

PENDETEKSIAN BAHASA ISYARAT INDONESIA SECARA *REAL-TIME* MENGGUNAKAN *LONG SHORT-TERM MEMORY (LSTM)*

Husna Moetia Putri ¹, Fadlisyah ², Wahyu Fuadi ³
^{1,2,3}Teknik Informatika, Fakultas Teknik Universitas Malikussaleh

Email: husna.170170029@mhs.unimal.ac.id

Abstrak

Bahasa Isyarat Indonesia (BISINDO) adalah salah satu cara teman Tuli untuk berkomunikasi. BISINDO muncul secara alami dari interaksi Tuli dengan lingkungannya dan dikenal sebagai budaya Tuli di Indonesia. Namun saat ini terdapat kendala dalam berkomunikasi antar teman Tuli dengan teman dengar dalam menggunakan fasilitas publik dikarenakan petugas pada pelayanan publik tersebut tidak dapat mengerti apa yang disampaikan oleh teman Tuli. Penelitian ini bertujuan untuk menjadi alat penghubung komunikasi satu arah antar teman Tuli dengan teman dengar yang diharapkan dapat mempermudah dalam berkomunikasi. Sistem yang dihasilkan akan mengklasifikasi dan mendeteksi gestur dari kosakata isyarat BISINDO secara langsung yang dikonversi menjadi sebuah teks. Klasifikasi BISINDO pada penelitian ini menggunakan metode *Long short-term memory (LSTM)* dan *Mediapipe Holistic* untuk mendeteksi kerangka pada tangan, wajah dan badan. Objek yang digunakan pada penelitian ini merupakan 30 kosakata isyarat BISINDO yang sering digunakan teman Tuli. Dari hasil evaluasi deteksi *real-time* penelitian ini mendapatkan akurasi sebanyak 92% untuk model 10 kelas dengan *bidirectional layer LSTM*, epoch 1000, *hidden layer* 64, batch size 32 dan mendapatkan akurasi sebanyak 65% untuk model 30 kelas dengan 2 *layer LSTM* epoch 500, *hidden layer* 64, batch size 64.

Kata kunci: *Bahasa Isyarat Indonesia, Klasifikasi, Long short-term memory, Mediapipe Holistic, Real-Time*

1. PENDAHULUAN

Kesulitan dalam berkomunikasi menjadi kendala bagi teman Tuli yang tidak dapat mempelajari bahasa dalam bentuk verbal atau memperoleh kemampuan berbicara secara normal. Salah satu cara mereka untuk berkomunikasi adalah menggunakan bahasa isyarat. Di Indonesia terdapat dua bahasa isyarat yaitu Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia

(BISINDO). SIBI dianggap bukan bahasa isyarat melainkan sebagai sistem dari suatu bahasa, sedangkan BISINDO muncul secara alami dari interaksi Tuli dengan lingkungannya sejak kecil dan dikenal sebagai budaya Tuli di Indonesia.

Bahasa sangat berperan dalam pembentukan identitas dan kedudukan sosial dalam masyarakat. Pada hal ini terdapat kendala antara teman Tuli dengan teman dengar untuk berkomunikasi, seperti halnya pada saat teman Tuli menggunakan fasilitas publik seperti rumah sakit, kepolisian dan sebagainya. Petugas pelayanan publik sangat susah untuk dapat mengerti apa yang disampaikan oleh teman Tuli dikarenakan kurangnya pengetahuan terhadap bahasa isyarat.

Dari uraian diatas penelitian ini memanfaatkan teknologi untuk membuat model komunikasi satu arah antar teman Tuli dengan teman dengar yang mendeteksi gestur pada BISINDO lalu dikonversi menjadi sebuah teks.

Pada penelitian (A.E. & Zul, 2021) melakukan klasifikasi 30 BISINDO menggunakan metode *Convolutional Neural Network* (CNN) dan *mobilenetV2* mendapatkan akurasi sebesar 54,8%. Namun peneliti menyimpulkan bahwa model yang digunakan belum optimal dalam mengklasifikasi dan menyarankan penelitian selanjutnya untuk menggunakan *Long short-term memory* (LSTM) agar dapat mendeteksi isyarat yang bersifat dinamis. Penelitian lain dilakukan oleh (Ghosh, 2021) untuk mendeteksi bahasa isyarat. Penelitian tersebut mendeteksi 5 kata dari *American Sign Language* (ASL) dengan menggunakan mediapipe dan metode LSTM dan mendapatkan hasil akurasi yang cukup akurat.

Berdasarkan penelitian sebelumnya, penulis menggunakan mediapipe holistic dan metode LSTM untuk mendeteksi dan klasifikasi gestur pada BISINDO dalam penelitian ini.

2. KAJIAN TEORITIS

2.1 Deteksi gerakan

Pada dasarnya deteksi gerakan akan mendeteksi citra video yang merupakan kumpulan citra-citra yang direkam secara *frame by frame* yang membentuk pergerakan dinamis. Perubahan dan pergerakan tersebut menjadi suatu

tantangan untuk dapat mengimplementasikan algoritma yang tepat dalam situasi yang cepat dan dinamis (Safwandi, 2015).

2.2 Long short-term memory (LSTM)

Long short-term memory (LSTM) merupakan pengembangan dari arsitektur *Recurrent Neural Network* (RNN). LSTM memiliki keunggulan dalam mengingat dan menyimpan informasi masa lampau serta mampu mempelajari suatu data yang bersifat sekuensial (Sadli et al., 2020). Terdapat struktur dasar pada LSTM yaitu input *layer*, hidden *layer*, dan output *layer*.

Terdapat dua fungsi aktivasi yang digunakan pada LSTM yaitu sigmoid dan tanh. Pada LSTM juga terdapat memory cell dan gerbang. Gerbang tersebut tersusun dari tiga gerbang yaitu *forget gate*, *input gate*, dan *output gate*. Persamaan metode LSTM dapat dilihat dibawah ini :

$$f_t = \sigma (w_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma (w_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$C_t \sim = \tanh (w_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * C_t \sim \quad (4)$$

$$o_t = \sigma (w_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh (C_t) \quad (6)$$

Dimana matriks w merupakan bobot, b merupakan nilai bias, dan f_t , i_t , $C_t \sim$, o_t , h_t merupakan keluaran dari forget gate, input gate, cell state, output gate, dan nilai output pada waktu (t).

2.3 MediaPipe

MediaPipe adalah kerangka kerja yang memungkinkan pengembang untuk membangun saluran ML multi-modal (video, audio, seri waktu apa pun). Sebagai kerangka node dan tepi atau landmark, mereka melacak titik-titik kunci di berbagai bagian tubuh. Semua titik koordinat dinormalisasi tiga dimensi. MediaPipe Holistic menggunakan model landmark pose, wajah dan tangan masing-masing untuk menghasilkan total 543 landmark (33 landmark pose, 468 landmark wajah, dan 21 landmark tangan per tangan).

2.4 Open CV

OpenCV (*Open Source Computer Vision Library*) adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara real-time.

2.5 Tensorflow Object Detection

Tensorflow merupakan kerangka dasar yang digunakan dalam proses machine learning dan sebagai *library* khusus untuk machine learning yang dikembangkan oleh google.

3. METODE PENELITIAN

3.1 Pengumpulan data

Data diperoleh dengan melakukan pengumpulan data dalam wujud data primer dan data sekunder. Gestur kosakata BISINDO dikumpulkan sebanyak 30 gestur berdasarkan kosakata isyarat yang digunakan sehari-hari.

3.2 Pengolahan data

Pengolahan data pada penelitian ini melalui beberapa proses yaitu:

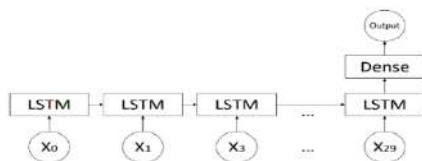
1. Mendeteksi landmarks tangan, wajah, dan badan.
Kamera webcam akan merekam tangan, wajah, dan badan menggunakan open cv dan mediapipe holistic untuk mendapatkan model landmark dan menempatkan 21 *keypoints* pada tangan, 468 *keypoints* pada wajah, dan 33 *keypoints* pada badan.
2. Ekstraksi *keypoints*
Keypoints yang telah diperoleh sebelumnya akan diekstraksi dengan menggabungkan nilai *keypoints* tersebut ke dalam array numpy.
3. Pembuatan folder
Folder ini berguna sebagai tempat penyimpanan data kosakata isyarat yang telah direkam oleh webcam dan telah dideteksi menggunakan *keypoints* mediapipe holistic.
4. Mengumpulkan *keypoints* pada gestur kosakata BISINDO
Pengumpulan *keypoints* dilakukan dengan menggunakan webcam dan merekam *keypoints* dari setiap gerakan kosakata pada BISINDO. Setiap kosakata memiliki 100 video sequence dan setiap sequences akan merekam sebanyak 30 frame. *Keypoints* yang telah direkam akan tersimpan otomatis kedalam folder yang sudah dibuat sebelumnya.

3.3 Preprocess data

Pada tahap ini setiap kelas kosakata BISINDO akan dibuatkan ke dalam label array, kemudian data akan dibagi sebanyak 95% sebagai data *training* dan sebanyak 5% sebagai data *testing* menggunakan *library scikit learn*.

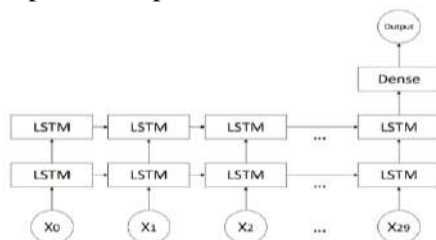
3.4 Pemodelan dan pelatihan

Penelitian ini menggunakan 3 jenis arsitektur LSTM yaitu 1 *layer* LSTM, 2 *layer* LSTM, dan Bidirectional LSTM seperti yang telah dilakukan pada penelitian (Halim & Rakun, 2018). Hal ini dilakukan untuk menentukan model mana yang paling efektif. Berikut desain arsitektur model LSTM dibawah ini.



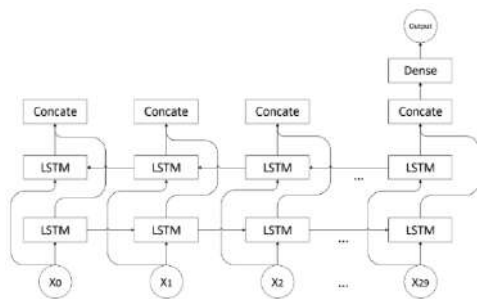
Gambar 1 Arsitektur model LSTM 1 layer

1 lapisan LSTM merupakan jaringan dasar yang hanya memiliki satu lapisan LSTM antara lapisan input dan output.



Gambar 2 Arsitektur model LSTM 2 layer

Perbedaan 2 lapisan LSTM dengan 1 lapisan LSTM terletak pada jumlah lapisan LSTM yang digunakan. Terdapat penambahan satu lapisan LSTM sehingga total menjadi 2 lapisan diantara input dan output. Setiap unit LSTM akan menghitung operasi untuk setiap input dan mentransfer output ke unit LSTM berikutnya.



Gambar 3 Arsitektur model Bidirectional LSTM

Bidirectional memiliki 2 lapisan LSTM, dimana lapisan pertama akan berjalan dari input awal sampai akhir dan lapisan kedua akan berjalan ke arah yang berlawanan dari lapisan pertama. Kemudian setiap output yang dihasilkan dari lapisan LSTM akan digabungkan menggunakan fungsi *concanete*.

Selanjutnya data akan dilatih dengan beberapa parameter yaitu jumlah batch size dengan nilai 32, 64, 128. Jumlah hidden *layer* dengan nilai 32, 64, 128, 256. Dan jumlah epoch dengan nilai 100, 500, dan 1000.

3.5 Evaluasi

Model yang telah dilatih akan di evaluasi menggunakan confusion matrix untuk mengukur performa model dalam kemampuan klasifikasi dengan melihat parameter pengukuran performanya seperti akurasi, recall, presisi, dan skor f1 dengan menggunakan fungsi-fungsi pengukuran yang sudah disediakan oleh *library sklearn*.

4. HASIL DAN PEMBAHASAN

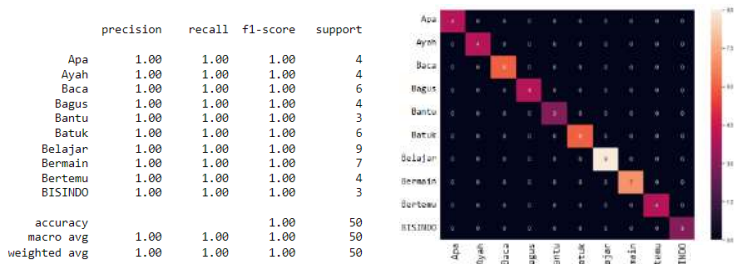
4.1 Pengujian

Setelah melakukan proses pelatihan, model yang sudah dilatih kemudian di evaluasi menggunakan confusion matrix untuk mengukur kinerja yang dilakukan pada data *testing*. Berikut hasil akurasi dari evaluasi.

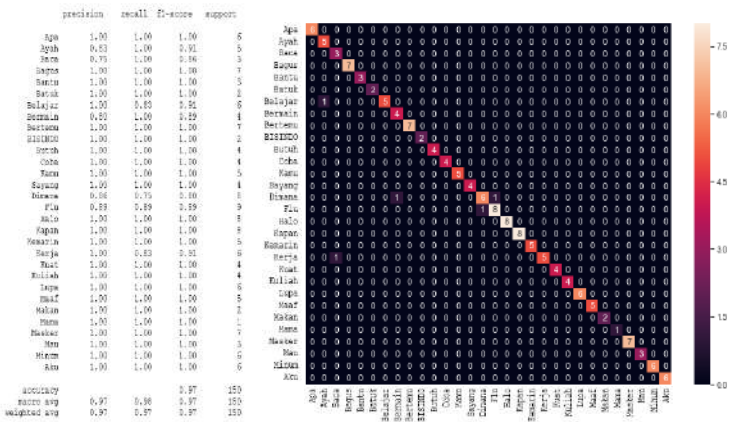
Tabel 1 Hasil akurasi evaluasi

Kelas	Model LSTM	Epoch	Hidden Layer											
			32			64			128			256		
			Batch Size			Batch Size			Batch Size			Batch Size		
			32	64	128	32	64	128	32	64	128	32	64	128
10 Kelas	1 Layer LSTM	100	0.98	0.68	0.06	0.84	0.84	0.78	0.94	0.76	0.98	0.94	0.62	0.92
		500	0.08	0.18	0.58	0.96	0.96	0.06	0.62	0.92	0.84	0.36	0.98	0.08
		1000	0.46	0.94	0.64	1	0.96	0.96	0.98	0.92	0.76	0.98	0.9	0.92
	2 Layer LSTM	100	0.84	0.06	0.68	0.96	0.66	0.88	0.82	0.86	0.9	0.08	0.08	0.64
		500	0.98	0.12	0.06	0.08	0.52	0.12	0.08	0.38	0.08	0.32	0.08	0.08
		1000	0.08	0.22	0.9	0.08	0.8	0.4	0.08	0.08	0.26	0.58	0.1	0.08

30 Kelas	BiLSTM	100	0.88	0.74	0.24	0.92	0.9	0.7	0.08	0.08	0.96	0.08	0.22	0.08
		500	0.94	0.74	1	0.78	0.72	0.76	1	0.9	0.96	0.08	0.08	0.08
		1000	0.74	1	0.66	1	0.88	0.68	0.78	0.9	0.88	0.08	0.08	0.3
	1 Layer LSTM	100	0.006	0.12	0.28	0.88	0.793	0.78	0.66	0.88	0.64	0.88	0.846	0.8
		500	0.84	0.006	0.573	0.493	0.006	0.83	0.04	0.04	0.806	0.91	0.94	0.14
		1000	0.006	0.56	0.006	0.366	0.853	0.186	0.64	0.94	0.806	0.726	0.813	0.753
	2 Layer LSTM	100	0.05	0.89	0.81	0.29	0.95	0.96	0.33	0.85	0.39	0.04	0.04	0.7
		500	0.04	0.04	0.04	0.84	0.97	0.89	0.04	0.04	0.04	0.01	0.21	0.02
		1000	0.18	0.92	0.03	0.04	0.04	0.9	0.04	0.04	0.04	0.01	0.04	0.04
BiLSTM	100	0.28	0.71	0.87	0.85	0.54	0.54	0.04	0.8	0.69	0.07	0.04	0.45	
	500	0.7	0.75	0.27	0.83	0.81	0.77	0.89	0.04	0.35	0.53	0.04	0.76	
	1000	0.57	0.75	0.04	0.84	0.81	0.57	0.87	0.96	0.79	0.04	0.04	0.28	



Gambar 4 Hasil evaluasi dengan akurasi tertinggi pada model yang menggunakan 10 kelas



Gambar 5 Hasil evaluasi dengan akurasi tertinggi pada model yang menggunakan 30 kelas

Berdasarkan hasil akurasi dari evaluasi diatas, model yang menggunakan 10 kelas mendapatkan akurasi tertinggi sebanyak 100% dengan model bidirectional *layer* LSTM, *epoch* 1000, *hidden layer* 64, dan *batch size* 32 pada gambar 4.

Sedangkan untuk model yang menggunakan 30 kelas mendapatkan akurasi tertinggi sebanyak 97% dengan model LSTM 2 *layer* dengan *Epoch* 500, *hidden layer* 64, dan *Batch Size* 64 pada gambar 5.

4.2 Hasil deteksi

Berikut merupakan hasil deteksi model yang telah melewati proses training dan testing. namun dikarenakan jumlah hasil deteksi model yang banyak, maka penulis hanya menampilkan ilustrasi model yang menghasilkan akurasi terbaik pada kelas yang digunakan pada penelitian ini. Untuk mengetahui keakuratan hasil deteksi *real-time* pada penelitian ini penulis melakukan pendeteksian sebanyak 5 kali percobaan.




Tabel 2 Hasil deteksi real-time pada model 10 kelas dengan bidirectional LSTM epoch 1000, *hidden layer* 64, *batch size* 32

Test	Kosakata Isyarat									
	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Belajar	Bermain	Bertemu	BISINDO
1	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Belajar	Bermain	Bertemu	BISINDO
2	Apa	Batuk	Baca	Bagus	Bantu	Batuk	Belajar	Bermain	Bertemu	Bertemu
3	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Ayah	Bermain	Bertemu	BISINDO
4	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Belajar	Baca	Bertemu	BISINDO
5	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Belajar	Bermain	Bertemu	BISINDO

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Kosakata isyarat yang berhasil terdeteksi}}{\text{Total keseluruhan percobaan pendeteksian}} \times 100\% \\
 &= \frac{46}{50} \times 100\% \\
 &= 0.92
 \end{aligned}$$

Pada tabel 2 diatas menunjukkan hasil deteksi *real-time* yang dilakukan penulis sebanyak 5 kali percobaan. Keberhasilan dan kegagalan dalam mengklasifikasi gestur kosakata isyarat dipengaruhi oleh beberapa faktor yaitu gestur kosakata isyarat yang dideteksi memiliki kesamaan dengan gestur kosakata isyarat yang lain dan cahaya yang kurang merata. Akurasi yang didapatkan setelah melakukan 5 kali percobaan mendapatkan nilai sebanyak 92%. Berikut beberapa hasil deteksi kosakata isyarat pada percobaan ke-5.

Tabel 3 Rincian hasil deteksi real-time model 10 kelas dengan bidirectional LSTM epoch 1000, hidden *layer* 64, batch size 32

Gambar	Isyarat	Hasil Deteksi
	Apa	Berhasil mendeteksi kosakata isyarat "Apa"
	Baca	Berhasil mendeteksi kosakata isyarat "Baca"
	Bantu	Berhasil mendeteksi kosakata isyarat "Bantu"

Tabel 4 Hasil deteksi real-time pada model 30 kelas dengan 2 *layer* LSTM epoch 500, hidden *layer* 64, batch size 64

Test	Kosakata Isyarat									
	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Belajar	Bermain	Bertemu	BISINDO
1	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Belajar	Bermain	Bertemu	BISINDO

2	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Belajar	Bermain	Bertemu	Bermain
3	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Ayah	Bermain	Bertemu	BISINDO
4	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Belajar	Bermain	Bantu	BISINDO
5	Apa	Ayah	Baca	Bagus	Bantu	Batuk	Belajar	Bermain	Bertemu	BISINDO

Tabel 4 Hasil deteksi real-time pada model 30 kelas dengan 2 layer LSTM epoch 500, hidden layer 64, batch size 64 (Lanjutan)

Test	Kosakata Isyarat									
	Butuh	Coba	Kamu	Sayang	Dimana	Flu	Halo	Kapan	Kemarin	Kerja
1	Butuh	Ayah	Butuh	Bantu	Bermain	Batuk	Bagus	Kapan	Belajar	Bertemu
2	Butuh	Batuk	Ayah	Sayang	Dimana	Ayah	Bagus	Kapan	Kemarin	Bertemu
3	Butuh	Coba	Batuk	Sayang	Dimana	Flu	Halo	Kapan	Kemarin	Bertemu
4	Butuh	Batuk	Batuk	Bantu	Dimana	Ayah	Halo	Batuk	Bagus	Bantu
5	Butuh	Ayah	Batuk	Bantu	Dimana	Batuk	Halo	Kapan	Kemarin	Bertemu

Tabel 4 Hasil deteksi real-time pada model 30 kelas dengan 2 layer LSTM epoch 500, hidden layer 64, batch size 64 (Lanjutan)

Test	Kosakata Isyarat									
	Kuat	Kuliah	Lupa	Maaf	Makan	Mama	Masker	Mau	Minum	Aku
1	Ayah	Belajar	Belajar	Ayah	Batuk	Ayah	Masker	Mau	Minum	Aku
2	Ayah	Belajar	Belajar	Maaf	Batuk	Ayah	Masker	Mau	Bagus	Aku
3	Kuat	Kuliah	Lupa	Maaf	Makan	Mama	Masker	Mau	Minum	Aku
4	Bagus	Belajar	Belajar	Butuh	Batuk	Ayah	Masker	Mau	Minum	Aku
5	Halo	Belajar	Belajar	Maaf	Makan	Ayah	Masker	Mau	Minum	Aku

$$\text{Akurasi} = \frac{\text{Kosakata isyarat yang berhasil terdeteksi}}{\text{Total keseluruhan percobaan pendeteksian}} \times 100\%$$

$$= \frac{98}{150} \times 100\%$$

$$= 0.65$$

Pada tabel 4 diatas menunjukkan hasil deteksi *real-time* yang dilakukan penulis sebanyak 5 kali percobaan. Keberhasilan dan kegagalan dalam mengklasifikasi gestur kosakata isyarat dipengaruhi oleh beberapa faktor yaitu gestur kosakata isyarat yang dideteksi memiliki kesamaan dengan gestur kosakata isyarat yang lain dan cahaya yang kurang merata. Akurasi yang didapatkan setelah melakukan 5 kali percobaan mendapatkan nilai sebanyak 65%. Berikut beberapa hasil deteksi kosakata isyarat pada percobaan ke-5.

Tabel 5 Rincian hasil deteksi real-time model 30 kelas dengan 2 *layer* LSTM epoch 500, hidden *layer* 64, batch size 64

Gambar	Isyarat	Hasil Deteksi
	Masker	Berhasil mendeteksi kosakata isyarat "Masker"
	Mau	Berhasil mendeteksi kosakata isyarat "Mau"
	Kuliah	Kosakata isyarat "Kuliah" tidak berhasil dideteksi oleh sistem dikarenakan gestur dari kosakata isyarat tersebut hampir mirip dengan kosakata isyarat "Belajar"

Berdasarkan pendeteksian yang dilakukan penulis, terdapat beberapa gestur kosakata isyarat yang sangat sulit untuk dideteksi oleh sistem dan memiliki probabilitas deteksi yang rendah, hal tersebut terjadi dikarenakan gestur dari kosakata isyarat tersebut memiliki kesamaan dengan gestur kosakata isyarat lain, cahaya yang kurang merata dan kemungkinan disebabkan oleh kumpulan data *training* yang kurang baik dalam proses perekaman.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pembahasan, pengklasifikasian gestur kosakata isyarat serta hasil pengujian pada penelitian ini maka dapat diambil kesimpulan sebagai berikut:

1. Model yang mendapatkan hasil terbaik untuk mendeteksi 10 isyarat yaitu model dengan BiLSTM, epoch 1000, hidden *layer* 46, batch size 64 mendapatkan nilai akurasi sebanyak 92% dan model untuk mendeteksi 30 isyarat yaitu model dengan 2 *layer* LSTM epoch 500, hidden *layer* 64, batch size 64 mendapatkan nilai akurasi sebanyak 65% pada pengujian *real-time*.
2. Menggunakan 2 *layer* LSTM dinilai cukup untuk mendeteksi gestur pada kosakata isyarat yang tidak terlalu kompleks.
3. Penggunaan *layer* yang memiliki jumlah lebih banyak kemungkinan mendapatkan hasil akurasi yang lebih baik lagi, akan tetapi lebih sulit untuk dilatih dan memakan waktu yang cukup lama.
4. Klasifikasi gerakan pada hasil deteksi *real-time* terdapat beberapa gerakan yang belum berhasil terdeteksi dengan benar dikarenakan memiliki gerakan yang hampir mirip dengan isyarat yang lain serta pencahayaan yang kurang merata. Hal tersebut dapat membuat mediapipe holistic landmarks pada tangan tidak dapat terdeteksi dengan baik pada tangan, wajah, dan badan objek yang sedang dideteksi.

5.2 Saran

Pada sistem yang telah dibangun tentulah terdapat ketidak sempurnaan, oleh karna itu penulis memiliki beberapa saran guna menyempurnakan penelitian ini di kemudian hari, adapun saran-nya sebagai berikut:

1. Menambahkan data kosakata isyarat yang lebih banyak.
2. Menambahkan variasi latar dan objek yang berbeda pada saat pengumpulan data.
3. Mengembangkan sistem agar dapat menjadi komunikasi dua arah antara teman Tuli dengan teman dengar.
4. Mengembangkan sistem agar dapat menyusun kosakata isyarat yang dideteksi menjadi sebuah kalimat.

DAFTAR PUSTAKA

- A.E., N. H., & Zul, M. I. (2021). Aplikasi Penerjemah Bahasa Isyarat Indonesia Menjadi Suara Berbasis Android Menggunakan Tensorflow. *Jurnal Komputer Terapan*, 7(1), 74–83.
- Ghosh, S. (2021). *Proposal of a Real-time American Sign Language Detector using MediaPipe and Recurrent Neural Network*. 9(7), 46–52.
- Halim, K., & Rakun, E. (2018). *Sign Language System for Bahasa Indonesia (Known as SIBI) Recognizer using Tensorflow and Long short-term memory*.
- Nofiaturrehman, F. (2018). *Problematika anak Tunarungu dan cara mengatasinya*. 6, 1–15.
- Sadli, M., Fajriana, Fuadi, W., Ermatita, & Pahendra, I. (2020). Electrical peak load forecasting using long short term memory and support vector machine. *IOP Conference Series: Materials Science and Engineering*. <https://doi.org/10.1088/1757-899X/725/1/012060>
- Safwandi. (2015). *Analisis Sistem Pendeteksian Warna Kulit dan Wajah Senyum dengan Menggunakan Metode Learning Vector Quantization*.
- Sutrisnadipraj, G., Shesilia K, N., Putri F, S., Yulianto, Y., Handayani, P., & Sembiring, W. P. (2019). Intervensi Psikoedukasi Dalam Mengatasi Stigma Dan Hambatan Komunikasi Pada Teman Tuli Yang Tergabung Dalam Gerkatina Kepemudaan. *Jurnal Bakti Masyarakat Indonesia*, 2(1). <https://doi.org/10.24912/jbmi.v2i1.4348>