

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Aplikasi**

Pengertian aplikasi berdasarkan istilahnya yaitu suatu program yang dibuat untuk menjalankan suatu fungsi yang telah ditentukannya, dimana program tersebut telah siap untuk digunakan oleh pengguna jasa aplikasi maupun oleh pengguna aplikasi lain [4]. Sementara menurut Deni Darmawan (2006) pengertian dari aplikasi itu sendiri adalah sebuah data – data ataupun instruksi – instruksi yang mampu memberikan respon berupa pengambilan keputusan yang berbasis komputer [5].

##### **2.1.1. Fungsi Aplikasi**

Aplikasi memiliki peran penting di berbagai bidang dalam kehidupan sehari hari, fungsi aplikasi di berbagai bidang antara lain :

###### **1. Pendidikan**

Aplikasi dapat dipakai sebagai bahan pengajaran. Contohnya dapat memberikan penyajian materi yang lebih menarik karena dilengkapi dengan menggunakan slide presentasi yang dapat dibuat di Microsoft Powerpoint.

###### **2. Kedokteran**

Aplikasi dapat memiliki peran penting juga di bidang kedokteran seperti adanya aplikasi Halodoc yang dapat memberikan kemudahan untuk konsultasi dengan dokter tanpa perlu datang ke rumah sakit.

###### **3. Bisnis**

Aplikasi dapat membantu seseorang atau sebuah perusahaan dalam mengolah bisnisnya, karena dengan menggunakan aplikasi dapat lebih mudah dalam mengatur masalah finansial perusahaan atau hal lainnya.

###### **4. Ilmu Pengetahuan**

Aplikasi dapat memberikan kemudahan dalam mendapatkan ilmu pengetahuan karena adanya aplikasi *search engine* atau mesin pencari seperti Google, Bing, dan yang lain sebagainya, yang dapat memberikan berbagai informasi.

## 5. Militer

Militer juga sudah mulai memanfaatkan aplikasi untuk memaksimalkan hasil yang akan didapat. Contohnya seperti sudah ada beberapa pesawat yang dapat dikontrol dari jarak jauh menggunakan aplikasi.

### 2.1.2. Jenis Jenis Aplikasi

Aplikasi memiliki beberapa jenis, antara lain :

#### 1. Aplikasi Office

Aplikasi Office atau Aplikasi perkantoran merupakan aplikasi yang paling umum yang memiliki fungsi untuk melakukan pemrosesan kata dan pemrosesan tabel. Pemrosesan kata yang dilakukan digunakan untuk menggantikan tugas pengetikan yang biasa dilakukan menggunakan alat tertentu. Dengan bantuan aplikasi pemrosesan kata tidak hanya bisa melakukan pengetikan akan tetapi dapat melakukan pengaturan spasi, margin (jarak pinggir kertas), jenis huruf (*font*), bahkan sampai melakukan pengecekan terhadap kata atau kalimat yang diketik. Sementara untuk pemrosesan tabel biasanya digunakan untuk melakukan perhitungan yang menggunakan tabel. Berbagai proses perhitungan dapat dilakukan seperti perjumlahan, pengurangan, bahkan dapat juga melakukan pencarian nilai seperti nilai rata – rata, nilai maksimal, nilai minimal, dan yang lain sebagainya. Aplikasi Office yang paling sering digunakan adalah *Microsoft Word*, *Microsoft Excel*, dan yang lain sebagainya.

#### 2. Aplikasi Multimedia

Aplikasi Multimedia merupakan salah satu yang paling sering digunakan dan memiliki keberagaman yang banyak. Hal ini karena aplikasi multimedia merupakan sebuah aplikasi yang menggabungkan antara penggunaan berbagai macam media seperti media teks, gambar, suara maupun film. Aplikasi multimedia pada umumnya terbagi menjadi 2 yaitu aplikasi multimedia yang dapat membuat dan aplikasi yang hanya dapat melihat saja. Contoh aplikasi multimedia adalah GOM Player.

#### 3. Aplikasi Internet

Aplikasi Internet merupakan suatu protokol yang berfungsi untuk menghubungkan antara satu orang atau satu pihak tertentu dengan orang lain atau pihak lain tanpa dibatasi oleh jarak. Karena fungsi inilah pada umumnya aplikasi internet digunakan sebagai alat komunikasi yang dapat menggantikan alat komunikasi yang terbatas oleh jarak. Aplikasi internet menyangkut dua sisi, seperti sisi penyedia atau sisi server dan sisi pengguna atau biasa disebut sebagai client. Beberapa contoh aplikasi internet adalah Facebook, Instagram, dan yang lain sebagainya.

## **2.2 Android**

Android merupakan salah satu sistem operasi yang menggunakan linux sebagai basisnya, yang dapat digunakan pada perangkat telepon seluler (*mobile*), seperti telepon pintar (*smartphone*) dan juga komputer tablet (PDA). Android memiliki sifat yang terbuka atau *open source* yang berarti android memberikan kebebasan terhadap para pengembang untuk mengembangkan aplikasinya sendiri [6].

### **2.2.1. Sejarah Android**

Android pertama kali dikembangkan oleh Andy Rubin, Rich Miner, Chris White dan Nick Sears yang merupakan pendiri dari Android, Inc itu sendiri pada tahun 2003. Pada awal didirikannya perusahaan ini bertempat di Palo Alto, California, kemudian perusahaan ini pun berhasil mencuri perhatian perusahaan besar lainnya yaitu Google yang kemudian Google pun membeli perusahaan mereka tersebut pada tanggal 17 Agustus 2005.

Ketika Android sudah berada dibawah Google pada tahun 2005, awalnya mereka tidak langsung melahirkan sebuah inovasi baru, hingga akhirnya pada tanggal 22 Oktober 2008 HTC Dream sebagai ponsel seluler komersial pertama yang menggunakan Android pun diluncurkan. 2 Tahun berikutnya Google kembali merilis ponsel pintar lain yang menggunakan teknologi Android yaitu Nexus One dengan bantuan HTC selama proses

pembuatannya, yang kemudian Android pun terus berkembang hingga pada tahun 2019 Android sudah memiliki berbagai versi yaitu [6]:

4. Android versi 1.0 (Alpha), 1.1(Beta), 1.5 (Cupcake), 1.6 (Donut)

Android versi 1.0 dengan kode nama Alpha merupakan android pertama yang dirilis pada tanggal 5 november 2007 dan pada 23 September 2008 dirilis versi komersialnya. Kemudian pada tanggal 9 November 2009 Android mengembangkan lagi versinya ke versi 1.1 dengan kode nama Beta, lalu pada tanggal 30 April 2009 Android dikembangkan lagi ke versi 1.5 dengan kode nama Cupcake dan terakhir untuk versi 1.6 dengan kode nama Donut dikembangkan pada tanggal 15 September 2009.

5. Android Versi 2.0 – 2.1 (Eclair)

Tak hanya sampai dengan versi 1.6, android kembali mengembangkan versinya ke versi 2.0 sampai dengan versi 2.1 dengan kode nama Eclair pada tanggal 9 Desember 2009.

6. Android Versi 2.2 (Froyo)

Pada Tanggal 20 Mei 2010, Android kembali mengembangkan versinya ke versi 2.2 dengan kode nama Froyo, yang memiliki level API di angka 8.

7. Android Versi 2.3 – 2.3.7 (Gingerbread)

Versi 2.3 sampai dengan versi 2.3.7 dinamai dengan kode nama Gingerbread, dimana untuk versi ini pertama kali dirilis pada tanggal 6 Desember 2010 untuk versi 2.3 – 2.3.2 yang memiliki level API di angka 9, dan versi 2.3.3 – 2.3.7 dirilis pada tanggal 9 Februari 2011 dengan level API 10.

8. Android Versi 3.0 – 3.2 (Honeycomb)

Android Versi 3.0 sampai dengan versi 3.2 memiliki kode nama Honeycomb, dan memiliki 2 tanggal perilsan yaitu untuk versi 3.0 – 3.1 dirilis pada tanggal 10 Mei 2011 dengan level API 12, dan untuk versi 3.2 dirilis pada tanggal 15 Juli 2011 dengan level API 13.

9. Android Versi 4.0 (Ice Cream Sandwich)

Android Versi 4.0 atau bisa disebut Ice Cream Sandwich memiliki level API 15 dan juga 2 versi yaitu 4.0.3 sampai dengan 4.0.4 yang dirilis pada tanggal 16 Desember 2011.

10. Android Versi 4.1 – 4.3 (Jelly Bean)

Android Jelly Bean merupakan sebutan untuk android dari versi 4.1 sampai dengan versi 4.3, untuk versi 4.1.x dirilis pada tanggal 9 Juli 2012 dengan level API 16, versi 4.2.x dirilis pada tanggal 13 November 2012 dengan level API 17 dan terakhir untuk versi 4.3.x dirilis pada tanggal 24 Juli 2013 dengan level API 18.

11. Android Versi 4.4 (KitKat)

Selang beberapa bulan setelah perilisasi versi 4.3.x, Android kembali merilis versi terbarunya yaitu versi 4.4 dengan kode nama KitKat, yang dirilis pada tanggal 31 Oktober 2013, dengan level API di angka 19.

12. Android Versi 5.0 (Lollipop)

Setelah merilis versi 4.4, Android mengembangkan kembali versi terbarunya yaitu dengan merilis versi 5.0 dan versi 5.1 yang diberi nama Lollipop, versi 5.0 dirilis pada tanggal 15 Oktober 2014 dengan level API 21, sementara untuk versi 5.1 dirilis pada tanggal 9 November 2015 dengan level API 22.

13. Android Versi 6.0 (Marshmallow)

Setelah kurang lebih 3 bulan dari perilisasi versi Android 5.1, Android kembali merilis versi terbarunya yaitu tepat pada tanggal 19 Agustus 2015 lahirlah Android versi 6.0 dengan kode nama Marshmallow yang memiliki level Api 23.

14. Android Versi 7.0 (Nougat)

Versi 7.0 yang diberi nama Nougat ini dirilis pada tanggal 22 Agustus 2016, versi ini memiliki level API 24.

15. Android Versi 8.0 (Oreo)

Ciri khas Android adalah memberikan nama versi mereka dengan nama makanan yang memiliki huruf awal yaitu huruf selanjutnya dari versi sebelumnya, jika versi sebelumnya adalah (N)ougat maka berikutnya adalah

huruf setelah N yaitu O, maka dari itu dinamakan lah versi ke 8 dari android ini dengan nama Oreo yang dirilis pada tanggal 21 Agustus 2017 dengan level API 26.

#### 16. Android Versi 9.0 (Pie)

Masih dengan ciri khas penamaan versinya, pada versi 9.0 Android kembali menamai versinya dengan nama makanan yaitu Pie, yang dirilis pada tanggal 6 Agustus 2018 dengan level API 28.

#### 17. Android Versi 10

Setelah kurang lebih 9 tahun Android memberikan ciri khas penamaan versinya dengan menggunakan nama makanan, pada versi 10 ini Android memberikan penamaan yang berbeda yaitu dengan angka sesuai dengan versinya, sehingga untuk versi 10 diberi nama Android 10. Versi ini dirilis pada tanggal 3 September 2019 yang memiliki Level API 29.

### 2.2.2. Android SDK

Android SDK merupakan singkatan dari *Android Standart Development Kit* yang berfungsi sebagai penyedia perlengkapan dan juga sebagai *Application Prgoramming Interface (API)* yang diperlukan untuk mengembangkan aplikasi Android, yang mana Android SDK sendiri menggunakan bahasa Java sebagai bahasa pemrogramannya [7]. Terdapat beberapa komponen yang terdapat di Android SDK, antara lain [8] :

1. *Debugger*
2. *Libraries*
3. *Handset Emulator*
4. Dokumentasi
5. Kode contoh
6. Tutorial.

### 2.2.3. Android JDK

Android JDK merupakan singkatan dari *Android Java Development Kit* dan sesuai namanya, Android JDK adalah sebuah API yang digunakan pada

bahasa pemrograman Java yang terdiri dari *Java Runtime Environment* (JRE) dan didalam JRE sendiri terdapat *Java Virtual Machine* (JVM) [8].

### 2.3 Android Studio

Android Studio merupakan salah satu IDE (*Integrated Development Environment*) yang dapat digunakan untuk melakukan pengembangan aplikasi Android. Android Studio pertama kali di rilis pada tanggal 16 Mei 2013 oleh perusahaan pemilik Android yaitu Google, sehingga menjadikan Android Studio IDE resmi untuk pengembangan aplikasi Android. Tujuan Google merilis aplikasi ini adalah untuk menggantikan aplikasi pengembang Android pendahulunya yaitu aplikasi *Eclipse* dan penggantian ini diresmikan pada cara Google “*I/O Conferences*”. Android Studio merupakan sebuah aplikasi yang memiliki sifat *open source* yang berarti Android Studio dapat digunakan oleh pengembang secara gratis dibawah lisensi *Apache 2.0* [8].

Dalam pengembangannya, Android Studio menggunakan IntelliJ IDEA yang mirip dengan aplikasi *Eclipse* disertai dengan tambahan ADT (*Android Development Tools*) Plugin. Beberapa fitur yang dimiliki Android Studio antara lain [9] :

1. Android Studio membuat projek yang dibangun berbasis pada *Gradle Build*.
2. Android Studio dapat memberikan *Refactory* dan pembenahan terhadap bug yang ditemukan dengan cepat.
3. Android Studio memiliki sebuah tools yang bernama “*Lint*”, tools ini memiliki fungsi yang dapat memonitor kecepatan, kegunaan dan juga kompetibelitas terhadap aplikasi yang dikembangkan dengan cepat.
4. Android Studio memiliki dukungan terhadap Proguard dan App-signing yang dapat memberikan keamanan aplikasi, serta memiliki dukungan terhadap *Google Cloud Platform* untuk setiap aplikasi yang sedang dikembangkan.
5. Android Studio memiliki GUI aplikasi *Android* yang lebih mudah.

### 2.4 Java

Pengertian Java dalam ilmu adalah sebuah sistem pemrograman yang diperkenalkan pada tahun 1995 dan berorientasikan objek yang diperkenalkan oleh sebuah perusahaan bernama Sun Microsystem Inc. Yang ketika itu dipimpin oleh James Gosling. Sebelum resmi diperkenalkan oleh Sun Microsystem pada tahun 1995, Java bermula dari sebuah proyek untuk menciptakan sebuah bahasa pemrograman yang memungkinkan semua *device* dapat menjalankannya tanpa harus terikat oleh suatu *platform* yang digunakan oleh *device* tersebut. Proyek ini dipelopori oleh James dan 3 orang rekannya yaitu Patrick Naughton, Mike Sheridan dan Bill Joy pada tahun 1991 dan menghasilkan sebuah bahasa pemrograman bernama “Oak” yang kemudian berganti nama menjadi Java [8].

Sebagai sebuah bahasa pemrograman, Java memiliki sintaks dan juga aturan pemrogramannya sendiri. Java juga dapat menjadi *platform* yang memiliki *virtual machine* dan *library* yang dapat digunakan untuk menulis serta menjalankan aplikasi yang dibangun menggunakan bahasa pemrograman Java [8]. Terdapat beberapa keunggulan lain yang dimiliki Java dibanding bahasa pemrograman lainnya, antara lain [10]:

1. Java memiliki sifat portable dan platform independent yang berarti Java dapat dijalankan diberbagai platform asalkan JVM tersedia untuk platform yang dituju tersebut.
2. Java memiliki kemampuan untuk melakukan pengelolaan memori yang lebih baik karena dapat melakukan dealokasi memori secara otomatis.
3. Java memiliki kemampuan untuk mengubah program Java menjadi byte-code sebelum dijalankan oleh suatu mesin tertentu, hal ini membuat java dapat berjalan diberbagai arsitektur prosesor ataupun di berbagai sistem operasi.
4. Java memiliki keunggulan yang membuat java dapat tangguh jika terjadi masalah serius pada program, dengan cara memeriksa seluruh akses sistem yang dilakukan program. Jika ditemukan sebuah kesalahan serius, maka Java akan membuat pengecualian atau disebut *exception* yang dapat



ditangani atau dikelola oleh program tanpa menimbulkan risiko kemacetan sistem.

5. Keamanan pada Java juga menjadi salah satu keunggulan dibanding bahasa pemrograman lainnya. Java tidak mendukung pointer yang berarti program tidak akan mendapatkan akses ke area sistem, sehingga dapat mejamin tidak ada virus yang menumpang pada program saat program sedang dijalankan.
6. Java dapat menggunakan metode *native code* yang berarti dapat ditulis dalam bahasa lain, seperti C++.

## 2.5 Sistem Rekomendasi

Sistem rekomendasi merupakan sebuah perangkat lunak (*software*) dan juga sistem yang digunakan untuk menyediakan rekomendasi atau saran untuk suatu barang tertentu untuk digunakan oleh pengguna. Rekomendasi yang diberikan berhubungan dengan berbagai macam hal yang melakukan penentuan keputusan seperti menentukan barang apa yang akan dibeli, musik apa yang akan di dengar, berita apa yang akan dibaca dan juga salah satunya menentukan resep makanan apa yang cocok dengan bahan yang ada. Sistem rekomendasi biasanya digunakan untuk membantu seseorang untuk menentukan suatu hal yang kurang dikuasainya karena kurangnya pengalaman atau kurang kompeten dalam hal tersebut, dan untuk menghitung jumlah kemungkinan yang mungkin ditawarkan agar tidak berlebihan [11].

Terdapat berbagai macam metode atau cara untuk mendapatkan rekomendasi, dua metode yang paling sering digunakan adalah *Collaborative filtering* dan *Content-based Recommendations*[12]. *Collaborative filtering* merupakan metode sistem rekomendasi yang merekomendasikan *item* berdasarkan setiap *user* yang memiliki kesukaan yang sama. Contoh kasus yang dapat menggunakan *Collaborative filtering* adalah diketahui terdapat dua orang yang suka bermain game bernama Joni dan Jono, Joni diketahui baru saja bermain game *Mobile Legends : Bang Bang* sementara Jono belum bermain game ini, tapi karena sistem mengetahui bahwa Jono memiliki kesamaan selera

dengan Joni yaitu suka bermain game, maka game *Mobile Legends : Bang Bang* tersebut akan direkomendasikan kepada Joni.

Sementara itu, berbeda dengan *Collaborative Filtering*, *Content-based Recommendations* melakukan rekomendasi berdasarkan *metadata* atau atribut dari setiap *item*. Contoh dari metode *Content-based recommendations* adalah jika Tono suka menonton sebuah serial kartun yang bernama *Family Guy*, dan kartun ini memiliki beberapa *metadata* berupa genre seperti “*Animated*“, “*Comedy*“, “*Adult*” dan misalkan Tono menonton serial kartun tersebut di sebuah aplikasi streaming video bernama *Netflix*, maka sistem dari aplikasi *Netflix* tersebut akan melakukan rekomendasi serial kartun lain yang memiliki *metadata* berupa *genre* yang sama seperti serial kartun *Family Guy* yang baru ditonton Tono tersebut misalnya serial kartun *The Simpsons* atau yang lainnya.

Metode *Content-based Recommendations* memiliki beberapa metode perhitungan untuk menentukan rekomendasi yang sesuai, salah satunya adalah metode *Jaccard Similarity*. Metode *Jaccard Similarity* merupakan metode yang dapat melakukan perhitungan tingkat kemiripan dari dua set data dengan menghitung jumlah *item* yang sama dari dua set data tersebut dan melakukan pembagian dengan jumlah *item* yang unik dari keduanya dan hasil pembagiannya dikalikan 100[12]. Singkatnya, metode ini merupakan rasio dari jumlah *item* yang dibagikan dari dua set dengan jumlah *item* yang berpotensi dibagikan dari dua set.

Dalam istilah matematika formal, *Jaccard Similarity* antara dua buah himpunan, A dan B adalah sebuah kardinalitas, atau jumlah elemen pada perpotongan antara himpunan A dan himpunan B dibagi dengan kardinalitas gabungan himpunan A dan himpunan B. *Jaccard Similarity* memiliki formula sebagai berikut :

$$\text{Jaccard Index} = \frac{(\text{Jumlah atribut yang sama dari kedua buah item})}{(\text{Jumlah atribut yang sama \& yang tidak sama dari kedua buah item})} * 100$$

Atau dapat juga ditulis dengan notasi sebagai berikut :

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} * 100$$

Berdasarkan rumus tersebut dapat ditentukan untuk tahapan perhitungan dari *Jaccard Index* sendiri akan menjadi sebagai berikut :

1. Hitung jumlah atribut yang sama pada dua buah item yang dibandingkan.
2. Hitung jumlah atribut yang sama dan yang tidak sama pada dua buah item yang dibandingkan.
3. Lakukan operasi pembagian terhadap angka yang ditemukan dari tahap (1) dengan angka yang ditemukan dari tahap (2).
4. Lakukan operasi perkalian dari angka yang ditemukan pada langkah (3) dengan angka 100.

Hasil dari metode *Jaccard Similarity* adalah dalam bentuk persentase, yang berarti jika seluruh atribut yang dimiliki dari dua buah himpunan atau set data yang dibandingkan adalah sama, maka tingkat *Jaccard Similarity* nya adalah 100% yang berarti data tersebut yang paling mirip dan jika tidak sampai 100% maka yang paling mendekati dari 100% lah yang paling memiliki tingkat similaritas tertinggi. Sebaliknya, jika kedua himpunan tidak memiliki atribut yang sama, maka tingkat *Jaccard Similarity*-nya adalah 0% dan jika angkanya 50% maka kedua data tersebut memiliki setengah atribut yang sama.

## **2.6 Image Recognition**

Prinsip dari Image Recognition adalah sebuah penggabungan teknologi dan matematika untuk melakukan *preprocess* untuk mendapatkan target berupa informasi gambar dan ekstrak fitur, kemudian *classifier* melakukan pengenalan terhadap gambar lalu dikategorikan menurut kategorinya yang sesuai, dan akhirnya hasil pengenalan (*recognition*) pun di terima berdasarkan kecocokan dari informasi yang disimpan [13].

Image Recognition diciptakan terinspirasi dari kemampuan mata manusia untuk dapat mengenali gambar atau *image* yang luar biasa. Image Recognition telah dikembangkan selama bertahun – tahun agar dapat “menggantikan” mata manusia dan juga otak manusia. Image Recognition menggunakan komputer untuk melakukan simulasi bagaimana proses manusia melakukan analisa dan mengerti terhadap target gambar yang dilihatnya. Dalam pengembangan teknologi ini selama 20 tahun terakhir, objek yang dapat

dikenali oleh Image Recognition semakin bertambah dari yang awalnya hanya sekedar tulisan berupa huruf atau angka hingga sampai dapat mengenali wajah manusia, lingkungan, ataupun objek lainnya [13].

Proses yang terjadi untuk melakukan Image Recognition umumnya terdiri dari 5 tahapan, diantaranya [13]:

1. *Acquisition of image information* (pengambilan informasi dari gambar)
2. *Image Preprocessing* (pra-pemrosesan gambar)
3. *Image Feature Extraction & Selection* (ekstraksi dan seleksi fitur gambar)
4. *Classifier Design* (penggolongan design)
5. *Decision Matching* (penyocokan keputusan)

## **2.7 Makanan**

Makanan merupakan salah satu kebutuhan pokok setiap manusia karena makanan dapat memberikan sumber tenaga, pertumbuhan tubuh, serta melindungi tubuh dari penyakit. Menurut Depkes RI (2003), makanan merupakan segala bahan yang telah diolah sehingga dapat dimakan oleh manusia kecuali air dan obat – obatan.

### **2.7.1. Bahan Makanan**

Pengertian bahan makanan menurut Harsono Haryosutomo dalam Sri Maryati (2000) merupakan bahan yang dihasilkan dari berbagai bidang seperti pertanian, perkebunan, peternakan dan juga dari teknologi makanan. Sementara menurut Isa Ramli, bahan makanan adalah bahan makanan yang masih segar, dan bahan masih belum tercampur dengan bahan pengawet apapun serta mengandung semua unsur gizi. Sementara menurut Suharjo (1986), bahan makanan adalah bahan – bahan yang dapat dikonsumsi setiap harinya yang berfungsi untuk memenuhi kebutuhan bagi pemeliharaan tubuh, pertumbuhan tubuh dan juga penggantian jaringan – jaringan yang rusak pada tubuh.

Dari beberapa pengertian tersebut, dapat disimpulkan bahwa bahan makanan adalah ilmu yang mempelajari tentang bahan – bahan hasil dari pertanian, peternakan, perikanan dan teknologi makanan yang berguna untuk memenuhi kebutuhan hidup manusia [14].

### 2.7.2. Resep Makanan

Pengertian resep adalah sekumpulan instruksi yang berisikan petunjuk untuk membuat suatu hidangan atau makanan dari bahan yang siap diolah. Petunjuk yang diberikan pada resep harus secara seksama dan tepat dari segala aspek seperti dari jumlah bahan, cara – cara atau langkah – langkahnya, pengolahan dan juga prosedur kerja untuk suatu hidangan, agar orang yang menggunakan resep tersebut dapat melakukan hal yang sama seperti yang diinginkan oleh resep tersebut.

Resep memiliki standarnya sendiri yang dimana informasi dimasukkan sebanyak mungkin, struktur untuk standar resep tersebut adalah sebagai berikut :

1. Nama resep.
2. Hasil (Total hasil, jumlah porsi, ukuran porsi).
3. Bahan dan jumlah yang diperlukannya.
4. Peralatan yang dibutuhkan.
5. Petunjuk mengolah hidangan, penyimpanan atau perawatan sederhana.
6. Petunjuk tentang porsi, piring atau alat – alat hidang serta hiasan – hiasan.
7. Petunjuk tentang tertib kerja, membereskan atau merapihkan alat – alat yang telah dipakai juga merapihkan tempat kerja dan menyimpan sisa – sisa bahan makanan.

## 2.8 API

*Application Programming Interface* atau yang biasa disebut dengan API adalah sebuah dokumentasi yang digunakan untuk pembangunan sebuah perangkat lunak (*software*), biasanya API terdiri dari beberapa komponen seperti interface, fungsi, kelas, struktur dan lain sebagainya. Programmer dapat memanfaatkan API dengan baik karena fungsi API sendiri yang memberikan kemudahan pada pekerjaan seorang programmer, seperti membantu dalam “membongkar” suatu software, atau bisa sebagai penyambung dengan perangkat lunak lain. Karena hal inilah API bisa dikatakan sebagai sebuah

jembatan penghubung dari 1 aplikasi ke aplikasi yang lainnya dan memungkinkan seorang programmer menggunakan sistem function , yang mana proses ini dikelola melalui sistem operasi [15].

### 2.8.1. API Clarifai

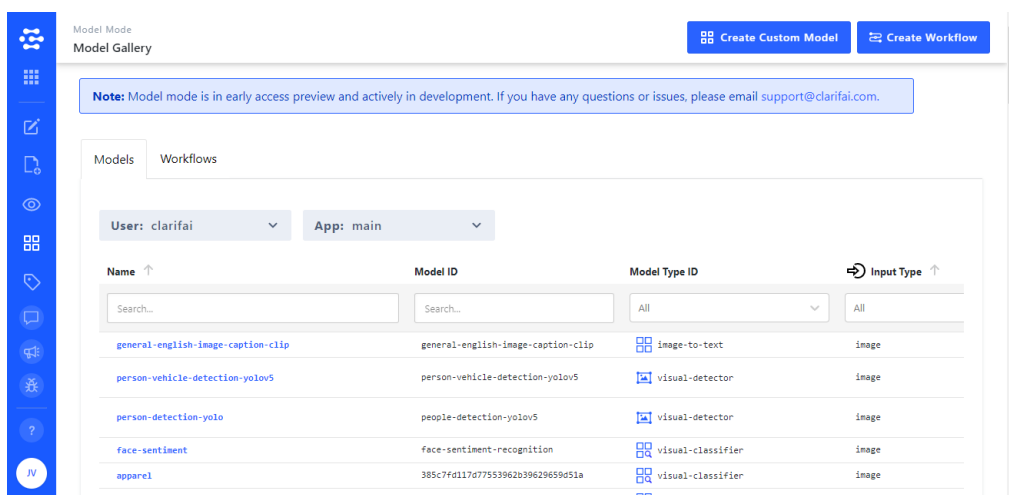
Clarifai API adalah API yang membantu membangun software yang menggunakan AI (*Artificial Intelligence*) khususnya untuk dapat mempelajari model gambar, video, teks dan data suara yang tidak terstruktur, dengan melakukan klasifikasi objek (*object classification*), deteksi (*detection*), pelacakan (*tracking*), pencarian visual (*visual search*), geolokasi (*geolocation*) dan pemrosesan bahasa alami (*natural language processing*) [16]. Clarifai sendiri diciptakan oleh seorang ahli yang sudah terkemuka di dalam bidang *machine learning* yang bernama Matthew Zeiler pada tahun 2013, dan sukses memimpin pasar sejak menjuari kompetisi yang bertema *Image Classification* yaitu kompetisi ImageNet 2013 dan pada tahun 2019 dinobatkan sebagai pemimpin dalam laporan Forrester's New Wave Computer Vision Platforms, satu – satunya *startup* yang menerima peringkat berbeda.

Clarifai merupakan bagian dari *Computer Vision* yang berarti clarifai menggunakan *pre-trained model* atau model yang sudah dilatih untuk dapat melakukan *deep learning* yang digunakan juga untuk melakukan pengenalan terhadap gambar, video, teks, dan audio data yang bersifat tidak terstruktur. Clarifai menyediakan fitur untuk *user* agar dapat mengembangkan atau melatih modelnya sendiri dan juga menyediakan fitur untuk menggunakan model yang sudah dikembangkan atau dilatih oleh clarifai. Untuk melakukan akses yang menghubungkan model tersebut dengan aplikasi yang diinginkan, clarifai menyediakan client dari beberapa bahasa pemrograman yang cukup populer, seperti Python, Java, NodeJS, C#, PHP, Swift, Rust, Go, dan C++. Client – client yang disediakan clarifai ini dibangun dengan menggunakan gRPC dan dapat di akses melalui channel HTTP+JSON dan juga dapat melalui channel gRPC itu sendiri.

### 2.8.1.1. Clarifai Model

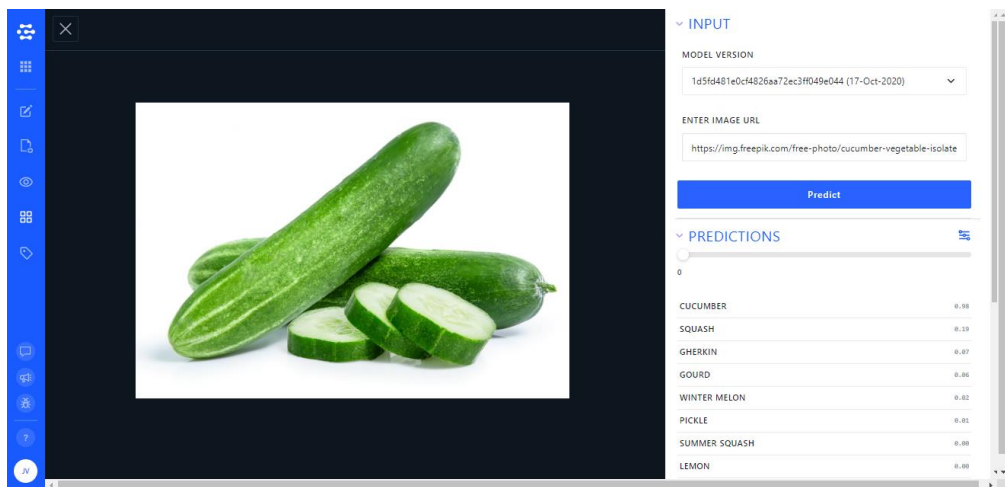
Clarifai menggunakan model untuk melakukan prediksi dan model yang terdapat pada clarifai memiliki 2 jenis yaitu *Pre-Built Clarifai Models* dan *Operator Models*. *Operator Models* merupakan model yang bersifat “tidak dapat dilatih (*non-trainable*)” atau yang memiliki “fungsi tetap (*fixed function*)” yang dapat membantu menghubungkan, mengarahkan, dan juga mengontrol *input* atau masukan dan *outputs* atau keluaran yang dikirim melalui *workflows*. *Pre-Built Clarifai Model* merupakan model *machine learning* yang bersifat “dapat dilatih” dan telah dikembangkan secara internal serta telah diuji secara menyeluruh. Model – model pada clarifai juga terbagi menjadi berbagai tipe yang dapat digunakan seperti *Context-Based Classifier*, *Clusterer*, *Concept Thresholder*, *Region Thresholder*, *Concept Synonym Mapper*, *Annotation Writer*, *Image Cropper*, *Random Sampler*, *KNN Classifier*, *Object Counter*, *Landmarks Align*, *Isolation Operator*, *Tesseract Operator*, *Text Aggregation Operator*, *Barcode Operator*, *Visual Detector*, *Text Classifier*, *Visual Classifier*, dan *Visual Embedder*.

Untuk mengetahui model apa saja yang telah disediakan oleh clarifai, pengguna dapat terlebih dahulu melakukan akses ke portal clarifai dan melakukan login atau *register* jika belum memiliki akun. Setelah berhasil masuk ke portal clarifai, pengguna membuat *application* baru kemudian masuk ke menu model mode seperti yang dapat dilihat pada gambar berikut.



Gambar 2.1 Model Mode Clarifai

Pada menu ini akan menampilkan list model yang dimiliki oleh Clarifai beserta dengan model id, tipe model tersebut, dan tipe masukan model tersebut. Selain itu pengguna juga dapat melihat model yang dilatihnya sendiri pada menu ini dengan cara menekan tombol *drop down list* lalu memilih user yang ingin ditampilkan modelnya. Pengguna dapat menguji model tersebut dengan cara memilih salah 1 model dan sebagai contoh akan menggunakan model “food” dengan id “food-item-recognition” kemudian akan diarahkan ke halaman detail model tersebut dan terdapat 3 menu pada halaman tersebut yaitu *versions*, *concept* dan *predict*, pengguna dapat menekan tombol *predict* lalu memilih versi model dan memasukkan gambar bahan makanan yang akan dideteksi lalu menekan tombol *predict* dan model akan melakukan prediksi dan kemudian menampilkan hasil prediksi seperti yang dapat dilihat pada gambar berikut.



Gambar 2.2 Contoh Penggunaan Model Clarifai

### 2.8.1.2. Clarifai Client

Clarifai Client merupakan *client* yang disediakan oleh Clarifai untuk membantu pengguna agar dapat menghubungkan Clarifai API dengan aplikasinya. Clarifai client dibangun dengan menggunakan *channel* gRPC yang sudah terenkripsi beserta dengan *channel* HTTPS+JSON *channel* untuk membuat *request*. Berdasarkan dokumentasi yang diberikan Clarifai pada situs resminya, Clarifai memilih menggunakan gRPC karena beberapa alasan, salah satunya adalah karena gRPC dapat memberikan layanan mikro ringan yang



berarti dapat memberikan latensi yang rendah dengan komunikasi throughput tinggi dapat memberikan layanan *machine learning* berkinerja tinggi dimana pun pelanggan berada, perbandingan lebih jelas mengapa clarifai lebih memilih gRPC dibanding HTTP+JSON dapat dilihat pada gambar berikut.

| Capability      | gRPC                           | HTTP APIs with JSON          |
|-----------------|--------------------------------|------------------------------|
| Protocol        | HTTP/2                         | HTTP                         |
| Payload         | Protobuf (small, binary)       | JSON (large, human readable) |
| Streaming       | Client, server, bi-directional | Client, server               |
| Browser support | No                             | Yes                          |

**Gambar 2.3 Perbandingan gRPC Clarifai dengan HTTP API**

Clarifai Client tersedia dalam berbagai bahasa pemrograman yang populer dan salah satunya adalah bahasa pemrograman java yang dapat digunakan pada aplikasi android. Hal yang harus disiapkan pengguna sebelum melakukan instalasi Clarifai Client pada aplikasinya adalah menyiapkan *Authorization Keys* yang dapat diperoleh dari API Key atau pun PAT. Perbedaan antara penggunaan API Key dan PAT (*Personal Access Token*) adalah API Key terhubung dengan aplikasi sehingga dapat ditemukan pada menu *application* pada portal clarifai sementara PAT terhubung dengan akun pengguna sehingga dapat ditemukan pada info akun pengguna pada portal clarifai dan karena seorang pengguna dapat memiliki beberapa *application*, maka dengan menggunakan PAT dapat lebih berguna, akan tetapi jika menggunakan PAT berarti pengguna harus menentukan ID *application* mana yang harus dijalankan *request*-nya.

Setelah memiliki *Authorization Keys*, langkah berikutnya untuk menginstall Clarifai Client adalah dengan melakukan instalasi library pada android studio dengan cara memasukkan *code* dibawah ke file *build.gradle* untuk level *module* dan letakkan dibagian dependencies.

```
dependencies {
    implementation 'com.clarifai:clarifai-grpc:8.0.0'
    implementation 'com.clarifai:clarifai-
api2:core:2.8.1'
}
```

Setelah memasukkan code tersebut, pastikan juga pada bagian repositories pada file *build.gradle* untuk level *project* telah terdapat *jcenter()* dan kemudian jalankan sinkronisasi dan jika berhasil tanpa memunculkan *error* maka library *clarifai* telah terinstall pada aplikasi. Selanjutnya adalah melakukan inisialisasi *client* dengan langkah pertama adalah melakukan import pada activity dengan memasukkan *code* berikut.

```
import com.clarifai.channel.ClarifaiChannel;
import
com.clarifai.credentials.ClarifaiCallCredentials;
import com.clarifai.grpc.api.*;
```

Kode diatas akan melakukan import terhadap Clarifai Channel, Clarifai Call Credentials dan Clarifai gRPC. Selanjutnya adalah membuat *channel* dari gRPC Clarifai Channel serta masukkan ClarifaiCallCredentials yang diisi dengan menggunakan *Authorization Keys* seperti kode dibawah ini.

```
Channel channel =
ClarifaiChannel.INSTANCE.getGrpcChannel();

V2Grpc.V2BlockingStub stub =
V2Grpc.newBlockingStub(channel)
    .withCallCredentials(new
ClarifaiCallCredentials("{YOUR_CLARIFAI_API_KEY}"));
```

Setelah memasukkan kode diatas maka Clarifai Client berhasil dipasang pada aplikasi dan Clarifai telah siap untuk digunakan.

### 2.8.2. API Masak Apa

API Masak Apa merupakan API penyedia resep makanan Indonesia dan tersedia dalam bahasa Indonesia, resep makanan yang didapat dari API ini berasal dari sebuah website resep makanan Indonesia yaitu ‘masakapahariini’, namun API ini bukan API official dari website tersebut. API ini dibuat oleh seorang pengguna github bernama Reski Arianto dengan initial commit atau commit pertama yang dilakukan pada tanggal 12 Oktober 2020 di akun github miliknya yang bernama ‘tomorisakura’. API ini dibangun dengan memanfaatkan teknologi Cheerio dan juga Node js, dan sampai sekarang masih

terus dilakukan maintenance atau pemeliharaan software oleh tomorisakura dan rekan – rekannya yaitu Ammar Alifian Fahdan dengan nama akun github yaoitu ‘kerupuksambel’ dan juga Akwan Maroso dengan nama akun github ‘akwanmaroso’. Penggunaan API “Masak Apa” adalah dengan menggunakan *endpoint usage* yang digabung dengan *base url* serta beberapa parameter sesuai kebutuhan yang diinginkan dan dipanggil dengan menggunakan method GET untuk mendapatkan informasi yang dibutuhkan dari API “Masak Apa”, dimana data keluaran dari API “Masak Apa” adalah data json. API “Masak Apa” dapat digunakan pada aplikasi android dengan bantuan library retrofit.

### 2.8.2.1. Dokumentasi API Masak Apa

API Masak Apa memiliki beberapa field yang dapat digunakan sebagai paramater pada endpoint usage, beberapa parameter tersebut antara lain adalah sebagai berikut :

#### 1. *Key*

*Key* digunakan sebagai *unique key* yang berfungsi untuk dapat melakukan akses ke *endpoint* berikutnya. Ciri khas dari *key* biasanya mengambil dari nama resep atau judul resep seperti contohnya : "resep-terong-mekar-crispy", "resep-sop-iga-sapi", "resep-sukun-goreng-teri-balado", dan yang lain sebagainya.

#### 2. *Page*

*Page* digunakan untuk menampilkan data berikutnya jika aplikasi yang dibangun menggunakan penomoran halaman. *Page* berisi nomor yang terdiri dari angka yang tidak terbatas yang berarti halaman pada API Masak Apa tidak memiliki batasan.

#### 3. *Tag*

*Tag* merupakan field yang memiliki kemiripan dengan *key*, perbedaan pada *tag* adalah *tag* digunakan sebagai *unique key* pada artikel kategori untuk menampilkan detail dari artikel tersebut. Contoh *tag* yaitu “makanan-gaya-hidup”.

#### 4. *Limit*

*Limit* merupakan field yang berfungsi untuk menentukan batasan dari hasil yang dimunculkan pada json, misalnya untuk membatasi hanya 5 resep yang muncul. Field *limit* dapat diisi dari angka 1 sampai dengan 10, yang berarti limit hasil yang dimunculkan adalah mulai dari 1 hasil sampai dengan 10 hasil.

### 2.8.2.2. Endpoint Usage API Masak Apa

Endpoint Usage digunakan untuk mengakses fitur – fitur yang terdapat pada API Masak Apa, dan digunakan atau diletakkan pada bagian akhir dari Base Url. API Masak Apa memiliki 1 Base Url dan 11 Endpoint Usage yaitu sebagai berikut :

#### 1. Base Url

Base Url merupakan url yang digunakan sebagai dasar untuk memanggil API Masak Apa dan juga untuk menggunakan Endpoint Usage yang akan diletakkan pada bagian akhir dari Base Url itu sendiri. Base Url dari API Masak Apa adalah <https://masak-apa.tomorisakura.vercel.app>.

#### 2. New Recipes

New Recipes merupakan endpoint pada API Masak Apa yang berfungsi untuk menampilkan list resep secara general atau secara umum dari API Masak Apa. List yang ditampilkan secara *default* akan menampilkan 10 resep yang berbeda – beda setiap kali mengakses endpoint ini. Usage pada endpoint ini adalah “/api/recipes” sehingga jika digabungkan dengan base url akan menjadi “<https://masak-apa.tomorisakura.vercel.app/api/recipes>”.

#### 3. New Recipes by Page

New Recipes by Page merupakan endpoint pada API Masak Apa yang berfungsi untuk menampilkan list daftar resep masakan berdasarkan halaman dimana setiap halaman terdiri dari 10 resep masakan. Endpoint New Recipes by Page memiliki sifat yang sama dengan New Recipes yaitu resep akan terus berubah setiap kali halaman di refresh. Usage pada endpoint ini adalah “/api/recipes/:page”, dengan ketentuan “:page” dapat

diganti dengan angka sehingga jika digabungkan dengan base url akan menjadi “<https://masak-apa.tomorisakura.vercel.app/api/recipes/1>” .

#### 4. New Recipes Limit

New Recipes Limit merupakan endpoint pada API Masak Apa yang berfungsi untuk mengatur jumlah resep yang dimunculkan pada list daftar resep. Usage pada endpoint ini adalah “[/api/recipes-length/?limit=size](https://masak-apa.tomorisakura.vercel.app/api/recipes-length/?limit=size)”, dengan ketentuan “**size**” dapat diganti dengan angka dengan batasan 1 sampai 10 yang berarti batasan jumlah resep yang dimunculkan adalah paling sedikit 1 dan paling banyak 10 sehingga jika digabungkan dengan base url akan menjadi “<https://masak-apa.tomorisakura.vercel.app/api/recipes-length/?limit=5>”.

#### 5. Recipes by Category

Recipes by Category merupakan endpoint pada API Masak Apa yang berfungsi untuk menampilkan list resep berdasarkan kategorinya. Usage pada endpoint ini adalah “[/api/category/recipes/:key](https://masak-apa.tomorisakura.vercel.app/api/category/recipes/:key)”, dengan ketentuan “**:key**” dapat diganti dengan *key* dari sebuah kategori seperti salah satu contohnya adalah “masakan-hari-*raya*”, sehingga jika digabungkan dengan base url akan menjadi “[https://masak-apa.tomorisakura.vercel.app/api/category/recipes/masakan-hari-\*raya\*](https://masak-apa.tomorisakura.vercel.app/api/category/recipes/masakan-hari-<i>raya</i>)”.

#### 6. Recipes Category

Recipes Category merupakan endpoint pada API Masak Apa yang berfungsi untuk menampilkan list kategori resep yang tersedia dimana terdapat 10 kategori resep dan memunculkan key dari tiap kategori tersebut untuk kemudian dapat digunakan pada Recipes by Category. Usage pada endpoint ini adalah “[/api/category/recipes](https://masak-apa.tomorisakura.vercel.app/api/category/recipes)” sehingga jika digabungkan dengan base url akan menjadi “<https://masak-apa.tomorisakura.vercel.app/api/category/recipes>”.

#### 7. Recipe Detail

Recipe Detail merupakan endpoint pada API Masak Apa yang berfungsi untuk menampilkan detail dari suatu resep yang dipilih. Usage pada

endpoint ini adalah “**/api/recipe/:key**” dengan ketentuan “**:key**” dapat diganti dengan *key* resep yang diinginkan, sehingga jika digabungkan dengan base url akan menjadi “<https://masak-apa.tomorisakura.vercel.app/api/recipe/resep-oseng-daging-kentang-masak-kecap>”.

#### 8. Search Recipes

Search Recipes merupakan endpoint pada API Masak Apa yang berfungsi untuk melakukan pencarian resep masakan sesuai dengan keyword yang ditentukan. Usage pada endpoint ini adalah “**/api/search/?q=parameter**” dengan ketentuan “**parameter**” diganti dengan keyword yang diinginkan, sehingga jika digabungkan dengan base url akan menjadi “<https://masak-apa.tomorisakura.vercel.app/api/search/?q=soto>”.

#### 9. Article Categorys

Article Cateogry merupakan endpoint pada API Masak Apa yang berfungsi untuk menampilkan list kategori artikel dimana terdapat 3 kategori artikel dan memunculkan key untuk dari tiap kategori tersebut untuk kemudian dapat digunakan pada Article by Category. Usage pada endpoint ini adalah “**/api/category/article**”, sehingga jika digabungkan dengan base url akan menjadi “<https://masak-apa.tomorisakura.vercel.app/api/category/article>”.

#### 10. Article by Category

Article by Category merupakan endpoint pada API Masak Apa yang berfungsi untuk menampilkan list artikel berdasarkan kategori serta memunculkan key dari setiap artikel untuk kemudian digunakan pada artichel detail. Usage pada endpoint ini adalah “**/api/category/article/:key**” dengan ketentuan “**:key**” dapat diganti dengan key dari kategori artikel yang diinginkan, sehingga jika digabungkan dengan base url akan menjadi “<https://masak-apa.tomorisakura.vercel.app/api/category/article/makanan-gaya-hidup>”.

#### 11. Article

Article merupakan endpoint pada API Masak Apa yang berfungsi untuk menampilkan artikel yang tersedia. Usage pada endpoint ini adalah “/api/articles/new”, sehingga jika digabungkan dengan base url akan menjadi [“https://masak-apa-tomorisakura.vercel.app/api/articles/new”](https://masak-apa-tomorisakura.vercel.app/api/articles/new).

## 12. Article Detail

Article Detail merupakan endpoint pada API Masak Apa yang berfungsi untuk menampilkan detail dari sebuah artikel yang diinginkan. Usage pada endpoint ini adalah “/api/article/:tag/:key”, dengan ketentuan “:tag” dapat diganti dengan kategori dari artikel tersebut dan “:key” dapat diganti dengan menggunakan unique key dari artikel tersebut, sehingga jika digabungkan dengan base url akan menjadi [“https://masak-apa-tomorisakura.vercel.app/api/article/makanan-gaya-hidup/papeda-dan-masakan-indonesia-timur”](https://masak-apa-tomorisakura.vercel.app/api/article/makanan-gaya-hidup/papeda-dan-masakan-indonesia-timur)

### 2.8.3. API Youtube

API Youtube merupakan API penyedia video – video dari website Youtube itu sendiri, yang dapat digunakan oleh para pengembang yang ingin mengembangkan sebuah aplikasi yang memerlukan video – video dari Youtube. API ini bersifat publik yang berarti Youtube sebagai penyedia API ini memberikan akses kepada pengguna API untuk dapat mengolah data maupun layanan yang ada di website tersebut.

#### 2.8.3.1. Instalasi API Youtube

Untuk melakukan instalasi API Youtube ke dalam aplikasi yang akan dikembangkan, dibutuhkan sebuah API Key yang dapat ditemukan pada google cloud platform, beberapa langkah yang untuk mendapatkan API Key dan melakukan instalasi API Youtube adalah sebagai berikut.

1. Membuat projek baru dengan masuk ke halaman google cloud platform yaitu pada *link* <https://console.cloud.google.com/>, kemudian menekan tombol *create new project* dan membuat *project* dengan nama sesuai dengan ketentuan yang diinginkan.

Project name \*  
Kasep

Project ID: kasep-357715. It cannot be changed later. [EDIT](#)

Location \*  
No organization [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

Gambar 2.4 Project Baru untuk Youtube API

- Langkah berikutnya adalah melakukan pengaktifan pada API Youtube Data V3 yaitu dengan masuk ke menu APIs and Services, kemudian pada kolom pencarian masukkan *keyword* “API Youtube Data V3” lalu pilih API Youtube Data V3 pada hasil pencarian kemudian tekan tombol *enable*.
- Selanjutnya adalah membuat kredensial untuk mendapatkan API Key yang akan digunakan pada Aplikasi dengan cara masuk ke sub menu *credentials* pada menu APIS and Services, kemudian menekan tombol *create credentials* lalu tekan tombol API Key dan salin API Key tersebut agar dapat digunakan pada aplikasi.

## API key 1

Use this key in your application by passing it with the `key=API_KEY` parameter.

Your API key

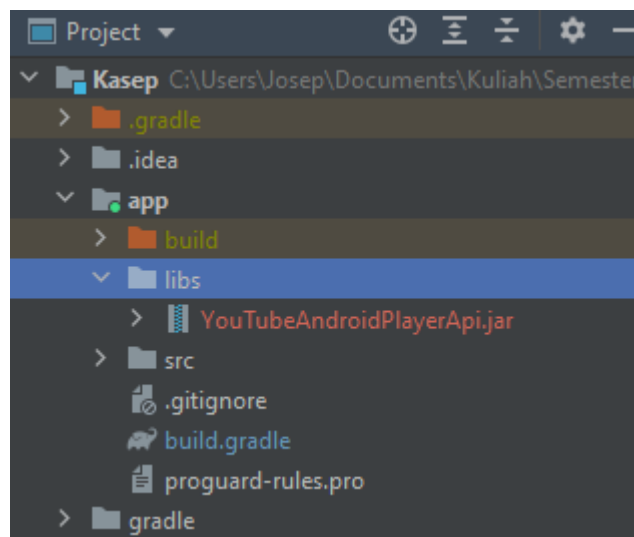
**⚠** This key is unrestricted. To prevent unauthorized use, we recommend restricting where and for which APIs it can be used. [Edit API key](#) to add restrictions. [Learn more](#)

[CLOSE](#)

Gambar 2.5 API Key pada Youtube API



4. Setelah berhasil mendapatkan API Key, langkah berikutnya adalah melakukan instalasi API pada aplikasi yang dikembangkan, dan pada contoh instalasi dilakukan pada aplikasi android. Instalasi dilakukan dengan melakukan *download* API terlebih dahulu pada *link* berikut <https://developers.google.com/youtube/android/player/downloads>, setelah *file* tersebut berhasil di-*download*, berikutnya adalah melakukan *extract* kemudian salin *file* dengan ekstensi .jar pada folder dengan nama "libs" dan tempel pada folder ....\nama-projek\app\libs.
5. Selanjutnya adalah dengan memasukkan file tersebut sebagai *library* dengan cara tekan kanan pada file dengan ekstensi .jar tersebut dan klik *add as library* dan tunggu hingga proses selesai, dan hasilnya akan seperti pada gambar dibawah ini yang berarti API Youtube telah terinstall pada aplikasi.



Gambar 2.6 YoutubeAndroidPlayerAPI.jar

#### 2.8.4. ML Kit Text Translation API

ML Kit merupakan *mobile SDK* yang dikembangkan oleh google untuk menghadirkan keahlian *Machine Learning* pada aplikasi *mobile* berbasis *android* dan *iOS*. ML Kit memberikan kemampuan *Vision* dan *Natural Language API* yang kuat namun tetap mudah digunakan oleh penggunanya yang bertujuan untuk membantu menyelesaikan berbagai kesulitan atau rintangan pada pengembangan aplikasi dan agar dapat menciptakan pengalaman penggunaan aplikasi yang baru untuk pengguna. ML Kit

sepenuhnya menggunakan model *Machine Learning* Google yang terbaik dikelasnya dan dapat digunakan oleh pengguna tanpa harus mengeluarkan uang atau bersifat gratis. ML Kit API juga bersifat “*all run on-device*” yang berarti dapat digunakan pada kasus yang terjadi secara langsung seperti digunakan pada saat *live stream* dan juga berarti bahwa ML Kit API dapat digunakan secara *offline* tanpa memerlukan koneksi internet.

ML Kit API awalnya merupakan bagian dari Firebase yang sebelumnya diberi nama “ML Kit for Firebase”, namun pada tanggal 3 Juni 2020 ML Kit memutuskan untuk memisahkan diri dari Firebase agar dapat fokus membangun API yang bersifat *on-device*[17]. Akan tetapi, Firebase tidak sepenuhnya melepas ML Kit, Firebase masih memiliki *Machine Learning* SDKnya karena Firebase membagi ML Kit menjadi 2, yang pertama adalah ML Kit sebagai produk baru yang berdiri sendiri dan diberi nama “ML kit” yang berisi *on-device* API serta produk kedua adalah *Firebase Machine Learning* yang berfokus pada *cloud-based* API dan penyebaran model khusus. Selain itu pada produk ML Kit yang baru tersebut juga menggabungkan *Google Mobile Vision* sehingga pada ML Kit terdapat 2 buah jenis API yang dapat digunakan yaitu Vision API dan Natural Language API. Vision API terdiri dari *Barcode Scanning, Face Detection, Image Labeling, Object Detection and Tracking, Text Recognition, Digital Ink Recognition, Pose Detection*, dan *Selfi Segmentation*, Sementara untuk Natural Language API terdiri dari *Language ID, On-Device Translation, Smart Reply*, dan *Entity Extraction*.

Text Translation API merupakan salah satu bagian dari Natural Language API yang berfungsi untuk melakukan terjemahan antara 1 bahasa ke bahasa yang lainnya secara *on-device* yang berarti aplikasi harus melakukan proses pengunduhan “kamus” terlebih dahulu. Text Translation API mampu melakukan terjemahan bahasa terhadap lebih dari 50 bahasa, namun tidak semua bahasa tersebut harus dilakukan pengunduhan terlebih dahulu, pengembang aplikasi dapat mengatur atau membatasi bahasa apa saja yang harus dilakukan pengunduhan terlebih dahulu pada aplikasi yang dikembangkannya tersebut. Kemudian untuk melakukan peng-*install*-an library

pada aplikasi android, cukup dengan memasukkan kode berikut ke bagian *dependencies* pada *file build.gradle* untuk level *module* dan lakukan sinkronasi maka ML Kit Text Translation API siap untuk digunakan pada aplikasi.

```
dependencies {  
    // ...  
  
    implementation 'com.google.mlkit:translate:17.0.0'  
}
```

## 2.9 UML (Unified Modelling Language)

UML atau *Unified Modeling Language* merupakan sebuah teknik yang digunakan untuk memodelkan suatu sistem. UML ditemukan oleh Grady Booch, Ivar Jacobson, dan James Rumbaugh pada akhir tahun 1980-an yang diawali dari bahasa *object-oriented*. Pada saat itu, Grady Booch memperkenalkan metode Booch yang kemudian metode ini digabungkan dengan teknik pemodelan objek yang diciptakan oleh James Rumbaugh sementara Ivan Jacobson berperan sebagai yang memperkenalkan prinsip dan metodologi dari UML.

Berkat kerjasama dari ketiganya, pada tahun 1997 sebuah metodologi yang bernama UML versi 1 resmi diperkenalkan hingga sampai saat ini dan terus berkembang sampai dengan versi 2.5 yang dirilis pada bulan Mei tahun 2015 silam. Pada versinya yang ke 2.5 ini, UML sudah memiliki lima belas buah diagram, yang dimana diagram – diagram tersebut dibagi lagi menjadi 2 kelompok yaitu *structure diagram* dan *behavior diagram*[18] :

### 1. *Structure Diagram*.

*Structure Diagram* digunakan untuk menggambarkan data dan hubungan yang bersifat statis dalam suatu sistem informasi. Diagram – diagram yang terdapat pada kelompok ini adalah *package*, *object*, *component*, *class*, *deployment*, *composite structure*, dan *profile diagram*.

### 2. *Behavior Diagram*.

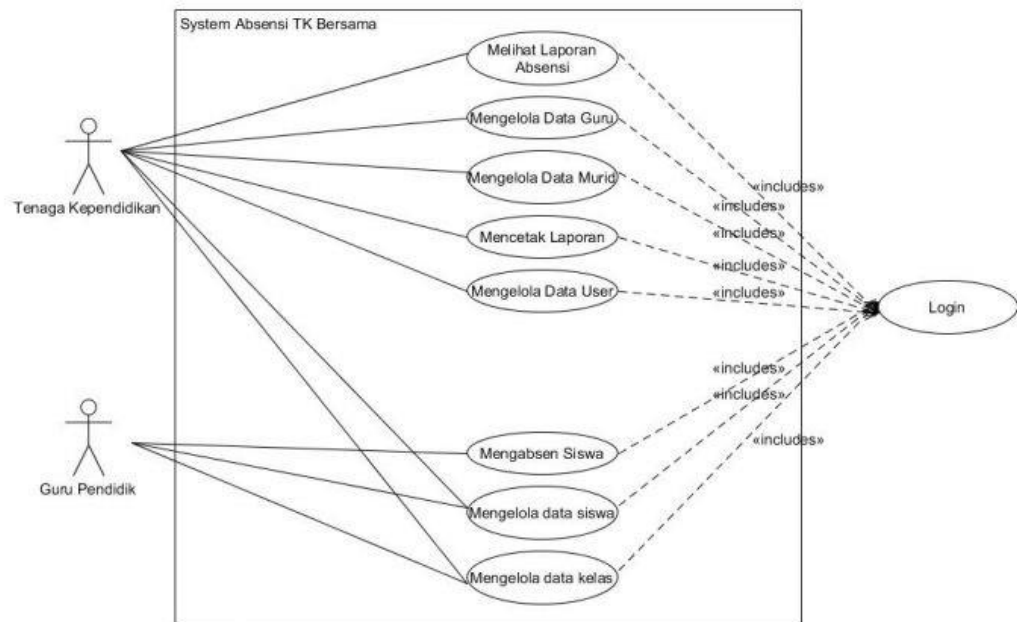
*Behavior Diagram* adalah kebalikan dari *Structure Diagram* dimana diagram jenis ini digunakan untuk menggambarkan hubungan dinamis diantara objek yang mewakili sistem informasi bisnis. Diagram – diagram yang terdapat pada kelompok ini adalah *sequence*, *timing*, *interaction overview*, *activity*, *use case*, *protocol state machine*, *communication*, dan *behavior state machine diagram*.

### 2.9.1. Use Case Diagram

Use Case Diagram adalah salah satu diagram yang termasuk kedalam *Behavior Diagram*, dimana diagram ini dapat digunakan untuk memodelkan aspek perilaku sistem. Use Case Diagram dapat digunakan untuk memvisualkan, menspesifikasikan dan juga mendokumentasikan kebutuhan perilaku sistem. Tujuan utama dari pemodelan use case adalah [19] :

1. Untuk membantu memutuskan dan juga mendeskripsikan kebutuhan – kebutuhan fungsional yang ada pada sistem yang dibangun.
2. Untuk memberikan deskripsi yang jelas dan konsisten dari apa yang seharusnya dilakukan, sehingga model use case dapat digunakan didalam keseluruhan pengembangan sistem dan sebagai acuan agar sistem dapat berjalan sesuai dengan fungsionalitas yang dimodelkan di usecase.
3. Untuk menyediakan basis dalam melakukan pengujian sistem untuk menguji apakah sistem telah berjalan sesuai dengan fungsionalitas yang diminta.
4. Untuk menyediakan kemampuan untuk melacak kebutuhan fungsionalitas sistem menjadi kelas – kelas dan operasi – operasi aktual di sistem.

Contoh *Use Case Diagram* dapat dilihat pada **Gambar 2.7**.



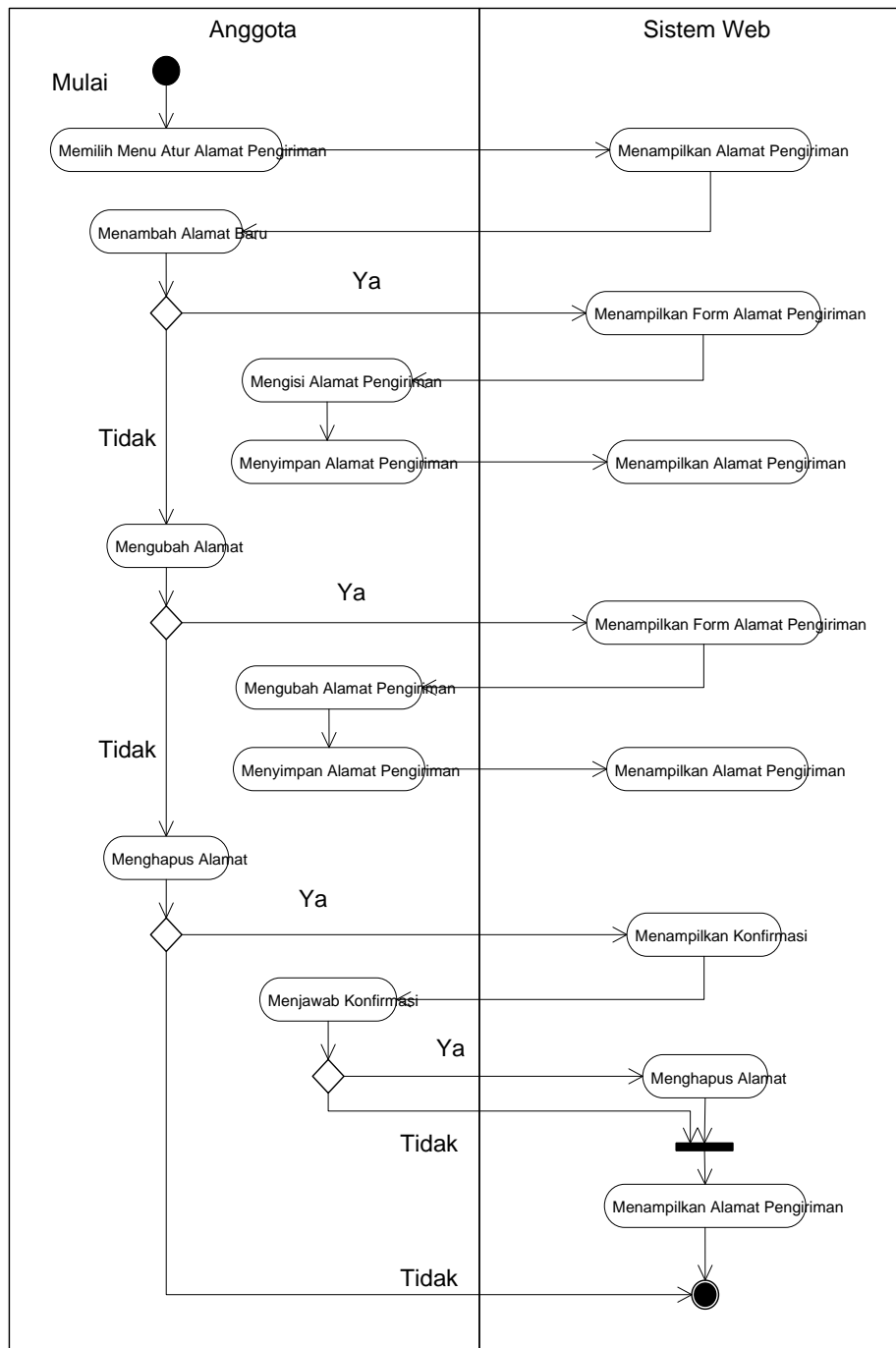
Gambar 2.7 Contoh Use Case Diagram

### 2.9.2. Activity Diagram

Activity diagram merupakan diagram untuk mengilustrasikan kegiatan utama dan hubungan di antara kegiatan dalam suatu proses. Diagram Activity memainkan peran yang mirip dengan diagram alir atau *flow chart* akan tetapi pada diagram ini memiliki prinsip dan notasi yang berbeda dimana pada diagram ini mendukung *behavior parallel*. Diagram activity dapat digunakan untuk beberapa hal, antara lain [19]:

1. Sebagai pandangan dalam yang dilakukan di operasi
2. Sebagai pandangan dalam bagaimana objek – objek dapat bekerja
3. Sebagai pandangan dalam aksi – aksi dan juga bagaimana pengaruhnya pada objek – objek.
4. Sebagai logika dari proses bisnis

Contoh dari *Activity Diagram* diilustrasikan pada **Gambar 2.8**.



Gambar 2.8 Contoh Activity Diagram

### 2.9.3. Class Diagram

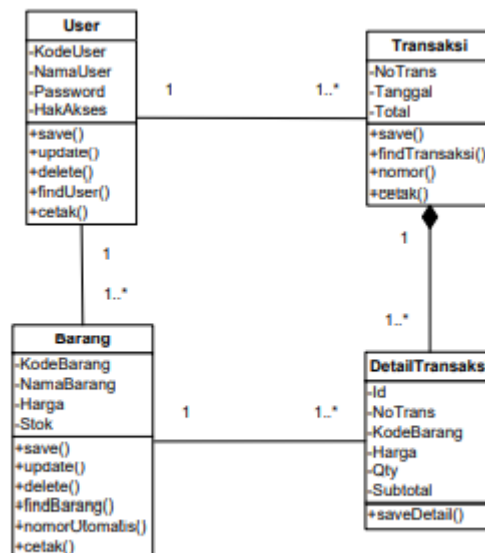
Use Case Diagram adalah salah satu diagram yang termasuk kedalam *Behavior Diagram*, dimana diagram ini dapat digunakan untuk memodelkan aspek perilaku sistem. Use Case Diagram dapat digunakan untuk memvisualkan,

menspesifikasikan dan juga mendokumentasikan kebutuhan perilaku sistem.

Tujuan utama dari pemodelan use case adalah [19] :

1. Dapat memodelkan kosa kata di sistem.
2. Dapat memodelkan distribusi tanggung jawab di sistem.
3. Dapat memodelkan tipe primitif.
4. Dapat memodelkan entitas yang bukan perangkat lunak.
5. Dapat memodelkan kolaborasi.
6. Dapat memodelkan skema logika basis data.

Contoh *Class Diagram* adalah diilustrasikan pada **Gambar 2.9**.



**Gambar 2.9** Contoh Class Diagram

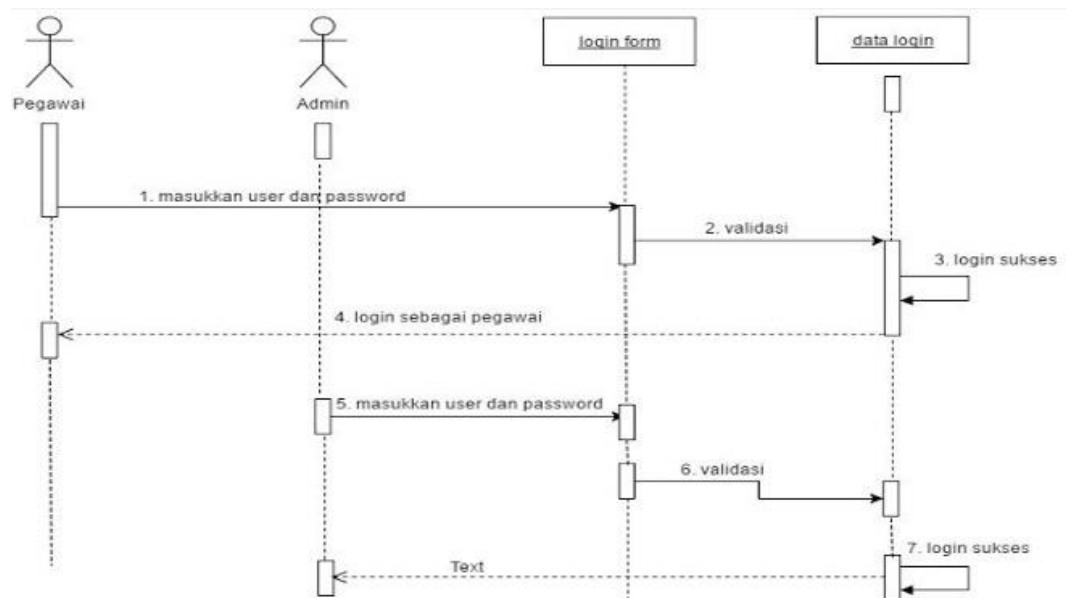
#### 2.9.4. Sequence Diagram

Sequence Diagram adalah diagram yang dapat menunjukkan interaksi yang terjadi dengan menampilkan setiap partisipan dengan menggunakan garis alir secara vertical dan juga melakukan pengurutan pesan dari atas ke bawah. Sequence diagram dapat menunjukkan pesan yang lewat di antara objek untuk *use case* tertentu dari waktu ke waktu. Sequence diagram mengilustrasikan objek – objek yang berpartisipasi di dalam suatu use case. Sequence Diagram memiliki beberapa kegunaan, antara lain[19] :

1. Untuk overview perilaku bisnis

2. Untuk menunjukkan objek – objek yang diperlakukan
3. Untuk Mendokumentasikan skenario dari suatu diagram sue case
4. Untuk memberikan jalur pengaksesan.

Contoh dari *Sequence Diagram* diilustrasikan pada **Gambar 2.10**.



**Gambar 2.10** Contoh Sequence Diagram