# BAB II

# TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini akan menguraikan teori-teori yang menunjang penelitian secara ilmiah. Tinjauan pustaka berkaitan dengan penerapan *augmented reality* pada media promosi (brosur). Teori-teori yang terkait ditujukan untuk menunjang pemahaman terhadap penelitian ini dan beberapa hasil penelitian sebelumnya yang telah dilakukan oleh peneliti-peneliti sebelumnya sehingga teori-teori yang dijelaskan dapat membangun dasar dari kerangka pemikiran yang dituliskan ke dalam model penelitian.

## Tinjauan Pustaka

Penelitian Gorbala & Hariadi, (2012) yang berjudul Aplikasi *Augmented Reality* Untuk Katalog Penjualan Rumah, menjelaskan bagaimana memasukkan teknologi *augmented reality* kedalam katalog penjualan rumah yang memunculkan objek 3D dari dalam katalog penjualan, yang juga menguji aplikasi AR untuk menampilkan model 3D terbaik yang sesuai dengan gambar aslinya. Dengan teknik *fiduciary marker* menggunakan *library Artoolkit* penelitian tersebut menghasilkan kesimpulan bahwa untuk mendapatkan hasil yang maksimal dalam aplikasi katalog rumah AR menggunakan *3DS Max* merupakan pilihan yang tepat karena bekerja sangat baik dibandingkan dengan *Google Sketchup* dan *Blender*.

Pada penelitian Gusairi, (2013) yang berjudul Penerapan *Augmented Reality* Untuk Produk Gerabah Keramik Sebagai Media Promosi Dengan Study Kasus Pada Kedai Keramik Sedyo Mulyo Bayat Klaten, menjelaskan bagaimana menerapkan teknologi *augmented reality* untuk membantu kampanye media, supaya dapat meningkatkan minat konsumen untuk memiliki keramik tembikar untuk menghias rumah mereka. Menggunakan AR-Media dengan teknik *fiduciary marker* penelitian tersebut menghasilkan aplikasi *augmented reality* yang dapat merealisasikan pemodelan souvenir gerabah dalam bentuk 3D yang menggantikan souvenir gerabah sebenarnya dan melengkapi media-media promosi kedai keramik, dan dengan adanya sistem yang baru membantu perusahaan dalam menggambarkan objek produk yang ditawarkan kepada calon pengguna produk dengan unik dan jelas.

Penelitian yang dilakukan oleh Yunita, (2013) yang berjudul Penerapan Augmented Reality Dengan Menggunakan Rancangan Miniatur Desain STMIK AMIKOM Yogyakarta Sebagai Media Promosi, membuat aplikasi *augmented reality* yang dapat diakses melalui web *browser* dan menguji jarak pembacaan *marker*. Dengan teknik yang sama *fiduciary marker* namun dengan *library* *Flartoolkit* penelitian ini menghsilkan kesimpulan bahwa media dapat berjalan dengan baik tanpa perlu menginstal aplikasinya, karena aplikasi dapat diakses menggunakan web *browser*, dengan syarat web *browser* tersebut sudah terinstal plug in *flash player*. Kesimpulan lain adalah jarak marker dengan kamera sangat berpengaruh dalam proses berjalannya program, bila terlalu dekat atau terlalu jauh maka kamera tidak dapat membaca *marker* dengan baik sehingga program tidak dapat mengenali *marker* tersebut.

Penelitian oleh Faizin, ( 2017 ) yang berjudul Aplikasi Media Pemasaran Properti dengan Menggunakan Teknologi Augmented Reality pada Perangkat Android, Objek 3D dapat mulai tampil dengan jarak minimal pendeteksian awal yaitu 20 cm. Sedangkan untuk jarak maksimal pendeteksian awal yaitu sekitar 105.9 cm. Pada saat pelacakan, objek 3D masih dapat tampil dari jarak sekitar 6.8 cm sampai dengan jarak sekitar 242.2 cm. Objek 3D dapat tampil dengan sudut pendeteksian awal maksimum sekitar 56.2 derajat. Sedangkan saat pelacakan, objek 3D masih dapat tampil sampai sekitar 87.8 derajat.

Penelitian yang dilakukan oleh Nadira, ( 2016 ) yang berjudul Implementasi Augmented Reality Pada Brosur Teknik Informatika Universitas Tanjungpura Menggunakan Metode Marker, dalam penelitian ini digunakan *Game engine Unity* untuk membangun aplikasi *Android* dan *Vuforia SDK* agar memungkinkan aplikasi yang dibangun menjadi aplikasi berteknologi *Augmented Reality*. *Marker* yang digunakan adalah brosur Teknik Informatika Universitas Tanjungpura yang berisi beberapa *marker*. Aplikasi ini dapat bermanfaat bagi calon mahasiswa yang memerlukan informasi tentang Teknik Informatika Universitas Tanjungpura. Dari tiga hasil pengujian, dapat disimpulkan bahwa aplikasi yang dibangun dapat menampilkan semua objek 3D dari gedung dan ruangan dan dapat membantu calon mahasiswa untuk mendapatkan informasi yang belum didapatkan pada brosur.

###### Tabel 2.1 Ringkasan Tinjauan Pustaka

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama peneliti** | **Tahun** | **Judul** | **Library** | **Kesimpulan** |
| Bregga Tedy Gorbala dan Mochamad Hari | 2010 | Aplikasi *Augmented Reality* Untuk Katalog Penjulan Rumah | Artoolkit | Dalam pembuatan aplikasi AR menggunakan 3DS Max lebih baik dibandingkan menggunakan Google Sketchup |
| Ahmad Gusairi | 2013 | Penerapan *Augmented reality* untuk Produk Gerabah Keramik sebagai Media Promosi dengan Studi Kasus pada Kedai Keramik Sedyo Mulyo Bayat Klaten | AR-Media | Aplikasi *augmented reality* dapat merealisasikan permodelan souvenir gerabah dalam bentuk 3D |
| Beti Yunita | 2013 | Penerapan *Augmented reality* dengan menggunakan Rancangan Miniatur Desain STMIK AMIKOM Yogyakarta sebagai Media Promosi | Flartoolkit | Jarak *marker* yang digunakan pada aplikasi AR dengan kamera sangat berpengaruh dalam proses berjalannya program |
| Ahmad Arif Faizin | 2017 | Aplikasi Media Pemasaran Properti Dengan Menggunakan Teknologi *Augmented Reality* Pada Perangkat Android | Artoolkit | Objek 3D dapat mulai tampil dengan jarak minimal pendeteksian awal yaitu 20 cm dan jarak maksimal pendeteksian awal yaitu sekitar 105.9 cm. |
| Zahra Nadira | 2016 | Implementasi *Augmented Reality* Pada Brosur Teknik Informatika Universitas Tanjungpura Menggunakan Metode Marker | Artoolkit | Aplikasi yang dibangun dapat menampilkan semua objek 3D dari gedung dan ruangan. |

## Landasan Teori

### *Augmented Reality*

Menurut Ronald (1997) *augmented reality* adalah penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat intergrasi antar benda dalam tiga dimensi, yaitu benda nyata benda maya terintergrasi 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 penjejakan yang efektif (Azuma, 2011).



###### Gambar 2.1 Contoh *Augmented Reality*

Paul Milgram dan Fumio Kishino pada tahun 1994, mendefinisikan *Milgram’s Reality-Virtuality Continuum*. Mereka menggambarkan sebuah *kontinum* yang membentang dari lingkungan nyata untuk lingkungan *virtual* murni. Mereka menyimpulkan bahwa AR lebih dekat dengan dunia nyata dan *augmented virtuality* lebih dekat dengan dunia *virtual* (Fathoni, Cahyono, & Kusuma, 2012).

Dalam penerapannya teknologi *Augmented Reality* memiliki beberapa komponen yang harus ada untuk mendukung kinerja dari proses pengolahan citra digital. Menurut J.C, Oliveira, & Giraldi, 2005 adapun komponen-komponen tersebut adalah sebagai berikut :

1. *Scene Generator*

*Scene Generator* adalah komponen yang bertugas untuk melakukan rendering citra yang ditangkap oleh kamera. Objek virtual akan di tangkap kemudian diolah sehingga dapat kemudian objek tersebut dapat ditampilkan.

1. *Tracking System*

*Tracking System* merupakan komponen yang terpenting dalam *Augmented Reality*. Dalam proses *tracking* dilakukan sebuah pendeteksian pola objek *virtual* dengan objek nyata sehingga sinkron diantara keduanya dalam artian proyeksi *virtual* dengan proyeksi nyata harus sama atau mendekati sama sehingga mempengaruhi validitas hasil yang akan didapatkan.

1. *Display*

Dalam pembangunan sebuah sistem yang berbasis AR dimana sistem tersebut menggabungkan antara dunia virtual dan dunia nyata ada beberapa parameter mendasar yang perlu diperhatikan yaitu optik dan teknologi video. Keduanya mempunyai keterkaitan yang tergantung pada faktor resolusi, fleksibilitas, titik pandang, tracking area. Ada batasan-batasan daalam pengembangan teknologi *Augmented Reality* dalam hal proses menampilkan objek. Di antaranya adalah harus ada batasan pencahayaan, resolusi layar, dan perbedaan pencahayaan citra antara citra virtual dan nyata.

1. *AR Devices*

Ada beberapa tipe media yang dapat digunakan untuk menampilkan objek berbasis *Augmented Reality* yaitu dengan menggunakan optik, sistem retina *virtual*, video penampilan, monitor berbasis AR dan proyektor berbasis AR.

#### Penerapan Augmented Reality

Anggi Andriyadi, menyatakan dalam bukunya yang berjudul “*Augmented Reality With Artoolkit”* bahwa terdapat beberapa bidang yang pernah menerapkan teknologi *augmented reality* adalah :

1. Kedokteran (*Medical*)

Contoh penggunaannya adalah pada pemeriksaan sebelum operasi, seperti CT Scan atau MRI, yang memberikan gambaran kepada ahli bedah mengenai anatomi internal pasien. Dari gambar-gambar ini kemudian pembedahan direncanakan. *Augmented reality* dapat diaplikasikan sehingga tim bedah dapat melihat data CT Scan atau MRI pada pasien saat pembedahan berlangsung. Penggunaan lain adalah untuk pencitraan ultrasonik, di mana teknisi ultrasonik dapat mengamati pencitraan fetus yang terletak di abdomen wanita yang hamil.



###### Gambar 2.2 Contoh Penggunaan AR Pada Bidang Kedokteran

1. Hiburan (*Entertaiment*)

Dunia hiburan membutuhkan *augmented reality* sebagai penunjang efek-efek yang akan dihasilkan oleh hiburan tersebut. Sebagai contoh, ketika sseorang wartawan cuaca memperkirakan ramalan cuaca, dia berdiri di depan layar hijau atau biru, kemudian dengan teknologi *augmented reality*, layar hijau atau biru tersebut berubah menjadi gambar animasi tentang cuaca tersebut, masuk ke dalam animasi tersebut.



###### Gambar 2.3 Contoh Penggunaan AR Dalam Dunia Hiburan

1. Latihan Militer (*Military Training*)

Kalangan militer telah bertahun-tahun menggunakan tampilan dalam kokpit yang menampilkan informasi kepada pilot pada kaca pelindung kokpit atau kaca depan helm penerbangan mereka, ini merupakan sebuah bentuk tampilan realitas tertambah. SIMNET, sebuah sistem permainan simulasi perang, juga menggunakan teknologi realitas tertambah. Dengan melengkapi anggota militer dengan tampilan kaca depan helm, aktivitas unit lain yang berpatisipasi dapat ditampilkan.

Contohnya, seorang tentara yang menggunakan perlengkapan tersebut dapat melihat helikopter yang dating. Dalam peperangan, tampilan medan perang yang nyata dapat digabungkan dengan informasi catatan dan sorotan untuk memperlihatkan unit musuh yang tidak terlihat tanpa perlengkapan ini.



###### Gambar 2.4 Contoh Penggunaan AR Dalam Kalangan Militer

1. *Engineering Design*

Seorang engineering Design membutuhkan *augmented reality* untuk menampilkan hasil design mereka secara nyata terhadap klien. Dengan *augmented reality* klien akan tahu, tentang spesifikasi yang lebih detail tentang desain mereka.



###### Gambar 2.5 Contoh Penggunaan AR Pada Engineering Design

1. *Robotics* dan *Telerobotics*

Dalam bidang robotika, seorang operator robot, menggunakan pengendali pencitraan visual dalam mengendalikan robot itu. Jadi, penerapan *augmented reality* dibutuhkan di dunia robot.



###### Gambar 2.6 *MekaMon* Contoh Penggunaan AR Dalam Bidang Robotika

1. *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.

#### Cara Utama Interaksi Dalam Aplikasi *Augmented Reality*

1. *Tangible Interface*

Berwujud *interface* yang mendukung interaksi secara langsung dengan dunia nyata dengan memanfaatkan objek nyata, salah satu contoh dari *tangible interface* ini adalah pada aplikasi virtual *fitting room* dan *game* *augmented reality pringleys* dimana tabung dari kemasan pringleys menjadi *marker* dan *control* dari *game* tersebut.

1. Kolaborasi *AR Interface*

Kolaborasi AR *Interface* menampilkan beberapa tampilan untuk mendukung suatu kegiatan dalam berbagi *interface* 3D untuk meningkatkan interaksi kolaboratif dengan banyak perangkat dan banyak *user*. *Interface* ini dapat diintergrasikan dengan aplikasi medis untuk melakukan diagnosa operasi.

1. *Interface Hybrid*

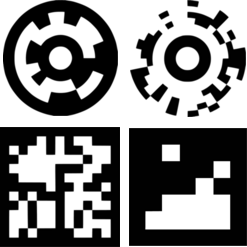
Menggabungkan berbagai macam *device* berbeda, tetapi saling melengkapi *interface* serta memungkinkan untuk berinteraksi melalui berbagai perangkat interaksi, contoh pada aplikasi *Augmented Reality* yang menggunakan sarung tangan dan kacamata.

#### Metode Augmented Reality

Ada beberapa metode yang digunakan pada *Augmented Reality* yaitu *marker* *based tracking* dan *markerless* (Patkar, Singh, & Birje, 2013).

1. *Marker Based Tracking*

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. *Marker Based Tracking* ini sudah lama dikembangkan sejak tahun 80-an dan pada awal 90-an mulai dikembangkan untuk penggunaan *Augmented Reality*.



###### Gambar 2.7 *Marker* Based Tracking

1. *Markless*

*Markerless* merupakan sebuah metoda pelacakan dimana AR menggunakan objek di dunia nyata sebagai *marker* atau tanpa menggunakan *marker* buatan.AR dengan teknik tanpa penanda ini menggunakan teknik pelacakan secara alami (*natural feature*) bukan pengenalan penanda (*fiducial marker*). Teknik ini menggunakan prinsip deteksi tepi, deteksi sudut dan tekstur dari gambar atau objek (Sari, Sulistyo, & Hantono, 2014).

Dengan metode *markerless* pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan objek-objek *virtual*. Seperti yang saat ini dikembangkan oleh perusahaan *Augmented Reality* terbesar di dunia *Total Immersion* dan *Qualcomm*, mereka telah membuat berbagai macam teknik *Markerless* *Tracking* sebagai teknologi andalan mereka, seperti *Face Tracking*, *3D Object Tracking*, *Motion Tracking*, dan *GPS Based Tracking*.

1. *Face Tracking*

Dengan menggunakan teknik algoritma yang mereka kembangkan, komputer juga dapat mengenali wajah manusia secara umum dengan cara mengenali posisi mata, hidung, dan mulut. Kemudian akan mengabaikan objek-objek lain di-sekitarnya.



###### Gambar 2.8 Contoh *Face Tracking*

1. *3D Object Tracking*

Berbeda dengan *face tracking* yang hanya mengenali wajah manusia. Dengan menggunakan teknik 3D *Object Tracking* dapat mengenali semua benda yang berada di sekitar seperti mobil, motor, meja, tv dan lain-lain.



###### Gambar 2.9 Contoh 3D *Object Tracking*

1. *Motion Tracking*

Teknik komputer ini dapat menangkap gerakan atau *Motion Tracking* yang telah dimulai digunakan secara ekstensif untuk memproduksi sebuah film-film yang mensimulasikan pada gerakangerakan tubuh. Contohnya pada film avatar, dimana James Cameron menggunakan teknik ini untuk membuat film tersebut terlihat seperti lebih *realtime*.



###### Gambar 2.10 Contoh *Motion Tracking*

1. *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 di dalam *smartphone*, aplikasi akan mengambil data dari *GPS* dan Kompas kemudian menampilkannya dalam bentuk arah yang kita inginkan secara *real-time*, bahkan ada beberapa aplikasi menampilkannya dalam bentuk 3D.



###### Gambar 2.11 Contoh *GPS Tracking*

### Marker

*Marker* yaitu sebuah gambar dengan pola unik yang dapat diambil dengan kamera serta dapat dikenali oleh aplikasi AR (Geroimenko, 2012). *Marker* dapat berupa foto sebuah objek nyata atau gambar buatan dengan pola unik. *Marker* ini menggunakan teknik pengenalan penanda atau *fiducial marker*. Berikut ini adalah metode *marker* dilihat dari segi objek:

1. *Single Marker*

Satu *marker* untuk satu objek.

1. *Multiple Marker*

Dua *marker* atau lebih untuk beberapa objek.

1. *Tangible Marker*

*Marker* yang bisa berinteraksi dengan *marker* lainnya.

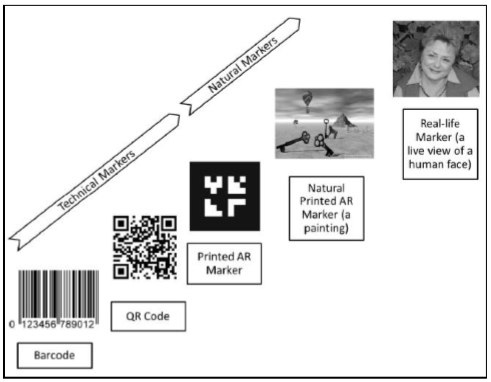
1. *Markerless*

*Marker* dalam bentuk apapun, tidak harus berbentuk kotak.

1. *Virtual Marker*

*Marker* dalam bentuk *virtual button* animasi.

Menurut (Geroimenko, 2012) *marker* telah mengalami beberapa kali evolusi yaitu : *barcode*, *QR Code*, *AR marker* buatan (*printed AR marker*), *AR marker* berupa gambar alami (*natural printed AR marker*) dan *marker* yang sebenarnya *(real life marker,* missal *human face)*. Penanda yang berupa *natural printed AR marker* dan *human face* merupakan kategori *markerless*. Pada *Vuforia marker* disebut dengan *image target*.



###### Gambar 2.12 Evolusi *marker* dalam teknologi pola / gambar yang berkaitan dengan aplikasi AR

*Image target* yang ideal dan dapat dilacak oleh sistem AR meliputi (Sari, Sulistyo, & Hantono, 2014) :

1. Fitur gambarnya kaya (polanya rumit), misalnya gambar pemandangan, gambar sekumpulan orang, kolase dan lain-lain.
2. Kontrasnya bagus, gelap dan terangnya jelas.
3. Tidak ada pengulangan pola, missal lapangan rumput, kotak-kotak
4. Grafik warnyanya 8 atau 24 bit format JPG atau PNG, ukurannya kurang dari 2MB, JPGs haruslah RGB atau GRAYSCALE (bukan CMYK).

### Vuforia Qualcomm

Vuforia Qualcomm merupakan *library* yang digunakan sebagai pendukung adanya *Augmented Reality* pada Android. Vuforia menganalisa gambar dengan menggunakan pendeteksi *marker* dan menghasilkan informasi 3D dari *marker* yang sudah dideteksi via API. Programmer juga dapat menggunakannya untuk membangun objek 3D virtual pada kamera (Michael, 2012). Adapun contoh nyata pembuatan objek 3D dengan menggunakan Vuforia adalah seperti ditunjukkan pada gambar di bawah ini.



###### Gambar 2.13 Objek mobil 3D virtual yang muncul pada kamera

Gambar tersebut adalah pengembangan aplikasi menggunakan platform Qualcomm AR. Platform tersebut terdiri dari 2 komponen (Michael, 2012) di-antaranya adalah:

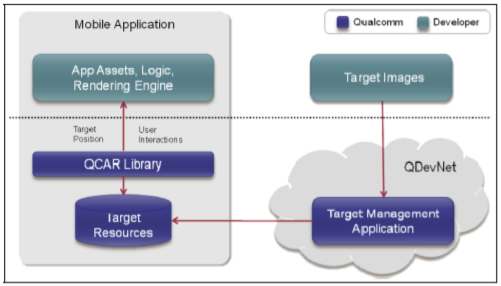
1. *Target Management System*

Qualcomm *target management system* mengijinkan pengembang untuk melakukan *upload* gambar dan menghasilkan kumpulan data pada target tujuan. Aplikasi dapat mencocokkan *image* dalam *frame* berikut dengan kumpulan datanya (Michael, 2012).

1. QCAR SDK Vuforia

Sebuah aplikasi Vuforia SDK berbasis AR menggunakan layar perangkat *mobile* sebagai "lensa ajaib" atau cermin ke dunia *augmented* dimana dunia nyata dan maya tampaknya hidup berdampingan. Aplikasi ini membuat kamera menampilkan gambar langsung pada layar untuk mewakili pandangan dari dunia fisik. Objek *Virtual* 3D kemudian ditampilkan pada kamera dan mereka terlihat menyatu di dunia nyata.

SDK Vuforia mengijinkan pengembang untuk melakukan koneksi antara aplikasi yang sudah dibuat dengan *library static* seperti libQCAR.a pada iOS atau libQCAR.so pada Android.



###### Gambar 2.14 Arsitektur library QCAR SDK

Gambar tersebut memberikan gambaran umum pembangunan aplikasi dengan Qualcomm AR *Platform*. *Platform* ini 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 app (Wirga, Pungkasanthi, Yuniarti, Kusnendar, & Darta, 2012).

1. Arsitektur Sistem

Menurut Professor Michael R. Lyu (2011) komponen inti dari QCAR Lib adalah sebagai berikut :

1. 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.

1. *Image Converter*

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

1. *Tracker*

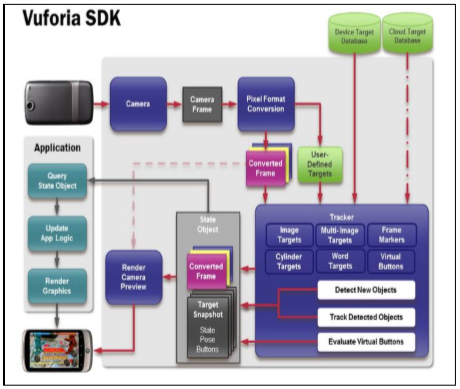
Mengandung algoritma *computer vision* yang dapat mendeteksi dan melacak objek dunia nyata yang ada pada *video* kamera. Berdasarkan gambar dari kamera, algoritma yang berbeda bertugas untuk mendeteksi *trackable* baru, dan mengevaluasi *virtual button*. Hasilnya akan disimpan dalam *state object* yang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

1. *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.

1. *Application Code*

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*.



###### Gambar 2.15 Diagram aliran data dari Vuforia SDK

1. Trackables

*Trackables* adalah kelas dasar yang mewakili semua benda dunia nyata bahwa SDK Vuforia dapat melacak *six degrees-of-freedom*. Setiap *trackable*, ketika dideteksi dan dilacak, memiliki nama, ID, status, dan pose informasi. *Target* Gambar, Gambar *Multi Target* dan *Marker*, semua *trackables* yang mewarisi sifat dari kelas dasar. *Trackables* yang diperbarui setiap frame diproses, dan hasilnya diteruskan ke aplikasi pada *state* objek (Wirga, Pungkasanthi, Yuniarti, Kusnendar, & Darta, 2012).

1. Marker

Dalam pembuatan *marker* dalam hal ini *markerless* diperlukan sebuah file gambar.JPG yang nantinya akan di-*upload* ke vuforia, *marker* yang telah di-*upload* akan dinilai kualitasnya oleh sistem, berikut adalah contohnya:

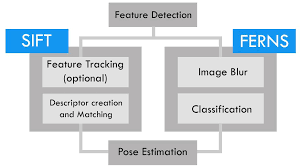


###### Gambar 2.16 Contoh *Marker*

Gambar di atas adalah contoh gambar yang sangat baik dalam proses pendektesian *marker*. Gambar tersebut memiliki *Feature* yang tinggi, detail dan ketajaman gambar tersebar disemua bagian gambar. Objek yang menyusun gambar tersebut menghasilkan tepi yang tajam dan mmemberikan kontras yang tinggi (Wirga, Pungkasanthi, Yuniarti, Kusnendar, & Darta, 2012).

1. Natural Feature Tracking and Detection

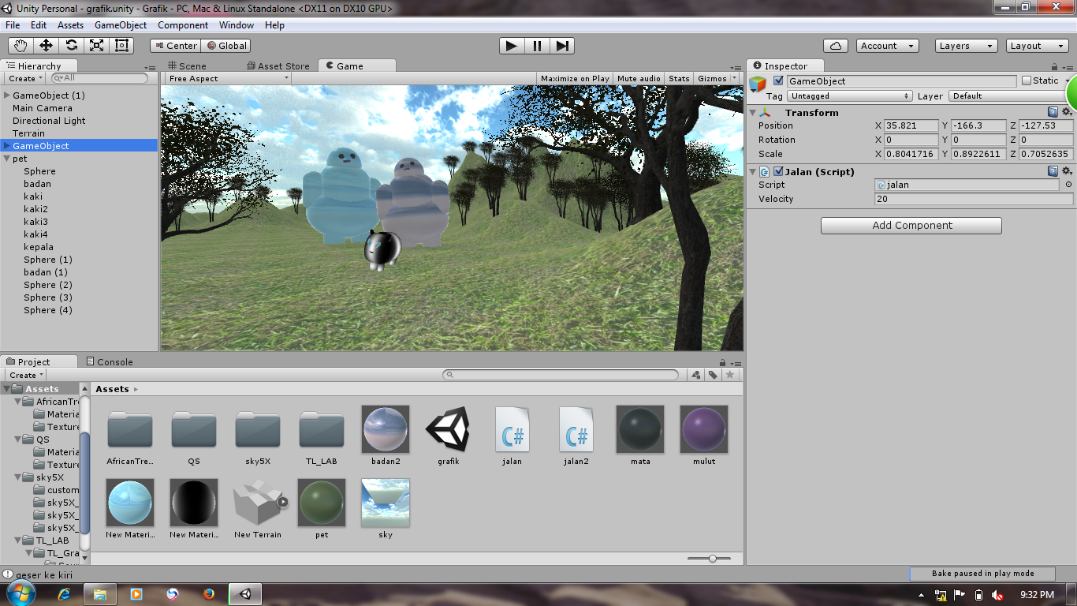
Qualcomm *Augmented reality* (QCAR) merupakan salah satu SDK untuk merancang aplikasi AR pada Vuforia. QCAR menerapkan konsep *natural features tracking* untuk mendeteksi dan mengenali *image target*. QCAR juga menerapkan metode FAST *(Features from Accelerate Segmen Test)*. Metode ini menekankan pada pendeteksian terhadap titik-titik *(interest fiew)* atau sudut pada gambar. Kemudian dilanjutkan dengan proses analisa tepi untuk mendapatkan deteksi sudut yang tepat. Deteksi sudut merupakan tahapan penting dalam pelacakan secara alami, misalnya *metoda Simultaneous Localization and Mapping* (SLAM), *localization*, pencocokan dan pengenalan gambar. Dibutuhkan kekuatan pelacakan *(robust)* untuk melakukan deteksi terhadap titik-titik atau fitur-fitur dari gambar yang ditemukan dalam *real-time frame-rate application*. Deteksi fitur yang cukup bagus yaitu SIFT *(Scale Invariant Features Transform)*. SIFT memberikan hasil deteksi fitur yang berkualitas bahkan dalam aplikasi yang kompleks. Selain teknik SIFT juga terdapat teknik lain yaitu FERNS. FERNS dapat menemukan titik-titik fitur yang lebih banyak dibanding SIFT bahkan dalam gambar yang kabur sekalipun. Akan tetapi FERNS membutuhkan kapasistas memori yang besar (Z & X, 2010).



###### Gambar 2.17 Alur SIFT dan FERN

### Unity 3D

Unity adalah sebuah *Tools* yang terintegrasi untuk membuat bentuk objek 3D pada Video Games, atau untuk konteks interaktif lain seperti Visualisasi Arsitektur atau animasi 3D real-time. Lingkungan dari pengembangan Unity 3D berjalan pada Microsoft Windows dan Mac Os X, serta permainan yang dibuat oleh Unity dapat berjalan pada Windows, Mac, Xbox 360, Playstation 3, Wii, iPad, iPhone, dan tidak ketinggalan pada *platform* Android. Unity juga dapat membuat *game* berbasis *browser* yang menggunakan Unity *web player plugin*, yang dapat bekerja pada Mac dan Windows, tapi tidak pada Linux. Web player yang dihasilkan juga digunakan untuk pengembangan pada widgets Mac.Unity pada dasarnya berisi atas editor untuk membangun/mendesain materi dari suatu permainan dan sebuah game *engine* untuk mengeksekusi produk akhir. *Tools* yang memiliki kemampuan seperti Unity yang telah ada sebelumnya di antaranya adalah *Director*, *Blender game Engine*, *Virtools*, *Torque Game builder*, dan *Gamestudio*, yang mana juga digunakan sebagai lingkungan grafik yang terintegrasi sebagai metode awal pengembangan (Wirga, Pungkasanthi, Yuniarti, Kusnendar, & Darta, 2012).



###### Gambar 2.18 Lembar kerja Unity

1. Fitur-fitur Pada Unity 3D
2. *Rendering*

*Graphics engine* yang digunakan adalah Direct3D (Windows, Xbox 360), OpenGL (Mac, Windows, Linux, PS3), OpenGL ES (Android, iOS), dan proprietary APIs (Wii). Ada pula kemampuan untuk *bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion* (SSAO), *dynamic shadows using shadow maps, render-to-texture and full-screen postprocessing effects*. Unity dapat mengambil format desain dari 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks and Allegorithmic Substance. Asset tersebut dapat ditambahkan ke *game project* dan diatur melalui *graphical user interface* Unity.

*ShaderLab* adalah bahasa yang digunakan untuk *shaders*, dimana mampu memberikan deklaratif “programming” dari *fixed-function pipeline* dan program shader ditulis dalam GLSL atau Cg. Sebuah shader dapat menyertakan banyak varian dan sebuah spesifikasi *fallback declarative*, dimana membuat Unity dapat mendeteksi berbagai macam video card terbaik saat ini, dan jika tidak ada yang kompatibel, maka akan dilempar menggunakan shader alternative yang mungkin dapat menurunkan fitur dan performa. Pada 3 Agustus 2013, seiring dengan diluncurkannya versi 4.2. Unity mengijinkan developer indie mengunakan *Realtime shadows* hanya untuk *Directional lights*, dan juga menambahkan kemampuan dari DirectX11 yang memberikan shadows dengan resolusi pixel yang lebih sempurna, textur untuk membuat objek 3d dari grayscale dengan lebih grafik facial, animasi yang lebih halus dan mempercepat FPS.

1. *Scripting*

*Script game engine* dibuat dengan Mono 2.6, sebuah implementasi *open-source* dari .NET *Framework*. *Programmer* dapat menggunakan UnityScript (bahasa terkustomisasi yang terinspirasi dari sintax ECMAScript, dalam bentuk JavaScript), C#, atau Boo (terinspirasi dari sintax bahasa pemrograman phyton). Dimulai dengan dirilisnya versi 3.0, Unity menyertakan versi MonoDevelop yang terkustomisasi untuk *debug script.*

1. *Asset Tracking*

Unity juga menyertakan Server Unity Asset – sebuah solusi terkontrol untuk *defeloper game asset* dan *script*. *Server* tersebut menggunakan PostgreSQL sebagai *backend*, sistem audio dibuat menggunakan FMOD *library* (dengan kemampuan untuk memutar Ogg Vorbis *compressed audio*), video *playback* menggunakan Theora codec, *engine* daratan dan vegetasi (dimana mensuport *tree billboarding, Occlusion Culling dengan Umbra*), *built-in lightmapping* dan *global illumination* dengan *Beast*, *multiplayer* *networking* menggunakan *RakNet*, dan navigasi mesh pencari jalur built-in.

1. *Platforms*

Unity support pengembangan ke berbagai *platform*. Di dalam *project*, *developer* memiliki kontrol untuk mengirim keperangkat *mobile*, *web* *browser*, *desktop*, dan *console*. Unity juga mengijinkan spesifikasi kompresi *textur* dan pengaturan resolusi di setiap *platform* yang didukung. Saat ini *platform* yang didukung adalah BlackBerry 10, Windows 8, Windows Phone 8, Windows, Mac, Linux, Android, iOS, Unity Web Player, Adobe Flash, PlayStation 3, Xbox 360, Wii U and Wii. Meskipun tidak semua terkonfirmasi secara resmi, Unity juga mendukung PlayStation Vita yang dapat dilihat pada game Escape Plan dan Oddworld: New „n‟ Tasty. Rencana *platform* berikutnya adalah PlayStation 4 dan Xbox One. Dan juga rumor untuk kedepanya mengatakan HTML akan menjadi platformnya, dan plug-in Adobe baru dimana akan disubtitusikan ke Flash Player, juga akan menjadi *platform* berikutnya.

1. *Asset Store*

Diluncurkan November 2010, *Unity Asset Store* adalah sebuah *resource* yang hadir di Unity editor. *Asset store* terdiri dari koleksi lebih dari 4,400 *asset packages*, beserta 3D *models*, *textures* dan *materials*, sistem *particle*, musik dan efek suara, tutorial dan *project*, *scripting* *package*, editor *extensions* dan *service online*.

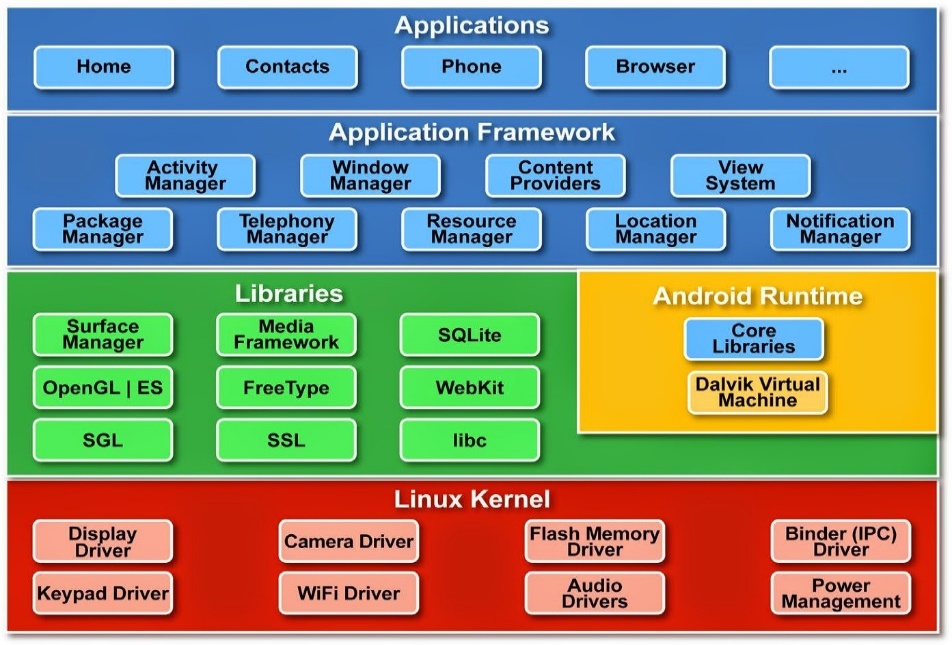
1. *Physics*

Unity juga memiliki support *built-in* untuk PhysX *physics engine* (sejak Unity 3.0) dari Nvidia (sebelumnya Ageia) dengan penambahan kemampuan untuk simulasi *real-time cloth* pada *arbitrary* dan *skinned meshes, thick ray cast,* dan *collision layers.*

### Android

Android adalah sistem operasi untuk telepon seluler 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 ponsel. 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 seluler. Di lain pihak, Google merilis kode–kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).



###### Gambar 2.19 Arsitektur Android

1. Versi Android

Berikut adalah Tabel tingkatan versi sistem operasi android.

|  |  |  |  |
| --- | --- | --- | --- |
| Nama | Versi | Peluncuran | Fitur |
| Cupcake | 1.5 | 27 April 2009 | * Dukungan Pihak ketiga * Perekaman dan pemutaran video * Fitur copy dan paste di browser * Transisi layar animasi dan pilihan auto-rotasi |
| Donut | 1.6 | 15 September 2009 | * Suara dan masukan teks pencarian ditingkatkan * Kamera lebih cepat dan terintegrasi * Multi-bahasa sistem * Support resolusi layar WVGA |
| Eclair | 2.0 – 2.1 | 26 Oktober 2009 | * Pengoptimalan *hardware* * Peningkatan Google Maps * Perubahan UI dengan browser baru dan dukungan HTML5 * Daftar kontak baru * Support *Flash* untuk kamera 3,2 MP, digital zoom * Bluetooth 2.1 |
| Froyo | 2.2 – 2.2.3 | 20 Mei 2010 | * Kecepatan memori dan kinerja ditingkatkan * Fungsi USB tethering dan Wi-Fi hotspot * Browser ditingkatkan * Mendukung kata sandi angka dan karakter unik * Keamanan ditingkatkan |
| Gingerbread | 2.3 – 2.3.7 | 6 Desember 2010 | * *Interface* lebih simpel dan peningkatan kecepatan * Mendukung layar besar * Virtual keyboard lebih cepat dan intuitif * Efek suara, equalizer, dan virtual * Mendukung |
| Honeycomb | 3.0 – 3.2.6 | 22 Februari 2011 | * Keyboard lebih cepat dan akurat * Akses cepat pada fitur-fitur kamera * Mampu menampilkan album pada layar penuh * Akselerasi kinerja perangkat * Mendukung keyboard eksternal, joystick |
| Ice Cream Sandwich | 4.0 – 4.0.4 | 18 Oktober 2011 | * Akses langsung aplikasi dari pengunci layar * Integrasi suara, pembuka kunci deteksi wajah * Sinkronisasi otomatis pada browser * Foto editor dan warna layar lebih halus * Kinerja stabil dan aplikasi kamera lebih baik |
| Jelly Bean | 4.1 – 4.3.1 | 9 Juli 2012 | * Peningkatan kinerja sistem * Peningkatan fitur keamanan * Mendukung Bluetooth dengan energi sedikit * Mendukung emoji |
| Kitkat | 4.4 – 4.4.4 | 31 Oktober 2013 | * Fitur perekam layar built-in * Peningkatan kemampuan aplikasi * Peningkatan fitur auto-focus kamera * Peningkatan perlindungan dan keamanan * Pemutaran musik offline |
| Lollipop | 5.0 – 5.1.1 | 12 November 2014 | * Peningkatan kunci layar * Oenambahan multi-bahasa * Peningkatan tampilan layar *interface* * Peningkatan notifikasi sistem |
| Marshmallow | 6.0 – 6.0.1 | 5 Oktober 2015 | * Fitur aplikasi stand-by * Kunci layar *fingerprint* * Arsip aplikasi lebih besar * Back-up full data otomatis |
| Nougat | 7.0 | Agustus / September 2016 | * Kemampuan kalibrasi warna * Peningkatan fitur aplikasi * Aplikasi instan, setelan cepat * Mendukung multi layar * Desain baru layar antar muka |
| Oreo | 8.0 | Agustus 2017 | * Project Treble, arsitektur modular yang membuatnya lebih mudah dan lebih cepat bagi pembuat perangkat keras untuk menghadirkan pembaruan Android. * Dukungan untuk emoji Unicode 10.0 dan penggantian semua emo berbentuk gumpalan dengan yang bulat dengan gradien dan garis besar * Pengaturan dan Pengaturan Cepat yang didesain ulang dengan latar belakang putih dan hitam masing-masing. * Pengaturan yang direstrukturisasi dengan mengelompokkan kembali dalam beberapa bagian entri yang serupa. |
| Pie | 9.0 | Agustus 2018 | * Antarmuka pengguna baru untuk menu pengaturan cepat. * Tombol “tangkapan layar” telah ditambahkan ke opsi daya. * Mode “Lockdown” baru yang menonaktifkan otentikasi biometrik setelah diaktifkan. * Transisi baru untuk beralih antar aplikasi, atau aktivitas dalam aplikasi. * Pemberitahuan perpesanan yang lebih kaya, tempat percakapan lengkap dapat dilakukan dalam pemberitahuan, gambar skala penuh, dan balasan cerdas yang mirip dengan aplikasi baru Google |

1. Android Versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

1. Android Versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

1. Android Versi 1.6 (*Donut*)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan control applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder dan galeri yang dintegrasikan CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine, kemampuan dial kontak, teknologi *text to change speech* tidak tersedia pada semua ponsel, pengadaan resolusi VWGA.

1. Android Versi 2.0 / 2.1 (*Éclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1(Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (*killerapps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih.Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

1. Android Versi 2.2 (*Froyo: Frozen Yogurt*)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan renderingpada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

1. Android Versi 2.3 (*Gingerbread*)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 danWebM, efek audio baru (reverb, equalization, headphone virtualization, dan bassboost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

1. Android Versi 3.0/3.1 (*Honeycomb*)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukungukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan *platform* Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

1. Android Versi 4.0 (*Ice Cream Sandwich*)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pegenalan wajah, jaringan data pemantauan penggunaan dan control, terpadu kontak jaringan social, perangkat tambahan fotografi, mencari email secara offline, dan berbagai informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

1. Android Versi 4.1 – 4.3 (*Jelly Bean*)

Jellybean dimulai pada bulan Juni 2012 dengan merilis Android 4.1. Google dengan cepat merilis versi 4.2 dan 4.3, keduanya berada di bawah label Jelly Bean, masing-masing pada bulan Oktober 2012 dan Juli 2013.

Berikut adalah fitur yang menyertainya:

1. Google Now, tool bantu yang menampilkan informasi yang relevan berdasarkan riwayat pencarian.
2. Project Butter untuk mendukung frame rate yang lebih tinggi saat menggesek menu dan layar rumah.
3. Mampu melihat foto dengan menggesek dari kamera untuk menuju ke filmstrip.
4. Widget menyetel ulang diri mereka untuk menambahkan yang baru.
5. Pemberitahuan fitur yang lebih banyak
6. Fitur gerak dan aksesibilitas baru.
7. Android Versi 4.4 (*Kitkat*)

Android KitKat dirilis pada November 2013 ,Android 4.4 adalah satu-satunya versi OS yang benar-benar menggunakan nama sepotong permen.dan menjadi salah satu versi Android yang paling disukai oleh pengguna Smartphone di dunia.

KitKat memiliki fitur yang istimewa dari OS Android sebelumnya.  
Berikut adalah fitur yang menyertainya:

1. Immersive mode untuk konsumsi konten yang lebih baik.
2. Bar navigasi yang lebih baik untuk masuk dan keluar dari mode Immersive.
3. Dukungan widget layar kunci.
4. Dialer baru dengan fitur Caller ID.
5. Wallpaper layar penuh.
6. Emoji keyboard untuk emoticon.
7. Aplikasi Hangouts dan perpesanan terpadu
8. Dukungan cloud print yang lebih baik.
9. Integrasi Google Now yang lebih cerdas dan handsfree.
10. Android Versi 5.0 (*Lollipop*)

Android 5.0 Lollipop pertama kali diperkenalkan pada Mei 2014.,Android lollipopmerupakan perancangan ulang terbesar untuk Android. Smartphone Google Nexus 6, bersama dengan tablet Nexus 9-nya, merupakan perangkat pertama yang memiliki Lollipop yang telah terpasang sebelumnya.

Peningkatan terbesar yang dilakukan oleh Lollipop adalah pengenalan Material Design yang dengan cepat menjadi bahasa desain terpadu yang diterapkan di seluruh produk Google.

Berikut adalah fitur yang dimilikinya:

1. Dukungan pengaturan cepat yang lebih baik.
2. Masa pakai baterai yang disempurnakan dengan mode Battery
3. Saver yang baru.
4. Layar kunci baru
5. Fitur Smart Lock melalui Layanan Google Play.
6. Mode tamu untuk berbagi perangkat.
7. Pemasangan tombol.
8. Android Versi 6.0 (*Marshmallow*)

Android 6.0 (Marshmallow) Di rilis pada tahun 2015. Ini Perangkat pertama yang dikirim bersama Marshmallow yang telah terpasang sebelumnya adalah smartphone Google Nexus 6P dan Nexus 5X, dengan tablet Pixel C-nya. Tujuan marshmallow memoles sudut kasar dan membuat versi Lollipop lebih baik lagi.

Berikut adalah fitur yang dimilikinya:

1. Dukungan sidik jari resmi untuk perangkat.
2. Dukungan untuk pembayaran seluler melalui Android Pay.
3. Model perizinan yang lebih baik untuk aplikasi.
4. Google Now di Tap.
5. Deep menghubungkan Apps.
6. Andoird Versi 7.0 (*Nougat*)

Android 7.0 (Nougat) Dirilis pada Tahun ,2016. Sebelum Nougat terungkap "Android N" dirujuk secara internal oleh Google sebagai "New York Cheesecake".

Berikut adalah fitur yang dimilikinya:

1. Doze on the Go untuk waktu siaga yang lebih baik lagi.
2. Multi Window untuk penggunaan dua aplikasi secara bersamaan.
3. Aplikasi Setelan yang Lebih Baik
4. Hapus semua di layar aplikasi baru-baru ini.
5. Balas langsung ke pemberitahuan
6. Notifikasi dibundel
7. Pengaturan Cepat akan mengubah kustomisasi.
8. Android Versi 8.0 (*Oreo*)

Pada bulan Maret 2017, Google Rilis Android 8.0 Oreo,bulan Agustus, Google mengkonfirmasi Oreo akan menjadi nama publik untuk Android 8.0. Sperti yang kita ketahui Ini adalah kedua kalinya Google memilih nama merek dagan untuk Android (Oreo dimiliki oleh Nabisco).

Adapaun sekarang versi ini adalah semua yang baru yang dimilikinya,diantaranya adalah sebagai berikut:

1. Pemberitahuan untuk prioritas dan kategorisasi yang lebih baik.
2. Pengelolaan warna lebih baik.
3. Android O memiliki koleksi emoji baru yang telah didesain ulang.
4. Waktu boot lebih cepat: Pada perangkat Pixel, sekarang bisa mengalami waktu boot dua kali lebih cepat dibandingkan dengan Nougat.
5. Mengisi otomatis dan mengingat kata sandi dalam aplikasi.

### Sweet Home 3D

*Sweet Home 3D* adalah software gratis khusus untuk desain interior rumah dengan tampilan 3D secara instant. Aplikasi ini memiliki fitur yang lengkap yang tidak kalah dengan aplikasi dengan fitur yang sama namun berbayar. Terdapat 4 panel utama pada aplikasi *Sweet Home 3D* yang akan saya jelaskan dibawah ini:

1. Panel Utama

Pada panel ini menunjukkan daftar objek yang tersedia dan dikelompokkan ke dalam kategori: Kamar Mandi, Kamar Tidur, Pintu dan jendela, Dapur dan Ruang Tamu, sehingga memudahkan kita untuk menemukan objek furnitur yang diinginkan. Daftar ini juga menunjukkan gambar kecil dari bagian yang dipilih dan informasi tentang desainernya.

1. Panel *Grid*

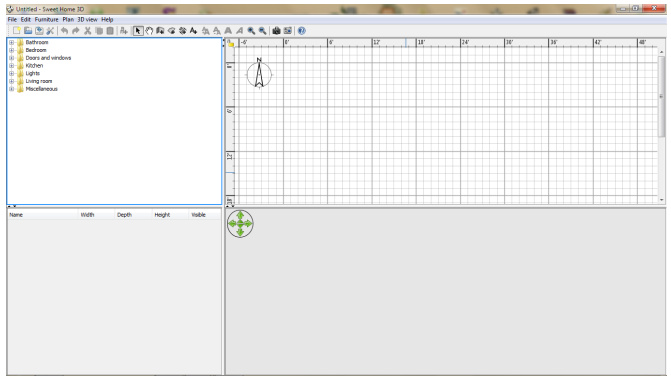
Disini memungkinkan kita untuk mengatur posisi furnitur di ruangan dan juga memberikan akses ke menu konteks dari mana kita dapat memilih untuk mengubah dimensi, membuat dinding, menghapus objek, zoom in dan out.

1. Panel Utama Kiri Bawah

Kita dapat melihat daftar rincian objek yang kita tambahkan ke rumah kita. Anda dapat memilih atribut *furniture* yang ingin Anda lihat dari daftar berikut: Nama, Lebar, Kedalaman, Tinggi, absis, Ordinat, Elevation, Angle, Warna, Movable, Door / Window dan Visible. Anda dapat memodifikasi atribut dan mengkonfigurasi dimensi objek dan posisinya.

1. Panel Pratinjau 3D

Panel ini dapat memberikan kita ide yang jelas tentang bagaimana jadinya rumah kita setelah kita dekorasi ulang. Dengan tampilan 3D tersebut kita dapat memutar tampilan ke sudut manapun, memperbesar atau memperkecil dan menangkap semua aspek dari desain rumah yang baru saja kita buat. Selain itu dalam aplikasi gratis Sweet Home 3D ini terdapat beberapa jenis modul perabotan dan aksesoris interior rumah yang memudahkan anda untuk membuat perencanaan interior design rumah anda (www.panduancara.com, 2013).



###### Gambar 2.19 Lembar Kerja *Sweet Home 3D*

### UML (*Unified Modeling Language*)

1. Pengertian UML

Menurut (Nugroho, 2009), ”UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek).” Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks, UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Apriliyanti, 2013).

1. Diagram UML

Berikut ini adalah diagram UML menurut (Henderi, 2009) yaitu:

1. *Use Case Diagram*

Secara grafis menggambarkan, interaksi secara sistem, sistem eksternal dan pengguna. Dengan kata lain *use case diagram* secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. Use case secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari tiap interaksi.

###### Tabel 2.2 Use Case Diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Actor | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
|  | Dependency | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
|  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
|  | *Include* | Menspesifikasikan bahwa use case sumber secara *eksplisit*. |
|  | Extend | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
|  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
|  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
|  | Use Case | Deskripsi dari urutan aksiaksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor. |
|  | Note | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

1. *Sequence Diagram*

Secara grafis menggambarkan bagaimana object berinteraksi satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi. *Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.

1. *State Diagram*

Untuk Digunakan untuk memodelkan *behaviour* objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek berbagai keadaan yang dapat diasumsikan oleh objek dan event-event (kejadian) yang menyebabkan objek dari satu state ke state yang lain. State diagram menggambarkan urutan keadaan yang dilalui *object* dalam suatu kelas, karena suatu kejadian menyebabkan suatu perpindahan aktivitas / state. State dari objek adalah penggolongan dari satu atau lebih nilai *attribute* pada kelas.

1. *Activity Diagram*

Secara grafis untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity Diagram* dapat juga digunakan untuk memodelkan action yang akan dilakukan saat operasi dieksekusi, dan memodelkan hasil dari action tersebut. *Activity Diagram* berupa *flow chart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

###### Tabel 2.3 Activity Diagram

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | Titik Awal |
|  | Titik Akhir |
|  | *Activity* |
|  | Pilihan untuk pengambilan keputusan |
|  | *Fork / Join*, untuk menunjukan kegiatan yang dilakukan parallel |
|  | *Rake*, menunjukan adanya dekomposisi |
|  | Tanda Waktu |
|  | Tanda Penerima |
|  | Aluran Akhir (*Flow Final*) |

1. *Class Diagram*

Menggambarkan struktur object sistem. Diagram ini menunjukan class diagram yang menyusun sistem dan hubungan antara class object tersebut. Diagram ini menunjukan class diagram yang menyusun sistem dan hubungan antara class object tersebut. *Class Diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. Objek adalah nilai tertentu dari setiap attribute kelas entity. *Class* menggambarkan keadaan (atribut / properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda / fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain.

## Metode Waterfall

Dalam perancangan aplikasi pada tugas akhir ini penulis menggunakan metode *waterfall*. Metode *Waterfall* ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut dengan “*Classic Life Cycle*” atau model *waterfall*. Model ini adalah model yang muncul pertama kali yaitu sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE).

*Sekuensial linier* mengusulkan sebuah pendekatan kepada pengembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut (Pressman, 2009) :

1. Rekayasa dan pemodelan sistem/informasi

Pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak.

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak, untuk memahami sifat program yang dibangun, perekayasa perangkat lunak (analis) harus memahami domain informasi, tingkah laku, unjuk kerja dan antarmuka (*interface*) yang diperlukan.

1. Desain

Desain perangkat lunak sebenarnya adalah proses *multi* langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural.

1. Generasi kode

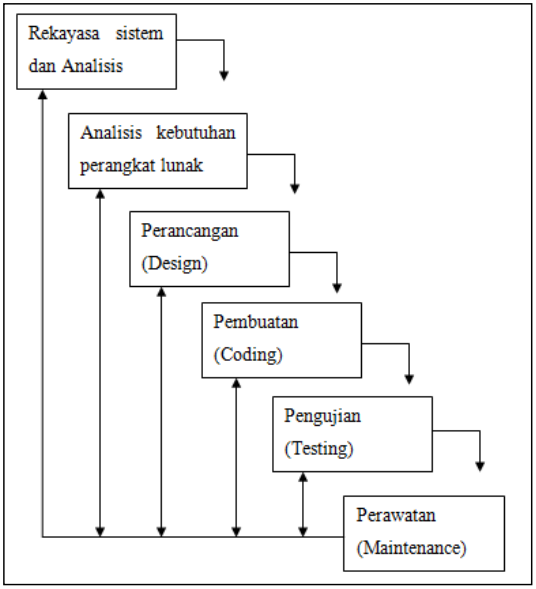
Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini.

1. Pengujian

Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

1. Pemeliharaan

Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi. Fase-fase pada Model *Waterfall* menurut (Somerville, 2003), yaitu:



###### Gambar 2.20 Model Waterfall