat dipanen [HBS-92]. Berdasarkan paparan tentang tanaman sengon di atas penulis dapat menyimpulkan bahwa tanaman sengon adalah tanaman yang hasil panennya antara 5 tahunan.

2.1.3 Tanaman Mahoni

Kayu mahoni merupakan salah satu jenis kayu khas daerah tropis. Di Indonesia, kayu mahoni sangat populer khususnya untuk banyak daerah di pulau Jawa. Kayu ini dikenal sebagai jenis kayu yang bernilai komersial tinggi sehingga banyak orang yang membudidayakan dan diperjual belikan pada pasar komoditas domestik. Jenis pohon penghasil kayu ini memiliki masa pertumbuhan yang cepat yakni kurang lebih dalam kurun waktu 7 hingga 15 tahun, pohon mahoni sudah tumbuh besar dan sudah bisa dipotong dan diambil kayunya [CHA-10].

Dalam dunia industri, kayu mahoni dikenal sebagai jenis kayu pertukangan yang baik. Kayu ini dapat dengan mudah dikerjakan seperti dipotong dan dibentuk sehingga kayu mahoni banyak digunakan sebagai bahan baku dalam pembuatan berbagai macam kerajinan dan produk-produk mebel seperti kerajinan tangan, aneka mebel ukiran, lemari, kursi, meja, furniture anak dll.

2.1.4 Tanaman Tumpangsari

Sistem tumpangsari adalah sistem bercocok tanaman hutan dimana di antara tanaman pokok kehutanan ditanami jenis tanaman pangan seperti palawija. Tumpangsari ditunjukan untuk memanfaatkan lingkungan sebaik-baiknya agar diperoleh produksi yang maksimum. Sistem tumpangsari dapat diatur berdasarkan sifat-sifat perakaran dan waktu penanaman [DEP-99]. Dengan pola tanam ini, akan dihasilkan lebih dari satu jenis panen dalam waktu yang bersamaan atau hampir bersamaan. Lebih dari satu hasil panen yang dihasilkan dalam satu waktu merupakan alah satu efisiensi produksi dalam kaitannya dengan waktu. Dalam kaitannya dengan ruang, pada pola tanam tumpang sari, masih ada *space* yang kosong pada jarak tanam tanaman dengan *habitus* tinggi seperti palawija atau tanaman tahunan yang lainnya. Ruang kosong itu yang dimanfaatkan untuk pertanaman tanaman yang lain sehingga penggunaan lahan lebih efisien.

2.2 Perangkat Lunak

Banyak orang menyamakan istilah perangkat lunak dengan program komputer. Sesungguhnya, pandangan ini terlalu dangkal. Perangkat lunak tidak hanya mencakup program, tetapi juga semua dokumentasi dan konfigurasi data yang berhubungan, yang diperlukan untuk membuat agar program beroperasi dengan benar. Sistem perangkat lunak biasanya terdiri dari sejumlah program yang terpisah, *file – file* konfigurasi yang digunakan untuk membuat program – program ini, dokumentasi sistem yang mendeskripsikan struktur sistem dan dokumentasi *user* yang menjelaskan bagaimana penggunaan sistem tersebut, dan untuk produk – produk perangkat lunak disediakan situs *web*, agar user bisa *men-download* informasi produk terbaru [SOM-03:5].

Perekayasa perangkat lunak bertugas mengembangkan produk perangkat lunak, yaitu perangkat lunak yang dapat dijual ke pelanggan. Secara umum ada dua tipe produk perangkat lunak, yaitu produk generik dan produk pesanan.

2.2.1 Produk generik

Produk generik merupakan sistem *stand-alone* (berdiri sendiri) standar yang diproduksi oleh organisasi pengembang dan dijual pada pasar terbuka ke siapapun yang bisa membelinya. Kadangkala perangkat lunak ini disebut sebagai perangkat lunak *shrink-wrapped* (dikecilkan dan dikemas). Contoh jenis produk ini adalah *database*, pengolah kata (*word processor*), paket untuk menggambar dan alat bantu manajemen proyek.

2.2.2 Produk pesanan

Produk pesanan (yang disesuaikan) merupakan sistem – sistem yang dipesan oleh pelanggan tertentu. Perangkat lunak dikembangkan khusus bagi pelanggan tersebut oleh kontraktor perangkat lunak. Beberapa contoh perangkat lunak jenis ini adalah sistem kontrol untuk piranti elektronik, sistem yang ditulis untuk mendukung proses tertentu dan sistem kontrol lalu lintas udara.

Perbedaan penting antara tipe – tipe perangkat lunak ini adalah pada produk generik organisasi yang mengembangkan perangkat lunak mengontrol spesifikasi perangkat lunak, sedangkan untuk produk pesanan spesifikasi biasanya dikembangkan dan dikontrol oleh organisasi yang membeli perangkat lunak tersebut. Pengembang perangkat lunak harus bekerja sesuai dengan spesifikasi tersebut.

2.3 Rekayasa Perangkat Lunak

Ian Sommerville dalam bukunya yang berjudul *Software Engineering* menyebutkan bahwa rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan.

Pada definisi ini, ada dua istilah kunci yaitu ‘*disiplin rekayasa*’ dan ‘*semua aspek produksi perangkat lunak*’.

1. *‘disiplin rekayasa’*

Perekayasa membuat suatu alat bekerja. Mereka menerapkan teori, metode, dan alat bantu yang sesuai, selain itu mereka menggunakannya dengan selektif dan selalu mencoba mencari solusi terhadap permasalahan, walaupun tidak ada teori atau metode yang mendukung. Perekayasa juga menyadari bahwa mereka harus bekerja dalam batasan organisasi dan keuangan, sehingga mereka berusaha mencari solusi dalam batasan – batasan ini.

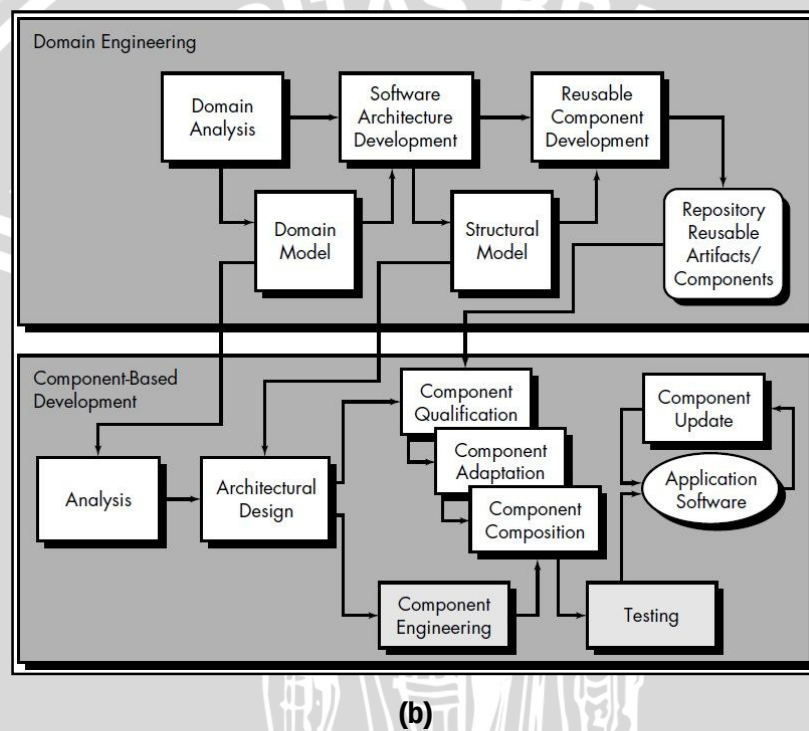
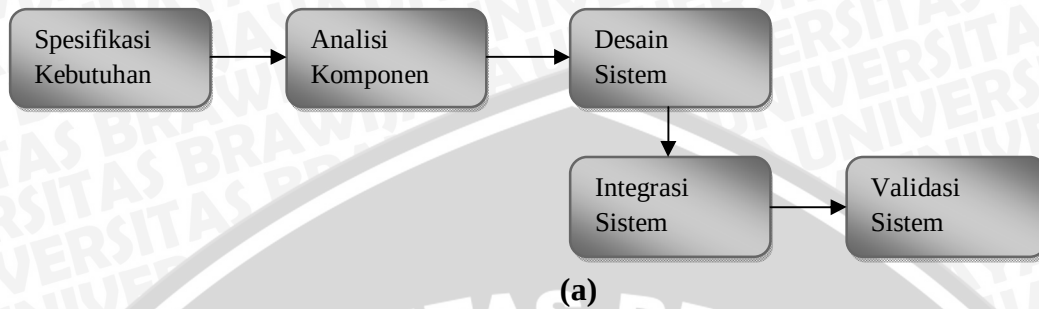
2. *‘semua aspek produksi perangkat lunak’*

Rekayasa perangkat lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak tetapi juga dengan kegiatan seperti manajemen proyek perangkat lunak dan pengembangan alat bantu, metode dan teori untuk mendukung produksi perangkat lunak.

Secara umum, perekayasa perangkat lunak memakai pendekatan yang sistematis dan terorganisir terhadap pekerjaan mereka karena cara ini seringkali paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Namun demikian, rekayasa ini sebenarnya mencakup masalah pemilihan metode yang paling sesuai untuk satu set keadaan dan pendekatan yang lebih kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan [SOM-03:7].

2.4 ***Component-Based Software Engineering***

Component-Based Software Engineering (CBSE) adalah metode pengembangan perangkat lunak yang menekankan pada perancangan dan pembangunan perangkat lunak dengan menggunakan komponen perangkat lunak yang sudah ada dan bersifat dapat digunakan kembali[PRE-01:721]. CBSE memiliki *software process* yang terdiri dari dua bagian utama yang berjalan paralel, yaitu *domain engineering* dan *component-based development*. Alur *software process* dari CBSE digambarkan pada Gambar 2.1.



Gambar 2.1 a dan b Software process dari CBSE

Sumber : [PRE-01:725]

1. Domain Engineering

Domain Engineering bertujuan untuk mengidentifikasi, membangun, mengumpulkan, dan mengalokasikan sekumpulan komponen perangkat lunak yang dapat dipakai pada pengembangan perangkat lunak pada saat ini dan di masa yang akan datang.

2. Domain Analysis

Pada tahap *domain analysis* akan dilakukan penggalan informasi terkait perangkat lunak yang akan dikembangkan melalui metode analisis kebutuhan sesuai dengan teknik *software analysis*.

3. Domain Model

Dalam *Domain Model* akan dilakukan pemodelan terhadap hasil *domain analysis* yang telah dilakukan sebelumnya. Pemodelan dapat dilakukan dengan *unified modelling language* untuk analisis yang bersifat *object-oriented*.

4. Software Architecture Development

Software architecture development merupakan tahap dimana terjadi perancangan pola arsitektur perangkat lunak yang nantinya akan dipakai pada pengembangan perangkat lunak yang akan dibangun.

5. Structural Model

Structural model adalah tahapan dimana dilakukan analisis lebih lanjut tentang pemodelan struktural perangkat lunak berdasarkan perancangan pola arsitektur perangkat lunak yang telah dilakukan. Analisis ini umumnya dilakukan dengan melakukan *break down* modul – modul fungsional dari perangkat lunak yang akan dibangun.

6. Reusable Components Development

Pada tahap *reusable component development* dilakukan proses identifikasi dan pencarian kebutuhan *software component* yang akan digunakan dalam membangun perangkat lunak.

7. Repository Reusable Artifacts / Components

Repository reusable artifacts / components adalah tahapan dimana komponen – komponen perangkat lunak dikumpulkan dan disimpan dalam bentuk *repository* sehingga siap diimplementasikan pada bagian *component-based development* atau dapat digunakan kembali di masa yang akan datang.

8. Component-Based Development

Component-based development bertujuan untuk melakukan pembangunan sistem perangkat lunak dengan menggunakan komponen – komponen perangkat lunak yang bersifat *reusable*.

9. Analysis

Pada tahap *analysis* ini akan dilakukan analisis kebutuhan lebih lanjut terkait dengan kebutuhan komponen – komponen yang dibutuhkan untuk membangun perangkat lunak berdasarkan analisis kebutuhan fungsional.

10. Architectural Design

Tahap *architectural design* merupakan tahapan perancangan arsitektural dari perangkat lunak yang akan dibangun. Perancangan arsitektural ini dilakukan berdasarkan pemodelan kebutuhanfungsionalitas perangkat lunak dan kebutuhan komponen yang telah dilakukan. Dalam perancangan arsitektural ini akan dijelaskan arsitektur perangkat lunak yang akan dibangun, seperti arsitektur penyimpanan data dan arsitektur *class* terkait fungsionalitas perangkat lunak. Umumnya, perancangan arsitektural ini dilakukan dengan *unified modelling language* apabila menggunakan *object-oriented analysis*.

11. Component Qualification

Pada tahap *component qualification* akan dijelaskan tentang kualifikasi dari komponen perangkat keras dan komponen perangkat lunak yang dibutuhkan dalam

pengembangan perangkat lunak yang akan dibangun. Tahap kualifikasi komponen ini bertujuan untuk mengetahui apakah komponen yang digunakan telah sesuai dan memenuhi syarat - syarat dari kebutuhan perangkat lunak yang dibangun.

12. Component Adaptation

Dalam tahap *component adaptation* dilakukan proses adaptasi dan modifikasi dari komponen – komponen perangkat lunak yang dipakai sehingga memenuhi kualifikasi kebutuhan komponen dari perangkat lunak yang dibangun.

13. Component Engineering

Tahap *component engineering* adalah tahap dimana komponen – komponen dari perangkat lunak yang dibangun, dikembangkan secara mandiri. Pengembangan komponen baru ini umumnya dilakukan dengan teknik *object-oriented programming*.

14. Component Composition

Tahap *component composition* adalah tahap utama dimana dilakukan pembangunan perangkat lunak dengan menggunakan komponen- komponen yang sudah ada. Pada tahap ini dilakukan penggabungan komponen – komponen perangkat lunak menjadi satu perangkat lunak yang padu.

15. Testing

Tahap *testing* adalah tahap dilakukannya proses pengujian perangkat lunak yang dibangun menggunakan teknik dan metode pengujian perangkat lunak tertentu. Pengujian dilakukan untuk mengetahui apakah fungsionalitas perangkat lunak sudah benar dan dapat memenuhi kebutuhan pengguna.

16. Application Software

Proses distribusi perangkat lunak dilakukan pada tahap *application software*. Perangkat lunak yang telah jadi dan telah diuji didistribusikan kepada pengguna atau klien untuk dapat dipakai sebagaimana mestinya.

17. Component Update

Pada tahap ini dilakukan pembaharuan (*update*) komponen perangkat lunak yang telah dibangun untuk meningkatkan kinerja dari perangkat lunak tersebut. Proses tersebut juga termasuk *maintenance* perangkat lunak yang telah didistribusikan.

2.5 Unified Modelling Language

Unified Modelling Language(UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML menggunakan *class* dan *operation* dalam konsep dasarnya, oleh sebab itu lebih cocok untuk penulisan perangkat lunak dalam bahasa – bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk *modeling* aplikasi prosedural dalam VB atau C [DHA-06:2].

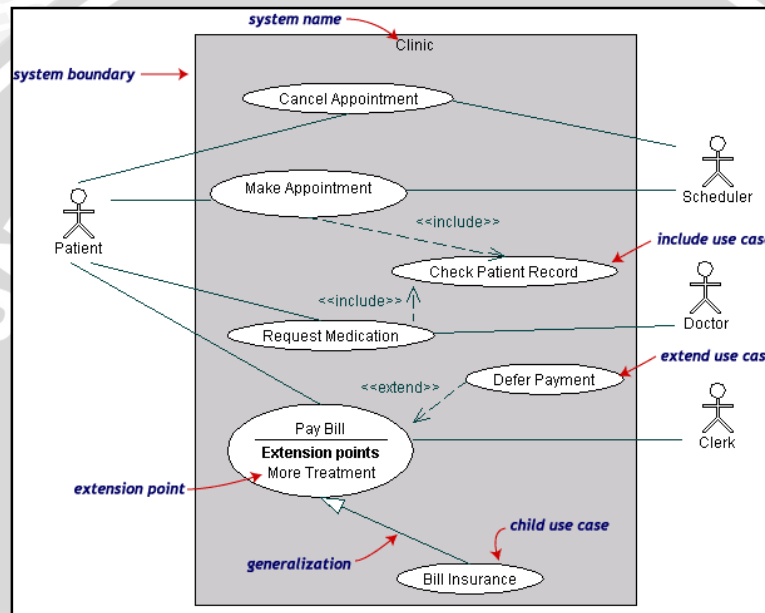
Dalam UML pemodelan digambarkan dengan penggunaan diagram – diagram yang masing – masing memiliki fungsi dan makna sendiri – sendiri. Diagram – diagram ini dapat digunakan sesuai dengan kebutuhan saat kita akan merancang perangkat lunak. Beberapa diantaranya adalah diagram *use case*, diagram *class*, diagram aktifitas, dan diagram *sequence*.

2.5.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan

“bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem [DHA-06:4].

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

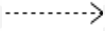
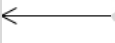
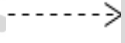


Gambar 2.2Contoh *use case diagram*

Sumber : [DHA-06:5]

Tabel 2.1 Keterangan simbol - simbol *use case diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya sebagai elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya (objek induk / <i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>Sistem</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.

9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Sumber : [DHA-06:5]

2.5.2 Class Diagram

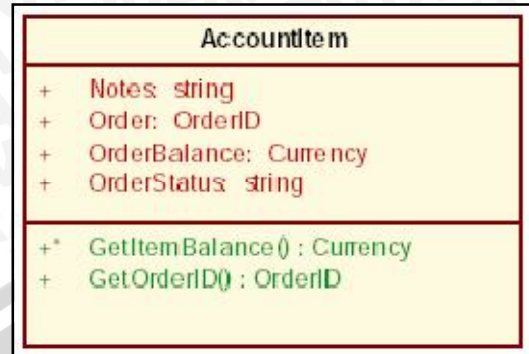
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah *object* dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain [DHA-06:5].

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

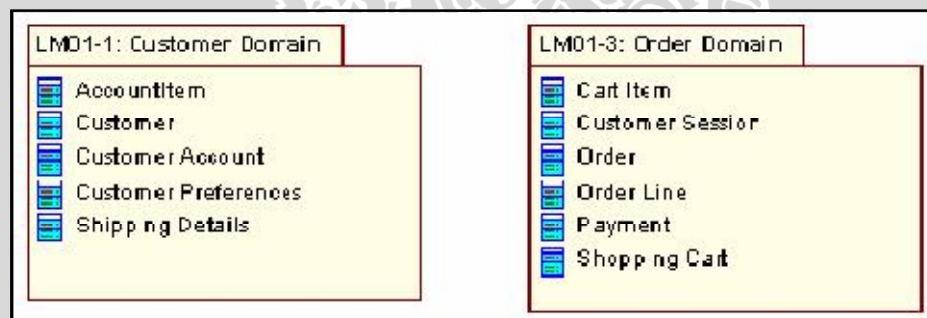
1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak - anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja



Gambar 2.3Contoh sebuah *class*

Sumber : [DHA-06:5]

Sesuai dengan perkembangan *class model*, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.



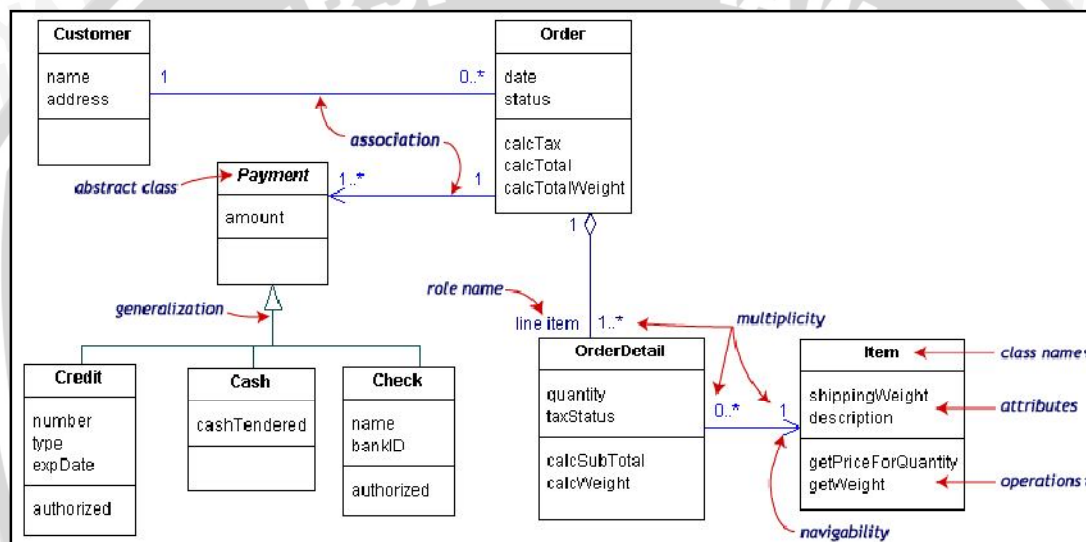
Gambar 2.4Contoh *package*

Sumber : [DHA-06:6]

Dalam penggunaan metode *object-oriented* dapat dipastikan *class – class* memiliki hubungan – hubungan tertentu sesuai dengan fungsinya sendiri – sendiri. Hubungan antar *class* ada beberapa, yaitu :

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).

3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

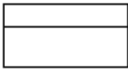
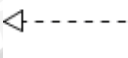
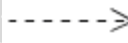


Gambar 2.5 Contoh class diagram

Sumber : [DHA-06:6]

Tabel 2.2 Keterangan simbol - simbol class diagram

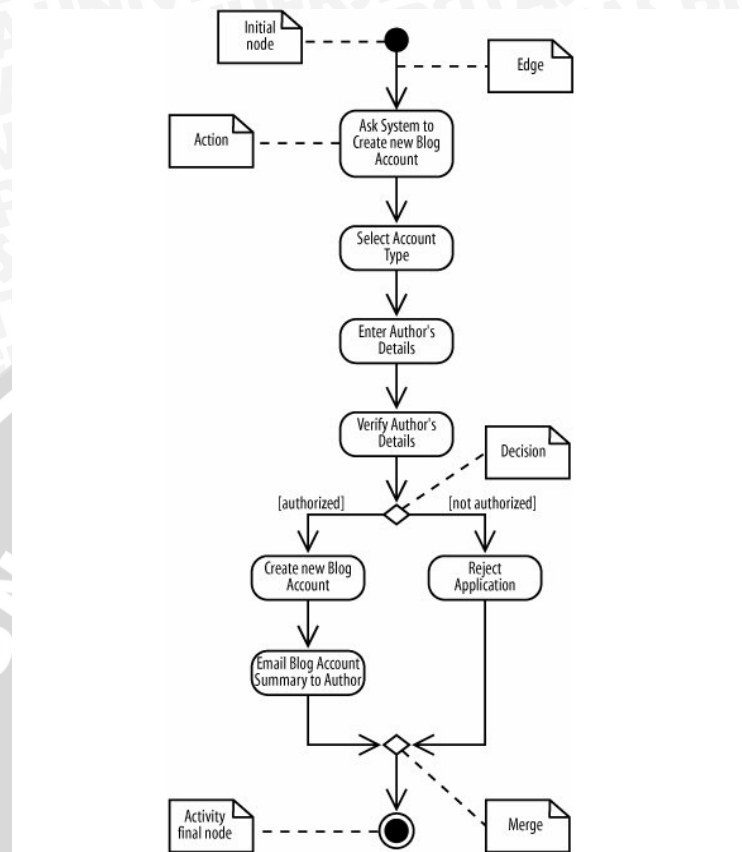
NO	GAMBAR	NAMA	KETERANGAN
1		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya (objek induk / <i>ancestor</i>).
2		N-ary Association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya atau elemen yang tidak mandiri.
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
8	1..*, 0...*, 1...1	<i>Multiplicity</i>	Menunjukkan jumlah objek yang diperbolehkan menurut logika.

Sumber : [DHA-06:6]

2.5.3 Activity Diagram

Tahap perancangan selanjutnya adalah pembuatan diagram aktivitas. Dengan diagram aktivitas memungkinkan untuk menentukan bagaimana sistem akan mencapai tujuannya [DHA-06]. Dalam perancangan perangkat lunak Pemetaan Lahan Hutan milik Perum Perhutani berbasis Android, diagram aktivitas digunakan untuk menggambarkan aliran aktivitas atau proses dari setiap use case dalam perangkat lunak Pemetaan Lahan Hutan milik Perum Perhutani berbasis Android. Gambar 2.8 menunjukkan contoh dari diagram aktivitas proses pembuatan akun blog.



Gambar 2.6Contoh activity diagram

Sumber : [DHA-06]

2.6 Pengujian Perangkat Lunak

Pengujian (*testing*) arsitektur dari perangkat lunak berorientasi objek menghasilkan sekumpulan *layered subSistem* s yang mengenkapsulasi kelas-kelas yang berkolaborasi. Setiap elemen sistem (subsistem dan *class*) melakukan fungsi yang membantu untuk mencapai kebutuhan sistem. Hal ini sangat penting untuk menguji sebuah *object-oriented Sistem* pada berbagai macam level yang berbeda dalam sebuah usaha untuk menemukan kesalahan-kesalahan yang mungkin terjadi dari kolaborasi kelas-kelas dan komunikasi subsistem melewati *architetural layer* [PRE-01:631].

2.6.1 Teknik Pengujian *Black-Box Testing*

Pengujian perangkat lunak memerlukan perancangan kasus uji (*test case*) agar dapat menemukan kesalahan dalam waktu singkat dan usaha minimum. Berbagai macam metode perancangan kasus uji telah berevolusi. Metode-metode ini menyediakan *developer* pendekatan sistematis untuk pengujian[PRE-01:443]. Teknik atau metode perancangan kasus uji yang digunakan adalah *black-box testing*. *Black-box testing* atau *behavioral testing* berfokus pada persyaratan fungsional perangkat lunak [PRE-01:459]. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black-box* merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan.

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori berikut :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi.

Pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi.

2.6.2 Strategi Pengujian

Strategi untuk pengujian perangkat lunak mengintegrasikan metode desain *test case* perangkat lunak ke dalam sederetan langkah yang direncanakan dengan baik, dan hasilnya adalah konstruksi perangkat lunak yang berhasil [PRE-01:477]. Sejumlah strategi pengujian perangkat lunak telah diusulkan di dalam literatur.

Strategi pengujian harus mengakomodasi pengujian tingkat rendah yang diperlukan untuk membuktikan bahwa segmen kode sumber yang kecil telah diimplementasikan dengan tepat, demikian juga pengujian tingkat tinggi yang memvalidasi fungsi-fungsi sistem mayor yang berlawanan dengan kebutuhan pelanggan. Proses pengujian dimulai dengan pengujian yang berfokus pada setiap modul secara individual (*unit testing*), dilanjutkan dengan pengujian integrasi (*integration testing*) dan berakhir pada pengujian validasi (*validation testing*) [PRE-01:481].

2.6.2.1 Pengujian Validasi

Validasi dapat ditentukan dengan berbagai cara, tetapi definisi yang sederhana adalah bahwa validasi berhasil bila perangkat lunak berfungsi dengan cara yang dapat diharapkan secara bertanggung jawab oleh pelanggan. Validasi perangkat lunak dicapai melalui sederetan pengujian *black-box* yang memperlihatkan konformitas dengan persyaratan. Rencana pengujian menguraikan kelas-kelas pengujian yang akan dilakukan, dan prosedur pengujian menentukan *test case* spesifik yang akan digunakan untuk mengungkap kesalahan dalam konformitas dengan persyaratan[PRE-01:495].

2.6.2.2 Pengujian Performa

Setelah semua langkah pengujian perangkat lunak secara terstruktur dilakukan, maka perlu dilakukan pengujian sistem di lingkungan dimana dia bekerja untuk mengetahui performa dari perangkat lunak tersebut. Pengujian sistem dirancang untuk menguji kinerja *run-time* dari perangkat lunak dalam konteks sistem terintegrasi. Pengujian performa melibatkan *monitoring* pemanfaatan sumber daya dari perangkat lunak yang diuji seperti perangkat lunak pendukung dan perangkat keras. Pengujian performa dilakukan secara spesifik sesuai dengan tipe perangkat lunak yang diuji. Pengujian performa bertujuan untuk mengungkap situasi yang menyebabkan degradasi dan kemungkinan kegagalan sistem [PRE-01:498].

2.7 Bahasa Pemrograman

2.7.1 Bahasa Pemrograman HTML dan CSS

Hypertext Markup Language (HTML) adalah sekumpulan *text* atau *file ASCII* yang berisi instruksi atau perintah program untuk web *browser* untuk menampilkan tampilan grafis sebuah halaman website. *File HTML* dapat dibuat menggunakan aplikasi *text editor* pada semua sistem operasi, contohnya adalah *Notepad* di Windows atau *simple text* di Macintosh. HTML merupakan cross platform karena walau pembuatannya menggunakan sistem operasi tertentu, namun akan memiliki tampilan yang sama di semua sistem operasi [SUG-07].

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Sama halnya *styles* dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa *file*. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda. Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML [SUG-07].

2.7.2 Bahasa Pemrograman PHP

Sistem kerja dari PHP (*Personal Home Page*) diawali dengan permintaan yang berasal dari halaman website oleh browser. Berdasarkan URL atau alamat website dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*. Selanjutnya *webserver* akan mencari berkas yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya.

Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke *webserver*, *webserver* akan memeriksa tipe file yang diminta

user. Jika tipe *file* yang diminta adalah PHP, maka akan memeriksa isi *script* dari halaman PHP tersebut. Apabila dalam *file* tersebut tidak mengandung *script* PHP, permintaan *user* akan langsung ditampilkan ke *browser*, namun jika dalam *file* tersebut mengandung *script* PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan *script-script* PHP dan mengolah *script* tersebut, sehingga dapat dikonversikan ke kode-kode HTML dan ditampilkan ke browser *user* [PER-13:4].

2.7.3 Bahasa Pemrograman Java

Java adalah bahasa pemrograman serbaguna. Java dapat digunakan untuk membuat suatu program sebagaimana program yang dibuat dengan bahasa lain seperti Pascal atau C++. Yang lebih menarik, Java juga mendukung sumber daya internet yang saat ini sangat populer, yaitu *world wide web* atau yang sering disebut sebagai *web* saja. Java juga mendukung aplikasi klien/server, baik dalam jaringan lokal (LAN) maupun jaringan berskala luas (WAN).

Dalam sejumlah literatur disebutkan bahwa Java merupakan hasil perpaduan sifat dari sejumlah bahasa pemrograman, yaitu C, C++, *Object-C*, *SmallTalk* dan *Common LISP*. Selain itu Java juga dilengkapi dengan unsur keamanan. Yang tak kalah penting adalah bahwa Java menambahkan paradigma pemrograman yang sederhana. Misalnya dalam bahasa C atau C++ yang menggunakan *pointer*, namun Java justru meninggalkannya karena sistem *pointer* dapat dikatakan sangat rumit, sehingga Java menjadi lebih mudah digunakan [KAD-09:2].

2.8 Basis Data

2.8.1 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management Sistem*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL memiliki beberapa keistimewaan, antara lain :

1. **Portabilitas.** MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server , Solaris, Amiga, dan masih banyak lagi.
2. **Perangkat lunak sumber terbuka.** MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. **Multi-user.** MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. **'Performance tuning'**, MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. **Ragam tipe data.** MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed/unsignedinteger, float, double, char, text, date, timestamp*, dan lain-lain.
6. **Perintah dan Fungsi.** MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam perintah (*query*).
7. **Keamanan.** MySQL memiliki beberapa lapisan keamanan seperti *levelsubnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. **Skalabilitas dan Pembatasan.** MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. **Konektivitas.** MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix *socket* (UNIX), atau *Named Pipes* (NT).
10. **Lokalisasi.** MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. **Antarmuka.** MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*ApplicationProgrammingInterface*).
12. **Klien dan Peralatan.** MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
13. **Struktur tabel.** MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.8.2 SQLite

SQLite adalah paket aplikasi yang menyediakan sistem database relational (RDBMS), sebagaimana vendor RDBMS lainnya seperti Oracle, MySQL, PostgreSQL dll. Kata *Lite* bukan berarti aplikasi RDBMS ini memiliki kemampuan yang sedikit / minim, tetapi mengacu pada keringanan/kemudahan dalam setup (instalasi), administrasi dan penggunaan.

Secara keseluruhan, SQLite menyediakan *database* relasional yang fungsional dan fleksibel dengan meminimalkan sumber daya dan kerumitan untuk pengembang dan pengguna [JAY-10:23].

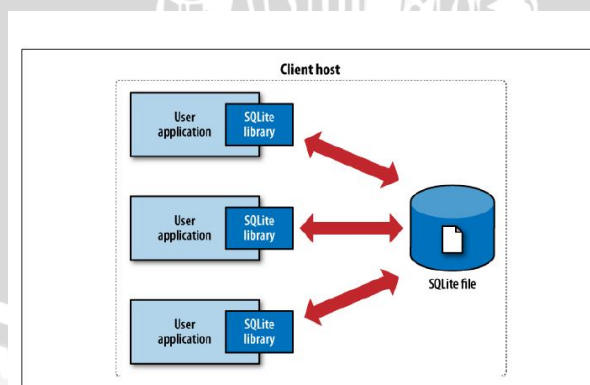


Figure 1-2. The SQLite server-less architecture.

Gambar 2.7 Contoh SQLite

Sumber: [JAY-10:25]

2.9 Google Maps Api

Google Maps adalah layanan aplikasi peta online yang disediakan oleh Google secara gratis. Layanan peta Google Maps secara resmi dapat diakses melalui situs <http://maps.google.com>. Google Maps mempunyai banyak fasilitas yang dapat dipergunakan misalnya pencarian lokasi dengan memasukkan kata kunci, kata kunci yang dimaksud seperti nama tempat, kota, atau jalan, fasilitas lainnya yaitu perhitungan rute perjalanan dari satu tempat ke tempat lainnya.

API (*Application Programming Interface*) merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. Bahasa pemrograman yang digunakan oleh Google Maps yang terdiri dari HTML, Javascript dan AJAX serta XML, memungkinkan untuk menampilkan peta Google Maps di *website* lain.

Dengan menggunakan Google Maps API, Google Maps dapat ditampilkan pada *website* eksternal. Agar aplikasi Google Maps dapat muncul di *website* tertentu, diperlukan adanya API key. API key merupakan kode unik yang digenerasikan oleh google untuk suatu *website* tertentu, agar *server* Google Maps dapat mengenali [AMR-11].

2.10 Sistem Operasi Android

Android adalah software untuk perangkat mobile yang mencakup sistem operasi, middleware dan aplikasi kunci. SDK (*Software Development Kit*) Android menyediakan alat dan API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada platform Android dengan menggunakan bahasa pemrograman Java [HER-11]. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam *mobile devices*.

2.11 GPS

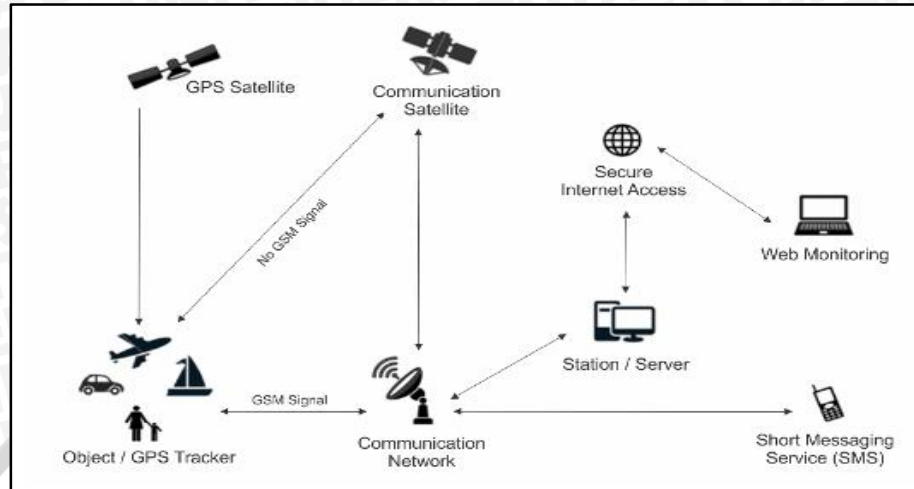
GPS (*Global Positioning Sistem*) adalah sistem navigasi yang berbasis satelit yang saling berhubungan yang berada di orbitnya. Setiap satelit mampu mengelilingi bumi hanya dalam waktu 12 jam. Sangat cepat, sehingga mereka selalu bisa menjangkau dimana pun posisi Anda di atas permukaan bumi. GPS *reciever* sendiri berisi beberapa *integrated circuit* (IC) sehingga murah dan teknologinya mudah untuk digunakan oleh semua orang. GPS dapat digunakan untuk berbagai kepentingan, misalnya mobil, kapal, pesawat terbang, pertanian dan diintegrasikan dengan komputer maupun laptop [OFF-13].

2.11.1 Tracking Sistem

Prinsip *Tracking* Sistem adalah dengan menggunakan teknologi GPS dan GSM. Sistem GPS adalah untuk mengumpulkan informasi lokasi yang akurat dari satelit, lalu bagian kedua adalah GSM untuk mengirim informasi lokasi tersebut sehingga dapat dilacak dari tempat lain. Informasi lokasi berupa koordinat yang lalu diterjemahkan ke dalam tampilan posisi pada peta digital.

Dengan *Real-Time GPS Tracking* Sistem , kita dapat duduk di PC atau komputer laptop, login ke sistem yang telah diinstall atau sistem *web-based* untuk memantau gerakan, arah dan kecepatan dari setiap kendaraan yang memiliki unit GPS *Tracking* Sistem. Selain itu, dengan *update hardware* dan program yang terus dikembangkan saat ini, anda juga dapat memanfaatkan beragam fitur seperti mematikan mesin, alarm bahaya, alarm kecepatan, sensor bahan bakar, sensor penumpang, komunikasi audio, serta data historis, laporan dan analisis.

Ada banyak aplikasi berbeda dan dikembangkan untuk memanfaatkan teknologi pelacakan dan membawa pengaruh besar bagi penelitian IPTEK, pertanian, pertahanan, keamanan, hingga usaha komersial [AMA-09].



Gambar 2.8 Tracking Sistem

Sumber: [AMA-09]

2.11.2 Asisted-Global Positioning Sistem (A-GPS)

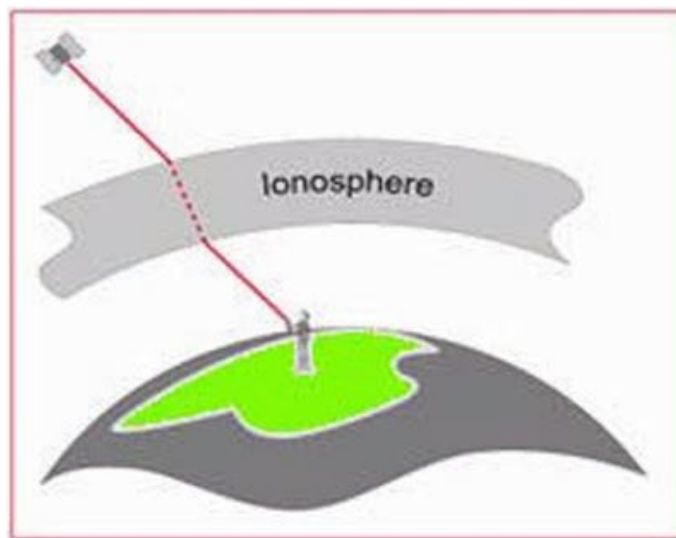
Frank Van Diggelen [DIG-09:1] mengatakan bahwa A-GPS merupakan penyempurnaan dari GPS sebagai satelit penentu posisi di belahan bumi. A-GPS jauh lebih efisien dan efektif dalam mengakses informasi dari satelit karena tidak perlu mencari data satu persatu dari ke-24 satelit yang ada, namun A-GPS telah mengetahui sasaran (satelit) mana yang dibutuhkan atau dituju [ELR-06:125].

Pada sistim A-GPS, telepon genggam akan menangkap sinyal satelit yang lalu dikirimkan ke *server* penyedia layanan telepon, hasil perhitungan lokasi yang dilakukan oleh *server* dikirimkan kembali ke telepon genggam. Peta juga dapat dikirimkan oleh *server* tersebut atau sudah disimpan pada telepon genggam. Sistem ini hanya berfungsi bila jaringan telepon genggam mampu disediakan oleh pengguna. Proses perhitungan data dilakukan oleh *server* penyedia jaringan telepon, maka telepon genggam tidak memerlukan prosesor yang canggih [ELR-06:125-126].

2.11.3 Sumber Kesalahan Perhitungan Posisi GPS (Global Positioning Sistem)

1. Penundaan Inosphere dan Atmosphere

Sebagaimana sinyal yang dilewatkan melalui lapisan ionosfer akan mengalami perlambatan, pengaruhnya seperti cahaya yang dibelokkan pada suatu kaca penghalang. Penundaan atmosfer ini menyebabkan kesalahan dalam perhitungan pada kecepatan sinyal (dalam ruang hampa kecepatan cahaya tetap). Ionosfer tidak menyebabkan konstanta penundaan pada sinyal. Terdapat beberapa faktor yang mempengaruhi penundaan yang disebabkan oleh lapisan ionosfer.



Gambar 2.9 Rambatan gelombang dari lapisan ionosfer

Sumber: [OFF-13]

2. Ketinggian Satelit dan receiver

Sinyal dari elevasi satelit yang rendah akan lebih banyak dipengaruhi, dari pada sinyal yang berasal dari elevasi satelit yang lebih tinggi. Hal kedua akan menambah jarak yang harus dilalui sinyal ke atmosfer. Kepadatan lapisan ionosfer dipengaruhi oleh matahari. Pada malam hari, pengaruh lapisan ionosfer ini sangat kecil dan sinyal turun perlahan. Jumlah kepadatan ionosfer meningkat bervariasi sesuai dengan siklus

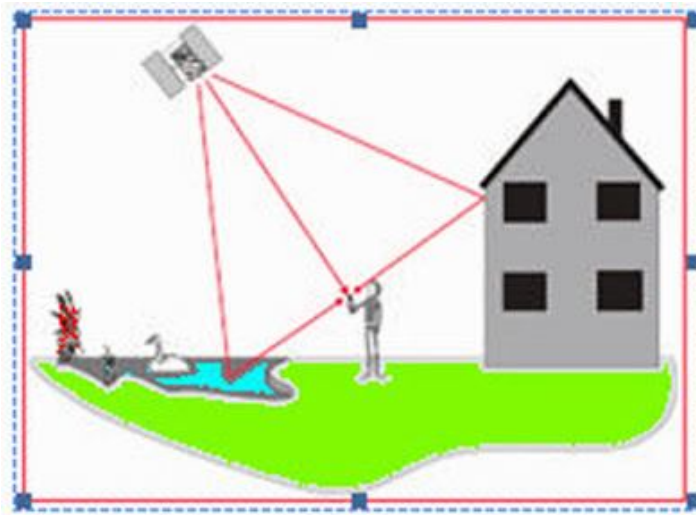
penyinaran (aktivitas matahari). Sebagai tambahan nyala api matahari terjadi secara acak dan juga mempunyai pengaruh pada kesalahan lapisan ionosfer.

3. Kesalahan *clock* Satelit dan *Receiver*

Sungguhpun *clock* dalam satelit akurat (sekitar 3 ns), kadang mengalami sedikit hanyutan dan menyebabkan sedikit kesalahan, yang dapat mempengaruhi ketelitian posisi. Departmen Pertahanan Amerika Serikat memonitor *clock* satelit menggunakan segmen kontrol dan hanyutan yang ditemukan dapat dibetulkan.

4. Kesalahan *Multipath*

Multipath terjadi bila posisi antenna penerima pada posisi terbuka pada permukaan refleksi yang sangat besar seperti danau atau bangunan. Sinyal satelit tidak berjalan langsung ke antenna namun membentur dahulu obyek yang ada didekatnya dan direfleksikan ke dalam antena yang menyebabkan kesalahan pengukuran.



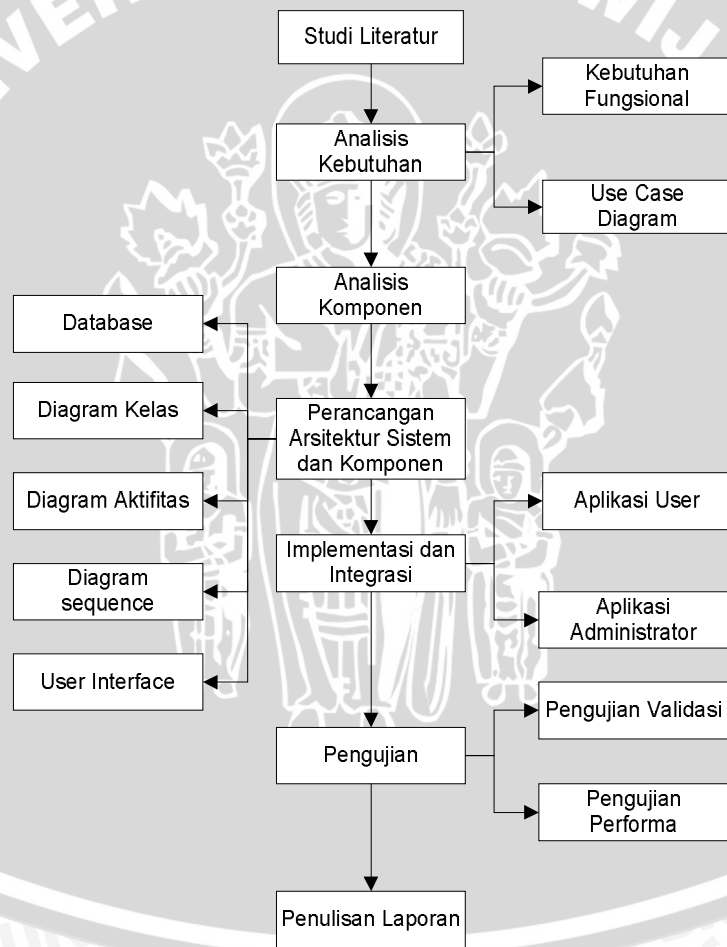
Gambar 2.10 Terjadinya benturan sinyal satelit pada obyek

Sumber: [OFF-13]

BAB III

METODE PENELITIAN

Pada bab ini dijelaskan langkah - langkah yang akan dilakukan dalam perancangan, implementasi dan pengujian dari aplikasi perangkat lunak yang akan dikembangkan. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan perangkat lunak selanjutnya.



Gambar 3.1: Metode Penelitian CBSE (*Component Based Software Engineering*)

Sumber : [Metode Penelitian]

Metodologi penelitian yang digunakan adalah adaptasi dari pengembangan model CBSE (mengacu pada Gambar 2.1). Adaptasi yang dilakukan adalah menggunakan tahap-tahap yang terdapat pada gambardi bawah ini. Diagram blok metode penelitian digambarkan pada Gambar 3.1.

3.1 Studi Literatur

Studi literatur atau *Domain Analysis* menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori - teori pendukung tersebut meliputi :

1. Hutan Produksi
2. Perangkat Lunak
3. Rekayasa Perangkat Lunak
4. *Component-Based Software Engineering*
5. *Unified Modelling Language*
6. Pengujian Perangkat Lunak
7. Bahasa Pemrograman
8. Basis Data
9. Google Maps Api
10. Sistem Operasi Android
11. GPS

3.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem perangkat lunak yang akan dibangun. Pada tahapan *Domain Model*, metode yang digunakan adalah *object-oriented analysis* dengan menggunakan bahasa pemodelan UML (*Unified Modeling Language*). Diagram *use case* digunakan untuk mendeskripsikan kebutuhan - kebutuhan dan fungsionalitas perangkat lunak dari perspektif *end-user*. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan (*requirements*) sistem perangkat lunak Pemetaan Lahan Hutan berbasis Android yang kemudian akan dimodelkan dalam diagram *use case*. Diagram

use case perangkat lunak Pemetaan Lahan Hutan berbasis Android terbagi menjadi 2 buah subsistem, yaitu subsistem *user* dan subsistem *administrator*. Tiap *use case* dalam diagram *use case* tersebut juga akan dijelaskan lebih rinci dalam skenario *use case*. Analisis penyimpanan data pada perangkat lunak Pemetaan Lahan Hutan berbasis Android juga dilakukan pada tahap ini.

3.3 Perancangan Sistem

Perancangan arsitektur sistem atau *Structural model* adalah tahap dimana penulis mulai merancang suatu sistem yang mampu memenuhi semua kebutuhan fungsional aplikasi dalam skripsi ini. Teori-teori dari tahap *domain analysis* lalu di modelkan melalui tahap *Domain Model* sehingga didapat diimplementasikan untuk merancang serta mengembangkan aplikasi Pemetaan Lahan Hutan berbasis Android. Perancangan Pemetaan Lahan Hutan berbasis Android terdiri dari beberapa tahapan yaitu: Perancangan dimulai dari perancangan struktur basis data berdasarkan analisis penyimpanan data yang telah dilakukan. Kemudian dilanjutkan dengan perancangan sistem perangkat lunak Pemetaan Lahan Hutan berbasis Android yang dilakukan dengan mengidentifikasi *class – class* dan *interface – interface* yang dibutuhkan, serta kemudian dimodelkan dalam diagram *class*. *Class* yang terdapat pada perangkat lunak ini menggunakan komponen-komponen yang sudah ada (*reusable*). Hubungan interaksi antar objek yang telah diidentifikasi, dimodelkan dalam *activity diagram* dan kemudian tahap perancangan dilanjutkan dengan perancangan antarmuka pengguna.

3.4 Implementasi

Implementasi perangkat lunak mengacu kepada perancangan perangkat lunak. Perancangan perangkat lunak digunakan sebagai dasar acuan implementasi perangkat lunak yang akan dilakukan. Implementasi perangkat lunak diawali dengan penjabaran spesifikasi lingkungan pengembangan perangkat lunak. Kemudian dilanjutkan dengan implementasi basis data dengan menggunakan MySQL dan SQLite.

Selanjutnya dijabarkan *mapping class* dengan *file* program saat implementasi perangkat lunak dan mencari komponen *class* mana saja yang sudah tersedia, seperti *class GPS service*. Pada tahapan ini, *class* yang sudah ada akan disesuaikan dengan *class* yang lainnya agar dapat memenuhi kebutuhan fungsional perangkat lunak. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman Java pada subsistem *user* dan pemrograman HTML, CSS, dan PHP pada subsistem *administrator*. Tahap terakhir dari implementasi adalah implementasi antarmuka berdasarkan perancangan yang telah dilakukan.

3.5 Pengujian dan Analisis

Pengujian perangkat lunak dilakukan untuk mengetahui apakah kinerja dan performa sistem perangkat lunak Pemetaan Lahan Hutan berbasis Android telah memenuhi spesifikasi kebutuhan yang melandasinya. Strategi pengujian perangkat lunak yang akan digunakan yaitu pengujian performa, dan pengujian validasi (*validation testing*). Metode pengujian yang akan digunakan adalah *black-box testing*. Proses pengujian perangkat lunak dimulai dari pengujian validasi, kemudian dilanjutkan dengan pengujian performa. Pada tahap pengujian validasi digunakan metode *black-box testing*. Analisis juga dilakukan untuk mengetahui hasil dari pengujian perangkat lunak sehingga dapat didapatkan kesimpulan dari pengembangan perangkat lunak yang telah dilakukan.

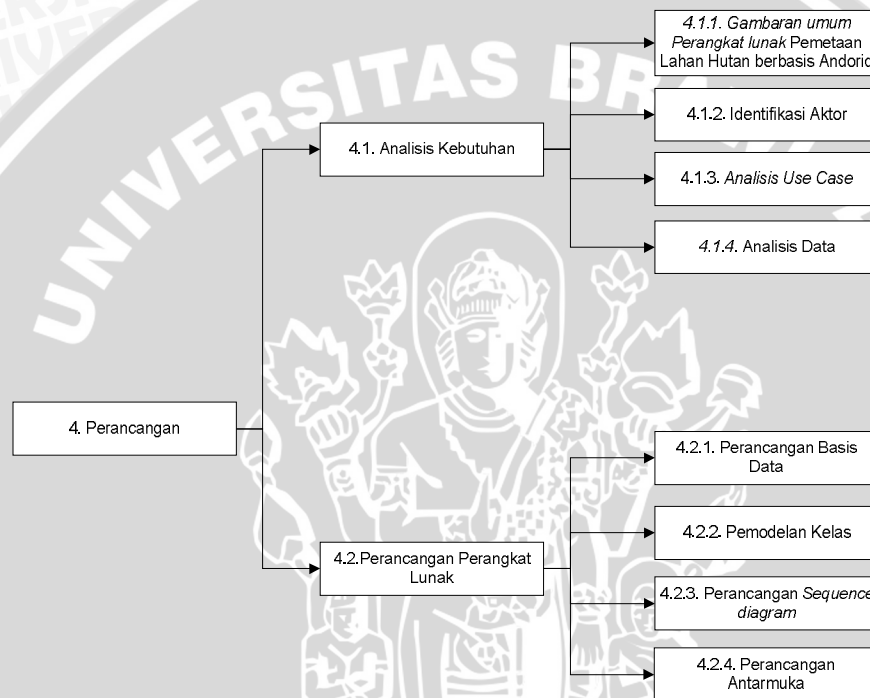
3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan perangkat lunak, implementasi perangkat lunak, dan pengujian perangkat lunak telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan – kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan perangkat lunak lebih lanjut.

BAB IV

PERANCANGAN

Bab ini membahas mengenai perancangan perangkat lunak Pemetaan Lahan Hutan berbasis Android. Perancangan dilakukan dengan model CBSE seperti yang telah dijelaskan di bab metodologi penelitian sub Bab 3.1.3. Diagram blok perancangan digambarkan pada Gambar 4.1.



Gambar 4.1 Diagram Blok Perancangan
Sumber : [Perancangan]

Tahap analisis kebutuhan terdiri atas empat langkah tahap. Tahap pertama melakukan penjabaran tentang gambaran umum perangkat lunak Pemetaan Lahan Hutan berbasis Android. Pada tahap kedua melakukan proses identifikasi aktor yang terlibat dalam sistem perangkat lunak. Tahap ketiga melakukan proses analisis data yang diperlukan. Tahap keempat melakukan analisis *use case* berdasarkan penjabaran gambaran umum perangkat lunak.

Desain sistem perangkat lunak memiliki empat tahap, yaitu perancangan basis data, pemodelan kelas untuk menggambarkan perancangan struktur *class-class* yang menyusun perangkat lunak Pemetaan Lahan Hutan berbasis Android, perancangan *activity diagram* dan perancangan antarmuka pengguna dari perangkat lunak Pemetaan Lahan Hutan berbasis Android. Integrasi sistem akan dimodelkan menggunakan *collaboration diagram* yang menggambarkan interaksi antar *object* di Pemetaan Lahan Hutan berbasis Android.

4.1 Analisis Kebutuhan

Proses analisis kebutuhan mengacu pada gambaran umum perangkat lunak Pemetaan Lahan Hutan berbasis Android dan hasil pengumpulan, pemahaman dan penetapan kebutuhan – kebutuhan yang ingin didapatkan oleh pengguna. Proses analisis kebutuhan ini diawali dengan penjabaran gambaran umum perangkat lunak Pemetaan Lahan Hutan berbasis Android, identifikasi aktor yang terlibat, analisis data yang akan disimpan, penjabaran tentang daftar kebutuhan dan kemudian memodelkannya ke dalam diagram *use case*. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan – kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

4.1.1 Gambaran Umum Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Pembahasan gambaran umum perangkat lunak Pemetaan Lahan Hutan berbasis Android terdiri atas dua bagian, yaitu deskripsi umum perangkat lunak Pemetaan Lahan Hutan berbasis Android dan cara penggunaan perangkat lunak Pemetaan Lahan Hutan berbasis Android.

4.1.2 Deskripsi Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Perangkat lunak ini adalah perangkat lunak yang berfungsi untuk pendataan dan pemetaan lahan hutan. Sistem yang ada saat ini yaitu dengan mencatat hasil

pemetaan pada sebuah buku, apabila ada tindakan reboisasi atau penebangan maka petugas terkait akan membuat data koordinat lahan dan keterangan kegiatan pada buku RPKH (Rencana Pengaturan Kelestarian Hutan) lalu memetakan pada sebuah peta yang telah disediakan. Buku RPKH tersebut juga berfungsi sebagai tempat untuk data warga sekitar yang ingin memanfaatkan lahan Perum Perhutani yang biasa disebut dengan tumpangsari.

Perangkat lunak yang akan dikembangkan dalam proyek skripsi ini adalah perangkat lunak Pemetaan Lahan Hutan berbasis Android. Perangkat lunak Pemetaan Lahan Hutan berbasis Android adalah perangkat lunak yang dapat digunakan untuk mendata lahan dan mencari data koordinat lahan hutan pada *mobile phone* bersistem operasi Android yang nantinya dikirim ke dalam *webservice* kantor Perum Perhutani KPH Malang BKPH Sumbermanjing dan dapat menampilkan bidang lahan/*polygon* pada *Google Maps Api*.

Perangkat lunak Pemetaan Lahan Hutan berbasis Android memiliki 2 bagian utama yaitu, subsistem aplikasi *User* dan subsistem aplikasi *Administrator*.

1. Aplikasi *User*

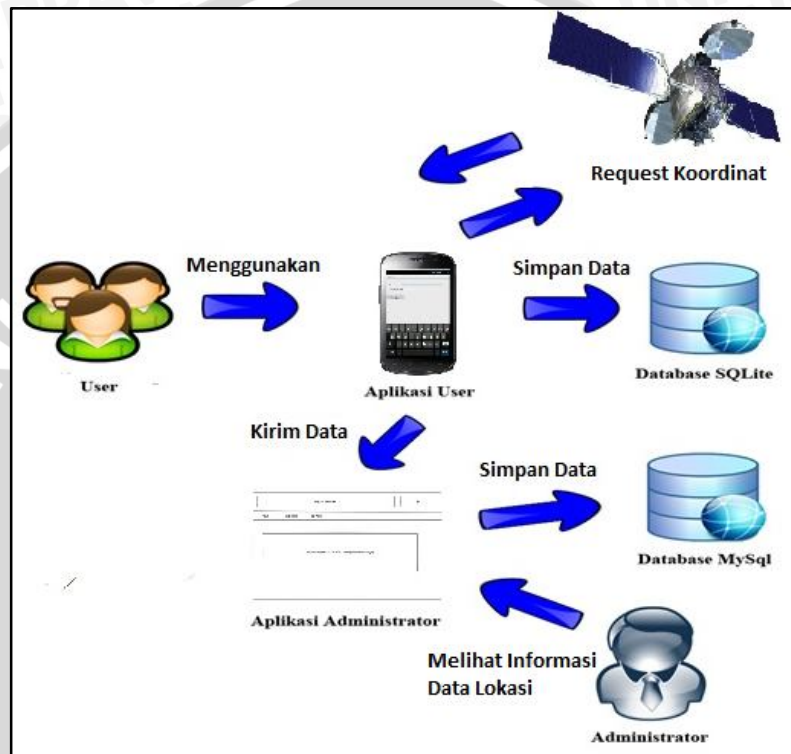
Aplikasi *User* adalah aplikasi Pemetaan Lahan Hutan berbasis Android yang dapat digunakan untuk mendata lahan dan mencari koordinat lahan klien pada *mobile phone* bersistem operasi Android yang nantinya dikirim ke *webservice* kantor Perum Perhutani KPH Malang BKPH Sumbermanjing dan dapat menampilkan bidang lahan/*polygon* pada *Google Maps Api*.

2. Aplikasi *Administrator*

Aplikasi *Administrator* dari perangkat lunak Pemetaan Lahan Hutan berbasis Android adalah aplikasi *web* yang digunakan untuk mencari dan melihat informasi data klien yang diperoleh dari aplikasi *user*.

4.1.3 Cara Penggunaan Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Perangkat lunak Pemetaan Lahan Hutan berbasis Android memiliki urutan langkah kerja seperti yang ditunjukkan pada Gambar 4.1.



Gambar 4.2 Urutan cara penggunaan perangkat lunak Pemetaan Lahan Hutan berbasis Android
Sumber : [Perancangan]

Pada perangkat lunak Pemetaan Lahan Hutan berbasis Android, data dan koordinat lokasi dikelola oleh user melalui aplikasi user. Kemudian, data dan koordinat lokasi yang dikelola disimpan pada *database SQLite*. Selanjutnya data dan koordinat klien tersebut dikirim ke aplikasi *administrator/webservice* kantor Perum Perhutani KPH Malang BKPH Sumbermanjing sebagai dasar menampilkan bidang/*polygon* lahan di dalam *google maps api* yang sudah ada dalam aplikasi *administrator*.

Perangkat lunak ini memiliki 8 proses utama yaitu: tambah data lokasi, tambah data detail tumpangsari, tambah data koordinat lokasi, ubah data lokasi, lihat

seluruh data lokasi, buat akun *user*, penentuan luas wilayah, dan penentuan waktu tebang. Berikut ini penjelasan dari masing - masing proses.

1. Tambah Data Lokasi

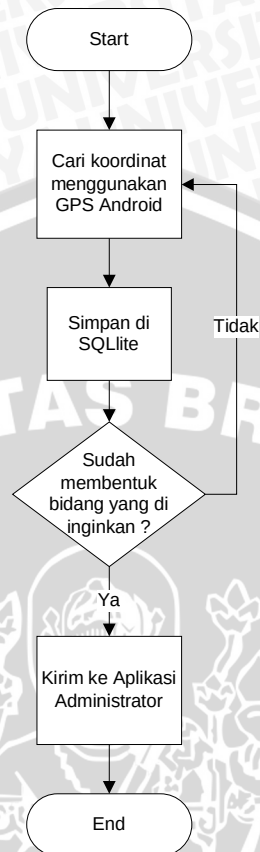
Pada proses ini, *user* dapat menambahkan data dari lokasi yang didapatkan oleh pegawai Perhutani di lapangan pada saat pendataan lahan. Data tersebut adalah nama lokasi, jenis tanaman, ketinggian dasar lokasi, bonita (kepadatan hutan), kepadatan kayu, kelashutan, bentuklapangan, dan waktu tanam lahan hutan.

2. Tambah Data Detail Tumpangsari

Pada proses ini, *user* dapat menambahkan data dari detail tumpangsari pada setiap data lokasi yang akan dipetakan. Data ini berfungsi untuk menjabarkan jenis tanaman dan waktu tanam tanaman tumpangsari apa yang ada pada lokasi tersebut.

3. Tambah Data Koordinat Lokasi

Pada proses ini, *user* dapat menambahkan data koordinat lokasi menggunakan GPS yang ada pada *smartphone* Android. Jadi setiap titik koordinat akan di catat oleh *smartphone* Android, apabila sudah membentuk bidang yang diinginkan, *user* dapat mengirimkan data tersebut ke aplikasi *administrator*. Berikut merupakan alur dari pencatatan data koordinat lokasi.



Gambar 4.3Diagram Alur Tambah Data Koordinat Lokasi
Sumber : [Perancangan]

4. Ubah Data Lokasi

Pada proses ini, *user* dapat mengubah data lokasi yang telah tersimpan pada *SQL lite* Android. Data tersebut berbentuk List sehingga memudahkan *user* untuk mengubah data yang diinginkan.

5. Lihat Seluruh Data Lokasi

Proses ini merupakan proses yang hanya bisa dilakukan *Administrator* yang ada di kantor Perhutani. *Administrator* dapat melihat seluruh data lokasi melalui aplikasi *administrator* yang berupa *website*. Data tersebut meliputi data lokasi, data tumpangsari, dan data koordinat yang telah berwujud bidang/ *polygon* lahan di dalam *google maps api*.

6. Buat Akun *User*

Buat Akun *User* merupakan proses yang hanya bisa dilakukan *Administrator* yang ada di kantor Perhutani. *Administrator* menginputkan nama lengkap, *username*, dan *password* yang nantinya dapat di gunakan oleh *user* di aplikasi *user*.

7. Penentuan Luas Wilayah

Pada proses ini, data koordinat yang telah dikirimkan oleh *user* akan di tampilkan pada aplikasi *administratordi* dalam *google maps api*. Inputan yang berupa titik – titik koordinat wilayah tersebut akan diolah oleh sistem sehingga didapat keluaran berupa Luas wilayah dari kumpulan titik – titik koordinat wilayah tersebut.

8. Penentuan Waktu Tebang

Pada proses ini, data jenis tanaman dan waktu tanam yang telah dikirimkan oleh *user* akan di proses oleh sistem berdasarkan jenis tanamannya. Jenis tanaman yang ada saat ini adalah Jati, Mahoni, dan Sengon. Pohon Jati membutuhkan waktu minimal 20 tahun, pohon Mahoni membutuhkan waktu minimal 15 tahun, dan pohon Sengon hanya memerlukan minimal 6 tahun untuk siap di panen / tebang. Sedangkan untuk tumpangsari, warga diberikan waktu hanya 2 tahun untuk memanfaatkan lahan hutan disela – sela tanaman pokok.

4.1.4 Identifikasi Aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor - aktor yang akan berinteraksi dengan perangkat lunak Pemetaan Lahan Hutan berbasis Android. Tabel 3.1 memperlihatkan aktor – aktor yang terlibat beserta penjelasannya masing-masing yang merupakan hasil dari proses identifikasi aktor.

Tabel 4.1Identifikasi aktor

Aktor	Deskripsi
User	<i>User</i> adalah sebagai pegawai/petugas lapangan di dalam kantor Perum PerhutaniKPH Malang BKPH Sumbermanjing yang hanya bisa menggunakan aplikasi <i>user</i> dari perangkat lunak Pemetaan Lahan Hutanberbasis Android.
Administrator	<i>Administrator</i> adalah pengguna aplikasi <i>administrator</i> yaitu <i>webservice</i> kantor Perum Perhutani KPH Malang BKPH Sumbermanjing yang memiliki hak untuk mengelola data dan koordinat klien dari perangkat lunak Pemetaan Lahan Hutanberbasis Androidlebih lanjut.

Sumber : [Perancangan]

4.1.5 Analisa Use Case

Analisis *use case* dibagi mejadi 2, yaitu daftar kebutuhan dan diagram *use case*. Pada daftar kebutuhan akan dibagi menjadi dua yaitu spesifikasi kebutuhan fungsional *user* dan kebutuhan fungsional *administrator*. Spesifikasi kebutuhan fungsional *user* ditunjukkan pada Tabel 4.2.

Tabel 4.2 Spesifikasi kebutuhan fungsional *user*

Nomor SRS	Kebutuhan	Use Case
SRS_001_01	<i>User</i> dapat <i>login</i> ke dalam sistem.	<i>Login</i> Klien
SRS_001_02	<i>User</i> dapat menambahkan informasi data lokasi.	Menambah Informasi Data Lokasi
SRS_001_03	<i>User</i> dapat mengubah informasi data lokasi.	Mengubah Informasi Data Lokasi
SRS_001_04	Sistem dapat mengirimkan	Mengirim Informasi

	informasi data lokasi yang ditentukan oleh <i>user</i> .	Data Lokasi
SRS_001_05	<i>User</i> dapat menambahkan informasi detail lokasi yang berupa koordinat gps.	Menambah Detail Lokasi
SRS_001_06	<i>User</i> dapat menambahkan informasi detail tumpangsari.	Menambah Detail Tumpangsari
SRS_001_07	<i>User</i> dapat <i>logout</i> dari sistem dan sistem dapat menghapus session dari <i>user</i> .	<i>Logout</i> Klien

Sumber : [Perancangan]

Spesifikasi kebutuhan fungsional *administrator* ditunjukkan pada Tabel 4.3.

Tabel 4.3 Spesifikasi kebutuhan fungsional *administrator*

Nomor SRS	Kebutuhan	Use Case
SRS_002_01	<i>Administrator</i> dapat <i>login</i> ke dalam sistem.	<i>Login Administrator</i>
SRS_002_02	<i>Administrator</i> dapat membuat akun klien baru.	Pembuatan Akun klien
SRS_002_03	<i>Administrator</i> dapat melihat semua informasi data klien.	Melihat Informasi Data klien
SRS_002_04	<i>Administrator</i> dapat mencari dan melihat semua informasi data lokasi.	Melihat Seluruh Informasi Data Lokasi
SRS_002_05	Sistem dapat menentukan Luas wilayah berdasarkan titik – titik koordinat wilayahnya.	Pembuatan Luas Wilayah

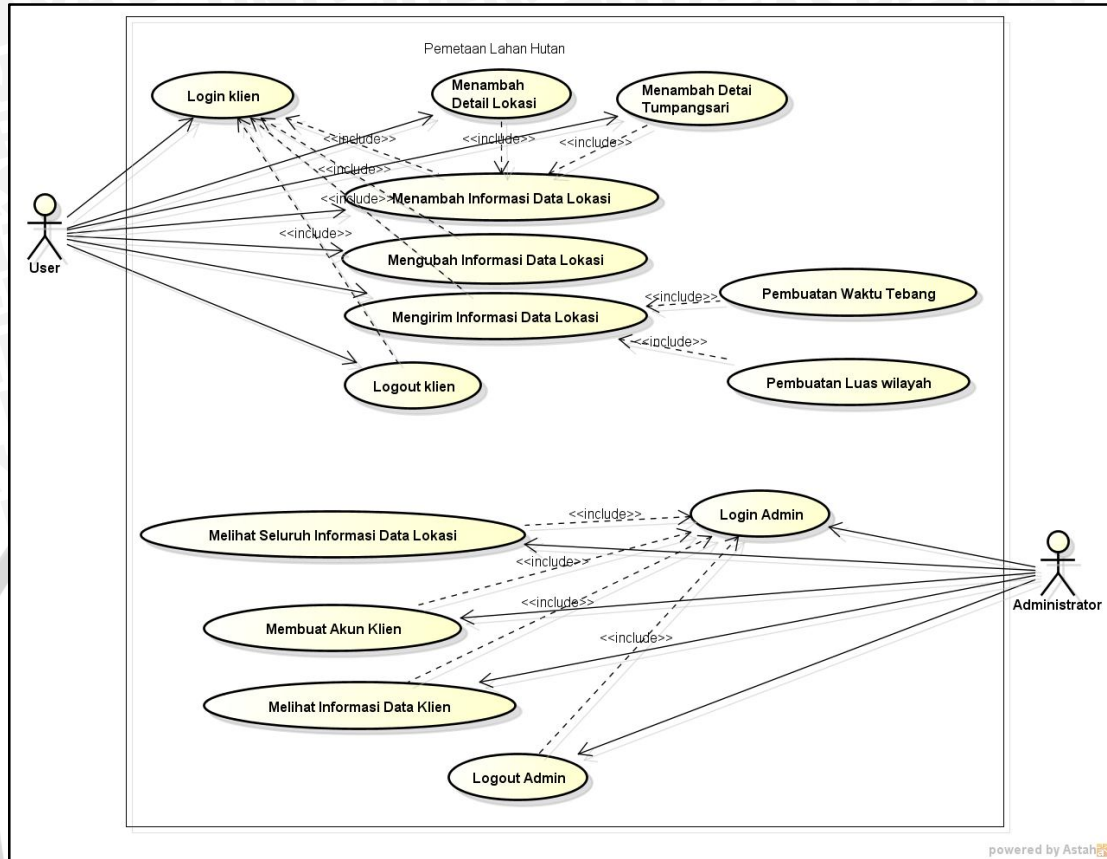
SRS_002_06	Sistem dapat menentukan waktu tebang dari data lokasi yang dikirim dari aplikasi <i>user</i> .	Pembuatan Waktu Tebang
SRS_002_07	<i>Administrator</i> dapat <i>logout</i> dari sistem dan sistem dapat menghapus session dari <i>administrator</i> .	<i>Logout Administrator</i>

Sumber : [Perancangan]

Diagram *use case* adalah salah satu diagram untuk memodelkan aspek perilaku sistem. Diagram *use case* menunjukkan sekumpulan *use case*, aktor, dan hubungannya. *Use case* merupakan fungsionalitas dari sistem yang diinisialisasi oleh aktor. Pemodelan diagram *use case* yang menggambarkan fungsionalitas perangkat lunak Pemetaan Lahan Hutan Berbasis Android dibagi menjadi dua yaitu diagram *use case* untuk aplikasi *user* dan diagram *use case* untuk aplikasi *administrator*.

4.1.5.1 Diagram Use Case Pemetaan Lahan Hutan Berbasis Android

Diagram *use case* ini melibatkan *user* dan *administrator* sebagai aktor dan 11 buah *use case*. 11 buah *use case* ini termasuk dalam bagian perangkat lunak Pemetaan Lahan Hutan Berbasis Android. Satu buah *use case* ini juga akan disertai dengan skenario *use case* untuk menjelaskan rangkaian aktivitas yang terjadi di *use case* tersebut. Diagram *use case* untuk aplikasi *user* ditunjukkan pada Gambar 4.3 berikut ini :



Gambar 4.4 Diagram use case
Sumber : [Perancangan]

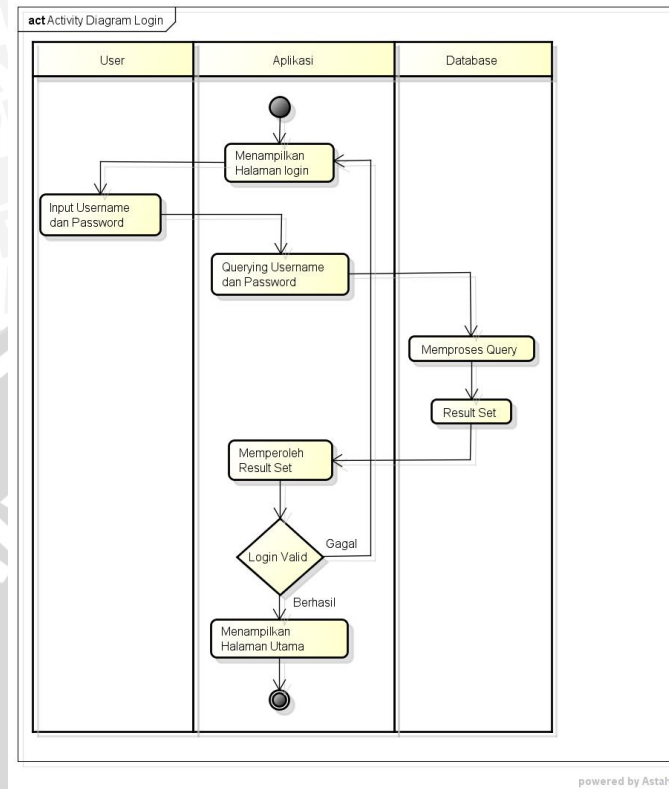
1. **Skenario Use Case Menunjukkan Login klien**

Tabel 4.4 Use case Menunjukkan Login klien

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_001_01
Nama	Login klien
Tujuan	Untuk menyeleksi user yang sah.
Deskripsi	Use case ini menjelaskan bagaimana user melakukan login untuk dapat menampilkan halaman utama user.
Aktor	User
Skenario Utama	

Kondisi Awal	Perangkat lunak memunculkan halaman <i>login</i> .
Aksi Aktor	Reaksi Sistem
User memasukkan data login(<i>username</i> dan <i>password</i>), lalu menekan tombol " <i>Login</i> ".	Perangkat lunak menerima data login dan kemudian aplikasi <i>administrator</i> melakukan pengecekan terhadap data yang dimasukkan oleh <i>user</i> . Jika data benar, maka akan ditampilkan halaman utama <i>user</i> .
Skenario Alternatif 1 : Jika <i>username</i> atau <i>password</i> kosong	
	Perangkat lunak akan menampilkan pesan peringatan bahwa <i>username</i> atau <i>password</i> kosong.
Skenario Alternatif 2 : Jika <i>username</i> dan <i>password</i> salah	
	Perangkat lunak akan menampilkan pesan peringatan bahwa <i>username</i> dan <i>password</i> salah.
Kondisi Akhir	Sistem akan menampilkan halaman utama dari aplikasi <i>user</i> .

Sumber : [Perancangan]



Gambar 4.5 Diagram aktivitas login
Sumber : [Perancangan]

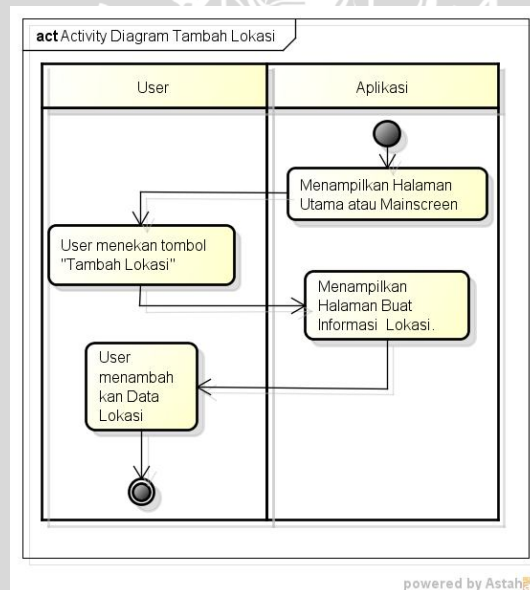
2. Skenario Use Case Menambah Informasi Data Lokasi di Database

Tabel 4.5 Use case Menambah Informasi Data Lokasi di Database

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_001_02
Nama	Menambah informasi data lokasi di database.
Tujuan	Untuk menambahkan informasi data lokasi.
Deskripsi	UseCase ini menjelaskan bagaimana user menambahkan informasi data tentang lokasi.
Aktor	User
Skenario Utama	
Kondisi Awal	User berhasil melakukan login dan masuk halaman utama user.

Aksi Aktor	Reaksi Sistem
User menekan tombol “Buat Lokasi” dan menambahkan data lokasi pada halaman tambah data lokasi.	Perangkat lunak menerima informasi data dan penambahan datalokasi yang akan dikirim ke aplikasi <i>administrator</i> .
Skenario Alternatif 1 : Jika <i>user</i> menekan tombol <i>back</i>	
	Perangkat lunak akan kembali ke halaman utama.
Kondisi Akhir	Informasi data lokasi berhasil ditambahkan pada aplikasi <i>user</i> .

Sumber : [Perancangan]



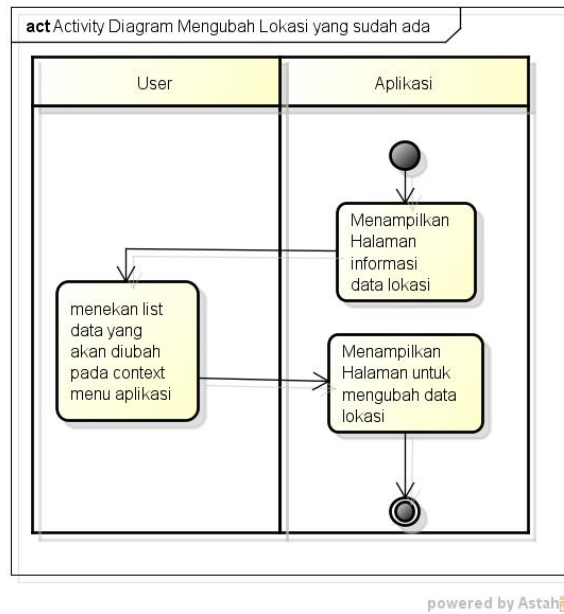
Gambar 4.6 Diagram aktivitas Menambah Informasi Data Lokasi di *Database*
Sumber : [Perancangan]

3. Skenario Use Case Mengubah Informasi Data Lokasi di Database

Tabel 4.6 Use case Mengubah Informasi Data Lokasi di Database

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_001_03
Nama	Mengubah informasi data lokasi di database.
Tujuan	Untuk mengubah informasi data lokasi.
Deskripsi	UseCase ini menjelaskan bagaimana user mengubah informasi data tentang lokasi.
Aktor	User
Skenario Utama	
Kondisi Awal	User berhasil masuk kedalam halaman informasi data lokasi.
Aksi Aktor	Reaksi Sistem
User menekan list data yang akan diubah pada context menu aplikasi.	Perangkat lunak menerima data berupa id informasi kemudian melakukan proses editing data dengan id informasi tersebut.
Kondisi Akhir	Informasi data lokasi berhasil diubah dan disimpan kedalam database.

Sumber : [Perancangan]



Gambar 4.7 Diagram aktivitas Mengubah Informasi Data Lokasi di *Database*
Sumber : [Perancangan]

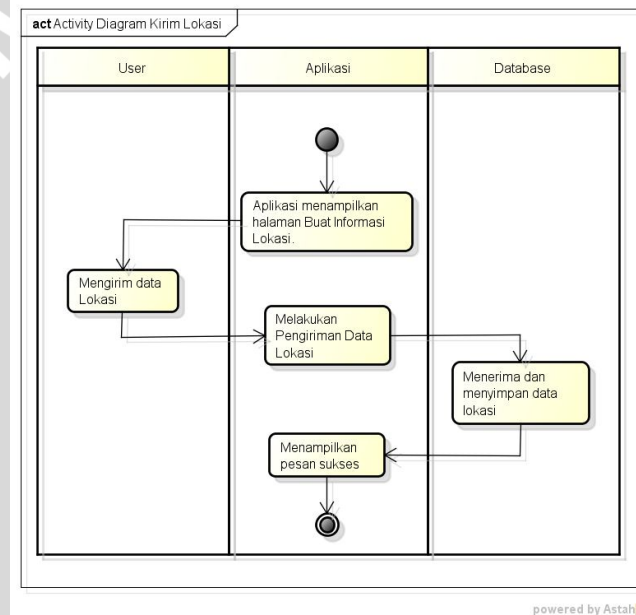
4. Skenario Use Case Mengirim Informasi Data Lokasi di *Database*

Tabel 4.7 Use case Mengirim Informasi Data Lokasi di *Database*

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_001_04
Nama	Mengirim informasi data lokasi di database.
Tujuan	Untuk mengirim informasi data lokasi ke aplikasi administrator.
Deskripsi	UseCase ini menjelaskan bagaimana user mengirim informasi data tentang lokasi ke aplikasi administrator.
Aktor	User
Skenario Utama	
Kondisi Awal	User berhasil melakukan login dan masuk halaman utama user.
Aksi Aktor	Reaksi Sistem

User memilih salah satu nama dari <i>Lokasi</i> dan kemudian menekan tombol “ <i>kirim</i> ”.	Perangkat lunak mengirim data berupa <i>id</i> informasi lokasi, data detail lokasi lalu melakukan proses pengiriman data dengan <i>id</i> informasi tersebut ke aplikasi <i>administrator</i> .
Kondisi Akhir	Informasi data dan detail lokasi berhasil dikirim pada aplikasi <i>administrator</i> dan disimpan di <i>database</i> .

Sumber : [Perancangan]



Gambar 4.8 Diagram aktivitas Mengirim Informasi Data Lokasi di *Database*

Sumber : [Perancangan]

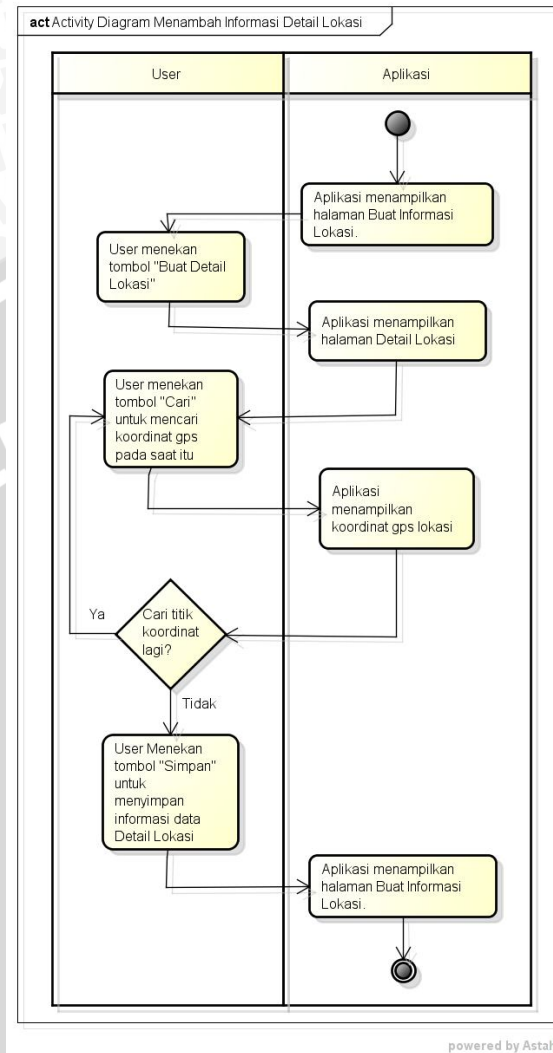
5. Skenario Use Case Menambah Informasi Detail Lokasi

Tabel 4.8 Use case Menambah Informasi Detail Lokasi

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_001_05
Nama	Menambah informasi detail lokasi.
Tujuan	Untuk menambahkan informasi detail lokasi.

Deskripsi	<i>UseCase</i> ini menjelaskan bagaimana <i>user</i> menambahkan informasi data tentang detail lokasi.
Aktor	<i>User</i>
Skenario Utama	
Kondisi Awal	Perangkat lunak menampilkan halaman tambah data lokasi.
Aksi Aktor	Reaksi Sistem
<i>User</i> menekan tombol “Buat Detail Lokasi” dan berhasil masuk halaman tambah detail lokasi.	Perangkat lunak menerima informasi data dan penambahan data detail lokasi yang akan dikirim ke aplikasi <i>administrator</i> .
Skenario Alternatif 1 : Jika <i>user</i> menekan tombol <i>back</i>	
	Perangkat lunak akan kembali ke tambah data lokasi.
Kondisi Akhir	Informasi data detail lokasi berhasil ditambahkan pada aplikasi <i>user</i> .

Sumber : [Perancangan]



Gambar 4.9 Diagram aktivitas Menambah Informasi Detail Lokasi
Sumber : [Perancangan]

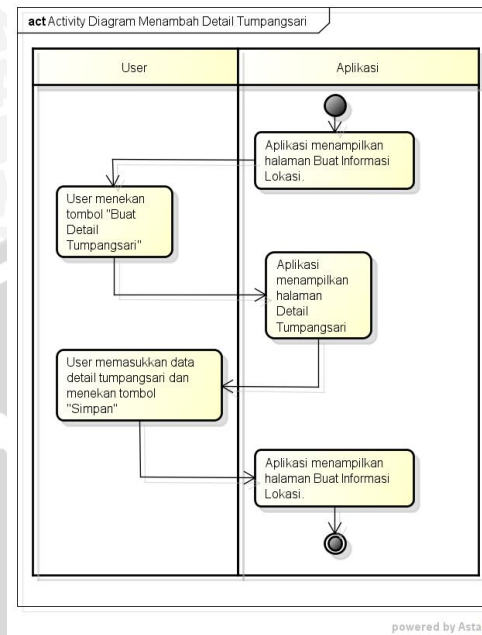
6. Skenario Use Case Menambah Informasi Detail Tumpangsari

Tabel 4.9 Use case Menambah Informasi Detail Tumpangsari

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_001_06
Nama	Menambah informasi detail tumpangsari.
Tujuan	Untuk menambakan informasi detail tumpangsari.
Deskripsi	UseCase ini menjelaskan bagaimana

	<i>user</i> menambahkan informasi data tentang detail tumpangsari.
Aktor	<i>User</i>
Skenario Utama	
Kondisi Awal	Perangkat lunak menampilkan halaman tambah data lokasi.
Aksi Aktor	Reaksi Sistem
<i>User</i> menekan tombol “Buat Detail Tumpangsari” dan menambahkan data tumpangsari pada halaman tambah detail tumpangsari.	Perangkat lunak menerima informasi data dan penambahan datadetail tumpangsari yang akan dikirim ke aplikasi <i>administrator</i> .
Skenario Alternatif 1 : Jika <i>user</i> menekan tombol <i>back</i>	
	Perangkat lunak akan kembali ke tambah data lokasi.
Kondisi Akhir	Informasi data detail tumpangsari berhasil ditambahkan pada aplikasi <i>user</i> .

Sumber : [Perancangan]



Gambar 4.10 Diagram aktivitas Menambah Informasi Detail Tumpangsari
Sumber : [Perancangan]

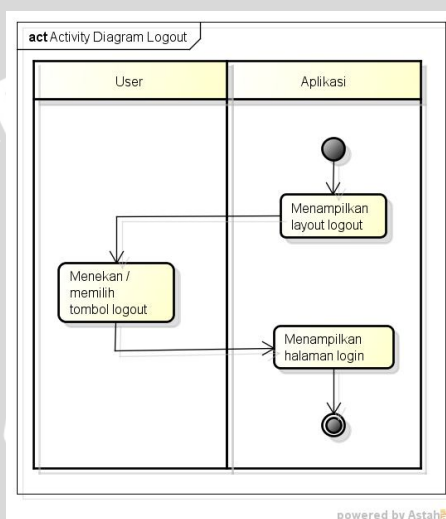
7. Skenario Use Case Logout klien

Tabel 4.10 Use case Logout klien

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_001_05
Nama	Logout klien
Tujuan	Untuk melakukan <i>logout</i> .
Deskripsi	UseCase ini menjelaskan bagaimana user melakukan <i>logout</i> .
Aktor	User
Skenario Utama	
Kondisi Awal	User berhasil melakukan login dan masuk halaman utama user.
Aksi Aktor	Reaksi Sistem
User menekantombol	Perangkat lunak melakukan penghapusan

“Logout” pada context menu aplikasi.	<i>sessionuser</i> dan masuk halaman <i>login</i> .
Kondisi Akhir	User berhasil <i>logout</i> dan tidak dapat kembali ke halaman utama tanpa melakukan <i>use case login</i> klien.

Sumber : [Perancangan]



Gambar 4.11 Diagram aktivitas Logout

Sumber : [Perancangan]

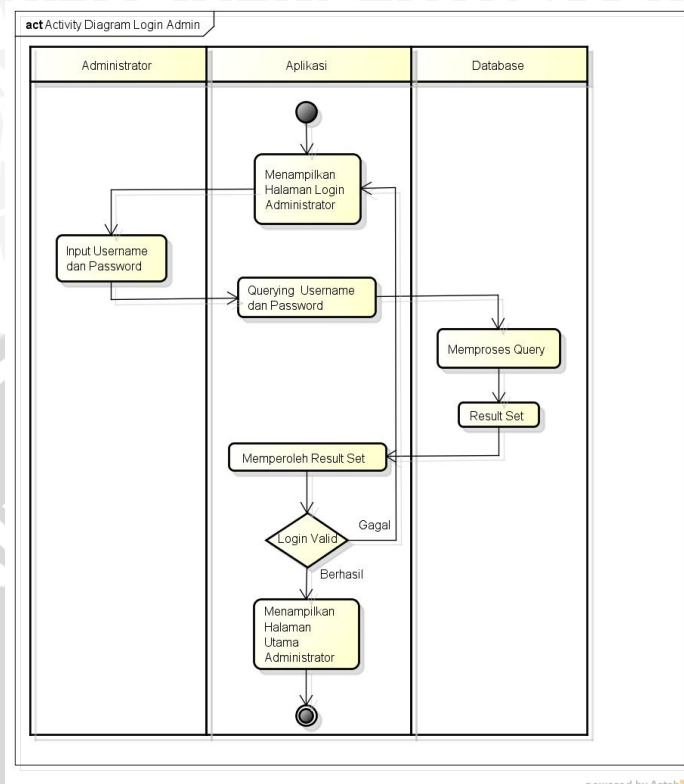
8. Skenario Use Case Menunjukkan Login Administrator

Tabel 4.11 Use case Menunjukkan Login administrator

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_002_01
Nama	Login administrator
Tujuan	Untuk menyeleksi <i>administrator</i> yang sah.
Deskripsi	Use case ini menjelaskan bagaimana <i>administrator</i> melakukan login untuk dapat menampilkan halaman utama <i>administrator</i> .
Aktor	Administrator

Skenario Utama	
Kondisi Awal	Perangkat lunak memunculkan halaman <i>login</i> .
Aksi Aktor	Reaksi Sistem
<i>Administrator</i> memasukkan data login(<i>username</i> dan <i>password</i>), lalu menekan tombol “Login”.	Perangkat lunak menerima data login dan kemudian aplikasi <i>administrator</i> melakukan pengecekan terhadap data yang dimasukkan oleh <i>administrator</i> . Jika data benar, maka akan ditampilkan halaman utama <i>administrator</i> .
Skenario Alternatif 1 : Jika <i>username</i> atau <i>password</i> kosong	
	Perangkat lunak akan menampilkan pesan peringatan bahwa <i>username</i> atau <i>password</i> kosong.
Skenario Alternatif 2 : Jika <i>username</i> dan <i>password</i> salah	
	Perangkat lunak akan menampilkan pesan peringatan bahwa <i>username</i> dan <i>password</i> salah.
Kondisi Akhir	Sistem akan menampilkan halaman utama dari aplikasi <i>administrator</i> .

Sumber : [Perancangan]



Gambar 4.12 Diagram aktivitas *Login administrator*
Sumber : [Perancangan]

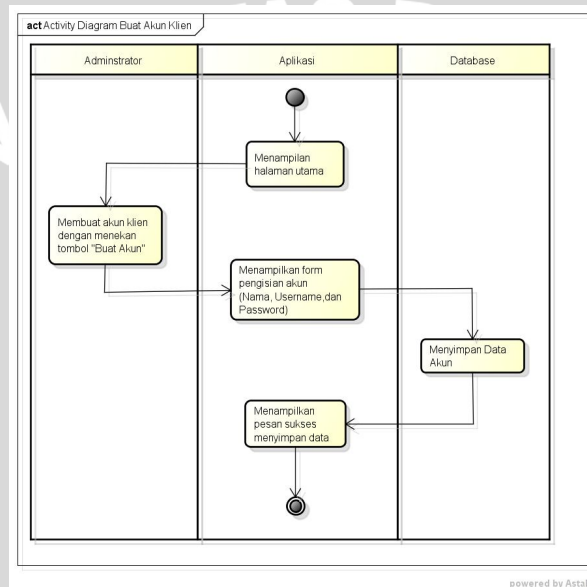
9. Skenario Use Case Membuat Akun Klien

Tabel 4.12 Use Case Membuat Akun Klien

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_002_02
Nama	Membuat Akun Klien.
Tujuan	Untuk membuat akun klien.
Deskripsi	UseCase ini menjelaskan bagaimana administrator melakukan pembuatan akun untuk klien atau user.
Aktor	Administrator
Skenario Utama	
Kondisi Awal	Administrator berhasil melakukan login dan masuk halaman utama administrator.

Aksi Aktor	Reaksi Sistem
<i>Administrator</i> memilih kata “Buat Akun”.	Sistem akan menampilkan <i>form</i> membuat akun klien.
Kondisi Akhir	Sistem menyimpan akun yang dibuat ke <i>database</i> .

Sumber : [Perancangan]



Gambar 4.13 Diagram aktivitas Membuat Akun Klien

Sumber : [Perancangan]

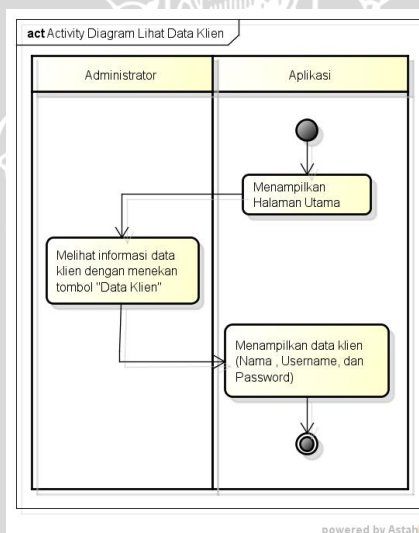
10. Skenario Use CaseMelihat Informasi Data Klien

Tabel 4.13UseCaseMelihat Informasi Data Klien

Skenario Kasus Pada Sistem	
NomorUse Case	SRS_002_03
Nama	Melihat Informasi Data Klien.
Tujuan	Untuk melihat informasi data klien.
Deskripsi	UseCase ini menjelaskan bagaimana <i>administrator</i> melakukan pencarian informasi data klien.
Aktor	<i>Administrator</i>

Skenario Utama	
Kondisi Awal	<i>Administrator</i> berhasil melakukan login dan masuk halaman utama <i>administrator</i> .
Aksi Aktor	Reaksi Sistem
<i>Administrator</i> melihat informasi data klien dan menekan tombol “Data Klien”.	Sistem akan menampilkan halaman data klien.
Kondisi Akhir	Sistem akan menampilkan data Klien (<i>username</i> , <i>nama</i> , dan <i>password</i>).

Sumber : [Perancangan]



Gambar 4.14 Diagram aktivitas Melihat Informasi Data Klien

Sumber : [Perancangan]

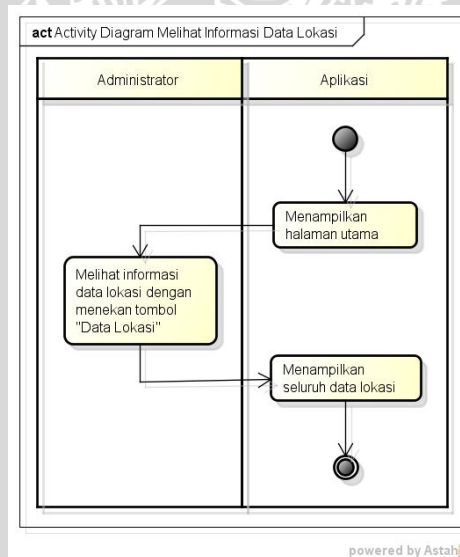
11. Skenario Use Case MelihatSeluruh Informasi Data Lokasi

Tabel 4.14 UseCase MelihatSeluruhInformasi Data Lokasi

Skenario Kasus Pada Sistem	
NomorUse Case	SRS_002_04
Nama	MelihatSeluruhInformasi Data Lokasi.

Tujuan	Untuk melihat informasi data lokasi.
Deskripsi	<i>UseCase</i> ini menjelaskan bagaimana <i>administrator</i> melakukan pencarian seluruh informasi data lokasi.
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi Awal	<i>Administrator</i> berhasil melakukan login dan masuk halaman utama <i>administrator</i> .
Aksi Aktor	Reaksi Sistem
<i>Administrator</i> menekan "Lokasi"	Sistem akan menampilkan seluruh informasi data lokasi.
Kondisi Akhir	Sistem akan menampilkan seluruh informasi data lokasi.

Sumber : [Perancangan]



Gambar 4.15 Diagram aktivitas Melihat Seluruh Informasi Data Lokasi

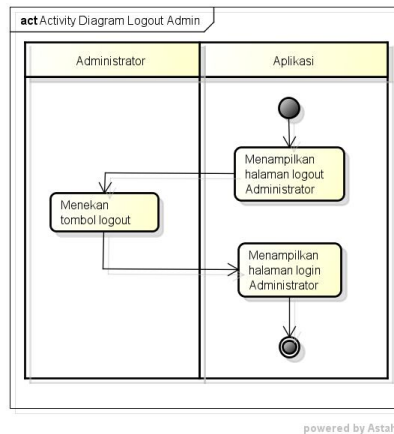
Sumber : [Perancangan]

12. Skenario Use Case Logout Admin

Tabel 4.15 *UseCaseLogout Admin*

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_002_06
Nama	<i>Logout Admin</i>
Tujuan	Untuk melakukan <i>logout</i> .
Deskripsi	<i>UseCase</i> ini menjelaskan bagaimana <i>administrator</i> melakukan <i>logout</i> .
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi Awal	<i>Administrator</i> berhasil melakukan login dan masuk halaman utama <i>administrator</i> .
Aksi Aktor	Reaksi Sistem
<i>Administrator</i> menekantombol “Logout” pada <i>context</i> menu aplikasi.	Perangkat lunak melakukan penghapusan <i>session administrator</i> dan masuk halaman <i>login admin</i> .
Kondisi Akhir	<i>Administrator</i> berhasil <i>logout</i> dan tidak dapat kembali kehalaman utama tanpa melakukan <i>use case login Admin</i> .

Sumber : [Perancangan]



Gambar 4.16 Diagram aktivitas *Logout Admin*
Sumber : [Perancangan]

4.1.6 Analisis Data

Analisis data bertujuan untuk mendapatkan struktur penyimpanan data yang dibutuhkan perangkat lunak Pemetaan Lahan Hutan berbasis Android. Struktur penyimpanan data pada perangkat lunak Pemetaan Lahan Hutan berbasis Android disusun berdasarkan analisis data sebagai berikut :

a. Aplikasi *User*

- o Data klien yang terdiri dari nama lengkap, *username*, dan *password*. Data ini berfungsi sebagai data pegawai Perhutani yang digunakan untuk *login* kedalam aplikasi *user*.
- o Data lokasi yang terdiri atas nama lokasi, jenis tanaman pokok, ketinggian dasar lokasi, bonita (kepadatan hutan), kepadatan bidang, kelas hutan, bentuk lapangan, waktu tanam lahan hutan, jenis tanaman tumpangsari dan waktu tanam tumpangsari. Data ini berfungsi untuk menyimpan data lokasi yang akan di petakan.
- o Data detail_lokasi terdiri dari longitude, dan latitude. Data ini berfungsi untuk menyimpan data koordinat gps lokasi yang nantinya akan ditampilkan pada aplikasi *administrator*.

b. Aplikasi *administrator*

- o Data *administrator* yang terdiri atas *username*, *password*, dan *email*. Data ini berfungsi untuk menyimpan data *administrator*, sehingga hanya *administrator* yang dapat masuk ke dalam aplikasi *administrator*.
- o Data klien yang terdiri atas *username*, *password*, dan nama lengkap. Data ini berfungsi untuk menyimpan data pegawai Perhutani yang digunakan untuk *login* kedalam aplikasi *user*.
- o Data lokasi yang terdiri atas nama lokasi, jenis tanaman pokok, ketinggian dasar lokasi, *bonita* (kepadatan hutan), kepadatan bidang, kelas hutan, bentuk lapangan, waktu tanam lahan hutan, jenis tanaman tumpangsari dan waktu tanam tumpangsari. Data ini berfungsi untuk menyimpan data lokasi yang akan di petakan.
- o Data detail_lokasi terdiri dari *longitude*, dan *latitude*. Data ini berfungsi untuk menyimpan data koordinat *gps* lokasi yang nantinya akan ditampilkan pada aplikasi *administrator*.

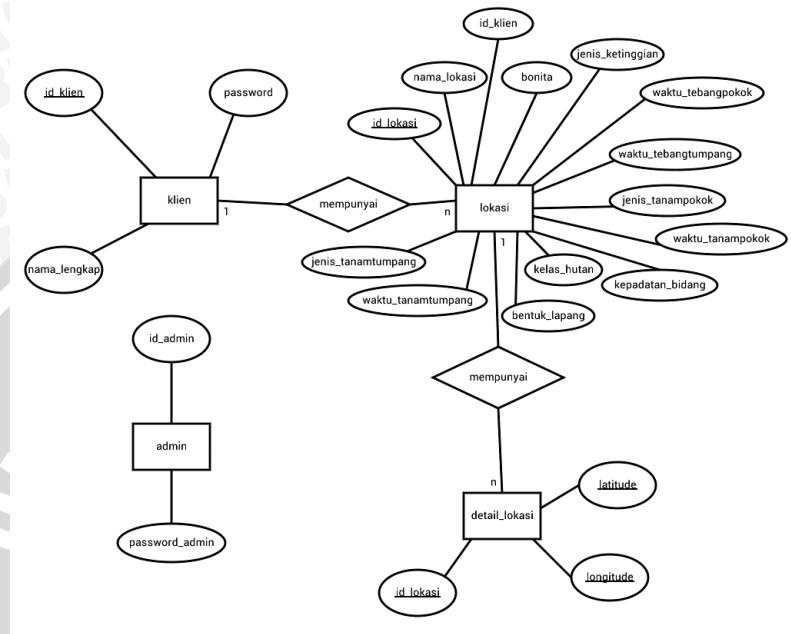
4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan dalam empat tahap, yaitu perancangan basis data, pemodelan diagram *class* untuk menggambarkan perancangan struktur *class – class* yang menyusun perangkat lunak perangkat lunak, perancangan *sequences diagram* untuk mengetahui proses dan alur program saat dijalankan, dan perancangan antarmuka pengguna dari perangkat lunak perangkat lunak Pemetaan Lahan Hutan Berbasis Android. Perancangan perangkat lunak pada skripsi ini menggunakan pendekatan desain berorientasi objek yang direpresentasikan dengan menggunakan UML (*Unified Modelling Language*).

4.2.1 Perancangan Basis Data

Basis data berfungsi sebagai tempat menyimpan data. Perancangan basis data digunakan untuk merancang basis data yang akan dibuat agar masukan dan keluaran

program sesuai dengan apa yang diharapkan. Perancangan basis data mengambil acuan dari proses analisis data yang dilakukan pada tahap analisis kebutuhan.



Gambar 4.17Diagram *EntityRelationship*
Sumber : [Perancangan]

Rancangan masing – masing tabel adalah sebagai berikut :

1. Tabel Admin

Nama tabel : Admin

Jumlah *field* : 2

Fungsi : Untuk menyimpan data admin

Tabel 4.16Struktur tabel admin

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	id_admin	TEXT	-	Id Klien
2	Password_admin	TEXT	-	Password Admin

Sumber : [Perancangan]

2. Tabel Klien

Nama tabel : Klien

Jumlah *field* : 3

Fungsi : Untuk menyimpan data klien

Tabel 4.17 Struktur tabel klien

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	id_klien	TEXT	-	Id Klien
2	nama_lengkap	TEXT	-	NamaKlien
3	Password	TEXT	-	Password Klien

Sumber : [Perancangan]

3. Tabel Lokasi

Nama tabel : Lokasi

Jumlah *field* : 10

Fungsi : Untuk menyimpan data lokasi

Tabel 4.18 Struktur tabel lokasi

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	id_lokasi	TEXT	-	Id Lokasi
2	id_klien	TEXT	-	Id Klien
3	nama_lokasi	TEXT	-	Nama Lokasi
4	jenis_tanampokok	TEXT	-	Jenis Tanaman Pokok
5	jenis_ketinggian	TEXT	-	Ketinggian Dasar Lokasi
6	Bonita	TEXT	-	Bonita (kepadatan hutan)

7	bentuk_lapang	TEXT	-	Bentuk Lapangan
8	kelas_hutan	TEXT	-	Kelas Hutan
9	kepadatan_bidang	TEXT	-	Kepadatan Bidang
10	waktu_tanampokok	TIME	-	Waktu Tanam Tanaman Pokok
11	jenis_tanamtumpang	TEXT	-	Jenis Tanaman Tumpang Sari
12	waktu_tanamtumpang	TIME	-	Waktu Tanam Tumpang Sari
13	waktu_tebangpokok	TIME	-	Waktu Tebang Tanaman Pokok
14	waktu_tebangtumpang	TIME	-	Waktu Tebang Tanaman Pokok

Sumber : [Perancangan]

4. Tabel detail_lokasi

Nama tabel : Detail_lokasi

Jumlah *field* : 3

Fungsi : Untuk menyimpan data detail lokasi

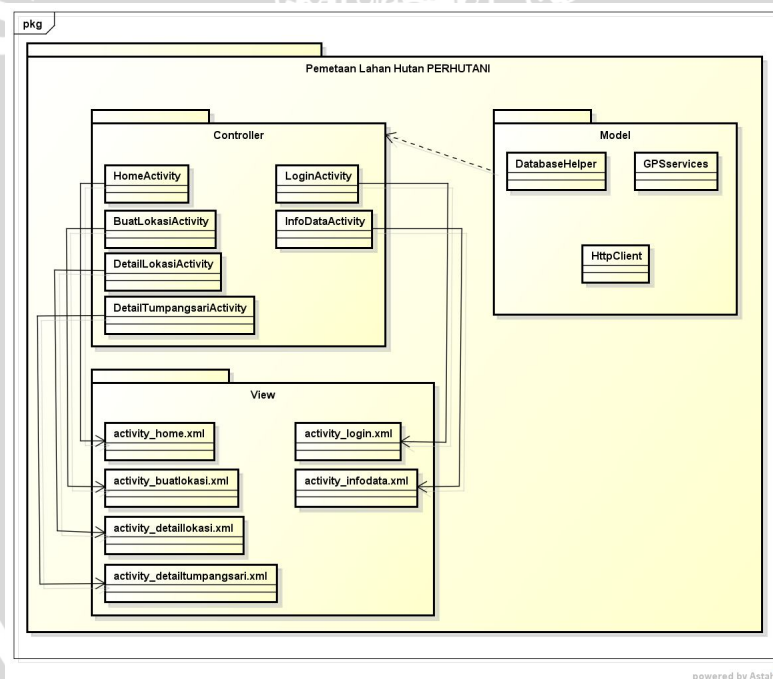
Tabel 4.19 Struktur tabel detail_lokasi

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	id_lokasi	TEXT	-	Id Lokasi
2	Longitude	REAL	-	longitude
3	Latitude	REAL	-	latitude

Sumber : [Perancangan]

4.2.2 Pemodelan Kelas

Pemodelan kelas memberikan gambaran pemodelan elemen-elemen kelas yang membentuk sebuah perangkat lunak. Kelas bisa didapatkan dengan menganalisis secara detail terhadap *use case* yang dimodelkan. Perangkat lunak Pemetaan Lahan Hutan Berbasis Android memiliki 22 kelas yang dibagi menjadi 2 komponen yaitu komponen *Helper*, dan komponen inti. Perangkat lunak Pemetaan Lahan Hutan Berbasis Android juga disusun menggunakan struktur *model-view-controller* (MVC). Gambar 4.18 menunjukkan pemodelan MVC diagram kelas dari perangkat lunak Pemetaan Lahan Hutan Berbasis Android yang akan dibuat.



Gambar 4.18 Pemodelan MVC Kelas Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android
Sumber : [Perancangan]

Pada pemodelan kelas di atas digambarkan *interface* dan beberapa kelas utama yang menyusun perangkat lunak Pemetaan Lahan Hutan Berbasis Android. Bagian *Controller* memiliki lima kelas yang menangani tiap fitur yang tersedia di Pemetaan

Lahan Hutan Berbasis Android. Bagian *Model* memiliki tiga buah kelas yaitu DataBaseHelper untuk menyimpan data dan mengambil data dari *SQL Lite*, GPSServices untuk mengaktifkan fungsi GPS yang kemudian dibutuhkan oleh perangkat lunak, dan HttpClient untuk menghubungkan aplikasi *user* yang berbentuk *mobile* dan aplikasi *administrator* yang berbentuk web. Bagian *views* terdiri dari kelas-kelas xml yang berfungsi sebagai *layout* di *platform* android.

4.2.2.1 Pemodelan Kelas Komponen Helper

Kelas komponen *helper* merupakan komponen yang sudah ada. Kelas ini terdiri dari empat kelas yaitu kelas DataBaseHelper, Kelas GPSServices, dan Kelas HttpClient Kelas SessionManager.

A. Kelas DataBaseHelper

Kelas DataBaseHelper digunakan untuk menyimpan data dari inputan *user* kedalam *SQL Lite*. Struktur kelas DataBaseHelper digambarkan di Gambar 4.19.

DatabaseHelper	
1	- NamaDB : String
2	- Lokasi_Field : Arrays
3	- Koordinat_Field : Arrays
4	+ onCreate(SQLiteDatabase db : int) : void
5	+ insertLokasi(Lokasi_Field : Arrays) : long
6	+ UpdateLokasi(Lokasi_Field : Arrays) : long
7	+ insertKoordinat(Koordinat_Field : Arrays) : long
8	+ ambilSemuaLokasi() : ArrayList
9	+ getKoordinatById(id : String) : List
10	+ deleteKoordinatById(Koordinat_Field : Arrays) : int

Gambar 4.19 Kelas DataBaseHelper Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Sumber : [Perancangan]

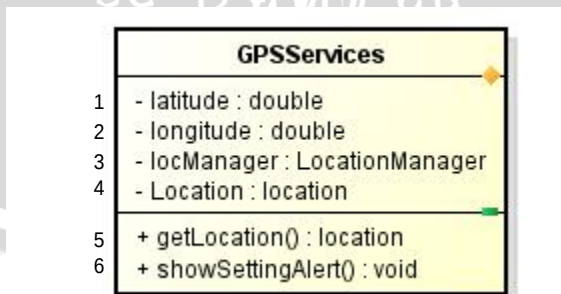
1. Baris 1-3 berfungsi untuk mendeklarasikan variabel yang akan dipakai. NamaDB merupakan nama table yang akan dibuat pada kelas ini, Lokasi_Field bertipe

Array yang berisi data lokasi, sedangkan Koordinat_Field bertipe Array yang berisi data detail lokasi.

2. Baris 4 *Method* onCreate berfungsi untuk membuat table yang akan digunakan sebagai tempat penyimpanan di SQL Lite.
3. Baris 5 *Method* insertLokasi berfungsi untuk menyimpan Lokasi_Field kedalam tabel yang telah dibuat di SQL Lite.
4. Baris 6 *Method* UpdateLokasi berfungsi untuk mengubah isi data Lokasi_Field kedalam tabel yang telah dibuat di SQL Lite.
5. Baris 7 *Method* InsertKoordinat berfungsi untuk menyimpan Koordinat_Field kedalam tabel yang telah dibuat di SQL Lite.
6. Baris 8 *Method* ambilSemuaLokasi berfungsi untuk memanggil semua data lokasi dari SQL Lite.
7. Baris 9 *Method* getKoordinatById berfungsi untuk memanggil data koordinat berdasarkan id lokasi dari SQL Lite.
8. Baris 10 *Method* deleteKoordinatById berfungsi untuk menghapus data koordinat berdasarkan id lokasi dari SQL Lite.

B. Kelas GPSServices

Kelas GPSServices berfungsi untuk mengaktifkan fungsi GPS yang digunakan untuk mencari titik koordinat suatu wilayah. Struktur kelas GPSServices digambarkan di Gambar 4.20.



Gambar 4.20 Kelas GPSServices Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Sumber : [Perancangan]

1. Baris 1-4 berfungsi untuk mendeklarasikan variabel yang akan dipakai.
2. Baris 5 *Method* getLocation yang digunakan untuk mengambil nilai latitude dan longitude dari suatu wilayah.
3. Baris 6 *Method* showSettingAlert berfungsi menampilkan *alert pop up window*.

C. Kelas HttpClient

Kelas HttpClient untuk menghubungkan aplikasi *user* yang berbentuk *mobile* dan aplikasi *administrator* yang berbentuk web. Struktur kelas HttpClient digambarkan di Gambar 4.21.

HttpClient	
1	- baseUrl : String
2	- params : ArrayList
3	- client : DefaultHttpClient
4	+ HttpClient() : void
5	+ AddParam(String name : int, String value : int) : void
6	+ executeRequest() : void

Gambar 4.21 Kelas HttpClient Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Sumber : [Perancangan]

1. Baris 1-3 berfungsi untuk mendeklarasikan variabel yang akan dipakai.
2. Baris 4 *Method* HttpClient yang diakses pertama kali ketika kelas HttpClient dijalankan. Pada method ini berisi inisialisasi variable params dan client.
3. Baris 5 *Method* AddParam berfungsi untuk mengimpan parameter yang ada di aplikasi *user* yang nantinya akan di kirimkan ke aplikasi *administrator*.
4. Baris 6 *Method* executeRequest berfungsi untuk mengirimkan parameter ke aplikasi *administrator*.

D. Kelas Komponen *Session Manager*

Komponen session manager merupakan kelas yang berfungsi menjaga *user* tetap *login* hingga *user* menekan tombol *logout*. Kelas session manager berfungsi untuk menyimpan *username* di dalam session. Gambar 4.22 menjelaskan struktur kelas *SessionManager*

SessionManager	
1	- pref : SharedPreferences
2	- editor : Editor
3	- context : Context
4	- PRIVATE_MODE : int
5	- PREF_USERNAME : String
6	- KEY_USERNAME : String
7	
8	+ SessionManager(context : Context) : void
9	+ createLoginSession(username : String) : void
10	+ getUserDetail() : HashMap<String,String>() : void
11	+ logoutUser() : void
12	+ isLoggedIn() : boolean

Gambar 4.22 Kelas *SessionManager* Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android
Sumber : [Perancangan]

1. Baris 7 merupakan deklarasi variabel key *username*.
2. Baris 9 merupakan method *createLoginSession*. Method ini digunakan untuk membuat session baru.
3. Baris 10 merupakan method *getUserDetail* digunakan untuk mengetahui data *user* yang sedang login.
4. Baris 11 merupakan method yang digunakan untuk *logout user* dengan cara menghapus session.

4.2.2.2 Pemodelan Kelas Komponen *Inti*

Komponen Inti merupakan kelas *controller* yang berfungsi sebagai penghubung antara kelas model pada komponen XML Parser ke kelas view. Komponen inti terdiri dari 6 kelas yaitu: kelas *HomeActivity*, *BuatLokasiActivity*,

DetailLokasiActivity, DetailTumpangsariActivity, InfoDataActivity, dan LoginActivity.

A. Kelas HomeActivity

Kelas HomeActivity berfungsi sebagai tampilan utama perangkat lunak Pemetaan Lahan Hutan berbasis Android. Gambar 4.23 menjelaskan struktur dari kelas HomeActivity. Pada kelas HomeActivity terdapat navigasi – navigasi fitur untuk mengakses halaman Buat Lokasi dan Informasi Data.

HomeActivity	
1	- session : SessionManager
2	+ onCreate(savedInstanceState : Bundle) : void
3	+ buatClick(v : View) : void
4	+ infoClick(v : View) : void
5	+ logoutClick(v : View) : void

Gambar 4.23 Kelas HomeActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Sumber : [Perancangan]

1. Baris 1 berfungsi untuk mendeklarasikan variabel yang akan dipakai.
2. Baris 2 merupakan *method* onCreate yang diakses pertama kali ketika kelas HomeActivity dijalankan. Pada method ini berisi bagaimana menampilkan data ke *interface*.
3. Baris 3 merupakan *method* untuk menampilkan halaman tambah data lokasi.
4. Baris 4 merupakan *method* untuk menampilkan halaman informasi data lokasi.
5. Baris 5 merupakan *method* untuk menghapus session dan menampilkan halaman login.

B. Kelas BuatLokasiActivity

Kelas BuatLokasiActivity berfungsi sebagai jembatan antara data lokasi dan *interface* tambah data lokasi. Gambar 4.24 menjelaskan struktur dari kelas BuatLokasiActivity.

BuatLokasiActivity	
1	- KEY_USERNAME : String
2	- KEY_IDLOKASI : String
3	- KEY_NAMALOKASI : String
4	- KEY_JENISTANAMPOKOK : String
5	- KEY_JENISKETINGIAN : String
6	- KEY_BONITA : String
7	- KEY_BENTUKLAPANG : String
8	- KEY_KELASHUTAN : String
9	- KEY KEPADATANBIDANG : String
10	- KEY_WAKTUTANAMPOKOK : String
11	- session : SessionManager
12	+ onCreate(savedInstanceState : Bundle) : void
13	+ tumpangsariClick(v : View) : void
14	+ detaillokasiClick(v : View) : void
15	+ uploadClick(v : View) : void
16	+ logoutClick(v : View) : void

Gambar 4.24 Kelas BuatLokasiActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Sumber : [Perancangan]

1. Baris 1-10 merupakan deklarasi variabel data lokasi yang akan disimpan di SQL *Lite* dan dikirimkan ke aplikasi *administrator*. Data yang tersebut terdiri atas, data *username* klien, id lokasi, nama lokasi, jenis tanaman pokok, ketinggian dasar lokasi, bonita, kepadatan bidang, kelas hutan, bentuk lapangan, dan waktu tanam lahan hutan.
2. Baris 11 merupakan deklarasi variable session yang berfungsi untuk menyimpan session dari *username* klien.
3. Baris 12 merupakan *method* *onCreate* yang diakses pertama kali ketika kelas BuatLokasiActivity dijalankan. Pada method ini berisi bagaimana menampilkan data ke *interface*.
4. Baris 13 merupakan *method* untuk menampilkan halaman tumpangsari.
5. Baris 14 merupakan *method* untuk menampilkan halaman detail lokasi.
6. Baris 15 merupakan *method* untuk mengirimkan data lokasi ke aplikasi administrator dan sekaligus menyimpan data lokasi ke SQL *Lite*.

7. Baris 16 merupakan *method* untuk menghapus session dan menampilkan halaman login.

C. Kelas DetailLokasiActivity

Kelas DetailLokasiActivity berfungsi sebagai jembatan antara data detail lokasi dan *interfacetambah* data detail lokasi. Gambar 4.25 menjelaskan struktur dari kelas DetailLokasiActivity.

DetailLokasiActivity	
1	- KEY_IDLOKASI : String
2	- KEY_LATITUDE : double
3	- KEY_LONGITUDE : double
4	- lv_koordinat : ListViewActivity
5	- gps : GPSServices
6	- session : SessionManager
7	+ onCreate(savedInstanceState : Bundle) : void
8	+ carikoordinatClick(v : View) : void
9	+ simpankoordinatClick(v : View) : void
10	+ updateListKoordinat() : void
11	+ logoutClick(v : View) : void

Gambar 4.25 Kelas DetailLokasiActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Sumber : [Perancangan]

1. Baris 1-3 merupakan deklarasi variabel data lokasi yang akan disimpan di SQL *Lite* dan dikirimkan ke aplikasi *administrator*. Data yang dikirimkan merupakan data id lokasi, latitude, dan longitude.
2. Baris 4 merupakan deklarasi variable yang berfungsi sebagai tempat list dari koordinat lokasi.
3. Baris 5 merupakan deklarasi variable gps yang berfungsi untuk mencari titik koordinat dari lokasi.
4. Baris 6 merupakan deklarasi variable session yang berfungsi untuk menyimpan session dari *username* klien.

5. Baris 7 merupakan *method* onCreate yang diakses pertama kali ketika kelas DetailLokasiActivity dijalankan. Pada method ini berisi bagaimana menampilkan data ke *interface* dan menampilkan list data koordinat terkini ke *interface*.
6. Baris 8 merupakan *method* untuk mencari koordinat dari suatu lokasi.
7. Baris 9 merupakan *method* untuk menyimpan koordinat lokasi kedalam SQL *Lite*.
8. Baris 10 merupakan *method* untuk mengambil data koordinat lokasi dari SQL *Lite* dan ditampilkan dalam bentuk list.
9. Baris 11 merupakan *method* untuk menghapus session dan menampilkan halaman login.

D. Kelas DetailTumpangsariActivity

Kelas DetailTumpangsariActivity berfungsi sebagai jembatan antara data tumpangsari dan *interface* tambah data tumpangsari. Gambar 4.26 menjelaskan struktur dari kelas DetailTumpangsariActivity.

DetailTumpangsariActivity	
1	- KEY_JENISTANAMTUMPANG : String
2	- KEY_WAKTUTANAMTUMPANG : String
3	+ onCreate(savedInstanceState : Bundle) : void
4	+ detailtumpangsariClick(v : View) : void
5	+ logoutClick(v : View) : void

Gambar 4.26 Kelas DetailTumpangsariActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Sumber : [Perancangan]

1. Baris 1 dan 2 merupakan deklarasi variabel data tumpangsari yang akan dikirimkan ke halaman tambah data lokasi dan dikirimkan ke aplikasi *administrator*. Data yang dikirimkan merupakan data jenis tanaman tumpangsari dan waktu tanam tanaman tumpangsari.

2. Baris 3 merupakan *method* onCreate yang diakses pertama kali ketika kelas DetailTumpangsariActivity dijalankan. Pada method ini berisi bagaimana menampilkan data ke *interface*.
3. Baris 4 merupakan *method* untuk mengirim data tumpangsari ke halaman tambah data lokasi.
4. Baris 5 merupakan *method* untuk menghapus session dan menampilkan halaman login.

E. Kelas InfoDataActivity

Kelas InfoDataActivity berfungsi sebagai jembatan antara data tumpangsari dan *interface* tambah data tumpangsari. Gambar 4.27 menjelaskan struktur dari kelas InfoDataActivity.

InfoDataActivity	
1	- lv_lokasi : ListViewActivity
2	- session : SessionManager
3	+ onCreate(savedInstanceState : Bundle) : void
4	+ updateTable() : void
5	+ logoutClick(v : View) : void

Gambar 4.27 Kelas InfoDataActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Sumber : [Perancangan]

1. Baris 1 merupakan deklarasi variable yang berfungsi sebagai tempat list dari data lokasi.
2. Baris 2 merupakan deklarasi variable session yang berfungsi untuk menyimpan session dari *username* klien.
3. Baris 3 merupakan *method* onCreate yang diakses pertama kali ketika kelas InfoDataActivitydijalankan. Pada method ini berisi bagaimana menampilkan data ke *interface* dan menampilkan list data lokasi terkini ke *interface*.
4. Baris 4 merupakan *method* untuk mengambil data lokasi dari SQL *Lite* dan ditampilkan dalam bentuk list.

5. Baris 5 merupakan *method* untuk menghapus session dan menampilkan halaman login.

F. Kelas LoginActivity

Kelas LoginActivity berfungsi sebagai jembatan antara data tumpangsari dan *interfacetambah* data tumpangsari. Gambar 4.28 menjelaskan struktur dari kelas LoginActivity.

LoginActivity	
1	- KEY_USERNAME : String
2	- KEY_PASSWORD : String
3	- session : SessionManager
4	+ onCreate(savedInstanceState : Bundle) : void
5	+ loginClicl(y : View) : void

Gambar 4.28 Kelas LoginActivity Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

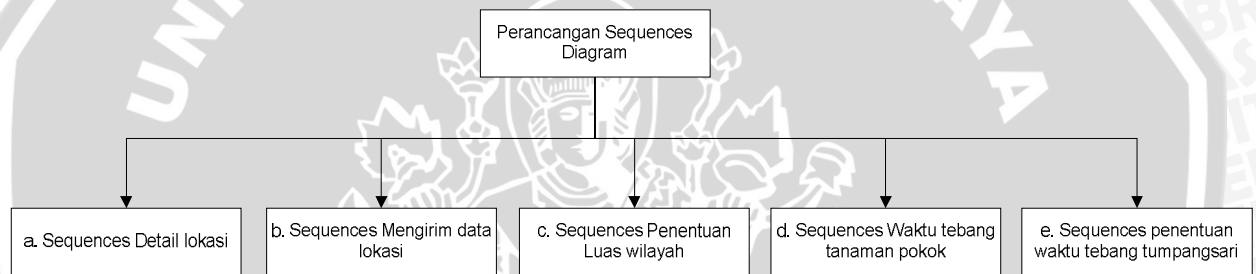
Sumber : [Perancangan]

1. Baris 1 dan 2 merupakan deklarasi variabel *username* dan *password* yang akan divalidasi ke *database* pada aplikasi *administrator*.
2. Baris 3 merupakan deklarasi variable session yang berfungsi untuk menyimpan session dari *username* klien.
3. Baris 4 merupakan *method* *onCreate* yang diakses pertama kali ketika kelas LoginActivity dijalankan. Pada method ini berisi bagaimana menampilkan data ke *interface*.
4. Baris 5 merupakan *method* untuk melakukan validasi *username* dan *password* ke *database* pada aplikasi *administrator*. Apabila valid, maka akan menuju ke halaman *Home/* utama.

4.2.3 Perancangan Sequences Diagram

Sequence diagram menunjukkan pemodelan aliran jalannya proses interaksi antar objek atau *class* yang disusun berdasarkan urutan waktu pada perangkat lunak

Pemetaan Lahan Hutan berbasis Android. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diagram sekuensial (*sequence diagram*) disusun dengan mengambil acuan pada *usecase* dan *class - class* yang membentuk fungsionalitas yang digambarkan pada *usecase* tersebut. Pada perangkat lunak Pemetaan Lahan Hutan berbasis Android, *sequence diagram* terdiri dari *sequence diagram* detail lokasi, kirim informasi data lokasi, penentuan Luas wilayah, penentuan waktu tebang tanaman pokok, dan penentuan waktu tebang tumpangsari. Gambar 4.29 menjelaskan diagram blok perancangan *sequences diagram*.

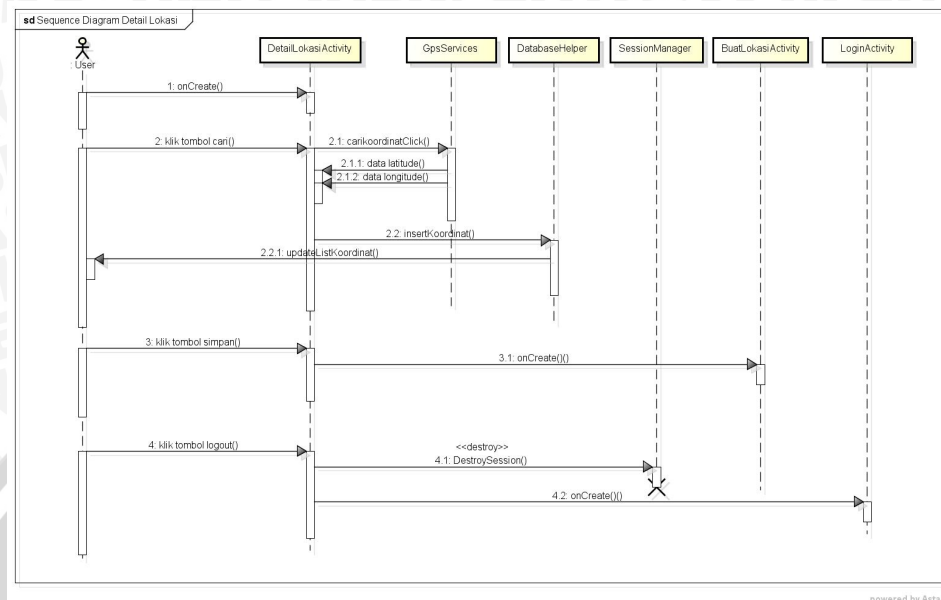


Gambar 4.29Diagram blok perancangan *sequences diagram*

Sumber : [Perancangan]

A. *Sequence diagram* detail lokasi

Sequences diagram detail lokasi menggambarkan proses ketika *user* akan mencari koordinat wilayah dan menyimpan di *SQL Lite*. Gambar 4.30 menjelaskan *sequences diagram* detail lokasi.



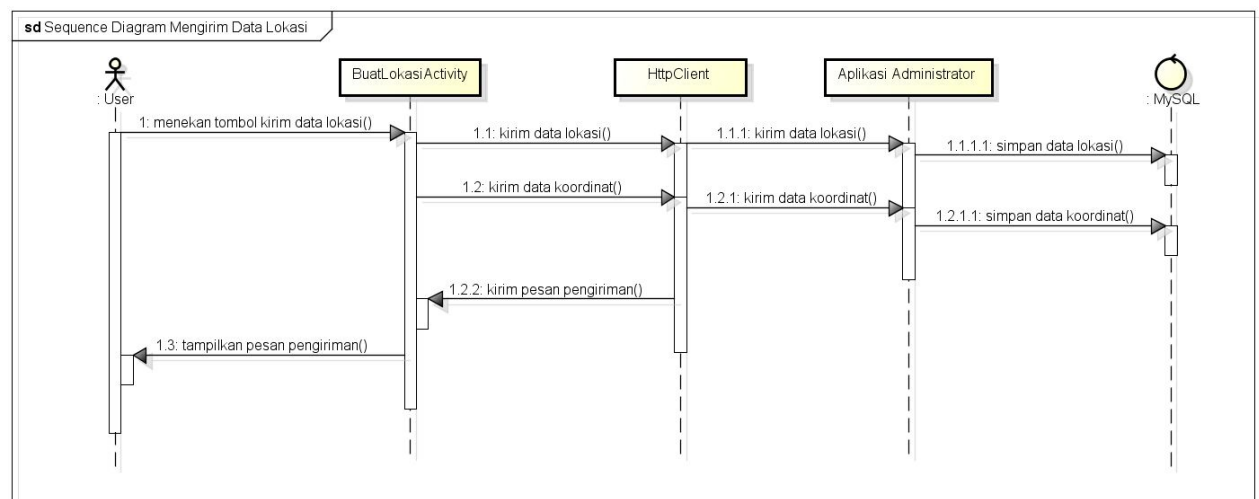
Gambar 4.30 Sequence diagram detail lokasi Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android
Sumber : [Perancangan]

1. User akan masuk ke dalam halaman Detail Lokasi atau kelas DetailLokasiActivity
2. User kemudian menekan tombol cari.
3. Kelas DetailLokasiActivity mengeksekusi *method* cariKoordinatClick().
4. Kelas GpsServices merespon dengan mengembalikan data berupa data longitude dan data latitude.
5. Kelas DetailLokasiActivity mengeksekusi *method* insertLokasi() yang berfungsi untuk menyimpan data longitude dan latitude kedalam database SQL *lite*.
6. Kelas DatabaseHelper mengeksekusi *method* updateListKoordinat() yang berfungsi untuk melihat list data koordinat terbaru dan dapat dilihat oleh user.
7. User menekan tombol simpan.
8. Kelas DetailLokasiActivity memanggil kelas BuatLokasiActivity dan menampilkan halaman tambah data lokasi.
9. User menekan tombol logout.
10. Kelas DetailLokasiActivity menghapus *sessionusername* user.

11. Kelas DetailLokasiActivity memanggil kelas LoginActivity dan menampilkan halaman login *user*.

B. *Sequence diagram* mengirim data lokasi

Sequences diagram mengirim data lokasi menggambarkan proses ketika *user* akan mengirimkan seluruh data ke aplikasi *administrator*. Gambar 4.31 menjelaskan *sequences diagram* mengirim data lokasi.



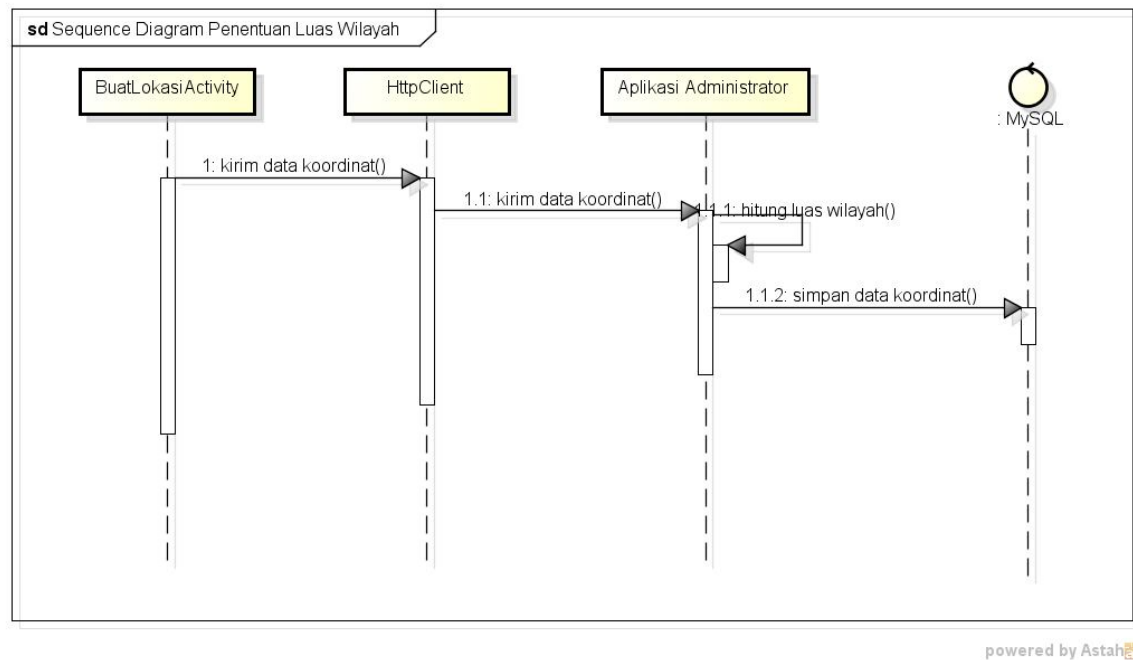
powered by Astah

Gambar 4.31 *Sequence diagram* mengirim data lokasi Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android
Sumber : [Perancangan]

1. *User* kemudian menekan tombol kirim.
2. Kelas BuatLokasiActivity mengirim data lokasi dan data koordinat ke kelas HttpClient.
3. Kelas HttpClient meneruskan untuk mengirim data tersebut ke aplikasi *administrator*.
4. Aplikasi *administrator* menyimpan data lokasi dan koordinat kedalam Database.
5. Kelas HttpClient memberikan kembali berupa status pengiriman.
6. Kelas BuatLokasiActivity menampilkan status pengiriman.

C. Sequence diagram Penentuan Luas wilayah

Sequences diagram Penentuan Luas wilayah menggambarkan proses ketika aplikasi *administrator* menentukan Luas wilayah menggunakan data koordinat yang telah dikirimkan oleh aplikasi *user*. Gambar 4.32 menjelaskan *sequences diagram* Penentuan Luas wilayah.

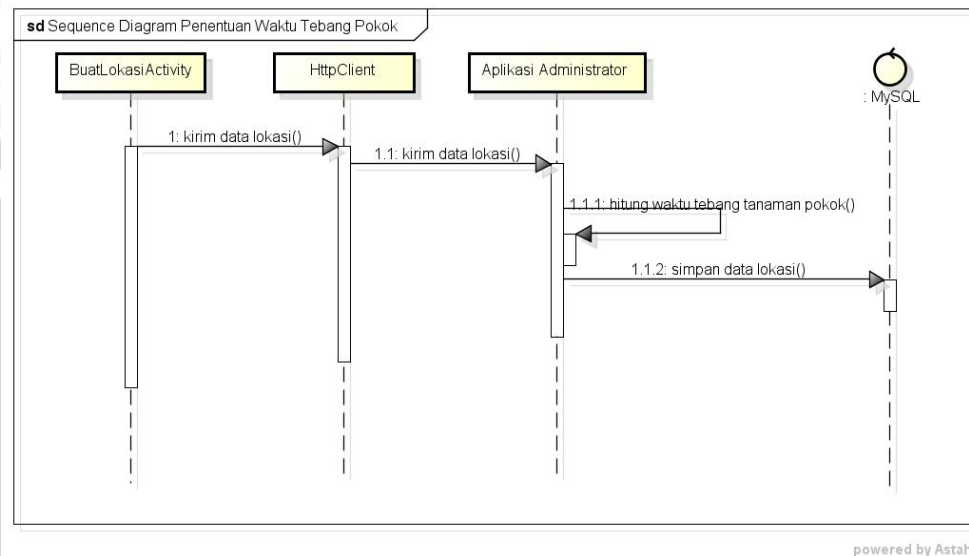


Gambar 4.32 *Sequence diagram* Penentuan Luas wilayah Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android
Sumber : [Perancangan]

1. Kelas BuatLokasiActivity mengirim data koordinat ke kelas HttpClient.
2. Kelas HttpClient meneruskan untuk mengirim data tersebut ke aplikasi *administrator*.
3. Aplikasi *administrator* melakukan perhitungan luas wilayah berdasarkan titik – titik koordinat dari wilayah tersebut.
4. Aplikasi *administrator* menyimpan data tersebut ke *database*.

D. *Sequence diagram* penentuan waktu tebang tanaman pokok

Sequences diagram penentuan waktu tebang tanaman pokok menggambarkan proses ketika aplikasi *administrator* menentukan waktu panen dari suatu wilayah hutan milik Perhutani berdasarkan data waktu tanam tanaman pokok dan jenis tanaman pokok yang telah dikirimkan oleh aplikasi *user*. Gambar 4.33 menjelaskan *sequences diagram* penentuan waktu tebang tanaman pokok.



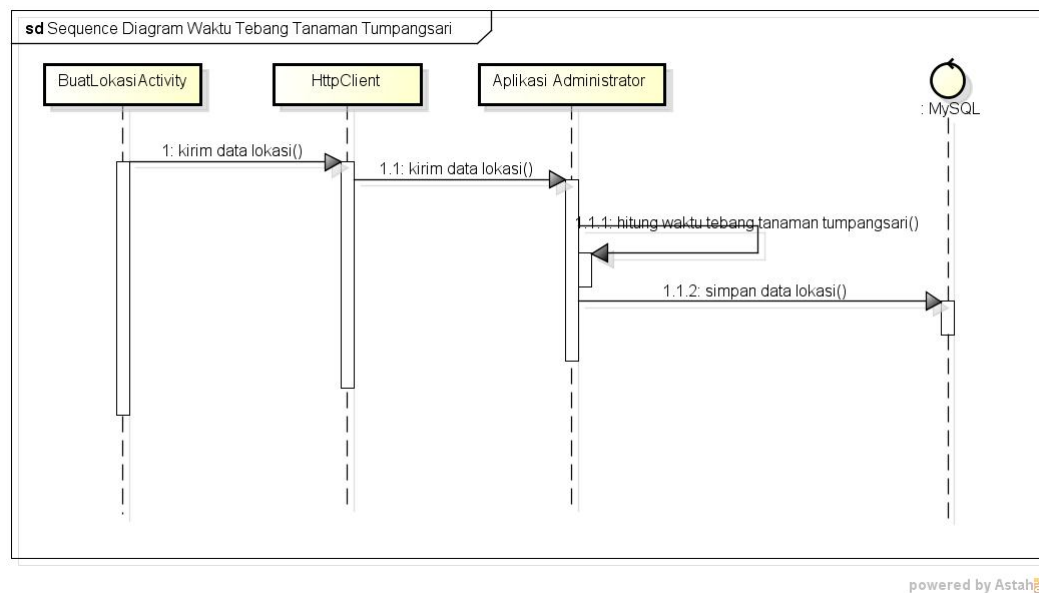
Gambar 4.33 *Sequence diagram* penentuan waktu tebang tanaman pokok Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Sumber : [Perancangan]

1. Kelas BuatLokasiActivity mengirim data lokasi ke kelas HttpClient.
2. Kelas HttpClient meneruskan untuk mengirim data tersebut ke aplikasi *administrator*.
3. Aplikasi *administrator* melakukan perhitungan waktu tebang tanaman pokok berdasarkan waktu tanam tanaman pokok dan jenis tanaman pokok dari wilayah tersebut.
4. Aplikasi *administrator* menyimpan data tersebut ke *database*.

E. Sequence diagram penentuan waktu tebang tanaman tumpangsari

Sequences diagram penentuan waktu tebang tanaman tumpangsari menggambarkan proses ketika aplikasi *administrator* menentukan waktu panen tumpangsari dari suatu wilayah hutan milik Perhutani berdasarkan data waktu tanam tanaman tumpangsari dan jenis tanaman tumpangsari yang telah dikirimkan oleh aplikasi *user*. Gambar 4.34 menjelaskan *sequences diagram* penentuan waktu tebang tanaman tumpangsari.



Gambar 4.34 Sequence diagram penentuan waktu tebang tanaman tumpangsari Perangkat Lunak Pemetaan Lahan Hutan Berbasis Android

Sumber : [Perancangan]

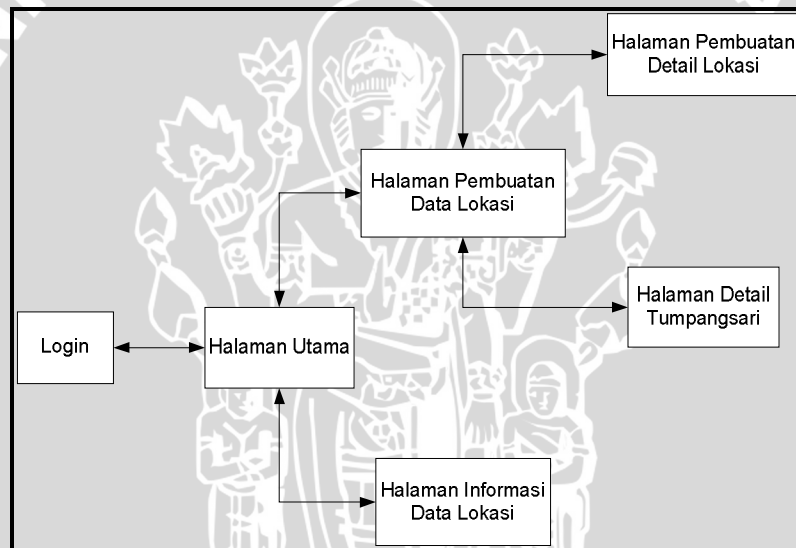
1. Kelas BuatLokasiActivity mengirim data lokasi ke kelas HttpClient.
2. Kelas HttpClient meneruskan untuk mengirim data tersebut ke aplikasi *administrator*.
3. Aplikasi *administrator* melakukan perhitungan waktu tebang tanaman tumpangsari berdasarkan waktu tanam tanaman tumpangsari dan jenis tanaman tumpangsari dari wilayah tersebut.
4. Aplikasi *administrator* menyimpan data tersebut ke *database*.

4.2.4 Perancangan Antarmuka

Pada bagian ini akan dijelaskan tentang perancangan antarmuka perangkat lunak Pemetaan Lahan Hutan berbasis Android. Antarmuka perangkat lunak ini akan digunakan oleh pengguna untuk berinteraksi dengan sistem perangkat lunak Pemetaan Lahan Hutan berbasis Android. Antarmuka perangkat lunak ini dibagi menjadi dua, yaitu antarmuka untuk aplikasi *user* dan aplikasi *administrator*.

4.1.4.1 Perancangan Antarmuka Aplikasi User

Antarmuka pada aplikasi pengguna berupa layout Android yang memudahkan pengguna untuk menggunakan aplikasi Pemetaan Lahan Hutan berbasis Android.



Gambar 4.35 Sitemap Antarmuka Aplikasi User

Sumber : [Perancangan]

1. Halaman Login

Halaman login merupakan salah satu antarmuka pengguna untuk aplikasi *user* yang berfungsi untuk *user* dalam melakukan *login*. Gambar 4.36 di bawah ini akan menunjukkan perancangan tampilan antarmuka dari halaman *login*.



Gambar 4.36 Halaman *Login* Pada Aplikasi *User*
Sumber : [Perancangan]

Keterangan Gambar 4.36:

1. Form untuk mengisi *username*.
2. Form untuk mengisi *password*.
3. Tombol *Login* untuk melakukan proses *login*.
4. Pesan “sukses” jika berhasil melakukan *login* dan pesan “gagal” jika tidak berhasil melakukan *login*.

2. **Halaman Utama**

Halaman utama adalah halaman untuk menampilkan data – data klien yang telah dicatat sebelumnya. Gambar 4.37 di bawah ini akan menunjukkan perancangan tampilan antarmuka dari halaman utama.



Gambar 4.37 Halaman Utama Pada Aplikasi User

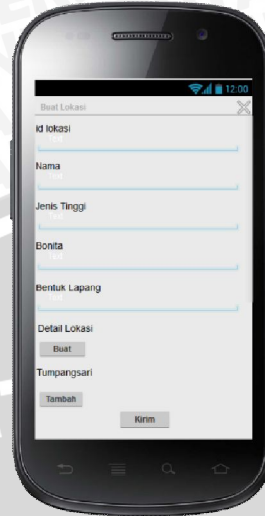
Sumber : [Perancangan]

Keterangan Gambar 4.37:

1. Tombol “Buat Lokasi” untuk melakukan proses pencatatan data lokasi pada halaman pengisian *form*.
2. Tombol “Info Lokasi” untuk melihat informasi lokasi yang telah di buat.
3. Tanda “Silang” merupakan proses untuk melakukan logout.

3. Halaman Pembuatan Data Lokasi

Halaman pengisian *form* adalah halaman untuk mencatat data – data lokasi. Gambar 4.38 di bawah ini akan menunjukkan perancangan tampilan antarmuka dari halaman pengisian *form*.



Gambar 4.38Halaman Pengisian *form* tambah lokasi Pada Aplikasi *User*

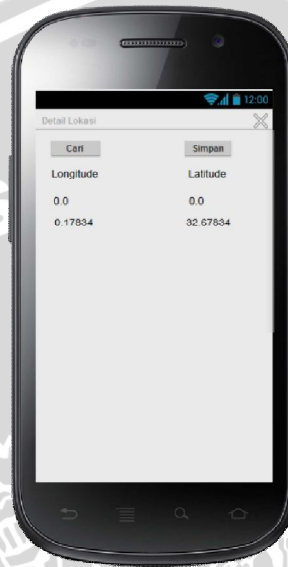
Sumber : [Perancangan]

Keterangan Gambar 4.38:

1. *Form* untuk mengisi id lokasi.
2. *Form* untuk mengisi nama lokasi.
3. *Form* untuk mengisi jenis ketinggian.
4. *Form* untuk mengisi bonita.
5. *Form* untuk mengisi bentuk lapang
6. Tombol “Buat” pada *form* Detail Lokasi merupakan Tombol *submit* untuk menuju ke halaman detail lokasi.
7. Tombol “Tambah” merupakan kondisi apakah lahan tersebut dipakai sebagai tumpangsari.
8. Tombol “Submit” untuk melakukan proses pencatatan dan penyimpanan data lokasi.
9. Tanda “Silang” merupakan proses untuk melakukan logout.

4. Halaman Pembuatan Detail Lokasi

Halaman detail lokasi adalah halaman untuk mencari koordinat lahan hutan milik Perhutani. Gambar 4.39 di bawah ini akan menunjukkan perancangan tampilan antarmuka dari halaman pembuatan detail lokasi.



Gambar 4.39 Halaman pembuatan detail lokasi

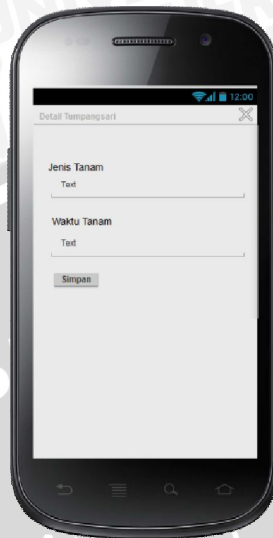
Sumber : [Perancangan]

Keterangan Gambar 4.39:

1. Tombol “Cari” untuk melakukan proses pencarian koordinat pada lahan hutan.
2. Tombol “Simpan” untuk melakukan proses penyimpanan data koordinat lokasi pada smartphone Android.
3. Tanda “Silang” merupakan proses untuk melakukan logout.

5. Halaman Pembuatan Detail Tumpangsari

Halaman detail tumpangsari adalah halaman untuk membuat informasi dari detail tumpangsari yang dipakai oleh warga yang memakai hutan milik Perhutani. Gambar 4.40 di bawah ini akan menunjukkan perancangan tampilan antarmuka dari halaman pembuatan detail tumpangsari.



Gambar 4.40 Halaman pembuatan detail tumpangsari

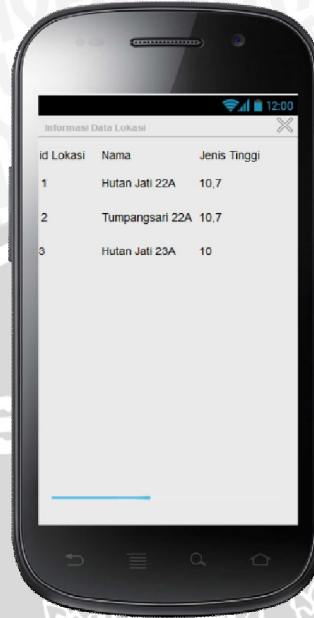
Sumber : [Perancangan]

Keterangan Gambar 4.40:

1. *Form* untuk mengisi Jenis tanam.
2. *Form* untuk mengisi Waktu tanam.
3. Tombol “Simpan” untuk melakukan proses penyimpanan data pada *Smartphone* Android.
4. Tanda “Silang” merupakan proses untuk melakukan logout.

6. Halaman Informasi Data Lokasi

Halaman detail adalah halaman untuk melihat seluruh informasi data lokasi yang telah disimpan. Gambar 4.41 di bawah ini akan menunjukkan perancangan tampilan antarmuka dari halaman informasi data lokasi.



Gambar 4.41 Halaman informasi data lokasi

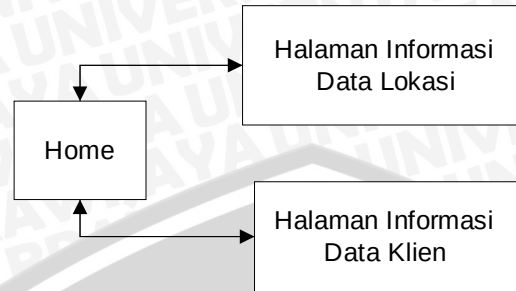
Sumber : [Perancangan]

Keterangan Gambar 4.41:

1. *Listview* yang berisi seluruh informasi data lokasi yang berfungsi untuk melihat informasi data lokasi yang telah di simpan di *smartphone* Android.
2. Tanda “Silang” merupakan proses untuk melakukan logout.

4.1.4.2 Perancangan Antarmuka Aplikasi Administrator

Antarmuka pada sisi *adminisnitrator* berupa halaman *web*. Perancangan antarmuka pada sisi *administrator* bertujuan untuk memudahkan *admin* untuk membuat akun pegawai, melihat, mencari, dan menghapus data lokasi di *server*. Gambar 4.42 menunjukkan *sitemap* dari aplikasi *administrator*.



Gambar 4.42 Sitemap Antarmuka Aplikasi Administrator

Sumber : [Perancangan]

1. Halaman Home

Halaman home merupakan halaman utama yang menampilkan gambar peta wilayah hutan Perhutani KPH Malang BKPH Sumbermanjing. Halaman ini berisi *link* untuk melihat Data Lokasi dan Data Klien. Di sebelah pojok kanan atas terdapat form yang berfungsi sebagai login Administrator. Gambar 4.43 menunjukkan rancangan antarmuka halaman home.

Logo dan Judul Web		Login	
Home	Data Lokasi	Data Klien	
Gambar Lokasi Hutan Perhutani KPH Malang BKPH Sumbermanjing			
Footer			

Gambar 4.43 Rancangan Antarmuka Halaman Home

Sumber : [Perancangan]

2. Halaman Data Lokasi

Halaman data lokasi adalah halaman yang menampilkan data-data lokasi yang disimpan di dalam basis data dalam bentuk tabel. Halaman ini juga berisi *link* untuk

melihat wilayah hutan yang telah didata dalam bentuk peta, detail tumpangsari, dan pembuatan waktu tebang. *Link* tersebut akan muncul apabila administrator telah login kedalam sistem. Gambar 4.44 menunjukkan rancangan antarmuka halaman data lokasi.

Nama Lokasi	Jenis Kertingian	Benita	Bentuk Lapangan	Kelas Hutan	Kepadatan Bidang	Waktu Kirim	Luas (ha)	Peta	Tumpangsari	Waktu Tebang
Hutan 1								Map	Link detail	Buat
Hutan 2										
Hutan 3										

Gambar 4.44 Rancangan Antarmuka Halaman Data Lokasi

Sumber : [Perancangan]

3. Halaman Data Klien

Halaman data klien adalah halaman yang menampilkan data-data klien yang disimpan di dalam basis data. Halaman ini juga berisi link untuk membuat data klien baru, mengubah data klien yang sudah ada, dan menghapus data klien. Gambar 4.45 menunjukkan rancangan antarmuka halaman data klien.

Login dan index Web		Login		
Home	Data Lokasi	Data Klien		
Buat Klien				
No	Nama	Username	Password	Keterangan
1				Edit Hapus
2				Edit Hapus
3				Edit Hapus
Footer				

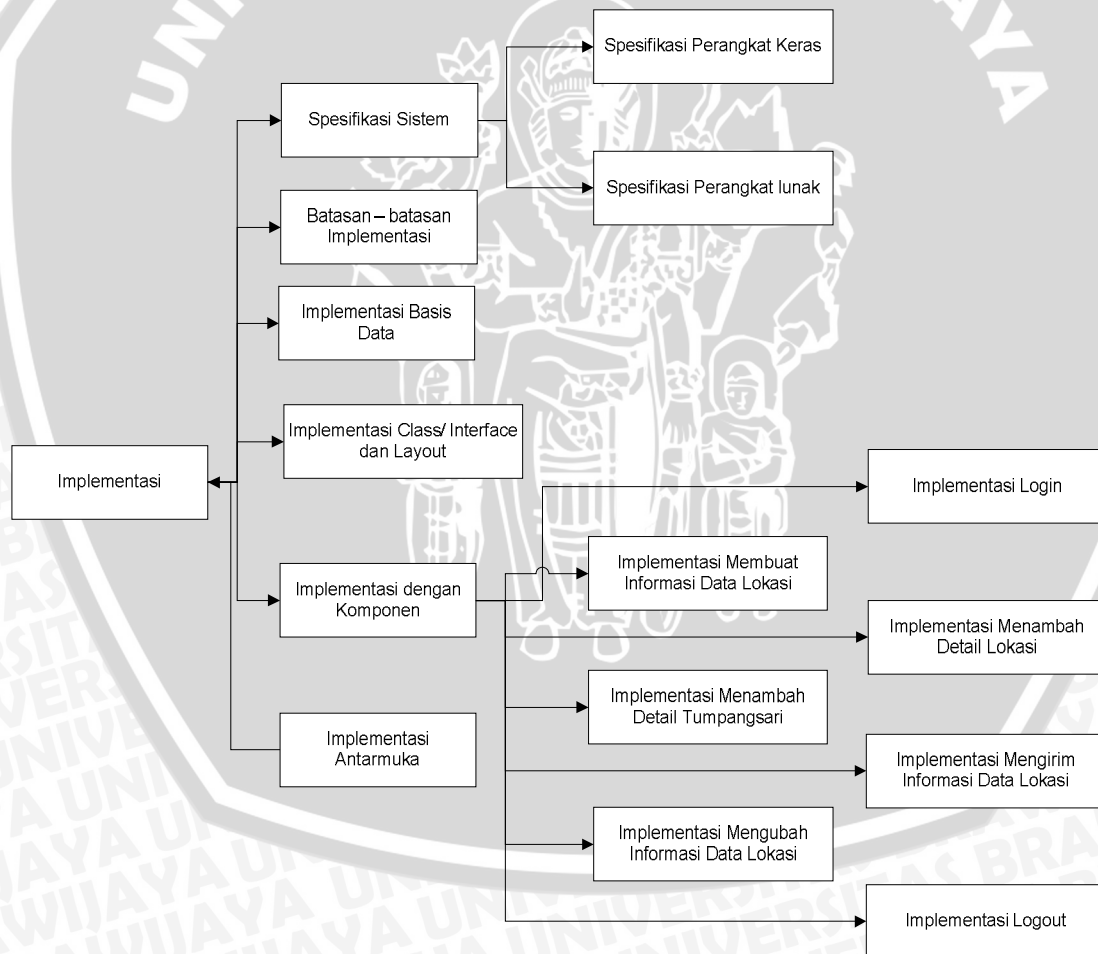
Gambar 4.45 Rancangan Antarmuka Halaman Data Klien

Sumber : [Perancangan]

BAB V

IMPLEMENTASI

Bab ini membahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak yang dibuat. Pembahasan terdiri atas penjelasan tentang spesifikasi sistem, batasan – batasan dalam implementasi, implementasi tiap *class* pada *file* program, implementasi dengan komponen, dan implementasi antarmuka. Tahap-tahap pembahasan implementasi yang dikerjakan digambarkan pada Gambar 5.1 berikut ini.



Gambar 5.1 Diagram Blok Implementasi
Sumber : [Implementasi]Spesifikasi Sistem

Hasil analisis kebutuhan yang telah diuraikan pada Bab 4 menjadi acuan untuk melakukan implementasi menjadi sebuah sistem yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1 Spesifikasi Sistem

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras meliputi spesifikasi prosesor, memori, hardisk, mother board, dan kartu grafis dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras

Laptop Asus Series A42J	
<i>Prosesor</i>	Inter(R) Core™ i5
<i>Memori (RAM)</i>	6 GB
<i>Hardisk</i>	500 GB
<i>MotherBoard</i>	ASUSTeK Computer Inc.
Kartu Grafis	AMD Radeon Graphic Processor
Ukuran layar	1366 x 768 piksel

Sumber : [Implementasi]

5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang dipakai dalam proses pengembangan perangkat lunak Pemetaan Lahan Hutan berbasis Android dijelaskan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

Laptop Asus Series A42J	
Sistem Operasi	Microsoft Windows 7 Ultimate 64-bit
Versi DirectX	DirectX 11
Bahasa Pemrograman	Java
<i>Software Development Kit</i>	Java Development Kit Update 21
<i>Programming Enviroment</i>	Java Runtime Enviroment 7
<i>Database Management System</i>	MySQL 5.1
<i>Integrated Development Enviroment</i>	Eclipse, Emulator Android, dan Notepad++

Sumber : [Implementasi]

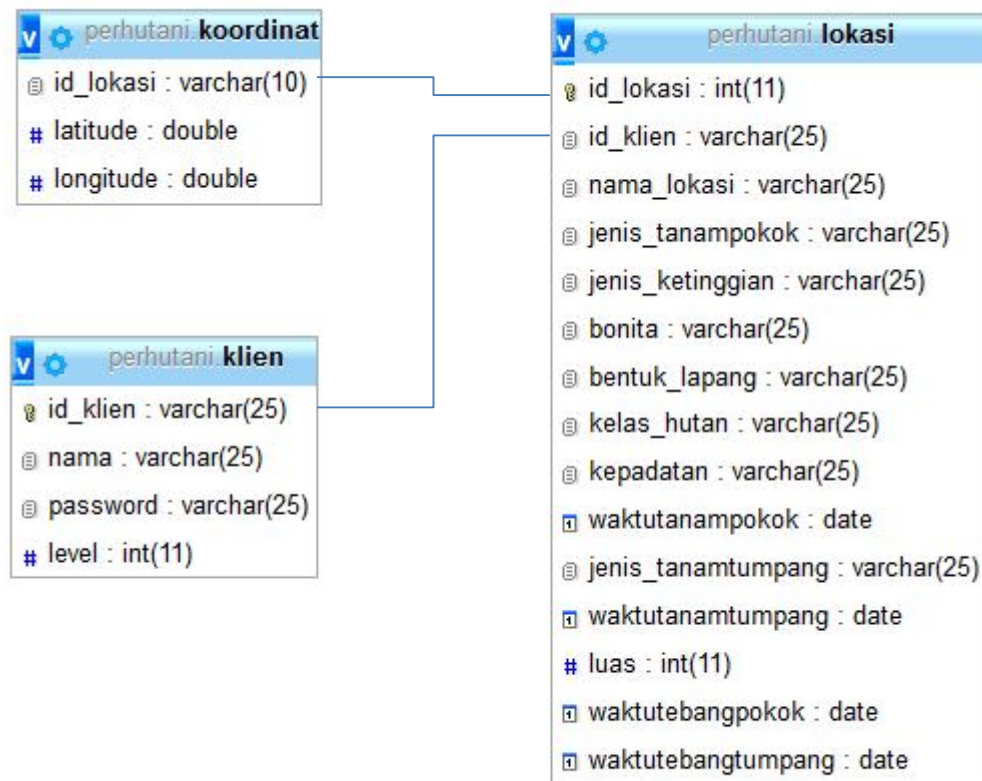
5.2 Batasan-Batasan Implementasi

Beberapa batasan-batasan dalam mengimplementasikan sistem adalah sebagai berikut:

- a. Pembuatan aplikasi Pemetaan Lahan Hutan berbasis Android ini dikerjakan dengan bahasa Pemrograman Java di lingkungan sistem operasi android.
- b. Sistem operasi android yang digunakan minimal Android OS, v2.3.6 (Gingerbread).
- c. Tingkat akurasi pengiriman data ke *server* tergantung kecepatan akses internet telepon cerdas pengguna.

5.3 Implementasi Basis Data

Implementasi penyimpanan data dilakukan dengan *database management system* MySQL pada aplikasi *administrator* dan *database management system*. Hasil implementasi penyimpanan data ini berupa *script – script* SQL. Hasil implementasi SQL pada *database* ini dimodelkan dalam diagram konseptual *entity relationship*.



Gambar 5.2 Diagram *Entity relationship* Aplikasi Pemetaan Lahan Hutan berbasis Android

Sumber : [Implementasi]

5.4 Implementasi Class/Interface dan Layout

Implementasi *class/interface* merupakan proses implementasi dari hasil perancangan pada bab perancangan Sub Bab 4.2.2 yang direalisasikan pada sebuah *file* program dengan ekstensi *.java. Tabel 5.3 menjelaskan mengenai pasangan antara *class* dengan *file* program yang digunakan untuk mengimplementasikannya.

Tabel 5.3 Implementasi *class* pada kode program *.java

No.	Package	Nama Class	Nama File Program
1	perhutani.activity	Login	Login.java

2	perhutani.activity	Home	Home.java
3	perhutani.activity	BuatLokasi	BuatLokasi.java
4	perhutani.activity	BuatLokasiTumpang	BuatLokasiTumpang.java
5	perhutani.activity	DetailLokasi	DetailLokasi.java
6	perhutani.activity	DetailTumpangsari	DetailTumpangsari.java
7	perhutani.activity	EditLokasi	EditLokasi.java
8	perhutani.activity	InfoData	InfoData.java
9	perhutani.adapter	Lokasi_objek	Lokasi_objek.java
10	perhutani.helper	DBMS	DBMS.java
11	perhutani.helper	GpsService	GpsService.java
12	perhutani.koneksi	HttpClient	HttpClient.java

Sumber : [Implementasi]

Tabel 5.4 merupakan implementasi *layout* yang akan digunakan di dalam aplikasi. *Layout* ini berfungsi untuk mengatur tampilan aplikasi di dalam sistem operasi android yang akan digunakan pengguna untuk berinteraksi dengan sistem. Pada operasi sistem android, *layout* dibuat menggunakan *tag-tag* XML dengan ekstensi *.xml.

Tabel 5.4 Implementasi Layout

No.	Folder	Nama Layout	Nama File Program
1	/res/layout	login	login.xml
2	/res/layout	home	home.xml
3	/res/layout	buatlokasi	buatlokasi.xml
4	/res/layout	buatlokasitumpang	buatlokasitumpang.xml
5	/res/layout	editlokasi	editlokasi.xml
6	/res/layout	infodata	infodata.xml
7	/res/layout	detaillokasi	detaillokasi.xml

8	/res/layout	detailtumpangsari	detailtumpangsari.xml
---	-------------	-------------------	-----------------------

Sumber : [Implementasi]

5.5 Implementasi dengan Komponen

Perangkat lunak Pemetaan Lahan Hutan berbasis Android memiliki beberapa fitur yang terbagi dalam beberapa fungsi. Beberapa fitur tersebut diproses menggunakan *resource* pada komponen-komponen yang telah ada atau menggunakan metode CBSE (*Component Based Software Engineering*). Pada penulisan skripsi ini mencantumkan beberapa *class* hasil implementasi dari perancangan pada bab IV sub bab Perancangan Model (*Class*). Beberapa *class* yang akan dicantumkan dalam penulisan makalah skripsi ini hanya untuk algoritma dari beberapa proses (operasi) saja sehingga tidak semua *method* akan dicantumkan.

5.4.1 Implementasi Login

Proses login dilakukan ketika user pertama kali masuk ke aplikasi Pemetaan Lahan Hutan berbasis Android. Operasi ini mengirimkan *username* dan *password* merupakan proses implementasi dari perancangan dengan komponen. Implementasi cuplikan *code login* dapat dilihat pada Gambar 5.3.

```

1  public void loginClick(View v) {
2      client.addParam("username", user.getText().toString());
3      client.addParam("password", pass.getText().toString());
4      client.setUrl("login.php");
5      client.executeRequest();
6      String response = client.getResponse();
7      if (response.equalsIgnoreCase("1")) {
8          prefs = getSharedPreferences("myData", MODE_PRIVATE);
9          editor = prefs.edit();
10         editor.putString("username", user.getText().toString());
11         editor.putString("password", pass.getText().toString());
12         editor.commit();
13         Intent myIntent = new Intent(this, Home.class);
14         startActivityForResult(myIntent, 1);
15         Toast.makeText(getApplicationContext(), "Username: " +
16         user.getText().toString() + "\nSukses", Toast.LENGTH_LONG).show();

```



```

17     finish();
18     } else {
19         Toast.makeText(getApplicationContext(),"Username: " +
20         user.getText().toString() + "\nGagal",
21         Toast.LENGTH_LONG).show();
22         pass.setText("");
23     }
24 }
25

```

Gambar 5.3 Implementasi Cuplikan Code Login
Sumber : [Implementasi]

Penjelasan cuplikan code dari Gambar 5.3 yaitu:

1. Baris 3-6 digunakan untuk mengirim data *username* dan *password* dari aplikasi *user* ke aplikasi *administrator*.
2. Baris 7-25 digunakan untuk menerima respon dari aplikasi *administrator*. Apabila kembalian bernilai 1 maka akan ada pesan sukses dan apabila kembalian bernilai 0 maka proses login akan gagal.
3. Baris 14-15 digunakan untuk menampilkan halaman Home.

5.4.2 Implementasi Membuat Informasi Data Lokasi

Proses ini merupakan proses untuk menyimpan data lokasi yang telah diinputkan oleh *user* ke dalam basis data aplikasi *user*. Implementasi cuplikan code membuat informasi data lokasi dapat dilihat pada Gambar 5.4.

```

1  publicvoid uploadClick(View v) {
2      db = DBMS.getInstance(this);
3      if (sid.contentEquals("")) {
4          Toast.makeText(this, "Data Id masih kosong",
5          Toast.LENGTH_SHORT).show();
6      } else {
7          db.insertLokasi(sid,onUser, snama, textspin, sjenis_tinggi,
8          sbonita, sbentuk_lapang,
9          skelas_hutan,skepadatan,
10         swaktutanampokok,i,postpok);
11         Intent intent_upload = new Intent(this, Home.class);
12         startActivity(intent_upload);

```

13	}
----	---

Gambar 5.4 Implementasi Cuplikan Code Membuat Informasi Data Lokasi

Sumber : [Implementasi]

Penjelasan cuplikan *code* dari Gambar 5.4 yaitu:

1. Baris 2 digunakan untuk menginisialisasikan kelas basis data dari aplikasi *user*.
2. Baris 3-4 digunakan untuk kondisi agar kolom id lokasi wajib diisi.
3. Baris 7-11 digunakan untuk menyimpan data lokasi pada basis data pada aplikasi *user*.
4. Baris 12-13 digunakan untuk menampilkan halaman Home.

5.4.3 Implementasi Menambah Detail Tumpangsari

Proses ini merupakan proses untuk menginput data detail tumpangsari lalu data di kirim ke halaman buat lokasi. Implementasi cuplikan *code* menambah detail tumpangsari dapat dilihat pada Gambar 5.5.

1	public void detailtumpangsariClick(View v) {
2	Intent intent_detail = new Intent(this , BuatLokasiTumpang. class);
3	intent_detail.putExtra("jenis", textspin1);
4	intent_detail.putExtra("waktu", swaktu);
5	startActivity(intent_detail);
6	}
7	
8	

Gambar 5.5 Implementasi Cuplikan Code Menambah Detail Tumpangsari

Sumber : [Implementasi]

Penjelasan cuplikan *code* dari Gambar 5.5 yaitu:

1. Baris 2-3 digunakan untuk menampilkan halaman buat lokasi.
2. Baris 4-5 digunakan untuk mengirim data detail tumpangsari ke halaman buat lokasi.

5.4.4 Implementasi Menambah Detail Lokasi

Proses ini berfungsi untuk mencari data koordinat lokasi dan disimpan kedalam basis data aplikasi *user*. Implementasi cuplikan *code* menambah detail lokasi dapat dilihat pada Gambar 5.6.

```

1  public void carikoordinatClick(View v) {
2      if (gps.canGetLocation()) {
3          latitude = gps.getLatitude();
4          longitude = gps.getLongitude();
5          db.insertKoordinat(sid, longitude, latitude);
6          updateListKoordinat();
7          Toast.makeText(getApplicationContext(), "Lokasi mu latitude: " + latitude +
8              " Longitude : " + longitude + " kembalian ", Toast.LENGTH_LONG).show();
9      } else {
10         gps.showSettingAlert();
11     }
12 }
13 public void simpankoordinatClick(View v) {
14     Intent intent_simpan = new Intent(this, BuatLokasiTumpang.class);
15     startActivity(intent_simpan);
16 }
17 }
18 private void updateListKoordinat() {
19     koordinat_list = new ArrayList<String>();
20     koordinat_adapter = new ArrayAdapter<String>(this,
21     android.R.layout.simple_list_item_1, db.getKoordinatById(sid));
22     lv_koordinat.setAdapter(koordinat_adapter);
23 }
24
25

```

Gambar 5.6 Implementasi Cuplikan Code Menambah Detail Lokasi

Sumber : [Implementasi]

Penjelasan cuplikan *code* dari Gambar 5.6 yaitu:

1. Baris 2 digunakan untuk pengecekan fungsi GPS pada Android.
2. Baris 3-4 digunakan untuk mendapatkan nilai koordinat lokasi dari satelit.
3. Baris 5 digunakan untuk menyimpan data detail lokasi pada basis data pada aplikasi *user*.
4. Baris 15-20 digunakan untuk menampilkan halaman buat lokasi.

- Baris 21-26 digunakan untuk proses pengambilan data detail lokasi dari basis data pada aplikasi *user* dan ditampilkan dalam bentuk Listview.

5.4.5 Implementasi Mengubah Informasi Data Lokasi

Proses ini merupakan proses untuk mengubah data lokasi yang telah disimpan kedalam basis data aplikasi *user*. Implementasi cuplikan *code* mengubah informasi data lokasi dapat dilihat pada Gambar 5.7.

```

1  lv_lokasi = (ListView) findViewById(R.id.lv_lokasi);
2  lv_lokasi.setOnItemClickListener(new OnItemClickListener() {
3  public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
4      data = db.ambilSemuaLokasi();
5      Intent intent_info = new Intent(getApplicationContext(),
6                          BuatLokasi.class);
7      intent_info.putExtra("id_lokasi", sid);
8      intent_info.putExtra("nama_lokasi", snama);
9      intent_info.putExtra("jenis_tanampokok", sjenis_tanampokok);
10     intent_info.putExtra("jenis_ketinggian", sjenis_tinggi);
11     intent_info.putExtra("bonita", sbonita);
12     intent_info.putExtra("bentuk_lapang", sbentuk_lapang);
13     intent_info.putExtra("kelas_hutan", skelas_hutan);
14     intent_info.putExtra("kepadatan_bidang", skepadatan);
15     intent_info.putExtra("waktutanampokok", swaktutanampokok);
16     intent_info.putExtra("jenis", jenis);
17     intent_info.putExtra("waktu", waktu);
18     intent_info.putExtra("i", i);
19     intent_info.putExtra("pospok", pospok);
20     intent_info.putExtra("postum", postum);
21     startActivity(intent_info);
22 }
23
24
```

Gambar 5.7 Implementasi Cuplikan *Code* Mengubah Informasi Data Lokasi
Sumber : [Implementasi]

Penjelasan cuplikan *code* dari Gambar 5.7 yaitu:

- Baris 1-2 digunakan untuk menginisialisasikan list data lokasi.
- Baris 6 digunakan untuk mengambil semua data lokasi dari basis data aplikasi *user* berdasarkan posisi list yang di pilih.

3. Baris 7-24 digunakan untuk mengirimkan data lokasi ke halaman buat lokasi untuk di ubah sesuai kebutuhan dan disimpan lagi di basis data aplikasi *user*.

5.4.6 Implementasi Mengirim Informasi Data Lokasi

Proses ini merupakan proses untuk mengirim data lokasi yang telah diinputkan oleh *user* ke aplikasi *administrator*. Implementasi cuplikan *code* mengirim informasi data lokasi dapat dilihat pada Gambar 5.8.

```

1  publicvoid uploadClick(View v) {
2      if (sid.contentEquals("")) {
3          Toast.makeText(this, "Data Id masih kosong",
4              Toast.LENGTH_SHORT).show();
5      } else {
6          client.setUrl("terimadata.php");
7          client.addParam("id_lokasi", sid);
8          client.addParam("id_klien", onUser);
9          client.addParam("nama_lokasi", snama);
10         client.addParam("jenis_tanampokok", textspin);
11         client.addParam("jenis_ketinggian", sjenis_tinggi);
12         client.addParam("bonita", sbonita);
13         client.addParam("bentuk_lapang", sbentuk_lapang);
14         client.addParam("kelas_hutan", skelas_hutan);
15         client.addParam("kepadatan", skepadatan);
16         client.addParam("waktutanampokok", swaktutanampokok);
17         client.addParam("i", i);
18         client.addParam("jml",
19             String.valueOf(cursor2.getCount()));
20         Cursor cursor2 = db.getKoordinat_id(sid);
21         for (int i = 0; i < cursor2.getCount(); i++) {
22             cursor2.moveToNext();
23             client.addParam("longitude" + i, cursor2.getString(1));
24             client.addParam("latitude" + i, cursor2.getString(2));
25         }
26         client.executeRequest();
27         Intent intent_upload = new Intent(this, Home.class);
28         startActivity(intent_upload);
29     }
30 }
```

Gambar 5.8 Implementasi Cuplikan Code Mengirim Informasi Data Lokasi
Sumber : [Implementasi]

Penjelasan cuplikan *code* dari Gambar 5.8 yaitu:

1. Baris 2-4 digunakan untuk kondisi agar kolom id lokasi wajib diisi.
2. Baris 6 digunakan untuk mengatur tujuan pengiriman data dari aplikasi *user* ke aplikasi *administrator*.
3. Baris 7-19 digunakan untuk mengirimkan data lokasi ke aplikasi *administrator*.
4. Baris 20-26 digunakan untuk mengambil seluruh data detail lokasi yang berada di basis data aplikasi *user* berdasarkan id lokasinya, lalu dikirimkan ke aplikasi *administrator*.

5.4.7 Implementasi Membuat Waktu Tebang Tanaman Pokok

Proses ini merupakan proses untuk membuat waktu tebang tanaman pokok dengan menggunakan inputan waktu tanam tanaman pokok dan jenis tanaman pokok yang berasal dari aplikasi *user*. Implementasi cuplikan *code* membuat waktu tebang tanaman pokok dapat dilihat pada Gambar 5.9.

1	\$waktutanampokok = \$_POST['waktutanampokok'];
2	\$jenis_tanampokok = \$_POST['jenis_tanampokok'];
3	\$waktutanampokok1 = str_replace('-', '/', \$waktutanampokok);
4	
5	if(\$jenis_tanampokok == 'jati'){
6	\$tebangpokok = date('Y-m-d',strtotime(\$waktutanampokok1 . "+20 years"));
7	}elseif (\$jenis_tanampokok == 'mahoni'){
8	\$tebangpokok = date('Y-m-d',strtotime(\$waktutanampokok1 . "+30 years"));
9	}elseif (\$jenis_tanampokok == 'sengon'){
10	\$tebangpokok = date('Y-m-d',strtotime(\$waktutanampokok1 . "+6 years"));
11	}

Gambar 5.9 Implementasi Cuplikan *Code* Membuat Waktu Tebang Tanaman Pokok
Sumber : [Implementasi]

Penjelasan cuplikan *code* dari Gambar 5.9 yaitu:

1. Baris 1-2 digunakan untuk mengambil inputan dari method post dan disimpan didalam variabel.
2. Baris 5-14 digunakan untuk kondisi untuk menentukan waktu tebang tanaman pokok berdasarkan jenis tanaman pokoknya.

5.4.8 Implementasi Membuat Waktu Tebang Tanaman Tumpangsari

Proses ini merupakan proses untuk membuat waktu tebang tanaman tumpangsari dengan menggunakan inputan waktu tanam tanaman tumpangsari dan jenis tanaman tumpangsari yang berasal dari aplikasi *user*. Implementasi cuplikan *code* membuat waktu tebang tanaman pokok dapat dilihat pada Gambar 5.10.

1	\$jenis_tanamtumpang = \$_POST['jenistanamtumpang'];
2	\$waktutanamtumpang = \$_POST['waktutanamtumpang'];
3	\$waktutanamtumpang1 = str_replace('-', '/', \$waktutanamtumpang);
4	
5	\$tebangtumpang = date('Y-m-d',strtotime(\$waktutanamtumpang1 . "+2 years"));
6	

Gambar 5.10 Implementasi Cuplikan *Code* Membuat Waktu Tebang Tanaman Tumpangsari

Sumber : [Implementasi]

Penjelasan cuplikan *code* dari Gambar 5.10 yaitu:

1. Baris 1-2 digunakan untuk mengambil inputan dari method post dan disimpan didalam variabel.
2. Baris 5-6 digunakan untuk kondisi untuk menentukan waktu tebang tanaman tumpangsari berdasarkan jenis tanaman tumpangsarinya.

5.4.9 Implementasi Membuat Luas Wilayah

Proses ini merupakan proses untuk membuat luas wilayah suatu lokasi berdasarkan data koordinat yang berasal dari aplikasi *user*. Implementasi cuplikan *code* membuat waktu tebang tanaman pokok dapat dilihat pada Gambar 5.11.

1	function codeAddress() {
2	var area = google.maps.geometry.spherical.computeArea([
3	<?php if (isset(\$_GET['id']) AND trim(\$_GET['id']) != " ") :
4	\$q = mysql_query("select * from koordinat where
5	id_lokasi='\$_GET[id]') or die(mysql_error());
6	if (mysql_num_rows(\$q) != 0) :
7	\$i = 0;
8	\$last = ";

```

9      while ( $row = mysql_fetch_assoc($q) ) :
10         if ( $i == 0 ) $last = 'new
11         google.maps.LatLng('.$row['latitude'].', '.$row['longitude'].');
12         echo 'new google.maps.LatLng('.$row['latitude'].', '.$row['longitude'].').';
13         $i++;
14     endwhile;
15     echo $last;
16     endif;
17     endif;?>]);
18     document.getElementById("tset").innerHTML = "Luas : "+area+" m<sup>2</sup>";}
19     window.onload = codeAddress;
20
21

```

Gambar 5.11 Implementasi Cuplikan Code Membuat Luas Wilayah
Sumber : [Implementasi]

Penjelasan cuplikan *code* dari Gambar 5.11 yaitu:

1. Baris 2-3 digunakan untuk variabel yang berfungsi sebagai tempat perhitungan luas wilayah dan hasilnya disimpan didalam variabel tersebut.
2. Baris 4-18 digunakan untuk mengambil data koordinat dari *database* aplikasi *administrator* yang akan digunakan sebagai inputan koordinat untuk menghitung luas wilayah.
3. Baris 19-20 digunakan untuk menampilkan hasil dari perhitungan luas wilayah.
4. Baris 21 digunakan untuk mengaktifkan fungsi perhitungan luas wilayah.

5.4.10 Implementasi Logout

Proses proses ini merupakan proses untuk keluar dari aplikasi *user*.

Implementasi cuplikan *code* logout dapat dilihat pada Gambar 5.12.

```

1      publicvoid logoutClick(View v) {
2          prefs = getSharedPreferences("myData", MODE_PRIVATE);
3          editor = prefs.edit();
4          editor.putString("username", null);
5          editor.putString("password", null);
6          editor.commit();
7          Intent myIntent = new Intent(this, Login.class);
8          startActivity(myIntent);
9      }

```

Gambar 5.12 Implementasi Cuplikan *Code* Logout
Sumber : [Implementasi]

Penjelasan cuplikan *code* dari Gambar 5.12 yaitu:

1. Baris 2-3 digunakan untuk mengatur *session* pada mode *private* dan diijinkan untuk proses *editing*.
2. Baris 4-5 digunakan untuk mengosongkan nilai *session*.
3. Baris 7 digunakan untuk kembali ke halaman login.

5.6 Implementasi Antarmuka

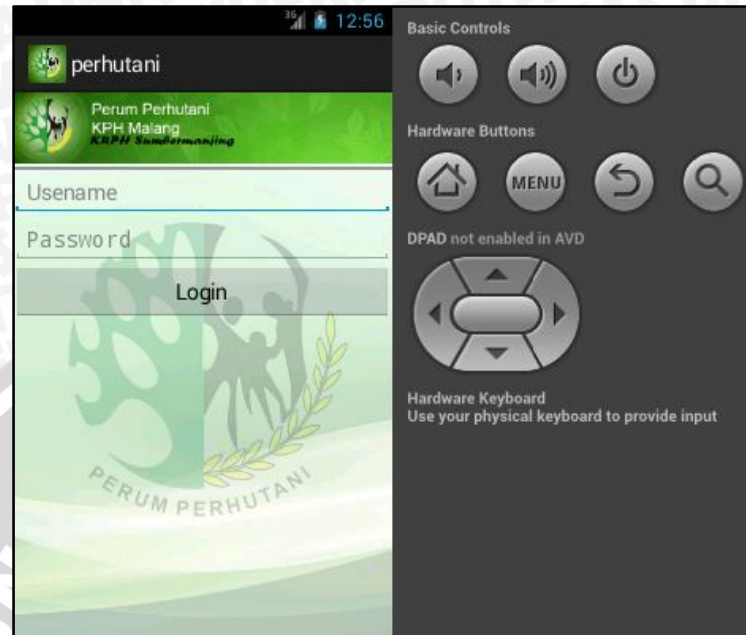
Implementasi antarmuka berdasarkan perancangan antarmuka pada bab perancangan Sub Bab 4.3. Antarmuka perangkat lunak ini dibagi menjadi 2, yaitu antarmuka aplikasi *user* dan aplikasi *administrator*.

5.5.1 Implementasi Antarmuka Aplikasi User

Antarmuka pengguna aplikasi *user* merupakan sebuah perangkat lunak Pemetaan Lahan Hutan berbasis Android. Antarmuka pengguna untuk aplikasi *user* terdiri atas halaman login, halaman *home*, halaman buat lokasi, halaman detail lokasi, halaman detail tumpangsari, dan halaman informasi data lokasi.

1. Antarmuka Login

Antarmuka login merupakan halaman yang pertama diakses oleh *user* ketika masuk ke dalam aplikasi Pemetaan Lahan Hutan berbasis Android. Halaman ini memiliki kolom *username* dan *password* untuk diisi oleh *user*. Pada halaman ini juga terlihat tombol login yang harus ditekan oleh *user* jika ingin masuk ke dalam aplikasi Pemetaan Lahan Hutan berbasis Android. Gambar 5.13 menunjukkan antarmuka login.



Gambar 5.13 Antarmuka Login
Sumber : [Implementasi]

2. Antarmuka Halaman Home

Antarmuka halaman *home* memuat tombol buat lokasi dan informasi data. Antarmuka halaman *home* juga menyediakan tombol logout pada pojok kanan atas antarmuka. Gambar 5.14 merupakan antarmuka halaman utama aplikasi Pemetaan Lahan Hutan berbasis Android.



Gambar 5.14 AntarmukaHalamanHome
Sumber : [Implementasi]

3. Antarmuka Membuat Informasi Data Lokasi

Implementasi antarmuka membuat informasi data lokasi merupakan halaman yang digunakan untuk membuat informasi data lokasi baru. Antarmuka ini berisi *form* untuk menginputkan data lokasi. Antarmuka halaman membuat informasi data lokasi juga menyediakan tombol kirim dan logoutpada pojok kanan atas antarmuka. Gambar 5.15 menunjukkan antarmuka membuat informasi data lokasi



Gambar 5.15 Antarmuka Membuat Informasi Data Lokasi
Sumber :[Implementasi]

4. Antarmuka Detail Tumpangsari

Implementasi antarmuka detail tumpangsari merupakan halaman yang digunakan untuk membuat informasi data detail tumpangsari. Antarmuka ini berisi *form* untuk menginputkan data detail tumpangsari. Antarmuka halaman detail tumpangsari juga menyediakan logout pada pojok kanan atas antarmuka dan tombol simpan yang berfungsi untuk menyimpan informasi detail tumpangsari. Gambar 5.16 menunjukkan antarmuka detail tumpangsari.



Gambar 5.16 AntarmukaDetailTumpangsari
Sumber : [Implementasi]

5. Antarmuka Detail Lokasi

Implementasi antarmuka detail lokasi merupakan halaman yang digunakan untuk membuat informasi data detail lokasi yang berupa informasi titik – titik koordinat lokasi tersebut. Data detail lokasi akan ditampilkan dalam bentuk *list*. Antarmuka ini berisi 2 tombol, yaitu tombol cari dan simpan. Tombol cari berfungsi untuk mencari titik koordinat lokasi tersebut dan tombol simpan berfungsi untuk menyimpan data detail lokasi. Antarmuka halaman detail lokasi juga menyediakan logout pada pojok kanan atas antarmuka. Gambar 5.17 menunjukkan antarmuka detail lokasi.



Gambar 5.17 Antarmuka Detail Lokasi
Sumber : [Implementasi]

6. Antarmuka Informasi Data Lokasi

Implementasi antarmuka informasi data lokasi merupakan halaman yang digunakan untuk menampilkan data lokasi dalam bentuk *list*. Data yang akan ditampilkan adalah data id lokasi dan nama lokasi. Antarmuka halaman detail lokasi juga menyediakan logout pada pojok kanan atas antarmuka. Gambar 5.18 menunjukkan antarmuka informasi data lokasi.



Gambar 5.18 Antarmuka Informasi Data Lokasi
Sumber : [Implementasi]

5.5.2 Implementasi Antarmuka Aplikasi Administrator

Antarmuka pengguna aplikasi *administrator* merupakan sebuah perangkat lunak Pemetaan Lahan Hutan berbasis *Web*. Antarmuka pengguna untuk aplikasi *administrator* terdiri atas halaman data lokasi, halaman data klien, halaman buat klien, dan halaman lihat peta.

1. Antarmuka Halaman Data Lokasi

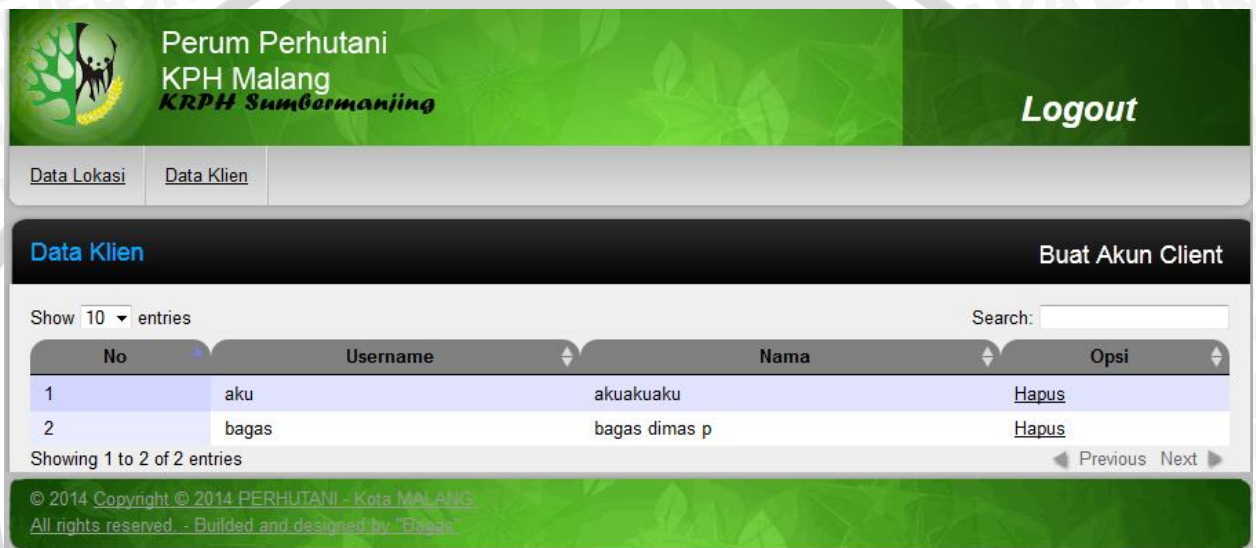
Antarmuka halaman data lokasi merupakan halaman yang pertama diakses ketika masuk ke dalam aplikasi *administrator*. Halaman ini menampilkan data lokasi dalam bentuk tabel. Data tersebut adalah id lokasi, nama lokasi, waktu tebang tanaman pokok, dan lihat peta. Gambar 5.19 menunjukkan antarmuka halaman data lokasi.



Gambar 5.19 Antarmuka Halaman Data Lokasi
Sumber : [Implementasi]

2. Antarmuka Halaman Data Klien

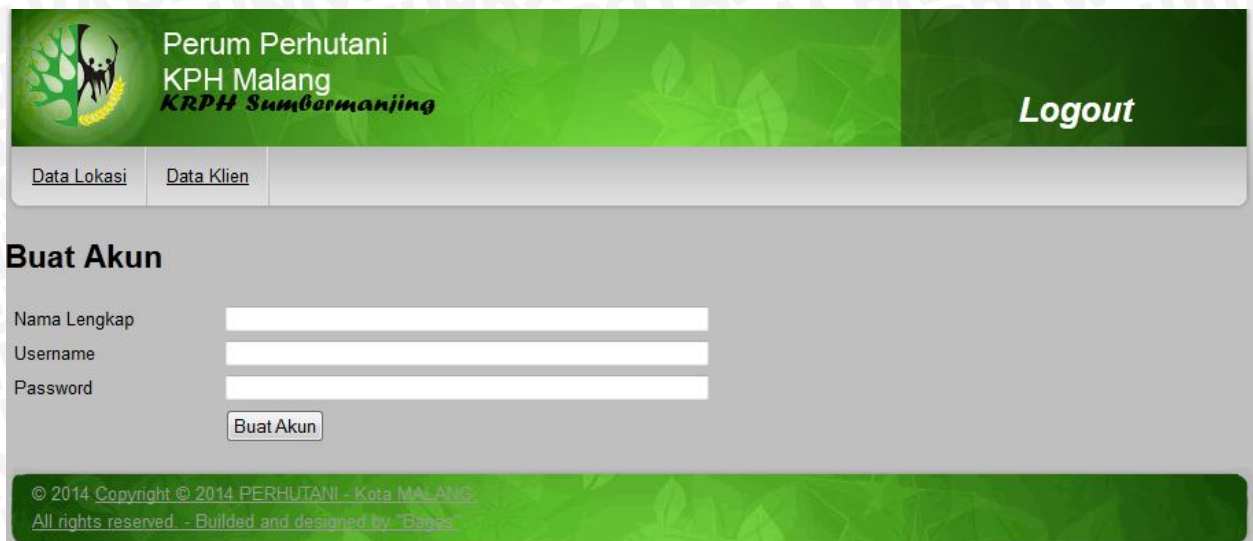
Antarmuka halaman data klien merupakan halaman yang digunakan untuk melihat informasi data klien yang ditampilkan dalam bentuk tabel. Data tersebut berisi *username*, nama lengkap klien, dan opsi untuk menghapus klien. Halaman ini memiliki tautan untuk membuat data klien baru. Gambar 5.20 menunjukkan antarmuka halaman data klien.



Gambar 5.20 Antarmuka Halaman Data Klien
Sumber : [Implementasi]

3. Antarmuka Halaman Buat Klien

Antarmuka halaman buat klien merupakan halaman yang digunakan untuk membuat data klien baru yang akan digunakan pada aplikasi *user*. Halaman ini berisi form *username*, *password*, dan nama lengkap dari klien. Gambar 5.21 menunjukkan antarmuka halaman buat klien.



Perum Perhutani
KPH Malang
KRPH Sumbermanjing

[Logout](#)

[Data Lokasi](#) [Data Klien](#)

Buat Akun

Nama Lengkap

Username

Password

© 2014 Copyright © 2014 PERHUTANI - Kota MALANG.
All rights reserved. - Bulided and designed by "Bangs"

Gambar 5.21 Antarmuka Halaman Buat Klien
Sumber : [Implementasi]

4. Antarmuka Halaman Lihat Peta

Antarmuka halaman lihat peta merupakan halaman yang digunakan untuk melihat titik koordinat lokasi yang sudah ditampilkan dalam wujud bidang / polygon didalam peta. Halaman ini juga menampilkan keterangan dari data lokasi tersebut. Data tersebut meliputi luas wilayah, id lokasi, nama lokasi, jenis tanaman pokok, jenis tanaman tumpangsari, ketinggian, bonita, bentuk lapangan, kelas hutan, kepadatan hutan, waktu tebang tanaman pokok, waktu tebang tanaman tumpangsari, dan petugas yang melakukan pendataan. Gambar 5.22 menunjukkan antarmuka halaman lihat peta.

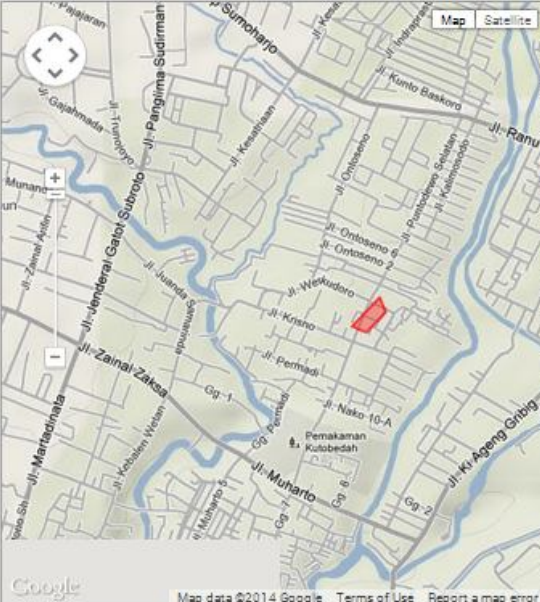


Perum Perhutani
 KPH Malang
 KRPH Sumbermanjing

Login

Data Lokasi

Peta Lokasi



KETERANGAN :

Luas : 7596.26930554092 m ²	
Anak Petak	: 1
Nama Lokasi	: hutan jati
Jenis Tanaman Pokok	: Jati
Jenis Tanaman Tumpangsari	: Palawija
Ketinggian	: 1000 mdpl
Bonita	: 10
Bentuk Lapangan	: datar
Kelas Hutan	: 10
Kepadatan Hutan	: 0.7 m ³ /ha
Waktu Tebang Tanaman Pokok	: 2034-04-19
Waktu Tebang Tanaman Tumpangsari	: 2016-04-19
Petugas	: bagas

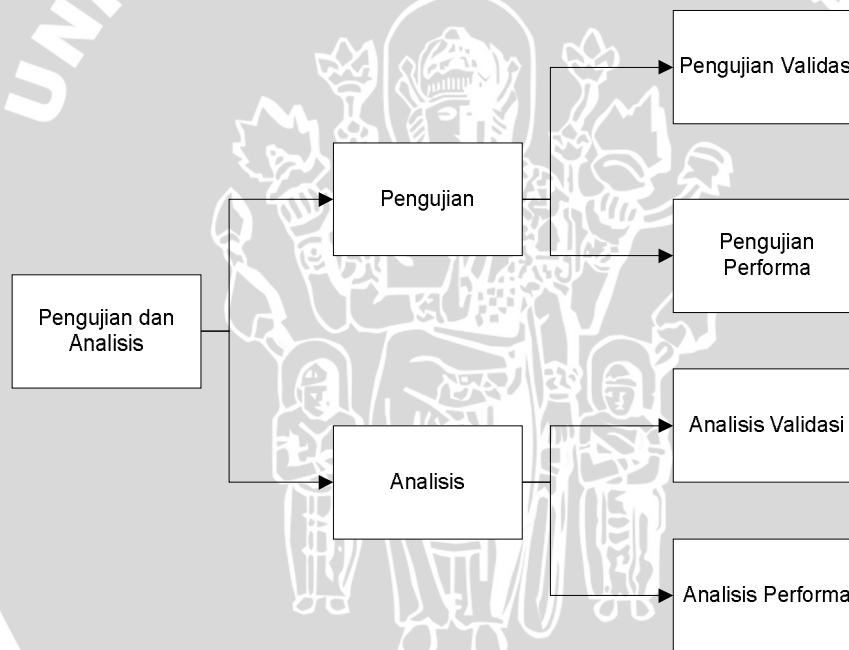
© 2014 Copyright © 2014 PERHUTANI - Kota MALANG
All rights reserved. - Built and Designed by "Saver"

Gambar 5.22 Antarmuka Halaman Lihat Peta
Sumber : [Implementasi]

BAB VI

PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan pengujian dan analisis yang dilakukan pada aplikasi Pemetaan Lahan Hutan berbasis Android yang telah dibangun. Pengujian yang dilakukan yaitu pengujian validasi dan pengujian performa. Pada pengujian validasi akan digunakan teknik pengujian *Black-Box Testing*. Pada pengujian performa akan dilakukan perbandingan antara alat GPS asli dengan A-GPS yang berada dalam Android untuk mengetahui ketepatan titik koordinat yang dihasilkan.



Gambar 6.1 Diagram Alir Pengujian dan Analisis
Sumber : [Pengujian dan Analisis]

6.1 Pengujian

6.1.1 Pengujian Validasi

Pengujian validasi menggunakan metode Black Box dimana pengujian yang dilakukan adalah pengujian fungsionalitas dari sistem, apakah sistem berfungsi dengan hasil yang diinginkan atau tidak. Berikut tabel hasil pengujian validasi sistem.

6.1.1.1 Pengujian Validasi Aplikasi User

Tabel 6.1 Pengujian Validasi Aplikasi User

No	Kasus Uji	Prosedur dan Input	Kondisi yang diharapkan	Hasil
1.	Login user	Login sebagai user	- Ditampilkan halaman login yang berisi kolom input username dan password. - Jika username dan password benar, maka user akan masuk ke halaman <i>home</i>.	Valid
2.	Menambah Informasi Data Lokasi	User memasukkan data yang diperlukan (id lokasi, nama lokasi, jenis tanaman pokok, ketinggian, bonita, bentuk lapangan, kelas hutan, kepadatan bidang, dan waktu tanam tanaman pokok). Lalu menekan tombol kirim di dalam halaman buat lokasi .	- Muncul pemberitahuan bahwa data telah dikirim. - Data Lokasi berhasil ditambahkan di SQL <i>Lite</i>.	Valid
3	Menginput Data Masukan	User memasukkan data yang diperlukan (id lokasi, nama lokasi, jenis tanaman pokok, ketinggian, bonita, bentuk lapangan, kelas hutan, kepadatan bidang, dan waktu tanam tanaman pokok) dengan menggunakan inputan	- Untuk beberapa data seperti ketinggian, bonita, kelas hutan, dan kepadatan bidang hanya dapat di inputkan data berupa angka. - Apabila di inputkan huruf maka system akan mengeluarkan notifikasi berupa peringatan agar	Invalid

		berupa huruf dan angka	menggunakan inputan berupa huruf saja.	
4.	Mengubah Informasi Data Lokasi	User Menekan tombol list data lokasi. Lalumengubah data yang diinginkan (id lokasi, nama lokasi, jenis tanaman pokok, ketinggian, bonita, bentuk lapangan, kelas hutan, kepadatan bidang, dan waktu tanam tanaman pokok). Kemudian menekan tombol kirim di dalam halaman edit lokasi .	• Muncul pemberitahuanbahwa data telah dikirim. • Data Lokasi berhasil diubah di <i>SQL Lite</i>.	Valid
5.	Mengirim Informasi Data Lokasi	User menekan tombol kirim.	• Muncul pemberitahuanbahwa data telah dikirim. • Seluruh informasi Data Lokasi berhasil dikirimkan ke aplikasi <i>server</i>.	Valid
6.	Menambah Data Detail Lokasi	User menekan tombol cari untuk mendapatkan koordinat lokasi tersebut. Lalu <i>user</i> menekan tombol simpan untuk menyimpan data detail lokasi.	• Muncul pemberituahuannilai latitude dan longitude lokasi. • Datadetail lokasi berhasil ditambahkan di <i>SQL Lite</i>	Valid
7.	Menambah Data Detail Tumpangsari	User memasukkan data yang diperlukan (jenis tanaman tumpangsari dan waktu tanam tanaman tumpangsari) di dalam halaman Detail Tumpangsari. Lalu menekan tombol simpan	• Muncul pemberitahuanbahwa data telah disimpan.	Valid

8.	Logout	User menekan tombol logout.	User berhasil keluar dari Aplikasi Pemetaan Lahan Hutan berbasis Android.	Valid
----	--------	-----------------------------	---	-------

Sumber : [Pengujian dan Analisis]

6.1.1.2 Pengujian Validasi Aplikasi Administrator

Tabel 6.2 Pengujian Validasi Aplikasi Administrator

No	Kasus Uji	Prosedur dan Input	Kondisi yang diharapkan	Hasil
1.	Login administrator	Login sebagai administrator	- Ditampilkan form login yang berisi kolom input username dan password. - Jika username dan password benar, maka user akan masuk ke halaman home.	Valid
2.	Pembuatan akun Klien	Administrator memasukkan data yang diperlukan (username, password, dan nama lengkap klien). Lalu menekan tombol "buat akun".	- Muncul notifikasi bahwa data telah dikirim. - Data Klien berhasil ditambahkan di database.	Valid
3.	Melihat Informasi Data Klien	Administrator menekan tombol "Data Klien".	Muncul list data klien yang ditampilkan dalam bentuk tabel.	Valid
4.	Melihat Seluruh Data Lokasi	Administrator menekan tombol "Data Lokasi". Lalu menekan tombol "lihat peta" apabila ingin melihat Lokasi yang diinginkan di peta.	Muncul list data lokasi (anak petak, nama lokasi, jenis tanaman pokok, dan waktu tebang tanaman pokok) yang ditampilkan dalam bentuk tabel pada halaman Data	Valid

			Lokasi. • Muncul letak lokasi ditampilkan pada peta dengan diwarnai menggunakan warna merah pada halaman Peta Lokasi. • Muncul keterangan lokasi (luas, anak petak, nama lokasi, jenis tanaman pokok, jenis tanaman tumpangsari, ketinggian, bonita, bentuk lapangan, kelas hutan, kepadatan bidang, waktu tebang tanaman pokok, dan waktu tebang tanaman tumpangsari) di samping gambar peta lokasi pada halaman Peta Lokasi.	
5.	Pembuatan Luas Wilayah	Sistem menggunakan titik koordinat lokasi sebagai inputan untuk menghitung luas wilayah.	• Luas wilayah ditampilkan di halaman Peta Lokasi.	Valid
6.	Pembuatan Waktu Tebang	Sistem menggunakan jenis tanaman pokok dan waktu tanam tanaman pokok sebagai inputan untuk menghitung waktu tebang tanaman pokok. Sistem menggunakan jenis tanaman tumpangsari dan waktu tanam tanaman tumpangsari	• Waktu tebang berhasil ditambahkan di <i>database</i>. • Waktu tebang tanaman pokok ditampilkan didalam tabel yang berada di halaman Data Lokasi. • Waktu tebang tanaman pokok dan	Valid

		sebagai inputan untuk menghitung waktu tebang tanaman tumpangsari.	tumpangsari ditampilkan di halaman Peta Lokasi.	
7.	Logout	<i>Administrator</i> menekan tombol logout.	<i>Administrator</i> berhasil keluar dari Aplikasi.	Valid

Sumber : [Pengujian dan Analisis]

6.1.2 Pengujian Performa

Pengujian performa dilakukan untuk mengetahui bagaimana performa perangkat lunak Pemetaan Lahan Hutan berbasis Android untuk pendataan dan pemetaan lahan dalam ketepatan pengambilan titik koordinat. Pengujian performa dilakukan dengan membandingkan perangkat GPS dan GPS pada 3 *Smartphone* Android atau *Assisted GPS* yang berbeda.

Pengujian performa dilakukan dengan menggunakan GPS GARMIN 60, Sony Experia Tipo, Sony Experia L, dan Lenovo A3000-H dengan spesifikasi perangkat keras yang ditunjukkan oleh Tabel 6.3, Tabel 6.4, Tabel 6.5, dan Tabel 6.6.

Tabel 6.3 Spesifikasi perangkat keras GPS

GPS GARMIN 60	
Waypoints/icons	500 with name and graphic symbol, 10 nearest (automatic), 10 proximity
Routes	50 reversible routes with up to 250 points each, plus MOB and TracBack® modes
Tracks	10K point automatic track log; 20 saved tracks 500 points each let you retrace your path in both directions
Trip computer	Current speed, average speed, resettable max. speed, trip timer and trip distance
Tables	Built-in celestial tables for best times to fish and hunt,

	<i>sun and moon rise, set and location</i>
Map datums	<i>More than 100 plus user datum</i>
Position format	<i>Lat/Lon, UTM/UPS, Maidenhead, MGRS, Loran TDs and other grids, including user grid</i>
GPS accuracy	Position: <15 meters, 95% typical, Velocity: 0.05 meter/sec steady state
DGPS(WAAS) accuracy	Position: <3 meters, 95% typical, Velocity: 0.05 meter/sec steady state
Interfaces	USB, RS232 with NMEA0183, RTCM104DGPS data format and proprietary Garmin
Antenna	Built-in quadrifilar, with external antenna connection (MCX)
Differential	DGPS, USCG and WAAS capable
Receiver	WAAS enabled, 12 parallel channel GPS receiver continuously tracks and uses up to 12 satellites to compute and update your position

Sumber : [Pengujian dan Analisis]



Gambar 6.2 GPS GARMIN 60

Sumber : [Pengujian dan Analisis]

Tabel 6.4 Spesifikasi perangkat lunak Smartphone Android Sony Xperia Tipo

Sony Xperia Tipo

<i>Operating system</i>	Android OS, v4.0.4 (<i>Ice Cream Sandwich</i>)
<i>Network/Bearer and Wireless Connectivity</i>	GSM 850 / 900 / 1800 / 1900, HSDPA 900 / 2100 - ST21i, HSDPA 850 / 1900 / 2100 - ST21a), Wi-Fi 802.11 b/g/n, Wi-Fi <i>hotspot</i> .
<i>Display</i>	TFT <i>capacitive touchscreen</i> , 256K colors, 320 x 480 pixels, 3.2 inches (~180 ppi pixel density)
<i>Chipset</i>	Qualcomm MSM7225AA, 800 MHz Cortex-A5
<i>Location</i>	Assisted GPS available

Sumber : Pengujian dan Analisis



Gambar 6.3 Sony Xperia Tipo
Sumber : [Pengujian dan Analisis]

Tabel 6.5 Spesifikasi perangkat lunak *Smartphone* Android Sony Xperia L

Sony Xperia L	
<i>Operating system</i>	Android OS, v4.1 (<i>Jelly Bean</i>), <i>upgradable to</i> v4.2.2 (<i>Jelly Bean</i>)
<i>Network/Bearer and Wireless Connectivity</i>	GSM 850 / 900 / 1800 / 1900 - C2104, C2105, HSDPA 900 / 2100 - C2105, HSDPA 850 / 1700 / 1900 / 2100 - C2104, Wi-Fi 802.11 b/g/n, Wi-Fi <i>hotspot</i> .
<i>Display</i>	TFT <i>capacitive touchscreen</i> , 16M colors, 480 x 854 pixels, 4.3 inci (~228 ppi pixel density).
<i>Chipset</i>	Qualcomm MSM8230 Snapdragon, Dual-core 1

	<i>GHz</i>
<i>Location</i>	<i>Assisted GPS available</i>

Sumber : Pengujian dan Analisis



Gambar 6.4 Sony Xperia L
Sumber : [Pengujian dan Analisis]

Tabel 6.6 Spesifikasi perangkat lunak *Smartphone* Android Lenovo A3000-H

Lenovo A3000-H	
<i>Operating system</i>	Android OS, v4.1 (<i>Jelly Bean</i>)
<i>Network/Bearer and Wireless Connectivity</i>	GSM 850 / 900 / 1800 / 1900 (<i>optional</i>), HSDPA (<i>optional</i>), Wi-Fi 802.11 b/g/n, Wi-Fi <i>hotspot</i> .
<i>Display</i>	IPS LCD <i>capacitive touchscreen</i> , 16M colors, 600 x 1024 <i>pixels</i> , 7.0 inci (~170ppi <i>pixel density</i>)
<i>Chipset</i>	MTK 8389/8125 Quad-core 1.2 GHz Cortex-A7
<i>Location</i>	<i>Assisted GPS available</i>

Sumber : [Pengujian dan Analisis]



Gambar 6.5 Lenovo A3000-H
Sumber : [Pengujian dan Analisis]

Perhitungan perbandingan perangkat GPS dan GPS pada *Smartphone* Sony Experia Tipo dalam memperoleh titik koordinat akan dijelas pada tabel 6.7 di bawah ini:

Tabel 6.7 Perbandingan alat GPS dan Assited GPS Sony Experia Tipo

No.	GPS		Sony Experia Tipo		Selisih Jarak (m)
	Latitude	Longitude	Latitude	Longitude	
1.	-7.96839	112.64966	-7.968397	112.649725	7,2
2.	-7.96871	112.65035	-7.968689	112.650357	2,4
3.	-7.96989	112.64988	-7.969861	112.649873	3,3
4.	-7.96961	112.6493	-7.969648	112.649263	5,8
5.	-7.87114	112.52705	-7.871145	112.527002	5,3
6.	-7.87122	112.52640	-7.871227	112.526377	2,6
7.	-7.93945	112.6251	-7.939471	112.625056	5,3
8.	-7.94174	112.63132	-7.941719	112.631305	2,8
9.	-7.969	112.6497	-7.968951	112.649686	5,6
10.	-7.9689	112.64953	-7.968895	112.6495	3,3
Rata – rata = $\frac{\text{Selisih Jarak } (n1 + n2 + n3 + n....)}{\text{Jumlah Total Koordinat}}$					4,36

Sumber : [Pengujian dan Analisis]

Perhitungan perbandingan perangkat GPS dan GPS pada *Smartphone* Sony Experia L dalam memperoleh titik koordinat akan dijelas pada tabel 6.8 di bawah ini:

Tabel 6.8 Perbandingan alat GPS dan Assited GPS Sony Experia L

No.	GPS		Sony Experia L		Selisih Jarak (m)
	Latitude	Longitude	Latitude	Longitude	
1.	-7.96839	112.64966	-7.96845	112.64967	6,7
2.	-7.96871	112.65035	-7.968681	112.650358	3,3
3.	-7.96989	112.64988	-7.969923	112.649855	4,5
4.	-7.96961	112.6493	-7.969647	112.649271	5,2
5.	-7.87114	112.52705	-7.871172	112.527064	3,8
6.	-7.87122	112.52640	-7.871195	112.526388	3
7.	-7.93945	112.6251	-7.939423	112.625115	3,4
8.	-7.94174	112.63132	-7.94174	112.631331	1,2
9.	-7.969	112.6497	-7.968966	112.649697	3,8
10.	-7.9689	112.64953	-7.968893	112.649501	2,5
Rata – rata = $\frac{\text{Selisih Jarak } (n1 + n2 + n3 + n....)}{\text{Jumlah Total Koordinat}}$					3,74

Sumber : [Pengujian dan Analisis]

Perhitungan perbandingan perangkat GPS dan GPS pada *Smartphone* Lenovo A3000-H dalam memperoleh titik koordinat akan dijelas pada tabel 6.8 di bawah ini:

Tabel 6.9 Perbandingan alat GPS dan Assited GPS Lenovo A3000-H

No.	GPS		Lenovo A3000-H		Selisih Jarak (m)
	Latitude	Longitude	Latitude	Longitude	
1.	-7.96839	112.64966	-7.968332	112.649673	6,6
2.	-7.96871	112.65035	-7.968683	112.650402	6,5
3.	-7.96989	112.64988	-7.969887	112.64987	1,1
4.	-7.96961	112.6493	-7.969668	112.649347	8,2
5.	-7.87114	112.52705	-7.871092	112.527008	7
6.	-7.87122	112.52640	-7.871223	112.5264	0,3
7.	-7.93945	112.6251	-7.939437	112.625143	4,9
8.	-7.94174	112.63132	-7.941763	112.631387	7,8
9.	-7.969	112.6497	-7.96896	112.649686	4,7
10.	-7.9689	112.64953	-7.968915	112.649542	2,1
Rata – rata = $\frac{\text{Selisih Jarak } (n1 + n2 + n3 + n....)}{\text{Jumlah Total Koordinat}}$					4,92

Sumber : [Pengujian dan Analisis]

Tabel 6.10 Perbandingan Luas Wilayah menggunakan alat GPS dan Assited GPS

A-GPS	Luas GPS GARMIN 60 (m ²)	Luas A-GPS (m ²)	Selisih Luas (m ²)
Sony Experia Tipo	10879,446	10698,722	180,724
Sony Experia L	10879,446	10921,546	42,1
Lenovo A3000 - H	10879,446	11257,251	377,805
Rata – rata = $\frac{\text{Selisih Jarak } (n1 + n2 + n3 + n....)}{\text{Jumlah Total Koordinat}}$			200,207

Sumber : [Pengujian dan Analisis]

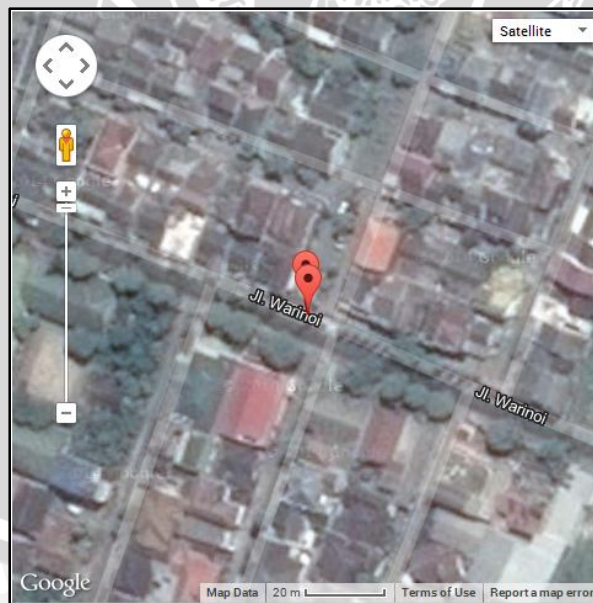
Dari perbandingan di atas, diperoleh rata-rata selisih jarak antara perangkat GPS dengan GPS pada *Smartphone* Sony Experia Tipo dengan hasil akhir 4,36 Meter, rata-rata selisih jarak antara perangkat GPS dengan GPS pada

Smartphone Sony Experia L dengan hasil akhir 3,74 Meter, dan rata-rata selisih jarak antara perangkat GPS dengan GPS pada *Smartphone* Lenovo A3000-H dengan hasil akhir 4,92 Meter sehingga dapat disimpulkan bahwa keakuratan antara perangkat GPS dengan A-GPS mempunyai perbedaan yaitu kurang lebih 3 - 5 Meter disetiap pengambilan titik koordinat.

Sedangkan selisih Luas wilayah antara perangkat GPS dengan GPS pada *Smartphone* Sony Experia Tipo adalah 180,724 meter², selisih perangkat GPS dengan GPS pada *Smartphone* Sony Experia L adalah 42,1 meter², dan selisih perangkat GPS dengan GPS pada *Smartphone* Lenovo A3000-H adalah 377,805 meter². Sehingga dapat disimpulkan bahwa GPS pada *Smartphone* Sony Experia L lebih akurat daripada 2 perangkat lainnya.

Berikut beberapa gambar hasil perbandingan selisih jarak dan luas bidang pada peta antara perangkat GPS dengan A-GPS dalam memperoleh titik koordinat.

6.1.2.1 Gambar Hasil Perbandingan Jarak Titik Koordinat Dalam Peta



Distance: **0.003316 km**

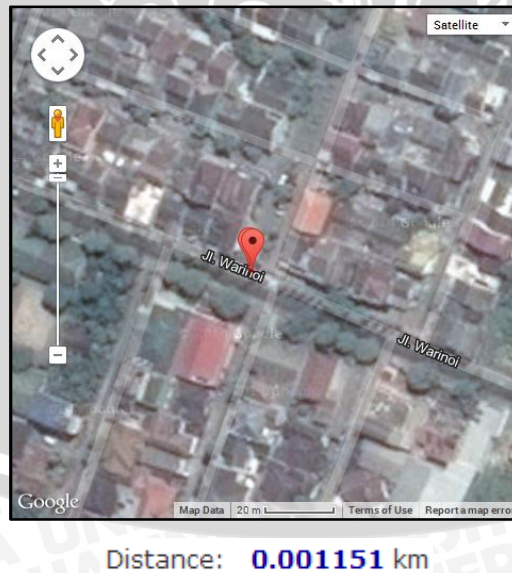
Gambar 6.6 Contoh selisih jarak perangkat GPS dan GPS pada *Smartphone* Sony Experia Tipo

Sumber : [Pengujian dan Analisis]



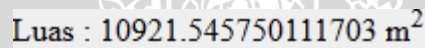
Gambar 6.7Contoh selisih jarak perangkat GPS dan GPS pada *Smartphone* Sony Experia L

Sumber : [Pengujian dan Analisis]



Gambar 6.8Contoh selisih jarak perangkat GPS dan GPS pada *Smartphone* Lenovo A3000-H

6.1.2.2 Gambar Hasil Perbandingan Luas Bidang Dalam Peta



Smartphone Sony Experia L

Luas : 10879.445625255083 m²

Sumber : [Pengujian dan Analisis]

Perangkat GPS menentukan lokasi dari minimal 3 satelit yang membentuk kawasan segitiga dengan mencari longitude, latitude, lokasi dalam peta, dan data lainnya yang diperlukan. Hal ini membutuhkan waktu kira-kira 12 menit untuk menentukan lokasi ketika perangkat khusus GPS dinyalakan. Selain itu perangkat GPS harus berada diluar ruangan, bahkan harus berada di bawah langit terbuka. Kekuatan sinyal dapat berkurang kalau perangkat GPS berada dibawah pohon. Namun setidaknya *user* tidak perlu memiliki akses ke operator ponsel untuk mengoperasikan GPS.

Sedangkan A-GPS dapat menemukan lokasi dengan lebih cepat, bahkan kurang dari 20 detik. Ini dikarenakan A-GPS langsung mencari satelit yang terdekat dengan lokasi ponsel saat itu melalui operator telekomunikasi. Dibutuhkan 3 komponen untuk bisa melakukan hal ini, yaitu satelit, *assistanceserver* (GSM) dan *receiver* A-GPS. Kekurangannya adalah masih tergantung pada *coverage* GSM operator.

6.2 Analisis

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian perangkat lunak Pemetaan Lahan Hutan milik Perum Perhutani berbasis Android yang telah dilakukan. Proses analisis mengacu pada dasar teori sesuai dengan hasil pengujian yang didapatkan. Analisis dilakukan terhadap metode pengembangan perangkat lunak dan hasil pengujian di setiap tahap pengujian. Proses analisis yang dilakukan meliputi analisis metode pengembangan perangkat lunak, analisis hasil pengujian validasi dan analisis hasil pengujian performa.

6.2.1 Analisis Metode Pengembangan Perangkat Lunak

Rekayasa perangkat lunak CBSE (*Component Based Software Engineering*) memiliki keuntungan jelas dalam mengurangi biaya dan resiko.

- Biaya, dari segi waktu pengerjaan penggunaan metode CBSE (*Component Based Software Engineering*) dapat mempercepat pengerjaan proyek. Sehingga dapat meminimalisir biaya yang dikeluarkan.
- Resiko, dengan menggunakan komponen-komponen yang sudah ada dan sudah diuji sebelumnya, maka akan tidak ada atau sedikit resiko yang timbul.

Pada umumnya kendala yang timbul adalah komponen-komponen yang ada tidak benar-benar mampu memenuhi spesifikasi yang diharapkan. Namun pada kasus penelitian ini, semua komponen sudah dapat memenuhi spesifikasi kebutuhan.

6.2.2 Analisis Hasil Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat konformitas antara hasil kinerja sistem dengan daftar kebutuhan. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas perangkat lunak Pemetaan Lahan Hutan milik Perum Perhutani berbasis Android telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan.

6.2.3 Analisis Hasil Pengujian Performa

Proses analisis terhadap hasil pengujian perbandingan ketepatan perangkat GPS dengan A-GPS dilakukan dengan menghitung rata-rata dari selisih jarak pada 3 *Smartphone* Android yang berbeda, diperoleh nilai keakuratan tertinggi terdapat pada *GPS Smartphone* Sony Experia L dengan selisih jarak sebesar 3,74 meter dan selisih luas sebesar 42,1 meter². Namun dalam kondisi pengukuran pada saat cuaca mendung A-GPS bekerja lebih maksimal daripada perangkat GPS, dan A-GPS lebih cepat menemukan lokasi daripada perangkat GPS.

BAB VII

PENUTUP

7.1 Kesimpulan

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

1. Metode pengembangan perangkat lunak yang digunakan dalam perangkat lunak GPS *tracker* berbasis Android menggunakan CBSE (*component-based software engineering*). Dengan model tersebut dapat mempermudah dalam proses pengembangan sistem dari segi biaya dan resiko dengan menggunakan komponen yang telah tersedia.
2. Aplikasi *user* dapat digunakan untuk pendataan dan pemetaan lahan hutan milik Perum Perhutani KPH Malang BKPH Sumber Manjing dan Aplikasi *administrator* perangkat lunak Pemetaan Lahan Hutan berbasis Android dapat digunakan untuk mengelola data lokasi dan data klien yang berasal dari Aplikasi *user*.
3. Aplikasi *user* perangkat lunak Pemetaan Lahan Hutan berbasis Android mengirimkan koordinat ke aplikasi *administrator* yang dikonversikan menjadi sebuah bidang dalam peta dan menampilkan luas (m^2) dari bidang dalam peta tersebut.
4. A-GPS dapat menemukan lokasi dengan lebih cepat, kurang dari 20 detik. Sedangkan GPS membutuhkan waktu kira-kira 12 menit untuk menentukan lokasi ketika perangkat GPS baru dinyalakan. Pada pengujian perbandingan ketepatan antara perangkat GPS dengan A-GPS dilakukan dengan menghitung rata-rata dari selisih jarak pada masing – masing 10 titik koordinat dan luas wilayah yang diperoleh dari tiga A-GPS *Smartphone* Android yang berbeda yaitu kurang lebih 3 - 5 meter untuk selisih jarak dan 200,207 meter² untuk selisih luas wilayah.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan perangkat lunak ini antara lain:

1. Komponen – komponen perangkat lunak yang digunakan dalam *component-based software engineering* sebaiknya dipilih dengan cermat agar tidak terjadi permasalahan dalam proses pengembangan perangkat lunak.
2. Untuk pengembangan lebih lanjut, perangkat lunak ini dapat dikembangkan dan digunakan pada semua kantor KPH Perhutani dan kantor BKPH Perhutani.
3. Untuk pengembangan lebih lanjut, pada sistem informasi *administrator* dapat menjadi salah satu bagian dari *webservice company profile* Perum Perhutani, sehingga dapat dilihat oleh masyarakat umum.



DAFTAR PUSTAKA

- [AMA-09] Amadeo Global Perkasa. 2009. *Mengenal Teknologi Tracking System*.
<http://www.gpstracker.co.id> [30 April 2014].
- [AMI-01] Asosiasi Meubel Indonesia. 2001. Pemasaran Kayu Jati pada Industri Meubel dan Kerajinan. Tidak Diterbitkan.
- [AMR-11] Amri, Syaiful. 2011. *Membangun Sistem Navigasi Di Surabaya Menggunakan Google Maps Api*. Institut Teknologi Sepuluh Nopember Surabaya.
- [CHA-10] Chandra Haekal. 2010. Rimba Kita - Apa itu kayu mahoni.
<http://rimbakita.blogspot.com/2012/11/kayu-mahoni.html> [14 Mei 2014].
- [DEP-00] Departement Kehutanan. 2000. PENJELASAN ATAS RANCANGAN PERATURAN PEMERINTAH REPUBLIK INDONESIA NOMOR ... TAHUN 2000.
<http://www.dephut.go.id/index.php/news/details/3> [14 Mei 2014].
- [DEP-99] Departement Kehutanan. 1999. KEPUTUSAN MENTERI KEHUTANAN DAN PERKEBUNAN NOMOR : 309/Kpts-II/1999.
<http://www.dephut.go.id/index.php/news/details/2981> [14 Mei 2014].
- [DHA-06] Dharwiyanti, Sri. 2006. *Pengantar Unified Modeling Language (UML)*. <http://ilmukomputer.org/2006/08/25/pengantar-uml/> [30 April 2014].
- [DIG-09] Diggelen, Frank van. 2009. *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House.
- [ELR-06] El-Rabbany, Ahmed. 2006. *Introduction to GPS: The Global Positioning System, second edition*. Norwood: Artech House.
- [HBS-92]. Hieronymus Budi Santoso. 1992. Budidaya Sengon. Kanisius. Yogyakarta.

- [HER-11] Hermawan,Stephanus.2011.*Mudahmembuataplikasiandroid*, Yogyakarta : Pemerbit ANDI.
- [JAY-10] Jay A. Kreibich. 2012. *Using SQLite*, United Statesof Amerika: PublisherO'Reilly Media. ISBN: 978-0-596-52118-9.
- [KAD-09] Kadir, Abdul. 2009. *Dasar Pemrograman JavaTM 2*, Yogyakarta : Penerbit ANDI.
- [MOB-12] Detik.com (Jakarta). 2012. 10 Mei.
- [OFF-13] Official U.S. Government information. 2013. *Global Positioning System (GPS) and related topics*. <http://www.gps.gov/systems/gps/> [2 Mei 2014].
- [PER-13] Permana, Budi. 2013. *Pemrograman PHP: Cepat Mahir Bahasa Pemrograman PHP*. <http://ilmukomputer.org/2013/01/22/cepat-mahir-bahasa-pemrograman-php/> [30 April 2014].
- [PRE-01] Pressman, Roger S. 2001. *Software Engineering : A Practitioner's Approach, Fifth Edition*. McGraw Hill.
- [SOM-03] Sommerville, Ian. 2003. *Software Engineering (Rekayasa Perangkat Lunak) / Edisi 6 / Jilid 1*, Jakarta : Penerbit Erlangga.
- [SUG-07] Sugiri., A.Md., S.pd. 2007. *Desain Web Menggunakan HTML dan CSS*. Andi Publisher. Yogyakarta.