BAB 1
LANDASAN TEORI

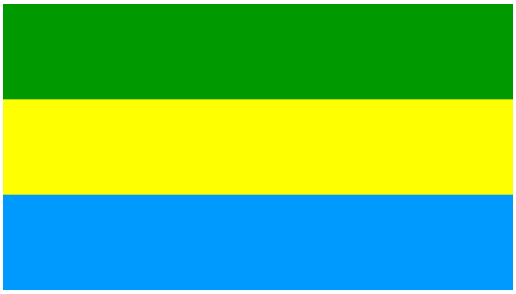
2.1 Kota Bandung

Kota Bandung adalah kota metropolitan terbesar di Provinsi Jawa Barat, sekaligus menjadi ibu kota provinsi tersebut. Kota ini terletak 140 km sebelah tenggara Jakarta, dan merupakan kota terbesar di wilayah Pulau Jawa bagian selatan. Sedangkan wilayah Bandung Raya (Wilayah Metropolitan Bandung) merupakan metropolitan terbesar ketiga di Indonesia setelah Jabodetabek dan Gerbangkertosusila.

Di kota ini tercatat berbagai sejarah penting, di antaranya sebagai tempat berdirinya sebuah perguruan tinggi teknik pertama di Indonesia (Technische Hoogeschool te Bandoeng - TH Bandung, sekarang Institut Teknologi Bandung - ITB)[4], lokasi ajang pertempuran pada masa kemerdekaan, serta pernah menjadi tempat berlangsungnya Konferensi Asia-Afrika 1955, suatu pertemuan yang menyuarakan semangat anti kolonialisme, bahkan Perdana Menteri India Jawaharlal Nehru dalam pidatonya mengatakan bahwa Bandung adalah ibu kotanya Asia-Afrika. Kota kembang merupakan sebutan lain untuk kota ini, karena pada zaman dulu kota ini dinilai sangat cantik dengan banyaknya pohon dan bunga-bunga yang tumbuh di sana. Selain itu Bandung dahulunya disebut juga dengan Parijs van Java karena keindahannya.

Kota Bandung juga dikenal sebagai kota belanja, dengan mall dan factory outlet yang banyak tersebar di kota ini, dan saat ini berangsur-angsur kota Bandung juga menjadi kota wisata kuliner. Dan pada tahun 2007, konsorsium beberapa LSM internasional menjadikan kota Bandung sebagai pilot project kota terkreatif se-Asia Timur Saat ini kota Bandung merupakan salah satu kota tujuan utama pariwisata dan pendidikan. Adapun bendera Bandung memiliki tiga strip horizontal setara hijau, kuning dan biru. Hijau mewakiliki kesejukan dan kemakmuran, kuning

mewakili kesejahteraan rakyat, dan biru mewakili kesetiaan [10]. Dibawah ini merupakan penampakan bendera Bandung yang dapat dilihat pada gambar 2.1



Gambar 1.1 Bendera Bandung

(https://id.wikipedia.org/wiki/Kota_Bandung)

2.1.1 Geografis

Kota Bandung dikelilingi oleh pegunungan, sehingga bentuk morfologi wilayahnya bagaikan sebuah mangkuk raksasa, secara geografis kota ini terletak di tengah-tengah provinsi Jawa Barat, serta berada pada ketinggian ±768 m di atas permukaan laut, dengan titik tertinggi di berada di sebelah utara dengan ketinggian 1.050 meter di atas permukaan laut dan sebelah selatan merupakan kawasan rendah dengan ketinggian 675 meter di atas permukaan laut.

Kota Bandung dialiri dua sungai utama, yaitu Sungai Cikapundung dan Sungai Citarum beserta anak-anak sungainya yang pada umumnya mengalir ke arah selatan dan bertemu di Sungai Citarum. Dengan kondisi yang demikian, Bandung selatan sangat rentan terhadap masalah banjir terutama pada musim hujan. Semetara iklim kota Bandung dipengaruhi oleh iklim pegunungan yang lembap dan sejuk, dengan suhu rata-rata 23.5 °C, curah hujan rata-rata 200.4 mm dan jumlah hari hujan rata-rata 21.3 hari per bulan.

2.1.2 Baleendah

Baleendah merupakan sebuah kecamatan di Kabupaten Bandung, Jawa Barat, Indonesia. Pada tahun 1970 sampai 1980-an, 90% wilayah Baleendah adalah daerah pertanian (persawahan). Namun, pada tahun 1980-an Kecamatan Baleendah direncanakan menjadi ibu kota Kabupaten Bandung. Maka dibangunlah

sarana/prasarana di wilayah itu termasuk gedung DPRD yang sangat megah. Perumahan umum, tempat ibadah dan sekolah pun dibangun di sana sehingga mengubah sebagian besar wilayah pertanian menjadi gedung dan bangunan lainnya. Namun, banjir besar yang melanda Baleendah sekitar tahun 1986 membuat para petinggi memindahkan ibu kota Kabupaten Bandung ke Soreang.

Baleendah merupakan daerah yang sering menjadi langganan banjir di Jawa Barat. Sejak puluhan tahun lalu, sudah sering terjadi banjir besar di daerah ini. Pada saat banjir, warga setempat mengungsi ke kecamatan tetangga, Dayeuhkolot, walaupun sebagian wilayah Dayeuhkolot sendiri sebenarnya juga sering mendapat sedikit terjangan banjir dari Baleendah. Kajian karakter Daerah Aliran Sungai (DAS) Citarum (2011) mendapatkan sekitar 1.651,5 ha (39%) wilayah Baleendah berpotensi terkena banjir setiap tahun. Baleendah mempunyai suhu udara yang sejuk dengan rentang suhu 14 °C (malam terendah) sampai 28 °C (siang tertinggi)

2.1.3 Dayeuhkolot

Dayeuhkolot merupakan sebuah kecamatan di Kabupaten Bandung, Jawa Barat. Dulu nama Dayeuhkolot adalah Karapyak, yang artinya rakit penyeberangan yang dibuat dari batang-batang bambu. Sampai tahun 1810 Karapyak adalah tempat kedudukan para bupati Bandung. Bupati Bandung saat itu adalah R.A. Wiranatakusumah II (masa jabatan 1794-1829). Gubernur Jenderal Hindia Belanda Daendels memerintahkan pemindahan pendopo kabupaten dari Karapyak ke tepi sungai Cikapundung, dengan alasan daerah berprospek lebih untuk dikembangkan.

Dayeuhkolot merupakan daerah rawan banjir di Bandung. Bersama dengan Baleendah, kecamatan ini menjadi daerah siaga bila musim penghujan tiba. Kajian karakter DAS Citarum (2011) mendapatkan bahwa 94% (sekitar 879,8 ha) wilayah Dayeuhkolot berpotensi terkena banjir setiap tahun. Wilayah ini termasuk DAS Citarum bagian hulu. Karena letak geografis Dayeuhkolot dan Baleendah yang berdekatan (bahkan berbatasan) dengan Kota Bandung, maka dapat dipastikan jalur transportasi dari/ke Kota Bandung yang padat pun terputus selama banjir dan melumpuhkan kegiatan ekonomi masyarakat. Hal inilah yang menjadi masalah bersama pemerintah Kabupaten dan Kota Bandung.

2.2 Sistem

Sistem merupakan suatu jaringan kerja dari prosedurnya yang saling berhubungan satu sama lain dan berkumpul bersama-sama dalam melakukan suatu kegiatan untuk menyelesaikan suatu sasaran tertentu [9]. Model umum dari sebuah sistem yaitu adalah masukan (*input*), proses, dan keluaran (*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 [11]:

1. Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen – komponen sistem tersebut dapat berupa subsistem
2. Batasan Sistem

   Ruang lingkup sistem merupakan daerah yang membatasi atara sistem dangan 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

   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

   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

   Energi yang dimasukan 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

mengoprasikan komputernya dan “data” adalah signal input untuk diolah menjadi informasi.

6. Keluaran Sistem

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

7. Pengolah Sistem

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

8. Sasaran Sistem

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

2.1.1 Sistem Pringatan 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.

Sistem peringatan dini berbasis pada masyarakat merupakan sistem yang berpusat pada masyarakat yang mana terdiri dari empat elemen diataranya 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 [12].

2.2 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 dinginkan. Menurut Kamus Besar Bahasa Indonesia, "Aplikasi adalah suatu penerapan dari rancangan sistem dalam mengolah data yang menerapkan aturan atupun ketentuan suatu bahasa pemrograman" [13].

2.3 Android

Android merupakan sistem operasi pada smartphone berbasis Linux. Perbedaan dari sistem operasi android dengan OS Symbian, iOS, Microsoft Windows Mobile dan lain sebagainya yaitu Android bersifat open source sebagai sistem operasi mobile dan sudah mendapat dukungan API (*Application Programming Interface*) yang dimanfaatkan secara penuh dengan biaya yang relatif murah [14].

Menurut Fachrul K & Gianto W dalam bukunya berjudul "Cepat Menguasai Pemrograman Android" bahwa, Android adalah salah satu sistem operasi yang pada awalnya kemudian berkembang menjadi bahasa pemrograman yang banyak dicari dan digunakan oleh programmer. Dasarnya Android merupakan sistem operasi berbasis Linux. Penggunaan sistem Android pada awalnya digunakan untuk melengkapi sistem operasi pada *gadget-gadget* seluler seperti smartphone. Perusahaan android dibawah naungan Android.Inc, adalah pengembang pertama kali sistem operasi ini. Android.Inc, pertama kali berdiri dikota Alto, salah satu kota terkenal di California, Amerika Serikat, tepatnya pada bulan Oktober tahun 2003. Kemudian sejak tahun 2005, Google membeli dan melanjutkan pengembangan menggunakan sumber daya yang dimiliki Google sendiri sehingga mudah digunakan [15].

2.3.1 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*.

Sejak rilis pada April 2009, Android dikembangkan dengan kode nama yang diambil dari makanan kudapan. Masing-masing versi dirilis sesuai urutan alfabet, versi-versi tersebut dapat dilihat pada tabel 2.1

Tabel 1.1 Perkembangan Operating System Android

Versi	Codename	Keterangan
1.0	Astro	Astro merupakan OS Android versi pertama dan dirilis pada 23 September 2008. Namun karena alasan trademark nama OS ini tidak dirilis pada versi pertama Android.
1.1	Bender	Bender merupakan versi Android kedua dan dirilis pada 9 Februari 2009. Versi ini merupakan update untuk memperbaiki beberapa *bugs*, mengganti API, dan menambahkan beberapa fitur.
1.5	Cupcake	Pada versi ini penamaan Android mulai menggunakan makanan kudapan sebagai nama di setiap versinya. Cupcake dijadikan sebagai codename pada versi ini karena merupakan versi ketiga OS dan huruf alphabet ketiga adalah 'C'. Banyak fitur yang ditambahkan pada versi ini seperti kemampuan untuk merekam dan menonton video, animasi layar, perangkat *bluetooth*, dan sebagainya.
1.6	Donut	Dirilis pada 15 September 2009. Pada versi ini UI lebih user friendly dan terdapat fitur *search box* yang ada pada home screen.
2.1	Éclair	Dirilis pada 3 Desember 2009, pada versi Éclair untuk pertama kalinya membawa fitur *Google Map* untuk mempermudah dalam bepergian. Perubahan UI dengan browser baru dan dukungan HTML5.
2.2	Froyo	Froyo merupakan singkatan dari *Frozen Yoghurt*, dirilis pada 20 Mei 2010. Sudah mendukung *Adobe Flash 10.1*, dan kecepatan kinerja aplikasi 2-5 kali lebih cepat.
2.3	Gingerbread	Dirilis pada 6 Desember 2010, versi Gingerbread mendapatkan desain ulang pada bagian UI (*User Interface*) serta teknologi *Near Field Communication* (NFC).
3.0	Honeycomb	OS versi ini didesain khusus untuk mengoptimalkan penggunaan pada tablet PC. Honeycomb mendukung

		multi prosesor dan akselerasi *hardware* untuk grafis. Dirilis pada 22 Februari 2011.
4.0	Ice Cream Sandwich	Versi Honeycomb menambahkan fitur *Face Unlocking* pada perangkat mobile untuk membuka handphone lebih cepat dan efisien. Dirilis pada 19 Oktober 2011 dan disertai beberapa update tambahan lainnya.
4.1	Jelly Bean	*Google Now* merupakan teknologi yang dimasukan pada update versi ini sebagai asisten digital bagi user, beberapa fitur baru pada kamera seperti panorama, daydream, dan screensaver. Dirilis pada tahun 2012.
4.4	KitKat	Pada awalnya diberi nama "Key Lime Pie" (KLP), nama itu dirubah menjadi KitKat karena sangat sedikit orang yang tau makanan bernama Key Lime Pie tersebut. Sistem operasi android versi ini sudah memiliki spesifikasi minimum dengan 512MB RAM sebagai standar penggunaan OS tersebut.
5.0	Lollipop	Diresmikan pada 25 Juni 2014 saat Google I/O, salah satu perubahan yang menonjol dalam rilis Lollipop adalah UI yang di desain ulang dan dibangun dengan bahasa desain. Perubahan lain termasuk notifikasi yang bisa dilihat melalui lock screen dan ditampilkan pada banner di bagian atas layar.
6.0	Marshmallow	Tahun 2015 versi Marshmallow dirilis membawa banyak fitur terbaru diantaranya sensor sidik jari untuk membuka kunci layar, dukungan USB type C, dan fitur Multi-Window untuk multitasking yang lebih nyaman.
7.0	Nougat	Nougat merupakan versi keluaran 2016 adalah salah satu upgrade terbesar untuk OS Android. Versi ini membawa peningkatan performa dan antarmuka yang lebih intuitif serta fitur Multi-Window yang telah disempurnakan.
8.0	Oreo	Android Oreo dirilis pada 2017, pengguna kini bisa mengatur notifikasi apa saja yang akan ditampilkan dan UI dirombak sehingga lebih *user-friendly* bagi pengguna ketika mengakses aplikasi dan mencari informasi.
9.0	Pie	Diresmikan pada tahun 2018, versi ini sudah didukung kemampuan kecerdasan buatan (AI). Handphone Android semakin pintar karena bisa mempelajari pola penggunaan secara otomatis.
10.0	Android 10	Pada tahun 2019 Android merubah penamaannya mengikuti urutan numerik mulai dari angka 10. Fitur terbaru pada versi ini adalah Dark Mode yang diklaim bisa meningkatkan performa baterai agar tahan lebih lama.

2.3.2 Arsitektur Android

Arsitektur Android terdiri dari beberapa lapisan yang memiliki fungsi masing-masing sebagai file sistem. Berikut penjelasan dari lapisan pada arsitektur yaitu sebagai berikut :

1. Application and Widgets

   Merupakan lapisan saat pengguna berhubungan dengan aplikasi saja, biasanya pengguna mengunduh aplikasi kemudian melakukan instalasi dan menjalankan aplikasi tersebut.
2. Application Frameworks

   Merupakan layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan yang akan dijalankan di sistem android.
3. Libraries

   Merupakan layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya.
4. Android Run Time

   Merupakan layer untuk membuat aplikasi Android dapa dijalankan dimana dalam prosesnya menggunakan implementasi Linux.
5. Linux Kernel

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

2.3.3 Android SDK (Software Development Kit)

Android SDK adalah tools API yang diperlukan untuk memulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *niddkeware*, dan *key application* yang dirilis oleh Google. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java.

2.3.4 Pertumbuhan Pengguna Android di Indonesia

Dalam perkembangan *Mobile Phone* di Indonesia telah menjadi sarana utama bagi masyarakat ini disebabkan karena beberapa alasan yaitu diantaranya adalah penggunaan *smartphone* lebih mudah, *portable*, murah, dan irit dibandingkan smartphone generasi-generasi sebelumnya. Smartphone juga digunakan sebagai media komunikasi oleh masyarakat Indonesia.

2.4 Java

Java merupakan bahasa pemrograman yang mampu dijalankan di berbagai perangkat termasuk perangkat *mobile*. Awalnya bahasa pemograman ini dibuat oleh James Gosling saat masih bergabung dengan *Sun Microsystems*. Saat ini bahas ini merupakan bagian dari Oracle tahun 1995. Bahasa pemograman ini banyak mengadopsi sintaksis dari bahasa pemograman C dan C++ tetapi dengan sintaksis model objek yang lebih. 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 [16].

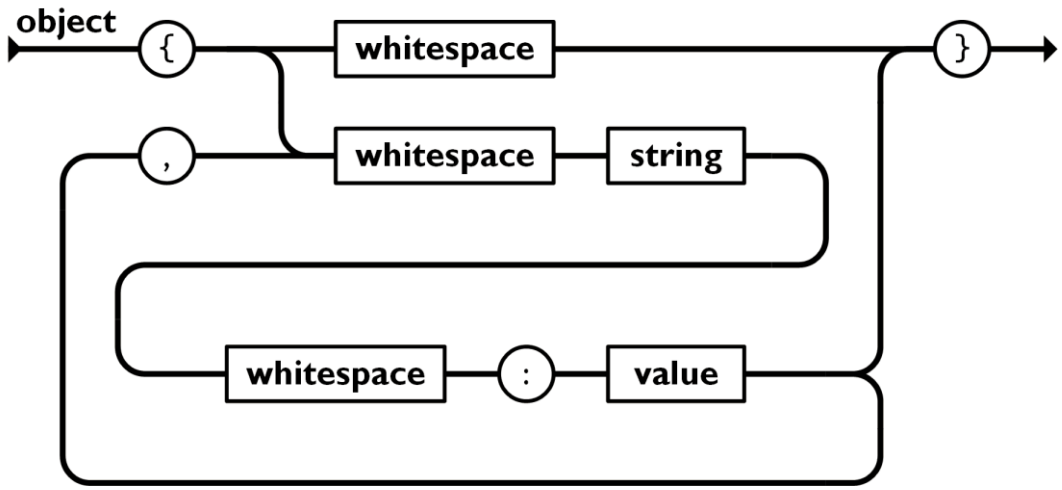
2.4.1 JavaScript Object Notation (JSON)

JSON (*JavaScript Object Notation*) merupakan format pertukaran data yang berukuran ringan dan mudah untuk dibaca dan ditulis. Selain itu JSON cukup mudah diterjemahkan kedalam bahasa mesin. Format teks yang independen, namun menggunakan konvensi yang dikenal bagi programer yaitu termasuk dalam keluarga Bahasa C, seperti C, C++, C#, Java, JavaScript, dan Python. Dikarenakan sifat tersebut membuat JSON sebagai salah satu bahasa yang dipakai untuk pertukaran data yang ideal. JSON mempunyai dua struktur utama, yaitu adalah sebagai berikut [17]:

1. Kumpulan dari pasangan nama atau nilai. Hal ini dinyatakan sebagai objek (*object*), struktur (*structure*), rekaman (*record*), tabel hash (*hash table*), kamus (*dictionary*), daftar berkunci (*keyed list*), atau *associative array* pada beberapa bahasa.

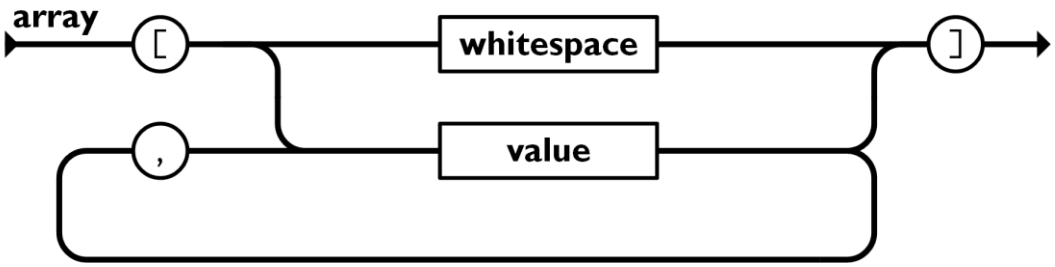
2. Daftar nilai terurutkan (*an ordered list of values*). Hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), serta urutan (*sequence*) pada beberapa bahasa.

Nilai dari struktur data ini merupakan struktur data universal. Pada dasarnya bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini layak disebut demikian karena format data yang mudah dipertukarkan dengan bahasa pemrograman yang juga berdasarkan pada struktur data ini [17]. Berikut merupakan gambar dari JSON Object dan JSON Array ditampilkan pada Gambar 2.2 dan Gambar 2.3



Gambar 1.2 JSON Object

(*https://www.json.org/json-en.html*)



Gambar 1.3 JSON Array

(*https://www.json.org/json-en.html*)

Menurut Peng, Cao, dan Xu (2011), penggunaan *JSON Parsing* dalam *HTTP Connection* bertujuan memberi kemudahan bagi user dalam menggunakan aplikasi mobile. Informasi yang ada di dalam website dapat ditampilkan di dalam sebuah aplikasi mobile dikarenakan *JSON Parsing* akan meringkas konten pada aplikasi sehingga tidak perlu ditampilkan secara keseluruhan seperti halnya *web browser*. Dalam *JSON Parsing*, dibutuhkan sebuah API (*Application Program Interface*) yang berfungsi untuk menghubungkan antara aplikasi mobile dan aplikasi website.

2.5 Android Studio

Android Studio merupakan *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Berikut merupakan lambang dari Android Studio dapat dilihat pada gambar 2.4

Gambar 1.4 Android Studio

(https://commons.wikimedia.org/wiki/Category:Android_Studio)

Selain merupakan *code editor* IntelliJ dan alat pengembang yang berguna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas dalam pembuatan aplikasi Android, contohnya :

1. Sistem bersi berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya akan fitur
3. Lingkungan terpadu developer bias mengembangkan aplikasi untuk semua perangkat Android
4. *Apply Change* untuk melakukan *push* pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi

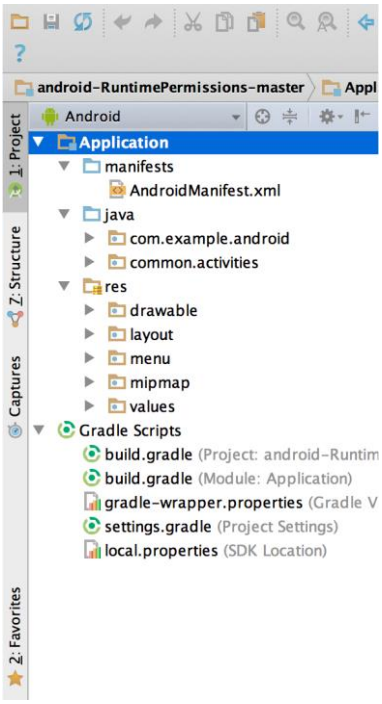
5. Template kode dan integrase GitHub untuk membantu developer membuat fitur aplikasi umum dan mengimpor kode sampel
6. Framework dan fitur pengujian yang lengkap
7. Fitur lint untuk merekam performa, kegunaan, kompabilitas versi, dan masalah lainnya
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrase *Google Cloud Messagin* dan *App Engine*

2.5.1 Struktur Project

Setiap project di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file resource. Jenis modul meliputi:

1. Modul aplikasi Android
2. Modul library
3. Modul Google App Engine

Secara default, Android Studio menampilkan file project dalam tampilan project Android, seperti yang ditampilkan pada gambar 2.5



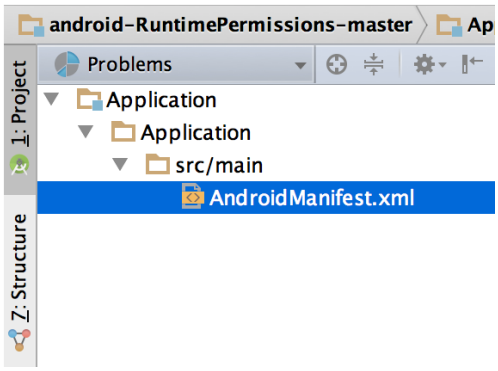
Gambar 1.5 File Project di Android Studio

(https://developer.android.com/studio/intro?hl=id)

Tampilan ini disusun menurut modul untuk memberikan akses cepat dengan tampilan yang simpel ke file sumber utama project. Semua file build terlihat di teratas di bagian *Gradle Script* dan setiap modul aplikasi berisi folder berikut:

1. Manifes: Berisi file AndroidManifest.xml
2. Java: Berisi file kode sumber Java, termasuk kode pengujian Junit
3. Res: Berisi semua resource non-kode, seperti kata letak XML, string UI, dan gambar bitmap

Developer juga dapat menyesuaikan tampilan file project untuk berfokus pada aspek spesifik dari pengembangan aplikasi developer. Misalnya, memilih tampilan *Problems* pada project akan menampilkan link ke file sumber yang berisi error coding dan sintaks yang dikenali, seperti tag penutup elemen XML yang tidak ada dalam lokasi file. Contoh file project dapat dilihat pada gambar 2.6



Gambar 1.6 File Project pada tampilan Problems

(https://developer.android.com/studio/intro?hl=id**)**

2.6 Flutter

Flutter adalah SDK untuk pengembangan aplikasi mobile yang dikembangkan oleh Google. Sama seperti *react native*, *framework* ini dapat digunakan untuk membuat atau mengembangkan aplikasi mobile yang dapat berjalan pada device iOS dan Android. Dibuat menggunakan Bahasa C, C++, Dart dan Skia. Pada *framework* ini semua kodenya di *compile* dalam kode native tanpa ada *intrepeter* pada prosesnya sehingga proses *compile*-nya menjadi lebih cepat. Dari segi penulisan kodenya, Flutter sangat berbeda dari *react native* dan lebih mendekati Java Android. Berikut merupakan lambang dari Flutter dapat dilihat pada gambar 2.7

Gambar 1.7 Flutter

(https://flutter.dev/)

Versi pertama Flutter dikenal sebagai “Sky” dan berjalan pada Android OS. Diresmikan pada acara *Dart Developer Summit* tahun 2015, dengan tujuan untuk mampu merender grafis secara konsisten pada 120 bingkai per detik. Komponen utama Flutter termasuk:

1. Flutter Engine

   Flutter Engine, ditulis terutama dengan bahasa pemrograman C++, memberikan dukungan rendering tingkat rendah menggunakan *library* grafik Skia milik Google. Selain itu, *flutter engine* juga berinteraksi dengan perkakas pengembangan perangkat lunak (SDK) spesifik-serambi (*flatform-specific*) seperti yang disediakan oleh Android dan iOS.
2. Foundation Library

   Foundation library, ditulis dengan bahasa pemrograman Dart, menyediakan fungsi dan class-class dasar yang digunakan untuk membangun aplikasi menggunakan Flutter, seperti API untuk berkomunikasi dengan engine.
3. Widget Specific Design

   Framework Flutter berisi dua set widget yang disesuaikan dengan bahasa desain tertenu. *Widget Material Design* menerapkan bahasa desain Google dengan nama yang sama, sedangkan widget 'Cupertino' meniru desain iOS milik Apple.

2.7 Unified Modeling 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. Dalam merancang sebuah sistem, mengatur kompleksitas adalah salah satu alasan utama harus ada pembuatan model. Pemodelan membantu para pengembang untuk dapat mendokumentasikan, mengerti keseluruhan sistem, dan mengkomunikasikan aspek-aspek penting dalam sistem yang sedang dirancang.

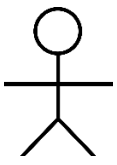
UML digunakan untuk memodelkan sistem mulai dari memodelkan informasi sistem untuk perusahaan hingga aplikasi web. UML menggunakan *class* dan *operation* sebagai konsep dasar untuk penulisan bahasa-bahasa berorientasi objek seperti C++, Java, C#,atau PHP. UML dideskripsikan menjadi beberapa bentuk diagram yaitu diantaranya [18] :

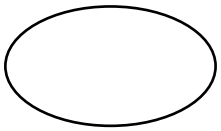
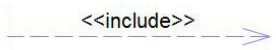
2.7.1 Use Case Diagram

Use Case Diagram adalah diagram yang menggambarkan kebutuhan sistem dari sudut pandang *user*, yang memperlihatkan hubungan-hubungan yang terjadi antara aktor dengan use case dalam sistem. Selain aktor, terdapat *communication line* dan *system boundaries* untuk menggambarkan *use case diagram* secara utuh bagaimana sistem digunakan untuk menyelesaikan pekerjaan sebuah aktor. *Communication line* menghubungkan aktor dan *use case* untuk menunjukan bahwa aktor berpartisipasi di dalam *use case*. Adapun *system bnoundaries* yang digunakan untuk menandakan pemisahan antara aktor dan *use case*.

Hubungan antar *use case* dapat diekspresikan menggunakan relasi <<*include*>> dan <<*extend*>>. Pada relasi <<*include*>> digunakan untuk memisahkan sistem menjadi bagian-bagian yang lebih mudah untuk diatur. Sementara untuk relasi <<*extend*>> menunjukan sebuah *use case* dapat sepenuhnya menggunakan *behavior* dari *use case* lain. Tabel 2.2 dibawah ini merupakan simbol-simbol yang digunakan pada *use case* [17].

Tabel 1.2 Daftar Simbol Use Case Diagram

No.	Simbol	Nama	Keterangan
1.		Aktor	Aktor adalah pengguna sistem yang tidak terbatas hanya manusia saja, namun bisa dalam bentuk sebuah sistem berkomunikasi dengan aplikasi lain serta membutuhkan input dan memberikan output, maka sistem tersebut dapat dianggap sebagai aktor.

2.		Use Case	Use case digambarkan sebagai sebuah fungsi tertentu dalam sebuah sistem maupun subsistem yang menghubungkan antara aktor dan sistem lainnya
3.		Association	Asosiasi digunakan untuk menghubungkan aktor dengan *use case*.
4.		Generalisasi	Penghubungan dimana aktor dan sistem merupakan generalisasi dari 1 sistem lainnya
5.		Extend	Relasi *use case* ke dalam sebuah *use case* dimana *use case* tambahan tersebut dapat berdiri sendiri tanpa *use case* tersebut
6.		Include	Relasi *use case* tambahan yang diperlukan untuk menjalankan fungsi pada *use case* utama atau sebagai syarat dalam menjalankan *use case* ini

2.7.2 Class Diagram

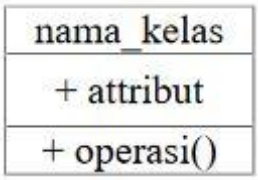
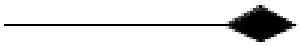
Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. *Class* menggambarkan keadaan (atribut / properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode / fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan obyek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* mempunyai sifat *visibility*, sebuah sifat yang dapat digunakan untuk mengatur akses ke atribut dan *operation*, yaitu adalah sebagai berikut :

1. *Private*, hanya dapat digunakan oleh metode pada *class* tersebut dan tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang memiliki metode bersangkutan dan *generalization*.
3. *Public*, dapat dipanggil oleh siapa saja.

4. *Package*, dapat dipanggil oleh *class* yang memiliki metode bersangkutan dan *class* lain yang berada di dalam *package* yang sama.

Berikut adalah simbol yang dimiliki *class diagram* dapat dilihat pada tabel 2.3.

Tabel 1.3 Daftar Simbol Class Diagram

No.	Simbol	Nama	Keterangan
1.		Kelas	Kelas merupakan himpunan dari objek-objek yang berisi atribut dan fungsi operasi yang ada
2.		Dependancy	Hubungan dimana perubahan yang terjadi pada suatu elemen akan berpengaruh ke elemen yang bersangkutan dengannya
3.		Aggregation	Menunjukan keseluruhan bagian hubungan dan biasanya disebut sebagai relasi "*whole-part*"
4.		Association	Relasi antarclass dengan arti umum, asosiasi biasanya juga disertai dengan *Multiplicity*
5.		Composition	Sebuah *class* tidak dapat berdiri sendiri, harus merupakan bagian dari *class* yang lain

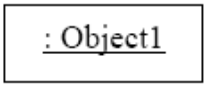
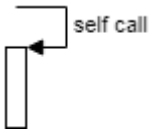
2.7.3 Sequence Diagram

Sequence Diagram menggambarkan interaksi antar obyek di dalam dan di sekitar sistem (termasuk pengguna, display dan sebagainya) berupa pesan yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence Diagram* digunakan sebagai gambaran skenario proses sebagai respon dari sebuah event untuk menghasilkan keluaran tertentu.

Dalam *sequence diagram* terdapat bagian-bagian yang disebut dengan *participant*, *time*, *message*, dan *activation bars*. *Participant* adalah bagian dari sistem yang memiliki interaksi antara satu dan sistem yang lainnya. *Time* menunjukan urutan dimana semua interaksi berlangsung sesuai dengan waktu dan ditunjukan dengan garis titik-titik vertikal pada *sequence diagram*. *Message* menunjukan penggunaan panah dari *participant* yang ingin mengirimkan pesan ke *participant* lain. *Message Caller* adalah sebutan untuk *participant* yang mengirim *message* dan akan diterima oleh *participant* lainnya yang disebut *Message Receiver*. *Activation bars* menunjukan bahwa sebuah *participant* sedang melakukan suatu aktivitas untuk jangka waktu tertentu.

Berikut adalah simbol yang dimiliki *sequence diagram* dapat dilihat pada tabel 2.4

Tabel 1.4 Daftar Simbol Sequence Diagram

No.	Simbol	Nama	Keterangan
1.		Object	Object merupakan intance dari sebuah *class*
2.		Lifeline	Lifeline menunjukan keberadaan sebuah object dalam waktu tertentu
3.		Activation	Activation menunjukan sebuah object yang akan melakukan suatu aksi
4.		Object Message	Menggambarkan pesan yang menghubungkan antar objek yang menunjukan kejadian yang sedang berlangsung
5.		Message to Self	Menggambarkan pesan yang menghubungkan objek itu sendiri yang menunjukan urutan kejadian yang berlangsung

2.7.4 Activity Diagram

Activiry Diagram adalah suatu cara dalam memodelkan event-event yang terjadi dalam diagram *use case*. *Activity diagram* menjelaskan tentang alur kegiatan dalam program yang sedang dirancang, bagaimana proses alur berawal, keputusan yang mungkin akan terjadi, dan bagaimana akhir dari hasil tersebut. Diagram ini disebut juga dengan state diagram khusus, dimana state ini berfungsi sebagai *action* dan sebagaian besar transisi dipanggil oleh state akhir sebelumnya (*internal processing*).

2.8 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 [17]:

1. Basis Data Konseptual

   Merupakan suatu proses yang membangun sebuah model data yang digunakan dalam sebuah perusahaan maupun organisasi, independen dari semua pertimbangan fisikal.
2. Basis Data Logikal

   Merupakan proses yang membangun sebuah model data yang digunakan dalam suatu perusahaan berdasarkan model data yang spesifik, namun tidak tergantung dari *Database Management System* (DBMS) tertentu dan pertimbangan fisikal lainnya.
3. Basis Data Fisikal

   Merupakan proses dalam menghasilkan deskripsi implementasi basis data pada *secondary storage*, proses ini menggambarkan hubungan dasar, organisasi file, dan indeks yang digunakan untuk akses yang efisien

terhadap data, setiap kendala integritas yang terkait dan tahap-tahap keamanan.

2.8.1 MySQL

MySQL *(My Structure Query Language*) merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS (*Database Managemetn 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) [19].

2.9 Firebase Cloud Messaging

Firebase adalah teknologi BaaS (*back-end as a service*) berbasis *cloud* dari Google yang menyediakan beberapa layanan yang salah satunya adalah layanan 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 ynag terhubung dari server [20]. Dalam penelitian ini FCM digunakan untuk mengirimkan notifikasi kepada pengguna aplikasi berupa peringatan cuaca melalui push notification.

Gambar 1.8 Firebase Cloud Messaging

(https://firebase.google.com/docs/cloud-messaging)

2.9.1 Notifikasi Firebase

Firebase Notifications adalah layanan gratis yang memungkinkan pemberitahuan pengguna yang ditargetkan untuk pengembang aplikasi mobile. Dibangun diatas *Firebase Cloud Messaging* dan FCM SDK, *Firebase Notifications* menyediakan berbagai opsi bagi developer dan organisasi yang mencari platform pemberitahuan yang fleksibel. Dengan GUI konsol *Notifications*, developer dapat membuatn interaksi dan mempertahankan basis penguna, membantu perkembangan aplikasi, dan mendukung kampanye pemasaran.

2.10 Application Programming Interface (API)

Application Programming Interface (API) merupakan suatu bahasa dan format pesan yang digunakan oleh program untuk dapat berkomunikasi dengan sistem operasi maupun pengendali program lainnya seperti sistem manajemen basis data (*Database Management System*) ataupun dengan protocol komunikasi. API di implementasikan dengan menuliskan fungsi (*function calls*) di dalam program, yang memberikan koneksi kepada subrutin yang diperlukan untuk eksekusi. Sehingga, sebuah API menunjukan bahwa beberapa modul program telah tersedia di dalam komputer untuk menjalankan operasi atau bahwa modul program tersebut dapat dihubungkan ke sebuah program yang telah ada untuk melakukan tugas yang telah ditentukan [21].

Jenis API berdasarkan penggunaannya terdapat 4 jenis API yang disesuaikan berdasarkan dengan hak aksesnya, yaitu sebagai berikut :

1. Public API

Public API merupakan API yang bisa digunakan oleh siapa saja dalam lintas platform. Jenis API ini paling mudah digunakan.

2. Private API

   Private API dibuat untuk keperluan internal dalam pengembangan aplikasi pada bagian *back end* yang digunakan untuk mengakses *front end* dari suatu website maupun aplikasi mobile.

3. Partner API

   Partner API merupakan jenis API yang dibuat untuk kepentingan umum tapi sebatas pihak yang sudah memiliki izin penggunaannya

4. Composite API

   Merupakan API yang menyimpan data dari berbagai server dalam satu tempat. Pengguna bisa mendapatkan berbagai jenis data dalam sekali akses dengan tujuan mempercepat *execution progress* (proses eksekusi) dan meningkatkan kinerja pendengar di antarmuka web.

2.10.1 REST (*Representational State Transfer*)

REST (*Representational State Transfer*) merupakan jenis arsitektur yang terdapat pada web untuk melayani suatu service. REST merepresentasikan cara interaksi antara suatu server dan client untuk melakukan proses pertukaran informasi melalui media yang sama. Dalam suatu jaringan agar suatu resource dapat diakses, maka diperlukan identifikasi dan suatu bentuk manipulasi. Dapat menggunakan URL (*Unirform Resource Identifier*) yang digunakan untuk mengidenntifikasi resource yang ada pada suatu jaringan, dan dapat membuat resource menjadi *addressable*, yang berarti resource dapat diketahui lokasinya dan dapat dimanipulasi dengan menggunakan suatu aplikasi [22].

REST dapat digunakan sebagai interface dari API untuk mengakses suatu resource. API yang mengikuti prinsip dari REST *architecture* memberikan kemudahan bagi developer untuk tidak perlu mengetahui bagaimana struktur dari API di dalam server. Dalam hal ini, server akan memberikan informasi bagaimana agar client dapat mengakses service yang diberikan oleh server.

Penggunaan protokol HTTP pada REST *architecture* untuk komunikasi antara client dan server terletak pada metode HTTP, yaitu GET, POST, PUT, dan DELETE. Metode ini dapat digunakan untuk mengakses resource yang ada pada server, bergantung dari instruksi yang diberikan oleh server. Dengan menggunakan protokol HTTP, URL dapat dijadikan sebagai media yang digunakan untuk mengakses resource dari server. Hal ini disebut dengan URL *tunneling*. URL *tunneling* menggunakan URL untuk mentransfer informasi antar sistem dalam jaringan dengan melakukan encode pada URL itu sendiri. Dengan mengirim metode HTTP, server dapat melakukan eksekusi terhadap suatu program yang menghasilkan atau mengambil suatu resource dan mengirimkannya kembali ke client.

2.10.2 Google Maps API

Google Maps API merupakan sebuah layanan (*service*) yang disediakan oleh Google kepada para pengembang sehingga para pengembang dapat memanfaatkan Google Map dalam mengembangkan aplikasi. Google Maps API menyediakan beberapa fitur yang dapat digunakan untuk memanipulasi peta serta menambah beberapa konten melalui beberapa jenis layanan (*service*) yang tersedia, selain itu Google mengijinkan kepada pengembang untuk membangun aplikasi *enterprise* di dalam websitenya maupun perangkat lunaknya [23].

Pengembang dapat memanfaatkan beberapa layanan-layanan yang ditawarkan oleh Google Maps setelah pengembang melakukan pedaftaran dan telah mendapatkan Google Maps API *Key*. Google menyediakan layanan ini kepada pengembang di seluruh dunia secara gratis. Berikut merupakan contoh *psycode* Google Map API dapat dilihat pada gambar 2.9

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title></title>
    <!-- Google Maps API reference -->
    <script src="https://maps.googleapis.com/maps/api/js?sensor=false&libraries=visualization">
    </script>
    <!--map references-->
    <link href="/mymap.css" rel="stylesheet" />
    <script src="/mymap.js"></script>
    <!-- USGS URL data source-->
    <script src="week.js">
    </script>
</head>
    <body>
        <div id="mapwindow"></div>
    </body>
</html>
```

Gambar 1.9 Google Map API

2.10.3 Twitter API

API Twitter atau Application Programming Interface (API) twitter adalah suatu program atau aplikasi yang disediakan oleh twitter untuk mempermudah developer lain dalam mengakses informasi yang ada di website twitter. Pendaftaran sebagai developer aplikasi twitter untuk menggunakan API twitter dapat dilakukan di lama https://dev.twitter.com. Setelah mendaftar developer akan mendapatkan consumer key, consumer access, access token dan access token secret yang akan digunakan sebagai syarat autentifikasi dari aplikasi yang akan kita bangun. Tujuan dari autentifikasi adalah untuk hak akses developer dalam mengunduh data yang ada di twitter [23].

2.10.4 OpenWeather API

Merupakan penyedia data cuaca harian, mingguan, serta bulanan. Layanan API yang disediakan oleh OpenWeather bisa digunakan untuk layanan kartografi pada aplikasi.