

PENGENALAN CITRA TULISAN TANGAN HURUF HIRAGANA MENGGUNAKAN SUPPORT VECTOR MACHINE

Tsabita Safana Mustofa¹⁾, Anggraini Puspita Sari^{*2)}, Hendra Maulana³⁾

1. Informatika, Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia
2. Informatika, Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia
3. Bisnis Digital, Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia

Article Info

Kata Kunci: Hiragana; Pengolahan citra; Pengenalan karakter; Pelurusan; SVM

Keywords: Character recognition; Hiragana; Flattening; Image processing; SVM

Article history:

Received 12 May 2025

Revised 18 June 2025

Accepted 21 July 2025

Available online 1 March 2026

DOI :

<https://doi.org/10.29100/jipi.v11i1.7841>

* Corresponding author.

Corresponding Author

E-mail address:

anggraini.puspita.if@upnjatim.ac.id

ABSTRAK

Pengenalan huruf Hiragana merupakan tantangan tersendiri bagi pembelajar bahasa Jepang, terutama karena bentuk karakter yang kompleks dan mirip satu sama lain. Penelitian ini bertujuan untuk mengembangkan sistem pengenalan citra tulisan tangan huruf Hiragana menggunakan metode *Support Vector Machine* (SVM). Dataset yang digunakan terdiri dari 2300 citra tulisan tangan huruf Hiragana dasar sebanyak 46 karakter, yang diperoleh dari hasil pemindaian tulisan tangan di atas kertas. Tahapan penelitian meliputi *preprocessing* (konversi ke *grayscale* dan *resize* citra) lalu diklasifikasi menggunakan SVM dengan berbagai kernel dan nilai parameter yang berbeda. Evaluasi model dilakukan dengan menggunakan *confusion matrix* yang menghasilkan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Hasil penelitian menunjukkan bahwa kernel RBF dengan parameter $C = 1000$ dan γ yang diatur secara default memberikan hasil terbaik dengan akurasi sebesar 93,04%, *precision* sebesar 93,84%, *recall* sebesar 93,04%, dan *F1-score* sebesar 92,92% pada skenario proporsi data 90:10. Temuan ini menunjukkan bahwa metode SVM mampu mengenali huruf Hiragana secara efektif. Sistem ini dapat menjadi alat bantu edukatif yang bermanfaat dalam proses pembelajaran bahasa Jepang serta memiliki potensi untuk dikembangkan lebih lanjut dalam aplikasi pembelajaran interaktif.

ABSTRACT

Hiragana characters recognition is a challenge for Japanese language learners, especially because the character shapes are complex and similar to each other. This research aims to develop a Hiragana handwritten image recognition system using the Support Vector Machine (SVM) method. The dataset used consists of 2300 handwritten images of basic Hiragana letters totaling 46 characters, obtained from scanning handwriting on paper. The research stages include preprocessing (conversion to grayscale and image resize) and then classified using SVM with various kernels and different parameter values. Model evaluation is done using confusion matrix which produces evaluation metrics such as accuracy, precision, recall, and F1-score. The results showed that the RBF kernel with parameters $C = 1000$ and γ set by default gave the best results with accuracy of 93.04%, precision of 93.84%, recall of 93.04%, and F1-score of 92.92% in the 90:10 data proportion scenario. This finding shows that the SVM method is able to recognize Hiragana letters effectively. This system can be a useful educational tool in the Japanese language learning process and has the potential to be further developed in interactive learning applications.

I. PENDAHULUAN

JEPANG merupakan salah satu negara yang mempunyai kondisi ekonomi paling stabil dan kuat di dunia yang dibuktikan dengan nilai tukar mata uang yen terhadap mata uang asing yang tergolong tinggi. Dalam beberapa tahun terakhir, Pemerintah Jepang secara terbuka menyampaikan bahwa negara mereka sedang mengalami masalah serius dalam hal kependudukan, khususnya akibat angka kelahiran yang terus menurun [1]. Kondisi ini berdampak pada kekurangan tenaga kerja di berbagai sektor industri. Sebagai solusi, Jepang mulai membuka lebar peluang kerja bagi para tenaga kerja asing untuk bekerja di negaranya. Hal ini menjadi peluang besar bagi para

pekerja dari berbagai negara, termasuk Tenaga Kerja Indonesia (TKI), untuk mencari penghidupan yang lebih baik di negeri sakura tersebut. Pada tahun 2019, terdapat lebih dari 40.000 pekerja asal Indonesia yang bekerja di Jepang dan angka tersebut terus mengalami peningkatan setiap tahunnya [2]. Selain gaji yang kompetitif dan prospek kerja yang menjanjikan, banyak masyarakat Indonesia juga tertarik dengan kebudayaan Jepang, seperti anime, manga, musik, dan masih banyak lagi. Salah satu aspek penting dari budaya Jepang yang banyak menarik perhatian adalah bahasa Jepang itu sendiri, yang memiliki sistem penulisan yang sangat berbeda dibandingkan dengan alfabet Latin yang umum digunakan di berbagai negara.

Sistem penulisan bahasa Jepang terbagi menjadi tiga komponen utama, yaitu Hiragana, Katakana, dan Kanji. Hiragana merupakan sistem penulisan dasar bahasa Jepang yang paling umum digunakan, khususnya untuk menuliskan kata-kata asli dalam bahasa Jepang yang tidak memiliki bentuk Kanji [3]. Huruf-huruf Hiragana memiliki ciri khas yang unik karena terdiri dari berbagai lengkungan, garis-garis halus, serta bentuk yang variatif, sehingga seringkali sulit untuk dibedakan satu sama lain oleh para pemula [4]. Kompleksitas dari huruf Hiragana inilah yang menjadi salah satu tantangan utama dalam proses pembelajaran huruf Hiragana. Akibatnya, banyak pembelajar pemula yang mengalami kesulitan dalam menghafal dan membedakan setiap karakter secara akurat, terutama tanpa bantuan metode pengenalan visual yang tepat dan efektif.

Perkembangan teknologi pengolahan citra telah memberikan kontribusi yang signifikan dalam bidang pengenalan karakter atau huruf, terutama dalam menangani bentuk-bentuk huruf yang mirip dan kompleks seperti huruf Hiragana. Untuk mempermudah proses dalam mengenali huruf Hiragana, dibutuhkan dukungan teknologi digital, khususnya yang berbasis *machine learning*, guna meningkatkan efisiensi dalam proses pembelajaran. *Machine learning* merupakan cabang dari kecerdasan buatan yang memungkinkan sistem untuk belajar secara mandiri tanpa harus diprogram berulang berdasarkan data yang telah diberikan [5]. Terdapat 3 macam *machine learning*, yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*. *Machine learning* memiliki beberapa jenis pendekatan dan algoritma yang dirancang untuk mengenali data dan membuat prediksi atau keputusan berdasarkan pola tersebut [6]. Salah satu metode *machine learning* yang sering digunakan adalah *Support Vector Machine* (SVM). SVM adalah salah satu algoritma *machine learning* yang termasuk dalam *Supervised Learning*. Pada perkembangannya, SVM terus mengalami penyempurnaan dan menjadi algoritma yang andal dalam berbagai studi dan aplikasi *machine learning*. Tujuan utama dari metode SVM adalah untuk memastikan margin maksimum antara *hyperplane* dengan data terdekat dari masing-masing kelas yang dikenal sebagai *support vectors* [7].

Berbagai penelitian sebelumnya telah dilakukan untuk mengembangkan sistem yang mampu mengenali karakter huruf yang menggunakan berbagai metode klasifikasi huruf. Salah satu penelitian yang relevan adalah penelitian yang dilakukan oleh Amrustian et al. pada tahun 2021 yang berjudul "Studi Komparasi Metode *Machine Learning* untuk Klasifikasi Citra Huruf Vokal Hiragana". Dalam penelitian tersebut, dilakukan perbandingan terhadap lima algoritma *machine learning* dalam mengklasifikasi huruf vokal Hiragana, yaitu *Naive Bayes*, SVM, *Decision Tree*, *Random Forest*, dan KNN. Dari hasil pengujian, metode SVM menunjukkan performa tertinggi dengan akurasi mencapai 97%, mengungguli keempat metode lainnya [8]. Kemudian, penelitian oleh Putri et al. pada tahun 2024 dengan judul "Pengolahan Citra Huruf Hijaiyah Menggunakan Algoritma Support Vector Machine" menunjukkan potensi besar metode SVM dalam pengenalan karakter. Dengan menggunakan 400 citra huruf Hijaiyah sebagai dataset, sistem berhasil mencapai akurasi hingga 99%. Capaian ini mengindikasikan bahwa SVM memiliki performa yang sangat baik, terutama jika diterapkan pada dataset yang mencukupi secara jumlah maupun keragaman data [9].

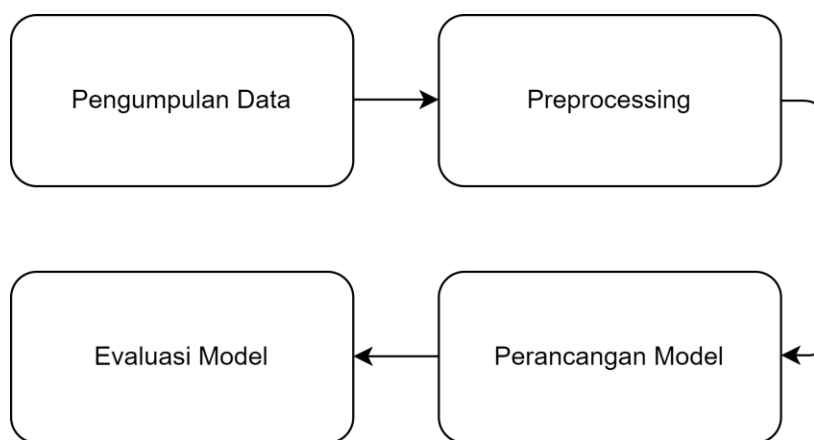
Pada penelitian ini, sistem diawali dengan *preprocessing* dengan mengubah citra tulisan tangan huruf Hiragana berwarna diubah ke *grayscale*, lalu *resize*, kemudian dilakukan *flatten* ke dalam vektor elemen intensitas yang siap digunakan model klasifikasi. SVM kemudian memanfaatkan vektor tersebut untuk membangun *hyperplane* dengan margin maksimum, sehingga huruf-huruf yang bentuknya mirip dapat dipisahkan lebih jelas dibanding algoritma sederhana seperti KNN atau *Naive Bayes*. Bagi pemula yang sedang mempelajari huruf Hiragana, sistem ini dapat menjadi alat bantu yang andal untuk mengenali karakter secara otomatis dan konsisten, sehingga mendukung proses pembelajaran yang interaktif dan efektif. Keunggulan tersebut muncul karena SVM andal mencari batas kelas yang optimal sekalipun fitur yang dipakai sederhana, sehingga akurasi tetap terjaga [10]. Pendekatan ini terbukti mampu meningkatkan akurasi karena SVM efektif dalam menangani data berdimensi tinggi dan dapat meminimalkan kesalahan klasifikasi. Dibanding dengan CNN, alur *preprocessing* yang dilanjutkan *flatten* kemudian diproses oleh model SVM jauh lebih ringan secara komputasi dan tidak menuntut dataset besar ataupun pelatihan yang kompleks.

Berdasarkan penelitian-penelitian sebelumnya, dapat diketahui bahwa metode SVM secara konsisten juga menunjukkan akurasinya yang tinggi, yang berarti terbukti efektif dalam mengklasifikasi sebuah objek [11]. Penelitian ini berinovasi dengan menata ulang *preprocessing*, normalisasi intensitas, perubahan ukuran, dan *flatten*

citra, lalu menyesuaikan parameter SVM secara sistematis agar model semakin peka terhadap variasi goresan tulisan tangan pada huruf Hiragana. Pendekatan terintegrasi ini memberi keuntungan utama dibandingkan algoritma sejenis pada penelitian terdahulu, seperti waktu inferensi lebih singkat karena vektor piksel langsung diproses tanpa ekstraksi fitur terpisah dan mereduksi kesalahan klasifikasi pada huruf-huruf yang bentuknya nyaris sama berkat margin maksimum SVM yang memisahkan kelas secara lebih optimal. Dengan demikian, sistem yang dikembangkan dapat menjadi alat bantu yang efektif untuk mempermudah pengenalan dan pembelajaran huruf Hiragana secara visual, sehingga mengatasi tantangan kesulitan dalam membedakan bentuk huruf yang serupa.

II. METODE PENELITIAN

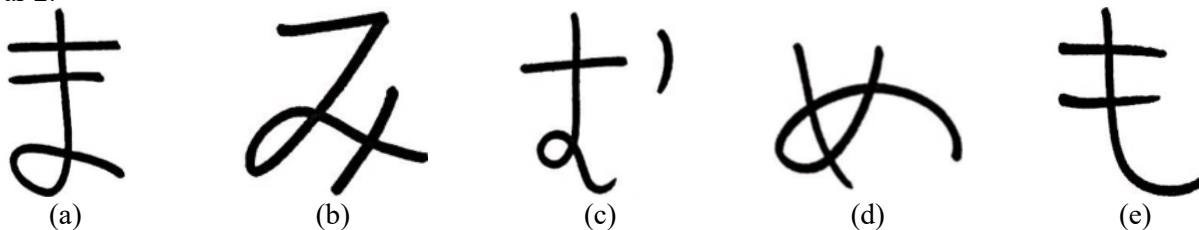
Tahapan-tahapan penelitian yang ditunjukkan pada Gambar 1 dilakukan secara sistematis untuk memastikan penelitian berjalan dengan terstruktur dan menghasilkan hasil yang sesuai dengan tujuan. Pengumpulan data dilakukan dengan mengumpulkan dataset yang sesuai sebagai bahan analisis. *Preprocessing* diterapkan untuk mempersiapkan data agar lebih siap digunakan pada proses berikutnya. Tahap berikutnya adalah perancangan model yang mencakup proses pemilihan algoritma serta parameter yang optimal. Setelah model dirancang dan dilatih, evaluasi dilakukan untuk mengukur performa model menggunakan metrik tertentu.



Gambar 1. Tahapan Penelitian

A. Pengumpulan Data

Pada tahap pengumpulan data dalam penelitian ini, data yang digunakan adalah huruf Hiragana dasar yang terdiri dari 46 huruf, yaitu A, I, U, E, O, KA, KI, KU, KE, KO, SA, SHI, SU, SE, SO, TA, CHI, TSU, TE, TO, NA, NI, NU, NE, NO, HA, HI, HU, HE, HO, MA, MI, MU, ME, MO, YA, YU, YO, RA, RI, RU, RE, RO, WA, WO, N. Dari 46 huruf tersebut, terdapat 50 gambar tulisan tangan huruf Hiragana pada setiap karakter sehingga total gambar yang didapat adalah 2300 citra. Citra tulisan tangan huruf Hiragana ditulis di kertas HVS lalu discan dengan menggunakan kamera *handphone*. Bentuk hasil citra huruf Hiragana dari pengumpulan data dapat dilihat pada Gambar 2.



Gambar 2. (a) Citra Huruf MA, (b) Citra Huruf MI, (c) Citra Huruf MU, (d) Citra Huruf ME, (e) Citra Huruf MO

B. Preprocessing

Tahap *preprocessing* adalah tahapan yang bertujuan membantu meningkatkan kompleksitas dalam proses kalsifikasi citra. Pada tahap *preprocessing*, serangkaian proses dilakukan untuk mempersiapkan citra agar sesuai dengan kebutuhan ekstraksi fitur dan pelatihan model. Berikut adalah tahapan *preprocessing* yang dilakukan pada penelitian ini.

1. Konversi Citra ke *Grayscale*

Pada tahap pertama *preprocessing* ini, citra yang awalnya berwarna akan diubah menjadi citra *grayscale*. Proses ini dilakukan karena pengolahan citra sering memerlukan waktu dan sumber daya komputasi yang

lebih besar [12]. Dalam citra *grayscale*, hanya ada satu *channel* yang menggambarkan intensitas atau kecerahan piksel yang lebih sederhana untuk dianalisis. Dengan mengubah citra menjadi *grayscale*, proses ekstraksi fitur dapat dilakukan dengan lebih efisien [13].

2. *Resize* Citra

Pada tahap ini, citra huruf Hiragana akan diubah ukurannya untuk memastikan bahwa semua citra yang akan diproses memiliki ukuran yang sama. Dengan mengubah ukuran citra, dimensi yang dimiliki menjadi cukup kecil untuk mengurangi beban komputasi [14]. Pada penelitian ini, citra huruf Hiragana dilakukan *resize* menjadi 64 x 64 piksel. Ukuran ini dipilih karena dapat mempertahankan fitur-fitur penting dalam citra tanpa kehilangan terlalu banyak detail yang berpotensi mengurangi akurasi model.

3. Pembagian Data

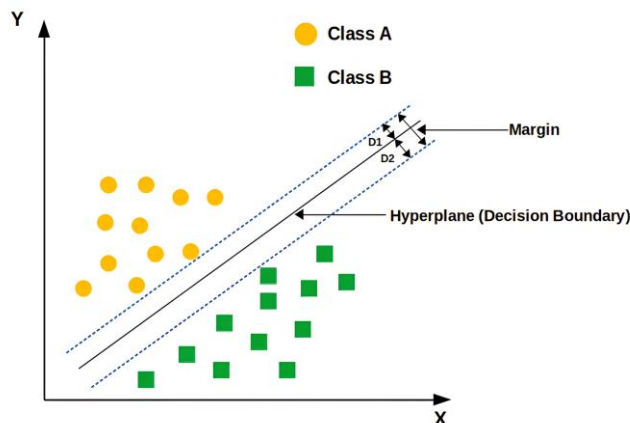
Setelah data mempunyai ukuran yang seragam, langkah selanjutnya adalah membagi data menjadi data latih dan data uji. Tahapan pembagian data ini dilakukan untuk memastikan bahwa model dapat dilatih dan diuji dengan data yang berbeda, sehingga performa model dapat dievaluasi dengan akurat dan menghindari *overfitting* [15]. Pembagian data dilakukan dengan proporsi 90% data latih dan 10% data uji. Proporsi tersebut digunakan untuk mengevaluasi kestabilan model pada data latih yang besar dan sedang. Sebelum dibagi, dataset diseimbangkan per karakter, sehingga setiap huruf Hiragana muncul dalam jumlah yang sama pada data latih dan data uji. Langkah ini mencegah bias terhadap karakter tertentu. Rasio ketidakseimbangan tetap 1:1 setelah pembagian, sehingga kinerja yang tercatat menggambarkan kemampuan model mengenali semua karakter secara adil. Jumlah pembagian data latih dan data uji terdapat pada Tabel 1.

TABEL I
 PEMBAGIAN DATA

Nama Data	%	Jumlah Data
<i>Training</i>	90	45
<i>Testing</i>	10	5

C. Model SVM

SVM merupakan salah satu metode yang banyak digunakan dalam berbagai bidang, termasuk pengenalan pola, klasifikasi gambar, dan analisis teks [16]. SVM merupakan salah satu algoritma klasifikasi yang populer karena kemampuannya menghasilkan model yang kuat meskipun data relatif sedikit dan berdimensi tinggi. SVM bekerja dengan cara mencari *hyperplane* atau batas keputusan yang memisahkan data dari kelas yang berbeda secara optimal dengan margin terlebar [17]. Semakin lebar margin, semakin kecil kemungkinan model salah saat dihadapkan pada data baru. Untuk mencapai hal ini, SVM memaksimalkan jarak antara pemilihan fungsi kernel yang memetakan data ke ruang berdimensi lebih tinggi ketika pemisahan linear tidak memadai. Dalam kasus data dua dimensi, *hyperplane* dapat berupa garis lurus, sementara untuk data berdimensi lebih tinggi, *hyperplane* dapat berbentuk bidang atau bahkan lebih kompleks lagi [18]. Parameter C mengontrol keseimbangan antara margin yang lebar dan kesalahan klasifikasi. Nilai C yang besar menekankan minimisasi kesalahan, sedangkan nilai C yang kecil menoleransi kesalahan demi margin yang lebih luas. Selama proses pelatihan, algoritma secara iteratif menyesuaikan bobot dan bias untuk menemukan *hyperplane* paling efektif dalam memisahkan sampel, dengan kata lain mencari bidang keputusan optimal yang membagi data ke dalam kelas-kelasnya [19]. Hasil pelatihan inilah yang kemudian digunakan pada tahap inferensi untuk mengklasifikasikan citra Hiragana baru secara akurat.



Gambar 3. Metode SVM

Proses klasifikasi SVM dibedakan menjadi dua jenis berdasarkan kernel yang digunakan, yaitu kernel linear dan

non-linear [20]. Untuk menangani permasalahan klasifikasi yang tidak dapat dipisahkan secara linear, SVM memanfaatkan teknik *kernel trick* digunakan oleh SVM dengan mentransformasikan data ke ruang berdimensi lebih tinggi agar data tersebut menjadi lebih mudah dipisahkan. Beberapa jenis kernel yang sering digunakan dalam metode SVM adalah RBF (*Radial Basis Function*) dan Polinomial. Perhitungan kernel-kernel tersebut terdapat pada Persamaan 1 dan 2.

1. Kernel RBF

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (1)$$

2. Kernel Polinomial

$$K(x_i, x_j) = (x_i \cdot x_j + c)^d \quad (2)$$

D. Confusion Matrix

Confusion matrix merupakan salah satu teknik yang digunakan untuk mengevaluasi performa model klasifikasi dengan membandingkan hasil prediksi model dengan kelas sebenarnya [20]. Di dalam *confusion matrix* terdapat empat elemen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Keempat komponen ini menjadi dasar dalam menghitung berbagai metrik evaluasi yang berfungsi untuk menilai seberapa baik kinerja suatu model klasifikasi. Beberapa metrik yang umum digunakan dalam evaluasi performa model terdapat pada Persamaan 3, 4, 5, dan 6.

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

$$Precision = \frac{TP}{(TP + FP)} \quad (4)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (5)$$

$$F1 - Score = \frac{(2 \times Recall \times Precision)}{(Recall + Precision)} \quad (6)$$

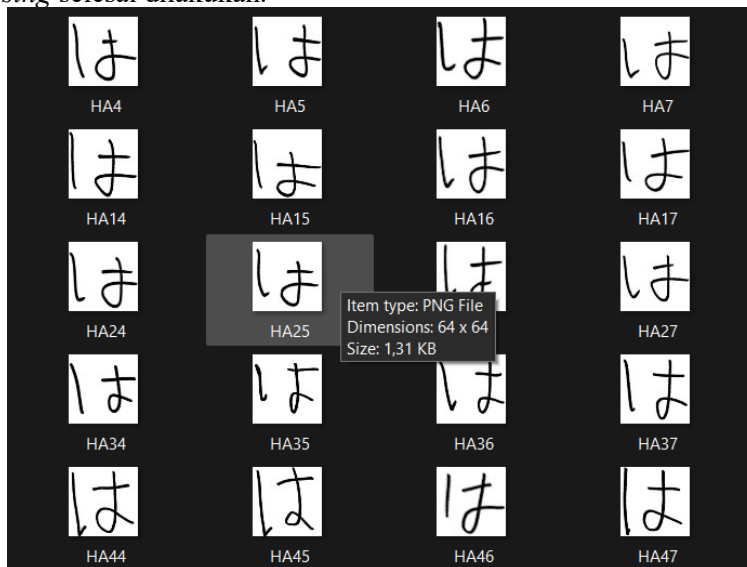
III. HASIL DAN PEMBAHASAN

Penelitian ini merancang beberapa skenario pengujian untuk model SVM. Setiap skenario ini memanfaatkan citra hasil *preprocessing* yang telah dilakukan *flatten* menjadi vektor sebelum dimasukkan ke model dengan menggunakan 2 jenis kernel, yaitu RBF dan Polinomial. Selain itu, tiap model akan diuji dengan pembagian data yang dibagi menjadi 90% data latih dan 10% data uji. Setiap skenario kemudian akan dievaluasi performanya berdasarkan data uji.

A. Preprocessing

Preprocessing merupakan tahap awal yang sangat penting dalam proses pengenalan pola citra tulisan tangan. Pada tahap ini, citra yang diperoleh dari dataset terlebih dahulu dipersiapkan agar dapat diolah lebih lanjut oleh algoritma klasifikasi. Salah satu langkah penting dalam *preprocessing* ini adalah konversi citra ke format *grayscale*, yaitu dengan mengubah gambar berwarna menjadi citra satu kanal intensitas. Langkah ini bertujuan untuk menyederhanakan informasi visual tanpa menghilangkan karakteristik bentuk yang dibutuhkan dalam proses klasifikasi. Dengan hanya mempertahankan informasi intensitas cahaya, proses komputasi menjadi lebih efisien dan ringan tanpa mengorbankan fitur penting dari karakter yang dikenali. Setelah citra dikonversi menjadi *grayscale*, tahap selanjutnya adalah melakukan penyesuaian ukuran. Karena dataset citra tulisan tangan yang digunakan memiliki ukuran bervariasi, perlu dilakukan *resizing* agar seluruh gambar memiliki dimensi yang seragam. Ukuran yang dipilih dalam penelitian ini adalah 64 x 64 piksel. Ukuran ini cukup kecil untuk mempercepat proses pelatihan model, namun tetap mampu mempertahankan detail yang relevan dari bentuk karakter huruf Hiragana. Keseragaman ukuran ini sangat penting agar setiap citra memiliki representasi fitur dengan dimensi yang sama, sehingga algoritma klasifikasi seperti SVM dapat mengolah data secara konsisten. Langkah terakhir dalam tahap *preprocessing* adalah proses *flattening*, yaitu mengubah citra 2 dimensi menjadi vektor 1 dimensi yang terdiri dari 4096 elemen. Hal ini dilakukan karena algoritma SVM menerima input dalam bentuk vektor fitur, bukan citra matriks. Setiap nilai intensitas piksel di dalam citra *grayscale* dimasukkan sebagai elemen dalam vektor, sehingga satu gambar akan direpresentasikan sebagai satu vektor yang siap dianalisis oleh model. Proses *flattening* ini memungkinkan informasi spasial dari citra tetap terwakili dalam bentuk numerik yang dapat dihitung oleh SVM untuk membentuk *hyperplane* pemisah antar kelas huruf Hiragana. Dengan cara ini, model mampu mempelajari pola-pola visual dari setiap huruf secara lebih efektif dan konsisten. Gambar 4

menunjukkan hasil konversi ukuran citra yang telah melalui tahap *grayscale* dan *resizing*, tersimpan di dalam folder setelah proses *preprocessing* selesai dilakukan.



Gambar 4. Hasil Konversi Citra

B. Model SVM dengan Kernel RBF

Pada model SVM dengan kernel RBF, skenario diujikan dengan nilai parameter C yang berbeda. Nilai C yang akan diuji yaitu 0,001 dan 1000, sedangkan nilai Gamma diatur secara default. Setelah dilakukan pengujian kernel dengan *hyperparameter*, tahap selanjutnya adalah mengevaluasi hasil performa model dan model dengan parameter terbaik akan dipilih berdasarkan hasil akurasi. Hasil akurasi setiap pembagian data terdapat pada Tabel 2.

TABEL II
HASIL AKURASI DENGAN MENGGUNAKAN KERNEL RBF

Proporsi Data	C	Gamma	Akurasi
90:10	0,001	<i>default</i>	0,8130
	1000	<i>default</i>	0,9304

Berdasarkan Tabel 2, hasil performa model dengan kernel RBF pada dua variasi nilai C yang berbeda menunjukkan hasil yang signifikan. Hasil pengujian model terbaik dihasilkan nilai C = 1000 dengan akurasi sebesar 93,04%. Dengan demikian, parameter C = 1000 merupakan parameter terbaik pada model SVM dengan menggunakan kernel RBF.

C. Model SVM dengan Kernel Polinomial

Pada model SVM dengan kernel polinomial, skenario diujikan dengan nilai parameter C yang berbeda. Nilai C yang akan diuji yaitu 0,001 dan 1000, sedangkan nilai Gamma diatur secara default. Setelah dilakukan pengujian kernel dengan *hyperparameter*, tahap selanjutnya adalah mengevaluasi hasil performa model dan model dengan parameter terbaik akan dipilih berdasarkan hasil akurasi. Hasil akurasi setiap pembagian data terdapat pada Tabel 3.

TABEL III
HASIL AKURASI DENGAN MENGGUNAKAN KERNEL POLINOMIAL

Proporsi Data	C	Gamma	Akurasi
90:10	0,001	<i>scale</i>	0.8913
	1000	<i>scale</i>	0,9261

Berdasarkan Tabel 3, hasil performa model dengan kernel polinomial pada dua variasi nilai C yang berbeda menunjukkan hasil yang signifikan. Hasil pengujian model terbaik dihasilkan nilai C = 1000 dengan akurasi sebesar 92,61%. Dengan demikian, parameter C = 1000 merupakan parameter terbaik pada model SVM dengan menggunakan kernel polinomial.

D. Evaluasi

Pada tahap ini, selain *confusion matrix* yang menampilkan distribusi prediksi benar dan salah untuk setiap huruf Hiragana, disertakan pula *classification report* yang memuat nilai *accuracy* dan jumlah sampel per kelas beserta rata-rata *macro* dan *weighted*. Kedua metrik ini saling melengkapi, karena *confusion matrix* memetakan pola

kesalahan antarkarakter sementara *classification report* menyajikan ukuran kinerja model secara kuantitatif dan rinci. Adapun hasil dari *classification report* terdapat pada Gambar 5.

Classification Report:				
	precision	recall	f1-score	support
A	1.00	1.00	1.00	5
CHI	1.00	1.00	1.00	5
E	1.00	1.00	1.00	5
FU	1.00	1.00	1.00	5
HA	0.71	1.00	0.83	5
HE	1.00	1.00	1.00	5
HI	1.00	0.80	0.89	5
HO	1.00	0.80	0.89	5
I	1.00	0.80	0.89	5
KA	1.00	1.00	1.00	5
KE	0.80	0.80	0.80	5
KI	1.00	1.00	1.00	5
KO	1.00	0.80	0.89	5
KU	0.83	1.00	0.91	5
MA	1.00	0.80	0.89	5
ME	0.75	0.60	0.67	5
MI	1.00	1.00	1.00	5
MO	1.00	1.00	1.00	5
MU	1.00	1.00	1.00	5
N	1.00	0.80	0.89	5
NA	0.83	1.00	0.91	5
NE	0.80	0.80	0.80	5
NI	1.00	1.00	1.00	5
NO	0.83	1.00	0.91	5
NU	0.80	0.80	0.80	5
O	1.00	1.00	1.00	5
RA	1.00	0.60	0.75	5
RE	0.80	0.80	0.80	5
RI	1.00	1.00	1.00	5
RO	1.00	1.00	1.00	5
RU	1.00	1.00	1.00	5
SA	0.83	1.00	0.91	5
SE	1.00	1.00	1.00	5
SHI	0.83	1.00	0.91	5
SO	1.00	1.00	1.00	5
SU	0.83	1.00	0.91	5
TA	1.00	1.00	1.00	5
TE	1.00	0.80	0.89	5
TO	1.00	0.80	0.89	5
TSU	1.00	1.00	1.00	5
U	0.83	1.00	0.91	5
WA	1.00	1.00	1.00	5
WO	0.83	1.00	0.91	5
YA	1.00	1.00	1.00	5
YO	0.83	1.00	0.91	5
YU	1.00	1.00	1.00	5
accuracy			0.93	230
macro avg	0.94	0.93	0.93	230
weighted avg	0.94	0.93	0.93	230

Gambar 5. Classification Report

Model SVM dengan kernel RBF dan parameter $C = 1000$ meraih akurasi keseluruhan 93,04 %, yang berarti hampir semua citra uji terklasifikasi dengan benar. Nilai *precision* rata-rata mencapai 93,84 %, menandakan bahwa sebagian besar prediksi benar memang sesuai kelasnya, sedangkan *recall* 93,04 % menunjukkan proporsi citra tiap huruf yang berhasil dikenali. *F1-score* 92,92 % menyeimbangkan kedua ukuran tersebut, memberi gambaran kinerja umum model. Tabel *classification report* memperlihatkan huruf-huruf seperti A, CHI, E, FU, HE, KI, MI, MO, MU, SU, TA, TSU, WA, YA, dan YU yang tercatat sempurna pada ketiga metrik yang dibuktikan dengan semua contoh berhasil dikenali tanpa satu pun kesalahan. Sebaliknya, kelas ME dan RA masih menjadi titik lemah, *precision* maupun *recall* turun ke kisaran 60–75 % karena beberapa citra ME terbaca sebagai NU atau KE, dan sebagian RA keliru dikenali sebagai RI. Kesulitan ini terjadi pada bentuk kait pendek dan lengkung mirip di antara huruf-huruf tersebut setelah citra diperkecil. Huruf lain yang sempat menurun, misalnya HA, KO, N, TE, dan TO, tetap memegang *precision* tinggi tetapi *recall* sedikit lebih rendah, pertanda model membuat sedikit FN ketika detail tinta tipis hilang. Meskipun demikian, sebagian besar kelas memiliki nilai di atas 0,80, sehingga tingkat akurasi pengenalan huruf Hiragana cukup konsisten. Secara keseluruhan, laporan klasifikasi ini menunjukkan bahwa konfigurasi RBF-SVM sudah memberikan hasil yang baik, namun kinerja model masih dapat diperbaiki dengan menambahkan variasi data pada huruf-huruf yang memiliki detail bentuk yang serupa dan rentan hilang. Adapun hasil *confusion matrix* yang ditunjukkan pada Gambar 6.

Dari Gambar 7, terdapat 4 hasil prediksi citra huruf Hiragana dimana 2 yang prediksinya benar dan 2 prediksinya yang salah. Dari empat gambar uji, dua huruf yang prediksinya benar yaitu huruf E dan TA, sedangkan dua lainnya yang salah prediksi adalah huruf KE yang dikenali sebagai huruf HA dan huruf I yang dikenali sebagai huruf SHI. Kesalahan ini muncul karena bentuk goresan pada huruf HA dan kait mirip pada huruf SHI cukup merupai lekukan utama pada huruf KE dan I. Saat gambar dilakukan *resize* menjadi 64 x 64 piksel, detail halus huruf hilang dan warnanya tampak lebih samar, sehingga pola garis yang terbaca model jadi lebih mirip dengan huruf lain. Maka dari itu, huruf-huruf yang mempunyai lengkungan pendek akan lebih mudah terjadi kesalahan prediksi. Dengan demikian, dapat diketahui bahwa prediksi huruf-huruf yang salah memprediksi adalah ketika huruf tersebut mempunyai pola yang mirip dengan huruf tertentu (7).

Pengujian ini menggunakan 2300 citra tulisan tangan yang dibuat pada kertas berwarna putih dengan tinta hitam, sehingga hasilnya bisa menurun jika diuji pada dataset yang lebih besar dan beragam. Kesalahan pengenalan utama paling sering muncul pada huruf-huruf dengan goresan serupa atau tinta yang tipis, menunjukkan pentingnya detail visual dalam tahap *preprocessing*. Meskipun hasilnya sudah cukup baik, pengujian masih terbatas pada citra hitam putih beresolusi 64 x 64 piksel, dengan variasi alat tulis, warna tinta, dan gaya tulisan yang belum sepenuhnya tercakup.

Untuk meningkatkan kualitas penelitian di masa depan, perlu dilakukan pengujian lanjutan dengan beberapa pendekatan tambahan. Pertama, meskipun metode SVM telah memberikan akurasi yang tinggi, teknik ekstraksi fitur seperti LBP atau HOG layak dicoba untuk menangkap pola goresan halus pada huruf Hiragana secara efektif dan akurat. Selain itu integrasi metode *deep learning* seperti *Convolution Neural Network* (CNN), dapat dieksplorasi karena cocok untuk mengolah dataset yang lebih besar dan beragam. Dengan menambah variasi dan jumlah dataset, model *deep learning* dapat belajar fitur yang lebih kompleks dan meningkatkan pengenalan huruf pada kondisi tulisan tangan yang bervariasi. Teknik *preprocessing* tambahan seperti augmentasi data, penyesuaian kontras, dan filter *Gaussian blur* juga dapat digunakan untuk meningkatkan kualitas cita serta mengurangi kesalahan akibat tinta tipis.

Penelitian ini berkontribusi signifikan dengan memanfaatkan metode SVM untuk mengenali citra tulisan tangan huruf Hiragana. Berbeda dengan penelitian sebelumnya yang dilakukan oleh Amrustian et al. (2021) [8], yang hanya menggunakan lima huruf vokal Hiragana. Penelitian ini mengumpulkan dataset sebanyak 46 huruf Hiragana yang dikumpulkan secara langsung dengan ditulis tangan, sehingga dataset yang dipakai lebih lengkap dan relevan dengan kondisi nyata. Selain itu, model SVM dengan penentuan nilai parameter juga terbukti efektif karena membantu model mencapai akurasi yang lebih tinggi (8).

IV. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem pengenalan citra tulisan tangan huruf Hiragana menggunakan metode SVM. Hasil pengujian menunjukkan bahwa metode ini mampu mengenali karakter Hiragana dengan tingkat akurasi yang tinggi, mencapai hingga 93.04% pada skenario terbaik menggunakan kernel RBF dengan parameter $C = 1000$ dan γ yang diatur secara *default* serta proporsi data latih dan uji sebesar 90:10. Metode SVM menunjukkan performa klasifikasi yang sangat baik dalam membedakan karakter yang serupa. Sistem ini berpotensi memberikan manfaat nyata dalam mendukung proses pembelajaran bahasa Jepang, khususnya dalam mengenali huruf Hiragana secara visual. Sistem ini juga dapat menjadi alat bantu edukatif yang bermanfaat dalam proses pembelajaran bahasa Jepang serta memiliki potensi untuk dikembangkan lebih lanjut dalam aplikasi pembelajaran interaktif.

DAFTAR PUSTAKA

- [1] O. Shinta Nurhidayati Salam and D. Pasanda, "IMPLIKASI HUKUM IMIGRASI JEPANG TERHADAP PEKERJA MIGRAN INDONESIA YANG OVERSTAY DI JEPANG," 2024, [Online]. Available: <https://binapatria.id/index.php/MBI>
- [2] I. Fardhani, N. Amrilizia, I. J. Fitriyah, and Y. Mulyati, "KURSUS BAHASA JEPANG SECARA DARING BAGI WARGA NEGARA INDONESIA DI JEPANG," 2021, [Online]. Available: <https://www.journal.itk.ac.id/index.php/sepakat>
- [3] P. Nur et al., "PENGENALAN HURUF JEPANG PADA ANAK-ANAK TAMAN PENDIDIKAN ALQURAN NURUL IMAN PONDOK MAJAPAHIT 2 MRANGGEN," 2021.
- [4] P. Raditha, C. Wardhani, Y. Mangaras, S. T. Florestiyanto, and M. Eng, "Pengenalan Karakter Tulisan Tangan Bahasa Jepang Hiragana menggunakan Support Vector Machine dan Scale Invariant Feature Transform Recognition Of Hiragana Japanese Handwriting Characters Using Support Vector Machine And Scale Invariant Feature Transform," *Jurnal Informatika dan Teknologi Informasi*, vol. 21, no. 2, pp. 135–144, 2024, doi: 10.31515/telematika.v21i2.12042.
- [5] O. P. Ayu and M. Riski Qisthiano, "IMPLEMENTASI TEKNOLOGI AI UNTUK SISTEM IDENTIFIKASI SPESIES HEWAN BERBASIS WEBSITE DENGAN PENDEKATAN MACHINE LEARNING," 2024.
- [6] Gefy Fitry Wijaya and Dwi Yuniarto, "Tinjauan Penerapan Machine Learning pada Sistem Rekomendasi Menggunakan Model Klasifikasi," *Populer: Jurnal Penelitian Mahasiswa*, vol. 3, no. 4, pp. 144–153, Dec. 2024, doi: 10.58192/populer.v3i4.2798.
- [7] R. Naa and A. Yuniar Rahman, "Klasifikasi Motif Kain Batik Papua Menggunakan Metode Multiclass Support Vector Machine (M-SVM)," vol. 11, no. 3, pp. 421–429, 2024.
- [8] M. A. Amrustian, V. F. Muliati, and E. E. Awal, "Studi Komparasi Metode Machine Learning untuk Klasifikasi Citra Huruf Vokal Hiragana," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 3, p. 905, Jul. 2021, doi: 10.30865/mib.v5i3.3083.

- [9] L. Amelia Putri, A. Sitorus, N. Fitriah, H. Virul, and S. Putri Rangkuti, "Pengolahan Citra Huruf Hijaiyah Menggunakan Algoritma Support Vector Machine," *Jurnal Ilmu Komputer Dan Teknologi Informasi*, vol. 2, no. 3, 2024, doi: 10.61132/neptunus.v2i2.168.
- [10] E. S. Panjaitan, H. Rumapea, and I. K. Jaya, "PENERAPAN METODE SUPPORT VECTOR MACHINE UNTUK PENGENALAN POLA AKSARA BATAK TOBA," vol. 4, no. 2, 2024, doi: 10.46880/tamika.Vol4No2(SEMNASTIK).pp49-55.
- [11] N. N. Arif, M. Magfirah, F. Febriani, R. Jannah, and M. Ula, "ANALISIS DAN PERBANDINGAN METODE CNN DAN SVM DALAM MENDETEKSI BATIK NUSANTARA," 2023.
- [12] A. Y. Yulestiono, M. M. Subagio, M. S. Bhakti, and A. P. Sari, "Perbandingan Kinerja Metode Convolutional Neural Network (CNN) dan VGG-16 dalam Klasifikasi Rambu Lalu Lintas," 2024.
- [13] D. Riana, D. U. E. Saputri, and S. Hadianti, "Klasifikasi Alexnet dan Deteksi Tepi Canny untuk Identifikasi Citra Repomedunm," *Jurnal Informasi dan Teknologi*, pp. 191–198, Apr. 2023, doi: 10.37034/jidt.v5i1.295.
- [14] R. S. Arhinza, A. P. Sari, and F. A. Akbar, "Klasifikasi Citra Aksara Lontara menggunakan K-NN dan Ekstraksi Fitur HOG," *Generation Journal*, vol. 8, no. 2, pp. 101–110, Sep. 2024, doi: 10.29407/gj.v8i2.22580.
- [15] S. Rahayu and Y. Yamasari, "Klasifikasi Penyakit Stroke dengan Metode Support Vector Machine (SVM)," *Journal of Informatics and Computer Science*, vol. 05, 2024.
- [16] N. E. W. Nugroho and A. Harjoko, "Transliteration of Hiragana and Katakana Handwritten Characters Using CNN-SVM," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 3, p. 221, Jul. 2021, doi: 10.22146/ijccs.66062.
- [17] S. Adiningsi, B. Pramono, A. M. Sajiah, and R. A. Saputra, "IDENTIFIKASI KUALITAS IKAN CAKALANG SEGAR BERBASIS CITRA MATA MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM) DENGAN FUNGSI KERNEL RADIAL BASIS FUNCTION," 2025.
- [18] H. Al Fathoni, A. Junaidi, and F. P. Aditiawan, "KLASIFIKASI TUTUPAN LAHAN PADA CITRA SENTINEL-2 DI KAWASAN IKN MENGGUNAKAN GOOGLE EARTH ENGINE," 2025.
- [19] S. F. Yunizar, A. P. Sari, and F. P. Aditiawan, "Pemanfaatan Model ResNet50 dan SVM untuk Klasifikasi Penyakit Daun Tebu," 2025, doi: 10.33050/cices.v11i1.3506.
- [20] F. M. Fathoni, C. A. Putra, and A. L. Nurlaili, "KLASIFIKASI PENYAKIT DAUN ANGGUR MENGGUNAKAN METODE K-NEAREST NEIGHBOR BERDASARKAN GRAY LEVEL CO-OCCURRENCE MATRIX," *Biner : Jurnal Ilmiah Informatika dan Komputer*, vol. 3, no. 1, pp. 8–15, Jan. 2024, doi: 10.32699/biner.v3i1.6332.