BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Penelitian

Penelitian ini bekerja sama dengan *makeup artist* yang berada di kota bandung dan sekitarnya untuk kerterkaitan data yang akan digunakan dalam pembuatan aplikasi.

2.2 Makeup Artist

Makeup Artist adalah seniman professional yang menggunakan kulit, terutama wajah, sebagai *medium* karyanya dan produk *makeup* sebagai alatnya. *Makeup Artist* bisa memiliki fokus yang berbeda,misalnya riasan untuk pengantin dan acara formal lainnya, rias tradisional dan adat, *Face and Body painting,* atau *special effect* yang biasa digunakan oleh film-film fiksi.[1].

2.3 Aplikasi

Aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia. Selain pengertian tersebut ada banyak pengertian dari para ahli mengenai kata aplikasi adalah sebagai berikut:

1. Ali Zaki dan Smitdev Community

   Menurut Ali Zaki dan *Smitdev Community*, Aplikasi merupakan komponen yang bermanfaat sebagai media untuk menjalankan pengolahan data ataupun berbagai kegiatan lainnya seperti pembuatan ataupun pengolahan dokumen dan file.

2. Sri Widianti

   Menurut Sri Widianti, Aplikasi merupakan sebuah *software* (perangkat lunak) yang bertugas sebagai *front end* pada sebuah sistem yang dipakai untuk mengelolah berbagai macam data sehingga menjadi sebuah informasi yang bermanfaat untuk penggunanya dan juga sistem yang berkaitan.

3. Harip Santoso

Menurut Harip Santoso, Aplikasi merupakan sebuah kelompok file (*class*, *form*, *report*) yang ditujukan sebagai pengeksekusi aktivitas tertentu yang saling berkaitan seperti contohnya aplikasi *payroll* dan aplikasi *fixed asset*.

4. Yuhefizer

Menurut Yuhefizar, Aplikasi adalah program yang sengaja dibuat dan dikembangkan sebagai pemenuh kebutuhan penggunanya dalam menjalankan suatu pekerjaan tertentu.

5. Hengky W.Pranama

Menurut Hengky W. Pramana**,** pengertian aplikasi adalah satu unit perangkat lunak yang sengaja dibuat untuk memenuhi kebutuhan akan berbagai aktivitas ataupun pekerjaan, seperti aktivitas perniagaan, periklanan, pelayanan masyarakat, game, dan berbagai aktivitas lainnya yang dilakukan oleh manusia.[2].

2.4 Android

Android adalah sistem operasi untuk mobile yang ditujukan untuk perangkat layar sentuh. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi. Aplikasi-aplikasi yang dibangun menggunakan Bahasa pemrograman *java.* Pada awalnya android dikembangkan oleh sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk mobile yaitu Android Inc yang kemudian dibeli oleh *Google Inc*. Saat ini android sudah mengeluarkan versi ke 8.0 dan terus berkembang. [3].

2.4.1. Versi Android

Android megeluarkan versi 1.0 pada tanggal 23 September 2008 yang merupakan versi pertama dari android dan hingga saat ini adalah versi 8.0 yang keluar pada 21 Agustus 2017.

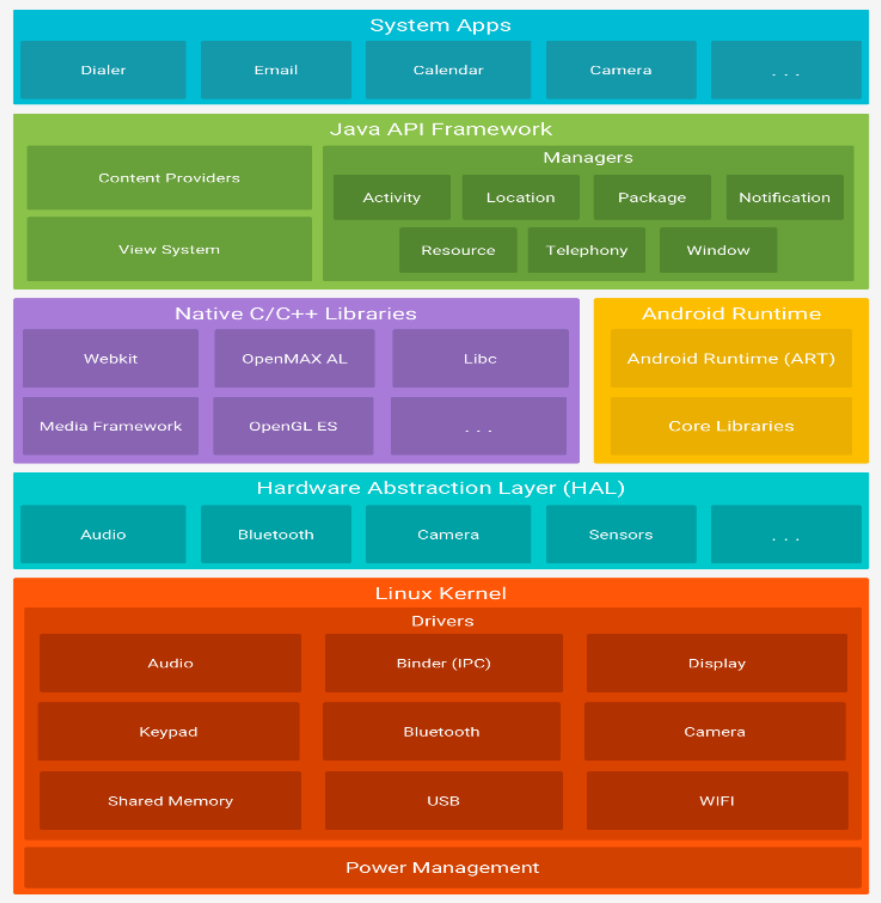
Berikut daftar versi android dapat dilihat pada tabel 2.1 :

Tabel 2. 1 Daftar Versi Android

Nama Versi	Versi	Tanggal Rilis
(*No Codename*)	1.0	23 September 2008
(*Internally known as "Petit Four*")	1.1	9 February 2009

Cupcake	1.5	27 April 2009
Donut	1.6	15 September 2009
Éclair	2.0	26 Oktober 2009
Froyo	2.2	20 Mei 2010
Gingerbread	2.3	6 Desember 2010
Honeycomb	3.0	22 Februari 2011
Ice Cream Sandwich	4.0	18 Oktober 2011
Jelly Bean	4.1	9 Juli 2012
Kitkat	4.4	31 Oktober 2013
Lollipop	5.0	12 November 2014
Marshmellow	6.0	5 Oktober 2015
Nougat	7.0	22 Agustus 2016
Oreo	8.0	21 Agustus 2017

Android merupakan tumpukan perangkat lunak berbasis Linux yang dibuat untuk berbagai perangkat dan faktor bentuk. Diagram berikut menunjukkan komponen besar dari platform Android :



Sumber : https://developer.android.com/guide/platform/

Gambar 2. 1 Tumpukan perangkat lunak android

Berikut adanya jelasan dari tumpukan perangkat lunak android :

2.4.1.1. *Linux Kernel*

Linux Kernel adalah fondasi platform android yang berfungsi threading dan manajemen memori tingkat rendah. Menggunakan kernel linux memungkinkan android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen untuk mengembangkan perangkat keras.

2.4.1.2. *Hardware Abstraction Layer* (HAL)

Hardware Abstraction Layer (HAL) menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja *Java API*. HAL teridiri dari beberapa modul yang masing-masing mengimplementasikan antarmuka untuk komponen perangkat tertentu.

2.4.1.3. *Android Runtime*

Untuk perangkat yang menjalankan Android versi 5.0 (*API level 21*) atau yang lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan tahap *Android Runtime* (ART). ART ditulis guna menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi *file* DEX, format *bytecode* yang didesain khusus untuk Android yang dioptimalkan untuk footprint memori minimal.

2.4.1.4. Pustaka C/C+ Asli

Komponen dan layanan sistem android banyak dibuat dari kode asli yang memerlukan pustaka asli yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja *Java API* mengekspos fungsionalitas beberapa pustaka asli pada aplikasi.

2.4.1.5. Kerangka Kerja *Java* API

Keseluruhan fitur Android OS tersedia melalui API yang ditulis dalam Bahasa *Java*. API ini membentuk elemen dasar yang diperlukan untuk membuat aplikasi Android

2.4.1.6. Aplikasi Sistem

Android dilengkapi dengan aplikasi inti untuk email, sms, kalender, internet, kontak, dan lainnya. Aplikasi tidak memiliki status khusus yang ingin dipasang pengguna, sehingga beberapa aplikasi pihak ketiga dapat menjadi aplikasi utama.

2.5 *Web Service*

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa *compiler*.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat

dipinjam oleh aplikasi lain tanpa perlu mengetahui detil pemrograman yang terdapat di dalamnya.

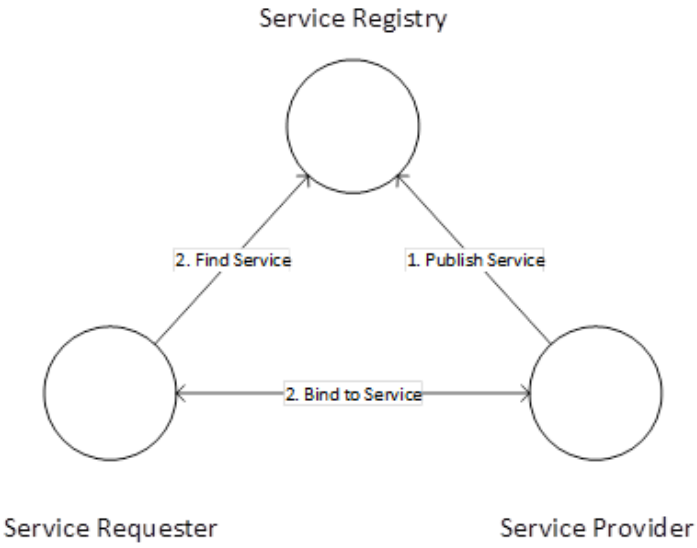
Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis *logic* atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-*upload* ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi *firewall*.

2.5.1 Arsitektur *Web Service*

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

a. *Service Provider*: Berfungsi untuk menyediakan layanan/*service* dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.
b. *Service Registry*: Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/*service* yang telah di-*register*.
c. *Service Requestor*: Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut. [4]



Sumber: Teknik Pemrograman Web Service PHP

Gambar 2. 2 Arsitektur Web Service

2.6 Java

Java menurut definisi Sun merupakan sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer stand alone ataupun lingkungan jaringan. Bahasa pemrograman *java* ini bersifat *case sensitive*, sehingga harus lebih memperhatikan bentuk tertentu, sehingga dalam menuliskan semua baris *code* tersebut dalam satu baris tersendiri [13].

Pada bulan Agustus 1991, *java* dikembangkan dengan nama "*Java* 2" yang merupakan generasi kedua dari *java* platform. Pergantian nama baru tersebut diterapkan di setiap semua edisi *Java* diantaranya : *Standard Edition* (J2SE), *Enterprise Edition* (J2EE), dan *Micro Edition* (J2ME) [5].

Java memiliki tiga *platform* yang masing-masing mengarahkan sebagai tujuan tertentu dan sebagai lingkungan komputasi yang berbeda-beda :

2.6.1. *Standard Edition (J2SE)*

J2SE merupakan inti dari sebuah bahasa pemrograman *Java*. J2SE didesain sebagai jalan pada komputer desktop dan komputer *workstations*.

2.6.2. *Enterprise Edition (J2EE)*

J2EE berfungsi sebagai built-in dan mendukung untuk *servelts*. JSP dan XML, edisi ini bertujuan untuk aplikasi yang berbasis *server*.

2.6.3. *Micro Edition (J2ME)*

J2ME didesain untuk piranti dengan keterbatasan memori, layar *display* terbatas dan power pemrosesannya terbatas juga.

Java dikenal sebagai bahasa pemrograman yang bersifat *strongly*, yang harus mendeklarasikan tipe data dari semua *variable* yang apabila salah dalam mengikuti aturan pendeklarasian *variable*, makan akan terjadi *error* pada saat proses kompilasi. Tipe data yang terdapat pada bahasa pemrograman *Java* pada umumnya yakni terdiri dari *integer, floating point*, *char*, dan *boolean* [5].

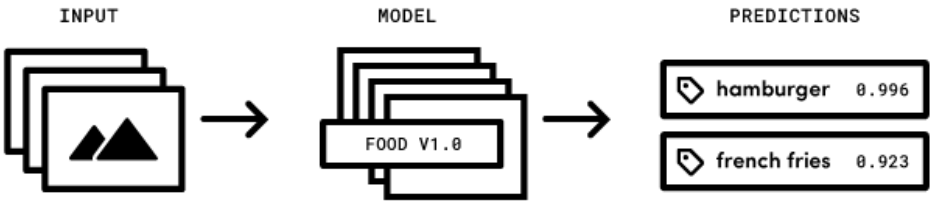
Dan untuk aturan penulisan pada bahasa pemrograman *Java* tidak jauh berbeda dengan bahasa pemrograman yang lain yakni harus diawali dengan huruf atau abjad, karakter mata uang, dan *underscore* (_) dan terdiri dari huruf atau abjad, angka,dan *underscore* (_). Dan tidak boleh menggunakan kata-kata yang dikenal oleh bahasa pemrograman *Java* (*keyword*), seperti *byte, case, int*, dan lainnya [5].

2.7 *Clarifai*

Clarifai adalah perusahaan kecerdasan buatan yang unggul dalam pengenalan visual (*Visual Recognition)*, Clarifai didirikan pada tahun 2013 oleh Mattew Zeiler, seorang ahli terkemuka dalam *Machine Learning*, Clarifai telah menjadi pemimpin pasar sejak memenangkan lima tempat teratas dalam klasifikasi citra pada kompetisi ImageNet 2013[6].

Clarifai adalah alat yang dapat mengindentifikasi atau mengenali gambar atau video yang dimasukan sebagai inputan yang dapat memberi hasil berupa prediksi tentang apa yang ada di dalam gambar atau video berupa besaran probabilitas kemungkinannya [7].

API *Clarifai* dibangun dengan konsep ide dasar sederhana dimana anda memasukan inputan berupa gambar atau video sebagai contoh apabila anda memilih model makanan untuk gambar masukkan maka Clarifai akan memprediksi gambar berdasarkan model makanan yang telah diketahui dan begitu seterusnya.



Sumber *: https://www.clarifai.com/developer/guide/#getting-started*

Gambar 2. 3 Konsep Kerja Clarifai

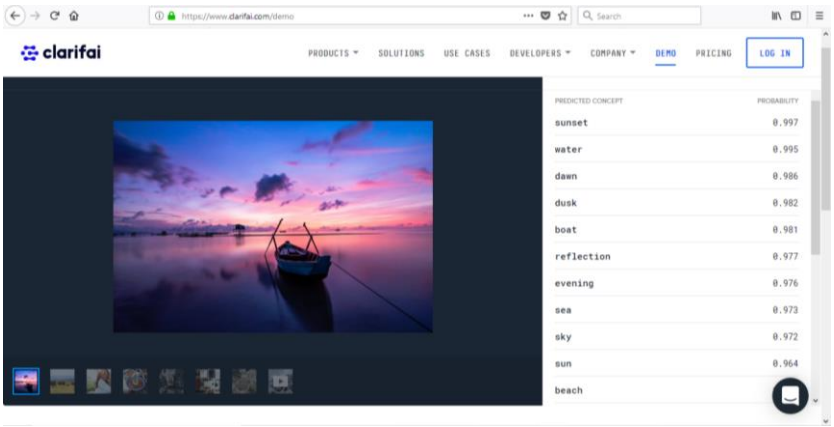
2.7.1. Model *Clarifai*

Gambar atau video yang menjadi data masukkan yang dijalankan ditandai oleh model dimana sebuah model adalah *classifier* yang terlatih dan dapat mengenali apa yang terdapat di dalam gambar atau video sesuai dengan yang telah diketahui. Model satu dan yang lainnya digunakan untuk mengetahui hal-hal yang berbeda pula. Menjalankan model yang berbeda untuk mengenali suatu gambar atau video dapat memberikan hasil yang berbeda secara drastis

Berikut adalah beberapa contoh model pada *Clarifai*:

2.7.1.1. *General Model*

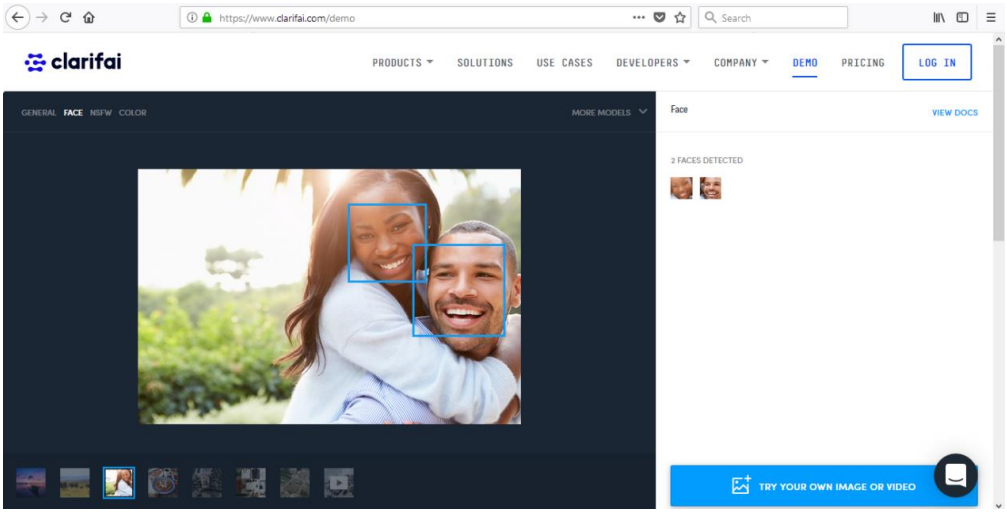
General model adalah model dimana berisi tag yang lebih umum dan berisi berbagai topik yang berbeda dan akan menghasilkan prediksi yang umum terdapat di gambar.



Gambar 2. 4 *General* model

2.7.1.2. *Face Model*

Face model memberikan hasil prediksi berupa besaran probabilitas yang terditeksi berdasarkan model wajah yang diketahui dari gambar yang anda masukkan



Gambar 2. 5 *Face Detection*

Masih banyak model pada clarifai seperti color, wedding, travel ,food, nfsw, moderation, demographic.

2.7.2. *Authentication*

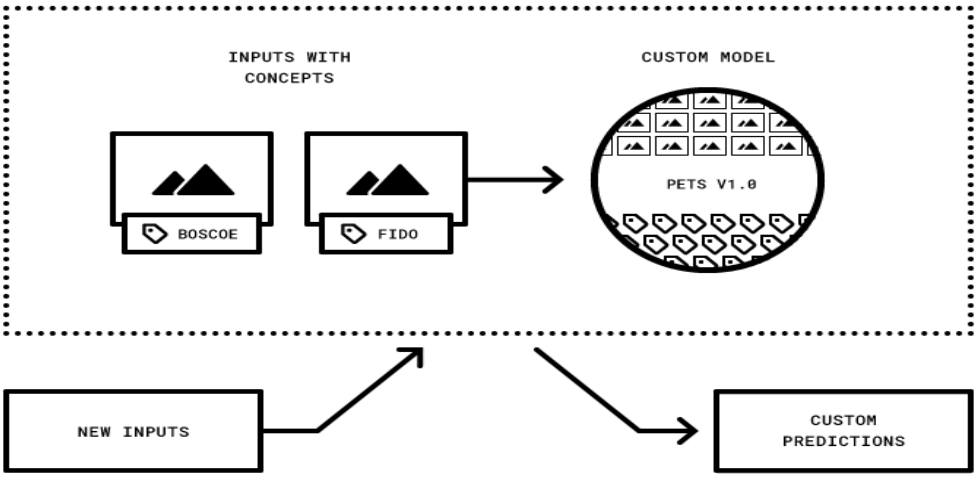
Otentikasi ke API ditangani melalui API k*ey*. Anda dapat membatasi cakupan API *key*, yang memungkinkan kunci untuk melakukan operasi yang sangat spesifik pada aplikasi yang diberikan, agar aplikasi anda tetap aman.

2.7.3. *Predict*

Predict, melakukan prediksi dan Analisa gambar yang anda masukkan dan memberi tahu anda apa isi yang terkandung di gambar. API akan mengembalikan daftar konsep dengan probabilitas yang sesuai dari seberapa besar kemungkinan konsep-konsep ini terkandung di dalam gambar. Bila anda membuat prediksi melalui API, anda akan diberi tahu model apa yang akan digunakan sebuah model berisi sekelompok konsep.Model hanya akan melihat konsep yang dikandungnya, anda dapat menggunakan model yang berbeda untuk menganalisa gambar dengan berbagai cara.

2.7.4. *Train*

Train atau pelatihan memungkinkan anda membuat sendiri model menggunakan konsep kustom anda sendiri. Anda mulai dengan menambahkan gambar yang anda sudah anda ketahui berisi konsep yang anda minati.anda tidak perlu memerlukan banyak gambar untuk memulai sebaiknya mulai dengan 10 dan tambahkan lebih banyak sesuai kebutuhan.kemudian membuat model dan menetukan konsep apa yang di kandungnya. Setelah membuat model, anda melatihnya berdasarkan gambar dan konsep yang akan diberikan. Operasi pelatihan ini bersifat asinkron mungkin perlu beberapa detik agar model anda dilatih dan siap sepenuhnya.setelah selesai latihan, anda bisa menggunakan model itu untuk memprediksi konsep tersebut pada gambar baru.



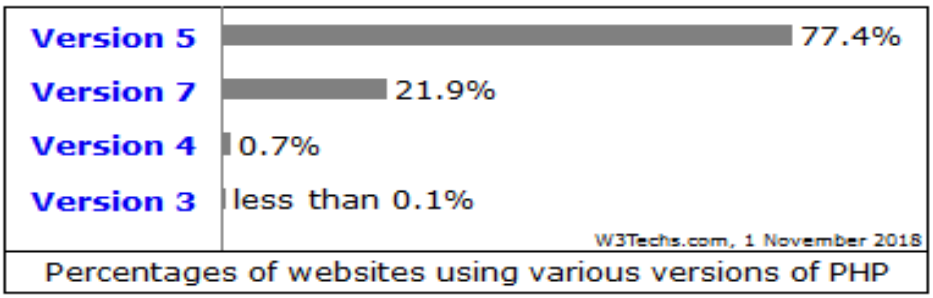
Sumber : *https://www.clarifai.com/developer/guide/train#train*

Gambar 2. 6 *Train*

2.8 PHP

PHP adalah bahasa pemrograman sumber terbuka yang ditujukan untuk pemrograman web dan dapat di aplikasikan ke HTML. PHP merupakan bahasa skrip yang ditanam dalam HTML. Ini berarti bahwa anda dapat menggabungkan kode PHP dan HTML dalam berkas yang sama [9]. PHP merupakan bahasa pemrograman yang akan menjalankan program di *server* dan hasilnya diintegrasikan ke dalam kode HTML. PHP diciptakan untuk membuat halaman *web* yang interaktif.

PHP cukup dikenal oleh banyak kalangan pemrogram *web*. Menurut W3Techs lembaga survei teknologi *web*, menyatakan 78.9% *website* menggunakan bahasa PHP [10]. PHP versi 5 adalah yang paling banyak digunakan saat ini, diikuti dengan versi 7 lalu versi 4 dan 3. Berikut adalah perbandingan penggunaan versi PHP yang dapat dilihat pada Gambar 7 :



Sumber: http://w3techs.com

Gambar 2. 7 Perbandingan Pengguna Versi PHP

Pemrograman PHP diapit dengan sintak <?php ?> dan ekstensi format *file php* adalah *.php*. *PHP* dapat digabung dengan *HTML*, namun hanya akan membaca kode yang berada di antara sintak <?php ?>, hasil pemrosesan akan ditampilkan dalam kode HTML. Contoh sebuah program PHP :

```
1.  <!DOCTYPE html>
2.  <html>
3.  <head>
4.  <title>Contoh Program PHP</title>
5.  </head>
6.  <body>
7.  <br>
8.  <?php
9.     echo 'Sekarang tanggal : '.date('j F Y').'.';
10. ?>
11. <br>
12. </body>
13. </html>
```

Gambar 2. 8 Contoh Bahasa pemograman PHP

2.9 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan. Sangat mudah dibaca dan menulisnya, serta mudah bagi komputer untuk mem-parsing dan menghasilkan. *JSON* format teks yang bebas Bahasa tetapi menggunakan konversi seperti C, C++, C#, *Java*, *Java*Script, Perl, Python, dan

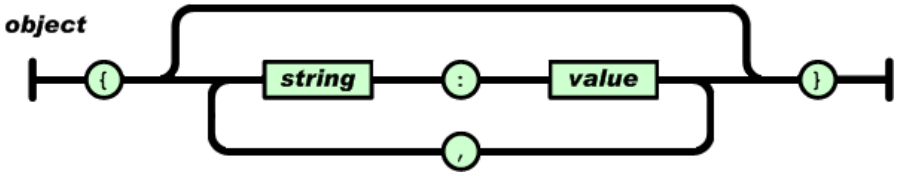
banyak lainnya. *JSON* kebanyakan juga sebagai Bahasa pertukaran data. *JSON* di bangun dengan dua struktur :

1. Kumpulan nama atau nilai. Dalam berbagai bahasa, ini derealisasikan sebagai *object*, *record*, *struct*, *dictionary*. *hash* tabel, *keyed list* atau *associative array*.
2. Daftar nilai yang terurut. Sebagian besar bahasa, ini direalisasikan sebagai *array*, *vector*, *list* atau *sequence*.

Struktur tersebut merupakan struktur data universal. Hampir semua Bahasa pemrograman yang modern sudah mendukungnya dalam satu bentuk atau lainnya. Masuk akal bahwa format data yang dapat dipertukarkan dengan bahasa pemrograman juga didasarkan pada struktur ini. *JSON* memiliki format – format tipe data seperti berikut [11] :

2.9.1. *Object*

Berikut adalah format *JSON object* yang dapat dilihat pada Gambar 9 :



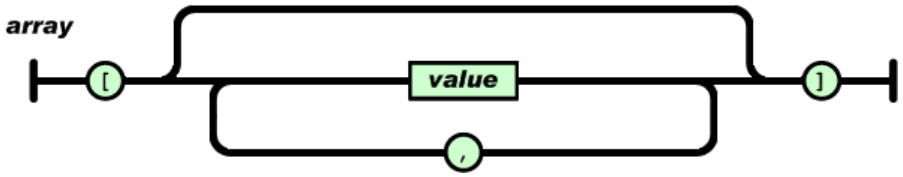
Sumber : *http://json.org/*

Gambar 2. 9 Format JSON *Object*

Object adalah kumpulan dari pasangan nama atau nilai tak berurutan. *Object* dimulai dengan "**{**" (kurung kurawal kiri) dan diakhiri dengan "**}**" (kurung kurawal kanan). Setiap nama diikuti oleh "**:**" (titik dua) dan pasangan nama atau nilai dipasangkan oleh "**,**" koma.

2.9.2. *Array*

Berikut adalah format *JSON array* yang dapat dilihat pada Gambar 10 :



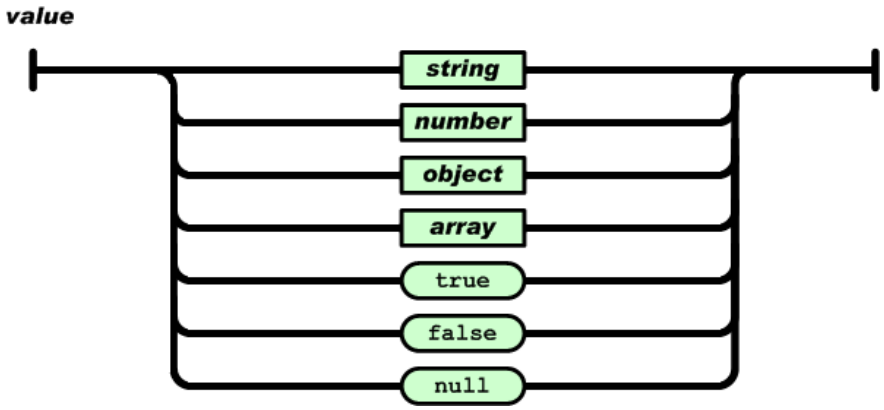
Sumber : *http://json.org/*

Gambar 2. 10 Format JSON *Array*

Array adalah kumpulan nilai yang diurutkan. *Array* dimulai dengan "**[**" (kurung siku kiri) dan diakhiri dengan "**]**" (kurung siku kanan). Nilai dipisahkan oleh "**,**" (koma).

2.9.3. *Value*

Berikut adalah format *JSON value* yang dapat dilihat pada Gambar 2. 11 :



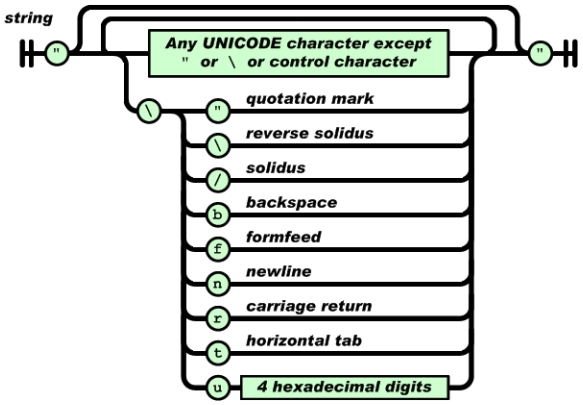
Sumber : *http://json.org/*

Gambar 2. 11 Format JSON *Value*

Value (nilai) dapat berupa *string* dalam tanda kutip ganda, *number*, *true* atau *false* atau *null*, *object* atau *array*. Struktur-struktur ini dapat diulang.

2.9.4. *String*

Berikut adalah format *JSON string* yang dapat dilihat pada Gambar 2.12 :



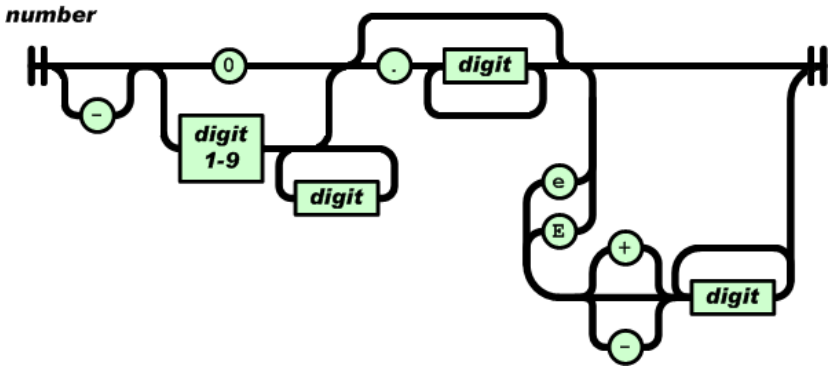
Sumber : *http://json.org/*

Gambar 2. 12 Format JSON *String*

String adalah urutan karakter kosong atau karakter *unicode* yang diapit oleh tanda kutip dua atau karekter khusus diawali dengan karakter "\" (garis miring terbalik). Karakter direpresentasikan sebagai *string* karakter tunggal. *String* sangat mirip dengan *string C* atau *Java*.

2.9.5. *Number*

Berikut adalah format *JSON number* yang dapat dilihat pada Gambar 2.13 :



Sumber : *http://json.org/*

Gambar 2. 13 Format JSON *Number*

Number sangat mirip dengan angka *C* atau *Java*, kecuali format *octal* dan *heksadesimal* tidak digunakan.

2.10 *MySQL*

MySQL adalah aplikasi basis data yang dijalanan di *server*, MySQL menggunakan sintak SQL dalam penggunaanya. MySQL bersifat relasional yang memungkinkan satu tabel dapat berelasi dengan yang lain, sehingga memungkinkan dilakukan normalisasi dengan tujuan untuk mengurangi redudansi. MySQL menggunakan *InnoDB* mampu menampung sebanyak 1017 kolom dan besar data mencapai 256TB [12].

Dalam penggunaan database MySQL, setiap peritah yang diketikkan disebut *query*. Perintah MySQL terdapat 3 sub perintah, yaitu *DDL (Data Definition Language)*, *DML (Data Manipulation Language)* dan *DCL (Data Control Language)*.

2.10.1. *DDL (Data Definition Language)*

Perintah dalam SQL yang pertama adalah perintah DDL. DDL sendiri mrupakan kependekan dari apa yang dikenal dengan nama Data Definition Language. DDL dapat berarti sebuah perintah yang berhubungan dengan pendefinisian dari suatu struktur database. Terdapat beberapa perintah DDL pada MySQL sebagai berikut :

1. *CREATE* berfungsi untuk membuat database baru, tabel baru, *view* baru dan kolom.
2. *ALTER* berfungsi untuk mengubah struktur tabel. Seperti mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom maupun memberikan atribut pada kolom.
3. *DROP* berfungsi untuk menghapus database dan tabel.
4. *TRUNCATE* berfungsi untuk Menghapus semua catatan dari tabel.
5. *COMMENT* berfungsi untuk Menambahkan komentar pada data.
6. *RENAME* berfungsi untuk mengubah nama obyek.

2.10.2. *DML (Data Manipulation Language)*

Data Manipulation Language (DML) adalah perintah yang digunakan untuk mengelolah data dalam database. Terdapat beberapa perintah DML pada MySQL sebagi berikut :

1. *SELECT* untuk menganampilkan data dari database.
2. *INSERT* untuk menambah data pada database.
3. *UPDATE* untuk merubah data dalam tabel.
4. *DELETEI* untuk menghapus data dari tabel.
5. *CALL* untuk memanggil subprogram SQL atau *Java*
6. *EXPLAIN PLAN* untuk menjelaskan jalur akses ke data.
7. *LOCK* TABEL untuk mengunci tabel.

2.10.3. DCL (Data Control Languange)

Data Control Language (DCL) ialah perintah yang digunakan untuk melakukan pengontrolan data dan server databasenya. Terdapat beberapa perintah DCL pada MySQL sebagai berikut :

1. *GRANT* berfungsi untuk memberikan hak akses pengguna ke database.
2. *REVOKE* berfungsi untuk menghilangkan hak akses yang telah diberikan dengan perintah *GRANT* [13].

Berikut adalah contoh sintak dasar yang perlu diketahui untuk mengelola basis data MySQL:

1. Membuat *Database*

```
1. create database db_mahasiswa;
```

Gambar 2. 14 Membuat Database di MySQL

2. Membuat Tabel

```
1. create tabel mahasiswa(
2.  id_mhs int(4) auto_increment primary key,
3.   nim varchar(12),
4.  nama_depan varchar(25) not null,
5.  nama_belakang varchar(25) not null,
6.  nama_belakang varchar(25) not null,
7.  )
```

Gambar 2. 15 Membuat Tabel di MySQL

3. Menambah Data

```
1. insert into mahasiswa(nim,nama_depan,nama_belakang)
2. value('10113082','Andhika','Eka');
```

Gambar 2. 16 Menambah Data di MySQL

4. Mengambil Data

```
1. select * from mahasiswa;
```

Gambar 2. 17 Mengambil Data di MySQL

5. Mengubah Data

```
1. update mahasiswa set nama_belakang = 'Putra'
2. where nim = '10113082';
```

Gambar 2. 18 Mengubah Data di MySQL

6. Menghapus Data

```
1. delete from mahasiswa where nim = '10113082';
```

Gambar 2. 19 Menghapus Data di MySQL

7. Menghapus Tabel

```
1. drop tabel mahasiswa;
```

Gambar 2. 20 Menghapus Tabel di MySQL

2.11 *Google Maps* API

Google Maps API adalah sebuah *library* yang berasal layanan dari *Google Maps* yang berbentuk *JavaScript*, yang berisi fungsi-fungsi pemrograman yang bisa di integrasikan kedalam web atau aplikasi yang sedang dibuat [14]. Hanya dengan menggunakan *Key Google Maps* API, aplikasi yang dibuat dapat mengakses *maps* pada *Google Maps*.

Google Maps API dapat digunakan secara gratis dan tidak perlu untuk mengeluarkan biaya sebagai lisensi. Tetapi *request* maksimal pada sebuah peta hanya diperbolehkan sebanyak 2500 *request*/ hari. Jika lebih pengguna harus membeli lisensi *Google Maps* API untuk bisnis. Karena *Google Maps* API berbasis

Web, maka untuk pembuatan aplikasinya hanya perlu menggunakan *tool Text Editor*, dan perangkat lain yang harus disiapkan yaitu *Browser*, dan koneksi internet.

Aplikasi perencanaan dan panduan wisata ini pun membutuhkan sebuah tampilan peta sebagai media informasi sebuah lokasi yang akurat maka dari itu dengan menerapkan teknologi *Google Maps* API pada aplikasi yang akan dibangun, selain bisa menampilkan lokasi geografis pada peta *Google*. Pada teknologi *Googel Maps* API ini pun bisa mengidentifikasi keberadaan si pengguna aplikasi itu berada.

2.11.1 *Sistem Penentuan Posisi Global* (GPS)

Global Positioning System (GPS) adalah nabigasi sistem atau penentuan posisi berbasis satelit. Sistem ini dirancang untuk memberikan informasi tentang posisi dan walti terus menerus di seluruh dunia. Penentuan posisi GPS dijelaskan menggunakan Nilai koordinat X dan Y di mana nilai garis bujur dan garis lintang. Sistem kerja GPS adalah mengirmkan sinyal dari GPS satelit ke perangkat yang sudah mempunyai fitur GPS, GPS tidak akan bekerja secara akurat dan optimal jika penggunaan berada did alam ruangan.

2.12 UML

UML *(Unified Modelling Language)* adalah bahasa pemodelan secara visual yang menjadi sarana perancangan sistem berorientasi objek, atau definisi UML yaitu bahasa yang sudah menjadi standar visualisasi, perancangan dan dokumentasi sistem perangkat lunak.

Beberapa tujuan dan fungsi dari penggunaan UML, diantaranya :

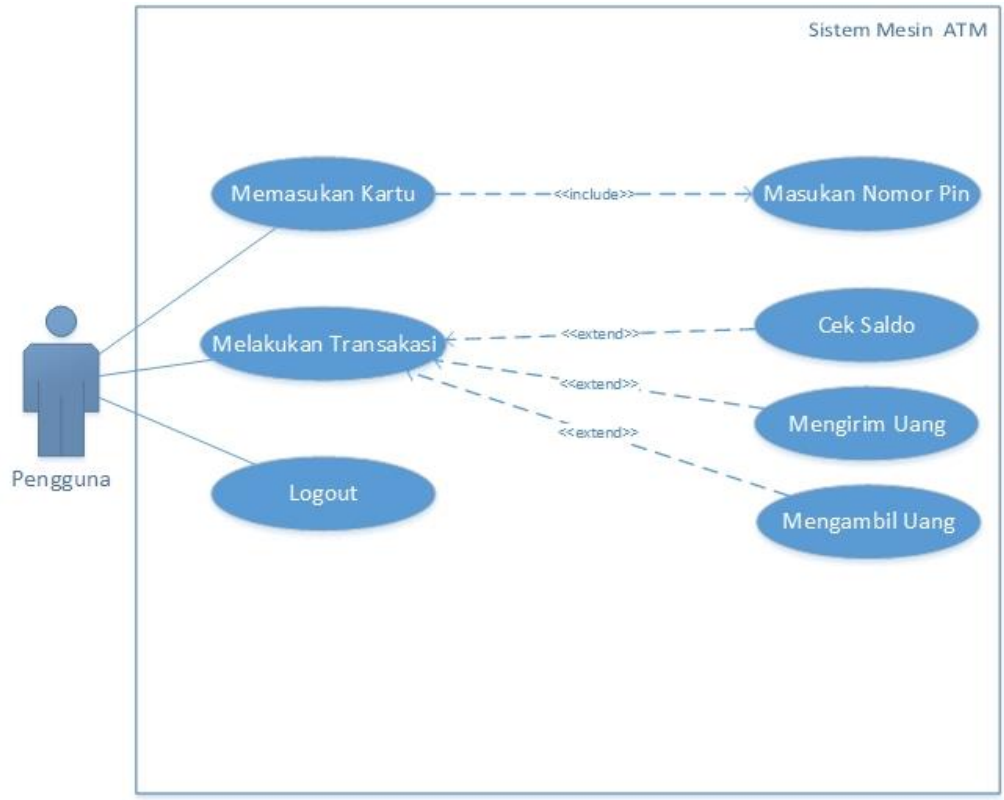
1. Dapat memberikan bahasa pemodelan visual kepada pengguna dari berbagai macam pemron proses rekayasa.
2. Dapat menyatukan praktek-praktek terbaik yang ada dalam pemodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa pemodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.

4. Dapat berguna sebagai blueprint, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak saja.
6. Dapat menciptakan suatu bahasa pemodelan yang nantinya dapat dipergunakan oleh manusia maupun mesin.

2.12.1. *Use case* Diagram

Use case diagram merupakan gambaran fungsionalitas dari beberapa atau semua *actor*, *use case*, dan interaksi untuk memperkenalkan suatu sistem. *Use case* diagram tidak menjelaskan secara detail tentang penggunaan *use case*, tetapi hanya memberikan gambaran singkat tentang hubungan *use case*, *actor*, dan sistem. Didalam *use case* hanya diketahui fungsi-fungsi yang berkaitan dengan pembangunan sistem yang akan dibuat.

Sebuah *use case* dapat di *include* oleh lebih dari suatu *use case* lain, shingga duplikasi fungsionalitas dapat dihindari dengan menarik keluar fungsionalitas yang *common*. *Use case* juga dapat meng *extend use case* lain dengan *behavior*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialis dari yang lain. Diagram *use case* memiliki beberapa *element* yang digunakan untuk menggambarkan rancangan dan penghubung antar komponen, baik *actor* dengan *use case* atau *use case* dengan *use case* yang lainnya.



Gambar 2. 21 Contoh *Use Case*

2.12.2. Skenario *Use case*

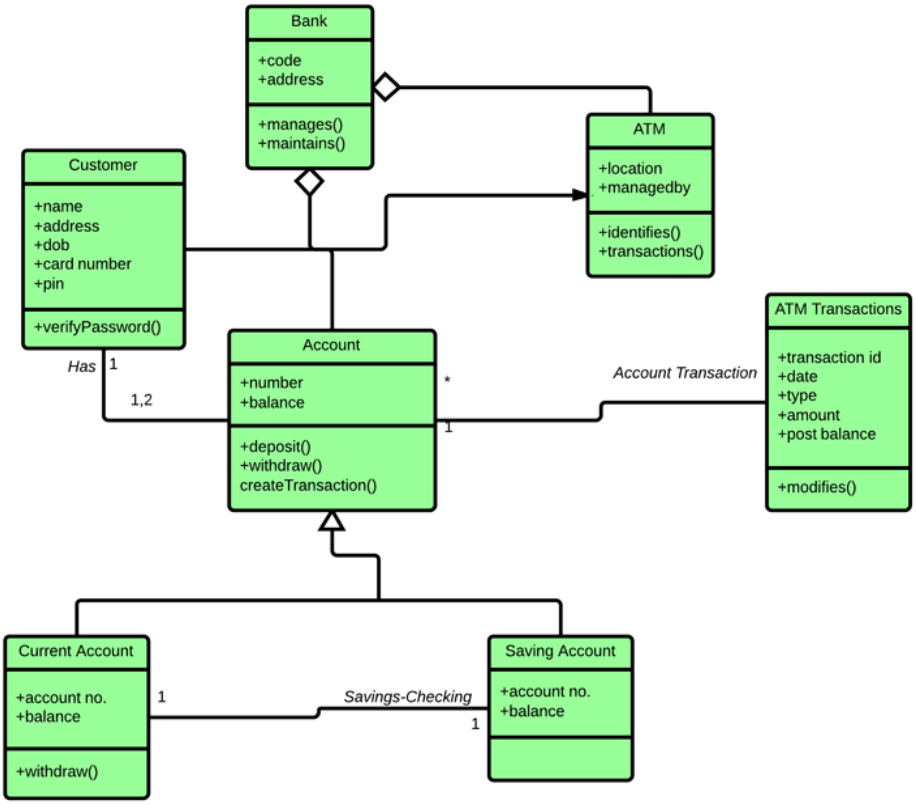
Skenario *Use case* merupakan hasil instansiasi dan penjelasan dari setiap *Use case*. Skenario *Use case* menceritakan detail yang terjadi. Format skenario *Use case* adalah berbentuk tabel.

Komponen-komponen Skenario *Use case* :

1. *Name*: Memberikan penjelasan singkat tentang nama dari *use case*.
2. *Actors:* Daftar aktor yang dapat mengakses *use case.*
3. *Goals*: Menjelaskan apa yang aktor coba dapatkan dari *use case*.
4. *Preconditions*: Kondisi sistem sebelum *use case* dijalankan.
5. *Summary*: Memberikan penjelasan singkat tentang deskripsi informal dari *use case*.
6. *Steps*: Menjelaskan setiap langkah yang dijalankan pada *use case*.
7. *Post conditions*: Kondisi sistem setelah *use case* dijalankan.

2.12.3. *Class diagram*

Class diagram merupakan elemen terpenting dalam sistem berorientasi objek, kelas mendeskripsikan satu blok pembangun sistem. *Class diagram* memiliki fitur yang memodelkan multiplisitas, ketampakan, penanda, *polymorphism*, dan karakter-karakter lainnya. UML menggunakan istilah fitur sebagai istilah umum yang meliputi *property* dan operasi sebuah kelas Terdapat empat bagian masing-masing bagan kelas, yaitu : nama, atribut dan perilaku.



Gambar 2. 22 Contoh *Class Diagram*

2.12.4. *Activity diagram*

Activity diagram digunakan untuk menganalisis proses, *activity diagram* biasanya dibuat untuk sebuah *use case* dan menampilkan beberapa skenario

Activity diagram bukan sebuah alat yang sempurna untuk menganalisis masalah dari system. Setiap *activity diagram* mempunyai satu *initial state*. *Activity diagram*

dapat diakhiri dengan memberikan *activity final node* yang digambarkan dengan lingkaran padat dengan mempunyai cincin dibagian luarnya.

2.12.4.1. *Activity Node*

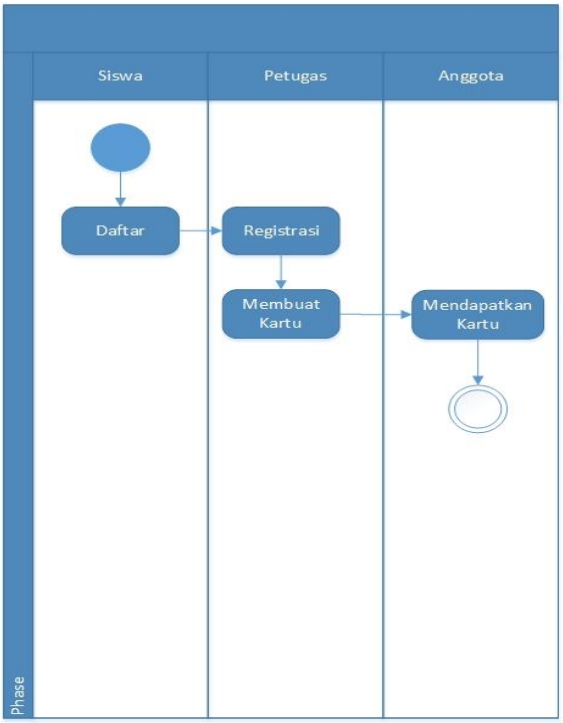
Activity node adalah sesuatu yang dilakukan atau sedang terjadi dalam *Activity diagram*. *Activity node* mempunyai simbol yang hampir mirip dengan *use case* namun mempunyai bentuk yang lebih ramping dan menyerupain bujur sangkar.

2.12.4.2. *Decision Node*

Descision node disebut *decision diamonds* di dalam flowchart. Simbol diamond adalah satu elemen yang membuat *activity diagram* mengingat akan *flowchart*, yang berguna untuk memberika kondisi percabangan.

2.12.4.3. *Transition Fork*

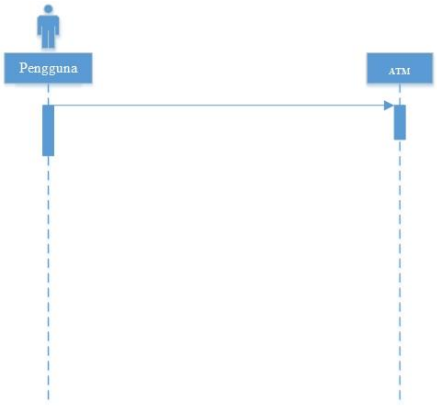
Transition fork untuk menggambarkan tingkah laku yang pararel atau bercabang. Sebuah *transition join* untuk mempertemukan tingkah laku yang pararel.



Gambar 2. 23 Contoh *Activity Diagram*

2.12.5. *Sequence diagram*

Diagram Sequence digunakan untuk memberikan gambaran interaksi antar objek di dalam dan di sekitar sistem berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* dapat diturunkan dari usecase dan untuk menurukan interaksi, relasi, dan metode dari objek sebuah sistem [20]. Diawali dengan men-*trigger* aktivitas, proses dan perubahan yang terjadi secara internal dan *output* apa yang dihasilkan. Setiap objek termasuk aktor memiliki *lifeline* vertikal. Pesan digambarkan sebagai garis panah dari satu objek ke objek yang lainnya.



Gambar 2. 24 Contoh *Sequence*