

Penggunaan Machine Learning Dalam Klasifikasi Bahasa Isyarat BISINDO Menggunakan Kamera

Nico Alexander¹, Romy Budhi Widodo¹, dan Windra Swastika²

¹Pusat Studi Human-Machine Interaction, Teknik Informatika, Universitas Ma Chung,
Villa Puncak Tidar Blok N-01, Kabupaten Malang, Jawa Timur 65151

²Pusat Studi Artificial Intelligence and Technopreneur, Teknik Informatika, Universitas Ma Chung,
Villa Puncak Tidar Blok N-01, Kabupaten Malang, Jawa Timur 65151

Correspondence: Nico Alexander (311910019@student.machung.ac.id)

Received: 04 12 23 - Revised: 18 12 23 - Accepted: 20 12 23 - Published: 25 12 23

Bahasa isyarat digunakan untuk berkomunikasi antara penyandang tunarungu dan tunawicara dengan pendengar. Permasalahan yang ada ialah bahasa isyarat jarang dipahami oleh masyarakat pada umumnya, sedangkan biaya penerjemah bahasa isyarat pada umumnya cukup mahal. Oleh karena itu, penggunaan teknologi Machine Learning dalam mengartikan bahasa isyarat menjadi penting untuk membantu meningkatkan komunikasi antara masyarakat umum dengan penyandang tunarungu dan tunawicara. Selain itu penggunaan teknologi Machine Learning dalam mengartikan bahasa isyarat dapat membuka peluang untuk pengembangan aplikasi cerdas yang lebih canggih, seperti sistem kontrol cerdas, robotika, dan rumah cerdas. Penelitian ini bertujuan untuk membuat prototipe aplikasi penggunaan teknologi Machine Learning dalam mengartikan bahasa isyarat BISINDO menggunakan kamera. Prototipe telah berhasil dibuat menggunakan empat jenis model classifier dari Machine Learning yaitu Random Forest, KNN, SVM, dan Decision Tree. Model-model yang digunakan berhasil mengidentifikasi bahasa isyarat SIBI, walaupun memiliki kekurangan dan masih lama dibandingkan dengan metode ketik. Kinerja model terbaik ialah Random Forest dengan nilai akurasi, presisi, f1, serta recall sebesar 97,7% yang diperoleh dari pengujian model dan 84% presisi secara real time. Serta mampu mencapai standar durasi paling singkat dalam mengklasifikasi gestur yaitu sebesar 34,6 kata per menit. Model yang lain memiliki kinerja yang cukup baik dengan kemampuan kinerja terendah diperoleh dari Decision Tree yaitu sebesar 89,3% untuk nilai akurasi, presisi, f1, dan recall yang diperoleh dari pengujian model. Kemudian memperoleh 75% nilai presisi secara real time dan memperoleh rata rata kecepatan klasifikasi 17.6 kata per menit.

Kata kunci: bahasa isyarat, machine learning, BISINDO, komunikasi, tunarungu, tunawicara

PENDAHULUAN

Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO) adalah dua sistem isyarat yang digunakan di Indonesia. BISINDO dikembangkan oleh penyandang tunarungu Indonesia pada tahun 1980-an, sedangkan SIBI disahkan oleh pemerintah Indonesia pada tahun 2010. Perbedaan utama antara BISINDO dan SIBI terletak pada kosakata dan aturan tata bahasa. BISINDO memiliki kosakata yang lebih sederhana dan aturan tata bahasa yang lebih fleksibel, sedangkan SIBI memiliki kosakata yang lebih kaya dan aturan tata bahasa yang lebih ketat. Namun, keduanya memiliki persamaan dalam hal penggunaan gerakan tangan dan ekspresi wajah untuk mengungkapkan makna (Mursita et al., n.d.). Beberapa penelitian yang menggunakan BISINDO dan SIBI adalah sebagai berikut (Indra Borman & Priyopradono, 2018; Mursita et al., n.d.; Sri Nugraheni et al., n.d.).

Penggunaan teknologi Machine Learning dalam mengartikan bahasa isyarat atau sign language, termasuk bahasa isyarat BISINDO menggunakan kamera menjadi topik penelitian yang menarik. Hal ini disebabkan karena bahasa isyarat merupakan bahasa alami bagi para penyandang tunarungu sedangkan bagi masyarakat biasa, bahasa isyarat bukanlah suatu hal yang umumnya diminati dan dipelajari karena masyarakat sudah terbiasa untuk melakukan komunikasi secara verbal. Karena jarang ditemukan masyarakat yang mengerti akan bahasa isyarat, para kaum tunarungu dan tunawicara mendapat kesulitan sehari – harinya, bahkan akan sulit untuk menyampaikan aspirasi secara langsung.

Namun, bahasa isyarat bukanlah bahasa internasional karena adanya perbedaan ekspresi isyarat antara negara dan wilayah. Hal ini memicu upaya pengembangan teknologi untuk mengartikan bahasa isyarat secara otomatis menggunakan teknologi Machine Learning. Salah satu penelitian yang membahas penggunaan Machine Learning untuk mengartikan bahasa isyarat menggunakan kamera adalah penelitian berjudul penggunaan deep learning dalam mengartikan bahasa isyarat yang ada di Vietnam (Vo et al., 2019).

Penelitian tersebut mengusulkan penggunaan dua jenis fitur (spatial features dan scene-based feature) untuk mengenali bahasa isyarat. Selain metode klasifikasi tradisional seperti SVM, bisa menggunakan Decision Tree, Random Forest, dan lain – lain. Penelitian tersebut menggunakan teknik Machine Learning untuk mengenali bahasa isyarat BISINDO secara akurat. Keunggulan dari teknik deep learning adalah mampu menangkap informasi dari banyak frame dalam sebuah urutan video, sehingga dapat membedakan isyarat dengan akurat.

Pada penelitian lainnya yang dilakukan oleh Nareshkumar dan Jaison pada tahun 2023, dibahas pula pengembangan sistem yang dapat menafsirkan alfabet dalam American Sign Language (ASL) menggunakan teknologi Machine Learning. Penelitian tersebut bertujuan untuk membangun sistem yang dapat memahami bahasa isyarat secara cepat dan membantu dalam komunikasi (Nareshkumar & Jaison, 2023).

Pada penelitian terdahulu (Daniel Adiwirinata Santoso, 2015), hasil yang diberikan memiliki akurasi berkisar dari 53,3% hingga 96,7%, dengan keseluruhan pengujian memiliki akurasi pada 80,67%. Hasil dari *Accuracy* nya adalah sebesar 0.963, *precision* sebesar 0.828, *recall* sebesar 0.8, dan *f1-score* sebesar 0.813 pada variasi model KNN pertama. Pada variasi kedua, hasil yang diperoleh adalah memiliki rentang akurasi berkisar 90% hingga 96.67%, dengan hasil akurasi keseluruhan pada 92.67%. Hasil dari *Accuracy* nya adalah sebesar 0.993, *precision* sebesar 0.9375, *recall* sebesar 1, dan *f1-score* sebesar 0.967. Perbedaan kedua model tersebut terletak pada *parameter tuning* pada model KNN nya yaitu pada model pertama memiliki 1 variasi dengan nilai *k* sebesar 5, sedangkan pada model kedua memiliki 5 variasi dengan *k* sebesar 3.

Penggunaan teknologi Machine Learning dalam mengartikan bahasa isyarat menjadi penting karena membantu meningkatkan komunikasi antara para penyandang tunarungu dan tunawicara dengan pendengar. Selain itu, penggunaan teknologi ini dapat mengurangi ketergantungan pada penerjemah bahasa isyarat manusia, yang pada umumnya memerlukan biaya yang cukup mahal. Di samping itu, penggunaan teknologi Machine Learning dalam mengartikan bahasa isyarat dapat membuka peluang untuk pengembangan aplikasi cerdas yang lebih canggih, seperti sistem kontrol cerdas, robotika, dan rumah cerdas.

MASALAH

Saat ini, deteksi gestur pada bahasa isyarat BISINDO masih dilakukan secara manual, yang tentunya memerlukan waktu dan tenaga yang cukup besar. Oleh karena itu, pengembangan aplikasi deteksi gestur pada bahasa isyarat BISINDO menggunakan teknologi Machine Learning diharapkan dapat memberikan solusi yang lebih efisien dan akurat.

METODE PELAKSANAAN

Metode pelaksanaan penelitian ini mencakup beberapa tahap. Pertama, pengumpulan data dilakukan dengan merekam video gestur bahasa isyarat BISINDO

menggunakan live cam. Partisipan yang mahir dalam bahasa tersebut akan melakukan gerakan yang merepresentasikan angka 0-10, alfabet, dan empat puluh kata sehari-hari sebagai berikut : *air, akan, ambil, anda / kamu, apa, atas, atau, bawah, belajar, berkata, bisa, dalam, dan, dari, dia, hanya, ini, itu, jadi, jika kalau, kerja, kita, luar, makan, mereka, milik / punya, minum, naik, nasi, orang, pergi, pulang, roti, saya, semua, seperti, siapa, tahu / paham, tahun, tapi, turun, untuk*. Data tersebut berupa gambar yang diambil 200 kali pada setiap *class* yang ada, kemudian data akan diolah untuk membentuk dataset pelatihan. Setelah itu, dilakukan pengolahan data dengan menggunakan MediaPipe untuk mendeteksi telapak tangan dalam setiap frame video. Fitur-fitur relevan diekstraksi dari data ini untuk melatih model klasifikasi.

Selanjutnya, implementasi teknologi machine learning dilakukan dengan menggunakan bahasa pemrograman Python. Beberapa model klasifikasi seperti SVM, *Decision Tree*, KNN, dan *Random Forest* akan dikembangkan dan dilatih menggunakan dataset yang telah diproses. Beberapa penelitian yang menggunakan SVM diantaranya (Fajar Pratama et al., n.d.; Lasulika, 2019; Nugroho et al., 2003), *Decision Tree* (Homepage et al., 2023; Irma Purnamasari & Rinaldi Dikananda, 2023; Mir'aatunnas Pratiwi et al., 2022), KNN (Argina, 2020; Sianturi et al., n.d.; Tangguh Admojo, 2020), dan *Random Forest* (Oon Wira Yuda et al., 2022; Ramadhan et al., 2019; Widjiyati, 2021), Berikut adalah penjelasan mengenai model klasifikasi yang digunakan:

1. *Support Vector Machine* (SVM)

Sebuah algoritma dalam machine learning yang digunakan untuk klasifikasi dan regresi. Tujuan utama SVM adalah menemukan hyperplane optimal yang dapat memisahkan dua kelas data dalam ruang fitur sedemikian rupa sehingga margin antara kedua kelas maksimal. Berikut ini adalah parameter yang digunakan:

- C (Parameter penalti untuk error): 1
- kernel (Jenis kernel yang digunakan): 'rbf'
- degree (Derajat kernel polynomial): 3
- gamma (Koefisien kernel): scale atau 1
- coef0 (Koefisien independen dalam kernel): 0
- shrinking (Apakah menggunakan shrinking heuristic): True
- probability (Apakah model menghasilkan estimasi probabilitas): False
- tol (Toleransi pada kriteria berhenti): 1e-3
- cache_size (Ukuran cache dalam memori): 200 (dalam satuan MB)

- `class_weight` (Bobot kelas untuk mengatasi masalah class imbalance): None
- `verbose` (Tingkat verbosity): False
- `max_iter` (Jumlah iterasi maksimal): -1 (no limit)
- `decision_function_shape` (Bentuk fungsi keputusan, misalnya "ovr" (one-vs-rest)): 'ovr'
- `break_ties` (Apakah memutuskan tipe class ketika sebaran perhitungan sebanding): False
- `random_state` (Seed untuk menghasilkan angka acak yang konsisten): None

2. *Decision Tree*

Decision Tree adalah sebuah model prediktif dalam machine learning yang mengambil bentuk struktur pohon keputusan. Pohon tersebut terdiri dari simpul-simpul yang merepresentasikan keputusan atau pengujian pada suatu fitur, cabang-cabang yang merepresentasikan hasil dari keputusan atau pengujian, dan daun-daun yang merepresentasikan label kelas atau nilai prediksi. Tujuan utama dari Decision Tree adalah untuk membuat keputusan dengan mengikuti serangkaian pengujian pada fitur-fitur data. Berikut ini adalah parameter yang digunakan:

- `criterion` (Kriteria yang digunakan untuk mengukur kualitas split pada setiap node): "gini"
- `splitter` (Metode yang digunakan untuk memilih split pada setiap node): "best"
- `max_depth` (Maksimal kedalaman dari pohon): None
- `min_samples_split` (Jumlah minimal sampel yang dibutuhkan agar dapat membagi suatu node): 2
- `min_samples_leaf` (Jumlah minimal sampel yang diperlukan agar suatu node menjadi leaf (daun)): 0
- `max_features` (Jumlah fitur yang dipertimbangkan untuk mencari split terbaik di setiap node): None
- `random_state` (Seed untuk menghasilkan angka acak yang konsisten): None
- `max_leaf_nodes` (Maksimal jumlah leaf (daun) yang diizinkan): None
- `min_impurity_decrease` (Threshold untuk memutuskan apakah suatu split akan dilakukan berdasarkan perbedaan impurity): 0
- `class_weight` (Bobot kelas untuk mengatasi masalah class imbalance): None
- `ccp_alpha` (Parameter cost-complexity pruning): 0

3. *K-Nearest Neighbor (KNN)*

K-Nearest Neighbors adalah algoritma klasifikasi dan regresi yang bekerja dengan cara menentukan label kelas atau nilai prediksi berdasarkan mayoritas tetangga terdekat dari suatu data yang akan diprediksi. Jarak antara data yang akan diprediksi dengan data pelatihan digunakan untuk menentukan tetangga terdekat. Berikut ini adalah parameter yang digunakan:

- `n_neighbors` (Jumlah tetangga yang digunakan untuk prediksi): 5
- `weights` (Bobot yang digunakan dalam prediksi, misalnya "uniform" atau "distance"): uniform
- `algorithm` (Algoritma yang digunakan untuk mencari tetangga terdekat): auto
- `leaf_size` (Ukuran leaf saat menggunakan algoritma BallTree atau KDTree): 30
- `p` (Parameter untuk jenis jarak yang digunakan): 2
- `metric` (Jenis metrik jarak): minkowski
- `metric_params` (Parameter tambahan untuk metrik jarak): None
- `n_jobs` (Jumlah pekerja (thread) yang digunakan untuk pelatihan): None

4. *Random Forest*

Random Forest adalah algoritma ensemble yang menggabungkan beberapa pohon keputusan untuk meningkatkan kinerja dan ketangguhan model. Setiap pohon dalam Random Forest dihasilkan secara acak dan independen, dan hasil prediksi diambil berdasarkan mayoritas suara dari semua pohon. Berikut ini adalah parameter yang digunakan:

- `n_estimators` (Jumlah pohon dalam ensemble): 100
- `criterion` (Kriteria yang digunakan untuk mengukur kualitas split pada setiap node): "gini" (untuk klasifikasi)
- `max_depth` (Maksimal kedalaman dari setiap pohon dalam ensemble): None (unlimited depth)
- `min_samples_split` (Jumlah minimal sampel yang dibutuhkan agar dapat membagi suatu node): 2
- `min_samples_leaf` (Jumlah minimal sampel yang diperlukan agar suatu node menjadi leaf): 1
- `min_weight_fraction_leaf` (Jumlah minimal berat sampel yang diperlukan agar suatu node menjadi leaf): 0.0

- `max_features` (Jumlah fitur yang dipertimbangkan untuk mencari split terbaik di setiap node): `sqrt`
- `max_leaf_nodes` (Maksimal jumlah leaf yang diizinkan): `None`
- `min_impurity_decrease` (Threshold untuk memutuskan apakah suatu split akan dilakukan berdasarkan perbedaan impurity): `0.0`
- `bootstrap` (Apakah sampel yang diambil dalam setiap pohon diambil dengan penggantian atau tidak): `True`
- `oob_score` (Apakah menghitung out-of-bag score atau tidak): `False`
- `n_jobs` (Jumlah pekerja (thread) yang digunakan untuk pelatihan. `None` untuk menggunakan semua yang tersedia): `None`
- `random_state` (Seed untuk menghasilkan angka acak yang konsisten): `None`
- `verbose` (Tingkat verbosity): `0`
- `warm_start` (Apakah menggunakan model yang sudah dilatih sebelumnya untuk memulai pelatihan baru): `False`
- `class_weight` (Bobot kelas untuk mengatasi masalah class imbalance): `None`
- `ccp_alpha` (Parameter cost-complexity pruning): `0`
- `max_samples` (Jumlah maksimal sampel yang diambil untuk pelatihan di setiap pohon): `None`

Prototipe akan diintegrasikan dengan live cam sehingga dapat melakukan klasifikasi secara real time. MediaPipe terus digunakan untuk mendeteksi telapak tangan dalam konteks waktu nyata, sementara model machine learning memberikan prediksi gestur berdasarkan data live cam. Prototipe akan diuji oleh lima responden secara *real-time*. Performa prototipe akan dievaluasi berdasarkan akurasi pengartian bahasa isyarat BISINDO dan waktu yang diperlukan untuk melakukan klasifikasi secara *real-time*. Hasil pengujian akan dianalisis untuk mengevaluasi kemampuan prototipe dalam mengartikan bahasa isyarat BISINDO dalam kondisi penggunaan sehari-hari. Selain itu, perbandingan waktu antara penggunaan prototipe dengan metode tradisional berbasis teks akan dievaluasi. Dengan metode pelaksanaan ini, diharapkan penelitian ini dapat memberikan kontribusi dalam meningkatkan aksesibilitas bagi penyandang tunarungu dalam pembelajaran dan interaksi sosial mereka.

HASIL DAN PEMBAHASAN

Dalam penelitian ini, hasil yang didapatkan didasarkan dari performa *f1*, *recall*, *precision*, dan *accuracy*. Berikut adalah tabel yang menunjukkan performa tersebut untuk 4 classifier yaitu SVM, Random Forest, KNN, dan Decision Tree.

Tabel 1 Hasil penilaian kinerja dari 4 model Machine Learning

Classifier	<i>f1</i>	<i>recall</i>	<i>precision</i>	Akurasi
SVM	0.961	0.961	0.961	0.961
<i>Random Forest</i>	0.979	0.979	0.979	0.979
KNN	0.966	0.966	0.966	0.966
<i>Decision Tree</i>	0.893	0.893	0.893	0.893

Serta berikut adalah tabel 2 hingga tabel 5 yang menunjukkan hasil dari Confusion Matrix setiap model:

Tabel 2 Confusion Matrix BISINDO SVM

Label	1	10	3	4	...	W	X	Y	Z
1	171	0	0	0	...	0	0	0	0
10	0	180	0	0	...	0	0	0	0
3	0	0	179	0	...	0	0	0	0
4	0	0	0	176	...	0	0	0	0
...
W	0	0	0	0	...	184	0	0	0
X	0	0	0	0	...	0	192	0	0
Y	0	0	0	0	...	0	0	200	0
Z	0	0	0	0	...	0	0	0	204

Tabel 3 Confusion Matrix BISINDO *Random Forest*

Label	1	10	3	4	...	W	X	Y	Z
1	189	0	0	0	...	0	0	0	0
10	0	181	0	0	...	0	0	0	0
3	0	0	192	0	...	0	0	0	0

4	0	0	1	174	...	0	0	0	0
...
W	0	0	0	0	...	183	0	0	0
X	0	0	0	0	...	0	188	0	0
Y	0	0	0	0	...	0	0	200	0
Z	0	0	0	0	...	0	0	0	203

Tabel 4 Confusion Matrix BISINDO KNN

Label	1	10	3	4	...	W	X	Y	Z
1	181	0	0	0	...	0	0	0	0
10	0	174	0	0	...	2	0	0	0
3	0	0	189	1	...	0	0	0	0
4	0	0	5	168	...	0	0	0	0
...
W	0	0	0	0	...	183	0	0	0
X	0	0	0	0	...	0	188	0	0
Y	0	0	0	0	...	0	0	200	0
Z	0	0	0	0	...	0	0	0	203

Tabel 5 Confusion Matrix BISINDO *Decision Tree*

Label	1	10	3	4	...	W	X	Y	Z
1	172	0	0	0	...	0	0	0	2
10	0	163	0	0	...	0	0	1	0
3	0	0	180	2	...	0	0	0	0
4	0	0	2	159	...	0	0	0	0
...
W	0	1	0	0	...	165	0	0	0
X	0	1	0	0	...	2	158	1	0
Y	0	5	0	0	...	0	0	186	0

Z	1	0	0	1	...	0	0	0	189
---	---	---	---	---	-----	---	---	---	-----

Tabel 1 menunjukkan hasil kinerja dari empat model meliputi f1, recall, akurasi, dan presisi yang diperoleh dari uji program dengan rumus sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = \frac{2 \times P \times R}{P + R}$$

Keterangan:

TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

FN = *False Negative*

P = *Precision*

R = *Recall*

Pada tabel tersebut dapat dilihat bahwa kinerja terbaik diperoleh dari klasifikasi menggunakan Random Forest dengan nilai akurasi f1, recall, dan Precision sebesar 0.979 atau setara 97,9%. Kemudian nilai akurasi terbaik berikutnya didapatkan pada model KNN dengan nilai presisi sebesar 0.966 atau setara 96,6%, kemudian SVM dengan presisi sebesar 0.961 atau setara 96.1 %, kemudian nilai presisi terendah dimiliki oleh Decision Tree dengan nilai sebesar 0.893 atau setara 89,3%.

Dari penjelasan keempat model di atas, sudah dapat dilihat bahwa Random Forest memiliki kinerja paling baik, dengan nilai akurasi, presisi, f1, dan recall sebesar 0.979 menunjukkan bahwa Random Forest mampu memberikan hasil prediksi yang paling akurat, mampu menunjukkan bahwa sebagian besar entitas yang dinyatakan sebagai relevan, hampir tidak melewatkan entitas yang sebenarnya ada dalam data, serta mampu menghasilkan hasil klasifikasi yang sangat baik secara keseluruhan, dengan keseimbangan yang sangat baik

antara kemampuan untuk mengidentifikasi entitas yang relevan (recall) dan kemampuan untuk menghindari kesalahan pengklasifikasian (precision)

Tabel 2 hingga 5 merepresentasikan Confusion matrix dari 4 model saat dilakukan prediksi dari data test. Dimana sumbu x label merupakan label data train, dan sumbu y merupakan label dari data test. Kemudian setiap kolom yang memiliki warna orange adalah kolom yang menunjukkan jumlah prediksi benar antara data train dan data test. Tabel Confusion matrix inilah yang digunakan sebagai perhitungan dari tingkat presisi sebelumnya.

Tabel 2 hingga 5 adalah potongan tabel yang merepresentasikan confusion matrix, namun dikarenakan ukuran tabel yang terlalu besar (78 x 78) maka pada tabel 2 hingga 5 ditampilkan potongannya saja. Pada tabel 2 hingga 5 yang utuh dapat dilihat bahwa terdapat beberapa kesamaan dimana model cukup kesulitan mendeteksi huruf “C” dan “T”, kata “BELAJAR” dan “MEREKA” dikarenakan memiliki banyak data yang dihapus saat proses preprocessing, sehingga hasil data yang dimiliki oleh class tersebut cenderung lebih sedikit daripada class yang lainnya.

Selain kinerja secara fl, recall, akurasi, dan presisi, model akan diuji secara *realtime* yang akan dinilai berdasar seberapa efektif dan akurat aplikasi dalam melakukan klasifikasi gestur secara real time. Penilaian yang pertama akan dilakukan dengan cara melakukan percobaan klasifikasi pada seluruh class sebanyak 5 kali, dan akan dihitung rata rata keberhasilan pada setiap class dari setiap model. Hasil dari uji diatas dapat dilihat pada tabel 6.

Kemudian penilaian kedua adalah dengan cara penguji akan merangkai beberapa buah kalimat dengan menggunakan aplikasi dari setiap model dan akan dibandingkan kecepatannya dengan cara mengetik kalimat tersebut secara manual menggunakan perangkat smartphone yang diukur menggunakan stopwatch yang hasilnya dapat dilihat pada tabel 7.

Tabel 6 Hasil perhitungan presisi *real time* dari 4 model

Rata-rata			
<i>Random Forest</i>	KNN	SVM	<i>Decision Tree</i>
84%	83%	80%	75%

Tabel 7 Hasil pengukuran durasi 4 model dan ketik secara *real time*

Kalimat	Saya Nico 23 Tahun			
<i>Random Forest</i>	KNN	SVM	<i>Decision Tree</i>	Ketik
14 detik	17 detik	18 detik	25 detik	5 detik
Kalimat	Kamu belajar apa			
<i>Random Forest</i>	KNN	SVM	<i>Decision Tree</i>	Ketik
4 detik	5 detik	6 detik	10 detik	3 detik
Kalimat	Saya minum 1 air makan 2 nasi			
<i>Random Forest</i>	KNN	SVM	<i>Decision Tree</i>	Ketik
8 detik	10 detik	11 detik	16 detik	6 detik

Pada tabel 6 dan 7 dapat dilihat bahwa saat dilakukan pengujian secara langsung, presisi terbaik dimiliki oleh *Random Forest* 84%, kemudian ada KNN sebesar 83%, SVM sebesar 80%, dan yang paling rendah dimiliki oleh *Decision Tree* yaitu sebesar 75%. Pada pengukuran durasi dalam mengutarakan sebuah 3 buah kalimat antara 4 model, durasi paling singkat dimiliki oleh *Random Forest*, namun meskipun demikian metode ketik memiliki waktu yang lebih cepat dibandingkan dengan seluruh model yang ada.

Tabel 8 Jumlah gestur dinamis dan statis

Jumlah Gestur	
Statis	Dinamis
47	30

Tabel 9 Rata-rata akurasi gestur statis

Rata-rata akurasi gestur statis			
Random Forest	KNN	SVM	Decision Tree
94%	89%	88%	83%

Tabel 10 Rata-rata akurasi gestur dinamis

Rata-rata akurasi gestur statis			
Random Forest	KNN	SVM	Decision Tree
70%	75%	67%	63%

Selain itu, dapat dilihat pada tabel 8 hingga tabel 10 dimana terdapat 3 tabel yang menunjukkan jumlah gestur dinamis dan statis, beserta rata-rata akurasi untuk gestur statis dan dinamis. bahwa kinerja dari setiap model lebih bagus saat mencoba mendeteksi gestur statis seperti angka, mayoritas alfabet, dan beberapa gestur corpus. Sedangkan untuk gestur yang dinamis seperti huruf “G” dan “R”, dan mayoritas corpus seperti “NAIK, TURUN, KITA” dan masih banyak corpus dinamis lainnya. Hal ini dikarenakan cara kerja dari prototipe yang mengambil satu *frame* dan langsung diprediksi tanpa melihat rangkaian gerakan yang dibuat oleh pengguna.

Alasan lain mengapa ada kesalahan klasifikasi adalah karena adanya kemiripan koordinat dari beberapa gestur, seperti pada contoh gestur huruf “G” dan kata “TAHUN” seperti pada tabel 11. Warna hijau pada tabel 11 menandakan kemiripan kedua gestur tersebut.

Tabel 11 Fitur huruf “G” dan kata “Tahun”

Gestur	G																				
Tangan Kiri	0.365	0.396	0.438	0.470	0.468	0.398	0.430	0.449	0.453	0.393	0.440	0.456	0.453	0.393	0.441	0.455	0.450	0.396	0.439	0.452	0.448
Tangan Kanan	0.602	0.557	0.542	0.560	0.600	0.552	0.549	0.556	0.567	0.592	0.606	0.602	0.599	0.628	0.644	0.632	0.621	0.659	0.672	0.658	0.645
Tangan Kiri	0.000	-0.007	-0.015	-0.022	-0.027	-0.032	-0.043	-0.046	-0.045	-0.033	-0.039	-0.036	-0.034	-0.041	-0.036	-0.030	-0.035	-0.036	-0.030	-0.026	
Tangan Kanan	0.489	0.461	0.421	0.385	0.359	0.416	0.373	0.367	0.369	0.415	0.367	0.369	0.380	0.416	0.371	0.377	0.391	0.420	0.380	0.382	0.394
Tangan Kanan	0.409	0.359	0.337	0.338	0.352	0.335	0.336	0.352	0.362	0.370	0.375	0.386	0.390	0.409	0.413	0.418	0.418	0.447	0.451	0.450	0.447
Tangan Kanan	0.000	0.006	0.005	0.002	-0.001	-0.010	-0.018	-0.021	-0.021	-0.017	-0.021	-0.018	-0.017	-0.022	-0.024	-0.016	-0.011	-0.026	-0.026	-0.018	-0.012

Gestur	TAHUN																				
Tangan Kiri	0.365	0.377	0.411	0.444	0.466	0.421	0.478	0.468	0.448	0.425	0.480	0.464	0.443	0.427	0.479	0.462	0.441	0.427	0.466	0.452	0.434
Tangan Kanan	0.486	0.441	0.406	0.400	0.417	0.393	0.403	0.415	0.415	0.428	0.442	0.454	0.453	0.467	0.478	0.486	0.484	0.507	0.514	0.516	0.514
Tangan Kiri	0.000	-0.004	-0.012	-0.019	-0.024	-0.024	-0.038	-0.043	-0.044	-0.029	-0.039	-0.036	-0.032	-0.033	-0.040	-0.030	-0.022	-0.039	-0.042	-0.039	-0.026
Tangan Kanan	0.540	0.512	0.478	0.452	0.437	0.442	0.419	0.426	0.437	0.442	0.424	0.436	0.451	0.448	0.433	0.445	0.458	0.458	0.446	0.456	0.467
Tangan Kanan	0.565	0.521	0.498	0.502	0.516	0.534	0.534	0.532	0.530	0.565	0.564	0.562	0.561	0.594	0.591	0.588	0.587	0.620	0.618	0.614	0.612
Tangan Kanan	0.000	0.003	0.004	0.004	0.004	0.005	0.003	-0.001	-0.003	0.001	0.003	0.002	0.001	-0.002	0.001	0.003	0.004	-0.004	-0.001	0.004	0.008

Dapat dilihat pada tabel 11 yang menunjukkan fitur atau koordinat landmark dari gestur huruf “G” dan kata “TAHUN”. Gestur dari huruf “G” dan kata “TAHUN” sama-sama dibuat menggunakan 2 tangan, dimana artinya sama-sama memiliki 126 fitur. Fitur – fitur tersebut didapat dari ekstraksi subkoordinat *landmark* melalui mediapipe untuk dua tangan, dimana setiap tangan memiliki 63 subkoordinat *landmark*. Kesemua 63 subkoordinat tersebut diperoleh dari 3 x 21 yaitu 3 subkoordinat (x,y,z) dan 21 *landmark*. Dari 126 fitur yang dimiliki oleh kedua class tersebut, lebih dari setengah atau lebih tepatnya 68 koordinat yang dimiliki oleh kedua class tersebut cukup mirip yang ditandai dengan kolom berwarna hijau. Hal

ini akan menyebabkan kesalahan atau berkurangnya keakuratan prototipe saat melakukan klasifikasi.

Dapat disimpulkan dari tabel 1 mengenai presisi model, tabel 2 hingga tabel 5 mengenai confusion matrix setiap model tabel 6 dan tabel 7 mengenai pengujian real time dari 4 model bahwa model yang memiliki presisi terbaik dan durasi tersingkat dimiliki oleh Random Forest. Akan tetapi tetap akan lebih cepat menggunakan metode ketik pada gadget smartphone.

KESIMPULAN

Berdasar dari tujuan dan batasan masalah yang sudah ditentukan padab Bagian pendahuluan, dimana penelitian ini ditujukan untuk membuat sebuah prototipe klasifikasi bahasa isyarat BISINDO telah berhasil dilakukan. Prototipe telah berhasil dibuat menggunakan 4 jenis model classifier dari Machine Learning yaitu Random Forest, KNN, SVM, dan Decision Tree. Setiap model dapat mengidentifikasi bahasa isyarat BISINDO, walaupun masih memiliki beberapa kekurangan dan masih lebih lama dibandingkan dengan metode ketik. Kinerja model terbaik diperoleh menggunakan Random Forest yaitu dengan nilai akurasi, presisi, f1, serta recall sebesar 97,7% yang didapatkan saat pengujian model dan 84% presisi secara real time. Serta mampu mencapai standar durasi paling singkat dalam mengklasifikasi gestur yaitu sebesar 34,6 kata per menit. Model yang lain walaupun tidak memiliki kinerja sebaik Random Forest, namun kemampuan diperoleh dari Decision Tree sudah cukup baik yaitu sebesar 89,3% untuk nilai akurasi, presisi, f1, dan recall secara program. Kemudian Decision Tree memperoleh 75% nilai presisi secara *real-time* dan memperoleh rata rata kecepatan klasifikasi 17.6 kata per menit.

UCAPAN TERIMA KASIH

Ucapan terimakasih disampaikan oleh penulis kepada Human-Machine Interaction sebagai tempat pusat studi saya melaksanakan penelitian. Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi, atas dukungan dana pada Riset Terapan, Skema Penelitian: Penelitian Terapan-Jalur Hilirisasi 2023. Kepada keluarga penulis yang telah memberikan dukungan moral. Kepada teman teman penulis yang sudah memberikan bantuan berupa partisipasi dan *support*.

DAFTAR PUSTAKA

- Argina, A. M. (2020). *Indonesian Journal of Data and Science Penerapan Metode Klasifikasi K-Nearest Neighbor pada Dataset Penderita Penyakit Diabetes*. 1(2), 29–33.
- Daniel Adiwiranata Santoso. (2015). *Rancang Bangun Aplikasi Klasifikasi Gestur Tangan Menggunakan K-Nearest Neighbors*.
- Fajar Pratama, R., Andraini, L., & Komputer, T. (n.d.). Menerapkan Algoritma Support Vector Machine (SVM) di Klasifikasi Masyarakat Tanjung Lowland di Lampung Timur. In *Portaldata.org* (Vol. 2, Issue 10).
- Homepage, J., Septhya, D., Rahayu, K., Rabbani, S., Fitria, V., Irawan, Y., & Hayami, R. (2023). *MALCOM: Indonesian Journal of Machine Learning and Computer Science Implementation of Decision Tree Algorithm and Support Vector Machine for Lung Cancer Classification Implementasi Algoritma Decision Tree dan Support Vector Machine untuk Klasifikasi Penyakit Kanker Paru*. 3, 15–19.
- Indra Borman, R., & Priyopradono, B. (2018). Implementasi Penerjemah Bahasa Isyarat Pada Bahasa Isyarat Indonesia (BISINDO) Dengan Metode Principal Component Analysis (PCA). *Z. A. Pagar Alam*, 03(1).
- Irma Purnamasari, A., & Rinaldi Dikananda, A. (2023). Klasifikasi Kualitas Berita Pada Majalah Menggunakan Metode Decision Tree. *Jurnal Teknologi Ilmu Komputer*, 1(2), 48–54. <https://doi.org/10.56854/jtik.v1i2.52>
- Lasulika, M. E. (2019). Komparasi Naïve Bayes, Support Vector Machine Dan K-Nearest Neighbor Untuk Mengetahui Akurasi Tertinggi Pada Prediksi Kelancaran Pembayaran Tv Kabel. *ILKOM Jurnal Ilmiah*, 11(1), 11–16. <https://doi.org/10.33096/ilkom.v11i1.408.11-16>
- Mir'aatunnas Pratiwi, U., Ibad, M., Epidemiologi Dan Biostatistika, B., S1, P., Masyarakat, K., Kesehatan, F., Nahdlatul, U., & Surabaya, U. (2022). *Klasifikasi Faktor Yang Berpengaruh Dalam Kehamilan Tidak Diinginkan Menggunakan Metode Algoritma Decision Tree*. 3(2). <https://doi.org/10.46306/lb.v3i2>
- Mursita, R. A., Tunarungu, R., Pascasarjana, M., & Upi Bandung, P. (n.d.). *Respon Tunarungu Terhadap Penggunaan Sistem Bahasa Isyarat Indonesia (SIBI) Dan Bahasa Isyarat Indonesia (BISINDO) Dalam Komunikasi Rohmah Ageng Mursita*. <http://www.change.org/id/petisi/>
- Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). *Support Vector Machine-Teori dan Aplikasinya dalam Bioinformatika 1*. <http://asnugroho.net>
- Oon Wira Yuda, Darmawan Tuti, Lim Sheih Yee, & Susanti. (2022). Penerapan Penerapan Data Mining Untuk Klasifikasi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Random Forest. *SATIN - Sains Dan Teknologi Informasi*, 8(2), 122–131. <https://doi.org/10.33372/stn.v8i2.885>

- Ramadhan, A., Susetyo, B., & Indahwati. (2019). Penerapan Metode Klasifikasi Random Forest Dalam Mengidentifikasi Faktor Penting Penilaian Mutu Pendidikan. *Jurnal Pendidikan Dan Kebudayaan*, 4(2), 169–182. <https://doi.org/10.24832/jpnk.v4i2.1327>
- Sianturi, F. A., Simangunsong, A., Simanjorang, R. M., Sijabat, P. I., Informatika, M., & Nusantara, P. (n.d.). *Fricles Ariwisanto Sianturi, Jurnal Media Informatika [JUMIN] Implementasi Algoritma Modified Nearest Neighbourt (M-KNN) Untuk Klasifikasi Buku*.
- Sri Nugraheni, A., Pratiwi Husain, A., & Unayah, H. (n.d.). *Optimalisasi Penggunaan Bahasa Isyarat Dengan SIBI Dan BISINDO Pada Mahasiswa Difabel Tunarungu Di Prodi Pgmi Uin Sunan Kalijaga*.
- Tangguh Admojo, F. (2020). *Indonesian Journal of Data and Science Klasifikasi Aroma Alkohol Menggunakan Metode KNN*. 1(2), 34–38.
- Widjiyati, N. (2021). Implementasi Algoritme Random Forest Pada Klasifikasi Dataset Credit Approval. *Jurnal Janitra Informatika Dan Sistem Informasi*, 1(1), 1–7. <https://doi.org/10.25008/janitra.v1i1.118>

