

BAB I

PENDAHULUAN

A. Latar Belakang

Kemampuan untuk berkomunikasi sangat penting bagi manusia untuk menjalani kehidupan sehari-hari. Namun, kelompok masyarakat tertentu, seperti penyandang tunarungu dan tunawicara, menghadapi kesulitan dalam berkomunikasi dengan masyarakat umum. Mereka menggunakan bahasa isyarat sebagai cara utama untuk menyampaikan perasaan, ide, dan informasi.

Survei Sosial Ekonomi Nasional 2020 menunjukkan bahwa sekitar 3,07 juta orang di Indonesia mengalami keterbatasan sensorik, menurut Sekretariat Jendral DPR RI [1]. Keterbatasan sensorik sudah termasuk tunawicara, dan tunarungu. Namun, masyarakat masih kurang memahami Bahasa Isyarat Indonesia (BISINDO) ataupun Sistem Isyarat Bahasa Indonesia (SIBI). Hal ini menghambat komunikasi antara penyandang tunarungu dan masyarakat umum, yang berdampak pada akses terbatas terhadap pendidikan, layanan publik, dan partisipasi sosial.

Kemajuan teknologi pada bidang *computer vision* dan pembelajaran mesin (*machine learning*) membuka peluang besar untuk menciptakan sistem penerjemah bahasa isyarat yang otomatis dan *real-time*. Sistem semacam ini dapat memfasilitasi komunikasi antara pengguna bahasa isyarat dan masyarakat umum. Salah satu pendekatan yang digunakan adalah dengan mengandalkan kamera RGB untuk menangkap gerakan tangan dan tubuh, yang kemudian diterjemahkan menjadi teks atau suara melalui model pembelajaran mendalam (*deep learning*).

Dalam penelitian ini, Bahasa Isyarat Amerika (*American Sign Language / ASL*) digunakan sebagai basis model karena dua alasan utama. Pertama, ketersediaan dataset ASL lebih lengkap dan terstruktur secara publik dibandingkan dengan dataset BISINDO yang masih sangat terbatas. Kedua, penelitian ini bersifat eksploratif dan bertujuan membangun fondasi

sistem penerjemah bahasa isyarat berbasis gerakan yang dapat dikembangkan lebih lanjut untuk bahasa isyarat lokal seperti BISINDO.

Dengan memanfaatkan ASL sebagai studi awal, sistem ini diharapkan dapat menjadi prototipe yang fleksibel, yang kemudian dapat disesuaikan dengan *gesture* dan struktur BISINDO apabila dataset lokal telah tersedia. Selain itu, sistem ini juga dapat digunakan sebagai alat bantu pembelajaran bahasa isyarat bagi masyarakat Indonesia yang ingin belajar ASL maupun BISINDO di masa mendatang.

Penelitian ini secara keseluruhan bertujuan untuk membangun sistem penerjemah *gesture* ASL dalam bentuk huruf dan kata, yang diintegrasikan ke dalam antarmuka video *real-time* menggunakan model Feedforward Neural Network (FNN) dan Bidirectional Long Short-Term Memory (BiLSTM). Kontribusinya diharapkan dapat mendukung teknologi inklusif dan mempercepat adopsi sistem komunikasi berbasis *gesture* di Indonesia.

B. Rumusan Masalah

Terdapat beberapa rumusan masalah yang dapat diambil dan dipertimbangkan berdasarkan latar belakang di atas.

1. Bagaimana mengembangkan sistem berbasis pengenalan pola menggunakan teknologi *hand-tracking* dan *machine learning* untuk menerjemahkan gerakan tangan (bahasa isyarat) menjadi huruf atau kata?
2. Bagaimana memastikan akurasi prediksi sistem terhadap berbagai gerakan tangan yang merepresentasikan huruf atau kata tertentu?
3. Bagaimana mengintegrasikan sistem penerjemahan bahasa isyarat ini dengan aplikasi lain, seperti mengetik otomatis pada perangkat komputer?

C. Batasan Masalah

Penelitian ini akan berkonsentrasi pada pengembangan sistem penerjemah bahasa isyarat. Untuk memastikan bahwa penelitian ini efektif dan mencapai hasil yang diharapkan, akan ada beberapa masalah yang akan dibatasi. Tujuan pembatasan ini adalah untuk menghindari masalah yang terlalu luas dan memberikan batasan yang jelas pada pengembangan sistem.

1. Dataset Tertentu

Data ASL (*American Sign Language*) yang hanya digunakan dalam penelitian ini terdiri dari *gesture* huruf A hingga Z serta beberapa kata seperti "*hello*", "*thank you*", "*please*" dan sebagainya. Bahasa isyarat tambahan seperti BISINDO, BSL, dan JSL tidak dipelajari. Ini dilakukan untuk mempersempit ruang lingkup dan memastikan data pelatihan konsisten.

2. Jumlah Kelas *Gesture* Terbatas

Sistem hanya dilatih dan diuji menggunakan 10 kata dari dataset WLASL dan 28 huruf ASL (A-Z, *space*, *delete*). Hal ini dilakukan untuk menjaga kompleksitas awal dan fokus pada evaluasi sistem dasar. Namun, jumlah ini masih jauh dari cakupan bahasa isyarat yang sesungguhnya, yang dapat mencakup ratusan kata dan frasa.

3. Jenis *Input Visual*

Ini bertujuan untuk membuat sistem dapat diakses secara luas tanpa memerlukan perangkat keras tambahan. Oleh karena itu, sistem hanya menerima *input* dari kamera RGB standar, seperti *webcam* laptop atau USB *webcam*. Kamera khusus seperti Kinect, RealSense, atau *multi-view* tidak digunakan.

4. Kondisi Pengujian Masih Terbatas

Pengujian dilakukan secara manual (*self-testing*) dalam kondisi ideal, seperti pencahayaan terang dan latar belakang bersih. Beberapa eksperimen dalam kondisi redup dan *background* gelap sudah dilakukan, namun belum dilakukan uji coba sistematis dengan variasi lingkungan yang kompleks.

5. Platform Implementasi

Implementasi sistem dilakukan pada *platform* komputer personal (PC) berbasis Windows dengan bahasa Python. Sistem belum dioptimalkan untuk *platform mobile* (Android/iOS), *embedded* (Raspberry Pi, Jetson), atau perangkat VR/AR.

6. Belum Dilakukan Uji dengan Pengguna Sesungguhnya (User Study)

Sistem belum diuji dengan pengguna tunarungu atau interpreter bahasa isyarat secara langsung. Hal ini menjadi keterbatasan utama dalam mengukur aspek kepraktisan, *usability*, dan kepuasan pengguna akhir terhadap sistem.

D. Tujuan Penelitian

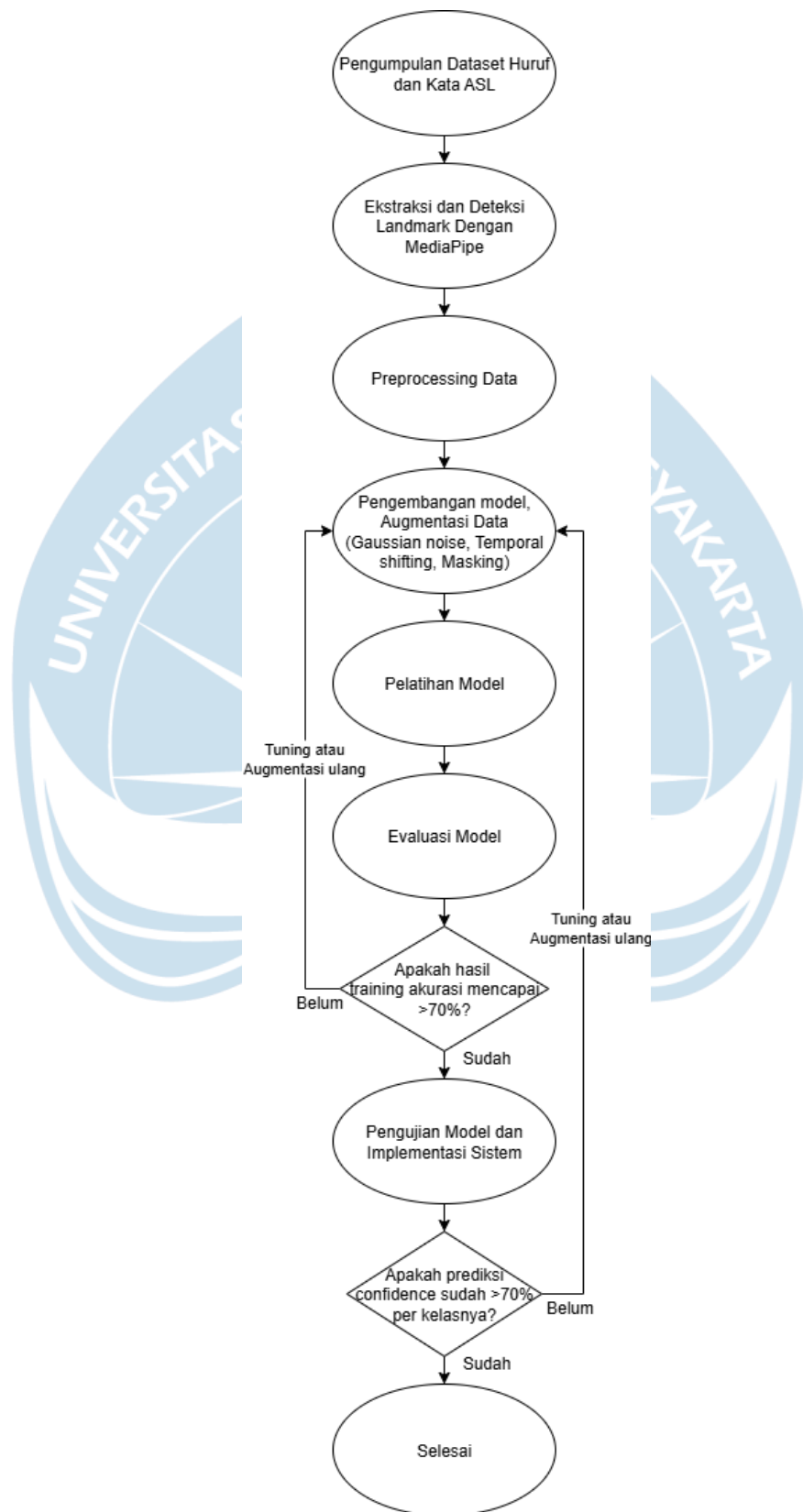
Tujuan dari penelitian ini adalah untuk membuat sistem otomatis yang dapat secara instan menerjemahkan teks dari *American Sign Language* (ASL). Teknologi *deep learning* digunakan dalam sistem ini, terutama arsitektur Feedforward Neural Network (FNN) untuk klasifikasi huruf dan arsitektur Bidirectional Long Short-Term Memory (BiLSTM) untuk klasifikasi kata yang didasarkan pada urutan gerakan tangan. MediaPipe, sebuah struktur pengawasan tangan dan estimasi pose yang efektif, mengumpulkan data *input* melalui koordinat *landmark*.

Secara lebih rinci, tujuan dari penelitian ini adalah:

1. Membangun model klasifikasi *gesture* berbasis FNN yang dapat mengidentifikasi alfabet bahasa isyarat menggunakan koordinat *landmark* tangan yang diperoleh dari gambar yang diekstraksi.
2. Mengembangkan model BiLSTM berbasis data sekuensial untuk mengenali *gesture* kata (*word-level*) dalam urutan pose dan gerak tangan selama beberapa *frame* video.
3. Mengimplementasikan sistem *real-time* menggunakan *webcam* standar, yang dapat mendeteksi, mengenali, dan menerjemahkan gerakan tangan ke dalam teks secara langsung menggunakan antarmuka berbasis Python.
4. Mengintegrasikan teknik *smoothing* dan *filtering* (seperti Exponential Moving Average dan *majority voting*) untuk meningkatkan kestabilan hasil prediksi sistem dalam kondisi nyata.

E. Metode Penelitian

Untuk membangun sistem penerjemah *American Sign Language* (ASL) pada *level* huruf dan kata, penelitian ini menggunakan pendekatan eksperimen terapan yang menggabungkan teknologi *computer vision*, *deep learning*, dan Teknik pose dan *hand tracking*. Sistem berbasis *webcam* secara *real-time* dengan akurasi tinggi dan responsivitas optimal adalah fokus utama. Gambar 1.1 merupakan tahapan metodologi penelitian yang dibagi menjadi beberapa fase.



Gambar 1. 1 Flowchart Metode Penelitian

1. Persiapan Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari dua jenis, sesuai dengan dua target keluaran sistem yaitu:

- a. Huruf ASL (Alfabet): Dataset berupa gambar tangan dalam format JPG diunduh dari Kaggle, mencakup huruf A–Z dan beberapa token tambahan seperti “*space*” dan “*delete*”.
- b. Kata ASL (*Word-Level*): Menggunakan video dataset WLASL (*Word-Level American Sign Language*) yang merupakan benchmark terbuka untuk pengenalan *gesture* tingkat kata dalam ASL.

Pemilihan dua dataset ini didasarkan pada kebutuhan sistem untuk menerima pengenalan *gesture* statis (huruf) maupun *gesture* dinamis (kata/frasa).

2. Ekstraksi *Landmark*

Untuk menjadikan data visual lebih terstruktur dan efisien dalam pengolahan *machine learning*, seluruh dataset dikonversi ke dalam bentuk koordinat *landmark* dengan bantuan *library* MediaPipe dari Google:

- a. Pada data huruf, digunakan MediaPipe Hands untuk mengekstraksi 21 titik *landmark* tangan (x, y, z).
- b. Pada data kata, digunakan MediaPipe Holistic yang mengekstrak 300 fitur dari pose tubuh dan tangan (33 pose + 21 tangan kiri + 21 tangan kanan).

Proses ekstraksi dilakukan menggunakan bahasa Python dan hasilnya disimpan dalam format JSON dan CSV agar kompatibel untuk *preprocessing* dan augmentasi.

3. *Preprocessing* dan *Augmentasi*

Setelah diekstraksi, data *landmark* melalui beberapa tahapan *Preprocessing*:

- a. **Normalisasi** koordinat ke rentang $[0, 1]$ untuk menghilangkan ketergantungan terhadap ukuran layar dan posisi kamera.
- b. **One-Hot Encoding** label untuk keperluan klasifikasi multikelas.
- c. **Oversampling** menggunakan *RandomOverSampler* diterapkan pada dataset WLASL untuk mengatasi ketidakseimbangan jumlah sampel antar kelas.
- d. **Augmentasi** data dilakukan pada pelatihan untuk meningkatkan kemampuan generalisasi model dan mengurangi risiko *overfitting* akibat keterbatasan jumlah data *gesture*. Augmentasi ini diterapkan secara probabilistik (50%) pada setiap *batch* data pelatihan, dengan tiga teknik utama:

1) **Gaussian Noise**

Ditambahkan *noise* acak dengan distribusi normal sebesar 0.01 terhadap semua nilai koordinat *landmark*. Tujuannya adalah mensimulasikan variasi kecil yang mungkin muncul akibat perbedaan tangan atau posisi kamera.

2) **Temporal Shifting**

Data *sequence* digeser secara acak hingga 30 *frame* menggunakan fungsi `torch.roll`, untuk menyesuaikan *timing gesture* yang bervariasi antar pengguna. Ini meniru kenyataan bahwa pengguna tidak selalu memulai *gesture* dari *frame* ke-0.

3) **Masking**

Diterapkan masking acak terhadap sekitar 2% fitur dalam *frame*, dengan mengatur nilai koordinat menjadi nol. Teknik ini membuat model lebih tahan terhadap *noise* atau kondisi di mana *landmark* gagal terdeteksi oleh MediaPipe.

4. Pelatihan Model

Pelatihan dilakukan dengan dua model *neural network* yang berbeda, disesuaikan dengan jenis data:

a. **Model FNN (Feedforward Neural Network)** untuk pengenalan huruf:

- 1) *Input*: 63 fitur ($21 \text{ titik} \times 3 \text{ koordinat}$).
- 2) *Arsitektur*: 3 *hidden layer*, aktivasi ReLU, Dropout untuk regularisasi.
- 3) *Output layer*: Softmax.
- 4) *Optimizer*: Adam, 30 *epoch*, *learning rate* awal 0.0005.
- 5) *Callback*: ReduceLROnPlateau dan EarlyStopping.
- 6) *Framework*: Tensorflow + Keras.

b. **Model BiLSTM + Attention** untuk pengenalan kata:

- 1) *Input*: *Sequence* 30 *frame* $\times$ 300 fitur.
- 2) *Arsitektur*: 2-layer BiLSTM, dengan *Attention layer* untuk memberi bobot fokus pada *frame* penting.
- 3) *Optimizer*: Adam, 50 *epoch*, *batch size* 16.
- 4) *Scheduler*: CosineAnnealingLR.
- 5) *Framework*: PyTorch.

Pemilihan FNN untuk huruf didasarkan pada *gesture* statis yang tidak memerlukan konteks temporal, sementara BiLSTM digunakan untuk kata karena mampu menangani urutan data dan variasi panjang gerakan.

5. Implementasi Sistem *Real-time*

Model yang telah dilatih diintegrasikan ke dalam aplikasi Python berbasis desktop yang bersifat *real-time* dan interaktif:

- a. **Inisialisasi Webcam:** menangkap video *frame*.
- b. **Tracking:** MediaPipe digunakan untuk mendeteksi dan mengekstrak *landmark* secara langsung.
- c. **Preprocessing Real-time:**
 - 1) Data *landmark* dinormalisasi.
 - 2) *Smoothing* dilakukan menggunakan Exponential Moving Average (EMA) untuk mengurangi *jitter* prediksi.
- d. **Inferensi:**
 - 1) Jika *confidence* > 60%, hasil klasifikasi ditampilkan.
 - 2) Untuk model huruf, prediksi dikirim ke sistem *input* otomatis menggunakan PyAutoGUI: huruf → ketik, “*space*” → spasi, “*delete*” → *backspace*.
- e. **Visualisasi:** Teks hasil prediksi dan *confidence* ditampilkan langsung pada video *stream*.

6. Pengujian dan Evaluasi

Sistem diuji secara langsung dengan mengamati performa dan stabilitas sistem:

- a. **Evaluasi Akurasi Deteksi *Landmark*:** kesesuaian deteksi MediaPipe dalam berbagai posisi dan pencahayaan.
- b. **Evaluasi Akurasi Model:**
 - 1) Menggunakan Confusion Matrix, Precision, Recall, dan F1-Score.
 - 2) Dilakukan pada data uji dan skenario *real-time*.

c. Evaluasi Responsivitas dan Stabilitas:

- 1) Kemampuan sistem merespon *gesture* dengan cepat.
- 2) Konsistensi hasil klasifikasi dengan *buffer voting* dan *smoothing*.

d. Kondisi Uji: berbagai pencahayaan, latar belakang, posisi tangan, dan jarak kamera.

F. Sistematika Penulisan

Laporan tugas akhir ini disusun secara sistematis dalam enam bab, dengan rincian sebagai berikut:

1. BAB I – PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metode penelitian, dan sistematika penulisan yang digunakan dalam laporan.

2. BAB II – TINJAUAN PUSTAKA

Bab ini memuat teori-teori yang relevan sebagai dasar dari penelitian, termasuk teori tentang *American Sign language* (ASL), *deep learning*, MediaPipe, model FNN dan BiLSTM, serta teknologi pendukung lainnya.

3. BAB III – LANDASAN TEORI

Bab ini menjelaskan konsep-konsep dan teknologi yang digunakan secara teknis, seperti algoritma *deep learning*, metode ekstraksi fitur dengan MediaPipe, dan pemrosesan data *gesture* secara *real-time*.

4. BAB IV – PERANCANGAN DAN PENGEMBANGAN SISTEM

Bab ini membahas proses pengembangan sistem secara detail, mulai dari deskripsi masalah, persiapan dataset, desain model *neural network*, hingga pengujian dan evaluasi performa model klasifikasi huruf dan kata ASL.

5. BAB V – IMPLEMENTASI MODEL DAN PENGUJIAN SISTEM

Bab ini menjelaskan bagaimana model yang telah dilatih diimplementasikan dalam aplikasi lokal berbasis Python, serta proses pengujian sistem dalam skenario penggunaan secara *real-time* menggunakan *webcam*.

6. BAB VI – PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan serta saran-saran yang dapat menjadi acuan untuk pengembangan lebih lanjut pada penelitian berikutnya.

