

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Dasar

Teori dasar adalah kumpulan definisi yang disusun rapi secara sistematis tentang variabel-variabel yang terdapat dalam suatu penelitian. Teori dasar yang dibahas merupakan ilmu-ilmu yang diteliti pada penelitian ini. Teori-teori yang digunakan adalah sebagai berikut:

##### 2.1.1 *Game*

Menurut Novaliendry (2013: 111-112), Kata “*Game*” berawal dari kata Bahasa Inggris yang berarti permainan. Permainan pada hal ini mengacu pada kelincahan intelektual. *Game* juga dapat didefinisikan sebagai arena keputusan dan aksi dari pemainnya dimana terdapat tujuan yang perlu diraih. *Game* juga dapat mempertajam kemampuan analisis pemainnya dalam pengolahan informasi dan pengambilan keputusan dengan cepat.

*Game* dapat di jalankan pada beberapa *platform* seperti:

1. *Disk Operating System* (DOS)

*Platform* ini sudah jarang digunakan karena terjadinya perpindahan sistem operasi ke *Windows*. Namun masih dapat ditemukan *resource* di internet yang memberi penjelasan serta programming pada DOS.

## 2. *Windows*

*Windows* memiliki *DirectX* yang di kembangkan *Microsoft* dimana *DirectX* mampu menghasilkan tampilan yang menarik pada *game*, *tools* dan *software* juga banyak tersedia untuk *platform* ini sehingga para pengembang banyak mengembang *game* untuk *Windows*.

## 3. *Linux*

Pengembang *game* untuk *platform* ini tidak banyak. Tetapi salah satu kelebihan dari *platform* ini adalah berbagai aplikasi tersedia secara gratis, walaupun komunitasnya kecil, berbagai *game* yang dibuat di *platform* ini tidak kalah dari *Windows*.

## 4. *Macintosh*

*Mac* atau *Macintosh* adalah sistem operasi yang dirilis *Apple*. *Macintosh* merupakan sistem operasi yang stabil dan memiliki berbagai fitur yang baik namun dari awal, *Mac* tidak berhubungan banyak dengan dunia *game*, tetapi *platform* ini memiliki fitur grafik dan suara yang lebih baik dari *Windows*.

## 5. *Console*

Persaingan *Windows* saat ini adalah *Xbox* dan *Playstation*. Pemrograman pada konsol tidak jauh berbeda dengan *Windows*. Namun tingkat kesulitan dalam mengembang *game* untuk *platform* ini adalah biaya peralatan dan lisensi yang relatif tinggi untuk menjaga mutu dan hak cipta.

## 6. *Mobile* atau *Handphone*

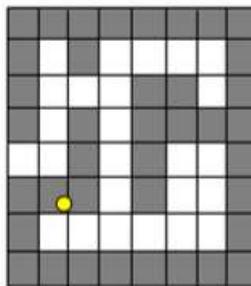
Sekarang sudah banyak *game* yang dibuat pada *platform* ini. *Game mobile* tidak secanggih *game* pada komputer dan konsol, namun dengan berbagai

kemudahan dan tersedianya berbagai peralatan, sebagian pengembang memilih untuk membuat *game* pada *platform* ini. *Platform* ini umumnya dirancang dengan menggunakan bahasa pemrograman Java.

*Game* dapat dibagi menjadi beberapa jenis atau biasanya disebut *genre*. Jenis-jenis *game* menurut Mahardhika & Dwi (2015: 10-11) yaitu :

1. *Maze Game*

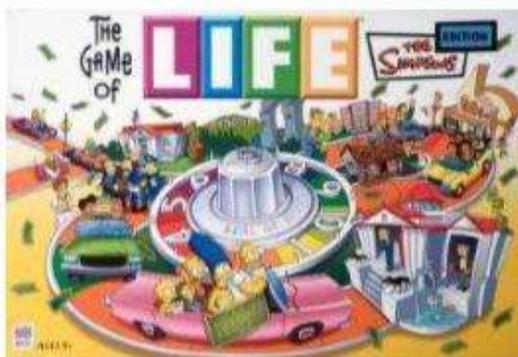
*Maze game* umumnya terdapat *maze* atau labirin yang dijadikan sebagai latar belakang *game* tersebut.



**Gambar 2.1** *Maze Game*  
**Sumber :** Astawa (2012)

2. *Board Game*

*Board game* mirip dengan *game* papan traditional seperti catur dan monopoli namun *game* ini berbentuk digital.



**Gambar 2.2** *Board Game*  
**Sumber :** Limantara, dkk (2015)

### 3. Card Game

*Card game* mirip dengan *game* kartu traditional namun, memiliki cara bermain yang lebih bervariasi dan berbentuk digital.



**Gambar 2.3** Card Game  
Sumber : Astuti, dkk (2013)

### 4. Battle Card Game

*Game* ini menggunakan kartu sebagai alat untuk berperang dengan kartu lainnya seperti *Battle Card Pokemon* dan *Yu-Gi-Oh!*

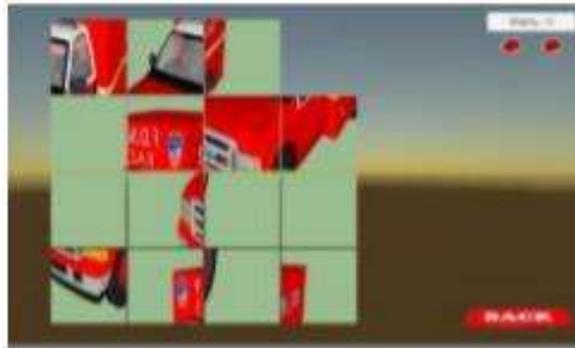
Kartu Monster			Kartu Sihir		
Kartu Aspek			Kartu Pertanyaan		
			Wah! yang menggunakan syring kulit sebagai media itu wah adalah ...	Serahkan tiga potongan masyarakat dalam struktur sosial maka akan di ...	Mengapa akan terlihat kegiatan dengan masyarakat di ...
			Perhatikan pada masa ini ...	Serahkan 7 ...	Mengapa akan terlihat kegiatan dengan masyarakat di ...

**Gambar 2.4** Battle Card Game  
Sumber : Roisa (2014)

### 5. Puzzle Game

*Game* ini merupakan permainan teka-teki dimana pengguna perlu berpikir secara logika untuk menyelesaikannya seperti *Tetris*, yang dimainkan

dengan cara menjatuhkan blok dan menghancurkan blok lain secara vertikal ataupun horizontal.



**Gambar 2.5** *Puzzle Game*  
**Sumber :** Prasetyo, dkk (2017)

#### 6. *Shoot Them Up*

*Shoot them up* umumnya berbentuk pesawat atau benda lain yang muncul dari atas, kiri ataupun kanan yang kemudian harus ditembak oleh pemain secepat dan sebanyak mungkin. Dulu permainan ini hanya dalam bentuk dua dimensi namun sekarang umumnya telah berbentuk tiga dimensi.



**Gambar 2.6** *Shoot Them Up*  
**Sumber :** Prasetyo, dkk (2017)

#### 7. *Side Scroller Game*

Saat pertama kali munculnya *game* jenis ini, pemain diwajibkan untuk bergerak searah secara horizontal dengan cara berjalan, merunduk

ataupun melompat untuk menghindari rintangan yang telah disediakan di jalan tersebut.



**Gambar 2.7** Side Scroller Game  
Sumber : Eko (2019)

#### 8. Fighting Game

*Fighting game* adalah *game* dimana pemain bertarung dengan musuh di dalamnya.



**Gambar 2.8** Fighting Game  
Sumber : Aisy, dkk (2017)

#### 9. Racing Game

*Game* ini umumnya berupa balap-balapan mobil, motor ataupun sepeda.



**Gambar 2.9** Racing Game  
Sumber : Suprayogo, dkk (2017)

#### 10. *Turn-Based Strategy Game*

Pada *game* ini, pemain mendapatkan giliran setelah gerakan pemain lain secara bergiliran seperti *game Civilization* dan *Empire*.

#### 11. *Real-Time Strategy Game*

*Game* ini mirip dengan *game turn-based strategy* tetapi di *game* ini para pemain melakukan gerakan pada saat yang sama dan tidak ada bergiliran.

Contoh *game* yang terkenal seperti *Mobile Legends*.

#### 12. *Simulation Game*

*Game* ini berbentuk simulasi dimana pemain dapat membangun rumah, kota ataupun menjalankan restoran. Contoh *game* seperti ini adalah *Sim City*, *Restaurant Story*.

#### 13. *First Person Shooter*

Di dalam *game* ini, perspektif pemain adalah sudut pandang orang pertama dan pemain melakukan baku tembak terhadap pemain lain.



**Gambar 2.10** First Person Shooter *Game*

**Sumber :** Singkoh, dkk (2016)

#### 14. *First Person Shooter 3D Vehicle Based*

*Game* ini mirip dengan *game first person shooter* namun, sudut pandang pemain bukan dari karakter di *game* tersebut, melainkan dari kendaraan yang dinaikinya.

#### 15. *Third Person 3D Game*

*Game* mirip dengan *game first person shooter* tetapi dengan sudut pandang orang ketiga.



**Gambar 2.11** Third Person 3D Game  
**Sumber :** Nugroho dan Khairani (2018)

#### 16. *Role Playing Game*

Pada *role playing game*, pemain memainkan sebuah karakter dimana pemain harus menjalankan alur cerita yang telah disediakan.



**Gambar 2.12** Role Playing Game  
**Sumber :** Abdi, dkk (2017)

### 17. *Adventure Game*

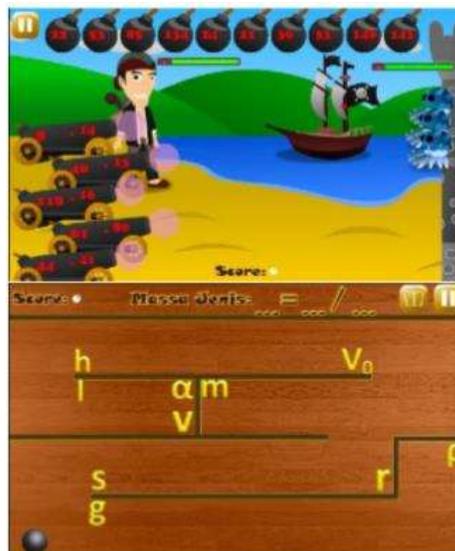
Di dalam *game* ini, pemain akan menjalankan sebuah petualangan dimana pemain akan mendapatkan benda sepanjang jalan yang akan berguna di sepanjang perjalanan berikutnya.



**Gambar 2.13** *Adventure Game*  
**Sumber :** Ridwan dan Prasetyawan (2017)

### 18. *Educational and Edutainment*

*Game* ini memiliki tujuan untuk menarik minat pemain untuk bermain sekaligus belajar.



**Gambar 2.14** *Education Game*  
**Sumber :** Nikensasi, dkk (2012)

### 19. Sport Game

*Game* ini menggunakan tema olahraga.



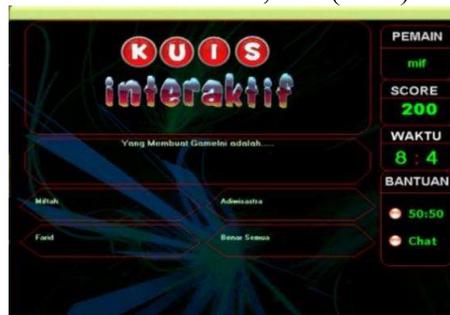
**Gambar 2.15** Sport Game  
Sumber : Nurhadiyan dan Arisona (2017)

### 20. Quiz Game

*Quiz Game* atau *game* kuis adalah *game* dalam bentuk kuis. Menurut Saputra & Rafiqin (2017:72) *game* kuis adalah suatu permainan dimana pemain diperlukan untuk menjawab pertanyaan yang benar.



**Gambar 2.16** Quiz Game  
Sumber : Arifin, dkk (2013)



**Gambar 2.17** Quiz Game  
Sumber : Adiwisastra (2015)



**Gambar 2.18** Quiz Game  
**Sumber :** Adiwijaya, dkk (2015)

### 2.1.2 Game Edukasi

Menurut Novaliendry (2013:112-113) Perkembangan *game* edukasi dimulai dari perkembangan *video game* yang pesat yang kemudian menjadikan *game* edukasi menjadi media yang interaktif dan banyak dikembangkan di industri *game*. Melihat betapa populernya *game* edukasi, para pendidik menggunakan kesempatan ini untuk menerapkan *game* pada kurikulum.

*Game* edukasi adalah permainan yang dirancang untuk mendidik pengguna dalam pengembangan konsep dan pemahaman serta membimbing dan melatih kemampuan mereka. Merancang *game* edukasi harus memenuhi beberapa kriteria seperti:

1. Nilai Keseluruhan (*Overall Value*)

Nilai keseluruhan dari sebuah permainan berpusat pada desain dan jangka waktu permainan.

2. Dapat Digunakan (*Usability*)

Aplikasi harus dirancang dengan tampilan yang mudah digunakan pengguna.

### 3. Keakuratan

Sebuah aplikasi harus sesuai dengan model yang dirancang pada saat tahap perancangan.

### 4. Kesesuaian (*Appropriateness*)

Aplikasi harus memiliki desain dan isi yang disesuaikan terhadap kebutuhan penggunanya.

### 5. Relevan (*Relevance*)

Aplikasi harus dapat mengarah pengguna dalam mencapai target atau kegagalan.

### 6. Umpan balik (*Feedback*)

*Feedback* berfungsi untuk memberi pemahaman kepada pengguna bahwa cara bermain pengguna sesuai dengan objek *game* atau tidak. seperti efek suara yang mengindikasikan keberhasilan atau kekalahan pada sebuah permainan.

## 2.1.3 Android

Android adalah sebuah sistem operasi yang bersifat *open-source* dan dikembangkan oleh *Google*. Android dapat digunakan pada perangkat mobile seperti ponsel pintar dan tablet. Android disebut *open-source* karena kode sistem operasi dapat dilihat, diganti, dan diunduh secara gratis Hutabri & Putri (2019:58).

Menurut Rofiq & Fathul Uzzy (2014: 50-51) fitur-fitur yang dimiliki Android adalah sebagai berikut:

1. *Framework* aplikasi, untuk melakukan daur ulang dan penggantian komponen.
2. *Browser* terintegrasi yang menggunakan *engine Open Source Webkit* yang mana *engine* ini juga digunakan pada *browser iPhone*.
3. Android menggunakan *VGA (Video Graphic Adapter)* yang berkemampuan untuk mengolah grafik 2D dan 3D.
4. Android mendukung berbagai teknologi konektivitas seluler, Bluetooth dan Wi-Fi.
5. Android memungkinkan pengguna untuk mengirim pesan dalam bentuk *SMS (Short Message Service)* dan *MMS (Multimedia Messaging Service)*.
6. Android dapat mengkompilasi dan mengeksekusi *software* yang ditulis dalam bahasa Java menggunakan mesin *virtual Dalvik* yang dirancang untuk penggunaan perangkat *mobile*.
7. Android mendukung berbagai format audio dan video.
8. Android memiliki dukungan perangkat keras tambahan seperti kamera, layer sentuh, GPS, pengukur kecepatan, magnetometer, akselerasi 2D dan 3D.
9. Mirip dengan *App Store* pada iOS, Android memiliki *Google Play Store* yang digunakan untuk mengunduh aplikasi pada *smartphone* tanpa menggunakan komputer.
10. Android memiliki fitur *Multi touch* dimana pengguna dapat menggunakan lebih dari satu jari saat berinteraksi dengan *layer* sentuh dalam waktu bersamaan.

11. Android memiliki area pengembangan yang luas, seperti *emulator*, *Debugging tools* dan *plugin* untuk Eclipse IDE.

Menurut Winata & Setiawan (2013: 36) Penjelasan arsitektur komponen utama dari Android adalah sebagai berikut:

1. *Linux kernel*, adalah tumpukan terbawah pada arsitektur Android. *Kernel Linux* digunakan untuk pembangunan sistem Android dan berfungsi sebagai lapisan pemisahan antara perangkat keras dan perangkat lunak.
2. *Android Runtime*, di dalam *Android Runtime* terdapat *Core Libraries* yang berisi serangkaian inti library Java dan *Dalvik Virtual Machine* yang berfungsi untuk memberikan daya dan pengoptimalan Android.
3. *Libraries*, berada pada lapisan yang sama dengan *Android Runtime* dan merupakan lapisan yang tersimpan *set-set library* dalam Bahasa pemrograman C/C++ yang ada dalam Android.
4. *Application Framework*, di lapisan ini meliputi program yang beroperasi untuk mengatur fungsi-fungsi dasar pada ponsel pintar dan merupakan serangkaian *tools* dasar seperti alokasi sumber daya ponsel pintar, aplikasi ponsel pintar, pergantian antar proses atau program, dan pelacakan lokasi fisik ponsel pintar.
5. *Applications*, dalam lapisan ini terdapat fungsi-fungsi pada ponsel pintar seperti menelepon, pengiriman pesan singkat, menjalankan *Web Browser*, pengaksesan pada kontak, melakukan permainan, dan lain-lain.

Menurut Herriyance, dkk (2017:66-68) versi-versi dari sistem operasi Android adalah sebagai berikut:

1. Android versi 1.0 dan 1.1
2. *Cupcake* (versi 1.5)
3. *Donut* (versi 1.6)
4. *Éclair* (versi 2.0-2.1)
5. *Froyo* (versi 2.2-2.2.3)
6. *Gingerbread* (versi 2.3-2.3.7)
7. *Honeycomb* (versi 3.0-3.2)
8. *Ice Cream Sandwich* (versi 4.0-4.0.4)
9. *Jelly Bean* (versi 4.1-4.3)
10. *KitKat* (versi 4.4)
11. *Lollipop* (versi 5.0-5.0.2)

Android *Lollipop* merupakan *update* yang *fantastic* karena membawa banyak perubahan seperti *Material Design* yang memiliki tampilan lebih berwarna dan responsif, notifikasi yang mudah diakses, hemat baterai dan fitur keamanan terenkripsi. Menurut Dongre, dkk (2017:71-72) Android *Lollipop* diresmikan dengan nama kode “Android L” pada tanggal 25 Juni 2014 pada acara *Google I/O*. Android *Lollipop* tersedia sebagai *update* resmi yang dapat diperbarui secara *Over-The-Air* (OTA) pada tanggal 12 November 2014, untuk perangkat tertentu yang menjalankan distribusi Android yang dilayani oleh *Google*, termasuk perangkat Nexus dan edisi *Google Play*. *Source code* nya menjadi tersedia pada tanggal 3 November 2014. kemudian, *Google* membuat perubahan internal pada platform, dengan *Android Runtime* (ART) secara resmi menggantikan *Dalvik* untuk peningkatan kinerja aplikasi dan pengoptimalan penggunaan baterai yang dikenal

secara internal sebagai proyek *Volta*. Fitur-fitur baru pada Android *Lollipop* adalah sebagai berikut:

1. Menggunakan *Android Runtime* (ART) yang menggunakan *ahead-of-time* (AOT) *compilation* dan peningkatan *Garbage Collection* (GC) dan membawa dukungan untuk CPU 64-bit.
2. OpenGL ES 3.1 dan *Android Extension Pack* (AEP) untuk Konfigurasi GPU yang mendukung.
3. Layar aktivitas terkini dengan tugas dan bukan berdasarkan aplikasi, hingga maksimum tugas yang di konfigurasi per aplikasi.
4. *Vector Drawable*, dimana skala tidak kehilangan definisi dan membawa dukungan untuk *print preview*.
5. *Material Design* yang membawa antarmuka pengguna yang baru, antarmuka layer kunci baru, dan tidak lagi mendukung *Widget*.
6. Antarmuka baru pada notifikasi dan pengaturan cepat.
7. Proyek *Volta* untuk meningkatkan masa pakai baterai.
8. Pencarian yang dapat dilakukan dalam pengaturan sistem untuk akses cepat ke pengaturan tertentu.
9. Layar kunci terdapat pintasan untuk pengaturan aplikasi dan notifikasi.
10. Login tamu dan beberapa akun pengguna disediakan di lebih banyak perangkat.
11. Input dan output audio melalui perangkat USB.
12. Pengembalian kemampuan aplikasi pihak ketiga untuk membaca dan mengubah data yang terletak di mana saja pada penyimpanan eksternal.

13. Penjepitan layar aplikasi untuk membatasi aktivitas pengguna.
14. Aplikasi yang baru saja digunakan diingat bahkan setelah *restart* perangkat.
15. *Web view* menerima pembaruan independen melalui Google Play dengan alasan keamanan dan tidak lagi melalui pembaruan seluruh sistem.
16. Penambahan beberapa bahasa baru.
17. *Tap and Go* yang berfungsi untuk bermigrasi dengan cepat ke perangkat Android baru dengan penggunaan Bluetooth dan NFC.

#### **2.1.4 Pengembangan Perangkat Lunak**

Menurut Kang & Cho (2015:2) terdapat beberapa langkah-langkah dasar yang meliputi empat tahap perkembangan yang harus diikuti pengembang dalam mengembangkan suatu aplikasi.

1. Pengaturan: Dalam tahap ini, pengembang memasang dan mengatur lingkungan pengembangan aplikasi serta membuat *Android Virtual Device* (AVD) dan menghubungkan perangkat-perangkat keras, di mana pengembang dapat memasang aplikasi.
2. Pengembangan: Merupakan tahap persiapan dan perkembangan proyek *Android*, yang terdapat semua kode sumber dan berkas sumber daya untuk aplikasi pengembang.
3. *Debugging and Testing*: Merupakan tahap pembangunan proyek dalam bentuk paket *debuggable* .apk yang dapat dipasang dan dijalankan pada *emulator* atau perangkat bersistem operasi *Android*.
4. Menerbitkan: Merupakan tahap konfigurasi dan membangun aplikasi untuk

dirilis dan didistribusikan kepada pengguna.

Menurut Apriyandi, dkk (2019:173) *Game development life cycle* merupakan suatu metode membangun dan mengembangkan *game*, dimana metode ini terdapat beberapa tahapan yaitu *initiation, pre-production, production, testing, beta, release*.

1. *Initiation* merupakan titik awal dari proyek pengembangan *game* yaitu bermula dari ide *game*.
2. *Pre-production* merupakan tahap yang berhubungan dengan desain *game* dan perancangan dengan penggunaan UML (*Unified Modeling Language*).
3. *Production* merupakan tahap penerjemahan rancangan desain *game, concept art* dan aspek lainnya yang menjadi unsur penyusunan *game*.
4. *Alpha Testing* merupakan tahap pengujian yang dilakukan pengembang untuk menguji performa, fungsionalitas dan kompatibilitas dari aplikasi.
5. *Beta Testing* merupakan tahap dimana pengujian dilakukan oleh pihak ketiga untuk pengujian penerimaan *game* dan untuk pendeteksian *error* dan keluhan.
6. *Release* merupakan tahap dimana *game* sudah dapat diluncurkan untuk massa.

*Game* yang dirancang berbentuk kuis dimana pengguna akan disajikan sepuluh soal pada setiap level dengan pilihan ganda dan pengguna diberikan waktu lima belas detik untuk menjawab setiap soal. *game* ini juga terdapat bagian yang berisi materi pembelajaran yang dapat di pelajari oleh pengguna.

### 2.1.5 *Unified Modeling Language (UML)*

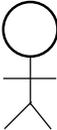
Menurut Urva & Siregar (2015: 93-95) *Unified Modeling Language (UML)* merupakan bahasa spesifikasi standar yang digunakan untuk mendokumentasi, menspesifikasi dan membangun perangkat lunak. UML adalah metodologi yang digunakan dalam pengembangan sistem berorientasi objek dan sebagai alat pendukung pengembangan sistem.

Alat bantu yang digunakan dalam perancangan sistem berorientasi objek berbasis UML adalah sebagai berikut :

#### 1. Diagram *Use Case*

Diagram *Use Case* adalah pemodelan perilaku sistem informasi yang akan dirancang. *Use case* menjelaskan interaksi yang terjadi antar satu atau lebih aktor dengan sistem informasi yang dirancang. *Use case* berfungsi untuk mengetahui fungsi-fungsi yang terdapat dalam suatu sistem informasi dan siapa yang memiliki hak dalam penggunaan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam diagram *use case* adalah sebagai berikut :

**Tabel 2.1** Simbol *Use Case*

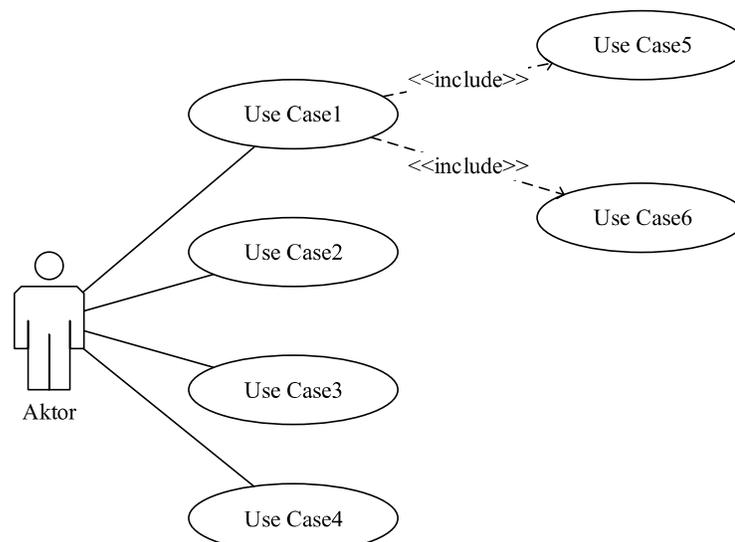
Gambar	Keterangan
 Use Case	Untuk penggambaran fungsi yang tersedia pada sistem yang bertukar pesan antara unit dan aktor, umumnya dinyatakan dengan penggunaan kata kerja di awal nama <i>use case</i> .
 Aktor	Aktor adalah orang yang menggunakan fungsi pada sistem. Untuk mengidentifikasi aktor, perlu dilakukan pembagian tugas yang berhubungan dengan peran terhadap sistem. Aktor atau sistem dapat muncul dengan beberapa peran. Aktor berinteraksi dengan <i>use case</i> namun tidak memiliki pengendalian terhadap <i>use case</i> .

**Tabel 2.1** ( Lanjutan ) Simbol *Use Case*

 Asosiasi	Asosiasi antara aktor dan <i>use case</i> untuk mengidentifikasi siapa atau apa yang berinteraksi dengan langsung
 Generalisasi	Asosiasi antara aktor dan <i>use case</i> untuk mengidentifikasi aktor yang berinteraksi secara pasif dengan sistem.
 Include	<i>Include</i> , adalah pemanggilan <i>use case</i> oleh <i>use case</i> lain, misalnya pemanggilan suatu fungsi program.
 Extend	<i>Extend</i> , adalah perluasan dari <i>use case</i> lain jika syarat atau kondisi tercapai.

**Sumber :** Urva & Siregar (2015)

Berikut adalah contoh perancangan diagram use case yang dapat dilakukan :

**Gambar 2.19** Diagram Use Case

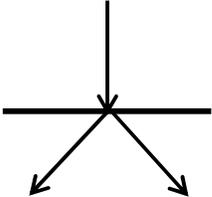
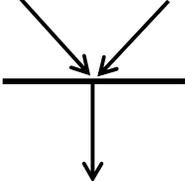
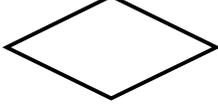
**Sumber :** Urva & Siregar (2015)

## 2. Diagram Aktivitas (*Activity Diagram*)

Diagram Aktivitas memvisualkan aliran kerja dari suatu proses sistem.

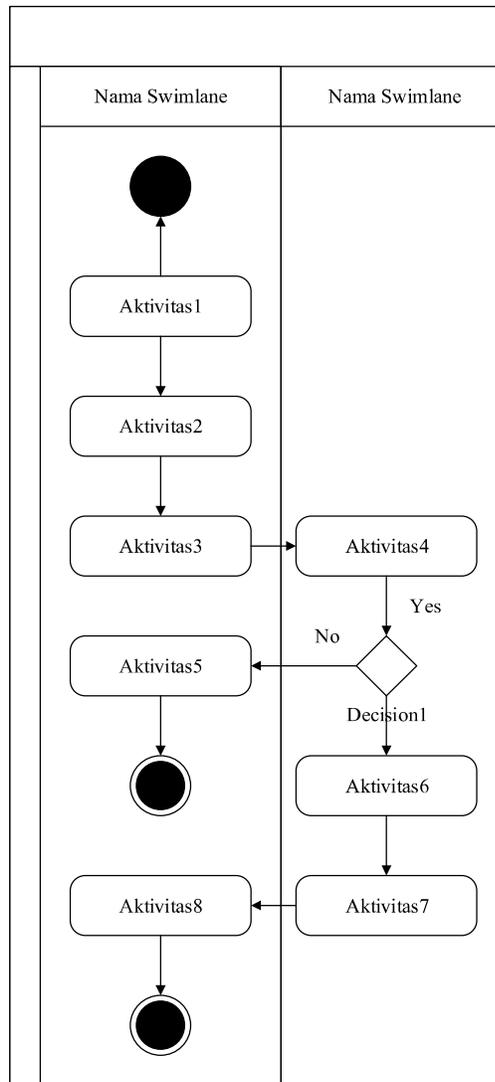
Simbol-simbol yang terdapat pada diagram aktivitas adalah sebagai berikut :

Tabel 2.2 Simbol Diagram Aktivitas

Gambar	Keterangan
	<i>Start point</i> , diposisikan pada pojok kiri atas yang menandakan bermulainya aktivitas.
	<i>End point</i> , menandakan berakhirnya aktivitas.
	<i>Activities</i> , menandakan sebuah proses atau kegiatan.
	<i>Fork</i> (percabangan), berfungsi untuk mengindikasikan proses yang dijalankan secara paralel atau penggabungan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan), berfungsi untuk mengindikasikan adanya dekomposisi.
	<i>Decision Points</i> , untuk mengindikasikan pengambilan keputusan seperti <i>true</i> dan <i>false</i> .
	<i>Swimlane</i> , pembagian diagram aktivitas untuk mengindikasikan siapa menjalankan apa.

Sumber : Urva & Siregar (2015)

Berikut adalah contoh perancangan diagram aktivitas yang dapat dilakukan :

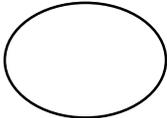
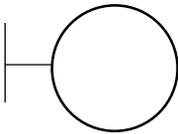
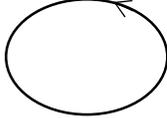
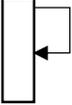


**Gambar 2.20** Diagram Aktivitas  
**Sumber :** Urva & Siregar (2015)

### 3. Diagram Sekuen (*Sequence Diagram*)

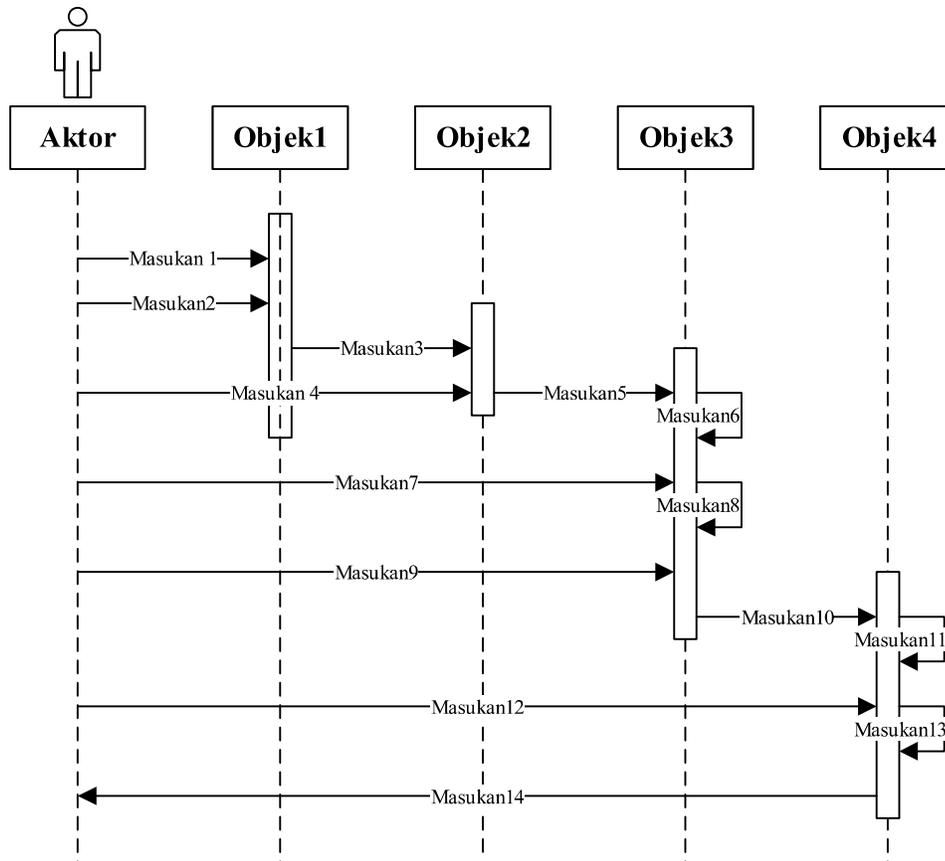
Diagram Sekuen memvisualkan perilaku objek terhadap *use case* dengan menjelaskan lama hidup objek serta pengiriman dan penerimaan pesan antar objek. Simbol-simbol yang terdapat pada diagram sekuen adalah sebagai berikut:

**Tabel 2.3** Simbol Diagram Sekuen

Gambar	Keterangan
	<i>Entity Class</i> , adalah komponen dari sistem yang terdapat kumpulan kelas yang berisi entitas yang membentuk gambaran awal sistem dan sebagai fondasi untuk penyusunan basis data.
	<i>Boundary Class</i> , terdapat kumpulan kelas yang menjadi interaksi atau <i>interface</i> antar satu atau lebih aktor pada sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
	<i>Control Class</i> , merupakan sebuah objek yang terdapat logika aplikasi yang tidak terdapat tanggung jawab terhadap entitas, misalnya aturan dan kalkulasi bisnis yang menyangkut berbagai objek.
	<i>Message</i> , merupakan symbol pengiriman pesan antar <i>class</i> ,
	<i>Recursive</i> , menandakan pengiriman pesan yang dikirimkan kepada dirinya sendiri.
	<i>Activation</i> , mewakili suatu pelaksanaan operasi objek, Panjang kotak <i>activation</i> menandakan jangka waktu aktivitas suatu operasi.
	<i>Lifeline</i> , berbentuk garis putus-putus yang berhubung dengan objek, di sepanjang lifeline terdapat <i>activation</i>

Sumber : Urva & Siregar (2015)

Berikut adalah contoh perancangan diagram sekuen yang dapat dilakukan :



**Gambar 2.21** Diagram Sekuen  
**Sumber :** Urva & Siregar (2015)

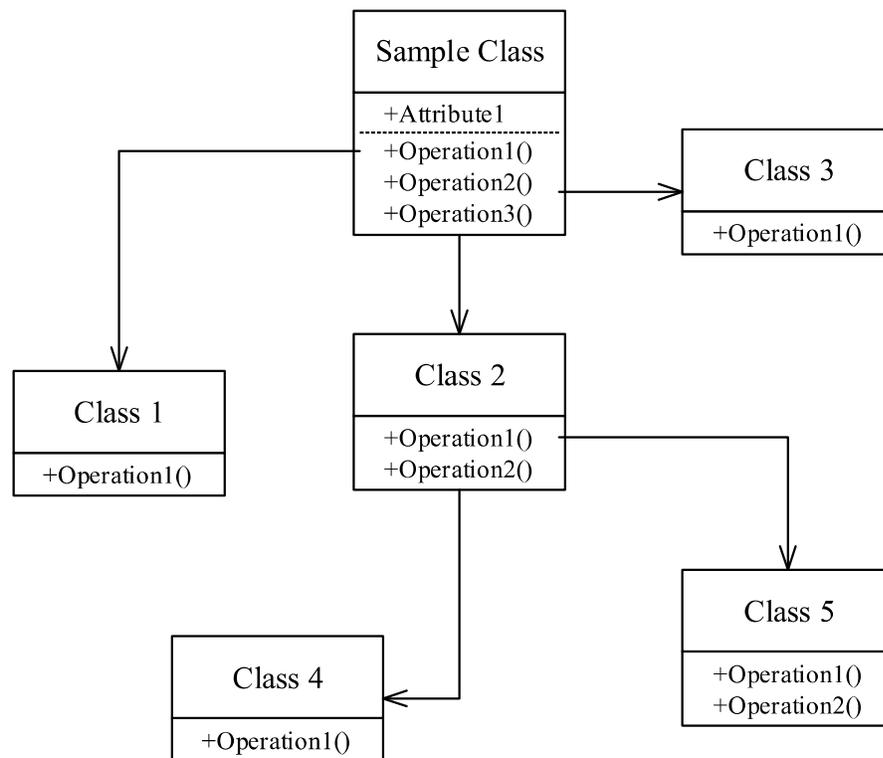
#### 4. Diagram Kelas (*Class Diagram*)

Diagram Kelas adalah hubungan antar kelas dan penjelasan dari tiap kelas yang terdapat pada model desain sebuah sistem, serta menunjukkan peraturan dan tanggung jawab entitas yang menentukan perilaku sistem. Diagram kelas juga mengindikasikan atribut dan operasi dari suatu kelas dan pembatas yang berkaitan dengan objek yang dihubungkan. Diagram kelas secara khas terdiri dari Kelas, Relasi, Asosiasi, Generalisasi dan Agregasi, Atribut, Operasi,

*Visibility*, tingkat akses objek eksternal pada suatu atribut atau operasi.

Hubungan antar kelas memiliki keterangan yang disebut sebagai kardinalitas.

Berikut adalah contoh perancangan diagram kelas yang dapat dilakukan :



**Gambar 2.22** Diagram Kelas  
**Sumber :** Urva & Siregar (2015)

### 2.1.6 Database

Menurut Hendini (2016:107-108) *database* adalah kumpulan dari tabel-tabel yang saling berelasi dan relasi tersebut dapat diindikasikan dengan kunci dari setiap tabel yang ada. *Database* juga berupa kumpulan data yang memvisualkan aktivitas dan pelakunya dalam sebuah organisasi. Sistem *database* adalah sistem komputer yang berguna untuk penyimpanan dan pengelolaan data.

## **2.2 Bahasa Inggris**

Menurut Hakim (2016:274-276) Penguasaan Bahasa Inggris menjadi suatu keahlian yang harus dipelajari sebagai orang Indonesia dimana berbahasa Inggris sudah harus dibiasakan sejak usia dini karena Bahasa Inggris tidak hanya digunakan pada saat sekolah, namun sampai dunia kerja hingga tua Bahasa Inggris akan lebih banyak digunakan. Penguasaan Bahasa Inggris sangat berpengaruh pada berbagai macam bidang seperti bisnis, pendidikan, pariwisata dan untuk masa depan dan globalisasi, terutama pada masalah persaingan di dalam dunia kerja.

Pada dunia bisnis, Bahasa Inggris sangat penting terutama ketika perusahaan menjalin hubungan kerjasama dengan perusahaan asing. Di dalam dunia usaha yang mendunia, banyak perusahaan Indonesia yang telah masuk ke pasar dunia dan tidak sedikit perusahaan internasional yang masuk ke pasar Indonesia dimana satu-satunya cara untuk berkomunikasi dengan efektif adalah menggunakan satu bahasa untuk semua yaitu Bahasa Inggris.

Di dalam dunia pendidikan, penguasaan Bahasa Inggris juga sangat penting karena orang-orang dapat mempelajari ilmu pengetahuan yang menggunakan pengantar Bahasa Inggris, dimana banyak buku serta informasi di dalam dunia internet sudah menggunakan Bahasa Inggris.

Indonesia merupakan salah satu negara yang penuh dengan keindahan alamnya, berbagai tempat telah dijadikan sebagai daerah pariwisata. Sehingga banyak turis dari berbagai negara berkunjung di negara Indonesia. Dengan menguasai Bahasa Inggris, masyarakat Indonesia akan lebih mudah dalam berkomunikasi dengan turis asing terutama masyarakat yang bekerja langsung di

daerah pariwisata. Sehingga tidak dapat ditolak bahwa Bahasa Inggris merupakan bahasa yang wajib dikuasai setiap individu didunia, setidaknya individu harus mampu berkomunikasi dalam hal konversasi sehari-hari.

Variabel pada penelitian ini adalah Bahasa Inggris dimana indikator yang akan digunakan adalah *Simple Present Tense, Simple Past Tense, Present Continuous Tense, Present Perfect Continuous Tense, Past Continuous Tense, Present Perfect Tense, Past Perfect Tense, dan Future Tense*. Menurut Elfitriani, dkk (2014,109-110) *tenses* merupakan suatu kalimat yang menyatakan waktu terlaksananya suatu peristiwa dan lima *tenses* yang penting untuk menguasai Bahasa Inggris dengan baik adalah *Simple Present Tense, Simple Past Tense, Present Continuous Tense, Present Perfect Tense, dan Future Tense*.

#### 1. *Simple Present Tense*

Menurut Elfitriani, dkk (2014:109-110) *Simple Present Tense* digunakan untuk menyatakan aktivitas dan kebiasaan sehari-hari, menyatakan fakta universal, memberi instruksi. Biasanya kalimat ini didampingi dengan kata-kata seperti: *always, never, often, normally, seldom, usually, sometimes*. Menurut Chamidy & Agasta (2011:159) Rumus dari *Simple Present Tense* adalah:

(+) S + verb I + O

(-) S + (do/does) not + verb I + O

(?) Do/does + S + verb I + O

Menurut Amrudin, dkk (2013:2-3) verb yang diakhiri dengan huruf s, -z, -x, ch dan =sh diikuti dengan akhiran -es sedangkan verb yang lain

menggunakan akhiran -s, kecuali *goes* dan *does*. Contoh dari penggunaan -es adalah *buzzes, fixes, catches, pushes*. *Adverb* yang umumnya digunakan pada *tense* ini adalah *usually, rarely, everyday, frequently, never, sometimes, always*.

1. Contoh kalimat (+) : *She studies English five days in a week.*

Uraian :                    S    verb I Object

2. Contoh kalimat (+) : *He plays football everyday.*

Uraian :                    S    verb1 Object

3. Contoh kalimat (-) : *She does not study English five days in a week.*

Uraian :                    S                    verb I Object

4. Contoh kalimat (-) : *I do not play football.*

Uraian :                    S                    v I    Object

5. Contoh kalimat (?) : *Does she study English five days in a week?*

Uraian :                    S    verb I Object

6. Contoh kalimat (?) : *Do they speak English well?*

Uraian :                    S    verb I Object

## 2. *Simple Past Tense*

Menurut Elfitriani, dkk (2014:109-110) *Simple Past Tense* digunakan untuk menyampaikan sebuah peristiwa yang telah terjadi. Biasanya kalimat ini didampingi dengan kata-kata seperti: *yesterday, 3 months ago, in 1998, the other day, last Tuesday*. Menurut Kurniawan dan Ament (2016) Bentuk dari *Simple Past Tense* adalah:



3. Contoh kalimat (-) : She did not fry an egg for dinner yesterday.

Uraian :                S            V1        O                    Adv

4. Contoh kalimat (-) : He did not attend the flag ceremony yesterday.

Uraian :                S                    V2                    O                    Adv

5. Contoh kalimat (?) : Did she fry an egg for dinner yesterday?

Uraian :                S        V1            O                    Adv

6. Contoh kalimat (?) : Did he attend the flag ceremony yesterday?

Uraian :                S    V1                    O                    Adv

### 3. *Present Continuous Tense*

Menurut Elfitriani, dkk (2014:109-110) *Present Continuous Tense* digunakan untuk menyampaikan sebuah peristiwa yang masih berlangsung pada saat kalimat ini diucapkan. Biasanya kalimat ini didampingi dengan kata-kata seperti: *at the moment, right now, now*. Menurut Chamidy & Agasta (2011:159) Rumus dari *Present Continuous Tense* adalah:

(+) S + to be + verb ing + O

(-) S + to be not + verb ing + O

(?) To be + S + verb ing + O

1. Contoh kalimat (+) : He is sleeping in his bedroom right now.

Uraian :                S    tobe    V-ing                    O                    Adv

2. Contoh kalimat (+) : I am cooking spaghetti at the moment.

Uraian :                S    tobe    V-ing                    O                    Adv

3. Contoh kalimat (-) : He is not working at the office right now.

Uraian :                S tobe     V-ing                O     Adv

4. Contoh kalimat (-) : *She is not feeling good right now.*

Uraian :                S tobe     V-ing     O     Adv

5. Contoh kalimat (?) : *Are you doing your homework right now?*

Uraian :                tobe S     V-ing                O     Adv

6. Contoh kalimat (?) : *Is she washing the dishes right now?*

Uraian :                tobe S     V-ing                O     Adv

#### 4. *Present Perfect Continuous Tense*

Menurut Menurut Kurniawan & Ament (2016) *Present Perfect Continuous Tense* digunakan untuk mengindikasikan sebuah peristiwa yang belum selesai yang berlanjut dari masa lampau hingga masa depan. *Present Perfect Continuous Tense* menggunakan ekspresi waktu seperti *for*, *since*, *before*, dan ekspresi waktu lainnya.

Bentuk dari *Present Perfect Continuous Tense* adalah:

(+) S + have/has + been + Verb-ing ± o ± adv.

(-) S + have/has + not + been + Verb-ing ± o ± adv.

(?) (WH) + have/has + S + been + Verb-ing ± o ± adv?

1. Contoh kalimat (+) : *You have been singing on the stage for more than an hour.*

Uraian :                S                                V-ing                O     Adv

2. Contoh kalimat (+) : *She has been eating diet plum for more than a month.*

Uraian :                    S                    V-ing    O                    Adv

3. Contoh kalimat (-) : You have not been eating healthy for quite some time.

Uraian :                    S                                    V-ing    O                    Adv

4. Contoh kalimat (-) : She has not been studying for exam since last semester.

Uraian :                    S                                    V-ing                    O    Adv

5. Contoh kalimat (?) : Have she been crying at home for more than a day?

Uraian :                                    S                    V-ing                    O    Adv

6. Contoh kalimat (?) : Has she been seeing michael lately?

Uraian :                                    S                    V-ing    O    Adv

##### 5. *Past Continuous Tense*

Menurut Kurniawan & Ament (2016) *Past Continuous Tense* dibentuk dari kalimat *past tense* dengan *verb "to be" + verb-ing*. Digunakan untuk menyatakan peristiwa yang berlanjut pada waktu masa lalu. Terkadang, *tense* ini juga digunakan untuk menyampaikan sebuah peristiwa yang sedang berlangsung saat peristiwa kedua berlangsung pada masa lalu.

Bentuknya adalah:

S + was/were + verb-ing + o + adv.

S + was/were + verb-ing + o + adv + S + verb II + o.

1. Contoh kalimat : a. He was sleeping at home last night.

Uraian :                    S                    verb-ing    O                    adv

2. Contoh kalimat : *They were doing their homework just now when*

Uraian :            S            V-ing            O            Adv

Lanjutan : *The teacher came into the classroom.*

Uraian :            S            V2                            O

3. Contoh kalimat : *They were eating burger when i brought them pizza.*

Uraian :            S            V-ing    O            S    V2            O

4. Contoh kalimat : *He was jogging at the park when i came to his house.*

Uraian :            S            V-ing            O            S    V2            O

5. Contoh kalimat : *He was painting his house when he broke the vase.*

Uraian :            S            V-ing            O            S    V2            O

## 6. *Present Perfect Tense*

Menurut Elfitriani, dkk (2014:109-110) *Present Perfect Tense* digunakan untuk menyampaikan sebuah peristiwa yang sudah terjadi hingga sekarang, peristiwa yang telah terjadi akhir-akhir ini, peristiwa yang sudah terjadi yang mempengaruhi situasi sekarang, dan peristiwa yang pernah terjadi sekali atau beberapa kali sebelum pembicaraan ini berlangsung. Kalimat ini biasanya didampingi dengan kata-kata seperti: *already, just, ever, until now, never, so far, not yet, up to now*. Menurut Kurniawan & Ament (2016)

Bentuk dari *Present Perfect Tense* adalah:

(+) S + have/has + Verb III ± o ± adv.

(-) S + have/has + not + Verb III ± o ± adv.

(?) (WH) + have/has + S + Verb III ± o ± adv?



2. Contoh kalimat (+) : *She will fail this semester.*

Uraian :                    S        V1        O

3. Contoh kalimat (+) : *I shall buy a new phone.*

Uraian :                    S        V1        O

4. Contoh kalimat (-) : *She is not going to fail this semester.*

Uraian :                    S be                    V1        O

5. Contoh kalimat (-) : *She will not fail this semester.*

Uraian :                    S                    V1        O

6. Contoh kalimat (-) : *I shall not buy a new phone.*

Uraian :                    S                    V1        O

7. Contoh kalimat (?) : *Is she going to fail this semester?*

Uraian :                    be S                    V1        O

8. Contoh kalimat (?) : *Will she fail this semester?*

Uraian :                    S V1        O

9. Contoh kalimat (?) : *Shall I buy a new phone?*

Uraian :                    S V1        O

*Game* yang dirancang terdiri atas sepuluh level dengan masing-masing level sepuluh soal.

## 2.3 Software Pendukung

### 2.3.1 Android Studio



**Gambar 2.23** Android Studio 3.5  
Sumber : Aplikasi

Menurut Juansyah Andi (2015:2-3) *Android Studio* adalah IDE (*Integrated Development Environment*) resmi yang di rilis *Google* pada acara *Google I/O Conference* pada 16 Mei 2013. *Android Studio* menggantikan *Eclipse* sebagai IDE resmi yang digunakan untuk mengembangkan aplikasi *Android* yang bersifat *open source* dan gratis.

*Android Studio* dikembangkan berdasarkan *IntelliJ IDEA* yang mirip dengan ADT plugin (*Android Development Tools*). Fitur-fitur yang terdapat pada *Android Studio* adalah :

- a. Proyek berbasis pada *Gradle Build*.
- b. *Refractory* dan pembenahan bug yang tepat.
- c. *Tools* baru bernama “*Lint*” yang digunakan untuk mengawasi kecepatan, kegunaan, serta kecocokan aplikasi dengan cepat.
- d. Mendukung *Proguard And App-signing* untuk keamanan.

- e. Memiliki GUI aplikasi *android* yang lebih mudah.
- f. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang dikembangkan.

Menurut Maiyana (2018:58) *Integrated Development Environment* merupakan aplikasi yang memiliki fungsi-fungsi terintegrasi seperti *code editor*, *compiler*, *debugger*, dan sebagainya yang diperlukan dalam mengembang suatu perangkat lunak.

Menurut Kusniyati & Pangondian Sitanggang (2016: 12-13) Android SDK merupakan *tools API (Application Programming Interface)* yang berfungsi untuk pengembangan aplikasi di *platform* Android dengan penggunaan Bahasa pemrograman Java. Menurut Maiyana (2018:58) *Android SDK* terdiri dari *libraries*, *debugger*, *handset emulator*, dokumentasi, contoh kode dan tutorial.

Menurut Maiyana (2018:58) *Android Virtual Device* adalah *emulator* yang digunakan untuk menjalankan aplikasi *android* yang telah dibuat. *Emulator* ini kemudian digunakan sebagai tempat untuk menguji dan menjalankan aplikasi *android* tanpa harus menggunakan perangkat *android* yang sebenarnya. Karakteristik AVD juga dapat ditentukan seperti, versi *android*, jenis dan ukuran layer dan memori. AVD dapat dibuat sebanyak yang diinginkan pengguna.

### 2.3.2 Java

Menurut Maiyana (2018:58-59) *JDK (Java Development Kit)* merupakan paket fungsi API untuk bahasa pemrograman *Java*, meliputi *JRE (Java Runtime Environment)* dan *JVM (Java Virtual Machine)*.

*Java* merupakan sebuah bahasa pemrograman yang berorientasi objek. *Java* diperkenalkan pada tahun 1995 oleh Sun Microsystem Inc., dan dipimpin oleh James Gosling pada saat itu. Penciptaan *Java* berawal dari keinginan Sun Microsystem untuk membuat sebuah bahasa pemrograman yang dapat dijalankan pada semua perangkat tanpa harus terikat oleh *platform* yang digunakan perangkat, sehingga *Java* bersifat *portable* dan *platform* mandiri. *Java* merupakan sebuah teknologi bahasa pemrograman yang memiliki sintaks dan peraturan pemrograman sendiri serta sebagai *platform* yang memiliki *virtual machine* dan *library* yang digunakan untuk menulis dan menjalankan aplikasi yang telah dibuat dengan bahasa pemrograman *Java*.

### 2.3.3 SQLite

SQLite adalah sebuah database *engine* yang tidak memerlukan server. Menurut Maiyana (2018:57) SQLite adalah suatu perangkat lunak *embedded* yang terkenal, dengan kombinasi SQL dan penggunaan memori yang sedikit dan performa yang cepat. SQLite menyediakan tiga metode yaitu:

1. *Constructor*

Menyediakan representasi versi dari database dan skema yang digunakan.

2. *OnCreate()*

Menyediakan *SQLiteDatabase object* yang digunakan dalam definisi tabel dan inisialisasi data.

### 3. *OnUpgrade()*

Menyediakan fasilitas konversi database dari database yang lama ke database versi baru atau sebaliknya.

## 2.4 Penelitian Terdahulu

1. *Game* edukasi pembelajaran bahasa Bali yang berjudul **“Rancang Bangun Aplikasi Game Edukasi Bahasa Bali pada Platform Android”** oleh Sanjaya, VOL. 2, NO. 1, April 2014 dengan standar ISSN 2252-3006 berbentuk permainan yang memberi pengetahuan dasar mengenai kosa kata bahasa bali untuk anak kelas 1 sekolah dasar yang berbasis android terdiri dari tiga kategori yaitu pengenalan kata, pengenalan angka serta penyusunan kalimat dalam bahasa Bali. *Game* ini dapat memberi pembelajaran pada anak sekaligus menarik anak untuk belajar sambil memainkan *Game* ini. Perbedaan dengan penelitian yang akan dilakukan terletak pada variabel yang digunakan.
2. Pengembangan *Game* Edukasi Bahasa Inggris Berbasis Android yang berjudul **“Pengembangan Education Game (Edugame) Berbasis Android Pada Mata Pelajaran Bahasa Inggris Untuk Peserta Didik Sekolah Dasar”** oleh Ramansyah, VOL. 2, NO.1, November 2015 dengan ISSN standar 2407-4489 yang berbasis android mengajarkan mengenai anggota badan dalam Bahasa Inggris untuk siswa Sekolah Dasar ini dikarenakan pembelajaran menggunakan buku kurang maksimal karena hanya menyajikan tulisan dan gambar tanpa suara cara membaca dengan

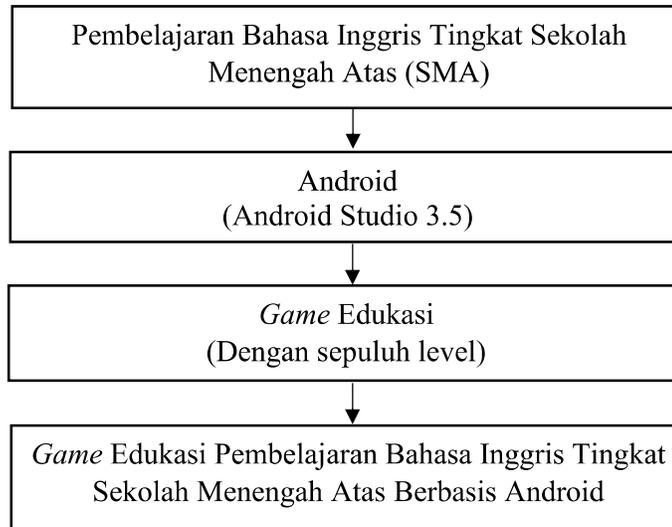
menggunakan Bahasa Inggris. *Game* ini akan menampilkan animasi bagian-bagian anggota tubuh beserta suara untuk mempelarinya menggunakan Bahasa Inggris. Perbedaan dengan penelitian yang akan dilakukan terletak pada lingkup objek yang diteliti.

3. Perancangan *Game* Edukasi yang berjudul **“Perancangan *Game* Edukasi Ilmu Pengetahuan Alam Dan Ilmu Pengetahuan Sosial Berbasis Android”** oleh Maryandi, dkk, VOL. 8, NO.2, Juni 2014 dengan standar ISSN 1978-8126 dilatarbelakangi oleh minimnya waktu guru dalam menjelaskan materi kepada siswa dan siswa mengalami kesulitan untuk menerima pelajaran di sekolah. *Game* berbasis android ini berbentuk kuis yang dapat dimainkan oleh siswa untuk menunjang daya ingat siswa sekaligus membantu siswa dalam mendapatkan materi tambahan dari kuis yang dimainkannya. Perbedaan dengan penelitian yang akan dilakukan terletak pada variabel yang digunakan.
4. Perancangan *Game* edukasi yang berjudul **“Pemanfaatan *Augmented Reality* Untuk *Game* Edukasi Bagi Anak Autis Tingkat Sekolah Dasar Di Rumah Pintar Salatiga”** Oleh Irsyadi dan Rohmah, VOL. 8, NO. 1, April 2017 dengan standar ISSN 2252-4983 yang dikarenakan sekolah terpadu Rumah Pintar Salatiga adalah sekolah untuk anak berkebutuhan khusus terutama tunagrahita dan autis dimana anak-anak ini mengalami kesulitan dalam memahami suatu pembelajaran. Selama ini pengenalan objek seperti hewan akan ditunjukkan dalam bentuk kertas dua dimensi, namun dengan metode ini, anak-anak mengalami kesulitan dalam

memahami objek tersebut. *Game* edukasi ini menggunakan *Augmented Reality* berbasis Android yang menampilkan objek dalam bentuk tiga dimensi sehingga memudahkan guru untuk menyampaikan materi serta memudahkan anak untuk memahami karena objek berbentuk lebih nyata. Perbedaan dengan penelitian yang akan dilakukan terletak pada metode dan variabel yang digunakan.

5. Perancangan *Game* edukasi yang berjudul **“Perancangan Aplikasi *Game* Edukasi Pembelajaran Anak Usia Dini Menggunakan *Linear Congruent Method (LCM)* Berbasis Android”** oleh Irsa, dkk, VOL. 6, NO. 1, Desember 2015 dengan standar e-ISSN 2477-3786 yang dilakukan untuk meningkatkan kemampuan dasar peserta didik Sekolah Dasar (SD) dalam hal membaca, menulis dan berhitung. Hal ini bertujuan untuk menyiapkan peserta didik dari usia dini untuk dapat hidup dalam masyarakat dan menghadapi perubahan dalam masyarakat baik dalam sisi ilmu pengetahuan, teknologi, sosial maupun budaya di tingkat lokal ataupun global. Sebab, menurut laporan *IEA Study of Reading Literacy*, kemampuan anak-anak sekolah dasar di Indonesia menduduki peringkat ke-30 dari 31 negara yang diteliti. *Game* ini menggunakan metode *Linear Congruent Method (LCM)* berbasis Android yang akan menampilkan halaman untuk pengenalan huruf, angka, belajar membaca dan belajar menulis. Perbedaan dengan penelitian yang akan dilakukan terletak pada metode dan variabel yang digunakan.

## 2.5 Kerangka Pemikiran



**Gambar 2.24** Kerangka Pemikiran  
**Sumber :** Data Penelitian (2019)

Penelitian ini dilatarbelakangi karena adanya permasalahan pembelajaran Bahasa Inggris di tingkat Sekolah Menengah Atas (SMA) yang masih menggunakan buku sebagai metode pembelajaran dan belum menggunakan aplikasi. Dan solusi dari permasalahan tersebut adalah dengan merancang *game* Edukasi Pembelajaran Bahasa Inggris Tingkat Sekolah Menengah Atas (SMA) Berbasis Android yang terdiri dari sepuluh tingkat *level* dan dirancang dengan menggunakan Android Studio versi 3.5.