Klasifikasi Citra Jamur Menggunakan SVM dengan PCA Berbasis Ekstraksi Fitur Hibrida

I Putu Andika Arsana Putra[a1], I Gusti Agung Gede Arya Kadyanan[a2]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana, Bali
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]putra.2308561063@student.unud.ac.id
[2]gungde@unud.ac.id

Abstract

The general public still faces significant difficulty in differentiating between poisonous and non-poisonous mushrooms due to their high visual similarity. This has led to numerous poisoning incidents due to the consumption of poisonous mushrooms. Between 2010 and 2020, there were 76 reported cases of poisoning involving 550 victims, 9 of whom died. To address this issue, a classification model was developed to differentiate between poisonous and non-poisonous mushrooms using Support Vector Machine (SVM) and Principal Component Analysis (PCA) algorithms based on hybrid feature extraction. The dataset for this study was obtained from Kaggle. The model built using PCA successfully accelerated the model training time compared to training the model without using PCA. Hyperparameter tuning was performed to find the best combination of parameters, resulting in RBF kernel, C value of 10, and gamma set to scale. The model was evaluated using a confusion matrix to determine accuracy and class-specific metrics. The model performed well, achieving 85% accuracy on the test data.

Keywords: *Mushrooms, Support Vector Machine (SVM), Principal Component Analysis (PCA), Classification, Machine Learning*

1. Pendahuluan

Jamur merupakan salah satu organisme di lingkungan tropis yang memiliki banyak spesies. Diperkirakan terdapat lebih dari 1.500.000 spesies jamur di dunia dan spesies yang baru dikenali hanya sekitar 74.000 spesies. Spesies jamur tersebut terbagi menjadi 3 kelas, yaitu jamur yang tidak beracun sehingga dapat dikonsumsi, jamur beracun, dan jamur yang tidak diketahui digabungkan ke dalam kelas beracun [1]. Spesies jamur yang sudah dikenali memiliki manfaat dalam berbagai bidang, seperti pangan, pertanian, kesehatan dan ekonomi. Namun, spesies jamur beracun sangat berbahaya untuk dikonsumsi karena dapat menyebabkan infeksi pada tubuh manusia bahkan sampai menyebabkan kematian. Sudah terdapat banyak kasus keracunan akibat mengkonsumsi jamur beracun. Tercatat sebanyak 76 kasus keracunan telah terjadi selama kurun waktu 10 tahun dari tahun 2010 sampai tahun 2020 di berbagai wilayah di Indonesia. Sebanyak 550 orang telah menjadi korban dan 9 diantaranya telah meninggal dunia [2]. Banyak nya kasus keracunan dalam mengonsumsi jamur ini disebabkan karena masyarakat kesulitan untuk membedakan antara jamur beracun dan tidak beracun yang memiliki tingkat kemiripan tinggi berdasarkan pengamatan visual. Oleh karena itu, pendekatan klasifikasi berbasis citra digital menjadi salah satu solusi untuk mengatasi kesulitan dalam mengklasifikasikan jamur beracun atau tidak secara otomatis.

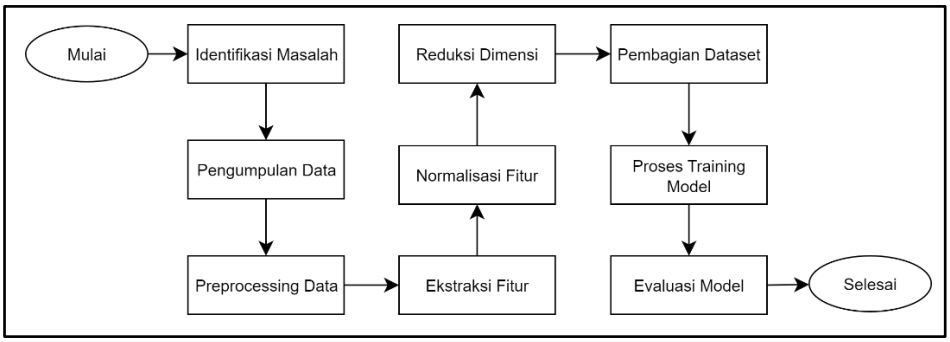
Penelitian ini bertujuan untuk mengembangkan suatu sistem klasifikasi citra jamur beracun dan tidak beracun berdasarkan citra dengan menggunakan *machine learning*. Penggunaan *machine learning* bertujuan agar model yang dikembangkan dapat mengenali pola-pola dari data yang sudah ada untuk melakukan suatu pengklasifikasian [3]. Model *machine learning* yang digunakan dalam penelitian ini adalah *Support Vector Machine* (SVM), karena memiliki kemampuan yang baik dalam memisahkan dua kelas berbeda dengan margin maksimum dan menangani data berdimensi tinggi. Sudah terdapat banyak penelitian sebelumnya yang menggunakan model SVM

untuk membangun sistem klasifikasi, salah satunya penelitian yang dilakukan oleh Talib dkk yang membangun model *machine learning* dengan SVM berbasis ektraksi fitur GLCM untuk mengklasifikasikan jenis cengkeh berdasarkan tekstur daun yang berhasil mencapai akurasi sebesar 56,67% [4]. Dalam penelitian ini menggunakan ekstraksi fitur hibrida untuk mengambil fitur-fitur penting dari citra jamur, yaitu dengan menggabungkan ekstraksi fitur histogram warna untuk mengekstraksi fitur warna, *Gray Level Co-occurrence Matrix* (GLCM) untuk mengekstraksi fitur tekstur, dan *Histogram of Oriented Gradients* (HOG) untuk mengekstraksi fitur bentuk. Untuk mengatasi masalah dimensi tinggi yang dihasilkan dari fitur gabungan, digunakan teknik reduksi dimensi *Principal Component Analysis* (PCA) yang berguna untuk menghindari *overfitting* dan mempercepat proses training model. Sistem ini diharapkan dapat membantu masyarakat awam untuk membedakan jamur yang beracun atau tidak berdasarkan citra.

Terdapat beberapa penelitian sebelumnya tentang klasifikasi jamur menggunakan data citra, seperti klasifikasi citra jenis jamur yang layak dikonsumsi dengan menggunakan algoritma multiclass SVM yang dilakukan oleh chusna dkk dengan menghasilkan akurasi tertinggi sebesar 83% [5], penelitian kedua dilakukan oleh Dedy Armiady untuk mengklasifikasikan jenis jamur dengan menggunakan algoritma *Stochastic Gradient Descent* yang menghasilkan akurasi sebesar 62.4% [3]. Selain itu terdapat juga penelitian klasifikasi jamur beracun dengan menggunakan data tabular atau tabel yang berisi atribut ciri-ciri jamur yang dilakukan oleh Batubara dkk. Penelitian ini menggunakan dua buah algoritma, yaitu algoritma *naive bayes* yang menghasilkan rata-rata akurasi sebesar 92% dan algoritma KNN yang menghasilkan rata-rata akurasi sebesar 98% [1]. Perbedaan penelitian saya dengan penelitian yang telah dilakukan sebelumnya terletak pada dataset dan penggunaan algoritma *machine learning* nya.

2. Metode Penelitian

Penelitian ini terdiri dari 9 proses alur untuk mendapatkan hasil klasifikasi citra jamur dengan menggunakan algoritma SVM yang dijelaskan pada diagram alir penelitian yang dapat dilihat pada gambar 1 dan akan dijelaskan lebih rinci dalam bentuk sub bab pada bagian metode penelitian.



Gambar 1. Diagram Alir Penelitian Sistem Klasifikasi Citra Jamur Menggunakan SVM

2.1. Identifikasi Masalah

Permasalahan utama dalam mengklasifikasikan jamur beracun atau tidak oleh masyarakat adalah ketidakmampuan dalam membedakan spesies jamur secara pengamatan visual karena memiliki tingkat kemiripan tinggi. Ketidakmampuan ini dapat berisiko dalam pemilihan jamur untuk dikonsumsi karena dapat menimbulkan keracunan apabila salah mengklasifikasikan spesies jamur. Oleh karena itu, dibutuhkan sistem klasifikasi otomatis berbasis citra yang dapat membantu dalam proses identifikasi jamur secara cepat dan akurat.

2.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang didapatkan dari kaggle, yaitu dataset *Edible & Poisonous Mushroom Classification* yang dapat diakses pada

p-ISSN: 2986-3929
e-ISSN: 3032-1948

https://www.kaggle.com/datasets/benedictusjason/edible-and-poisonous-mushroom-classification. Dataset tersebut terdiri atas tiga folder utama, yaitu *train*, *test*, dan *val*, yang masing-masing berisi 2.256, 282, dan 282 citra jamur. Pada penelitian ini hanya mengambil data dari folder *train* saja, dengan melakukan penyesuaian pada penamaan subfolder menjadi tidak beracun untuk *edible* yang memiliki 1200 citra jamur dan beracun untuk *poisonous* yang memiliki 1056 citra jamur. Citra-citra ini disimpan dalam format .jpg atau .png dan mode warna tiga kernel yaitu RGB dengan resolusi, dimensi dan ukuran file yang sangat bervariasi. Untuk contoh citra jamur dapat dilihat pada tabel 1 yang berisi 2 citra jamur beracun dan 2 citra jamur tidak beracun.

Tabel 1. 2 Contoh Citra Jamur Masing-Masing Kelas

Beracun	**Tidak Beracun**
	
Gambar 2. Jamur *Cortinarius Rubellus*	**Gambar 3.** Jamur *Agaricus Subrufescens*
	
Gambar 4. Jamur *Amanita Bisporigera*	**Gambar 5.** Jamur *Cantharellus Cibarius*

2.3. Preprocessing Data

Pada tahap *preprocessing*, seluruh citra jamur dari dataset yang telah dikumpulkan akan dikondisikan agar memenuhi standar input yang sesuai untuk proses ekstraksi fitur dan pelatihan model. Proses *preprocessing* pada penelitian ini dimulai dengan membaca setiap citra dari subfolder kelas beracun dan tidak beracun dengan menggunakan OpenCV. Selanjutnya, dilakukan pengambilan sampel data citra secara acak sebanyak 1.000 citra jamur dari masing-masing kelas. Tahap akhir dalam proses *preprocessing* pada penelitian ini adalah mengubah ukuran setiap citra menjadi 128×128 piksel agar seluruh citra memiliki ukuran dimensi yang seragam. Standardisasi ukuran ini bertujuan untuk memastikan kompatibilitas dengan fungsi-fungsi ekstraksi fitur, sekaligus mengurangi beban komputasi selama pelatihan model.

2.4. Ekstraksi Fitur

Tahap ekstraksi fitur bertujuan untuk mengubah citra jamur menjadi representasi numerik yang dapat dipahami oleh algoritma SVM. Penelitian ini menggunakan pendekatan ekstraksi fitur hibrida dengan menggabungkan tiga jenis ekstraksi fitur, yaitu Histogram Warna, *Gray Level Co-occurrence Matrix* (GLCM), dan *Histogram of Oriented Gradients* (HOG).

- Histogram Warna
  Histogram warna merupakan metode ekstraksi fitur untuk merepresentasikan warna dalam citra pada tiga kanal warna (*Red, Green,* dan *Blue*). Histogram ini dihitung dengan pembagian bin yang merupakan proses membagi rentang nilai piksel dari 0 sampai 255 untuk citra 8-bit ke dalam beberapa kelompok interval yang disebut bin. Dalam penelitian ini, masing-masing kanal warna dibagi menjadi 8 bin, sehingga menghasilkan pembagian

bin sebesar (8, 8, 8) [6]. Setelah dihitung, histogram dinormalisasi ke rentang 0 sampai 1 agar fitur berada dalam skala yang seragam dengan menggunakan rumus dibawah ini:

$$H_{norm}(i) = \frac{H(i)}{\sum_{j=1}^{N} H(j)} \quad (1)$$

Keterangan:

$H(i)$	:	Frekuensi jumlah piksel dari bin ke-(i) pada histogram warna.
$\sum_{j=1}^{N} H(j)$	:	Total seluruh frekuensi bin dalam histogram yaitu jumlah total piksel yang dihitung dalam histogram warna.
N	:	Jumlah total bin histogram yang pada penelitian ini terdapat 512 bin (8x8x8).
$H_{norm}(i)$	:	Nilai normalisasi histogram pada bin ke-(i), yaitu proporsi piksel dalam bin ke-(i) terhadap total seluruh piksel.

Dalam penelitian ini, perhitungan untuk melakukan ektraksi fitur histogram warna menggunakan library cv2 dan numpy dari python. Fungsi cv2.calcHist(...) digunakan untuk menghitung histogram warna, cv2.normalize(...) digunakan untuk normalisasi menggunakan rumus 1 dan flatten() digunakan untuk menjadikan histogram menjadi satu array vektor fitur.

- GLCM

Grey Level Co-occurrence Matrix merupakan metode untuk mengekstraksi fitur tekstur pada citra dengan mengggunakan matriks untuk mencerminkan nilai frekuensi kemunculan pasangan piksel berdasarkan nilai intensitas keabuan tertentu dan hubungan spasialnya [7]. Dari matriks GLCM tersebut, dapat dihitung berbagai properti statistik, seperti *contrast, dissimilarity, homogeneity, energy, ASM, dan correlation* untuk merepresentasikan tekstur citra dengan menggunakan rumus matematis dibawah.

$$Contrast = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i-j)^2 \cdot P(i,j) \quad (2)$$

$$\mathrm{Dissimilarity} = \sum_{\mathrm{i}=0}^{\mathrm{N}-1} \sum_{\mathrm{j}=0}^{\mathrm{N}-1} |\mathrm{i}-\mathrm{j}| \cdot \mathrm{P}(\mathrm{i},\mathrm{j}) \quad (3)$$

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

$$Correlation = \frac{\sum_i j\,(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i \sigma_j} \quad (5)$$

$$ASM = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i,j)^2 \quad (6)$$

$$Energy = \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i,j)^2} \quad (7)$$

Keterangan:

$Contrast$	:	Mengukur tingkat perbedaan kontras antara piksel.
$\mathrm{Dissimilarity}$	:	Mengukur ketidaksamaan antar piksel.
$Homogeneity$	:	Mengukur seberapa dekat distribusi elemen GLCM ke diagonal utama.
$Correlation$	:	Mengukur hubungan linear antara nilai intensitas piksel.
ASM	:	Mengukur kehomogenan atau keseragaman tekstur.
$Energy$	:	Mengukur energi atau kekuatan pola tekstur.
$P(i,j)$	:	Probabilitas terjadinya pasangan piksel dengan intensitas (i) yang merupakan nilai intensitas piksel pertama dalam pasangan piksel dan (j) yang merupakan nilai intensitas piksel kedua dalam pasangan piksel pada jarak dan arah tertentu.
N	:	Jumlah nilai intensitas berbeda yang mungkin terdapat dalam citra seperti 256 untuk citra 8-bit.
μ	:	Rata-rata nilai intensitas piksel dalam GLCM.

σ : Sebaran nilai intensitas piksel.

Dalam penelitian ini, perhitungan untuk melakukan ektraksi fitur GLCM menggunakan library *greycomatrix* dan *greycoprops*. *Greycomatrix* digunakan untuk membangun matriks GLCM dari citra *grayscale* dengan menghitung frekuensi kemunculan pasangan nilai intensitas piksel dalam arah dan jarak tertentu. *Greycoprops* digunakan untuk menghitung nilai statistik tekstur dari matriks GLCM yang dihasilkan oleh *greycomatrix* seperti *contrast, dissimilarity, homogeneity, correlation, ASM* dan *energy* sesuai dengan perhitungan rumus yang sudah dijelaskan pada rumus 2 sampai 7.

- HOG
  HOG adalah metode ekstraksi fitur untuk merepresentasikan bentuk dan struktur objek dalam citra yang bekerja dengan menganalisis arah gradien atau garis tepi dari setiap piksel dalam suatu cell [8]. Sebelum di ekstraksi dengan HOG, citra harus dikonversi terlebih dahulu ke *grayscale*. Proses HOG diawali dengan menghitung gradien horizontal (Gx) dan vertikal (Gy), kemudian menentukan besar magnitude dan arah gradien pada setiap piksel. Nilai magnitude ini dikelompokkan ke dalam histogram berdasarkan arah gradiennya yang dibagi menjadi 9 bin dari rentang 0° hingga 180°. Histogram ini dihitung pada sel kecil dan dinormalisasi menjadi blok untuk menghasilkan vektor fitur. Berikut ini rumus-rumus yang digunakan pada HOG.

$$G_x = I(x+1, y) - I(x-1, y) \tag{8}$$

$$G_y = I(x, y+1) - I(x, y-1) \tag{9}$$

$$Magnitude: M = \sqrt{G_x^2 + G_y^2} \tag{10}$$

Keterangan:
$I(x, y)$: Intensitas piksel pad koordinat (x, y).
M : Besaran gradien yang digunakan untuk menyusun histogram arah gradien.

Dalam penelitian ini, perhitungan untuk melakukan ekstraksi fitur HOG dilakukan menggunakan library hog dari modul skimage.feature. Fungsi hog digunakan untuk menghitung vektor fitur berdasarkan arah dan besarnya gradien pada setiap piksel dalam citra *grayscale* sesuai dengan rumus 8 sampai 10.

Setelah ketiga jenis fitur selesai diekstrak, maka seluruh fitur tersebut digabungkan menggunakan fungsi numpy.concatenate() untuk membentuk satu vektor fitur per citra. Vektor hasil gabungan ini yang akan digunakan dalam proses pelatihan model klasifikasi.

2.5. Normalisasi Fitur

Setelah selesai melakukan proses ekstraksi fitur dan mendapatkan dimensi fitur setiap citra, perlu dilakukan proses normalisasi fitur untuk menyamakan skala antar fitur agar tidak terdapat fitur yang mendominasi selama proses pelatihan hanya karena memiliki nilai numerik yang lebih besar. Teknik normalisasi yang digunakan pada penelitian ini adalah Standardisasi dengan menggunakan pustaka StandardScaler dari scikit-learn. Standardisasi merupakan proses mengubah setiap fitur agar memiliki rata-rata (*mean*) adalah 0 dan standar deviasi adalah 1 dengan menggunakan rumus *Z-score normalization* berikut:

$$z = \frac{x - \mu}{\sigma} \tag{11}$$

Keterangan:
x : Nilai asli fitur
μ : Rata-rata seluruh nilai fitur.
σ : Standar deviasi dari nilai fitur.

2.6. Reduksi Dimensi

Dimensi fitur yang dihasilkan pada tahap ekstraksi fitur akan sangat besar dikarenakan akan menggabungkan hasil ektraksi fitur dari tiga metode berbeda yang dapat menyebabkan *overfitting* dan meningkatkan waktu komputasi. Oleh karena itu, dilakukan proses reduksi dimensi menggunakan *Principal Component Analysis* (PCA) untuk menurunkan kompleksitas data dan mempercepat proses pelatihan model tanpa menghilangkan informasi penting yang terdapat dalam citra. PCA merupakan metode mengekstraksi fitur penting dari kumpulan fitur citra dengan memisahkan atau menguraikan citra untuk menghasilkan koefisien yang tidak relevan dan menghilangkannya [9]. Penelitian ini menggunakan metode PCA untuk mengubah data berdimensi tinggi ke dimensi yang lebih rendah dengan memaksimalkan variansi data pada sumbu-sumbu baru. Pengimplementasian metode PCA dilakukan dengan menggunakan pustaka scikit-learn dengan proses normalisasi fitur terlebih dahulu menggunakan StandardScaler.

2.7. Pembagian Dataset

Setiap citra memiliki representasi vektor fitur nya masing-masing yang diperoleh pada tahap ekstraksi fitur. Vektor fitur tersebut wajib dinormalisasi terlebih dahulu sebelum melakukan pembagian dataset. Pembagian dataset dilakukan pada data citra yang sudah direduksi dimensi dengan algoritma PCA. Seluruh vektor fitur citra dikumpulkan ke dalam sebuah array dua dimensi bernama x, sedangkan label kelas jamur (beracun atau tidak beracun) disimpan dalam array satu dimensi y. Kedua array ini kemudian digunakan sebagai input untuk proses pembagian dataset. Pembagian dataset dilakukan dengan rasio 80% data *training* dan 20% data *testing*. Digunakan juga strategi stratified sampling untuk menjaga proporsi antara citra kelas beracun dan tidak beracun tetap seimbang.

2.8. Proses Training Model

Model SVM yang digunakan dalam penelitian ini berasal dari pustaka scikit-learn dengan kelas SVC (*Support Vector Classification*). Selama proses pelatihan menggunakan data training, model SVM hanya menggunakan satu parameter kernel saja, yaitu kernel RBF (*Radial Basis Function*) karena kernel ini mampu menangani data non-linear secara efektif. Untuk mengoptimalkan performa model, dilakukan *tuning hyperparameter* dengan mengatur berbagai nilai untuk parameter C dan gamma. Terdapat enam nilai berbeda yang dicoba untuk parameter C, yaitu 0.01, 0.1, 1, 10, 100, dan 1000, serta empat nilai untuk gamma, yaitu 'scale', 'auto', 0.01, dan 0.1. Proses *tuning hyperparameter* ini dilakukan dengan menggunakan GridSearchCV, yang secara otomatis mencari kombinasi terbaik antara C dan gamma untuk memperoleh akurasi model yang paling baik. Untuk meningkatkan kemampuan generalisasi model dan menghindari adanya *overfitting*, digunakan metode *K-Fold Cross Validation*, khususnya 5-Fold Cross Validation yang terintegrasi dengan GridSearchCV. Proses ini membagi data *training* menjadi 5 bagian (fold), di mana model dilatih pada empat bagian dan divalidasi pada satu bagian secara bergiliran, kemudian dirata-ratakan hasilnya untuk menentukan kinerja terbaik. Terdapat 120 model dengan parameter dan fold berbeda yang dilatih pada penelitian ini.

2.9. Evaluasi Model

Evaluasi model merupakan tahapan untuk mengukur kinerja dari model SVM yang sudah dilatih untuk melakukan klasifikasi citra. Kinerja model dievaluasi menggunakan data *testing* yang sudah dipersiapkan sebelumnya. Metode evaluasi yang digunakan adalah *confusion matriks* yang menunjukkan jumlah prediksi benar dan salah dari masing-masing kelas serta label masing-masing kelas yang sebenarnya. Dengan menggunakan metode *confusion matriks*, dapat diperoleh informasi mengenai performa model SVM secara menyeluruh, seperti nilai akurasi untuk mengukur seberapa banyak prediksi model yang benar dibandingkan dengan semua data yang diuji, *precision* untuk mengukur berapa banyak dari prediksi positif model yang benar-benar positif*, recall* untuk mengukur seberapa banyak dari data yang benar-benar positif berhasil diklasifikasikan oleh model*, dan F1-score* yang merupakan rata-rata harmonis dari precision dan recall berdasarkan perhitungan berikut:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

$$F1-Score = 2.\frac{Precision\ .\ Recall}{Precision+Recall} \quad (15)$$

Keterangan:

$TP\ (True\ Positive)$: Jumlah citra kelas beracun yang berhasil diklasifikasikan dengan benar sebagai citra jamur beracun.

$TN\ (True\ Negative)$: Jumlah citra kelas tidak beracun yang berhasil diklasifikasikan dengan benar citra jamur tidak beracun.

$FP\ (False\ Positive)$: Jumlah citra kelas beracun yang salah diklasifikasikan sebagai citra jamur tidak beracun.

$FN\ (False\ Negative)$: Jumlah citra kelas tidak beracun yang salah diklasifikasikan sebagai citra jamur beracun.

3. Hasil dan Diskusi

Penelitian ini mengembangkan sistem klasifikasi jamur beracun atau tidak dengan memanfaatkan ekstraksi fitur hibrida yaitu histogram warna, GLCM, serta HOG dan algoritma SVM untuk memisahkan data berdasarkan warna, tekstur dan bentuk dari citra tersebut. Penelitian ini menggunakan bahasa pemrograman python beserta dengan sejumlah pustakanya seperti os, cv2, numpy, sklearn (scikit-learn), skimage (scikit-image), seaborn dan matplotlib.pyplot untuk membangun model *machine learning*. Hasil dari penelitian ini akan ditulis dalam bentuk sub bab pada bagian hasil dan diskusi.

3.1. Hasil Ekstraksi Fitur

Proses ekstraksi fitur hibrida dengan menggabungkan histogram warna, *Gray Level Co-occurrence Matrix* (GLCM), dan *Histogram of Oriented Gradients* (HOG) dapat menghasilkan total 2.282 fitur untuk setiap citra jamur. Ekstraksi fitur dengan histogram warna menghasilkan 512 fitur untuk setiap citra, dengan rincian 8 bin untuk masing-masing kanal warna RGB. Fitur yang dihasilkan dari ekstraksi fitur histogram warna ini mencerminkan distribusi warna secara menyeluruh pada citra dan mampu membedakan citra jamur berdasarkan intensitas dan komposisi warna yang berbeda satu sama lain dengan menghasilkan tiga fitur statistik utama, yaitu *mean, variance*, dan *median*. Nilai *mean* menggambarkan rata-rata intensitas warna pada keseluruhan gambar, *variance* mencerminkan sebaran atau keragaman warna, dan *median* merepresentasikan nilai tengah dari distribusi warna. Berikut ini hasil dari ekstraksi fitur Histogram warna pada beberapa citra jamur yang dapat dilihat pada tabel 2.

Tabel 2. Contoh Beberapa Hasil Ekstraksi Fitur Histogram Warna

Gambar	Kelas	*Mean*	*Variance*	*Median*
Amanita_smithiana_000024_1.jpg	Beracun	133.904968	4231.500551	130.0
Galerina_marginata_000023_1.jpg	Beracun	97.657531	1420.861414	93.0
Tremella_fuciformis_000019_0.jpg	Tidak Beracun	73.032409	4163.728869	45.0
Truffles_000027_1.jpg	Tidak Beracun	106.471984	7877.725167	90.0

Ekstraksi fitur GLCM digunakan untuk menangkap tekstur dari citra dengan menghasilkan 6 fitur statistik, yakni *contrast, dissimilarity, homogeneity, energy, correlation, dan ASM.* Berikut ini hasil dari ekstraksi fitur GLCM pada beberapa citra jamur yang dapat dilihat pada tabel 3.

Tabel 3. Contoh Beberapa Hasil Ekstraksi Fitur GLCM

Gambar	*Contrast*	*Dissimilarity*	*Homogeneity*	*ASM*	*Energy*	*Correlation*
Amanita_smithiana_000024_1.jpg	519.179	10.4157	0.3039	0.0386	0.9379	0.001496
Galerina_marginata_000023_1.jpg	236.1748	9.3276	0.1745	0.021	0.8720	0.000443
Tremella_fuciformis_000019_0.jpg	864.087	18.5384	0.1016	0.016	0.8831	0.000272
Truffles_000027_1.jpg	1019.883	14.0108	0.4428	0.146	0.9336	0.021371

Sementara itu, HOG digunakan untuk mengekstraksi fitur bentuk dari tepi dan gradien arah pada citra. Hasil dari ektraksi fitur HOG adalah 1764 fitur yang merepresentasikan struktur visual dan siluet dari objek jamur dan akan digunakan untuk proses training model. Tujuan penggabungan fitur dari ketiga metode ini adalah untuk memperoleh representasi fitur citra jamur yang lebih menyeluruh dalam aspek warna, tekstur, dan bentuk morfologinya.

3.2. Hasil Pelatihan Model

Setelah proses pelatihan model selesai melakukan *tuning hyperparameter* menggunakan GridSearchCV dan *5-Fold Cross Validation*, didapatkan bahwa model terbaik adalah model dengan kombinasi parameter C bernilai 10 dan gamma bernilai scale dengan mencapai akurasi tertinggi model sebesar 78.63% pada data training. Hasil dari pelatihan model terbaik SVM dengan melakukan reduksi dimensi menggunakan metode PCA menunjukan adanya peningkatan efesiensi waktu dalam proses pelatihan. Hal ini disebabkan karena metode PCA dapat mereduksi jumlah fitur dari 2.282 fitur menjadi 454 fitur utama, dengan mempertahankan 95% variansi data untuk mengeliminasi fitur-fitur yang redundan atau tidak relevan. Untuk rincian akurasi model terbaik pada masing-masing fold setelah melakukan proses pelatihan dengan menerapkan metode reduksi dimensi PCA dapat dilihat pada tabel 4.

Tabel 4. Hasil Akurasi *5-Fold Cross Validation* Pada Model Terbaik

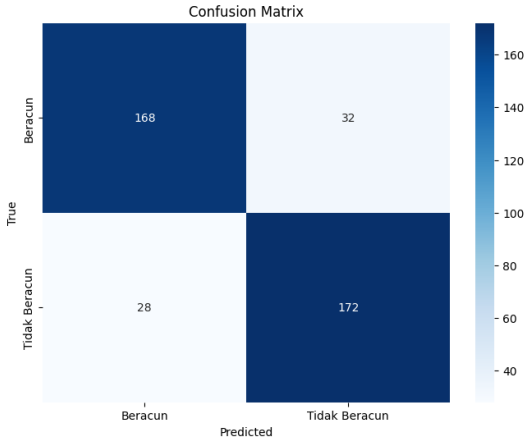
5-Fold Cross Validation	Akurasi
Fold 1	0.7719
Fold 2	0.8000
Fold 3	0.7625
Fold 4	0.8094
Fold 5	0.7875

Pada tabel 4 menampilkan hasil pengujian performa model terbaik menggunakan metode *5-Fold Cross Validation*, yang bertujuan untuk mengevaluasi stabilitas akurasi modelnya. Melalui lima kali pengulangan pengujian pada *fold* data yang berbeda, model ini berhasil mencatatkan akurasi yang bervariasi yaitu 77.19% pada *fold* 1, 80% pada *fold* 2, 76.25% pada *fold* 3, 80.94% pada *fold* 4, dan 78.75% pada *fold* 5. Hasil ini menunjukkan bahwa performa model yang relatif stabil di setiap *fold* dan model dapat mencapai akurasi rata-rata *cross-validation* terbaik di angka 78.63% yang mengindikasikan kemampuan generalisasi yang baik terhadap data pelatihan.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

3.3. Hasil Evaluasi Model

Hasil evaluasi model SVM dengan menggunakan PCA pada data *testing* divisualisasikan menggunakan *confusion matriks* untuk menampilkan informasi jumlah hasil klasifikasi citra jamur dari model terhadap kelas citra yang sesungguhnya. Untuk gambar *confusion matriks* dapat dilhat pada pada gambar 6.



Gambar 6. Confusion Matriks Model SVM dengan PCA

Confusion matrix pada gambar 6 menjelaskan bahwa model dapat mengklasifikasikan 168 citra jamur beracun dengan benar (*True Positive*) dan 172 citra jamur tidak beracun dengan benar (*True Negatif*) dari 200 sampel citra jamur dari masing masing kelas, yaitu beracun dan tidak beracun yang terdapat pada data *testing*. Model masih belum seutuhnya sempurna karena terdapat beberapa kesalahan dalam melakukan pengklasifikasian citra jamur, seperti 32 citra jamur yang sebenarnya beracun yang diklasifikasikan oleh model menjadi jamur tidak beracun (*False Negative*) dan 28 citra jamur yang sebenarnya tidak beracun yang diklasifikasikan oleh model menjadi jamur beracun (*False Positive*).

Informasi dari *confusion matriks* juga digunakan untuk menghitung nilai akurasi dari model yang telah dilatih dengan menggunakan data *training* dan menghitung nilai metrik evaluasi laiinnya seperti precision, recall dan f-1 score untuk masing-masing kelas.Hasil dari Evaluasi model menunjukkan bahwa model ini sudah mampu mengklasifikasikan citra jamur sesuai dengan kelasnya yaitu beracun atau tidak beracun dengan akurasi yang cukup tinggi hingga mencapai akurasi sebesar 0.85 atau 85%. Untuk informasi metrik evaluasinya dapat dilihat pada tabel 5.

Tabel 5. Hasil Evaluasi Metrik

Kelas	*Precision*	*Recall*	*F1-Score*	*Support*
Beracun	0.86	0.84	0.85	200
Tidak Beracun	0.84	0.86	0.85	200

Pada tabel 5 mengandung informasi bahwa kelas beracun menghasilkan nilai precision sebesar 0.86 yang berarti 86% citra yang benar-benar beracun dari semua citra jamur yang diprediksi sebagai beracun oleh model, nilai recall sebesar 0.84 yang berarti model berhasil mendeteksi 84% dari total citra jamur yang memang beracun dan nilai f1-score sebesar 0.85 sedangkan untuk metrik evaluasi pada kelas tidak beracun menghasilkan nilai precision sebesar 0.84 yang berarti 84% citra yang benar-benar tidak beracun dari semua citra jamur yang diprediksi sebagai tidak beracun oleh model, nilai recall sebesar 0.86 yang berarti model berhasil mendeteksi 86% dari total citra jamur yang memang tidak beracun dan nilai f1-score sebesar 0.85. Nilai f1-score untuk

kedua kelas sama yaitu 85% yang menandakan bahwa model bekerja seimbang dalam mengenali kedua kelas tersebut, baik dalam hal ketepatan maupun kelengkapan.

4. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan, Penelitian ini sudah berhasil untuk membangun suatu model *machine learning* dengan menggunakan *Support Vektor Machine* (SVM) berbasis ekstraksi fitur hibrida yang sudah direduksi dimensi dengan metode *Principal Component Analysis* (PCA) untuk melakukan klasifikasi citra jamur beracun atau tidak. Dengan dilakukannya reduksi dimensi fitur menggunakan PCA, model *machine learning* yang dibangun telah mengalami peningkatan dalam segi kecepatan waktu untuk melakukan pelatihan model dibandingkan tanpa menggunakan PCA. Hasil evaluasi model pada data *testing* mendapatkan akurasi 85% untuk model yang telah dibangun. Hal ini menunjukan bahwa model sudah dapat mengenali dan mengklasifikasikan jamur ke dalam 2 kelas berbeda, yaitu beracun dan tidak beracun. Dengan adanya penelitian ini diharapkan dapat membantu masyarakat awam untuk membedakan jamur beracun dan tidak beracun melalui bentuk visual nya, tetapi model yang telah dibangun tidak dapat dijadikan satu-satunya acuan untuk mengambil keputusan bahwa jamur tersebut dapat dimakan atau beracun karena model masih terdapat beberapa salah pengklasifikasian citra jamur yang akan berbahaya jika jamur beracun salah pengklasifikasian. Maka dari itu, saran untuk penelitian selanjutnya dapat membangun model yang lebih canggih dengan menggunakan algoritma *deep learning* seperti CNN agar dapat mencapai akurasi peforma model terbaik.

Daftar Pustaka

[1] G. M. C. Batubara, A. Desiani, and A. Amran, “Klasifikasi Jamur Beracun Menggunakan Algoritma Naïve Bayes dan K-Nearest Neighbors,” *Jurnal Ilmu Komputer dan Informatika*, vol. 3, no. 1, pp. 33–42, Jun. 2023, doi: https://doi.org/10.54082/jiki.68 .

[2] I. P. Putra, “KASUS-KASUS KERACUNAN JAMUR LIAR DI INDONESIA,” *JURNAL EKOLOGI KESEHATAN*, vol. 20, no. 3, pp. 215–230, Mar. 2022, doi: https://doi.org/10.22435/jek.v20i3.4943 .

[3] A. Dedy, “Klasifikasi Jenis Jamur Berdasarkan Citra Gambar Menggunakan Algoritma Stochastic Gradient Descent,” *Data Sciences Indonesia (DSI)*, vol. 4, no. 2, pp. 1–9, Dec. 2024, doi: https://jurnal.itscience.org/index.php/dsi/article/view/5014 .

[4] S. Talib, S. Sudin, and M. D. Suratin, “PENERAPAN METODE SUPPORT VECTOR MACHINE (SVM) PADA KLASIFIKASI JENIS CENGKEH BERDASARKAN FITUR TEKSTUR DAUN”, Accessed: Jun. 30, 2025. [Online]. Available: https://doi.org/10.30656/prosisko.v11i1.7911 .

[5] N. L. Chusna, M. I. Shalahudin, U. Riyanto, and A. D. Alexander, “Klasifikasi Citra Jenis Tanaman Jamur Layak Konsumsi Menggunakan Algoritma Multiclass Support Vector Machine,” *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 1, Jun. 2022, doi: https://doi.org/10.47065/bits.v4i1.1624 .

[6] M. G. T. Akbar and H. Leidiyana, “Klasifikasi pada Citra Bunga dengan Ekstraksi Fitur Color Histogram,” 2023. Accessed: Jul. 01, 2025. [Online]. Available: http://dx.doi.org/10.22441/format.2023.v12.i1.007 .

[7] S. A. Rosiva Srg, M. Zarlis, and Wanayumini, “Identifikasi Citra Daun dengan GLCM (Gray Level Co-Occurence) dan K-NN (K-Nearest Neighbor),” *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 2, pp. 477–488, Mar. 2022, doi: https://doi.org/10.30812/matrik.v21i2.1572 .

[8] S. Amraee, M. Chinipardaz, and M. Charoosaei, “Analytical study of two feature extraction methods in comparison with deep learning methods for classification of small metal objects,” *Vis Comput Ind Biomed Art*, vol. 5, no. 1, Dec. 2022, Available: https://vciba.springeropen.com/articles/10.1186/s42492-022-00111-6 .

[9] K. Nurfitri, D. A. Pradana, and I. Widaningrum, “Jurnal Rekayasa Teknologi dan Komputasi PENERAPAN ALGORITMA PRINCIPAL COMPONENT ANALYSIS (PCA) DAN K-NEAREST NEIGHBORS (KNN) PADA KLASIFIKASI”, Accessed: Jun. 30, 2025. [Online]. Available: https://doi.org/10.24269/jrtekom.v1i1.6454 .