



**IMPLEMENTASI TEKNOLOGI *AUGMENTED REALITY* PADA
MAJALAH INTERAKTIF BERBASIS ANDROID MENGGUNAKAN
METODE ALGORITMA *FEATURES FROM ACCELERATED SEGMENT
TEST(FAST) CORNER DETECTION*
(STUDI KASUS MAJALAH BINARY PROGRAM STUDI SISTEM
INFORMASI UNIVERSITAS JEMBER)**

SKRIPSI

diajukan guna memenuhi salah satu syarat
untuk menyelesaikan pendidikan di Program Studi Sistem Informasi
Universitas Jember dan mendapat gelar Sarjana Sistem Informasi

Oleh

**Rozi Rikza Akbar
NIM 112410101058**

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2016



**IMPLEMENTASI TEKNOLOGI *AUGMENTED REALITY* PADA
MAJALAH INTERAKTIF BERBASIS ANDROID MENGGUNAKAN
METODE ALGORITMA *FEATURES FROM ACCELERATED SEGMENT
TEST(FAST) CORNER DETECTION*
(STUDI KASUS MAJALAH BINARY PROGRAM STUDI SISTEM
INFORMASI UNIVERSITAS JEMBER)**

SKRIPSI

diajukan guna memenuhi salah satu syarat
untuk menyelesaikan pendidikan di Program Studi Sistem Informasi
Universitas Jember dan mendapat gelar Sarjana Sistem Informasi

Oleh

**Rozi Rikza Akbar
NIM 112410101058**

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT, yang telah memberikan kekuatan dan kelancaran dalam menyelesaikan skripsi ini;
2. Ibuku tercinta Halimah serta Ayahku M. Fathorrozi atas segala kesabaran dan dukungan, kasih sayang, dan motivasi yang luar biasa selama ini;
3. Kakakku tercinta Rosita Fitrah Dewi;
4. Adikku tercinta M. Hasan Ulil Abror Al-Rozi
5. Sahabat-sahabatku;
6. Dosen Pembimbing;
7. Almamater Program Studi Sistem Informasi Universitas Jember

MOTTO

“Sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain, dan hanya kepada Tuhanmulah hendaknya kamu berharap.”

(Q.S Al Insyirah : 6-8)

“God’s plan is always more beautiful than our desire.”

(Anonymous)

“Ilmu adalah lebih baik daripada kekayaan kerana kekayaan harus dijaga, sedangkan ilmu menjaga kamu.”

(Ali bin Abi Thalib)

"Kemenangan yang seindah-indahnya dan sesukar-sukarnya yang boleh direbut oleh manusia ialah menundukan diri sendiri."

(Ibu Kartini)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Rozi Rikza Akbar

NIM : 112410101058

menyatakan sesungguhnya bahwa karya ilmiah yang berjudul “Implementasi Teknologi *Augmented Reality* Pada Majalah Interaktif Berbasis Android Menggunakan Metode Algoritma *Features from Accelerated Segment Test (FAST) Corner Detection*. (Studi Kasus Majalah *Binary* Program Studi Sistem Informasi Universitas Jember)” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isisnya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 03 Juni 2015

Yang menyatakan,

Rozi Rikza Akbar

NIM. 112410101058

SKRIPSI

**IMPLEMENTASI TEKNOLOGI *AUGMENTED REALITY* PADA
MAJALAH INTERAKTIF BERBASIS ANDROID MENGGUNAKAN
METODE ALGORITMA *FAST CORNER DETECTION*
(STUDI KASUS MAJALAH BINARY PROGRAM STUDI SISTEM
INFORMASI UNIVERSITAS JEMBER)**

Oleh :

Rozi Rikza Akbar

NIM 112410101058

Pembimbing

Pembimbing Utama : Dr. Saiful Bukhori ST., M.Kom

NIP 19681113 199412 1 001

Pembimbing Anggota : Nelly Oktavia Adiwijaya S.Si.,MT

NIP 19841024 200912 2 008

PENGESAHAN PEMBIMBING

Skripsi berjudul “Implementasi Teknologi *Augmented Reality* Pada Majalah Interaktif Berbasis Android Menggunakan Metode Algoritma Features from Accelerated Segment Test (*FAST*) *Corner Detection*. (Studi Kasus Majalah *Binary* Program Studi Sistem Informasi Universitas Jember)”, telah diuji dan disahkan pada:

Hari tanggal : Senin, 09 Mei 2016

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Dr. Saiful Bukhori ST., M.Kom.

Nelly Oktavia Adiwijaya S.Si.,MT.

NIP. 196811131994121001

NIP. 198410242009122008

PENGESAHAN PENGUJI

Skripsi berjudul “Implementasi Teknologi *Augmented Reality* Pada Majalah Interaktif Berbasis Android Menggunakan Metode Algoritma Features from Accelerated Segment Test (*FAST*) *Corner Detection*. (Studi Kasus Majalah *Binary* Program Studi Sistem Informasi Universitas Jember)”, telah diuji dan disahkan pada:

Hari tanggal : Senin, 09 Mei 2016

Tempat : Program Studi Sistem Informasi Universitas Jember

Penguji I,

Penguji II,

Anang Andrianto ST.,MT

NIP 19690615 1997021002

Muhamad Arief Hidayat S.Kom.,.M.Kom.

NIP 19810123 2010121003

Mengesahkan

Ketua Program Studi

Prof. Drs. Slamir, M.Comp.Sc.,Ph.D

NIP. 19670420 1992011001

RINGKASAN

Implementasi Teknologi *Augmented Reality* Pada Majalah Interaktif Berbasis Android Menggunakan Metode Algoritma *Fast Corner Detection* (Studi Kasus Majalah Binary Program Studi Sistem Informasi Universitas Jember); Rozi Rikza Akbar, 112410101058; 2015; 99 halaman; Program Studi Sistem Informasi Universitas Jember.

Majalah merupakan salah satu media cetak yang sangat populer belakangan ini tak terkecuali dengan majalah Binary Program Studi Sistem Informasi Universitas Jember. Tapi dalam perkembangannya media cetak ini terkesan 1 arah karena hanya bisa dibaca dan terkadang pembaca tidak bisa menerima informasi yang lebih karena hanya terdapat gambar di majalah tersebut. Pada penelitian ini, *Augmented Reality* dapat diimplementasikan kedalam majalah agar majalah terlihat lebih menarik dan atraktif untuk dibaca karena pembaca juga dapat berperan dalam hal ini. *Augmented Reality* menawarkan keunggulannya dalam menggabungkan objek 3D, video dan suara dengan lingkungan realitas. Sedangkan untuk mengimbangi hal tersebut, teknologi *Augmented Reality* harus lebih cepat dalam memproses marker sebagai acuan objek-objek yang akan ditampilkan.

Pada penelitian ini akan dipaparkan mengenai algoritma *FAST (Features from Accelerated Segment Test) Corner Detection* yang akan digunakan dalam teknologi *Augmented Reality*. Selain itu juga akan dibahas apa saja yang dapat mempengaruhi baik tidaknya pengimplementasian algoritma *FAST Corner Detection*. Algoritma *FAST Corner Detection* merupakan salah satu algoritma dalam pengenalan objek 2D yang memanfaatkan tingkat kecerahan suatu objek gambar 2D yang akan dijadikan marker sehingga setiap titik tetapan algoritma ini sangat presisi.

Aplikasi Binary *Augmented Reality* ini dapat dijalankan pada android versi Ice Cream Sandwich dan seterusnya. Fitur-fitur yang diperlihatkan juga sederhana. Sedangkan pada tampilan utama yakni berupa tampilan tangkapan kamera untuk mendeteksi marker.

PRAKATA

Puji syukur kehadirat Allah Subhanahu Wata'ala atas segala Rahmat dan Karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Implementasi Teknologi *Augmented Reality* Pada Majalah Interaktif Berbasis Android Menggunakan Metode Algoritma *Fast Corner Detection*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamir, M.CompSc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember.
Dr. Saiful Bukhori, ST., M.Kom, selaku Dosen Pembimbing Utama dan Nelly Oktavia Adiwijaya S.Si.,MT, selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini.
2. Dr. Saiful Bukhori, ST., M.Kom, selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa.
3. Aba M. Fathorrazi dan Ummi Halimah yang telah sangat memotivasi penulis dan memberikan doa yang tulus.
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember.
5. Adikku M.H. Ulil Abror Al-Rozi dan Kakakku Rosita Fitrah Dewi yang memberikan semangat dan motivasi.
6. Sabahat terbaik yang telah bersama – sama melewati skripsi.
7. Teman-teman mahasiswa Program Studi Sistem Informasi Universitas Jember.
8. Semua pihak yang telah membantu baik tenaga maupun pikiran dalam pelaksanaan kegiatan penelitian dan penyusunan skripsi ini.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua

pihak demi perbaikan dan kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 03 Juni 2016

Penulis



DAFTAR ISI

SKRIPSI.....	i
PERSEMBAHAN	iii
MOTTO	iv
PERNYATAAN.....	v
PENGESAHAN PEMBIMBING.....	vii
PENGESAHAN PENGUJI.....	viii
RINGKASAN	ix
PRAKATA.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan dan Manfaat.....	3
1.3.1 Tujuan	3
1.3.2 Manfaat	3
1.4 Batasan Masalah.....	4
1.5 Sistematika Penulisan.....	5
BAB 2 TINJAUAN PUSTAKA	6
2.1 Augmented Reality (AR).....	6
2.2 Sistem Display Augmented Reality	6
2.3 Citra Digital	7

2.3.1	Citra Biner (Monokrom)	7
2.3.2	Citra <i>Grayscale</i>	8
2.3.3	Citra Warna (true color)	8
2.4	Algoritma FAST Corner Detection	9
2.5	Vuforia Qualcomm	12
2.6	Multimedia Development Life Cycle (MDLC)	12
BAB 3	METODOLOGI PENELITIAN	15
3.1	Pengonsepan	15
3.1.1	Jenis Penelitian	15
3.1.2	Tempat Penelitian	15
3.1.3	Alat Penelitian	16
3.1.4	Gambaran Umum Aplikasi	16
3.2	Desain dan Perancangan Aplikasi	17
3.3	Pengumpulan Bahan	17
3.4	Implementasi Pembuatan Aplikasi	18
3.5	Pengujian Aplikasi	19
BAB 4	DESAIN DAN PERANCANGAN APLIKASI	20
4.1	Analisis Kebutuhan Perangkat Lunak	20
4.2	Diagram Alir(Flowchart)	21
4.3	Storyboard	22
4.4	Implementasi Perancangan	23
4.9	Pengujian Blackbox dan Whitebox	24
4.9.1	Pengujian Whitebox	24
4.9.2	Pengujian Blackbox	24
BAB 5	HASIL DAN PEMBAHASAN	28

5.1	Pemrosesan Marker	28
5.1.1	Resize	30
5.1.2	Convert to <i>Grayscale</i>	31
5.1.3	Convert to Histogram	32
5.1.4	<i>Threshold</i>	32
5.1.5	Point Mark.....	32
5.2	Analisis Pengenalan Markerless.....	33
5.3.1	Analisis Alur Kerja AR.....	33
5.3.2	Analisis Kualitas Marker.....	35
5.3	Analisis Output.....	41
5.3.1	Analisis Output Objek 3D.....	41
5.3.2	Analisis Output Video.....	43
5.3.3	Analisis Output Suara/Musik.....	44
BAB 6	PENUTUP	45
6.1	Kesimpulan.....	45
6.2	Saran.....	46
DAFTAR PUSTAKA	47
LAMPIRAN	49

DAFTAR GAMBAR

Gambar 2-1 Pembentukan citra untuk *display augmented reality* 6

Gambar 2-2 Gambar susunan piksel pada citra monokrom 7

Gambar 2-3 Gradasi warna citra *grayscale* 8

Gambar 2-4 Citra Warna 8

Gambar 2-5 Deteksi sudut pertitik FAST Corner Detection 10

Gambar 2-6 Diagram alir algoritma FAST Corner Detection 11

Gambar 2-7 Multimedia Development Life Cycle 12

Gambar 4-1 Diagram alir gambaran aplikasi 21

Gambar 4-2 Storyboard aplikasi Binary Augmented Reality 22

Gambar 5-1 Diagram alir pembuatan marker 28

Gambar 5-2 Proses upload gambar marker 29

Gambar 5-3 Notifikasi gambar tidak memenuhi syarat menjadi marker 29

Gambar 5-4 Daftar gambar marker 30

Gambar 5-5 Proses export dataset gambar marker 30

Gambar 5-6 Proses resize marker 31

Gambar 5-7 Proses convert marker ke grayscale 31

Gambar 5-8 Proses convert grayscale marker ke histogram 32

Gambar 5-9 Proses treshold marker dari histogram 32

Gambar 5-10 Penentuan corner 33

Gambar 5-11 Diagram alir pengenalan markerless 33

Gambar 5-12 Kode program inialisasi QCAR 34

Gambar 5-13 Kode program Tracking dan Daftar ImageTarget 34

Gambar 5-14 Kode program render objek 35

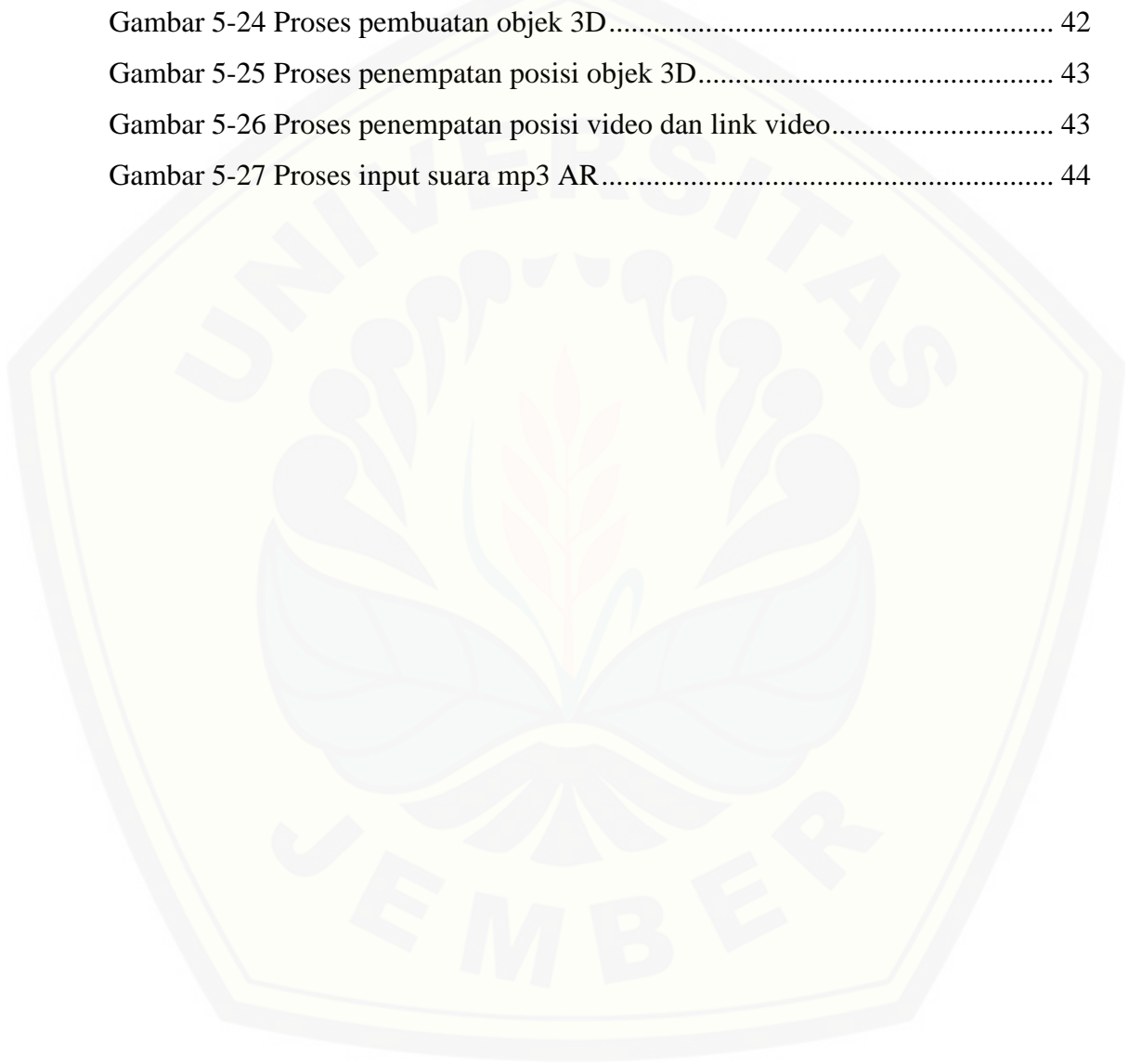
Gambar 5-15 Sample gambar high quality marker 36

Gambar 5-16 Sample gambar setelah diproses menggunakan FAST Corner Detection 37

Gambar 5-17 Histogram sample gambar high quality marker 37

Gambar 5-18 Sample gambar medium quality marker 38

Gambar 5-19 Sample gambar setelah diproses	39
Gambar 5-20 Histogram sample gambar medium quality marker.....	39
Gambar 5-21 Sample gambar low quality marker	40
Gambar 5-22 Setelah diprosesoleh vuforia menggunakan algoritma FAST	40
Gambar 5-23 Histogram sample gambar low quality marker	41
Gambar 5-24 Proses pembuatan objek 3D.....	42
Gambar 5-25 Proses penempatan posisi objek 3D.....	43
Gambar 5-26 Proses penempatan posisi video dan link video.....	43
Gambar 5-27 Proses input suara mp3 AR.....	44



DAFTAR TABEL

Tabel 4.1 Tabel Pengujian Blackbox 25



BAB 1 PENDAHULUAN

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

1.1 Latar Belakang

Perkembangan teknologi pada saat ini berkembang dengan begitu pesat. Teknologi- teknologi baru terus bermunculan untuk membantu pekerjaan manusia menjadi lebih mudah. Seiring dengan perkembangan tersebut, muncul teknologi baru dalam hal informasi, edukasi dan komunikasi. Salah satu bentuk media yang berkembang adalah majalah. Majalah pada saat ini masih banyak yang menggunakan cara konvensional meskipun ada majalah yang sudah berbentuk digital. Majalah digunakan untuk mengetahui produk dan informasi terkini tentang hal yang berkaitan dengan tema majalah tersebut. Misalnya majalah IT membicarakan tentang produk-produk IT terkini dan informasi terkini tentang IT. Salah satu majalah IT adalah majalah Binary yang ada di Program Studi Sistem Informasi Universitas Jember.

Majalah binary merupakan majalah yang dicetak oleh UKM Binary Program Studi Sistem Informasi yang berfokus pada pembahasan masalah teknologi terbaru dan hal-hal yang berkaitan dengan kegiatan mahasiswa di Program Studi Sistem Informasi Universitas Jember. Majalah ini masih menggunakan majalah konvensional dalam menyampaikan informasi yakni dengan menggunakan majalah baca, sehingga menerapkan teknologi *Augmented Reality* (AR) pada majalah akan membuat majalah ini lebih menarik untuk dibaca dan sesuai dengan era majalah yang berkembang pada saat ini.

Augmented Reality (AR) adalah istilah untuk benda-benda nyata dan digital yang digabungkan di lingkungan nyata, terdapat integrasi antar benda tersebut dalam bentuk tiga dimensi yaitu benda digital terintegrasi dalam dunia nyata dan berjalan secara interaktif dalam waktu nyata. Selain menambahkan

benda digital dalam lingkungan nyata, realitas ditambah juga berpotensi menghilangkan benda-benda yang sudah ada. Menambah sebuah lapisan gambar digital dimungkinkan untuk menghilangkan atau menyembunyikan lingkungan nyata dari pandangan pengguna. Misalnya, untuk menyembunyikan sebuah meja dalam lingkungan nyata, perlu digambarkan lapisan representasi tembok dan lantai kosong yang diletakkan di atas gambar meja nyata, sehingga menutupi meja nyata dari pandangan pengguna (Azuma R. T., 1997).

Pada paper berjudul “*Implementation of Augmented Reality System for Smartphone Advertisements*” oleh Young-geun Kim dan Won-jung Kim dibangun sebuah aplikasi yang digunakan untuk mempresentasikan objek tiga dimensi pada *leaflet*/brosur dengan memanfaatkan *augmented reality* sebagai alat bantu. Alat bantu yang digunakan untuk mempresentasikan objek 3D dan informasinya ialah Vuforia SDK dan Engine Unity 3D. Gambar yang digunakan untuk menampilkan objek 3D diupload ke vuforia. Proses selanjutnya memberikan *corner point* dan koordinat dengan menggunakan algoritma FAST sehingga gambar ini memiliki koordinat dan *corner point* yang unik.

Penelitian lain yang berjudul “*Machine Learning for high-speed Corner Detection*” oleh Edward Rosten dan Tom Drummond yang dilakukan di Fakultas Teknik Cambridge University UK. Penelitian ini dilakukan dengan cara membandingkan beberapa algoritma untuk mengenal sudut dari sebuah image. Algoritma yang dibandingkan antara lain FAST, Harris, Shi & Tomasi, DoG, Harris-Laplace, dan SUSAN Algorithm. Hasil dari penelitian ini menyatakan bahwa algoritma tercepat yang dapat menganalisis *corner* dengan cepat adalah algoritma FAST(*Features from Accelerated Segment Test*).

Berdasarkan paper tersebut, penelitian ini akan menggunakan *augmented reality* sebagai alat untuk memanggil objek tiga dimensi, Unity sebagai Application Builder dan Vuforia SDK sebagai library Augmented Reality yang didalamnya terdapat algoritma FAST *Corner Detection*. Hasil akhir dari penelitian ini dapat membangun Aplikasi Majalah Interaktif agar lebih menarik dengan mengimplementasikan teknologi *Augmented Reality* (AR) berbasis android dengan menggunakan algoritma FAST *Corner Detection*. Dengan

menerapkan algoritma ini diharapkan Augmented Reality yang akan diterapkan pada Majalah Binary dapat berjalan dengan optimal sehingga dengan *device* yang memiliki spesifikasi kecil seperti smartphone dapat digunakan untuk aplikasi ini.

1.2 Rumusan Masalah

Berdasarkan dari beberapa permasalahan yang telah diuraikan diatas, maka dapat diambil rumusan masalah yaitu bagaimana merancang dan membangun aplikasi untuk mengimplementasikan *Augmented Reality* dengan menggunakan algoritma *FAST Corner Detection* pada hardware yang memiliki spesifikasi kecil seperti smartphone Android.

1.3 Tujuan dan Manfaat

Tujuan dan manfaat berisi tentang tujuan dari penelitian pengimplementasian teknologi Augmented Reality dengan algoritma *FAST Corner Detection* untuk digunakan pada majalah interaktif *Binary* Program Studi Sistem Informasi Universitas Jember. Sedangkan pada bagian manfaat berisi tentang manfaat apa saja yang akan diperoleh pada penelitian ini, baik bagi peneliti sendiri maupun bagi objek pada penelitian ini.

1.3.1 Tujuan

Tujuan dari penelitian ini adalah :

- a. Merancang dan membangun sebuah aplikasi majalah interaktif dengan menerapkan *augmented reality* berbasis *android*.
- b. Mengimplementasikan *markerless based tracking* dalam majalah interaktif.
- c. Menerapkan algoritma *FAST Corner Detection* pada *markerless based tracking*.

1.3.2 Manfaat

Manfaat penelitian ini adalah :

a. Manfaat akademis

Penelitian ini diharapkan dapat memberikan informasi dan kontribusi untuk orang lain yang membutuhkan sesuatu yang berkenaan dengan judul penelitian ini. Selain itu, penelitian ini merupakan salah satu upaya untuk menambah varian judul penelitian di Program Studi Sistem Informasi Universitas Jember.

b. Manfaat bagi peneliti

- 1) Mengetahui bagaimana proses penerapan algoritma *FAST Corner Detection* untuk pengembangan aplikasi berbasis Augmented Reality yang digunakan pada majalah Program Studi Sistem Informasi Universitas Jember.
- 2) Mengetahui cara penerapan Augmented Reality dengan markerless based tracking.

c. Manfaat bagi objek penelitian

- 1) Memberikan inovasi baru pada dunia majalah. Sehingga majalah tidak hanya dipandang sebagai benda yang hanya bisa dibaca saja. Namun bisa juga berinteraksi dengan pembaca dengan menggunakan teknologi *Augmented Reality*.
- 2) Membantu redaksi *Binary* dalam pengembangan majalah *Binary* dalam pemanfaatan teknologi khususnya *Augmented Reality* agar lebih menarik untuk diikuti.

1.4 Batasan Masalah

Terdapat beberapa batasan masalah yang diangkat sebagai parameter pengerjaan penelitian ini di antaranya sebagai berikut :

- a. Aplikasi yang dibangun berbasis *smartphone android*.
- b. Aplikasi yang dibangun menggunakan algoritma *FAST Corner Detection*.
- c. Aplikasi ini digunakan untuk mengembangkan majalah *Binary* yang terdapat di Program Studi Sistem Informasi.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

a. Pendahuluan

Bab ini terdiri atas latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah dan sistematika penulisan.

b. Tinjauan Pustaka

Bab ini berisi tentang kajian materi, penelitian terdahulu dan informasi apa saja yang digunakan dalam penelitian ini. Dimulai dari kajian pustaka mengenai pengertian dari *Augmented Reality* hingga algoritma *FAST Corner Detection*.

c. Metodologi Penelitian

Bab ini berisi tentang metode apa yang digunakan selama penelitian. Dimulai dari tahap pencarian permasalahan hingga pengujian aplikasi *Binary Augmented Reality*.

d. Desain dan Perancangan Sistem

Bab ini berisi tahap perancangan dan desain yang akan digunakan untuk membuat aplikasi *Binary Augmented Reality*.

e. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari hasil penelitian yang telah dilakukan dengan menggambarkan hasil dari pengaplikasian algoritma *FAST Corner Detection* pada aplikasi *Binary Augmented Reality*.

f. Kesimpulan dan Saran

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan serta saran untuk penelitian selanjutnya.

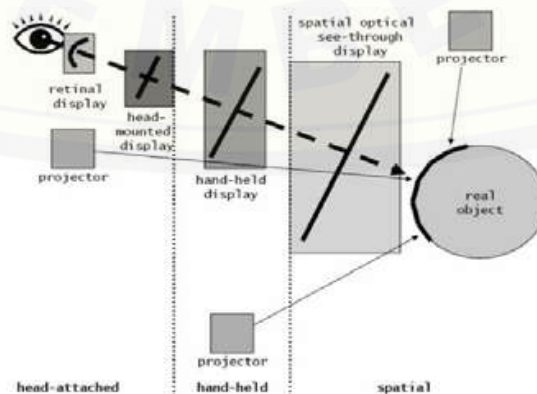
BAB 2 TINJAUAN PUSTAKA

2.1 Augmented Reality (AR)

Augmented Reality adalah teknologi baru yang melibatkan bagian dari komputer grafis dalam dunia nyata. Menurut Ronald Azuma, definisi dari AR adalah menggabungkan dunia nyata dan virtual, bersifat interaktif secara real time, dan merupakan animasi 3D (Azuma R. T., 2011). Paul Milgram dan Fumio Kishino pada tahun 1994, mendefinisikan Milgram's Reality-Virtuality Continuum. Mereka menggambarkan sebuah kontinum yang membentang dari lingkungan nyata untuk lingkungan virtual murni. Mereka menyimpulkan bahwa AR lebih dekat dengan dunia nyata dan augmented virtuality lebih dekat dengan dunia virtual (Milgram, Takemura, Utsumi, & Kishino, 1994).

2.2 Sistem Display Augmented Reality

Sistem *display* AR yakni sistem manipulasi citra yang pada dasarnya menggunakan seperangkat optik, elektronik, dan komponen mekanik untuk membentuk citra dalam jalur optik antara mata pengamat dan objek fisik yang akan digabungkan dengan teknik AR. Bergantung kepada optik yang digunakan, citra bisa dibentuk pada sebuah benda datar atau suatu bentuk permukaan yang kompleks (tidak datar). Jadi, objek yang ditampilkan AR nampak nyata dimata user.



Gambar 2-1 Pembentukan citra untuk *display augmented reality*

Gambar 2.1 mengilustrasikan kemungkinan citra akan dibentuk untuk mendukung AR, peletakan *display* bergantung dari pandangan pengguna dan objek, dan tipe citra seperti apa yang akan dihasilkan (*planar atau curved*) (Azuma R. T., 1997).

2.3 Citra Digital

Citra digital adalah pemrosesan gambar 2 dimensi menggunakan komputer. Suatu Citra dapat di definisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitude f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x,y, dan amplitude f secara keseluruhan berhingga (finite) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital.

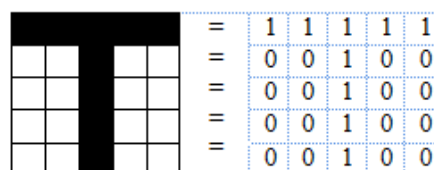
Citra digital dapat ditulis dalam bentuk matrik sebagai berikut (Putra, Binerisasi Citra Tangan dengan Metode Otsu, 2004)

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \dots\dots(1)$$

Berdasarkan jenisnya, citra digital dapat dibagi menjadi 3 (Sutoyo & dkk, 2009), yaitu:

2.3.1 Citra Biner (Monokrom)

Citra Biner hanya memiliki 2 buah warna, yaitu hitam dan putih. Warna hitam bernilai 1 dan warna putih bernilai 0. Untuk menyimpan kedua warna ini dibutuhkan 1 bit di memori. Contoh dari susunan piksel pada citra monokrom adalah gambar 2.2:



Gambar 2-2 Gambar susunan piksel pada citra monokrom

2.3.2 Citra *Grayscale*

Mempunyai kemungkinan warna hitam untuk nilai minimal dan warna putih untuk nilai maksimal. Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna tersebut. Semakin besar jumlah bit warna yang disediakan di memori, maka semakin halus gradasi warna yang terbentuk.

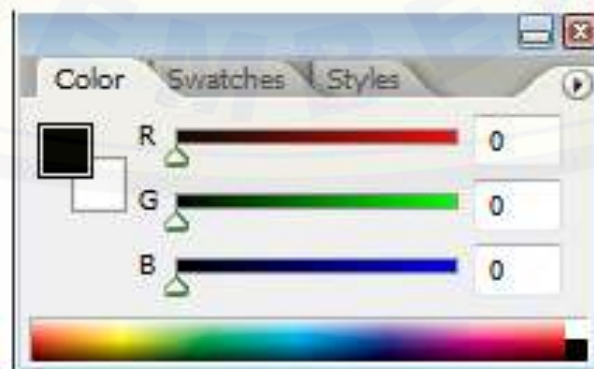


Gambar 2-3 Gradasi warna citra *grayscale*

Gambar 2.3 menunjukkan tingkat gradasi skala keabuan mulai dari tergelap sampai paling terang. Jika intensitas *grayscale* memiliki 8 bit, maka jumlah tingkatan *grayscale*-nya dapat dihitung $2^8 = 256$. Jadi, tingkatan *grayscale*-nya berjumlah 256 dengan level 0 sampai 255 dengan 0 merupakan intensitas tergelap dan 255 adalah intensitas paling terang.

2.3.3 Citra Warna (true color)

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi tiga warna dasar, yaitu merah, hijau, dan biru (RGB = Red, Green, Blue). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte (nilai maksimum 255 warna), jadi satu piksel pada citra warna diwakili oleh 3 byte.



Gambar 2-4 Citra Warna

Gambar 2.4 menunjukkan bahwa masing-masing warna RGB dapat di campur dengan warna lain sehingga dapat menampilkan warna lain selain 3 warna dasar tersebut.

2.4 Algoritma FAST Corner Detection

Corner Detection (deteksi sudut) merupakan suatu pendekatan yang digunakan pada suatu sistem visi komputer untuk mendeteksi sudut-sudut dari suatu objek. Nama lain dari *corner detection* adalah *interest point detection*. Pendekatan ini biasanya digunakan untuk mendeteksi suatu objek untuk menentukan kekhususan pada suatu objek misal dari segi bentuk objek dan pelacakan objek.

Adapun definisi dari sudut sendiri adalah perpotongan antara dua sisi (*edge*). Sebuah sudut juga dapat didefinisikan sebagai titik yang memiliki dua sisi dominan dan berbeda arah dari titik tersebut (Afissunani, Saleh, & Assidiqi, 2012). Dalam hal ini, sudut akan dimanfaatkan sebagai informasi sebuah image sehingga sebuah objek image dapat dengan mudah dikenali dengan menyematkan *corner point* pada titik minat (*interest point*) sebuah objek image.

FAST (Features from Accelerated Segment Test) Corner Detection merupakan algoritma penentuan *corner point* yang ditemukan oleh Edward Rosten. Pada *FAST Corner Detection*, proses penentuan *corner point*-nya adalah dengan cara merubah gambar menjadi warna hitam-putih dan menjalankan algoritmanya. Algoritma ini menentukan *corner point* dengan menentukan sebuah titik yakni p dari gambar masukan dengan 16 *pixel* disamping p diperiksa. Ada 3 kasus berbeda yang ditetapkan untuk masing-masing perbandingan yakni :

$$C = \begin{cases} |I_p - I_n| < t & \text{normal} \\ I_n - I_p > t & \text{brighter} \\ I_p - I_n > t & \text{darker} \end{cases} \dots\dots(2)$$

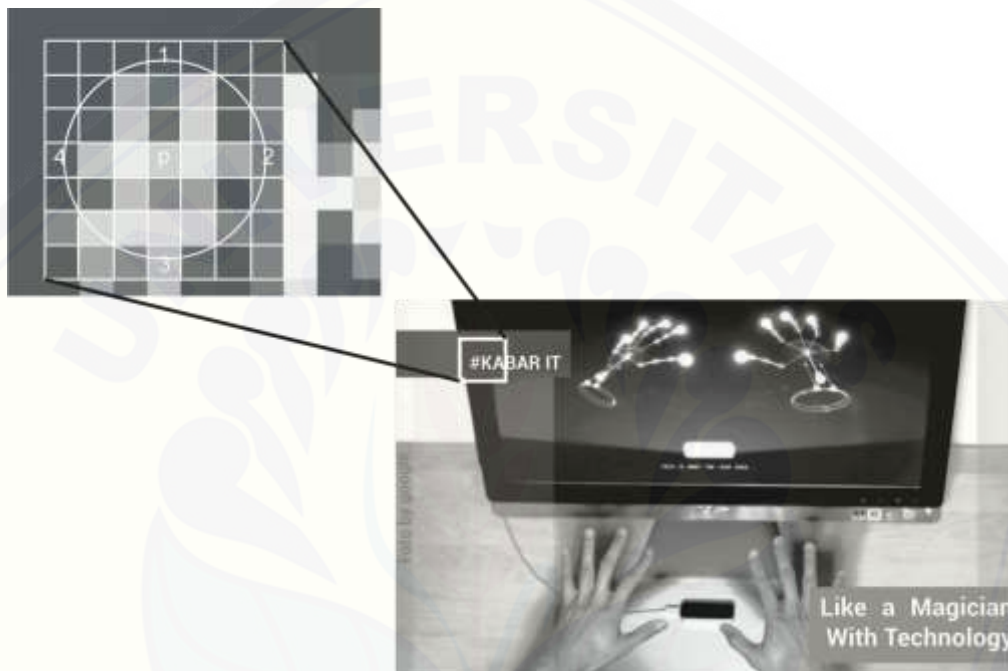
Keterangan :

I_p = intensitas titik pusat,

I_n = titik intensitas tetangga ke- n ,

t = *threshold*

Pada pengimplementasian *FAST Corner Detection* ini menggunakan jenis *high-speed test FAST Corner Detection* yang digunakan untuk mengecualikan jumlah besar dari *non-corners* (bukan sudut). Pengimplementasian algoritma ini dimulai dari penentuan titik p pada koordinat yang didefinisikan oleh koordinat (x_p, y_p) pada suatu digital. Setelah itu, intensitas titik p akan dibandingkan dengan 4 titik disekitarnya.



Gambar 2-5 Deteksi sudut pertitik FAST Corner Detection

Gambar 2.5 menunjukkan titik pertama yang akan dibandingkan adalah pada koordinat (x, y_{p-3}) . Titik kedua terletak pada koordinat (x_{p+3}, y) . Titik ketiga terletak pada koordinat (x, y_{p+3}) . Sedangkan titik ke empat terletak pada koordinat (x_{p-3}, y_p) . Titik p akan menjadi sudut (*corner*) apabila nilai intensitas titik p bernilai lebih besar atau lebih kecil daripada intensitas sedikitnya tiga titik disekitarnya. Proses ini tidak hanya berhenti disini, melainkan titik p akan digeser ke posisi selanjutnya (+1) dan titik tersebut akan dibandingkan lagi dengan 4 titik disekitarnya. Proses ini akan terjadi sampai semua titik pada citra digital dibandingkan.



Gambar 2-6 Diagram alir algoritma FAST Corner Detection

Gambar 2.6 merupakan diagram alir algoritma *FAST Corner Detection*. Adapun penjelasan tahap-tahap proses *FAST corner detection* adalah seperti berikut :

1. Menentukan sebuah titik p pada citra digital dengan posisi awal $(,)$.
2. Menentukan keempat titik seperti yang telah dijelaskan sebelumnya. Titik pertama $(n=1)$ terletak pada koordinat (-3) titik kedua $(n=2)$ terletak pada koordinat $(+3)$, titik ketiga terletak pada koordinat $(n=3)$ terletak pada koordinat (-3) , titik keempat $(n=4)$ terletak pada koordinat (-3)
3. Setelah keempat titik ditemukan, langkah selanjutnya bandingkan intensitas titikpusat p dengan keempat titik disekitar. Jika terdapat paling sedikit 3 titik yang memenuhi syarat berikut, maka titik pusat p .
4. Mengecek apakah masih ada pixels yang perlu dicek kembali. Jika tidak ada, hentikan proses. Jika ada, maka menaljutkan proses selanjutnya.
5. Mengulangi proses 1-3 sampai seluruh titik pada citra digital sudah dibandingkan intensitasnya. Gambar 5.11 merupakan gambaran dari proses 1-3.

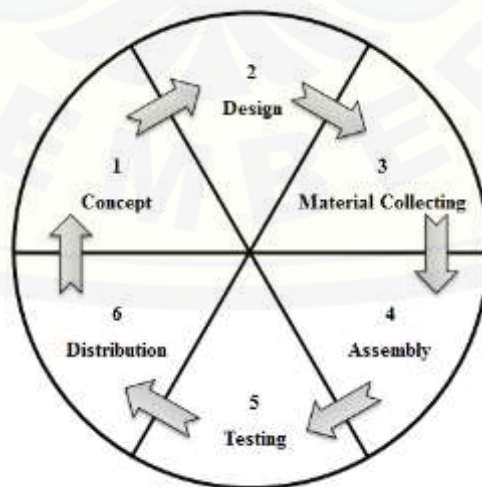
2.5 Vuforia Qualcomm

Vuforia Qualcomm merupakan *library* yang digunakan sebagai pendukung adanya *Augmented reality* pada *Android*. *Vuforia* menganalisa gambar dengan menggunakan pendeteksi *marker* maupun bukan *marker* dan menghasilkan informasi 3D dari *marker* yang sudah dideteksi via *API*. Programmer juga dapat menggunakannya untuk membangun objek 3D virtual pada kamera.

Vuforia Qualcomm menggunakan algoritma *FAST Corner Detection* sehingga mampu mengenali *marker* maupun *non-marker* secara lebih spesifik. Hal ini dapat kita lihat pada *vuforia* terdapat *feature*. Jika kita aktifkan *feature*-nya, maka akan terlihat image *marker* yang kita *upload* warnanya akan berubah menjadi *grayscale*. Selanjutnya image tersebut akan diproses kembali dengan menentukan *corner point*-nya.

2.6 Multimedia Development Life Cycle (MDLC)

MDLC merupakan sebuah model yang dikembangkan untuk digunakan pada proses pengembangan perangkat lunak khususnya perangkat lunak yang memiliki basis multimedia. Metodologi pengembangan multimedia terdiri dari 6 tahapan, yaitu *concept*(pengonsepan), *design*(pendesainan/perancangan), *material collecting*(pengumpulan materi), *assembly*(pembuatan), *testing*(pengujian) dan *distribution*(pendistribusian) (Ariesto, 2003) seperti gambar 2.7.



Gambar 2-7 Multimedia Development Life Cycle

Keenam tahap ini pada praktiknya tidak diharuskan dikerjakan secara berurutan, tahap-tahap tersebut bisa bertukar posisi. Akan tetapi, tahap *concept* harus dilakukan terlebih dahulu untuk menentukan tujuan suatu aplikasi ini akan dibuat.

Keterangan dari model MDLC :

1. *Concept* (Pengonsepan)

Tahap pengonsepan merupakan tahap untuk menentukan dasar-dasar dari proyek multimedia yang akan dibuat dan dikembangkan. Pada tahap ini yang harus dilakukan adalah memnentukan tujuan dan pengguna aplikasi multimedia yang akan dibuat. Adapun tujuan daripada tahap ini adalah mengukur kemampuan pengguna sehingga informasi yang disampaikan oleh aplikasi multimedia tersebut dengan mudah dapat tersampaikan kepada pengguna. Selain itu, hal yang harus dilakukan adalah menentukan jenis penelitian, tempat penelitian, alat penelitian dan lain-lain.

2. *Design*(pendesainan/perancangan)

Tahap pendesainan merupakan tahap pembuatan rancangan atau arsitektur program, tampilan, dan kebutuhan material yang akan dipakai pada tahap selanjutnya yaitu pengumpulan bahan. Oleh sebab itu biasanya kita dapat mengetahui apa saja yang nanti pada tahap selanjutnya akan dikumpulkan dengan mengetahui desain daripada aplikasi yang akan dibuat. Pada tahap ini biasanya terdapat *flowchart* dan *storyboard* untuk menggambarkan alir tautan dari scene satu ke scene lainnya.

3. *Material Collecting*(pengumpulan bahan)

Pengumpulan bahan merupakan tahap yang dilakukan setelah pendesainan. Pada tahap ini hal yang dilakukan adalah pengumpulan bahan-bahan yang akan dipakai pada tahap selanjutnya yakni pembuatan aplikasi. Bahan-bahan yang dikumpulkan bisa berupa informasi, gambar, objek 3D, video, audio dan lain-lain. Tahap ini dapat dikerjakan bersamaan dengan tahap selanjutnya karena ada kemungkinan bahan-bahan tambahan yang dibutuhkan pada saat pembuatan aplikasi. Akan tetapi hal ini tidak selalu terjadi tergantung dengan keadaan pada saat proses pembuatan aplikasi.

4. *Assembly*(pembuatan)

Assembly merupakan tahap yang dilakukan setelah desain dan pengumpulan bahan. Pada tahap ini hal yang dilakukan adalah menyatukan semua bahan-bahan yang telah dikumpulkan sebelumnya menjadi sebuah aplikasi berdasar pada desain yang telah dibuat sebelumnya yakni *storyboard* dan *flowchart*

5. *Testing*(pengujian)

Tahap pengujian merupakan tahap yang dilakukan setelah tahap pembuatan selesai dimana pengujian ini bertujuan untuk menguji aplikasi apakah masih terdapat kekurangan atau *bug*. Pengujian ini ada beberapa tahap yakni tahap *alpha*, pengujian tahap ini dilakukan oleh pembuat aplikasi. Setelah itu ada tahap *beta*, pengujian ini melibatkan *end-user*.

6. *Distribution*(pendistribusian)

Tahapan pendistribusian merupakan tahap evaluasi untuk sebuah produk, dalam hal ini aplikasi yang telah dibuat dengan mendistribusikan kepada pengguna. Pada tahap ini pembuat akan mengevaluasi apakah ada kekurangan maupun *bug* pada aplikasi yang telah dibuat. Tahap ini dapat digunakan sebagai masukan untuk tahap *concept* pada produk selanjutnya sehingga aplikasi selanjutnya yang dibuat atau diperbaiki akan menjadi lebih baik.

BAB 3 METODOLOGI PENELITIAN

Pada bab ini membahas tentang pengumpulan data dan perancangan yang digunakan untuk membangun aplikasi majalah interaktif *Binary Augmented Reality* dan pengaplikasian algoritma *FAST Corner Detection* untuk mendeteksi marker yang akan digunakan sebagai penanda objek yang akan ditampilkan. Adapun metodologi penelitiannya menggunakan metode MDLC (*Multimedia Development Life Cycle*). Peneliti memutuskan menggunakan metode ini dikarenakan metode ini merupakan metode yang cocok untuk penelitian ini terkait dengan basis penelitiannya yang menggunakan multimedia.

3.1 Pengonsepan

Pengonsepan diperlukan untuk menentukan tujuan utama daripada penelitian ini. Sebagaimana telah ditulis pada bab 1 tentang tujuan penelitian ini dibuat, maka berdasarkan hal ini dilakukan pembuatan sebuah aplikasi yang dapat mengimplementasikan AR, *markerless based tracking*, dan juga algoritma *FAST corner detection* pada majalah *binary*.

3.1.1 Jenis Penelitian

Pada penelitian ini digunakan dua jenis penelitian, yaitu penelitian kualitatif dan penelitian kuantitatif. Jenis penelitian kualitatif digunakan karena penelitian ini menganalisa studi literatur dan melakukan wawancara untuk pengumpulan data kriteria dan jenis penelitian kuantitatif digunakan karena dalam penelitian ini menerapkan serta mengkaji teori yang sudah ada sebelumnya.

3.1.2 Tempat Penelitian

Penelitian dilakukan di Program Studi Sistem Informasi Universitas Jember khususnya di Unit Kegiatan Mahasiswa (UKM) *Binary*.

3.1.3 Alat Penelitian

Alat yang digunakan dalam penelitian ini adalah hardware berupa satu unit laptop atau komputer yang didalamnya terdapat software sebagai berikut :

1. *Windows 8*
2. *Genymotion*
3. *Unity 5*
4. *Monodevelop*
5. *Vuforia SDK*
6. *Microsoft Office 2010*
7. *Corel Draw X4*
8. FileZilla
9. Browser

Sedangkan hardware lainnya adalah sebagai berikut :

1. Smartphone Android minimal Jelly Bean OS

3.1.4 Gambaran Umum Aplikasi

Aplikasi *Binary Augmented Reality* ini merupakan aplikasi *augmented reality* yang pada dasarnya memiliki 3 macam aksi untuk menampilkan objek. Gambaran umum dari ketiga aksi tersebut adalah sebagai berikut.

a. Memunculkan Objek 3D

Aplikasi ini memanfaatkan kamera sebagai media untuk menangkap gambar yang sudah dicetak pada majalah *binary*. Lalu gambar tersebut discan dan dicek apakah cocok dengan gambar yang sudah ditetapkan oleh algoritma *FAST Corner Detection*, jika cocok maka objek akan ditampilkan dengan cara merender objek yang akan ditampilkan terlebih dahulu sampai akhirnya objek ditampilkan pada layar monitor.

Algoritma *FAST Corner Detection* pada tahap ini mengacu pada proses proses deteksi marker. Algoritma ini digunakan pada tahap pemrosesan gambar yang telah diupload kedalam vuforia. Gambar tersebut akan dijadikan ketetapan suatu objek. Jadi, algoritma *FAST Corner Detection* akan menghasilkan titik-titik koordinat yang akan dideteksi dan disesuaikan dengan tangkapan gambar pada

kamera *augmented reality*. Koordinat-koordinat inilah yang nantinya menentukan keunikan setiap gambar pada objek-objek *augmented reality* yang berbeda.

b. Memutar *sound* atau musik

Proses ini hampir sama dengan memunculkan objek 3D, perbedaannya hanya terletak pada proses pemanggilannya. Ketika gambar telah terdeteksi dan cocok, maka langkah selanjutnya adalah memutar *sound* atau music yang telah ada pada aplikasi ini. Begitu juga ketika gambar tidak terdeteksi lagi, maka *sound* atau music akan berhenti dimainkan.

c. Menampilkan video

Pada proses ini, sama persis dengan proses memutar sound. Ketika gambar terdeteksi dan cocok, maka langkah selanjutnya adalah aplikasi memunculkan video yang sesuai dengan gambar yang terdeteksi.

3.2 Desain dan Perancangan Aplikasi

Desain dibangun menggunakan beberapa teknik desain aplikasi yakni *flowchart* dan *storyboard*. Teknik ini digunakan pada aplikasi multimedia ini dengan tujuan mempermudah *developer* aplikasi dalam pembuatan aplikasi.

a. *Flowchart*

Flowchart merupakan gambaran dari aplikasi yang menampilkan alur dari suatu aksi ke aksi lain. Pada setiap tahapnya terdapat penjelasan sehingga akan ditemukan urutan penyusunan aplikasi yang dibuat.

b. *Storyboard*

Storyboard merupakan suatu gambaran dari skenario setiap *scene* pada aplikasi dengan kata lain *storyboard* merupakan visualisasi ide dari aplikasi yang dibangun.

3.3 Pengumpulan Bahan

Mengumpulkan bahan-bahan yang diperlukan untuk pembangunan aplikasi. Diantaranya terdapat gambar 2D, video, audio dan objek 3D.

a. Gambar 2D

Gambar 2 dimensi akan digunakan sebagai *marker augmented reality*. Adapun format gambarnya berupa gambar berekstensi JPG.

b. Video

Video akan diupload pada *cloud* sehingga nanti dapat diakses oleh aplikasi pada saat dibutuhkan. Adapun format video yang akan dipakai berekstensi mp4.

c. Audio

Audio akan digunakan sebagai objek pada *augmented reality* pada saat kamera mendeteksi *marker* yang sesuai dengan konfigurasi penempatan *marker* dan *audio*. Adapun format audio yang akan dipakai berekstensi mp3.

d. Objek 3D

Objek 3 dimensi akan digunakan sama halnya dengan audio. Perbedaannya adalah objek 3 dimensi akan muncul pada layar monitor sesuai dengan *marker* yang dideteksi oleh kamera. Aplikasi akan menyesuaikan objek 3 dimensi seolah-olah berada di dunia nyata. Adapun format objek 3 dimensi yang akan dipakai berekstensi fbx.

3.4 Implementasi Pembuatan Aplikasi

Tahap ini merupakan tahap implementasi desain ke dalam kode program yang akan menjadi aplikasi. Bahasa pemrograman yang digunakan adalah bahasa pemrograman C#. Pada pembuatan aplikasi ini terdapat software pembantu yang digunakan untuk membentuk aplikasi yang dapat berjalan di platform smartphone android yakni Engine Unity 3D yang digunakan untuk mengedit dan mensetting letak objek output pada AR dengan bantuan 2 library. Library yang digunakan adalah Vuforia SDK dan Android SDK. Vuforia SDK digunakan untuk membaca marker pada kamera smartphone android serta membantu untuk memberikan behavior pada aplikasi AR. Sedangkan android SDK digunakan untuk membantu proses building aplikasi Binary Augmented Reality sesuai dengan versi android yang akan digunakan saat ini.

Gambar yang digunakan adalah markerless sehingga gambar yang digunakan diharuskan memiliki banyak perbedaan warna dan sudut-sudut perbedaan warna yang banyak. Gambar tersebut sebelumnya diupload kedalam website vuforia untuk dicari *corner*-nya dimana vuforia dalam prosesnya menggunakan algoritma *FAST Corner Detection*. Setelah gambar diproses baru kemudian diexport dalam bentuk package yang akan dikolaborasikan dengan package vuforia yang sebelumnya telah diimport terlebih dahulu.

3.5 Pengujian Aplikasi

Pengujian aplikasi merupakan tahap ujicoba terhadap aplikasi dengan tujuan memeriksa kembali aplikasi yang telah dibuat apakah terdapat kesalahan. Apabila terdapat kesalahan, aplikasi akan diperbaiki untuk memperbaiki dan menyempurnakan aplikasi yang telah dibuat. Pengujian ini akan menggunakan metode *blackbox* dan *whitebox*. Pengujian *whitebox* adalah pengujian pada modul pengkodean program yang dilakukan oleh peneliti. Pengujian ini dilakukan dengan menghitung *independent path* dengan menggunakan *cyclomatic complexity*. Untuk menghitung *cyclomatic complexity* digunakan rumus dibawah ini:

$$V(G) = E - N + 2 \dots \dots \dots (3)$$

Keterangan :

V(G) = Jumlah Minimum path yang kita perlukan

E = jumlah edge grafik alir

N = jumlah node grafik alir

Pengujian *blackbox* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Pengetahuan khusus dari kode aplikasi atau struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Metode ini memfokuskan pada keperluan fungsionalitas dari *software* (Rusadi, 2013).

BAB 4 DESAIN DAN PERANCANGAN APLIKASI

Pada bab ini menguraikan tentang proses prancangan untuk mengimplementasikan algoritma *FAST Corner Detection* pada aplikasi *Augmented Reality* majalah *Binary*. Proses perancangan sistem dimulai dari analisis kebutuhan fungsional dan non fungsional sistem, dilanjutkan dengan diagram alir dan *storyboard*.

4.1 Analisis Kebutuhan Perangkat Lunak

Aplikasi yang akan dirancang adalah aplikasi yang memiliki fungsi untuk menangkap *image marker* sebagai *trigger* untuk menampilkan objek-objek tertentu yang sudah *disetting*. Adapun didalam perancangan aplikasi ini terdapat batasan-batasan. Aplikasi yang akan dibuat ini merupakan media untuk membuat sebuah majalah lebih interaktif dan menarik daripada hanya sekedar dapat dibaca.

Adapun metode yang digunakan untuk membuat markerless tersebut adalah *FAST Corner Detection*. Metode ini digunakan untuk mendapatkan sudut-sudut koordinat pada sebuah gambar untuk dijadikan marker. Kelebihan daripada metode ini adalah penggunaannya mudah karena sudah terdapat library yang mendukung yakni Vuforia serta dapat mempercepat analisa gambar yang sudah ditetapkan sebagai marker.

Analisis kebutuhan yang dilakukan adalah mengidentifikasi permasalahan yang ada dan dijadikan bahan untuk mulai membangun aplikasi *Augmented Reality Binary*. Adapun hal yang dibutuhkan disini meliputi data kebutuhan fungsional dan kebutuhan non-fungsional.

Kebutuhan fungsional aplikasi ini yaitu:

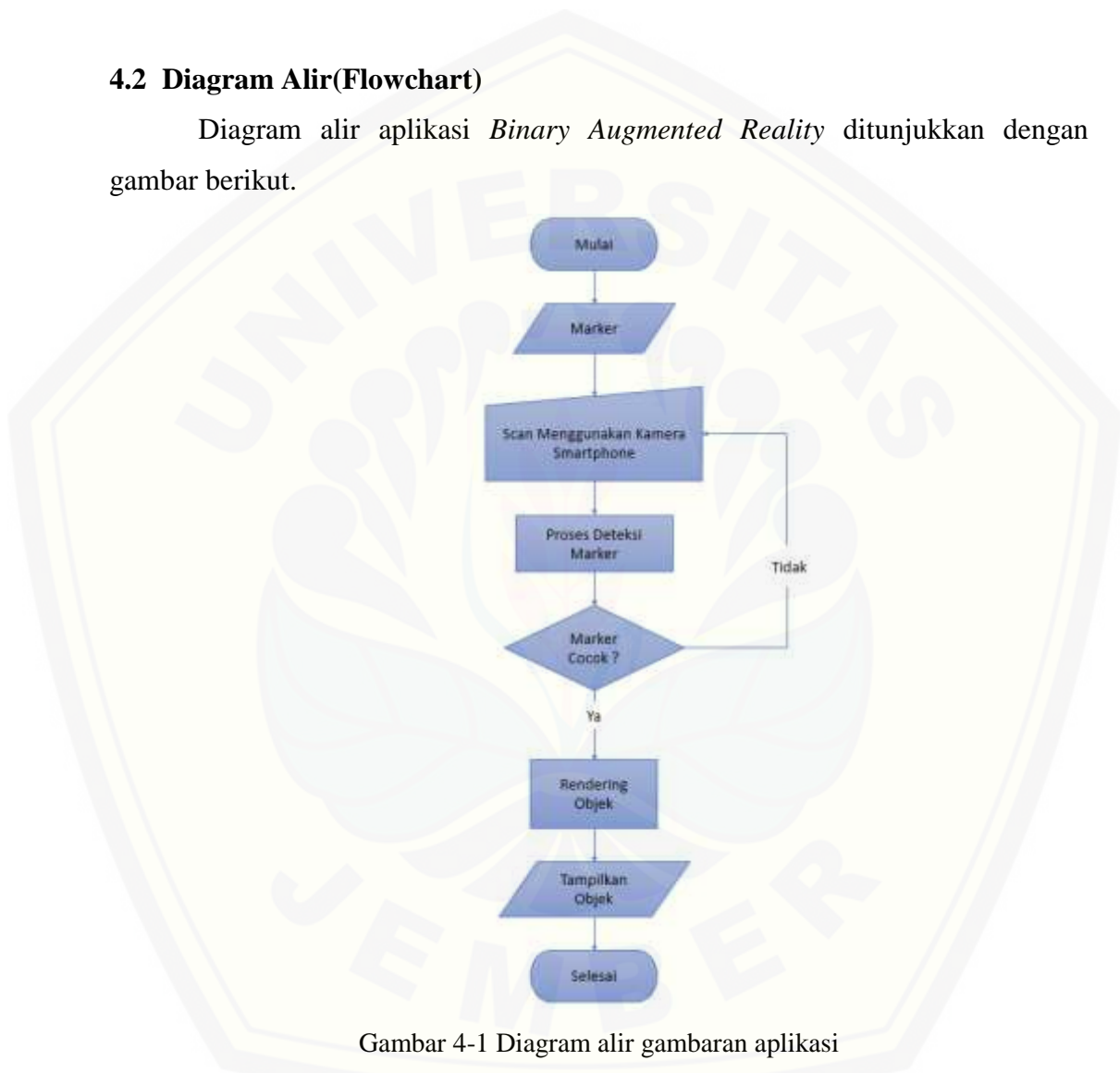
1. Aplikasi menggunakan algoritma *FAST Corner Detection* untuk mengidentifikasi objek gambar 2 dimensi yang dijadikan sebagai marker.
2. Aplikasi dapat menampilkan beberapa objek multimedia seperti objek 3 dimensi, objek video dan objek musik.
3. Aplikasi dapat menampilkan objek video secara online.

Sedangkan kebutuhan non-fungsional aplikasi ini yakni:

1. Pengembangan aplikasi *Augmented Reality Binary* dibuat *user friendly* agar menarik bagi pengguna aplikasi.
2. Aplikasi dapat berjalan optimal pada smartphone android saja.
3. Aplikasi dapat mengupdate objek 3D, video dan music.

4.2 Diagram Alir(Flowchart)

Diagram alir aplikasi *Binary Augmented Reality* ditunjukkan dengan gambar berikut.

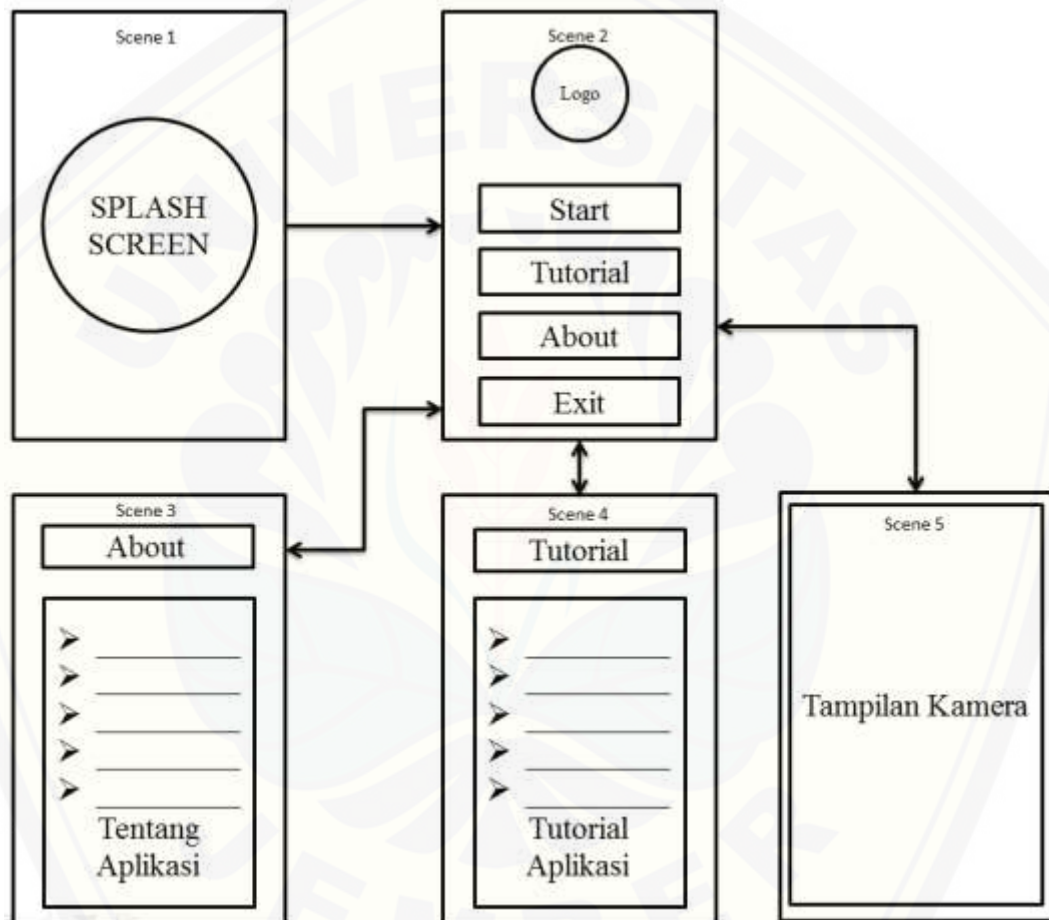


Gambar 4-1 Diagram alir gambaran aplikasi

Gambar 4.1 menunjukkan diagram alir dari alir kerja gambaran aplikasi ini. Diagram alir ini menggambarkan proses kerja pada menu utama saat aplikasi menggunakan kamera sebagai alat bantu mencari marker. Mulai dari inputan marker sampai menampilkan objek. Pertama-tama *marker* akan di-*scan* menggunakan kamera pada *smartphone* setelah itu, *marker* akan diproses apakah sesuai dengan *marker* yang sudah dikonfigurasi sebelumnya pada saat pembuatan

aplikasi di *Game Engine Unity 3D*. Setelah itu dilakukan pengecekan dengan algoritma *FAST Corner Detection*. Jika cocok, maka aplikasi akan merender objek yang akan ditampilkan dan menampilkan objek yang sesuai dengan marker tersebut. Tapi jika tidak cocok, maka aplikasi akan kembali men-*scan marker* kembali.

4.3 Storyboard



Gambar 4-2 Storyboard aplikasi Binary Augmented Reality

Gambar 4.2 menunjukkan *storyboard* dan alir *scene* mulai dari *scene 1* sampai *scene 5*. Aplikasi ini hanya memiliki 5 *scene* akan tetapi aplikasi ini memiliki *scene* paling utama yakni pada *scene 4* karena di *scene* inilah *augmented reality* diimplementasikan. Keterangan dari *scene* tersebut adalah sebagai berikut :

1. *Scene 1*, *scene* tersebut adalah *scene splash screen* untuk memunculkan logo awal dari aplikasi ini.

2. *Scene 2*, *scene* ini merupakan *scene* untuk menampilkan menu-menu yang terdapat pada aplikasi ini.
3. *Scene 3*, *scene* ini adalah *scene* untuk menampilkan informasi tentang aplikasi ini.
4. *Scene 4*, *scene* ini adalah *scene* untuk menampilkan tutorial cara penggunaan aplikasi ini.
5. *Scene 5*, *scene* ini adalah *scene* utama yang digunakan untuk men-*scan* objek 2 dimensi dengan menggunakan kamera. Tampilan pada *scene* ini berupa gambar yang ditangkap oleh kamera dan objek *output* disesuaikan dengan *marker*.

4.4 Implementasi Perancangan

Tahap ini dilakukan setelah tahap desain perancangan selesai kedalam bahasa pemrograman. Bahasa pemrograman yang digunakan adalah bahasa pemrograman C#. Sedangkan objek yang dipersiapkan adalah video yang menyangkut tentang materi majalah binary dengan format mp4, objek 3D dengan format fbx dan musik dengan format mp3. Pada pembuatan aplikasi ini terdapat software pembantu yang digunakan untuk membentuk aplikasi yang dapat berjalan di platform smartphone android yakni Engine Unity 3D yang digunakan untuk mengedit dan mensetting letak objek output pada AR dengan bantuan 2 library. Library yang digunakan adalah Vuforia SDK dan Android SDK. Vuforia SDK digunakan untuk membaca marker pada kamera smartphone android serta membantu untuk memberikan behavior pada aplikasi AR. Sedangkan android SDK digunakan untuk membantu proses building aplikasi Binary Augmented Reality sesuai dengan versi android yang akan digunakan saat ini.

Gambar yang digunakan adalah markerless sehingga gambar yang digunakan diharuskan memiliki banyak perbedaan warna dan sudut-sudut perbedaan warna yang banyak. Gambar tersebut sebelumnya diupload kedalam website vuforia untuk dicari *corner*-nya dimana vuforia dalam prosesnya menggunakan algoritma FAST *Corner Detection*. Setelah gambar diproses baru

kemudian diexport dalam bentuk package yang akan dikolaborasikan dengan package vuforia yang sebelumnya telah diimport terlebih dahulu.

4.9 Pengujian Blackbox dan Whitebox

Pengujian adalah proses untuk menemukan error pada perangkat lunak sebelum dikirim kepada pengguna. Pengujian Software adalah kegiatan yang ditujukan untuk mengevaluasi atribut atau kemampuan program dan memastikan bahwa itu memenuhi hasil yang dicari, atau suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji (under test). Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya (Rouf, Pengujian Perangkat Lunak Dengan Menggunakan Metode White Box dan Black Box, 2012). Pengujian pada penelitian ini dilakukan dengan 2 cara yakni pengujian dengan cara white box dan black box. Pengujian white box merupakan pengujian yang menggunakan test case yang digambarkan dengan notasi diagram alir. Sedangkan pengujian black box merupakan pengujian yang dilakukan dengan cara menguji aspek fundamental sistem tanpa melihat struktur logika aplikasi/software. Metode ini digunakan untuk mengetahui aplikasi/software dapat berfungsi dengan benar sesuai dengan rancangan aplikasi yang sebelumnya telah ditetapkan.

4.9.1 Pengujian Whitebox

Pengujian white box merupakan pengujian dengan cara menghitung kompleksitas sistem yang telah dibangun dan mengecek sistem berdasarkan alur perancangan yang telah dibuat sebelumnya. Pengujian white box pada aplikasi Binary Augmented Reality dilakukan dengan menggunakan metode independent path yakni metode yang menggunakan cyclomatic complexity. Listing aplikasi Binary Augmented Reality dapat dilihat pada lampiran A.

4.9.2 Pengujian Blackbox

Pada pengujian tahap ini yang dilakukan adalah pengujian pada aspek fundamental aplikasi Binary Augmented Reality tanpa memperhatikan struktur

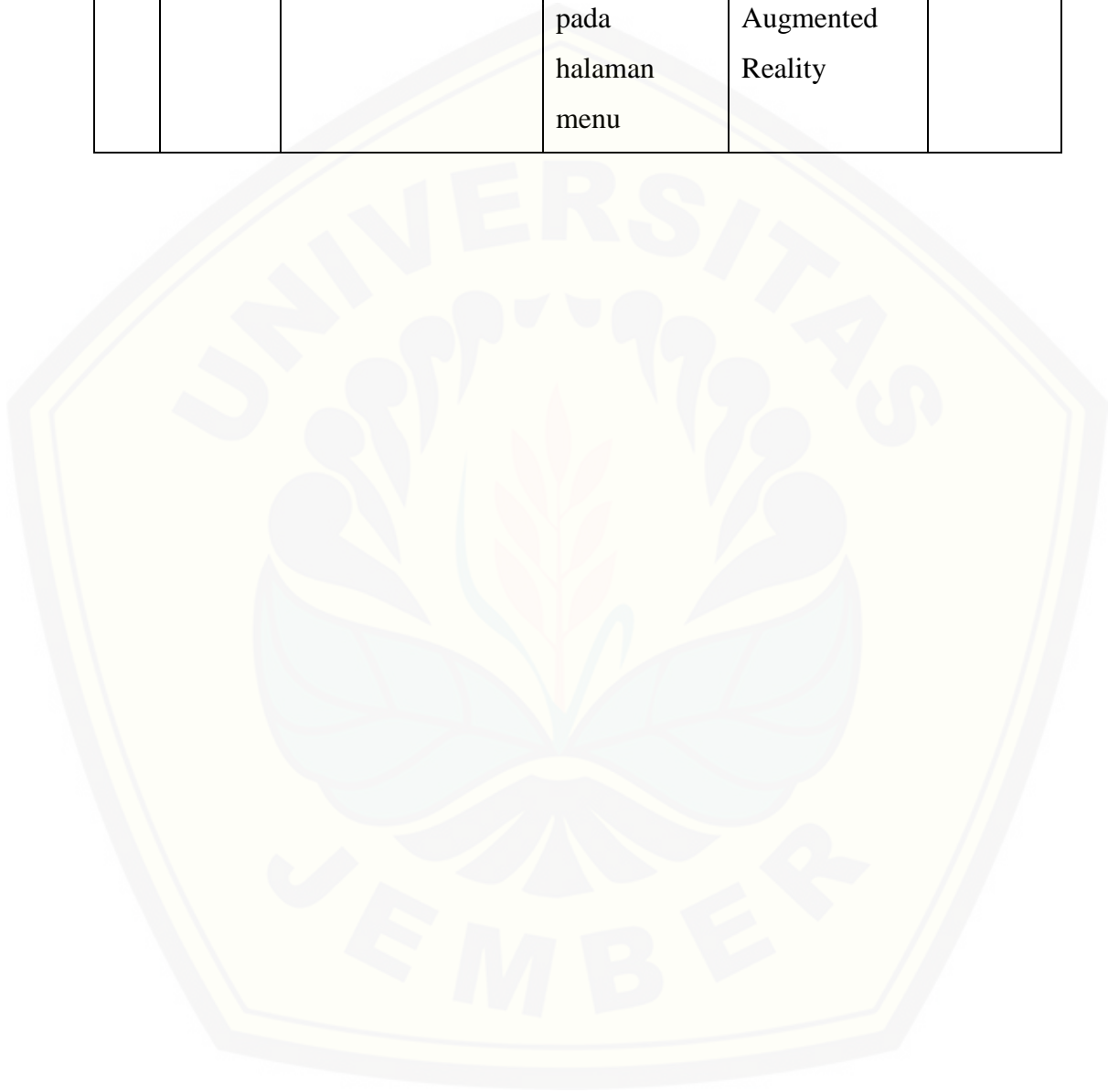
logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah aplikasi Binary Augmented Reality berfungsi dengan benar. Pengujian ini merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dicobakan kemudian dieksekusi pada perangkat lunak. Kemudian dari perangkat lunak dicek kembali apakah fungsi sudah sesuai dengan yang diharapkan.

Tabel 4.1 Tabel Pengujian Blackbox

No	Menu	Fungsi	Kasus	Hasil	Ket
1.	Start	Untuk menampilkan kamera yang memiliki fungsi untuk menangkap gambar marker kemudian menampilkan objek 3D, memutar music dan memutar video	User menyentuh menu start pada halaman menu	Menampilkan halaman start yang menampilkan tangkapan visualisasi dari kamera	OK
			User mengarahkan kamera ke gambar marker yang menampilkan objek 3D	Menampilkan objek 3D dari gambar marker	OK
			User mengarahkan kamera ke gambar marker yang	Memutar music dari gambar marker	OK

			dapat mentrigger aplikasi memutar musik		
			User mengarahkan kamera ke gambar marker yang menampilkan video	Menampilkan gambar latar 2D dan icon X atau play dari objek video	OK
			User menyentuh icon play pada video	Memutar video	OK
2.	Tutorial	Untuk menginformasikan kepada user tentang tata cara penggunaan aplikasi Binary Augmented Reality	User menyentuh menu tutorial pada halaman menu	Menampilkan tata cara penggunaan aplikasi Binary Augmented Reality	OK
3.	About	Untuk menjelaskan kepada user tentang informasi-informasi mengenai aplikasi Binary Augmented Reality mulai dari versi, author dan	User menyentuh menu about pada halaman menu	Menampilkan informasi mengenai aplikasi Binary Augmented Reality	OK

		sebagainya.			
4.	Exit	Untuk keluar dari aplikasi	User menyentuh tombol exit pada halaman menu	Keluar dari aplikasi Binary Augmented Reality	OK



BAB 6

PENUTUP

Bab ini adalah bagian akhir dari penulisan skripsi yang didalamnya berisi tentang kesimpulan dan saran. Adapun kesimpulan merupakan hasil dari penelitian dan jawaban dari rumusan masalah. Sedangkan saran merupakan sesuatu yang belum ditempuh atau dilaksanakan untuk mengembangkan penelitian tentang algoritma ini.

6.1 Kesimpulan

Kesimpulan dari aplikasi yang telah dibangun dalam penelitian ini adalah sebagai berikut :

1. Pengimplementasian teknologi augmented reality pada aplikasi ini dapat berjalan sesuai dengan perancangan awal yakni dapat menampilkan objek tiga dimensi, video dan musik dengan bantuan library vuforia SDK.
2. Penerapan metode MDLC dapat memberi manfaat yang baik bagi penulis mulai dari tahap pertama yakni analisis kebutuhan sampai pada maintenance aplikasi. Hal ini dikarenakan MDLC ini dilakukan secara bertahap dan terstruktur. Sehingga jika terjadi kesalahan dalam salah satu tahap, maka perbaikan akan cepat dapat dilakukan.
3. Aplikasi Binary Augmented Reality dibangun dengan menggunakan Game Engine Unity 3D dengan bantuan Vuforia SDK. Aplikasi ini dapat menampilkan secara real-time objek 3D, music dan video pada saat kamera menemukan marker yang sesuai.
4. Algoritma high speed test *FAST Corner Detection* dapat secara cepat menentukan *corner* dari suatu gambar yang dijadikan marker dengan beberapa melakukan image processing terlebih dahulu.
5. Pendeteksian marker dengan menggunakan algoritma high speed test *FAST Corner Detection* dapat mengenali suatu gambar yang ditangkap kamera dengan cepat dan baik. Sehingga dapat digunakan secara real-time.

6.2 Saran

Penelitian ini memiliki hasil yang belum sempurna. Oleh karena itu diperlukan pengembangan agar dapat meningkatkan hasil yang bisa dicapai. Adapun saran untuk mengembangkan penelitian ini adalah sebagai berikut :

1. Diharapkan algoritma *FAST Corner Detection* ini dapat diimplementasikan ke lebih banyak objek yang bersangkutan.
2. Pada penelitian ini, algoritma *FAST Corner Detection* belum terdapat penghitungan manual untuk mengetahui lebih dalam manfaat nyata dari penggunaan algoritma ini untuk mengenali suatu objek dua dimensi secara cepat.
3. Aplikasi yang sudah dibuat dapat dikembangkan ke platform lainnya seperti Windows Phone, iOS, desktop maupun perangkat lainnya.

DAFTAR PUSTAKA

- Afissunani, A., Saleh, A., & Assidiqi, M. H. (2012). MULTI MARKER AUGMENTED REALITY UNTUK APLIKASI MAGIC BOOK . *Engineering*, 1-4.
- Ariesto, H. S. (2003). *Multimedia Interaktif dan Flash*. Yogyakarta: PT Graha Ilmu.
- Azuma, R. T. (1997). Presence: Teleoperators and Virtual. *A Survey of Augmented Reality*, 355-385.
- Azuma, R. T. (2011). Indirect Augmented Reality. *Computers and Graphics*, 810-822.
- Basuki, A. (2005). *Pengolahan Citra Digital Menggunakan Visual Basic*. Yogyakarta: Graha Ilmu.
- Fegyver, Z. (2013, November 19). *Experiment on the FAST corner detector*. Dipetik Mei 18, 2015, dari matlabtricks: <http://matlabtricks.com/post-27/experiment-on-the-fast-corner-detector>
- Kim, Y.-g., & Kim, W.-j. (2014). Implementation of Augmented Reality System for Smartphone Advertisements. *Journal of Multimedia and Ubiquitous Engineering*, 385-392.
- Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1994). Augmented Reality: A class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies*, 282-292.
- Putra, D. (2004). Binerisasi Citra Tangan dengan Metode Otsu. *Jurusan Teknik Elektro*, 11-13.
- Rosa, & Salahudin, M. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak*. Bandung: Modula.
- Rosten, E. (2006, May). *FAST Corner Detection -- Edward Rosten*. Dipetik Mei 17, 2015, dari edwardrosten: <http://www.edwardrosten.com/work/fast.html>
- Rouf, A. (2012). Pengujian Perangkat Lunak Dengan Menggunakan Metode White Box dan Black Box. *Jurnal Teknologi Informasi*, 1-7.

Rusadi, W. A. (2013, May 1). *White Box Testing & Black Box Testing*. Retrieved January 8, 2015, from Bangwildan: bangwildan.web.id/berita-176-white-box-testing--black-box-testing.html

Sommerville, I. (2011). *Software Engineering 9th Edition*. United States of America: Addison-Wesley Publishing Company Inc.

Sutoyo, T., & dkk. (2009). *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.



LAMPIRAN

Lampiran A. White Box Testing

A. Listing Program

1. Menampilkan Objek3D

a. OnTrackableStateChanged

```

46     public void OnTrackableStateChanged(
47         TrackableBehaviour.Status previousStatus,
48         TrackableBehaviour.Status newStatus)
49     {
50         if (newStatus == TrackableBehaviour.Status.DETECTED ||
51             newStatus == TrackableBehaviour.Status.TRACKED ||
52             newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
53         {
54             OnTrackingFound();
55             sound.Play();
56         }
57         else
58         {
59             OnTrackingLost();
60             sound.Stop();
61         }
62     }

```

b. OnTrackingFound

```

71     private void OnTrackingFound()
72     {
73         Renderer[] rendererComponents = GetComponentsInChildren<Renderer>(true);
74         Collider[] colliderComponents = GetComponentsInChildren<Collider>(true);
75
76         // Enable rendering:
77         foreach (Renderer component in rendererComponents)
78         {
79             component.enabled = true;
80         }
81
82         // Enable colliders:
83         foreach (Collider component in colliderComponents)
84         {
85             component.enabled = true;
86         }
87
88         Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");
89     }

```

c. OnTrackingLost

```

92     private void OnTrackingLost()
93     {
94         Renderer[] rendererComponents = GetComponentsInChildren<Renderer>(true);
95         Collider[] colliderComponents = GetComponentsInChildren<Collider>(true);
96
97         // Disable rendering:
98         foreach (Renderer component in rendererComponents)
99         {
100             component.enabled = false;
101         }
102
103         // Disable colliders:
104         foreach (Collider component in colliderComponents)
105         {
106             component.enabled = false;
107         }
108
109         Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " lost");
110     }

```

2. Memutar Musik

Listing program pada memutar music sama dengan listing program menampilkan objek 3D.

3. Memutar video

a. OnTrackableStateChanged

```

71     public void OnTrackableStateChanged(
72         TrackableBehaviour.Status previousStatus,
73         TrackableBehaviour.Status newStatus)
74     {
75         if (newStatus == TrackableBehaviour.Status.DETECTED ||
76             newStatus == TrackableBehaviour.Status.TRACKED ||
77             newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
78         {
79             OnTrackingFound();
80         }
81         else
82         {
83             OnTrackingLost();
84         }
85     }

```

b. OnTrackingFound

```

93 | private void OnTrackingFound()
94 | {
95 |     Renderer[] rendererComponents = GetComponentInChildren<Renderer>();
96 |     Collider[] colliderComponents = GetComponentInChildren<Collider>();
97 |
98 |     // Enable rendering:
99 |     foreach (Renderer component in rendererComponents)
100 |     {
101 |         component.enabled = true;
102 |     }
103 |
104 |     // Enable colliders:
105 |     foreach (Collider component in colliderComponents)
106 |     {
107 |         component.enabled = true;
108 |     }
109 |
110 |     Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");
111 |
112 |     // Optionally play the video automatically when the target is found
113 |
114 |     VideoPlaybackBehaviour video = GetComponentInChildren<VideoPlaybackBehaviour>();
115 |     if (video != null && video.AutoPlay)
116 |     {
117 |         if (video.VideoPlayer.IsPlayableOnTexture())
118 |         {
119 |             VideoPlayerHelper.MediaState state = video.VideoPlayer.GetStatus();
120 |             if (state == VideoPlayerHelper.MediaState.PAUSED ||
121 |                 state == VideoPlayerHelper.MediaState.READY ||
122 |                 state == VideoPlayerHelper.MediaState.STOPPED)
123 |             {
124 |                 // Pause other videos before playing this one
125 |                 PauseOtherVideos(video);
126 |
127 |                 // Play this video on texture where it left off
128 |                 video.VideoPlayer.Play(false, video.VideoPlayer.GetCurrentPosition());
129 |             }
130 |             else if (state == VideoPlayerHelper.MediaState.REACHED_END)
131 |             {
132 |                 // Pause other videos before playing this one
133 |                 PauseOtherVideos(video);
134 |
135 |                 // Play this video from the beginning
136 |                 video.VideoPlayer.Play(false, 0);
137 |             }
138 |         }
139 |     }
140 |
141 |     mHasBeenFound = true;
142 |     mLostTracking = false;
143 | }

```

c. OnTrackingLost

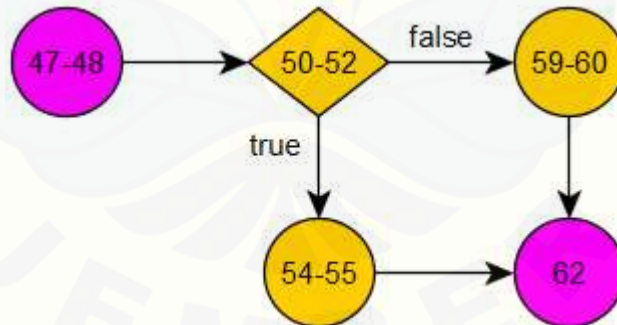
```

146 private void OnTrackingLost()
147 {
148     Renderer[] rendererComponents = GetComponentInChildren<Renderer>();
149     Collider[] colliderComponents = GetComponentInChildren<Collider>();
150
151     // Disable rendering:
152     foreach (Renderer component in rendererComponents)
153     {
154         component.enabled = false;
155     }
156
157     // Disable colliders:
158     foreach (Collider component in colliderComponents)
159     {
160         component.enabled = false;
161     }
162
163     Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " lost");
164
165     mLostTracking = true;
166     mSecondsSinceLost = 0;
167 }
    
```

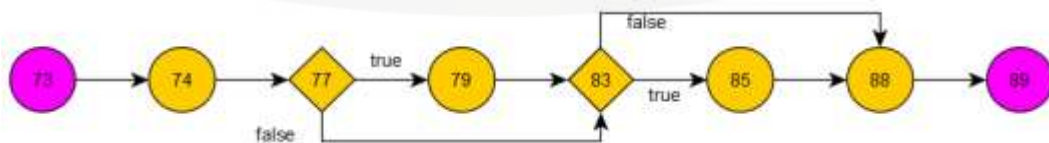
B. Diagram Alir

1. Menampilkan Objek 3D

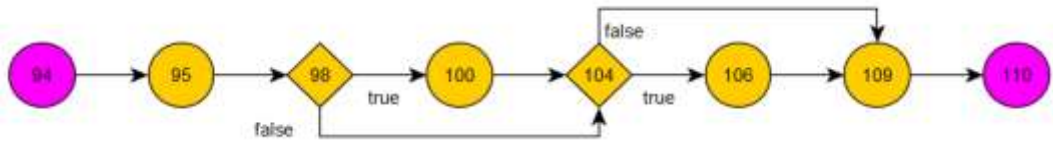
a. OnTrackableStateChanged



b. OnTrackingFound



c. OnTrackingLost

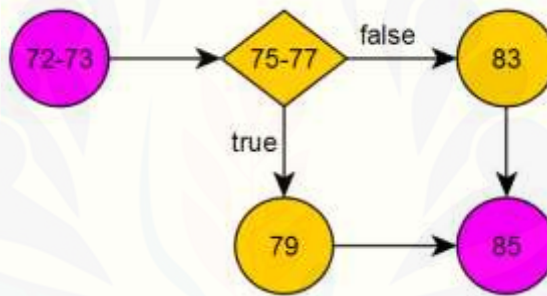


2. Memutar Musik

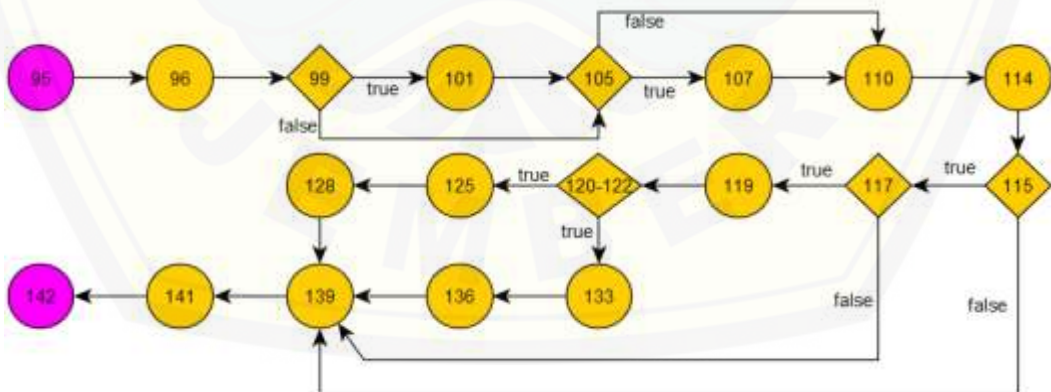
Diagram Alir pada memutar music sama dengan diagram alir menampilkan objek 3D.

3. Memutar Video

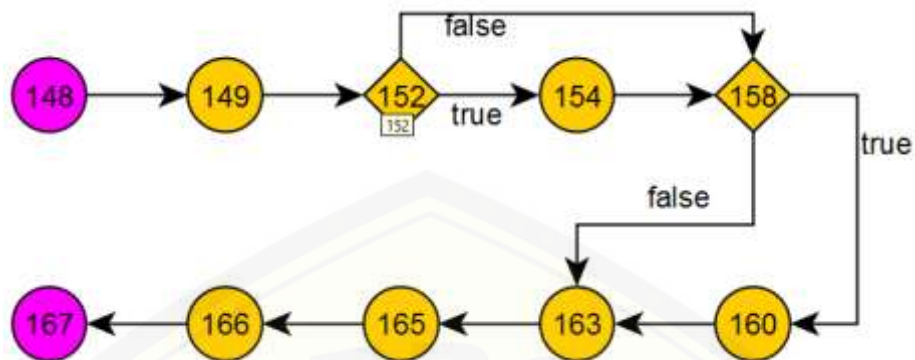
a. OnTrackableStateChanged



b. OnTrackingFound



c. OnTrackingLost



C. Cyclomatic Complexity

1. Menampilkan Objek 3D

a. OnTrackableStateChanged

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

$$V(G) = 5 - 5 + 2$$

$$V(G) = 2$$

b. OnTrackingFound

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

$$V(G) = 9 - 8 + 2$$

$$V(G) = 3$$

c. OnTrackingLost

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

$$V(G) = 9 - 8 + 2$$

$$V(G) = 3$$

2. Memutar Musik

Penghitungan Cyclomatic Complexity(CC) pada saat memutar music sama dengan penghitungan CC Objek 3D

3. Memutar Video

a. OnTrackableStateChanged

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

$$V(G) = 5 - 5 + 2$$

$$V(G) = 2$$

b. OnTrackingFound

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

$$V(G) = 23 - 19 + 2$$

$$V(G) = 6$$

c. OnTrackingLost

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

$$V(G) = 11 - 10 + 2$$

$$V(G) = 3$$

D. Independent Path

1. Menampilkan Objek 3D

a. OnTrackableStateChanged

$$\text{Jalur 1} = 1-2-3-5$$

$$\text{Jalur 2} = 1-2-4-5$$

b. OnTrackingFound

$$\text{Jalur 1} = 1-2-3-4-5-6-7-8$$

$$\text{Jalur 3} = 1-2-3-4-5-7-8$$

$$\text{Jalur 4} = 1-2-3-5-6-7-8$$

c. OnTrackingLost

$$\text{Jalur 1} = 1-2-3-4-5-6-7-8$$

$$\text{Jalur 3} = 1-2-3-4-5-7-8$$

$$\text{Jalur 4} = 1-2-3-5-6-7-8$$

2. Memutar music

Independent Path memutar music sama dengan Independent Path
Menampilkan objek 3D.

3. Memutar Video

a. OnTrackableStateChanged

Jalur 1 = 1-2-3-5

Jalur 2 = 1-2-4-5

b. OnTrackingFound

Jalur 1 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17

Jalur 2 = 1-2-3-5-6-7-8-9-10-11-12-13-14-15-16-17

Jalur 3 = 1-2-3-4-5-7-8-9-10-11-12-13-14-15-16-17

Jalur 4 = 1-2-3-4-5-6-7-8-9-15-16-17

Jalur 5 = 1-2-3-4-5-6-7-8-9-10-15-16-17

Jalur 6 = 1-2-3-4-5-6-7-8-9-10-11-12-18-19-15-16-17

c. OnTrackingLost

Jalur 1 = 1-2-3-4-5-6-7-8-9-10

Jalur 2 = 1-2-3-5-6-7-8-9-10

Jalur 3 = 1-2-3-4-5-7-8-9-10

E. Kebenaran Jalur Perhitungan BCO

1. Menampilkan Objek 3D

a. OnTrackableStateChanged

Test Case	OnTrackableStateChanged
Target yang diharapkan	Mendapatkan status pencarian marker dari kamera
Hasil Pengujian	Benar

Path / Jalur	1-2-3-5
--------------	---------

b. OnTrackingFound

Test Case	OnTrackingFound
Target yang diharapkan	Melakukan render objek 3D jika marker ditemukan
Hasil Pengujian	Benar
Path / Jalur	1-2-3-4-5-6-7-8

c. OnTrackingLost

Test Case	OnTrackingLost
Target yang diharapkan	Melakukan disable render objek 3D jika marker tidak ditemukan
Hasil Pengujian	Benar
Path / Jalur	1-2-3-4-5-6-7-8

2. Memutar Musik

Menghitung jalur kebenaran BCO memutar music sama dengan menghitung jalur kebenaran BCO menampilkan objek 3D

3. Memutar Video

a. OnTrackableStateChanged

Test Case	OnTrackableStateChanged
Target yang diharapkan	Mendapatkan status pencarian marker dari kamera
Hasil Pengujian	Benar
Path / Jalur	1-2-3-5

b. OnTrackingFound

Test Case	OnTrackingFound
-----------	-----------------

Target yang diharapkan	Melakukan render video jika marker ditemukan
Hasil Pengujian	Benar
Path / Jalur	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17

c. OnTrackingLost

Test Case	OnTrackingLost
Target yang diharapkan	Melakukan disable render video jika marker tidak ditemukan
Hasil Pengujian	Benar
Path / Jalur	1-2-3-4-5-6-7-8-9-10

Lampiran B. Hasil Pembuatan Sistem

Hasil pembuatan sistem adalah tahap yang mempresentasikan implementasi perancangan sistem yang telah dibuat menjadi aplikasi yang dapat digunakan oleh user. Sistem yang dibangun diharapkan dapat mempermudah user menggunakan teknologi AR dan menarik user untuk menggunakan aplikasi ini. Aplikasi ini menggunakan bahasa C#, vuforia SDK dan Android SDK dalam pembuatannya.

Berikut merupakan tampilan-tampilan pada aplikasi *Binary Augmented Reality*.

A. Halaman Splash Screen

Splash screen merupakan tampilan awal pada saat aplikasi ini dijalankan. Halaman ini menampilkan logo Engine unity 3D dan binary.



Halaman Splash Screen

B. Halaman Menu

Pada halaman ini terdapat beberapa pilihan yang dapat dilihat dan dipilih oleh user dengan cara menyentuh tombol yang akan dipilih. Ada beberapa menu yang ditampilkan yakni *Start*, *Tutorial*, *About* dan *Exit*.



C. Halaman *Tutorial*

Halaman ini akan muncul ketika *user* menyentuh tombol *tutorial* pada halaman *menu*. Isi dari halaman ini adalah cara penggunaan aplikasi *Binary Augmented Reality*.



D. Halaman *About*

Halaman ini akan muncul ketika *user* menyentuh tombol *about* pada halaman sebelumnya yakni halaman *menu*. Isi dari halaman ini adalah informasi mengenai aplikasi *Binary Augmented Reality*, versi aplikasi, dan pembuat aplikasi.



E. Halaman Start

Halaman *start* merupakan halaman utama dalam aplikasi ini karena pada saat *user* menyentuh tombol *start*, aplikasi akan menampilkan tampilan utama yakni tampilan kamera dari *android device*.



Cara penggunaan halaman ini cukup sederhana yakni *user* mengarahkan kamera ke *image* yang telah dipersiapkan sebelumnya. *Image* yang dimaksud ialah gambar yang ada pada majalah *binary*. Akan tetapi tidak semua gambar pada majalah terdapat *output*-nya, melainkan hanya pada *image* yang telah dipersiapkan saja.

F. Tampilan *Output* Objek 3D

Setelah *user* masuk ke halaman *start*, aplikasi akan menampilkan kamera yang berfungsi untuk menangkap *marker* sebagai acuan menampilkan objek yang akan ditampilkan. Pada *output* 3D, aplikasi akan mengidentifikasi *marker* yang sudah diatur untuk mengeluarkan objek 3D seperti pada gambar dibawah ini saat membuat pengaturannya di unity3D.



G. Tampilan Output Suara

Pada tampilan *output* suara, tentunya tetap menggunakan kamera sebagai alat untuk mencari marker yang telah diatur. Adapun langkahnya sama dengan langkah dalam tampilan *output* objek 3D. Perbedaannya hanya pada objek *output*-nya yakni dalam hal ini adalah suara yang berekstensi mp3 dan tidak terdapat sesuatu yang keluar pada kamera seperti pada gambar dibawah ini.



F. Tampilan output video

Tampilan *output video* tidak jauh berbeda juga dengan proses output pada tampilan *output* objek 3D. Perbedaannya terdapat pada objek *output*-nya yakni *video*. *Video* tersebut tidak termasuk dalam aplikasi, melainkan *video* akan di *stream* melalui *cloud* yang telah diatur sedemikian rupa. Sehingga pada

penggunaan aplikasi ini diharuskan menggunakan layanan internet guna untuk Memutar *video* yang akan ditonton.

Pada saat kamera menemukan marker untuk *output video*, aplikasi akan menampilkan gambar latar dan gambar *icon* sesuai dengan kondisi *output*-nya. Jika aplikasi belum terhubung dengan layanan *internet*, maka akan video tidak akan bisa di *play* dan akan muncul *icon* silang seperti pada gambar dibawah ini.



Sedangkan jika aplikasi telah terhubung dengan layanan *internet*, maka akan muncul *icon* seperti pada gambar dibawah.

