

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

1. Berdasarkan penelitian yang dilakukan Ilmawan Mustaqim (2017) dengan judul "Pengembangan Media Pembelajaran Berbasis *Augmented Reality*", Media pembelajaran yang menarik juga sangat diperlukan bagi siswa SMK, dikarenakan dalam pembelajaran SMK lebih mengutamakan praktikum dari pada teori. Perlu dipertimbangkan dalam pemilihan media yaitu tujuan pembelajaran, efektif, mudah diperoleh, peserta didik, penggunaan, tidak kaku, biaya, dan kualitas. Salah satu perkembangan media pembelajaran yang saat ini masih baru adalah media pembelajaran dengan menggunakan *Augmented Reality*. *Augmented Reality* merupakan aplikasi penggabungan dunia nyata dengan dunia maya dalam bentuk dua dimensi maupun tiga dimensi yang diproyeksikan dalam sebuah lingkungan nyata dalam waktu yang bersamaan. *Augmented Reality* dapat digunakan dalam hiburan, kedokteran, mekanik, dan media pembelajaran.
2. Berdasarkan penelitian yang dilakukan Muntahanah (2017) dengan judul "Penerapan Teknologi *Augmented Reality* Pada Katalog Rumah Berbasis Android (Studi Kasus PT. Jashando Han Saputra)". Penelitian ini akan menerapkan teknologi AR kedalam katalog rumah berbasis android. AR

Katalog rumah berbasis android ini memerlukan *video streaming* yang diambil dari kamera sebagai sumber masukan, kemudian aplikasi ini akan melacak dan mendeteksi marker (penanda) dengan menggunakan sistem *tracking*, setelah marker dideteksi, model rumah 3D muncul di atas *marker* seolah-olah model tersebut nyata. Untuk membuat model rumah ini, model harus dibuat terlebih dahulu dengan menggunakan perangkat lunak untuk membuat objek 3D, dalam kasus ini menggunakan *sweethome 3D*. Pengujian aplikasi ini dilakukan menggunakan kamera *handphone* 5MP, menggunakan 2 jenis *marker*, dan seluruh proses dari pembuatan aplikasi ini menggunakan *Unity 3D*. Dari hasil pengujian, aplikasi ini berjalan dengan baik ketika jarak 27 cm dengan sudut pandang dan memiliki cahaya yang cukup. Itu merupakan proses *tracking* yang ideal.

3. Berdasarkan penelitian yang dilakukan oleh Zwingly Ch Rawis (2018) dengan judul “Penerapan *Augmented Reality* Berbasis Android Untuk Mengenalkan Pakaian Adat Tountemboan” Pada zaman modern masalah yang dihadapi adalah kurangnya media untuk mengenalkan warisan-warisan kebudayaan yang ada, salah satunya Pakaian adat Tountemboan. Karena itu dibuatlah sebuah media informasi yang dapat mengenalkan Pakaian adat Tountemboan. Agar dapat di akses dengan mudah maka aplikasi dibuat berbasis android dan memanfaatkan teknologi *Augmented Reality User Defined Target* yang memungkinkan kita menambah objek virtual pada lingkungan nyata sehingga dapat mudah digunakan. Metode yang digunakan dalam penelitian ini adalah *Multimedia Development Life Cycle*.

4. Berdasarkan penelitian dari Ariawan Djoko Rachmanto (2018) dengan judul “Implementasi *Augmented Reality* Sebagai Media Pengenalan Promosi Universitas Nurtanio Bandung Menggunakan *Unity 3D*” Penelitian ini merupakan upaya untuk menerapkan teknologi *Augmented Reality* sebagai media promosi. Metode yang digunakan dalam penelitian ini adalah modifikasi dari *Multimedia Development Life Cycle* (MDLC) yang meliputi *Concept, Design, Material Collecting, Assembly, Testing, dan Distribution*. Hasil dari penelitian ini berupa media pengenalan promosi Universitas Nurtanio Bandung khususnya Fakultas Ilmu Komputer dan Informatika, yang dibangun dengan *processing* dari pustaka *Unity 3D*.
5. Berdasarkan Feby Zulham Adami (2018) dengan judul “Penerapan Teknologi *Augmented Reality* Pada Media Pembelajaran Sistem Pencernaan Berbasis Android” *Augmented reality* (AR) adalah jenis teknologi interaktif menggabungkan benda nyata dan virtual yang akan menghasilkan objek 3D yang akan ditampilkan pada layar. *Augmented reality* yang telah diaplikasikan memiliki cara kerja berdasarkan deteksi citra atau gambar dan biasa disebut marker, dengan menggunakan kamera *smartphone* kemudian mendeteksi marker yang telah di cetak. *Augmented reality* banyak digunakan diberbagai bidang, salah satunya bidang pendidikan. Pada bidang pendidikan *augmented reality* digunakan sebagai media pembelajaran agar lebih menarik.

II.2. Landasan Teori

II.2.1. Komputer Grafik

Komputer grafik berkaitan dengan ilmu yang menghasilkan gambar dan animasi (atau urutan gambar) dengan menggunakan komputer. Hal ini mencakup sistem perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan untuk membuat gambar. Menghasilkan gambar foto-realistis adalah salah satu yang sangat kompleks dan bidang yang sangat diminati, karena hampir tak terbatas dari aplikasi. Bidang komputer grafik telah berkembang pesat selama 10-20 tahun terakhir, dan banyak sistem perangkat lunak telah dikembangkan untuk menghasilkan berbagai macam komputer grafik. (Sulistyo, 2018).

Komputer grafik melalui pemrograman grafik menghasilkan citra dari bentuk geometri yang primitive seperti titik, garis lurus dan garis lengkung. Lingkaran dan bentuk - bentuk dasar geometri lainnya. Komputer grafik memainkan peranan penting dalam visualisasi dan *virtual-reality*. Sedangkan sistem visual bekerja sebaliknya, menduga bentuk geometri primitive dan ciri lainnya yang merupakan penyederhanaan dari citra asal yang sifatnya lebih kompleks. (Lumenta, 2018).

II.2.2. Augmented Reality

Augmented Reality (AR) adalah sebuah istilah untuk lingkungan yang menggabungkan dunia nyata dan dunia virtual serta dibuat oleh komputer sehingga batas antara keduanya menjadi sangat tipis. Tujuan utama dari AR adalah untuk menciptakan lingkungan baru dengan menggabungkan interaktivitas

lingkungan nyata dan virtual sehingga pengguna merasa bahwa lingkungan yang diciptakan adalah nyata. Dengan kata lain, pengguna merasa tidak ada perbedaan yang dirasakan antara AR dengan apa yang mereka lihat/rasakan di lingkungan nyata. Dengan bantuan teknologi AR (seperti visi komputasi dan pengenalan pola) lingkungan nyata di sekitar kita akan dapat berinteraksi dalam bentuk digital (virtual). Informasi tentang objek dan lingkungan di sekitar kita dapat ditambahkan kedalam sistem AR yang kemudian informasi tersebut ditampilkan di atas layar dunia nyata secara *real-time* seolah-olah informasi tersebut adalah nyata. Informasi yang ditampilkan oleh subjek virtual membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata. Gambar di bawah ini merupakan *augmented reality*, yang lebih dekat ke sisi kiri, lingkungan bersifat nyata dan benda bersifat maya, sementara dalam *augmented reality*, yang lebih dekat ke sisi kanan, lingkungan bersifat maya dan benda bersifat maya. (Ariawan : 2018)

Konsep *Augmented Reality* pertama kali dikemukakan oleh Ronald T. Azuma pada tahun 1997 saat ia bekerja di perusahaan Boeing. Ada tiga karakteristik yang menyatakan suatu teknologi menerapkan konsep AR yaitu :

- a. Mampu mengkombinasikan dunia nyata dan dunia maya,
- b. Mampu memberikan informasi secara interaktif dan *realtime*,
- c. Mampu menampilkan dalam bentuk 3D (tiga dimensi).

Tujuan utama dari AR adalah untuk menciptakan lingkungan baru dengan menggabungkan interaktivitas lingkungan nyata dan virtual sehingga pengguna merasa bahwa lingkungan yang diciptakan adalah nyata. Dengan bantuan

teknologi AR (seperti visi komputasi dan pengenalan objek) lingkungan nyata disekitar kita akan dapat berinteraksi dalam bentuk digital (*virtual*). Informasi tentang objek dan lingkungan disekitar kita dapat ditambahkan ke dalam sistem AR yang kemudian informasi tersebut ditampilkan diatas *layer* dunia nyata secara *real-time* seolah-olah informasi tersebut adalah nyata. Informasi yang ditampilkan oleh objek *virtual* membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata (Rumanjar. R, Lumenta. A dan Sugiarto B. A, 2015).

AR memiliki tiga karakteristik yaitu bersifat interaktif (meningkatkan interaksi dan persepsi pengguna dengan dunia nyata), menurut waktu nyata (*real time*) dan berbentuk 3 dimensi. Terdapat dua metode AR yang dikembangkan saat ini yaitu *Marker Based Tracking* dan *Markless AR*. Disini penulis menggunakan metode *Marker Based Tracking*. *Marker Based Tracking* adalah AR yang menggunakan *marker* atau penanda objek dua dimensi yang memiliki suatu pola yang akan dibaca komputer melalui media webcam atau kamera yang tersambung dengan komputer (Haryani. P dan Triyono. J, 2017).

Sistem dalam *Augmented Reality* bekerja dengan menganalisa secara *real-time* obyek yang ditangkap dalam kamera. Berkat perkembangan pesat teknologi *handphone*, *Augmented Reality* tersebut bisa diimplementasikan pada perangkat yang memiliki GPS, kamera, akselerometer dan kompas. Kombinasi dari ketiga sensor tersebut dapat digunakan untuk menambahkan informasi dari obyek yang ditangkap kamera. Cara kerja AR terbagi 2 macam metode, yaitu:

a. *Marker Augmented Reality (Marker Based Tracking)*

Aplikasi *augmented* ini berjalan dengan memindai tanda atau yang lebih sering disebut sebagai *marker*. *Marker* biasanya merupakan ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. Komputer akan mengenali posisi dan orientasi *marker* dan menciptakan dunia *virtual* 3D yaitu titik (0,0,0) dan 3 sumbu yaitu X,Y,dan Z.

b. *Augmented Reality tanpa Marker (Markerless Augmented Reality)*

Salah satu metode *Augmented Reality* yang saat ini sedang berkembang adalah metode "*Markerless Augmented Reality*", dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan elemen-elemen *digital*. Apakah *Markerless AR* dalam menjalankan aplikasi tidak melakukan pemindaian *marker*? Sekalipun dinamakan dengan *markerless* namun aplikasi tetap berjalan dengan melakukan pemindaian terhadap *object*, namun ruang lingkup yang dipindai lebih luas dibanding dengan *marker AR*. Seperti yang saat ini dikembangkan oleh perusahaan *Augmented Reality* terbesar di dunia Total Immersion, mereka telah membuat berbagai macam teknik *Markerless Tracking*. terdapat beberapa teknik *markerless* yaitu:

1. *Face Tracking*

Dengan menggunakan algoritma yang mereka kembangkan, komputer dapat mengenali wajah manusia secara umum dengan cara mengenali posisi mata, hidung, dan mulut manusia, kemudian akan mengabaikan objek-objek lain di sekitarnya seperti pohon, rumah, dan benda-benda lainnya.

2. *3D Object Tracking*

Berbeda dengan *Face Tracking* yang hanya mengenali wajah manusia secara umum, teknik *3D Object Tracking* dapat mengenali semua bentuk benda yang ada disekitar, seperti mobil, meja, televisi, dan lain-lain.

3. *Motion Tracking*

Pada teknik ini komputer dapat menangkap gerakan, *Motion Tracking* telah mulai digunakan secara ekstensif untuk memproduksi *film-film* yang mencoba mensimulasikan gerakan.

4. *GPS Based Tracking*

Teknik *GPS Based Tracking* saat ini mulai populer dan banyak dikembangkan pada aplikasi *smartphone* (*iPhone* dan *Android*). Dengan memanfaatkan fitur GPS dan kompas yang ada didalam *smartphone*, aplikasi akan mengambil data dari GPS dan kompas kemudian menampilkannya dalam bentuk arah yang kita inginkan secara *realtime*, bahkan ada beberapa aplikasi menampilkannya dalam bentuk 3D.

II.2.3. Arsitektur *Augmented Reality*

Arsitektur teknologi *Augmented Reality* menurut Indriani. R, Sugiarto. B dan Purwanto. A, (2016) seperti yang terlihat pada Gambar II.1 :



Gambar II.1 *Arsitektur Augmented Reality*

a. *Input*

Pada AR, proses input ini sistem mendeteksi sensor dari benda nyata. Seperti gambar, sensor getaran, lokasi, hingga sensor gerakan.

b. Kamera

Kamera/Alat penangkap sensor lainnya Disini sensor-sensor dari dunia nyata diterima dan dijadikan informasi yang nantinya akan di proses oleh sistem.

c. Processor

Disinilah proses inti dari kera teknologi AR. Pada bagian ini, sistem akan memproses informasi yang masuk dan menemukan informasi apa yang akan dikeluarkan.

d. *Output*

Output, menampilkan informasi-informasi yang sudah ada di proses. *Output* dapat berupa Monitor, Layar Ponsel, dst.

II.2.4. Pahlawan Nasional

Pahlawan Nasional adalah gelar yang diberikan kepada warga negara Indonesia yang berjuang melawan penjajahan yang gugur demi membela bangsa dan negara Indonesia, atau menghasilkan prestasi dan karya yang luar biasa bagi pembangunan dan kemajuan bangsa dan negara Republik Indonesia. Berkat perjuangan yang sangat heroik, Pahlawan Nasional diabadikan gambarnya di dalam uang kertas Indonesia. Namun banyak masyarakat yang tidak mengetahui sejarah dari para Pahlawan Nasional yang terdapat pada uang kertas tersebut. Masyarakat Indonesia hanya menggunakan uang sebagai alat jual beli saja. (Muhammad Hadi Noor Seto : 2018)

II.2.5. Benda Bersejarah

Benda Bersejarah adalah salah satu objek wisata untuk menambah wawasan dan pengetahuan yang didalamnya terdapat benda-benda warisan budaya masa lalu. Museum mampu mengkomunikasikan dan memamerkan benda kepada masyarakat untuk kebutuhan pendidikan dan kesenangan. Salah satu museum yang berada di Jawa Tengah yaitu Museum RA Kartini yang terletak di kabupaten Rembang. Pengunjung umum kurang begitu tertarik mengunjungi museum karena mereka kurang begitu peduli dengan budaya dan sejarah Indonesia. Masyarakat menilai museum adalah tempat yang membosankan dan kurang menarik untuk dikunjungi karena sebagian orang beranggapan bahwa museum sebagai tempat yang kuno jauh dari sentuhan teknologi. (Indriana Novitasari : 2020)

II.2.6. Android

Android merupakan salah satu sistem operasi berbasis *linux* pada *smartphone* yang banyak digunakan karena memiliki berbagai fungsi yang dapat memudahkan pekerjaan. Sistem operasi *Android* dapat digambarkan sebagai jembatan antara *smartphone* dan penggunanya, sehingga pengguna dapat mengoperasikan *smartphone* dan dapat menggunakan aplikasi yang tersedia. *Smartphone* berbasis *Android* sangat membantu *user* dalam melakukan berbagai aktivitas seperti berkomunikasi, berbelanja secara *online*, presentasi, dan lain sebagainya. Keunggulan *system* operasi *Android* dibanding dengan sistem operasi *smartphone* lainnya adalah *Android* bersifat *open source code* sehingga memudahkan para pengembang untuk membuat dan memodifikasi aplikasi atau fitur-fitur yang belum terdapat pada sistem operasi *Android* sesuai dengan kebutuhan. (Sendi Herlambang, 2018).

Sejarah sistem operasi *Android* dimulai sejak diumumkannya *Open Handset Alliance* di akhir tahun 2007. Sebenarnya ide untuk membuat sistem operasi *open source* buat peranti *embedded* sudah ada sejak lama. Karena *backing* dari Google yang sangat agresiflah, maka *Android* bisa sangat terkenal hanya dalam beberapa tahun saja. Beberapa perusahaan telekomunikasi kini mulai memiliki atau menawarkan perangkat *android* dalam produk – produknya. Tidak hanya ponsel, tapi perangkat lain seperti *tablet*, *notebook*, televisi, bahkan hingga ke peranti *embedded* di *mobile* memiliki sistem operasi *android*. Sebenarnya *android* pada awalnya tidak dikembangkan oleh google, tapi dikembangkan oleh sebuah perusahaan bernama *Android.Inc*. Karena google

melihat banyaknya *user* yang *online* dengan perangkat *mobile*, maka google mengira bahwa perangkat *mobile* ini memiliki masa depan yang cerah. Sehingga *Android Inc* diakuisisi oleh Google di tahun 2005. Pada waktu yang sama, *Apple* di tahun 2007 juga mengembangkan sebuah sistem operasi *iOS* untuk *iPhone* – nya dengan fitur – fitur yang sangat inovatif, seperti *multitouch* dan adanya *open market* untuk aplikasi.

Android pun kemudian meniru fitur ini dan cepat beradaptasi untuk menambahkan fitur ini dan fitur tambahan lainnya, seperti *programmer* bisa punya kontrol yang lebih terhadap sistem operasi, dan *multitasking*. *Android* punya banyak tambahan yang cocok untuk dunia korporat, seperti *exchange support*, *remote wipe*, dan *Virtual Private Network (VPN)*. Ini mirip dengan *BlackBery* yang dikembangkan oleh *RIM*. Dengan beragamnya peranti dan adaptasi, *android* berkembang pesat dan jumlah *user* – nya pun bertambah. Tapi bagi *developer*, ini juga berarti ada tantangan yang potensial. Karena aplikasi harus bisa adaptasi dengan ukuran layar dan resolusi yang berbeda, ada juga *keyboard*, *sensor hardware*, versi *OS*, konfigurasi sistem, dan kecepatan data *wireless* yang berbeda – beda.

Tiap variasi dari aspek – aspek tersebut bisa menyebabkan munculnya perilaku yang tidak terprediksi. Namun mengecek satu – satu untuk variasi tersebut juga tidak memungkinkan. Karena jumlahnya yang sangat banyak. Karena itu, sejak awal *Android* sudah diatur agar bisa memiliki tampilan yang seragam, dan *user experience* yang sama di semua *platform*. Caranya dengan melakukan abstraksi pada perbedaan *hardware*. Sistem operasi *Android* menutup

aplikasi dari modifikasi yang *hardware specific*. Tapi di sisi lain juga memungkinkan fleksibilitas untuk melakukan *tune up* pada aspek yang diperlukan. Google menyediakan banyak kemudahan untuk pengembang *software* pihak ketiga, seperti *plugin Android Development Tool* (ADT) untuk *Eclipse* (dan juga *tool standalone*) termasuk di dalamnya kemampuan untuk *logging* secara *realtime*, *emulator* realistis yang menjalankan kode *ARM native* dan pelaporan *error*.

II.2.7 Tools Pembangunan Android

Untuk membangun sebuah sistem operasi *Android* dapat menggunakan Mac, *Windows* PC, ataupun *Linux*. *Tools* yang dibutuhkan gratis dan dapat di *download* dari *web*. Berikut adalah beberapa *tools* yang digunakan untuk membangun aplikasi *android*.

1. JDK (*Java Development Kit*)
2. Android SDK
3. ADT (*Android Development Tools*) (Ni Kadek Ceryna, dkk : 2018 : 102)

II.2.8. Perkembangan Android

Adapun perkembangan Android adalah sebagai berikut :

a. Android versi 1.1

Android memang diluncurkan pertama kali pada tahun 2007, namun sistem operasi ini mulai dirilis dan diterapkan ke berbagai gadget pada tanggal 9 Maret 2009 silam. Android versi 1.1 merupakan Android awal yang dimana versi ini baru memberikan sentuhan di beberapa aplikasinya seperti sistem antar

muka bagi pengguna (user interface) yang lebih baik, serta beberapa aplikasi yang lain.

b. Android versi 1.5 (Cupcake)

Pada bulan Mei 2009 Android kembali mengalami perubahan versi. Android versi 1.1 kemudian disempurnakan dengan Android versi 1.5 atau yang dikenal sebagai Android Cupcake.

c. Android versi 1.6 (Donut)

Donut (versi 1.6) diluncurkan dalam tempo kurang dari 4 bulan semenjak peluncuran perdana Android Cupcake, yaitu pada bulan September 2009.

d. Android versi 2.0/2.1 (Eclair)

Masih ditahun yang sama, Android kembali merilis operating sistem versi terbarunya, yaitu Android versi 2.0/2.1 Eclair. Android Eclair diluncurkan oleh Google 3 bulan setelah peluncuran.

e. Android versi 2.2 (Froyo: Frozen Yoghurt)

Butuh 5 bulan bagi Google untuk melakukan regenerasi dari Android Eclair versi sebelumnya ke versi Froyo Frozen Yoghurt. Pada tanggal 20 Mei 2010, Android versi 2.2 alias Android Froyo ini dirilis.

f. Android versi 2.3 (Gingerbread)

7 bulan kemudian Android kembali melakukan gebrakan dengan merilis kembali Android versi 2.3 atau yang dikenal sebagai Android Gingerbread.

g. Android versi 3.0/3.1 (Honeycomb)

Pada bulan Mei 2011 Android versi 3.0/3.1 atau Android Honeycom dirilis. Android Honeycomb merupakan sebuah sistem operasi Android yang tujuannya memang dikhususkan bagi penggunaan tablet berbasis Android.

h. Android versi 4.0 (ICS: Ice Cream Sandwich)

Android ICS atau Ice Cream Sandwich juga dirilis pada tahun yang sama dengan Honeycomb, yaitu pada bulan Oktober 2011.

i. Android versi 4.1 (Jelly Bean)

Android Jelly Bean merupakan versi Android yang terbaru pada saat ini. Salah satu gadget yang menggunakan sistem operasi Jelly Bean adalah Google Nexus 7 yang diprakarsai oleh ASUS, vendor asal Taiwan yang juga menjadi teman satu kampung halaman dengan Acer.

j. Android versi 4.4 (Kit Kat)

Kehadiran android kitkat merupakan peluncuran produk OS anyar yang diluncurkan pada 4 september 2013, sebelumnya banyak kabar beredar jikalau android akan meluncurkan OS baru yang bernama Android Key Lime Pie namun setelah di analisa tidak sesuai dengan ejaan orang umum, sehingga namanya diganti dengan OS Android KitKat yang sebagian besar orang sudah familiar dengan itu.

k. Android versi 5.0.2 (Lollipop)

Android Lollipop merupakan keberadaan OS Android yang memang saat ini sudah menjadi trend baru di industri smartphone, hal ini tak lepas dari keunikan dan kelebihan yang banyak di miliki dari OS tersebut. Kehadiran

android versi ini amat di nanti oleh sekian banyak orang karna diharapkan sistem operasi Lollipop ini bias lebih baik dibandingkan versi-versi sebelumnya.

1. Android versi 6.0 (Marshmallow)

Android 6.0 Marshmallow adalah versi dari sistem operasi mobile Android. Pertama kali diperkenalkan Mei 2015 di Google I / O di bawah kode nama Android M, secara resmi dirilis pada Oktober 2015. Android Marshmallow memperkenalkan model izin aplikasi didesain ulang sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi. (Harni Kusniyati Dan Dicky : 2018)

II.2.9. Vuforia

Vuforia adalah *Augmented Reality Software Development Kit* (SDK) untuk perangkat *mobile* yang memungkinkan pembuatan aplikasi *Augmented Reality*. Dulunya lebih dikenal dengan QCAR (*Qualcomm Company Augmentend Reality*). QCAR menggunakan teknologi *Computer Vision* untuk mengenali dan melacak gambar planar (Target Image) dan objek 3D sederhana, seperti kotak, secara *real-time* (Adnin.S.N, Widiartha I.B.K dan Suksmadana.I.M.B, 2016). *Vuforia* merupakan SDK yang disediakan oleh *Qualcomm* untuk membantu para *developer* membuat aplikasi-aplikasi *Augmented Reality* (AR) di *mobile phones* (iOS, *Android*) SDK. AR *Vuforia* memberikan cara berinteraksi yang memanfaatkan kamera *mobile phone* untuk digunakan sebagai perangkat masukan, sebagai mata elektronik yang mengenali penanda tertentu, sehingga di

layar bisa ditampilkan perpaduan antara dunia nyata dan dunia yang digambar oleh aplikasi. *Vuforia Qualcomm* merupakan *library* yang digunakan sebagai pendukung adanya AR pada *Android* untuk program *Unity 3D* (Mahendra, I. B. M. 2016).

II.2.9.1. Arsitektur *Vuforia*

Vuforia SDK memerlukan beberapa komponen penting agar dapat bekerja dengan baik. Komponen – komponen tersebut adalah :

a. Kamera

Kamera dibutuhkan untuk memastikan bahwa setiap *frame* ditangkap dan diteruskan secara efisien ke *tracker*. Para *developer* hanya tinggal memberi tahu kamera kapan mereka mulai menangkap dan berhenti.

b. *Image Converter*

Mengkonversi *format* kamera (misalnya YUV12) kedalam *format* yang dapat dideteksi oleh *OpenGL* (misalnya RGB565) dan untuk *tracking* (misalnya *luminance*).

c. *Tracker*

Mengandung algoritma *computer vision* yang dapat mendeteksi dan melacak objek dunia nyata yang ada pada *video* kamera. Berdasarkan gambar dari kamera, algoritma yang berbeda bertugas untuk mendeteksi *trackable* baru, dan mengevaluasi *virtual button*. Hasilnya akan disimpan dalam *state object* yang akan digunakan oleh *video background renderer* dan dapat diakses dari *application code*.

d. *Video Background Renderer*

Me-render gambar dari kamera yang tersimpan di dalam *state object*. Performa dari *video background renderer* sangat bergantung pada *device* yang digunakan.

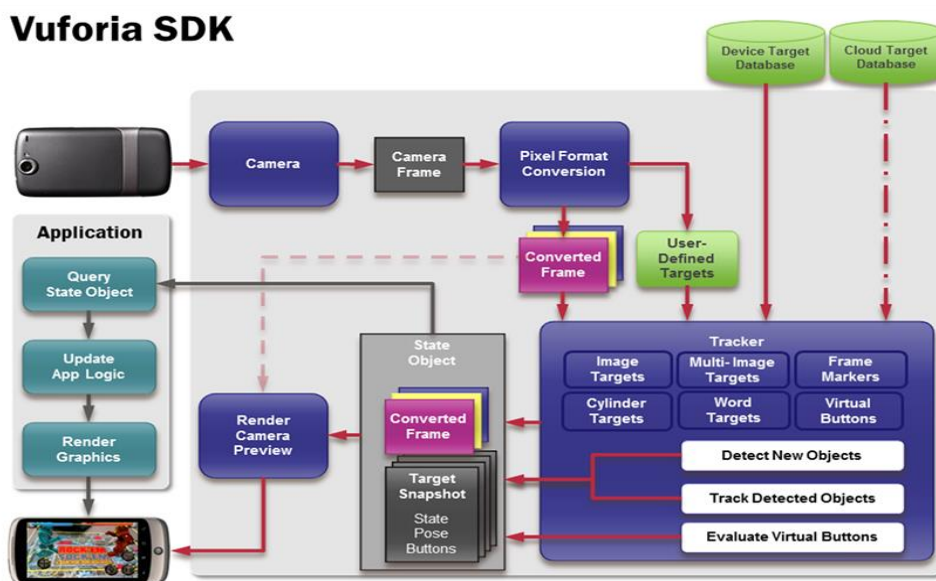
e. *Application Code*

Menginisialisasi semua komponen di atas dan melakukan tiga tahapan penting dalam *application code* seperti :

1. *Query state object* pada *target* baru yang terdeteksi atau *marker*.
2. *Update* logika setiap *input* baru dimasukkan.
3. *Render* grafis yang ditambahkan (*augmented*).

f. *Target Resources*

Dibuat menggunakan *online Target Management System*. *Assets* yang diunduh berisi sebuah konfigurasi xml (*config.xml*) yang memungkinkan *developer* untuk mengkonfigurasi beberapa fitur dalam *trackable* dan *binary file* yang berisi *datastrackable*. *Diagram* aliran *Data Vuforia* seperti yang terlihat pada Gambar II.2 :



Gambar II.2. Diagram Aliran Data Vuforia

II.2.10 Unity

Unity adalah perangkat lunak *game engine* untuk membangun permainan 3 Dimensi (3D). *Game engine* merupakan komponen yang ada dibalik layar setiap *video game*. *Mesh* merupakan bentuk dasar dari objek 3D. Pembuatan *mesh* tidak dilakukan pada Unity. Sementara *GameObjects* adalah konten untuk semua komponen lainnya. Semua objek dalam permainan disebut *GameObjects*. Material digunakan dan dihubungkan dengan *mesh* atau *renderer* partikel yang melekat pada *GameObjects*. Material berhubungan dengan penyaji *mesh* atau partikel yang melekat pada *GameObjects* tersebut. Mereka memainkan bagian penting dalam mendefinisikan bagaimana objek ditampilkan. *Mesh* atau partikel tidak dapat ditampilkan tanpa material karena material meliputi referensi untuk *shader* yang digunakan untuk menempatkan tekstur ke *GameObjects*. Unity mendukung pengembangan aplikasi Android. Sebelum dapat menjalankan aplikasi yang dibuat

dengan *unity* Android diperlukan pengaturan lingkungan pengembang Android pada perangkat. Untuk itu pengembang perlu mendownload dan menginstal SDK Android dan menambahkan perangkat fisik ke *system*. *Unity Android* memungkinkan pemanggilan fungsi *custome* yang ditulis dalam C/C++ secara langsung dan java secara tidak langsung dari script C# (Rumanjar. R, Lumenta. A dan Sugiarto B. A, 2015).

II.2.11. Unity 3D Engine

Unity 3D Engine adalah perangkat lunak *game engine* untuk membangun permainan 3 Dimensi (3D). *Game engine* merupakan komponen yang ada di balik layar dasar dari objek 3D. Pembuatan mesh tidak dilakukan pada *Unity*. Sementara *game object* adalah konten untuk semua komponen lainnya. Semua objek dalam permainan disebut *game objects*. (Ariawan : 2018)

II.2.12. Unified Modeling Language (UML)

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak (Urva. G dan Siregar. H. F, 2015p).

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar

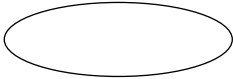
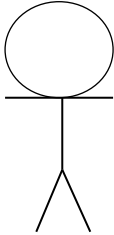

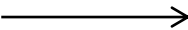
bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem (Urva. G dan Siregar. H. F, 2015p).

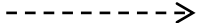
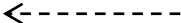
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram*, yaitu :

Tabel II.1. Simbol Use Case

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i>
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.




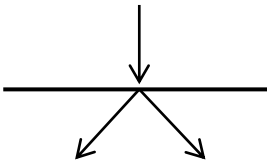
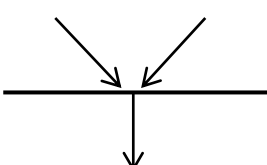
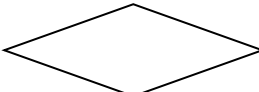
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

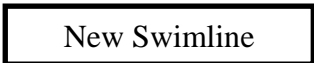


(Sumber : Ade Hendini, 2016 : 110)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .

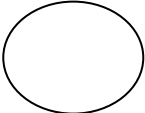
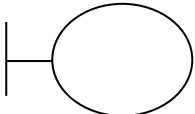

	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.
	<i>Signal</i> sebagai pengirim pesan dari aktifitas yang terjadi. <i>Signal</i> pengirim yang digambarkan dengan <i>convex poligon</i> .
	<i>Signal</i> sebagai penerima pesan dari aktifitas yang terjadi. <i>Signal</i> penerima yang digambarkan dengan <i>poligon</i> terbuka.

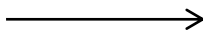
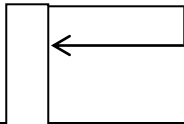


(Sumber : Ade Hendini, 2016 : 110)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas,

	contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini, 2016 : 110)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini, 2016 : 110)