

ANALISIS PERBANDINGAN HASIL KERJA ALGORITMA C4.5 DAN NAIVE BAYES DALAM MENGENALI SPESIES JAMUR YANG BERACUN

Muhammad Ar Rafi Saifuddin^{1*}, Bagas Muizul Rohmadan², Muhammad Alfa Reza³, Naufal Shafly Fawwaz⁴, Khairy Mumtaz Alhafidz⁵

^{1,2,3,4,5}Prodi Sistem Informasi, Universitas Bina Sarana Informatika, JL. Unkris, No. Raya No.2, JL. Jatiwringin Raya, Jaticepaka, Pondokgede, Bekasi, Jawa Barat, 17411

Keywords:

C4.5, Naive Bayes,
Klasifikasi Jamur, Orange
Data Mining

Correspondent Email:

naufalshaflyfawwaz@gmail.com

Abstrak. Jamur adalah makanan yang umum dikonsumsi, tetapi tidak semua jenis jamur bisa dimakan karena ada yang beracun. Karena itu, penting untuk bisa membedakan jamur yang aman dari yang beracun secara akurat. Penelitian ini bertujuan mengklasifikasikan jamur dengan membandingkan dua metode data mining yaitu C4.5 dan Naive Bayes, berdasarkan 22 ciri fisik jamur dari dataset Mushroom. Analisis dilakukan menggunakan Orange Data Mining dengan metode 10-fold cross validation agar hasilnya lebih stabil. Hasil menunjukkan bahwa metode C4.5 memiliki performa terbaik dengan akurasi 98,5%, precision 98,6%, recall 98,5%, F1-score 98,5%, dan AUC 0,983. Sementara itu, Naive Bayes mencapai akurasi 96,2%, precision 96,4%, recall 96,2%, F1-score 96,2%, dan AUC 0,998. Namun, performa Naive Bayes tidak stabil karena metode ini mengasumsikan semua fitur bebas satu sama lain, padahal tidak sesuai dengan karakteristik dataset. Dari Confusion Matrix dan ROC Curve, C4.5 menunjukkan kemampuan membedakan kelas yang lebih baik serta tingkat kesalahan prediksi lebih rendah dibandingkan Naive Bayes. Kesimpulannya, metode C4.5 lebih efektif dan akurat dalam mengklasifikasikan jamur, serta membantu dalam mengidentifikasi jamur yang bisa dimakan dan yang beracun.



Copyright © [JITET](http://www.jitet.org) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. Mushrooms are commonly consumed foods, but not all types are edible because some are poisonous, making it important to accurately distinguish safe mushrooms from toxic ones. This study aims to classify mushrooms by comparing two data mining methods, C4.5 and Naive Bayes, based on 22 physical characteristics from the Mushroom dataset, using Orange Data Mining with 10-fold cross-validation to ensure more stable results. The findings show that C4.5 achieved the best performance with an accuracy of 98.5%, precision of 98.6%, recall of 98.5%, F1-score of 98.5%, and an AUC of 0.983, while Naive Bayes reached an accuracy of 96.2%, precision of 96.4%, recall of 96.2%, F1-score of 96.2%, and an AUC of 0.998 but exhibited less stability due to its assumption of feature independence, which does not match the dataset's characteristics. Based on the Confusion Matrix and ROC Curve, C4.5 showed better class discrimination and a lower prediction error rate compared to Naive Bayes. In conclusion, C4.5 is more effective and accurate for mushroom classification and supports the identification of both edible and poisonous mushrooms.

1. PENDAHULUAN

Indonesia memiliki beragam jenis jamur karena memiliki iklim tropis, tetapi hal ini juga menyebabkan masalah besar dalam hal keamanan pangan. Banyak orang kesulitan membedakan jamur yang bisa dimakan dengan jamur yang beracun karena bentuknya hampir sama. Laporan kasus keracunan jamur liar di Indonesia dari tahun 2010 hingga 2020 menunjukkan terdapat 76 kasus dengan total 550 orang yang terkena, di antaranya 9 orang meninggal [1]. Kasus ini kerap terjadi karena kesalahan dalam mengenali jamur secara visual, seperti jamur *Clitocybe* yang sering dikira sebagai jamur *Termitomyces* atau *Agaricus* karena bentuknya yang mirip [2]. Karena itu, dibutuhkan sistem pengklasifikasian otomatis yang tepat agar bisa mengurangi risiko kecelakaan akibat kesalahan identifikasi manusia.

Menggunakan teknik *data mining* ternyata sangat efektif dalam menyelesaikan masalah klasifikasi jamur. Algoritma *Decision Tree C4.5* terbukti cukup kuat dalam mengolah data jenis jamur. Penelitian terbaru oleh [3] menunjukkan bahwa algoritma ini berhasil mencapai tingkat akurasi 99,90% dalam mengelompokkan jamur berdasarkan bentuk fisiknya. Hasil ini didukung juga oleh [4] yang membandingkan beberapa algoritma pada dataset jamur, di mana *Decision Tree C4.5* mencapai akurasi 97,77%, yang jauh lebih tinggi dibandingkan algoritma *Gaussian Naive Bayes* yang hanya mampu mencapai akurasi 63,55% pada dataset yang sama.

Meskipun *C4.5* memberikan hasil yang lebih baik, algoritma *Naive Bayes* tetap dianggap sebagai standar perbandingan yang penting karena kemampuannya dalam memproses data dengan cepat. Berbeda dengan penelitian Prabudifa, penelitian [5] menunjukkan bahwa *Naive Bayes* dapat mencapai tingkat akurasi hingga 95% ketika digunakan bersama dengan metode seleksi fitur yang didasarkan pada aturan asosiasi. Di sisi lain, [6] mencatat bahwa akurasi *Naive Bayes* mencapai 92%, yang meskipun cukup baik, masih kurang dibandingkan dengan *K-Nearest Neighbor (KNN)* yang mencapai 98% dalam mengklasifikasikan jamur beracun. Ketidakstabilan hasil *Naive Bayes* ini menunjukkan adanya peluang penelitian lebih lanjut untuk menguji secara langsung antara

Naive Bayes dan *C4.5* menggunakan dataset morfologi jamur.

Perdebatan tentang hasil kerja *C4.5* dan *Naive Bayes* juga berlangsung di berbagai masalah lain, yang menunjukkan bahwa kelebihan suatu algoritma sangat bergantung pada jenis data yang digunakan. Dalam studi [7] yang memprediksi kelulusan mahasiswa, *C4.5* berperforma lebih baik dengan akurasi mencapai 90%, dibandingkan *Naive Bayes* yang hanya 85%. Hal serupa juga ditemukan oleh [8] dalam kasus memprediksi stok barang, metode *C4.5* mencapai tingkat keakuratan sebesar 96,80%, sedangkan metode *Naive Bayes* hanya mencapai 91,20%. Namun, hasil yang berlawanan ditemukan oleh [9] mana *Naive Bayes* justru lebih unggul daripada *Decision Tree* dengan akurasi 89% dibandingkan 74% pada kasus klasifikasi kelulusan. Perbedaan hasil dari penelitian sebelumnya ini menunjukkan perlunya studi perbandingan yang spesifik pada dataset jamur yang memiliki risiko keselamatan tinggi.

Penelitian ini membandingkan secara langsung dua algoritma, yaitu *C4.5* dan *Naive Bayes*, dengan fokus pada mengurangi kesalahan klasifikasi kritis, khususnya *false negative*, dalam mengidentifikasi jamur beracun. Kesalahan ini memiliki dampak langsung terhadap keamanan pangan. Berbeda dengan penelitian sebelumnya yang hanya fokus pada akurasi, penelitian ini menggunakan metrik *Precision* dan *Recall* sebagai acuan utama untuk menilai risiko kesalahan dalam memprediksi jamur beracun sebagai jamur aman. Penelitian ini menggunakan dataset morfologi jamur dari *UCI Machine Learning Repository* dan skema evaluasi yang konsisten. Hasilnya memberikan kontribusi baru berupa evaluasi yang menyeluruh dan aplikatif untuk pengembangan sistem klasifikasi jamur beracun yang lebih dapat diandalkan dalam konteks keamanan pangan.

2. TINJAUAN PUSTAKA

2.1 Konsep Klasifikasi *Data Mining*

Klasifikasi adalah salah satu teknik dalam *data mining* yang memiliki peran penting, terutama dalam analisis data tekstual seperti sentimen. Proses klasifikasi melibatkan serangkaian tahapan, mulai dari pengumpulan data (*crawling*), *preprocessing* data (termasuk

cleaning data, tokenization, stop removal, dan case folding), pembagian data, hingga proses klasifikasi menggunakan metode tertentu [10]

2.2 Algoritma C4.5 (Decision Tree)

Algoritma C4.5 merupakan bagian dari model *Decision Tree* yang berperan sebagai pembangun model prediksi. Model pohon keputusan yang dihasilkan ini selanjutnya dievaluasi menggunakan berbagai metrik yang komprehensif, mencakup akurasi, presisi, *recall*, *F1-score*, dan *Area Under Curve (AUC)* [11]

2.3 Algoritma Naive Bayes

Algoritma Naive Bayes adalah metode klasifikasi yang memanfaatkan prinsip probabilitas. Algoritma ini bekerja dengan menggabungkan *prior probability* dan *probabilitas bersyarat* dalam sebuah rumus untuk menghitung probabilitas tiap klasifikasi yang mungkin (Kusnaeni et al., 2024). Keunggulan utama dari algoritma ini adalah efisiensinya karena hanya memerlukan jumlah data pelatihan yang relatif sedikit untuk menentukan parameter dalam proses klasifikasi [12]

2.4 Metrik Evaluasi Kinerja

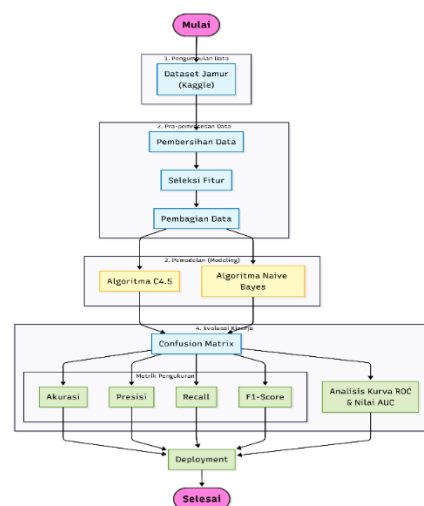
Untuk menilai seberapa baik sebuah algoritma klasifikasi bekerja, digunakan metrik kinerja standar. Performa algoritma yang baik dapat dilihat dari nilai *confusion matrix*, yang mencakup akurasi, presisi, *recall*, nilai *Area Under Curve (AUC)*, maupun *Receiver Operating Characteristic (ROC)* (Singgalen, 2022). Selain itu, dalam kasus data dunia nyata yang sering mengalami ketidakseimbangan data (*data imbalance*), masalah tersebut dapat diatasi dengan menerapkan metode *oversampling* seperti *Synthetic Minority Oversampling Technique (SMOTE)* [13]

3. METODE PENELITIAN

Penelitian ini membandingkan kedua algoritma klasifikasi, yaitu *Decision Tree* dan *Naive Bayes*, untuk mengetahui mana yang lebih baik dalam melakukan klasifikasi. Seluruh proses pengelolaan data, mulai dari tahap persiapan data hingga pengecekan hasil model, dilakukan dengan bantuan perangkat lunak Orange Data Mining. Alasan pemilihan perangkat lunak ini karena kemampuannya

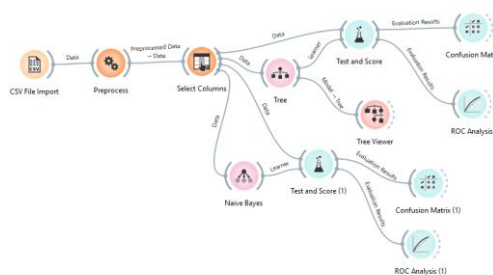
sebagai kerangka kerja yang lengkap dan berbasis komponen, di mana pengguna dapat dengan mudah memvisualisasikan [14].

alur data melalui widget di drawing board serta melakukan pemrograman secara visual [15] Sementara itu, tahapan-tahapan penelitian yang telah disusun secara terarah ditampilkan dalam Gambar 1.



Sumber: Penulis (202)

Gambar 1. Tahapan Penelitian



Sumber: Penulis (202)

Gambar 2. Desain widget model klasifikasi mushroom

3.1 Pengumpulan Data (Data Collection)

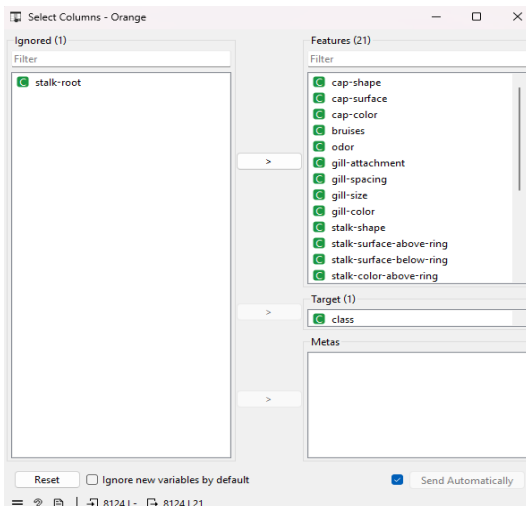
Dataset yang digunakan dalam penelitian ini adalah *Mushroom Dataset*, yang diambil dari repositori Kaggle (*UCI Machine Learning*). Dataset ini berisi 8.124 data dengan 22 atribut yang menjelaskan karakteristik fisik jamur, seperti bentuk tutup (*cap-shape*), warna tutup (*cap-color*), aroma (*odor*), hingga habitat jamur (batubara et al). Dataset ini memiliki dua kategori hasil (label), yaitu *edible* (dapat

dimakan) dan *poisonous* (beracun), yang akan dijadikan dasar dalam proses klasifikasi.

3.2 Pra-pemrosesan Data (*Preprocessing*)

Sebelum memulai pemodelan, data mentah harus melewati proses pra-pemrosesan untuk memastikan data memiliki kualitas terbaik. Tahapan ini meliputi:

- a. Pemberisan Data (*Data Cleaning*): Tahap ini bertujuan untuk menangani nilai yang hilang. Dalam dataset jamur, atribut *stalk-root* (akar tangkai) sering kali memiliki nilai yang kosong secara signifikan, sehingga perlu diperbaiki dengan cara seperti mengisi nilai yang hilang atau menghapus atribut tersebut agar tidak memengaruhi hasil secara tidak adil [16].
- b. Seleksi Fitur (*Feature Selection*): Tahap ini bertujuan memilih atribut yang paling penting dan berpengaruh terhadap klasifikasi, sekaligus menghilangkan atribut yang tidak memberikan kontribusi besar demi meningkatkan kecepatan dan efisiensi model komputasi.



Sumber: Penulis (202)

Gambar 3. Seleksi fitur dan penentuan atribut akibat target menggunakan *widget select columns*

c. Pembagian Data (*Data Splitting*)

Tahap terakhir dalam pra-pemrosesan data adalah membagi data. Penelitian ini menggunakan metode *Cross Validation* sebagai cara utama untuk mengecek kualitas model. Metode ini dipilih karena dianggap

lebih tepat dibandingkan cara pembagian data biasa, karena bisa mengurangi kesalahan yang muncul akibat pemisahan data tidak seimbang. Dasar prinsip dari metode ini adalah membagi dataset menjadi beberapa bagian atau yang disebut *fold*s [17].

3.3 Pemodelan (*Modeling*)

Pada tahap ini, dataset yang telah diperbaiki dibagi menjadi data latih dan data uji. Dua jenis algoritma klasifikasi digunakan untuk membuat model prediksi :

- a. Algoritma C4.5 (*Decision Tree*): Algoritma ini adalah metode pembelajaran terawasi yang cocok untuk memproses data berupa kategori. Algoritma ini membuat pohon keputusan secara bertahap dengan membagi data di setiap titik keputusan berdasarkan kondisi tertentu, sehingga data terbagi menjadi bagian-bagian yang lebih seragam. Dalam C4.5, atribut yang memberikan nilai *Information Gain* terbesar dipilih sebagai dasar pembagian yang terbaik untuk mengurangi ketidakpastian atau entropi dalam data [18].

1) Rumus *Entropy* (Ketidakmurnian Data)

$$Entropy(S) \quad (1)$$

$$= \sum_{i=1}^n -p_i \log_2(p_i)$$

Rumus *Entropy* digunakan untuk mengukur tingkat ketidakpastian atau kekacauan dalam suatu kelompok data berdasarkan teori informasi. Cara menghitungnya adalah dengan menjumlahkan hasil perkalian dari probabilitas setiap kelas dan logaritma basis dua dari probabilitas tersebut, kemudian diberi tanda negatif. Jika distribusi kelas dalam data merata, maka nilai *Entropy* akan besar, sedangkan jika data cenderung memiliki satu jenis kelas saja, nilai *Entropy* akan kecil. Dengan demikian, *Entropy* menjadi cara untuk mengetahui tingkat homogenitas dari suatu data dan digunakan dalam algoritma pohon keputusan untuk memilih atribut yang paling berguna dalam membagi data.

2) Rumus *Information Gain*

$$Gain(S, A) \quad (2)$$

$$= Entropy(S)$$

$$- \sum_{\{i=1\}}^n \left(\frac{|S_i|}{|S|} \right)$$

$$\cdot Entropy(S_i)$$

Rumus *Information Gain* digunakan untuk mengetahui seberapa besar informasi yang diperoleh dari pemisahan data berdasarkan suatu atribut. Nilai ini dihitung dengan cara mengurangi *Entropy* awal dari sekumpulan data dengan *Entropy* dari setiap bagian data yang terbentuk. Setiap bagian diberi bobot sesuai dengan jumlah data dalam bagian tersebut. Atribut yang memiliki *Information Gain* lebih tinggi dianggap lebih baik dalam mengurangi ketidakpastian dan membuat pengelompokan data lebih rapi. Karena itu, *Information Gain* menjadi dasar utama dalam algoritma *Decision Tree* seperti *ID3* untuk memilih atribut yang paling bermanfaat sebagai pemisah

b. Algoritma *Naive Bayes*: Metode ini adalah cara untuk mengelompokkan data ke dalam satu atau beberapa kategori yang sudah ditentukan sebelumnya. *Naive Bayes*

menggunakan cara menghitung probabilitas dan statistik yang pertama kali diperkenalkan oleh seorang ilmuwan asal Inggris bernama Thomas Bayes. Metode ini memperkirakan kemungkinan situasi di masa depan berdasarkan pengalaman di masa lalu. Kelebihan utama dari metode ini adalah sifatnya yang sederhana, namun tetap bisa memberikan akurasi yang baik, prosesnya cepat, hemat waktu, dan hanya membutuhkan sedikit data latihan untuk mendapatkan parameter-parameter yang diperlukan. Selain itu, algoritma ini mampu menghadapi data yang tidak relevan dan tidak terlalu terganggu oleh data yang tidak penting [19].

1) Rumus *Teorema Bayes*

$$P(H|X) \quad (3)$$

$$= \frac{P(X|H) \cdot P(H)}{P(X)}$$

Rumus *Teorema Bayes* digunakan untuk menghitung peluang suatu hipotesis setelah memperhitungkan bukti yang ditemukan. Dalam pendekatan ini, peluang awal dari hipotesis tersebut akan diperbarui berdasarkan seberapa besar kemungkinan bukti tersebut muncul jika hipotesis benar, kemudian dibandingkan dengan peluang total munculnya bukti tersebut. Dengan proses pembaruan ini, *Teorema Bayes* membantu menganalisis hubungan antara hipotesis dan bukti secara lebih akurat, sehingga memberikan dasar yang kuat dalam membuat keputusan berdasarkan probabilitas. Konsep ini sering digunakan dalam berbagai metode klasifikasi dan inferensi, termasuk dalam algoritma pembelajaran mesin, karena mampu menggabungkan informasi awal dengan data baru untuk menghasilkan estimasi probabilitas yang lebih tepat.

2) Prediksi (Untuk banyak atribut)

$$P(X|H) \quad (4)$$

$$= \prod_{j=1}^n P(X_j|H)$$

Hasil prediksi kelas Y ditentukan oleh probabilitas tertinggi (*argmax*):

$$Y = \underset{H}{\operatorname{argmax}} P(H|X) \quad (5)$$

Dalam proses memprediksi menggunakan metode *Naïve Bayes*, probabilitas kemunculan sebuah bukti atau atribut dihitung dengan cara mengalikan probabilitas setiap atribut terhadap kelas yang sedang diperiksa. Metode ini mengasumsikan bahwa setiap atribut bekerja secara mandiri, sehingga probabilitas gabungan bisa dicari dengan mengalikan probabilitas masing-masing atribut terhadap kelas tersebut. Setelah semua probabilitas untuk setiap kelas dihitung, langkah selanjutnya adalah memilih kelas yang memiliki probabilitas tertinggi sebagai hasil prediksi. Cara kerjanya ini membuat metode *Naïve Bayes* menjadi metode klasifikasi yang cepat dan efektif, terutama ketika terdapat banyak atribut, karena meskipun perhitungannya sederhana, metode ini mampu memberikan prediksi yang akurat berdasarkan distribusi probabilitas dari data.

3.4 Evaluasi (Penilaian)

Evaluasi model dilakukan dengan metode *Knowledge Discovery in Databases* (KDD), di mana langkah terakhir adalah memahami dan mengevaluasi hasil dari proses *data mining*. Untuk mengetahui seberapa baik model berfungsi, penelitian ini menggunakan *Confusion Matrix*, yaitu tabel yang membandingkan hasil prediksi model dengan nilai sebenarnya dari data. Komponen penting dalam *Confusion Matrix* adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). [20].

- a. Akurasi mengukur seberapa besar prediksi yang benar, baik yang menyatakan hal positif maupun negatif, dibandingkan dengan seluruh data yang ada. Rumus untuk menghitung akurasi adalah akurasi =

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (6)$$

- b. Presisi menunjukkan seberapa tepat prediksi positif yang diberikan sesuai dengan data sebenarnya yang positif. Rumusnya adalah

$$\text{Presisi} = \frac{Presisi}{TP} \times 100\% \quad (7)$$

- c. *Recall*, atau sensitivitas, mengukur seberapa baik sistem bisa menemukan informasi yang benar, yaitu kemampuannya mengenali data yang sebenarnya positif. Rumusnya adalah *Recall* =

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \quad (8)$$

- d. *F1-Score* adalah rata-rata harmonis dari presisi dan recall, yang memberikan gambaran tunggal mengenai kinerja model secara seimbang. Rumusnya adalah *F1-Score* =

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (9)$$

Selain menggunakan *Confusion Matrix*, hasil model juga dicek dengan kurva *ROC* dan nilai *AUC*. *AUC* digunakan untuk mengetahui seberapa baik model dalam membedakan antara kelas positif dan negatif, dengan nilai yang berada antara 0 hingga 1. Nilai *AUC* semakin tinggi, semakin baik kemampuan model dalam memprediksi kelas yang tepat.

3.5 Deployment

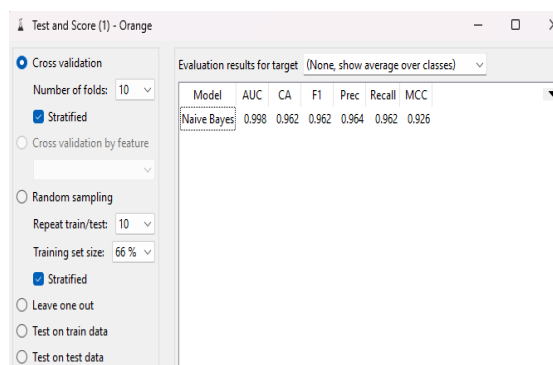
Tahap terakhir dalam penelitian ini adalah deployment atau penyebaran. Pada tahap ini, model klasifikasi yang telah diuji dan terbukti memiliki performa terbaik (berdasarkan hasil evaluasi akurasi, presisi, dan recall) dipilih sebagai model akhir. Hasil dari penelitian ini kemudian diaplikasikan untuk memprediksi data baru di masa depan yang belum memiliki informasi kelas.

4 HASIL DAN PEMBAHASAN

4.1 Hasil Komparasi Performa Algoritma Menggunakan 10-Fold Cross Validation

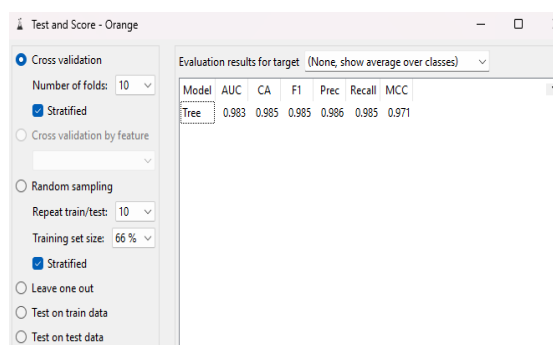
Berdasarkan hasil uji coba, *Naive Bayes* menghasilkan *AUC* sebesar 0,998, akurasi 0,962, *precision* 0,964, *recall* 0,962, *F1-score* 0,962, serta *MCC* 0,926. Meskipun *AUC* yang dicapai sangat tinggi, hasil lain menunjukkan bahwa model ini masih membuat kesalahan dalam prediksi, terutama pada kelas dengan fitur yang berkorelasi tinggi. Hal ini terjadi karena *Naive Bayes* menganggap semua fitur saling tidak tergantung, padahal data jamur memiliki fitur yang saling berkaitan. Oleh karena itu, model ini kurang efektif dalam memprediksi kelas yang benar. Di sisi lain, algoritma *C4.5* menunjukkan performa yang lebih baik dalam semua metrik. Model ini mencapai *AUC* sebesar 0,983, akurasi 0,985, *precision* 0,986, *recall* 0,985, *F1-score* 0,985, dan *MCC* 0,971. Hasil ini menunjukkan bahwa *C4.5* mampu mengenali pola dalam data dengan lebih baik, terutama karena pohon keputusan tidak mengasumsikan bahwa fitur saling tidak tergantung. Model ini juga mampu menangani data yang memiliki hubungan antar fitur secara efektif.

Hasil yang konsisten di setiap *fold* menunjukkan bahwa prediksi algoritma ini lebih stabil dibandingkan *Naive Bayes*. Secara keseluruhan, hasil menunjukkan bahwa *C4.5* adalah algoritma terbaik untuk dataset jamur yang digunakan. Model ini mampu membedakan jenis jamur yang bisa dimakan dan beracun dengan akurasi tinggi dan kesalahan yang minim. Sementara itu, *Naive Bayes*, meskipun memiliki *AUC* yang tinggi, menunjukkan performa yang kurang baik pada metrik lain karena keterbatasan dalam asumsi independensi fitur. Temuan ini sesuai dengan penelitian sebelumnya yang menyatakan bahwa algoritma pohon keputusan lebih efektif untuk dataset dengan struktur fitur yang kompleks dan saling berkorelasi. Dengan demikian, dapat disimpulkan bahwa *C4.5* lebih direkomendasikan untuk proses klasifikasi pada dataset jamur ini.



Sumber: Penulis (202)

Gambar 4. Hasil dari *cross validation* berdasarkan *Naive Bayes*



Sumber: Penulis (202)

Gambar 5. Hasil dari *cross validation* berdasarkan *C4.5*

4.2 Rangkuman Hasil Pengujian Metrik Utama

Tabel 1 menunjukkan ringkasan hasil penilaian dua algoritma berdasarkan uji coba menggunakan metode *10-fold cross validation*.

Tabel 1. Tabel ringkasan hasil perbandingan *C4.5* dan *Naive Bayes*

Algoritma	AUC	Akurasi (CA)	Precision	Recall	F1-Score
<i>C4.5 (Decision Tree)</i>	0.983	0.985	0.986	0.985	0.985
<i>Naive Bayes</i>	0.998	0.962	0.964	0.962	0.962

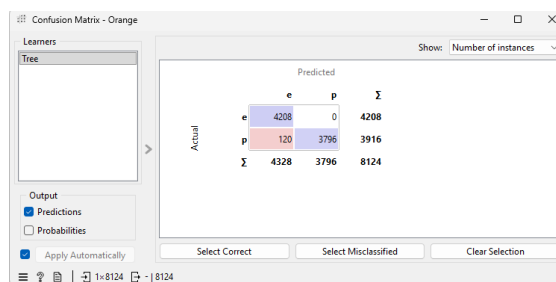
Sumber: Penulis (2025)

Dari tabel tersebut, algoritma *C4.5* menunjukkan performa terbaik pada hampir semua metrik penilaian, seperti *akurasi*, *presisi*, *recall*, *F1-score*, dan *MCC*. Sementara itu,

algoritma *Naive Bayes* memiliki nilai *AUC* yang sangat baik, tetapi performa pada metrik lain masih lebih rendah dibandingkan *C4.5*. Hasil ini menunjukkan bahwa algoritma *C4.5* lebih stabil dan akurat dalam mengklasifikasikan data dataset jamur dibandingkan *Naive Bayes*. Setelah memperoleh hasil validasi yang benar dengan menggunakan metode *10-fold cross validation*, dilakukan analisis lanjutan dengan *Confusion Matrix* untuk melihat bagaimana prediksi benar dan salah dibagikan oleh masing-masing algoritma. *Confusion Matrix* memberikan informasi yang lebih jelas mengenai kemampuan model dalam mengklasifikasikan setiap kelas, yaitu *edible (e)* dan *poisonous (p)*, serta memudahkan perhitungan berbagai ukuran penilaian seperti *akurasi*, *presisi*, *recall*, dan *F1-score*.

4.3 Confusion Matrix C4.5 dan Naive Bayes

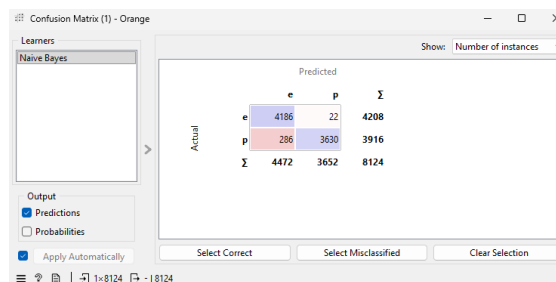
Hasil evaluasi menggunakan *Confusion Matrix*, performa algoritma *C4.5* dan *Naive Bayes* dianalisis secara lengkap dengan menghitung nilai *akurasi*, *presisi*, *recall*, dan *F1-score* berdasarkan rumus standar. Untuk algoritma *C4.5*, jumlah prediksi benar pada kelas *edible* mencapai 4.208 data dan prediksi benar pada kelas *poisonous* sebanyak 3.796 data. Sementara itu, kesalahan prediksi pada kelas *poisonous* tercatat sebanyak 120 data, sedangkan tidak ada kesalahan prediksi pada kelas *edible*. Berdasarkan distribusi ini, diperoleh nilai *True Positive (TP)* sebesar 3.796, *True Negative (TN)* sebesar 4.208, *False Positive (FP)* sebesar 0, dan *False Negative (FN)* sebesar 120. Dengan memasukkan nilai-nilai tersebut ke dalam rumus *akurasi* $(TP + TN)/(TP + TN + FP + FN)$, diperoleh akurasi sebesar 98,52%. Nilai presisi yang dihitung menggunakan rumus $TP/(TP + FP)$ menghasilkan nilai 100%, karena tidak ada prediksi positif yang salah. Selanjutnya, *recall* dihitung menggunakan rumus $TP/(TP + FN)$, yang menghasilkan nilai 96,94%. Sedangkan *F1-score*, yang merupakan rata-rata harmonis dari *presisi* dan *recall*, diperoleh sebesar 98,44%. Hasil ini menunjukkan bahwa *C4.5* mampu memberikan performa klasifikasi yang sangat stabil dan akurat dengan tingkat kesalahan yang rendah, terutama dalam mendeteksi kelas *poisonous*.



Sumber: Penulis (2025)

Gambar 6. Hasil dari *Confusion Matrix C4.5*

Dalam algoritma *Naive Bayes*, hasil dari *Confusion Matrix* menunjukkan bahwa model berhasil mengklasifikasikan 4.186 data *edible* dan 3.630 data *poisonous* secara benar. Meski demikian, ada beberapa kesalahan dalam prediksi, yaitu 22 data *edible* yang salah dikategorikan sebagai *poisonous* dan 286 data *poisonous* yang salah dinyatakan sebagai *edible*. Jadi, nilai *TP (True Positive)* dari model ini adalah 3.630, *TN (True Negative)* sebanyak 4.186, *FP (False Positive)* sejumlah 22, dan *FN (False Negative)* mencapai 286. Dengan menggunakan rumus yang sama untuk menghitung akurasi, diperoleh angka akurasi sebesar 96,21%, yang lebih rendah dibandingkan dengan algoritma *C4.5*. Nilai presisi model dihitung dengan rumus $TP/(TP + FP)$, yaitu 99,40%, menunjukkan bahwa sebagian besar prediksi yang diberi label positif benar. Namun, nilai *recall* yang dihitung dengan rumus $TP/(TP + FN)$ hanya mencapai 92,70%, artinya model ini lebih sering gagal mengenali data *poisonous* dibandingkan *C4.5*. Berdasarkan itu, nilai *F1-score* yang dihitung dengan formula harmonis antara presisi dan *recall* adalah 95,93%, yang lebih kecil dibandingkan dengan algoritma *C4.5*.

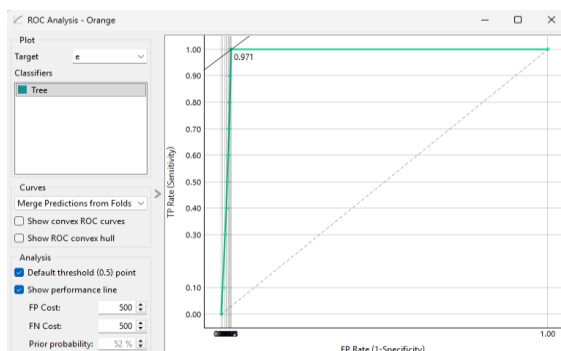


Sumber: Penulis (2025)

Gambar 7. Hasil dari *Confusion matrix Naive Bayes*

4.4 Analisis Kemampuan Pembeda C4.5 dan Naive Bayes Berdasarkan Grafik ROC

Analisis kurva ROC dilakukan untuk mengetahui seberapa baik model dalam membedakan antara kelas makanan yang aman dan beracun. Kurva ROC menunjukkan hubungan antara Tingkat Kebenaran Positif (TPR) dan Tingkat Pemalsuan Positif (FPR) pada berbagai nilai ambang. Model yang bagus akan memiliki kurva ROC yang mendekati sudut kiri atas grafik, artinya TPR tinggi dan FPR rendah. Pada algoritma C4.5, kurva ROC menunjukkan performa yang sangat baik dengan bentuk kurva yang hampir vertikal, mendekati titik ideal. Titik-titik pada kurva menggambarkan bahwa model memiliki tingkat kesalahan prediksi yang sangat rendah untuk berbagai nilai ambang. Nilai *Area Under Curve* (AUC) sebesar 0,971 menunjukkan bahwa C4.5 sangat mampu membedakan kedua kelas tersebut. Semakin mendekati angka 1, semakin baik kemampuan model, dan nilai AUC yang diperoleh menunjukkan bahwa C4.5 sangat stabil dan efektif.

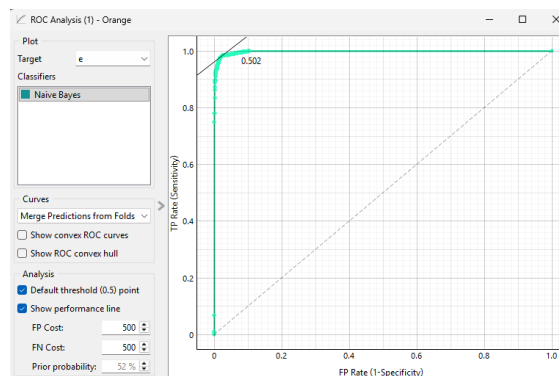


Sumber: Penulis (2025)

Gambar 8. Hasil dari ROC Curve C4.5

Sementara itu, algoritma *Naive Bayes* juga menunjukkan kurva ROC yang cukup memuaskan, meskipun bentuk kurvanya lebih melengkung perlahan dibandingkan dengan C4.5. Ada beberapa titik di kurva yang menunjukkan kenaikan tingkat *False Positive Rate* pada berbagai nilai *threshold*, yang menunjukkan bahwa model lebih rentan terhadap perubahan data. Nilai AUC untuk *Naive Bayes* adalah 0,502, yang menunjukkan kemampuan model dalam membedakan kelas jauh lebih rendah dibandingkan C4.5. Nilai AUC yang mendekati 0,5 berarti performa model hampir sama dengan tebakan secara

acak. Hal ini terkait dengan asumsi dasar *Naive Bayes* yang membutuhkan setiap fitur bersifat *independen*, padahal pada dataset jamur, fitur-fitur tersebut saling berkaitan.



Sumber: Penulis (2025)

Gambar 9. Hasil dari ROC Curve Naive Bayes

4.5 Evaluasi Menyeluruh Keandalan Kedua Algoritma

Secara keseluruhan, hasil penghitungan semua metrik menunjukkan bahwa algoritma C4.5 memiliki performa yang lebih baik dan lebih seimbang dibandingkan dengan *Naive Bayes*. C4.5 unggul dalam semua metrik penilaian, terutama pada akurasi dan *recall*, yang menunjukkan kemampuan model dalam mengenali kedua kelas, yaitu jamur yang bisa dimakan dan beracun. Perbedaan hasil ini menunjukkan bahwa struktur dataset jamur yang memiliki hubungan antar atribut lebih cocok diproses dengan algoritma berbasis pohon keputusan dibandingkan dengan model probabilistik yang mengasumsikan fitur independen seperti *Naive Bayes*.

5 KESIMPULAN

Berdasarkan hasil evaluasi yang mencakup *10-fold cross validation*, *Confusion Matrix*, dan analisis kurva ROC, dapat disimpulkan bahwa algoritma C4.5 memiliki performa yang lebih baik dibandingkan *Naive Bayes* dalam mengklasifikasikan data jamur pada penelitian ini. C4.5 menunjukkan nilai akurasi, *precision*, *recall*, dan *F1-score* yang lebih tinggi, serta jumlah kesalahan klasifikasi yang lebih rendah, terutama pada kelas jamur beracun yang penting untuk dideteksi dengan tepat. Kurva ROC algoritma C4.5 menunjukkan kemampuan membedakan kelas yang sangat baik dengan

AUC mendekati 1, yang menunjukkan stabilitas model dalam memisahkan kedua kelas.

Di sisi lain, meskipun *Naive Bayes* memiliki nilai *AUC* yang baik, performanya menurun pada metrik lain karena asumsi independensi fitur tidak sepenuhnya terpenuhi dalam dataset jamur. Oleh karena itu, *C4.5* terbukti lebih efektif dan akurat dibandingkan *Naive Bayes* untuk digunakan dalam klasifikasi data jamur dalam penelitian ini.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] I. P. Putra, "KASUS-KASUS KERACUNAN JAMUR LIAR DI INDONESIA," *JURNAL EKOLOGI KESEHATAN*, vol. 20, no. 3, pp. 215–230, Mar. 2022, doi: 10.22435/jek.v20i3.4943.
- [2] I. P. Putra and R. Hermawan, "Identifikasi Jamur Beracun *Clitocybe* sp. di Gresik, Indonesia (Studi Kasus)," *Media Penelitian dan Pengembangan Kesehatan*, vol. 31, no. 2, pp. 119–124, Aug. 2021, doi: 10.22435/mpk.v31i2.4352.
- [3] R. Fathurrohman, M. Fajri S, S. M. Cahyono, D. F. Abdillah, and V. Ramadhan, "Data Mining Pada Klasifikasi Jamur Menggunakan Algoritma C.45 Berdasarkan Karakteristik Morfologi Mushroom," Jun. 2024. doi: <https://doi.org/10.31294/simpatik.v4i1.3529>.
- [4] M. Yusuf Prabdifa, O. Sukma Mayangsari, A. Desiani, D. Alwine Zayanti, R. Primartha, and D. Novita, "Ensemble Classifier dengan Teknik Weighted Voting untuk Klasifikasi Jamur Beracun," vol. 12, no. 2, pp. 39–50, Jun. 2025, doi: <https://doi.org/10.35957/jatiasi.v12i2.10168>.
- [5] M. R. Al Aziz, M. T. Furqon, and L. Muflikhah, "Klasifikasi Jamur Dapat Dimakan atau Beracun Menggunakan Naïve Bayes dan Seleksi Fitur berbasis Association Rule Mining," Aug. 2022. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [6] 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.
- [7] A. Rahmayanti, L. Rusdiana, and S. Suratno, "PERBANDINGAN METODE ALGORITMA C4.5 DAN NAÏVE BAYES UNTUK MEMPREDIKSI KELULUSAN MAHASISWA," *Walisongo Journal of Information Technology*, vol. 4, no. 1, pp. 11–22, Aug. 2022, doi: 10.21580/wjit.2022.4.1.9654.
- [8] Rian Pratama, B. Huda, E. Novalia, and H. Kabir, "Perbandingan Algoritma C4.5 dan Naïve Bayes dalam Menentukan Persediaan Stok," *METIK JURNAL*, vol.6, no. 2, pp. 115–122, Dec. 2022, doi: <https://doi.org/10.47002/metik.v6i2.379>.
- [9] S. Alim, Anwari, and Hozairi, "IMPLEMENTASI ORANGE DATA MINING UNTUK KLASIFIKASI KELULUSAN MAHASISWA DENGAN MODEL K-NEAREST NEIGHBOR, DECISION TREE SERTA NAIVE BAYES ORANGE DATA MINING IMPLEMENTATION FOR STUDENT GRADUATION CLASSIFICATION USING K- NEAREST NEIGHBOR, DECISION TREE AND NAIVE BAYES MODELS," 2021. doi: 10.21107/nero.v6i2.237.
- [10] D. Duei Putri, G. F. Nama, and W. E. Sulistiono, "Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 10, no. 1, Jan. 2022, doi: 10.23960/jitet.v10i1.2262.
- [11] M. A. Nasution, Z. A. Ulumuddin, and A. Fitri, "Analisis Faktor Risiko Diabetes Melitus Menggunakan Algoritma C4.5: Implementasi pada Aplikasi Orange," Oct. 2024. doi: <https://doi.org/10.56495/saintek.v1i3.1345>.
- [12] F. M. Anto, L. S. Abimanyu, and T. Herdi, "Penerapan Algoritma Naïve Bayes Dengan Feature Selection Pada Data Penjualan Konstruksi," *Jurnal Ilmiah FIFO*, vol. 15, no. 2, p. 102, Apr. 2024, doi: 10.22441/fifo.2023.v15i2.002.
- [13] M. Rafly Gusmansyah and H. Hendrawan, "Peningkatan Kinerja Analisis Sentimen pada Ulasan Aplikasi Identitas Kependudukan Digital (IKD) di Indonesia Menggunakan Algoritma Support Vector Machine (SVM) dan SMOTE," vol. 10, no. 1, pp. 2541–1179, 2025, doi: <https://doi.org/10.24252/instek.v10i1.55292>.
- [14] E. Mardiani *et al.*, "Membandingkan Algoritma Data Mining Dengan Tools Orange untuk Social Economy," *Digital Transformation Technology*, vol. 3, no. 2,

- pp.686–693, Nov. 2023, doi: 10.47709/digitech.v3i2.3256.
- [15] E. Pranadjaya, E. Sudira Pangestu, C. O. Sereati, S. Octaviani, and M. Darmawan, “Perbandingan Algoritma Machine Learning menggunakan Orange Data Mining untuk Klasifikasi Jenis Kendaraan pada Sistem Tilang Digital,” 2024. doi: <https://doi.org/10.25170/jurnalelektro.v17i1.5429>.
- [16] C. Mirna Wati, A. Charis Fauzan, and Harliana, “PERFORMANCE COMPARISON OF MUSHROOM TYPE CLASSIFICATION BASED ON MULTI-SCENARIO DATASET USING DECISION TREE C4.5 AND C5.0,” vol. 4, Jun. 2022, doi: 10.34288/jri.v4i3.173.
- [17] A. Putri, Jasmir, and B. Purnama, “Klasifikasi Kelayakan Penerima Beasiswa Menggunakan Naive Bayes dengan Optimasi Atribut Berbasis K-Means Clustering,” vol. 5, pp. 1450–1462, 2025, doi: 10.57152/malcom.v5i4.2312.
- [18] S. Tangirala, “Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm*,” 2020. doi: <http://dx.doi.org/10.14569/IJACSA.2020.0110277>.
- [19] A. Pebdika, R. Herdiana, and D. Solihudin, “KLASIFIKASI MENGGUNAKAN METODE NAIVE BAYES UNTUK MENENTUKAN CALON PENERIMA PIP,” 2023. doi: <https://doi.org/10.36040/jati.v7i1.6303>.
- [20] D. Wulan Yekti Rahayu, K. Umam, and M. Rini Handayani, “Performance of Machine Learning Algorithms on Imbalanced Sentiment Datasets Without Balancing Techniques,” 2025. doi: <https://doi.org/10.30871/jaic.v9i3.958>