

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Berdasarkan penelitian yang dilakukan Fitri Nuraeni (2018) dengan judul “Perancangan Aplikasi Penagihan Angsuran Kredit Berbasis Android Pada Perusahaan Pembiayaan” Seperti halnya aplikasi penagihan angsuran kredit yang dibuat untuk memenuhi kebutuhan Permata *Finance* dalam menyampaikan informasi pada nasabahnya. Aplikasi ini dirancang dengan menggunakan metode pendekatan terstruktur menggunakan *flowmap*, diagram konteks dan basis data digambarkan melalui relasi antar table. Untuk aplikasi di client dibangun berbasis android sedangkan untuk administator menggunakan aplikasi berbasis web. Dengan menggunakan aplikasi ini, informasi tagihan angsuran dan tanggal jatuh tempo lebih awal kepada nasabah sehingga dapat mencegah keterlambatan pembayaran ataupun penunggakan pembayaran.

Berdasarkan penelitian Akhmad Rizaldi Firdaus (2018) dengan judul “Pengembangan Aplikasi Pembayaran Pulsa Telepon Seluler Dan Pulsa Listrik Berbasis Android Menggunakan Framework Xamarin’ Dengan berkembangnya dan serta berevolusinya media komunikasi saat ini membuat suatu kegiatan yang dahulu sulit untuk dilakukan, sekarang dapat dilakukanya dengan mudah dimulai dari era telekomunikasi yang dahulu hanya menggunakan jaringan pesawat

telepon antar desa, namun dengan kendala biaya banyak orang yang sudah meninggalkan pemasangan jaringan pesawat telepon dan dengan kendala itu distributor pembuatan *Handphone* di dunia memulai debutnya untuk menciptakan telepon seluler yang menggunakan pengisian pulsa untuk melakukan panggilan antar telepon dan pengiriman pesan singkat, namun disaat pengguna ingin melakukan pengisian pulsa tetapi terhalang oleh kendala waktu dan tempat sehingga pengguna harus menunggu atau mencari dimana tempat yang dapat melakukan pengisian ulang pulsa.

Berdasarkan penelitian yang dilakukan Rendy Ryan, dkk (2018) dengan judul Aplikasi Pemesanan Air Mineral Berbasis Android Pada PT. Citra Golden Tunggal Pangkalpinang, Pada penelitian ini, dibangun sebuah aplikasi pada telepon seluler pintar berbasis Android. Tujuannya adalah untuk menjawab kebutuhan akan teknologi yang memudahkan dalam berbelanja secara praktis dan mobile. Model pengembangan aplikasi dengan menggunakan waterfall, dan menggunakan metode berorientasi objek untuk pengembangan aplikasinya. Pengujian dilakukan dengan menggunakan teknik Blackbox. Hasil yang diharapkan dengan adanya aplikasi ini dapat mempermudah pelanggan untuk melakukan pemesanan.

Berdasarkan penelitian dari Riki Subagia (2020) dengan judul “Sistem Informasi Rental Mobil Berbasis Web” Perkembangan teknologi informasi telah bergerak dengan cepat. Hal ini menimbulkan perubahan, dimana pelanggan tidak lagi puas berinteraksi hanya secara fisik, namun pelanggan menginginkan suatu nilai lebih yang bisa diperoleh dari perusahaan, maka Sisrem Informasi Rental

Mobil Berbasis Web Menggunakan php perlu untuk dibuat. Sehingga customer tidak perlu repot-repot datang ke tempat rental mobil hanya untuk melihat jenis-jenis mobil yang disewakan. Sistem informasi rental mobil online dapat dilaksanakan di komputer rumah, kantor, warnet, dan dimanapun asalkan ada media yang terhubung dengan internet. Maka dari itu, dirancanglah sistem aplikasi rental mobil berbasis web untuk mempermudah calon penyewa mobil dalam mencari mobil yang ada di sekitar tempatnya dan bisa diakses lewat internet tanpa harus ke lokasi tempat penyewaan terlebih dahulu.

Berdasarkan penelitian dari Intan Septavia (2018) dengan judul “Sistem Informasi Penyewaan Mobil Berbasis Web Di Jasa Karunia Tour And Travel” Jasa Karunia Tour and Travel adalah perusahaan yang bergerak dalam bidang jasa rental mobil. Dalam rangka memberikan kenyamanan kepada para pelanggannya pada saat melakukan transaksi maka Jasa Karunia Tour and Travel perlu aplikasi rental mobil. Tujuannya adalah untuk perbaikan sistem aplikasi financial sehingga memberi kemudahan kepada para pelanggan pada saat melakukan aktivitas bertransaksi. Metodologi yang digunakan dalam pembuatan Tugas Akhir ini adalah menggunakan metodologi Waterfall (Sommerville, 2003) yang terdiri dari tahapan Analisa Kebutuhan.

Sedangkan penelitian yang dilakukan penulis untuk merancang sistem aplikasi berbasis android dalam rental mobil, dimana pelanggan melakukan rental dengan menggunakan perangkat android dan pihak perusahaan menerima rental dari pelanggan dengan menggunakan aplikasi berbasis website dan mempermudah pelanggan dalam melakukan penyewaan dan pembayaran rental mobil sehingga

memperoleh hasil penyewaan dan pembayaran rental mobil berdasarkan permintaan dari pelanggan, dan sistem yang dirancang oleh penulis mempermudah pekerjaan perusahaan yaitu dengan memanfaatkan aplikasi *Android* proses penyewaan dan pembayaran rental mobil bisa diterima dengan cepat.

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

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

Pengertian aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi *user* dengan menggunakan kemampuan komputer. (Fergiawan Listianto, dkk : 2017)

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. (Ari Setiaji, dkk : 2020).

### **II.2.2. Android**

*Android* adalah suatu sistem operasi yang berbasis *Linux* dengan perangkat *Mobile device*, *Android* menyediakan platform terbuka bagi para pengembang

untuk menciptakan aplikasi-aplikasi baru untuk membantu kebutuhan manusia. Awal terdirinya android dimulai dari *Google Inc* yang membeli *Android inc* pendatang bat dari perangkat lunak untuk ponsel/*smartphone*. Semakin banyak *smartphone* dan *PC Tablet* menggunakan sistem operasi dengan versi yang berbdeda. Semakin tinggi versinya maka fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi *Android* adalah *HTC Dream* yang dirilis pada tanggal 22 Oktober 2008. Perkembangan sistem *Android* berkembang sesuai versi keluarnya adapun versi *Android* dimulai dari (Muhamad Zaelani : 2017)

**Tabel. II.1 Versi Android**

| Versi   | Nama                            | Rillis                                                            | Catatan                                                                   |
|---------|---------------------------------|-------------------------------------------------------------------|---------------------------------------------------------------------------|
| 1.0     | <i>Android 1.0</i>              | 23 September 2008                                                 | <i>Android</i> pertama hanya untuk <i>smartphone</i>                      |
| 1.1     | <i>Android 1.1</i>              | 9 Februari 2008                                                   |                                                                           |
| 1.5     | <i>Cupcake</i>                  | 30 April 2009                                                     | Mulai pakai kode nama                                                     |
| 1.6     | <i>Donut</i>                    | 15 September 2009                                                 |                                                                           |
| 2.0-2.1 | <i>Éclair</i>                   | 26 Oktober 2009 (2.0)<br>12 Januari 2010 (2.1)                    |                                                                           |
| 2.2     | <i>Froyo</i>                    | 20 Mei 2010                                                       |                                                                           |
| 2.3     | <i>Gingerbread</i>              | 6 Desember 2010                                                   | Digunakan pada <i>smartphone</i> jenis lama                               |
| 3.0-3.2 | <i>Honeycomb</i>                | 22 Februari 2011 (3.0)<br>10 Mei 2011 (3.1)<br>15 Juli 2011 (3.2) | Hanya untuk tablet                                                        |
| 4.0     | <i>ICS (Ice Cream Sandwich)</i> | 19 Oktober 2011                                                   | <i>Smartphone</i> dan tablet                                              |
| 4.1-4.3 | <i>Jelly Bean</i>               | 9 Juli 2012 (4.1)<br>13 November 2012 (4.2)<br>24 Juli 2013 (4.3) | <i>Update</i> untuk memperbaiki dan menambah <i>fitur</i> pada <i>ICS</i> |

|     |                    |                                              |                                                                                                                                 |
|-----|--------------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| 4.4 | <i>Kit kat</i>     | 3 September 2013                             |                                                                                                                                 |
| 5.0 | <i>Lollipop</i>    | 12 november 2014 (5.0)<br>9 Maret 2018 (5.1) |                                                                                                                                 |
| 6.0 | <i>Marshmallow</i> | 5 oktober 015                                | Terdapat <i>daze mode</i> , <i>Do Not Disturb mode</i> , mendukung <i>USB tipe C</i> , mendukung pembacaan <i>fingerprint</i> . |
| 7.0 | <i>Nougat</i>      | 22 Agustus 2017                              | Multitasking                                                                                                                    |

(Sumber: Muhamad Zaelani : 2017)

Adapun penjelasan dari versi as android diatas adalah sebagai berikut :

#### 1. *Android 1.0: Astro (Alpha)*

Kedua versi awal *Android* ini mungkin agak asing kamu dengarkan. Pasalnya versi *Android 1.0 Astro (Alpha)* dan *Android 1.1* 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 1.1 : Bender (Beta)*

*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*.

#### 3. *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.

#### 4. *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.

#### 5. *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*.

#### 6. *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.

#### 7. *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.

#### 8. *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 perilisan *Android 3.0 & 3.2 Honeycomb* pada 10 Mei 2011 ini untuk mendukung *Samsung* yang mulai merilis perangkat *tablet*, *Samsung Galaxy Tab Series* untuk menyaingi *Apple iPad*

#### 9. *Ice Cream Sandwich (ICS)*

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.

#### 10. *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.

#### 11. *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.

#### 12. *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.

### 13. *Marshmallow*

*Android 6.0 Marshmallow* menjadi suksesor dari versi *Android* sebelumnya. Sistem operasi ini sendiri pertama kali diperkenalkan pada Mei 2018 dan mulai dirilis pada Oktober 2018 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*.

### 14. *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 2017 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. (Muhamad Zaelani ; 2017)

### II.2.3. *Android Studio*

*Android Studio* adalah *Integrated Development Enviroment* (IDE) untuk sistem operasi *Android* yang dibangun berdasarkan perangkat lunak IntelliJ IDEA. Selain editor kode dan alat pengembang yang didesain khusus untuk pengembangan *android*, *Android studio* merupakan pengganti dari *Eclipse Android Development Tools* (ADT) yang sebelumnya merupakan IDE utama

untuk pengembangan aplikasi *android*. *Android studio* menyediakan banyak fitur untuk meningkatkan produktivitas dalam pengembangan aplikasi berbasis android seperti tersedianya sistem pembangunan berbasis *gradle* yang *fleksibel*, emulator yang memiliki banyak fitur, integrasi dengan GitHub, dukungan bawaan untuk *Google Cloud Platform* sehingga mempermudah pengembang aplikasi untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine*. (Roberto Kaban : 2019)

#### II.2.4 Basis Data (*Database*)

Basis data atau *database* merupakan koleksi dari data-data yang terorganisir dengan rapi sehingga data dapat dengan mudah disimpan dan dimanipulasi. Kita dapat menjumpai pemanfaatan *database* dalam kehidupan sehari-hari, seperti penggunaan mesin ATM, sistem akademi kuniversitas/sekolah, sistem informasi penjualan. Salah satu tujuan dari *database* adalah memberikan pengguna suatu pandangan abstrak dari data, yaitu sistem menyembunyikan rincian bagaimana data disimpan dan dipelihara. Sisem *database* harus dibuat semudah mungkin untuk dimengerti karena kebanyakan pengguna sistem *database* adalah orang-orang yang kurang terlatih di bidang teknologi. (Ni Kadek Ceryna, dkk : 2018)

#### II.2.5. MySQL

*MySQL* adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar

untuk mengakses databasenya. *MySQL* termasuk jenis *RDBMS (Relational Database Management System)*. Pada *MySQL*, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. (Priyo Sutopo, dkk : 2017)

*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 : 2017)

## 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 : 2017).

PHP adalah salah satu *server side* yang dirancang khusus untuk aplikasi web. PHP disisipkan diantara bahasa HTML dan karena bahasa *server side*, maka bahasa PHP akan dieksekusi di *server*, sehingga yang dikirimkan ke browser adalah hasil jadi dalam bentuk HTML, dan kode PHP tidak akan terlihat. PHP termasuk *Open Source Product*. Jadi, dapat diubah *source code* dan mendistribusikannya secara bebas (Priyo Sutopo, dkk : 2017)

PHP (singkatan rekursif: *PHP Hypertext Preprocessor*) merupakan script yang dapat disisipkan ke dalam HTML. PHP dipakai untuk membuat sistem berbasis web yang dinamis. PHP bahasa pemrograman yang *server-side* sehingga program tersebut akan dijalankan atau diproses oleh *server*. Hingga saat ini PHP sudah rilis versi 7.4.1 pada 18 desember 2019. (Roberto kaban : 2019)

## **II.2.8. 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)

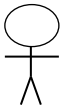
#### **II.2.8.1. Use Case Diagram**

*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.2 di bawah ini :

**Tabel II.2. Simbol Use Case**

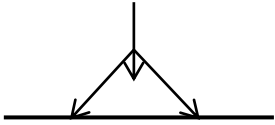
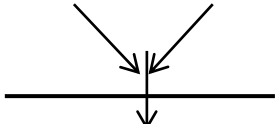
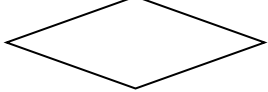
| Gambar                                                                              | Keterangan      | Deskripsi                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <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 : 2017)

### II.2.8.2. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat di lihat pada tabel II.3 dibawah ini:

**Tabel II.3. 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 pararel 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.                                                        |

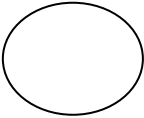
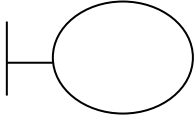
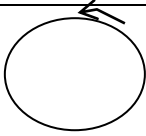
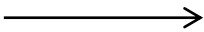
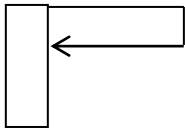
(Sumber : Ade Hendini : 2017)

### II.2.8.3. 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.4 di bawah ini :

**Tabel II.4. 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 : 2017)

#### II.2.8.4. 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, 2017 : 111).

**Tabel II.5. 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 : 2017)**