

# PERANCANGAN DAN ANALISA KINERJA TEKNOLOGI *AUGMENTED REALITY* PADA ORGAN TUBUH MANUSIA BERBASIS *SMARTPHONE* ANDROID

Fikri Azmi Naufal : 1310651148<sup>1)</sup>

Dosen Pembimbing: Mudafiq Riyen Pratama S.Kom., M.Kom. <sup>2)</sup>

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember

Jl. Karimata No. 49 Jember Indonesia

E-mail : fikri.azmi@rocketmail.com<sup>1)</sup>, mudafiq.riyan@unmuhjember.ac.id<sup>2)</sup>

## ABSTRAK

Salah satu teknologi yang cukup menarik saat ini yang dapat dikembangkan pada telepon pintar adalah *Augmented Reality* (AR). AR dapat memberikan informasi yang dapat lebih mudah dipahami oleh pengguna. Karena kelebihan yang dimiliki, AR dapat dimanfaatkan untuk membuat aplikasi pembelajaran. Salah satu pelajaran yang dapat memanfaatkan teknologi ini adalah pelajaran IPA mengenai organ tubuh manusia. Aplikasi pembelajaran organ tubuh manusia berbasis *smartphone* android ini memanfaatkan teknologi *augmented reality* dan menggunakan *marker* berupa *text* (*text recognition*). Aplikasi ini dibuat menggunakan *software* unity 3D dengan library *vuforia* dan untuk pembuatan objek 3D menggunakan *software* blender. Pengujian aplikasi terdiri dari dua tahap, yaitu pengujian sistem pada device dengan spesifikasi yang berbeda dan pengujian pendeteksian *marker* yang meliputi pengujian jenis *font*, *multi target*, jarak, cahaya, dan sudut. Hasil pengujian menunjukkan kamera dapat mengenali *marker* yang berupa *single marker* ataupun *multit marker* dengan jenis *font* sans serif, serif, retro dan comic. Sedangkan proses pendeteksian *marker* yang baik dilakukan dengan jarak 15 cm - 30cm dengan ukuran font 100 pt, cahaya ideal 200lx - 500lx dan sudut 30°-90°.

**Kata kunci:** *Augmented Reality*, Android, Font, Text, *Vuforia* SDK.

## I. PENDAHULUAN

Android merupakan teknologi *mobile* yang sedang berkembang di dunia. Pengguna perangkat Android tersebar diseluruh segmen masyarakat dunia. Fasilitas yang diberikan sangat mempermudah pengguna peranti ini dalam aktifitas sehari-hari. Ditambah lagi Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak, sehingga aplikasi yang dapat digunakan menjadi sangat beragam.[1]

Salah satu teknologi yang dapat diterapkan pada platform android adalah teknologi *augmented reality* atau yang biasa disebut AR. Menurut Ronald T.

Azuma, *augmented reality* didefinisikan sebagai penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata (*real-time*), dan terdapat integrasi antar benda dalam 3D, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktifitas dimungkinkan melalui perangkat-perangkat input tertentu, dan integrasi yang baik memerlukan penjejakan yang efektif [2].

Selama ini, *Augmented Reality* diaplikasikan dengan menggunakan *marker* berupa gambar. Penggunaan *marker* berupa gambar membuat penggunaan ruang pada obyek yang dilacak menjadi tidak efisien dan kurang menarik[3]. Didalam *augmented reality* terdapat sebuah fitur deteksi *marker* berdasarkan pengenalan text. Dengan adanya fitur tersebut membuat aplikasi *Augmented Reality* lebih praktis dan bisa digunakan dimanapun tanpa perlu mencetak *marker* [4].

Aplikasi *Augmented Reality* akan sangat efisien bila dapat dikembangkan dalam peranti bergerak. Sebuah *text* dideteksi menggunakan kamera dari perangkat Android secara *real time*, lalu memunculkan informasi lain secara virtual pada layar perangkat tersebut. Metode ini selain lebih interaktif dan

menghibur, juga dapat meningkatkan efisiensi karena langsung dapat diaplikasikan oleh para pengguna Android secara praktis. Bukan menggunakan *webcam* pada komputer pribadi.

Berdasarkan latar belakang diatas, maka akan dibangun sebuah aplikasi mobile dengan sistem operasi android yang memanfaatkan teknologi *augmented reality* untuk mencari kombinasi jarak, cahaya, sudut dan jenis *font* yang ideal pada proses pendeteksian *marker* sehingga proses pendeteksian menjadi lebih cepat.

## II. TINJAUAN PUSTAKA

### 2.1 *Augmented Reality*

*Argument reality* adalah sebuah istilah untuk lingkungan yang menggabungkan dunia nyata dan dunia virtual yang dibuat oleh komputer sehingga batas keduanya menjadi sangat tipis [5]. *Augmented reality* sebagai sistem yang memiliki beberapa karakteristik diantaranya, menggabungkan

lingkungan nyata dan virtual, berjalan secara interaktif dalam waktu nyata (*real time*), berintegrasi dalam tiga dimensi [2].

Beberapa komponen yang diperlukan dalam pembuatan aplikasi *augmented reality* adalah sebagai berikut [6] :

1. Komputer, berfungsi sebagai perangkat yang digunakan untuk mengendalikan semua proses yang akan terjadi dalam sebuah aplikasi.
2. *Marker*. Komputer akan mengenali posisi dan orientasi dari marker dan akan menciptakan objek virtual yang berupa obyek 3D yaitu pada titik (0, 0, 0) dan 3 sumbu (X, Y, Z).
3. Kamera, merupakan perangkat yang berfungsi sebagai *recording sensor*.

Tujuan dari *augmented Reality* adalah mengambil dunia nyata sebagai dasar dengan menggabungkan beberapa teknologi virtual dan menambahkan data konstektual agar pemahaman manusia sebagai pengguna menjadi semakin jelas [5].

## 2.2 Tiga Dimensi

Benda tiga dimensi adalah bentuk dari benda yang memiliki panjang, lebar, dan tinggi yang memiliki bentuk. Konsep tiga dimensi (3D) menunjukkan sebuah objek atau ruang memiliki tiga dimensi geometris yang terdiri dari: kedalaman, lebar dan tinggi. Karakteristik 3D, mengacu pada tiga dimensi spasial, bahwa 3D menunjukkan suatu titik koordinat cartesian X, Y dan Z. [7]

## 2.3 Android

Android adalah sistem operasi berbasis Linux yang digunakan untuk telepon seluler (mobile) seperti telepon pintar (*smartphone*) dan komputer tablet (PDA). Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang digunakan oleh bermacam peranti bergerak. Android kini telah menjelma menjadi sistem operasi mobile terpopuler didunia. [8]

### 2.3.5 Qualcomm Vuforia SDK

Qualcom vuforia menggunakan komputer vision untuk mengolah gambar grafis dan memodifikasinya dengan obyek lain yang seolah-olah muncul di dunia nyata [9]. Berikut ini beberapa fitur yang dimiliki vuforia software development kit adalah sebagai berikut [9] :

1. Mendeteksi dengan cepat target lokal dengan kapasitas melacak beberapa target secara simultan.
2. Pendeteksian dalam keadaan kurang cahaya dan bahkan ketika target tertutupi sebagian.
3. Kapasitas pelacakan yang tinggi, yang membuat aplikasi terus melacak *target*.

## 2.6 Black Box Testing

Pengujian *black box testing* digunakan untuk menguji fungsi - fungsi khusus dari perangkat lunak. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut. Dari keluaran yang dihasilkan, kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya. [10]

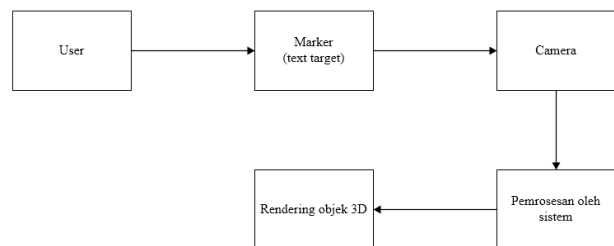
Pengujian black box mencakup dua hal sebagai berikut [11]:

1. Pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada output yang dihasilkan yang merespon input yang dipilih dan kondisi eksekusi.
2. Pengujian yang dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu.

## III. ANALISIS DAN PERANCANGAN SISTEM

### 3.1 Arsitektur Sistem

Arsitektur sistem yang akan dibangun terdiri dari beberapa komponen yaitu ; *user*, *marker*, kamera. *User* adalah pengguna yang menggunakan aplikasi, *user* mengarahkan kamera ke *marker* sehingga *marker* dapat ditangkap oleh kamera. Kemudian pada kamera akan dilakukan *tracking* atau pelacakan pada *marker* untuk mengidentifikasi *marker* yang ditangkap oleh kamera. *Marker* yang telah teridentifikasi oleh kamera akan melakukan *render* objek 3D diatas *marker* yang telah teridentifikasi oleh sistem. *Marker* yang digunakan dalam proses pendeteksian berupa text (*Text recognition*).

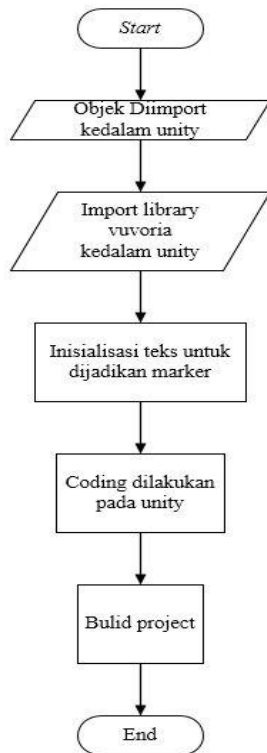


Gambar 1 Blok Diagram Arsitektur Sistem

### 3.2. Proses Pembuatan *Augmented Reality*

*Aplikasi augmented reality* organ tubuh manusia dibangun menggunakan *software unity 3D*. Dalam proses pembuatannya dibutuhkan sebuah *library* untuk proses pendeteksian *marker*. *Library* yang digunakan dalam membangun aplikasi *augmented reality* ini menggunakan *library vuforia*. Selain *library*, objek 3D juga dibutuhkan dalam pembuatan aplikasi *augmented reality*. Pada pembuatan aplikasi, objek 3D dalam format *.blend* harus diimport ke Unity 3D. Setelah objek diimport ke Unity 3D, lalu lakukan *split* objek 3D, *split* objek

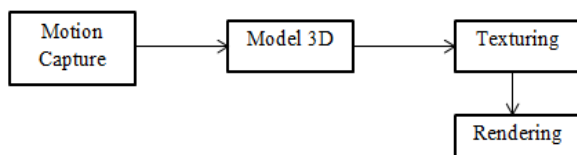
3D berguna ketika proses pendeteksian oleh *marker* berhasil, objek 3D ditampilkan tepat diatas *marker*. Untuk melakukan proses pemanggilan objek 3D beserta fiturnya dilakukan proses *coding* pada unity. Proses *coding* yang dilakukan pada unity ini menggunakan *compiler* monodevelop dengan bahasa pemrograman *c#*. Setelah proses selesai dilakukan makan tahap ahir dari pembuatan aplikasi *augmented reality* ini adalah mengekspor *project* sehingga aplikasi dapat dijalankan pada *smartphone* android .



Gambar 2 Flowchart Pembuatan Aplikasi Augmented Reality

### 3.3 Pemodelan Objek 3D

Aplikasi *augmented reality* organ tubuh manusia menggunakan objek 3D. Pembuatan objek 3D dibuat menggunakan *software* blender. Objek hasil olahan blender yang memiliki *extensi* .blend akan diekspor menjadi *file* .fbx supaya dapat dijalankan sempurna pada unity3D. Dalam proses pemodelan objek 3D membutuhkan perancangan yang dibagi dengan beberapa tahapan untuk pembentukannya. Berikut ini adalah blok diagram proses pemodelan objek 3D.



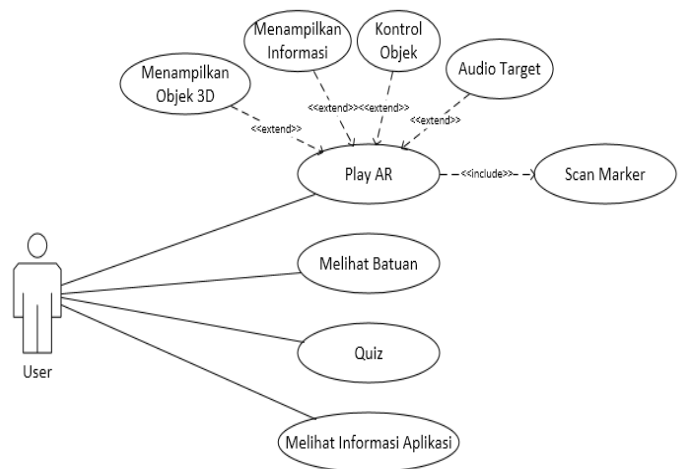
Gambar 3 Blok Diagram Pemodelan Objek 3D

*Motion Capture*, yaitu langkah awal untuk menentukan bentuk model objek yang akan dibangun dalam bentuk 3D. Setelah menentukan model yang akan dibangun dalam bentuk 3D langkah selanjutnya yaitu *modeling* 3D. Pada tahap ini objek 3D dibentuk dari sebuah objek dasar hingga terbentuk sebuah objek 3D yang akan dibangun. Tahap selanjutnya yaitu *texturing*. Proses *texturing* ini untuk menentukan karakteristik sebuah materi objek dari segi tekstur. *Texture* kemudian bisa digunakan untuk membuat berbagai variasi warna *pattern*, tingkat kehalusan atau kekasaran sebuah lapisan objek secara lebih detail. Tahap ahir dari proses pemodelan objek 3D yaitu *rendering*. *Rendering* adalah proses akhir dari keseluruhan proses pemodelan ataupun animasi komputer. Dalam *rendering*, semua data-data yang sudah dimasukkan dalam proses *modeling*, *texturing*, pencahayaan dengan parameter tertentu akan diterjemahkan dalam sebuah bentuk output (tampilan akhir pada model objek 3D).

### 3.4 Pemodelan Sistem

Pemodelan sistem perlu dilakukan untuk memberikan gambar yang jelas dan lengkap tentang rancang bangun dan implementasi bagaimana sistem tersebut dibuat. Pemodelan sistem dimodelkan dengan menggunakan UML (*Unified Modeling Language*).

*Use case diagram* digunakan untuk memodelkan atau menggambarkan batasan sistem dan fungsi-fungsi utamanya. Mendiskripsikan fungsi dari sebuah sistem dari perspektif pengguna, *use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri, dengan memberi sebuah gambaran tentang bagaimana sistem tersebut digunakan. *Use case* terdiri dari tiga bagian yaitu definisi *use case*, definisi aktor, *scenario use case*. Berikut ini adalah *use case diagram* yang digunakan untuk membangun aplikasi pembelajaran organ tubuh manusia dapat dilihat pada gambar 3.5



Gambar 3 Use Case Diagram Aplikasi

## IV. IMPLEMENTASI DAN PENGUJIAN

### 4.1 Implementasi Sistem

Implementasi sistem merupakan tahap penerjemah perancangan aplikasi sesuai dengan hasil analisis yang telah dilakukan. Setelah implementasi dilakukan maka tahapan selanjutnya adalah pengujian sistem. Berikut ini adalah spesifikasi perangkat yang digunakan dalam pembangunan aplikasi *augmented reality* organ tubuh manusia.

Tabel 2 Spesifikasi perangkat dalam implementasi aplikasi

Hardware	Software
Processor Intel (R) Celeron (R) CPU B820 @1.70GHz	Sistem Operasi Microsoft windows 7 ultimate 64-bit
6 GB RAM	Unity-3D v 5.3.6p1
500 Gb	Blender 2.78.3
	vuforia 6.0.117
	Adobe Photoshop cc 2014
	Android SDK

### 4.2 Pengujian Fungsional

Pengujian fungsional menggunakan metode *black box testing*. Pengujian pengujian *black box testing* yang dilakukan pada aplikasi *augmented reality* berguna untuk mengetahui pengaruh pendeteksian kamera *augmented reality* terhadap *marker*. Pengujian ini berupa pengujian jarak, sudut cahaya, jenis *font*, dan *multi target*.

#### 4.2.1 Pengujian Jarak

Jarak yang digunakan dalam pengujian ini 5 cm sampai 35 cm sedangkan *marker* yang digunakan dalam pengujian ini menggunakan *font arial* dengan ukuran 100pt. Pengujian jarak dilakukan untuk memperoleh data uji standar jarak yang optimal antara *marker* dan kamera. Pada pengujian jarak digunakan cahaya 350 - 600 lx dan sudut 60°. Berikut tabel pengujian jarak yang telah dilakukan.

Tabel 3 Hasil pengujian jarak

Jarak (cm)	Device 1	Device 2	Device 3	Device 4
5	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>
10	Rendering objek 3D gagal	proses <i>rendering</i> 4 detik	proses <i>rendering</i> 4 detik	proses <i>rendering</i> 2 detik
15	Rendering objek 3D gagal	proses <i>rendering</i> 3 detik	proses <i>rendering</i> 2 detik	proses <i>rendering</i> 2 detik
25	Rendering objek 3D gagal	proses <i>rendering</i> 3 detik	proses <i>rendering</i> 2 detik	proses <i>rendering</i> 1 detik
30	Rendering objek 3D gagal	proses <i>rendering</i> 3 detik	proses <i>rendering</i> 3 detik	proses <i>rendering</i> 1 detik
35	Rendering objek 3D gagal	proses <i>rendering</i> 8 detik	proses <i>rendering</i> 7 detik	proses <i>rendering</i> 4 detik

Hasil pengujian yang ditunjukkan pada pengujian jarak didapat bahwa pada jarak 5 cm

*marker* tidak dapat terdeteksi. Pada jarak tersebut ukuran *marker* melebihi *frame* kamera dan kamera tidak dapat fokus pada jarak tersebut. Pada jarak 35 cm proses pendeteksian membutuhkan waktu lebih lama dibandingkan dengan jarak 10 cm sampai 30 cm, hal ini dikarenakan ukuran *marker* terlalu kecil sehingga kamera membutuhkan waktu yang lebih lama untuk fokus terhadap *marker*.

#### 4.2.2 Pengujian Sudut

Pada pengujian sudut digunakan jarak *marker* dengan kamera yaitu 20 cm. Karena jarak tersebut adalah jarak yang cukup baik untuk deteksi *marker*, sehingga hasil yang diharapkan pada pengujian sudut ini efektif. *Marker* yang digunakan dalam pengujian ini menggunakan *font arial* dengan ukuran 100pt. Sedangkan cahaya yang digunakan yaitu 200 - 500 lx. Pengujian sudut dimulai dari 0° dihitung dari garis lurus pada sebelah *marker* (sumbu X). Berikut ini adalah tabel dari hasil dari pengujian sudut.

Tabel 4 pengujian sudut

Sudut	Device 1	Device 2	Device 3	Device 4
0°-9°	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>
10° - 19°	Rendering objek 3D gagal	proses <i>rendering</i> 7detik	proses <i>rendering</i> 5 detik	proses <i>rendering</i> 4 detik
20° - 39°	Rendering objek 3D gagal	proses <i>rendering</i> 5 detik	proses <i>rendering</i> 3 detik	proses <i>rendering</i> 2 detik
40° - 59°	Rendering objek 3D gagal	proses <i>rendering</i> 3 detik	proses <i>rendering</i> 2 detik	proses <i>rendering</i> 2 detik
60° - 79°	Rendering objek 3D gagal	proses <i>rendering</i> 2 detik	proses <i>rendering</i> 3 detik	proses <i>rendering</i> 2 detik
80° - 90°	Rendering objek 3D gagal	proses <i>rendering</i> 2 detik	proses <i>rendering</i> 2 detik	proses <i>rendering</i> 2 detik

Hasil pengujian yang ditunjukkan pada Tabel 4.11 didapat bahwa pada Sudut 0°-9° kamera tidak dapat menangkap *marker* sepenuhnya tidak tertangkap oleh kamera *augmented reality*. Untuk melakukan proses pendeteksian yang ideal dilakukan pada sudut 10° - 90°. Namun pada sudut 90° objek yang nampak ketika proses *rendering* objek 3D hanya tampak di bagian atas, sehingga objek 3D tidak terlihat secara jelas. Dari pengujian sudut dapat disimpulkan bahwa semakin tinggi tingkat kemiringan pada proses pendeteksian *marker*, maka semakin tinggi tingkat keberhasilan dalam proses pendeteksian *marker*.

#### 4.2.3 Pengujian Cahaya

Pengujian pencahayaan dilakukan menggunakan *software* pengukur intensitas cahaya yaitu light meter dengan satuan lx. Pengujian ini dilakukan pada jarak 25 cm dengan sudut

kemiringan kamera 75° terhadap *marker*. Sementara *marker* yang digunakan yaitu *font* arial dengan ukuran 100 pt.

Tabel 5 Hasil pengujian pencahayaan

Intensitas cahaya (lx)	Device 1	Device 2	Device 3	Device 4
> 10	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>
11 - 60	Rendering objek 3D gagal	proses rendering 18 detik	proses rendering 17 detik	proses rendering 12 detik
80 - 170	Rendering objek 3D gagal	proses rendering 8 detik	proses rendering 5 detik	proses rendering 5 detik
170 - 350	Rendering objek 3D gagal	proses rendering 5 detik	proses rendering 4 detik	proses rendering 2 detik
350 - 700	Rendering objek 3D gagal	proses rendering 3 detik	proses rendering 3 detik	proses rendering 3 detik
700 - 1000	Rendering objek 3D gagal	proses rendering 2 detik	proses rendering 1 detik	proses rendering 1 detik
>10000	Rendering objek 3D gagal	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>	Kamera tidak dapat menangkap <i>marker</i>

Cahaya memegang peran penting dalam proses pendeteksian *marker*. Pada *device* ke 2 sampai *device* ke 5 ketika melakukan pengujian pencahayaan dengan skala intensitas cahaya 11 - 60 lx kamera dapat melakukan pendeteksian *marker* kemudian melakukan proses rendering. Namun karena jumlah intensitas cahaya yang kurang maka objek yang ditampilkan terkadang hilang (*ontrackablelost*). Selain itu pada skala intensitas cahaya 11 - 60 lx proses pendeteksian membutuhkan waktu yang lebih lama dibandingkn dengan skala intensitas cahaya 80 - 170 lx, 170 - 350 lx, 350 - 700 lx, 700 - 1000 lx. Pada skala intensitas cahaya lebih dari 10000 lx kamera gagal menangkap *marker* hal ini dikarenakan skala intensitas cahaya yang ditangkap oleh kamera terlalu banyak (*over exposure*). Dari pengujian pencahayaan dapat disimpulkan bahwa intensitas cahaya yang dapat melakukan proses pendeteksian *marker* dengan bail memiiki skala cahaya 80 lx sampai 1000 lx.

#### 4.2.4 Pengujian Font

Pengujian jenis *font* dilakukakn untuk mengetahui jenis *font* yang dapat terdeteksi oleh kamera *augmented reality*. *font* yang digunakan dalam pengujian ini menggunakan beberapa jenis *font* dengan ukuran *font* 100pt. Pengujian ini dilakukan pada jarak 25 cm dengan sudut kemiringan kamera 75° terhadap *marker*. Sedangkan cahaya yang digunakan yaitu 200 - 500 lx. Karena jarak sudut, dan intensitas cahaya yang ideal adalah pada kondisi tersebut. Berikut ini hasil dari pengujian *font*.

Tabel 6 Hasil pengujian jenis Font

Jenis Font	Device 1	Device 2	Device 3	Device 4
Serif : - Book Antiqua - Cambria - Century - <b>Cooper Black</b> - Times New Roman	Rendering objek 3D gagal	Rata –rata proses rendering membutuhkan waktu 5 detik	Rata –rata proses rendering membutuhkan waktu 3 detik	Rata –rata proses rendering membutuhkan waktu 2 detik
Sans Serif : - Franklin Gothic - Gill Sans MT - <b>Impact</b> - Lucida Bright - Verdana	Rendering objek 3D gagal	Rata –rata proses rendering membutuhkan waktu 4 detik	Rata –rata proses rendering membutuhkan waktu 3 detik	Rata –rata proses rendering membutuhkan waktu 2 detik
Comic : - Axel Gibby Comic - Comic Sans MS - Life is goofy - <b>Luckiest</b> - <b>Business</b>	Rendering objek 3D gagal	Rata –rata proses rendering membutuhkan waktu 6 detik	Rata –rata proses rendering membutuhkan waktu 5 detik	Rata –rata proses rendering membutuhkan waktu 2 detik
Retro : - Cafeandbrewery - PointOne - <b>Titania</b> - <b>Salsa</b> - <b>Klarissa</b>	Rendering objek 3D gagal	Rata –rata proses rendering membutuhkan waktu 5 detik	Rata –rata proses rendering membutuhkan waktu 5 detik	Rata –rata proses rendering membutuhkan waktu 2 detik

Dari hasil pengujian *font* didapat proses pendeteksian menggunakan jenis *font* comic dan retro lebih lama dibandingkan dengan jenis *font* serif dan san serif. Hal ini dikarenakan jenis *font* comic dan retro memiliki pola yang sedikit rumit sehingga sistem membutuhkan waktu yang lebih lama untuk melakukan proses identifikasi. Sedangkan waktu tercepat dalam proses pendeteksian menggunakan jenis *font* serif.

Terdapat beberapa jenis *font* yang tidak dapat terdeteksi oleh kamera *augmented reality* dikarenakan jenis *font* tersebut memiliki pola dengan tingkat kerumitan yang tinggi sehingga tidak dapat dikenali oleh sistem. Adapun jenis *font* yang tidak bisa dideteksi oleh kamera *augmented reality* untuk dilakukan proses rendering adalah sebagai berikut.

Tabel 7 Hasil pengujian jenis Font

Jenis Font	Device 1	Device 2	Device 3	Device 4
Blackletter : <b>Canterbury</b> <b>Hamppplate</b> <b>Hamppplate</b> <b>EmbossedGermanica</b> <b>PlainGermanica</b>	Font tidak dapat teridentifikasi	Font tidak dapat teridentifikasi	Font tidak dapat teridentifikasi	Font tidak dapat teridentifikasi
Script : <i>Cliver Script</i> <i>Della Script</i> <i>Futura Script</i> <i>Playball</i>	Font tidak dapat teridentifikasi	Font tidak dapat teridentifikasi	Font tidak dapat teridentifikasi	Font tidak dapat teridentifikasi
Ghotic : <i>Offense Personal Use Only</i> <b>BOEAT</b> <b>BGORYGASTLE</b> <i>ithoruet</i> <i>Minter</i> <b>MALERTMENT</b>	Font tidak dapat teridentifikasi	Font tidak dapat teridentifikasi	Font tidak dapat teridentifikasi	Font tidak dapat teridentifikasi

#### 4.2.5 Pengujian *Multi Target*

Pengujian terakhir yang dilakukan untuk menguji masukan dan keluaran dari *augmented reality* adalah pengujian *multi target*. Pengujian *multi target* dilakukan untuk mengetahui pengaruh sistem apabila *marker* yang tertangkap kamera *augmented reality* lebih dari satu *marker*. *Font* yang digunakan dalam pengujian ini menggunakan *font impact* jenis sans serif dengan ukuran *font* 100pt. Pengujian ini dilakukan pada jarak 25 cm dengan sudut kemiringan kamera 75° terhadap *marker*. Sedangkan cahaya yang digunakan yaitu 200 - 500 lx. Karena jarak sudut, dan intensitas cahaya yang ideal adalah pada kondisi tersebut. Berikut ini hasil dari pengujian *multi target*.

Tabel 8 Hasil pengujian *multi target*

Marker	Device 1	Device 2	Device 3	Device 4
Single marker	Rendering objek 3D gagal	Objek 3D dapat ditampilkan beserta fitur.	Objek 3D dapat ditampilkan beserta fitur.	Objek 3D dapat ditampilkan beserta fitur.
Multi marker	Rendering objek 3D gagal	Seluruh <i>marker</i> dapat terdeteksi dan objek 3D dapat ditampilkan secara bersamaan.  Masing masing fitur dari objek tidak dapat tampil.	Seluruh <i>marker</i> dapat terdeteksi dan objek 3D dapat ditampilkan secara bersamaan.  Masing masing fitur dari objek tidak dapat tampil.	Seluruh <i>marker</i> dapat terdeteksi dan objek 3D dapat ditampilkan secara bersamaan.  Masing masing fitur dari objek tidak dapat tampil.

Dari pengujian *multi target* didapat bahwa ketika kamera *augmented reality* menangkap *marker* yang berjumlah lebih dari satu (*Multi marker*), kamera dapat mendeteksi *marker* dan melakukan proses *rendering* objek 3D yang berbeda secara bersamaan. Namun ketika proses *rendering* objek 3D berhasil fitur berupa rotasi, penjelasan, dan audio dari masing masing objek 3D tidak dapat ditampilkan secara bersamaan.

#### V. KESIMPULAN

Berdasarkan hasil penelitian pada aplikasi *augmented reality* organ tubuh manusia didapatkan kesimpulan sebagai berikut:

- Objek 3D dibangun menggunakan *software blender* sedangkan untuk membangun aplikasi *augmented reality* menggunakan *software unity 3D* dengan *library vuforia*. Dalam proses pembuatan objek 3D terdapat beberapa tahapan diantaranya *motion capture*, model 3D, *texturing*, dan *rendering*. Objek 3D yang telah jadi di-export dengan *extensi* .3ds, .obj, dan .fbx untuk di-import kedalam unity. Pada *software unity* dilakukan proses inialisasi *text* yang nantinya akan dijadikan sebagai *marker*. Untuk menampilkan objek 3D beserta fitur dilakukan proses *coding*.

- Jenis *font* yang baik untuk proses pendeteksian kamera *augmented reality* memiliki pola yang tidak rumit diantaranya sans serif, serif, comic, dan retro.
- Marker* yang terdeteksi oleh kamera *augmented reality* berjumlah lebih dari satu (*multi marker*) dapat menampilkan Objek 3D yang berbeda secara bersamaan. Akan tetapi fitur berupa audio, rotasi, dan deskripsi tidak dapat tampil pada masing masing objek.
- Aplikasi *augmented reality* organ tubuh manusia dapat berjalan pada perangkat *mobile* bersistem operasi android dengan versi 4.0 (Ice Cream Sandwich) yang memiliki RAM minimal 1 GB.
- Pada *device smartfren andromax-i* ketika melakukan proses pendeteksian *marker* kamera gagal melakukan proses *rendering* objek 3D (aplikasi force close). Hal ini dikarenakan pada *device smartfren andromax-I* memiliki RAM 512 MB. Dengan kapasitas RAM yang kecil maka akan menambah *swapping memory* ke *virtual memory*.
- Pada *device* dengan spesifikasi layar *xhdpi* (extra high density) ketika akan menjalankan menu quiz tampilan menjadi kecil hal ini dikarenakan aplikasi *augmented reality* organ tubuh manusia dibuat dengan ukuran layar *hdpi* (*High density*).
- Kamera *augmented reality* dapat melacak *marker* dan melakukan proses *rendering* untuk menampilkan objek 3D meski intensitas cahaya rendah (skala intensitas cahaya 11-60 lx). Begitu pula dengan intensitas cahaya yang tinggi (skala intensitas cahaya lebih dari 10000 lx) kamera gagal menangkap *marker* hal ini dikarenakan skala intensitas cahaya yang ditangkap oleh kamera terlalu banyak (*over exposure*).
- Jarak ideal dalam melakukan proses pendeteksian *marker* terjadi pada jarak 15 cm sampai 30 cm dengan sudut kemiringan *marker* terhadap kamera 30° atau lebih.

#### DAFTAR PUSTAKA

- Speckmann, B. (2008). The Android mobile platform. Access.
- Azuma, T.R. (1997). *A Survey of Augmented Reality*. Presence: Teleoperators and Virtual Environments.
- Rizki, Yoze. (2012). Markerless Augmented Reality Pada Perangkat Android. Jurnal Teknik Elektro. Institut Teknologi Sepuluh November Surabaya.
- Widiyanto, mala (2014). Modifikasi text recognition pada augmented reality dalam perancangan system informasi tanaman. Universitas Kristen satya wacana.
- Akhmad Afissunani, Akuwan saleh, M. Hasbi Assidiqi. (2014). *Multi Marker Augmented Reality untuk Aplikasi Magic Book*.

- [6] Martono, K.T. (2011) . *Augmented Reality Sebagai Metafora Baru dalam Teknologi Interaksi Manusia dan Komputer*. Semarang, Indonesia: JURNAL SISTEM KOMPUTER.
- [7] Rahmat, Berki. (2011). *Analisis dan Perancangan Sistem Pengenalan Bangun Ruang Menggunakan Augmented Reality*. Medan, Indonesia: Universitas Sumatera Utara.
- [8] Adam, Arie, Lumenta, & Robot, J. R. (2014). *Implementasi Teknologi Augmented Reality pada Agen Penjualan Rumah*. Teknik Elektro dan Komputer , 19-25.
- [9] Murya, Yosef. (2014). *Pemograman Android Black Box*. Jasakom.
- [10] Amin, D. & Govilkar, S. (2015). *Comparative study of augmented reality SDK's*. International Journal on Computational Sciences & Applications (IJCSA) 11-26.
- [11] Myers, G. J. (2004). *The Art of Software Testing, Second Edition*. John Wiley & Sons, Inc. Canada.
- [12] IEEE. (1990). *IEEE Standard Glossary of Software Engineering Terminology*. Inc 345 Eas 47th Street New York, NY 10017-2394, USA