

## BAB II LANDASAN TEORI

### 2.1. Tinjauan Studi

Cara utama yang dapat ditempuh dalam melakukan penelitian ini adalah dengan mencari beberapa sumber penelitian yang berhubungan dengan objek penelitian. Studi penelitian digunakan sebagai bahan perbandingan antara penelitian yang sudah pernah dilakukan dengan penelitian yang dirancang. Setelah melakukan evaluasi dari penelitian yang sudah pernah dilakukan maka dapat diambil satu atau lebih kesimpulan mengenai kelemahan dari sistem-sistem tersebut. Yang bertujuan untuk melakukan penyempurnaan dan pengembangan aplikasi dari hasil evaluasi sebelumnya. Beberapa studi penelitian yang telah dilakukan sebagai pendukung proses penelitian yang lebih baik lagi dari penelitian sebelumnya.

Titon Agung Saputra, Kriswandani, dan Novisita Ratu (2018), yang berjudul “Pengembangan Media Pembelajaran Menggunakan Aplikasi *Construct 2* Pada Materi Aljabar Kelas VII” dengan menggunakan metode *Borg & Gall*. Penulis menjelaskan masalah yang dialami oleh siswa kelas VII SMP Pangudi Luhur Salatiga adalah guru menyampaikan pembelajaran seperti biasa, sehingga siswa kurang tertarik dan kurang bersemangat dalam mengikuti pelajaran matematika. Selain itu, media pembelajarannya masih menggunakan papan tulis dan buku siswa. Sedangkan di SMP Pangudi Luhur terdapat ruang *laboratorium*, dengan jumlah komputer sebanyak 30 *unit*. Dengan berkembangnya teknologi pada saat ini maka peneliti mengembangkan media pembelajaran *game* edukasi berbasis *platform game* untuk siswa kelas VII SMP. Hasil dari uji N-Gain diperoleh nilai 4,90 yang berarti terjadi peningkatan sedang dari nilai pretest dan posttest siswa. Siswa menyatakan bahwa aplikasi tersebut menarik, tidak membosankan, dan dapat membantu dalam belajar[4].

Muhammad Defri Enkasyarif, dan Richi Dwi Agustia (2017), yang berjudul “Pembangunan *Game* Edukasi Petualangan Kolek Sebagai Media Pembelajaran Matematika Berbasis *Android*” dengan menggunakan metode

*MDLC versi Luther-Sutopo*. Penulis menjelaskan masalah yang dialami SMP N 1 Tanjungsari yaitu Matematika sering dianggap sebagai mata pelajaran yang sulit, oleh sebab itu banyak siswa yang memiliki nilai buruk. Metode pembelajaran yang digunakan masih konvensional seperti ceramah dan diskusi. Hanya 55% siswa yang menyukai matematika, dan 87,9% siswa merasa kesulitan. Maka dibangunlah sebuah game edukasi berbasis android yang membantu mengajarkan pelajaran matematika untuk anak SMP[5].

Andang Sapto Pramono, dan Endah Sudarmilah (2019), yang berjudul “Rancang Bangun *Game* Edukasi Petualangan *Geometri* Berbasis *Android*” dengan menggunakan metode *Sytem Development Life Cycle (SDLC)*. Penulis menjelaskan masalah yang dialami yaitu sarana atau media yang digunakan untuk melatih kemampuan pengukuran *geometri* anak jumlahnya masih sedikit. Mengingat perkembangan teknologi yang begitu pesat, peneliti mencoba memecahkan masalah ini lewat rekayasa perangkat lunak yang diyakini cukup populer, murah dan yang paling penying disukai anak-anak. Tujuan peneliti ialah menciptakan media untuk melatih kemampuan pengukuran geometri khususnya untuk kelas V melalui aplikasi berbasis *android*[6].

Muhammad Rizky Rahadi, Kodrat Iman Satoto, dan Ike Pertiwi Windasari (2016), yang berjudul “Perancangan *Game Math Adventure* Sebagai Media Pembelajaran Matematika Berbasis *Android*” dengan menggunakan metode *Multimedia Development Life Cycle (MDLC)*. Penulis menjelaskan masalah yang dialami anak-anak dari usia 6 sampai 9 tahun dalam belajar khususnya matematika yaitu jenuh, bosan, dan takut akan belajar matematika. Dengan berkembangnya teknologi pada saat ini maka dengan itu peneliti membuat game edukasi matematika berbasis android, karena lebih efektif jika disaat anak memainkan *game* ini, anak tidak bosan dan tidak takut untuk belajar matematika dengan harapan anak-anak bisa dengan mudah mengingat dalam belajar menghitung juga semangat anak untuk belajar akan lebih terpacu dan meningkatkan kualitas belajar anak[7].

Jada Ario Yustin, Herry Sujaini, dan M. Azhar Irwansyah (2016), yang berjudul “Rancang Bangun Aplikasi *Game* Edukasi Pembelajaran Matematika

Menggunakan *Construct 2*” dengan menggunakan metode *Multimedia Development Life Cycle (MDLC)*. Penulis menjelaskan masalah pembelajaran anak-anak khususnya pelajaran matematika, yang seharusnya masa perkembangan kecerdasan anak mencapai 80% terjadi ketika usia 8 tahun, dan 100% ketika anak mencapai usia 8-18 tahun malah menurun. Dari masalah tersebut, peneliti membangun sebuah *game* edukasi menggunakan *Construct 2*, sebagai media alternatif pembelajaran matematika, didapatkan hasil peningkatan nilai sebesar 32% pada *game* edukasi matematika. Sedangkan dengan media pembelajaran buku mengalami peningkatan sebesar 28%, sehingga peneliti menyimpulkan bahwa *game* edukasi matematika dapat meningkatkan kemampuan anak dalam mempelajari matematika dan dapat digunakan sebagai media alternatif pembelajaran[8].

Penelitian ini memiliki fokus permasalahan yang umumnya terjadi pada pembelajaran siswa sekolah khususnya pelajaran matematika. Dengan proses analisi, perancangan, hasil serta penerapan diharapkan dapat memberikan kemudahan siswa dalam belajar matematika.

### **2.3. Tinjauan Pustaka**

#### **2.2.1. Game Edukasi**

*Game* yang memiliki *content* pendidikan lebih dikenal dengan istilah *game edukasi*. *Game edukasi* ini bertujuan untuk memancing minat belajar anak terhadap materi pelajaran matematika sambil bermain *game*, dengan perasaan senang diharapkan siswa bisa lebih mudah memahami materi pelajaran yang disajikan. Jenis ini sebenarnya lebih mengacu kepada isi dan tujuan *game*, bukan jenis yang sesungguhnya, *game* merupakan sebuah *tools* yang efektif untuk mengajar karena mengandung prinsip-prinsip pembelajaran dan teknik *instruksional* yang efektif digunakan dalam penguatan pada level-level yang sulit[8].

### 2.2.2. Construct 2

Construct 2 adalah game dengan *software fitur HTML5, Game Creator* yang dirancang khusus untuk *game 2D (platform game)*. Dengan Construct 2 memungkinkan siapa saja membuat *game* tanpa harus memiliki pengalaman pemrograman. Dikembangkan oleh Scirra Ltd, hal ini ditujukan terutama untuk para *non programmer* yang ingin menciptakan sebuah *game* secara *drag and drop* menggunakan editir visual dan berbasis sistem logika perilaku.

Pengembangan permainan Construct 2 dapat memublishnya ke beberapa platform seperti *HTML5 website, Google Chrome Webstore, Facebook, Phonegap (Android), Windows Phone, Windows 8*. Pada Construct 2 telah disediakan 70 *visual effect* yang menggunakan *engine WebGL*. Selain itu juga dilengkapi dengan 20 built-in plugin dan behavior (perilaku objek) sehingga kita bisa membuat *sprite, objek teks*, mengkoneksikan dengan *facebook*, menambah musik, memanipulasi penyimpanan data *game* dan lain sebagainya[8].

Berikut beberapa tampilan pada *Construct 2*:

#### a. Start Page

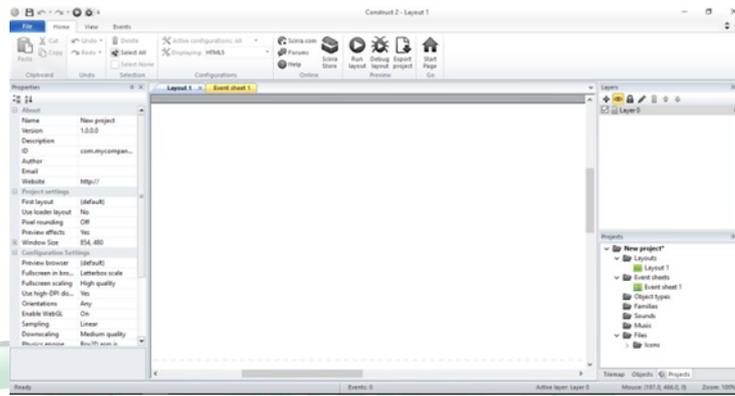
Halaman muka adalah halaman yang akan muncul pertama kali pada saat membuka sebuah program. Ini adalah tampilan awal dari *Construct 2*. Pada halaman ini tersedia *link-link* yang memungkinkan untuk membuat proyek baru, melanjutkan proyek yang sedang dikerjakan, melihat contoh-contoh proyek, mempelajari manual dan tutorial dan lain sebagainya.



Gambar 2. 1. Start Page

b. *User Interface*

Bagian berikut menjelaskan beberapa komponen penting pada *user interface Construct 2* disertai penjelasan umum dari setiap komponennya.



Gambar 2. 2. User Interface

c. *Menu Bar dan Ribbon Tabs*

Tampilan menu dalam *Construct 2* menggunakan bentuk *ribbon*. Tombol berbentuk roda gigi lambing *Scirra* akan menampilkan *drop-down ribbon* (bentuknya berbeda-beda, bergantung pada versi *engine*).



Gambar 2. 3. Menu Bar dan Ribbon Tabs

d. *Layout dan Tab*

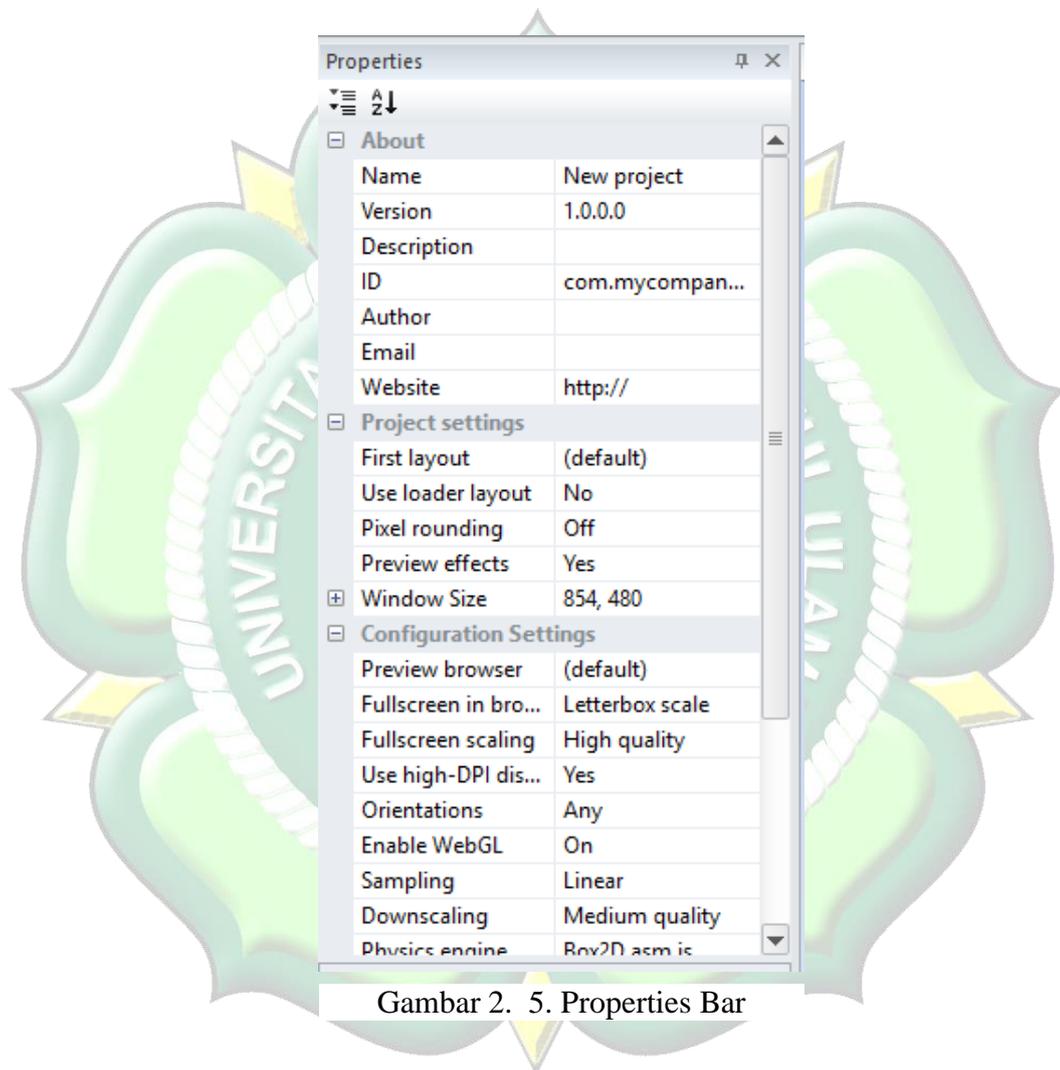
*Layout* adalah tempat untuk menempatkan objek, mendesain level, dan sebagainya. Tampilan *layout* dibagi menjadi dua, yaitu *layout* dan *windows size*. *Layout* adalah seluruh lembar kerja berwarna putih, sedangkan *windows size* merepresentasikan ukuran layar yang digunakan. Sedangkan *Tab* berfungsi sebagai pengganti *layout* maupun *event sheet* yang ingin dikerjakan. *Tab* dapat diatur urutannya dengan cara *drag*.



Gambar 2. 4. Layout dan Tab

e. *Properties Bar*

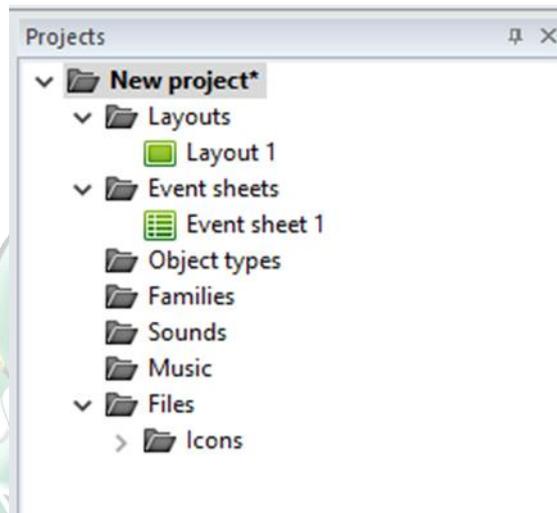
Berisi daftar pengaturan objek yang dapat diubah-ubah sesuai kebutuhan. Isi dari *propertis bar* dapat berbeda-beda, bergantung pada objek apa yang akan dipilih. Opsi pengaturan ditunjukkan dalam kolom kiri, sedangkan di kolom kanan dapat diisi nilai sesuai yang diinginkan. Pada *properties description* terdapat informasi tambahan mengenai fungsi *property* tersebut.



Gambar 2. 5. Properties Bar

f. *Project Bar*

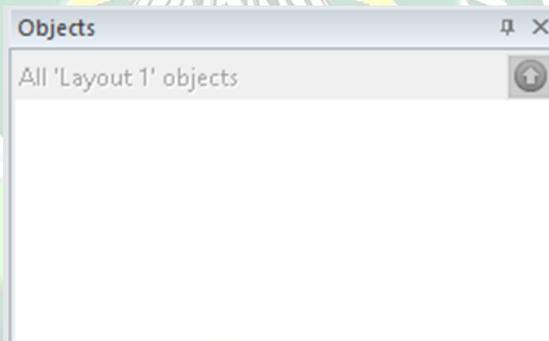
Memberi gambaran umum tentang segala hal dalam proyek *game* yang dibuat, seperti jumlah *event*, *layout*, dan objek-objek yang dimiliki.



Gambar 2. 6. Project Bar

g. *Object Bar*

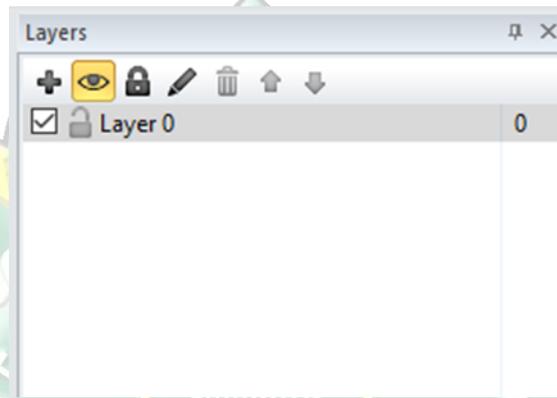
Berfungsi menunjukkan objek secara spesifik, berdasarkan isi suatu folder dalam *project bar*. *Drag and drop* dapat dilakukan jika ingin memasukkan objek ke dalam *layout*.



Gambar 2. 7. Object Bar

*h. Layers Bar*

Digunakan untuk menambah, mengedit, maupun menghapus suatu *layer* dalam *layout*. *Layer* berperan besar untuk menciptakan “kedalaman” di *game* yang dibuat. Semakin besar nilai *layer*, maka semakin besar pula prioritas objek dalam *layout* tersebut.



Gambar 2. 8. Layers Bar

### 2.2.3. Pemrograman Android

Pemrograman android terdiri dari dua kata pemrograman (proses / cara / pembuatan) dan Android merupakan suatu sistem operasi untuk perangkat *mobile* yang berbasis linux untuk mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform open source* atau terbuka bagi para pengembang untuk membuat aplikasi mereka[9].

Tabel 2. 1. Jenis - jenis Versi Android

| No. | OS Android Versi | Nama        |
|-----|------------------|-------------|
| 1.  | Versi 1.0        | Astro       |
| 2.  | Versi 1.1        | Bander      |
| 3.  | Versi 1.5        | Cupcake     |
| 4.  | Versi 1.6        | Donut       |
| 5.  | Versi 2.0        | Éclair      |
| 6.  | Versi 2.2        | Froyo       |
| 7.  | Versi 2.3        | Gingerbread |

|     |           |                    |
|-----|-----------|--------------------|
| 8.  | Versi 3.0 | Honeycomb          |
| 9.  | Versi 4.0 | Ice Cream Sandwich |
| 10. | Versi 4.1 | Jelly Bean         |
| 11. | Versi 4.4 | Kitkat             |
| 12. | Versi 5.0 | Lollipop           |
| 13. | Versi 6.0 | Marshmallow        |
| 14. | Versi 7.0 | Nougat             |
| 15. | Versi 8.0 | Oreo               |
| 16. | Versi 9.0 | Pie                |
| 17. | Versi 10  | Android 10         |

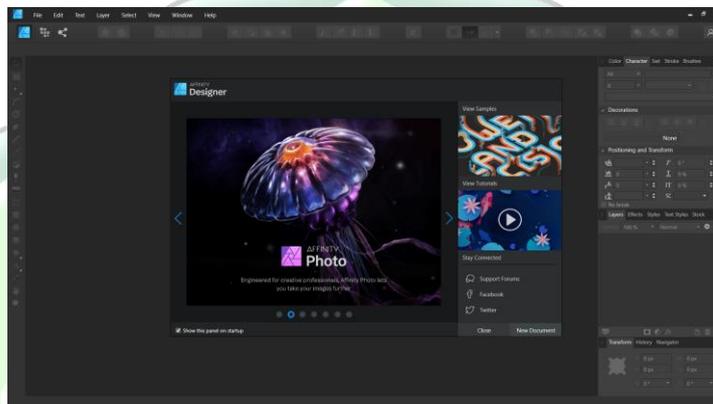
#### 2.2.4. Apache Cordova

*Apache cordova* merupakan *framework* berbasis teknologi web yang digunakan dalam pengembangan aplikasi *smartphone* berbasis android, ios maupun *microsoft*. Salah satu kelebihan dari *apache cordova* yaitu para programmer web tidak perlu lagi menguasai teknologi native dari android untuk mengembangkan aplikasi android. Cukup menguasai pemrograman HTML, Javascript, CSS maupun XML sebagai bahasa pemrograman *front end*. Ketiga Bahasa tersebut berfungsi sebagai pendukung pembuatan *interface* sedangkan PHP dan AJAX digunakan sebagai bahasa pemrograman *back end*. PHP lebih tertuju untuk mendukung koneksi dengan database sedangkan AJAX sebagai pemrograman untuk sinkronisasi antara *client* dan *server*.

*Apache Cordova* dikembangkan untuk digunakan dalam *smartphone multiplatform* baik android, ios maupun *Microsoft*. Dan untuk menjalankan aplikasi ini tidak perlu dalam mode *online* cukup dengan mode *offline* pun bisa digunakan sebagai skenario untuk mengeksekusi pemrograman yang telah dibuat. Dan untuk plugin pun sudah disediakan oleh cordova seperti barcode scanner, map, kamera dll, yang sekiranya dibutuhkan oleh para programmer mobile[10].

### 2.2.5. Affinity Designer

Affinity Designer adalah software editor grafis yang dirancang sebagai alternatif lain dari Adobe Illustrator. Serif didirikan pada tahun 1990 sebagai pengembang perangkat lunak kelas atas untuk desktop PC/Windows dengan biaya rendah untuk pengguna pemula, kini perusahaan tersebut mencoba mengambil pasar perangkat lunak grafis (*software grafis*) dengan menghadirkan Affinity Designer, Photo, dan Publisher.



Gambar 2.9. Affinity Designer

### 2.2.6. Metode Pengembangan Sistem

Dalam perancangan aplikasi metode yang digunakan adalah pengembangan GRAPPLE (*Guidelines for Rapid APPLICATION Engineering*). GRAPPLE merupakan metode pengembangan sistem RAD (*Rapid Application Development*), sehingga sistem informasi manajemen dapat dirancang secara cepat. Metodologi GRAPPLE merupakan metodologi pengembangan sistem yang setiap tahapannya terdiri dari beberapa Tindakan, dan setiap tindakanya menghasilkan produk berupa diagram UML[11].

### 2.1.2. Unified Modelling Language (UML)

*Unified Modelling Language (UML)* adalah suatu metode permodelan secara visual yang berfungsi sebagai sarana perancang sistem berorientasi objek. Definisi UML adalah sebagai suatu bahasa yang sudah menjadi standar pada visualisasi. Perancangan dan juga pendokumentasian sistem aplikasi[12].

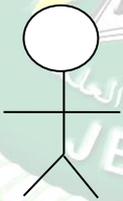
UML sendiri mempunyai banyak diagram yang digunakan untuk melakukan pemodelan data maupun sistem, salah satunya adalah *use case diagram*, *sequence diagram*, *activity diagram*, serta masih banyak lagi.

a. *Use Case Diagram*

*Use Case Diagram* adalah suatu urutan interaksi yang saling berkaitan antara sistem dan *actor*. *Use case* dijalankan melalui cara menggambarkan tipe interaksi antara user suatu program (sistem) dengan sistemnya sendiri. *Use case* melalui sebuah cerita yang mana sebuah sistem itu dipakai. *Use case* juga dipakai untuk membentuk perilaku (*behaviour*) sistem yang akan dibuat. Sebuah *use case* menggambarkan sebuah interaksi antara pengguna (*actor*) dengan sistem yang sudah ada.

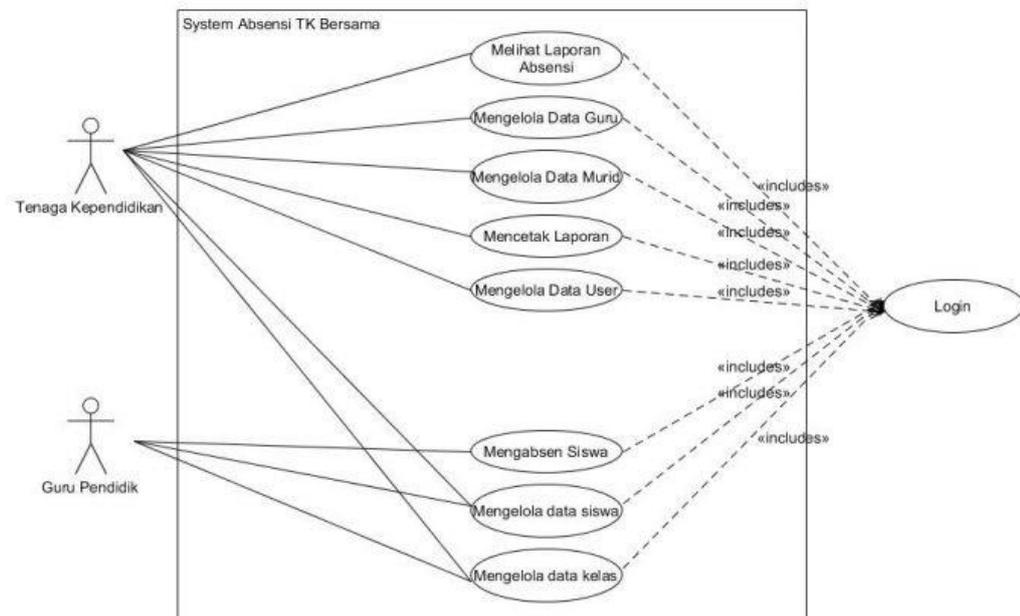
*Use Case Diagram* memaparkan proses kegiatan dan proses bisnis yang dilakukan oleh pengguna (*actor*). Pada diagram *Use Case* ini, setiap proses digambarkan kedalam sebuah *Use Case* berbentuk oval dengan relasi yang ada didalamnya[13][14].

Tabel 2. 2. Tabel Simbol Use Case

| No | Simbol  | Nama               | Deskripsi  |
|----|---|--------------------|--|
| 1  |  | <i>Actor</i>       | Merupakan peran orang, sistem yang lain, atau alat Ketika berhubungan dengan <i>use case</i> . |
| 2  |  | <i>Case</i>        | <i>Abstraksi</i> dari penghubung antara <i>actor</i> dengan <i>use case</i> .                  |
| 3  |  | <i>Association</i> | Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> .  |

|   |  |                     |  |
|---|--|---------------------|--|
| 4 |  | <i>Generalisasi</i> | Menunjukkan <i>spesialisasi actor</i> untuk dapat berpartisipasi dengan <i>use case</i> .  |
| 5 |  | <i>Include</i>      | Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> .                               |
| 6 |  | <i>Extend</i>       | Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi. |

Contoh Use Case Diagram



Gambar 2. 9. Use Case Diagram

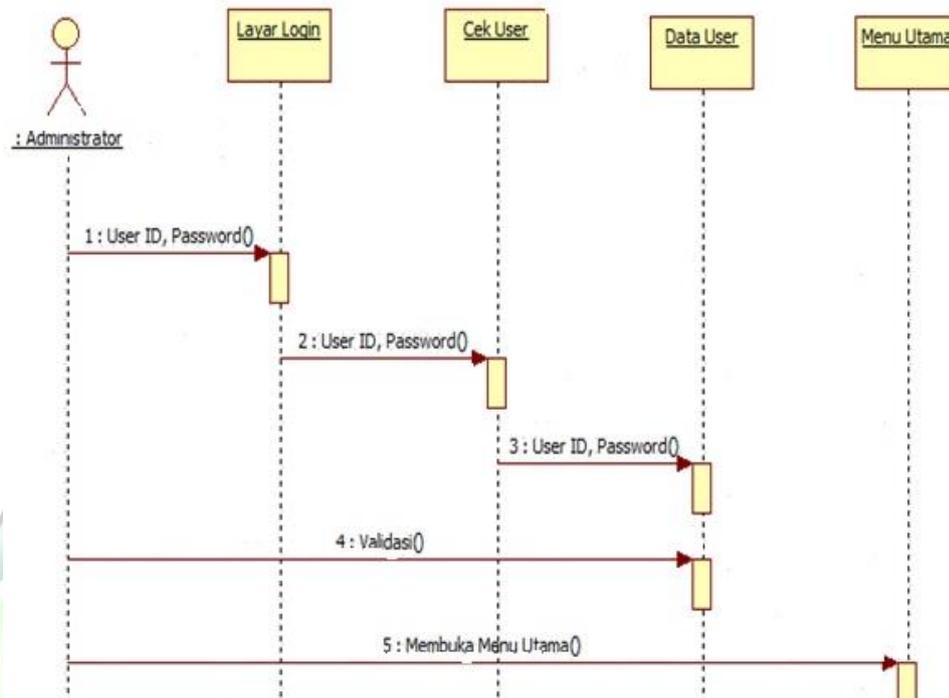
b. *Sequence Diagram*

*Sequence Diagram* adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut. *Sequence diagram* digunakan untuk menjelaskan perilaku pada sebuah skenario dan menggambarkan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang dipakai saat interaksi. Semua pesan digambarkan dalam urutan pada eksekusi [15][14].

Tabel 2. 3. Tabel Simbol Sequence Diagram

| No | Simbol  | Nama            | Deskripsi  |
|----|---|-----------------|--|
| 1  |  | Objek           | Menjelaskan bagaimana sebuah objek beroperasi pada sebuah sistem.              |
| 2  |  | Pesan           | Menjelaskan alur pesan yang dikirim antar objek.                               |
| 3  |  | Kotak Aktivasi  | Menjelaskan tentang lama waktu yang diperlukan dalam mengerjakan suatu projek. |
| 4  |  | <i>Lifeline</i> | Menjelaskan tentang aktivitas dari objek.                                      |

### Contoh dari *Sequence Diagram*



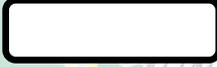
Gambar 2. 10. Sequence Diagram

#### c. *Activity Diagram*

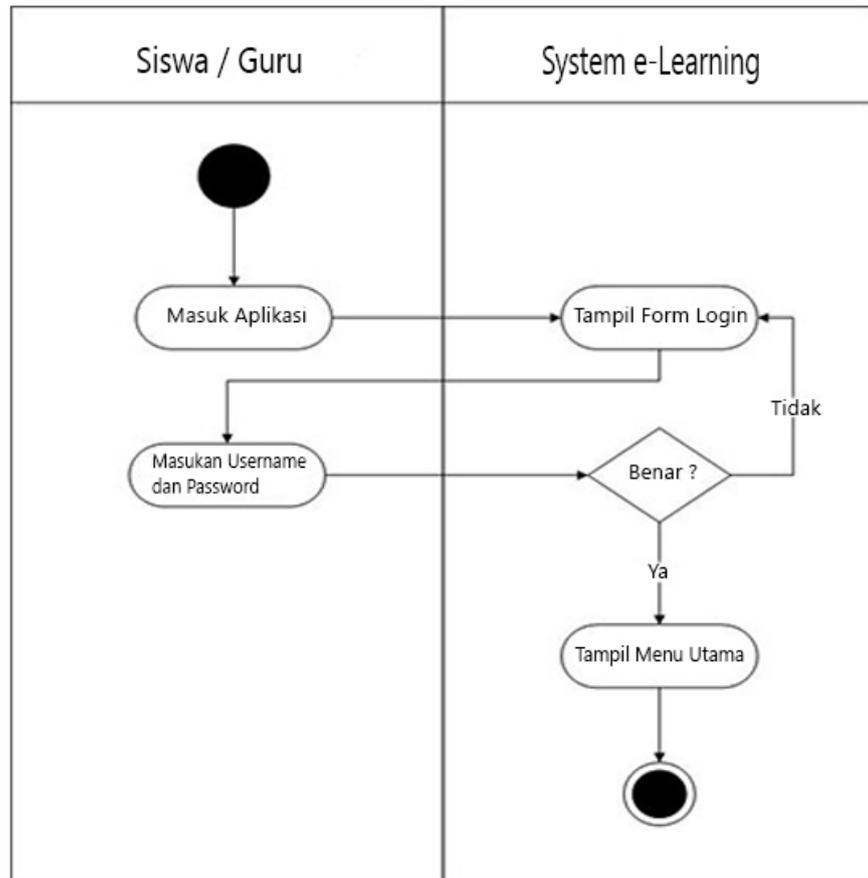
*Activity Diagram* merupakan sesuatu yang menjelaskan tentang alir kegiatan dalam program yang sedang dirancang, bagaimana proses alur berwal, keputusan yang mungkin terjadi, dan bagaimana sistem akan berakhir.

*Activity Diagram* juga dapat menjelaskan metode paralel yang mungkin terjadi pada eksekusi. *Activity diagram* adalah state diagram khusus, yang mana state ini berfungsi sebagai action dan Sebagian besar transisi ditrigger oleh akhir state sebelumnya (*internal processing*)[16][14].

Tabel 2. 4. Tabel Simbol Activity Diagram

| No | Simbol  | Nama                  | Deskripsi   |
|----|---|-----------------------|---|
| 1  |    | <i>Start State</i>    | Menunjukkan dimulainya suatu proses kerja suatu sistem.           |
| 2  |    | <i>Flow Control</i>   | Menunjukkan alur kerja dari suatu aktivitas ke aktivitas lainnya. |
| 3  |   | <i>Activity State</i> | Menunjukkan Aktivitas dalam suatu alur kerja.                     |
| 4  |  | <i>Action State</i>   | Menunjukkan Tahap-tahap pada sebuah activity.                     |
| 5  |  | <i>End State</i>      | Menunjukkan berakhirnya suatu proses kerja dalam suatu sistem.    |

Contoh dari *Activity Diagram*



Gambar 2. 11. Activity Diagram

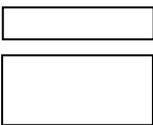
d. *Class Diagram*

Merupakan visual dari struktur sistem program pada jenis-jenis yang di bentuk. *Class Diagram* merupakan alur jalannya database pada sebuah sistem.

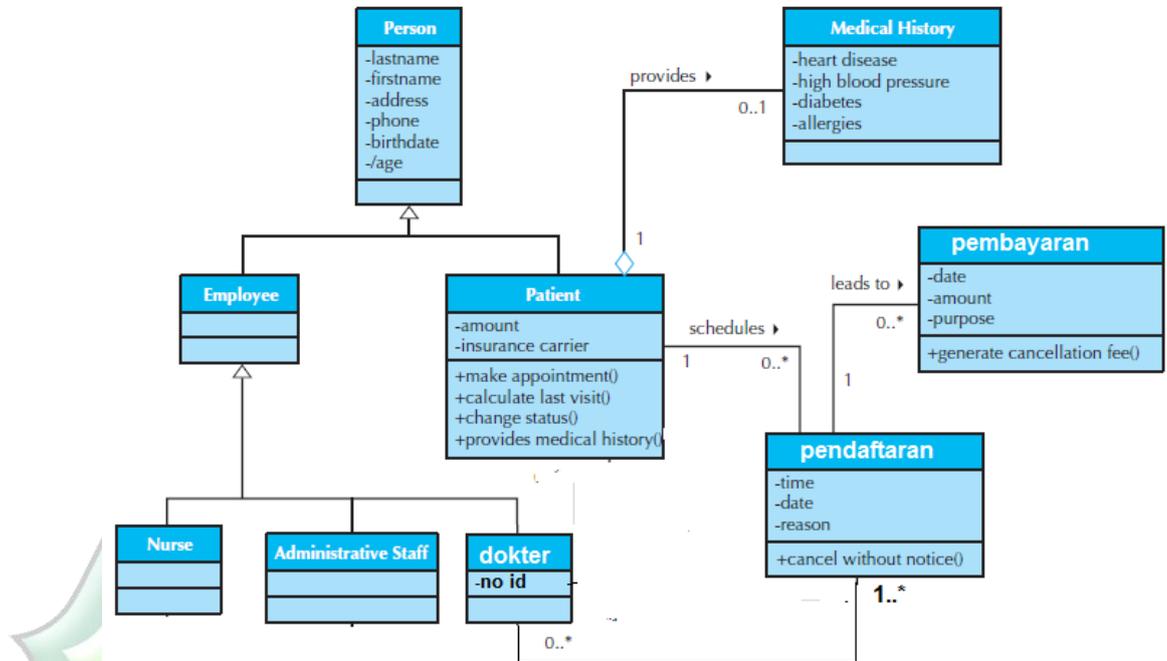
*Class Diagram* merupakan penjelasan proses database dalam suatu program. Dalam sebuah laporan sistem maka *class diagram* ini wajib ada, *class diagram* menjelaskan struktur sistem dari segi pendefinisian *class-class* yang akan dibuat untuk membangun sebuah sistem[17][14].

Tabel 2. 5. Tabel Simbol Class Diagram

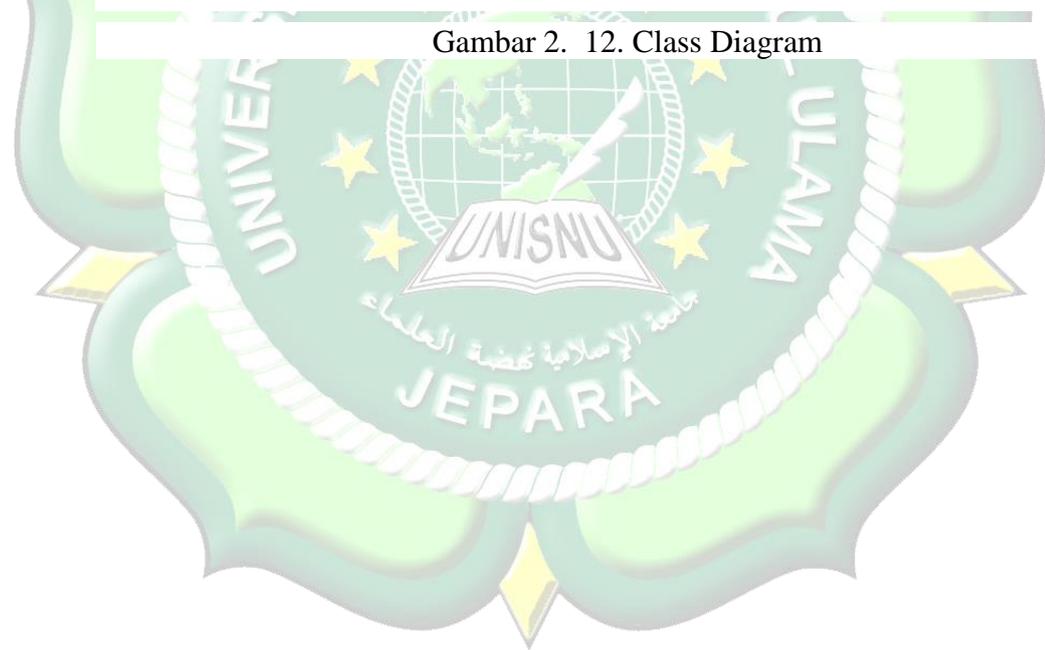
| No | Simbol | Nama            | Deskripsi   |
|----|--------|-----------------|---|
| 1  | _____  | <i>Asosiasi</i> | Menunjukkan relasi antar kelas dengan makna umum. |

|   |   |                     |   |
|---|---|---------------------|---|
| 2 |    | <i>Class</i>        | Merupakan <i>class</i> pada struktur sistem.  |
| 3 |    | <i>Agregasi</i>     | Menunjukkan relasi antar <i>class</i> dengan makna semua bagian.  |
| 4 |    | <i>Generalisasi</i> | Menunjukkan relasi antar <i>class</i> dengan makna <i>generalisasi – spesialisasi</i> (Umum Khusus).              |
| 5 |  | Antarmuka           | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.   |
| 6 |  | Asosiasi Berarah    | Menunjukkan relasi antar <i>classi</i> dengan makna <i>class</i> yang satu digunakan oleh <i>class</i> yang lain. |

Contoh dari *Class Diagram*

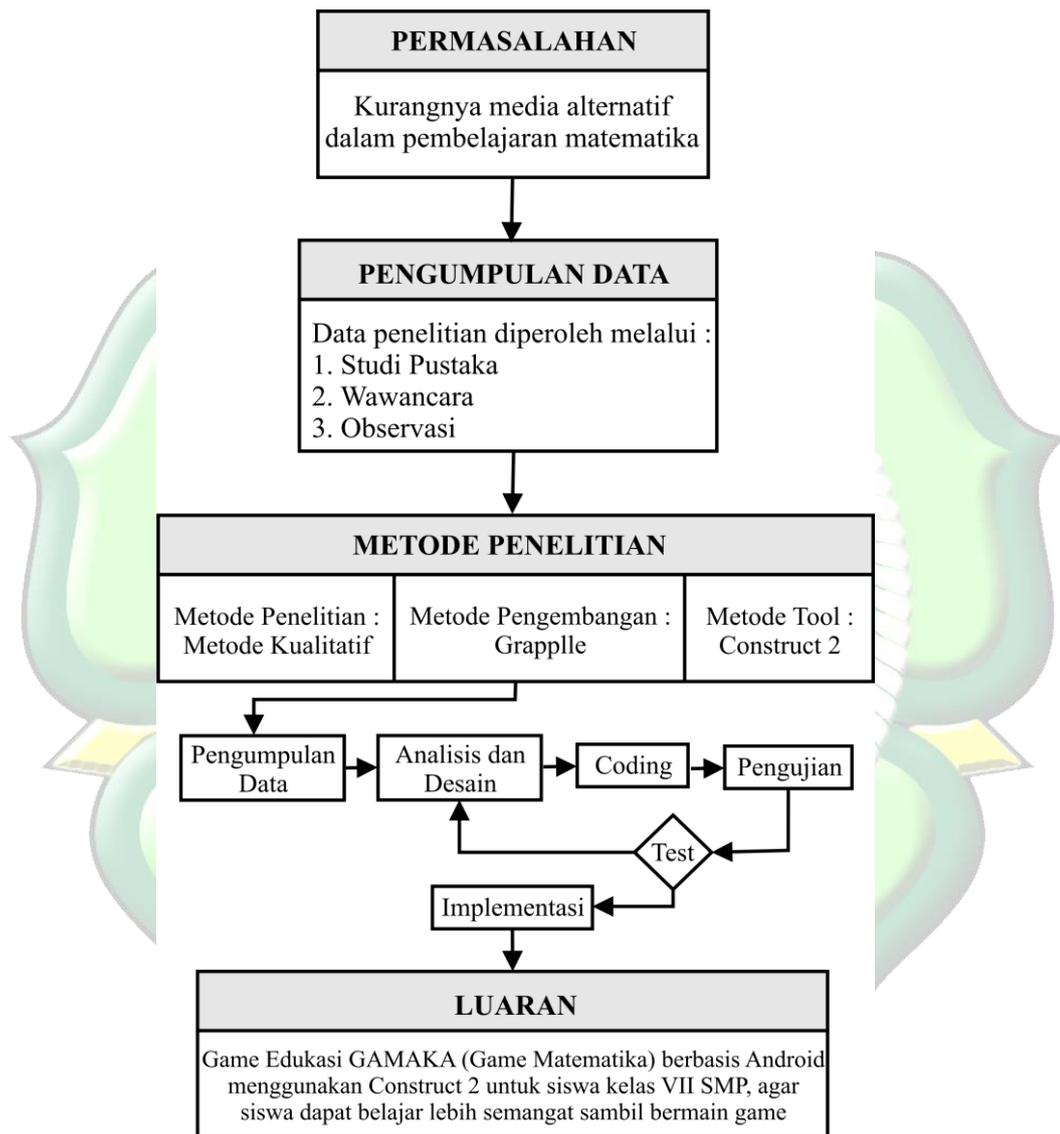


Gambar 2. 12. Class Diagram



## 2.1. Kerangka Pemikiran

Kerangka pemikiran ini merupakan suatu bentuk pemikiran dari penulis dimulai dari awal pembuatan judul sampai dalam tahap melakukan penelitian. Kerangka Pemikiran adalah suatu bagian alur yang menghubungkan masalah dan pendekatan penelitian yang dihasilkan dari teori yang ada di landasan teori.



Gambar 2. 13. Kerangka Pemikiran