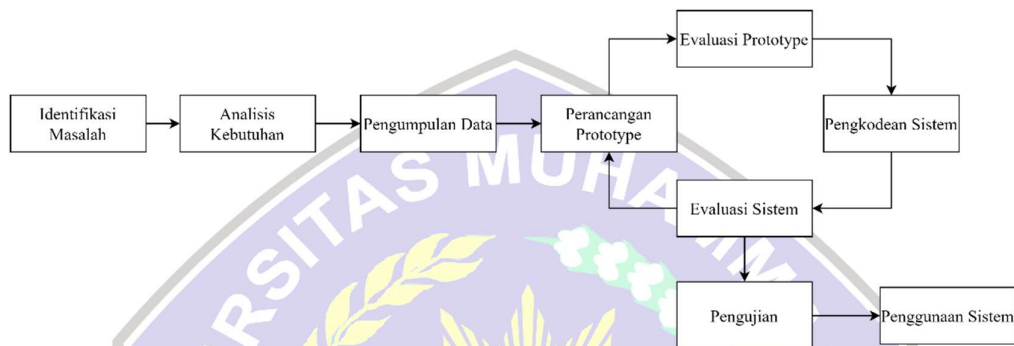


BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Bab ini membahas metodologi penelitian dengan menggunakan metode *prototype*, yang terdiri dari beberapa tahapan utama, yaitu identifikasi masalah, analisis kebutuhan, pengumpulan data, perancangan *prototype*, evaluasi *prototype*, pengkodean sistem, evaluasi sistem, pengujian sistem, dan penggunaan sistem.



Gambar 3.1 Alur Tahapan Penelitian

Penelitian ini menggunakan metode *prototype* yang terdiri dari beberapa tahapan utama. Pada Gambar 3.1, Tahap pertama adalah identifikasi masalah untuk memahami kendala dalam pembelajaran ilustrasi sketsa dan menentukan kebutuhan sistem. Selanjutnya, dilakukan analisis kebutuhan dan pengumpulan *Dataset* sketsa sebagai data pelatihan *model*. *Prototype* sistem dirancang meliputi desain antarmuka pengguna (UI) dan struktur sistem, lalu dievaluasi untuk memastikan kesesuaiannya.

Setelah *prototype* disetujui, sistem dikembangkan melalui pengkodean menggunakan algoritma CNN dengan *framework TensorFlow* serta *Firestore* sebagai *database*. Sistem kemudian dievaluasi untuk mengukur kinerja *model*, diuji untuk memastikan fungsionalitas, dan diimplementasikan dalam lingkungan nyata untuk mendukung pembelajaran ilustrasi sketsa. Semua tahapan dirancang secara terstruktur untuk menghasilkan aplikasi pembelajaran berbasis AI yang solutif.

3.1.1 Identifikasi Masalah

Pada tahap ini, identifikasi masalah berfokus pada tantangan yang dihadapi dalam pembelajaran sketsa, terutama bagi pemula yang kesulitan menggambar pola dengan akurat. Metode pembelajaran tradisional seringkali terbatas pada buku atau demonstrasi manual, yang membuat pengguna kesulitan memahami proporsi, perspektif, dan komposisi garis. Keterbatasan ini mempengaruhi proses belajar karena pengguna tidak mendapatkan umpan balik secara langsung atas kesalahan yang dibuat.

Solusi yang ditawarkan adalah pengembangan aplikasi berbasis teknologi AI dan *Augmented Reality*. Aplikasi ini memungkinkan pengguna untuk menggambar dengan panduan visual secara *real-time* melalui *overlay* AR dan mendapatkan umpan balik otomatis menggunakan algoritma *Convolutional Neural Network* (CNN).

3.1.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk menentukan elemen-elemen penting yang diperlukan dalam pengembangan aplikasi pembelajaran sketsa berbasis AI dan AR. Pada tahap ini, kebutuhan perangkat lunak, perangkat keras, dan *Dataset* diidentifikasi untuk memastikan aplikasi dapat berjalan secara optimal. Dengan pemahaman yang mendalam terhadap kebutuhan sistem, proses pengembangan dapat diarahkan agar sesuai dengan tujuan dan harapan pengguna.

a. Perangkat Lunak

1. Sistem operasi Android untuk mendukung aplikasi yang dikembangkan.
2. *Framework TensorFlow* untuk implementasi CNN.
3. *ARCore* untuk mendukung teknologi *Augmented Reality*.
4. *Supabase* sebagai database untuk menyimpan data pengguna dan hasil analisis sketsa.
5. *Python* sebagai bahasa pemrograman utama untuk pelatihan *model* AI.
6. *Android Studio* untuk pengembangan antarmuka aplikasi.

b. Perangkat Keras

1. *Smartphone* Android dengan spesifikasi minimal: RAM 4GB, prosesor dengan kemampuan AI, dan kamera berkualitas tinggi untuk mendukung AR.
2. Laptop atau komputer dengan spesifikasi minimum: prosesor *intel core i5* atau setara, RAM 8GB, GPU dengan dukungan CUDA untuk pelatihan *model CNN*.

c. *Dataset*

Dataset yang digunakan terdiri dari kumpulan gambar sketsa dalam berbagai kategori, seperti hewan, bunga, dan bentuk geometris. Setiap kategori memiliki langkah-langkah pola (*step*) untuk memandu pengguna, seperti pola dasar, detail tambahan, outline, dan pewarnaan. *Dataset* ini harus diproses dengan teknik seperti *grayscale* atau deteksi tepi untuk meningkatkan efektivitas pelatihan *model*.

3.1.3 Pengumpulan Data

Pengumpulan data dilakukan untuk mendukung pengembangan aplikasi dengan informasi yang relevan dan akurat. Data diperoleh melalui berbagai metode, seperti studi literasi, wawancara, dan observasi, yang dirancang untuk memahami kebutuhan pengguna serta tantangan dalam pembelajaran sketsa. Proses ini bertujuan untuk menyediakan dasar yang kuat bagi desain dan implementasi aplikasi.

a. Studi Literasi

Dilakukan dengan membaca literatur dan penelitian sebelumnya tentang teknologi AI dan AR dalam pembelajaran, algoritma CNN untuk analisis gambar, serta aplikasi berbasis AR di bidang pendidikan.

b. Wawancara

Melibatkan wawancara dengan seniman, seorang pendiri komunitas CokroSketsa yang memiliki pengalaman mengajar menggambar. Wawancara bertujuan untuk memahami tantangan yang dihadapi pengguna dalam pembelajaran sketsa dan kebutuhan fitur yang relevan untuk aplikasi.

c. Observasi

Observasi dilakukan dengan melihat langsung bagaimana pemula belajar menggambar sketsa menggunakan metode tradisional. Tujuan observasi adalah mengidentifikasi kelemahan metode tradisional yang dapat diatasi dengan aplikasi berbasis teknologi.

3.1.4 Objek Penelitian

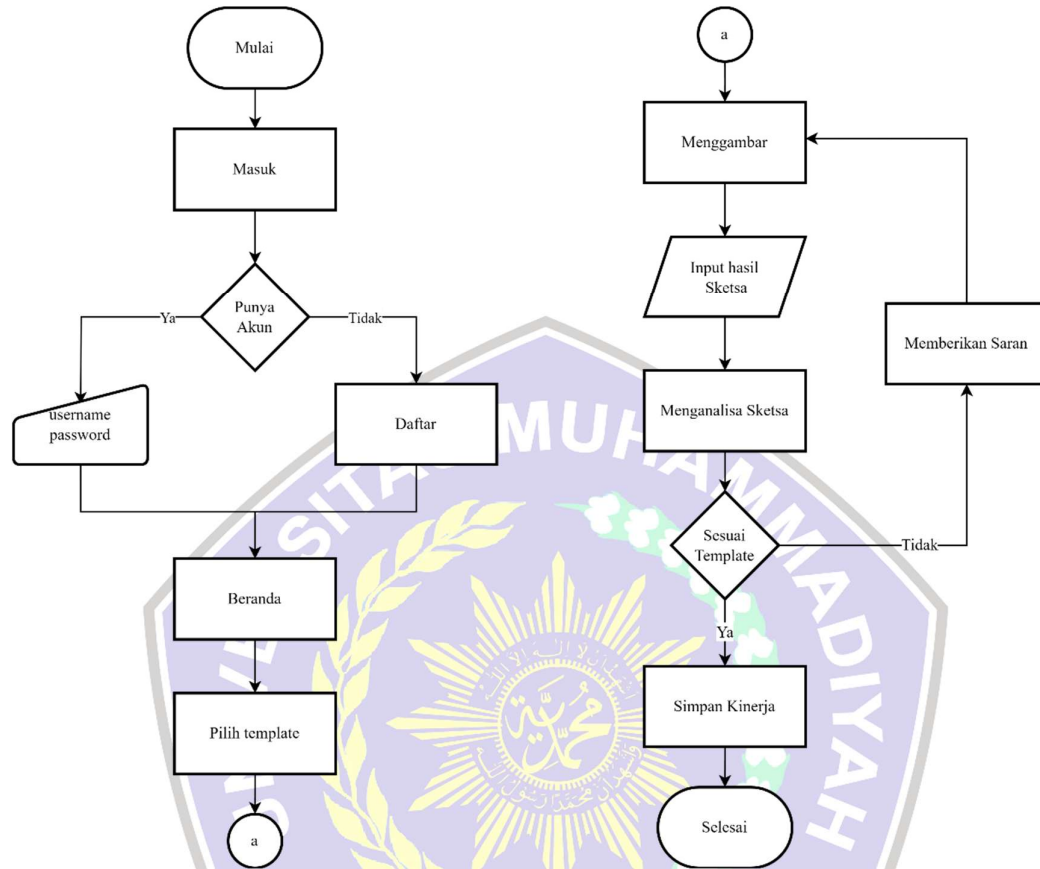
Objek penelitian ini adalah pengembangan aplikasi pembelajaran seni sketsa yang bertujuan untuk membantu pengguna dalam menggambar sketsa menggunakan teknologi *Artificial Intelligence* (AI) dan *Augmented Reality* (AR). Aplikasi ini menggunakan algoritma *Convolutional Neural Network* (CNN) untuk menganalisis dan mendeteksi kesalahan sketsa secara real-time, seperti ketidaksesuaian garis, proporsi, dan bentuk dibandingkan dengan *template* pola gambar yang telah ditentukan.

Augmented Reality (AR) berperan dalam menampilkan *template* gambar sketsa secara overlay di media gambar melalui kamera ponsel, sehingga pengguna dapat mengikuti panduan visual yang lebih interaktif. Selain itu, Integrasi AI melalui CNN memungkinkan sistem memberikan umpan balik otomatis yang membantu pengguna dalam memperbaiki kesalahan sketsa mereka secara lebih cepat dan akurat.

Penelitian ini akan diuji coba secara langsung pada komunitas CokroSketsa Ponorogo sebagai subjek penelitian. Komunitas ini dipilih karena memiliki fokus utama pada pengembangan seni sketsa dan rutin melakukan aktivitas menggambar sketsa bersama. Dengan melibatkan komunitas ini, diharapkan pengujian aplikasi dapat memberikan gambaran nyata terkait efektivitas dan penerimaan teknologi dalam mendukung proses pembelajaran seni sketsa.

3.2 Perancangan Sistem

3.2.1 Flowchart Sistem



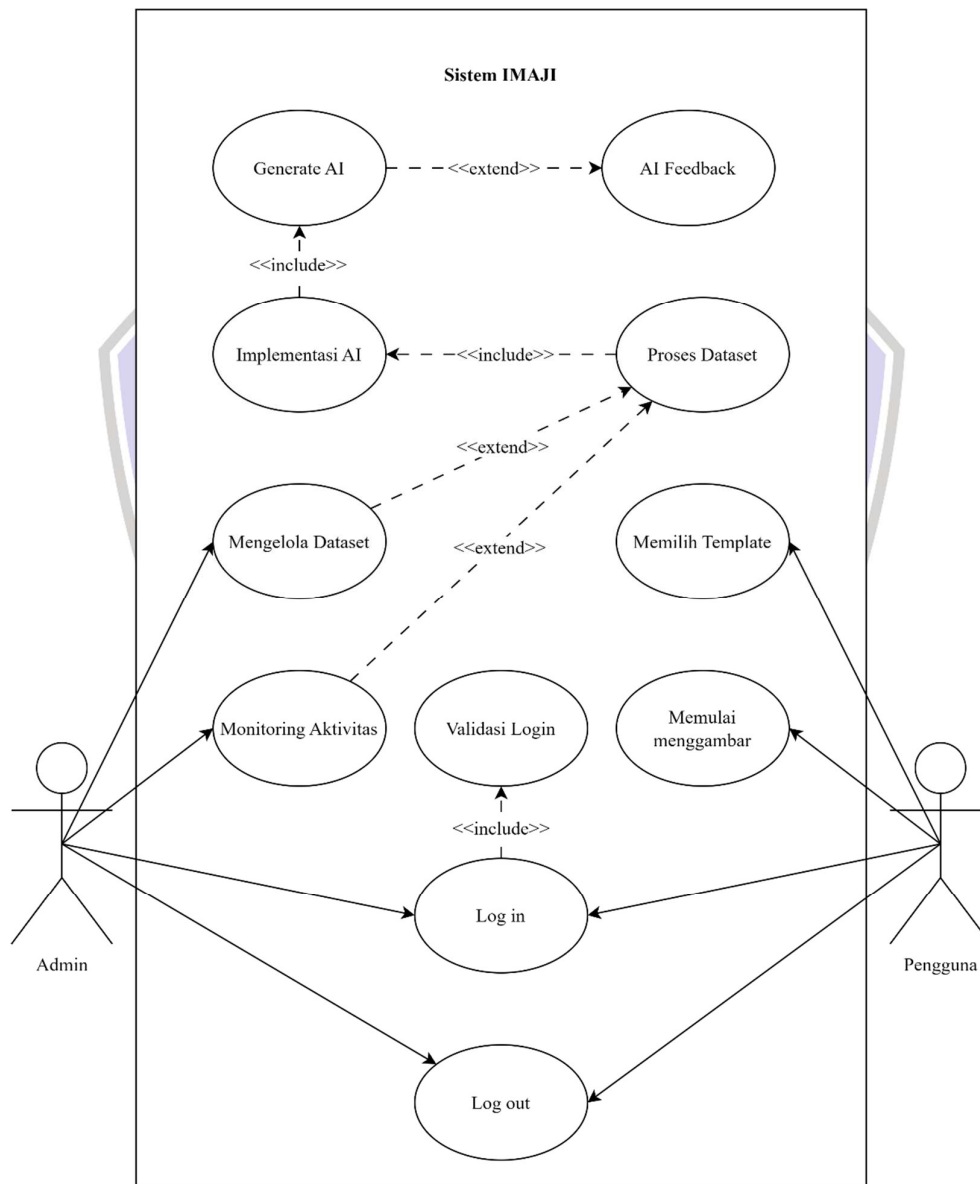
Gambar 3.2 Flowchart Sistem

Diagram alur sistem ini menggambarkan proses interaksi pengguna dengan aplikasi mulai dari awal hingga akhir. Pada Gambar 3.2 Proses dimulai dengan pengguna masuk ke aplikasi. Jika pengguna tidak memiliki akun, mereka diarahkan untuk mendaftar terlebih dahulu. Setelah berhasil masuk, pengguna diarahkan ke beranda untuk memilih *template* sketsa yang diinginkan.

Setelah memilih *template*, pengguna memulai proses menggambar dengan panduan dari aplikasi. Hasil gambar kemudian diunggah melalui fitur input hasil sketsa, di mana sistem akan menganalisis sketsa tersebut untuk menentukan tingkat kesesuaian dengan *template*. Jika hasil analisis menunjukkan ketidaksesuaian, sistem akan memberikan saran kepada pengguna untuk perbaikan.

Jika hasil sketsa sesuai dengan *template*, aplikasi akan menyimpan kinerja pengguna sebagai data yang dapat digunakan untuk evaluasi lebih lanjut. Proses ini berakhir dengan tahap selesai, yang menandakan bahwa semua langkah dalam aplikasi telah dilalui. Diagram ini mencerminkan alur kerja yang sistematis dan dirancang untuk mendukung pengguna dalam meningkatkan keterampilan menggambar mereka.

3.2.2 Diagram Use Case



Gambar 3.3 Use Case Diagram

Rancangan *use case* aplikasi ini melibatkan dua aktor utama, yaitu Pengguna dan *Admin*. Seperti yang terlihat pada Gambar 3.3 Pengguna menggunakan aplikasi untuk berbagai aktivitas seperti menggambar dengan panduan *template*, menerima umpan balik dari AI, dan mengunggah hasil gambar untuk dianalisis oleh sistem. Sedangkan, *Admin* memiliki peran untuk mengelola *template* gambar, pengguna, serta data terkait aplikasi.

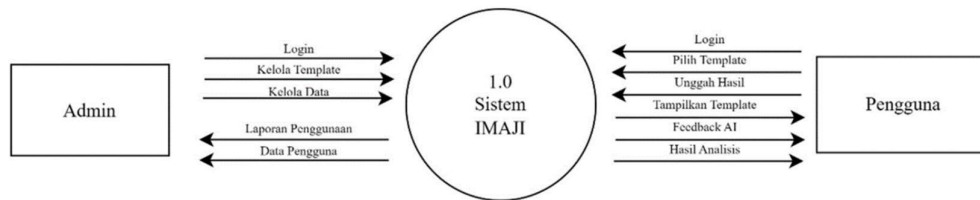
Proses dimulai dengan fitur *Login*, di mana baik Pengguna maupun *Admin* dapat masuk atau mendaftar ke dalam aplikasi menggunakan kredensial masing-masing. Setelah berhasil masuk, *User* dapat melanjutkan dengan memilih *template* yang akan digunakan untuk menggambar. Sistem kemudian memanfaatkan teknologi AR dalam fitur Proses Menggambar dengan AR untuk menampilkan *template* secara visual sebagai panduan menggambar.

Setelah proses menggambar selesai, *User* mengunggah hasil gambarnya melalui fitur Analisis Hasil Gambar, di mana sistem menggunakan algoritma CNN untuk membandingkan hasil gambar dengan *template* yang dipilih. Jika ditemukan kesalahan, sistem memberikan *Feedback* AI berupa saran perbaikan kepada pengguna.

Di sisi lain, *Admin* memiliki kemampuan untuk mengelola *template*, termasuk menambah, memperbarui, atau menghapus *template* gambar. Selain itu, *admin* juga dapat mengelola data pengguna, seperti melihat atau memperbarui informasi pengguna yang terdaftar. Rancangan ini memastikan aplikasi berjalan dengan baik untuk mendukung aktivitas *User* dan mempermudah pengelolaan oleh *Admin*.

3.2.3 Data Flow Diagram (DFD)

a. DFD Level 0 (ConTEXT Diagram)



Gambar 3.4 DFD Level 0

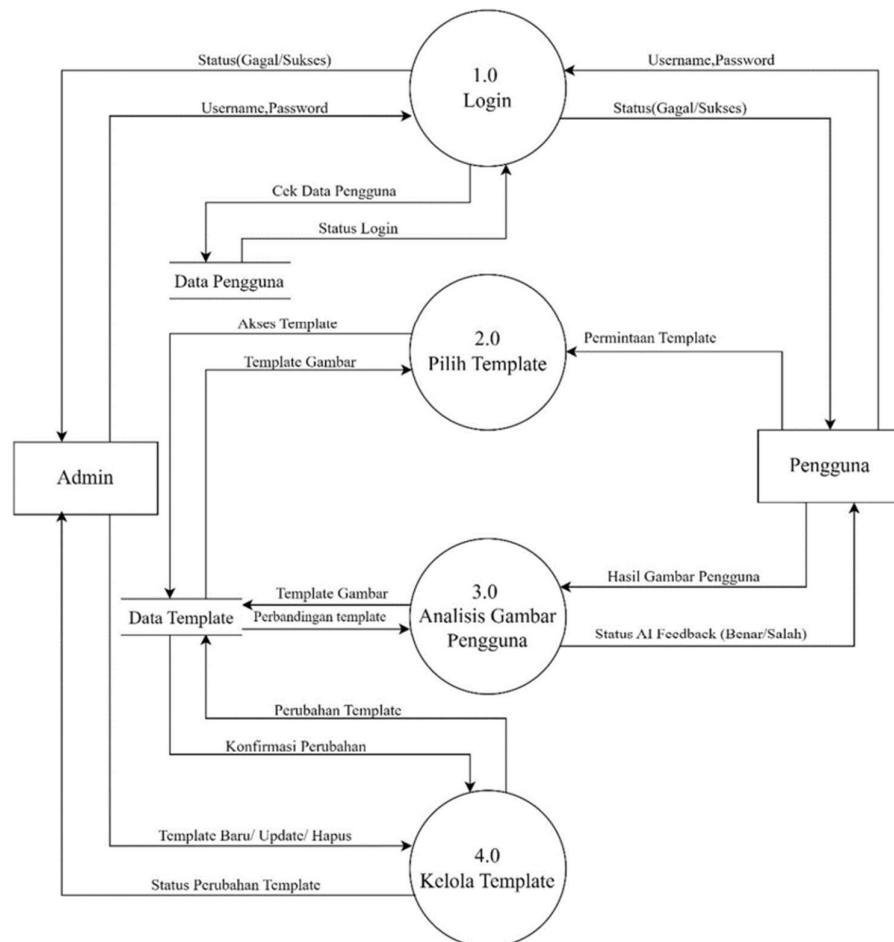
DFD Level 0 untuk Sistem utama menggambarkan hubungan antara entitas eksternal, proses utama, dan alur data. Entitas Eksternal yang terlibat terdiri dari *Pengguna* dan *Admin*, yang berinteraksi langsung dengan Sistem Imaji sebagai proses utama.

User berinteraksi dengan sistem melalui beberapa alur data, seperti Login, Pilih *Template*, dan Unggah Hasil Gambar. Sebagai respon, sistem mengirimkan data kembali ke *User*, seperti Tampilan *Template*, *Feedback* AI, dan Hasil Analisis Gambar. Hal ini mendukung pengalaman pengguna dalam menggambar dan menerima saran perbaikan secara otomatis.

Di sisi lain, *Admin* menggunakan sistem untuk *Login*, Mengelola Data Pengguna. Sebagai keluaran, sistem memberikan Laporan Penggunaan dan Data Pengguna kepada *admin* untuk membantu dalam pengelolaan aplikasi.

Dengan alur data ini, Sistem utama berfungsi sebagai pusat pengolahan data yang mendukung kebutuhan pengguna dan *admin* dalam satu kesatuan proses yang terintegrasi.

b. DFD level 1



Gambar 3.5 DFD level 1

DFD Level 1 merupakan dekomposisi dari proses utama yang ada pada DFD Level 0. Diagram ini menjelaskan secara lebih rinci alur data dan interaksi antara proses, entitas eksternal, dan penyimpanan data dalam sistem. Dijelaskan pada Gambar 3.5, sistem ini memiliki empat proses utama yang saling terhubung, yaitu proses login, pemilihan *template*, analisis gambar pengguna, dan pengelolaan *template*.

Berikut adalah penjelasan masing-masing proses pada DFD Level 1:

1. Proses *Login* (1.0)

Proses ini bertujuan untuk mengautentikasi pengguna dan *admin* yang ingin mengakses sistem. Pengguna atau *admin* memasukkan *Username* dan *password* ke dalam sistem. Sistem akan memvalidasi data yang diterima dengan mencocokkan

informasi yang tersimpan dalam database. Jika data valid, sistem memberikan status "Sukses" dan pengguna diarahkan ke halaman beranda atau fitur yang sesuai. Jika data tidak valid, sistem memberikan status "Gagal" dan meminta pengguna untuk mengulangi proses *login* atau mendaftar jika belum memiliki akun.

2. Proses Pilih *Template* (2.0)

Proses ini digunakan oleh pengguna untuk memilih *template* gambar yang ingin digunakan sebagai panduan menggambar. Sistem menampilkan daftar *template* yang tersedia berdasarkan data yang telah diunggah oleh *admin*. Setelah pengguna memilih *template*, data *template* tersebut dikirimkan ke pengguna untuk digunakan dalam fitur *Augmented Reality* (AR).

3. Proses Analisis Gambar Pengguna (3.0)

Pada proses ini, pengguna mengunggah hasil gambar yang telah dibuat berdasarkan *template* yang dipilih. Sistem akan menganalisis gambar tersebut menggunakan algoritma *Convolutional Neural Network* (CNN). Hasil analisis berupa status kecocokan gambar dengan *template* dan umpan balik berupa rekomendasi atau koreksi diberikan kepada pengguna. Proses ini bertujuan untuk meningkatkan kualitas gambar pengguna dengan memberikan evaluasi berbasis AI.

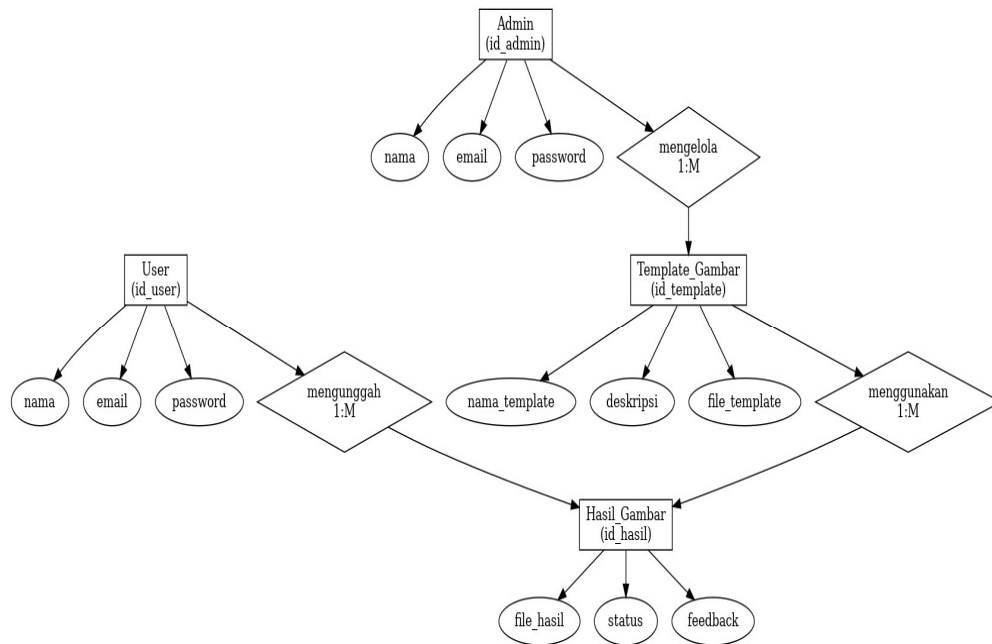
4. Proses Kelola *Template* (4.0)

Proses ini hanya dapat diakses oleh *admin*. Di sini, *admin* memiliki kontrol penuh untuk melakukan pengelolaan *template*, baik itu menambahkan *template* baru, memperbarui, maupun menghapus *template* yang sudah ada. Setiap perubahan yang dilakukan *admin* akan langsung berdampak pada data *template* yang tersedia bagi pengguna. Setelah perubahan dilakukan, sistem akan memberikan konfirmasi bahwa status perubahan berhasil. Proses ini memastikan bahwa hanya *template* yang relevan dan terbaru yang tersedia di sistem.

5. Alur Data

Setiap proses di DFD Level 1 terhubung melalui alur data yang mencakup informasi penting seperti kredensial pengguna, data *template*, hasil analisis, dan laporan sistem. Data yang mengalir dari pengguna dan *admin* ke sistem, maupun sebaliknya, direpresentasikan secara jelas untuk memastikan sistem berjalan sesuai dengan alur kerja yang telah dirancang.

3.2.4 Entity Relationship Diagram (ERD)



Gambar 3.6 Perancangan ERD

Di tahap ini, Pada gambar 3.6 dirancang struktur basis data untuk mendukung fungsionalitas aplikasi. Desain basis data diwujudkan dalam bentuk *Entity-Relationship Diagram* (ERD), yang menggambarkan entitas utama, atribut, serta relasi antar entitas.

Berikut adalah penjelasan dari masing-masing entitas yang terdapat pada ERD:

a. Entitas *Admin*

Entitas *Admin* merepresentasikan pihak pengelola sistem yang bertugas mengelola *template* gambar serta data pengguna dalam aplikasi.

1. Atribut:

- id_admin*: Identitas unik untuk setiap *admin*.
- nama*: Nama *admin*.
- email*: Alamat *email* *admin* untuk login.
- password*: Kata sandi untuk autentikasi.

2. Relasi:

- Admin* memiliki relasi 1:M dengan entitas *Template* Gambar, menunjukkan bahwa satu *admin* dapat mengelola banyak *template*.

b. Entitas *User*

Entitas *User* merepresentasikan pengguna aplikasi yang menggunakan *template* untuk menggambar, mengunggah hasil gambar, dan menerima umpan balik dari sistem.

1. Atribut:

- a. *id_user*: Identitas unik untuk setiap pengguna.
- b. *nama*: Nama pengguna.
- c. *email*: Alamat *email* pengguna untuk login.
- d. *password*: Kata sandi untuk autentikasi.

2. Relasi

- a. *User* memiliki relasi 1:M dengan entitas Hasil Gambar, yang menunjukkan bahwa satu *User* dapat menghasilkan dan mengunggah banyak hasil gambar.
- b. *User* juga memiliki relasi 1:M dengan entitas *Template* Gambar, yang menunjukkan bahwa satu *User* dapat menggunakan banyak *template* untuk menggambar.

b. Entitas *Template* Gambar

Entitas *Template* Gambar menyimpan informasi terkait *template* gambar yang dapat digunakan sebagai panduan oleh pengguna.

1. Atribut:

- a. *id_template*: Identitas unik untuk setiap *template*.
- b. *nama_template*: Nama *template* gambar.
- c. *deskripsi*: Deskripsi singkat mengenai *template*.
- d. *file_template*: *File template* berupa gambar atau sketsa.

2. Relasi:

- a. *Template* gambar memiliki relasi 1:M dengan entitas Hasil Gambar, menunjukkan bahwa satu *template* dapat digunakan untuk menghasilkan banyak hasil gambar.
- b. *Template* ini dikelola oleh *admin* melalui relasi 1:M dengan entitas *Admin*.

c. Entitas Hasil Gambar

Entitas Hasil Gambar menyimpan informasi terkait hasil gambar yang diunggah oleh *User*. Entitas ini juga mencatat status analisis dan umpan balik yang diberikan oleh sistem berbasis CNN.

1. Atribut:

- a. *id_hasil*: Identitas unik untuk setiap hasil gambar.
- b. *file_hasil*: *File* hasil gambar yang diunggah oleh *User*.
- c. *status*: Status analisis, seperti "Sesuai" atau "Tidak Sesuai".
- d. *Feedback*: Umpan balik yang diberikan oleh sistem untuk membantu perbaikan gambar.

2. Relasi:

- a. Hasil gambar memiliki relasi M:1 dengan entitas *Template* Gambar, menunjukkan bahwa satu hasil gambar dibuat berdasarkan satu *template* tertentu.
- b. Hasil gambar juga memiliki relasi M:1 dengan entitas *User*, menunjukkan bahwa satu hasil gambar diunggah oleh satu *User*.

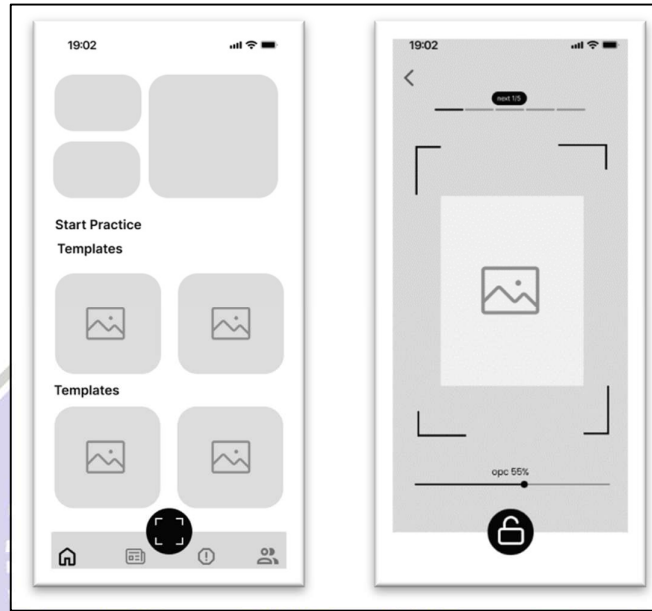
d. Relasi Antar Entitas

ERD ini menggambarkan hubungan logis antara entitas dalam sistem, dengan rincian sebagai berikut:

1. *Admin* Mengelola *Template* (1:M): Satu *admin* dapat mengelola banyak *template* gambar.
2. *User* Menggunakan *Template* (1:M): Satu *User* dapat memilih dan menggunakan banyak *template* untuk menggambar.
3. *Template* Digunakan untuk Hasil Gambar (1:M): Satu *template* dapat digunakan untuk menghasilkan banyak hasil gambar.
4. *User* Mengunggah Hasil Gambar (1:M): Satu *User* dapat mengunggah banyak hasil gambar ke sistem.

3.2.5 Perancangan Antarmuka

Perancangan antarmuka pada sistem ini dirancang untuk memberikan pengalaman pengguna yang nyaman dan terarah dalam menjalankan aktivitas menggambar dengan bantuan teknologi kecerdasan buatan (AI).



Gambar 3.7 Antarmuka Sistem

Interface utama terdiri dari beberapa tahap yang dirancang secara berurutan, seperti pada gambar 3.7, dimulai dari halaman *Home* yang menampilkan pilihan untuk memulai praktik dengan berbagai *template* gambar yang tersedia. Setelah pengguna memilih *template*, sistem menampilkan area praktik yang memandu pengguna dalam menggambar dengan memperlihatkan *template* secara transparan di tengah layar. Fitur indikator langkah-langkah di bagian atas membantu pengguna memantau tahap penyelesaian aktivitas yang sedang berlangsung.

Selama proses menggambar, sistem memberikan umpan balik langsung (*real-time Feedback*) berdasarkan hasil evaluasi kesesuaian gambar pengguna terhadap *template*. Setelah pengguna menekan tombol "Check Drawing", sistem akan menghitung nilai Mean Squared Error (MSE) antara gambar pengguna dan *template*. Berdasarkan nilai MSE tersebut, sistem menampilkan umpan balik berupa pesan teks otomatis seperti, "Hebat! Gambar Anda sangat mirip contoh." jika sangat

mirip, “Proporsi sudah baik, bisa sedikit dirapikan.” jika cukup mirip , dan “Perhatikan bentuk dan garis agar lebih sesuai contoh.” jika sama sekali tidak mirip.



Gambar 3. 8 Antarmuka Sistem (2)

Setelah proses menggambar selesai, sistem melakukan analisis kesesuaian gambar dengan membandingkannya dengan *template* yang dipilih, kemudian menampilkan persentase kemiripan dalam bentuk *similarity score* seperti pada gambar 3.8. Fitur ini memastikan pengguna dapat menilai kualitas hasil gambar mereka dengan lebih objektif.

Untuk mendukung kemudahan navigasi, desain *Interface* menggunakan ikon yang jelas seperti ikon kunci untuk mengunci gambar saat praktik berlangsung, serta ikon centang dan simpan yang berfungsi sebagai konfirmasi hasil akhir dan penyimpanan data. Perancangan ini mengutamakan kesederhanaan dan fungsionalitas, sehingga cocok digunakan dalam aplikasi edukasi berbasis latihan menggambar dengan bantuan AI.

3.2.6 Perancangan Database.

Database merupakan komponen penting dalam aplikasi Imaji yang digunakan untuk menyimpan data pengguna, *admin*, *template* gambar, serta hasil

gambar yang dihasilkan pengguna. Pada bagian ini, dijelaskan perancangan tabel-tabel yang digunakan, struktur kolomnya, serta relasi antar tabel.

a. Tabel *user*

Tabel *user* digunakan untuk menyimpan data pengguna aplikasi. Data ini mencakup informasi pribadi dan kredensial login. Struktur tabel adalah sebagai berikut:

Tabel 3.1 Struktur tabel *user*

Kolom	Tipe Data	Keterangan
<i>id_user</i>	INT	<i>Primary Key</i>
<i>nama</i>	<i>VARCHAR</i> (20)	
<i>email</i>	<i>VARCHAR</i> (50)	
<i>password</i>	<i>VARCHAR</i> (20)	

b. Tabel *admin*

Tabel *admin* digunakan untuk menyimpan data *admin* aplikasi. *Admin* memiliki akses untuk mengelola *template* gambar dan memantau aktivitas pengguna. Struktur tabel adalah sebagai berikut:

Tabel 3.2 Struktur tabel *admin*

Kolom	Tipe Data	Keterangan
<i>id_admin</i>	INT	<i>Primary Key</i>
<i>nama</i>	<i>VARCHAR</i> (20)	
<i>email</i>	<i>VARCHAR</i> (50)	
<i>password</i>	<i>VARCHAR</i> (20)	

c. Tabel *template_gambar*

Tabel *template_gambar* digunakan untuk menyimpan data *template* gambar yang tersedia di aplikasi. *Template* ini digunakan sebagai panduan oleh pengguna untuk menggambar. Struktur tabel adalah sebagai berikut:

Tabel 3.3 Struktur tabel *template_gambar*

Kolom	Tipe Data	Keterangan
<i>id_template</i>	INT	<i>Primary Key</i>

<i>nama_template</i>	<i>VARCHAR (20)</i>	
<i>deskripsi</i>	<i>TEXT</i>	
<i>file_template</i>	<i>VARCHAR (255)</i>	<i>Path/ File Template</i>

d. Tabel *hasil_gambar*

Tabel *hasil_gambar* digunakan untuk menyimpan data hasil gambar yang diunggah oleh pengguna setelah menggambar menggunakan panduan *template*. Tabel ini juga menyimpan status kesesuaian hasil dengan *template* serta *Feedback* dari sistem. Struktur tabel adalah sebagai berikut:

Tabel 3.4 Struktur tabel *hasil_gambar*

Kolom	Tipe Data	Keterangan
<i>id_hasil</i>	INT	<i>Primary Key</i>
<i>id_user</i>	INT	<i>Foreign Key(user)</i>
<i>id_template</i>	INT	<i>Foreign Key(template_gambar)</i>
<i>file_hasil</i>	<i>VARCHAR (255)</i>	<i>Path/ File Template</i>
<i>status</i>	<i>ENUM</i>	<i>“Correct”/ “Incorrect”</i>
<i>Feedback</i>	<i>TEXT</i>	

e. Relasi antar tabel

Relasi antar tabel dalam database ini dirancang sebagai berikut:

1. Relasi antara Tabel *user* dan *hasil_Gambar*

id_user pada tabel *hasil_gambar* menjadi *Foreign Key* yang mengacu pada *Primary Key* *id_user* di tabel *user*. Relasi ini memastikan setiap hasil gambar dikaitkan dengan pengguna yang mengunggahnya.

2. Relasi antara Tabel *template_gambar* dan *hasil_gambar*

id_template pada tabel *hasil_gambar* menjadi *Foreign Key* yang mengacu pada *Primary Key* *id_template* di tabel *template_gambar*. Relasi ini memastikan setiap hasil gambar dikaitkan dengan *template* yang digunakan. Relasi ini membantu dalam pengorganisasian data dan memudahkan pengambilan informasi secara terstruktur.

3.2.7 Pembagian Dataset

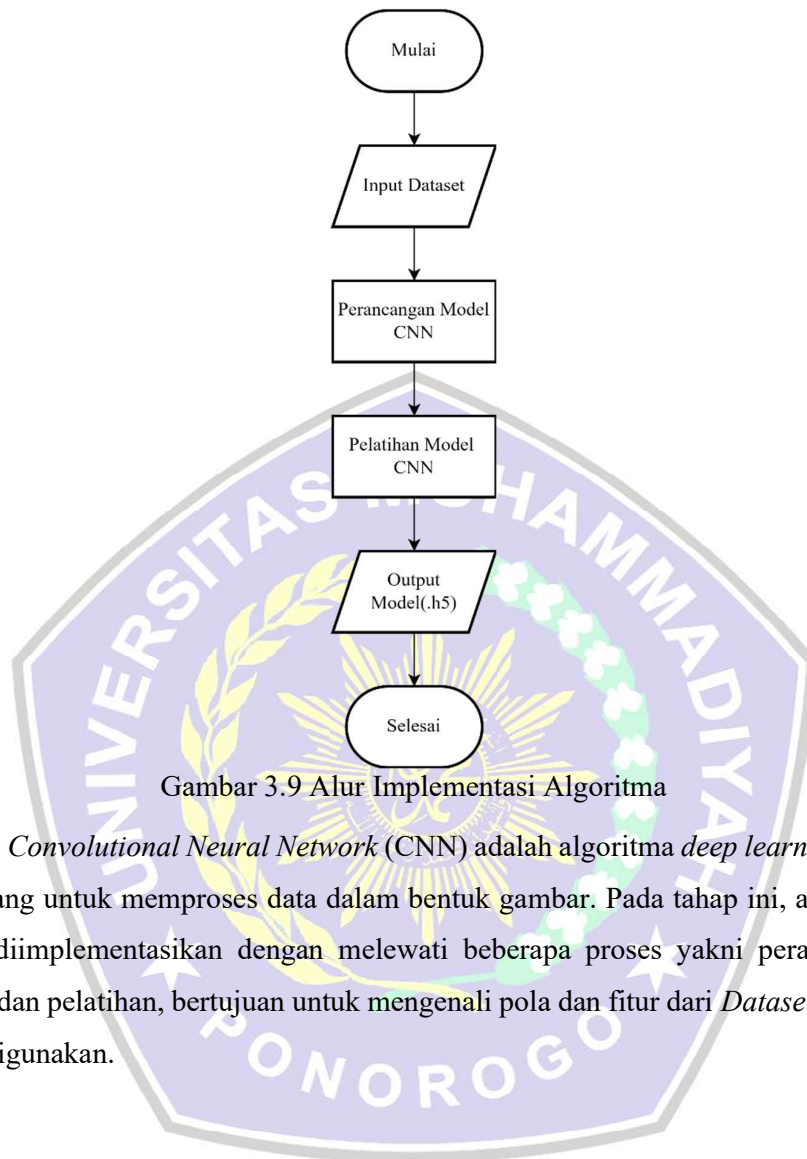
Proses pembagian data pada *Dataset* dilakukan setelah melalui tahap *preprocessing* gambar sketsa. Dalam penelitian ini, terdapat dua fase utama, yaitu fase *training* (pelatihan) dan fase *testing* (pengujian). Fase *training* menggunakan data latih (*train*) dan data validasi (*validation*). Data latih digunakan untuk melatih *model* CNN dalam mengenali pola gambar sketsa dari tiga kategori objek, yaitu kucing, ikan, dan anjing, yang masing-masing terdiri dari empat tahapan gambar (step1 hingga step4). Data validasi digunakan untuk mengevaluasi performa *model* selama proses pelatihan, guna memastikan *model* tidak mengalami *overfitting* dan mampu melakukan generalisasi dengan baik.

Sementara itu, data uji (test) digunakan dalam fase pengujian untuk menilai performa akhir *model* setelah proses pelatihan selesai. Pembagian *Dataset* dilakukan secara manual dengan rasio komposisi data sebesar 70% data latih, 15% data validasi, dan 15% data uji, dengan memperhatikan distribusi yang seimbang untuk setiap kelas dan setiap tahapan gambar sketsa. Berikut tabel Pembagian *Dataset* penelitian ini:

Tabel 3.5 Pembagian Dataset

Kelas	Jumlah Data	Data Latih	Data Validasi	Data Uji
Anjing_step1	800	500	150	150
Anjing_step2	800	500	150	150
Anjing_step3	800	500	150	150
Anjing_step4	800	500	150	150
Ikan_step1	800	500	150	150
Ikan_step2	800	500	150	150
Ikan_step3	800	500	150	150
Ikan_step4	800	500	150	150
Kucing_step1	800	500	150	150
Kucing_step2	800	500	150	150
Kucing_step3	800	500	150	150
Kucing_step4	800	500	150	150
Total	9600	6000	1800	1800

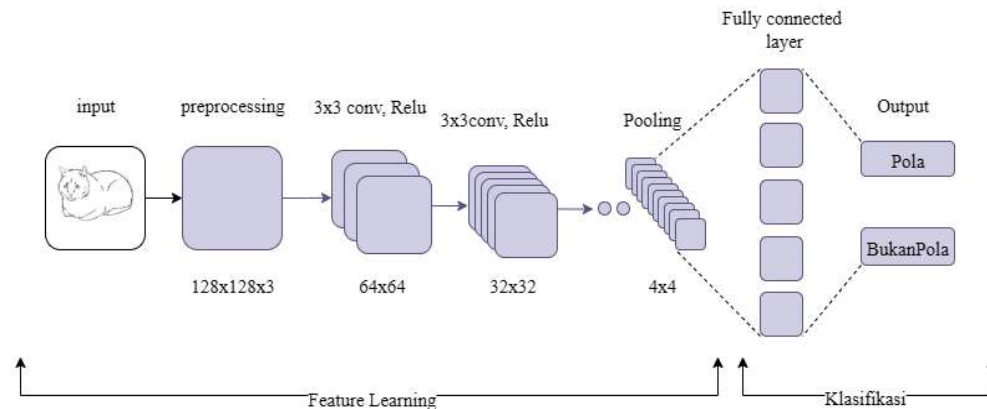
3.2.8 Implementasi Algoritma



Gambar 3.9 Alur Implementasi Algoritma

Convolutional Neural Network (CNN) adalah algoritma *deep learning* yang dirancang untuk memproses data dalam bentuk gambar. Pada tahap ini, algoritma CNN diimplementasikan dengan melewati beberapa proses yakni perancangan *model* dan pelatihan, bertujuan untuk mengenali pola dan fitur dari *Dataset* gambar yang digunakan.

a. Perancangan *Model* CNN



Gambar 3.10 Perancangan Arsitektur *Model* CNN *MobileNetV2*

Pada Gambar 3.10 menunjukkan rancangan arsitektur *model* yang digunakan dalam penelitian ini, yaitu *MobileNetV2*. *Model* ini dipilih karena memiliki keunggulan dalam hal kecepatan dan efisiensi, sehingga cocok untuk diterapkan pada perangkat mobile seperti smartphone Android.

MobileNetV2 merupakan arsitektur jaringan saraf konvolusional yang dikembangkan khusus untuk penggunaan pada perangkat dengan sumber daya terbatas. *Model* ini menggunakan struktur *depthwise separable convolution* dan *inverted residuals* yang membuatnya lebih ringan dibandingkan CNN konvensional, namun tetap mempertahankan performa yang kompetitif dalam pengenalan citra.

Pada tahap awal, citra sketsa yang telah diproses akan dimasukkan sebagai input ke dalam *model*. *MobileNetV2* akan melakukan ekstraksi fitur visual secara bertahap melalui blok-blok konvolusi yang saling terhubung. Setiap blok terdiri dari kombinasi operasi konvolusi, fungsi aktivasi ReLU, dan normalisasi batch untuk menjaga stabilitas pelatihan.

Citra akan diproses hingga menghasilkan vektor fitur berdimensi 1280, yang kemudian akan diteruskan ke tahap klasifikasi. Pada tahap klasifikasi, vektor fitur tersebut akan diproses oleh lapisan *fully connected* dengan fungsi aktivasi softmax untuk menghasilkan output klasifikasi multi-kelas. Dalam penelitian ini,

model dirancang untuk mengenali 12 kelas sketsa, yaitu gabungan dari 3 kategori objek (ikan, kucing, anjing) dan 4 tahapan menggambar.

b. Pelatihan *Model* CNN

Proses pelatihan *model MobileNetV2* dilakukan dengan pendekatan transfer learning, yaitu memanfaatkan bobot awal dari *model* yang telah dilatih pada *Dataset* besar (seperti *ImageNet*), kemudian disesuaikan (*fine-tuned*) dengan *Dataset* sketsa buatan sendiri.

Dataset pelatihan dibagi menjadi beberapa *batch* untuk efisiensi memori, dan pelatihan dilakukan selama beberapa *epoch*. Proses ini bertujuan untuk menyesuaikan *model* dengan karakteristik citra sketsa, sehingga dapat mengenali pola-pola garis dan bentuk yang khas dari gambar tangan.

Setelah pelatihan selesai, *model* diekspor ke dalam format .h5, lalu dikonversi ke format .tflite agar dapat diintegrasikan ke dalam aplikasi *Android*. Format ini memungkinkan *model* dijalankan secara efisien di perangkat *mobile* tanpa mengganggu performa sistem.

Dengan menggunakan *MobileNetV2*, sistem dapat melakukan proses klasifikasi gambar sketsa secara cepat dan akurat, sekaligus tetap ringan untuk dijalankan pada perangkat dengan spesifikasi menengah ke bawah.

3.2.9 Pengintegrasian *Model* CNN

Pengintegrasian *model* CNN dalam penelitian Imaji bertujuan untuk menghubungkan hasil pelatihan *model* dengan aplikasi yang digunakan oleh pengguna. *Model* CNN yang telah terlatih untuk mendeteksi dan mengenali pola sketsa akan diintegrasikan ke dalam aplikasi berbasis *Android* yang dikembangkan. Proses ini melibatkan penerapan *model* yang dilatih menggunakan *TensorFlow Lite*, yang memungkinkan aplikasi untuk menjalankan *model* CNN secara efisien di perangkat *mobile*. Dengan pengintegrasian ini, pengguna dapat mendapatkan umpan balik secara *real-time* saat membuat sketsa, dimana sistem akan mendeteksi kesalahan atau ketidaksesuaian dengan pola yang diinginkan dan memberikan koreksi atau saran untuk perbaikan.

3.2.10 Penerapan CNN untuk AI *Feedback*

Sistem AI *Feedback* dalam aplikasi IMAJI memanfaatkan *model* CNN *MobileNetV2* yang telah dikonversi ke dalam format *TensorFlow Lite* (.tflite) untuk memungkinkan inferensi langsung di perangkat *Android*. Pemilihan *MobileNetV2* didasarkan pada efisiensi komputasi dan ukuran *model* yang ringan, sehingga cocok untuk aplikasi *real-time* di perangkat *mobile* dengan keterbatasan sumber daya. *Model* ini dilatih untuk mengenali pola-pola visual pada sketsa berdasarkan kategori objek dan tahap gambar (*step*).

Secara teknis, ketika pengguna selesai menggambar, sistem melakukan akuisisi gambar melalui tangkapan layar (*screenshot*) dari hasil sketsa pengguna. Gambar tersebut kemudian diubah ukurannya menjadi 256x256 piksel dan dikonversi ke *ByteBuffer* dengan format RGB normalisasi $[0,1][0,1][0,1]$ untuk memenuhi spesifikasi input *model* *MobileNetV2*. Hasil inferensi berupa vektor probabilitas $y=[y_1, y_2, \dots, y_n]$ dengan n adalah jumlah kelas (misalnya 12 label dari 3 objek x 4 step). Label dengan nilai probabilitas maksimum $y'=\max(y_i)$ dianggap sebagai hasil prediksi *model*.

Namun, karena prediksi label saja belum cukup untuk menilai kualitas visual sketsa, sistem juga menghitung nilai *Mean Squared Error*(MSE) antara gambar pengguna (I) dan gambar *template* (T) dengan persamaan, menggunakan rumus 2.1 :

$$MSE = \frac{1}{n \cdot 3} \sum_{i=1}^n [(R_i - R'_i) + (G_i - G'_i) + (B_i - B'_i)^2] \div 225$$

Nilai MSE tersebut dikategorikan dalam tiga tingkatan evaluasi: sangat mirip ($MSE < 0.02$), cukup baik ($0.02 \leq MSE < 0.05$), dan perlu perbaikan ($MSE \geq 0.05$). Kategori ini digunakan sistem untuk menyusun umpan balik yang kontekstual. Dengan demikian, MSE berfungsi sebagai dasar logika pengambilan keputusan dalam sistem evaluasi otomatis.

Secara keseluruhan, integrasi CNN dalam sistem ini tidak hanya digunakan untuk klasifikasi, tetapi juga sebagai basis pengambilan keputusan dalam memberikan koreksi gambar berbasis AI. Analisis prediksi probabilistik dan

pengukuran jarak piksel (MSE) dikombinasikan untuk membentuk sistem *Feedback* yang adaptif, obyektif, dan efisien dalam membantu pengguna meningkatkan keterampilan menggambar secara bertahap.

3.3 Evaluasi Sistem

Evaluasi sistem dilakukan untuk menilai sejauh mana sistem yang dikembangkan, yakni aplikasi IMAJI, dapat mencapai tujuan fungsional dan performa yang telah ditetapkan. Evaluasi ini menjadi acuan untuk memastikan bahwa sistem mampu berjalan secara optimal, baik dari segi teknis maupun pengalaman pengguna akhir. Terdapat tiga aspek utama yang dievaluasi dalam penelitian ini, yaitu evaluasi *model* CNN, evaluasi sistem umpan balik berbasis AI, dan evaluasi aplikasi secara keseluruhan.

a. Evaluasi *model* CNN

Evaluasi terhadap *model Convolutional Neural Network* (CNN) bertujuan untuk mengukur performa *model* dalam mengklasifikasikan gambar sketsa berdasarkan objek dan tahapannya. *Model* yang digunakan adalah *MobileNetV2*, yang telah dilatih menggunakan *Dataset* sketsa ilustrasi. Evaluasi dilakukan menggunakan data uji (*test set*) yang belum pernah dilihat *model* sebelumnya. Metrik evaluasi yang digunakan mencakup:

1. Akurasi, Mengukur rasio prediksi benar terhadap seluruh data uji.
2. *Confusion Matrix*, Menunjukkan distribusi hasil prediksi *model* terhadap kelas-kelas yang tersedia.
3. *Classification Report*, Menyajikan nilai *precision*, *recall*, dan *F1-score* untuk setiap kelas, serta nilai rata-rata makro dan tertimbang.

Tujuan dari evaluasi ini adalah untuk menilai kemampuan generalisasi *model* serta mengidentifikasi potensi bias terhadap kelas tertentu.

b. Evaluasi AI *Feedback*

Evaluasi sistem umpan balik otomatis dilakukan untuk menilai efektivitas fungsi *showFeedbackBottomSheet()*, yang memberikan saran atau koreksi

berdasarkan nilai *Mean Squared Error*(MSE) antara gambar pengguna dan *template*. Nilai MSE digunakan sebagai indikator tingkat kemiripan visual.

Evaluasi dilakukan dengan menguji beberapa sketsa pengguna dan mencocokkannya dengan sketsa acuan (*template*). Rentang nilai MSE digunakan untuk menentukan kategori *Feedback* yang akan diberikan oleh sistem. Aspek yang dievaluasi meliputi:

1. Akurasi pemberian *Feedback* berdasarkan nilai MSE.
2. Kesesuaian pesan terhadap kualitas gambar.
3. Keterbacaan dan relevansi pesan umpan balik bagi pengguna.

Evaluasi ini bertujuan untuk memastikan bahwa sistem mampu memberikan koreksi yang tepat, mudah dipahami, dan mendukung proses belajar pengguna.

c. Evaluasi Aplikasi

Evaluasi aplikasi mencakup penilaian terhadap fungsi-fungsi utama yang terdapat dalam aplikasi IMAJI. Penilaian dilakukan melalui pengujian fitur-fitur sebagai berikut:

1. Fungsi klasifikasi gambar dan pemrosesan *model* AI.
2. Stabilitas dan kecepatan sistem, termasuk dalam kondisi normal maupun saat menggunakan fitur *Augmented Reality* (AR).
3. Responsivitas UI, terutama dalam menampilkan hasil klasifikasi dan *Feedback*.
4. Kesesuaian fitur dengan tujuan pembelajaran menggambar, seperti panduan overlay, mode AR/manual, dan penyimpanan hasil gambar.

3.4 Pengujian Sistem

Penelitian ini menggunakan metode *White Box Testing* untuk menguji fungsionalitas sistem berdasarkan analisis struktur internal, alur logika, dan jalur kode yang digunakan dalam pengembangan aplikasi IMAJI. Metode ini dipilih karena sistem memiliki komponen kompleks seperti pemrosesan *model* CNN, pengolahan gambar, serta interaksi logika antara modul *Augmented Reality* (AR), input pengguna, dan evaluasi berbasis AI *Feedback*.

Pengujian dilakukan pada fungsi-fungsi utama di dalam aplikasi, seperti pemanggilan *model TensorFlow Lite*, pengolahan gambar bitmap dari kamera, pemberian umpan balik otomatis, dan penyimpanan hasil latihan ke database Supabase. Setiap fungsi diuji dengan berbagai parameter, seperti resolusi gambar input, tingkat kontras sketsa, stabilitas pencahayaan sekitar, serta ketepatan deteksi pola berdasarkan langkah menggambar yang aktif.

Salah satu aspek penting dalam pengujian adalah mengukur akurasi sistem dalam mendeteksi dan memberikan umpan balik terhadap sketsa yang digambar pengguna. Akurasi sistem dapat dihitung dengan menggunakan rumus (2.2) berikut:

$$Akurasi\ Sistem = \frac{Hasil\ Benar}{Total\ Pengujian} \times 100\%$$

Dengan menggunakan rumus ini, kita dapat mengevaluasi sejauh mana sistem mampu memberikan umpan balik yang tepat dan akurat pada setiap skenario pengujian. Berikut adalah tabel skenario pengujian untuk fitur utama aplikasi pembelajaran sketsa :

Tabel 3.6 Skenario Pengujian Sistem

No.	Fitur yang Diuji	Fungsi/Kode yang Diuji	Skenario Pengujian	Input	Hal yang Diharapkan
1	Panduan Gambar AR (Overlay)	<code>placeImageOnPlane()</code>	Menampilkan overlay AR jika anchor valid	Posisi anchor ARCore dari kamera	Gambar <i>template</i> muncul stabil di atas kertas dan tidak berpindah
2	Inferensi <i>Model CNN</i>	<code>runInference()</code>	Mengeksekusi <i>model</i> dan mengambil label serta confidence score	Gambar hasil sketsa (bitmap)	Sistem mengembalikan label prediksi dan nilai probabilitas secara benar

No.	Fitur yang Diuji	Fungsi/Kode yang Diuji	Skenario Pengujian	Input	Hal yang Diharapkan
3	Evaluasi Kemiripan (MSE)	<code>calculateMSE(img1, img2)</code>	Menghitung nilai kesalahan gambar	Dua bitmap (hasil gambar dan <i>template</i>)	Nilai MSE dikembalikan dengan akurasi tinggi (rentang 0–1)
4	Saran Koreksi Sketsa Otomatis	<code>showFeedbackBottomSheet()</code>	Memberikan saran berdasarkan nilai MSE	Nilai MSE dari hasil evaluasi	Pesan umpan balik ditampilkan sesuai rentang nilai MSE
5	Mode Manual Overlay	<code>switchToManualOverlay()</code>	Menampilkan overlay tanpa AR dan dapat dipindahkan secara manual	Overlay ditempatkan di layar	Gambar <i>template</i> muncul, dan bisa diatur posisi manual
6	Simpan dan Upload Hasil	<code>finishDrawing()</code> + upload API Retrofit	Menyimpan gambar hasil dan metadata ke Supabase	Bitmap gambar, <code>userId</code> , skor MSE	Data berhasil diunggah dan tersimpan di database serta storage

No.	Fitur yang Diuji	Fungsi/Kode yang Diuji	Skenario Pengujian	Input	Hal yang Diharapkan
7	Upload Template(A dmin)	uploadTemplateTo Supabase ()	Mengunggah h template baru untuk pengguna	File gambar template	Template baru tersimpan ke storage, dan metadata berhasil dicatat di database Supabase

Tabel 3.6 menunjukkan skenario pengujian untuk fitur utama aplikasi pembelajaran sketsa. Pengujian dilakukan secara sistematis untuk memastikan setiap fitur berjalan sesuai spesifikasi, mulai dari tahap input data hingga hasil akhir yang diharapkan.

