

SIBI (Sistem Bahasa Isyarat Indonesia) berbasis *Machine Learning* dan *Computer Vision* untuk Membantu Komunikasi Tuna Rungu dan Tuna Wicara

Saiful Nur Budiman¹, Sri Lestanti², Haris Yuana³, Beta Nurul Awwalin⁴

Universitas Islam Balitar, Indonesia

Article Info

Article History

Received: 05-09-2023

Revises : 11-10-2023

Accepted: 05-12-2023

Keywords

Sign Language (SIBI);

Deep Learning;

Computer Vision;

✉ Corresponding Author

Saiful Nur Budiman,

Universitas Islam Balitar,

Tel. +62 85731644664

sync.saifulnb@gmail.com

ABSTRACT

The Indonesian Sign Language System (SIBI) is used to translate sign language into text or speech. SIBI helps improve communication between people using sign language and those who do not understand it. Unlike commonly used languages, SIBI sign language is less known to most people due to a lack of interest. To address this, an artificial intelligence-based application was developed, focusing on deep learning to recognize SIBI sign language hand movements in real-time. The model was created with 20 epochs, a batch size of 16, and a learning rate of 0.001. It consists of 13 layers, with the ReLU activation function used for the input layer, while the output layer uses Sigmoid. The ADAM optimizer was used to expedite the model creation process. The image dataset used had a size of 300x300 pixels. In the classification testing of the SIBI alphabet results in this study, it was tested using distance tests. The distance between the webcam and the SIBI language speaker was divided into two categories: 40 cm and 60 cm. For a 40-cm distance, an accuracy of 87.50% was obtained, and for a 60-cm distance, an accuracy of 79.17% was achieved. One limitation of this study is that two alphabets, J and Z, were not included in the dataset. This is because recognition of these two alphabets requires not only finger pattern recognition but also recognition of their gesture patterns.

PENDAHULUAN

Dalam konteks Indonesia, bahasa isyarat Indonesia (BISINDO) digunakan sebagai alat komunikasi utama oleh komunitas tuna rungu dan tuna wicara. Namun, karena BISINDO bukan merupakan bahasa resmi negara, maka masih banyak orang yang tidak memahami bahasa tersebut. Sistem Isyarat Bahasa Indonesia (SIBI) dibuat untuk membantu memperbaiki komunikasi antara orang yang menggunakan bahasa isyarat dengan orang yang tidak memahami bahasa tersebut. SIBI merupakan bentuk komunikasi visual yang digunakan oleh komunitas tuna rungu di Indonesia. SIBI memiliki alfabet manual yang terdiri dari gerakan tangan dan jari yang mewakili huruf-huruf dalam alfabet bahasa Indonesia. Bagi banyak individu tuna rungu, SIBI adalah bahasa utama mereka dan mereka mengandalkannya untuk berkomunikasi dengan dunia luar.

Di negara Indonesia peminat dalam mempelajari bahasa isyarat terbilang cukup tinggi. Tetapi minimnya media dalam hal pembelajaran bahasa isyarat di negara Indonesia menjadi masalah yang membuat masyarakat kesulitan untuk mengartikan bahasa isyarat. Kurikulum pendidikan yang digunakan di Indonesia juga tidak mampu memfasilitasi kebutuhan penyandang disabilitas rungu maupun wicara. Untuk memudahkan komunikasi dan

meningkatkan inklusivitas, maka pengembangan teknologi pengenalan bahasa isyarat menggunakan komputer vision sebagai salah satu solusinya. Teknologi tersebut dapat membantu menerjemahkan bahasa isyarat ke dalam teks atau suara untuk memungkinkan komunikasi yang lebih mudah antara orang yang menggunakan bahasa isyarat dengan orang yang tidak mengerti bahasa isyarat.

Penelitian ini menggunakan prediksi untuk mengklasifikasikan bahasa isyarat SIBI menggunakan *deep learning* algoritma *Convolutional Neural Network* (CNN). Tujuan utama dari CNN mempelajari fitur tingkat tinggi dalam data melalui proses konvolusi. CNN cocok untuk pengenalan objek melalui gambar atau suara [1]. CNN mengubah data *input* dari lapisan *input* melalui semua lapisan yang terhubung ke satu set skor *class* oleh lapisan *output*. Lapisan *input* pada penelitian ini menerima masukan citra tiga dimensi dalam bentuk spasial dari ukuran gambar (lebar x tinggi). Selain itu memiliki kedalaman saluran warna dimana pada kasus ini menggunakan saluran warna RGB. Di dalam CNN ekstraksi fitur dan reduksi dimensi dapat dilakukan sekaligus di dalam jaringan sehingga dapat mengurangi *preprocessing* [2].

Penelitian ini didasarkan pada penelitian terdahulu sebagai contohnya pada penelitian yang dilakukan oleh Daroya dkk pada tahun 2018 yang memakai model *Densely Connected Convolutional Neural Network* (DenseNet) yang memperoleh akurasi 90.3%. Namun pada penelitian ini tidak dijelaskan berapa masa *training* yang harus dilakukan dalam proses pembentukan model [3]. Penelitian selanjutnya, dilakukan oleh Darmatasi pada tahun 2021 yang menggunakan Gradient-Convolutional untuk membuat model jaringannya. Dengan adanya *filter gradient* membuat objek memiliki tepi yang lebih tajam sehingga bentuk dari objek terlihat makin jelas. Diperoleh hasil akurasi terbaik 85% untuk pengenalan alfabet SIBI [4]. Pada penelitian selanjutnya dilakukan oleh Suci dkk membuat model Alexnet untuk mengenali bahasa isyarat BISINDO. Nilai akurasi yang diperoleh sekitar 98,3%. Walaupun hasil akurasi bagus, namun citra yang digunakan haruslah menggunakan *background* hijau, sehingga kurang begitu bagus untuk realisasi penggunaannya [5]. Selanjutnya penelitian rujukan dari Vinna dkk, yang menggunakan model ResNet50V2 memiliki hasil *testing* yang sangat bagus yakni 100%. Terdapat kendala untuk beberapa gambar tangan karena memiliki kemiripan bentuk pola seperti huruf B, D, U dan V sehingga akurasinya menurun sekita 98-99% [6]. Mengacu dari penelitian sebelumnya, maka penelitian ini membahas masalah utama pada klasifikasi bahasa isyarat SIBI yaitu pada keterbatasan pembentukan model saat *training*, untuk mengatasi hal tersebut peneliti menggunakan CNN dengan model optimasi *Adaptive Moment Estimator* (ADAM) [7]. Model optimasi digunakan untuk memperbaharui bobot secara iteratif yang didasarkan pada data *training*. Dengan begitu model yang terbentuk dapat digunakan untuk melakukan klasifikasi alfabet SIBI yang akan membantu pembaca memahami SIBI dan menghilangkan hambatan komunikasi dengan saudara kita yang tuna rungu ataupun tuna wicara.

METODE

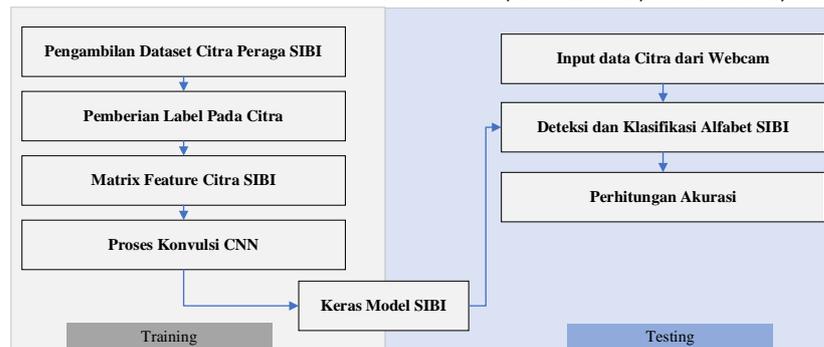
Tahapan Penelitian

a. Pengumpulan Dataset

Pada gambar 1, ditunjukkan langkah-langkah yang dilakukan peneliti untuk menghasilkan klasifikasi alfabet terhadap bahasa isyarat SIBI. Secara garis besar dibagi menjadi dua bagian yakni tahap *training* dan tahap *testing*. Pada bagian *training* digunakan untuk membentuk model dari alfabet SIBI berdasarkan dataset citra pola jari tangan yang dituturkan oleh pakar untuk mengisyaratkan SIBI. Hasil dari proses *training* nantinya digunakan sebagai proses klasifikasi pada bagian *testing*.

SIBI (Sistem Bahasa Isyarat Indonesia) berbasis Machine Learning dan Computer Vision Untuk Membantu Komunikasi Tuna Rungu dan Tuna Wicara

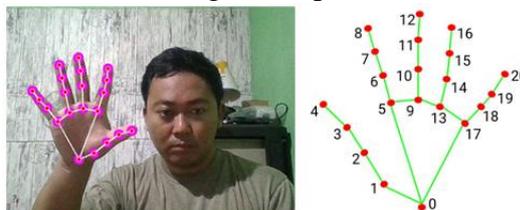
Saiful Nur Budiman, Sri Lestanti, Haris Yuana, Beta Nurul Awwalin



Gambar 1. Diagram Alir Klasifikasi Alfabet SIBI

Sesuai dengan gambar 1, tahapan pada proses *training* pertama kali dengan mengambil dataset citra peraga SIBI. Proses pengambilan dataset untuk SIBI menggunakan *MediaPipe* merupakan langkah kunci dalam pengembangan teknologi pengenalan bahasa isyarat yang dapat membantu orang dengan gangguan pendengaran berkomunikasi dengan lebih efektif. *MediaPipe* merupakan salah satu teknologi yang dapat digunakan untuk melakukan pelacakan gerakan tangan dan tubuh dengan menggunakan kamera. Sehingga memungkinkan peneliti untuk mengumpulkan data mengenai bahasa isyarat yang digunakan oleh para pemakainya.

Hal yang harus dilakukan pada pertama kali yakni pengambilan dataset dengan mengumpulkan video dari para penutur bahasa isyarat Indonesia yang sedang melakukan gerakan tangan yang relevan dengan pembelajaran bahasa isyarat. Video-video tersebut kemudian diolah menggunakan *MediaPipe* untuk mengekstrak informasi mengenai posisi dan gerakan tangan. *MediaPipe* menggunakan jaringan saraf untuk mengidentifikasi daerah yang mungkin berisi tangan manusia [8]. Hasil dari tahap ini adalah sebuah *bounding box* yang menunjukkan posisi perkiraan tangan. Setelah tangan terdeteksi, langkah berikutnya adalah mengestimasi posisi titik-titik penting *hand landmark*. Titik-titik ini mencakup ujung jari, pangkal jari, pergelangan tangan, dan lain-lain [9]. *MediaPipe* menggunakan model regresi untuk memprediksi posisi *hand landmark* berdasarkan gambar yang diberikan. Untuk mengatasi pergerakan tangan yang cepat dan perubahan pose, *MediaPipe* melibatkan teknik pelacakan tangan. Pelacakan tangan memanfaatkan informasi dari *frame* sebelumnya untuk memperbaiki dan memprediksi posisi tangan pada *frame* saat ini. Teknik pelacakan ini membantu menjaga konsistensi dalam pelacakan tangan dan meminimalkan kesalahan yang mungkin terjadi [10]. Pada gambar 2 dicontoh penggunaan *MediaPipe Hand Tracking* secara *real-time* dapat membaca *hand landmark* dengan sempurna.



Gambar 2. Deteksi Tangan dan *Hand Landmark*

Data yang dikumpulkan meliputi citra dari setiap *frame* pola jari tangan pada bahasa isyarat SIBI. Data dikumpulkan dari Guru pengajar di SLB YPL B Kota Blitar yang sudah terbiasa menggunakan bahasa isyarat SIBI untuk kegiatan belajar mengajar di sekolah. Setiap alfabet diambil citranya sebanyak 30 kali pengambilan. Setiap citra memiliki resolusi 300x300 piksel yang diambil dari beberapa sudut pengambilan kamera. Sebagai contoh pada gambar 3 diambil 30 sampel untuk mengenali pola alfabet E.

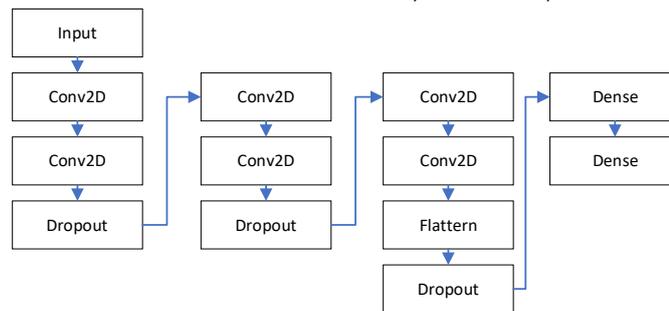


Gambar 3. Pengambilan Dataset

b. Training Model

Tahapan berikutnya sesuai gambar 1 pada proses *training* adalah pemberian label, ekstraksi fitur, dan proses konvolusi CNN. Tahapan tersebut dilakukan menggunakan bantuan *Teachable Machine*. Data tersebut akan dilatih dengan cara mengenali pola gerakan jari tangan yang telah dikumpulkan dari seluruh alfabet bahasa isyarat SIBI. Data pelatihan disimpan pada berkas dengan nama *keras_model.h5* sedangkan data pelabelan alfabet SIBI disimpan dalam *labels.txt* yang mana terdapat 24 *class* yang mewakili alfabet SIBI dari A sampai Z (J dan Z tidak digunakan). Model yang dibuat terdiri dari 13 lapis. Dimana semua lapisan konvolusi ukuran kernel yang diterapkan menjadi 4x4. Jumlah kernel *filter* dilapisan konvolusi diatur ke 64, 128, 256, 512. Untuk fungsi *dropout* diatur ke 0.5 dan menggunakan fungsi optimasi *ADAM*. *Learning rate* diatur ke 0.001, *batch size* 16, *epoch* 20. Lapisan masukan dan lapisan tersembunyi dirancang menggunakan fungsi aktivasi *ReLU*. Sedangkan lapisan keluaran menggunakan fungsi aktivasi Sigmoid. *Featur map* yang dihasilkan dari ekstraksi fitur masih berbentuk *array* multidimensi, sehingga diperlukan lapisan *flatten* untuk *reshape* fitur menjadi sebuah vektor agar dapat digunakan sebagai *input* dari *fully-connected layer* [11].

Fungsi optimizer *ADAM* pada *deep learning* adalah sangat penting dalam proses pelatihan model. Salah satu fungsi utama *ADAM* adalah mengatur laju pembelajaran (*learning rate*) secara adaptif untuk setiap parameter dalam model. Hal ini memungkinkan model untuk belajar dengan cepat pada proses awal. Kemudian menyesuaikan laju pembelajaran secara otomatis seiring dengan berjalannya pelatihan. Hal ini sangat penting karena laju pembelajaran yang terlalu tinggi dapat menyebabkan model melewati optimal *local* dan laju pembelajaran yang terlalu rendah dapat membuat proses pelatihan menjadi lambat. Selain itu, *ADAM* juga menggunakan momentum untuk mengurangi variasi yang terjadi pada proses pembelajaran, sehingga mempercepat konvergensi model [12]. Dengan menggabungkan adaptasi laju pembelajaran dan momentum, *ADAM* dapat secara efisien menemukan minimum global dari fungsi objektif. Hal ini bertujuan untuk pelatihan model *deep learning*. *ADAM* juga memiliki kemampuan untuk menangani masalah *gradien* yang jarang atau tidak terdistribusi dengan baik, yang seringkali terjadi dalam model *deep learning* [13]. Dengan menggunakan estimasi momen pertama dan kedua, *ADAM* dapat mengurangi dampak dari *gradien* yang menyimpang. Sehingga memungkinkan model untuk tetap stabil dan efisien dalam proses pelatihan. Pada gambar 4 ditunjukkan struktur arsitektur dari model yang digunakan.



Gambar 4. Arsitektur Model CNN yang digunakan

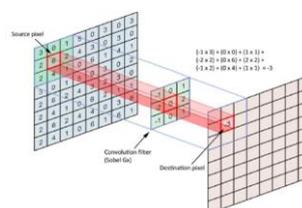
c. *Testing*

Pada tahap testing akan digunakan untuk memperoleh nilai prediksi alfabet SIBI berdasarkan model yang telah dibuat sebelumnya. Uji coba dua kali percobaan terhadap jarak tangkap kamera *webcam* terhadap gerakan pola alfabet SIBI. Hal ini dilakukan untuk mengukur tingkat keakurasian dari model yang telah dibuat. Jarak yang digunakan yakni 40 cm dan 60 cm.

Machine Learning dan Deep Learning

Machine learning cabang dari kecerdasan buatan yang fokus pada pengembangan algoritma dan model komputer yang dapat belajar dari data dan melakukan tugas tertentu tanpa perlu pemrograman eksplisit. Tujuannya adalah untuk mengajarkan komputer untuk mengenali pola, membuat prediksi, dan mengambil keputusan berdasarkan data yang diberikan [14]. *Machine learning* memiliki aplikasi yang luas dalam berbagai industri, termasuk pemrosesan gambar, pemrosesan bahasa alami, otomasi industri, pengenalan suara, pengobatan, keuangan. Hal ini akan terus berkembang dan memungkinkan komputer untuk mengambil keputusan dan tugas yang semakin kompleks.

Deep Learning merupakan bagian dari *Machine Learning* yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data dengan fungsi transformasi non-linear yang berlapis dan mendalam bahkan sampai ratusan lapisan [15]. Salah satu algoritma *Deep Learning* yakni CNN merupakan jenis arsitektur jaringan saraf tiruan yang dirancang khusus untuk tugas-tugas pengolahan gambar dan penglihatan komputer. CNN mengambil inspirasi dari cara manusia memproses visual, seperti pengenalan objek dan fitur dalam gambar. Ringkasan model lapisan konvolusi yang digunakan ditunjukkan pada gambar 5 [16]. Pada proses konvolusi memanfaatkan *filter* layaknya gambar. *Filter* tersebut memiliki ukuran tinggi, lebar dan kedalaman. Pada penelitian ini kedalaman merupakan saluran RGB. Nilai *filter* inilah nantinya yang menjadi parameter yang akan di-*update* pada proses *learning*.



Gambar 5. Ringkasan Model CNN

HASIL DAN PEMBAHASAN

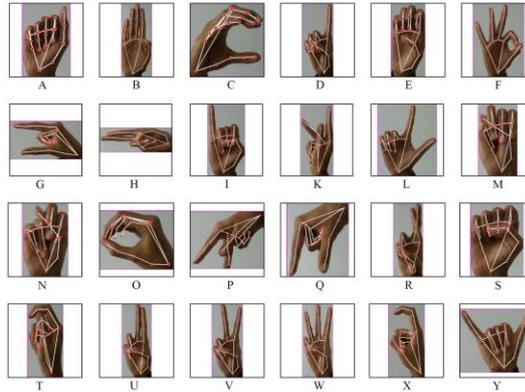
Hasil Penelitian

Sesuai gambar 1, pada penelitian ini terdiri dari dua tahap yakni tahap *training* dan *testing*. Pada tahap *training* prosesnya memasukkan dataset kemudian pembentukan model CNN. Sedangkan pada tahap *testing* prosesnya memasukkan citra secara *real-time* kemudian

dilakukan klasifikasi alfabet SIBI. Berikut dijelaskan tentang bagaimana proses pembangunan model dari penggunaan dataset sampai proses evaluasi model yang telah dibangun.

a. *Loading* Dataset

Pada tahapan pertama dilakukan proses *load* dataset SIBI. Dataset tersebut mewakili 24 kelas alfabet SIBI. Untuk alfabet J dan Z tidak dapat dikenali pada penelitian ini, dikarenakan kedua alfabet tersebut selain menggunakan pola haruslah disertai *gesture*. Pada gambar 6, ditunjukkan citra dataset yang digunakan.



Gambar 6. Citra Dataset SIBI

b. Hasil *Training* Model

Tujuan dari traning model supaya model tersebut dapat digunakan untuk mengenali alfabet SIBI. Proses yang dilakukan dengan membuat model *sequential* dengan parameternya ditunjukkan tabel 1. Berdasarkan dari proses *training* terakhir dengan epoch sebanyak 20 kali diperoleh nilai *loss* 0.0188 dan nilai *accuracy* 0.9972 Untuk setiap proses percobaan yang dilakukan selama pembentukan model ditunjukkan pada tabel 2.

Tabel 1. Model *Sequential*

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 221, 221, 64)	3136
conv2d_1 (Conv2D)	(None, 109, 109, 64)	65600
dropout (Dropout)	(None, 109, 109, 64)	0
conv2d_2 (Conv2D)	(None, 106, 106, 128)	131200
conv2d_3 (Conv2D)	(None, 52, 52, 128)	262272
dropout_1 (Dropout)	(None, 52, 52, 128)	0
conv2d_4 (Conv2D)	(None, 49, 49, 256)	524544
conv2d_5 (Conv2D)	(None, 23, 23, 256)	1048832
flatten (Flattern)	(None, 1354424)	0
dropout_2 (Dropout)	(None, 1354424)	0
dense (Dense)	(None, 512)	69337600
dense_1 (Dense)	(None, 24)	12312

Total params: 71385496 (272.31 MB)

Trainable params: 71385496 (272.31 MB)

Non-trainable params: 0 (0.00 bytes)

Tabel 2. Hasil

Epoch	Loss	Accuracy
1	3.1995	0.2278
2	0.6615	0.8181
3	0.2722	0.9153
4	0.2082	0.9347

5	0.1127	0.9681
6	0.1049	0.9667
7	0.0684	0.9847
8	0.0717	0.9792
9	0.0370	0.9917
10	0.0412	0.9889
11	0.0391	0.9917
12	0.2902	0.9306
13	0.0456	0.9792
14	0.0972	0.9764
15	0.1251	0.9681
16	0.0998	0.9764
17	0.2201	0.9542
18	0.0974	0.9750
19	0.0493	0.9875
20	0.0229	0.9917

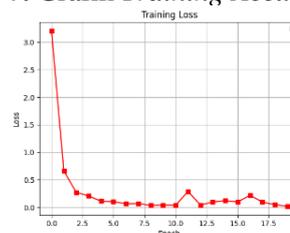
Pembahasan

Pembuatan grafik hasil pelatihan dibuat untuk mengetahui perubahan nilai *loss* dan *accuracy* selama proses pembentukan model. Pada gambar 7 ditunjukkan grafik hasil selama pelatihan dimana nilai akurasi mengalami kenaikan dan penurunan pada setiap *epoch*. Pada epoch kedua mengalami kenaikan nilai 0.8181 dan terus naik nilainya. Terjadi penurunan akurasi namun tidak secara signifikan dan masih berada di rentang nilai 0.9306. Nilai akurasi tertinggi berada pada *epoch* ke 20 dengan nilai akurasi 0.9917 atau mendekati nilai 1. Hal ini dapat diartikan bahwa model yang dibentuk memiliki kinerja yang baik untuk proses prediksi dengan benar.

Dengan meminimalkan nilai *loss* selama proses pelatihan, model yang dibentuk menjadi lebih baik dalam menyesuaikan diri. Dengan data pelatihan dapat memberikan prediksi yang lebih akurat pada data baru. Hal ini dapat ditunjukkan dari hasil grafik yang terbentuk pada gambar 8 mengenai nilai *loss* yang terjadi pada setiap *epoch*. Setelah dijalankan iterasi 20 *epoch*, ditunjukkan bahwa nilai *loss* mengalami penurunan secara signifikan. Pada *epoch* ke 20 nilai *loss* mencapai 0.0229 yang mana dapat diartikan model yang dibentuk selama proses pelatihan sudah optimal sehingga memungkinkan prediksi yang akurat.



Gambar 7. Grafik *Training Accuracy*



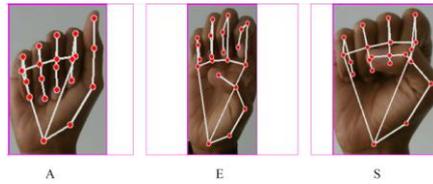
Gambar 8. Grafik *Training Loss*

Berikutnya untuk mengetahui tingkat akurasi prediksi dari aplikasi klasifikasi alfabet SIBI, maka dilakukan pengujian terhadap jarak. Sampai seberapa jauh kamera mampu mengkalsifikasikan alfabet SIBI ini dengan baik. Pengujian dilakukan sebanyak 5 kali percobaan untuk masing-masing alfabet SIBI dengan jarak 40 cm dan 60 cm. Hasil dari pengujian disajikan pada tabel 3. Pengujian dengan jarak 40 cm yang mana menghasilkan nilai akurasi 87.50%. Nilai ini diperoleh dari 24 aalfabet SIBI pada penelitian ini yang tepat diklasifikasikan hanya 21. Untuk klasifikasi alfabet E, S kurang begitu bagus melakukan klasifikasi dikarenakan dataset yang digunakan polanya hampir memiliki kesamaan dengan alfabet A. Hal ini dapat dilihat dari gambar 9 dimana pola dari penulangan jari tangannya mirip yakni menggenggam dengan sedikit menggeser ibu jari ke atas. Sedangkan untuk pola alfabet A, terpisah untuk pola alfabet E, dan menyatu kebawah untuk pola alfabet S.

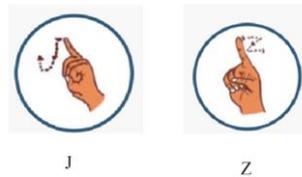
Tabel 3. Hasil Pengujian Akurasi Klasifikasi Alfabet SIBI

Alfabet	Jarak (cm)	Confidence	Klasifikasi	Alfabet	Jarak (cm)	Confidence	Klasifikasi
A	40	0.99998105	A	A	60	0.9998621	A
B	40	0.94997394	B	B	60	0.81166923	B
C	40	0.7598816	C	C	60	0.49721867	C
D	40	0.40131235	D	D	60	0.45833766	D
E	40	0.9997907	A	E	60	0.97981215	A
F	40	0.5078021	F	F	60	0.42877135	F
G	40	0.5047388	G	G	60	0.5040481	G
H	40	0.89980584	H	H	60	0.82828254	H
I	40	0.85537606	I	I	60	0.82183677	I
K	40	0.9930587	A	K	60	0.98801696	A
L	40	0.3698431	L	L	60	0.27383047	L
M	40	0.8552639	M	M	60	0.7190816	M
N	40	0.68306935	N	N	60	0.370212	M
O	40	0.6212528	O	O	60	0.58953756	O
P	40	0.6581041	P	P	60	0.43787125	P
Q	40	0.81235254	Q	Q	60	0.70379996	Q
R	40	0.30723852	R	R	60	0.54051876	B
S	40	0.99985754	A	S	60	0.72600514	A
T	40	0.8636346	T	T	60	0.831537	T
U	40	0.5386022	U	U	60	0.5266826	U
V	40	0.74263257	V	V	60	0.6195681	V
W	40	0.63078666	W	W	60	0.62955797	W
X	40	0.9365203	X	X	60	0.6954069	X
Y	40	0.9068196	Y	Y	60	0.96482897	Y
Akurasi			87.50%	Akurasi			79.17%

Untuk jarak 60 cm hasil akurasi yang diperoleh sekitar 79,17%. Penurunan akurasi ini disebabkan karena gambar pola jari untuk alfabet SIBI yang ditangkap oleh *webcam* semakin jauh, sehingga nilai *confidence* untuk hasil klasifikasi kurang baik. Sebagai contoh untuk klasifikasi alfabet E, K, N, R dan S kurang begitu baik hasil klasifikasinya. Alfabet yang ada pada SIBI di penelitian ini juga masih terbatas yakni hanya 24 dari 26 alfabet SIBI, dimana alfabet J dan Z masih belum dapat dikenali. Khusus untuk kedua alfabet tersebut selain dari pola jari, diperlukan juga *gesture* gerakan jari. Sebagai gambaran pola gerakan untuk memperagakan alfabet J dan Z pada bahasa SIBI ditunjukkan pada gambar 10.



Gambar 9. Perbandingan Alfabet A, E dan S pada SIBI



Gambar 10. Pola dan Gesture Alfabet J dan Z pada Bahasa SIBI

SIMPULAN DAN SARAN

Penelitian ini dikembangkan untuk mengklasifikasikan alfabet SIBI menggunakan *convolutional neural network*. Tujuan dari penelitian ini supaya dapat membaca pola bahasa isyarat SIBI yang dituturkan oleh penyandang tuna rungu dan tuna wicara. Alfabet SIBI terdapat 26, namun pada penelitian ini digunakan hanya 24 alfabet saja. Hal ini dikarenakan huruf J dan Z harus disertai *gesture* gerakan jari. Sedangkan pada penelitian ini hanya dapat membaca pola tangan saja, tidak disertai dengan *gesture*. Model yang digunakan menggunakan algoritma CNN dengan 20 epoch, 16 batch size dan *learning rate* 0.001. Pada lapisan tersembunyi digunakan fungsi aktivasi ReLU sedangkan fungsi aktivasi untuk lapisan keluaran menggunakan Sigmoid. Model yang dibentuk adalah model *sequential* dengan *optimizer* ADAM dengan nilai terakhir pada *loss* 0.0188 dan *accuracy* 0.9972. Pada penelitian ini ADAM digunakan karena dapat mengurangi laju pembelajaran secara adaptif untuk menghindari konvergensi yang lambat. Untuk menguji seberapa baik hasil proses klasifikasinya maka peneliti melakukan pengujian jarak pada aplikasi yang telah dikembangkan. Pada proses pengujian dengan jarak 40 cm menghasilkan akurasi sebesar 87,50%. Sedangkan pada jarak 60 cm dari *webcam* menghasilkan akurasi sebesar 79,17%. Kemampuan *webcam* dalam mengenali pola gerakan jari bahasa SIBI ternyata pada penelitian ini dipengaruhi juga oleh jarak. Kekurangan dari penelitian ini yakni alfabet J dan Z yang tidak mampu dikenali karena harus menggunakan pola gerakan tangan (*gesture*) selain dari pola jari. Akan lebih baik jika aplikasi yang dikembangkan dapat mengenalinya lebih maksimal dengan menggunakan sistem BISINDO yang menggunakan dua tangan serta *gesture*. Selain dari *webcam*, penggunaan Sensor Leap *Motion Controller* dapat digunakan untuk inputan tambahan supaya mendapatkan hasil klasifikasi yang lebih baik lagi.

DAFTAR PUSTAKA

- [1] B. Rahmat and B. Nugroho, *Pemrograman Deep Learning dengan Python*. Indomeida Pustaka, 2021.
- [2] D. Prasetyawan and R. Gatra, "Model Convolutional Neural Network untuk Mengukur Kepuasan Pelanggan Berdasarkan Ekspresi Wajah," *Jurnal Teknik Informatika dan Sistem Informasi (JuTISI)*, vol. 8, 2022.

- [3] R. Daroya, D. Peralta, and P. Naval, "Alphabet Sign Language Image Classification Using Deep Learning," presented at the Conf. Proceedings TENCON, 2018.
- [4] Darmatasia, "Pengenalan Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Gradient-Convolutional Neural Network," *Jurnal Informatika Sains dan Teknologi (INSTEK)*, vol. 6, 2021.
- [5] S. Dwijayanti, Hermawati, S. I. Taqiyah, H. Hikmarika, and B. Y. Suprpto, "Indonesia Sign Language Recognition using Convolutional Neural Network," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, 2021.
- [6] V. R. S. Nastiti, R. A. Muhammad, and B. P. Putra, "Pendeteksi Bahasa Isyarat Gestur Tangan dengan Menggunakan Model CNN ResNet50V2," *Rekayasa Sistem dan Teknologi Informasi (RESTI)*, vol. 6, 2022.
- [7] H. P. A. Tjahyaningtjas, W. Yustanti, and A. Prihanto, "Analisa Learning rate dan Batch size Pada Klasifikasi Covid Menggunakan Deep learning dengan Optimizer ADAM," *Journal Information Engineering and Educational Technology (JIEET)*, vol. 5, 2021.
- [8] F. Zhang *et al.*, "MediaPipe Hands: On-device Real-time Hand Tracking," presented at the CVPR Workshop on Computer Vision for Augmented and Virtual Reality, Seattle, USA, 2020.
- [9] Indriani, M. Harris, and A. S. Agoes, "Applying Hand Gesture Recognition for User Guide Application Using MediaPipe," in *Proceedings of the 2nd International Seminar of Science and Applied Technology (ISSAT 2021)*, 2021, vol. 207.
- [10] M. K. Hossen and M. S. Uddin, "A dataset for Assessing Real-time Attention Levels of the Students During Online Classes," *Data in Brief*, vol. 51, 2023.
- [11] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) pada Ekspresi Manusia," *ALGOR Journal*, vol. 2, 2020.
- [12] M. Reyad, A. M. Sarhan, and M. Arafa, "A Modified ADAM Algorithm for Deep Neural Network Optimization," *Neural Computing and Applications*, vol. 35, pp. 17095-17112, 2023.
- [13] D. Irfan, R. Rosnelly, M. Wahyuni, J. T. Samudra, and A. Rangga, "Perbandingan Optimasi SGD, ADADELTA dan ADAM Dalam Klasifikasi Hydrangea Menggunakan CNN," *Journal of Science and Social Research*, vol. 5, pp. 244-253, 2022.
- [14] F. D. Telaumbanua, P. Hulu, T. Z. Nadeak, R. R. Lumbantong, and A. Dharma, "Penggunaan Machine Learning Di Bidang Kesehatan," *Jurnal Teknologi dan Ilmu Komputer Prima (Jutikomp)*, vol. 2, 2020.
- [15] L. B. Ardianto, M. I. Wahyuddin, and W. Winarsih, "Implementasi Deep Learning untuk Sistem Keamanan Data Pribadi Menggunakan Pengenalan Wajah dengan Metode Eigenface Berbasis Android," *Jurnal Teknologi Informasi dan Komunikasi (JTIK)*, vol. 5, 2021.
- [16] T. Israldi and E. H. S. S. F. Syafria, "Klasifikasi American Sign Language Menggunakan Convolutional Neural Network," *Building of Informatics, Technology and Science (BITS)*, vol. 4, 2022.