

**PENERAPAN TEKNOLOGI *AUGMENTED REALITY*
PADA PENGENALAN SISTEM TATA SURYA
BERBASIS ANDROID**



SKRIPSI

Diajukan untuk memenuhi salah satu syarat kelulusan Program Sarjana

RIDWAN SETIAWAN

12135288

Program Studi Teknik Informatika

STMIK Nusa Mandiri Jakarta

Jakarta

2017

PERSEMBAHAN

Pada lembar persembahan ini, penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang sangat mendukung penulis dalam pembuatan dan penyusunan skripsi ini, adapun yang dimaksud adalah sebagai berikut:

1. Kepada orang tua penulis dan keluarga penulis yang selalu memberikan dukungan berupa materi dan do'a kepada penulis sehingga penulis bisa menyelesaikan karya tulis ini.
2. Kepada Nada Fia Restu yang selalu memberikan motivasi dan semangat.
3. Kepada Ibu Yulia dan Pak Wawan yang membimbing dan tak pernah berhenti untuk mendukung penulis.
4. Kepada Doni Permadi yang selalu setia dan memberikan dukungan yang luar biasa kepada penulis.
5. Kepada Dedi Hisban yang memberikan jasa layanan print dan dukungan kepada penulis.
6. Kepada Rafli Sugara dan Agung Prayogo yang selalu ada disaat penulis membutuhkan.

Semoga Tuhan Yang Maha Kuasa memberikan balasan yang setimpal dengan kebaikan yang penulis sampaikan diatas. Amin Ya Rabbal'Alamin.

SURAT PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan dibawah ini:

Nama : Ridwan Setiawan
NIM : 12135288
Perguruan Tinggi : STMIK Nusa Mandiri Jakarta

Dengan ini menyatakan bahwa skripsi yang telah saya buat dengan judul: **“Penerapan Teknologi Augmented Reality Pada Pengenalan Sistem Tata Surya Berbasis Android”**, adalah asli (orisinil) atau tidak plagiat (menjiplak) dan belum pernah diterbitkan/dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya tanpa ada paksaan dari pihak manapun juga. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa skripsi yang telah saya buat adalah karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari **Sekolah Tinggi Manajemen Informatika & Komputer Nusa Mandiri** dicabut/dibatalkan.

Dibuat di : Jakarta

Pada tanggal : 7 Agustus 2017

Yang menyatakan



Ridwan Setiawan

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan dibawah ini, saya:

Nama : Ridwan Setiawan
NIM : 12135288
Program Studi : Teknik Informatika
Perguruan Tinggi : STMIK Nusa Mandiri Jakarta

Dengan ini menyetujui untuk memberikan ijin kepada pihak **Sekolah Tinggi Manajemen Informatika & Komputer Nusa Mandiri**, Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti-Free Right*) atas karya ilmiah kami berjudul: **“Penerapan Teknologi *Augmented Reality* Pada Pengenalan Sistem Tata Surya Berbasis Android”**, beserta perangkat yang diperlukan (apabila).

Dengan **Hak Bebas Royalti Non-Eksklusif** ini pihak **Sekolah Tinggi Manajemen Informatika & Komputer Nusa Mandiri** berhak menyimpan, mengalih-media atau *format*-kan, mengelolanya dalam pangkalan data (*database*), mendistribusikannya dan menampilkan atau mempublikasikannya di *internet* atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari kami selama tetap mencantumkan nama kami sebagai penulis/pencipta karya ilmiah tersebut.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak **Sekolah Tinggi Manajemen Informatika & Komputer Nusa Mandiri**, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta
Pada tanggal : 7 Agustus 2017
Yang menyatakan



Ridwan Setiawan

PERSETUJUAN DAN PENGESAHAN SKRIPSI

Skripsi ini diajukan oleh:

Nama : RIDWAN SETIAWAN
NIM : 12135288
Program Studi : TEKNIK INFORMATIKA
Jenjang : STRATA-1
Judul Skripsi : Penerapan Teknologi Augmented Reality Pada Pengenalan Sistem Tata Surya Berbasis Android

Telah dipertahankan pada periode 2017-A dihadapan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh SARJANA KOMPUTER (S.Kom) pada Program STRATA-1 Program Studi Teknik Informatika di Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri.

Jakarta, 14 September 2017

PEMBIMBING SKRIPSI

Dosen Pembimbing : Yulia Safitri, M.Kom

Asisten Pembimbing : Wawan Gunawan, S.Kom

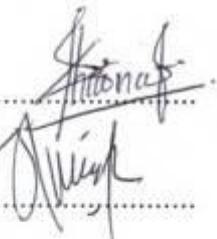
ain

.....
.....

DEWAN PENGUJI

Penguji I : Astriana Mulyani, S.Si, M.Kom

Penguji II : Syaifur Rahmatullah Abdul R,
: M.Kom


.....
.....

PANDUAN PENGGUNAAN HAK CIPTA

Skripsi sarjana yang berjudul “**Penerapan Teknologi Augmented Reality Pada Pengenalan Sistem Tata Surya Berbasis Android**” adalah hasil karya tulis asli RIDWAN SETIAWAN dan bukan hasil terbitan sehingga peredaran karya tulis hanya berlaku dilingkungan akademik saja, serta memiliki hak cipta. Oleh karena itu, dilarang keras untuk menggandakan baik sebagian maupun seluruhnya karya tulis ini tanpa seizin penulis.

Referensi kepustakaan diperkenankan untuk dicatat tetapi pengutipan atau peringkasan isi tulisan hanya dapat dilakukan dengan seizin penulis dan disertai ketentuan pengutipan secara ilmiah dengan menyebutkan sumbernya.

Untuk keperluan perizinan pada pemilik dapat menghubungi informasi yang tertera di bawah ini:

Nama : Ridwan Setiawan
Alamat : Jl. Madrasah I RT.007/009 Jakarta Barat
No. Telp : 087868476002
Email : rsetiawan85@yahoo.co.id

KATA PENGANTAR

Puji syukur alhamdulillah, penulis panjatkan kehadiran Allah SWT, yang telah melimpahkan rahmat, taufik dan hidayah-Nya, sehingga pada akhirnya penulis dapat menyelesaikan Skripsi ini dengan baik. Dimana skripsi ini penulis sajikan dalam bentuk buku yang sederhana. Adapun judul skripsi yang penulis ambil sebagai berikut, “**PENERAPAN TEKNOLOGI *AUGMENTED REALITY* PADA PENGENALAN SISTEM TATA SURYA BERBASIS ANDROID**”. Sebagai salah satu syarat kelulusan Program Strata Satu (S1)STMIK Nusa Mandiri.

Penulis menyadari bahwa dalam proses penulisan skripsi ini banyak mengalami kendala, namun berkat bantuan, bimbingan, kerjasama dari berbagai pihak dan berkah dari Allah SWT sehingga kendala-kendala yang dihadapi tersebut dapat diatas. Oleh karena itu pada kesempatan ini, iinkanlah penulis menyampaikan ucapan terimakasih kepada:

1. Ketua STMIK Nusa Mandiri.
2. Pembantu Ketua I STMIK Nusa Mandiri.
3. Ketua Program Studi Teknik Informatika STMIK Nusa Mandiri.
4. Bapak Yulia Safitri M.Kom selaku Dosen Pembimbing I Skripsi.
5. Bapak Wawan Gunawan, S.Kom,.M.T selaku Dosen Pembimbing II Skripsi.
6. Bapak/Ibu Dosen Teknik Informatika STMIK Nusa Mandiri yang telah memberikan penulis dengan semua bahan yang diperlukan.
7. Orang tua tercinta yang telah memberikan dukungan moral maupun spiritual.

8. Rekan-rekan mahasiswa yang telah banyak memberikan bantuan, dorongan serta motivasi sehingga skripsi ini dapat terselesaikan.

Serta semua pihak yang terlalu banyak untuk disebut satu persatu sehingga terwujudnya penulisan ini. Penulis menyadari penulisan skripsi ini masih jauh sekali dari sempurna, untuk itu penulis mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan dimasa yang akan datang.

Akhir kata semoga skripsi ini dapat berguna bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

Jakarta, 07 Agustus 2017

Penulis

Ridwan Setiawan

ABSTRAK

Ridwan Setiawan (12135288), Penerapan Teknologi *Augmented Reality* Pada Pengenalan Sistem Tata Surya Berbasis Android.

Augmented reality (AR) adalah jenis teknologi interaktif menggabungkan benda nyata dan virtual yang akan menghasilkan objek 3D yang akan ditampilkan pada layar. *Augmented reality* yang telah diaplikasikan memiliki cara kerja berdasarkan deteksi citra atau gambar dan biasa disebut *marker*, dengan menggunakan kamera *smartphone* kemudian mendeteksi *marker* yang telah dicetak. *Augmented reality* banyak digunakan diberbagai bidang, salah satunya bidang pendidikan. Pada bidang pendidikan *augmented reality* digunakan sebagai media pembelajaran agar lebih menarik. Teknologi *augmented reality* ini dapat diterapkan dalam pembelajaran sistem tata surya. Penggunaan teknologi *augmented reality* diharapkan bisa menampilkan objek berupa planet-planet dalam sistem tata surya secara virtual 3D dengan menggunakan gambar yang dijadikan *marker*. Pembuatan aplikasi ini dibangun menggunakan *software Unity 3D*.

Kata Kunci: *Augmented reality, 3D, Unity3D, Marker, Sistem Tata Surya.*

ABSTRACT

Ridwan Setiawan (12135288), *Application of Augmented Reality Technology on The Introduction of Android Based Solar System.*

Augmented reality (AR) is a type of interface technology combining real and virtual objects will generate 3D objects that will be displayed on the screen. Augmented reality has been applied has a way of working is based on detection of image or picture and so-called marker, using a smartphone's camera then detects markers that have been printed. Augmented reality widely used in various fields, one of which is education. In the field of education augmented reality is used as a medium of learning to make it more interesting. Augmented reality technology can be applied in learning of the solar system. The use augmented reality technology is expected to display the objects form a planets in the solar system virtual 3D images made using marker. Making an applciation built using software Unity3D

Keywords: *Augmented reality, 3D, Unity3D, Marker, Solar System.*

DAFTAR ISI

LEMBAR JUDUL SKRIPSI	i
LEMBAR PERSEMBAHAN	ii
LEMBAR PERNYATAAN KEASLIAN SKRIPSI.....	iii
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYAILMIAH ...	iv
LEMBAR PERSETUJUAN DAN PENGESAHAN SKRIPSI	v
LEMBAR PANDUAN PENGGUNAAN HAK CIPTA.....	vi
.....	
KATA PENGANTAR	vii
ABSTRAK.....	ix
DAFTAR ISI	xi
DAFTAR SIMBOL	xiv
DAFTAR GAMB	xviii
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Perumusan Masalah.....	2
1.3. Maksud dan Tujuan	3
1.4. Metode Peneltiaan.....	3
1.4.1. Metode Pengumpulan Data	4
1.4.1. Metode Pengembangan Aplikasi.....	4
1. Metode SDLC.....	4
2. Model Waterfall.....	5
1.5. Ruang Lingkup	7
BAB II LANDASAN TEORI	8
2.1. Tinjauan Jurnal.....	8
2.2. Konsep Dasar Program	9
2.2.1. Augmented Reality	9

2.2.2. Sejarah Augmented Reality	10
2.2.3. Sejarah OS Android	11
2.2.4. Android SDK (Software Development Kit)	14
2.2.5. Vuforia SDK (Software Development Kit)	15
2.2.6. C#	16
2.2.7. Unified Modeling Language (UML)	16
2.2.8. Diagram UML	17
1. Use Case Diagram	17
2. Class Diagram	17
3. Activity Diagram.....	18
4. Sequence Diagram.....	19
2.3. Metode Algoritma	19
2.3.1. Algoritma Fast Corner Detection	19
2.4. Pengujian Aplikasi	21
2.5. Peralatan Pendukung.....	22

BAB III ANALISA KEBUTUHAN DAN PERANCANGAN SISTEM..... 24

3.1. Analisa Kebutuhan Sistem	24
3.1.1. Identifikasi Masalah	25
3.1.2. Analisa Kebutuhan	25
1. Microsoft Windows 7	26
2. Unity Editor	27
3. JDK (Java Development Kit)	27
4. CorelDraw X4	28
5. Enterprise Architect	29
3.2. Desain.....	29
3.2.1. RancanganAlgoritma.....	30
1. Markerless	30
2. Algoritma FAST Corner Detection	30
3.2.2. Software Architecture.....	33
3.2.3. Perancangan Antarmuka (Interface Design).....	41

3.3. Implementasi	52
3.4. Pengujian	57

BAB IV PENUTUP..... 62

4.1. Kesimpulan.....	62
----------------------	----

4.2. Saran - Saran.....	62
-------------------------	----

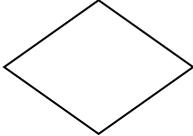
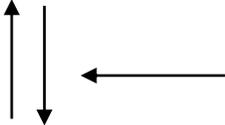
DAFTAR PUSTAKA

DAFTAR RIWAYAT HIDUP

LEMBAR KONSULTAN BIMBINGAN

LAMPIRAN

DAFTAR SIMBOL

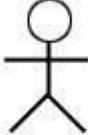
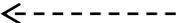
a.Simbol <i>Flow Chart</i>	
	<p>TERMINAL Digunakan untuk menggambarkan awal dan akhir dari suatu kegiatan.</p>
	<p>DECISION Digunakan untuk menggambarkan proses pengujian suatu kondisi yang ada.</p>
	<p>FLOW LINE Digunakan untuk menggambarkan hubungan proses dari satu proses ke proses lainnya.</p>
	<p>PROCCES Digunakan untuk analisa sistem untuk menggambarkan proses dalam suatu prosedur</p>
	<p>INPUT / OUTPUT Digunakan untuk menggambarkan proses memasukan data yang berupa pembacaan data dan sekaligus proses keluaran yang berupa pencetakan data.</p>
	<p>DOCUMENT Digunakan untuk iput berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas</p>
	<p>CONNECTOR Digunakan untuk menghubungkan bagian diagram alir pada halaman yang sama.</p>

b. Simbol *Use Case*

	<p>ACTOR Menspesifikasikan peran yang pengguna gunakan ketika berinteraksi dengan use case</p>
	<p>SYSTEM BOUNDARY Menspesifikasikan paket yang menampilkan sistem secara terbatas.</p>
	<p>EXTEND Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.</p>
	<p>ASSOCIATION Menghubungkan antara objek satu dengan objek lainnya.</p>
	<p>USE CASE Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.</p>

c. Simbol Activity Diagram

	<p>ACTIVITY Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain</p>
	<p>INITIAL NODE Bagaimana objek dibentuk atau diawali atau bisa disebut titik awal.</p>
	<p>ACTIVITY FINAL NODE Bagaimana objek diakhiri atau bisa disebut titik akhir.</p>
	<p>EXTEND Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.</p>
	<p>ASSOCIATION Menghubungkan antara objek satu dengan objek lainnya.</p>

d. Simbol <i>Sequence Diagram</i>	
	OBJECT LIFELINE <i>Object Lifeline</i> : menggambarkan <i>object</i> apa saja yang terlibat.
	ACTOR <i>Actor</i> : menggambarkan hubungan <i>actor</i> yang terlibat.
	ACTIVATION <i>Activation</i> : menggambarkan hubungan antara <i>object</i> dengan <i>message</i> .
	MESSAGE (CALL) <i>Message(call)</i> : menggambarkan alur <i>message</i> yang merupakan kejadian objek pengirim <i>lifeline</i> ke objek penerima <i>lifeline</i> .
	MESSAGE (RETURN) <i>Message(return)</i> : menggambarkan alur pengambilan <i>message</i> ke objek pemanggil dan tanda bahwa objek penerima telah menyelesaikan prosesnya.

e. Simbol <i>Class Diagram</i>				
<table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">Site Config</td> </tr> <tr> <td>+sqlDNS:string</td> </tr> <tr> <td>+Adminemail:String</td> </tr> </table>	Site Config	+sqlDNS:string	+Adminemail:String	CLASS <i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan <i>property/atributclass</i> . Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> .
Site Config				
+sqlDNS:string				
+Adminemail:String				

<p><u>1..n Owned by 1</u></p>	<p>ASSOCIATION Sebuah <i>asosiasi</i> merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i>. Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum <i>multiplisitas</i> pada sebuah <i>relationship</i>. (Contoh: <i>One-to-one</i>, <i>one-to-many</i>, <i>many-to-many</i>).</p>
	<p>DEPENDENCY Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i>. Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain.</p>
	<p>GENERALISATION Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).</p>

DAFTAR GAMBAR

	Halaman
Gambar I.1 Ilustrasi Model Waterfall	5
Gambar II.1 Unity Editor.....	21
Gambar II.2 <i>CorelDraw</i>	22
Gambar III.1 <i>Windows 7 Ultimate</i>	26
Gambar III.2 <i>Unity Editor</i>	27
Gambar III.3 <i>JDK (Java Development Kit)</i>	28
Gambar III.4 <i>CorelDRAW X4</i>	28
Gambar III.5 <i>Enterprise Architect</i>	29
Gambar III.6 Tampilan Titik Awal Yang di Uji.....	32
Gambar III.7 Empat Titik Koordinat.....	32
Gambar III.8 Tiga Titik Yang Memenuhi Syarat <i>FAST Corner Detection</i>	33
Gambar III.9 <i>Diagram Use Case</i> Aplikasi	34
Gambar III.10 <i>Activity Diagram Tracking Marker</i>	35
Gambar III.11 <i>ActivityDiagram</i> Cara Penggunaan.....	36
Gambar III.12 <i>Activity Diagram</i> Tentang	36
Gambar III.13 <i>Activity Diagram</i> Keluar.....	37
Gambar III.14 <i>Sequence Diagram</i> Mendeteksi <i>Marker</i>	38
Gambar III.15 <i>Sequence Diagram</i> Membaca <i>Marker</i>	39
Gambar III.16 <i>Sequence Diagram</i> Merender <i>Marker</i>	40
Gambar III.17 <i>Class Diagram</i> Aplikasi	41
Gambar III.18 Rancangan <i>Interface</i> Menu Utama	42

Gambar III.19 Rancangan <i>Interface</i> Menu Mulai	43
Gambar III.20 Rancangan <i>Interface</i> Menu Cara Penggunaan.....	44
Gambar III.21 Rancangan <i>Interface</i> Menu Tentang.....	45
Gambar III.22 <i>Marker</i> Matahari.....	46
Gambar III.23 <i>Marker</i> Merkurius.....	47
Gambar III.24 <i>Marker</i> Venus	47
Gambar III.25 <i>Marker</i> Bumi	48
Gambar III.26 <i>Marker</i> Mars	49
Gambar III.27 <i>Marker</i> Jupiter	49
Gambar III.28 <i>Marker</i> Saturnus	50
Gambar III.29 <i>Marker</i> Uranus.....	51
Gambar III.30 <i>Marker</i> Neptunus	51
Gambar III.31 <i>Splash Screen</i>	52
Gambar III.32 <i>Menu</i> Utama.....	53
Gambar III.33 Menu Mulai <i>Augmented Reality</i> Matahari.....	53
Gambar III.34 Menu Mulai <i>Augmented Reality</i> Merkurius	54
Gambar III.35 Menu Mulai <i>Augmented Reality</i> Venus.....	54
Gambar III.36 Menu Mulai <i>Augmented Reality</i> Bumi	54
Gambar III.37 Menu Mulai <i>Augmented Reality</i> Mars.....	55
Gambar III.38 Menu Mulai <i>Augmented Reality</i> Jupiter	55
Gambar III.39 Menu Mulai <i>Augmented Reality</i> Saturnus	55
Gambar III.40 Menu Mulai <i>Augmented Reality</i> Uranus.....	55
Gambar III.41 Menu Mulai <i>Augmented Reality</i> Neptunus.....	56

Gambar III.42 Menu Cara Penggunaan.....	56
Gambar III.43 Menu info.....	57
Gambar III.44 <i>Flowchart</i> Diagram.....	58
Gambar III.45 <i>Diagram</i> Alir	59

DAFTAR TABEL

	Halaman
Tabel III.1 Pengujian Black Box	60

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Seiring dengan perkembangan zaman, ilmu pengetahuan semakin berkembang dan semakin dipelajari. Ilmu pengetahuan biasanya dirangkum dalam sebuah buku yang sering disebut buku ensiklopedia. Dalam penjelasan di buku hanya berisikan teori dan gambar dua dimensi. Hal ini menimbulkan rasa malas bagi masyarakat mempelajari tentang sistem tata surya. Hampir semua elemen dari anak-anak sampai dewasa lebih tertarik pada teknologi komputer. Inilah yang menyebabkan banyak lembaga atau orang yang mencoba membuat sistem pembelajaran ilmu pengetahuan berbasis teknologi komputer.

Menurut Mustika dkk, (2015:278), “*Augmented reality* (AR) merupakan cara alami untuk mengeksplorasi objek 3D dan data, AR merupakan suatu konsep perpaduan antara *virtual reality* dan *world reality*. Sehingga objek-objek virtual 2 dimensi (2D) atau 3 dimensi (3D) seolah-olah terlihat nyata dan menyatu dengan dunia nyata. Pada teknologi AR, pengguna dapat melihat dunia nyata yang ada di sekelilingnya dengan penambahan objek virtual yang dihasilkan oleh komputer“. *Augmented reality* (AR) sangat berpotensi dalam menarik, menginspirasi, dan memotivasi untuk melakukan pengontrolan dari berbagai perspektif berbeda, yang sebelumnya tidak menjadi bahan pertimbangan dalam dunia pendidikan. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktivitas melalui perangkat-

perangkat input tertentu. Pada teknologi AR, pengguna dapat melihat dunia nyata yang ada di sekelilingnya dengan penambahan obyek virtual yang dihasilkan oleh komputer.

Penggabungan antara buku dengan teknologi AR menciptakan media baru buku berbasis AR. Agar dapat bekerja dengan sempurna media AR secara garis besar mempunyai dua komponen utama, yaitu buku dilengkapi dengan marker pada setiap halaman, yang kedua peralatan untuk menangkap marker dan menampilkan hasilnya.

Tata surya merupakan sekumpulan benda langit yang terdiri atas sebuah bintang yang disebut dengan matahari dan semua objek yang terikat oleh gravitasinya. Pembelajaran yang memanfaatkan AR merupakan sesuatu yang baru dalam masyarakat. Dengan penggunaan teknologi AR ini sebagai sarana edukasi berbasis Android yang sudah ditambah visualisasi yang menarik,interaktif dan mudah dipahami. Penulis tertarik untuk mengangkat skripsi dengan judul "**Penerapan Teknologi *Augmented Reality* Pada Pengenalan Tata Surya Berbasis Android**".

1.2. Perumusan Masalah

Berdasarkan tinjauan latar belakang diatas dapat diidentifikasi permasalahan sebagai berikut :

1. bagaimana membangun aplikasi interaktif tata surya menggunakan teknologi *augmented reality* dibuat secara detail dan mudah digunakan.
2. Apakah penggunaan teknologi *augmented reality* dapat meningkatkan daya tarik untuk mempelajari sistem tata surya.

1.3 Maksud dan Tujuan

Maksud dari penulisan skripsi ini adalah:

1. Dengan adanya aplikasi ini dapat digunakan sebagai media interaktif dan menarik untuk pembelajaran umum.
2. Pemanfaatan teknologi dibidang edukasi merupakan suatu upaya yang positif untuk mengenalkan masyarakat pada perkembangan teknologi dengan tujuan agar tidak canggung terhadap teknologi saat berhadapan dengan perkembangan teknologi.
3. Aplikasi ini diharapkan dapat menambah pengalaman dan pelajaran serta acuan untuk dapat membangun aplikasi selanjutnya.

Adapun tujuan dari penulisan skripsi ini adalah sebagai salah satu syarat untuk kelulusan pada program Strata 1(S1) Program Studi Teknik Informatika pada Sekolah Tinggi Manajemen Informatika dan Komputer(STMIK) Nusa Mandiri.

1.4 . Metode Penelitian

Metode penelitian yang akan dilakukan oleh penulis yaitu tahapan-tahapan yang akan dilakukan oleh penulis mulai dari perumusan masalah sampai pada kesimpulan yang membentuk suatu alur sistematis. Metode ini dijadikan penulis sebagai pedoman penelitian penulisan ini guna untuk mencapai hasil yang dicapai, tidak menyimpang dari permasalahan tujuan yang telah ditentukan sebelumnya.

1.4.1. Metode Pengumpulan Data

Metode yang digunakan dalam pengumpulan data pada skripsi ini adalah :

1. Observasi

Penulis mengumpulkan data-data dengan mencari materi-materi tentang system tata surya dengan teknologi *augmented reality* berbasis android dan mengkaji lebih jauh buku-buku tentang pembuatan *augmented reality* berbasis android.

2. Studi Pustaka

Penulis melakukan metode studi pustaka sebagai pendukung untuk mendapatkan berbagai macam informasi dengan cara mencari buku-buku, Jurnal-Jurnal yang berhubungan dengan pembahasan yang diperlukan untuk dijadikan sebagai referensi dalam penulisan skripsi ini.

1.4.2. Metode Pengembangan Aplikasi

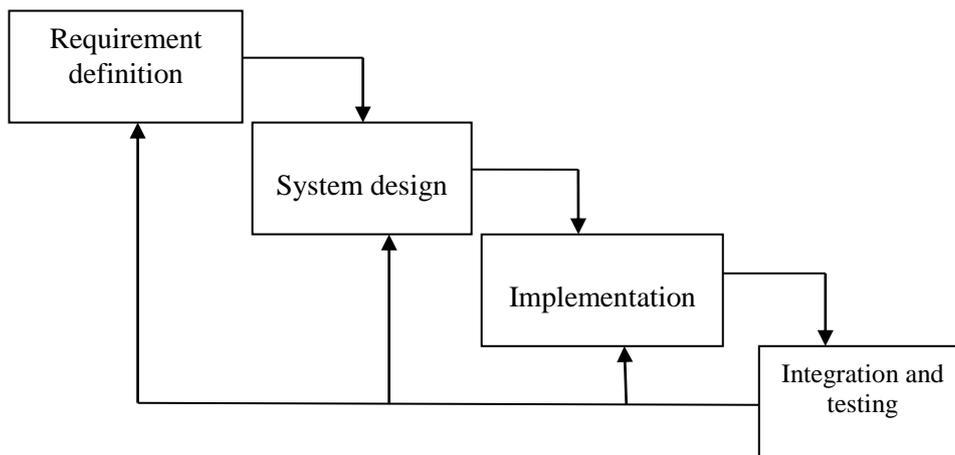
Metode pengembangan aplikasi yang digunakan penulis adalah metode SDLC (*System Development Life Cycle*) model *Waterfall*, sebuah metode pengembangan software yang bersifat sekuensial.

1. Metode SDLC

Menurut Shalahuddin dan Rosa (2014:26) SDLC atau *Software Development Life Cycle* adalah proses mengembangkan atau mengubah suatu system perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan system-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik). Seperti halnya proses metamorfosis pada kupu-kupu, untuk kupu-kupu yang indah maka dibutuhkan beberapa tahap untuk dilalui, sama halnya dengan membuat perangkat lunak, memilih daur tahapan yang dilalui agar menghasilkan perangkat lunak yang berkualitas.

2. Model *Waterfall*

Model SDLC air terjun (*waterfall*) kadang dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak (Pressman, 2012:46). Berikut adalah gambar model air terjun:



Gambar I.1. Ilustrasi model *waterfall*

Sumber : Pressman (2012:46)

a. *Requirement*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung.

b. *System Desain*

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk terstruktur data, arsitektur perangkat

lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

d. *Integration & Testing*

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

1.5. Ruang Lingkup

Untuk mempermudah, lebih terarah, dan berjalan dengan lebih baik maka perlu suatu batasan masalah. Adapun ruang lingkup permasalahan yang dibahas dalam penulisan skripsi ini, antara lain :

1. Aplikasi ini nantinya akan dapat dijalankan dalam peranti *gadget* dengan *operating system* android, yang akan berisi *augmented reality* dari planet-planet sistem tata surya.
2. Dalam aplikasi ini akan dicoba menggunakan *marker* yang berbeda dengan aplikasi AR yang lain yaitu dengan gambar dari objek 3D-nya.

3. Dalam studi ini tidak mengutamakan *modeling* objek 3D-nya karena dalam permasalahan ini hanya mengutamakan penggunaan *augmented reality* berbasis android.
4. Menggunakan *software Unity Editor* untuk membuat *augmented reality* ditambah dengan *plug in* Vuforia SDK.
5. Pengujian aplikasi ini menggunakan *smartphone* dengan *operating system* android versi 6.0.1 *marshmallow* dengan resolusi layar 5 inch.

BAB II

LANDASAN TEORI

2.1. Tinjauan Jurnal

Menurut Budihartanti dan Adami, (2016:1), “*Augmented reality* adalah jenis teknologi interaktif menggabungkan benda nyata dan virtual yang akan menghasilkan objek 3D yang akan ditampilkan pada layar. *Augmented reality* yang telah diaplikasikan memiliki cara kerja berdasarkan deteksi citra atau gambar dan biasa disebut *marker*, dengan menggunakan kamera *smartphone* kemudian mendeteksi *marker* yang telah dicetak”

Tata surya adalah sekumpulan langit, yaitu Matahari dan planet-planetnya yaitu merkurius, venus, bumi, mars, Jupiter, saturnus, uranus, dan neptunus beserta 165 buah satelit planet yang sudah diketahui sampai sekarang, serta objek-objek tata surya lainnya seperti asteroid, meteorid, planetoid, komet, debu antar-planet, dan sebuah daerah yang dinamakan sabuk kuiper (Admiranto, 2017:3).

Dalam dunia pendidikan telah memiliki media pembelajaran yang interaktif dalam materi system tata surya tetapi masih berupa aplikasi yang berbasis dua dimensi. Salah satu penerapan AR di dunia pendidikan adalah untuk membantu mempelajari system tata surya sebagai sarana edukasi berbasis *Augmented reality* dengan visualisasi yang menarik, interaktif dan mudah dipahami.

Menurut Angga dan Wahyu, (2014: 2), “Dengan perkembangan teknologi saat ini, terutama dibidang teknologi AR, maka perlu diimplementasikan dalam dunia pembelajaran terutama pembelajaran tata surya menggunakan teknologi AR. Agar dapat membantu masyarakat mendapatkan informasi tentang tata surya”

2.2. Konsep Dasar Program

Program secara umum dapat diartikan sebagai ekspresi pernyataan atau kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman, sehingga dapat dieksekusi oleh program.

Aplikasi merupakan penerapan, penyimpanan data, permasalahan ke dalam suatu sarana atau media yang digunakan untuk menerapkan atau mengimplementasikan suatu hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari data, permasalahan atau pekerjaan. Jadi, dalam hal ini hanya berbentuk tampilan data yang berubah, sedangkan isi yang termuat dalam data tersebut tidak mengalami perubahan.

2.2.1. *Augmented Reality*

Augmented reality (AR) merupakan sebuah teknik untuk menggabungkan dunia nyata dan dunia maya, teknik ini memungkinkan sebuah objek di dunia maya ditampilkan dengan objek lain di dunia nyata secara bersamaan (Roedavan, 2016:302). Dengan kata lain, ada sisipan benda maya pada keadaan nyata yang dapat dilihat dari sebuah layar dengan inputan perekaman sebuah kamera. Pengguna akan merasa melihat benda 2d/3d tersebut dalam layar dengan sebuah *marker* (*marker*, *marker-texture* (*surface*), *face detection/recognition*, *motion detection*, dan *GPS & Digital Compass*.) sebagai titik acuan fokus kamera. Tidak seperti realitas maya (*virtual reality*) yang sepenuhnya menggantikan kenyataan, AR sekedar menambahkan atau melengkapi saja.

Dengan bantuan teknologi *Augmented Reality*, lingkungan nyata disekitar kita akan dapat berinteraksi dalam bentuk digital (virtual). Informasi-informasi tentang objekdan lingkungan disekitar kita dapat ditambahkan kedalam system *Augmented reality* yang kemudian informasi tersebut ditampilkan diatas *layer* dunia nyata secara *real-time* seolah-olah informasi tersebut adalah nyata.

Dalam cabang bidang teknologi AR belum terlalu lama, namun memiliki perkembangan yang sangat cepat. Perkembangan *Augmented reality* pada industri *mobile phone* juga mempunyai perkembangan yang paling cepat.

2.2.2. Sejarah *Augmented Reality*

Sejarah tentang *augmented reality* dimulai dari tahun 1957-1962, ketika seorang penemu yang bernama Morton Heilig, seorang sinematografer, 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 Krueger menemukan *video place* yang memungkinkan pengguna, dapat berinteraksi dengan objek virtual untuk pertama kalinya. Tahun 1989, Jaron Lanier, memperkenalkan Virtual Reality dan menciptakan bisnis komersial pertama kali 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 sistem AR, yang disebut *Virtual Fixtures*, yang digunakan di Angkatan Udara AS Armstrong Labs, dan menunjukkan manfaatnya pada manusia, dan pada tahun 1992 juga, Steven Feiner, Blair MacIntyre dan dorée

Pada tahun 1999, Hirokazu Kato, mengembangkan ArToolkit di HITLab dan didemonstrasikan di SIGGRAPH, pada tahun 2000, Bruce.H.Thomas, mengembangkan ARQuake, sebuah Mobile Game AR yang ditunjukkan di *International Symposium on Wearable Computers*. Pada tahun 2008, Wikitude *AR Travel Guide*, memperkenalkan Android G1 Telephone yang berteknologi AR, tahun 2009, Saqoosha memperkenalkan FLARToolkit yang merupakan perkembangan dari ArToolkit. FLARToolkit memungkinkan kita memasang teknologi AR di sebuah website, karena output yang dihasilkan FLARToolkit berbentuk Flash. Ditahun yang sama, Wikitude Drive meluncurkan sistem navigasi berteknologi AR di Platform Android. Tahun 2010, Acrossair menggunakan teknologi AR pada I-Phone 3GS.

2.2.3. Sejarah OS Android

Menurut Sadeli (2014:2), Android merupakan sistem operasi berbasis Linux yang digunakan untuk perangkat mobile (*smartphone*) atau perangkat tablet (PDA). Sifat platform android yang terbuka bagi para pengembang untuk mengembangkan aplikasi buatan sendiri membuat android menjadi sistem operasi mobile yang populer hingga saat ini. Android didirikan pada tahun 2003 oleh Andy Rubin, Nick Sears, Rich Miner dan Chris White sebelum di pinang oleh Google pada bulan juli 2005, dengan harga estimasi USD 50 juta. Google sebagai perusahaan raksasa membeli *Android inc* beserta teknologinya. Google melanjutkan untuk mengembangkan android untuk bersaing pada pasar *smartphone*. Dalam usaha pengembangan *android*, dibentuk *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Berikut beberapa versi sistem operasi android antara lain :

1. Android 1.0 (Astro) API Level 1

Android versi ini dirilis pada tanggal 23 september 2008, versi ini disematkan untuk pertama kalinya pada ponsel *smartphone* HTC Dream.

2. Android 1.1 (Bender) API Level 2

Android versi ini dirilis pada tanggal 9 februari 2009, versi ini memiliki beberapa fungsi seperti pembaruan estetis pada aplikasi, jam alarm, voice search, pengiriman pesan dengan GMAIL dan pemberitahuan email.

3. Android 1.5 (Cupcake) API Level 3

Android versi ini dirilis pada pertengahan mei 2009, versi ini memiliki beberapa fungsi pembaruan seperti merekam dan menonton video, mengunggah video ke youtube, dukungan Bluetooth A2DP, animasi layar, serta keyboard yang dapat disesuaikan sistem.

4. Android 1.6 (Donut) API Level 4

Android versi ini dirilis pada bulan 15 september 2009, terdapat penambahan serta perbaikan dari versi sebelumnya seperti penggunaan indicator baterai, dial kontak, kamera, resolusi VWGA dan beberapa fitur lainnya.

5. Android 2.0/2.1 (Eclair) API Level 5-7

Android versi ini dirilis pada tanggal 26 oktober 2009, terdapat penambahan fitur pengoptimalan hardware, peningkatan Google Maps, perubahan UI (*User Interface*) dengan browser baru dan dukungan HTML5.

6. Android 2.2 Froyo (Froze Yoghurt) API Level 8

Android versi ini dirilis pada tanggal 20 Mei 2010, pada versi ini terdapat dukungan Adobe Flash 10.1, peningkatan kinerja lebih baik sampai dua kali lipat dari versi sebelumnya yang terintegrasi dengan V8 JavaScript engine dipakai Google Chrome, pemasangan aplikasi dalam SD Card, WiFi Hotspot portable, auto update aplikasi android market.

7. Android 2.2 (Gingerbread) API Level 9-10

Android versi ini dirilis pada tanggal 6 Desember 2010, terdapat peningkatan khusus pada aplikasi game, resolusi WXGA dan beberapa fitur lain. Pada versi ini memiliki pengguna terbanyak dibandingkan dengan versi sebelumnya.

8. Android 3.0/3.1 (Honeycomb) API Level 11-13

Android versi ini dirilis pada tanggal 22 Februari 2011. Android versi ini mendukung ukuran layar yang lebih besar. UI (*User Interface*) pada honeycomb berbeda karena didesain khusus untuk tablet. Tablet pertama yang menggunakan versi ini adalah Motorola Xoom.

9. Android 4.0 – 4.0.4 ICS (Ice Cream Sandwich) API Level 14-15

Android versi ini dirilis pada tanggal 19 Oktober 2011. *Smartphone* yang pertama kali menggunakan sistem operasi ini adalah Samsung Galaxy Nexus. Versi ini memiliki fitur baru seperti membuka kunci dengan pengenalan wajah, jaringan social dan beberapa fitur tambahan lainnya.

10. Android 4.1 – 4.3.1 (Jelly Bean) API Level 16-18

Android versi ini banyak sekali fitur penambahan seperti widget baru, photo sphere, power control lock screen gadget dan beberapa fitur lain. *Smartphone* pertama kali

yang menggunakan versi ini adalah LG Google Nexux 4. Android versi ini dirilis pada tanggal 24 juli 2013.

11. Android 4.4 – 4.4.4 (KitKat) API Level 19

Android versi ini dirilis pada tanggal 32 oktober 2013. Beberapa pembaruan seperti antar muka dengan bar status, navigasi transparan pada layar depan, sensor batching, step detector dan counter API, peningkatan audio, aksesibilitas API dan beberapa fitur lainnya.

2.2.4. Android SDK (*Software Development Kit*)

Menurut Irawan (2012:24), Android SDK merupakan perangkat lunak untuk membuat dan mengembangkan aplikasi android. Didalamnya terdapat *library*, android emulator, serta perangkat lunak lainnya yang dibutuhkan untuk membuat sebuah aplikasia android. Pengembangan aplikasi android umumnya dilakukan dengan bahasa pemrograman java, meski menggunakan bahasa lain seperti bahasa C juga bisa dilakukan menggunakan Android Native Development Kit. Android SDK akan selalu di-update menyesuaikan dengan versi system operasi android yang sudah resmi dirilis.

Beberapa fitur-fitur Android yang paling penting adalah:

- A. *Framework* aplikasi yang mendukung penggantian komponen dan reusable.
- B. Mesin Virtual Dalvik dioptimalkan untuk perangkat mobile.
- C. Integrated browser berdasarkan engine open source WebKit.
- D. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware).
- E. SQLite untuk penyimpanan data (*database*).

- F. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephony (tergantung hardware).
- G. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
- H. Kamera, GPS, kompas, dan accelerometer (tergantung hardware).
- I. Lingkungan development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

2.2.5. Vuforia SDK (Software Development Kit)

Vuforia Software Development Kit (SDK) dikembangkan oleh Qualcomm memungkinkan seorang developer untuk membuat aplikasi berbasis teknologi *augmented reality*. Teknologi *Augmented Reality* (AR) telah banyak diimplementasikan dalam berbagai kebutuhan untuk media pembelajaran interaktif, hiburan, berita, informasi, Untuk mengembangkan aplikasi maupun *game* dengan teknologi AR *vuforia* dapat digunakan sebagai pendukung adanya *Augmented reality* pada Android untuk program Unity 3D. *Vuforia* menganalisa gambar dengan menggunakan pendeteksi marker dan menghasilkan informasi 3D dari marker yang sudah dideteksi via API (*application programming interface*).

Vuforia mendukung berbagai jenis target 2D dan 3D termasuk target gambar '*markles*', 3D multi target konfigurasi, dan bentuk *Marker Frame*. Fitur tambahan *Deteksi Oklusi Local* menggunakan 'Tombol Virtual', *runtime* pemilihan gambar target, dan kemampuan untuk membuat dan mengkonfigurasi ulang set pemrograman pada saat *runtime*.

2.2.6. C#

Menurut Filus (2017) C# (dibaca “See-Sharp”) dikenal dengan bahasa pemrograman sederhana yang digunakan untuk tujuan umum, dalam artian bahasa pemrograman ini dapat digunakan untuk berbagai fungsi misalnya untuk pemrograman *server-side* pada *website*, membangun aplikasi desktop ataupun mobile, pemrograman game dan sebagainya. Selain itu, C# juga bahasa pemrograman yang berorientasi objek, jadi C# juga mengusung konsep objek seperti *inheritance*, *class*, *polymorphism* dan *encapsulation*.

Dalam prakteknya C# sangat bergantung dengan framework yang disebut .NET Framework, framework inilah yang nanti digunakan untuk mencompile dan menjalankan kode C#. C# dikembangkan oleh Microsoft dengan merekrut Anders Helsberg. Tujuan dibangunnya C# adalah sebagai bahasa pemrograman utama dalam lingkungan .NET Framework

2.2.7. *Unified Modeling Language (UML)*

UML menurut Shalahuddin dan Rosa (2014:133) adalah Salah standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

Unified Modelling Language (UML) menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan UML adalah :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktik-praktik terbaik yang terdapat dalam pemodelan. UML menyediakan beberapa notasi dan artifact standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. Artifact didalam UML didefinisikan sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat.

2.2.8. Diagram UML

1. *Use Case Diagram*

Menurut Shalahuddin dan Rosa (2014:155) *use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) system informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang akan ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

2. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat pada perangkat lunak,

sehingga tidaklah ada gunanya lagi sebuah perancangan karena apa yang dirancang dan hasil jadinya tidak sesuai.

3. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah system atau proses bisnis atau menu yang ada pada perangkat lunak. atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

4. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu, Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case*.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah di definisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuan yang harus dibuat juga semakin banyak.

2.3. Metode Algoritma

Metode algoritma yang digunakan penulis adalah algoritma *fast corner detection*.

2.3.1. *Algoritma Fast Corner Detection*

Menurut setiawan, dkk (2016:29) *FAST (Feature Form Accelerated segment Test)* adalah suatu algoritma yang dikembangkan oleh Edward Rosten, Reid Porter, dan Tom Drummond. *Fast corner detection* ini dibuat dengan tujuan konsekuensi menurunkan tingkat akurasi pendeteksian sudut. *FAST corner detection* dimulai dengan menentukan suatu titik p pada koordinat (x_p, y_p) pada citra dan membandingkan intensitas titik p dengan 4 titik disekitarnya. Titik pertama terletak pada koordinat (x, y_p-3) , titik kedua

terletak pada koordinat (x_p+3, y) , titik ketiga terletak pada koordinat (x, y_p+3) , dan titik keempat terletak pada koordinat (x_p-3, y) .

Jika nilai intensitas titik p bernilai lebih besar atau lebih kecil dari pada intensitas sedikitnya tiga titik disekitarnya ditambah dengan suatu intensitas batas ambang (*Threshold*), maka dapat dikatakan bahwa titik p adalah suatu sudut. Setelah itu titik p akan digeser ke posisi (x_p+1, y_p) dan melakukan intensitas keempat titik disekitarnya lagi. Iterasi ini terus dilakukan sampai semua titik pada citra sudah dibandingkan.

Vuforia menggunakan *algoritma FAST corner detection* untuk mendefinisikan seberapa baik gambar dapat dideteksi dan dilacak menggunakan *Vuforia SDK*. Peringkat ini ditampilkan dalam *Target Manager* dan kembali untuk setiap target upload melalui web API. *Rating Augmentable* dapat berkisar dari 0 sampai 5 untuk setiap gambar yang diberikan. Semakin tinggi *rating augmentable* dari target gambar, semakin kuat kemampuan deteksi dan pelacakan yang dikandungnya. Sebuah *rating* dari nol menunjukkan bahwa target tidak dilacak sama sekali oleh sistem *Augmented Reality*, sedangkan *rating* bintang 5 menunjukkan bahwa sebuah gambar dengan mudah dilacak oleh sistem *Augmented Reality*.

2.4. Pengujian Aplikasi

Pengujian aplikasi yang digunakan penulis adalah *Black-Box Testing* (pengujian kotak hitam) dan *White-Box Testing* (pengujian kotak putih).

1. *Black-Box Testing* (Pengujian Kotak Hitam)

Menurut Shalahuddin dan Rosa (2014:275), *black-box testing* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah.

2. *White-Box Testing* (Pengujian Kotak Putih)

Menurut Shalahuddin dan Rosa (2014:276), *white-box testing* adalah menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian kotak putih dilakukan dengan memeriksa logik dari kode program. Pembuatan kasus uji bisa mengikuti standar pengujian dari standar program yang seharusnya.

2.5. Peralatan Pendukung

1. Unity Editor

Unity editor digunakan untuk mengembangkan game multi platform, unity itu bagus dan penuh perpaduan dengan aplikasi yang professional. Editor ini dibuat setelah ribuan jam yang telah di habiskan untuk membuatnya menjadi nomor satu dalam urutan *rangking* teratas untuk *editor game*. Grafis pada unity dibuat dengan grafis tingkat tinggi untuk OpenGL dan direct. Unity mendukung semua format file, terutama format umum seperti semua format dari *art applications*. Unity cocok dengan versi 64-bit dan dapat beroperasi pada Mac OS x, Windows dan dapat menghasilkan game untuk Mac, Windows, Wii, iPhone, Ipad dan Android.



Gambar II.1. Unity Editor

(Sumber : <https://unity3d.com/unity>)

2. CorelDraw

CorelDraw dikembangkan oleh Corel, sebuah perusahaan perangkat lunak yang bermarkas di Ottawa, Kanada. Corel draw memiliki kegunaan untuk mengolah gambar, oleh karena itu banyak digunakan pada pekerjaan dalam bidang oublikasi atau percetakan ataupun di bidang lain yang membutuhkan proses visualisasi.



Gambar II.2 *CorelDRAW*

(sumber : https://community.coreldraw.com/talk/coreldraw_graphics_suite_x4)

BAB III

ANALISA DAN PERANCANGAN SOFTWARE

3.1. Analisa Kebutuhan Software

Sistem yang dibuat merupakan informasi planet-planet tata surya berbasis *augmented reality* yaitu pembelajaran atau penyalur informasi mengenai sistem tata surya. Aplikasi yang dibuat seolah-olah pengguna dapat berinteraksi langsung dengan objek-objek tiga dimensi yang tampil. Pada aplikasi ini juga terdapat teks yang berisi penjelasan mengenai sistem tata surya. *Marker* yang digunakan dalam aplikasi ini berjenis *markerless*, yaitu *marker* dalam wujud gambar atau benda.

3.1.1 Identifikasi Masalah

Pada penulisan skripsi ini permasalahan yang akan diteliti dalam pembuatan aplikasi *augmented reality* pengenalan tata surya adalah:

1. Bagaimana mengimplementasikan teknologi *Markerless Augmented Reality* untuk media pembelajaran materi mengenal sistem tata surya untuk siswa SD?
2. Bagaimana meningkatkan prestasi belajar mata pelajaran ilmu pengetahuan alam khususnya materi mengenal sistem tata surya?

3.1.2. Analisa Kebutuhan Aplikasi

Tahap analisa kebutuhan mencakup *hardware*, *software*, aplikasi dan *output* yang digunakan adalah sebagai berikut:

- 1) Vendor : ASUSTek COMPUTER INC.
- 2) Tipe : X45U
- 3) Prosesor : AMD E-450 APU with Radeon(tm) HD Graphics

- 4) Display Adapter : AMD Radeon HD 6320 Graphi
- 5) RAM : 2 GB
- 6) Hardisk : 500 GB

Perangkat *Gadget* atau *Smartphone* yang digunakan penulis mempunyai klasifikasi sebagai berikut:

- 1) Vendor : Xiaomi
- 2) Tipe : Redmi 4A
- 3) Processor : Quad-core Max 1.40GH
- 4) RAM : 2 GB
- 5) Memory Internal : 16 GB
- 6) Resolution : 5 Inch
- 7) Operating Sistem : 6.0.1 Marshmallow

3.2. Desain

Sebelum aplikasi Android ini diimplementasikan dalam bentuk *package.apk*, maka perlu dirancang terlebih dahulu. Tahap perancangan bertujuan untuk memenuhi kebutuhan pengguna dan memberikan gambaran yang jelas mengenai aplikasi yang kita buat. Keseluruhan dari perancangan ini akan di implementasikan dalam *Gadget* berbasis *android* dengan menggunakan perangkat lunak *Unity Editor*.

3.2.1 Rancangan Algoritma

1. Coding List Algoritma FAST Corner Detection

```
#include <stdlib.h>

#include "fast.h"
xy* fast9_detect_nonmax(const byte* im, int xsize, int ysize, int
stride, int b, int* ret_num_corners)
{
    xy* corners;
    int num_corners;
    int* scores;
    xy* nonmax;
    corners = fast9_detect(im, xsize, ysize, stride, b,
&num_corners);
    scores = fast9_score(im, stride, corners, num_corners, b);
    nonmax = nonmax_suppression(corners, scores,
num_corners, ret_num_corners);
    free(corners);
    free(scores);
    return nonmax;
}
xy* fast10_detect_nonmax(const byte* im, int xsize, int ysize, int
stride, int b, int* ret_num_corners)
{
    xy* corners;
    int num_corners;
    int* scores;
    xy* nonmax;
    corners = fast10_detect(im, xsize, ysize, stride, b,
&num_corners);
    scores = fast10_score(im, stride, corners, num_corners,
b);
    nonmax = nonmax_suppression(corners, scores,
num_corners, ret_num_corners);
    free(corners);
    free(scores);
    return nonmax;
}
xy* fast11_detect_nonmax(const byte* im, int xsize, int ysize, int
stride, int b, int* ret_num_corners)
{
```

```

        xy* corners;
        int num_corners;
        int* scores;
        xy* nonmax;
        corners = fast11_detect(im, xsize, ysize, stride, b,
&num_corners);
        scores = fast11_score(im, stride, corners, num_corners,
b);
        nonmax = nonmax_suppression(corners, scores,
num_corners, ret_num_corners);
        free(corners);
        free(scores);
        return nonmax;
}
xy* fast12_detect_nonmax(const byte* im, int xsize, int ysize, int
stride, int b, int* ret_num_corners)
{
        xy* corners;
        int num_corners;
        int* scores;
        xy* nonmax;
        corners = fast12_detect(im, xsize, ysize, stride, b,
&num_corners);
        scores = fast12_score(im, stride, corners, num_corners,
b);
        nonmax = nonmax_suppression(corners, scores,
num_corners, ret_num_corners);
        free(corners);
        free(scores);
        return nonmax;
}

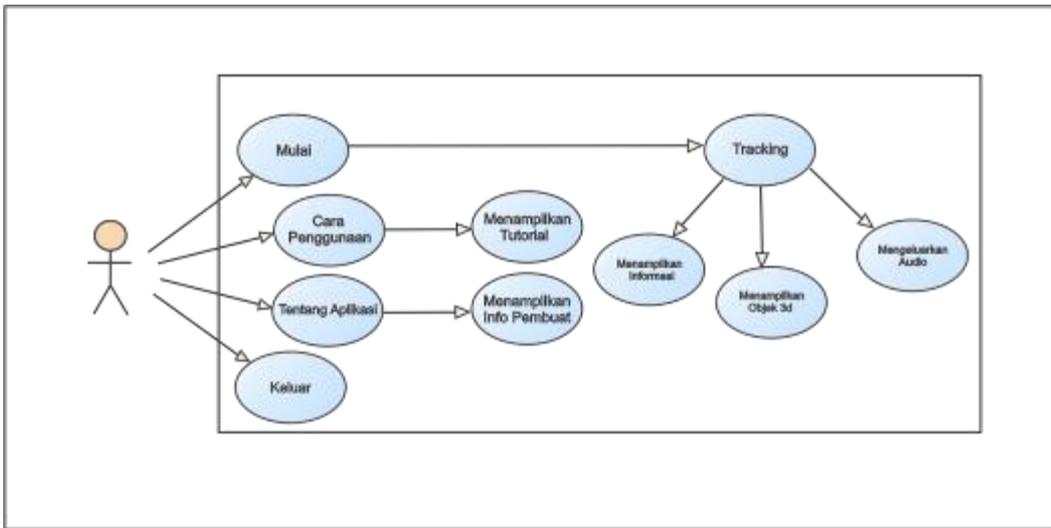
```

3.2.2. *Software Architecture*

Perancangan aplikasi ini menggunakan UML(*Unified Modeling Language*) sebagai pemodelan sistem, adapun urutan perancangan untuk diagram-diagram yang terdapat di UML pada aplikasi ini adalah:

1. *Diagram Use Case*

Use Case Diagram mendeskripsikan hubungan-hubungan yang terjadi antara aktor dengan aktivitas yang terdapat pada sistem. Aktor dalam sistem adalah pengguna, sedangkan sistem adalah aplikasi *augmented reality* tata surya. Adapun *use case diagram* dalam aplikasi ini sebagai berikut:



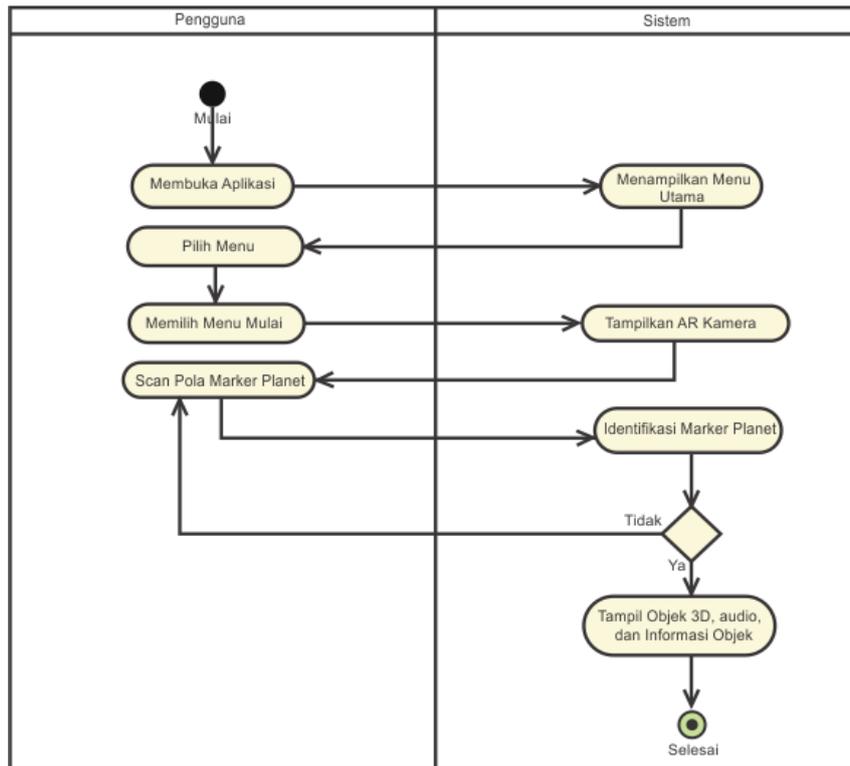
Gambar III.9
Diagram Use Case Aplikasi

2. Activity Diagram

Activity diagram memodelkan aliran aktifitas yang terjadi pada *use case*. Adapun *activity diagram* dari penjabaran masing-masing *use case diagram* dapat dijelaskan pada *activity diagram*.

a. Activity Diagram Tracking Marker

Activity diagram tracking marker menjelaskan aliran aktifitas kerja aktor menjalankan aplikasi AR untuk mentracking marker yang di arahkan oleh kamera AR.

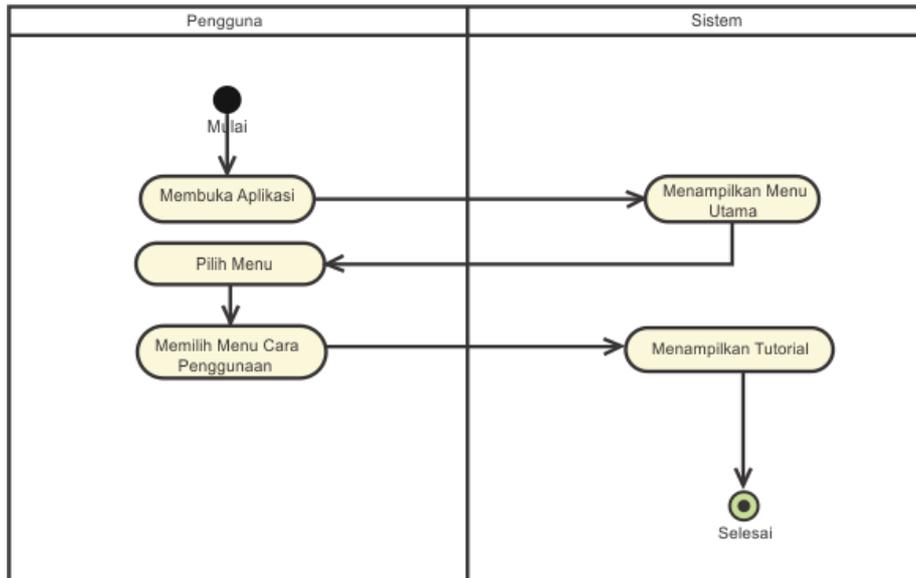


Gambar III.10

Activity Diagram Tracking Marker

b. *Activity Diagram* Cara Penggunaan

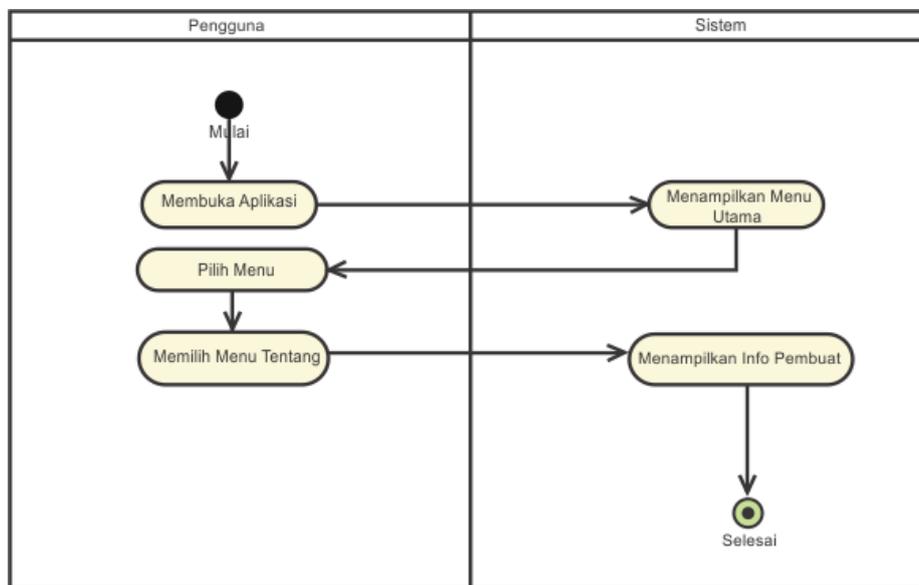
Activity diagram cara penggunaan menunjukkan aktifitas dari menu pilihan untuk menampilkan cara penggunaan aplikasi AR tata surya.



Gambar III.11

Activity Diagram Cara Penggunaan

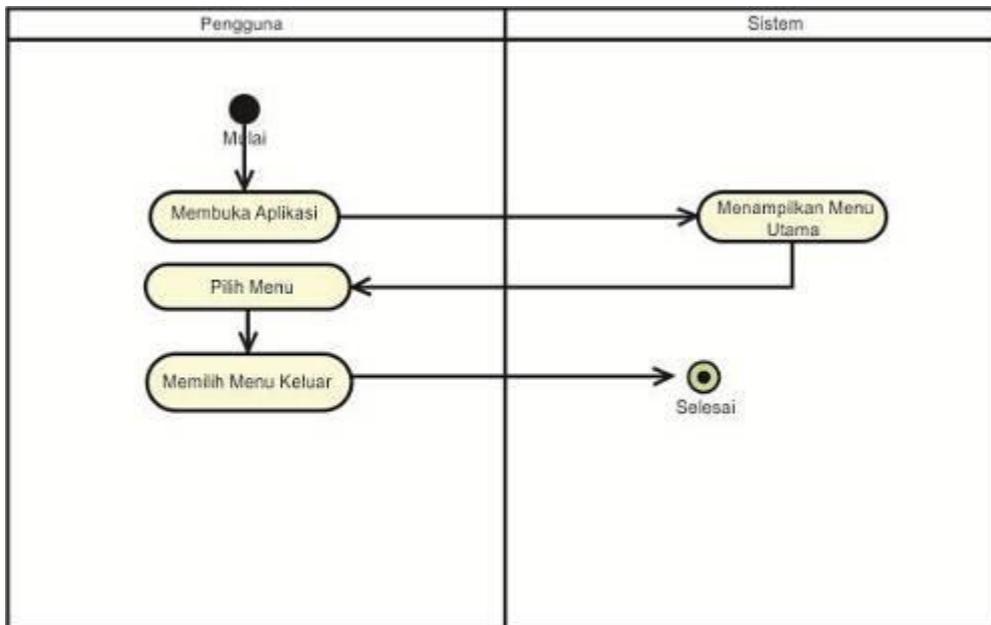
c. *Activity Diagram Tentang*



Gambar III.12

Activity Diagram Tentang

d. *Activity Diagram Keluar*



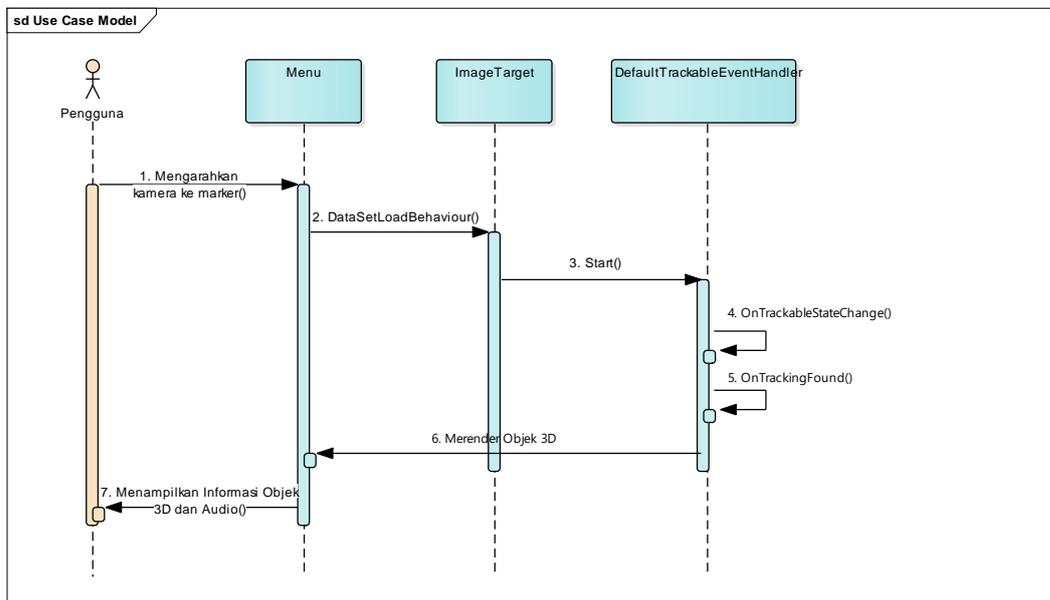
Gambar III.13

Activity Diagram Keluar

3. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek. Interaksi antar objek berupa pengiriman data antar objek dalam urutan waktu. Adapun *sequence diagram* aplikasi *augmented reality* tata surya adalah sebagai berikut:

a. *Sequence Diagram Mendeteksi Marker*

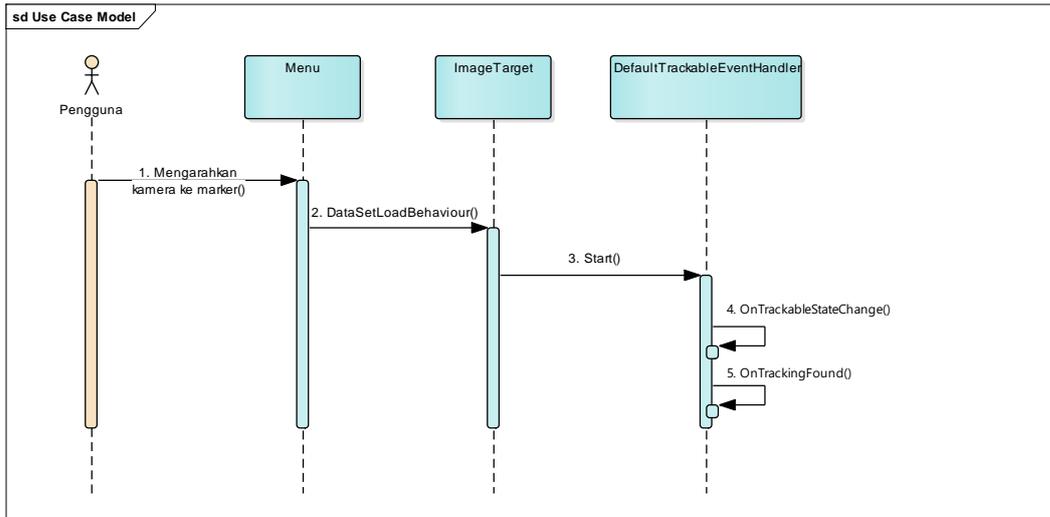


Gambar III.14

Sequence Diagram Mendeteksi Marker

Pada *sequence diagram* diatas pengguna mengarahkan kamera AR ke *marker* buku planet tata surya. Sistem akan mengakses image target pada `DataSetLoadBehaviour` untuk mencocokkan image target dengan hasil tangkapan kamera AR, setelah itu akan memunculkan objek 3D, informasi, dan audio.

b. *Sequence Diagram Membaca marker*

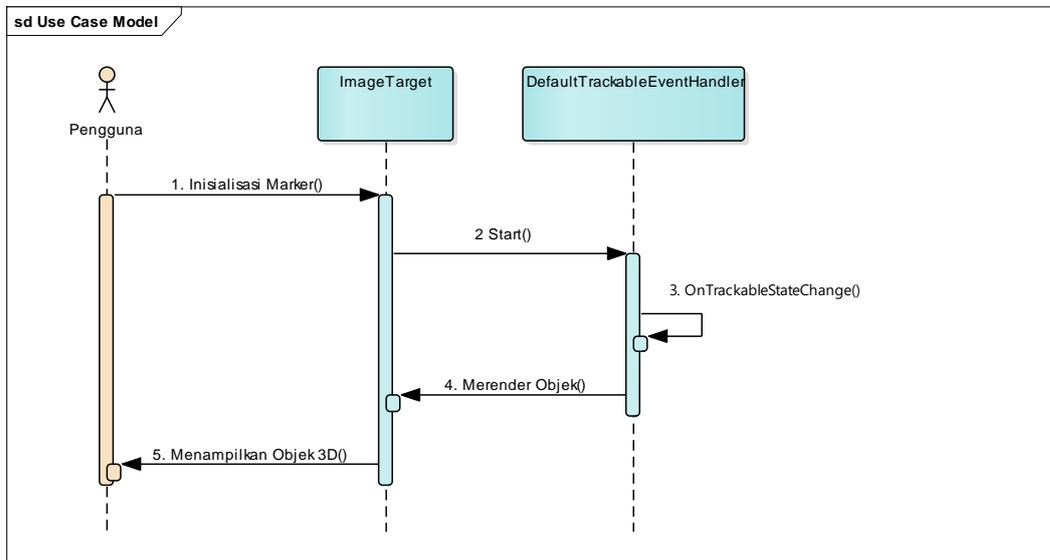


Gambar III.15

Sequence Diagram Membaca Marker

Pada *sequence diagram* diatas pengguna mengarahkan kamera AR ke marker buku planet tata surya. Hasil tangkapan kamera AR akan dicocokkan dengan *image target*.

c. *Sequence Diagram Merender Marker*



Gambar III.16

Sequence Diagram Merender Objek

4. *Class Diagram*

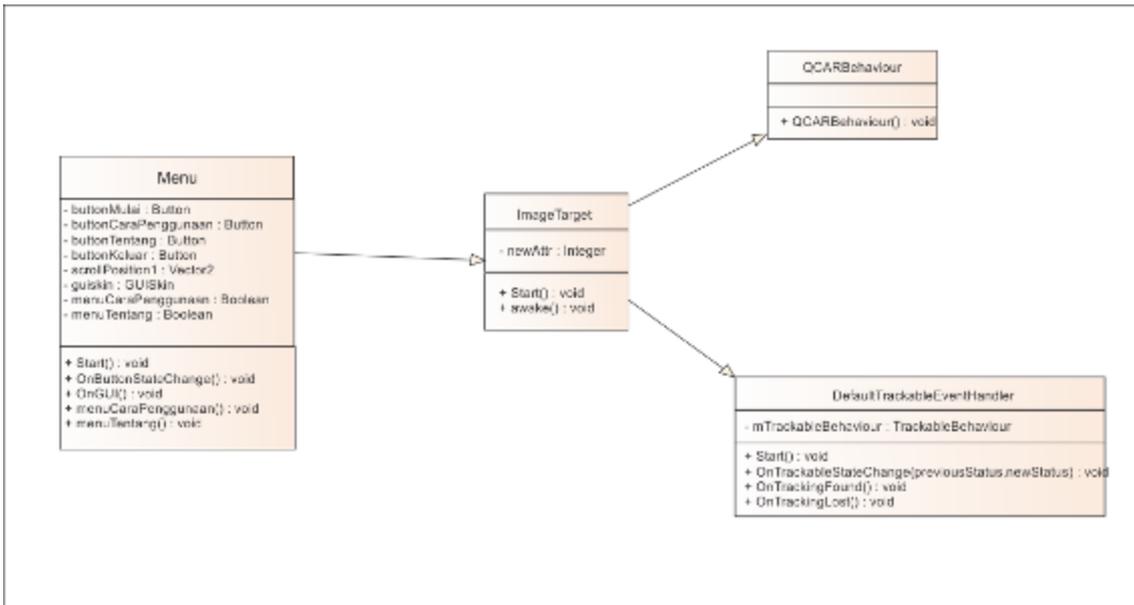
Class diagram merupakan inti dari pengembangan dan desain berorientasi objek.

Class diagram menggambarkan hubungan antar objek dan terstruktur system.

Sistem tersebut akan menampilkan kelas, atribut, dan hubungan antar kelas

ketika suatu sistem telah selesai membuat diagram. Adapaun *class diagram* pada

aplikasi ini sebagai berikut:



Gambar III.17

Class Diagram Aplikasi

3.2.3. Perancangan Antarmuka (*Interface Design*)

Antarmuka pengguna (*user interface*) merupakan mekanisme komunikasi antara pengguna (*user*). Tujuan utamanya adalah mengkomunikasikan fitur-fitur yang terdapat pada sistem agar *user* mengerti dan dapat menggunakan aplikasi dengan baik.

User interface pada aplikasi diantaranya *activity* dan *user interface* yang terdiri dari komponen. Adapun perancangan *interface* pada aplikasi ini adalah sebagai berikut:

1. Perancangan tampilan menu utama

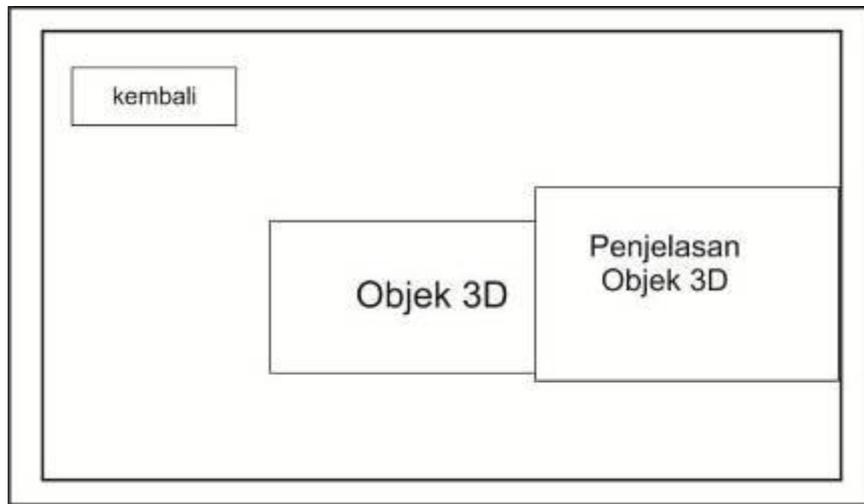
Pada tampilan menu utama di desain sederhana yang didalamnya terdapat *background* dengan *format file .jpg* dan *.png* teks judul dengan tulisan “*Augmented Reality Tata Surya*” dan terdapat empat *button* yang terdiri dari Mulai, Cara Penggunaan, Tentang, dan Keluar.



Gambar III.18
Rancangan *Interface* Menu Utama

Pada perancangan halaman menu utama akan ditampilkan *background* berupa gambar system tata surya diikuti judul aplikasi dibagian atas dan tombol-tombol menu yang digunakan untuk menjalankan aplikasi.

2. Perancangan tampilan menu mulai



Gambar III.19

Rancangan *Interface* Menu Mulai

Pada perancangan menu mulai akan ditampilkan *capture* kamera yang menampilkan objek tiga dimensi sesuai dengan *marker* yang ditujukan ke kamera. Terdapat tombol kembali ke halaman menu utama, gambar menampilkan penjelasan objek tiga dimensi, dan saat objek 3 dimensi tampil akan ada suara *dubber* yang menjelaskan masing-masing objek yang sesuai dengan *marker*.

3. Perancangan tampilan menu cara penggunaan



Gambar III. 20
Rancangan *Interface* Menu Cara Penggunaan

Pada perancangan halaman menu cara penggunaan akan ditampilkan penjelasan singkat mengenai bagaimana cara penggunaan aplikasi sistem tata surya menggunakan *augmented reality*. Terdapat tombol kembali yang akan digunakan untuk kembali ke halaman sebelumnya.

4. Perancangan tampilan menu tentang



Gambar III.21

Rancangan *Interface* Menu Tentang

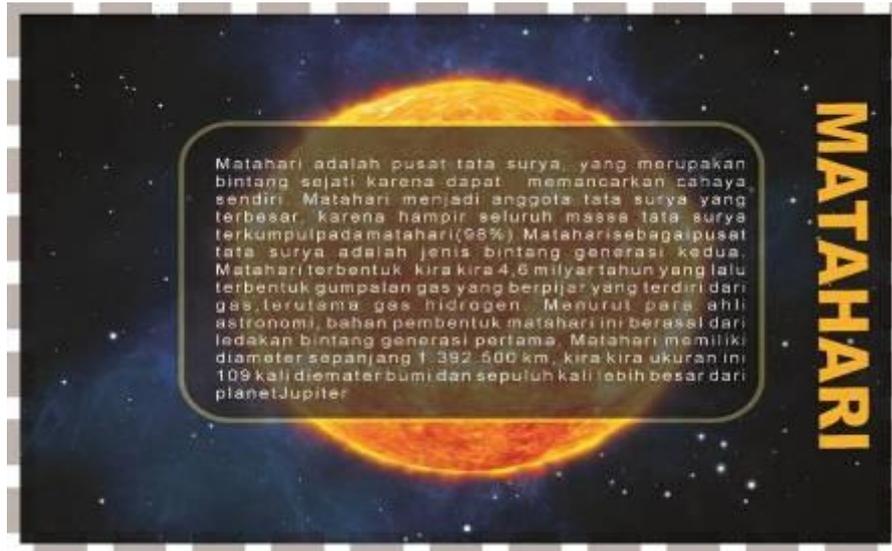
Pada perancangan halaman menu tentang akan ditampilkan penjelasan secara singkat mengenai pembuat aplikasi sistem tata surya menggunakan *augmented reality* ini. Selain itu, terdapat tombol kembali yang akan digunakan untuk kembali ke halaman sebelumnya.

5. Perancangan *marker*

Pada aplikasi *augmented reality* system tata surya menggunakan *markerless*, yaitu *marker* yang tidak berupa barcode hitam putih melainkan gambar atau benda. Terdapat 9 buah *marker* seperti berikut.

a. *Marker Matahari*

Marker matahari berbentuk gambar *full colour* yang terdiri dari judul *marker* dengan *background* berupa gambar matahari serta penjelasan singkat tentang matahari



Gambar III.22

Marker Matahari

b. *Marker Merkurius*

Marker merkurius berbentuk gambar *full colour* yang terdiri dari judul *marker* dengan *background* berupa gambar merkurius serta penjelasan singkat tentang merkurius.

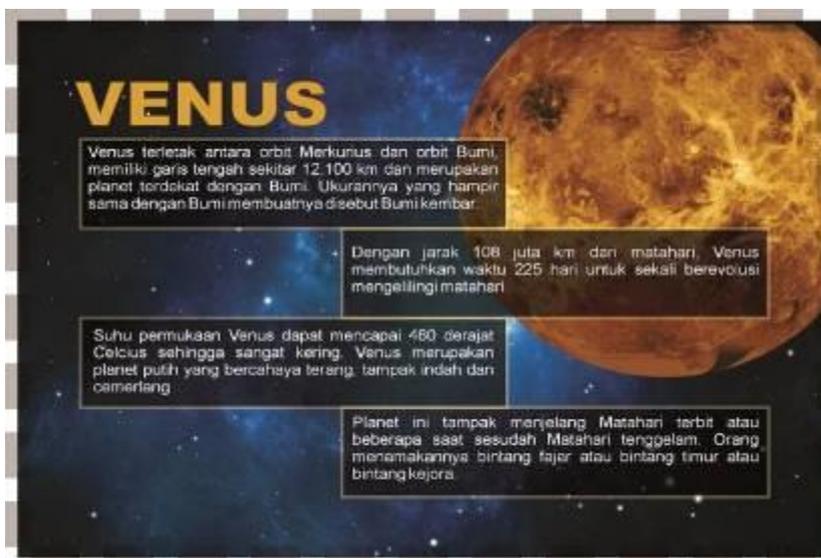


Gambar III.23

Marker Merkurius

c. *Marker Venus*

Marker venus berbentuk gambar *full colour* yang terdiri dari judul *marker* dengan *background* berupa gambar venus serta penjelasan singkat tentang venus.



Gambar III.24

Marker Venus

d. *Marker Bumi*

Marker bumi berbentuk gambar *full colour* yang terdiri dari judul *marker* dengan *background* berupa gambar bumi serta penjelasan singkat tentang bumi.



Gambar III.25

Marker Bumi

e. *Marker Mars*

Marker mars berbentuk gambar *full colour* yang terdiri dari judul *marker* dengan *background* berupa gambar mars serta penjelasan singkat tentang mars.



Gambar III.26

Marker Mars

f. *Marker Jupiter*

Marker jupiter berbentuk gambar *full colour* yang terdiri dari judul *marker* dengan *background* berupa gambar jupiter serta penjelasan singkat tentang jupiter.

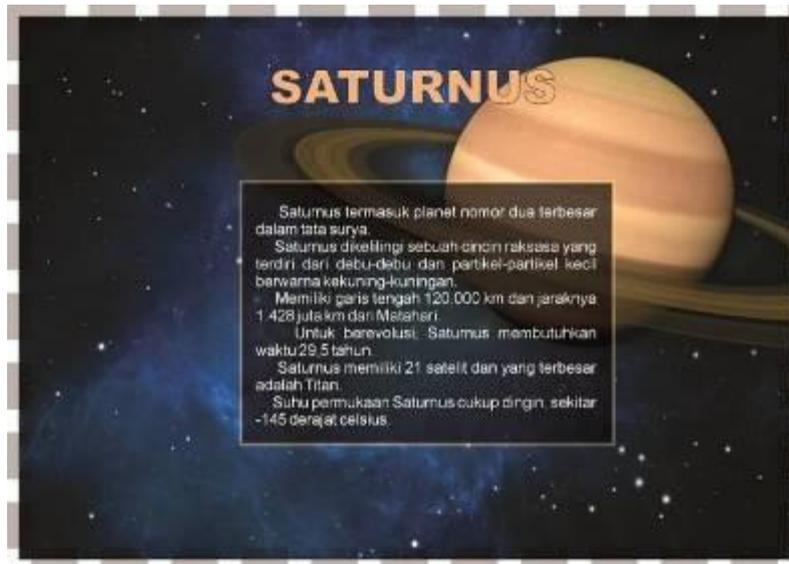


Gambar III.27

Marker Jupiter

g. *Marker* Saturnus

Marker saturnus berbentuk gambar *full colour* yang terdiri dari judul *marker* dengan *background* berupa gambar saturnus serta penjelasan singkat tentang saturnus.



Gambar III.28

***Marker* Saturnus**

h. *Marker* Uranus

Marker uranus berbentuk gambar *full colour* yang terdiri dari judul *marker* dengan *background* berupa gambar uranus serta penjelasan singkat tentang uranus.

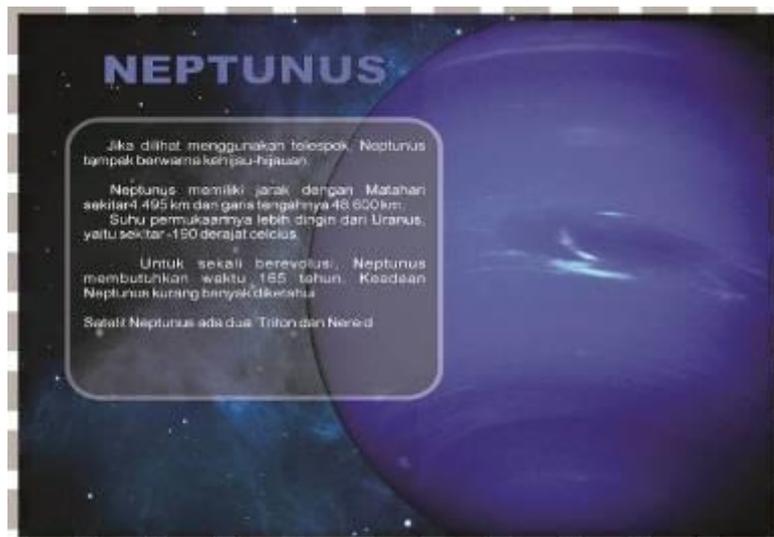


Gambar III.29

Marker Uranus

i. *Marker Neptunus*

Marker neptunus berbentuk gambar *full colour* yang terdiri dari judul *marker* dengan *background* berupa gambar neptunus serta penjelasan singkat tentang neptunus



Gambar III.30

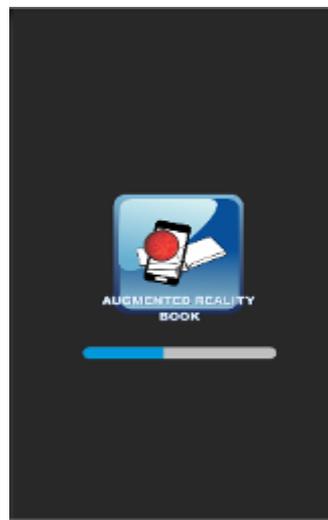
Marker Neptunus

3.3. Implementasi

Implementasi Interface merupakan tampilan dari aplikasi yang dibangun. Aplikasi ini dijalankan pada *smatphone* yang memiliki sistem operasi *android*. Adapun tampilan yang merancang menggunakan *software unity editor* sebagai berikut:

1. *Splash Screen*

Splash screen sebagai pembuka sebelum masuk menu utama. *Splash screen* ditampilkan untuk mengenalkan aplikasi pada pengguna.

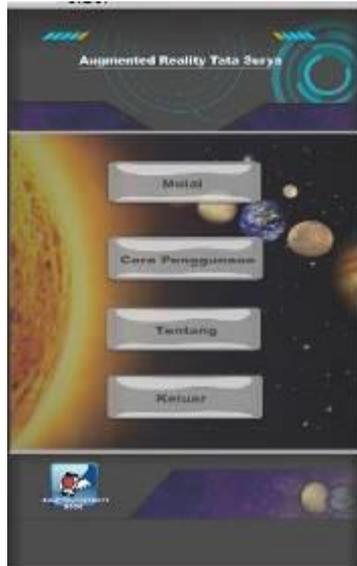


Gambar III. 31

Splash Screen

2. Menu utama

Halaman menu utama ditampilkan setelah halaman *splash screen*. Halaman menu utama menampilkan beberapa *button* menu yang akan mengarah ke halaman selanjutnya. *Button-button* tersebut adalah mulai, cara penggunaan, tentang dan keluar.



Gambar III.32

Menu Utama

3. Menu Mulai

Menu mulai akan tampil ketika tombol mulai pada menu utama ditekan. Menu ini akan menjalankan kamera AR yang jika diarahkan pada marker planet akan menampilkan objek 3D planet, audio penjelasan planet, dan info penjelasan planet sistem tata surya.



Gambar III.33

Menu Mulai *Augmented Reality* Matahari



Gambar III.34

Menu Mulai *Augmented Reality* Merkurius



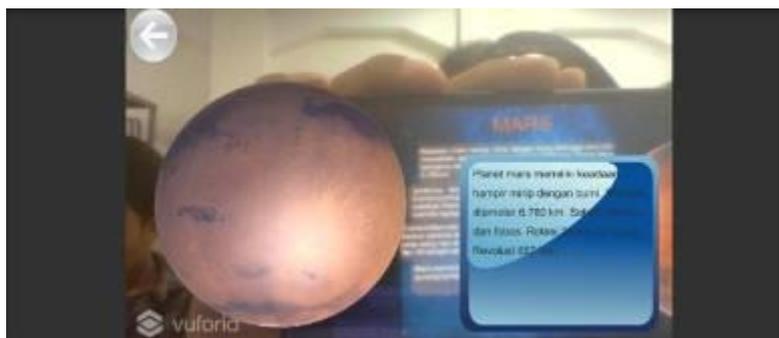
Gambar III.35

Menu Mulai *Augmented Reality* Venus



Gambar III.36

Menu Mulai *Augmented Reality* Bumi



Gambar III.37

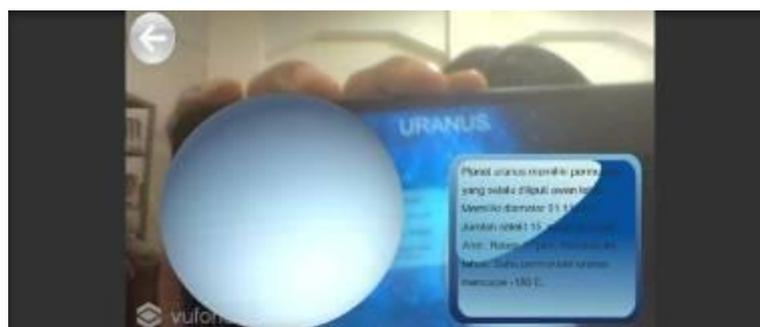
Menu Mulai *Augmented Reality*



Mars

Gambar III.38

Menu Mulai *Augmented Reality* Jupiter



Gambar III.39

Menu Mulai *Augmented Reality* Uranus

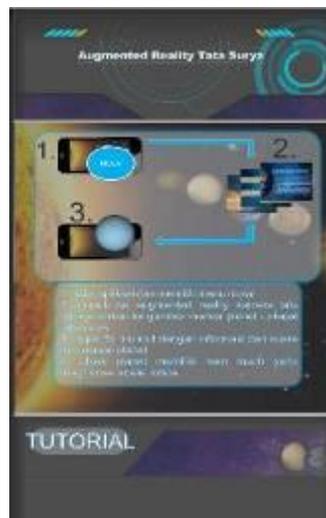


Gambar III.40

Menu Mulai *Augmented Reality* Neptunus

4. Menu Cara Penggunaan

Menu cara penggunaan ini berisi tutorial yang berfungsi membantu pengguna dalam menjalankan aplikasi AR pengenalan sistem tata surya. Menu ini terdapat gambar tutorial dan penjelasan penggunaan.



Gambar III.41

Menu Cara Penggunaan

5. Menu Tentang

Menu tentang ini menampilkan sekilas penjelasan tentang aplikasi dan pembuat aplikasi serta kontak yang bisa dihubungi. Serta terdapat nama sekolah tinggi yang menunjukkan bahwa aplikasi ini dibuat untuk penyusunan skripsi.



Gambar III.42

Menu Info

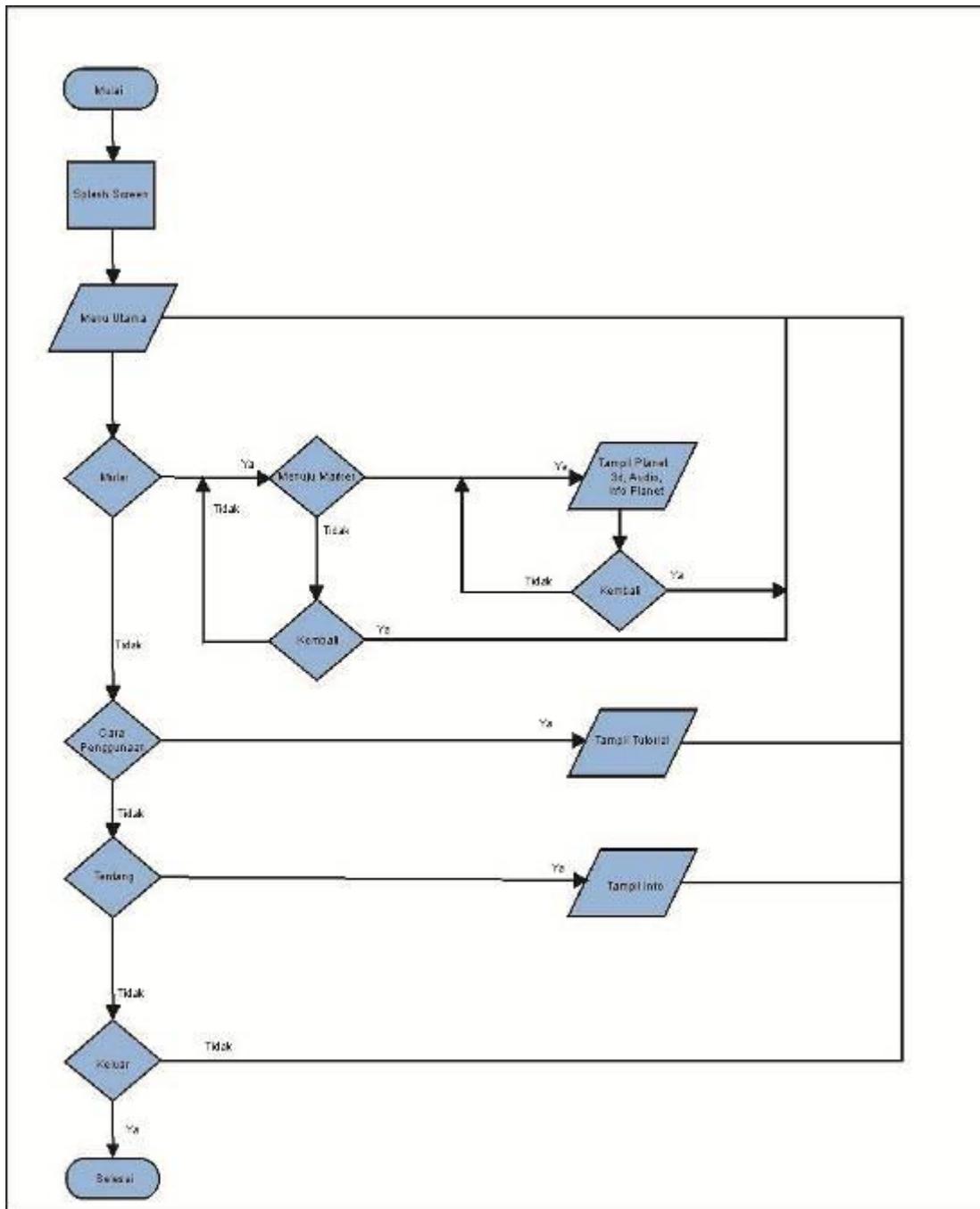
3.4. Pengujian

Pengujian aplikasi ini menggunakan pengujian *white box* dan *black box*. Dengan menggunakan skema diagram alir, berikut merupakan diagram alir dalam aplikasi AR pengenalan sistem tata surya.

1. White Box

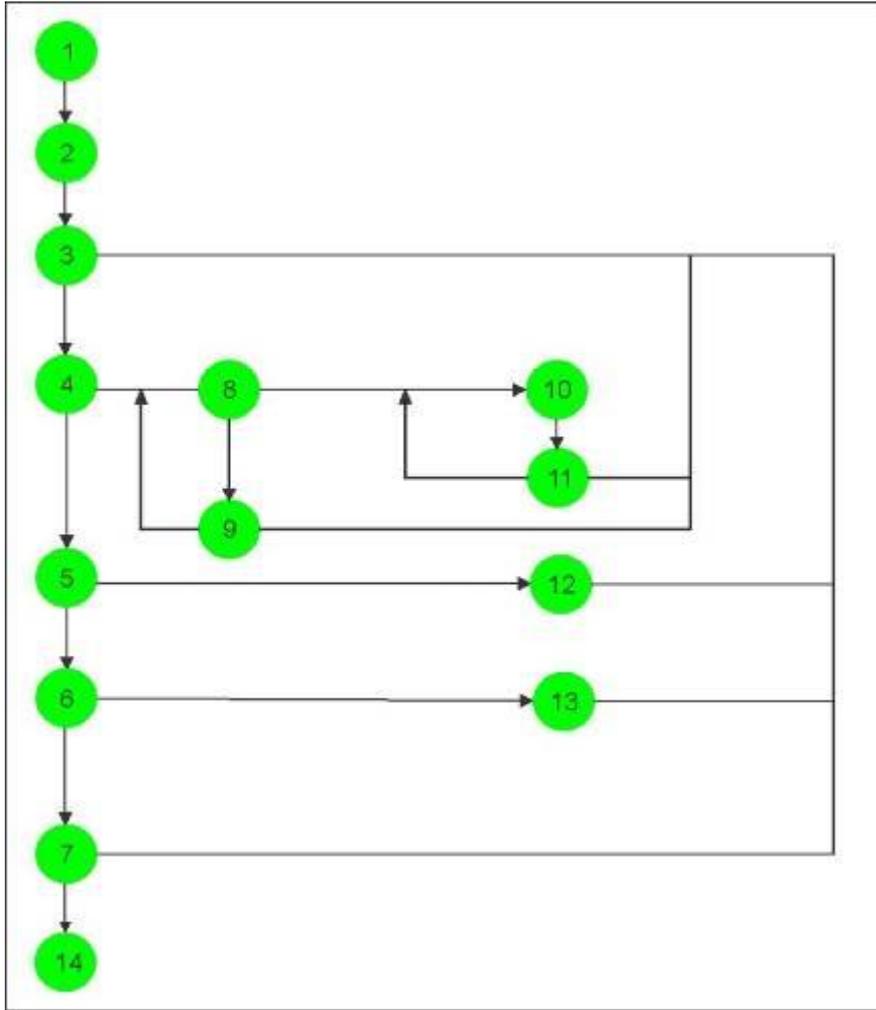
Dalam tahap pengujian *white box* ini penulis menggunakan skema diagram alir untuk mendapatkan hasil pengujian yang akurat. Berikut merupakan *flowchart* dan

skema diagram alir aplikasi *augmented reality* pengenalan sistem tata surya berbasis android :



Gambar III.43

Flowchart Diagram



Gambar III.44

Diagram Alir

Dari skema diagram alir diatas dapat diperoleh kompleksitas siklomatis dengan rumus sebagai berikut:

$$V(G) = E - N + 2$$

E = Jumlah *Edge* yang ditentukan dengan gambar anak panah

N = Jumlah simpul grafik alir yang ditentukan dengan gambar lingkaran

$$V(G) = 21 - 14 + 2 = 9$$

- a) 1-2-3-4-5-6-7

- b) 1-2-3-4-8-9
- c) 1-2-3-4-8-9-8
- d) 1-2-3-4-8-10
- e) 1-2-3-4-8-10-11
- f) 1-2-3-4-8-10-11-10
- g) 1-2-3-4-5-12
- h) 1-2-3-4-5-6-13
- i) 1-2-3-4-5-6-7-14

Ketika aplikasi dijalankan ,maka akan terlihat satu set baris yang dihasilkan adalah 1-2-3-4-5-6-7-3-4-8-9-3-4-8-9-8-3-4-8-10-3-4-8-10-11-3-4-8-10-11-10-3-4-5-12-3-4-5-6-13-3-4-5-6-7-14 dan terlihat bahwa simpul telah dieksekusi satu kali.

2. *Black Box*

Pengujian selanjutnya dilakukan untuk memastikan suatu masukan menjelaskan proses yang tepat dan menghasilkan *output* yang sesuai dengan rancangan.

Tabel III.1

Pengujian *Black Box*

No.	Komponen pengeujian	Unjuk Kerja	Keterangan
1	Buka Aplikasi	Menginstall dan menjalankan aplikasi di <i>smartphone</i> yang memiliki sistem android	Berhasil
2	<i>Splash Screen</i>	Menampilkan <i>splash screen</i> dan loading pada saat splash screen	Berhasil

		berjalan	
3	Menu Utama	Menampilkan menu mulai, cara penggunaan, tentang dan keluar	Berhasil
4	Menu Cara Penggunaan	Menampilkan tutorial penggunaan aplikasi AR sistem tata surya	Berhasil
5	Menu Tentang	Menampilkan penjelasan singkat tentang aplikasi dan pembuat	Berhasil
6	Menu Keluar	Menutup aplikasi dengan menekan menu keluar	Berhasil
7	Menu Mulai	Membuka AR kamera setelah menekan menu mulai	Berhasil
8	Planet Objek 3D	Menampilkan objek 3D planet pada sistem tata surya setelah mendeteksi gambar marker	Berhasil
9	Deskripsi penjelasan planet	Menampilkan deskripsi yang menjelaskan planet pada sistem tata surya saat objek 3D terdeteksi	Berhasil
10	Audio	Mengeluarkan suara <i>dubber</i> yang menjelaskan planet-planet sistem tata surya saat objek 3D terdeteksi	Berhasil

BAB IV

PENUTUP

4.1. Kesimpulan

Kesimpulan yang dapat diperoleh dari pembuatan Aplikasi Pengenalan Sistem Tata Surya *Augmented Reality* adalah :

1. Aplikasi pengenalan sistem tata surya *augmented reality* dapat berjalan di perangkat *mobile* bersistem operasi android dengan spesifikasi RAM diatas 1,5 GB.
2. Aplikasi ini memberikan informasi mengenai sistem tata surya.
3. Menarik minat dalam mempelajari sistem tata surya karena lebih interaktif.
4. Pendeteksian *marker* dapat memberikan informasi pada pengguna saat diarahkan kamera AR dan *marker* pada aplikasi pengenalan sistem tata surya *augmented reality* berjumlah 9 *marker*.

4.2. Saran

Saran yang dapat diperoleh dari pembuatan aplikasi pengenalan sistem tata surya *augmented reality*, antara lain:

1. Animasi yang terdapat pada aplikasi masih statis, maka diharapkan agar dapat dinamis atau bergerak.
2. Aplikasi masih bersifat offline, maka disarankan bersifat online agar bisa mendapatkan lebih banyak lagi informasi.
3. Aplikasi masih berjalan dalam satu platform yaitu android. Kelemahan ini menjadi acuan untuk dapat dikembangkan lagi agar dapat digunakan di beberapa platform.

Daftar Pustaka

- Adami , Feby Zulham dan Cahyani Budihartanti. 2016.Penerapan Teknologi *Augmented Reality* Pada Media Pembelajaran Sistem Pencernaan Berbasis Android. Jakarta: Jurnal Teknik Komputer AMIK BSI. Vol II, NO. 1:122-131
- Admiranto, Agustinus Gunawan. 2017. EKSPLORASI TATA SURYA. Bandung : Mizan Pustaka.
- Irawan. 2012.membuat APLIKASI ANDROID untuk Orang Awam.Palembang : Penerbit Maxicom.
- K.Lee, "Augmented Reality in Education and Training," *TechTrends Link. Res. Pr. Improve Learn.*, vol.56, no.2, pp. 13-21,Mar.2012.
- Maulana, Angga dan Wahyu Kusuma. 2014. Aplikasi *Augmented Reality* Sebagai Media Pembelajaran Tata Surya. Depok : Jurnal Prosiding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2014). Vol. 8, ISSN : 2302-3740
- Mustika, dkk. 2015. Implementasi *Augmented Reality* Sebagai Media Pembelajaran Interaktif. Palembang : Citec Journal. Vol. 2, No. 4, ISSN : 2460-4259
- Pressman, Roger S. 2012. Rekayasa Perangkat Lunak – Buku Satu, Pendekatan Praktisi(Edisi 7). Yogyakarta : Andi.
- Qualcomm. Developing with Vuforia. Diambil dari : <https://library.vuforia.com/getting-started> (9 Mei 2017)
- Roedavan Rickman. 2016. Unity - Tutorial Game Engine. Bandung : Informatika Bandung.
- Sadeli, Muhammad. 2014. Toko Buku Online dengan ANDROID. Palembang : Maxikom.
- Safaat, Nazaruddin. 2015. Pemrograman Aplikasi Mobile Smartphone dan Tablet Pc Berbasis Android Revisi Kedua. Bandung: Informatika.
- Setiawan, Erwin, Undang Syaripudin dan Yana Aditya Gerhana. 2016. Implementasi Teknologi Augmented Reality Pada Buku Panduan Wudhu Berbasis Mobile Android.Bandung:Jurnal Jurusan Teknik Informatika, Fakultas Sains dan Teknologi. Vol. 1, ISSN : 2527-9165
- Sukamto,Rosa A. dan M.Shalahuddin.2014.Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek.Bandung : Informatika Bandung.

DAFTAR RIWAYAT HIDUP

A. Biodata Mahasiswa

N.I.M : 13131067
Nama Lengkap : Ridwan Setiawan
Tempat & Tanggal Lahir : Jakarta, 22-01-1995
Alamat Lengkap : Jl.Madrasah I,Kebon jeruk 007/09 Jakarta Barat

B. Riwayat Pendidikan Formal & Non-Formal

1. SDN 04 PG Jakarta, lulus tahun 2007
2. SMPN 229 Jakarta, lulus tahun 2010
3. SMK Cipta Karya II Jakarta, lulus tahun 2013

Jakarta, 7 Agustus 2017



Ridwan Setiawan

	LEMBAR KONSULTASI BIMBINGAN SKRIPSI
	SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER NUSA MANDIRI

NIM : 12135288
 Nama : Ridwan Setiawan
 Dosen Pembimbing I : Yulia Safitri, M.Kom
 Judul Skripsi : Penerapan Teknologi Augmented Reality Pada Pengenalan Sistem Tata Surya Berbasis Android

No.	Tanggal Bimbingan	Pokok Bahasan	Paraf Dosen Pembimbing I
1	12 April 2017	Bimbingan Perdana Dan Pengajuan Judul	
2	20 Mei 2017	Bimbingan Bab I	
3	26 Mei 2017	Revisi Bab I & Bimbingan Bab II	
4	02 Juni 2017	Revisi Bab II & Bimbingan Bab III	
5	09 Juni 2017	Revisi Bab III	
6	18 Juli 2017	Demo Aplikasi	
7	24 Juli 2017	Bimbingan Bab IV	
8	08 Agustus 2017	Acc Keseluruhan	

Bimbingan Skripsi

- Dimulai Tanggal : 12 April 2017
- Diakhiri pada tanggal : 08 Agustus 2017
- Jumlah Pertemuan Bimbingan : 8 Kali Pertemuan

Disetujui Oleh,
Dosen Pembimbing I



Yulia Safitri, M.Kom

	LEMBAR KONSULTASI BIMBINGAN SKRIPSI
	SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER NUSA MANDIRI

NIM : 12135288
 Nama : Ridwan Setiawan
 Dosen Pembimbing II: Wawan Gunawan, S.Kom., M.T.
 Judul Skripsi : Penerapan Teknologi Augmented Reality Pada
 Pengenalan Sistem Tata Surya Berbasis Android

No.	Tanggal Bimbingan	Pokok Bahasan	Paraf Dosen Pembimbing II
1	14 April 2017	Bimbingan Perdana Dan Pengajuan Judul	W
2	19 April 2017	Bimbingan Bab I	W
3	27 April 2017	Revisi Bab I & Bimbingan Bab II	W
4	04 Mei 2017	Revisi Bab II & Bimbingan Bab III	W
5	11 Mei 2017	Revisi Bab III	W
6	20 Juli 2017	Demo Aplikasi	W
7	26 Juli 2017	Bimbingan Bab IV	W
8	09 Agustus 2017	Acc Keseluruhan	W

Bimbingan Skripsi

- Dimulai Tanggal : 14 April 2017
- Diakhiri pada tanggal : 09 Agustus 2017
- Jumlah Pertemuan Bimbingan : 8 Kali Pertemuan

Disetujui Oleh,
Dosen Pembimbing II



Wawan Gunawan, S.Kom., M.T.

LAMPIRAN

Splash.cs

```
using UnityEngine;
using System.Collections;

public class splash : MonoBehaviour {

    void Start() {
        StartCoroutine(Example());
    }

    IEnumerator Example() {
        yield return new WaitForSeconds(3);
        Application.LoadLevel ("loading");
    }
}
```

Loading.cs

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class loading : MonoBehaviour {

    public Transform LoadingBar;

    [SerializeField] private float currentAmount;
    [SerializeField] private float speed;

    // Update is called once per frame
    void Update () {
        if (currentAmount < 100) {
            currentAmount += speed * Time.deltaTime;
            Debug.Log ((int)currentAmount);
        } else {
            Application.LoadLevel ("main menu");
        }

        LoadingBar.GetComponent<Image> ().fillAmount = currentAmount / 100;
    }
}
```

Menu.cs

```
using UnityEngine;
using System.Collections;

public class menu : MonoBehaviour {

    public void GoToMainMenu(){
        Application.LoadLevel("main menu");
    }
    public void GoToCaraPenggunaan(){
        Application.LoadLevel ("tutorial");
    }
    public void GoToTentang(){
        Application.LoadLevel ("tentang");
    }

    public void GoToARCamera(){
        Application.LoadLevel("AR tatasurya");
    }
    public void ExitApplication(){
        Application.Quit ();
    }
}
```

CameraFocusMode.cs

```
using UnityEngine;
using System.Collections;
using Vuforia;

public class CameraFocusMode : MonoBehaviour {
    void Start ()
    {
        var vuforia = VuforiaARController.Instance;
        vuforia.RegisterVuforiaStartedCallback(OnVuforiaStarted);
        vuforia.RegisterOnPauseCallback(OnPaused);
    }

    private void OnVuforiaStarted()
    {
        CameraDevice.Instance.SetFocusMode(
            CameraDevice.FocusMode.FOCUS_MODE_CONTINUOUSAUTO);
    }
}
```

```

private void OnPaused(bool paused)
{
    if (!paused) // resumed
    {
        // Set again autofocus mode when app is resumed
        CameraDevice.Instance.SetFocusMode(
            CameraDevice.FocusMode.FOCUS_MODE_CONTINUOUSAUTO);
    }
}
}

```

DefaultTrackableEventHandler.cs

```

/*=====
=====
Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
=====
=====*/

```

```

using UnityEngine;
using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;

namespace Vuforia
{
    /// <summary>
    /// A custom handler that implements the ITrackableEventHandler interface.
    /// </summary>
    public class DefaultTrackableEventHandler : MonoBehaviour,
        ITrackableEventHandler
    {
        //-----Begin sound-----
        public AudioSource soundTarget;
        public AudioClip clipTarget;
        private AudioSource[] allAudioSources;

        //-----begin panel-----
        public Transform TextDescription;
        public Transform PanelDescription;

        //-----to stop all sounds
        void StopAllAudio()
        {

```

```

        allAudioSources = FindObjectsOfType(typeof(Audio
Source)) as AudioSource[] ;
        foreach (AudioSource audioS in allAudioSources)
        {
            audioS.Stop();
        }
    }

    //function to play sound
    void playSound(string ss)
    {
        clipTarget = (AudioClip)Resources.Load(ss);
        soundTarget.clip = clipTarget;
        soundTarget.loop = false;
        soundTarget.playOnAwake = false;
        soundTarget.Play();
    }
    #region PRIVATE_MEMBER_VARIABLES

    private TrackableBehaviour mTrackableBehaviour;

    #endregion // PRIVATE_MEMBER_VARIABLES

    #region UNITY_MONOBEHAVIOUR_METHODS

    void Start()
    {
        mTrackableBehaviour = GetComponent<TrackableBehaviour>();
        if (mTrackableBehaviour)
        {
            mTrackableBehaviour.RegisterTrackableEventHandler(this);
        }
        //Register / add the AudioSource as object
        soundTarget = (AudioSource)gameObject.AddComponent<AudioSource>();
    }

    #endregion // UNITY_MONOBEHAVIOUR_METHODS

    #region PUBLIC_METHODS

    /// <summary>
    /// Implementation of the ITrackableEventHandler function called when the

```

```

    /// tracking state changes.
    /// </summary>
    public void OnTrackableStateChanged(
        TrackableBehaviour.Status previousStatus,
        TrackableBehaviour.Status newStatus)
    {
        if (newStatus == TrackableBehaviour.Status.DETECTED ||
            newStatus == TrackableBehaviour.Status.TRACKED ||
            newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
        {
            OnTrackingFound();
        }
        else
        {
            OnTrackingLost();
        }
    }

#endregion // PUBLIC_METHODS

#region PRIVATE_METHODS

    private void OnTrackingFound()
    {
        Renderer[] rendererComponents = GetComponentsInChildren<Renderer>(true);
        Collider[] colliderComponents = GetComponentsInChildren<Collider>(true);

        // Enable rendering:
        foreach (Renderer component in rendererComponents)
        {
            component.enabled = true;
        }

        // Enable colliders:
        foreach (Collider component in colliderComponents)
        {
            component.enabled = true;
        }

        Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");

        //Play Sound, If detect an target
        if (mTrackableBehaviour.TrackableName == "mercurius")

```

```

    {
        playSound ("sounds/audio-mercurius");
    }
    if (mTrackableBehaviour.TrackableName == "matahari")
    {
        playSound ("sounds/audio-matahari");
    }
}

private void OnTrackingLost()
{
    Renderer[] rendererComponents = GetComponentsInChildren<Renderer>(true);
    Collider[] colliderComponents = GetComponentsInChildren<Collider>(true);

    // Disable rendering:
    foreach (Renderer component in rendererComponents)
    {
        component.enabled = false;
    }

    // Disable colliders:
    foreach (Collider component in colliderComponents)
    {
        component.enabled = false;
    }

    Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " lost");

    TextDescription.gameObject.SetActive(false);
    PanelDescription.gameObject.SetActive(false);

    //Stop All Sounds if target lost
    StopAllAudio();
}

#endregion // PRIVATE_METHODS
}
}

```

