

Analisis Perbandingan Tingkat Performansi Metode GNB, RF, DT, KNN, dan SVM pada Klasifikasi Citra Tangan

Mohammad Farid Naufal^{1*}, Subrata², Marcella Audi Susanto³, Alvin Fernando Susanto⁴, Victor Manuel Soesilo⁵, Rony Hartono Irawan⁶

¹²³⁴⁵⁶Program Studi Teknik Informatika, Universitas Surabaya, Surabaya, Jawa Timur

Email: ^{1*}faridnaufal@staff.ubaya.ac.id, ²s160420002@student.ubaya.ac.id, ³s160420004@student.ubaya.ac.id, ⁴s160420013@student.ubaya.ac.id, ⁵s160420035@student.ubaya.ac.id, ⁶s160420041@student.ubaya.ac.id

Abstrak

Tangan merupakan organ tubuh yang kerap digunakan untuk berbagai aktivitas dan menghasilkan pola-pola tertentu. Pada penelitian ini, kami menerapkan algoritma *Gaussian Naïve Bayes* (GNB), *Random Forest* (RF), *Decision Tree* (DT), *K-Nearest Neighbors* (KNN), dan *Support Vector Machine* (SVM) untuk mengklasifikasikan citra tangan dalam berbagai pola. Tujuan dari penelitian ini adalah membandingkan tingkat performa dari lima algoritma tersebut. Kami menggunakan *K-Fold Cross Validation* dengan nilai parameter $cv=5$ untuk memvalidasi model kami. Kami menggunakan sebanyak 3 *dataset* citra tangan, yakni *American Sign Language*, *Hand Gesture*, dan *Rock-Paper-Scissors*. Masing-masing *dataset* tersebut akan diolah pada tahap *preprocessing* (*smoothing*, *sharpening*, dan *resize*) guna menghasilkan citra yang lebih bersih dan lebih baik untuk diolah. Dari hasil uji coba, kami memperoleh algoritma terbaik untuk *dataset American Sign Language* adalah KNN dan SVM. Sedangkan algoritma terbaik untuk *dataset Rock-Paper-Scissors* adalah SVM, dan algoritma terbaik untuk *dataset Hand Gesture* adalah KNN.

Kata Kunci: klasifikasi, citra, tangan, algoritma, machine learning.

Comparative Analysis of Performance Levels of GNB, RF, DT, KNN, and SVM Methods on Hand Image Classification

Abstract

Hands are organs of the body that are often used for various activities and produce certain patterns. In this study, we apply *Gaussian Naïve Bayes* (GNB), *Random Forest* (RF), *Decision Tree* (DT), *K-Nearest Neighbors* (KNN) algorithms, and *Support Vector Machine* (SVM) to classify hand images in various patterns. The purpose of this study is to compare the performance levels of the five algorithms. We use *K-Fold Cross Validation* with parameter value $cv=5$ to validate our model. We used 3 datasets of hand imagery, namely *American Sign Language*, *Hand Gesture*, and *Rock-Paper-Scissors*. Each of these datasets will be processed at the preprocessing stage (*smoothing*, *sharpening*, and *resizing*) to produce a cleaner and better image for processing. From the test results, we obtained the best algorithms for the *American Sign Language* dataset are KNN and SVM. Meanwhile, the best algorithm for *Rock-Paper-Scissors* dataset is SVM, and the best algorithm for *Hand Gesture* dataset is KNN.

Keywords: classification, image, hand, algorithm, machine learning.

I. PENDAHULUAN

1.1. Latar Belakang

Tangan merupakan organ tubuh yang paling sering digunakan untuk melakukan berbagai aktivitas, seperti bermain gunting-batu-kertas, bahasa isyarat, gestur tubuh, dan

masih banyak lagi. Dalam melakukan aktivitas-aktivitas tersebut, tentunya terdapat berbagai pola tangan yang terbentuk bergantung pada tujuannya. Sebagai manusia yang berakal, akan mudah bagi kita untuk mengenali sekaligus memahami maksud dari masing-masing pola tersebut. Akan

tetapi, manusia juga membutuhkan alat yang nantinya diharapkan dapat membantu mereka mengenali pola-pola tersebut ketika mereka merasa tidak tahu. Untuk itu, *machine learning* akan membantu mewujudkan hal tersebut.

Bersesuaian dengan namanya, *machine learning* merupakan aplikasi komputer yang menerapkan algoritma matematika guna mempelajari sekumpulan data untuk menghasilkan prediksi sesuai dengan tujuan tertentu (Roihan et al., 2020). *Machine learning* memiliki berbagai metode yang dapat digunakan sesuai dengan masing-masing pengguna. Dalam penelitian ini, penulis menggunakan 5 metode untuk melakukan klasifikasi citra tangan yang meliputi *American Sign Language*, *Hand Gesture*, dan *Rock-Paper-Scissors*. Kelima metode tersebut diantaranya adalah *Gaussian Naïve-Bayes* (NB), *Random Forest* (RF), *Decision Tree* (DT), *K-Nearest Neighbors* (KNN), dan *Support Vector Machine* (SVM).

1.2. Tinjauan Literature

Ditinjau dari penelitian terdahulu, terdapat beberapa penelitian yang masih berkaitan dengan jurnal ini. Berdasarkan penelitian Parlindungan & Rizaldi (2020) dengan judul “Identifikasi dan Klasifikasi Gestur Tangan dengan Sinyal EMG”, didapatkan hasil bahwa metode SVM dan RF menghasilkan nilai yang relatif sama. Sedangkan hasil terbaik yang diperoleh dalam penelitian tersebut adalah metode CNN yang mana tidak digunakan dalam penelitian ini.

Pada penelitian kedua yang dilakukan oleh Gunawan & Putra (n.d.) yang berjudul “Perbandingan Identifikasi Penggunaan American Sign Language Menggunakan Klasifikasi Multi-Class SVM, Backpropagation Neural Network, K-Nearest Neighbor dan Naive Bayes”, ditemukan hasil bahwa keempat metode yang digunakan menghasilkan nilai diatas 90%. Sementara itu, hasil tertinggi diperoleh melalui metode Multi-Class SVM.

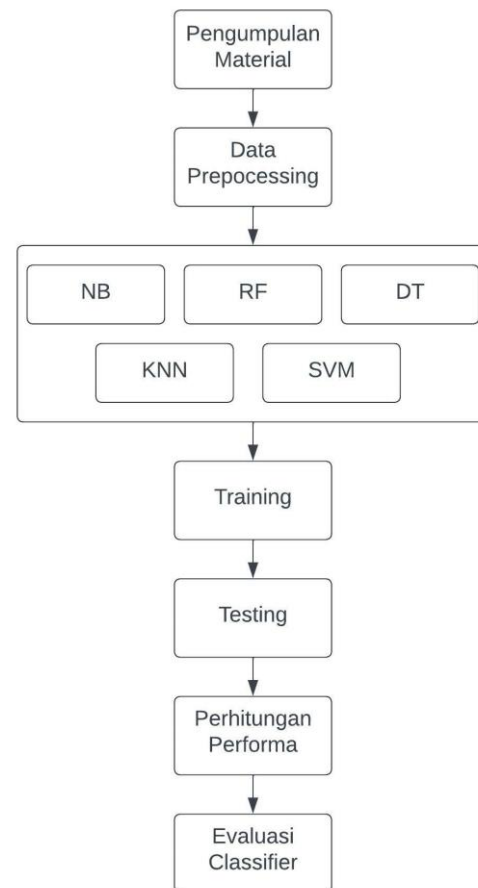
1.3. Pernyataan Masalah

Akurasi dari banyak *classifier* yang digunakan akan menghasilkan penurunan nilai semakin sering *classifier* tersebut dipakai. Pada suatu waktu, *classifier* juga tidak dapat mengklasifikasikan gambar selalu tepat. Hal ini disebabkan oleh adanya satu atau lebih fitur yang bertentangan, baik dari segi bentuk, background, maupun hal-hal lainnya.

Untuk mengatasi masalah tersebut pada ketiga dataset citra tangan yang kami gunakan, maka dilakukan *data preprocessing* untuk meningkatkan akurasi prediksi dari *classifier*. *Classifier* yang akan digunakan adalah 5 *classifier* yang telah disebutkan pada bagian sebelumnya. Sementara itu, ada beberapa label yang dimiliki oleh masing-masing dataset. Pada dataset *American Sign Language*, label yang digunakan adalah angka dari 0-9 dan huruf a-z. Pada dataset *Hand Gestures*, label yang digunakan adalah “*closed fist*”, “*finger circle*”, “*finger symbols*”, “*multi finger bend*”, “*open palm*”, “*semi open palm*”, “*semi open fist*”, dan “*single finger bend*”. Sementara itu, label yang digunakan pada dataset *Rock-Paper-Scissors* adalah “*rock*”, “*paper*”, dan “*scissors*”.

II. METODE PENELITIAN

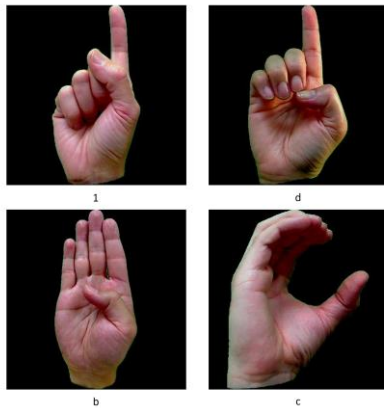
Proses metodologi penelitian yang dilakukan dalam penelitian ini dimulai dari pengumpulan material (dataset), *preprocessing* data, pembentukan *classifier*, melakukan validasi menggunakan *K-Fold Cross Validation*, melakukan *training* pada *classifier*, dan melakukan *testing*.



Gambar 1. Alur Metode Penelitian

2.1. Pengumpulan Material

Material yang digunakan dalam penelitian ini adalah 3 buah dataset, yakni *American Sign Language*, *Hand Gesture*, dan *Rock-Paper-Scissors*. Pada dataset *American Sign Language*, label yang digunakan adalah angka dari 0-9 dan huruf a-z dengan total 2.515 gambar. Pada dataset *Hand Gestures*, label yang digunakan adalah “*closed fist*”, “*finger circle*”, “*finger symbols*”, “*multi finger bend*”, “*open palm*”, “*semi open palm*”, “*semi open fist*”, dan “*single finger bend*” dengan total 2.167 gambar. Sementara itu, label yang digunakan pada dataset *Rock-Paper-Scissors* adalah “*rock*”, “*paper*”, dan “*scissors*” dengan total 2.251 gambar.

Gambar 1. Contoh citra *dataset American Sign Language*Gambar 2. Contoh citra *dataset Hand Gesture*Gambar 3. Contoh citra *dataset Rock-Paper-Scissors*

2.2. Preprocessing

Tahap *preprocessing* dilakukan untuk pada setiap gambar dalam *dataset* guna meningkatkan performa hasil dari masing-masing *classifier*. Jenis *preprocessing* yang dilakukan dalam ketiga *dataset* yang digunakan adalah *smoothing*, *sharpening*, dan *resize*.

Pada *smoothing*, metode yang digunakan adalah *Gaussian Blur* dengan kernel 5x5 untuk menghilangkan *noise* pada gambar.

Selanjutnya, dilakukan proses *sharpening* dengan fungsi *filter2D()* yang terdapat pada *OpenCV library*. Hal ini dilakukan untuk mengembalikan detail yang hilang setelah proses *smoothing*. Kernel yang digunakan adalah matriks berukuran 3x3 sebagai berikut.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Proses terakhir adalah *resize*, dimana seluruh gambar dalam ketiga *dataset* diubah kedalam ukuran 128x128.

Pada *preprocessing* ketiga *dataset* tidak digunakan *segmentation* dengan pertimbangan bahwa *background* dari keseluruhan gambar pada masing-masing *dataset* adalah sama.

Setelah selesai dilakukan *preprocessing*, keseluruhan gambar dalam *dataset* dipecah ke dalam data *training* dan data *testing*. Fungsi yang digunakan adalah *train_test_split()* yang terdapat pada *library* Sklearn dengan detail parameter sebagai berikut.

Tabel 1. Parameter Split

Parameter	Detail
Random_state	21
Test_size	0.3

Proses *splitting* bertujuan untuk membagi *dataset* ke dalam data *training* dan data *testing*. Besar pembagian tersebut adalah 70% untuk data *training* dan 30% untuk data *testing*.

2.3. Pembentukan Model Klasifikasi

Pada penelitian ini, digunakan 6 metode *machine learning* yang berbeda untuk melakukan klasifikasi citra. Metode yang digunakan adalah *Gaussian Naïve-Bayes* (NB), *Random Forest* (RF), *Decision Tree* (DT), *K-Nearest Neighbors* (KNN), dan *Support Vector Machine* (SVM).

Pada algoritma *Gaussian Naïve-Bayes* (NB), parameter yang digunakan adalah *default* yang berisi *prior probability*, kemudian dilakukan *smoothing* dengan default float 1e-9 dengan tujuan stabilisasi kalkulasi.

Pada algoritma *Random Forest* (RF), parameter yang digunakan adalah *n_estimator*. *N_estimator* menjelaskan banyak *tree* yang terbentuk dari serangkaian *dataset* (dalam *forest*). Banyak *tree* yang digunakan adalah 100.

Tabel 2. Parameter RF

Parameter	Detail
N_estimators	100

Pada algoritma *Decision Tree* (DT), parameter yang digunakan adalah *criterion*, *random_state*, *max_depth*, dan *min_samples_leaf*. *Criterion* merupakan metode yang digunakan untuk mengukur kualitas saat melakukan *splitting*. Adapun nilai yang didukung dalam metode ini yaitu “gini” untuk impuritas Gini dan “entropy” untuk perolehan informasi. *Random_state* adalah metode yang mengontrol nilai acak pada estimator, dimana pengacakan dilakukan sebanyak 100 kali. *Max_depth* merupakan metode ini mengatur jumlah maksimum kedalaman *tree*. *Min_samples_leaf* adalah sampel minimal yang diperlukan untuk menjadi *leaf node*.

Tabel 3. Parameter DT

Parameter	Detail
Criterion	“gini”, “entropy”
Random_state	100
Max_depth	3

Min_samples_leaf	5
------------------	---

Pada algoritma *K-Nearest Neighbors* (KNN), parameter yang digunakan adalah *n_neighbors* dan *metric*. *N_neighbors* adalah parameter yang menjelaskan banyak titik (*neighbors*) terdekat yang berada di sekitar titik yang sedang diuji. *Metric* merupakan metode perhitungan jarak antar dua buah titik yang digunakan. Metode yang digunakan pada KNN penulis adalah euclidean, yakni mengukur jarak dua buah titik dengan menggunakan rumus sebagai berikut.

$$d = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

Tabel 4. Parameter KNN

Parameter	Detail
N_neighbors	1-16
Metric	Euclidean

Pada algoritma *Support Vector Machine* (SVM), parameter yang digunakan adalah kernel. Kernel merupakan metode yang digunakan untuk mengelompokkan data dalam ruang berdimensi tinggi, sehingga ditemukan *hyperlane* untuk klasifikasi data. Dalam kernel ini, digunakan metode linear untuk memisahkan data yang mana pemetaan ruang dimensi tinggi tidak optimal untuk diaplikasikan dalam dataset yang digunakan.

Tabel 5. Parameter SVM

Parameter	Detail
Kernel	Linear

2.4. Training

Data *training* diambil secara acak sebanyak 30% dari keseluruhan gambar yang terdapat dalam masing-masing *dataset*. *Training* algoritma NB, RF, DT, KNN, dan SVM menggunakan kriteria *default* dari *library* sklearn.

2.5. Testing

Testing dilakukan dengan menggunakan algoritma *Stratified K-Fold Cross Validation*. Parameter yang digunakan adalah *K=5*. Hal ini berarti *dataset* yang digunakan akan dibagi menjadi 20% dalam setiap iterasi (total = 5 iterasi). 20% yang diambil dalam stratifikasi ini bersifat acak.

2.6. Perhitungan Performa

Performa dari suatu algoritma *machine learning* dapat diukur menggunakan *accuracy*, *precision*, *recall*, dan *F1* (Naufal, 2021). *Accuracy* menunjukkan tingkat keakuratan dari *classifier*. *Precision* menunjukkan rasio prediksi positif yang benar terhadap hasil prediksi positif. *Recall* menunjukkan rasio prediksi positif terhadap total data yang benar positif. *F1 score* merupakan perbandingan rata-rata *precision* dan *recall* yang dibobotkan.

Rumus dari masing-masing perhitungan tersebut adalah sebagai berikut.

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

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

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

$$F1\ Score = \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

III. HASIL DAN PEMBAHASAN

Dari hasil prediksi yang telah diperoleh, setiap *dataset* memiliki tingkat akurasi tertinggi pada *classifier* yang berbeda – beda. Nilai akurasi klasifikasi pada setiap *dataset* dapat dilihat pada tabel berikut:

Tabel 6. Tingkat Akurasi *Classifier* pada setiap *dataset*

Classifier/Dataset	Rock Paper Scissors	Hand Gestures	American Sign Language
Gaussian Naive Bayes	75.95%	19.51%	65.82%
Random Forest	93.46%	54.99%	96.67%
Decision Tree	93.46%	54.99%	96.69%
K-Nearest Neighbors	94.67%	61.13%	98.14%
Support Vector Machine	95.74%	44.70%	98.01%

Adapun *runtime* yang tercatat pada setiap *classifier* dari *dataset - dataset* yang telah di test. Data dari setiap *runtime* dapat dilihat pada tabel berikut:

Tabel 7. *Runtime Classifier* pada setiap *Dataset*

Classifier/Dataset	Rock Paper Scissors	Hand Gestures	American Sign Language
Gaussian Naive Bayes	3.61 s	5.12 s	7.23 s
Random Forest	1min 15s	1min 47s	1min 21s
Decision Tree	39.1 s	48.7 s	1min 40s
K-Nearest Neighbors	1min 5s	13min 44s	16min 42s
Support Vector Machine	51.6 s	2min 40s	1min 30s

Hasil diperoleh menggunakan *library Python autotime*. Dapat dilihat bahwa setiap algoritma dari setiap dataset memiliki waktu *runtime* yang berbeda-beda.

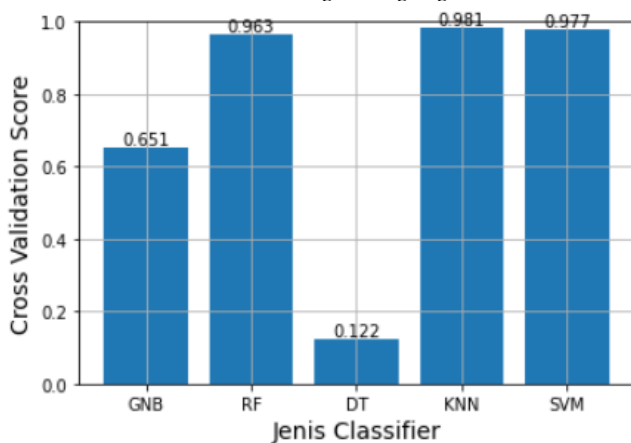
Untuk *dataset Rock-Paper-Scissors*, *runtime* tercepat terjadi pada *Gaussian Naive Bayes classifier* dan terlama pada *Random Forest classifier*. Untuk tingkat akurasi terendah pada dataset ini, terjadi pada *Gaussian Naive Bayes classifier* dengan tingkat akurasinya yaitu 75.95%, namun untuk tingkat akurasi paling tinggi diperoleh pada algoritma *Support Vector Machine classifier* dengan tingkat akurasi 95.74%.

Kemudian pada *dataset Hand Gestures*, *runtime* tercepat terjadi pada *Gaussian Naive Bayes classifier* dan terlama pada *K-Nearest Neighbors classifier*. Akurasi terendah pada *dataset* ini terdapat pada *Gaussian Naive Bayes classifier* dengan tingkat akurasi hanya berkisar 19.51%. sedangkan akurasi tertinggi terdapat pada *K-Nearest Neighbors* dengan akurasi 61.13%.

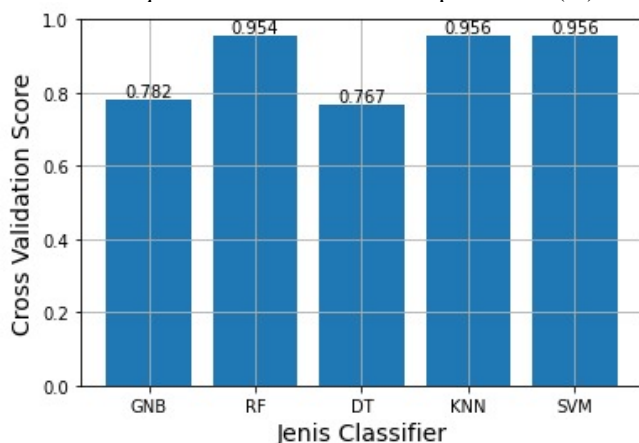
Pada *dataset* terakhir yaitu *American Sign Language*. Sama seperti *dataset* sebelumnya, *Gaussian Naive Bayes classifier* merupakan algoritma dengan *runtime* tercepat, sedangkan *K-Nearest Neighbors classifier* memiliki *runtime* terlama. Akurasi terendah pada *dataset* ini terdapat pada *Gaussian Naive Bayes classifier* dengan tingkat akurasi 65.82%, sedangkan akurasi tertinggi terjadi pada algoritma *K-Nearest Neighbors* dengan tingkat akurasi 98.14%. Akan tetapi, hasil dari algoritma *Support Vector Machine* juga mendekati akurasi dari KNN, yakni 98.01%. Untuk itu, dapat dikatakan bahwa akurasi terbaik terdapat pada algoritma KNN dan SVM.

Adapun nilai *Cross Validation* untuk setiap *classifier* pada *dataset* – *dataset* yang diuji. Nilai perbandingan dapat dilihat pada gambar berikut:

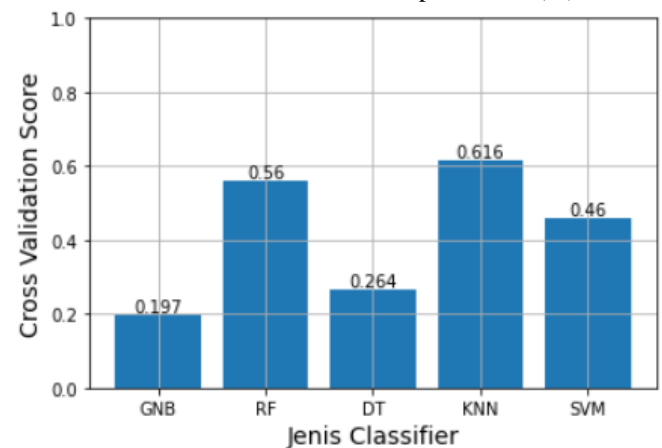
Gambar 2. Perbandingan skor *Cross Validation* pada *dataset American Sign Language*



Gambar 3. Perbandingan skor *Cross Validation* pada *dataset Rock-Paper-Scissors* dalam bentuk persentase (%)



Gambar 4. Perbandingan skor *Cross Validation* pada *dataset Hand Gestures* dalam bentuk persentase (%)



IV. KESIMPULAN

Dari hasil pembahasan dan diskusi yang telah dilakukan, dapat dikatakan bahwa algoritma *classifier* yang digunakan pada setiap *dataset* memiliki tingkat efektivitas yang berbeda – beda tergantung pada *dataset* yang sedang digunakan. Pada *dataset Rock-Paper-Scissors* dan *American Sign Language*, *Support Vector Machine* merupakan *classifier* terbaik dengan tingkat akurasi tertinggi dan *runtime* yang optimal jika dibandingkan dengan *classifier* yang lainnya. Sedangkan pada *dataset Hand Gestures*, *K-Nearest Neighbors* merupakan algoritma terbaik.

Kesimpulannya adalah untuk memperoleh hasil klasifikasi yang paling efektif dan optimal, diperlukan pertimbangan dari faktor – faktor yang mempengaruhi hasil klasifikasi. Mulai dari jumlah gambar yang akan diklasifikasi, jumlah kategori yang terbentuk dari *dataset*, jenis *preprocessing* yang perlu dilakukan untuk memperoleh hasil klasifikasi yang optimal, dan jenis *classifier* yang akan digunakan.

Dari hasil percobaan yang telah kelompok kami lakukan, kami merekomendasikan untuk menggunakan *classifier* SVM dan KNN untuk melakukan uji coba pada *dataset* citra pola tangan. Secara keseluruhan, SVM dan KNN merupakan algoritma paling optimal untuk ketiga *dataset* walaupun pada *dataset Hand Gestures* SVM bukanlah algoritma terbaik.

REFERENSI

- [1] Gunawan, V. A., & Putra, L. S. A. (n.d.). *Perbandingan Identifikasi Penggunaan American Sign Language Menggunakan Klasifikasi Multi-Class SVM, Backpropagation Neural Network, K-Nearest Neighbor dan Naive Bayes*.
- [2] Parlindungan, R., & Rizaldi, M. (2020). Identifikasi dan Klasifikasi Gestur Tangan dengan Sinyal EMG. *Prosiding-Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung*, 242–249.

- [3] Roihan, A., Sunarya, P. A., & Rafika, A. S. (2020).
Pemanfaatan Machine Learning dalam Berbagai Bidang.

IJCIT (Indonesian Journal on Computer and Information Technology), 5(1).
