



# **RANCANG BANGUN APLIKASI PENENTUAN LOKASI LAYANAN UMUM MENGGUNAKAN AUGMENTED REALITY BERBASIS ANDROID**

**TUGAS AKHIR**

**Program Studi**

**S1 Sistem Informasi**

**INSTITUT BISNIS  
DAN INFORMATIKA**

**stikom**  
SURABAYA

**Oleh:**

**LESTANTO ATMAJI**

**09410100073**

## DAFTAR ISI

	Halaman
ABSTRAK .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI .....	ix
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR .....	xiv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	4
1.5 Manfaat Penelitian .....	4
1.6 Sistematika Penulisan .....	4
BAB II LANDASAN TEORI .....	6
2.1 Pengertian Augmented Reality .....	6
2.2 Sejarah Augmented Reality .....	6
2.3 Global Positioning System (GPS) .....	9
2.4 Cara penentuan lokasi pada GPS .....	10
2.5 Akurasi GPS .....	10
2.6 Android .....	11
2.7 Platform Google Android .....	12
2.8 Life Cycle Aplikasi Android .....	13
2.9 Dalvik Virtual Machine (DVM) .....	15

2.10	Android SDK (Software Development Kit).....	16
2.11	Anatomi Aplikasi Android.....	17
2.12	Arsitektur Android .....	18
2.13	Perhitungan sudut azimuth.....	19
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM .....</b>		<b>21</b>
3.1	Analisa Masalah .....	21
3.1.1	Arsitektur aplikasi .....	22
3.1.2	Arsitektur aplikasi pada Smartphone Android .....	23
3.2	Use Case Diagram .....	24
3.3	Activity Diagram .....	26
3.3.1	Activity Diagram List Menu Lokasi Layanan Umum .....	26
3.3.2	Activity Diagram Detail Lokasi Layanan Umum .....	27
3.3.3	Activity Diagram Augmented Reality View.....	28
3.3.4	Activity Diagram Memilih Radius.....	30
3.4	Sequence Diagram.....	31
3.4.1	Sequence Diagram List Menu Lokasi Layanan Umum .....	31
3.4.2	Sequence Diagram Detail Lokasi Layanan Umum .....	32
3.4.3	Sequence Diagram Augmented Reality View.....	33
3.4.4	Sequence Diagram Memilih Radius.....	34
3.5	Class Diagram.....	35
3.5.1	Class SensorActivity .....	37
3.5.2	Class AugmentedView .....	37
3.5.3	Class AugmentedActivity .....	38

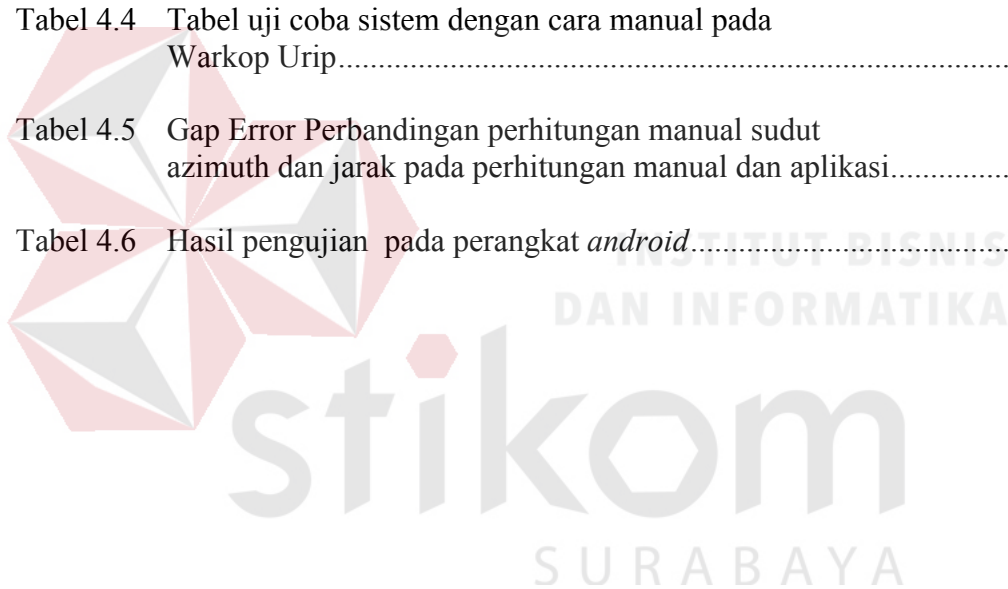
	Halaman
3.5.4 Class MainActivity .....	38
3.5.5 Class DataSource .....	39
3.5.6 Class NetworkDataSource .....	40
3.5.7 Class GoogleDataSource .....	40
3.5.8 Class PhysicalLocationUtility.....	41
3.5.9 Class ScreenPositionUtility .....	41
3.5.10 Class PaintableObject .....	42
3.5.11 Class PaintableBox .....	42
3.5.12 Class PaintableBoxedText .....	43
3.5.13 Class PaintableCircle .....	43
3.5.14 Class PaintableGps .....	44
3.5.15 Class PaintableIcon.....	44
3.5.16 Class PaintableLine.....	45
3.5.17 Class PaintablePoint .....	45
3.5.18 Class PaintablePosition.....	46
3.5.19 Class PaintableRadarPoints .....	46
3.5.20 Class PaintableText.....	47
3.6 Desain Input Output.....	47
3.6.1 Layout List Menu.....	47
3.6.2 Layout Augmented Reality View .....	48
A. Gadget Radar Pada Layout Augmented Reality View.....	49
B. Gadget Augmented Reality Item.....	50
BAB IV IMPLEMENTASI DAN EVALUASI.....	52



	Halaman
4.1 Kebutuhan Sistem .....	52
4.2 Implementasi Sistem .....	52
4.2.1 Google Map API .....	52
A. Mengambil data lokasi layanan umum terdekat dengan memanfaatkan Google Map API pada aplikasi .....	53
B. Input data lokasi layanan umum dengan memanfaatkan Google Map.....	55
4.2.2 Form Splash Screen .....	57
4.2.3 Form List Menu .....	58
4.2.4 Form Augmented Reality View .....	59
A. Gadget Radar Pada form Augmented Reality View.....	61
B. Gadget Augmented Reality Item Pada Layout Augmented Reality View .....	62
4.3 Evaluasi .....	63
4.3.1 Uji coba fungsi aplikasi.....	63
4.3.2 Uji coba kompabilitas system .....	73
BAB V PENUTUP.....	78
5.1 Kesimpulan.....	78
5.2 Saran .....	79
DAFTAR PUSTAKA .....	80

## DAFTAR TABEL

	Halaman
Tabel 3.1 Tabel keterangan <i>use case</i> aplikasi penentuan lokasi layanan umum .....	28
Tabel 4.1 tabel uji coba sistem dengan cara manual pada ATM Bank BCA 6932-Tropodo .....	64
Tabel 4.2 Tabel uji coba sistem dengan cara manual pada MM Resto Raya Bypass Juanda Baru .....	66
Tabel 4.3 Tabel uji coba sistem dengan cara manual pada Hotel Sinar II .....	69
Tabel 4.4 Tabel uji coba sistem dengan cara manual pada Warkop Urip .....	70
Tabel 4.5 Gap Error Perbandingan perhitungan manual sudut azimuth dan jarak pada perhitungan manual dan aplikasi .....	73
Tabel 4.6 Hasil pengujian pada perangkat <i>android</i> .....	75



## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Taksonomi Isu Pervasive Computing.....	7
Gambar 2.2 Rumus perhitungan sudut azimuth .....	20
Gambar 2.3 Ilustrasi sudut azimuth.....	20
Gambar 3.1 Arsitektur aplikasi penentuan lokasi layanan umum.....	22
Gambar 3.2 Arsitektur aplikasi pada smartphone Android.....	23
Gambar 3.3 Use Case Diagram Aplikasi Penentuan Lokasi Layanan Umum.....	24
Gambar 3.4 <i>Activity diagram list menu lokasi layanan umum</i> .....	27
Gambar 3.5 <i>Activity Diagram Detail Lokasi Layanan Umum</i> .....	28
Gambar 3.6 <i>Activity Diagram Augmented Reality View</i> .....	29
Gambar 3.7 <i>Activity Diagram Memilih Radius</i> .....	31
Gambar 3.8 <i>Secuence diagram list menu lokasi layanan umum</i> .....	32
Gambar 3.9 <i>Sequence Diagram Detail Lokasi Layanan Umum</i> .....	33
Gambar 3.10 <i>Sequence Diagram Augmented Reality View</i> .....	34
Gambar 3.11 <i>Sequence Diagram Memilih Radius</i> .....	35
Gambar 3.12 <i>Class Aplikasi penentuan lokasi layanan umum</i> .....	36
Gambar 3.13 <i>Class SensorActivity</i> .....	37
Gambar 3.14 <i>Class AugmentedView</i> .....	37
Gambar 3.15 <i>Class AugmentedAcivity</i> .....	38
Gambar 3.16 <i>Class MainActivity</i> .....	39
Gambar 3.17 <i>Class DataSource</i> .....	39
Gambar 3.18 <i>Class NetworkDataSource</i> .....	40
Gambar 3.19 <i>Class GoogleDataSource</i> .....	41

Gambar 3.20	<i>Class PhysicalLocationUtility</i> .....	41
Gambar 3.21	<i>Class ScreenPositionUtility</i> .....	42
Gambar 3.22	<i>Class PaintableObject</i> .....	42
Gambar 3.23	<i>Class PaintableBox</i> .....	43
Gambar 3.24	<i>Class PaintableBoxedText</i> .....	43
Gambar 3.25	<i>Class PaintableCircle</i> .....	44
Gambar 3.26	<i>Class PaintableGPS</i> .....	44
Gambar 3.27	<i>Class PaintableIcon</i> .....	45
Gambar 3.28	<i>Class PaintableLine</i> .....	45
Gambar 3.29	<i>Class PaintablePoint</i> .....	46
Gambar 3.30	<i>Class PaintablePosition</i> .....	46
Gambar 3.31	<i>Class PaintableRadarPoint</i> .....	47
Gambar 3.32	<i>Class PaintableText</i> .....	47
Gambar 3.33	List menu .....	48
Gambar 3.34	Augmented Reality View .....	49
Gambar 3.35	Radar pada <i>layout augmented reality view</i> .....	49
Gambar 3.36	Augmented Reality Item Pada Layout Augmented Reality View .....	49
Gambar 4.1	Gambar <i>sourcode</i> untuk <i>request</i> pencarian lokasi restoran atau <i>café</i> terdekat .....	53
Gambar 4.2	<i>Google Map API</i> mengembalikan nilai <i>output</i> berupa <i>JSON</i> yang berisi informasi yang lengkap tentang lokasi terdekatnya ....	54
Gambar 4.3	<i>Website Google for Business</i> halaman input detail lokasi bisnis ..	55
Gambar 4.4	<i>Website Google for Business</i> halaman verifikasi .....	56
Gambar 4.5	<i>Website Google for Business</i> pada halaman <i>dashboard</i> .....	56

Gambar 4.6	<i>Splash screen</i> aplikasi.....	57
Gambar 4.7	<i>Form List Menu</i> .....	59
Gambar 4.8	<i>Form Augmented Reality View</i> .....	60
Gambar 4.9	Radar pada <i>layout augmented reality view</i> .....	62
Gambar 4.10	<i>Augmented Reality</i> Item pada <i>form Augmented Reality View</i> .....	63
Gambar 4.11	Uji coba aplikasi pada ATM Bank BCA 6932-Tropodo.....	65
Gambar 4.12	Uji coba aplikasi pihak ketiga pada ATM Bank BCA 6932-Tropodo .....	66
Gambar 4.13	Uji coba aplikasi pada MM Resto Raya Bypass Juanda Baru.....	67
Gambar 4.14	Uji coba aplikasi pihak ketiga pada MM Resto Raya Bypass Juanda Baru.....	68
Gambar 4.15	Uji coba pada Hotel Sinar II.....	69
Gambar 4.16	Uji coba aplikasi pihak ketiga pada Hotel Sinar II.....	70
Gambar 4.17	Uji coba pada Warkop Urip.....	71
Gambar 4.18	Uji coba aplikasi pihak ketiga pada Warkop Urip.....	72
Gambar 4.19	Aplikasi berjalan pada <i>device android, galaxy S2</i> dan <i>operating system gingerbread(2.3.4)</i> .....	74
Gambar 4.20	aplikasi berjalan pada <i>device android, NEXUS 4</i> dan <i>operating system Kitkat(4.4)</i> .....	74
Gambar 4.21	Radar menunjukan lokasi sejajar pada arah pandang pengguna ...	75
Gambar 4.22	Perangkat menunjukan lokasi pada langit-langit karena lokasi sejajar pada arah pandang pengguna.....	77

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Layanan umum adalah tempat untuk memenuhi kebutuhan tertentu, di mana layanan umum tersebut meliputi restoran, hotel, stasiun pengisian bahan bakar umum (SPBU), dan *automatic teller machine (ATM)*. Pada saat ini pencarian lokasi layanan umum telah di dukung oleh perangkat pencarian lokasi yang biasa disebut sistem navigasi atau pemandu arah. Sistem navigasi pengetahuan tentang petunjuk arah atau kompas dan peta serta teknik penggunaannya haruslah dimiliki dan dipahami. Sistem tersebut telah dikembangkan pada sistem navigasi kendaraan, yaitu perangkat navigasi berkendaraan modern yang menggunakan perangkat peta digital dan informasi posisi dengan satelit *GPS* sebagai alat bantu navigasi. Maka sistem navigasi dapat digunakan untuk memandu pengendara, sehingga pengendara bisa mengetahui jalur mana yang sebaiknya dipilih untuk mencapai tujuan yang diinginkan. Sistem navigasi kendaraan sudah merupakan perlengkapan standar kendaraan mewah, dan bisa dibeli perangkat portabel dari penjual sistem navigasi kendaraan. Beberapa jenis kendaraan yang dilengkapi *GPS*.

Masalah yang ditemukan adalah tidak semua pengguna dapat membaca peta, baik itu dalam bentuk fisik maupun aplikasi penyedia layanan peta. Mereka mengalami kesulitan dalam proses pengenalan daerah yang baru mereka kunjungi dengan peta yang disediakan, sebab itulah pengetahuan tentang petunjuk arah atau

kompas dan peta serta teknik penggunaannya haruslah dimiliki dan dipahami. Pada saat menggunakan perangkat navigasi, pengguna harus memasukan nama lokasi yang dituju. Dalam melakukan hal tersebut, dapat terjadi kesalahan dimana pengguna salah mengetik nama lokasi yang dituju. Proses itu kami rasa kurang efektif. Untuk meminimalisasi hal tersebut maka pengguna dapat memanfaatkan aplikasi pencarian lokasi layanan umum dengan menggunakan *Augmented Reality* dimana pengguna tidak harus mengetik nama lokasi yang dituju, melainkan dapat langsung mengarahkan kamera pada *smartphone android*-nya ke arah lokasi yang akan dituju.

Berdasarkan uraian di atas maka diusulkan Aplikasi *GPS* dan *GIS* untuk mencari lokasi dan jarak layanan umum dengan Peta dan *Augmented Reality Camera-View* pada Perangkat Bergerak Berbasis *Android*. Aplikasi tersebut dapat memudahkan pencarian lokasi layanan umum yang dibutuhkan hanya dengan memilih pada menu lokasi layanan umum misalnya SPBU, restoran, dan hotel, tanpa harus mengetik nama lokasi yang dibutuhkan, sehingga dapat mempercepat pencarian lokasi layanan umum. Aplikasi tersebut melakukan *tracking* lokasi kita saat ini (menentukan lokasi di mana kita berada), lalu memanggil ke *web service* untuk pencarian lokasi layanan umum dengan radius *default* seratus meter dari lokasi kita berada, kemudian dengan menggunakan *augmented reality* yaitu dengan menggunakan kamera pada *device Android* yang digabungkan dengan obyek maya (penanda lokasi layanan umum). Aplikasi pencarian layanan umum tersebut mempunyai sudut pandang berbeda dalam sistem navigasi, yaitu dengan kondisi nyata pada saat ini secara *Real-Time* yang diharapkan akan mudah

dipahami dan dimengerti dari pada harus melihat peta yang membutuhkan pengetahuan dan pemahaman tentang membaca peta dan arah mata angin.

## 1.2 Perumusan masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas, maka dapat dirumuskan permasalahan dalam tugas akhir ini adalah bagaimana merancang bangun aplikasi penentuan lokasi layanan umum yaitu hotel, SPBU (stasiun pengisian bahan bakar), restoran, dan *ATM (automatic teller machine)* pada *smartphone Android* yang mampu menentukan lokasi layanan umum dengan menggunakan *augmented reality*.

## 1.3 Batasan Masalah

Adapun batasan masalah yang digunakan di dalam tugas akhir ini yaitu :

1. Aplikasi ini hanya bisa diterapkan pada *smartphone android* yang memiliki fitur sensor *GPS*, kompas, dan *accelerometer* dan mensyaratkan ketiga sensor tersebut aktif (*on*).
2. Aplikasi yang dibuat membutuhkan koneksi internet dan menggunakan *database online* sebagai media penyimpanan data lokasi layanan umum.
3. Ruang lingkup data layanan umum yaitu hotel, SPBU (stasiun pengisian bahan bakar), restoran, dan *ATM (automatic teller machine)* dalam aplikasi ini hanya sebatas yang di peroleh dari masukan penulis.
4. Aplikasi ini berbasis *Android* dengan versi 2.3.7 (*Gingerbread*).
5. Bahasa pemrograman yang dipakai adalah *Java* dengan *framework Android SDK*.



6. Aplikasi penentuan lokasi layanan umum menggunakan *augmented reality* ini adalah *location based AR* jadi aplikasi ini tidak berinteraksi dengan gambar yang ditangkap oleh kamera (*image processing*), melainkan menggunakan titik lokasi untuk menambahkan informasi ke dalam pemandangan nyata.

#### **1.4 Tujuan**

Berdasarkan masalah di atas maka tujuan dari dalam tugas akhir ini adalah bagaimana merancang bangun aplikasi penentuan lokasi layanan umum yaitu hotel, SPBU (stasiun pengisian bahan bakar), restoran, dan *ATM (automatic teller machine)* pada *smartphone Android* yang mampu menentukan lokasi layanan umum dengan menggunakan *augmented reality*.

#### **1.5 Manfaat Penelitian**

Manfaat penelitian yang diharapkan dari aplikasi penentuan lokasi SPBU berbasis *Android* ini adalah dapat memudahkan pencarian lokasi layanan umum terdekat dengan informasi jarak lokasi tersebut dari lokasi saat ini.

#### **1.6 Sistematika Penulisan**

Laporan Tugas Akhir (TA) ini ditulis dengan sistematika penulisan sebagai berikut:

#### **Bab I : Pendahuluan**

Bab ini berisi tentang latar belakang diambilnya topik Tugas Akhir, rumusan masalah dari topik Tugas Akhir, batasan masalah atau ruang lingkup pekerjaan Tugas Akhir, dan tujuan dari Tugas Akhir ini.

## **Bab II : Landasan Teori**

Bab ini menjelaskan tentang teori yang mendukung pokok pembahasan tugas akhir yang meliputi definisi dan konsep pariwisata pengertian GPS, Google Android, Google Maps, Google Maps API.

## **Bab III : Metode Penelitian dan Perancangan Sistem**

Bab ini berisi penjelasan tentang tahap-tahap yang dikerjakan dalam penyelesaian Tugas Akhir yang terdiri dari studi pustaka, identifikasi masalah dan tujuan, analisis sistem dengan menggunakan UML (Unified Modeling Language), dan desain antarmuka, dan implementasi menggunakan smartphone android.

## **Bab IV : Evaluasi dan Implementasi**

Dalam bab ini berisi penjelasan tentang evaluasi dari sistem yang telah dibuat secara keseluruhan serta melakukan pengujian terhadap aplikasi, apakah dapat menyelesaikan masalah. Pengujian dilakukan dengan uji coba manual apakah sama dengan yang di terapkan dengan sistem, serta analisis terhadap komabilitas dari sistem.

## **Bab V : Penutup**

Bab ini berisi kesimpulan dan saran. Saran yang dimaksud adalah saran terhadap kekurangan dari aplikasi yang ada kepada pihak lain yang ingin meneruskan topik TA ini. Tujuannya adalah agar pihak lain tersebut dapat menyempurnakan aplikasi sehingga bisa menjadi lebih baik dan berguna.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian *Augmented Reality***

*Augmented Reality (AR)* adalah suatu teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata. Sistem ini lebih dekat kepada lingkungan nyata (*real*). Karena itu, unsur *reality* lebih diutamakan pada sistem ini. Sistem ini berbeda dengan *virtual reality* yang sepenuhnya merupakan *virtual environment (VR)*. *Augmented Reality* mengizinkan pengguna untuk berinteraksi secara *real-time* dengan sistem. (Azuma, 1997) mendefinisikan *augmented reality* sebagai 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.

#### **2.2 Sejarah *Augmented Reality***

*Augmented Reality (AR)* merupakan salah satu terapan aplikasi pervasive computing. Kemunculan AR diawali dengan konsep kontinum realitas-virtualitas. Kontinum realitas-virtualitas mendefinisikan kerangka umum realitas campuran dan menggambarkan hubungan antara dunia nyata dan dunia virtual (Milgram and Kishino, 1994).

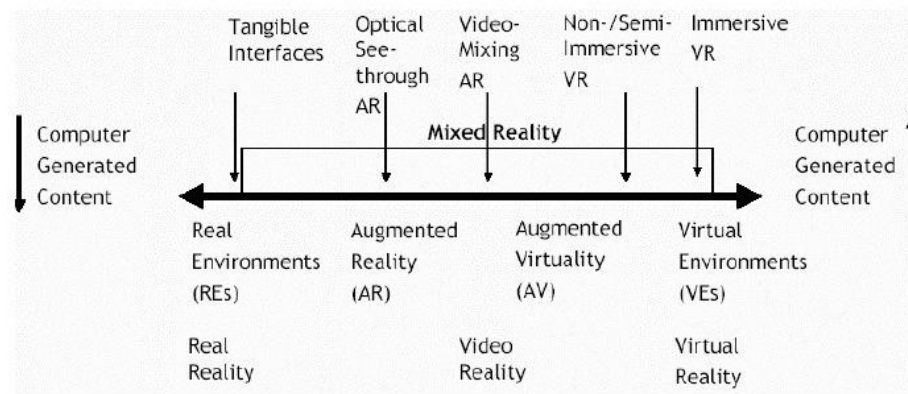


Figure 1-3: Reality-virtuality Continuum (Milgram and Kishino 1994)

Gambar 2.1. Taksonomi Isu Pervasive Computing (Milgram and Kishino,1994).

Pada Gambar 2.1, ditunjukkan bahwa sisi paling kiri merupakan lingkungan nyata yang hanya berisi objek- objek nyata, dan sisi paling kanan merupakan lingkungan virtual yang berisi objek virtual. Dalam realitas ditambah (*augmented reality*), yang lebih dekat ke sisi kiri, maka lingkungan bersifat nyata dan objek bersifat virtual. Sementara dalam virtualitas ditambah (*virtual reality*), yang lebih dekat ke sisi kanan, maka lingkungan bersifat virtual dan objek bersifat nyata. *Augmented Reality* dan *Virtual Reality* jika digabungkan akan menjadi realitas campuran (*mixed reality*).

Secara umum, AR mempunyai tiga karakteristik utama yaitu a) mengkombinasikan dunia nyata dan dunia virtual; b) interaktif dalam waktu nyata (*real time*); dan c) terdaftar dalam 3D. Dengan kata lain, sistem AR merupakan sebuah sistem yang dapat menggabungkan informasi yang diolah oleh perangkat dengan lingkungan nyata, secara interaktif dan pada waktu nyata, serta mendampirkan objek virtual dan objek fisik. AR juga dapat diartikan sebagai salah satu bentuk interaksi manusia-komputer yang dapat menambahkan objek virtual ke dalam penginderaan nyata yang didukung oleh kamera video dalam

waktu nyata. Dalam bahasa yang sederhana, AR dapat dipahami sebagai teknologi yang mengizinkan grafis virtual yang dibangkitkan komputer yang secara tepat melapisi objek fisik dalam waktu nyata.

Sejarah tentang *augmented reality* dimulai dari tahun 1957-1962, ketika seorang penemu bernama Morton Heilig, seorang *cinematographer*, menciptakan dan mematenkan sebuah simulator yang disebut *Sensorama* dengan visual, getaran dan bau. Pada tahun 1966, Ivan Sutherland menemukan *head-mounted display* yang dia *claim* adalah, jendela ke dunia virtual. Tahun 1975 seorang ilmuwan bernama Myron Kruger menemukan *Videoplace* yang memungkinkan pengguna dapat berinteraksi dengan obyek virtual untuk pertama kalinya. Tahun 1989, Jaron Lanier, memperkenalkan *Virtual Reality* dan menciptakan bisnis komersial pertama kalinya di dunia maya, tahun 1992 mengembangkan *Augmented Reality* untuk melakukan perbaikan pada pesawat *Boeing*, dan pada tahun yang sama, LB Rosenberg mengembangkan salah satu fungsi AR, yang disebut *Virtual Fixtures*, yang digunakan di Angkatan Udara AS Armstrong Labs, dan menunjukan manfaatnya pada manusia, dan pada tahun 1992 juga, Steven Feiner, Blair MacIntyre dan Doore Seligmann, memperkenalkan untuk pertama kalinya *Major Paper* untuk perkembangan *Prototype AR*.

(Sood, 2012) Augmented Reality juga dimanfaatkan oleh militer pada pesawat tempur mereka. Pola interaksi Tampilan penargetan pada pesawat adalah yang paling tertua dari pola interaksi AR. Pengalaman Augmented menggunakan pola Tampilan Head-Up pada kokpit pesawat menambahkan informasi tentang benda-benda nyata dalam pandangan menjadi augmented reality.

### 2.3 Global Positioning System (GPS)

*Global Positioning System (GPS)* adalah suatu sistem radio navigasi penentuan posisi dengan menggunakan satelit. *GPS* dapat memberikan posisi suatu objek di muka bumi secara cepat, akurat (tiga dimensi koordinat  $x,y,z$  ) dan memberikan informasi waktu serta kecepatan bergerak secara kontinu di seluruh dunia pada setiap saat tanpa bergantung cuaca (Abidin, 2000). *GPS* merupakan sistem navigasi atau sistem penentuan posisi dari beberapa mm sampai beberapa meter dan ketelitian kecepatan sampai beberapa cm/detik serta ketelitian waktu sampai *nano* detik.

Sistem *GPS* terdiri dari tiga segmen utama, yaitu segmen kontrol (*control segment*) yang terdiri dari stasiun-stasiun pemonitor dan pengontrol satelit, segmen satelit (*space segment*) yang terdiri dari pengguna sipil dan militer. Satelit *GPS* mempunyai konstelasi 24 satelit dalam enam orbit yang mendekati lingkaran. Setiap orbit ditempati oleh 4 buah satelit dengan interval antara yang tidak sama. Orbit satelit *GPS* berinklinasi 55° terhadap bidang *equator* dengan ketinggian rata-rata dari permukaan bumi sekitar 10.100 km (Haryanto, 2004). Dengan adanya 24 satelit di angkasa, 4 sampai dengan 10 satelit *GPS* setiap saat akan selalu dapat diamati di seluruh permukaan bumi. Sinyal satelit *GPS* dipancarkan secara *broadcast* oleh satelit *GPS* secara kontinu serta tidak tergantung cuaca. Dengan mengamati sinyal satelit menggunakan *receiver GPS* seseorang dapat menentukan posisi (lintang, bujur) di permukaan bumi. Informasi lainnya yang didapat dari satelit *GPS* selain posisi adalah kecepatan arah, jarak, dan waktu.

## 2.4 Cara penentuan lokasi pada GPS

Satelit *GPS* mengorbit bumi dua kali dalam sehari dengan lintasan yang sangat presisi dan mentransmisikan sinyal informasi secara *continue* ke bumi. *GPS receiver* memanfaatkan informasi ini dengan berperan sebagai sebuah alat pengukur yang menghitung jarak antara antenna *receiver* dengan berbagai satelit *GPS*. Kemudian *GPS receiver* menetapkan posisi melalui proses *trilateration* dengan mencari perpotongan tiap vektor satelit-satelit tersebut. Jarak antara antenna *receiver* dengan waktu ketika sinyal diterima.

Sebuah *GPS receiver* setidaknya harus menangkap sinyal dari 3 buah satelit untuk menghitung posisi 2 dimensi (bujur dan lintang) dan pergerakannya. Dengan 4 buah satelit atau lebih, *receiver* dapat menghitung posisi 3 dimensi (bujur, lintang, dan ketinggian dari permukaan laut).

## 2.5 Akurasi GPS

*GPS* menyediakan posisi dengan ketepatan akurasi 15 meter, yang berarti jika *GPS receiver* memberikan koordinat terhadap sesuatu lokasi tertentu, maka boleh diharapkan lokasi sebenarnya berada dalam radius 15 meter dari koordinat tersebut (El-Rabbany, 2002). Ketepatan *GPS* tergantung daripada lokasi *GPS receiver*-nya dan halangan terhadap sinyal satelit *GPS*. Meski secara umum, *GPS* menawarkan tingkat ketelitian 135 meter, namun akurasi ini dapat ditingkatkan dengan berbagai teknik, seperti *Assisted GPS (A-GPS)*, *Differential GPS (D-GPS)*, atau *Wide Area Augmentation System (WAAS)*. Pada kenyataannya di lapangan, akurasi dapat bervariasi sesuai keadaan.

## 2.6 Android

*Android* adalah sistem operasi untuk *smartphone* yang dibuat oleh *Google Corporation*. Sistem operasi ini dikembangkan dengan memanfaatkan *linux kernel*. (*Google*, 2010). Perencanaan Sistem dan Teori Penunjang *Android*. Versi *stable* terbaru dari *Android* adalah *Honeycomb* 3.1 yang dirilis pada bulan Mei 2011 dengan *linux* versi 2.6.37. pada versi ini fitur *Android* sudah dilengkapi dengan *High Performance WiFi Lock*, *Download Manager*, *Streaming Audio formats* dan juga *browser Chrome* versi 8 menggunakan HTML5. Pada versi 3.0 ke atas perangkat selular yang digunakan tidak lagi perangkat selular genggam melainkan menggunakan perangkat selular berbentuk tablet, yang memiliki ukuran lebih besar, kurang lebih dua kali lipat di banding perangkat selular biasa.

Dari segi arsitektur sistem, *Android* merupakan sekumpulan *framework* dan *virtual machine* yang berjalan di atas *kernel linux*. *Virtual machine android* bernama *Dalvik Virtual Machine* (DVM), *engine* ini berfungsi untuk menginterpretasikan dan menghubungkan seluruh kode mesin yang digunakan oleh setiap aplikasi dengan menghubungkan seluruh kode mesin yang digunakan oleh setiap aplikasi dengan *kernel linux*. Sementara untuk *framework* aplikasi sebagian besar dikembangkan oleh *Google* dan sebagian yang lain dikembangkan oleh pihak ketiga (*developer*). Beberapa *framework* yang dikembangkan oleh *Android* sendiri misalnya fungsi untuk telepon seperti panggilan telepon, SMS, *video call*. Untuk *browser Android* menggunakan *Google Chrome*.

Aplikasi yang dikembangkan di atas *Android* dibuat dengan menggunakan kode *Java* seperti halnya J2ME yang telah lama digunakan pada *platform* perangkat selular umumnya. Namun secara siklus program memiliki



perbedaan mendasar antara J2ME dengan *Java* yang ada pada *Android*. Kode *Java* pada *Android* lebih dekat dengan J2SE, hingga saat ini *Android* telah banyak digunakan pada produk *smartphone* seperti *Samsung*, *LG*, *SonyEricson*, *Nexian* dan juga *HTC*. Dengan dukungan *Software Development Kit (SDK)* dan *Application Programming Interface (API)* dari *Google* memberikan kemudahan bagi pihak ketiga (*developer*) untuk membangun aplikasi yang dapat berjalan pada sistem operasi *Android*.

## 2.7 Platform Google Android

*Android* merupakan suatu *software stack* untuk *mobile device* (*Google inc*, 2003). Di dalamnya terdapat sistem operasi, *middleware*, dan *key application*. Aplikasi pada *platform* ini dikembangkan dalam bahasa pemrograman *Java*. *Android* ini memiliki banyak fitur di antaranya adalah:

1. Merupakan sebuah *Application Framework* sehingga *programmer* dapat menggunakan beberapa fungsi yang telah disediakan.
2. *Dalvik virtual machine*. Tapi aplikasi dalam *Android* memiliki *instance virtual machine* yang dapat bekerja secara efisien dalam lingkungan memori yang terbatas.
3. *Integrated browser web browser* berbasis *WebKit engine* terdapat pada *browser default Android* atau pun dapat diintegrasikan dengan aplikasi lain.
4. *Optimized graphic Libary* grafis 2D yang kaya dan 3D berbasis *OpenGL ES 1.0* yang mendukung akselerasi *hardware*.
5. *SQLite*. Basis data relasional yang ringan namun sangat *powerfull*.
6. *Media Support*. Mendukung berbagai format audio, video, dan gambar (MPEG4,H264,MP3,AAC,AMR, JPG,PNG,GIF).

7. *GSM Telephone*, Mendukung fungsi komunikasi GSM.
8. *Bluetooth*. EDGE, 3G, dan WiFi. Mendukung komunikasi pada jaringan (tergantung *hardware*).
9. Kamera, *GPS*, kompas dan *accelerometer*. Mendukung berbagai fitur yang disediakan oleh *hardware*.
10. *Device emulator*, *tools* untuk *debugging*, *profiling* memori dan performa, *plug-in* untuk *Eclipse IDE*.

## 2.8 Life Cycle Aplikasi Android

Pada banyak kasus, tiap aplikasi pada *Android* masing-masing memiliki *Linux* proses. Proses ini diciptakan untuk aplikasi tersebut pada saat kode program tersebut akan dieksekusi, akan terus berjalan sampai tidak dibutuhkan lagi dan sistem memerlukan memori untuk aplikasi lain.

Konsep dasar pemrograman *Android* adalah suatu proses pada aplikasi tidak langsung diatur oleh aplikasi tersebut. Proses tersebut ditentukan oleh sistem melalui kombinasi: informasi aplikasi bagi sistem, seberapa pentingnya aplikasi tersebut bagi pengguna dan ketersediaan memori.

Contoh kasus *life-cycle bug* adalah *IntentReceiver* yang menciptakan *thread* pada saat menerima *intent* pada *method onReceiveIntent* dan selesai dari fungsi tersebut. Setelah fungsi tersebut selesai, sistem menganggap bahwa *IntentReceiver* tersebut menjadi tidak aktif sehingga ia memiliki proses yang tidak dibutuhkan (kecuali terdapat komponen aplikasi lain di dalamnya). Oleh karena itu, sistem dapat menghapus proses tersebut kapan pun apabila dibutuhkan. Hal tersebut akan menghapus semua *thread* yang berjalan pada proses tersebut. Solusi

masalah ini adalah memakai *Service* dari *IntentReceiver*, sehingga sistem mengetahui bahwa ada sesuatu yang aktif dalam proses tersebut.

Sebagai *programmer*, kita harus mengetahui bagaimana cara yang tepat untuk merancang aplikasi. Kesalahan pada perancangan akan membuat suatu aplikasi akan dihentikan oleh sistem pada saat yang tidak diinginkan. Untuk menentukan proses mana yang akan dihapus pada situasi *low memory*, *Android* memiliki “*importance hierarchy*” berdasarkan komponen dan *state* yang sedang berjalan. Urutan berdasarkan kepentingan tersebut adalah :

1. *Foreground process*. Adalah suatu proses yang dibutuhkan untuk berinteraksi dengan pengguna saat ini. Berbagai komponen suatu aplikasi dapat mengakibatkan proses di dalamnya menjadi *foreground* proses. Hanya akan terdapat beberapa proses seperti itu dalam sistem, proses ini hanya akan dihapus apabila sistem tidak memiliki pilihan lain. Contohnya apabila memori tinggal sedikit sehingga proses *foreground* tidak lagi bisa berjalan. Hal ini diperlukan untuk menjaga *user interface* tetap interaktif dengan *user*.
2. *Visible process*. Adalah proses yang memegang *Activity* yang dapat dilihat *user* pada layar tetapi tidak *foreground* (pada saat *method onPause()* dipanggil). Hal ini dapat terjadi, sebagai contoh *activity foreground* muncul dengan dialog sehingga *activity foreground* muncul dengan dialog sehingga *activity* sebelumnya terlihat di belakangnya. Proses ini sangat penting dan tidak akan dihapus kecuali untuk menjaga aplikasi *foreground* tetap berjalan.
3. *Service process*. Adalah proses yang memegang *service* yang dimulai dengan *method startServices()*. Meskipun proses ini secara langsung tidak dapat dilihat oleh pengguna, biasanya proses ini menjalankan perintah dari pengguna

(seperti memutar lagu, men-*download* data dari jaringan dan lain-lain). Sistem akan tetap membiarkan proses ini berjalan dengan selama proses *foreground* dan *visible* masih bisa jalan.

4. *Background process*. Adalah proses yang memegang *Activity* yang saat ini tidak dapat dilihat oleh *user* (*method onStop()* sedang dipanggil). Proses ini tidak secara langsung mempengaruhi *user*. Apabila *activity life cycle* telah dipenuhi dengan benar, sistem dapat menghapus proses ini sewaktu-waktu untuk memberikan memori tambahan. Biasanya banyak ditemukan proses semacam ini, sehingga proses tersebut disimpan dalam *LRU (least Recently Used)* untuk menjaga agar proses yang baru saja dilihat oleh *user* akan dihapus paling akhir pada kondisi *low memory*.
5. *Empty proses*. Adalah proses yang tidak memegang suatu komponen yang sangat aktif. Alasan mengapa membiarkan proses semacam ini adalah untuk melakukan *cache* sehingga proses *startup* menjadi lebih cepat apabila aplikasi tersebut akan dijalankan kembali. Sistem sering kali menghapus proses semacam ini untuk mencapai keseimbangan dari *system resource* antara *cached proses* dan *kernel caches*.

## 2.9 Dalvik Virtual Machine (DVM)

Salah satu elemen kunci dari *Android* adalah *dalvik virtual machine (DVM)*. *Android* berjalan di dalam *dalvik virtual machine (DVM)* bukan di *Java Virtual Machine (JVM)*, sebenarnya banyak persamaannya dengan *Java Virtual Machine (JVM)* seperti *Java ME (Java Mobile Edition)*, tetapi *Android* menggunakan *Virtual Machine* sendiri yang menurut saya dikostumisasi dan

dirancang untuk memastikan bahwa beberapa fitur berjalan lebih efisien pada perangkat *mobile*.

*Dalvik virtual machine (DVM)* adalah “*register based*” sementara *java virtual machine (JVM)* adalah “*stack based*”, *DVM* didesain oleh Dan Bomsten dan beberapa *engineers Google* lainnya. *DVM* menggunakan *kemel Linux* untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi C / C+ sama halnya seperti pada *OS Linux* kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami Arsitektur dan proses sistem dari *kemel Linux* yang digunakan dalam *Android* tersebut.

Semua *hardware* yang berbasis *Android* dijalankan dengan menggunakan *Virtual machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas bahasa *Java* dan di kompilasi menggunakan *tools* yang disediakan dalam *SDK Android*.

## **2.10 Android SDK (Software Development Kit)**

*Android SDK* adalah *tools API (Application Programming Interface)* yang diperlukan untuk memulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java* (Harahap, 2011). *Android* merupakan *subset* perangkat lunak untuk *smartphone* yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh *Google*.

Saat ini disediakan *Android SDK* sebagai alat bantuan *API* untuk mulai mengembangkan aplikasi pada platform *Android* menggunakan bahasa pemrograman *Java*. Sebagai *platform* aplikasi netral, *Android* memberi anda kesempatan untuk membuat aplikasi yang bukan merupakan aplikasi bawaan *smartphone*.

Beberapa fitur-fitur *Android* yang paling penting adalah :

1. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*.
2. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile*.
3. *Integrated browser* berdasarkan *engine open source WebKit*.
4. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi *OpenGL ES 1.0* (Opsional akselerasi *hardware*).
5. *SQLite* untuk menyimpan data.
6. *Media Support* yang menangani audio, video, dan gambar.
7. *Bluetooth, EDGE, dan WiFi* (tergantung *hardware*).
8. Kamera, *GPS*, kompas, dan *accelerometer* (tergantung *hardware*).
9. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tool* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE *Eclipse*.

## 2.11 Anatomi Aplikasi *Android*

Terdapat empat *building blocks* pada aplikasi *Android* yaitu :

1. *Activity*
2. *Intent Receiver*
3. *Service*
4. *Content Provider*

Tidak semua aplikasi membutuhkan keempat blok ini, tetapi suatu aplikasi dibuat menggunakan kombinasi beberapa blok ini. Setelah memutuskan blok komponen yang akan digunakan, blok tersebut didaftarkan pada suatu *file* yang disebut dengan *AndroidManifest.xml*. *file* XML ini digunakan untuk menyatakan komponen apa saja yang dibutuhkan oleh sebuah aplikasi dan kemampuan serta kebutuhan aplikasi tersebut.

## 2.12 Arsitektur *Android*

Arsitektur sistem terdiri atas 5 *layer*, pemisahan *layer* bertujuan untuk memberikan abstraksi sehingga memudahkan pengembangan aplikasi. *Layer-layer* tersebut adalah *layer* aplikasi, *layer framework* aplikasi, *layer libraries*, *layer runtime*, dan *layer kernel*.

Secara garis besar arsitektur *Android* dapat dijelaskan dan digambarkan sebagai berikut :

### 1. *Applications* dan *widgets*

*Applications* dan *widget* ini adalah layar di mana berhubungan dengan aplikasi saja, di mana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut, di *layer* inilah terdapat seperti aplikasi inti termasuk klien *email*, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman *Java*.

### 2. *Application Frameworks*

*Android* adalah “*Open Development Platform*” yaitu *Android* menawarkan kepada pengembang atau memberi kemampuan kepada penembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resource*, menjalankan *service*

*background*, mengatur *alarm*, dan menambahkan tambahan seperti status *notifications*, dan masih banyak lagi. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang berkategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan komponen yang sudah digunakan (*reuse*).

### 3. *Liberaries*

*Liberaries* ini adalah *layer* di mana fitur-fitur *Android* berada, biasanya para pembuat aplikasi kebanyakan mengakses *liberaries* untuk menjalankan aplikasi. Berjalan di atas *kernel*, *layer* ini meliputi berbagai *library C/ C++* inti seperti *Libe* dan *SSL*.

### 4. *Android Runtime*

*Layer* yang membuat aplikasi *Android* dijalankan di mana dalam prosesnya menggunakan implementasi *Linux*, *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi *Android*.

### 5. *Linux kernel*

*Linux kernel* adalah *layer* di mana inti dari *operating system* dari android itu sendiri, berisi *file-file system* yang mengatur *system processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi *Android* lainnya. *Linux kernel* yang digunakan *Android* adalah *Linux kernel release 2.6*.

## 2.13 Perhitungan sudut *azimuth*

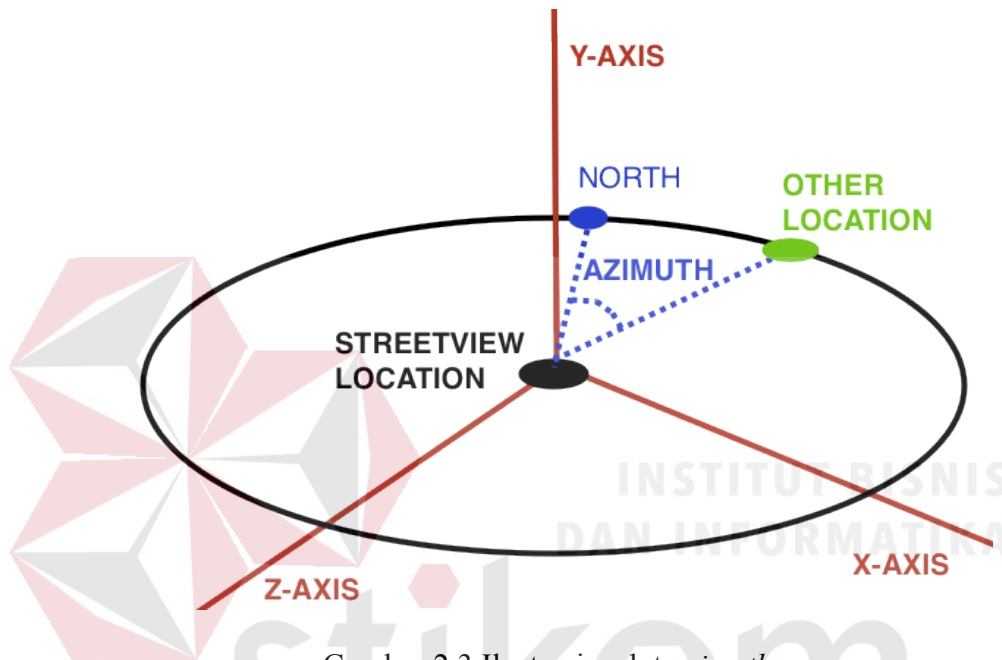
*Azimuth* yaitu sudut antara utara magnetis (nol derajat) dengan titik/sasaran yang kita tuju relatif terhadap sudut utara. Sudut *azimuth* pada masing-masing titik dihitung berdasarkan sudut *arc-tangent* antara perbedaan *latitude* dengan lokasi pengguna dibagi dengan perbedaan *longitude*. Gambar 2.2



adalah Rumus perhitungan sudut *azimuth*. Gambar 2.3 adalah ilustrasi sudut *azimuth*.

$$\alpha_{AB} = \tan^{-1}((X_b - X_a) / (Y_b - Y_a)) * (10,1)$$

Gambar 2.2 Rumus perhitungan sudut *azimuth*



Gambar 2.3 Ilustrasi sudut *azimuth*

## BAB III

### ANALISIS & PERANCANGAN SISTEM

#### 3.1. Analisa Masalah

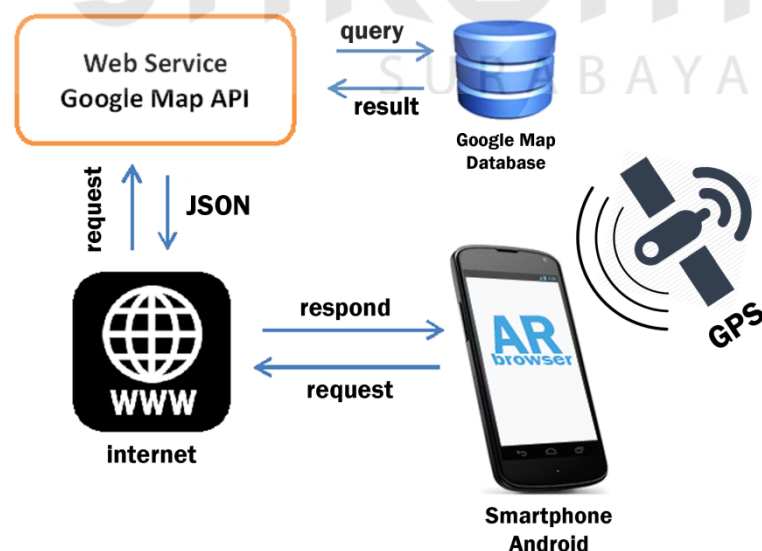
Pada saat melakukan perjalanan baru, seperti mengunjungi tempat wisata ataupun tempat asing untuk maksud tertentu, di mana wisatawan tidak tahu lokasi layanan umum seperti *ATM*, *SPBU*, restoran ataupun hotel terdekat, sedangkan ke-empat tempat tersebut adalah tempat yang penting. Butuh informasi yang tepat dan cepat untuk melayani kebutuhan wisatawan tentang lokasi layanan umum tersebut. Kemajuan dalam hal teknologi telah mengubah cara bepergian dan melakukan kegiatan perjalanan, dengan adanya peta digital wisatawan dapat mengetahui lokasi layanan umum. Jika wisatawan tersebut ingin menggunakan peta digital haruslah menguasai ilmu tentang membaca peta, namun tidak semua orang dapat membaca peta, atau biasa dikenal dengan buta peta. Akhirnya kebutuhan dalam memperoleh informasi lokasi tujuannya tidak berguna.

Berdasarkan uraian di atas maka diusulkan Aplikasi *GPS* dan *GIS* untuk mencari lokasi dan jarak layanan umum dengan Peta dan *Augmented Reality Camera-View* pada Perangkat Bergerak Berbasis *Android*. Aplikasi tersebut dapat memudahkan pencarian lokasi layanan umum yang dibutuhkan hanya dengan memilih pada menu lokasi layanan umum misalnya *SPBU*, restoran, dan hotel, tanpa harus mengetik nama lokasi yang dibutuhkan, sehingga dapat mempercepat pencarian lokasi layanan umum. Aplikasi tersebut melakukan *tracking* lokasi kita saat ini (menentukan lokasi di mana kita berada), lalu memanggil ke *web service* untuk pencarian lokasi layanan umum dengan radius *default* seratus meter dari lokasi kita berada, kemudian dengan menggunakan *augmented reality* yaitu

dengan menggunakan kamera pada *device Android* yang digabungkan dengan obyek maya (penanda lokasi layanan umum). Aplikasi pencarian layanan umum tersebut mempunyai sudut pandang berbeda dalam sistem navigasi, yaitu dengan kondisi nyata pada saat ini secara *Real-Time* yang diharapkan akan mudah dipahami dan dimengerti dari pada harus melihat peta yang membutuhkan pengetahuan dan pemahaman tentang membaca peta dan arah mata angin.

### 3.1.1. Arsitektur aplikasi

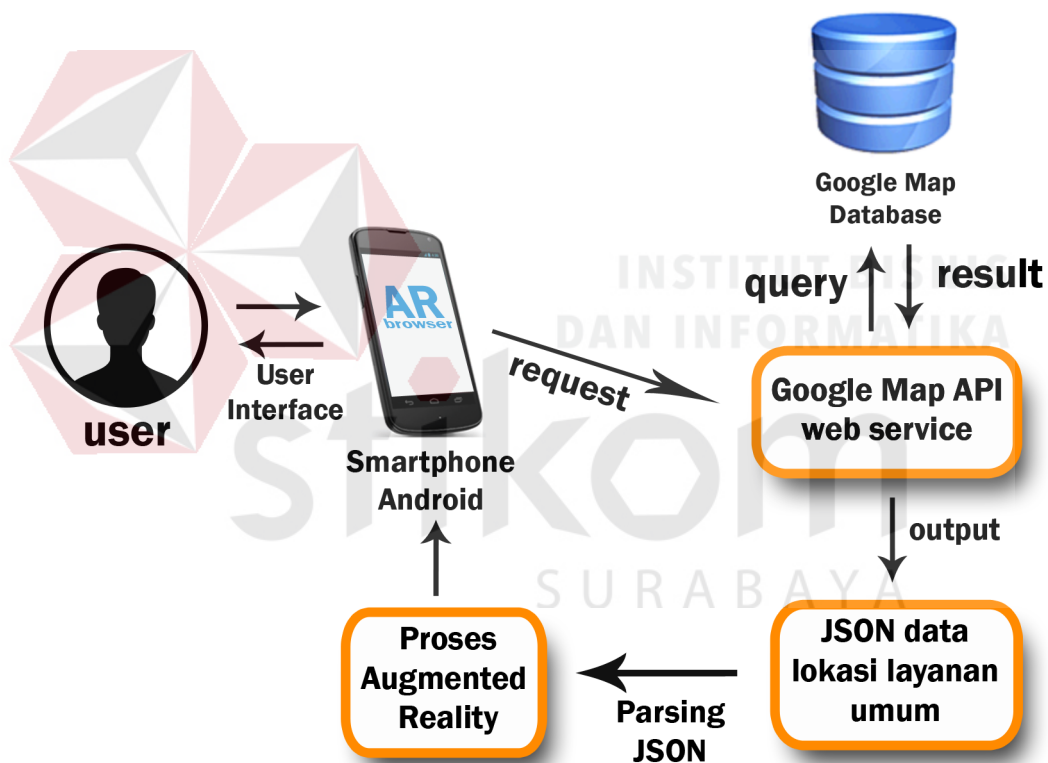
Sistem yang di bangun akan menampilkan informasi yang berkaitan dengan pencarian lokasi layanan umum, di mana informasi tersebut didukung oleh teknologi *augmented reality*. Seorang wisatawan membutuhkan koneksi internet pada smartphone Android ketika memakai aplikasi, untuk keperluan *upload* (mengirim) data lokasi saat ini, dan *download* (mengambil) data hasil pencarian lokasi layanan umum. Gambar 3.1 adalah gambar arsitektur aplikasi penentuan lokasi layanan umum



Gambar 3.1. Arsitektur aplikasi penentuan lokasi layanan umum.

### 3.1.2. Arsitektur aplikasi pada *Smartphone Android*

Dalam pembuatan aplikasi *Smartphone Android* dibutuhkan koneksi yang menghubungkan *smartphone* dengan *Web Service* yang di sediakan oleh *Google*. Lalu data *output* dari *Google Map* diolah kembali menggunakan proses *Augmented Reality* yaitu perhitungan sudut *azimuth* dan *inklinasi* menjadi keterangan dalam tampilan visual. Gambar 3.2 menggambarkan arsitektur aplikasi pada *smartphone Android*.



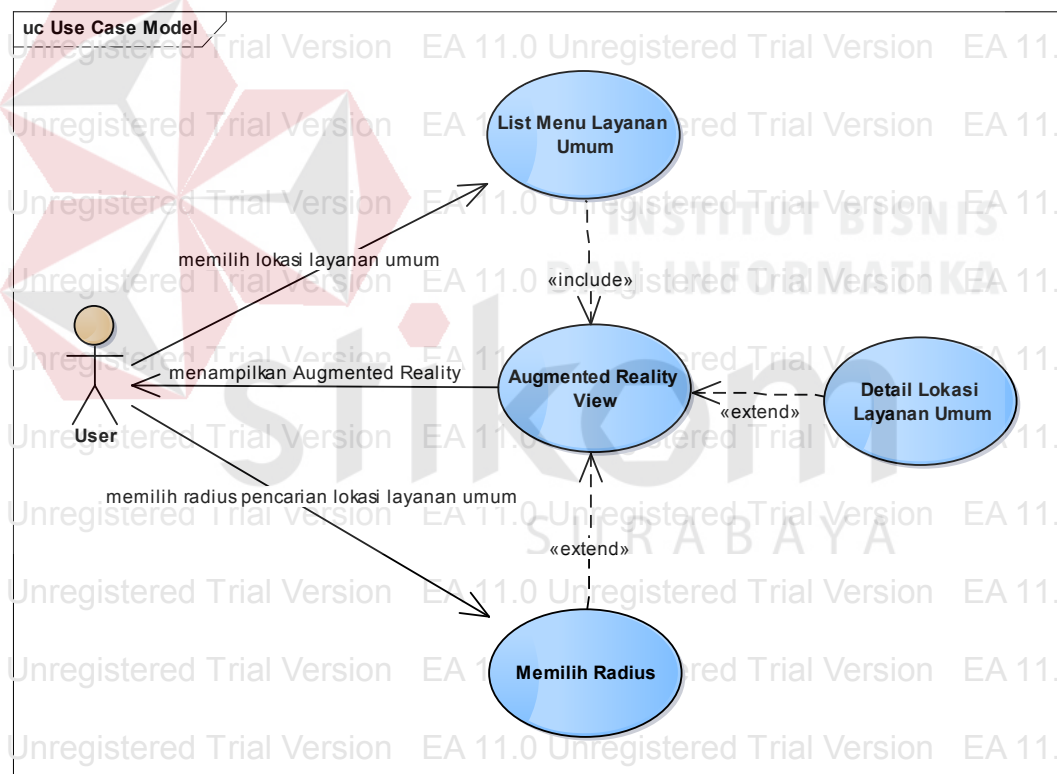
Gambar 3.2. Arsitektur aplikasi pada *smartphone Android*.

Untuk dapat menggunakan aplikasi penentuan lokasi layanan umum pada *smartphone Android*, langkah pertama yang harus dilakukan oleh *user* adalah melakukan *download* aplikasi penentuan lokasi layanan umum. setelah aplikasi diinstal pada *smartphone Android*, *user* membutuhkan koneksi internet agar

aplikasi dapat mengakses *Web Service* untuk mengambil data-data lokasi layanan umum. setelah data lokasi layanan umum di olah dengan *parsing JSON* dan proses *Augmented Reality* barulah user dapat menggunakan aplikasi penentuan lokasi layanan umum tersebut.

### 3.2 Use Case Diagram

*Use case diagram* digunakan untuk menentukan apa yang dapat dilakukan oleh sistem atau untuk menentukan kebutuhan fungsional utama dari sistem. Berikut akan dijelaskan *use case diagram*.



Gambar 3.3. *Use Case Diagram* Aplikasi Penentuan Lokasi Layanan Umum

*Use Case* ini dimulai dari pertama kali *user* membuka aplikasi terlebih dahulu, kemudian *actor user* dapat memilih list menu lokasi layanan umum yang ingin dicari. Setelah memilih kategori sistem akan mengirim *request* pada *Google*

*Map API*, di mana data *request* akan berisi data lokasi saat ini dan radius pencarian Proses yang digunakan untuk mengolah data dari data lintang dan bujur pada peta, kemudian disesuaikan dengan lokasi saat ini (lintang dan bujur) disesuaikan dengan radius pencarian lokasi layanan umum. Lalu proses yang digunakan untuk mengolah data perhitungan dari *Augmented Activity*, kemudian diolah dengan *camera view* ke dalam bentuk *Augmented View*, yaitu pengabungan objek maya dari proses *Augmented Activity* yang di sesuaikan dengan besarnya sudut pandang kamera *smartphone Android* dan sensor *accelerometer* untuk mengetahui sudut *azimuth user* dari arah utara kompas. Seperti tabel X yang menggambarkan keterangan singkat dari *use case* aplikasi penentuan lokasi layanan umum.

Tabel 3.1 Tabel keterangan *use case* aplikasi penentuan lokasi layanan umum.

Nama Use Case	Deskripsi
<b>List Menu Layanan Umum</b>	Proses yang digunakan untuk menentukan mana lokasi layanan umum yang hendak dicari.
<b>Pilih Radius Pencarian</b>	Proses yang digunakan untuk menentukan radius pencarian lokasi layanan umum.
<b><i>Augmented Reality View</i></b>	Proses yang digunakan untuk mengolah data dari data lintang dan bujur pada peta, kemudian disesuaikan dengan lokasi saat ini (lintang dan bujur) disesuaikan dengan radius pencarian lokasi layanan umum. Kemudian proses selanjutnya adalah pengabungan objek maya dari proses <i>Augmented Activity</i> yang di sesuaikan dengan besarnya sudut pandang kamera <i>smartphone Android</i> dan sensor <i>accelerometer</i> untuk mengetahui sudut <i>azimuth user</i> dari arah utara kompas

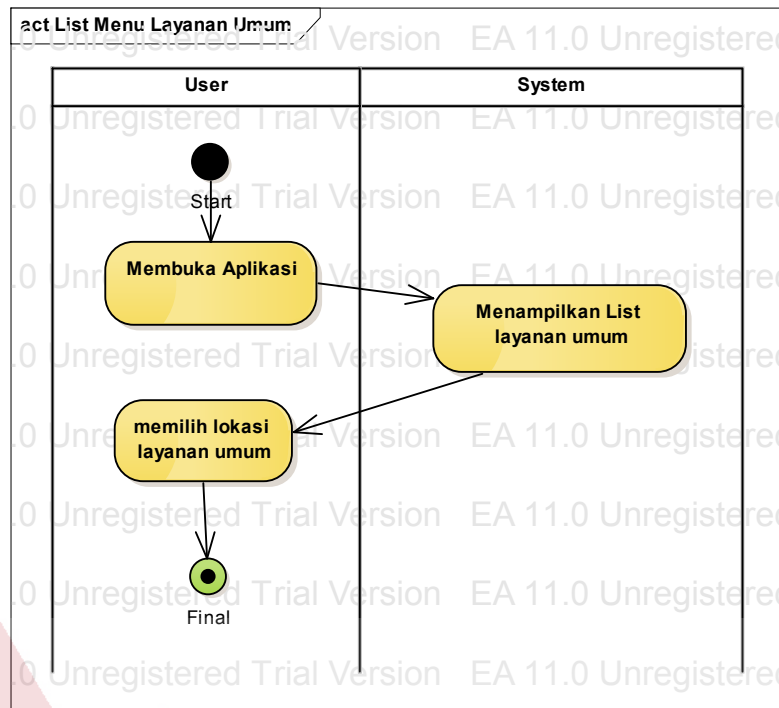
<b><i>Detail Lokasi Layanan Umum</i></b>	Proses memberikan detail lokasi pencarian layanan umum, dari informasi lintang dan bujur sebuah lokasi
--	--

### 3.3 Activity Diagram

Dari *Use Case* yang ada, dibutuhkan *Activity Diagram* untuk menjelaskan proses/aliran yang terjadi pada tiap *Use Case*. *Activity Diagram* adalah salah satu bentuk diagram UML yang paling mudah di mengerti dikarenakan simbol menyerupai simbol *flowchart*, yang sangat berguna untuk menerangkan langkah-langkah ke pihak lain.

#### 3.3.1 Activity Diagram List Menu Lokasi Layanan Umum

Proses ini dimulai dari *user* membuka aplikasi dan masuk ke menu utama setelah aplikasi menampilkan *splashscreen*. *User* akan diberi beberapa *option* untuk memilih lokasi layanan umum yang mana, setelah memilih lokasi layanan umum yang ingin dicari maka sistem akan mencatat pencarian *user* untuk digunakan dalam proses selanjutnya. Untuk lebih jelasnya dapat di lihat pada gambar 3.4, adalah *activity diagram list* menu lokasi layanan umum.

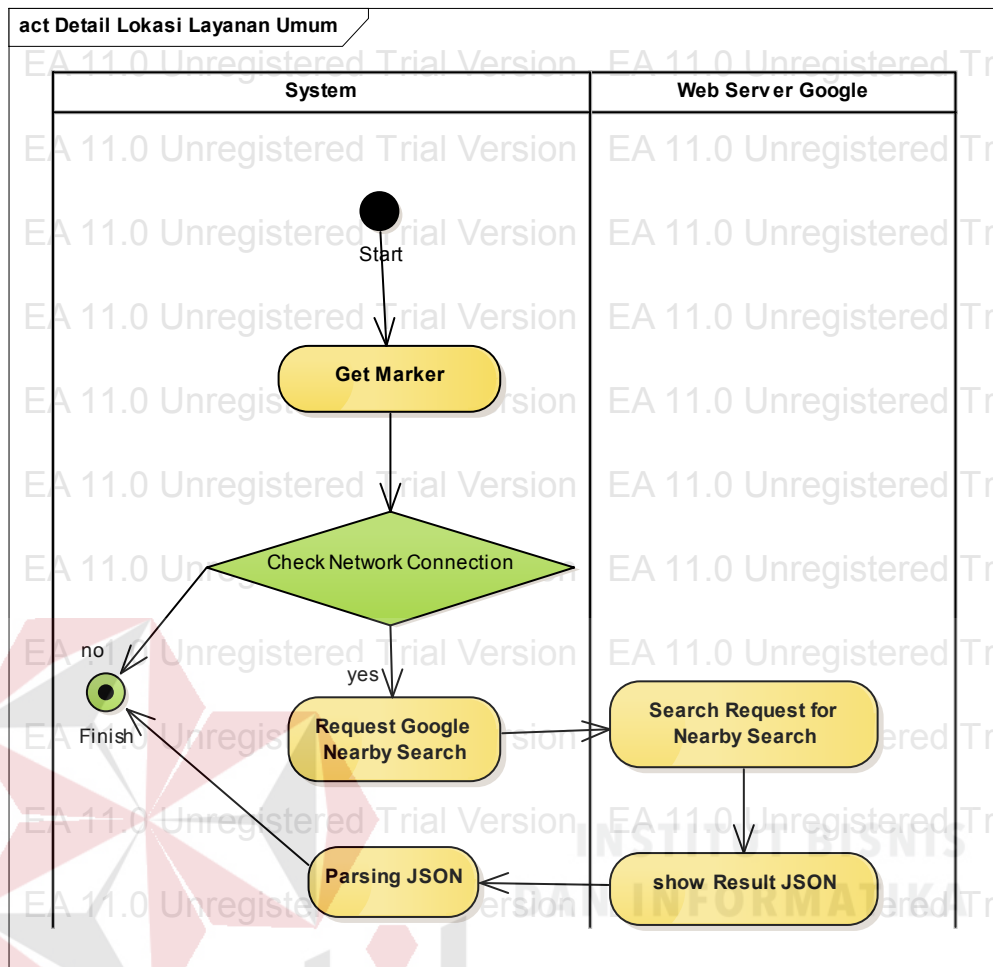


Gambar 3.4. *Activity diagram list menu lokasi layanan umum.*

### 3.3.2 Activity Diagram Detail Lokasi Layanan Umum

Proses dimulai dari sistem mengambil *marker* lalu mengecek apakah aplikasi terhubung dengan internet, karena aplikasi ini membutuhkan koneksi internet, jika terhubung dengan internet maka sistem melakukan *request* pencarian lokasi terdekat dan dikirim ke *Web Server Google* yaitu *Google Maps API*, lalu *Web Server Google* melakukan pencarian lokasi terdekat berdasarkan lokasi dari sistem. *Web server Google* akan menampilkan hasil pencarian berupa *JSON (Java Script Object Nation)* yaitu detail tentang lokasi terdekat dari mulai nama, *geolocation*, type tempat, dan lain-lain, sehingga setelah *web server Google* menampilkan hasil pencarian masih perlu di saring informasinya di ambil yang perlu saja dengan proses *Parsing JSON* yaitu memilah-milah dan menampungnya dalam Data Array untuk proses selanjutnya. Gambar 3.5 adalah *Activity Diagram Detail Lokasi Layanan Umum*



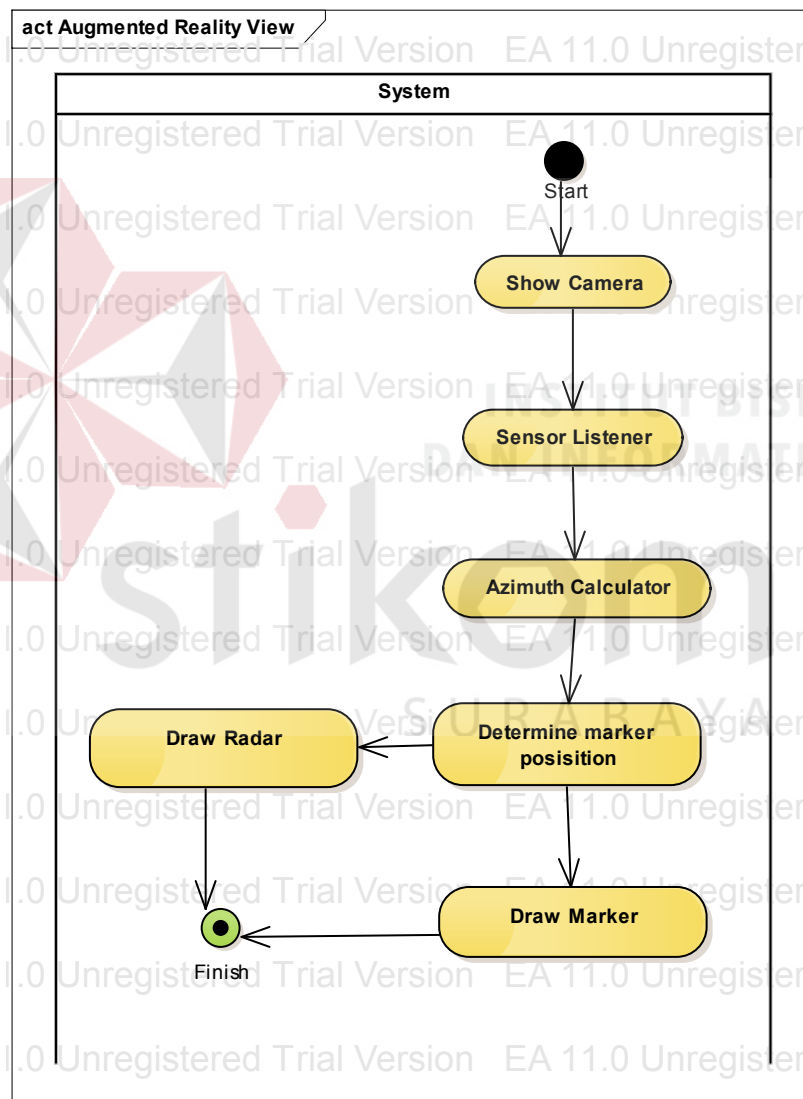


Gambar 3.5. Activity Diagram *Detail* Lokasi Layanan Umum

### 3.3.3 Activity Diagram *Augmented Reality View*

Proses dimulai di dalam sistem setelah *user* memilih lokasi layanan umum, yaitu menampilkan kamera, setelah itu mendengarkan semua sensor-sensor yang di butuhkan untuk mengetahui lokasi saat ini, dan posisi *smartphone* saat ini juga, proses ini membutuhkan sensor *GPS* untuk mengetahui lokasi saat ini, dan *accelerometer* untuk mengetahui posisi *smartphone* sedang dalam kondisi *portrait*, atau *landscape*, *flatback* dan lain-lain, perlu juga kompas untuk menentukan lokasi saat ini berada di berapa derajat dari arah utara. Lalu proses selanjutnya adalah menghitung semua lokasi layanan umum dan lokasi *user* saat

ini dengan cara menghitung derajat *azimuth* dan masuk ke proses selanjutnya yaitu menentukan posisi *marker* pada layar *smartphone* dengan perhitungan sudut *azimuth* dari proses sebelumnya tersebut. Lalu setelah disesuaikan dengan layar dan dilakukan perhitungan sudut *azimuth* maka *marker* dan radar untuk *augmented reality view* akan di gambar. Untuk lebih jelasnya dapat dilihat pada gambar 3.6 yaitu *Activity Diagram Augmented Reality View*.

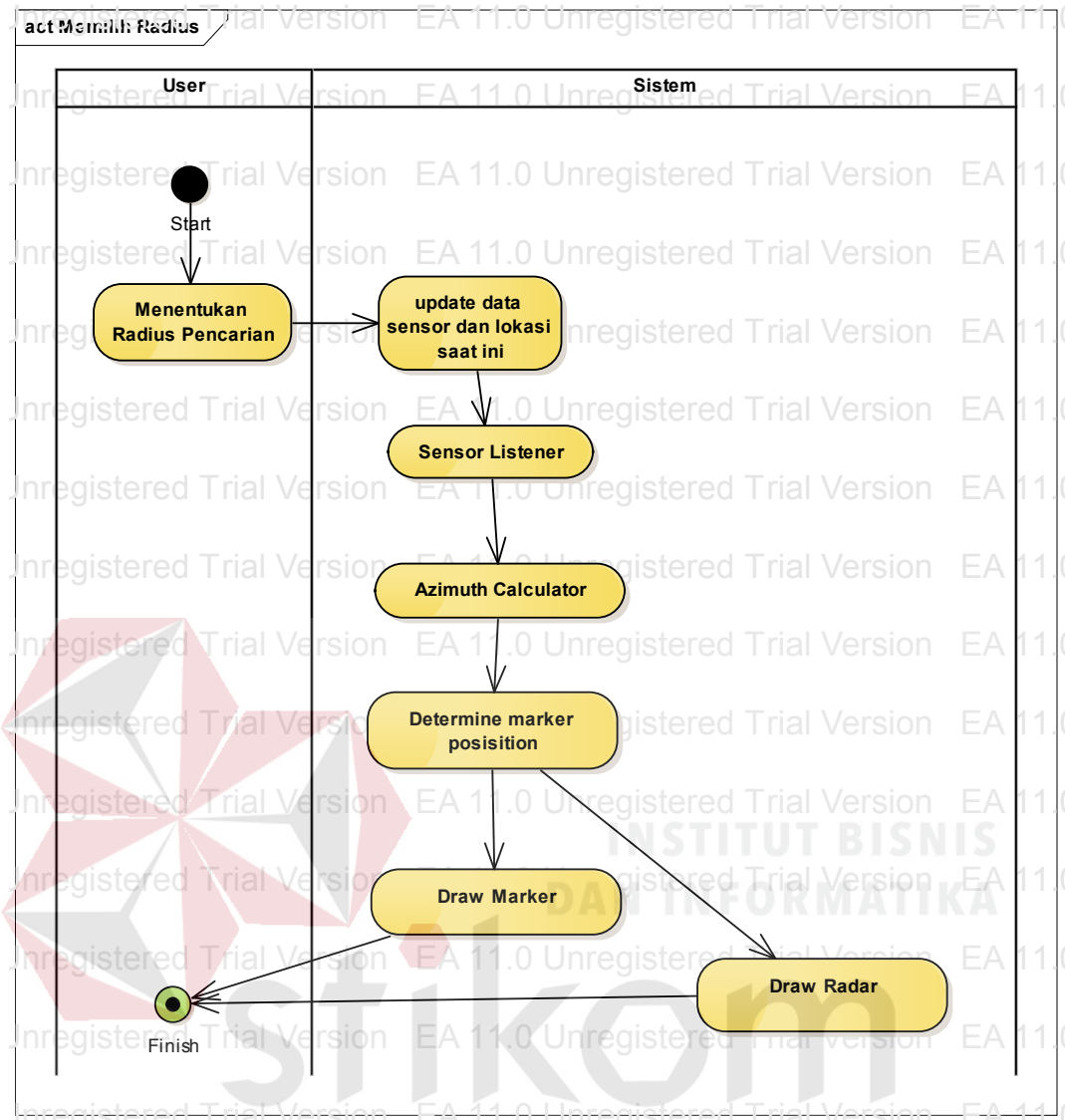


Gambar 3.6. *Activity Diagram Augmented Reality View*.

### 3.3.4 Activity Diagram Memilih Radius

Proses ini adalah proses *extend*, jadi jika *user* tidak menentukan radius dalam proses menentukan radius pencarian maka sistem sudah memberikan nilai default untuk radius pencarian lokasi layanan umum tersebut. Proses ini dimulai dari *user* menentukan radius pencarian dari 0 – 100km lalu proses selanjutnya adalah sistem melakukan *set* terhadap radius untuk keperluan proses selanjutnya yaitu pencarian lokasi layanan umum dengan radius yang di tentukan oleh *user* tersebut. Untuk lebih jelasnya dapat dilihat pada gambar 3.7 yaitu *Activity Diagram Memilih Radius*.





Gambar 3.7. Activity Diagram Memilih Radius.

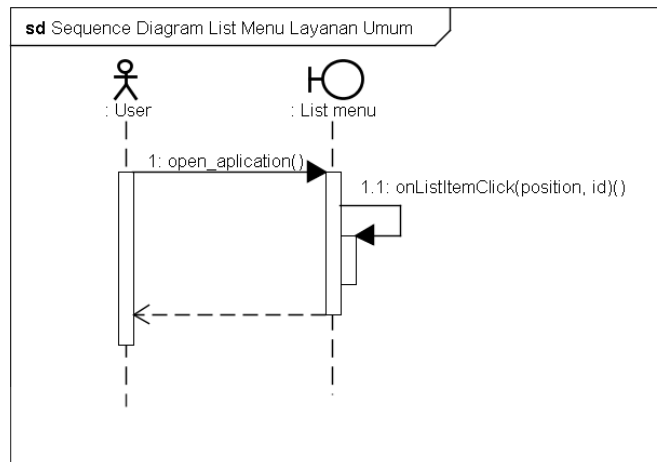
### 3.4 Sequence Diagram

*Sequence Diagram* digunakan untuk menggambarkan interaksi antar objek berdasarkan urutan waktu yang digambarkan dari atas kebawah.

#### 3.4.1 Sequence Diagram List Menu Lokasi Layanan Umum

Proses ini dimulai dari *user* membuka aplikasi dan masuk ke menu utama aplikasi. *User* akan diberi beberapa *option* untuk memilih lokasi layanan umum yang mana, setelah memilih lokasi layanan umum yang ingin dicari maka sistem

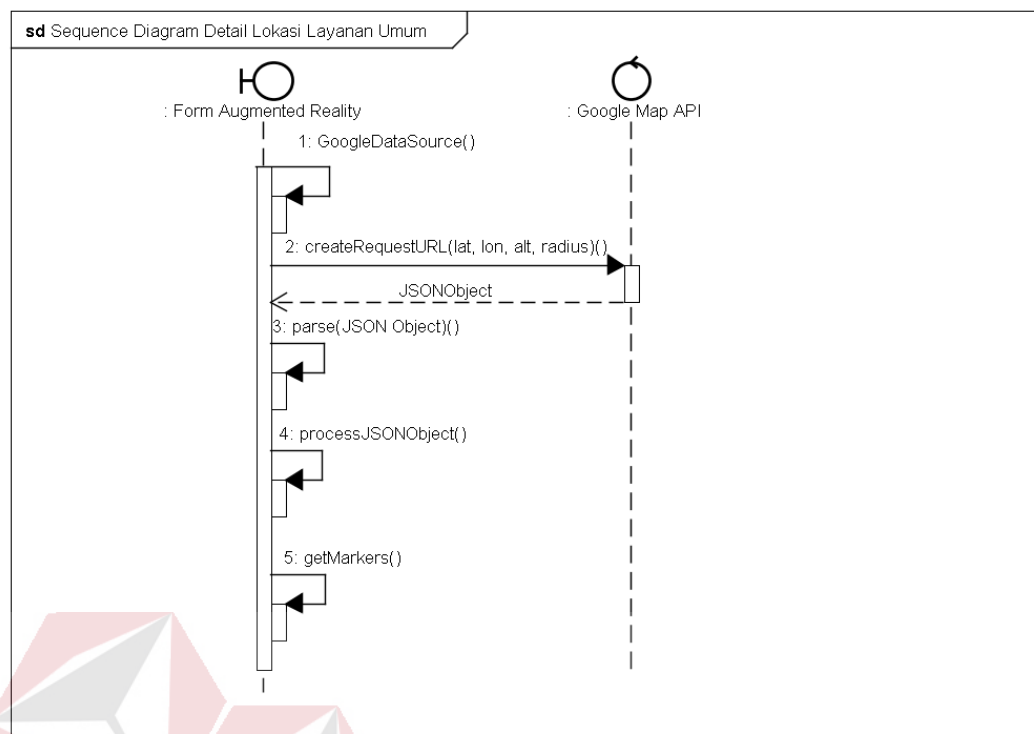
akan mencatat pencarian *user* untuk digunakan dalam proses selanjutnya. Untuk lebih jelasnya dapat di lihat pada gambar 3.8, adalah *secuence diagram list menu* lokasi layanan umum.



gambar 3.8. *Secuence diagram list menu* lokasi layanan umum.

### 3.4.2 Sequence Diagram Detail Lokasi Layanan Umum

Proses dimulai dari sistem mengambil *marker*, lalu sistem melakukan *request* pencarian lokasi terdekat dan dikirim ke *Web Server Google* yaitu *Google Maps API*, lalu *Web Server Google* melakukan pencarian lokasi terdekat berdasarkan lokasi dari sistem. *Web server Google* akan menampilkan hasil pencarian berupa *JSON (Java Script Object Nation)* yaitu detail tentang lokasi terdekat dari mulai nama, *geolocation*, type tempat, dan lain-lain, sehingga setelah *web server Google* menampilkan hasil pencarian masih perlu di saring informasinya di ambil yang perlu saja dengan proses *Parsing JSON* yaitu memilah-milah dan menampungnya dalam *Data Array* untuk proses selanjutnya. Gambar 3.9 adalah *Sequence Diagram Detail Lokasi Layanan Umum*.

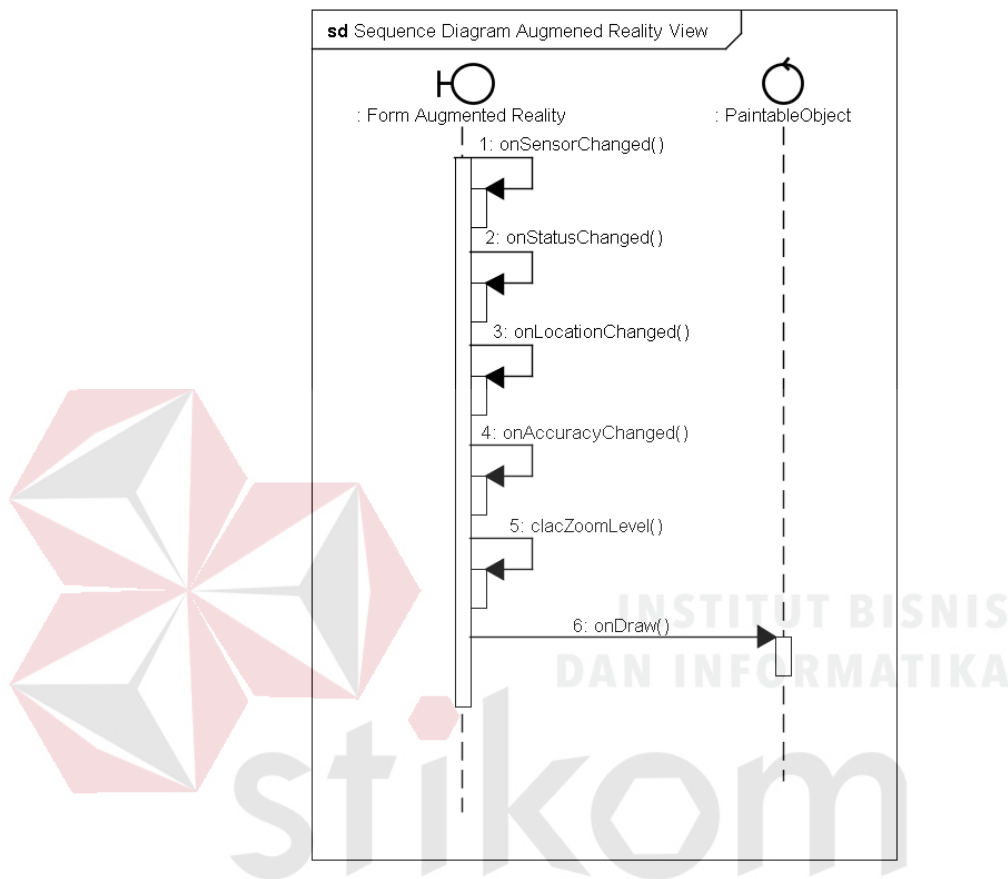


Gambar 3.9. *Sequence Diagram Detail Lokasi Layanan Umum.*

### 3.4.3. *Sequence Diagram Augmented Reality View*

Proses dimulai di dalam sistem setelah *user* memilih lokasi layanan umum, yaitu menampilkan kamera, setelah itu mendengarkan semua sensor-sensor yang di butuhkan untuk mengetahui lokasi saat ini, dan posisi smartphone saat ini juga, proses ini membutuhkan sensor *GPS* untuk mengetahui lokasi saat ini, dan *accelerometer* untuk mengetahui posisi *smartphone* sedang dalam kondisi *portrait*, atau *landscape*, *flatback* dan lain-lain, perlu juga kompas untuk menentukan lokasi saat ini berada di berapa derajat dari arah utara. Lalu proses selanjutnya adalah menghitung semua lokasi layanan umum dan lokasi *user* saat ini dengan cara menghitung derajat *azimuth* dan masuk ke proses selanjutnya yaitu menentukan posisi *marker* pada layar *smartphone* dengan perhitungan sudut *azimuth* dari proses sebelumnya tersebut. Lalu setelah disesuaikan dengan layar,

maka *marker* dan radar untuk *augmented reality view* akan di gambar. Untuk lebih jelasnya dapat dilihat pada gambar 3.10 yaitu *Sequence Diagram Augmented Reality View*.

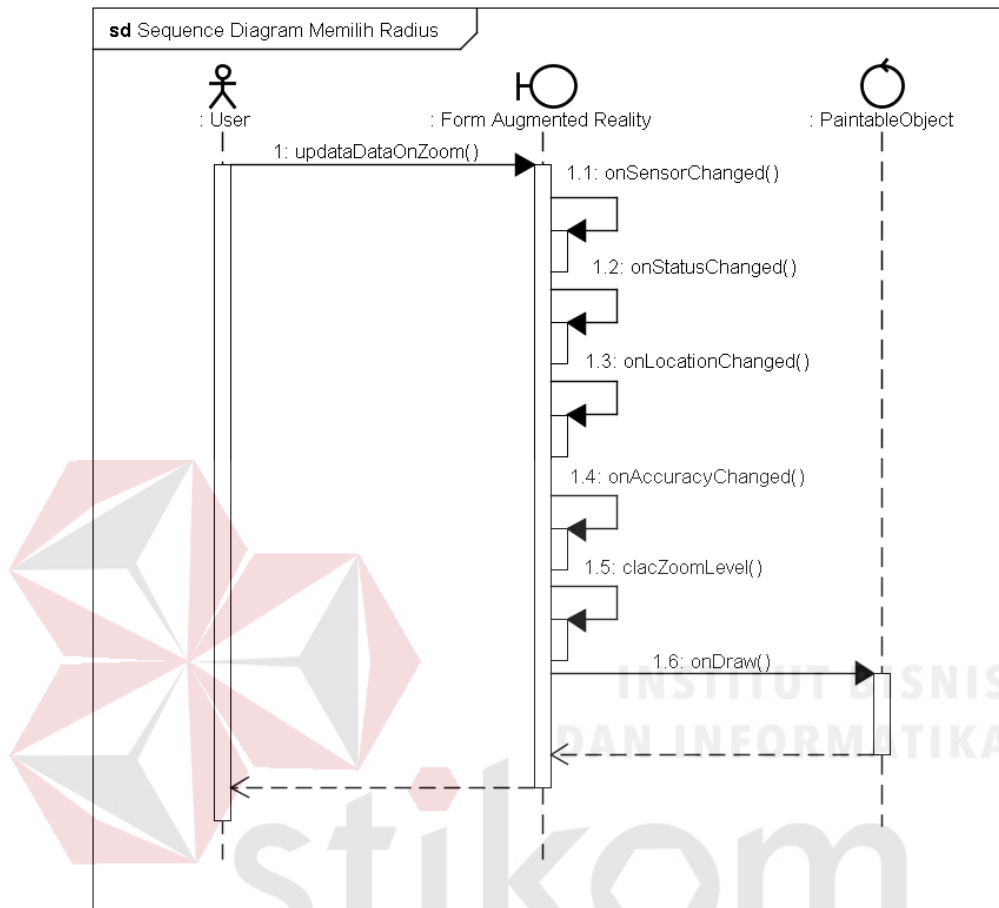


Gambar 3.10. *Sequence Diagram Augmented Reality View*.

### 3.3.4 Sequence Diagram Memilih Radius

Proses ini adalah proses *extend*, jadi jika *user* tidak menentukan radius dalam proses menentukan radius pencarian maka sistem sudah memberikan nilai default untuk radius pencarian lokasi layanan umum tersebut. Proses ini dimulai dari *user* menentukan radius pencarian dari 0 – 100km lalu proses selanjutnya adalah sistem melakukan *set* terhadap radius untuk keperluan proses selanjutnya yaitu pencarian lokasi layanan umum dengan radius yang di tentukan oleh *user*

tersebut. Untuk lebih jelasnya dapat dilihat pada gambar 3.11 yaitu *Sequence Diagram Memilih Radius*.

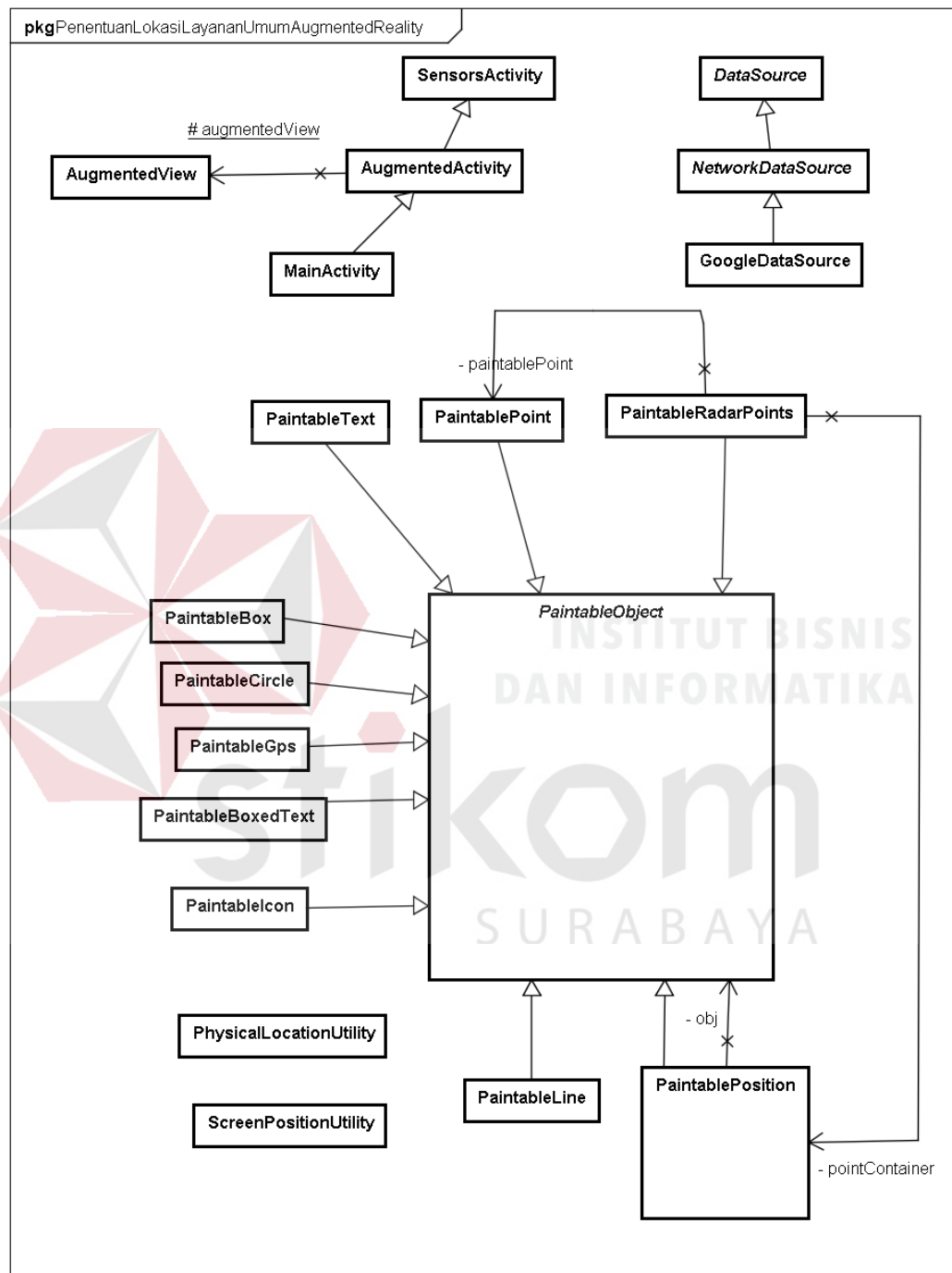


gambar 3.11 yaitu *Sequence Diagram Memilih Radius*.

### 3.5 Class Diagram

*Class Diagram* digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam sistem dan relasi antar kelas tersebut (menunjukkan interaksi antar kelas di dalam aplikasi). *Class diagram* dalam aplikasi penentuan lokasi layanan umum dapat dilihat pada gambar 3.12.

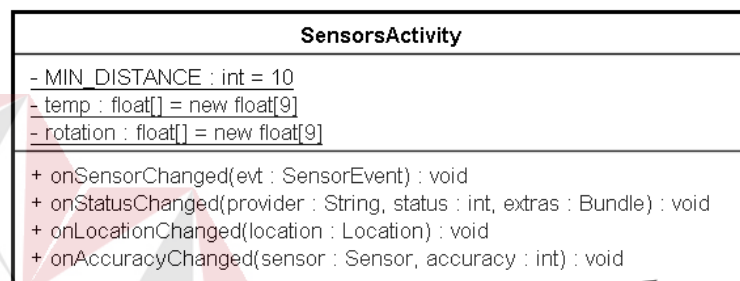




Gambar 3.12. Class Aplikasi penentuan lokasi layanan umum

### 3.5.1 Class *SensorActivity*

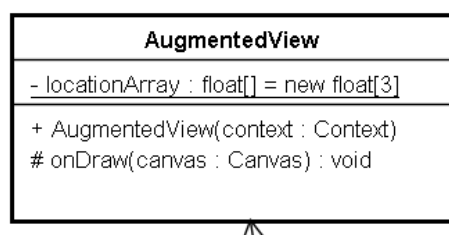
Pada kelas *SensorActivity* kita mulai dengan menetapkan nilai variable-variable dari sensor dengan menggunakan *SensorEventListener*, ketika mendapatkan nilai-nilai yang berubah dari sensor *compass* dan *accelerometer* dan menyimpannya. Setelah menyimpannya, kita mengetahui koordinat saat ini dan menyimpannya di temporary. *Class SensorActivity* dapat dilihat pada gambar 3.13.



Gambar 3.13. *Class SensorActivity*

### 3.5.2 Class *AugmentedView*

Kelas ini di rancang untuk menggambar *radar*, mengontrol radius, dan *marker* yang menunjukkan data melalui tangkapan kamera *smartphone*. Inisialisasi radar pada kelas ini adalah turunan dari kelas radar. *Location Array* adalah digunakan untuk menyimpan lokasi *marker* yang kita kerjakan nanti. *Updated* digunakan ketika kami memperbarui data yang kami dapatkan dari *Google Map*. *Class AugmentedView* dapat dilihat pada gambar 3.14.

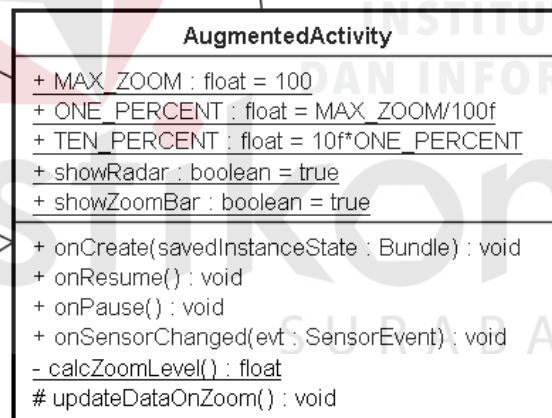


Gambar 3.14. *Class AugmentedView*

### 3.5.3 Class *AugmentedActivity*

Pada kelas ini kita akan mulai mengatur pandang permukaan Kamera, lalu kemudian mengatur tampilan konten dasar *Camera Surface* ini. Setelah ini, kita membuat sebuah *instance* dari *AugmentedView*, mengatur parameter untuk posisi atau tata letak *marker*, dan menambahkannya ke layar. Kami kemudian menambahkan *scrollbar* radius dan text detail *scrollbar* radius ke layar, dan memanggil metode untuk memperbarui data.

Kami menggunakan metode untuk mendengarkan perubahan pada kompas dan *accelerometer* sensor. Kita mulai dengan mendapatkan kemajuan pada *scrollbar* radius sebagai tingkat *zoom*. Gambar 3.15 adalah class *AugmentedActivity*



Gambar 3.15. Class *AugmentedActivity*

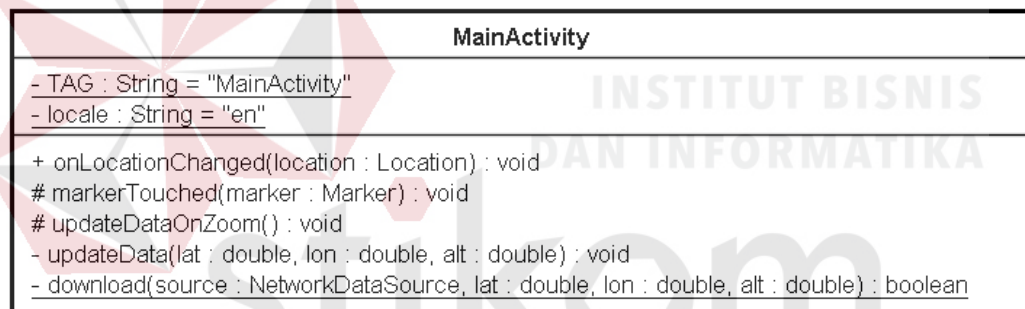
### 3.5.4 Class *MainActivity*

Pada kelas ini kami membuat *NetworkDataSource* untuk *Google Map API*, dan menambahkan mereka ke sumber data Peta. Dalam *OnStart()* metode, kita mendapatkan data lokasi terakhir dan melakukan *update* data kami dengan data tersebut. Sekarang mari kita lihat implementasi kelas ini tentang

*updateDataOnZoom()*. *updateDataOnZoom()*, adalah update data ketika radius pencarian berubah dan kita mendapatkan lokasi dan kemudian memanggil *UpdateData ()* dan memberikan informasi lokasi baru.

Dalam metode *UpdateData* melakukan *download* data yang kita butuhkan untuk menunjukkan data lokasi layanan umum dari *google maps* api.

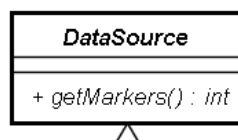
Dalam *metodh download*, kita *me-request* sebuah *URL* untuk mendapatkan data. Setelah ini, kami *parsing JSON* hasil dari *request URL* dan menyimpan data yang kita dapatkan di Daftar *marker*. Data tersebut kemudian ditambahkan ke *ARData* melalui *ARData.addMarkers*. *Class MainActivity* dapat dilihat pada gambar 3.16.



Gambar 3.16. *Class MainActivity*

### 3.5.5 Class *DataSource*

*DataSource* adalah kelas abstrak. Gambar 3.17. adalah gambar *class DataSource*

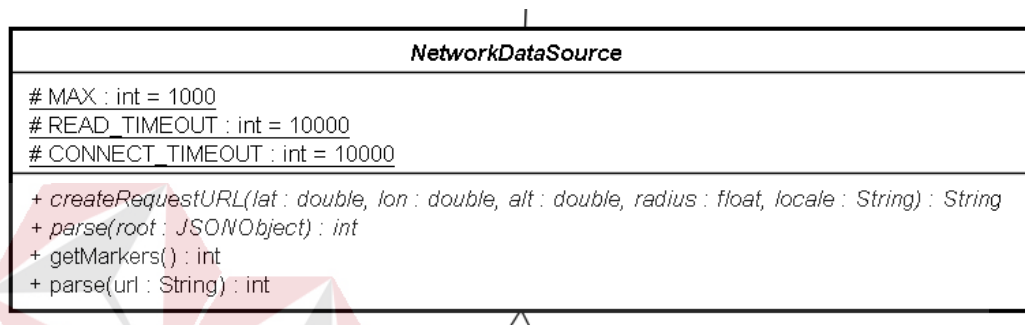


Gambar 3.17. *Class DataSource*

### 3.5.6 Class *NetworkDataSource*

*NetworkDataSource* berisi konfigurasi dasar untuk mendapatkan data dari *Google Map API*.

Dalam *parse ()*, kita mendapatkan objek dari sumber data kami dan kemudian memanggil *parse ()* metode *class DataSource*. Gambar 3.18. adalah gambar *class NetworkDataSource*



Gambar 3.18. Class *NetworkDataSource*

### 3.5.7 Class *GoogleDataSource*

*String URL* menyimpan dasar *URL* pencarian lokasi layanan umum. Kami akan membangun parameter dinamis dalam *createRequestURL()*.

Dalam *createRequestURL ()*, kita merumuskan lengkap *URL request Google API Nearby search* dengan *output JSON*. Pencarian *google API* memungkinkan kita untuk mencari lokasi layanan umum dengan mudah.

Metode *parsing* mengambil *URL* dalam bentuk parameter *string*. *URL* kemudian dimasukkan, dan *JSONObject* diciptakan dari *String* yang dihasilkan, seperti dapat dilihat pada gambar 3.19. adalah gambar *class GoogleDataSource*

GoogleDataSource
+ dataTanto : String = "" - BASE_URL : String = "https://m...
+ GoogleDataSource(res : Resources, param : int) + createRequestURL(lat : double, lon : double, alt : double, radius : float, locale : String) : String + parse(root : JSONObject) : int - processJSONObject(jo : JSONObject) : Marker

Gambar 3.19. *Class GoogleDataSource*

### 3.5.8 Class *PhysicalLocationUtility*

Kelas ini digunakan untuk mewakili lokasi pengguna di dunia nyata dalam tiga dimensi. Gambar 3.20 adalah gambar *class PhysicalLocationUtility*

PhysicalLocationUtility
- latitude : double = 0.0 - longitude : double = 0.0 - altitude : double = 0.0 - x : float[] = new float[1] - y : double = 0.0d - z : float[] = new float[1]
+ PhysicalLocationUtility() + set(latitude : double, longitude : double, altitude : double) : void + getLatitude() : double + getLongitude() : double + getAltitude() : double

Gambar 3.20. *Class PhysicalLocationUtility*

### 3.5.9 Class *ScreenPositionUtility*

*ScreenPositionUtility* digunakan ketika menampilkan baris untuk radar. Gambar 3.21 adalah gambar *class ScreenPositionUtility*

ScreenPositionUtility
- x : float = 0f - y : float = 0f
+ ScreenPositionUtility() + set(x : float, y : float) : void + getX() : float + setX(x : float) : void + getY() : float + setY(y : float) : void

Gambar 3.21. Class *ScreenPositionUtility*

### 3.5.10 Class *PaintableObject*

*PaintableObject* adalah kelas dasar untuk *user interface*. Gambar 3.22.

adalah gambar class *PaintableObject*

<i>PaintableObject</i>
<pre> + PaintableObject() + getWidth() : float + getHeight() : float + paint(canvas : Canvas) : void + setFontSize(size : float) : void + paintLine(canvas : Canvas, x1 : float, y1 : float, x2 : float, y2 : float) : void + paintRect(canvas : Canvas, x : float, y : float, width : float, height : float) : void + paintRoundedRect(canvas : Canvas, x : float, y : float, width : float, height : float) : void + paintBitmap(canvas : Canvas, bitmap : Bitmap, src : Rect, dst : Rect) : void + paintBitmap(canvas : Canvas, bitmap : Bitmap, left : float, top : float) : void + paintCircle(canvas : Canvas, x : float, y : float, radius : float) : void + paintText(canvas : Canvas, x : float, y : float, text : String) : void + paintObj(canvas : Canvas, obj : PaintableObject, x : float, y : float, rotation : float, scale : float) : void + paintPath(canvas : Canvas, path : Path, x : float, y : float, width : float, height : float, rotation : float, scale : float) : void </pre>

Gambar 3.22. Class *PaintableObject*

### 3.5.11 Class *PaintableBox*

Kelas *PaintableBox* memungkinkan kita untuk menggambar kotak garis.

Pada kelas ini memungkinkan Anda untuk mengatur hanya lebar dan tinggi kotak, sedangkan yang kedua memungkinkan Anda untuk mengatur warna nya juga.

Gambar 3.23. adalah gambar Class *PaintableBox*

PaintableBox
- width : float = 0 - height : float = 0 - borderColor : int = Color.rgb(255,255,255) - backgroundColor : int = Color.argb(128,0,0,0)
+ PaintableBox(width : float, height : float) + PaintableBox(width : float, height : float, borderColor : int, bgColor : int) + set(width : float, height : float) : void + set(width : float, height : float, borderColor : int, bgColor : int) : void + paint(canvas : Canvas) : void + getWidth() : float + getHeight() : float

Gambar 3.23. Class *PaintableBox*

### 3.5.12 Class *PaintableBoxedText*

*PaintableBoxedText* menarik teks pada kanvas dengan kotak di sekitarnya. Pertama Perbedaan utama dari *PaintableBox* adalah bahwa dari baru Metode untuk menyesuaikan *text* agar tidak keluar dari kotaknya. Gambar 3.24 adalah Class *PaintableBoxedText*.

PaintableBoxedText
- width : float = 0 - height : float = 0 - areaWidth : float = 0 - areaHeight : float = 0 - bt : String = null - fontSize : float = 12
+ PaintableBoxedText(btInit : String, fontSizeInit : float, maxWidth : float) + PaintableBoxedText(btInit : String, fontSizeInit : float) + set(btInit : String, fontSizeInit : float) : void + set(btInit : String, fontSizeInit : float, maxWidth : float) : void - prepTxt(btInit : String, fontSizeInit : float, maxWidth : float) : void + paint(canvas : Canvas) : void + getWidth() : float + getHeight() : float

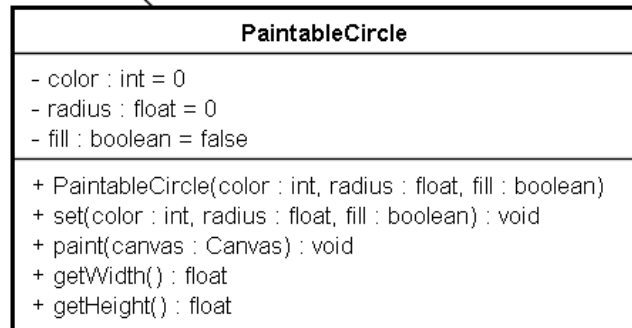
Gambar 3.24. Class *PaintableBoxedText*

### 3.5.13 Class *PaintableCircle*

Pada kelas ini memungkinkan kita untuk mengatur radius dan warna lingkaran. metode menarik lingkaran dengan sifat tertentu pada yang diberikan



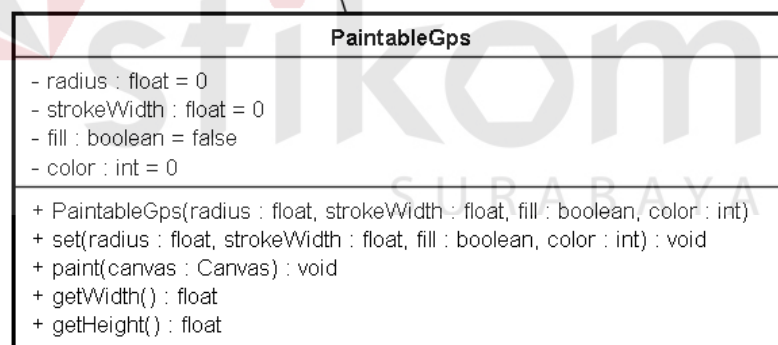
kanvas. Metode *The getWidth ()* dan *getHeight ()* mengembalikan diameter karena ini adalah sebuah lingkaran. Gambar 3.25. adalah gambar *class PaintableCircle*



Gambar 3.25. *Class PaintableCircle*

#### 3.5.14 *Class PaintableGps*

*PaintableGps* sangat mirip *PaintableCircle*, kecuali class ini juga memungkinkan kita untuk mengatur lebar *stroke* lingkaran. Gambar 3.26. *Class PaintableGPS*

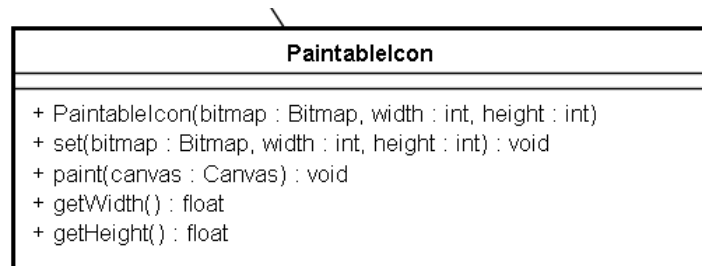


Gambar 3.26. *Class PaintableGPS*

#### 3.5.15 *Class PaintableIcon*

Kami menggunakan *PaintableIcon* untuk menggambar *icon* untuk layanan umum. Konstruktor mengambil *bitmap* yang bisa ditarik, bersama dengan lebar

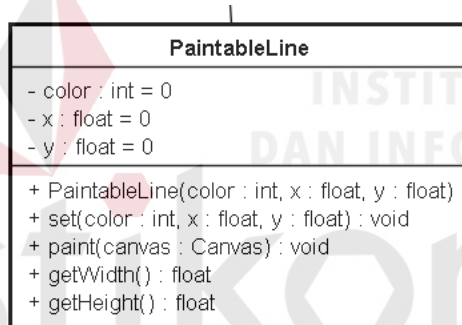
dan tinggi menurut yang itu harus ditarik dan kemudian memanggil *set () metodh* untuk mengatur mereka. Gambar 3.27. *Class PaintableIcon*



Gambar 3.27. *Class PaintableIcon*

### 3.5.16 Class *PaintableLine*

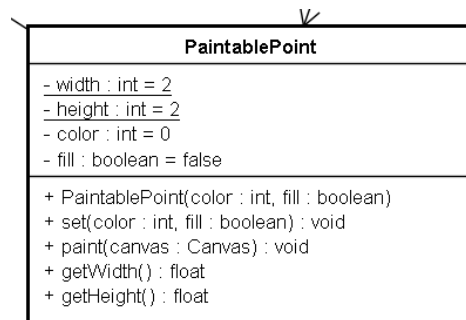
*PaintableLine* memungkinkan kita untuk melukis garis dalam warna tertentu ke sebuah disediakan Kanvas. Gambar 3.28. adalah *Class PaintableLine*



Gambar 3.28. *Class PaintableLine*

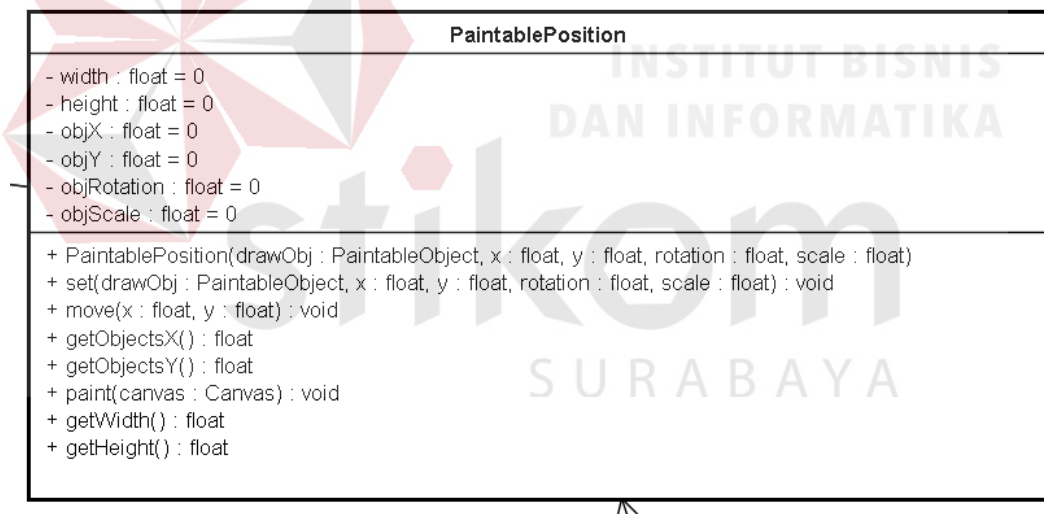
### 3.5.17 Class *PaintablePoint*

*PaintablePoint* digunakan untuk menggambar satu titik di atas kanvas tunggal. Hal ini digunakan ketika membuat radar kami. Gambar 3.29. adalah *Class PaintablePoint*

Gambar 3.29. *Class PaintablePoint*

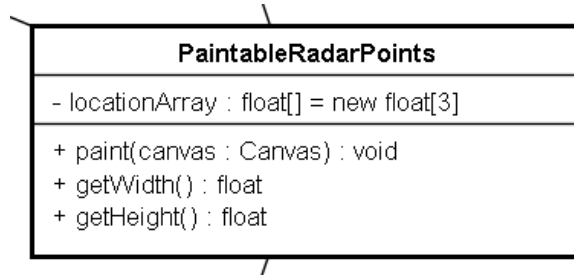
### 3.5.18 Class *PaintablePosition*

*PaintablePosition* meluas *PaintableObject* dan menambahkan kemampuan untuk memutar dan skala hal yang digambar. Gambar 3.30. adalah *Class PaintablePosition*

Gambar 3.30. *Class PaintablePosition*

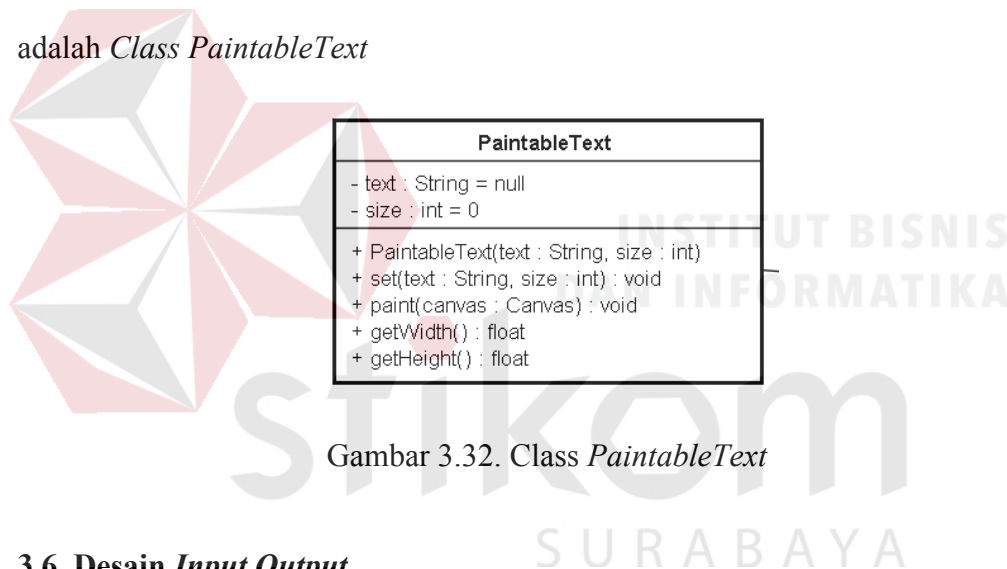
### 3.5.19 Class *PaintableRadarPoints*

*Class PaintableRadarPoints* digunakan untuk menggambar posisi relatif semua *marker* pada radar. Gambar 3.31. adalah *Class PaintableRadarPoint*

Gambar 3.31. Class *PaintableRadarPoint*

### 3.5.20 Class *PaintableText*

*PaintableText* merupakan *extends* dari *PaintableObject* yang memanggil teks. Kami menggunakannya untuk menampilkan teks pada radar. Gambar 3.32. adalah Class *PaintableText*

Gambar 3.32. Class *PaintableText*

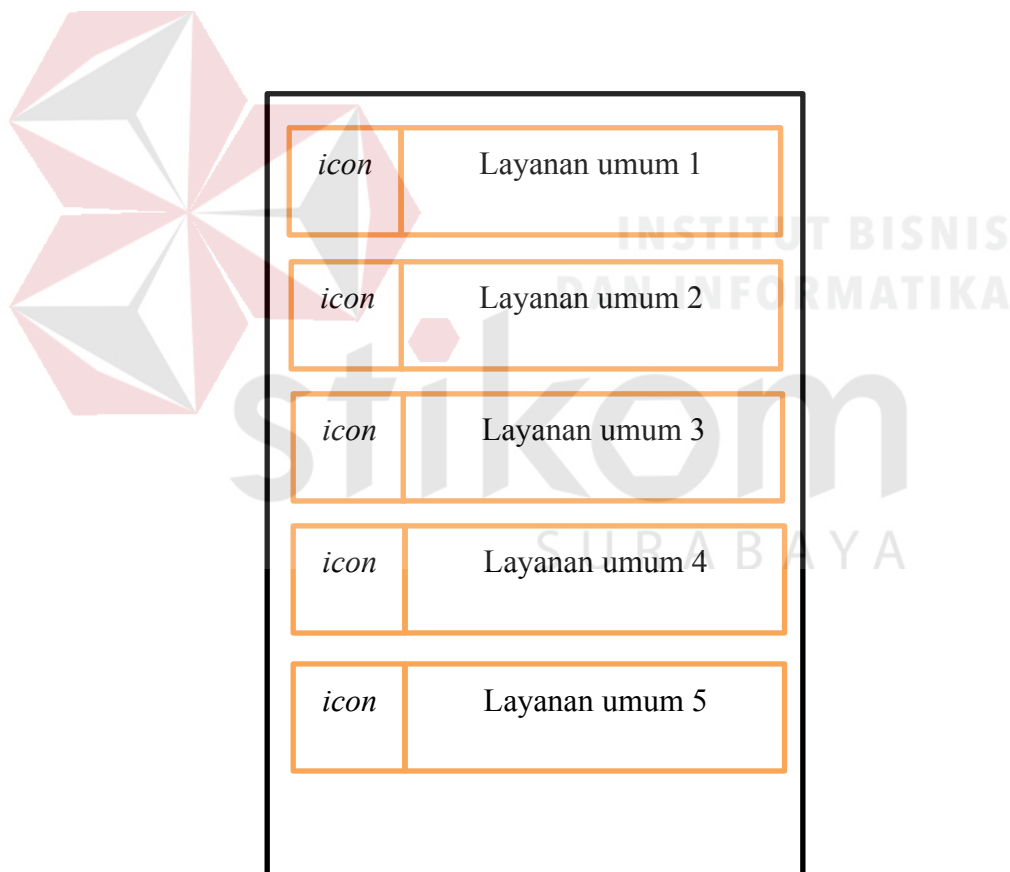
## 3.6 Desain *Input Output*

Desain *input output* adalah visualisasi dari tampilan aplikasi. Desain *input output* aplikasi pencarian lokasi layanan umum dengan menggunakan *augmented reality* pada *smartphone android* adalah sebagai berikut :

### 3.6.1 Layout List Menu

Rancangan *layout* ini digunakan pada saat pertama kali aplikasi pencarian lokasi layanan umum dijalankan. *User* dapat memilih lokasi layanan umum yang ingin dicari, contohnya *user* akan mencari lokasi *restaurant* terdekat, *user* dapat memilih salah satu list menu dari layanan umum yaitu *restaurant*.

Tampilan dari *list* menu ber-orientasi *portrait*, dengan tujuan mempercepat *user* dalam memilih *list* menu. *User* selalu memegang *smartphone* mereka dengan posisi *portrait*, dan *user* tidak akan kebingungan mengarahkan *smartphonenya* karena *list* menu juga ber-orientasi *portrait*. Racangan *list* menu terdiri dari gambar *icon* layanan umum, dan di sebelah kanan-nya adalah detail nama layanan umum. Dengan adanya *icon* layanan umum tersebut diharapkan *user* akan dapat mengerti maksud dari *list* menu itu tanpa membaca nama layanan umum tersebut. Untuk lebih jelasnya *list* menu dapat dilihat pada Gambar 3.33.

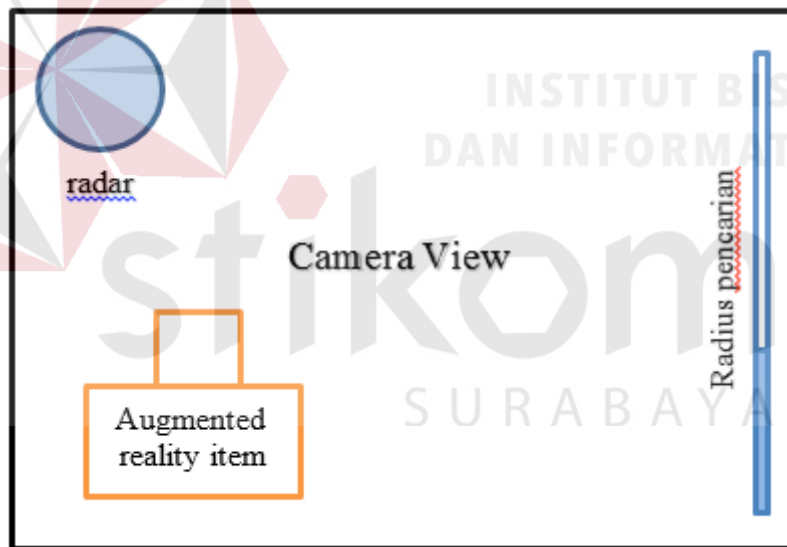


Gambar 3.33 *List* menu.

### 3.6.2 *Layout Augmented Reality View*

Rancangan layout ini digunakan pada saat *user* telah memilih lokasi layanan umum. *Layout Augmented Reality View* ini adalah tampilan dari kamera

*smartphone* untuk menggambarkan lokasi didepan layar *smartphone*, dan terdiri dari *widget* radar, radius dan *augmented reality item*. Tampilan layout berorientasi *landscape*, dikarenakan akan banyak *augmented reality item* yang tertangkap. Bagian kanan layar terdapat *scrollbar* untuk merubah radius, dengan cara menggeser *scrollbar* radius pencarian, jika ke atas maka radius pencarian akan semakin luas, jika kebawah akan semakin kecil atau sempit. Radar di pojok kiri atas adalah untuk navigasi pencarian lokasi diluar jangkauan tampilan kamera. *Augmented reality item* adalah *object* maya yang di gambar pada kamera *smartphone* untuk menunjukkan lokasi layanan umum yang di cari, Gambar 3.34 adalah *Augmented Reality View*.

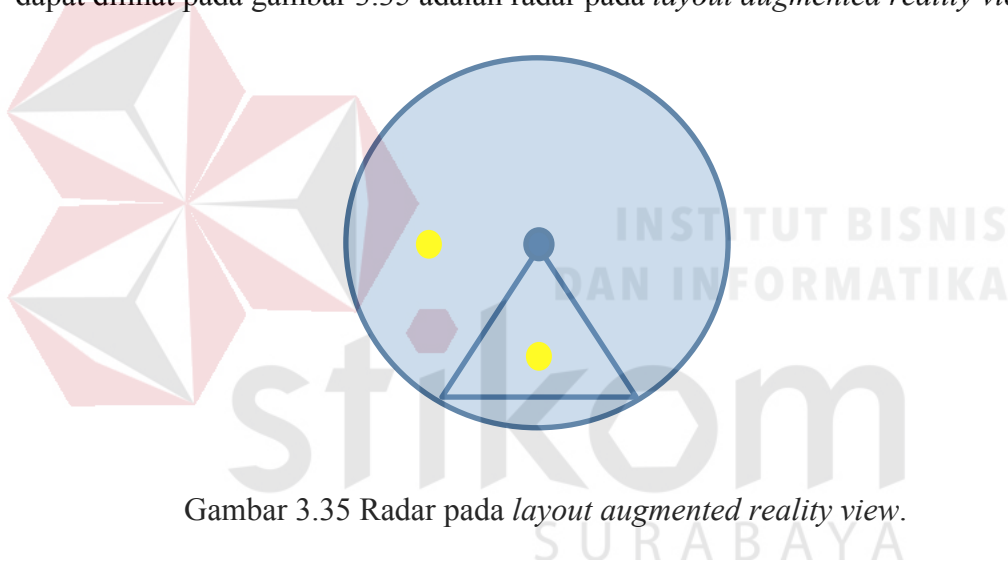


Gambar 3.34 *Augmented Reality View*.

#### A. Gadget Radar Pada Layout Augmented Reality View

*Gadget* Radar pada *layout augmented reality view* digunakan untuk bantuan alat navigasi, jadi *user* dapat melihat *augmented reality item* yang tidak tampak pada jarak pandangnya. Letaknya berada pada atas kiri *layout augmented*

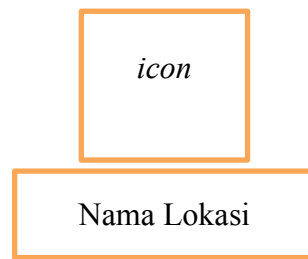
*reality view*, berbentuk lingkaran dengan pusat di tengahnya, lingkaran tersebut merupakan radius dari pencarian lokasi, dan titik tengah adalah lokasi saat ini. Pada bagian tengah lingkaran tersebut ada segitiga, untuk menggambarkan jarak pandang *user*. Terdapat titik-titik kecil di bagian dalam lingkaran adalah lokasi layanan umum yang di pilih oleh *user*, jadi jika *user* mengarahkan *smartphone* ke arah jarak pandang yang tidak ada *item* layanan umum, maka *user* dapat mengarahkan jarak pandangnya dengan melihat arah radar, maka *user* masih dapat terbantu oleh adanya bantuan navigasi berupa radar tersebut. Untuk lebih jelasnya dapat dilihat pada gambar 3.35 adalah radar pada *layout augmented reality view*.



Gambar 3.35 Radar pada *layout augmented reality view*.

## B. Gadget Augmented Reality Item Pada Layout Augmented Reality View

*Gadget augmented reality* item digunakan untuk penanda lokasi layanan umum dengan keterangan. *Icon* dibuat dengan gambar yang mudah di ingat untuk mempermudah penandaan dan detail nama lokasi, agar *user* dapat dengan cepat membedakan tiap *itemnya*. Gambar 3.36 adalah *Augmented Reality Item Pada Layout Augmented Reality View*.



Gambar 3.36 Augmented Reality Item Pada Layout Augmented Reality View.





## **BAB IV**

### **IMPLEMENTASI DAN EVALUASI**

Bab ini berisi tentang implementasi dan evaluasi dalam pengembangan aplikasi yang dibangun.

#### **4.1. Kebutuhan Sistem**

Aplikasi penentuan lokasi layanan umum dengan menggunakan augmented reality ini nantinya akan dapat di akses melalui smartphone android, dengan memanfaatkan sensor-sensor yang dimiliki oleh kebanyakan smartphone tersebut dan membutuhkan koneksi internet. Sehingga kebutuhan minimal yang harus dimiliki oleh smartphone tersebut adalah sebagai berikut :

1. Operating system Android 2.3
2. Ram 512MB
3. Sensor Accelerometer
4. Sensor Compass
5. Sensor GPS
6. Kamera
7. Koneksi internet

#### **4.2. Implementasi Sistem**

##### **4.2.1. Google Map API**

Pada aplikasi ini memerlukan *Google Map API (Application Programming Interface)* sebagai tempat mengambil data lokasi layanan umum. *Google* menyediakan *Framework* yang sangat berguna bagi pengembangnya.

### A. Mengambil data lokasi layanan umum terdekat dengan memanfaatkan *Google Map API* pada aplikasi

*Google Map API (Application Programming Interface)* menyediakan *Framework* yang sangat berguna bagi pengembangnya, untuk memanfaatkan berbagai fitur yang disediakan oleh *Google* seperti misalnya: *AdSense*, *Search Engine*, *Translation* maupun *YouTube*.

Dalam hal ini *Google* telah menyediakan fitur pencarian lokasi terdekat berdasarkan *tags*, *keyword*, maupun *type* dari lokasi yang hendak dicari, lalu *Google* akan memberikan output berupa data *JSON (JavaScript Object Notation)* atau notasi objek *JavaScript*, tidak semua *output* dari *Google Map API* digunakan untuk, maka untuk menyaring data yang diperlukan maka *output* dari *Google Map API* tersebut dilakukan proses penyaringan data yang biasa disebut *parsing JSON*.

Langkah pertama dalam meminta data lokasi terdekat dari *Google Map API* adalah dengan mendaftar ke dalam *account Google*, hal ini bertujuan untuk mendapatkan *API Key*, yang nantinya akan digunakan untuk melakukan *request* pencarian lokasi terdekat. *API Key* adalah *system login* untuk tiap pengembang *Google*, dan digunakan untuk membatasi tiap pengembang melakukan *request* ke *web server Google*. Jadi *Google* masih bisa memantau pengembang mereka. Gambar 4.1 adalah *sourcecode* untuk pencarian lokasi restoran atau cafe yang terdekat.

```
https://maps.googleapis.com/maps/api/place/radarssearch/json?location=48.859294,2.347589&radius=5000&types=food|cafe&sensor=false&keyword=vegetarian&key=AddYourOwnKeyHere
```

Gambar 4.1. Gambar *sourcecode* untuk *request* pencarian lokasi restoran atau *cafe* terdekat.

Dari *request* di atas, Gambar X adalah *Google Map API* mengembalikan nilai *output* berupa *JSON* yang berisi informasi yang lengkap tentang lokasi terdekatnya.

```
{
  "html_attributions" : [
    "Listings by \u003ca href=\"http://www.yellowpages.com.au/\" \u003eYellow Pages\u003c/a\u003e"
  ],
  "results" : [
    {
      "formatted_address" : "529 Kent Street, Sydney NSW, Australia",
      "geometry" : {
        "location" : {
          "lat" : -33.8750460,
          "lng" : 151.2052720
        }
      },
      "icon" : "http://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png",
      "id" : "827f1ac561d72ec25897df088199315f7cbbc8ed",
      "name" : "Tetsuya's",
      "rating" : 4.30,
      "reference" :
        "CnRmAAAAmm3dlSVT3E7rIvwQ0IHBA4sayvxWEc4nZaXSSjRtfKRGoYnfr3d5AvQGk4e0u3oOErXsIJwtd3Wck1Onyw6pCzr8swW4E7dZ6wP4dV6AsXPvodwdVyyqHgyGE_K8DqSp5McW_nFcci_-ljXb5Phv-RIQTzv5BjlGS0ufgTslfC6dqBoU7tw8NKUDHg28bPJlL0vGVWVgbTg",
      "types" : [ "restaurant", "food", "establishment" ]
    },
    {
      "formatted_address" : "Upper Level, Overseas Passenger Terminal/5 Hickson Road, The Rocks NSW, Australia",
      "geometry" : {
        "location" : {
          "lat" : -33.8583790,
          "lng" : 151.2100270
        }
      },
      "icon" : "http://maps.gstatic.com/mapfiles/place_api/icons/cafe-71.png",
      "id" : "f181b872b9bc680c8966df3e5770ae9839115440",
      "name" : "Quay",
      "rating" : 4.10,
      "reference" :
        "CnRiAAAAADmPD0kn3znv_fX78Ma6X5_t7caEGNdSWnpwMIIdDNZkLpVKPnQJXP1ghlySO-ixqs28UtDmJaOlCHn18pxpj7UQjRzR4Kmye6Gijoqoox9bpkaCAJatbJGZEIIUwRbTNIE_L2jGo5BDqiosqU2F5QdBIQbXKrvfQuo6rmu8285j7bDBoUrGrN4r6XQ-PVm260PFt5kwc3EfY",
      "types" : [ "cafe", "bar", "restaurant", "food", "establishment" ]
    },
    {
      "status" : "OK"
    }
  ]
}
```

Gambar 4.2. *Google Map API* mengembalikan nilai *output* berupa *JSON* yang

berisi informasi yang lengkap tentang lokasi terdekatnya.

## B. Input data lokasi layanan umum dengan memanfaatkan *Google Map*

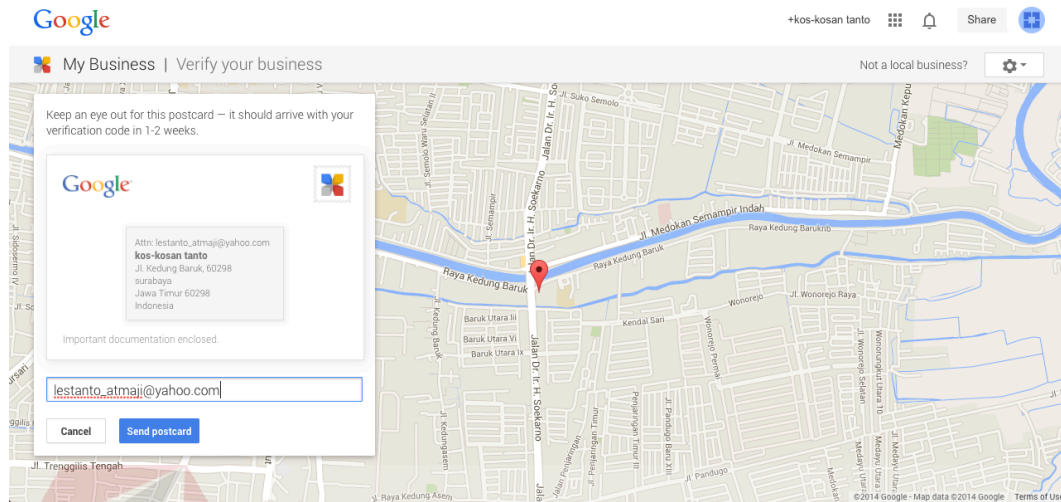
Kami menggunakan database pihak ke-tiga yaitu *Google Maps*, karena *Google Maps* menyediakan fitur web service yang dapat digunakan oleh pengembangnya untuk mencari data lokasi pada *database Google Maps*. Selain kita dapat merequest data lokasi kita juga dapat melakukan *upload* data lokasi bisnis kita agar disimpan dalam *database Google Maps*, agar dapat dicari oleh pelanggan bisnis kita.

Berikut adalah langkah-langkah dalam melakukan *upload* pada database *Google Maps*. Kunjungi Google Place for Business untuk melakukan *signup* untuk Google Places atau *login* ke akun Google yang kita punya. Lalu kita harus melakukan input data detail lokasi bisniss kita. Gambar 4.3 Adalah situs Google for Business halaman input detail lokasi bisnis.

Gambar 4.3. *Website Google for Business* halaman input detail lokasi bisnis.

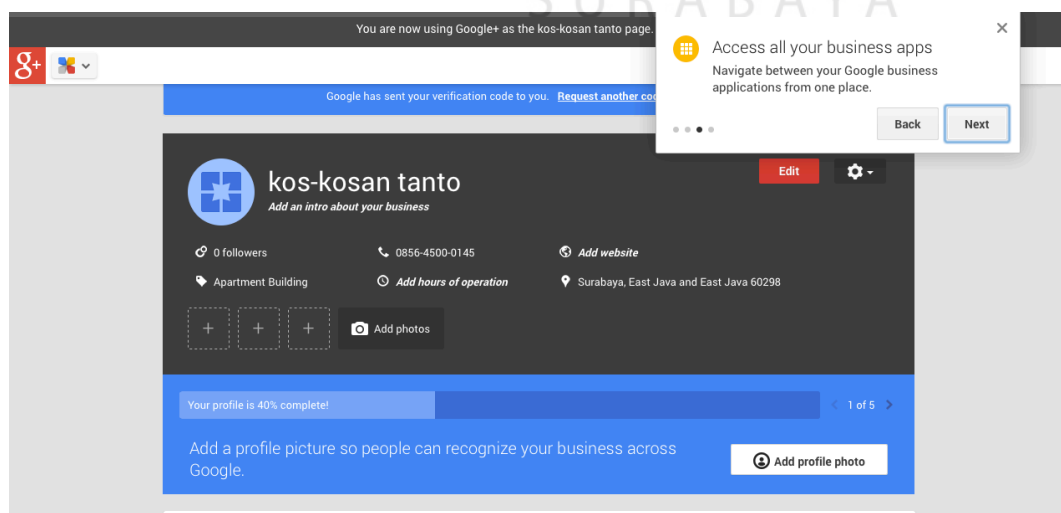
Setelah data-data yang dibutuhkan oleh Google for Business terpenuhi kita akan di arahkan menuju halaman informasi verifikasi, Google for Business membutuhkan waktu satu sampai dengan dua minggu untuk melakukan verifikasi

data kita, lalu akan memberi notifikasi melalui email. Gambar 4.4 Adalah situs Google for Business halaman verifikasi.



Gambar 4.4. Website Google for Business halaman verifikasi.

lalu kita akan di arahkan ke halaman dashboard dimana detail lokasi yang kita upload akan tampil dihalaman tersebut. Kita dapat menambahkan informasi tambahan pada halaman ini, dan dapat di tuliskan review mengenai bisnis kita oleh pelanggan kita, yang memiliki akun di *Google*. Gambar 4.5 Adalah situs Google for Business pada halaman dashboard.



Gambar 4.5. Website Google for Business pada halaman dashboard.

#### 4.2.2 Form Splash Screen

Splash Screen merupakan aplikasi pembuka dari aplikasi penentuan lokasi layanan umum ketika pertama kali aplikasi di jalankan, yang di tampilkan selama beberapa detik. Gambar 4.6 merupakan tampilan dari splash screen aplikasi.



Gambar 4.6. *Splash screen* aplikasi.

#### 4.2.3 Form List Menu

Form List ini digunakan untuk memilih lokasi layanan umum. Pada form ini ditampilkan list pilahan layanan umum yang ingin dicari, contohnya user akan mencari lokasi restaurant terdekat, user dapat memilih salah satu list menu dari layanan umum yaitu restaurant.

Seperti rancangan desain *input* dan *output* pada bab sebelumnya, tampilan dari list menu ber-orientasi portrait, dengan tujuan mempercepat user dalam memilih list menu. Karena sebagian user selalu memegang smartphone mereka dengan posisi portrait, dan user tidak akan kebingungan mengarahkan smartphonanya karena list menu juga ber-orientasi portrait. List menu terdiri dari gambar icon layanan umum, dan di sebelah kanan-nya adalah detail nama layanan umum. Dengan adanya icon layanan umum tersebut diharapkan user akan dapat mengerti maksud dari list menu itu tanpa membaca nama layanan umum tersebut. Untuk lebih jelasnya list menu dapat dilihat pada Gambar 4.7.





Gambar 4.7. *Form List Menu*

#### 4.2.4 Form Augmented Reality View

Form ini tampil pada saat user telah memilih lokasi layanan umum. Form ini adalah tampilan utama dari aplikasi pencarian lokasi layanan umum, kita akan menampilkan informasi lokasi layanan umum secara real time dengan menggunakan *augmented reality*, yaitu penggabungan objek maya (marker lokasi)



dengan objek nyata (kamera *view*). Tampilan form augmented reality view dapat dilihat pada Gambar 4.8.



Gambar 4.8. *Form Augmented Reality View.*

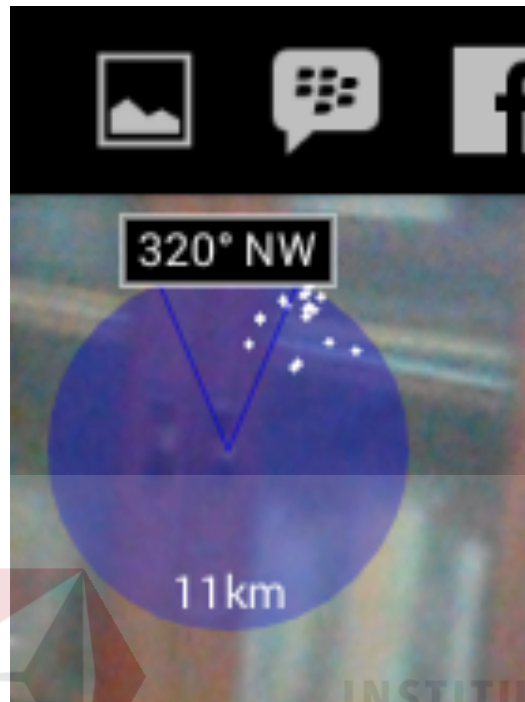
Pada saat form ini tampil aplikasi akan menjalankan proses yang panjang mulai dari aplikasi menampilkan kamera, menentukan lokasi pengguna saat ini, menentukan orientasi posisi smartphone, dan menentukan arah pengguna saat ini terhadap kutub utara (sensor kompas), yaitu dengan menjalankan sensor listener. Proses kedua adalah aplikasi akan mendownload data pencarian lokasi berdasarkan radius pencarian dan list menu pencarian, kemudian dari data lokasi layanan umum yaitu longitude dan altitude, maka dihitung sudut azimuth dari data lokasi tersebut, lalu di sesuaikan dengan ukuran objek pada layar yang sudah di sesuaikan dengan kondisi user saat ini (PaintableObject). Proses yang terakhir adalah menggambar objek-objek informasi lokasi layanan umum tsb. Pada form augmented relity view terdapat dua *widget* untuk memudahkan pengguna, yaitu *widget* radar dan radius.

Tampilan form augmented reality ber-orientasi landscape, dikarenakan jika orientasi layar landscape akan banyak *augmented reality item* yang tertangkap. Bagian kanan layar terdapat scrollbar untuk merubah radius, dengan cara menggeser scrollbar radius pencarian, jika ke atas maka radius pencarian akan semakin luas, jika kebawah akan semakin kecil atau sempit. Informasi tentang jarak radius pencarian ada di dalam radar. Letak radar di pojok kiri atas adalah untuk navigasi pencarian lokasi diluar jangkauan tampilan kamera. Augmented reality item adalah object maya yang di gambar pada kamera smartphone untuk menunjukkan lokasi layanan umum yang di cari,

#### **A. Gadget Radar Pada form Augmented Reality View**

Gadget Radar pada form augmented reality view digunakan untuk bantuan alat navigasi, jadi user dapat melihat augmented reality item yang tidak tampak pada jarak pandangnya. Letaknya berada pada atas kiri layout augmented reality view, berbentuk lingkaran dengan pusat di tengahnya, lingkaran tersebut merupakan radius dari pencarian lokasi, dan titik tengah adalah lokasi saat ini. Bagian segitiga di dalam lingkaran radar adalah untuk menggambarkan jarak pandang user. Terdapat titik-titik kecil di bagian dalam lingkaran adalah marker lokasi layanan umum, jadi jika user mengarahkan smartphone ke arah jarak pandang yang tidak ada marker layanan umum, maka tidak tampak *augmented reality* itemnya, maka user dapat mengarahkan jarak pandangnya dengan melihat arah radar, maka user masih dapat terbantu oleh adanya bantuan navigasi berupa radar tersebut. Di dalam linkaran ada informasi radius dengan format jarak kilometer, ini untuk memudahkan pengguna mengetahui radius yang di jangkau

untuk pencarian layanan umum. Untuk lebih jelasnya dapat dilihat pada gambar 4.9 adalah radar pada layout augmented reality view.



Gambar 4.9. Radar pada *layout augmented reality view*.

#### **B. Gadget Augmented Reality Item Pada Layout Augmented Reality View**

Gadget augmented reality item digunakan untuk penanda lokasi layanan umum dengan keterangan nama lokasinya. Icon dibuat dengan gambar yang mudah di ingat untuk mempermudah penandaan dan detail nama lokasi, agar user dapat dengan cepat membedakan tiap itemnya.



Gambar 4.10. *Augmented Reality* Item pada form *Augmented Reality View*.

### 4.3. Evaluasi

Tahap evaluasi ini bertujuan untuk mengetahui apakah proses utama pada system ini menghasilkan output seperti yang diharapkan. Proses utama pada aplikasi penentuan lokasi layanan umum menggunakan *augmented reality* ini adalah melakukan proses mengolah data lokasi layanan umum ke dalam *augmented reality* pada smartphone android. Tahapan uji coba fungsi aplikasi dan evaluasi uji komabilitas system.

#### 4.3.1. Uji coba fungsi aplikasi

Uji coba ini bertujuan untuk mengetahui apakah system yang dibuat dapat berjalan dengan benar sesuai dengan fungsi-fungsinya. Skenario uji coba ini dijalankan dengan cara membandingkan hasil perhitungan sudut azimuth yang di peroleh dari aplikasi penentuan lokasi dengan menggunakan *Augmented Reality*, dari aplikasi tersebut nantinya akan dibandingkan dengan perhitungan sudut azimuth hasil perhitungan manual menggunakan azimuth calculator yang diterapkan pada aplikasi penentuan lokasi layanan umum. Setelah perhitungan

azimuth secara manual dilakukan maka dibandingkan dengan aplikasi dari pihak ketiga untuk memastikan apakah perhitungan manual sudah benar dilakukan dan aplikasi penentuan lokasi layanan umum sudah sesuai dengan perhitungan pada aplikasi pihak ketiga.

Uji coba yang pertama adalah mencari lokasi ATM, langkah pertama dalam melakukan uji coba dilakukan dengan cara *manual*, yaitu menentukan posisi pengguna, dalam kasus ini pengguna berada pada posisi latitude -7.360418 dan longitude 112.764438. Selanjutnya adalah mencari lokasi layanan umum dengan *request Google Map API*. Diambil beberapa sampel untuk penentuan lokasi layanan umum. Selanjutnya data lokasi layanan umum akan di hitung dengan menggunakan rumus perhitungan sudut azimuth dengan menggunakan azimuth calculator. Berikut adalah Tabel 4.1, tabel uji coba sistem.

Tabel 4.1. Tabel uji coba sistem dengan cara manual pada ATM Bank BCA 6932-Tropodo.

Lokasi	Lat	Lng	Nama lokasi	Azimuth	distance
Awal	-7.360418	112.764438	ATM Bank BCA	333.3°	74 meter
Tujuan	-7.359825	112.764137	6932-Tropodo		

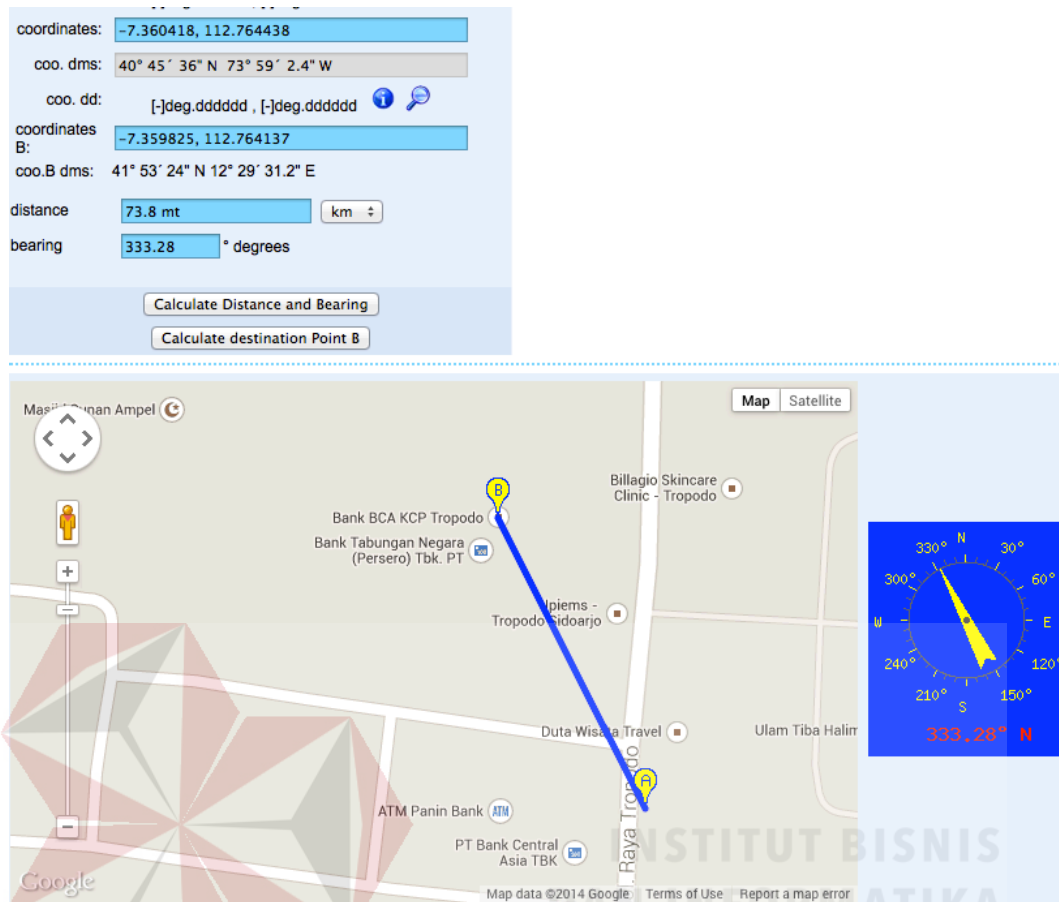
Uji coba selanjutnya adalah dengan menggunakan aplikasi penentuan lokasi layanan umum, dengan tujuan mengetahui posisi horizontal lokasi layanan umum, yang ditampilkan pada layar device dengan cara mengarahkan kamera pada sudut azimuth sesuai hasil uji coba yang pertama, dan apakah jarak lokasi sudah dihitung dengan benar. Dalam kasus ini pengguna berada masih berada pada posisi latitude -7.360418 dan longitude 112.764137. Hasil uji coba manual adalah 333.3° azimuth dan jaraknya adalah 78 meter. Uji coba aplikasi pada ATM



Bank BCA 6932-Tropodo dapat dilihat pada gambar 4.11. Uji coba pada pihak ketiga dilakukan dengan melakukan input lokasi pengguna saat ini dan lokasi layanan umum yang dituju. Hasil dari perhitungan jarak dan sudut azimuth pada lokasi ATM Bank BCA 6932-Tropodo menggunakan aplikasi pihak ketiga dapat dilihat pada gambar 4.12.



gambar 4.11. Uji coba aplikasi pada ATM Bank BCA 6932-Tropodo.



gambar 4.12. Uji coba aplikasi pihak ketiga pada ATM Bank BCA 6932-Tropodo.

Uji coba yang kedua adalah mencari lokasi restaurant pada list menu lokasi layanan umum, lokasi latitude pengguna adalah -7.372135, dan longitude pengguna adalah 112.759615, dan hasil dari uji coba manual dari lokasi pengguna dengan lokasi layanan umum pada kasus ini dapat dilihat pada table 4.2 uji coba perhitungan sudut azimuth dan jarak secara manual.

Tabel 4.2. Tabel uji coba sistem dengan cara manual pada MM Resto Raya

Bypass Juanda Baru.

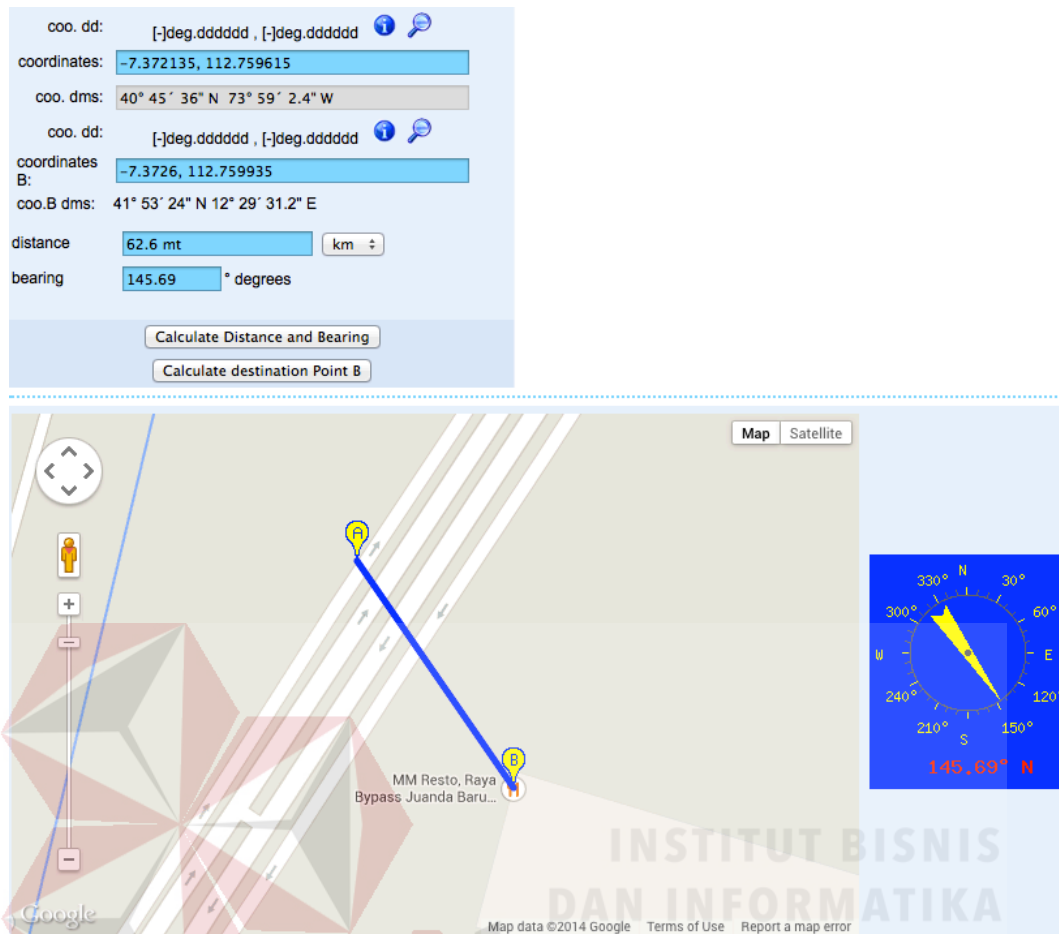
Lokasi	Lat	Lng	Nama lokasi	Azimuth	distance
Awal	-7.372135	112.759615	MM Resto Raya	145.7°	63 m
Tujuan	-7.3726	112.759935	Bypass Juanda		

pada perhitungan manual adalah  $145.7^\circ$  sudut azimuth dan berjarak 63 m, hasil dari aplikasi kami mengarahkan *device* pada sudut dari titik nol utara. Uji coba aplikasi pada MM Resto Raya Bypass Juanda Baru dapat dilihat pada gambar 4.13. Uji coba pada pihak ketiga dilakukan dengan melakukan input lokasi pengguna saat ini dan lokasi layanan umum yang dituju. Hasil dari perhitungan jarak dan sudut azimuth pada lokasi MM Resto Raya Bypass Juanda Baru menggunakan aplikasi pihak ketiga dapat dilihat pada gambar 4.14.



Gambar 4.13. Uji coba aplikasi pada MM Resto Raya Bypass Juanda Baru.





Gambar 4.14. Uji coba aplikasi pihak ketiga pada MM Resto Raya Bypass Juanda Baru.

Uji coba yang kedua adalah mencari lokasi hotel pada list menu lokasi layanan umum, lokasi latitude pengguna adalah -7.368515, dan longitude pengguna adalah 112.763514, dan hasil dari uji coba manual dari lokasi pengguna dengan lokasi layanan umum pada kasus ini dapat dilihat pada table 4.3 uji coba perhitungan sudut azimuth dan jarak secara manual.

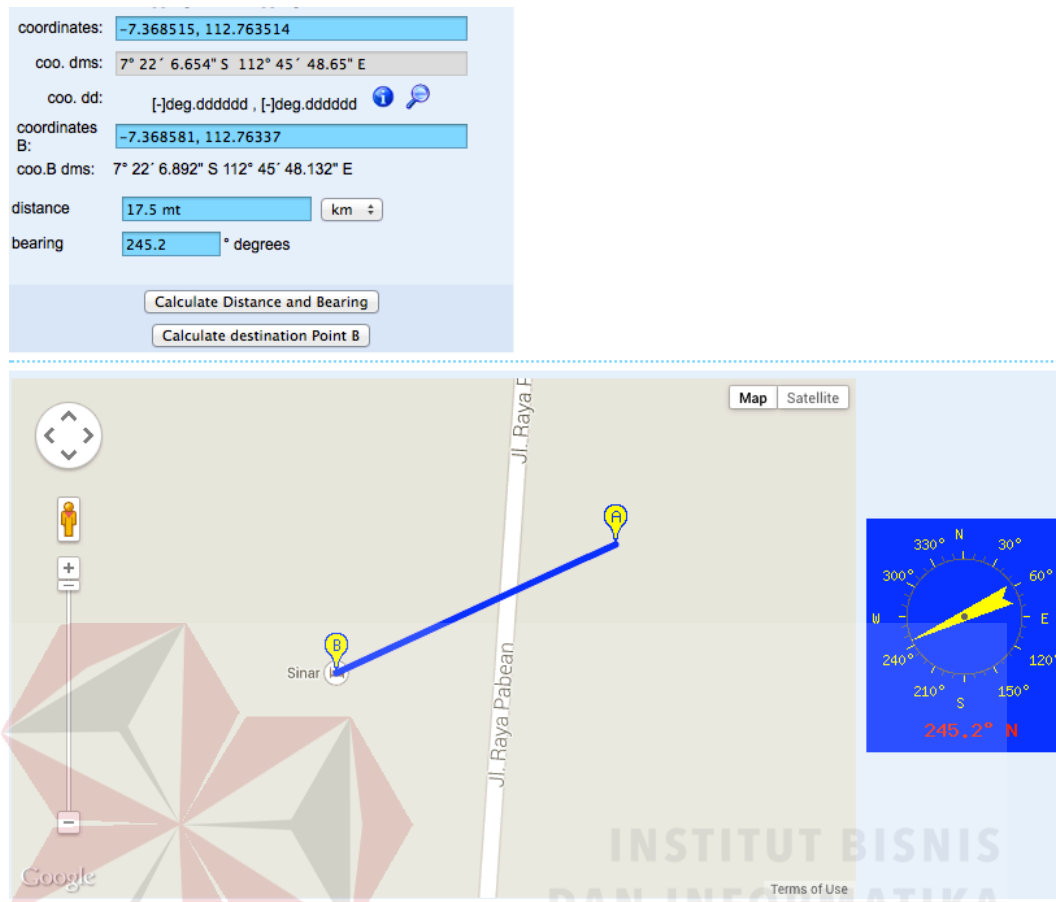
Tabel 4.3 tabel uji coba sistem dengan cara manual pada Hotel Sinar II.

Lokasi	Lat	Lng	Nama lokasi	Azimuth	distance
Awal	- 7.368515	112.763514	Hotel Sinar II	245.2°	21 m
Tujuan	- 7.368581	112.76337			

pada perhitungan manual adalah  $245.2^\circ$  sudut azimuth dan berjarak 21 m, hasil dari aplikasi kami mengarahkan *device* pada sudut dari titik nol utara. Uji coba aplikasi pada Hotel Sinar II dapat dilihat pada gambar 4.15. Uji coba pada pihak ketiga dilakukan dengan melakukan input lokasi pengguna saat ini dan lokasi layanan umum yang dituju. Hasil dari perhitungan jarak dan sudut azimuth pada lokasi Hotel Sinar II menggunakan aplikasi pihak ketiga dapat dilihat pada gambar 4.16.



Gambar 4.15. Uji coba pada Hotel Sinar II



Gambar 4.16. Uji coba aplikasi pihak ketiga pada Hotel Sinar II

Uji coba yang keempat adalah mencari lokasi Low Budget Cafe atau Warung Kopi pada list menu lokasi layanan umum, lokasi latitude pengguna adalah -7.371744, dan longitude pengguna adalah 112.759864, dan hasil dari uji coba manual dari lokasi pengguna dengan lokasi layanan umum pada kasus ini dapat dilihat pada table 4.4 uji coba perhitungan sudut azimuth dan jarak secara manual.

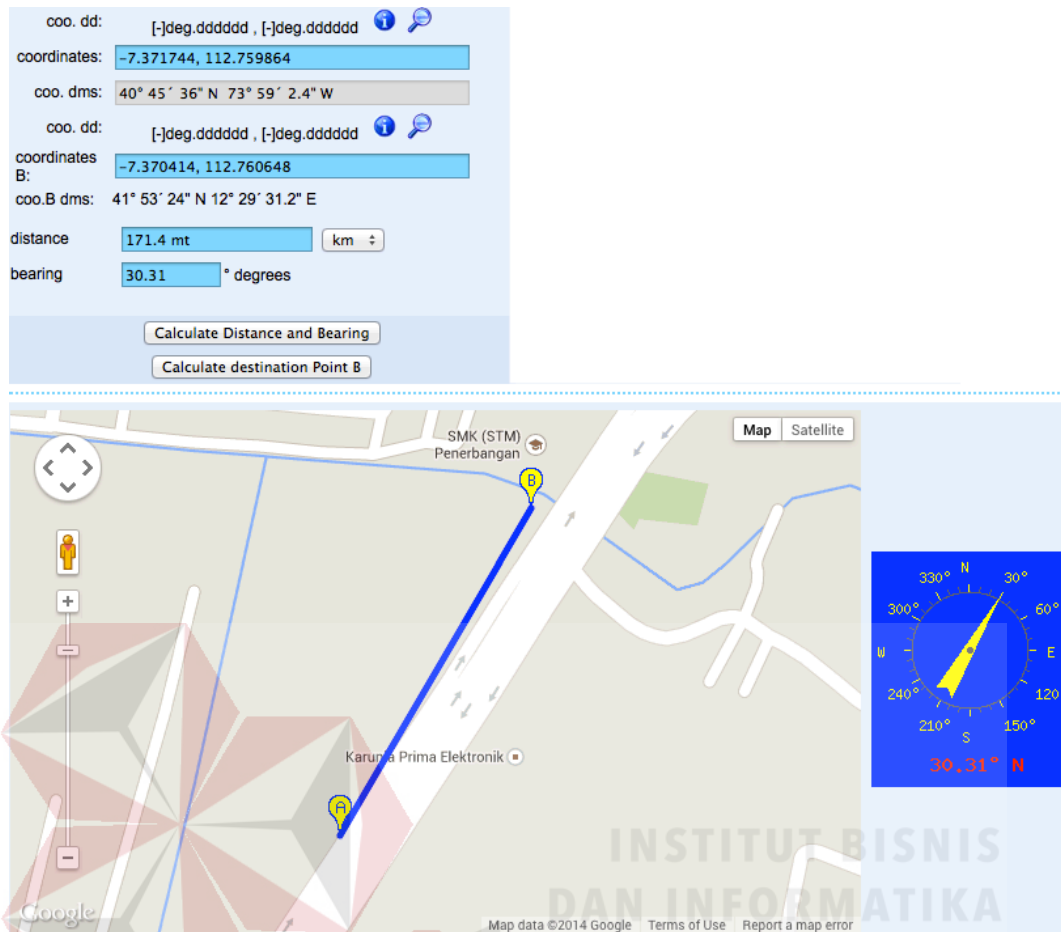
Tabel 4.4 tabel uji coba sistem dengan cara manual pada Warkop Urip.

Lokasi	Lat	Lng	Nama lokasi	Azimuth	distance
Awal	-7.371744	112.759864	Warkop Urip	30,3°	171m
Tujuan	-7.370414	112.760648			

pada perhitungan manual adalah  $30,3^\circ$  sudut azimuth dan berjarak 171 m, hasil dari aplikasi kami mengarahkan *device* pada sudut dari titik nol utara. Uji coba aplikasi pada Warkop Urip dapat dilihat pada gambar 4.17. Uji coba pada pihak ketiga dilakukan dengan melakukan input lokasi pengguna saat ini dan lokasi layanan umum yang dituju. Hasil dari perhitungan jarak dan sudut azimuth pada lokasi Warkop Urip menggunakan aplikasi pihak ketiga dapat dilihat pada gambar 4.18.



Gambar 4.17. Uji coba pada Warkop Urip



Gambar 4.18. Uji coba aplikasi pihak ketiga pada Warkop Urip

Jadi dapat disimpulkan Gap error dari keempat sampel diatas yang dapat dilihat perbedaan antara hasil perhitungan sudut azimuth dan jarak antara pengguna dan lokasi layanan umum secara manual yang di terapkan pada aplikasi penentuan lokasi layanan umum dengan aplikasi perhitungan sudut azimuth dan jarak, yaitu aplikasi pihak ketiga, yang dapat dilihat pada tabel 4.5. Bahwa perbedaan antara hasil perhitungan dalam aplikasi dengan hasil perhitungan aplikasi pihak ketiga tidak berbeda secara signifikan. Sehingga dapat dikatakan bahwa hasil perhitungan dalam aplikasi ini valid.

Table 4.5. Gap Error Perbandingan perhitungan manual sudut azimuth dan jarak pada perhitungan manual dan aplikasi

Nama lokasi	perhitungan manual		perhitungan aplikasi pihak ketiga		Gap Error	
	Azimuth	distance	Azimuth	distance	selisih azimuth	selisih jarak
ATM Bank BCA 6932-Tropodo	333.3°	74 m	333,2°	73,8 m	0,1°	0,2 m
MM Resto Raya Bypass Juanda Baru	145.7°	63 m	145,69	62,6 m	0,1°	0,4 m
Hotel Sinar II	245.2°	21 m	245,2°	17,5 m	0°	0 m
Warkop Urip	30,3°	171 m	30,31°	171,4 m	0,1°	0,4 m

#### 4.4. Uji coba kompatibilitas system

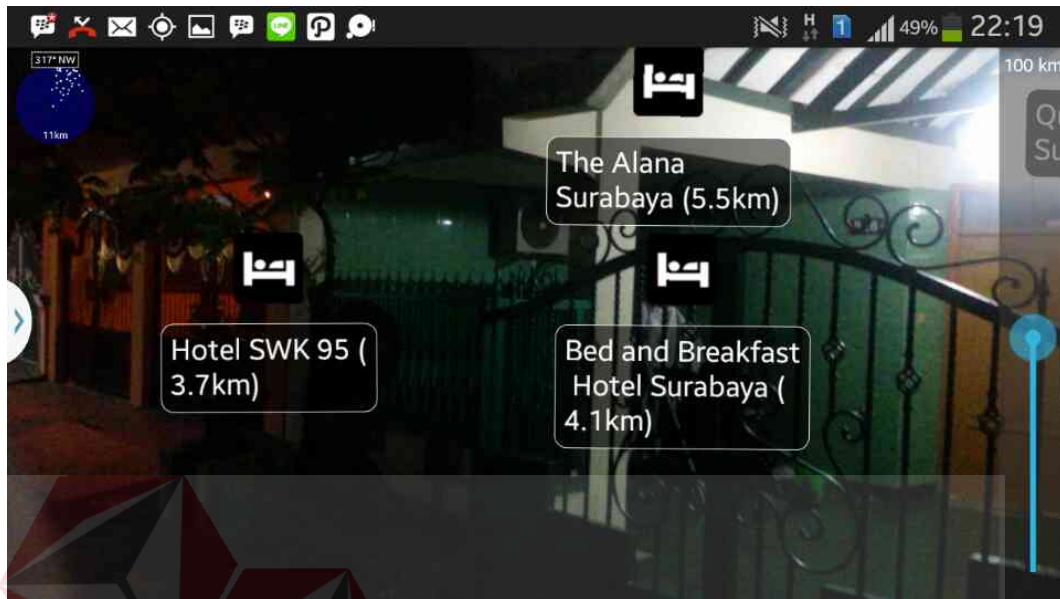
Proses uji coba ini dilakukan untuk mengetahui tingkat kompatibilitas system. Uji coba kompatibilitas system dilakukan dengan menggunakan 2 versi Operating System Android, yaitu minimum requirement dan maximum requirement Android Version, berikut adalah daftar versi android tersebut :

1. Android GingerBread (2.3.4)
2. Android Kitkat (4.4)

Uji coba dilakukan dengan menggunakan *device* Android karena menggunakan banyak sensor yang tidak ada pada emulator. Uji coba ini bertujuan untuk mengetahui tingkat kompatibilitas dari fungsi-fungsi yang ada pada aplikasi apakah dapat berfungsi pada minimum requirement dan maximum requirement, dan kedua *device* dengan *operating system* tersebut dapat berjalan dengan baik, seperti dapat dilihat pada gambar 4.19 aplikasi berjalan pada *device android*,



*galaxy S2* dan operating system *gingerbread(2.3.4)*. Pada gambar 4.20 aplikasi berjalan pada *device android, NEXUS 4* dan operating system *Kitkat(4.4)*.



Gambar 4.19. Aplikasi berjalan pada *device android, galaxy S2* dan operating system *gingerbread(2.3.4)*.



Gambar 4.20 aplikasi berjalan pada *device android, NEXUS 4* dan operating system *Kitkat(4.4)*.

Berikut adalah spesifikasi perangkat *android* yang dilakukan *testing* terhadap aplikasi penentuan lokasi layanan umum menggunakan *augmented reality* tersebut.

Tabel 4.6. Hasil pengujian pada perangkat *android*

Nama perangkat	Spesifikasi perangkat		Hasil
Samsung Galaxy SII	OS	2.3.4 GingerBread	Ok
	Processor	Dual-core 1.2 GHz	
	RAM	1 GB RAM	
	Internal Memory	16 GB	
	Back Camera	Ya ada	
LG Nexus 4	OS	4.4.2 KitKat	Ok
	Processor	Quad-core 1.5 GHz	
	RAM	2 GB RAM	
	Internal Memory	16 GB	
	Back Camera	Ya ada	
Sony Xperia U	OS	2.3.4 GingerBread	Ok
	Processor	Dual-core 1 GHz	
	RAM	512 MB RAM	
	Internal Memory	4 GB	
	Back Camera	Ya Ada	
Samsung Galaxy Grand 2	OS	4.3 JellyBean	Ok
	Processor	Quad-core 1,2 GHz	
	RAM	1,5 GB RAM	
	Internal Memory	8 GB	
	Back Camera	Ya ada	

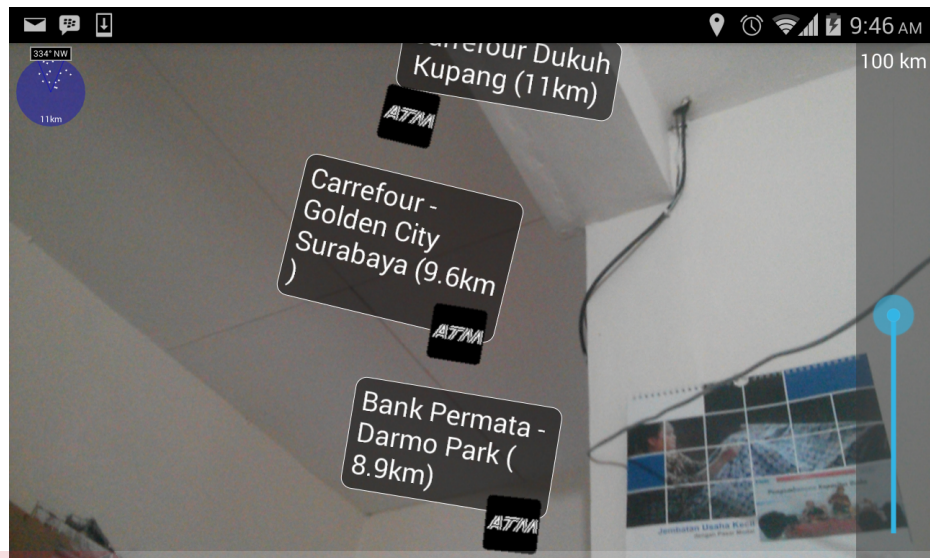


Bug akan terjadi ketika radius yang ditentukan oleh pengguna jauh dan data yang ditampilkan banyak dengan lokasi sejajar pada arah pandang pengguna seperti dapat dilihat pada gambar 4.21 radar menunjukkan lokasi sejajar dengan arah pandang pengguna.

Karena tabrakan sudah dihindari dengan cara lokasi yang lebih jauh akan ditempatkan di lokasi layar perangkat lebih keatas, maka data lokasi akan kurang valid, karena pengguna akan mengarahkan perangkat hingga ke arah langit-langit, seperti dapat dilihat pada gambar 4.22, perangkat menunjukkan lokasi pada langit-langit karena lokasi sejajar pada arah pandang pengguna



Gambar 4.21. Radar menunjukkan lokasi sejajar pada arah pandang pengguna



Gambar 4.22. Perangkat menunjukan lokasi pada langit-langit karena lokasi sejajar pada arah pandang pengguna

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Setelah dilakukan analisis, perancangan sistem dan pembuatan aplikasi penentuan lokasi layanan umum menggunakan *augmented reality* pada *smartphone android*, serta dilakukan evaluasi hasil penelitiannya, maka dapat diambil kesimpulan sebagai berikut :

1. Data yang dibutuhkan untuk menerapkan teknologi *augmented reality* adalah sudut azimuth, antara lokasi layanan umum dan pengguna didapatkan dari perhitungan menggunakan persamaan trigonometri.
2. Berdasarkan sudut azimuth yang dibentuk antara pengguna dengan lokasi layanan umum serta arah yang di tuju kompas pada *device*, dapat diketahui letak horizontal lokasi layanan umum pada layar *device*.
3. Tema aplikasi yang dibuat berpengaruh terhadap desain tampilan. Pada tugas akhir ini dibuat untuk memudahkan atau mempercepat pengguna melakukan pencarian lokasi layanan umum, jadi desain aplikasi tidak menggunakan kalimat yang panjang, dan di dukung oleh ikon-ikon yang mudah dimengerti oleh pengguna sehingga pengguna dapat mengerti tanpa membaca detail atau nama dari lokasi layanan umum.
4. Metode *Augmented Reality* yang digunakan adalah *location based*, jadi data yang diolah adalah data lokasi pengguna dengan data lokasi layanan umum, data gambar pemandangan nyata tidak diolah menggunakan image processing untuk menampilkan informasi ke dalam kondisi nyata, melainkan hanya untuk menggambarkan kondisi nyata saat ini. Sehingga

Aplikasi akan menampilkan kondisi nyata dalam kamera perangkat *android* dan informasi layanan umum ditampilkan dengan mengolah data lokasi pengguna dengan lokasi layanan umum.

## 5.2. Saran

Saran untuk pengembangan system adalah dengan membuat system yang serupa namun lebih sempurna. Adapun beberapa saran yang dapat di sampaikan untuk mengembangkan system yang dibuat antara lain :

1. *Augmented reality* akan menjadi teknologi yang populer di masa depan, oleh karena itu perlu dikembangkan aplikasi yang lebih dari sekedar penentuan lokasi, mungkin dapat ditambahkan sistem navigasi untuk tempat yang ingin di tuju.
2. Dapat di implementasikan pada *platform* lain yang memiliki sensor-sensor yang di butuhkan seperti pada *google glass*, mungkin fungsinya lebih baik karena *google glass* adalah *wearable device*.
3. Baterai pada *device* yang sedang menjalankan aplikasi akan boros, dikarenakan semua sensor hidup (*accelerometer*, *GPS*, dan kompas), mungkin dapat di *reduce* pemakaian sensornya, agar baterai bisa lebih irit.

## DAFTAR PUSTAKA

Abidin, Dr H. Z., 2000. “*Penentuan Posisi dengan GPS dan Aplikasinya.*” PT.

Pradnya Paramita, Jakarta.

Azuma, Ronald T, 1997. “*Survey of Augmented Reality.*” Malibu.

El-Rabbany, Ahmed., 2002. “*Introduction to GPS The Global Positioning System*”.

London.

GoogleInc, 2010. “Documentation- Android” (<https://developer.google.com>),

diakses pada 18 maret 2013)

Herlambang, Soendoro, dan Haryanto, Tanuwijaya., 2004. “*Sistem Informasi:*

*Konsep, Teknologi, dan Manajemen.*” Graha Ilmu, Yogyakarta.

Safaat H, Nazruddin,. 2011. “*Pemrograman Aplikasi Mobile Smartphone dan*

*Tablet PC Berbasis Android.*” Informatika, Bandung.

