BAB 2
TINJAUAN PUSTAKA

2.1 Tempat Penelitian

Tempat penelitian di BBWS Cimanuk-Cisanggarung yang berlokasikan di Jl. Ahmad Yani Kav.1 Cirebon, Jawa Barat. Pada tahap ini merupakan tinjauan tempat penelitian yang terdiri dari profil BBWS Cimanuk-Cisanggarung, kedudukan, tugas, fungsi, visi dan misi.

2.1.1 Profil Balai Besar Wilayah Sungai Cimanuk Cisanggarung

Balai Besar Wilayah Sungai Cimanuk-Cisanggarung merupakan unit pelaksana teknik (UPT) pada Direktorat Jenderal Sumber Daya Air, Kementerian Pekerjaan Umum dan Perumahan Rakyat. Wilayah Kerja BBWS Cimanuk-Cisanggarung, wilayah sungai Cimanuk-Cisanggarung, termasuk klasifikasi wilayah sungai lintas Provinsi Jawa Barat dan Jawa Tengah, meliputi wilayah Kabupaten Majalengka, Kabupaten Indramayu, Kabupaten Kuningan, Kabupaten Cirebon, Kota Cirebon, dan Kabupaten Brebes. [1]

2.1.2 Kedudukan

Kedudukan dalam BBWS sebagai berikut:

a. Balai Besar Wilayah Sungai berada di bawah dan bertanggung jawab kepada Direktur Jenderal Sumber Daya Air
b. Balai Besar dipimpin oleh Kepala Balai. [1]

2.1.3 Tugas

Balai Besar Wilayah Sungai mempunyai tugas melaksanakan pengelolaan sumber daya ari di wilayah sungai yang meliputi perencanaan, pelaksanaan konstruksi, operasi dan pemeliharaan dalam rangka konservasi dan pendayagunaan sumber daya air dan pengendalian daya rusak air pada sungai danau, waduk, bendungan, dan tampungan air lainnya, irigasi, air tanah, air baku, rawa, tambak dan pantai. [1]

2.1.4 Fungsi

Adapun fungsi dari BBWS Cimanuk-Cisanggarung, diantaranya :

1. Penyusunan pola dan rencana pengelolaan sumber daya air pada wilayah sungai.
2. penyusunan rencana dan program, studi kelayakan dan perencanaan teknis/desain/pengembangan sumber daya air.
3. persiapan, penyusunan rencana dan dokumen pengadaan barang dan jasa.
4. Pelaksanaan pengadaan barang dan jasa serta penetapan pemenang selaku Unit Layanan Pengadaan (ULP).
5. Pengendalian dan pengawasan konstruksi pelaksanaan pembangunan sumber daya air.
6. Penyusunan rencana dan pelaksanaan pengelolaan kawasan lindung sumber air pada wilayah sungai.
7. Pengelolaan sumber daya air yang meliputi konservasi dan pendayagunaan sumber daya air serta pengendalian daya rusak air pada wilayah sungai.
8. Pengelolaan sistem hidrologi.
9. Pengelolaan sistem informasi sumber daya air.
10. Pelaksanaan operasi dan pemeliharaan sumber daya air pada wilayah sungai.
11. Pelaksanaan bimbingan teknis pengelolaan sumber daya air yang menjadi kewenangan provinsi dan kabupaten/kota.
12. Penyiapan rekomendasi teknis dalam pemberian ijin atas penyediaan, peruntukan, penggunaan dan pengusahaan sumber daya air pada wilayah sungai.
13. Fasilitasi kegiatan tim koordinasi pengelolaan sumber daya air pada wilayah sungai.
14. Pemberdayaan masyarakat dalam pengelolaan sumber daya air.
15. Pelaksanaan penyusunan laporan akuntansi keuangan dan akuntansi barang milik negara selaku Unit Akuntansi Wilayah.

16. Pelaksanaan pemungutan, penerimaan dan penggunaan biaya jasa pengelolaan sumber daya air (BJPSDA) sesuai dengan ketentuan peraturan undang - undang, dan
17. Pelaksanaan urusan tata usaha dan rumah tangga balai serta koordinasi dengan instansi terkait. [1]

2.1.5 Visi dan Misi

Visi BBWS Cimanuk-Cisanggarung adalah terwujudnya kemanfaatan sumber daya air yang berkelanjutan untuk sebesar-besar kesejahteraan rakyat. [12] Sedangkan Misi dari BBWS Cimanuk-Cisanggarung, sebagai berikut:

1. Konservasi SDA secara berkelanjutan.
2. Pendayagunakan SDA secara adil serta memenuhi persyaratan kualitas dan kuantitas untuk berbagai kebutuhan masyarakat.
3. Pengendalikan daya rusak air.
4. Pemberdayakan dan meningkatkan peran masyarakat dan Pemerintah dalam pengelolaan SDA.
5. Peningkatkan keterbukaan serta ketersediaan data dan informasi dalam pengelolaan SDA. [12]

2.1.6 Logo

Balai Besar Wilayah Sungai Cimanuk-Cisanggarung merupakan unit pelaksana teknik (UPT) pada Direktorat Jenderal Sumber Daya Air, Kementerian Pekerjaan Umum dan Perumahan Rakyat. Maka untuk logo BBWS mengikuti logo Pekerjaan Umum (PU). Logo PU dapat dilihat pada Gambar 2.1 Logo PU



Sumber : https://www.pu.go.id/article/89/motto-dan-lambang
Gambar 2.1 Logo PU

A. Makna Logo

Logo secara keseluruhan menggambarkan peran Kementerian Pekerjaan Umum dan Perumahan Rakyat untuk mewujudkan Infrastruktur Pekerjaan Umum dan Perumahan Rakyat Yang Handal Dalam Mendukung Indonesia Yang Berdaulat, Mandiri, dan Berkepribadian Berlandaskan Gotong Royong".

B. Warna

Warna Dasar Lambang adalah Kuning (kuning Kunyit) memiliki makna lambang keagungan, Ketuhanan Yang Maha Esa, Kemakmuran, menggambarkan bahwa dalam mengemban fungsi dan peranan pembangunan di bidang Pekerjaan Umum dan Perumahan Rakyat diabadikan dalam rasa puji syukur terhadap Keagungan Tuhan Yang Maha Esa.

Warna Biru kehitam-hitaman mengandung arti Keadilan Sosial, Keteguhan hati, Kesetiaan pada tugas dan Ketegasan bertindak.

C. Arti Simbolis

Baling-Baling : Menggambarkan "DINAMIKA". Berdaun 3 yang merupakan segitiga berdiri tegak lurus menggambarkan "STABILITAS". Secara keseluruhan menggambarkan "DINAMIKA YANG STABIL" dan "STABILITAS YANG DINAMIS".

Bagian Daun Baling-Baling Yang Mengarah Keatas : Melambangkan "PENCIPTAAN PERUMAHAN DAN PERMUKIMAN".

Bagian Lengkungnya Dari Daun Baling-Baling : Memberikan perlindungan untuk ruang kerja dan tempat tinggal bagi manusia.

Bagian Daun Baling-Baling Yang Mengarah Ke Kiri Dengan Bagian Lengkungnya Yang Telungkup : Menggambarkan penguasaan bumi dan alam dan pengusahaan untuk sebesar-besarnya kemakmuran rakyat. Garis Horizontal: bentang jalan jembatan di atas sungai sebagai usaha untuk pembukaan dan pembinaan daerah.

Bagian Daun Baling-Baling Yang Mengarah Ke Kanan Dengan Bagian Lengkungnya Yang Terlentang : Menggambarkan usaha pengendalian dan penyaluran untuk dimanfaatkan bagi sebesar-besarnya kemakmuran rakyat. Garis Horizontal: Menggambarkan penampang dari saluran air

Baling-Baling Dengan 3 Daun Ini Menggambarkan : Tiga unsur kekaryaan Kementerian Pekerjaan Umum dan Perumahan Rakyat, yaitu: Tirta, Wisma (Cipta), dan Marga. Trilogi Kementerian Pekerjaan Umum dan Perumahan Rakyat adalah Bekerja Keras, Bergerak Cepat, dan Bertindak Tepat. [13]

2.2 Landasan Teori

Beberapa landasan teori menjadi acuan dalam menulis tugas akhir seperti penjelesan sungai, wilayah sungai, daerah aliran sungai, banjir, pengungsian, rekomendasi tempat pengungsian, *Simple Additive Weighting* (SAW), *euclidean*, teknologi informasi, teknologi geotagging, Google API, Android, Android Studio, JDK, *emulator*, *website*, *framework*, *Model View Controller (MVC)*, *codeIgniter*, ,PHP, MySQL, *Object-Oriented and Desaign* (OODA).

2.2.1 Sungai, Wilayah Sungai, Daerah Aliran Sungai

Berikut pengertian masing-masing dari sungai, wilayah sungai, daerah aliran sungai:

a. Sungai adalah alur atau wadah air alami dan/atau buatan berupa jaringan pengaliran air beserta air di dalamnya, mulai dari hulu sampai muara, dengan dibatasi kanan dan kiri oleh garis sempadan.
b. Wilayah sungai adalah kesatuan wilayah pengelolaan sumber daya air dalam satu atau lebih daerah aliran sungai dan/atau pulau-pulau kecil yang luasnya kurang dari atau sama dengan 2.000 (dua ribu) Km2.
c. Daerah aliran sungai adalah suatu wilayah daratan yang merupakan satu kesatuan dengan sungai dan anak-anak sungainya, yang berfungsi menampung, menyimpan, dan mengalirkan air yang berasal dari curah hujan ke laut secara alami, yang batas di darat merupakan pemisah topografis dan batas di laut sampai dengan daerah perairan yang masih terpengaruh aktivitas daratan. [13]

2.2.2 Banjir

Berikut pengertian banjir yang dipaparkan oleh kementerian PU dan BNPB Jakarta:

1. Banjir adalah peristiwa meluapnya air sungai melebihi palung sungai. [13]
2. Banjir terjadi apabila sejumlah besar air menggenangi permukaan tanah yang biasanya kering. Banjir adalah bencana alam yang paling sering terjadi dan paling banyak merugikan, baik dalam segi kemanusiaan maupun ekonomi. Menurut data dari Badan Nasional Penanggungan Bencana (BNPB), 90 % dari kejadian bencana alam di Indonesia berhubungan dengan banjir. [14]

Salah satu banjir yang pernah terjadi di daerah Ciledug, Cirebon yang terjadi akibat luapan sungai Ciledug pada tanggal 24 February 2018, foto banjir Ciledug dapat dilihat pada Gambar 2.2 Banjir Ciledug.

.



Sumber:
http://cdn.metrotvnews.com/dynamic/photos/2014/07/06/4825/a2.jpg?w=1111
Gambar 2.2 Banjir Ciledug

Berikut beberapa penyebab banjir yang sering didengar oleh masyarakat diantara lain :

1. Hujan dalam jangka waktu yang panjang.
2. Erosi Tanah atau buruknya penanganan sampah yang menyebabkan air dari sungai dan saluran-saluran meluap dan menggenangi daerah sekitarnya.
3. Pembangunan dan perkembangan tempat pemukiman, dimana tanah kehilangan daya serap air hujan. Perkembangan tempat pemukiman bisa menyebabkan 6 kali lipat, terutama di Kota besar yang perkembangannya tidak terencana dengan baik.
4. Bendungan dan saluran air yang rusak, walaupun tidak sering terjadi namun bisa menyebabkan banjir terutama saat hujan deras dengan durasi yang panjang.
5. Keadaan Tanah dan Tanaman, tanah yang ditumbuhi banyak tanaman mempunyai daya serap air yang besar. Tanah yang tertutup semen, paving atau aspal sama sekali tidak menyerap air.

6. Daerah Bebatuan, daya serap air sangat kurang sehingga bisa menyebabkan banjir kiriman. [14]

Banyak hal yang dapat dilakukan oleh masyarakat untuk mencegah banjir salah satunya adalah mengurangi dampak. Dengan kondisi saat ini, mengurangi dampak dari bencana banjir adalah hal yang paling realistis untuk kita lakukan. Bagaimana caranya?

1. Mengurangi kemungkinan dengan melakukan tindakan pencegahan dilingkungan masing-masing.
2. Mengurangi risiko dengan mengetahui cara pencegahannya agar kerusakan yang timbul akibat bencana dapat berkurang.
3. Mengurangi Korban yang timbul akibat kurangnya kesiapan, oleh karena itu persiapan yang baik akan bisa membantu masyarakat untuk melakukan tindakan yang tepat guna dan tepat waktu.
4. Menjalin kerjasama dengan baik untuk memperlancar proses penanggulangan bencana, karena pada hakikatnya Pemerintah, Dunia Usaha dan Masyarakat wajib bekerja sama dalam upaya penanggulangan bencana.
5. Peringatan dini adalah serangkaian kegiatan pemberian peringatan sesegera mungkin kepada masyarakat tentang kemungkinan terjadinya bencana pada suatu tempat oleh lembaga yang berwenang.
6. Mitigasi adalah serangkaian upaya untuk mengurangi risiko bencana, baik melalui pembangunan fisik maupun penyadaran dan peningkatan kemampuan menghadapi ancaman bencana. [14]

Pada Gambar 2.3 Pelebaran Sungai merupakan salah satu bentuk upaya penanggulangan banjir yang berada di sungai cikalong DAS kalijaga, Perumnas Cirebon.



Gambar 2.3 Pelebaran Sungai

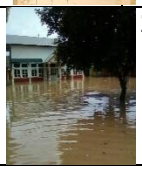
2.2.3 Pengungsian

Pengungsian adalah seseorang atau sekelompok orang yang meninggalkan tempat tinggalnya atau wilayahnya untuk berlindung dari atau mencari tempat aman dari suatu bencana. Biasanya bencana yang terjadi dalam pengungsian ini berupa bencana dalam bentuk banjir, longsor, letusan merapi, tsunami, dan kebakaran. Pengungsi biasanya ditempatkan yang memudahkan para relawan dalam menolongnya. Durasi pengungsian berdasarkan kondisi dan situasi bencana yang terjadi, jika sudah aman maka pengungsian diperbolehkan menempatkan kembali rumah atau wilayanya.

2.2.4 Kebanjiran di DAS Cimanuk-Cisanggarung

Kebanjiran yang sering terjadi di daerah aliran sungai Cimanuk-Cisanggarung dapat dilihat pada Tabel 2.1 Kebanjiran di DAS Cimanuk-Cisanggarung

Tabel 2.1 Kebanjiran di DAS Cimanuk-Cisanggarung

DAS	Sungai	Daerah	Banjir	Tgl	Pengungsian
Cibuaya	Pangkalan	Pangkalan		23 /01/ 2014	
Cimanuk	Cimanuk	Jatibarang		12 /02/ 2018	
Cimanuk	Cimanuk	Pilangsari		16 /03/ 2018	
Cimanuk	Cimanuk	Sekarwangi		22 /11/ 2017	
Jatiroke	Jamblang	Grogol		10 /02/ 2018	
Kalibangka	Pangka Deres	Pangarengan		17 /02/ 2018	
Ciberes	Ciberes	Kalimekar		02 /02/ 2017	
Cisanggarung	Cisanggarung	Tawangsari		26 /02/ 2018	
Cisanggarung	Cisanggarung	Bojongsari		01 /03/ 2018	
Cisanggarung	Cijengklok	Cilengkrang		23 /02/ 2018	
Cisanggarung	Cisanggarung	Cigobong		23 /02/ 2018	

2.2.5 Rekomendasi Tempat

Rekomendasi tempat adalah saran tempat yang sifatnya menganjurkan, membenarkan, atau menguatkan mengenai tempat yang terbaik. Rekomendasi tempat ini sangat penting dan membantu masyarakat dalam meyakinkan yang masih bingung mencari tempat. Dalam rekomendasi ini biasanya memiliki kriteria yang menjadi acuan utama, misal tempat yang nyaman atau tempat yang sering dipilih oleh orang lain.

2.2.6 Simple Additive Weighting (SAW)

Metode *Simple Additive Weighting* (SAW) dapat pula disebut sebagai metode penjumlahan terbobot. Pada dasarnya metode SAW digunakan untuk mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif di setiap kriteria yang ada. Alternatif yang ada dihauskan untuk melalui poses normalisasi matrik keputusan (X) dengan berdasarkan skala yang berasal dari bobot masing-masing atribut. Berikut merupakan langkah penyelesaian dengan menggunakan metode SAW :

1. Menentukan Ci yang merupakan kriteria saja yang akan dijadikan acuan dalam pengambilan keputusan.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan (Ci), kemudian kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R [4]. rumus yang digunakan untuk melakukan normalisasi sebagai berikut:

$$Rij = \begin{cases} \dfrac{X_{ij}}{\max_i X_{ij}} & \text{Jika j adalah atribut keuntungan (benefit)} \\ \dfrac{\min_i X_{ij}}{X_{ij}} & \text{Jika j adalah atribut biaya (cost)} \end{cases}$$

Sumber Gambar : Teknologi Informasi dan Ilmu Komputa [4].

Gambar 2.4 Rumus Normalisasi

2.2.7 Euclidean

Euclidean distance adalah perhitungan jarak dari 2 buah titik dalam *Euclidean space*. *Euclidean space* diperkenalkan oleh seorang matematikawan dari Yunani sekitar tahun 300. untuk mempelajari hubungan antara sudut dan jarak. *Euclidean* ini biasanya diterapkan pada 2 dimensi. kemudian juga bisa sederhana jika diterapkan pada dimensi lain yang lebih tinggi [11].

Perhitungan Algoritma Euclidean Distance

Euclidean ini berkaitan dengan *Teorema Phytagoras* dan biasanya diterapkan pada 1, 2 dan 3 dimensi.

1. **Penerapan 1 dimensi**

   Semisal ingin menghitung jarak *Euclidean* 1 dimensi. Titik pertama adalah 4, titik kedua adalah -10. Caranya adalah kurangkan -10 dengan 4. sehingga menghasilkan -14. Cari nilai absolut dari nilai -14 dengan cara mempangkatkannya sehingga mendapat nilai 196. Kemudian diakarkan sehingga mendapatkan nilai 14. Sehingga jarak *euclidean* dari 2 titik tersebut adalah 14 yang ditunjukkan pada Persamaan (1) [11].

   $$Ed= \sqrt{(X-Y)^2} \quad \text{.....(1)}$$

   Ket : X= Titik Pertama, Y= Titik Kedua

2. **Penerapan 2 Dimensi**

   Berbeda dengan penerapan euclidean distance 1 dimensi, aplikasi pariwisata kota Palembang menerapkan euclidean distance dengan model 2 dimensi dikarenakan sesuai dengan bentuk pencarian menggunakan 2 titik koordinat pada wilayah permukaan bumi, yang ditunjukan oleh Persamaan (2).

   $$Ed= \sqrt{(X_2-X_1)^2 + (Y_2-Y_1)^2} \quad \text{.....(2)}$$

   Persamaan (2) dapat diimplementasikan kedalam bentuk koordinat seperti yang ditunjukkan oleh Persamaan (3). [11]

   $$Ed= \sqrt{(Lat_2-Lat_1)^2 + (Lng_2-Lng_1)^2} \quad \text{.....(3)}$$

2.2.8 Teknologi Informasi

Teknologi Informasi adalah kajian, desain, pengembanganm 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 pengertia 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. [15]

2.2.9 Teknologi Geotagging

Geotagging merupakan proses penambahan informasi *geospasial* pada berbagai media digital. Media yang telah mengalami proses *geotagging* akan memiliki informasi koordinat berupa *longitude* (bujur), *latitude* (lintang) dan *altitude* (ketinggan). Hal tersebut memungkinkan media dapat diposisikan secara tepat pada peta. Pada penelitian ini metode *Geotagging* yang digunakan untuk melakukan penambahan informasi lokasi pada metadata foto adalah *Geotagging*. Algoritma penambahan ditulis di dalam script aplikasi dan otomatis akan berjalan saat pengguna mengunggah laporan. Lokasi yang ditambahkan merupakan lokasi pengguna yang didapatkan oleh aplikasi [7]. Pada Gambar 2.5 Teknologi Geotagging merupakan salah satu contoh penerapan teknologi geotagging dan pada Tabel 2.2 Penjelasan Gambar 2.5 Teknologi Geotagging



Gambar 2.5 Teknologi Geotagging

Tabel 2.2 Penjelasan Gambar 2.5 Teknologi Geotagging

Orieantation	***Latitude***
X : 318.0	36.066138179999996
Y : 0.0	***Longitude***
Z : 60.0	-94.17368506000001

2.2.10 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 mengntegrasikan 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. [7]

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, Google maps yang menunjukan 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&c
allback=initMap"
    async defer></script>
```

Pada sebuah *source code* terdapat kata *YOUR_API_KEY*, dimana kata tersebut diganti dengan kunci API yang telah dapatkan pada tahap pertama.

Google MAPS API untuk Android :

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

Pada folder app/res/values/google_maps_api.xml ubah kata *YOUR_KEY_HERE* dengan kunci API yang telah dapatkan 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>
```

2.2.11 Firebase API

Firebase merupakan teknologi *back-end as a service* (BaaS) berbasis *cloud* dari Google yang menyediakan berbagai layanan, salah satunya adalah fitur *push notification* bernama *firebase cloud messaging*. *Firebase cloud messaging* merupakan perkembangan dari Google *Cloud Messaging* (GCM) yang menyediakan layanan agar developer dapat mengirimkan pesan notifikasi ke perangkat android dari server.

Alur kerja bagaimana *firebase cloud messaging* dapat mengirimkan pesan ke perangkat android pengguna yang yang dijelaskan tahapnya sebagai berikut :

a. Ketika aplikasi pada perangkat baru pengguna dijalankan, secara otomatis akan melakukan registrasi FCM token.

b. *Firebase Cloud Messaging* melakukan *generate* FCM token dan menyimpannya pada akun pengguna.
c. Aplikasi mengaktifkan *trigger* untuk meminta *Firebase Cloud Messaging* mengirimkan *push notification*.
d. Server memproses data yang ada pada basis data dan menyeleksi sesuai dengan FCM token yang diminta.
e. Server mengirimkan daftar FCM token yang menjadi tujuan dari pesan *push notification*.
f. *Firebase Cloud Messaging* mengirimkan pesan *push notification* ke perangkat android yang telah diseleksi. [10]

2.2.12 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikan Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008. Logo Android dapat dilihat pada Gambar 2.6 Logo Android.



Sumber :https://vignette.wikia.nocookie.net/tolololpedia/images/4/4f/Logo_Android.png/revision/latest?cb=20130226093554

Gambar 2.6 Logo Android

Antarmuka pengguna Android umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk, dan mencubit untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks. Selain perangkat layar sentuh, Google juga telah mengembangkan Android TV untuk televisi, Android Auto untuk mobil, dan Android Wear untuk jam tangan. Masing-masing memiliki antarmuka pengguna yang berbeda Varian Android juga digunakan pada komputer jingjing, komputer permainan, kamera digital, dan peralatan elektonik lainnya. [16]

1. Fitur Android

Perangkat Android memiliki beberapa fitur perangkat keras di dalamnya. Inilah yang dapat di manfaatkan developer dalam membangun aplikasi. Diantaranya:

1. *Touchscreen*
2. *GPS*
3. *Accelerometer*
4. *SD Card*

Android memiliki banyak fitur perangkat lunak yang dapat digunakan oleh developer dalam mengembangkan aplikasi, yakni:

1. Internet
2. Audio dan Video Support
3. Contact
4. Security
5. Google APIs [16]

2. Version

Sistem operasi Android terdiri dari beberapa versi. Setiap versi Android terbaru memiliki nama-nama unik tersendiri dan memiliki beberapa jenis kelebihan mulai dari tampilan hingga optimasi keamanan. Berikut daftar nama OS Android menurut versi.

a. Android versi 1.0 dirilis tanggal 23 September 2008.
b. Android versi 1.1 dirilis tanggal 09 Februari 2009.
c. Android versi 1.5 Cupcake dirilis tanggal 30 April 2009.
d. Android versi 1.6 Donut dirilis tanggal 15 September 2009.
e. Android versi 2.0 Eclair dirilis tanggal 26 Oktober 2009.
f. Android versi 2.2 Froyo dirilis tanggal 10 Mei 2010.
g. Android versi 2.3 Gingerbread dirilis tanggal 06 Desember 2010.
h. Android versi 3.0 Honeycomb dirilis tanggal 22 Februari 2011.
i. Android versi 4.0 Ice Cream Sandwich tanggal 19 Oktober 2011.
j. Android versi 4.1 Jelly Bean dirilis tanggal 09 Juli 2012.
k. Android versi 4.4 Kitkat dirilis tanggal 31 Oktober 2013.
l. Android versi 5.0 Lolipop dirilis tanggal 17 Oktober 2014.
m. Android versi 6.0 Marshmallow dirilis tanggal 28 Mei 2015.
n. Android versi 7.0 Nougat dirilis tanggal 22 Agustus 2016.
o. Android versi 8.0 Oreo dirilis ttanggal 21 Agustus 2017. [16]

2.2.13 Android Studio

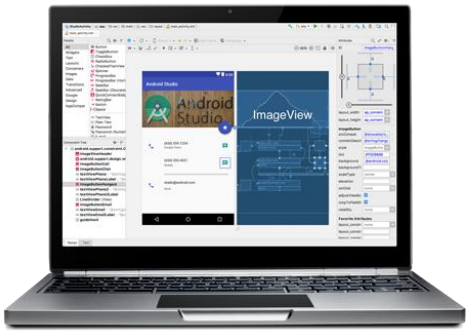
Android studio merupakan sebuah software tools *Integrated Developmenr Enviroment* (IDE) untuk paltform Android. Android studio ini diluncurkan pada tanggal pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk Manajer Google, Ellie Powers. Logo Android Studio dapat dilihat pada Gambar 2.7 Logo Android Stuido



Sumber : https://i.blogs.es/6e0b73/android-studio/450_1000.png
Gambar 2.7 Logo Android Stuido

Android studio ini bersifat *free* dibawah free dibawah Apache License 2.0 Android Studio awalnya dimulai dengan versi 0.1 pada bulan mei 2013, Kemudian dibuat versi beta 0.8 yang dirilis pada bulan julni 2014, yang paling terbaru dirilis

v.3 pada bulan Oktober 2017. Berbasiskan JetBrains' Intellij IDEA, Studio didesain khusus untuk Android Development. Sekarang ini sudah bisa didownload untuk Windows, Mac OS X, dan Linux. Lokasi download bebas dari aplikasi ini di https://developer.Android.com/studio/index.html?hl=id. [16]. Tampilan Android Studio dapat dilihat pada Gambar 2.8 Tampilan Android Studio



Sumber :
https://developer.android.com/studio/images/hero_image_studio.png?hl=id
Gambar 2.8 Tampilan Android Studio

Syarat Hardware:

Windows :

1. Microsoft Windows 7/8/10 (32 atau 64 bit)
2. RAM minimum 3 GB, RAM yang digunakan disarankan 8 GB; ditambah 1 GB untuk Emulator Android
3. Ruang Disk minimum yang tersedia 2 GB. Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)
4. Resolusi layar minumum 1280 x 800

Mac :

1. Mac OS X 10.10 (Yosemite) atau lebih tinggi, hingga 10.12 (mac OS Sierra).
2. RAM minimum 3 GB, RAM yang disarankan 8 GB; dtambah 1 GB untuk Emulator Android
3. Ruang Disk minimum yang tersedia 2 GB. Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)
4. Resolusi layar minumum 1280 x 800

Linux :

1. Desktop GNOME atau KDE. Telah diuji pada Ubuntu 12.04, Pricise Pangolin (distribusi 64-bit dan 32-bit).
2. Distribisi 64-bit yang mampu menjalankan aplikasi 32-bit.
3. GNU C Library (glibc) 2.19 atau yang lebih baru.
4. RAM minimum 3 GB, RAM yang disarankan 8 GB; dtambah 1 GB untuk Emulator Android.
5. Ruang Disk minimum yang tersedia 2 GB. Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator).
6. Resolusi layar minumum 1280 x 800. [16]

2.2.14 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. [16]

2.2.15 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. [16]. Tampilan *emulator* dapat dilihat pada Gambar 2.9 Tampilan Emulator.



Sumber : https://developer.android.com/images/tools/studio/studio-feature-emulator_2x.jpg?hl=id

Gambar 2.9 Tampilan Emulator

2.2.16 Website

Website merupakan sebuah halaman berisi informasi yang gapat 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 sesame 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 bias berupa *personal computer*, laptop, PDA, maupun *cell phone*. [17]

2.2.17 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 [18]

2.2.18 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 sautu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada user.

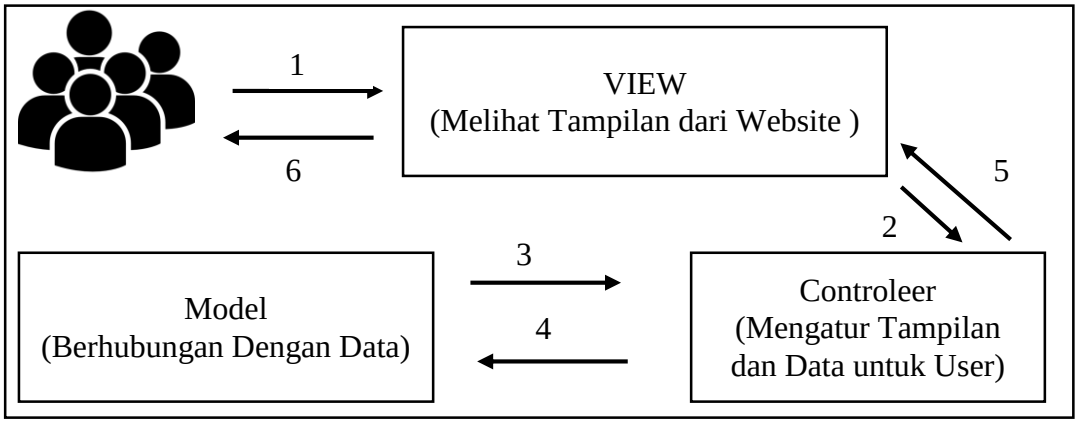
2. *Model*

   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. [18]

Alur MVC dapat dilihat pada Gambar 2.10 Alur MVC.



Gambar 2.10 Alur MVC

2.2.19 CodeIgniter

CodeIgniter adalah sebuah *framework* berbasis PHP yang kuat dengan *footprint* yang sangat kecil, dibangun untuk pengembangan yang membutuhkan *toolkit* sederhana dan elegan untuk membuat aplikasi web dengan fitur lengkap. *CodeIgniter* pertama kali ditulis oleh Rick Elis (http://www.ellislab.com) , seorang musisi rock yang beralih profesi menjadi programmer dalam riset kecil-kecilannya dan menghasilkan suatu *framework* PHP yang berukuran kecil, ringan serta memenuhi fitur umum aplikasi PHP. Namun, sejak tahun 2014 *CodeIgniter* telah dimiliki oleh *British Columbia Institute of Technology* (BICT).

Kelebihan *CodeIgniter:*

1. Berukuran kecil
2. Dokumentasi yang sangat lengkap
3. *CodeIgniter* mampu berjalan baik pada hampir semua platform hosting. *CodeIgniter* juga mendukung database-database paling umum, termasuk MySQL.
4. Kinerja yang cepat
5. Sangat mudah di integrasikan
6. Sedikit konfigurasi
7. Mudah dipelajari

Kekurangan *CodeIgniter*:

1. *CodeIgniter* tidak cocok digunakan pada project website skala besar
2. *Library/plugin* yang sangat terbatas
3. Belum adanya editor khusus *CodeIgniter*
4. Masih banyak kelonggaran dalam hal coding [18]

2.2.20 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 Preprosesor*) 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 komputre 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 refrensi 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 Memiliku kelemahan keamanan. Jad programmer harus jeli dan berhati-hati dalam melakukan pemrograman dan konfigurasi PHP. [19]

2.2.21 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 masih-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. [20]

2.2.22 Object-Oriented Analysis and Desaign (OOAD)

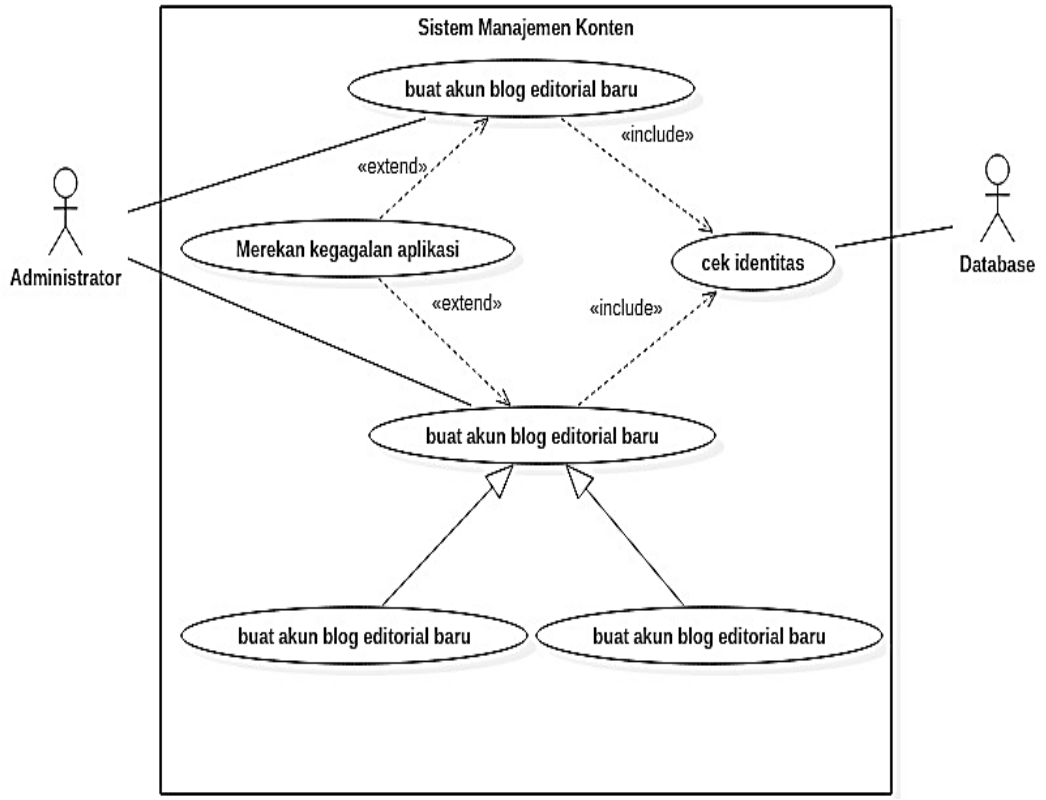
Analisis dan Desain berorientasi objek memiliki model yang berbeda dengan pengembangan perangkat lunak secara tradisional, yang beradasarkan fungsi dan prosedur. Lebih jelasnya, *Object-oriented* adalah cara untuk mengembangkan perangkat lunak dengan membangun objek yang dapat dengan mudah diganti, dirubah 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 [21]. Contoh *use case* dapat dilihat pada Gambar 2.11 Contoh Use Case Diagram.



Sumber : Learning UML. 2.0 [21]

Gambar 2.11 Contoh Use Case Diagram

2. Skenario Use Case

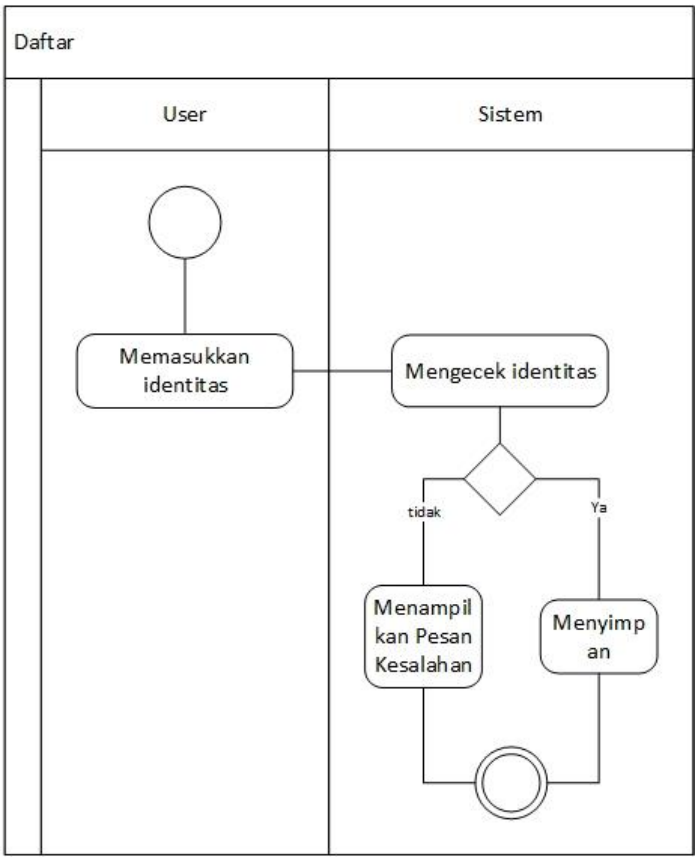
Sebuah diagram yang menunjukkan use case dan aktor mungkin menjadi titik awal yang bagus, tetapi tidak memberikan detail yang cukup untuk desainer sistem untuk benar-benar memahami persis bagaimana sistem dapat terpenuhi. Cara terbaik untuk mengungkapkan informasi penting ini adalah dalam bentuk penggunaan skenario *use case* berbasis teks per *use case*-nya [21]. Dasar format penulisan skenario *use case* dapat di lihat pada Tabel 2.3 Penulisan Skenario Use Case.

Tabel 2.3 Penulisan Skenario Use Case

Use Case Name	Berisi nama *use case*	
Related Requirement	Berisi kode kebutuhan yang terkait dengan skenario	
Goal in Context	Tujuan yang ingin dicapai *use case*	
Precondition	Kondisi sistem ketika akan dilangsungkannya skenario *use case*	
Successful End Condition	Kondisi jika *use case* berhasil terpenuhi	
Failed End Condition	Kondisi jika *use case* gagal terpenuhi	
Primary Actor	Aktor utama yang berasosiasi dengan *use case*	
Secondary Actor	Aktor selain aktor utama yang berasosiasi dengan *use case*	
Trigger	Aktivitas yang dilakukan untuk mengawali *use case*	
Included Cases	*Use case* yang di-*include*	
Main Flow	Step	Action
	1	Langkah-langkah aksi dari aktivitas *use case*
	2	
	3	
Extension	Step	Branching Action
	2.1	Langkah-langkah aksi lain selain aktivitas utama dari use case
	2.2	

3. Diagram Aktivitas

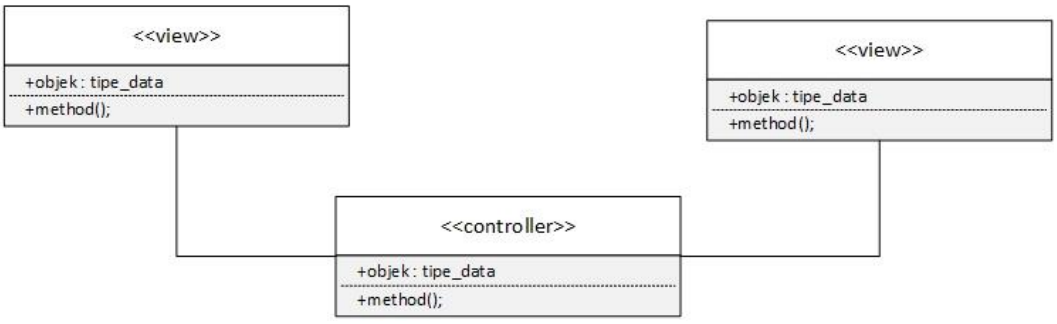
Diagram Aktivitas adalah sebuah tahapan yang lebih focus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Di mana biasanya dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Diagram Aktivitas ini sendiri memiliki struktur yang mirip dengan *flowchart* atau data flow diagram pada perancangan terstruktur. Diagram Aktivitas dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram* [21]. Contoh activity diagram dapat dilihat pada Gambar 2.12 Contoh Diagram Aktivitas [22].



Gambar 2.12 Contoh Diagram Aktivitas

4. *Class Diagram*

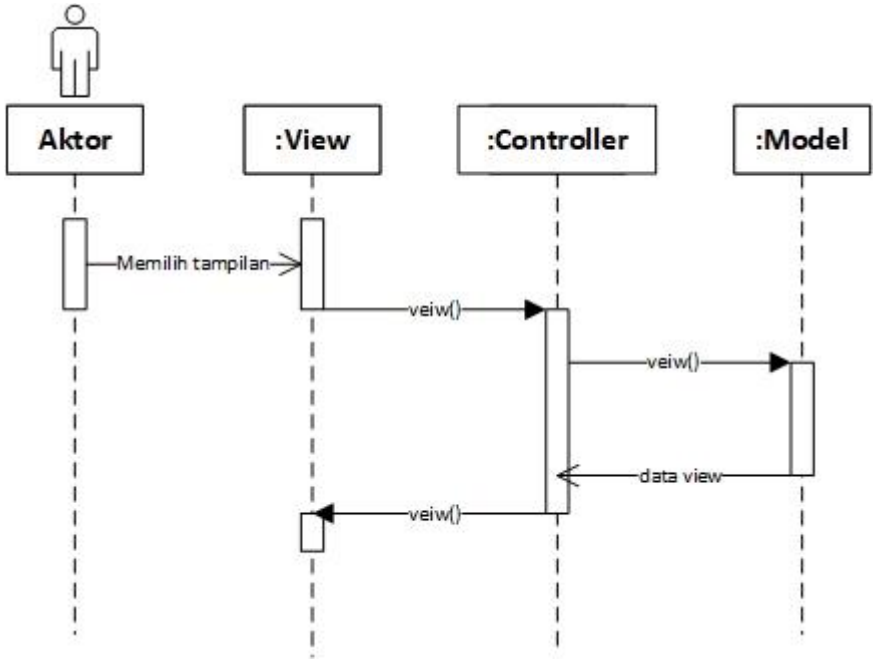
Class diagram adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket dan objek serta hubungan satu sama lain. *Class diagram* juga menjelaskan hubungan antar *class* secara keseluruhan di dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan [21]. Contoh class diagram dapat dilihat pada Gambar 2.13 Contoh Class Diagram [21].



Gambar 2.13 Contoh Class Diagram

5. *Sequence Diagram*

Sequance 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 gari putus dan waktu yang ditunjukkan dengan *progress vertical*. Manfaat dari *sequence* diagram adalah memberikan gambaran detail dari setiap *use case diagram* yang dibuat sebelumnya [21]. Contoh sequence diagram dapat dilihat pada Gambar 2.14 Contoh Sequence Diagram [21].



Gambar 2.14 Contoh Sequence Diagram

2.2.23 Sublime Text

Sublime Text adalah editor teks berfitur untuk mengedit file teks lokal. Ini memiliki banyak fitur built-in untuk membantu dalam mengedit kode, seperti penyorotan sintaks, indentasi otomatis, pengenalan tipe file, file / folder sidebar berguna untuk dengan mudah mengedit beberapa file dalam direktori, macro untuk mengotomatisasi tugas-tugas berulang, dan tab dan opsi *split-window* untuk melihat dan mengedit banyak file secara bersamaan. Dengan banyaknya fitur programmer-

centric *Sublime Text* , memanfaatkan editor ini dapat meningkatkan produktivitas tanpa menghambat seperti *Visual Studio* dan *Eclipse*. IDE ini tentu memiliki tempat mereka, tetapi kesederhanaan *Sublime Teks* dapat lebih disukai dalam skenario lain. Selain banyak fitur built-in yang berguna, Sublime Text dibangun dari dasar agar dapat diperluas dan komunitas *Sublime Teks* telah memperhatikannya. Sudah banyak ekstensi yang berguna termasuk antarmuka untuk sistem kontrol versi seperti git, paket snippet untuk jQuery dan PHP, dan penyorotan sintaks untuk bahasa pembungkus CSS populer seperti LESS dan SASS. Selanjutnya, komunitas telah membuat *Package Manager* untuk membuat penemuan, menginstal, memperbarui, dan mengelola plugin ini semudah mengetik hotkey dan beberapa karakter. [23]