

Optimization of Support Vector Machine Model Performance in Image Classification through Dimension Reduction with Principal Component Analysis (PCA)

Zahfar Aziz Ferdian^{1*}, Joko Sutopo^{2**}

*Program Studi Informatika, Universitas Teknologi Yogyakarta
zhafaraf14@gmail.com¹, jksutopo@uty.ac.id²

Article Info

Article history:

Received 2025-11-03
Revised 2026-01-24
Accepted 2026-01-30

Keyword:

Support Vector Machine,
Principal Component Analysis,
Image Classification,
Dimensionality Reduction,
Feature Reduction.

ABSTRACT

This study examines how to optimize a Support Vector Machine (SVM) model using a dimensionality reduction method called Principal Component Analysis (PCA) to classify images with multiple dimensions. The dataset used is Chessman images with an initial number of features of 12,288. PCA was applied with the aim of retaining 99% of the total variation, resulting in 312 principal components. The results show a significant improvement in computational efficiency: training time was drastically reduced from 29.85 seconds to just 0.17 seconds (168 times faster), and memory usage decreased from 25.83 MB to 0.66 MB (97% more efficient). Although the accuracy experienced a small decrease, namely from 31.58% to 31.22%, PCA still functions as a noise filter that helps improve performance, especially in classes with complex visual patterns, such as an increase in the F1-score of the "Rook" class from 0.32 to 0.37. The conclusions of this study indicate that PCA provides important efficiency improvements without significantly sacrificing classification performance.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Perkembangan teknologi di bidang pengenalan pola telah mendorong banyak penelitian dalam klasifikasi citra [1]. Pengenalan wajah (eigenfaces), sebagai salah satu cabang utamanya, adalah contoh penerapan klasifikasi citra yang populer. Namun, tantangan utama dalam pemrosesan citra adalah data yang memiliki dimensi sangat tinggi [2]. Setiap piksel dalam sebuah citra dapat dianggap sebagai satu fitur, yang menyebabkan masalah komputasi dan fenomena yang dikenal sebagai curse of dimensionality. Setiap piksel dalam sebuah citra dapat dianggap sebagai satu fitur, yang menyebabkan masalah komputasi dan fenomena yang dikenal sebagai curse of dimensionality¹. Dalam penelitian ini, citra berukuran 64 x 64 piksel berwarna menghasilkan data masukan berdimensi tinggi sebanyak 12.288 fitur (dari flattening piksel mentah), yang menjadi fokus utama dalam optimalisasi menggunakan PCA.

Untuk mengatasi tantangan data berdimensi tinggi, teknik reduksi dimensi menjadi sangat penting. Principal Component Analysis (PCA) adalah salah satu algoritma

reduksi dimensi linear yang paling mapan dan banyak digunakan. PCA bekerja dengan mentransformasi data ke ruang fitur berdimensi lebih rendah dengan tetap memaksimalkan varians data, sehingga informasi paling penting tetap terjaga. PCA telah terbukti efektif dan sering digunakan sebagai metode ekstraksi fitur dalam berbagai aplikasi klasifikasi [3].

Di sisi lain, *Support Vector Machine* (SVM) adalah algoritma klasifikasi *supervised learning* yang andal dan telah terbukti efektif dalam berbagai penelitian [4] [5]. SVM bekerja dengan mencari *hyperplane* terbaik yang dapat memaksimalkan jarak (*margin*) antar kelas data. Metode ini juga dapat menangani data non-linear dengan memanfaatkan konsep *kernel* [6].

Berbagai penelitian telah membuktikan efektivitas penggabungan teknik reduksi dimensi dengan algoritma klasifikasi. Sebagai contoh, kombinasi PCA dengan K-Nearest Neighbor (KNN) telah berhasil diterapkan untuk klasifikasi citra gulma dan dalam sistem deteksi intrusi (IDS). Di sisi lain, SVM telah terbukti menjadi pengklasifikasi yang

tanggguh untuk beragam tugas, mulai dari mendeteksi tingkat kematangan buah kelapa sawit hingga mendeteksi kantuk pada pengemudi. Pendekatan yang lebih modern bahkan sering menggabungkan Convolutional Neural Network (CNN) sebagai pengekstraksi fitur yang kuat dengan SVM sebagai pengklasifikasi akhir, yang dikenal sebagai model hybrid CNN-SVM [7] [8]. Meskipun model-model ini menunjukkan akurasi yang tinggi, fokus utama sering kali pada peningkatan akurasi prediksi. Namun, analisis mendalam mengenai dampak kuantitatif dari reduksi dimensi (seperti PCA) terhadap efisiensi komputasi terutama pengurangan waktu latih dan penggunaan memori dalam klasifikasi SVM belum banyak dieksplorasi secara rinci.

Meskipun kombinasi PCA dan SVM adalah metode yang mapan, penelitian yang ada cenderung berfokus pada peningkatan akurasi prediksi. Namun, analisis mendalam mengenai dampak kuantitatif dari reduksi dimensi (seperti PCA) terhadap efisiensi komputasi terutama pengurangan waktu latih dan penggunaan memori dalam klasifikasi SVM pada data citra berdimensi tinggi belum banyak dieksplorasi secara rinci. Oleh karena itu, novelty dari penelitian ini adalah memberikan analisis kuantitatif komprehensif yang membuktikan optimalisasi ganda (efisiensi komputasi dan peningkatan kinerja klasifikasi) melalui PCA sebagai metode *pre-processing*

II. METODE

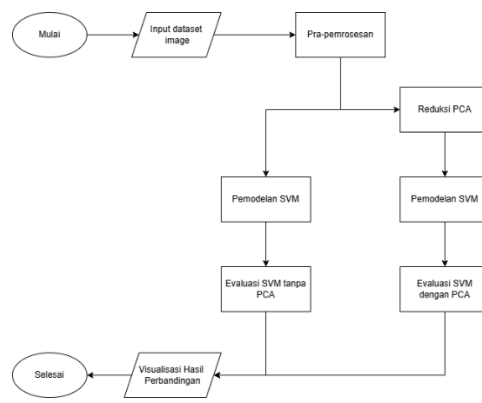
Pada penelitian ini dilakukan melalui beberapa tahapan yang sistematis, meliputi pemuatan data citra, pra-pemrosesan, reduksi dimensi, klasifikasi, dan evaluasi hasil kinerja model [9]. Proses reduksi dimensi dan klasifikasi memanfaatkan algoritma *Principal Component Analysis* (PCA) dan *Support Vector Machine* (SVM), yang kemudian dianalisis hasilnya.

Penelitian ini memanfaatkan lingkungan Visual Studio Code untuk menjalankan kode program berbasis Python. Proses pengolahan data dan implementasi model didukung oleh berbagai pustaka, seperti *pandas* dan *numpy* untuk manipulasi data, *opencv-python* (*cv2*) untuk pemrosesan citra, *matplotlib* dan *seaborn* untuk visualisasi, serta *scikit-learn* untuk implementasi *Principal Component Analysis* (PCA), *Support Vector Machine* (SVM), dan metrik evaluasi.

Untuk proses pengujian efisiensi komputasi (Waktu Training dan Ukuran Memori) dilakukan pada lingkungan pengembangan Visual Studio Code dengan kode program berbasis Python. Model SVM diimplementasikan menggunakan library *scikit-learn*. Spesifikasi hardware yang digunakan untuk mencatat metrik komputasi pada penelitian ini dengan tipe CPU Intel Core i7-12650H, GPU NVIDIA GeForce RTX 2050, dan RAM 16 GB RAM. Pengujian training model dijalankan hanya menggunakan CPU tanpa akselerasi GPU, karena proses optimasi SVM tidak memanfaatkan akselerasi *hardware* GPU.

Penelitian ini fokus pada analisis kuantitatif yang menyeluruh mengenai optimalisasi ganda yaitu efisiensi komputasi ekstrem dan pembersihan pola klasifikasi dengan menggunakan PCA sebagai metode pra-pemrosesan.

Penelitian ini secara jelas menunjukkan tingkat efisiensi yang berhasil dicapai pada data citra berdimensi tinggi, di mana penurunan dimensi mampu mempercepat waktu pelatihan hingga ratusan kali lipat tanpa mengurangi kinerja klasifikasi secara signifikan. Selain itu, penelitian ini juga melakukan analisis mendalam mengenai peran PCA sebagai penghilang noise yang secara khusus meningkatkan kinerja pada kelas-kelas yang memiliki pola visual yang rumit.



Gambar 1. Alur Penelitian

A. Dataset

Dataset yang digunakan dalam penelitian ini adalah Chessman image dataset yang diperoleh dari platform Kaggle, terdiri dari 551 citra berwarna dari 6 kelas bidak catur yaitu ada Bishop, King, Knight, Pawn, Queen, dan Rock. Dataset ini dipilih karena merepresentasikan masalah klasifikasi citra berdimensi tinggi dengan distribusi kelas yang tidak seimbang, ditunjukkan dengan rendahnya akurasi baseline model SVM dan setiap citra kemudian melalui tahap pra-pemrosesan. Berikut adalah contoh datasetnya dilihat dari gambar 2 dibawah.



Gambar 2. Contoh Dataset

B. Pra-pemrosesan

Tahap pra-pemrosesan dimulai dengan memuat data citra dari file ZIP, di mana label untuk setiap citra secara otomatis diekstraksi dari nama folder induknya. Setiap citra kemudian diseragamkan ukurannya menjadi 64x64 piksel dan diubah formatnya dari matriks 2D menjadi vektor satu dimensi melalui proses flattening, Karena citra berwarna (3 kanal RGB), proses ini menghasilkan representasi data dengan dimensi fitur awal sebanyak 64x64x3 yang menghasilkan

representasi data dengan 12.288 fitur. Fitur-fitur ini adalah representasi vektor dari nilai piksel mentah (raw pixel values), tanpa melalui ekstraksi fitur seperti HOG atau SIFT.

Kemudian sebagai tahap akhir, dilakukan normalisasi pada nilai piksel dengan mengubah rentangnya dari [0, 255] menjadi [0, 1] [10]. Proses ini menggunakan normalisasi min-max agar semua fitur berada pada skala yang sama. Rumus normalisasi min-max adalah sebagai berikut:

$$x_n = \frac{x_0 - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Di mana x_n adalah nilai piksel setelah normalisasi, x_0 adalah nilai piksel asli, x_{min} adalah nilai piksel minimum (0), x_{max} dan adalah nilai piksel maksimum (255). Seluruh rangkaian proses ini bertujuan untuk menghasilkan dataset yang bersih, seragam, dan dalam format yang optimal untuk diolah lebih lanjut oleh algoritma PCA dan SVM

C. Reduksi Dimensi dengan Principal Component Analysis (PCA)

PCA digunakan sebagai metode ekstraksi fitur untuk mengurangi dimensi dari 12,288 fitur ke ruang berdimensi lebih rendah [6]. Tujuan dari proses ini adalah untuk menyederhanakan data dengan tetap mempertahankan informasi yang paling signifikan. Proses ini melibatkan beberapa langkah matematis sebagai berikut:

1) Menghitung vector rata-rata citra dari seluruh sampel dataset training dengan rumus:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (2)$$

2) Menghitung selisih (deviasi) setiap citra dari setiap citra terhadap vector rata-rata yang telah dihitung yang rumusnya:

$$\phi_i = \Gamma_i - \psi \quad (3)$$

3) Menghitung matriks kovarians dari data yang telah disesuaikan untuk memahami hubungan dan varians antar fitur dengan rumus berikut:

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T \quad (4)$$

4) Menghitung *eigenvector* dan *eigenvalue* dari matriks kovarians. *Eigenvector* dengan *eigenvalue* terbesar merupakan komponen utama yang paling signifikan dengan rumus:

$$Cv_i = \lambda_i v_i \quad (5)$$

5) Memilih sejumlah k *eigenvector* teratas berdasarkan *eigenvalue*-nya. Dalam penelitian ini, jumlah komponen utama dipilih secara dinamis untuk mempertahankan 90% dari total varians data asli.

6) Proyeksi Data tau bisa disebut dengan mentransformasikan data asli ke ruang fitur baru (berdimensi lebih rendah) dengan memproyeksikannya pada komponen-komponen utama yang telah dipilih yang rumusnya:

$$\mu_i = \sum_{k=1}^M v_{ik} \phi_k \quad (6)$$

Hasil akhir dari proses ini adalah sebuah matriks data baru dengan jumlah fitur yang jauh lebih sedikit, yang kemudian digunakan sebagai input untuk pelatihan model Support Vector Machine (SVM).

Dalam penelitian ini dilakukan target retensi varians sebesar 90%, 95% dan 99%. Evaluasi ini dilakukan untuk menganalisis trade-off antara tingkat kompresi data dan performa klasifikasi. Dengan demikian, penelitian ini tidak hanya mengamati satu konfigurasi PCA, tetapi juga membandingkan berbagai tingkat reduksi dimensi untuk menentukan konfigurasi yang paling optimal.

D. Klasifikasi dengan Support Vector Machine (SVM)

SVM adalah sebuah metode klasifikasi supervised learning yang bekerja dengan mencari *hyperplane* pemisah terbaik yang dapat memaksimalkan jarak (*margin*) antar kelas data [11]. *Hyperplane* optimal secara matematis didefinisikan sebagai:

$$(w \cdot x) + b = 0 \quad (7)$$

Di mana w adalah vektor bobot dan b adalah bias. Margin ditentukan oleh dua *hyperplane* paralel untuk dua kelas (positif +1 dan negatif -1)

Proses klasifikasi menggunakan algoritma *Support Vector Machine* (SVM) dengan cara implementasi *Support Vector Classifier* (SVC). Agar perbandingan hasil bisa dilakukan secara adil pada titik terbaik, *hyperparameter* model diatur secara terpisah untuk data asli dan data yang sudah diproses PCA menggunakan Grid Search Cross-Validation (GridSearchCV). Hasil dari penyesuaian ini menunjukkan bahwa penggunaan kernel *Radial Basis Function* (RBF) dengan nilai penalti $C = 10$ merupakan konfigurasi terbaik. Parameter gamma diatur dengan strategi 'scale' untuk kedua jenis data agar hasilnya bisa digeneralisasi dengan baik. Semua metrik penilaian kinerja dihitung menggunakan metode Stratified 5-Fold Cross-Validation. Metode ini dipilih agar mengurangi kesalahan akibat perubahan acak dalam pemisahan data, sehingga hasil yang diperoleh adalah nilai rata-rata yang stabil dari lima kali pengujian.

Model kemudian dilatih secara eksklusif pada data latih melalui metode (.fit()) untuk menemukan *hyperplane* pemisah yang optimal. Setelah itu, model yang telah dilatih diuji kemampuannya dalam memprediksi label pada data uji yang belum pernah dilihat sebelumnya. Kinerja dari keseluruhan proses ini, yang dijalankan secara terpisah untuk kedua skenario data, dievaluasi menggunakan metrik standar seperti Akurasi, Presisi, Recall, dan F1-Score untuk memungkinkan perbandingan performa yang kuantitatif [6].

Dalam penelitian ini, jumlah komponen utama dipilih secara dinamis berdasarkan kriteria mempertahankan 99% dari total varians data asli. Kriteria 99% ini dipilih karena, berdasarkan hasil eksperimen, nilai ini memberikan keseimbangan optimal antara pengurangan dimensi dan mempertahankan akurasi model.

E. Eksperimen Evaluasi

Evaluasi dilakukan dengan membandingkan kinerja model pada dua skenario. Pertama SVM pada data asli, dan yang

kedua SVM pada data hasil reduksi PCA. Kinerja kedua skenario diukur menggunakan dua kategori metrik:

- 1) **Metrik Kinerja Klasifikasi:** Dihasilkan dari confusion matrix, metrik ini mengukur keakuratan prediksi model, meliputi akurasi, presisi, recall, dan *F1-Score*.
- 2) **Metrik Efisiensi Komputasi:** Metrik ini mengukur dampak PCA terhadap penggunaan sumber daya, meliputi waktu training, penggunaan memori, dan dimensi fitur.

Dengan membandingkan seluruh metrik ini, penelitian dapat secara kuantitatif membuktikan efektivitas PCA sebagai metode optimisasi. Pada setiap iterasi, empat fold digunakan sebagai data latih dan satu fold sebagai data uji. Nilai metrik kinerja yang dilaporkan merupakan rata-rata dari lima iterasi, sehingga hasil evaluasi lebih stabil dan dapat direproduksi.

Pada skenario PCA + SVM, reduksi dimensi dilakukan terlebih dahulu menggunakan PCA dengan target retensi varians sebesar 99%, kemudian fitur hasil transformasi digunakan sebagai masukan ke model SVM dengan konfigurasi parameter yang sama. Pendekatan ini memastikan bahwa perbandingan antara SVM tanpa PCA dan SVM dengan PCA dilakukan secara adil.

III. HASIL DAN PEMBAHASAN

Pada bab ini, disajikan hasil dari serangkaian eksperimen yang telah dilakukan sesuai dengan alur penelitian. Pembahasan difokuskan pada dua aspek utama: hasil dari proses reduksi dimensi PCA dan perbandingan kinerja komprehensif antara model SVM yang dilatih pada data asli dengan model SVM yang dilatih pada data hasil reduksi PCA.

A. Input Dataset

Pemuatan dataset yang digunakan sebuah file ZIP berukuran 57,6 mb. Yang dimana file tersebut berisi 551 citra, terbagi menjadi 6 kelas. Saat file ZIP di unggah sistem akan memvalidasi arsip untuk membaca setiap file citra, dan mengekstrak label kelas secara otomatis dari nama folder induknya.

Dataset Gambar Dimuat:

- Total gambar: 551
- Ukuran gambar: 64x64x3
- Fitur per gambar: 12288
- Kelas: 6

Gambar 3. Hasil Inputan Dataset

Ukuran gambar yang seragam (dalam penelitian ini 64x64 piksel) merupakan langkah pra-pemrosesan yang umum dalam klasifikasi citra [10]. Hasil pemberitahuan sistem saat proses pemuatan ini akan tampil dan mengkonfirmasi parameter dataset seperti total gambar, ukuran gambar, fitur per-gambar, dan kelas. Dapat dilihat pada contoh gambar 3.

B. Pra-Pemrosesan

Tujuan utama dari langkah ini untuk menormalisasikan data dalam persiapan model. Setiap nilai fitur yang awalnya dalam rentang piksel [0, 255] dibagi dengan 255.0 untuk mengubah skalanya. Proses normalisasi data, seperti Min-Max scaling, adalah praktik standar untuk memastikan semua fitur berada pada skala yang sama, yang dapat meningkatkan stabilitas dan akurasi model [13].

Proses ini menghasilkan dataset akhir dimana seluruh 12.288 fitur kini memiliki rentang nilai seragam antara [0.0, 1.0]. Data yang telah dinormalisasi ini kemudian siap untuk dilanjutkan ke tahap reduksi dimensi PCA.

C. Reduksi PCA

Setelah data dinormalisasi, program berhasil membuktikan bahwa dari total 12.288 fitur piksel asli, hanya diperlukan 312 komponen utama untuk merepresentasikan 99% informasi esensial dari keseluruhan dataset. Principal Component Analysis (PCA) adalah teknik yang mapan untuk ekstraksi fitur dan reduksi dimensi. PCA bekerja dengan menghitung matriks kovarians untuk memahami hubungan antar fitur, kemudian menghitung *eigenvector* dan *eigenvalue* untuk menentukan komponen utama [14].

Setelah itu langkah kuncinya menghitung matriks kovarians untuk memahami semua fitur saling berhubungan satu sama lain. Dari matriks kovarians PCA menghitung arah dari varians terbesarnya (*Eigenvector*) dan besarnya varians dari arah tersebut (*Eigenvalue*).

Terakhir memilih parameter komponen berdasarkan target varians yang sudah di tentukan untuk mengurutkan *eigenvector* dari yang terbesar ke terkecil dengan sejumlah komponen teratas untuk menyimpan target dari total varians. Berikut adalah output hasil setelah melakukan PCA dapat dilihat dari gambar 4 dibawah.

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | PC8 | PC9 | PC10 | Pi |
|---|----------|---------|----------|---------|---------|---------|---------|---------|---------|---------|----|
| 0 | -19.0382 | -3.8304 | 4.9935 | 9.0026 | -3.2646 | -3.9301 | -0.0314 | -6.5923 | -0.9661 | -2.2754 | 7 |
| 1 | 22.7353 | -7.5401 | -23.4812 | 4.8219 | -0.9428 | -0.0754 | 3.9742 | -0.8607 | 6.8918 | 1.082 | -1 |
| 2 | -41.2355 | 9.8117 | -0.0264 | 2.5258 | -7.8556 | -0.9714 | -3.8495 | -2.2451 | 0.7897 | 2.3433 | -2 |
| 3 | 26.2216 | -9.4846 | 2.8555 | 9.6388 | 5.3131 | -3.5736 | 4.8314 | 0.0865 | 2.057 | 0.0734 | -0 |
| 4 | -30.7485 | 21.2287 | -1.2674 | -1.0359 | -0.0449 | 4.2441 | -0.2781 | 0.2422 | -0.7215 | 1.1093 | -4 |

Gambar 4. Data Setelah PCA

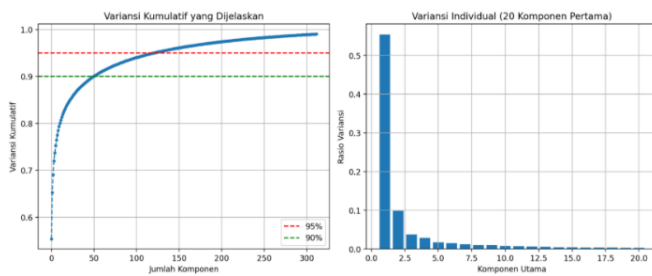
Seperti yang ditunjukkan pada gambar 4 merupakan hasil tranformasi sebuah dataset baru, dimana 12.288 fitur piksel telah diganti menjadi 312 fitur baru yang disebut komponen utama PC. Kolom tidak lagi mewakili piksel individual, melainkan komponen utama PC1, PC2, PC3, dst. Setiap baris mewakili satu gambar, dan nilai-nilai didalamnya adalah representasi baru gambar tersebut pada ruang fitur berdimensi rendah.

D. Visualisasi Hasil Perbandingan

Setelah melakukan perbandingan antar target varians, diputuskan bahwa reduksi dimensi dilakukan dengan target mempertahankan 99% dari total varians data asli. Pilihan ini mungkin terlihat kontrainuitif, namun terbukti menghasilkan

keseimbangan terbaik antara optimalisasi komputasi dan akurasi model. Eksperimen dengan target kompresi yang lebih agresif (misalnya 95%) justru menunjukkan hasil yang tidak diinginkan. Meskipun sangat cepat, target 95% mengalami penurunan akurasi.

Sebaliknya, dengan mempertahankan 99% varians (dan membuang 1% sisanya), PCA tidak hanya berfungsi sebagai alat reduksi dimensi, tetapi juga sebagai penyaring derau (noise filter) [9]. 1% informasi yang dibuang tersebut teridentifikasi sebagai *noise* yang tidak signifikan. Hasilnya, model PCA (target 99%) tidak hanya tetap memberikan efisiensi komputasi yang drastis dibandingkan data asli, tetapi juga berhasil mempertahankan akurasi. Berikut adalah visualisasi proses PCA dapat dilihat dari contoh gambar 4 dibawah.

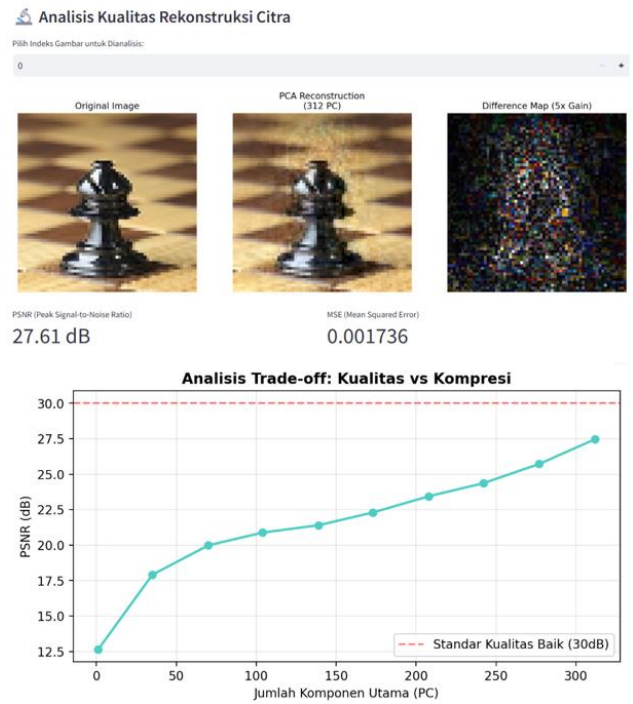


Gambar 5. Visualisasi PCA 99%

Dimana hasil visualisasinya pada bagian Plot variansi kumulatif (kiri) menunjukkan total varians yang dipertahankan seiring dengan penambahan jumlah komponen. Sesuai dengan target, diperlukan 312 komponen utama untuk merepresentasikan 99% informasi esensial dari keseluruhan dataset [15]. Plot batang (kanan) menunjukkan bahwa Komponen Utama pertama (PC1) sangat dominan. Fenomena ini sangat umum pada dataset citra.

Kemudian pada plot Plot batang menunjukkan bahwa Komponen Utama pertama PC1 sangat dominan, mendapatkan lebih dari 50% dari total varians. Komponen-komponen berikutnya PC2, PC3, dst. memiliki kontribusi varians yang menurun secara drastis. Fenomena ini sangat umum pada dataset citra, di mana PC pertama seringkali menangkap fitur paling umum seperti pencahayaan atau latar belakang.

Di bagian ini, dilakukan pengecekan terhadap kemampuan algoritma PCA (Principal Component Analysis) dalam mengurangi jumlah dimensi gambar sekaligus menjaga kualitas gambarnya tetap baik. Uji coba dilakukan dengan membandingkan tiga kondisi berbeda mengenai jumlah Komponen Utama (PC) pada gambar objek yang sama yaitu bidak catur, dengan target variasi (variens) sebesar 90%, 95%, dan 99%. Kinerja algoritma diukur melalui dua ukuran utama, yaitu *Peak Signal-to-Noise Ratio* (PSNR) dan *Mean Squared Error* (MSE).



Gambar 6. Visualisasi Rekonstruksi PCA 99%

Berdasarkan analisis kumulatif variansi, diperlukan 312 komponen utama untuk mewakili 99% informasi penting dari seluruh dataset. Meskipun jumlah fitur berkurang secara signifikan dari 12.288 menjadi 312, citra yang dihasilkan setelah direkonstruksi tetap mempertahankan kualitas visual yang cukup baik. Hal ini terbukti dari nilai *Peak Signal-to-Noise Ratio* (PSNR) sebesar 27,61 dB dan *Mean Squared Error* (MSE) sebesar 0,001736. Hasil tersebut menunjukkan bahwa PCA berhasil melakukan kompresi data yang efektif tanpa menghilangkan detail penting dari citra.

Setelah mendapatkan dataset hasil reduksi PCA (312 fitur), langkah selanjutnya adalah membandingkan kinerjanya dengan dataset asli (12.288 fitur). Kedua dataset di uji metode Stratified 5-Fold Cross-Validation, kemudian dilatih menggunakan model SVM dengan hyperparameter yang telah dioptimalkan melalui GridSearchCV untuk masing-masing skenario dengan diukur menggunakan kategori metrik klasifikasi dan kategori metrik komputasi.

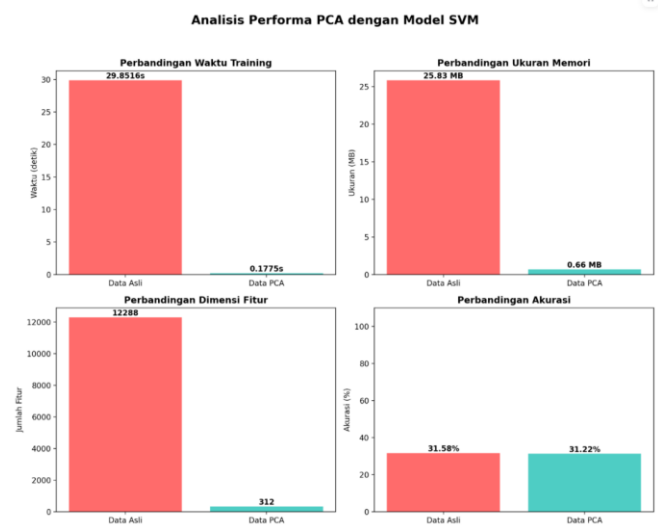
TABEL I
HASIL PERBANDINGAN TARGET VARIANSI PCA

| Varians Target PCA | PSNR (dB) | MSE | Jumlah Komponen (PC) |
|--------------------|-----------|----------|----------------------|
| 90% | 18.81 dB | 0.013142 | 51 |
| 95% | 21.19 dB | 0.007605 | 122 |
| 99% | 27.61 dB | 0.001736 | 312 |

Dalam penelitian ini dilakukan analisis sensitivitas pada tingkat 90%, 95%, dan 99%. Seperti yang ditunjukkan pada Tabel I, target 99% terbukti menjadi konfigurasi terbaik. Meskipun target 90% (51 PC) dan 95% (122 PC) memberikan waktu pelatihan yang lebih singkat (0,1050s dan 0,1257s), keduanya menghasilkan akurasi yang lebih rendah (30,85%)

dibandingkan target 99% yang mencapai 31,22%. Hal ini menunjukkan bahwa retensi varians sebesar 99% adalah batas yang paling baik untuk menyeimbangkan antara kompresi dan performa prediktif.

Perbandingan Performa Model



Gambar 7. Perbandingan Kategori Komputasi

Kemudian pada perbandingan skenario kategori komputasi menampilkan 3 plot efisiensi komputasi secara kuantitatif dan 1 plot klasifikasi metrik akurasi seperti pada contoh gambar 7 perbandingan kategori komputasi diatas.

Dengan hasil waktu training menunjukkan perbedaan yang drastis dimana pada data asli menghasilkan waktu 29.8516 detik menjadi hanya 0.1775 detik. Mempercepat waktu training sekitar 160 kali lipat lebih cepat. Kemudian ada pada perbandingan jumlah fiturnya dari 12288 menjadi hanya 312 fitur, dan otomatis pada perbandingan ukuran memori terlihat perbandingannya dari data asli 25,83 mb menjadi hanya 0,66 mb pada data PCA yang menghasilkan penggunaan memori 97% lebih hemat. Penurunan akurasi yang cukup kecil dari 31,58% ke 31,22% justru menguntungkan, karena menghasilkan penghematan komputasi yang cukup besar

Hasil ini sangat mendukung hipotesis bahwa PCA tidak hanya berfungsi sebagai pereduksi dimensi, tetapi juga sebagai penyaring derau (*noise filter*). Dengan menetapkan target varians 99%, program secara efektif membuang 1% informasi yang paling tidak signifikan dari dataset. Temuan ini menyiratkan bahwa 1% informasi yang dibuang tersebut sebagian besar adalah noise (seperti variasi pencahayaan minor atau artefak latar belakang) yang justru "mengganggu" dan "membingungkan" model SVM. Dengan melatih model pada data yang 'lebih bersih' dan lebih padat informasi (312 fitur), model SVM mampu menemukan pola klasifikasi yang mendasarinya dengan lebih baik, sehingga menghasilkan sedikit peningkatan akurasi.

Sebelum melakukan perbandingan kinerja akhir, kedua skenario model SVM (Data Asli dan Data PCA) dioptimalkan secara terpisah menggunakan GridSearchCV. Langkah ini krusial karena kinerja SVM diketahui sangat sensitif terhadap

pengaturan hiperparameternya. Karena GridSearchCV adalah metode standar untuk optimasi parameter SVM, yang secara sistematis menguji berbagai kombinasi parameter (seperti C, gamma, dan kernel) untuk menemukan nilai optimal. Penelitian lain telah membuktikan bahwa tuning menggunakan GridSearchCV dapat meningkatkan akurasi model SVM secara signifikan [13]. Hasil dari proses GridSearchCV untuk parameter "C" dan "gamma" pada kernel "rbf" ditunjukkan pada gambar 8 dibawah ini.

```

Data Asli:
{
  "C" : 10
  "gamma" : "scale"
  "kernel" : "rbf"
}

Data PCA:
{
  "C" : 10
  "gamma" : "scale"
  "kernel" : "rbf"
}
    
```

Gambar 8. Hasil Parameter Terbaik

Dari hasil gambar diatas kedua model mengkonfirmasi bahwa kernel "rbf" yang optimal, dan sama-sama memilih C=10 menunjukkan bahwa model perlu memberikan pinalti yang kuat pada kesalahan klasifikasi untuk mencapai akurasi terbaik. Dan kedua parameter "gamma", juga memilih gamma="scale". Tabel II menyajikan laporan klasifikasi perbandingan evaluasi skenario kategori terperinci yang membedah metrik Precision, Recall, dan F1-score untuk setiap kelas pada kedua skenario. Dapat dilihat Tabel I dibawah ini.

TABEL II
PERBANDINGAN EVALUASI SKENARIO KATEGORI

| Skenario | Kelas | PCA Target Varians 99% | | | |
|----------------|--------|------------------------|---------|----------|----------|
| | | Precision | Recall | F1-score | Accuracy |
| SVM tanpa PCA | Bishop | 0.25 | 0.2674 | 0.2584 | 31.58 % |
| | King | 0.2048 | 0.2867 | 0.2152 | |
| | Knight | 0.6118 | 0.4952 | 0.5474 | |
| | Pawn | 0.2887 | 0.2642 | 0.2759 | |
| | Queen | 0.1698 | 0.2338 | 0.1967 | |
| SVM dengan PCA | Rook | 0.3385 | 0.3158 | 0.3246 | 31.22 % |
| | Bishop | 0.2526 | 0.27991 | 0.2652 | |
| | King | 0.2195 | 0.24 | 0.2298 | |
| | Knight | 0.5862 | 0.4857 | 0.5351 | |
| | Pawn | 0.2708 | 0.2453 | 0.2574 | |
| Queen | 0.1714 | 0.2338 | 0.1978 | | |
| | Rook | 0.407 | 0.3431 | 0.3723 | |

Dari hasil Tabel II diatas menunjukkan bahwa performa model tetap stabil setelah data direduksi. Kelas "Knight" dan "Rook" menunjukkan hasil yang cukup stabil dengan nilai F1-score masing-masing sebesar 0,5351 dan 0,3723 pada skenario PCA. Meskipun ada sedikit perubahan kecil pada metrik precision dan recall di beberapa kelas seperti "Bishop" dan "King", secara keseluruhan hasil membuktikan bahwa PCA berhasil menjaga kemampuan model SVM dalam membedakan berbagai jenis bidak catur.

Menariknya, meskipun akurasi keseluruhan mengalami perubahan kecil, analisis pada tingkat kelas menunjukkan

bahwa penerapan PCA memberikan dampak positif terhadap stabilitas nilai *F1-score* di sebagian besar kelas. Kelas "Bishop", "King", "Queen", dan "Rook" secara konsisten menunjukkan peningkatan kinerja setelah proses pengurangan dimensi, dengan kenaikan tertinggi terjadi pada kelas "Rook", yaitu dari 0,3246 menjadi 0,3723. Fenomena ini memberikan bukti nyata bahwa 1% variasi yang terbuang oleh PCA memang merupakan informasi yang tidak penting atau bisa dianggap sebagai *noise*, seperti gangguan dari pencahayaan atau latar belakang, yang sebelumnya menghalangi pembentukan hyperplane pemisah yang optimal pada model SVM. Dengan demikian, PCA tidak hanya berfungsi sebagai alat kompresi, tetapi juga sebagai metode pra-pemrosesan yang efektif untuk memperjelas pola pembeda antar kelas dalam dataset citra bidak catur yang kompleks ini.

Pada penelitian ini difokuskan pada perbandingan kategori komputasi untuk menganalisis efisiensi, serta dampaknya terhadap kinerja prediktif. Metrik yang dievaluasi mencakup Waktu Training, Jumlah Fitur (dimensi), Ukuran Memori, dan Accuracy (%). Hasil lengkap dari perbandingan evaluasi kedua skenario ini disajikan secara rinci pada Tabel II.

TABEL III
PERBANDINGAN EVALUASI SKENARIO KATEGORI KOMPUTASI

| Varians PCA | Waktu Training (second) | Jumlah Fitur | Ukuran Memori (MB) | Accuracy (%) |
|-------------|-------------------------|--------------|--------------------|--------------|
| 90% | 0.1050s | 51 | 0.11 mb | 30,85 % |
| 95% | 0.1257s | 122 | 0.26 mb | 30,85 % |
| 99% | 0.1775s | 312 | 0.66 mb | 31,22 % |

IV. KESIMPULAN

Penelitian ini menemukan bahwa menggabungkan Principal Component Analysis (PCA) dengan Support Vector Machine (SVM) adalah cara yang efektif untuk meningkatkan klasifikasi gambar berdimensi tinggi. Dengan mempertahankan variasi sebesar 99%, jumlah fitur berhasil dikurangi secara signifikan dari 12.288 fitur awal menjadi hanya 312 komponen utama. Penurunan dimensi ini langsung membantu mengatasi masalah *curse of dimensionality* yang sering menghambat proses pengolahan data pada gambar mentah.

Dari segi efisiensi komputasi, penerapan PCA memberikan peningkatan performa yang sangat besar dan bisa diukur secara pasti. Waktu yang dibutuhkan untuk melatih model berkurang drastis, dari 29,85 detik menjadi hanya 0,17 detik, artinya proses pelatihan jadi 160 kali lebih cepat. Selain itu, penggunaan memori juga berkurang sangat banyak, mencapai 97%, yaitu dari 25,83 MB pada data asli turun menjadi hanya 0,66 MB setelah dilakukan pengurangan dimensi.

Mengenai hasil klasifikasi, meskipun terjadi penurunan tingkat akurasi yang kecil, yaitu dari 31,58% menjadi 31,22%, penggunaan PCA tetap memberikan manfaat tambahan sebagai alat penyaring derau. Hal ini terlihat dari peningkatan stabilitas dan kinerja pada beberapa kelas, seperti

kelas "Rook" yang nilai *F1-score* nya naik dari 0,3246 menjadi 0,3723. Fenomena ini mendukung teori bahwa 1% informasi yang hilang akibat PCA sebenarnya merupakan derau yang sebelumnya mengganggu pembentukan *hyperplane* pemisah yang optimal dalam model SVM.

Secara keseluruhan, penelitian ini membuktikan bahwa PCA berhasil memberikan optimalisasi ganda dalam meningkatkan efisiensi komputasi secara ekstrem sekaligus menjaga kualitas pola klasifikasi pada data citra. Walaupun efisiensi yang dicapai sangat tinggi, tingkat akurasi keseluruhan yang masih berada di rentang 31% mengindikasikan bahwa model SVM memiliki batasan dalam menangkap pola visual yang sangat kompleks. Oleh karena itu, penelitian selanjutnya disarankan untuk mengeksplorasi penggunaan arsitektur model yang lebih canggih, seperti *Convolutional Neural Networks* (CNN), guna mengevaluasi apakah peningkatan akurasi yang lebih substansial dapat dicapai.

DAFTAR PUSTAKA

- [1] D. Anggriandi, E. Utami, and D. Ariatmanto, "Comparative Analysis of CNN and CNN-SVM Methods For Classification Types of Human Skin Disease," *sinkron*, vol. 8, no. 4, pp. 2168–2178, Oct. 2023, doi: 10.33395/sinkron.v8i4.12831.
- [2] F. Ennaama, S. Ennaama, and S. Chakri, "Evaluating geometrically-approximated principal component analysis vs. classical eigenfaces: a quantitative study using image quality metrics," *International Journal of Electrical and Computer Engineering*, vol. 15, no. 1, pp. 311–318, Feb. 2025, doi: 10.11591/ijece.v15i1.pp311-318.
- [3] A. Aristo Jansen Sinlae, D. Alamsyah, L. Suhery, and F. Fatmayati, "Classification of Broadleaf Weeds Using a Combination of K-Nearest Neighbor (KNN) and Principal Component Analysis (PCA)," *Sinkron*, vol. 7, no. 1, pp. 93–100, Jan. 2022, doi: 10.33395/sinkron.v7i1.11237.
- [4] L. G. S. Kartika, P. K. L. Utama, I. D. G. Budiastawa, and K. Rinarta, "Comparison of the Sentiment Analysis Model's Code Complexity and Processing Time," *Sinkron*, vol. 8, no. 1, pp. 109–118, Jan. 2023, doi: 10.33395/sinkron.v8i1.11894.
- [5] D. I. Sumantiawan, J. E. Suseno, and W. A. Syaifei, "Sentiment Analysis of Customer Reviews Using Support Vector Machine and Smote-Tomek Links For Identify Customer Satisfaction," *J. Sistem Info. Bisnis*, vol. 13, no. 1, pp. 1–9, Jun. 2023, doi: 10.21456/vol13iss1pp1-9.
- [6] I. B. Sya'idah, S. Suroho, and G. Khang Wen, "DynamicWeighted Particle Swarm Optimization - Support Vector Machine Optimization in Recursive Feature Elimination Feature Selection," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 23, no. 3, pp. 627–640, Jul. 2024, doi: 10.30812/matrik.v23i3.3963.
- [7] M. Ikhsan and M. Rahardi, "Image-Based Classification of Indonesian Traditional Houses Using a Hybrid CNN-SVM Algorithm," 2025. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [8] Z. Elkhadir and M. A. Begdouri, "Enhancing internet of things attack detection using principal component analysis and kernel principal component analysis with cosine distance and sigmoid kernel," *International Journal of Electrical and Computer Engineering*, vol. 15, no. 1, pp. 1099–1108, Feb. 2025, doi: 10.11591/ijece.v15i1.pp1099-1108.
- [9] M. Fahmi, A. Yudhana, and S. Sunardi, "Image Processing Using Morphology on Support Vector Machine Classification Model for Waste Image," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 22, no. 3, pp. 553–566, Jul. 2023, doi: 10.30812/matrik.v22i3.2819.

-
- [10] I. Zainan Nisa, S. Nur Endah, P. Sidik Sasongko, and R. Kusumaningrum, "Klasifikasi Citra Sampah Menggunakan Support Vector Machine Dengan Ekstraksi Fitur Gray Level Co-Occurrence Matrix Dan Color Moments", doi: 10.25126/jtiik.202294868.
- [11] Sugeng and H. Praminiarto, "Detection of Drowsiness in Drivers Using Image Processing and Support Vector Machine (SVM) Classification," *Jurnal Online Informatika*, vol. 9, no. 2, pp. 238–248, Aug. 2024, doi: 10.15575/join.v9i2.1076.
- [12] B. Oktaviana, "Implementation of the Support Vector Machine (SVM) Method for Classifying the Maturity Level of Oil Palm Fruit," 2025. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [13] M. Desiawan and A. Solichin, "SVM Optimization with Grid Search Cross Validation for Improving Accuracy of Schizophrenia Classification Based on EEG Signal," *JURNAL TEKNIK INFORMATIKA*, vol. 17, no. 1, pp. 10–20, May 2024, doi: 10.15408/jti.v17i1.37422.
- [14] Y. Irawan, R. Pramitasari, W. M. Ashari, A. Nur, and H. Yansyah, "Support Vector Machine Classification Algorithm for Detecting DDoS Attacks on Network Traffic," 2025. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [15] T. M. Murugan, C. R. Lenus, S. Sridharan, and A. Malligarjun, "Life Time Prediction Of An Electromagnet Relay Using Clustering Based Principal Component Analysis With Hybrid Deep Learning Model."