
ENSEMBLE ALGORITMA *RANDOM FOREST CLASSIFIER* DAN *ADABOOST* DALAM PERANCANGAN APLIKASI PENERJEMAH BAHASA ISYARAT

Andi¹⁾, Rezeki Ongsa Ong²⁾, Thamrin³⁾, Roseline⁴⁾^{1,2}Program Studi Sistem Informasi³Program Studi Magister Manajemen⁴Program Studi Akuntansi^{1,2,3}Institut Bisnis Informasi dan Teknologi Bisnis Medan

Jl. Mahoni No. 16, Gaharu, Kota Medan Telp:061-4530505

email: andi@itnb.ac.id¹⁾, rezeki.ongsa@itnb.ac.id²⁾, thamrin@itnb.ac.id³⁾, roselinezhuo@gmail.com⁴⁾

Abstrak

Penelitian ini mengembangkan aplikasi penerjemah bahasa isyarat berbasis web yang bertujuan untuk memudahkan komunikasi bagi penyandang disabilitas tunarungu. Aplikasi ini menggunakan ensemble algoritma *Random Forest Classifier* dan *AdaBoost* untuk mendeteksi dan menerjemahkan gestur tangan ke dalam teks secara *real-time*. Dataset yang digunakan bersifat lokal, dengan masing-masing gestur terdiri dari 100 citra data latih. Sistem ini dirancang untuk mengenali huruf A hingga Z serta beberapa kata umum seperti "hello", "done", "thank you", "I love you", "sorry", "please", dan "you are welcome". Hasil pengujian terhadap 14 skenario menunjukkan bahwa aplikasi berhasil menerjemahkan dengan tingkat akurasi sebesar 92%, sementara pengujian model mencatat akurasi *testing* sebesar 99,88%, lebih tinggi dibandingkan penggunaan *Random Forest* tunggal yang hanya mencapai 97,00%. Peningkatan akurasi sebesar $\pm 2,97\%$ menunjukkan bahwa kombinasi algoritma tersebut efektif dalam meningkatkan performa dan kestabilan sistem. Aplikasi ini diharapkan dapat membantu penyandang tunarungu berkomunikasi lebih efektif, meningkatkan aksesibilitas informasi, serta mendorong pengembangan teknologi di bidang penerjemahan bahasa isyarat.

Kata Kunci: Aplikasi, Penerjemah Bahasa Isyarat, Ensemble Algoritma, *Random Forest Classifier*, *Adaptive Boosting*

1. Pendahuluan

Tunarungu atau disabilitas pendengaran adalah seseorang yang mengalami gangguan pendengaran, baik sebagian ataupun menyeluruh, dan biasanya memiliki hambatan dalam berbahasa dan berbicara. Tunarungu memiliki keterbatasan dalam indra sehingga berdampak pada kehidupannya sehari-hari terutama dalam kemampuan berbahasa yang digunakan sebagai sarana komunikasi penting untuk bersosialisasi dengan orang lain [1]. Oleh sebab itu, penyandang tunarungu menggunakan bahasa isyarat untuk berkomunikasi dengan individu lainnya [2]. SIBI atau Sistem Isyarat Bahasa Indonesia merupakan bahasa yang digunakan oleh penyandang tunarungu sebagai media untuk berkomunikasi dengan masyarakat [3].

Namun, terdapat kesulitan dalam berkomunikasi tunarungu dengan orang-orang yang tidak mengerti bahasa isyarat sehingga diperlukan teknologi informasi yang dapat membantu menerjemahkan bahasa isyarat kedalam bahasa lisan sehingga memudahkan komunikasi bagi kedua belah pihak [3]. Penelitian yang membahas mengenai implementasi teknologi informasi dalam penerjemahan bahasa isyarat sudah pernah dilakukan sebelumnya yaitu pada tahun 2022 yang membahas mengenai perancangan aplikasi penerjemah bahasa isyarat ke dalam huruf abjad menggunakan algoritma klasifikasi Convolutional Neural Network (CNN). Hasil penelitian tersebut menyimpulkan bahwa algoritma CNN dalam menerjemahkan bahasa isyarat mendapatkan akurasi 0,995 atau 99,5% [4]. Penelitian terbaru yaitu di tahun 2023 mengimplementasikan algoritma Long-Short Term Memory (LSTM) dalam pengembangan aplikasi penerjemah bahasa isyarat. Hasil penelitian menunjukkan bahwa algoritma LSTM mendapatkan akurasi di antara 75%-85% [3].

Pada penelitian ini akan dikembangkan aplikasi penerjemah bahasa isyarat secara *real-time* dengan menggunakan ensemble algoritma *Random Forest Classifier* dan *AdaBoost*. Penelitian ini memilih algoritma *Random Forest Classifier* dikarenakan berdasarkan penelitian terdahulu algoritma ini memiliki kelebihan dalam hal kecepatan komputasi, kemampuan pemrosesan data yang lebih baik serta tingkat akurasi yang stabil dibandingkan algoritma CNN dan LSTM [5][6].

Random Forest Classifier adalah algoritma ensemble berbasis Decision Tree yang membangun banyak pohon keputusan secara acak dan menggabungkan hasilnya melalui voting mayoritas. Pendekatan ini membuat model lebih stabil, akurat, dan tahan terhadap overfitting. Sementara itu, *AdaBoost* (*Adaptive Boosting*) adalah metode boosting yang meningkatkan kinerja model dengan memfokuskan pelatihan pada data yang sebelumnya salah

diklasifikasikan. Dengan memberi bobot lebih pada kesalahan, AdaBoost membantu meningkatkan presisi dan generalisasi model.

Oleh sebab itu kombinasi model ensemble ini diharapkan dapat menerjemahkan bahasa isyarat secara akurat dengan hasil yang tepat.

2. Landasan Teori

Digital Image

Citra digital atau digital *image* adalah gambar yang terdiri dari elemen gambar, juga dikenal sebagai piksel, masing-masing dengan jumlah representasi numerik yang terbatas dan diskrit untuk intensitas atau tingkat keabuaannya [7].

Digital Image Processing

Pengolahan citra digital (*digital image processing*) mempelajari cara sebuah citra dibentuk, dianalisis, dan diolah sedemikian rupa agar menghasilkan data-data atau informasi yang dapat dipahami secara manusiawi. Salah satu kegunaan pengolahan citra yaitu untuk memperbaiki kesalahan sinyal pada gambar atau piksel yang rusak yang disebabkan oleh adanya gangguan pada saat pembuatan gambar tersebut [7].

Kecerdasan Buatan

Kecerdasan buatan atau yang sering dikenal dengan *Artificial Intelligence* (AI) merupakan kemampuan mesin atau sistem komputer untuk meniru atau menunjukkan kecerdasan manusia [8].

Machine Learning

Machine Learning (ML) adalah cabang ilmu komputer yang berkembang dari bidang pengenalan pola dan teori pembelajaran komputer dalam kecerdasan buatan [9].

Knowledge Discovery

Knowledge Discovery atau *Knowledge Discovery in Database* (KDD) adalah salah satu untuk memperoleh pengetahuan dari basis data yang memiliki tabel-tabel yang saling berhubungan atau berelasi. Hasil dari pengetahuan yang didapatkan dari proses tersebut menjadi dasar sebagai basis pengetahuan (*Knowledge Base*) dalam pengambilan keputusan [10].

Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah bagian dari *deep learning* yang membuat gambar untuk dikenali dan diklasifikasi. Metode CNN adalah sebuah kelas pada *neural network* yang berjalan dalam bidang pemrosesa data-data yang memiliki topologi dalam bentuk *grid*, contohnya yaitu gambar [11][12].

Algoritma Random Forest Classifier

Random Forest Classifier merupakan sebuah metode *ensemble* yang terdiri atas sekumpulan *decision tree* (pohon keputusan) yang digunakan untuk klasifikasi data ke suatu kelas [13]. Langkah awal untuk menentukan pohon keputusan adalah dengan menghitung nilai *entropy* dan *information gain*. Terdapat dua buah formula dari algoritma *Random Forest Classifier* dalam membentuk pohon keputusan antara lain [14]:

a. Entropi.

Pendekatan entropi sebagai penentu ketidakmurnian atribut. Penghitungan nilai entropi dapat dilihat pada persamaan berikut:

$$Entropy(S) = \sum_{i=1}^n - p_i \log_2 p_i \quad (1)$$

Keterangan:

S : Himpunan dataset

n : Banyaknya jumlah kelas

p_i : Proporsi dari S_i terhadap S

b. Information Gain.

Information gain merupakan nilai perolehan informasi dalam memilah simpul. Perhitungan *gain* sangat berpengaruh terhadap setiap node teratas dan node pemisah. Perhitungan gini masih berlanjut ketika hasil akhir gini masih berupa angka dan berhenti ketika hasil akhir gini adalah nol.

$$Information\ Gain\ (A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (2)$$

Keterangan:

A : Atribut

S : Himpunan dataset

$|S_i|$: Jumlah sampel untuk nilai ke-i

$|S|$: Banyaknya jumlah data

Adaptive Boosting

Algoritma *Adaptive Boosting* merupakan algoritma yang digunakan untuk pengambilan keputusan. *Adaboost* merupakan *ensemble learning* yang digunakan pada algoritma *boosting*. Algoritma ini ditujukan untuk *supervised learning* yang memberikan nilai atribut pada dataset yang digambarkan oleh koleksi atribut dan termasuk salah satu dari serangkaian kelas yang saling berhubungan. Langkah-langkah pada algoritma *Adaboost* adalah sebagai berikut [15]:

1. Inisial setiap bobot awal dengan rumus berikut untuk semua i .

$$W_i^1 = \frac{1}{n} \quad (3)$$

2. Untuk setiap $m = 1, 2, 3, \dots, M$. Lakukan:
3. Bangun model klasifikasi $G_m(X_i)$ menggunakan bobot $W_i^{(m)}$.
4. Hitung kesalahan klasifikasi dengan rumus:

$$\varepsilon_m = \frac{\sum_{i=1}^n w_i^{(m)}, G_m(X_i) \neq Y_i}{\sum_{i=1}^n w_i^{(m)}} \quad (4)$$

5. Menghitung nilai α menggunakan rumus:

$$\alpha_m = \log \frac{1 - \varepsilon_m}{\varepsilon_m} \quad (5)$$

6. Perbarui koefisien pembobotan kesalahan klasifikasi dengan rumus:

$$w_i^{m+1} = w_i^m \exp(-\alpha_m), G_m(X_i) = Y_i \quad (6)$$

$$w_i^{m+1} = w_i^m \exp(\alpha_m), G_m(X_i) \neq Y_i \quad (7)$$

7. Penentuan klasifikasi menggunakan rumus:

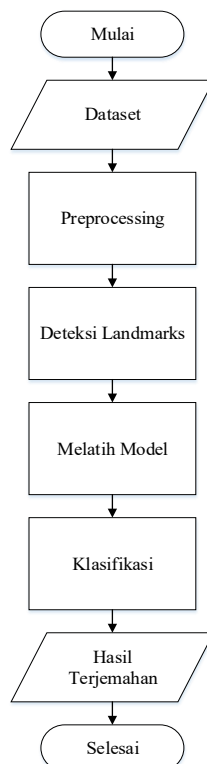
$$Y = \text{Sign} \left(\sum_{m=1}^M \alpha_m G_m(x) \right) \quad (8)$$

Penelitian Terdahulu

Penelitian yang membahas mengenai implementasi teknologi informasi dalam penerjemahan bahasa isyarat sudah pernah dilakukan sebelumnya yaitu pada tahun 2022 yang membahas mengenai perancangan aplikasi penerjemah bahasa isyarat ke dalam huruf abjad menggunakan algoritma klasifikasi *Convolutional Neural Network* (CNN). Hasil penelitian tersebut menyimpulkan bahwa algoritma CNN dalam menerjemahkan bahasa isyarat mendapatkan akurasi 0,995 atau 99,5% [4]. Penelitian terbaru yaitu di tahun 2023 mengimplementasikan algoritma *Long-Short Term Memory* (LSTM) dalam pengembangan aplikasi penerjemah bahasa isyarat. Hasil penelitian menunjukkan bahwa algoritma LSTM mendapatkan akurasi di antara 75%-85% [3].

3. Metode Penelitian

Berikut ini, Gambar 1 menunjukkan *flowchart* tahapan metode penelitian yang dilaksanakan pada penelitian ini.



Gambar 1. *Flowchart* Tahapan Metode Penelitian

Berdasarkan Gambar 1 diuraikan tahapan metode penelitian yang akan dilakukan pada penelitian ini antara lain:

a. *Dataset*.

Tahap di mana data bahasa isyarat dikumpulkan. *Dataset* ini akan menjadi dasar untuk melatih model pengenalan bahasa isyarat. *Dataset* pada penelitian ini diambil dari dataset lokal dengan setiap kata bahasa isyarat masing-masing memiliki 100 gambar data latih. Contoh salah satu dataset yang digunakan dapat dilihat pada Gambar 2 berikut.



Gambar 2. Contoh *Dataset* Penelitian

b. *Preprocessing*.

Tahap *preprocessing* pada penelitian ini dilakukan dengan cara melakukan proses *resize* terhadap *dataset* gambar menjadi ukuran 200x200 dan dilakukan proses konversi ke *grayscale* ketika proses *capture dataset* dilakukan. Asumsikan terdapat sebuah citra dengan ukuran 200x200 seperti Gambar 3 berikut.



Gambar 3. *Dataset* Bahasa Isyarat Makan

Kemudian dari citra RGB pada Gambar 3 akan diakses sebuah *pixel* secara *random* untuk mengetahui nilai RGB salah satu *pixel*. Pada *pixel* $x = 100$ dan $y = 100$ diketahui Nilai RGB = 57 34 16. Dari data tersebut maka dapat dihitung *Grayscale* dengan cara sebagai berikut:

Diketahui = Red (merah) = 57

Green (hijau) = 34

Blue (biru) = 16

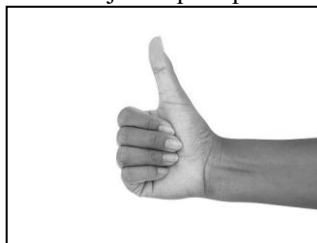
$Grayscale = 0.2989 R + 0.5870 G + 0.1140 B$

$Grayscale = (0.2989 \times 57) + (0.5870 \times 34) + (0.1140 \times 16)$

$Grayscale = 17.0373 + 19.958 + 1.824$

$Grayscale = 38.8193$

Citra *Grayscale* merupakan citra dengan ukuran 8 bit. Sehingga memiliki $(2^8 - 1)$ kemungkinan warna mulai dari 0 hingga 255 dimana 0 berarti hitam dan 255 berarti putih. Warna abu memiliki *range* dari 1 hingga 254 dimulai dari abu paling gelap hingga abu terang dan akhirnya mendekati putih. Jadi hasil *grayscale* = 38.8193 merupakan warna abu-abu yang cenderung mendekati hitam. Kemudian aplikasikan rumus tersebut di semua *pixel* dan hasil dari konversi citra RGB pada Gambar 3.3 akan menjadi seperti pada Gambar 4 berikut.



Gambar 4. Citra *Grayscale*

c. Deteksi *Landmarks*.

Selanjutnya, dataset yang telah melewati tahap *preprocessing* dilanjutkan pada tahapan berikutnya yaitu mendeteksi *landmarks* atau titik-titik penting pada gerakan tangan dalam bahasa isyarat. Berikut ini uraian cara kerja deteksi *landmarks* yang digunakan:

1) Model *Hands*:

MediaPipe menyediakan model *Hands* yang digunakan untuk mendeteksi titik landmark tangan dalam citra.

2) Mendeteksi *Landmarks*:

Dengan menggunakan model *Hands*, citra masukan diproses untuk mendeteksi titik-titik *landmark* tangan.

Titik-titik *landmark* ini adalah titik-titik penting dalam tangan yang digunakan untuk mengidentifikasi pose tangan.

3) *Landmarks* pada tangan:

Setiap tangan dapat memiliki beberapa *landmarks*, yang mewakili titik-titik penting seperti ujung jari, pangkal jari, dan titik-titik lainnya yang membentuk pose tangan.

Jumlah total *landmarks* yang dideteksi tergantung pada pose tangan yang diidentifikasi dalam citra.

4) Penggunaan *Landmarks*:

Setelah mendeteksi *landmarks*, koordinatnya digunakan sebagai fitur masukan untuk model klasifikasi. Dalam kasus ini, koordinat *landmarks* tangan digunakan sebagai fitur masukan untuk model klasifikasi bahasa isyarat.

Dengan menggunakan *landmarks* tangan sebagai fitur masukan, model dapat belajar untuk mengenali pola-pola yang berkaitan dengan bahasa isyarat dari citra tangan yang diberikan. *Landmarks* ini menjadi representasi penting dari tangan dalam konteks pengenalan bahasa isyarat.

d. Melatih Model.

Setelah proses pelatihan *landmarks* dataset dilakukan, selanjutnya dilakukan pelatihan dengan menggunakan algoritma *Random Forest Classifier* dan *AdaBoost*. Asumsikan terdapat 3 kelas citra yaitu makan, tidur, dan minum. Berikut ini uraian cara kerja perhitungan manual dari algoritma *Random Forest Classifier* dan *AdaBoost* antara lain:

1) Memilih fitur terbaik:

Misalkan dipilih fitur pertama sebagai pemisah. Fitur ini adalah jumlah piksel hitam dalam gambar.

2) Memisahkan data:

Pertama, kita hitung jumlah piksel hitam dalam setiap gambar untuk semua kata.

Berdasarkan nilai fitur pertama (jumlah piksel hitam), dipisahkan dataset menjadi dua kelompok: satu dengan nilai fitur kurang dari atau sama dengan nilai ambang batas, dan satu lagi dengan nilai fitur lebih besar dari nilai ambang batas.

3) Membuat keputusan:

Jika jumlah piksel hitam kurang dari atau sama dengan nilai ambang batas, maka pergi ke simpul anak kiri.

Jika jumlah piksel hitam lebih besar dari nilai ambang batas, maka pergi ke simpul anak kanan.

4) Rekursi:

Langkah-langkah di atas diulangi untuk setiap simpul anak, baik yang kiri maupun kanan, hingga tidak ada lagi data atau kriteria pemberhentian lainnya terpenuhi.

Contoh perhitungan manual:

Misalkan memiliki *dataset* sebagai berikut (dalam jumlah piksel *grayscale*):

1) "DONE": 100

2) "HALO": 150

3) "SORRY": 120

Misalkan dipilih nilai ambang batas (misalnya, 130).

1) Karena $100 \leq 130$, kita tempatkan kata "DONE" di simpul anak kiri.

2) Karena $150 > 130$, kita tempatkan kata "HALO" di simpul anak kanan.

Dengan demikian, didapatkan satu pohon keputusan sederhana yang memisahkan kata-kata tersebut berdasarkan jumlah piksel hitam dalam gambar.

e. Klasifikasi.

Dengan menggunakan *landmarks* yang terdeteksi dan model klasifikasi yang telah dibuat, maka selanjutnya diklasifikasikan bahasa isyarat ke dalam kategori yang sesuai. Ini adalah tahap di mana terjemahan dari bahasa isyarat dilakukan, di mana model mengidentifikasi isyarat yang dilakukan dan menentukan artinya berdasarkan kelas yang telah dipelajari selama pelatihan.

f. Hasil Terjemahan.

Output dari sistem, berupa hasil terjemahan dari bahasa isyarat. Hasil ini disajikan dalam bentuk representasi visual secara *real time* serta teks hasil terjemahan yang memungkinkan pengguna untuk memahami pesan yang disampaikan dalam bahasa isyarat.

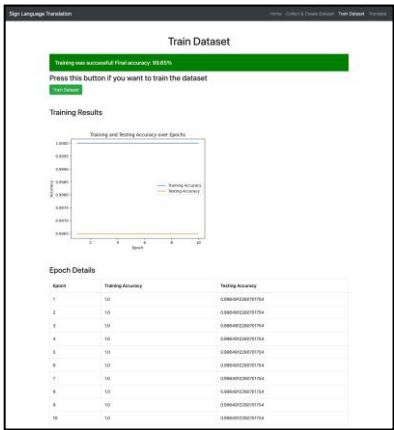
Perancangan Sistem

Tahapan perancangan sistem yang dilakukan pada penelitian ini adalah berupa perancangan tampilan aplikasi dengan menggunakan *software* Balsamiq Mockup 3.

4. Hasil Penelitian

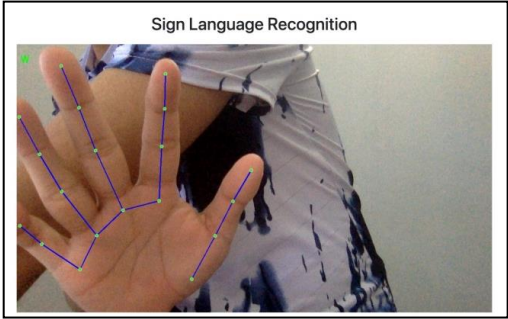
Hasil penelitian ini berupa dibangunnya sebuah aplikasi penerjemah bahasa isyarat menggunakan ensemble algoritma *Random Forest Classifier* dan *AdaBoost*. Hasil aplikasi berupa aplikasi berbasis *website* dengan keseluruhan rancangannya sebagai berikut:

1. Halaman *Training Dataset*.
Halaman yang berisikan sebuah tombol untuk melakukan proses *training dataset* untuk membuat model klasifikasi penerjemah bahasa isyarat dengan menggunakan ensemble algoritma *Random Forest Classifier* dan *AdaBoost*.



Gambar 5. Halaman *Training Dataset*

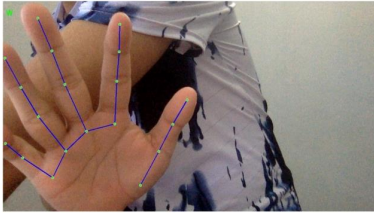
2. Halaman Penerjemahan Bahasa Isyarat.
Halaman yang berisikan fitur penerjemahan bahasa isyarat secara *real time* melalui *webcam* sistem. Pada tampilan ini setiap gerakan bahasa isyarat akan dideteksi dengan ensemble algoritma *Random Forest Classifier* dan *AdaBoost* kemudian diterjemahkan menjadi teks.



Gambar 6. Halaman Penerjemahan Bahasa Isyarat

Berikut ini, Tabel 1 menunjukkan skenario hasil pengujian yang dilakukan.

Tabel 1. Hasil Pengujian

No.	Skenario	Hasil Pengujian	Hasil	Kesimpulan
1	Isyarat Kata "Hello"	Sign Language Recognition 	W	Gagal
..		...	...	...
14	Isyarat Huruf "W"	Sign Language Recognition 	W	Berhasil

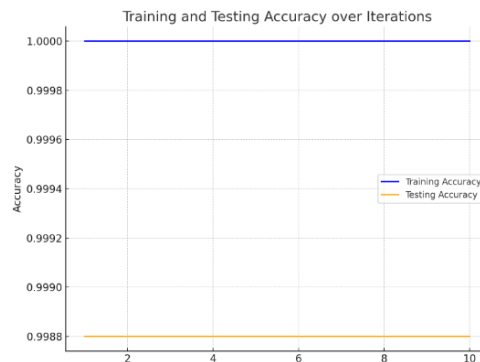
Persentase Akurasi = (Jumlah Berhasil/Total Pengujian) * 100%

= (13/14) * 100%

= (0,92) * 100%

= 92%

Berikutnya dilakukan pengujian berulang sebanyak 10 kali untuk menguji akurasi dari model yang diusulkan dalam penelitian ini. Hasil pengujian ditunjukkan pada Gambar 7 berikut.



Gambar 7. Grafik Hasil Pengujian

Selanjutnya, Tabel 2 menunjukkan hasil pengujian berulang dari model pada penelitian ini.

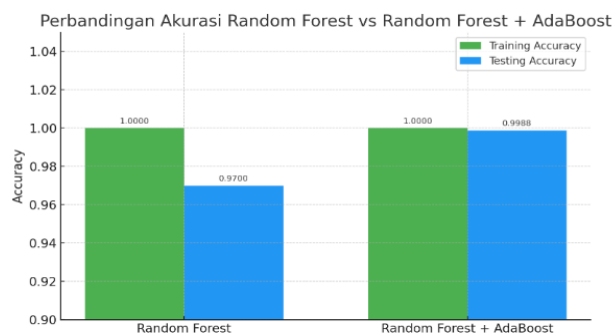
Tabel 2. Hasil Pengujian *Epoch*

Percobaan	Training Accuracy	Testing Accuracy
1	1,0	0,9988
2	1,0	0,9988
3	1,0	0,9988
4	1,0	0,9988
5	1,0	0,9988
6	1,0	0,9988
7	1,0	0,9988
8	1,0	0,9988
9	1,0	0,9988
10	1,0	0,9988

Untuk mengevaluasi efektivitas penggunaan ensemble model, dilakukan perbandingan kinerja antara algoritma *Random Forest* secara tunggal dengan *Random Forest* yang digabungkan dengan *AdaBoost*. Pengujian dilakukan pada dataset yang sama menggunakan parameter pelatihan yang serupa. Perbandingan ini bertujuan untuk mengamati peningkatan akurasi serta kestabilan model dalam proses klasifikasi gestur tangan.

Tabel 3. Hasil Perbandingan Kinerja Model Ensemble

Model	Training Accuracy	Testing Accuracy	Catatan
<i>Random Forest</i>	1,0	0,9700	Akurasi tinggi, namun sedikit fluktuasi pada <i>testing</i>
<i>Random Forest</i> + <i>AdaBoost</i>	1,0	0,9988	Lebih stabil dan konsisten, tingkat generalisasi lebih baik



Gambar 8. Grafik Batang Hasil Perbandingan Akurasi Model Ensemble

Hasil penelitian ini menunjukkan bahwa aplikasi penerjemah bahasa isyarat yang dibangun dengan pendekatan algoritma ensemble mampu bekerja secara efektif dan akurat. Dari total 14 skenario pengujian yang dilakukan pada berbagai isyarat kata dan huruf, sistem berhasil menerjemahkan dengan tepat pada 13 skenario. Hal ini

menghasilkan tingkat akurasi sebesar 92%, yang menunjukkan bahwa model penerjemah sudah cukup andal dalam menerjemahkan gestur tangan ke dalam bentuk teks secara real-time.

Lebih lanjut, untuk memastikan kestabilan dan konsistensi performa model, dilakukan pengujian sebanyak 10 kali terhadap proses pelatihan. Hasil yang didapatkan menunjukkan bahwa baik akurasi pelatihan (*training accuracy*) maupun akurasi pengujian (*testing accuracy*) tetap stabil di angka 1,0 dan 0,9988. Hal ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang sangat baik terhadap data baru yang belum pernah dilihat sebelumnya, tanpa mengalami overfitting.

Dalam tahap lanjutan, dilakukan perbandingan antara algoritma *Random Forest* dan *Random Forest* yang digabungkan dengan *AdaBoost*. Dari hasil pengujian, terlihat bahwa meskipun *Random Forest* saja sudah mampu memberikan hasil yang baik dengan akurasi pengujian sebesar 0,9700, penerapan ensemble model dengan kombinasi *AdaBoost* memberikan peningkatan signifikan dengan akurasi pengujian mencapai 0,9988. Hal ini menunjukkan bahwa penggunaan metode *boosting* seperti *AdaBoost* mampu memperkuat kinerja model dasar (*base learner*) dengan cara fokus pada kesalahan prediksi yang sebelumnya kurang akurat.

Performa model ensemble yang lebih unggul dapat dijelaskan oleh cara kerja *AdaBoost* yang memberikan bobot lebih besar pada data yang sulit diklasifikasi dengan benar oleh model sebelumnya. Dengan demikian, kesalahan yang dilakukan oleh *Random Forest* dapat diminimalkan, sehingga hasil akhir menjadi lebih presisi dan stabil. Hasil ini mendukung bahwa kombinasi *Random Forest* + *AdaBoost* sangat cocok digunakan dalam aplikasi penerjemahan bahasa isyarat yang menuntut akurasi tinggi dan kestabilan dalam waktu nyata (*real-time*).

Secara keseluruhan, hasil penelitian ini membuktikan bahwa penggunaan algoritma ensemble tidak hanya meningkatkan akurasi tetapi juga memperbaiki kekuatan generalisasi model. Aplikasi yang dirancang menunjukkan potensi yang besar untuk diimplementasikan secara nyata dalam membantu komunikasi masyarakat penyandang disabilitas pendengaran dan wicara, khususnya dalam konteks interaksi sehari-hari.

5. Kesimpulan

Adapun kesimpulan yang diuraikan berkaitan dengan hasil penelitian yang didapatkan yaitu: Aplikasi yang dibangun berhasil memfasilitasi komunikasi bagi penyandang disabilitas tunarungu dengan menerjemahkan bahasa isyarat ke dalam teks secara real-time. Berdasarkan pengujian yang dilakukan, aplikasi ini mencapai akurasi 92% pada uji coba skenario penerjemahan isyarat secara langsung dan mencapai 99,88% akurasi pada pengujian model menggunakan kombinasi algoritma *Random Forest* + *AdaBoost*. Jika dibandingkan dengan penggunaan algoritma *Random Forest* saja, yang menghasilkan akurasi pengujian sebesar 97,00%, maka terdapat peningkatan akurasi sebesar 2,97%.

6. Daftar Pustaka

- [1] S. M. Ulfah and S. Ubaidah, "Penerapan Bahasa Isyarat dalam Pembelajaran bagi Anak Berkebutuhan Khusus Tuna Rungu," *J. Disabil. Stud. Res.*, vol. 2, no. 1, pp. 06–23, 2023.
- [2] R. Haris Alfikri, M. Siswo Utomo, H. Februriyanti, and E. Nurwahyudi, "Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode CNN Berbasis Android," *J. Teknoinfo*, vol. 16, no. 2, pp. 183–197, 2022, [Online]. Available: <https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index>.
- [3] S. Nur, A. N. Assyifa, and H. Nurjannah, "Pengembangan Aplikasi Penerjemah Bahasa Isyarat Indonesia (BISINDO) Menggunakan Metode Long-Short Term Memory," *Edusaintek J. Pendidikan, Sains dan Teknol.*, vol. 11, no. 1, pp. 13–30, 2024.
- [4] M. Yunus and Y. Anwar, "Aplikasi Penerjemah Bahasa Isyarat Indonesia Ke Dalam Huruf Abjad," *J. Sintaks Log.*, vol. 2, no. 1, pp. 257–262, 2022, doi: 10.31850/jsilog.v2i1.1726.
- [5] T. He, C. Xie, Q. Liu, S. Guan, and G. Liu, "Evaluation and Comparison of Random Forest and A-LSTM Networks for Large-scale Winter Wheat Identification," *Mdpi*, vol. 11, no. 14, pp. 1–21, 2019.
- [6] V. R. Kakularam and L. R. Parvathy, "An Innovative Analysis of Predicting Melanoma Skin Cancer Using CNN and Random Forest Algorithm," *Adv. Parallel Comput.*, pp. 352–359, 2022, doi: 10.3233/APC220049.
- [7] F. D. Marleny, *Pengolahan Citra Digital Menggunakan Python*. Jawa Tengah: CV. Pena Persada, 2021.
- [8] Muttaqin *et al.*, *Implementasi Artificial Intelligence (AI) Dalam Kehidupan*. Medan: Yayasan Kita Menulis, 2023.
- [9] A. A. Permana *et al.*, *Machine Learning*. Padang: PT Global Eksekutif Teknologi, 2022.
- [10] Amna *et al.*, *Data Mining*. Padang: PT Global Eksekutif Teknologi, 2023.
- [11] M. Peng, C. Wang, T. Chen, G. Liu, and X. Fu, "Dual temporal scale convolutional neural network for micro-expression recognition," *Front. Psychol.*, vol. 8, no. 10, pp. 1–12, 2017, doi: 10.3389/fpsyg.2017.01745.
- [12] B. P. Hartato, "Penerapan Convolutional Neural Network pada Citra Rontgen Paru-Paru untuk Deteksi SARS-CoV-2," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 747–759, 2021, doi: 10.29207/resti.v5i4.3153.
- [13] Andi, Thamrin, A. Susanto, E. Wijaya, and D. Djohan, "Analysis of the random forest and grid search algorithms in early detection of diabetes mellitus disease," *J. Mantik*, vol. 7, no. 2, pp. 2685–4236, 2023, doi: 10.35335/mantik.v7i2.3981.

- [14] F. Diba, M. S. Lydia, and P. Sihombing, “Analisis Random Forest Menggunakan Principal Component Analysis Pada Data Berdimensi Tinggi,” *Indones. J. Comput. Sci.*, vol. 12, no. 4, pp. 2152–2160, 2023, doi: 10.33022/ijcs.v12i4.3329.
- [15] M. R. Raihan, Y. H. Chrisnanto, and A. K. Ningsih, “KLASIFIKASI PENENTUAN KELAYAKAN PINJAMAN KOPERASI DENGAN ALGORITMA CART MENGGUNAKAN ALGORITMA ADABOOST,” *INFOTECH J.*, vol. 8, no. 2, pp. 74–83, 2022.