

# Pengenalan Bahasa Isyarat Indonesia menggunakan Mediapipe dengan Model Random Forest dan Multinomial Logistic Regression (*Introduction to Indonesian Sign Language Using Mediapipe With Random Forest Models and Multinomial Logistic Regression*)

Imam Suyudi<sup>1\*</sup>, Sudadio Sudadio<sup>2</sup>, Suherman Suherman<sup>3</sup>

Universitas Gadjah Mada, Daerah Istimewa Yogyakarta<sup>1,2,3</sup>

[Imamsuyudi@gmail.com](mailto:Imamsuyudi@gmail.com)



## Riwayat Artikel

Diterima pada 10 Februari 2023

Revisi 1 pada 16 Februari 2023

Revisi 2 pada 21 Februari 2023

Disetujui pada 23 Februari 2023

## Abstract

**Purpose:** This research aims to create a random forest machine learning model and logistic regression that can perform the sign language recognition of the Indonesian Sign Language System (SIBI) using a regular RGB camera with the MediaPipe framework.

**Research methodology:** Both variables in this study are measured using Innovative Work Behavior (IWB) Scale from Janssen (2000) and Connor-Davidson Resilience Scale (CD-RISC) from Connor & Davidson (2003) that was distributed through Google Form link. The data analysis is done with the support of the 25th version of SPSS (Statistical Package for Social Science).

**Results:** Resilience has a significant correlation with innovative work behavior among college students.

**Limitations:** No strict controls of questionnaire administration, the questionnaire consists of 6 different measurements from the research team, and can't be fully generalized to the college students population.

**Contribution:** New findings of correlation between two variables among new samples.

**Keywords:** *Sign Language Recognition, Indonesian Sign Language, SIBI, MediaPipe*

**How to cite:** Suyudi, I., Sudadio, S., Suherman, S. (2022). Pengenalan Bahasa Isyarat Indonesia menggunakan Mediapipe dengan Model Random Forest dan Multinomial Logistic Regression. *Jurnal Ilmu Siber dan Teknologi Digital*, 1(1), 65-80.

## 1. Pendahuluan

Bahasa isyarat merupakan salah satu cara berkomunikasi utama bagi tunarungu di seluruh dunia. Setiap negara memiliki bahasa isyarat masing-masing yang berbeda, bahkan banyak daerah yang juga memiliki bahasa ataupun dialek isyarat yang berbeda. Menurut World Federation of the Deaf, terdapat lebih dari 300 bahasa isyarat di seluruh dunia dan 70 juta tunarungu menggunakannya (Murray, 2018), sedangkan berdasarkan Kementerian Kesehatan (2014) di Indonesia terdapat 2.5 juta tunarungu. Di sisi lain jumlah masyarakat umum yang mempunyai kemampuan untuk memahami dan menggunakan bahasa isyarat sangat terbatas. Hal ini tentu menjadi masalah karena penutur bahasa isyarat tidak dapat berkomunikasi dengan mudah dengan masyarakat umum. Mayoritas teknologi komunikasi dikembangkan untuk mendukung komunikasi lisan ataupun tertulis tetapi tidak mempertimbangkan bahasa isyarat. Ketika alat dan teknologi komunikasi seperti Zoom dan Whatsapp menjadi bagian penting dari kehidupan masyarakat, tunarungu memiliki kesulitan menggunakan teknologi tersebut. Pengenalan bahasa isyarat adalah salah satu teknologi yang dapat membantu komunikasi penutur bahasa isyarat dan masyarakat umum. Dengan pengenalan bahasa isyarat penutur bahasa isyarat dapat diterjemahkan kedalam bentuk lisan ataupun teks yang dapat dengan mudah dimengerti oleh masyarakat umum, sehingga komunikasi dengan penutur bahasa isyarat dapat dilakukan.

Review mengenai pengenalan bahasa isyarat telah dilakukan oleh [Rastgoo et al. \(2021\)](#). Menurut penelitian tersebut terdapat beberapa perangkat utama yang dapat digunakan untuk pengenalan bahasa isyarat. Leap Motion Controller, sensor mekanis yang dipasang pada sebuah sarung tangan, kamera RGB dan yang paling umum digunakan kamera RGB + depth. Kamera RGB + depth umum digunakan karena dapat mengekstrak informasi 3D pada sebuah gestur tangan. Informasi 3D penting didapatkan karena bahasa isyarat memiliki gestur yang kompleks dan sering terjadi self-occlusion jika hanya menggunakan informasi 2D seperti kamera RGB biasa. Beberapa penelitian pengenalan bahasa isyarat Indonesia telah dilakukan se-belumnya dengan metode dan hasil yang beragam. [Suharijito et al. \(2021\)](#) menggunakan model Convolutional Neural Network (CNN) yang merupakan model Inflated 3D dikombinasikan dengan metode transfer learning pada Sistem Bahasa Isyarat Indonesia (SIBI) mendapatkan akurasi 97.5%. [Humaira et al. \(2018\)](#) menggunakan perangkat leap motion controller dengan metode K-Nearest Neighbor pada alfabet SIBI. Leap Motion Controller akan mendeteksi koordinat setiap titik sendi pada tangan kemudian titik tersebut akan dihitung jaraknya ke telapak tangan menggunakan euclidean distance. Penelitian tersebut mendapatkan akurasi sebesar 78%, dengan  $K = 5$ . [Handhika et al. \(2018\)](#) menggunakan perangkat kamera Microsoft Kinect yang dapat menerima informasi 3D dan metode Hidden-Markov Model (HMM) untuk pengenalan Bahasa Isyarat Indonesia (BISINDO). Penelitian tersebut mendapatkan akurasi sekitar 60%. [Ridwang \(2017\)](#) melakukan pengenalan bahasa isyarat berupa 24 alfabet SIBI statis dan tidak mencakup huruf J dan Z yang dinamis. Penelitian tersebut menggunakan perangkat Leap Motion Controller dengan algoritma naive bayes, penelitian tersebut mencapai akurasi 95%.

Namun penelitian yang sudah disebutkan diatas memiliki keterbatasan yaitu dibutuhkan perangkat khusus seperti Leap Motion Controller atau depth camera untuk mendapatkan informasi tiga dimensi dari gestur bahasa isyarat. MediaPipe merupakan sebuah kerangka kerja open source oleh google yang dapat merekonstruksi kerangka tangan tiga dimensi dari data dua dimensi yang didapatkan kamera RGB biasa. Sudah terdapat beberapa penelitian yang menggunakan rekonstruksi kerangka tangan dari sumber 2 dimensi, baik menggunakan MediaPipe sebagai pipeline ataupun tidak untuk pengenalan bahasa isyarat lain. Zimmermann dan Brox (2017) membuat estimasi pose tangan tiga dimensi untuk pengenalan bahasa isyarat dengan dataset RWTH German Fingerspelling Database dan mendapatkan akurasi sebesar 72.8%. [Zhang et al. \(2020\)](#) melakukan pengenalan bahasa isyarat menggunakan MediaPipe sebagai pipeline pada bahasa isyarat American Sign Language. Pada penelitian ini penulis menggunakan MediaPipe sebagai pipeline dan membuat model random forest dan Multinomial Logistic Regression untuk pengenalan bahasa isyarat Indonesia.

## **1.2 Rumusan Masalah**

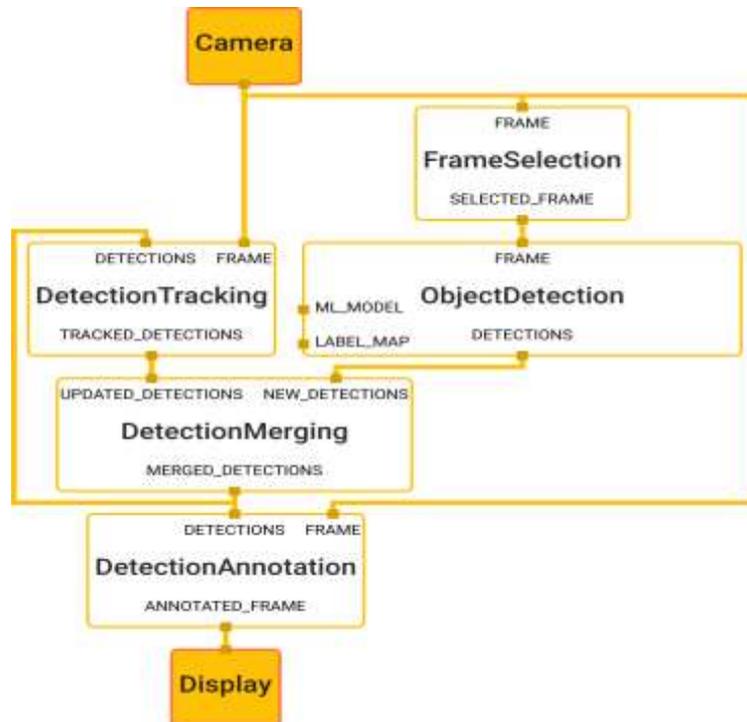
Berdasarkan latar belakang yang sudah dipaparkan sebelumnya, sudah terdapat beberapa implementasi pengenalan bahasa isyarat pada bahasa isyarat Indonesia baik pada Sistem Bahasa Isyarat Indonesia (SIBI) ataupun Bahasa Isyarat Indonesia (BISINDO) dengan berbagai metode dan hasil yang berbeda. Namun penelitian yang sudah ada memiliki keterbatasan dimana dibutuhkan perangkat khusus seperti Leap Motion Controller atau depth camera untuk pengenalan bahasa isyarat. Oleh karena itu, penelitian ini melakukan pengenalan bahasa isyarat dengan kamera RGB biasa menggunakan kerangka kerja MediaPipe dengan model random forest dan multinomial logistic regression.

## **2. Tinjauan pustaka dan pengembangan hipotesis**

### **2.1 Mediapipe**

Mediapipe adalah kerangka kerja open source yang didesain oleh Google. MediaPipe dapat digunakan sebagai pipeline untuk melakukan inferensi dari data citra ([Lugaresi et al., 2019](#)). MediaPipe menggunakan model machine learning dengan akurasi tinggi untuk pendeteksian dan pelacakan, melacak titik titik pada bagian tubuh atau disebut juga landmarks. Secara umum landmarks adalah titik koordinat tiga dimensi yang dinormalisasi pada rentang  $[0, 1]$ . Peneliti atau pengembang dapat menggunakan model MediaPipe tersebut dengan memodifikasi pipeline atau implementasi sistem dengan menggunakan model built-in MediaPipe. Salah satu implementasi menggunakan MediaPipe yang dapat dilakukan adalah pembuatan dataset bahasa isyarat dengan data berupa landmarks tiga dimensi. MediaPipe menyediakan API untuk beberapa implementasi. MediaPipe Face Mesh dapat

mendeteksi 468 landmarks tiga dimensi pada wajah. MediaPipe pose detection dapat merekonstruksi 33 landmarks pada postur tubuh. MediaPipe Hands untuk mendeteksi 21 landmarks kerangka tangan. MediaPipe Holistic untuk mendeteksi dan pelacakan 468 landmarks wajah, 33 pose tubuh dan 21 landmarks kerangka tangan (Lugaresi et al., 2019). Alur kerja MediaPipe secara umum dapat dilihat pada gambar 1.



## 2.2 NumPy

NumPy merupakan pustaka open source untuk bahasa pemrograman Python. NumPy mendukung operasi array dan matriks multi-dimensi berukuran besar, NumPy juga mendukung berbagai fungsi matematis pada array tersebut (Harris et al., 2020). NumPy memiliki keunggulan penggunaan memori dan runtime yang lebih kecil dibandingkan struktur data built in python, list.

## 2.3 Scikit-learn

Scikit-learn adalah pustaka pembelajaran mesin untuk bahasa pemrograman python. Scikit-learn menyediakan API untuk implementasi berbagai algoritma pembelajaran mesin, baik permasalahan klasifikasi, regresi ataupun clustering seperti support vector machines, random forests, gradient boosting dan k-means (Pedregosa et al., 2011). Scikit-learn didesain untuk bekerja dengan pustaka NumPy dan SciPy.

## 2.4 OpenCV

OpenCV (Open Source Computer Vision Library) adalah pustaka open source untuk melakukan berbagai operasi terkait computer vision secara real time. Operasi yang dapat dilakukan OpenCV mulai dari prapemrosesan citra sampai pembuatan model pembelajaran mesin (Bradski, 2000). OpenCV ditulis dalam bahasa pemrograman C++ tetapi memiliki wrapper python, sehingga terdapat API dalam bahasa pemrograman python.

## 2.5 Logistic Regression

Logistic regression merupakan algoritma klasifikasi untuk dua kategori kelas. Model logistic regression menggunakan binomial probability distribution function. Label kelas akan di-mapping menjadi 1 untuk kelas positif dan 0 untuk kelas negatif (Putra, 2022). Logistic regression merupakan algoritma klasifikasi biner dan tidak dapat melakukan klasifikasi lebih dari dua kelas. Pada persamaan 1

menunjukkan probabilitas sebuah data termasuk dalam suatu kelas. Dimana variabel  $p$  sebagai probabilitas,  $\exp$  sebagai eksponen (dengan aproksimasi bernilai 2,72),  $\alpha$  konstanta persamaa, dan  $\beta$  koefisien dari prediktor.

$$P = \frac{\exp(\alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \dots)}{1 + \exp(\alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \dots)} \quad (3.1)$$

### 2.5.1 Limited-memory BFGS

Limited-memory BFGS adalah algoritma optimasi yang digunakan pada model logistic regression penelitian ini. Limited-memory BFGS merupakan pendekatan algoritma Broyden-Fletcher-Goldfarb-Shanno (BFGS) menggunakan penyimpanan komputer yang minimal. L-BFGS menggunakan estimasi inverse hessian matriks untuk pencarian pada variabel ([Liu and Nocedal, 1989](#)). Algoritma L-BFGS hanya menyimpan vektor yang merepresentasikan aproksimasi secara implisit. Karena sifat algoritma L-BFGS yang membutuhkan penyimpanan komputer secara linear, metode L-BFGS digunakan untuk masalah dengan banyak variabel seperti penelitian ini.

## 2.6 Random Forest

### 2.6.1 Decision Tree

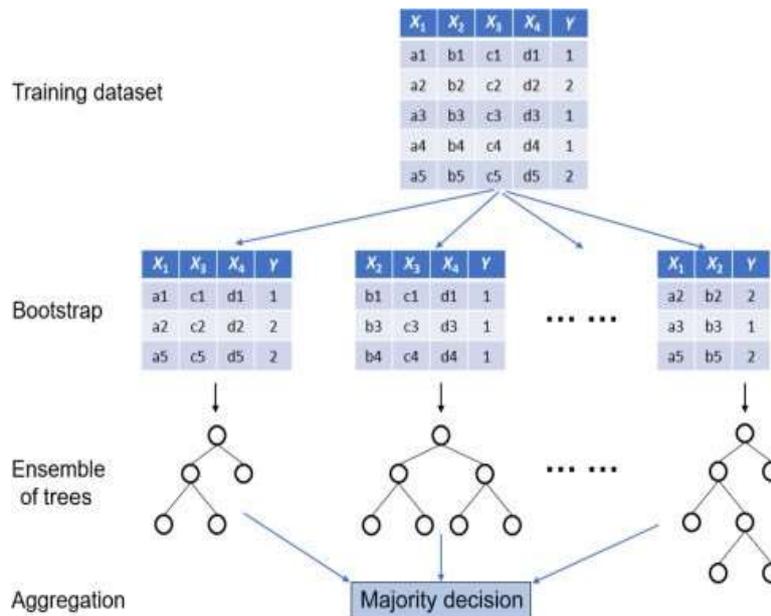
Algoritma decision tree merupakan model supervised machine learning. Decision tree memiliki struktur berupa binary tree dengan titik awal disebut root. Setiap percabangan menuju internal node ditentukan berdasarkan atribut data. Pada akhir cabang disebut leaf dan menandakan hasil klasifikasi yang dibuat. Decision tree menggunakan pemisahan biner rekursif untuk menentukan struktur tree (Setiawan, 2023). Setiap atribut dan percabangan yang berbeda dicoba dan diuji menggunakan cost function. Percabangan dengan cost terbaik akan dipilih. Terdapat beberapa cost function yang dapat digunakan decision tree, salah satu yang umum adalah gini impurity. Gini impurity secara matematis didapatkan dengan:

$$G(k) = \sum_{i=0}^n P(i) \times (1 - P(i)) \quad (3.2)$$

Pada persamaan 3.2 tersebut variabel  $P$  adalah probabilitas kelas dan variabel  $i$  menandakan kelasnya. Setiap atribut akan dihitung skor dan dipilih gini dengan nilai terendah. Skor gini terbaik yang mungkin adalah 0 yang berarti klasifikasi dapat dilakukan dengan tepat 100% pada atribut tersebut ([Rokach and Maimon, 2005](#)).

### 2.6.2 Random Forest Classifier

Random forest terdiri dari beberapa individu decision trees yang bekerja sebagai ensemble. Setiap tree pada random forest membuat prediksi kelas dan kelas dengan jumlah prediksi paling banyak menjadi prediksi model random forest ([Breiman, 2001](#)).



Gambar 2. Ilustrasi random forest (Misra and Li, 2020)

### 2.7 Ridge Regression

Ng (2004) menuliskan regularisasi mengontrol kompleksitas model dengan memberikan penalti pada model dengan nilai parameter yang tinggi. Dengan menghindari model yang terlalu kompleks, overfit pada model dapat dihindari. Ridge Regression atau L2 regularisasi menambahkan penalti pada loss function. Regularisasi memiliki parameter regularization rate ( $\lambda$ ) yang menentukan seberapa besar penalti digunakan untuk model. Jika  $\lambda$  terlalu besar dapat menyebabkan underfitting karena model tidak dapat menyesuaikan parameter berdasarkan data. Sebaliknya, jika  $\lambda$  terlalu kecil maka penalti yang diberikan tidak signifikan dan dapat menyebabkan overfitting. Secara matematis L2 regularisasi atau ridge regression dapat ditulis seperti berikut:

$$Loss = Error (Y - \hat{Y}) + \lambda \sum_1^n w_i^2 \quad (3,3)$$

### 2.8 Confusion Matrix

Metriks untuk menguji performance seperti precision, recall, dan akurasi membutuhkan data yang didapatkan dari hasil penelitian. Terdapat 4 atribut yang digunakan untuk membuat confusion matrix, Kulkarni et al. (2020):

1. True Positive: Jumlah data yang bernilai positif baik data riil dan nilai prediksi.
2. False Positive: Jumlah data yang bernilai positif pada nilai prediksi tetapi riilnya negatif.
3. True Negative: Jumlah data yang bernilai negatif baik pada data riil dan nilai prediksi.
4. False Negative: Jumlah data yang bernilai negatif pada nilai prediksi tetapi riilnya positif.

#### 2.8.1 Accuracy

Akurasi merupakan salah satu metriks yang paling umum digunakan karena mudah dihitung dan digunakan. Kekurangan dari metriks akurasi adalah kurang akurat digunakan ketika jumlah data tidak seimbang. Untuk menghitung akurasi menggunakan confusion matrix didapatkan dengan rumus:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3,4)$$

#### 2.8.2 Precision and recall

Precision adalah perbandingan antara True Positive (TP) dengan banyaknyadata dengan nilai prediksi positif. Secara matematis dapat ditulis dengan rumus:

$$Precision = \frac{TP}{TP + FP} \quad (3,5)$$

Sedangkan *recall* adalah perbandingan antara *True Positive (TP)* dengan banyaknya data yang riilnya positif. Secara matematis dapat ditulis dengan rumus:

$$Recall = \frac{TP}{TP + FN} \quad (3,6)$$

*Precision* dan *Recall* merupakan sebuah *tradeoff* dimana ketika *recall* tinggi, *precision* cenderung rendah, begitu juga sebaliknya. Hal ini disebabkan *precision* berusaha mengecilkan *False Positive* sedangkan *recall* mengecilkan *False Negative*

### 2.8.3 F1 Score

F-1 Score merupakan harmonic mean dari *precision* dan *recall*. Secara matematis dapat ditulis dengan rumus:

$$F1\ Score = \frac{2TP}{2TP + FP + FN} \quad (.37)$$

## 3. Metodologi

Terdapat dua tahap utama dalam penelitian ini. Tahap pertama adalah pembuatan dataset 24 alfabet dengan kamera RGB menggunakan MediaPipe, dimana setiap alfabet memiliki 500 data menggunakan tangan kiri dan 500 data menggunakan tangan kanan. Setiap data memiliki 21 titik kerangka dan setiap titik memiliki 3 koordinat yaitu: X, Y dan Z. Dataset tersebut disimpan dalam format csv. Tahap kedua adalah pembuatan model Logistic Regression dan random forest menggunakan pustaka scikit-learn.



Gambar 3. 21 titik kerangka tangan MediaPipe

## 4. Hasil dan Diskusi

### 4.1 Pembuatan Dataset

Pembuatan dataset alfabet SIBI dilakukan untuk setiap huruf dengan gestur statis, huruf J dan Z tidak termasuk dalam data karena kedua huruf tersebut memiliki gestur dinamis. Setiap kelas huruf memiliki 500 data untuk tangan kanan dan 500 data untuk tangan kiri. Sehingga jumlah total data untuk setiap kelas adalah 1000. Setiap data memiliki 21 titik kerangka tangan dan setiap titik memiliki 3 koordinat: X, Y dan Z. Sehingga setiap data memiliki 63 fitur. Alur proses pembuatan seperti berikut:

1. Penulis memperagakan gestur alfabet SIBI.
2. Detektor telapak tangan MediaPipe menerima data citra keseluruhan dan memprediksi *bounding box* dari telapak tangan.
3. MediaPipe menerima citra yang sudah di-crop berdasarkan *bounding box* telapak tangan dan mendeteksi 21 titik kerangka tangan.
4. 21 titik kerangka tangan yang dideteksi disimpan dalam file format csv.

Cuplikan hasil dataset yang sudah dibuat dapat dilihat pada gambar 6.1. Perlu diperhatikan koordinat yang tertulis dalam rentang -1 sampai 1 relatif terhadap telapak tangan, bukan terhadap posisi piksel dalam citra. Dataset yang disimpan dalam file *coords.csv* memiliki 24001 baris data, baris pertama untuk

header kolom dan 1000 data untuk setiap 24 kelas yang ada untuk baris lainnya. File coords.csv memiliki 64 kolom, kolom pertama sebagai label kelas dan 63 kolom berikutnya merupakan fitur yang menjadi data latih.

	A	B	C	D	E	F
1	class	x1	y1	z1	x2	y2
2	A	0.29956114292144775	0.6396486759185791	0.0	0.41998860239982605	0.6508626937866211
3	A	0.3218234181404114	0.7440775036811829	0.0	0.4336669445037842	0.7314149141311646
4	A	0.35703152418136597	0.7470749616622925	0.0	0.46487727761268616	0.7274889945983887
5	A	0.30298250913619995	0.6739739775657654	0.0	0.4186301827430725	0.6308207511901855
6	A	0.4096491038799286	0.8267799615859985	0.0	0.5384194850921631	0.749610960483551
7	A	0.40277111530303955	0.9381195902824402	0.0	0.5220469832420349	0.8132013082504272
8	A	0.38539788126945496	0.9276976585388184	0.0	0.5075526833534241	0.7801883220672607
9	A	0.390968382358551	0.9160727262496948	0.0	0.5086851119995117	0.7705343961715698
10	A	0.39449581503868103	0.9066445231437683	0.0	0.5053357481956482	0.7700906991958618
11	A	0.3838082551956177	0.8937861919403076	0.0	0.49350643157958984	0.7593477368354797
12	A	0.36862656474113464	0.8864946365356445	0.0	0.47822961211120453	0.7566302418708801
13	A	0.3445330262184143	0.8711742162704468	0.0	0.45746496319770813	0.7506887912750244
14	A	0.3260747194290161	0.8444877862930298	0.0	0.439521461725235	0.7274706959724426
15	A	0.31675389409065247	0.795820951461792	0.0	0.43038779497146606	0.6799494028091431
16	A	0.3095289468765259	0.7535542249679565	0.0	0.4243965148925781	0.6607152223587036
17	A	0.3042856752872467	0.7175690531730652	0.0	0.4258832335472107	0.6274664998054504
18	A	0.2896284759044647	0.6738890409469604	0.0	0.40817582607269287	0.615824818611145
19	A	0.28848695755004883	0.6494420766830444	0.0	0.4076049327850342	0.6175686120986938
20	A	0.2725546360015869	0.6491900086402893	0.0	0.40490013360977173	0.6293092370033264

Gambar 4. Cuplikan Dataset

## 4.2 Hasil Model Random Forest

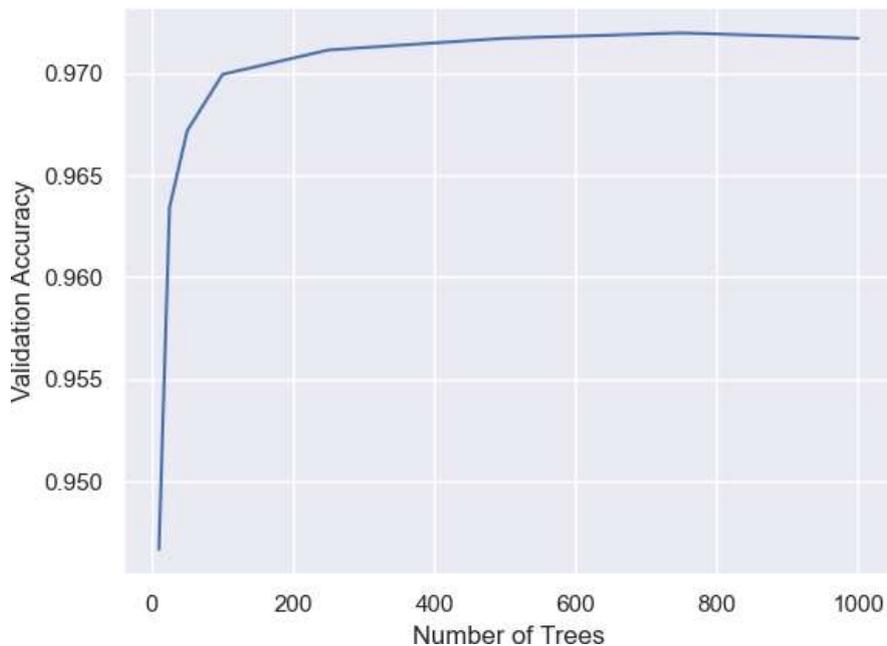
### 4.2.1 Hasil Pencarian Parameter Terbaik

Random Forest menggunakan bagging yaitu mengambil sampel dari data yang ada ketimbang menggunakan keseluruhan data dan hanya mengambil sebagian fitur secara acak untuk membangun sebuah tree. Jika jumlah data besar tetapi jumlah tree sedikit, terdapat kemungkinan ada data yang tidak pernah terambil. Jika jumlah fitur banyak tetapi jumlah tree sedikit, maka ada kemungkinan terdapat fitur yang tidak pernah terpilih ketika membangun sebuah tree. Kedua kasus tersebut dapat menurunkan kemampuan prediksi model random forest. Disaat yang sama jumlah tree yang terlalu besar membutuhkan komputasi dan waktu pelatihan yang lebih lama tanpa secara signifikan meningkatkan akurasi. Pada penelitian ini penulis menguji jumlah *tree* yang optimal dimana kenaikan jumlah *tree* sudah tidak signifikan meningkatkan akurasi, dengan dataset yang sudah dibuat sebelumnya menggunakan MediaPipe. Jumlah *tree* yang diuji pada penelitian ini adalah: 10, 25, 50, 100, 250, 500, 750 dan 1000. Akurasi setiap model random forest terhadap jumlah *tree* dapat dilihat pada tabel 6.1.

Tabel 1. Akurasi Model Random Forest

Jumlah Tree	Akurasi
10	94,66%
25	96,34%
50	96,71%
100	96,99%
250	97,11%
500	97,17%
750	97,19%
1000	97,17%

Grafik perkembangan akurasi model random forest terhadap jumlah *tree* dapat dilihat pada gambar 6.2. Terlihat model dengan jumlah *tree* 10 sudah memiliki akurasi cukup tinggi mencapai 94,66%, model dengan jumlah *tree* 25 memiliki kenaikan akurasi signifikan yaitu mencapai 96,34%, jika jumlah *tree* 50 maka akurasi yang didapatkan bertambah sedikit menjadi 96,71%, jumlah *tree* 100 terlihat menjadi asimtot dengan akurasi 96,99%, dimana kenaikan jumlah *tree* berikutnya tidak menambah akurasi secara signifikan dan akurasi tetap bernilai sekitar 97%. Pada penelitian ini jumlah *tree* yang digunakan pada data uji adalah pada jumlah sebanyak 750 karena memiliki nilai akurasi yang tinggi.



Gambar 5. Grafik Akurasi Model Random Forest Terhadap Jumlah Tree

#### 4.2.2 Hasil Evaluasi Model Random Forest

Model random forest dengan akurasi terbaik sebesar 97,19% yaitu dengan jumlah tree 750 disimpan. Model tersebut kemudian diuji dengan data test yang belum pernah ditemui sebelumnya oleh model, didapatkan akurasi tes sebesar 97,89% dan macro-averaged F1-score juga 97,89%. Setelah model membuat prediksi pada test data didapatkan precision, recall dan F1-Score untuk setiap kelas yang disimpan pada tabel 6.2. Prediksi model untuk setiap data pada data test kemudian disimpan dan dibuat menjadi confusion matrix yang dapat dilihat pada gambar 6.3.

Tabel 2. Precision, Recall dan F1 Score Model Random Forest

No	Huruf	Precision	Recall	F1 Score
1	A	99,46%	97,37%	98,40%
2	B	97,21%	99,43%	98,31%
3	C	95,37%	95,37%	95,37%
4	D	97,41%	97,84%	97,62%
5	E	97,55%	97,55%	97,55%
6	F	98,93%	95,36%	97,11%
7	G	99,04%	99,52%	99,28%
8	H	97,73%	97,18%	97,45%
9	I	99,05%	99,05%	99,05%
10	K	96,69%	94,09%	95,37%

11	L	97,52%	98,01%	97,77%
12	M	97,99%	98,48%	98,24%
13	N	96,21%	98,07%	97,13%
14	O	96,50%	96,98%	96,74%
15	P	98,49%	99,49%	98,99%
16	Q	100%	99,48%	99,74%
17	R	97,04%	97,52%	97,28%
18	S	96,48%	96,48%	96,48%
19	T	97,56%	100%	98,77%
20	U	97,95%	96,46%	97,20%
21	V	98,51%	98,02%	98,26%
22	W	98,07%	99,02%	98,54%
23	X	99,48%	98,45%	98,96%
24	Y	99,53%	100%	99,77%
25	<i>Macro-averaged</i>	97,91%	97,88%	97,89%

#### 4.2.3 Confusion Matrix Model Random Forest

Pada confusion matrix yang sudah dibuat dapat dilihat akurasi untuk setiap kelas. Dapat diamati model dapat memprediksi dengan baik dimana semua kelas dapat diprediksi dengan cukup baik dan hanya mengalami kesalahan akurasi yang sedikit. Kelas dengan kesalahan cukup banyak adalah kelas K yang diprediksi sebagai R sebanyak 6 kali. Hal ini kemungkinan dapat terjadi karena gestur K sedikit menyerupai R, dimana K berupa membuka jari telunjuk dan jari tengah, sedangkan R juga membuka jari telunjuk dan jari tengah tetapi kedua jari disilang. Ilustrasi gestur bahasa isyarat dapat dilihat pada gambar 6.3.



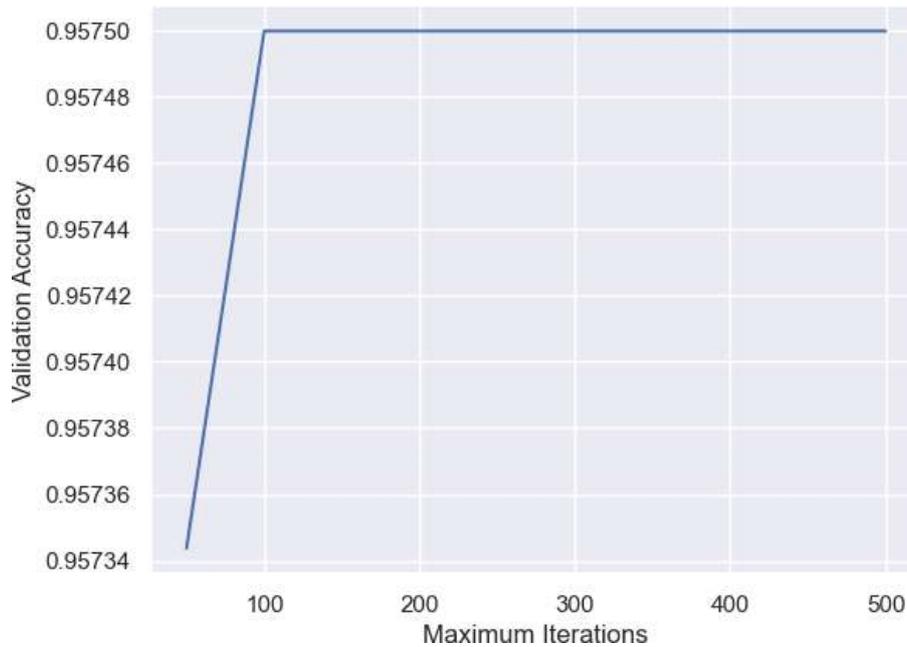
Gambar 6. Ilustrasi gestur bahasa isyarat SIBI

Kelas lain dengan kesalahan cukup banyak adalah kelas S yang diprediksi sebagai N sebanyak 4 kali. Kesalahan prediksi kemungkinan disebabkan karena gestur kelas S menyerupai N. Kedua kelas tersebut



500	95,74%
750	95,74%

Akurasi model logistic regression dengan *solver* liblinear dan L1 regularisasi terhadap maksimum iterasi dapat dilihat pada tabel 6.3 dan grafiknya dapat dilihat pada gambar 6.5. Model dengan maksimum iterasi 50 sudah mendapat akurasi cukup tinggi yakni sebesar 95,73%, meski begitu model dengan maksimum iterasi lebih tinggi tidak dapat menambah akurasi. Akurasi model tetap pada nilai 95,74%.

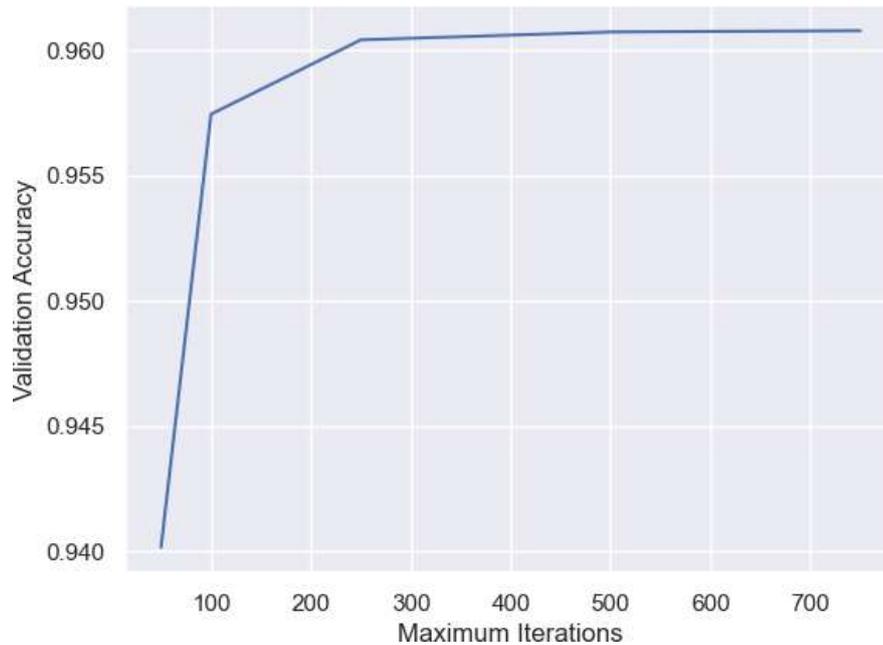


Gambar 8. Grafik Akurasi Model Logistic Regression L1 Regularisasi terhadap Maksimum Iterasi

Tabel 4. Akurasi Model Logistic Regression L2 Regularisasi

Maksimum Iterasi	Akurasi
50	94,01%
100	95,74%
250	96,04%
500	96,07%
750	96,07%

Akurasi model logistic regression dengan *solver* lbfgs dan L2 regularisasi terhadap maksimum iterasi dapat dilihat pada tabel 6.4 dengan grafiknya dapat dilihat pada gambar 6.6. Model dengan maksimum iterasi 50 mendapat akurasi cukup tinggi mencapai 94,01%, sedangkan model dengan maksimum iterasi 100 terdapat kenaikan akurasi signifikan mencapai 95,74%. Jika maksimum iterasi ditambah menjadi 250 didapatkan sedikit kenaikan akurasi mencapai 96,04%, meski begitu nilai maksimum iterasi 250 terlihat menjadi asimtot dimana maksimum iterasi yang lebih besar tidak dapat menambah akurasi dan akurasi tetap bernilai sekitar 96,07%.



Gambar 9. Grafik Akurasi Model Logistic Regression terhadap Maksimum Iterasi

#### 4.3.2 Hasil Evaluasi Model Logistic Regression

Model logistic regression dengan akurasi terbaik sebesar 96,07% yaitu dengan maksimum iterasi 750 dan L2 regularisasi disimpan. Model yang sudah disimpan tersebut kemudian diuji pada test data, dimana model belum pernah menemui sebelumnya. Didapatkan akurasi tes sebesar 96,34% dan macro-averaged F1-score sebesar 96,40%. Setelah model diuji pada test data didapatkan precision, recall dan F1-score untuk setiap kelas yang dapat dilihat pada tabel 6.5. Prediksi model untuk setiap data pada data tes disimpan dan dibuat sebuah confusion matrix yang dapat dilihat pada gambar 6.7.

Tabel 5. Precision, recall dan F1 Score Model Logistic Regression

No	Huruf	Precision	Recall	F1 Score
1	A	98,37%	95,26%	96,79%
2	B	98,30%	98,86%	98,58%
3	C	96,24%	94,91%	95,57%
4	D	95,11%	92,64%	93,86%
5	E	97,55%	97,55%	97,55%
6	F	97,33%	93,81%	95,54%
7	G	100%	97,12%	98,54%
8	H	95,60%	98,31%	96,94%
9	I	99,53%	99,53%	99,53%
10	K	92,93%	91,94%	92,43%
11	L	96,08%	97,51%	96,79%
12	M	97,46%	96,97%	97,22%
13	N	93,02%	96,62%	94,79%
14	O	92,27%	95,98%	94,09%
15	P	100%	100%	100%

16	Q	100%	100%	100%
17	R	96,97%	95,05%	96,00%
18	S	98,47%	96,98%	97,72%
19	T	97,54%	99,00%	98,26%
20	U	89,84%	84,85%	87,27%
21	V	87,38%	92,57%	89,90%
22	W	97,13%	99,02%	98,07%
23	X	98,46%	99,48%	98,97%
24	Y	98,62%	100%	99,30%
25	<i>Macro-averaged</i>	96,43%	96,41%	96,40%

#### 4.3.3 Confusion Matrix Model Logistic Regression

Confusion matrix pada gambar 6.7 dapat diamati jumlah prediksi untuk setiap data, dimana terdapat data dengan prediksi akurat dan data yang salah diprediksi menjadi kelas lain. Secara umum dapat terlihat bahwa model dapat memprediksi secara akurat dimana mayoritas kelas diprediksi sesuai dengan kelasnya dan membentuk garis miring yang pekat. Meski begitu masih terdapat beberapa kelas dengan frekuensi kesalahan cukup banyak. Salah satu kelas dengan kesalahan prediksi cukup besar adalah kelas U yang diprediksi sebagai V sebanyak 19 kali, dan sebaliknya kelas V diprediksi kelas U sebanyak 12 kali. Hal tersebut kemungkinan disebabkan karena gestur huruf V dan U yang saling menyerupai. Huruf V memiliki gestur kepalan tangan dengan membuka jari telunjuk dan jari tengah dengan kedua jari tersebut terdapat jarak sehingga seolah membentuk huruf V. Sedangkan gestur huruf U juga berupa kepalan tangan dengan membuka jari telunjuk dan jari tengah tetapi kedua jari tersebut tidak berjarak.

Kelas lain dengan kesalahan cukup besar adalah kelas C yang diprediksi sebagai O yang sebanyak 8 kali. Kesalahan ini juga terjadi cukup banyak pada model random forest. Hal ini kemungkinan disebabkan karena kedua kelas tersebut memiliki gestur saling menyerupai. Huruf C memiliki gestur tangan yang membentuk huruf C dengan ibu jari tidak menyentuh jari telunjuk. Sedangkan huruf O serupa seperti gestur huruf C tetapi ibu jari menyentuh jari telunjuk sehingga menyerupai bentuk huruf O. Terdapat beberapa kelas lain dengan kesalahan prediksi cukup banyak. Kelas D diprediksi sebagai kelas O sebanyak 7 kali, kelas D sebagai diprediksi sebagai kelas L sebanyak 5 kali, kelas R diprediksi sebagai K sebanyak 5 kali, kelas U diprediksi sebagai kelas K sebanyak 5 kali, dan masih terdapat beberapa kesalahan lainnya. Beberapa kesalahan prediksi kemungkinan disebabkan karena kedua kelas memiliki gestur yang serupa. Namun juga ada kesalahan prediksi meskipun kedua kelas memiliki gestur yang jelas berbeda. Salah satu penyebab yang mungkin adalah ketika pembuatan dataset menggunakan MediaPipe tidak berhasil membuat 21 titik kerangka tangan yang akurat, yang mungkin disebabkan oleh pengambilan sudut gambar yang tidak ideal.



Tabel 6. Akurasi Setiap Kelas

No	Huruf	Random Forest	Logistic Regression
1	A	97,36%	95,26%
2	B	99,42%	98,85%
3	C	95,37%	94,90%
4	D	97,83%	92,64%
5	E	97,54%	97,54%
6	F	95,36%	93,81%
7	G	99,51%	97,11%
8	H	97,17%	98,30%
9	I	99,05%	99,52%
10	K	94,08%	91,93%
11	L	98,00%	97,51%
12	M	98,48%	96,96%
13	N	98,06%	96,61%
14	O	96,98%	95,97%
15	P	99,49%	100%
16	Q	99,48%	100%
17	R	97,52%	95,04%
18	S	96,48%	96,98%
19	T	100%	99,00%
20	U	96,46%	84,84%
21	V	98,01%	92,57%
22	W	99,02%	99,02%
23	X	98,44%	99,48%
24	Y	100%	100%
Rata-rata		97,87%	96,41%

## 5. Kesimpulan

Penelitian ini berhasil membuat model random forest dan logistic regression dengan akurasi tinggi untuk pengenalan bahasa isyarat indonesia. Pembuatan dataset dengan hanya menyimpan 21 titik kerangka tangan menggunakan kerangka kerja MediaPipe terbukti efektif. Adapun kesimpulan yang didapatkan penulis adalah sebagaiberikut:

1. Model random forest memiliki asimtot pada jumlah tree 100 dengan nilai akurasi mencapai sekitar 97%.
2. Model logistic regression memiliki asimtot pada maksimum iterasi 250 dan L2 regularisasi dengan nilai akurasi mencapai sekitar 96%.
3. Kedua model masih memiliki kelemahan dimana terdapat beberapa kesalahan prediksi pada dua kelas dengan gestur serupa.

Saran penelitian-penelitian berikutnya adalah pembuatan dataset dengan jumlah data lebih besar, peraga gestur bahasa isyarat lebih banyak dan dataset dibuat dengan kondisi yang berbeda-beda. Untuk pembuatan model pembelajaran mesin dapat juga diuji menggunakan algoritma lainnya.

### Daftar Pustaka

- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Breiman, L. (2001). Random forests. *Machine Learning 2001 45:1*, 45:5–32.
- Handhika, T., Zen, R. I., Murni, Lestari, D. P., and Sari, I. (2018). Gesture recognition for Indonesian sign language (bisindo). *Journal of Physics: Conference Series*, 1028:012173.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.
- Humaira, F., Supria, S., Herumurti, D., and Widarsono, K. (2018). Real time sibi sign language recognition based on k-nearest neighbor. *undefined*, 5.
- Kulkarni, A., Chong, D., and Batarseh, F. A. (2020). Foundations of data imbalance and solutions for a data democracy. *Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering*, pages 83–106.
- Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical Programming 1989 45:1*, 45:503–528.
- Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., Grundmann, M., and Research, G. (2019). MediaPipe: A framework for building perception pipelines.
- Misra, S. and Li, H. (2020). Noninvasive fracture characterization based on the classification of sonic wave travel times. *Machine Learning for Subsurface Characterization*, pages 243–287.
- Murray, J. (2018). 70 million deaf people. 200+ sign languages. unlimited potential.
- Ng, A. Y. (2004). Feature selection,  $l_1$  vs.  $l_2$  regularization, and rotational invariance. *Proceedings, Twenty-First International Conference on Machine Learning, ICML2004*, pages 615–622.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Putra, T. I. Z. M., Suprpto, S., & Bukhori, A. F. (2022). Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Inception Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia. *Jurnal Ilmu Siber Dan Teknologi Digital*, 1(1), 1–28. <https://doi.org/10.35912/jisted.v1i1.1509>
- Rastgoo, R., Kiani, K., and Escalera, S. (2021). Sign language recognition: A deep survey. *Expert Systems with Applications*, 164:113794.
- Ridwang, R. (2017). Pengenalan bahasa isyarat Indonesia (sibi) menggunakan Leap Motion Controller dan algoritma data mining Naïve Bayes. *Jurnal INSYPRO (Information System and Processing)*, 2.
- Rokach, L. and Maimon, O. (2005). Decision trees. *Data Mining and Knowledge Discovery Handbook*, pages 165–192.
- Setiawan, E., Nurhatisyah, N., & Nanra, S. . . . (2023). Pengontrolan Bahaya Kebakaran Berbasis IOT pada Ruang Server SMFR Balai Monitor Spektrum Frekuensi Radio Kelas II Batam. *Jurnal Ilmu Siber Dan Teknologi Digital*, 1(1), 41–51. <https://doi.org/10.35912/jisted.v1i1.1800>
- Suharjito, Thiracitta, N., and Gunawan, H. (2021). Sibi sign language recognition using convolutional neural network combined with transfer learning and non-trainable parameters. *Procedia Computer Science*, 179:72–80.
- Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., and Grundmann, M. (2020). MediaPipe hands: On-device real-time hand tracking.