

BAB 2

TINJAUAN PUSTAKA

2.1. Cuaca

Cuaca menurut Kamus Besar Bahasa Indonesia (selanjutnya disebut KBBI) adalah keadaan udara pada satu tempat tertentu dengan jangka waktu terbatas. Keadaan udara yang dimaksud dapat berupa temperatur, kelembapan, kecepatan angin, cahaya matahari, dan sebagainya [13]. Setiap daerah dapat memiliki cuaca yang masing-masing berbeda, bahkan pada daerah yang sama pun dapat memiliki cuaca yang berbeda pada waktu yang berbeda.

National Geographic menjelaskan lebih lanjut mengenai komponen utama atau bagian dari cuaca. Komponen-komponen utama tersebut antara lain temperatur, tekanan atmosfer, angin, kelembapan, presipitasi, dan awan. Komponen tersebut digunakan bersamaan untuk menggambarkan cuaca pada saat tertentu. Komponen ini juga digunakan oleh ilmuwan meteorologi untuk membuat prediksi cuaca di masa mendatang [14].

2.2. Presipitasi

Presipitasi merupakan salah satu bagian dari cuaca. Presipitasi adalah bentuk air padat (dikenal sebagai salju) atau bentuk air cair (dikenal sebagai hujan) yang jatuh sampai permukaan tanah [15]. Presipitasi yang sering dijumpai di Indonesia adalah curah hujan.

Presipitasi digolongkan ke dalam beberapa bentuk presipitasi. Bentuk-bentuk itu antara lain [15]:

a. Gerimis

Gerimis umumnya jatuh dari awan stratus. Gerimis berupa tetes-tetes air kecil yang mana diameternya berkisar antara 0,2 sampai dengan 0,5 mm serta intensitas hujannya kurang dari 1 mm/jam.

b. Hujan

Hujan biasanya berasal dari awan nimbostratus dan cumulonimbus. Diameter tetesnya lebih besar dari 0,5 mm dengan intensitas hujan lebih dari 1,25 mm/jam.

c. Salju

Salju merupakan gumpalan kristal es dalam bentuk serpihan-serpihan. Ukuran serpihannya bergantung pada kadar air dan kelembaban pada sekitar kristal.

d. Pelet Es

Pelet es atau *ice pellet* sering kali disebut *sleet*, yakni tetes hujan yang membeku. Pelet es ini biasanya berdiameter kurang dari 5 mm.

e. Batu Es

Batu es atau *hail* adalah bongkahan es bergerigi atau bulat, sering ditandai dengan lapisan konsentris menyerupai struktur bawang. Batu es umumnya berdiameter lebih dari 5 mm.

2.3. Prediksi Cuaca

Prediksi cuaca telah menjadi sesuatu hal yang sangat penting bagi kehidupan manusia selama ribuan tahun. Salah satu contoh penggunaan prediksi cuaca adalah pada bidang agrikultural. Agrikultural membutuhkan prediksi cuaca yang tepat untuk menentukan kapan waktu yang tepat untuk menanam, untuk melakukan irigasi, dan kapan memanen hasil. Budaya kuno dari *Aztec of Mesoamerica* sampai kepada budaya orang Mesir di Afrika dan orang India di Asia, telah menjadi ahli astronomi serta orang yang mampu memprediksi pola cuaca musiman [14].

2.4. World Wide Web

World Wide Web (selanjutnya disebut Web) menurut KBBI adalah sistem untuk mengakses, memanipulasi, dan mengunduh dokumen hipertaut yang terdapat dalam komputer yang dihubungkan melalui internet; jejaring; jaringan [13]. Web saat ini menjadi salah satu teknologi yang sangat digunakan dalam

kehidupan sehari-hari manusia. Web dapat diakses melalui aplikasi *browser* yang terdapat pada berbagai *platform* seperti *Personal Computer*, Android, iOS, Windows Phone, dan lain sebagainya.

2.5. Android

Android merupakan salah satu sistem operasi yang digunakan pada *smartphone*. Arsitektur Android pertama kali dikembangkan oleh Android Inc. yang kini dimiliki oleh Google dan meluncurkan *Android Open Source Project* pada tahun 2007. Tujuan pembangunan *platform* ini adalah untuk membangun sebuah alat teknologi canggih yang memberikan efisiensi waktu dan biaya sejalan dengan peningkatan layanan dan menyediakan fitur terbaik kepada pengguna [16]. Pembangunan aplikasi di Android dapat menggunakan bahasa pemrograman Java, Kotlin, maupun C/C++ apabila diintegrasikan dengan *Native Development Kit* (selanjutnya disebut NDK). Riwayat perkembangan sistem operasi Android sejak awalnya hingga saat ini dapat dilihat pada Tabel 2.2.

Tabel 2.2 Riwayat Perkembangan Sistem Operasi Android

<i>Codename</i>	<i>Version</i>	<i>API Level/NDK Release</i>
<i>(no codename)</i>	1.0	<i>API level 1</i>
<i>(no codename)</i>	1.1	<i>API level 2</i>
<i>Cupcake</i>	1.5	<i>API level 3, NDK 1</i>
<i>Donut</i>	1.6	<i>API level 4, NDK 2</i>
<i>Eclair</i>	2.0	<i>API level 5</i>
<i>Eclair</i>	2.0.1	<i>API level 6</i>
<i>Eclair</i>	2.1	<i>API level 7, NDK 3</i>
<i>Froyo</i>	2.2.x	<i>API level 8, NDK 4</i>
<i>Gingerbread</i>	2.3 - 2.3.2	<i>API level 9, NDK 5</i>
<i>Gingerbread</i>	2.3.3 - 2.3.7	<i>API level 10</i>
<i>Honeycomb</i>	3.0	<i>API level 11</i>
<i>Honeycomb</i>	3.1	<i>API level 12, NDK 6</i>
<i>Honeycomb</i>	3.2.x	<i>API level 13</i>
<i>Ice Cream Sandwich</i>	4.0.1 - 4.0.2	<i>API level 14, NDK 7</i>
<i>Ice Cream Sandwich</i>	4.0.3 - 4.0.4	<i>API level 15, NDK 8</i>
<i>Jelly Bean</i>	4.1.x	<i>API level 16</i>
<i>Jelly Bean</i>	4.2.x	<i>API level 17</i>
<i>Jelly Bean</i>	4.3.x	<i>API level 18</i>
<i>KitKat</i>	4.4 - 4.4.4	<i>API level 19</i>
<i>Lollipop</i>	5.0	<i>API level 21</i>
<i>Lollipop</i>	5.1	<i>API level 22</i>
<i>Marshmallow</i>	6.0	<i>API level 23</i>
<i>Nougat</i>	7.0	<i>API level 24</i>
<i>Nougat</i>	7.1	<i>API level 25</i>
<i>Oreo</i>	8.0.0	<i>API level 26</i>

<i>Codename</i>	<i>Version</i>	<i>API Level/NDK Release</i>
<i>Oreo</i>	8.1.0	<i>API level 27</i>
<i>Pie</i>	9	<i>API level 28</i>
<i>Android 10</i>	10	<i>API level 29</i>

Sumber Tabel : <https://source.android.com/setup/start/build-numbers>

2.6. Pemrograman Berorientasi Obyek

Pemrograman berorientasi obyek atau *Object Oriented Programming* (selanjutnya disebut OOP) adalah pendekatan yang berfokus pada interaksi obyek untuk saling komunikasi dan berbagi informasi. Pemrograman tipe ini merupakan alternatif dari penggunaan pemrograman berorientasi prosedural yang tradisional, dimana fokus orientasinya yakni pada prosedur pelaksanaannya. OOP berurusan dengan *minimum of three software engineering goals*, antara lain *reuseability*, *extensibility*, dan *flexibility*. Pentingnya penggunaan OOP adalah OOP mampu menyediakan pola desain yang baik dan dapat menjadi acuan bagi pengguna dalam membuat obyek baru serta *user friendly* [17].

Faktor-faktor yang penting dalam setiap OOP adalah *object*, *class*, *inheritance*, dan *polymorphism*. *Object* berperan sebagai mediator antara program dengan metode yang ditambahkan di dalamnya. *Class* berisikan seluruh metode-metode serta definisinya yang digunakan dalam program. *Inheritance* menyediakan ketentuan bagi pengguna untuk menggunakan kode yang sama dalam programnya namun mengubah definisi dalam variabelnya untuk mengurangi waktu eksekusi dan mengurangi kemungkinan terjadinya kesalahan. *Polymorphism* memungkinkan pengguna untuk mendefinisikan sebuah kode yang dapat digunakan oleh banyak variabel sehingga pelaksanaan eksekusi program menjadi lebih cepat [17].

2.7. Unified Modelling Language

Unified Modelling Language (selanjutnya disebut UML) merupakan standarisasi tata pemodelan untuk pembangunan perangkat lunak dan sistem. Kelebihan utama penggunaan UML adalah sebagai berikut [18]:

a. Bahasa Formal

Setiap elemen dalam bahasa ini memiliki makna yang kuat dan mendalam, sehingga penggunaannya dapat yakin bahwa tidak akan terjadi kesalahpahaman ketika menggunakan suatu model untuk segi tertentu dalam sistem.

b. Sederhana

Seluruh bahasa ini dibangun dengan kesederhanaan dan notasi yang mudah.

c. Komprehensif

Bahasa ini dapat menjelaskan seluruh aspek penting dalam sistem.

d. Kapasitas Dinamis

Ketika dibutuhkan, bahasa ini mampu mengatasi pemodelan sistem proyek yang sangat besar, selain itu juga mampu menangani proyek yang kecil.

e. Dibangun Berdasarkan Pengalaman

UML adalah puncak dari praktek terbaik dalam komunitas OOP selama 15 tahun.

f. Merupakan Standar Umum

UML dikelola oleh kelompok standarisasi terbuka dengan kontribusi aktif dari vendor dan akademisi seluruh dunia.

Saat ini UML telah mencapai versi 2.0. Versi ini diterbitkan untuk menjawab tantangan-tantangan baru yang dihadapi oleh pelaku pemodelan sistem dan perangkat lunak. Diagram-diagram yang terdapat dalam UML akan dijabarkan dalam Tabel 2.3.

Tabel 2.3 Diagram dalam UML

Tipe Diagram	Yang Dapat Dimodelkan	Dikenalkan Dalam Versi
<i>Use Case</i>	Interaksi antara sistem dengan pengguna maupun pihak eksternal, serta membantu memetakan kebutuhan sistem.	UML 1.x
<i>Activity</i>	Aktivitas sekuensial dan paralel dalam sistem.	UML 1.x
<i>Class</i>	<i>Classes</i> , tipe-tipe, <i>interfaces</i> , dan relasi diantaranya.	UML 1.x
<i>Object</i>	Contoh <i>object</i> dari <i>class</i> dalam <i>class diagram</i> yang penting terhadap sistem.	UML 1.x secara informal

Tipe Diagram	Yang Dapat Dimodelkan	Dikenalkan Dalam Versi
<i>Sequence</i>	Interaksi antara <i>object</i> dengan <i>object</i> dimana urutan interaksinya penting untuk diperhatikan.	UML 1.x
<i>Communication</i>	Bagaimana <i>object</i> berinteraksi dan koneksi yang dibutuhkan untuk mendukung interaksi tersebut.	Ganti nama dari <i>collaboration diagram</i> dalam UML 1.x
<i>Timing</i>	Interaksi antara <i>object</i> dengan <i>object</i> dimana waktu kejadian menjadi penting untuk diperhatikan.	UML 2.0
<i>Interaction Overview</i>	Digunakan untuk mengumpulkan <i>sequence</i> , <i>communication</i> , dan <i>timing diagram</i> bersama untuk menangkap interaksi penting yang terjadi dalam sistem.	UML 2.0
<i>Composite Structure</i>	Bagian dalam dari <i>class</i> atau <i>component</i> dan dapat menjelaskan hubungan <i>class</i> dalam konteks yang diberikan.	UML 2.0
<i>Component</i>	<i>Component</i> yang penting dalam sistem dan antarmuka yang digunakan untuk berinteraksi dengan yang lainnya.	UML 1.x, tetapi menerima makna baru dalam UML 2.0
<i>Package</i>	Susunan hierarkis kelompok <i>class</i> dan <i>component</i>	UML 2.0
<i>State Machine</i>	Kedudukan suatu <i>object</i> melalui masa hidupnya dan kejadian yang dapat mengubah kedudukan itu.	UML 1.x
<i>Deployment</i>	Bagaimana sistem akhirnya dihadapkan dalam situasi dunia nyata yang diberikan.	UML 1.x

Sumber Tabel : *Learning UML 2.0 (2006)* [18]

2.8. Basis Data

Basis data atau *database* adalah kumpulan data yang secara khusus menggambarkan aktivitas satu atau lebih relasi dalam organisasi. Basis data telah menjadi kebutuhan bagi semua perangkat lunak karena data telah menjadi suatu aset yang sangat berharga dan harus dikelola secara optimal sehingga dapat dipertanggungjawabkan [19]. Pengelolaan basis data menggunakan sistem manajemen basis data.

Sistem manajemen basis data atau *database management system* (selanjutnya disebut DBMS) adalah perangkat lunak yang didesain untuk mendukung dalam memelihara dan menyelenggarakan kumpulan data yang besar. DBMS memiliki kelebihan sebagai berikut [19]:

a. Independensi Data

Program aplikasi secara ideal seharusnya tidak membeberkan detail gambaran data dan penyimpanannya. DBMS menyediakan gambaran abstraksi data yang menyembunyikan detail-detail tersebut.

b. Akses Data Efisien

DBMS menyelenggarakan berbagai macam teknik canggih untuk menyimpan dan mengambil data secara efisien. Fitur ini sangat penting khususnya apabila datanya disimpan pada perangkat penyimpanan eksternal.

c. Data Integritas dan Keamanan

Jika data dapat diakses melalui DBMS, DBMS dapat memaksakan integritas data. Contohnya sebelum memasukkan informasi gaji kepada seorang pegawai, DBMS dapat memeriksa bahwa alokasi dana departemennya memadai. DBMS juga dapat memberikan kontrol akses yang bekerja mengenai data apa yang terlihat kepada berbagai golongan pengguna.

d. Administrasi Data

Ketika beberapa pengguna berbagi data, Pemusatan administrasi data dapat memberikan peningkatan yang signifikan. Profesional berpengalaman yang memahami bagaimana sifat data yang dikelola, dan bagaimana kelompok pengguna lainnya menggunakannya, DBMS dapat bertanggungjawab dalam mengelola representasi data untuk meminimalkan *redundancy* dan untuk setelan penyimpanan data yang bagus untuk membuat pengambilan data yang efisien.

e. Akses Berbarengan dan Pemulihan Kegagalan

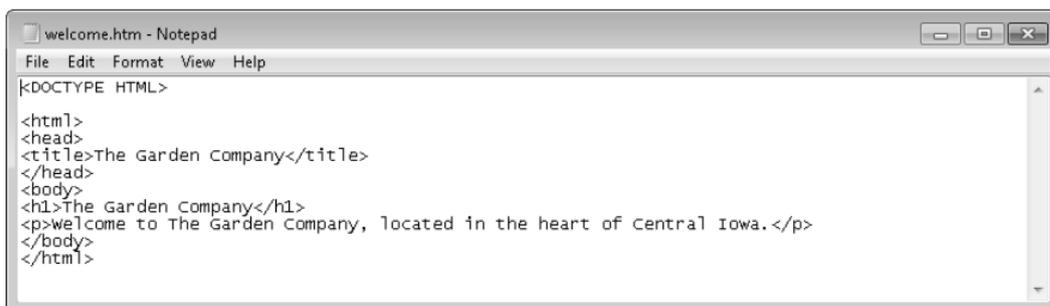
DBMS menjadwalkan akses-akses yang bersamaan mengakses data seperti halnya pengguna berpikir bahwa data sedang diakses oleh satu pengguna dalam satu waktu saja. Lebih lanjut, DBMS melindungi pengguna dari efek kesalahan sistem.

f. Pengurangan Waktu Pembangunan Aplikasi

DBMS mendukung fungsi penting yang umum digunakan oleh banyak aplikasi yang mengakses data dalam DBMS. Hal ini dalam hubungannya dengan antarmuka tingkat tinggi kepada data, difasilitasi pengembangan aplikasi yang cepat. Aplikasi dengan DBMS juga cenderung lebih kuat dibanding aplikasi *stand-alone* karena banyak tugas penting yang dipegang oleh DBMS.

2.9. Hypertext Markup Language

Hypertext Markup Language (selanjutnya disebut HTML) adalah *markup language* yang mendasari web. Makna dari HTML menurut susunan katanya terbagi ke dalam tiga kata, yakni *Hypertext*, *Markup*, dan *Language*. *Hypertext* artinya teks yang di-*click* akan membuka dokumen baru. Hal ini merujuk pada kemampuan halaman web untuk terhubung dengan halaman lainnya. *Markup* artinya *tags* yang menerapkan ketentuan tata letak dan format ke dalam teks biasa. Secara harafiah, teks biasa itu dibungkus dengan *tags*. *Language* menunjukkan fakta bahwa HTML dipertimbangkan sebagai bahasa pemrograman [20]. Contoh sintaks HTML dapat dilihat pada Gambar 2.1.



```
welcome.htm - Notepad
File Edit Format View Help
<!DOCTYPE HTML>
<html>
<head>
<title>The Garden Company</title>
</head>
<body>
<h1>The Garden Company</h1>
<p>welcome to The Garden Company, located in the heart of Central Iowa.</p>
</body>
</html>
```

Sumber Gambar : *HTML5: Step by Step (2001)* [20]

Gambar 2.1 Contoh Sintaks HTML

2.10. JavaScript

JavaScript pertama kali dikenalkan pada tahun 1995 sebagai jalan untuk menambah program ke halaman web di *Netscape Navigator browser*. Saat ini JavaScript tidak hanya bisa digunakan *web browser* saja, namun beberapa

platform lainnya juga. Basis data seperti MongoDB dan CouchDB, menggunakan JavaScript sebagai *scripting* dan bahasa *query* mereka. *Platform desktop* dan pemrograman *server* seperti proyek *Node.js*, menyediakan sebuah lingkungan untuk pemrograman JavaScript di luar *browser* [21]. Contoh sintaks JavaScript dapat dilihat pada Gambar 2.2.

```

let num = Number(prompt("Pick a number"));

if (num < 10) {
  console.log("Small");
} else if (num < 100) {
  console.log("Medium");
} else {
  console.log("Large");
}

```

Sumber Gambar : *Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming (2018)* [21]

Gambar 2.2 Contoh Sintaks JavaScript

2.11. *PHP: Hypertext Preprocessor*

PHP: Hypertext Preprocessor (selanjutnya disebut PHP) adalah salah satu bahasa pemrograman yang paling terkenal dalam pembangunan web. PHP telah menjadi pilihan faktual bagi para pengembang aplikasi web, terlebih karena PHP memiliki kemampuan kapasitas dinamis, mudah digunakan, dan didukung oleh banyak basis data dan format data [22]. Contoh sintaks PHP dapat dilihat pada Gambar 2.3.

PHP memiliki beberapa fitur unik jika dibandingkan dengan bahasa pemrograman *server* lainnya. Fitur unik tersebut antara lain [22]:

a. Performansi

Sintaks yang ditulis dalam PHP dieksekusi lebih cepat dibanding bahasa pemrograman *server* lainnya. Mesin PHP 5.0 didesain dengan manajer memori yang dioptimasi untuk meningkatkan performa.

b. Mobilitas

PHP tersedia untuk UNIX, Microsoft Windows, Mac OS, dan OS/2. Kemampuan untuk membangun aplikasi *cross-platform* merupakan poin lebih karena dapat mencakup banyak segmen pasar.

c. Kemudahan Penggunaan

PHP merupakan bahasa pemrograman yang sangat canggih, sintaksnya jelas dan konsisten, serta dokumentasi yang lengkap untuk lebih dari 5000 fungsi yang terdapat dalamnya.

d. Sumber Terbuka

PHP merupakan proyek *open source* atau sumber terbuka, artinya bahasa pemrograman ini dikembangkan oleh banyak voluntir dari seluruh dunia karena kode sumber tersedia di web secara gratis dan dapat digunakan tanpa lisensi.

e. Dukungan Komunitas

Dukungan komunitas PHP menawarkan kreatifitas serta imajinasi dari ratusan pengembang yang tersebar di seluruh dunia. Hasil komunitas ini dapat dilihat pada *PHP Extension and Application Repository* atau PEAR maupun *PHP Extension Community Library* atau PECL.

f. Dukungan Aplikasi Pihak Ketiga

Salah satu kekuatan PHP adalah dukungan yang luas terhadap berbagai macam basis data, seperti MySQL, PostgreSQL, Oracle, dan Microsoft SQL Server.

```
<html>
  <head></head>
  <body>
    <div>
      <?php echo sqrt(49); ?>
    </div>
  </body>
</html>
```

Sumber Gambar : *PHP: A Beginner's Guide (2009)* [22]

Gambar 2.3 Contoh Sintaks PHP

2.12. Laravel

Laravel adalah sebuah *framework* aplikasi web dengan sintaks yang ekspresif dan elegan [23]. Laravel diciptakan oleh Taylor Otwell pada tahun 2011 di bawah lisensi *Massachusetts Institute of Technology*. Menurut Amar InfoTech dengan melihat pada grafik Google Trend, Laravel merupakan *framework* PHP yang paling populer di tahun 2018 [24].

2.13. Brackets

Brackets adalah sebuah aplikasi teks editor modern yang *open source* serta memahami desain web. Brackets didirikan oleh Adobe di bawah lisensi MIT. Brackets dibangun dengan tujuan untuk mendorong para pengembang web, khususnya para pengembang *frontend* menuju tingkat selanjutnya [25]. Walaupun demikian, aplikasi ini mendukung peluasan fungsi oleh pihak ketiga untuk menyediakan fasilitas yang dibutuhkan oleh para pengembang perangkat lunak lainnya. Contoh tampilan antarmuka Brackets dapat dilihat pada Gambar 2.4.



Sumber Gambar : <http://brackets.io/>

Gambar 2.4 Contoh Tampilan Antarmuka Brackets

2.14. Java

Java adalah sebuah bahasa pemrograman yang dikembangkan oleh Sun Microsystems dan mulai diperkenalkan kepada publik pada tahun 1995. Java juga merupakan bahasa pemrograman yang berorientasi obyek, sehingga Java memudahkan pengguna dalam membuat aplikasi yang berskala besar [26]. Contoh sintaks Java dapat dilihat pada Gambar 2.5.

```

import java.io.IOException;

public class TomcatEmbedded {

    private static final String EMPTY = "";

    public static void main(String... args)
        throws Exception {
        File baseFolder = new File(System.getProperty("user.dir"));
        File appsFolder = new File(baseFolder, child: "apps");

        Tomcat tomcat = new Tomcat();
        tomcat.setBaseDir(baseFolder.getAbsolutePath());
        tomcat.setPort(8080);
        tomcat.getHost().setAppBase(appsFolder.getAbsolutePath());

        // Call the connector to create the default connector.
        tomcat.getConnector();

        tomcat.addWebapp(EMPTY, docBase: ".");
        Wrapper wrapper = tomcat.addServlet(EMPTY, servletName: "hello", new HelloServlet());
        wrapper.setLoadOnStartup(1);
        wrapper.addMapping(s: "/*");

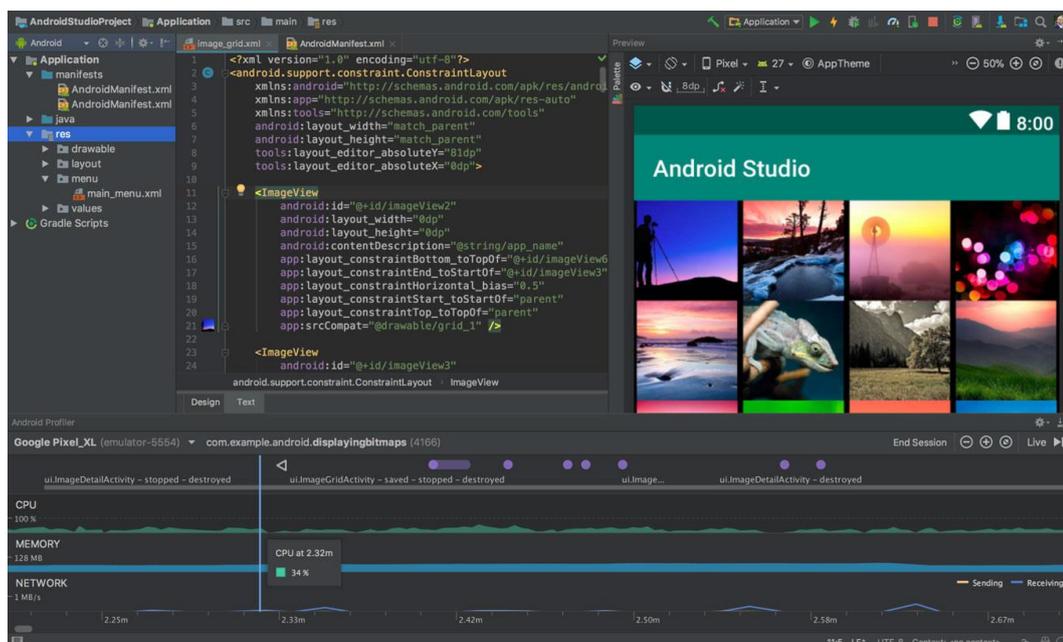
        tomcat.start();
        tomcat.getServer().await();
    }
  
```

Sumber Gambar : <https://medium.com/oracledevs/>

Gambar 2.5 Contoh Sintaks Java

2.15. Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (selanjutnya disebut IDE) untuk sistem operasi Android. IDE ini dibangun di atas perangkat lunak *JetBrains IntelliJ IDEA* dan didesain khusus hanya untuk pengembangan Android. Android Studio menggantikan *Eclipse Android Development Tool* yang sebelumnya merupakan IDE utama dalam pembangunan aplikasi Android. Android Studio pertama kali dipublikasikan di konferensi *Google I/O* pada tanggal 16 Mei 2013 [27]. Saat ini Android Studio telah diperbaharui hingga versi 3.5.1. Contoh tampilan antarmuka Android Studio dapat dilihat pada Gambar 2.6.



Sumber Gambar : <https://developer.android.com/studio>

Gambar 2.6 Contoh Tampilan Antarmuka Android Studio

2.16. Application Program Interface

Application Program Interface (selanjutnya disebut API) adalah sebuah alat yang membuat data dalam sebuah halaman web dapat dicerna oleh sebuah komputer [28]. Secara sederhana, API merupakan sebuah antarmuka yang digunakan oleh komputer untuk mengakses *server* layaknya seorang pengguna

yang mengakses *server* melalui halaman web. Tujuan dari API adalah pengalihan kegiatan yang dilakukan oleh pengguna kepada komputer untuk mempercepat kecepatan proses, sehingga waktu yang dibutuhkan oleh pengguna dalam menyelesaikan tugasnya menjadi lebih singkat.

2.17. *Google Directions API*

Google Directions API adalah sebuah layanan yang mencari arah rute perjalanan antarlokasi. Pencarian rute perjalanan yang disediakan terbagi berdasarkan jenis alat transportasinya, termasuk di dalamnya transit, mengemudi, berjalan kaki, maupun bersepeda [29]. Layanan ini dapat dimanfaatkan oleh para pengembang aplikasi untuk memperoleh rute perjalanan dari suatu titik lokasi menuju titik lokasi lainnya tanpa harus menyimpan data wilayah.

Layanan ini diakses melalui antarmuka HTTP dengan data yang diminta dibangun sebagai kalimat URL menggunakan teks atau koordinat bujur dan lintang untuk mengidentifikasi lokasi tertentu, bersamaan dengan kunci API-nya. Contoh permintaan layanan *Google Directions API* dapat dilihat pada Gambar 2.7. Setiap permintaan layanan ini akan menghasilkan respon berupa data dalam format *JavaScript Object Notation* (selanjutnya disebut JSON). Contoh respon layanan *Google Directions API* dapat dilihat pada Gambar 2.8.

```
https://maps.googleapis.com/maps/api/directions/json?origin=Disneyland&destination=Universal+Studios+Hollywood&key=YOUR_API_KEY
```

Sumber Gambar : <https://developers.google.com/maps/documentation/directions/start>

Gambar 2.7 Contoh Permintaan Layanan *Google Directions API*

```

    "duration" : {
      "text" : "51 mins",
      "value" : 3062
    },
    "end_address" : "Universal Studios Blvd, Los Angeles, CA 90068, USA",
    "end_location" : {
      "lat" : 34.1330949,
      "lng" : -118.3524442
    },
    "start_address" : "Disneyland (Harbor Blvd.), S Harbor Blvd, Anaheim, CA 92802,
USA",
    "start_location" : {
      "lat" : 33.8098177,
      "lng" : -117.9154353
    },
    ... Additional results truncated in this example[] ...

    "overview_polyline" : {
      "points" :
"knjnmEnjunUbKkFEA?_@|kMBEe@qIloF@wH@eFFk@WOUI_@?u@j@k@`@EXLTZHh@Y`AgApAaCrCUd@eDpDuAtAoApA(Y1
ZiBdBaIhGkFrDeCtBuFxFmIdJmOjPaChDeBldiAdD)ApGcDxU)@hEmAxD)[tt@yMb\\yEdEqFnJqB~DeFxmGk~Vsmr[uKs
VoCxEsEtG)BcChHkWh@]t@{AxEcClLkCjLi@`CwBfHaEzJuBdEyEhIaBnCiF|K_Os\\
(MdZwAbDaKbUiB|CgCnDkDbEiE|FqBlDsLdXqQra@kX|m@aF|KcHtIm@pAaE~JcTwh@w\\`v@gQv`@}F`M
qK`PeGsIyGfJiG~GeLhLgIpIcE~FsDxHcFfLgDzH(CxwAbBgC|B)F|DiQaKsbBdeA(k@~\\oc@bWoKjGaEsCoEsEwDxFs
Uh~wJfOySx[uBnCGCbCoFlDmDvAiCr@eRzDunXc_EvAiFpCaC|AgGpEwHsFoQnQoTrTqBlCyDnGmCfEmDpDyGzGsIsHuZs
YwBpB=C`CqBlAsBhAqCxAoBrAqDdDcNfMgHhHiFtReBtCkD|GqAhBwBzBsG~FoAhAaCbDeBvD_BlEyM`@uBvKiA~DmAlC
kA|B)@lBcChHoJnXcB`GoAnIS~CIjFDd]A|QMLD(@jH[vAk@`CoGxRgPsf@aBbHoB~HeMx^eDtJ)BnG(DhJU)`mBscCjD
aAx@mAnAgCnEmAp@uAj@(Cr@wBfKb@kBSsEW(GV)BEeCWyAWwHs@qH?
cIHkDXuDn@mCt@mE`BsH|CyAp@)AdAaAtAy@lBg@pCa@jE]fEcBhRq@pJKlCk@hLFrB@lD`@xCeA`DoBxD
aHvM_FnImDsFeCpDeC|ChExDiJrHcBtAkDpDwObVucPFeCdHoIl\\uBjIuClJsEvMyDbMqAhEoDlJ(C|J)FlZuBfLyDlXw
B~QkArG_AnDiAxC(G|OgEdLaE`LkEbEwG~KgHnLoEjGgDxCaC`BuJdFkFtCgCnBuClD_HdMgEsHcBpE_C|BuEsCmPlIuE|
B_EtDeBhCgAdCw@rCi@|DSfECrCAdCS~Di@jDYhA_AlC(AxAcL`U(GvM_DjFkBsBsB`BqDhBaEfaTveEmEz@iCr@gDrAiF
nCCEsCaE~D_@JmFdGQDwBvCeErEoD|BcFjC|DbEuD~D`@Zr@h@?d@W@)@vAgCbEaHfMqa`Cy@dAg@bA0`@gCi@w@W"
    },
    "summary" : "I-5 N and US-101 N",
    "warnings" : [],
    "waypoint_order" : []
  },
}
"status" : "OK"
}

```

Sumber Gambar : <https://developers.google.com/maps/documentation/directions/start>

Gambar 2.8 Contoh Respon Layanan *Google Directions API*

2.18. Dark Sky API

Dark Sky API adalah sebuah layanan yang memungkinkan penggunanya untuk melihat cuaca di dunia belahan manapun. Hasil yang akan diberikan (apabila tersedia) adalah cuaca saat ini, prediksi cuaca menit per menit sampai dengan satu jam yang akan datang, prediksi cuaca jam per jam dan hari per hari sampai dengan tujuh hari mendatang, observasi cuaca jam per jam dan hari per hari sampai satu dekade sebelumnya, dan peringatan cuaca berbahaya untuk Amerika Serikat, Kanada, negara anggota Uni Eropa, dan Israel [30].

Permintaan layanan *Dark Sky API* dibagi menjadi 2, yaitu permintaan prediksi, dan permintaan mesin waktu. Permintaan prediksi akan mengembalikan hasil prediksi cuaca saat ini sampai dengan pekan selanjutnya. Permintaan mesin waktu akan mengembalikan hasil observasi kondisi cuaca atau prediksi cuaca untuk suatu waktu di masa lampau atau di masa depan. Contoh permintaan prediksi dapat dilihat pada Gambar 2.9, sedangkan permintaan mesin waktu dapat dilihat pada Gambar 2.10. Contoh respon permintaan prediksi dapat dilihat pada Gambar 2.11, sedangkan respon permintaan mesin waktu dapat dilihat pada Gambar 2.12 [30].

Kondisi cuaca yang dihasilkan oleh *Dark Sky API* dapat didefinisikan ke dalam komponen-komponen cuaca. Komponen cuaca yang dimaksud antara lain temperatur semu, tekanan atmosfer, tutupan awan, titik embun, kelembapan, tingkat presipitasi air, fase bulan, jarak badai terdekat, arah badai terdekat, ozon, tipe presipitasi, turun salju, temperatur, ringkasan teks, indeks UV, hembusan angin, kecepatan angin, dan arah angin. Kedua jenis permintaan layanan *Dark Sky API* akan mengembalikan hasil dengan komponen yang sama dalam bentuk JSON [30].

```
https://api.darksky.net/forecast/[key]/[latitude],[longitude]
```

Sumber Gambar : <https://darksky.net/dev/docs#overview>

Gambar 2.9 Contoh Permintaan Prediksi *Dark Sky API*

```
https://api.darksky.net/forecast/[key]/[latitude],[longitude],[time]
```

Sumber Gambar : <https://darksky.net/dev/docs#overview>

Gambar 2.10 Contoh Permintaan Mesin Waktu *Dark Sky API*

```
{
  "latitude": 42.3601,
  "longitude": -71.0589,
  "timezone": "America/New_York",
  "currently": {
    "time": 1509993277,
    "summary": "Drizzle",
    "icon": "rain",
    "nearestStormDistance": 0,
    "precipIntensity": 0.0089,
    "precipIntensityError": 0.0046,
    "precipProbability": 0.9,
    "precipType": "rain",
    "temperature": 66.1,
    "apparentTemperature": 66.31,
    "dewPoint": 60.77,
    "humidity": 0.83,
    "pressure": 1010.34,
    "windSpeed": 5.59,
    "windGust": 12.03,
    "windBearing": 246,
    "cloudCover": 0.7,
    "uvIndex": 1,
    "visibility": 9.84,
    "ozone": 267.44
  },
  "minutely": {
    "summary": "Light rain stopping in 13 min., starting again 30 m
```

Sumber Gambar : <https://darksky.net/dev/docs#overview>

Gambar 2.11 Contoh Respon Prediksi Dark Sky API

```

{
  "latitude": 42.3601,
  "longitude": -71.0589,
  "timezone": "America/New_York",
  "hourly": {
    "summary": "Snow (6-9 in.) and windy starting in the at",
    "icon": "snow",
    "data": [
      {
        "time": 255589200,
        "summary": "Mostly Cloudy",
        "icon": "partly-cloudy-night",
        "precipIntensity": 0,
        "precipProbability": 0,
        "temperature": 22.8,
        "apparentTemperature": 16.46,
        "dewPoint": 15.51,
        "humidity": 0.73,
        "pressure": 1026.78,
        "windSpeed": 4.83,
        "windBearing": 354,
        "cloudCover": 0.78,
        "uvIndex": 0,
        "visibility": 9.62
      },
      ...
    ]
  },
},

```

Sumber Gambar : <https://darksky.net/dev/docs#overview>

Gambar 2.12 Contoh Respon Mesin Waktu Dark Sky API