

BAB II

LANDASAN TEORI

Bab ini akan membahas mengenai konsep dasar yang digunakan sebagai acuan dan penunjang serta tools yang digunakan untuk membangun aplikasi yang meliputi Aplikasi, Augmented Reality, Android, Android SDK, UML, Alpha dan Beta Testing, Skala Likert.

2.1 Aplikasi

Aplikasi adalah perangkat lunak yang siap pakai untuk menjalankan instruksi-instruksi dari *user* (pengguna) aplikasi tersebut dengan tujuan untuk membantu berbagai kegiatan yang dilakukan oleh manusia. Ada beberapa pendapat dari para pengembang tentang aplikasi yang akan penulis kemukakan, yaitu sebagai berikut [2]:

1. **Menurut Pranama (2012)**

“Aplikasi adalah satu unit perangkat lunak yang sengaja dibuat untuk memenuhi kebutuhan akan berbagai aktivitas ataupun pekerjaan, seperti aktivitas perniagaan, periklanan, pelayanan masyarakat, game, dan berbagai aktivitas lainnya yang dilakukan oleh manusia.”

2. **Menurut Yuhefizar (2012)**

“Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Jadi aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak dengan computer untuk memudahkan pekerjaan atau tugas-tugas seperti penerapan, penggunaan dan penambahan data yang dibutuhkan. ”

3. **Menurut Verman dkk (2009)**

“Aplikasi adalah perangkat intruksi khusus dalam komputer yang dirancang agar kita menyelesaikan tugas-tugas tertentu.”

2.2 Augmented Reality

AR (*Augmented Reality*) atau dalam bahasa Indonesianya realitas tertambah adalah teknologi yang menggabungkan benda maya dua dimensi

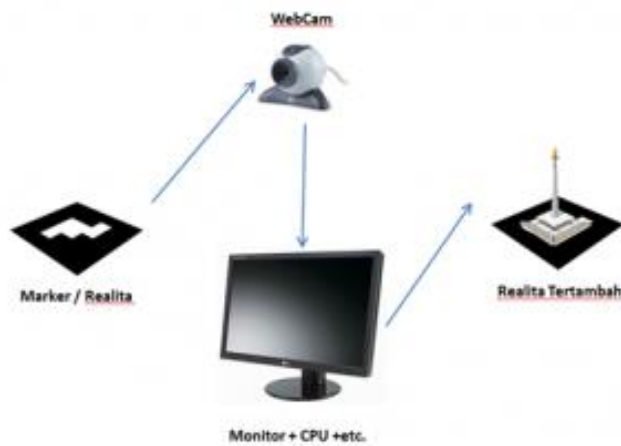
ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi kemudian memproyeksikan benda-benda maya tersebut dan memungkinkan pengguna melihat dunia nyata dan dunia virtual dalam waktu nyata (*real time*). Berbeda realitas maya atau VR (*Virtual Reality*) yang sepenuhnya menggantikan kenyataan, realitas ditambah sekedar atau melengkapi kenyataan. *Virtual Reality* atau Realitas Maya adalah teknologi yang memungkinkan *user* dapat berinteraksi dengan suatu lingkungan berdimensi 3 yang disimulasikan oleh komputer terhadap suatu objek nyata atau imajinasi, sehingga membuat *user* seolah-olah terlibat secara fisik pada lingkungan tersebut.

Pada *Augmented Reality* ada tiga karakteristik yang menjadi dasar atas sistem tersebut, diantaranya adalah kombinasi pada dunia nyata dan virtual, interaksi yang berjalan secara real-time, dan karakteristik yang terakhir adalah 6 bentuk objek yang berupa model 3 dimensi atau 3D [3].

“Realitas ditambah / AR dapat diaplikasikan untuk semua indra, tidak hanya visual, termasuk pendengaran, sentuhan dan penciuman. Gunanya untuk memperkaya pengalaman penggunanya, membantu persepsi dan interaksi penggunanya dengan dunia nyata.”[4] Teknologi ini biasanya digunakan pada bidang komunikasi, medis, militer dan manufaktur yang memiliki tingkat resiko lebih besar dan membutuhkan tambahan dari objek semu yang meniru objek nyata sebelum diimplementasikan.

2.2.1 Cara kerja Augmented Reality

Prinsip kerja *Augmented Reality* yaitu mendeteksi citra (gambar), kemudian citra yang akan diolah adalah permukaan bidang datar. Penerapannya yaitu pada kamera yang telah dikalibrasi akan mendeteksi marker yang berupa bidang datar, kemudian setelah proses mengenali dan menandai pola permukaan sudut bidang datar, *kamera* akan melakukan komparasi kepada marker, apakah marker sesuai dengan *database* yang dimiliki atau tidak. Apabila tidak dikenali, informasi marker yang tersimpan tidak akan diolah, akan tetapi apabila sesuai, maka informasi pada marker bidang datar akan digunakan untuk menampilkan objek 3D dan melakukan *render* pada objek 3D.



Gambar 2.1 Cara Kerja Augmented Reality

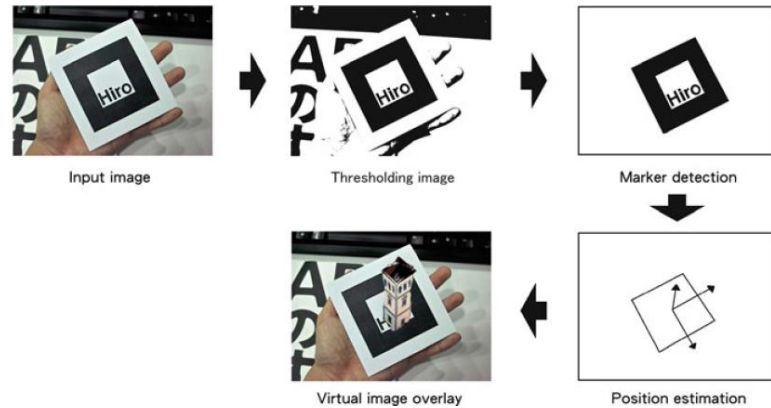
Pada umumnya Augmented Reality membutuhkan alat masukan (*input device*) seperti kamera atau *Webcam*, alat keluaran (*output device*) seperti monitor atau *Head Mounted Display (HMD)*, alat pelacak (*tracker*) agar benda maya tambahan berupa penanda (*marker*) yang dihasilkan berjalan secara *real-time* atau walaupun benda nyata yang menjadi induknya digeser-geser akan tetap muncul di atas marker, dan komputer untuk menjalankan program AR. “Salah satu aspek yang paling penting dari *Augmented Reality* adalah untuk membangun teknik yang tepat untuk interaksi intuitif antara pengguna dan konten virtual AR aplikasi.”[5]

2.2.2 Metode Augmented Reality Markerless-based

Ada beberapa metode yang digunakan pada *Augmented Reality*, salah satunya adalah *Markerless Based Tracking*. Metode ini memiliki fungsi tanpa marker, metode yang dapat menampilkan elemen – elemen digital secara langsung tanpa menggunakan media marker. Metode ini merupakan teknologi *Augmented Reality* yang dihadirkan dari gabungan teknologi *computer vision* dan *image processing* yang mencari informasi dari sebuah citra secara langsung. Dimana marker tersebut dibutuhkan untuk memunculkan objek 3D yang akan dimunculkan setelah proses *tracking* dan *positioning* dilakukan.

Tracking marker merupakan proses yang dilakukan oleh sistem ketika melakukan pemindaian gambar, dalam hal ini terhadap orientasi bidang datar. Vuforia Sdk sebagai penyedia layanan melakukan proses *computer vision*, dimana

pada proses tersebut Vuforia menggunakan metode NFT (*Natural Feature Tracking*). *Natural Feature* merupakan salah satu metode untuk memperluas jangkauan dan menstabilkan gambar pada sistem pelacakan *Augmented Reality*.



Gambar 2.2 Proses Tracking dari Teknologi Augmented Reality

Secara garis besar proses *tracking marker* dari teknologi *Augmented Reality* dapat dilihat dari gambar 2.2 dimana pada awal proses kamera akan melacak marker, yang dilanjutkan pada *thresholding image* (pemisahan warna menjadi 1 dan 0) dimana 1 untuk warna putih dan 0 untuk warna hitam, lalu dilanjutkan pencocokan *marker*, kemudian diakhiri dengan penempatan animasi objek 3D. Untuk proses *tracking* dengan metode *markerless*, algoritma yang berjalan di aplikasi AR akan mengenali pola, warna dan fitur lainnya termasuk sudut kemiringan dan kontras warna bidang datar.

Ibanez, A. S., dan Figueras, J. P. [6] mengatakan bahwa “untuk mengenali gambar Vuforia memproses fitur-fiturnya dan membandingkan hasilnya ke *database* yang terkandung dalam dataset aplikasi. Jika hasil cocok, maka gambar tersebut dianggap telah dikenali. Oleh karena itu jika gambar tidak jelas, kemungkinan gambar tidak akan terdeteksi.”

2.2.3 Vuforia

Vuforia merupakan *Software Development Kit* (SDK) yang memungkinkan pembangunan aplikasi *Augmented Reality* pada perangkat *mobile* dengan dukungan untuk Android, iOS dan ekstensi Unity3D yang akan dibangun. Vuforia memiliki arsitektur yang kuat untuk mengoptimalkan proses *tracking*,

registering dan *graphics visualization* memungkinkan berbagai fitur untuk *tracking* dan *registering frame marker* (marker acuan), gambar dan teks .

“Arsitektur pada komponen utama Vuforia SDK terdiri dari 6 macam yaitu kamera, *image converter*, *tracker*, *video background renderer*, *application code*, dan *target resource*.” [6]

1. Kamera

Setiap *frame* ditangkap oleh kamera dapat dipastikan dan dapat melewati pelacak. Pengembang hanya perlu untuk memberitahu kapan kamera harus memulai atau berhenti menangkap sebuah gambar. kemudian *frame* kamera secara otomatis sampai di perangkat, tergantung format dan ukuran gambarnya.

2. Image Converter

Format pixel converter tunggal mengkonversi antara format kamera ke format yang sesuai dengan OpenGL-ES *rendering* dan untuk pelacakan. Konversi ini termasuk kedalam *down-sampling*. fungsinya untuk membuat gambar kamera dengan resolusi berbeda dalam konversi *stack frame*.

3. Tracker

Pelacak yang berisi algoritma visi komputer yang mendeteksi dan melacak benda di dunia nyata kedalam frame video kamera. Dari hasil gambar kamera, terdeteksi target baru atau marker oleh algoritma yang berbeda, kemudian memeriksa tombol virtual. Hasil akan disimpan ke sebuah *state object* menggunakan *renderer* video background dan dapat diakses dari kode aplikasi. Pelacak hanya dapat mengaktifkan satu data yang aktif dalam satu waktu walaupun dapat memuat beberapa set data.

4. Video Background Renderer

Video background renderer membuat gambar kamera dapat disimpan dalam *state* objek. Kinerja video *Background Renderer* dioptimalkan untuk perangkat tertentu.

5. Application Code

Developer aplikasi harus melakukan 3 kunci langkah dalam pengkodean aplikasi termasuk menginisialisasi semua komponen di atas. *State object* akan diperbarui untuk setiap frame yang akan diproses, metode yang dipanggil akan dibuat oleh aplikasi. *Developer* aplikasi harus :

- Memperbaharui status dari elemen untuk pendeteksian target yang baru, yaitu dengan cara mengantrikan *state object*.
- Memasukkan data yang baru dengan cara memperbaharui logika aplikasi.
- Membuat lapisan dari tampilan *Augmented*

6. Target Resource

Target Resource dibuat menggunakan *Target Management System* secara *on-line*. Dataset yang sudah di-download berisi file konfigurasi XML yang memungkinkan pengembang untuk mengkonfigurasi fitur *trackable* dan file biner tertentu yang berisi dataset *trackable*. Asset ini akan dikompilasi oleh pengembang aplikasi ke dalam paket installer aplikasi dan digunakan pada saat *run-time* oleh Vuforia SDK.

2.2.4 Unity 3D

Unity 3D merupakan sebuah *game engine*, yaitu software pengolah gambar, grafik, suara, input, dan lain-lain yang ditunjukkan untuk membuat game. Selain game, unity engine ini dapat mengerjakan pemrosesan grafik 2D dan 3D.

Kelebihan Unity dibanding dengan *game engine* lainnya adalah kemampuan membuat game cross platform. Dengan Unity 3D, game yang Anda buat dapat dimainkan di berbagai perangkat, seperti *smartphone* dan game console. Unity sendiri dapat membuat berbagai macam game, seperti RPG (*Role Playing Game*), shooter, racing, dan lain sebagainya. Berikut adalah fitur-fitur yang ada pada Unity 3D. [7]

1. Rendering

Unity 3D mendukung penggunaan *software* pengolahan gambar lain, seperti 3ds Max, Softimage, Allegorithmic Substance , Blender, Modo, ZBrush, Maya, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks untuk meningkatkan kualitas tokoh dalam game.

2. Scripting

Unity dibangun menggunakan *Scripting Mono 2.6*. Mono 2.6 yang merupakan implementasi open source dari .Net Framework. Unity juga mendukung beberapa bahasa pemrograman antara lain JavaScript, C#, dan Boo (menggunakan sintaks Phyton).

3. Asset Store

Asset store merupakan tempat untuk mendapatkan asset yang digunakan untuk menunjang pembuatan game. *Asset* pada Unity 3D dibagi menjadi dua, yaitu eksternal dan internal. *Asset* eksternal merupakan asset yang ditambahkan dari sumber di luar Unity 3D, seperti 3D model, Texture, Sound Effect. *Asset* internal merupakan asset yang sudah ada di dalam Unity, seperti Materials, Shaders, Cube Maps, Physics Materials, dan Prefabs.

4. Platform

Unity 3D dapat dijalankan secara cross platform. Platform yang didukung Antara lain PlayStation 3, Blacberry 10, Windows 8, Windows Phone 8, Windows, Mac, Linux, Android, Xbox One, iOs, Unity Web Player, Adobe Flash, Xbox 360, Wii U dan Wii.

2.2.5 Blender 3D

Blender 3D merupakan aplikasi visualisasi 3D yang mempunyai fitur cukup lengkap dan populer. Software ini bersifat *Open Source* atau dapat digunakan oleh siapapun, kualitas pencitraan digital tidak kalah dengan software-software grafis 3D lainnya seperti 3DS Max. Selain membuat animasi 3D, Blender 3D juga dapat digunakan untuk pengeditan video, mengolah dan membuat animasi, dan membuat *game*.

2.3 Stimulasi Bayi

Stimulasi adalah rangsangan yang dilakukan kepada bayi sejak lahir (akan lebih baik lagi sejak di dalam kandungan) dilakukan setiap hari, untuk merangsang semua sistem indera (pengecapan, perabaan, penglihatan, perabaan, pembauan, pendengaran). Kemudian harus dapat merangsang motorik kasar dan motoric halus kaki, tangan dan jari-jari, mengajak berkomunikasi, termasuk merangsang perasaan yang menyenangkan untuk bayi dan anak-anak (Herawati 2011) [15].

Otak pada anak sudah mempunyai sel syaraf yang bermilyaran jumlahnya. Ketika dilahirkan, namun setelah dilahirkan jumlah tersebut banyak yang hilang. Ketika otak mendapatkan suatu stimulus yang baru, maka otak juga akan mempelajari sesuatu yang baru. Stimulus tersebut akan menyebabkan sel syaraf membentuk sebuah koneksi baru untuk menyimpan informasi. Sel-sel yang terpakai untuk menyimpan informasi akan mengembang, sedangkan yang jarang atau tidak terpakai akan musnah. Maka dari itu stimulasi yang diberikan secara rutin menjadi sangat penting. Stimulasi yang terus-menerus diberikan secara rutin akan memperkuat hubungan antar syaraf yang telah terbentuk sehingga secara otomatis fungsi otak akan menjadi semakin baik [22].

Anak yang mendapat stimulasi yang terarah akan lebih cepat berkembang dibandingkan anak yang kurang atau bahkan tidak mendapat stimulasi sama sekali. Stimulasi juga dapat berfungsi sebagai penguat yang bermanfaat bagi perkembangan anak. Berbagai macam stimulasi seperti stimulasi visual (penglihatan), verbal (bicara), auditif (pendengaran), taktil (sentuhan) dll dapat mengoptimalkan perkembangan anak [16].

Berikut adalah tabel stimulasi pada usia 0-24 bulan berdasarkan pedoman SDIDTK yang menekankan pada 4 aspek perkembangan [19]:

Tabel 2.1 Tabel Stimulasi Bayi Usia 0-12 Bulan

Usia	Motorik Kasar	Motorik Halus	Bicara dan Bahasa	Sosialisasi dan Kemandirian
0 – 3 Bulan	<ul style="list-style-type: none"> - Mengangkat kepala - Berguling-guling - Menahan kepala tetap tegak 	<ul style="list-style-type: none"> - Melihat, meraih dan menendang mainan gantung - Memperhatikan benda bergerak - Memegang benda - Meraba dan merasakan bentuk permukaan 	<ul style="list-style-type: none"> - Berbicara - Menirukan suara - Mengenali berbagai suara 	<ul style="list-style-type: none"> - Memberi rasa aman dan kasih sayang - Mengajak bayi tersenyum - Meniru ocehan dan mimik muka bayi - Mengayun bayi - Meninabobokan
3 – 6 Bulan	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 0-3 Bulan - Menyangga berat - Mengembangkan kontrol terhadap kepala 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 0-3 bulan - Memegang benda dengan kuat - Memegang dengan kedua tangan 	<ul style="list-style-type: none"> - Lanjutkan stimulasi 0-3 bulan - Mencari sumber suara - Menirukan kata-kata 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 0-3 bulan - Bermain ci-luk-ba - Melihat dirinya dikaca - Berusaha meraih mainan

	<ul style="list-style-type: none"> - Duduk 	<ul style="list-style-type: none"> - Makan sendiri - Mengambil benda kecil 		
6 – 9 Bulan	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 3-6 bulan - Merangkak - Menarik ke posisi berdiri - Berjalan berpegangan - Berjalan dengan bantuan 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 3-6 bulan - Memasukan benda kedalam wadah - Memegang alat tulis dan mencoret-coret - Bermain mainan yang mengapung di air - Membuat bunyi-bunyian - Menyembunyikan dan mencari benda 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 3-6 bulan - Menyebutkan nama gambar dibuku - Menunjuk dan menyebutkan nama gambar-gambar 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 3-6 bulan - Permainan bersosialisasi

9 – 12 Bulan	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 6-9 bulan - Bermain bola - Membungkuk - Berjalan sendiri - Naik tangga 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 6-9 bulan - Menyusun balok-balok - Menggambar - Bermain di dapur 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 6-9 bulan - Menirukan kata-kata - Berbicara dengan boneka - Bersenandung dan bernyanyi 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 6-9 bulan - Minum sendiri dari sebuah cangkir - Makan bersama-sama - Menarik mainan yang letaknya agak jauh
12 - 24 Bulan	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 9-12 bulan - Dapat berjalan mundur - Melompat di satu tempat - Mengajak bermain keluar rumah 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 9-12 bulan - Belajar melepas baju sendiri - Belajar memakai baju sendiri - Melatih tentang kebersihan diri 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 9-12 bulan - Menunjukkan kebanggaan saat berhasil melakukan sesuatu 	<ul style="list-style-type: none"> - Lanjutkan stimulasi usia 9-12 bulan - Mengikuti perintah sederhana dan membantu menyelesaikan tugas sederhana. - Membangun relasi berteman dengan anak lain

2.4 Android

Menurut Nazaruddin Safaat yang dikutip oleh Gugy Munzy [8] dalam penelitiannya, “Android adalah sistem operasi berbasis Linux bagi telepon seluler seperti telepon pintar dan komputer tablet.” Android juga menyediakan *platform* terbuka bagi para pengembang untuk mengembangkan aplikasi yang telah mereka bangun untuk digunakan di berbagai macam peranti gerak.

Google Inc. sepenuhnya membangun Android dan menjadikannya bersifat terbuka (*Open Source*) sehingga pengembang dapat dengan bebas menggunakan Android tanpa perlu mengeluarkan biaya untuk mendapatkan lisensi resmi dari Google, Selain itu dapat membangun aplikasi dengan platform yang berbasis Android tanpa adanya batasan. Android Software Development Kit (SDK) menyediakan *Application Programming Interface* (API) dan *tools* yang dibutuhkan untuk memulai perancangan aplikasi pada platform Android berbasis bahasa pemrograman Java. [9]

Hingga saat tulisan ini dibuat, ada beberapa versi dari Sistem Operasi Android yang telah di rilis oleh google, antara lain:

1. Versi Android 1.1
2. Versi Android 1.5 (Cupcake)
3. Versi Android 1.6 (Donut)
4. Versi Android 2.0/2.1 (Eclair)
5. Versi Android 2.2 (Froyo: Frozen Yoghurt)
6. Versi Android 2.3 (Gingerbread)
7. Versi Android 3.0/3.1 (Honeycomb)
8. Versi Android 4.0 - (Ice Cream Sandwich)
9. Versi Android 4.1 - 4.3 (OS JB: Jelly Bean)
10. Versi Android 4.4 (KitKat)
11. Versi Android 5.0 (Lollipop)
12. Versi Android 6.0 (Marshmallow)
13. Versi Android 7.0 (Nougat)
14. Versi Android 8.0 (Oreo)
15. Versi Android 9.0 (Pie)

16. Versi Android 10 (Android Q)

Untuk penggunaan aplikasi maupun dalam pembuatan aplikasi, penulis menyarankan untuk menggunakan OS Android dengan versi minimal 5.0 (Lollipop). Bagian ini diperlukan untuk Analisis Kebutuhan Fungsional pada Bab 3 sebagai kebutuhan perangkat lunak.

2.5 Android SDK Manager

Android Software Development Kit Manager merupakan software emulator Android. SDK Manager dapat menjalankan aplikasi-aplikasi yang terdapat pada Android karena bersifat emulator. Para *programmer* juga dapat menggunakan Android SDK ini sebagai *tools* untuk membangun aplikasi berbasis platform Android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Nazaruddin Safaat juga mengungkapkan bahwa Android SDK merupakan tools API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada platform Android yang menggunakan bahasa pemrograman Java. [10]

2.6 UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk melakukan perancangan, mendokumentasikan, visualisasi, menspesifikasikan pada sistem perangkat lunak. Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. [11]

Dengan menggunakan UML, kita dapat dengan mudah membangun sebuah model untuk semua jenis aplikasi perangkat lunak, Terlebih lagi aplikasi yang telah dibangun tersebut juga dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta dapat ditulis dalam bentuk bahasa pemrograman apapun. UML menggunakan *class* dan *operation* dalam konsep dasarnya, yang membuatnya lebih cocok untuk pembuatan perangkat lunak dalam bahasa berorientasi objek seperti C#, Java dan VB.NET. Selain itu, UML juga dapat digunakan untuk pemodelan aplikasi prosedural dalam VB atau C.

Notasi UML merupakan hasil turunan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering). [12]

Terdapat 10 macam diagram yang tersedia pada UML untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk pemodelan proses bisnis.
2. *Conceptual Diagram* untuk pemodelan konsep yang terdapat pada aplikasi.
3. *Sequence Diagram* untuk pemodelan pengiriman pesan (message) antar objects.
4. *Collaboration Diagram* untuk pemodelan interaksi antar objects.
5. *State Diagram* untuk pemodelan *objects behaviour* pada suatu sistem.
6. *Activity Diagram* untuk pemodelan perilaku objects dan *use case* pada suatu sistem.
7. *Class Diagram* untuk pemodelan struktur kelas.
8. *Object Diagram* untuk pemodelan struktur dari object object.
9. *Component Diagram* untuk pemodelan komponen object.
10. *Deployment Diagram* untuk pemodelan distribusi pada aplikasi yang akan dibangun.

Dibawah ini adalah UML yang digunakan oleh penulis dalam penelitian ini yaitu use case diagram, sequence diagram, class diagram dan collaboration diagram.

1. Use Case Diagram

Use case diagram berfungsi untuk memodelkan semua bisnis proses berdasarkan perspektif pengguna sistem. Diagram *Use case* ini terdiri dari diagram untuk *actor* dan *use case*. *Actor* merupakan orang yang berinteraksi dengan sistem aplikasi. Kumpulan operasi yang dikerjakan oleh *actor* di representasikan oleh *Use Case*. *Use case* digambarkan dengan bentuk elips yang di dalamnya dituliskan nama operasi. Use Case menggunakan garis lurus untuk menghubungkan *Actor* yang melakukan operasi.

2. Sequence Diagram

Model statis dapat digambarkan dengan penggunaan Diagram Class dan diagram Object. Selain itu ada juga pemodelan yang sifatnya dinamis, seperti *Diagram Interaction*. *Sequence Diagram* menjelaskan tentang bagaimana suatu operasi dilakukan, dan diagram ini juga termasuk salah satu diagram *interaction*; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Seluruh objek yang masih berkaitan dengan proses jalannya operasi diurutkan dari yang paling kiri hingga kanan, hal ini mengacu pada waktu terjadinya dalam pesan yang terurut.

3. Class Diagram

Class diagram merepresentasikan struktur class yang statis pada sebuah sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. Ada berbagai cara untuk menghubungkan class yg satu dengan yang lain, salah satu contohnya adalah dengan cara *associated* (*terhubung satu sama lain*), *specialized* (ada bagian class yang merupakan spesialisasi dari class lain), *dependent* (class yang bergantung kepada class lain), dan *package* (kumpulan class sebagai satu unit). Dalam sebuah sistem biasanya memiliki beberapa *class diagram*.

4. Collaboration Diagram

Diagram *Collaboration* menggambarkan diagram *interaction*. Diagram ini lebih menekankan pada peranan masing-masing objek yang sama dengan diagram *Sequence*, dan lebih memfokuskan pada kegiatan obyek pada saat pesan itu dikirimkan.

2.7 Konsep Grafika

DDA (*Digital Differential analyzer*) merupakan algoritma untuk membentuk garis dengan menghitung nilai Dx dan Dy, menggunakan rumus $Dy = m \cdot Dx$ (Maliki, Irfan. 2011). Dengan menentukan dua titik endpoint (dalam kasus ini adalah titik awal dan titik akhir) maka dapat dibuat sebuah garis. Setelah dilakukan perhitungan akan diperoleh garis-garis yang terbentuk dari titik koordinat, yang kemudian nilai koordinat tersebut akan dikonversikan menjadi *integer*.

2.7.1 Scaling

Scaling atau Penskalaan adalah suatu operasi yang membuat suatu objek berubah ukurannya baik menjadi mengecil ataupun membesar secara seragam atau tidak seragam tergantung pada faktor penskalaan (scalling factor) yaitu (sx,sy) yang diberikan. sx adalah faktor penskalaan menurut sumbu x dan sy faktor penskalaan menurut sumbu y. Koordinat baru diperoleh dengan :

$$x'' = x + sx \quad (x,y) = \text{titik asal sebelum diskala}$$

$$y'' = y + sy \quad (x'',y'') = \text{titik setelah diskala}$$

Nilai lebih dari 1 menyebabkan objek diperbesar, sebaliknya bila nilai lebih kecil dari 1, maka objek akan diperkecil. Bila (sx,sy) mempunyai nilai yang sama, maka skala disebut dengan uniform scalling. Adapun Scaling dalam transformasi 3D adalah dengan rumus :

$$\begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.7.2 Rotasi

Rotasi atau Putaran merupakan suatu operasi yang dapat membuat objek bergerak berputar pada sumbu putar yang dipilih berdasarkan sudut putaran tertentu. Diperlukan sudut rotasi dan pivot point (xp,yp) untuk melakukan rotasi objek akan dirotasi.

Putaran biasa dilakukan pada satu titik terhadap suatu sumbu tertentu misalnya sumbu x, sumbu y atau garis tertentu yang sejajar dengan sembarang sumbu tersebut. Titik acuan putaran dapat sembarang baik di titik pusat atau pada titik yang lain.

Jika putaran dilakukan searah dengan jarum jam, nilai sudut akan bernilai negatif. Sebaliknya, jika dilakukan berlawanan arah dengan arah jarum nilai sudut akan bernilai positif.

Rotasi dapat dinyatakan dengan :

$$x'' = r \cos(\phi + \theta) = r \cos \phi \cos \theta - r \sin \phi \sin \theta$$

$$y'' = r \sin(\phi + \theta) = r \cos \phi \sin \theta + r \sin \phi \cos \theta$$

sedangkan di ketahui $x = r \cos \phi$, $y = r \sin \phi$, lakukan substitusi, maka :

$$x'' = x \cos \theta - y \sin \theta$$

$$y'' = x \sin \theta + y \cos \theta$$

2.8 Alpha Beta Testing

Pengujian aplikasi yakni dengan pengujian alpha adalah untuk mengidentifikasi dan menghilangkan sebanyak mungkin masalah sebelum akhirnya sampai kepada pengguna, dan dilakukan setelah pengembang menyelesaikan *software*. Sedangkan Pengujian adalah istilah yang digunakan untuk menggambarkan proses pengujian aplikasi secara objektif di mana *software* tersebut telah sampai atau dikirim kepada pelanggan yang menggunakannya pada dunia nyata. Pengujian beta biasanya dilakukan menjelang akhir dari siklus pengembangan produk dan idealnya sebagai validasi bahwa perangkat lunak siap dirilis untuk pelanggan yang sesungguhnya.[13]

Karena menurut Romeo, S.T. [14] sangat tidak mungkin bagi pengembang *software* untuk meramalkan bagaimana pengguna akan menggunakan program. Instruksi-instruksi yang dapat digunakan mungkin akan diintegrasikan dengan salah, kombinasi-kombinasi data yang aneh diluar kebiasaan, keluaran yang kelihatannya jelas bagi tester mungkin akan tidak demikian halnya bagi pengguna.

2.8.1 Metode Black-box Testing

Menurut Ron Patton [13], dalam pengujian *black-box*, penguji hanya tahu apa yang seharusnya dilakukan oleh *software*, namun penguji tidak dapat melihat bagaimana hal itu bisa beroperasi. Berdasarkan hal tersebut, *tester* dapat menentukan nomor-nomor tertentu lebih atau kurang cenderung gagal dan dapat menyesuaikan pengujian yang didasarkan pada informasi tersebut.

Penulis memilih menggunakan metode *Black box*, agar pengujian berfokus pada persyaratan fungsional perangkat lunak dan untuk meminimalisir kegagalan pada hasil output aplikasi.

2.9 Skala Likert

Skala likert adalah skala pengukuran yang dikembangkan oleh Likert (1932). Skala likert mempunyai empat atau lebih butir-butir pertanyaan yang dikombinasikan sehingga membentuk sebuah skor/nilai yang merepresen-tasikan sifat individu, misalkan pengetahuan, sikap, dan perilaku. Dalam proses analisis data, komposit skor, biasanya jumlah atau rata-rata, dari semua butir pertanyaan dapat digunakan [20].

Menurut Saifuddin Anwar yang dikutip oleh Risnita mengatakan bahwa Skala Likert berwujud kumpulan pertanyaan-pertanyaan sikap yang ditulis, disusun dan dianalisis sedemikian rupa sehingga respons seseorang terhadap pertanyaan tersebut dapat diberikan angka (skor) dan kemudian dapat diinterpretasikan. Skala Likert berisi banyak item (*multiple item measure*) tidak terdiri dari satu stimulus atau satu pernyataan saja.[21].

Rumus untuk mencari hasil presentasi dari jawaban kuisisioner dari skala *likert* adalah sebagai berikut:

$$P = \frac{S}{Skorideal} \times 100$$

Keterangan:

P = Nilai persentase yang dicari.

S = Jumlah frekuensi jawaban dikali dengan skala jawaban.

Skor ideal = Skala tertinggi jawaban dikalikan dengan jumlah *responden*.