



Pengembangan *Augmented Reality Mobile Navigation*
Universitas Jember Dengan Pendekatan Skenario

SKRIPSI

Oleh
Ahmad Syafiq Kamil
NIM 102410101086

PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER

2015



Pengembangan *Augmented Reality Mobile Navigation*
Universitas Jember Dengan Pendekatan Skenario.

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan pendidikan di Program Studi Sistem Informasi Universitas Jember dan mendapat gelar Sarjana Sistem Informasi

Oleh

Ahmad Syafiq Kamil

NIM 102410101086

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

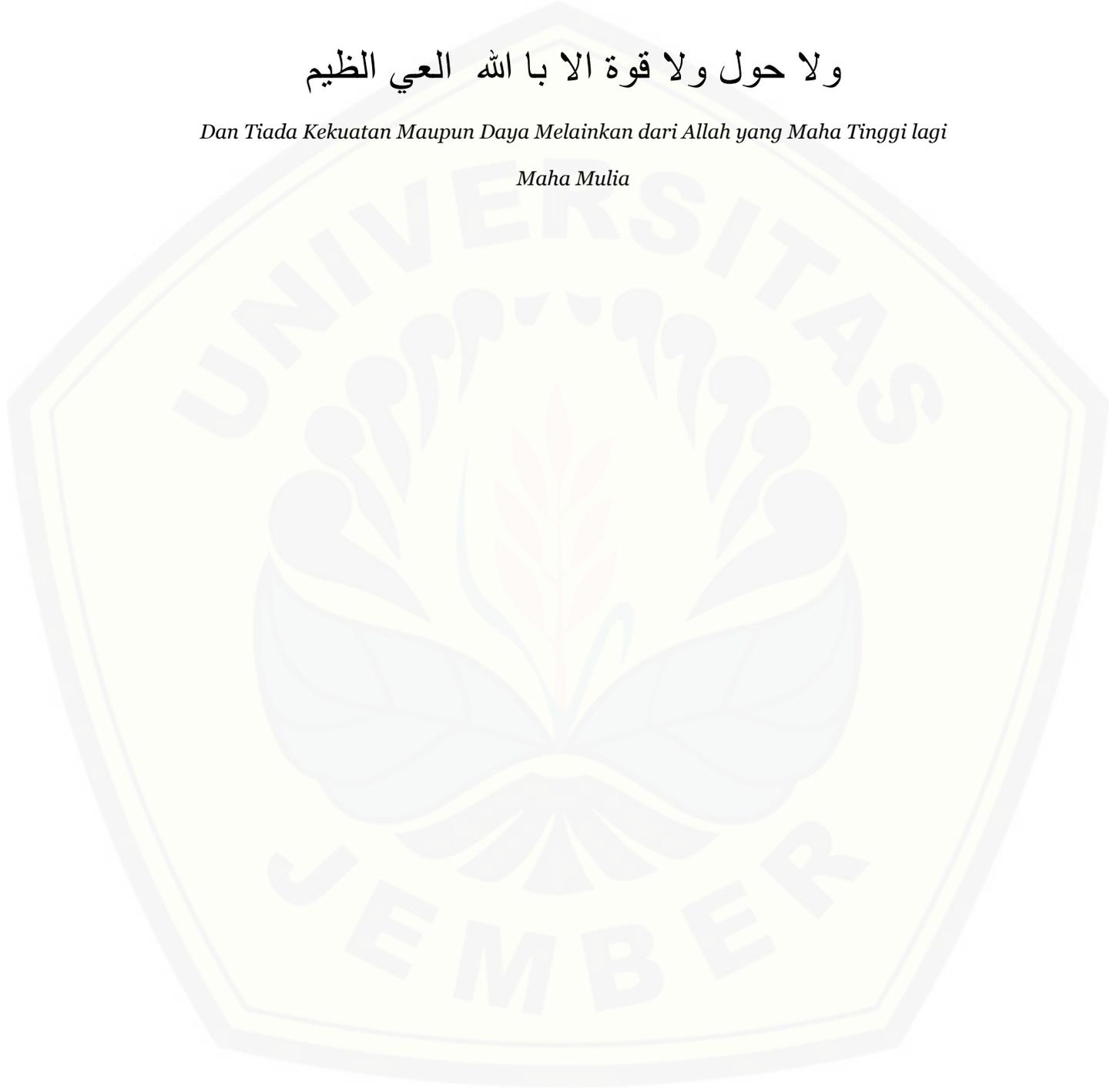
1. Ucapan syukur tiada terkira kepada Allah SWT yang maha pengasih lagi maha penyayang. Sholawat serta salam selalu terucap kepada Nabi Besar Muhammad SAW semoga syafaatnya selalu terlimpah untuk kita.
2. Teruntuk Abi Mashudi dan Umi Robiah, tiada kata-kata yang cukup untuk mengungkapkan rasa cintaku pada beliau.
3. Teruntuk Muhammad Reza Wardana, Zidna Mazidah dan Erda Afifah yang selalu menjadi motivasi sendiri bagi saya serta almarhum Muhammad Sirojul Umam yang telah lebih dulu meninggalkan keluarga, semoga menjadi penyelamat keluarga di surga kelak.
4. Teruntuk Nayirotul Faiqoh, terima kasih telah menemani selama ini dan semoga untuk selamanya.

MOTTO

ولا حول ولا قوة الا بالله العلي العظيم

Dan Tiada Kekuatan Maupun Daya Melainkan dari Allah yang Maha Tinggi lagi

Maha Mulia



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ahmad Syafiq Kamil

NIM : 102410101086

menyatakan sesungguhnya bahwa karya ilmiah yang berjudul “Pengembangan *Augmented Reality Mobile Navigation* Universitas Jember Dengan Pendekatan Skenario” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isisnya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 2014

Yang menyatakan,

Ahmad Syafiq Kamil

NIM. 102410101086

SKRIPSI

Pengembangan *Augmented Reality Mobile Navigation*
Universitas Jember Dengan Pendekatan Berbasis Skenario.

Oleh:

Ahmad Syafiq Kamil
NIM. 102410101095

Menyetujui

Pembimbing Utama

Pembimbing Anggota

Nelly Oktavia Adiwijaya, S.Si., MT
NIP. 198410242009122008

Yanuar Nurdiansyah ST,.M.Cs.
NIP. 198201012010121004

PENGESAHAN

Skripsi berjudul “Pengembangan *Augmented Reality Mobile Navigation* Universitas Jember Dengan Pendekatan Skenario”, telah diuji dan disahkan pada:

Hari tanggal : Rabu, 11 Maret 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Penguji 1,

Penguji 2,

Anang Andrianto ST.,MT
NIP.196906151997021002

Windi Eka Yulia Retnani S.Kom., MT
NIP. 19840305 201012 2 002

Mengesahkan
Ketua Program Studi

Prof. Drs. Slamir, M.Comp.Sc.,Ph.D
NIP. 19670420 1992011001

Ahmad Syafiq Kamil

*Developing Augmented Reality Mobile Navigation Jember University
with Scenario Approach*

Navigasi, Universitas Jember, *Augmented Reality Mobile Navigation, Augmented Reality*

ABSTRACT

Navigation means for determine position and direction of travel where in actual field or in map. Navigation has been growing rapidly to be modern navigation which using GPS for obtain user position. Position has been obtained will be projected in map. Navigation helps people to find route to the desired location. Smartphone has GPS receiver make navigation technology is growing rapidly. More developer has been developed application in navigation field. Jember University has 1.120.508,5 m² land area and has 8 graduate program also has 15 undergraduate program. jember university has many building scattered in their region so that sometimes many people difficult to the desired location. Many people also can't read map. Growing technologies and problems, then it takes an application that can help people to find the shortest route. The application is also able to facilitate users who can not read a map. These application is Augmented Reality Mobile Navigation. The application able to find the shortest route from user location to desired location. This application is presenting route in the form of the map and in the form of augmented reality. With this application is expected to assist users in finding the nearest route and guide users to the site.

RINGKASAN

PENGEMBANGAN *AUGMENTED REALITY MOBILE NAVIGATION* UNIVERSITAS JEMBER DENGAN PENDEKATAN BERBASIS SKENARIO;

Ahmad Syafiq Kamil;102410101086;2014;109 halaman; Program Studi Sistem Informasi Universitas Jember.

Universitas Jember adalah sebuah perguruan tinggi negeri yang terletak di kota jember. Universitas Jember memiliki 13 fakultas, 2 program studi dan 8 program studi pasca sarjana magister dan doktoral. Setiap fakultas memiliki gedung yang tersebar di seluruh area Univeritas Jember yang memiliki luas area seluas 1.120.508,5 m² (Tim Penyusun 2012). Banyaknya gedung yang tersebar di Universitas Jember, menyebabkan banyak orang awam kebingungan untuk mencapai lokasi gedung yang ingin dituju. Sistem navigasi dapat menjadi alternatif dalam penentuan rute lokasi fakultas yang dicari.

Hadirnya *smartphone* dengan sistem operasi Android yang memiliki fitur penerima sinyal *Global Positioning System* (GPS) memberikan kemudahan pada pengguna untuk mendapatkan lokasi pengguna. GPS menggunakan satelit untuk mendapatkan lokasi pengguna. Salah satu teknologi yang ada dalam GPS untuk mendapatkan lokasi pengguna adalah AVL (*Automated Vehicle Locater*). AVL memungkinkan pengguna melacak posisi pengguna secara real time.

Penelitian sebelumnya telah dilakukan oleh beberapa peneliti, seperti yang dilakukan oleh Sung Hyun Jang dan Andrew Hudson-Smith yang mengembangkan aplikasi *Mobile Augmented Reality* untuk pejalan kaki. Aplikasi ini menuntun pengguna untuk sampai di lokasi yang diinginkan oleh pengguna dan divisualisasikan menggunakan teknologi *Augmented Reality* (Jang and Hudson-Smith 2010). Peneliti menggunakan pendekatan skenario untuk mengembangkan aplikasi tersebut. Penelitian sebelumnya juga telah dilakukan oleh Andreas Möller yang mengembangkan aplikasi *Mobile Navigation* yang diadaptasi menggunakan *Vision-*

Based Localization. (Möller, et al. 2012). Adam Wojciechowski juga telah meneliti penggunaan sinyal *GPS* dan kompas terhadap posisi relatif dan orientasi dari sebuah ponsel android sehingga penggunaan *Augmented Reality* terhadap bidang navigasi akan sangat nyata (Wojciechowski 2012).

Berdasarkan ulasan di atas, maka penelitian pengembangan *Mobile Navigation* di lingkungan Universitas Jember dapat menjadi alternatif sarana penunjuk arah di Universitas Jember, juga dalam hal pengembangan teknologi dalam area kampus Universitas Jember. Penelitian ini akan dikembangkan dengan menggunakan teknologi *Augmented Reality*. Aplikasi ini juga akan menggunakan *GPS Tracking* untuk menentukan lokasi awal pengguna dan lokasi tujuan. Selain itu, aplikasi ini akan mampu menentukan rute terdekat dan menuntun pengguna menggunakan rute tersebut. Metode yang akan digunakan oleh peneliti adalah pendekatan skenario untuk analisis kebutuhan sehingga sesuai dengan apa yang dibutuhkan oleh pengguna.

PRAKATA

Puji syukur kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis telah dapat menyelesaikan skripsi dengan judul “Pengembangan Augmented Reality Mobile Navigation Universitas Jember Dengan Pendekatan Skenario”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah membantu penyusunan skripsi ini. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamir, M.CompSc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember.
2. Nelly Oktavia Adiwijaya S.Si., M.T., selaku Dosen Pembimbing Utama, Dosen Pembimbing Akademik dan menjadi ibunda selama penulis menjadi mahasiswa serta Yanuar Nurdiansyah ST.,M.Cs. selaku Dosen Pembimbing Anggota yang telah penulis anggap seperti orang tua penulis.
3. Anang Andrianto ST.,MT selaku Dosen Penguji I serta Windi Eka Yulia Retnani S.Kom., MT selaku Dosen Penguji II yang telah mengarahkan penulis untuk lebih baik dalam penyusunan skripsi.
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember.
5. Abi Mashudi dan Umi Robiah yang selalu mencintai penulis.
6. Adek-adekku Muhammad Reza Wardana, Zidna Mazidah, Erda Afifah dan Muhammad Sirajul Munir.
7. Nayirotul Faiqoh yang selalu ada untuk memberi lebih dari motivasi kepada penulis.
8. Uklam Foundation, Maji, Bombom, Awang, Rasya, Indra, Doci, Hamdan, Yusa, Nay, Anggi, Keceng, Ain, dan Iwed.
9. Keluarga besar Zerone

10. Bu Nyai Lilik Istiqomah selaku ibu pengasuh selama penulis tinggal, keluarga besar kamar 4 dan keluarga besar PPM Al-Jauhar.
11. Serta banyak pihak yang tidak dapat penulis sebutkan satu persatu, termasuk mie sedap yang selalu menemani dikala akhir bulan.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Februari 2015

Penulis

DAFTAR ISI

SKRIPSI	i
SKRIPSI	i
PERSEMBAHAN.....	ii
MOTTO	iii
PERNYATAAN	iv
PENGESAHAN	vi
<i>ABSTRACT</i>	vii
RINGKASAN	viii
PRAKATA	x
DAFTAR ISI	xii
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvii
BAB 1 . PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan dan Manfaat	3
1.4 Ruang Lingkup Studi	4
BAB 2 . TINJAUAN PUSTAKA.....	5
2.1. Universitas Jember	5
2.2. Navigasi.....	6
2.3. <i>Global Positioning System</i>	6
2.4. Sistem Infomasi Geografis	8
2.5. <i>Google Maps</i>	10
2.6. <i>Augmented Reality</i>	11
2.7. Pendekatan Skenario dan Fungsi.....	17
2.8. Pengembangan Aplikasi	22
2.9. <i>Unified Modeling Language (UML)</i>	23
2.10. Pengujian Aplikasi	26

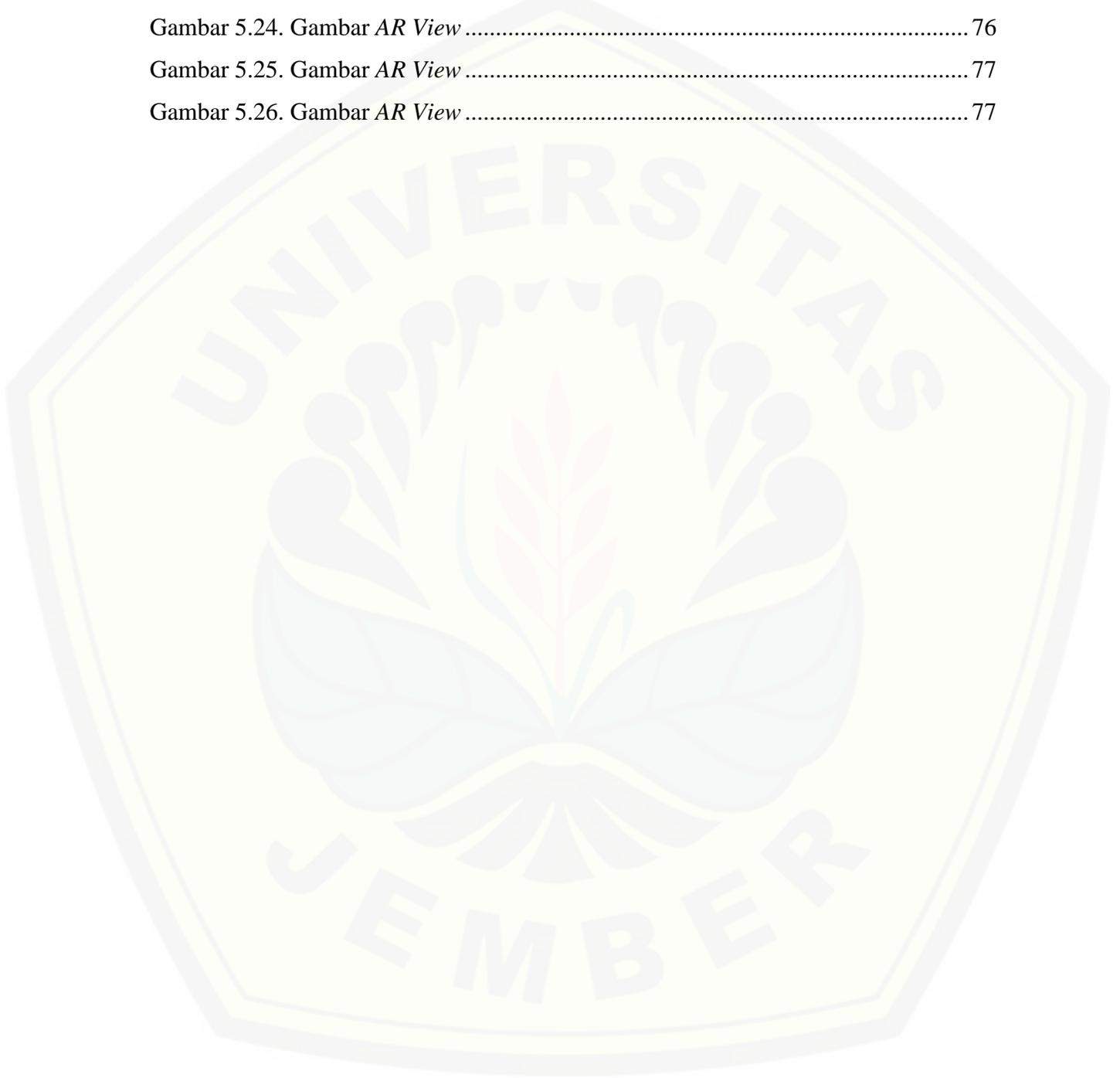
BAB 3 . METODOLOGI PENELITIAN	28
3.1. Tahap Perencanaan.....	28
3.2. Penentuan Metode dan Implementasi	28
3.3. Pengembangan aplikasi	29
BAB 4 . DESAIN DAN PERANCANGAN	32
4.1. Deskripsi umum	32
4.2. Pengumpulan Data	33
4.3. Identifikasi Kebutuhan	34
4.4. Perancangan <i>Prototype</i> dan Perancangan Aplikasi	38
BAB 5 . HASIL DAN PEMBAHASAN.....	49
5.1 Evaluasi <i>Prototype</i>	49
5.2. Implementasi Aplikasi.....	49
5.3. Pengujian Aplikasi	69
5.4. Menggunakan Sistem	72
BAB 6 . PENUTUP.....	78
6.1 Kesimpulan.....	78
6.2. Saran	78
DAFTAR PUSTAKA	80
LAMPIRAN 1	82
A. Pengembangan <i>Prototype</i>	82
B. Pengujian <i>White Box</i>	88
C. Pengujian <i>Black Box</i>	106
D. Pengujian Pendekatan Skenario	108

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Cara Kerja <i>GPS</i>	7
Gambar 2.2. Gambar peta yang didapatkan dari <i>google maps</i>	11
Gambar 2.3. Contoh Masukan (Lazuardy 2012).....	12
Gambar 2.4. Contoh Komponen Kamera (Lazuardy 2012).....	12
Gambar 2.5. Contoh Komponen Prosesor (Lazuardy 2012).....	13
Gambar 2.6. Contoh Keluaran (Lazuardy 2012).....	13
Gambar 2.7. Contoh Marker (Lazuardy 2012).....	14
Gambar 2.8. Contoh <i>Face Tracking</i> (Lazuardy 2012).....	15
Gambar 2.9. Contoh <i>Object Tracking</i> (Lazuardy 2012).....	15
Gambar 2.10. Contoh <i>Motion Tracking</i> (Lazuardy 2012)	16
Gambar 2.11. Contoh <i>GPS Based Tracking</i> (Lazuardy 2012).....	16
Gambar 2.12 Contoh <i>Dependency Chart</i>	22
Gambar 2.13. Contoh <i>Use Case Diagram</i> (Almendros-Jimenez and Iribarne 2009). 23	
Gambar 2.14. Contoh <i>User Interaction Diagram</i> (Almendros-Jimenez and Iribarne 2009)	24
Gambar 2.15. Contoh <i>User Interface Diagram</i> (Almendros-Jimenez and Iribarne 2009)	25
Gambar 2.16. Contoh <i>GUI Class Diagram</i> (Almendros-Jimenez and Iribarne 2009)25	
Gambar 2.17. Contoh <i>Entity Relationship Diagram</i>	26
Gambar 2.18. contoh perhitungan siklomatik	26
Gambar 4.1. Pencocokan Data Pada <i>Google Earth</i>	33
Gambar 4.2. Data Gedung yang ada di Universitas Jember	33
Gambar 4.3. Titik-Titik Lokasi Gedung yang Berada di Universitas Jember	34
Gambar 4.4. Tampilan <i>Splash Screen</i>	39
Gambar 4.5. Tampilan Home.....	39
Gambar 4.6. Tampilan <i>Map</i>	40
Gambar 4.7. Tampilan <i>AR View</i>	40

Gambar 4.8. <i>Use Case ARMN</i>	41
Gambar 4.9. <i>Gambar User Interaction Diagram Aplikasi ARMN</i>	44
Gambar 4.10 <i>User Interface Diagram</i> dari Aplikasi ARMN	45
Gambar 4.11. <i>Splash Screen Class Diagram</i>	46
Gambar 4.12. <i>Class Diagram Home Activity</i>	46
Gambar 4.13. <i>Class Diagram Navigation Activity</i>	47
Gambar 4.14. <i>Class Diagram Ar Activity</i>	47
Gambar 4.15. <i>Entity Relationship Diagram</i> dari Aplikasi ARMN.....	48
Gambar 5.1. <i>Tampilan Splash Screen</i>	50
Gambar 5.2. <i>Source Code</i> dari <i>Splash Activity</i>	52
Gambar 5.3. <i>Progress Dialog Checkgps</i>	53
Gambar 5.4. <i>Alert Dialog</i> Ketika <i>Gps</i> Tidak Aktif.....	53
Gambar 5.5. <i>Tampilan Setting</i>	54
Gambar 5.6. <i>Progress Dialog</i> Mengambil Lokasi	55
Gambar 5.7. <i>Tampilan Home Activity</i>	55
Gambar 5.8. <i>source code home activity</i>	60
Gambar 5.9. <i>Progress Dialog</i> Pada <i>Navigation Activity</i>	61
Gambar 5.10. <i>Tampilan Map</i> dan Rute Pada <i>Navigation Activity</i>	62
Gambar 5.11. <i>Source Code</i> Dari <i>Navigation Activity</i>	64
Gambar 5.12. <i>Tampilan AR Activity</i>	65
Gambar 5.13. <i>Source Code AR Activity</i>	69
Gambar 5.14. <i>Diagram Alir Method Oncreate</i> Kelas <i>Splash Activity</i>	70
Gambar 5.15. <i>Dependency Chart</i> Dari Aplikasi ARMN	72
Gambar 5.16. <i>Gambar Splash Screen</i>	72
Gambar 5.17. <i>Progress Dialog Check Gps</i>	73
Gambar 5.18 <i>Progress Dialog Getting Location</i>	73
Gambar 5.19. <i>Gambar Home Screen</i>	74
Gambar 5.20. <i>Memilih Lokasi Tujuan</i>	74
Gambar 5.21. <i>Gambar Progress Dialog Getting Location</i>	75

Gambar 5.22. Gambar <i>Map</i> Beserta Rute Dan Titik Gedung Tujuan.....	75
Gambar 5.23. Gambar <i>Ar View</i>	76
Gambar 5.24. Gambar <i>AR View</i>	76
Gambar 5.25. Gambar <i>AR View</i>	77
Gambar 5.26. Gambar <i>AR View</i>	77



DAFTAR TABEL

	Halaman
Tabel 2.1 Matrik Pendekatan Skenario (Hart, Nilsson and Raphael 1986)	18
Tabel 2.2. Tabel Matrik dari Skenario	19
Tabel 2.3. Fungsionalitas (Jang and Hudson-Smith 2010)	20
Tabel 2.4. Contoh <i>Test Case</i>	21
Tabel 3.1. tabel <i>facet</i> pendekatan skenario	29
Tabel 4.1. Tabel Matrik Skenario	35
Tabel 4.2. Tabel Fungsional.....	37
Tabel 4.3. Tabel <i>Use Case Scenario</i>	41
Tabel 5.1. Tabel Test Case Dari <i>User Interaction Diagram Splash Screen</i>	71

BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi, metodologi penelitian dan sistematika penulisan.

1.1 Latar Belakang

Beberapa dekade terakhir, penggunaan sistem informasi geografis (SIG) telah berkembang secara pesat. Penggunaan SIG telah terintegrasi ke dalam kehidupan sehari-hari. Banyaknya aplikasi yang menggunakan SIG dapat membantu kehidupan sehari-hari. Contoh aplikasi SIG yang telah ada yaitu aplikasi yang dapat membantu orang dalam mencari lokasi yang diinginkan.

Universitas Jember adalah sebuah perguruan tinggi negeri yang terletak di kota Jember. Universitas Jember memiliki 13 fakultas, 2 program studi dan 8 program studi pasca sarjana magister dan doktoral. Setiap fakultas memiliki gedung yang tersebar di seluruh area Universitas Jember yang memiliki luas area seluas 1.120.508,5 m² (Tim Penyusun 2012). Banyaknya gedung yang tersebar di Universitas Jember, menyebabkan banyak orang awam kebingungan untuk mencapai lokasi gedung yang ingin dituju. Sistem navigasi dapat menjadi alternatif dalam penentuan rute lokasi fakultas yang dicari.

Hadirnya *smartphone* dengan sistem operasi Android yang memiliki fitur penerima sinyal *Global Positioning System* (GPS) memberikan kemudahan pada pengguna untuk mendapatkan lokasi pengguna. GPS menggunakan satelit untuk mendapatkan lokasi pengguna. Salah satu teknologi yang ada dalam GPS untuk mendapatkan lokasi pengguna adalah AVL (*Automated Vehicle Locater*). AVL memungkinkan pengguna melacak posisi pengguna secara real time.

Penelitian sebelumnya telah dilakukan oleh beberapa peneliti, seperti yang dilakukan oleh Sung Hyun Jang dan Andrew Hudson-Smith yang mengembangkan aplikasi *Mobile Augmented Reality* untuk pejalan kaki. Aplikasi ini menuntun

pengguna untuk sampai di lokasi yang diinginkan oleh pengguna dan divisualisasikan menggunakan teknologi *Augmented Reality* (Jang and Hudson-Smith 2010). Peneliti menggunakan pendekatan skenario untuk mengembangkan aplikasi tersebut. Penelitian sebelumnya juga telah dilakukan oleh Andreas Möller yang mengembangkan aplikasi *Mobile Navigation* yang diadaptasi menggunakan *Vision-Based Localization*. (Möller, et al. 2012). Adam Wojciechowski juga telah meneliti penggunaan sinyal *GPS* dan kompas terhadap posisi relatif dan orientasi dari sebuah ponsel android sehingga penggunaan *Augmented Reality* terhadap bidang navigasi akan sangat nyata (Wojciechowski 2012).

Berdasarkan ulasan di atas, maka penelitian pengembangan *Mobile Navigation* di lingkungan Universitas Jember dapat menjadi alternatif sarana penunjuk arah di Universitas Jember, juga dalam hal pengembangan teknologi dalam area kampus Universitas Jember. Penelitian ini akan dikembangkan dengan menggunakan teknologi *Augmented Reality*. Aplikasi ini juga akan menggunakan *GPS Tracking* untuk menentukan lokasi awal pengguna dan lokasi tujuan. Selain itu, aplikasi ini akan mampu menentukan rute terdekat dan menuntun pengguna menggunakan rute tersebut. Metode yang akan digunakan oleh peneliti adalah pendekatan skenario untuk analisis kebutuhan sehingga sesuai dengan apa yang dibutuhkan oleh pengguna.

1.2 Rumusan Masalah

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

- a. Bagaimana penerapan pendekatan secara skenario untuk mengembangkan aplikasi *Mobile Navigation* menggunakan teknologi *Augmented Reality*?
- b. Bagaimana merancang dan membangun sebuah aplikasi *Mobile Navigation* menggunakan teknologi *Augmented Reality* ?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat berisi tentang tujuan penelitian rancang dan bangun aplikasi mobile navigation menggunakan teknologi augmented reality dan manfaat yang akan diperoleh dari penelitian ini akan di bagi menjadi 3 bagian yaitu, manfaat akademis, manfaat bagi peneliti dan manfaat bagi objek penelitian .

a. Tujuan

Tujuan dari penelitian ini adalah :

- 1) Memudahkan pengguna dalam menentukan rute tercepat tiap fakultas, sehingga pencarian fakultas akan semakin efisien.
- 2) Membangun sebuah aplikasi *Android Mobile Navigation* berbasis *Augmented Reality*.

b. Manfaat

Manfaat dari penelitian ini adalah :

1) Manfaat Akademis

Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi berkaitan dengan judul penelitian ini. Sehingga dapat dijadikan bahan referensi dan bahan inspirasi untuk dapat dikembangkan dalam penelitian selanjutnya.

2) Manfaat bagi Peneliti

- a) Mengetahui bagaimana merancang dan membangun sebuah aplikasi *Mobile Navigation* menggunakan teknologi *Augmented Reality*.
- b) Mengetahui Bagaimana penerapan pendekatan secara skenario untuk mengembangkan aplikasi *Mobile Navigation* menggunakan teknologi *Augmented Reality*.

3) Manfaat bagi pengguna

- a) Membantu pengguna dalam mencari rute terdekat antar fakultas Universitas Jember.

- b) Memudahkan dalam membaca peta.
- c) Menuntun pengguna untuk menuju fakultas yang diinginkan

1.4. Ruang Lingkup Studi

Ruang lingkup studi dalam penelitian ini merupakan lingkup batasan-batasan masalah. Batasan masalah dalam penelitian ini diantaranya adalah sebagai berikut :

- a. Penelitian ini merupakan penelitian yang bersifat kualitatif.
- b. Daerah penelitian hanya terbatas pada lingkungan universitas jember.
- c. Aplikasi berupa aplikasi *mobile*.
- d. Aplikasi ini hanya untuk digunakan dengan berjalan kaki, bukan dengan menggunakan kendaraan.

BAB 2 . TINJAUAN PUSTAKA

Penelitian ini membutuhkan beberapa landasan teori yang akan digunakan untuk memperkuat dan mengarahkan penelitian agar tidak keluar dari kaidah keilmuan yang ada. Berdasarkan studi pustaka yang telah dilakukan dalam penelitian ini membutuhkan beberapa dasar teori yaitu :

- a. Universitas Jember
- b. Navigasi
- c. Global Positioning System
- d. Sistem Informasi Geografis
- e. Google Maps
- f. Augmented Reality
- g. Pendekatan Skenario
- h. Perancangan Aplikasi
- i. *Unified Modeling Language (UML)*
- j. Pengujian Aplikasi

2.1. Universitas Jember

Universitas Jember (UNEJ) adalah sebuah perguruan tinggi negeri yang terletak di kota Jember, sebuah kota berhawa tropis di bagian tenggara Provinsi Jawa Timur. Kampus UNEJ berada di kawasan hijau yang ramah lingkungan sehingga memberikan ketenangan dalam melaksanakan kegiatan akademik (Tim Penyusun 2012).

UNEJ memiliki luas lahan yaitu 1.120.508,5 m² dengan penataan bangunan perkantoran , gedung pendidikan dan lahan penunjang seluas 127.508,5 m². Saat ini, UNEJ berkembang telah memiliki 8 program studi (PS) Pasca Sarjana S2 dan S3 dan Program Sarjana S1 yang terdiri 15 Fakultas/PS setara Fakultas, yaitu : (1) Fakultas Hukum, (2) Fakultas Ilmu Sosial Dan Ilmu Politik, (3) Fakultas Pertanian, (4) Fakultas Ekonomi, (5) Fakultas Keguruan Dan Ilmu Pendidikan, (6) Fakultas Sastra,

(7) Fakultas Teknologi Pertanian, (8) Fakultas Kedokteran Gigi, (9) Fakultas Matematika Dan Ilmu Pengetahuan Alam, (10) Fakultas Kedokteran, (11) Fakultas Teknik, (12) Fakultas Kesehatan Masyarakat, (13) Fakultas Farmasi, (14) Program Studi Ilmu Keperawatan, dan (15) Program Studi Sistem Informasi

2.2. Navigasi

Navigasi menurut Somantri berasal dari bahasa Yunani yang terdiri dari kata *navis* yang artinya perahu atau kapal dan *agake* yang artinya mengarahkan, secara harfiah artinya mengarahkan sebuah kapal dalam pelayaran. Dari waktu ke waktu seiring dengan perkembangan zaman kata navigasi tidak lagi hanya digunakan dalam dunia maritime tetapi sering juga digunakan di daratan dan udara. Navigasi adalah cara menentukan posisi dan arah perjalanan baik di medan sebenarnya maupun pada peta.

Teknologi navigasi dibedakan menjadi 2 bagian yakni navigasi kuno dan navigasi modern. Pada saat ini sistem navigasi berpusat kepada perkembangan kompas yang menjadi lebih sempurna dan kemudian berkembang menjadi navigasi radar dan pada akhirnya menjadi teknologi navigasi berbasis melalui alat yang bernama *Global Positioning System (GPS)*. Prinsip kerja navigasi adalah menentukan arah dan posisi, yaitu arah yang akan dituju dan posisi keberangkatan di medan sebenarnya kemudian diproyeksikan pada peta, kompas dan GPS.

2.3. Global Positioning System

Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang dikembangkan oleh Departemen Pertahanan Amerika dengan 27 satelit yang mendukungnya. Dahulu GPS merupakan sebuah perangkat yang berdiri sendiri untuk berbagai keperluan navigasi seperti untuk menentukan letak, kecepatan, arah, dan waktu. navigasi pesawat terbang, kapal, bahkan juga digunakan untuk mempelajari kebiasaan migrasi satwa (Mulyoto 2013). Gambar 2.1 adalah gambar dari cara kerja *GPS*. Pada gambar tersebut *GPS* memiliki 3 bagian yaitu :

a. Bagian kontrol

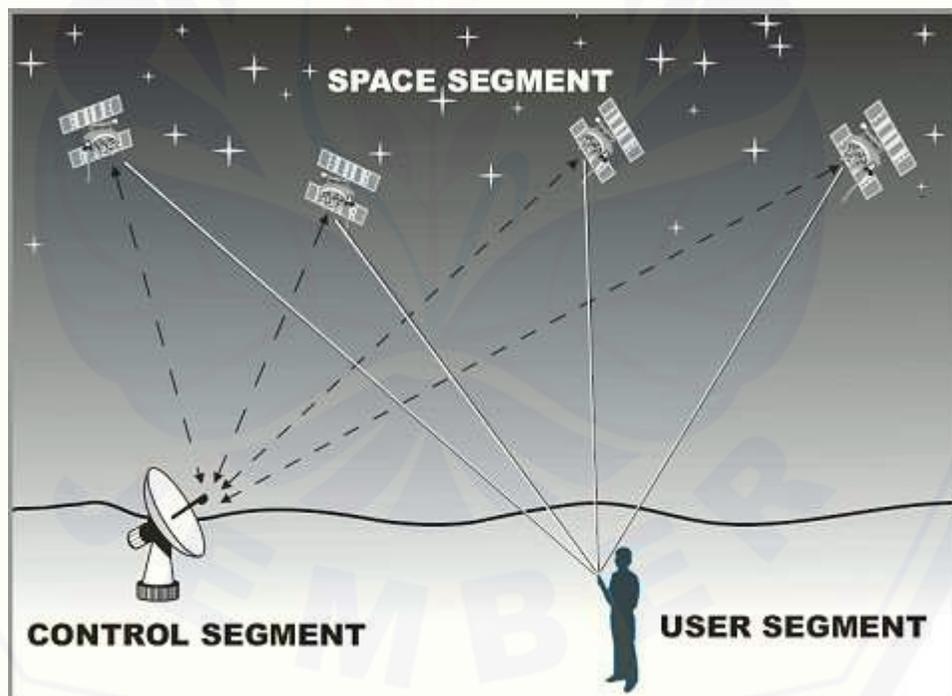
Bagian ini menerima sinyal berupa orbit satelit, lokasi, ketinggian dan kecepatan. Bagian ini akan mengoreksi sinyal yang telah didapat dan dikirim kembali ke satelit. Data yang telah dikoreksi akan dikirimkan ke alat penerima sinyal *GPS*

b. Bagian angkasa

Bagian ini merupakan sekumpulan satelit yang diatur sedemikian rupa sehingga setiap alat navigasi dapat menerima sinyal paling sedikit empat buah satelit. Tiap satelit memiliki kode yang berfungsi untuk mengukur jarak antar alat navigasi dengan satelit. Kekuatan sinyal juga akan membantu alat dalam perhitungan.

c. Bagian pengguna

Alat navigasi membutuhkan paling sedikit sinyal dari 3 buah satelit untuk menunjukkan koordinat sebuah titik dalam bentuk dua dimensi. Untuk menunjukkan data ketinggian dari sebuah titik, diperlukan tambahan sinyal dari satu buah satelit.



Gambar 2.1. Cara Kerja *GPS*

2.4. Sistem Informasi Geografis

Sistem informasi geografis adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Atau dalam arti yang lebih sempit, adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi menurut lokasinya, dalam sebuah *database*.

Definisi sistem informasi geografis menurut Aronoff adalah sistem yang berbasis komputer yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografis. SIG dirancang untuk mengumpulkan, menyimpan, dan menganalisis objek-objek dan fenomena di mana lokasi geografis merupakan karakteristik yang penting atau kritis untuk dianalisis (Aronoff 1989). Dengan demikian, SIG merupakan sistem komputer yang memiliki empat kemampuan berikut dalam menangani data yang bereferensi geografis : (a) masukan, (b) manajemen data (penyimpanan data dan pemanggilan data), (c) analisis dan manipulasi data, dan (d) keluaran (Prahasta 2009).

2.4.1. Subsistem Sistem Informasi Geografis

SIG mempunyai kemampuan untuk menghubungkan berbagai data pada suatu titik tertentu di bumi, menggabungkannya, menganalisa, dan akhirnya memetakan hasilnya. Data yang diolah pada SIG adalah data spasial yaitu sebuah data yang berorientasi geografis dan merupakan lokasi yang memiliki sistem koordinat tertentu, sebagai dasar referensinya. Sehingga aplikasi SIG dapat menjawab beberapa pertanyaan seperti lokasi, kondisi, tren, pola dan pemodelan. Kemampuan inilah yang membedakan SIG dengan sistem informasi lainnya. SIG dapat diruaikan menjadi beberapa subsistem

1) *Data Input*

Subsistem ini bertugas untuk mengumpulkan, mempersiapkan, dan menyimpan data spasial dan atributnya dari berbagai sumber. Sub-sistem ini pula yang bertanggung jawab dalam mengonversikan atau mentransformasikan format-format

data aslinya ke dalam format yang dapat digunakan oleh perangkat SIG yang bersangkutan.

2) *Data Output*

Sub-sistem ini bertugas untuk menampilkan atau menghasilkan keluaran (termasuk mengekspornya ke format yang dikehendaki) seluruh atau sebagian basis data (spasial) baik dalam bentuk *softcopy* maupun *hardcopy* seperti halnya tabel, grafik, *report*, peta, dan lain sebagainya.

3) *Data Management*

Sub-sistem ini mengorganisasikan baik data spasial maupun tabel-tabel atribut terkait ke dalam sebuah sistem basis data sedemikian rupa hingga mudah dipanggil kembali atau di *retrieve*, di *update*, dan di *edit*.

4) *Data Manipulation dan Analysis*

Sub-sistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu sub-sistem ini juga melakukan manipulasi (evaluasi dan penggunaan fungsifungsi dan operator matematis & logika) dan pemodelan data untuk menghasilkan informasi yang diharapkan.

2.4.2. Komponen Sistem Informasi Geografis

Selain sub sistem tersebut, terdapat lima komponen yang membentuk sistem informasi geografis yaitu :

a. Orang

SIG memiliki tingkatan pengguna dari tingkat spesialis teknis yang mendesain dan memelihara sistem sampai pada pengguna yang memanfaatkan SIG untuk mempermudah pekerjaan pengguna perangkat lunak dan aplikasi SIG.

b. Perangkat lunak

Karakteristik SIG membutuhkan perangkat lunak yang mendukung analisis, penyimpanan dan visualisasi informasi geografis. Perangkat lunak sendiri terdiri dari sistem operasi, compiler dan program aplikasi.

c. Perangkat Keras

Kehandalan suatu Sistem Informasi Geografis akan sangat didukung jika memiliki spesifikasi perangkat keras yang handal. Kehandalan ini bisa dilihat dari kemampuan *processor* yang lebih cepat, memory yang lebih tinggi, harddisk yang lebih besar, dan *Video Graphic Adapter card* yang lebih bagus dibandingkan dengan sistem informasi biasa.

d. Data

Data merupakan komponen terpenting dalam SIG. Data yang dibutuhkan terdiri dari data spasial (data peta) dan data non-spasial (data atribut). Data spasial berintegrasi dengan data non-spasial pada setiap fiturnya. Pembangunan dan pengolahan basis data yang lebih besar maka data tabular tersebut dapat direlasikan ke sumber data lain yang berada di luar tools SIG melalui *Database Management System (DBMS)*.

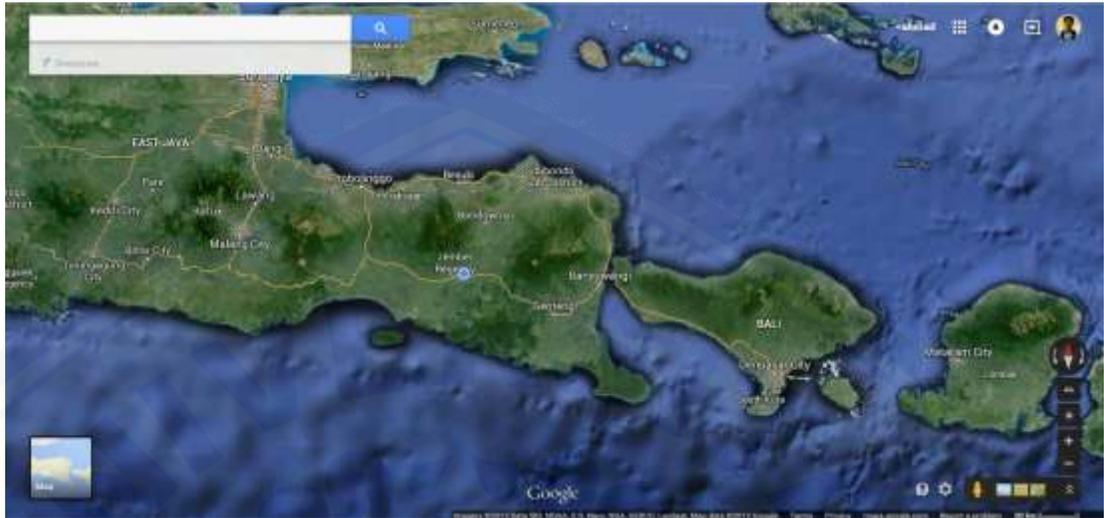
e. Metode

Perlu metode dan cara penerapan yang unik untuk setiap permasalahan SIG. Oleh karena itu, SIG yang baik tergantung pada aspek disain yang bagus dan aturan bisnis atau kondisi nyata.

2.5. Google Maps

Google Maps adalah sebuah layanan dari google yang menyediakan informasi secara detail tentang wilayah secara geografi dan situs di seluruh dunia. *Google Maps* menawarkan pemandangan dari udara dan satelit berbagai tempat di dunia (Gibilisco 2013).

Google Maps telah dikembangkan untuk dapat berjalan di berbagai *platform*, salah satunya adalah berjalan di *platform mobile* yaitu *android*. *Google* telah menyediakan *application program interface(API)* untuk para pengembang aplikasi. *API* adalah sekumpulan *routines*, *protocol* dan alat untuk membangun aplikasi. Contoh gambar peta yang telah didapatkan dari *google maps* seperti pada gambar 2.2.



Gambar 2.2. Gambar Peta Yang Didapatkan dari *Google Maps*

2.6. *Augmented Reality*

Augmented Reality adalah penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat *input* tertentu, dan integrasi yang baik memerlukan penjejakan yang efektif (Azuma 1997).

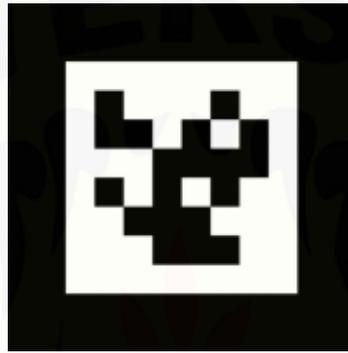
Selain menambahkan benda maya dalam lingkungan nyata, *Augmented Reality* juga berpotensi menghilangkan benda-benda yang sudah ada. Menambah sebuah lapisan gambar maya dimungkinkan untuk menghilangkan atau menyembunyikan lingkungan nyata dari pandangan pengguna. Misalnya, untuk menyembunyikan sebuah meja dalam lingkungan nyata, perlu digambarkan lapisan representasi tembok dan lantai kosong yang diletakkan di atas gambar meja nyata, sehingga menutupi meja nyata dari pandangan pengguna. Aplikasi *Mobile Navigation* berbasis *Augmented Reality*. *Augmented reality* memiliki komponen dan metode untuk membangun sebuah aplikasi berbasis *augmented reality*.

2.6.1 Komponen *Augmented Reality*

Augmented Reality terdiri dari beberapa komponen yaitu

a. Masukan

Masukan dapat berupa apa saja, contoh marker, gambar 2D, gambar 3D, sensor *WI-FI*, sensor gerakan, *GPS*, dan sensor-sensor yang lain seperti terlihat pada gambar 2.3.



Gambar 2.3. Contoh Masukan (Lazuardy 2012)

b. Kamera

Kamera disini sebagai perantara untuk input yang berupa gambar, baik itu marker, gambar 2D maupun 3D seperti terlihat pada gambar 2.4.



Gambar 2.4. Contoh Komponen Kamera (Lazuardy 2012)

c. Prosesor

Prosesor dibutuhkan untuk memproses input yang masuk dan kemudian memberikan tahapan output. Contoh dari prosesor adalah seperti pada gambar 2.5.



Gambar 2.5. Contoh Komponen Prosesor (Lazuardy 2012)

d. Keluaran

Dapat berupa HMD, monitor TV, LCD, dan monitor ponsel. Contoh keluaran seperti pada gambar 2.6.



Gambar 2.6. Contoh Keluaran (Lazuardy 2012)

2.6.2. Metode *Augmented Reality*

Terdapat dua metode yang digunakan oleh teknologi *Augmented Reality* yaitu *Marker Based Tracking*, *Markerless Based Tracking* (Lazuardy 2012).

a. *Marker Based Tracking*

Marker Based Tracking sudah dikembangkan sejak tahun 1980 dan pada awal 1990 mulai dikembangkan untuk penggunaan *Augmented Reality*. *Marker* merupakan ilustrasi hitam putih persegi dengan batas hitam tebal dan latar belakang putih seperti pada gambar 2.7. Komputer akan mengenali posisi dan orientasi marker dan menciptakan dunia virtual 3D .



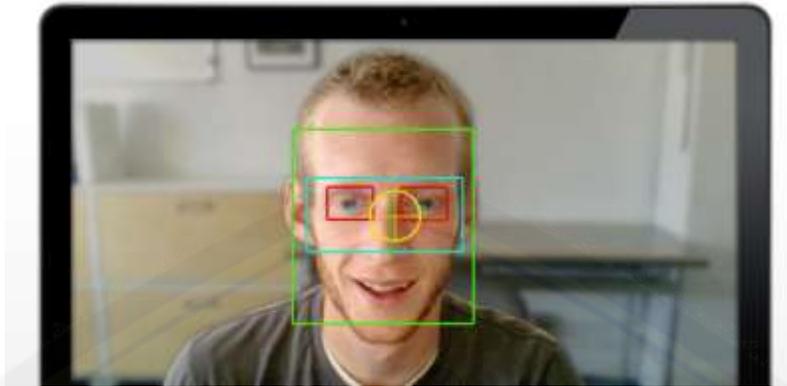
Gambar 2.7. Contoh Marker (Lazuardy 2012)

b. *Markerless Based Tracking*

Metode selanjutnya yang sedang berkembang adalah metode *Markerless Based Tracking*. Metode ini tidak perlu lagi menggunakan sebuah marker untuk menampilkan elemen-elemen digital.

1) *Face Tracking*

Menggunakan teknik algoritma yang telah dikembangkan, komputer juga dapat mengenali wajah manusia secara umum dengan cara mengenali posisi mata, hidung dan mulut seperti terlihat pada gambar 2.8. Kemudian akan mengabaikan objek-objek lain disekitarnya seperti pohon, rumah dan benda lainnya.



Gambar 2.8. Contoh *Face Tracking* (Lazuardy 2012)

2) *3D Object Tracking*

Berbeda dengan face tracking yang hanya mengenali wajah manusia. Dengan menggunakan teknik *3D Object Tracking* dapat mengenali semua benda yang berada disekitar seperti mobil, motor, meja, tv dan lain-lain. Contoh penggunaan *3D Object Tracking* seperti pada gambar 2.9.



Gambar 2.9. Contoh *Object Tracking* (Lazuardy 2012)

3) *Motion Tracking*

Teknik komputer ini dapat menangkap gerakan atau *Motion Tracking* yang telah dimulai digunakan secara ekstensif untuk memproduksi sebuah film-film yang mensimulasikan pada gerakan-gerakan tubuh. Gambar 2.10 menunjukkan contoh dari *Motion tracking*.



Gambar 2.10. Contoh *Motion Tracking* (Lazuardy 2012)

4) *GPS Based Tracking*

Teknik *GPS Based Tracking* saat ini mulai populer dan banyak dikembangkan pada aplikasi smartphone (*iPhone* dan *Android*). Dengan memanfaatkan fitur GPS dan kompas yang ada didalam smartphone, aplikasi akan mengambil data dari GPS dan kompas kemudian menampilkannya dalam bentuk arah yang kita inginkan secara realtime, bahkan ada beberapa aplikasi menampikannya dalam bentuk 3D. salah satu penggunaan *Augmented Reality* yang menggunakan *GPS based tracking* terlihat pada gambar 2.11.



Gambar 2.11. Contoh *GPS Based Tracking* (Lazuardy 2012)

2.7. Pendekatan Skenario dan Fungsi

Pendekatan skenario adalah pendekatan yang menggunakan sudut pandang user sebagai alat menghitung kebutuhan yang sesuai dengan apa yang dibutuhkan oleh sistem. (Giboin, et al. 2004)

a. Skenario

Pendekatan skenario telah diperkenalkan dalam Interaksi Manusia-Komputer dan komunitas *Computer-Supported Cooperative Work* untuk mengatasi kekurangan yang ada pada pendekatan secara tradisional (Giboin, et al. 2004). Pendekatan skenario menggunakan sudut pandang pengguna dalam proses pembuatan desain. Pendekatan skenario ini didefinisikan sebagai deskripsi, terkadang secara narasi mengenai hal detail yang dilakukan oleh pengguna dan pengalamannya ketika menggunakan sistem. Pendekatan skenario mengumpulkan kebutuhan pengguna dengan membentuk properti, kualitas atau kriteria yang harus ada dalam sistem sehingga dapat diterima oleh pengguna, juga membentuk kriteria yang harus ditemukan ketika menguji sistem.

Analisis kebutuhan dengan menggunakan pendekatan skenario terdapat 3 cara yaitu : *Charasteristics* , *Representation* dan *Facet*. *Charasteristics* dan *Representation* menspesifikasikan tipe dari pendekatan skenario, sistem digambarkan dalam format textual dan situasi yang spesifik. Pada *Facet* lebih merujuk pada konten dari skenario, dimana aktor adalah pengguna yang terlibat langsung dalam skenario, memiliki peran tertentu, menggunakan sumber daya tertentu dalam cara tertentu. Gambaran matrik dari pendekatan skenario seperti terlihat pada tabel 2.1.

Tabel 2.1 Matrik Pendekatan Skenario (Hart, Nilsson and Raphael 1986)

Charasteristik	Representation	Facets
Tujuan :	Tekstual :	Aktor
Sebelum :	<i>Graphical:</i>	Kronologi
Sesudah :	Informal :	Sumber daya:
Batasan :	Formal :	Aliran:
Skenario :		Lingkungan:
Contoh spesifik :		
Relevansi waktu hidup :		
Contoh pengecualian :		

Salah satu contoh pendekatan skenario dalam penelitian sebelumnya yaitu *Exploring Mobile Augmented Reality Navigation System for Pedestrians* yang dilakukan oleh Sung Hung Jang dan Andrew Hudson-Smith (Jang and Hudson-Smith 2010) adalah :

- a. Skenario 1 : Nyonya Smith ingin bertemu dengan anaknya di apartemen baru
 Nyonya Smith ingin bertemu anaknya yang sedang kuliah di sebuah universitas. Dia tidak pernah berkunjung ke kota London dan ingin menemui anaknya yang berada di apartemen dekat dengan *Russel Square*. Dia memiliki peta kota London tapi tidak memiliki waktu untuk membaca peta dan mengorientasikan ke dunia nyata. Pada skenario ini menunjukkan potensi dari sistem navigasi menggunakan teknologi AR yang dapat melapisi informasi rute ke dunia fisik.
- b. Skenario 2 : Henry ingin menemukan jalan di sekitar kampus
 Henry adalah mahasiswa baru di UCL. Dia menghadiri kuliah dan seminar setiap hari, tetapi ruang kuliah dan ruang seminar tersebar seluruh kampus. Dia selalu menggunakan *Google Maps* untuk menemukan rute menuju kelasnya. Henry ingin memiliki peta gedung UCL sehingga dia dapat memilih ruang kuliah dari menu

dan menemukan jalan secara mudah. Skenario ini berfokus untuk menggunakan sistem navigasi AR di area tertentu seperti kampus.

c. Skenario 3 : Henry dan Jin ingin saling bertemu untuk makan siang bersama

Henry memiliki teman baru bernama Jim dan mereka berjanji untuk makan siang bersama, tetapi mereka adalah mahasiswa baru di UCL sehingga tidak mengetahui letak gedung tiap fakultas. Mereka berada di gedung yang berbeda dan hanya memiliki waktu satu jam untuk makan siang. Mereka memilih untuk bertemu di lokasi antara lokasi mereka sekarang. Skenario ini berfokus pada saling berbagi lokasi antar pengguna.

Setelah skenario telah didapatkan, maka akan dibuat tabel matrik dari ketiga skenario seperti tabel 2.2.

Tabel 2.2. Tabel Matrik dari Skenario

Komponen	Skenario 1	Skenario 2	Skenario 3
Aktor			
• Profil	• Nyonya smith	• Henri	• Henri dan jin
• Peran	• Orang tua	• Mahasiswa baru	• Mahasiswa baru
• Tugas	• Mencari apartemen	• Ingin menghadiri kuliah	• Ingin saling bertemu
Kronologi	• Nyonya smith berada di london	• Henri berada di kampus	• Henri dan jin berada di gedung yang berbeda
	• Membuka peta	• Membuka <i>google maps</i>	• Ingin bertemu di antara lokasi gedung
	• Mengikuti rute dari peta	• Mengikuti rute yang telah ditemukan oleh <i>google maps</i>	
Sumber daya	• Peta London	• <i>Smartphone</i>	• <i>Smartphone</i>
Aliran			
• <i>Input</i>	• Lokasi yang	• Lokasi yang akan dituju	• Lokasi yang akan

• <i>Ouput</i>	akan dituju	• Rute yang akan dituju	dituju
	• Rute yang akan dituju		• Rute yang akan dituju
lingkungan	• london	• University Collage London	• University Collage London

b. Fungsi

Setiap skenario menggambarkan situasi yang berbeda. Fungsi dari ketiga skenario dapat dilihat di tabel 2.3.

Tabel 2.3. Fungsionalitas (Jang and Hudson-Smith 2010)

	Skenario 1	Skenario 2	Skenario 3
Fungsi Umum	<ul style="list-style-type: none"> • Menampilkan rute dalam <i>AR View</i> • Mendeteksi lokasi pengguna secara otomatis • Menampilkan lokasi tujuan • <i>Update</i> lokasi pengguna secara berkala 		
Fungsi Khusus	Terdapat tampilan <i>input</i> yang pengguna	Memilih tujuan pada tempat area yang ditentukan	Berbagi infomasi lokasi antar pengguna

c. Pengujian Pada Pendekatan Skenario

Pengguna mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Tahap evaluasi ini menggunakan *test case* yang sesuai dengan

pendekatan skenario (Johannes Ryser 99). Terdapat 2 *Test case* yang digunakan untuk mengevaluasi pendekatan skenario yaitu:

- 1) Test Case yang berasal dari *user interaction diagram*.

Test case ini berupa tabel yang berasal dari *user interaction diagram*.

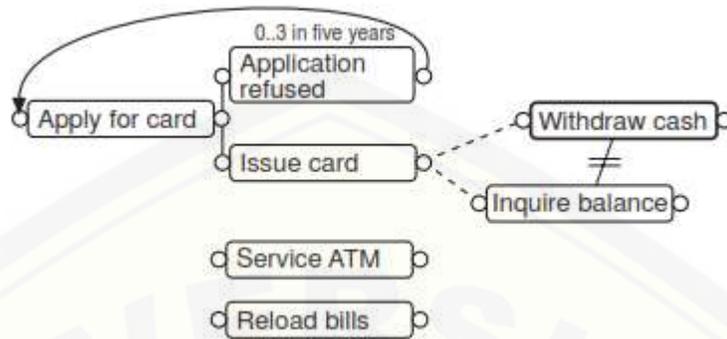
Tabel 2.4 adalah contoh tabel dari test case tersebut.

Tabel 2.4. Contoh *Test Case*

Test Preparation		Pengguna menekan ikon aplikasi	
No	status	Kondisi	Keluaran
1	<i>Delay</i> 3 detik	aplikasi mulai	<i>Splash screen</i> tampil
2	<i>Delay</i> 3 detik	Aplikasi mulai mengecek GPS	<i>Progress dialog</i> muncul
3	Cek <i>GPS</i>	Aplikasi menemukan GPS	Menuju halaman Home
4	Cek <i>GPS</i>	Aplikasi tidak dapat menemukan GPS	<i>Alert Dialog</i> muncul
5	<i>Alert Dialog</i>	Memilih pilihan keluar	Keluar aplikasi
6	Cek <i>GPS</i>	Aplikasi tidak dapat menemukan GPS	<i>Alert Dialog</i> muncul
7	<i>Alert Dialog</i>	Memilih pilihan setting	Masuk ke halaman <i>setting</i>
8	Halaman <i>setting</i>	Mengaktifkan <i>GPS</i>	<i>Alert dialog</i> muncul
9	<i>Alert Dialog</i>	Menekan <i>Ok</i>	Keluar dari aplikasi

- 2) Test Case yang berasal dari *dependency chart*.

Dependency chart berfungsi untuk menangkap logika dan timing antar skenario. Dependency chart menggambarkan scenario sebagai persegi panjang dengan tepi yang membulat dan memiliki konektor berbentuk bulat. Gambar 2.12 adalah contoh dari *dependency diagram*.



Gambar 2.12 Contoh *Dependency Chart*

2.8. Pengembangan Aplikasi

SDLC (*System Development Life Cycle*) atau siklus hidup pengembangan sistem dalam rekayasa perangkat lunak adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut.

Salah satu model SDLC (*Software Development Life Cycle*) adalah model prototip. Model prototip menurut Howard adalah salah satu pendekatan dalam rekayasa perangkat lunak yang secara langsung mendemonstrasikan bagaimana perangkat lunak atau komponen-komponen perangkat lunak akan bekerja dalam lingkungannya sebelum tahapan konstruksi aktual dilakukan (Indra 2013).

Model *prototype* dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Agar pelanggan dapat membayangkan perangkat lunak yang akan dibuat, maka dibuatlah program *prototype* dengan yang apa yang diinginkan oleh pengguna. Program *prototype* dapat berupa tampilan sederhana dengan simulasi alur perangkat lunak. Program *prototype* ini akan di evaluasi oleh pengguna hingga sesuai dengan yang diinginkan oleh pengguna (A.S. and Shalahuddin 2013).

Cakupan tahap dari model *prototype* terdiri dari:

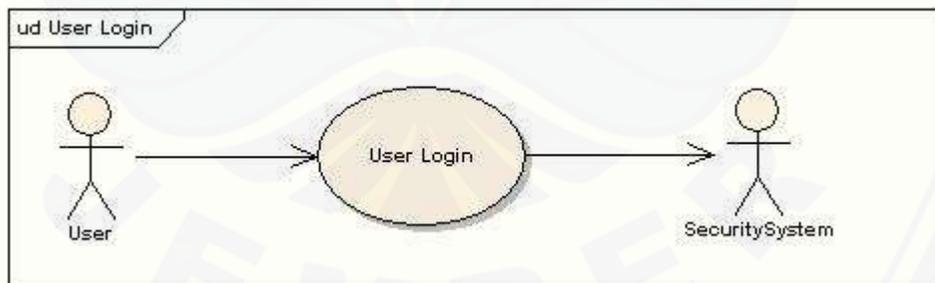
- Mendefinisikan objektif secara keseluruhan dan mengidentifikasi kebutuhan yang sudah diketahui.
- Melakukan perancangan secara cepat sebagai dasar untuk membuat *prototype*.
- Menguji coba dan mengevaluasi *prototype* dan kemudian melakukan penambahan dan perbaikan-perbaikan terhadap *prototype* yang sudah dibuat.

2.9. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman yang berorientasi objek. UML merupakan metode pengembangan perangkat lunak yang menggunakan metode grafis (Brigida 2013). Dalam penerapan UML terhadap UI meliputi :

2.9.1. Use Case Diagram

Use Case adalah alat untuk memodelkan perilaku suatu sistem informasi yang akan dibuat. *Use case* menggambarkan interaksi antar satu atau lebih aktor dengan sistem informasi yang akan dibuat sehingga dapat mengetahui fungsi yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi tersebut (A.S. and Shalahuddin 2013). Pada gambar 2.13 adalah contoh *use case*



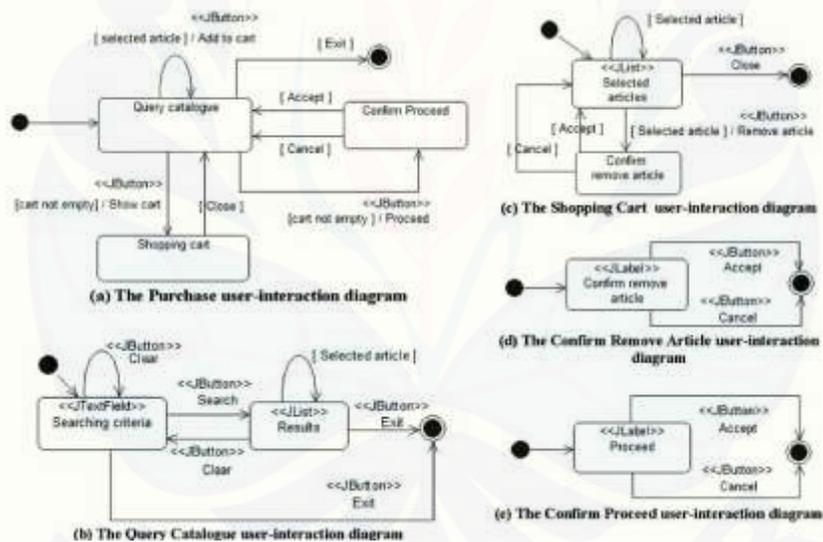
Gambar 2.13. Contoh *Use Case Diagram* (Almendros-Jimenez and Iribarne 2009).

2.9.2. Use Case Scenario

Skenario menggambarkan tabel yang digunakan untuk membuat dan menjelaskan keterangan secara lebih rinci mengenai setiap *usecase* yang digunakan.

2.9.3. User Interaction Diagram

User Interaction Diagram adalah bentuk diagram aktifitas yang menggambarkan aktifitas pengguna terhadap aplikasi. Diagram ini mirip dengan diagram aktivitas (A.S. and Shalahuddin 2013). Pada *activity diagram* akan menggambarkan alur kerja sistem secara garis besar sedangkan pada *user interaction diagram* menggambarkan alur kerja dari *user interface*. Pada gambar 2.14 adalah contoh *user interaction diagram*. *Interaction Diagram* memiliki beberapa simbol yaitu: *states*, *transition* dan *condition*. *States* menggambarkan *terminal state* dapat dilabeli dengan `<<textfield>>`, `<<jlist>>` dan `<<jlabel>>`. *Transition* dapat dilabeli dengan stereotip, kondisi atau keduanya. Kondisi dapat menggambarkan *user choices* atau *data logic* (Almendros-Jimenez and Iribarne 2009).

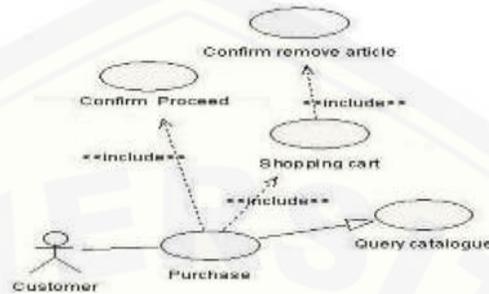


Gambar 2.14. Contoh *User Interaction Diagram* (Almendros-Jimenez and Iribarne 2009)

2.9.4. User Interface Diagram

User interface diagram adalah bentuk khusus dari *use case* yang ada pada sebelumnya. *User interface diagram* adalah bentuk perluasan dari *use case*. *User interface diagram* menambahkan beberapa *use case* dan untuk menghubungkan tiap

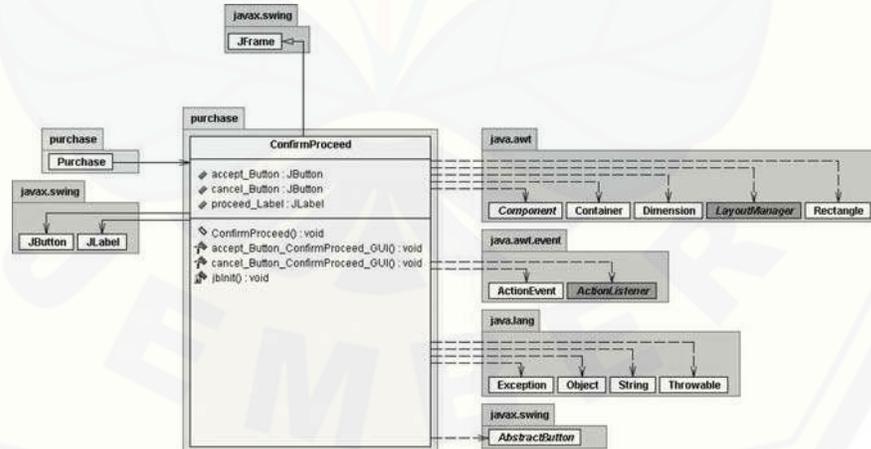
use case digunakan *include* dan *generalization* (Almendros-Jimenez and Iribarne 2009). Gambar 2.15 adalah contoh dari *user interface diagram*.



Gambar 2.15. Contoh *User Interface Diagram* (Almendros-Jimenez and Iribarne 2009)

2.9.5. GUI Class Diagram

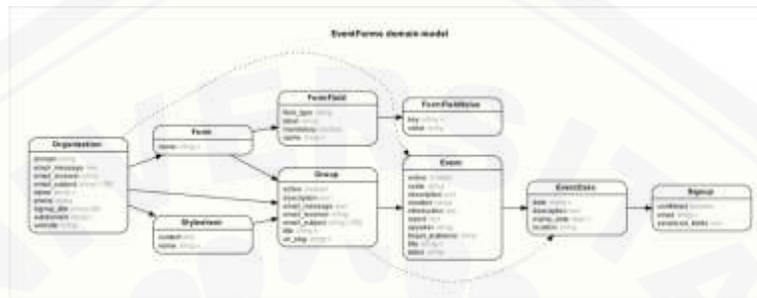
Class diagram adalah sebuah tool yang menggambarkan struktur sistem dari segi kelas yang akan ada pada sistem (A.S. and Shalahuddin 2013). *GUI class diagram* di dapat dari *use case* dan *user interaction diagram*. Pada gambar 2.16 adalah contoh dari *GUI class diagram*, dimana pada jendela didapat dari use case dan komponen *GUI* di dapat dari *user interaction diagram* (Almendros-Jimenez and Iribarne 2009).



Gambar 2.16. Contoh *GUI Class Diagram* (Almendros-Jimenez and Iribarne 2009)

2.9.6. Entity Relationship Diagram

Entity Relationship Diagram (ERD) menggambarkan notasi grafis dari pemodelan data. ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Gambar 2.17 adalah contoh dari ERD



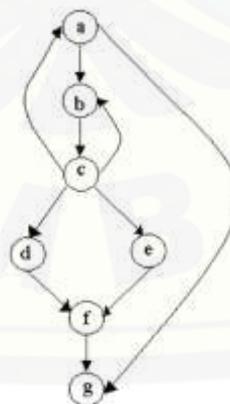
Gambar 2.17. Contoh *Entity Relationship Diagram*

2.10. Pengujian Aplikasi

Pengujian adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean. Sebuah perangkat lunak perlu dijaga kualitasnya sehingga tergantung kepada kepuasan pelanggan (A.S. and Shalahuddin 2013).

2.10.1. Pengujian *White Box*

Pengujian secara white box adalah pengujian yang mengukur kinerja sistem. Teknik pengujian yang digunakan yaitu pembuatan diagram alir, perhitungan siklomatik. Contoh perhitungan siklomatik terdapat pada gambar 2.18



Gambar 2.18. contoh perhitungan siklomatik

2.10.2. Pengujian *Black Box*

Pengujian ini akan menguji aplikasi secara fungsional dimana pengujian ini dilakukan dalam bentuk tertulis. Pengujian ini berfungsi untuk memeriksa apakah aplikasi tersebut telah berjalan sesuai dengan fungsinya. Pengujian ini juga meliputi seberapa baik sistem melaksanakan fungsinya, termasuk perintah yang dijalankan pengguna, pencarian dan proses eksekusi data, pengguna layar dan integrasi.

BAB 3. METODOLOGI PENELITIAN

Jenis penelitian ini adalah kualitatif, yaitu adalah penelitian tentang riset yang bersifat deskriptif dan cenderung menggunakan analisis. Sumber data akan diperoleh dari hasil pengamatan langsung di lapangan dan dicocokkan dengan data yang ada di google maps

Bagian ini akan menjelaskan tahapan penelitian dilakukan agar aplikasi dapat berjalan dengan baik. Langkah-langkah penelitian akan dilakukan sebagai berikut :

3.1. Tahap Perencanaan

Tahap awal dalam mengembangkan aplikasi ini dibagi menjadi dua tahap. Tahap tahap tersebut adalah :

3.1.1 Studi pustaka terhadap teori yang terkait

Tahap ini adalah tahap dimana penulis mengumpulkan studi pustaka yang berhubungan dengan pengembangan aplikasi ini.

3.1.2. Pengumpulan Data

Data didapatkan dengan cara terjun langsung ke lapangan (*Ground Check*) menggunakan *GPS*. Setelah melakukan *Ground Check* maka data tersebut akan dicocokkan posisinya dengan data yang telah ada di *Google Earth*. Setelah data tersebut cocok, maka data akan dimasukkan dalam *MySQL*.

3.2. Penentuan Metode dan Implementasi

Tahap ini peneliti akan menggunakan metode pendekatan scenario dan fungsi sebagai analisis pengambiln kebutuhan. Penelitian ini hanya menggunakan *Facet* sebagai bentuk matrik deskripsi pada objek penelitian. Bentuk matrik facet dapat dilihat di tabel 3.1. Metode yang akan digunakan dalam pengembangan *augmented reality* adalah menggunakan metode *markerless* yaitu *GPS-Based Markerless*.

Tabel 3.1. tabel *facet* pendekatan scenario

Komponen	Skenario
Aktor	<ul style="list-style-type: none"> • Profil • Peran • Tugas
Kronologi	
Sumber daya	
Aliran	<ul style="list-style-type: none"> • <i>Input</i> • <i>Ouput</i>
lingkungan	

3.3. Pengembangan aplikasi

Tahap pengembangan aplikasi ini akan menggunakan metode prototip untuk perancangan aplikasi. Tahapan yang digunakan sesuai dengan tahapan yang ada pada tinjauan pustaka. Adapun tahapannya adalah:

3.3.1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah proses menemukan kebutuhan atau kondisi yang harus dipenuhi dalam suatu sistem. Analisis kebutuhan sistem pada aplikasi ini menggunakan sebuah pendekatan yaitu pendekatan skenario dan fungsi.

3.3.2. Merancang Prototipe

Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pengguna (misalnya dengan membuat *input* dan format *output*)

3.3.3. Evaluasi Prototipe

Evaluasi ini dilakukan oleh Pengguna apakah prototyping yang sudah dibangun sudah sesuai dengan keinginan pengguna. Pada tahap ini, peneliti menggunakan konsep perancangan *Object Oriented Programming (OOP)*. Dalam penggambaran perancangan sistem menggunakan *Visual Paradigm*. Adapun data perancangannya meliputi :

- a. *Use Case*
- b. *Use Case Scenario*
- c. *User Interaction Diagram*
- d. *User Interface Diagram*
- e. *GUI Class Diagram*
- f. *Entity Relationship Diagram*

Jika sudah sesuai maka langkah d akan diambil. Jika tidak *prototyping* direvisi dengan mengulang langkah a, b, dan c.

3.3.4. Mengkodekan sistem

Dalam tahap ini prototyping yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai. Pada tahap ini peneliti akan membangun aplikasi berbasis *mobile* dengan android sebagai sistem operasinya. Dalam membangun aplikasi tersebut peneliti menggunakan bahasa *Java* dan *XML* sebagai dasar pemrograman android dan menggunakan manajemen data yaitu *DBMS MySQL*.

3.3.5. Menguji sistem

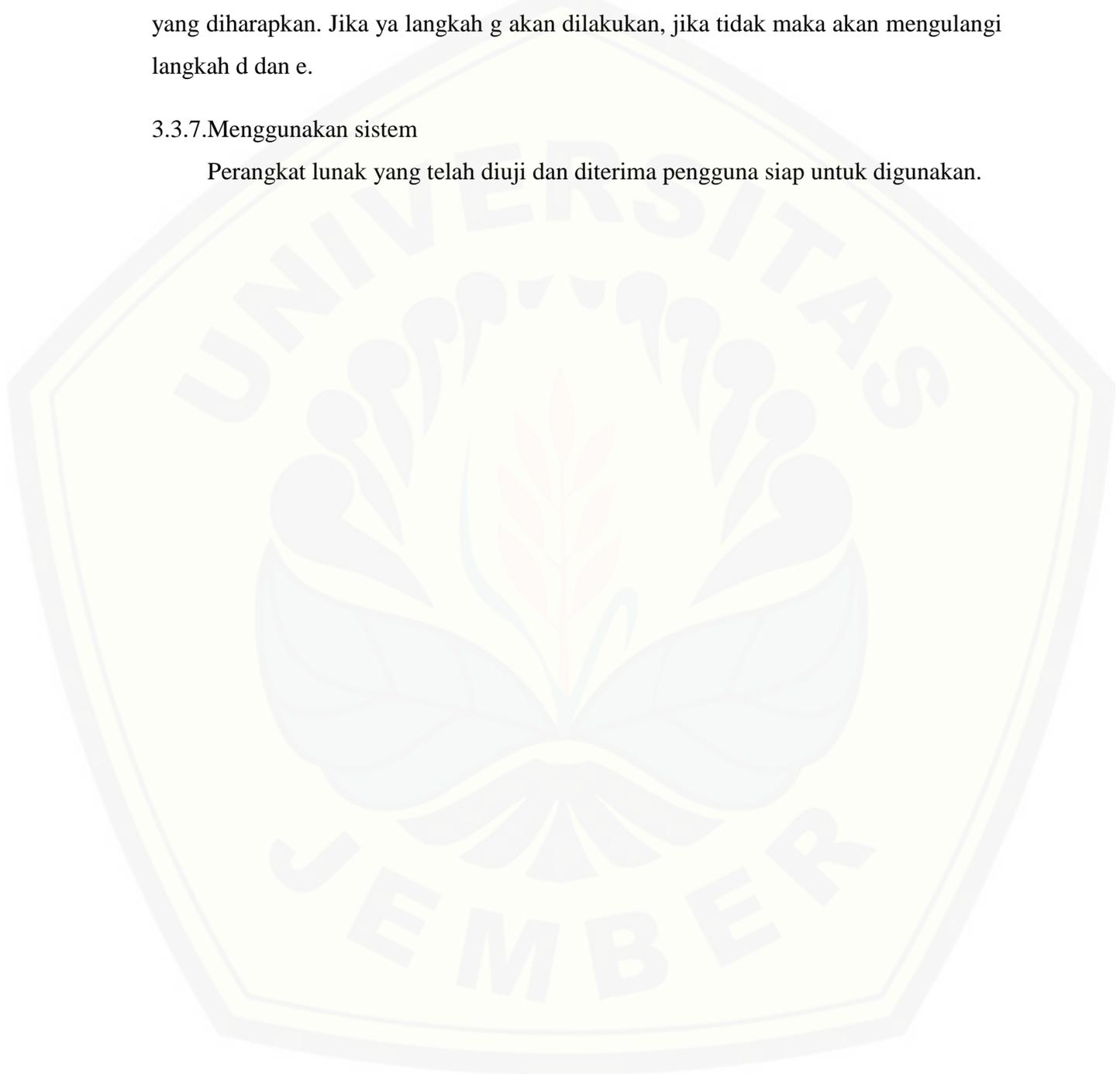
Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan untuk mengetahui sistem telah sesuai dengan desain yang diinginkan oleh user dan dalam menjalankan aplikasi tersebut sesuai dengan fungsinya tanpa ada kesalahan. Dalam pengujian sistem akan dilakukan secara *white box* dan *black box*.

3.3.6. Evaluasi Sistem

Pengguna mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya langkah g akan dilakukan, jika tidak maka akan mengulangi langkah d dan e.

3.3.7. Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima pengguna siap untuk digunakan.



BAB 4 . DESAIN DAN PERANCANGAN

Pada bab ini akan diuraikan tentang bagaimana proses perancangan aplikasi *Augmented Reality Mobile Navigation* di Universitas Jember menggunakan pendekatan skenario. Proses desain dan perancangan sistem yang dilakukan berdasarkan dengan data-data yang telah diperoleh.

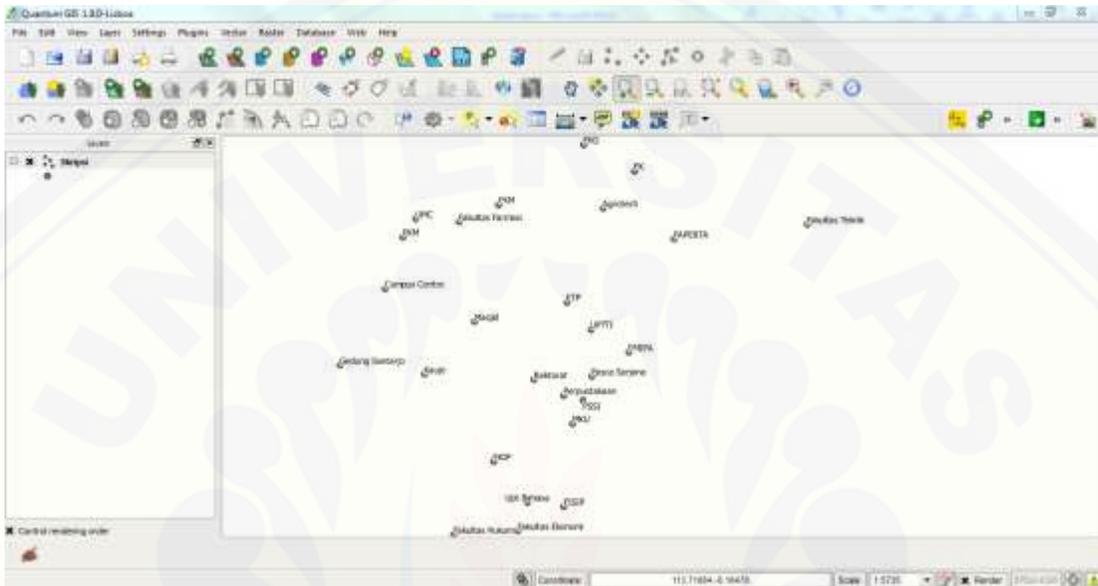
4.1. Deskripsi umum

Deskripsi umum aplikasi yang akan dibangun pada penelitian ini adalah aplikasi *ARMN* yang dapat membantu pengguna untuk mendapatkan rute tercepat yang akan ditempuh menuju sebuah gedung yang ada di Universitas Jember. Aplikasi ini akan menggunakan teknologi *augmented reality* yang akan menambahkan informasi arah di layar *smartphone*, sehingga akan memudahkan pengguna yang tidak dapat membaca peta dengan baik. Aplikasi ini berupa sebuah aplikasi *smartphone* dengan sistem operasi android yang sedang populer sehingga akan menghasilkan sebuah aplikasi dengan format .apk. aplikasi ini diharapkan dapat dijangkau oleh semua pihak yang sedang berada di Universitas Jember.

Aplikasi yang dibangun dalam penelitian ini adalah aplikasi *Augmented Reality Mobile Navigation* (ARMN) di lingkungan Universitas Jember. Aplikasi dapat memudahkan pengguna dalam mencari rute yang akan ditempuh untuk menuju lokasi yang diinginkan, dalam hal ini lokasi tersebut adalah gedung yang ada di lingkungan Universitas Jember. Aplikasi adalah sebuah aplikasi *smartphone* dengan android sebagai dasar sistem operasi pada *smartphone* tersebut.

Fungsi aplikasi ini telah diuraikan sebelumnya menggunakan pendekatan skenario, dimana aplikasi ini memiliki 2 fungsi yaitu fungsi umum dan fungsi khusus. Fungsi umum pada aplikasi ini adalah mampu memilih lokasi tujuan. mampu menampilkan informasi rute dengan *Augmented Reality* serta mampu menampilkan lokasi tujuan. Fungsi khusus pada aplikasi adalah dapat menerima *update* lokasi pengguna secara otomatis dan mampu mendeteksi lokasi pengguna.

Setelah data tersebut cocok, maka data akan di ekspor dalam bentuk .kml sehingga data dapat diimpor ke aplikasi *QuantumGis* untuk dapat diimport ke *MySQL* seperti terlihat pada gambar 4.2 dan 4.3.



Gambar 4.3. Titik-Titik Lokasi Gedung yang Berada di Universitas Jember

4.3. Identifikasi Kebutuhan

Identifikasi kebutuhan sistem pada aplikasi ini menggunakan pendekatan skenario untuk mendapatkan fungsi-fungsi yang diharapkan ada pada aplikasi tersebut.

a. Pendekatan Skenario

Penelitian kali ini peneliti menggunakan *facet* sebagai prosedur pendekatan skenario dan menggunakan 2 skenario. Skenario pada penelitian ini adalah sebagai berikut :

a. Skenario 1

Pak joko dan keluarga ingin mengunjungi henri yang sedang menempuh kuliah di jurusan Sistem Informasi. Pak joko kesulitan mencari letak gedung jurusan Sistem Informasi dikarenakan tidak adanya peta seluruh gedung yang ada di Universitas Jember. Pada skenario ini akan sangat potensial jika menambahkan navigasi yang menggunakan *Augmented Reality* guna menampilkan informasi rute ke dunia nyata. Dengan adanya *Augmented Reality* akan menambah kemudahann dalam penggunaan peta.

b. Skenario 2

Henri ingin bertemu dengan joko di gedung teknik, tetapi henri adalah mahasiswa baru di universitas jember sehingga tidak mengetahui lokasinya sekarang berada, sedangkan mereka harus bertemu dalam waktu 1 jam. Pada skenario ini joko ingin mengetahui lokasinya dan ketika joko mencari gedung Teknik, joko ingin terus mengetahui lokasinya secara otomatis.

Berdasarkan dua skenario ini dapat dibentuk menjadi sebuah tabel matrik yang dapat mempermudah peneliti untuk menentukan kebutuhan fungsional dari kedua skenario tersebut. Tabel matrik dapat dilihat pada tabel 4.1 di bawah ini.

Tabel 4.1. Tabel Matrik Skenario

Komponen	Skenario 1	Skenario 2
Aktor	<ul style="list-style-type: none"> • Profil • Peran • Tugas 	<ul style="list-style-type: none"> • Pak Joko dan Keluarga • Orang tua henri • Mencari gedung jurusan sistem informasi
Kronologi	<ul style="list-style-type: none"> • Pak joko berada di rektorat • Pak joko akan menuju ke PSSI 	<ul style="list-style-type: none"> • Henri dan joko • Mahasiswa baru • Ingin saling bertemu

	<ul style="list-style-type: none"> • Pak joko membuka aplikasi ARMN di <i>smartphonenya</i> • Menentukan lokasi yang akan dituju • Mengikuti rute peta yang telah disediakan • Mengikuti rute yang telah disediakan oleh peta tersebut • Menekan tombol menuju AR View • Mengikuti petunjuk arah yang telah disediakan 	<ul style="list-style-type: none"> • gedung PSSI • Henri akan menuju gedung teknik • Membuka aplikasi ARMN pada <i>smartphonenya</i> • Menentukan rute yang akan dituju • Mengikuti rute yang telah disediakan oleh peta tersebut • Menekan tombol menuju AR View • Mengikuti petunjuk arah yang telah disediakan
Sumber daya	<ul style="list-style-type: none"> • <i>Smartphone</i> 	<ul style="list-style-type: none"> • <i>Smartphone</i>
Aliran		
<ul style="list-style-type: none"> • <i>Input</i> • <i>Ouput</i> 	<ul style="list-style-type: none"> • Lokasi yang akan dituju • Rute yang akan dituju 	<ul style="list-style-type: none"> • Lokasi yang akan dituju • Rute yang akan dituju
lingkungan	<ul style="list-style-type: none"> • Universitas Jember 	<ul style="list-style-type: none"> • Universitas Jember

b. Fungsional

Berdasarkan hasil dari pendekatan skenario maka dapat disimpulkan beberapa fungsi yang harus ada dalam pengembangan aplikasi ini ada pada tabel 4.2.

Tabel 4.2. Tabel Fungsional

	Skenario 1	Skenario 2
Fungsi Umum	<ul style="list-style-type: none"> a. Dapat mengetahui lokasi pengguna saat ini b. Dapat memilih lokasi tujuan c. Menampilkan informasi rute dengan <i>Augmented Reality</i> 	
Fungsi Khusus	<ul style="list-style-type: none"> a. Menampilkan informasi rute dengan peta 	<ul style="list-style-type: none"> b. <i>Update</i> lokasi pengguna secara otomatis

Tabel ini mendefinisikan kebutuhan fungsional baik umum maupun khusus. Tabel ini juga dapat digunakan sebagai bahan pengujian pada tahapan pengujian aplikasi.

c. Fungsi-fungsi dalam Aplikasi *Augmented Reality Mobile Navigation*

Banyaknya fitur pada aplikasi *Augmented Reality Mobile Navigation* ini bisa dilihat dari tabel 4.2. Semua fitur tersebut bebas digunakan untuk siapapun karena tidak ada batasan untuk hak akses. Beberapa fitur yang disediakan adalah :

a. Dapat mengetahui lokasi pengguna

Merupakan fungsi awal aplikasi ini dimana pengguna akan secara otomatis dapat mengetahui posisi awal dimana pengguna berada.

b. Dapat memilih lokasi tujuan

Fitur ini mengharuskan pengguna untuk memilih lokasi tujuan yaitu gedung-gedung yang ada di lingkungan Universitas Jember. Lokasi tersebut akan dipilih dari daftar yang tersedia dalam sebuah *dropdown* yang berisi data gedung-gedung yang tersedia. Data pada *dropdown* tersebut akan di dapat dari *MySQL*.

c. Menampilkan informasi rute dengan *Augmented Reality*

Pada fitur ini pengguna akan melihat sebuah arah yang akan menuntun pengguna ke tujuan yang diinginkan, juga terdapat estimasi waktu dan jarak tempuh ke tujuan yang diinginkan. Pengguna juga akan dituntun secara nyata karena pada fitur ini menggunakan kamera yang hidup dan ditampilkan pada layar *smartphone*.

d. Menampilkan informasi rute dengan peta

Fitur ini akan menampilkan sebuah peta yang terdapat sebuah rute yang telah didapatkan dari fitur memilih tujuan.

e. *Update* lokasi pengguna secara otomatis

Fitur ini akan mencari *update* lokasi pengguna secara otomatis.

4.4. Perancangan *Prototype* dan Perancangan Aplikasi

Tahapan ini adalah tahapan perancangan *prototype* dan perancangan aplikasi. Perancangan *prototype* menggunakan *wireframe* untuk simulasi tampilan.

4.4.1. Perancangan *prototype*

Merancang *prototipe* adalah pembuatan tampilan sementara yang akan disajikan pada *user*. *Prototipe* yang disajikan merupakan hasil dari analisis kebutuhan yang telah didapat sebelumnya.

a. *Splash Screen*



Gambar 4.4. Tampilan *Splash Screen*

Gambar 4.4 merupakan tampilan *prototipe splash screen*. Tampilan ini akan muncul pada awal aplikasi dijalankan. Fungsi tampilan ini adalah sebagai tampilan awal aplikasi dan mengecek ketersediaan koneksi dan GPS sehingga aplikasi dapat dijalankan. Ketika pada proses pengecekan tidak terdapat koneksi dan GPS, maka akan muncul sebuah *Toast* untuk menuju halaman *setting*.

b. *Home*



Gambar 4.5. Tampilan Home

Gambar 4.5 adalah *prototipe* untuk halaman kedua. Pada fitur ini pengguna akan memilih lokasi tujuan. Fitur ini juga akan memberikan informasi kepada pengguna lokasi pengguna saat ini

c. *Map*



Gambar 4.6. Tampilan *Map*

Gambar 4.6 menunjukkan tampilan *prototipe* pada tampilan *map*. Prototipe ini berfungsi menunjukkan menampilkan map dari google map dan arah tujuan pengguna.

d. *AR View*



Gambar 4.7. Tampilan *AR View*

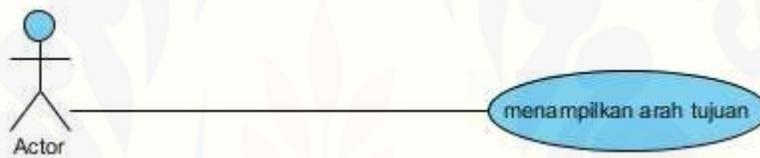
Gambar 4.7 menunjukkan prototipe untuk tampilan *AR View*. *AR View* ini akan berupa tampilan kamera dengan sebuah gambar penunjuk arah. Pada prototipe ini penunjuk arah akan menunjukkan arah yang sesuai dengan arah yang didapatkan dari *google maps*.

4.4.2. Perancangan Aplikasi.

Tahapan ini penulis menggunakan UML UI dalam perancangan aplikasi. Perancangan aplikasi ini dimulai dari *use case* hingga *entity relationship diagram*.

a. Use Case

Gambar 4.8 adalah *use case* dari aplikasi ARMN. Aplikasi ini memiliki satu *use case* yaitu menampilkan arah tujuan.



Gambar 4.8. Use Case ARMN

b. Use Case Scenario

Tabel 4.3 merupakan skenario dari *use case* aplikasi ARMN. *Use case scenario* ini didapat dari *use case* yang telah didefinisikan sebelumnya.

Tabel 4.3. Tabel Use Case Scenario

Actor		User
Entry Condition	User menekan aplikasi	
Exit Condition	User mengikuti petunjuk arah yang ada di AR View	
Skenario Normal		
NO	USER	SISTEM
1	Menekan tombol aplikasi	
2		Menampilkan loading screen

3	Mengecek ketersediaan GPS
4	Masuk ke halaman Home
5	Menampilkan <i>progress dialog getting location</i>
6	Sistem menampilkan halaman home dan lokasi
7	User memilih lokasi yang tersedia
8	User menekan tombol GO
9	Sistem menuju ke halaman map
10	Menampilkan <i>progress dialog getting location</i>
11	Menampilkan rute pada halaman map
12	User menekan tombol AR View
13	Menuju halaman AR View
14	Sistem menampilkan AR View

Skenario alternatif

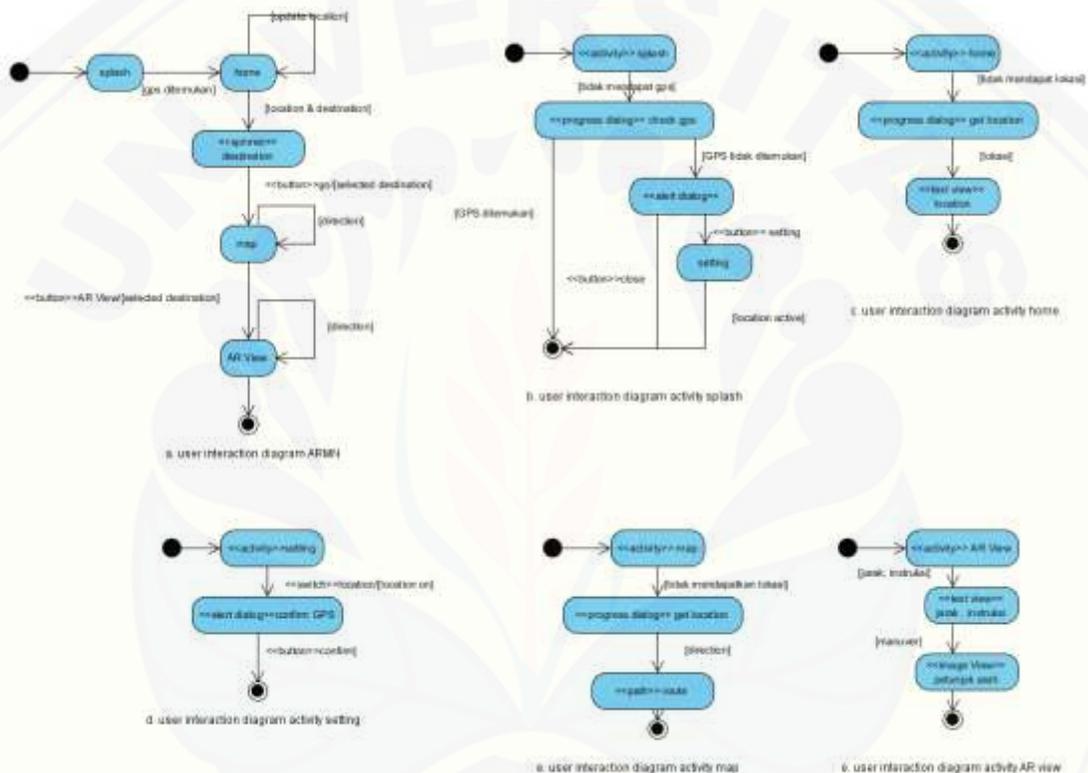
USER	SISTEM
3.1	Tidak menemukan ketersediaan GPS
3.2	Menampilkan <i>progress dialog</i> untuk menuju ke halaman setting lokasi
3.3	Menekan tombol GPS
3.4	Jika menekan ya
3.5	Menuju halaman setting lokasi
3.6	Menekan switch aktifkan gps
3.7	Keluar dari aplikasi
3.8	Jika menekan tidak
3.9	Keluar dari aplikasi

c. *User Interaction Diagram*

Tiap *use case diagram* akan digambarkan dalam sebuah *user interface diagram*. Gambar 4.9 adalah *user interaction diagram* pada aplikasi ini. Interaksi user pada aplikasi secara keseluruhan terlihat pada gambar a. aplikasi dimulai dari *state splash*. Pada *state splash* terdapat interaksi user pada state tersebut. Pada *state splash* dimulai dengan sebuah *activity splash*. Activity ini akan mencari GPS melalui sebuah *progress dialog* yaitu *check gps*. Pada *progress dialog* ini akan muncul dua kondisi yaitu jika menemukan GPS dan tidak menemukan GPS. Ketika aplikasi menemukan GPS, maka aplikasi akan melanjutkan ke *state home*. Ketika tidak menemukan GPS, maka akan muncul sebuah *alert dialog*. *alert dialog* memiliki 2 *button* yaitu *button setting* dan *button close*. Ketika menekan tombol *close* maka aplikasi akan menutup jika menekan tombol *setting*, maka akan ke *state setting*. Pada *state setting* terdapat sebuah *switch* yang akan mengaktifkan GPS. Setelah GPS aktif maka akan muncul sebuah *alert dialog* untuk mengonfirmasi GPS telah aktif.

Setelah *state splash* menemukan GPS, maka interaksi user akan diteruskan ke *state home*. *State home* akan diturunkan ke *activity home*. *Activity home* akan mengeluarkan sebuah *progress dialog* ketika tidak mendapatkan lokasi. *Progress dialog* akan mendapatkan lokasi pengguna dan lokasi tujuan. Setelah lokasi didapatkan, maka lokasi tersebut akan diletakkan di sebuah *textview* dan sebuah *spinner*. Setelah itu pengguna akan menekan sebuah tombol dengan lokasi yang telah di pilih dan menuju ke *state map*. *State map* akan diturunkan ke *activity map*. Pada *activity map* juga tidak menemukan lokasi untuk menjadi *marker* pada *map*. *Map* akan menampilkan sebuah *progress dialog* untuk mendapatkan lokasi dan *direction*. Setelah mendapatkan lokasi dan *direction*, maka lokasi akan menjadi *marker* dan *direction* akan menjadi *path* yang mengoverlay *activity map*. Pengguna akan menekan tombol *ar view* dan akan menuju ke *state ar view*.

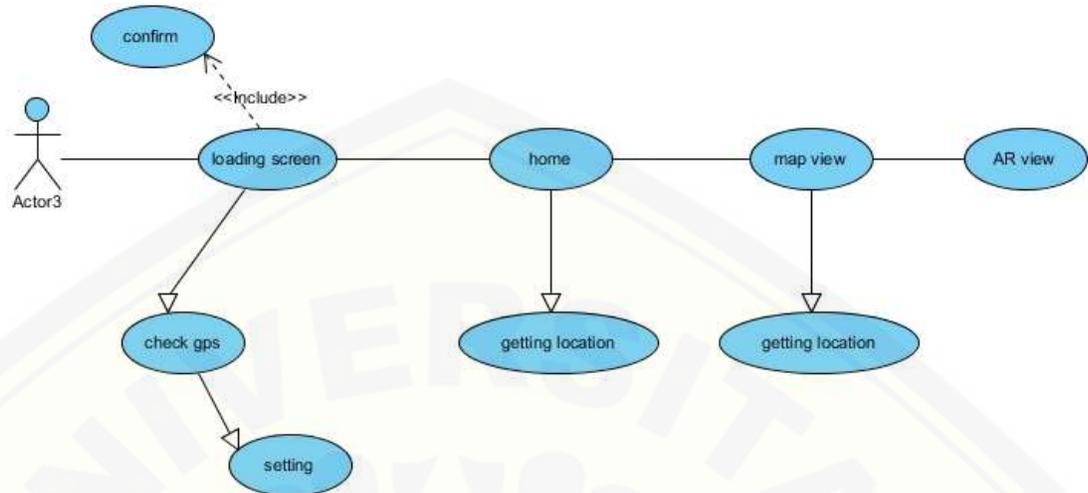
State ar view akan diturunkan ke *activity ar view*. Activity ini akan terus mengambil data dari *google map server*. Aplikasi ini akan menampilkan *textview* jarak dan instruksi yang diperoleh dari data *google map server*. aplikasi juga akan mengambil sebuah *tag maneuver* yang akan mengubah penunjuk arah yang sesuai dengan *tag* yang diterima oleh aplikasi.



Gambar 4.9. Gambar *User Interaction Diagram* Aplikasi ARMN

d. *User Interface Diagram*

Gambar 4.10 adalah *user interface diagram* dari aplikasi ini. Tahapan ini merupakan lanjutan dari tahapan *user interaction diagram*. Diagram ini berdasarkan diagram sebelumnya yaitu *use case diagram* dan *user interaction diagram*. Pada *User Interface Diagram* komponen yang digunakan sama dengan *Use Case* dan ditambahkan *State* dari *User Interaction Diagram*.

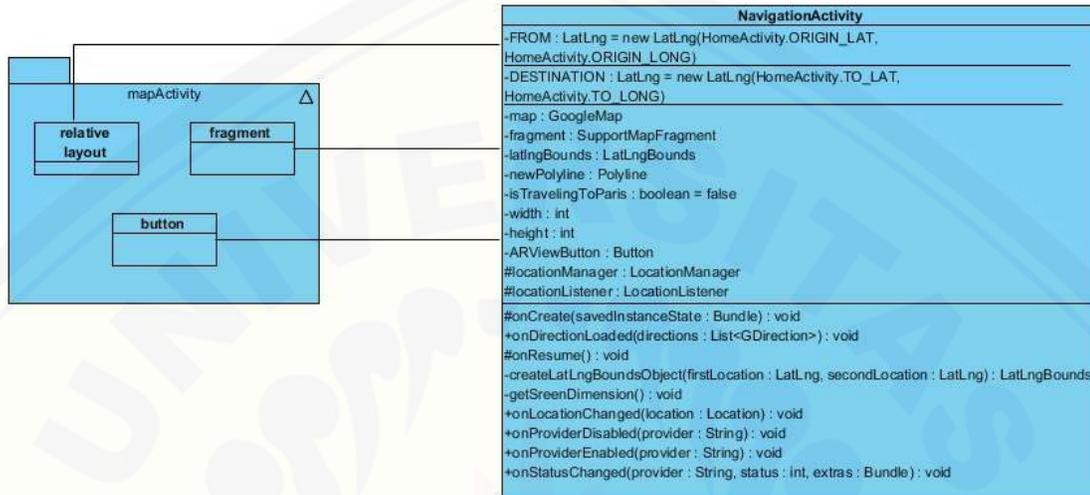


Gambar 4.10 User Interface Diagram dari Aplikasi ARMN

e. User Interface Class Diagram

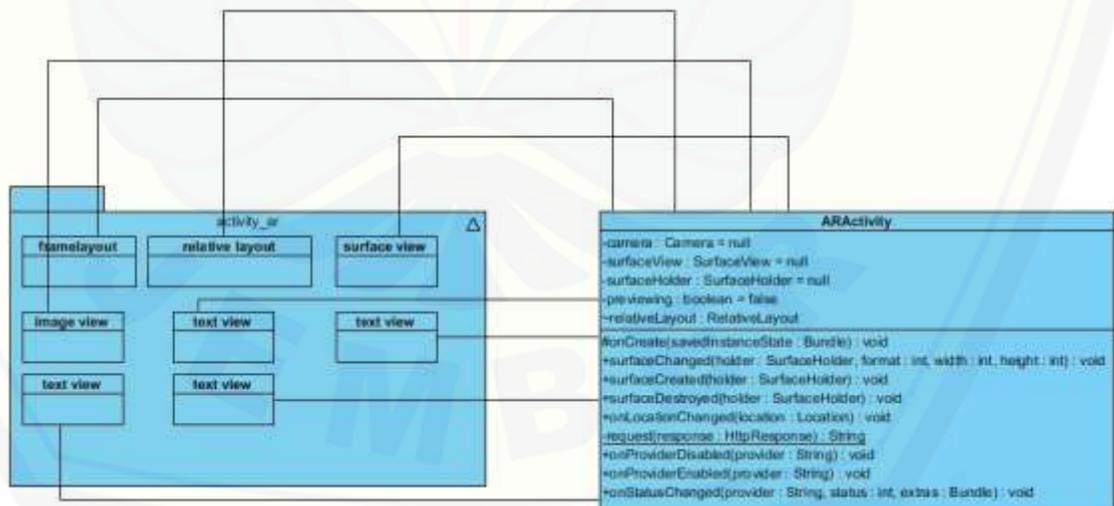
Tahap selanjutnya dari perancangan ini adalah perancangan User Interface Class Diagram. Komponen user interface dari aplikasi ini didapat dari *layout xml* dari android. Pada gambar 4.11 terdapat class *splash activity* dengan komponen yang akan di muat pada *splash activity*. *Splash activity* akan memuat sebuah *relative layout*. *Splash activity* juga akan memuat sebuah tampilan *alert dialog* yang memuat 2 button dan satu label jika pada aplikasi tidak menemukan ketersediaan *GPS*. *Splash activity* juga memiliki sebuah progress dialog yaitu *check gps* untuk mengecek ketersediaan *GPS*. Pada *progress dialog check gps* ini terdapat 2 *text view* dan satu *image view*. Ketika *GPS* tidak ditemukan maka akan diarahkan ke halaman *setting* dimana pada halaman ini memuat *text view* dan sebuah *switch*.

Class diagram selanjutnya adalah *class* dari *Navigation* activity. *Class* ini memuat beberapa komponen yaitu *relative layout*, *fragment* dan *button*. Gambar 4.13 menunjukkan *class diagram* pada *class home activity*.



Gambar 4.13. *Class Diagram Navigation Activity*

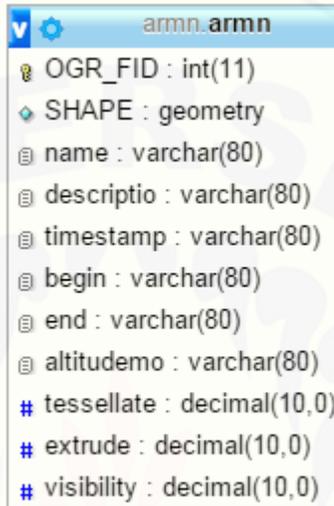
Gambar 4.14 menunjukkan gambar *class diagram* pada *AR Activity*. *AR Activity* memuat beberapa komponen yaitu *framelayout*, *relative layout*, *surface view*, *image view*, dan 4 *text view*. Semua komponen tersebut harus dimuat oleh *AR Activity*.



Gambar 4.14. *Class Diagram Ar Activity*

f. *Entity Relationship Diagram*

Tahap selanjutnya adalah *entity relationship diagram (ERD)*. ERD pada aplikasi ini ditunjukkan pada gambar 4.15.



Gambar 4.15. *Entity Relationship Diagram* dari Aplikasi ARMN

BAB 5 . HASIL DAN PEMBAHASAN

Bab ini akan menjelaskan tahapan selanjutnya dari model prototip dari pengembangan aplikasi *Augmented Reality Mobile Navigation*.

5.1 Evaluasi *Prototype*

Tahap ini akan mengevaluasi *prototype*. *Prototype* akan dievaluasi sesuai dengan fungsi yang telah dikumpulkan pada identifikasi kebutuhan. Pengembangan aplikasi ini dilakukan hingga 3 kali *prototype*. Seluruh pengembangan akan dilampirkan di halaman lampiran 1.A.

5.2. Implementasi Aplikasi

Tahap ini penulis mulai mengerjakan pembuatan aplikasi, tahap ini akan menjelaskan pula hasil dari aplikasi ARMN. Peneliti menggunakan bahasa *JAVA* dan *XML* dalam pengembangan aplikasi. Tahap ini akan menghasilkan beberapa *interface* dari aplikasi ARMN. Interface hasil dari pengembangan aplikasi ini akan dijelaskan sebagai berikut.

a. Splash Screen

Tampilan *splash screen* adalah tampilan yang berada pada awal aplikasi dijalankan.gambar 5.1 menunjukkan tampilan pada aplikasi ketika awal dijalankan. Tampilan splash screen akan mengecek ketersediaan *GPS*.



Gambar 5.1. Tampilan *Splash Screen*

Gambar 5.2 menunjukkan *source code* dari *activity splash screen*. Activity ini akan berupa tampilan *full screen* yang di *delay* selama 3 detik. Ketika *activity splash screen* telah berjalan 3 detik, maka aplikasi akan menjalankan sebuah inner class dengan yang meng-*extend* sebuah *class AsyncTask* dari *SDK android*. Kelas *checkGPS* yang menjadi *inner class* dari *class splashactivity* ini berfungsi untuk mengecek ketersediaan GPS. Class *checkGPS* mewarisi 3 *method* dari *class AsyncTask*.

Fungsi *class AsyncTask* adalah mengerjakan perintah secara *background* sehingga pada activity utama tidak terlihat sedang mengerjakan perintah. Ketika *checkGPS* dijalankan maka perintah yang ada dalam *method preexecute* akan dijalankan. *Method preexecute* memuat perintah menjalankan *progress dialog* seperti pada gambar 5.3, selagi menjalankan *progress dialog*, *class checkGPS* akan menjalankan *method doInBackground*.

Method doInBackground akan mengecek apakah GPS pada *smartphone* tersebut aktif atau tidak. Jika GPS aktif maka status akan berisi *true* jika GPS tidak aktif maka

status akan berisi false. Setelah mengerjakan *doinbackground* maka akan dilanjutkan pada *method onpostexecute*.

Method onpostexecute akan menerima status yang telah dihasilkan oleh *method doinbackground*. Ketika status tersebut bernilai *true*, maka aplikasi akan menuju home activity. ketika status tersebut bernilai *false*, maka aplikasi akan menampilkan sebuah *alert dialog* yang mengarahkan pengguna ke halaman setting jika menekan tombol *setting* dan akan menutup aplikasi jika menekan tombol *close* seperti pada gambar 5.4. Gambar 5.5 adalah halaman *setting* lokasi jika aplikasi tidak menemukan *GPS* yang aktif pada *smartphone* tersebut. *Method* ini juga menghilangkan kerja dari *progress dialog*

```
public class SplashActivity extends Activity {  
    ProgressDialog pd;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        requestWindowFeature(Window.FEATURE_NO_TITLE);  
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
            WindowManager.LayoutParams.FLAG_FULLSCREEN);  
        setContentView(R.layout.activity_splash);  
        new Handler().postDelayed(new Runnable() {  
  
            @Override  
            public void run() {  
                // TODO Auto-generated method stub  
                new checkGPS().execute();  
            }  
        }, 3000);  
    }  
}
```

```
private class checkGPS extends AsyncTask<Void, Void, Void>{

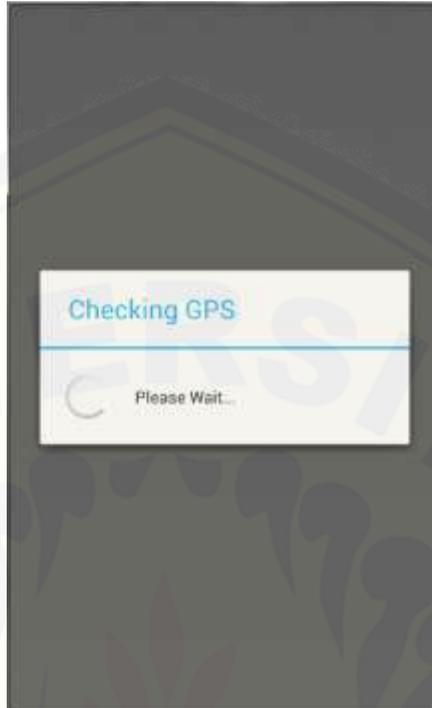
    Boolean status ;
    String provider = Settings.Secure.getString(getContentResolver(),
        Settings.Secure.LOCATION_PROVIDERS_ALLOWED);
    @Override
    protected void onPreExecute() {
        // TODO Auto-generated method stub
        super.onPreExecute();
        pd = ProgressDialog.show(SplashActivity.this, "Checking GPS", "Please Wait...");
    }
    @Override
    protected void doInBackground(Void... params) {
        // TODO Auto-generated method stub
        try {
            Thread.sleep(5000);
            if (!provider.equals("")) {
                status = true;
            }else {
                status = false;
            }
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        return null;
    }
    @Override
    protected void onPostExecute(Void result) {
        // TODO Auto-generated method stub
        super.onPostExecute(result);
        pd.dismiss();
        if (status == true) {
            startActivity(new Intent(SplashActivity.this, HomeActivity.class));
        }else {
            new AlertDialog.Builder(SplashActivity.this)
                .setTitle("GPS is settings")
                .setMessage("GPS is not enabled. Do you want to go to settings menu?")
                .setPositiveButton("Setting", new OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        // TODO Auto-generated method stub
                        startActivity(new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS));
                        finish();
                    }
                }).setNegativeButton("cancel", new OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        // TODO Auto-generated method stub
                        finish();
                    }
                }).show();
        }
    }
}
```

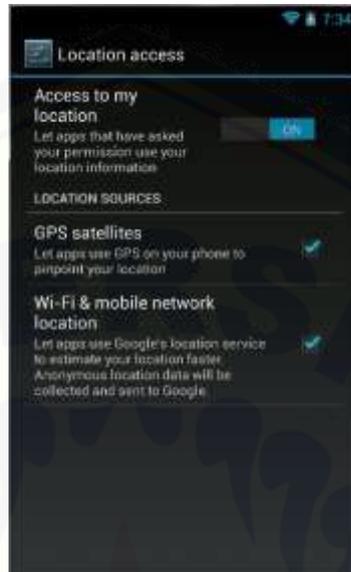
Gambar 5.2. Source Code dari Splash Activity



Gambar 5.3. *Progress Dialog Checkgps*



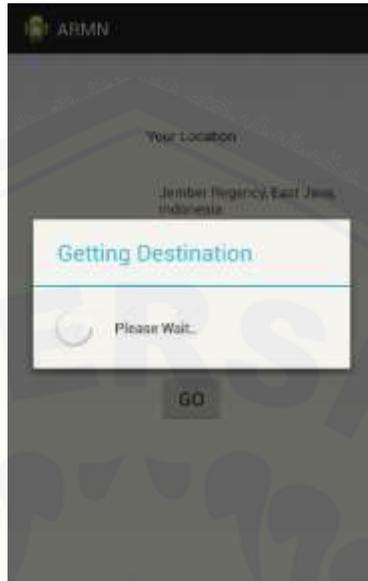
Gambar 5.4. *Alert Dialog Ketika Gps Tidak Aktif*



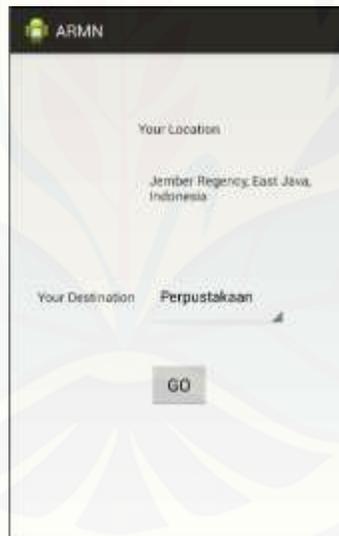
Gambar 5.5. Tampilan *Setting*

b. Home Activity

Activity selanjutnya adalah *Home Activity*. *Home activity* akan dimulai dengan sebuah *progress dialog* seperti pada gambar 5.6. *Progress dialog* ini berfungsi untuk mengambil lokasi awal dari *GPS* dan mengambil data lokasi tujuan yang akan diletakkan pada sebuah *spinner*. Lokasi awal adalah lokasi yang didapatkan dari perhitungan minimal 3 *BTS*. Gambar 5.7 adalah tampilan *home activity* setelah *progress dialog* selesai mengambil lokasi pengguna dan lokasi tujuan. *Home activity* akan menunjukkan lokasi pengguna dan sebuah *spinner*. *Spinner* ini akan dipilih oleh pengguna dan berisi lokasi gedung Universitas Jember yang menjadi tujuan dari pengguna. Setelah memilih lokasi tujuan maka pengguna akan menekan tombol *go*. Tombol *go* akan mengarahkan pengguna ke *navigation activity*.



Gambar 5.6. *Progress Dialog* Mengambil Lokasi



Gambar 5.7. Tampilan *Home Activity*

Gambar 5.8 adalah *source code* dari *home activity*. *Home activity* akan menjalankan dua kelas yang meng-*extends class AsyncTask* yaitu *gettingLocation* dan *downloadJSON*. Kelas *gettingLocation* mewarisi kelas *AsyncTask* dan juga

menggunakan interface `locationListener`. Kelas `gettingLocation` adalah kelas yang berisi sebuah *progress dialog* yang berada pada *method onpreexecute*.

Method doinbackground pada kelas `gettingLocation` berfungsi untuk mengambil lokasi pengguna dalam bentuk *latitude* dan *longitude*. Setelah lokasi pengguna didapatkan maka lokasi *latitude* dan *longitudenya* akan dikirimkan ke *google maps server* untuk mendapatkan alamat dari lokasi tersebut.

Method onpostexecuted akan dikerjakan setelah *method doinbackground* selesai. *Method onpostexecuted* akan meletakkan alamat pada *textview* yang telah tersedia. Setelah perintah tersebut telah dikerjakan, maka kelas `downloadJson` akan dikerjakan.

Kelas `downloadJSON` adalah kelas yang berfungsi untuk mengambil data yang ada pada *database ARMN*. Kelas ini akan menerima data dari *database* dalam bentuk *JSON*. Data tersebut akan di *parsing* di *method doinbackground*. Data tersebut akan ditempatkan di sebuah *spinner* yang ketika *spinner* tersebut dipilih maka akan mengisi *variabel latitude* dan *longitude*. setelah memilih *spinner* lokasi yang diinginkan, pengguna akan menekan tombol *go*. Tombol ini juga berfungsi untuk mengirimkan *variabel latitude,longitude* lokasi sementara dan *variabel latitude,longitude* lokasi tujuan.

```
public class HomeActivity extends FragmentActivity {
    TextView local;
    Button go;
    final Context context = this;
    JSONObject jsonobject;
    JSONArray jsonarray;
    ArrayList<String> dataList;
    ArrayList<Data> data;
    Spinner destination;
    ProgressDialog pd;
    Geocoder GeocoderAsync;
    double latAsync = 0.0;
    double lonAsync = 0.0;
    String Address = "";

    Double ORIGIN_LAT, ORIGIN_LONG, TO_LAT, TO_LONG;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        local = (TextView) findViewById(R.id.Location);
        go = (Button) findViewById(R.id.go);
        destination = (Spinner) findViewById(R.id.destination);
        new GettingLocation(HomeActivity.this).execute();
        new DownloadJSON().execute();

        go.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent i = new Intent(HomeActivity.this,NavigationActivity.class);
                i.putExtra("ORIGIN_LAT", ORIGIN_LAT);
                i.putExtra("ORIGIN_LONG", ORIGIN_LONG);
                i.putExtra("TO_LAT", TO_LAT);
                i.putExtra("TO_LONG", TO_LONG);
                startActivity(i);
            }
        });
    }

    private class DownloadJSON extends AsyncTask<Void, Void, Void> {

        @Override
        protected void onPreExecute() {
            // TODO Auto-generated method stub
            super.onPreExecute();
            pd = ProgressDialog.show(HomeActivity.this, "Getting Destination",
                "Please Wait..");
        }
    }
}
```

```
@Override
protected void doInBackground(Void... params) {
    // Locate the WorldPopulation Class
    data = new ArrayList<Data>();
    // Create an array to populate the spinner
    dataList = new ArrayList<String>();
    // JSON file URL address
    jsonobject = JSONfunctions
        .getJSONfromURL("http://syafiq.codemiring.com/index.php/welcome/getdata");

    try {
        // Locate the NodeList name
        jsonarray = jsonobject.getJSONArray("data");
        for (int i = 0; i < jsonarray.length(); i++) {
            jsonobject = jsonarray.getJSONObject(i);

            Data worldpop = new Data();

            worldpop.setName(jsonobject.optString("name"));
            worldpop.setLat(jsonobject.optString("lat"));
            worldpop.setLng(jsonobject.optString("lng"));
            data.add(worldpop);

            // Populate spinner with country names
            dataList.add(jsonobject.optString("name"));
        }
    } catch (Exception e) {
        Log.e("Error", e.getMessage());
        e.printStackTrace();
    }
    return null;
}

@Override
@Override
protected void onPostExecute(Void args) {

    pd.dismiss();
    // Locate the spinner in activity_main.xml
    Spinner mySpinner = (Spinner) findViewById(R.id.destination);

    // Spinner adapter
    mySpinner.setAdapter(new ArrayAdapter<String>(HomeActivity.this,
        android.R.layout.simple_spinner_dropdown_item, dataList));

    // Spinner on item click listener
    mySpinner
        .setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

            @Override
            public void onItemSelected(AdapterView<?> arg0,
                View arg1, int position, long arg3) {
                // TODO Auto-generated method stub

                TO_LAT = Double.parseDouble(data.get(position)
                    .getLat());
                TO_LONG = Double.parseDouble(data.get(position)
                    .getLng());

                Log.i("to", "" + TO_LAT + " " + TO_LONG);
            }

            @Override
            public void onNothingSelected(AdapterView<?> arg0) {
                // TODO Auto-generated method stub
            }
        });
}
}
```

```

private void getAddress(Context context, double lat, double lng) {
    List<Address> addresses = null;
    GeocoderAsync = new Geocoder(context, Locale.getDefault());
    try {
        addresses = GeocoderAsync.getFromLocation(lat, lng, 1);

        String address = addresses.get(0).getAddressLine(0);
        String city = addresses.get(0).getAddressLine(1);
        String country = addresses.get(0).getCountryName();
        Address = Html.fromHtml(address + ", " + city + "<br>" + country)
            .toString();
    } catch (Exception e) {
        e.printStackTrace();
        Address = "Refresh for the address";
    }
}

private class GettingLocation extends AsyncTask<Void, Void, Void> implements
    LocationListener {
    private Context ContextAsync;

    public GettingLocation(Context context) {
        this.ContextAsync = context;
    }

    Dialog progress;
    private String providerAsync;
    private LocationManager locationManagerAsync;

    String thikanaAsync = "Scan sms for location";

    Geocoder GeocoderAsync;

    Location location;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progress = ProgressDialog.show(ContextAsync, "Loading data",
            "Please wait...");
    }

    @Override
    protected Void doInBackground(Void... arg0) {
        // TODO Auto-generated method stub
        locationManagerAsync = (LocationManager) ContextAsync
            .getSystemService(ContextAsync.LOCATION_SERVICE);

        Criteria criteria = new Criteria();
        criteria.setAccuracy(Criteria.ACCURACY_COARSE);
        criteria.setCostAllowed(false);
        criteria.setPowerRequirement(Criteria.NO_REQUIREMENT);
        providerAsync = locationManagerAsync.getBestProvider(criteria,
            false);

        if (locationManagerAsync
            .isProviderEnabled(LocationManager.GPS_PROVIDER)) {
            providerAsync = LocationManager.GPS_PROVIDER;
        } else if (locationManagerAsync
            .isProviderEnabled(LocationManager.NETWORK_PROVIDER)) {
            providerAsync = LocationManager.NETWORK_PROVIDER;
        } else if (locationManagerAsync
            .isProviderEnabled(LocationManager.PASSIVE_PROVIDER)) {
            providerAsync = LocationManager.PASSIVE_PROVIDER;
        }

        location = locationManagerAsync.getLastKnownLocation(providerAsync);
        if (location != null) {
            latAsync = location.getLatitude();
            lonAsync = location.getLongitude();
        }
    }
}

```

```

        } else {
        }

        getAddress(ContextAsync, latAsync, lonAsync);

        return null;
    }

    @Override
    protected void onPostExecute(Void result) {

        super.onPostExecute(result);
        progress.dismiss();
        onLocationChanged(location);
        Log.v("latAsync_lonAsync", latAsync + "_" + lonAsync);
        ORIGIN_LAT = latAsync;
        ORIGIN_LONG = lonAsync;
        String latitude = Double.toString(latAsync);
        String longitude = Double.toString(lonAsync);
        local.setText(Address);

    }

    @Override
    public void onLocationChanged(Location location) {
        // TODO Auto-generated method stub
        locationManagerAsync.requestLocationUpdates(providerAsync, 0, 0,
            this);
    }

    @Override
    public void onProviderDisabled(String provider) {
        // TODO Auto-generated method stub
    }

    -
    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub
    }

    }
}

```

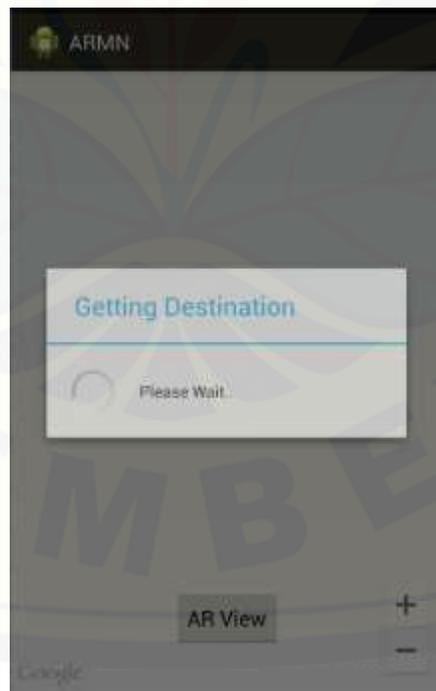
Gambar 5.8. *source code home activity*

c. Navigation Activity

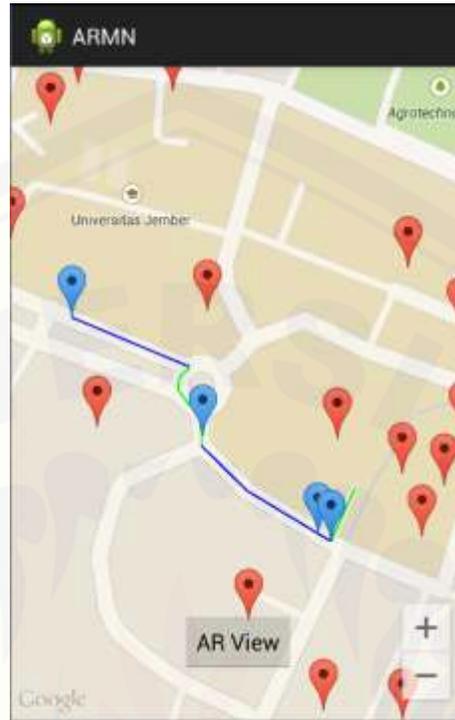
Activity ini berfungsi untuk menampilkan sebuah map dengan lokasi tujuan dalam bentuk point dan sebuah rute dari lokasi pengguna menuju lokasi tujuan. *Activity* ini dimulai dengan sebuah progress dialog yang berfungsi mengambil lokasi tujuan yang terlihat pada gambar 5.9.

Gambar 5.10 adalah tampilan dari *navigation activity* yang berupa *map*. Pada *map* tersebut terdapat sebuah rute dari lokasi awal pengguna menuju lokasi tujuan dan titik titik lokasi tujuan. Titik titik tersebut akan menampilkan informasi berupa nama gedung yang ada di Universitas Jember jika titik titik tersebut di tekan.

Source code pada *navigation activity* ditunjukkan pada gambar 5.11. kelas *navigation activity* berfungsi untuk menampilkan sebuah *map*, rute dan titik titik lokasi. Kelas *navigation activity* juga akan menjalankan sebuah inner class yaitu kelas *downloadJson*. Kelas ini juga akan memiliki sebuah *button* yang akan mengarahkan pengguna menuju halaman AR Activity. Kelas *downloadJSON* hampir sama dengan kelas *downloadJSON* yang ada pada kelas *home activity*. Pada *navigation activity* data tersebut tidak ditempatkan pada sebuah *spinner*, tetapi ditempatkan pada sebuah method dari variabel *map* yaitu *add marker*. Setelah data tersebut ditempatkan pada variabel tersebut, maka *map* akan menampilkan titik titik lokasi gedung di Universitas Jember.



Gambar 5.9. *Progress Dialog* Pada *Navigation Activity*



Gambar 5.10. Tampilan *Map* dan *Rute* Pada *Navigation Activity*

```
public class NavigationActivity extends FragmentActivity implements DCACallback{

    LatLng FROM ,DESTINATION ;
    double origin_lat,origin_long,to_lat,to_long;
    private GoogleMap map;
    private SupportMapFragment fragment;
    private LatLngBounds latlngBounds;
    private Polyline newPolyline;
    private int width, height;
    private Button ARViewButton;
    protected LocationManager locationManager;
    protected LocationListener locationListener;
    ArrayList<String> nama;
    ArrayList<Double> lat,lng;
    ArrayList<Data> data;
    JSONObject jsonobject;
    JSONArray jsonarray;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_map);
        getScreenDimension();
        fragment = ((SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map));
        map = fragment.getMap();

        ARViewButton = (Button) findViewById(R.id.ARViewButton);
        locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 2000, 2, this);

        Bundle extras = getIntent().getExtras();
        origin_lat = extras.getDouble("ORIGIN_LAT");
        origin_long = extras.getDouble("ORIGIN_LONG");
        to_lat = extras.getDouble("TO_LAT");
        to_long = extras.getDouble("TO_LONG");
        FROM = new LatLng(origin_lat,origin_long);
```

```

DESTINATION = new LatLng(to_lat,to_long);
new DownloadJSON().execute();
GDirectionsApiUtils.getDirection(this, FROM, DESTINATION,
    GDirectionsApiUtils.MODE_WALKING);
ARViewButton.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent i = new Intent(NavigationActivity.this,ARActivity.class);
        i.putExtra("to_lat", to_lat);
        i.putExtra("to_long", to_long);
        startActivity(i);
    }
});
}

@Override
public void onDirectionLoaded(List<GDirection> directions) {
    // TODO Auto-generated method stub
    for (GDirection direction : directions) {
        Log.e("MainActivity",
            "onDirectionLoaded : Draw GDirections Called with path "
                + directions);
        GDirectionsApiUtils.drawGDirection(direction, map);
    }
}

@Override
protected void onResume() {
    super.onResume();
    latLngBounds = createLatLngBoundsObject(FROM, DESTINATION);
    map.moveCamera(CameraUpdateFactory.newLatLngBounds(latLngBounds, width,
        height, 150));
}
}

```

```

private LatLngBounds createLatLngBoundsObject(LatLng firstLocation,
    LatLng secondLocation) {
    if (firstLocation != null && secondLocation != null) {
        LatLngBounds.Builder builder = new LatLngBounds.Builder();
        builder.include(firstLocation).include(secondLocation);

        return builder.build();
    }
    return null;
}

private void getSreenDimension() {
    Display display = getWindowManager().getDefaultDisplay();
    width = display.getWidth();
    height = display.getHeight();
}

private class DownloadJSON extends AsyncTask<Void, Void, Void> {
    ProgressDialog pd;
    @Override
    protected void onPreExecute() {
        // TODO Auto-generated method stub
        super.onPreExecute();
        pd = ProgressDialog.show(NavigationActivity.this, "Getting Destination",
            "Please Wait..");
    }
}

```

```

@Override
protected void doInBackground(Void... params) {
    // Locate the WorldPopulation Class
    data = new ArrayList<Data>();
    // Create an array to populate the spinner
    nama = new ArrayList<String>();
    // JSON file URL address
    jsonobject = JSONfunctions
        .getJSONfromURL("http://syafiq.codemiring.com/index.php/welcome/getdata");

    try {
        // Locate the NodeList name
        jsonarray = jsonobject.getJSONArray("data");
        for (int i = 0; i < jsonarray.length(); i++) {
            jsonobject = jsonarray.getJSONObject(i);

            Data worldpop = new Data();

            worldpop.setName(jsonobject.optString("name"));
            worldpop.setLat(jsonobject.optString("lat"));
            worldpop.setLng(jsonobject.optString("lng"));
            data.add(worldpop);

            // Populate spinner with country names
            nama.add(jsonobject.optString("name"));
        }
    } catch (Exception e) {
        Log.e("Error", e.getMessage());
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(Void args) {
    double lat,lng;
    pd.dismiss();
    // Locate the spinner in activity_main.xml
    for (int i = 0; i < data.size(); i++) {
        lat = Double.parseDouble(data.get(i).getLat());
        lng = Double.parseDouble(data.get(i).getLng());
        map.addMarker(new MarkerOptions()
            .position(new LatLng(lat,lng))
            .title(data.get(i).getName()));

        Log.i("data",data.get(i).getName());
    }
}
}
}

```

Gambar 5.11. Source Code Dari Navigation Activity

d. AR Activity

Gambar 5.12 menunjukkan tampilan pada *AR Activity*. Tampilan *AR Activity* berupa tampilan dari kamera yang mendapatkan tambahan informasi seperti penunjuk arah, jarak dan perintah navigasi. Pada *AR Activity* penunjuk arah tersebut dapat berubah arah sesuai dengan arah yang akan ditempuh. Tiap ruas jalan akan dianggap sebuah *step* oleh *google maps server*.

Penunjuk arah tersebut akan berubah setiap jarak yang ditempuh kurang dari 15 meter dari lokasi pengguna menuju step selanjutnya. Penunjuk arahpun akan berubah sesuai dengan dengan *tag maneuver* yang telah didapatkan dari *google maps api*.



Gambar 5.12. Tampilan AR Activity

Gambar 5.13 adalah gambar dari *source code* dari *AR Activity*. *activity* ini menggunakan *surfaceview* untuk menangkap hasil keluaran dari kamera. *Surfaceview* digunakan untuk menyajikan *preview* dari kamera. *Activity* ini menggunakan *surfaceHolder* dan *locationListener* sebagai *interfacenya*. *Activity* ini menerima lokasi tujuan dari *navigation activity* yang di kirim melalui sebuah *bundle*.

Setelah menerima data lokasi tersebut, data akan dikirim ke *google maps server* untuk mendapatkan *JSON* rute yang harus ditempuh. Pada *AR Activity* data tersebut tidak dirubah menjadi sebuah rute, tetapi hanya diambil *tag* jarak, *tag* instruksi dan *tag maneuver*. Proses tersebut berada pada *method getLocation*. Pada *method* tersebut akan didapatkan nilai yang ada di tag-tag tersebut.

Method getLocation juga menghitung jarak yang akan ditempuh. Jika jarak dari lokasi pengguna dengan *step* selanjutnya kurang dari 15 meter, maka penunjuk akan

berubah sesuai dengan *tag maneuver* yang telah didapatkan. Setiap berubahnya penunjuk arah, jarak dan instruksi maka akan ditangani oleh *method Handler*

```
public class ARActivity extends Activity implements SurfaceHolder.Callback,
    LocationListener {
    private Camera camera = null;
    private SurfaceView surfaceView = null;
    private SurfaceHolder surfaceHolder = null;
    private boolean previewing = false;
    RelativeLayout relativeLayout;
    ImageView gambar;
    String dataJarak = null;
    String dataPerintah = null;
    TextView tv_jarak, tv_perintah;
    double to_lat, to_long;
    public static int background;
    public Handler handler = new Handler() {
        @Override
        public void handleMessage(Message msg) {
            gambar.setBackgroundResource(background);
            tv_jarak.setText(dataJarak);
            tv_perintah.setText(Html.fromHtml(dataPerintah));
        }
    };
    String maneuver = "";
    @TargetApi(Build.VERSION_CODES.GINGERBREAD)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        StrictMode.enableDefaults();

        getWindow().setFormat(PixelFormat.TRANSLUCENT);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        LocationManager locationManager;
        locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
```

```
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
    3000, 10, this); // location(location);
setContentView(R.layout.activity_ar);

Bundle extras = getIntent().getExtras();
to_lat = extras.getDouble("to_lat");
to_long = extras.getDouble("to_long");
gambar = (ImageView) findViewById(R.id.imageView1);
RelativeLayout = (RelativeLayout) findViewById(R.id.containerImg);
RelativeLayout.setDrawingCacheEnabled(true);
tv_jarak = (TextView) findViewById(R.id.jarak);
tv_perintah = (TextView) findViewById(R.id.perintah);
surfaceView = (SurfaceView) findViewById(R.id.surfaceView1);
surfaceHolder = surfaceView.getHolder();
surfaceHolder.addCallback(this);
}

@Override
public void surfaceChanged(SurfaceHolder holder, int format, int width,
    int height) {
    // TODO Auto-generated method stub
    if (previewing) {
        camera.stopPreview();
        previewing = false;
    }
    try {
        if (this.getResources().getConfiguration().orientation != Configuration.ORIENTATION_LANDSCAPE) {
            camera.setDisplayOrientation(90);
        }
        camera.setPreviewDisplay(surfaceHolder);
        camera.startPreview();
        previewing = true;
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

@Override
public void surfaceCreated(SurfaceHolder holder) {
    // TODO Auto-generated method stub
    try {
        camera = Camera.open();
    } catch (RuntimeException e) {
        Toast.makeText(
            getApplicationContext(),
            "Device camera is not working properly, please try after sometime.",
            Toast.LENGTH_LONG).show();
    }
}

@Override
public void surfaceDestroyed(SurfaceHolder holder) {
    // TODO Auto-generated method stub
    camera.stopPreview();
    camera.release();
    camera = null;
    previewing = false;
}

@Override
public void onLocationChanged(Location location) {
    // TODO Auto-generated method stub
    String url = "http://maps.googleapis.com/maps/api/directions/json?" + "origin=" + location.getLatitude() + ","
        + location.getLongitude() + "&destination=" + to_lat + "," + to_long
        + "&sensor=false&units=metric&mode=Walking" ;
    Log.i("url", url);
    getLocation(url);
}
}
```

```

private void getLocation(String url) {
    String json = "";
    int jaraknya=0,durasi=0;
    try {
        HttpClient httpClient = new DefaultHttpClient();
        HttpGet httpget = new HttpGet();
        httpget.setURI(new URI(url));
        HttpResponse response = httpClient.execute(httpget);
        json = request(response);
    } catch (Exception e) {
        Log.e("error :", e.getMessage());
    }
    try {
        JSONObject jobject = new JSONObject(json);
        JSONArray manueverArray = jobject.getJSONArray("routes");
        jobject = new JSONObject("{"legs:"+ manueverArray.getJSONObject(0).getString("legs").toString() + "}");
        manueverArray = jobject.getJSONArray("legs");
        jobject = new JSONObject("{"steps:"+ manueverArray.getJSONObject(0).getString("steps").toString() + "}");
        manueverArray = jobject.getJSONArray("steps");
        //for (int i = 0; i < manueverArray.length(); i++) {

        JSONObject jarak = new JSONObject("{"distance:"+ manueverArray.getJSONObject(0).getString("distance").toString() + "}");
        JSONArray v_jarak = jarak.getJSONArray("distance");
        try {
            jaraknya=Integer.parseInt(v_jarak.getJSONObject(0).getString("value").toString());
            dataJarak = v_jarak.getJSONObject(0).getString("value").toString();
            dataPerintah = manueverArray.getJSONObject(0).getString("html_instructions").toString();
            Log.i("data jarak", dataJarak+" "+dataPerintah);
            if(manueverArray.length()>0 || jaraknya>15){
                if(jaraknya<16){
                    try{
                        manuever=manueverArray.getJSONObject(1).getString("manuever").toString();
                    } catch (Exception e) {
                        manuever=manueverArray.getJSONObject(2).getString("manuever").toString();
                    }
                }else{
                    manuever=manueverArray.getJSONObject(0).getString("manuever").toString();
                }
            }else{
                manuever="Sampai Tujuan";
            }
        } catch (Exception e) {
            manuever="Jalan Lurus";
        }
    } catch (Exception e) {
        manuever=e.getMessage();
        Log.i("test",manuever);
    }
    // Toast.makeText(this, manuever+" "+jaraknya+"m", Toast.LENGTH_LONG).show();
    if (manuever.equals("turn-right")) {
        // Toast.makeText(this, "lurus", Toast.LENGTH_SHORT).show();
        background=R.drawable.icon_hanan;
    } else if (manuever.equals("turn-left")) {
        // Toast.makeText(this, "lurus", Toast.LENGTH_SHORT).show();
        background=R.drawable.icon_kiri;
    } else if (manuever.equals("turn-sharp-left")) {
        background = R.drawable.icon_turn_sharp_left;
    } else if (manuever.equals("uturn-right")) {
        background = R.drawable.icon_uturn_right;
    } else if (manuever.equals("uturn-left")) {
        background = R.drawable.icon_uturn_left;
    } else if (manuever.equals("turn-sharp-right")) {
        background = R.drawable.icon_turn_sharp_right;
    } else if (manuever.equals("roundabout-left")) {
        background = R.drawable.icon_roundabout;
    } else if (manuever.equals("roundabout-right")) {
        background = R.drawable.icon_roundabout;
    } else {
        background=R.drawable.icon_lurus;
    }
}

```

```

    }
    this.handler.sendMessage(this.handler.obtainMessage());
}

private static String request(HttpResponse response) {
    try {
        InputStream in = response.getEntity().getContent();
        BufferedReader reader = new BufferedReader(
            new InputStreamReader(in));
        StringBuilder str = new StringBuilder();
        String line = null;
        boolean first = true;
        while ((line = reader.readLine()) != null) {
            if (first) {
                first = false;
            } else {
                str.append("\n");
            }
            str.append(line);
        }
        in.close();
        return str.toString();
    } catch (Exception ex) {
        return "Error request";
    }
}

@Override
public void onProviderDisabled(String provider) {
    // TODO Auto-generated method stub
}

@Override
public void onProviderEnabled(String provider) {
    // TODO Auto-generated method stub
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
    // TODO Auto-generated method stub
}
}
}

```

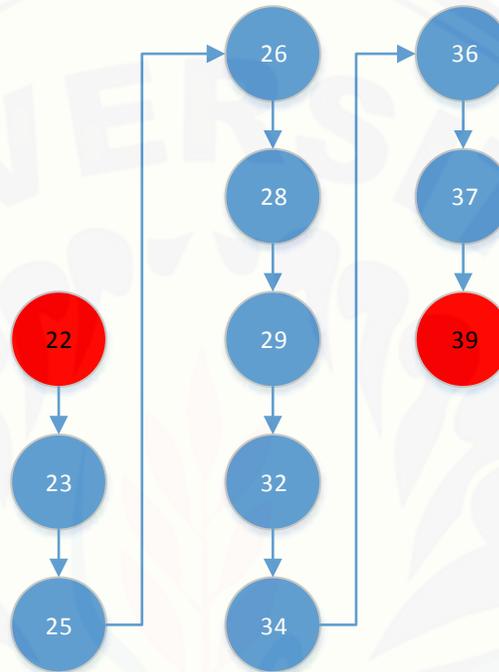
Gambar 5.13. Source Code AR Activity

5.3. Pengujian Aplikasi

Pengujian aplikasi ini dimulai dengan mengerjakan pembuatan aplikasi. Pengujian aplikasi akan dilakukan setelah mengkodekan sistem. Tahapan pengujian sistem akan dibagi menjadi tiga tahap yaitu pengujian secara *white box* dan *black box* serta pengujian pendekatan skenario.

5.3.1. Pengujian White Box

Pengujian secara white box adalah pengujian yang mengukur kinerja sistem. Teknik pengujian yang digunakan yaitu pembuatan diagram alir, perhitungan siklomatik. Pengujian pada aplikasi ini selengkapnya akan dijelaskan di lampiran 1.B.



Gambar 5.14. Diagram Alir *Method Oncreate* Kelas *Splash Activity*

Gambar 5.14 menunjukkan diagram alir dari method oncreate yang ada didalam kelas splash activity. perhitungan siklomatik berdasarkan dari diagram alir dengan rumus $V(G) = p+1$, dimana p adalah predikat. Predikat ini berupa simbol *diamond* yang terdapat pada diagram alir G. berdasarkan diagram alir tersebut maka nilai perhitungan siklomatik pada diagram ini adalah,

$$V(G) = 0+1 = 1$$

5.3.2. Pengujian Black Box

Pengujian ini akan menguji aplikasi ARMN secara fungsional dimana pengujian ini dilakukan dalam bentuk tertulis. Pengujian ini berfungsi untuk memeriksa apakah aplikasi tersebut telah berjalan sesuai dengan fungsinya. Pengujian ini juga meliputi

seberapa baik sistem melaksanakan fungsinya, termasuk perintah yang dijalankan pengguna, pencarian dan proses eksekusi data, pengguna layar dan integrasi. Pengujian pada aplikasi ini juga akan dijelaskan selengkapnya di lampiran 1.C.

5.3.3. Pengujian Pendekatan Skenario

Cara untuk menguji sebuah aplikasi yang menggunakan pendekatan skenario adalah dengan menggunakan 2 *test case* yang telah dijelaskan di tinjauan pustaka.

a. *Test Case* dari *User Interaction Diagram*

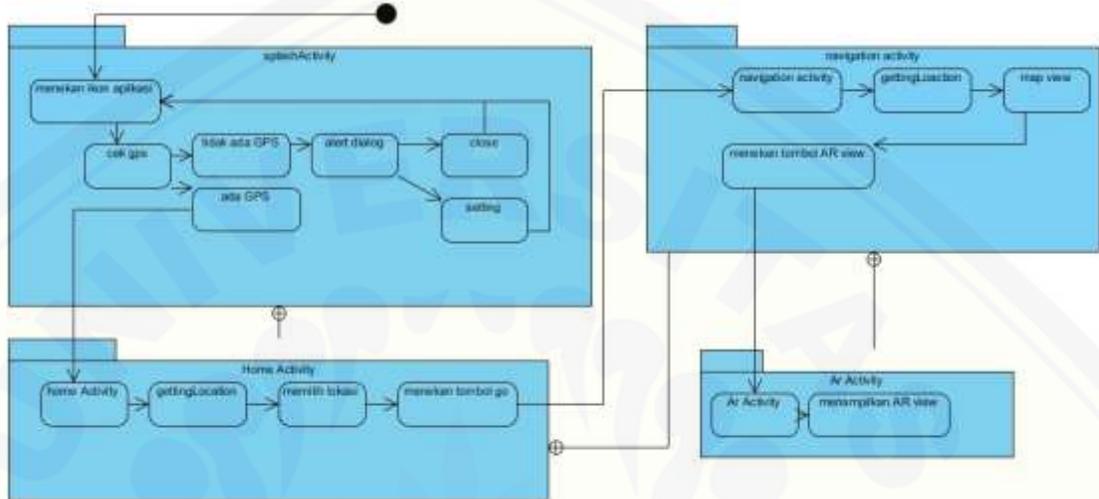
Test case didapatkan dari tiap *user Interaction diagram*. Tabel 5.1 adalah tabel yang menunjukkan *test case* dari *splash screen*. *Test case* ini akan dijelaskan selengkapnya pada lampiran 1.D.

Tabel 5.1. Tabel Test Case Dari *User Interaction Diagram Splash Screen*

Test Preparation		Pengguna menekan ikon aplikasi	
No	status	Kondisi	Keluaran
1	Delay 3 detik	aplikasi mulai	Splash screen tampil
2	Delay 3 detik	Aplikasi mulai mengecek GPS	Progress dialog muncul
3	Cek GPS	Aplikasi menemukan GPS	Menuju halaman Home
4	Cek GPS	Aplikasi tidak dapat menemukan GPS	Alert Dialog muncul
	Alert Dialog	Memilih pilihan keluar	Keluar aplikasi
5	Cek GPS	Aplikasi tidak dapat menemukan GPS	Alert Dialog muncul
6	Alert Dialog	Memilih pilihan setting	Masuk ke halaman setting
7	Halaman setting	Mengaktifkan GPS	Alert dialog muncul
8	Alert Dialog	Menekan Ok	Keluar dari aplikasi

b. *Test Case* yang berasal dari *dependency chart*

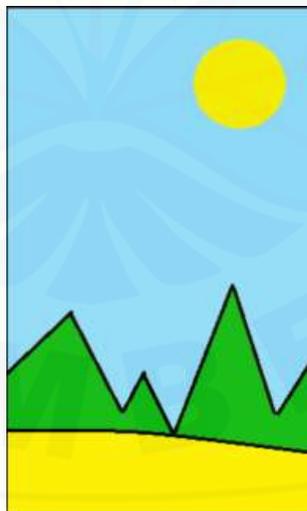
Test case ini berupa *dependency chart* yang berasal dari skenario. Gambar 5.15 adalah *dependency chart* dari aplikasi ini.



Gambar 5.15. *Dependency Chart* Dari Aplikasi ARMN

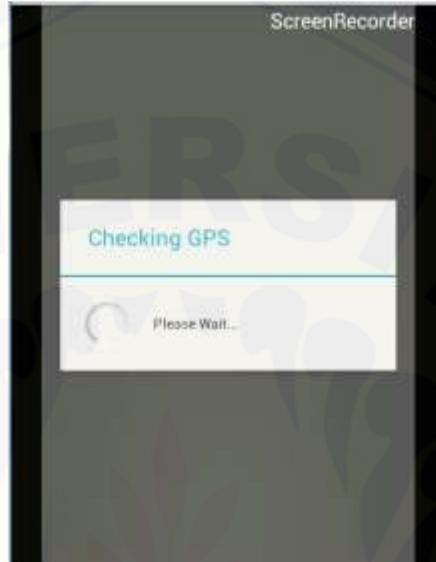
5.4. Menggunakan Sistem

Setelah aplikasi di evaluasi, maka aplikasipun siap digunakan. Gambar 5.16 adalah *screenshot* dari aplikasi ketika digunakan.



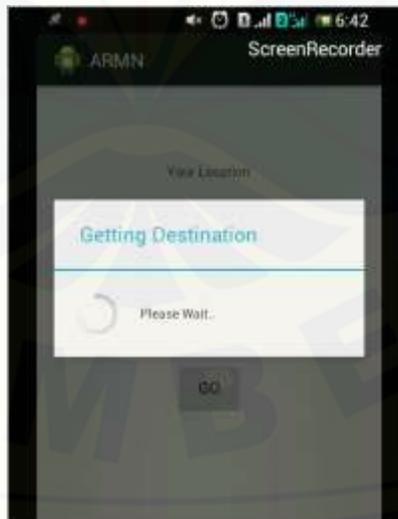
Gambar 5.16. Gambar *Splash Screen*

Gambar 5.17 menunjukkan aplikasi sedang melakukan pengecekan GPS. Aplikasi menggunakan progress dialog untuk menampilkan bahwa aplikasi sedang mengecek GPS



Gambar 5.17. *Progress Dialog Check Gps*

Gambar 5.18 adalah gambar dimana aplikasi sedang mengambil data GPS dan data lokasi tujuan pada server.



Gambar 5.18 *Progress Dialog Getting Location*

Gambar 5.19 adalah tampilan *home screen* yang telah mendapatkan lokasi pengguna dan lokasi tujuan.



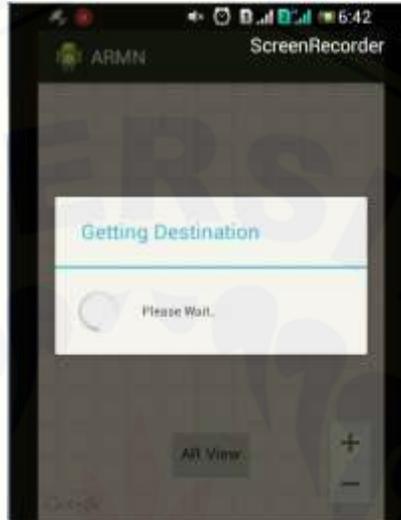
Gambar 5.19. Gambar *Home Screen*

Gambar 5.20 adalah gambar dimana pengguna akan memilih lokasi tujuan yang ada pada spinner lokasi. Setelah memilih lokasi, pengguna menekan tombol *go* untuk menuju tampilan selanjutnya.



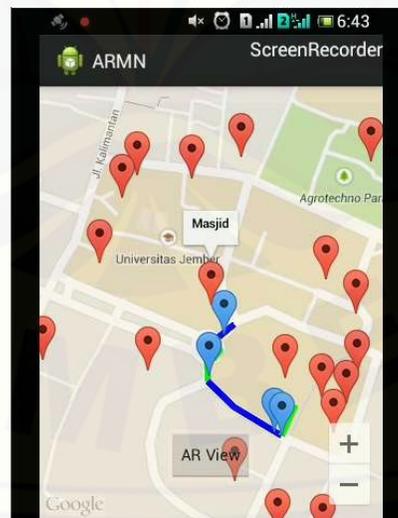
Gambar 5.20. Memilih Lokasi Tujuan

Gambar 5.21 menunjukkan *progress dialog* yang sedang mengambil lokasi dan mengirim lokasi pengguna dan lokasi tujuan. Data yang telah dikirim akan menjadi rute. Data yang diambil akan menjadi titik lokasi tujuan.



Gambar 5.21. Gambar *Progress Dialog Getting Location*

Gambar 5.22 adalah gambar rute dan titik lokasi tujuan yang dioverlay ke tampilan peta. Pengguna menekan tombol *AR View* untuk menuju ke tampilan *AR View*.



Gambar 5.22. Gambar *Map Beserta Rute Dan Titik Gedung Tujuan*

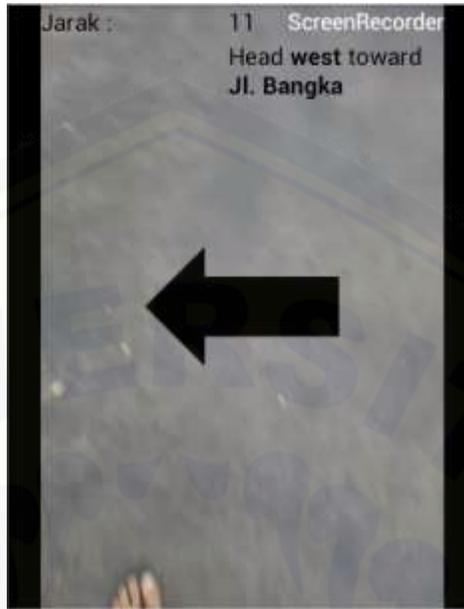
Gambar 5.23 hingga gambar 5.26 adalah gambar dari tampilan AR view. Gambar ini berisi sebuah penunjuk arah, jarak dan sebuah instruksi.



Gambar 5.23. Gambar Ar View



Gambar 5.24. Gambar AR View



Gambar 5.25. Gambar AR View



Gambar 5.26. Gambar AR View

BAB 6. PENUTUP

6.1 Kesimpulan

Berdasarkan analisis yang peneliti lakukan, maka dapat diambil beberapa kesimpulan. Aplikasi ini membantu pengguna untuk menuju lokasi gedung yang ada di Universitas Jember dikarenakan aplikasi ini mengambil rute dari lokasi pengguna menuju lokasi tujuan. Aplikasi ini mengirimkan lokasi pengguna dan lokasi tujuan ke *server google map* sehingga aplikasi ini akan mendapatkan rute yang diinginkan.

Aplikasi ini juga membantu pengguna yang susah memahami peta karena aplikasi ini menggunakan *augmented reality* untuk menunjukkan jalan.

Pendekatan skenario dapat membantu pengembang aplikasi untuk menemukan kebutuhan-kebutuhan aplikasi. Pendekatan skenario lebih mengikutsertakan pengguna dalam menentukan kebutuhan pengguna.

Lokasi awal dan lokasi akhir terkadang mendapatkan posisi yang berubah-ubah, dikarenakan untuk mendapatkan lokasi awal dibutuhkan minimal 3 BTS. Lokasi-lokasi tersebut biasanya dipengaruhi oleh kuat sinyal BTS sehingga rute yang didapatkan terkadang berubah-ubah.

6.2. Saran

Beberapa saran berikut diharapkan dapat memberikan perbaikan sistem dalam penelitian selanjutnya. Diharapkan tampilan aplikasi lebih menarik, sehingga akan lebih menarik pengguna dalam menggunakan aplikasi tersebut. Aplikasi ini membutuhkan koneksi yang stabil untuk mengirim dan mendapatkan data dari google server, untuk penelitian berikutnya diharapkan penggunaan aplikasi dapat digunakan ketika aplikasi tidak mendapatkan koneksi yang stabil.

Ketika aplikasi ini digunakan, aplikasi ini membutuhkan spesifikasi hardware smartphone yang lebih tinggi, sehingga diharapkan dalam penelitian aplikasi serupa dapat digunakan dalam kondisi hardware yang lebih rendah.

Aplikasi ini juga terkadang tidak mendapatkan lokasi yang pengguna yang sesuai. Kekurangan ini dikarenakan terkadang aplikasi tidak mendapatkan lokasi

yang akurat. Diharapkan pada penelitian selanjutnya pengambilan lokasi pengguna lebih akurat.

Penelitian ini hanya berfungsi dengan *mode walking*. Diharapkan kedepannya aplikasi serupa dapat berfungsi dalam mode yang lain, seperti *mode bicycle*, *driving* dan lain lain.

Aplikasi ini masih dalam tahap pengembangan sehingga terkadang rute yang didapatkan tidak akurat. Diharapkan kedepannya aplikasi serupa bisa mendapatkan hasil yang lebih akurat.

DAFTAR PUSTAKA

- A.S., Rosa, and M. Shalahuddin. *Rekayasa Perangkat Lunak*. Bandung: Informatika, 2013.
- Almendros-Jimenez, Jesus M., and Luis Iribarne. "User Interaction and Interface Design with UML." In *Human Computer Interaction: Concepts, Methodologies, Tools, and Applications*, by Panayiotis Zaphiris, & Chee Siang Ang, 404-431. New York: Information Science Reference , 2009.
- Aronoff, Stanley. *Geographic Information System: A Guide to The Technology*. canada: WDL Publication, 1989.
- Azuma, Ronald T. "A Survey of Augmented Reality." *Teleoperators and Virtual Environments* 6, 1997: 355-385.
- Brigida. <http://informatika.web.id/>. februari 14, 2013.
<http://informatika.web.id/pengertian-uml.htm> (accessed februari 10, 2015).
- Gibilisco, Stan. <http://whatis.techtarget.com/>. february 2013.
<http://whatis.techtarget.com/definition/Google-Maps> (accessed 06 29, 2014).
- Giboin, Alain, Fabien Gandon, Olivier Corby, and Rose Dieng. "Assessment of Ontology-based Tools: A Step Towards Systemizing the Scenario Approach." *INRIA*, 2004: 93.
- Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." *IEEE Transactions on Systems Science and Cybernetics SSC4*, 1986: 100-107.
- Indra, Anisa. *Prototyping Model*. Juni 19, 2013.
<http://www.varia.web.id/2013/06/prototyping-model.html> (accessed 3 1, 2014).
- Jang, Sung Hyun, and Andrew Hudson-Smith. *Exploring Mobile Augmented Reality Navigation System for Pedestrians*. London: University College London, 2010.

- Johannes Ryser, Martin Glinz. "A Scenario-Based Approach to Validating and Testing Software Systems Using Statecharts." *International Conference on Software and Systems Engineering and their Applications*. Paris: CNAM, 99.
- Lazuardy, Senja. *Tekno Kompas*. 04 19, 2012.
<http://tekno.kompas.com/read/2012/04/09/12354384/augmented.reality.masa.depan.interaktivitas>.
- Möller, Andreas, Matthias Kranz, Robert Huitl, Stefan Diewald, and Luis Roalter. *A Mobile Indoor Navigation System Interface Adapted to Vision-Based Localization*. Munchen: ACM, 2012.
- Mulyoto, Itto Wirawan. *Herman Class*. 09 2013.
<http://www.hermantolle.com/class/2013/09/tren-gps-pada-smartphone/>
(accessed 2 27, 2015).
- Prahasta, Eddy. *Sistem Informasi Geografis : Konsep-konsep Dasar (Perspektif Geodesi & Geomatika)*. Bandung: Informatika, 2009.
- Tim Penyusun. *Buku Pedoman Kemahasiswaan Pengenalan Kehidupan dan Pembinaan Pengembangan Mahasiswa Baru Tahun Akademik 2012/2013*. Jember: Jember University Press, 2012.
- Wojciechowski, Adam. "Mobile Vision Based Augmented Reality Navigation System." *JOURNAL OF APPLIED COMPUTER SCIENCE*, 2012: 103-118.

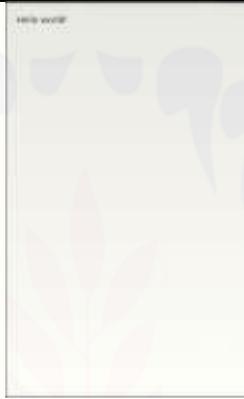
LAMPIRAN 1

A. Pengembangan *Prototype*

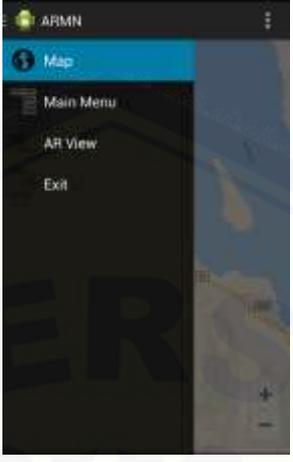
Lampiran ini akan menyajikan prototype pengembangan aplikasi beserta kelemahannya.

1. Prototype 1

Tabel 1. Tahap Pengembangan *Prototype* 1

No	Tampilan	Gambar	Kelemahan
1	<i>Splash Screen</i>		Delay lebih cepat mengecek GPS
2	<i>Home Screen</i>		Masih belum bisa mengambil data dari server

3 *Menu*



Tidak bisa *overlay* rute

4 *Map Screen*



Masih belum ada rute

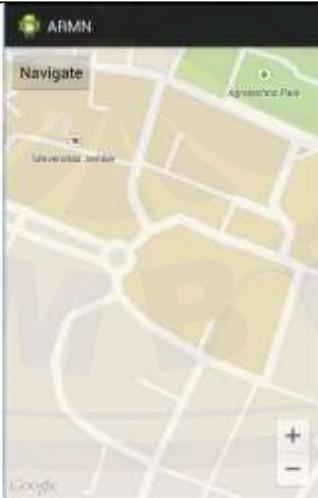
5 *AR View*



Masih belum berfungsi
AR View

2. Prototype 2

Tabel 2. Tahap Pengembangan *Prototype 2*

No	Tampilan	Gambar	kelemahan
1	<i>Splash screen</i>		Delay masih lebih cepat
2	<i>Home Screen</i>		Masih belum bisa mengambil dari server
3	<i>Map View</i>		Sudah bisa menampilkan rute, tetapi masih hanya demo

4	<i>AR View</i>		<i>AR View</i> masih belum berfungsi
---	----------------	--	--------------------------------------

3. Prototype 3

Tabel 3. Tahap Pengembangan *Prototype 3*

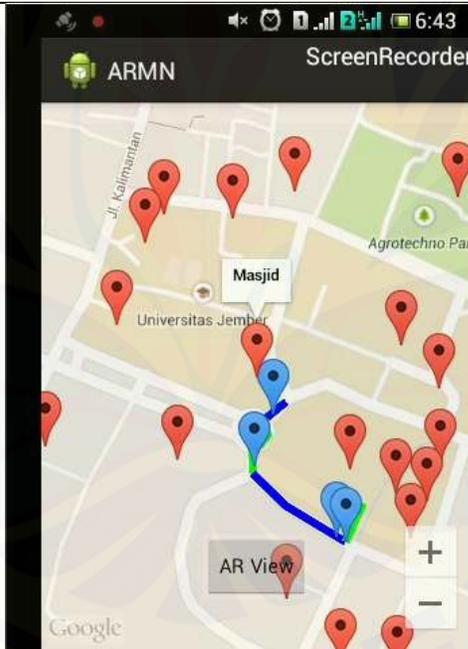
No	Tampilan	Gambar	kelemahan
1	<i>Splash screen</i>		Fungsi normal, tampilan masih belum tidak menarik

2 *Home Screen*



Fungsi normal, dapat mengambil data dari server

3 *Map View*



Dapat mengambil rute yang akan ditempuh oleh pengguna

4 AR View Jarak : 38 ScreenRecorderAR View telah berfungsi

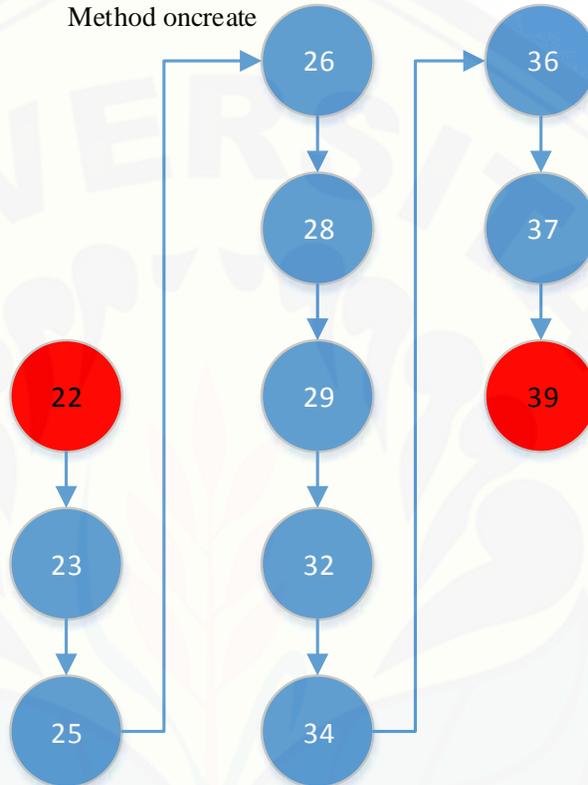


B. Pengujian White Box

1. SplashActivity

Splashactivity class

Method oncreate



Gambar 1. Cyclomatic Complexity SplashActivity Method Oncreate

Nilai siklomatik yang didapatkan dari *method oncreate*

$$V(G) = p + 1 = 0 + 1 = 1$$

SplashActivity class : inner class checkGPS
Method onPreexecute

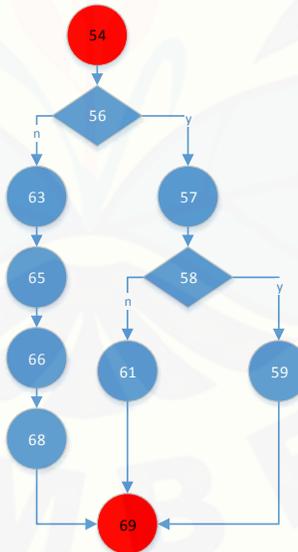


Gambar 2. Cyclomatic Complexity SplashActivity Method OnPreexecute

Nilai siklomatik yang didapatkan dari *method OnPreExecute*

$$V(G) = p + 1 = 0 + 1 = 1$$

SplashActivity class : inner class checkGPS
Method doInBackground

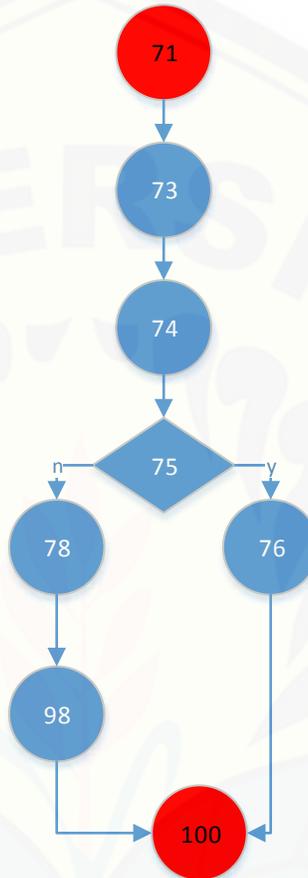


Gambar 3. Cyclomatic Complexity SplashActivity Method doInBackground

Nilai siklomatik yang didapatkan dari *method doInBackground*

$$V(G) = p + 1 = 2 + 1 = 3$$

SplashActivity class : inner class checkGPS
Method onPostExecute

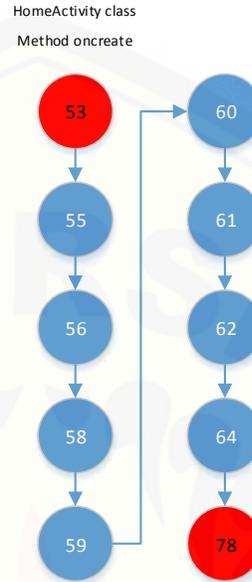


Gambar 4. *Cyclomatic Complexity SplashActivity Method onPostExecute*

Nilai siklomatik yang didapatkan dari *method onPostExecute*

$$V(G) = p + 1 = 1 + 1 = 2$$

2. *HomeActivity*



Gambar 5. *Cyclomatic Complexity HomeActivity Method onCreate*

Nilai siklomatik yang didapatkan dari *method onCreate*

$$V(G) = p + 1 = 0 + 1 = 1$$

HomeActivity class : inner class DownloadJson
Method onPreexecute

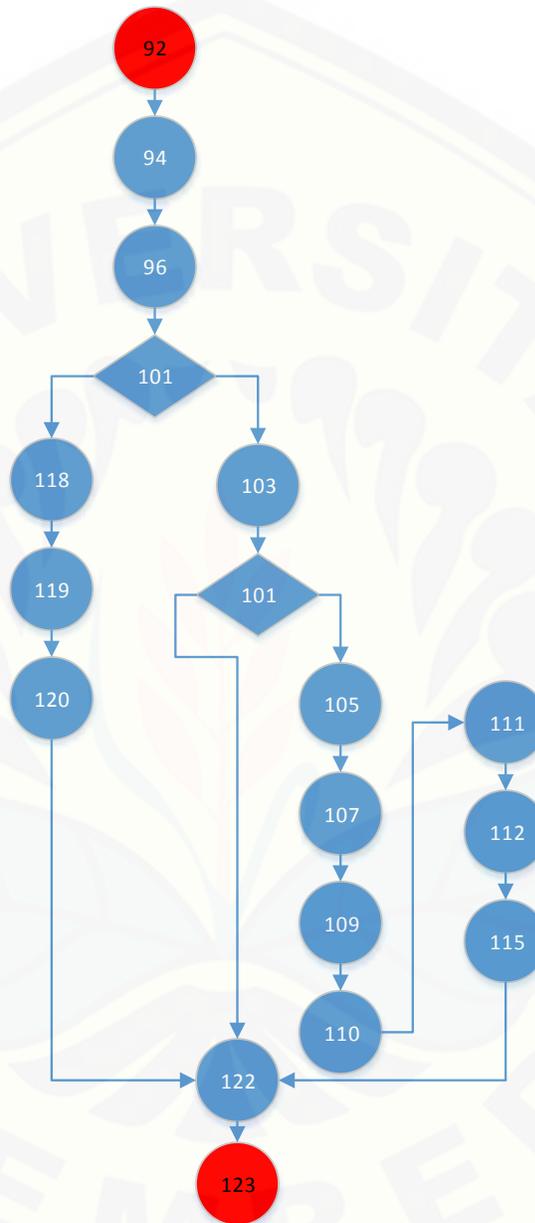


Gambar 6. *Cyclomatic Complexity HomeActivity Method onPreexecute*

Nilai siklomatik yang didapatkan dari *method onPreexecute*

$$V(G) = p + 1 = 0 + 1 = 1$$

homeActivity class : inner class DownloadJson
Method doInBackground

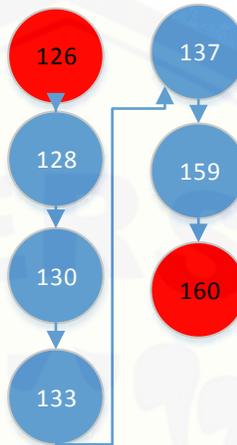


Gambar 7. Cyclomatic Complexity HomeActivity Method doInBackground

Nilai siklomatik yang didapatkan dari *method doInBackground*

$$V(G) = p + 1 = 2 + 1 = 3$$

HomeActivity class : inner class DownloadJson
Method onPostExecute



Gambar 8. Cyclomatic Complexity HomeActivity Method onpostexecute

Nilai siklomatik yang didapatkan dari *method onpostExecute*

$$V(G) = p + 1 = 0 + 1 = 1$$

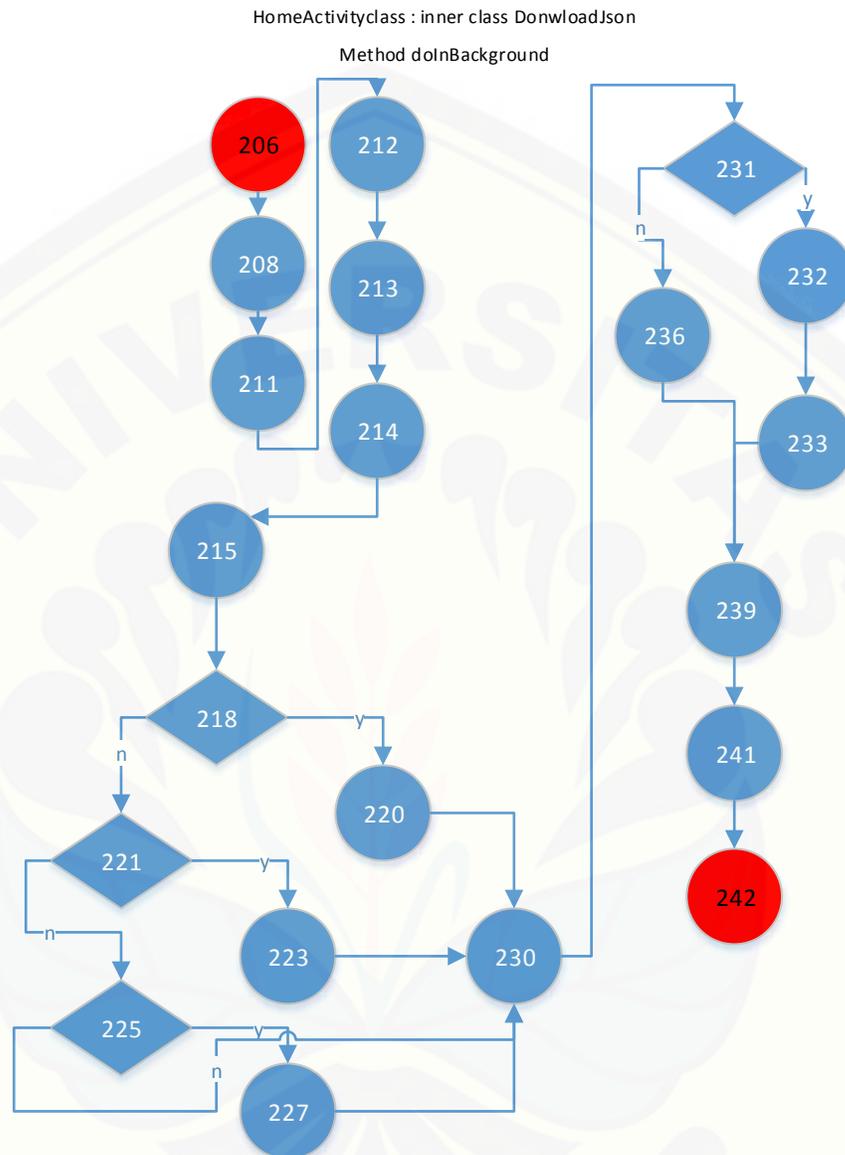
HomeActivity class : inner class getLocation
Method onpreexecute



Gambar 9. Cyclomatic Complexity HomeActivity Method onpreexecute

Nilai siklomatik yang didapatkan dari *method onPreExecute*

$$V(G) = p + 1 = 0 + 1 = 1$$

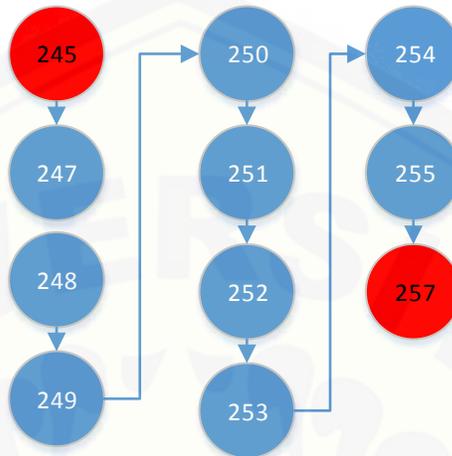


Gambar 10. Cyclomatic Complexity HomeActivity Method doInBackground

Nilai siklomatik yang didapatkan dari *method doInBackground*

$$V(G) = p + 1 = 4 + 1 = 5$$

HomeActivity class : inner class DownloadJson
Method onPostExecute



Gambar 11. Cyclomatic Complexity HomeActivity Method onPostExecute

Nilai siklomatik yang didapatkan dari *method onPostExecute*

$$V(G) = p + 1 = 0 + 1 = 1$$

HomeActivity class : inner class DownloadJson
Method onLocationChanged



Gambar 12. Cyclomatic Complexity HomeActivity Method OnlocationChanged

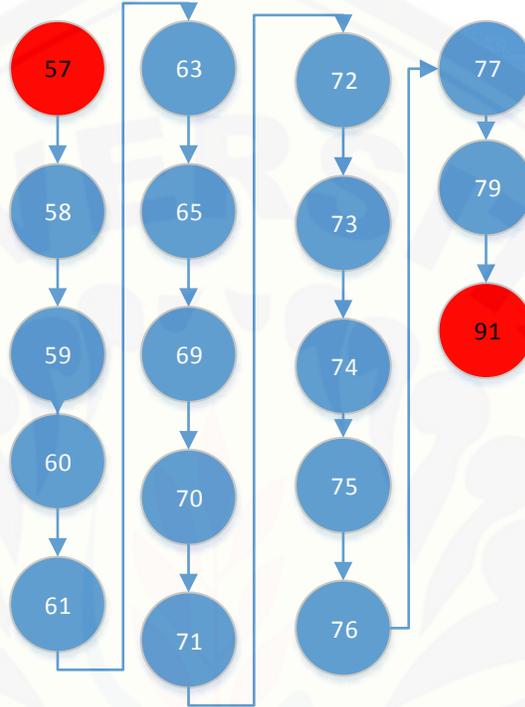
Nilai siklomatik yang didapatkan dari *method onLocationChanged*

$$V(G) = p + 1 = 0 + 1 = 1$$

3. *NavigationActivity*

Navigation Activity class

Method oncreate



Gambar 13. *Cyclomatic Complexity NavigationActivity Method oncreate*

Nilai siklomatik yang didapatkan dari *method oncreate*

$$V(G) = p + 1 = 0 + 1 = 1$$

Navigation Activity class
Method onResume



Gambar 14. Cyclomatic Complexity NavigationActivity Method onResume

Nilai siklomatik yang didapatkan dari *method onResume*

$$V(G) = p + 1 = 0 + 1 = 1$$

Navigation Activity class
Method getScreenDimension



Gambar 15. Cyclomatic Complexity NavigationActivity Method getScreenDimension

Nilai siklomatik yang didapatkan dari *method getScreenDimension*

$$V(G) = p + 1 = 0 + 1 = 1$$

navigationActivity class : inner class

DownloadJson

Method onpreexecute



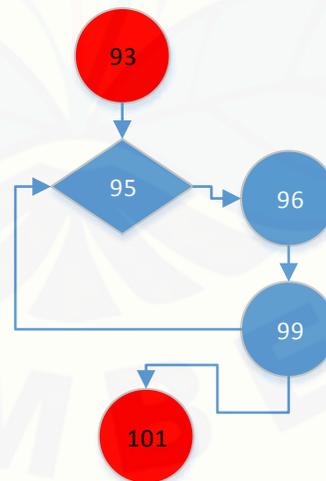
Gambar 16. Cyclomatic Complexity NavigationActivity Method onPreexecute

Nilai siklomatik yang didapatkan dari *method onPreExecute*

$$V(G) = p + 1 = 0 + 1 = 1$$

Navigation Activity class

Method ondirectionLoaded

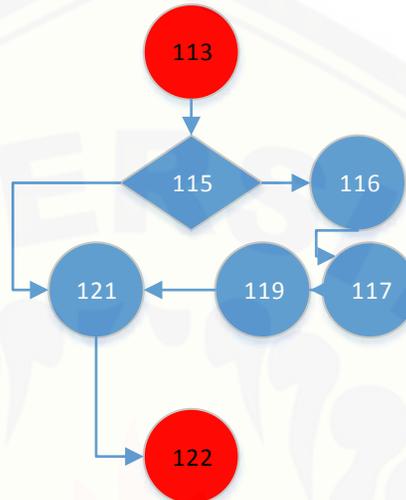


Gambar 17. Cyclomatic Complexity NavigationActivity Method OndirectionLoaded

Nilai siklomatik yang didapatkan dari *method onDirectionLoaded*

$$V(G) = p + 1 = 1 + 1 = 2$$

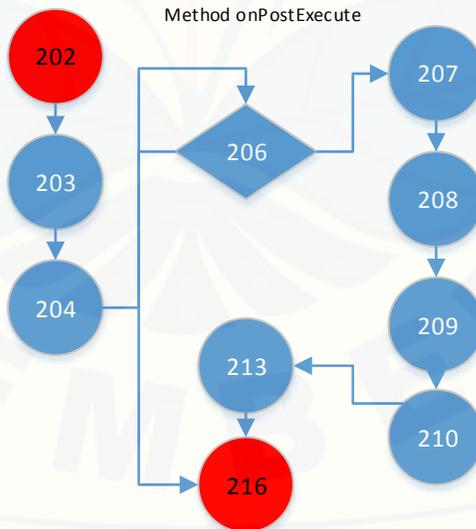
Navigation Activity class
Method createLatLangBound



Gambar 18. Cyclomatic Complexity NavigationActivity Method createLatLangBound
Nilai siklomatik yang didapatkan dari *method createLatLangBound*

$$V(G) = p + 1 = 1 + 1 = 2$$

navigationActivity class : inner class DonwloadJson
Method onPostExecute

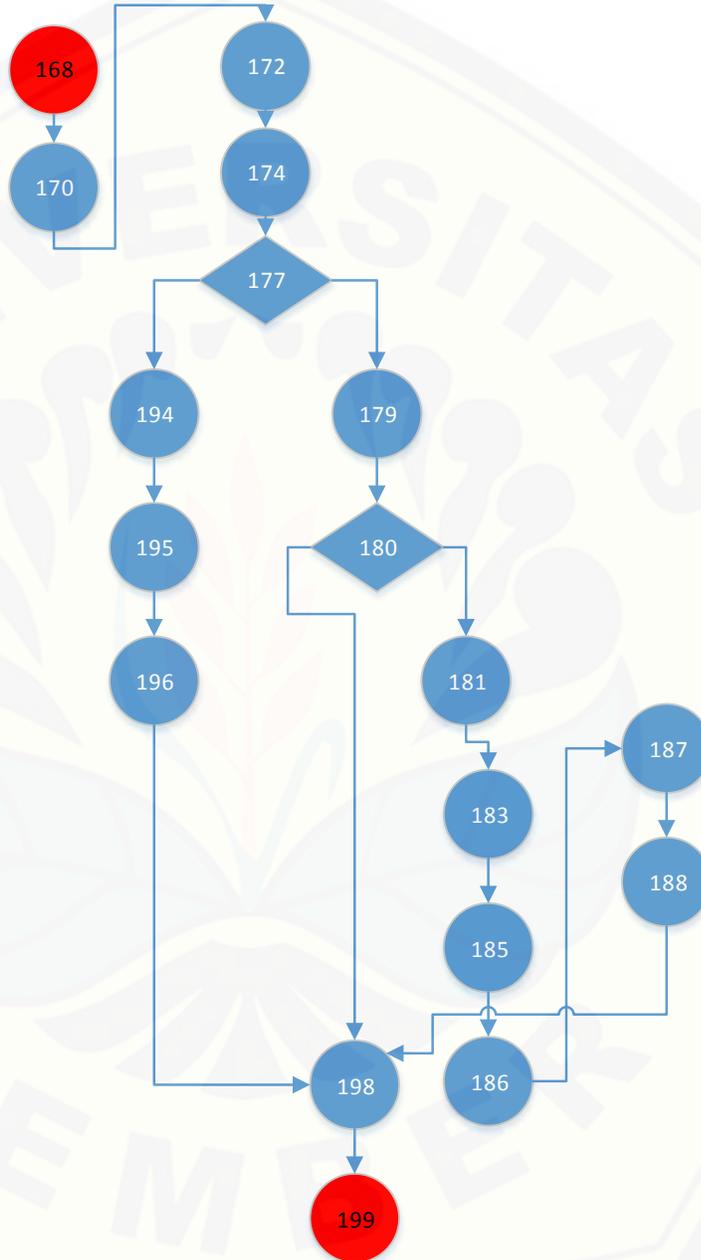


Gambar 19. Cyclomatic Complexity NavigationActivity Method onpostExecute
Nilai siklomatik yang didapatkan dari *method onpostexecute*

$$V(G) = p + 1 = 1 + 1 = 2$$

navigationActivity class : inner class DownloadJson

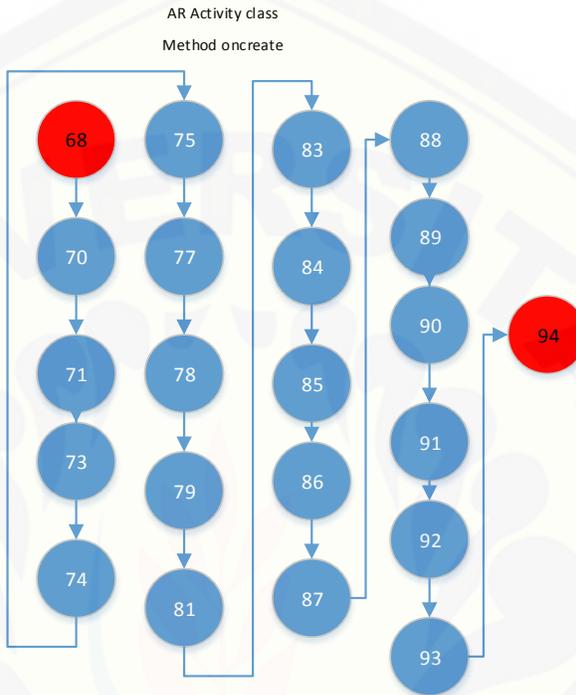
Method doInBackground



Gambar 20. Cyclomatic Complexity NavigationActivity Method doInbackground
Nilai siklomatik yang didapatkan dari method DoInbackground

$$V(G) = p + 1 = 2 + 1 = 3$$

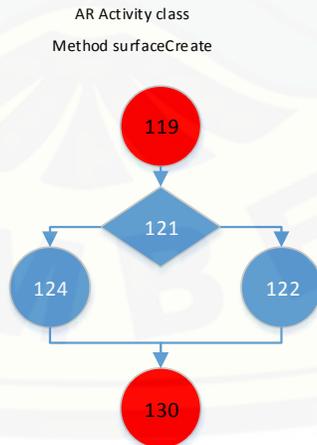
4. *ARActivity*



Gambar 21. Cyclomatic Complexity *ARActivity Method oncreate*

Nilai siklomatik yang didapatkan dari *method onCreate*

$$V(G) = p + 1 = 0 + 1 = 1$$

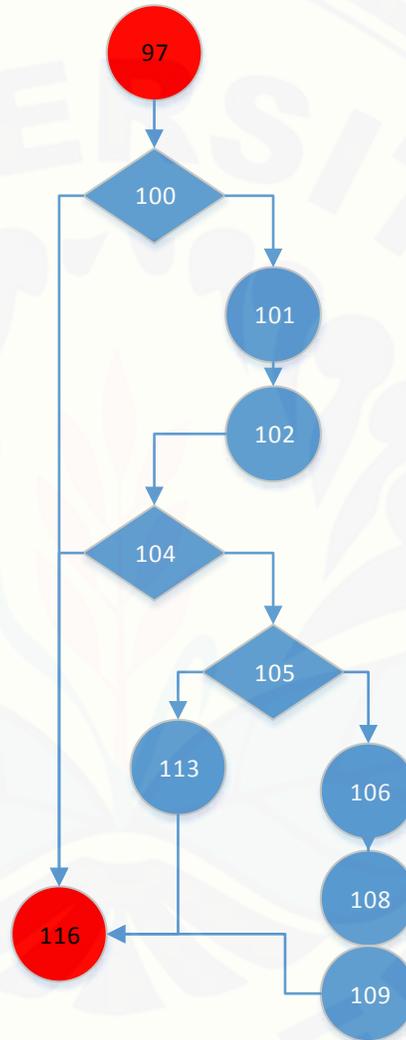


Gambar 22. Cyclomatic Complexity *ARActivity Method surfaceCreate*

Nilai siklomatik yang didapatkan dari *method surfaceCreate*

$$V(G) = p + 1 = 1 + 1 = 1$$

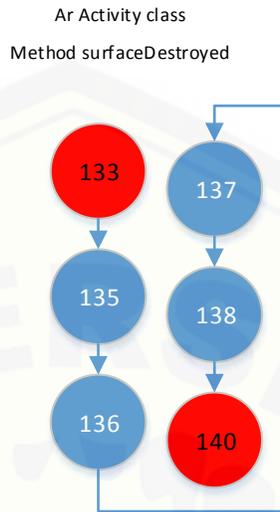
AR Activity class
Method surfaceChanged



Gambar 23. Cyclomatic Complexity ARActivity Method surfaceChanged

Nilai siklomatik yang didapatkan dari *method surfaceChanged*

$$V(G) = p + 1 = 3 + 1 = 4$$



Gambar 24. *Cyclomatic Complexity ARActivity Method surfaceDestroyed*
Nilai siklomatik yang didapatkan dari *method surfaceDestroyed*
 $V(G) = p + 1 = 0 + 1 = 1$

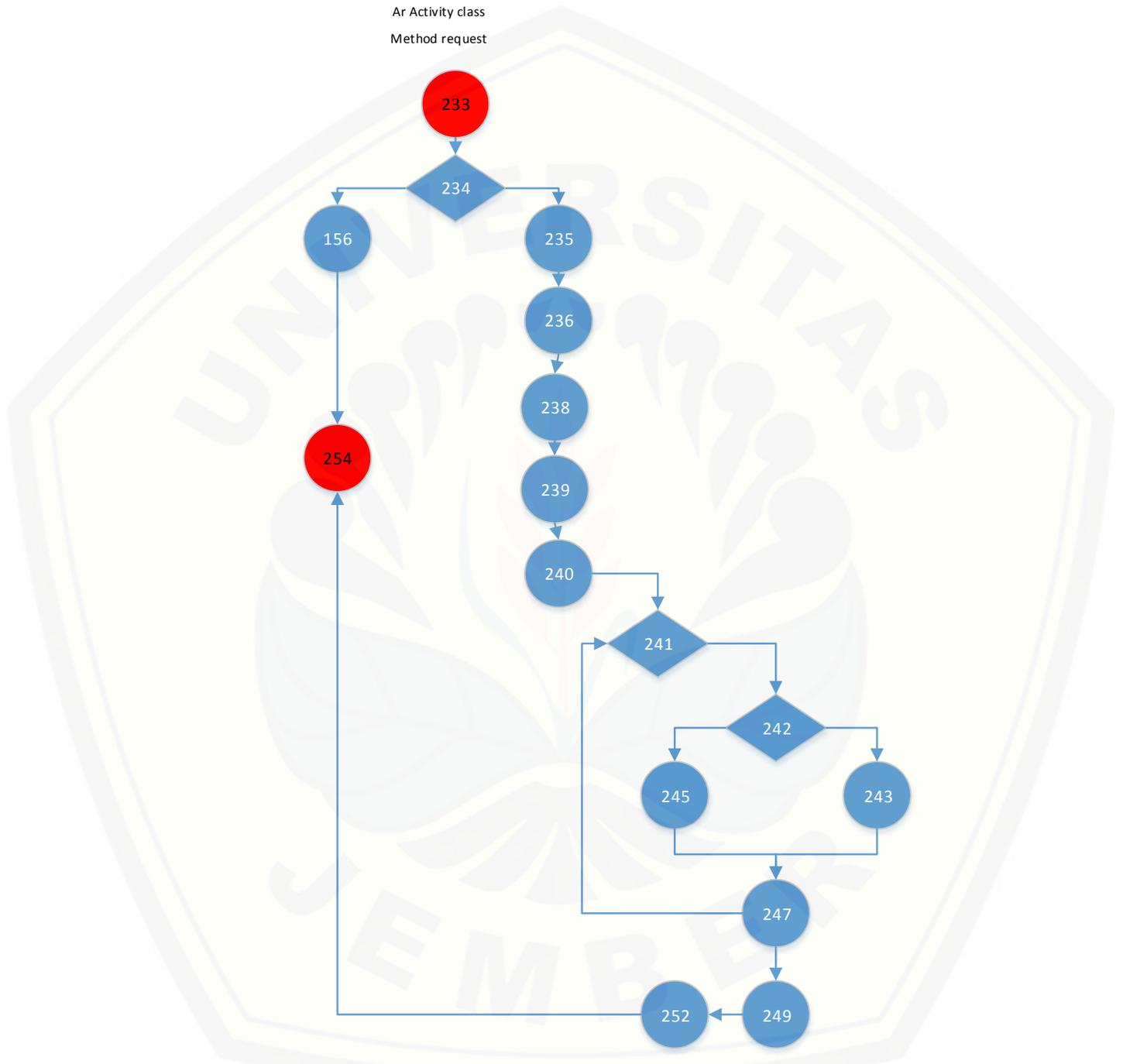
Ar Activity class
Method onLocationChanged



Gambar 25. *Cyclomatic Complexity ARActivity Method onlocationChanged*

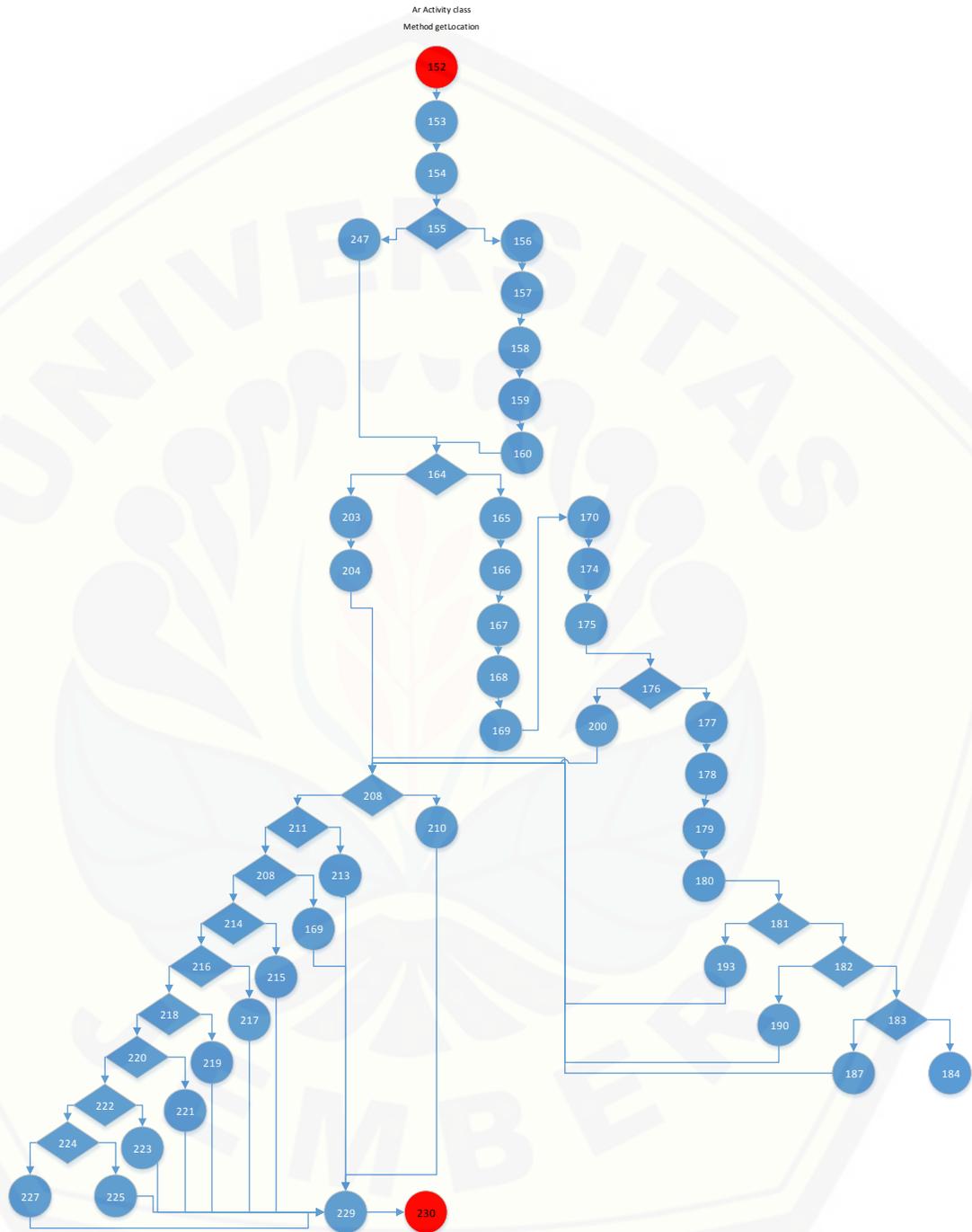
Nilai siklomatik yang didapatkan dari *method onLocationChanged*

$$V(G) = p + 1 = 0 + 1 = 1$$



Gambar 26. Cyclomatic Complexity ARActivity Method request
Nilai siklomatik yang didapatkan dari *method request*

$$V(G) = p + 1 = 3 + 1 = 4$$



Gambar 27. Cyclomatic Complexity ARActivity Method getLocation

Nilai siklomatik yang didapatkan dari *method getLocation*

$$V(G) = p + 1 = 15 + 1 = 16$$

C. Pengujian *Black Box*

Tabel 4. Tabel pengujian *black box*

Skenario Normal			
NO	USER	SISTEM	KESIMPULAN
1	Menekan tombol aplikasi		<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
2		Menampilkan loading screen	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
3		Mengecek ketersediaan GPS	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
4		Masuk ke halaman Home	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
5		Menampilkan <i>progress dialog getting location</i>	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
6		Sistem menampilkan halaman home dan lokasi	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
7	User memilih lokasi yang tersedia		<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
8	User menekan tombol GO		<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
9		Sistem menuju ke halaman map	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
10		Menampilkan <i>progress dialog getting location</i>	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
11		Menampilkan rute pada halaman map	<input checked="" type="checkbox"/> berhasil

		<input type="checkbox"/> gagal
12	User menekan tombol AR View	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
13	Menuju halaman AR View	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
14	Sistem menampilkan AR View	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal

Skenario alternatif

	USER	SISTEM
3.1		Tidak menemukan ketersediaan GPS <input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
3.2		Menampilkan <i>progress dialog</i> untuk menuju ke halaman setting lokasi <input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
3.3	Menekan tombol GPS	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
3.4	Jika menekan ya	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
3.5		Menuju halaman setting lokasi <input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
3.6	Menekan switch aktifkan gps	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
3.7		Keluar dari aplikasi <input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
3.8	Jika menekan tidak	<input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal
3.9		Keluar dari aplikasi <input checked="" type="checkbox"/> berhasil <input type="checkbox"/> gagal

D. Pengujian Pendekatan Skenario

1. *Test Case* dari *User Interaction Diagram*

Tabel 5. *Test Case* dari *user interaction diagram* pada *splash activity*

Test Preparation		Pengguna menekan ikon aplikasi	
No	status	Kondisi	Keluaran
1	<i>Delay</i> 3 detik	aplikasi mulai	<i>Splash screen</i> tampil
2	<i>Delay</i> 3 detik	Aplikasi mulai mengecek GPS	<i>Progress dialog</i> muncul
3	Cek <i>GPS</i>	Aplikasi menemukan GPS	Menuju halaman Home
4	Cek <i>GPS</i>	Aplikasi tidak dapat menemukan GPS	<i>Alert Dialog</i> muncul
	<i>Alert Dialog</i>	Memilih pilihan keluar	Keluar aplikasi
5	Cek <i>GPS</i>	Aplikasi tidak dapat menemukan GPS	<i>Alert Dialog</i> muncul
6	<i>Alert Dialog</i>	Memilih pilihan setting	Masuk ke halaman <i>setting</i>
7	Halaman <i>setting</i>	Mengaktifkan <i>GPS</i>	<i>Alert dialog</i> muncul
8	<i>Alert Dialog</i>	Menekan <i>Ok</i>	Keluar dari aplikasi

Tabel 6. *Test Case* dari *user interaction diagram* pada *Home activity*

Test Preparation		Aplikasi menemukan GPS	
No	Status	Kondisi	Keluaran
1	Mengambil lokasi pengguna	Aplikasi mengambil data lokasi pengguna	<i>Progress dialog</i>
2	<i>Progress dialog</i>	Aplikasi mengambil data lokasi gedung	<i>Progress dialog</i>
3	Data telah didapatkan	Masuk ke halaman <i>home</i>	<i>Halaman home</i>

4	Halaman home	Pengguna memilih lokasi tujuan yang ada di <i>spinner</i>	Halaman <i>home</i>
4	Halaman <i>home</i>	Pengguna menekan tombol <i>go</i>	Menuju halaman <i>navigation activity</i>

Tabel 7. *Test Case* dari *user interaction diagram* pada *Navigation Activity*

Test Preparation		User menekan tombol <i>go</i> di halaman <i>Home</i>	
No	Status	Kondisi	Keluaran
1	Mengirim data ke <i>server google maps</i>	Data akan dikirim ke <i>server</i>	<i>Progress dialog</i>
2	<i>Progress dialog</i>	Data <i>direction</i> diterima oleh aplikasi	<i>Progress dialog</i>
3	<i>Map</i>	Rute lokasi pengguna dan tujuan di <i>overlay</i> ke map	<i>Map</i>
3	<i>Map</i>	Pengguna menekan tombol <i>AR view</i>	Menuju halaman <i>AR view</i>

Tabel 8. . *Test Case* dari *user interaction diagram* pada *ARActivity*

Test Preparation		User menekan tombol <i>AR View</i> di Halaman <i>Navigation</i>	
No	Status	Kondisi	keluaran
1	Halaman <i>AR view</i>	Mengirim data ke <i>server</i>	Halaman <i>AR view</i>
2	Halaman <i>AR view</i>	Data diterima dan diparsing sesuai dengan kebutuhan	Petunjuk arah, jarak dan instruksi berubah sesuai data yang diterima oleh aplikasi

