

BAB 2

LANDASAN TEORI

2.1 Bahasa Isyarat Indonesia (BISINDO)

Bahasa isyarat merupakan medium komunikasi utama yang digunakan oleh komunitas tunarungu dalam kehidupan sehari-hari. Di Indonesia, terdapat dua bentuk bahasa isyarat yang dikenal luas, yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). Di antara kedua sistem tersebut, BISINDO lebih banyak digunakan oleh komunitas tunarungu karena dianggap lebih mudah dipahami [10]. BISINDO memanfaatkan kombinasi gerakan tangan, lengan, bahasa tubuh, serta ekspresi wajah sebagai unsur utama dalam penyampaian makna. Seluruh gerakan tersebut berkembang secara alami sehingga lebih sesuai dengan interpretasi dan kebiasaan komunikasi komunitas tunarungu [10]. Dalam penelitian ini, kosakata yang dijadikan fokus pada *website* pembelajaran adalah alfabet BISINDO.

2.2 *Gesture Recognition*

Gesture recognition merupakan salah satu bidang eksplorasi penting dalam *computer vision* yang memiliki beragam aplikasi dalam sistem *Human-Computer Interaction* (HCI) [11]. Secara umum, gestur dapat diklasifikasikan menjadi dua jenis, yaitu gestur statis (*static gesture*) yang merepresentasikan postur tangan pada satu *frame* tertentu, dan gestur dinamis (*dynamic gesture*) yang melibatkan gerakan tangan berkelanjutan lintas *frame* [11].

Pendekatan untuk memperoleh data gestur tangan terbagi menjadi dua, yaitu *sensor-based* yang memerlukan perangkat tambahan yang dipasangkan pada tangan pengguna, dan *vision-based* yang hanya memanfaatkan kamera untuk menangkap gambar atau video gerakan tangan [11]. Pendekatan *vision based* lebih praktis karena tidak memerlukan perangkat khusus dan dapat dijalankan secara *real-time*. *Pipeline* umum sistem *gesture recognition vision based* mencakup tahapan akuisisi data, praproses, ekstraksi fitur, serta klasifikasi [11]. Aplikasi *gesture recognition* mencakup berbagai domain, di antaranya pengenalan bahasa isyarat [11].

2.3 Computer Vision

Computer vision merupakan bidang penelitian dalam kecerdasan buatan yang berupaya memberikan kemampuan kepada komputer untuk memperoleh pemahaman dari data visual. Informasi yang berasal dari gambar atau video diproses menggunakan berbagai teknik komputasi sehingga sistem dapat mengenali dan menginterpretasikan objek maupun pola yang terdapat di dalamnya. Pengembangan teknologi ini melibatkan perpaduan antara ilmu komputer, *machine learning*, dan pengolahan gambar guna meniru proses pengamatan visual yang dilakukan manusia [12]. Beberapa tugas utama yang termasuk dalam ruang lingkup *computer vision* adalah klasifikasi gambar dan deteksi objek [12].

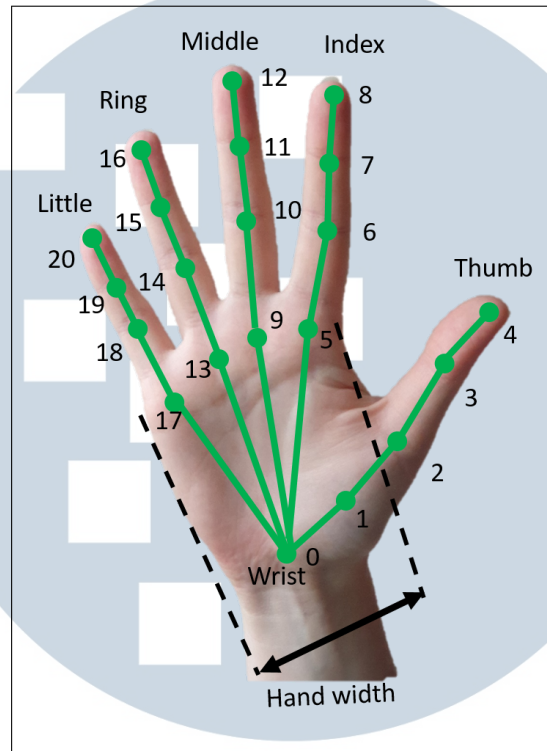
Pada umumnya, alur kerja sistem *computer vision* terdiri atas beberapa tahapan, mulai dari akuisisi data visual, praproses, ekstraksi fitur, inferensi menggunakan model, hingga pascaproses untuk menghasilkan *output* akhir [12]. Perkembangan *computer vision* mengalami peningkatan yang signifikan seiring kemajuan *deep learning*, khususnya melalui pemanfaatan *Convolutional Neural Network* (CNN) yang memiliki kemampuan untuk mempelajari sekaligus mengekstraksi fitur visual secara otomatis dari data berlabel dalam jumlah besar [12].

2.4 MediaPipe

MediaPipe merupakan *framework* yang dikembangkan oleh Google untuk memfasilitasi pembangunan sistem yang dapat memproses multimedia lintas platform dengan dukungan eksekusi secara *real-time* [13]. Salah satu komponen yang banyak dimanfaatkan dalam *framework* ini adalah *MediaPipe Hands*, yaitu modul pelacakan tangan berbasis *deep learning* yang dirancang untuk mendeteksi sekaligus melacak pergerakan tangan secara langsung melalui masukan kamera tanpa memerlukan perangkat keras tambahan [13].

MediaPipe Hands beroperasi melalui dua tahapan utama. Tahap pertama adalah deteksi telapak tangan (*palm detection*) dengan memanfaatkan detektor berbasis *single shot* untuk menghasilkan kotak pembatas (*bounding box*) yang menunjukkan posisi tangan pada setiap *frame*. Tahap berikutnya adalah estimasi *landmark* tangan, yaitu proses ketika model regresi memprediksi 21 titik *landmark* tangan secara langsung dari area tangan yang telah dipotong (*crop*) berdasarkan hasil deteksi sebelumnya [13]. Dua puluh satu titik *landmark* tersebut

merepresentasikan seluruh sendi jari serta beberapa titik pada telapak tangan, seperti yang dapat dilihat pada Gambar 2.1.



Gambar 2.1. Ilustrasi 21 titik *landmark* tangan

Setiap titik *landmark* direpresentasikan oleh tiga komponen koordinat, yaitu (x, y, z) . Nilai x dan y menunjukkan posisi titik pada *frame* citra yang telah dinormalisasi ke rentang 0 sampai 1, sedangkan nilai z menggambarkan kedalaman relatif titik terhadap posisi pergelangan tangan [13]. Dalam penelitian ini, *MediaPipe Hands* digunakan untuk mendeteksi kedua tangan secara bersamaan sehingga menghasilkan total $21 \times 2 = 42$ titik *landmark*, di mana setiap titik memiliki tiga nilai koordinat. Seluruh data koordinat tersebut kemudian disusun ke dalam matriks berukuran 42×3 dan dimanfaatkan sebagai masukan bagi model 1D CNN pada proses klasifikasi alfabet BISINDO.

2.5 1D Convolutional Neural Network (1D CNN)

1D *Convolutional Neural Network* (CNN) merupakan salah satu arsitektur dalam *deep learning* yang telah banyak menunjukkan kinerja yang baik pada berbagai bidang aplikasi. Pada data berdimensi satu, 1D CNN umum dimanfaatkan untuk tugas klasifikasi maupun pengenalan pola pada sinyal sekuensial [8],

termasuk dalam pengenalan aktivitas manusia berbasis data sensor [14]. Dalam penelitian ini, pendekatan tersebut diterapkan pada data koordinat *landmark* tangan untuk mengenali alfabet BISINDO. Setiap sampel direpresentasikan dalam bentuk matriks berukuran 42×3 , yang terdiri atas 42 titik *landmark* dari kedua tangan, dengan masing-masing titik memiliki tiga nilai koordinat (x,y,z) . Adapun komponen utama arsitektur yang digunakan dijelaskan pada bagian berikut.

2.5.1 Lapisan Konvolusi 1D

Lapisan konvolusi 1D mengekstraksi fitur lokal dari urutan masukan dengan cara menggeser *kernel* sepanjang dimensi sekuens [8]. Proses *Forward propagation* pada lapisan Conv1D ke- l dirumuskan sebagai berikut:

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} \text{Conv1D}(w_{ik}^{l-1}, s_i^{l-1}) \quad (2.1)$$

x_k^l adalah masukan neuron ke- k pada lapisan l , b_k^l adalah bias, s_i^{l-1} adalah keluaran neuron ke- i pada lapisan sebelumnya, dan w_{ik}^{l-1} adalah bobot *kernel* dari neuron ke- i ke neuron ke- k .

2.5.2 Fungsi Aktivasi ReLU

Hasil keluaran dari setiap operasi konvolusi selanjutnya diproses menggunakan fungsi aktivasi *Rectified Linear Unit* (ReLU) [8]. Persamaan *Rectified Linear Unit* (ReLU) didefinisikan sebagai berikut:

$$\text{ReLU}(x) = \max(0, x) \quad (2.2)$$

Nilai masukan yang bernilai negatif akan dipetakan menjadi nol, sedangkan nilai positif diteruskan tanpa perubahan. Karakteristik tersebut membuat ReLU memiliki kompleksitas komputasi yang rendah sekaligus membantu menjaga aliran gradien selama proses pembelajaran jaringan, sehingga pelatihan model dapat berlangsung lebih efektif.

2.5.3 Batch Normalization

Untuk meningkatkan kestabilan proses pembelajaran, setiap keluaran lapisan konvolusi dinormalisasi menggunakan teknik *Batch Normalization* (BN) [15]. Metode ini bekerja dengan menyesuaikan distribusi aktivasi berdasarkan statistik yang diperoleh dari setiap *mini-batch*. Tahap pertama dilakukan dengan menghitung *Batch Mean* dan selanjutnya *Batch Variance*, dengan persamaan sebagai berikut:

$$\mu_b = \frac{1}{m} \sum_{i=1}^m x_i \quad (2.3)$$

$$\sigma_b^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_b)^2 \quad (2.4)$$

Pada persamaan tersebut, μ_b menyatakan nilai rata-rata aktivasi dalam satu *mini-batch*, sedangkan σ_b^2 merupakan nilai variansi aktivasi pada *mini-batch* yang sama. Setelah nilai rata-rata dan variansi diperoleh, dilakukan proses *Normalization* terhadap setiap aktivasi menggunakan persamaan berikut:

$$\hat{x}_i = \frac{x_i - \mu_b}{\sqrt{\sigma_b^2 + \epsilon}} \quad (2.5)$$

Pada persamaan tersebut, $\hat{x}_i$ merupakan hasil normalisasi dari aktivasi x_i , sedangkan ϵ merupakan konstanta bernilai kecil yang digunakan untuk menghindari permasalahan numerik saat proses pembagian. Selanjutnya, proses *Batch Normalization Output* dilakukan melalui penskalaan dan pergeseran hasil normalisasi menggunakan parameter γ dan β , yang dirumuskan pada persamaan berikut:

$$y_i = \gamma \hat{x}_i + \beta \quad (2.6)$$

Pada persamaan tersebut, y_i merupakan keluaran akhir proses *Batch Normalization*. Sementara itu, parameter γ dan β dipelajari oleh model untuk mengatur skala dan pergeseran distribusi aktivasi setelah proses normalisasi dilakukan.

2.5.4 Lapisan *Dropout*

Dropout adalah teknik regularisasi yang bekerja dengan menonaktifkan sejumlah neuron secara acak pada setiap iterasi pelatihan guna mengurangi risiko *overfitting* [8]. Operasi *Dropout* dirumuskan menggunakan persamaan berikut:

$$\tilde{h}_i = h_i \cdot r_i, \quad r_i \sim \text{Bernoulli}(1 - p) \quad (2.7)$$

$\tilde{h}_i$ merupakan keluaran neuron setelah proses *dropout*, h_i adalah keluaran neuron sebelum *dropout*, sedangkan r_i merupakan variabel Bernoulli yang menentukan apakah neuron dipertahankan ($r_i = 1$) atau dinonaktifkan ($r_i = 0$). Nilai p menyatakan probabilitas neuron dinonaktifkan selama proses pelatihan.

2.5.5 *Global Average Pooling 1D*

Berbeda dengan *pooling* konvensional, lapisan *Global Average Pooling 1D* (GAP1D) mereduksi dimensi representasi dengan menghitung rata-rata seluruh posisi sekuens untuk setiap kanal fitur secara global [14]. Operasi *Global Average Pooling 1D* dirumuskan menggunakan persamaan berikut:

$$g_j = \frac{1}{n} \sum_{i=1}^n C_j(i) \quad (2.8)$$

di mana n adalah panjang sekuens, $C_j(i)$ adalah nilai fitur pada posisi ke- i dari kanal ke- j , dan g_j adalah representasi global kanal tersebut. Operasi ini mengubah tensor berukuran $(N, 42, 256)$ menjadi vektor berukuran $(N, 256)$, mengurangi jumlah parameter secara signifikan sekaligus memberikan representasi ringkas dari seluruh urutan *landmark*.

2.5.6 Lapisan *Dense* dan *Softmax*

Representasi fitur yang dihasilkan oleh lapisan GAP1D kemudian diteruskan ke dua lapisan *Dense* yang masing-masing terdiri atas 256 dan 128 neuron dengan fungsi aktivasi ReLU. Untuk mengurangi risiko *overfitting*, diterapkan mekanisme *Dropout* sebagai teknik regularisasi selama proses pelatihan [14]. Selanjutnya, *Dense Layer* menghasilkan nilai logit yang kemudian diproses menggunakan

fungsi *Softmax* untuk memperoleh probabilitas setiap kelas. Persamaan *Softmax* dirumuskan sebagai berikut [8].

$$P_i = \frac{e^{z_i}}{\sum_{k=1}^C e^{z_k}} \quad (2.9)$$

Pada persamaan tersebut, P_i menyatakan probabilitas sampel termasuk ke dalam kelas ke- i , z_i merupakan nilai *logit* untuk kelas ke- i , sedangkan C menunjukkan jumlah keseluruhan kelas alfabet BISINDO. Nilai probabilitas yang dihasilkan berada pada rentang $0 \leq P_i \leq 1$ dengan total probabilitas seluruh kelas memenuhi $\sum_{i=1}^C P_i = 1$. Hasil klasifikasi ditentukan berdasarkan kelas yang memiliki nilai probabilitas terbesar.

2.6 End-User Computing Satisfaction (EUCS)

End-User Computing Satisfaction (EUCS) merupakan model evaluasi yang digunakan untuk mengukur tingkat kepuasan pengguna terhadap sistem berbasis komputer [16]. Pendekatan ini berorientasi pada persepsi pengguna akhir sebagai dasar penilaian, terutama terkait pengalaman pengguna dalam memanfaatkan sistem untuk memperoleh informasi maupun menyelesaikan tugas tertentu [16]. Penilaian dalam model EUCS dilakukan melalui lima dimensi utama, yaitu:

1. *Content*: Dimensi ini mengevaluasi kesesuaian, kelengkapan, dan kebermanfaatan informasi yang disediakan sistem bagi pengguna.
2. *Accuracy*: Dimensi ini berfokus pada tingkat ketepatan, keandalan, serta validitas informasi yang dihasilkan oleh sistem.
3. *Format*: Aspek ini mengevaluasi kualitas penyajian informasi, termasuk tata letak, struktur tampilan, dan desain antarmuka yang mendukung kemudahan pemahaman pengguna.
4. *Ease of Use*: Dimensi ini digunakan untuk mengukur kemudahan penggunaan sistem, baik dalam proses navigasi maupun interaksi pengguna terhadap fitur yang tersedia.
5. *Timeliness*: Aspek ini berkaitan dengan kemampuan sistem dalam menyediakan informasi atau memberikan respons terhadap permintaan pengguna secara tepat waktu.

Tingkat kepuasan pengguna pada setiap dimensi diukur menggunakan skala *Likert* dengan rentang nilai 1 hingga 5. Selanjutnya, nilai *End-User Computing Satisfaction* (EUCS) dihitung menggunakan persamaan *EUCS* berikut [16]:

$$EUCS = \frac{\sum_{i=1}^n S_i}{n} \quad (2.10)$$

Pada persamaan tersebut, *EUCS* menyatakan nilai rata-rata kepuasan pengguna, S_i merupakan skor yang diperoleh pada dimensi ke- i , sedangkan n menunjukkan jumlah dimensi yang digunakan dalam model EUCS, yaitu sebanyak lima dimensi. Semakin mendekati nilai 5, semakin tinggi tingkat kepuasan pengguna terhadap sistem. Sebaliknya, nilai yang semakin mendekati 1 menunjukkan bahwa sistem masih memerlukan perbaikan untuk memenuhi kebutuhan pengguna [16].

2.7 Penelitian Terdahulu

Pada penelitian ini, kajian terhadap penelitian terdahulu difokuskan pada pengembangan model pengenalan bahasa isyarat, khususnya BISINDO, serta pengembangan sistem pembelajaran bahasa isyarat. Ringkasan penelitian terdahulu yang menjadi acuan dalam penelitian ini disajikan pada Tabel 2.1.

Tabel 2.1. Ringkasan Penelitian Terdahulu

Judul Penelitian	Peneliti & Tahun	Metode/Model	Dataset & Hasil Utama
<i>Indonesian Sign Language Image Detection Using Convolutional Neural Network (CNN) Method</i>	[6] Sihananto et al. (2023)	CNN dengan arsitektur 2 lapisan konvolusi, <i>pooling</i> , dan <i>fully connected layer</i>	Dataset: SIBI dan BISINDO (abjad A–Z). Hasil: akurasi SIBI 93,29% (epoch 50), akurasi BISINDO 82,32% (epoch 40)
Lanjut pada halaman berikutnya			

Judul Penelitian	Peneliti & Tahun	Metode/Model	Dataset & Hasil Utama
<i>A Machine Learning-Driven Web Application for Sign Language Learning</i>	[7] Orovwode et al. (2024)	CNN berbasis <i>Keras</i> dengan 3 lapisan konvolusi, di- <i>deploy</i> ke web menggunakan Flask	Dataset: 44.654 gambar ASL (24 kelas). Hasil: akurasi model 94,68%, rata-rata latensi prediksi 64,02 ms/step
<i>E-Course Information System Development for Web-Based Bisindo Learning</i>	[5] Reihan et al. (2022)	Sistem <i>e-course</i> berbasis <i>website</i> menggunakan PHP (Laravel) dan ReactJS dengan metode <i>Waterfall</i>	Sistem pembelajaran BISINDO daring. Hasil: 100% valid (<i>black-box</i>), skor SUS 84,8 (kategori baik)
Pengembangan <i>Website Speech To Video</i> Bahasa Isyarat Indonesia (BISINDO) Berbasis Algoritma <i>Long Short Term Memory</i>	[17] Deleviar et al. (2025)	LSTM untuk konversi suara ke video BISINDO, pengembangan sistem menggunakan metode RUP	Dataset: 800 sampel audio (dimensi $124 \times 129 \times 1$). Hasil: akurasi model LSTM 95%, respons pengguna positif

Tabel 2.1 menunjukkan bahwa penelitian bahasa isyarat telah dilakukan dengan berbagai pendekatan. Penelitian oleh [6] menggunakan CNN dengan 2 lapisan konvolusi, *pooling*, dan *fully connected layer*, menghasilkan akurasi 93,29% pada dataset SIBI dan 82,32% pada dataset BISINDO. Penelitian oleh [7] menggunakan CNN berbasis *Keras* dengan 3 lapisan konvolusi pada 44.654 sampel ASL (24 kelas), mencapai akurasi 94,68% dan di *deploy* menggunakan Flask. Penelitian oleh [5] berfokus pada sistem *e-course* BISINDO berbasis PHP (Laravel) dan ReactJS dengan metode *Waterfall*, memperoleh skor SUS 84,8. Penelitian oleh [17] menerapkan LSTM untuk mengonversi suara ke video BISINDO dengan metode RUP, menghasilkan akurasi 95% pada 800 sampel audio.