

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Profil PT Remaja Rosdakarya**

PT Remaja Rosdakarya berdiri sejak 15 Mei 1961 di Bandung. Awalnya perusahaan ini bernama CV Remaja Karya dan memiliki anak perusahaan dengan nama CV Rosda. Kemudian pada tahun 1970, CV Remaja Karya meluaskan bidang usahanya di bidang percetakan. Lima tahun kemudian, didirikan pula PT Rosda Jayaputra yang berlokasi di Jakarta. Pada tahun 1990, CV Remaja Karya digabungkan dengan CV Rosda dengan nama PT Remaja Rosdakarya, dan tepat pada 1 Januari 1998, dilakukan *merger* PT Rosda Jayaputra ke dalam PT Remaja Rosdakarya yang memiliki dua unit bisnis strategis, yaitu Penerbit dan Percetakan berskala besar. Dari tahun ke tahun PT Remaja Rosdakarya terus berbenah diri, maka pada tahun 2011, PT Remaja Rosdakarya mulai melangkahkan kaki ke arah penerapan sistem manajemen mutu, yaitu dengan menerapkan ISO 9001:2008.

Produk buku yang diterbitkan PT Remaja Rosdakarya pun semakin banyak dan beragam. Beberapa jenis buku yang telah diterbitkan, di antaranya adalah buku-buku perguruan tinggi (pendidikan, komunikasi, filsafat, sosiologi, psikologi, dan sebagainya), buku-buku agama Islam baik untuk perguruan tinggi maupun buku agama Islam populer, buku umum ilmiah, populer ataupun bergambar, buku-buku anak/perpustakaan yang terdiri atas buku-buku ini ditujukan untuk anak usia pra-sekolah dan usia sekolah dengan maksud untuk menambah pengetahuan, mengembangkan kreativitas dan imajinasi anak. Buku-buku tersebut disusun secara berseri, seperti seri dunia dalam bahaya, seri biologi, seri ekologi, seri binatang, seri lingkungan, seri ensiklopedi, seri Nabi, seri Walisanga, dan lain-lain.

Untuk mengikuti perkembangan zaman, sekarang ini perusahaan menerbitkan buku dalam bentuk digital di bawah imprint eRosda, sedangkan buku berbahasa Inggris/asing dan berskala internasional di bawah imprint Rosda International.

Saat ini, perusahaan juga telah dan sedang menjalin kerja sama dengan berbagai penerbitan luar negeri, instansi pemerintah, maupun lembaga pendidikan.

Kerjasama penerbitan yang telah dan sedang perusahaan jalankan, yaitu dengan Pusat Bahasa (dengan menerbitkan buku-buku sastra dan kebahasaan) dan dengan perguruan tinggi, seperti Universitas Pendidikan Indonesia Bandung, Universitas Negeri Malang, Universitas Negeri Jakarta, Universitas Negeri Surabaya, dan Universitas Padjadjaran (dengan menerbitkan buku-buku untuk perguruan tinggi).

Untuk mempermudah pemasaran dan akses pendistribusian, perusahaan memiliki cabang penjualan yang biasa disebut Bagian Penjualan Penerbit (BPP) yang tersebar di beberapa ibukota provinsi, yakni Jakarta, Bandung, Yogyakarta, dan Surabaya.

Di samping penjualan langsung ataupun melalui toko buku ternama di Indonesia, perusahaan juga melayani penjualan atau pemesanan melalui website [www.rosda.id](http://www.rosda.id) Adapun di luar negeri perusahaan memiliki perwakilan di Malaysia.

Adapun Divisi Percetakan perusahaan ini berkedudukan di Jl. Raya Cimahi Padalarang No. 93, Bandung. Pada divisi ini perusahaan melayani proses penggandaan buku dari penerbit sendiri maupun penerbit lain. Selain itu, melayani pencetakan katalog, kalender, poster, dan lain-lain.

### **2.1.1 Logo PT Remaja Rosdakarya**

Logo dari PT Remaja Rosdakarya dapat dilihat pada Gambar 2.1.



**Gambar 2. 1 Logo PT Remaja Rosdakarya**

## 2.1.2 Visi dan Misi

PT Remaja Rosdakarya mempunyai Visi dan Misi sebagai berikut:

### 2.1.2.1 Visi PT Remaja Rosdakarya

Penyedia konten, produk cetak, dan digital berskala internasional yang terpercaya dengan teknologi terbaik.

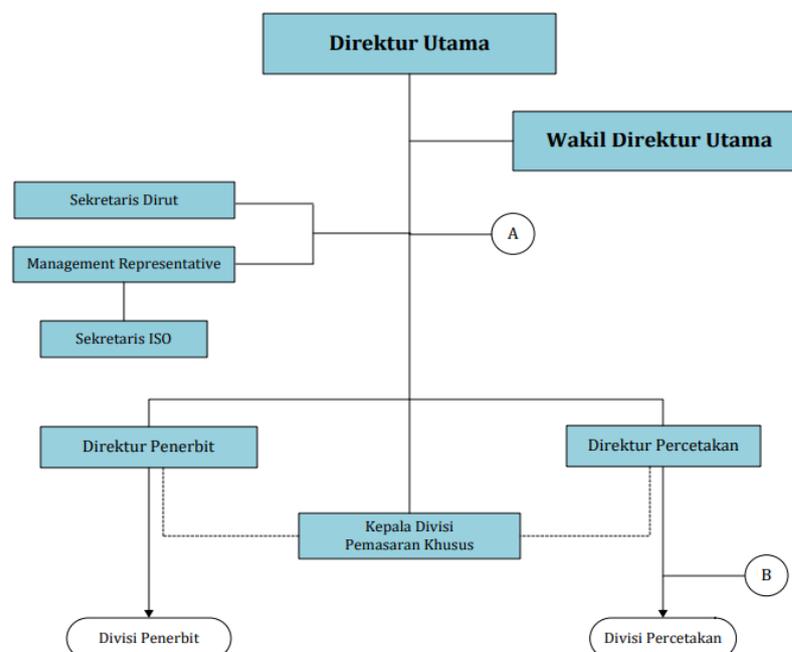
### 2.1.2.2 Misi PT Remaja Rosdakarya

Misi PT Remaja Rosdakarya meliputi:

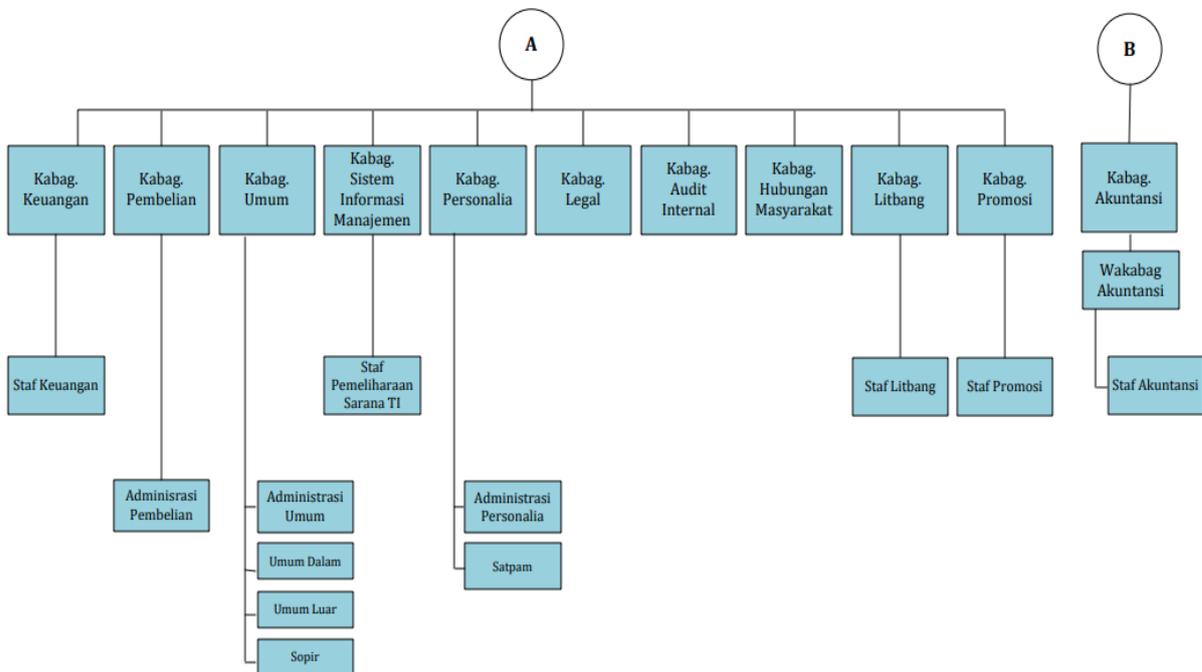
- Menghasilkan konten, produk cetak, dan digital yang berkualitas untuk pasar nasional dan internasional.
- Menghasilkan tingkat keuntungan yang dapat menjamin keberlanjutan usaha.
- Mengembangkan kompetensi di bidang konten, produk cetak, dan digital yang didukung budaya kerja yang efektif dan efisien.
- Berkontribusi terhadap pembangunan karakter dan kecerdasan bangsa.

## 2.1.3 Struktur Organisasi

Adapun struktur organisasi di PT Remaja Rosdakarya yang dapat dilihat pada Gambar 2.2 Struktur Organisasi dan Gambar 2.3 *Detail* Struktur Organisasi.



**Gambar 2.2 Struktur Organisasi**



**Gambar 2. 3 Detail Struktur Organisasi**

## 2.2 Landasan Teori

Landasan teori menjelaskan beberapa teori-teori dan penjelasan yang berkaitan dengan aplikasi yang akan dibangun. Landasan teori yang digunakan dalam aplikasi *sales tracking* berbasis android meliputi pengertian *monitoring*, *sales*, *tracking*, aplikasi, teknologi informasi, *google maps*, *MySQL*, *Android*, *Java*, *Emulator*, *Framework*, *MVC*, *Laravel*, *Website*, *JSON*, dan *UML*.

### 2.2.1 Monitoring

Salah satu proses *controlling* adalah melakukan *monitoring*. *Monitoring* sendiri merupakan proses mengumpulkan serta menganalisis informasi hasil dari penerapan program salah satunya memeriksa secara *reguler* untuk melihat kegiatan/program tersebut berjalan sesuai rencana atau tidak, sehingga masalah yang muncul dapat diselesaikan. *Monitoring* dapat digambarkan dalam bentuk grafik yang mencakup penjualan per *salesman* per cabang yang berlangsung setiap bulan. Adapun monitoring yang akan dilakukan PT Remaja Rosdakarya nantinya melibatkan *owner* dan *salesman* itu sendiri.

### **2.2.2 Sales**

*Sales* merupakan kegiatan atau aktivitas yang bertugas untuk menjual produk, menawarkan produk, mendistribusikan, dan bertujuan untuk meningkatkan penjualan produk. Tugas dari seorang *sales* adalah mencari pelanggan, serta mengajak pelanggan supaya untuk meminati produk yang dibawa atau ditawarkan *sales*, kemudian mendengarkan komentar terhadap produk barang dan jasa yang telah ditawarkan. *Sales* harus dapat menumbuhkan dan membesarkan motivasi kustomer untuk membeli produk yang ditawarkan tersebut [5].

### **2.2.3 Tracking**

*Tracking* ialah suatu proses untuk memantau, mengawasi lokasi keberadaan dari objek tertentu, mulai dari pergerakannya, jalurnya. Pergerakan yang dimaksud adalah perpindahan ketika posisi objek dari suatu koordinat ke koordinat yang lain. Inilah yang disebut dengan *tracking* yakni merekam semua data perpindahan yang telah dilakukan objek tersebut [6].

### **2.2.4 Aplikasi**

Aplikasi merupakan program yang berisikan perintah-perintah untuk melaksanakan pengolahan data, aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal. Aplikasi juga adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word dan Microsoft Excel.

### **2.2.5 Teknologi Informasi**

Teknologi Informasi adalah kajian, desain, pengembangan implementasi, dukungan, atau manajemen sistem informasi yang berbasis komputer, khususnya aplikasi perangkat lunak dan perangkat keras. Lebih lanjut, teknologi informasi berkaitan dengan penggunaan komputer dan perangkat lunak untuk mengubah, menyimpan, melindungi, memproses, memindah, dan menggunakan informasi secara aman.

Saat ini istilah teknologi informasi semakin berkembang, meliputi banyak aspek yang berkaitan dengan komputasi dan teknologi. Kombinasi antara komputer dan komunikasi, di sebut juga sebagai teknologi informasi. Oleh karena itu,

teknologi informasi dalam pengertian umum menggambarkan teknologi yang menghasilkan, memanipulasi, menyimpan, mengomunikasikan, dan/atau menyebarkan (diseminasi) informasi. Kesimpulannya penggunaan komputer dalam hubungannya untuk menangani informasi saat ini, disebut sebagai teknologi informasi. Dalam perkembangannya, teknologi informasi kemudian dikaitkan dengan penggunaan internet sebagai media pengiriman dan penerimaan informasi [7].

### 2.2.6 Google Maps API

Google API yaitu bagian dari *Framework* Google. Google menyediakan berbagai API (*Application Programming Interface*) yang sangat berguna bagi pengembang aplikasi. API diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi dengan fungsi-fungsi yang dikerjakan. Adapun beberapa pemanfaatan Google API dalam aplikasi yang dibuat yaitu Google *Maps* API, dan *Firebase* API.

Google *maps* merupakan sebuah layanan peta dunia *virtual* yang disediakan oleh Google Inc. Google *maps* dapat dilekatkan sebagai elemen dalam tata letak antarmuka pengguna yang di rancang oleh pemrogram. Untuk menghubungkan aplikasi perangkat lunak dengan google *maps* diperlukan sebuah kunci yang diistilahkan sebagai *API Key*. Fitur Google *Maps* ini dapat digunakan secara embeded dalam beberapa bahasa pemrograman termasuk Android melalui sebuah jembatan aplikasi yang disebut *Application Programming Interface* (API). Google menyediakan API untuk pengembang mengintegrasikan aplikasinya dengan fitur Google *Maps*. Dengan menggunakan Google *Maps* API, peta google akan dapat diintegrasikan dengan berbagai platform pemrograman, sehingga peta google dapat tampil di halaman aplikasi yang dibuat.

Cara menggunakan Google *Maps* API, tahap pertama dapatkan kunci API terlebih dahulu.

*Website* : <https://developers.google.com/maps/documentation/javascript/tutorial>

Android : <https://developers.google.com/maps/documentation/android-api/start>

Jika belum membuat projek API, maka silahkan kunjungi <https://console.developers.google.com/> untuk membuat suatu project.

Tahap kedua membuat program Google *Maps* baik dalam website maupun Android, Contoh google *maps* yang menunjukkan daerah Cirebon dengan *latitude* - 6.7063, *longitude* 108.557 :

### Google MAPS API untuk website :

#### Keterangan :

```
<script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"
  async defer></script>
```

Pada sebuah *source code* terdapat kata *YOUR\_API\_KEY*, dimana kata tersebut diganti dengan kunci API yang telah didapatkan pada tahap pertama.

### Google MAPS API untuk Android :

```
<string name="google_maps_key" templateMergeStrategy="preserve"
translatable="false">YOUR_KEY_HERE</string>
</resources>
```

Pada folder *app/res/values/google\_maps\_api.xml* ubah kata *YOUR\_KEY\_HERE* dengan kunci API yang telah didapatkan pada tahap pertama.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple Map</title>
    <meta name="viewport" content="initial-scale=1.0">
    <meta charset="utf-8">
    <style>
      #map {
        height: 100%;
      }
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
    </style>
  </head>
  <body>
    <div id="map"></div>
    <script>
      var map;
      function initMap() {
        map = new google.maps.Map(document.getElementById('map'), {
          center: {lat: -6.7063, lng: 108.557},
          zoom: 8
        });
      }
    </script>
    <script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback
```

```

=initMap"
  async defer></script>
</body>
</html>

```

### 2.2.7 Location Based Services (LBS)

*Location Based Services* adalah layanan berbasis lokasi atau istilah umum yang sering digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi *Global Positioning Service* (GPS) dan *cell-based location* dari Google.

Selain itu, LBS tersebut terdiri dari beberapa komponen di antaranya *mobile devices*, *communication network*, *position component*, dan *service and content provider*. *Mobile devices* merupakan komponen yang sangat penting. Piranti *mobile* tersebut diantaranya adalah *smartphone*, PDA, dan lainnya yang dapat berfungsi sebagai alat navigasi atau seperti halnya alat navigasi berbasis GPS.

Komponen *communication network* ini berupa jaringan telekomunikasi bergerak yang memindahkan data pengguna dari perangkat ke penyedia layanan. *Position component* yang dimaksud adalah posisi pengguna harus ditentukan. Posisi ini dapat didapatkan dengan jaringan telekomunikasi atau dengan GPS. Sedangkan *service and content provider* adalah penyedia layanan yang menyediakan layanan berbeda ke pengguna seperti pencarian rute, kalkulasi posisi, dan lainnya [8].

### 2.2.8 Geofencing

*Geofencing* merupakan perangkat lunak yang digunakan bersamaan dengan *Global Positioning System* (GPS) dalam menentukan batas-batas geografis atau parameter *virtual* dari suatu peta. Program yang menggunakan *geofencing* dapat mengatur suatu *triggers* yang dapat memberikan informasi atau notifikasi apabila suatu target tertentu masuk atau keluar dari suatu batasan yang telah ditetapkan sebelumnya. Beberapa teknik dari *geofencing* adalah *Geofence Area*, *Proximity with Point of Interest*, *Route adherence*, dan *Route and schedule adherence*. Secara garis besar, koordinat geografis digunakan untuk mengetahui posisi target dan juga untuk membuat suatu batasan daerah tertentu (*mapping*) sebagai pagar *virtual* (*geofence*) suatu daerah. Sistem akan menentukan posisi target yang dilacak berada

di luar atau di dalam wilayah *geofencing*. Teknologi ini juga dapat memungkinkan untuk pendeteksian kedekatan antara posisi target dengan area *geofencing* tertentu.

Penggunaan *geofencing* dalam mengawasi pengiriman kendaraan akan dilakukan dengan cara menggambarkan area-area *virtual* pada suatu lokasi-lokasi tertentu, seperti gerbang tol, jalan ataupun suatu kota, sehingga semua jalur penting yang dilewati dapat di monitor [9].

### 2.2.9 Global Positioning System

*Global Positioning System* merupakan teknologi penentu lokasi yang banyak digunakan saat ini. GPS adalah suatu sistem radio navigasi penentuan lokasi menggunakan satelit. Dengan bantuan satelit, akan diperoleh posisi yang akurat dan cepat dengan koordinat 3 dimensi (x,y,z) ditambah dengan informasi waktu dan kecepatan bergerak. Posisi unit GPS akan ditentukan berdasarkan titik-titik koordinat latitude dan longitude yang diperolehnya dari nilai derajat dari suatu titik yang diukur.

Penggunaan GPS akan membantu menemukan letak koordinat *latitude* dan *longitude* dari target. GPS yang digunakan akan memanfaatkan GPS yang terdapat pada *smartphone* yang akan diaktifkan pada saat target dijalan. Perkembangan penelitian lebih lanjut dapat menggunakan GPS Tracker khusus yang dapat dipasang ke kendaraan. Koordinat dari GPS nantinya akan dikirimkan ke server untuk membantu mendeteksi *geofencing*.

### 2.2.10 MySQL

*My Structur Query Language* (MySQL) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, *Postrage SQL*, dan lain-lain. MySQL merupakan DBMS yang multithread, multi-user yang bersifat gratis di bawah lisensi GNU *General Public Licence* (GPL). Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. MySQL dimiliki dan disponsori oleh sebuah perusahaan Swedia, yaitu MySQL AB. MySQL AB memegang hak cipta kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah David Axmark, Allan Larson, dan

Michael Monty Widenius. Seperti yang telah disebutkan sebelumnya, MySQL bersifat gratis atau *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung/*support* dengan *database* MySQL sehingga apabila kita mempelajarinya dengan sungguh-sungguh kita dapat mengaplikasikan PHP & MySQL dalam membuat aplikasi website maupun dalam membuat website [10].

### 2.2.11 Android

Android pertama dikembangkan oleh Android Inc yang kemudian telah rilis pada tahun 2007 yang bersamaan dengan didirikannya *Open Ponsel* Android. Pada pengertiannya android merupakan suatu sistem operasi yang sifatnya *open source* berbasis linux untuk merancang perangkat seluler *touchscreen* seperti telepon cerdas dan komputer tablet [11]. Statusnya bukan hanya menjadi suatu sistem operasi di *smarthphone*, sekarang ini android adalah *platform* yang sangat lengkap sekali terutama dari secara sistem operasinya, aplikasi dan *tools* pengembangannya, *market* aplikasi android dan dukungan yang luar biasa sangat tinggi dari komunitas *open source* di dunia, oleh karena itu akan lebih berkembang pesat secara teknologinya atau jumlah *device* yang ada di dunia [12].



**Gambar 2.4 Android**

Android 5.0 pertama kali diperkenalkan di bawah *codename* "Android L" pada 25 Juni 2014 selama presentasi *keynote* pada konferensi pengembang Google I / O. Di samping Lollipop, presentasi difokuskan pada sejumlah *platform* Android yang berorientasi dan teknologi baru, termasuk Android TV, pada *platform* Android *Auto*, dpt dipakai pada platform komputasi Android Wear, dan *platform* pelacakan kesehatan Google Fit.

Android Lollipop menawarkan *user interface* yang lebih menarik dan modern serta kinerja perangkatnya pun sudah optimal. Maka dari itu sistem android digunakan untuk membangun sebuah aplikasi sales tracking. Android yang digunakan adalah Android versi 5.0 (Lollipop). Karena dengan kelebihan-kelebihan pada android Lollipop ini dapat memaksimalkan dalam pembuatan sistem aplikasi *sales tracking*.

#### **2.2.11.1 Android SDK (*Software Development Kit*)**

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*. Saat ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk desktop modern), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista. Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau yang lebih baru. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan plugin *Android Development Tools (ADT)*, dengan ini pengembang dapat menggunakan teks editor untuk mengedit *file* Java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan *debug* aplikasi Android dan pengendalian perangkat Android (misalnya, *reboot*, menginstal paket perangkat lunak dengan jarak jauh).

#### **2.2.11.2 ADT (*Android Development Tools*)**

*Android Development Tools (ADT)* adalah plugin untuk Eclipse yang didesain untuk pengembangan aplikasi Android. ADT memungkinkan Eclipse untuk digunakan dalam membuat aplikasi Android baru, membuat *User Interface*, menambahkan komponen berdasarkan *framework* API Android, *debug* aplikasi, dan pemaketan aplikasi Android.

#### **2.2.11.3 Android Studio**

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment (IDE)* untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan *editor* kode IntelliJ dan alat

pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

- a. Sistem versi berbasis *Gradle* yang *fleksible*.
- b. Emulator yang cepat dan kaya fitur.
- c. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
- d. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
- e. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
- f. Alat pengujian dan kerangka kerja yang ekstensif.
- g. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
- h. Dukungan C++ dan NDK.
- i. Dukungan bawaan untuk Google *Cloud Platform*, mempermudah pengintegrasian Google *Cloud Messaging* dan *App Engine*.

### **2.2.12 Java Development Kit (JDK)**

Salah satu requirement dari Android Studio adalah JDK atau *Java Development Kit*. Maka dari itu, langkah pertama yang harus dilakukan sebelum menginstal Android Studio adalah menginstal JDK. JDK dapat diunduh di <http://www.oracle.com/technetwork/java/javase/download/jdk8-downloads-2133151.html>. [13]

### **2.2.13 Emulator**

Untuk mencoba program yang dibuat di Android Studio, diperlukan *emulator* Android. Atau dapat pula langsung dijalankan menggunakan *smartphone* Android. Perlu diketahui bahwa menjalankan program Android melalui *emulator* membutuhkan RAM komputer lebih dibandingkan menggunakan *smartphone*. Sebaiknya komputer memiliki RAM 8 GB atau lebih jika ingin menggunakan *emulator*. Ada Beberapa software emulator Android yang dapat digunakan misalnya *Android Virtual Device* (AVD) yang disediakan Android SDK atau

*emulator* yang disediakan pihak ketiga seperti Genymotion [13]. Tampilan *emulator* dapat dilihat pada Gambar 2.5 Tampilan Emulator.



Sumber : [https://developer.android.com/images/tools/studio/studio-feature-emulator\\_2x.jpg?hl=id](https://developer.android.com/images/tools/studio/studio-feature-emulator_2x.jpg?hl=id)

**Gambar 2.5 Tampilan Emulator**

#### 2.2.14 PHP

Pengertian PHP menurut beberapa para ahli. Diantaranya, Menurut Sibero (2012:49), “PHP (*Personal Home Page*) adalah pemrograman (*interpreter*) adalah proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”.

Menurut Kustiyahningsih (2011:114), “PHP (atau resminya PHP: *Hypertext Preprocessor*) adalah skrip bersifat *server-side* yang ditambahkan ke dalam HTML. PHP sendiri merupakan singkatan dari *Personal Home Page Tools*. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke HTML sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server side* berarti pengerjaan kode program dilakukan di *server*, baru kemudian hasilnya dikirimkan ke browser”.

Berdasarkan pengertian PHP yang dipaparkan oleh para ahli, dapat ditarik kesimpulan bahwa PHP (PHP: *Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML.

Kelebihan PHP :

1. PHP adalah bahasa *multiplatform* yang artinya dapat berjalan di berbagai mesin dan sistem operasi (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah *system* lainnya.
  2. PHP bersifat *Open Source* yang berarti dapat digunakan oleh siapa saja secara gratis.
  3. *Web Server* yang mendukung PHP dapat ditemukan di mana-mana dari mulai Apache, IIS Lighttpd, nginx, hingga Xitamu dengan konfigurasi yang relatif mudah dan tidak berbelit-belit, bahkan banyak yang membuat dalam bentuk paket atau *package* (PHP, MySQL, dan Web Server).
  4. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis, komunitas dan *developer* yang siap membantu dalam pengembangan.
  5. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
  6. Banyak bertebaran Aplikasi dan Program PHP yang gratis dan siap pakai seperti WordPress, PrestaShop, dan lain-lain.
  7. Dapat mendukung banyak *database*, seperti MySQL, Oracle, MS-SQL.
- Kekurangan PHP :
1. PHP tidak mengenal *Package*.
  2. Jika tidak di-*encoding*, maka kode PHP dapat dibaca semua orang dan untuk meng-*encoding*-nya dibutuhkan *tool* dari Zend yang mahal sekali biayanya.
  3. PHP Memiliki kelemahan keamanan. Jad programmer harus jeli dan berhati-hati dalam melakukan pemrograman dan konfigurasi PHP [14].

### 2.2.15 Framework

Menurut Wikipedia (*Framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah kompleks. Jadi singkatnya *framework* adalah wadah/kerangka kerja dari sebuah website yang akan kita bangun. Dengan adanya kerangka kerja tersebut, maka akan lebih mempercepat waktu kita dalam pembuatan website sekaligus memudahkan melakukan perbaikan dari sebuah website. Sehingga, apabila di ibaratkan website kita itu adalah rumah, maka *framework* ini adalah kerangka dari rancangan rumah yang akan kita buat.

Keuntungan menggunakan *framework*:

1. Memberikan struktur yang baik dalam program yang kita buat
2. Dengan menggunakan *framework*, berarti harus menyesuaikan cara orang bekerja sesuai dengan gaya *framework* yang dipakai.
3. Dengan *framework*, pengerjaan *website* bisa lebih cepat
4. Meningkatkan keamanan
5. Lebih bebas dalam pengembangan jika dibandingkan CMS
6. Mempermudah dalam melakukan maintenance

Contoh Framework sebagai berikut : CodeIgnitier, Laravel, CakePHP, Yii Framework, Zend, Symfony [15].

### 2.2.16 MVC

*Model-View-Controller* atau disingkat MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan antara data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Dalam implementasi kebanyakan *framework* dalam aplikasi website adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web.

Terdapat 3 Jenis komponen yang membangun suatu pola MVC dalam suatu aplikasi, yaitu:

#### 1. *View*

Merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada user.

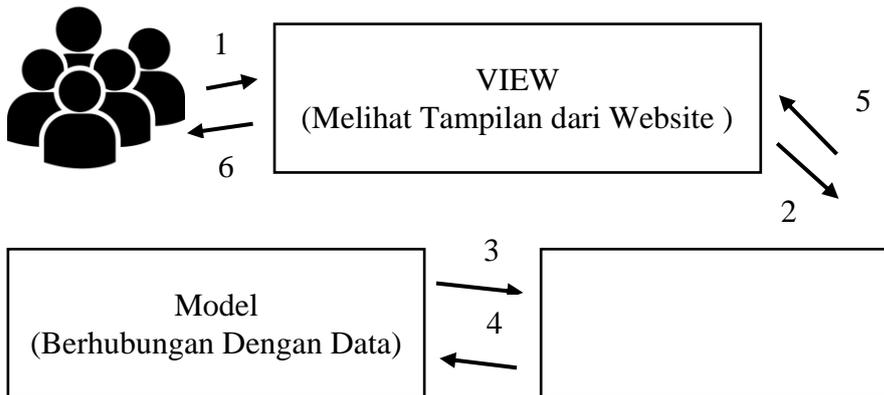
#### 2. *Model*

Biasanya berhubungan langsung dengan database untuk memanipulasi data, menangani validasi dari bagian *controller*, namun tidak berhubungan langsung dengan *view*.

### 3. *Controller*

Merupakan bagian yang mengatur hubungan antara bagian *Model* dan *View*. *Controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi. [15]

Alur MVC dapat dilihat pada Gambar 2.6 Alur MVC.



**Gambar 2.6 Alur MVC**

#### 2.2.17 Laravel Versi 5.6

Laravel 5.6 adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (*Model View Controller*). Laravel adalah pengembangan website berbasis MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintal-sintal yang ekspresif, jelas dan menghemat waktu.

Beberapa fitur unggulan laravel adalah sebagai berikut:

1. *Bundles* yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam aplikasi.
2. Eloquent ORM merupakan penerapan PHP lanjutan menyediakan metode *internal* dari pola *Active record* yang mengatasi malah pada hubungan objek *database*.
3. *Application Logic* merupakan bagian dari aplikasi, menggunakan *Controller* atau bagian *Route*.
4. *Route*

5. *Restful Controllers* memisahkan logika dalam melayani HTTP GET dan POST
6. *Class Auto Loading* menyediakan *loading* otomatis untuk *class* PHP
7. *View Composer* adalah kode *unit* logikal yang dapat dieksekusi ketika *view* sedang loading
8. *Migration* menyediakan sistem kontrol untuk skema *database*.
9. *Unit Testing* banyak tes untuk mendeteksi dan mencegah regresi.

Berikut beberapa kelebihan yang dimiliki Laravel:

1. *Coding* yang simpel
2. Tersedia *generator* yang canggih dan memudahkan, yaitu Artisan CLI.
3. Fitur *Schema Builder* untuk berbagai *database*.
4. Fitur *Migration & Seeding* untuk berbagai *database*.
5. Fitur *Query Builder* yang dapat memudahkan kita untuk melakukan manajemen data di *database*.
6. Eloquent ORM yang sangat memudahkan.
7. Fitur pembuatan *package* dan *bundle* yang memudahkan.
8. Dukungan Paket *Library* yang banyak

Berikut beberapa kekurangan yang dimiliki Laravel:

1. Ukuran file yang cukup besar
2. Memerlukan koneksi internet untuk instalasi dan mendownload
3. Laravel membutuhkan PHP versi 5.4 ke atas, sehingga tidak semua *hosting* yang *support* dengan Laravel.

### 2.2.18 Website

Website merupakan sebuah halaman berisi informasi yang dapat dilihat jika komputer Anda terkoneksi dengan internet. Dengan adanya website, semua orang di dunia bias mendapatkan dan mengelola informasi dengan berbagai sumber yang tersedia di internet. Website sendiri saat ini bias memuat berbagai macam media, mulai dari teks, gambar, suara, bahkan video.

Penemu website adalah Sir Timothy John ‘Tim’ Berners-lee, sedangkan website yang tersambung dengan jaringan pertama kali muncul pada tahun 1991. Pada awalnya, Tim menciptakan website dengan tujuan untuk mempermudah arus pertukaran dan memperbaharui informasi kepada sesama peneliti di CERN, tempat

dia bekerja. Pada tanggal 30 April 1993, CERN menginformasikan bahwa WWW dapat digunakan gratis oleh semua orang.

Website ditulis atau secara dinamik dikonversi menjadi *Hyper Text Markup Language* (HTML) dan diakses melalui sebuah program software yang biasa disebut *web browser*. Halaman web dapat dilihat atau diakses melalui jaringan *computer* dan internet, sedangkan perangkatnya bisa berupa *personal computer*, laptop, PDA, maupun *cell phone* [16].

### **2.2.19 Web Service**

*Web service* adalah aplikasi sekumpulan data (*database*), perangkat lunak (software) atau bagian dari perangkat lunak yang dapat diakses secara *remote* oleh berbagai piranti dengan sebuah perantara tertentu. Secara umum, *web service* dapat diidentifikasi dengan menggunakan URL seperti hanya web pada umumnya. Namun yang membedakan *web service* dengan web pada umumnya adalah interaksi yang diberikan oleh *web service*. Berbeda dengan URL web pada umumnya, URL *web service* hanya mengandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi. *Web service* dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah *database* ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di *platform* apa sebuah data itu dikonsumsi. *Web service* mampu menunjang interoperabilitas. Sehingga *web service* mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada. [17]

### **2.2.20 JSON**

JSON (*JavaScript Object Notation*) merupakan format yang ringan untuk memasukan data ke dalam sebuah variabel. Sangat mudah dimengerti dan diimplementasikan oleh manusia, dan mudah juga untuk komputer dalam melakukan parsingnya.

JSON merupakan bagian dari bahasa pemrograman JavaScript (Standard ECMA-262 3rd Edition – December 1999). JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C, termasuk C, C++, C#, Java, JavaScript, Perl,

Python, dan sebagainya. Kelebihan inilah yang membuat JSON menjadi sebuah bahasa data-interchange yang ideal.

JSON dibangun dalam dua struktur :

1. Beberapa pasangan dari nama/nilai. Dalam beberapa bahasa perograman biasa disebut dengan istilah *object*, *record*, *struct*, tabel *hash*, *key list* atau *associative array*.
2. Nilai-nilai yang terusun secara *ordered list*. Biasa disebut dengan array, vector, list atau daftar dalam bahasa pemrograman.

JSON adalah struktur data yang universal, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON (<http://www.myusro.info/2013/01/pengertian-json-javascript-object.html>) [18].

### 2.2.21 UML

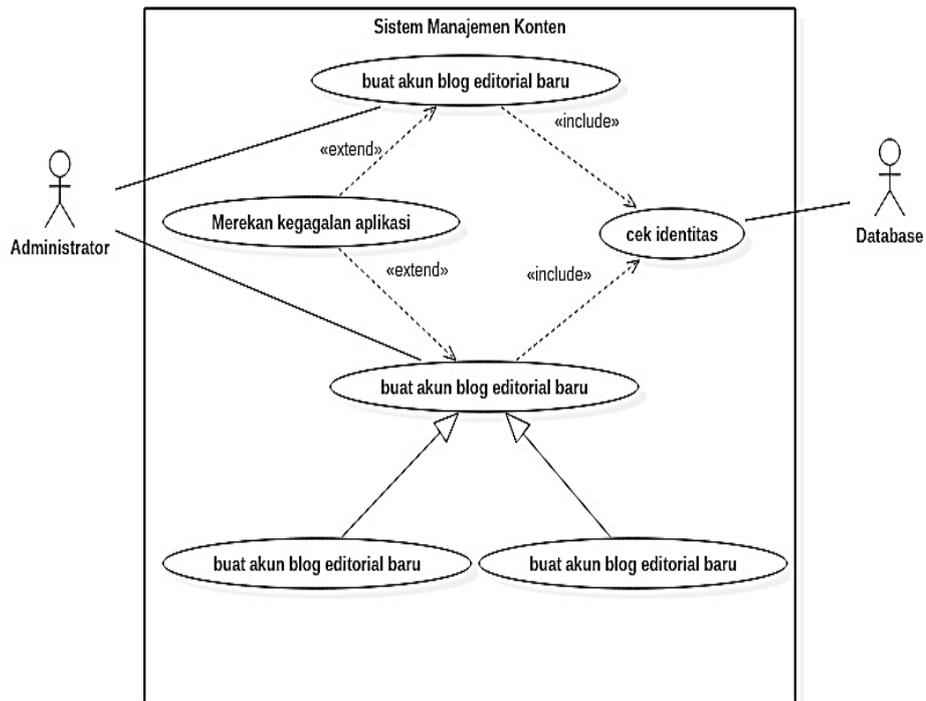
Analisis dan Desain berorientasi objek memiliki model yang berbeda dengan pengembangan perangkat lunak secara tradisional, yang berdasarkan fungsi dan prosedur. Lebih jelasnya, *Object-oriented* adalah cara untuk mengembangkan perangkat lunak dengan membangun objek yang dapat dengan mudah diganti, dirubah dan digunakan ulang. Dalam sebuah lingkungan *object-oriented*, perangkat lunak merupakan kumpulan dari objek diskrit yang datanya dienkapsulasi sebagaimana fungsi dari objek tersebut dalam dunia nyata. Setiap objek memiliki atribut dan *method*.

Dokumentasi UML menyediakan 10 macam *diagram* untuk membuat model aplikasi berorientasi objek yang 4 di antaranya adalah :

#### 1. *Use Case Diagram*

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Di dalam *use case diagram* ini sendiri lebih ditekankan kepada apa yang diperbuat sistem dan bagaimana sebuah sistem itu bekerja. Sebuah *use case* merepresentasikan sebuah interaksi antara *actor* dengan sistem. *Use case* merupakan bentuk dari sebuah pekerjaan tertentu, misalnya *login* ke dalam

sistem, *posting* dan sebagainya, sedangkan seorang *actor* adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [19]. Contoh *use case* dapat dilihat pada Gambar 2.7 Contoh *Use Case Diagram*.



**Gambar 2.7** Contoh *Use Case Diagram*

## 2. Skenario *Use Case*

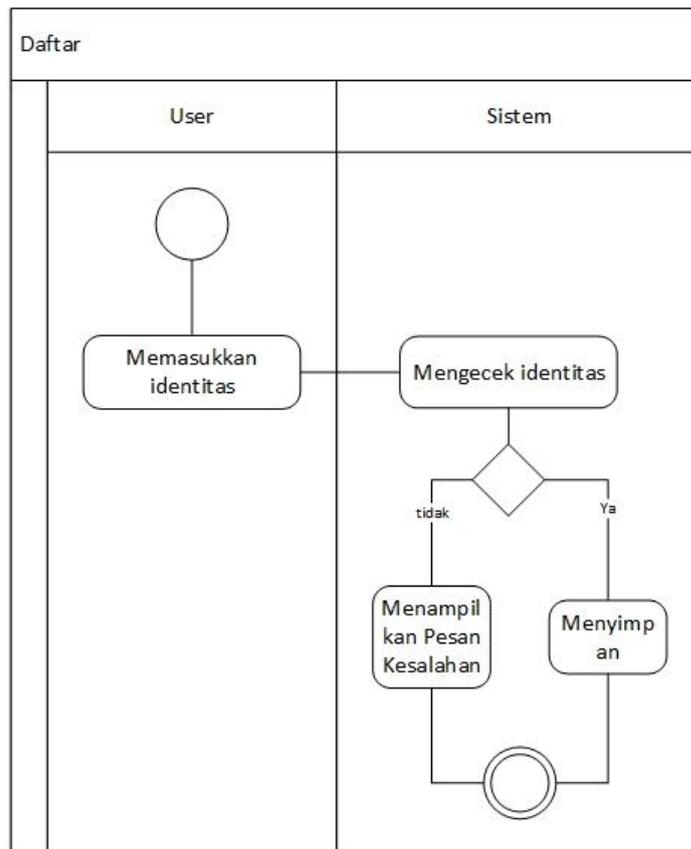
Sebuah *diagram* yang menunjukkan *use case* dan aktor mungkin menjadi titik awal yang bagus, tetapi tidak memberikan detail yang cukup untuk desainer sistem untuk benar-benar memahami persis bagaimana sistem dapat terpenuhi. Cara terbaik untuk mengungkapkan informasi penting ini adalah dalam bentuk penggunaan skenario *use case* berbasis teks per *use case*-nya [19]. Dasar format penulisan skenario *use case* dapat di lihat pada tabel 2.1.

Tabel 2.1 Penulisan Skenario *Use Case*

<b><i>Use Case Name</i></b>	Berisi nama <i>use case</i>	
<b><i>Related Requirement</i></b>	Berisi kode kebutuhan yang terkait dengan skenario	
<b><i>Goal in Context</i></b>	Tujuan yang ingin dicapai <i>use case</i>	
<b><i>Precondition</i></b>	Kondisi sistem ketika akan dilaksanakannya skenario <i>use case</i>	
<b><i>Successful End Condition</i></b>	Kondisi jika <i>use case</i> berhasil terpenuhi	
<b><i>Failed End Condition</i></b>	Kondisi jika <i>use case</i> gagal terpenuhi	
<b><i>Primary Actor</i></b>	Aktor utama yang berasosiasi dengan <i>use case</i>	
<b><i>Secondary Actor</i></b>	Aktor selain aktor utama yang berasosiasi dengan <i>use case</i>	
<b><i>Trigger</i></b>	Aktivitas yang dilakukan untuk mengawali <i>use case</i>	
<b><i>Included Cases</i></b>	<i>Use case</i> yang di-include	
<b><i>Main Flow</i></b>	<b><i>Step</i></b>	<b><i>Action</i></b>
	1	Langkah-langkah aksi dari aktivitas <i>use case</i>
	2	
	3	
<b><i>Extension</i></b>	<b><i>Step</i></b>	<b><i>Branching Action</i></b>
	2.1	Langkah-langkah aksi lain selain aktivitas utama dari <i>use case</i>

### 3. Diagram Aktivitas

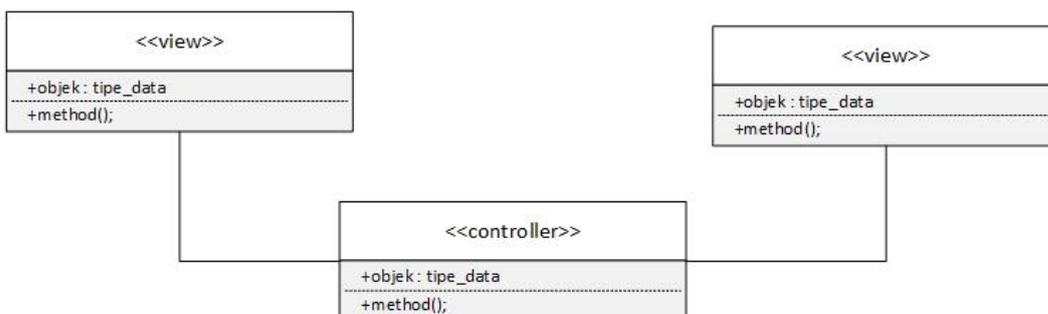
Diagram Aktivitas adalah sebuah tahapan yang lebih *focus* kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Di mana biasanya dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Diagram Aktivitas ini sendiri memiliki struktur yang mirip dengan *flowchart* atau data *flow diagram* pada perancangan terstruktur. *Diagram* Aktivitas dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram* [19]. Contoh *activity diagram* dapat dilihat pada Gambar 2.8 Contoh *Diagram* Aktivitas [20].



**Gambar 2.8** Contoh *Diagram* Aktivitas

#### 4. *Class Diagram*

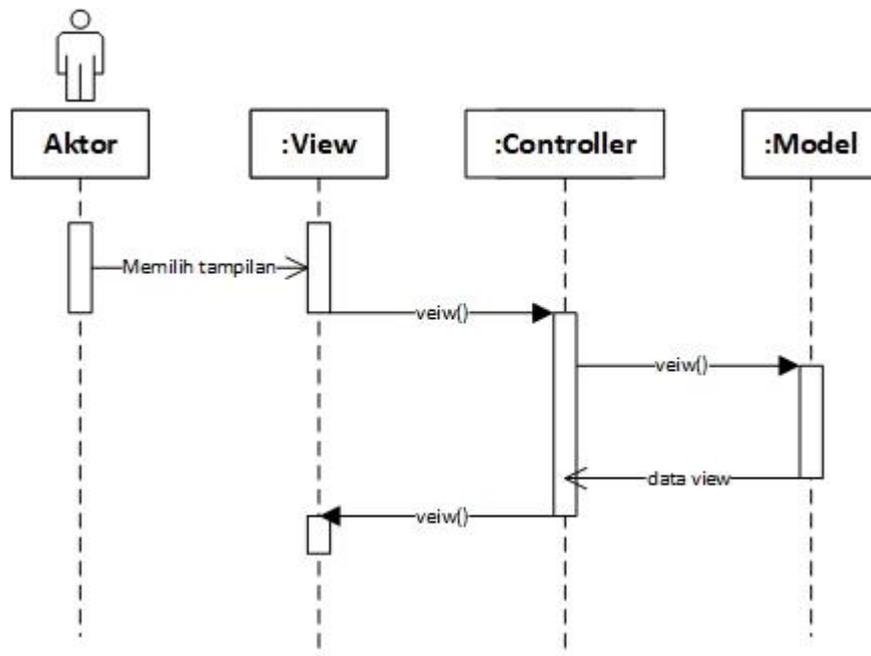
*Class diagram* adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket dan objek serta hubungan satu sama lain. *Class diagram* juga menjelaskan hubungan antar *class* secara keseluruhan di dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan [19]. Contoh *class diagram* dapat dilihat pada Gambar 2.9 Contoh *Class Diagram* [19].



**Gambar 2.9** Contoh *Class Diagram*

## 5. Sequence Diagram

*Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram jenis ini memberikan kejelasan sejumlah objek dan pesan-pesan yang diletakkan di antaranya di dalam sebuah *use case*. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, *message* yang digambarkan dengan garis putus dan waktu yang ditunjukkan dengan *progress vertical*. Manfaat dari *sequence diagram* adalah memberikan gambaran detail dari setiap *use case diagram* yang dibuat sebelumnya [19]. Contoh *sequence diagram* dapat dilihat pada *Gambar 2.10 Contoh Sequence Diagram* [19].



**Gambar 2.10 Contoh Sequence Diagram**

