Studi Komparasi Metode Machine Learning untuk Klasifikasi Citra Huruf Vokal Hiragana

Muhammad Afrizal Amrustian^{1,*}, Vika Febri Muliati², Elsa Elvira Awal³

¹ Fakultas Informatika, Prodi Informatika, Institut Teknologi Telkom Purwokerto, Purwokerto, Indonesia

² Prodi Sistem Informasi, Universitas Siber Asia, Jakarta, Indonesia

³ Fakultas Teknik dan Ilmu Komputer, Prodi Informatika, Universitas Buana Perjuangan Karawang, Karawang, Indonesia

Email: ^{1,*}afrizal.amru@ittelkom-pwt.ac.id, ²vikamuliati@lecturer.unsia.ac.id, ³elsaelvira@ubpkarawang.ac.id

Email Penulis Korespondensi: afrizal.amru@ittelkom-pwt.ac.id

Abstrak—Bahasa Jepang masuk ke dalam salah satu bahasa yang sulit untuk dibaca. Tulisan Jepang yang tidak menggunakan alfabet merupakan alasan dari sulitnya bahasa Jepang untuk dibaca. Bahasa Jepang terdiri dari tiga jenis yakni kanji, katakana, dan hiragana. Huruf hiragana menjadi jenis penulisan yang umum digunakan. Selain itu huruf hiragana memiliki sifat kursif, sehingga masing-masing tulisan dari setiap orang akan berbeda. Sehingga diperlukan sebuah cara untuk mengenali tulisan hiragana. Metode machine learning dapat digunakan untuk membaca huruf hiragana dengan cara mengenali huruf tersebut melalui citra. Namun, belum banyak penelitian terkait studi komparasi terkait klasifikasi citra huruf hiragana. Penelitian ini berfokus untuk melakukan studi komparasi metode machine learning untuk klasifikasi citra huruf Jepang. Penelitian ini dilakukan dengan tujuan mempermudah pembacaan huruf hiragana. Huruf Jepang yang dipelajari pada penelitian ini adalah huruf vokal hiragana. Sedangkan metode *machine learning* yang dikomparasi adalah *Naïve Bayes*, *Support Vector Machine*, *Decision Tree*, *Random Forest*, dan *K-Nearest Neighbor*. Hasil dari studi komparasi menunjukkan bahwa metode *K-Nearest Neighbor* merupakan metode terbaik untuk klasifikasi citra huruf vokal hiragana. *K-Nearest Neighbor* mendapatkan akurasi sebesar 89.4% dengan error rate yang rendah.

Kata Kunci: Huruf hiragana ; Klasifikasi citra; *Naïve Bayes*; SVM; decision tree; random forest; KNN

Abstract—Japanese is one of the most difficult languages to understand and read. Japanese writing that does not use the alphabet is the reason for the difficulty of the Japanese language to read. There are three types of Japanese, namely kanji, katakana, and hiragana. Hiragana letters are the most commonly used type of writing. In addition, hiragana has a cursive nature, so each person's writing will be different. Machine learning methods can be used to read Japanese letters by recognizing the image of the letters. The Japanese letters that are used in this study are hiragana vowels. This study focuses on conducting a comparative study of machine learning methods for the image classification of Japanese letters. The machine learning methods that were successfully compared are *Naïve Bayes*, *Support Vector Machine*, *Decision Tree*, *Random Forest*, and *K-Nearest Neighbor*. The results of the comparative study show that the *K-Nearest Neighbor* method is the best method for image classification of hiragana vowels. *K-Nearest Neighbor* gets an accuracy of 89.4% with a low error rate.

Keywords: Hiragana; Image Classification; *Naïve Bayes*; SVM; Decision Tree; Random Forest; KNN

1. PENDAHULUAN

Bahasa merupakan alat yang digunakan manusia untuk saling berkomunikasi. Ketika manusia lahir maka akan mempelajari bahasa ibu masing-masing. Setiap negara memiliki karakter dan sifat bahasa yang berbeda untuk setiap negaranya. Seperti bahasa Jepang yang memiliki struktur penulisan yang berbeda, yakni tidak menggunakan huruf latin. Ragam bahasa yang berbeda membuat orang akan membutuhkan usaha lebih jika memahami suatu bahasa. Tingkat kesulitan suatu bahasa akan berbeda-beda. Jepang merupakan salah satu masuk dalam bahasa yang memiliki tingkat kesulitan tinggi walaupun telah dipelajari secara khusus [1] [2]. Salah satu kesulitan dalam bahasa Jepang adalah membaca teks bahasa Jepang [3]. Hal ini dikarenakan bahasa Jepang memiliki tiga cara penulisan yakni kanji, katakana, dan hiragana. Dari tiga jenis tersebut, hiragana lebih umum digunakan dan memiliki karakteristik penulisan dan jumlah huruf yang berbeda sehingga ada rintangan tersendiri dalam memahaminya [4]. Huruf hiragana bersifat kursif sehingga tulisan tangan pada masing-masing orang akan berbeda.

Beberapa penelitian dilakukan untuk melakukan klasifikasi tulisan huruf hiragana menggunakan citra yang diambil dari tulisan tangan. Setiawan dan Aulia menggunakan metode *K-Nearest Neighbor* (KNN) untuk mengenali citra tulisan hiragana. Sebelum diklasifikasi menggunakan metode KNN, citra huruf hiragana di proses terlebih dahulu menggunakan algoritma DWT dan DCT. Hasil klasifikasi dari penelitian setiawan dan aulia adalah tingkat akurasi 82.61% dengan algoritma DWT dan 58.70% dengan algoritma DCT [5]. Klasifikasi huruf hiragana juga menjadi objek pada penelitian Umam dan Handoko. Hasil akurasi dari klasifikasi huruf hiragana adalah 82% [6]. Komparasi pengenalan citra antara KNN dan *Support Vector Machine* (SVM) dilakukan untuk memastikan metode mana yang paling unggul dalam klasifikasi citra. Hasilnya KNN memiliki akurasi yang lebih tinggi 10% jika dibandingkan dengan SVM [7]. Metode *Naïve Bayes* dibandingkan dengan metode KNN untuk mengetahui metode terbaik untuk klasifikasi citra. Dilihat dari sisi akurasi dan waktu proses klasifikasi, *Naïve Bayes* memiliki hasil diatas *K-Means* [8]. Komparasi untuk mengklasifikasi citra wajah suku di Indonesia dilakukan oleh Hartono *et al.* Metode yang dikomparasi adalah KNN, SVM, dan *Decision Tree*. Hasil dari komparasi tersebut adalah urutan kinerja metode klasifikasi berdasarkan nilai presisi, recall, akurasi, dan F1. Urutan tersebut adalah sebagai

berikut SVM, KNN, dan *Decision Tree* [9]. Sulaiman *et al* membandingkan algoritma *Decision Tree* dan *Naïve Bayes* untuk menganalisa citra wajah. Hasil akurasi yang didapat dari masing-masing algoritma adalah 58.64% untuk algoritma *Decision Tree* dan 75.45% untuk algoritma *Naïve Bayes* [10].

Berdasarkan dari penelitian sebelumnya bahwa metode *machine learning* dapat digunakan untuk melakukan klasifikasi citra. Namun studi komparasi terhadap metode *machine learning* dalam melakukan klasifikasi citra pada huruf hiragana. Oleh karena itu pada penelitian ini akan dijelaskan hasil dari komparasi berbagai macam metode klasifikasi. Metode yang akan dikomparasi adalah *Naïve Bayes*, J48, SVM, *Random Forest*, *Decision Tree*, dan KNN. Sedangkan objek klasifikasinya adalah citra huruf vokal hiragana. Paper ini terdiri dari pendahuluan, metode penelitian, hasil dan diskusi, dan kesimpulan. Bagian kedua akan menjelaskan metode penelitian yang digunakan pada penelitian ini.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Pada bagian ini akan dijelaskan tahapan penelitian, seperti yang disajikan pada Gambar 1. Metode penelitian yang dilakukan terdiri dari pengumpulan data, pra-pemrosesan data, komparasi kinerja metode klasifikasi, evaluasi dan validasi, hasil dan pembahasan. Gambar 1 menjelaskan alur dari metode penelitian yang dimulai dari pengumpulan data hingga penjelasan hasil dan pembahasan.

Gambar 1. Tahapan Penelitian

2.1.1 Pengumpulan Data

Pada proses pengumpulan data, citra huruf hiragana yang diklasifikasi adalah huruf vokal hiragana yang terdiri dari a, i, u, e, o. Sumber data citra huruf hiragana adalah hasil tulisan tangan dan mengambil citra dari google. Untuk tulisan tangan, masing-masing huruf hiragana akan ditulis menggunakan tangan dan diambil citranya. Gambar 2 merupakan contoh citra yang diambil dari hasil tulisan tangan orang yang berbeda.

Gambar 2. Citra tulisan tangan huruf hiragana

Sumber data yang kedua adalah pengambilan citra huruf hiragana dari google. Selain dari hasil tulisan tangan, data yang digunakan dalam penelitian ini adalah data citra yang diambil dari google. Gambar 3 adalah salah satu contoh data citra yang diambil dari google.

Gambar 3. Citra huruf hiragana dari google

Semua data akan digabung menjadi satu, sehingga masing-masing huruf vokal hiragana terdiri atas 24 citra dan data huruf vokal hiragana secara keseluruhan ada 120 citra.

2.1.2 Pra-pemrosesan Data

Pada sub bagian ini akan dijelaskan terkait pra-pemrosesan data. Tahap pra-pemrosesan data terdiri dari tiga bagian yakni *resize image*, transformasi citra, dan pembagian data. Masing-masing tahap akan dijelaskan lebih lanjut pada bagian berikut.

2.1.2.1 *Resize Image*

Tahap awal pra-pemrosesan data adalah *resize image*. *Resize image* diperlukan untuk menyeragamkan citra yang didapat sehingga mempermudah pengolahan tanpa menghilangkan kualitas citra [11] [12]. *Resize image* menggunakan package EImage yang tersedia pada R package. Ukuran yang diset adalah 15 x 15 pixel. Pada gambar 4 merupakan data citra yang sebelum *resize* dan setelah *resize*.

Gambar 4. Citra sebelum dan sesudah *resize*

2.1.3 Transformasi Citra

Tahap selanjutnya adalah transformasi citra. Citra yang telah diresize akan dirubah warnanya menjadi *grayscale*. Hal ini dilakukan agar nilai piksel yang akan diambil tidak terlalu luas [13]. Sehingga citra yang ditransformasi menjadi warna *greyscale* hanya memiliki rentang piksel dari 0 ke 1, 0 sebagai warna putih, 1 sebagai warna hitam, dan nilai diantaranya adalah warna abu. Setelah menjadi citra *grayscale* akan diambil nilai piksel dari warna tersebut. Gambar 5 adalah gambar yang belum ditransformasi dan gambar 6 adalah gambar yang telah ditransformasi ke *greyscale*.

Gambar 5. Citra sebelum dirubah menjadi *greyscale*

Gambar 6. Citra setelah dirubah menjadi *greyscale*

Setelah citra ditransformasi menjadi *greyscale*, nilai piksel dari citra akan diambil untuk dijadikan fitur pembeda. Sehingga akan ada sebanyak 225 nilai piksel dari ukuran gambar 15x15 piksel yang dapat dijadikan fitur dari citra tersebut. Tabel 1 merupakan sampel dari nilai piksel yang telah diambil dari citra *greyscale*.

Tabel 1. Nilai piksel dari citra

Piksel 1	Piksel 2	Piksel 3	...	Piksel 225	Label
0.991285	0.991285	0.991285	...	0.984314	A
0.984314	0.984314	0.984314	...	1	A
1	1	1	...	1	A
1	1	1	...	0.752614	A
0.762092	0.759477	0.75817	...	0.790849673	A
0.999891068	0.993137255	0.998366013	...	1	I
0.984313725	0.983986928	0.984422658	...	0.984313725	I
1	1	1	...	1	I
1	1	1	...	1	I
0.776470588	0.778588235	0.777189542	...	0.764	I
1	1	1	...	1	U
0.984313725	0.984313725	0.984313725	...	0.984313725	U
1	1	1	...	1	U
1	1	1	...	1	U
0.743572985	0.739869281	0.734771242	...	0.745664488	U
0.984313725	0.984313725	0.984313725	...	0.984313725	E
1	1	1	...	1	E
1	0.996187364	0.532026144	...	0.996732026	E
0.654357298	0.658169935	0.646405229	...	0.687581699	E
0.786928105	0.806535948	0.797930283	...	0.773202614	E
1	1	1	...	0.999455338	O
0.984313725	0.984313725	0.984313725	...	0.984313725	O
1	1	1	...	1	O
1	1	1	...	1	O
0.814052288	0.802941176	0.808169935	...	0.741503268	O

2.1.4 Pembagian Data

Pada tahap pembagian data, metode *k-fold cross validation* digunakan untuk membagi dataset yang ada. Metode ini diambil agar menghindari terjadinya *overfitting* pada model klasifikasi dan mendapatkan performa metode klasifikasi secara umum [14]. Dataset akan dibagi menjadi 10 *fold*, dari masing-masing *fold* akan dihitung performa metode klasifikasi dan diambil rata-rata dari performa tersebut.

2.1.5 Komparasi Kinerja Klasifikasi Data

2.1.5.1 Naïve Bayes

Naïve bayes merupakan sebuah metode untuk memprediksi sebuah kelas berdasarkan perhitungan probabilitas. Metode ini umum digunakan karena kemudahan dalam membangun model klasifikasinya dan menguji modelnya. Perhitungan teori dari *Naïve Bayes* dapat dilihat dari persamaan 1 berikut [15].

$$P(H/X) = \frac{P\left(\frac{H}{X}\right)P(H)}{P(X)} \quad (1)$$

Dimana $P(H/X)$ merupakan *posterior* probabilitas dari H berdasarkan kondisi X . $P(H)$ adalah probabilitas *prior* dari H . $P(X)$ didapatkan dari probabilitas *prior* X .

2.1.5.2 SVM

SVM merupakan sebuah metode yang bersifat supervised, SVM dapat digunakan untuk membuat sebuah model dari hasil analisa data baik untuk kasus klasifikasi maupun regresi [16]. SVM bekerja dengan mencari titik terluar dari masing-masing data untuk kemudian ditarik sebuah garis pemisah yang paling optimal [17]. Dari garis pemisah yang telah didapat maka akan terdapat dua atau lebih kelompok data. SVM dapat melakukan baik itu linear dan non-linear klasifikasi. Gambar 7 merupakan ilustrasi dari metode SVM.

Gambar 7. Ilustrasi Metode SVM [18]

Pada metode SVM, titik terluar dari masing-masing kelompok data akan dicari. Setelah titik terluar data ditemukan maka akan dihitung margin antar titik untuk kemudian ditarik garis pemisah antar kelompok data. Hal tersebut akan terus diulang hingga ditemukan garis pemisah terbaik untuk memisahkan masing-masing kelompok data.

2.1.5.3 Decision Tree (J48)

Decision Tree merupakan metode yang mengobservasi sebuah data berdasarkan model yang berbentuk pohon. Sebuah pohon pada *Decision Tree* akan belajar dengan membagi hasil prediksi berdasarkan nilai yang dihitung dari atribut atau fitur [16]. Pembagian tersebut akan dilakukan hingga didapatkan kelas dari data tersebut. Gambar 8 merupakan ilustrasi dari metode *Decision Tree*

Gambar 8. Ilustrasi *Decision Tree* [16].

2.1.5.4 Random Forest

Random forest merupakan sebuah integrasi dari *bagging* dan beberapa *Decision Tree* [19]. Kombinasi beberapa tree akan menghasilkan beberapa suara yang kemudian diambil suara terbanyak atau voting. Hasil voting ini lah yang menjadi hasil akhir dari klasifikasi. Ilustrasi dari metode *Random Forest* dapat dilihat pada gambar 9. Pada gambar 9 *Random Forest* merupakan kumpulan dari *Tree 1*, *Tree 2*, ..., *Tree b* setiap *tree* akan menghasilkan sebuah keputusan. Dari setiap keputusan yang dihasilkan akan dilakukan voting, k merupakan hasil voting dari seluruh *tree* yang ada di dalam *Random Forest*.

Gambar 9 Ilustrasi *random forest*

2.1.5.5 KNN

KNN merupakan algoritma yang bekerja dengan mengelompokkan data berdasarkan nilai yang berdekatan dengan data tersebut [20]. Perhitungan jarak antara data dengan kelompok menggunakan perhitungan jarak *euclidian*. Jarak *euclidian* didapat berdasarkan perhitungan pada persamaan 2 berikut ini [21].

$$d(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

2.1.6 Evaluasi dan Validasi

Pada tahap ini akan dilakukan evaluasi dari masing-masing metode. Evaluasi dilakukan berdasarkan *confusion matrix*. Berdasarkan *confusion matrix* maka dapat dilakukan evaluasi kinerja metode. Evaluasi dilakukan berdasarkan tingkat akurasi, nilai presisi dan *recall*, dan perhitungan *error rate*. Detail dari evaluasi akan dijelaskan pada bagian hasil dan pembahasan.

3. HASIL DAN PEMBAHASAN

Tahap ini akan menjelaskan hasil evaluasi dari masing-masing metode yang digunakan pada penelitian ini. Evaluasi menjadi hal yang penting agar kita mendapatkan sebuah metode yang cocok untuk mengklasifikasi citra huruf hiragana. Hasil yang akan dibahas adalah tingkat akurasi masing-masing metode, nilai presisi dan *recall*, dan *error rate*. Pertama adalah hasil dari penelitian terkait hasil akurasi penelitian. Hasil tersebut dapat dilihat pada tabel 2. Tabel 2 menunjukkan persentase *Correctly Classified Instance*, persentase *Incorrectly Classified Instance*, *Running Time*, dan nilai *Kappa Statistic*. *Correctly Classified Instance* adalah jumlah data yang terklasifikasi dengan benar sesuai dengan kelasnya. *Incorrectly Classified Instance* merupakan banyaknya data yang terklasifikasi tidak sesuai dengan kelasnya. *Running time* adalah waktu yang dibutuhkan untuk membangun model klasifikasi. *Kappa Statistic* adalah cara untuk mengetahui tingkat konsistensi suatu metode dalam mengklasifikasi data. Gambar 10 merupakan hasil visualisasi dari akurasi masing-masing metode. Visualisasi ini akan mempermudah untuk melihat metode yang memiliki akurasi paling tinggi atau paling rendah.

Tabel 2. Hasil akurasi

Metode	<i>Correctly Classified Instance (%)</i>	<i>Incorrectly Classified Instance (%)</i>	<i>Running Time</i>	<i>Kappa Statistic</i>
Naïve Bayes	74.2424	25.7576	0.02	0.6762
SVM	96.9697	3.0303	0.25	0.9620
Decision Tree	78.7879	21.2121	0.03	0.7344
Random Forest	88.6364	11.3636	0.16	0.8573
KNN	89.3939	10.6061	0.01	0.8667

Gambar 10. Visualisasi hasil akurasi

Dari tabel 2 dapat dilihat bahwa SVM memiliki akurasi yang paling tinggi jika dibandingkan dengan metode yang lain. Namun dari sisi *running time* SVM adalah yang paling lambat. Metode yang memiliki *running time* tercepat adalah KNN, walaupun akurasinya lebih kecil 10% jika dibandingkan dengan SVM. Selain dari hasil akurasi dan *running time*, hasil *kappa statistic* juga penting dalam melihat hasil akurasi dari sebuah metode. *Kappa statistic* digunakan untuk mengukur kesesuaian antara hasil dan alat ukur yang digunakan. Nilai rata-rata *kappa statistic* yang dianggap tinggi adalah sekitar 0.8 – 0.9.

Tabel 3 merupakan hasil dari perhitungan presisi dan recall pada masing-masing metode. Perhitungan presisi berfungsi untuk mengetahui jumlah persentase kebenaran dari hasil klasifikasi yang benar. Sedangkan *recall* adalah perhitungan untuk mengetahui persentase hasil klasifikasi yang benar berdasarkan data yang asli.

Tabel 3. Perhitungan presisi dan *recall*

Metode	Nilai Presisi (%)	Nilai <i>Recall</i> (%)
Naïve Bayes	77.5	50.0
SVM	97.0	97.0
Decision Tree	80.0	78.8
Random Forest	71.8	71.2
KNN	89.6	89.4

Metode SVM memiliki nilai presisi dan *recall* yang paling tinggi dibandingkan dengan metode lain. Nilai presisi metode SVM sebesar 97.0% yang berarti persentase kebenaran dari hasil klasifikasi yang benar sangat tinggi. Nilai *recall* metode SVM sebesar 97.0% yang mengartikan bahwa persentase perbandingan antara hasil klasifikasi yang benar dan data yang sebenarnya sangat tinggi. Metode Naïve Bayes memiliki nilai presisi dan *recall* yang paling rendah diantara metode lainnya.

Pada tabel 4 disajikan hasil klasifikasi berdasarkan error selama proses klasifikasi berlangsung. Tabel 4 memberikan kita informasi berupa *mean absolute error*, *root mean squared error*, *relative absolute error*, dan *root relative squared error*. Semakin kecil nilai *error*, maka akan semakin bagus metode tersebut dalam memecahkan masalah. Metode yang memiliki nilai *error* yang paling kecil akan memiliki kapasitas dan kemampuan klasifikasi.

Tabel 4. Nilai error

Metode	<i>Mean Absolute Error</i>	<i>Root Mean Squared Error</i>	<i>Relative Absolute Error (%)</i>	<i>Root Relative Squared Error (%)</i>
Naïve Bayes	0.1016	0.3156	31.8325	78.9747
SVM	0.2427	0.3200	76.0621	80.0804
Decision Tree	0.0990	0.2833	31.0155	70.8923
Random Forest	0.1482	0.2292	46.4440	57.3628
KNN	0.0557	0.2061	17.4662	51.5669

Berdasarkan hasil dari tabel 3 maka KNN memiliki nilai error yang paling rendah diantara metode yang lain. Walaupun SVM memiliki tingkat akurasi, running time, *kappa statistic*, presisi, dan *recall* yang tinggi, namun SVM memiliki nilai error yang tinggi jika dibandingkan dengan metode yang lain. Sedangkan KNN merupakan nomor 2 tertinggi untuk tingkat akurasi, running time, *kappa statistic*, presisi, dan *recall*. Namun nilai error metode KNN adalah yang terendah, sehingga KNN dapat diandalkan dalam mengklasifikasi citra huruf vokal hiragana.

4. KESIMPULAN

Penelitian ini telah berhasil melakukan komparasi performa dari beberapa teknik klasifikasi. Beberapa metode klasifikasi yang dikomparasi adalah naïve bayes, SVM, decision tree, random forest, dan KNN. Metode yang telah disebutkan berhasil melakukan klasifikasi data citra huruf vokal hiragana. Berdasarkan sisi akurasi urutan metode yang memiliki hasil akurasi dari yang paling tinggi hingga paling rendah adalah sebagai berikut, SVM, KNN, Random Forest, Decision Tree, dan Naive Bayes. Sedangkan berdasarkan *error rate* urutan metode yang mendapatkan nilai *error rate* terendah hingga tertinggi adalah sebagai berikut, KNN, Decision Tree, Naive Bayes, Random Forest, dan SVM. Berdasarkan hasil tersebut maka dapat diambil kesimpulan bahwa KNN adalah metode yang paling tepat untuk mengklasifikasi citra huruf hiragana. Hal itu terlihat dari hasil akurasi dan nilai error yang didapat. Pada penelitian selanjutnya adalah memperluas data citra yang diklasifikasi. Data citra yang diperluas didapatkan dari jenis huruf hiragana maupun jenis huruf jepang yang lainnya yakni kanji dan katakana. Menggunakan metode KNN yang merupakan hasil terbaik diantara metode lain, diharapkan mampu mengklasifikasi jenis huruf jepang yang lainnya.

REFERENCES

- [1] P. UNIDA, "Bahasa Tersulit di Dunia, Memang ada Bahasa yang Sulit?," 2020. [Online]. Available: <http://pps.unida.gontor.ac.id/bahasa-tersulit-di-dunia-memang-ada-bahasa-yang-sulit/>. [Accessed: 11-Jun-2021].
- [2] R. Eriawan and M. A. Putri, "Faktor Penyebab Kesulitan Belajar Bahasa Jepang Mahasiswa Tahun Masuk 2018 Kelas International Program Studi Pendidikan Bahasa Inggris UNP," *OMIYAGE*, vol. 3, no. 2, pp. 20–32, 2020.
- [3] S. Monoarfa, "Tingkat Kemampuan Membaca Teks Berbahasa Jepang Dalam Penguasaan Huruf Kanji Sebagai Dasar Terjemahan Yang Tepat," *Stud. Soc. Sci.*, vol. 2, no. 1, p. 22, 2019.
- [4] R. I. Arief, L. Wahyuni, and K. Puspita, "Penerapan Metode Linear Congruent Method (Lcm) Pada Perangkat Lunak Tebak Huruf Hiragana Berbasis Android," *IT J.*, vol. 7, no. 1, pp. 63–72, 2019.

- [5] S. Aulia and A. Setiawan, "Pengenalan Tulisan Tangan Karakter Hiragana Menggunakan Dct, Dwt, Dan K-Nearest Neighbor," *J. Elektro dan Telekomun. Terap.*, vol. 4, no. 1, p. 467, 2017.
- [6] C. Umam and L. Budi Handoko, "Convolutional Neural Network (CNN) Untuk Identifikasi Karakter Hiragana," in *Prosiding Seminar Nasional Lppm Ump*, 2020, vol. 0, no. 0, pp. 527–533.
- [7] R. Umar, I. Riadi, and D. A. Farock, "Komparasi Image Matching Menggunakan Metode K-Nearest Neighbor (KNN) dan Metode Support Vector Machine (SVM)," *J. Appl. Informatics Comput.*, vol. 4, no. 2, pp. 124–131, 2020.
- [8] K. Auliasari and M. Kertaningtyas, "Studi Komparasi Klasifikasi Pola Tekstur Citra Digital Menggunakan Metode K-Means Dan Naïve Bayes," *J. Inform.*, vol. 18, no. 2, pp. 175–185, 2018.
- [9] S. Hartono, H. Sujaini, and A. Perwitasari, "Komparasi Algoritma Nonparametrik untuk Klasifikasi Citra Wajah Berdasarkan Suku di Indonesia," *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 3, pp. 338–343, 2020.
- [10] H. Sulaiman, D. Riana, and A. Rifai, "Perbandingan Algoritma Decision Tree C4 . 5 dan Naive Bayes pada Analisis Tekstur Gray Level Co - occurrence Matrix Menggunakan Citra Wajah," vol. 10, pp. 470–479, 2021.
- [11] D. Abdullah and E. D. Putra, "Komparasi Perbaikan Kualitas Segmentasi Pada Citra Digital Metode Fuzzy C-Means Dan Otsu," *Pseudocode*, vol. 4, no. 1, pp. 71–80, 2017.
- [12] D. S. Tobias and A. R. Widiarti, "Deteksi Glaukoma pada Citra Fundus Retina dengan Metode K-Nearest Neighbor," in *Seminar Nasional Ilmu Komputer(SNIK 2016)*, 2016, pp. 92–99.
- [13] F. Muwardi and A. Fadlil, "Sistem Pengenalan Bunga Berbasis Pengolahan Citra dan Pengklasifikasi Jarak," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 3, no. 2, pp. 124–131, 2017.
- [14] D. Berrar, "Cross-validation," *Encycl. Bioinforma. Comput. Biol. ABC Bioinforma.*, vol. 1, pp. 542–545, 2018.
- [15] M. Vadivukarassi, N. Puviarasan, and P. Aruna, "Sentimental Analysis of Tweets Using Naive Bayes Algorithm," *World Appl. Sci. J.*, vol. 35, no. 1, pp. 54–59, 2017.
- [16] E. Ahishakiye, E. O. Omulo, D. Taremwa, and I. Niyonzima, "Crime prediction using Decision Tree (J48) classification algorithm," *Int. J. Comput. Inf. Technol.*, vol. 06, no. 03, pp. 188–195, 2017.
- [17] T. B. Sasongko, "Komparasi dan Analisis Kinerja Model Algoritma SVM dan PSO-SVM (Studi Kasus Klasifikasi Jalur Minat SMA)," *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 2, pp. 244–253, 2016.
- [18] F. Rahutomo, P. Y. Saputra, and M. A. Fidyawan, "IMPLEMENTASI TWITTER SENTIMENT ANALYSIS UNTUK REVIEW FILM MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE," *J. Inform. Polinema*, vol. 4, no. 2, pp. 93–100, 2016.
- [19] Y. Xu, X. Zhao, Y. Chen, and Z. Yang, "Research on a mixed gas classification algorithm based on extreme random tree," *Appl. Sci.*, vol. 9, no. 9, 2019.
- [20] Mustakim and G. Oktaviani F, "Algoritma K-Nearest Neighbor Classification Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa," vol. 13, no. 2, pp. 195–202, 2016.
- [21] A. Pandey and A. Jain, "Comparative Analysis of KNN Algorithm using Various Normalization Techniques," *Int. J. Comput. Netw. Inf. Secur.*, vol. 9, no. 11, pp. 36–42, 2017.