

BAB 2

LANDASAN TEORI

2.1 Transportasi

Transportasi adalah sekelompok fasilitas yang dapat berupa barang maupun jasa yang ditujukan untuk kepentingan publik dan masyarakat dalam pelayanan pemindahan. Transportasi membutuhkan jasa maupun fasilitas dasar dan alat – alat yang dibutuhkan masyarakat sebagai fungsi negara ataupun wilayah dan pekerjaan umum yang dibutuhkan dalam sistem transportasi. Dalam kegiatan transportasi berisi tiga hal yaitu adanya muatan yang dikirim, ketersediaan kendaraan sebagai alat angkut dan adanya jalur yang dilalui. Selain itu terdapat proses bergerak dari titik asal dan ada transportasi asal dan tujuan dimana kegiatan tersebut terhenti [9].

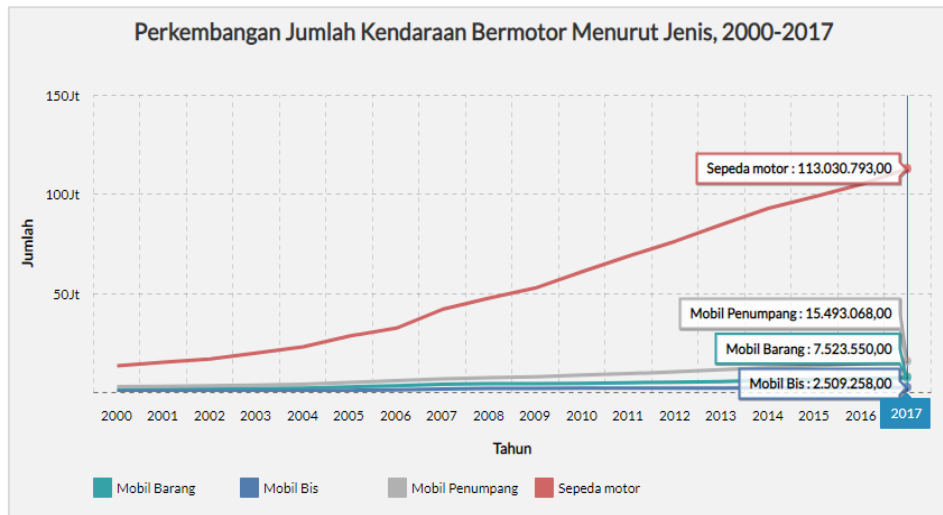
2.1.1 Transportasi Darat

Transportasi darat merupakan transportasi yang digerakan oleh manusia, hewan ataupun mesin yang berjalan di jalur darat. Transportasi ini terdiri dari dua yaitu adalah jalan raya dan rail road. Jalan raya yaitu peralatan transportasi seperti manusia, hewan, mesia, sepeda motor, truk dan lainnya. Sedangkan rail road yaitu peralatan transportasi seperti kereta api yaitu lokomotif, trailer, kereta penumpang dan lainnya [9].

2.1.2 Sepeda Motor Sebagai Transportasi Darat di Indonesia

Penggunaan kendaraan sepeda motor di Indonesia telah menjadi transportasi utama yang paling banyak digunakan. Penggunaan kendaraan sepeda motor ini banyak sekali digunakan dalam membantu para pengendara dalam berpergian. Pengguna sepeda motor ini pun berasal dari berbagai kalangan mulai dari para remaja, orang dewasa hingga pengguna yang sudah lanjut usia. Berdasarkan data dari Badan Pusat Stastitika (BPS) mengenai perkembangan jumlah kendaraan bermotor pada tahun 2000 hingga 2017 menunjukkan bahwa kendaraan sepeda motor terus mengalami peningkatan [2]. Dari data tersebut didapat bahwa jumlah kendaraan sepeda motor pada tahun 2017 menempati urutan pertama dengan jumlah lebih dari 113 juta unit kendaraan [2]. Berdasarkan data tersebut terlihat

bahwa penggunaan sepeda motor sebagai sarana transportasi lebih banyak diminati. Berikut data statistik pengguna sepeda motor di Indonesia disajikan pada gambar 2.1 [10]



Gambar 2.1 Perkembangan Kendaraan Sepeda Motor (2000-2017) [10]

2.1.3 Permasalahan Pengendara Sepeda Motor

Lalu lintas didefinisikan sebagai gerak suatu kendaraan dan orang di jalan lalu lintas. Lalu lintas adalah prasaran yang di tujukan bagi kendaraan, orang maupun barang yang berupa kалан dan fasilitas pendukung [11]. Namun terkadang kelancaran lalu lintas sering terganggu khususnya yang berdampak pada pengendara sepeda motor. Adapun beberapa faktor yang menjadi penyebab terganggunya kelancaran pengendara khususnya sepeda motor dalam berlalu lintas yaitu diantaranya tidak adanya pelebaran jalan, perilaku berkendara, rendahnya disiplin, dan juga hujan [11].

2.1.4 Pengaruh Hujan Bagi Pengendara Motor

Hujan menjadi salah satu faktor yang mempengaruhi terhadap kelancaran lalu lintas. Beberapa hal yang menyebabkan kelancaran lalu lintas bagi kendaraan bermotor terganggu yaitu diantaranya :

- 1) Licinnya jalanan dan genangan air yang dapat menghambat laju kendaraan bermotor sehingga kemacetan mungkin terjadi. Hal ini cukup mengganggu

pengendara bermotor khususnya sepeda motor karena pengendara akan melambatkan laju kendaraannya saat hujan turun untuk menghindari kecelakaan lalu lintas sehingga memungkinkan kemacetan mungkin terjadi di daerah tersebut [11].

- 2) Hujan deras yang berangsur lama akan membuat jalan – jalan tertentu menjadi banjir. Banjir merupakan genangan air yang cukup tinggi yang menutupi jalanan. Dampak banjir bagi pengendara yang mempengaruhi kelancaran lalu lintas yaitu ketika genangan air yang tinggi yang masuk ke dalam saringan udara, knalpot kendaraan bermotor dapat menyebabkan kendaraan mogok. Hal ini dapat berdampak pada penumpukan kendaraan sehingga terjadi kemacetan lalu lintas [3].

2.2 Sistem

Sistem merupakan suatu jaringan kerja dari prosedur-prosedurnya yang saling berhubungan satu sama lain dan berkumpul bersama-sama dalam melakukan suatu kegiatan serta untuk menyelesaikan suatu sasaran tertentu [12].

Model umum dari sebuah sistem yaitu adalah masukan (*input*), proses, dan keluran (*output*). Hal ini merupakan konsep dari sistem yang sangat sederhana karena sebuah sistem memiliki beberapa karakteristik atau sifat – sifat tertentu yang mencirikan bahwa hal tersebut dapat dikatakan sebuah sistem. Adapun beberapa karakteristik yang dimaksud yaitu diantaranya [13]:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen – komponen sistem tersebut dapat berupa subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan

sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem adalah Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber – sumber daya mengalir dari satu subsistem ke subsistem yang lain.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan kedalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan dan masukan sinyal Contoh di dalam suatu unit sistem komputer, “program” adalah maintenance input yang digunakan untuk mengoperasikan komputernya dan “data” adalah signal input untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsitem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi.

7. Pengolah Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan data menjadi keluaran.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan

2.2.1 Sistem Peringatan Dini

Sistem peringatan dini merupakan suatu terminologi yang umum digunakan dalam manajemen resiko bencana. Menurut UNISDR Sistem peringatan dini merupakan serangkaian dari kapasitas yang diperlukan dalam menghasilkan serta menyebarkan informasi peringatan yang bermakna tepat pada waktunya untuk memungkinkan individu, masyarakat maupun organisasi yang terancam ancaman bahaya untuk dapat bersiap atau mengambil tindakan secara tepat dan dalam waktu yang memadai untuk mengurangi kemungkinan kerugian atau kehilangan [14].

Sistem peringatan dini berbasis pada masyarakat merupakan sistem yang berpusat pada masyarakat yang mana terdiri dari empat elemen di antaranya yaitu pengetahuan tentang risiko, pemantauan, analisis dan peramalan ancaman bahaya, komunikasi atau penyebaran pesan siaga peringatan dan juga kemampuan setempat untuk dalam merespon peringatan yang diterima. Menurut De Leon peramalan lokal yang melibatkan langsung partisipasi masyarakat merupakan alternatif yang tepat dalam mengimplementasikan sistem peringatan dini berbasis masyarakat [14].

2.3 Aplikasi

Aplikasi secara istilah dapat diartikan sebagai suatu program yang siap untuk digunakan serta dibangun untuk melaksanakan suatu fungsi bagi penggunanya serta penggunaan aplikasi dapat digunakan berdasarkan sasaran yang dituju. Menurut kamus komputer eksekutif pengertian aplikasi memiliki arti yaitu sebagai pemecahan masalah yang menggunakan suatu teknik pemrosesan data yang umumnya berpacu pada sebuah komputasi yang diharapkan ataupun suatu pemrosesan data yang diinginkan [15].

2.3.1 Aplikasi Mobile

Aplikasi Mobile adalah sebuah aplikasi yang dapat memungkinkan pengguna dalam melakukan mobilitas dengan perangkat seperti PDA, telepon seluler atau Handphone. Dengan aplikasi berbasis mobile dapat mempermudah pengguna dalam melakukan kegiatan maupun aktifitas. Penggunaa perangkat mobile sebagai perangkat multimedia hiburan paling banyak diminati oleh kurang lebih 70%

pengguna *mobile*. Fitur yang digemari sebagai media hiburan meliputi permainan, musik, dan video [16].

2.4 Android

Android merupakan sebuah sistem operasi yang bersifat open source dengan basis Linux. Android biasanya dirancang bagi perangkat seluler dengan layar sentuh seperti smartphone maupun komputer tablet. Pada awalnya android dikembangkan oleh Android, Inc., dengan bantuan finansial dari Google, Kemudian pada tahun 2005 Google membelinya. Pada tahun 2007 sistem operasi ini dirilis secara resmi, bersama dengan dibangunnya perangkat Android pertama [17].

2.4.1 Perkembangan Android

Platform Android awalnya dikembangkan oleh perusahaan bernama Android Inc yang mana merupakan perusahaan baru yang bergerak dibidang perangkat lunak untuk perangkat *mobile*. Pada tahun 2005 perusahaan Android Inc dibeli oleh Google. Pada 5 november 2007 pada perilisan perdana Android, bersama Open Handset Alliance mendukung pengembangan standar terbuka untuk perangkat *mobile*. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android yaitu ARM Holdings, Atheros Communication, Garmin Ltd, Sony Ericsson, Thosiba corp dan Vodafone Group Plc [18].

Sejak rilis pada April 2009, versi Android dikembangkan dengan kode nama yang dinamai berdasarkan makanan pencuci mulut dan makanan manis. Masing-masing versi dirilis sesuai dengan urutan alfabet, yakni *Cupcake* (1.5), *Donut* (1.6), *Eclair* (2.0–2.1), *Froyo* (2.2–2.2.3), *Gingerbread* (2.3– 2.3.7), *Honeycomb* (3.0–3.2.6), *Ice Cream Sandwich* (4.0–4.0.4), *Jelly Bean* (4.1–4.3), *KitKat* (4.4+), *Lollipop* (5.0+), *Marshmallow* (6.0+), *Nougat* (7.0+) , *Oreo* (8.0+) dan *Pie* (9.0+) [19].

2.4.2 Arsitektur Android

Secara garis besar arsitektur Android terdiri dari beberapa lapisan yaitu *Applications and Widgets*, *Applications Framework*, *Libraries*, *Android Run-time*

serta *Linux Kernel* sebagai file sistem. Berikut penjelasan dari lapisan pada arsitektur android yaitu sebagai berikut [18] :

1. *Application and Widgets*

Application dan Widget adalah lapisan saat pengguna berhubungan dengan aplikasi saja, biasanya pengguna mengunduh aplikasi kemudian melakukan instalasi dan menjalankan aplikasi tersebut.

2. *Applications Frameworks*

Applications Frameworks adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem android, karena pada layer inilah aplikasi dapat dirancang dan dibuat

3. *Libraries*

Libraries adalah layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya.

4. *Android Run Time*

Android Run Time merupakan layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux.

5. *Linux Kernel*

Linux Kernel adalah layer dimana inti dari operating system dari Android itu berada. Berisi file-file system yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi android lainnya.

2.4.3 Fundamental Android

Adapun beberapa fundamental atau pembentuk dalam aplikasi android yaitu terdiri dari 4 jenis komponen utama yaitu diantaranya [18] :

1. *Activities*

Activities merupakan komponen yang menyediakan *User Interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi

2. *Service*

Pada fundamental ini tidak memiliki GUI tetapi berjalan secara background contohnya memainkan musik atau mengambil data dari jaringan.

3. *Broadcast Receiver*

Berfungsi untuk menerima dan menyampaikan notifikasi. Sebagai contoh notifikasi penggunaan *baterai*.

4. *Content Provider*

Berfungsi untuk membuat kumpulan aplikasi data secara spesifik sehingga aplikasi dapat digunakan oleh aplikasi lain.

2.5 Pemrograman OOP

Object Oriented Programming (OOP) adalah cara pengembangan suatu aplikasi berdasarkan pada objek-objek yang terdapat di dunia nyata. Objek ini merupakan gabungan antara struktur data dan perilaku dalam satu entitas. Pemrograman berorientasi objek merupakan cara baru dalam berfikir dan berlogika dalam menghadapi permasalahan yang coba diatasi dengan bantuan komputer. Pemrograman berorientasi objek berbeda dengan Pemrograman Terstruktur dengan melihat permasalahan lewat pengamatan dunia nyata dimana setiap objek yaitu entitas tunggal memiliki kombinasi antara struktur data dengan fungsi tertentu, berbeda dengan pemrograman terstruktur dimana struktur data dengan fungsi didefinisikan secara terpisah dan tidak berhubungan satu sama lain [20].

Dalam pengembangan pemrograman berorientasi objek terdapat konsep-konsep dan sifat-sifat yang digunakan. Konsep-konsep tersebut yaitu diantaranya [20] :

1. *Class*, yaitu merupakan konsep yang membungkus (enkapsulasi) data dan abstraksi prosedural yang dibutuhkan dalam menggambarkan isi serta tingkah laku dari entitas. Kelas juga merupakan deskripsi tergeneralisir yang digambarkan sebagai objek yang sama.
2. *Object*, digambarkan sebagai orang, benda, tempat atau lainnya yang ada di dunia nyata yang dibutuhkan oleh aplikasi. Objek memiliki atribut serta metoda.
3. *Attribute*, menggambarkan suatu data yang mampu memberikan informasi kelas maupun objek tempat atribut tersebut berada.
4. *Method*, yaitu merupakan prosedur atau fungsi yang terdapat dalam objek dengan atribut. Metode digunakan dalam pengaksesan terhadap data yang terdapat pada objek tersebut.

5. *Message*, yaitu merupakan suatu alat komunikasi antar objek. Hubungan antar objek tergantung pada problem domain dan tanggung jawab dari sistem.
6. *Event*, yaitu merupakan suatu kejadian pada waktu yang terbatas serta menggambarkan rangsangan dari luar sistem.
7. *State*, abstraksi suatu nilai atribut dan *link* dalam sebuah objek. State adalah tanggapan dari objek terhadap kumpulan *event* masukan.
8. *Scenario*, urutan *event* yang terjadi sepanjang proses eksekusi sistem.

2.5.1 Java

Java merupakan bahasa pemrograman yang mampu dijalankan di berbagai perangkat termasuk perangkat *mobile*. Awalnya bahasa pemrograman ini dibuat oleh James Gosling saat masih bergabung dengan *Sun Microsystems*. Saat ini bahas ini merupakan bagian dari Oracle tahun 1995. Bahasa pemrograman ini banyak mengadopsi sintaksis dari bahasa pemrograman C dan C++ tetapi dengan sintaksis model objek yang lebih. Aplikasi yang dibangun dengan java umumnya dikompilasi ke dalam p-code (bytecode) serta mampu dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum selain itu java secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Java menjadi bahasa pemrograman yang paling populer saat ini dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak [21].

2.6 Unified Modelling Language (UML)

Unified Modeling Language (UML) merupakan salah satu alat bantu yang handal dalam pengembangan sistem yang berorientasi objek. Hal ini dikarenakan dalam UML terdapat pemodelan visual sehingga memungkinkan bagi pengembang sistem dapat membuat cetak biru atas visi mereka dalam bentuk yang mudah dimengerti serta dilengkapi dengan mekanisme yang efektif dalam berbagi dan mengkomunikasikan rancangan satu dengan yang lain [22].

Diagram UML dapat dideskripsikan menjadi beberapa bentuk diagram yaitu diantaranya [22] :

1. *Use Case Diagram*

Use Case diagram merupakan diagram yang menggambarkan kebutuhan sistem dari sudut pandang pengguna serta memperlihatkan hubungan-hubungan yang terjadi antara aktor dengan sistem.

2. *Activity Diagram*

Activity Diagram merupakan suatu cara dalam memodelkan *event-event* yang terjadi dalam diagram use case.

3. *Class Diagram*

Class Diagram merupakan suatu spesifikasi yang jika diinstansiasi dapat menghasilkan suatu objek dan juga merupakan inti dari pengembangan serta desain berorientasi objek. *Class* menggambarkan atribut ataupun properti suatu sistem serta metode dan fungsi. *Class* diagram menggambarkan struktur, deskripsi, *package* dan objek serta hubungannya satu sama lain.

4. *Sequence Diagram*

Sequence diagram memberi gambaran interaksi antara objek di dalam dengan di sekitar sistem berupa pesan yang digambarkan terhadap waktu. *Sequence* diagram terdiri dari dimensi waktu (vertikal) dan dimensi objek terkait (horizontal). *Sequence* diagram dapat digunakan sebagai gambaran skenario ataupun rangkaian langkah-langkah proses sebagai respons dari sebuah event untuk menghasilkan keluaran tertentu.

2.7 Basis Data

Secara umum sistem basis data merupakan sekumpulan tabel berisi data yang saling berhubungan (basis data dalam komputer) dan sekumpulan dari program DBMS (*Data Base Management System*) yang memungkinkan satu atau lebih pengguna dapat mengakses dan memanipulasi data pada tabel-tabel tersebut. Dalam perancangan basis data terdapat suatu proses dalam menciptakan perancangan basis data untuk mendukung operasi serta tujuan perusahaan. Perancangan basis data terbagi menjadi 3 yaitu [23]:

1. Desain basis data konseptual

Desain basis data konseptual merupakan suatu proses membangun model suatu perusahaan didasarkan pada informasi yang terdapat dari perusahaan maupun organisasi tanpa adanya pertimbangan perencanaan fisik.

2. Desain basis data logikal

Desain basis data logikal merupakan proses membangun suatu model informasi pada perusahaan didasarkan pada model data yang spesifik namun tidak tergantung dari *Database management system* (DBMS) khusus dan pertimbangan fisik lain.

3. Desain basis data fisik

Desain basis data fisik merupakan proses pembuatan deskripsi dari implementasi basis data yang terdapat pada penyimpanan. Proses ini mendeskripsikan relasi dasar, berkas dan indeks yang digunakan dalam mencapai efisiensi akses ke dalam data.

2.7.1 MySql

MySQL (*My Structure Query Language*) merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS (*Database Management System*) dengan kemampuan *multithread* dan *multi-user* yang telah mencapai 6 juta instalasi di seluruh dunia. MySQL tersedia pada perangkat lunak *open source* dibawah lisensi GNU GPL (*General Public License*) yaitu MySQL AB, namun mereka juga menjual dibawah lisensi komersial dimana penggunaannya tidak memiliki kecocokan dengan penggunaan GPL. RDBMS (*Relational Database Management System*) didistribusikan secara *open source* dibawah lisensi GPL (*General Public License*). Setiap orang bebas dalam menggunakan MySQL, tetapi tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*) [24].

2.8 JavaScript Object Notation (JSON)

JSON (*JavaScript Object Notation*) merupakan format pertukaran data, yang mudah dibaca, ditulis, serta mudah diterjemahkan oleh manusia dan dibuat oleh komputer. Format ini dibuat berdasarkan dari Bahasa Pemrograman JavaScript.

JSON merupakan format teks yang tidak bergantung kepada bahasa pemrograman apapun karena JSON menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dan lainnya. Oleh karena hal tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data [25].

2.9 *Application Programming Interface (API)*

Application Programming Interface (API) merupakan sebuah teknologi yang dapat memfasilitasi pertukaran informasi atau data antara dua atau beberapa aplikasi perangkat lunak. API merupakan antarmuka virtual antara dua fungsi perangkat lunak yang saling bekerja sama. API mendefinisikan bagaimana cara programmer memanfaatkan fitur tertentu dari sebuah komputer. API tersedia untuk sistem file, sistem windowing, serta sistem jaringan, serta sistem database [26].

2.9.1 Cuaca API

Pemerintah Kota Bandung telah memasang sensor cuaca di 50 kelurahan di Kota Bandung bekerjasama dengan *Cyberindo Aditama (CBN)*. Alat deteksi cuaca tersebut bisa mengetahui informasi terjadinya hujan yang berpotensi banjir sensor yang terpasang berbeda dengan yang dimiliki oleh Badan Meteorologi, Klimatologi dan Geofisika (BMKG). Sensor yang dipasang tersebut langsung ditempatkan di Kelurahan. Sedangkan sensor BMKG melalui satelit. Data informasi yang terkumpul selanjutnya dapat digunakan secara terbuka baik oleh Kelurahan, Badan Penanggulangan Bencana Daerah, dan Badan Meteorologi, Klimatologi dan Geofisika. Data cuaca yang dipasang oleh Pemerintah Kota Bandung ini dapat digunakan oleh siapapun [4]. Berikut langkah – langkah yang diperlukan untuk menggunakan data Cuaca API :

1. Pengguna dapat mengakses data sensor cuaca ini pada alamat berikut
<http://45.126.132.55:4444/>
2. Sebelumnya pengguna dapat mendaftarkan diri dengan mengunjungi
<http://45.126.132.55:4444/signup/>

3. Setelah terdaftar maka langkah selanjutnya adalah melakukan *Generate Token* untuk mendapatkan *access_token* untuk memanggil (*request*) data API. Berikut langkah – langkah yang perlu dilakukan
 - a. Masuk Melalui *Shell* atau untuk pengguna OS Windows dapat menggunakan *command prompt*
 - b. Selanjutnya masukan perintah berikut pada *shell* atau *command prompt*, pastikan *curl* telah terpasang

Tabel 2.1 Generate Token Cuaca API

```
curl -X POST -d
"grant_type=password&username=<username>&password
=<password>" -u"<client_id>:<client_secret>"
http://45.126.132.55:4444/o/token/
```

Pada perintah diatas pengguna harus mengganti *<username>* dan *<password>* dengan *username* dan *password* yang pengguna telah daftarkan sebelumnya. Sedangkan untuk *<client_id>* dengan *<client_secret>* yaitu ubah dengan nilai dibawah ini

Tabel 2.2 Client Data Cuaca API

```
Client id:
HNwI00LifXc2eClGdu00lRrh3rMe3Wwb2ho6sF5s
Client secret:
T2uLnt7xd4Iq4FGJhhk6IoCBbDYEGhaH8DEI5OKxorMYqi44D
FiPozzoLZ1b0slbEqadhH8Frpsr0JhZsFk3STmzTjux8BlvAC
6XHoeio1MJLPhYCnefG3nZORizptgW
```

- c. Setelah itu pengguna akan mendapatkan *access_token* dengan *refresh_token*. Adapun token ini hanya bertahan selama 10 jam saja. Berikut contoh hasil dari eksekusi perintah sebelumnya

Tabel 2.3 Hasil Token Cuaca API

```
{"access_token":
"Y4NGTfhB6lKSFCgSFAlYwCAlJKcH4D", "expires_in":
36000, "refresh_token":
```

```
"BlxUGy7qxYT6av1KQ3DIcKlgUbd6pu", "token_type":
"Bearer", "scope": "write read groups"}
```

4. Setelah itu maka pengguna dapat memanggil *request* untuk mendapatkan data sensor Cuaca. Berikut contoh hasil *response* sensor cuaca API

Tabel 2.4 Response Cuaca API

```
Array
(
  [0] => Array
    (
      [0] => stdClass Object
        (
          [weeklyrainin] => 4.83
          [yearlyrainin] => 30.02
          [baromin] => 27.55
          [windchillf] => 72.1
          [uv] => 0
          [indoortemp] => 80.6
          [rainin] => 0.00
          [windgustmph] => 0.00
          [dailyrainin] => 1.38
          [indoorhumidity] => 69
          [time]=>2019-03-
30T14:57:04.327096842Z
          [ID] => IBANDUNG78
          [windspeedmph] => 0.00
          [dewptf] => 69.4
          [humidity] => 91
          [solarradiation] => 0.00
          [monthlyrainin] => 13.78
          [tempf] => 72.1
          [winddir] => 337
        )
      )
    )
)
```

Untuk penggunaan API lebih detail pengguna dapat mengakses dokumentasi pada <http://45.126.132.55:4444/docs/>. Adapun beberapa alamat *request* yang disediakan oleh Cuaca API dalam mendapatkan informasi cuaca dan sensor, berikut ditunjukkan pada tabel 2.5

Tabel 2.5 Url permintaan data Cuaca API

Url	Deskripsi
http://45.126.132.55:4444/rawdata/	Menampilkan informasi cuaca dengan keseluruhan parameter cuaca yang didapatkan
http://45.126.132.55:4444/recentdata/	Menampilkan informasi cuaca terbaru dari setiap sensor

http://45.126.132.55:4444/dailydata/	Menampilkan informasi cuaca dari setiap sensor dalam kurun waktu satu hari
http://45.126.132.55:4444/weeklydata/	Menampilkan informasi cuaca dari setiap sensor dalam kurun waktu satu minggu
http://45.126.132.55:4444/monthlydata/	Menampilkan informasi cuaca dari setiap sensor dalam kurun waktu satu bulan
http://45.126.132.55:4444/sensor/	Menampilkan informasi daftar sensor berikut id, lokasi serta titik koordinat sensor berada

Selain itu disediakan beberapa parameter yang dapat digunakan untuk melakukan *filter* dalam menampilkan data pada *request* Cuaca API ditunjukkan pada tabel 2.6

Tabel 2.6 Parameter request Cuaca API

Parameter	Request	Deskripsi
ID	?id=IBANDUNG24	Untuk menampilkan data berdasarkan ID
Limit	?limit=10	Untuk membatasi jumlah data yang ditampilkan
Start	?start=2018-07-09T00:00:00	Untuk menampilkan data berdasarkan tanggal awal
End	?end=2018-07-10T12:00:00	Untuk menampilkan data berdasarkan tanggal awal
Asc	?asc=False	Untuk menampilkan data secaraurut baik <i>ascending</i> atau <i>descending</i>
Celcius	?celcius=True	Untuk menampilkan data suhu dengan format <i>celsius</i>
Data Lain	?baromin=True	Untuk menampilkan data lain seperti data "dewptf", "windchillf", "winddir", "windspeedmph", "rainin", "solarradiation", "uv", "baromin" dan lainnya

Adapun daftar lokasi sensor cuaca yang telah terpasang di Wilayah Kota Bandung dapat dilihat pada tabel 2.7

Tabel 2.7 Daftar Lokasi Sensor Cuaca di Wilayah Kota Bandung

No	Id	Lokasi	Latitude	Longitude
1	IBANDUNG4	BOJONGKONENG	-6.885	107.647
2	IBANDUNG7	BABAKAN SILIWANGI	-6.888	107.61
3	IBANDUNG8	BALAI KOTA BANDUNG	-6.911	107.609
4	IBANDUNG10	PASCAL HYPERSQUARE	-6.914	107.597
5	IBANDUNG24	FIBERSTAR	-6.938	107.619
6	IBANDUNG26	GUDANG FIBERSTAR ANTAPANI	-6.929	107.664
7	IBANDUNG29	CIGADUNG	-6.88	107.628
8	IBANDUNG31	KECAMATAN LENGKONG	-6.928	107.623
9	IBANDUNG32	KELURAHAN PALEDANG	-6.928	107.616
10	IBANDUNG33	KELURAHAN TURANGGA	-6.94	107.631
11	IBANDUNG34	KELURAHAN BURANGRANG	-6.929	107.619

No	Id	Lokasi	Latitude	Longitude
12	IBANDUNG35	KELURAHAN CIJAGRA	-6.945	107.627
13	IBANDUNG36	KELURAHAN CIKAWAO	-6.928	107.615
14	IBANDUNG37	KELURAHAN LINGKAR SELATAN	-6.931	107.624
15	IBANDUNG38	KELURAHAN MALABAR	-6.921	107.623
16	IBANDUNG40	KELURAHAN PASIRKALIKI	-6.908	107.602
17	IBANDUNG41	KECAMATAN CICENDO	-6.907	107.595
18	IBANDUNG42	KELURAHAN HUSEIN SASTRANEGARA	-6.907	107.588
19	IBANDUNG43	KELURAHAN PAMOYANAN	-6.904	107.592
20	IBANDUNG44	KELURAHAN PAJAJARAN	-6.9	107.591
21	IBANDUNG45	KELURAHAN ARJUNA	-6.909	107.592
22	IBANDUNG47	KECAMATAN COBLONG	-6.883	107.611
23	IBANDUNG48	KELURAHAN LEBAK SILIWANGI	-6.885	107.611
24	IBANDUNG49	KELURAHAN CIPAGANTI	-6.887	107.603
25	IBANDUNG50	KELURAHAN DAGO	-6.879	107.616
26	IBANDUNG51	KELURAHAN LEBAK GEDE	-6.89	107.617
27	IBANDUNG52	KELURAHAN SEKELOA	-6.883	107.62
28	IBANDUNG53	KELURAHAN SADANG SERANG	-6.892	107.627
29	IBANDUNG60	KELURAHAN CISARANTEN ENDAH	-6.932	107.672
30	IBANDUNG61	KECAMATAN SUKAJADI	-6.893	107.588
31	IBANDUNG62	KELURAHAN PASTEUR	-6.891	107.602
32	IBANDUNG63	KELURAHAN SUKAMISKIN	-6.91	107.677
33	IBANDUNG65	KELURAHAN ANCOL	-6.941	107.614
34	IBANDUNG66	KELURAHAN CIGERELENG	-6.946	107.611
35	IBANDUNG68	KELURAHAN HEGARMAHAH	-6.878	107.603
36	IBANDUNG70	KELURAHAN CITARUM	-6.903	107.616
37	IBANDUNG71	KELURAHAN CIATEUL	-6.936	107.611
38	IBANDUNG72	KELURAHAN CIHAPIT	-6.908	107.62
39	IBANDUNG73	KECAMATAN CIDADAP	-6.874	107.6
40	IBANDUNG75	KECAMATAN ARCAMANIK	-6.918	107.685
41	IBANDUNG76	KECAMATAN REGOL	-6.946	107.612
42	IBANDUNG77	KELURAHAN TAMAN SARI	-6.904	107.608
43	IBANDUNG78	KELURAHAN BALONG GEDE	-6.926	107.607
44	IBANDUNG79	KELURAHAN CISEUREUH	-6.949	107.611
45	IBANDUNG80	KELURAHAN SUKAWARNA	-6.886	107.578
46	IBANDUNG81	KELURAHAN SUKAGALIH	-6.886	107.586

2.9.2 Google Maps API

Google Maps API merupakan sebuah tool atau library yang disediakan oleh Google untuk menampilkan peta digital pada website, penggunaan Google Maps

pada website dapat melalui javascript. Dalam Google Maps terdapat Markers yang dapat digunakan dalam menandakan lokasi pada suatu tempat pada sebuah peta [27]. Beberapa tujuan dari penggunaan Google Maps API digunakan untuk menampilkan lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya [25].

2.9.3 Google Directions API

Google Directions API merupakan sebuah layanan yang dikembangkan oleh *google* yang diperuntukan kepada pengembang untuk mencari jalur rute dari satu tempat ke tempat tujuan. Dengan memasukkan parameter *latitude* dan *longitude* akan didapatkan titik – titik jalur berupa *polyline* yang akan digunakan untuk menggambarkan rute pada *map*. Adapun layanan *Google Direction API* pada penelitian ini digunakan untuk menunjukkan jalur perjalanan pengendara sepeda motor [28].

2.9.4 Accuweather API

Accuweather API merupakan suatu layanan yang memberikan informasi cuaca suatu daerah berdasarkan lokasi. Selain informasi cuaca, *Accuweather* API juga menyediakan suatu informasi prediksi perkiraan cuaca [29]. Adapun pada penelitian ini *accuweather* API digunakan untuk melakukan prediksi cuaca lokasi tujuan pengendara.

2.10 GPS

GPS (*Global Positioning System*) merupakan sistem satelit navigasi serta penentuan posisi yang dikembangkan oleh Amerika Serikat. Sistem ini dirancang untuk memberikan posisi serta kecepatan tiga-dimensi serta informasi tentang waktu, secara berkala di seluruh dunia tanpa bergantung waktu dan cuaca, dan bagi beberapa orang secara simultan. GPS saat ini telah banyak digunakan di seluruh dunia dalam berbagai bidang aplikasi yang membutuhkan informasi tentang posisi, kecepatan, percepatan ataupun waktu yang teliti. GPS dapat menentukan posisi dengan ketelitian beragam dari beberapa millimeter hingga puluhan.

Pada umumnya GPS terdiri atas tiga bagian utama, yaitu segmen angkasa (*space segment*) terdiri dari satelit-satelit GPS, segmen sistem kontrol (*control system segment*) terdiri dari stasiun-stasiun pemonitor serta pengontrol satelit, dan segmen pemakai (*user segment*) terdiri dari pengguna GPS termasuk alat-alat penerima dan pengolah sinyal data GPS. Dasar dari konsep penentuan posisi dengan GPS yaitu dengan jarak, dengan pengukuran jarak secara simultan ke beberapa satelit GPS yang koordinatnya. Penentuan posisi pada GPS dapat dilakukan dengan berbagai macam metode yang masing - masing memiliki karakteristik tersendiri [30].

2.11 *Firebase Cloud Messaging*

Firebase Cloud Messaging (FCM) adalah teknologi BaaS (*back-end as a service*) berbasis *cloud* dari Google yang menyediakan beberapa layanan yang salah satunya adalah layana fitur notifikasi (*push notification*) bernama FCM (*firebase cloud messaging*). FCM merupakan perkembangan dari GCM (*Google cloud messaging*) yang menyediakan layanan sehingga pengembang dapat mengirimkan pesan notifikasi ke semua perangkat android yang terhubung dari *server* [31]. Adapun dalam penelitian ini FCM digunakan untuk mengirimkan notifikasi pesan peringatan kepada pengendara melalui *push notification*.

2.12 *Algoritma C4.5*

Algoritma C4.5 merupakan ekstensi dari algoritma ID3 dalam menghasilkan pohon keputusan, algoritma C4.5 rekursif mengunjungi setiap node keputusan, memilih split optimal sampai tidak ada perpecahan lanjut yang memungkinkan. Pada dasarnya algoritma C4.5 mengubah data menjadi pohon keputusan dan aturanaturan keputusan (*rule*) [6]. Berikut beberapa tahapan dalam membangun sebuah pohon keputusan dengan algoritma C4.5 yaitu [6] :

1. Menyiapkan data *training*. Data tersebut telah dikelompokkan ke dalam kelas - kelas tertentu.
2. Menghitung nilai gain dari setiap atribut. Nilai gain tertinggi akan dijadikan sebagai akar.

3. Perhitungan gain didapat dari nilai *entropy*. Berikut persamaan dalam mencari nilai *entropy* persamaan 1

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \dots\dots\dots(1)$$

Keterangan :

S : himpunan kasus

A : fitur

n : jumlah partisi (S)

pi : proporsi dari S_i terhadap (S)

4. Selanjutnya dapat dihitung nilai gain dengan persamaan 2 berikut

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \dots\dots\dots(2)$$

Keterangan :

S : himpunan kasus

A : atribut

n : jumlah partisi atribut (A)

|S_i| : jumlah kasus pada partisi ke-i

|S| : jumlah kasus dalam (S)

Adapun pada penelitian ini digunakan algoritma C4.5 dalam mengkalifikasikan keadaan cuaca data sensor Cuaca API. Pada proses pelatihan digunakan dataset cuaca wilayah Kota Bandung yang dimiliki oleh *World Weather Online* pada bulan November 2018.

2.13 Haversine Formula

Metode Haversine merupakan suatu cara yang digunakan dalam menghitung jarak antara titik permukaan bumi dengan parameter garis lintang (*longitude*) serta garis bujur (*latitude*) sebagai nilai masukan. *Haversine formula* merupakan sebuah persamaan penting pada navigasi yang dapat memberikan informasi jarak lingkaran besar antara dua titik pada permukaan bumi ataupun benda bulat berdasarkan lintang dan bujur. Dengan asumsi bahwa bumi berbentuk lingkaran ataupun bulat sempurna dengan ukuran jari-jari (*R*) 6.367,45 km (kilometer) dan lokasi dari dua

titik koordinat bola (lintang dan bujur) masing - masing dengan lon1, lat1 dan lon2, lat2 sehingga rumus dapat dituliskan dengan persamaan 3 [32]:

$$\begin{aligned}
 x &= (lon2 - lon1) * \cos\left(\frac{lat1 + lat2}{2}\right) \\
 y &= lat2 - lat1 \\
 d &= \text{sqrt}(x * x + y * y) * R \dots\dots\dots(3)
 \end{aligned}$$

Keterangan:

x = Longitide (Lintang)

y = Lattitide (Bujur)

d = Jarak

R = Radius Bumi = 6371 km

1 derajat = 0,0174532925

2.14 Metode Pengujian Aplikasi

Pengujian perangkat lunak aplikasi diperlukan untuk memastikan perangkat lunak yang dibangun dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Pengembang maupun penguji *software* perlu menguji program yang sudah dibangun agar kesalahan maupun kekurangan dari aplikasi dapat dideteksi dan diperbaiki. Pengujian merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisahkan dari siklus hidup pengembangan software seperti halnya analisis, desain, dan pengkodean [33].

2.14.1 Pengujian Black Box

Pengujian *Black Box* berfokus pada spesifikasi fungsional dari sistem perangkat lunak. Pengujian dapat mendefinisikan kumpulan kondisi masukan (input) dan melakukan pengetesan pada spesifikasi fungsional program. Pada pengujian *Black Box* cenderung untuk menemukan hal-hal berikut yaitu [33] :

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).

5. Kesalahan inisialisasi dan terminasi.

2.14.2 Pengujian Beta

Pada penelitian ini pengujian beta dilakukan dengan memberikan kuesioner pada responden. Untuk perhitungan hasil dari kuesioner digunakan perhitungan Skala Likert yaitu teknik skala non-komparatif dan undimensional (hanya mengukur sifat tunggal) secara alami. Responden diminta untuk menunjukkan tingkat kesepakatan melalui pernyataan yang diberikan dengan skala ordinal. Dari kedua pengertian diatas, disimpulkan bahwa skala Likert adalah metode perhitungan kuesioner yang dibagikan kepada responden untuk mengetahui skala sikap suatu obyek tertentu [34].

2.14.3 Skala Likert

Skala Likert digunakan untuk mengukur sikap, pendapat dan persepsi sekelompok orang tentang fenomena sosial. Fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabel penelitian. Dengan skala likert maka variabel yang akan dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan titik tolak dalam menyusun item – item instrumen berupa pernyataan atau pertanyaan [35].

Jawaban setiap item instrumen menggunakan skala likert mulai dari sangat positif hingga sangat negatif seperti berikut [35] :

- a. Sangat Setuju
- b. Setuju
- c. Ragu – Ragu
- d. Tidak Setuju
- e. Sangat Tidak Setuju

