

PENERAPAN ALGORITMA *RANDOM FOREST CLASSIFIER* PADA SISTEM DETEKSI SIMBOL SISTEM ISYARAT BAHASA INDONESIA (SIBI)

Prabandalu Enggar Wiraswendro^{1*}, Hari Soetanto²

^{1,2}Fakultas Teknologi Informasi, Teknik Informatika, Universitas Budi Luhur
Email: ¹prabandaluenggar@gmail.com, ²hari.soetanto@budiluhur.ac.id

(Naskah masuk: 12 Agustus 2022, diterima untuk diterbitkan: 1 September 2022)

Abstrak

Bahasa isyarat adalah bahasa yang mengutamakan gerakan tubuh dan mimik muka sebagai simbol untuk saling berkomunikasi. Kelompok penyandang tunarungu dan tunawicara merupakan pengguna utama dari bahasa isyarat. Salah satu jenis bahasa isyarat yang ada di Indonesia adalah SIBI (Sistem Isyarat Bahasa Indonesia). SIBI (Sistem Isyarat Bahasa Indonesia) merupakan bentuk bahasa lisan yang diubah ke dalam bahasa isyarat. Bahasa isyarat tidak hanya digunakan sebagai alat komunikasi dan interaksi antar penyandang tunarungu dan tunawicara, tetapi juga dengan orang normal. Namun, hingga saat ini masih terdapat kesenjangan komunikasi antara penyandang tunarungu dan tunawicara dengan orang normal. Maka dibuat sebuah rancangan sistem yang dapat mendeteksi simbol bahasa isyarat SIBI (Sistem Isyarat Bahasa Indonesia). Sistem ini dibuat dengan menggunakan algoritma *Random Forest Classifier* dengan bantuan *MediaPipe Holistic* dan *OpenCV* serta dibuat dengan menggunakan bahasa pemrograman *Python*. Pada penelitian ini dibuat dataset yang dibatasi dengan 10 (sepuluh) *class* simbol yang mewakili kata dalam SIBI (Sistem Isyarat Bahasa Indonesia) dengan total 8734 baris data yang kemudian dilakukan *preprocessing* dengan membagi dataset menjadi 70% data latih dan 30% data uji. Pada penelitian ini, berisi tahapan-tahapan seperti membuat deteksi, membuat dataset, melatih model model klasifikasi, dan pengujian. Pengujian dilakukan dengan perhitungan akurasi menggunakan *Confusion Matrix* kemudian didapat tingkat *Accuracy* 98,6%, *Precision* 98,6%, dan *Recall* 98,66%. Dengan dibuatnya sistem deteksi SIBI (Sistem Isyarat Bahasa Indonesia) diharapkan dapat mengurangi kesenjangan antara penyandang tunarungu dan tunawicara dengan orang normal dalam berkomunikasi. Berkontribusi dalam pengetahuan mengenai deteksi simbol pada SIBI (Sistem Isyarat Bahasa Indonesia) menggunakan algoritma *Random Forest Classifier* serta *MediaPipe Holistic* dan *OpenCV*. Serta melestarikan dan mempopulerkan SIBI (Sistem Isyarat Bahasa Indonesia) sebagai bahasa isyarat yang ada di Indonesia.

Kata kunci: bahasa isyarat, sistem isyarat bahasa Indonesia (sibi), random forest classifier, mediapipe holistic, opencv, confusion matrix

APPLICATION OF *RANDOM FOREST CLASSIFIER* ALGORITHM IN INDONESIAN SIGN LANGUAGE SYSTEM (SIBI) DETECTION SYSTEM

Abstract

Sign language is a language that prioritizes body movements and facial expressions as symbols to communicate with each other. Deaf and speech-impaired groups are the main users of sign language. One type of sign language in Indonesia is SIBI (Indonesian Sign Language System). SIBI (Indonesian Sign Language System) is a form of spoken language that is converted into sign language. Sign language is not only used as a means of communication and interaction between deaf and mute people, but also with normal people. However, until now there is still a communication gap between people who are deaf or mute and normal people. Then made a system design that can detect the sign language symbol SIBI (Indonesian Sign Language System). This system is made using the *Random Forest Classifier* algorithm with the help of *MediaPipe Holistic* and *OpenCV* and is made using the *Python* programming language. In this study, a dataset was created which was limited to 10 (ten) classes of symbols representing words in the SIBI (Indonesian Sign System) with a total of 8734 data lines which were then pre-processed by dividing the dataset into 70% training data and 30% test data. This study contains stages such as making detection, creating datasets, training classification models, and testing. The test is carried out by calculating the accuracy using the *Confusion Matrix* and then getting the *Accuracy* rate of 98.6%, *Precision* of 98.6%, and *Recall* of 98.66%. With the creation of the SIBI (Indonesian Language Sign System) detection system,

it is hoped that it can reduce the gap between people who are deaf or speech impaired and normal people in communicating. Contribute to knowledge about symbol detection in SIBI (Indonesian Sign System) using the Random Forest Classifier algorithm as well as MediaPipe Holistic and OpenCV. As well as preserving and popularizing SIBI (Indonesian Sign System) as a sign language in Indonesia.

Keywords: *sign language, indonesian sign language system (sibi), random forest classifier, mediapipe holistic, opencv, confusion matrix*

1. PENDAHULUAN

Bahasa merupakan peranan penting sebagai alat komunikasi antara manusia satu dengan manusia lain. Dengan adanya bahasa cara manusia berkomunikasi terbilang unik, karena bahasa secara alami tidak hanya ditulis atau diucapkan tetapi juga diisyaratkan dan dapat disandikan ke dalam media menggunakan stimulus audio, visual, atau taktil seperti halnya bahasa isyarat. Bahasa isyarat digunakan sebagai bahasa alternatif yang sering digunakan oleh para penyandang tunarungu (tuli) dan tunawicara (bisu) sebagai pengguna utama untuk berkomunikasi. Pada dasarnya bahasa isyarat menggunakan bahasa tubuh yang menggunakan gerak tangan, wajah, dan gerak tubuh untuk mempresentasikan sebuah kata tetapi bahasa isyarat yang berupa alfabet juga dapat digunakan untuk membantu proses komunikasi pada kata-kata yang tidak memiliki bahasa tubuh dalam bahasa isyarat [1].

Orang tunarungu adalah orang yang mengalami kehilangan kemampuan mendengar baik itu sebagian atau seluruhnya yang diakibatkan kerusakan fungsi pendengaran baik sebagian atau seluruhnya hingga membawa dampak kompleks terhadap kehidupannya [2]. Sedangkan tunawicara adalah gangguan verbal pada seseorang sehingga mengalami kesulitan berkomunikasi melalui suara [3]. Bahasa isyarat adalah bahasa yang digunakan sehari-hari oleh penyandang disabilitas tunarungu dan tunawicara untuk berkomunikasi [4]. Di Indonesia terdapat salah satu jenis bahasa isyarat, yaitu SIBI (Sistem Isyarat Bahasa Indonesia). SIBI (Sistem Isyarat Bahasa Indonesia) adalah media komunikasi untuk penyandang tunarungu yang memadukan antara bahasa lisan, isyarat, mimik dan gerak lainnya [5]. SIBI (Sistem Isyarat Bahasa Indonesia) sengaja dibuat dan diresmikan oleh pemerintah Indonesia untuk mempresentasikan tata bahasa lisan Indonesia ke dalam isyarat buatan.

Namun, hingga saat ini masih terdapat kesenjangan komunikasi antara penyandang tunarungu dan tunawicara dengan orang normal. Untuk menyamakan makna atau persepsi antara keduanya memang memiliki pemahaman yang rumit, khususnya bagi orang normal yang belum pernah belajar bahasa isyarat. Saat ini terdapat teknologi *Machine Learning* yang dipadukan dengan *Computer Vision* sehingga dapat digunakan untuk mengatasi permasalahan ini. *Machine Learning* adalah salah satu cabang dari disiplin ilmu kecerdasan buatan (*Artificial Intelligence*) yang membahas mengenai pembangunan sistem berdasarkan pada data [6].

Computer Vision merupakan ilmu komputer yang bekerja dengan cara meniru kemampuan visual manusia [7].

Maka dibuatlah sebuah sistem deteksi menggunakan algoritma *Random Forest Classifier* dengan bantuan *MediaPipe Holistic* dan pustaka *OpenCV* serta bahasa pemrograman *Python* untuk mendeteksi simbol pada SIBI (Sistem Isyarat Bahasa Indonesia). Pada penelitian ini dibuat dataset yang dibatasi dengan 10 (sepuluh) *class* simbol yang mewakili kata dalam SIBI (Sistem Isyarat Bahasa Indonesia). Kemudian dilakukan pengujian dengan perhitungan *Confusion Matrix* untuk mengetahui tingkat akurasi dari model klasifikasi yang dibangun. Dengan dibuatnya sistem deteksi SIBI (Sistem Isyarat Bahasa Indonesia) diharapkan dapat mengurangi kesenjangan antara penyandang tunarungu dan tunawicara dengan orang normal dalam berkomunikasi.

Penelitian ini berkontribusi dalam pengetahuan mengenai deteksi simbol pada SIBI (Sistem Isyarat Bahasa Indonesia) menggunakan algoritma *Random Forest Classifier* serta *MediaPipe Holistic* dan *OpenCV*. Penelitian ini dapat menjadi referensi terkait dengan sistem deteksi bahasa isyarat. Serta menghasilkan sebuah dataset yang terbagi ke dalam 10 (sepuluh) *class* simbol SIBI (Sistem Isyarat Bahasa Indonesia). Dengan adanya sistem ini diharapkan dapat membantu melestarikan dan mempopulerkan SIBI (Sistem Isyarat Bahasa Indonesia) sebagai bahasa isyarat yang ada di Indonesia.

2. METODE PENELITIAN

2.1 Data Penelitian

Dataset yang digunakan sebagai data penelitian berupa koordinat-koordinat *landmark* yang direkam melalui citra video dengan manusia yang memeragakan simbol SIBI (Sistem Isyarat Bahasa Indonesia). Dataset diambil dan dikumpulkan secara *real-time* melalui *webcam*. Terdapat sebanyak 501 *landmark* yang berasal dari *pose landmark* dan *face landmark*. Setiap *landmark* memiliki titik-titik koordinat yang ditampung ke dalam variabel x , y , z , dan v (*visibility*) kemudian disimpan dalam file dengan format *excel (.csv)*. Dataset yang dibuat terdiri dari 10 (sepuluh) *class* simbol bahasa isyarat yang diantaranya adalah “Hai”, “Aku”, “Kamu”, “Terima”, “Kasih”, “Maaf”, “Cinta”, “Makan”, “Minum”, dan “Tidur”.

Pada dataset ini terdapat total 8734 baris data dengan masing-masing *class* lebih dari 800 data.

Pengambilan dataset dari citra video dilakukan selama kurang lebih 2 menit 30 detik untuk setiap *class*-nya. Simbol bahasa isyarat pada SIBI (Sistem Isyarat Bahasa Indonesia) diambil berdasarkan referensi website <https://pmpk.kemdikbud.go.id/sibi/> yang merupakan laman resmi. Dataset diambil menggunakan *webcam* eksternal. Posisi *webcam* diletakkan sejajar dengan subjek dengan jarak antara subjek dengan *webcam* adalah kurang lebih 1,2 meter. Dengan posisi *webcam* adalah *landscape*. Terdapat 4 (empat) subjek yang menjadi model dataset dengan latar belakang bebas yang ditunjukkan pada Gambar 1.



Gambar 1. Subjek Dataset SIBI

2.2 Data Penelitian

Dibutuhkan suatu gambaran untuk menjelaskan tahapan-tahapan yang digunakan pada penelitian dari awal hingga akhir. Dengan adanya tahapan penelitian ini, diharapkan agar alur penelitian tidak menyimpang. Dalam penelitian ini terdiri dari tahapan-tahapan yang diantaranya adalah Membuat Deteksi, Membuat Dataset, Melatih Model *Machine Learning*, dan Pengujian. Gambar 2 menunjukkan alur atau tahapan pada penelitian ini.



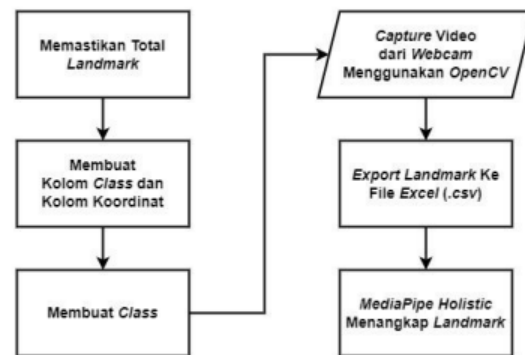
Gambar 2. Tahapan Penelitian

2.2.1 Membuat Deteksi

Pada tahapan ini, yang akan dilakukan adalah membuat beberapa deteksi. Memastikan bahwa pustaka *OpenCV* dan *MediaPipe Holistic* dapat bekerja dengan baik serta *webcam* yang dimiliki dapat merekam citra video secara *real-time*. *OpenCV* merupakan *API Library* yang menyediakan pustaka *Computer Vision* dan *Machine Learning*. *OpenCV* dibangun untuk menyediakan infrastruktur umum untuk aplikasi *Computer Vision* dan untuk mempercepat penggunaan persepsi mesin dalam produk komersial [8]. *MediaPipe* adalah kerangka kerja yang memungkinkan pengembang untuk membangun saluran *Machine Learning* multimodal (video, audio, seri waktu apapun) [9].

2.2.2 Membuat Dataset

Tahapan selanjutnya adalah pembuatan dataset. Pembuatan dataset juga memanfaatkan *MediaPipe Holistic* yang merupakan pustaka *Machine Learning* pra-bangun yang memungkinkan untuk melakukan estimasi *pose* dan *landmark* dengan sangat mudah dan akurat. Kemudian akan menangkap semua data sendi dan wajah pada bahasa isyarat pada *frame* yang diambil saat melakukan deteksi sehingga data tersebut dapat digunakan untuk melatih model *Machine Learning* nantinya. Data yang didapat di-render menggunakan *OpenCV* sehingga data dapat direkam melalui *webcam* secara *real-time* untuk mendeteksi *pose* bahasa isyarat yang berbeda. Koordinat-koordinat tersebut diekspor ke dalam file *excel (.csv)*. Tahapan membuat dataset ditunjukkan pada Gambar 3.

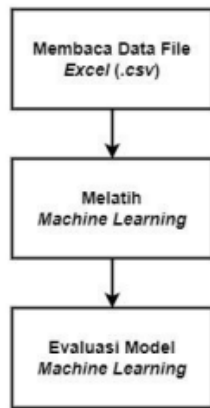


Gambar 3. Tahapan Membuat Dataset

2.2.3 Melatih Model *Machine Learning*

Setelah selesai membuat dataset, kemudian membangun dan melatih model *Machine Learning* khusus dengan menggunakan *scikit-learn*. Dengan *scikit-learn*, model *Machine Learning* yang dibangun dapat mengklasifikasikan masing-masing komponen bahasa isyarat yang berbeda. Model klasifikasi yang dibangun menggunakan algoritma *Random Forest Classifier*. Algoritma *Random Forest Classifier* merupakan pengembangan dari *decision tree*, setiap *decision tree* akan melakukan *training* data menggunakan sampel individu dan setiap atribut dipecah menjadi *tree* yang akan dipilih antara atribut subset yang bersifat acak [10]. Pada tahapan ini, pembuatan model *Machine Learning* akan dibagi menjadi 3 (tiga) bagian.

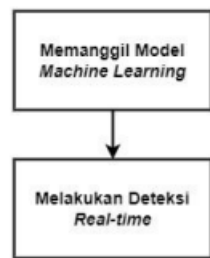
Dimana bagian pertama adalah membaca data dari file *excel (.csv)* yang telah dibuat pada tahapan sebelumnya. Bagian kedua adalah *preprocessing* dengan melakukan *train test split* dan melatih model klasifikasi *Machine Learning* agar dapat mendeteksi 10 (sepuluh) *class* bahasa isyarat berdasarkan koordinat gabungan. Terakhir, bagian ketiga merupakan evaluasi dari model terbaik yang telah dibangun dan menyimpan model tersebut ke dalam file format *pickle (.pkl)*. Gambar 4 menunjukkan tahapan dalam melatih model *Machine Learning*.



Gambar 4. Tahapan Melatih Model Machine Learning

2.2.4 Pengujian

Pada tahapan pengujian atau *testing*, model *Machine Learning* yang sudah dilatih kemudian disimpan. Model yang telah disimpan dapat digunakan untuk deteksi baru pada data koordinat dari *webcam*. Sehingga model yang telah dibangun tersebut dapat digunakan ke dalam *pipeline* pada saat deteksi yang dilakukan secara *real-time*. Tahapan *testing* ditunjukkan pada Gambar 5.



Gambar 5. Tahapan Testing

Pada tahapan selanjutnya, pengujian akan dilakukan dengan menggunakan penghitungan *Confusion Matrix* untuk mengevaluasi hasil dari sistem yang dibangun. *Confusion Matrix* adalah tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah [11]. *Confusion Matrix* yang akan digunakan pada rancangan pengujian ini adalah *Confusion Matrix Multi Class*. Tabel *Confusion Matrix* dapat dilihat pada Gambar 6 di bawah ini.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <small>Type I Error</small>
	0 (Negative)	FN (False Negative) <small>Type II Error</small>	TN (True Negative)

Gambar 6. Tabel Confusion Matrix

Confusion Matrix digunakan untuk membandingkan data hasil prediksi dengan data aktual. Setelah membandingkan kedua data tersebut, kemudian pengujian dilakukan dengan menghitung untuk mengetahui nilai *Accuracy*, *Precision*, dan *Recall*. Nilai *Accuracy* menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar. Maka, *Accuracy* merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Dengan kata lain, *Accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual. Rumus perhitungan *Accuracy* dapat dilihat pada Persamaan (1).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \quad \dots(1)$$

Nilai *Precision* menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Maka, *Precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Dari semua kelas positif yang telah diprediksi dengan benar, berapa banyak data yang benar-benar positif. Rumus perhitungan *Precision* dapat dilihat pada Persamaan (2).

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad \dots\dots(2)$$

Nilai *Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Maka, *Recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Rumus perhitungan *Recall* dapat dilihat pada Persamaan (3).

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad \dots\dots\dots(3)$$

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Metode

Pemilihan algoritma *Random Forest Classifier* pada penelitian ini dikarenakan algoritma ini mampu menghasilkan prediksi dengan tingkat akurasi tinggi dan mudah dipahami. Algoritma ini dapat bekerja secara efisien jika diterapkan pada dataset berskala besar. Seperti halnya pada sistem deteksi bahasa isyarat SIBI (Sistem Isyarat Bahasa Indonesia), sehingga mampu memberikan hasil yang jauh lebih baik. Pada penelitian ini, terdapat penjelasan dari tahapan-tahapan dalam mengimplementasikan metode.

3.1.1 Tahapan Membuat Deteksi

Tahap membuat deteksi merupakan tahapan awal setelah melakukan *install* dan *import* dependensi yang dibutuhkan. Awal dari tahap ini adalah mendapatkan *feed* dari *webcam* yang terhubung, sehingga deteksi dapat dilakukan secara *real-time*. Selanjutnya adalah memasukkan model *MediaPipe*

Holistic di dalam deteksi. Kemudian mengulang setiap *frame* yang masuk melalui *webcam* dan memeriksa apakah *webcam* terbuka (*on*) atau tidak (*off*) dan juga membaca *feed* dari *webcam*.

Ketika membaca bingkai (*frame*) menggunakan *OpenCV*, secara *default* umpan (*feed*) yang didapat akan berada dalam format *BGR* (*Blue, Green, Red*). Sehingga *feed* tersebut akan diwarnai ulang dalam format *RGB* (*Red, Green, Blue*), supaya dapat bekerja dengan *MediaPipe Holistic*. Setelah memasukkan model *MediaPipe Holistic* dalam deteksi yang selanjutnya disimpan dalam sebuah variabel. Kemudian umpan (*feed*) akan dikembalikan ke dalam format *BGR* (*Blue, Green, Red*) karena disesuaikan dengan *default* pada saat *me-render frame* menggunakan *OpenCV*.

Selanjutnya adalah menggambar *face landmark*, *right hand landmark*, *left hand landmark*, dan *pose landmark* pada setiap *frame* yang masuk melalui *webcam*. Setelah program dijalankan, dapat dilihat hasil yang ditampilkan di layar melalui *pop-up desktop GUI* deteksi yang *di-render* oleh *OpenCV* melalui *webcam* dengan *pipeline* yang telah dibangun menggunakan *MediaPipe Holistic* secara *real-time*. Contoh penerapan *MediaPipe Holistic* secara *real-time* dapat dilihat pada Gambar 7.



Gambar 7. Membuat Deteksi

3.1.2 Tahapan Membuat Dataset

Pada tahapan ini, hal pertama yang dilakukan sebelum membuat dataset adalah memastikan berapa banyak *landmark* yang dimiliki untuk ditangani nantinya. Pastikan total *landmark* yang dimiliki adalah 501 *landmark* dengan rincian 468 *face landmark* dan 33 *pose landmark*. Selanjutnya adalah mengatur kolom untuk mereferensikan setiap koordinat dengan nama kolomnya. Sehingga nantinya nama kolom pertama akan digunakan untuk mewakili *class* dan kolom lainnya mewakili koordinat *landmark*. Koordinat yang dimiliki adalah nilai *x*, *y*, *z*, dan *v* (*visibility*) dengan model wajah tidak memiliki komponen *visibility* sehingga *face landmark* hanya akan memiliki nilai 0 (*no*).

Nama kolom *class* yang dimiliki mewakili *pose* atau simbol bahasa isyarat seperti yang sudah

ditentukan yaitu “**Hai**”, “**Aku**”, “**Kamu**”, “**Terima**”, “**Kasih**”, “**Maaf**”, “**Cinta**”, “**Makan**”, “**Minum**”, dan “**Tidur**”. Kemudian nama kolom yang mewakili koordinat dimulai dengan *x1*, *y1*, *z1*, *v1* hingga sampai dengan *x501*, *y501*, *z501*, dan *v501* sebanyak dengan *landmark* yang dimiliki. Pada tahap ini, hanya menamai kolom yang akan digunakan dan belum mengeksport koordinat-koordinat ke dalam file *excel* (*.csv*).

Setelah melalui tahapan-tahapan tersebut, tahapan berikutnya dalam membuat dataset adalah membuat file *excel* baru menggunakan metode penulis dan menetapkan parameter kata kunci. File tersebut berisi kolom-kolom yang dapat menampung *class* dan koordinat-koordinat *landmark*. Selanjutnya adalah membuat nama *class*, pembuatan *class* akan dilakukan sebanyak jumlah *class* yang dibutuhkan. Setiap *class* akan mengumpulkan koordinat untuk *pose* dan *face landmark*.

Selanjutnya adalah membuat deteksi dengan baris kode yang sama dengan tahap membuat deteksi sebelumnya. Hanya saja ditambahkan sebuah baris kode yang dapat mengeksport koordinat-koordinat *landmark* ke dalam kolom file *excel* (*.csv*) pada saat merekam gerakan bahasa isyarat. *Class* yang dibuat akan diekspor ke dalam kolom *class* di dalam file *excel* (*.csv*) kemudian koordinat *pose* dan *face landmark* akan diekspor ke dalam kolom koordinat dengan nilai *x*, *y*, *z*, dan *v* (*visibility*) sebanyak 501 *landmark*.

3.1.3 Tahapan Melatih Model Machine Learning

Pada tahapan ini, terdapat 3 (tiga) bagian utama yaitu membaca dataset, melatih *Machine Learning*, dan evaluasi model *Machine Learning*.

a) Membaca Dataset

Tahap pertama adalah membaca data dari dataset yang telah dikumpulkan pada proses sebelumnya. Pemrosesan yang dilakukan pada tahap ini fungsi dari pustaka *scikit-learn* yaitu *train test split* yang akan memberikan partisi sehingga memungkinkan untuk membuat data latih (*train*) dan data uji (*test*). Tahap kedua setelah membaca file *excel* (*.csv*) adalah mengatur *feature* dan *target value* pada data. *Feature* merupakan informasi yang dapat digunakan untuk memprediksi *target value*, sehingga *feature* akan menjadi nilai koordinat pada nilai *x*, *y*, *z*, dan *v* (*visibility*). *Target value* merupakan apa yang akan dicoba untuk diprediksi, sehingga yang akan menjadi *target value* adalah *class*.

Pada Gambar 8 dan 9 menunjukkan nilai *Feature* dan *Target Value*. Selanjutnya adalah melakukan *preprocessing* dengan membuat data latih dan data uji menggunakan fungsi *train test split*. Perbandingan besarnya data latih dengan data uji adalah 70:30 atau 70% untuk data latih dan 30% untuk data uji. Dari 8734 total data yang telah dilakukan *preprocessing* didapat jumlah data latih 6133 dan data uji 2621.

	x1	y1	z1	v1	x2	y2	z2	v2	x3	y3	...
0	0.483488	0.474086	-1.825573	0.999753	0.520488	0.376458	-1.554430	0.999283	0.544793	0.371922	...
1	0.489124	0.474071	-1.677123	0.999764	0.523273	0.375830	-1.595434	0.999323	0.547985	0.370939	...
2	0.492425	0.473784	-1.812436	0.999770	0.523756	0.374633	-1.725019	0.999349	0.548386	0.369351	...
3	0.490344	0.463620	-1.600654	0.999777	0.521514	0.365175	-1.516715	0.999378	0.545415	0.360788	...
4	0.487833	0.458316	-1.559011	0.999783	0.518873	0.359926	-1.476635	0.999405	0.542086	0.355954	...
...											
8729	0.462801	0.501628	-0.644433	0.999739	0.495464	0.454432	-0.814805	0.999136	0.512903	0.457374	...
8730	0.416616	0.501052	-0.655071	0.999758	0.446597	0.450200	-0.626042	0.999205	0.466623	0.450819	...
8731	0.367456	0.500950	-1.064755	0.999758	0.399194	0.450715	-1.037650	0.999233	0.419009	0.450854	...
8732	0.328786	0.521379	-1.108459	0.999732	0.358307	0.455757	-1.082664	0.999219	0.377389	0.453619	...
8733	0.296758	0.553514	-1.156767	0.999656	0.324824	0.480404	-1.137196	0.989088	0.345017	0.473533	...

8734 rows x 2004 columns

Gambar 8. Feature

```

0      Hai
1      Hai
2      Hai
3      Hai
4      Hai
...
8729   Tidur
8730   Tidur
8731   Tidur
8732   Tidur
8733   Tidur
Name: class, Length: 8734, dtype: object

```

Gambar 9. Target Value

b) Melatih *Machine Learning*

Tahap pertama adalah *import* dependensi model klasifikasi yang berhubungan sehingga memungkinkan untuk membuat *Machine Learning pipeline*. *Pipeline* akan digunakan untuk pelatihan serta melewati fungsi *preprocessing* yang disebut *standard scaler*. Fungsi ini digunakan untuk menormalkan data sehingga mengurangi rata-rata dan membaginya dengan standar deviasi sehingga tidak ada satu *feature* yang menaungi *feature* lainnya. Kemudian memasukkan algoritma *Random Forest Classifier* dan melatih model *Machine Learning*.

Tahap kedua adalah menyiapkan *pipeline* dengan algoritma *Random Forest Classifier* sebagai model *Machine Learning*. Di dalam *pipeline* yang dibangun tersebut pertama data akan diteruskan ke *standard scaler* untuk membawa semua data ke dalam model *Random Forest Classifier* untuk dilatih.

Tahap berikutnya adalah melatih model *Machine Learning* tersebut. Dalam melatih model, yang dilakukan adalah melakukan *loop* pada *pipeline* dengan menggunakan metode *fit* yang artinya melatih dengan benar pada data latihan dan data uji. Setelah melatih model *Machine Learning*, akan didapat sebuah model klasifikasi *Random Forest Classifier* yang dapat dievaluasi untuk tahapan berikutnya.

c) Evaluasi Model *Machine Learning*

Evaluasi model dilakukan dengan menggunakan data uji. Evaluasi model bertujuan untuk menguji dan melihat seberapa baik akurasi dari model tersebut. Pertama yang dilakukan adalah *loop* model di dalam kamus metode *fit*. Kemudian membuat prediksi yang melewati data uji *feature* dan disimpan dalam sebuah variabel. Selanjutnya adalah mencetak nilai akurasi (*accuracy score*) yang didapat sebesar 98%.

r-f 0.9862647844334224

Gambar 10. Akurasi Model *Random Forest Classifier*

3.1.4 Tahapan Pengujian

Pada tahap pengujian, hal yang harus dilakukan adalah membuat deteksi secara *real-time* dengan model *Machine Learning* yang sudah dibangun. Hal yang dilakukan pertama adalah *reload* model yang telah disimpan di dalam file *pickle* (.pkl). Kemudian membuat deteksi dengan model tertentu dan menerapkan pengklasifikasian khusus. Pengujian merupakan proses mengeksekusi program. Pengujian diperlukan untuk mengetahui setiap perkembangan sistem untuk mengevaluasi, menganalisa, dan mengetahui akurasi atau kesamaan hasil yang dicapai oleh sistem yang telah dirancang.

Pengujian yang dilakukan pada penelitian ini adalah menghitung *Accuracy*, *Precision*, dan *Recall* pada implementasi algoritma *Random Forest Classifier* dalam pengklasifikasian dan prediksi bahasa isyarat SIBI (Sistem Isyarat Bahasa Indonesia) menggunakan *Confusion Matrix*. Pengujian dilakukan pada model klasifikasi yang telah dibangun dari dataset (.csv) dengan jumlah data sebanyak 8734 baris data. Terdapat 10 (sepuluh) *class* simbol bahasa isyarat yang diantaranya “Hai”, “Aku”, “Kamu”, “Terima”, “Kasih”, “Maaf”, “Cinta”, “Makan”, “Minum”, dan “Tidur” dengan masing-masing *class* lebih dari 800 data yang terdapat pada dataset. Hasil perhitungan pengujian direpresentasikan ke dalam gambar tabel *Confusion Matrix Multi Class 10x10*.

		Nilai Aktual										
		Aku	Cinta	Hai	Kamu	Kasih	Maaf	Makan	Minum	Terima	Tidur	Total
Nilai Prediksi	Aku	277	0	0	0	4	4	0	0	1	0	286
	Cinta	0	244	0	0	3	0	1	0	0	0	248
	Hai	1	0	296	2	0	0	0	0	0	0	299
	Kamu	0	1	1	298	0	0	0	0	1	0	301
	Kasih	1	2	0	0	250	0	0	1	0	0	254
	Maaf	0	0	0	0	1	252	6	0	0	0	259
	Makan	0	0	1	0	0	3	242	0	0	0	246
	Minum	0	0	0	0	0	0	0	250	0	0	250
	Terima	1	0	0	0	0	0	0	0	229	0	230
	Tidur	0	1	0	0	0	0	0	0	0	247	248
Total	280	248	298	300	258	259	249	251	231	247	2585	

Gambar 11. *Confusion Matrix Testing Model*

Berdasarkan Gambar 11, akan dicari nilai *Accuracy*, *Precision*, dan *Recall* menggunakan Rumus yang digunakan telah dijabarkan dalam Persamaan (1), Persamaan (2), dan Persamaan (3). yang dapat dilihat pada gambar tabel berikut:

Berdasarkan hasil perhitungan pengujian pada Gambar 12, didapat nilai akurasi yang sangat baik. Hal ini dipengaruhi oleh data *training* pada dataset (.csv) dan dapat dilihat bahwa model (.pkl) mampu memprediksi data dengan tingkat akurasi yang sangat baik. Sehingga dapat diketahui hasil pengujian algoritma *Random Forest Classifier* pada sistem deteksi SIBI (Sistem Isyarat Bahasa Indonesia)

dengan tingkat akurasi *Accuracy* 98,6%, *Precision* 98,6%, dan *Recall* 98,66%.

Pengujian	
<i>Accuracy</i>	$= ((277 + 244 + 296 + 298 + 250 + 252 + 242 + 250 + 229 + 247) / 2621) * 100\%$ $= (2585 / 2621) * 100\%$ $= 0,986 * 100\%$
<i>Precision</i>	$= ((277 / 280) + (244 / 248) + (296 / 298) + (298 / 300) + (250 / 258) + (252 / 259) + (242 / 249) + (250 / 251) + (229 / 231) + (247 / 247) / 10) * 100\%$ $= ((0,989 + 0,984 + 0,993 + 0,993 + 0,969 + 0,973 + 0,972 + 0,996 + 0,991 + 1) / 10) * 100\%$ $= (9,860 / 10) * 100\%$ $= 0,9860 * 100\%$
<i>Recall</i>	$= ((277 / 286) + (244 / 248) + (296 / 299) + (298 / 301) + (250 / 254) + (252 / 259) + (242 / 246) + (250 / 250) + (229 / 230) + (247 / 248) / 10) * 100\%$ $= ((0,969 + 0,984 + 0,990 + 0,990 + 0,984 + 0,973 + 0,984 + 1 + 0,996 + 0,996) / 10) * 100\%$ $= (9,866 / 10) * 100\%$ $= 0,9866 * 100\%$

Gambar 12. Perhitungan Pengujian

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan yang juga telah melalui tahapan perancangan dan pembuatan sistem serta pengujian sistem dengan menggunakan dataset yang dibuat (.csv) kemudian model klasifikasi (.pkl) dengan algoritma *Random Forest Classifier* dapat disimpulkan penggunaan algoritma *Random Forest Classifier* pada sistem deteksi SIBI (Sistem Isyarat Bahasa Indonesia) dapat berjalan secara efisien, dikarenakan sistem ini memiliki dataset berskala besar sehingga mampu memberikan akurasi dengan nilai yang tinggi.

Setelah dilakukan *training* pada model klasifikasi *Machine Learning* menggunakan *Random Forest Classifier* tingkat akurasi model tersebut sebesar 98.6%. Serta dari hasil training dilakukan perhitungan *Confusion Matrix* dengan hasil *training* menggunakan *Random Forest Classifier* serta *MediaPipe Holistic* didapatkan nilai performa dengan tingkat *Accuracy* 98,6%, *Precision* 98,6%, dan *Recall* 98,66%.

Adapun saran yang dapat diberikan supaya penelitian ini dapat berjalan lebih baik lagi yaitu menambah jumlah dataset dengan beberapa aspek yaitu jumlah simbol bahasa isyarat (*class*), variasi subjek yang menjadi model dataset, variasi tipe kamera dengan resolusi yang berbeda-beda, dan durasi pada saat merekam dataset.

Selanjutnya melakukan pembagian data dengan rasio pembagian yang lebih beragam untuk mendapatkan data yang optimal. Kemudian menggunakan *Machine Learning* pada bidang *Computer Vision* selain dengan *MediaPipe Holistic* dan *OpenCV*. Terakhir adalah menemukan metode

selain dengan algoritma *Random Forest Classifier* yang lebih memudahkan untuk melakukan klasifikasi terhadap dataset.

DAFTAR PUSTAKA

- [1] D. Yolanda, K. Gunadi, and E. Setyati, "Pengenalan Alfabet Bahasa Isyarat Tangan Secara Real-Time dengan Menggunakan Metode Convolutional Neural Network dan Recurrent Neural Network," *J. Infra*, vol. 8, no. 1, pp. 203–208, 2020, [Online]. Available: <https://publication.petra.ac.id/index.php/teknik-informatika/article/view/9791>
- [2] F. N. Rahmah, "Problematika Anak Tunarungu Dan Cara Mengatasinya," *Quality*, vol. 6, no. 1, p. 1, 2018, doi: 10.21043/quality.v6i1.5744.
- [3] L. Kurnia, "Kata Kunci: Anak usia dini, interaksi sosial, orangtua, dan tunawicara," vol. 1, no. 1, pp. 39–54, 2020.
- [4] Z. Zakaria, R. A. Firmansyah, and Y. A. Prabowo, "Rancang bangun Flex Sensor Gloves untuk penerjemah Bahasa Isyarat menggunakan K-Nearest Neighbors," *Semin. Nas. Sains dan Teknol. Terap. VII*, pp. 361–366, 2019, [Online]. Available: <https://ejurnal.itats.ac.id/sntekpan/article/view/597/400>
- [5] A. Sri Nugraheni, A. Pratiwi Husain, and H. Unayah, "Optimalisasi Penggunaan Bahasa Isyarat Dengan Sibi Dan Bisindo Pada Mahasiswa Difabel Tunarungu Di Prodi Pgm Uin Sunan Kalijaga," *Holistika*, pp. 28–33, 2021, [Online]. Available: jurnal.umj.ac.id/index.php/holistika
- [6] A. U. Zailani, A. Perdananto, Nurjaya, and Sholihin., "Pengenalan Sejak Dini Siswa Smp Tentang Machine Learning Untuk Klasifikasi Gambar Dalam Menghadapi Kommas: Jurnal Pengabdian Kepada Masyarakat," *KOMMAS J. Pengabd. Kpd. Masy.*, vol. 1, no. 1, pp. 7–15, 2020, [Online]. Available: <http://openjournal.unpum.ac.id/index.php/kommas/article/view/4599>
- [7] T. A. Dompeipen and M. E. I. Najoan, "Computer Vision Implementation for Detection and Counting the Number of Humans," *J. Tek. Inform.*, vol. 16, no. 1, pp. 65–76, 2021.
- [8] H. Muchtar and R. Apriadi, "Implementasi Pengenalan Wajah Pada Sistem Penguncian Rumah Dengan Metode Template Matching Menggunakan Open Source Computer Vision Library (Opencv)," *Resist. (elektRONika kEndali Telekomun. tenaga List. kOmpuTeR)*, vol. 2, no. 1, p. 39, 2019, doi: 10.24853/resistor.2.1.39-42.
- [9] H. M. Putri, W. Fuadi, T. Informatika, F. Teknik, U. Malikussaleh, and M. Holistic, "Pendeteksian Bahasa Isyarat Indonesia Secara Real-Time Menggunakan Long," 2022.
- [10] V. Sari, F. Firdausi, and Y. Azhar, "Perbandingan Prediksi Kualitas Kopi Arabika dengan Menggunakan Algoritma SGD, Random Forest dan Naive Bayes," *Edumatic J. Pendidik. Inform.*, vol. 4, no. 2, pp. 1–9, 2020, doi: 10.29408/edumatic.v4i2.2202.
- [11] D. Normawati and S. A. Prayogi, "Implementasi Naive Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 5, no. 2, pp. 697–711, 2021.