

**APLIKASI *LOCATION BASED SERVICE* (LBS) UNTUK BERBAGI
LOKASI MENGGUNAKAN *SHORT MESSAGE SERVICE* (SMS)
BERBASIS MOBILE ANDROID**

(Skripsi)

Oleh :

MUHAQIQIN



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG**

2016

ABSTRACT

APPLICATION LOCATION BASED SERVICE (LBS) FOR LOCATION SHARING USING SHORT MESSAGE SERVICE (SMS) BASED ON MOBILE ANDROID

By

MUHAQIQIN

These days more and more smartphone users. Of the number of smartphone users, the operating system that has many users is the android. Android based smartphone is very easy to use, so many people choose to use android based smartphone. Along with the development of science and technology, emerging technology development is a GPS (Global Positioning System). Generally a GPS system on android smartphone is only used to determine the user's location without being able to share your location with other users using the Short Message Service (SMS), so the user difficult to find the location of other users by using the GPS system on android smartphone. In this study, researchers designing and building applications can store and share your location using Short Message Service (SMS) that allows users to store location records that have been visited and helped find the location of other users. The conclusion of this study is the application *ShareLoc* successfully constructed and useful for every user who has used this *ShareLoc* applications, proved by questionnaire application testing achieves good.

Keyword : *ShareLoc*, Save, Location, GPS, LBS, SMS, Play Store, Android, *Smartphone*.

ABSTRAK

APLIKASI *LOCATION BASED SERVICE* (LBS) UNTUK BERBAGI LOKASI MENGGUNAKAN *SHORT MESSAGE SERVICE* (SMS) BERBASIS MOBILE ANDROID

Oleh

MUHAQIQIN

Dewasa ini pengguna *smartphone* semakin banyak. Dari banyaknya pengguna *smartphone*, sistem operasi yang mempunyai banyak pengguna adalah android. *Smartphone* berbasis android sangat mudah digunakan, sehingga banyak orang memilih untuk menggunakan *smartphone* berbasis android. Seiring dengan perkembangan ilmu dan teknologi, muncul suatu perkembangan teknologi yaitu GPS (*Global Positioning System*). Umumnya sistem GPS pada *smartphone* android hanya digunakan untuk mengetahui lokasi pengguna saja tanpa bisa berbagi lokasi dengan pengguna lainnya menggunakan *Short Message Service* (SMS), sehingga pengguna sulit untuk menemukan lokasi pengguna lain dengan menggunakan sistem GPS pada *smartphone* android. Dalam penelitian ini, peneliti merancang dan membangun aplikasi dapat menyimpan dan berbagi lokasi menggunakan *Short Message Service* (SMS) sehingga memudahkan pengguna dalam menyimpan catatan lokasi yang telah dikunjungi dan membantu menemukan lokasi pengguna lain. Kesimpulan yang didapat dari penelitian ini adalah aplikasi *ShareLoc* berhasil dibangun dan berguna bagi setiap pengguna yang telah menggunakan aplikasi *ShareLoc* ini, dibuktikan dengan kuisisioner pengujian aplikasi yang mendapat nilai baik.

Kata Kunci : *ShareLoc*, Simpan, Lokasi, GPS, LBS, SMS, Play Store, Android, *Smartphone*.

**APLIKASI *LOCATION BASED SERVICE* (LBS) UNTUK BERBAGI
LOKASI MENGGUNAKAN *SHORT MESSAGE SERVICE* (SMS)
BERBASIS MOBILE ANDROID**

Oleh :

MUHAQIQIN

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA KOMPUTER

pada

Jurusan Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG**

2016

Judul Skripsi

**: APLIKASI LOCATION BASED SERVICE
(LBS) UNTUK BERBAGI LOKASI
MENGUNAKAN SHORT MESSAGE
SERVICE (SMS) BERBASIS MOBILE
ANDROID**

Nama Mahasiswa

: Muhaqiqin

Nomor Pokok Mahasiswa

: 1217051044

Jurusan

: Ilmu Komputer

Fakultas

: Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Didik Kurniawan, S.Si., M.T.
NIP 19800419 200501 1 004

Febi Eka Febriansyah, M.T.
NIP 19800219 200604 1 001

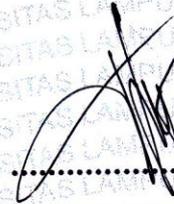
2. Ketua Jurusan Ilmu Komputer

Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP 19640616 198902 1 001

MENGESAHKAN

1. Tim Penguji

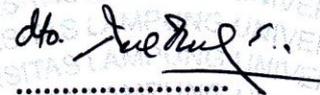
Ketua : Didik Kurniawan, S.Si., M.T.



Sekretaris : Febi Eka Febriansyah, M.T.



**Penguji
Bukan Pembimbing : Dwi Sakethi, S.Si., M.Kom.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Prof. Warsito, S.Si., D.E.A., Ph.D.

NIP 19710212 199512 1 001



Tanggal Lulus Ujian Skripsi : 1 Juli 2016

PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul “Aplikasi *Location Based Service* (LBS) Untuk Berbagi Lokasi Menggunakan *Short Message Service* (SMS) Berbasis Mobile Android” merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, Juli 2016



MUHAQIQIN

NPM. 1217051044

RIWAYAT HIDUP



Penulis dilahirkan pada tanggal 25 Mei 1993 di Natar, sebagai anak ke lima dari 7 bersaudara dengan Ayah bernama Tuanku Imam HWS dan Ibu bernama Budi Yati.

Penulis menyelesaikan pendidikan formal pertama kali pendidikan dasar di SD N 1 Natar dan selesai pada tahun 2006.

Pendidikan menengah pertama di SMP Negeri 1 Natar diselesaikan penulis pada tahun 2009, kemudian melanjutkan ke pendidikan menengah atas di SMK Negeri 2 Bandar Lampung yang diselesaikan penulis pada tahun 2012.

Pada tahun 2012 penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Selama kuliah, penulis mengikuti organisasi Saintek Universitas Lampung periode 2012/2013 hingga periode 2014/2015. Selama menjadi mahasiswa beberapa kegiatan yang dilakukan penulis antara lain:.

1. Pada bulan Januari 2015 penulis melaksanakan kerja praktek di *Software House* Konsep Dot Net.
2. Pada bulan Juli 2015 penulis melaksanakan Kuliah Kerja Nyata (KKN) di Desa Mercu Buana Kecamatan Way Kenanga Kabupaten Tulang Bawang Barat.

PERSEMBAHAN

Dari relung hati yang terdalam
Aku ucapkan syukur atas nikmat-Mu Ya Allah
Karena telah memberiku kekuatan dalam setiap langkahku
Dengan segala kerendahan dan ketulusan hati
Kupersembahkan Skripsi ini kepada yang teramat kucintai.

Ayahanda Tuanku Imam HWS dan Ibunda Budi Yati yang selalu memberikan
do'a, motivasi, dan semangat tiada hentinya.
Terima kasih atas semua perjuangan, pengorbanan, kesabaran, curahan kasih
sayang kepadaku dan tak pernah berhenti membimbing serta mengajarku.

Dan untuk
keluarga besar dan saudara-saudaraku
Yang selama ini selalu memberikan apresiasi, perhatian, bantuan, motivasi dan
kebersamaan yang tak akan pernah terlupakan.

Keluarga Ilmu Komputer 2012,
Serta Almamater tercinta, Universitas Lampung.

MOTO

“Belajar dari hari kemarin, hidup untuk hari ini,

Berharap untuk hari esok.

Yang terpenting tidak berhenti bertanya.”

(Albert Einstein)

“Orang-orang yang sukses telah belajar membuat diri mereka
melakukan hal yang harus dikerjakan ketika hal itu memang harus
dikerjakan, entah mereka menyukainya atau tidak.”

(Aldus Huxley)

Kebanggaan kita yang terbesar adalah bukan tidak pernah gagal,
tetapi bangkit kembali setiap kali kita jatuh.”

(Confucius)

SANWACANA

Assalamualaikum wr. wb.

Alhamdulillah, segala puji bagi Allah SWT yang telah melimpahkan rahmat, hidayah, kesehatan dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan tugas akhir yang berjudul “Aplikasi *Location Based Service* (LBS) Untuk Berbagi Lokasi Menggunakan *Short Message Service* (SMS) Berbasis Mobile Android” dengan baik dan lancar.

Terima kasih penulis ucapkan kepada semua pihak yang telah membantu dan berperan besar dalam menyusun tugas akhir ini, antara lain:

1. Kedua orang tua tercinta, Bapak Tuanku Imam HWS dan Ibu Budi Yati, Kakakku tercinta Halimah Arifni, Nur Arini, Ekys Mardiani, Muhammad Kamil, Adikku tercinta Bagus Muhammad Wahyudi, Ali Anwar Arifin, dan keluarga besar yang selalu memberi do’a, motivasi dan kasih sayang yang tak terhingga.
2. Bapak Didik Kurniawan, S.Si.,MT. sebagai pembimbing utama, yang telah membimbing penulis dan memberikan ide, kritik serta saran sehingga penulisan tugas akhir ini dapat diselesaikan.
3. Bapak Febi Eka Febriansyah, MT. sebagai pembimbing kedua, yang telah membimbing penulis dan memberikan bantuan, kritik serta saran dalam pembuatan tugas akhir ini.
4. Bapak Dwi Sakethi, S.Si., M.Kom. sebagai pembahas, yang telah memberikan masukan yang bermanfaat dalam perbaikan skripsi ini.

5. Bapak Prof. Warsito, S.Si.,D.E.A.,Ph.D. selaku Dekan FMIPA Universitas Lampung.
6. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc., selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
7. Bapak Didik Kurniawan, S.Si.,MT., selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung dan selaku Pembimbing Akademik selama penulis menjadi mahasiswa Ilmu Komputer Universitas Lampung
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu dan pengalaman dalam hidup untuk menjadi lebih baik.
9. Ibu Anita A. Md. Dan Pak Irshan yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer.
10. Sausan Salwa Karimah yang telah memberi do'a dan motivasi untuk menyelesaikan skripsi ini.
11. Keluarga Ilmu Komputer 2012 : Deby Ariyandi, M. Shandy Putra, Afriska Amidya, Beta Yolanda, Haryati, Iluh Astika S, Rani Cahyani, Rizki Aprtiani, Fildan, Didin, Abet, Yudha, Anita, Icha, Ichal, Furqon, Taqiya, Bintang, Owen, Pranata, Dian, Dipa, Moko, Eka, Eko, Erika, Erlina, Cindona, Febi, Hendire, Ichan, Indah, Ivan, Lia, Adit, Abi, Yobi, Juan, Rahman, Nafi, Nikko, Nila, Maya, Puja, Nurul, Concon, Rahmat, Rayvicky, Ciwo, Ridwan, Rio, Riska, Roni, Uchi, Varisa, Yuni.
12. Keluarga KKN Mercu Buana, Keluarga Pak Sarno, Arman, Ari, Gadis, Dita, Emia, Dewi Kalian punya ruang sendiri di hati ini.

13. Mas Nurkholis yang telah membukakan MIPA Terpadu dan ruang baca serta menyiapkan ruang seminar.

14. Almamater Tercinta.

DAFTAR ISI

	Halaman
DAFTAR ISI	ix
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvii
DAFTAR KODE	xviii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
BAB II TINJAUAN PUSTAKA	
2.1 Pengertian Angkutan Umum	6
2.2 Sistem Informasi Geografis	6
2.2.1 Definisi Sistem Informasi Geografis	7
2.2.2 Manfaat Sistem Informasi Geografis	7
2.3 <i>Global Positioning System (GPS)</i>	8
2.4 <i>Short Message Service (SMS)</i>	10

2.5 Cara Kerja SMS	12
2.6 Google Maps API	13
2.6.1 Pengertian API	13
2.6.2 Pengertian Google Maps API	13
2.7 Android	13
2.7.1 Sejarah Android	14
2.7.2 Arsitektur Android	16
2.7.2.1 <i>Application</i> dan <i>Widgets</i>	16
2.7.2.2 <i>Application Frameworks</i>	17
2.7.2.3 <i>Libraries</i>	18
2.7.2.4 <i>Android Run Time</i>	18
2.7.2.5 <i>Linux Kernel</i>	19
2.7.3 Komponen Aplikasi Android	19
2.7.3.1 <i>Activities</i>	20
2.7.3.2 <i>Service</i>	20
2.7.3.3 <i>Broadcast Receiver</i>	21
2.7.3.4 <i>Content Provider</i>	21
2.7.4 Distribusi Android	21
2.8 <i>Android Development Tools</i>	19
2.8.1 <i>Android Software Development Kit (SDK)</i>	22
2.8.2 <i>Android Development Tools (ADT)</i>	22
2.8.3 <i>Eclipse</i>	23
2.9 Metodologi Pengembangan Sistem	24
2.9.1 <i>Extreme Programming (XP)</i>	24

2.9.2	<i>Agile Software Development</i>	26
2.9.3	<i>Unified Modeling Language (UML)</i>	28
2.9.3.1	<i>Use Case Diagram</i>	29
2.9.3.2	<i>Activity Diagram</i>	29
2.9.3.3	<i>Class Responsibility – Collaboration (CRC)</i>	30
2.9.3.4	<i>Class Diagram</i>	31
2.9.3.4	<i>Sequence Diagram</i>	32
2.10	Pengujian Perangkat Lunak	34
2.10.1	<i>Equivalence Partitioning</i>	34

BAB III METODOLOGI PENELITIAN

3.1	Waktu dan Tempat Penelitian.....	36
3.2	Metodologi Penelitian.....	36
3.2.1	Alur Penelitian.....	36
3.2.2	Metode Pengembangan Sistem	40
3.2.3	Metode Pengumpulan Data.....	43
3.3	Analisis dan Perancangan Sistem.....	43
3.3.1	Identifikasi Masalah.....	43
3.3.2	Analisis Sistem.....	45
3.3.3	Analisis <i>User Requirement</i>	45
3.3.4	Perancangan Sistem.....	46
3.3.4.1	Perancangan UML (<i>Unified Modelling Language</i>)....	46
3.3.4.2	Perancangan Antarmuka.....	84
3.4	Metode Pengujian Sistem.....	91

BAB IV HASIL DAN PEMBAHASAN

4.1 Pembahasan.....	96
4.2 Implementasi Kode Program.....	96
4.2.1 Kode Program <i>Splashscreen</i>	97
4.2.2 Kode Program <i>Request Location</i>	97
4.2.3 Kode Program <i>My Location</i>	99
4.2.4 Kode Program <i>History</i>	102
4.2.5 Kode Program <i>MyMaps</i>	104
4.2.6 Kode Program <i>Help</i>	105
4.2.7 Kode Program <i>About</i>	107
4.3 Hasil.....	108
4.4 Implementasi Sistem.....	108
4.5 <i>Layout Aplikasi ShareLoc</i>	110
4.5.1 <i>Layout Halaman Splash Screen</i>	110
4.5.2 <i>Layout Halaman My Location</i>	111
4.5.3 <i>Layout Halaman Request Location</i>	113
4.5.4 <i>Layout Menu History</i>	115
4.5.5 <i>Layout Menu Help</i>	120
4.5.6 <i>Layout Menu About</i>	122
4.6 Hasil Pengujian.....	123
4.6.1 Pengujian Fungsional.....	123
4.6.1.1 Pengujian Versi Android.....	124
4.6.1.2 Pengujian Resolusi Layar dan Densitas Layar.....	125
4.6.1.3 Pengujian <i>User Interface</i>	126

4.6.1.4 Pengujian Fungsi dari Menu Aplikasi.....	128
4.6.1.5 Pengujian Koneksi Internet.....	130
4.6.2 Pengujian Non Fungsional.....	130
BAB V SIMPULAN DAN SARAN	
5.1 Simpulan.....	138
5.2 Saran.....	138
DAFTAR PUSTAKA.....	136
LAMPIRAN	

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Komponen Dasar LBS (Steigner, 2006).....	5
Gambar 2.2 Satelit yang Berorbit.....	9
Gambar 2.3 GPS Navigasi.....	10
Gambar 2.4 GPS Geodetik.....	10
Gambar 2.5 Cara Kerja SMS.....	12
Gambar 2.6 Arsitektur Android.....	16
Gambar 2.7 Siklus Extreme Programming (Pressman, 2010).....	25
Gambar 2.8 <i>Class Responsibility – Collaboration (CRC)</i>	31
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	37
Gambar 3.2 <i>Use CaseDiagram</i>	47
Gambar 3.3. <i>Activity Diagram</i> Posisi Sekarang.....	48
Gambar 3.4. <i>Activity Diagram</i> Menyimpan Lokasi.....	49
Gambar 3.5. <i>Activity Diagram</i> Mengirim Lokasi.....	50
Gambar 3.6. <i>Activity Diagram</i> Memperbarui Lokasi.....	51
Gambar 3.7. <i>Activity Diagram</i> Mengirim SMS.....	52
Gambar 3.8. <i>Activity Diagram</i> Menampilkan Peta Lokasi.....	53
Gambar 3.9. <i>Activity Diagram</i> Menampilkan Rute.....	54
Gambar 3.10. <i>Activity Diagram</i> Menampilkan Navigasi.....	55

Gambar 3.11. <i>Activity Diagram</i> Akses Lokasi Tersimpan	56
Gambar 3.12. <i>Activity Diagram</i> Menghapus Lokasi Tersimpan	57
Gambar 3.13. <i>Activity Diagram</i> Mengirim Lokasi Tersimpan	58
Gambar 3.14. <i>Activity Diagram</i> Menampilkan Peta Lokasi Tersimpan	59
Gambar 3.15. <i>Activity Diagram</i> Mengaktifkan/Menonaktifkan Aplikasi	60
Gambar 3.16. <i>Activity Diagram</i> Memproses SMS	61
Gambar 3.17. <i>Activity Diagram</i> Mengimport Data Lokasi Tersimpan	62
Gambar 3.18 <i>Activity Diagram</i> Mengexport Data Lokasi Tersimpan	63
Gambar 3.19. <i>Activity Diagram</i> Bantuan	64
Gambar 3.20. <i>Activity Diagram</i> Tentang Aplikasi	65
Gambar 3.21. <i>Activity Diagram</i> Tentang Aplikasi	66
Gambar 3.22. <i>Sequence Diagram</i> Menampilkan Posisi Sekarang	67
Gambar 3.23. <i>Sequence Diagram</i> Menyimpan Lokasi	68
Gambar 3.24. <i>Sequence Diagram</i> Mengirim Lokasi	69
Gambar 3.25. <i>Sequence Diagram</i> Perbarui Lokasi	70
Gambar 3.26. <i>Sequence Diagram</i> Mengirim SMS	71
Gambar 3.27. <i>Sequence Diagram</i> Peta Lokasi	72
Gambar 3.28. <i>Sequence Diagram</i> Menampilkan Rute	73
Gambar 3.29. <i>Sequence Diagram</i> Menampilkan Navigasi	74
Gambar 3.30. <i>Sequence Diagram</i> Akses Lokasi Tersimpan	75
Gambar 3.31. <i>Sequence Diagram</i> Menghapus Lokasi Tersimpan	76
Gambar 3.32. <i>Sequence Diagram</i> Mengirim Lokasi Tersimpan	77

Gambar 3.33. <i>Sequence Diagram</i> Menampilkan Peta Lokasi Tersimpan	78
Gambar 3.34. <i>Sequence Diagram</i> Mengaktifkan/Menonaktifkan Aplikasi	79
Gambar 3.35. <i>Sequence Diagram</i> Memproses SMS	80
Gambar 3.36 <i>Sequence Diagram</i> Mengimport Data Lokasi Tersimpan	81
Gambar 3.37 <i>Sequence Diagram</i> Mengexport Data Lokasi Tersimpan	82
Gambar 3.38. <i>Sequence Diagram</i> Bantuan	83
Gambar 3.39. <i>Sequence Diagram</i> Tentang Aplikasi	84
Gambar 3.40. <i>Design Layout Splash Screen</i>	85
Gambar 3.41. <i>Design Layout Menu Request Location</i>	86
Gambar 3.42. <i>Design Layout Input Nomor Telepon</i>	86
Gambar 3.43 <i>Design Layout Menu My Location</i>	87
Gambar 3.44. <i>Design Layout Layout Menu History</i>	88
Gambar 3.45. <i>Design Layout Layout Menu Maps</i>	89
Gambar 3.46. <i>Design Layout Layout Menu Help</i>	90
Gambar 3.47. <i>Design Layout Menu About</i>	91
Gambar 4.1 <i>Layout Splash Screen</i>	110
Gambar 4.2 <i>Layout Menu My Location</i>	111
Gambar 4.3 <i>Layout Menu Request Location</i>	114
Gambar 4.4 <i>Layout Dialog Send SMS</i>	115
Gambar 4.5 <i>Layout Menu History</i>	116
Gambar 4.6 <i>Layout Edit Location Name</i>	117
Gambar 4.7 <i>Layout Menu More</i>	118

Gambar 4.8 <i>Layout Maps</i>	119
Gambar 4.9 <i>Layout Menu Search</i>	120
Gambar 4.10 <i>Layout Submenu Menu Help</i>	121
Gambar 4.11 <i>Layout Menu Help</i>	122
Gambar 4.12 <i>Layout Menu About</i>	123
Gambar 4.13 Grafik Presentasi Rata-rata Jawaban Responden per Kategori Penilaian pada Variabel <i>User Friendly</i>	134
Gambar 4.14 Grafik Presentasi Rata-rata Jawaban Responden per Kategori Penilaian pada Variabel Interaktif	135
Gambar 4.15 Aplikasi <i>ShareLoc</i> Pada Play Store	136
Gambar 4.16 Komentar Aplikasi <i>ShareLoc</i> Pada Play Store	137

DAFTAR TABEL

	Halaman
Tabel 2.1. Simbol dan Keterangan <i>Use Case</i>	29
Tabel 2.2. Simbol dan Keterangan <i>Activity Diagram</i>	30
Tabel 2.3. Simbol dan keterangan <i>Class Diagram</i>	31
Tabel 2.4. Simbol dan keterangan <i>Sequence Diagram</i>	32
Tabel 3.1 Daftar Pengujian <i>Equivalence Partitioning (EP)</i>	92
Tabel 4.1 Daftar <i>Class</i> Pada Aplikasi <i>ShareLoc</i>	109
Tabel 4.2 Daftar <i>Layout</i> pada Aplikasi <i>ShareLoc</i>	109
Tabel 4.3 Tabel Pengujian Versi Android.....	124
Tabel 4.4 Pengujian Resolusi Layar dan Densitas Layar.....	125
Tabel 4.5 Pengujian <i>User Interface</i>	126
Tabel 4.6 Pengujian Fungsi dari Menu Aplikasi.....	127
Tabel 4.7 Pengujian Koneksi Internet dan GPS.....	130
Tabel 4.8 Interval dan Kategori Penilaian.....	132
Tabel 4.9 Hasil Penilaian Variabel <i>User Friendly</i>	132
Tabel 4.10 Hasil Penilaian Variabel Interaktif.....	134

DAFTAR KODE

	Halaman
Kode 4.1 Potongan Kode Program <i>Splash Screen</i>	97
Kode 4.2 Potongan Kode Program <i>Request Location</i>	98
Kode 4.3 Potongan Kode Program XML Menu <i>Request Location</i>	99
Kode 4.4 Potongan Kode Program Menu <i>My Location</i>	100
Kode 4.5 Potongan Kode Program XML Menu <i>My Location</i>	101
Kode 4.6 Potongan Kode Program Menu <i>History</i>	102
Kode 4.7 Potongan Kode Program XML Menu <i>History</i>	103
Kode 4.8 Potongan Kode Program <i>MyMaps</i>	104
Kode 4.9 Potongan Kode Program XML <i>MyMaps</i>	105
Kode 4.10 Potongan Kode Program <i>Help</i>	106
Kode 4.11 Potongan Kode Program XML <i>Help</i>	106
Kode 4.12 Potongan Kode Program <i>About</i>	107
Kode 4.13 Potongan Kode Program XML <i>About</i>	107

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini pengguna *smartphone* semakin banyak. Dari banyaknya pengguna *smartphone*, sistem operasi yang mempunyai banyak pengguna adalah Android. Menurut laporan dari perusahaan riset teknologi dan pasar, *International Data Corporation* (IDC), mengenai pangsa pasar sistem operasi *smartphone* di kuartal kedua 2015 dan tahun 2015 secara keseluruhan, Android masih memimpin pangsa pasar OS dengan total 329 juta unit (96,8 persen) *smartphone* Android dikirimkan pada kuartal kedua 2015, sedangkan secara keseluruhan pada tahun 2015, sekitar 271 juta unit *smartphone* yang dikirimkan merupakan *smartphone* Android dengan pangsa pasar 82,2 persen.

Pada masa ini, *smartphone* berbasis Android telah banyak dikembangkan oleh berbagai perusahaan telekomunikasi, sehingga perkembangannya cepat dan teknologi-teknologi yang terdapat pada *smartphone* Android menjadi semakin canggih. Salah satu teknologi yang terdapat pada *smartphone* Android adalah *Global Positioning System* (GPS).

Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang dikembangkan oleh Departemen Pertahanan Amerika pada tahun 1970. Awalnya,

GPS dikembangkan sebagai sistem militer untuk kebutuhan militer Amerika. Setelah itu, barulah GPS digunakan oleh masyarakat umum, dan sekarang GPS dapat diakses oleh militer dan masyarakat umum. GPS memberikan posisi dan informasi waktu terus menerus, dimanapun dan pada kondisi cuaca seperti apapun. Karena GPS melayani pengguna dengan jumlah yang tak terhingga, dengan alasan keamanan, GPS dibuat menjadi sistem satu arah (pasif). Karena itu, pengguna hanya dapat menerima sinyal satelit tanpa bisa mengirim sinyal ke satelit (El-Rabbany, 2002). Dengan adanya sistem GPS pada *smartphone* Android, pengguna dapat mengetahui posisi atau lokasi dimana pengguna tersebut berada, sehingga pengguna tidak mudah tersesat.

Umumnya sistem GPS pada *smartphone* Android hanya digunakan untuk mengetahui lokasi pengguna saja tanpa bisa berbagi lokasi dengan pengguna lainnya menggunakan *Short Message Service* (SMS), sehingga pengguna sulit untuk menemukan lokasi pengguna lain dengan menggunakan sistem GPS pada *smartphone* Android. *Smartphone* Android yang telah memiliki sistem GPS pun umumnya tidak dapat menyimpan informasi lokasi yang telah dikirimkan oleh GPS, sehingga pengguna tidak dapat melihat catatan tempat-tempat yang telah dikunjungi pada *smartphone* mereka.

Hal tersebut yang menjadi latar belakang dalam melakukan penelitian yang disajikan dalam skripsi yang berjudul “Aplikasi *Location Based Service* (LBS) Untuk Berbagi Lokasi Menggunakan *Short Message Service* (SMS) Berbasis Mobile Android”, dengan harapan membantu setiap pengguna *smartphone*

Android untuk dapat menyimpan dan berbagi lokasi menggunakan *Short Message Service* (SMS) sehingga memudahkan pengguna dalam menyimpan catatan lokasi yang telah dikunjungi dan membantu menemukan lokasi pengguna lain.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah “Bagaimana merancang dan membangun aplikasi yang dapat menyimpan dan berbagi lokasi menggunakan *Short Message Service* (SMS) sebagai bantuan untuk menyimpan catatan lokasi yang telah dikunjungi dan membantu menemukan lokasi pengguna lain berbasis Android”.

1.3 Batasan Masalah

Aplikasi *ShareLoc* untuk berbagi lokasi menggunakan *Short Message Service* (SMS) berbasis mobile Android ini memiliki batasan sebagai berikut.

1. Aplikasi ini dioperasikan hanya dalam *smartphone* yang mendukung OS (*Operating System*) Android.
2. Aplikasi ini hanya dapat dijalankan jika fitur GPS pada *device* aktif.
3. Aplikasi ini mengambil koordinat lokasi via *GPS Provider* atau *Network Provider* pada *smartphone* berbasis Android.
4. Aplikasi ini dapat mengirim dan membalas SMS dengan format tertentu secara otomatis.
5. Aplikasi ini dapat membagikan Lokasi/Koordinat *device* menggunakan SMS.

6. Aplikasi mempunyai fitur ON/OFF untuk mengaktifkan atau mematikan aplikasi.
7. Aplikasi dapat menampilkan lokasi kedalam Map (Google Maps).
8. Aplikasi ini dapat diakses secara *offline* kecuali untuk fitur Map yang hanya dapat diakses secara *online*.
9. Aplikasi ini dibangun dan dirancang menggunakan bahasa pemrograman Java.

1.4 Tujuan

Penelitian ini bertujuan untuk merancang dan membangun aplikasi yang dapat menyimpan dan berbagi lokasi menggunakan *Short Message Service* (SMS) sebagai bantuan untuk menyimpan catatan lokasi yang telah dikunjungi dan membantu menemukan lokasi pengguna lain berbasis Android.

1.5 Manfaat

Manfaat yang diperoleh dari penelitian ini diharapkan dapat membantu pengguna *smartphone* Android untuk dapat menyimpan dan berbagi lokasi menggunakan *Short Message Service* (SMS) sehingga memudahkan pengguna dalam menyimpan catatan lokasi yang telah dikunjungi dan membantu menemukan lokasi pengguna lain.

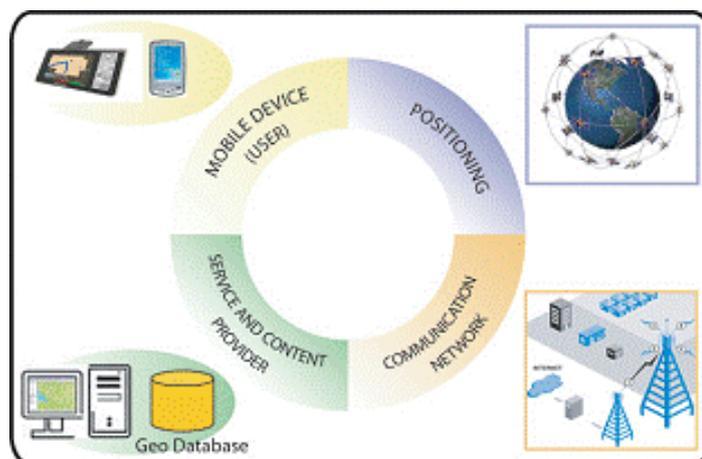
BAB II

TINJAUAN PUSTAKA

2.1 *Location Based Services (LBS)*

Location Based Service (LBS) atau Layanan Berbasis Lokasi merupakan layanan informasi yang memanfaatkan kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak dan dapat diakses dengan perangkat bergerak melalui jaringan telekomunikasi bergerak (Steiniger, 2006).

Dalam Layanan Berbasis Lokasi terdapat lima komponen penting seperti terlihat pada Gambar 2.1.



Gambar 2.1 Komponen Dasar LBS (Steigner, 2006).

Setiap komponen mempunyai fungsi (Steigner, 2006).

1. *Mobile Devices*, merupakan suatu alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Informasi dapat diberikan dalam bentuk suara, gambar, dan text.
2. *Communication Network*, komponen ini mengirim data pengguna dan informasi yang diminta dari *Mobile* terminal ke *Service Provider* kemudian mengirimkan kembali informasi yang diminta ke pengguna. *Communication network* dapat berupajaringan seluler (GSM, CDMA), *Wireless Local Area Network* (WLAN), atau *Wireless Wide Area Network* (WWAN).
3. *Positioning Component*, digunakan untuk memproses suatu layanan maka posisi pengguna harus diketahui.
4. *Service and Application Provider*, penyedia layanan menawarkan berbagai macam layanan kepada pengguna dan bertanggungjawab untuk memproses informasi yang diminta oleh pengguna.
5. *Data and Content Provider*, penyedia layanan tidak selalu menyimpan semua data yang dibutuhkan yang bisa diakses oleh pengguna. Untuk itu, data dapat diminta dari *data* dan *content provider*.

2.2 Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) atau juga dikenal sebagai *Geographic Information System* (GIS) pertama pada tahun 1960 yang bertujuan untuk menyelesaikan permasalahan geografis. Empat puluh tahun kemudian GIS berkembang tidak hanya bertujuan untuk menyelesaikan permasalahan geografi

saja, tetapi sudah merambah ke berbagai bidang, seperti analisis penyakit epidemik (demam berdarah) dan analisis kejahatan (kerusuhan), termasuk analisis kepariwisataan. Kemampuan dasar dari SIG adalah mengintegrasikan berbagai operasi basis data seperti *query*, menganalisisnya serta menampilkannya dalam bentuk pemetaan berdasarkan letak geografisnya. Inilah yang membedakan SIG dengan sistem informasi lain (Prahasta, 2014).

2.2.1 Definisi Sistem Informasi Geografis

Istilah *geography* digunakan karena SIG dibangun berdasarkan pada geografi atau spasial. Geografi atau spasial ini mengarah pada spesifikasi lokasi dalam suatu *space*. SIG merupakan sistem komputer yang berbasis pada sistem informasi yang digunakan untuk memberikan bentuk digital dan analisis terhadap permukaan geografi bumi. Geografi adalah informasi mengenai permukaan bumi dan semua obyek yang berada di atasnya, sedangkan sistem informasi geografis (SIG) adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Sistem informasi geografis merupakan bentuk sistem informasi yang menyajikan informasi dalam bentuk grafis dengan menggunakan peta sebagai antarmuka. SIG tersusun atas konsep beberapa lapisan (*layer*) dan relasi (Prahasta, 2014).

2.2.2 Manfaat Sistem Informasi Geografis

Pada dasarnya fungsi dari Sistem Informasi Geografis ialah meningkatkan kemampuan menganalisis informasi spasial secara terpadu untuk perencanaan dan pengambilan keputusan. SIG dapat memberikan informasi kepada pengambil

keputusan untuk analisis dan penerapan *database* keruangan (Prahasta, 2014). SIG mampu memberikan kemudahan-kemudahan yang diinginkan. Dengan SIG kita akan dimudahkan dalam melihat fenomena kebumihan dengan perspektif yang lebih baik. SIG mampu mengakomodasi penyimpanan, pemrosesan, dan penayangan data spasial digital bahkan integrasi data yang beragam, mulai dari citra satelit, foto udara, peta bahkan data statistik. SIG juga mengakomodasi dinamika data, pemutakhiran data yang akan menjadi lebih mudah (Swastikayana, 2011).

2.3 *Global Positioning System (GPS)*

Global Positioning System adalah sistem untuk menentukan posisi dan navigasi secara global dengan menggunakan satelit. Sistem yang pertama kali dikembangkan oleh Departemen Pertahanan Amerika ini digunakan untuk kepentingan militer maupun sipil (survei pemetaan dan informasi geografi).

Sistem GPS, yang nama aslinya adalah NAVSTAR GPS (*Navigation Satellite Timing and Ranging Global Positioning System*), mempunyai tiga segmen yaitu : satelit, pengontrol, dan penerima/pengguna. Satelit GPS yang mengorbit bumi, dengan orbit dan kedudukan yang tetap (koordinat pasti), seluruhnya berjumlah 24 buah dimana 21 buah aktif bekerja dan 3 buah sisanya adalah cadangan.

- Satelit bertugas untuk menerima dan menyimpan data yang ditransmisikan oleh stasiun-stasiun pengontrol, menyimpan dan menjaga informasi waktu berketelitian tinggi (ditentukan dengan jam atomic di satelit), dan

memancarkan sinyal dan informasi secara kontinu ke pesawat penerima (*receiver*) dari pengguna.

- Pengontrol bertugas untuk mengendalikan dan mengontrol satelit dari bumi baik untuk mengecek kesehatan satelit, penentuan dan prediksi orbit dan waktu, sinkronisasi waktu antar satelit, dan mengirim data ke satelit.
- Penerima bertugas menerima data dari satelit dan memprosesnya untuk menentukan posisi (posisi tiga dimensi yaitu koordinat di bumi plus ketinggian), arah, jarak, dan waktu yang diperlukan oleh pengguna. Ada dua macam tipe penerima yaitu tipe NAVIGASI dan tipe GEODETIC .yang termasuk tipe NAVIGASI antara lain : *Trimble Ensign, Trimble Pathfinder, Garmin, Sony* dan lain sebagainya. Sedangkan tipe GEODETIC antara lain : *Topcon, Leica, Astech, Trimble seri 4000* dan lain-lain.



Gambar 2.2 Satelit yang Berorbit

- GPS Navigasi adalah GPS *handheld* yang mempunyai ketelitian 3-10 meter. Biasanya bisa digenggam. Selain berfungsi sebagai perangkat navigasi juga bisa digunakan untuk pemetaan.



Gambar 2.3 GPS Navigasi

- GPS Geodetik adalah GPS yang mempunyai ketelitian tinggi sampe kelas milimeter. Alat ini terdiri dari base dan rover.
- Tipe Geodetik dual frekuensi dapat memberikan ketelitian posisi hingga mencapai millimeter. Tipe ini biasa digunakan untuk aplikasi *precise positioning* seperti pembangunan jaring titik kontrol, survey deformasi, dan geodinamika. Harga *receiver* tipe ini cukup mahal, mencapai ratusan juta rupiah untuk 1 unitnya (Maulana, 2014).



Gambar 2.4 GPS Geodetik

2.4 *Short Message Service (SMS)*

Short Message Service (SMS) adalah salah satu fasilitas dari teknologi GSM yang memungkinkan mengirim dan menerima pesan-pesan singkat berupa *text* dari *Mobile Station (MS)*. Layanan SMS juga memungkinkan pengiriman pesan dalam

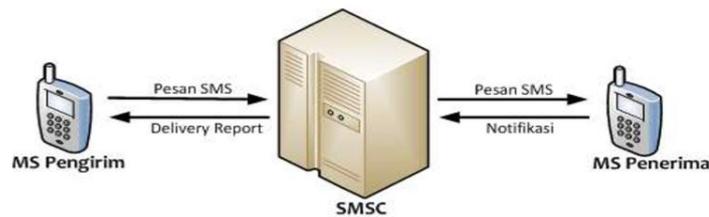
bentuk alphanumeric, layanan SMS ini banyak diaplikasikan pada sistem komunikasi tanpa kabel (*wireless*). Teknologi *wireless* dipelopori dari kawasan Eropa yang diawali pada kebutuhan bersama terhadap satu sistem jaringan baru yang dapat menjadi *standard* jaringan yang berlaku dan dapat diterapkan di seluruh kawasan Eropa. Dalam sistem baru juga harus terdapat kemampuan yang dapat mengantisipasi mobilitas pengguna serta kemampuan melayani lebih banyak pengguna untuk menampung penambahan jumlah pelanggan baru.

Karena hal ini tidak dapat dilakukan dengan mempertahankan sistem analog, maka kemudian diputuskan untuk merombak sistem dan menggantinya dengan sistem digital. *Standard* baru diperkenalkan dengan nama *Global Standard for Mobile Communications* (GSM). GSM pada awalnya adalah kepanjangan dari *Groupe Speciale Mobile*, sebuah badan gabungan dari para ahli yang melakukan studi bersama untuk menciptakan *standard* GSM tersebut.

Pada bulan Desember 1992, dilakukan pengiriman pesan menggunakan SMS dari sebuah *Personal Computer* (PC) ke telepon *mobile* (bergerak) dalam jaringan GSM milik Vodafone Inggris, kemudian merambah ke benua Amerika yang dipelopori oleh beberapa operator komunikasi bergerak berbasis digital seperti BellSouth Mobility, PrimeCo dan operator lainnya. Teknologi yang digunakan dari pengiriman SMS yaitu *Store and forward service*, jadi SMS yang dikirim akan disimpan sementara di *server SMS center* kemudian dialihkan ke nomor tujuan.

2.5 Cara Kerja SMS

Mekanisme dari sistem SMS ialah *store and forward*, dimana sistem dapat melakukan pengiriman *short message* dari satu terminal ke terminal lainnya. Hal ini dapat dilakukan berkat adanya sebuah entitas dalam sistem SMS yang bernama *Short Message Service Center* (SMSC), disebut juga *Message Center* (MC). SMSC merupakan sebuah perangkat yang melakukan tugas *store and forward traffic short message* seperti pada Gambar 2.5 tersebut.



Gambar 2.5 Cara Kerja SMS

Pengiriman pesan SMS secara *store and forward* yaitu pengirim SMS memasukkan pesan SMS dan nomor tujuan dan kemudian mengirimkannya (*store*) ke *server* SMS (*SMS Center*) yang kemudian bertanggung jawab mengirim pesan SMS tersebut (*forward*) ke nomor telepon tujuan. Hal ini berarti bahwa pengirim dan penerima SMS tidak berada dalam status berhubungan (*connected*) satu sama lain, ketika akan saling bertukar pesan SMS. Pesan yang dikirim oleh pengirim ke SMSC yang kemudian menunggu untuk dapat meneruskan pesan tersebut ke penerima. Ketika status penerima dalam keadaan aktif pesan segera dikirim oleh SMSC ke nomor tujuan beserta isi pesan pengirim. Pengirim akan menerima *delivery report* bahwa pesan telah terkirim (*message sent*).

2.6 Google Maps API

2.6.1 Pengertian API

Application Programming Interface (API) bukan hanya satu *set class* dan *method* atau fungsi dan *signature* yang sederhana. API yang bertujuan utama untuk mengatasi ketidakpahaman dalam membangun *software* yang berukuran besar, berawal dari sesuatu yang sederhana sampai ke yang kompleks dan merupakan perilaku komponen yang sulit dipahami (Halim, 2011).

2.6.2 Pengertian Google Maps API

Seperti yang tercatat oleh Svennerberg, Google Maps API adalah API yang paling populer di internet. Pencatatan yang dilakukan pada bulan Mei tahun 2010 menyatakan bahwa 43% *mashup* (aplikasi dan situs web yang menggabungkan dua atau lebih sumber data) menggunakan Google Maps API. Beberapa tujuan dari penggunaan Google Maps API adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya (Halim, 2011).

2.7 Android

Pada awal peluncurannya, Google meyakini bahwa *platform* perangkat *mobile* Android memiliki kesempatan yang sangat besar dalam pengembangan aplikasi. Google mengumumkan *Open Handset Alliance* (OHA) dan *platform* Android pada November 2007, dan meluncurkan *Android Software Development Kit* (SDK) pertama yang masih dalam versi *beta* di waktu yang sama. Dalam waktu yang tidak lama, lebih dari satu juta orang mengunduh Android SDK dari *website*

Google. Di Amerika Serikat, *T-Mobile* mengumumkan perangkat *mobile* Android bergelar G1 pada Oktober 2008, dan diperkirakan ratusan ribu perangkat G1 terjual pada akhir tahun yang sama. Android memiliki potensi yang besar untuk menghilangkan batasan dan kendala yang selama ini muncul dalam mengembangkan suatu perangkat lunak versi *mobile phone*. Dari berbagai hal yang telah disiapkan *Google*, Android berhasil mengembangkan pasar aplikasi perangkat *mobile*, serta memberikan kesempatan besar untuk para pengembang perangkat lunak untuk mengambil keuntungan dari aplikasi yang dibuatnya (Andry, 2011).

2.7.1 Sejarah Android

Android adalah sebuah sistem operasi yang sengaja diciptakan untuk perangkat *mobile* berbasis *Linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya *Google Inc.* membeli *Android Inc.* yang merupakan pendatang baru yang membuat piranti lunak untuk perangkat ponsel pintar/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance* (OHA), sebuah konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*.

Pada saat perilis perdana Android, 5 November 2007, Android bersama OHA menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, *Google* merilis kode-kode Android di bawah lisensi *Apache*, sebuah

lisensi perangkat lunak dan *open platform* perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS), dan yang kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal dengan *Open Handset Distribution* (OHD).

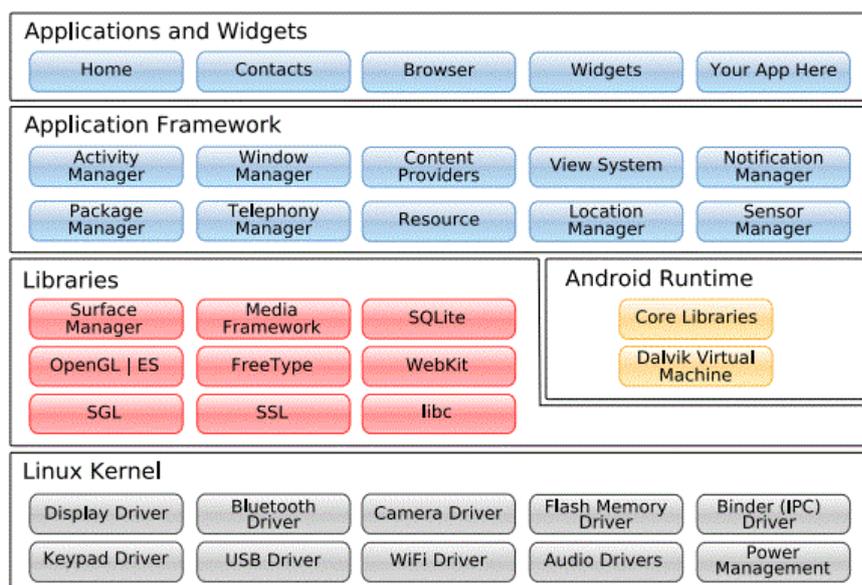
Pada Juli 2005, Google bekerjasama dengan *Android Inc.*, perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri *Android Inc.* bekerjasama pada Google, diantaranya Andy Rubin, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Pada internal perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh *kernel Linux*. Hal ini menunjukkan bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

Sekitar September 2007, sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler. Akhirnya Google mengenalkan *Nexus One*, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh *HTC Corporation* dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android ARM Holding*, *Atheros Communications*, diproduksi oleh *Asustek Computer Inc*, *Garmin Ltd*, *Softbank*, *Sony Ericsson*, *Toshiba Corp.*, dan *Vodafone Group Plc*. Seiring pembentukan OHA (*Open Handset Alliance*), OHA mengumumkan produk perdananya, Android perangkat *mobile* yang merupakan modifikasi dari *kernel*

Linux 2.6. Sejak Android dirilis, telah dilakukan pembaruan berupa pembaruan *bug* dan penambahan fitur baru. Pada penghujung tahun 2009, diperkirakan terdapat 18 jenis telepon seluler yang menggunakan sistem operasi Android (Safaat, 2012).

2.7.2 Arsitektur Android

Secara garis besar, Arsitektur Android dapat dijelaskan dan ditunjukkan seperti pada Gambar 2.6.



Gambar 2.6. Arsitektur Android

2.7.2.1 Application dan Widgets

Applications and Widgets ini adalah *layer* yang membatasi penggunaannya hanya untuk berhubungan dengan aplikasi saja, seperti halnya saat pengguna mengunduh aplikasi kemudian melakukan instalasi dan menjalankan aplikasi tersebut. Pada *layer* terdapat aplikasi ini termasuk klien *email*, program SMS,

kalender, peta, *browser* atau kontak. Semua aplikasi ditulis dengan menggunakan bahasa pemrograman *Java*.

2.7.2.2 Application Frameworks

Android adalah “*Open Development Platform*”, yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang baik dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notifications*, serta lain sebagainya. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang berkategori inti. Arsitektur aplikasi dirancang supaya pengguna dapat dengan mudah menggunakan kembali komponen yang sudah digunakan (*reuse*).

Sehingga dapat dikatakan bahwa *Applications Framework* ini merupakan *layer* yang dapat digunakan pembuat aplikasi melakukan pengembangan/ pembuatan aplikasi yang dijalankan di sistem operasi Android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-provider* yang berupa SMS dan panggilan telepon.

Komponen yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut.

- a. *Views*.
- b. *Content Provider*.
- c. *Resource Manager*.

d. *Notifications Manager*.

e. *Activity Manager*.

2.7.2.3 Libraries

Libraries adalah *layer* yang merupakan tempat fitur-fitur Android berada. Para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas *kernel Linux*, *layer* ini meliputi berbagai *library C/C++* inti seperti *Libe* dan *SSL*, serta:

- a. *libraries* media untuk pemutaran media audio dan video.
- b. *libraries* untuk manajemen tampilan.
- c. *libraries graphics* mencakup *SGL* dan *OpenSGL* untuk grafis *2D* dan *3D*.
- d. *libraries SQLite* untuk dukungan *database*.
- e. *libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*.
- f. *libraries LiveWebcote* mencakup modern *web browser* dengan *engine embedded web view*.
- g. *libraries 3D* yang mencakup implementasi *OpenGL ES 1.0 API's*.

2.7.2.4 Android Run Time

Android Run Time merupakan *layer* yang membuat aplikasi Android dapat dijalankan yang dalam prosesnya menggunakan implementasi *Linux*. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi *Android*. *Android Run Time* dibagi menjadi dua bagian, yaitu.

- a. *Core Libraries*, yaitu aplikasi Android dibangun dalam bahasa *Java*, sementara *Dalvik* sebagai mesin virtualnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menerjemahkan bahasa *Java/ C* yang ditangani oleh *Core Libraries*.
- b. *Dalvik Virtual Machine*, yaitu mesin virtual berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dan merupakan pengembangan yang mampu membuat *Linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

2.7.2.5 Linux Kernel

Linux Kernel adalah *layer* inti dari sistem operasi Android itu berada. Berisi *file-file system* yang mengatur *system processing, memory, resource, drivers*, dan sistem-sistem operasi Android lainnya. *Linux kernel* yang digunakan Android adalah *Linux Kernel release 2.6*. (Safaat, 2012).

2.7.3 Komponen Aplikasi Android

Aplikasi Android ditulis dalam bahasa pemrograman *Java*. Kode *Java* dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, prosesnya di-*package* oleh *tools* yang dinamakan “*apt tools*” kedalam paket Android sehingga menghasilkan *file* dengan ekstensi *.apk (dot apk)*. *File apk* ini yang disebut dengan aplikasi, dan kemudian dapat di-*install* di perangkat Android.

2.7.3.1 Activities

Suatu *activity* menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi Android mungkin hanya terdapat satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan oleh pengguna. Untuk pindah dari satu *activity* ke *activity* yang lainnya, dapat dilakukan dengan satu *event*, misalnya klik tombol, memilih opsi atau menggunakan *trigger* tertentu. Secara hirarki sebuah *windows activity* dinyatakan dengan *method* `Activity setContentView()`. `ContentView` adalah objek yang berada pada *root* hirarki.

2.7.3.2 Service

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Misalnya *media player* sedang memutar lagudari *list* yang ada, aplikasi ini memiliki dua atau lebih *activity* yang memungkinkan pengguna untuk memilih lagu, atau menulis SMS sambil *media player* tetap berjalan. Untuk menjaga musik tetap dijalankan, *activity player* dapat menjalankan *service*. *Service* dijalankan pada *thread* utama dari proses aplikasi.

2.7.3.3 Broadcast Receiver

Broadcast Provider berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Sebagai contoh *broadcast* seperti notifikasi zona waktu berubah, baterai *low*, gambar telah selesai diambil oleh kamera, atau perubahan referensi bahasa yang digunakan.

Broadcast receiver tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang diterima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau getaran perangkat.

2.7.3.4 Content Provider

Content Provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file system* seperti *database SQLite*. *Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika menggunakan aplikasi yang membutuhkan peta (*map*), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka inilah fungsi *Content Provider* untuk menyediakan data yang diperlukan oleh aplikasi (Safaat, 2012).

2.7.4 Distribusi Android

Perangkat *mobile* yang pertama menggunakan sistem operasi Android untuk menjalankan perangkatnya adalah *HTC Dream*, yang dirilis pada 22 Oktober

2008. Saat ini, 900 juta orang di seluruh dunia telah menggunakan perangkat *mobile* berbasis Android, baik itu *smartphone* maupun *tablet* PC (Lam, 2013).

Android menggunakan penamaan yang cukup unik dan menarik bagi setiap versinya, yaitu menggunakan nama-nama makanan penutup (*dessert*) yang bercita rasa manis.

2.8 Android Development Tools

Dalam proses merancang dan membangun aplikasi *ShareLoc* para perangkat *mobile* berbasis Android, diperlukan beberapa *tools* sebagai berikut.

2.8.1 Android Software Development Kit (SDK)

Android SDK merupakan *tools* API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi Android dengan menggunakan bahasa pemrograman *Java* (Developer, 2014).

2.8.2 Android Development Tools (ADT)

Android Development Tools merupakan suatu *plugin* yang dirancang untuk *IDE Eclipse* agar mempermudah dalam membangun dan mengembangkan suatu aplikasi Android pada *IDE Eclipse*. Dengan *ADT*, pengembang dapat mem-*package source* menjadi satu *file installer* Android, yaitu *file* dengan ekstensi *.apk* (*dot apk*). Semakin tinggi *platform* atau versi Android yang digunakan, maka semakin tinggi pula *ADT* yang digunakan (Developer, 2014).

2.8.3 Eclipse

Eclipse merupakan komunitas *open source* yang bertujuan menghasilkan *platform* pemrograman terbuka. *Eclipse* terdiri dari *framework* yang dapat dikembangkan lebih lanjut, peralatan bantu untuk membuat dan mengelola *software* sejak awal hingga diluncurkan. *Platform Eclipse* didukung oleh ekosistem besar yang terdiri dari vendor teknologi, *start-up inovative*, *university*, *institutional and individual research*. Banyak orang mengenal *Eclipse* sebagai IDE (*Integrated Development Environment*) untuk bahasa pemrograman *Java*, tetapi *Eclipse* lebih dari sekedar IDE untuk *Java*.

Secara umum, *Eclipse* digunakan untuk membangun *software* inovatif berstandar industri, dan alat bantu beserta *framework*-nya membantu pekerjaan menjadi lebih mudah. Komunitas *Eclipse* memiliki lebih dari 60 proyek *open source*. Proyek-proyek tersebut secara konsep terbagi menjadi 7 kategori, yaitu.

- a. *Enterprise Development*.
- b. *Embedded and Device Development*.
- c. *Rich Client Platform*.
- d. *Rich Internet Applications*.
- e. *Application Frameworks*.
- f. *Application Lifecycle Management (ALM)*.
- g. *Service Oriented Architecture (SOA)*.

Eclipse menggunakan EPL (*Eclipse Public License*), yaitu lisensi yang memungkinkan organisasi untuk menjadikan *Eclipse* sebagai produk komersialnya, dan saat yang sama meminta orang yang melakukan perubahan

untuk berkontribusi hasilnya kembali kepada komunitas (Eclipse, 2014).

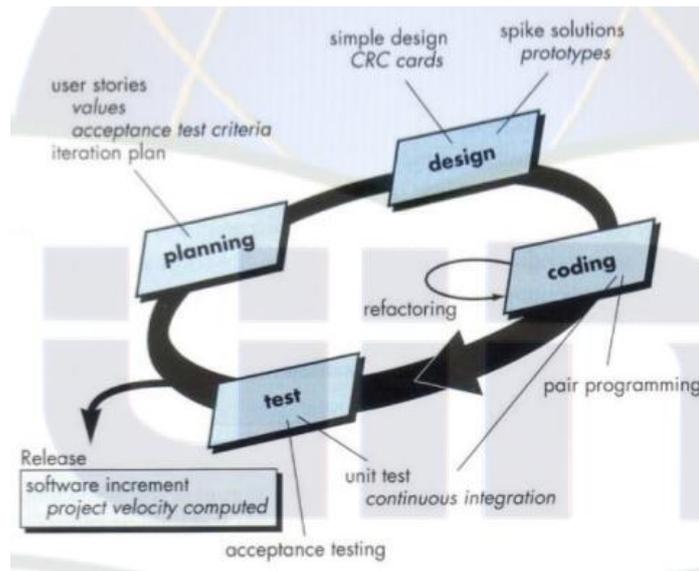
2.9 Metodologi Pengembangan Sistem

Metodologi pengembangan sistem yang digunakan yaitu *Agile Software Development* dan proses model menggunakan *Extreme Programming (XP)*, sedangkan desain menggunakan *Unified Modeling Language (UML)*.

2.9.1 *Extreme Programming (XP)*

Model proses ini diciptakan dan dikembangkan oleh Kent Beck. *Extreme Programming (XP)* adalah sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel. Walaupun menggunakan kata programming, XP tidak hanya terfokus pada coding tetapi meliputi seluruh area pengembangan perangkat lunak.

Menurut Kent Beck, XP ringan, efisien, resiko rendah, mudah disesuaikan, dapat diprediksi, ilmiah dan mudah dikembangkan. Suatu model yang menekankan pada keterlibatan user secara langsung, pengujian dan *pay-as-you-go design*.



Gambar 2.7 Siklus Extreme Programming (Pressman, 2010).

Siklus *Extreme Programming* membagi sebuah proyek menjadi 4 fase besar (Pressman, 2010), antara lain.

1. *Planning*, pengumpulan user stories dari klien yang klien tetapkan prioritasnya. Setiap story ditetapkan harga dan lama pembangunan, jika terlalu besar, story dapat dipecah menjadi beberapa story yang lebih kecil. Periksa dan pertimbangkan resiko.
2. *Design*, menggunakan *Unified Modeling Language (UML)* serta memanfaatkan kartu CRC (*Class Responsibility-Collaborator*) untuk identifikasi dan mengatur class-class pada konsep *Object Oriented*. Jika menemui kesulitan, *prototype* dibangun (ini dinamakan *spike solution*). Lakukan *refactoring*, yaitu mengembangkan desain setelah pengkodean.
3. *Coding*, menyiapkan unit *test* sebelum pengkodean dipakai sebagai fokus pemrogram untuk membuat program. *Pair programming* dilakukan untuk *real time program solving* dan *real time quality assurance*.

4. *Testing*, menggunakan unit *test* yang dipersiapkan sebelum pengkodean.

2.9.2 Agile Software Development

Kata *Agile* berarti bersifat cepat, ringan, bebas bergerak dan waspada. Kata *Agile* diambil dari kata *Agility*. Menurut Steven Goldman et al., “*Agility* adalah dinamis, konten yang spesifik, perubahan yang agresif, dan berorientasi pada pembangunan”. Konsep *Agile Software Development* dicetuskan oleh Kent Beck dan 16 rekannya. Mereka tergabung dalam *Agile Alliance*, lalu mereka menandatangani “*Manifesto for Agile Software Development*”.

1. Interaksi dan personel lebih penting daripada proses dan alat.
2. Software yang berfungsi lebih penting daripada dokumentasi yang lengkap.
3. Kolaborasi dengan klien lebih penting dari pada negosiasi kontrak.
4. Sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana (Pressman, 2010).

Namun demikian, sama seperti model proses yang lain, *Agile Software Development* memiliki kelebihan dan tidak cocok untuk semua jenis proyek, produk, orang dan situasi. *Agile Software Development* memungkinkan proses model yang toleransi terhadap perubahan kebutuhan sehingga perubahan dapat cepat ditanggapi. Namun disisi lain menyebabkan produktifitas menurun.

Menurut *Agile Alliance* dalam *Agile Manifesto* (Pressman 2010), ada 12 prinsip bagi mereka yang ingin berhasil dalam penerapan *Agile Software Development*.

1. Kepuasan klien adalah prioritas utama dengan menghasilkan produk lebih awal dan terus menerus.
2. Menerima perubahan kebutuhan, sekalipun diakhir pengembangan.
3. Penyerahan hasil/*software* dalam hitungan waktu dua minggu sampai dua bulan.
4. Bagian bisnis dan pembangun kerjasama tiap hari selama proyek berlangsung.
5. Membangun proyek di lingkungan orang-orang yang bermotivasi tinggi yang bekerja dalam lingkungan yang mendukung dan yang dipercaya untuk dapat menyelesaikan proyek.
6. Komunikasi dengan berhadapan langsung adalah komunikasi yang efektif dan efisien.
7. *Software* yang berfungsi adalah ukuran utama dari kemajuan proyek.
8. Dukungan yang stabil dari sponsor, pengembang, dan pengguna diperlukan untuk menjaga perkembangan yang berkesinambungan.
9. Perhatian kepada kehebatan teknis dan desain yang bagus meningkatkan sifat *agile*.
10. Kesederhanaan adalah penting.
11. Arsitektur, kebutuhan dan desain yang bagus muncul dari tim yang mengatur dirinya sendiri.
12. Secara periodik tim evaluasi diri dan mencari cara untuk lebih efektif dan segera melakukannya (Pressman, 2010).

2.9.3 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP). Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif.

Unified Modeling Language merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group (OMG)*, sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar COBRA (*Common Object Request Broker Architecture*).

UML lahir dari penggabungan banyak bahasa permodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corp.* UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Fowler, 2004).

2.9.3.1 Use Case Diagram

Use Case Diagram adalah diagram yang mendeskripsikan interaksi antara pengguna dengan aplikasi. Kesimpulannya *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem (Tangguh, 2013). Simbol dan keterangan *use case* pada Tabel 2.1.

Tabel 2.1. Simbol dan keterangan *use case*

SIMBOL	KETERANGAN
Aktor 	Mewakili peran orang, system yang lain atau alat ketika berkomunikasi dengan <i>use case</i> .
	Abstraksi dari interaksi antara sistem dan actor.
Association 	Abstraksi dari penghubung antara actor dan <i>use case</i> .
Generalisasi 	Menunjukkan spessialisasi actor untuk dapat berpartisipasi dalam <i>use case</i> .
Extend 	Mempesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
Include 	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.

2.9.3.2 Activity Diagram

Activity Diagram atau Diagram Aktivitas menggambarkan alur aktivitas dalam aplikasi, menjelaskan proses masing-masing alur berawal dan proses aplikasi

berakhir. Diagram aktivitas juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi (Tangguh, 2013). Simbol dan keterangan *activity diagram* seperti pada Tabel 2.2.

Tabel 2.2. Simbol dan keterangan *activity diagram*

SIMBOL	KETERANGAN
Status Awal 	Status awal aktivitas system, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan system, aktivitas biasanya diawali dengan kata kerja.
Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Status akhir 	Status akhir yang dilakukan system, sebuah diagram aktivitas memiliki sebuah status akhir.

2.9.3.3 Class Responsibility – Collaboration (CRC)

Class Responsibility – Collaboration adalah sebuah teknik untuk mengecek adanya interaksi antar obyek. Sebelah kiri berisi tanggung jawab, sebelah kanan berisi kelas yang melaksanakan tanggung jawab tersebut, seperti terlihat pada Gambar 2.8.

Class	
Responsibility	Collaborator

Gambar 2.8 *Class Responsibility – Collaboration (CRC)*

Dari sini akan didapatkan satu set kelas dan tanggungjawabnya, serta kelas lain yang berhubungan dengan kelas tersebut.

2.9.3.4 *Class Diagram*

Class Diagram atau Diagram Kelas merupakan diagram yang memodelkan sekumpulan kelas, *interface*, kolaborasi dan relasinya. Diagram kelas digambarkan dengan bentuk kotak (Tangguh, 2013). Simbol dan keterangan *class diagram* seperti pada Tabel 2.3.

Tabel 2.3. Simbol dan keterangan *Class Diagram*

SIMBOL	KETERANGAN
CLASS 	Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.
Nary Association 	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
Generalization 	Hubungan dimana objek anak (descendent) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (oncestor).

Tabel 2.3. Simbol dan keterangan *Class Diagram* (Lanjutan)

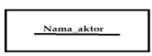
SIMBOL	KETERANGAN
	Operasi yang benar-benar dilakukan oleh suatu objek.

2.9.3.5 *Sequence Diagram*

Sequence Diagram biasanya digunakan untuk tujuan analisa dan desain, memfokuskan pada identifikasi metode di dalam sebuah sistem (Tangguh, 2013).

Simbol dan keterangan *sequence diagram* seperti pada Tabel 2.4.

Tabel 2.4. Simbol dan keterangan *Sequence Diagram*

SIMBOL	KETERANGAN
	Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan dan ditempatkan di bagian atas diagram.
<p>OBJEK</p> 	Sebuah objek: Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan dan ditempatkan di bagian atas diagram.
<p>Garis hidup objek</p> 	Menandakan kehidupan obyek selama urutan dan diakhiri tanda X pada titik dimana kelas tidak lagi berinteraksi.

Tabel 2.4. Simbol dan keterangan *Sequence Diagram* (Lanjutan)

SIMBOL	KETERANGAN
<p>Objek sedang aktif berinteraksi</p> 	<p>Fokus kontrol: Persegi panjang yang sempit panjang ditempatkan diatas sebuah garis hidup dan menandakan ketika suatu objek mengirim atau menerima pesan.</p>
<p>pesan</p> 	<p>Objek mengirim satu pesan ke objek lainnya.</p>
<p>Create</p> 	<p>Menyatakan suatu objek membuat objek yang lain,arah panah mengarah pada objek yang dibuat.</p>
<p>Masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan masukan ke objek lainnya arah panah mengarah pada objek yang dikirim.</p>
<p>Keluaran</p> 	<p>Objek metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Destroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah objek yang diakhiri.</p>

2.10 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses menjalankan dan mengevaluasi sebuah perangkat lunak secara manual maupun otomatis untuk menguji apakah perangkat lunak sudah memenuhi persyaratan atau belum (Clune, 2011). Singkat kata, pengujian adalah aktivitas untuk menemukan dan menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya.

Pendekatan *Black-Box* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Jiang, 2012). Teknik pengujian yang digunakan dalam penelitian ini terbagi atas pengujian fungsional dengan menggunakan metode *Black Box* yaitu *Equivalence Partitioning*.

2.10.1 *Equivalence Partitioning*

Equivalence Partitioning (EP) merupakan metode *Black Box testing* yang membagi domain masukan dari program kedalam kelas-kelas sehingga *test cases* dapat diperoleh. *Equivalence Partitioning* berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. Kasus uji yang didesain untuk *Equivalence Partitioning* berdasarkan pada evaluasi dari kelas ekuivalensi untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak. Kondisi masukan dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi *boolean*. Kesetaraan kelas dapat didefinisikan menurut panduan berikut (Pressman, 2010).

1. Jika masukan kondisi menentukan kisaran, satu sah dan dua diartikan tidak valid kesetaraan kelas.
2. Jika masukan membutuhkan nilai, kondisi tertentu satu sah dan dua tidak valid kesetaraan kelas diartikan.
3. Jika masukan kondisi menentukan anggota dari set, satu sah dan satu tidak valid kesetaraan kelas diartikan.
4. Jika kondisi yang *input, boolean* satu sah dan satu tidak *valid* kelas diartikan.

BAB III

METODOLOGI PENELITIAN

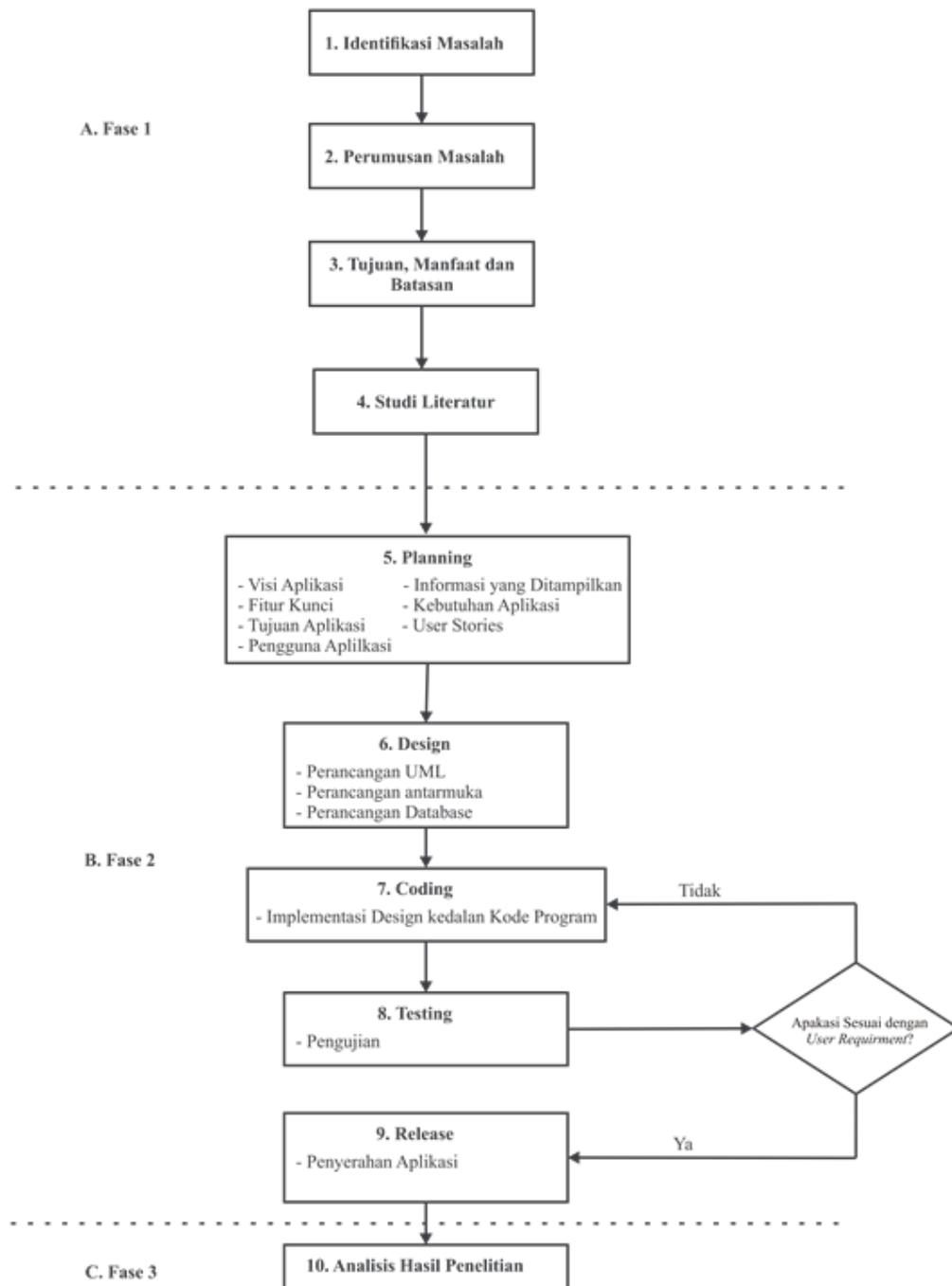
3.1 Waktu dan Tempat Penelitian

Penelitian dilaksanakan di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung yang berada di jalan Soemantri Brojonegoro No.1 Gedung Meneng, Bandar Lampung. Penelitian ini dilakukan pada semester ganjil tahun ajaran 2015-2016.

3.2 Metodologi Penelitian

3.2.1 Alur Penelitian

Penelitian dilakukan berdasarkan diagram alir metodologi penelitian yang terdapat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

Penjelasan dari diagram alir metodologi penelitian pada Gambar 3.1 adalah sebagai berikut.

A. Fase 1

1. Tahap pertama pada fase pertama adalah mengidentifikasi masalah, yaitu pengenalan masalah untuk menentukan masalah yang akan dipecahkan dalam penelitian.
2. Tahap kedua adalah perumusan masalah, yaitu pertanyaan penelitian yang membutuhkan jawaban melalui pengumpulan data yang umumnya disusun dalam bentuk kalimat tanya.
3. Tahap ketiga adalah menentukan tujuan, manfaat dan batasan. Target pencapaian dalam tahap ini adalah diketahuinya tujuan dan manfaat dari Aplikasi *Location Based Service* (LBS) Untuk Berbagi Lokasi Menggunakan *Short Message Service* (SMS) Berbasis Mobile Android. Sedangkan batasan digunakan untuk membatasi pembahasan dan ruang lingkup penelitian.
4. Tahap keempat adalah studi literatur, yaitu untuk mendapatkan gambaran yang menyeluruh tentang apa yang sudah dikerjakan sebagai teori-teori yang akan dijadikan landasan penelitian.

B. Fase 2

5. Tahap kelima pada fase ke-dua merupakan pengembangan sistem. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Extreme Programming*. Tahap pertama yang dilakukan dalam pengembangan sistem ini adalah *planning*. Pada tahapan ini

perencanaan terhadap *software* yang diinginkan mengacu pada *user stories*. *User stories* menggambarkan fitur dan fungsi yang dibutuhkan terhadap *software* tersebut.

6. Tahap keenam adalah *Design*. Pada tahap ini akan dilakukan perancangan sistem dan perancangan *interface* yang menggunakan *Unified Modelling Language* (UML), pada tahap ini akan diketahui semua entitas luar, input dan output yang terlibat dalam sistem serta *usecase*, *class diagram*, *class responsibility collaborator* (CRC), *activity diagram*, *sequence diagram* yang digunakan dalam analisis sistem. Selanjutnya akan dilakukan perancangan *interface*, perancangan ini dilakukan untuk merancang tata letak sistem sesuai dengan analisis kebutuhan sistem.
7. Tahap ketujuh adalah *coding*. Pada tahap ini akan dilakukan pembuatan program yang akan mengimplementasikan desain yang sebelumnya telah dibuat.
8. Tahap kedelapan adalah *testing*. Program yang telah selesai akan diuji menggunakan *Black Box* dengan metode *Equivalence Partitioning* (EP). Jika sistem tidak bekerja sesuai analisis, maka kembali ke tahap *coding*, namun jika sistem bekerja sesuai analisis maka dilakukan tahap selanjutnya.
9. Tahap kesembilan adalah *release*. Pada tahap ini akan dilakukan penyerahan sistem aplikasi ke-*user* (*roll-out*) melalui PlayStore.

C. Fase 3

10. Tahap kesepuluh adalah analisis hasil penelitian. Pada tahap ini menjelaskan berhasil atau tidaknya dalam suatu penelitian dengan permasalahan yang telah dideskripsikan di awal.

3.2.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang dipilih dalam penelitian ini adalah *Extreme Programming* (XP). Tahap-tahap yang dilakukan dalam pengembangan sistem aplikasi ini adalah.

1) *Planning*

Pada tahapan ini perencanaan terhadap *software* yang diinginkan mengacu pada *user stories*. *User stories* menggambarkan fitur dan fungsi yang dibutuhkan terhadap *software* tersebut. Ketika semua *user stories* telah ditentukan, *developer* akan menentukan lama pengerjaan untuk tiap-tiap *user stories*.

Adapun beberapa fungsi dan kebutuhan dari aplikasi adalah sebagai berikut:

- a. Menentukan visi aplikasi,
- b. Menentukan fitur kunci,
- c. Menentukan tujuan aplikasi,
- d. Menentukan pengguna aplikasi,
- e. Menentukan informasi yang dibutuhkan,
- f. Mengidentifikasi dan menganalisis kebutuhan aplikasi,

- g. Membuat *User Stories* sebagai inti perencanaan dari XP.

2) *Design*

Perancangan sistem dan perancangan *interface* berlangsung pada tahap ini.

Perancangan atau desain sistem dalam penelitian ini menggunakan *Unified Modelling Language* (UML). Diagram-diagram UML yang dibuat dalam penelitian ini adalah sebagai berikut.

- a. *Use casediagram.*
- b. *Activitydiagram.*
- c. *Class diagram.*
- d. *Class Responsibility Collaborator.*
- e. *Sequence Diagram.*

Perancangan *interface* (antarmuka) yang dirancang dalam aplikasi ini adalah sebagai berikut.

- a. *Layoutsplash screen.*
- b. *Layoutmenu* utama (Posisi Sekarang).
- c. *Layout menu* Riwayat.
- d. *Layout Maps*
- e. *Layout menu* Bantuan.
- f. *Layout menu* Tentang aplikasi.

3) *Coding*

Pada tahap ini, proses pengembangan tidak langsung melakukan implementasi terhadap desain yang telah dibuat. Pembuatan *unit test* untuk tiap-tiap *stories* yang nantinya akan diimplementasikan. Saat *unit test* selesai dibuat, pengembang lebih baik fokus terhadap apa yang akan diimplementasikan untuk melewati *unit test*. Tahap *coding* ini akan mengacu pada desain yang sebelumnya telah dibuat. Karena pembuatan *unit test* dilakukan terlebih dahulu. Maka implementasi desain sebaiknya dibuat untuk melewati *unit test* yang dibuat.

Proses pembuatan program (*coding*) dilakukan pada tahap ini menggunakan bahasa pemrograman Java dengan bantuan aplikasi Eclipse. Pada aplikasi ini dibuat beberapa *class* java yang menjelaskan jalannya aplikasi khususnya pada fungsi mengirim dan menerima data lokasi melalui *Short Message Service* (SMS), serta menampilkannya dalam bentuk peta. Aplikasi *ShareLoc* ini menggunakan *library* API dari Google Maps dengan peta yang digunakan bersumber dari satelit Google. Aplikasi ini juga membutuhkan akses internet, akses SMS, serta pengaturan akses GPS yang harus dihidupkan pada perangkat Android. Setelah semua terpenuhi di dalam *class* java, maka selanjutnya aplikasi akan menampilkan *friend's location* (lokasi teman) untuk pengguna kedalam Google Maps *Viewer*, yaitu dengan mengambil data koordinat melalui SMS yang dikirim otomatis oleh *friend's smartphone* yang telah ter-*install* aplikasi ini sebagai balasan SMS dari *user*.

4) *Testing*

Pada tahap ini, dilakukan pengujian *black box* dengan metode *Equivalence Partitioning* (EP).

5) *Release*

Play Store adalah layanan konten digital milik Google yang mencakup toko untuk produk-produk seperti musik/lagu, buku, aplikasi, permainan, ataupun pemutar media berbasis *cloud*. Pada tahap ini adalah tahap *upload* aplikasi *ShareLoc* ke dalam Play Store, proses ini dilakukan agar pengguna dapat mengunduh aplikasi melalui Play Store secara gratis.

3.2.3 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan pada penelitian ini adalah Studi Literatur. Studi literatur yang digunakan yaitu buku-buku, jurnal, prosiding dan internet yang menyajikan informasi tentang GIS, Eclipse, Android dan bahasa pemrograman Java. Studi literatur ini juga akan menggunakan aplikasi yang sejenis sebagai pembanding.

3.3 Analisis dan Perancangan Sistem

3.3.1 Identifikasi Masalah

Indonesia merupakan negara yang sangat luas, dengan luas yang mencapai 1.910.931,32 Km (Badan Pusat Statistik, 2015). Serta Indonesia yang terbagi menjadi 34 provinsi memiliki 98 kota, 416 kabupaten, 7.024 kecamatan, dan 81.626 desa (Badan Pusat Statistik, 2015). Setiap provinsi, kota, kabupaten,

kecamatan, dan desa memiliki nama yang beragam, namun tidak jarang ditemukannya nama yang sama. Nama-nama tersebut digunakan untuk membuat alamat dari suatu daerah untuk memudahkan masyarakat dalam mencari lokasi daerah. Alamat digunakan untuk tanda dari suatu lokasi, seperti lokasi rumah, kantor, tempat wisata, dan sebagainya. Alamat sangat penting untuk menemukan lokasi, karena alamat dapat digunakan untuk bertanya letak dari lokasi yang dicari, jika tidak ada alamat maka lokasi akan sulit ditemukan karena tidak adanya data untuk dicari atau ditanyakan.

Adanya alamat dari suatu lokasi sangat penting, namun tidak jarang masyarakat yang tidak tahu akan alamat lokasi mereka berada atau bahkan alamat rumah mereka, sehingga sulit untuk mencari keberadaan mereka. Tidak hanya pendatang yang tidak tahu alamat mereka berada, penduduk lokal pun terkadang lupa atau bahkan tidak tahu alamat setempat. Kurangnya pengetahuan akan alamat ini yang membuat suatu lokasi sulit untuk ditemukan. Untuk membantu memudahkan dalam pencarian lokasi seseorang atau lokasi suatu tempat, maka “Aplikasi *Location Based Service* (LBS) Untuk Berbagi Lokasi Menggunakan *Short Message Service* (SMS) Berbasis Mobile Android” ini dibangun. Aplikasi ini pun dibangun untuk dapat membantu pengguna dalam menyimpan koordinat lokasi dari tempat yang dikunjungi agar pengguna dapat memudahkan dalam mengingat lokasi tersebut, serta dapat berbagi lokasi yang telah mereka kunjungi kepada pengguna yang lain.

3.3.2 Analisis Sistem

Berdasarkan identifikasi masalah di atas, maka sistem yang dibutuhkan adalah.

1. Peta lokasi teman dengan rute dan navigasi untuk menunjukkan arah yang tepat menuju lokasi tersebut.
2. Peta lokasi yang telah disimpan pada *database* dengan rute dan navigasi untuk menunjukkan arah yang tepat menuju lokasi tersebut.
3. Peta lokasi tempat yang dikirim oleh pengguna lain dengan rute dan navigasi untuk menunjukkan arah yang tepat menuju lokasi tersebut.

3.3.3 Analisis User Requirement

Kebutuhan dasar aplikasi ini adalah sebagai berikut.

1. Sistem dapat menampilkan *Current Location* (Posisi Sekarang).
2. Sistem dapat mengirim SMS.
3. Sistem dapat menampilkan rute menuju lokasi yang dikirimkan melalui SMS.
4. Sistem dapat menyimpan *Current Location* (Posisi Sekarang) dan lokasi yang dikirimkan pengguna lain.
5. Sistem dapat mengirimkan lokasi yang telah disimpan.
6. Sistem dapat merespon beberapa aksi yang diberikan oleh *user*.

3.3.4 Perancangan Sistem

Perancangan sistem adalah tahap setelah analisis dari siklus pengembangan sistem. Perancangan sistem di sini berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan dapat berfungsi. Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang harus diselesaikan. Tahap ini termasuk mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah dilakukan instalasi akan benar-benar sesuai dengan rancangan awal sistem.

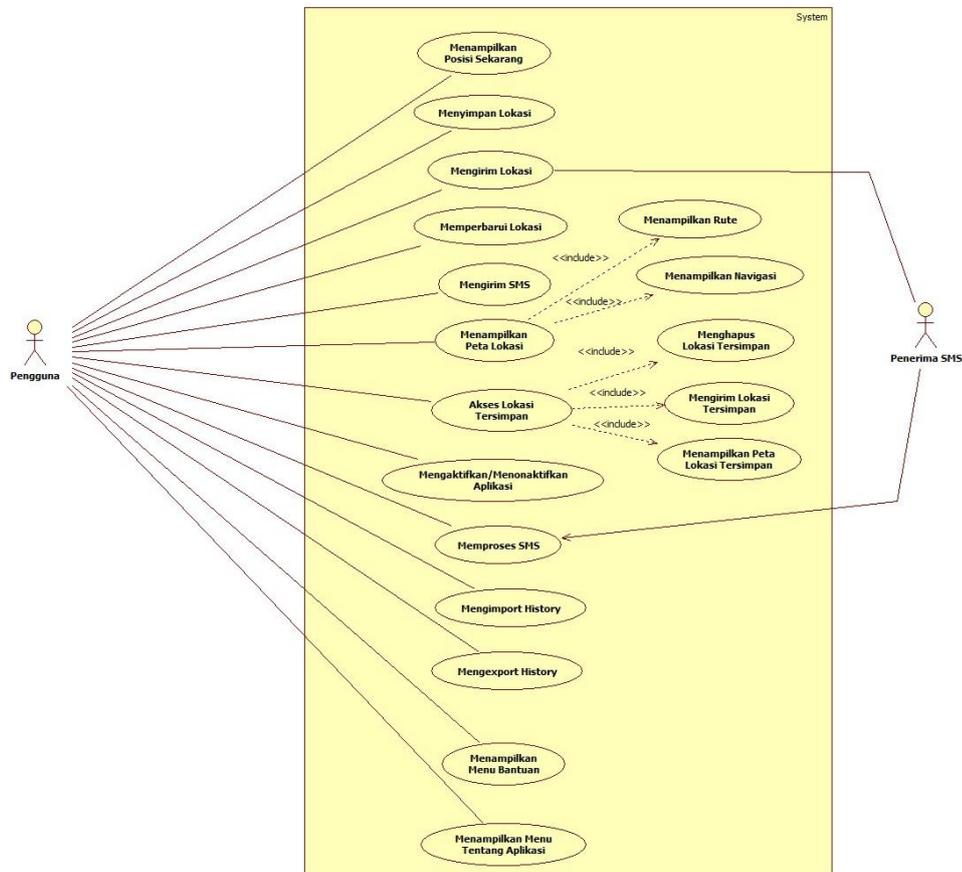
3.3.4.1 Perancangan UML (*Unified Modelling Language*)

Perancangan sistem atau desain system dilakukan dengan memodelkan permasalahan dalam bentuk diagram-diagram UML sebagai berikut.

1. *Use Case Diagram*

Use case diagram di bawah ini menjelaskan fungsionalitas dari aplikasi *ShareLoc* berbasis Android. Pada aplikasi ini, pengguna dapat melakukan 16 interaksi antara lain menampilkan posisi pengguna, menyimpan lokasi, mengirim lokasi, memperbarui lokasi, mengirim SMS, menampilkan peta lokasi serta rute dan navigasinya, akses lokasi tersimpan yang dapat dihapus dan dikirim ataupun ditampilkan kedalam map, mengaktifkan atau menonaktifkan aplikasi, memproses sms masuk, meng-*import* dan meng-*export* data tersimpan, menampilkan bantuan, dan menampilkan tentang

aplikasi. *Use case diagram* aplikasi *ShareLoc* dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Use Case Diagram*

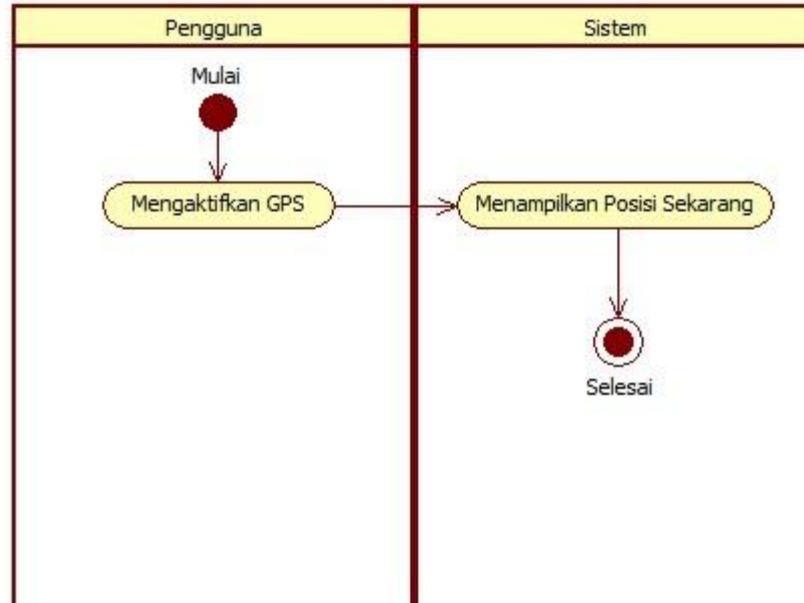
2. *ActivityDiagram*

Diagram aktivitas atau *Activity Diagram* menggambarkan aliran fungsionalitas sistem. Pada aplikasi *ShareLoc* terdapat 18 *activity diagram*, yaitu sebagai berikut.

a) *Activity Diagram* Posisi Sekarang

Activity Diagram Posisi Sekarang dimulai dengan pengguna mengaktifkan GPS, setelah itu sistem akan menampilkan detail

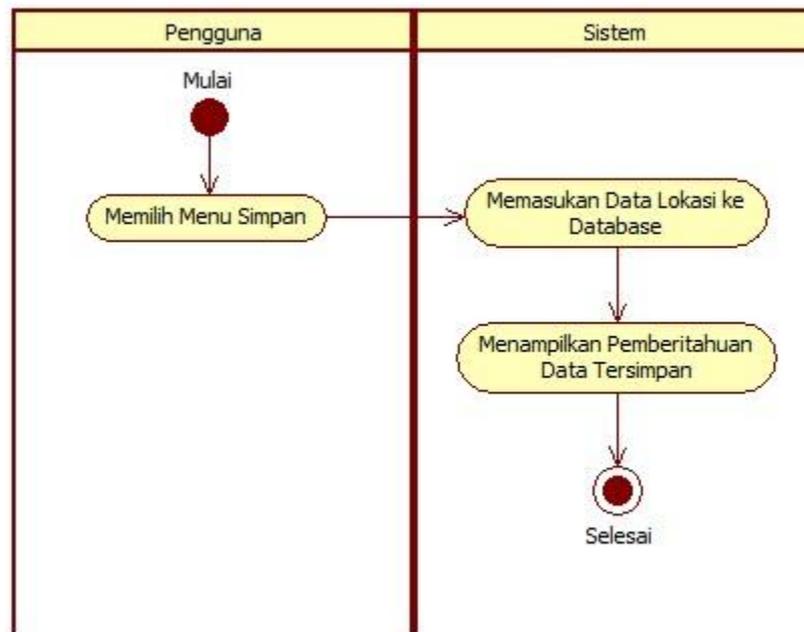
koordinat posisi pengguna saat itu. *Activity Diagram* Posisi Sekarang disajikan pada gambar Gambar 3.3.



Gambar 3.3. *Activity Diagram* Posisi Sekarang

b) *Activity Diagram* Menyimpan Lokasi

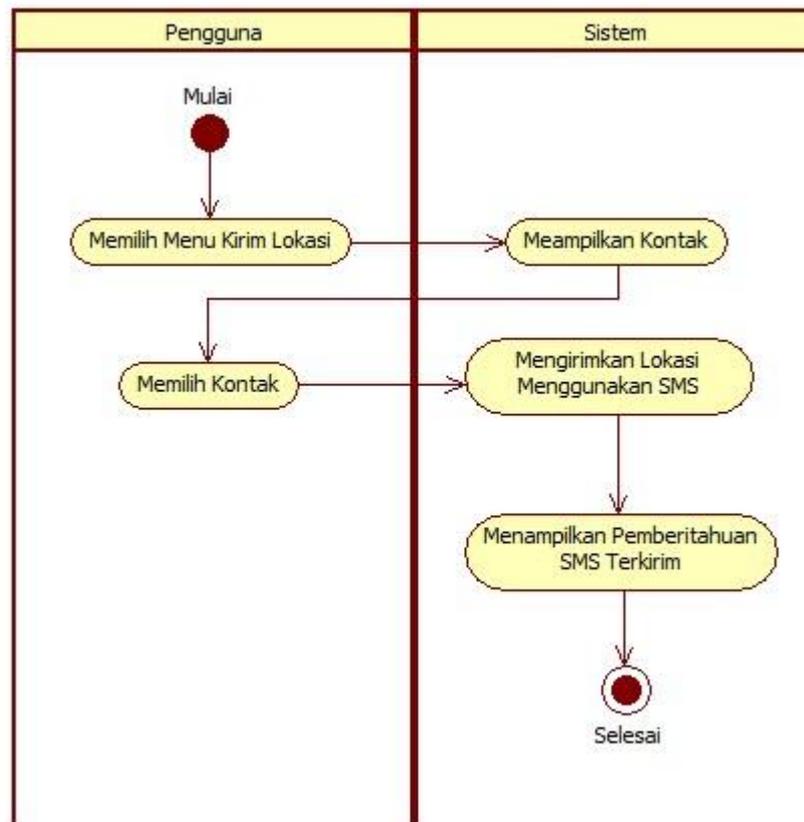
Activity Diagram Menyimpan Lokasi dimulai dengan pengguna memilih menu “Simpan” kemudian sistem akan menyimpan data lokasi pengguna kedalam database dan menampilkan pemberitahuan data tersimpan. *Activity Diagram* Menyimpan Lokasi disajikan pada Gambar 3.4.



Gambar 3.4. *Activity Diagram* Menyimpan Lokasi

c) *Activity Diagram* Mengirim Lokasi

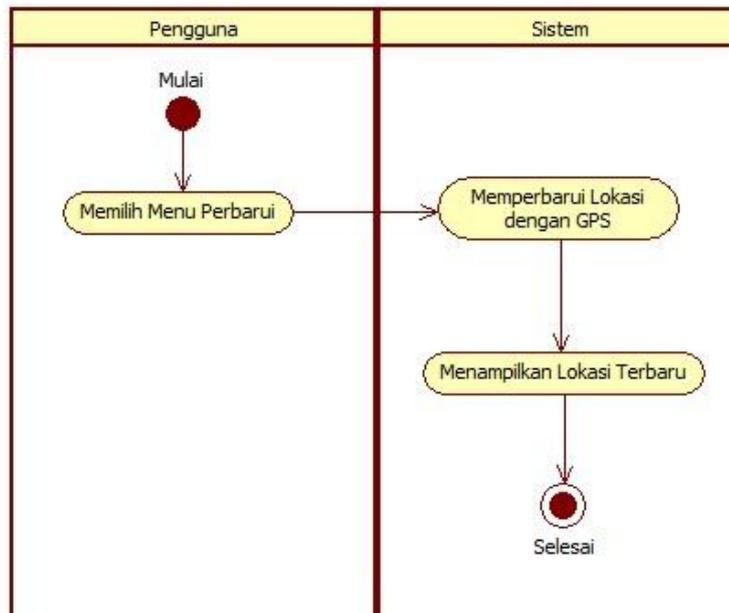
Activity Diagram Mengirim Lokasi dimulai dengan pengguna memilih menu “Kirim Lokasi” yang selanjutnya sistem akan menampilkan sub menu kontak yang ada pada *smartphone* pengguna. Setelah itu pengguna memilih kontak dan tekan tombol “Kirim”, kemudian sistem akan mengirimkan SMS yang berisi text detail lokasi dengan format SMS yang telah ditentukan kepada nomor yang telah dipilih sebelumnya, dan sistem akan menampilkan pemberitahuan status SMS, berhasil terkirim atau gagal. *Activity Diagram* Mengirim Lokasi disajikan pada Gambar 3.5.



Gambar 3.5. *Activity Diagram Mengirim Lokasi*

d) *Activity Diagram Memperbarui Lokasi*

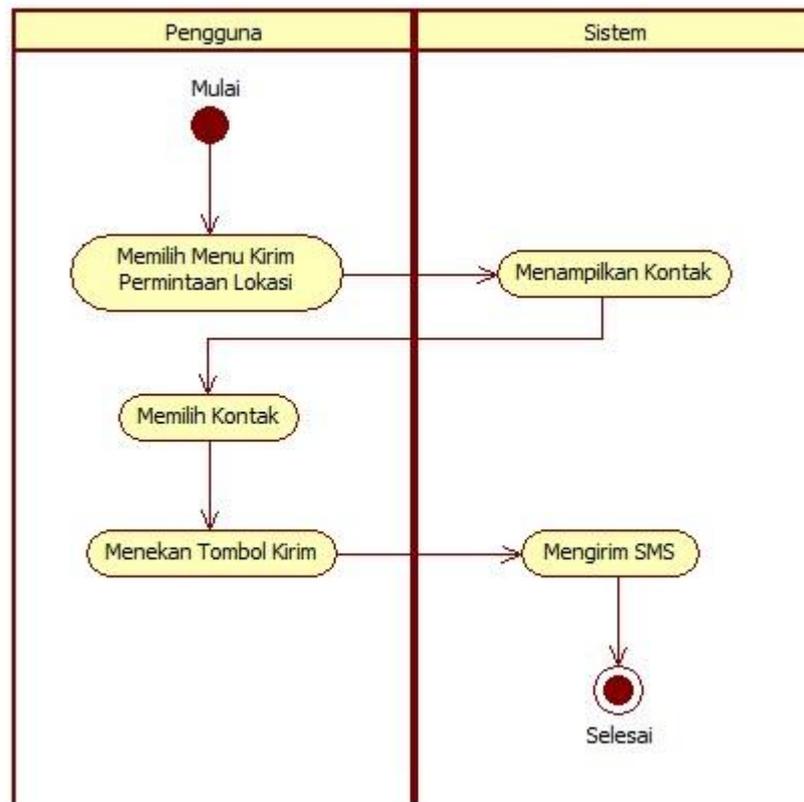
Activity Diagram Memperbarui Lokasi dimulai dengan pengguna memilih menu “Perbarui” yang kemudian sistem akan memperbarui lokasi dengan GPS dan menampilkan lokasi terbaru. *Activity Diagram Memperbarui Lokasi* disajikan dalam Gambar 3.6.



Gambar 3.6. *Activity Diagram* Memperbarui Lokasi

e) *Activity Diagram* Mengirim SMS

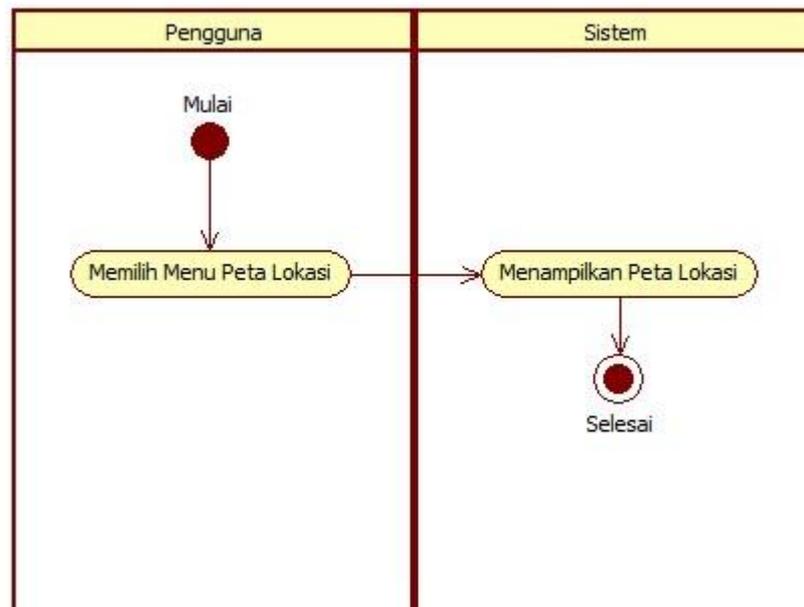
Activity Diagram Mengirim SMS dimulai dengan pengguna memilih menu “Kirim Permintaan Lokasi” yang selanjutnya sistem akan menampilkan sub menu kontak yang ada pada *smartphone* pengguna. Setelah itu pengguna memilih kontak dan tekan tombol “Kirim”, kemudian sistem akan mengirimkan SMS dengan format permintaan lokasi yang telah ditentukan kepada nomor yang telah dipilih sebelumnya. *Activity Diagram* Mengirim SMS disajikan pada Gambar 3.7.



Gambar 3.7. *Activity Diagram* Mengirim SMS

f) *Activity Diagram* Menampilkan Peta Lokasi

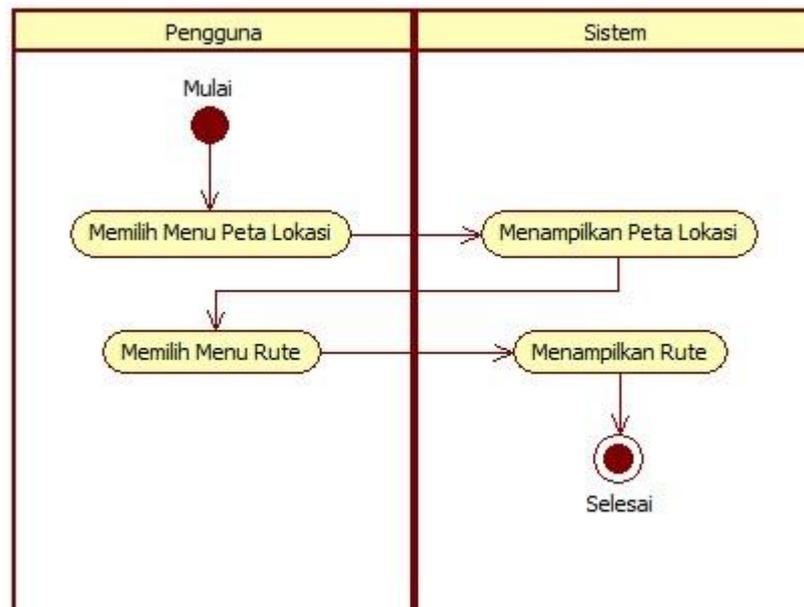
Activity Diagram Menampilkan Peta Lokasi dimulai dengan pengguna memilih menu “Peta Lokasi” dan kemudian sistem akan menampilkan peta lokasi pengguna saat itu. *Activity Diagram* Menampilkan Peta Lokasi disajikan dalam Gambar 3.8.



Gambar 3.8. *Activity Diagram* Menampilkan Peta Lokasi

g) *Activity Diagram* Menampilkan Rute

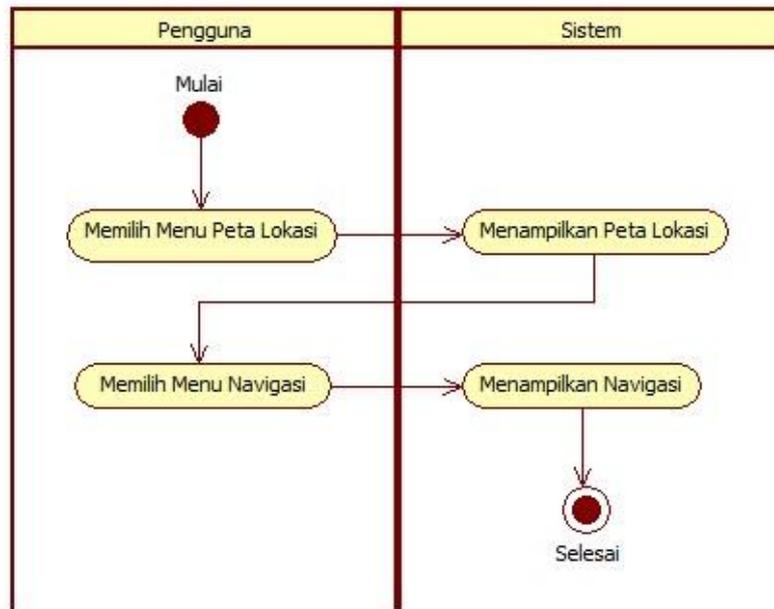
Activity Diagram Menampilkan Rute dimulai dengan pengguna memilih menu “Peta Lokasi” dan kemudian sistem akan menampilkan peta lokasi, setelah itu pengguna memilih menu “Rute” dan sistem akan menampilkan rute dari posisi pengguna ke tujuan yang telah ditandai berdasarkan koordinat yang didapat dari lokasi tersimpan ataupun lokasi yang diterima melalui SMS. *Activity Diagram* Menampilkan Rute disajikan dalam Gambar 3.9.



Gambar 3.9. *Activity Diagram* Menampilkan Rute

h) *Activity Diagram* Menampilkan Navigasi

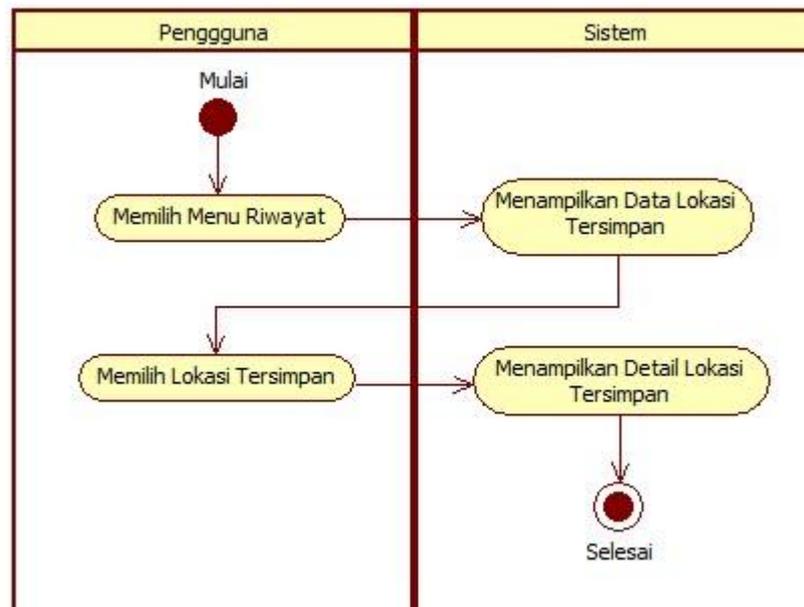
Activity Diagram Menampilkan Navigasi dimulai dengan pengguna memilih menu “Peta Lokasi” dan kemudian sistem akan menampilkan peta lokasi, setelah itu pengguna memilih menu “Navigasi” dan sistem akan membuka aplikasi Navigation yang dikembangkan oleh Google agar petunjuk jalan yang diberikan lebih baik. *Activity Diagram* Menampilkan Navigasi disajikan dalam Gambar 3.10.



Gambar 3.10. *Activity Diagram* Menampilkan Navigasi

i) *Activity Diagram* Akses Lokasi Tersimpan

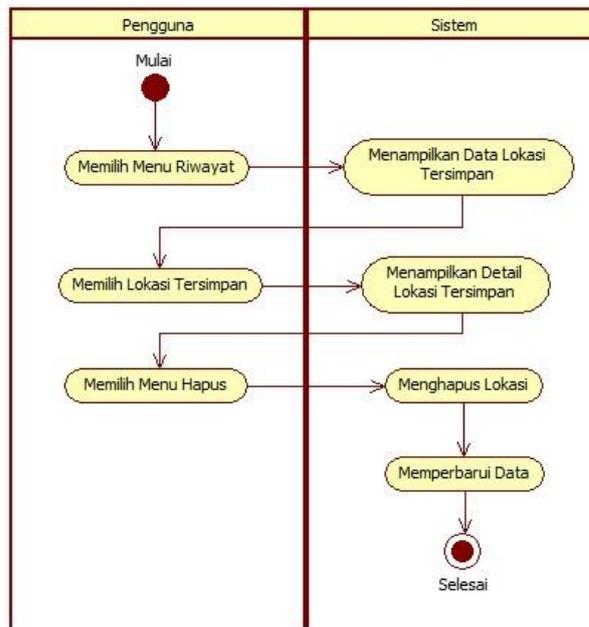
Activity Diagram Akses Lokasi Tersimpan dimulai dengan pengguna memilih menu “Riwayat” yang kemudian sistem akan menampilkan data lokasi tersimpan, setelah itu pengguna dapat memilih lokasi tersimpan dan menekan lokasi yang pengguna pilih untuk menampilkan detail lokasi tersimpan. *Activity Diagram* Akses Lokasi Tersimpan disajikan dalam Gambar 3.11.



Gambar 3.11. *Activity Diagram* Akses Lokasi Tersimpan

j) *Activity Diagram* Menghapus Lokasi Tersimpan

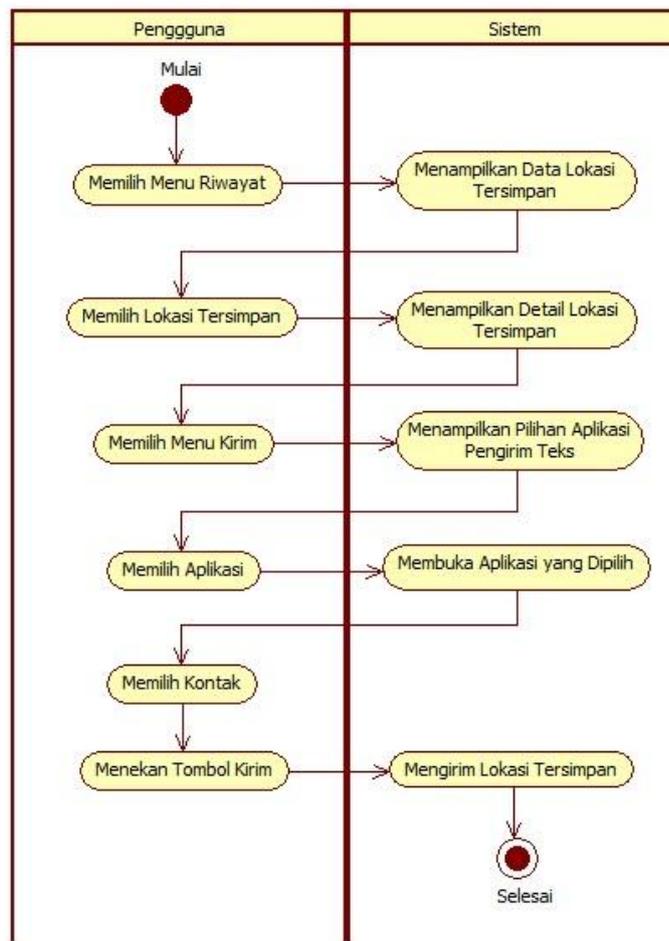
Activity Diagram Menghapus Lokasi Tersimpan dimulai dengan pengguna memilih menu “Riwayat” yang kemudian sistem akan menampilkan data lokasi tersimpan, setelah itu pengguna dapat memilih lokasi tersimpan dan menekan lokasi yang pengguna pilih untuk menampilkan detail lokasi tersimpan. Kemudian pengguna memilih menu “Hapus” dan sistem akan menghapus data tersimpan serta memperbarui data yang ditampilkan. *Activity Diagram* Menghapus Lokasi Tersimpan disajikan dalam Gambar 3.12.



Gambar 3.12. *Activity Diagram* Menghapus Lokasi Tersimpan

k) *Activity Diagram* Mengirim Lokasi Tersimpan

Activity Diagram Mengirim Lokasi Tersimpan dimulai dengan pengguna memilih menu “Riwayat” yang kemudian sistem akan menampilkan data lokasi tersimpan, setelah itu pengguna dapat memilih lokasi tersimpan dan menekan lokasi yang pengguna pilih untuk menampilkan detail lokasi tersimpan. Kemudian pengguna memilih menu “Kirim” dan sistem akan menampilkan pilihan aplikasi pengirim text, kemudian pengguna memilih aplikasi dan sistem akan membuka aplikasi yang dipilih, setelah pengguna memilih kontak atau menentukan tujuan pengiriman lokasi dan menekan tombol kirim maka sistem akan mengirim data lokasi dalam bentuk text. *Activity Diagram* Mengirim Lokasi Tersimpan disajikan dalam Gambar 3.13.

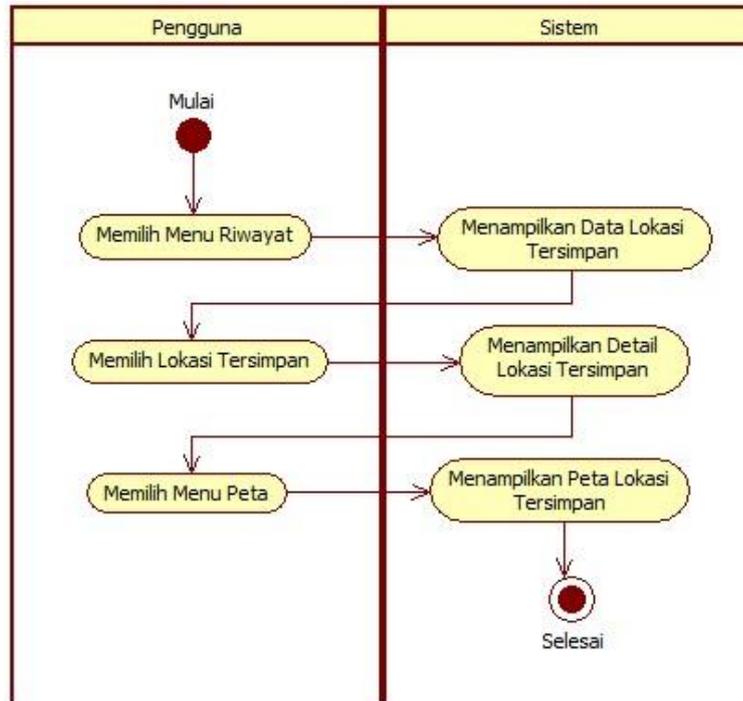


Gambar 3.13. *Activity Diagram* Mengirim Lokasi Tersimpan

1) *Activity Diagram* Menampilkan Peta Lokasi Tersimpan

Activity Diagram Menampilkan Peta Lokasi Tersimpan dimulai dengan pengguna memilih menu “Riwayat” yang kemudian sistem akan menampilkan data lokasi tersimpan, setelah itu pengguna dapat memilih lokasi tersimpan dan menekan lokasi yang pengguna pilih untuk menampilkan detail lokasi tersimpan. Kemudian pengguna memilih menu “Peta” dan sistem akan menampilkan data tersimpan kedalam Google Maps Viewer.

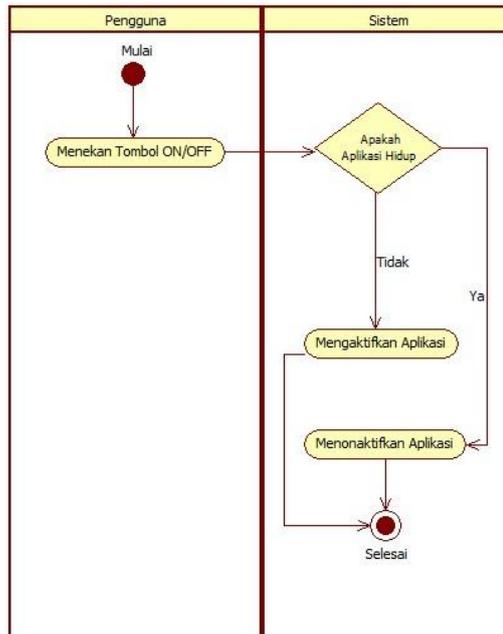
Activity Diagram Menampilkan Peta Lokasi Tersimpan disajikan dalam Gambar 3.14.



Gambar 3.14. *Activity Diagram* Menampilkan Peta Lokasi Tersimpan

m) *Activity Diagram* Mengaktifkan/Menonaktifkan Aplikasi

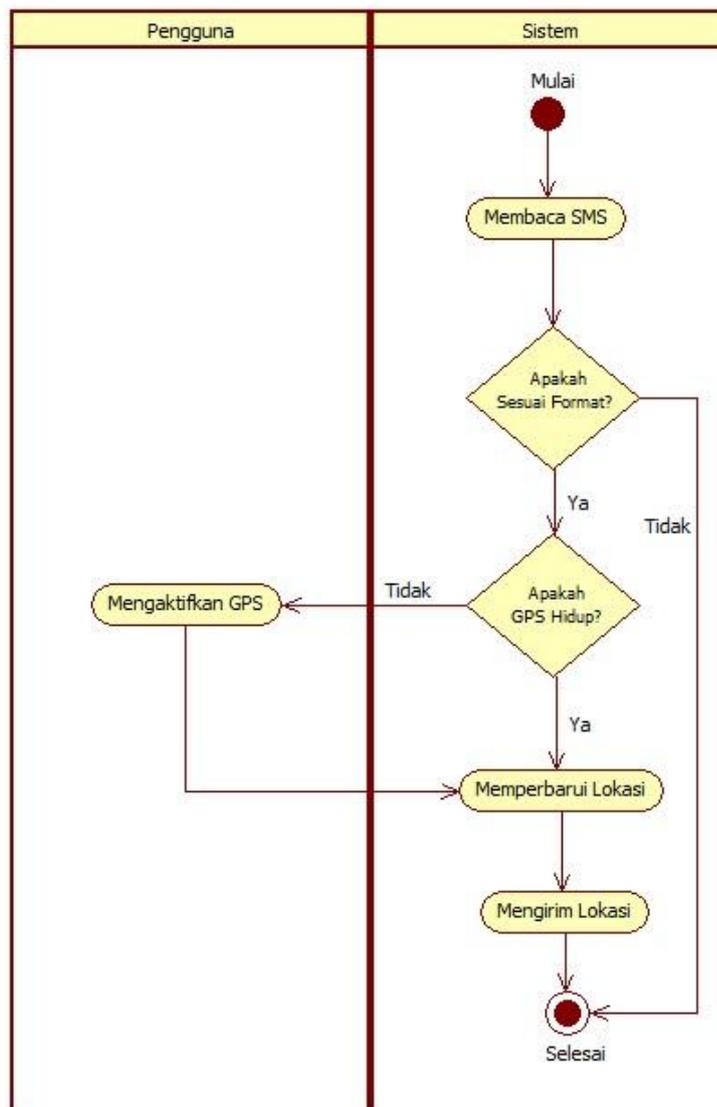
Activity Diagram Mengaktifkan/Menonaktifkan Aplikasi dimulai dengan pengguna menekan tombol “ON/OFF” yang kemudian sistem akan memeriksa apakah aplikasi aktif, jika aktif maka aplikasi akan dinonaktifkan, jika tidak aktif maka aplikasi akan diaktifkan. *Activity Diagram* Mengaktifkan/Menonaktifkan Aplikasi disajikan dalam Gambar 3.15.



Gambar 3.15. *Activity Diagram* Mengaktifkan/Menonaktifkan Aplikasi

n) *Activity Diagram* Memproses SMS

Activity Diagram Memproses SMS dimulai ketika *smartphone* menerima SMS, sistem akan memproses setiap sms yang diterima, jika SMS yang diterima tidak mempunyai format SMS yang sesuai maka SMS tidak akan diproses lebih lanjut, namun jika SMS mempunyai format yang tepat maka sistem akan memeriksa apakah GPS aktif, jika GPS tidak aktif maka pengguna harus mengaktifkan GPS lebih dulu, dan jika GPS aktif atau telah aktif maka sistem akan memperbarui lokasi dan membalas SMS secara otomatis dengan mengirimkan lokasi terbaru. *Activity Diagram* Memproses SMS disajikan dalam Gambar 3.16.

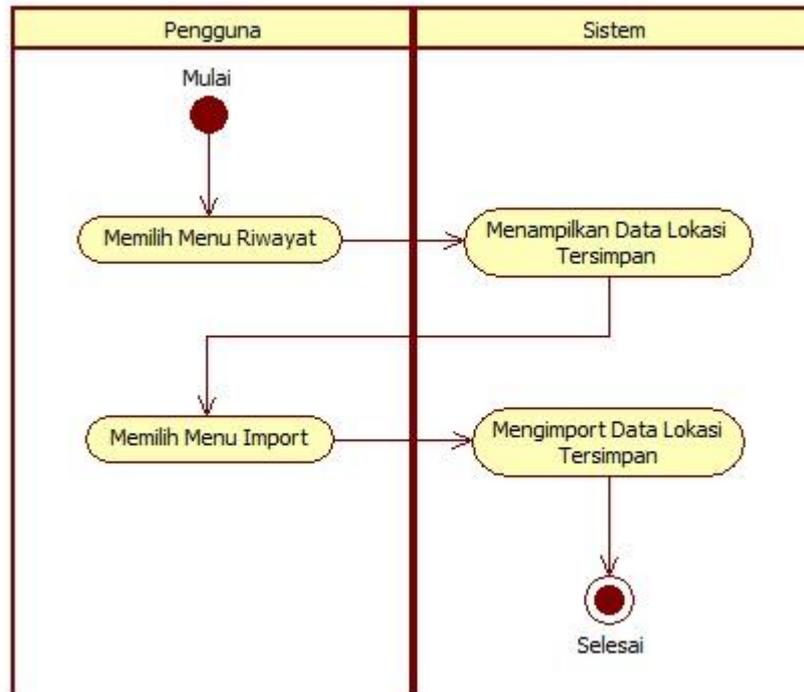


Gambar 3.16. *Activity Diagram* Memproses SMS

o) *Activity Diagram* Mengimport Data Lokasi Tersimpan

Activity Diagram Meng-import Data Lokasi Tersimpan dimulai dengan pengguna memilih menu “Riwayat” yang kemudian sistem akan menampilkan data lokasi tersimpan. Selanjutnya pengguna harus menekan tombol *Import* untuk mengimport data lokasi tersimpan. Data yang dapat import haruslah berupa *file* yang

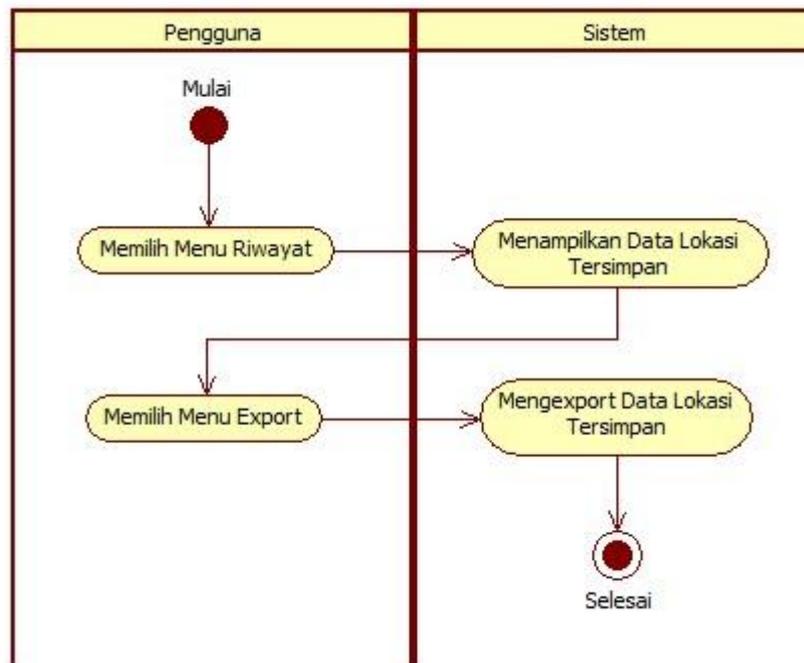
berkekestensi “.csv”. *Activity Diagram* Mengimport Data Lokasi Tersimpan disajikan dalam Gambar 3.17.



Gambar 3.17. *Activity Diagram* Mengimport Data Lokasi Tersimpan

p) *Activity Diagram* Mengexport Data Lokasi Tersimpan

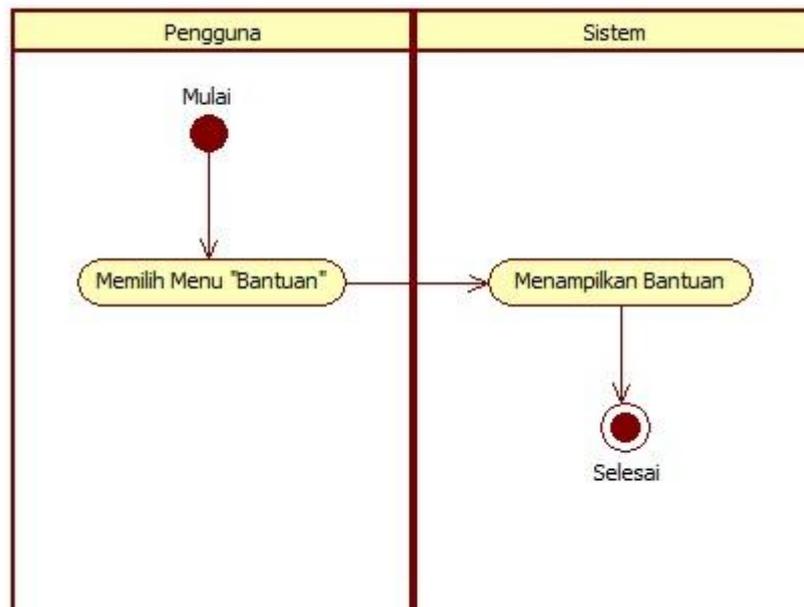
Activity Diagram Mengimport Data Lokasi Tersimpan dimulai dengan pengguna memilih menu “Riwayat” yang kemudian sistem akan menampilkan data lokasi tersimpan. Selanjutnya pengguna harus menekan tombol *Import* untuk meng-*export* data lokasi tersimpan. Hasil *export* dari data lokasi tersimpan ini berupa file yang berkekestensi “.csv”. *Activity Diagram* Meng-*export* Data Lokasi Tersimpan disajikan dalam Gambar 3.18.



Gambar 3.18 *Activity Diagram* Mengexport Data Lokasi Tersimpan

q) *Activity Diagram* Bantuan

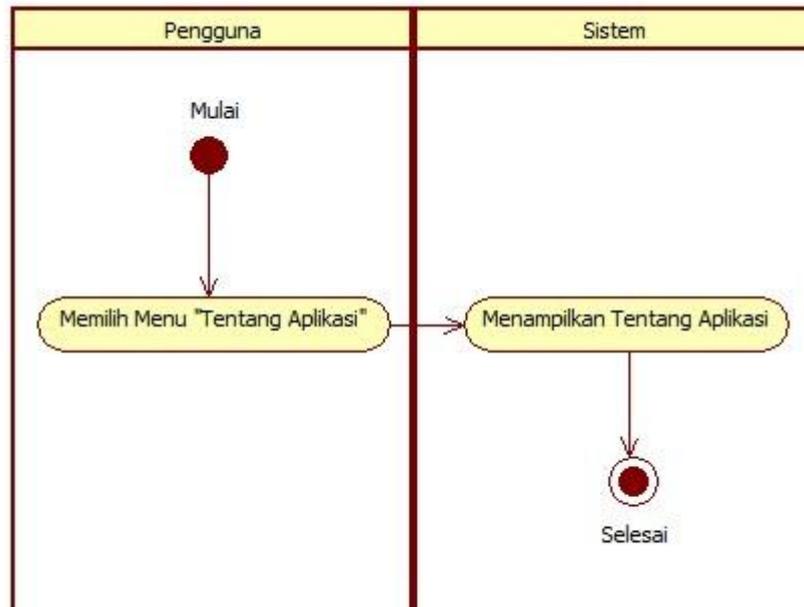
Activity Diagram Bantuan dimulai dengan pengguna memilih menu “Bantuan” yang kemudian sistem akan menampilkan bantuan mengenai penggunaan aplikasi *ShareLoc* ini. *Activity Diagram* Bantuan disajikan dalam Gambar 3.19.



Gambar 3.19. *Activity Diagram* Bantuan

r) *Activity Diagram* Tentang Aplikasi

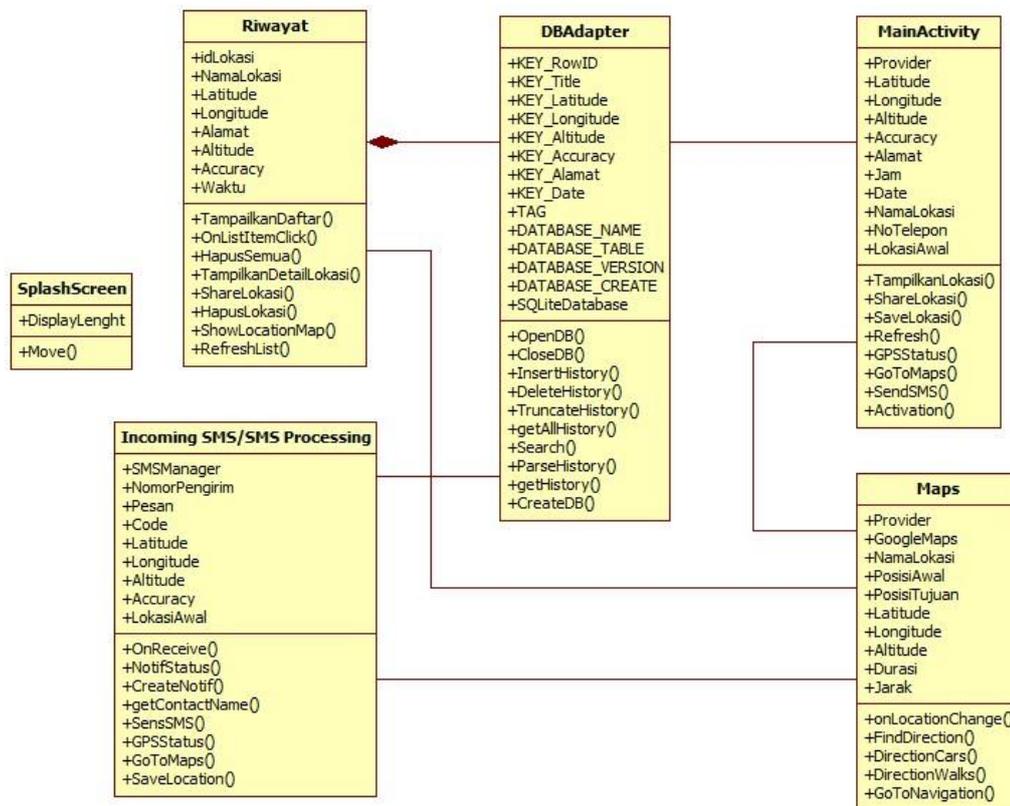
Menu “Tentang Aplikasi” memberikan informasi yang berkaitan dengan aplikasi *ShareLoc*, yaitu untuk mengetahui tujuan dibuatnya aplikasi. *Activity diagram* menu “Tentang Aplikasi” disajikan pada Gambar 3.20.



Gambar 3.20. *Activity Diagram* Tentang Aplikasi

3. *Class Diagram*

Class Diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain. *Class diagram* pada aplikasi *ShareLoc* yaitu sebagai berikut.



Gambar 3.21. Activity Diagram Tentang Aplikasi

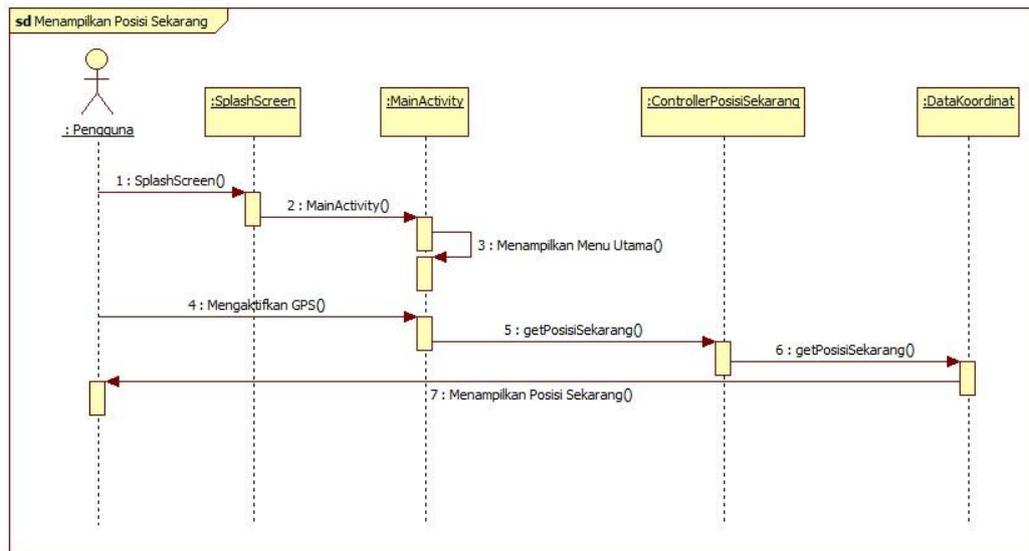
4. Sequence Diagram

Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Pada aplikasi *ShareLoc* terdapat 18 *sequence diagram*, yaitu sebagai berikut.

a) Sequence Diagram Menampilkan Posisi Sekarang

Untuk dapat “Menampilkan Posisi Sekarang” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul halaman utama aplikasi, kemudian sistem akan memeriksa apakah GPS aktif, jika tidak maka pengguna harus mengaktifkan GPS terlebih dahulu,

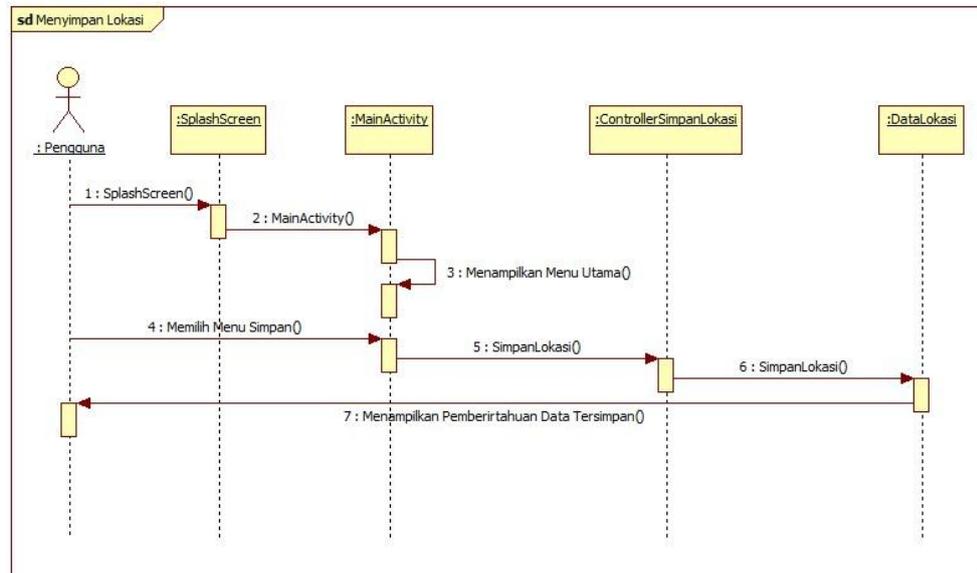
jika aktif atau telah diaktifkan maka sistem akan menampilkan koordinat serta detail lokasi pada halaman utama aplikasi. Untuk lebih jelasnya *sequence diagram* menu “Menampilkan Posisi Sekarang” disajikan pada Gambar 3.22.



Gambar 3.22. *Sequence Diagram* Menampilkan Posisi Sekarang

b) *Sequence Diagram* Menyimpan Lokasi

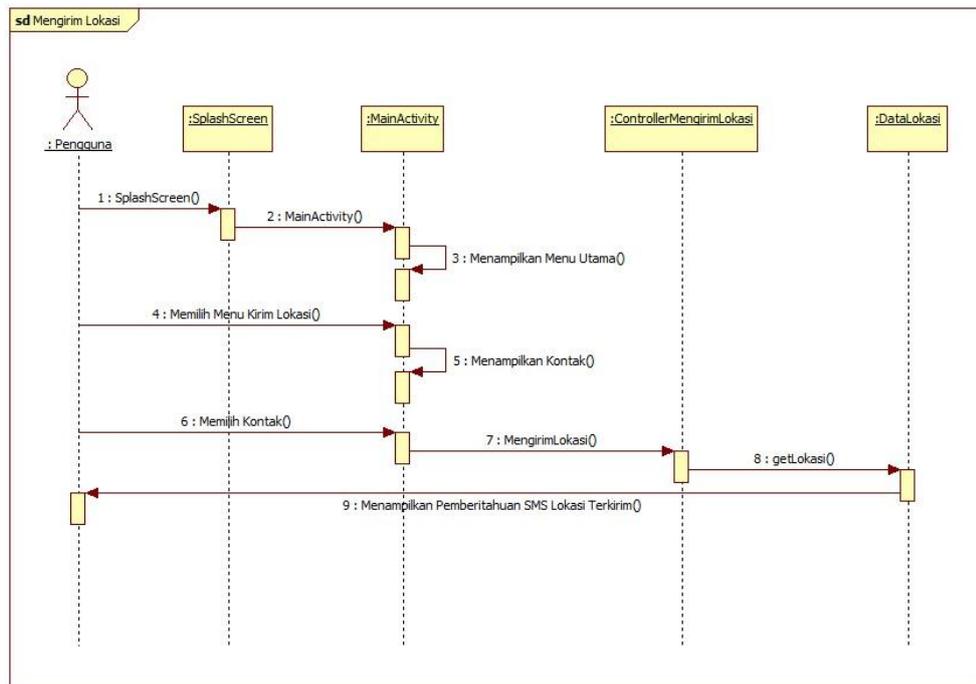
Untuk dapat memilih menu “Menyimpan Lokasi” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Simpan Lokasi”. Setelah pengguna memilih menu “Simpan Lokasi” sistem akan menyimpan lokasi kedalam *database* kemudian menampilkan pemberitahuan status data yang disimpan, berhasil atau gagal. Untuk lebih jelasnya *sequence diagram* menu “Menyimpan Lokasi” disajikan pada Gambar 3.23.



Gambar 3.23. *Sequence Diagram* Menyimpan Lokasi

c) *Sequence Diagram* Mengirim Lokasi

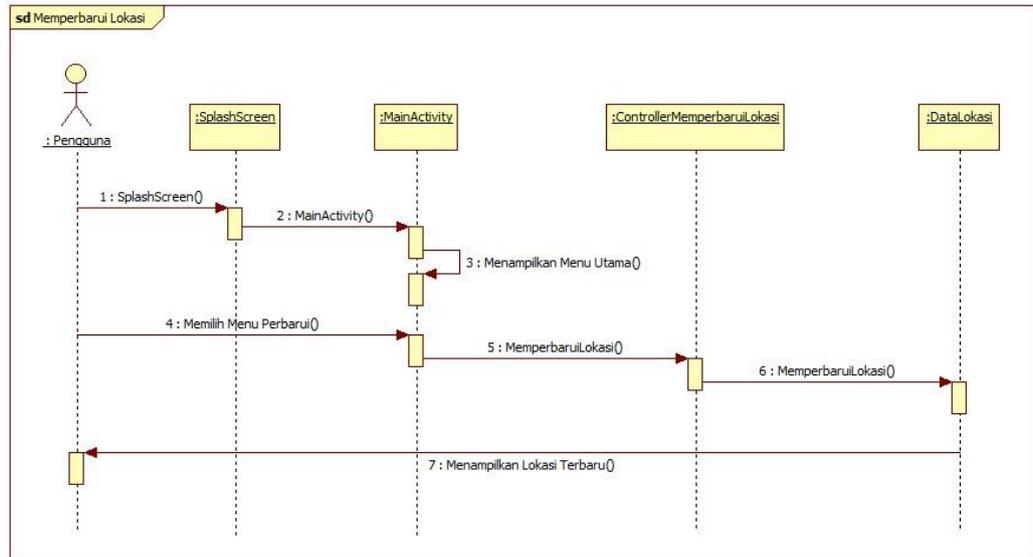
Untuk dapat memilih menu “Mengirim Lokasi” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Mengirim Lokasi”. Setelah pengguna memilih menu “Mengirim Lokasi”, sistem akan menampilkan daftar kontak pada *smartphone*, lalu pengguna dapat memilih kontak dan menekan tombol kirim, kemudian sistem akan mengirimkan lokasi kepada kontak yang telah dipilih dan setelah itu sistem akan menampilkan status pengiriman lokasi, berhasil atau gagal. Untuk lebih jelasnya *sequence diagram* menu “Mengirim Lokasi” disajikan pada Gambar 3.24.



Gambar 3.24. *Sequence Diagram* Mengirim Lokasi

d) *Sequence Diagram* Memperbarui Lokasi

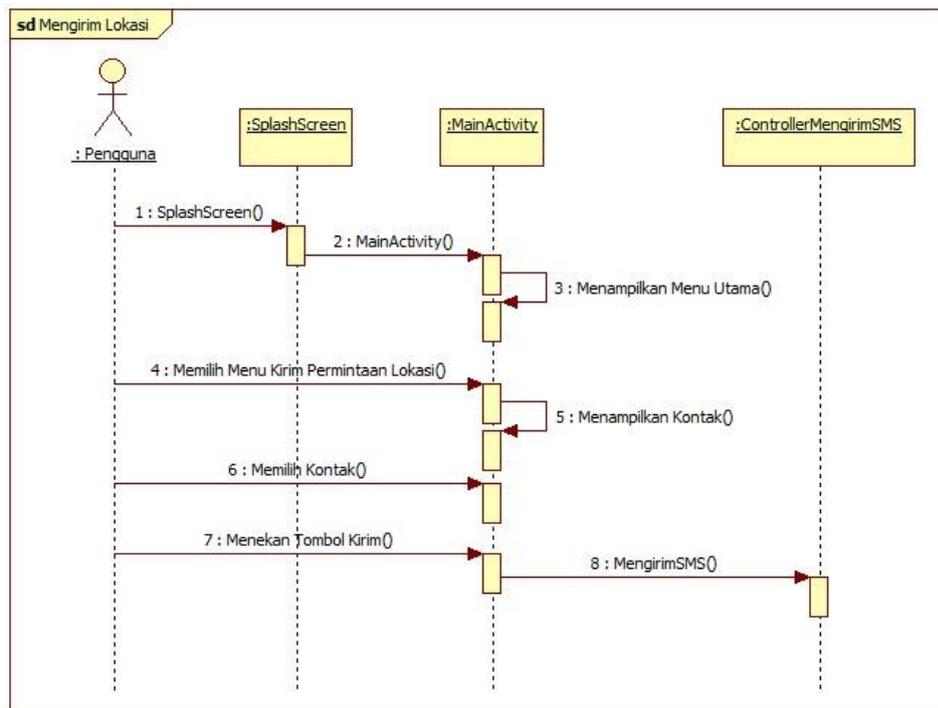
Untuk dapat memilih menu “Memperbarui Lokasi” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, dan pengguna dapat memilih menu “Perbarui Lokasi”, maka sistem akan memperbarui lokasi dengan GPS dan menampilkannya pada halaman utama aplikasi. *Sequence diagram* menu “Perbarui Lokasi” disajikan pada Gambar 3.25.



Gambar 3.25. *Sequence Diagram* Perbarui Lokasi

e) *Sequence Diagram* Mengirim SMS

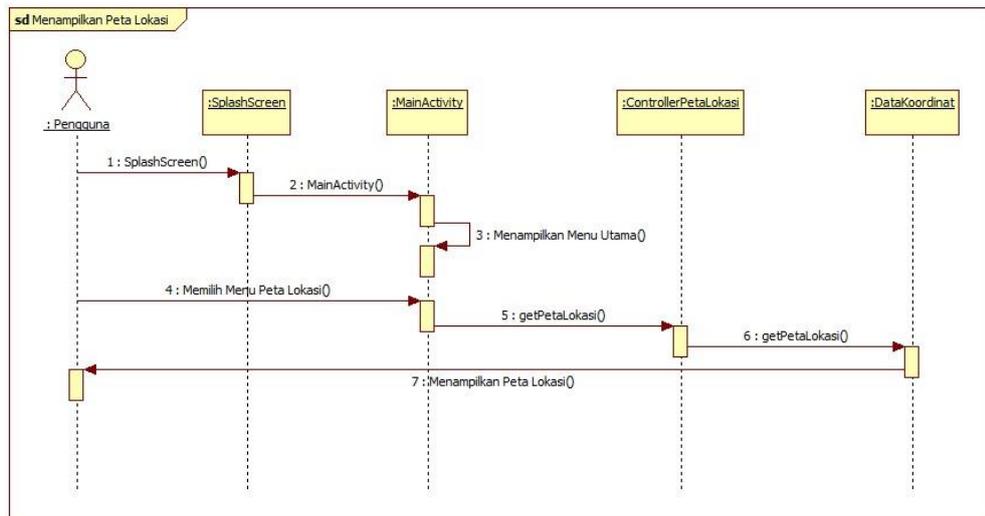
Untuk dapat memilih menu “Mengirim SMS” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Mengirim SMS”. Setelah pengguna memilih menu “Mengirim SMS”, sistem akan menampilkan daftar kontak pada *smartphone*, lalu pengguna dapat memilih kontak dan menekan tombol kirim, kemudian sistem akan mengirimkan lokasi kepada kontak yang telah dipilih dan setelah itu sistem akan menampilkan status pengiriman lokasi, berhasil atau gagal. Untuk lebih jelasnya *sequence diagram* menu “Mengirim SMS” disajikan pada Gambar 3.26.



Gambar 3.26. *Sequence Diagram* Mengirim SMS

f) *Sequence Diagram* Peta Lokasi

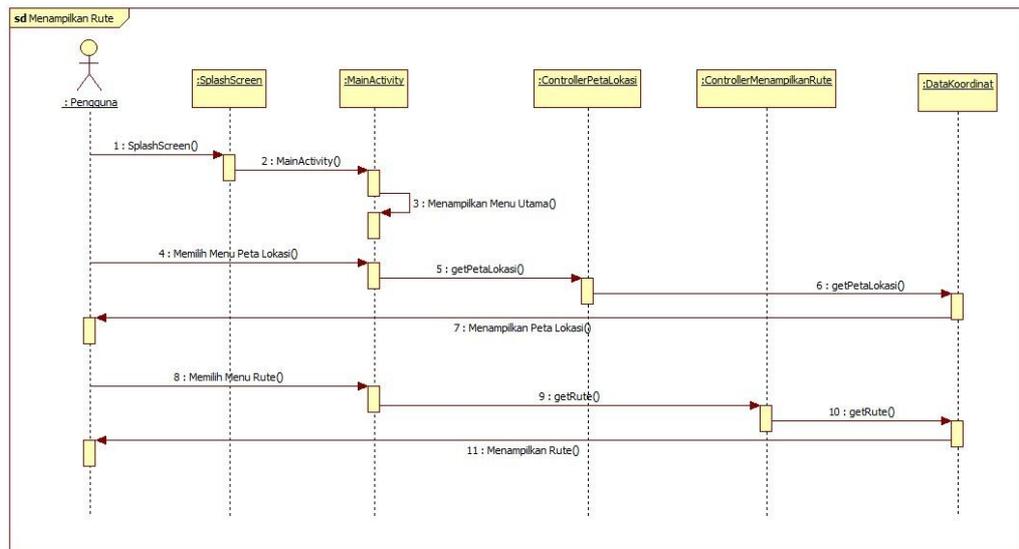
Untuk dapat memilih menu “Peta Lokasi” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Peta Lokasi”. Setelah pengguna memilih menu “Peta Lokasi”, sistem akan menampilkan “Posisi Sekarang” dalam bentuk Peta. Untuk lebih jelasnya *sequence diagram* menu “Peta Lokasi” disajikan pada Gambar 3.27.



Gambar 3.27. *Sequence Diagram* Peta Lokasi

g) *Sequence Diagram* Menampilkan Rute

Untuk dapat memilih menu “Menampilkan Rute” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Peta Lokasi” pada halaman utama atau dapat melalui “Detail Data Tersimpan” dan juga melalui lokasi yang diterima melalui SMS. Setelah masuk pada halaman “Peta Lokasi”, sistem akan menampilkan “Posisi Sekarang” atau “Lokasi Tersimpan” atau lokasi yang diterima melalui SMS dalam bentuk Peta. Kemudian Lokasi tersebut akan dijadikan tujuan dan pengguna dapat memilih menu rute sehingga sistem akan menampilkan rute ke tujuan. Untuk lebih jelasnya *sequence diagram* menu “Menampilkan Rute” disajikan pada Gambar 3.28.

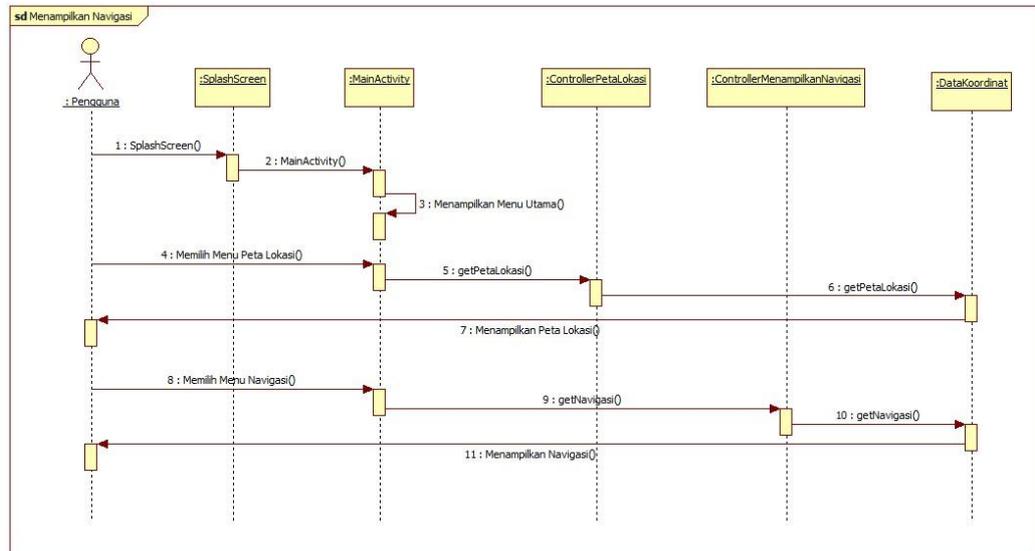


Gambar 3.28. *Sequence Diagram* Menampilkan Rute

h) *Sequence Diagram* Menampilkan Navigasi

Untuk dapat memilih menu “Menampilkan Navigasi” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Peta Lokasi” pada halaman utama atau dapat melalui “Detail Data Tersimpan” dan juga melalui lokasi yang diterima melalui SMS. Setelah masuk pada halaman “Peta Lokasi”, sistem akan menampilkan “Posisi Sekarang” atau “Lokasi Tersimpan” atau lokasi yang diterima melalui SMS dalam bentuk Peta. Kemudian Lokasi tersebut akan dijadikan tujuan dan pengguna dapat memilih menu navigasi sehingga sistem akan membuka aplikasi Navigation yang dikembangkan oleh Google untuk menampilkan rute *step by step* ke tujuan. Untuk lebih jelasnya

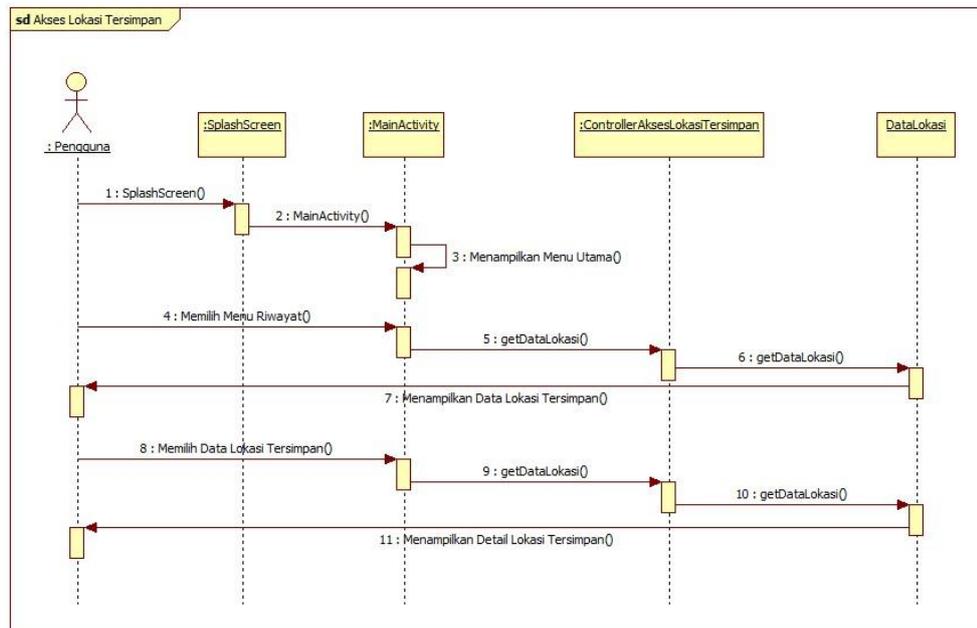
sequence diagram menu “Menampilkan Navigasi” disajikan pada Gambar 3.29.



Gambar 3.29. *Sequence Diagram* M enampilkan Navigasi

i) *Sequence Diagram* Akses Lokasi Tersimpan

Untuk dapat memilih menu “Akses Lokasi” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Riwayat” pada halaman utama. Setelah masuk pada halaman “Riwayat”, sistem akan menampilkan daftar “Lokasi Tersimpan” dan pengguna dapat memilih lokasi tersimpan sehingga sistem akan menampilkan detail dari lokasi tersimpan tersebut. Untuk lebih jelasnya *sequence diagram* menu “Akses Lokasi Tersimpan” disajikan pada Gambar 3.30.

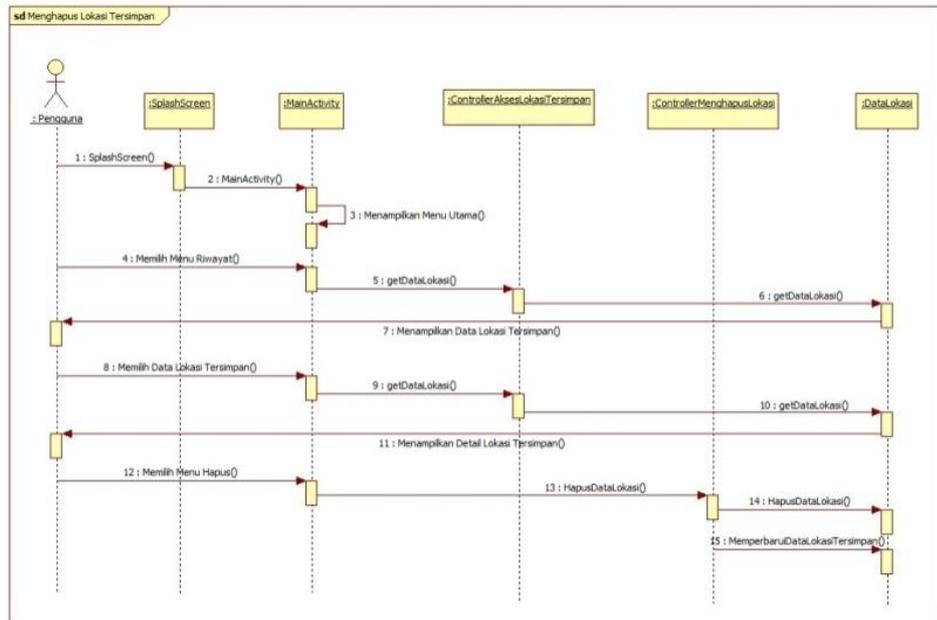


Gambar 3.30. *Sequence Diagram* Akses Lokasi Tersimpan

j) *Sequence Diagram* Menghapus Lokasi Tersimpan

Untuk dapat memilih menu “Menghapus Lokasi Tersimpan” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Riwayat” pada halaman utama. Setelah masuk pada halaman “Riwayat”, sistem akan menampilkan daftar “Lokasi Tersimpan” dan pengguna dapat memilih lokasi tersimpan sehingga sistem akan menampilkan detail dari lokasi tersimpan tersebut. Setelah detail lokasi tersimpan ditampilkan, maka pengguna dapat memilih menu “Hapus” sehingga sistem akan menghapus data tersimpan dan memperbarui daftar data tersimpan.

Untuk lebih jelasnya *sequence diagram* menu “Menghapus Lokasi Tersimpan” disajikan pada Gambar 3.31.

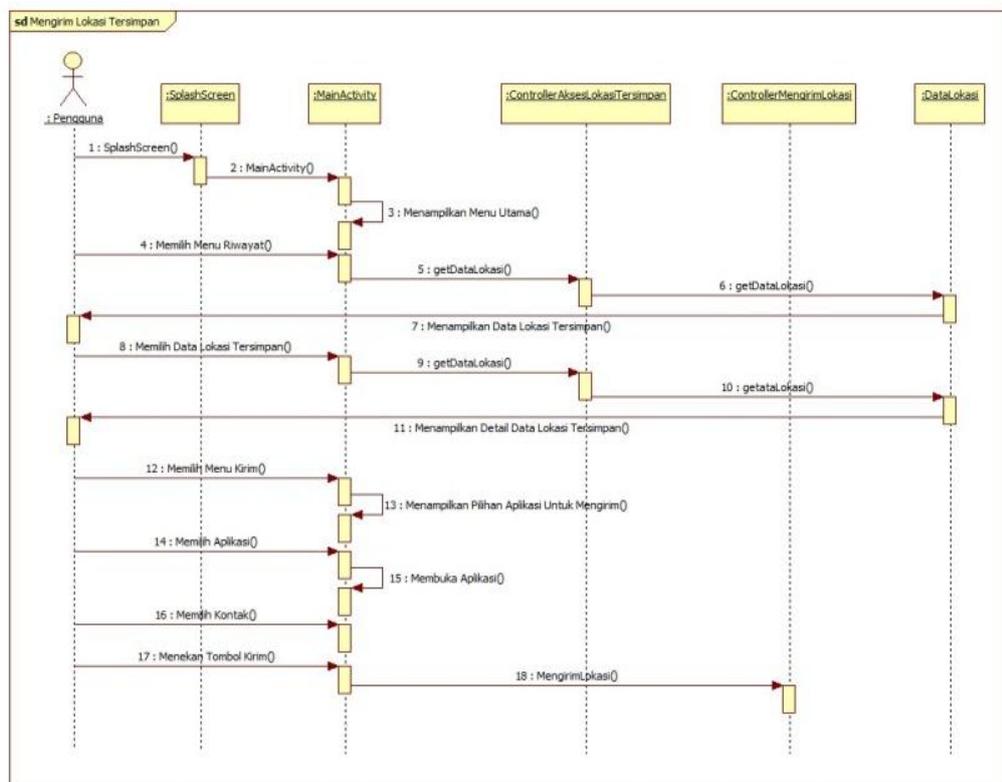


Gambar 3.31. *Sequence Diagram* Menghapus Lokasi Tersimpan

k) *Sequence Diagram* Mengirim Lokasi Tersimpan

Untuk dapat memilih menu “Menghapus Lokasi Tersimpan” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Riwayat” pada halaman utama. Setelah masuk pada halaman “Riwayat”, sistem akan menampilkan daftar “Lokasi Tersimpan” dan pengguna dapat memilih lokasi tersimpan sehingga sistem akan menampilkan detail dari lokasi tersimpan tersebut. Setelah detail lokasi tersimpan ditampilkan, maka pengguna dapat memilih menu “Bagikan Lokasi” sehingga sistem

akan menampilkan daftar aplikasi yang dapat mengirimkan pesan teks, kemudian pengguna memilih aplikasi yang akan digunakan untuk mengirimkan lokasi dan memasukkan kontak yang dituju juga menekan tombol kirim sehingga sistem akan mengirimkan lokasi kepada kontak yang dituju. Untuk lebih jelasnya *sequence diagram* menu “Mengirim Lokasi Tersimpan” disajikan pada Gambar 3.32.

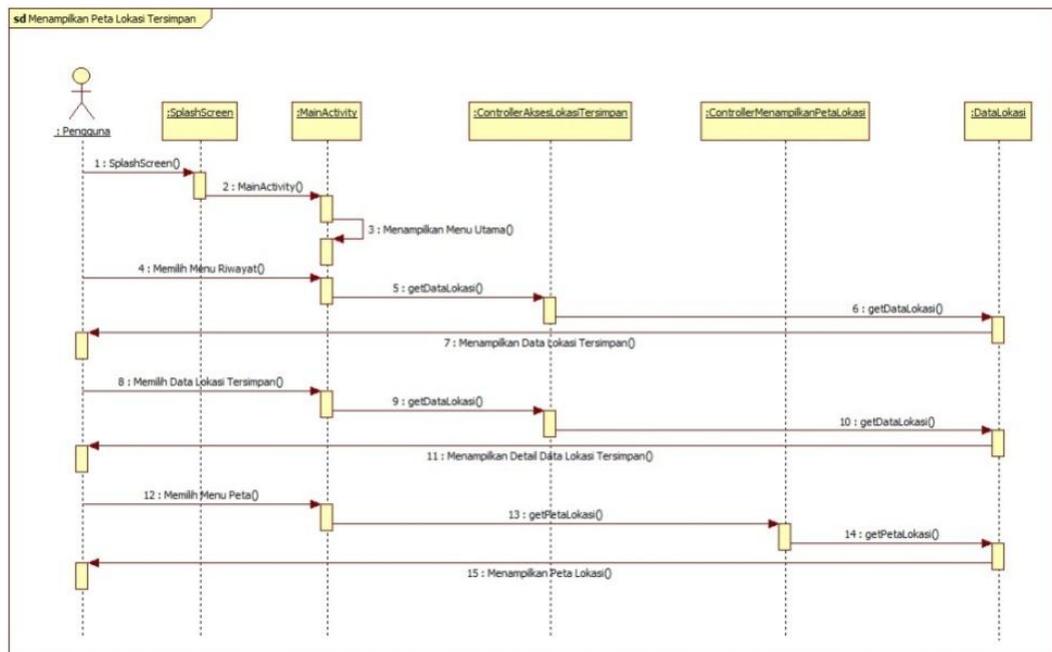


Gambar 3.32. *Sequence Diagram* Mengirim Lokasi Tersimpan

1) *Sequence Diagram* Menampilkan Peta Lokasi Tersimpan

Untuk dapat memilih menu “Menampilkan Peta Lokasi Tersimpan” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya

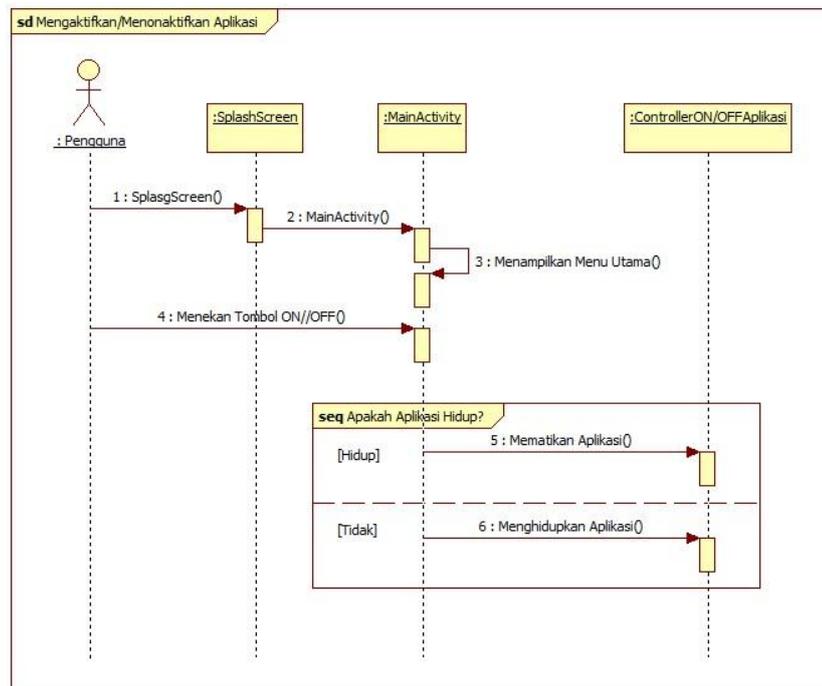
akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Riwayat” pada halaman utama. Setelah masuk pada halaman “Riwayat”, sistem akan menampilkan daftar “Lokasi Tersimpan” dan pengguna dapat memilih lokasi tersimpan sehingga sistem akan menampilkan detail dari lokasi tersimpan tersebut. Setelah detail lokasi tersimpan ditampilkan, maka pengguna dapat memilih menu “Peta Lokasi” sehingga sistem akan menampilkan lokasi tersimpan kedalam bentuk peta. Untuk lebih jelasnya *sequence diagram* menu “Menampilkan Peta Lokasi Tersimpan” disajikan pada Gambar 3.33.



Gambar 3.33. *Sequence Diagram* Menampilkan Peta Lokasi Tersimpan

m) *Sequence Diagram* Mengaktifkan/Menonaktifkan Aplikasi

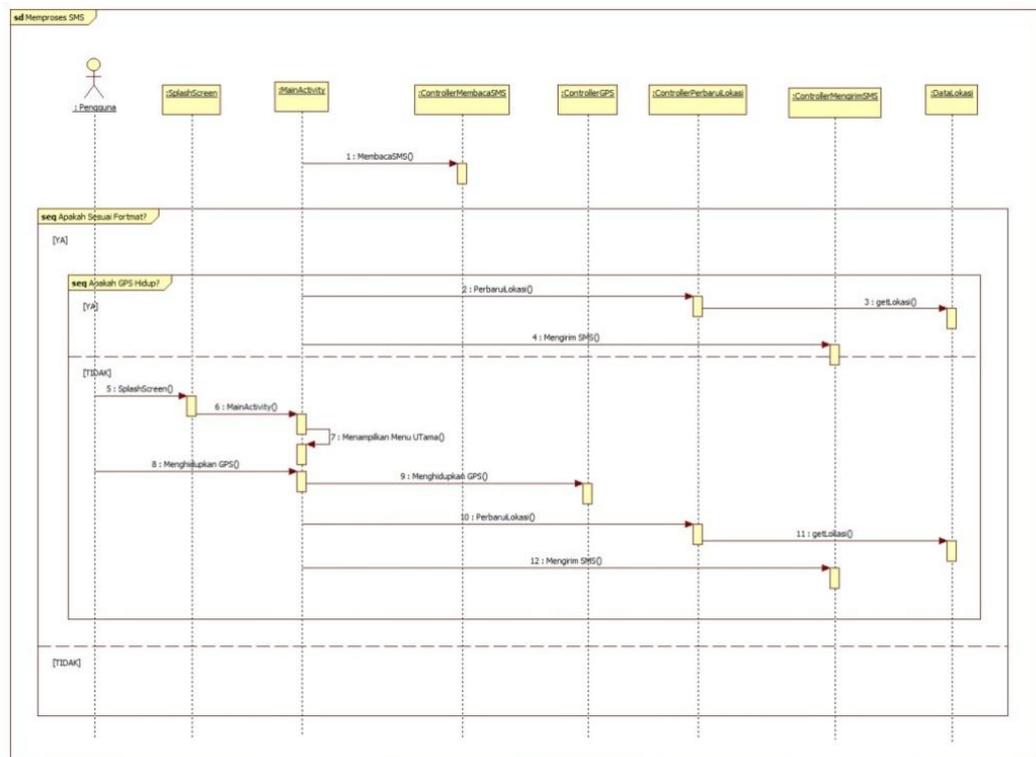
Untuk dapat memilih menu “Mengaktifkan/Menonaktifkan Aplikasi” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Mengaktifkan/Menonaktifkan Aplikasi” pada halaman utama, kemudian sistem akan memeriksa apakah aplikasi aktif, jika tidak aktif maka sistem akan mengaktifkan aplikasi, jika aktif maka sistem akan menonaktifkan aplikasi. Untuk lebih jelasnya *sequence diagram* menu “Mengaktifkan/Menonaktifkan Aplikasi” disajikan pada Gambar 3.34.



Gambar 3.34. *Sequence Diagram* Mengaktifkan/Menonaktifkan Aplikasi

n) *Sequence Diagram Memproses SMS*

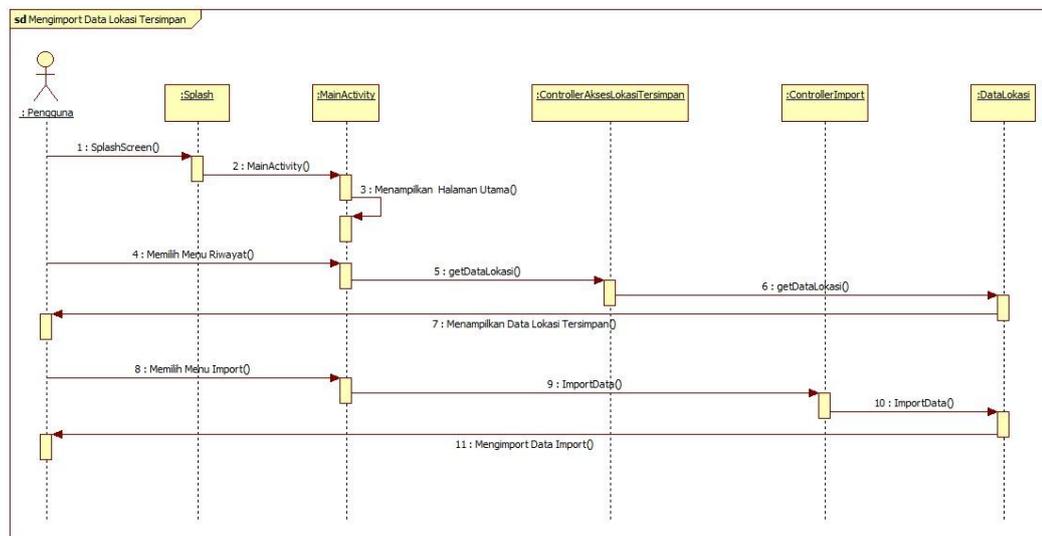
Untuk dapat “Memproses SMS” maka sistem terlebih dahulu membaca SMS yang diterima oleh *smartphone*. Selanjutnya sistem akan memeriksa apakah GPS aktif atau tidak. Jika tidak maka sistem akan menampilkan pengaturan GPS dan pengguna dapat mengaktifkan GPS, dan jika aktif atau telah diaktifkan maka sistem akan memperbarui lokasi pengguna dan membalas sms yang diterima dengan sms yang berisi lokasi terbaru dan format sms sesuai dengan format untuk mengirimkan lokasi. Setelah itu aplikasi akan menampilkan pemberitahuan status pengiriman lokasi, berhasil atau gagal. Untuk lebih jelasnya *sequence diagram* menu “Memproses SMS” disajikan pada Gambar 3.35.



Gambar 3.35. *Sequence Diagram Memproses SMS*

o) *Sequence Diagram* Mengimport Data Lokasi Tersimpan

Untuk dapat mengimport data lokasi tersimpan, maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Riwayat” pada halaman utama. Setelah masuk pada halaman “Riwayat”, sistem akan menampilkan daftar “Lokasi Tersimpan”. Kemudian pengguna harus menekan tombol *Import* selanjutnya aplikasi akan membuka *file manager* untuk mencari letak *file .csv* yang akan di impor. Setelah pengguna menentukan *file* maka aplikasi akan meng-*import* data ke dalam database aplikasi.

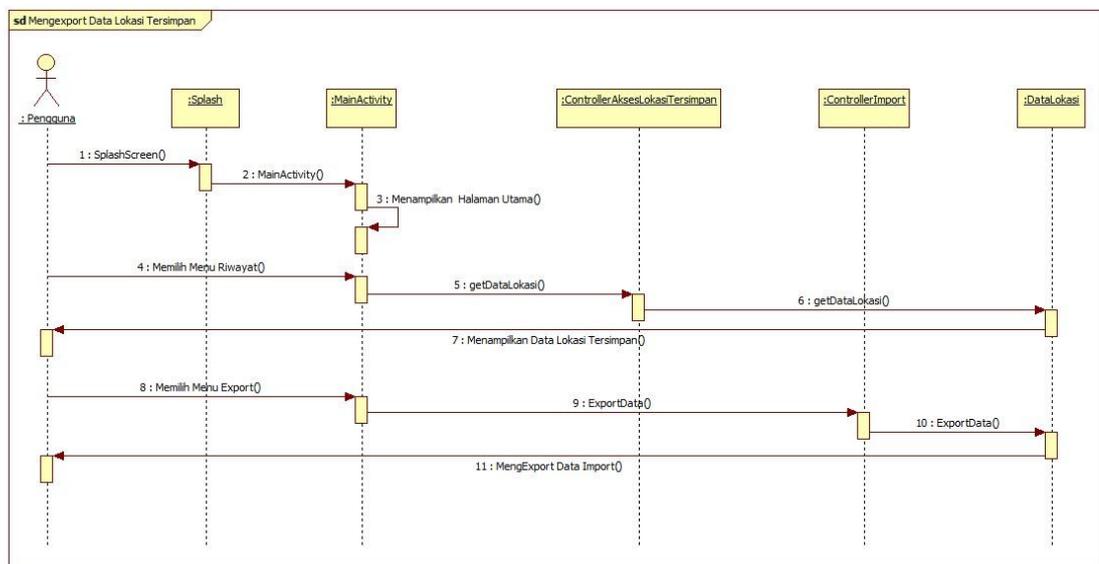


Gambar 3.36 *Sequence Diagram* Mengimport Data Lokasi Tersimpan

p) *Sequence Diagram* Mengexport Data Lokasi Tersimpan

Untuk dapat mengexport data lokasi tersimpan, maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk

ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Riwayat” pada halaman utama. Setelah masuk pada halaman “Riwayat”, sistem akan menampilkan daftar “Lokasi Tersimpan”. Kemudian pengguna harus menekan tombol *Export* selanjutnya aplikasi akan mengekport data lokasi yang tersimpan di database ke dalam file yang berekstensi *.csv*.

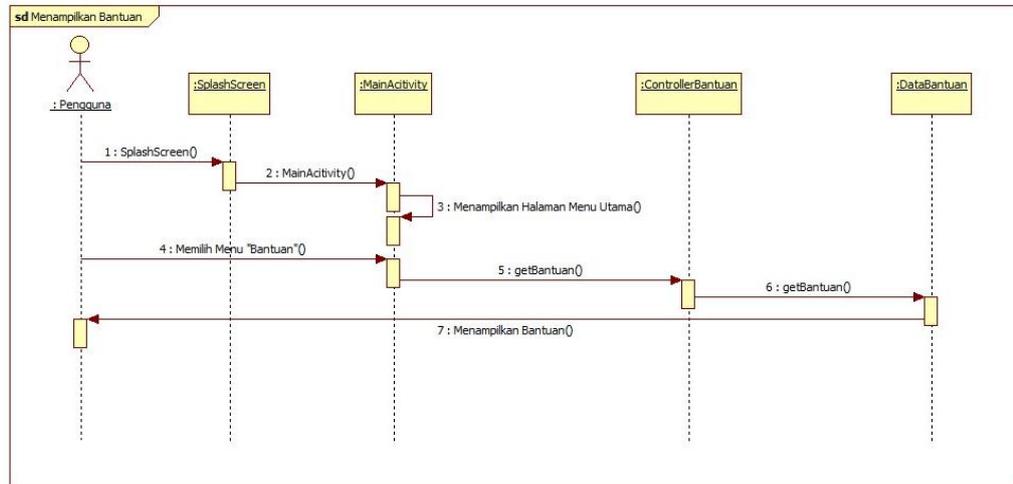


Gambar 3.37 *Sequence Diagram* Mengexport Data Lokasi Tersimpan

q) *Sequence Diagram* Bantuan

Untuk dapat memilih menu “Bantuan” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, dan pengguna dapat memilih menu “Bantuan”, maka sistem akan menampilkan bantuan tentang penggunaan aplikasi

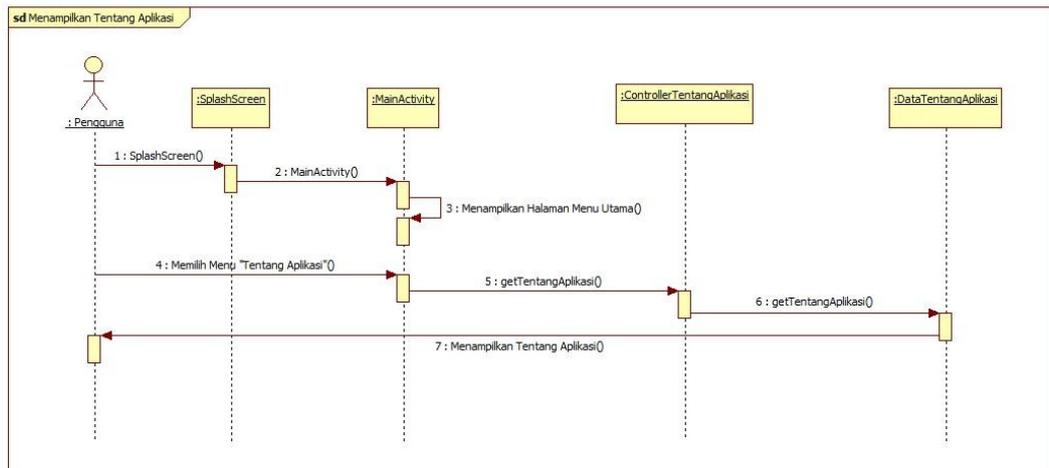
ShareLoc. *Sequence diagram* menu “Bantuan” disajikan pada Gambar 3.38.



Gambar 3.38. *Sequence Diagram* Bantuan

r) *Sequence Diagram* Tentang Aplikasi

Untuk dapat memilih menu “Tentang Aplikasi” maka pengguna terlebih dahulu menjalankan aplikasi *ShareLoc* kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, dan pengguna dapat memilih menu “Tentang Aplikasi”, maka sistem akan menampilkan informasi tentang aplikasi *ShareLoc*. *Sequence diagram* menu “Tentang Aplikasi” disajikan pada Gambar 3.39.



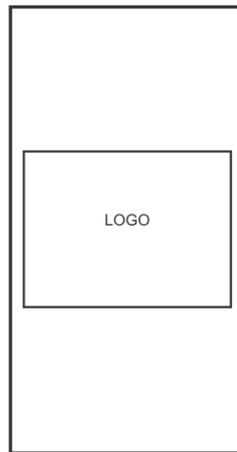
Gambar 3.39. *Sequence Diagram* Tentang Aplikasi

3.3.4.2 Perancangan Antarmuka

Perancangan antarmuka merupakan proses penggambaran bagaimana sebuah tampilan (*interface*) sistem dibentuk. Aplikasi *ShareLoc* dirancang dengan tampilan yang *user friendly*, sehingga diharapkan dapat mempermudah pengguna dalam menggunakan aplikasi ini. Dalam Aplikasi ini terdapat beberapa *layout* atau *form* antara lain.

1. *Layout Splash screen*

Splash Screen adalah adalah tampilan awal ketika pengguna menjalankan aplikasi, *Splash screen* digunakan sebagai identitas aplikasi. Perancangan *layout splash screen* aplikasi *ShareLoc* disajikan pada Gambar 3.40.



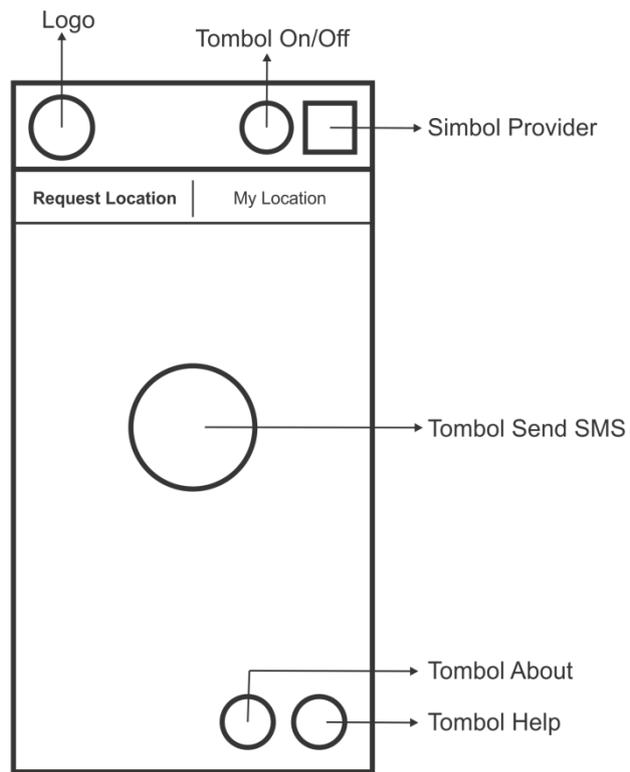
Gambar 3.40. *Design Layout Splash Screen*

2. *Layout Request Location*

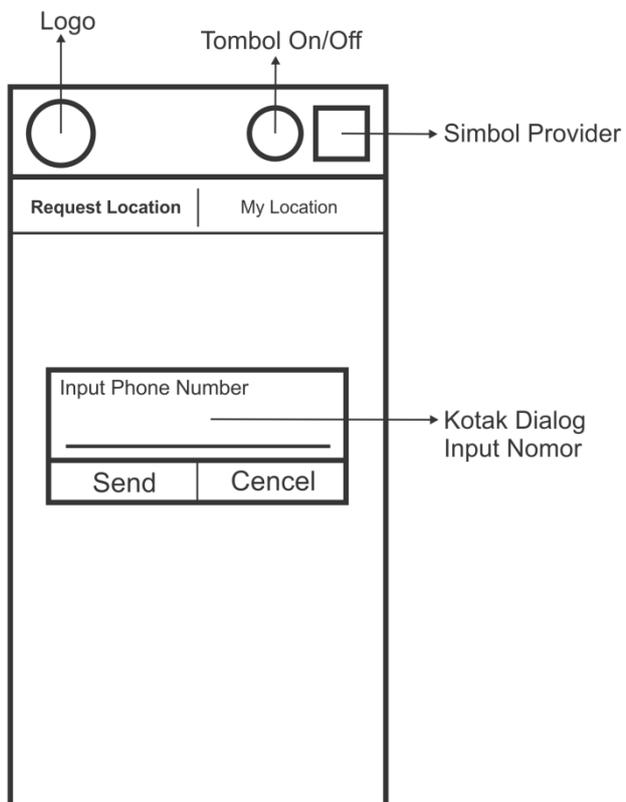
Menu utama pada aplikasi ini terdapat dua halaman yaitu halaman *request location* dan *my location* yang disajikan dengan menggunakan menu tab.

Pada Menu *request location* terdapat tombol “*Send SMS*” yang berguna untuk mengirimkan permintaan lokasi kepada pengguna aplikasi *Shareloc* lain dengan menggunakan SMS. Selain tombol “*Send SMS*”, pada halaman ini juga terdapat tombol “*On/Off*”, tombol “*About*” dan “*Help*”.

Perancangan *layout request location* aplikasi *ShareLoc* disajikan pada Gambar 3.41.



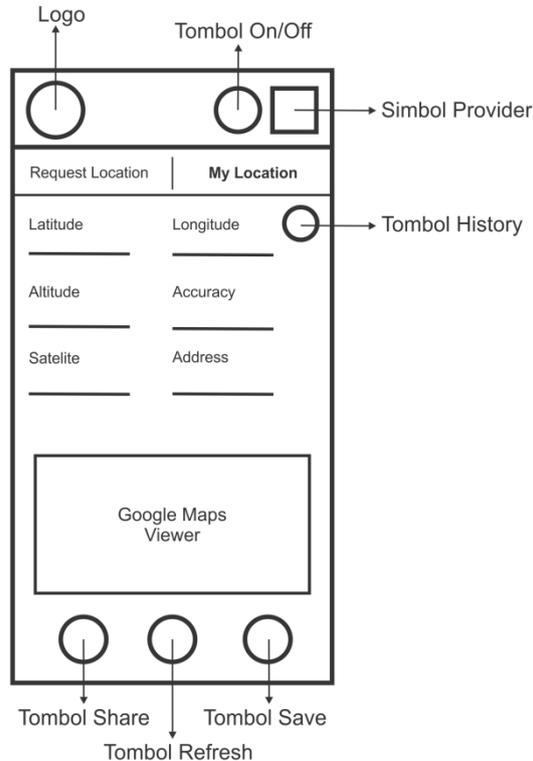
Gambar 3.41. *Design Layout Menu Request Location*



Gambar 3.42. *Design Layout Input Nomor Telepon*

3. *Layout Menu My Location*

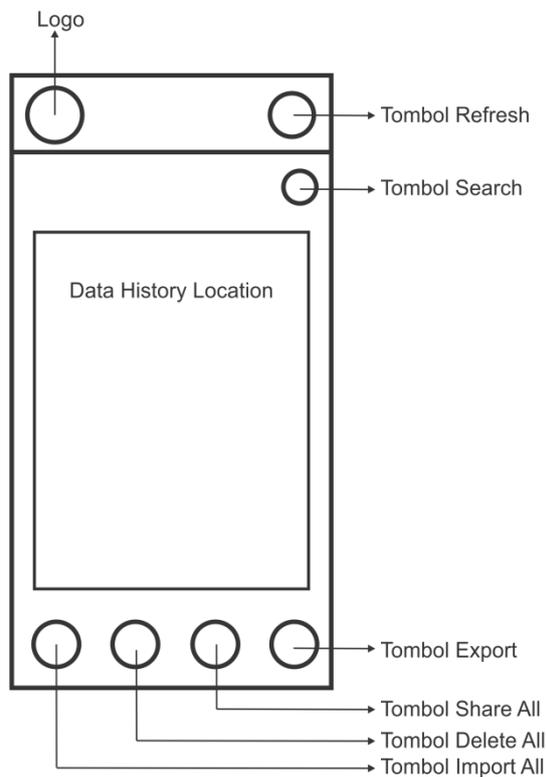
Pada menu ini akan ditampilkan informasi lokasi pengguna. Adapaun informasi yang ditampilkan diantara lain *latitude*, *longitude*, *altitude*, *Accuracy*, *Satelite*, dan *Address*. Selain informasi lokasi pengguna dari beberapa yang disebutkan, pada menu ini juga menampilkan peta lokasi pengguna dengan menggunakan *Google Maps Viewer*. Menu *My Location* juga memiliki tombol "*History*" yang berfungsi untuk menampilkan menu *History*, tombol "*Share*" berfungsi untuk membagikan informasi lokasi pengguna, tombol "*Refresh*" yang berfungsi untuk memperbarui lokasi pengguna serta tombol "*Save*" berfungsi untuk menyimpan informasi lokasi pengguna ke dalam *database* aplikasi. Perancangan *layout* menu *My Location* disajikan pada Gambar 3.43.



Gambar 3.43 *Design Layout Menu My Location*

4. *Layout Menu History*

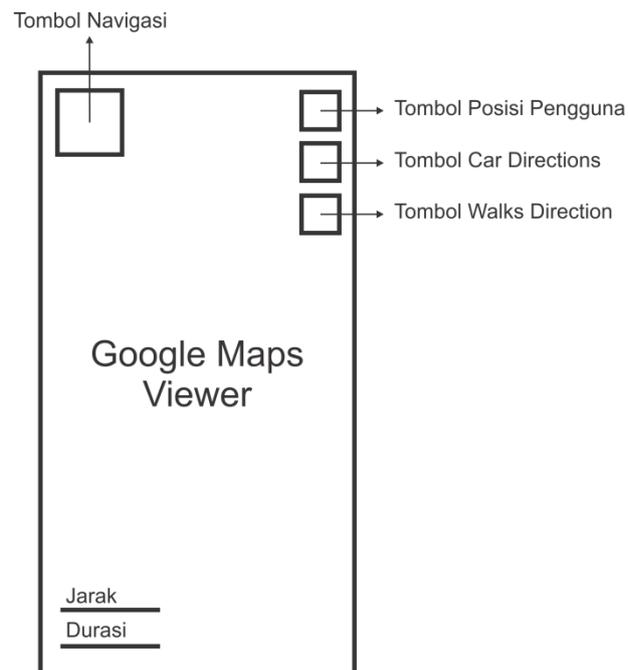
Pada menu ini akan ditampilkan *history* atau data lokasi yang telah disimpan oleh pengguna. *Layout* menu *history* ini memiliki beberapa fungsi, antara lain mencari data berdasarkan tanggal tersimpan, menghapus seluruh data, dan menampilkan detail data lokasi. Tampilan detail data lokasi pun memiliki beberapa fungsi, yaitu menghapus, mengirim, dan menampilkan data lokasi kedalam peta. Perancangan *layout* menu *history* disajikan pada Gambar 3.44.



Gambar 3.44. *Design Layout Layout Menu History*

5. *Layout* menu Maps

Menu Maps pada aplikasi ini digunakan untuk menampilkan data lokasi tersimpan, dan lokasi yang diterima melalui SMS kedalam bentuk peta. Menu Maps pada aplikasi ini memiliki beberapa fungsi yang dapat digunakan oleh pengguna, seperti menampilkan posisi pengguna, menampilkan rute mobil atau pejalan kaki, serta membuka aplikasi navigasi dari Google untuk mendapatkan detail rute *step by step* menuju lokasi tujuan. Peta yang digunakan dalam aplikasi ini adalah Google Maps. dalam Perancangan *layout* menu “Maps” disajikan pada Gambar 3.45.



Gambar 3.45. *Design Layout Layout* MenuMaps

6. *Layout Menu Help*

Menu *Help* berguna untuk memberikan informasi kepada pengguna tentang bagaimana menggunakan aplikasi *ShareLoc*. Perancangan *layout* menu *help* disajikan pada Gambar 3.46.



Gambar 3.46. *Design Layout Layout Menu Help*

7. *Layout menu About.*

Menu *About* berisi tentang manfaat aplikasi *ShareLoc* serta informasi pengembang. Perancangan *layout* menu *about* disajikan pada Gambar 3.47.



Gambar 3.47. *Design Layout Menu About*

3.4 Metode Pengujian Sistem

Metode pengujian sistem dalam penelitian ini adalah pengujian *Black Box* dengan metode *Equivalence Partitioning* (EP). Pengujian ini berguna untuk membuktikan semua fungsi-fungsi pada aplikasi berjalan dengan baik.

Pada pengujian ini diyakinkan bahwa masukan dan respon yang diterima sama sehingga terjadi kecocokan antara aplikasi dan pengguna. Metode ini dipilih karena metode ini dapat mencari kesalahan pada fungsi aplikasi, *interface* aplikasi dan kesalahan pada struktur data aplikasi. Rancangan daftar pengujian disajikan pada Tabel 3.1.

Tabel 3.1 Daftar Pengujian *Equivalence Partitioning* (EP)

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	Versi Android	Pengujian kompatibilitas versi operatif system Android	Pengujian pada Android versi 4.0 (Ice Cream Sandwich)	Kompatibel dengan Android versi 4.0 (Ice Cream Sandwich)
			Pengujian pada Android versi 4.1 (Jelly Bean)	Kompatibel dengan Android versi 4.1 (Jelly Bean)
			Pengujian pada Android versi 4.4 (KitKat)	Kompatibel dengan Android versi 4.4 (KitKat)
			Pengujian pada Android Versi 5.0 (Lollipop)	Kompatibel dengan Android Versi 5.0 (Lollipop)
			Pengujian pada Android Versi 6.0 (Marshmallow)	Kompatibel dengan Android Versi 6.0 (Marshmallow)
2	Resolusi Layar dan Densitas Layar	Pengujian Resolusi Layar dan Densitas Layar pada Android	Pengujian pada Android dengan resolusi 3 inch	Tampilan terlihat sesuai atau baik pada Android dengan resolusi 3 inch
			Pengujian pada Android dengan resolusi 4 inch	Tampilan terlihat sesuai atau baik pada Android dengan resolusi 4 inch
			Pengujian pada Android dengan resolusi 5 inch	Tampilan terlihat sesuai atau baik pada Android dengan resolusi 5 inch
			Pengujian pada Android dengan resolusi 7 inch	Tampilan terlihat sesuai atau baik pada Android dengan resolusi 7 inch

Tabel 3.1 Daftar Pengujian *Equivalence Partitioning* (EP) (Lanjutan)

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
3	<i>User Interface</i>	Pengujian Pada Icon <i>ShareLoc</i>	Klik <i>icon ShareLoc</i> pada perangkat Android pengguna	Menampilkan layout <i>splash screen</i>
		Pengujian Pada Menu Utama <i>ShareLoc</i>	Klik Tombol “On/Off”	Menghidupkan/Mematikan Aplikasi <i>ShareLoc</i>
			Klik tombol “Save”	Menyimpan lokasi posisi sekarang
			Klik tombol “Share”	Menampilkan <i>Dialog</i> pilih aplikasi untuk berbagi lokasi
			Klik Tombol “Option”	Menampilkan menu <i>option</i>
			Klik tombol “Maps”	Menampilkan <i>Layout</i> Peta Lokasi Sekarang pada Google Maps
			Klik tombol “Refresh”	Melakukan pencarian lokasi posisi sekarang
			Klik tombol “Send SMS”	Menampilkan <i>Dialog</i> Pilih Kontak
4	Fungsi Menu <i>Option</i>	Pengujian Menu <i>Option</i>	Klik tombol “History”	Menampilkan <i>Layout</i> Daftar data lokasi lokasi
			Klik tombol “Help”	Menampilkan <i>Layout</i> Bantuan penggunaan aplikasi
			Klik tombol “About”	Menampilkan <i>Dialog</i> Tentang aplikasi <i>ShareLoc</i>
	Fungsi Menu <i>History</i>	Pengujian Pada Menu <i>History</i>	Klik tombol “Rename”	Menampilkan <i>Dialog</i> untuk mengubah nama lokasi tersimpan
			Klik tombol “More”	Menampilkan <i>Dialog</i> Detail Informasi Data Lokasi Tersimpan
			Klik tombol “Option”	Menampilkan Menu <i>Option</i>
			Klik tombol “Share”	Menampilkan <i>Dialog</i> pilih aplikasi untuk berbagi lokasi
			Klik tombol “Delete All”	Menghapus seluruh data lokasi tersimpan

Tabel 3.1 Daftar Pengujian *Equivalence Partitioning* (EP) (Lanjutan)

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
4	Fungsi <i>Layout</i> Detail Data Lokasi Tersimpan	Pengujian pada <i>Layout</i> Detail Data Lokasi Tersimpan	Klik tombol “ <i>Search</i> ”	Menampilkan <i>Dialog</i> pencarian berdasarkan tanggal
			Klik tombol “ <i>Delete</i> ”	Menghapus Data Tersimpan yang sedang dilihat
			Klik tombol “ <i>Share</i> ”	Mengirimkan lokasi Data Tersimpan yang dilihat melalui aplikasi yang dipilih dan kepada kontak yang dipilih pengguna
			Klik tombol “ <i>Maps</i> ”	Menampilkan data lokasi tersimpan kedalam Google Maps Viewer
	Fungsi <i>Layout</i> <i>Maps</i>	Pengujian pada <i>Layout</i> <i>Maps</i>	Klik tombol “Posisi Pengguna”	Tampilan peta fokus pada posisi pengguna
	Fungsi <i>Layout</i> <i>Maps</i>	Pengujian pada <i>Layout</i> <i>Maps</i>	Klik tombol “ <i>Car Directions</i> ”	Menampilkan rute mobil menuju lokasi tujuan serta menampilkan jarak dan estimasi waktu
			Klik tombol “ <i>Walk Directions</i> ”	Menampilkan rute pejalan kaki menuju lokasi tujuan serta menampilkan jarak dan estimasi waktu
			Klik tombol “ <i>Navigasi</i> ”	Membuka aplikasi <i>Navigation</i> untuk menampilkan rute <i>step by step</i> menuju lokasi tujuan
5	Koneksi Internet dan GPS	Pengujian pada koneksi internet dan GPS saat Melakukan pencarian rute	Koneksi internet stabil dan GPS aktif	Aplikasi berjalan dengan lancar saat sedang melakukan pencarian rute

Tabel 3.1 Daftar Pengujian *Equivalence Partitioning* (EP) (Lanjutan)

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
5	Koneksi Internet dan GPS	Pengujian pada koneksi internet dan GPS saat Melakukan pencarian rute	Koneksi internet tidak stabil dan GPS aktif	Aplikasi berjalan dengan lancar saat sedang melakukan pencarian rute
			Koneksi internet stabil dan GPS tidak aktif	Aplikasi berjalan dengan lancar saat sedang melakukan pencarian rute
			Koneksi internet tidak stabil dan GPS tidak aktif	Aplikasi berjalan dengan lancar saat sedang melakukan pencarian rute

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Dari hasil penelitian yang dilakukan, penulis dapat mengambil simpulan sebagai berikut.

1. Aplikasi *ShareLoc* telah berhasil dibangun, yang dapat menyimpan dan berbagi lokasi menggunakan *Short Message Service* (SMS) sebagai bantuan untuk menyimpan catatan lokasi yang telah dikunjungi dan membantu menemukan lokasi pengguna lain berbasis Android.
2. Teknologi GIS telah berhasil diterapkan untuk menentukan letak pengguna dan menampilkan peta pada Google Maps.
3. Dari hasil pengujian maka dapat disimpulkan semua fungsi pada aplikasi dapat dijalankan dengan baik.

5.2 Saran

1. Aplikasi ini nantinya dapat dikembangkan sehingga nantinya memiliki fitur grup akses yang membatasi pengguna lain untuk mengetahui informasi lokasi *user*.
2. Aplikasi ini nantinya dapat dikembangkan sehingga kompatibel pada *platform* selain Android, seperti iOS, BlackBerry OS ataupun Windows Phone.

DAFTAR PUSTAKA

- Andry. 2011. *Android A sampai Z*. Pcplus: Jakarta.
- Clune, T.L., R.B. Rood. 2011. *Software Testing and Verification In ClimateModel Development*. *IEEE Journal, Focus: Climate Change Software*. September-October, pp. 49-55.
- Developers, Android. 2014. *Android Developers*. [Online]. Tersedia: <http://developer.android.com/index.html>. Diakses pada tanggal 20 November 2015.
- Eclipse. 2014. *About the Eclipse Foundation*. [Online]. Tersedia: <https://www.eclipse.org/org/>. Diakses pada tanggal 20 November 2015.
- Fowler, Martin. 2004. *UML Distilled Panduan Singkat Bahasa pemodelan Objek Standar, Edisi 3*. Andi Publishing, Yogyakarta.
- Halim, J I., et al. 2011. *Framework Pemetaan Data Berbasis Peta dengan Menggunakan Google Maps API (Skripsi)*. Universitas Bina Nusantara. Jakarta.
- Jiang, F., Y. Lu. 2012. *Software Testing Model Selection Research based on Yin-Yang Testing Theory*. In: *IEEE Proceeding of International Conference on Computer Science and Information Processing (CISP)*, pp. 590-594.
- Meier, Reto. 2009. *Professional Android Application Development*. Wiley Publishing, Inc: Canada.

- Nugroho, Adi. 2015. *Analisis dan Perancangan Sistem Informasi Dengan Metodologi Berorientasi Objek*. Bandung: Informatika.
- Prahasta, Eddy, 2014, *Sistem Informasi Geografis Konsep – Konsep Dasar (Prespektif Geodesi & Geomatika)*. Bandung: Informatika Bandung.
- Pressman, Roger S. 2001. *Software Engineering A Practitioner’s Approach Fifth Edition*. McGraw-Hill Companies, Inc, New York.
- Pressman, R.S. 2010. *Software Engineering: A Practitioner’s Approach, 7th Edition*. McGraw-Hill: New York.
- Roger, Rick, John Lombardo, et al. 2009. *Android Application Development*. O’Reilly Media: USA.
- Safaat, Nazruddin. 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android Edisi Revisi*. Informatika: Bandung.
- Steiniger, S., Neun, M., dan Edwardes, A. 2006. *Foundations of Location Based Services, Lecture notes*. [Online] Tersedia:http://www.spatial.cs.umn.edu/Courses/Fall11/8715/papers/IM7_steiniger.pdf. Diakses pada 20 November 2015.
- Swastikayana, I Wayan Eka. 2011. *Sistem Informasi Geografis Berbasis Web Untuk Pemetaan Pariwisata Kabupaten Gianyar*. Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Pembangunan Nasional”Veteran”. Yogyakarta.