# BAB 2
## LANDASAN TEORI

### 2.1 Tata rias wajah korektif

Bertujuan untuk mengubah penampilan fisik yang dinilai kurang sempurna. Tata rias wajah korektif merupakan jenis tata rias wajah yang paling sering dilakukan oleh masyarakat. Maka tata rias korektif selalu berhubungan dengan panampilan natural dan sederhana. Namun lebih elegan,karena dapat menkoreksi kekurangan dan kelebihan di wajah kita agar terlihat lebih segar. [5]

#### 2.1.1 Tata rias wajah untuk mode/ seni (Styling make up)

Merupakan kegiatan mengubah penampilan murni untuk tujuan seni. Melukis tubuh (*Body painting*) merupakan salah satu contoh kegiatan styling make up. [5]

#### 2.1.2 Tata rias wajah untuk karakterisasi

Banyak digunakan untuk kepentingan dunia akting dan hiburan. Setiap warna dan bahan kosmetika yang digunakan ditujukan untuk membentuk karakter/watak tertentu, misalnya penggunaan eye shadow gelap untuk memberi karakter galak. [5]

### 2.2 Kosmetik

Kosmetik adalaah zat perawatan yang digunakan untuk meningkatkan penampilan atau aroma tubuh manusia. Kosmetik umumnya merupakan campuran dari beragam senyawa kimia, beberapa terbuat dari sumber-sumber alami dan kebanyakan dari bahan sintetis. Perihal atau tata cara menggunakan kosmetik disebut dengan tata rias atau make up. [6]

Di Amerika Serikat, Food and Drug Administration (FDA), badan yang mengatur industri kosmetik, mendefinisikan kosmetik sebagai “produk yang dimaksudkan untuk digunakan pada tubuh manusia untuk membersihkan, mempercantik, mempromosikan daya tarik, atau mengubah penampilan tanpa

mempengaruhi struktur atau fungsi tubuh". Definisi ini juga mencakup bahan apapun yang digunakan sebagai komponen produk kosmetik. FDA secara khusus mengecualikan sabun dari kategori ini, meskipun secara luas sabun juga tergolong kosmetik. [6]

### 2.2.1 Merek

Merek merupakan nama, istilah, tanda, simbol/lambang/logo, desain, warna, gerak, atau kombinasi atribut-atribut produk lainnya yang diharapkan dapat memberikan identitas yang membedakannya dengan produk pesaing. Pada dasarnya merek juga merupakan janji produsen untuk secara konsisten menyampaikan serangkaian ciri-ciri/fitur, manfaat, dan layanan tertentu kepada para konsumen.

Menurut Kotler, yang dimaksud dengan merek adalah janji penjual untuk selalu konsisten dalam memberikan feature, manfaat maupun jasa tertentu pada pembeli atau pedagang. Dari sisi pelanggan atau pembeli sendiri, jumlah penawaran yang tertinggi sudah dapat diperkirakan. Oleh karena itu, pelanggan pasti menginginkan hasil yang maksimal, karena budget (anggaran) yang dipunyai terbatas. Dengan cara berusaha untuk mencari informasi terkait produk yang ditawarkan. Merek yang dianggap bernilai tinggi, yang akan dihargai oleh konsumen dan dianggap bisa merefleksikan karakter pemakai merek tersebut.

Undang-Undang Republik Indonesia Nomor 20 Tahun 2016 Tentang Merek dan Indikasi Geografis Pasal 1 menyatakan bahwa, Merek adalah tanda yang dapat ditampilkan secara grafis berupa gambar, logo, nama, kata, huruf, angka, susunan warna, dalam bentuk 2 (dua) dimensi dan/atau 3 (tiga) dimensi, suara, hologram, atau kombinasi dari 2 (dua) atau lebih unsur tersebut untuk membedakan barang dan/atau jasa yang diproduksi oleh orang atau badan hukum dalam kegiatan perdagangan barang dan/atau jasa.

Penggunaan suatu merek oleh perusahaan mempunyai empat macam tujuan yaitu:

a. Sebagai identitas perusahaan yang membedakannya dengan produk pesaing, sehingga pelanggan mudah mengenali dan melakukan pembelian ulang.
b. Sebagai alat promosi yang menonjolkan daya tarik produk (misalnya dengan bentuk desain dengan warna-warna yang menarik).
c. Untuk membina citra, yaitu dengan memberikan keyakinan, jaminan kualitas, serta citra prestise tertentu kepada konsumen.
d. Untuk mengendalikan dan mendominasi pasar. Artinya, dengan membangun merek yang terkenal, dengan bercitra baik, dan telah dilindungi hak eksklusif berdasarkan hak cipta/paten, maka perusahaan dapat meraih dan mempertahankan loyalitas konsumen.

Dalam suatu merek juga terkandung enam macam makna yaitu:

a. Atribut. Merek menyampaikan atribut-atribut tertentu, misalnya produk tahan lama (awet), mahal, desain berkualitas, nilai jual kembali yang tinggi, dan sebagainya.
b. Manfaat. Merek bukanlah sekadar sekumpulan atribut, sebab yang dibeli konsumen adalah manfaat, bukannya atribut. Atribut harus diterjemahkan ke dalam manfaat fungsional dan emosional/psikologis. Misalnya atribut mahal dapat diungkapkan dalam manfaat emosional seperti “Mobil ini dapat menaikkan gengsiku”. sedangkan atribut tahan lama dapat dicerminkan dalam manfaat fungsional seperti “Saya tidak perlu membeli mobil baru lagi selama beberapa tahun mendatang”.
c. Nilai-Nilai. Merek juga menyatakan nilai-nilai yang dianut produsennya. Contohnya Mercedes (sebagaimana umumnya produk-produk buatan Jerman) mencerminkan kinerja tinggi, keamanan dan prestise.
d. Budaya. Dalam merek terkandung pula budaya tertentu. Mercedes mencerminkan budaya Jerman, seperti terorganisasi rapi, efisien, dan berkualitas tinggi.
e. Kepribadian. Merek bisa pula memproyeksikan kepribadian tertentu. Apabila suatu merek divisualisasikan dengan orang, binatang, atau suatu obyek, apa yang akan terbayangkan? Mercedes memberi kesan pimpinan

yang berwibawa (orang), singa yang berkuasa (binatang), atau istana yang megah (obyek).

f. Pemakai. Merek juga menyiratkan tipe konsumen yang membeli atau menggunakan produknya. Contohnya, mobil Mercedes kerap kali dipersepsikan sebagai mobil untuk eksekutif puncak paruh baya.

Agar suatu merek dapat mencerminkan makna-makna yang ingin disampaikan, maka ada beberapa persyaratan yang harus diperhatikan yaitu:

a. Merek harus khas atau unik
b. Merek harus menggambarkan sesuatu mengenai produk dan pemakaiannya
c. Merek harus menggambarkan kualitas produk
d. Merek harus mudah diucapkan, dikenali, dan diingat
e. Merek tidak boleh mengandung arti yang bumk di negara dan dalam bahasa lain.
f. Merek harus dapat menyesuaikan diri (*adaptable*) dengan produk-produk baru yang mungkin ditambahkan ke dalam lini produk.

Kemampuan merek sebagai alat untuk menarik konsumen untuk membeli barang yang terkenal mereknya, akan menghambat perusahaan-perusahaan baru untuk mengembangkan usaha yang sama. Sebagai alternatif perusahaan lain akan mengembangkan barang generik, yaitu barang yang sama jenisnya yang adakalanya tidak bermerek, tetapi dijual dengan harga yang jauh lebih murah dari barang yang terkenal mereknya.

## 2.3 YouTube

YouTube adalah sebuah situs web berbagi videoyang dibuat oleh tiga mantan karyawan PayPalpada Februari 2005. Situs ini memungkinkan pengguna mengunggah, menonton, dan berbagi video. Perusahaan ini berkantor pusat di San Bruno, California, dan memakai teknologi Adobe Flash Video dan HTML5untuk menampilkan berbagai macam konten video buatan pengguna, termasuk klip film, klip TV, dan video musik. Selain itu ada pula konten amatir seperti blog video, video orisinal pendek, dan video pendidikan. [7]

Kebanyakan konten di YouTube diunggah oleh individu, meskipun perusahaan-perusahaan media seperti CBS, BBC, Vevo, Hulu, dan organisasi lain sudah mengunggah material mereka ke situs ini sebagai bagian dari program kemitraan YouTube. Pengguna tak terdaftar dapat menonton video, sementara pengguna terdaftar dapat mengunggah video dalam jumlah tak terbatas. Video-video yang dianggap berisi konten ofensif hanya bisa ditonton oleh pengguna terdaftar berusia 18 tahun atau lebih. Pada November 2006, YouTube, LLC dibeli oleh Google dengan nilai US$1,65 miliar dan resmi beroperasi sebagai anak perusahaan Google. [7]

### 2.3.1 Youtube API

Application Programming Interface, atau YouTube API, memungkinkan pengembang mengakses statistik video dan data saluran YouTube melalui dua jenis panggilan, REST dan XML-RPC.

Untuk memakai YouTube API, seorang pengembang harus memiliki Developer ID. Ini adalah properti tambahan yang terpasang di akun YouTube si pengembang. Informasi yang tersedia untuk para pengembang mirip dengan informasi yang dapat diperoleh dengan mengakses umpan RSS YouTube.

Per Maret 2006, panggilan API dari Flash dinonaktifkan karena masalah keamanan.

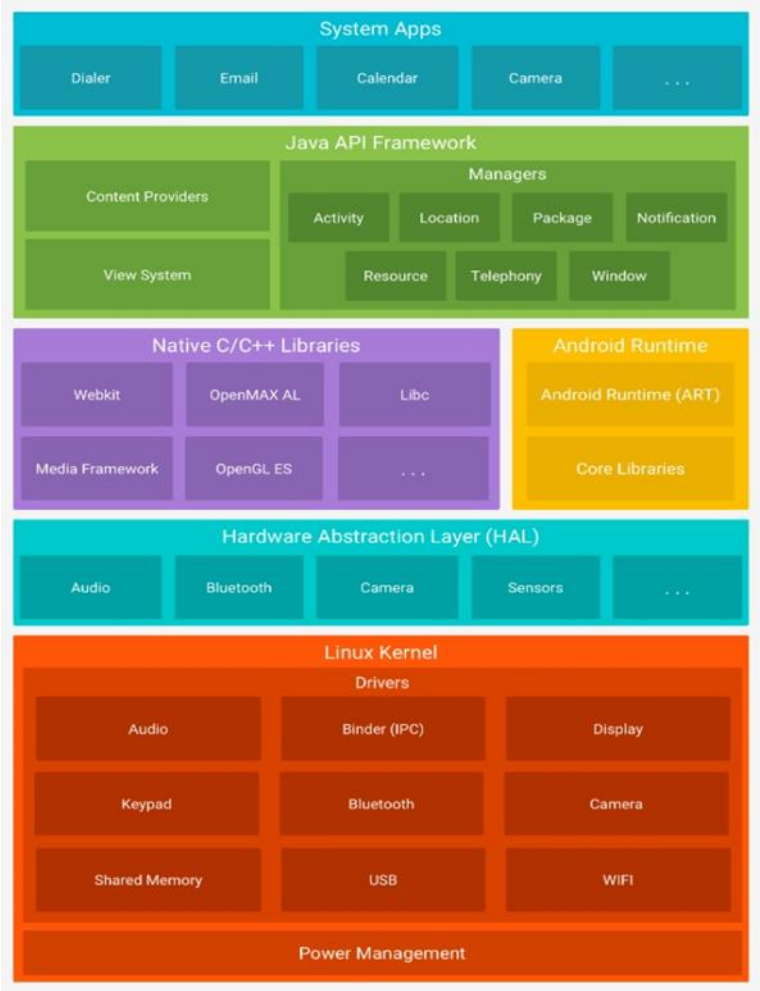
## 2.4 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finasial dari Google yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didrikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak dan telekomunikasi yang bertujuan untuk memajukan standard terbuka perangkat seluler. Ponsel pertama android mulai dijual pada bulan oktober 2008.

Android kini menguasai ratusan juta perangkat mobile di lebih dari 190 negara di seluruh dunia. Ini adalah Installed base terbesar dari setiap platform seluler dan berkembang dengan cepat, setiap hari satu juta pengguna lainnya menyalakan perangkat android mereka untuk pertama kalinya dan memulai mencari aplikasi, game dan konten digital lainnya. [8]

### 2.4.1 Arsitektur Android

Android adalah tumpukan perangkat lunak berbasis Linux sumber terbuka yang dibuat untuk berbagai perangkat dan faktor bentuk. Diagram berikut menunjukkan komponen besar dari platform Android. [8]




*Sumber gambar: https://developer.android.com/guide/platform/?hl=id*

**Gambar 2.1 Arsitektur Android**

1. Linux Kernel

Fondasi platform Android adalah kernel Linux. Sebagai contoh, Android Runtime (ART) bergantung pada kernel Linux untuk fungsionalitas dasar seperti threading dan manajemen memori tingkat rendah.

Menggunakan kernel Linux memungkinkan Android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen perangkat untuk mengembangkan driver perangkat keras untuk kernel yang cukup dikenal.

2. Hardware Abstraction Layer (HAL)

Hardware Abstraction Layer (HAL) menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja Java API yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Bila API kerangka kerja melakukan panggilan untuk mengakses perangkat keras, sistem Android memuat modul pustaka untuk komponen perangkat keras tersebut.

3. Android Runtime

Untuk perangkat yang menjalankan Android versi 5.0 (API level 21) atau yang lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan tahap Android Runtime (ART). ART ditulis guna menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi file DEX, format bytecode yang didesain khusus untuk Android yang dioptimalkan untuk footprint memori minimal. Buat rantai aplikasi, misalnya Jack, mengumpulkan sumber Java ke bytecode DEX, yang dapat berjalan pada platform Android. Beberapa fitur utama ART mencakup:

1. Kompilasi mendahului waktu (AOT) dan tepat waktu (JIT)

2. Pengumpulan sampah (GC) yang dioptimalkan

3. Dukungan debug yang lebih baik, mencakup profiler sampling terpisah, pengecualian diagnostik mendetail dan laporan kerusakn dan kemampuan untuk mengatur titik pantau guna memantau bidang tertentu.

Sebelum ke Android versi 5.0 (API level 21), Dalvik adalah waktu proses Android. Jika aplikasi Anda berjalan baik pada ART, semestinya berfungsi baik juga pada Dalvik, tetapi mungkin tidak sebaliknya.

Android juga menyertakan serangkaian pustaka waktu proses inti yang menyediakan sebagian besar fungsionalitas bahasa pemrograman Java, termasuk beberapa fitur bahasa Java 8, yang digunakan kerangka kerja Java API.

4. Pustaka C/C++ Asli

Banyak komponen dan layanan sistem Android inti seperti ART dan HAL dibuat dari kode asli yang memerlukan pustaka asli yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja Java API mengekspos fungsionalitas beberapa pustaka asli pada aplikasi. Misalnya, Anda bisa mengakses OpenGL ES melalui kerangka kerja Java OpenGL API Android guna menambahkan dukungan untuk menggambar dan memanipulasi grafik 2D dan 3D pada aplikasi Anda. Jika Anda mengembangkan aplikasi yang memerlukan kode C atau C++, Anda bisa menggunakan Android NDK untuk mengakses beberapa pustaka platform asli langsung dari kode asli.

5. Kerangka Kerja Java API

Keseluruhan rangkaian fitur pada Android OS tersedia untuk Anda melalui API yang ditulis dalam bahasa Java. API ini membentuk elemen dasar yang Anda perlukan untuk membuat aplikasi Android dengan menyederhanakan penggunaan kembali inti, komponen dan layanan sistem modular, yang menyertakan berikut ini:

a. Tampilan Sistem yang kaya dan luas bisa Anda gunakan untuk membuat UI aplikasi, termasuk daftar, kisi, kotak teks, tombol, dan bahkan browser web yang dapat disematkan
b. Pengelola Sumber Daya, memberikan akses ke sumber daya bukan kode seperti string yang dilokalkan, grafik, dan file layout
c. Pengelola Notifikasi yang mengaktifkan semua aplikasi guna menampilkan lansiran khusus pada bilah status

d. Pengelola Aktivitas yang mengelola daur hidup aplikasi dan memberikan back-stack navigasi yang umum. fitur bahasa Java 8, yang digunakan kerangka kerja Java API.

6. Aplikasi Sistem

Android dilengkapi dengan serangkaian aplikasi inti untuk email, perpesanan SMS, kalender, menjelajahi internet, kontak, dll. Aplikasi yang disertakan bersama platform tidak memiliki status khusus pada aplikasi yang ingin dipasang pengguna. Jadi, aplikasi pihak ketiga dapat menjadi browser web utama, pengolah pesan SMS atau bahkan keyboard utama (beberapa pengecualian berlaku, seperti aplikasi Settings sistem).

Aplikasi sistem berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh developer dari aplikasi mereka sendiri. Misalnya, jika aplikasi Anda ingin mengirimkan pesan SMS, Anda tidak perlu membangun fungsionalitas tersebut sendiri—sebagai gantinya Anda bisa menjalankan aplikasi SMS mana saja yang telah dipasang guna mengirimkan pesan kepada penerima yang Anda tetapkan. [8]

### 2.4.2 Versi Android

Versi Android diawali dengan dirilisnya Android beta pada bulan November 2007. Versi komersial pertama, Android 1.0, dirilis pada September 2008. Android dikembangkan secara berkelanjutan oleh Google dan Open Handset Alliance (OHA), yang telah merilis sejumlah pembaruan sistem operasi ini sejak dirilisnya versi awal.

Sejak April 2009, versi Android dikembangkan dengan nama kode yang dinamai berdasarkan makanan pencuci mulut dan penganan manis. Masing-masing versi dirilis sesuai urutan alfabet, yakni Cupcake (1.5), Donut (1.6), Eclair (2.0–2.1), Froyo (2.2–2.2.3), Gingerbread (2.3–2.3.7), Honeycomb (3.0–3.2.6), Ice Cream Sandwich (4.0–4.0.4), Jelly Bean (4.1–4.3), KitKat (4.4+), Lollipop (5.0+), Marshmallow (6.0+), hingga yang terbaru adalah Nougat (7.0+) dan selanjutnya versi android terbaru yang ditunggu-tunggu adalah Android O (8.0+).

Pada tanggal 3 September 2013, Google mengumumkan bahwa sekitar 1 miliar perangkat seluler aktif di seluruh dunia menggunakan OS Android.

Berikut adalah versi android berdasarkan tanggal rilis:

1. Android 6.0 Marshmallow (API level 23)

Android Marshmallow memperkenalkan model izin yang didesain ulang: sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi. Sebuah sistem opt-in sekarang digunakan, di mana pengguna akan diminta untuk memberikan atau menolak izin individu (seperti kemampuan untuk mengakses kamera atau mikrofon) untuk aplikasi ketika mereka dibutuhkan. Aplikasi mengingat hibah izin mereka, dan mereka dapat disesuaikan oleh pengguna setiap saat. Model izin baru akan digunakan hanya oleh aplikasi yang dikompilasi untuk Marshmallow menggunakan kit pengembangan perangkat lunak (SDK) tersebut, sementara semua aplikasi lainnya akan terus menggunakan model izin sebelumnya.

Marshmallow juga memiliki skema manajemen daya baru bernama Doze yang mengurangi tingkat aktivitas aplikasi latar belakang saat perangkat menentukan bahwa itu tidak sedang aktif ditangani oleh pengguna, yang, menurut Google, menggandakan pemakaian baterai perangkat. Hal ini juga memperkenalkan pilihan untuk mengatur ulang semua pengaturan jaringan, tersedia untuk pertama kalinya pada Android, yang membersihkan pengaturan terkait jaringan untuk WI-FI, Bluetooth dan koneksi seluler.

Android Marshmallow memberikan dukungan asli untuk pengenalan sidik jari, memungkinkan penggunaan sidik jari untuk membuka perangkat dan otentikasi Play Store dan pembelian Android Pay; API standar juga tersedia untuk melaksanakan otentikasi berbasis sidik jari dalam aplikasi lain. Android Marshmallow mendukung USB Type-C, termasuk kemampuan untuk menginstruksikan perangkat untuk mengisi daya perangkat lain melalui USB. Marshmallow juga memperkenalkan “pranala yang diverifikasi” yang dapat dikonfigurasi untuk membuka langsung dalam aplikasi tertentu mereka tanpa petunjuk pengguna lanjut.

Versi API Android yang disediakan oleh Marshmallow adalah 23. Alat pengembang Android Marshmallow tersedia di Pengelola SDK di bawah tingkat API “MNC”.

2. Android 7.0-7.1 Nougat (API level 24-25)

Android “Nougat” (kode nama N dalam pengembangan) adalah rilis 7.0 besar dari sistem operasi Android. Ini pertama kali dirilis sebagai pratinjau pengembang pada tanggal 9 Maret 2016, dengan gambar pabrik untuk perangkat Nexus saat ini, serta dengan “Program Beta Beta” baru yang memungkinkan perangkat yang didukung ditingkatkan versinya ke versi Android Nougat melalui over-the –air update. Rilis terakhir adalah pada tanggal 22 Agustus 2016. Pratinjau akhir pembuatannya dirilis pada tanggal 18 Juli 2016, dengan nomor bangunan NPD90G.

Pada tanggal 19 Oktober 2016, Google merilis Android 7.1.1 sebagai pratinjau pengembang untuk Nexus 5X, Nexus 6P dan Pixel C. Pratinjau kedua mulai tersedia pada 22 November 2016, sebelum versi final diluncurkan ke publik pada bulan Desember. 5, 2016.

3. Android 8.0 Oreo (API level 26)

Android Oreo adalah rilis utama ke 8 dari sistem operasi Android. Ini pertama kali dirilis sebagai preview pengembang pada tanggal 21 Maret 2017, dengan gambar pabrik untuk perangkat Nexus dan Pixel saat ini. Pratinjau pengembang terakhir dirilis pada tanggal 24 Juli 2017, dengan rilis stabil yang diharapkan pada bulan Agustus atau September 2017. [8]

## 2.5 Google Cloud Speech To Text

Google Cloud Speech-to-Text memungkinkan pengembang mengubah audio menjadi teks dengan menerapkan model jaringan saraf yang kuat dalam API yang mudah digunakan. API mengakui 120 bahasa dan varian untuk mendukung basis pengguna global Anda. Anda dapat mengaktifkan perintah-dan-kontrol suara, mentranskripsikan audio dari pusat panggilan, dan banyak lagi. Ini dapat

memproses streaming real-time atau audio yang direkam sebelumnya, menggunakan teknologi pembelajaran mesin Google.

Didukung oleh pembelajaran mesin. Menerapkan algoritma jaringan saraf dalam pembelajaran yang paling canggih ke audio untuk pengenalan suara dengan akurasi yang tak tertandingi. Akurasi Cloud Speech-to-Text meningkat seiring waktu ketika Google meningkatkan teknologi pengenalan ucapan internal yang digunakan oleh produk Google.

Mengenali 120 bahasa dan varian. Cloud Speech-to-Text dapat mendukung basis pengguna global Anda, mengenali 120 bahasa dan varian. Anda juga dapat memfilter konten yang tidak pantas dalam hasil teks untuk semua bahasa.

Secara otomatis mengidentifikasi bahasa lisan. Menggunakan Cloud Speech-to-Text Anda dapat mengidentifikasi bahasa apa yang diucapkan dalam ujaran (terbatas pada empat bahasa). Ini dapat digunakan untuk pencarian suara (seperti, “Berapa suhu di Paris?”) Dan kasus penggunaan perintah (seperti, “Balikkan volume.”)

Mengembalikan transkripsi teks secara real-time untuk audio bentuk-pendek atau format panjang. Cloud Speech-to-Text dapat melakukan streaming hasil teks, segera mengembalikan teks seperti yang dikenali dari streaming audio atau saat pengguna berbicara. Sebagai alternatif, Cloud Speech-to-Text dapat mengembalikan teks yang dikenali dari audio yang disimpan dalam file. Alat ini mampu menganalisis audio bentuk-pendek dan format panjang.

Secara otomatis mentranskripsi kata benda yang tepat dan pemformatan spesifik konteks. Cloud Speech-to-Text dirancang untuk bekerja dengan baik dengan ucapan kehidupan nyata dan dapat secara akurat mentranskripsikan kata benda yang tepat (seperti, Sundar Pichai) dan bahasa format yang tepat (seperti, tanggal, nomor telepon). Google mendukung lebih dari 10x kata benda yang tepat dibandingkan dengan jumlah kata di seluruh Oxford English Dictionary.

Menawarkan pemilihan model yang dibuat sebelumnya, yang dirancang untuk kasus penggunaan Anda. Cloud Speech-to-Text hadir dengan beberapa model pengenalan suara yang dibuat sebelumnya sehingga Anda dapat

mengoptimalkan kasus penggunaan Anda (seperti, perintah suara). Contoh: Model transkripsi video pra-bangun kami ideal untuk pengindeksan atau subtitling video dan / atau konten multispeaker dan menggunakan teknologi pembelajaran mesin yang mirip dengan teks YouTube. [9]

## 2.6 Database

Databse dapat dijelaskan sebagai kumpulan koleksi data yang terintegrasi. Data adalah sebuah representasi dari beberapa objek fisik. Database berisikan penjelasan tentang struktur yang memilikinya database saling terintegrasi dan memiliki hubungan antar data didalamnya.

### 2.6.1 NoSQL Database Model

NoSQL (Not Only SQL) merupakan salah satu jenis dari database management system yang menggunakan kelompok data yang tidak bersifat non-relational dimana database tidak dibangun kedalam tabel-tabel dan tidak menggunakan SQL dalam melakukan manipulasi data. NOSQL database management system sangat berguna ketika bekerja dengan jumlah data yang sangat besar dan data-data didalamnya tidak memerlukan model.

### 2.6.2 Klasifikasi NoSQL Database

Database NoSQL model dibagi menjadi 4 jenis yaitu :

1. Key-Value Databse

Penyimpanan data menggunakan key value dilakukan dalam sebuah form yang berisi key dan value

2. Column Oriented Database

Penyimpanan data berbasis kolom dilakukan dengan meyimpan data dalam sekumpulan kolom, kolom tersebut disimpan dalam sebuah file yang sama yang disebut dengan column families. Database jenis ini biasa dipakai untuk aplikasi yang secara intensif dipakai oleh user.

3. Document Database

Penyimpanan data dalam dokumen ini hampir mirip dengan key-value databse, namun data yang disimpan dalam dokumen database ini disimpan dalam format JSON atau XML. Database jenis ini digunakan untuk aplikasi yang data-data didalamnya sering mengalami perubahan seperti aplikasi Customer Relationship Management.

4. Graph Database

Database grafik menggunakan struktur grafik untuk menyimpan data. Databsae jenis ini cocok digunakan oleh aplikasi yang memiliki banyak interkoneksi antar data seperti media sosial.

### 2.6.3 NoSQL Database Query

Penggunaan query adalah salah satu bagian yang dijabarkan tidak secara rinci pada database NoSQL. Salah satu kemungkinan dari proses query pada database NoSQL adalah dengan menggunakan perintah SQL yang sangat terbatas. Database NoSQL menggunakan operasi semacam get (key) untuk mendapatkan data dari key yang dipanggil, put (key,value) untuk melakukan proses insert atau update serta perintah delete (key) untuk menghapus key yang ada di database. Beberapa database NoSQL memerlukan perintah tambahan execute (key) untuk mejalankan perintahperintah diatas.

Distribusi data secara horizontal pada database NoSQL menyebabkan tidak didukungnya perintah join atau order by seperti yang ada pada database relational.

Apabila diperlukan, maka operasi join maupun order by dapat dilakukan dari sisi client. Operasi seleksi pada database NoSQL sering dijelaskan pada level API.

## 2.7 Object Oriented Analysis Design

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memeriksa requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-

kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

## 2.8 Unified Modelling Language

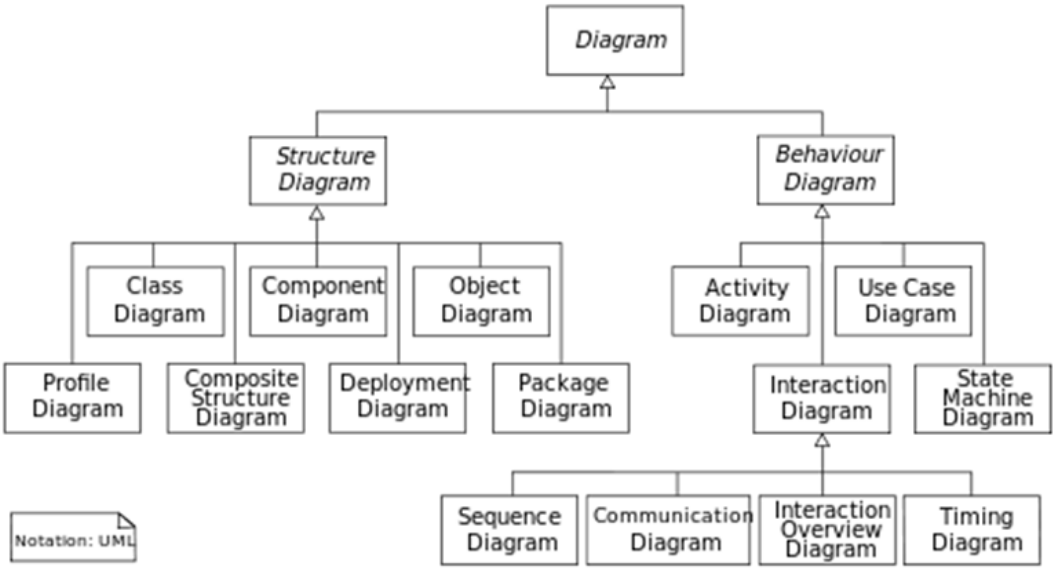
Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock , dsb. Masa itu terkenal dengan masa perang metodologi (method war) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita

bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.




**Gambar 2.2 Diagram UML**

Berikut penjelasan singkat dari pembagian kategori tersebut adalah :

1. Structure Diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari suatu sistem yang dimodelkan
2. Behavior Diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

### 2.8.1 Class Diagram

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat

kelaskelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas sebagai berikut:

3. Kelas main yaitu kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
4. Kelas yang menangani tampilan sistem (view) yaitu kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
5. Controller yaitu kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case, kelas ini biasa disebut kelas proses.
6. Model yaitu kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

### 2.8.2 Object Diagram

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan.

### 2.8.3 Use Case Diagram

Diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fingsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.[20]

Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang.
2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

### 2.8.4 Activity Diagram

Diagram aktivitas asatu activity diagram menggambarkan workflow (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktifitas menggambarkan aktifitas bukan apa yang dilakukan aktor.

Diagram aktifitas juga banyak digunakan untuk mendefinisikan hal-hak berikut:

1. Rancangan proses bisnis dimana setiap urutan aktifitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / user interface dimana setiap aktifitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktifitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang ditampilkan pada diagram aktifitas

### 2.8.5 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirim dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objekobjek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

## 2.9 Algoritma Naïve Bayes

Algoritma Naive Bayes merupakan sebuah metoda klasifikasi menggunakan metode probabilitas dan statistik yg dikemukakan oleh ilmuwan Inggris Thomas Bayes. Algoritma Naive Bayes memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dr Naïve Bayes Classifier ini adalah asumsi yg sangat kuat (naïf) akan independensi dari masing-masing kondisi / kejadian.

Naive Bayes Classifier bekerja sangat baik dibanding dengan model classifier lainnya. Hal ini dibuktikan pada jurnal Xhemali, Daniela, Chris J. Hinde, and Roger G. Stone. “Naive Bayes vs. Decision trees vs. Neural networks in the classification of training web pages.” (2009), mengatakan bahwa “Naïve Bayes Classifier memiliki tingkat akurasi yg lebih baik dibanding model classifier lainnya”.

Keuntungan penggunan adalah bahwa metoda ini hanya membutuhkan jumlah data pelatihan (training data) yang kecil untuk menentukan estimasi parameter yg diperlukan dalam proses pengklasifikasian. Karena yg diasumsikan sebagai variabel independent, maka hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians.

Tahapan dari proses algoritma Naive Bayes adalah:

1. Menghitung jumlah kelas / label.
2. Menghitung Jumlah Kasus Per Kelas
3. Kalikan Semua Variable Kelas
4. Bandingkan Hasil Per Kelas

Kelebihan naïve bayes

1. Mudah untuk dibuat
2. Hasil bagus

Kekurangan naïve bayes

1. Asumsi independence antar atribut membuat akurasi berkurang (karena biasanya ada keterkaitan) [10]