BAB 2
TINJAUAN PUSTAKA

2.1 Wolf Guitar Master

Wolf Guitar Master adalah tempat pembuatan gitar handmade lokal, dengan tangan kreatif dari *one man luthier (satu orang pengrajin)* tanpa campur tangan orang lain menjadi kelebihan tersendiri sehingga gitar yang diproduksi menjadi berkarakter khas.



Gambar 2.1 Logo Wolf Guitar Master

2.1.1 Sejarah Wolf Guitar Master

Wolf Guitar Master berdiri di kota bandung pada tanggal 18 September 2013 tepatnya di Sarijadi Blok 18 No. 36. Peralatan yang seadanya mampu menghasilkan kualitas yang memukau karena Wolf Guitar Master memiliki *Luthier* yang berpengalaman yaitu Wawan yang akrab disapa Apuy hampir 35 tahun di dunia gitar dan murid dari mastro *Luthier* di bandung yang pernah melambungkan nama *custom* gitar ternama di bandung SECCO dan GENTA yaitu Alm, Ki Anong Naeni.

Wolf Guitar Master dan Apuy sekarang berdiri dengan kerabat-kerabatnya dan membentuk perusahaan *custom* gitar yang mengandalkan kualitas dan mengutamakan *Tone* dan *Playability* serta menyediakan aksesoris sesuai dengan permintaan pelanggan, dari bentuk *inlay, binding, softcase, hardcase, soundboard, soundhole*, dll. Kayu yang digunakan adalah kayu pilihan dan berkualitas karena Wolf Guitar Master ingin menghasilkan gitar dengan suara

yang baik, memiliki nilai artistik dan terlebihnya bisa membawa musisi indonesia untuk berkarya yang lebih baik.

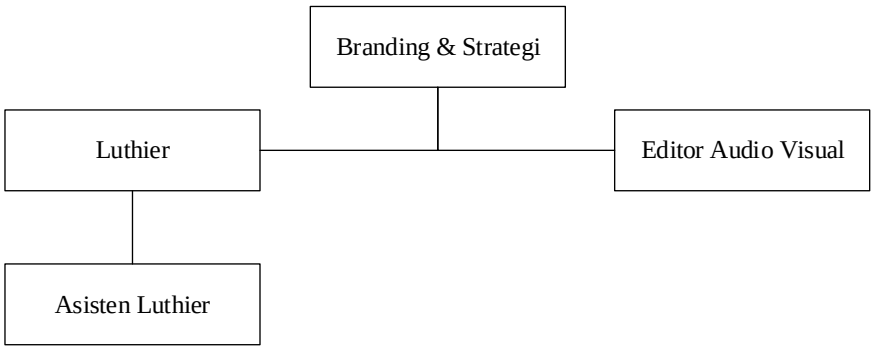
2.1.2 Visi dan Misi Wolf Guitar Master

Visi Wolf Guitar Master adalah menjadi pelopor gitar berbahan kayu solid di indonesia dan dapat bersaing dengan produk internasional.

Misi Wolf Guitar Master adalah pengenalan gitar berbahan kayu solid dan pengenalan *branding* di indonesia.

2.1.3 Struktur Organisasi dan Deskripsi Pekerjaan

Struktur organisasi dan deskripsi pekerjaan di wolf guitar master



Gambar 2.2 Struktur Organisasi Wolf Guitar Master

Penjelasan deskripsi pekerjaan setiap bagian dari struktur organisasi Wolf Guitar Master.

1. Branding & Strategi

   Berperan dalam bagian marketing, pemasaran, desain produk, dan mengatur produksi.

2. Luthier

   Pengrajin gitar utama, membuat gitar sesuai dengan model yang telah di buat oleh bagian branding & strategi

3. Editor Audio Visual

Berperan sebagai dokumentasi dari produksi gitar *custom.*

4. Asisten Luthier

Berperan sebagai pembantu luthier utama atau sebagai pengganti sementara luthier utama.

2.2 Web3D

Di dalam jurnal Bentar Priyopradono dan Fajar Antariksa Perdana tentang Web3D untuk desain interior, Web3D pada dasarnya memiliki konsep konten 3D interaktif yang ditampilkan melalui web [2]. Istilah web3D dimaksudkan dapat mengeksplorasi konten web dalam bentuk tiga dimensi (3D), Web3D merupakan desain dari situs *web* dalam bentuk *virtual* 3D yang digunakan untuk segala hal yang dapat dilihat dari tiga sisi yaitu, sisi X, Y, dan Z secara bersamaan. Setiap pengguna dapat melihat objek dengan jelas dari setiap sisi [2]. Objek dapat bergerak dikendalikan oleh pengguna menggunakan mouse dan keyboard. Web3D pada awalnya merupakan ide untuk menampilkan dan menavigasi situs *web* mengunakan grafis 3D, namun istilah ini sekarang digunakan untuk semua konten 3D interaktif yang ditampilkan dalam web html yang dapat dilihat dan disajikan melalui *web browser*. Teknologi Web3D membutuhkan Web3D *viewer* untuk dapat menampilkan konten 3D, contohnya adalah WebGL (*Web-based Graphics Library*) yang berupa javascript API untuk rendering grafis 2D dan 3D secara interaktif [2].

2.3 WebGL

WebGL (*Web-based Graphics Library*) merupakan sebuah platform *Application Programming Interfaces (APIs)* library 3D grafis yang memungkinkan internet browser untuk membuat bentuk 3D dengan cara yang sederhana dan efisien [2]. *WebGL* pertama kali diperkenalkan oleh Vladimir Vukicevic seorang *software engineer* pada tahun 2007, pada tahun 2009 standarisasi *WebGL* mulai dikembangkan oleh Khronos Group dan Opera, dan mulai diikuti oleh Google (Chrome), Mozila (Firefox), Apple (Safari), dan para

pengembang 3D. *WebGL* menggunakan pendekatan rendering berbasis klien dimana elemen bagian-bagian dari suatu kesatuan model 3 dimensi di download dari *server*, semua proses yang diperlukan untuk memproses gambar dilakukan secara lokal menggunakan *hardware* grafis klien [2]. *WebGL* dikembangkan untuk mengatasi beberapa masalah, sebagian besar aplikasi *Web3D* yang tersedia membutuhkan *plugin*, sseperti flash3D, Silverlight, Java3D atau perangkat lunak tambahan yang dapat menjalankan konten 3D di *web browser*, untuk menjawab permasalahan ini, teknologi *WebGL* menyediakan sarana pemrosesan grafis langsung dari *web browser* tanpa harus menggunakan *plugin* [2].

Dibandingkan dengan beberapa teknologi seperti Flash3D, Silverlight, Java3D, dan yang lainnya, *WebGL* memiliki beberapa keuntungan diantaranya [2]:

1. *WebGL* diprogram menggunakan javascript, hal ini memudahkan dalam integrasi aplikasi *WebGL* dengan library javascript seperti jquery dan teknologi HTML5 lainnya.
2. Tidak seperti *OpenGL* dan teknologi yang sejenis, dimana adanya operasi khusus dalam mengalokasikan dan mengkosongkan memori secara manual. *WebGL* secara otomatis akan mengosongkan memori yang sudah tidak dipakai, sehingga mempermudah pemrogramman, mengurangi kode yang dibutuhkan dan mudah dipahami.
3. *WebGL* dapat diimplementasikan pada platform Android.
4. Kinerja aplikasi *WebGL* setara dengan aplikasi *standalone*, karena kemampuan dari *WebGL* yang dapat mengakses *hardware* grafis lokal.

WebGL adalah platform API yang membawa OpenGL ES 2.0 ke *web* sebagai konten 3D dalam konteks HTML. *WebGL* menggunakan bahasa OpenGL, GLSL ES dan dapat dikombinasikan dengan konten web. *WebGL* sangat cocok untuk *Web3D* secara dinamis menggunakan bahasa pemrogramman javascript. Beberapa library *WebGL* yang dapat digunakan adalah [2]:

1. CopperLicht
2. Three.js
3. Processing.js
4. OSG.js
5. GLGE
6. Cessium
7. Babylon.js
8. A-Frame

2.4 Multimedia

Menurut Robin dan Linda “multimedia merupakan alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio, dan video” [3].

Terdapat komponen penting dalam multimedia, yaitu [3]:

1. Harus ada komputer atau perangkat yang mengkoordinasikan apa yang dilihat dan dindengarkan yang dapat berinteraksi dengan penggunanya.
2. Harus adanya keterkaitan yang menghubungkan informasi dengan pengguna dengan baik.
3. Harus adanya navigasi yang memandu pengguna dalam penggunaannya.
4. Multimedia dapat menyediakan tempat untuk mengumpulkan, memproses, serta mengkomunikasikan informasi ke pengguna.

2.4.1 Objek Multimedia

Menurut Sutopo, objek-objek multimedia adalah sebagai berikut [4]:

1. Teks

   Teks adalah data atau informasi dalam bentuk susunan-susunan karakter. Sebagian besar multimedia menggunakan tesks karena sangat efektif untuk menyampaikan ide kepada pengguna.

2. Gambar

   Gambar sangat bermanfaat untuk mengilustrasikan data atau informasi yang akan disampaikan, terutama suatu informasi yang sulit dijelaskan melalui kata-kata .

3. Animasi

   Animasi merupakan simulasi gerakan yang dihasilkan dengan menggerakan sejumlah gambar dari frame ke frame yang lain. Frame adalah gambar tunggal pada sejumlah rentetan gambar yang membentuk animasi.

4. Video

   Video menyediakan informasi yang kaya dan hidup untuk multimedia. Video dapat digunakan untuk menjelaskan sesuatu hal yang sangat sulit dijelaskan sama halnya dengan gambar.

5. Audio

   Multimedia akan cepat bosan jika tidak tersedianya audio (suara). Suara disini bisa berbentuk percakapan, musik, atau efek yang digunakan untuk menunjang multimedia yang lain.

2.5 Basis Data

Dalam buku Abdul Kadir tentang konsep dan tuntunan praktis basis data, Chou mendefinisikan basis data sebagai kumpulan informasi bermanfaat yang diorganisasikan ke dalam tatacara yang khusus [5]. Menurut Fabbri dan Schwab basis data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data [5]. Menurut Date basis data dapat dianggap sebagai tempat untuk sekumpulan data terkomputerisasi dan sistem basis data pada dasarnya adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan.

Secara sederhana basis data dapat diartikan sebagai kumpulan informasi yang telah disusun rapi sedemikan rupa guna meminimalisir pengulangan informasi [5].

Penggunaan sistem basis data bisa melakukan operasi, antara lain [5]:

1. Menambahkan file baru ke sistem basis data.
2. Mengosongkan berkas.
3. Menyisipkan data ke suatu berkas.
4. Mengambil data yang ada pada suatu berkas.
5. Mengubah data pada suatu berkas.
6. Menghapus data pada suatu berkas.
7. Menyajikan suatu informasi yang diambil dari sejumlah berkas.

2.6 DBMS

Menurut Abdul Kadir DBMS (*Database Management System*) dapat diartikan sebagai suatu program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memanipulasi, dan memperoleh data atau informasi dengan praktis dan efisien [5]. Sedangkan menurut Rosa A.S dan M. Shalahuddin DBMS adalah suatu sistem aplikasi yang dapat digunakan untuk menyimpan, mengelola, dan menampilkan data [6]. Suatu sistem Aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

1. Menyediakan fasilitas untuk mengelola akses data.
2. Mampu menangani integritas data.
3. Mampu menangani akses data.
4. Mampu menangani *backup* ata.

DBMS memiliki 4 keunggulan yaitu:

1. Kepraktisan

   Sistem yang berbasis kertas akan menggunakan kertas yang sangat banyak untuk menyimpan informasi, sedangkan DBMS menggunakan media penyimpanan sekunder yang berukuran kecil tetapi padat informasi.

2. Kecepatan

   Mesin dapat mengambil atau mengubah data jauh lebih cepat daripada manusia.

3. Mengurangi kejemuan

   Orang cenderung menjadi bosan jika melakukan tindakan-tindakan berulang yang menggunakan tangan.

4. Kekinian

   Informasi yang tersedia pada DBMS akan bersifat mutakhir dan akurat setiap saat.

DBMS saat ini telah menyediakan pengaturan keamanan terhadap basis data berdasarkan. Keamanan data sangat penting pada basis data, yaitu untuk menghindari pengaksesan data-data sensitif yang bersifat rahasia oleh orang yang tidak memiliki hak atau wewenang [5].

Beberapa macam DBMS versi komersial dan *open source* yang paling banyak digunakan, yaitu:

1. Oracle
2. Microsoft SQL Server
3. MySQL
4. PostgresSQL
5. SQLlite

2.7 SQL

SQL atau *Structured Query Languange* merupakan bahasa yang digunakan untuk mengelola data mulai dari menambahkan, mengubah, menghapus, dan menyediakan data pada sistem manajemen basis data (DBMS) [5]. Contoh pengaksesan data pada sistem manajemen basis data (DBMS) dengan bahasa SQL, adalah sebagai berikut [5]:

1. Memasukkan data
2. Mengubah data
3. Menghapus data
4. Menampilkan data

2.8 ERD

ERD (*Entity Relationship Diagram*) merupakan rancangan pemodelan awal basis data. ERD (*Entity Relationship Diagram*) digunakan untuk pemodelan basis data relasional [5].



Gambar 2.3 Contoh *Entity Relationship Diagram*

2.9 Pemrogramman Berorientasi Objek

Menurut Rosa A.S. dan M. Shalahuddin dalam buku Rekayasa Perangkat Lunak, Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya [6]. Metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti aplikasi yang dikembangkan untuk *platform* yang berbeda-beda, sehingga dibutuhkan metodologi pengembangan yand dapat mencakup semua jenis aplikasi. Ada beberapa keuntungan dengan menggunakan metodologi berorientasi objek, yaitu [6]:

1. Meningkatkan produktivitas
2. Kecepatan pengembangan
3. Kemudahan pemeliharaan
4. Konsisten
5. Meningkatkan kualitas perangkat lunak

2.9.1 Konsep Dasar Berorientasi Objek

Pendekatan berorientasi objek adalah teknik dalam memandang suatu permasalahan yang ada dalam membangun suatu sistem. Pendekatan berorientasi objek akan melihat sistem sebagai sekumpulan objek yang merepresentasikan objek-objek yang ada di dunia nyata [6].

Konsep pendekatan berorientasi diterapkan dalam rekayasa perangkat lunak pada tahap analisis, perancangan, pemrogramman, dan pengujian perangkat lunak. Ada berbagai macam teknik yang dapat digunakan pada tahapan-tahapan tersebut sesuai dengan aturan dan alat bantu pemodelan tertentu [6].

Beberapa konsep dasar berorientasi objek yang harus dipahami adalah [6]:

1. *Class* (kelas)

   Kelas adalah kumpulan objek yang memilika karakteristik yang sama. Sebuah kelas memiliki sifat (atribut), fungsi (metode/operasi), dan hubungan. Suatu kelas dapat diturunkan ke kelas yang lain, dimana atribut yang ada pada kelas tersebut dapat di wariskan ke kelas yang baru.

2. *Object* (Objek)

   Objek meruapakan abstraksi dari sesuatu yang mewakili dunia nyata seperti benda, manusia, tempat, atau hal-hal lain yang sifatnya abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi, dan memiliki fungsi yang dapat digunakan untuk mengakses informasi tersebut.

3. *Method* (Metode)

   Metode atau fungsi pada kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas bisa memiliki lebih dari satu metode atau fungsi. Metode atau fungsi yang dapat digunakan untuk memanipulasi informasi yang ada pada objek itu sendiri.

Metode atau fungsi dapat berasal dari suatu *event* (kejadian atau persitiwa), aktivitas, atau keadaan di dunia nyata. Kelas sebaiknya memiliki metode *get* dan *set* untuk setiap atribut agar konsep enkapsulasi tetap terjaga. Metode atau fungsi *get* digunakan untuk mengakses nilai dari suatu atribut yang ada di dalam objek, dan *set* digunakan untuk memanipulasi nilai yang dari suatu atribut yang ada di dalam objek. Metode atau fungsi *get* dan *set* digunakan agar kelas yang lain tidak dapat mengakses atribut yang ada secara langsung.

4. *Attribute* (Attribute)

   Atribut variabel global yang dimiliki oleh suatu kelas. Atribut dapat berupa nilai atau elemen-elemen data yand dimiliki objek dalam kelas objek.

5. *Abstraction* (Abstraksi)

   Abstraksi merupakan prinsip yang merepresentasikan suatu hal yang komplesk di dunia nyata yang menjadikannya satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. *Encapsulation* (Enkapsulasi)

   Pembungkus atribut dan metode yang dimiliki suatu kelas atau objek untuk menyembunyikan nilai atau cara kerja yang ada di dalam kelas tersebut agar kelas lain tidak mengetahui nilai suatu atribut atau metode yang ada di dalam kelas.

7. *Inheritance* (Pewarisan)

   Mekanisme yang memungkinkan suatu objek mewarisi sebagian atau seluruh atribut maupun metode dari objek lain sebagai bagian dari objek tersebut.

8. *Interface* (Antarmuka)

   Antarmuka mirip dengak kelas, hanya saja di dalam kelas *interface* tidak adanya atribut hanya memiliki motode atau fungsi yang dideklarasikan tanpa isi. Deklarasi metode atau fungsi yang ada di dalam kelas *interface* dapat diimplementasikan oleh kelas lain. Sebuah kelas dapat mengimplementasikan lebih dari satu *interface* dimana kelas yang mengimplementasikan *interface* tersebut akan mendeklarasikan kembali metode yang ada pada *interface* yang diimplementasi sekaligus mendefinisikan isi dari metode tersebut.

9. *Reusability*

   Pemanfaatak kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yand melibatkan objek tersebut.

10. Generalisasi dan Spesialisai

    Hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Mekanisme generalisasi dan spesialisasi sama dengan mekanisme pewarisan (*inheritance*).

11. Komunikasi Antar Objek

    Komunikasi antar objek dilakukan melalui pesan yang dikirim dari satu objek ke objek lainnya.

12. *Polymorphism* (Polimorfisme)

    Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

13. *Package*

    Package merupakan sebuah wadah atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa

kelas yang memiliki nama yang sama dapat disimpan di dalam *package* yang berbeda untuk menghindari ambigu.

2.10 UML

UML (*Unified Modeling Language*) merupakan teknik pemrogramman berorientasi objek sebagai salah satu bentuk pemodelan dalam pembangunan perangkat lunak. UML muncul karena adanya kebutuhan pemodelan visual untuk menggambarkan, membangun, dan dokumentasi dalam pembangunan sistem perangkat lunak secara spesifik. UML merupakan bahasa untuk pemodelan dan komunikasi di dalam sistem yang menggunakan diagram dan teks pendukung [6].

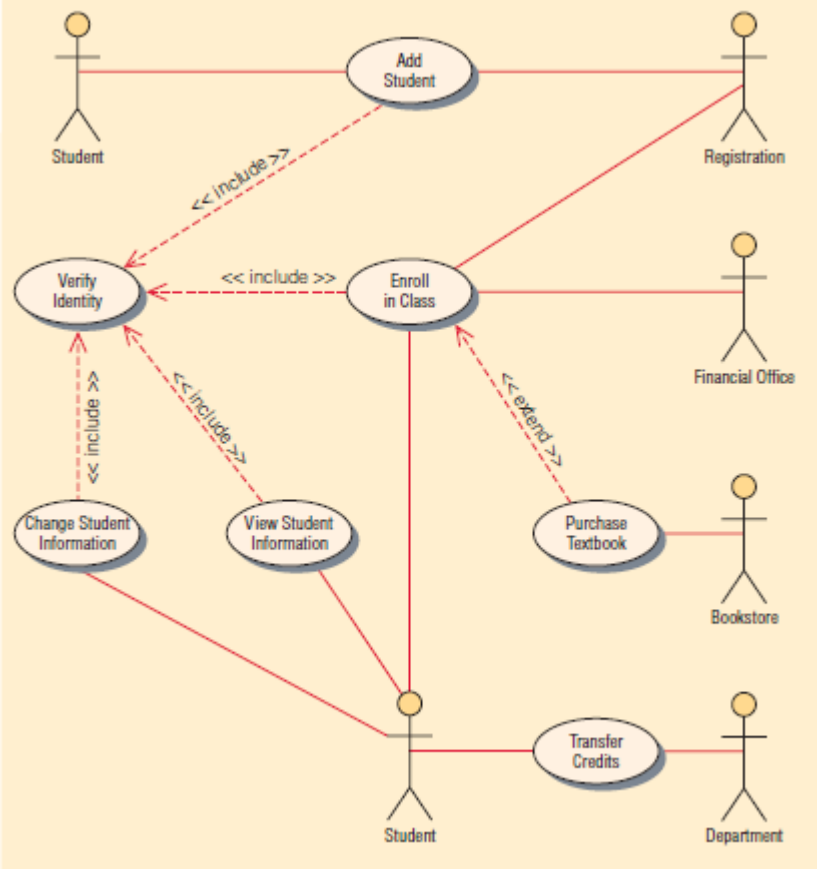
Ada 6 model yang sering digunakan dalam UML yaitu:

1. *Use Case Diagram*
2. *Use Case Scenario*
3. *Activiy Diagram*
4. *Sequence Diagram*
5. *Class Diagram*

2.10.1 Use Case Diagram

Diagram *use case* merupakan pemodelan untuk suatu aktivitas sistem yang akan dibuat. *Use case diagram* sealalu menjelaskan 3 hal, pertama adalah aktor menginisialisasi *event*, kedua *event* tersebut memicu *trigger use case*, dan ketiga *use ase* melakukan tindakan yang dipicu oleh *event* [7].

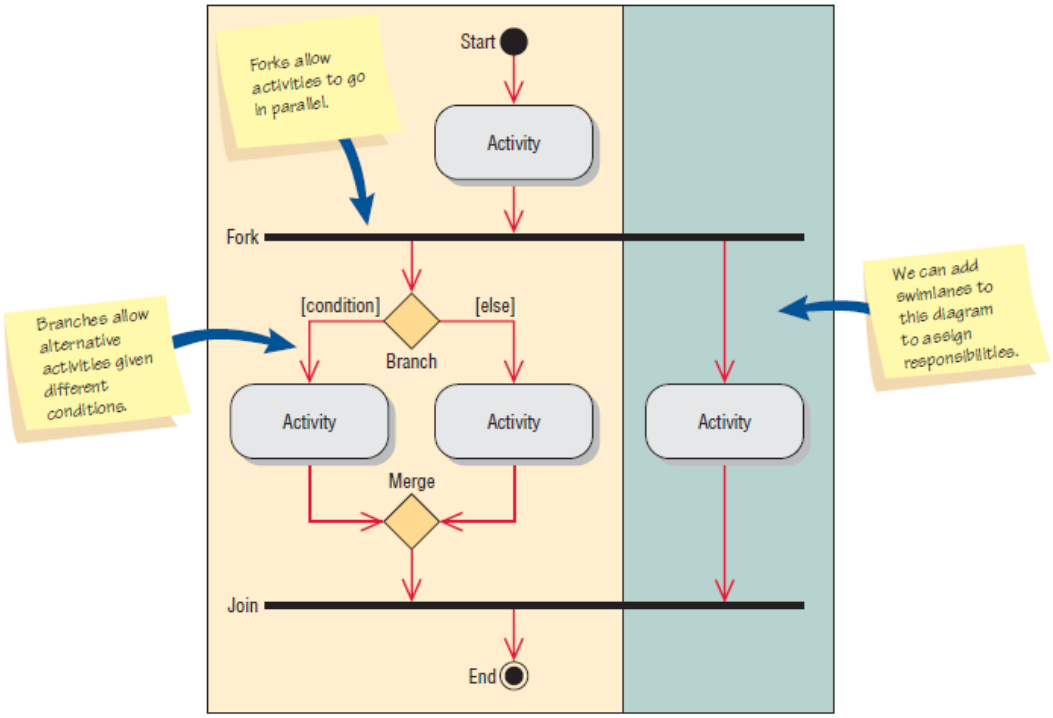
Penamaan *use case* didefinisikan sesederhana mungkin dan dapat dipahami hal utama yang ada pada *use case diagram* adalah aktor dan *use case*.



Gambar 2.4 Contoh *Use Case Diagram*

2.10.2 Activity Diagram

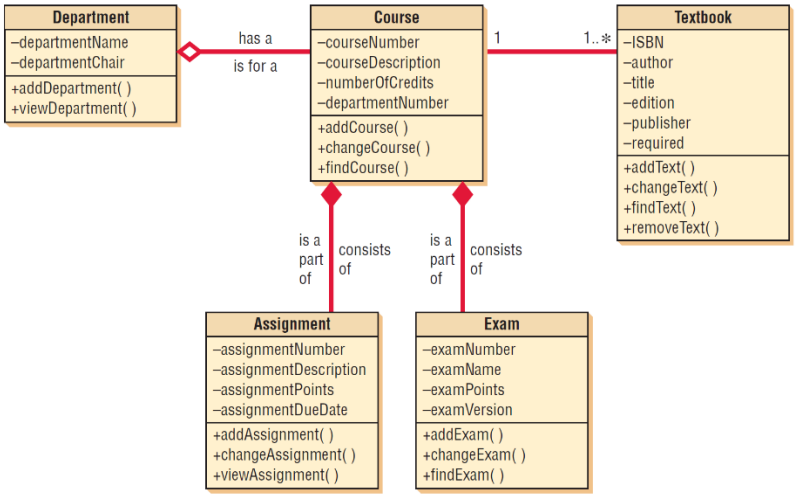
Penggunaan *activity diagram* dilakukan untuk membantu memahami *use case*, membaca alur kontrol yang komplesk, maka perlu memodelkan alur proses yang terjadi [7].



Gambar 2.5 Contoh *Activity Diagram*

2.10.3 Class Diagram

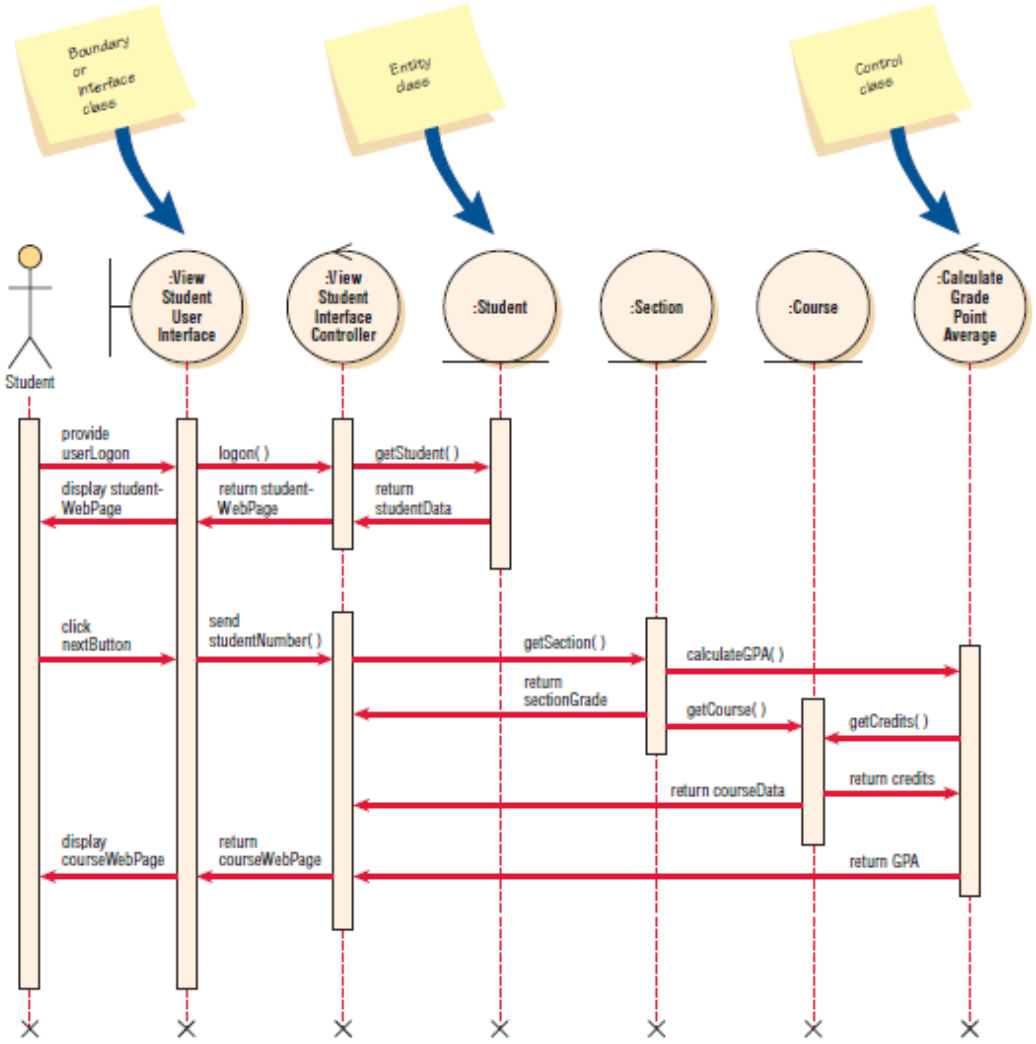
Class diagram berfungsi untuk menujukkan fitur statis dari sistem dan tidak mewakili suatu proses tertentu, *class diagram* digunakan untuk menunjukan sifat dari hubungan antar kelas [7].



Gambar 2.6 Contoh *Class Diagram*

2.10.4 Sequence Diagram

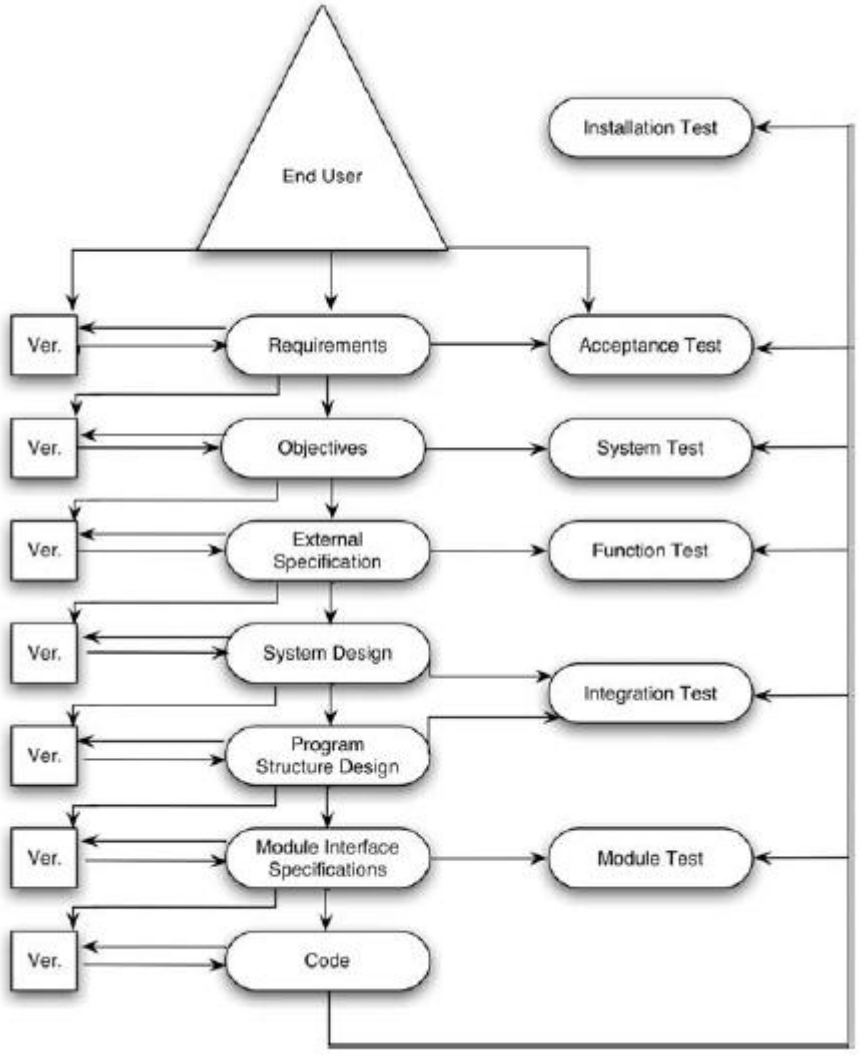
Sequence diagram dapat menggambarkan perilaku objek pada *use case* dengan mendeksripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. *Sequence diagram* biasanya diawali dengan menunjukkan aktor dan kelas yang berinteraksi untuk proses tertentu. dalam *sequence diagram* terdapat beberapa simbol yaitu simbol aktor, *lifeline* dan *object* [7].



Gambar 2.7 Contoh *Sequence Diagram*

2.11 Pengujian Sistem

Pengujian sistem digunakan sebagai tahap akhir sebagai syarat apakah sistem yang dibangun sudah sesuai dengan kebutuhan atau belum. Menurut Myers, *software testing* merupakan suatu proses atau tahapan proses yang dibuat untuk memastikan kode-kode yang digunakan dalam pembangunan sistem berjalan sebagaimana mestinya, juga sebalikya, agar sistem tidak terjadi hal-hal yang tidak diinginkan [8].



Gambar 2.8 *High Order Testing*

Banyak proses pengujian yang dapat dilakukan untuk pengujian sistem, tetapi tidak semua pengujian dilakukan, hanya beberapa pengujian saja yang dilakukan sesuai dengan kebutuhan. Proses pengujian yang biasa dilakukan yaitu *functional testing* dan *acceptance testing (beta testing)*. Ada pula yang menambahkan pengujian yang lain seperti *performance testing*.

2.11.1 Teknik Pengujian Sistem

Pada dasarnya untuk melakukan pengujian sistem terdapat 2 teknik yang dapat digunakan yaitu *black box* dan *white box*

1. *Black box testing* adalah teknik pengujian yang mengabaikan mekanisme internal suatu sistem atau komponen dan hanya berfokus pada keluaran yang dihasilkan sebagai respon terhadap input dan kondisi eksekusi yang dipilih [9].
2. *White box testing* adalah teknik pengujian yang memperhitungkan mekanisme internal suatu sistem atau komponen [9].

Testing Type	Specification	General Scope	Opacity	Who generally does it?
Unit	Low-Level Design Actual Code Structure	Small unit of code no larger than a class	White Box	Programmer who wrote code
Integration	Low-Level Design High-Level Design	Multiple classes	White Box Black Box	Programmers who wrote code
Functional	High Level Design	Whole product	Black Box	Independent tester
System	Requirements Analysis	Whole product in representative environments	Black Box	Independent tester
Acceptance	Requirements Analysis	Whole product in customer's environment	Black Box	Customer
Beta	Ad hoc	Whole product in customer's environment	Black box	Customer
Regression	Changed Documentation High-Level Design	Any of the above	Black Box White Box	Programmer(s) or independent testers

Gambar 2.9 *Software Testing Level*

Test ID	Description	Expected Results	Actual Results

Gambar 2.10 Format Teknik Pengujian *Black Box*

2.12 Snowball Sampling

Menurut Sugiyono *snowball sampling* adalah teknik penentuan sampel yang mula-mula jumlahnya kecil, kemudian membesar [10]. Ibarat bola salju yang menggelinding yang lama-lama menjadi besar. Dalam penentuan sampel, pertama-tama dipilih satu atau dua orang, tetapi karena dengan dua orang ini belum merasa lengkap terhadap data yang diberikan, maka peneliti mencari orang lain yang dipandang lebih tahu dan dapat melengkapi data yang diberikan oleh dua orang sebelumnya. Begitu seterusnya. Sehingga jumlah sample semakin banyak [10].

2.13 Skala Likert

Instrumen penelitian digunakan untuk mengukur nilai variabel yang akan diteliti. Instrumen penelitian yang dipakai adalah dalam bentuk kuesioner. Untuk menghasilkan data kuantitatif maka instrumen penelitian yang ada harus memiliki skala. Pada skala likert, bobot skala yang yang digunakan untuk setiap jawaban yang dihasilkan adalah sebagai berikut [11]:

Tabel 2.1 Bobot Skala Likert

Keterangan	Bobot
Sangat Setuju	5
Setuju	4
Ragu-ragu	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Untuk menghitung nilai dari jawaban kuesioner yang dihasilkan menggunakan rumus yaitu:

$$P = \frac{S}{\text{Skor Ideal}} X\ 100\%$$

Keterangan:

P : hasil persentase

S : jumlah dari nilai jawaban yang diberikan oleh responden

Skort Ideal : bobot nilai tertinggi dikalikan dengan jumlah sampel

2.14 Adobe Illustrator

Adobe illustrator merupakan aplikasi pengolah gambar yang dikembangkan oleh *Adobe Sistem* sebagai perusahan pengembang perangkat lunak untuk desain grafis. Adobe illustraotr digunakan untuk mengedit atau membuat desain-desain berbasis vektor [12].



Gambar 2.11 Logo Adobe Illustrator CS6

2.15 Blender

Blender merupakan aplikasi pembuat grafis 3D yang dapat diunduh secara gratis oleh umum. Fitur utama blender adalah untuk modeling 3d, *modelling, rigging, animation, simulation, rendering, conpositing, motion tracking,* dan *video editing* atau sebagai alat untuk membuat aset-aset *game* [13].



Gambar 2.12 Logo Blender

2.16 glTF

GL Transmission Format (GLTF) adalah format model 3D yang digunakan untuk menjembatani antara alat pembuatan konten 3D dan aplikasi 3D modern dengan menyediakan format yang efisien [14].

glTF didesain untuk tujuan sebagai berikut [14]:

1. *Compact file sizes*
2. *Fast loading*
3. *Runtime-independence*
4. *Complete 3d scene representation*
5. *Extensibility*



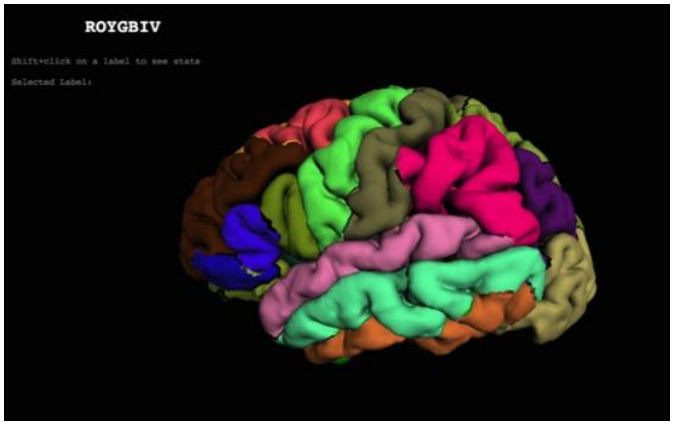
Gambar 2.13 Logo Format glTF

2.17 Three.js

Three.js merupakan javascript API yang digunakan ntuk membuat dan menampilkan animasi 3D pada web browser. *Coding* yang ada pada three.js dapat digunakan pada HTML5 menggunakan canvas, SVG, atau *WebGL* [15].

Three.js menggunakan WebGL yang mana merupakan javascript API untuk merender grafik 2D dan 3D interaktif pada web browser tanpa memerlukan plugin browser tambahan. Pada dasarnya API *WebGL* dapat digunakan secara

langsung untuk menampilkan model 3D. Three.js menyederhakanan struktur program yang ditulis terutama untuk pengembangan grafik 3D yang kompleks. Three.js dapat dijalankan di semua web browser yang telah mendukung *WebGL* [15].



Gambar 2.14 Contoh Pemanfaatan Three.js

2.18 Slim Framework

Slim Framework dibangun oleh Josh Lockhart, ia adalah seorang senior developer dari newmediacampaigns.com. Slim Framework dikatakan *micro framework* karena slim adalah *framework* yang fokus pada kebutuhan pokok yang diperlukan sebuah aplikasi web seperti menerima sebuah *HTTP request*, mengirimkan *request* tersebut ke kode yang sesuai, dan mengembalikan *HTTP response* [16].

Micro framework biasanya digunakan untuk proyek sekala kecil yang memiliki tujuan khusus dan tingkat kompleksitas yang rendah. Fitur-fitur yang ada pada Slim Framework adalah [16]:

1. *HTTP Router*
2. *Middleware*
3. *Dependency Injection*
4. *PSR-7 Support*



Gambar 2.15 Logo Slim Framework

2.19 GTmetrix

GTmetrix dibuat oleh GT.net sebagai alat untuk dilakukannya *performance testing* untuk aplikasi berbasis *web*. GTmetrix dapat memberikan rekomendasi untuk mengoptimasi performa *web* yang telah di tes. [17]