JIKA (Jurnal Informatika) Universitas Muhammadiyah Tangerang *P ISSN : 2549-0710*
Tangerang, April 2025, Vol 9, No.2 pp 209-218 *E ISSN : 2722-2713*

PENDEKATAN MLP DALAM KLASIFIKASI BAHASA ISYARAT: ANALISIS JARAK *EUCLIDEAN LANDMARK* TANGAN

Kadek Feny Sugiantari[1)], Putu Hendra Suputra[2)], Luh Joni Erawati Dewi[3)], Gede Bakti Pratama Putra[4)]

[1,2,3,4] S1 Ilmu Komputer, Universitas Pendidikan Ganesha, Jalan. Udayana No. 11, Singaraja
Co Responden Email: feny@undiksha.ac.id

Article history
Received 16 Jan 2025
Revised 28 Mar 2025
Accepted 13 Apr 2025
Available online 30 May 2025

Keywords
Classification,
MLP,
Hand Landmark,
Sign Language,
Euclidean Distance

Abstract

Advances in Computer Vision technology in the field of Artificial Intelligence has promotes new inclusive breakthroughs to support communication for individuals with disabilities, such us the hearing and speech impaired. This study develops a classification model for SIBI sign language, specifically for number 0-9, using the Euclidean distance between hand landmarks as features. The research process includes data collection, hand landmark extraction using Mediapipe, Euclidean distance feature extraction, model training with a Multi Layer Perceptron, evaluation, and real-time implementation. The results show that the model successfully classifies the poses for numbers 0-9 and non-poses, with an accuracy of 87.17%. A threshold was applied during the evaluation and real-time implementation stages to ensure that all input data is classified with a high level of confidence. The results of this research can support the learning process of SIBI sign language for individuals with disabilities.

Riwayat
Diterima 16 Jan 2025
Revisi 28 Mar 2025
Disetujui 13 Apr 2025
Terbit online 30 Mei 2025

Kata Kunci
Klasifikasi,
MLP,
Landmark Tangan,
Bahasa Isyarat,
Jarak *Euclidean*

Abstrak

Kemajuan teknologi *Computer Vision* dalam bidang Kecerdasan Buatan mendorong terobosan baru yang inklusif untuk mendukung komunikasi bagi penyandang disabilitas, seperti tunarungu dan tunawicara. Kajian ini mengembangkan model klasifikasi bahasa isyarat angka SIBI, khususnya angka 0-9 dengan menggunakan jarak *Euclidean* antar *landmark* tangan sebagai fitur. Tahapan penelitian mencakup pengumpulan data, ekstraksi *landmark* tangan dengan Mediapipe, ekstraksi fitur jarak *Euclidean*, pelatihan model dengan *Multi Layer Perceptron*, evaluasi, dan implementasi *real-time.* Hasil kajian menunjukkan model berhasil mengklasifikasikan *pose* angka 0-9 dan *non-pose*, dengan akurasi 87.17% dan penerapan *threshold* pada tahap evaluasi serta implementasi *real-time* untuk memastikan semua *input* data terklasifikasi dengan tingkat kepercayaan tinggi. Hasil penelitian ini dapat menunjang proses pembelajaran bahasa isyarat SIBI bagi penyandang disabilitas.

PENDAHULUAN

Kemajuan pesat dalam *Computer Vision* dan Kecerdasan Buatan (AI) telah mendorong inovasi teknologi komputer yang inklusif di berbagai bidang, terutama dalam menjembatani komunikasi alami bagi penyandang disabilitas, khususnya tunarungu dan tunawicara. Interaksi alami seperti bahasa isyarat visual menjadi bentuk komunikasi utama bagi mereka dalam merepresentasikan huruf dan angka. Indonesia memiliki dua jenis bahasa isyarat, yaitu Bahasa Isyarat Indonesia (BISINDO) yang mengadopsi kebudayaan dan bahasa daerah, sehingga implementasi bahasa isyarat di setiap daerahnya berbeda-beda. Kemudian, Sistem Isyarat Bahasa Isyarat Indonesia (SIBI) yang bentuk tatanan sistemasis isyarat jari, tangan, serta beberapa gerakan anggota tubuhnya melambangkan kosakata tertentu dalam Bahasa Indonesia (Aisyah Muhammad Amin *et al*., 2022). Penelitian ini berfokus pada pengenalan *pose* tangan bahasa isyarat angka SIBI 0 sampai 9, dimana setiap angka diwakili oleh *pose* tangan tertentu dengan posisi jari yang spesifik. *Pose* tersebut berfungsi sebagai fitur unik untuk mengenali perbedaan satu angka dengan angka lainnya.

Terdapat beberapa penelitian yang telah membahas sistem pengenalan *pose* tangan dalam bahasa isyarat, diantaranya penelitian oleh Ibnu Ahdiat Miharja (2024) yang

menggunakan Mediapipe tepatnya modul *Hand Tracking* untuk mendeteksi pergerakan tangan angka 0-4 berdasarkan jumlah jari yang terangkat. Sistem yang dihasilkan berhasil mendeteksi gerakan angka dengan stabil. Berikutnya, penelitian oleh Arfah Athiroh (2022) yang berhasil mengembangkan sistem untuk mendeteksi posisi dan bentuk tangan bahasa isyarat angka 1-10 dengan memisahkan objek (tangan) dari latar belakang berdasarkan tingkat keabuan citra. Begitu juga, penelitian oleh Arpita Halder & Akshit Tayade, (2021) yang memanfaatkan Mediapipe untuk mengekstrak 21 titik *landmark pose* tangan bahasa isyarat. Model yang dihasilkan dilatih dengan algoritma *Support Vector Machine* (SVM), yang secara umum akan tetap mengklasifikasikan *input* data ke dalam salah satu kelas tertentu. Hal ini menjadi keterbatasan algoritma SVM dalam menangani masalah klasifikasi bahasa isyarat, karena memungkinkan *pose* di luar bahasa isyarat tetap diklasifikasikan ke dalam salah satu kelas tertentu, sehingga mengganggu proses penerjemahan bahasa isyarat.

Untuk mengatasi keterbatasan tersebut, penelitian ini mengusulkan pendekatan dengan memanfaatkan *landmark* tangan dari Mediapipe dan model *Multi Layer Perceptron* (MLP), yakni jenis jaringan saraf tiruan yang efektif digunakan dalam masalah klasifikasi. Beberapa kelebihan model MLP, diantaranya mampu menyesuaikan diri dengan data, bisa memprediksikan hubungan kompleks antar fitur dalam dataset, lebih *reliabel* terhadap *noise* dalam data, serta mampu memperoleh tingkat akurasi yang tinggi, sehingga cocok digunakan dalam masalah klasifikasi *pose* tangan karena dapat menangkap pola gerakan tangan yang kompleks (Pardede *et al*., 2022). Pada penelitian ini, MLP digunakan untuk mengklasifikasikan *pose* tangan berdasarkan fitur jarak *Euclidean* antar titik *landmark*. Jarak *Euclidean* diartikan sebagai matriks yang biasa digunakan untuk mengukur jarak, terutama yang berkaitan dengan metode perbandingan (Insani *et al*., 2023). Jarak *Euclidean* membantu dalam menentukan tingkat kemiripan dua *landmark* tangan berdasarkan posisi geometrisnya. Selain itu, penelitian ini juga menerapkan *threshold* pada tahap evaluasi dan implementasi *real-time* untuk menetapkan batasan pada model dalam mengklasifikasikan setiap *input* data yang diterima.

METODE PENELITIAN

Prosedur penelitian ini terdiri atas beberapa tahapan, meliputi *Data Collection, Hand Landmark Extraction, Feature Extraction, Model Training, Model Evaluation,* dan *Real-Time Implementation* seperti *flowchart* pada Gambar 1.

Gambar 1. *Flowchart* Prosedur Penelitian

Tahapan-tahapan pada prosedur penelitian di atas, akan diuraikan lebih lanjut seperti berikut ini:

1. *Data Collection*

Pada tahap *Data Collection* atau pengumpulan data, terdapat beberapa metode yang digunakan, diantaranya eksplorasi dan analisis data dari sumber internet, yaitu Kaggle. Data yang diperoleh berupa gambar *corpus pose* bahasa isyarat angka SIBI 0-9 yang terdiri dari 10 kelas. Kemudian, pengumpulan data secara langsung oleh peneliti melalui pengambilan gambar menggunakan kamera laptop. Data terdiri dari 1 kelas *random* atau *non-pose* yang mencakup semua variasi *pose* tangan di luar *pose* angka

Tabel 1. Contoh Kelas *Pose*

pose_0	***pose_1***	***pose_2***	***pose_3***	***pose_4***	***pose_5***
					
pose_6	***pose_7***	***pose_8***	***pose_9***	***pose_random***	
					

0-9. Kelas *random* ditujukan untuk meningkatkan keyakinan model klasifikasi saat diimplementasikan. Dengan kata lain, kelas *random* mewakili kondisi-kondisi yang ada pada *real-word*. Sehingga, model mampu mengenali dan membedakan *pose* tangan yang tidak sepenuhnya sesuai dengan kelas 0-9, serta lebih *robust* dalam menghadapi kasus *input* ambigu atau tidak dikenali, yang mungkin terjadi dalam situasi nyata. Oleh karena itu, jumlah kelas yang digunakan pada penelitian ini terdiri dari 11 kelas.

Setelah semua data terkumpul, dilakukan wawancara untuk mengevaluasi serta memvalidasi bahwa data yang digunakan memang sudah sesuai dengan kebutuhan klasifikasi bahasa isyarat angka SIBI. Validator atau *expert judgment* pada tahap ini adalah Guru SLB di salah satu sekolah khusus penyandang disabilitas. Contoh kelas *pose* yang digunakan pada klasifikasi ini dapat dilihat seperti Tabel 1 di atas.

2. *Hand Landmark Extraction*

Tahap *Hand Landmark Extraction* dilakukan menggunakan Mediapipe *Hand Landmark Model* untuk mengidentifikasi 21 titik kunci atau *keypoints* pada tangan, yang kemudian digunakan untuk membangun dataset dalam pelatihan model (Chen *et al*., 2022). Proses dimulai dengan menangkap gambar *pose* tangan 2D, kemudian mendeteksi posisi *landmark* tangan pada gambar. *Landmark* yang terdeteksi meliputi titik-titik pada pergelangan tangan, jari-jari, dan pangkal jari berupa koordinat kartesius 3D (x, y, z). Koordinat yang dihasilkan biasanya berada dalam rentang 0 hingga 1, karena Mediapipe secara otomatis menormalisasi koordinat *landmark* berdasarkan ukuran gambar yang diproses. Hal ini memungkinkan sistem untuk mendeteksi *landmark* dengan konsisten dan akurat meskipun gambar memiliki ukuran yang berbeda. Ilustrasi tahap *Hand Landmark Extraction* dapat dilihat pada Gambar 2.

Gambar 2. Ilustrasi *Hand Landmark Extraction*

Banyak penelitian yang telah menggunakan metode *hand landmark extraction* dengan mediapipe, diantaranya penelitian oleh Muhammad Rifki Pratama Nautica (2022) yang menggunakan mediapipe untuk mendeteksi gestur tangan sebagai alat bantu ajar berhitung bagi anak-anak, hasil menunjukkan aplikasi praktis dalam proses ekstraksi *landmark* untuk mengembangkan sistem edukasi berbasis teknologi. Selanjutnya, penelitian terkait pengembangan sistem yang dapat mengenali gerakan tangan dalam bahasa isyarat SIBI melalui ekstraksi *landmark* tangan dari sebuah citra yang diambil (Ardiansyah *et al*., 2024). Berikutnya, penelitian yang membandingkan proses pengembangan media pembelajaran berhitung berbasis gestur jari tangan menggunakan Mediapipe dan arsitektur *Convolutional Neural Network* (CNN) (Heri Pratikno *et al*., 2023). Hasil pengujian menunjukkan bahwa penggunaan mediapipe menghasilkan akurasi

yang lebih tinggi, dibandingkan dengan CNN. Kemudian, penelitian yang menggunakan mediapipe *hands* untuk mengekstrak koordinat *landmark* tangan selama proses pengumpulan data melalui *webcam*, sehingga menghasilkan dataset dari pola tangan untuk setiap abjad SIBI, dengan akurasi pengujian yang cukup tinggi (Maryamah *et al*., 2023). Selanjutnya, penelitian oleh R. B. Wishnumurti, (2023) yang menggunakan *hand landmark* mediapipe untuk mendeteksi gerakan tangan secara *real-time* dengan menghitung jari yang terangkat berdasarkan posisi *landmark*. Terakhir, penelitian yang menggunakan mediapipe untuk mengumpulkan dataset melalui rekaman gestur tangan, dan berhasil membuktikan bahwa penggunaan gestur tangan untuk mengendalikan presentasi *PowerPoint* secara *real-time* adalah solusi inovatif (Agustiani *et al*., 2024). Hal ini menunjukkan bahwa penggunaan mediapipe cukup efektif dalam mengekstrak *landmark* tangan untuk berbagai aplikasi pengenalan pose maupun gestur tangan.

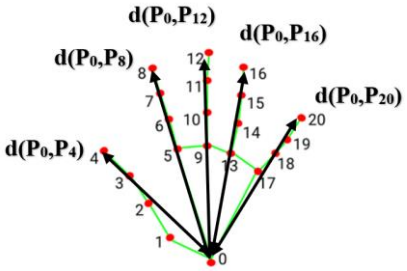
3. *Feature Extraction*

Tahap *Feature Extraction* dilakukan dengan menyeleksi fitur dari 21 titik *landmark* tangan yang telah dihasilkan pada tahap sebelumnya. Peneliti menggunakan beberapa pasang *landmark* yang dianggap relevan terhadap *pose* tangan angka 0-9, diantaranya (0,4), (0,8), (0,12), (0,16), dan (0,20). Selanjutnya, setiap pasang *landmark* dihitung jarak *Euclidean* nya dengan menggunakan rumus di bawah ini.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Keterangan:
d = jarak
x_1, y_1, z_1 = koordinat titik pertama
x_2, y_2, z_2 = koordinat titik kedua

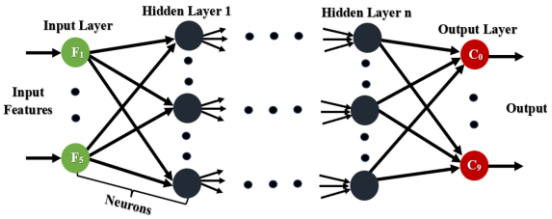
Hasil perhitungan tersebut digunakan untuk membangun dataset dalam pelatihan model klasifikasi. Setelah nilai jarak diperoleh, dilakukan proses normalisasi dengan memilih kembali jarak acuan dari 21 titik *landmark*. Proses ini dilakukan dengan cara membagi nilai fitur dengan jarak acuan. Sehingga, nilai fitur yang dihasilkan lebih relatif terhadap ukuran tangan dalam gambar, bukan nilai absolut koordinat, sehingga ukuran dan orientasi tangan akan konsisten ketika dilakukan proses deteksi oleh mediapipe. Hasil akhir dari tahap ini adalah dataset berupa numerik yang terdiri dari nilai jarak relatif antar pasangan koordinat *landmark* yang telah dinormalisasi. Berikut ilustrasi fitur jarak *Euclidean* yang digunakan dapat dilihat pada Gambar 3. di bawah ini.



Gambar 3. Ilustrasi *Feature Extraction*

4. *Model Training*

Tahap *Model Training* diawali dengan pembagian data (*data splitting*) dari dataset fitur sebelumnya. Setelah itu, dilakukan pelatihan model dengan menggunakan model MLP. Model MLP dirancang dengan beberapa lapisan (*layer*) diantaranya *input layer* yang menerima dataset 5 fitur jarak *Euclidean*, beberapa *hidden layer* untuk meningkatkan kemampuan model dalam mengenali *pose* tangan, dan *output layer* yang terdiri dari 11 *neuron* untuk mengklasifikasi 11 kelas yaitu *pose* angka dari 0-9 dan kelas *non-pose*. Proses pelatihan dilakukan sesuai jumlah *epoch* yang telah ditentukan dengan memantau performa pada data validasi dalam pelatihan (Adeyanju *et al*., 2021). Ilustrasi arsitektur yang digunakan dapat dilihat pada Gambar 4. di bawah ini.



Gambar 4. Ilustrasi Arsitektur Model MLP

Beberapa penelitian terdahulu telah menerapkan model MLP untuk menangani permasalahan klasifikasi, seperti penelitian terkait pengembangan sistem pengenalan bahasa isyarat (SLR) yang membuktikan bahwa model MLP mampu mencapai tingkat akurasi yang cukup tinggi, sehingga meningkatkan performa sistem dan MLP menjadi salah pilihan yang tepat dalam pengembangan model klasifikasi (Rajalingam *et al*., 2022). Kemudian, penelitian terkait klasifikasi penduduk kurang mampu melalui

penerapan model MLP dalam menangkap karakteristik dari data numerik dengan cukup efektif (Gulo & Lubis, 2024).

5. *Model Evaluation*

Pada tahap *Model Evaluation*, kinerja model klasifikasi diuji menggunakan data *testing* yang dibandingkan hasil prediksinya dengan nilai *threshold* yang ditentukan untuk menyaring hasil prediksi berdasarkan tingkat probabilitasnya. Jadi hanya prediksi yang memiliki nilai probabilitas atau *confidence* di atas nilai *threshold* tertentu yang dianggap valid, sementara prediksi di bawah nilai tersebut akan diklasifikasikan ke dalam kelas "Tidak Dikenali" dengan label (-1). Hal ini membantu meningkatkan kehandalan model dalam mengenali kelas-kelas *non-pose*, sehingga dalam implementasinya dapat mengurangi prediksi yang salah.

Kemudian, keseluruhan nilai baik nilai di atas maupun di bawah *threshold* dihitung menggunakan beberapa metrik evaluasi, yaitu *accuracy* untuk mengukur proporsi prediksi benar dari total prediksi. Selanjutnya, *precision* yang menunjukkan seberapa banyak prediksi positif yang benar, *recall* mengukur seberapa banyak total kasus positif yang berhasil diidentifikasi oleh model, dengan parameter yang sama. Terakhir, *f1-score* memberikan gambaran yang lebih seimbang terkait kinerja model.

Selain itu, evaluasi juga diimplementasikan melalui *Confusion matrix* yang divisualisasikan menggunakan *heatmap* untuk memudahkan pemahaman dan analisa terkait informasi yang disajikan. Dengan *heatmap*, hasil prediksi model dapat dilihat terkait seberapa sering model memprediksi dengan benar (di diagonal utama) dan dimana kecenderungan prediksi salah terjadi (di luar diagonal utama). Setiap baris pada *Confusion matrix* menunjukkan jumlah data aktual untuk setiap kelas, sementara kolom menunjukkan jumlah prediksi untuk kelas tersebut. Dengan menggunakan metode evaluasi ini, peneliti dapat memahami seberapa baik model bekerja dalam mengenali *pose* tangan bahasa isyarat angka SIBI serta mengetahui kehandalan model saat mengeksekusi data baru.

6. *Real-Time Implementation*

Pada tahap *Real-Time Implementation*, model klasifikasi diintegrasikan dengan *webcam* pada laptop untuk mendeteksi *pose* tangan secara langsung. Proses dimulai dengan menginisialisasi *webcam* untuk menangkap video *real-time*, kemudian diproses menggunakan mediapipe untuk mendeteksi 21 *landmark* tangan secara otomatis. Setelah *landmark* tangan terdeteksi, fitur pada model dihitung, kemudian diprediksi secara *real-time*. Pada tahap ini, diterapkan *threshold* untuk menampilkan hasil prediksi yang konsisten antara tahapan evaluasi dan implementasi model, dimana input yang dianggap tidak valid akan diprediksi sebagai "Tidak Dikenali". Hasil prediksi ini ditampilkan secara langsung pada jendela video, sehingga memungkinkan pengguna untuk melihat angka yang dikenali oleh sistem.

HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan model klasifikasi bahasa isyarat angka SIBI yang dikembangkan dan divisualisasikan secara interaktif menggunakan *Jupyter Notebook*. Pembahasan terkait hasil penelitian ini, akan diuraikan sebagai berikut:

1. *Data Collection*

Hasil tahap *data collection* meliputi pengumpulan data *corpus pose* angka 0-9 yang diperoleh dari Kaggle sejumlah 200 gambar untuk masing-masing kelas, sementara kelas *random* berhasil dikumpulkan sebanyak 290 gambar melalui kamera laptop. Untuk memastikan model tidak bias terhadap salah satu kelas tertentu, dilakukan penambahan data pada masing-masing kelas angka 0-9, agar jumlah sampel antar kelas tidak jauh berbeda. Sehingga, jumlah data dari setiap kelas yang digunakan berkisar 280 hingga 300 gambar.

2. *Hand Landmark Extraction*

Hasil proses ekstraksi *landmark* tangan dalam penelitian ini adalah koordinat *landmark* tangan yang terdiri dari koordinat x, y, z dan disimpan dalam format file .csv. Setiap baris dalam file tersebut mewakili satu titik *landmark* pada tangan, dengan kolom data yang dihasilkan mencakup kolom pertama menunjukkan label kelas, kolom kedua menunjukkan koordinat x, kolom ketiga menunjukkan koordinat y, dan kolom keempat menunjukkan koordinat z.

Dari 3.161 gambar yang digunakan pada penelitian ini, hanya 2.257 baris data *landmark* tangan yang berhasil diekstraksi. Perbedaan ini menunjukkan bahwa tidak semua gambar dapat dideteksi oleh model. Hal ini terjadi,

karena kualitas gambar atau posisi tangan dalam gambar yang diekstrak kurang jelas, sehingga nilai *confidence* yang diberikan oleh model terhadap beberapa gambar tersebut berada di bawah nilai ambang batas yang ditentukan. Kondisi demikian, yang menyebabkan model gagal dalam mendeteksi dan mengekstrak *landmark* tangan pada gambar.

Dalam prosesnya, inisialisasi model *Hand Landmark* pada Mediapipe dilakukan melalui modul *mp.solutions.hands* dengan beberapa parameter seperti terlihat pada Tabel 2.

Tabel 2. Parameter Inisialisasi Model *Hand Landmark* Mediapipe

Parameter	***Value***	**Keterangan**
static_image_ mode	*True*	Memproses gambar statis
max_num_ hands	1	Deteksi hanya pada satu tangan
min_detection _confidence	0.7	Tingkat kepercayaan minimal 70%

Kemudian, proses ekstraksi dilakukan melalui pembacaan setiap gambar menggunakan *library OpenCV*, dan menjalankan fungsi *extract_hand_ landmarks* yang akan mendeteksi dan mengekstraksi gambar dengan mengembalikan tiga nilai koordinat *landmark* tangan (x, y, z). Jika tidak, fungsi akan mengembalikan nilai *None* atau kosong.

Proses ini dilakukan untuk semua gambar pada data *corpus* yang dikelompokkan berdasarkan kelas menggunakan *os.listdir* yang mengintegrasikan setiap subfolder dalam direktori dataset. Hasil dari ekstraksi tersebut disimpan dalam list *row*. Kolom data yang dihasilkan mencakup label kelas serta koordinat *landmark* tangan yang terdeteksi, dan disusun dalam format tabel menggunakan fungsi *pandas.DataFrame* kemudian disimpan dalam format file .csv, seperti terlihat pada Tabel 3.

Tabel 3. Contoh Hasil *Hand Landmark Extraction*

Class	**x_0**	**...**	**z_20**
0	0.3791055 083274841	...	-0.142570 0634717941
1	0.4943549 036979675	...	-0.070178 3448457717
....	....	...	....
random	0.5172299 742698669	...	-0.137395 2031135559

3. *Feature Extraction*

Hasil pada tahap *Feature Extraction* ini adalah nilai jarak *Euclidean* dari beberapa pasang *landmark* yaitu (0,4), (0,8), (0,12), (0,16), dan (0,20) yang telah dinormalisasi menggunakan satu pasang jarak acuan yang relatif konsisten dalam berbagai kondisi *pose* tangan, sehingga tidak terpengaruh oleh jarak tangan ke kamera.

Dalam prosesnya, perhitungan fitur jarak *Euclidean* hanya menggunakan beberapa koordinat *landmark* yang terpilih. Perhitungan tersebut didefinisikan dalam fungsi *euclidean_distance* yang diterapkan pada setiap baris dataset menggunakan fungsi *apply* dan hasilnya disimpan dalam kolom *dist_0_4*, *dist_0_8*, *dist_0_12*, *dist_0_16*, dan *dist_0_20*, bersama dengan kolom *class* pada format file .csv seperti pada Tabel 4.

Table 4. Fitur Jarak *Euclidean*

Class	***dist_0_4***	**...**	***dist_0_20***
0	0.33982836 947274	...	0.286031446 33456135
1	0.30401831 8707534	...	0.155664256 40585776
....	....	...	....
random	0.45512197 837830515	...	0.288110628 2551966

Setelah proses perhitungan fitur jarak *Euclidean*, dilakukan proses normalisasi. Proses diawali dengan menghitung jarak acuan yang telah ditentukan yaitu jarak *Euclidean* antara *landmark* 0 dan 9. Nilai jarak acuan ini kemudian divalidasi untuk memastikan tidak ada nilai 0 atau negatif yang dapat menyebabkan *error* pada proses normalisasi. Kemudian, setiap fitur jarak *Euclidean* dinormalisasi dan hasilnya disimpan kembali dalam format file .csv, seperti pada Tabel 5.

Tabel 5. Fitur Jarak *Euclidean*

Class	***dist_0_4***	**...**	***dist_0_20***
0	1.20490090 52150161	...	1.014157673 6021127
1	1.18748897 83153284	...	0.502853524 4609066
....	....	...	....
random	1.31195864 63276134	...	0.521502847 1665571

4. *Model Training*

Model klasifikasi bahasa isyarat angka SIBI dirancang menggunakan model MLP dengan akurasi selama proses pelatihan sebesar 93%. Proses diawali dengan membagi

dataset menjadi dua yaitu 80% *data train* dan 20% *data test*, yang dilakukan menggunakan kode program *Python* untuk memastikan model dilatih dengan data yang cukup dan diuji dengan data yang tidak ada di dalam pelatihan.

Setelah itu, model MLP dirancang dengan *input* menggunakan *X_train.shape[1]* yang akan memberikan jumlah fitur atau kolom pada dataset. Model menggunakan arsitektur *Sequential* yang terdiri dari tiga *hidden layer*, yaitu *hidden layer* pertama (lapisan *Dense*) memuat 64 *neuron*, kedua dengan 32 *neuron*, dan ketiga dengan 16 *neuron*. Setiap *layer* menggunakan fungsi aktivasi *ReLU* untuk memetakan nilai negatif menjadi 0, dan mempertahankan nilai positif. Kemudian, *Dropout* sebesar 30% untuk mencegah *overfitting* dengan menghapus secara acak *neuron* tertentu, serta *BatchNormalization* untuk stabilitas pelatihan. Terakhir, *output layer* yang terdiri dari 11 *neuron* sesuai jumlah kelas yang diklasifikasikan dan menggunakan fungsi aktivasi *softmax* untuk menghasilkan nilai probabilitas pada klasifikasi *multiclass*.

Pelatihan model klasifikasi dilakukan melalui penentuan *hyperparameter* dengan metode *tuning hyperparameter* yaitu *Grid Search* untuk mencari kombinasi terbaik dari *learning rate* (0.001, 0.0005, dan 0.0001) dan *batch size* (16, 32, 64). Sehingga diperoleh hasil terbaik seperti pada Tabel 3 berikut ini.

Tabel 6. *Hyperparameter*

Hyperparameter	***Value***
learning_rate	0.0005
batch_size	64

Serta penggunaan fungsi *loss sparse_ categorical_crossentropy* pada tahap ini dikarenakan *output* label berupa angka.

Selain itu, implementasi *callback* dalam pelatihan ini seperti *EarlyStopping* untuk menghentikan pelatihan jika model tidak menunjukkan perbaikan, dan *ReduceLROnPlateau* untuk menurunkan *learning rate* secara dinamis jika ditemukan stagnasi, sehingga dapat memaksimalkan proses pelatihan sesuai parameter pada maksimum 100 *epoch* dengan *validation split* sebesar 20%. Proses pelatihan mencatat metrik akurasi dan *loss* untuk evaluasi performa model. Setelah proses pelatihan selesai, model disimpan dalam format file .keras dan label *encoder* disimpan dalam format file .pkl untuk digunakan kembali pada tahap berikutnya.

5. *Model Evaluation*

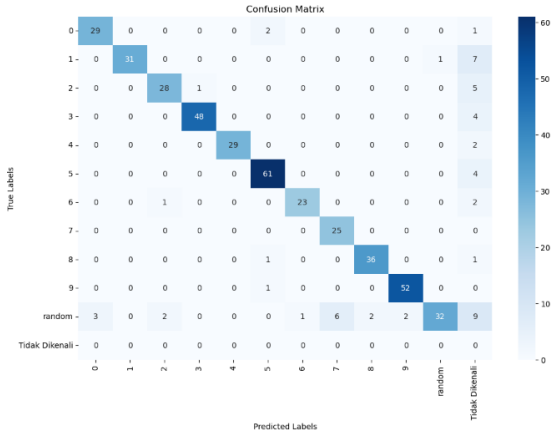
Evaluasi model klasifikasi terhadap data *test* disertai penerapan *threshold* menghasilkan nilai *accuracy* sebesar 87.17%, *precision* 94.92%, *recall* 87.17%, dan *f1-score* 90.16%. Hasil ini diperoleh dengan menghitung jumlah prediksi secara keseluruhan termasuk untuk kelas ”Tidak Dikenali” (-1) yang menampung hasil prediksi di bawah nilai *threshold* yang ditentukan sesuai dengan visualisasi *confusion matrix*.

Proses evaluasi dilakukan dengan memprediksi *output probabilities* yang dihasilkan oleh model menggunakan fungsi *np.argmax* (kelas dengan probabilitas tertinggi). Dalam perhitungan evaluasi beberapa parameter metrik yang digunakan didefinisikan dalam fungsi seperti terlihat pada Tabel 7. di bawah ini.

Tabel 7. Parameter Metrik Evaluasi

Fungsi	**Parameter**
accuracy_score	*y_test_encoded*
	predicted_labels_with _threshold
precision_score, recall_score, f1_score	*y_test_encoded*
	predicted_labels_with _threshold
	average='weighted'
	zero_division=0

Berikutya hasil implementasi *Confusion Matrix* menggunakan *heatmap* ditunjukkan sesuai visualisasi pada Gambar 5. Hasil menunjukkan model memiliki performa yang cukup baik untuk klasifikasi dengan mayoritas prediksi yang dihasilkan benar. Meskipun, masih terdapat beberapa kesalahan prediksi, khususnya pada kelas *random* yang memiliki 1 *False Positive* (data yang sebenarnya bukan kelas *random* tetapi diprediksi sebagai kelas *random*) dan 25 *False Negatif* (data yang sebenarnya kelas *random* tetapi diprediksi sebagai kelas lain). Hal ini menunjukkan pentingnya pengumpulan data tambahan untuk kelas *random*, sehingga model lebih baik dalam mempelajari *pose* diluar kelas angka. Selain itu, terlihat beberapa sampel dari kelas lain diklasifikasikan ke dalam kelas “Tidak Dikenali”, karena model tidak dapat mengenali beberapa input yang berada di bawah nilai threshold. Sehingga, hasil prediksi yang dihasilkan oleh model menunjukkan tingkat kepercayaan yang tinggi.



Gambar 5. *Confusion Matrix*

6. *Real-Time Implementation*

Implementasi secara *real-time* dilakukan menggunakan laptop dengan spesifikasi prosesor Intel(R) Core(TM) i3-1005G1 CPU yang memiliki kecepatan 1.20GHz dan 4 inti, sehingga mampu memproses data secara *real-time*. Kapasitas RAM sebesar 8192MB juga memberikan ruang yang cukup untuk menjalankan model bersamaan dengan aplikasi pendukung lainnya. Selain itu, penggunaan DirectX 12 juga memastikan pemrosesan grafis dapat berjalan dengan lancar.

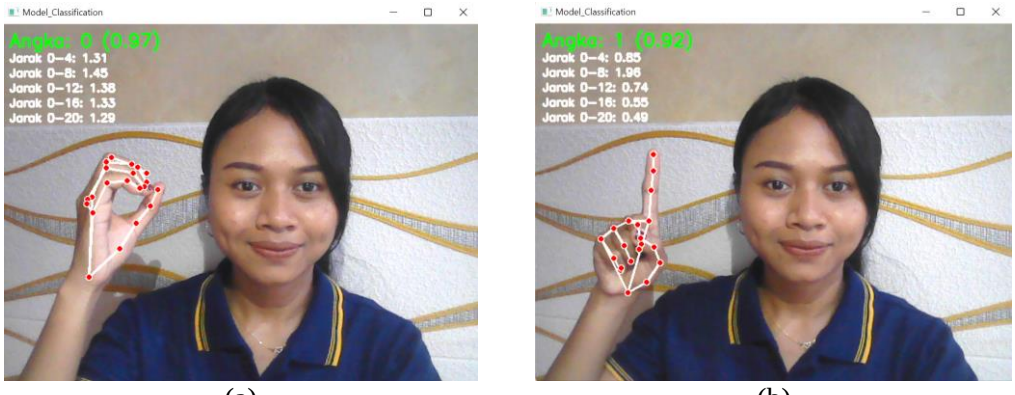
Proses dilakukan dengan menetapkan label untuk setiap kelas termasuk kelas *non-pose* yang diwakili dengan nilai 10 sebagai kelas ”Tidak Dikenali”, serta penerapan *threshold* sebesar 0.7 untuk menentukan apakah suatu prediksi cukup akurat diklasifikasikan. Pada implementasi kode program untuk memulai akses kamera digunakan *cap = cv2.VideoCapture(0)* dan memastikan kamera tersedia menggunakan *if not cap.isOpened()*, jika tidak, program akan berhenti. Inisialisasi Mediapipe *Hands* dilakukan dengan parameter deteksi dinamis seperti terlihat pada Tabel 8.

Tabel 8. Parameter Inisialisasi Mediapipe *Hands* pada *Implementation Real-Time*

Parameter	*Value*	Keterangan
static_image_ mode	*False*	Memungkinkan deteksi secara *real-time*
max_num_ hands	1	Deteksi hanya pada satu tangan
min_detection _confidence	0.7	Tingkat kepercayaan minimal 70%

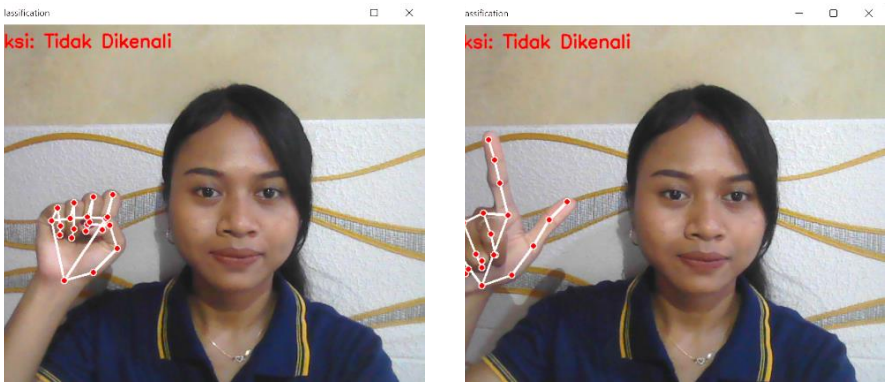
Selama kamera aktif, setiap *frame* dibaca menggunakan *cap.read()*, dan jika gagal, program keluar dari *loop*. *Frame* yang berhasil dibaca dikonversi dari format BGR ke RGB agar sesuai dengan *input* mediapipe, kemudian diproses untuk mendeteksi *landmark* tangan yang digunakan untuk menghitung jarak *Euclidean* sebagai fitur prediksi.

Prediksi angka secara *real-time* dilakukan berdasarkan jarak *Euclidean* dari *landmark* tangan yang terdeteksi dan menampilkannya pada *frame* video. Pada implementasi kode programnya digunakan fungsi *predict_landmark (distances)* untuk mendapatkan ID kelas (*class_id*) dan tingkat kepercayaan (*confidences*). Kemudian, ID kelas diubah menjadi label angka menggunakan *class_labels.get (class_id, ”Tidak Dikenali”)*. Pada implementasi model ini, pengguna akan menerima *feedback* visual berupa prediksi nilai angka dilengkapi dengan nilai-nilai fitur jarak *Euclidean* yang dihasilkan, apabila hasil *input* dianggap valid seperti pada Gambar 6. Sementara itu, untuk *input* model yang dianggap tidak valid, yakni data diluar *pose* angka 0-9 serta data dengan



(a) (b)

Gambar 6. Hasil *Input* Valid



Gambar 7. Hasil *Input* Tidak Valid

probabilitas untuk prediksi berada di bawah *threshold*, akan memberikan *feedback* kepada pengguna berupa prediksi ”Tidak Dikenali”, seperti pada Gambar 7.

KESIMPULAN

Berdasarkan pembahasan pada bab sebelumnya, dapat disimpulkan bahwa penelitian ini berhasil mengembangkan model klasifikasi bahasa isyarat angka menggunakan MLP dengan fitur jarak *Euclidean landmark* tangan. Peneliti berhasil mengekstrak *landmark* tangan dari data *corpus* yang dikumpulkan, dengan jumlah data berkisar pada 280-300 gambar setiap kelasnya. Data diekstraksi menggunakan model *landmark* tangan Mediapipe, dan disimpan dalam format .csv. Selanjutnya, penelitian ini telah membangun dataset fitur jarak *Euclidean* dari beberapa pasang *landmark* tangan yang relevan, dan dinormalisasi sehingga dataset yang dihasilkan lebih relatif terhadap dimensi gambar.

Model klasifikasi dirancang menggunakan arsitektur MLP yang terdiri atas tiga *hidden layer*, serta penerapan teknik *Batch Normalization* dan *Dropout* untuk menghindari *overfitting*. Selain itu, *EarlyStopping* dan *ReduceLROnPlateau* juga digunakan untuk mengoptimalkan proses pelatihan dan mencegah *stagnasi*. Akurasi pelatihan model diperoleh sebesar 93%, sementara dalam pengujian model menghasilkan *accuracy* sebesar 87.17%, *precision* 94.92%, *recall* 87.28%, dan *F1-score* 90.16%. Berdasarkan hasil *confusion matrix*, model menunjukkan performa yang cukup baik, dengan sebagian besar prediksi yang dihasilkan bernilai benar. Meskipun beberapa sampel dari kelas lain diklasifikasikan ke dalam kelas “Tidak Dikenali” karena *input* di bawah nilai *threshold*, hal ini menandakan bahwa setiap prediksi yang dihasilkan oleh model memiliki tingkat kepercayaan yang tinggi.

Dari hasil penelitian ini, disarankan agar penelitian selanjutnya dapat mengeksplorasi metode atau algoritma lainnya yang dapat meningkatkan kinerja model klasifikasi. Selain itu, *pose* yang diklasifikasikan bisa mencakup angka di luar rentang 0-9 dalam bahasa isyarat SIBI.

REFERENSI

Adeyanju, I. A., Bello, O. O., & Adegboye, M. A. (2021). Machine learning methods for sign language recognition: A critical review and analysis. *Intelligent Systems with Applications*, *12*, 200056. https://doi.org/10.1016/j.iswa.2021.200056

Agustiani, A. D., Putri, S. M., Hidayatullah, P., & Sholahuddin, M. R. (2024). *Penggunaan MediaPipe untuk Pengenalan Gesture Tangan Real-Time dalam Pengendalian Presentasi*. *16*(2).

Aisyah Muhammad Amin, N., Pribadi, F., & Kunci, K. (2022). Urgensi Bahasa Isyarat dalam Pendidikan Formal sebagai Media Komunikasi dan Transmisi Informasi Penyandang Disabilitas Rungu dan Wicara. *Jurnal Hasil Pemikiran, Penelitian, Dan Pengembangan Keilmuan Sosiologi Pendidikan*, *9*(1), 77–86.

Athiroh, A. (2022). Meningkatkan Akurasi Pembacaan Bahasa Isyarat Angka Dengan Menggunakan Model *Landmark* Tangan dan Algoritma *Thresholding*. Universitas Pendidikan Indonesia. Diakses dari: https://repository.upi.edu/perpustakaan.u

pi.edu.

Ardiansyah, A. R., Nur'azizan, A. H., & Fernandis, R. (2024). Implementasi Deteksi Bahasa Isyarat Tangan Menggunakan OpenCV dan MediaPipe. *Stains (Seminar Nasional Teknologi & Sains)*, *3*(1), 331–337.

Arpita Halder, & Akshit Tayade. (2021). Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning. *International Journal of Research Publication and Reviews*, *2*(5), 9–17. www.ijrpr.com

Chen, R. C., Manongga, W. E., Dewi, C., & Chen, H. Y. (2022). Automatic Digit Hand Sign Detection With Hand Landmark. *Proceedings - International Conference on Machine Learning and Cybernetics*, *2022-September*(September), 6–11. https://doi.org/10.1109/ICMLC56445.2022.9941325

Gulo, S. H., & Lubis, A. H. (2024). Penerapan Multi-Layer Perceptron untuk Mengklasifikasi Penduduk Kurang Mampu. *Explorer*, *4*(2), 51–59.

Heri Pratikno, Muhammad Rifki Pratama, Yosefine Triwidyastuti, & Musayyanah. (2023). Pengenalan Gestur Jari Tangan Sebagai Media Pembelajaran Berhitung Bagi PAUD Berbasis Visi Komputer Dan Deep Learning. *Journal of Computer Electronic and Telecommunication*, *4*(1). https://doi.org/10.52435/complete.v4i1.355

Insani, C. N., Arifin, N., & Rasyid, M. R. (2023). Deteksi Gerakan Bahasa Isyarat Menggunakan Euclidean Distance. *Informatik : Jurnal Ilmu Komputer*, *19*(1), 99–106. https://doi.org/10.52958/iftk.v19i1.5658

Maryamah, M., Pratama, M. A., Erfit, M. R., Farhani, N. M., & Hartono, I. A. (2023). Klasifikasi Abjad SIBI (Sistem Bahasa Isyarat Indonesia) menggunakan Mediapipe dengan Metode Deep Learning. *Prosiding Seminar Nasional Sains Data*, *3*(1), 134–141. https://doi.org/10.33005/senada.v3i1.102

Mediapipe. (2023). "*Hands*". Tersedia pada https://mediapipe.readthedocs.io/en/latest/solutions/hands.html (diakses tanggal 20 September 2024).

Miharja, I. A. (2024). Sistem Pembacaan Deretan Empat Angka Secara *Computer Vision* Melalui Deteksi *Gesture* Jari Tangan Menggunakan Mediapipe. Tugas Akhir, Fakultas Teknologi dan Informatika, Universitas Dinamika. Diakses dari: https://repository.dinamika.ac.id/id/eprint/7752/1/20410200016-2024-UNIVERSITASDINAMIKA.pdf.

Nautica, M. R. P. (2022). *Hand Gesture Detection* Sebagai Alat Bantu Ajar Berhitung Menggunakan Mediapipe dan *Convolutional Neural Network* Secara *Realtime*. Tugas Akhir, Fakultas Teknologi dan Informatika, Universitas Dinamika. Diakses dari: https://repository.dinamika.ac.id/id/eprint/6650/13/18410200038-2022-UNIVERSITAS%20DINAMIKA.pdf.

Pardede, D., Hayadi, B. H., & Iskandar. (2022). Kajian Literatur Multi Layer Perceptron Seberapa Baik Performa Algoritma Ini. *Journal of Ict Aplications and System*, *1*(1), 23–35. https://doi.org/10.56313/jictas.v1i1.127

Rajalingam, B., Kumar, R. S., Deepan, P., Santosh Kumar Patra, P., & Bavankumar, S. (2022). A Smart System for Sign Language Recognition using Machine Learning Models. *Proceedings - 2022 4th International Conference on Advances in Computing, Communication Control and Networking, ICAC3N 2022*, *December*, 1125–1131. https://doi.org/10.1109/ICAC3N56670.2022.10074007

Wishnumurti, R. B. (2023). *Deteksi Pose dan Penjumlahan Jari dari Gerakan Tangan*. *13519203*.