Indonesian Journal of Multidisciplinary on Social and Technology
Homepage: https://journal.ilmudata.co.id/index.php/ijmst
Vol. 4 No. 3 (2026) pp: 5111-5126
P-ISSN: 2986-6790, e-ISSN: 2986-6782

Klasifikasi Kualitas Apel Berdasarkan Ciri Menggunakan Metode PCA dan SVM

Gregorius Nathanyel Benedict, Irene Carl Lewis, Panicia, Ryan Hermanto, Paquita Melawi, Samuel Gregorio Criswell Kastono, Melfanny Leono
Teknik Informatika, Universitas Tarumanagara
gregorius.535230114@stu.untar.ac.id, irene.535230185@stu.untar.ac.id, panicia.535230059@stu.untar.ac.id, ryan.535230147@stu.untar.ac.id, paquita.535230062@stu.untar.ac.id, samuel.535230077@stu.untar.ac.id, melfanny.535230152@stu.untar.ac.id

Abstrak

Penelitian ini bertujuan mengklasifikasikan kualitas buah apel ke dalam kelas apel bagus dan apel jelek menggunakan Principal Component Analysis (PCA) dan Support Vector Machine (SVM) berbasis citra digital. Otomasi penilaian kualitas diperlukan agar proses pemeriksaan buah dalam jumlah besar dapat dilakukan secara lebih cepat dan konsisten. Tahapan penelitian dimulai dari pre-processing citra dengan konversi gambar berwarna menjadi grayscale, kemudian dilakukan ekstraksi fitur warna, bentuk, dan tekstur. Dataset hasil ekstraksi diperiksa terhadap missing value dan outlier; outlier dideteksi menggunakan Mahalanobis Distance sebelum data dibagi menjadi data training dan testing dengan rasio 80:20. Reduksi dimensi selanjutnya dilakukan menggunakan PCA. Dua komponen utama, yaitu PC1 dan PC2, dipertahankan karena secara kumulatif mampu merepresentasikan 98,09% variansi data. Data hasil transformasi PCA digunakan sebagai masukan untuk membangun model klasifikasi SVM. Evaluasi dilakukan menggunakan confusion matrix serta metrik akurasi, precision, recall, dan F1-score, disertai cross-validation untuk menilai konsistensi model. Hasil pengujian menunjukkan akurasi sebesar 87,85% pada data training dan 88,40% pada data testing, sedangkan akurasi cross-validation mencapai 86,80%. Pada data testing, precision mencapai 93%, recall 85%, dan F1-score 89%. Hasil tersebut menunjukkan bahwa kombinasi PCA dan SVM efektif untuk klasifikasi kualitas apel berbasis ciri visual dengan dimensi fitur yang lebih ringkas. Pendekatan ini berpotensi mendukung proses quality control buah secara otomatis dan efisien.

Kata kunci: *Apel, Citra Digital, Klasifikasi, PCA, SVM.*

Abstract

This study aims to classify apple quality into good and poor categories using Principal Component Analysis (PCA) and Support Vector Machine (SVM) based on digital images. Automated quality assessment is needed to enable large-scale fruit inspection to be conducted more quickly and consistently. The research process begins with image pre-processing by converting color images into grayscale, followed by the extraction of color, shape, and texture features. The extracted dataset was examined for missing values and outliers, with outliers detected using Mahalanobis Distance before the data were divided into training and testing sets with an 80:20 ratio. Dimensionality reduction was then performed using PCA. Two principal components, PC1 and PC2, were retained because they cumulatively represented 98.09% of the data variance. The PCA-transformed data were used as input to build the SVM classification model. Model performance was evaluated using a confusion matrix and accuracy, precision, recall, and F1-score metrics, along with cross-validation to assess model consistency. The results showed an accuracy of 87.85% on the training data and 88.40% on the testing data, while cross-validation accuracy reached 86.80%. On the testing data, precision reached 93%, recall 85%, and F1-score 89%. These results indicate that the combination of PCA and SVM is effective for classifying apple quality based on visual features while reducing feature dimensionality. This approach has the potential to support automated and efficient fruit quality control processes.

Keywords: Apple, Digital Image, Classification, PCA, SVM

1. Pendahuluan

Apel merupakan salah satu komoditas hortikultura yang dapat hidup di Indonesia, khususnya di dataran tinggi dengan iklim sejuk. Berdasarkan data yang diperoleh dari Badan Pusat Statistik (BPS) produksi tanaman buah Apel pada tahun 2021–2023 berjumlah 1.425.703 ton dengan Jawa Timur sebagai wilayah dengan produksi buah apel terbanyak (Badan Pusat Statistik Indonesia, 2024).

Di Indonesia, tingginya tingkat produksi buah apel sebanding dengan tingkat konsumsinya, yaitu sebesar 1,08 kg per kapita per tahun berdasarkan data BPS pada tahun 2020 (Bawindaputri et al., 2022). Konsumsi buah apel meningkat karena kesadaran masyarakat Indonesia terhadap pola hidup sehat. Namun, tidak semua apel yang dipanen memiliki kualitas dan kondisi yang baik.

Apel merupakan buah yang relatif cepat mengalami pembusukan bila tidak ditangani dengan benar (Sundari et al., 2023). Ketika buah apel yang dipanen berjumlah banyak, sulit untuk mengidentifikasi kondisi apel secara kasat mata. Oleh karena itu, dibutuhkan teknologi untuk mengetahui kualitas apel yang berjumlah banyak dengan efektif dan efisien dengan memilahnya ke dalam kategori segar atau busuk. Dengan demikian, sistem klasifikasi ini dapat dikembangkan sebagai teknologi untuk membantu para petani buah dalam memisahkan buah apel yang segar dan busuk sehingga distribusi buah apel dengan kualitas yang baik dapat dilakukan dan risiko kerugian penjual karena mendistribusikan buah yang rusak dapat diminimalkan.

1.2. Tujuan dan Kegunaan

Proyek ini bertujuan untuk membangun sistem klasifikasi menggunakan citra visual buah apel sehingga didapatkan hasil akhir berupa klasifikasi buah apel yang segar dan busuk dengan akurasi yang tinggi. Klasifikasi dilakukan menggunakan algoritma klasifikasi *Support Vector Machine* (SVM) dengan fitur warna, tekstur, dan bentuk direduksi dimensinya terlebih dahulu dengan menggunakan metode *Principal Component Analysis* (PCA) untuk mendukung otomasi pengawasan kualitas buah apel dengan cepat dan akurat. Seiring dengan perkembangan teknologi, pendekatan terhadap *quality control* kini telah berkembang ke arah otomatisasi berbasis teknologi, termasuk penggunaan *machine learning*, sensor digital, dan pengolahan citra. Hal ini dapat mengurangi kesalahan manusia dan berkontribusi untuk pengembangan *quality control* dalam berbagai bidang kehidupan.

2. Metode Penelitian

2.1. Data Mining

Data mining merupakan proses mengekstraksi pengetahuan dari kumpulan data yang cukup besar. Proses penyaringan data dapat memberikan nilai tambah dengan cara menemukan pola yang tersembunyi. Dalam melakukan analisis data, *data mining* memiliki beberapa algoritma, diantaranya, yaitu K-Means, K-Median, C45, Naive Bayes, Classifier, dan lain–lain (Asyuti & Setyawan, 2023).

Tujuan *data mining* adalah mengembangkan metode tradisional menjadi lebih modern sehingga dapat mengatasi keterbatasan dari metode tradisional, seperti menangani masalah analisis jumlah data yang sangat besar, banyaknya variabel (dimensi) yang terlibat, dan data yang beragam serta berbeda sifat.

2.2. Fruit 360 Dataset

Dataset merupakan sekumpulan data yang tersusun dalam bentuk baris dan kolom secara sistematis yang menjadi dasar dalam analisis data. Terdapat banyak platform yang menyediakan dataset secara daring yang dapat mendukung peneliti untuk menganalisis data. Proses pemilihan, pre-processing, dan pemanfaatan dataset harus dilakukan dengan teliti, serta harus memastikan dataset yang digunakan sesuai dengan tujuan penelitian.

Dataset Fruit-360 adalah kumpulan data citra buah-buahan yang dirancang khusus untuk analisis dan pengelolaan data menggunakan teknik *data mining*, khususnya dalam bidang pengenalan pola dan klasifikasi citra. Dataset ini dibuat oleh Mihai Oltean dan tersedia secara terbuka di platform Kaggle (Oltean, 2017).

Data dikumpulkan dengan cara memutar objek menggunakan motor berkecepatan rendah (3 rpm) selama 20 detik, kemudian direkam menggunakan kamera Logitech C920 dengan latar belakang kertas putih. Untuk mengekstraksi objek dari latar belakang, digunakan algoritma khusus berbasis metode *flood fill*, yang menandai piksel latar belakang berdasarkan jarak warna dan menggantinya dengan warna putih (Oltean, 2017).

2.3. Citra Digital

Citra merupakan representasi visual dari objek yang menggambarkan ciri khusus objek berdasarkan kondisi variabel tertentu (Syarifah et al., 2022). Sedangkan, citra digital adalah sekumpulan titik yang sering disebut

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

dengan piksel. Setiap piksel dalam citra digital direpresentasikan sebagai satu kotak kecil yang memiliki koordinat posisi tertentu. Citra atau *image* dapat didefinisikan sebagai fungsi dua dimensi f (x,y) yang dihasilkan dari alat sensor, yang mencatat nilai atribut citra pada semua titik elemen (x,y) (Marpaung et al., 2022).

Saat ini, citra digital telah menjadi sangat populer dan digunakan secara luas. Berbagai perangkat elektronik dirancang khusus untuk menghasilkan citra digital, seperti kamera digital, *scanner*, mikroskop digital, dan perangkat lainnya. Proses pengolahan citra juga dapat dilakukan melalui perangkat lunak seperti Adobe Photoshop, yang memungkinkan pengguna untuk memperbaiki atau memodifikasi citra agar kualitasnya meningkat. Selain itu, citra digital tersedia dalam berbagai format file, antara lain .jpg, .png, .gif, .mpg, .tif, .ps, .ras, .rgb, dan format lainnya (Marpaung et al., 2022).

2.3.1. Grey Level Co-Occurrence Matrix (GLCM)

Grey Level Co-Occurrence Matrix (GLCM) adalah metode analisis tekstur yang pertama kali dikenalkan oleh Robert Haralick pada tahun 1973 (Muhathir et al., 2021). GLCM merupakan suatu teknik untuk menghitung nilai statistik orde kedua dari sebuah citra digital. Perhitungan nilai statistik tersebut dapat dilakukan dengan cara menghitung probabilitas hubungan kedekatan antara dua buah piksel dalam citra. Hubungan ini dihitung berdasarkan jarak tertentu (d) dan sudut orientasi (θ).

Jarak proses kerja dari metode GLCM dimulai dengan membentuk ko-okurensi pada data citra, yaitu catatan seberapa sering pasangan nilai piksel tertentu muncul secara berdampingan dalam arah dan jarak yang ditentukan. Ko-okurensi merupakan kejadian dari banyaknya satu level pada piksel yang saling bertetanggaan dengan nilai piksel berdasarkan pada jarak dan orientasi sudut. Jarak yang dimaksud merujuk pada seberapa jauh antar piksel, biasanya antara 1 – 10 piksel. Sedangkan orientasi arah pengukuran bisa berupa sudut 0˚,45˚,90˚, dan 135˚(Andono & Rachmawanto, 2021).

Setelah matriks ko-okurensi terbentuk, ada beberapa fitur tekstur yang sering digunakan untuk menganalisis tekstur lebih lanjut, yaitu sebagai berikut (Andono & Rachmawanto, 2021):

1. *Contrast* (Kontras)

   Kontras digunakan untuk mengukur nilai dari tingkat perbedaan keabuan antara piksel dan tetangganya di seluruh citra. Kontras dapat dihitung dengan persamaan (1).

   Rumus:

   $$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(i-j)^2.\ P(i,j) \qquad (1)$$

   Keterangan:

   $i, j =$ tingkat keabuan dari pasangan piksel
   $P(i,j) =$ probabilitas kemunculan pasangan piksel i dan j dalam GLCM
   $(i-j)^2 =$ kuadrat selisih nilai keabuan, semakin besar berarti semakin kontraks

2. *Correlation* (Korelasi)

   Korelasi digunakan untuk mengukur sejauh mana tingkat hubungan linear antara suatu piksel dan tetangganya. Korelasi dapat dihitung dengan persamaan (2).

   Rumus:

   $$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\frac{(i-\mu_i)(j-\mu_j).P(i,j)}{\sigma_i\sigma_j} \qquad (2)$$

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Keterangan:

$\mu_i, \mu_j =$ rata-rata nilai keabuan baris dan kolom GLCM
$\sigma_i \sigma_j =$ standar deviasi baris dan kolom GLCM
$P(i,j) =$ nilai probabilitas dari GLCM

3. *Energy* (Energi)

   Energi digunakan untuk mengukur keteraturan atau keseragaman tekstur keabuan dalam citra. Energi juga dikenal sebagai *Angular Second Moment* (ASM). Energi dapat dihitung dengan persamaan (3).

   Rumus:

$$\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i,j)^2 \qquad (3)$$

   Keterangan:

   $P(i,j) =$ Nilai probabilitas dari GLCM

4. *Homogenity* (Homogenitas)

   Homogenitas digunakan untuk melihat seberapa dekat nilai elemen GLCM pada diagonal utama. Homogenitas dapat dihitung dengan persamaan (4).

   Rumus:

$$\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{P(i,j)}{1+|i-j|} \qquad (4)$$

   Keterangan:

   $P(i,j) =$ nilai piksel ke-i dan ke-j
   $i =$ piksel pada sumbu x
   $j =$ piksel pada sumbu y

2.3.2. Red, Green, Blue (RGB)

RGB terdiri dari 3 elemen warna, yaitu *Red*(R), *Green*(G), dan *Blue*(B). Setiap warna ini memiliki nilai intensitas pada rentang 0 – 225. Misalnya untuk warna merah murni akan dituliskan menjadi [255,0,0], atau dapat dinyatakan dalam bentuk normalisasi, yaitu [1,0,0] (Saputra et al., 2021).

Dalam pengelolaan citra digital, ada rumus perbedaan warna yang paling umum digunakan pada ruang warna RGB, yaitu rumus perbedaan warna berdasarkan jarak (*distance*) dan rumus perbedaan warna sudut (*angle*). Berikut ini adalah langkah-langkah umum dalam proses perhitungan perbedaan warna tersebut (Saputra et al., 2021):

1. Mengkonversi RGB ke dalam bentuk matriks dua dimensi, dimana jumlah baris dan kolom sesuai dengan tinggi dan lebar gambar.

2. Setiap elemen matriks mempresentasikan satu piksel gambar, yang memiliki komponen warna *red, green, blue.*

3. Nilai warna RGB yang diambil pada setiap piksel akan disimpan dalam format 24-bit, dimana masing-masing warna menggunakan 8-bit.

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

4. Jika gambar disimpan dalam format 32-bit, maka akan ada satu komponen tambahan yaitu nilai *alpha* (transparansi). Untuk mengektrak nilai warna dari R, G, B dari 32-bit, maka perlu menggunakan *bitwise operation* (seperti penggeseran bit) untuk memisahkan masing-masing komponen warnanya.

Selain model RGB yang umum, ada juga model normalisasi RGB. Normalisasi RGB adalah model yang menyatakan persentase kontribusi dari masing-masing komponen warna R, G, B terhadap total intensitas warna pada suatu citra digital. Model normalisasi RGB ini bermanfaat untuk mengurangi pengaruh pencahayaan, sehingga akan lebih stabil ketika digunakan dalam proses seperti segmentasi warna atau deteksi objek (Saputra et al., 2021).

Normalisasi RGB dapat dilakukan dengan cara membagi setiap komponen warna dengan jumlah total dari ketiga komponen tersebut. Persamaan normalisasi RGB dapat didefinisikan pada persamaan (5) (Saputra et al., 2021).

$$r = \frac{R}{R+G+B},\ g = \frac{G}{R+G+B},\ b = \frac{B}{R+G+B} \qquad (5)$$

Sehingga, $r + g + b = 1$. Model ini umum digunakan untuk klasifikasi objek berdasarkan warna, seperti deteksi warna kulit, dan lain-lain (Saputra et al., 2021).

2.4. Pre-processing

Pre-processing merupakan tahapan untuk melakukan transformasi awal data pada citra supaya lebih mudah dianalisis dan menghilangkan informasi yang tidak relevan (Shevira et al., 2022). Tahapan *pre-processing* ada beragam tergantung pada kebutuhan, contohnya sebagai berikut (Subandi & Yudhana, 2023):

sebagai berikut (Subandi & Yudhana, 2023):

1. *Cleaning* data merupakan proses menghapus data yang tidak konsisten, tidak lengkap, ataupun yang mengandung *noise* (Bahtiar, 2023).

2. *Resizing* mengandung teknik mengubah ukuran citra ke ukuran yang diinginkan supaya semua citra memiliki ukuran yang sama.

3. Konversi citra adalah tahapan dimana jika dataset yang diperoleh dalam bentuk RGB, maka citra akan mengkonversikannya kedalam bentuk *grayscale*.

4. Normalisasi adalah transformasi data supaya data tetap berada dalam skala tertentu.

2.5. Ekstraksi Fitur

Ekstraksi fitur adalah tahapan dalam pengelolaan citra yang memiliki tujuan untuk memperoleh informasi mengenai ciri dan karakteristik tertentu dalam sebuah citra. Untuk mendapatkan informasi yang penting dari suatu citra, sangat penting untuk melalui proses ekstraksi fitur. Jika peneliti telah mendapatkan informasi penting dari suatu citra, maka deskripsi dan interpretasi mengenai suatu objek yang akurat akan didapat (Haerya, 2025). Beberapa fitur yang dapat diekstraksi dari sebuah citra antara lain, fitur warna, tekstur, dan bentuk, sebagai berikut:

1. Fitur warna, merupakan informasi yang memberikan pengetahuan tentang distribusi warna dalam suatu citra sehingga peneliti dapat membedakan objek berdasarkan karakteristik warnanya. Ekstraksi fitur warna dapat dilakukan dengan cara menghitung nilai rata-rata, variansi, atau histogram dari masing-masing kanal warna RGB (Haerya, 2025).

2. Fitur tekstur, merupakan informasi yang memberikan pengetahuan tentang permukaan objek sebuah citra secara visual. Ekstraksi fitur dapat diaplikasikan dalam berbagai penelitian seperti *image retrieval, image classification, image processing*, dan lainnya (Mustofa et al., 2022).

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

3. Fitur bentuk, merupakan informasi yang memberikan pengetahuan tentang karakteristik geometris dari sebuah objek dalam citra. Ekstraksi fitur bentuk ini penting untuk membedakan objek yang memiliki warna atau tekstur yang serupa tapi bentuk yang berbeda (Haerya, 2025).

2.6. Mahalanobis Distance

Mahalanobis Distance adalah perhitungan jarak antara sebuah variabel dengan pusat dari semua pengamatan, yang mempertimbangkan korelasi antar variabel serta variansi dari sebuah data. *Mahalanobis Distance* selain digunakan untuk analisis klaster, klasifikasi, dan *Principal Component Analysis* (PCA), juga digunakan untuk mendeteksi *outlier* (Anggoro et al., 2016).

Metode ini memungkinkan untuk melakukan perhitungan jarak yang akurat terutama ketika data memiliki korelasi antar fitur atau perbedaan skala di antara fitur-fitur tersebut. *Mahalanobis Distance* merupakan ukuran jarak antara dua vektor yang memperhitungkan korelasi dalam kumpulan data (Nurhakim et al., 2024). Relasi antara kedua vektor ini dapat didapat melalui persamaan (6) dibawah ini (Kantacarini et al., 2021)

$$D^2 = (x-\mu)^T S^{-1}(x-\mu) \qquad (6)$$

Keterangan:

x = vektor data
μ = vektor mean dari data
T = transpose
S = matriks kovarians dari data
S^{-1} = invers dari matriks kovarians

2.7. Outlier

Outlier merupakan data pengamatan yang menyimpang atau tidak konsisten pada deret waktunya. Secara umum ada 4 macam jenis *outlier*, yaitu sebagai berikut (Sihombing et al., 2022; Badan Pusat Statistik Indonesia, 2024):

1. *Innovational Outlier* (IO) adalah jenis *outlier* yang menyebabkan gangguan atau kejadian dengan dampak mengikuti pola proses ARMA (*Autoregressive Moving Average*).

2. *Additive Outlier* (AO) adalah jenis *outlier* yang hanya memberikan pengaruh pada data deret waktu di satu periode tertentu saja.

3. *Temporary Change* (TC) adalah suatu kejadian dimana *outlier* menghasilkan dampak awal sebesar ω pada waktu tertentu, kemudian secara bertahap sesuai dengan besarnya δ.

4. *Level Shift* (LS) adalah kejadian yang menimbulkan perubahan mendadak dan bersifat permanen dalam deret waktu pada titik waktu tertentu.

Salah satu penyebab munculnya *outlier* adalah kesalahan pada saat proses pengambilan data, seperti kesalahan input data ataupun pencatatan yang tidak akurat. Hal ini dapat menyebabkan nilai data menjadi tidak sesuai dengan pola umum data lainnya. Untuk mendeteksi keberadaan outlier dalam suatu pengamatan, peneliti harus memiliki asumsi awal untuk menduga mengenai variabel mana yang mungkin mengandung *outlier* (Sihombing et al., 2022).

2.8. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) adalah salah satu metode statistik yang digunakan untuk mereduksi dimensi data menjadi sejumlah kecil komponen utama sambil mempertahankan informasi yang ada sebanyak mungkin (Aryasatya et al., 2024).

Principal Component Analysis (PCA) bertujuan untuk menyederhanakan data dengan cara mengurangi kompleksitas serta meminimalkan risiko *overfitting* akibat banyaknya variabel bebas dengan mengurangi dimensi

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

menjadi beberapa komponen utama yang mencakup keseluruhan informasi (Anggraini, 2022). Hal ini dapat dilakukan dengan cara mentransformasi variabel bebas asli menjadi kumpulan variabel baru yang saling tidak berkorelasi, atau yang biasa disebut dengan *principal components*. Setelah itu, PCA akan menghasilkan komponen-komponen yang bebas dari multikolineritas yang dapat digunakan sebagai variabel bebas baru untuk melakukan sebuah analisis, misalnya analisis dengan metode regresi atau metode lainnya, untuk mengetahui pengaruhnya terhadap variabel terikat (Rianti et al., 2024).

Berikut ini merupakan langkah perhitungan PCA (Hediyati & Suartana, 2021):

1. Menghitung *mean* ($\bar{X}$) dari data pada setiap dimensi menggunakan persamaan (7) dan mengurangi setiap variabel nya dengan mean yang telah dihitung:

$$\bar{X}_j = \frac{1}{m}\sum_{i=1}^{m} X_{ij} \qquad (7)$$

Keterangan:

$\bar{X}_j$ = mean kolom ke-j
m = jumlah total data
X_i = data sampel
X_{ij} = nilai data baris ke-1, dan kolom ke-j

2. Menghitung *covariance matrix* menggunakan persamaan (8):

$$C_X = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})^T \qquad (8)$$

Keterangan:

C_X = covarians matrix
n = jumlah total data
X_i = data sampel
$\bar{X}$ = mean

3. Menghitung nilai *eigen* dan vektor *eigen* dari covariance matrix dengan menggunakan persamaan (9) lalu urutkan nilai *eigen* dan vector *eigen* secara *descending* (besar ke kecil).

$$C_X v_m = \lambda_m v_m \qquad (9)$$

Keterangan:

v_m = eigen vektor
λ_m = eigen value
C_X = covarians matrix

4. Pemilihan prinsipal komponen (PK), PK merupakan deretan vektor *eigen* yang sesuai dengan urutan nilai *eigen* pada tahap 3

5. Menghasilkan dataset baru

2.9. Support Vector Machine (SVM)

Boser, Guyon, dan Vapnik adalah orang yang mengembangkan salah satu algoritma klasifikasi, yaitu *Support Vector Machine* (SVM) (Yanti & Asana, 2023). Metode SVM digunakan untuk mengklasifikasikan data yang memiliki karakteristik berbeda ke dalam beberapa kelas. Inti dari SVM adalah mencari *hyperplane* optimal (Rabbani et al., 2023). *Hyperplane* memiliki fungsi sebagai pemisah di antara dua kelas. Untuk menemukan *hyperplane*, dapat dilakukan dengan cara mengukur margin *hyperplane*, lalu mencari titik maksimalnya. Margin

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

merupakan jarak antara *hyperplane* dengan titik data terdekat dari masing-masing kelas. Titik-titik data tersebut sering disebut dengan vektor pendukung atau *support vector* (Haqimi & Roshinta, 2024).

SVM memiliki dua jenis *hyperplane*, yaitu *hyperplane* linear, dan *hyperplane* non-linear. Disebut *hyperplane* linear jika data dapat dipisahkan secara lurus atau linear. Disebut *hyperplane* non-linear jika data tidak dapat dipisahkan secara linear. Untuk menangani data yang tidak bisa dipisahkan secara lurus, SVM akan menggunakan fungsi kernel (Haqimi & Roshinta, 2024). Fungsi kernel adalah fungsi pemetaan implisit yang mentransfromasikan data dari ruang berdimensi rendah ke ruang berdimensi lebih tinggi, hal ini dilakukan supaya data yang awalnya sulit dipisakan secara linear menjadi lebih mudah dipisahkan di ruang yang baru, tanpa perlu melakukan perhitungan transformasi secara eksplisit lagi (Arifin et al., 2023). Dalam konteks ini ini, rumus umum untuk SVM linear dapat dinyatakan pada persamaan (10) dibawah ini.

$$f(x) = sign\ (w.x + b) \qquad (10)$$

Keterangan:

$f(x) =$ fungsi prediksi
$w =$ vektor normal hyperplane
$x =$ vektor fitur input
$b =$ bias atau intercept

Dengan memanfaatkan berbagai jenis fungsi kernel, SVM mampu untuk menangani data yang memiliki sifat non-linear secara efektif. Seperti kernel linear, kernel polinomial atau kernel *Gaussian*, dan kernel *Radial Basis Function* (RBF) yang dijelaskan sebagai berikut (Rabbani et al., 2023):

1. Kernel Linear

   Kernel linear mengukur produk titik (*dot product*) antara dua vektor input di ruang asli tanpa melakukan pemetaan ke ruang fitur yang berdimensi lebih tinggi. Pengukuran ini dapat dilakukan dengan menggunakan persamaan (11) dibawah ini.

   $$K(x, y) = (x, y) \qquad (11)$$

   Keterangan:

   $x, y =$ vektor input

2. Kernel Polinomial

   Kernel polinomial mengevaluasi hubungan polinomial antara dua vektor input dalam ruang asli. Hubungan ini dapat dihitung dengan menggunakan persamaan (12) dibawah ini.

   $$K(x, y) = (x, y + c)^d \qquad (12)$$

   Keterangan:

   $x, y =$ vektor input
   $c =$ koefisien bebas
   $d =$ derajat polinomial

3. Kernel *Radial Basis Function* (RBF)

   Kernel *Radial Basis Function* (RBF) menggunakan fungsi *Gaussian* yang dikenal sebagai fungsi basis radial, untuk menilai tingkat kemiripan antara dua vektor input dalam ruang fitur. Perhitungan kesamaan ini, dapat menggunakan persamaan (13) dibawah ini.

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

$$K(x, x') = \exp\left(-\gamma \left|\left|x_i - x_j\right|\right|^2\right) \quad (13)$$

Keterangan:

x_i, x_j = dua vektor input
y = parameter yang mengatur jarak pengaruh dari satu titik data
$\left|x_i - x_j\right|^2$ = jarak euclidean kuadrat antar dua vektor

Parameter yang mengatur jarak pengaruh dari satu titik data digambarkan dengan persamaan (14). Selain itu, beberapa rumus juga akan digunakan dalam proses klasifikasi SVM non-linear menggunakan kernel RBF yang dijelaskan sebagai berikut(Cristianini & Shawe-Taylor, 2000).

$$\gamma = \frac{1}{2\sigma^2} \quad (14)$$

Keterangan:

σ = skala kernel Gaussian (ditentukan secara mandiri dengan rentang umum 0.5 – 5)

Selanjutnya proses pelatihan dilanjutkan dengan mencari bobot α_i setiap data latih untuk menentukan data mana yang akan menjadi *support vector* untuk membentuk garis *hyperplane*. Rumus penentuan tersebut dapat dilihat pada persamaan (15) dengan kenfsls yang dapat dilihat pada persamaan (16).

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_i) \quad (15)$$

Dengan kendala:

$$\sum_{i=1}^{n} \alpha_i y_j = 0, \quad 0 \leq \alpha_i \leq C \quad (16)$$

Keterangan:

α_i = bobot untuk setiap data pelatihan
y_i = label kelas pada data ke-i
$K(x_i, x_j)$ = fungsi kernel
C = box constraint dengan skala yang ditentukan dengan rentang tertentu

Setelah nilai α_i didapatkan, selanjutnya model SVM dapat dilatih supaya dapat menentukan kelas dari data baru yang belum pernah dilihat dengan cara menjumlahkan pengaruh semua data penting terhadap data baru x, lalu ditambah nilai bias b. Hal ini dilakukan dengan fungsi keputusan yang dapat dilihat pada persamaan (17).

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \quad (17)$$

Keterangan:

$f(x)$ = skor klasifikasi input x (+1 Jika positif dan -1 jika negatif)
x = vektor fitur data baru yang ingin diklasifikasikan
b = bias dari hyperplane

Perhitungan b dibutuhkan dalam hal ini dengan tujuan agar garis pemisah tidak bergeser terlalu jauh ke satu sisi. Nilai b dapat dihitung dengan mengambil salah satu data yang menjadi *support vector*, yaitu data yang punya bobot $0 \leq \alpha \leq C$. Perhitungan nilai b dapat dilihat pada persamaan (18).

$$b = y_k - \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) \quad (18)$$

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Keterangan:

x_k = support vector dengan $0 < \alpha_k < C$
y_k = label sebenarnya dari support vector x_k

Metode SVM merupakan salah satu metode yang popular dan banyak digunakan pada penelitian yang menggunakan citra digital sebagai data yang ingin diklasifikasikan. Salah satunya pada penelitian yang dilakukan oleh Yusuf Amrozi dkk pada tahun 2022 yang mengklasifikan jenis pisang menggunakan fitur warna, tekstur, dan bentuk menggunakan metode klasifikasi SVM dengan fitur yang sebelumnya sudah direduksi terlebih dahulu dengan menggunakan metode reduksi dimensi PCA. Hasil dari klasifikasi yang dilakukan Yusuf Amrozi dkk menghasilkan akurasi klasifikasi sebesar 89,86% dari total data citra yang digunakan sebanyak 1.256 citra. Dari hasil penelitian ini, dapat dikatakan bahwa SVM merupakan metode klasifikasi yang tepat untuk mengklasifikasikan citra digital terutama pada jenis buah, tidak hanya pisang saja, buah apel juga dapat diklasifikasikan kondisinya apakah "bagus" atau "jelek" dengan menggunakan fitur warna, tekstur, dan bentuk, dengan metode reduksi PCA dan klasifikasi SVM yang akan dilakukan dalam penelitian ini (Amrozi et al., 2022).

2.10. Confusion Matrix

Confusion Matrix adalah salah satu teknik evaluasi klasifikasi suatu model. Cara kerja *confusion matrix* adalah dengan cara membandingkan matriks dari prediksi dengan kelas asli serta prediksi nilai klasifikasi (Hasibuan et al., 2024). *Confusion Matrix* merupakan tabel yang teridiri 4 kombinasi berbeda dari *predicted values* dan *actual values*, yaitu (Wafa et al., 2022):

1. *True Negative* (TN) dikategorikan sebagai data yang sebenarnya bersifat negatif terdeteksi benar.
2. *True Positive* (TP) dikategorikan sebagai data yang sebenarnya positif terdeteksi benar.
3. *False Negative* (FN) dikategorikan sebagai data yang sebenarnya positif, namun terdeteksi sebagai data negatif.
4. *False Positive* (FP) dikategorikan sebagai data yang sebenarnya negatif, tapi terdeteksi sebagai data positif.

Dari proses klasifikasi pada *confusion matrix*, dapat dihitung tiap nilainya dengan *accuracy*, *recision*, *recall*, dan *F1-score*. Berikut ini merupakan penjelasan dari tiap perhitungannya (Wafa et al., 2022):

1. *Accuracy*, merupakan gambaran tingkat keakuratan model yang digunakan dalam klasifikasi. Rumus perhitungan akurasi dapat dilihat pada persamaan (19).

$$\left(\frac{(TP+TN)}{(TP+FP+TN+FN)}\right) \text{ X } 100\% \qquad (19)$$

2. *Precision*, merupakan gambaran seberapa tepat prediksi positif yang dihasilkan model dibandingkan dengan jumlah total prediski positif yang dibuat. Rumus perhitungan presisi dapat dilihat pada persamaan (20).

$$\left(\frac{(TP)}{(TP+FP)}\right) \text{ X } 100\% \qquad (20)$$

3. *Recall*, merupakan gamabran sejauh mana sebuah model berhasil menemukan kembali seluruh data yang benar-benar positif. Rumus perhitungan *recall* dapat dilihat pada persamaan (21).

$$\left(\frac{(TP)}{(TP+FN)}\right) \text{ X } 100\% \qquad (21)$$

4. *F1-score*, merupakan gambaran perbandingan rata-rata harmonik dari *precision* dan *recall* yang telah di bobotkan. Rumus perhitungan *f1-score* dapat dilihat pada persamaan (22).

$$\left(\frac{2\ x\ precision\ x\ recall}{(precision+recall)}\right) \text{ X } 100\% \qquad (22)$$

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

2.11. Skenario Pengujian

Pada penelitian ini, dilakukan serangkaian pengujian guna mengevaluasi efektivitas metode yang diterapkan. Pengujian dilakukan secara bertahap, dimulai dari tahap *pre-processing* gambar, pendeteksian *missing value*, penghapusan terhadap *outlier* jika ada, reduksi dimensi menggunakan PCA, hingga klasifikasi menggunakan SVM. Setiap tahap memiliki peran penting dalam memastikan data yang digunakan optimal dan model yang dihasilkan memiliki performa klasifikasi yang baik. Berikut merupakan tabel skenario pengujiannya:

Tabel 1. Skenario Pengujian dengan Input

No	Skenario	Input
1	Preprocessing terhadap gambar apel bagus dan jelek untuk menghasilkan dataset *grayscale* dan ekstraksi fitur.	Dataset gambar apel bagus dan jelek.
2	Pengecekan *missing value* dan penghapusan terhadap *outlier* dengan *Mahalanobis Distance Mean*	Dataset hasil ekstraksi fitur dari gambar yang telah diproses.
3	Pemisahan data menjadi data *train* dan data *test* dengan proporsi 80:20 setelah *outlier* dihapus	Dataset bersih hasil *preprocessing* dan deteksi *outlier*.
4	Transformasi serta analisis PCA terhadap data *train*, serta menyesuaikan data *test* berdasarkan parameter *train*	Data *train* dan *test* bersih hasil *split* dari *preprocessing* (termasuk deteksi *outlier*) dengan Proporsi Kumulatif 0.90 atau 90%.
5	Pelatihan dan pengujian model SVM terhadap data *train* dan *test* yang sudah direduksi dimensinya.	Data *train* dan *test* hasil PCA dengan label yang sesuai.
6	Evaluasi performa model SVM berdasarkan akurasi, *precision*, *recall*, dan *confusion matrix*	Hasil prediksi dari model SVM terhadap data *test*.

Tabel 1 merupakan skenario pengujian yang dirancang untuk mengevaluasi apakah penerapan PCA dan SVM dalam proses klasifikasi kualitas apel berdasarkan ciri bentuk, warna, dan tekstur sudah baik. Skenario pertama dilakukan dengan melakukan *pre-processing* pada gambar apel bagus dan jelek untuk menghasilkan data dalam bentuk *grayscale* serta melakukan ekstraksi fitur guna membentuk dataset awal. Skenario kedua bertujuan untuk membersihkan dataset dari *outlier* maupun *missing value*. Untuk pendeteksian dan penghapusan *outlier* dilakukan dengan menggunakan metode Mahalanobis Distance Mean agar data lebih representatif untuk analisis lebih lanjut. Skenario ketiga dilakukan dengan membagi dataset yang telah bersih menjadi data *training* dan data *testing* dengan proporsi 80:20. Selanjutnya, pada skenario keempat, dilakukan reduksi dimensi menggunakan PCA terhadap data *train*, kemudian data *test* disesuaikan transformasinya menggunakan parameter PCA dari *train*, dengan tujuan mempertahankan proporsi variansi kumulatif sebesar 90%. Skenario kelima adalah pelatihan dan pengujian model klasifikasi SVM menggunakan data yang telah dibagi. Terakhir, skenario keenam difokuskan pada evaluasi performa model SVM berdasarkan metrik akurasi, *precision*, *recall*, serta *confusion matrix* untuk mengukur seberapa baik model dalam mengklasifikasikan kualitas apel.

3. Hasil dan Pembahasan

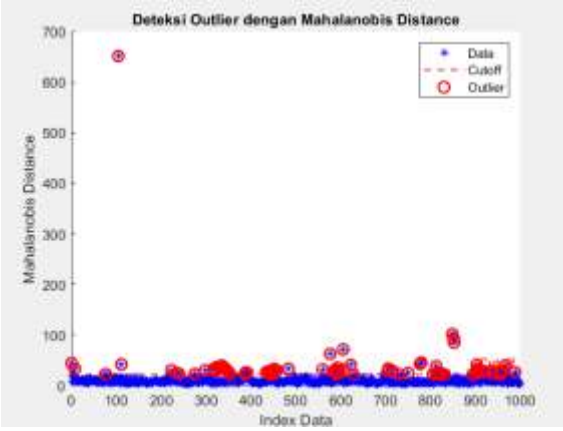
Rangkaian

3.1. Hasil Pre-processing

Tahap *pre-processing* dimulai dengan konversi gambar berwarna ke *grayscale*, diikuti ekstraksi fitur warna, tekstur, dan bentuk. Selanjutnya, dilakukan deteksi *missing value* dan penghapusan *outlier* menggunakan *Mahalanobis Distance Mean*, dan diakhiri dengan pembagian data *train* dan *test* dengan proporsi 80:20. Dikarenakan data tidak memiliki *missing value*, proses dapat dilanjut langsung ke pendeteksian *outlier* menggunakan metode *Mahalanobis Distance Mean.* Metode ini digunakan karena lebih sensitif dan akurat

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

dibandingkan metode *Euclidean Distance*. Prosesnya dimulai dengan menghitung *mean* dan matriks kovarians tiap variabel, dilanjutkan dengan perhitungan jarak *Mahalanobis* tiap observasi dan dilanjut dengan menentukan *cutoff* untuk menetapkan data *outlier*. Pada hasil pengujian dataset mentah, didapat bahwa nilai *cutoff* bernilai 21,4004. Hal ini, memberikan kesimpulan bahwa *Mahalanobis Distance* menganggap sebuah observasi adalah *outlier* ketika melewati *cutoff*.



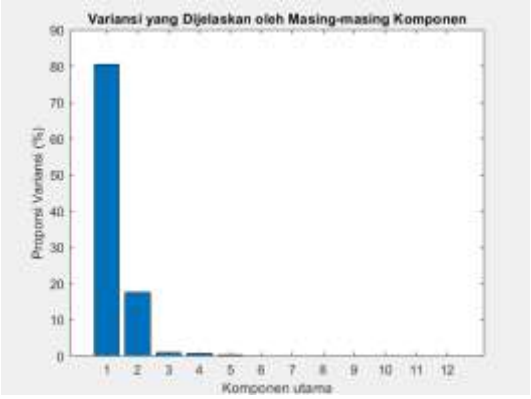
Gambar 1. Visualisasi Outlier dengan Mahalanobis Distance

Gambar 1 menunjukkan titik observasi yang dianggap *outlier* ditandai dengan lingkaran merah. Perlu diperhatikan bahwa garis putus-putus merupakan batas *cutoff* yang harus dihitung terlebih dahulu. Hal ini menunjukkan bahwa pendekatan *Mahalanobis Distance Mean* efektif dalam mengidentifikasi data yang secara statistik berbeda dari populasi utama, dan dapat mempengaruhi analisis jika tidak ditangani.

Setelah data bersih dari *outlier*, tahap terakhir dalam *pre-processing* adalah melakukan pembagian data menjadi data *train* dan data *test*. Pada penelitian ini digunakan rasio 80:20, di mana 80% data digunakan untuk pelatihan dan 20% sisanya digunakan untuk pengujian. Proses pembagian ini dilakukan secara acak namun tetap mempertahankan proporsi antara dua kelas apel agar tetap seimbang. Pembagian ini menjadi penting untuk memastikan bahwa model yang akan dilatih dan diuji memiliki distribusi data yang adil dan representatif.

3.2. Hasil PCA

Proses selanjutnya adalah melakukan reduksi dimensi menggunakan *Principal Component Analysis* (PCA). Proses PCA dilakukan hanya pada data *train* untuk menghitung parameter seperti rata-rata, kovarians, *eigenvalue*, dan *eigenvector*. Setelah itu, data *test* ditransformasikan ke ruang yang sama menggunakan parameter tersebut agar tidak terjadi *data leakage*. Berikut adalah hasil visualisasi dari proporsi kumulatif dan nilai proporsi pada setiap komponen utama atau PC.



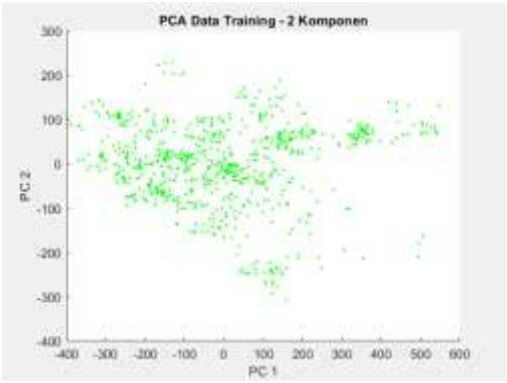
Gambar 2. Visualisasi Barplot Terhadap Proporsi Variansi

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Tabel 2. Nilai Proporsi Kumulatif Komponen Utama

Komponen Utama	Nilai Proporsi Variansi (%)	Proporsi Kumulatif (%)
PC1	80.4502	80.4502
PC2	17.6485	98.0988

Terlihat pada Gambar 2 dan Tabel 2, bahwa proporsi kumulatif PC2 mencapai 98.0988% dari total variansi data. Hal ini menunjukkan bahwa sebagian besar informasi yang terkandung dalam fitur awal sudah dapat direpresentasikan hanya dengan dua komponen utama saja. Dengan kata lain, lebih dari 90% informasi penting dari data berhasil dipertahankan, sementara jumlah fitur berhasil dikurangi secara signifikan. Lalu, berikut merupakan grafik 2D antar PC1 dan PC2.

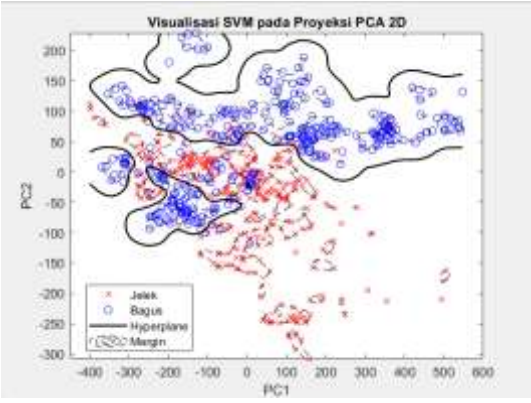


Gambar 3. Grafik PCA dari PC1 dan PC2

Terlihat pada Gambar 3, bahwa grafik menunjukkan proyeksi data *train* ke dalam dua dimensi utama hasil ekstraksi PCA, yaitu PC1 dan PC2. Setiap titik merepresentasikan satu sampel data yang telah ditransformasikan dari ruang fitur asli ke ruang PCA berdasarkan arah maksimum variansi. Penyebaran data yang cukup tersebar di sepanjang kedua sumbu menunjukkan bahwa komponen ini berhasil menangkap informasi dominan dari dataset, dengan PC1 sebagai komponen dengan varian terbesar dan PC2 sebagai ortogonalnya.

3.3. Hasil SVM

Setelah dilakukan reduksi dimensi menggunakan metode PCA, tahap selanjutnya dalam proses klasifikasi kualitas apel adalah membangun dan melatih model SVM. Komponen utama yang terbentuk dari PCA adalah dua (PC1 dan PC2). Kedua komponen ini dipilih karena secara kumulatif mampu merepresentasikan sebagian besar informasi penting dalam dataset, sehingga cukup representatif untuk digunakan dalam klasifikasi sekaligus memungkinkan visualisasi dua dimensi. Model SVM yang dibangun kemudian dievaluasi dari tiga aspek utama, yaitu visualisasi hasil klasifikasi, analisis *confusion matrix* untuk mengetahui distribusi kesalahan klasifikasi, serta evaluasi metrik seperti akurasi, precision, recall, dan F1-score baik pada data *train* maupun data *test*. Berikut adalah hasil visualisasi SVM yang telah didapat.



Gambar 4. Visualisasi 2 Dimensi Klasifikasi SVM

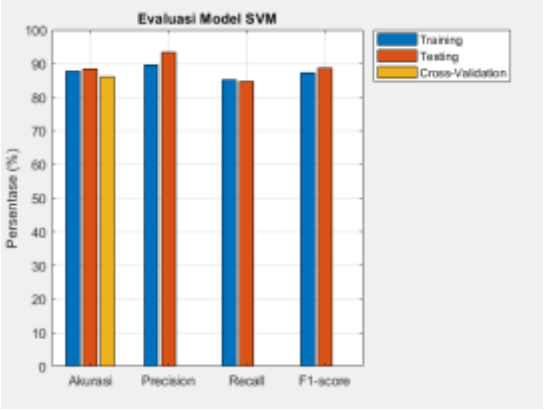
DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Gambar 4 menyajikan visualisasi dari hasil klasifikasi model SVM yang ditampilkan dalam ruang dua dimensi hasil proyeksi PCA, yaitu komponen utama pertama (PC1) dan kedua (PC2). Titik-titik data yang berwarna merah berbentuk silang (x) menunjukkan sampel apel dengan label kualitas jelek, sedangkan titik-titik biru berbentuk lingkaran menunjukkan sampel apel dengan kualitas bagus. Terlihat juga, grafik menampilkan *hyperplane* sebagai garis hitam padat yang berfungsi sebagai batas pemisah antara dua kelas. SVM mencari *hyperplane* sedemikian rupa agar jarak ke titik data terdekat dari masing-masing kelas (*margin*) adalah maksimum. Garis hitam putus-putus di sekitar *hyperplane* menunjukkan margin atau ambang batas toleransi klasifikasi. Dari distribusi data dapat diamati bahwa sebagian besar titik dari kedua kelas berhasil dipisahkan oleh *hyperplane* dengan baik, meskipun terdapat beberapa area di mana data dari kelas berbeda saling tumpang tindih. Hal ini merupakan indikasi bahwa meskipun SVM cukup efektif dalam memisahkan kelas, namun terdapat bagian dari distribusi fitur yang memang bersifat ambigu atau tumpang tindih secara alami dalam ruang fitur tersebut. Visualisasi ini sangat berguna untuk membantu memahami distribusi data setelah proses PCA dan memberikan gambaran umum mengenai efektivitas klasifikasi oleh model SVM. Setelah didapat visualisasi 2 dimensi, maka bisa dilanjutkan untuk mengetahui *confusion matrixnya*.



Gambar 5. Confusion Matrix pada Data Train dan Data Test

Dapat dilihat pada Gambar 5, evaluasi performa model juga dilakukan melalui *confusion matrix* untuk data *train* dan *test*. Pada data *train*, model berhasil mengklasifikasikan 304 data apel jelek dan 332 data apel bagus dengan benar. Sementara itu, sebanyak 35 apel jelek diklasifikasikan salah sebagai bagus dan 53 apel bagus diklasifikasikan salah sebagai jelek. Ini menunjukkan bahwa meskipun performa model cukup baik, terdapat beberapa kesalahan terutama pada pengenalan apel bagus. Sedangkan pada *confusion matrix* untuk data *test*, performa model tetap stabil dengan hasil klasifikasi 77 apel jelek sebagai jelek dan 83 apel bagus sebagai bagus. Sementara itu, terdapat juga kesalahan klasifikasi berjumlah 21, yang masih tergolong rendah. Hal ini menunjukkan bahwa model tidak mengalami *overfitting* secara signifikan, karena perbedaan performa antara data *train* dan *test* relatif kecil. Hasil ini menunjukkan bahwa kombinasi antara PCA dan SVM berhasil digunakan dalam mengelompokkan kualitas buah apel ke dalam dua kategori, yaitu bagus dan jelek. Dikarenakan *confusion matrix* sudah didapat, maka selanjutnya adalah memberikan visualisasi terkait akurasi dari model.



Gambar 6. Visualisasi Evaluasi Model SVM

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Tabel 3. Hasil Evaluasi Model SVM

	Akurasi (%)	Precision (%)	Recall (%)	F1-Score (%)
Training	87.85	90.00	85.00	87.00
Testing	88.40	93.00	85.00	89.00
Cross-Validation	86.80	-	-	-

Terlihat pada Gambar 6 dan Tabel 3, bahwa model SVM menunjukkan performa yang konsisten antara data *training* dan data *testing*. Akurasi pada data training mencapai 87.85%, sedangkan pada data testing bahkan sedikit meningkat menjadi 88.40%, menunjukkan bahwa model tidak mengalami *overfitting*. Hal ini diperkuat oleh hasil *cross-validation* sebesar 86.80%, yang mengindikasikan bahwa model memiliki kemampuan generalisasi yang stabil terhadap variasi data. Lalu, untuk Precision pada data *testing* mencapai 93.00%, lebih tinggi dibandingkan precision pada data *training* yang sebesar 90.00%. Ini menunjukkan bahwa proporsi prediksi kelas "Bagus" yang benar relatif lebih tinggi pada data yang belum pernah dilihat model sebelumnya. Sementara itu, *Recall* pada data *training* dan *testing* masing-masing adalah 85.00% dan 85.00%, yang berarti sebagian besar data apel yang benar-benar bagus berhasil dikenali oleh model. F1-Score sebagai matrix harmonik dari precision dan recall menunjukkan nilai 87.00% pada *training* dan 89.00% pada *testing*. Hasil ini mencerminkan keseimbangan yang

4. Kesimpulan

Secara garis besar, penelitian ini bertujuan untuk mengklasifikasikan kualitas buah apel ke dalam dua kelas, yaitu apel bagus dan apel jelek yang didasarkan dari fitur visual seperti bentuk, warna, dan tekstur menggunakan kombinasi metode *Principal Component Analysis* (PCA) dan *Support Vector Machine* (SVM). Berdasarkan seluruh proses yang telah dilakukan, maka didapat beberapa kesimpulan sebagai berikut: 1). Metode PCA dan SVM berhasil diterapkan untuk mengklasifikasikan kualitas apel secara efektif melalui serangkaian tahapan yang dimulai dari *pre-processing*, deteksi *missing value* maupun *outlier*, pembagian data, reduksi dimensi, hingga proses klasifikasi model, 2). Hasil reduksi dimensi menggunakan PCA menunjukkan bahwa dua komponen utama (PC1 dan PC2) dapat merepresentasikan 98,09% variansi data, yang berarti sebagian besar informasi penting dari fitur awal tetap terjaga dalam bentuk yang lebih sederhan, 3). Model SVM memberikan performa klasifikasi yang baik dengan akurasi 87,85% pada data *training*, dan 88,40% pada data *testing*. Metrik evaluasi seperti *precision*, *recall*, dan *F1-score* yang berada di kisaran 85% hingga 93%. Evaluasi menggunakan *cross-validation* menghasilkan nilai akurasi sebesar 86,80% yang menunjukkan kemampuan generalisasi model terhadap data baru. Kelebihan penelitian ini ialah proses yang efisien dan cocok untuk klasifikasi berbasis citra, sedangkan kekurangan penelitian ini ialah ukuran dataset yang masih terbatas. Pengembangan lebih lanjut dapat dilakukan dengan menambah jumlah dan variasi data, lalu membandingkan metode ini dengan metode yang lain seperti metode yang menggunakan *deep learning*.

References

1. Andono, P. N., & Rachmawanto, E. H. (2021). Evaluasi ekstraksi fitur GLCM dan LBP menggunakan multikernel SVM untuk klasifikasi batik. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 5*(1), 1-9.
2. Anggoro, A., Mustafid, M., & Rahmawati, R. (2016). Pengendalian kualitas data atribut multivariat dengan Mahalanobis Distance dan T2 Hotelling (studi kasus PT Metec Semarang). *Jurnal Gaussian, 5*(3), 311-320.
3. Anggraini, M. P. (2022). *Analisis PCA (Principal Component Analysis) pada sidik jari kromatografi lapis tipis tumbuhan anting-anting (Acalypha indica L.) dari daerah Banyuwangi berdasarkan berbagai metode pengeringan* [Thesis, Universitas Islam Negeri Maulana Malik Ibrahim].
4. Amrozi, Y., Yuliati, D., Susilo, A., Novianto, N., & Ramadhan, R. (2022). Klasifikasi jenis buah pisang berdasarkan citra warna dengan metode SVM. *Jurnal Sisfokom (Sistem Informasi dan Komputer), 11*(3), 394-399. https://doi.org/10.32736/sisfokom.v11i3.1502
5. Arifin, Z., Rahman, D. F., Rintyarna, B. S., & Daryanto, D. (2023). Penerapan algoritma Support Vector Machine berbasis kernel Radial Basis Function dalam klasifikasi sel kanker. *BIOS: Jurnal Teknologi Informasi dan Rekayasa Komputer, 4*(2), 100-106.
6. Aryasatya, F., Katrina, A., Syabila, R. F., & Siregar, F. (2024). Identifikasi faktor-faktor esensial dalam hasil evaluasi siswa menggunakan teknik Principal Component Analysis (PCA). *Innovative: Journal of Social Science Research, 4*(6), 6423-6437.
7. Asyuti, S., & Setyawan, A. A. (2023). Data mining dalam penggunaan presensi karyawan denga cluster means. *Jurnal Ilmiah Sains Teknologi dan Informasi, 1*(1), 1-10.
8. Badan Pusat Statistik Indonesia. (2024, June 10). *Produksi tanaman buah-buahan, 2021-2023*. https://www.bps.go.id/id/statistics-table/2/NjIjMg==/produksi-tanaman-buah-buahan.html
9. Bahtiar, R. (2023). Implementasi data mining untuk prediksi penjualan kusen terlaris menggunakan metode K-Nearest Neighbor. *Jurnal Informatika MULTI, 1*(3), 203-214.

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

10. Bawindaputri, T. A., Astuti, R., & Lestari, E. R. (2022). Analisis keberlanjutan rantai pasok apel di Kabupaten Malang menggunakan dinamika sistem. *Jurnal Teknologi Industri Pertanian, 32*(2), 121-136.
11. Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.
12. Haerya, P. (2025). *Implementasi ekstraksi fitur Gray Level Co-Occurrence Matrix dan Hue Saturation Value untuk deteksi penyakit daun padi* [Thesis, Universitas Sulawesi Barat].
13. Haqimi, N. A., & Roshinta, T. A. (2024). Analisis spam komentar Instagram menggunakan Support Vector Machine dengan variasi hyperparameter. *Jurnal Informatika: Jurnal Pengembangan IT, 9*(3), 242-253.
14. Hasibuan, R. A., Ratnawati, D. E., & Perdana, R. S. (2024). Analisis sentimen kebijakan ekspor pasir laut pada sosial media Twitter menggunakan algoritma Support Vector Machine. *Jurnal Sistem Informasi, Teknologi Informasi, dan Edukasi Sistem Informasi (JUST-SI), 5*(1), 24-33. https://doi.org/10.25126/justsi.v5i1.373
15. Hediyati, D., & Suartana, I. M. (2021). Penerapan Principal Component Analysis (PCA) untuk reduksi dimensi pada proses clustering data produksi pertanian di Kabupaten Bojonegoro. *JIEET (Journal of Information Engineering and Educational Technology), 5*(2), 49-54.
16. Kantacarini, T., Erny Herwindiaty, D., & Hendryli, J. (2021). Kinerja metode Mahalanobis Distance yang dibentuk dari dua ukuran pusat dan dua dispersi multivariat (untuk ukuran similaritas klasifikasi image). *Jurnal Ilmu Komputer dan Sistem Informasi, 9*(1), 248-251.
17. Marpaung, F., Aulia, F., & Nabila, R. C. (2022). *Computer vision dan pengolahan citra digital*. Pustaka Aksara.
18. Muhathir, M., Santoso, M. H., & Larasati, D. A. (2021). Wayang image classification using SVM method and GLCM feature extraction. *Journal of Informatics and Telecommunication Engineering, 4*(2), 373-382.
19. Mustofa, M. I., Furqon, M. T., & Ratnawati, D. E. (2022). Penggunaan metode ekstraksi fitur tekstur Gray Level Co-Occurrence Matrix dan K-Nearest Neighbor untuk identifikasi jenis penyakit tanaman apel. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 6*(9), 4451-4458.
20. Nurhakim, B., Dikananda, A. R., Hamonangan, R., & Prasetyo, R. A. (2024). *Data cerdas di dunia pendidikan: Penerapan machine learning untuk kepuasan mahasiswa*. Mega Press Nusantara.
21. Oltean, M. (2017). *Fruits-360 dataset* [Data set]. Kaggle. https://www.kaggle.com/datasets/moltean/fruits
22. Rabbani, S., Safitri, D., Rahmadhani, N., Sani, A. A. F., & Anam, M. K. (2023). Perbandingan evaluasi kernel SVM untuk klasifikasi sentimen dalam analisis kenaikan harga BBM: Comparative evaluation of SVM kernels for sentiment classification in fuel price increase analysis. *MALCOM: Indonesian Journal of Machine Learning and Computer Science, 3*(2), 153-160.
23. Rianti, R., Andarsyah, R., & Awangga, R. M. (2024). Penerapan PCA dan algoritma clustering untuk analisis mutu perguruan tinggi di LLDIKTI Wilayah IV. *Nuansa Informatika, 18*(2), 2614-5405. https://journal.fkom.uniku.ac.id/ilkom
24. Saputra, M. R., Ansori, A. S. R., & Saputra, R. E. (2021). Deteksi kulit manusia pada gambar menggunakan algoritma RGB dan HSV. *eProceedings of Engineering, 8*(1).
25. Shevira, S., Made, I., Suarjaya, A. D., & Buana, P. W. (2022). Pengaruh kombinasi dan urutan pre-processing pada tweets bahasa Indonesia. *JITTER-Jurnal Ilmiah Teknologi dan Komputer, 3*(2).
26. Sihombing, P. R., Suryadiningrat, S., Sunarjo, D. A., & Yuda, Y. P. A. C. (2022). Identifikasi data outlier (pencilan) dan kenormalan data pada data univariat serta alternatif penyelesaiannya. *Jurnal Ekonomi dan Statistik Indonesia, 2*(3), 307-316.
27. Subandi, R., & Yudhana, A. (2023). Pre-processing pada klasifikasi citra medis pneumonia. *Decode: Jurnal Pendidikan Teknologi Informasi, 4*(1), 86-93.
28. Sundari, U. Y., Hidayatullah, M. A., & Fiardilla, F. (2023). Pengaruh teknik pengemasan, jenis kemasan dan kondisi penyimpanan terhadap sifat fisik dan organoleptik pada buah apel. *Jurnal Penelitian UPR, 3*(1), 17-23.
29. Syarifah, A., Riadi, A. A., & Susanto, A. (2022). Klasifikasi tingkat kematangan jambu bol berbasis pengolahan citra digital menggunakan metode K-Nearest Neighbor. *Jurnal Informatika Merdeka Pasuruan, 7*(1), 464720.
30. Wafa, H. S., Hadiana, A. I., & Umbara, F. R. (2022). Prediksi penyakit diabetes menggunakan algoritma Support Vector Machine (SVM). *Informatics and Digital Expert (INDEX), 4*(1), 40-45.
31. Yanti, N. P. D. T., & Asana, I. M. D. P. (2023). Sistem klasifikasi pengajuan kredit dengan metode Support Vector Machine (SVM). *Jurnal Sistem Cerdas, 6*(2), 123-133.

DOI: https://doi.org/10.69693/ijmst.v4i3.13422
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)