

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian tentang Augmented Reality (AR) telah banyak dilakukan, diantaranya adalah dari (Mustofa Arifin Tri Nugroho 2016). Metode *waterfall*. Teknologi *augmented reality* dapat digunakan sebagai sarana untuk mengenalkan wayang kepada masyarakat dengan cara yang lebih mudah dan lebih menarik. Aplikasi ini akan menampilkan gambar 3D dan juga memberikan informasi dari masing-masing wayang yang terdeteksi.

Ada juga penelitian tentang AR yang dilakukan oleh (Rifa'i, Listyorini and Latubessy 2014), dengan memasukkan teknologi AR kedalam katalog penjualan rumah pada Perumahan Muna Permai, sehingga katalog rumah ini menjadi lebih real dengan adanya objek 3D pada rumah. Dan penelitian ini di jalankan pada platform mobile android, dimana aplikasi AR ini memerlukan video streaming yang diambil dari kamera smartphone sebagai sumber masukan, kemudian aplikasi ini akan melacak dan mendeteksi marker (penanda) dengan menggunakan sistem tracking, setelah marker terdeteksi, model rumah 3D pada 8 katalog akan muncul diatas marker seolah-olah model rumah tersebut nyata.

Penelitian selanjutnya dilakukan oleh (Ratnawati 2016), tujuan utama dari penelitian ini untuk melakukan pengembangan aplikasi profil sekolah berbasis Augmented Reality sebagai media informasi profil sekolah, dengan menggunakan model pengembangan model waterfall yang terdiri dari analisis, desain, implementasi dan pengujian.

2.2 Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi atau pernyataan yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output (Jogiyanto, 2005).

Menurut Kamus Kamus Besar Bahasa Indonesia (2001 : 52), “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu”.

Dari pengertian diatas, maka dapat diambil kesimpulan bahwa aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer.

2.3 Augmented Reality

Ronald T. Azuma (1997) mendefinisikan *Augmented reality* sebagai penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antarbenda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat *input* tertentu, dan integrasi yang baik memerlukan penjejukan yang efektif.

Anggi Andriyadi (2011: 10), mengemukakan bahwa bidang-bidang yang pernah menerapkan teknologi Augmented Reality adalah :

a. Kedokteran (*Medical*)

Teknologi pencitraan sangat dibutuhkan di dunia kedokteran, seperti misalnya, untuk simulasi operasi, simulasi pembuatan vaksin virus, dll.

b. Hiburan (*Entertainment*)

Augmented Reality sekarang sudah dipakai di dunia *entertainment*. Bentuknya beragam, ada yang dipakai untuk efek perfilman, permainan untuk di *smartphone*, majalah, dll

c. Latihan Militer (*Military Training*)

Militer telah menerapkan *Augmented Reality* pada latihan tempur mereka. Sebagai contoh, militer menggunakan *Augmented Reality* untuk membuat sebuah permainan perang dimana prajurit masuk kedalam dunia game tersebut dan seolah-olah seperti melakukan perang sesungguhnya.

d. Engineering

Biasanya *Augmented Reality* digunakan untuk latihan para *Engineer* untuk bereksperimen. Misalnya ahli Engineering Mesin menggunakan *Augmented Reality* untuk memperbaiki mobil yang rusak.

e. Robotics dan Telerobotics

Dalam bidang robotika, seorang operator robot menggunakan pencitraan visual dalam mengendalikan robot itu.

f. Consumer Design

Virtual Reality telah digunakan dalam mempromosikan produk. Sebagai contoh, seorang pengembang menggunakan brosur virtual untuk memberikan informasi yang lengkap secara 3D, sehingga pelanggan dapat mengetahui secara jelas produk yang ditawarkan.

2.4 Android

Android adalah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi (Nazruddin, Safaat, 2012:1). Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. Yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Menurut Priyanta F (2011), android merupakan suatu *software stack* untuk *mobile device*. Di dalamnya terdapat sistem operasi, *middleware*, dan *key application*. Aplikasi pada *platform* ini dikembangkan dalam bahasa pemrograman java. Android sendiri memiliki banyak fitur diantaranya adalah:

- a) Merupakan sebuah *Application Framework* sehingga *programmer* dapat menggunakan beberapa fungsi yang telah disediakan.
- b) *Dalvik virtual machine*. Tiap aplikasi dalam android memiliki *instance virtual machine* yang dapat bekerja secara efisien dalam lingkungan memori terbatas.
- c) *Integrated browser*. Web browser berbasis *webkit engine* terdapat pada *browser default* android atau pun dapat diintegrasikan dengan aplikasi lain.

- d) *Optimized graphics*. Library grafis 2D yang kaya dan 3D berbasis open GLES 1.0 yang mendukung akselerasi hardware.
- e) *SQLite*. Berbasis data relasional yang ringan namun sangat *powerfull*.
- f) *Media Support*. Mendukung berbagai format audio, video, dan gambar (MPEG4, H.264, MP3, ACC, AMR, JPG, PNG, GIF)
- g) *GSM Telephony*. Mendukung fungsi komunikasi GSM.
- h) *Bluetooth, EDGE, 3G dan WIFI*. Mendukung komunikasi pada jaringan (tergantung hardware).
- i) *Kamera, GPS, kompas dan accelerometer*. Mendukung berbagai fitur yang disediakan oleh hardware.
- j) *Tools developer* yang lengkap. Termasuk *device emulator, tools* untuk *debugging, profiling* memori dan perform, dan *plugin*.

2.4.1 Versi Android

Pada penghujung tahun 2010 diperkirakan hampir semua vendor seluler di dunia menggunakan Android sebagai *operating system* (Nazruddin Safaat, 2012 : 10). Adapaun versi-versi android yang pernah dirilis adalah sebagai berikut :

1. Android versi 1.1
2. Android versi 1.5 (*Cupcake*)
3. Android versi 1.6 (*Donut*)
4. Android versi 2.0/2.1 (*Eclair*)
5. Android versi 2.2 (*Froyo*)
6. Android versi 2.3 (*Gingerbread*)
7. Android versi 3.0/3.1 (*Honeycomb*)
8. Android versi 4.0 (*Ice Cream Sandwich*)
9. Android versi 4.1 (*Jelly Bean*)
10. Android versi 4.4 (*Kitkat*)
11. Android versi 5.0 (*Lolipop*)
12. Android versi 6.0 (*Marshmallow*)

2.4.2 Kelebihan Android

Beberapa hal yang menjadi kelebihan android seperti berikut : Terbuka (*open source*), *platform* terbuka yang berdasarkan pada Linux yang merupakan *platform* terbuka.

- a) Probabilitas tinggi, karena semua program ditulis dengan bahasa pemrograman java.
- b) Menyediakan kualitas suara dan grafik yang tinggi, karena di dalam mesin android sudah ter-*built in* standart suara dan video seperti MP3, ACC. Android juga menyediakan berbagai macam *library/service* yang dapat digunakan untuk fungsi *browser*, peta, GPS dan konektivitas *SQL Database*. (Priyanta F, 2011 : 2).

2.5 Unity

Menurut Rimahirdani (2012), unity adalah *software* penyusun yang terintegrasi untuk membuat *Game* 3D atau konten interaktif lain seperti visualisasi arsitektur atau konten 3D interaktif lainnya. Unity Berjalan di Microsoft Windows dan Mac OS dan dapat mengembangkan *game* yang berjalan di *Windows, Mac, Xbox 360, PlayStation3, Web, Wii, iOS, AnDrone* dan baru-baru ini *Flash*. Adapun fitur-fitur yang dimiliki oleh Unity 3D antara lain sebagaiberikut:

1. Integrated Development Environment (IDE) atau lingkungan pengembangan terpadu.
2. Penyebaran hasil aplikasi pada banyak platform.
3. Engine grafis menggunakan Direct 3D (Windows), OpenGL (Mac, Windows), OpenGL ES (iOS), dan proprietary API (Wii).
4. Game Scripting melalui Mono. Scripting yang dibangun pada Mono, implementasi open source dari NET Framework. Selain itu pemrograman dapat menggunakan UnityScript (bahasa kustom dengan sintaks JavaScript inspired), bahasa C# atau Boo (yang memiliki sintaks Python- inspired).

2.6 Vuforia

Menurut Aggarwal (2014), 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*). Ini menggunakan teknologi *Computer Vision* untuk mengenali dan melacak gambar planar (Target Image) dan objek 3D sederhana, seperti kotak, secara real-time. Kemampuan registrasi citra memungkinkan pengembang untuk mengatur posisi dan virtual orientasi objek, seperti model 3D dan media lainnya, dalam kaitannya dengan gambar dunia nyata ketika hal ini dilihat melalui kamera perangkat *mobile*. Obyek maya kemudian melacak posisi dan orientasi dari gambar secara *real-time* sehingga perspektif pengguna pada objek sesuai dengan perspektif mereka pada target gambar, sehingga muncul bahwa objek virtual adalah bagian dari adegan dunia nyata.

Vuforia SDK menggunakan beberapa tipe target, 2D dan 3D, termasuk pengaturan untuk target yang jumlahnya lebih dari satu (multi target), target berbentuk silinder untuk mendeteksi gambar yang berada pada permukaan silinder, target *marker less*, *marker frame*, dan pengenalan target berbasis *cloud* yang dapat melacak lebih dari satu juta target secara simultan. Beberapa fitur yang dimiliki vuforia SDK diantaranya:

1. Mendeteksi dengan cepat target lokal dengan kapasitas melacak lima target secara simultan.
2. Pendeteksian dalam keadaan kurang cahaya dan bahkan ketika target tertutup sebagian.
3. Kapasitas pelacakan yang tinggi, yang membuat aplikasi terus melacak target dan membantu dalam menjaga konsistensi referensi ditambah dari suatu objek bahkan ketika target tidak lagi terlihat pada kamera. (Amin dan Govilkar, 2015).

2.6.1 Arsitektur Vuforia

Vuforia SDK memerlukan beberapa komponen penting agar dapat bekerja dengan baik. Komponen - komponen tersebut antara lain:

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. *ImageConverter*

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 BackgroundRenderer*

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

e. *ApplicationCode*

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 aplikasi setiap *input* baru dimasukkan.
3. *Render* grafis yang ditambahkan (*augmented*).

f. *TargetResources*

Dibuat menggunakan *on-line 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 *database trackable*. (Aggarwal, 2014)

2.6.2 Marker

Menurut Rizki (2011), *Marker* adalah suatu pola yang dibuat dalam bentuk gambar yang akan dikenali oleh kamera. *Marker* merupakan kunci dari *Augmented Reality* yang akan digunakan untuk mengenali objek. Selain itu informasi *marker* juga akan digunakan untuk menampilkan objek. Ada beberapa jenis (metode) penggunaan *marker augmented reality*, yaitu *marker base tracking* dan *markerless tracking*. Terdapat perbedaan antara pelacakan berbasis *marker* (*marker based tracking*) dan pelacakan *markerless* (*markerless tracking*). Pada pelacakan berbasis *marker* posisi kamera dan orientasi kamera dihitung dengan *marker* yang telah ditetapkan. Sementara pelacakan *markerless*, menghitung posisi antara kamera/pengguna dan dunia nyata tanpa referensi apapun, hanya menggunakan titik-titik fitur alami (edge, corner, garis atau model 3D). Citra atau pola gambar yang baik dalam pembuatan *marker* mempunyai sifat sebagai berikut:

- a) Kaya detail, misal : pemandangan jalan, sekelompok orang, dan lain sebagainya.
- b) Memiliki kontras yang baik, yaitu memiliki daerah terang dan gelap.
- c) Tidak ada pola berulang, misal : banyak kotak yang berukuran sama dalam satu gambar atau pola marker.
- d) Gambar harus 8 atau 24 bit dengan format PNG dan JPG. Warna gambar harus RGB atau GrayScale dan bukan CMYK.

2.7 Blender

Blender adalah salah satu software *open source* yang digunakan untuk membuat konten multi objek khususnya 3Dimensi. Ada beberapa kelebihan yang dimiliki blender dibandingkan *software* sejenis. Berikut kelebihannya:

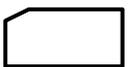
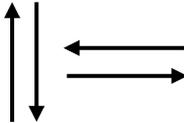
1. *Open Source*, Blender merupakan salah satu software *open source*, dimana kita bisa bebas memodifikasi *source code* untuk keperluan pribadi maupun komersial, asal tidak melanggar *General Public License* (GNU) yang digunakan Blender.
2. *Multi Platform*, Karena sifatnya yang *open source*, Blender tersedia untuk berbagai macam sistem operasi seperti Linux, Mac dan Windows.

3. *Update*, Dengan status yang *Open Source*, Blender bisa dikembangkan oleh siapapun. Sehingga *update software* ini jauh lebih cepat dibandingkan software sejenis lainnya.
4. *Free*, Blender merupakan sebuah *software* yang Gratis. Blender gratis bukan karena tidak laku, melainkan karena luar biasanya fitur yang mungkin tak dapat dibeli dengan uang, selain itu dengan digratiskannya *software* ini, siapapun bisa berpartisipasi dalam mengembangkannya untuk menjadi lebih baik.
5. *Lengkap*, Blender memiliki fitur yang lebih lengkap dari *software* 3D lainnya. Blender tersedia fitur *Video editing, Game Engine, Node Compositing, Sculpting*. Dan bukan lagi *plugin*, tapi sudah *include* atau di *bundling*.
6. *Ringan*, Blender relatif ringan jika dibandingkan software sejenis. Hal ini terbukti dengan sistem minimal untuk menjalankan Blender. Hanya dengan RAM 512 dan prosesor Pentium 4 dan VGA *on board*, Blender sudah dapat berjalan dengan baik.
7. *Komunitas Terbuka*, Tidak perlu membayar untuk bergabung dengan komunitas Blender yang sudah tersebar di dunia. Dari yang baru sampai yang sudah ahli terbuka untuk menerima masukan dari siapapun, selain itu mereka juga saling berbagi tutorial dan *file* secara terbuka. Salah satu contoh nyatanya adalah *OPEN MOVIE* garapan *Blender Institute*. (Adam, 2014).

2.8 Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut (Jogiyanto: 2005:796).

Tabel 2.1 Simbol-simbol *Flowchart* (Sumber:Jogiyanto H.M, 2005:795)

Simbol	Nama Simbol	Fungsi
	Simbol Dokumen	Menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer.
	Simbol Kegiatan Manual	Menunjukkan pekerjaan manual.
	Simbol <i>Decision</i>	Menunjukkan kondisi dalam pengambilan keputusan
	Simbol kartu plong	Menunjukkan <i>input/output</i> yang menggunakan kartu plong (<i>punched card</i>)
	Simbol proses	Menunjukkan kegiatan proses dari operasi program komputer.
	Simbol operasi luar	Menunjukkan operasi yang dilakukan di luar proses operasi komputer.
	Simbol garis alir	Menunjukkan arus dari proses
	Simbol penghubung	Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain.

2.9 Metode Pengembangan Multimedia

Menurut Sutopo (2003), Metode pengembangan multimedia dilakukan berdasarkan enam tahap, yaitu *concept* (pengonsepan), *design* (pendesignan), *material collecting* (pengumpulan materi), *assembly* (pembuatan), *testing* (pengujian), dan *distribution* (pendistribusian). Keenam tahap ini tidak harus berurutan dalam praktiknya, tahap-tahap

tersebut dapat saling bertukar posisi. Meskipun begitu, tahap *concept* memang harus menjadi hal yang pertama kali dikerjakan.

a. *Concept*(konsep)

Tahap *concept* (konsep) yaitu menentukan tujuan dan siapa pengguna program (*identifikasi audience*), macam aplikasi (presentasi, interaktif, dan lain-lain), tujuan aplikasi (informasi, hiburan, pelatihan, dan lain-lain), dan spesifikasi umum.

b. *Design*(perancangan)

Design (perancangan) adalah membuat spesifikasi secara rinci mengenai arsitektur program, gaya, tampilan dan kebutuhan material / bahan untuk program. Spesifikasi dibuat cukup rinci sehingga pada tahap berikutnya, yaitu *material collecting* dan *assembly* tidak diperlukan keputusan baru, tetapi menggunakan apa yang sudah ditentukan pada tahap *design*. Namun demikian, sering terjadi penambahan bahan atau bagian aplikasi ditambah, dihilangkan, atau diubah pada awal pengerjaan proyek.

c. *Material Collecting* (pengumpulan bahan)

Material Collecting (pengumpulan bahan) adalah tahap pengumpulan bahan yang sesuai dengan kebutuhan yang dikerjakan. Bahan-bahan tersebut, antara lain gambar *clip art*, foto, animasi, dan lain-lain yang dapat diperoleh secara gratis atau dengan pemesanan kepada pihak lain sesuai dengan rancangannya. Tahap ini dapat dikerjakan secara parallel dengan tahap *assembly*.

d. *Assembly*(pembuatan)

Tahap *assembly* (pembuatan) adalah tahap pembuatan semua objek atau bahan aplikasi supaya siap untuk digunakan. Pembuatan aplikasi berdasarkan *storyboard*, dan bagan alir (*flowchart*) berasal pada tahap *design*.

e. *Testing*(pengujian)

Tahap *testing* (pengujian) dilakukan setelah menyelesaikan tahap pembuatan (*assembly*) dengan menjalankan aplikasi / program dan dilihat apakah ada kesalahan atau tidak.

f. *Distribution*

Tahap ini aplikasi akan disimpan dalam suatu media penyimpanan. Tahap ini juga dapat disebut tahap evaluasi untuk pengembangan produk yang sudah jadi supaya menjadi lebih baik. Hasil evaluasi ini dapat digunakan sebagai masukan untuk tahap *concept* pada produk selanjutnya.