

## BAB II

### TINJAUAN PUSTAKA

#### II.1. Penelitian Terdahulu

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu :

Berdasarkan penelitian dari Heni Vidia Sari, 2017, dengan judul “Pengembangan Media Pembelajaran Berbasis Web Untuk Mengukur Hasil belajar Siswa Pada Mata Pelajaran Komputer Jaringan Dasar Program Keahlian Teknik Komputer Dan Jaringan” Penelitian dan pengembangan ini didasari oleh terbatasnya media pembelajaran untuk mata pelajaran Jaringan Dasar. Tujuan penelitian dan pengembangan ini adalah (1) mengembangkan dan menghasilkan media pembelajaran berbasis web pada mata pelajaran Jaringan Dasar, (2) mengetahui kelayakan media pembelajaran berbasis web yang dikembangkan, dan (3) mengetahui efisiensi penggunaan media pembelajaran berbasis web yang dikembangkan terhadap hasil belajar siswa. Model penelitian dan pengembangan media pembelajaran berbasis web ini menggunakan model pengembangan *ADDIE*.

Berdasarkan penelitian dari Anggara Sukma Ardiyanta, 2017, dengan judul “*Desain Prototipe Media Pembelajaran Simulasi Sistem Rem Mobil Untuk Pembelajaran Siswa Smk Jurusan Otomotif*”. Penelitian ini merupakan penelitian

pengembangan media pembelajaran berbasis game simulasi. Simulasi pada dasarnya adalah aplikasi untuk mewakili benda atau sistem kerja yang sesungguhnya. Tahapan penelitian yang dilakukan oleh peneliti menggunakan metode waterfall, dimulai dari studi pustaka, pembuatan objek 3 dimensi berdasarkan bentuk sistem yang sesungguhnya, kemudian dilanjutkan dengan pengintegrasian objek agar dapat dimainkan. Peneliti mengembangkan media pembelajaran simulasi sistem rem mobil dengan menggunakan perangkat lunak Blender3D dan dikombinasikan dengan Unity3D dengan bahasa pemrograman menggunakan python. Blender 3D dipilih karena kemudahan dan kebutuhan sistem yang relatif kecil, sedangkan Unity3D dipilih karena kemudahan interface. Hasil penelitian ini adalah purwarupa dari aplikasi sistem rem dalam bentuk gambar animasi 3D.

Berdasarkan penelitian dari Abdul Haris Pito, 2018, dengan judul “Media Pembelajaran Dalam Perspektif Al-quran” Penelitian ini bertujuan menggambarkan penggunaan media dalam pandangan Islam. Pada dunia pendidikan, guru dituntut agar mampu menggunakan media pembelajaran, yang mengikuti perkembangan zaman, dan sesuai dengan kondisi siswa. Menggunakan metode deskriptif tulisan ini berusaha menggambarkan kesesuaian teori tentang media pembelajaran dengan penggunaan media dalam Alquran dan Hadis. Dalam pembahasan ditemukan banyak ayat-ayat Alquran dan hadis nabi yang sejalan dengan teori tentang penggunaan media di zaman sekarang seperti penggunaan media audio, visual, sampai kepada media pembelajaran berbasis teknologi multimedia.

Berdasarkan penelitian dari Perancangan Aplikasi Alat Musik Tradisional Berbasis Android terdapat informasi tentang jenis alat musik, cara pembuatan alat musik dan cara memainkan alat musik dapat di manfaatkan sebagai media belajar bagi masyarakat luas khususnya generasi muda. Metode pengumpulan data yaitu Metode Studi Pustaka, Wawancara dan Observasi serta menggunakan Spiral Methode sebagai metode pengembangan perangkat lunak dibangun dengan Android Studio versi 3.0.1 dan basis data SQ Lite versi 3.8.2 sebagai Database Management System (DBMS) dengan target minimum Android versi 4.4 (Kitkat). metode pengujian menggunakan metode pengujian Beta Test. Dapat diakses dari Android Market yaitu Google Play Store(Harri Gunawan,2018; 73)

Sedangkan penelitian yang dilakukan penulis bertujuan untuk Alat musik tradisional adalah instrumen atau alat yang sengaja diciptakan atau diadaptasikan dengan tujuan supaya dapat menghasilkan suara musik. Alat musik sengaja dibuat bahkan dari bentuk, gaya dan juga menggunakan bahan-bahan yang berbeda-beda. Menurut sejarah alat musik pada awalnya dibuat dari benda-benda disekitar yang mudah ditemukan seperti kerang atau kulit-kulit binatang dan juga bagian tanaman. Seiring berkembangnya zaman alat musik berevolusi dengan muncul berbagai macam variasi dan kualitas bahanpun ikut diperhatikan (Risyan Arief Setyawan, 2018 ; 296)

## **II.2. Landasan Teori**

### **II.2.1. Aplikasi**

Istilah aplikasi berasal dari bahasa Inggris *application* yang berarti penerapan, lamaran ataupun penggunaan. Sedangkan secara umum, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta jasa pengguna aplikasi lain yang dapat digunakan oleh pengguna yang akan dituju. Menurut kamus komputer eksekutif, pengertian aplikasi merupakan pemecahan masalah yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Aplikasi biasanya berupa perangkat lunak yang berbentuk *software* yang berisi kesatuan perintah atau program yang dibuat untuk melaksanakan sebuah pekerjaan yang diinginkan. (Ayu Rizka Inayah, 2017 : 3).

## **II.4. Musik Tradisional**

Musik tradisional adalah musik yang berkembang di seluruh wilayah kepulauan dan merupakan kebiasaan turun-temurun yang masih dijalankan dalam masyarakat. Dengan demikian, penulis menyimpulkan pengertian musik tradisional adalah musik dan tradisi yang memiliki arti ilmu atau seni menyusun nada atau suara dalam urutan, kombinasi, dan hubungan temporal untuk menghasilkan komposisi (suara) yang mempunyai kesatuan dan kesinambungan (Wahyuzi Andriansyah, 2017;2).

### II.2.2. Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi *Android* dan bersifat *open source* atau gratis. Peluncuran *Android Studio* ini diumumkan oleh *Google* pada 16 mei 2013 pada event *Google I/O Conference* untuk tahun 2013. Sejak saat itu, *Android Studio* menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi *Android*.

*Android studio* sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT *plugin* (*Android Development Tools*). *Android studio* memiliki fitur :

- a. Projek berbasis pada *Gradle Build*
- b. *Refactory* dan pembenahan bug yang cepat
- c. *Tools* baru yang bernama “*Lint*” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung *Proguard And App-signing* untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah
- f. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan. (Andi Juansyah : 2017)

#### II.2.2.1. Sejarah *Android*

*Android* adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*,

pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode-kode Android di bawah *lisensi Apache*, sebuah *lisensi* perangkat lunak dan standar terbuka perangkat seluler. (Harni Kusniyati, dkk : 2018).

*Android* SDK merupakan tools bagi para *programmer* yang ingin mengembangkan aplikasi berbasis *google Android*. *Android* SDK mencakup seperangkat alat pengembangan yang komprehensif. *Android* SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan tutorial. . (Harni Kusniyati, dkk : 2018).

#### **II.2.2.2. Urutan Versi OS *Android***

##### **1. *Android* 1.0 & 1.1 : Astro (Alpha) & Bender (Beta)**

Kedua versi awal *Android* ini mungkin agak asing kamu dengarkan. Pasalnya versi *Android* 1.0 *Astro (Alpha)* dan *Android* 1.1 *Bender (Beta)* ini belum diluncurkan secara publik untuk kebutuhan komersil. *Platform Android* sendiri pertama kali diluncurkan pada September 2008 dengan andil Andy Rubin yang saat ini dikenal sebagai Bapak *Android*. Walaupun belum menggunakan nama makanan

manis, kedua sistem operasi *Android* ini tentu menjadi *pionir*. Pasalnya di sinilah *Android* bermula lewat *smartphone* pertama, *HTC Dream*.

## **2. Android 1.5 : Cupcake**

Mulai dari versi ini, *Android* menggunakan nama makanan manis untuk setiap versi yang diluncurkan. *Android 1.5 Cupcake* sendiri dirilis pada tanggal 30 April 2009 dengan berbagai fitur di sebuah perangkat *smartphone* untuk menggantikan *featured phone* kala itu.

## **3. Android 1.6 : Donut**

Tentu pada awal perilisannya, sistem operasi *Android* tetap memiliki banyak *bug* yang pengembangnya perlu mengadakan perbaikan. Hal ini dilakukan pada *Android 1.6 Donut* yang dirilis pada 15 September 2009. Ya artinya belum genap setahun semenjak perilsan *Android 1.5 Cupcake* atau hanya berselang 5 bulan saja. *Android* pun menambahkan beberapa pembaruan, terutama dukungan pada layar *smartphone* yang lebih besar.

## **4. Android 2.0 & 2.1 : Eclair**

Sama seperti versi sebelumnya, *Android 2.0 & 2.1 Eclair* masih berfungsi untuk menutupi *bug* yang masih ditemukan pada sistem operasi *mobile* ini. Di samping itu, *Android* juga menambah berbagai fitur di dalamnya.

Mulai dari dukungan *Bluetooth* hingga fitur kamera yang mulai menjadi nilai jual *smartphone* kala itu. *Android 2.0 & 2.1 Eclair* digunakan pada perangkat seperti *HTC Nexus One*.

## 5. **Android 2.2 : Froyo (Frozen Yoghurt)**

Mulai versi ini *Android* tampaknya sudah mulai dikenal luas oleh berbagai *brand smartphone*. *Android 2.2 Frozen Yoghurt* alias *Froyo* ini dirilis pertama kali pada tanggal 20 Mei 2010. Walaupun sudah mulai dipergunakan pada beberapa *brand*, namun tetap saja *Android* masih kalah bersaing dengan *Symbian* yang mendominasi pasar *featured phone*. *Android 2.2 Froyo* memberikan peningkatan pada kecepatan kerja, fitur *USB tethering*, *WiFi hotspot*, serta fitur keamanan.

## 6. **Android 2.3: Gingerbread**

Belum setahun berselang lagi, *Android 2.3 Gingerbread* kembali diluncurkan pada Desember 2010 dengan berbagai peningkatan yang cukup signifikan. Hal ini terutama pada tampilan tatap muka alias *user interface* yang digunakan. Mulai versi ini, banyak *brand smartphone* mulai melirik menggunakan sistem operasi *Android*. Salah satunya *Samsung Galaxy Series* yang populer hingga saat ini.

## 7. **Android 3.0 & 3.2: Honeycomb**

Untuk para pengguna *smartphone* mungkin akan agak asing dengan versi *Android* yang satu ini. *Android 3.0 & 3.2 Honeycomb* yang menggunakan ikon lebah ini memang diperuntukkan penggunaannya khusus untuk perangkat *tablet*. Tentu perilsan *Android 3.0 & 3.2 Honeycomb* pada 10 Mei 2011 ini untuk mendukung *Samsung* yang mulai merilis perangkat *tablet*

## 8. Android 4.0 : Ice Cream Sandwich

Selain itu, *Android* pun juga merilis versi *Android 4.0 Ice Cream Sandwich* yang kembali diperuntukkan untuk perangkat *smartphone*. *Android 4.0 Ice Cream Sandwich* sendiri dirilis pada 19 Oktober 2011 silam. Punya versi nama paling panjang hingga saat ini, *Android 4.0 Ice Cream Sandwich* memberikan banyak pembaruan. Mulai dari animasi yang semakin halus, sederhana, dan mudah digunakan.

## 9. Android 4.1 & 4.3 : Jelly Bean

Peningkatan signifikan terasa saat menggunakan *Android 4.1 & 4.3 Jelly Bean*. Sistem operasi ini sendiri pertama kali dirilis pada Juni 2012 dengan membawa sejumlah peningkatan terutama di sektor pengolahan grafis. Dengan begini, tentu *Android 4.1 & 4.3 Jelly Bean* bisa memberikan peningkatan fungsi pada *user interface* dan *teknologi Vsync* yang digunakannya.

## 10. Android 4.4 : KitKat

Menggunakan nama *brand* cemilan terkenal, *Android 4.4 KitKat* pertama kali dirilis pada Oktober 2013. Versi *Android* terbaik ini pun bisa dikatakan menjadi favorit karena mendukung hampir seluruh perangkat *smartphone* di dunia. Hal ini dikarenakan *Android 4.4 KitKat* dapat memberikan optimalisasi yang baik termasuk pada perangkat *smartphone* yang memiliki *spesifikasi* kurang mumpuni alias cukup rendah saat itu.

## 11. Android 5.0 & 5.1: Lollipop

Mulai beberapa versi ke belakang, *Android* dan *Google* pun mulai secara rutin memperbarui sistem operasi mereka dalam selang waktu setahun.

Termasuk *Android 5.0 & 5.1 Lollipop* yang dirilis dan diresmikan pada Juni 2014. Bisa dibilang *Android 5.0 & 5.1 Lollipop* menjadi *pionir* dibuatnya *smartphone flagship* dengan *spesifikasi* cukup mumpuni. Versi *Android* ini sudah mendukung arsitektur 64-bit yang sudah memungkinkan penggunaan RAM di atas 3GB. Salah satunya *ASUS Zenfone 2* yang sudah mengusung RAM 4GB saat itu.

## **12. Android 6.0 : Marshmallow**

*Android 6.0 Marshmallow* menjadi suksesor dari versi *Android* sebelumnya. Sistem operasi ini sendiri pertama kali diperkenalkan pada Mei 2017 dan mulai dirilis pada Oktober 2017 silam. Sistem operasi ini secara jelas memberikan peningkatan pada sistem keamanan dengan dihadirkannya *fingerprint sensor* sebagai sistem keamanan *biometrik* yang digunakan. Selain digunakan untuk mengunci layar, *fingerprint sensor* ini dapat digunakan untuk otentikasi *Google Play Store* dan pembelian dengan menggunakan *Android Pay*.

## **13. Android 7.0 & 7.1 : Nougat**

Untuk saat ini, sistem operasi *Android* ini masih digunakan beberapa *smartphone* yang baru dirilis belakangan ini. *Android 7.0 & 7.1 Nougat* pertama kali diperkenalkan pada Juni 2018 dengan menampilkan ikon robot *Android* dengan batangan *Nougat*. Sistem operasi *Android 7.0 & 7.1 Nougat* mengalami perubahan dari segi tampilan antarmuka. Selain itu ada juga *fitur splitscreen* untuk membagi tampilan layar untuk dua aplikasi sekaligus.

## **14. Android 8.0 & 8.1 : Oreo**

*Android 8.0 & 8.1 Oreo* menjadi sistem operasi *Android* paling terbaru yang banyak digunakan saat ini. Sistem operasi ini dirilis secara stabil mulai Agustus

2017 sudah mengalami pembaruan lewat versi *Android 8.1 Oreo*. *Android 8.0 & 8.1 Oreo* menjadi versi *Android* kedua yang menggunakan makanan manis dari nama *brand* terkenal setelah *Android 4.4 KitKat*. Sistem operasi ini menawarkan pengalaman *multitasking* yang makin mumpuni dibanding versi sebelumnya.

Selain itu ada juga *Project Treble* yang memungkinkan pengguna mendapat pembaruan lebih cepat.

### **15. Android 9.0 : Pie**

Kemudian ada *Android 9.0 Pie* yang secara resmi diperkenalkan pada Agustus 2018. Sistem operasi *Android* ini memberi banyak perubahan, terutama untuk HP dengan desain baru. Misal *Android 9.0 Pie* memberikan navigasi berupa *gesture* yang menggantikan tombol fisik *Home*, *Back*, dan *Recent Apps*. Fitur lainnya yang cukup berguna adalah sistem notifikasi, pengatur kecerahan, hingga sistem *screenshot* terbaru yang lebih memudahkan.

### **16. Android Q Beta**

Terakhir ada *Android Q Beta* yang baru diluncurkan pada 13 Maret 2019 dan saat ini masih terbatas pada beberapa perangkat HP *Android* saja. Seperti pada seri *smartphone Google*, yakni *Google Pixel*, *Google Pixel XL*, *Google Pixel 2*, *Google Pixel 2 XL*, *Google Pixel 3*, *Google Pixel 3 XL*, dan *Google Pixel 3 Lite*. Salah satu fitur *Android Q Beta* adalah *Dark Mode* alias mode gelap yang diklaim mampu meningkatkan performa baterai. (Harni Kusniyati Dan Dicky : 2018)

*Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak.

## 1. Kelebihan *Android*

- a. *Switching* dan multitasking yang lebih baik *Android* sangat mendukung *multitasking* aplikasi, kini hal tersebut kembali ditingkatkan. Dalam *Honeycomb* pengguna dapat dengan mudah berpindah aplikasi hanya dengan menyentuh sebuah *icon* pada system bar.
- b. Kapasitas yang lebih baik untuk beragam *widget* Kapabilitas terhadap beragam *widget* dijanjikan bakal makin memanjakan para penggunanya. Contohnya *widget* untuk email *Gmail* yang dipamerkan *Google*, pengguna tidak perlu membuka aplikasi *Gmail* untuk melihat isi di dalamnya.
- c. Peningkatan kemampuan *copy-paste* Beberapa seri *Android* terdahulu memang sudah bisa melakukan *copy-paste*, namun beberapa pengguna masalah pemilihan teks yang agak sulit. Kini hal tersebut coba diselesaikan, selain *copy-paste* *Google* juga menambah *share it* pada teks yang diseleksi.
- d. *Browser Chrome* Lebih Cepat Ada satu fitur yang hilang dalam browser *Chrome* yang diletakkan pada *Android* terdahulu, kemampuan Tab. *Chrome* yang ada di *Honeycomb* kini dapat melakukan hal tersebut. Selain itu pengguna juga bisa mensinkronisasi antara *browser* di ponsel dengan *Chrome* yang ada di komputer.
- e. Notifikasi yang mudah terdengar. Dengan layar yang lebih besar, otomatis membuat *google* lebih leluasa menempatkan notifikasi pada layar.
- f. Peningkatan *Drag and Drop* serta *Multitouch* Ukuran layar yang lebih besar, menuntut *google* untuk meningkatkan kemampuan *multitouch* di dalam android, tak terkecuali fitur *drag and drop*. Pada demo yang ditayangkan,

pengguna bisa melakukan *drag and drop* untuk memindahkan *email* di dalam aplikasi *gmail*.

## 2. Kekurangan *Android*

- a. Koneksi Internet yang terus menerus. Kebanyakan ponsel *Android* memerlukan koneksi internet yang simultan atau terus menerus aktif, itu artinya anda harus siap berlangganan paket GPRS 20 Jurnal Sains dan Teknologi Utama, Volume XI, Nomor 1, April 2018 yang sesuai dengan kebutuhan dan batre yang boros karena GPRS yang terus menyala.
- b. Iklan. Aplikasi di Ponsel *Android* memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut, akan selalu ada Iklan yang terpampang. (Sulihati , 2018).

### II.2.3. Basis Data (*Database*)

Pangkalan data atau basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi. Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Model yang umum digunakan sekarang adalah model relasional, yang mewakili semua informasi

dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel (Neni Purwati dan Hendra Kurniawan, 2017 : 50).

#### II.2.4. Normalisasi

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data / *database*, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik tanpa redundansi. Tujuan normalisasi adalah mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah modifikasi data.

##### 1. Proses Normalisasi

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

##### 2. Tahapan Normalisasi :

- a) Bentuk tidak normal : Menghilangkan perulangan grup.

Suatu tabel dikatakan dalam keadaan *first normal form* (1NF) jika :

- 1) Tidak ada perulangan *record* data dalam tabel.

- 2) Setiap sel memiliki satu nilai saja. Artinya tidak ada perulangan group dan *array*.
  - 3) Data yang diinputkan memiliki tipe data yang sama dengan tipe data kolom dalam tabel.
- b) Bentuk Normal pertama (1NF) : Menghilangkan ketergantungan sebagian.  
Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ke satu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.
  - c) Bentuk Normal kedua (2NF) : Menghilangkan ketergantungan transitif.  
Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan *key* sudah tergantung penuh terhadap *key*-nya.
  - d) Bentuk Normal ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan *key* tidak tergantung transitif terhadap *key*-nya.  
(Mukhlisulfatih Latief : 2018)

## II.2.6. MySQL

MySQL (*My Structure Query Language*) merupakan sebuah program pembuat database yang bersifat *Open Source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *Windows* maupun *linux*. *MySQL* juga merupakan sebuah perangkat lunak sistem manajemen basis

data *SQL* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi *user*. *MySQL* juga sering dikenal dengan nama sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam table yang terpisah. Tabel -table tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa table dalam suatu permintaan. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan sebagainya dapat digunakan aplikasi berbasis web seperti *PHP MyAdmin* dengan aplikasi *XAMPP*. (Saipul Anwar, 2018)

### II.2.7. PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, *Java*, dan *Perl* serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* yang akan menterjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain PHP adalah *akronim* dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML”. (Astria Firman, 2018 : 30).

### II.3. UML (Unified Modelling Language)

*Unified Modelling Language* (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi.

*Unified Modeling Language* (UML) biasa digunakan untuk :

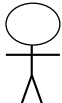
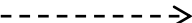
- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.

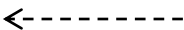
- f. Menyampaikan atau memperluas *functionality* dengan *stereotypes*. (Omni Alfina dan Fitriana Harahap : 2019)

### II.3.1. Use Case Diagram

*Use case* diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

**Tabel II.1. Simbol Use Case**

| Gambar                                                                              | Keterangan      | Deskripsi                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Use case</i> | Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .         |
|  | Aktor           | Sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. |
|  | Asosiasi        | Penghubung antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.     |
|  | Asosiasi        | Penghubung antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.                                                        |
|  | <i>Include</i>  | Merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.                                 |

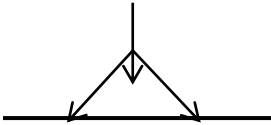
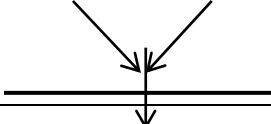
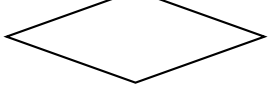
|                                                                                   |               |                                                                                   |
|-----------------------------------------------------------------------------------|---------------|-----------------------------------------------------------------------------------|
|  | <i>Extend</i> | Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi. |
|-----------------------------------------------------------------------------------|---------------|-----------------------------------------------------------------------------------|

(Sumber : Ade Hendini, 2018)

### II.3.2. Activity Diagram

*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

**Tabel II.2. Simbol Activity Diagram**

| Gambar                                                                              | Keterangan                    | Deskripsi                                                                                                                       |
|-------------------------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|    | <i>Start point</i>            | Diletakkan pada pojok kiri atas dan merupakan awal aktifitas.                                                                   |
|   | <i>End point</i>              | Akhir aktifitas.                                                                                                                |
|  | <i>Activites</i>              | Menggambarkan suatu proses/kegiatan bisnis.                                                                                     |
|  | <i>Fork</i><br>(Percabangan). | Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu. |
|  | <i>Join</i><br>(penggabungan) | Digunakan untuk menunjukkan adanya dekomposisi.                                                                                 |
|  | <i>Decision Points</i>        | Menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .                                                 |
|  | <i>Swimlane</i>               | Pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.                                                        |

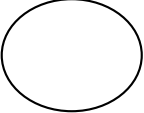
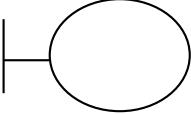
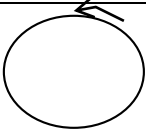
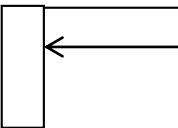
(Sumber : Ade Hendini, 2018 : 109)

### II.3.3. Sequence Diagram

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar

objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

**Tabel II.3. Simbol *Sequence Diagram***

| Gambar                                                                              | Keterangan            | Deskripsi                                                                                                                                                               |
|-------------------------------------------------------------------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <i>Entity Class</i>   | Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.      |
|    | <i>Boundary Class</i> | Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.        |
|   | <i>Control class</i>  | Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. |
|  | <i>Message</i>        | Simbol mengirim pesan antar <i>class</i> .                                                                                                                              |
|  | <i>Recursive</i>      | Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.                                                                                                      |
|  | <i>Activation</i>     | Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.                                                 |
|  | <i>Lifeline</i>       | Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .                                                                   |

(Sumber : Ade Hendini, 2018 : 110)

### II.3.4. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalisation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2018 : 111).

**Tabel II.4. Multiplicity Class Diagram**

| <b>Multiplicity</b> | <b>Penjelasan</b>                                               |
|---------------------|-----------------------------------------------------------------|
| 1                   | Satu dan hanya satu                                             |
| 0..*                | Boleh tidak ada atau 1 atau lebih                               |
| 1..*                | 1 atau lebih                                                    |
| 0..1                | Boleh tidak ada, maksimal 1                                     |
| n..n                | Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4 |

**(Sumber : Ade Hendini, 2018 : 110)**