

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Dalam penelitian ini peneliti memaparkan satu penelitian terdahulu yang berjudul **Implementasi *Augmented Reality* (AR) menggunakan *Unity 3D* dan *Vuforia SDK*** yang dilakukan oleh Ida Bagus Mahendra. Penelitian ini membahas bagaimana mengembangkan aplikasi yang dapat menampilkan model hewan 3D dalam lingkungan *Augmented Reality* sehingga proses pembelajaran tidak membosankan. Dengan memanfaatkan teknologi *Augmented Reality* pada kartu, model dari hewan bisa ditampilkan secara *virtual* sehingga *user* bisa mengetahui bentuk dan animasi interaksi dari hewan yang ditampilkan dan juga dapat menarik minat untuk belajar lebih lanjut mengenai hewan tersebut (Mahendra, 2016).

Penelitian selanjutnya yang dilakukan oleh Atmoko Nugroho dan Basworo Ardi Pramono merupakan penelitian yang berjudul **Aplikasi *Mobile Augmented Reality* Berbasis *Vuforia* dan *Unity* Pada Pengenalan Objek 3D Dengan Studi Kasus Gedung M Universitas Semarang**. Penelitian ini merupakan penelitian yang dapat digunakan untuk membantu mahasiswa agar lebih mudah memahami materi objek 3D, selain itu juga sebagai sarana peneliti untuk menyajikan materi 3D objek dalam bentuk yang lebih interaktif. Dalam penggunaan teknologi *Augmented Reality* perlu diperhatikan kemiringan dan fokus kamera *smartphone* dengan *Marker* (Nugroho & Pramono, 2017).

Muhammad Rizqy Mubaraq, Helmi Kurniawan dan Alfa Saleh membuat

penelitian yang berjudul **Implementasi *Augmented Reality* pada media pembelajaran buah-buahan berbasis *Android***. Penelitian ini bertujuan untuk memberikan kemudahan dan relevansi untuk mendukung sistem belajar mengajar agar lebih baik secara kualitas maupun kuantitas kepada pengguna (*user*). Dengan tujuan untuk memberikan manfaat dan kemudahan terhadap pengguna dalam mengenal, mengetahui dan memahami. Pengujian pada aplikasi ini menunjukkan bahwa parameter jarak, tingkat kontras dari *Marker* dari sudut kemiringan kamera juga berpengaruh terhadap *output* objek 3D buah yang ditampilkan oleh aplikasi (Mubaraq, Kurniawan, & Saleh, 2018).

Selanjutnya penelitian yang dilakukan oleh Chaulina Alfianti Oktavia, Rosandi Fila Setiawan, dan Andrew Christanto adalah **Merancang aplikasi *Augmented Reality* untuk pengenalan ruangan menggunakan *Marker 3D objects tracking***. Aplikasi *Augmented Reality* Untuk Pengenalan Ruangan Menggunakan *Marker 3D Objects Tracking* ini untuk mahasiswa baru dapat mengetahui letak ruangan-ruangan yang ada di kampus, dengan begitu mahasiswa baru tidak lagi kesulitan untuk mengetahui letak ruangan-ruangan yang ada di kampus dan memudahkan mereka, dan juga dalam perancangan aplikasi ini terdapat beberapa kekurangan yang harus diperhatikan yaitu pada desain *3D Objects*, intensitas cahaya dalam hal ini berkaitan dengan keberhasilan memunculkan objek *virtual* (Oktavia, Setiawan, & Christianto, 2019).

Penelitian yang dilakukan oleh Rivi Hamdani dan Meini Sondang Sumbawati adalah penelitian yang berjudul **Pengembangan Media Pembelajaran Berbasis *Augmented Reality* Pada Mata Kuliah Sistem Digital di jurusan Teknik Informatika UNESA**. Pembuatan media pembelajaran yang berbasis *Augmented*

Reality sistem digital yang akan dipasang pada *handphone* dengan operasi sistem *Android* dan dibuat juga buku *Augmented Reality* sistem digital yang berisikan materi-materi dan gambar *Marker*. Dengan adanya media pembelajaran *Augmented Reality* yang menarik dapat mengurangi rasa jenuh dan bosan (Hamdani & Sumbawati, 2020).

II.2. Landasan Teori

Berikut ini adalah beberapa landasan teori yang dikutip dari beberapa referensi berupa jurnal-jurnal yang berkaitan:

II.2.1. Perancangan

Perancangan adalah langkah pertama dalam fase pengembangan rekayasa produk atau sistem. Perancangan itu adalah proses penerapan berbagai teknik dan prinsip yang bertujuan untuk mendefinisikan sebuah peralatan, satu proses atau satu sistem secara detail yang membolehkan dilakukan realisasi fisik (Nadeak, dkk, 2016).

II.2.2. Aplikasi

Istilah aplikasi berasal dari bahasa inggris "*application*" yang berarti penerapan, lamaran ataupun penggunaan. Sedangkan secara istilah, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Aplikasi biasanya berupa perangkat lunak yang berbentuk *software* yang berisi kesatuan perintah atau program yang dibuat untuk melaksanakan sebuah pekerjaan yang diinginkan (Nurdiana & Zarkasi, 2017).

II.2.3. Logo

Logo merupakan huruf atau lambang yang mengandung makna, terdiri atas satu kata atau lebih sebagai lambang atau nama perusahaan dan sebagainya (Sumber: <https://kbbi.kemdikbud.go.id/entri/logo> diakses pada 16:45 26 Januari 2022).

II.2.4. Universitas

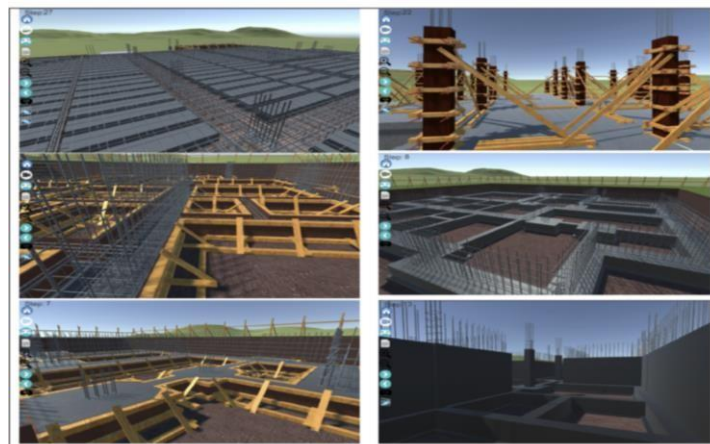
Universitas merupakan perguruan tinggi yang terdiri atas sejumlah fakultas yang menyelenggarakan pendidikan ilmiah dan/atau profesional dalam sejumlah disiplin ilmu tertentu (Sumber: <https://kbbi.kemdikbud.go.id/entri/universitas> diakses pada 10:39 21 Desember 2021).

II.2.5. *Augmented Reality*

Augmented Reality merupakan sebuah terobosan dan inovasi bidang multimedia dan *image processing* yang sedang berkembang. Teknologi ini mampu mengangkat sebuah benda yang sebelumnya datar atau dua dimensi seolah-olah menjadi nyata, bersatu dengan lingkungan sekitarnya.

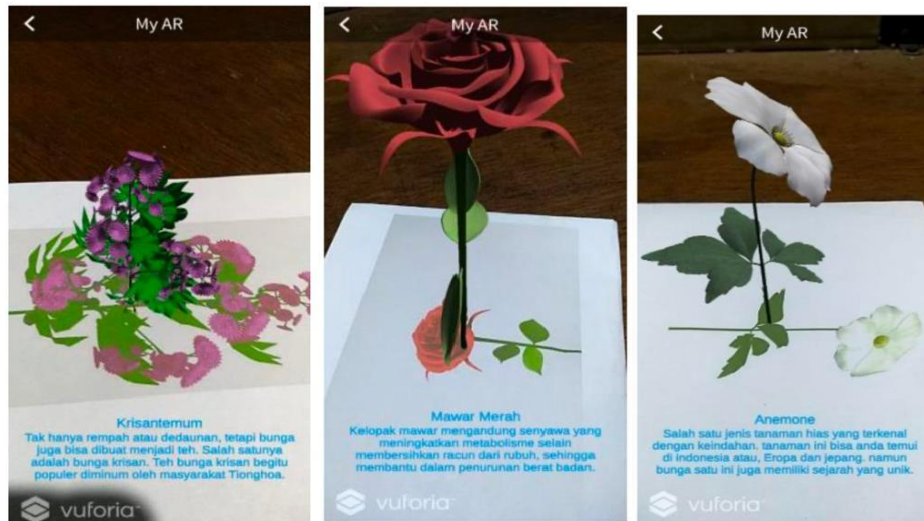
Augmented Reality adalah sebuah variasi dari *Virtual Environment* atau yang lebih dikenal dengan *Virtual Reality*. Teknologi *Virtual Reality* dalam penggunaannya menempatkan pengguna ke dalam lingkup *Virtual* sehingga pengguna dapat merasakan sensasi masuk ke dalam lingkungan aplikasi. Sementara itu, secara bersamaan teknologi *Augmented Reality* mampu menambahkan realita di dunia nyata dengan unsur objek *Virtual* di mana batas antara dinding dunia nyata dan dunia maya seakan tidak ada (Arifitama, 2017).

Sistem *Augmented Reality* pada umumnya bekerja berdasarkan pendeteksian citra, yang sering disebut dengan istilah *Marker*. Prinsip kerjanya sebenarnya cukup sederhana. Kamera atau *webcam* akan mendeteksi *Marker* yang diberikan, kemudian sistem akan mengenali dan menandai pola *Marker*, citra yang tertangkap kamera atau *webcam* akan diolah oleh sistem dengan melakukan perhitungan apakah *Marker* sesuai dengan *database* yang dimiliki. Informasi citra yang diterima tidak akan diolah bila *Marker* tidak sesuai dengan *database* yang dimiliki sistem, tetapi bila sesuai maka informasi tersebut akan digunakan untuk menampilkan teks, video, objek 3D atau animasi yang telah dibuat sebelumnya (Maulana, 2017). Gambar berikut dapat mendeskripsikan perbandingan antara teknologi *Virtual Reality* dan *Augmented Reality*.



**Gambar II.2.5. 1 Contoh Aplikasi Penggunaan Virtual Reality
(Sumber: Bashabsheh, dkk, 2019)**

Konsep teknologi *Virtual Reality* adalah interaksi yang dilakukan oleh pengguna sebagai *user* terasa seakan berada di dalam dunia maya atau 3D. Sehingga proses pengembangan aplikasi lebih ditekankan pada pembuatan lingkungan serta interaksi yang dapat dilakukan pada objek 3D secara *realtime* oleh pengguna. Sedangkan *Augmented Reality* dapat dilihat pada Gambar II.2.5.2, sebagai berikut:



Gambar II.2.5.2 Aplikasi Penggunaan *Augmented Reality*
(Sumber: Khoiruddin, 2020)

Teknologi *Augmented Reality* yang merupakan pengembangan dari *Virtual Reality* memiliki konsep yang berbeda. Ketika *Virtual Reality* menarik pengguna seakan masuk ke dalam lingkungan 3D, maka *Augmented Reality* menambahkan realita yang ada dan nyata di dunia kita dengan objek yang terangkat (*Augmented*). Dimana teknologi ini seakan menghilangkan dunia maya 3D, menyatu dengan dunia nyata.

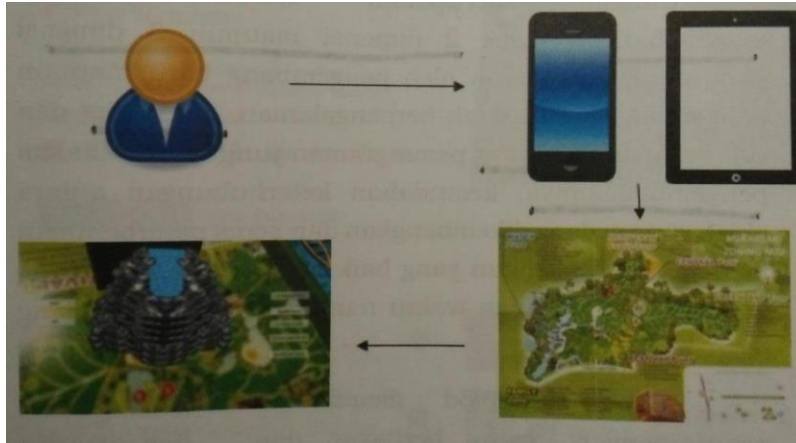
Terdapat empat komponen yang harus diperhatikan dalam hal pengembangan dan penggunaan *Augmented Reality*, yaitu antara lain sebagai berikut:

1. Komponen pertama yaitu perangkat keras. Perangkat keras yang digunakan dapat berupa PC, laptop, *smartphone* maupun *tablet*. PC dan laptop digunakan sebagai alat pengembangan aplikasi, sedangkan *smartphone* dan *tablet* digunakan sebagai dasar tempat aplikasi akan ditanamkan atau diinstal.
2. Komponen kedua yaitu perangkat lunak hasil pengembangan yang telah

dilakukan di sebuah *software* pembentuk aplikasi *Augmented Reality*. *Platform* dari aplikasi saat ini dapat dijalankan pada *platform* PC, *Android*, dan *IOS*.

3. Komponen ketiga adalah alat pemindaian atau *Scanner* untuk melakukan pemindaian pola dan mengaktifkan *Augmented Reality*. Alat yang dapat digunakan sebagai alat pengindai berupa *webcam* untuk PC maupun kamera yang sudah tersedia pada *smartphone* maupun *tablet*.
4. Komponen terakhir adalah *Marker* sebagai lokasi titik kemunculan dari objek *Augmented Reality*. Pengembangan dari *Marker* sendiri memerlukan Teknik pembentukan pola. Pola dapat berupa, hitam-putih atau non pola. Setiap pola akan terlebih dahulu diuji seberapa kompatibel atau layak digunakan sebagai *Marker* dengan *software* khusus (Arifitama, 2017).

Penggunaan teknologi *Augmented Reality* dapat dilihat pada Gambar II.2.5.3, dimana *user* sebagai *user* pertama, mengambil perangkat pertama sebagai *smartphone* atau *tablet* yang telah *terinstall* aplikasi *Augmented Reality* dan menjalankan aplikasi. Kemudian, *user* dengan perangkat yang digenggamnya melakukan pemindaian terhadap *Marker* yang sudah disediakan, dan ketika *user* memanfaatkan teknologi *Augmented Reality*, objek akan muncul sehingga memungkinkan pengguna untuk berinteraksi dengan objek.



Gambar II.2.5. 3 Mekanisme Penggunaan Teknologi *Augmented Reality*
(Sumber: Arifitama, 2017)

II.2.6. *Unity*

Aplikasi *Unity 3D* merupakan *game engine* merupakan sebuah *software* pengolah gambar, grafik, suara, *input* yang ditujukan untuk membuat suatu *game* ataupun tidak harus untuk membuat *game*. Kelebihan dari *game engine* ini adalah bisa membuat *game* 3D dan 2D dan sangat mudah digunakan (Nugroho & Pramono, 2017).

Javascript dan *C#* merupakan bahasa pemrograman yang dipakai dalam pengembangannya, kemudahan keterhubungan antara objek yang sedang dikembangkan dan *script* pemrograman menjadikannya pilihan yang baik bagi pengembang yang memiliki keterbatasan waktu namun memiliki segudang ide.

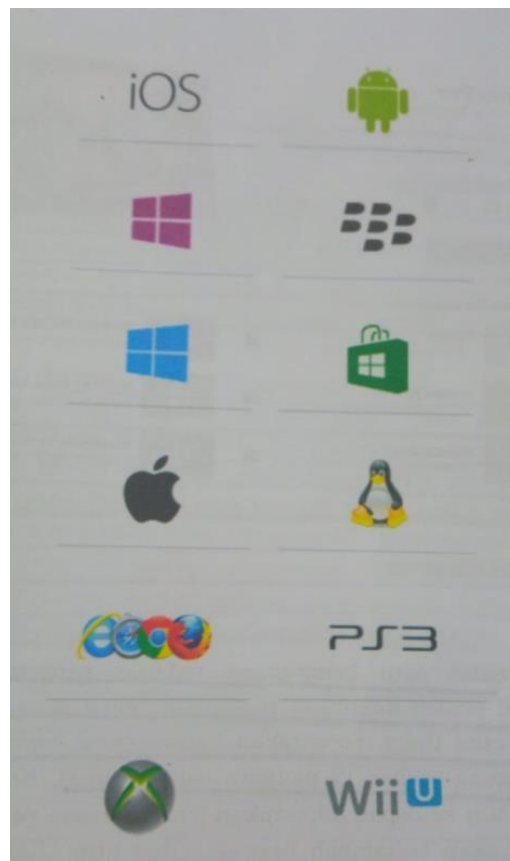
Adapun *Unity 3D* memiliki 2 plan lisensi pengembangan, yaitu berbayar dan *free* dengan keterbatasan fitur. Namun, keterbatasan tersebut tidak begitu terasa untuk pengembangan aplikasi sederhana. Fitur yang diberikan oleh *Unity3D* sudah cukup untuk pengembangan aplikasi baru dalam mempublikasikan aplikasi buatannya dan menjualnya secara *free* tanpa royalti kepada pihak *Unity3D*. Namun, tetap dengan ketentuan dan persyaratan yang diberikan oleh pihak

Unity3D.

Kebijakan lisensi penggunaan *Unity 3D* adalah, pengembang baru dapat menggunakan *Unity* di PC secara gratis maupun berbayar. Setelah mempelajari versi gratis dari *unity*, pengembang juga dapat membeli versi Pro perizinan pada *Unity Pro* dapat diubah.

Salah satu keuntungan terbesar penggunaan *Unity3D* adalah dukungan *multiplatformnya* yang luas. Situs resmi *Unity* mengatakan bahwa *game engine* ini mendukung sekitar 12 platform yang berbeda (Arifitama, 2017).

Daftar *platform* yang didukung oleh *Unity* saat ini dapat dilihat pada Gambar II.2.5.1.1, sebagai berikut:



Gambar II.2.6.1 Dukungan *Multiplatform Unity*
(Sumber: Arifitama, 2017)

II.2.6.1. *Vuforia SDK*

Vuforia adalah *Augmented Reality Software Development Kit* (SDK) untuk perangkat *mobile* yang memungkinkan pembuatan aplikasi *Augmented Reality*. Menggunakan teknologi *Computer Vision* untuk mengenali dan melacak gambar (*Target Image*) dan objek 3D sederhana, seperti kotak dan kaleng secara *real-time*.

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

1. **Kamera**

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

2. **Image Converter**

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

3. **Tracker**

Mengandung algoritma *computer vision* yang dapat mendeteksi dan melacak objek dunia nyata yang ada pada video kamera.

4. **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.

5. **Target Resources**

Dibuat menggunakan *on-line Target Management System*. *Assets* yang

diunduh berisi sebuah konfigurasi xml yang memungkinkan *developer* untuk mengkonfigurasi beberapa fitur dalam *trackable* dan *binary file* yang berisi *database trackable* (Wijaya & Purba, 2018).

6. *Application Code*

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

- a. *Query state object* pada target baru yang terdeteksi atau *Marker*.
- b. *Update* logika aplikasi setiap *input* baru dimasukkan.
- c. *Render* grafis yang ditambahkan (*Augmented*). (Rahayu, Yusmansyah & Pratama, 2017)

Vuforia SDK sendiri telah mendukung pengembangan aplikasi untuk perangkat yang berbasis *iOS* dan *Android*. Fitur pembuatan *Marker* yang telah disediakan oleh *Vuforia SDK* memudahkan para pengembang baru untuk membuat *Marker* sesuai dengan kebutuhan aplikasi, yang akan digunakan dan di-extract ke dalam *software Unity3D*. jenis *Marker* yang dapat dibuat ada dua, yaitu *Marker* berpola dan *Markerless*.

Marker berpola adalah salah satu jenis *Marker* yang dapat digunakan sebagai penanda lokasi dari objek 3D yang akan ter-*augmented*. Bentuk dari *Marker* berpola bercirikan warna hitam putih serta membentuk pola tertentu. *Marker* berpola dapat dilihat pada Gambar II.2.5.2.1, sebagai berikut:



Gambar II.2.6.1. 1 Marker Berpola
(Sumber: Satria & Prihandoko, 2018)

Marker Markerless merupakan jenis *Marker* yang digunakan dan berguna sama seperti *Marker* berpola, yaitu sebagai penanda lokasi objek *Augmented Reality*, namun bedanya hanya dari bentuk *Marker* tersebut. *Marker* tidak berpola memiliki kekhasan, yaitu dapat berupa gambar apapun selama mengandung pola di dalamnya walaupun secara tersirat tidak terlihat. *Marker Markerless* dapat dilihat pada Gambar II.2.5.2.2, sebagai berikut:



Gambar II.2.6.1. 2 Marker Markerless
(Sumber: Rahayu, Yusmansyah, Pratama, 2017)

Vuforia SDK memudahkan dan mempercepat kerja pengembang dalam membuat aplikasi dengan teknologi *Augmented Reality* karena *library* dan fungsi-fungsi intinya sudah dibuat oleh *Qualcomm*. Pengembang tinggal berimajinasi dan mengembangkan aplikasi menarik menggunakan SDK ini.

II.2.6.2. 3D (Tiga Dimensi)

Berbeda dengan desain 2 dimensi yang hanya mengenal 2 parameter yaitu

panjang dan lebar. Dalam konsep 3D, kita bisa mendapatkan dimensi ketebalan. Objek 3D dipresentasikan dalam sebuah bidang yang tergolong memiliki 3 buah koordinat *axis* yang terdiri dari sumbu X, Y, dan Z. *Axis* X adalah *axis* mendatar atau *horizontal*, *axis* Y adalah *axis* tegak atau *vertikal*, sedangkan *axis* Z adalah *axis* yang menembus layar monitor kedalam (menunjukkan kedalaman ruang) (Aslah, dkk, 2017).

II.2.7. *Android*

Android adalah sistem operasi *mobile phone* berbasis *linux*. *Android* bersifat *opensource* yang *source codenya* diberikan gratis bagi para pengembang untuk menciptakan aplikasi mereka. agar dapat berjalan di *Android*. Pada mulanya, *Android* adalah salah satu produk dari *Android Inc.*, namun *Google Inc.* mengakuisisi *Android Inc.*, dan semua kekayaan intelektual milik *Android Inc.* diperoleh *Google Inc.* yang kemudian mengembangkan kembali system *Android* (Wasil, dkk, 2020).

Android merupakan satu kumpulan lengkap perangkat lunak yang dapat berupa sistem operasi, *middleware*, dan aplikasi kunci perangkat *mobile*. *Android* terdiri dari satu tumpukan yang lengkap, mulai dari *boot loader*, *device driver*, dan fungsi-fungsi pustaka, hingga perangkat lunak API (*Application Programming Interface*), termasuk aplikasi SDK (*Software Development Kit*) (Khairunnisyah & Naswiro, 2016).

II.2.7.1. Sejarah *Android*

Perjalanan *Android* dimulai sejak Oktober 2003 ketika 4 orang pakar IT, Andi Rubin, Rich Miner, Nick Sears dan Chris White mendirikan *Android.Inc*,

di California US. Visi *Android* untuk mewujudkan *mobile device* yang lebih peka dan mengerti pemiliknya, kemudian menarik raksasa dunia maya *Google*. *Google* kemudian mengakuisisi *Android* pada Agustus 2005. OS *Android* dibangun berbasis platform *Linux* yang bersifat *open source*, senada dengan *Linux*, *Android* juga bersifat *Open Source*. Dengan nama besar *Google* dan konsep *open source* pada OS *Android*, tidak membutuhkan waktu lama bagi *Android* untuk bersaing dan menyisihkan *Mobile OS* lainnya seperti *Symbian*, *Windows Mobile*, *Blackberry* dan *iOS*. Kini siapa yang tak kenal *Android* yang telah menjelma menjadi penguasa *Operating System* bagi *Smartphone* (Lengkong, Sinsuw & Lumenta, 2015).

II.2.7.2. Struktur Aplikasi Android

Struktur aplikasi *Android* atau fundamental aplikasi ditulis dalam bahasa pemrograman *Java*. Kode *Java* dikompilasi bersama dengan *resource file* yang dibutuhkan oleh aplikasi. Dimana prosesnya di *package* oleh *tools* yang dinamakan *apt tools* kedalam paket *Android*. Sehingga menghasilkan file dengan ekstensi *apk*. File *apk* ini yang disebut dengan aplikasi, dan nantinya dapat dijalankan pada peralatan mobile. Ada empat komponen pada aplikasi *Android*, yaitu antara lain sebagai berikut:

1. *Activities*

Activities merupakan komponen untuk menyajikan tampilan pemakai (*user interface*) kepada pengguna.

2. *Service*

Service merupakan komponen yang tidak memiliki tampilan pemakai (*user interface*), tetapi *service* berjalan secara *backgrounds*.

3. **Broadcast Receiver**

Broadcast Receiver merupakan komponen yang berfungsi menerima dan bereaksi untuk menyampaikan notifikasi.

4. **Content Provider**

Content Provider merupakan komponen yang membuat kumpulan aplikasi data secara spesifik, sehingga bisa digunakan aplikasi lain (Dewi, dkk, 2018).

II.2.7.3. 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*) (Dewi, dkk, 2018).

II.2.7.4. Perkembangan Android

Android beberapa kali melakukan pembaruan versinya, kebanyakan nama untuk setiap versinya diambil dari nama makanan. Berikut versi-versinya:

1. *Android v.1.0 alpha*
2. *Android v1.1 bender (beta)*
3. *Android v1.5 cupcake*
4. *Android v1.6 donut*
5. *Android v2.0 – v2.1 éclair*
6. *Android v2.2 frozen yoghurt (froyo)*

7. *Android v2.3 gingerbread*
8. *Android v3.0 – v3.2 honeycomb*
9. *Android v4.0 ice cream sandwich*
10. *Android v4.1 – 4.3.1 jelly bean*
11. *Android v4.4 kitkat*
12. *Android v5.0 – 5.1 lollipop*
13. *Android v6.0 marshmallow*
14. *Android v7.0 – 7.1.2 nougat*
15. *Android v8.0 oreo*

Android memiliki keunggulan seperti mudah dikustomisasi dan dimodifikasi, merupakan salah satu sistem operasi yang cepat dan responsif yang sangat nyaman untuk penggunaannya, sistem operasi yang dapat diaplikasikan di banyak peralatan elektronik, *Android* mendukung segala layanan *google* mulai dari gmail sampai *google reader* (Lambonan, Sengkey & Najoran, 2019).

II.2.7.5. SDK Android Manager

SDK Android Manager merupakan *toolkit* yang digunakan sebagai *emulator* ketika aplikasi selesai dibangun. Kegunaan dari *SDK Manager* untuk pengembangan *Augmented Reality* tidak hanya sampai tahap *deployment* ke dalam bentuk aplikasi “.apk”. Tanpa adanya instalasi *SDK Android Manager*, maka aplikasi yang telah selesai dikembangkan tidak dapat dilakukan *deployment* ke bentuk aplikasi *Android* dengan format ekstensi “.apk” (Arifitama, 2017).

II.2.8. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah bahasa spesifikasi standar yang

dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Hendini, 2016).

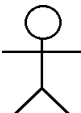
II.2.8.1. Use Case Diagram

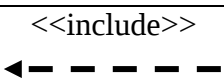
Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Hendini, 2016).

Use case merupakan diagram yang harus dibuat pertama kali saat pemodelan perangkat lunak berorientasi objek yang dilakukan (Ayu & Fitri, 2019).

Berikut ini merupakan simbol-simbol yang terdapat pada *Use Case Diagram*, dapat dilihat pada Tabel II.2.8.1 sebagai berikut:

Tabel II.2.8.1 Simbol Use Case Diagram

No	Gambar	Nama	Keterangan
1		<i>Use case</i>	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
2		<i>Actor / Aktor</i>	<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

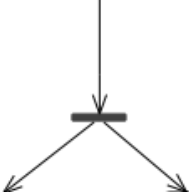
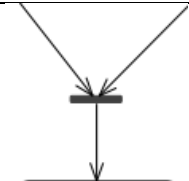
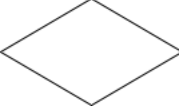
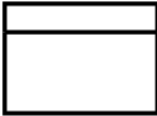
			yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa <i>actor</i> berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
3		<i>Asosiasi</i>	Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
4		<i>Asosiasi</i>	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
5		<i>Include</i>	<i>Include</i> , merupakan di dalam use case lain (<i>required</i>) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.
6		<i>Extend</i>	<i>Extend</i> , merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.

(Sumber: Hendini, 2016)

II.2.8.2. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Berikut ini merupakan simbol-simbol yang terdapat pada *Activity Diagram*, dapat dilihat pada Tabel II.2.8.2 sebagai berikut:

Tabel II.2.8.2 Simbol Activity Diagram

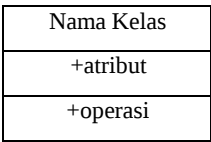
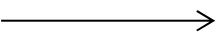
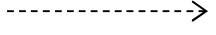
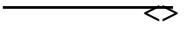
No	Gambar	Nama	Keterangan
1		<i>Start Point</i>	<i>Starty Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
2		<i>End Point</i>	<i>End Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
3		<i>Activities</i>	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
4		<i>Fork/Percabangan</i>	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5		<i>Join/Penggabungan</i>	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
6		<i>Decision Points</i>	<i>Decision Points</i> menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
7		<i>Swimlane</i>	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber: Hendini, 2016)

II.2.8.3. Class Diagram

Class Diagram atau kelas diagram salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. ia bersifat statis, dalam artian diagram kelas bukan menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi. Berikut ini merupakan simbol-simbol yang terdapat pada *class diagram*, dapat dilihat pada Tabel II.2.8.3 sebagai berikut:

Tabel II.2.8.3 Simbol Class Diagram

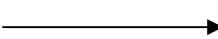
No	Gambar	Nama	Keterangan
1		<i>Class</i>	Kelas pada struktur <i>system</i> .
2		Antar muka / <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3		Asosiasi <i>/Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i> .
4		Asosiasi berarah/ <i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5		Generalisasi	Relasi antar kelas dengan makna generalisasi - spesialisasi (umum khusus).
6		Kebergantungan <i>/ Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7		Agresiasi / <i>Aggregation</i>	Relasi antar kelas dengan makna semua-bagian.

(Sumber: Ayu & Fitri, 2019)

II.2.8.4. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antarobjek (Hendini, 2016). Berikut ini merupakan simbol-simbol yang terdapat pada *Sequence Diagram*, dapat dilihat pada Tabel II.2.8.4 sebagai berikut:

Tabel II.2.8.4 Simbol Sequence Diagram

No	Gambar	Nama	Keterangan
1		<i>Entity Class</i>	Bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
2		<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
3		<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.
4		<i>Message</i>	Simbol mengirim pesan antar <i>class</i> .
5		<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.

6		<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .
7		<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

(Sumber: Hendini, 2016)