

BAB 3

METODE PENELITIAN

3.1 ALAT YANG DIGUNAKAN

Dalam penelitian ini digunakan alat dan bahan sebagai penunjang untuk merancang sistem klasifikasi bahasa isyarat menggunakan CNN pada python untuk pembelajaran.

3.1.1 *Software*

Perangkat lunak yang digunakan pada perancangan sistem klasifikasi bahasa isyarat menggunakan CNN pada python untuk pembelajaran adalah:

Tabel 3.1 Detail Software

Jenis	Komponen
Sistem operasi	Windows 10 64 bit
Layanan <i>Cloud Storage</i>	Google drive
Program aplikasi	<ul style="list-style-type: none">▪ Jupyter notebook▪ Google colab
Bahasa pemrograman	Python

3.1.2 *Hardware*

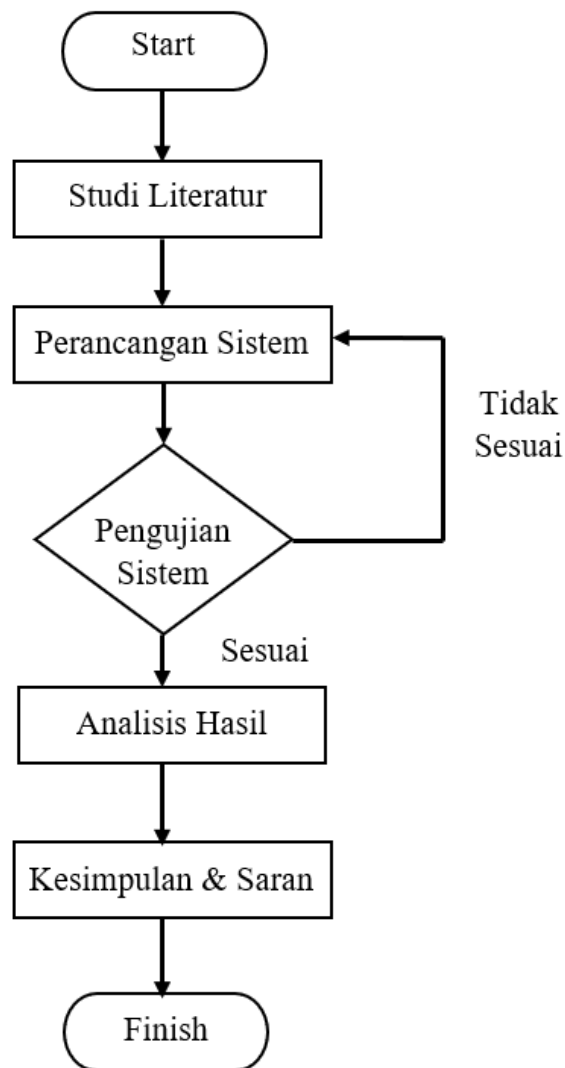
Perangkat keras yang digunakan pada perancangan sistem klasifikasi bahasa isyarat menggunakan CNN pada python untuk pembelajaran yaitu *webcam* eyesec dan laptop dengan tipe *acer spin sp111-33-c3a9*. Berikut spesifikasi laptop yang disusun dalam bentuk tabel:

Tabel 3.2 Detail Hardware

<i>Processor</i>	Intel Celeron N4020
HDD	466 GB
RAM	3.8 GB

3.2 ALUR PENELITIAN

Pada tahap ini, penjelasan mengenai langkah-langkah perencanaan penelitian diuraikan agar memudahkan penulis dalam memulai dan menyelesaikan penelitian serta mencapai hasil yang diinginkan. Adapun penulis menyusun tahapan penelitian sebagaimana pada gambar 3.1 berikut:



Gambar 3.1 Alur Tahapan Penelitian

Berdasarkan gambar tahapan penelitian terdapat 6 alur yang akan penulis lakukan, hal tersebut dijelaskan bahwa:

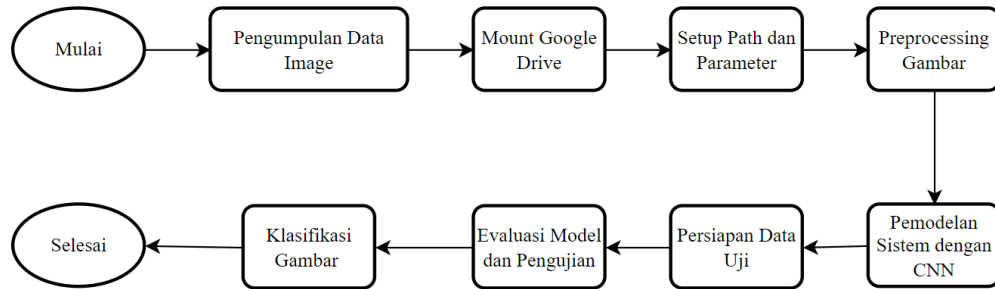
1. Langkah awal dimulai dengan melakukan studi literatur, yang merupakan proses di mana peneliti mencari informasi yang relevan dengan penelitian yang sedang dilakukan. Studi literatur ini melibatkan membaca buku, jurnal, situs web, serta referensi lainnya yang terkait dengan topik penelitian. Pada tahap

- ini, penulis melakukan pencarian terhadap penelitian sebelumnya yang relevan untuk digunakan sebagai pedoman dalam menjalankan penelitian ini. Studi literatur berperan penting dalam memperoleh referensi dan pengetahuan mengenai klasifikasi bahasa isyarat, CNN, bahasa pemrograman python, dan topik lain yang relevan dengan penelitian ini, serta materi pendukung lainnya.
2. Langkah kedua adalah perancangan sistem. Tahap ini bertujuan untuk membangun sistem dapat mengklasifikasi 10 bahasa isyarat yang telah ditentukan.
 3. Tahap selanjutnya adalah pengujian sistem. Tahap ini sistem yang telah dirancang dan telah melewati beberapa tahap, akan dilakukan beberapa pengujian terkait salah satunya *training* data citra.
 4. Setelahnya merupakan analisis hasil. Pada tahapan ini dilakukan analisa terhadap sistem, misalnya apakah sistem yang dibuat termasuk ke dalam sistem yang baik dan sesuai. Selain itu, tahapan ini memaparkan hasil perhitungan terkait *accuracy* dan lainnya.
 5. Kemudian, penulis mengambil kesimpulan dan saran dari penelitian yang telah dilakukan.
 6. Terakhir penulis melakukan *finishing* laporan sebagai bentuk tanggung jawab dari penelitian yang telah diselesaikan.

3.2.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan selama ± 7 bulan pada bulan Desember 2023 – Juni 2024. Dalam kurun waktu tersebut penulis akan melakukan serangkaian proses yaitu; literatur *review* dan penentuan topik, perancangan dan pembuatan sistem (bahasa pemrograman python) untuk mengklasifikasi kosakata bahasa isyarat yang telah ditentukan, analisis hasil data, hingga penyelesaian laporan. Hal tersebut akan penulis sesuaikan dengan rujukan batasan masalah penelitian sehingga tidak keluar dari topik dan tujuan penelitian. Dalam menyelesaikan penelitian ini bertempat di wilayah Banyumas - Purwokerto (tempat tinggal penulis) dan di Laboratorium Institut Teknologi Telkom Purwokerto.

3.2.2 Proses Perancangan



Gambar 3.2 Diagram Perancangan Sistem







Gambar 3.2 merupakan perancangan sistem yang dibuat, penjelasannya sebagai berikut:

a. Pengumpulan Data *Image*

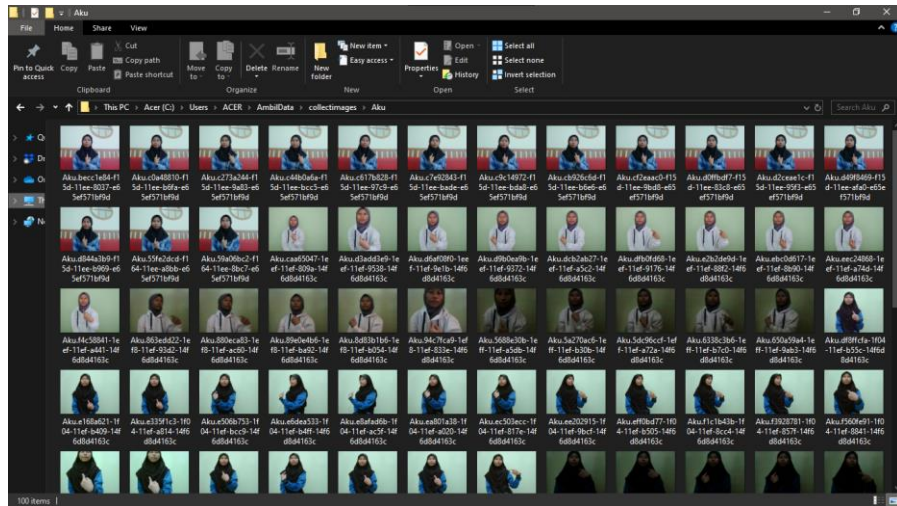
Pada tahap ini adalah mengumpulkan dataset dengan memanfaatkan *webcam* dan kamera PC berupa gerakan bahasa isyarat yang terkait penelitian. Dalam pengambilan data ini menggunakan jupyter notebook. Ada 10 kelas pada penelitian diantaranya: Aku, Bahagia, Halo, Kamu, Maaf, Sedih, Teman, Tentu, Terimakasih, dan Ya.

Tabel 3.3 Gesture Bahasa Isyarat

	
Aku	Bahagia
	
Halo	Kamu

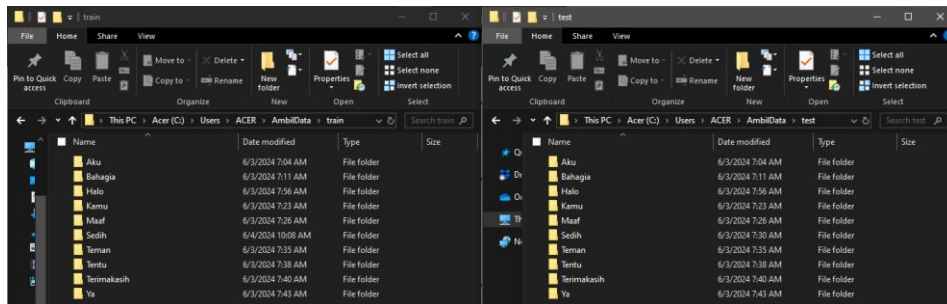
	
Maaf	Sedih
	
Teman	Tentu
	
Terimakasih	Ya

Masing-masing kelas, data *image* yang dikumpulkan adalah sejumlah 100 *image*. Jadi, untuk total data yang digunakan dalam penelitian ini adalah sejumlah 1000 (10 kelas x 100 *image* data per kelas) data *image*. Selain itu, data yang digunakan memiliki beberapa kondisi pencahayaan dan beberapa posisi dari *gesture* tangan bahasa isyarat. Untuk kondisi pencahayaan, ada yang dalam kondisi pencahayaan gelap dan kondisi pencahayaan terang. Hal ini dimaksudkan untuk menambah variasi data agar mengetahui bagaimana kinerja model yang dihasilkan nantinya.



Gambar 3.3 Contoh Collect Data Image ‘Aku’ sejumlah 100 image

Selanjutnya data yang diperoleh akan dibagi secara mandiri menjadi 2 file yaitu untuk *train* dan *test*. Untuk file *train* sejumlah 75 image per kelas, sedangkan *test* sejumlah 25 image per kelas.



Gambar 3.4 Pembagian Train dan Test

Setelah itu, proses selanjutnya yaitu pengunggahan ke *google drive*. Digunakannya *gdrive* adalah untuk memudahkan dalam mengakses data, memanfaatkan fitur *backup*, serta dapat melakukan integrasi langsung dengan *google colab*.

b. Mount Google Drive

Google Drive di-*mount* menggunakan *drive.mount* untuk mengakses dataset yang disimpan didalamnya saat awal membuat kode program. Hal ini berkaitan dengan efisiensi dalam menyimpan data, sehingga dapat dengan mudah pula untuk mengelola, memperbarui, dan menyimpan dataset dalam struktur folder terorganisir.

c. *Setup Path dan Parameter*

Digunakan untuk memuat dataset yang akan digunakan dalam proses pelatihan dan pengujian model. Dalam pengaturan ini dapat mempermudah dan memastikan bahwa data terstruktur dengan cara yang mendukung efisiensi. Selain itu, dalam langkah ini nantinya akan digunakan untuk menyimpan model yang telah selesai dilatih, sehingga nantinya model dapat digunakan kembali tanpa perlu melalui proses pelatihan ulang.

d. *Preprocessing Gambar*

Setelah data yang dibutuhkan telah dikumpulkan secara keseluruhan, tahapan selanjutnya adalah *preprocessing* gambar. Pada tahap ini melakukan normalisasi pada dataset gambar untuk memastikan nilai piksel berada dalam rentang 0-1. Proses ini dilakukan untuk mengefektifkan model dalam pembelajaran karena berkaitan dengan konsistensi nilai piksel, memungkinkan model mencapai performa yang optimal karena berkaitan dengan mempercepat konvergensi, dan memungkinkan dalam meningkatkan akurasi model karena berkaitan dengan normalisasi membantu model lebih mudah mengenali dan mempelajari fitur penting *image*. Selain itu juga, sebelum data dinormalisasi data *image* untuk *train* yang akan digunakan akan di-*resize* ke ukuran 64x64 piksel saat proses mempersiapkan dataset untuk *training*.

e. *Pemodelan Sistem dengan CNN*

Pemodelan sistem merupakan pembuatan model sistem untuk “Klasifikasi Bahasa Isyarat Menggunakan CNN Pada Python Untuk Pembelajaran”, dimana tentunya memanfaatkan *Convolutional Neural Network* (CNN) dalam pembuatannya. Model *Convolutional Neural Network* (CNN) dibangun menggunakan Keras Sequential API. Dalam tahap ini juga disebut dengan mendefinisikan arsitektur model CNN menggunakan lapisan-lapisan *convolutional*, *pooling*, dan *fully connected*. Berikut arsitektur yang digunakan pada penelitian ini: *input layer*, *convolutional layers*, *pooling layers*, *flatten layer*, *dense layers*, *dropout layer*, dan *output layer*. Aktivasi yang digunakan adalah ReLU, dan lapisan *dropout* diterapkan untuk mengurangi *overfitting*.

f. Persiapan Data Uji

Setelah tahapan pemodelan sistem telah dilakukan, proses selanjutnya adalah persiapan data uji. Pada proses ini menggunakan ‘ImageDataGenerator’ dari pustaka Keras, hal tersebut berkaitan untuk mengelola data gambar yang akan diuji. Pada tahap ini juga dilakukan *rescale* untuk normalisasi gambar pada data *testing*. Hal tersebut adalah untuk memastikan bahwa data uji memiliki skala yang sama dengan data latih sebelumnya yang juga dinormalisasi.

g. Evaluasi Model dan Pengujian

Pada kode program yang telah diberikan, tahapan evaluasi model dan pengujian dilakukan dengan menggunakan beberapa teknik dan alat untuk memastikan bahwa model yang telah dilatih dievaluasi dengan baik dan hasilnya dianalisis dengan tepat. Untuk evaluasi model dan pengujian memanfaatkan beberapa hal seperti Tensorflow, Keras, OpenCV, dan *scikit-learn* untuk mengevaluasi kinerja dari model. Diantaranya untuk memberikan informasi dari model yang dibuat adalah dengan menggunakan *confusion matrix* dan *classification report*. Keduanya tadi adalah untuk mengetahui performa sistem. Parameter yang nantinya dicari meliputi *accuracy*, *precision* dan *recall*. Parameter ini didapatkan dari *confusion matrix*.

Nantinya hasil *confusion matrix* yang diperoleh dari evaluasi model adalah tabel dengan ukuran 10 x 10, hal ini berkaitan dengan sejumlah kelas yang ada dalam penelitian, dan penelitian saat ini ada sejumlah 10 kelas. Untuk pemahaman TP, TN, FP, dan FN pada setiap kelas adalah sebagai berikut:

1. Kelas “Aku”
 - TP (Aku): n (benar memprediksi “Aku” sebagai “Aku”)
 - FP (Aku): n (kelas lain yang diprediksi sebagai “Aku”)
 - FN (Aku): n (“Aku” diprediksi sebagai kelas lain)
 - TN (Aku): n (Semua prediksi selain baris dan kolom “Aku”)
2. Kelas “Bahagia”
 - TP (Bahagia): n (benar memprediksi "Bahagia" sebagai " Bahagia")
 - FP (Bahagia): n (kelas lain yang diprediksi sebagai “Bahagia”)

- FN (Bahagia): n (“Bahagia” diprediksi sebagai kelas lain)
 - TN (Bahagia): n (Semua prediksi selain baris dan kolom “Bahagia”)
3. Kelas “Halo”
- TP (Halo): n (benar memprediksi “Halo” sebagai “Halo”)
 - FP (Halo): n (kelas lain yang diprediksi sebagai “Halo”)
 - FN (Halo): n (“Halo” diprediksi sebagai kelas lain)
 - TN (Halo): n (Semua prediksi selain baris dan kolom “Halo”)
4. Kelas “Kamu”
- TP (Kamu): n (benar memprediksi “Kamu” sebagai “Kamu”)
 - FP (Kamu): n (kelas lain yang diprediksi sebagai “Kamu”)
 - FN (Kamu): n (“Kamu” diprediksi sebagai kelas lain)
 - TN (Kamu): n (Semua prediksi selain baris dan kolom “Kamu”)
5. Kelas “Maaf”
- TP (Maaf): n (benar memprediksi “Maaf” sebagai “Maaf”)
 - FP (Maaf): n (kelas lain yang diprediksi sebagai “Maaf”)
 - FN (Maaf): n (“Maaf” diprediksi sebagai kelas lain)
 - TN (Maaf): n (Semua prediksi selain baris dan kolom “Maaf”)
6. Kelas “Sedih”
- TP (Sedih): n (benar memprediksi “Sedih” sebagai “Sedih”)
 - FP (Sedih): n (kelas lain yang diprediksi sebagai “Sedih”)
 - FN (Sedih): n (“Sedih” diprediksi sebagai kelas lain)
 - TN (Sedih): n (Semua prediksi selain baris dan kolom “Sedih”)
7. Kelas “Teman”
- TP (Teman): n (benar memprediksi “Teman” sebagai “Teman”)
 - FP (Teman): n (kelas lain yang diprediksi sebagai “Teman”)
 - FN (Teman): n (“Teman” diprediksi sebagai kelas lain)
 - TN (Teman): n (Semua prediksi selain baris dan kolom “Teman”)
8. Kelas “Tentu”
- TP (Tentu): n (benar memprediksi “Tentu” sebagai “Tentu”)
 - FP (Tentu): n (kelas lain yang diprediksi sebagai “Tentu”)
 - FN (Tentu): n (“Tentu” diprediksi sebagai kelas lain)
 - TN (Tentu): n (Semua prediksi selain baris dan kolom “Tentu”)

9. Kelas “Terimakasih”

- TP (Terimakasih): n (benar memprediksi “Terimakasih” sebagai “Terimakasih”)
- FP (Terimakasih): n (kelas lain yang diprediksi sebagai “Terimakasih”)
- FN (Terimakasih): n (“Terimakasih” diprediksi sebagai kelas lain)
- TN (Terimakasih): n (Semua prediksi selain baris dan kolom “Terimakasih”)

10. Kelas “Ya”

- TP (Ya): n (benar memprediksi “Ya” sebagai “Ya”)
- FP (Ya): n (kelas lain yang diprediksi sebagai “Ya”)
- FN (Ya): n (“Ya” diprediksi sebagai kelas lain)
- TN (Ya): n (Semua prediksi selain baris dan kolom “Ya”)

Sedangkan untuk pengertian *accuracy*, *precision* dan *recall* adalah sebagai berikut: *Accuracy* adalah pengujian atau matrik evaluasi yang mengukur sejauh mana hasil prediksi atau klasifikasi mendekati nilai aktual secara keseluruhan. Metrik ini mengukur rasio prediksi yang benar dibandingkan dengan semua prediksi yang dilakukan. Untuk klasifikasi multi kelas, persentase akurasi dapat dihitung menggunakan formula yang tersedia dalam persamaan berikut:

$$\begin{aligned} Accuracy &= \frac{\textit{The number of true predictions}}{\textit{The number of predictions}} \cdot 100\% \\ &= \frac{TP + TN}{TP + FP + FN + TN} \cdot 100\% \end{aligned} \quad (3.1)$$

Precision adalah kemampuan dari classifier untuk mengidentifikasi jumlah prediksi positif yang secara relatif benar. Presisi dihitung sebagai rasio dari *true positive* dibagi dengan jumlah *true positive* dan *false positive* untuk setiap kelas. Persentase presisi pada klasifikasi multi kelas dapat dihitung menggunakan rumus yang tersedia dalam persamaan berikut:

$$Precision = \frac{TP}{TP + FP} \quad (3.2)$$

Recall adalah kemampuan classifier untuk mendeteksi semua kasus positif dalam confusion matrix. Recall dihitung sebagai rasio antara true positive dan jumlah dari true positive dan false negative untuk setiap kelas. Persentase recall pada klasifikasi multi kelas dapat dihitung menggunakan rumus yang tersedia dalam persamaan berikut:

$$Recall = \frac{TP}{TP + FN} \quad (3.3)$$

h. Klasifikasi Gambar

Klasifikasi gambar ini merupakan tahap akhir dari kode program yang digunakan untuk mengklasifikasikan gambar individu dan menampilkan kelas yang diprediksi. Dengan memuat model yang telah disimpan dan melakukan prediksi pada gambar selain yang ada pada *train dan test*, hal ini adalah untuk menguji kinerja model pada data yang belum pernah dilihat sebelumnya. Klasifikasi gambar ini membantu memastikan apakah model berperforma dengan baik atau tidaknya pada data baru.