

# KLASIFIKASI TINGKAT KEMATANGAN PISANG DENGAN METODE HSV, GLCM, PCA, SVM

Aaron Sebastian <sup>1)</sup> Vivian <sup>2)</sup> Joshua Atmadjaya <sup>3)</sup>

<sup>1)2)3)</sup> Teknik Informatika Universitas Tarumanagara  
Jl. Letjen S. Parman No. 1, Jakarta Barat 11440

email : <sup>1)</sup> aaron.535230034@stu.untar.ac.id , <sup>2)</sup> vivian.535230066@stu.untar.ac.id, <sup>3)</sup> joshua.535230052@stu.untar.ac.id

## ABSTRAK

Penelitian ini bertujuan untuk mengklasifikasi tingkat kematangan pisang secara otomatis menggunakan kombinasi metode HSV, GLCM, PCA dan SVM. Proses dimulai dengan pengambilan citra pisang dari kaggle yang dibagi menjadi 4 jenis kematangan, yaitu sangat matang, matang, busuk dan mentah. Fitur warna diekstrak menggunakan metode HSV, sedangkan fitur tekstur diambil melalui metode GLCM. Fitur-fitur ini selanjutnya direduksi dimensinya menggunakan PCA untuk meningkatkan efisiensi dalam klasifikasi. Langkah terakhir adalah melakukan klasifikasi dengan algoritma SVM. Penelitian ini membuat beberapa perbandingan pada hasil eksperimen yaitu proporsi 60% data pelatihan dan 40% data pengujian, proporsi 70% data pelatihan dan 30% data pengujian, dan proporsi 80% data pelatihan dan 20% data pengujian. Hasil eksperimen pada proporsi 80% data pelatihan dan 20% data pengujian menunjukkan hasil performa tertinggi dan paling optimal. Temuan ini memiliki potensi untuk diterapkan dalam sektor pertanian guna mendukung proses penyebaran buah secara otomatis dan efisien.

## Key words

Klasifikasi, HSV, GLCM, PCA, SVM

## 1. Pendahuluan

Buah pisang atau *Musa paradisiaca* (dalam bahasa ilmiah) merupakan jenis buah dengan yang saat matang kulit akan berwarna kuning dan mengandung karbohidrat dan mineral [1]. Menurut data dari Badan Pusat Statistik (BPS), pisang menjadi buah tropis dengan volume produksi terbesar di Indonesia dalam beberapa tahun terakhir. Namun, salah satu tantangan dalam distribusi dan konsumsi pisang adalah tingkat kematangan yang cepat setelah proses panen. Pisang yang belum matang tidak disukai konsumen karena teksturnya yang keras dan rasanya yang sepat, tetapi pisang yang terlalu matang atau busuk tidak dapat dikonsumsi dan dapat menyebabkan kerugian ekonomi. Oleh karena itu, klasifikasi tingkat kematangan pisang secara akurat menjadi hal yang krusial baik itu bagi petani, ritel maupun konsumen [2].

Pada umumnya, dengan melihat warna pada buah pisang dapat membedakan tingkat kematangannya. Namun, mata manusia terkadang tidak akurat karena beberapa faktor diantaranya faktor usia yang membuat fokus penglihatan manusia berkurang. Beberapa kelemahan yang dapat terjadi jika melakukan klasifikasi kematangan buah pisang secara manual adalah proses yang lama, tingkat konsisten yang kurang dan juga tingkat akurasi yang rendah. Hal tersebut terjadi karena penentuan dilakukan secara subjektif dan faktor kelelahan yang dialami oleh pekerja.

Klasifikasi tingkat kematangan pisang sangat penting dalam dunia industri maupun perdagangan buah. Hingga saat ini, proses klasifikasi umumnya masih dilakukan secara manual dengan pengamatan visual manusia. Metode ini sangat subjektif dan dapat menimbulkan ketidakkonsistenan dalam penilaian apabila dilakukan dengan orang yang berbeda. Oleh karena itu, sistem klasifikasi otomatis berdasarkan teknik pemrosesan gambar dan kecerdasan buatan diperlukan untuk meningkatkan efisiensi, konsistensi, dan akurasi ketika menilai tingkat kematangan pisang.

Penelitian ini memanfaatkan dataset *Banana Ripeness Classification* yang diambil dari Kaggle, yang terdiri dari gambar pisang dalam empat kategori tingkat kematangan. Kategorinya dibagi menjadi *overripe* (sangat matang), *ripe* (matang), *rotten* (busuk), dan *unripe* (mentah). Untuk mengolah dan mengekstraksi informasi penting dari gambar, digunakan beberapa metode populer seperti HSV (*Hue, Saturation, Value*) dan RGB (*Red, Green, Blue*) untuk menangkap ciri warna, serta GLCM (*Gray Level Co-occurrence Matrix*) untuk menangkap ciri tekstur. *Overfitting* atau komputasi yang berat bisa terjadi jika jumlah fitur dari citra sangat besar, maka digunakan juga model PCA (*Principal Component Analysis*) untuk mereduksi dimensi fitur dan SVM (*Support Vector Machine*) untuk mengklasifikasi tingkat kematangan pada buah pisang secara akurat dan efisien.

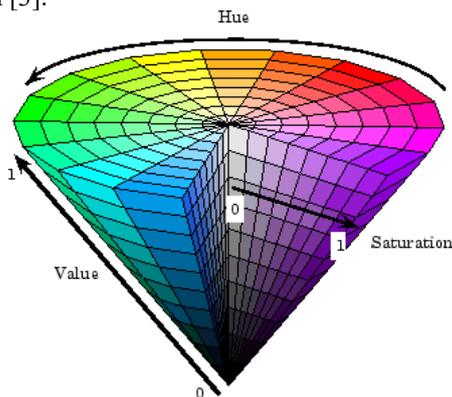
Maksud dari penelitian ini adalah untuk mengembangkan sistem klasifikasi otomatis berdasarkan pengelolaan citra dan pembelajaran mesin agar dapat dengan akurat, cepat dan objektif dalam mengklasifikasi tingkat kematangan pisang sebagai solusi dari metode penilaian manual yang subjektif dan tidak konsisten.

Adapun tujuan dari penelitian ini adalah untuk membangun dan mengembangkan model klasifikasi tingkat kematangan pisang berdasarkan citra digital dengan memanfaatkan kombinasi dari fitur warna (RGB dan HSV) dan tekstur (GLCM), meningkatkan efisiensi model dengan mereduksi dimensi dari fitur yang diekstraksi (PCA) dan mengimplementasikan dan mengevaluasi algoritma *support vector machine* (SVM) dalam mengklasifikasikan tingkat kematangan pisang secara otomatis berdasarkan fitur – fitur yang telah diekstrak dan direduksi.

## 2. Metode Penelitian

### 2.1. Hue, Saturation, Value (HSV)

Model HSV menggambarkan warna dengan istilah *Hue*, *Saturation* dan *Value*. *Hue* merujuk pada warna dasar, seperti merah, violet, dan kuning. Keunggulan utama dari ruang warna HSV dibandingkan dengan RGB adalah kemampuannya untuk secara lebih efektif memisahkan info tentang warna, tingkat kejenuhan dan kecerahan [3].



Gambar 1 Visualisasi model HSV

Adapun rumus HSV adalah sebagai berikut [4]:

#### 1. Rumus H (*Hue*)

$$\theta = \cos^{-1} \left( \frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}} \right) \dots\dots\dots(1)$$

Rumus (1) adalah rumus yang sering digunakan untuk menentukan sudut *Hue* dalam beberapa perubahan dari RGB ke HSV, khususnya yang memanfaatkan pendekatan geometris (contohnya mempersepsi ruang warna RGB sebagai heksagon).

$$H = \begin{cases} 0 & \text{Jika } B \leq G \\ 360 - \theta & \text{Jika } B > G \end{cases} \dots\dots\dots(2)$$

Rumus (2) juga merupakan metode untuk menyesuaikan *Hue* berdasarkan kedudukan relatif B dan G, memastikan *Hue* tertahan dalam kisaran 0-360 derajat.

#### 2. Rumus S (*Saturation*)

$$S = \begin{cases} 0 & \text{Jika } V = 0 \\ 1 - \frac{\min(r,g,b)}{v} & \text{Jika } V > 0 \end{cases} \dots\dots\dots(3)$$

Apabila Value (V) sama dengan 0 (warna hitam), maka *saturation* juga akan 0. Jika V lebih besar dari 0, *Saturation* diukur berdasarkan seberapa jauh warna tersebut berbeda dari warna abu-abu, yang dihitung dengan membandingkan nilai terendah dari komponen warna dengan Value.

#### 3. Rumus V (*Value*)

$$V = \max(r, g, b) \dots\dots\dots(4)$$

*Value* ditentukan sebagai angka tertinggi dari komponen R, G, dan B yang telah dinyatakan dalam bentuk normalisasi (biasanya dalam skala 0-1 atau 0-255 derajat)

### 2.2. Gray Level Co-occurrence Matrix (GLCM)

*Gray Level Co-occurrence Matrix* (GLCM) adalah pendekatan statistik yang berfokus pada dua variabel, yang digunakan untuk menganalisis tekstur pada gambar. Dalam GLCM terdapat beberapa karakteristik yang diukur untuk menganalisis tekstur, diantaranya atribut *Contrast*, *Correlation*, *Energy*, *Homogeneity*, *Entropy*, *Shade*, dan *Prominence* [5] [6] [7].

#### 1. *Contrast* yaitu untuk mengukur intensitas kontras di antara piksel-piksel sekitarnya. Adapun rumusnya pada persamaan 5.

$$I = \sum_x \sum_y (x, y)^2 p(x, y) \dots\dots\dots(5)$$

#### 2. *Correlation* yaitu untuk mengukur tingkat keterkaitan antar piksel dengan sekitarnya. Adapun rumusnya pada persamaan 6 .

$$corr = \sum_x \sum_y \frac{(x-\mu_x)(y-\mu_y) P(x,y)}{\sigma_x \sigma_y} \dots\dots\dots(6)$$

#### 3. *Energy* yaitu total dari elemen kuadrat GLCM yang telah dinormalisasi. Berikut rumusnya pada persamaan 7.

$$E = \sum_x \sum_y P(x, y)^2 \dots\dots\dots(7)$$

#### 4. *Homogeneity* yaitu ukuran yang menilai seberapa dekat distribusi elemen dalam GLCM dengan diagonal GLCM. Ini menunjukkan ukuran karakteristik homogenitas gambar. Adapun rumusnya pada persamaan 8.

$$H = \sum_x \sum_y \frac{p(x,y)}{1+|x-y|} \dots\dots\dots(8)$$

- 5. *Entropy* menunjukkan tingkat ketidakteraturan dalam distribusi tingkat keabu-abuan pada citra. Nilainya akan besar jika elemen dalam GLCM memiliki nilai yang secara relatif serupa. Sebaliknya nilai akan rendah jika elemen GLCM mendekati angka 0 dan 1 yang dapat dihitung dengan menggunakan rumus 9.

$$Entropy = \sum_x = 1 \sum_y p(x, y), \log p(x, y) \dots\dots(9)$$

- 6. *Shade* yaitu ukuran yang menunjukkan kemiringan (*skewness*) dari distribusi nilai-nilai dalam matriks GLCM jika dibandingkan dengan rata-ratanya. Nilai ini menunjukkan seberapa condong pola tekstur di dalam gambar. Semakin besar nilainya, semakin miring (tidak seimbang) penyebarannya terhadap pusat. Adapun rumusnya dapat dilihat pada persamaan 10.

$$Shade = \sum_x \sum_y (x + y - \mu_x - \mu_y)^3 p(x, y) \quad (10)$$

- 7. *Prominence* yaitu ukuran yang menunjukkan ketajaman puncak atau derajat kemencengan dari distribusi GLCM. Fitur ini bermanfaat untuk mendeteksi struktur tekstur dengan perbedaan kontras yang sangat tinggi atau adanya outlier dalam tekstur. Adapun rumusnya dapat dilihat pada persamaan 11.

$$prominence = \sum_x \sum_y (x + y - \mu_x - \mu_y)^4 p(x, y) \quad (11)$$

### 2.3. Principal Component Analysis (PCA)

PCA atau *Principal Component Analysis* (PCA) adalah metode yang dapat mengubah kumpulan data yang sebelumnya memiliki hubungan menjadi data yang tidak berhubungan satu sama lain (Komponen Utama). [8]. PCA merupakan teknik yang digunakan untuk mengurangi dimensi, yang membantu mengurangi jumlah fitur dalam data sambil tetap menjaga variabilitas tertinggi. Dengan mengurangi dimensi, PCA memungkinkan algoritma SVM bekerja dengan lebih efisien, terutama pada dataset dengan banyak fitur. Keunggulan dari metode PCA ini adalah kemampuan untuk mengubah data yang kompleks menjadi data yang lebih sederhana sambil tetap mempertahankan informasi dari dataset asli [9].

Tahapan PCA yang ditunjukkan pada rumus (12) yaitu pada awalnya setiap contoh data gambar diubah menjadi vektor-vektor, kemudian disatukan menjadi matriks x [10].

$$X = \begin{bmatrix} x11 & x12 & \dots & x1N \\ X21 & \ddots & \ddots & X2N \\ \vdots & \ddots & \ddots & \vdots \\ XM1 & XM2 & \ddots & XMN \end{bmatrix}_{M \times N} \dots\dots\dots(12)$$

$$m = \frac{1}{N} \begin{bmatrix} \sum_{k=1}^N x1k \\ \sum_{k=1}^N x2k \\ \vdots \\ \sum_{k=1}^N xMk \end{bmatrix} \dots\dots\dots(13)$$

Dimana:  
M = banyaknya baris dari matriks X  
N = banyaknya kolom dari matriks X  
m = vektor rata-rata dari matriks X

Setelah itu matriks  $\bar{m}$  terbentuk dengan cara menggandakan vektor m sebanyak N kali sesuai dengan rumus (14)

$$\bar{m} = [m \quad m \quad \dots \quad m]_{M \times N} \dots\dots\dots(14)$$

Selanjutnya dilakukan standarisasi data untuk memastikan bahwa semua fitur dalam dataset berada pada skala yang sama [11].

$$Z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \dots\dots\dots(15)$$

Dimana :  
 $Z_{ij}$  = nilai hasil standarisasi fitur ke-j pada data ke-i  
 $X_{ij}$  = nilai asli dari fitur ke-j pada data ke-i  
 $\mu_j$  = nilai rata-rata dari fitur ke-j  
 $\sigma_j$  = standar deviasi dari fitur ke-j

Tahapan selanjutnya adalah menghitung Matriks *covariance* C dengan rumus (16)

$$C = (X - \bar{m})(X - \bar{m})^T \dots\dots\dots(16)$$

Setelah matriks *covariance* didapatkan, langkah selanjutnya adalah menemukan vektor eigen yang terkait dengan nilai eigen dari matriks C. Kemudian, dilakukan pengurutan vektor eigen dengan mempertimbangkan nilai eigen terbesar, lalu pilih sejumlah vektor eigen tertinggi (maksimal) yang disebut matriks P. Tahap selanjutnya adalah mengubah matriks X yang berukuran MxN menjadi matriks Y dengan dimensi yang lebih kecil. Y merupakan matriks komponen utama yang terkait dengan transformasi linear P yang dinyatakan pada persamaan (17).

$$Y = P (X - \bar{m}) \dots\dots\dots(17)$$

Matriks Y yang dihasilkan adalah matriks yang berisi fitur-fitur yang digunakan untuk klasifikasi tingkat kematangan pisang. Semua tahapan yang berada pada metode PCA diterapkan untuk mendapatkan data latih dan data uji.

### 2.4. Support Vector Machine (SVM)

SVM atau *Support Vector Machine* adalah suatu metode pembelajaran yang memanfaatkan ruang hipotesis dengan fungsi-fungsi linear pada fitur yang memiliki dimensi tinggi. Tujuan SVM adalah untuk

mengidentifikasi fungsi pemisah yang paling optimal antara dua kelas. Dalam studi ini, fungsi pemisah yang dicari disebut fungsi linear [12].

Metode *multiclass* SVM menangani klasifikasi yang bukan biner seperti klasifikasi pisang berdasarkan sangat matang, matang, busuk dan mentah melalui pendekatan *One Against All (OAA)* atau *One Against One (OAO)*. Pendekatan OAA mengatasi masalah *multiclass* dengan  $N$  *decision boundary* dari pencarian *hyperplane* untuk setiap kelas ke- $i$  terhadap kelas lainnya. Sementara itu, OAO menyelesaikan masalah dengan  $N(N-1)/2$  *decision boundary* dari pencarian *hyperplane* antara setiap kelas dan setiap kelas lain [13].

Dalam metode OAA, untuk skenario klasifikasi dengan  $k$ -kelas, bertujuan untuk menemukan  $k$  fungsi pemisah, di mana  $k$  merupakan jumlah kelas. Misalkan ada sebuah fungsi pemisah yang dinamakan  $p$ . Pada pendekatan ini,  $p_i$  dilatih dengan semua data dari kelas lainnya yang diberikan label-1. Jika kita memiliki data untuk pelatihan  $(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)$  dimana  $x_l$  merupakan elemen dari  $R^n$ , dengan  $l = 1, 2, \dots, i$  sebagai data input dan  $y_i$  merupakan elemen dari  $S = \{1, 2, \dots, k\}$  sebagai kelas dari  $x_i$  yang bersangkutan, maka fungsi pemisah ke- $i$  bertujuan untuk menyelesaikan optimasi berikut (dalam masalah primal) [14].

$$\min_{w^i} \frac{1}{2} (w^i)^T w^i + c \sum_{j=1}^l t_j^i \dots\dots\dots(18)$$

Dengan syarat:

$$w^i x_j + b^i \geq 1 - t_j^i, \text{ jika } y_i = i \dots\dots\dots(19)$$

$$w^i x_j + b^i \leq -1 + t_j^i, \text{ jika } y_i \neq i \dots\dots\dots(20)$$

$$t_j \geq 0, j = 1, 2, \dots, k \dots\dots\dots(21)$$

Pada penelitian ini untuk mencari setiap *hyperplane* dari 4 (empat) kelas yang digunakan yaitu kelas sangat matang, matang, busuk dan mentah digunakan metode *multiclass* SVM dengan pendekatan OAA. Hal ini disebabkan karena metode OAA menunjukkan kinerja yang lebih unggul dibandingkan dengan metode OAO dan lebih cocok untuk jumlah label yang relatif sedikit.

Secara sederhana, konsep SVM adalah untuk memisahkan dua kelas dengan menemukan *hyperplane* yang paling optimal pada suatu ruang input dengan memperbesar jarak antar kelompok. Saat melakukan prediksi klasifikasi potensi pada data, algoritma SVM terkenal karena metode yang menghasilkan tingkat akurasi yang tinggi. Berikut adalah persamaan algoritma SVM (22) [15]

$$y_i (x_i \cdot w + b) \geq 0 \dots\dots\dots(22)$$

Untuk klasifikasi nonlinear, SVM memanfaatkan teknik kernel *trick* yang efektivitasnya tergantung pada tipe fungsi kernel yang dipilih sesuai dengan karakteristik *dataset*. Kernel *trick* berfungsi untuk mentransformasikan vektor input dari ruang berdimensi rendah ke ruang

berdimensi tinggi. Rumus fungsi kernel *trick* adalah sebagai berikut (23)

$$K(x_i, x_j) = \phi_i(x_i) \cdot \phi_j(x_j) \dots\dots\dots(23)$$

Terdapat berbagai jenis kernel yang sering dipakai, antara lain *polynomial*, RBF, dan *sigmoid*. Kernel *polynomial* biasanya digunakan untuk klasifikasi data ketika data latih yang tersedia sudah dalam bentuk yang normal. Beberapa parameter yang sering digunakan *degree* ( $d$ ), koefisien (coef),  $C$ , dan *gamma*. Persamaan untuk kernel *polynomial* adalah sebagai berikut (24)

$$K(\bar{x}_i, \bar{x}_j + 1)^d \dots\dots\dots(24)$$

*Radial Basis Function Kernel* (RBF) Kernel atau kelas *Gaussian* mengandalkan parameter yang biasanya digunakan dalam kernel RBF, yaitu  $\gamma$  dan  $C$ . Kernel *Gaussian* sering dipilih karena kemampuan kerjanya yang sangat baik. Rumus untuk kernel *Gaussian* adalah sebagai berikut (25)

$$K(\bar{x}_i, \bar{x}_j) = \exp(-\gamma \|\bar{x}_i - \bar{x}_j\|^2) \dots\dots\dots(25)$$

Kernel *sigmoid* merupakan salah satu jenis kernel yang berasal dari jaringan saraf tiruan atau JST. Parameter yang biasanya diterapkan pada kernel sigmoid adalah  $\gamma$  dan  $C$ , yang persamaannya dapat ditulis sebagai berikut (26)

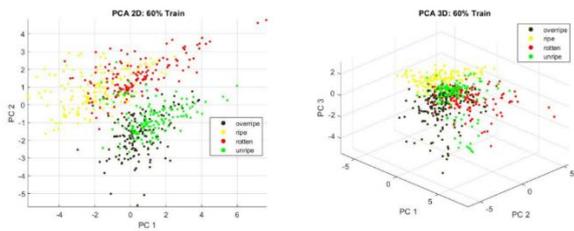
$$K(\bar{x}_i, \bar{x}_j) = \tanh(\alpha \bar{x}_i \cdot \bar{x}_j + \beta) \dots\dots\dots(26)$$

### 3. Hasil Percobaan

Berdasarkan pengembangan model klasifikasi citra tingkat kematangan pisang, yang dimulai dari tahap preprocessing (GLCM, HSV, dan PCA) hingga penerapan SVM sebagai metode klasifikasi, maka tercapai berbagai hasil dan temuan. Pengujian dari performa dan efisiensi model merupakan salah satunya. Hasil dan temuan pada pengembangan kemudian dibagi berdasarkan proporsi data latih dan data uji yang telah ditetapkan. Berikut adalah hasil dari eksperimen tersebut:

1. Proporsi 60% Data Pelatihan dan 40% Data Pengujian

Pada skenario ini, sebanyak 60% dari total dataset (600 gambar) digunakan sebagai data pelatihan, dan 40% sisanya (400 gambar) dialokasikan untuk data pengujian. Tujuan dari proporsi ini adalah untuk mengevaluasi model terhadap kondisi standar, dimana data pelatihan mencukupi, tetapi tetap terdapat ruang untuk menguji generalisasi model terhadap data baru.



Gambar 2 Hasil PCA skenario 60% dan 40%

Hasil PCA pada Gambar 2 menyatakan bahwa separasi antar kelas cukup terlihat, walaupun terdapat beberapa kelas yang menunjukkan hasil separasi yang kurang baik. Terdapat juga kelas yang mengalami tumpang tindih.

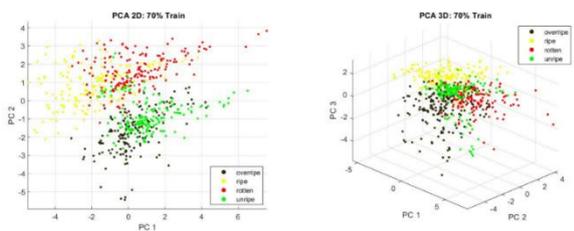


Gambar 3 Confusion matrix skenario 60% dan 40%

Hasil berikutnya merupakan *confusion matrix* dari model klasifikasi SVM pada Gambar 3. *Confusion matrix* tersebut menunjukkan bahwa model menunjukkan performa yang tinggi pada tiap kelas. Pada model yang tidak menerapkan PCA, tingkat kesalahan baik itu dari sisi recall dan precision hanya berkisar 2% hingga 5%, sedangkan pada model yang menerapkan PCA, terdapat kesalahan yang mencapai 12% pada nilai recall dari kelas ripe, dan kesalahan 11,2% pada nilai precision dari kelas rotten.

2. Proporsi 70% Data Pelatihan dan 30% Data Pengujian

Pada skenario ini, sebanyak 70% dari total dataset (700 gambar) digunakan sebagai data pelatihan, dan 30% sisanya (300 gambar) dialokasikan untuk data pengujian. Tujuan dari proporsi ini adalah untuk memberikan data pelatihan yang lebih besar kepada model, sehingga diharapkan dapat meningkatkan kemampuan model untuk generalisasi terhadap data baru.



Gambar 4 Hasil PCA skenario 70% dan 30%

Berdasarkan hasil PCA pada Gambar 4 tersebut, dapat dilihat bahwa distribusi antar kelas terlihat lebih terkonsentrasi. Separasi antar kelas pada skenario ini

terlihat lebih baik dibandingkan dengan skenario sebelumnya.

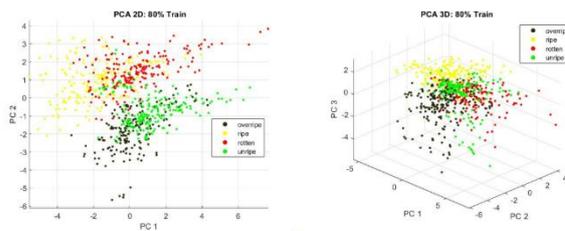


Gambar 5 Confusion matrix skenario 70% dan 30%

Hasil berikutnya merupakan *confusion matrix* dari model klasifikasi SVM pada Gambar 5. *Confusion matrix* tersebut menunjukkan bahwa proporsi ini memiliki performa yang cenderung lebih rendah dibandingkan proporsi sebelumnya, baik itu dari segi *recall* dan *precision* pada model yang menggunakan PCA dan yang tidak. Tingkat kesalahan pada model yang menggunakan pca didominasi oleh angka diatas 5%. Batas atas dari kesalahan model yang tidak menggunakan PCA meningkat ke angka 6,6%.

3. Proporsi 80% Data Pelatihan dan 20% Data Pengujian

Pada skenario ini, sebanyak 80% dari total dataset (800 gambar) digunakan sebagai data pelatihan, dan 20% sisanya (200 gambar) dialokasikan untuk data pengujian. Tujuan dari proporsi yang menitikberatkan pada data pelatihan adalah untuk memaksimalkan tingkat pembelajaran model terhadap variasi dari fitur yang tersedia dalam *dataset*. Diharapkan model mampu mengenali pola – pola yang penting secara lebih akurat, sehingga dapat menghasikan prediksi yang lebih tepat pada data baru.



Gambar 6 Hasil PCA skenario 60% dan 40%

Hasil PCA pada Gambar 6 tidak menunjukkan perubahan yang signifikan jika dibandingkan dengan dua skenario sebelumnya. Walaupun demikian, terlihat bahwa kepadatan antar kelas meningkat.



Gambar 7 Confusion matrix skenario 60% dan 40%

Hasil berikutnya merupakan *confusion matrix* dari model klasifikasi SVM pada Gambar 7. *Confusion matrix* tersebut menunjukkan bahwa proporsi ini memiliki performa yang paling tinggi dibandingkan dengan proporsi lainnya. Pada model yang tidak menerapkan PCA, hampir setiap kelas memiliki nilai *precision* 100%, hanya terdapat satu kelas yaitu *ripe* yang memiliki 5,7% tingkat kesalahan. Untuk model yang menerapkan PCA, performanya juga paling tinggi dibandingkan dengan proporsi lainnya, dimana tingkat kesalahan pada nilai *precision* dan *recall* berkisar pada 2% hingga 6%.

Akurasi Model Klasifikasi SVM				
Proporsi	Dataset Uji	PCA	Akurasi	Running Time
60% Training dan 40% Testing	Training	Tidak	98,67%	0,24189s
		Ya	96,67%	0,20497s
	Testing	Tidak	97,75%	0,05697s
		Ya	94,00%	0,05252s
70% Training dan 30% Testing	Training	Tidak	99,14%	0,27308s
		Ya	97,00%	0,27332s
	Testing	Tidak	95,67%	0,05037s
		Ya	93,67%	0,04730s
80% Training dan 20% Testing	Training	Tidak	99,50%	0,37361s
		Ya	97,00%	0,35765s
	Testing	Tidak	98,50%	0,03792s
		Ya	96,00%	0,03428s

Gambar 8 Akurasi model klasifikasi SVM

Berdasarkan hasil evaluasi terhadap akurasi dan efisiensi model klasifikasi dengan SVM pada Gambar 8, diperoleh bahwa model dengan proporsi 80% dan 20% tanpa menggunakan PCA menghasilkan akurasi tertinggi di angka 99,50%. Model paling efisien berasal dari proporsi yang sama dengan menggunakan PCA, dimana *running time* (waktu pemrosesan) berada di angka 0,03428s.

#### 4. Kesimpulan

Hasil yang ditampilkan pada laporan penelitian ini merupakan sampel dari satu kali running program, sehingga penting untuk digarisbawahi bahwa setiap kali program dijalankan ulang, hasil akurasi, PCA, dan *confusion matrix* dapat bervariasi meskipun tidak signifikan. Hal tersebut bergantung pada *dataset* yang dipilih saat pengacakan dan kondisi program saat eksekusi. Oleh karena itu, analisis dan pembahasan berikut ini merupakan representasi dari salah satu skenario program yang berhasil dieksekusi, sehingga tidak bersifat absolut, tetapi sebagai gambaran umum terhadap kinerja model SVM dalam klasifikasi data citra tingkat kematangan pisang. Penelitian ini menerapkan pendekatan klasifikasi gambar yang menggunakan metode *support vector machine* (SVM) dengan *kernel Gaussian* (RBF), untuk mendeteksi tingkat kematangan pisang yang dibagi ke 4 kelas yaitu *unripe*, *ripe*, *overripe*, dan *rotten*. Data citra tingkat kematangan pisang juga melalui preprocessing dengan metode ekstraksi fitur yaitu HSV dan GLCM, serta reduksi dimensi dengan

menggunakan *principal component analysis* (PCA). Proses klasifikasi dilaksanakan dengan pendekatan *One-vs-All* karena terdapat lebih dari 2 kelas pada dataset.

Berdasarkan perbandingan dari tiga scenario proporsi data, maka dapat disimpulkan bahwa semakin besar data pelatihan, maka performa dan akurasi model juga akan meningkat, karena kemungkinan model untuk mengenali pola pada gambar semakin besar. Penerapan PCA juga menunjukkan adanya peningkatan efisiensi model klasifikasi, dengan *running time* yang lebih singkat. Walaupun mampu menurunkan kompleksitas dari komputasi dan meningkatkan efisiensi, penggunaan dimensi reduksi dengan metode PCA perlu diimplementasikan dengan seksama, karena terdapat kemungkinan dimana fitur – fitur utama yang bersifat krusial terhadap klasifikasi gambar hilang direduksi. Secara keseluruhan, model klasifikasi berbasis SVM pada penelitian ini menunjukkan hasil yang baik dalam mendeteksi tingkat kematangan pisang dalam data citra.

#### REFERENSI

- [1] S. P. Adenugraha, V. Arinal, and D. I. Mulyana, "Klasifikasi Kematangan Buah Pisang Ambon Menggunakan Metode KNN dan PCA Berdasarkan Citra RGB dan HSV," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 1, p. 9, Jan. 2022, doi: 10.30865/mib.v6i1.3287.
- [2] Y. Amrozi, D. Yuliati, A. Susilo, N. Novianto, and R. Ramadhan, "Klasifikasi Jenis Buah Pisang Berdasarkan Citra Warna dengan Metode SVM," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 11, no. 3, pp. 394–399, Dec. 2022, doi: 10.32736/sisfokom.v11i3.1502.
- [3] M. Muchtar and R. A. Muchtar, "PERBANDINGAN METODE KNN DAN SVM DALAM KLASIFIKASI KEMATANGAN BUAH MANGGA BERDASARKAN CITRA HSV DAN FITUR STATISTIK," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, Apr. 2024, doi: 10.23960/jitet.v12i2.4010.
- [4] D. Wandu and N. Hayati, "DETEKSI KELAYUAN BUNGA MAWAR DENGAN METODE TRANSFORMASI RUANG WARNA HSI DAN HSV," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 5, 2021.
- [5] R. P. Suhendri, "Metode Grayscale Co-occurrence Matrix (GLCM) Untuk Klasifikasi Jenis Daun Jambu Air Menggunakan Algoritma Neural Network," *Journal of Information Technology*, vol. 01, no. 01, 2019.
- [6] N. Dewi Saputri, K. kunci-Pempek, T. Blok Citra, and J. Syaraf Tiruan, "Identifikasi Kadar Ikan Pada Pempek Menggunakan Teknik Blok Citra Dengan Fitur GLCM Dan Metode JST," *Jurnal Algoritme*, vol. 3, no. 1, 2022.
- [7] S. M. Vijithananda et al., "GLCM Texture Feature Analysis of MRI-ADC Images to Differentiate Glioma Grades Using Machine Learning Techniques," Aug. 30, 2022. doi: 10.36227/techrxiv.20624406.v1.
- [8] M. Mayasari, D. Iskandar Mulyana, M. Betty Yel, and S. Tinggi Ilmu Komputer Cipta Karya Informatika Jl Raden, "KOMPARASI KLASIFIKASI JENIS TANAMAN RIMPANG MENGGUNAKAN PRINCIPAL COMPONENT ANALYSIS, SUPPORT VECTOR MACHINE, K-NEAREST NEIGHBOR DAN DECISION

- TREE,” *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 6, no. 2, 2022.
- [9] S. Yulihartati and A. X. Ramadhanu Universitas Putra Indonesia YPTK Padang Jl Raya Lubuk Begalung nan Padang, “IMPLEMENTASI ALGORITMA KNN, PCA PADA KLASIFIKASI BUAH PIR MADU, PIR HIJAU DAN APEL MERAH,” *Jurnal Mahasiswa Teknik Informatika*, vol. 9, no. 4, p. 5593, 2025.
- [10] R. Kosasih, “Pengenalan Wajah Menggunakan PCA dengan Memperhatikan Jumlah Data Latih dan Vektor Eigen,” *Jurnal Informatika Universitas Pamulang*, vol. 6, no. 1, p. 1, Mar. 2021, doi: 10.32493/informatika.v6i1.7261.
- [11] R. Rianti, R. Andarsyah, and R. M. Awangga, “Penerapan PCA dan Algoritma Clustering untuk Analisis Mutu Perguruan Tinggi di LLDIKTI Wilayah IV,” *NUANSA INFORMATIKA*, vol. 18, no. 2, pp. 2614–5405, 2024.
- [12] D. Irawan, E. B. Perkasa, Y. Yurindra, D. Wahyuningsih, and E. Helmud, “Perbandingan Klasifikasi SMS Berbasis Support Vector Machine, Naive Bayes Classifier, Random Forest dan Bagging Classifier,” *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 10, no. 3, pp. 432–437, Dec. 2021, doi: 10.32736/sisfokom.v10i3.1302.
- [13] N. G. A. Dasriani, L. A. G. Pariandi, and I. M. Y. Dharma, “Analisis Sentimen Program Jaminan Kesehatan Nasional Menggunakan Multiclass Support Vector Machine,” *BIOS: Jurnal Teknologi Informasi dan Rekayasa Komputer*, vol. 6, no. 1, pp. 20–30, Nov. 2024, doi: 10.37148/bios.v6i1.136.
- [14] S. Talib, S. Sudin, M. Dzikrullah Suratin, J. K. Ahmad Dahlan No, S. Kec Ternate Selatan, and K. Ternate Maluku Utara, “PENERAPAN METODE SUPPORT VECTOR MACHINE (SVM) PADA KLASIFIKASI JENIS CENGKEH BERDASARKAN FITUR TEKSTUR DAUN,” *Jurnal RESTIA*, vol. 2, no. 1, 2024.
- [15] R. S. Wijaya, A. Qur’ania, and I. Anggraeni, “Klasifikasi Penyakit Cacar Monyet Menggunakan Support Vector Machine (SVM),” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 4, pp. 1253–1260, Jul. 2024, doi: 10.57152/malcom.v4i4.1417.

**Aaron Sebastian**, saat ini sebagai Mahasiswa program studi Teknik Informatika Universitas Tarumanagara.

**Vivian**, saat ini sebagai Mahasiswa program studi Teknik Informatika Universitas Tarumanagara.

**Joshua Atmadjaya**, saat ini sebagai Mahasiswa program studi Teknik Informatika Universitas Tarumanagara.