BAB 2
LANDASAN TEORI

2.1 Landasan Teori

Landasan teori berfungsi menjelaskan teori-teori dasar terkait teknologi yang digunakan serta informasi yang relevan dengan aplikasi yang akan dibangun. Dalam pembangunan aplikasi ini, landasan teori meliputi beberapa konsep dasar, seperti definisi buah, teknologi yang digunakan, serta perangkat lunak dan keras yang mendukung, seperti Android, Java, dan teknologi lainnya yang berkaitan dengan pengembangan aplikasi..

2.2 Deteksi

Deteksi adalah proses mengenali atau mengidentifikasi keberadaan serta karakteristik suatu objek, materi, atau kejadian dengan menggunakan alat, teknologi, atau metode khusus. Tujuan deteksi adalah menemukan hal-hal yang mungkin tidak terlihat atau diketahui secara langsung. Proses ini digunakan dalam berbagai bidang seperti teknologi informasi, ilmu pengetahuan, kesehatan, keamanan, dan pengembangan sistem kecerdasan buatan. Deteksi sering menjadi langkah awal dalam menyelesaikan masalah atau memperdalam pemahaman terhadap lingkungan. Tujuan akhirnya adalah memberikan solusi yang efektif melalui implementasi metode yang tepat sesuai dengan permasalahan yang dihadapi[8].

Proses atau cara kerja deteksi sangat bervariasi tergantung pada jenis deteksi dan teknologi seperti apa yang digunakan. Secara umum, proses deteksi melibatkan beberapa langkah dasar diantaranya :

1. Pendeteksian

Pemilihan sensor atau alat deteksi yang sesuai dengan tujuan deteksi yang akan dibangun atau dikembangkan.

2. Pengambilan data

Informasi yang diterima oleh sensor direkam atau diukur dalam bentuk yang dapat diproses lebih lanjut. Data tersebut dapat berupa nilai numerik, citra, suara, atau jenis data lainnya yang relevan dengan tujuan deteksi.

3. Pemrosesan data

Data yang dihasilkan oleh sensor kemudian diproses menggunakan teknik-teknik khusus atau algoritma untuk mengenali pola, mencari tanda-tanda yang relevan, atau mengidentifikasi informasi yang penting sesuai dengan kebutuhan untuk melakukan deteksi.

4. Analisis dan Identifikasi

Data yang telah diproses dipecahkan, dianalisis, dan dibandingkan dengan parameter yang telah ditentukan sebelumnya. Ini bertujuan untuk mengidentifikasi atau mendeteksi kejadian, objek, atau pola tertentu.

5. Pemberian Informasi atau tindakan

Hasil dari analisis data digunakan untuk memberikan informasi kepada pengguna atau sistem. Ini bisa berupa notifikasi, tindakan otomatis, atau penyimpanan data untuk penggunaan lebih lanjut.

Setiap aplikasi atau sistem yang dibangun memiliki proses pendekatan dan teknologi yang berbeda, namun prinsip dasarnya tetap sama, yaitu untuk mendeteksi objek tertentu, informasi, menganalisis data, dan memberikan respon yang sesuai berdasarkan hasil pada informasi yang didapat.

2.3 Buah

Buah merupakan bagian dari suatu tumbuhan yang berasal dari bunga atau putik dari tumbuhan tersebut dan biasanya memiliki biji. Buah mengandung berbagai macam vitamin, mineral, dan serat pangan yang berperan untuk membantu proses metabolisme dalam tubuh[9]. Vitamin dan mineral pada buah tidak dapat digantikan oleh bahan pangan lainnya.

2.4 Tingkat Kematangan Buah

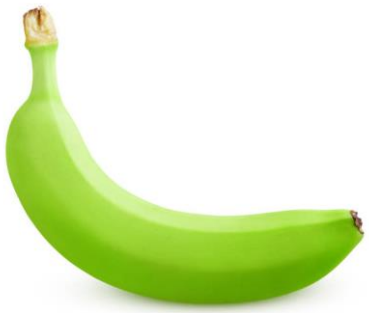
Tingkat kematangan buah adalah tahap perkembangan buah di mana ia mencapai kondisi optimal untuk dikonsumsi, baik dari segi rasa, tekstur, aroma, maupun nilai gizi. Tingkat kematangan ini dapat bervariasi untuk setiap jenis buah dan biasanya ditandai oleh perubahan fisik seperti warna, kekerasan, dan aroma. Kematangan yang tepat sangat penting untuk memastikan kualitas dan kesegaran buah saat dikonsumsi atau dijual[10].

2.4.1 Tingkat Kematangan Buah Pisang

Berikut adalah tingkat kematangan buah pisang pada umumnya, diantaranya sebagai berikut[11]:

1. Mentah (Hijau)

Buah pisang mentah memiliki ciri-ciri kulit berwarna hijau, tekstur keras, dan rasa agak pahit dan sepat karena kandungan pati yang tinggi, dengan kandungan nutrisi yang terdiri dari pati tinggi dan gula rendah, serta kaya akan serat, vitamin C, dan beberapa mineral seperti potasium dan magnesium.



Gambar 2.1 Pisang Mentah

Pisang mentah biasanya tidak dimakan langsung karena rasanya yang kurang enak, namun sering digunakan dalam masakan tertentu seperti keripik pisang atau sebagai bahan tambahan dalam hidangan yang dimasak.

2. Matang (Kuning)

Buah pisang matang memiliki ciri-ciri kulit berwarna kuning, tekstur lembut dan empuk, serta rasa manis yang khas, dengan kandungan nutrisi yang tinggi dalam gula karena sebagian besar pati telah diubah menjadi gula sederhana, juga mengandung vitamin B6, vitamin C, dan serat pangan yang baik untuk pencernaan, sering dimakan langsung sebagai camilan, digunakan dalam salad buah, atau diolah menjadi jus dan berbagai hidangan pencuci mulut.

Gambar 2.2 Pisang Matang

2.4.2 Tingkat Kematangan Buah Pepaya

Berikut adalah tingkat kematangan buah pepaya pada umumnya, diantaranya sebagai berikut[12]:

1. Mentah (Hijau)

Pepaya mentah memiliki kulit berwarna hijau dan daging buah yang keras serta tidak berasa manis. Pada tahap ini, pepaya biasanya memiliki rasa yang kurang enak dan teksturnya tidak cocok untuk dikonsumsi.

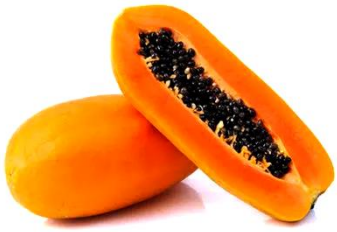
Gambar 2.3 Pepaya Mentah

Pepaya mentah mengandung enzim papain yang memiliki sifat proteolitik, membantu memecah protein dalam makanan dan mendukung pencernaan. Papain juga memiliki sifat antiinflamasi, yang bermanfaat untuk mengurangi peradangan dan mempercepat penyembuhan luka. Pepaya mentah juga kaya akan vitamin A dan C.

2. Matang (Kuning/Oranye)

Pepaya matang memiliki kulit berwarna kuning atau oranye dengan daging buah yang lembut dan manis. Buah pepaya pada tahap ini sangat ideal untuk

dikonsumsi segar, dibuat jus, atau digunakan dalam berbagai resep kuliner. Daging buah yang lembut dan aroma yang kuat menandakan kematangan optimal



Gambar 2.4 Pepaya Matang

Pepaya matang menawarkan manfaat kesehatan yang lebih besar karena tingginya kandungan beta-karoten, yang dikonversi menjadi vitamin A dalam tubuh dan mendukung kesehatan mata serta sistem kekebalan tubuh. Selain itu, pepaya matang mengandung serat yang baik untuk pencernaan, serta antioksidan yang membantu melawan radikal bebas.

2.4.3 Tingkat Kematangan Buah Nanas

Berikut adalah tingkat kematangan buah nanas pada umumnya, diantaranya sebagai berikut[13]:

1. Mentah (Hitam/Hijau)

Nanas mentah berwarna hijau atau hitam pada kulitnya, dengan daging buah yang keras dan rasanya asam atau kurang manis. Pada tahap ini biasanya terlalu keras dan tidak enak untuk dikonsumsi langsung.

Gambar 2.5 Nanas Mentah

Nanas mentah mengandung enzim bromelain, yang memiliki sifat proteolitik yang serupa dengan papain dari pepaya. Bromelain dapat membantu pencernaan protein dan memiliki efek antiinflamasi. Selain itu, nanas mentah memiliki kandungan vitamin C yang tinggi, yang berfungsi sebagai antioksidan.

2. Matang (Kuning/Oranye)

Nanas matang menunjukkan perubahan warna dari hijau ke kuning atau oranye di kulitnya. Daging buah menjadi lebih lembut dan rasanya sangat manis serta beraroma segar. Nanas pada tahap ini sangat baik untuk dimakan langsung atau digunakan dalam berbagai hidangan.

Gambar 2.6 Nanas Matang

Nanas matang memiliki rasa yang lebih manis dan mengandung lebih banyak bromelain dan vitamin C dibandingkan nanas mentah. Bromelain dalam nanas matang dapat membantu mengurangi peradangan, mempercepat penyembuhan luka, dan meredakan gangguan pencernaan. Vitamin C yang terkandung juga membantu meningkatkan sistem kekebalan tubuh dan kesehatan kulit.

2.5 Android

Android adalah sistem operasi yang dirancang khusus untuk perangkat mobile yang berbasis pada platform Linux. Awalnya, sistem operasi ini dikembangkan oleh Android Inc, namun pada tahun 2005, perusahaan tersebut diakuisisi oleh Google. Upaya pengembangan Android juga melibatkan pembentukan Open Handset Alliance (OHA) pada tahun 2007, OHA terdiri dari sejumlah perusahaan teknologi, termasuk Texas Instruments, Broadcom

Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, dan T-Mobile[14].

Selain itu android merupakan platform mobile generasi baru yang memberikan peluang bagi pengembang untuk melakukan pengembangan sesuai dengan kebutuhan. Sistem operasi yang menjadi dasar Android di lisensikan di bawah GNU General Public License Version 2 (GPLv2), yang umumnya dikenal dengan istilah Copyleft. Konsep Copyleft ini mengharuskan setiap perbaikan atau modifikasi yang dilakukan oleh pihak ketiga tetap berada di bawah lisensi yang sama[15].

2.5.1 Versi Android

Versi Android dimulai dengan merilis Android versi beta pada bulan November tahun 2007. Versi pertama yang dirilis secara komersial adalah versi Android 1.0, yang dirilis pada bulan September 2008. Sejak April 2009, versi Android dikembangkan dengan nama kode yang unik yang didasarkan pada jenis makanan pencuci mulut dan makanan manis[16].

Berikut adalah versi Android yang sudah dirilis mulai sejak tahun 2009 sampai saat ini, dapat dilihat pada Tabel 2.1 berikut.

Tabel 2. 1 Daftar Versi Android

Nama	Kode	Versi	Tanggal Rilis
Android 1.0	-	1.0	September 23, 2008
Android 1.1	Petit Four	1.1	February 9, 2009
Android Cupcake	Cupcake	1.5	April 27, 2009
Android Donut	Donut	1.6	September 15, 2009
Android Éclair	Éclair	2.0	October 27, 2009
		2.0.1	December 3, 2009
		2.1	January 11, 2010
Android Froyo	Froyo	2.2 – 2.2.3	May 20, 2010

Android Gingerbread	Gingerbread	2.3 - 2.3.2	December 6, 2010
		2.3.3 -2.3.7	February 9, 2011
Android Honeycomb	Honeycomb	3.0	February 22, 2011
		3.1	May 10, 2011
		3.2 - 3.2.6	July 15, 2011
Android Ice Cream Sandwich	Ice Cream Sandwich	4.0 – 4.0.2	October 18, 2011
		4.0.3 – 4.0.4	December 16, 2011
Android Jelly Bean	Jelly Bean	4.1 – 4.1.2	July 9, 2012
		4.2 – 4.2.2	November 13, 2012
		4.3 – 4.3.1	July 24, 2013
Android KitKat	Key Lime Pie	4.4 – 4.4.4	October 31, 2013
		4.4W – 4.4W.2	June 25, 2014
Android Lollipop	Lemon Meringue Pie	5.0 – 5.0.2	November 4, 2014
		5.1 – 5.1.1	March 2, 2015
Android Marshmallow	Macadamia Nut Cookie	6.0 – 6.0.1	October 2, 2015
Android Nougat	New York Cheesecake	7.0	August 22, 2016
		7.1 – 7.1.2	October 4, 2016
Android Oreo	Oatmeal Cookie	8.0	August 21, 2017
		8.1	December 5, 2017

Android Pie	Pistachio Ice Cream	9	August 6, 2018
Android 10	Quince Tart	8.1	December 5, 2017
Android 11	Red Velvet Cake	10	September 3, 2019
Android 12	Snow Cone	11	September 8, 2020
Android 12L	Snow Cone v2	12	October 4, 2021
Android 13	Tiramisu	13	August 15, 2022
Android 14	Upside Down Cake	14	2023

2.5.2 Android SDK (Software Development Kit)

Android SDK (Software Development Kit) adalah serangkaian perangkat pengembangan yang diberikan oleh Google untuk membantu pembuat aplikasi Android. Paket ini mencakup berbagai komponen yang diperlukan, debugger, software libraries, emulator, dokumentasi, sample code dan tutorial yang membantu dalam proses pengembangan aplikasi untuk sistem operasi Android. Ini memungkinkan para pengembang untuk memanfaatkan fitur-fitur Android yang beragam dan membuat aplikasi yang kompatibel dengan berbagai versi sistem operasi Android[17].

2.5.3 ADT(Android Development Tools)

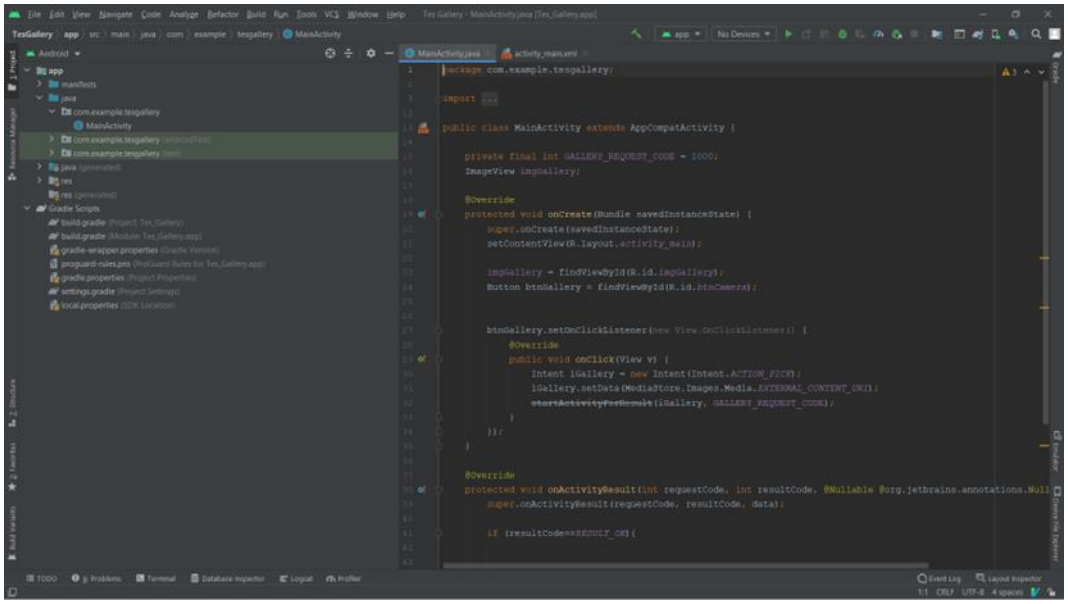
ADT (Android Development Tools) adalah sebuah plugin yang dibuat untuk menggabungkan lingkungan pengembangan Eclipse dengan Android SDK. Awalnya, ADT menjadi bagian penting dari lingkungan pengembangan Eclipse untuk membantu proses pembuatan aplikasi Android. Plugin ini memberikan fitur-fitur seperti penyunting kode yang dioptimalkan untuk bahasa pemrograman Java yang digunakan dalam pengembangan aplikasi Android, manajemen proyek yang

lebih mudah, alat visual untuk mendesain antarmuka pengguna, serta alat bantu debugging dan pengujian yang terintegrasi[18].

2.6 Android Studio

Android Studio adalah sebuah lingkungan pengembangan terpadu Integrated Development Environment (IDE) yang direkomendasikan oleh Google untuk menciptakan aplikasi Android. Software ini difokuskan pada proses pembuatan, pengujian, dan distribusi aplikasi Android, menyediakan fitur-fitur seperti penyunting kode yang cerdas, alat visual untuk merancang antarmuka pengguna, manajemen proyek yang efisien, pengelolaan sumber daya terintegrasi, serta alat bantu pengujian dan debugging yang kuat, mendukung bahasa pemrograman seperti Java dan Kotlin, dan terintegrasi dengan Android SDK untuk mengakses fungsi-fungsi dan fitur-fitur dari sistem operasi Android[19].

Berikut adalah tampilan ruang kerja dari aplikasi android studio yang dapat dilihat pada Gambar 2.6 berikut.



Gambar 2.7 Ruang Kerja Android Studio

2.7 Bahasa Pemrograman

Bahasa pemrograman (*Programming Language*) adalah sebuah instruksi standar yang mengarahkan komputer untuk melakukan tugas tertentu. Bahasa ini terdiri dari kumpulan aturan sintaksis dan semantik yang digunakan untuk

menggambarkan program komputer. Dengan menggunakan bahasa pemrograman, seorang pengembang program bisa secara spesifik menentukan jenis data yang akan dioperasikan oleh komputer, cara penyimpanan atau penyaluran data, serta langkah-langkah yang harus diambil dalam berbagai kondisi atau situasi secara detail[20].

Fungsi utama dari bahasa pemrograman yaitu memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang diinginkan. Selain itu keluaran dari kode – kode bahasa pemrograman tersebut berupa program / aplikasi. Secara umum terdapat banyak bahasa pemrograman yang dapat digunakan untuk membangun dan mengembangakan perangkat lunak, jenis-jenis dari bahasa pemrograman yang secara umum sering digunakan diantaranya Java, Visual Basic, C++, C, Cobol, PHP, Python, dan bahasa pemrograman lainnya[21].

Selain itu, bahasa pemrograman memiliki beberapa tingkatan bahasa yang dapat dikategorikan sebagai berikut :

1. Bahasa Tingkat Tinggi (*High Level Language*)

   Bahasa tingkat tinggi dikategorikan sebagai bahasa yang mendekati bahasa manusia. Contohnya bahasa Basic, Visual basic, Pascal, Java, PHP.

2. Bahasa Tingkat Menengah (*Middle Level Language*)

   Dikatakan sebagai bahasa tingkat menengah, karena bahasa tersebut dapat masuk ke dalam bahasa tingkat tinggi maupun bahasa tingkat rendah. Contohnya bahasa C.

3. Bahasa Tingkat Rendah (*Low Level Language*)

   Bahasa tingkat rendah merupakan kebalikan dari bahasa tingkat tinggi, dimana bahasa tingkat rendah ini dikategorikan sebagai bahasanya yang jauh dari bahasa manusia. Contohnya bahasa Assembly.

2.7.2 Java

Java adalah sebuah bahasa pemrograman berorientasi objek yang awalnya dikembangkan oleh *Sun Microsystems Inc.* Salah satu penggunaan yang paling signifikan dari Java adalah untuk membuat aplikasi asli (*native*) di *platform* android. Bahasa pemrograman ini memiliki sifat *multiplatform*, yang memungkinkannya digunakan di berbagai *platform* termasuk *desktop*, *android*, dan bahkan sistem operasi *linux*[22].

Java juga mencakup sebuah platform dengan *virtual machine* dan perpustakaan (*library*) yang diperlukan untuk menulis dan menjalankan program-program yang ditulis dengan bahasa *Java*. Tujuan utama di balik penciptaan bahasa pemrograman *Java* adalah keinginan untuk menciptakan bahasa yang bisa dijalankan di berbagai perangkat tanpa bergantung pada platform tertentu, menjadikannya portabel dan independen terhadap platform (*platform independent*)[23].

Gambar 2.8 Logo Java

Sumber : https://www.eteknix.com/oracle-is-killing-off-java/

Selain itu, bahasa pemrograman *java* memliki beberapa komponen penting, diantaranya adalah sebagai berikut :

1. JDK (*Java Development Kit*)

Java Development Kit (JDK) merupakan komponen inti dari Java. Komponen ini memberikan semua *tools, executables, binaries* yang diperlukan untuk menyusun, men-*debug*, dan mengeksekusi sebuah program Java.

2. JVM (*Java Virtual Machine*)

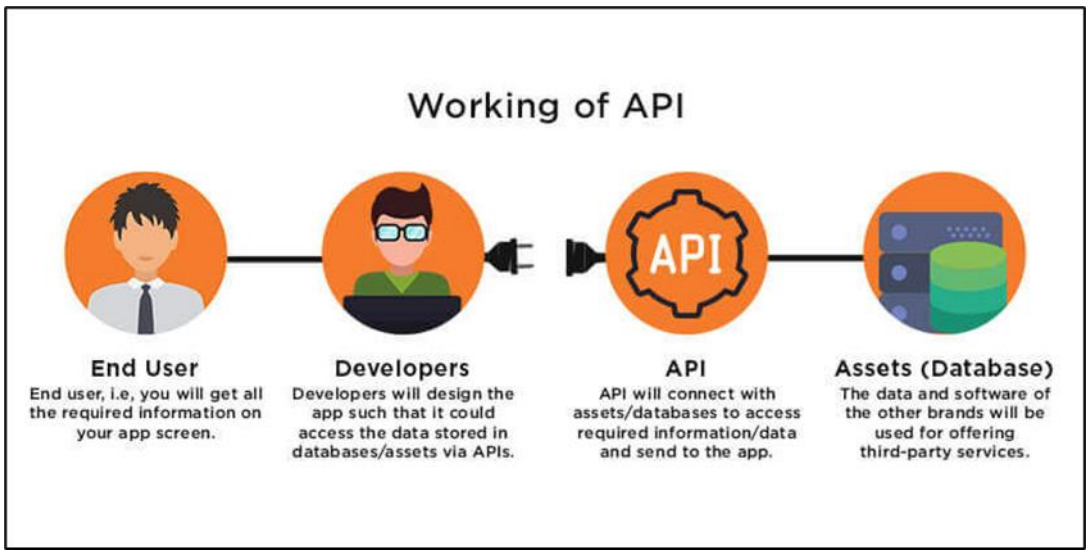
Java Virtual Machine (JVM) kerap dianggap sebagai jantung dari bahasa pemrograman Java. Ketika menjalankan program Java, JVM bertugas untuk mengonversi *byte code* menjadi kode yang lebih spesifik.

3. JRE (*Java Runtime Environment*)

Java Runtime Environment (JRE) merupakan implementasi dari JVM. JVM memberikan *platform* untuk mengeksekusi program-program dengan bahasa pemrograman *Java*.

2.8 API (Application Programming Interface)

API (*Application Programming Interface*) adalah kumpulan instruksi dan aturan yang memungkinkan perangkat lunak berbeda untuk saling berkomunikasi. *API* juga merupakan antarmuka yang dapat digunakan untuk mengakses aplikasi atau layanan dari sebuah program yang dibangun[24]. Melalui *API*, pengembang dapat membuat aplikasi yang menggunakan layanan atau fungsi dari sistem atau perangkat lain tanpa harus mengetahui detail internal dari sistem atau perangkat tersebut. Singkatnya, *API* menyediakan cara yang jelas dan terstruktur untuk berinteraksi dengan fitur-fitur yang ada dalam suatu sistem atau platform tertentu[25].



Gambar 2.9 Cara Kerja *API*

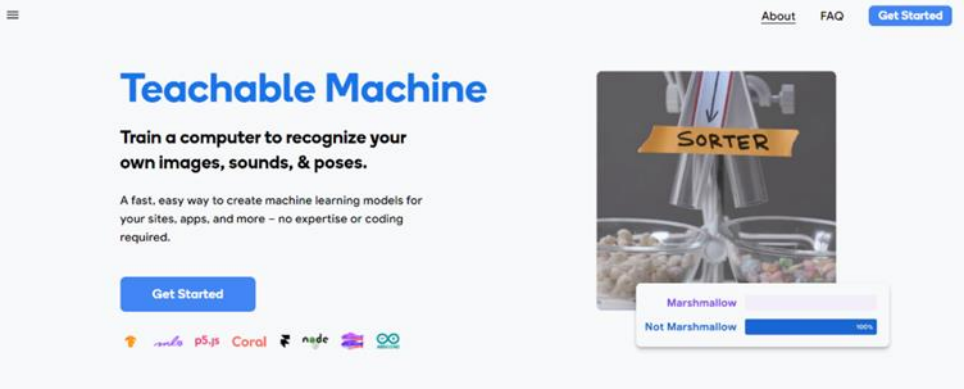
Sumber: https://www.freecodecamp.org/news/design-an-api-application-program-interface/

API bekerja dengan memberikan akses kepada aplikasi atau sistem lain untuk menggunakan perintah tertentu guna mengakses fitur atau data dan mampu mengijinkan pengguna untuk membuat *custom software* yang disediakan oleh sistem atau aplikasi yang menggunakan *API*[26]. Sistem atau aplikasi yang menggunakan *API* dapat mengirimkan permintaan ke sistem atau aplikasi sumber yang memiliki *API*, dan kemudian menerima respons berupa data yang diminta atau informasi yang diperlukan sesuai kebutuhan[27].

2.8.1 Teachable Machine

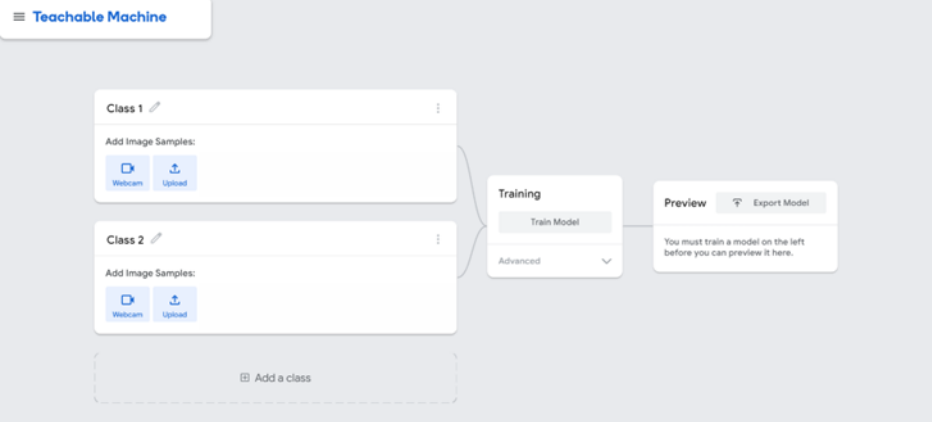
Teachable Machine merupakan alat yang dapat digunakan untuk membuat sebuah model klasifikasi yang mudah digunakan untuk mengembangkan aplikasi machine learning. Teachable Machine adalah aplikasi berbasis web yang mampu membantu membuat machine learning model secara cepat, mudah dan dapat diakses oleh semua. Alat ini memungkinkan pengguna untuk melatih model pembelajaran mesin menggunakan dataset yang mereka buat sendiri[28].

Proses untuk melakukan training data dilakukan langsung melalui website teachable machine pada link: https://teachablemachine.withgoogle.com/ , berikut adalah tampilan utama dari teachable machine yang dapat dilihat pada Gambar 2.9 berikut.



Gambar 2.10 Halaman Utama Teachable Machine

Dengan Teachable Machine, pengguna dapat mengumpulkan data dari berbagai sumber, seperti gambar dari kamera, suara dari mikrofon, atau data teks untuk melakukan training model. Setelah data diperoleh nantinya pengguna dapat melakukan untuk proses pemodelan[29]. Berikut adalah halaman untuk melakukan training model menggunakan image / gambar yang dapat dilihat pada Gambar 2.10 berikut.



Gambar 2.11 Halaman Training Model Teachable Machine

Teachable Machine memberikan cara yang intuitif bagi pengguna untuk memahami bagaimana pembelajaran mesin berlangsung dan bagaimana data yang berbeda dapat mempengaruhi model yang dihasilkan untuk mengenali atau mengklasifikasikan pola-pola tertentu.

2.8.2 TensorFlow Lite

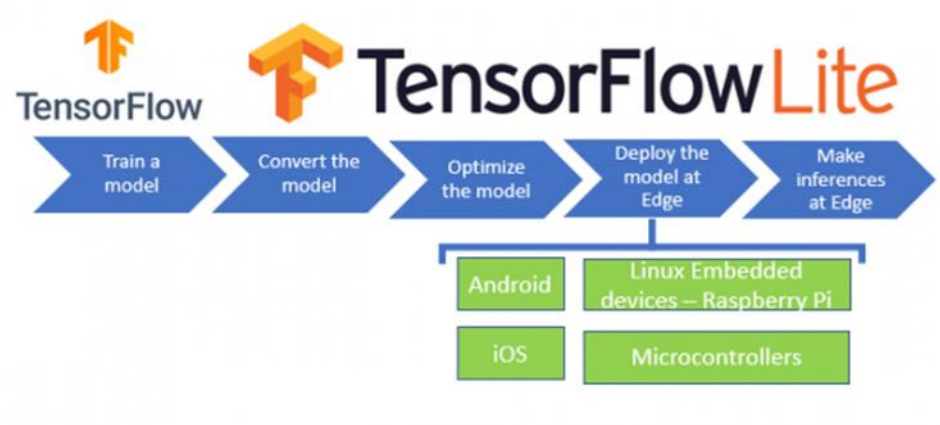
TensorFlow Lite merupakan varian dari TensorFlow yang lebih ringan, dirancang khusus untuk menjalankan model kecerdasan buatan (AI) di perangkat mobile dan embedded[30]. Fokus utamanya adalah pada performa yang cepat dan efisien, memungkinkan aplikasi untuk menggunakan model AI tanpa perlu koneksi internet yang stabil[31]. TensorFlow Lite memungkinkan penggunaan untuk menjalankan model secara lokal di perangkat seperti smartphone dan perangkat IoT dengan sumber daya terbatas. TensorFlow Lite menyediakan alat pengoptimalan dan konversi untuk memastikan model dapat dijalankan dengan baik dan efisien di perangkat yang memiliki batasan daya atau komputasi.

TensorFlow Lite digunakan untuk berbagai keperluan, termasuk integrasi dengan aplikasi mobile untuk menjalankan inferensi model machine learning langsung di perangkat pengguna, penggunaan pada perangkat edge computing seperti mikrokontroler, Raspberry Pi, dan perangkat IoT untuk menjalankan model machine learning secara lokal, optimasi model untuk mengurangi ukuran agar dapat berjalan lebih efisien di perangkat dengan sumber daya terbatas, serta pengembangan prototipe[32].

Gambar 2.12 Logo Tensorflow Lite

Sumber : https://adiandrea.id/articles/2018-05/tensor-flow-lite-android

Tensorflow Lite dikembangkan oleh *google* dan dirancang khusus untuk memungkinkan inferensi model *machine learning* (penggunaan model yang sudah dilatih) dengan cepat dan efisien di perangkat yang memiliki keterbatasan perangkat keras dan daya. Berikut adalah cara kerja atau konsep dasar tentang *tensorflow lite*:



Gambar 2.13 Cara Kerja Tensorflow Lite

Sumber : https://softscients.com/2021/05/08/tensorflow-lite-converter/

1. Pelatihan model

Tensorflow Lite memungkinkan untuk mengambil model *machine learning* yang sudah dilatih dengan *Tensorflow* (atau framework lainya) dan mengkonversinya menjadi format yang dapat dijalankan di *Tensorflow Lite.* Model ini dapat berupa model klasifikasi, deteksi objek, pengenalan suara, atau jenis model *machine learning* lainnya.

2. Konversi Model

Proses konversi model pada *Tensorflow Lite* merupakan proses penting untuk mengubah model yang sudah dilatih ke dalam format yang ringan (.tflite) sehingga

dapat dijalankan pada perangkat dengan sumber daya terbatas. Berikut adalah langkah-langkah proses konversi model ke dalam bentuk *Tensorflow Lite:*

1) Melakukan training model

Proses pertama yaitu dengan melatih model menggunakan data yang sesuai untuk dilakukan proses pelatiihan model.

2) Konversi model

Proses ini adalah langkah untuk melakukan konversi model dari *Tensorflow* ke dalam bentuk *Tensorflow Lite* (.tflite) menggunakan *framework* atau *converter* yang dapat mendukung proses konversi dalam bentuk (.tflite).

3) Validasi dan pengujian

Model *Tensorflow Lite* yang dihasilkan harus divalidasi dan diuji untuk mengetahui dan memastikan bahwa model tersebut memberikan hasil yang akurat sesuai dengan model asli yang dilakukan untuk melakukan training model.

4) Integrasikan ke Sistem atau Aplikasi

Model *Tensorflow Lite* yang telah dikonversi kemudian di integrasikan ke dalam sistem atau aplikasi yang akan menggunakannya. Integrasi ini dapat dilakukan melalui pustaka atau API yang disediakan oleh *Tensorflow Lite*.

3. Inferensi Model

TensorFlow Lite dirancang untuk menjalankan inferensi (prediksi) model pada perangkat target. Proses inferensi pada *TensorFlow Lite* adalah tahap ketika model machine learning yang telah dikonversi ke format ringan (.tflite) dijalankan di perangkat terbatas untuk melakukan tugas spesifik, seperti klasifikasi gambar, deteksi objek, atau pengenalan suara. Proses ini melibatkan untuk memberikan data input ke model, yang kemudian menghasilkan output yang diinginkan atau prediksi berdasarkan model tersebut.

4. Optimasi Kinerja

TensorFlow Lite mengimplementasikan beragam teknik optimasi guna meningkatkan kinerja model dalam melakukan inferensi pada perangkat yang memiliki keterbatasan sumber daya. Hal ini melibatkan serangkaian optimasi yang dioptimalkan khusus untuk CPU, GPU, serta penggunaan accelerator khusus seperti TPU. Upaya optimasi kinerja yang dilakukan oleh TensorFlow Lite menjadi

langkah penting untuk memastikan model machine learning dapat beroperasi secara efisien di perangkat dengan sumber daya yang terbatas.

5. Dukungan untuk Berbagai Perangkat

TensorFlow Lite dioptimalkan agar responsif dan efisien dalam penggunaan sumber daya pada perangkat seperti ponsel cerdas, sistem IoT, atau perangkat embedded lainnya. Hal ini memungkinkan model untuk dijalankan dengan performa optimal sesuai dengan keterbatasan yang ada. Dukungan ini memungkinkan para pengembang untuk mengimplementasikan model machine learning di berbagai perangkat dengan sumber daya yang berbeda, termasuk perangkat dengan keterbatasan daya komputasi dan memori.

2.9 JSON

JSON (*JavaScript Object Notation*) merupakan sebuah format yang digunakan sebagai pertukaran data yang ringan, mudah dibaca serta mudah diterjemahkan dan dibuat (*generate*) oleh manusia dan dapat mudah diproses oleh komputer[33].

JSON adalah format teks yang tidak bergantung pada bahasa pemrograman apapaun, karena menggunakan gaya bahasa yang umum digunakan oleh para programmer yaitu keluarga *C*, termasuk *C, C++, Java, JavaScript, Python* dan lain-lain. *JSON* juga digunakan secara luas dalam pengembangan RESTful *API*, aplikasi web, dan aplikasi mobile untuk menyimpan dan mentransfer atau saling bertukar data melalui jaringan[34].

2.9.1 Firebase

Firebase adalah sebuah layanan database berbasis cloud yang digunakan developer dalam pengembangan aplikasi Android, iOS dan web yang dikembangkan oleh google[35]. Firebase menyediakan berbagai fitur yang dapat digunakan oleh pengembang untuk mengembangkan dan meningkatkan kualitas dari aplikasi yang dibangun, diantaranya[36]:

1. Authentication menyediakan layanan solusi untuk proses authentikasi pengguna seperti proses login dan registrasi.

2. Cloud Storage menyediakan solusi untuk menyimpan data dan mengambil data dari server cloud.
3. Real-time Database menyediakan database yang dapat diakses secara real-time, sehingga memudahkan dalam pengembangan aplikasi yang membutuhkan data secara real-time.
4. Hosting menyediakan layanan fasilitas untuk melakukan hosting pada aplikasi web dan mobile.
5. Cloud Messaging menyediakan layanan fasilitas untuk mengirim pesan dan notifikasi pada aplikasi yang dibangun.
6. Cloud Functions menyediakan fasilitas untuk melakukan eksekusi kode server-side tanpa harus mengatur infrastruktur sendiri.

Cara kerja dari firebase ini adalah dengan menyediakan SDK (*Software Development Kit*) yang nantinya dapat di integrasikan ke dalam aplikasi, sehingga memudahkan pengembangan aplikasi untuk mengakses fitur-fitur tersebut melalui API yang disediakan oleh Firebase.

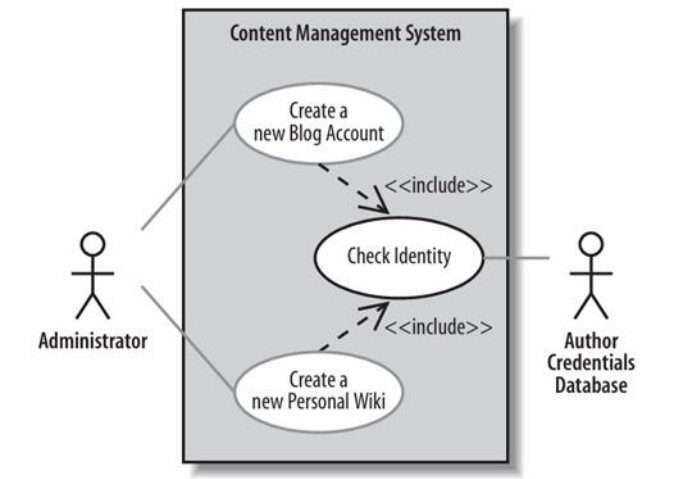
2.10 UML (Unified Modelling Language)

Unified Modeling Language (UML) adalah bahasa pemodelan standar untuk mengembangkan perangkat lunak dan sistem. UML juga merupakan salah satu materi penting dalam sebuah analisis dan perancangan sistem informasi. UML memiliki beberapa bentuk diagram yang dikelompokan dalam beberapa aspek berdasarkan pandangan yang berbeda. Sebagai bahasa pemodelan, UML menyediakan seperangkat notasi grafis yang beragam untuk menggambarkan suatu komponen dalam sistem, hubungan antara komponen, dan aliran informasi atau interaksi di antara komponen-komponen yang ada di dalamnya[37].

2.10.1 Use Case Diagram

Use case diagram adalah satu dari berbagai jenis diagram UML (Unified Modelling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor. Komponen-komponen utama dalam diagram use case meliputi aktor, use case, asosiasi, include, extend, dan hubungan generalisasi. Aktor merupakan entitas, baik manusia maupun sistem, yang terlibat dalam atau mendapatkan

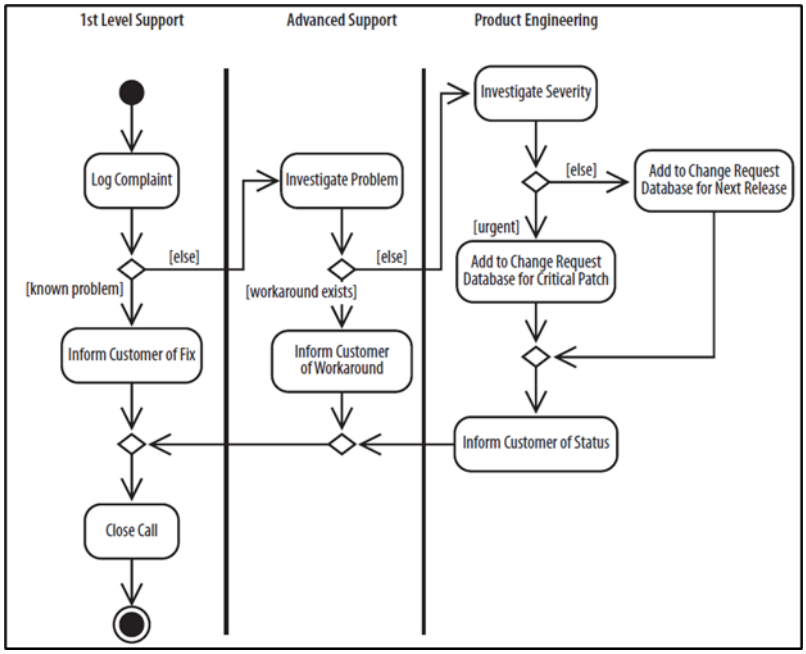
manfaat dari sistem secara eksternal. Berikut adalah contoh use case diagaram yang dapat dilihat pada Gambar 2.14 berikut.



Gambar 2.14 Contoh Use Case Diagram

2.10.2 Activity Diagram

Activity Diagram (Diagram Aktivitas) adalah jenis diagram UML yang digunakan untuk menggambarkan aliran kerja atau rangkaian aktivitas yang terjadi dalam suatu proses atau sistem. Diagram aktivitas memvisualisasikan serangkaian langkah atau aktivitas yang dilakukan oleh berbagai aktor atau komponen sistem dalam rangka mencapai tujuan tertentu. Sebuah aktivitas bisa dipasangkan dengan satu use case atau lebih. Aktivitas merepresentasikan proses yang sedang berjalan, sementara use case menjelaskan bagaimana pelaku menggunakan sistem untuk melakukan aktivitas tersebut. Adapun contoh dari activity diagram dapat dilihat pada Gambar 2.15 berikut.



Gambar 2. 15 Contoh Activity Diagram

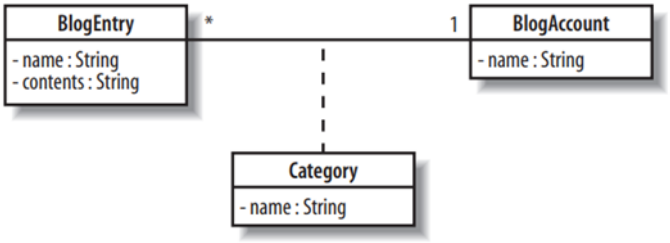
Adapun dalam alur *activity* diagram terdapat beberapa elemen yang dapat dijelaskan sebagai berikut:

1. *Initial Node* adalah sebuah lingkaran yang memiliki fungsi untuk menunjukan bahwa sebuah aktifitas sudah dimulai.
2. *Actions* adalah sebuah persegi yang memiliki fungsi untuk menunjukan sebuah tampilan, proses komputasi, dan sebuah langkah penting dalam proses tersebut.
3. *Path* adalah sebuah jalur yang dimana arah panah menunjukan kemana langkah selanjutnya akan dilakukan dalam proses tersebut.
4. *Decision a*dalah sebuah gambar berlian yang berfungsi sebagai penkondisian yang menganalogikan *if-else statement* dalam program. Hasil dari pengkodisian biasanya diberi label dengan *boolean.*
5. *Final Node a*dalah sebuah lingkaran didalam lingkaran yang memiliki fungsi untuk menunjukan bahwa proses tersebut sudah selesai.

2.10.3 Class Diagram

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class diagram dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat

langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time. Adapun contoh dari class diagram dapat dilihat pada Gambar 2.16 berikut.



Gambar 2.16 Contoh Class Diagram

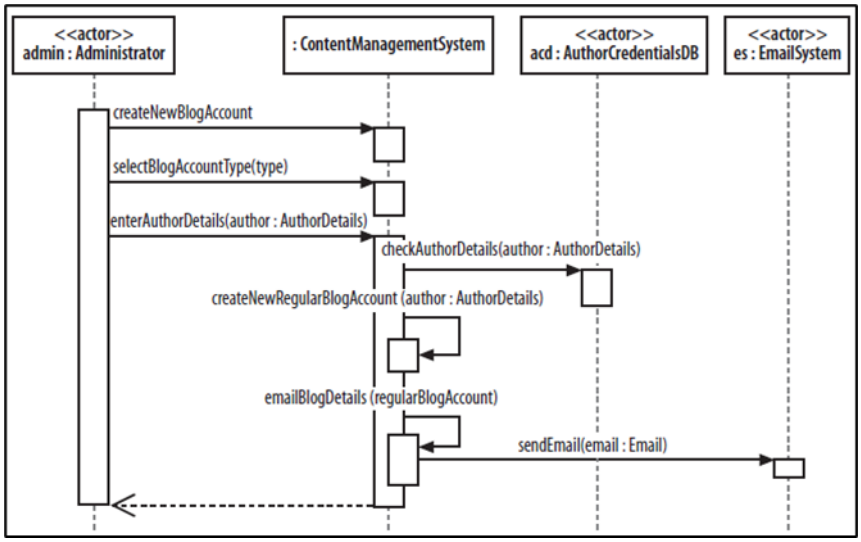
Didalam *class diagram* terdapat beberapa elemen yang perlu diperhatikan, elemen-elemen tersebut dapat dijelaskan sebagai berikut:

1. *Class* adalah sebuah *blueprint* dari objek yang dapat dibangun kembali dan merepresentasikan objek tersebut.
2. *Visibility* adalah bagaimana caranya sebuah kelas dapat memperlihatkan isi kelasnya ke kelas lain. Terdapat 4 tipe dari *visibility* yaitu *public*, *protected*, *package*, dan *private.*
3. *Class State*: *Attributes* adalah informasi yang merepresentasikan keadaan dari sebuah objek. *Attributes* biasanya disimpan didalam kelas atau dapat disimpan di garis relasi asosiasi dengan kelas lainnya. Terdapat beberapa elemen dalam *attributes* seperti nama, tipe, dan *visibility*.
4. *Class Behavior: Operations* adalah sebuah deskripsi mengenai apa yang kelas tersebut bisa lakukan, namun tidak menjelaskan bagaimana proses tersebut akan dilakukan. Sama seperti *attributes*, *operations* juga memiliki elemen seperti nama, tipe, *visibility*, dan *parameter*.

2.10.4 Sequence Diagram

Sequence Diagram (Diagram Urutan) adalah salah satu jenis diagram dalam UML (Unified Modeling Language) yang menggambarkan interaksi antara objek dalam suatu skenario atau proses tertentu. Diagram ini memvisualisasikan urutan pesan atau panggilan yang dikirim antara objek-objek dalam sistem, menunjukkan bagaimana objek-objek berinteraksi satu sama lain dan waktu terjadinya interaksi.

Sequence Diagram berkaitan dengan menangkap urutan interaksi antara bagian-bagian sistem. Adapun contoh dari sequence diagram dapat dilihat pada Gambar 2.17 berikut.



Gambar 2.17 Contoh Sequence Diagram

Dalam *sequence diagram* terdapat beberapa elemen yang perlu diperhatikan, berikut adalah penjelasannya:

1. *Participants* adalah bagian dari sistem yang berinteraksi didalam diagram tersebut.
2. *Lifeline* adalah keadaan yang menyatakan bahwa bagian tertentu hanya aktif di titik tersebut.
3. *Activation Bars* adalah sebuah indikator yang dimana akan mengaktifkan *participants* ketika menerima sebuah pesan.
4. *Message* adalah sebuah pesan yang dikirimkan oleh *participants* ke *participants* lainnya yang menyebabkan interaksi antar objek.

Fragment adalah sebuah daerah yang berada didalam *sequence diagram* yang memiliki fungsi untuk melanjutkan interaksi diantara objek.