Jurnal Riset Multidisiplin Edukasi
https://journal.hasbaedukasi.co.id/index.php/jurmie
Halaman : 457 - 480

IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE (SVM) PADA APLIKASI KONVERSI TULISAN TANGAN KE DOKUMEN TEKS BERBASIS MOBILE

Sahrul Irfanul Husni Damanik[1], Muhammad Zulfansyuri Siambaton[2], Tasliyah Haramaini[3]
Universitas Islam Sumatera Utara, Medan, Indonesia [1,2,3]
Email: sahrulirfanulhusnidamanik@gmail.com[1], zulfansyuri@ft.uisu.ac.id[2] , tasliyah@ft.uisu.ac.id[3]

Informasi

Volume : 3
Nomor : 11
Bulan : November
Tahun : 2026
E-ISSN : 3062-9624

Abstract

The manual digitization of handwritten documents requires considerable time and effort and is prone to transcription errors (human error). Previous studies have developed website-based handwriting conversion applications using Deep Learning approaches, which require large datasets and substantial computational resources. This study aims to implement the Support Vector Machine (SVM) algorithm in a mobile-based handwriting-to-text document conversion application called INKORA and evaluate its accuracy in recognizing handwritten characters. The application was developed using the Flutter framework and integrated with a Python-based backend using FastAPI through an Application Programming Interface (API). Handwritten images captured using a camera or selected from the gallery undergo several preProcessing stages, including Grayscale conversion, binarization, noise reduction, character segmentation, and size normalization to 28 × 28 pixels. The features of each character are extracted using Histogram of Oriented Gradients (HOG) with 9 orientations, a cell size of 4 × 4 pixels, and a block size of 2 × 2 cells, resulting in 1,296 features. These features are then classified using an SVM with a Radial Basis Function (RBF) kernel into 52 Latin alphabet classes, consisting of uppercase A–Z and lowercase a–z letters. The model was trained and tested using the EMNIST ByClass dataset, comprising 52,000 training samples and 5,200 testing samples. The evaluation results indicate a classification accuracy of 74.50%. Classification errors primarily occur among letters with similar shapes, particularly between uppercase and lowercase characters. Testing under various image conditions shows that conversion results are influenced by paper color and type, image sharpness, writing inclination, and inter-character spacing. Therefore, the combination of HOG and SVM can be implemented in a mobile application for handwriting conversion, although the accuracy of word and sentence conversion remains affected by image quality and character segmentation Processes.

Keyword: *Support Vector Machine, Histogram of Oriented Gradients, Mobile Application, Handwriting Conversion, EMNIST.*

Abstrak

Digitalisasi dokumen tulisan tangan yang dilakukan secara manual membutuhkan waktu dan tenaga yang besar serta rentan terhadap kesalahan penulisan ulang (human error). Penelitian sebelumnya telah mengembangkan aplikasi konversi tulisan tangan berbasis website menggunakan pendekatan Deep Learning yang membutuhkan dataset dalam jumlah besar dan sumber daya komputasi yang tinggi. Penelitian ini bertujuan mengimplementasikan algoritma Support Vector Machine (SVM) pada aplikasi konversi tulisan tangan ke dokumen teks berbasis Mobile bernama INKORA serta mengetahui tingkat akurasinya dalam mengenali karakter tulisan tangan. Aplikasi dibangun menggunakan

framework Flutter dan terhubung dengan backend Python berbasis FastAPI melalui Application Programming Interface (API). Citra tulisan tangan yang diperoleh dari kamera atau galeri diproses melalui tahapan konversi Grayscale, binerisasi, pengurangan noise, segmentasi karakter, dan normalisasi ukuran menjadi 28 × 28 piksel. Fitur setiap karakter diekstraksi menggunakan Histogram of Oriented Gradients (HOG) dengan 9 orientasi, ukuran sel 4 × 4 piksel, dan blok 2 × 2 sel sehingga menghasilkan 1.296 fitur. Fitur tersebut kemudian diklasifikasikan menggunakan SVM dengan kernel Radial Basis Function (RBF) ke dalam 52 kelas huruf Latin A–Z dan a–z. Model dilatih dan diuji menggunakan dataset EMNIST ByClass dengan 52.000 data latih dan 5.200 data uji. Hasil pengujian menunjukkan akurasi klasifikasi sebesar 74,50%. Kesalahan klasifikasi terutama terjadi pada huruf yang memiliki kemiripan bentuk antara huruf kapital dan huruf kecil. Pengujian pada berbagai kondisi citra menunjukkan bahwa hasil konversi dipengaruhi oleh warna dan jenis kertas, ketajaman gambar, kemiringan, serta jarak antarhuruf. Dengan demikian, kombinasi HOG dan SVM dapat diimplementasikan pada aplikasi Mobile untuk konversi tulisan tangan, meskipun ketepatan konversi kata dan kalimat masih dipengaruhi oleh kualitas citra dan proses segmentasi karakter.

Kata Kunci : *Support Vector Machine, Histogram of Oriented Gradients, Aplikasi Mobile, Konversi Tulisan Tangan, EMNIST.*

A. PENDAHULUAN

Dalam era digitalisasi yang terus berkembang, kebutuhan untuk mengubah informasi dari bentuk fisik menjadi bentuk digital semakin meningkat di berbagai bidang, mulai dari pendidikan, medis, hukum, hingga administrasi perkantoran. Salah satu bentuk informasi yang paling sering dijumpai namun paling sulit diproses secara otomatis adalah tulisan tangan. Berbeda dengan teks cetak yang memiliki bentuk huruf seragam, tulisan tangan memiliki variasi bentuk, ukuran, kemiringan, dan gaya penulisan yang berbeda-beda antar individu, sehingga pengenalan tulisan tangan menjadi tantangan tersendiri dalam bidang pengolahan citra digital dan kecerdasan buatan (Wijaya, 2019).

Dalam penelitian Santri (2025), membangun aplikasi berbasis website menggunakan Tesseract.js dengan pendekatan Deep Learning, khususnya Convolutional Neural Network (CNN), untuk mengenali tulisan tangan yang diunggah dalam bentuk gambar. Meskipun pendekatan tersebut menunjukkan hasil yang cukup baik, terdapat beberapa kelemahan yang dapat dijadikan celah pengembangan. Secara umum, Deep Learning membutuhkan dataset dalam jumlah besar dan sumber daya komputasi yang tinggi agar dapat memberikan hasil yang optimal, sehingga kurang efisien apabila diterapkan pada perangkat dengan spesifikasi terbatas. Beberapa studi komparasi bahkan menunjukkan bahwa performa algoritma Deep Learning dapat menurun signifikan atau mengalami overfitting ketika data latih yang tersedia relatif terbatas, sementara algoritma pembelajaran mesin klasik seperti Support Vector Machine (SVM) tetap mampu memberikan hasil yang stabil pada kondisi tersebut (Naufal, 2021). Selain itu, aplikasi berbasis website mengharuskan pengguna untuk selalu terhubung

dengan browser, serta kurang optimal apabila digunakan untuk pengambilan gambar secara langsung melalui kamera dibandingkan aplikasi yang berjalan secara native pada perangkat Mobile.

Berdasarkan permasalahan tersebut, penelitian ini berupaya memberikan solusi alternatif dengan menggunakan algoritma klasifikasi yang berbeda, yaitu Support Vector Machine (SVM), serta mengubah basis pengembangan aplikasi dari website menjadi aplikasi Mobile. SVM merupakan algoritma pembelajaran mesin yang bekerja dengan mencari Hyperplane terbaik untuk memisahkan data ke dalam beberapa kelas dengan margin pemisah yang maksimal. Beberapa penelitian telah membuktikan efektivitas SVM dalam mengenali pola tulisan tangan, misalnya pada pengenalan pola tanda tangan dengan akurasi yang baik (Gunawan & Reswan, 2021), identifikasi kepribadian melalui tulisan tangan dengan akurasi di atas 97% (Rosyidah & Rochmawati, 2019), serta identifikasi penulis melalui pola tulisan tangan menggunakan kombinasi fitur GLCM dan histogram (Handhayani, 2017). Bahkan pada beberapa kasus klasifikasi citra, SVM tercatat mampu memberikan akurasi yang lebih tinggi dibandingkan CNN ketika dikombinasikan dengan ekstraksi fitur yang tepat (Prasetyo dkk., 2023), sehingga menunjukkan bahwa SVM tetap menjadi alternatif yang relevan untuk dibandingkan dengan pendekatan Deep Learning, khususnya pada aplikasi dengan keterbatasan sumber daya komputasi seperti aplikasi Mobile.

Untuk mendukung proses klasifikasi menggunakan SVM, penelitian ini menggunakan metode ekstraksi fitur Histogram of Oriented Gradients (HOG), Adapun pemilihan basis aplikasi Mobile didasarkan pada pertimbangan bahwa perangkat Mobile telah menjadi perangkat yang paling banyak digunakan masyarakat sehari-hari dan umumnya sudah dilengkapi kamera beresolusi tinggi, serta framework Flutter kini banyak digunakan dalam pengembangan aplikasi Mobile lintas platform karena kemudahan dan performanya (Rahmawati & Sari, 2024; Sudradjat, 2021).

Berdasarkan uraian latar belakang yang telah dijelaskan, rumusan masalah dalam penelitian ini adalah bagaimana mengatasi kesulitan dalam mengonversi tulisan tangan menjadi dokumen teks secara praktis melalui perangkat mobile, serta bagaimana mengetahui tingkat ketepatan pengenalan tulisan tangan yang memiliki variasi bentuk dan karakter penulisan.

Untuk memfokuskan penelitian agar sesuai dengan tujuan yang ingin dicapai, batasan masalah dalam penelitian ini meliputi pengembangan aplikasi mobile berbasis Android menggunakan framework Flutter dengan bahasa pemrograman Dart, sedangkan proses

pelatihan dan klasifikasi menggunakan algoritma Support Vector Machine (SVM) dengan bahasa pemrograman Python. Citra masukan yang digunakan berupa tulisan tangan dengan format .jpg, .jpeg, atau .png yang diperoleh melalui kamera maupun galeri perangkat. Pengenalan tulisan tangan dibatasi pada huruf alfabet Latin A–Z dan a–z, tanpa mencakup angka, tanda baca, tulisan bersambung, simbol, karakter khusus, aksara daerah, maupun karakter non-Latin. Selain itu, citra masukan harus memiliki kondisi tulisan yang terlihat jelas dan tidak mengalami gangguan berat, seperti buram, pencahayaan ekstrem, tulisan terpotong, atau karakter yang saling bertumpuk. Hasil konversi tulisan tangan dibatasi pada penyimpanan teks digital dalam format Teks (.txt) dan Microsoft Word (.docx), serta tidak mencakup pembentukan dokumen dalam format lainnya.

Tujuan yang ingin dicapai dalam penelitian ini adalah mengimplementasikan algoritma Support Vector Machine (SVM) pada aplikasi konversi tulisan tangan menjadi dokumen teks berbasis mobile, serta mengetahui tingkat akurasi algoritma Support Vector Machine (SVM) dalam mengenali tulisan tangan berdasarkan fitur yang diekstraksi menggunakan metode Histogram of Oriented Gradients (HOG).

B. METODE PENELITIAN

Penelitian ini dilaksanakan di Laboratorium Fakultas Teknik, Program Studi Teknik Informatika, Universitas Islam Sumatera Utara (UISU), Medan, selama kurang lebih empat bulan, yaitu Juli hingga Oktober. Penelitian menggunakan metode Research and Development (R&D) yang bertujuan mengembangkan aplikasi mobile berbasis Android untuk mengonversi tulisan tangan menjadi teks digital menggunakan algoritma Support Vector Machine (SVM) dan ekstraksi fitur Histogram of Oriented Gradients (HOG). Perangkat yang digunakan meliputi laptop MSI Modern 14 C11M dan smartphone Android Infinix HOT 11S NFC, dengan dukungan perangkat lunak Flutter, Dart, Python, FastAPI, Scikit-learn, OpenCV, scikit-image, NumPy, Pillow, dan Microsoft Word. Tahapan penelitian mencakup studi literatur, analisis kebutuhan, perancangan sistem, pengembangan aplikasi, pengujian, evaluasi, perbaikan, dan penyusunan laporan. Aplikasi dirancang untuk mengambil atau memilih gambar tulisan tangan melalui kamera maupun galeri, memproses citra, mengenali huruf kapital dan huruf kecil alfabet Latin, menampilkan hasil konversi, serta menyimpannya dalam format dokumen digital.

Proses pengenalan tulisan tangan dilakukan melalui beberapa tahapan, yaitu konversi citra menjadi grayscale, binerisasi, penghilangan noise, segmentasi baris, kata, dan karakter, serta normalisasi ukuran karakter menjadi 28 × 28 piksel. Selanjutnya, metode HOG digunakan

untuk mengekstraksi karakteristik bentuk dan arah gradien huruf menjadi vektor fitur numerik sebanyak 1.296 fitur, kemudian diklasifikasikan menggunakan algoritma SVM dengan kernel Radial Basis Function (RBF) berdasarkan dataset EMNIST ByClass yang terdiri atas 52 kelas karakter. Hasil klasifikasi disusun menjadi kata dan kalimat, kemudian dilakukan koreksi otomatis menggunakan metode Levenshtein Distance berbasis kamus untuk membantu memperbaiki kesalahan pengenalan. Pengujian dilakukan melalui metode Black Box Testing untuk mengevaluasi fungsi aplikasi serta pengujian kinerja algoritma dengan membandingkan hasil prediksi terhadap label sebenarnya guna memperoleh nilai akurasi. Selain itu, pengujian penerimaan pengguna dilakukan melalui kuesioner untuk mengetahui penilaian responden terhadap aplikasi yang dikembangkan.

C. HASIL DAN PEMBAHASAN

Implementasi Sistem

Hasil Pengujian Akurasi Model pada Dataset EMNIST ByClass

Tabel 1 Hasil Pengujian Akurasi SVM

No.	Parameter Pengujian	Hasil
1	Dataset	EMNIST *ByClass*
2	Jumlah kelas karakter	52 kelas
3	Karakter yang digunakan	A–Z dan a–z
4	Jumlah data latih per kelas	1.000 gambar
5	Jumlah keseluruhan data latih	52.000 gambar
6	Jumlah data uji per kelas	100 gambar
7	Jumlah keseluruhan data uji	5.200 gambar
8	Jumlah prediksi benar dari s	3.874 gambar
9	Jumlah prediksi salah	1.326 gambar
10	Akurasi pengujian	74,50%
11	Tingkat kesalahan	25,50%

Berdasarkan Tabel 1, hasil pengujian menunjukkan bahwa model SVM memperoleh tingkat akurasi sebesar 74,50%. Dari total 5.200 gambar data uji, sebanyak 3.874 gambar berhasil diklasifikasikan dengan benar, sedangkan 1.326 gambar lainnya mengalami kesalahan klasifikasi.

Berdasarkan perhitungan tersebut, dapat diketahui bahwa model mampu mengenali sekitar 74,50% karakter dari keseluruhan data uji. Sementara itu, sebesar 25,50% karakter lainnya belum berhasil diklasifikasikan dengan tepat.

Hasil Tampilan Antarmuka Pengguna (User Interface)

Hasil Implementasi Halaman Utama

Halaman utama merupakan tampilan awal yang digunakan pengguna untuk mengakses fitur konversi tulisan tangan pada aplikasi INKORA. Halaman ini menampilkan identitas aplikasi, informasi penggunaan metode *Histogram of Oriented Gradients* (HOG) dan *Support Vector Machine* (SVM), serta dua pilihan sumber gambar, yaitu kamera dan galeri.

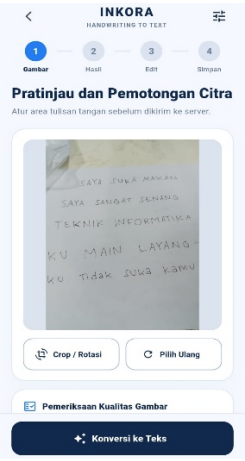


Gambar 1 Hasil Implementasi Halaman Utama Aplikasi INKORA

Pengguna dapat memilih menu Ambil Foto untuk mengambil gambar tulisan tangan secara langsung menggunakan kamera perangkat, sedangkan menu Pilih dari Galeri digunakan untuk mengambil gambar yang telah tersimpan. Selain itu, halaman utama menyediakan informasi koneksi server dan petunjuk penggunaan aplikasi untuk membantu pengguna memahami tahapan konversi.

Hasil Implementasi Pratinjau dan Pemotongan Citra

Halaman pratinjau dan pemotongan citra digunakan untuk menampilkan gambar tulisan tangan yang dipilih pengguna sebelum dilakukan proses konversi. Pada halaman ini, pengguna dapat menyesuaikan area gambar melalui fitur pemotongan (*crop*) dan rotasi agar bagian tulisan yang akan dikenali dapat ditentukan dengan lebih tepat.



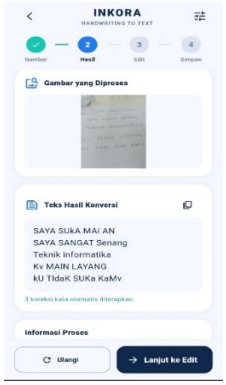
Gambar 2 Hasil Implementasi Halaman Pratinjau dan Pemotongan Citra

Fitur ini bertujuan untuk mengurangi bagian gambar yang tidak diperlukan, seperti area kosong atau objek di sekitar tulisan. Setelah gambar disesuaikan, pengguna dapat menekan

tombol Konversi ke Teks untuk mengirimkan citra menuju *backend* *Fast*API dan memulai proses pengenalan tulisan tangan menggunakan SVM dan HOG.

Hasil Implementasi Halaman Hasil Konversi

Halaman hasil konversi digunakan untuk menampilkan keluaran dari proses pengenalan tulisan tangan. Setelah citra dikirimkan ke *backend*, sistem melakukan prapemrosesan citra, segmentasi karakter, ekstraksi fitur HOG, dan klasifikasi karakter menggunakan model SVM.



Gambar 3 Hasil Implementasi Halaman Hasil Konversi

Hasil pengenalan ditampilkan dalam bentuk teks digital yang dapat dibaca oleh pengguna. Halaman ini juga menampilkan citra yang diproses serta informasi pendukung, seperti metode pengenalan dan waktu pemrosesan. Pengguna dapat membandingkan teks hasil konversi dengan citra tulisan tangan sebelum melanjutkan ke tahap penyuntingan.

Hasil Implementasi Fitur Penyuntingan Teks

Fitur penyuntingan teks merupakan fasilitas yang disediakan untuk memperbaiki hasil pengenalan tulisan tangan apabila terdapat karakter atau kata yang belum sesuai dengan citra aslinya. Pada halaman ini, teks hasil konversi ditampilkan dalam area penyuntingan sehingga pengguna dapat melakukan perubahan secara langsung.



Gambar 4 Hasil Implementasi Fitur Penyuntingan Teks

Selain penyuntingan manual, aplikasi menyediakan saran koreksi kata sebagai bantuan untuk memperbaiki sebagian kesalahan pengenalan. Pengguna dapat memeriksa saran yang tersedia dan menentukan hasil teks yang sesuai. Setelah proses penyuntingan selesai, pengguna dapat melanjutkan ke halaman penyimpanan dokumen.

Hasil Implementasi Fitur Penyimpanan Dokumen

Halaman penyimpanan dokumen merupakan tahap akhir dalam proses penggunaan aplikasi INKORA. Fitur ini digunakan untuk menyimpan hasil konversi tulisan tangan yang telah diperiksa atau diperbaiki menjadi dokumen digital dengan format *Microsoft Word* (.docx).



Gambar 5 Hasil Implementasi Fitur Penyimpanan Dokumen

Pengguna dapat menentukan nama dokumen dan memilih lokasi penyimpanan melalui perangkat *Android*. Dokumen yang dihasilkan dapat dibuka kembali menggunakan aplikasi pengolah kata yang mendukung format .docx, sehingga hasil konversi dapat digunakan untuk keperluan penulisan, pengarsipan, maupun pengolahan dokumen lebih lanjut.

Secara keseluruhan, antarmuka aplikasi INKORA dirancang untuk mendukung proses konversi tulisan tangan secara bertahap dan terstruktur. Integrasi antara aplikasi *Flutter* dan *backend Fast*API memungkinkan citra tulisan tangan diproses menggunakan metode SVM dan HOG, sedangkan fitur penyuntingan dan penyimpanan dokumen memberikan kemudahan bagi pengguna dalam memanfaatkan teks digital yang dihasilkan.

Hasil Implementasi Algoritma *Support Vector Machine* (SVM)

Implementasi algoritma *Support Vector Machine* (SVM) pada aplikasi INKORA dilakukan untuk mengklasifikasikan karakter tulisan tangan berdasarkan fitur yang diperoleh melalui metode *Histogram of Oriented Gradients* (HOG). Algoritma SVM berperan dalam menentukan kelas huruf dari setiap karakter yang telah melalui tahapan prapemrosesan citra, segmentasi, normalisasi ukuran, dan ekstraksi fitur. Proses klasifikasi ini dijalankan pada *backend* yang dikembangkan menggunakan bahasa pemrograman *Python*.

Pada penelitian ini, model SVM dilatih menggunakan dataset EMNIST *ByClass* yang difokuskan pada 52 kelas karakter huruf Latin, terdiri atas 26 huruf kapital (A–Z) dan 26 huruf kecil (a–z). Data gambar karakter yang digunakan dalam pelatihan diproses menjadi citra berukuran 28 × 28 piksel. Selanjutnya, setiap citra diekstraksi menggunakan HOG untuk menghasilkan vektor fitur numerik yang menjadi masukan bagi algoritma SVM.

Model yang telah dilatih kemudian disimpan dalam format .joblib agar dapat digunakan kembali pada proses pengenalan tulisan tangan tanpa melakukan pelatihan ulang. Model tersebut memuat komponen klasifikasi serta pemetaan label yang digunakan untuk mengubah hasil prediksi menjadi karakter huruf.

Hasil Pengujian Sistem (Evaluasi)

Hasil Fungsional Aplikasi (Black Box Testing)

Tabel 2 Hasil Pengujian Fungsional Aplikasi INKORA

No.	Fitur	Hasil yang Diharapkan	Hasil Pengujian
1	Koneksi API	Aplikasi terhubung dengan *backend Fast*API	Berhasil
2	Kamera	Gambar tulisan tangan dapat diambil	Berhasil
3	Galeri	Gambar dapat dipilih dari penyimpanan	Berhasil
4	Crop dan rotasi	Area dan orientasi gambar dapat disesuaikan	Berhasil
5	Konversi	Hasil pengenalan ditampilkan sebagai teks	Berhasil
6	Penyuntingan	Pengguna dapat memperbaiki hasil konversi	Berhasil
7	Penyimpanan Word	Teks berhasil disimpan dalam format .docx	Berhasil
8	Pengenalan huruf	Pada foto uji berisi 18 huruf A (3 baris, masing-masing 6 huruf), satu huruf A terpecah menjadi dua potongan sehingga baris kedua terbaca 7 huruf. Hasil yang diharapkan adalah 6 huruf per baris.	Tidak Berhasil
9	Pemisahan huruf	kata “Sahrul” terbaca Sqhrul, yaitu hanya 5 dari 6 huruf yang benar	Tidak Berhasil
10	Ganggaun citra terlalu berat (coretan dll.)	Tidak dapat di proses oleh sistem	Tidak Berhasil

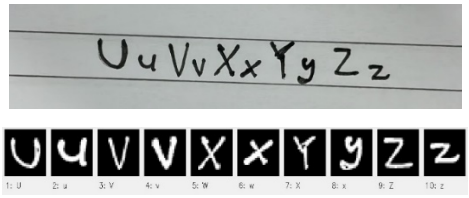
Berdasarkan Tabel 2, tujuh skenario pengujian fungsional utama yang telah dilaksanakan menunjukkan hasil berhasil. Persentase keberhasilan pengujian dihitung sebagai berikut.

Keberhasilan = (Jumlah pengujian berhasil / Jumlah pengujian dilaksanakan) × 100%

Keberhasilan = (7 / 10) × 100% = 70%

Hasil tersebut menunjukkan bahwa seluruh skenario fungsional utama yang diuji telah berjalan sesuai dengan keluaran yang diharapkan. Namun, keberhasilan fungsional tidak menunjukkan tingkat akurasi pengenalan tulisan tangan. Oleh karena itu, evaluasi terhadap algoritma SVM dan HOG dilakukan secara terpisah.

Hasil Prapemrosesan Citra



Gambar 6 Perbandingan Citra Asli dan Hasil Prapemrosesan

Proses prapemrosesan pada sistem INKORA meliputi konversi citra menjadi *Grayscale*, binerisasi, pengurangan gangguan citra, serta penanganan garis dan kemiringan gambar. Hasil pengolahan kemudian digunakan sebagai masukan pada proses segmentasi karakter.

Pengujian dilakukan dengan membandingkan gambar asli dan hasil prapemrosesan. Perbandingan tersebut digunakan untuk mengamati apakah bentuk tulisan tetap terjaga dan area karakter dapat dibedakan dari latar belakang.

Hasil Pengaruh Kondisi Citra

Tabel 3 Hasil Pengujian Pengaruh Kondisi Citra

Kondisi Citra, Hasil dari aplikasi, persentase (%), dan keterangan
1. Kertas Putih Polos
 mendapatkan hasil 96% (12,5:13) Keterangan pada gambar di atas yaitu: pada huruf “k” seharusnya Huruf “K”, mendapat pengungurangan hasil 0,5 poin karena hasil ragu.

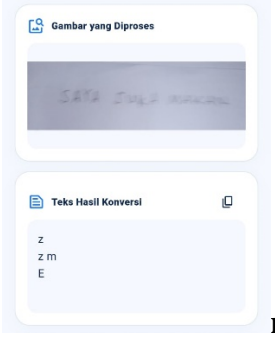
2. Kertas bergaris



mendapatkan hasil 80% (10:2)

Keterangan pada gambar di atas yaitu: pada huruf "t" salah penyebutan mungkin system mengira *noise* pada garis di Tengah huruf, dan huruf "J", "u" ragu, mendapat pengungurangan hasil 2 poin karena hasil salah dan ragu.

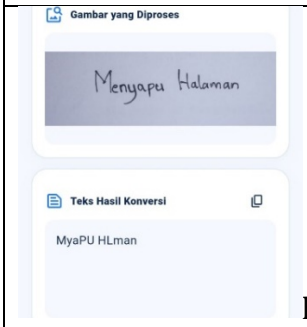
3. Gambar buram



mendapatkan hasil 0%

Keterangan pada gambar di atas yaitu: citra tidak dapat terdeteksi

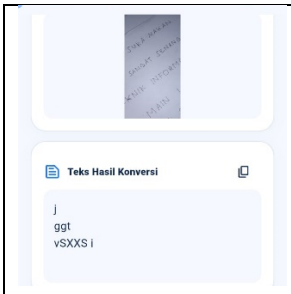
4. Huruf saling berdekatan



hasil 42% (6:14)

Keterangan pada gambar di atas yaitu: sistem tidak mendukung pada huruf yang terlalu dekat pada satu sama lain karena terhitung menjadi satu kelas pada beberapa huruf.

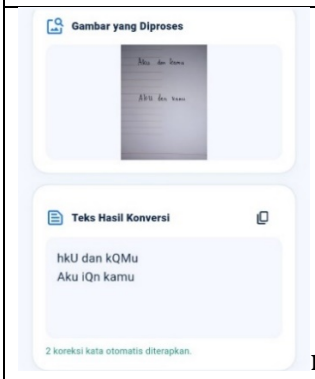
5. Gambar miring



mendapatkan hasil 0%

Keterangan pada gambar di atas yaitu: citra tidak dapat terdeteksi karena, hanya huruf lurus yang bisa di proses pada sistem

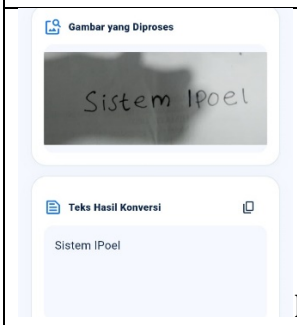
6. Kertas semi polos dan bergaris



mendapatkan hasil 70% (14:20)

Keterangan pada gambar di atas yaitu: karakter tulisan bervariasi

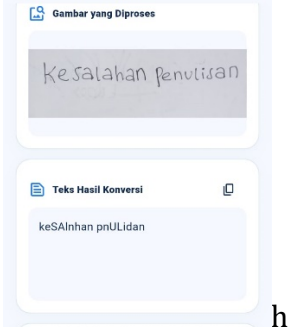
7. Pencahayaan rendah



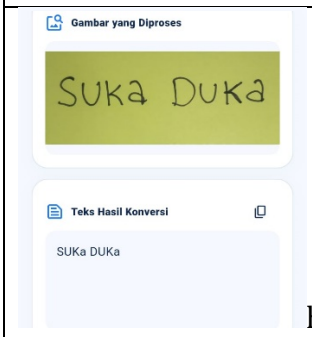
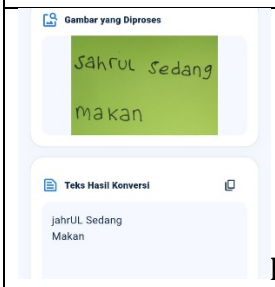
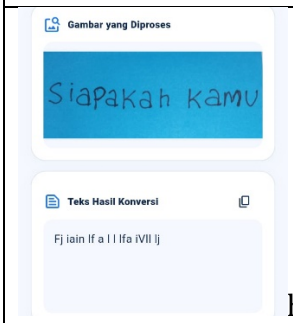
hasil 100%

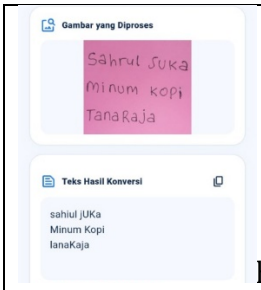
Keterangan pada gambar di atas yaitu: gaya huruf tulisan sesuai dengan data sistem yang dilatih

8. Kertas gangguan huruf timbul



hasil 33% (6:18)

Keterangan pada gambar di atas yaitu: gangguan huruf di tandai pada huruf sebaliknya ikut terdeteksi oleh sistem.
9. Kertas warna kuning
 hasil 100% Keterangan pada gambar di atas yaitu: karakter tulisan sesuai dengan data latih
10. Kertas warna hijau
 hasil 88% (15:17) Keterangan pada gambar di atas yaitu: pada huruf “S” menyerupai huruf j maka terbaca salah.
11. Kertas warna biru
 hasil 0% Keterangan pada gambar di atas yaitu: pada kali ini sistem tidak mendukung untuk memproses kertas warna biru
12. Kertas warna pink



hasil 85% (23:27)

Keterangan pada gambar di atas yaitu: cukup baik untuk proses pada kertas warna pink, kesalahan hanya di variasi tulisan huruf yang tidak sesuai pada data sistem.

Berdasarkan Tabel 3, hasil pengujian digunakan untuk mengidentifikasi kondisi citra yang memberikan tingkat kesalahan paling rendah maupun paling tinggi. Perbedaan hasil tersebut selanjutnya dianalisis untuk mengetahui pengaruh kondisi citra terhadap proses prapemrosesan, segmentasi karakter, dan untuk pembacaan huruf di pengaruhi dari akurasi dataset.

Pembahasan

Analisis Pengujian Fungsional Aplikasi

Berdasarkan hasil pengujian fungsional menggunakan metode *Black Box Testing*, tujuh skenario utama pada aplikasi INKORA telah berhasil dijalankan sesuai dengan fungsi yang diharapkan. Skenario tersebut meliputi koneksi aplikasi dengan *backend Fast*API, pengambilan gambar melalui kamera, pemilihan gambar dari galeri, pemotongan dan rotasi citra, proses konversi tulisan tangan, penyuntingan teks, serta penyimpanan hasil dalam format *Microsoft Word* (.docx).

Persentase keberhasilan fungsional yang diperoleh dari sepuluh skenario pengujian tersebut adalah 70%. Hasil ini menunjukkan bahwa komponen antarmuka aplikasi *Flutter* dapat digunakan untuk menjalankan tahapan konversi tulisan tangan, mulai dari penerimaan citra hingga penyimpanan dokumen digital.

Komunikasi antara aplikasi mobile dan *backend Fast*API juga dapat dilakukan melalui jaringan Wi-Fi. Hal ini menunjukkan bahwa proses pengiriman citra dari perangkat *Android* menuju *backend* dan penerimaan hasil pengenalan telah berfungsi sesuai dengan rancangan sistem.

Analisis Prapemrosesan dan Segmentasi Karakter

Tahap prapemrosesan dan segmentasi karakter memiliki peranan penting dalam menentukan kualitas citra yang diberikan kepada model SVM. Pada sistem INKORA, citra tulisan tangan terlebih dahulu melalui pengolahan gambar untuk memperjelas bentuk tulisan dan mengurangi gangguan sebelum dilakukan pemisahan karakter.

Proses prapemrosesan meliputi konversi citra menjadi *Grayscale*, binerisasi, serta penanganan gangguan berupa garis atau kemiringan gambar. Tahapan ini bertujuan untuk menghasilkan representasi tulisan yang lebih sesuai bagi proses segmentasi.

Berdasarkan pengamatan selama pengembangan sistem, salah satu kendala yang ditemukan adalah karakter tulisan tangan yang memiliki jarak terlalu dekat atau saling terhubung. Kondisi tersebut dapat menyebabkan dua karakter terdeteksi sebagai satu bagian, sehingga jumlah karakter hasil segmentasi tidak sesuai dengan jumlah karakter pada teks acuan.

Sebagai contoh, pada salah satu pengujian pengembangan, kata yang seharusnya terbaca sebagai "buku" menghasilkan prediksi "bum". Setelah dilakukan percobaan pemisahan karakter yang saling menempel, hasil prediksi berubah menjadi "buku". Kondisi tersebut menunjukkan bahwa ketepatan pemisahan karakter berpengaruh terhadap hasil klasifikasi.

Namun, pemisahan karakter juga memiliki keterbatasan. Karakter tunggal yang secara alami mempunyai bentuk lebar, seperti huruf m dan w, dapat mengalami pemotongan yang tidak tepat apabila sistem hanya mempertimbangkan ukuran karakter.

Dengan demikian, segmentasi merupakan salah satu faktor yang memengaruhi kemampuan sistem mengenali tulisan tangan. Kesalahan pada tahap ini dapat menyebabkan hasil klasifikasi menjadi tidak sesuai meskipun model SVM telah berhasil melakukan proses prediksi.

Analisis Kinerja Algoritma SVM

Implementasi metode HOG dan algoritma SVM dilakukan untuk mengenali 52 kelas karakter huruf Latin, yang terdiri atas 26 huruf kapital dan 26 huruf kecil. Metode HOG digunakan untuk menghasilkan vektor fitur berdasarkan distribusi arah gradien pada citra karakter, sedangkan SVM digunakan untuk menentukan kelas karakter berdasarkan fitur yang diperoleh.

Model utama yang digunakan dalam aplikasi INKORA dilatih menggunakan 52.000 sampel dari dataset EMNIST *ByClass*, dengan jumlah 1.000 sampel untuk setiap kelas huruf. Model menggunakan kernel *Radial Basis Function* (RBF) untuk melakukan klasifikasi terhadap fitur yang telah diekstraksi.

Berdasarkan hasil evaluasi model utama pada 5.200 sampel pengujian EMNIST, diperoleh akurasi klasifikasi sebesar 74,50%. Hasil tersebut menunjukkan bahwa model mampu mengenali sebagian besar karakter pada data pengujian, tetapi masih ditemukan kesalahan klasifikasi pada sejumlah kelas huruf.

Kesalahan tersebut dapat dipengaruhi oleh kemiripan bentuk antarhuruf, terutama karakter kapital dan huruf kecil yang memiliki karakteristik visual hampir sama. Sebagai contoh, karakter O dan o, C dan c, serta I dan l dapat memiliki bentuk yang sulit dibedakan apabila hanya diamati berdasarkan karakteristik visual tanpa mempertimbangkan konteks kata.

Selain itu, perbedaan bentuk tulisan tangan antar individu juga memengaruhi hasil klasifikasi. Karakter dengan label yang sama dapat memiliki bentuk, ketebalan goresan, dan kemiringan yang berbeda. Kondisi tersebut menyebabkan model harus mengenali variasi fitur dari setiap kelas karakter.

Hasil evaluasi tersebut menunjukkan bahwa penerapan SVM dan HOG dapat digunakan untuk melakukan klasifikasi karakter tulisan tangan, tetapi tingkat ketepatannya masih dipengaruhi oleh karakteristik dataset dan variasi bentuk tulisan.

Analisis Hasil Konversi Kata dan Kalimat

Pengujian konversi kata dan kalimat dilakukan untuk mengevaluasi kemampuan sistem dalam mengubah citra tulisan tangan menjadi teks digital secara keseluruhan. Pada tahap ini, keberhasilan pengenalan tidak hanya bergantung pada klasifikasi karakter menggunakan SVM, tetapi juga pada ketepatan prapemrosesan, segmentasi, urutan karakter, dan pemisahan antarkata.

Berdasarkan pengujian selama tahap pengembangan, ditemukan beberapa kesalahan pengenalan berupa pergantian karakter, penghilangan karakter, dan perbedaan penggunaan huruf kapital serta huruf kecil.

Sebagai contoh, hasil pengenalan pada salah satu gambar yang memiliki teks acuan "Aku membaca buku di kamar" menghasilkan prediksi "AKU meMbaca bum di Kamar". Hasil tersebut menunjukkan bahwa sebagian besar susunan kata dapat dikenali, tetapi masih terdapat kesalahan pada kapitalisasi dan pengenalan karakter dalam kata tertentu. Oleh karena itu, hasilnya merupakan evaluasi pengembangan dan tidak dapat dianggap sebagai hasil pengujian akhir yang sepenuhnya independen.

Analisis Keterbatasan Sistem

Berdasarkan hasil implementasi dan evaluasi pengembangan, aplikasi INKORA memiliki beberapa keterbatasan dalam melakukan pengenalan tulisan tangan. Keterbatasan pertama berkaitan dengan variasi bentuk karakter tulisan tangan. Setiap individu dapat memiliki gaya penulisan yang berbeda, sehingga satu karakter dapat mempunyai bentuk yang bervariasi. Perbedaan tersebut memengaruhi karakteristik fitur HOG yang diperoleh dan hasil klasifikasi SVM.

Keterbatasan kedua terdapat pada proses segmentasi karakter. Huruf yang saling menempel atau memiliki jarak terlalu dekat dapat menyebabkan sistem gagal memisahkan karakter secara tepat. Kesalahan tersebut selanjutnya memengaruhi proses ekstraksi fitur dan klasifikasi.

Keterbatasan ketiga berkaitan dengan perbedaan karakteristik data pelatihan dan citra yang diperoleh melalui kamera perangkat *Android*. Model utama dilatih menggunakan citra karakter dari dataset EMNIST, sedangkan gambar yang diambil melalui kamera dapat memiliki variasi pencahayaan, ketajaman, kemiringan, dan latar belakang.

Keterbatasan berikutnya berkaitan dengan kebutuhan koneksi antara aplikasi dan *backend*. Pada implementasi saat ini, proses pengenalan tulisan tangan dijalankan melalui layanan *Fast*API pada perangkat server. Oleh karena itu, aplikasi memerlukan koneksi jaringan yang memungkinkan perangkat *Android* berkomunikasi dengan *backend*.

Selain itu, algoritma SVM pada penelitian ini melakukan klasifikasi berdasarkan fitur karakter individual dan tidak secara langsung memahami hubungan antarkarakter dalam suatu kata atau kalimat. Hal tersebut menyebabkan sistem masih membutuhkan tahap penyusunan teks, koreksi otomatis, dan penyuntingan manual untuk membantu menghasilkan dokumen digital yang sesuai.

Pembahasan Keseluruhan Sistem

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, aplikasi INKORA berhasil mengintegrasikan antarmuka mobile berbasis *Flutter* dengan *backend* Python menggunakan *FastAPI*. Integrasi tersebut memungkinkan pengguna mengambil citra tulisan tangan melalui kamera atau galeri, melakukan penyesuaian citra, menjalankan proses konversi menggunakan metode ekstraksi fitur *Histogram of Oriented Gradients* (HOG) dan algoritma klasifikasi *Support Vector Machine* (SVM), serta memperoleh hasil berupa teks digital.

Tabel 4 Hasil Analisis Keseluruhan Proses Pengenalan Tulisan Tangan

No.	Aspek Analisis	Fokus Analisis	Hasil dan Pembahasan

1	Prapemrosesan Citra	Menyiapkan citra tulisan tangan sebelum memasuki proses pengenalan.	Tahap prapemrosesan berperan dalam menyiapkan citra agar dapat diproses pada tahap selanjutnya. Kualitas gambar, pencahayaan, kemiringan, dan ketajaman tulisan dapat memengaruhi hasil pengenalan karakter.
2	Segmentasi Karakter	Memisahkan bagian tulisan menjadi karakter atau bagian yang dapat dikenali oleh sistem.	Proses segmentasi masih menjadi salah satu kendala dalam pengenalan tulisan berbentuk kata dan kalimat. Karakter yang saling berdekatan atau bersambung dapat menyebabkan kesalahan pemisahan sehingga memengaruhi urutan karakter yang dihasilkan.
3	Ekstraksi Fitur HOG	Mengambil karakteristik bentuk karakter berdasarkan arah gradien pada citra.	Metode HOG digunakan untuk menghasilkan representasi fitur numerik dari citra karakter. Konfigurasi yang digunakan meliputi 9 orientasi, ukuran sel 4 × 4 piksel, blok 2 × 2 sel, dan normalisasi L2-Hys, sehingga menghasilkan 1.296 fitur.
4	Klasifikasi Karakter SVM	Mengklasifikasikan karakter ke dalam 52 kelas huruf Latin, yaitu A–Z dan a–z.	Algoritma SVM dengan kernel RBF digunakan untuk melakukan klasifikasi karakter. Berdasarkan hasil pengujian dataset EMNIST yang terdiri atas 52.000 data latih dan 5.200 data uji, diperoleh akurasi sebesar 74,50%. Hasil tersebut menunjukkan kemampuan model dalam mengenali karakter berdasarkan data pengujian.

5	Pengenalan Kata dan Kalimat	Menyusun hasil pengenalan karakter menjadi kata dan kalimat dengan mempertahankan urutan serta pemisahan antarkata.	Pengujian menggunakan citra tulisan tangan berbentuk kalimat menunjukkan bahwa hasil konversi masih mengalami kesalahan pada segmentasi, klasifikasi karakter, dan penyusunan teks. Hal ini menunjukkan bahwa keberhasilan klasifikasi karakter belum sepenuhnya menjamin ketepatan hasil konversi kalimat.
6	Koreksi Hasil Pengenalan Teks	Memperbaiki kesalahan karakter atau kata pada hasil konversi.	Tahap koreksi membantu memperbaiki hasil pengenalan yang kurang tepat. Sebagai contoh, teks yang awalnya terbaca "Saya MahaSifwg" dapat diperbaiki menjadi "Saya Mahasiswa". Namun, hasil koreksi tetap perlu diperiksa dan dikonfirmasi oleh pengguna agar sesuai dengan tulisan aslinya.

Berdasarkan Tabel 4, dapat diketahui bahwa setiap tahapan memiliki peran yang saling berkaitan dalam menghasilkan teks digital. Prapemrosesan citra dan segmentasi menjadi tahapan awal yang memengaruhi kualitas data sebelum dilakukan ekstraksi fitur. Selanjutnya, HOG menghasilkan representasi fitur karakter yang digunakan oleh SVM untuk melakukan klasifikasi terhadap 52 kelas huruf Latin.

Hasil pengujian klasifikasi menggunakan dataset EMNIST menunjukkan akurasi sebesar 74,50%. Namun, hasil pengujian menggunakan citra tulisan tangan berbentuk kata dan kalimat masih menunjukkan adanya kesalahan pada proses segmentasi, pengenalan karakter, dan penyusunan teks. Oleh karena itu, nilai akurasi klasifikasi karakter pada dataset EMNIST tidak dapat disamakan dengan tingkat ketepatan konversi tulisan tangan secara keseluruhan.

Dari aspek fungsional, tujuh skenario pengujian utama menunjukkan hasil berhasil. Hal tersebut menunjukkan bahwa aplikasi telah mampu menjalankan fungsi pengambilan gambar,

penyesuaian citra, konversi tulisan tangan, penyuntingan hasil, dan penyimpanan dokumen Word sesuai dengan fungsi yang dirancang.

Dengan demikian, implementasi algoritma SVM pada aplikasi INKORA telah menghasilkan sistem yang dapat digunakan untuk melakukan konversi tulisan tangan menjadi teks digital. Meskipun demikian, peningkatan kualitas prapemrosesan citra, ketepatan segmentasi, kemampuan klasifikasi karakter, serta mekanisme koreksi hasil pengenalan masih menjadi aspek yang perlu dikembangkan agar aplikasi dapat menghasilkan konversi kata dan kalimat yang lebih akurat.

D. KESIMPULAN

Berdasarkan hasil penelitian mengenai implementasi algoritma Support Vector Machine (SVM) pada aplikasi konversi tulisan tangan menjadi dokumen teks berbasis Mobile, dapat disimpulkan bahwa algoritma SVM berhasil diimplementasikan pada aplikasi INKORA berbasis Android menggunakan framework Flutter melalui tahapan pengolahan citra, ekstraksi fitur menggunakan metode Histogram of Oriented Gradients (HOG), dan klasifikasi karakter sehingga menghasilkan teks digital yang dapat diperiksa serta disunting oleh pengguna. Berdasarkan pengujian menggunakan dataset EMNIST ByClass dengan 52 kelas huruf Latin A–Z dan a–z, yang terdiri atas 52.000 data pelatihan dan 5.200 data pengujian, diperoleh tingkat akurasi sebesar 74,50%. Hasil tersebut menunjukkan bahwa kombinasi metode HOG dan algoritma SVM mampu mengenali sebagian besar karakter tulisan tangan pada data pengujian, meskipun masih ditemukan kesalahan klasifikasi pada beberapa huruf yang memiliki kemiripan bentuk.

E. DAFTAR PUSTAKA

AlKendi, W., Gechter, F., Heyberger, L., & Guyeux, C. (2024). Advancements and challenges in handwritten text recognition: A comprehensive survey. Journal of Imaging, 10(1), 18. https://doi.org/10.3390/jimaging10010018

Burges, C. J. C. (1998). A tutorial on Support Vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2), 121–167. https://doi.org/10.1023/A:1009715923555

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297. https://doi.org/10.1007/BF00994018

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. Dalam

2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (Vol. 1, hlm. 886–893). IEEE. https://doi.org/10.1109/CVPR.2005.177

de Souza, C. R. B., & Redmiles, D. F. (2009). On the roles of APIs in the coordination of collaborative software development. Computer Supported Cooperative Work (CSCW), 18(5–6), 445–475.

Dijaya, R., & Setiawan, H. (2023). Buku ajar pengolahan citra digital. Umsida Press. https://doi.org/10.21070/2023/978-623-464-075-5

Ecma International. (2021). ECMA-376: Office Open XML file formats. Diakses 27 Oktober 2026, dari https://ecma-international.org/publications-and-standards/standards/ecma-376/

Elnemr, H. A., Zayed, N. M., & Fakhreldein, M. A. (2016). Feature extraction techniques: Fundamental concepts and survey. Dalam N. K. Kamila (Ed.), Handbook of research on emerging perspectives in intelligent pattern recognition, analysis, and image Processing. IGI Global. https://doi.org/10.4018/978-1-4666-8654-0.ch013

Fitriyah, H., & Wihandika, R. C. (2021). Dasar-dasar pengolahan citra digital. Universitas Brawijaya Press.

Garrido-Munoz, C., Rios-Vila, A., & Calvo-Zaragoza, J. (2026). Handwritten text recognition: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 48(4), 4367–4387. https://doi.org/10.1109/TPAMI.2025.3646002

Gomez, A., & Vesey, A. (2024). On the design, development, and testing of modern APIs. Software Engineering Institute, Carnegie Mellon University. https://www.sei.cmu.edu/library/on-the-design-development-and-testing-of-modern-apis/

Gunawan, G., & Reswan, Y. (2021). Desain aplikasi pengenalan pola tanda tangan menggunakan metode Support Vector Machine (SVM). Jurnal Media Infotama, 17(1). https://doi.org/10.37676/jmi.v17i1.1311

Handhayani, T. (2017). Identifikasi penulis melalui pola tulisan tangan menggunakan algoritma Support Vector Machine. Jurnal Muara Sains, Teknologi, Kedokteran dan Ilmu Kesehatan, 1(1), 210–217. https://doi.org/10.24912/jmstkik.v1i1.432

Hermansyah, E. N., & Wibowo, A. (2025). Dasar membangun aplikasi Mobile dengan Flutter. UNW Press.

Jogiyanto, H. M. (2005). Analisis dan desain sistem informasi: Pendekatan terstruktur teori dan praktik aplikasi bisnis. Andi Offset.

Kadir, A. (2018). Dasar logika pemrograman komputer (Edisi ke-2). Elex Media Komputindo.

Kurniawan, I., Humaira, & Rozi, F. (2020). REST API menggunakan NodeJS pada aplikasi transaksi jasa elektronik berbasis Android. JITSI: Jurnal Ilmiah Teknologi Sistem Informasi, 1(4). https://doi.org/10.62527/jitsi.1.4.18

Kusnadi, Riana, D., & Syahrani, M. (2022). Pengelolaan citra digital dengan menggunakan metode transformasi Grayscale dan pemerataan histogram. JTIK (Jurnal Teknik Informatika Kaputama), 6(1), 108–119.

Microsoft. (2026). Structure of a WordProcessingML document. Diakses 27 Oktober 2026, dari https://learn.microsoft.com/en-us/office/open-xml/word/structure-of-a-wordProcessingml-document

Munir, R. (2004). Pengolahan citra digital dengan pendekatan algoritmik. Informatika.

Munir, R. (2020). Pengantar pengolahan citra digital [Bahan kuliah IF4073 Interpretasi dan Pengolahan Citra]. Institut Teknologi Bandung.

Nalepa, J., & Kawulok, M. (2019). Selecting training sets for Support Vector machines: A review. Artificial Intelligence Review, 52(2), 857–900. https://doi.org/10.1007/s10462-017-9611-1

Naufal, M. F. (2021). Analisis perbandingan algoritma SVM, KNN, dan CNN untuk klasifikasi citra cuaca. Jurnal Teknologi Informasi dan Ilmu Komputer, 8(2), 311–318.

Nockels, J., Gooding, P., Ames, S., & Terras, M. (2022). Understanding the application of handwritten text recognition technology in heritage contexts: A systematic review of Transkribus in published research. Archival Science, 22(3), 367–392. https://doi.org/10.1007/s10502-022-09397-0

Nurhayati, E., & Agussalim. (2023). Rancang bangun back-end API pada aplikasi Mobile AyamHub menggunakan framework Node JS Express. JUSTIN: Jurnal Sistem dan Teknologi Informasi, 11(3). https://doi.org/10.26418/justin.v11i3.66823

Pangaribuan, H., & Simanjuntak, P. (2021). Analisis kualitas perbandingan citra dengan metode segmentasi citra. Jurnal Teknik Informatika UNIKA Santo Thomas, 6(2), 289–297. https://doi.org/10.54367/jtiust.v6i2.1470

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Prasetyo, A. R., Sussi, & Aditya, B. (2023). Analisis perbandingan algoritma Support Vector Machine (SVM) dan Convolutional Neural Network (CNN) untuk sistem deteksi katarak.

Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer, 3(1), 1–10. https://doi.org/10.51903/juritek.v3i1.604

Rachmad, A. (2008). Pengolahan citra digital menggunakan teknik filtering adaptive noise removal pada gambar bernoise. Dalam Prosiding Seminar Nasional Teknoin 2008. Universitas Islam Indonesia. https://journal.uii.ac.id/Teknoin/article/view/2116

Rahmawati, I., & Sari, D. P. (2024). Aplikasi berbasis Android menggunakan Flutter framework untuk keperluan perizinan tugas keluar pada PT. XYZ. JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), 9(2), 979–993. https://doi.org/10.29100/jipi.v9i2.5489

Ratna, S. (2020). Pengolahan citra digital dan histogram dengan Python dan text editor PyCharm. Technologia: Jurnal Ilmiah, 11(3), 181–186.

Reddy, M. (2011). API design for C++. Morgan Kaufmann.

Rosyidah, U., & Rochmawati, N. (2019). Analisis kepribadian melalui tulisan tangan menggunakan metode Support Vector Machine. JINACS (Journal of Informatics and Computer Science), 1(2), 91–96. https://doi.org/10.26740/jinacs.v1n02.p91-96

Safaat H, N. (2012). Pemrograman aplikasi Mobile smartphone dan tablet PC berbasis Android (Edisi revisi). Informatika.

Salcedo-Sanz, S., Rojo-Álvarez, J. L., Martínez-Ramón, M., & Camps-Valls, G. (2014). Support Vector machines in engineering: An overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(3), 234–267. https://doi.org/10.1002/widm.1125

Salsabilla, D., & Wahyusari, R. (2025). Perbandingan metode reduksi noise pada citra digital berbasis website. Jurnal Komtika (Komputasi dan Informatika), 9(2), 195–204. https://doi.org/10.31603/komtika.v9i2.14966

Santri, A. L. (2025). Implementasi algoritma deep learning pada aplikasi konversi tulis tangan ke dokumen teks [Skripsi]. Fakultas Teknik, Universitas Islam Sumatera Utara.

Sitorus, L. (2015). Algoritma dan pemrograman. Andi Offset.

Sudradjat, B. (2021). Penggunaan teknologi Flutter dalam aplikasi Mobile untuk pengembangan kedai kopi. REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer, 6(1), 1–8. https://doi.org/10.33395/remik.v6i1.11123

Sugianela, Y., & Suciati, N. (2019). Ekstraksi fitur pada pengenalan karakter aksara Jawa berbasis histogram of oriented gradient. JUTI: Jurnal Ilmiah Teknologi Informasi, 17(1), 64–72. https://doi.org/10.12962/j24068535.v17i1.a819

Sulistya, Y. I., Istighosah, M., Azis, H., Septiadi, A. D., Wardhana, A. C., Adhitama, R., Amrulloh, A., & Septiara, M. (2025). Pengantar pemrograman Dart: Panduan cepat untuk pemula.

Penerbit Widina.

Surinta, O., Karaaba, M. F., Schomaker, L. R. B., & Wiering, M. A. (2015). Recognition of handwritten characters using local gradient feature descriptors. Engineering Applications of Artificial Intelligence, 45, 405–414.

Suwanto, G., Adam, R. I., & Garno. (2021). Identifikasi citra digital jenis beras menggunakan metode ANFIS dan Sobel. JIP (Jurnal Informatika Polinema), 7(2), 123–128.

Syuhada, F. (2023). Klasifikasi citra tulisan tangan aksara Sasak dengan metode histogram of oriented gradients dan multinomial logistic regression. Jurnal Informatika dan Teknik Elektro Terapan, 11(3). https://doi.org/10.23960/jitet.v11i3.3247

Tolle, H., Pinandito, A., Kharisma, A. P., & Dewi, R. K. (2017). Pengembangan aplikasi perangkat bergerak (konsep dan implementasi). UB Press.

Turban, E., King, D., Lee, J., Liang, T.-P., & Turban, D. C. (2012). Electronic commerce 2012: A managerial and social networks perspective (7th ed.). Pearson.

van Rossum, G. (1995). Python tutorial (Technical Report CS-R9526). Centrum voor Wiskunde en Informatica (CWI).

Wijaya, A. S., Chamidah, N., & Santoni, M. M. (2019). Pengenalan karakter tulisan tangan dengan K-Support Vector nearest neighbor. IJEIS (Indonesian Journal of Electronics and Instrumentation Systems), 9(1), 33–44. https://doi.org/10.22146/ijeis.38729

Wijaya, B. A., Putra, J., Dharshinni, N. P., Girsang, B. S. P., & Fawwaz, I. (2023). Pemrograman Mobile dengan Flutter. UNPRI Press.