

## BAB III

### LANDASAN TEORI

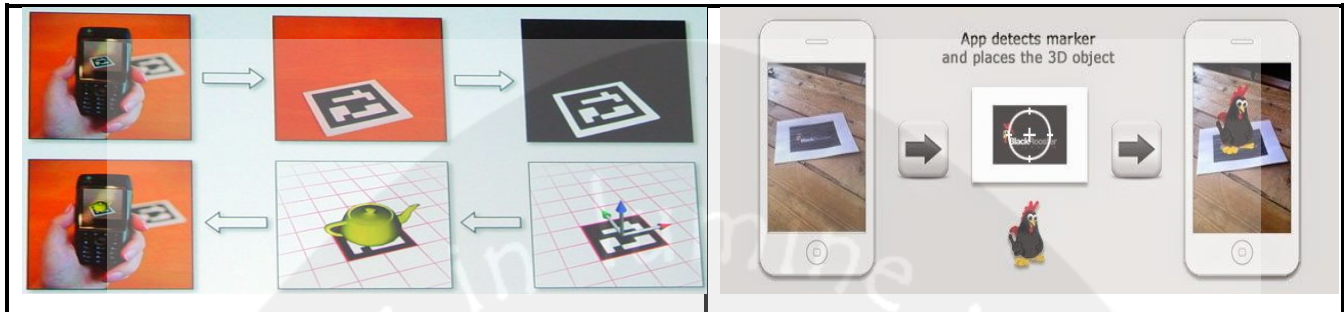
Bab ini akan membahas uraian dasar teori yang akan digunakan penulis dalam melakukan perancangan dan pembuatan program yang dapat dipergunakan sebagai pembanding atau acuan di dalam pembahasan masalah.

#### **3.1 Augmented Reality**

*Augmented Reality* adalah menggabungkan dunia nyata dan virtual, bersifat interaktif secara *real time*, dan merupakan animasi 3D (Azuma, 1997). *Augmented Reality* adalah kombinasi dari lingkup nyata dan virtual, yang isinya lebih nyata dibandingkan dengan virtual. Hal ini akan menjadikan lingkungan di mana seseorang berfikir mengenai penambahan elemen virtual ke dalam lingkungan yang nyata (Daniel & Dieter, 2009). Artinya dapat menambahkan objek virtual atau orang ke pemandangan yang nyata, dengan menggunakan *Augmented Reality* lingkungan virtual atau pengguna seolah-olah ditambahkan kedalam dunia nyata. Di dalam teknologi *Augmented Reality* tidak terpisah dari *Augmented Reality Display*. *Augmented Reality Display* adalah image pembentukan sistem yang menggunakan seperangkat komponen optik, elektronik, dan mekanik untuk menghasilkan gambar suatu tempat pada jalur optik di antara mata pengamat dan benda fisik untuk dapat ditambah (Oliver & Ramesh, 2005). Tergantung pada optik yang digunakan, gambar dapat dibentuk melalui perangkat atau pada permukaan tidak datar yang lebih kompleks, serta bagaimana posisi pengamat atau pengguna berada.

Objek virtual menampilkan informasi dalam bentuk label atau obyek virtual yang hanya dapat dilihat dengan kamera ponsel atau komputer. Sistem *Augmented Reality* bekerja dengan menganalisis objek secara *real-time* yang ditangkap oleh kamera. Pesatnya perkembangan teknologi *smartphone*, sehingga *Augmented Reality* dapat di implementasikan pada perangkat yang

memiliki GPS, kamera, *accelerometer* dan kompas. Kombinasi dari tiga sensor dapat digunakan untuk memberikan tambahan informasi dari objek yang ditangkap oleh kamera.



**Gambar 3.1 Cara Kerja *Augmented Reality***

### **3.2 Rumah Adat**

Menurut KBBI rumah adat adalah rumah tempat diselenggarakannya upacara adat istiadat. Budiharjo (1992) rumah adalah aktualisasi diri yang diejawantahkan dalam bentuk kreativitas dan pemberian makna bagi kehidupan penghuninya. Selain itu rumah adalah cerminan diri, yang disebut Pedro Arrupe sebagai "*Status Confering Function*", kesuksesan seseorang tercermin dari rumah dan lingkungan tempat huniannya.

Pada dasarnya rumah adat adalah bangunan yang memiliki ciri khas khusus, digunakan untuk tempat hunian oleh suatu suku bangsa tertentu. Rumah adat merupakan salah satu representasi kebudayaan yang paling tinggi dalam sebuah komunitas suku/masyarakat. Keberadaan rumah adat di setiap daerah di Indonesia sangat beragam dan mempunyai arti penting dalam perspektif sejarah, warisan, dan kemajuan masyarakat dalam sebuah peradaban.

Rumah-rumah adat di Indonesia memiliki bentuk dan arsitektur masing-masing daerah sesuai dengan budaya adat lokal. Rumah adat pada umumnya dihiasi ukiran-ukiran indah. Pada jaman dulu, rumah adat yang tampak paling indah biasa

dimiliki para keluarga kerajaan atau ketua adat setempat menggunakan kayu-kayu pilihan dan pengerjaannya dilakukan secara tradisional melibatkan tenaga ahli dibidangnya, Banyak rumah-rumah adat yang saat ini masih berdiri kokoh dan sengaja dipertahankan dan dilestarikan sebagai simbol budaya Indonesia.



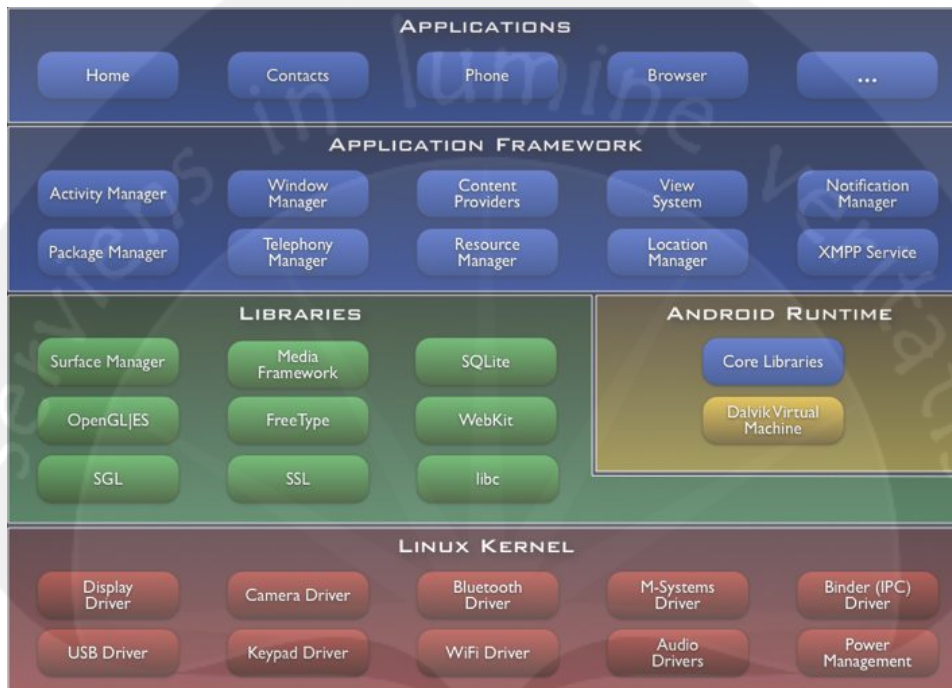
**Gambar 3.2 Rumah Adat**

### **3.3 Android**

Menurut situs resmi *android* (<http://www.android.com>) dan Lessard & Kessler (2010) serta Bharati et.al (2010) *android* adalah sistem operasi untuk telepon genggam yang berbasis *Linux*. *Android* menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *android Inc.*, pendatang baru yang membuat peranti lunak untuk telepon genggam. 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*. Pada saat perilisan perdana *android*, 5 November 2007, *android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat telepon genggam. Di lain pihak, *Google* merilis kode-kode *android* di bawah lisensi perangkat lunak dan standar terbuka perangkat seluler. Terdapat beberapa versi pada sistem operasi *android*, mulai dari versi 1.5 (*CupCake*), versi 1.6 (*Donut*), versi 2.1 (*Eclair*), versi 2.2 (*Froyo*), versi 2.3 (*GingerBread*), versi 3.0 (*HoneyComb*), versi 4,0 (*Ice Cream Sandwich*), versi 4.3

(JellyBean), hingga versi yang terbaru yaitu versi 4.4 (KitKat).

Android sendiri memiliki *Software Development Kit* (SDK) bernama android SDK yang menyediakan seperangkat alat dan *Application Programming Interface* (API) yang dibutuhkan untuk memulai mengembangkan aplikasi-aplikasi pada *platform* android menggunakan bahasa pemrograman *Java*



**Gambar 3.3 Arsitektur Android**

### **3.4 Vuforia Qualcomm**

Vuforia adalah *Augmented Reality Software Development Kit* (SDK) untuk perangkat telepon genggam 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*) 2D dan objek 3D sederhana (<http://www.qualcomm.eu/products/augmented-reality>) seperti kotak, secara *real-time*. SDK Vuforia mendukung berbagai jenis target 2D dan 3D termasuk Target Gambar 'markerless', 3D Multi target konfigurasi, dan bentuk *Marker Frame*. Fitur tambahan dari SDK termasuk Deteksi Oklusi Lokal menggunakan 'Virtual

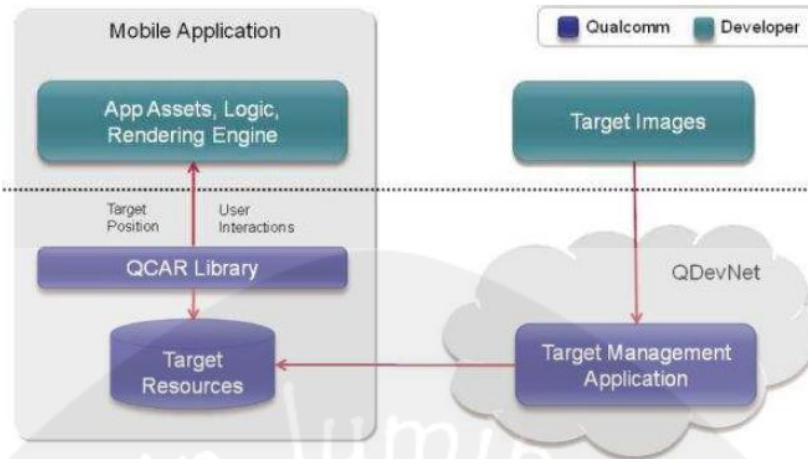
*Button'*, *runtime* pemilihan gambar target, dan kemampuan untuk membuat dan mengkonfigurasi ulang set pemrograman pada saat *runtime*. *Vuforia* menyediakan *Application Programming Interfaces (API)* di *C++*, *Java*, *Objective-C*. *Vuforia SDK* mendukung pembangunan untuk *IOS* dan *android* menggunakan *vuforia* karena itu kompatibel dengan berbagai perangkat telepon genggam termasuk *iPhone (4/4S)*, *iPad*, dan telepon genggam *android* dan *tablet* yang menjalankan *android* sistem operasi versi 2.2 atau yang lebih besar dan prosesor *ARMv6* atau 7 dengan *FPU (Floating Point Unit )* kemampuan pengolahan. Dalam pengembangan aplikasi menggunakan *Vuforia Qualcomm* ini terdiri dari 2 komponen diantaranya adalah:

#### 3.4.1 *Target Manager System*

Mengizinkan pengembang melakukan *upload* gambar yang sudah diregistrasi oleh *marker* dan kemudian melakukan *download target* gambar yang akan dimunculkan.

#### 3.4.2 *QCAR SDK Vuforia*

Mengizinkan pengembang untuk melakukan koneksi antara aplikasi yang sudah dibuat dengan *library static* i.e *libQCAR.a* pada *Ios* atau *libQCAR.so* pada *android*. pembangunan aplikasi dengan *qualcomm Augmented Reality platform* yang terdiri dari *SDK QCAR* dan *Target System Management* yang dikembangkan pada *portal QdevNet*. User meng-upload gambar masukan untuk target yang ingin dilacak dan kemudian men-download sumber daya target, yang dibundel dengan app. *SDK QCAR* menyediakan sebuah objek yang terbagi *libQCAR.so* yang harus dikaitkan dengan aplikasi.



**Gambar 3.4** Arsitektur *Library QCAR SDK*

### **3.5 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 pengembang 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 telepon genggam yang digunakan.

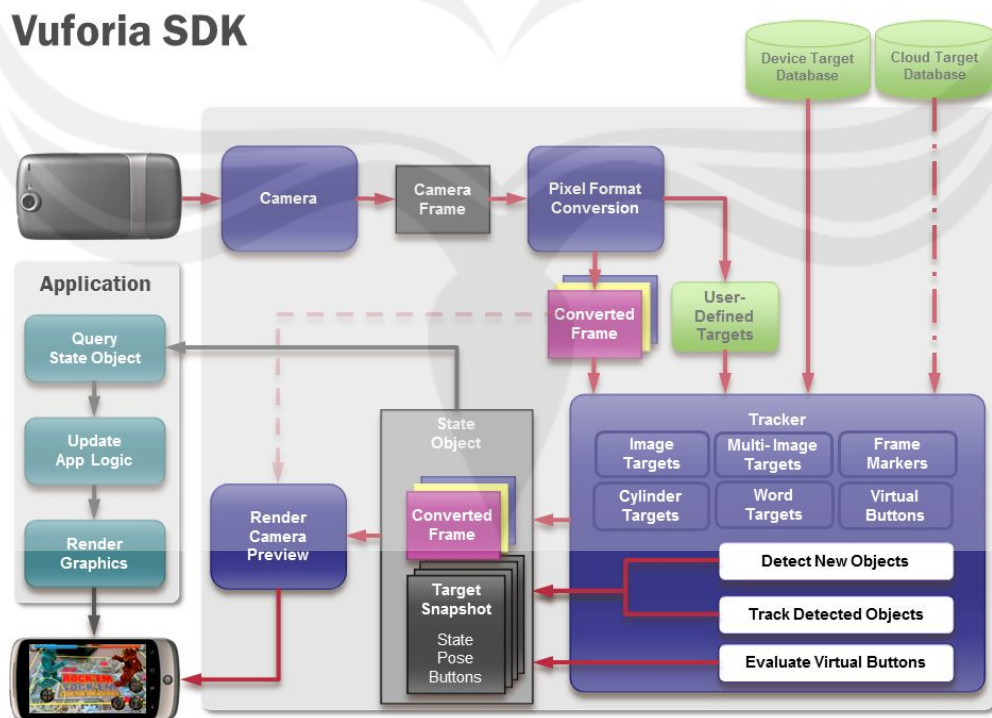
e. *Application Code*

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

f. *Target Resource*

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 *database trackable*.



Gambar 3.5 Diagram Aliran Data Vuforia

### 3.6 Bahasa Pemrograman C#

Bahasa pemrograman C# adalah sebuah bahasa pemrograman umum yang berorientasi objek yang dikembangkan oleh *Microsoft* sebagai bagian dari inisiatif kerangka *.NET Framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti *Java*, *Delphi*, *Visual Basic*, dan lain-lain dengan beberapa penyederhanaan. C# kadang-kadang dapat disebutkan sebagai bahasa pemrograman yang paling mencerminkan dasar dari CLP dimana semua program-program *.NET* berjalan, dan bahasa ini sangat bergantung pada kerangka tersebut sebab secara spesifik didesain untuk mengambil manfaat dari fitur-fitur yang tersedia pada CLR.

### 3.7 Sketchup 8

Pada tanggal 27 April 2006, Google mengumumkan Google SketchUp, yang bebas *download* versi *SketchUp*. Versi gratis ini beda dengan versi *SketchUp Pro*, tetapi terpadu mencakup alat untuk meng-*upload* konten ke *Google Earth* dan *Google 3D Warehouse*, repositori model dibuat dalam *SketchUp*.

Versi gratis dari *Google Sketchup 3D* hanya dapat mengekspor ke SKP dan *Google Earth*. Kmz format file, versi Pro bisa mengekspor dan memasukkan *.3ds*, *.Dae*, *DWG*, *.DXF*, *.Fbx*, *.Obj*, *.XSI*, dan *.wrl* format file. *Google SketchUp* juga dapat menyimpan "*screenshot*" dari model *.Bmp*, *.Png*, *.Jpg*, *.Tif*, dengan versi Pro juga mendukung *.Pdf*, *.Eps*, *.Epx*, *.DWG*, dan *.DXF*. Informasi lokasi GPS selalu disimpan dalam file KMZ. Bangunan desain sendiri dapat disimpan dalam SKP.

### 3.8 Unity

Menurut situs resmi Unity (<http://unity3d.com/unity>) Unity adalah *tool* untuk membuat 3D *video game* atau konten



interaktif lainnya seperti visualisasi arsitektur atau *real-time* 3D animasi. Editor berjalan pada Windows dan Mac OS X dan dapat menghasilkan permainan untuk Windows, Mac, Wii, iPad, iPhone, serta *android platform*. Bahasa pemrograman yang digunakan bermacam-macam, mulai dari *Javascript*, C# dan *Boo*. Pada *unity*, tidak bisa melakukan desain atau *modelling*, dikarenakan *unity* bukan *tool* untuk mendesain. Jadi untuk mendesain, memerlukan 3D editor lain seperti 3D Max atau Blender, kemudian *export* menjadi *format.fbx* atau langsung *format file blend*. *Unity* lebih *simple* dan *powerfull* daripada *game engine* lainnya dengan berbagai *asset* yang dimiliki. Ada dua lisensi utama: *Unity* dan *Unity Pro*, pada versi *Unity Pro* merupakan versi berbayar dan versi *non pro* merupakan versi gratis. Pada versi *unity pro* terdapat fitur tambahan, seperti membuat tekstur dan penambahan efek. Versi gratis, menampilkan *splash screen* (dalam permainan mandiri) dan *watermark* (dalam game web). Baik *unity* maupun *unity pro* termasuk dalam lingkungan pengembangan, *tutorial*, *sample project* dan konten, dukungan dengan *forum*, *wiki*, dan *update* dalam versi yang sama.

### **3.9 Marker**

*Marker* digunakan sebagai pola yang dibuat dalam bentuk gambar yang akan dikenali oleh kamera (Hirzer, 2008). Pola *marker* dapat dibuat dengan menggunakan *photoshop* ataupun *software* lainnya. Untuk *marker* standar yang sering digunakan, pola yang dikenali adalah pola *marker* berbentuk segi panjang dengan kotak-kotak hitam didalamnya. Ada 2 buah metode *Augmented Reality*, yaitu003A

#### a. *Augmented Reality* dengan *marker*

Merupakan metode *Augmented Reality* yang membutuhkan sebuah *marker* (kertas dengan lambang/symbol khusus) yang akan diproses dan dikenali oleh sistem sebagai alas/permukaan yang akan memproyeksikan objek *virtual* hasil *Augmented Reality* di atas permukaan *marker* tersebut.

b. *Markerless Augmented Reality*

Dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan elemen-elemen digital. Seperti yang saat ini dikembangkan oleh perusahaan *Augmented Reality* terbesar di dunia *Total Immersion*, yang telah membuat berbagai macam teknik *Markerless Tracking* sebagai teknologi andalan mereka.

