

KLASIFIKASI EDIBILITAS JAMUR SECARA OTOMATIS MENGGUNAKAN ALGORITMA RANDOM FOREST BERBASIS MORFOLOGI

Muhammad Reza Pahlevi¹, Imam Tahyudin², Ades Tikaningsih³

^{1,2} Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Amikom Purwokerto
Jl. Let. Jend. Pol. Soemarto no. 126 Purwokerto Utara Banyumas

¹rezam.pahlevi963@gmail.com

²imam.tahyudin@dosen.amikompurwokerto.ac.id

³adestikaningsih92@gmail.com

Abstract

Mushrooms exhibit high morphological diversity; however, some species are poisonous and harmful if consumed. The physical similarities between edible and poisonous mushrooms often complicate manual identification. This research aims to develop an automated mushroom classification system based on the Random Forest algorithm to distinguish between edible and poisonous mushrooms. It leverages all 21 attributes (18 categorical, 3 numerical) from a comprehensive Kaggle dataset comprising 61,069 entries. The research methodology follows a modified CRISP-DM workflow, from data collection through to implementation. Crucial data preprocessing steps were extensively performed, including handling data duplicates and imputing missing values (using median and mode). Subsequently, class labels were transformed (edible=0, poisonous=1), and One-Hot Encoding was applied to categorical features for appropriate numerical representation. Numerical features such as cap-diameter and stem-height were normalized using Standard Scaling to balance their contributions. The data was then split 80:20 for training and testing. The Random Forest model was developed with optimal parameters (n_estimators=200, max_depth=15, class_weight="balanced") for efficiency and robustness against class imbalance. Evaluation results demonstrated excellent performance with an overall accuracy of 99.34%, along with balanced precision, recall, and F1-score values of 0.99 for both classes. Feature importance analysis identified stem-width, stem-height, and cap-diameter as the most influential attributes. The learning curve indicated model stability without significant overfitting. Implementation on new mushroom samples also confirmed consistent predictive capability, making this model suitable as an automated decision support system for detecting poisonous mushrooms.

Keywords: Musrom, Clasification, Random Forest, Morphology, Machine Learning

I. PENDAHULUAN

Jamur merupakan organisme eukariotik yang memiliki keragaman spesies yang sangat luas, dengan estimasi mencapai 1,5 juta jenis di seluruh dunia. Namun, hanya sekitar 74.000 jenis yang telah teridentifikasi, sehingga pengetahuan mengenai klasifikasi jamur masih terbatas[1]. Beberapa jenis jamur diketahui memiliki kandungan racun yang dapat membahayakan kesehatan manusia jika dikonsumsi secara tidak tepat. Kesulitan dalam membedakan antara jamur yang dapat dikonsumsi dan yang beracun seringkali disebabkan oleh kemiripan morfologi di antara spesies-spesies tersebut (ResearchGate)[2].

Kesalahan dalam mengidentifikasi jamur dapat berakibat fatal, karena beberapa jamur beracun mengandung toksin yang dapat menyebabkan gejala mulai dari gangguan gastrointestinal ringan hingga kematian[3]. Oleh karena itu diperlukan, metode yang efektif dan akurat untuk mendeteksi jamur berbahaya. Perkembangan teknologi dalam bidang pembelajaran mesin (machine learning) menawarkan solusi potensial untuk permasalahan ini. Salah satu algoritma yang menonjol dalam klasifikasi adalah Random Forest, yang dikenal karena kemampuannya dalam menangani dataset

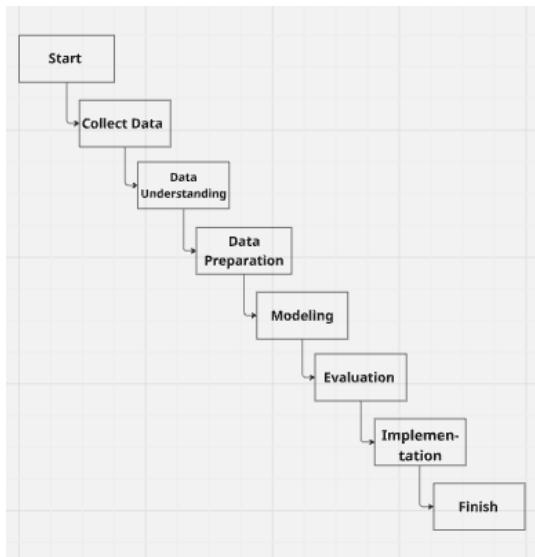
yang besar dan kompleks serta memberikan akurasi yang tinggi dalam berbagai aplikasi klasifikasi (Politeknik Pratama).

Penelitian sebelumnya telah menunjukkan efektivitas berbagai algoritma dalam klasifikasi jamur. Misalnya, algoritma Naïve Bayes dan k-Nearest Neighbors (kNN) telah diterapkan untuk mengklasifikasikan jamur berdasarkan ciri-cirinya, dengan kedua algoritma mencapai akurasi lebih dari 90%[4]. Salah satu penelitian CNN untuk mengklasifikasikan jamur berdasarkan genus (dengan karakteristik morfologi) mencapai akurasi 89% untuk pelatihan dan 82% untuk validasi pada 1200 data gambar, menggunakan tiga *convolution layer*, *MaxPooling layer*, dan *dropout layer* untuk mengurangi *overfitting*[5].

Melihat potensi besar dan keunggulan Random Forest dalam menangani kompleksitas data, penelitian ini berfokus pada eksplorasi mendalam efektivitas algoritma tersebut. Dengan demikian, penelitian ini bertujuan untuk mengeksplorasi efektivitas Random Forest dalam mengklasifikasikan jamur beracun dan tidak beracun.

II. METODOLOGI PENELITIAN

Metode yang digunakan dalam penelitian ini mengacu pada alur proses CRISP-DM (Cross Industry Standard Process for Data Mining) yang dimodifikasi agar sesuai dengan kebutuhan klasifikasi jamur menggunakan algoritma Random Forest[6]. Proses ini terdiri dari beberapa tahapan utama, yaitu: *Collect Data, Data Understanding, Data Preparation, Modeling, Evaluation, Implementation, dan Finish*[7].



Gambar 1. Alur Metode Penelitian

A. Collect Data

Collect Data yaitu teknik cara-cara yang dapat digunakan oleh peneliti untuk pengumpulan data[8]. Dalam penelitian ini, data diperoleh dari Kaggle pada tautan berikut ini <https://www.kaggle.com/datasets/devzohaib/mushroom-edibility-classification>. Dataset tersebut berisi 61.069 data jamur dengan 20 atribut seperti bentuk, warna, permukaan, serta label klasifikasi ‘*edible*’ atau ‘*poisonous*’

B. Data Understanding

Data Understanding atau bisa disebut dengan EDA yaitu salah satu strategi untuk melakukan analisis data sehingga mempermudah untuk melakukan eksplorasi secara mendalam sebelum leakukan pemodelan[9]. Dalam dataset ini terdapat 61.069 data dengan detail data dapat dilihat pada tabel 1. Dataset ini memiliki 21 atribut dengan 18 data kategorikal dan 3 data numerikal. Data kategorikal sendiri merupakan data yang skala pengukurannya terdiri dari sekumpulan kategorik ordinal atau nominal[10]. Data numerikal adalah data kuantitatif yang nilainya berbentuk angka dan dapat diukur atau dihitung mencakup variabel yang nilainya berupa angka, seperti tinggi badan, berat badan, atau usia[11]. Detail informasi atribut data terpadat pada tabel 2

Tabel 1. Sample Data

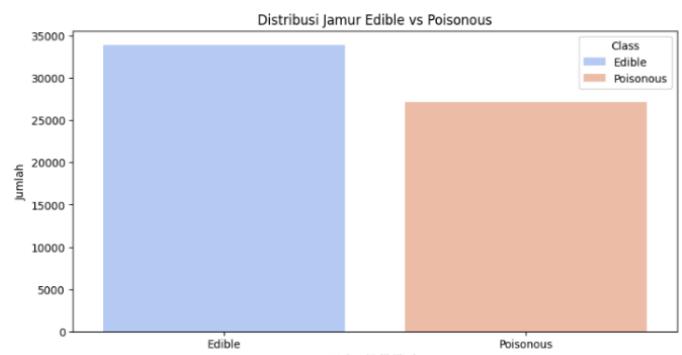
| | Class | Cap-Diameter | Cap-Shape | Cap-surface | Cap-color | |
|---|-------|--------------|-----------|-------------|-----------|------|
| 0 | p | 15.26 | x | g | o | |
| 1 | p | 16.60 | x | g | o | |
| 2 | p | 14.07 | x | g | o | |

| | | | | | | |
|---|---|-------|---|---|---|------|
| 3 | p | 14.17 | f | h | e | |
| 4 | p | 14.64 | x | h | o | |

Tabel 2. Attribute Data

| # | Column | Non-Null Count | Dtype |
|----|----------------------|----------------|---------|
| 0 | class | 61069 non-null | object |
| 1 | cap-diameter | 61069 non-null | float64 |
| 2 | cap-shape | 61069 non-null | object |
| 3 | cap-surface | 46949 non-null | object |
| 4 | cap-color | 61069 non-null | object |
| 5 | does-bruise-or-bleed | 61069 non-null | object |
| 6 | gill-attachment | 51185 non-null | object |
| 7 | gill-spacing | 36006 non-null | object |
| 8 | gill-color | 61069 non-null | object |
| 9 | stem-height | 61069 non-null | float64 |
| 10 | stem-width | 61069 non-null | float64 |
| 11 | stem-root | 9531 non-null | object |
| 12 | stem-surface | 22945 non-null | object |
| 13 | stem-color | 61069 non-null | object |
| 14 | veil-type | 3177 non-null | object |
| 15 | veil-color | 7413 non-null | object |
| 16 | has-ring | 61069 non-null | object |
| 17 | ring-type | 58598 non-null | object |
| 18 | spore-print-color | 6354 non-null | object |
| 19 | habitat | 61069 non-null | object |
| 20 | season | 61069 non-null | object |

Untuk distribusi kelas nya disini terdapat kelas edible (jamur yang bisa dimakan) dan kelas poisonus(jamur yang beracun) adalah 27181 untuk jamur edible dan 33888 untuk jamur poisonus.



Gambar 2. Distribusi Kelas

C. Data Preparation

Data preprocessing adalah tahap krusial dalam analisis data yang bertujuan untuk mengubah data mentah menjadi format yang sesuai dan berkualitas untuk analisis lebih lanjut. Proses ini mencakup berbagai teknik seperti pembersihan data (data cleaning), transformasi data, normalisasi, dan reduksi dimensi. Menurut Fan et al. data preprocessing melibatkan serangkaian teknik untuk meningkatkan kualitas data mentah, seperti penghapusan outlier dan imputasi nilai yang hilang, yang esensial untuk memastikan keandalan analisis data operasional bangunan[12].

Pentingnya data preprocessing terletak pada kemampuannya untuk meningkatkan akurasi dan efisiensi model analisis data. Data yang tidak diproses dengan baik dapat menyebabkan hasil analisis yang bias atau tidak akurat. Teknik data preprocessing memiliki pengaruh signifikan dalam pengembangan dan penerapan model machine learning, khususnya dalam konteks aplikasi pendidikan[13].

Dalam dataset ini awalnya terlihat ada banyak duplikat dan data kosong didalamnya yang dapat dilihat pada gambar 3 dan 4, sehingga dibutuhkan pembersihan data atau data cleaning.

```
## Cek duplikasi
print("\nJumlah baris duplikat:", df.duplicated().sum())

Jumlah baris duplikat: 146
```

Gambar 3. Cek Duplikasi

```
## Cek jumlah missing values di setiap kolom
print(df.isnull().sum())

class 0
cap-diameter 0
cap-shape 0
cap-surface 14120
cap-color 0
does-bruise-or-bleed 0
gill-attachment 9884
gill-spacing 25063
gill-color 0
stem-height 0
stem-width 0
stem-root 51538
stem-surface 38124
stem-color 0
veil-type 57892
veil-color 53656
has-ring 0
ring-type 2471
spore-print-color 54715
habitat 0
season 0
dtype: int64
```

Gambar 4. Cek Missing Value

Untuk mengatasi duplikasi data kita dapat menghapus kolom yang samanya. Untuk mengatasi missing value kita dapat menggunakan metode imputasi sederhana dengan menggunakan mean, median, dan modus untuk mengisi data yang kosong dan pada kasus ini kita gunakan median dan modus saja[14]. Pemilihan modus untuk fitur kategorikal dilakukan karena nilai-nilai ini tidak memiliki urutan numerik dan modus merepresentasikan kategori yang paling sering muncul, menjamin integritas distribusi data. Sementara itu, median dipilih untuk fitur numerik karena median kurang sensitif terhadap outlier dibandingkan mean, sehingga memberikan representasi nilai tengah yang lebih robust pada distribusi data yang mungkin skewed.

| Deskripsi | Nilai |
|---|-------|
| Jumlah baris setelah menghapus duplikat | 60923 |
| Jumlah baris duplikat setelah dihapus | 0 |

Gambar 5. Hasil Penanganan Duplikasi Data

```
class 0
cap-diameter 0
cap-shape 0
cap-surface 0
cap-color 0
does-bruise-or-bleed 0
gill-attachment 0
gill-spacing 0
gill-color 0
stem-height 0
stem-width 0
stem-root 0
stem-surface 0
stem-color 0
veil-type 0
veil-color 0
has-ring 0
ring-type 0
spore-print-color 0
habitat 0
season 0
dtype: int64
```

Gambar 6. Hasil Penanganan Missing Value

Langkah selanjutnya adalah transformasi label, Transformasi label kelas dilakukan dengan mengubah nilai kategorikal pada kolom class, dari 'e' (edible) menjadi 0 dan 'p' (poisonous) menjadi 1. Tujuan dari transformasi ini adalah untuk mengonversi label target ke dalam bentuk numerik agar dapat diproses oleh algoritma machine learning, khususnya pada tahap pelatihan dan evaluasi model klasifikasi.

```
[51] # Ubah label kelas (e=0, p=1)
df['class'] = df['class'].map({'e': 0, 'p': 1})

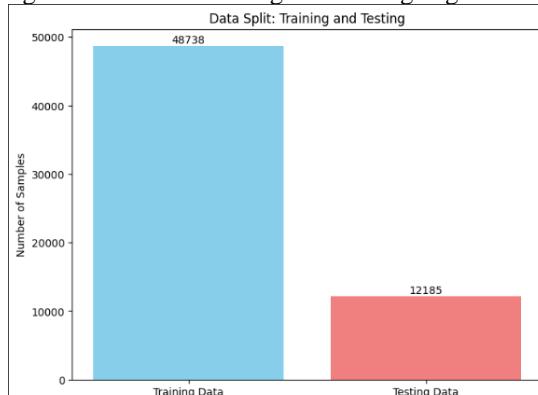
edible (e) diubah menjadi 0 poisonus (p) diubah menjadi 1
```

Gambar 7. Konversi Target ke Numerik

Setelah dilakukan konversi label pada fitur target class menjadi nilai numerik (e = 0 dan p = 1), tahap selanjutnya adalah melakukan encoding terhadap fitur-fitur kategorikal dalam dataset. Setelah transformasi label, seluruh fitur kategorikal dalam dataset diubah menjadi representasi numerik menggunakan *One-Hot Encoding* dengan fungsi pd.get_dummies(df, drop_first=True) dari pustaka Pandas. Proses ini mengonversi setiap nilai unik pada fitur kategorikal (misalnya, cap-shape, cap-color, habitat) menjadi kolom biner (0 atau 1)[15]. Penggunaan drop_first=True sangat penting untuk menghindari masalah multikolinearitas, di mana satu kolom *dummy* dapat diprediksi dari kombinasi kolom *dummy* lainnya, sehingga mencegah redundansi informasi dan memastikan model belajar secara efisien dari setiap atribut.

Langkah selanjutnya adalah normalisasi fitur numerik (cap-diameter dan stem-height) menggunakan *StandardScaler* dari pustaka Scikit-Learn. Pemilihan *StandardScaler* ini adalah karena fitur-fitur tersebut memiliki skala yang berbeda dan seringkali memiliki distribusi data yang mendekati normal. Hal ini bertujuan untuk mencegah dominasi fitur tertentu dalam model karena perbedaan skala[16]. *StandardScaler* mengubah data sehingga memiliki rata-rata nol (mean=0) dan varians satu (std=1), yang memastikan bahwa fitur-fitur ini berkontribusi secara proporsional terhadap kinerja model tanpa didominasi oleh fitur dengan rentang nilai yang lebih besar. Ini esensial untuk model berbasis jarak atau gradien seperti beberapa komponen dalam Random Forest yang mungkin terpengaruh oleh skala data. Hasil transformasi kemudian ditimpak kembali ke kolom aslinya untuk memastikan data siap digunakan dalam proses pemodelan.

Split data, pada proses ini saya melakukan split data dengan perbandingan 20:80, dengan data testing 20% dan data training. Untuk detail data split terdapat 48738 data training dan 12185 data testing sesuai dengan gambar 8.



Gambar 8. Data Training Testing

D. Modeling

Lanjut pada tahap Modeling, Pada tahap ini dilakukan proses pembentukan model klasifikasi menggunakan algoritma Random Forest. Random Forest merupakan algoritma *ensemble learning* yang menggabungkan sejumlah pohon keputusan secara paralel, dan hasil akhir klasifikasi ditentukan melalui metode *majoritas suara* (*majority voting*), di mana setiap pohon memberikan kontribusi terhadap keputusan akhir[17]. Algoritma ini dipilih karena kemampuannya dalam menangani data berukuran besar, tahan terhadap overfitting, dan efektif dalam memproses data dengan fitur-fitur kategorikal maupun numerik.

Pada penelitian ini, model dibangun menggunakan pustaka scikit-learn dengan class RandomForestClassifier. Parameter-parameter pada model diatur sedemikian rupa untuk mengoptimalkan kinerja klasifikasi. Pemilihan parameter-parameter ini dilakukan berdasarkan serangkaian eksperimen dan *fine-tuning* untuk mendapatkan kinerja optimal pada dataset ini. Meskipun tidak dijelaskan secara rinci proses *tuning* otomatis seperti *grid search* atau *random search*, nilai-nilai yang dipilih (*n_estimators*=200, *max_depth*=15, *min_samples_split*=5, *min_samples_leaf*=2) merupakan hasil eksplorasi empiris yang ditemukan memberikan keseimbangan terbaik antara akurasi tinggi dan pencegahan *overfitting*. Khususnya, *n_estimators*=200 memberikan stabilitas prediksi yang memadai tanpa menambah beban komputasi yang berlebihan; *max_depth*=15 membatasi kompleksitas pohon untuk mengurangi risiko *overfitting* sambil mempertahankan kemampuan generalisasi yang kuat. Penggunaan *class_weight*="balanced" sangat krusial untuk mengatasi sedikit ketidakseimbangan kelas dalam dataset, memastikan bahwa model tidak bias terhadap kelas mayoritas.

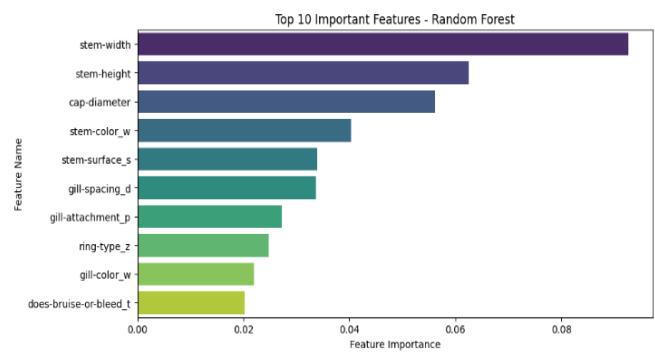
```
from sklearn.ensemble import RandomForestClassifier
# • Buat model Random Forest sesuai spesifikasi terbaru
rf_model = RandomForestClassifier(
    n_estimators=200,
    max_depth=15,
    min_samples_split=5,
    min_samples_leaf=2,
    class_weight="balanced",
    random_state=42
)
```

Gambar 9. Modeing Random Forest

Setelah model Random Forest berhasil dilatih, dilakukan analisis terhadap feature importance guna mengidentifikasi fitur-fitur yang paling berpengaruh terhadap proses klasifikasi. Visualisasi pada Gambar 10 memperlihatkan sepuluh fitur teratas yang memberikan kontribusi terbesar dalam membedakan antara jamur yang dapat dikonsumsi dan jamur yang beracun.

Dari grafik tersebut, dapat dilihat bahwa fitur stem-width menjadi fitur yang paling dominan, diikuti oleh stem-height, cap-diameter, dan stem-color_w. Ketiga fitur utama tersebut merupakan fitur numerik yang mencerminkan dimensi fisik dari jamur, yang ternyata berperan besar dalam keputusan model. Fitur-fitur kategorikal seperti stem-surface_s, gill-spacing_d, gill-attachment_p, serta ring-type_z juga memiliki kontribusi signifikan. Menariknya, fitur does-bruise-or-bleed_t yang mengindikasikan apakah jamur memar atau mengeluarkan cairan, juga masuk dalam daftar fitur penting meskipun nilainya lebih rendah dibanding fitur-fitur lainnya.

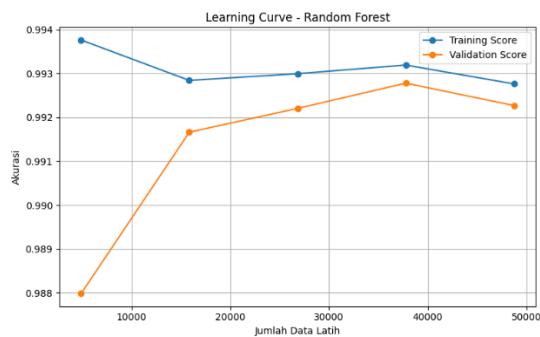
Informasi ini dapat dimanfaatkan untuk memahami logika model dalam membuat prediksi, serta menjadi bahan pertimbangan apabila dilakukan feature selection untuk meningkatkan efisiensi pemodelan. Dengan mengetahui fitur-fitur kunci, proses klasifikasi tidak hanya menjadi lebih transparan, tetapi juga dapat diarahkan untuk fokus pada atribut-atribut yang paling relevan secara biologis dan statistikal.



Gambar 10. Top 10 Fittur

Untuk memahami performa model terhadap variasi jumlah data latih, dilakukan analisis *learning curve*. Learning curve adalah representasi grafis yang memplot performa model (misalnya akurasi atau *error*) terhadap jumlah sampel data pelatihan. Tujuannya adalah untuk mendiagnosis apakah model mengalami *underfitting* atau *overfitting*, serta untuk mengevaluasi apakah penambahan data latih akan meningkatkan kinerja model secara signifikan. Seperti yang ditampilkan pada Gambar 11, berdasarkan grafik tersebut, akurasi model pada data pelatihan dan validasi cenderung stabil, menunjukkan bahwa model tidak mengalami *overfitting*.

signifikan. Seiring bertambahnya data latih, akurasi validasi meningkat hingga titik tertentu, mengindikasikan bahwa penambahan data memberikan dampak positif terhadap generalisasi model.



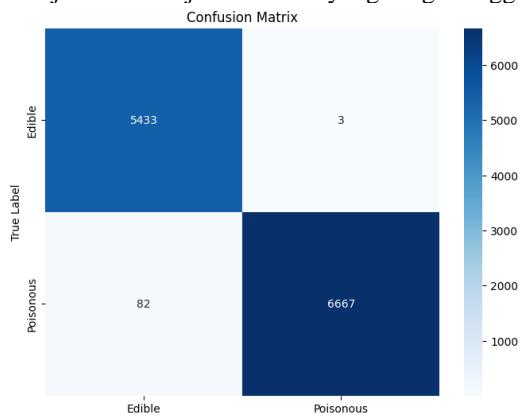
Gambar 11. Learning Curve

Kemudian dilakukan evaluasi awal untuk menilai performa model Random Forest terhadap data pengujian. Model menunjukkan hasil yang sangat baik dengan akurasi sebesar 99,34%, serta nilai precision, recall, dan f1-score yang seimbang pada kedua kelas. Hal ini mengindikasikan bahwa model mampu melakukan klasifikasi secara efektif, baik untuk kelas "edible" maupun "poisonous". Nilai f1-score yang tinggi pada kedua kelas juga menunjukkan keseimbangan antara presisi dan sensitivitas model.

III. HASIL DAN PEMBAHASAN

A. Evaluasi

Evaluasi model dilakukan untuk mengukur performa algoritma Random Forest dalam mengklasifikasikan jenis jamur ke dalam dua kategori, yaitu *edible* (dapat dimakan) dan *poisonous* (beracun). Proses pengujian dilakukan menggunakan data uji yang telah dipisahkan dari data pelatihan pada tahap praproses. Berdasarkan hasil pengujian, model mampu mencapai akurasi keseluruhan sebesar 99%, yang menunjukkan kinerja klasifikasi yang sangat tinggi.



Gambar 12. Confusion Matrix

Gambar 12 menyajikan *confusion matrix* yang memberikan gambaran rinci mengenai prediksi model. Dari total 5.436 sampel jamur yang tergolong *edible*, model hanya keliru memprediksi sebanyak 3 sampel sebagai *poisonous*. Sebaliknya, dari 6.749 sampel *poisonous*, terdapat 82 sampel yang diklasifikasikan secara salah sebagai *edible*. Meskipun

jumlah kesalahan relatif kecil, klasifikasi yang salah terhadap jamur beracun sebagai dapat dimakan tetap perlu menjadi perhatian karena berpotensi berbahaya dalam penerapan nyata.

Tabel 3. Evaluasi Random Forest

| Label | Precision | Recall | F1-Score | Support |
|----------------------|-----------|--------|----------|---------|
| 0 (Edible) | 0.99 | 1.00 | 0.99 | 5.436 |
| 1 (Poisonous) | 1.00 | 0.99 | 0.99 | 6.749 |
| Accuracy | | | 0.99 | 12.185 |
| Macro Avg | 0.99 | 0.99 | 0.99 | 12.185 |
| Weighted Avg | 0.99 | 0.99 | 0.99 | 12.185 |

Performa klasifikasi juga dianalisis menggunakan metrik evaluasi berupa *precision*, *recall*, dan *f1-score*, seperti ditunjukkan pada Tabel 3. Untuk kelas *edible*, *precision* sebesar 0.99, *recall* sebesar 1.00, dan *f1-score* sebesar 0.99, menunjukkan bahwa model sangat jarang salah dalam mengidentifikasi jamur dapat dimakan. Untuk kelas *poisonous*, *precision* mencapai 1.00, *recall* sebesar 0.99, dan *f1-score* sebesar 0.99, yang menunjukkan bahwa hampir seluruh jamur beracun berhasil diidentifikasi dengan tepat. Rata-rata makro maupun rata-rata berbobot dari ketiga metrik evaluasi tersebut sama-sama berada di angka 0.99, menandakan bahwa model memiliki performa yang seimbang dan konsisten pada kedua kelas.

Selain itu, untuk mengevaluasi kestabilan model terhadap peningkatan jumlah data latih, dilakukan analisis *learning curve* yang ditampilkan pada Gambar 11. Kurva menunjukkan bahwa akurasi model pada data pelatihan dan validasi berada pada tingkat yang tinggi dan relatif stabil. Selisih kecil antara keduanya mengindikasikan bahwa model tidak mengalami *overfitting* secara signifikan. Seiring dengan bertambahnya jumlah data pelatihan, akurasi validasi meningkat secara bertahap hingga mencapai titik konvergen, yang mengindikasikan bahwa penambahan data berdampak positif terhadap kemampuan generalisasi model.

B. Implementasi

Setelah proses pelatihan dan evaluasi model selesai, langkah selanjutnya adalah mengimplementasikan model untuk mengklasifikasikan data jamur baru. Tujuan dari implementasi ini adalah untuk mengetahui bagaimana model berperforma terhadap data yang benar-benar baru dan belum pernah dilatih sebelumnya. Sebanyak lima sampel jamur baru dengan beragam karakteristik morfologis digunakan sebagai data uji implementasi, yang dimasukkan secara manual. Data ini telah melalui proses pra-pemrosesan yang konsisten dengan data pelatihan, termasuk *one-hot encoding* untuk fitur kategorikal dan normalisasi MinMaxScaler untuk fitur numerik (cap-diameter dan stem-height), memastikan kesesuaian format data untuk prediksi. Nilai masing-masing fitur pada lima sampel jamur baru ditampilkan pada Tabel 4.

Tabel 4. Sample Data Baru

| N o | Cap Diam eter | Cap Shape | Cup Surfa ce | Bru se | Gill Attach ment | Gill Spac ing | Gill color | Stem Height |
|--------|---------------------|--------------|--------------------|-----------|------------------------|---------------------|---------------|----------------|
| 1 | 5.0 | x | w | f | e | c | w | 10.0 |
| 2 | 7.0 | f | n | t | a | d | l | 15.0 |

| | | | | | | | | |
|---|-----|---|---|---|---|---|---|------|
| 3 | 3.5 | c | k | f | e | c | g | 8.0 |
| 4 | 5.5 | s | y | t | p | d | r | 9.0 |
| 5 | 2.0 | f | p | f | x | c | w | 15.0 |

Model kemudian menghasilkan prediksi untuk kelima sampel jamur. Hasilnya ditunjukkan pada Tabel 5 Model memprediksi bahwa Jamur 1, 3, dan 5 termasuk ke dalam kelas *poisonous*, sementara Jamur 2 dan 4 diklasifikasikan sebagai *edible*. Hasil ini memperlihatkan bahwa model dapat mengidentifikasi pola pada data baru dengan cukup baik, dan menunjukkan konsistensi dalam melakukan klasifikasi berdasarkan fitur morfologis.

Tabel 5. Hasil Implementasi Model

| No | Prediksi |
|----|-----------|
| 1 | Poisonous |
| 2 | Edible |
| 3 | Poisonous |
| 4 | Edible |
| 5 | Poisonous |

IV. KESIMPULAN

Penelitian ini berhasil mengimplementasikan algoritma Random Forest untuk mengklasifikasikan jenis jamur menjadi dua kelas, yaitu edible dan poisonous. Model yang dibangun menunjukkan performa sangat baik dengan akurasi sebesar 99% serta nilai precision, recall, dan f1-score yang seimbang pada kedua kelas. Analisis feature importance menunjukkan bahwa fitur morfologis seperti diameter tudung, tinggi batang, dan warna insang berperan penting dalam proses klasifikasi. Hasil implementasi model terhadap data jamur baru juga menunjukkan kemampuan prediksi yang konsisten dan akurat. Secara keseluruhan, model ini layak digunakan sebagai sistem pendukung keputusan dalam mendeteksi jamur beracun berdasarkan karakteristik morfologinya.

V. SARAN

Penelitian ini telah berhasil membangun model klasifikasi jamur beracun yang sangat akurat menggunakan algoritma Random Forest, memanfaatkan atribut morfologis dan lingkungan secara komprehensif, sehingga dapat menjadi fondasi kuat untuk sistem pendukung keputusan otomatis dalam upaya pencegahan keracunan jamur. Bagi peneliti lain, artikel ini membuka berbagai peluang pengembangan lebih lanjut yang dapat mencakup integrasi dataset jamur dari wilayah geografis berbeda atau perluasan cakupan spesies untuk meningkatkan generalisasi model, mempertimbangkan variasi morfologi dan habitat jamur; selain itu, pengembangan model menjadi aplikasi seluler atau *web-based* yang memungkinkan pengguna memotret atau memasukkan ciri-ciri jamur secara langsung untuk mendapatkan prediksi instan dapat dipertimbangkan; dan terakhir, eksplorasi metode ekstraksi fitur dari data non-tabular, seperti gambar melalui *deep learning* untuk fitur visual atau data genetik (*DNA barcoding*), dapat meningkatkan akurasi dan robustabilitas klasifikasi secara

signifikan. Dengan demikian, artikel ini menjadi landasan bagi penelitian lanjutan yang lebih dalam dan pengembangan aplikasi praktis dalam bidang deteksi jamur beracun.

REFERENSI

- [1] I. P. and P. D. M. Putra, “*Implementasi Metode Cnn Dalam Klasifikasi Gambar Jamur Pada Analisis Image Processing*,” Nov. 2020. Accessed: Apr. 08, 2025. [Online]. Available: <https://dspace.uii.ac.id/handle/123456789/23677>
- [2] B. Zhang, Y. Zhao, and Z. Li, “*Using Deep Convolutional Neural Networks To Classify Poisonous And Edible Mushrooms Found In China*.” [Online]. Available: <https://drive.google.com/drive/folders/13NFDI5UhLHPSL2WMcOrFs6QhhmrDjxQ?usp=sharing>
- [3] W. Ketwongsa, S. Boonlue, and U. Kokaew, “A New Deep Learning Model for The Classification of Poisonous and Edible Mushrooms Based on Improved AlexNet Convolutional Neural Network,” *Applied Sciences (Switzerland)*, vol. 12, no. 7, Apr. 2022, doi: 10.3390/app12073409.
- [4] 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: 10.54082/jiki.68.
- [5] U. Sri Rahmadhani and N. Lysbetti Marpaung, “*Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN*,” vol. 8, no. 2, 2023.
- [6] D. B. Saputra, V. Atina, F. E. Nastiti, and F. I. Komputer, “*Penerapan Model Crisp-Dm Pada Prediksi Nasabah Kredit Menggunakan Algoritma Random Forest*,” 2024. [Online]. Available: <http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/indexDwiBagusSaputra> <http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index>
- [7] S. Nuanmeesri and W. Sriurai, “*Development of the Edible and Poisonous Mushrooms Classification Model by using the Feature Selection and the Decision Tree Techniques*,” *Int J Eng Adv Technol*, vol. 9, no. 2, pp. 3061–3066, Dec. 2019, doi: 10.35940/ijeat.B4115.129219.
- [8] F. Azhiman, R. N. Dasmen, A. Putra, and W. Agustian, “*Collecting Data Desa di Kecamatan Rambutan Kabupaten Banyuasin untuk Pengimplementasian Sistem Digital Desa*,” *Jurnal Abdi Masyarakat Indonesia*, vol. 2, no. 6, pp. 1733–1742, Nov. 2022, doi: 10.54082/jamsi.532.
- [9] J. B. Angela, Islamiyah, and Ahmad Irsyad, “*Implementasi Visualisasi Data Berbasis Web Pada Exploratory Data Analysis Profil Kesehatan Kota Samarinda*,” *Kreatif Teknologi dan Sistem Informasi*

- (*KRETISI*), vol. 1, no. 1, pp. 9–16, Jul. 2023, doi: 10.30872/kretisi.v1i1.447.
- [10] A. Wijayanto, U. Sayyid, and A. R. Tulungagung, “Revitalisasi Penggunaan Media Serta Metode Belajar Dalam Pembelajaran Matematika Dan Teknik.” [Online]. Available: <https://www.researchgate.net/publication/378314350>
- [11] H. N. Putri, D. Retno, and S. Saputro, “Clustering Data Campuran Numerik dan Kategorik Menggunakan Algoritme Ensemble Quick RObust Clustering using linKs (QROCK),” 2022, [Online]. Available: <https://journal.unnes.ac.id/sju/index.php/prisma/>
- [12] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, “A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data,” Mar. 29, 2021, *Frontiers Media S.A.* doi: 10.3389/fenrg.2021.652801.
- [13] J. Burdack, F. Horst, S. Giesselbach, I. Hassan, S. Daffner, and W. I. Schöllhorn, “Systematic Comparison of the Influence of Different Data Preprocessing Methods on the Performance of Gait Classifications Using Machine Learning,” *Front Bioeng Biotechnol*, vol. 8, Apr. 2020, doi: 10.3389/fbioe.2020.00260.
- [14] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, “A survey on missing data in machine learning,” *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00516-9.
- [15] J. A. Samuels and J. I. Samuels, “One-Hot Encoding and Two-Hot Encoding: An Introduction”, doi: 10.13140/RG.2.2.21459.76327.
- [16] D. Singh and B. Singh, “Feature wise normalization: An effective way of normalizing data,” *Pattern Recognit*, vol. 122, p. 108307, 2022, doi: <https://doi.org/10.1016/j.patcog.2021.108307>.
- [17] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, “An improved random forest based on the classification accuracy and correlation measurement of decision trees,” *Expert Syst Appl*, vol. 237, p. 121549, 2024, doi: <https://doi.org/10.1016/j.eswa.2023.121549>.