BAB II
TEORI PENUNJANG

Bab ini berisi penjelasan mengenai konsep dasar yang digunakan sebagai acuan atau penunjang dalam perancangan aplikasi serta tools yang digunakan untuk membangun aplikasi. Diantaranya 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. Penulis akan mengemukakan beberapa pendapat dari para pengembang tentang aplikasi, yaitu sebagai berikut [2]:

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

   "Aplikasi adalah perangkat intruksi khusus dalam komputer yang dirancang agar kita menyelesaikan tugas-tugas tertentu."
2. **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."
3. **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. "

2.2 Augmented Reality

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

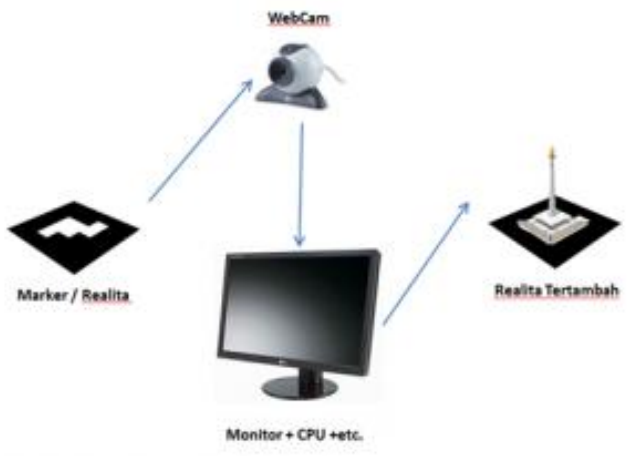
ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata (*real time*). Tidak seperti realitas maya atau VR (*Virtual Reality*) yang sepenuhnya menggantikan kenyataan, realitas tertambah sekedar atau melengkapi kenyataan. *Virtual 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 tertambah / 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 militer, medis, komunikasi, dan manufaktur yang mempunyai risiko besar dan membutuhkan tambahan benda-benda semu yang meniru benda-benda nyata sebelum diimplementasikan.

2.2.1 Cara kerja Augmented Reality

Augmented Reality bekerja berdasarkan deteksi citra (gambar), dan citra yang digunakan adalah marker. Prinsip kerjanya adalah kamera yang telah dikalibrasi akan mendeteksi marker yang diberikan, kemudian setelah mengenali dan menandai pola marker, *webcam* akan melakukan perbandingan apakah marker sesuai dengan *database* yang dimiliki atau tidak. Bila tidak, maka informasi marker tidak akan diolah, tetapi bila sesuai maka informasi marker akan digunakan untuk me-*render* dan menampilkan objek 3D atau animasi yang telah dibuat sebelumnya.



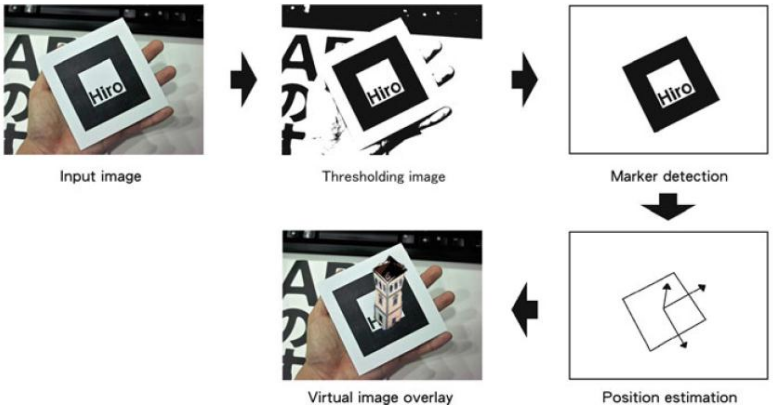
Gambar 2.1 Cara Kerja Augmented Reality

Pada umumnya Augmented Reality membutuhkan alat masukkan (*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* Marker-based

Ada beberapa metode yang digunakan pada *Augmented Reality* salah satunya adalah *Marker Based Tracking*. Metode ini merupakan teknologi *Augmented Reality* yang dihadirkan dari gabungan teknologi *computer vision* dan *image processing* yang mencari informasi dari sebuah gambar secara langsung. Marker yang akan digunakan harus terlebih dahulu dibuat, dengan cara mendaftarkan gambar yang akan menjadi penanda agar bisa memunculkan objek 3D kedalam sebuah *database*. Dalam hal ini penulis menggunakan Vuforia sebagai pembuat *database* marker. 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, terhadap *database* yang sudah ada. Dalam hal ini 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 menstabilkan dan memperluas jangkauan dalam 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 *tresholding image* (pemisahan warna menjadi 1 dan 0) dimana 1 untuk warna putih dan 0 untuk warna hitam, lalu dilanjutkan pencocokan *marker*, dan diakhiri dengan penempatan animasi objek 3D.

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

Kamera memastikan setiap *frame* ditangkap dan bisa melewati pelacak. Pengembang hanya memberitahu kapan kamera harus memulai atau berhenti menangkap sebuah gambar. lalu *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 tunggal berisi algoritma visi komputer yang mendeteksi dan melacak benda di dunia nyata kedalam frame video kamera. Berdasarkan gambar kamera, algoritma yang berbeda mendeteksi target baru atau marker, dan memeriksa tombol virtual. Hasilnya disimpan dalam sebuah *state object* yang digunakan oleh *renderer* video background dan dapat diakses dari kode aplikasi. Itu pelacak dapat memuat beberapa set data, tetapi hanya satu saja yang dapat aktif dalam satu waktu.

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

Pengembang aplikasi harus menginisialisasi semua komponen di atas dan melakukan 3 kunci langkah dalam kode aplikasi. Untuk setiap *frame* yang diproses, *state object* akan diperbarui dan aplikasi akan membuat metode yang akan dipanggil. Pengembang aplikasi harus :

- Mengantrikan *state object* untuk pendeteksian target yang baru, marker atau memperbarui status dari elemen.
- Memperbaharui logika aplikasi dengan input data yang baru.
- 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 dikompile 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 ditunjukan untuk membuat game. Unity 3D merupakan *game engine* multiplatform yang mampu di-*publish* secara *standalone* (.exe), berbasis web, android, Ios, XBOX, maupun PS3, dengan catatan mendapat lisensi.

Kelebihan Unity dibanding dengan *game engine* lainya 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**

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

2. **Scripting**

   Scripting yang digunakan pada Unity dibangun menggunakan Mono 2.6. Mono 2.6 merupakan implementasi open source dari .Net Framework. Bahasa pemrograman yang didukung oleh Unity 3D, 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**

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

2.2.5 Blender 3D

Blender 3D adalah perangkat lunak visualisasi 3D yang mempunyai fitur yang cukup lengkap, dan populer. Software ini bersifat *Open Source* atau gratis, kualitas pencitraan digital tidak kalah dengan software-software grafis 3D lainnya seperti 3DS Max. Blender 3D dapat digunakan untuk membuat animasi 3D dan ada fitur tambahan yang membuat software ini semakin menarik yaitu bisa membuat sebuah game dengan game engine yang ada pada software ini.

2.3 Stimulasi Bayi

Stimulasi adalah adalah rangsangan yang dilakukan sejak bayi baru lahir (bahkan sebaiknya sejak di dalam kandungan) dilakukan setiap hari, untuk merangsang semua sistem indera (pendengaran, penglihatan, perabaan, pembauan, pengecapan). Selain itu harus pula merangsang gerak kasar dan halus kaki, tangan dan jari-jari, mengajak berkomunikasi, serta merangsang perasaan yang menyenangkan bayi dan anak-anak (Herawati 2011) [15].

Ketika dilahirkan, otak anak sudah mempunyai sel syaraf yang bermilyaran jumlahnya, namun jumlah itu banyak yang hilang setelah dilahirkan. Ketika otak mendapatkan suatu stimulus yang baru, maka otak 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. Di sinilah pentingnya suatu stimulasi yang rutin diberikan. 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 bahkan tidak mendapat stimulasi. 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 0-12 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	- Mengangkatkan kepala - Berguling-guling - Menahan kepala tetap tegak	- Melihat, meraih dan menendang mainan gantung - Memperhatikan benda bergerak - Memegang benda - Meraba dan merasakan bentuk permukaan	- Berbicara - Menirukan suara - Mengenali berbagai suara	- Memberi rasa aman dan kasih sayang - Mengajak bayi tersenyum - Meniru ocehan dan mimik muka bayi - Mengayun bayi - Meninabobokan
3 – 6 Bulan	- Lanjutkan stimulasi usia 0-3 Bulan - Menyangga berat - Mengembangkan kontrol terhadap kepala	- Lanjutkan stimulasi usia 0-3 bulan - Memegang benda dengan kuat - Memegang dengan kedua tangan	- Lanjutkan stimulasi 0-3 bulan - Mencari sumber suara - Menirukan kata-kata	- Lanjutkan stimulasi usia 0-3 bulan - Bermain ci-luk-ba - Melihat dirinya dikaca - Berusaha meraih mainan

	- Duduk	- Makan sendiri - Mengambil benda kecil		
6 – 9 Bulan	- Lanjutkan stimulasi usia 3-6 bulan - Merangkak - Menarik ke posisi berdiri - Berjalan berpegangan - Berjalan dengan bantuan	- 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	- Lanjutkan stimulasi usia 3-6 bulan - Menyebutkan nama gambar dibuku - Menunjuk dan menyebutkan nama gambar-gambar	- Lanjutkan stimulasi usia 3-6 bulan - Permainan bersosialisasi

9 – 12 Bulan	- Lanjutkan stimulasi usia 6-9 bulan - Bermain bola - Membungkuk - Berjalan sendiri - Naik tangga	- Lanjutkan stimulasi usia 6-9 bulan - Menyusun balok-balok - Menggambar - Bermain di dapur	- Lanjutkan stimulasi usia 6-9 bulan - Menirukan kata-kata - Berbicara dengan boneka - Bersenandung dan bernyanyi	- Lanjutkan stimulasi usia 6-9 bulan - Minum sendiri dari sebuah cangkir - Makan bersama-sama - Menarik mainan yang letaknya agak jauh

2.4 Android

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

Google Inc. sepenuhnya membangun Android dan menjadikannya bersifat terbuka (*Open Source*) sehingga para pengembang dapat menggunakan Android tanpa perlu mengeluarkan biaya untuk mendapatkan lisensi dari Google dan dapat membangung Android tanpa adanya batasan-batasan. Android Software Development Kit (SDK) menyediakan alat dan Application Programming Interface (API) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. [9]

Hingga saat ini, terdapat beberapa versi dari sistem operasi Android, antara lain:

1. Android versi 1.1
2. Android versi 1.5 (Cupcake)
3. Android versi 1.6 (Donut)
4. Android versi 2.0/2.1 (Eclair)
5. Android versi 2.2 (Froyo: Frozen Yoghurt)
6. Android versi 2.3 (Gingerbread)
7. Android versi 3.0/3.1 (Honeycomb)
8. Android versi 4.0 (ICS: Ice Cream Sandwich)
9. Android versi 4.1 - 4.3 (Jelly Bean)
10. Android versi 4.4 (KitKat)
11. Android versi 5.0 (Lollipop)
12. Android versi 6.0 (Marshmallow)

Versi Android yang penulis sarankan untuk penggunaan aplikasi maupun dalam pembuatan aplikasi dengan minimal versi Android 2.3.1 (Gingerbread). Bagian ini diperlukan untuk Analisis Kebutuhan Fungsional pada Bab 3 sebagai kebutuhan perangkat lunak.

2.5 Android SDK Manager

Android Software Development Kit Manager adalah salah satu emulator Android. SDK Manager dapat menjalankan aplikasi-aplikasi yang terdapat pada Android karena bersifat emulator. Android SDK juga merupakan *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis Google 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 mendokumentasikan, menspesifikasikan, dan membangun 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 membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut bisa berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan 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]

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (message) antar objects.
4. *Collaboration Diagram* untuk memodelkan interaksi antar objects.
5. *State Diagram* untuk memodelkan perilaku objects di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku Use Case dan objects di dalam system.
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur object.
9. *Component Diagram* untuk memodelkan komponen object.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

Berikut adalah UML yang penulis gunakan dalam penelitian ini yaitu use case diagram, sequence diagram, collaboration diagram dan class diagram.

1. Use Case Diagram

Use case diagram digunakan untuk memodelkan semua bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambar-kan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.

2. Sequence Diagram

Diagram Class dan diagram Object merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti *Diagram Interaction*. *Diagram sequence* merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-objek yang

berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

3. Class Diagram

Class diagram menggambarkan struktur statis class di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu class tergantung/menggunakan *class* yang lain), *specialed* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (group bersama sebagai satu unit). sebuah sistem biasanya mempunyai beberapa *class* diagram.

4. Collaboration Diagram

Diagram *Collaboration* juga merupakan diagram *interaction*. Diagram membawa informasi yang sama dengan diagram *Sequence*, tetapi lebih memusatkan atau memfokuskan pada kegiatan obyek dari waktu pesan itu dikirimkan. Setiap pesan pada diagram *Collaboration* mempunyai angka yang terurut. Pesan yang tingkatannya tertinggi adalah angka 1. Pesan yang berada pada tingkat yang sama memiliki prefix yang sama, namun suffix berbeda bergantung pada posisinya; hanya untuk angka 1, 2, dan seterusnya.

2.7 Alpha dan Beta Testing

Tujuan Alpha Testing atau pengujian alpha adalah untuk mengidentifikasi dan menghilangkan sebanyak mungkin masalah sebelum akhirnya sampai ke *user*, serta dilakukan setelah *software* selesai oleh pengembang. Sedangkan Pengujian beta atau beta testing adalah istilah yang digunakan untuk menggambarkan proses pengujian eksternal di mana *software* tersebut dikirim ke kelompok atau memilih pelanggan yang menggunakannya dalam lingkungan 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 secara virtual, meramalkan bagaimana pengguna atau pelanggan akan menggunakan program. Instruksi-instruksi yang dapat digunakan mungkin akan diintepresentasikan dengan salah, kombinasi-kombinasi data yang aneh

diluar kebiasaan, keluaran yang kelihatannya jelas bagi tester mungkin akan tidak demikian halnya bagi pengguna.

2.7.1 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 ini, *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 untuk melihat apakah program aplikasi menghasilkan output yang diinginkan atau tidak.

2.8 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 rataan, 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 tidak terdiri dari hanya satu stimulus atau satu pernyataan saja melainkan selalu berisi banyak item (*multiple item measure*) [21].

Adapun rumusan untuk mendapatkan hasil dari skala *likert* adalah sebagai berikut:

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

Keterangan:

P = Nilai persentase yang dicari.

S = Jumlah frekuensi jawaban dikali dengan skala jawaban.

Skorideal = Skala tertinggi jawaban dikalikan dengan jumlah *sample*.