

# APLIKASI *MOBILE AUGMENTED REALITY* SEBAGAI MEDIA PEMBELAJARAN PENGENALAN *HARDWARE* KOMPUTER BERBASIS ANDROID

Aries Wiharto<sup>1</sup>, Cahyani Budihartanti<sup>2</sup>

Program Studi Teknik Informatika – STMIK NUSA MANDIRI<sup>1,2</sup>

<sup>1</sup>Arieswihh27@nusamandiri.ac.id

<sup>2</sup>cahyani.cbh@nusamandiri.ac.id

**Abstrak** – Dalam mempelajari komponen-komponen yang terdapat komputer terdapat beberapa masalah yang dialami oleh masyarakat khususnya anak-anak yang beranjak dewasa. Anak-anak lebih mengenal *smartphone* dibandingkan laptop dan komputer. *Unity 3D* adalah sebuah *tools* yang terintegrasi untuk membuat bentuk objek tiga dimensi pada *video games* atau untuk konteks interaktif lain seperti visualisasi arsitektur atau animasi 3D *real-time*. *Blender* adalah salah satu *software open source* yang digunakan untuk membuat konten multimedia khususnya 3D, ada beberapa kelebihan yang dimiliki *Blender* dibandingkan *software* sejenis<sup>2</sup>. *Augmented Reality* untuk menyajikan informasi ditekankan pada pandangan mengenai dunia terbuka yang mampu memberikan kesempatan untuk berinteraksi dengan lingkungan grafis di sekitar kita secara langsung. *QR Code* merupakan gambar garis tegak yang biasanya ditempelkan pada *item toko ritail*, kartu identitas, dan surat pos untuk mengidentifikasi sejumlah produk tertentu. Dan untuk hasil diharapkan dapat membantu menjelaskan materi tentang pengenalan *hardware* komputer dan dapat lebih membantu pelajar untuk memahami materi.

**Kata Kunci:** *Augmented Reality, Blender, QR Code, Unity 3D*

## I. PENDAHULUAN

Banyak cara yang digunakan oleh masyarakat dalam mempelajari suatu hal, baik dengan cara membaca, menonton televisi, mendengarkan maupun menggunakan alat bantu. Alat bantu yang sering digunakan oleh pengguna salah satunya adalah perangkat *smartphone*. *Smartphone* selain dapat digunakan sebagai alat komunikasi seperti menelpon, mengirim pesan, *browsing* internet, *smartphone* juga dapat digunakan sebagai alat pembelajaran. Dengan menggunakan alat bantu *smartphone* diharapkan proses pembelajaran yang awalnya hanya menggunakan buku dan alat peraga yang bersifat monoton di mana otak siswa dipaksa untuk mengingat dan menyimpan berbagai informasi dapat disiasati dengan menggunakan teknologi baru yang dapat menarik minat siswa dalam belajar yaitu dengan menerapkan teknologi *Augmented Reality* dengan teknik pembelajaran dalam bentuk animasi tiga dimensi (3D), di mana maksud dalam penerapan teknik ini adalah agar proses pembelajaran terlihat lebih nyata dan menarik. Di mana teknik animasi dibangun dengan menggunakan *blender* serta proses pembangunannya dengan menggunakan teknik *Augmented Reality*.

## II. LANDASAN TEORI

### 2.1 Unity 3D

*Unity 3D* adalah sebuah perangkat lunak penyedia layanan *games*. *Unity 3D* adalah alat *authoring* yang terintegrasi untuk menciptakan rekaman permainan 3D atau konten interaktif lainnya seperti visualisasi arsitektur animasi 3D. Lingkungan pengembangan *Unity 3D* yang berjalan pada *Microsoft Windows* dan *Mac OS X*, dan permainan yang dihasilkan dapat dijalankan pada *Windows, Mac, Xbox 360, PlayStation 3, Wii, iPad, iPhone*, serta *platform Android*. Bahasa pemrograman yang dapat diterima *Unity* adalah *JavaScript, CS Script (C#)*, dan *Boo Script*.

### 2.2 Blender 3D

*Blender* adalah salah satu *software open source* yang digunakan untuk membuat konten multimedia khususnya 3D, ada beberapa kelebihan yang dimiliki *Blender* dibandingkan *software* sejenis [1]. Berikut beberapa kelebihannya. *Open Source Blender* merupakan salah satu *software open source*, di mana kita bisa bebas memodifikasi *source* kodenya untuk keperluan pribadi maupun komersial, asal tidak melanggar *GNU General Public License* yang digunakan *Blender*. *MultiPlatform*, karena sifatnya yang *open source*, *Blender* tersedia untuk berbagai macam operasi sistem seperti *Linux, Mac* dan *Windows*. Sehingga *file* yang dibuat menggunakan *Blender* versi *Linux* tak akan berubah ketika dibuka di *Blender* versi *Mac* maupun *Windows*. *Update*, dengan status yang *Open Source*, *Blender* bisa dikembangkan oleh siapapun. Sehingga

*update software* ini jauh lebih cepat dibandingkan *software* sejenis lainnya. Bahkan dalam hitungan jam, terkadang *software* ini sudah ada *update*-annya. *Update* tersebut tak tersedia di situs resmi *blender.org* melainkan di *graphicall.org*. Lengkap, *Blender* memiliki fitur yang lebih lengkap dari *software* 3D lainnya. Coba cari *software* 3D selain *Blender* yang di dalamnya tersedia fitur *Video editing*, *Game Engine*, *Node Compositing*, *Sculpting*. Bukan *plugin*, tapi sudah *include* atau di *bundling* seperti *Blender*.

*Blender* dapat digunakan untuk membuat visualisasi 3D, serta siaran dan *video* berkualitas bioskop, sedangkan penggabungan mesin 3D *real-time* memungkinkan penciptaan konten 3D interaktif untuk pemutaran yang berdiri sendiri. *Blender* memiliki berbagai macam kegunaan termasuk pemodelan, menjiwai, *rendering*, *texturing*, *rigging*, *scripting*, *composite*, *post-produksi* dan banyak lagi. *Blender* tersedia untuk berbagai sistem operasi, seperti:

- a. *Microsoft windows*
- b. *NetBSD*
- c. *OpenBSD*
- d. *Solaris*
- e. *Linux*
- f. *FreeBSD*
- g. *Mac Os X*

### 2.3 Android

*Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *Linux* yang mencakup sistem operasi, *middleware* dan aplikasi [2]. Dalam pengembangan aplikasi *Android* menyediakan *Android SDK* yang menyediakan *tools* dan *API* untuk para pengembang aplikasi dengan *platform Android*.

*Android* memiliki beberapa fitur yang menarik bagi yang ingin mengembangkan aplikasi, di antaranya sebagai berikut:

- a. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
- b. Mesin virtual *Dalvik* berjalan di atas *Linux* kernel dan dioptimalkan untuk perangkat *mobile*.
- c. *Integrated browser* berdasarkan *open source engine WebKit*.
- d. Grafis yang dioptimalkan dan didukung oleh *library* grafis 2D yang terkustomisasi, grafis 3D berdasarkan spesifikasi *OpenGL ES 1,0* (Opsional akselerasi *hardware*).
- e. *SQLite* untuk tempat penyimpanan data.
- f. *Media support* yang mendukung *audio*, *video*, dan gambar (*MPEG4*, *H.264*, *MP3*, *AAC*, *ARM*, *JPG*, *PNG*, *GIF*).
- g. *GSM Telephony* (tergantung *hardware*).
- h. *Bluetooth*, *EDGE*, *3G*, dan *Wifi* (tergantung *hardware*).
- i. Dukungan perangkat tambahan *Android* dapat memanfaatkan kamera, layar sentuh, *accelerometer*, *magnetometers*, *GPS*, Akselerasi 2D (dengan perangkat Orientasi, *Scrolling*, konversi format piksel) dan akselerasi grafis 3D.

- j. *Multi-touch*: kemampuan layaknya *handset* modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.

Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, *profile* dan kinerja memori, dan *plugin* untuk *Eclipse IDE*.

### 3.4 Augmented Reality

*Augmented reality* adalah sebuah interaksi langsung atau tidak langsung dari sebuah dunia lingkungan fisik dunia nyata yang telah ditambahkan dengan menambah komputer *virtual* yang dihasilkan informasi. *AR* adalah dua jenis teknologi interaktif dan terdaftar dalam 3D serta menggabungkan benda nyata dan *virtual* [3]. Sedangkan menurut pendapat lain, *Augmented Reality* adalah konsep pelapisan konten *visual* (seperti grafik) yang merupakan sebuah teknologi untuk menggabungkan benda maya baik 2D mau pun 3D ke dalam lingkungan nyata lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata [4].

### 3.5 Pengujian Black Box

*Black Box testing* yaitu adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat dari isinya, tapi cukup dikenai proses *testing* di bagian luar [5].

### 3.6 SDLC

*Waterfall*, merupakan *SDLC* tertua karena sifatnya yang natural. Urutan *SDLC waterfall* ini sifatnya serial dari proses perencanaan, analisis, desain, dan implementasi pada sistem [6].

### 3.7 UML

*UML (Unified Modelling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi obyek [7]. Dalam *UML* sendiri terdapat beberapa diagram yang wajib dikuasai yaitu:

- a. *Use case Diagram*  
*Use case* diagram ini menggambarkan tindakan-tindakan yang dilakukan oleh seorang aktor. Dalam hal ini aktor adalah pelajar yang akan menggunakan aplikasi.
- b. *Sequence Diagram*  
*Sequence* diagram ini menunjukkan ketika pelajar memilih *menu* Panduan akan menam-pilkan panduan atau cara menggunakan aplikasi *Augmented Reality*.
- c. *Activity Diagram*  
*Activity* diagram ini menjelaskan aktivitas yang dapat dilakukan pelajar ketika menggunakan aplikasi. Untuk memulai aplikasi, pelajar dapat

memilih *menu AR Kamera*. Untuk melihat cara pemakaian dapat memilih *menu Panduan* dan untuk keluar dari aplikasi dapat memilih *menu Keluar*.

- d. *Class Diagram*  
*Class Diagram* menggambarkan struktur objek, deskripsi objek, *class*, dan relasi satu sama lain.
- e. *Component Diagram*  
*Component diagram* adalah diagram *UML* yang menampilkan komponen dalam sistem dan hubungan antara mereka.

### III. METODOLOGI PENELITIAN

Penelitian dilakukan dalam beberapa tahap, yaitu analisis kebutuhan, desain, pengkodean, *testing* dan *support*, di mana aplikasi dapat diimplementasi kan pada *smartphone*. Model pengembangan aplikasi ini berguna untuk mendapatkan data atau informasi mengenai sistem berjalan untuk dianalisis supaya penulis dapat mengetahui kelemahan dan kekurangan yang ada pada aplikasi yang dibahas.

1. Analisis Kebutuhan Aplikasi  
Ini adalah tahap di mana penulis membuat sebuah konsep jalan cerita dan menganalisis masalah kemudian membuat keputusan hal apa yang dibutuhkan untuk pembuatan aplikasi ini dengan mendesain dan memilih *software Unity 3D, Blender, Bartender* dan *Vuforia* dan menggunakan *Android OS Lollipop*.
2. Desain  
Untuk langkah awal yang dilakukan sebelum membuat objek, aplikasi, komponen atau struktur. Dalam tahapan desain melakukan pelatihan dan tutorial. Langkah awalnya adalah mencari tekstur warna dengan animasi yang ada sehingga tampilan yang keluar pun akan terlihat menarik. Pada tahapan ini penulis menggunakan *software Unity 3D*. Di sini penulis menggunakan model, karena menggunakan objek jadi penulis menggunakan model, yaitu:
  - a. Rancang Algoritma  
Rancang Algoritma di sini penulis menggunakan Algoritma *FAST Corner Detection*.
  - b. *Software Architecture*  
*Software Architecture* di sini menggunakan model *OOP*.
  - c. Model *UML*  
Model *UML* untuk penggambaran desain. Menggunakan *Use case diagram, Activity diagram, Class diagram, Component diagram* dan *Sequence diagram*.
  - d. *User Interfaces*  
*User interfaces*-nya adalah menampilkan *button ARKamera, button Panduan button Keluar, button Kembali, button Zoom In, button Zoom Out, button Rotate, button Info, button Next, button Previous* serta *button Icon Back*.
3. Pengkodean  
Pada tahap pengkodean sebuah desain harus diterjemahkan ke dalam bentuk bahasa yang dapat

dibaca oleh sistem atau mesin. Pembuatan program ini menggunakan Bahasa Pemrograman *C#* di *Vuforia*.

4. *Testing*  
Program yang akan dibuat akan melewati suatu *testing* yang meliputi pengujian *WhiteBox* dan pengujian *BlackBox*. Di mana pengujian *WhiteBox* diasumsikan bahwa pengujian spesifik sangat penting untuk memperoleh *test case*. Sasarannya adalah memeriksa semua pernyataan program. Sedangkan pengujian *BlackBox* dilakukan setelah program ini telah diuji coba kepada pengguna. Dan pengujian ini bertujuan untuk mengasumsikan bahwa program menghasilkan *output* yang sesuai dengan rancangan.
5. *Support*  
Pada tahap ini sistem akan di-*install* dan digunakan pada beberapa *smartphone* dengan spesifikasi berbeda-beda untuk mengetahui spesifikasi *smartphone* yang mendukung sistem ini. Di sini penulis menggunakan *OS Android smartphone Lollipop* dan juga dilakukan pemeliharaan sistem seperti memperbaiki *error* yang tidak ditemukan pada tahap-tahap sebelumnya.

Tabel 1. *Hardware dan Software Support*

| <i>Hardware</i>          | <i>Software</i>                |
|--------------------------|--------------------------------|
| <i>Personal Computer</i> | <i>Windows 7 64 bit only</i>   |
| Laptop                   | <i>Intel Processor Core i3</i> |
|                          | <i>Graphic card with DX9</i>   |
|                          | <i>RAM 4 GB</i>                |

### Teknik Pengumpulan Data

Dalam membangun aplikasi ini terlebih dahulu dilakukan pengumpulan data yaitu:

- a. Penelitian Lapangan (*field research*)  
Pada tahap awal dalam penelitian ini dikumpulkan data terlebih dahulu dengan studi lapangan kepada pengguna yang mempelajari *hardware* komputer.
- b. Penelitian Kepustakaan (*Library Research*)  
Pada tahap ini dilakukan pengumpulan data dengan mempelajari buku-buku (*literature*) mengenai sistem operasi *Android*, dan *augmented reality* dan langkah-langkah membangun aplikasi yang berjalan di bawahnya, referensi dari internet dan segala hal yang berkaitan dalam membangun aplikasi.

### Analisis Kebutuhan

Untuk merancang aplikasi interaktif penulis menganalisis kebutuhan dalam pembuatan aplikasi interaktif di antaranya:

1. *Software*  
*Software* yang digunakan dalam penyusunan skripsi ini adalah:
  - a. *Unity 3D*
  - b. *Blender*
  - c. *Vuforia*
2. *Hardware*  
*Hardware* minimal yang digunakan untuk bisa membuat aplikasi ini adalah *Personal Computer* dengan spesifikasi sebagai berikut:
  - a. *Windows 7 64 bit only*
  - b. *Intel Processor Core i3*
  - c. *Graphic card with DX9*
  - d. *RAM 4 GB*

**Rancangan Algoritma**

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

*FAST (Feture Form Accelerated segment Test)* adalah suatu algoritma yang dikembangkan oleh Edward Rosten, Reid Porter, and Tom Drummond. *FAST corner detection* ini dibuat dengan tujuan mempercepat waktu komputasi secara *real-time* dengan konsekuensi menurunkan tingkat akurasi pendeteksian sudut. *FAST corner detection* dimulai dengan menentukan suatu titik *p* pada koordinat (*x<sub>p</sub>*, *y<sub>p</sub>*) pada citra dan membandingkan intensitas titik *p* dengan 4 titik di sekitarnya. Pada *FAST Coner 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*. Di samping *p* diperiksa, ada 3 kasus berbeda yang ditetapkan untuk masing-masing perbandingan yakni:

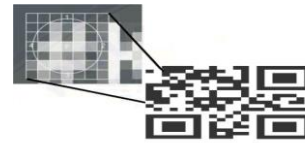
$$C = \begin{cases} |I_p - I_n| < t & \text{normal} \\ I_n - I_p > t & \text{brighter} \\ I_p - I_n > t & \text{darker} \end{cases}$$

Keterangan :

- I<sub>p</sub>* = intensitas titik pusat,
- I<sub>n</sub>* = titik intesitas tetangga ke-*n*,
- t* = *threshold*

Pada pengimplementasian *FAST Corner Detection* ini menggunakan *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<sub>p</sub>*, *y<sub>p</sub>*) pada suatu *digital*. Setelah itu,

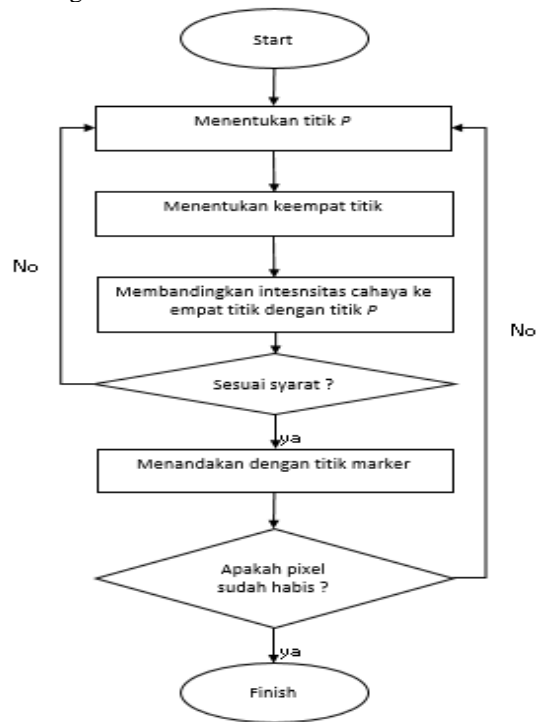
intensitas titik *p* akan dibandingkan dengan 4 titik disekitarnya.



Sumber : Penelitian

**Gambar 1. Deteksi sudut pertitik FAST Corner Detection**

Gambar 1 menunjukkan titik pertama yang akan dibandingkan adalah pada koordinat (*x*, *y<sub>p</sub>-3*). Titik kedua terletak pada koodinat (*x<sub>p</sub>+3*, *y*). Titik koordinat ketiga terletak pada koordinat (*x*, *y<sub>p</sub>+3*). Sedangkan titik ke empat terletak pada koordinat (*x<sub>p</sub>-3*, *y<sub>p</sub>*). Titik *p* akan menjadi sudut (*corner*) apabila intensitas titik *p* bernilai lebih besar atau lebih kecil daripada intensitas sedikitnya tiga titik di sekitarnya. Proses ini tidak berhenti di sini, melainkan titik *p* akan digeser ke posisi selanjutnya (+1) dan titik tersebut akan dibandingkan lagi dengan empat titik di sekitarnya. Proses ini akan terjadi sampai semua titik pada citra digital dibandingkan.



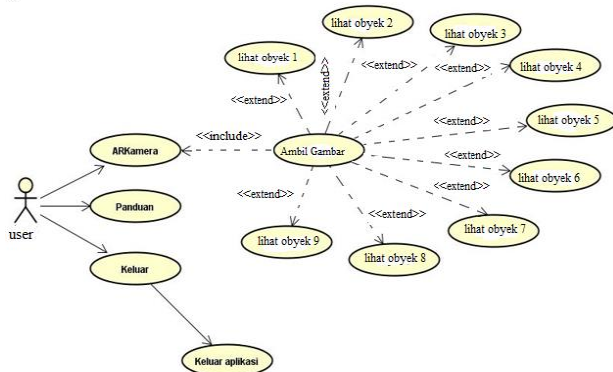
**Gambar 2. Diagram alir algoritma FAST Corner Detection**

Gambar 2, merupakan diagram alir algoritma *FAST Corner Detection*, adapun penjelasan tahap proses *FAST Corne Detection* adalah sebagai 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 ke empat (*n=4*) terletak pada koordinat (-3).

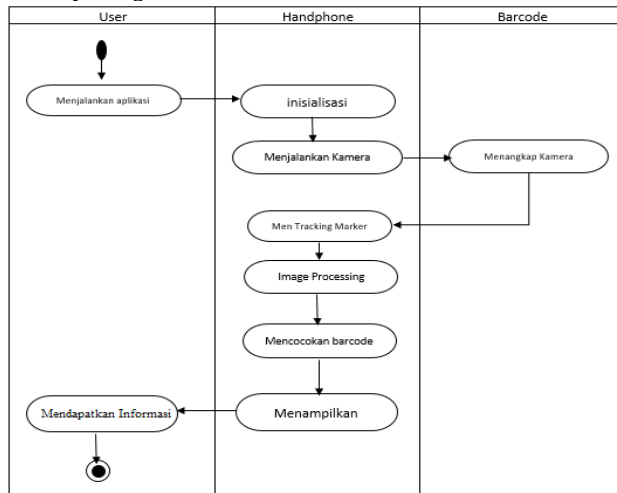
3. Setelah keempat titik ditemukan, langkah selanjutnya bandingkan intensitas titik pusat  $p$  dengan ke empat titik di sekitar. Jika terdapat paling sedikit 3 titik yang memenuhi syarat berikut, maka titik pusat  $p$ .
4. Mengecek apakah masih ada *pixel* yang perlu di cek kembali. Jika tidak ada, hentikan *proses*. Jika ada, maka melanjutkan *proses* selanjutnya.
5. Mengulangi *proses* 1-3 sampai seluruh titik pada citra *digital* sudah dibandingkan intensitasnya.

**Software Architecture  
Usecase Diagram**



**Gambar 3. Usecase Aplikasi Pembelajaran**

**Activity Diagram**

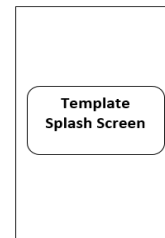


**Gambar 4. Activity Aplikasi Pembelajaran**

**User Interface**

*User Interface* merupakan rancangan tampilan dari aplikasi. Berikut ini adalah perancangan *User Interface* dari *splash screen*, *Menu utama*, *Button ARKamera*, *Panduan* dan *Keluar*.

- a. *Interface Splash screen*  
Tampilan *Splash Screen* merupakan awal dari pembukaan aplikasi. Berikut *Interface* dari *Splash Screen*.

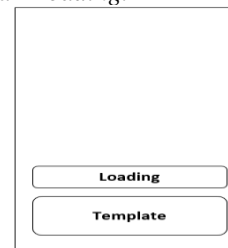


Sumber : Hasil penelitian

**Gambar 5. Interface Splash Screen.**

- b. *Interface Loading*

Tampilan *Interface Loading* merupakan proses ke dua setelah proses *Splash Screen* aplikasi. Berikut *Interface* dari *Loading*.

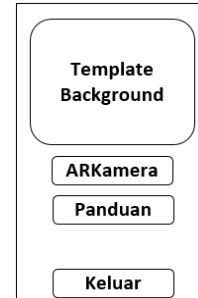


Sumber: Hasil penelitian

**Gambar 6. Interface Loading.**

- c. *Interface Menu Utama*

Setelah selesai tampilan *Splash Screen*. Langsung menuju ke *menu utama*. Pada *menu utama* terdapat tiga *Button*, yaitu *Button ARKamera*, *Button Panduan* dan *Button Keluar*.

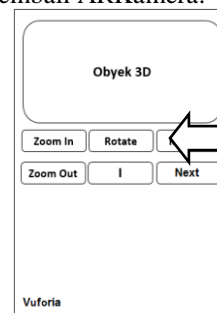


Sumber: Penelitian

**Gambar 7. Interface Menu utama**

- d. *Interface Kembali ARKamera*

*Menu Kembali* aplikasi ini berfungsi untuk keluar setelah menggunakan aplikasi *ARKamera* dengan tujuan menuju *Menu Utama*. Berikut ini gambar *Interface Kembali ARKamera*.

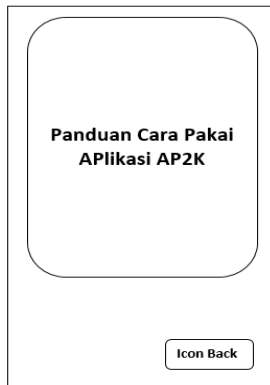


Sumber : Penelitian

**Gambar 8. Interface Kembali ARKamera**

e. *Interface Button Panduan*

Panduan merupakan *scene* cara menggunakan aplikasi yang penulis buat ini. Berikut ini gambar *Interface Button Panduan*.

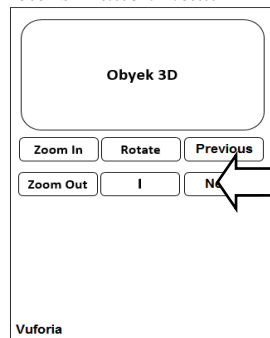


Sumber : Hasil penelitian

**Gambar 9. Interface Button Panduan**

f. *Interfaces Deteksi Button Next*

Deteksi *Button Next* merupakan *scene* ketika menekan *Button Next* untuk melanjutkan pendeteksian dan juga menampilkan obyek 3D dari *QRCode* yang telah terdeteksi. Berikut ini gambar *Interface Deteksi Button Next*.

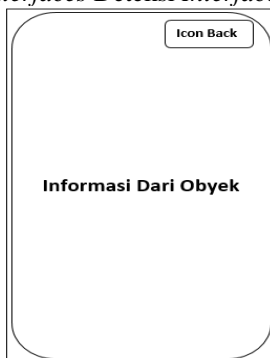


Sumber : Hasil penelitian

**Gambar 10. Interface Button Next.**

g. *Interfaces Info 3D*

Deteksi *Interfaces Info* merupakan *scene* ketika menekan tombol Info untuk melihat informasi tentang obyek 3D yang diinginkan. Berikut ini gambar *Interfaces Deteksi Interfaces Info*.



Sumber : Hasil penelitian

**Gambar 11. Interface Info**

**IV. IMPLEMENTASI DAN PENGUJIAN**

**a. Tampilan *QRCode***

Tampilan *QRCode* ini berisi tentang desain dari *QRCode* atau lebih dikenal sebagai Barcode. Di sini dijelaskan bahwa setelah kamera diarahkan ke media *QRCode* akan menampilkan obyek 3D tentang pengenalan *hardware* komputer. Berikut adalah tampilan dari *QRCode*.



Sumber : Penelitian

**Gambar 12. Tampilan *QRCode***

**b. Tampilan pembuka**

Tampilan pembuka ini berisi *Splash Screen* aplikasi AR pembelajaran pengenalan *hardware* komputer. Pada *Splash Screen* ini terdapat logo nama aplikasi yaitu AP2K. *Splash Screen* ini akan tampil beberapa detik sebelum *Menu Loading*. Berikut ini adalah tampilan dari *Splash Screen* aplikasi AR pembelajaran pengenalan *hardware* komputer.

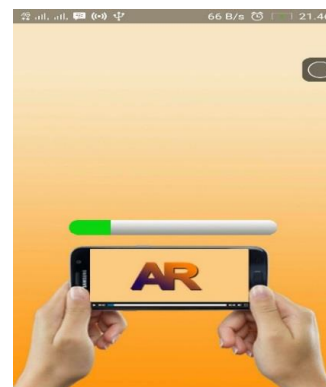


Sumber : Penelitian

**Gambar 13. *Splash Screen* AP2K**

**c. Tampilan *Loading***

Tampilan *Loading* berisi aplikasi AR pembelajaran pengenalan *hardware* komputer. Pada tampilan *Loading* ini terdapat *background* serta tampilan waktu *Loading*. Tampilan *Loading* ini akan tampil beberapa detik sebelum menuju ke *Menu* utama.



Sumber : Penelitian

**Gambar 14. Tampilan *Loading***

**d. Tampilan Menu Utama**

Setelah *Loading*, aplikasi akan menampilkan *menu* utama. Pada *menu* utama terdapat 3 *Button*, yaitu *Button* ARKamera, Panduan dan Keluar. Desain tampilan *menu* utama ini dibuat sederhana namun tetap menarik untuk *user*. Berikut ini adalah tampilan dari *menu* utama aplikasi AR pembelajaran pengenalan *hardware* komputer.



Sumber : Penelitian

**Gambar 15. Menu Utama**

**e. Tampilan Button Panduan**

*Menu* Panduan terdapat pada *menu* utama. *Menu* *Button* Panduan berisi penjelasan tata cara penggunaan aplikasi AP2K serta disediakan *Button* *Icon* *Back* pada *scene* Panduan. berikut ini adalah tampilan dari *menu* *Button* Panduan.

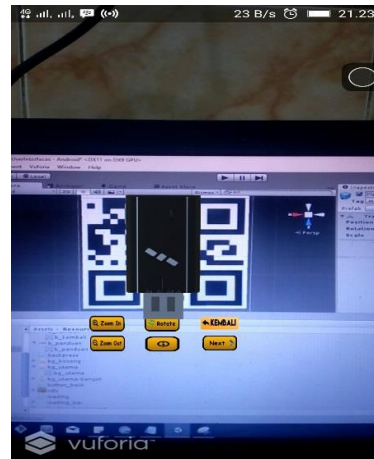


Sumber : Penelitian

**Gambar 16. Tampilan Button Panduan**

**f. Tampilan Deteksi QRCode Scene 0**

Tampilan selanjutnya adalah tampilan deteksi *QRCode* *Scene*. Tampilan ini terbuka ketika *user* menekan *Button* ARKamera pada *menu* utama. Setelah menekan *Button* ARKamera, maka kamera *smartphone* akan aktif. Deteksi *QRCode* ini digunakan untuk mendeteksi sebuah *marker* yang ditunjukkan oleh *user*. Terdapat beberapa *Button* yang akan tampil ketika kita mengarahkan kamera kearah *QRCode*, yaitu *Zoom In*, *Zoom Out*, *Rotate*, *Info*, *Kembali*, *Next* serta *Previous*. Berikut adalah tampilan deteksi *QRCode* ketika kamera aktif. Dan di sini terdapat pula *Button* *Kembali* yang fungsinya untuk mengarahkan *user* untuk kembali ke *main menu*. Berikut tampilan Deteksi *QRCode*.



Sumber : Penelitian

**Gambar 17. Tampilan Deteksi QRCode Scene 0 (Flashdisk)**

**g. Tampilan QRCode Button Info**

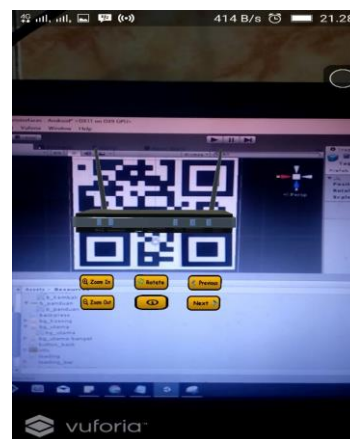
Di setiap *scene* terdapat *Button* *Info*, sesuai dengan namanya fungsi dari *Button* *Info* adalah memberikan informasi tentang obyek yang diinginkan. Berikut adalah *Tampilan* *QRCode* *Button* *Info*.



Sumber : Penelitian

**Gambar 18. Tampilan QRCode Button Info (Flashdisk)**

**h. Tampilan Deteksi QRCode Button Next Scene 1**



Sumber : Penelitian

**Gambar 19. Tampilan Deteksi QRCode Button Next (Router)**

Tampilan Deteksi *QR Code Button Next* ini bermaksud untuk menuju ke *scene* berikutnya. Gambar 19 adalah tampilan Deteksi *QR Code Button Next*.

### Pengujian *Black Box*

Pengujian *black box* yaitu pengujian perangkat lunak dari segi spesifikasi fungsional tanpa pengujian desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, melakukan masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

**Tabel 2. Pengujian *Black Box* terhadap Aplikasi**

| Skenario   | Bagian Yang Diuji | Tindakan Pengujian   | Fungsi Sistem                      | Hasil Yang Diinginkan       | Status |
|------------|-------------------|----------------------|------------------------------------|-----------------------------|--------|
| Object AR  | Tombol Next       | Klik Tombol Next     | Untuk ke object berikutnya         | Tampil ke object berikutnya | Sukses |
| Object AR  | Tombol Zoom In    | Klik Tombol Zoom In  | Object di perbesar                 | Tampilan Object di perbesar | Sukses |
| Object AR  | Tombol Zoom Out   | Klik Tombol Zoom Out | Object di perkecil                 | Tampilan Object di perkecil | Sukses |
| Object AR  | Tombol Previous   | Klik Tombol Previous | Untuk ke object sebelumnya         | Tampil ke object sebelumnya | Sukses |
| Object AR  | Tombol Info       | Klik Tombol Info     | Untuk menampilkan info dari object | Tampil Info dari object     | Sukses |
| Object AR  | Tombol Rotate     | Klik Tombol Rotate   | Untuk merotasi dari object         | Tampil rotasi dari object   | Sukses |
| Object AR  | Tombol Kembali    | Klik Tombol kembali  | Untuk kembali ke menu utama        | Tampil menu utama           | Sukses |
| Menu Utama | Tombol Panduan    | Klik Tombol Panduan  | Untuk menampilkan panduan aplikasi | Tampil panduan aplikasi     | Sukses |
| Menu Utama | Tombol QR Code    | Klik Tombol QR Code  | Untuk Mengambil Barcode            | Tampil ambil gambar barcode | Sukses |
| Menu Utama | Tombol Keluar     | Klik Tombol Keluar   | Untuk keluar dari aplikasi         | Keluar dari aplikasi        | Sukses |

## V. KESIMPULAN DAN SARAN

### Kesimpulan

Berdasarkan uraian di atas maka dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi yang dibangun dapat menjadi media pembelajaran yang menarik dengan menggunakan efek 3D.
2. Penerapan *Augmented Reality* pada aplikasi ini sudah cukup baik.
3. Aplikasi media pembelajaran ini dapat digunakan kapan saja dan di mana saja.

### Saran

Berikut adalah saran untuk pengembangan sistem selanjutnya:

1. Diharapkan penelitian berikutnya dapat lebih menarik lagi dengan adanya penambahan suara pada aplikasi.
2. Jumlah obyek 3D, masih sedikit, diharapkan dapat diperbanyak lagi agar informasi yang didapat juga semakin luas.

## DAFTAR PUSTAKA

- [1] Wirawan, P. A. 2013. *mblender video*. yogyakarta: panduaji.net (May 12, 2017).
- [2] Safaat, Nazruddin. 2012. *Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC*. Bandung. Penerbit Informatika.
- [3] Furrh, Borko. 2011. *Handbook of Augmented Reality*. Florida. Springer.
- [4] Rickman, R. 2014. *Unity Tutorial Game Engine*. Bandung: Informatika Bandung.
- [5] Rizky, Soetam. 2011. *Konsep Dasar Rekayasa Perangkat Lunak*. Jakarta. Prestasi Pustakaraya
- [6] A. S., Rosa dan Shalahuddin, M. 2014. *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung. Informatika.
- [7] A. S., Rosa dan Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung. Informatika.