



Analisis Perbandingan Performa Klasifikasi KNN dan SVM pada Dataset Sign Language MNIST: Citra Grayscale vs Binarized (Otsu Thresholding)

Adi Pratama Putra¹, Annisa Nur Aulia Nadiva², Dunik Andriyani³, Sulthan Fauzan Fackhri⁴

^{1,2,3,4}Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang

2310631170062@student.unsika.ac.id¹, 2310631170068@student.unsika.ac.id², 2310631170134@student.unsika.ac.id³, 2310631170050@student.unsika.ac.id⁴

Abstrak

Pengenalan bahasa isyarat merupakan bidang penelitian penting untuk mendukung aksesibilitas komunikasi bagi individu penyandang gangguan pendengaran dan wicara. Penelitian ini menyajikan analisis perbandingan performa klasifikasi K-Nearest Neighbors (KNN) dan Support Vector Machine (SVM) pada dataset Sign Language MNIST dengan dua skema representasi citra: Grayscale asli dan citra Biner hasil Otsu Thresholding. Dataset terdiri dari 27.455 data latih dan 7.172 data uji, berupa citra gestur tangan American Sign Language (ASL) berukuran 28x28 piksel untuk 24 kelas huruf. Otsu Thresholding diterapkan menggunakan fungsi `cv2.threshold()` dengan flag `THRESH_BINARY + THRESH_OTSU` pada setiap citra secara individual. Normalisasi dilakukan dengan membagi nilai piksel dengan 255,0 sehingga berada dalam rentang [0,1]. Hasil eksperimen menunjukkan: KNN mencapai akurasi 80,59% pada Grayscale dan 79,87% pada Otsu dengan waktu 13,50 detik dan 13,02 detik; SVM mencapai akurasi 84,19% pada Grayscale dan 82,91% pada Otsu dengan waktu 158,10 detik dan 140,57 detik. Representasi Grayscale secara konsisten menghasilkan akurasi lebih tinggi pada kedua algoritma. SVM Grayscale menjadi konfigurasi terbaik dengan akurasi 84,19%. Classification report SVM Otsu menunjukkan weighted F1-Score 0,83 dengan variasi per kelas yang signifikan: kelas D (F1=1,00) terbaik, kelas R (F1=0,44) terburuk.

Kata kunci: Pengenalan Bahasa Isyarat, KNN, SVM, Otsu Thresholding, Klasifikasi Citra.

1. Pendahuluan

Bahasa isyarat merupakan sistem komunikasi visual yang menjadi sarana utama bagi individu penyandang gangguan pendengaran dan wicara. Di Indonesia, Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (Bisindo) adalah dua sistem yang paling banyak digunakan. Kesenjangan komunikasi antara pengguna bahasa isyarat dan masyarakat umum masih menjadi tantangan, sehingga pengembangan sistem pengenalan bahasa isyarat berbasis kecerdasan buatan menjadi kebutuhan yang semakin mendesak.

Dataset Sign Language MNIST merupakan dataset publik yang diadaptasi dari MNIST digit klasik untuk pengenalan gestur tangan statis ASL (Davenport, 2017). Dataset ini memiliki format citra 28x28 piksel Grayscale dengan 27.455 sampel latih dan 7.172 sampel uji yang merepresentasikan 24 kelas huruf (A-Y, tidak termasuk J dan Z yang memerlukan gerakan dinamis). Setiap sampel terdiri dari satu kolom label dan 784 kolom nilai piksel dalam format CSV.

Representasi fitur citra merupakan faktor penentu kualitas model klasifikasi. Citra Grayscale mempertahankan informasi gradien intensitas dengan 256 level (0-255), sementara Otsu Thresholding menghasilkan representasi biner dengan hanya dua nilai piksel melalui penentuan ambang batas T^* secara adaptif berdasarkan histogram citra (Otsu, 1979). Metode Otsu memaksimalkan variansi antar kelas sesuai persamaan: $\sigma_B^2(t) = \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2$. Pertanyaan utama penelitian ini adalah: apakah binarisasi Otsu meningkatkan atau menurunkan performa KNN dan SVM pada Sign Language MNIST?

K-Nearest Neighbors (KNN) adalah algoritma lazy learning yang mengklasifikasikan sampel berdasarkan mayoritas label dari K tetangga terdekat menggunakan jarak Euclidean (Cover & Hart, 1967). Dalam penelitian ini KNN dikonfigurasi dengan `n_neighbors=5`. Support Vector Machine (SVM) mencari hyperplane optimal dengan margin terbesar menggunakan kernel RBF: $K(x, x') = \exp(-\gamma \|x - x'\|^2)$ (Cortes & Vapnik, 1995), dikonfigurasi dengan `kernel='rbf'` dan `random_state=42`.

Dataset Sign Language MNIST memiliki kompleksitas intra-kelas yang tinggi, terutama pada kelompok huruf yang secara visual serupa seperti M, N, S, R, dan T. Penelitian sebelumnya menggunakan berbagai pendekatan fitur untuk dataset ini, namun belum ada kajian sistematis yang membandingkan dampak binarisasi Otsu terhadap performa KNN dan SVM secara bersamaan. Kebaruan penelitian ini terletak pada evaluasi komprehensif yang mencakup akurasi, presisi, recall, F1-Score per kelas, dan waktu komputasi untuk keempat konfigurasi tersebut menggunakan kode yang dapat direproduksi sepenuhnya.

Pengenalan gestur tangan statis menghadirkan tantangan komputasi visual yang jauh lebih kompleks dibandingkan pengenalan karakter optik (OCR) atau digit tulisan tangan. Tantangan utama dalam klasifikasi American Sign Language (ASL) adalah tingginya variasi intra-kelas yang disebabkan oleh perbedaan ukuran tangan, sudut pencahayaan, dan tingkat rotasi, serta tingginya kemiripan antar-kelas (inter-class similarity). Beberapa gestur alfabet, seperti huruf 'M', 'N', 'S', dan 'T', seluruhnya direpresentasikan oleh bentuk tangan mengepal (closed fist), di mana pembedanya hanyalah posisi ibu jari yang terselip di antara jari-jari lainnya. Secara teoritis, metode binarisasi seperti Otsu Thresholding sering kali diajukan sebagai solusi untuk mengurangi dimensi ruang pencarian fitur dan menghilangkan gangguan latar belakang (background noise). Otsu Thresholding berasumsi bahwa citra memiliki histogram bimodal (piksel latar depan dan latar belakang yang terdistribusi secara terpisah). Namun, pada citra tangan yang memiliki latar belakang homogen, binarisasi berisiko menyebabkan kompresi lossy yang ekstrem, menghilangkan kedalaman spasial (spatial depth) dan batas bayangan (shadow boundaries) yang justru memuat fitur diskriminatif paling krusial. Oleh karena itu, pengujian empiris terhadap sejauh mana kompresi informasi ini berdampak pada model parametrik (SVM) maupun non-parametrik (KNN) menjadi sangat esensial untuk dieksplorasi.

Sistem Pendukung Keputusan (SPK) merupakan sistem berbasis komputer yang dirancang untuk membantu proses pengambilan keputusan melalui pengolahan data, informasi, dan model tertentu sehingga menghasilkan rekomendasi yang lebih objektif, efektif, dan akurat. Perkembangan teknologi informasi telah mendorong pemanfaatan SPK dalam berbagai bidang karena kemampuannya dalam mengintegrasikan data, metode analisis, dan pengetahuan untuk mendukung penyelesaian masalah yang kompleks. Keberhasilan implementasi SPK tidak terlepas dari dukungan sistem informasi yang terstruktur, pengelolaan data yang baik, serta pengembangan perangkat lunak yang sesuai dengan kebutuhan pengguna. Selain itu, penerapan metode pengambilan keputusan multikriteria memungkinkan proses evaluasi berbagai alternatif berdasarkan sejumlah kriteria yang telah ditentukan sehingga menghasilkan keputusan yang lebih rasional dan dapat dipertanggungjawabkan. Dalam era transformasi digital, SPK menjadi salah satu instrumen penting bagi organisasi untuk meningkatkan efektivitas kerja, efisiensi operasional, serta kualitas pengambilan keputusan dalam berbagai sektor, termasuk pendidikan, bisnis, dan manajemen organisasi (Arifin, 2021; Fatta, 2017; Jogiyanto, 2018; McLeod & Schell, 2016; Marimin, 2017; Turban et al., 2018; Yakub, 2019; Romney & Steinbart, 2018; Laudon & Laudon, 2020; Pressman & Maxim, 2020).

Tujuan penelitian: (1) mengimplementasikan Otsu Thresholding pada dataset Sign Language MNIST menggunakan OpenCV; (2) melatih dan mengevaluasi KNN ($n_neighbors=5$) pada representasi Grayscale dan Binary Otsu; (3) melatih dan mengevaluasi SVM (kernel RBF, $random_state=42$) pada kedua representasi; serta (4) membandingkan akurasi, waktu komputasi, dan F1-Score per kelas dari keempat konfigurasi secara komprehensif.

2. Metode Penelitian

Penelitian ini menggunakan pendekatan eksperimental komparatif. Seluruh implementasi menggunakan Python 3 dengan library pandas, numpy, cv2 (OpenCV), scikit-learn, matplotlib, seaborn, dan modul time pada platform Google Colaboratory dengan sesi runtime CPU standar.

2.1. Dataset Sign Language MNIST

Dataset Sign Language MNIST dimuat dari dua file CSV menggunakan `pandas.read_csv()`. File `sign_mnist_train.csv` berisi 27.455 baris dan `sign_mnist_test.csv` berisi 7.172 baris, masing-masing terdiri dari 785 kolom: satu kolom 'label' (nilai 0-24, tanpa kelas 9 untuk huruf J) dan 784 kolom piksel. Tahap awal prapemrosesan melibatkan pemisahan array data untuk memisahkan atribut target (label kelas) dari matriks fitur (piksel citra) pada masing-masing data latih dan data uji.

2.2. Praproses: Otsu Thresholding dan Normalisasi

Transformasi Otsu dilakukan melalui fungsi `apply_otsu_thresholding()` yang menginisialisasi array `binarized_dataset = np.zeros_like(dataset_pixels)` kemudian mengiterasi setiap sampel. Untuk setiap sampel i , array 1D berukuran 784 di-reshape: `img_2d = dataset_pixels[i].reshape(28, 28).astype(np.uint8)`. Selanjutnya `cv2.threshold(img_2d, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)` dipanggil, menghasilkan `ret` (nilai $T^*_{optimal}$) dan `thresh_img` (citra biner). Citra biner di-flatten dengan `thresh_img.flatten()` dan disimpan ke `binarized_dataset[i]`. Fungsi ini dijalankan terpisah untuk X_{train} (27.455 sampel) dan X_{test} (7.172 sampel).

Normalisasi dilakukan dengan membagi seluruh nilai piksel dengan 255,0: $X_{train_gray_norm} = X_{train} / 255,0$, $X_{test_gray_norm} = X_{test} / 255,0$, $X_{train_otsu_norm} = X_{train_otsu} / 255,0$, $X_{test_otsu_norm} = X_{test_otsu} / 255,0$. Hasilnya empat variabel fitur ternormalisasi dalam rentang $[0,1]$. Langkah normalisasi dilakukan setelah transformasi Otsu sehingga data Grayscale dan Binary berada pada skala yang sama dan perbandingan berlangsung secara adil.

2.3. Konfigurasi Model KNN dan SVM

Algoritma K-Nearest Neighbors (KNN) merupakan metode non-parametrik dan lazy learning yang melakukan klasifikasi berdasarkan kedekatan jarak pada ruang dimensi fitur. Pada penelitian ini, jarak antara data uji x dan data latih x_i dihitung menggunakan metrik jarak Euclidean dalam ruang berdimensi 784, yang secara matematis dirumuskan sebagai

$$d(x, x_i) = \sqrt{\sum_{j=1}^{784} (x^{(j)} - x_i^{(j)})^2}$$

Di sisi lain, Support Vector Machine (SVM) merupakan model parametrik yang secara proaktif mencari hyperplane pemisah optimal (optimal separating hyperplane) dengan memaksimalkan margin antara kelas-kelas yang berbeda. Karena data citra Sign Language MNIST tidak dapat dipisahkan secara linear pada dimensi aslinya, digunakan fungsi kernel Radial Basis Function (RBF) untuk memetakan data ke ruang fitur berdimensi lebih tinggi tak hingga. Fungsi kernel RBF didefinisikan sebagai:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2)$$

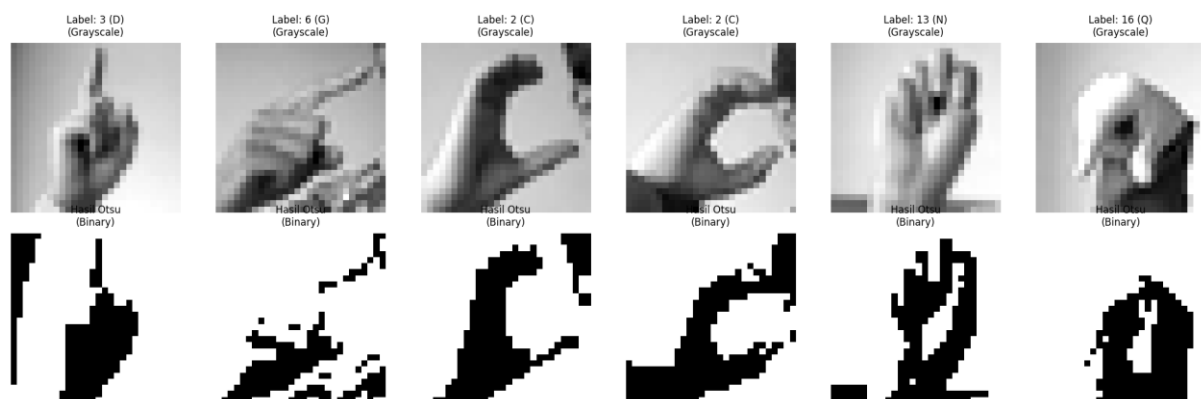
Nilai parameter γ (gamma) mengatur seberapa jauh pengaruh satu sampel pelatihan tunggal. Untuk klasifikasi 24 kelas, model diimplementasikan menggunakan pendekatan One-vs-One (OvO), yang secara internal membangun $\frac{N(N-1)}{2}$ atau setara dengan 276 pengklasifikasi biner independen. Keputusan akhir kelas diprediksi melalui mekanisme pemungutan suara terbanyak (majority voting). Dalam tahap implementasi, model KNN diinisialisasi dengan parameter tetangga terdekat sebanyak 5 ($K=5$) menggunakan jarak Euclidean default. Model SVM diinisialisasi menggunakan kernel RBF dengan `random_state=42` untuk memastikan hasil dapat direproduksi. Pengukuran waktu komputasi dilakukan secara komprehensif dengan menghitung total waktu eksekusi yang dibutuhkan oleh model sejak fase pelatihan (fitting) dimulai hingga fase pengujian (predicting) selesai dilakukan. Evaluasi performa model diukur menggunakan skor akurasi untuk keseluruhan sistem dan classification report untuk mendapatkan detail metrik per kelas. Seluruh eksperimen dieksekusi pada Google Colaboratory dengan sesi runtime CPU standar menggunakan pustaka (library) Python yang mencakup pandas, numpy, cv2 (OpenCV), scikit-learn, matplotlib, seaborn, dan time (Pedregosa et al., 2011).

3. Hasil dan Pembahasan

3.1. Visualisasi Transformasi Citra

Fungsi `visualize_comparison()` menampilkan 6 sampel pertama dataset latih dalam dua baris: baris atas citra Grayscale asli dan baris bawah hasil Binary Otsu. Label 6 sampel pertama adalah: 3 (D), 6 (G), 2 (C), 2 (C), 13 (N), dan 16 (Q), sesuai dengan nilai `y_train[0:6]` dari dataset. Gambar 1 menunjukkan hasil visualisasi tersebut. Citra Grayscale mempertahankan gradien intensitas halus pada kontur jari, tekstur telapak tangan, dan bayangan, memberikan 256 level informasi per piksel. Citra Binary Otsu menghasilkan siluet dua nilai (0 hitam atau 255 putih) yang drastis menyederhanakan representasi. Kelas 16 (Q) yang memiliki bentuk gestur sangat khas tetap terdistingksi dengan baik pada kedua representasi, konsisten dengan F1-Score tertinggi (0,98) pada classification report SVM Otsu. Sebaliknya, informasi pembeda halus untuk kelas yang serupa seperti N (13) dan C (2) berkurang signifikan setelah binarisasi.

Gambar 1. Visualisasi Perbandingan Citra Grayscale vs Binary Otsu (6 Sampel Pertama Dataset Latih)



Gambar 1. Visualisasi Perbandingan Citra Grayscale (Baris Atas) dan Binary Otsu (Baris Bawah) untuk 6 Sampel Pertama Dataset Latih: Label 3(D), 6(G), 2(C), 2(C), 13(N), 16(Q)

3.2. Hasil Akurasi Klasifikasi

DOI: <https://doi.org/10.69693/ijmst.v4i2.9933>

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Tabel 1 merangkum hasil akurasi keempat konfigurasi pada 7.172 sampel data uji. SVM Grayscale menghasilkan akurasi tertinggi 84,19%, diikuti SVM Otsu 82,91%, KNN Grayscale 80,59%, dan KNN Otsu 79,87%. Representasi Grayscale secara konsisten menghasilkan akurasi lebih tinggi pada kedua algoritma.

Penurunan akurasi akibat binarisasi Otsu sebesar 0,72 poin persentase pada KNN (80,59% → 79,87%) dan 1,28 poin persentase pada SVM (84,19% → 82,91%). Pada 7.172 sampel uji, penurunan 0,72% setara 52 sampel salah klasifikasi tambahan untuk KNN, dan penurunan 1,28% setara 92 sampel untuk SVM. SVM mengungguli KNN dengan selisih 3,60% pada Grayscale dan 3,04% pada Otsu.

Grafik perbandingan akurasi (Gambar 2, kiri) menunjukkan batang biru (#3498db) untuk Grayscale dan merah (#e74c3c) untuk Otsu Binary pada dua kelompok KNN dan SVM. Perbedaan tinggi batang antara Grayscale dan Otsu lebih jelas terlihat pada kelompok SVM (1,28pp) dibandingkan KNN (0,72pp), mengindikasikan SVM lebih sensitif terhadap kualitas representasi fitur dalam proses pembentukan hyperplane optimal.

Tabel 1. Tabel Perbandingan Akurasi Model Klasifikasi

Model	Representasi Citra	Akurasi (%)
KNN (k=5)	Grayscale	80,59
KNN (k=5)	Binary Otsu	79,87
SVM (RBF)	Grayscale	84,19
SVM (RBF)	Binary Otsu	82,91

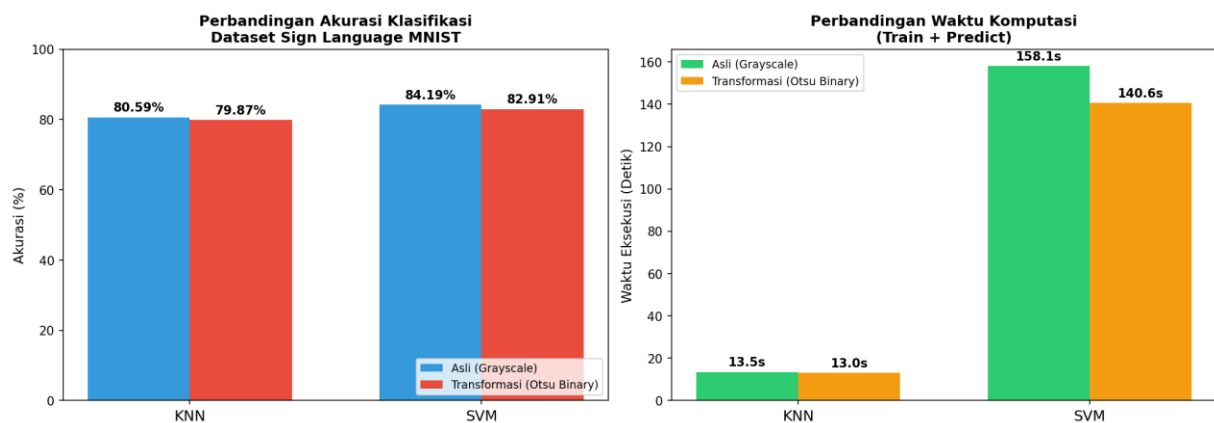
3.3. Perbandingan Waktu Komputasi

Tabel 2 menyajikan waktu komputasi (pelatihan + prediksi) dari keempat konfigurasi. KNN Grayscale: 13,50 detik; KNN Otsu: 13,02 detik; SVM Grayscale: 158,10 detik; SVM Otsu: 140,57 detik. KNN sekitar 10-12 kali lebih cepat dari SVM karena bersifat lazy learner yang tidak memerlukan fase optimasi eksplisit saat pelatihan.

Grafik waktu komputasi (Gambar 2, kanan) menggunakan warna hijau (#2ecc71) untuk Grayscale dan oranye (#f39c12) untuk Otsu. SVM Otsu (140,57 detik) lebih cepat dari SVM Grayscale (158,10 detik), selisih 17,53 detik (11,1%). Hal ini karena data biner memiliki variasi nilai lebih rendah sehingga optimasi quadratic programming pada kernel RBF konvergen lebih cepat. Sebaliknya KNN Otsu (13,02 detik) lebih cepat dari KNN Grayscale (13,50 detik), selisih 0,48 detik (3,6%). Binarisasi Otsu tidak menawarkan keuntungan komputasi yang cukup signifikan untuk mengimbangi penurunan akurasi yang terjadi.

Tabel 2. Tabel Perbandingan Waktu Komputasi (Detik)

Model	Representasi Citra	Waktu (detik)
KNN (k=5)	Grayscale	13,50
KNN (k=5)	Binary Otsu	13,02
SVM (RBF)	Grayscale	158,10
SVM (RBF)	Binary Otsu	140,57



Gambar 2. Grafik Perbandingan Akurasi (Kiri: Biru=Grayscale, Merah=Otsu) dan Waktu Komputasi (Kanan: Hijau=Grayscale, Oranye=Otsu)

3.4. Analisis Classification Report

Tabel 3 menyajikan classification report lengkap model SVM pada representasi citra biner Otsu dengan tingkat akurasi keseluruhan sebesar 82,91%. Berdasarkan metrik evaluasi tersebut, ditemukan hasil bahwa Kelas 3 (huruf D) berhasil mencapai F1-Score sempurna senilai 1,00, dengan tingkat presisi dan recall yang masing-masing menyentuh angka 1,00 dari total support sebanyak 245 sampel. Selain itu, Kelas 15 (huruf P) dan Kelas 16 (huruf Q) juga menunjukkan performa yang sangat impresif dengan raihan F1-Score masing-masing sebesar 0,97 dan 0,98. Kelompok kelas ini diketahui memiliki bentuk geometri gestur tangan yang sangat khas secara struktural, sehingga polanya tetap mudah diidentifikasi dan dibedakan oleh model pembentuk hyperplane SVM meskipun informasinya telah direduksi ke dalam bentuk representasi biner.

Sebaliknya, degradasi performa yang cukup masif terdeteksi pada beberapa kelompok gestur spesifik. Kelas 17 (huruf R) menunjukkan performa klasifikasi terburuk dalam eksperimen ini dengan perolehan F1-Score hanya sebesar 0,44, di mana nilai presisinya anjlok ke angka 0,31 meskipun nilai recall-nya berada di tingkat 0,72 dari 144 sampel support. Rendahnya nilai presisi pada angka 0,31 ini mengindikasikan bahwa hanya terdapat 31% dari total prediksi label R oleh model yang benar-benar valid, sementara sisanya merupakan bias di mana kelas gestur lain keliru diklasifikasikan sebagai huruf R. Di sisi lain, Kelas 12 (huruf M) menunjukkan anomali pola yang berkebalikan, yaitu memiliki presisi yang sangat tinggi mencapai 0,99 namun dengan tingkat recall yang rendah sebesar 0,55 dengan nilai F1-Score sebesar 0,70. Pola ini menandakan bahwa banyak sampel asli dari huruf M yang gagal diidentifikasi oleh sistem (false negative yang tinggi) akibat tingginya kemiripan visual yang membingungkan dengan huruf N dan S ketika dikonversi menjadi citra biner. Sementara itu, Kelas 11 (huruf L) mencatat tingkat recall sempurna senilai 1,00, tetapi memiliki presisi yang relatif rendah pada angka 0,74 dengan skor akhir F1-Score sebesar 0,85.

Analisis lebih mendalam terhadap tingkat kesalahan klasifikasi (misclassification rate) menunjukkan bahwa penurunan performa pada representasi biner (Otsu) memiliki pola yang spesifik dan terpusat pada kelompok gestur tertentu. Huruf 'R' (kelas 17), yang mencatat F1-Score terendah (0,44), melibatkan gestur jari telunjuk dan jari tengah yang saling menyilang (crossed fingers). Pada representasi Grayscale, persilangan jari ini menghasilkan perbedaan gradien intensitas cahaya atau efek bayangan yang memungkinkan SVM membedakan batas antar-jari dengan jelas. Namun, ketika diterapkan binarisasi Otsu, piksel di area persilangan jari tersebut melebur menjadi satu blok piksel putih yang homogen (bernilai 255), sehingga menghilangkan topologi struktur jari secara keseluruhan. Hal yang sama terjadi pada kelompok huruf 'M', 'N', 'S', dan 'T' yang didominasi oleh bentuk tangan mengepal. Kemiripan topologi area kepalan tangan pada citra biner menyebabkan algoritma kesulitan mendeteksi posisi spesifik ibu jari. Fenomena ini membuktikan bahwa batas bayangan dan pencahayaan (illumination shading) pada citra gestur tangan bukanlah sebuah gangguan (noise), melainkan fitur spasial yang mendefinisikan geometri 3D dari tangan ketika diproyeksikan ke dalam citra 2D.

Secara keseluruhan, metrik rata-rata tertimbang (weighted average) dari pengujian ini menunjukkan nilai presisi sebesar 0,86, recall sebesar 0,83, dan F1-Score sebesar 0,83 dari total keseluruhan 7.172 sampel uji. Sementara itu, metrik rata-rata makro (macro average) mencatat nilai presisi 0,83, recall 0,83, dan F1-Score sebesar 0,82. Munculnya selisih antara nilai weighted average (0,83) dan macro average (0,82) pada aspek F1-Score ini menegaskan adanya variasi performa yang cukup signifikan antar-kelas di dalam sistem. Kondisi ini terjadi karena kelas-kelas dengan jumlah sampel (support) besar dan skor F1 tinggi mendominasi dan mendorong nilai rata-rata tertimbang menjadi lebih tinggi dibandingkan dengan perhitungan rata-rata makro yang memperlakukan semua kelas secara setara tanpa memedulikan distribusi jumlah datanya.

Tabel 3. Tabel Classification Report SVM Otsu per Kelas (Akurasi 82,91%)

Kelas	Presisi	Recall	F1-Score	Support
0 (A)	0,94	1,00	0,97	331
1 (B)	1,00	0,91	0,95	432
2 (C)	1,00	0,94	0,97	310
3 (D)	1,00	1,00	1,00	245
4 (E)	0,94	0,96	0,95	498
5 (F)	0,84	0,92	0,88	247
6 (G)	0,88	0,88	0,88	348
7 (H)	1,00	0,86	0,92	436
8 (I)	0,92	0,92	0,92	288
10 (K)	0,73	0,69	0,71	331
11 (L)	0,74	1,00	0,85	209
12 (M)	0,99	0,55	0,70	394
13 (N)	0,82	0,71	0,76	291
14 (O)	0,91	0,91	0,91	246
15 (P)	0,95	1,00	0,97	347
16 (Q)	0,97	1,00	0,98	164
17 (R)	0,31	0,72	0,44	144
18 (S)	0,48	0,82	0,60	246
19 (T)	0,68	0,61	0,65	248
20 (U)	0,70	0,68	0,69	266
21 (V)	0,91	0,57	0,70	346
22 (W)	0,65	0,80	0,72	206
23 (X)	0,72	0,76	0,74	267
24 (Y)	0,88	0,73	0,80	332
Macro avg	0,83	0,83	0,82	7.172
Weighted avg	0,86	0,83	0,83	7.172

3.5. Diskusi

Penurunan akurasi yang konsisten pada representasi Binary Otsu dibandingkan Grayscale untuk kedua algoritma mengkonfirmasi bahwa informasi gradien intensitas pada citra Grayscale memiliki nilai diskriminatif penting. Berbeda dengan citra dokumen atau karakter cetak berlatar belakang kompleks di mana binarisasi umumnya

meningkatkan performa, pada citra tangan Sign Language MNIST dengan latar belakang relatif bersih, penghilangan gradien justru merugikan. Binarisasi Otsu secara efektif melakukan kompresi lossy dari 256 level intensitas menjadi 2 level, menghilangkan informasi yang tidak dapat dipulihkan.

Keunggulan SVM atas KNN pada kedua representasi sesuai karakteristik teoretis SVM yang dirancang untuk ruang fitur berdimensi tinggi (784 dimensi) melalui kernel trick. KNN mengalami degradasi pada dimensi tinggi karena semua jarak cenderung konvergen (curse of dimensionality) (Cover & Hart, 1967; Cortes & Vapnik, 1995). SVM Otsu (140,57 detik) lebih cepat dari SVM Grayscale (158,10 detik) namun menghasilkan akurasi 1,28% lebih rendah, sehingga trade-off kecepatan tidak mengimbangi kerugian akurasi. Temuan ini menyatakan Otsu efektif hanya pada citra dengan kontras tinggi dan latar belakang tidak seragam.

Keterbatasan penelitian: (1) tidak dilakukan hyperparameter tuning (grid search untuk C dan gamma pada SVM, nilai K optimal pada KNN); (2) tidak digunakan metode ekstraksi fitur seperti HOG atau LBP yang dapat memberikan representasi lebih informatif (3) classification report hanya dihasilkan untuk SVM Otsu dalam kode, sehingga perbandingan F1-Score per kelas antar representasi tidak dapat dilakukan secara langsung. Untuk penelitian lanjutan, disarankan eksplorasi hyperparameter tuning, ekstraksi fitur berbasis shape descriptor, dan metode binarisasi adaptif seperti Sauvola Thresholding sebagai alternatif Otsu

3.6. Analisis Kompleksitas Algoritma dan Implikasi Deployment

Temuan dari sisi waktu komputasi memberikan pandangan yang menarik terkait skalabilitas dan kelayakan implementasi algoritma pada perangkat dengan sumber daya terbatas (edge devices atau smartphone). Meskipun KNN menunjukkan waktu eksekusi gabungan yang jauh lebih singkat (sekitar 13 detik) dibandingkan SVM (sekitar 140-158 detik), perlu dipahami perbedaan fundamental dari kompleksitas ruang dan waktu kedua algoritma ini.

KNN memiliki kompleksitas pelatihan $\mathcal{O}(1)$ karena tidak membangun struktur model internal, melainkan hanya menyimpan data latih di dalam memori. Namun, kompleksitas prediksi KNN adalah $\mathcal{O}(N \cdot d)$ untuk setiap citra baru, di mana N adalah jumlah data latih (27.455 sampel) dan d adalah dimensi fitur (784 piksel). Hal ini membuat KNN sangat tidak efisien untuk aplikasi penerjemahan bahasa isyarat real-time (video langsung), karena perangkat keras harus melakukan komputasi perhitungan jarak yang masif pada setiap frame citra yang masuk dan pada saat yang bersamaan menghabiskan alokasi RAM yang besar untuk menampung seluruh data latih.

Sebaliknya, SVM membutuhkan waktu yang secara signifikan lebih lama pada fase pelatihan karena algoritma harus memecahkan optimasi pemrograman kuadratik (quadratic programming optimization) dengan kompleksitas waktu pelatihan berkisar antara $\mathcal{O}(N^2)$ hingga $\mathcal{O}(N^3)$. Penurunan waktu eksekusi sebesar 11,1% pada model SVM Otsu dibandingkan SVM Grayscale secara empiris membuktikan bahwa vektor yang berisikan nilai diskrit biner (0 dan 1) mampu mempercepat proses konvergensi fungsi kernel dalam menemukan vektor pendukung (support vectors). Namun, setelah model SVM berhasil dilatih, kompleksitas prediksinya menjadi sangat efisien, yaitu $\mathcal{O}(N_{sv} \cdot d)$, di mana N_{sv} adalah jumlah support vectors yang jumlahnya direduksi secara signifikan dan jauh lebih sedikit daripada total keseluruhan sampel data latih N.

Oleh karena itu, untuk kebutuhan sistem pengenalan bahasa isyarat di dunia nyata (production deployment), SVM dengan representasi citra Grayscale merupakan pilihan arsitektur yang jauh lebih rasional dibandingkan KNN. Beban komputasi yang tinggi pada model SVM hanya terjadi satu kali (yakni saat fase training di server komputasi), sedangkan pada fase inferencing (penggunaan aplikasi akhir oleh pengguna), SVM mampu memberikan hasil prediksi seketika (low latency) dengan tingkat akurasi yang lebih superior (84,19%) dan konsumsi memori yang jauh lebih ringan.

4. Kesimpulan

Penelitian ini membandingkan performa KNN ($n_neighbors=5$) dan SVM (kernel RBF, $random_state=42$) pada dataset Sign Language MNIST dengan representasi Grayscale dan Binary Otsu Thresholding. Berdasarkan eksperimen pada 27.455 data latih dan 7.172 data uji, diperoleh: (1) SVM Grayscale menjadi konfigurasi terbaik dengan akurasi 84,19%; (2) KNN Grayscale mencapai 80,59%, KNN Otsu 79,87%, dan SVM Otsu 82,91%; (3) binarisasi Otsu menurunkan akurasi 0,72% pada KNN dan 1,28% pada SVM; (4) SVM Otsu (140,57 detik) lebih cepat dari SVM Grayscale (158,10 detik), sedangkan KNN Otsu (13,02 detik) lebih cepat dari KNN Grayscale (13,50 detik).

Binarisasi Otsu tidak terbukti menguntungkan sebagai strategi praproses untuk Sign Language MNIST karena menghilangkan informasi gradien yang krusial untuk membedakan gestur serupa seperti R, S, T, M, dan N, sebagaimana tercermin dari F1-Score terendah pada kelas R (0,44) dalam classification report SVM Otsu. Representasi Grayscale ternormalisasi ($/255.0$) dengan SVM kernel RBF merupakan pilihan yang lebih optimal. Penelitian lanjutan disarankan mengeksplorasi hyperparameter tuning menggunakan grid search, metode ekstraksi fitur (HOG, LBP), dan model deep learning berbasis CNN.

Ucapan Terimakasih

Penulis mengucapkan terima kasih kepada seluruh pihak yang telah memberikan dukungan dalam pelaksanaan penelitian ini. Dataset Sign Language MNIST yang digunakan tersedia secara publik melalui platform Kaggle (Davenport, 2017). Kode eksperimen dijalankan pada platform Google Colaboratory.

Reference

- Arifin, Z. (2021). *Sistem pendukung keputusan berbasis komputer*. Deepublish.
- Cortes, C., & Vapnik, V. (1995). *Support-vector networks*. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- Cover, T., & Hart, P. (1967). *Nearest neighbor pattern classification*. *IEEE Transactions on Information Theory*, 13(1), 21–27. <https://doi.org/10.1109/TIT.1967.1053964>
- Davenport, T. (2017). *Sign Language MNIST*. Kaggle Dataset. <https://www.kaggle.com/datamunge/sign-language-mnist>
- Fatta, H. A. (2017). *Analisis dan perancangan sistem informasi*. Andi.
- Jogiyanto, H. M. (2018). *Analisis dan desain sistem informasi*. Andi.
- Laudon, K. C., & Laudon, J. P. (2020). *Management information systems: Managing the digital firm* (16th ed.). Pearson.
- Marimin. (2017). *Teknik dan aplikasi pengambilan keputusan kriteria majemuk*. Grasindo.
- McLeod, R., & Schell, G. P. (2016). *Management information systems* (12th ed.). Pearson.
- Otsu, N. (1979). *A threshold selection method from gray-level histograms*. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66. <https://doi.org/10.1109/TSMC.1979.4310076>
- 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, E. (2011). *Scikit-learn: Machine learning in Python*. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.
- Romney, M. B., & Steinbart, P. J. (2018). *Accounting information systems* (14th ed.). Pearson.
- Turban, E., Pollard, C., & Wood, G. (2018). *Information technology for management* (11th ed.). Wiley.
- Yakub. (2019). *Pengantar sistem informasi*. Graha Ilmu.