

## PENGEMBANGAN APLIKASI PENERJEMAH BAHASA ISYARAT INDONESIA (BISINDO) MENGGUNAKAN METODE LONG-SHORT TERM MEMORY

Siti Nur<sup>1\*</sup>, Aghisna Nur Assyifa<sup>2</sup>, Habilah Nurjannah<sup>3</sup>

<sup>1,2,3</sup>Universitas Islam Nusantara, Indonesia

\*e-mail: [sitinur.uninus@gmail.com](mailto:sitinur.uninus@gmail.com)

**Abstract:** The Indonesian Sign Language Translator (BISINDO) application is intended for individuals who are deaf and mute assist them in communicating with normal people. Due to the limitations of technology to assist individuals who are deaf and mute, the Indonesian Sign Language Translator application was developed. Additionally, it is hoped that this application will enable individuals who are deaf and mute to learn Indonesian Sign Language in real-time. LSTM generally consists of a cell, input, gate, and forget gate. LSTM is highly suitable for classifying, processing, and making predictions based on time series data. In this research, hand landmark recognition is used as a medium to store finger gestures from user inputs, which will then be saved in the form of coordinate points and stored in a file in csv format. Subsequently, this file will be trained and called back to determine the output.

**Keywords:** Hand Gesture Recognition, Deep Learning, LSTM

**Abstrak:** Aplikasi penerjemah Bahasa Isyarat Indonesia (BISINDO) merupakan aplikasi yang ditujukan kepada penderita tunawicara dan tunarungu agar dapat membantu berkomunikasi dengan orang normal, karena dengan adanya keterbatasan teknologi untuk membantu penyandang tunarungu dan tunawicara, maka dikembangkanlah aplikasi penerjemah Bahasa Isyarat Indonesia (BISINDO) dan juga aplikasi ini diharapkan agar penderita tunawicara dan tunarungu dapat mempelajari Bahasa Isyarat Indonesia (BISINDO) secara *real time*. LSTM secara umum terdiri dari *cell*, *input*, *gate*, dan *forget gate*. LSTM sangat cocok untuk mengklasifikasikan, memproses, dan membuat prediksi berdasarkan data time series, pada penelitian ini menggunakan *hand landmark recognition* sebagai media untuk menyimpan gestur jari dari inputan *user* yang kemudian akan disimpan dalam bentuk titik koordinat dan akan disimpan dalam file dengan format csv, kemudian file ini akan di *training* dan akan dipanggil kembali untuk menentukan *output*. Dalam beberapa percobaan pengujian, terdapat perbedaan data yang digunakan. Menggunakan 500 data, akurasi adalah 75%. Namun, menggunakan 1000 dan 1500 data hanya meningkatkan akurasi menjadi 85%. Ini menunjukkan bahwa jumlah data bukan satu-satunya faktor yang mempengaruhi peningkatan akurasi. Beberapa faktor lain yang mempengaruhi akurasi adalah kualitas data saat pengambilan koordinat oleh mesin pembelajaran, serta kesesuaian gestur jari saat proses pengolahan dan tampilan *output*.

**Kata kunci:** Hand Gesture Recognition, Deep Learning, LSTM

Copyright (c) 2024 The Authors. This is an open access article under the CC BY-SA 4.0 license (<https://creativecommons.org/licenses/by-sa>)

### PENDAHULUAN

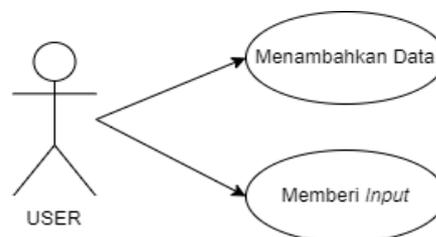
BISINDO adalah bahasa yang digunakan oleh komunitas tunarungu di Indonesia sebagai alat komunikasi mereka. Namun, terdapat kesulitan dalam berkomunikasi antar komunitas tunarungu dengan orang-orang yang tidak mengerti

bahasa isyarat. Oleh karena itu diperlukan teknologi yang dapat membantu menerjemahkan bahasa isyarat kedalam bahasa lisan sehingga memudahkan komunikasi bagi kedua belah pihak. Menurut Winaldi & Setyawan (2018) di Indonesia masih kekurangan media pembelajaran untuk pengenalan dan pembelajaran tentang bahasa isyarat Indonesia berbasis IT dan juga sulitnya bagi anak-anak tunarungu dan tunawicara untuk berkomunikasi dengan orang normal, untuk itu perlu adanya dibuat suatu sistem yang dapat menghubungkan penyandang tunarungu dan tunawicara dengan orang normal. Penelitian aplikasi pengenalan bahasa isyarat secara *real-time* banyak mengalami kendala. Faktor-faktor yang mempengaruhi antara lain yaitu besarnya tingkat kemiripan data citra latih proses pelacakan (*tracking*) terutama pada segmentasi objek dengan latar belakang, sehingga hasil capture tidak diinterpretasikan secara maksimal (Winaldi & Setyawan, 2018). Tentunya penelitian ini memiliki maksud dan tujuan, yaitu memberikan solusi terhadap permasalahan yang telah diuraikan di atas. Adapun tujuan penelitian ini diantaranya: 1) Mengimplementasikan *Long Short-Term Memory* (LSTM) untuk menerjemahkan data input ke Bahasa Isyarat Indonesia (BISINDO), 2) Mengetahui performa model *Long Short-Term Memory* (LSTM) dalam menerjemahkan Bahasa Isyarat Indonesia (BISINDO).

Dalam konteks penelitian, kebaruan merujuk pada kontribusi baru yang dimana dalam penelitian ini terdapat: 1) Peningkatan akurasi: fokus pada peningkatan akurasi aplikasi penerjemah BISINDO menggunakan metode LSTM. Mengoptimalkan arsitektur LSTM, menambahkan lapisan tambahan, atau menggabungkan Teknik pembelajaran mesin lainnya untuk meningkatkan kinerja aplikasi, 2) Pengembangan antarmuka pengguna: pada pengembangan antarmuka pengguna yang lebih intuitif dan mudah digunakan, menambahkan fitur seperti ini panduan interaktif, peringatan kesalahan, atau integrasi suara untuk meningkatkan pengalaman pengguna, 3) Penerjemahan *Real-Time*: menggunakan teknik deteksi gerakan untuk memulai terjemahan sejak awal gerakan pengguna dan menyajikan hasil secara langsung, 4) Evaluasi dan Pengujian: selain mengembangkan aplikasi, perlu memerlukan evaluasi dan pengujian yang komprehensif. Dengan melakukan pengujian pengguna dengan partisipan berbahasa isyarat untuk mengukur kualitas terjemahan, kendala aplikasi dan kepuasan pengguna.

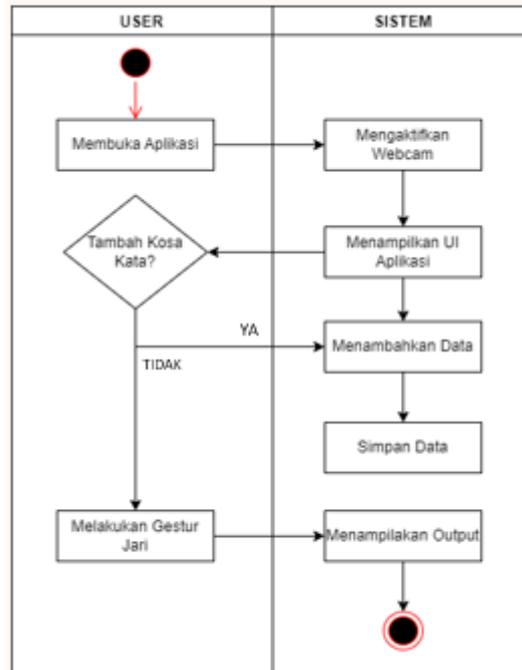
## METODE

Jenis penelitian yang digunakan adalah kuantitatif. Kuantitatif adalah jenis penelitian yang menggunakan pendekatan kuantitatif dalam pengumpulan dan analisis data. Dengan penelitian Meta-analisis yang dilakukan dengan menggabungkan dan menganalisis hasil dari penelitian-penelitian sebelumnya untuk memperoleh kesimpulan yang lebih luas. Rancangan penelitian yang digunakan menggunakan metode *Long Short-Term Memory*, LSTM merupakan bagian dari *Recurrent Neural Network* (RNN). LSTM secara umum terdiri dari *cell*, *input gate*, dan *forget gate*. LSTM sangat cocok untuk mengklasifikasikan, memproses, dan membuat prediksi berdasarkan data *time series*, karena pada proses prediksi ada kelangkaan durasi yang tidak diketahui diantara peristiwa penting dalam rangkaian waktu. Arsitektur umum LSTM terdiri dari *memory cell*, *input gate*, *output gate*, dan *forget gate*. LSTM *cell* mengambil masukan dan menyimpan untuk beberapa waktu. Secara intuitif, *input gate* mengontrol sejauh mana nilai baru akan berjalan ke dalam *cell*, *forget gate* akan mengontrol sejauh mana nilai tetap di dalam *cell*, dan *output gate* mengontrol sejauh mana nilai dalam *cell* digunakan untuk menghitung aktivasi keluaran dari unit LSTM. *Posttest Only Control Group Design* yaitu menggunakan satu kelas eksperimen dan satu kelas kontrol. Perlakuan yang diberikan oleh peneliti pada kelas eksperimen adalah penerapan model pembelajaran *conceptual understanding procedures*, sedangkan untuk kelas kontrol dengan penerapan pembelajaran konvensional. Setelah melaksanakan pembelajaran, siswa diberikan test akhir untuk mengetahui peningkatan pemahaman konsep.



**Gambar 1.** Diagram *Use Case User*

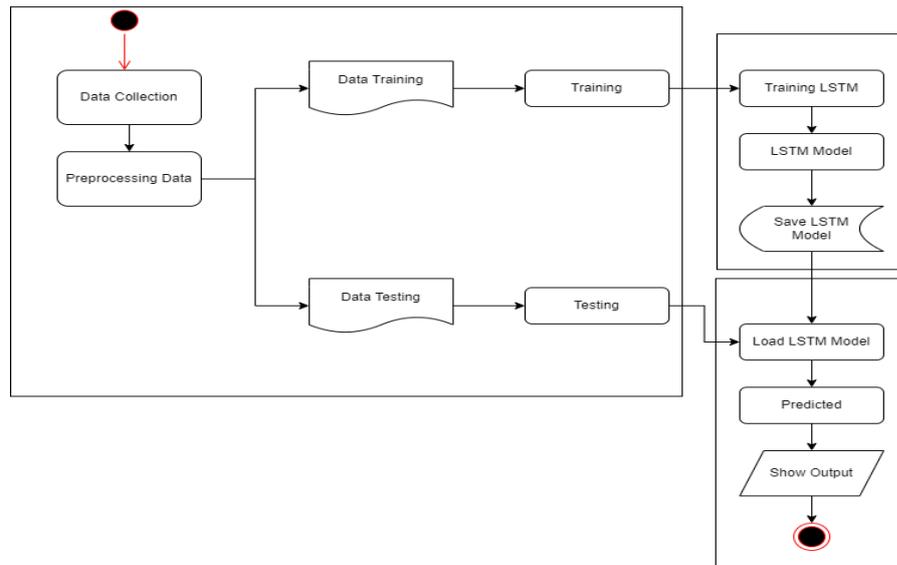
Pada *use case* diatas *user* dapat menambahkan dan memberi input pada aplikasi, setelah aplikasi dijalankan akan muncul antarmuka berupa layar kamera dan *hand landmark* akan mengikuti gestur jari *user*.



Gambar 2. Diagram Activity

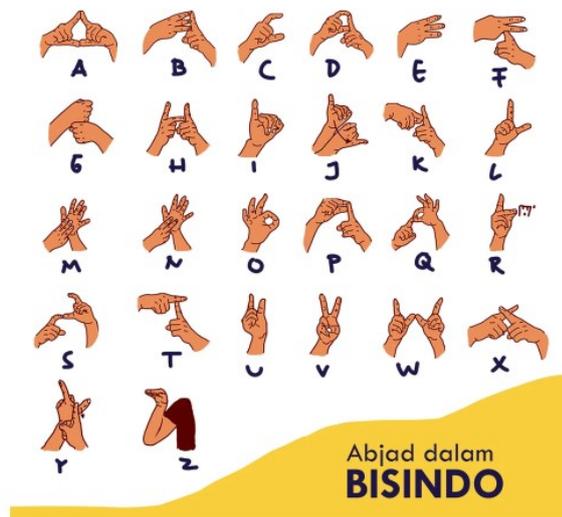
Menjelaskan tentang skema Bahasa Isyarat Indonesia, pertama *user* akan membuka aplikasi terlebih dahulu, setelah itu sistem akan mengaktifkan *webcam* pada komputer dan *user* dapat melakukan gestur jari sebagai inputan ke sistem, ketika masuk ke halaman utama *user* dapat menambahkan data dengan cara menekan tombol angka “1” pada *keyboard*.

Teknik pengolahan data terdapat flowchart sistem dari mulai pengkoleksian data menuju ke *preprocessing* data, data yang telah diproses akan dinormalisasikan dan akan melalui *training* data dan *testing* data, setelah koordinat terkumpul maka sistem akan di tampilkan hasil ketika melakukan gestur jari.



**Gambar 3.** Flowchart Sistem

Teknik pengumpulan data jumlah data yang digunakan pada penelitian ini berjumlah 1000 data. Proses pengumpulan data dilakukan dengan membuat titik koordinat pada aplikasi dengan mengacu pada BISINDO, pada aplikasi yang dikembangkan sudah terdapat fitur untuk menambah data yang akan otomatis membuat titik koordinat, dengan menekan tombol “1” pada *keyboard* akan memasuki mode *logging keypoint* untuk mengubah dari gestur tangan menjadi titik koordinat.



**Gambar 4.** Alphabet BISINDO  
(Sumber: Yayasan Peduli Kasih ABK)

**Tabel 1.** Contoh Data Training BISINDO

No	Data	Terjemahan
1		A
2		B
3		C
4		D
5		1
6		2
7		3

Teknik analisis data pengguna pada aplikasi ini terdapat 3 jenis pengguna diantaranya penderita tunarungu dan tunawicara serta orang normal. Tunarungu adalah seseorang yang kehilangan kemampuan mendengarnya sedangkan tunawicara adalah seseorang yang kehilangan kemampuan untuk berkomunikasi secara verbal, kedua penyandang disabilitas ini merupakan target pengguna utama dalam aplikasi ini sedangkan pada orang normal dapat juga menggunakan aplikasi ini sebagai media untuk belajar Bahasa Isyarat Indonesia (BISINDO).

Menurut Yunita & Setyati (2019) salah satu kebutuhan perkembangan teknologi saat ini adalah metode interaksi dan komunikasi antara pengguna dan komputer. Untuk mempermudah penggunaan, berbagai inovasi teknologi telah dilakukan untuk menggabungkan berbagai perangkat menjadi satu. Pengguna lebih membutuhkan alat komunikasi yang dapat digunakan dengan alami tanpa perlu kontak langsung dengan perangkat input. Contohnya, gerakan tubuh manusia di depan kamera komputer dapat diinterpretasikan oleh komputer, untuk mengatasi masalah ini, penelitian dilakukan dalam bidang deteksi isyarat tangan, syarat dan gerakan tangan yang diterima oleh kamera dapat digunakan untuk mengontrol pergerakan mouse yang dikenal sebagai “kamera mouse”. Metode yang digunakan dalam penelitian ini adalah algoritma *convex hull* dengan menggunakan algoritma *convex hull* dapat dihitung jumlah jari tangan yang kemudian digunakan sebagai acuan dalam menggerakkan *mouse*. Meskipun penelitian tentang kamera *mouse* sudah banyak dilakukan sebelumnya, namun implementasinya masih memerlukan peralatan tambahan. Penelitian ini bertujuan untuk mengembangkan penelitian yang sudah ada, yaitu pengenalan isyarat tangan dengan implementasi pergerakan mouse secara *real-time* dari video. Dengan menggunakan pengenalan isyarat tangan dan metode *convex hull*, pengenalan tangan akan menjadi lebih mudah hanya dengan menggunakan kamera. Dalam 75 percobaan, penelitian ini mencapai tingkat akurasi sebesar 68%.

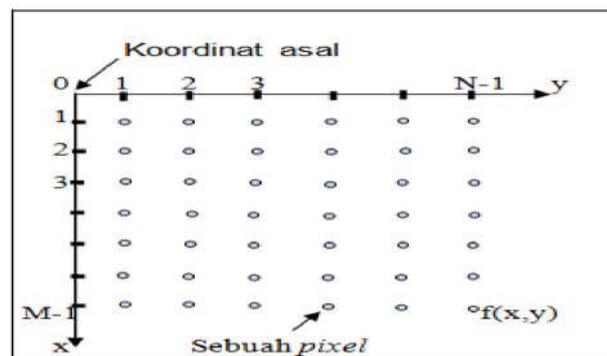
**Table 2.** Perbandingan Metode DS-L dan Metode Sintaks

Pembandingan	Metode DS-L	METODE SINTAKS
Jumlah Variasi Gestur	Lebih banyak	Terbatas
Waktu Eksekusi	Terdapat <i>delay</i> yang lebih lama	Mendekati <i>Real-Time</i>

<b>Penggunaan Memori</b>	Menggunakan teknik <i>Gesture Recording</i> artinya lebih mudah dalam pendefinisian gestur, makin banyak gestur yang direkam maka performa semakin rendah	Pendefinisian tiap gestur memiliki sintaks berbeda namun menghasilkan performa tinggi
<b>Tipe Gestur</b>	Lebih cocok untuk gestur statis	Gestur statis maupun dinamis dapat difasilitasi dengan baik
<b>Popularitas Metode</b>	Jarang dipakai	Lebih banyak dipakai

Referensi untuk perbandingan dari sumber yang menyelidiki atau membandingkan kedua metode tersebut sehingga dapat menjadi pertimbangan saat memilih metode yang sesuai dengan kebutuhan terkait pengenalan gestur.

Menurut (Raharja & Harsadi, 2018) Citra Digital merupakan representasi, turunan atau tiruan dari suatu objek, citra dapat berupa gambar, baik dalam bentuk foto, sinyal video analog seperti yang ada pada monitor televisi atau dalam bentuk digital yang dapat disimpan langsung pada media penyimpanan. Citra digital adalah jenis citra yang dapat diolah oleh komputer, citra merupakan representasi dua dimensi dari citra analog, yang melalui proses sampling, diubah menjadi citra diskrit dengan M baris dan N kolom. Dalam bidang dua dimensi, koordinat citra digital terkait dengan sumbu (x,y).



**Gambar 5.** Koordinat Citra Digital  
(Sumber: Raharja & Harsadi, 2018)

Citra digital secara matematis terlihat seperti dibawah ini dimana x menunjukkan baris dan y menunjukkan kolom:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \dots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Gambar 6. Fungsi Matematis Citra Digital

(Sumber: Raharja & Harsadi, 2018)

Seperti pada layar monitor, koordinat citra dimulai dari pojok kiri atas. Secara matematis dimulai dari (0,0) dan berakhir di (M-1,N-1). Namun secara implementasi di Matlab, koordinat citra dimulai dari (1,1) dan berakhir di (M,N). perlu diingat pula bahwa untuk mengakses piksel citra, penulisan indeks secara matematis pada citra bersesuaian juga dengan penulisan indeks pada Matlab. Artinya  $f(x,y)$  secara matematis serupa dengan  $f(\text{baris}, \text{kolom})$  di Matlab (Priyanto, 2017).

Fungsi  $f(x,y)$  dapat dipisahkan menjadi dua komponen, yaitu:

1.  $i(x,y)$  adalah jumlah cahaya yang berasal dari sumbernya (*illumination*).
2.  $r(x,y)$  adalah derajat kemampuan objek memantulkan cahaya (*reflecton*).

Penelitian ini dilakukan di kampus Universitas Islam Nusantara. Penelitian ini dilaksanakan selama 3 bulan dimulai pada bulan Maret sampai dengan Mei 2023.

## HASIL DAN PEMBAHASAN

### Hasil

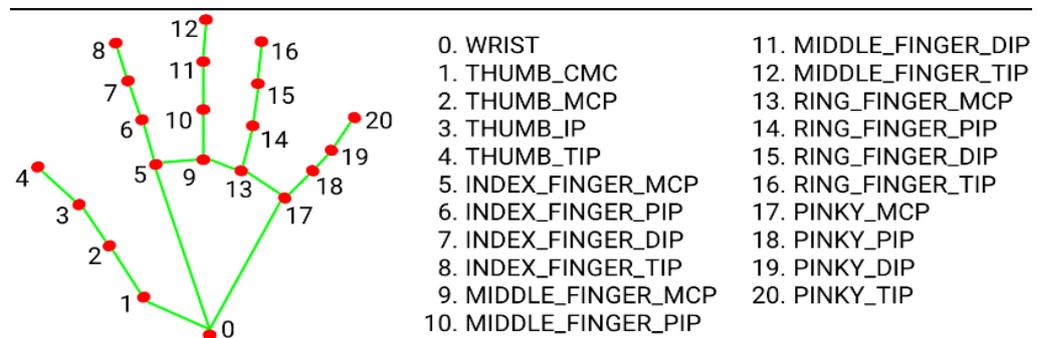
Pada proses pengambilan data pada penelitian ini menggunakan metode *realtime keypoint classifie*, artinya proses pengambilan data dapat langsung melalui aplikasi dan langsung menggunakan gestur tangan dengan bantuan *hand landmark* untuk menentukan titik koordinat pada jari. Dengan menekan tombol angka “1” pada *keyboard* akan mengaktifkan mode *keylogging* dan menekan tombol huruf yang akan dipelajari oleh mesin untuk menyimpan titik koordinat ke file csv.



**Gambar 7.** Contoh pengambilan Koordinat Untuk Dipelajari Mesin

Pada gambar diatas merupakan contoh pengambilan data dengan *keylogging* koordinat pada jari dengan cara menekan tombol angka “1” pada *keyboard*.

Setelah dilakukan proses pengambilan data maka akan dilakukan *preprocessing* data yang akan dilakukan dengan membuat *hand landmark*.



**Gambar 8.** Koordinat *Hand Landmark*

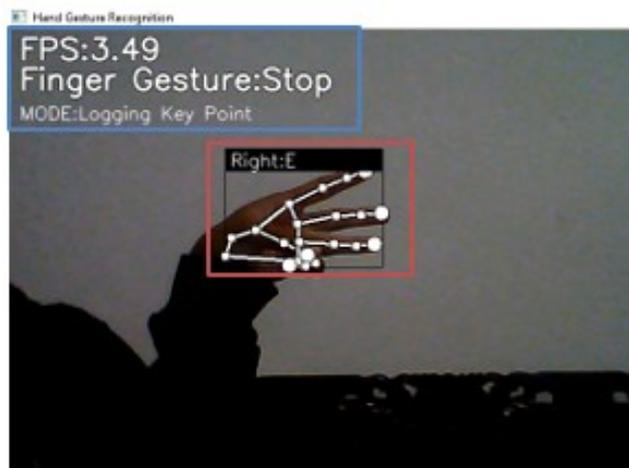
(Sumber: [geeksforgeeks.org](https://www.geeksforgeeks.org))

Titik-titik pada jari tersebut akan diubah menjadi index koordinat dan kemudian akan disimpan dalam bentuk file csv.

```

model > keypoint_classifier > keypoint.csv
1 0,0,0,0,0,-0.2597402597402597,-0.23376623376623376,-0.3961038961038961,-0.5064935064935064,-0.35714285714285715,-0.78571428
2 0,0,0,0,0,0.2631578947368421,-0.029239760081871343,0.5380116959064327,-0.1286549707602339,0.7543859649122807,-0.14619883046
3 0,0,0,0,0,0.26627218934911245,-0.03550295857988166,0.5384615384615384,-0.14201183431952663,0.7633136094674556,-0.1715976331
4 0,0,0,0,0,0.2573099415204678,-0.03508771929824561,0.543859649122807,-0.15204678362573099,0.7719298245614035,-0.187134502923
5 0,0,0,0,0,0.2679738562091503,0.006535947712418301,0.5490196078431373,-0.0457516339869281,0.7712418300653595,-0.039215686274
6 0,0,0,0,0,0.27167630057803466,-0.017341040462427744,0.5540132947976878,-0.10404624277456648,0.7745664739884393,-0.115606036
7 0,0,0,0,0,0.27647058823529413,-0.01764705882352941,0.5647058823529412,-0.1,0.7823529411764706,-0.11176470588235295,0.964705
8 0,0,0,0,0,0.2727272727272727,-0.02597402597402597,0.551948051948052,-0.11038961038961038,0.7662337662337663,-0.13636363636
9 0,0,0,0,0,0.27388535031847133,-0.01910828025477707,0.5477707006369427,-0.10191082802547771,0.7643312101910829,-0.1210191082
10 0,0,0,0,0,0.2682926829268293,0.006097560975609756,0.5548780487804879,-0.0548780487804879,0.7804878048780488,-0.0426829268
11 0,0,0,0,0,0.26993865030674846,-0.006134969325153374,0.558282288588957,-0.0736196319018405,0.7852760736196319,-0.06748466257
12 0,0,0,0,0,0.2710843373493976,0.006024096385542169,0.5602409638554217,-0.06626506024096386,0.7771084337349398,-0.06024096385
13 0,0,0,0,0,0.2621951219512195,0.0,0.5548780487804879,-0.06707317073170732,0.7804878048780488,-0.06707317073170732,0.95731707
14 0,0,0,0,0,0.26380368098159507,0.0,0.5521472392638037,-0.06134969325153374,0.7791411042944786,-0.06134969325153374,0.9570552
15 0,0,0,0,0,0.2709677419354839,0.0,0.0064516129032258064,0.5612903225806452,-0.07741935483870968,0.7935483870967742,-0.077419354
16 0,0,0,0,0,0.2709677419354839,0.0,0.5612903225806452,-0.07096774193548387,0.7935483870967742,-0.07096774193548387,0.97419354
17 0,0,0,0,0,0.2681564245810056,0.0,0.547486033519553,-0.0670391061452514,0.770949720670391,-0.08379888268156424,0.94972067039
18 0,0,0,0,0,0.26256983240223464,0.00558659217877095,0.5418994413407822,-0.0670391061452514,0.770949720670391,-0.0670391061452
19 0,0,0,0,0,0.26285714285714284,-0.005714285714285714,0.5542857142857143,-0.08,0.7828571428571428,-0.08571428571428572,0.9714
20 0,0,0,0,0,0.2696629213483146,0.0,0.5561797752808989,-0.06179775280898876,0.78808988764484944,-0.05056179775280899,0.95505617
21 0,0,0,0,0,0.2742857142857143,-0.005714285714285714,0.5542857142857143,-0.07428571428571429,0.7771428571428571,-0.08,0.96571
22 0,0,0,0,0,0.26857142857142857,-0.005714285714285714,0.56,-0.07428571428571429,0.7828571428571428,-0.07428571428571429,0.965
23 0,0,0,0,0,0.2727272727272727,-0.005681818181818182,0.5568181818181818,-0.07386363636363637,0.7784090909090909,-0.073863636
    
```

Gambar 9. Contoh Titik Koordinat Yang Dibuat Pada Sistem



Gambar 10. Perancangan Antarmuka Aplikasi

Pada gambar diatas merupakan antarmuka dari aplikasi penerjemah Bahasa Isyarat Indonesia (BISINDO). Pada kotak berwarna biru merupakan area untuk menampilkan fps pada komputer, gestur jari dan mode pada penguncian *keypoint*. Pada kotak berwarna merah merupakan area *hand landmark* untuk mengambil titik koordinat tangan yang kemudian akan disimpan sebagai *point* data untuk menentukan *output* yang akan dikeluarkan.

### Pembahasan

Aplikasi pada penelitian ini cukup baik dalam membaca setiap point koordinat yang telah disimpan dan per karakter pada data dapat terbaca, meskipun masih terdapat beberapa karakter yang masih butuh lebih dari satu kali percobaan untuk menghasilkan *output* karakter yang diinginkan. Untuk menguji akurasi sistem

maka dibutuhkan formula *confussion matrix* untuk menguji akurasi,, presisi dan *recall*. *Confussion matrix* dikenal sebagai tabel kontingensi atau *error matrix* merupakan alat umum digunakan untuk mengevaluasi performa model klasifikasi, *confussion matrix* memberikan gambaran tentang berapa banyak instance yang diklasifikasikan dengan benar dan salah oleh model, serta jenis kesalahan yang dilakukan. Rumus *confussion matrix* :

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Penjelasan rumus *confussion matrix* :

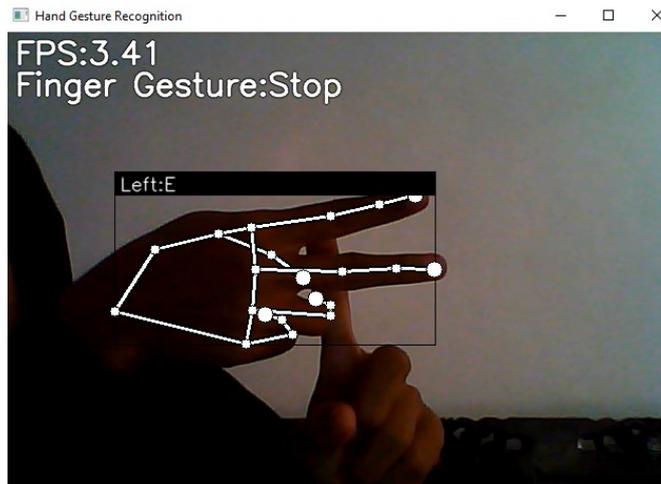
1. True Positive (TP): jumlah instance yang benar diklasifikasikan sebagai positif oleh model
2. False Negative (FN): jumlah instance yang salah diklasifikasikan sebagai negative oleh model
3. False Positive (FP): jumlah instance yang salah diklasifikasikan sebagai positif oleh model
4. True Negative (TN): jumlah instance yang benar diklasifikasikan sebagai negative oleh model

Untuk menghitung *confussion matrix* perlu membandingkan prediksi model dengan label kelas yang sebenarnya. Misalkan memiliki dataset yang terdiri dari N instance dengan dua kelas: positif (P) dan negatif (N). Dengan memasukkan nilai-nilai TP, FN, FP dan TN ke dalam rumus *confussion matrix* dapat membuat tabel kontingensi yang akan memberikan gambaran lengkap tentang performa model klasifikasi, kemudian dapat digunakan untuk menghitung berbagai metrik evaluasi seperti akurasi, presisi, *recall*, F1-Score.

Hasil pengujian sistem menunjukkan hasil akurasi yang sangat baik yaitu sebesar 85% pada jumlah data sebanyak 1000 data dan 1500 data, pada kedua percobaan menggunakan 1000 dan 1500 data memiliki persamaan perolehan akurasi, sedangkan pada percobaan dengan menggunakan 500 data memperoleh tingkat akurasi sebesar 75% pada percobaan tersebut sistem mengalami penurunan

tingkat akurasi sebanyak 10% hal ini disebabkan karena adanya pengurangan jumlah data yang sangat drastis.

Pada pengujian data, jumlah data terbaik yang dapat dipakai aplikasi sebanyak kurang lebih 1000 data karena jika menggunakan data dibawah 1000 akan menurunkan akurasi rata-rata pada sistem pengujian tetapi jika menggunakan data diatas 1000 akurasi tidak bertambah.



Pada gambar diatas dapat dibuktikan bahwa jika gestur jari berubah sedikit saja maka hasil akan berubah maka dari itu dibutuhkan data yang cukup agar sistem dapat membedakan data yang satu dengan data yang lain. Semakin sedikit data maka sistem akan sulit membedakan gestur tangan yang akan dibaca.

**Tabel 3.** Tingkat akurasi dari setiap banyak data

500 data	accuracy	0.75	75
----------	----------	------	----

	macro avg	0.74	0.79	0.72	75
	weighted avg	0.76	0.75	0.71	75
1000 data	accuracy			0.85	3750
	macro avg	0.88	0.83	0.83	3750
	weighted avg	0.87	0.85	0.84	3750
1500 data	accuracy			0.85	3750
	macro avg	0.88	0.83	0.83	3750
	weighted avg	0.87	0.85	0.84	3750

Pengembangan aplikasi penerjemah bahasa isyarat Indonesia menggunakan metode *Long-Short Term Memory* (LSTM) telah membawa banyak manfaat dan solusi dalam memfasilitasi komunikasi antara orang-orang dengan gangguan pendengaran dan penutur bahasa isyarat. LSTM adalah bentuk khusus dari jaringan saraf rekurensi (RNN) yang memiliki kemampuan unik untuk memproses informasi temporal jangka panjang. Dalam konteks aplikasi penerjemah bahasa isyarat Indonesia, metode LSTM digunakan untuk menganalisis pola gerakan tangan, posisi, dan ekspresi wajah dalam bahasa isyarat (Putri et al., 2022). Salah satu keunggulan utama metode LSTM adalah kemampuannya dalam mengatasi masalah-masalah yang sering terjadi dalam penerjemahan bahasa isyarat. Bahasa isyarat memiliki aspek-aspek khusus, seperti aspek spasial, temporal, dan linguistik. LSTM mampu mempelajari dan mengingat pola-pola temporal yang kompleks, yang penting dalam memahami bahasa isyarat yang sangat bergantung pada gerakan dan urutan tangan.

Pengembangan aplikasi penerjemah bahasa isyarat Indonesia menggunakan metode LSTM melibatkan beberapa tahapan. Pertama, data latihan yang terdiri dari rangkaian gerakan tangan dan korespondensi teks terkait dikumpulkan. Kemudian, model LSTM dilatih menggunakan data latihan ini untuk mengenali dan memahami gerakan tangan, ekspresi wajah, dan bahasa tubuh lainnya dalam bahasa isyarat. Selama proses pelatihan, LSTM secara bertahap memperoleh pemahaman yang lebih baik tentang hubungan antara gerakan tangan dan teks terjemahan yang dihasilkan. Salah satu indikator efektivitas metode LSTM dalam pengembangan

aplikasi penerjemah bahasa isyarat Indonesia adalah tingkat akurasi pengenalan gerakan tangan dan teks terjemahan yang dihasilkan. Dengan kemampuannya dalam mempelajari pola-pola temporal yang kompleks, LSTM dapat meningkatkan kualitas dan ketepatan terjemahan bahasa isyarat (Sindarto et al., 2022). Selain itu, kecepatan juga merupakan faktor penting dalam aplikasi penerjemah bahasa isyarat. Metode LSTM telah terbukti mampu menghasilkan hasil terjemahan secara *real-time* dengan waktu respons yang cepat, memungkinkan komunikasi yang lancar antara penutur bahasa isyarat dan non-penutur bahasa isyarat. Keunggulan lainnya dari metode LSTM adalah kemampuannya untuk mengadaptasi dan diperbarui secara berkelanjutan. Model LSTM dapat ditingkatkan dengan menambahkan lebih banyak data latihan dan menerapkan teknik optimisasi yang lebih baik. Hal ini memungkinkan pengembang untuk terus meningkatkan kualitas aplikasi penerjemah bahasa isyarat Indonesia sehingga dapat memberikan pengalaman yang lebih baik bagi pengguna.

Efektivitas metode LSTM dalam pengembangan aplikasi penerjemah bahasa isyarat Indonesia juga sangat tergantung pada kualitas dataset latihan yang digunakan. Dataset yang besar, beragam, dan mencakup berbagai tipe gerakan tangan dan ekspresi wajah dapat meningkatkan kemampuan model LSTM dalam mengenali dan memahami bahasa isyarat. Semakin beragam dataset latihan, semakin baik model LSTM dapat beradaptasi dengan variasi gerakan tangan yang mungkin terjadi dalam bahasa isyarat. Meskipun metode LSTM memiliki keunggulan yang signifikan dalam pengembangan aplikasi penerjemah bahasa isyarat Indonesia, masih ada beberapa tantangan yang perlu diatasi. Salah satunya adalah pengenalan gerakan tangan yang kompleks dan korelasi antara gerakan dan makna di dalam bahasa isyarat. Meskipun demikian, dengan adanya data latihan yang memadai dan model yang terus diperbarui, LSTM dapat mempelajari pola-pola yang kompleks ini dengan lebih baik (Majesty et al., 2023).

Metode LSTM juga dapat dioptimalkan dengan menggabungkannya dengan teknik-teknik lain, seperti pengenalan citra dan pengenalan suara. Dalam pengembangan aplikasi penerjemah bahasa isyarat Indonesia, penggabungan LSTM dengan metode pengenalan citra dapat memperkaya pemahaman model terhadap gerakan tangan dan ekspresi wajah. Dengan demikian, aplikasi dapat lebih

akurat dalam menerjemahkan bahasa isyarat. Secara keseluruhan, pengembangan aplikasi penerjemah bahasa isyarat Indonesia menggunakan metode *Long-Short Term Memory* (LSTM) telah terbukti efektif dalam memfasilitasi komunikasi antara penutur bahasa isyarat dan non-penutur bahasa isyarat. Dengan kemampuannya yang unggul dalam mempelajari pola temporal dan mengenali gerakan tangan yang kompleks, metode LSTM memberikan harapan besar untuk meningkatkan aksesibilitas dan inklusi bagi komunitas tunarungu. Dalam upaya meningkatkan efektivitas aplikasi penerjemah bahasa isyarat Indonesia, terus dilakukan penelitian dan pengembangan lebih lanjut untuk memperbaiki dan mengoptimalkan metode LSTM serta integrasi dengan teknologi lainnya.

### **SIMPULAN**

Kesimpulan yang dapat diambil dari penelitian ini adalah pada beberapa percobaan pengujian terdapat beberapa perbedaan data yang dipakai. Dengan memakai data 500 data mendapatkan akurasi sebesar 75% dan dengan menggunakan 1000 dan 1500 data mendapatkan akurasi sebesar 85% artinya disini pengujian dengan data diatas 1000 tidak dapat meningkatkan akurasi karena kuantitas data bukan merupakan satu-satunya point yang dapat memperbesar angka akurasi, terdapat beberapa faktor yang dapat mempengaruhi akurasi diantaranya kualitas data saat proses pengambilan koordinat oleh mesin *learning* dan kecocokan gestur jari dari saat proses pengolahan data oleh mesin dan saat akan ditampilkan kembali untuk ditampilkan dalam bentuk *output*.

### **DAFTAR RUJUKAN**

Bakti, M. B. S., & Pranoto, Y. M. (2019). Pengenalan Angka Sistem Isyarat Bahasa Indonesia

Dengan Menggunakan Metode Convolutional Neural Network. *Seminar Nasional Inovasi Teknologi.*, 11-16.

Darmatasia, D. (2021.). Pengenalan Sistem Isyarat Bahasa Indonesia (Sibi) Menggunakan Gradient-Convolutional Neural Network. *Jurnal INSTEK (Informatika Sains Dan Teknologi)*, <https://doi.org/10.24252/instek.v6i1.18637>., 6(1), 56.

- Hakkun, R. Y., Zikky, M., & Rafsanjani, B. (2022). Game Edukasi Pengenalan Sistem Isyarat Bahasa Indonesia (Sibi) Menggunakan Myo Armband Pada Arsitektur Client-Server. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, 9(3). <https://doi.org/10.507-512>.
- Haliza, N., Kuntarto, E., & Kusmana, A. (2020). Pemerolehan bahasa anak berkebutuhan khusus (tunarungu) dalam memahami bahasa. *Jurnal Genre (Bahasa, Sastra, Dan Pembelajarannya)*, *2* (1), 5-11. <https://doi.org/10.26555/jg.v2i1.2051>.
- Hikmatia A.E, N., & Ihsan Zul, M. (2021). Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow. *Jurnal Komputer Terapan*, <https://doi.org/10.35143/jkt.v7i1.4629>, 7(Vol. 7 No. 1), 74–83.
- Holawati, M., Auliasari, K., & Ariwibisono, F. (2020). Pengembangan Aplikasi Pengenalan Bahasa Isyarat Abjad Sibi Menggunakan Metode Convolutional Neural Network (Cnn). *JATI (Jurnal Mahasiswa Teknik Informatika)*. <https://doi.org/10.36040/ja>, 6(1), 134–144.
- K. R. Srinath. (2017). Python -The Fastest Growing Programming Language. *International Research Journal of Engineering and Technology*. [www.irjet.net](http://www.irjet.net), 4(12), 354-357.
- Majesty, D., Awangga, R. M., & Fauzan, M. N. (2023). *Voice Cloning: Membuat Sendiri Suara Artifisial Menggunakan Metode Sequence to Sequence Speech Synthesis*. Penerbit Buku Pedia.
- Mardiana, A., & Wahyuni, T. (2019). Rancang Bangun Aplikasi Android Pengenalan Kosakata untuk Disabilitas Tunarungu menggunakan Metode Sistem Isyarat Bahasa Indonesia. *Infotech Journal.*, 5(1), 64–68.
- Putri, H. M., Fadlisyah, F., & Fuadi, W. (2022). Pendeteksian Bahasa Isyarat Indonesia Secara Real-Time Menggunakan Long Short-Term Memory (LSTM). *Jurnal Teknologi Terapan and Sains 4.0*, 3(1), 663–675.

- Prasmatio, R. M., Rahmat, B., & Yuniar, I. (2020). *DETEKSI DAN PENGENALAN IKAN MENGGUNAKAN*. , 1(2), 510–521.
- Raharja, B. D. (2018). Implementasi Kompresi Citra Digital Dengan Mengatur Kualitas Citra Digital. *Jurnal Ilmiah SINUS*, 16(2), 71–77. <https://doi.org/10.30646/sinus.v16i2.363>, 71–77. .
- Shalahudin, A., Iswahyudi, C., Studi, P., Informatika, T., & Industri, F. T. (2019). 1 , 2 , 3. 7(1),. 1-8.
- Sholawati, M., Auliasari, K., & Ariwibisono, F. (2022). Pengembangan Aplikasi Pengenalan Bahasa Isyarat Abjad Sibi Menggunakan Metode Convolutional Neural Network (Cnn). *JATI (Jurnal Mahasiswa Teknik Informatika)*. <https://doi.org/10.36040/j>, 6(1), 134–144. .
- Sindarto, S. S., Ratnawati, D. E., & Arwani, I. (2022). Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI) dengan Metode Convolutional Neural Network pada Perangkat Lunak berbasis Android. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(5), 2129–2138.
- Winaldi, I., & Setyawan, A. (2018). Aplikasi Pengenalan Bahasa Isyarat Untuk Penyandang Tuna Tunggu Berbasis Android (Studi Kasus : SLB Madina Serang). *JSiI (Jurnal Sistem Informasi)*, <https://doi.org/10.30656/jsii.v5i2.779>, 5(2), 70–73.
- Yunita, H., & Setyati, E. (2019). Hand Gesture Recognition Sebagai Pengganti Mouse Komputer Menggunakan Kamera. *Jurnal ELTIKOM*, <https://doi.org/10.31961/eltikom.v3i2.114>, 3(2), 64–76.
- Yunus, M., & Anwar, Y. (2022). *Aplikasi Penerjemah Bahasa Isyarat Indonesia Ke Dalam Huruf Abjad.*, 2(1), 257-262.
- Zulpicha, E. (2018). Konflik Kebijakan Penggunaan Sistem Bahasa Isyarat Indonesia Di Lingkungan Pendidikan Formal. *Jurnal Analisa Sosiologi*. <https://doi.org/10.20961/jas.v6i1.18190>, 6(1).