BAB 2
TINJAUAN PUSTAKA

2. 1 Mata

Mata adalah organ vital yang berfungsi sebagai sistem penglihatan pada manusia dan banyak hewan. Struktur mata yang kompleks memungkinkan kita untuk melihat dan memahami dunia di sekitar kita. Cahaya yang masuk melalui pupil mata difokuskan oleh lensa ke retina. Retina mengubah cahaya menjadi sinyal listrik yang dikirim ke otak melalui saraf optik, memungkinkan kita untuk melihat gambar. Mata manusia dapat membedakan jutaan warna dan merespons perubahan cahaya dengan cepat, berkat iris yang mengatur ukuran pupil. Selain itu, mata memiliki beberapa bagian penting lainnya seperti kornea yang melindungi mata dan membantu memfokuskan cahaya, dan badan vitreus, substansi gelatin yang mengisi ruang mata dan membantu mempertahankan bentuk bola mata [13].

2. 2 Tunanetra

Tunanetra adalah istilah yang digunakan untuk menggambarkan kondisi seseorang yang memiliki keterbatasan dalam indra penglihatannya. Secara etimologis, kata "tuna" berarti rusak atau rugi dan "netra" berarti mata, sehingga tunanetra merujuk pada individu yang mengalami kerusakan atau hambatan pada organ mata. Kondisi ini bisa terjadi baik secara anatomis, yaitu struktur dan keterhubungan tubuh, maupun fisiologis, yaitu fungsi tubuh. Tunanetra dapat meliputi mereka yang memiliki ketajaman visual sangat rendah hingga mereka yang tidak mampu melihat sama sekali [14].

Ada dua jenis utama tunanetra yaitu Low Vision (Kurang Awas) dan Blind (Buta). Low Vision adalah kondisi di mana seseorang masih dapat membedakan gelap dan terang atau memiliki sedikit kemampuan melihat. Sementara itu, Blind atau kebutaan total adalah ketika seseorang tidak memiliki penglihatan sama sekali dan tidak dapat membedakan antara gelap dan terang. Penyandang tunanetra sering kali mengembangkan kemampuan indra lainnya, seperti pendengaran atau perabaan, untuk membantu dalam navigasi dan aktivitas sehari-hari [14].

Tunanetra sering kali mengandalkan indra-indra lainnya, seperti pendengaran, perabaan, penciuman, dan pengecap, untuk menggantikan keterbatasan penglihatannya. Penyandang tunanetra juga dapat menggunakan alat bantu teknologi, dan pelatihan khusus untuk membantu dalam aktivitas sehari-hari.

2. 3 Skala Likert

Skala Likert, atau Likert Scale, merupakan alat penelitian yang digunakan untuk mengukur sikap dan pendapat. Dalam skala ini, responden diminta untuk mengisi kuesioner yang menunjukkan tingkat persetujuan terhadap serangkaian pernyataan atau pertanyaan. Pernyataan atau pertanyaan tersebut biasanya disebut sebagai variabel penelitian dan ditetapkan secara spesifik oleh peneliti. Nama skala ini berasal dari penciptanya, Rensis Likert, seorang ahli psikologi sosial dari Amerika Serikat [15].

Cara untuk menginterpretasikan hasil skala Likert adalah melalui analisis interval. Untuk menganalisis secara kuantitatif, jawaban responden diberi bobot atau skor. Sebagai contoh, terdapat pernyataan "Apakah Anda setuju bahwa aplikasi ini dapat membantu dalam mendeteksi objek yang ada disekitar?" Bobot atau skor yang diberikan untuk jawaban ini misalnya Sangat Setuju (SS) = 5, Setuju (S) = 4, Kurang Setuju (KS) = 3, Tidak Setuju (TS) = 2, dan Sangat Tidak Setuju (STS) = 1. Dengan jumlah responden sebanyak 10 orang, rincian dan perhitungannya adalah sebagai berikut: Jawaban Sangat Setuju (SS) = 3 responden x 5 = 15, Setuju (S) = 3 responden x 4 = 12, Kurang Setuju (KS) = 1 responden x 3 = 3, Tidak Setuju (TS) = 2 responden x 2 = 4, Sangat Tidak Setuju (STS) = 1 responden x 1 = 1, sehingga total skor adalah 25 [15].

Skor maksimum dapat dihitung sebagai 10 x 5 = 50 (jumlah responden dikalikan dengan skor tertinggi pada skala Likert), dan skor minimum adalah 10 x 1 = 10 (jumlah responden dikalikan dengan skor terendah). Indeks (%) kemudian dihitung sebagai (25 / 50) x 100 = 50% (Total Skor dibagi Skor Maksimum dikalikan 100).

Penilaian berdasarkan interval indeks adalah sebagai berikut:

Indeks 0% – 19,99%: Sangat Tidak Setuju

Indeks 20% – 39,99%: Tidak Setuju

Indeks 40% – 59,99%: Kurang Setuju

Indeks 60% – 79,99%: Setuju

Indeks 80% – 100%: Sangat Setuju

Dengan nilai indeks 50% dari perhitungan tersebut, dapat disimpulkan bahwa responden "Kurang Setuju" bahwa aplikasi kurang dapat membantu dalam mendeteksi objek yang ada di sekitar.

2. 4 Smartphone

Smartphone adalah perangkat elektronik genggam yang menggabungkan fungsi telepon seluler dengan fitur-fitur canggih yang sering ditemukan pada komputer pribadi. Perangkat ini dilengkapi dengan layar sentuh, koneksi internet, dan kemampuan untuk menjalankan berbagai aplikasi. Dengan smartphone, pengguna dapat melakukan panggilan telepon, mengirim pesan teks, mengakses email, menjelajahi web, dan menggunakan media sosial. Selain itu, smartphone juga memungkinkan pengguna untuk mengambil foto, merekam video, dan menggunakan GPS untuk navigasi [16].

Sejarah smartphone dimulai pada awal tahun 1990-an, dengan IBM yang dikreditkan sebagai pembuat smartphone pertama yang dikenal dengan nama Simon. Sejak itu, smartphone telah mengalami perkembangan pesat, baik dari segi desain maupun fungsionalitas. Perkembangan teknologi telah membuat smartphone menjadi lebih ramping, lebih cepat, dan lebih kuat, dengan kamera yang lebih baik dan penyimpanan yang lebih besar. Smartphone modern seperti iPhone dan perangkat Android menawarkan berbagai aplikasi yang dapat diunduh untuk memenuhi kebutuhan spesifik pengguna [17].

Manfaat smartphone sangat luas, mempengaruhi hampir semua aspek kehidupan sehari-hari. Smartphone membantu orang tetap terhubung, meningkatkan produktivitas, dan memberikan hiburan. Aplikasi pendidikan memungkinkan belajar di mana saja, sementara aplikasi kesehatan membantu

pengguna memantau kebugaran dan kesehatan mereka. Dengan kemampuan kamera yang semakin canggih, smartphone juga telah mengubah cara orang-orang mengabadikan dan berbagi momen penting dalam hidupnya.

2. 5 Android

Android adalah sistem operasi berbasis Linux yang dikembangkan khusus untuk perangkat mobile seperti smartphone dan tablet. Diciptakan oleh perusahaan bernama Android Inc., yang kemudian diakuisisi oleh Google pada tahun 2005, Android telah menjadi salah satu platform mobile paling populer di dunia. Android pertama kali diperkenalkan oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White pada Oktober 2003. Awalnya, proyek ini dimaksudkan sebagai sistem operasi khusus kamera digital, tetapi visinya berkembang menjadi sesuatu yang jauh lebih besar - sebuah platform seluler yang terbuka dan dapat diakses oleh berbagai pengembang [18].

Seiring berjalannya waktu, Android telah mengalami beberapa pembaruan sejak pertama kali diluncurkan. Pada saat penulisan ini, Android telah mencapai generasi ke-14, yang dikenal dengan nama Upside Down Cake. Perkembangan versi Android dari awal perilisannya hingga saat ini dapat terlihat dalam Tabel 2.1 yang disajikan di bawah ini.

Tabel 2. 1 Versi Android

No.	Versi Android	Nama	Tanggal Rilis
1	1.0	Astro/Alpha	23 September 2008
2	1.1	Bender/Beta	9 Februari 2009
3	1.5	Cupcake	30 April 2019
4	1.6	Donut	15 September 2009
5	2.0 – 2.1	Eclair	26 Oktober 2009
6	2.2 – 2.23	Froyo	20 Mei 2010
7	2.3 – 2.37	Gingerbread	6 Desember 2010
8	3.0 – 3.2.6	Honeycomb	22 Februari 2011
9	4.0 – 4.0.4	Ice Cream Sandwich	19 Oktober 2011
10	4.1 – 4.3.1	Jelly Bean	27 Juni 2012
11	4.4 – 4.4.4	Kitkat	31 Oktober 2013
12	5.1 – 5.1.1	Lollipop	12 November 2014
13	6.0 – 6.0.1	Marshmallow	5 Oktober 2015
14	7.1 – 7.1.2	Nougat	22 Agustus 2016
15	8.0 – 8.1	Oreo	21 Agustus 2017

16	9	Pie	6 Agustus 2018
17	10	Quince Tart/ Queen Cake	3 September 2019
18	11	Red Velvet Cake	8 September 2020
19	12	Snow Cone	4 Oktober 2021
20	13	Tiramisu	10 Februari 2022
21	14	Upside Down Cake	5 Oktober 2023

2. 6 Aplikasi

Aplikasi merupakan perangkat lunak yang dirancang untuk memberikan fungsi atau layanan tertentu kepada pengguna. Aplikasi dapat diakses dan dijalankan pada berbagai perangkat, seperti smartphone, tablet, komputer, dan perangkat lainnya. Dengan perkembangan teknologi, aplikasi telah menjadi bagian integral dari kehidupan sehari-hari, memainkan peran penting dalam berbagai bidang [18].

Salah satu jenis aplikasi yang paling umum adalah aplikasi mobile. Aplikasi mobile dirancang khusus untuk digunakan pada perangkat seluler, seperti Android atau iOS. Contoh aplikasi mobile termasuk media sosial seperti Facebook dan Instagram, dan aplikasi kesehatan seperti MyFitnessPal. Aplikasi mobile memberikan kemudahan akses informasi dan layanan, serta memungkinkan pengguna untuk berinteraksi dengan teknologi secara lebih mudah dan cepat. Aplikasi dapat dikategorikan berdasarkan jenisnya, seperti [18].

1. Aplikasi perangkat bergerak (mobile apps)

   Aplikasi perangkat bergerak adalah aplikasi yang dirancang untuk perangkat bergerak, seperti smartphone dan tablet. Aplikasi perangkat bergerak biasanya didistribusikan melalui toko aplikasi, seperti Google Play Store dan App Store.

2. Aplikasi Desktop

   Aplikasi desktop adalah perangkat lunak yang dirancang untuk dijalankan di lingkungan desktop atau komputer pribadi. Berbeda dengan aplikasi web yang diakses melalui browser, aplikasi desktop memerlukan instalasi langsung pada perangkat pengguna. Aplikasi desktop biasanya memiliki antarmuka pengguna yang dioptimalkan untuk penggunaan pada monitor dan perangkat

input seperti keyboard dan mouse.

3. Aplikasi Web

Aplikasi web adalah perangkat lunak yang dirancang untuk diakses melalui browser web pada perangkat seperti komputer, tablet, atau smartphone. Berbeda dengan aplikasi mobile yang diunduh dan diinstal secara langsung pada perangkat, aplikasi web dapat diakses secara online tanpa memerlukan instalasi khusus. Pengguna dapat mengaksesnya melalui URL (Uniform Resource Locator) pada browser mereka.

2. 7 Kamera Smartphone

Kamera smartphone telah mengalami evolusi yang signifikan sejak pertama kali diperkenalkan. Pada awalnya, kamera pada ponsel hanya berfungsi sebagai fitur tambahan dengan kualitas yang terbatas. Namun, seiring dengan kemajuan teknologi, kamera smartphone kini mampu menghasilkan foto dengan kualitas yang setara dengan kamera profesional. Produsen ponsel pintar terus berinovasi untuk memperbaiki kualitas gambar, menambahkan fitur-fitur canggih seperti mode malam, zoom optik, dan stabilisasi gambar [16].

Kamera smartphone bekerja dengan menggunakan teknologi yang kompleks namun mudah dimengerti. Proses dimulai ketika pengguna menekan tombol pengambil gambar atau memulai aplikasi kamera di ponsel pintar. Saat itu, lensa di belakang ponsel akan membuka dan sensor gambar di dalamnya akan mulai menerima cahaya dari objek. Selanjutnya, hasil foto ditampilkan di layar ponsel pengguna sehingga pengguna dapat melihat gambar yang ingin diambil atau rekam. Selama proses ini, beberapa ponsel juga menggunakan teknologi stabilisasi gambar untuk mengurangi getaran dan goyangan yang mungkin terjadi, menghasilkan gambar yang lebih jelas dan tajam. Hasil foto yang diambil dari kamera smartphone biasa berformat jpg [16].

2. 8 Mikrofon Smartphone

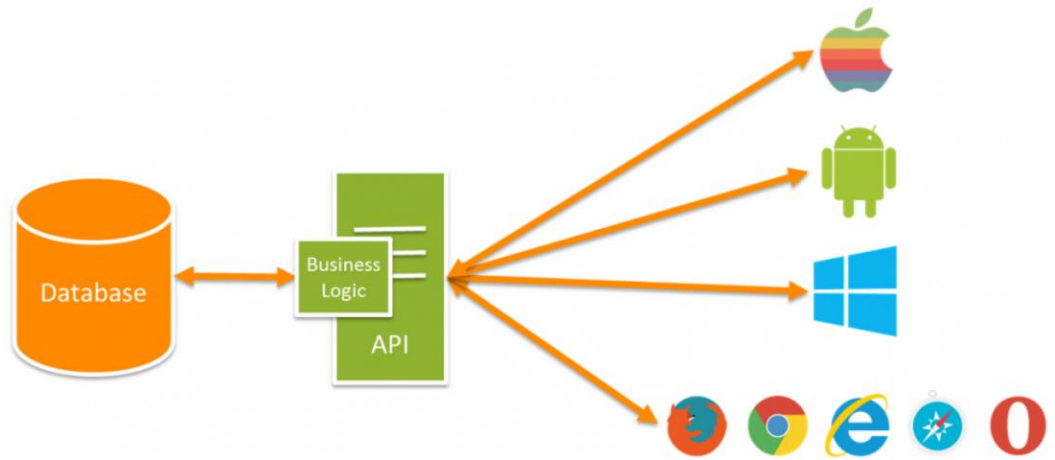
Mikrofon pada smartphone adalah komponen penting yang memungkinkan perangkat untuk merekam suara dan memfasilitasi komunikasi. Setiap smartphone

dilengkapi dengan mikrofon internal yang dirancang untuk menangkap suara dengan jelas, baik saat melakukan panggilan telepon, merekam video, atau menggunakan asisten suara. Mikrofon ini biasanya terletak di bagian bawah dekat port pengisian atau di depan dekat speaker [19]. Teknologi mikrofon telah berkembang seiring waktu, dengan beberapa smartphone kini memiliki beberapa mikrofon untuk mengurangi kebisingan latar belakang dan meningkatkan kualitas suara, terutama lingkungan yang bising atau saat merekam audio dari jarak jauh.

2. 9 Aplication Programming Interface

Aplication Programming Interface (API) adalah seperangkat aturan dan protokol yang memungkinkan aplikasi untuk berkomunikasi satu sama lain. API memungkinkan pengembang untuk menggunakan fungsionalitas atau layanan dari aplikasi atau platform lain tanpa perlu mengetahui atau memahami detail internalnya. API menyediakan titik-titik akses yang telah ditentukan untuk memudahkan pertukaran data atau permintaan layanan antar sistem yang berbeda. Dengan demikian, API berperan sebagai jembatan yang menghubungkan aplikasi dan memungkinkan interaksi yang efisien serta koordinasi antara berbagai komponen perangkat lunak [20].

API dapat memiliki berbagai jenis, termasuk RESTful API, SOAP API, dan GraphQL, yang masing-masing memiliki cara kerja dan protokol komunikasi yang berbeda. Pengembang menggunakan API untuk mengintegrasikan fungsi-fungsi eksternal ke dalam aplikasi mereka, mengakses data dari sumber eksternal, atau menyediakan layanan mereka kepada aplikasi lain. API menjadi fondasi penting dalam pengembangan perangkat lunak modern, memungkinkan kolaborasi antara aplikasi dan sistem yang berbeda secara efisien, dan mendukung ekosistem perangkat lunak yang terhubung secara luas [20]. Gambaran umum cara kerja API dapat dilihat pada Gambar 2.1 berikut.



Gambar 2. 1 Cara Kerja API

Sumber: https://www.codepolitan.com/mengenal-apa-itu-web-api-5a0c2855799c8

Untuk melakukan request kepada API client harus memenuhi beberapa syarat tertentu sebagai berikut [20]:

1. URL

URL merupakan alamat atau lokasi yang digunakan untuk mengidentifikasi dan menentukan lokasi sumber daya di internet. URL memungkinkan pengguna dan perangkat lunak untuk merujuk atau mengakses berbagai jenis sumber daya, seperti halaman web, gambar, file teks, atau layanan online lainnya.

2. Method

Method dalam API menentukan jenis permintaan atau aksi yang dapat dilakukan oleh klien (pengguna atau aplikasi yang menggunakan API) terhadap server atau sumber daya yang terdapat dalam API tersebut. Beberapa metode umum yang digunakan dalam API HTTP termasuk:

a. GET: digunakan untuk mengambil informasi atau data dari sumber daya.
b. POST: digunakan untuk mengirimkan data baru atau membuat sumber daya baru di server.
c. PUT: Digunakan untuk memperbarui atau membuat sumber daya di server.
d. DELETE: Digunakan untuk menghapus sumber daya di server.

e. PATCH: Digunakan untuk memperbarui sebagian dari sumber daya di server.

3. Headers

Headers menyediakan metadata yang membantu server dan klien berkomunikasi dan memahami bagaimana data harus diinterpretasikan atau diproses. Dalam permintaan HTTP, headers umumnya berisi informasi seperti tipe konten yang diinginkan, jenis bahasa yang diinginkan, serta informasi tentang klien yang membuat permintaan. Contoh headers dalam permintaan HTTP termasuk "Accept" untuk menentukan jenis konten yang diinginkan, dan "User-Agent" untuk mengidentifikasi jenis dan versi perangkat lunak klien yang digunakan.

4. Body

Body berada di bawah headers dan terdiri dari konten yang sesuai dengan jenis media atau tipe konten yang ditentukan dalam header "Content-Type". Dalam permintaan HTTP, body dapat berisi data yang akan dikirimkan ke server. Misalnya, dalam permintaan POST untuk mengirimkan formulir, data formulir tersebut dapat berada di dalam body. Dalam permintaan GET, biasanya tidak ada body, karena data biasanya dikirimkan melalui parameter URL.

2. 10 Replicate API

Replicate adalah platform yang memungkinkan untuk menjalankan model AI open-source. Pengguna dapat menggunakan API yang disediakan oleh Replicate untuk mengintegrasikan model seperti LLaVA ke dalam aplikasi pengguna atau bahkan melatih ulang (fine-tune) model dengan dataset pengguna sendiri yang memudahkan pengembangan dan penerapan model AI tanpa perlu infrastruktur komputasi yang kompleks [21].

Untuk menggunakan Replicate, pengguna harus membuat akun replicate untuk mendapatkan token yang akan digunakan untuk menjalankan model misalnya dalam penelitian ini menggunakan model LLaVA, setelah token didapatkan maka pengguna dapat melakukan request kepada API Replicate dengan format JSON, format JSON tersebut berisi “version” yang berisi nomor versi dari model yang akan digunakan, lalu data berupa objek “input” yang berisi “image”, yang

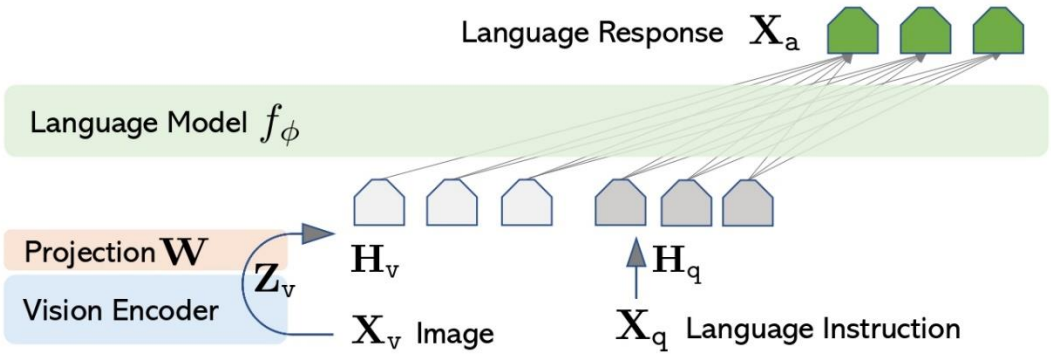
merupakan sumber gambar yang akan diproses dan “prompt”, yang berisi pertanyaan untuk sumber gambar yang dikirimkan. Setelah melakukan request maka API akan mengirimkan response berupa JSON. Contohnya ketika melakukan request dengan parameter “version”:“01359160a4cff57c6bxxx”, lalu “input”: {"image": "https://Cat_August_2010-4.jpg","prompt": "what is that?"}. Dari hasil request tersebut akan didapatkan JSON objek yang berisi “ide” yang merupakan hasil pemrosesan gambar beserta pertanyaan yang diberikan, contohnya {"id": "gth1vveg81rgg0cf1xnawvffhr", ..., ...} yang di mana Id tersebut akan digunakan untuk melakukan request sekali lagi dengan metode GET dengan url https://api.replicate.com/v1/predictions/{id}, dari request tersebut akan didapatkan data JSON objek, yang di mana pada objek tersebut terdapat array “output” berisi hasil dari proses request sebelumnya. Contohnya { ..., "output": ["The ","image ","features ","a ","cat ","laying ","on ","a ","white ","surface, "], ...} [22].

Replicate menawarkan berbagai model yang telah dibuat oleh komunitas dan siap digunakan dalam produksi, termasuk model untuk menghasilkan gambar, teks, dan pemulihan suatu gambar. Replicate sangat berguna bagi pengembang yang ingin mengintegrasikan layanan AI ke dalam aplikasi atau proyek mereka tanpa harus membangun infrastruktur machine learning yang kompleks.

2. 11 LLaVA

LLaVA, yang merupakan singkatan dari Large Language and Vision Assistant, adalah sebuah model yang menggabungkan kemampuan pemrosesan bahasa alami (NLP) dengan pengolahan gambar. Model ini dikembangkan untuk menangani tugas-tugas yang melibatkan pemahaman baik teks maupun gambar, sehingga LLaVA menjadi model yang cocok untuk berbagai aplikasi yang memerlukan analisis multimodal. Model LLaVA dapat menjawab pertanyaan tentang gambar, mendeskripsikan gambar, dan memahami konteks dari teks yang berhubungan dengan gambar. Salah satu kemampuan utama LLaVA adalah menghasilkan deskripsi yang beragam dan detail tentang gambar yang diberikan, ini termasuk mendeskripsikan objek-objek dalam gambar, aksi yang terjadi, dan konteks umum dari gambar tersebut. LLaVA dapat menjawab pertanyaan yang

berkaitan dengan gambar yang diberikan, misalnya, pengguna dapat mengunggah gambar dan menanyakan detail spesifik mengenai apa yang ada di dalam gambar tersebut [22], selain itu model LLaVA juga mendukung berbagai bahasa salah satunya yaitu bahasa Indonesia. Berikut adalah arsitektur dari model LLaVA pada gambar 2.2.



Gambar 2. 2 Arsitektur LLaVA

Sumber: https://huggingface.co/datasets/huggingface/documentation-images/resolve/main/transformers/model_doc/llava_architecture.jpg

Penjelasan mengenai arsitektur pada model LLaVA adalah sebagai berikut [22]:

1. Dasar dari arsitektur LLaVa adalah encoder visual CLIP yang sudah terlatih, khususnya varian ViT-L/14. Komponen ini memproses gambar input (Xv) melalui lapisan Transformer untuk mengekstrak fitur (Zv), sehingga memungkinkan model memahami informasi visual secara akurat.
2. Model Bahasa (Vicuna): Kemampuan linguistik LLaVa bergantung pada Vicuna, yaitu varian dari LLM yang dilambangkan dengan fϕ. Vicuna memahami dan menghasilkan respons bahasa (Xa) berdasarkan instruksi bahasa input (Xq).
3. Proyeksi Linier: Komponen ini, yang diwakili oleh matriks yang dapat dilatih (W) dan ini berfungsi sebagai jembatan antara fitur visual (Zv) serta ruang penyematan model bahasa. Komponen ini mengubah fitur visual menjadi token visual (Hv) dan menyelaraskannya dengan ruang penyematan kata model bahasa untuk memfasilitasi percakapan multimodal.

Biaya penggunaan model LLaVA-13B di Replicate API yang dijalankan pada GPU Nvidia A40 dengan tarif sekitar $0.000725 per detik, waktu proses untuk model ini sangat bervariasi tergantung inputannya, dengan begitu dapat dihitung berdasarkan durasi penggunaannya yaitu dengan proses biasanya selesai dalam waktu 10 detik, jadi bisa diasumsikan untuk melakukan 1 kali request maka Replicate API membutuhkan sekitar $0,007 atau sekitar Rp.115.

Untuk menggunakan model LLaVA melalui Replicate API menggunkan android studio melibatkan beberapa langkah, berikut langkah-langkah untuk menggunakan LLaVA melalui Replicate API:

a. Membuat model untuk Request dan Response

```
public class PredictionRequest {
    private String version;
    private Inputs inputs;

    // Constructor, getters and setters
}

public class Inputs {
    private String input; // atau tipe data sesuai dengan kebutuhan LLaVA

    // Constructor, getters and setters
}

public class PredictionResponse {
    // Definisikan sesuai dengan struktur response dari API
}
```

b. Membuat request

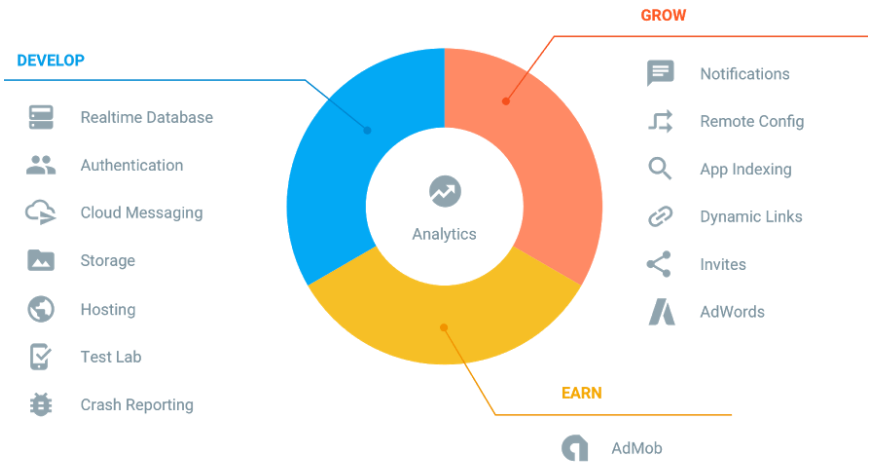
```
PredictionRequest request = new PredictionRequest();
request.setVersion("MODEL_VERSION"); // Sesuaikan dengan versi model yang digunakan
Inputs inputs = new Inputs();
inputs.setInput("YOUR_INPUT"); // Sesuaikan dengan input yang diperlukan oleh model
request.setInputs(inputs);

Call<PredictionResponse> call = apiService.createPrediction(request);
call.enqueue(new Callback<PredictionResponse>() {
    @Override
    public void onResponse(Call<PredictionResponse> call, Response<PredictionResponse>
response) {
        if (response.isSuccessful()) {
            PredictionResponse prediction = response.body();
            // Lakukan sesuatu dengan response
        } else {
            // Tangani error response
        }
    }

    @Override
    public void onFailure(Call<PredictionResponse> call, Throwable t) {
        // Tangani kegagalan request
    }
});
```

Model LLaVA cocok digunakan untuk membangun aplikasi chatbot

berdasarkan gambar atau juga aplikasi yang dapat memberikan deskripsi suatu gambar. Penggunaan model ini memudahkan pengembang dalam membangun aplikasi berbasis AI tanpa memerlukan pembangunan *mechine learning* yang rumit.

2. 12 Firebase

Firebase adalah platform pengembangan aplikasi mobile dan web yang menyediakan berbagai layanan dan alat untuk memudahkan pengembang dalam membangun dan mengelola aplikasi. Salah satu fitur utama Firebase adalah penyimpanan data real-time yang memungkinkan aplikasi untuk menyinkronkan data secara otomatis antara perangkat pengguna [23]. Firebase didesain berdasarkan tiga prinsip utama yaitu Develop, Grow, dan Earn. Informasi lebih lanjut dapat ditemukan pada Gambar 2.2



Gambar 2. 3 Pilar Firebase

Sumber:

https://miro.medium.com/v2/resize:fit:888/1*3c_CO9t1Ysp5axxHk96XUw.png

2. 12. 1 Firebase Storage

Firebase Storage merupakan layanan penyimpanan yang disediakan oleh Google dan digunakan terutama untuk data yang dihasilkan oleh pengguna, seperti audio, gambar, dan video Layanan ini memberikan cara sederhana untuk

mengunggah dan mengunduh file secara aman hanya dengan beberapa baris kode. Firebase Storage dibangun di atas infrastruktur Google Cloud yang cepat dan aman untuk para pengembang aplikasi yang perlu menyimpan dan menyajikan konten yang dihasilkan oleh pengguna, seperti foto atau video. Pengguna dapat menggunakan SDK Admin Firebase untuk mengelola dan membuat URL file yang akan diunduh serta menggunakan API Google Cloud Storage untuk mengakses file pengguna [23].

Dengan menggunakan firebase storage, Dengan Firebase Storage, pengguna dapat mengunggah dan mengunduh berkas, serta mengaksesnya melalui berbagai platform dan perangkat dengan menggunakan SDK resmi Firebase. Selain itu, Firebase Storage juga menyediakan fitur-fitur tambahan seperti penanganan akses keamanan, monitoring operasi berkas, dan integrasi dengan fitur lain dalam ekosistem Firebase seperti Firebase Authentication.

2. 13 Text to Speech

Text to Speech adalah teknologi yang memungkinkan konversi teks tertulis menjadi suara atau ucapan yang dapat didengar oleh pengguna. Dengan bantuan algoritma dan Natural Language Processing, TTS mengubah kata-kata, kalimat, atau teks yang lebih panjang menjadi suara yang alami dan mudah didengar oleh pengguna. Teknologi TTS menyediakan aksesibilitas bagi orang-orang dengan gangguan penglihatan, memungkinkan mereka mendengarkan konten alih-alih membacanya. Beberapa perangkat lunak TTS yang populer antara lain Google Text-to-Speech, Microsoft Azure Text to Speech, Amazon Polly, AppleVoice, Readspeaker, dan Speaktor. Speaktor, misalnya, mengubah teks menjadi suara dengan bantuan kecerdasan buatan [24].

Teknologi TTS memungkinkan pembaca layar untuk mengubah teks menjadi suara, membantu orang tunanetra mengakses dan memahami konten yang ditampilkan di layar komputer, ponsel, atau perangkat lainnya. Asisten virtual seperti Siri, Google Assistant, dan Alexa menggunakan TTS untuk memberikan respon suara terhadap pertanyaan atau perintah pengguna. Ini membantu dalam interaksi manusia-mesin yang lebih intuitif dan efisien.

Kemajuan dalam teknologi TTS terus membuka peluang baru untuk memperluas aksesibilitas, meningkatkan pengalaman pengguna, dan menghasilkan inovasi dalam berbagai bidang. Pengembangan dalam teknologi sintesis suara telah menghasilkan suara yang lebih alami dan menyerupai manusia. Ini meningkatkan kemampuan TTS untuk menyampaikan pesan dengan lebih jelas dan emosional, meningkatkan pengalaman pengguna.

2. 14 Speech to Text

speech to text adalah teknologi *speech recognition* yang memungkinkan konversi bahasa lisan menjadi teks melalui komputasi linguistik. Ini sering disebut sebagai *speech recognition* atau *speech recognition computer*. Aplikasi, alat, dan perangkat tertentu dapat mentranskripsi aliran audio secara real-time untuk menampilkan teks dan bertindak berdasarkan suara tersebut. Teknologi ini bekerja dengan mendengarkan audio dan memberikan transkrip yang dapat diedit secara verbatim pada perangkat yang diberikan. Program komputer menggunakan algoritma linguistik untuk mengurutkan sinyal auditori dari kata-kata yang diucapkan dan mentransfer sinyal tersebut ke dalam teks menggunakan karakter yang disebut Unicode [25].

Penerapan speech to text telah berkembang dari penggunaan sehari-hari di telepon di rumah menjadi aplikasi di industri seperti pemasaran, perbankan, dan medis. Aplikasi pengenalan suara menunjukkan bagaimana teknologi suara ke teks dapat meningkatkan efisiensi tugas-tugas sederhana dan meluas ke tugas yang secara tradisional dilakukan oleh manusia.

2. 15 Android Studio

Android Studio adalah lingkungan pengembangan terpadu (IDE) yang dikembangkan oleh Google untuk memudahkan pengembangan aplikasi Android. IDE ini menyediakan berbagai fitur seperti editor kode yang canggih, pemantauan kinerja aplikasi, emulator perangkat Android, dan integrasi dengan berbagai alat pengembangan Android. Dengan Android Studio, para pengembang dapat membuat, menguji, dan mendebug aplikasi Android secara efisien. IDE ini juga mendukung penggunaan Kotlin sebagai bahasa pemrograman resmi selain Java,

serta menyediakan berbagai template dan alat bantu untuk mempercepat proses pengembangan aplikasi mobile dengan platform Android [26].

2. 16 Kotlin

Kotlin adalah bahasa pemrograman yang modern dan ekspresif yang dirancang untuk berjalan di atas Java Virtual Machine (JVM). Dikembangkan oleh JetBrains, Kotlin menghadirkan sejumlah fitur inovatif yang mempermudah pengembangan perangkat lunak. Salah satu fitur terkemuka Kotlin adalah sintaksis yang ringkas dan bersih, yang memungkinkan pengembang menulis kode dengan lebih mudah dan efisien dibandingkan dengan Java [27].

Keunggulan lain dari Kotlin adalah interoperabilitas yang baik dengan Java. Hal ini memungkinkan pengembang untuk menggunakan kode Kotlin dan Java secara bersamaan dalam proyek yang sama, membuat proses migrasi dari Java ke Kotlin menjadi lebih lancar. Kotlin juga menawarkan dukungan penuh untuk pemrograman berorientasi objek dan fungsional, memungkinkan pengembang untuk mengadopsi gaya pemrograman yang sesuai dengan kebutuhan proyek. Fitur-fitur modern seperti ekstensi fungsi, lambdas, dan coroutine juga memberikan daya ungkit ekstra untuk pengembangan aplikasi yang responsif dan efisien [27].

2. 17 JSON

JSON (JavaScript Object Notation) adalah format yang digunakan untuk pertukaran data yang ringan, mudah dibaca, dan mudah dipahami oleh manusia maupun mesin. JSON terdiri dari pasangan nama dan nilai yang disusun secara terstruktur dalam format teks. Setiap pasangan ini terdiri dari sebuah nama (dalam bentuk string) dan nilai yang terkait, yang bisa berupa string, angka, objek, array, boolean, atau null. JSON sering digunakan dalam aplikasi web dan pengembangan perangkat lunak modern untuk mengirim dan menerima data antara server dan klien, karena kemampuannya yang serbaguna dan kepopulerannya dalam lingkungan pengembangan web [28].

JSON memungkinkan pengembang untuk dengan mudah mengirim dan menerima data terstruktur dalam format yang konsisten, memfasilitasi interoperabilitas antara berbagai platform dan bahasa pemrograman. Karena

sifatnya yang ringan dan efisien, JSON juga banyak digunakan dalam pengembangan aplikasi mobile dan Internet of Things (IoT), di mana sumber daya terbatas menjadi faktor penting. Dengan kemampuannya yang mendukung struktur data kompleks dan fleksibilitas dalam representasi nilai, JSON telah menjadi salah satu standar de facto dalam pertukaran data di lingkungan pengembangan perangkat lunak modern [28].

2. 18 Unified Modeling Language (UML)

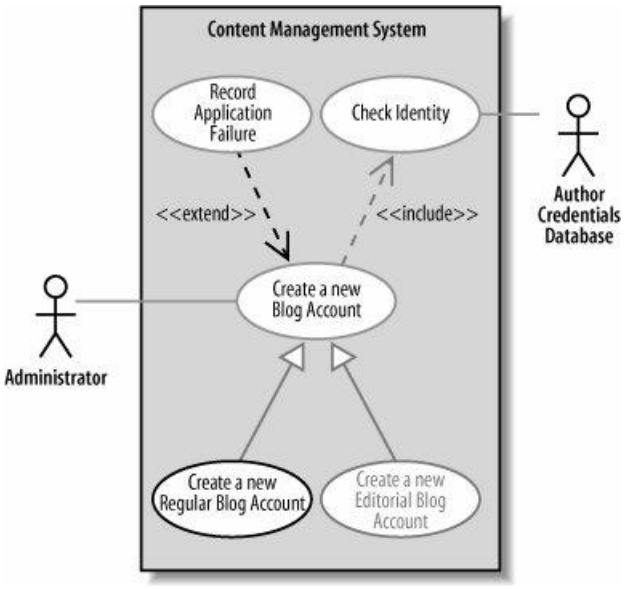
Unified Modeling Language (UML) adalah suatu bahasa pemodelan yang digunakan dalam pengembangan perangkat lunak untuk mendokumentasikan, merancang, dan memodelkan sistem perangkat lunak. UML menyediakan notasi standar yang dapat dipahami oleh berbagai pemangku kepentingan, seperti pengembang perangkat lunak, analis bisnis, dan manajer proyek [29]. Dengan notasi yang ter standarisasi, UML membantu mengurangi ambiguitas dan memudahkan tim pengembang dalam merancang, mengimplementasikan, dan memelihara sistem. UML juga mendukung konsep pemodelan berorientasi objek, yang memungkinkan pengembang untuk memandang sistem sebagai kumpulan objek yang saling berinteraksi [30].

Berdasarkan penjelasan yang diperoleh dari beberapa sumber referensi, dapat ditarik kesimpulan bahwa UML (Unified Modeling Language) memegang peran penting sebagai alat bantu dalam proses pemodelan sistem. UML memudahkan keterhubungan langsung antara pengembang sistem, sehingga meningkatkan efektivitas dalam pembangunan sistem. UML memiliki berbagai jenis diagram, empat di antaranya yakni Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram.

2. 18. 1 Use Case Diagram

Diagram use case adalah jenis diagram dalam teknik pemodelan sistem yang digunakan untuk menggambarkan fungsionalitas sistem dari sudut pandang pengguna atau aktor eksternal. Usecase diagram mengidentifikasi dan menggambarkan interaksi antara aktor-aktor eksternal dengan fitur-fitur atau fungsi-fungsi utama yang dimiliki oleh sistem. Setiap use case, yang mewakili suatu

fungsi atau aktivitas khusus, digambarkan sebagai oval dan terhubung dengan aktor-aktor yang terlibat dalam interaksi tersebut melalui garis-garis. Contoh dari use case diagram dapat dilhat pada gambar 2.3.



Gambar 2. 4 Contoh Usecase

Sumber: Learning UML 2.0 [30]

2. 18. 2 Use Case Scenario

Use Case Scenario adalah uraian langkah-langkah yang menggambarkan bagaimana aktor (pengguna atau sistem lain) berinteraksi dengan sistem untuk mencapai tujuan tertentu. Use Case Scenario menguraikan alur masukan pengguna dan menetapkan jalur yang berhasil serta gagal untuk mencapai tujuan yang memungkinkan tim untuk lebih memahami apa yang dilakukan sistem dan bagaimana sistem berkinerja. Berikut ini adalah salah satu contoh Sequence diagram yang dapat dilihat pada Gambar 2.4.

Use case name	Create a new Blog Account	
Related Requirements	Requirement A.1.	
Goal In Context	A new or existing author requests a new blog account from the `Administrator`.	
Preconditions	The system is limited to recognized authors and so the author needs to have appropriate proof of identity.	
Successful End Condition	A new blog account is created for the author.	
Failed End Condition	The application for a new blog account is rejected.	
Primary Actors	Administrator.	
Secondary Actors	Author Credentials Database.	
Trigger	The `Administrator` asks the CMS to create `a new blog account`.	
Main Flow	**Step**	**Action**
	1	The `Administrator` asks the system to create a new blog account.
	2	The `Administrator` selects an account type.
	3	The `Administrator` enters the author's details.
	4	The author's details are verified using the Author Credentials Database.
	5	The new blog account is created.
	6	A summary of the new blog account's details are emailed to the author.
Extensions	**Step**	**Branching Action**
	4.1	The Author Credentials `Database` does not verify the author's details.
	4.2	The author's new blog account application is rejected.

Gambar 2. 5 Contoh Use Case Skenario

Sumber: Learning UML 2.0 [30]

Dari gambar di atas terdapat beberapa elemen yang ada di dalam use case scenario tersebut. Untuk penjelasan dari elemen-elemen tersebut adalah sebagai berikut:

1. Use Case Name, use case name merupakan nama deskriptif yang digunakan untuk mengidentifikasikan skenario use case tertentu.
2. Related Requirements, related requirements merupakan kode identifikasi mengenai persyaratan mana yang dipenuhi oleh use case ini.
3. Goal In Context, goal in context merupakan tujuan utama yang harus dipenuhi dari suatu use case.
4. Preconditions, preconditions merupakan kondisi yang harus dipenuhi sebelum use case dimulai.
5. Successful End Condition, successful end condition merupakan hasil atau

kondisi sistem yang seharusnya jika use case berhasil dijalankan.

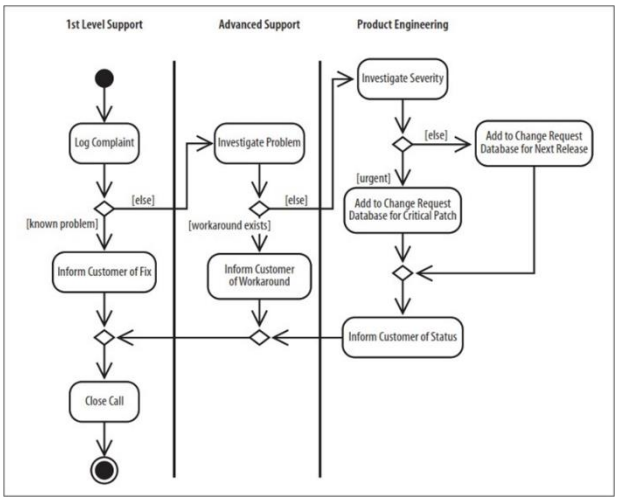
6. Failed End Condition, failed end conditions merupakan hasil kondisi sistem yang seharusnya jika proses dalam use case terjadi kesalahan.
7. Primary Actors, primary actors merupakan pengguna atau entitas utama yang terlibat dalam sebuah use case, aktor ini biasanya yang memicu atau secara langsung menerima informasi dari eksekusi use case.
8. Secondary Actors, secondary actors merupakan pengguna atau entitas yang terlibat dalam sebuah use case. Berbeda dari primary actors, aktor ini hanya menjadi bagian pendukung saja dan bukan pemain utama dalam sebuah use case.
9. Trigger, trigger merupakan kondisi yang dipicu oleh aktor yang menyebabkan use case dieksekusi.
10. Main Flow, main flow merupakan bagian yang digunakan untuk mendeskripsikan setiap langkah penting dalam eksekusi use case.
11. Extensions, extensions merupakan bagian yang digunakan untuk mendeskripsikan langkah – langkah alternatif apa pun dari yang dijelaskan di alur utama.

Dalam setiap Use Case Scenario, langkah-langkah yang diambil oleh pengguna dan respons sistem didefinisikan dengan jelas, sering kali dalam bentuk aliran kerja (workflow) atau diagram. Ini membantu pengembang memahami persyaratan fungsional dan non-fungsional dari sistem yang akan dibangun, serta memastikan bahwa solusi yang dihasilkan sesuai dengan kebutuhan pengguna.

2. 18. 3 Activity Diagram

Activity diagram adalah salah satu jenis diagram UML (Unified Modeling Language) yang digunakan untuk menggambarkan alur kerja atau aktivitas suatu sistem atau proses bisnis. Diagram ini memberikan gambaran visual tentang bagaimana berbagai aktivitas saling terkait dan bagaimana alur kontrol di antara

aktivitas-aktivitas tersebut. Dalam activity diagram, aktivitas direpresentasikan oleh kotak oval, sedangkan aliran antar aktivitas diindikasikan oleh panah. Diagram ini membantu para pengembang perangkat lunak dan pemangku kepentingan lainnya untuk memahami secara lebih rinci bagaimana suatu proses atau sistem beroperasi. Activity diagram juga dapat digunakan untuk mengidentifikasi hubungan antara aktivitas-aktivitas, menunjukkan percabangan keputusan, dan menggambarkan paralelisme dalam eksekusi aktivitas, menjadikannya alat yang efektif untuk merancang dan memodelkan sistem yang kompleks. Contoh dari activity diagram dapa dilihat pada gambar 2.5.



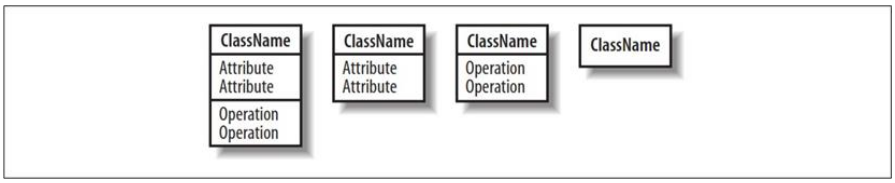
Gambar 2. 6 Activity Diagram

Sumber: : Learning UML 2.0 [30]

2. 18. 4 Class Diagram

Class Diagram adalah representasi visual dari struktur kelas dan hubungan antara kelas-kelas dalam suatu sistem perangkat lunak. Ini menggambarkan entitas-entitas yang terlibat dalam sistem, beserta atribut-atribut dan metode-metode yang dimilikinya. Setiap kelas direpresentasikan oleh sebuah persegi panjang yang berisi nama kelas di bagian atas, diikuti oleh daftar atribut dan metode. Garis-garis panah menghubungkan kelas-kelas untuk menunjukkan hubungan antar mereka, seperti asosiasi, pewarisan, dan agregasi. Class diagram membantu pengembang perangkat lunak memahami struktur sistem secara keseluruhan, memfasilitasi perancangan yang efektif, dan memungkinkan komunikasi yang jelas antar anggota tim

pengembangan. Contoh dari class diagram dapat dilihat pada gambar 2.6.

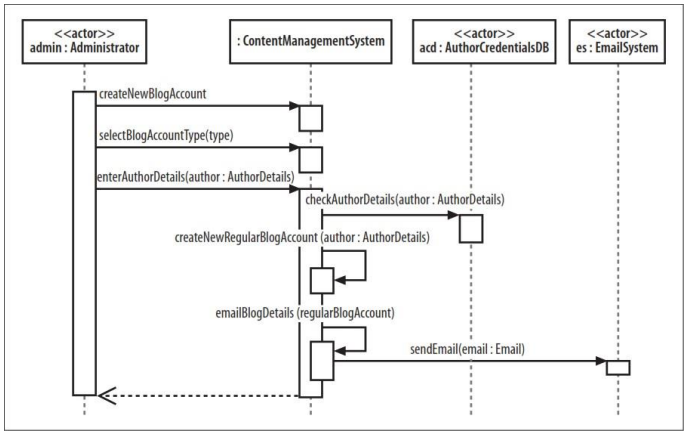


Gambar 2. 7 Contoh Class Diagram

Sumber: Learning UML 2.0 [30]

2. 18. 5 Sequence Diagram

Sequence Diagram adalah jenis diagram interaksi dalam pemodelan perangkat lunak yang menggambarkan interaksi antara objek-objek dalam suatu sistem pada suatu waktu tertentu. Diagram ini memberikan pandangan visual terhadap alur eksekusi suatu proses atau fungsi dalam sistem secara kronologis, dengan menggambarkan pesan-pesan yang dikirim antar objek. Setiap objek direpresentasikan sebagai kotak vertikal, dan garis-garis horizontal di antaranya menunjukkan pesan-pesan yang dikirim. Keteraturan waktu diekspresikan dari kiri ke kanan, sehingga membantu pengguna memahami bagaimana objek-objek berinteraksi sepanjang waktu. Contoh dari sequence diagram dapat dilihat pada gambar 2.7.



Gambar 2. 8 Contoh Sequence Diagram

Sumber: : Learning UML 2.0 [30]