

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka yang diambil dari beberapa penelitian sebelumnya yang digunakan sebagai rujukan atau pendukung penelitian yang sedang dilakukan, diantaranya :

2.1.1 Tinjauan Terhadap Literatur 01

Penulis Alvino Octaviano (2018) dengan judul “PENERAPAN *AUGMENTED REALITY* BERBASIS ANDROID SEBAGAI BAHAN AJAR UNTUK MESIN INJECTION”. Penelitian yang dilakukan penulis adalah membuat media pembelajaran mengenai sistem injeksi sepeda motor honda, di karenakan banyak faktor yang menyebabkan siswa mendapatkan nilai rendah dalam kompetensi memahami sistem bahan bakar mesin khususnya sistem *injektion* pada sepeda motor honda (hanik, 2015). Faktor kurangnya persediaan alat praga mesin sistem PGM-FI yang menyebabkan siswa kurang memahami materi *injektion* di sekolah SMK Fadilah. Metode penelitian yang digunakan untuk membangun sistem pada penelitian ini mengadaptasi metode pengembangan sistem *Rapid Application Development* (RAD) sistem ini mempunyai 3 fase, yaitu kebutuhan perencanaan, proses desain RAD dan fase implementasi (kendall, 2010). Hasil dari penelitian ini adalah panduan pengenalan mesin injection di SMK Fadilah berbasis *Augmented reality* pada perangkat android.

2.1.2 Tinjauan Terhadap Literatur 02

Penulis Ilmawan Mustaqim (2017) dengan judul “PENGEMBANGAN MEDIA PEMBELAJARAN BERBASIS *AUGMENTED REALITY*”. Penelitian yang dilakukan penulis adalah melihat sistem pembelajaran yang di perlukan guru untuk membantu menyampaikan materi dalam sebuah proses pembelajaran. Proses pembelajaran yang baik haruslah membuat aspek interaktif dan memberikan ruang yang lebih bagi siswa untuk dapat mengembangkan kreatifitas dan kemandirian. Kegiatan pembelajaran yang menyenangkan sangat di pengaruhi oleh berbagai faktor, salah satunya adalah pemilihan media pembelajaran yang di gunakan haruslah menarik dan interaktif namun tidak mengurangi isi materi yang di sampaikan. Tujuan d penelitian ini adalah membantu proses pembelajaran

yang di butuhkan oleh guru untuk menyampaikan materi dalam sebuah proses pembelajaran. Metode yang digunakan dalam penelitian ini adalah menambahkan objek maya kedalam objek nyata di waktu yang bersamaan. Hasil dari penelitian ini adalah menerapkan aplikasi berbasis *Augmented Reality* untuk media pembelajaran di sekolah.

2.1.3 Tinjauan Terhadap Literatur 03

Penulis Hellik Hermawan (2019) dengan judul “PENGEMBANGAN MEDIA PEMBELAJARAN MESIN MENGGUNAKAN TEKNOLOGI *AUGMENTED REALITY*”. Penelitian ini bertujuan membantu bidang pendidikan terutama bagi pengajar dengan teknologi yang di gunakan sebagai media pembelajaran untuk menampilkan objek 3D yaitu *Augmented Reality*, penggunaan *Augmented Reality* telah memudahkan pengajar memahami suatu imajiner dan meningkatkan keterampilan siswa. Metode yang digunakan pada penelitian ini adalah metode pengembangan sistem, pada penelitian ini metode pengembangan menggunakan metode *multimedia development life cycle* (MDLC) dengan versi *luther-sutopo* pada metode ini terdapat 6 tahapan yaitu pengonsepan, perancangan, pengumpulan materi, pembuatan, pengujian dan pendistribusian. Metode pengumpulan data menggunakan beberapa metode yaitu pengamatan (*observasi*), wawancara (*interview*), dan studi pustaka. Hasil akhir dari penelitian ini berupa aplikasi berbasis *augmented reality markerless* ini memiliki interaksi berupa *login*, *register*, tombol perpindahan dari *scene* satu ke *scene* lainnya dan perubahan objek pada tampilan utama maka akan muncul menu 12 materi yang akan di pelajari. Setelah memilih materi yang akan di pelajari, akan masuk ke kamera untuk memindai objek, setelah di arah kan ke objek dan objek berhasil di pindai, akan muncul gambar objek di sertai dengan keterangan.

2.1.4 Tinjauan Terhadap Literatur 04

Penulis Muhammad Dzikrillah (2017) dengan judul “MEMBANGUN APLIKASI *AUGMENTED REALITY* PADA SIMULASI PEMELIHARAAN SISTEM KARBURATOR DAN BUSI PADA MOBIL BERBASIS DEKSTOP”. Penelitian yang dilakukan penulis adalah melihat bahwa media pembelajaran dan simulasi sistem pemeliharaan mesin dapat meml siswa dan guru melakukan proses belajar dan menjelaskan kepada siswa bagaimana langkah - langkah sistem pemeliharaan mesin pada mobil dengan komponen karburator dan busi. Masalah yang akan dibahas dalam penelitian

ini adalah bagaimana siswa agar dapat lebih mengerti sistem pemeliharaan mesin, dan kurangnya alat yang di gunakan pada saat praktikum. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode penelitian dan pengembangan (*research and development*). Dan metode pembangunan perangkat lunak menggunakan model *waterfall*. Hasil dari penelitian ini untuk mempermudah mahasiswa dan pendidik dalam proses belajar mengajar mengenai karburator dan busi pada mobil, dan aplikasi media pembelajaran sistem pemeliharaan mesin ini dapat menjadi alat penghubung untuk mengatasi keterbatasan alat praktikum.

2.1.5 Tinjauan Terhadap Literatur 05

Penulis Aufa Anggun Probo Kusumo (2016) dengan judul “RANCANG BANGUN APLIKASI 3D *SPARE PART* SEPEDA MOTOR UNTUK PEMBELAJARAN TEKNIK OTOMOTIF”. Penelitian yang dilakukan penulis adalah melihat sering terjadinya miss komunikasi antar murid dan guru di dalam pembelajaran, pencitraan ilustrasi pemodelan dari mesin - mesin yang di pelajari masih menggunakan gambaran 2D yang kurang merinci bentuk *part* mesin sepeda motor. Salah satu media informasi yang efektif dan efisien adalah aplikasi yang di dalamnya terdapat pemodelan *part* mesin sepeda motor dengan bentuk 3D menggunakan teknologi *Augmented Reality* berserta penjelasan serta penggolongan *part* mesin sepeda motor. Dengan adanya aplikasi informasi yang berbasis android inilah, di harapkan informasi akan lebih mudah di terima oleh siswa. Metode penelitian yang digunakan adalah metode studi kepustakaan, metode observasi dan metode wawancara. Hasil dari penelitian ini bertujuan membuat aplikasi 3D *spare part* mesin sepeda motor untuk pengenalan mesin sepeda motor melalui mata pelajaran “Proses-proses mesin konversi energi” kelas 1 SMK Adi ~ mo.

2.1.6 Tinjauan Terhadap Literatur 06

Penulis Rusliyawati, Agus Wantoro² Adit Nurmansyah, (2020) dengan judul “PENERAPAN *AUGMENTED REALITY* (AR) DENGAN KOMBINASI TEKNIK MARKER UNTUK VISUALISASI MODEL RUMAH PADA PERUM PRAMUKA GARDEN RESIDENCE” penelitian yang dilakukan oleh penulis adalah membantu

penjualan dan promosi bangunan rumah melalui media AR yang mana AR dapat digunakan membantu visualisasi konsep abstrak untuk pemahaman dan struktur suatu model objek yang dirancang memberikan informasi lebih detail pada pengguna dari objek nyata. Dengan adanya aplikasi ini penjualan dapat lebih meningkat dan menarik banyak konsumen. Metode yang di gunakan adalah metode pengujian yang dilakukan yaitu *Extrime Programming*, Aplikasi pemodelan rumah pada perumahan pramuka garden residence menggunakan metode *Alpha* dan *Beta* dengan *Augmented Reality* yang dijalankan pada smartphone. Hasil penelitian yang di hasilkan adalah meningkatkan daya jual dan promosi rumah agar lebih menarik dan di minati oleh konsumen.

2.2 **Augmented Reality**

Menurut penjelasan Haller, Billinghurst, dan Thomas (2007), riset *Augmented Reality* bertujuan untuk mengembangkan teknologi yang memperbolehkan penggabungan secara *real-time* terhadap digital *content* yang dibuat oleh komputer dengan dunia nyata. *Augmented Reality* memperbolehkan pengguna melihat objek maya dua dimensi atau tiga dimensi yang diproyeksikan terhadap dunia nyata. (*Emerging Technologies of Augmented Reality: Interfaces and Design*).

Ronald T. Azuma (1997) mendefinisikan *Augmented Reality* adalah penggabungan benda-benda nyata dan maya di kehidupan nyata, berjalan secara interaktif dalam waktu nyata, dan dapat di integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan i nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat input tertentu, dan integrasi yang baik memerlukan hasil yang efektif.

Augmented Reality (AR) menurut Brian (2012) merupakan suatu istilah yang berkaitan dengan lingkungan yang menggabungkan dunia nyata dengan dunia visual yang diciptakan oleh komputer menjadikan batas antara keduanya menjadi sangat tipis. Sistem ini lebih mengarah terhadap lingkungan secara nyata atau real, realitiy lebih mengutamakan sistem ini.

Augmented Reality (AR) bertujuan untuk mengambil dunia nyata sebagai dasar dengan menggabungkan beberapa teknologi *virtual* dan menambahkan data yang konteks agar pemahaman manusia sebagai penggunaanya menjadi semakin jelas. Data ini dapat

berupa komentar audio, data lokasi, konteks sejarah atau dalam bentuk lainnya (Rahmat, 2011).

Dapat disimpulkan *Augmented Reality* adalah teknologi yang menggabungkan benda maya dua dimensi atau tiga dimensi kedalam sebuah lingkungan nyata tiga dimensi dan membuat benda-benda maya tersebut dalam waktu nyata. *Augmented Reality* bertujuan untuk mengambil dunia nyata sebagai dasar dengan menggabungkan pada beberapa teknologi *virtual* dan menambahkan data konstektual agar pemahaman manusia sebagai penggunaanya menjadi semakin rinci. Data konstektual ini dapat berupa komentar audio, data lokasi, konteks sejarah atau dalam bentuk lainnya (Rahmat, 2011).

Dengan bantuan teknologi *Augmented Reality*, lingkungan nyata disekitar kita akan dapat dibuat dalam bentuk *virtual*. Informasi-informasi tentang objek dan lingkungan disekitar kita dapat ditambahkan ke dalam sistem *Augmented Reality* yang kemudian informasi tersebut ditampilkan diatas layar dunia nyata secara *real time* seolah-olah informasi tersebut adalah nyata (Fernando, 2013).

Tujuan dari *Augmented Reality* adalah mengambil dunia nyata sebagai dasar dengan menggabungkan beberapa teknologi *virtual* dan menambahkan data konstektual agar pemahaman manusia sebagai penggunaanya menjadi semakin jelas. Data konstektual ini dapat berupa komentar audio, data lokasi, konteks sejarah, atau dalam bentuk lainnya. Saat ini *Augmented Reality* telah banyak digunakan dalam berbagai bidang seperti kedokteran, militer, manufaktur, hiburan, museum, *game*, pendidikan, dan lain-lain. (Rahmat,2011).

Dalam perkembangannya *Augmented Reality* telah digunakan pada beberapa bidang, antara lain:

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.

2. Hiburan

Dunia hiburan membutuhkan *Augmented Reality* sebagai penunjang efek efek yang akan dibutuhkan oleh hiburan tersebut. Sebagai contoh, ketika seseorang wartawan memperkirakan ramalan cuaca, dia berada di depan layar hijau atau biru, kemudian dengan adanya teknologi *Augmented Reality*, layar hijau atau biru tersebut dirubah

menjadi gambar animasi tentang cuaca tersebut, sehingga seolah-olah wartawan tersebut, ada di dalam animasi tersebut.

3. Arsitektur

Arsitektur menggunakan *blue print* untuk menampilkan *Augmented Reality* berupa bangunan secara *virtual*, sehingga mempercepat pekerjaan dibidang arsitektur, dan dapat memperkirakan kelemahan-kelemahan yang ada pada bangunan.

4. Kedokteran

Dalam ilmu kedokteran teknologi pencitraan sangat dibutuhkan, seperti misalnya, untuk simulasi operasi, simulasi pembuatan vaksin virus, dan terkadang memberikan instruksi yang dapat memaksimalkan proses operasi *realtime*. Untuk itu, bidang kedokteran menerapkan *Augmented Reality* pada visualisasi penelitian mereka.

5. Militer

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

6. Consumer Design

Virtual reality telah digunakan dalam mempromsikan produk. Sebagai contoh, seorang pengembang menggunakan brosur *virtual* untuk memberikan informasi yang lengkap secara 3D, sehingga pelanggan dapat mengetahui secara jelas, produk yang ditawarkan. (Andriyadi,2011).

2.2.1 Sejarah *Augmented Reality*

Sejarah teknologi *Augmented Reality* dimulai sejak tahun 1957 sampai 1962, seorang peneliti dan sinematografi yang bernama Morton Heiling menciptakan dan mempatenkan penemuannya yaitu sebuah simulator yang dilengkapi dengan visual, getaran dan aroma yang kemudian dikenal dengan nama sensorama. sinematografi adalah sebuah teknik gabungan dari seni dan teknologi dari fotografi yang melibatkan sejumlah skema dan pengaturan konfigurasi dalam pengambilan gambar yang tepat. Pada tahun 1966, Ivan

Sutherland menemukan teknologi *Head-Mounted Display*, yang di akui sebagai jendela menuju dunia *virtual* (Anggi Andriyadi, 2010).

Tahun 1975 seorang ilmuwan Myron Kruger mengembangkan sebuah teknologi *Video place* yang memungkinkan pengguna dapat berinteraksi dengan objek *virtual* untuk pertama kalinya. Tahun 1989, Jaron Lanier menciptakan bisnis komersial pertama di dunia maya dengan teknologi *Virtual Reality*. Tahun 1992, Jaron mengembangkan teknologi *Augmented Reality* untuk melakukan simulasi perbaikan pada sistem pesawat jenis boeing. Pada tahun yang sama, LB Rosenberg mengembangkan salah satu sistem pada *Augmented Reality*, yang kemudian disebut dengan *Virtual Fixtures* dan kemudian digunakan oleh Angkatan Udara Amerika Serikat, yaitu Armstong Labs. Tahun 1992 juga, Blair MacIntyre, Steven Feiner dan Doree Seligmann mengembangkan dan memperkenalkan Prototype *Augmented Reality* yaitu *Major Paper* (Ramadar. 2014).

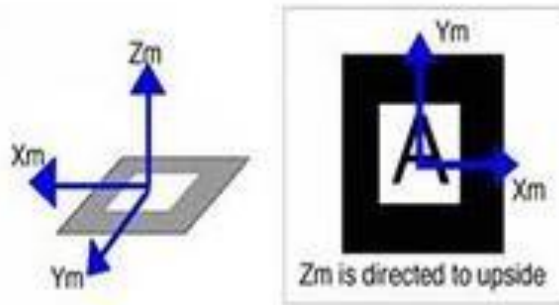
Pada tahun 1999, Hirokazu Kato mengembangkan ArToolkit di HITLab dan didemonstrasikan di SIGGRAPH, pada tahun 2000. Pada tahun 2008, Wikitude *Augmented Reality* Travel Guide memperkenalkan Android G1 *Telephone* yang berteknologi *Augmented Reality*. Tahun 2009, Saqoosha memperkenalkan FLARToolkit yang merupakan perkembangan dari ArToolkit. FLARToolkit memungkinkan kita memasang teknologi AR disebuah *website*, karena *output* yang dihasilkan FLARToolkit berbentuk *Flash*. Ditahun yang sama, Wikitude Drive meluncurkan system navigasi berteknologi AR di *Platform* Android Tahun 2010, Acrossair menggunakan teknologi *Augmented Reality* pada iPhone 3Gs (Sutoyo, 2009).

2.2.2 Metode Augmented Reality

Ada dua metode yang saat ini digunakan pada *Augmented Reality* yaitu: 1. *Marker Augmented Reality (Marker Based Tracking)*

Marker ini menggunakan ilustrasi yang berwarna hitam dan putih berbentuk persegi dengan garis hitam yang tebal dengan *background* berwarna putih (Haller, Michael, Mark Billingham, Bruce Thomas, 2009). Komputer akan mengetahui posisi dan orientasi *marker* dan membuat dunia *virtual* 3D pada titik (0, 0, 0) dengan tiga sumbunya

yaitu X, Y, dan Z. *Marker Based Tracking* dikembangkan sejak tahun 1980-an dan pada tahun 1990-an mulai dikenal oleh masyarakat dan dikembangkan untuk *Augmented Reality*.



Gambar 2.1 *Marker Based Tracking*

Sumber: Anggi Ariyadi (2010)

2. *Markerless Augmented Reality*

Metode *Markerless Augmented Reality* ini merupakan metode dari teknologi *Augmented Reality* yang pada saat ini sangat berkembang, dengan metode ini pengguna tidak perlu menggunakan *marker* yang memperlihatkan elemen yang berbentuk digital, dengan *tool* yang disediakan dari Qualcomm untuk perkembangan teknologi *Augmented Reality* yang berbasis *mobile device* ini Greader memudahkan para *developer* untuk membuat aplikasi *markerless* (Qualcomm, 2012).

Sekarang banyak yang sudah dikembangkan dan diteliti oleh beberapa perusahaan dari teknologi *Augmented Reality* ini dan yang terbesar di dunia seperti perusahaan Immersion dan Qualcomm, mereka membuat beberapa macam dari teknik *Markerless Tracking* ini sebagai teknologi andalan, contohnya Motion Tracking, 3D Object Tracking, selain itu *Greader Face Tracking*, dan *GPS Based Tracking* (Steven, 2010).

a. *Face Tracking*

Dengan menggunakan algoritma yang dikembangkan oleh *Augmented Reality*, komputer dapat mengenali wajah manusia secara umum dengan cara mengenali posisi mata, hidung, dan mulut manusia, kemudian akan mengabaikan objek-objek lain di sekitarnya seperti pohon, rumah, dan benda-benda lainnya. Teknik ini pernah

digunakan di Indonesia pada Pekan Raya Jakarta 2010 dan *Toy Story 3 Event* (Widiansyah, Firman, 2014).

b. *3D Object Tracking*

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

c. *Motion Tracking*

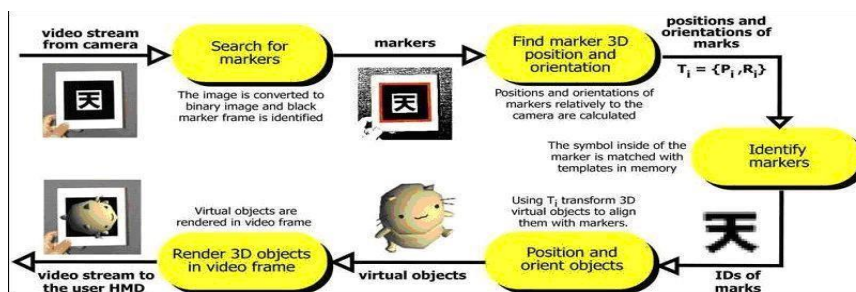
Komputer dapat menangkap gerakan, *Motion Tracking* telah mulai digunakan secara ekstensif untuk memproduksi film-film yang mencoba mensimulasikan gerakan. Contohnya pada film Avatar, dimana James Cameron menggunakan teknik ini untuk membuat film tersebut dan menggunakan secara *realtime* (Fernando, 2013).

d. *GPS Based Tracking*

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

2.2.3 Cara Kerja Augmented Reality

Cara kerja *Augmented Reality* menurut Ronald T. Azuma dapat dilihat pada gambar diagram dibawah ini.



Gambar 2.2 Diagram Alur Kerja *Augmented Reality*

(Ronald T. Azuma,1997)

Kamera akan mencari lokasi objek. Pada tahap ini gambar diterjemahkan menjadi gambar biner dan marker berwarna hitam diidentifikasi oleh sistem aplikasi. Kemudian mencari posisi dan orientasi marker dimana komputer melakukan perhitungan terhadap posisi dan orientasi berbasis data yang diterima kamera. Kemudian marker tersebut diidentifikasi untuk mengkalkulasi dimana meletakkan obyek 3D.

Augmented Reality dapat ditampilkan pada berbagai perangkat seperti kacamata, *smartphone* dan sebagainya. Agar perangkat dapat berfungsi dengan baik sejumlah data tertentu dapat berupa video, gambar, animasi, teks dan model 3D perlu digunakan, sehingga pengguna dapat merasakan berinteraksi dengan *Augmented Reality*. Berikut beberapa komponen yang digunakan pada AR :

- Proyeksi

Komponen ini mengacu pada proyektor yang lebih kecil dari yang biasa ada pada headset *Augmented Reality*, yang mengambil informasi dari sensor dan memproyeksikan konten yang terkomputerisasi ke permukaan untuk dilihat. For your information, sebenarnya, pemanfaatan proyeksi di *Augmented Reality* belum sepenuhnya dirancang untuk dapat digunakan dalam barang atau layanan komersial.

- Kamera dan Sensor

Kamera dan sensor digunakan untuk mengumpulkan informasi kolaborasi pengguna dan mengirimkannya untuk diproses. Kamera pada *smartphone* memiliki kemampuan untuk memeriksa lingkungan dan dengan data tersebut, akan menghasilkan model 3D.

2.3 Unity

Unity adalah sebuah *game multi platform* yang dapat digunakan pada PC, Mac, Wii, iPhone, iPad, Android dan *browser*. Banyak hal yang bisa dilakukan dengan unity, antara lain fitur *audio reverb zone, particle effect, sky box, rendering, lighting, sound effect*, dan *physic game* (Sudarmilah, E., dkk, 2013). Unity 3D adalah sebuah *game engine* yang berbasis *cross-platform* yang sangat *user friendly*. Unity dapat digunakan untuk membuat sebuah *game* yang bisa digunakan pada perangkat komputer, web, ponsel pintar android,

iOS, PS3, Wii dan X-BOX. Grafis pada unity dibuat dengan grafis tingkat tinggi untuk OpenGL dan directX.

- Refleksi

Beberapa gadget AR memiliki cermin untuk membantu mata manusia melihat gambar virtual. Beberapa darinya memiliki variasi cermin kecil yang ditebuk dan beberapa lagi memiliki cermin sisi ganda untuk memantulkan cahaya ke kamera dan mata pengguna. Tujuan dari cara refleksi tersebut adalah untuk memainkan pengaturan gambar yang tepat.

Unity adalah mesin pengembangan terintegrasi yang menyediakan *outof-the-box* fungsionalitas untuk membuat *game*, konten 3D interaktif, arsitektur bangunan dan memainkan simulasi. Unity bisa digunakan untuk membuat *game* 2D atau 3D, *game* FPS, *game* PC dan *game online*, dengan dukungan konversi mobile android, Iphone, Blackberry, Windows, Linux, Flash dan Webplayer.



Gambar 2.3 Logo Unity

(Sumber : Johansen, 2011)

Unity 3D dibuat dengan menggunakan bahasa pemrograman C++, Unity 3D mendukung bahasa pemrograman lain seperti JavaScript, C# dan Boo, Unity memiliki kemiripan dengan *game engine* lainnya seperti, *Blender game engine*, *Virtools*, *Game studio*, kelebihan dari Unity 3D, Unity dapat dioperasikan pada *platform* Windows dan Mac Os dan dapat menghasilkan *game* untuk Windows, Mac, Linux, Wii, iPad, iPhone, google Android dan juga browser. *Game* Unity 3D juga mendukung dalam pembuatan *game* untuk *console game* Xbox 360 dan PlayStation. (Yulianto Bambang, 2012). Pada Unity 3D terdapat beberapa hal penting untuk membuat atau membangun suatu karya, diantaranya yaitu:

1. *Project*

Project merupakan kumpulan dari komponen-komponen yang dikemas menjadi satu dalam sebuah *software* agar bisa di *build* menjadi sebuah aplikasi. Pada Unity 3D, *project* berisi identitas aplikasi yang meliputi nama *project*, *platform building*. Kemudian *package* apa saja yang akan digunakan, satu atau beberapa *scene* aplikasi, *asset*, dan lain-lain.

2. *Scene*

Scene dapat disebut juga dengan layar atau tempat untuk membuat layar aplikasi. *Scene* dapat dianalogikan sebagai level permainan, meskipun tidak selamanya *scene* adalah level permainan. Misal, level 1 anda letakkan pada *scene* 1, level 2 pada *scene* 2 dan seterusnya. Namun *scene* tidak selamanya berupa level, bisa jadi lebih dari satu level anda letakkan dalam satu *scene*. Game menu biasanya juga diletakkan pada satu *scene* tersendiri. Suatu *scene* dapat berisi beberapa game object. Antara satu *scene* dengan *scene* lainnya bisa memiliki game object yang berbeda.

3. *Asset* dan *Package*

Asset dan *package* adalah mirip, suatu *asset* dapat terdiri dari beberapa *package*. *Asset* atau *package* adalah sekumpulan objek yang disimpan. Objek dapat berupa *game object*, *terrain*, dan lain sebagainya. Dengan adanya *asset* dan *package* tidak perlu kesulitan dalam membuat objek lagi jika anda telah membuatnya sebelumnya, anda dapat mengimportnya dari *project* lama anda.

4. Vuforia SDK

Vuforia adalah *Augmented Reality Software Development Kit* (SDK) untuk perangkat bergerak yang memungkinkan pembuatan aplikasi *Augmented Reality*. Vuforia menggunakan teknologi *computer vision* untuk mengenali dan melacak *marker* atau *image target* dan objek 3D sederhana , seperti kotak , secara *real-time*.

2.4 Vuforia SDK

Vuforia adalah *library* pengembangan aplikasi *augmented reality* yang dikembangkan oleh *vendor* semi konduktor Qualcomm asal California, Amerika Serikat. *Library* vuforia memungkinkan *de* *r* mengembangkan sebuah pengalaman 3D yang interaktif. Vuforia *library* menawarkan keuntungan antara lain :

- a. *Computer vision* teknologi untuk menyelaraskan *marker* dengan objek yang berbentuk 3D.

- b. Didukung oleh berbagai *development tools*, seperti Unity 3D dan Xcode.
- c. Bebas royalty, baik pengembangan atau penjualan (Putra, dkk, 2012).

Vuforia adalah *Augmented Reality Software Development Kit* (SDK) untuk perangkat *mobile* yang memungkinkan pembuatan aplikasi *Augmented Reality*. SDK Vuforia tersedia digabungkan dengan Unity yang bernama *Vuforia Augmented Reality Extension for Unity*. Vuforia merupakan SDK yang disediakan oleh Qualcomm untuk membantu para *developer* membuat aplikasi *Augmented Reality* di ponsel (iOS, Android). SDK Vuforia sudah sukses dipakai di beberapa aplikasi-aplikasi *mobile* untuk kedua platform tersebut (Nugroho dan Pramono, 2017).

Vuforia memberikan cara berinteraksi dengan memanfaatkan kamera *mobile phone* untuk digunakan sebagai perangkat masukan, sebagai mata elektronik yang mengenali penanda tertentu. Dengan kata lain Vuforia adalah SDK untuk *computer vision based Augmented Reality*. Jenis aplikasi AR yang lain adalah *GPS-based AR*. Vuforia SCK memiliki banyak fitur yang berhubungan dengan *Augmented Reality* yaitu :

1. Perangkat Lunak dapat memindai dan mengenali *QR code*.
2. Vuforia SDK dapat memindai dan mengenali gambar dan teks, dan memiliki keterbatasan kemampuan memindainya seperti sudut atau bentuk yang melengkung sulit untuk dikenali.
3. Vuforia SDK mampu mengidentifikasi permukaan objek.
4. Vuforia SDK dapat mengenali target benda silinder kecil sebagai *marker*.
5. Vuforia SDK mampu mengenali kata dalam bahasa inggris tertentu.
6. Perangkat lunak ini dapat menggunakan fitur penyimpanan berbasis *cloud*.
7. Vuforia SDK memiliki *build-in virtual* manajemen tombol *virtual*, fitur ini sangat menarik untuk menambahkan *virtual* interaksi antara pengguna dan perangkat lunak.

2.5 Google SketchUp

Google *SketchUp* adalah salah satu program aplikasi gratis dari Google yang berbasis 3D. Aplikasi ini dapat digunakan untuk membantu menyempurnakan sebuah gagasan atau ide desain kedalam model 3D. Sketchup dirancang untuk eksplorasi desain sehingga memiliki kemampuan kreasi yang sangat tinggi, mengamati, dan memodifikasi gagasan 3D dengan cepat dan mudah karena ditunjang oleh *tool-set*, yang digunakan untuk




mendesain secara mudah dan sederhana. Sketchup adalah perangkat lunak 3D yang dirancang untuk pembuat film, pengembang permainan, aplikasi dan profesi lain yang terkait bidang 3D (Irham Fa'idh Faiztyan, 2015).

Google sketchup merupakan *software* untuk membuat, memodifikasi, dan mempertukarkan model 3D. Google sketchup mulai banyak digunakan karena memiliki *interface* yang menarik atau *simple*, mudah digunakan, didukung berbagai *open source plugin* yang mendukung kinerja Sketchup, *warehouse* atau gudang model 3D yang sangat lengkap dan terorganisir sehingga memudahkan pengguna dalam mencari model 3D dan tidak membutuhkan spesifikasi komputer yang tinggi.

Adapun *tools* yang terdapat di Google SketchUp beserta fungsinya yaitu :

Tabel 2.1 *Tools* Google SketchUp

NO.	Simbol	Nama	Fungsi
1.		<i>Rectangle, circle, arc</i>	Membentuk bentuk datar untuk proses dalam 3D.
2.		<i>Push / Pull</i>	Dasar untuk membuat 3D, setelah membuat bidang datar di bidang gambar.
3.		<i>Select</i>	memilih atau mengubah objek / mode saat menggunakan alat atau perintah lain.
4.		<i>Make Components</i>	Berbeda dengan Groups, objek objek tunggal digabungkan dengan <i>Make Components</i> .
5.		<i>Paint bucket</i>	mewarnai atau menempelkan material pada suatu objek.
6.		<i>Eraser</i>	Menghapus gambar atau bahan.







7.		<i>Line</i>	Menggambar garis lurus.
8.		<i>Circle</i>	Menggambar objek bulat.
9.		<i>Arc</i>	Menggambar setelah lingkaran.


Tabel 2.2 *Tools* Google SketchUp (Lanjutan 1)

10.		<i>Polygon</i>	Menggambar objek persegi 4,6,8 dan lain-lain.
11.		<i>Freehand</i>	Menggambar gratis atau bebas.
12.		<i>Move</i>	Memindahkan objek.
13.		<i>Rotate</i>	Memutar objek.
14.		<i>Follow me</i>	Memindahkan objek yang disesuaikan
15.		<i>Scale</i>	Mengubah ukuran objek kecil pada skala ke kanan.
16.		<i>Offset</i>	Gandakan sederet objek yang disesuaikan
17.		<i>Tape Measure Tool</i>	Mengukur objek
18.		<i>Dimension</i>	Memberikan dimensi pada objek
19.		<i>Protractor</i>	Mengukur sudut kemiringan desain dan membuat elmen garis.

20.		<i>Text</i>	Menyisipkan teks.
21.		<i>Axes</i>	Memindahkan atau mengubah arah sumbu gambar.
22.		<i>Teks 3D</i>	Membuat teks tiga dimensi.
23.		<i>Orbit</i>	Memutar tampilan objek.
24.		<i>Pan</i>	Memindahkan tampilan objek secara vertical dan horizontal.

Tabel 2.3 *Tools* Google SketchUp (Lanjutan 1)

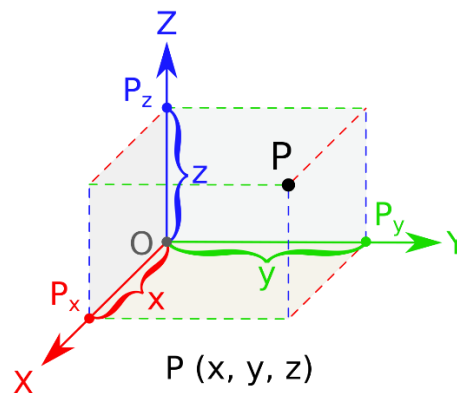
25.		<i>Zoom</i>	Memperbesar atau memperkecil <i>view</i> terhadap objek
26.		<i>Zoom Window</i>	Untuk memperbesar objek layar
27.		<i>Previous</i>	Batalkan pratinjau
28.		<i>Posisi kamera</i>	Posisikan kamera (tampilan Anda) pada ketinggian tertentu untuk memeriksa objek yang saling berhadapan atau melewati model.
29.		<i>Look Around</i>	Putar kamera (tampilan Anda) dari titik diam.
30.		<i>Walk</i>	Pergi melalui model.

31.		<i>Selection Plane</i>	Menggunakan efek bagian untuk menunjukkan geometri dalam model.
-----	---	------------------------	---

2.6 3 Dimensi

Menurut penjelasan Nana Sudjana (2011) pengertian media tiga dimensi adalah suatu alat peraga yang mempunyai panjang, lebar serta tinggi dan dapat diamati dari sudut pandang mana saja. Dimensi didefinisikan sebagai ukuran yang menyatakan keberadaan sesuatu seperti panjang, lebar dan tinggi.

Garis yang menghubungkan dua buah titik memiliki satu ukuran yaitu panjang disebut berdimensi satu. Sementara sebuah persegi panjang yang memiliki dua ukuran yaitu panjang dan lebar disebut berdimensi dua. Dan sesuatu dikatakan berdimensi tiga ketika ia memiliki tiga ukuran yaitu panjang, lebar dan tinggi atau kedalaman. Secara umum, pengertian tiga dimensi adalah bentuk dari benda yang memiliki panjang, lebar dan tinggi yang memiliki kapasitas yang disebut dengan *volume*. Sistem koordinat tiga dimensi memiliki tiga sumbu, yaitu x, y dan z.



Gambar 2.4m koordinat tiga dimensi

(Sumber : https://id.m.wikipedia.org/wiki/3_dimensi)

2.7 *Unified Modeling Language (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 (Nugroho, 2010). Berdasarkan pendapat yang telah dikemukakan maka dapat disimpulkan bahwa UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan mendokumentasikan dari sebuah sistem pengembangan perangkat lunak berbasis objek (*Object Oriented Programming*).

Menurut Henderi (2008:6), langkah-langkah penggunaan *Unified Modeling Language* (UML) sebagai berikut :

1. Buatlah daftar *business process* dari *level* tertinggi untuk mendefinisikan aktivitas dan proses yang mungkin muncul.
2. Petakan *use case* untuk setiap *business process* untuk mendefinisikan dengan tepat fungsional yang harus disediakan oleh system, kemudian perhalus *use case diagram* dan lengkapi dengan *requirement*, *constraints* dan catatan-catatan lain.
3. Buatlah *deployment diagram* secara kasar untuk mendefinisikan arsitektur fisik sistem.
4. Definisikan *requirement* lain non *fungsional*, *security* dan sebagainya yang juga harus disediakan oleh sistem.
5. Berdasarkan *use case diagram*, mulailah membuat *activity diagram*.
6. Definisikan obyek-obyek *level atas package* atau *domain* dan buatlah *sequence* dan/atau *collaboration* untuk tiap alur *perjalanan*, jika sebuah *use case* memiliki kemungkinan alur normal dan *error*, buat lagi satu diagram untuk masing-masing alur.
7. Buatlah rancangan *user interface* model yang menyediakan antar muka bagi pengguna untuk menjalankan *skenario use case*.
8. Berdasarkan model-model yang sudah ada, buatlah *class diagram*. Setiap *package* atau *domian* dipecah menjadi hirarki *class* lengkap dengan atribut dan metodenya. Akan lebih baik jika untuk setiap *class* dibuat unit test untuk menguji fungsionalitas *class* dan interaksi dengan *class* lain.
9. Setelah *class diagram* dibuat, kita dapat melihat kemungkinan pengelompokkan *class* menjadi komponen-komponen karena itu buatlah *component diagram* pada tahap ini. Juga, definisikan test integrasi untuk setiap komponen meyakinkan ia bereaksi dengan baik.

10. Perhalus *deployment* diagram yang sudah dibuat. Detilkan kemampuan dan *requirement* piranti lunak, sistem operasi, jaringan dan sebagainya. Petakan komponen ke dalam *node*.
11. Mulailah membangun sistem. Ada dua pendekatan yang tepat digunakan : Pendekatan *use case* dengan menempatkan setiap *use case* kepada tim pengembang tertentu untuk mengembangkan unit kode yang lengkap dengan test dan pendekatan komponen yaitu menempatkan setiap komponen kepada tim pengembang tertentu.

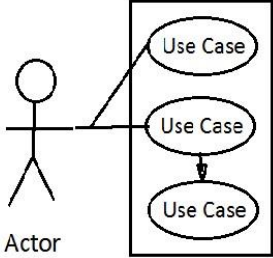

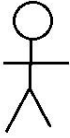
Menurut Nugroho (2010:10), Sesungguhnya tidak ada batasan yang tegas diantara berbagai konsep dan konstruksi dalam UML, tetapi untuk menyederhanakannya, kita membagi sejumlah besar konsep dan dalam UML menjadi beberapa *view*. Suatu *view* sendiri pada dasarnya merupakan sejumlah konstruksi pemodelan UML yang merepresentasikan suatu aspek tertentu dari sistem atau perangkat lunak yang sedang kita kembangkan. Pada peringkat paling atas, *view-view* sesungguhnya dapat dibagi menjadi tiga area utama, yaitu: klasifikasi struktural (*structural classification*), perilaku dinamis (*dynamic behaviour*), serta pengolahan atau manajemen model (*model management*).

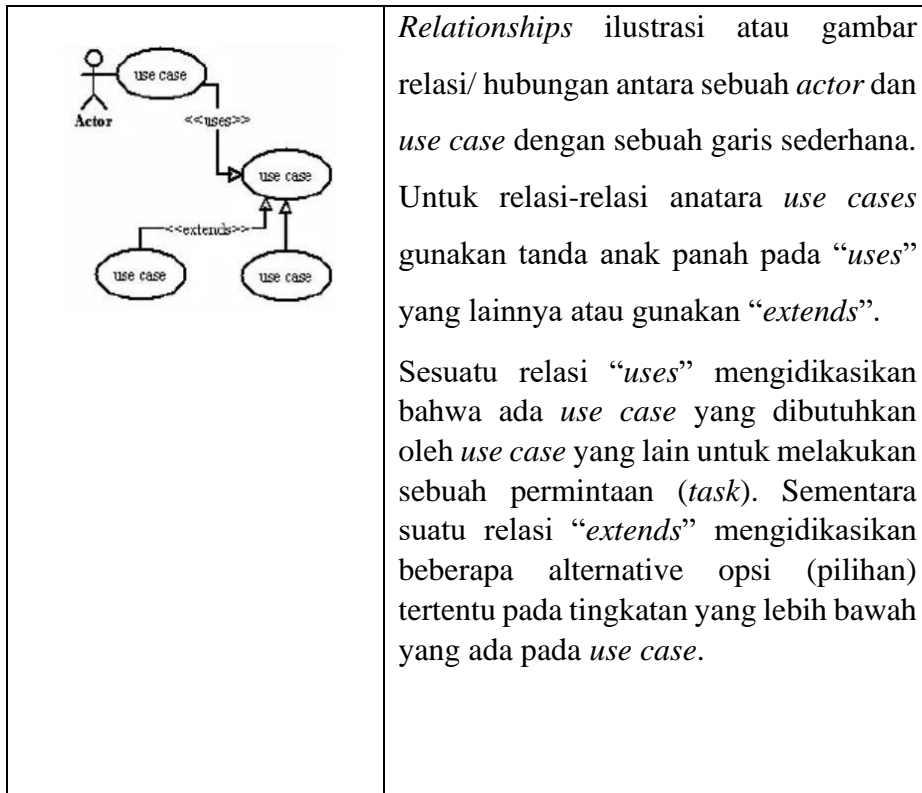
Menurut Henderi (2008:5) jenis diagram *Unified Modeling Language*) adalah sebagai berikut :

1. *Use Case Diagram* secara grafis menggambarkan interaksi antara 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 setiap interaksi.

Tabel 2.4 Simbol dan Notasi Dasar *Use Case Diagram*

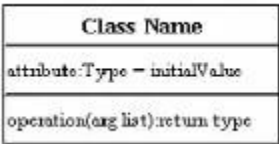
Simbol	Notasi Dasar <i>Use Case Diagram</i>

 <p>Actor</p>	<p>Gambar batasan (<i>boundries</i>) sebuah system menggunakan empat persegi panjang yang berisi <i>use case – use case</i>. Tempatkan <i>actor-actor</i> yang terlihat pada setiap <i>use case</i> pada bagian luar <i>boundries</i> system.</p>
	<p>Gambar <i>use case</i> menggunakan lingkaran berbentuk bulat telur (<i>ovals</i>) beri nama <i>ovals</i> tersebut dengan kata kerja (<i>verbs</i>) yang menggambarkan fungsi sistem</p>
 <p>Actor</p>	<p><i>Actor</i> adalah seseorang atau sesuatu yang harus berinteraksi dengan sistem atau sistem yang dibangun/ dikembangkan.</p>

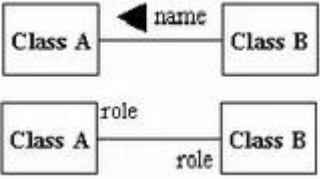


2. *Class Diagram* menggambarkan struktur object sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut.

Tabel 2.5 Simbol dan Notasi Dasar *Class Diagram*

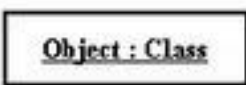
Simbol	Notasi Dasar <i>Class Diagram</i>
	<p>Classes dengan bentuk empat persegi panjang yang dibagi kedalam ruang ruang terpisah (<i>compartments</i>). Nama <i>class</i> ditempatkan pada bagian pertama</p>
	<p>(rata tengah, di-<i>bold</i>, dan huruf besar), daftar atribut diletakan pada bagian kedua, dan tuliskan oprasi-oprasi pada <i>class</i> dibagian ketiga.</p>

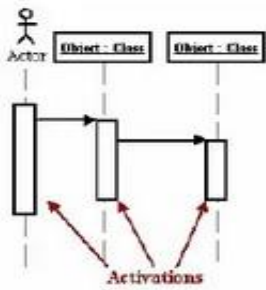
<div style="border: 2px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Active class</div>	<p><i>Active classes</i> adalah <i>class</i> yang memulai dan mengontrol aliran/ arus/ arah aktifitas. Sementara <i>passive class</i> menyimpan data dan melayani (<i>men-serve</i>) <i>class</i> yang lain. Gambarkan <i>active class</i> pada sebuah <i>bolder</i> tebal dibagian tengah.</p>
<div style="display: flex; align-items: center; gap: 20px;"> <div style="border: 1px solid black; padding: 5px;"> + <i>public</i> - <i>private</i> # <i>protected</i> </div> <div style="border: 1px solid black; padding: 5px;"> Class Name - attribute - attribute + operation + operation + operation </div> </div>	<p><i>Visibility</i> (“penglihatan”) untuk menandakan siapa yang bisa mengakses informasi-informasi yang diisi kedalam sebuah <i>class</i>. <i>Private visibiliy</i> artinya informasi yang ada dalam sebuah <i>class</i> disembunyikan/ dipartisi dari pihak luar. <i>Public visibility</i> mengizinkan semua <i>class</i> yang lainnya untuk melihat nilai informasi. <i>Protected visibility</i> mengizinkan <i>class-class</i> yang ada yang merupakan turunannya untuk mengakses informasi yang ada didalamnya karena mereka merupakan <i>class</i> turunan dari <i>class</i> induknya/ <i>inherited</i>.</p>

	<p><i>Associations</i> adalah representasi gambaran relasi statis diantara <i>classclass</i>. Tempatkan nama <i>associations</i> pada bagian atas, di, atau dibawah garis <i>associations</i>.</p> <p>Gunakan tanda anak panah yang berisi sebuah kata yang mengindikasikan relasi secara langsung.</p> <p>Letakan <i>role</i> (aturan/ ketentuan) pada bagian akhir <i>associations</i>. Aturan merepresentasikan arah bagi kedua kelas untuk saling berhubungan satu sama lain. Pada umumnya sebuah "nama" tidak digunakan untuk menunjukan role sebuah class.</p>
---	--

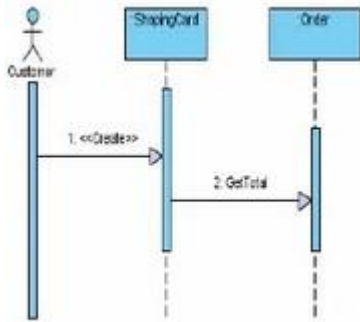
3. *Sequence Diagram* secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi.

Tabel 2.6 Simbol dan Notasi Dasar *Sequence Diagram*

Simbol	Notasi Dasar <i>Sequence Diagram</i>
	<p>Class roles menggambarkan jalan sebuah objek akan menunjukan reaksi (berkelakuan) dalam sebuah keadaan (<i>konteks</i>). Gunakan simbol objek UML untuk menggambarkan aturan-aturan <i>class</i>, tetapi bukan daftar <i>attribute attribute</i> objek.</p>



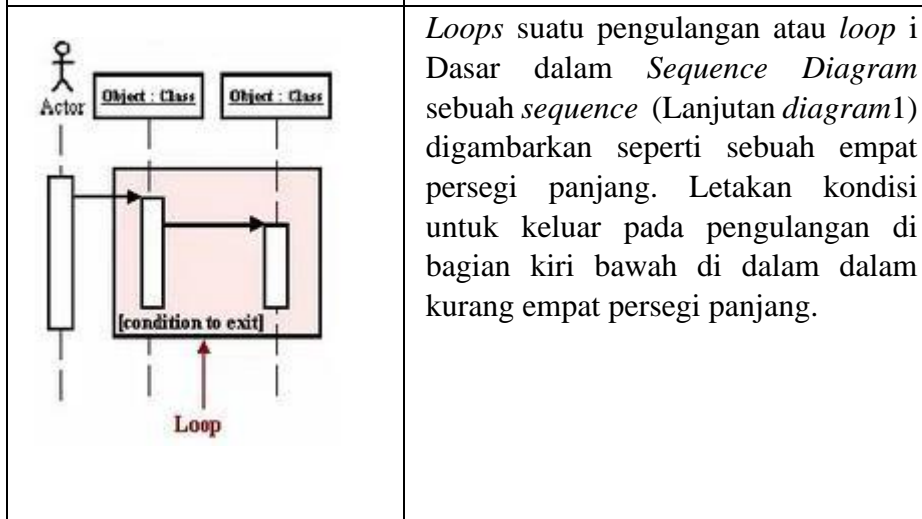
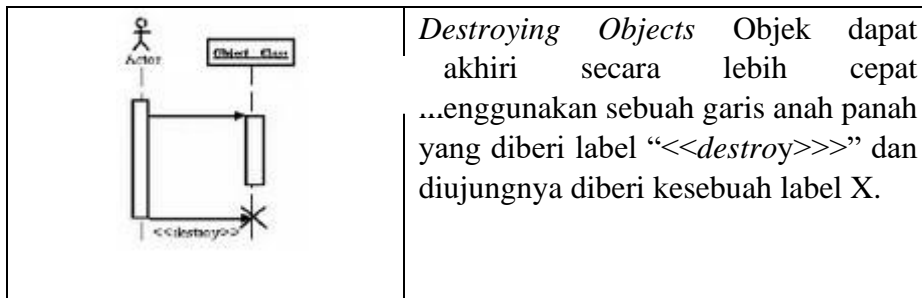
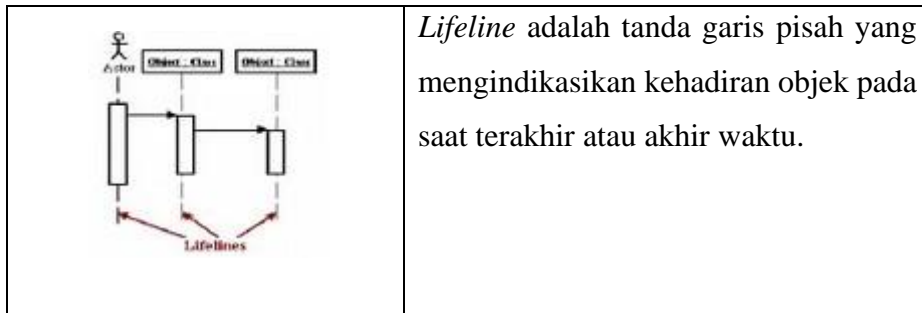
Activation boxes merepresentasikan waktu yang dibutuhkan oleh sebuah objek untuk melaksanakan sebuah tugas atau perintah secara lengkap.



Message adalah anak panah yang merepresentasikan komunikasi antara objek. *Message* berguna untuk mengirimkan perintah kepada *LifeLine*. *Message* bisa berasal dari *actor* kepada *LifeLine* atau dari *LifeLine* kepada *LifeLine* yang lain. Gunakan setengah garis anak panah untuk merepresentasikan pesan - pesan *asynchronous*. Pesan *asynchronous* dikirim dari sebuah objek yang tidak akan menunggu respon dari penerima sebelum melanjutkan perintahnya.

Arrow	Message type
	Simple
	Synchronous
	Asynchronous
	Balking
	Time out

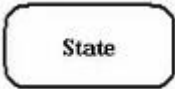
Berbagai tipe *message* untuk *sequence diagram* dan *collaboration diagram*.

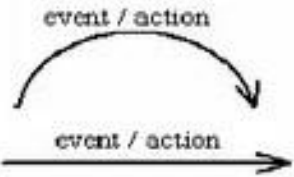



4. *State Chart Diagram* digunakan untuk memodelkan *behaviour* objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek berbagai keadaan yang dapat diasumsikan oleh objek dan *event-event* (kejadian) yang menyebabkan objek beralih dari satu *state* ke *state* yang lain.


Tabel 2.7 Simbol dan Notasi Dasar *Statechart Diagram*

Simbol	Notasi Dasar <i>Statechart Diagram</i>
--------	--

	<p><i>States</i> merepresentasikan keadaan “<i>life</i>” sebuah objek selama objek tersebut ada. State digambarkan dalam bentuk empat persegi panjang yang keempat sudutnya berbentuk bundar.</p>
---	---

	<p><i>Transition</i> suatu anak panah yang tebal merepresentasikan jalan antara state yang berlainan dari sebuah objek. Berikan nama <i>transisi</i> (peralihan) dengan kejadian yang menjadi pemicunya dan aksi yang dihasilkan olehnya.</p>
---	---


	<p><i>Initial Pseudo State</i> digunakan untuk memulai <i>Statechart Diagram</i>. <i>Inisitial Pseudo State</i> digambar menggunakan <i>arrow</i> yang pada bagian awalnya terdapat lingkaran berisi penuh <i>bold</i>.</p>
---	---

	<p><i>Final State</i> digunakan untuk mengakhiri diagram <i>statecahart</i>. <i>Final State</i> digambar menggunakan <i>arrow</i> yang diujungnya terdapat lingkaran yang berisi lingkaran berisi penuh <i>bold</i>.</p>
---	--

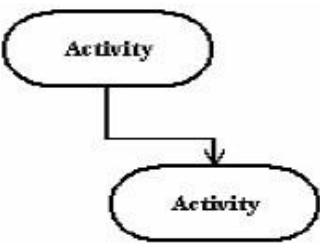
5. *Activity Diagram* secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan action yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari action tersebut.

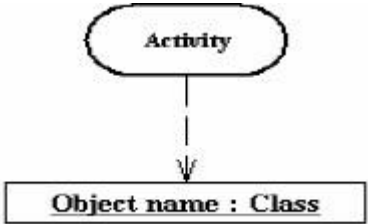
Tabel 2.8 Simbol dan Notasi Dasar *Activity Diagram*

Simbol	Notasi dasar <i>Activity Diagram</i>
--------	--------------------------------------

	<p><i>Action state</i> adalah representasi atau gambaran dari aksi yang tidak bisa diganggu oleh aksi yang berasal dari objek - objek. <i>Action state</i> digambarkan dalam bentuk empat</p>
---	---

	<p>persegi panjang yang pada sudut-sudutnya melingkar.</p>
--	--

	<p><i>Action Flow</i> digambarkan dalam bentuk anak panah yang mengilustrasikan relasi antara <i>action</i> pada <i>state</i>.</p>
---	--

	<p><i>Object flow</i> menunjuk kepada kegiatan penciptaan dan memodifikasi oleh objek melalui <i>activities</i>. Sebuah tanda panah objek <i>flow</i> dari suatu aksi kepada suatu objek berarti bahwa aksi tersebut membuat atau mempengaruhi objek tersebut. Sementara suatu tanda panah dari objek kepada suatu aksi mengindikasikan bahwa aksi <i>state</i> tersebut menggunakan objek.</p>
---	---

2.8 Metode Pengembangan Sistem

Metode pengembangan sistem adalah suatu kerangka kerja yang digunakan untuk menstrukturkan dan mengendalikan proses pengembangan suatu system informasi. Dalam

penelitian ini peneliti menggunakan metode pengembangan sistem *Interactive System Multimedia Design and Development (IMSDD)*.

2.8.1 Interactive System Multimedia Design and Development (IMSDD)

Interactive System Multimedia Design and Development (IMSDD), yaitu system multimedia interaktif yang membutuhkan perencanaan yang sangat hati-hati pada struktur navigasi dan pendekatan dalam pembuatan interaktifitas menggunakan metafora desain yang tepat (Dastbaz, 2003).

Tahapan-tahapan dalam IM menurut Dastbaz (2003, p130-132) adalah sebagai berikut :

1. Kebutuhan Sistem (*System Requirment*)

Pada tahap ini suatu definisi umum dari IMS (*Interactive Multimedia System*) dan lingkungannya ditentukan. Tahap ini memiliki fungsi-fungsi kunci sebagai berikut :

- a. Untuk menyediakan definisi sistem seperti pembuatan *outline* mengenai tujuan dan sasaran dari sistem yang akan dibuat.
- b. Untuk memastikan siapakah pengguna dari sistem yang akan dibuat dan jika ada kebutuhan spesifik lain yang perlu dipertimbangkan.
- c. Evaluasi *hardware, software, dan authoring tools* yang dibutuhkan lalu pilih secara tepat.
- d. Pertimbangkan secara tepat *delivery platform* yang dibutuhkan oleh sistem.

2. Pertimbangan Desain (*Design Considerations*)

Tujuan dari langkah ini adalah untuk menggambarkan secara jelas panduan tentang detail desain. Langkah ini mencakup :

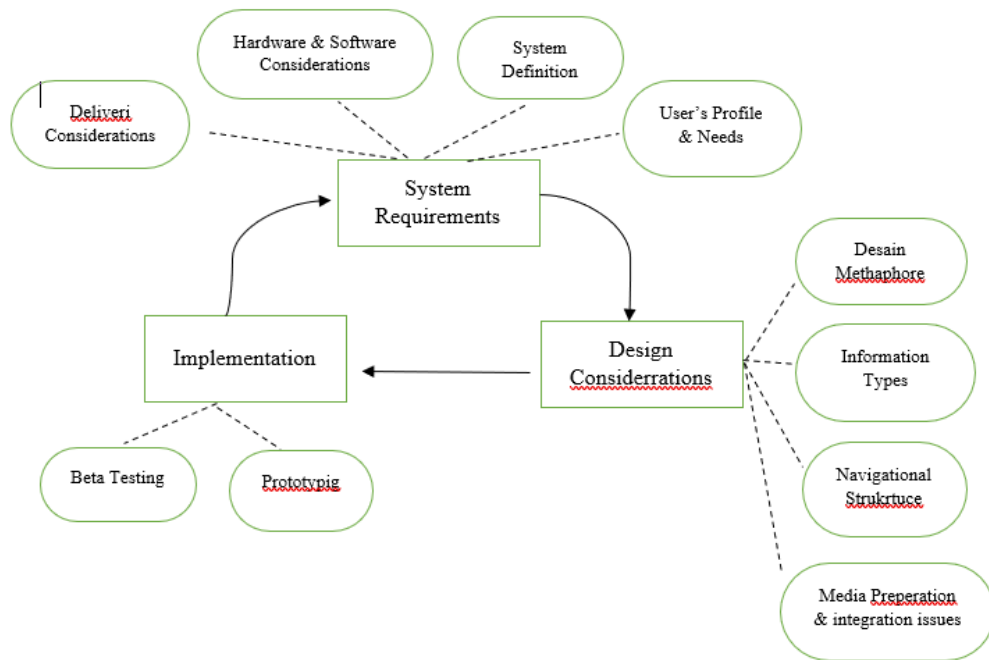
- a. Metafora Desain, Memilih sebuah model nyata untuk digunakan sebagai solusi kunci desain *interface* bagi sistem contohnya film, buku, game, dll.
- b. Format dan tipe informasi, yaitu untuk mendefinisikan tipe informasi yang dibutuhkan untuk diintegrasikan ke dalam sistem tersebut, seperti teks, grafik, suara, vidio, dan animasi.

- c. Struktur navigasi, yaitu untuk menyatakan suatu strategi navigasi yang jelas, termasuk fitur dan struktur *link* yang akan menghindari masalah masalah yang berkaitan dengan sistem hypermedia, misalnya disorientasi.
- d. Sistem Kontrol, menjelaskan tipe dan fitur-fitur dari *control* dan peralatan yang dibutuhkan oleh sistem.

3. Implementasi (*Implementation*)

Saat fitur desain telah ditentukan maka tahap implementasi dari sistem dimulai menggunakan multimedia *authoring tools*. Tahap implementasi terdiri atas :

- a. Membuat Prototipe dari Sistem.
- b. Melakukan beta *test* terhadap prototype untuk mencari masalah yang mungkin dari *control* atau desain.



Gambar 2.5 Siklus IMSDD

2.9 Metode Pengujian

Pengujian perangkat lunak bertujuan untuk memastikan, bahwa perangkat lunak yang telah dikembangkan sudah berjalan sesuai dengan fungsional yang diharapkan pengguna akhir. Pengembangan perangkat lunak melakukan sesi khusus untuk menguji perangkat

lunak agar eror dapat di deteksi sejak awal. Pengujian perangkat lunak ini menjamin kualitas perangkat lunak yang merupakan bagian dari daur hidup pengembangan perangkat lunak (M. Sidi Mustaqbal, 2015).

2.9.1 *Blackbox Testing*

Menurut Black (2009 :3), Tester menggunakan *behavioral test* (disebut juga *Black-Box Tests*), sering digunakan untuk menemukan *bug* dalam *high level operations*, pada tingkatan fitur, *profil* operasional dan skenario *customer*. *Tester* dapat membuat pengujian fungsional *black box* berdasarkan pada apa yang harus sistem lakukan. *Behavioral testing* melibatkan pemahaman rinci mengenai *domain* aplikasi, masalah bisnis yang dipecahkan oleh sistem dan misi yang dilakukan sistem. *Behaviora test* paling baik dilakukan oleh penguji yang memahami desain sistem, setidaknya pada tingkat yang tinggi sehingga mereka dapat secara efektif menemukan bug umum untuk jenis desain.

Menurut Mustaqbal, dkk (2015, 34) *Blackbox testing* berfokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi input dan melakukan pengetesan pada fungsional program. *Blackbox testing* adalah tahapan yang digunakan untuk menguji kelancaran program yang telah dibuat. Pengujian ini dilakukan agar tidak terjadi kesalahan alur program yang telah dibuat. Metode pengujian *Blackbox Testing* dapat menemukan beberapa kesalahan seperti :

1. Fungsi-fungsi yang tidak sesuai atau hilang
2. Kesalahan dari kinerja
3. Kesalahan yang terjadi pada *interface*
4. Kesalahan dalam struktur data atau akses database *eksternal*
5. Kesalahan dari inisialisasi dan terminasi

Keuntungan penggunaan metode *Blackbox Testing* adalah Penguji tidak perlu melihat kode program secara detail, dapat digunakan untuk menilai konsistensi program, programmer dan tes keduanya saling bergantung satu sama lain. Kekurangan *Blackbox Testing* bila spesifikasi program yang dibuat kurang jelas dan ringkas, maka akan sulit membuat dokumentasi setepat mungkin, kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer.

