

Klasifikasi Jamur Beracun Menggunakan Algoritma Naïve Bayes dan K-Nearest Neighbors

Gracia Mianda Caroline Batubara¹, Anita Desiani^{*2}, Ali Amran³

^{1,2,3}Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sriwijaya, Indonesia
Email: ¹graciamianda2002@gmail.com, ²anita_desiani@unsri.ac.id, ³ali_amran@mipa.unsri.ac.id

Abstrak

Jamur adalah salah satu organisme eukariot heterotrof dengan jenis yang sangat banyak, sekitar 1.500.000 di dunia. Namun, pengenalan akan jamur masih sangat kurang, dimana jumlah jamur yang sudah dikenali hanya sebanyak 74.000 jenis. Beragamnya jenis jamur ini membuat pengenalan akan klasifikasi jamur menjadi sangat penting agar manusia tidak mengonsumsi jamur beracun yang akan memberikan dampak negatif. Penelitian ini bertujuan untuk menemukan algoritma terbaik dalam pengklasifikasian jamur beracun dan tidak beracun. Klasifikasi jamur berdasarkan ciri-cirinya dapat dilakukan melalui penerapan algoritma Naïve Bayes dan k-Nearest Neighbors (kNN) pada dataset jamur. Hasilnya, algoritma Naïve Bayes memberikan rata-rata akurasi sebesar 92%, lebih kecil dibanding k-Nearest Neighbors yang memberikan rata-rata akurasi sebesar 98%. Rata-rata presisi algoritma Naïve Bayes dan k-Nearest Neighbors sama, yaitu 92,5%. Rata-rata recall algoritma Naïve Bayes sebesar 91,5% dan algoritma k-Nearest Neighbors sebesar 98%. Berdasarkan rata-rata akurasi, presisi, dan recall kedua algoritma tersebut, dapat disimpulkan bahwa algoritma k-Nearest Neighbors lebih baik dibanding algoritma Naïve Bayes dalam klasifikasi jamur beracun. Namun, rata-rata akurasi, presisi, dan recall dari algoritma Naïve Bayes masih tergolong sangat baik karena nilainya berada diatas 90%.

Kata kunci: Jamur, k-Nearest Neighbors, Naïve Bayes, Perbandingan

Abstract

Mushrooms are one of the most abundant heterotrophic eukaryotic organisms, around 1,500,000 in the world. However, the recognition of mushrooms is still very lacking, where the number of fungi that have been recognized is only 74,000 species. The diversity of these types of mushrooms makes the recognition of the classification of mushrooms very important so that humans do not consume poisonous mushrooms that will give a negative impact. This study aims to find the best algorithm for classifying poisonous and edible mushrooms. Classification of mushroom based on their characteristics can be done through the application of the Naïve Bayes algorithm and k-Nearest Neighbors (kNN) on the mushroom dataset. As a result, the Naïve Bayes algorithm provides an average accuracy of 92%, smaller than k-Nearest Neighbors which provides an average accuracy of 98%. The average precision of the Naïve Bayes algorithm and k-Nearest Neighbors is the same, which is 92.5%. The average recall of the Naïve Bayes algorithm is 91.5% and the k-Nearest Neighbors algorithm is 98%. Based on the average accuracy, precision, and recall of the two algorithms, it can be concluded that the k-Nearest Neighbors algorithm is better than the Naïve Bayes algorithm in the classification of poisonous mushrooms. However, the average accuracy, precision, and recall of the Naïve Bayes algorithm is still very good because the value is above 90%.

Keywords: Comparison, k-Nearest Neighbors, Mushrooms, Naïve Bayes

1. PENDAHULUAN

Jamur merupakan organisme eukariot heterotrof yang memerlukan senyawa organik untuk nutrisinya yang bersifat uniseluler maupun multiseluler [1]. Terdapat lebih dari 1.500.000 jenis keanekaragaman jamur di dunia dan yang sudah dikenali hanya sekitar 74.000 jenis [2]. Jamur-jamur tersebut terbagi menjadi 3 kelas, yaitu jamur yang pasti dapat dikonsumsi, jamur yang pasti beracun, dan jamur yang tidak diketahui beracun atau tidak. Kelas ketiga ini digabungkan ke dalam kelas beracun [3]. Beberapa jamur yang sudah dikenali bermanfaat dalam bidang pangan, pertanian, kesehatan, ekonomi. Dalam bidang pangan, jamur dapat dikonsumsi sebagai bahan pelengkap makanan karena memiliki rasa yang enak serta nilai kecukupan gizi yang baik [4]. Namun, terdapat juga jenis jamur yang

menyebabkan infeksi pada manusia [5]. Dampak buruk lain yang dapat terjadi apabila manusia mengonsumsi jamur beracun adalah halusinasi, iritasi gastrointestinal, nekrosis hati, dan kematian.

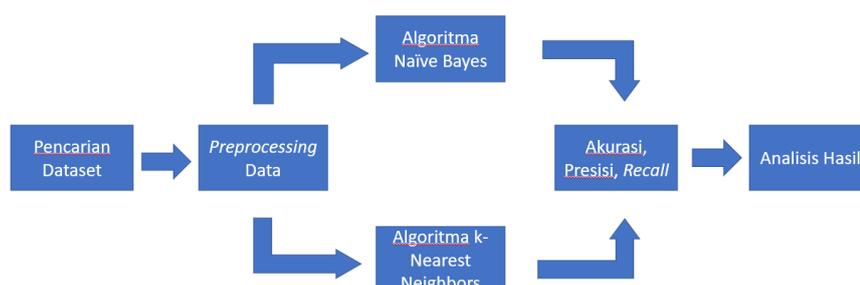
Hal ini menunjukkan betapa pentingnya pengenalan terhadap jamur agar manusia dapat dengan bijaksana memanfaatkan jamur tersebut sesuai dengan kebutuhannya. Salah satu cara pengenalan jamur adalah dengan melihat ciri-ciri jamur itu sendiri. Jamur beracun memiliki ciri, seperti memiliki cincin dan cawan dibagian batang, memiliki bitnik-bintik di bawah tudung jamur, dan berwarna sangat mencolok. Salah satu cara mendeteksi jenis jamur adalah dengan melakukan pengklasifikasian. Klasifikasi merupakan cara pengelompokan benda berdasarkan ciri-ciri yang dimiliki oleh objek klasifikasi [6]. Pengklasifikasian ini dilakukan dengan *data mining*.

Penelitian mengenai klasifikasi jamur pernah dilakukan oleh Wibowo, *et al* [7] yang melakukan pengklasifikasian jenis jamur famili Agaricus dan Lepiota dengan metode Naïve Bayes, C4.5, dan SVM (*Support Vector Machines*). Hasilnya, metode C45 dan SVM memiliki tingkat akurasi paling baik, yaitu sebesar 100%. Penelitian lain pernah dilakukan oleh Maurya dan Singh [8] yang mengklasifikasikan jamur menggunakan metode SVM dengan akurasi 76,6%. Banyak metode matematis yang digunakan dalam proses klasifikasi, diantaranya Naïve Bayes dan k-Nearest Neighbors (kNN). Algoritma Naïve Bayes merupakan salah satu algoritma pembelajaran induktif yang paling efektif dan efisien untuk *machine learning* dan *data mining* [9]. Kelebihan algoritma ini adalah Naïve Bayes menunjukkan akurasi dan kecepatan yang tinggi bila diterapkan pada database yang besar dan hanya membutuhkan data *training* dengan jumlah yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian [9]. Salah satu contoh penelitian menggunakan algoritma Naïve Bayes adalah klasifikasi masyarakat miskin yang dilakukan oleh Annur [10] dengan hasil akurasi sebesar 73%. Algoritma k-Nearest Neighbors merupakan salah satu algoritma yang telah banyak digunakan dalam pengklasifikasian [11]. Kelebihan algoritma k-Nearest Neighbors, yaitu sederhana dan mudah dipelajari, proses pelatihan yang sangat cepat, tetap dapat digunakan pada data yang memiliki bias dan tetap efektif untuk digunakan pada data berjumlah besar. Kekurangan dari algoritma k-Nearest Neighbors adalah keterbatasan memori, biasanya nilai k, komputasi kompleks, dan terpengaruh akan data-data yang tidak relevan [12]. Penelitian yang menggunakan metode k-Nearest Neighbors memberikan hasil yang sangat baik, contohnya klasifikasi penentuan pengajuan kredit yang dilakukan oleh Kurniawan, *et al* [13] dengan akurasi sebesar 93%.

Penelitian ini menerapkan algoritma Naïve Bayes dan k-Nearest Neighbors dalam mengklasifikasikan dataset jamur untuk menentukan apakah jamur yang diuji merupakan jamur beracun atau jamur tidak beracun (*poisonous* atau *edible*). Dataset yang digunakan diambil dari dataset *Mushroom Classification* yang terdiri atas 22 atribut. Atribut-atribut tersebut merupakan beberapa ciri jamur dan akan digunakan dalam proses klasifikasi. Dalam penelitian ini, akan dihitung akurasi, recall, dan presisi untuk mengukur kinerja algoritma Naïve Bayes dan k Nearest Neighbor. Hasil tersebut kemudian akan dibandingkan untuk menemukan algoritma terbaik dalam pengklasifikasian jamur beracun dan tidak beracun.

2. METODE PENELITIAN

Dalam penelitian ini, metode yang digunakan untuk mengklasifikasikan sebuah jamur termasuk jamur beracun atau tidak adalah metode Naïve Bayes dan k-Nearest Neighbors (kNN). Tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Dari Gambar 1 dapat diketahui bahwa tahapan pertama dalam penelitian ini adalah mencari dataset. Dataset tersebut kemudian akan diperiksa kelengkapannya terlebih dahulu melalui tahap *preprocessing*. Setelah semua data telah dapat diolah, tahap selanjutnya adalah menerapkan algoritma Naïve Bayes dan k-Nearest Neighbors pada dataset tersebut. Hasilnya, didapatkan akurasi, presisi, dan *recall* dari masing-masing algoritma. Hasil tersebut kemudian akan dibandingkan dan didapatkan algoritma terbaik dalam mengklasifikasikan jamur beracun.

2.1. Deskripsi Data

Data yang digunakan dalam penelitian ini adalah dataset klasifikasi jamur (*Mushroom Classification*) yang bersumber dari situs Kaggle (<https://www.kaggle.com/datasets/uciml/mushroom-classification>) berformat csv. Terdapat 23 atribut dimana 22 atribut merupakan atribut ciri dan satu lainnya merupakan atribut kelas. Kelas terdiri atas 2 label, yaitu e untuk *edible* atau tidak beracun dan p untuk *poisonous* atau beracun. Jumlah keseluruhan data adalah sebanyak 8.124 data, terdiri dari 3.916 data merupakan jamur beracun dan 4.208 data merupakan jamur tidak beracun. Atribut-atribut yang digunakan, yaitu *class*, *cap shape*, *cap surface*, *cap color*, *bruises*, *odor*, *gill attachment*, *gill spacing*, *gill size*, *gill color*, *stalk shape*, *stalk root*, *stalk surface above ring*, *stalk surface below ring*, *stalk color above ring*, *stalk color below ring*, *veil type*, *veil color*, *ring number*, *ring type*, *spore print color*, *population*, dan *habitat*. Atribut dan penjelasannya dapat dilihat pada Tabel 1.

Tabel 1. Penjelasan dari Atribut

Attribute	Feature Meaning
<i>Class</i> / Kelas	p = poisonous/beracun (1), e = edible/tidak beracun (0)
<i>Cap Shape</i> / Bentuk Tutup	x = kerucut (0), b = lonceng (1), s = cekung (2), f = datar (3), k = kenop (4), c = kerucut (5)
<i>Cap Surface</i> / Tutup Permukaan	s = halus (0), y = bersisik (1), f = berserat (2), g = alur (3)
<i>Cap Color</i> / Warna Topi	n = coklat (0), y = kuning (1), w = putih (2), g = abu-abu (3), e = merah (4), p = pink (5), b = buff (6), u = ungu (7), c = kayu manis (8), r = hijau (9)
<i>Bruises</i> / Memar	t = memar (0), f = tidak memar (1)
<i>Odor</i> / Bau	p = pungent (0), a = almond (1), l = anise (2), n = tidak berbau (3), f = foul (4), c = creosote (5), y = amis (6), s = pedas (7), m = apek (8)
<i>Gill Attachment</i> / Lampiran Insang	f = bebas (0), a = menurun (1)
<i>Gill Spacing</i> / Jarak Insang	c = dekat (0), w = ramai (1)
<i>Gill Size</i> / Ukuran Insang	n = sempit (0), b = lebar (1)
<i>Gill Color</i> / Warna Insang	k = hitam (0), n = brown/coklat (1), g = abu-abu (2), p = pink (3), w = putih (4), h = chocolate/cokleat (5), u = ungu (6), e = merah (7), b = buff(8), r = hijau (9), y = kuning (10), o = oranye (11)
<i>Stalk Shape</i> / Bentuk Tangkai	e = membesar (0), t = meruncing (1)
<i>Stalk Root</i> / Akar Tangkai	e = equal (0), c = club (1), b = bulbous (2), r = berakar (3), ? = missing/nilai hilang
<i>Stalk Surface Above Ring</i> / Tangkai Permukaan Atas Cincin	s = halus (0), f = berserat (1), k = sangat halus (2), y = bersisik (3)
<i>Stalk Surface Below Ring</i> / Tangkai Permukaan Bawah Cincin	s = halus (0), f = berserat (1), y = bersisik (2), k = sangat halus (3)
<i>Stalk Color Above Ring</i> / Warna Tangkai Atas Cincin	w = putih (0), g = abu-abu (1), p = pink (2), n = coklat (3), b = buff (4), e = merah (5), o = oranye (6), c = kayu manis (7), y = kuning (8)

<i>Stalk Color Below Ring</i> / Warna Tangkai Bawah Cincin	w = putih (0), p = pink (1), g = abu-abu (2), b = buff (3), n = coklat (4), e = merah (5), y = yellow (6), o = oranye (7), c = kayu manis (8)
<i>Veil Type</i> / Tipe Penutup	p = sebagian (0)
<i>Veil Color</i> / Warna Penutup	w = putih (0), n = coklat (1), o = oranye (2), y = kuning (3)
<i>Ring Number</i> / Banyaknya Cincin	o = satu (0), t = dua (1), n = tidak ada (2)
<i>Ring Type</i> / Tipe Cincin	p = liontin (0), e = sarang laba-laba (1), l = besar (2), f = <i>flaring</i> (3), n= tidak ada (4)
<i>Spore Print Color</i> / Warna Spora	k = hitam(0), n = <i>brown</i> /coklat (1), u = ungu (2), h = chocolate/cokelat (3), w = putih (4), r = hijau (5), o = oranye (6), y = kuning (7), b = buff (8)
<i>Population</i> / Populasi	s = tersebar (0), n = banyak (1), a = berlimpah (2), v = beberapa (3), y = soliter (4), c = berkelompok (5)
<i>Habitat</i> / Habitat	u = urban (0), g = rerumputan (1), m = padang rumput (2), d = kayu (3), p = jalan (4), w = limbah (5), l = daun (6)

2.2. Preprocessing Data

Tahap *preprocessing* sangat penting dilakukan karena kualitas dari input data sangat mempengaruhi kualitas *output* analisis yang dihasilkan [14]. Pada tahapan ini, kelengkapan data (ada atau tidaknya data hilang) akan diperiksa. Selain itu, akan dilakukan pemilihan terhadap atribut-atribut yang ada. Atribut yang tidak berpengaruh akan dihapus atau tidak akan digunakan. Dalam penelitian ini, terdapat 2 atribut yang tidak akan digunakan atau akan dihapus. Pertama, atribut *Stalk Root* karena di dalamnya terdapat banyak *missing value* atau data hilang. Kedua, atribut *Veil Type* karena dalam dataset, atribut ini hanya memiliki satu tipe, yaitu p, sehingga hasilnya tidak akan berpengaruh dalam proses klasifikasi. Maka, terdapat 21 atribut yang akan digunakan dalam klasifikasi, dimana 20 diantaranya adalah atribut ciri dan satu lainnya merupakan kelas.

Teknik yang digunakan dalam pengklasifikasian ini adalah *persentase split* dengan ratio 8:2. Artinya, data akan dibagi menjadi 2 bagian, yaitu data *training* dan data *testing*. Data *training* merupakan data yang digunakan dalam proses latih atau pembelajaran sedangkan data *testing* merupakan data yang akan diuji. Sebanyak 80% data akan menjadi data *training* dan 20% data akan menjadi data *testing*. Dalam algoritma kNN, nilai k yang digunakan adalah 51.

2.3. Algoritma Naïve Bayes

Algoritma Naïve Bayes merupakan salah satu pengklasifikasi statistik yang dapat memprediksi probabilitas keanggotaan kelas suatu data yang akan masuk ke dalam kelas tertentu, sesuai dengan perhitungan probabilitas [14]. Menurut Ratniasih [15] langkah-langkah melakukan klasifikasi menggunakan algoritma Naïve Bayes adalah sebagai berikut:

- Menghitung jumlah kategori dari setiap variabel
- Menghitung peluang pada setiap kategori
- Menentukan frekuensi atau jumlah kemunculan pada setiap kategori
- Menentukan kategori dengan nilai maksimal

Perhitungan Naïve Bayes dapat dilakukan dengan rumus pada persamaan (1) [15] :

$$P(H | X) = \frac{P(X | H) P(H)}{P(X)} \quad (1)$$

dimana:

- X : data dengan kelas yang belum diketahui
H : hipotesis data X merupakan suatu kelas spesifik

- $P(H | X)$: probabilitas hipotesis H berdasakaan kondisi X
 $P(H)$: probabilitas hipotesis H
 $P(X | H)$: probabilitas X berdasarkan kondisi pada hipotesis H
 $P(X)$: probabilitas hipotesis X

2.4. Algoritma k-Nearest Neighbors (kNN)

k-Nearest Neighbors didasarkan pada fungsi jarak yang mengukur perbedaan atau kesamaan antara dua contoh. kNN termasuk dalam contoh *lazy learning*. *Lazy learning* menyimpan data percobaan (*training data*) pada saat percobaan dan menunda pembelajaran (*learning*) sampai saat pengklasifikasian. Menurut Chumuang, *et al* [16], tahap perhitungan dengan algoritma kNN, sebagai berikut:

- Tentukan banyaknya tetangga k (sebaiknya ganjil)
- Hitung arak dari data untuk dibandingkan dengan dataset *training*, dapat dihitung dengan persamaan jarak Euclidean

$$\text{dist}(p,q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2)$$

dimana:

$\text{dist}(p,q)$ = jarak antara p dan l

p_i = nilai ke-i pada data p

q_i = nilai ke-i pada data q

- Atur urutan menaik dari jarak (urutkan dari kecil ke besar) dan pilih himpunan k paling sedikit dari dataset terkecil
- Tentukan bahwa jawaban dengan data yang akan diprediksi adalah kelompok data yang memiliki jumlah k pertama dari kumpulan data terbesar
- Tetapkan kelas kelas terdekat dengan titik pertimbangan

2.5. Evaluasi Hasil

Evaluasi hasil dilakukan dengan *confusion matrix*. *Confusion matrix* merupakan sebuah tabel yang menyatakan jumlah data uji yang diklasifikasikan secara benar dan jumlah data uji yang diklasifikasikan secara salah [17]. Pengertian lain, *confusion matrix* merupakan sebuah matriks yang menampilkan visualisasi kinerja dari algoritma klasifikasi menggunakan data dalam matriks yang membagi klasifikasi prediksi dalam bentuk *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Adapun bentuk confusion matriks untuk klasifikasi dua kelas, dapat dilihat pada Tabel 2 [18].

Tabel 2. *Confusion Matrix*

Kelas	Nilai Aktual	
	Positif	Negatif
Nilai Positif	<i>True Positive</i> (TP)	<i>False Positive</i> (FP)
Nilai Negatif	<i>False Negative</i> (FN)	<i>True Negative</i> (TN)

Keterangan [19]:

- True Positive* (TP) adalah jumlah data positif yang diklasifikasikan positif.
- False Negative* (FN) adalah jumlah data negatif yang diklasifikasikan positif.
- False Positive* (FP) adalah jumlah data positif yang diklasifikasikan negatif.
- True Negative* (TN) adalah jumlah data negatif yang diklasifikasikan negatif.

Dari *confusion matrix* dapat dihitung akurasi, presisi, dan *recall*. Ketepatan klasifikasi dapat dilihat dari akurasi klasifikasi. Akurasi klasifikasi menunjukkan performasi model klasifikasi secara keseluruhan. Semakin tinggi akurasi klasifikasi, semakin baik performasi model klasifikasi dan sebaliknya. Rumus untuk menghitung akurasi dapat dilihat pada persamaan (3) [20]:

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (3)$$

Presisi merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Rumus untuk menghitung presisi dapat dilihat pada persamaan (4) [21]:

$$\text{Presisi} = \frac{TP}{TP + FP} \times 100\% \quad (4)$$

Recall merupakan tingkat keberhasilan sistem dalam meemukan kembali sebuah informasi. Rumus untuk menghitung recall dapat dilihat pada persamaan (5) [20]:

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \quad (5)$$

2.6. Analisis Hasil

Proses ini merupakan proses terakhir dari keseluruhan penelitian ini. Nilai akurasi, presisi, dan *recall* yang dihasilkan oleh pengujian menggunakan algoritma Naïve Bayes dan k-Nearest Neighbors akan dibandingkan untuk mengetahui algoritma mana yang memiliki keakuratan lebih baik dalam menciptakan model.

3. HASIL DAN PEMBAHASAN

3.1. Hasil

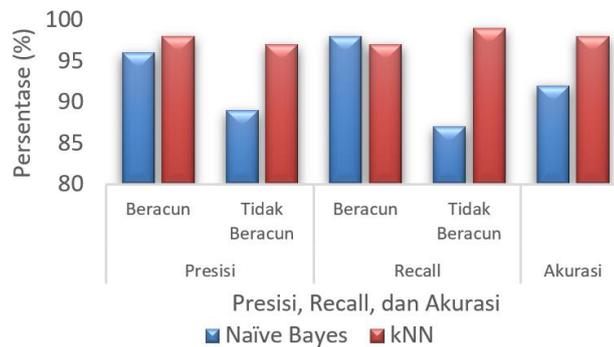
Penerapan algoritma Naïve Bayes dan k-Nearest Neighbors pada klasifikasi jamur beracun memberikan hasil yang berbeda. Confusion matriks hasil penerapan algoritma Naïve Bayes dan k-Nearest Neighbors pada dataset Klasifikasi Jamur dapat dilihat pada Tabel 3.

Tabel 3. Confusion Matrix Naïve Bayes dan k Nearest Neighbor

Naïve Bayes				k-Nearest Neighbors			
Kelas		Nilai Aktual		Kelas		Nilai Aktual	
		Beracun	Tidak Beracun			Beracun	Tidak Beracun
Nilai Prediksi	Beracun	836	27	Nilai Prediksi	Beracun	851	12
	Tidak Beracun	103	659		Tidak Beracun	22	740

Dari Tabel 3 dapat dilihat bahwa Naïve Bayes memprediksi 836 jamur beracun sebagai jamur beracun, 103 jamur beracun sebagai tidak jamur beracun, 27 jamur tidak beracun sebagai jamur beracun, dan 659 jamur beracun sebagai jamur beracun. Algoritma k-Nearest Neighbors memprediksi 851 jamur beracun sebagai jamur beracun, 22 jamur beracun sebagai tidak jamur beracun, 12 jamur tidak beracun sebagai jamur beracun, dan 740 jamur beracun sebagai jamur beracun.

Algoritma Naïve Bayes memprediksi 1.495 secara tepat dan 130 data ditebak pada kelas yang salah. Akurasi yang diperoleh dari penerapan algoritma Naïve Bayes adalah sebesar 92%. Nilai presisi untuk kelas beracun adalah 96% dan untuk kelas tidak beracun 89%. Nilai *recall* untuk kelas beracun adalah 86% dan 97% untuk kelas tidak beracun. Algoritma k-Nearest Neighbors memprediksi 1.591 secara tepat dan 34 data ditebak pada kelas yang salah. Akurasi yang diperoleh dari penerapan algoritma k-Nearest Neighbors adalah sebesar 98%. Nilai presisi untuk kelas beracun adalah 98% dan untuk kelas tidak beracun 97%. Nilai recall untuk kelas beracun dan tidak beracun masing-masing adalah 97% dan 99% untuk kelas tidak beracun. Secara lebih jelas, nilai-nilai tersebut dapat dilihat pada Gambar 1.

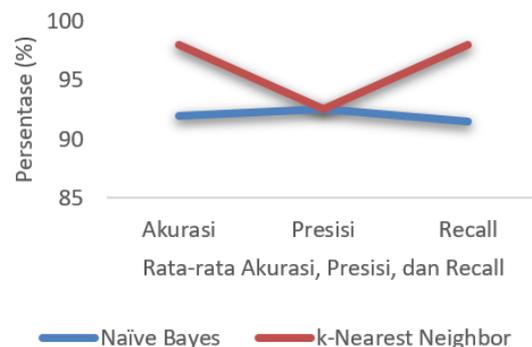


Gambar 2. Nilai-nilai Akurasi, Presisi, dan Recall Naïve Bayes dan kNN

Dari Gambar 2 dapat diketahui bahwa presisi terbaik dihasilkan algoritma k-Nearest Neighbors dalam menentukan jamur beracun, nilainya sebesar 98%. Recall terbaik dihasilkan algoritma k Nearest Neighbor dalam menentukan jamur tidak beracun, yaitu sebesar 99%. Akurasi terbaik dihasilkan oleh algoritma k-Nearest Neighbors, yaitu 98%.

3.2. Perbandingan Hasil Kedua Metode

Hasil prediksi algoritma Naïve Bayes dan k-Nearest Neighbors menunjukkan bahwa kedua algoritma tersebut bekerja dengan baik dalam mengklasifikasikan jamur beracun. Dalam algoritma Naïve Bayes, terlihat variabel-variabel yang paling berpengaruh karena algoritma ini menghitung peluang setiap kejadian. Hasil akurasi, presisi, dan recall kedua algoritma tersebut akan dicari rata-ratanya untuk menentukan algoritma terbaik dalam klasifikasi jamur beracun. Algoritma Naïve Bayes memiliki rata-rata akurasi 92%, rata-rata presisi 92,5%, dan recall 91,5%. Algoritma k-Nearest Neighbors menghasilkan rata-rata akurasi 98%, presisi 92,5%, dan recall 98%. Perbandingan rata-rata nilai rata-rata akurasi, presisi, dan recall dapat dilihat pada Gambar 3.



Gambar 3. Perbandingan Rata-rata Akurasi, Presisi, dan Recall

Dari Gambar 3 dapat dilihat bahwa nilai presisi dan recall yang dihasilkan algoritma Naïve Bayes dan k-Nearest Neighbors tidak terlalu berbeda dan sangat baik digunakan dalam pengklasifikasian jamur beracun karena berada di atas 90%. Rata-rata akurasi dan recall algoritma k-Nearest Neighbors lebih baik dibanding Naïve Bayes, yaitu masing-masing bernilai 98% sedangkan rata-rata presisi untuk kedua algoritma adalah sama, yaitu sebesar 92,5%.

4. KESIMPULAN

Dari penelitian ini dapat diketahui bahwa pengklasifikasian dataset jamur dengan menggunakan algoritma Naïve Bayes dan k-Nearest Neighbors sangat baik karena nilai rata-rata keseluruhannya berada di atas 90%. Rata-rata akurasi dan recall algoritma Naïve Bayes lebih besar dibanding algoritma k-Nearest Neighbors. Rata-rata presisi algoritma Naïve Bayes dan k-Nearest Neighbors sama besarnya. Presisi terbaik dimiliki oleh algoritma k Nearest Neighbor dalam memprediksi jamur beracun. Recall

terbaik dimiliki oleh algoritma k-Nearest Neighbors dalam memprediksi jamur tidak beracun. Berdasarkan rata-rata akurasi, presisi, dan recall kedua algoritma, dapat disimpulkan bahwa algoritma k-Nearest Neighbors lebih baik dibanding algoritma Naïve Bayes dalam klasifikasi jamur beracun.

DAFTAR PUSTAKA

- [1] J. B. Reece, L. A. Urry, S. A. Cain, Michael L. Wasserman, P. V. Minorsky, and R. B. Jackson, *Campbell Biology Tenth Edition*, 10th ed. Benjamin Cummings, 2013.
- [2] D. L. Hawksworth, "The Magnitude of Fungal Diversity: The 1.5 Million Species Estimate Revisited," *Mycol. Res.*, vol. 105, no. 12, pp. 1422–1432, Dec. 2001, doi: 10.1017/S0953756201004725.
- [3] E. Guillamón *et al.*, "Edible Mushrooms: Role in the Prevention of Cardiovascular Diseases," *Fitoterapia*, vol. 81, no. 7, pp. 715–723, Oct. 2010, doi: 10.1016/j.fitote.2010.06.005.
- [4] Hasanuddin, "Jenis Jamur Kayu Makroskopis Sebagai Media Pembelajaran Biologi (Studi di TNGL Blangjerango Kabupaten Gayo Lues)," *Biot. J. Ilm. Biol. Teknol. dan Kependidikan*, vol. 2, no. 1, p. 38, Feb. 2018, doi: 10.22373/biotik.v2i1.234.
- [5] E. Tyasrini, T. Winata, and S. Susantina, "Hubungan Antara Sifat Dan Metabolit Candida Spp. Dengan Patogenesis Kandidiasis," *Maranatha J. Med. Heal.*, vol. 6, no. 1, 2006.
- [6] F. A. D. Aji Prasetya Wibawa, Muhammad Guntur Aji Purnama, Muhammad Fathony Akbar, "Metode-metode Klasifikasi," *Pros. Semin. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 1, p. 134, 2018.
- [7] A. Wibowo, Y. Rahayu, A. Riyanto, and T. Hidayatulloh, "Classification Algorithm for Edible Mushroom Identification," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, Mar. 2018, pp. 250–253, doi: 10.1109/ICOIACT.2018.8350746.
- [8] P. Maurya and N. P. Singh, "Mushroom Classification Using Feature-Based Machine Learning Approach," 2020, pp. 197–206.
- [9] Syarli and A. A. Muin, "Metode Naive Bayes Untuk Prediksi Kelulusan," *J. Ilm. Ilmu Komput.*, vol. 2, no. 1, pp. 22–26, 2016, [Online]. Available: <https://media.neliti.com/media/publications/283828-metode-naive-bayes-untuk-prediksi-kelulu-139fcea.pdf>.
- [10] H. Annur, "Klasifikasi Masyarakat Miskin Menggunakan Metode Naive Bayes," *Ilk. J. Ilm.*, vol. 10, no. 2, pp. 160–165, Aug. 2018, doi: 10.33096/ilkom.v10i2.303.160-165.
- [11] L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of Improving K-Nearest-Neighbor for Classification," in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, 2007, pp. 679–683, doi: 10.1109/FSKD.2007.552.
- [12] S. Mutrofin, A. Izzah, A. Kurniawardhani, and M. Masrur, "Optimasi Teknik Klasifikasi Modified K Nearest Neighbor Menggunakan Algoritma Genetika," *J. Penelit. Eksakta*, vol. 10, no. 1, pp. 130–134, 2014.
- [13] Y. I. Kurniawan and T. I. Barokah, "Klasifikasi Penentuan Pengajuan Kartu Kredit Menggunakan K-Nearest Neighbor," *J. Ilm. Matrik*, vol. 22, no. 1, pp. 73–82, Mar. 2020, doi: 10.33557/jurnalatrik.v22i1.843.
- [14] H. Junaedi, H. Budianto, I. Maryati, and Y. Melani, "Data Transformation pada Data Mining," *Pros. Konf. Nas. Inov. dalam Desain dan Teknol.*, vol. 7, pp. 93–99, 2011.
- [15] A. Z. A. M. Bajabir, "Penerapan Metode Naive Bayes untuk Prediksi Menentukan Karyawan Tetap pada PT. YSP Industries Indonesia," *Sekol. Tinggi Teknol. Pelita Bangsa*, vol. 72, pp. 1–62, 2018.
- [16] N. L. Ratniasih, "Optimasi Data Mining Menggunakan Algoritma Naive Bayes Dan C4.5 Untuk Klasifikasi Kelulusan Mahasiswa," *J. Teknol. Inf. dan Komput.*, vol. 5, no. 1, pp. 28–34, 2019, doi: 10.36002/jutik.v5i1.634.
- [17] N. Chumuang *et al.*, "Mushroom Classification by Physical Characteristics by Technique of k-

- Nearest Neighbor,” in *2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, Nov. 2020, pp. 1–6, doi: 10.1109/iSAI-NLP51646.2020.9376820.
- [18] A. Indriani, “Klasifikasi Data Forum dengan menggunakan Metode Naïve Bayes Classifier,” *Semin. Nas. Apl. Teknol. Inf.*, pp. 1–10, 2014.
- [19] O. Arifin and T. B. Sasongko, “Analisa Perbandingan Tingkat Performansi Metode Support Vector Machine dan Naive Bayes Classifier untuk Klasifikasi Jalur Minat SMA,” *Semin. Nas. Teknol. Inf. dan Multimed. 2018*, pp. 67–72, 2018.
- [20] B. P. Pratiwi, A. S. Handayani, and S. Sarjana, “Pengukuran Kinerja Sistem Kualitas Udara dengan Teknologi WSN Menggunakan Confusion Matrix,” *J. Inform. Upgris*, vol. 6, no. 2, Jan. 2021, doi: 10.26877/jiu.v6i2.6552.
- [21] E. Prasetyo, *Data Mining: Konsep dan Aplikasi menggunakan MATLAB*. Yogyakarta: ANDI, 2012.

Halaman Ini Dikosongkan