

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini ide dan proses pencarian penyelesaian permasalahan aplikasi yang dibuat didapat dari penelitian yang pernah dibuat sebelumnya sebagai acuan dan perbandingan. Penelitian sebelumnya terdapat materi yang hamper sama dengan penelitian ini.

Pembuatan aplikasi 3D *viewer mobile* yang bertujuan untuk menggambarkan proses perobekan Bendera Belanda di hotel Majapahit pada tahun 1940-an dengan teknologi *virtual reality*. Dimana berupa aplikasi *virtual reality* yang menggambarkan tentang perobekan bendera Belanda di hotel Majapahit tahun 1940. Membangun aplikasi *virtual reality* tentang bangunan hotel Majapahit dengan aplikasi *Unity 3D* dan *Autodesk 3Ds Max* sedangkan pembuatan aplikasi bangunan Universitas Sahid Surakarta dengan *Unity 3D* dan *Google SketchUp* (Berta Sihite dkk., 2013).

Pembuatan aplikasi teknologi *virtual realty* bagi pelestarian bangunan arsitektur yang bertujuan untuk melestarikan bangunan arsitektur bersejarah dengan teknologi *virtual reality*. Dimana berupa aplikasi *virtual reality* bangunan arsitektur bersejarah yang di visualkan melalui *Head Mounted Displays (HMD)*, *BOOM displays*, papan meja *virtual* dan *Cave Automatic Virtual Environment (CAVE)* dengan kacamata 3D. Membuat aplikasi bangunan arsitektur bersejarah dengan memanfaatkan berbagai teknologi sedangkan pembuatan aplikasi gedung Universitas Sahid Surakarta dengan *unity3D* dan *Google SketchUp* (Yudi Nugraha B, 2014).

Pembuatan 3D *Virtual Reality* : *Virtual UNS* menggunakan *Unity 3D Game Engine* Berbasis *Android* yang bertujuan untuk memberikan gambaran visual terhadap kampus sehingga menambah wawasan mengenai UNS, serta sebagai sarana hiburan yang interaktif. Membuat *virtual reality* tentang lingkungan UNS menggunakan *Unity 3D*, *Google SketchUp* Dan *Autodesk 3Ds Max* sedangkan

Pembuatan *virtual reality* tentang Universitas Sahid Surakarta menggunakan Unity3D dan Google Sketchup (Adam Firmansyah, 2015).

2.2 Kerangka Pemikiran

Penyusunan Tugas Akhir aplikasi visualisasi gedung Universitas Sahid Surakarta menggunakan teknologi *virtual reality* ini disusun melalui beberapa tahapan dalam suatu kerangka pemikiran, seperti pada Gambar 2.1. Adapun kerangka berpikir yang dijalankan adalah sebagai berikut :

1) Latar Belakang Masalah

Latar belakang masalah merupakan tahapan yang digunakan untuk melihat dasar permasalahan dari Universitas Sahid yang mana mahasiswa, dosen, *staff* dan tamu kesulitan untuk menuju keruangan yang dituju karena begitu banyak ruangan di Universitas Sahid Surakarta .

2) Rumusan Masalah

Hasil kajian dari latar belakang masalah yang dijadikan dasar untuk mengambil suatu simpulan bagaimana merancang dan membuat visualisasi dari gedung Universitas Sahid Surakarta.

3) Menguasai Aplikasi Yang Digunakan

Menguasai *Unity 3D*, *Google SketchUp* maupun *Adobe Illustrator* yang dibutuhkan untuk pengerjaan visualisasi gedung Universitas Sahid Surakarta.

4) Pengumpulan Data

Pengumpulan data digunakan untuk mendukung pengerjaan visualisasi Universitas Sahid Surakarta yang dilakukan dengan wawancara ataupun observasi, serta mencari sumber landasan teori seperti buku, jurnal dan sumber lainnya yang relevan dan terpercaya.

5) Analisis Dan Perancangan Aplikasi

Analisis dilakukan untuk mengetahui apakah sebuah aplikasi berjalan dengan baik atau tidak. Perancangan merupakan bangunan awal atau pondasi untuk membangun suatu aplikasi yang akan dikembangkan.

6) Implementasi

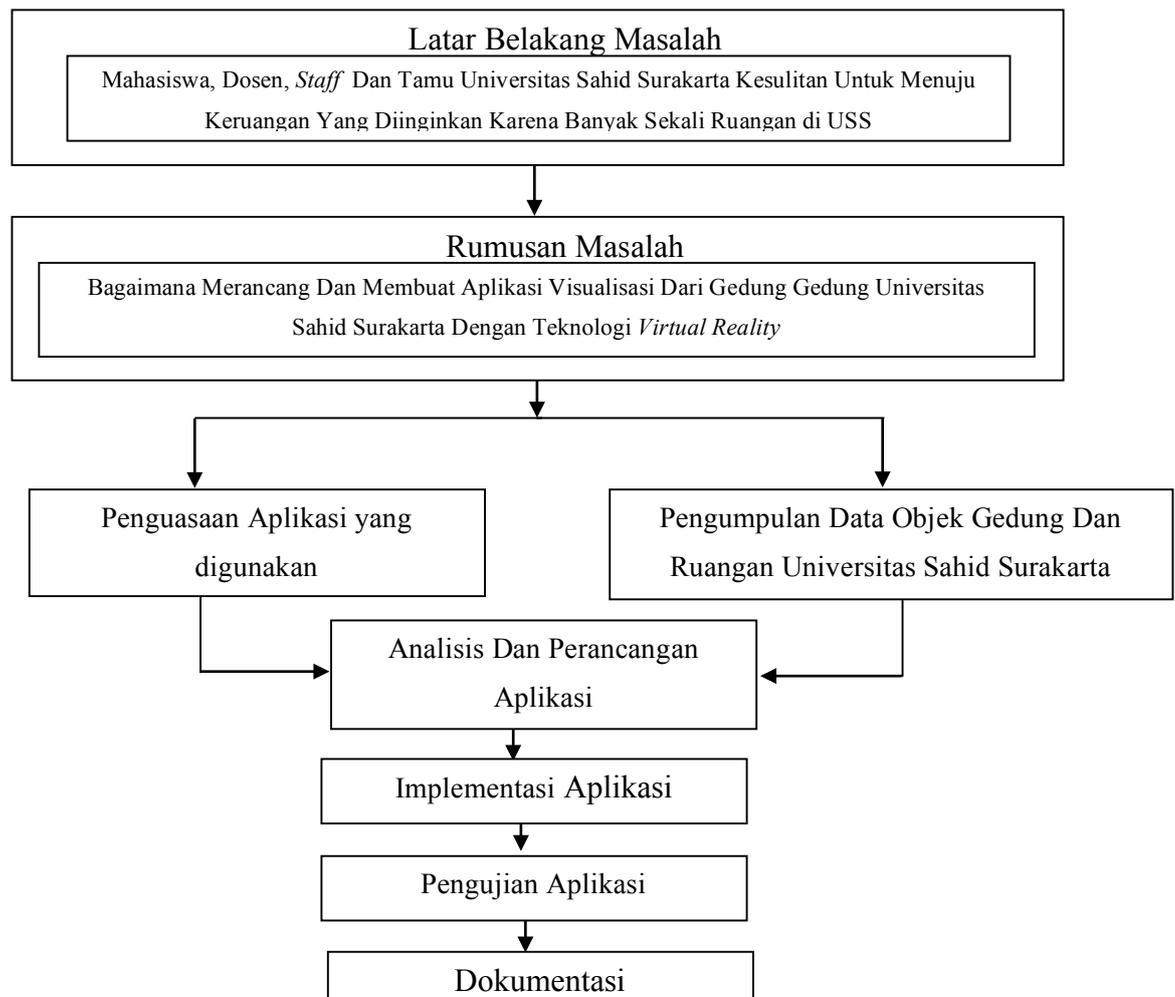
Aplikasi visualisasi Universitas Sahid Surakarta yang telah dikembangkan kemudian di implementasikan dengan teknologi yang *virtual reality*.

7) Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk memastikan apakah aplikasi yang dibangun berjalan dengan baik atau memiliki banyak *bug*.

8) Dokumentasi

Dokumentasi merupakan pengukuhan tugas akhir visualisasi Universitas Sahid Surakarta yang mana untuk tahap awal berupa proposal untuk pendukung pengerjaan aplikasi dan laporan Tugas Akhir untuk pengukuhan aplikasi yang telah selesai dibuat.



Gambar 2.1. Bagan Kerangka Pemikiran

2.3 Landasan Teori Pendukung

Beberapa istilah yang berhubungan dengan pembuatan Aplikasi Visualisasi Gedung Universitas Sahid Surakarta Menggunakan Teknologi *Virtual Reality* :

2.3.1 Multimedia

Multimedia adalah media yang menggabungkan dua unsur atau lebih media yang terdiri dari teks, grafik, gambar, audio, video dan animasi secara terintegrasi (M. Yanyan H dan Irawan A, 2013).

Multimedia terbagi menjadi dua kategori yaitu :

- 1) Multimedia Linier adalah suatu multimedia yang tidak dilengkapi dengan alat pengontrol apapun yang dapat dioperasikan oleh pengguna. Multimedia linier berjalan secara sekuensial. Contoh dari multimedia linier adalah TV dan film.
- 2) Multimedia Interaktif adalah suatu multimedia yang dilengkapi dengan alat pengontrol yang dapat dioperasikan oleh pengguna sehingga pengguna dapat memilih apa yang dikehendaki untuk proses selanjutnya. Contoh multimedia interaktif adalah multimedia pembelajaran interaktif, aplikasi game dan lain-lain.

2.3.2 Aplikasi

Aplikasi merupakan program yang dapat langsung digunakan untuk menjalankan perintah dari pengguna dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi, aplikasi memiliki arti pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berjalan pada komputasi yang diinginkan.

Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya. Aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user (Hasan Abdurahman dan Asep R.R, 2014).

2.3.3 Visualisasi Informasi

Visualisasi Informasi adalah suatu metode penggunaan komputer untuk menemukan metode terbaik dalam menampilkan data untuk mengingat informasi dengan cara penerimaan alami manusia serta memberikan cara untuk melihat data yang sulit dilihat dengan pemikiran sehingga peneliti bisa mengamati simulasi dan komputasi, juga memperkaya proses penemuan ilmiah dan mengembangkan pemahaman yang lebih dalam dan tidak diduga, salah satu contohnya adalah dengan menampilkan data/informasi dalam bentuk gambar (Tonni Limbong, 2013).

Adapun sebagai Prinsip Visualisasi Informasi, adalah:

- 1) Fokus pada konten
- 2) Perbandingan
- 3) Integritas
- 4) Resolusi tinggi
- 5) Utilisasi konsep terdahulu dan teruji melewati waktu

Ada berbagai Teknik Visualisasi Informasi antara lain:

- 1) Pengumpulan Jumlah dengan mengakumulasi elemen individual ke dalam sebuah unit yang lebih besar untuk menghadirkannya seperti sesuatu yang lengkap.
- 2) *Overview* dan detail menyediakan baik tinjauan global maupun kemampuan detail *zooming*.
- 3) Fokus dan konteks menunjukkan detail dari satu atau lebih daerah di dalam sebuah konteks global yang lebih besar.
- 4) *Drill-Down* untuk memilih item individual yang lebih kecil dari sebuah tampilan untuk sebuah pendapat detail atau analisis.
- 5) *Brushing* untuk memilih atau spesifikasi nilai, kemudian melihat item yang tepat di tempat lain pada tampilan.

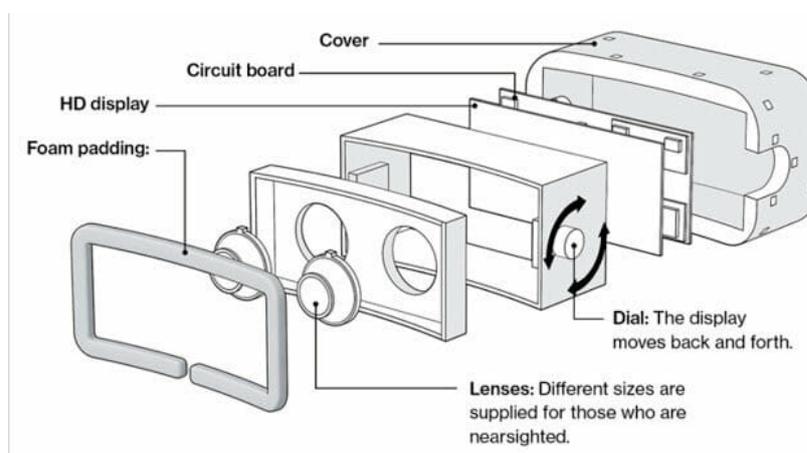
2.3.4 Virtual Reality

Virtual Reality (VR) atau realitas maya merupakan teknologi yang dapat digunakan berinteraksi dengan lingkungan yang visual yang sudah disimulasikan oleh komputer (*computer-simulated environment*), suatu lingkungan yang benar-

benar ditiru keasliannya atau hanya gambaran saja / tidak nyata. Lingkungan realitas maya menampilkan pengalaman visual, yang ditampilkan pada sebuah komputer atau stereoskopik, namun beberapa simulasi dengan tambahan informasi hasil pengindraan, misalnya suara melalui *speaker / headphone*. Pengguna dapat saling terhubung dengan suatu lingkungan nyata atau maya baik melalui penggunaan alat inputan seperti *keyboard* dan *mouse*, atau melalui alat *multimodal* seperti sarung tangan, *polhemus boom arm* dan ban jalan segala arah. Lingkungan yang ditiru misalnya simulasi untuk pilot atau pelatihan tempur atau untuk *game* (Berta Sihite dkk., 2013).

Virtual reality memberikan kontribusi dalam konteks video *game*, arsitektur dan pariwisata, aplikasi yang sangat luas mencakup berbagai bentuk hiburan, pendidikan dan bidang lainnya yang memungkinkan pengalaman dengan relevansi. Pendekatan baru untuk *virtual reality* dimana teknologi menjadi alat berguna dalam bentuk bisnis, komersial atau iklan, misalnya pariwisata dan benda pusaka (Jose Paulo G dkk, 2015).

Menurut Yudi Nugraha B (2014) cara kerja virtual reality ada beberapa cara salah satunya *Head Mounted Display (HMD)* dimana menyatukan cara pandang kedua bola mata. Gambar dari HD *display* diproyeksikan melalui dua lensa yang berbeda sehingga menghasilkan sebuah visual yang terasa nyata. Gambar 2.2 merupakan salah satu jenis *Head Mounted Display (HMD)* atau yang dikenal sebagai kacamata *virtual reality*.



Gambar 2.2 Cara kerja *Head Mounted Display (HMD)*

2.3.5 Gedung

Menurut Peraturan Pemerintah Republik Indonesia nomor 36 tahun 2005 tentang peraturan pelaksanaan undang-undang nomor 28 tahun 2002 tentang bangunan gedung terdapat pada pasal 1 yang mana peraturan pemerintahan tentang bangunan gedung yang diambil pada poin 1 sampai 4 sebagai berikut :

- 1) Bangunan gedung adalah wujud fisik hasil pekerjaan konstruksi yang menyatu dengan tempat kedudukannya, sebagian atau seluruhnya berada di atas dan / atau di dalam tanah dan / atau air, yang berfungsi sebagai tempat manusia melakukan kegiatannya, baik untuk hunian atau tempat tinggal, kegiatan keagamaan, kegiatan usaha, kegiatan sosial, budaya, maupun kegiatan khusus.
- 2) Bangunan gedung umum adalah bangunan gedung yang fungsinya untuk kepentingan publik, baik berupa fungsi keagamaan, fungsi usaha, maupun fungsi sosial dan budaya.
- 3) Bangunan gedung tertentu adalah bangunan gedung yang digunakan untuk kepentingan umum dan bangunan gedung fungsi khusus, yang dalam pembangunan dan / atau pemanfaatannya membutuhkan pengelolaan khusus dan/atau memiliki kompleksitas tertentu yang dapat menimbulkan dampak penting terhadap masyarakat dan lingkungannya.
- 4) Klasifikasi bangunan gedung adalah klasifikasi dari fungsi bangunan gedung berdasarkan pemenuhan tingkat persyaratan administratif dan persyaratan teknisnya.

2.3.6 *Unity3D Engine*

Unity3D suatu *game engine* dengan lisensi *source proprietary*, namun untuk lisensi pengembangan dibagi menjadi dua yaitu gratis dan berbayar sesuai perangkat target pengembang aplikasi. *Unity3D* tidak membatasi publikasi aplikasi, pengguna *Unity3D* dengan lisensi gratis dapat mempublikasikan aplikasi yang dibuat tanpa harus membayar lisensi, tetapi versi gratis dibatasi beberapa fitur tertentu yang hanya disediakan untuk berbayar.

Unity3D dapat mengelola beberapa data seperti objek tiga dimensi, suara, texture, dan sebagainya. *Unity3D* lebih konsentrasi kepada pembuatan grafik tiga dimensi. Dari beberapa *game engine* yang sama-sama menangani grafik tiga dimensi

Unity3D dapat menangani lebih banyak yaitu *Windows, MacOS X, iOS, PS3, wii, Xbox 360* dan *Android*.

Unity3D memiliki kerangka kerja (*framework*) lengkap. Sistem inti menggunakan pilihan Bahasa pemrograman yaitu *C#, JavaScript* maupun *boo*. *Unity3D* menyediakan beberapa alat untuk mempermudah pengembangan yaitu *Unity Tree* dan *Terrain Creator* untuk mempermudah pembuatan vegetasi dan *terrain* serta *MonoDevelop* untuk proses pemrograman (Berta Sihite dkk., 2013).

2.3.7 *Google SketchUp*

SketchUp pertama kali dirilis oleh *Last Software* pada tahun 2000. Sejak *Google* mengakuisisi *Last Software* pada tahun 2006, *SketchUp* lebih dikenal sebagai *Google SketchUp* dan telah berhasil berkembang dengan sangat pesat.. *Google SketchUp* merupakan program grafis tiga dimensi yang dikembangkan dibawah naungan *Google* yang mengkombinasikan *tools* sederhana namun handal dalam desain tiga dimensi pada komputer. Program grafis *Google SketchUp* pendatang baru pada dunia perangkat lunak grafis tiga dimensi. Selain fitur yang *user-friendly*, *Google SketchUp* juga sediakan menjadi dua versi yaitu gratis dan berbayar. Versi gratis *Google SketchUp* memiliki intuitif dan mudah digunakan, *re-imageine* ruang hidup, dan model-model bangunan untuk *Google Earth*, sedangkan untuk versi berbayar atau *Pro Google SketchUp* memiliki semua fitur yang ada di *Google SketchUp Plus*, menukar file dengan *software* lain, membuat desain dokumen menarik hingga membuat laporan, .pdf dan sebagainya (Sari Indah A.S, 2011).

a. Kelebihan

Ada banyak kelebihan yang dimiliki oleh *Google SketchUp* dibandingkan dengan perangkat lunak grafis 3D lainnya, di antaranya :

- 1) Intuitif, mudah digunakan, dan gratis bagi semua orang untuk menggunakannya.
- 2) Dapat memodelkan segala sesuatu yang dapat diimajinasikan.
- 3) *Google SketchUp* membuat pemodelan 3D menjadi menyenangkan.
- 4) Dapat memperoleh model-model secara *online* dan gratis (di *Google 3D Warehouse*).

5) Dapat segera dijelajahi karena dilengkapi dengan *lusinan video tutorial*, *Help Center* dan komunitas pengguna di seluruh dunia.

b. Kekurangan

Selain berbagai kelebihan yang dimiliki, *Google SketchUp* juga masih memiliki beberapa kekurangan yaitu :

- 1) Hanya dapat digunakan pada beberapa *Operating System* tertentu, yaitu : *Windows: XP, Vista*, dan *7* dan *Mac OS X* (versi 10.5 keatas).
- 2) *Google SkecthUp* tampilan objek belum halus dibandingkan aplikasi serupa yang lainnya.

2.3.8 *Adobe Illustrator CC*

Adobe Illustrator merupakan sebuah program perangkat lunak atau program *graphic design* pengolah image berbasis vektor , vektor itu sendiri merupakan sekumpulan titik dan garis yang saling terhubung yang merupakan perpaduan dari warna-warna sehingga membentuk sebuah objek menggambar yang diciptakan oleh *Adobe Systems* yang menggunakan vektor.

Adobe illustrator CC adalah aplikasi *software* ilustrasi profesional dibuat oleh *Adobe System Incorporated*. *Adobe illustrator* dapat membuat semua dari grafik sederhana, ikon, dan teks yang kompleks dengan banyak *layer* dapat digunakan menata halaman dalam persentasi ataupun halaman *web adobe illustrator* menawarkan berbagai macam *tool* penting. Menggunakan kombinasi dengan berbagai macam perintah menu dapat membuat banyak ilustrasi sesuai keinginan dengan kemampuan untuk membuat grafik yang kompleks dari menguasai operasi dasar *adobe illustrator* (Chris Botello, 2014).

2.3.9 *Android*

Android merupakan sebuah *mobile platform* pertama yang lengkap, *opensource* yang dikembangkan dengan menggunakan *Software Development Kit* untuk mengembangkan aplikasi yang *powerfull* dan kaya akan fitur (Lauren Darcey dan Shane Conder, 2012).

Android merupakan sebuah *comprehensive open-source platform* yang didesain untuk perangkat *mobile*. *Android* dimiliki oleh *Open Handset Alliance* dan dipelopori oleh *Google* (Marko Gargenta and Masumi N, 2011).

Android adalah gabungan dari tiga komponen, komponen tersebut yaitu: sebuah *open-source operating system* yang digunakan untuk perangkat *mobile*, sebuah *open-source platform* yang digunakan untuk membuat aplikasi *mobile* dan sebuah perangkat, terutama *mobilephone*, yang menjalankan *Android operating system* dan aplikasi yang dibuat di sistem operasi *Android* (Reto Meier, 2011).

Berikut ini adalah sejumlah *tools* yang disediakan oleh *Android SDK* untuk para pengembang:

- 1) *Emulator.exe* : *Emulator.exe* adalah *tool* yang sangat penting dan berguna pada *Android SDK* yang berfungsi untuk menjalankan *emulator Android*.
- 2) *Adb.exe* : *Android Debug Bridge* atau *adb* merupakan *command-line programming* yang berfungsi untuk memberikan perintah ke *Emulator.exe*.
- 3) *MKSDCARD.exe* : *MKSDCARD.exe* adalah sebuah *tool* yang berfungsi untuk membuat sebuah partisi *drive* yang kecil pada *drive* yang ada dan menyimpan *file* yang diuji pada aplikasi yang memerlukan *read* atau *write* pada *Memory Card*.
- 4) *DX.exe* : *DX.exe* adalah *compiler* pada *Android SDK* yang berfungsi untuk membuat sebuah *file* dengan format *.dex* pada saat menjalankan *file-file java*.

2.3.10 *Android SDK (Software Development Kit)*

Android SDK pertama kali rilis pada 12 November 2007 dan para pengembang memiliki kesempatan untuk memberikan umpan balik dari pengembangan *SDK*. Pada September 2008, *T-Mobile* memperkenalkan ketersediaan *T-Mobile G1* yang merupakan *smartphone* pertama berbasis *Platform Android*, kemudian *Google* merilis *Android SDK 1.0*. *Google* membuat *source code* dari *Platform Android* menjadi tersedia dibawah lisensi *Apache's Open Source*.

Android SDK merupakan *tool* yang bagi para penggunanya yang ingin mengembangkan aplikasi berbasis *Google Android*. *Android SDK* mencakup

seperangkat alat pengembangan yang komprehensif. *Android SDK* terdiri dari *debugger*, *libraries*, *handset*, *emulator*, dokumentasi, contoh kode dan tutorial (Wandy Damarullah dkk., 2013).

Android SDK dapat mentransfer *java class* ke *file* dari *file* bertipe *.dex*. *file .dex* ke *Dalvik Virtual Machine* melakukan fungsi eksekutor. Selain itu beberapa fungsi lain dari *tools* seperti *Android emulator* dan *debugger logcat* (Yi-Jen Mon, 2013).

2.3.11 JDK (*Java Development Kit*)

Java Development Kit merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode *Java* menjadi *bytecode* yang dapat dimengerti dan dapat dijalankan oleh *Java Runtime Environment*. *Java Development Kit* wajib terinstal pada komputer yang akan melakukan proses pembuatan aplikasi berbasis *Java*. Namun *Java Development Kit* tidak wajib terinstall dikomputer yang akan menjalankan aplikasi yang dibangun menggunakan *Java*. *Java Development Kit* merupakan *superset* dari *JRE (Java Runtime Environment)* dan berisi semua yang ada di dalam *JRE*, ditambah *compiler* dan *debugger* yang diperlukan untuk mengembangkan *applet* dan aplikasi (Merliana dan Timotius W, 2010).

2.3.12 Bahasa Pemrograman C#

C# (C Sharp) adalah sebuah bahasa pemrograman berbasis objek yang didukung oleh *Microsoft .NET Framework*. *Microsoft .NET Framework* merupakan perantara agar aplikasi dengan bahasa pemrograman yang didukung dapat berkomunikasi dengan sistem operasi. Selain itu, *Microsoft .NET Framework* juga mendukung bahasa pemrograman *C#* agar dapat berkomunikasi dengan bahasa pemrograman lainnya yang terdapat didalam *Microsoft .NET Framework* seperti *Visual Basic .NET*, *F#*, ataupun *C++*. Dengan kata lain, aplikasi yang kita buat dapat menggunakan komponen-komponen lain yang dibuat dengan menggunakan *VB .NET*, *J#*, atau *C++* (Erico Darmawan H dan Laurentius R, 2011).

2.3.13 Perangkat *Input*

Perangkat *input* adalah sarana bagi pengguna berinteraksi dengan dunia maya. Perangkat ini mengirim sinyal ke sistem tentang tindakan pengguna, sehingga memberikan reaksi yang tepat kembali kepada pengguna melalui perangkat *output* secara *real-time*. Perangkat ini dapat diklasifikasikan misalnya sebagai alat *tracking*, *point-input*, *bio-controler* dan perangkat suara.

Alat *tracking* kadang disebut juga sebagai sensor posisi, yang digunakan dalam pelacakan posisi pengguna. Contoh jenis sensornya seperti elektromagnetik, *ultra-sonik*, optik, mekanik dan sensor *gyroscopic*, sarung tangan data, pengendali saraf dan otot atau bio. Contoh alat *point-input* adalah *mouse 3D*, *joystick*, *pointer*, dan lengan mekanik dengan *display visual* (BOOM).

Namun demikian, perangkat lain yang lebih canggih juga digunakan, seperti sarung tangan interaktif dan perangkat pengenalan suara seperti yang digunakan pada *Nintendo Wii* (video game konsol). Bahkan, gerakan tubuh pengguna dapat dilacak menggunakan setelan pakaian atau baju khusus dengan perangkat sudut pengukuran ditempatkan pada titik ataupun sendi, atau perangkat pelacakan nirkontak, yang melibatkan penggunaan sensor optik, suara *ultrasonic*, emisi inframerah, atau medan elektromagnetik.

Pendeteksi gerakan *real-time* ini digunakan untuk menangkap posisi pengguna dan gerakannya. Sehingga hal ini memungkinkan pengguna untuk berinteraksi dengan dunia maya secara intuitif dengan menggunakan gerakan alami. Perangkat input yang memberi informasi gaya, daya ataupun gerakan untuk pengguna disebut perangkat *haptic*. Teknologi *haptic* menyajikan berbagai bentuk penginderaan gerakan dan gaya serta kekuatan dan memberi umpan balik taktil untuk pengguna.

Melalui perangkat ini pengguna dapat berinteraksi dengan dunia maya melalui gerakan tangan untuk menyentuh, menggeser dan menangkap objek virtual dengan sensasi yang mendekati benda nyata. Namun, karena banyaknya keterbatasan yang dimiliki *haptic* maka biasanya pengguna lebih banyak memanfaatkan isyarat *visual* untuk membantu mereka mengatasi kendala ini (Yudi Nugraha B, 2014).

2.3.14 Perangkat *Output*

Perangkat *output* adalah alat yang menerima umpan balik dari *VR Engine* dan menyajikannya ke pengguna melalui alat *output* yang sesuai untuk merangsang indra. Beberapa klasifikasi alat output berdasarkan pada indera adalah: grafis (*visual*), *audio* (pendengaran), *haptic* (perabaan atau gaya), bau dan rasa.

Tampilan *visual* adalah *output* paling populer dalam sistem VR sedangkan jenis tampilan lain adalah sebagai pelengkap. Tampilan visual adalah perangkat yang dikhususkan untuk mata pengguna yang menyajikan dunia 3D. Ada enam kategori tampilan *visual* yang tentunya masing-masing memberikan tingkat yang imersif yang berbeda, yakni *display desktop*, *head-mounted displays* (HMD), *arm-mounted displays*, *single-screen displays*, *surround-screen displays* (*CAVE*, *Panoramic screen*) dan *volumetric displays*. Alat *output* yang juga populer dalam VR juga sebagai alat input adalah sarung tangan interaktif, *speaker*, *earphone* dan kaca mata 3D untuk tampilan *stereo*.

Tampilan *visual stereoscopic* digunakan untuk menyajikan dunia maya dalam 3D. Ada dua jenis *render stereoscopic* yang dikembangkan untuk proyeksi gambar ke layar yaitu pasif dan aktif. *Stereoscopic* aktif memerlukan kacamata rana (*shutter glasses*). Setiap mata menerima gambar yang terpisah untuk memberikan kesan penglihatan *stereo*. *Stereoscopic* pasif menggunakan polarisasi atau kacamata merah / hijau. Dua set gambar diproyeksikan secara bersamaan. Setiap mata menerima gambar yang berbeda karena setiap lensa kacamata filter keluar satu set gambar. Jenis solusi pasif tidak bisa dipakai pada display desktop (Yudi Nugraha B, 2014).

2.3.15 *Storyboard*

Storyboard adalah sketsa gambar yang disusun berurutan sesuai naskah, *stroyboard* dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji, sehingga menghasilkan persepsi yang sama pada ide cerita. Salah satu keunntngan *storyboard* adalah dapat membuat pengguna untuk mengalami perubahan dalam alur cerita untuk memicu reaksi atau ketertarikan yang lebih

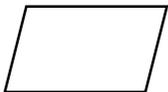
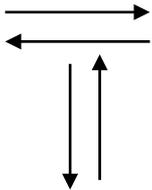
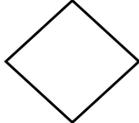
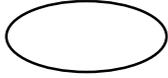
dalam. Secara kronologis untuk membangun rasa penasaran dan ketertarikan (Nelly Indriani dan Irwan Setiawan, 2012).

2.3.16 *Flowchart*

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program (Meita R dan Sukadi, 2014).

Berikut adalah simbol yang digunakan dalam menyusun *flowchart*, kegiatan yang diawali serta aturan yang diterapkan dalam penggunaan simbol disajikan pada Tabel 2.1

Tabel 2.1. Simbol *Flowchart*

Simbol	Nama	Keterangan
	Simbol Proses	Melambangkan kegiatan pemrosesan input, menuliskan operasi-operasi yang dikenakan pada input, maupun operasi lainnya.
	Simbol <i>Input-Output</i> (I/O)	Merepresentasikan fungsi I/O yang membuat sebuah data dapat diproses (<i>input</i>) atau ditampilkan (<i>output</i>) setelah mengalami eksekusi informasi.
	Simbol Garis Alir	Menghubungkan setiap langkah dalam flowchart dan menunjukkan kemana arah aliran diagram. Anak panah ini harus mempunyai arah dari kiri ke kanan atau dari atas ke bawah.
	Simbol Percabangan	Melambangkan percabangan, yaitu pemeriksaan terhadap suatu kondisi. Hasil dari pemeriksaan dalam simbol ini adalah YES atau NO.
	Simbol Terminator	Terminator berfungsi untuk menandai awal dan akhir dari suatu <i>flowchart</i> .

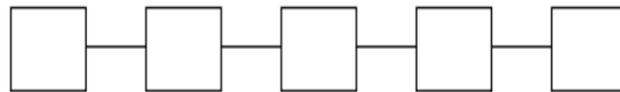
2.3.17 Struktur Navigasi

Struktur navigasi adalah alur dari suatu program. Menentukan struktur navigasi merupakan hal yang sebaiknya dilakukan sebelum membuat suatu aplikasi multimedia (Nelly Indriani dan Irwan Setiawan, 2012).

Ada 4 macam bentuk dasar dari struktur navigasi yang biasa digunakan dalam proses pembuatan aplikasi multimedia, yaitu :

1) Linier

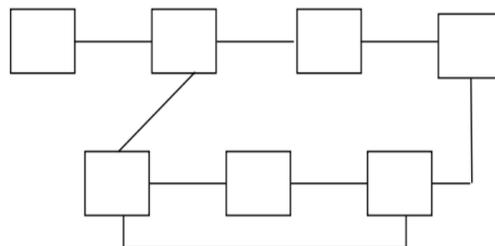
Linier merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurut. Struktur ini menampilkan satu demi satu tampilan layar secara berurut menurut urutannya dan tidak diperbolehkan adanya percabangan. Tampilan yang ditampilkan adalah satu halaman sebelumnya atau satu halaman sesudahnya. Pemakai menelusuri program secara berurutan. Navigasi Linier dapat disajikan pada Gambar 2.3.



Gambar 2.3. Struktur Linier

2) Non Linier

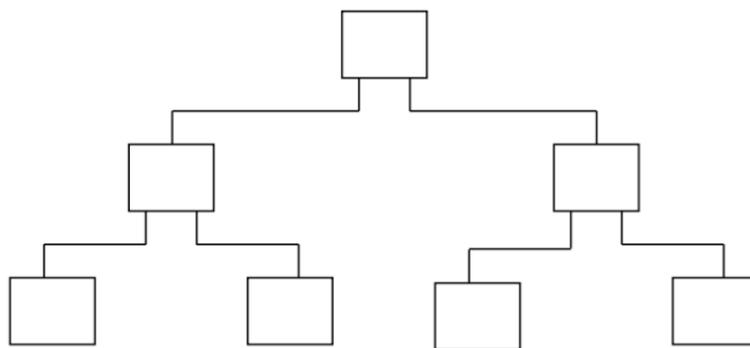
Pada struktur ini diperkenankan membuat percabangan, pemakai bebas menelusuri program tanpa dibatasi rute. Percabangan ini berbeda dengan percabangan pada struktur hirarki, pada percabangan struktur ini tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada *master page* dan *slave page*. Navigasi Non Linier dapat disajikan pada Gambar 2.4.



Gambar 2.4. Struktur Non Linier

3) Hirarki

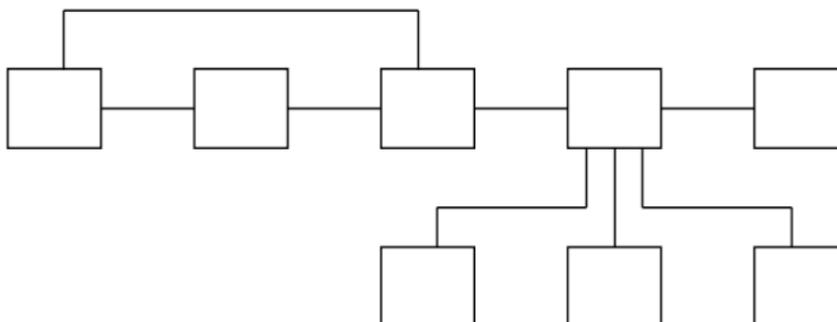
Struktur hirarki merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *master page*, halaman utama ke satu. Halaman utama ini akan mempunyai halaman percabangan yang disebut *slave page*, halaman pendukung. Jika salah satu halaman pendukung diaktifkan maka tampilan tersebut akan bernama *master page*, halaman utama kedua. Navigasi Hirarki dapat disajikan pada Gambar 2.5.



Gambar 2.5. Struktur Hirarki

4) Campuran

Struktur campuran merupakan gabungan dari ketiga struktur sebelumnya. Struktur ini banyak digunakan dalam pembuatan aplikasi multimedia sebab dapat memberikan keinteraksian yang lebih tinggi. Navigasi Campuran dapat disajikan pada Gambar 2.6.



Gambar 2.6. Struktur Campuran

2.3.18 Pengujian *Black-Box*

Menurut Rosa A.S dan M. Shalahuddin (2011), *black-box* testing adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

- 1) Fungsi yang tidak benar atau hilang.
- 2) Kesalahan *interface*.
- 3) Kesalahan dalam struktur data atau akses *database* eksternal.
- 4) Kesalahan kinerja.
- 5) Inisialisasi dan kesalahan terminasi.