

Analisis Perbandingan Algoritma C4.5 dan Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Jamur

R Rahmadhani, Alwis Nazir*, Fadhilah Syafria, Liza Afriyanti

Fakultas Sains dan Teknologi, Prodi Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim, Pekanbaru, Indonesia

Email: ¹11950115165@students.uin-suska.ac.id, ²alwis.nazir@uin-suska.ac.id, ³fadhilah.syafria@uin-suska.ac.id, ⁴liza.afriyanti@uin-suska.ac.id

Email Penulis Korespondensi: alwisnazir@uin-suska.ac.id

Submitted: 29/11/2023; Accepted: 19/12/2023; Published: 22/12/2023

Abstrak—Jamur adalah organisme yang terdiri dari beberapa sel, mengandung spora, bersifat eukariotik (memiliki membran inti sel), dan tidak memiliki klorofil sehingga, jamur bergantung pada organisme lain untuk mendapatkan makanan. Jamur memiliki bentuk yang sangat identik mulai dari ukuran, bentuk, bau, maupun warna. Sehingga sulit bagi orang awam membedakan mana jamur beracun dan jamur tidak beracun. Kesalahan dalam mengidentifikasi jamur dapat berakibat fatal karena ini bisa saja menyebabkan keracunan ketika mengonsumsi jamur. Maka dari itu, perlunya edukasi dalam melakukan klasifikasi jamur beracun dan jamur tidak beracun. Dengan menerapkan berbagai algoritma klasifikasi, ini dapat mengukur performa mana yang lebih baik dari tiap-tiap algoritma tersebut. Pada penelitian sebelumnya yang dilakukan oleh beberapa peneliti dalam mengklasifikasi jamur terdapat perbedaan hasil akurasi pada masing-masing algoritma. Oleh karena itu, pada penelitian ini akan mengangkat pertanyaan tentang bagaimana mengukur atau melakukan perbandingan performa algoritma pada klasifikasi dengan menggunakan algoritma C4.5 dan algoritma Modified K-Nearest Neighbor (MKNN). Hasil yang diperoleh dalam melakukan perbandingan performa algoritma C4.5 dan algoritma Modified K-Nearest Neighbor (MKNN) pada penelitian ini, dimana algoritma C4.5 berhasil memperoleh tingkat akurasi sebesar 98,52%, presisi sebesar 98,55%, recall sebesar 98,52%, dan f1-score sebesar 98,51%. Sebaliknya, algoritma Modified K-Nearest Neighbor (MKNN) dengan menggunakan nilai K=10 meraih tingkat akurasi sebesar 96,62%, presisi sebesar 96,69%, recall sebesar 96,62%, dan nilai f1-score sebesar 96,57%.

Kata Kunci: Jamur; Klasifikasi; Perbandingan; Algoritma C4.5; Algoritma Modified K-Nearest Neighbor (MKNN)

Abstract—Mushrooms are organisms that consist of several cells, contain spores, are eukaryotic (have a cell nucleus membrane), and do not have chlorophyll, so fungi depend on other organisms to get food. Mushrooms have very identical shapes, starting with size, shape, smell, and color. So it is difficult for ordinary people to differentiate between poisonous mushrooms and non-poisonous mushrooms. Mistakes in identifying mushrooms can have fatal consequences because they can cause poisoning when consuming mushrooms. Therefore, there is a need for education in classifying poisonous and non-poisonous mushrooms. By applying various classification algorithms, it can be determined which algorithm performs better. In previous research conducted by several researchers on classifying mushrooms, there were differences in the accuracy results for each algorithm. Therefore, this research will raise the question of how to measure or comparison algorithm performance in classification using the C4.5 algorithm and the Modified K-Nearest Neighbor (MKNN) algorithm. The results obtained by comparison the performance of the C4.5 algorithm and the Modified K-Nearest Neighbor (MKNN) algorithm in this research show that the C4.5 algorithm managed to obtain an accuracy level of 98.52%, precision of 98.55%, recall of 98.52%, and f1-score of 98.51%. In contrast, the Modified K-Nearest Neighbor (MKNN) algorithm using the value K=10 achieved an accuracy level of 96.62%, precision of 96.69%, recall of 96.62%, and f1-score value of 96.57%.

Keywords: Mushrooms; Classification; Comparison; C4.5 Algorithm; Modified K-Nearest Neighbor (MKNN) Algorithm

1. PENDAHULUAN

Jamur adalah organisme yang terdiri dari beberapa sel, mengandung spora, mempunyai sifat eukariotik atau memiliki membran inti sel, dan tidak memiliki klorofil sehingga jamur bergantung pada organisme lain untuk mendapatkan makanan [1], [2]. Komponen makanan seperti selulosa, glukosa, lignin, protein, dan senyawa pati diperoleh jamur dari organisme lain dan dimanfaatkan selama proses fotosintesis [3]. Jamur memiliki berbagai jenis yang hampir serupa mulai dari bentuk, ukuran, bau, maupun warna. Hal ini menyebabkan kesulitan membedakan mana jamur beracun dan jamur tidak beracun. Dampak dari kurangnya informasi ini mengakibatkan terjadinya kasus keracunan jamur yang bahkan bisa berakibat fatal dan menyebabkan kematian [4].

Dalam sebuah meta-analisis tahun 2005 yang melibatkan 28.018 kasus keracunan jamur dari tahun 1951 hingga 2002 di seluruh dunia, ditemukan bahwa frekuensi laporan keracunan jamur mengalami peningkatan yang signifikan [4]. Berdasarkan laporan dari Centers for Disease Control and Prevention (CDC) pada tahun 2008, lebih dari 41.000 orang meninggal akibat keracunan jamur yang tidak disengaja. Sementara itu, World Health Organization (WHO) telah mencatat 0,346 juta kematian akibat jamur sejak tahun 2004 [5]. Dalam kurun waktu 10 tahun (2010-2020), sebanyak 76 kasus keracunan jamur liar terjadi di seluruh Indonesia. Selama periode ini, 550 orang menjadi korban dan dilaporkan 9 orang diantaranya meninggal dunia [6], [7]. Informasi kasus keracunan jamur yang terdeteksi berasal dari 5 wilayah besar yaitu Sumatera, Jawa, Kalimantan, Sulawesi, dan Nusa Tenggara (NTB dan NTT) [7].

Korban keracunan jamur yang meninggal antara lain 3 orang dari Sumatera Selatan dan 2 orang dari Jawa Timur pada tahun 2013, lalu 1 orang dari Jawa Tengah pada tahun 2015, 2 orang dari Kalimantan Barat di tahun

2017, dan 1 orang dari Jawa Tengah pada tahun 2018 [7]. Berdasarkan permasalahan yang sudah dipaparkan dimana dengan terjadinya peningkatan jumlah kasus keracunan jamur telah mendorong pentingnya melakukan penelitian ini untuk mengklasifikasikan jamur beracun dan jamur tidak beracun sebagai upaya peningkatan pengetahuan masyarakat tentang jamur dan dapat menghindari terjadinya kasus keracunan jamur yang mana dapat menyebabkan kematian. Pada penelitian ini salah satu bidang ilmu yang bisa digunakan untuk identifikasi tersebut adalah data mining.

Data mining adalah proses penerapan metode tertentu untuk menemukan pola atau informasi menarik dalam kumpulan data [8]. Data mining juga disebut sebagai proses memanfaatkan penalaran pola, statistik, dan teknologi matematika untuk mencari korelasi, pola, dan tren dalam database besar [9], [10]. Beberapa teknik tersebut dimana ini mencakup teknik deskripsi, estimasi, prediksi, klasifikasi, klustering, dan asosiasi [10]. Pada penelitian ini, analisis data mining akan dilakukan melalui penerapan klasifikasi.

Klasifikasi merupakan tahapan di mana data dipetakan ke dalam kelompok atau kelas yang sudah ditentukan sebelumnya, tujuannya adalah untuk mengenali pola atau fungsi yang mampu meramalkan konsep atau kelas data yang labelnya masih belum diketahui. [11], [12], [13]. Pada penelitian ini, teknik klasifikasi dimanfaatkan untuk memprediksi mana jamur beracun dan jamur tidak beracun. Metode klasifikasi yang akan diterapkan dalam penelitian ini melibatkan perbandingan antara dua algoritma klasifikasi, yakni algoritma C4.5 dan Modified K-Nearest Neighbor.

Algoritma C4.5 adalah algoritma yang dirancang untuk memisahkan dataset besar menjadi subset yang lebih kecil melalui penerapan aturan keputusan yang terstruktur [14]. Algoritma C4.5 digunakan dalam pembentukan pohon keputusan (Decision Tree) [15]. Secara umum, algoritma C4.5 menunjukkan tingkat akurasi yang signifikan, sebagaimana yang terungkap dalam hasil penelitian yang dilakukan oleh Anida Nirwana, Amril Mutoi Siregar, dan Rahmat dengan judul "Klasifikasi Permasalahan Kredit Macet Pada Bank Menggunakan Algoritma Decision Tree C4.5" dengan memperoleh hasil nilai accuracy sebesar 99,9753%, nilai untuk precision sebesar 100%, recall sebesar 99,8%, dan f-measure sebesar 99,9% [16].

Dalam penelitian berikutnya yang dilakukan oleh Nurfazriah Attamami, Agung Triyudi, dan Rima Tamara Aldisa yang berjudul "Analisis Performa Algoritma Klasifikasi Naïve Bayes dan C4.5 untuk Prediksi Penerima Bantuan Jaminan Kesehatan," menyimpulkan bahwa Algoritma C4.5 berhasil mencapai tingkat akurasi yang sangat tinggi, sebesar 99,04%, sementara algoritma Naïve Bayes mencapai tingkat akurasi sebesar 92,97%. [17]. Sementara itu algoritma Modified K-Nearest Neighbor (MKNN) merupakan hasil pengembangan dari algoritma K-Nearest Neighbor (KNN), yang melibatkan penambahan proses validasi pada data pelatihan dan penghitungan nilai weight voting untuk menentukan bobot perhitungan [18], [19].

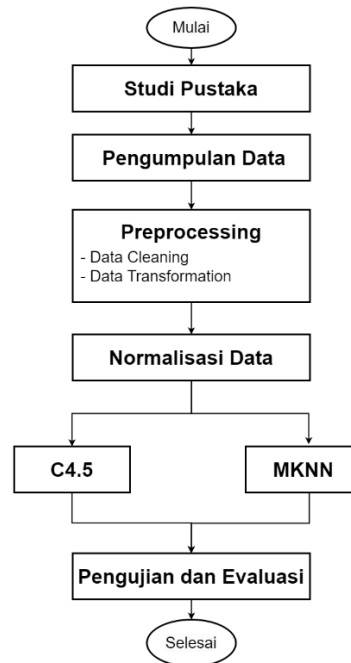
Penelitian yang menerapkan algoritma Modified K-Nearest Neighbor (MKNN) telah diteliti oleh Reza Chaidir, Hardian Oktavianto, dan Reni Umilasari dalam penelitian berjudul "Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) Untuk Klasifikasi Penerima Dana Bantuan COVID-19". Penelitian ini mencapai tingkat akurasi sekitar 94,92%, presisi sekitar 90,48%, dan recall sekitar 95% [20]. Penelitian yang dilakukan oleh Hadi Dwi Abdullah Hamid, Nurul Hidayat, dan Ratih Kartika Dewi dalam penelitian mereka mengenai Algoritma Modified K-Nearest Neighbor (MKNN) berjudul "Diagnosis Penyakit Tanaman Cabai Menggunakan Modified K-Nearest Neighbor (MKNN)" menyimpulkan bahwa pada $K=5$, tingkat akurasi yang dicapai adalah 94%. Sementara itu, untuk $K=8$, akurasi mencapai 92%. Pada uji coba dengan $K=11$ dan $K=14$, didapatkan tingkat akurasi sebesar 88%. [21].

Selanjutnya penelitian yang dilakukan oleh Bertho Erizal, Martaleti Bettiza, Alena Uperiaty berjudul "Pebandingan Algoritma Modified K-Nearest Neighbor dan Naïve Bayes untuk Klasifikasi Data Rekam Medis Berdasarkan ICD-10 (Studi Kasus: RSUD Ahmad Tabib Kota Tanjungpinang)" berhasil mencapai tingkat akurasi sekitar 70% pada Modified K-Nearest Neighbor, sementara Naïve Bayes mencapai 72%. Hasil ini diperoleh dari uji coba ketiga dengan melibatkan 89 data uji dan 365 dataset [22].

Pada penelitian ini, menggunakan dataset jamur yang diperoleh dari situs Kaggle.com. Pada dataset ini mempunyai 23 atribut dengan total data sebanyak 8124 data dimana mencakup dua kelas, yaitu jamur beracun (poison) sebanyak 3916 data dan jamur tidak beracun (edible) sebanyak 4208 data. Dengan membandingkan algoritma klasifikasi, dapat mengukur performa kinerja dari algoritma-algoritma tersebut untuk menentukan mana hasil dari algoritma yang lebih baik. Pada penelitian ini akan melakukan perbandingan antara algoritma C4.5 dan Modified K-Nearest Neighbor (MKNN) mengenai klasifikasi jamur dimana hasil yang ingin dicapai dari perbandingan kedua algoritma tersebut adalah untuk mengukur nilai akurasi, presisi, recall, dan f1-score.

2. METODOLOGI PENELITIAN

Metodologi penelitian adalah serangkaian langkah-langkah yang menghubungkan konsep-konsep yang akan diamati dan diteliti melalui penelitian yang sedang dilaksanakan. Pada bagian ini, akan dijelaskan mengenai tahapan-tahapan yang diterapkan dalam penelitian ini. Tahapan penelitian dapat ditemukan dalam ilustrasi pada gambar 1 berikut:



Gambar 1. Tahapan Metodologi Penelitian

Pada gambar 1 diatas, tahapan penelitian ini dimulai dengan mencari studi pustaka dari sumber referensi yang akurat, kemudian dilakukan pengumpulan data. Setelah data didapat maka dilakukan tahap preprocessing dimana pada penelitian ini menggunakan dua tahapan yaitu data cleaning dan data transformation. Kemudian dilanjutkan dengan menormalisasikan data. Setelah data dinormalisasi, maka dilakukan klasifikasi algoritma C4.5 dan algoritma Modified K-Nearest Neighbor (MKKN). Terakhir dilakukan tahapan pengujian dan evaluasi.

2.1 Studi Pustaka

Dalam tahap ini, dilakukan studi pustaka untuk menghimpun berbagai referensi terkait dengan topik penelitian. Referensi ini dapat diperoleh dari berbagai sumber seperti buku, jurnal, dan tugas akhir sebelumnya yang relevan dengan fokus penelitian. Proses ini dimaksudkan untuk memperkaya dasar argumen yang disajikan dengan merujuk pada teori-teori yang telah diajukan oleh ahli-ahli yang berkompeten.

2.2 Pengumpulan Data

Data yang digunakan pada penelitian ini berasal dari dataset jamur yang diperoleh dari situs Kaggle.com. Jumlah total data yang terdapat dalam dataset ini mencapai 8124, yang terbagi menjadi dua kelas, yakni jamur beracun (poison) sebanyak 3916 data dan jamur tidak beracun (edible) sebanyak 4208 data. Dataset ini terdiri dari 23 atribut, meliputi cap-shape, cap-surface, cap-color, gill-attachment, stalk-surface-below-ring, stalk-shape, ring-number, veil-type, gill-spacing, stalk-root, stalk-surface-above-ring, stalk-color-below-ring, veil-color, gill-size, stalk-color-above-ring, odor, bruises%3f, ring-type, gill-color, spore-print-color, population, habitat, dan class.

2.3 Preprocessing

Data yang terkumpul akan mengalami tahap preprocessing, yang melibatkan data cleaning dan data transformation. Proses data cleaning dalam penelitian ini bertujuan untuk mengatasi data yang hilang atau kosong dengan menggunakan metode modus, yaitu nilai yang paling sering muncul karena data bersifat objek (kategorikal). Mengingat data bersifat objek, dilakukan pula data transformation, di mana data yang semula bersifat objek diubah menjadi bentuk numerik.

2.4 Normalisasi Data

Normalisasi merupakan teknik yang penting dalam pengolahan data yang bertujuan untuk menyamakan skala nilai-nilai atribut agar berada dalam rentang tertentu. Dalam penelitian ini, penerapan normalisasi Min-Max digunakan. Normalisasi Min-Max bertujuan untuk mengubah rentang nilai atribut sehingga nilainya berada di antara 0 dan 1. Adapun perhitungan dari normalisasi min-max ditunjukkan pada rumus berikut [23]:

$$x_n = \frac{x_0 - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

Dimana:

x_n = Nilai atribut x yang akan dinormalisasi

x_0 = Nilai asli dari atribut x

x_{min} = Nilai terkecil pada atribut
 x_{max} = Nilai terbesar pada atribut

2.5 Algoritma C4.5

Algoritma C4.5 merupakan metode yang dirancang untuk mengelompokkan dataset besar menjadi subset yang lebih kecil melalui penerapan aturan keputusan yang terstruktur [14]. Algoritma ini digunakan untuk menciptakan struktur pohon keputusan yang mudah dimengerti, dapat diadaptasi dengan mudah, dan menarik karena dapat direpresentasikan dalam bentuk gambar [24]. Proses perhitungan algoritma ini melibatkan beberapa tahapan, dimulai dengan penentuan nilai entropy, kemudian menetapkan nilai gain. Struktur pohon keputusan kemudian dibangun berdasarkan hasil perhitungan ini [25]. Tahapan-tahapan dalam perancangan sebuah pohon keputusan menggunakan algoritma C4.5 yaitu [11]:

- a. Mempersiapkan data training
- b. Melakukan perhitungan entropy, dimana entropy digunakan sebagai indikator tingkat ketidakpastian, terutama dalam konteks variasi pilihan terkait dengan nilai atribut tertentu. Semakin besar nilai entropy maka semakin besar pula perbedaan pilihan (ketidakpastian). Untuk menghitung entropy dapat dilakukan dengan rumus berikut :

$$\text{Entropy}(S) = - \sum_{i=1}^n p_i * \log_2 p_i \tag{2}$$

Dimana :

S = Himpunan kasus
n = Jumlah Partisi S
 p_i = Proporsi S_i terhadap S

- c. Melakukan perhitungan gain, yang merupakan langkah dalam pemilihan atribut untuk menentukan akar pada pohon keputusan. Proses ini melibatkan perhitungan nilai pada gain untuk setiap atribut, dan atribut dengan nilai gain tertinggi akan dipilih sebagai akar pertama. Berikut adalah rumus perhitungan gain :

$$\text{Gain}(S, A) = - \sum_{i=1}^n \frac{|S_i|}{|S|} = \text{Entropy}(S_i) \tag{3}$$

Dimana :

S = Himpunan kasus
A = Atribut
n = Jumlah partisi atribut A
 $|S_i|$ = Jumlah kasus pada partisi ke-i
 $|S|$ = Jumlah kasus dalam himpunan S

- d. Ulangi langkah kedua dan ketiga secara berulang sampai semua cabang memiliki kelas yang sama. Proses percabangan akan berhenti jika:
 1. Semua kasus di simpul (node) n memiliki kelas yang sama.
 2. Pada skenario terpartisi, tidak ada lagi variabel independen yang tersisa.
 3. Tidak ada cabang yang memiliki kasus kosong.

2.6 Algoritma Modified K-Nearest Neighbor

Algoritma Modified K-Nearest Neighbor (MKKN) merupakan penyempurnaan dari algoritma K-Nearest Neighbor (KNN). Dalam MKKN, disertakan langkah validasi tambahan pada data pelatihan serta penghitungan nilai weight voting untuk menentukan bobot perhitungan. [18], [19]. Berikut adalah tahapan dalam perhitungan klasifikasi Modified K-Nearest Neighbor (MKKN) [22]:

- a. Menghitung jarak antara data training menggunakan Euclidean dengan rumus berikut:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \tag{4}$$

Keterangan:

$d(x, y)$ = Jarak Euclidean data x dengan data y
 x_i = Nilai atribut ke-i dari data x
 y_i = Nilai atribut ke-i dari data y

- b. Validitas digunakan sebagai tambahan informasi tentang data dan diterapkan pada variabel dalam menghitung weight voting dengan menggunakan rumus berikut:

$$\text{Validity}(x) = \frac{1}{H} \sum_{i=1}^n s(\text{label}(x), \text{label}(N_i(x))) \tag{5}$$

Keterangan:

H = Jumlah titik terdekat
Label(x) = Kelas x
Label($N_i(x)$) = Label kelas titik terdekat x

- c. Weight voting (nilai bobot) digunakan untuk memberi nilai bobot yang dilakukan pada setiap data testing dengan berikut:

$$W(x) = \text{Validity}(x) \frac{1}{d+\alpha} \tag{6}$$

Keterangan :

W(x) = Perhitungan weight voting

Validity(x) = Nilai Validity

d = Jarak Euclidean

α = Regular smoothing akan menggunakan nilai 0.5

2.7 Pengujian dan Evaluasi

Pada penelitian ini, pengujian dilakukan dengan menerapkan Confusion Matrix. Confusion Matrix adalah hasil evaluasi berbasis tabel dari klasifikasi data mining. Bentuk umum Confusion Matrix dapat dilihat pada tabel 1 berikut [26]:

Tabel 1. Tabel Confusion Matrix

	Kelas Prediksi	
	Positif	Negatif
Kelas Sebenarnya	TP	FN
	FP	TN

Keterangan :

TP : Data bernilai positif pada kelas sebenarnya diprediksi sebagai data bernilai positif

FN : Data bernilai positif pada kelas sebenarnya tapi diprediksi sebagai data bernilai negatif

FP : Data bernilai negatif pada kelas sebenarnya tapi diprediksi sebagai data bernilai positif

TN : Data bernilai negatif pada kelas sebenarnya diprediksi sebagai data bernilai negatif

Berdasarkan tabel 1 diatas, pengukuran kinerja Confusion Matrix memiliki empat istilah yaitu True Positive (TP), False Negative (FN), False Positive (FP), dan True Negative (TN). Hasil evaluasi dalam penelitian ini dimana akan membandingkan hasil akurasi, presisi, recall, dan f1-score dari klasifikasi algoritma C4.5 dan Modified K-Nearest Neighbor (MKKN). Rumus perhitungan akurasi, presisi, recall, dan f1-score yaitu sebagai berikut [5]:

- a. Akurasi dengan rumus berikut:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

Keterangan :

TP : Data bernilai positif pada kelas sebenarnya diprediksi sebagai data bernilai positif

FN : Data bernilai positif pada kelas sebenarnya tapi diprediksi sebagai data bernilai negatif

FP : Data bernilai negatif pada kelas sebenarnya tapi diprediksi sebagai data bernilai positif

TN : Data bernilai negatif pada kelas sebenarnya diprediksi sebagai data bernilai negatif

- b. Presisi dengan rumus berikut:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{8}$$

Keterangan :

TP : Data bernilai positif pada kelas sebenarnya diprediksi sebagai data bernilai positif

FP : Data bernilai negatif pada kelas sebenarnya tapi diprediksi sebagai data bernilai positif

- c. Recall dengan rumus berikut:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{9}$$

Keterangan :

TP : Data bernilai positif pada kelas sebenarnya diprediksi sebagai data bernilai positif

FN : Data bernilai positif pada kelas sebenarnya tapi diprediksi sebagai data bernilai negatif

- d. F1-Score dengan rumus berikut:

$$F1 - \text{Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

3. HASIL DAN PEMBAHASAN

3.1 Pengolahan Data

Pada tahap ini, dilakukan persiapan data awal yang diperoleh dari situs Kaggle.com. Dataset yang diambil dari situs tersebut digunakan sebagai sumber informasi dalam penelitian ini. Data yang digunakan mencakup informasi

tentang jamur dengan total 23 atribut. Beberapa atribut tersebut meliputi: cap-shape, cap-surface, cap-color, gill-attachment, stalk-surface-below-ring, stalk-shape, ring-number, veil-type, gill-spacing, stalk-root, stalk-surface-above-ring, stalk-color-below-ring, veil-color, gill-size, stalk-color-above-ring, odor, bruises%3f, ring-type, gill-color, spore-print-color, population, habitat, dan class. Jumlah data dalam dataset jamur ini mencapai 8124, yang terdiri dari dua kelas, yaitu jamur beracun (poison) sebanyak 3916 dataset dan jamur tidak beracun (edible) sebanyak 4208 dataset. Detail data jamur penelitian tersedia dalam tabel 2 berikut:

Tabel 2. Dataset Penelitian

	cap-shape	cap-surface	cap-color	...	spore-print-color	population	habitat	class
1	convex	smooth	brown	...	black	scattered	urban	poison
2	convex	smooth	yellow	...	brown	numerous	gresses	edible
3	bell	smooth	white	...	brown	numerous	meadow	edible
4	convex	scaly	white	...	black	scattered	urban	poison
5	convex	smooth	gray	...	brown	abudant	gresses	edible
6	convex	scaly	yellow	...	black	numerous	gresses	edible
7	bell	smooth	white	...	black	numerous	meadow	edible
8	bell	scaly	white	...	brown	scattered	meadow	edible
9	convex	scaly	white	...	black	several	gresses	poison
10	bell	smooth	yellow	...	black	scattered	meadow	edible
...
8115	flat	scaly	cinnamn	...	white	cluseted	woods	poison
8116	convex	smooth	brown	...	orange	several	leaves	edible
8117	knibbed	scaly	brown	...	white	several	leaves	poison
8118	knibbed	smooth	red	...	white	several	woods	poison
8119	knibbed	scaly	brown	...	white	several	woods	poison
8120	knibbed	smooth	brown	...	buff	cluseted	leaves	edible
8121	convex	smooth	brown	...	buff	several	leaves	edible
8122	flat	smooth	brown	...	buff	cluseted	leaves	edible
8123	knibbed	scaly	brown	...	white	several	leaves	poison
8124	convex	smooth	brown	...	orange	cluseted	leaves	edible

Pada tabel 2 diatas, ditunjukkan dataset jamur yang digunakan pada penelitian ini dimana dataset akan dilakukan tahap preprocessing data agar dapat dilakukan perhitungan klasifikasi. Preprocessing yang dipakai dalam penelitian ini terbagi atas dua tahap yaitu cleaning data dan transformation data. Pentingnya cleaning data dalam penelitian ini sangatlah signifikan. Cleaning data merupakan tahap pembersihan data terhadap nilai yang kosong, tidak lengkap, missing value, serta melakukan balance pada data yang unbalance untuk meningkatkan efisien dalam melakukan pengolahan data dan pengurangi tingkat error yang belih rendah. Data yang bersih dan terstruktur merupakan aspek yang krusial dalam memastikan validitas hasil penelitian. Pada proses cleaning data, terdapat 2480 dataset yang missing value. Sehingga dilakukan cleaning data dengan menerapkan modus yaitu nilai yang sering muncul untuk mengisi nilai-nilai yang mengalami missing value.

Kemudian dilakukan transformation data yang mana merupakan proses mengubah data dalam kategori tertentu. Transformation data yang diterapkan pada penelitian ini menggunakan teknik label encoding yang merupakan teknik pra-pemrosesan yang digunakan untuk mengubah nilai yang data yang awalnya berupa katagori (objek) menjadi data numerik agar dapat melakukan perhitungan klasifikasi. Dataset jamur yang sudah dilakukan transformation data dapat dilihat pada tabel 3 berikut:

Tabel 3. Transformasi Data

	cap-shape	cap-surface	cap-color	...	spore-print-color	population	habitat	class
1	2	3	0	...	0	3	4	poison
2	2	3	9	...	1	2	0	edible
3	0	3	8	...	1	2	2	edible
4	2	2	8	...	0	3	4	poison
5	2	3	3	...	1	0	0	edible
...
8120	4	3	0	...	2	1	1	edible
8121	2	3	0	...	2	4	1	edible
8122	3	3	0	...	2	1	1	edible
8123	4	2	0	...	7	4	1	poison
8124	2	3	0	...	5	1	1	edible

Berdasarkan tabel 3 diatas, dimana menampilkan hasil pada dataset jamur dengan jumlah data sebanyak 8124 data yang sudah dilakukan transformation data. Setelah melakukan transformation data maka tahap

selanjutnya akan dilakukan normalisasi data. Pada penelitian akan menggunakan normalisasi min-max, dimana ini mengubah menjadi nilai minimal dan maksimal. Nilai minimal yang digunakan adalah 0 dan nilai maksimal yang digunakan adalah 1. Berikut merupakan contoh dari perhitungan normalisasi min-max pada atribut cap-shape.

Diketahui :

$$x_0 = 2$$

$$x_{\min} = 0$$

$$x_{\max} = 5$$

$$x_n = \frac{x_0 - x_{\min}}{x_{\max} - x_{\min}}$$

$$x_n = \frac{2 - 0}{5 - 0}$$

$$x_n = 0.4$$

$$x_n = 0.4$$

Dari contoh perhitungan manual normalisasi min-max pada salah satu atribut jamur yaitu cap-shape diatas, dimana menghasilkan nilai atribut yaitu 0,4. Kemudian dilanjutkan proses normalisasi min-max keseluruhan atribut pada dataset jamur sehingga dihasil pada perhitungan dari normalisasi data pada tabel 4:

Tabel 4. Normalisasi Data

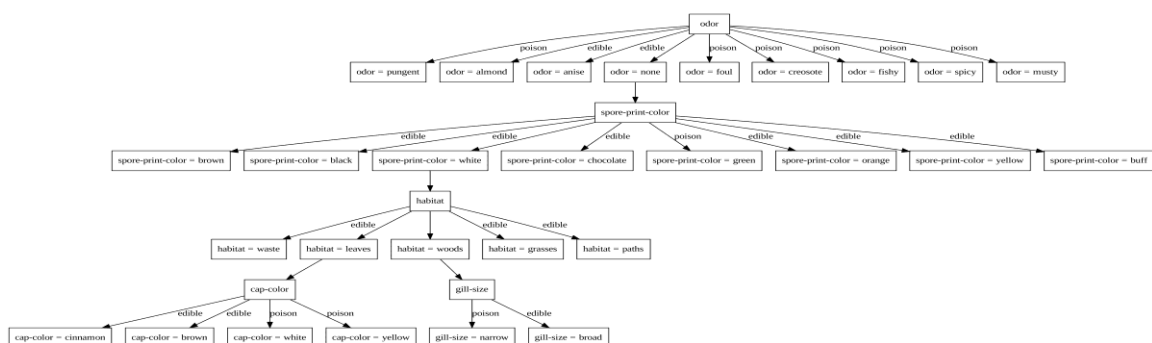
	cap-shape	cap-surface	cap-color	...	spore-print-color	population	habitat	class
1	0.4	1.0	0.0	...	0.0	0.6	0.6	poison
2	0.4	1.0	1.0	...	0.1	0.4	0.0	edible
3	0.0	1.0	0.8	...	0.1	0.4	0.3	edible
4	0.4	0.6	0.8	...	0.0	0.6	0.6	poison
5	0.4	1.0	0.3	...	0.1	0.0	0.0	edible
...
8120	0.8	1.0	0.0	...	0.3	0.2	0.2	edible
8121	0.4	1.0	0.0	...	0.3	0.8	0.2	edible
8122	0.6	1.0	0.0	...	0.3	0.2	0.2	edible
8123	0.8	0.6	0.0	...	0.9	0.8	0.2	poison
8124	0.4	1.0	0.0	...	0.6	0.2	0.2	edible

Pada tabel 4 diatas, dimana merupakan hasil dari dilakukannya proses normalisasi data. Dengan dilakukan normalisasi data dengan menerapkan teknik normalisasi min-max akan mengubah rentang nilai atribut sehingga nilainya berada di antara 0 hingga 1. Kemudian dari jumlah data jamur sebanyak 8124 dataset, data dibagi menjadi dua dengan persentase 80% data training dan 20% data testing. Dimana 80% dari jumlah data 8124 diperoleh data training sebanyak 6499 dataset dan 20% dari jumlah data 8124 diperoleh data testing sebanyak 1625 dataset. Dari data yang diperoleh selanjutnya akan dilakukan perhitungan klasifikasi algoritma C4.5 dan Modified K-Nearest Neighbor (MKKN) untuk memperoleh hasil akurasi, presisi, recall, dan f1-score dari masing-masing algoritma tersebut, dimana hasil ini akan dilakukan perbandingan.

3.2 Pengujian dan Pemodelan

3.2.1 Algoritma C4.5

Dalam tahap ini, data pelatihan diolah menerapkan algoritma C4.5 dengan tujuan dapat menghasilkan model aturan yang akan membentuk struktur pohon keputusan. Tahap awal dalam pembentukan pohon keputusan melibatkan penghitungan jumlah kelas jamur dan total entropy untuk keseluruhan dataset. Setelah mendapatkan nilai total entropy, langkah berikutnya adalah menghitung nilai entropy dan gain dari setiap atribut. Nilai gain tertinggi akan menentukan akar dari pohon keputusan. Terakhir, dilakukan proses berulang-ulang sampai semua atribut memiliki kelas yang sama. Berikut pohon keputusan yang ditunjukkan pada gambar 2 dibentuk dari hasil perhitungan.



Gambar 2. Pohon Keputusan

Berdasarkan gambar 2 dapat dilihat pohon keputusan yang dihasilkan menunjukkan node akar adalah odor. Meskipun pada awalnya terdapat 23 atribut, dimana setelah dilakukan perhitungan model algoritma C4.5 terdapat atribut paling berpengaruh yaitu: odor, spore-print-color, habitat, cap-color, dan gill-size. Model atau aturan ini nantinya akan digunakan untuk memprediksi data baru.

3.2.2 Algoritma Modified K-Nearest Neighbor (MKNN)

Dalam tahap ini, sejumlah eksperimen dibuat untuk mencapai variasi hasil akurasi. Pengujian akan melibatkan penggunaan 10 nilai K yang berbeda untuk menentukan K yang optimal. Nilai-nilai K yang diuji antara lain: K=10, K=20, K=30, K=40, K=50, K=60, K=70, K=80, K=90, dan K=100. Hasil akurasi untuk setiap nilai K dapat ditemukan dalam tabel 5 berikut:

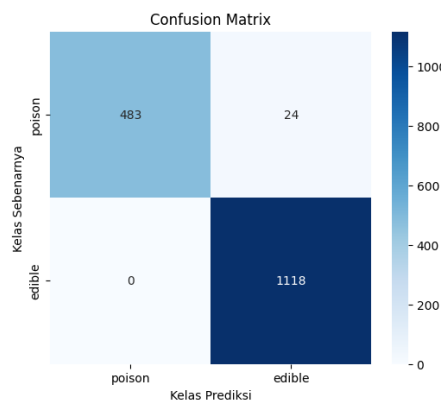
Tabel 5. Hasil Peforma Nilai K

Nilai K	Akurasi	Presisi	Recall	F1-Score
10	96,62%	96,69%	96,62%	96,57%
20	85,11%	87,13%	85,11%	83,66%
30	85,05%	87,09%	85,05%	83,58%
40	68,31%	47,23%	68,31%	55,84%
50	68,31%	47,23%	68,31%	55,84%
60	68,31%	47,23%	68,31%	55,84%
70	68,31%	47,23%	68,31%	55,84%
80	68,31%	47,23%	68,31%	55,84%
90	68,31%	47,23%	68,31%	55,84%
100	68,31%	47,23%	68,31%	55,84%

Berdasarkan tabel diatas dapat dilihat bahwa pada saat dilakukan pengujian, tingkat akurasi tergantung pada nilai K yang digunakan. Pada K=10, ditemukan bahwa akurasi mencapai 96,62%, namun terjadi penurunan menjadi 85,11% saat K=20. Pengujian dengan K=30 juga menghasilkan akurasi sebesar 85,05%. Selanjutnya, pada rentang nilai K antara 40 hingga 100, akurasi menurun drastis menjadi 68,31%. Maka dapat ditarik kesimpulan bahwa semakin besar nilai K maka kinerja model menurun. Hal menarik terjadi saat K=10, di mana tidak hanya akurasi yang tinggi, tetapi juga nilai presisi, recall, dan f1-score menunjukkan performa yang lebih baik daripada nilai K yang lain. Oleh karena itu, penelitian ini akan menggunakan nilai K=10 sebagai pembandingan dengan algoritma C4.5.

3.3 Evaluasi

Tahap eveluasi ini dilakukan untuk mengevaluasi kinerja kedua model algoritma klasifikasi yang digunakan pada penelitian ini, dimana dilakukan perbandingan hasil prediksi. Perhitungan Confusion Matrix yang digunakan dalam penelitian ini melibatkan hasil akurasi, presisi, recall, dan f1-score. Pada penelitian ini dilakukan percobaan perbandingan data training dan data testing sebanyak tiga kali dengan rasio perbandingan yaitu 80:20, 70:30, dan 60:40. Selanjutnya, akan dijelaskan hasil Confusion Matrix dari algoritma C4.5 dan Modified K-Nearest Neighbor (MKNN). Implementasi model algoritma C4.5 menggunakan perbandingan data training dan data testing sebanyak 80:20 menghasilkan Confusion Matrix sebagaimana terlihat pada gambar 3 di bawah ini:



Gambar 3. Confusion Matrix Algoritma C4.5

Berdasarkan ilustrasi pada gambar 3 di atas, dapat diambil kesimpulan bahwa Confusion Matrix dari algoritma C4.5 dengan menggunakan 1625 data uji menunjukkan hasil sebagai berikut: 483 data dengan prediksi yang tepat sebagai kelas jamur beracun (poison), sementara hanya 24 data yang salah diprediksi sebagai jamur beracun (poison). Selain itu, 1118 data tepat diprediksi sebagai jamur tidak beracun (edible), dan tidak ada data yang salah diprediksi sebagai jamur tidak beracun (edible). Eksperimen ini dilakukan dalam kerangka

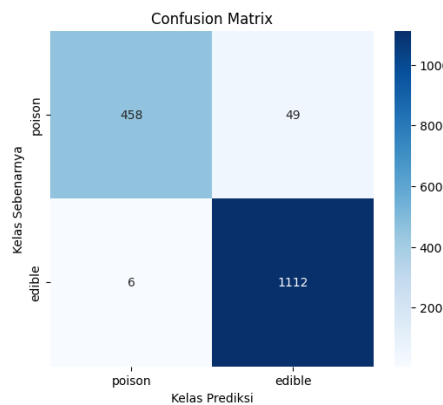
perbandingan antara data pelatihan dan data uji, dengan rasio 80:20, 70:30, dan 60:40. Hasilnya dinilai berdasarkan nilai akurasi, presisi, recall, dan f1-score. dimana dapat dilihat pada tabel 6 di bawah ini:

Tabel 6. Hasil Peforma Algoritma C4.5

Performa Algoritma C4.5					
	Perbandingan	Akurasi	Presisi	Recall	F1-Score
Algoritma C4.5	80:20:00	98,52%	98,55%	98,52%	98,51%
	70:30:00	93,60%	93,58%	93,60%	93,44%
	60:40:00	87,20%	86,98%	87,20%	86,26%

Pada tabel diatas didapatkan hasil dari peforma algoritma C4.5 yang mana pada rasio perbandingan 80:20 memperoleh hasil akurasi sebesar 98,52%, presisi sebesar 98,55%, recall sebesar 98,52%, dan f1-score sebesar 98,51%. Pada rasio perbandingan 70:30 memperoleh hasil akurasi sebesar 93,60%, presisi sebesar 93,58%, recall sebesar 93,60%, dan f1-score sebesar 93,44%. Pada rasio perbandingan 60:40 memperoleh hasil akurasi sebesar 87,20%, presisi sebesar 86,98%, recall sebesar 87,20%, dan f1-score sebesar 86,26%.

Hasil dari menerapkan model algoritma Modified K-Nearest Neighbor (MKNN) menggunakan perbandingan data training dan data testing sebanyak 80:20 menghasilkan Confusion Matrix seperti yang terlihat dalam ilustrasi pada gambar 4 di bawah ini:



Gambar 4. Confusion Matrix Algoritma Modified K-Nearest Neighbor (MKNN)

Dari gambar di atas, dapat disimpulkan bahwa melalui penerapan algoritma Modified K-Nearest Neighbor (MKNN) dengan menggunakan 1625 data uji, terdapat 458 data yang berhasil diprediksi dengan benar sebagai kelas jamur beracun (poison). Selain itu, 49 data diprediksi secara keliru sebagai kelas jamur beracun (poison). Sebanyak 1112 data berhasil diprediksi dengan benar sebagai kelas jamur tidak beracun (edible), namun terdapat 6 data yang salah diprediksi sebagai kelas tidak beracun (edible). Pada percobaan perbandingan data training dan data testing dengan rasio perbandingan yaitu 80:20, 70:30, dan 60:40 dimana hasil nilai akurasi, presisi, recall, dan f1-score dapat dilihat pada tabel 7 di bawah ini.:

Tabel 7. Hasil Peforma Algoritma Modified K-Nearest Neighbor (MKNN)

Performa Algoritma Modified K-Nearest Neighbor (MKNN)					
	Perbandingan	Akurasi	Presisi	Recall	F1-Score
Algoritma MKNN	80:20:00	96,62%	96,69%	96,62%	96,57%
	70:30:00	78,18%	82,00%	78,18%	71,33%
	60:40:00	80,86%	80,45%	80,86%	77,25%

Pada tabel diatas didapatkan hasil dari peforma algoritma Modified K-Nearest Neighbor (MKNN) dimana pada rasio perbandingan 80:20 berhasil mendapatkan tingkat akurasi sekitar 96,62%, dengan presisi mencapai 96,69%, recall sekitar 96,62%, dan f1-score sekitar 96,57%. Pada rasio perbandingan 70:30 memperoleh hasil akurasi sebesar 78,18%, presisi sebesar 82,00%, recall sebesar 78,18%, dan f1-score sebesar 71,33%. Pada rasio perbandingan 60:40 memperoleh nilai akurasi sebesar 80,86%, presisi sebesar 80,45%, recall sebesar 80,86%, dan f1-score sebesar 77,25%.

Berdasarkan perhitungan yang telah dilakukan, diperoleh perbandingan hasil kinerja antara algoritma C4.5 dan Algoritma Modified K-Nearest Neighbor (MKNN) dalam hal hasil akurasi, presisi, recall, dan f1-score. Rincian hasil perbandingan ini dapat ditemukan dalam tabel 8 berikut:

Tabel 8. Perbandingan Peforma Algoritma C4.5 dan Algoritma Modified K-Nearest Neighbor (MKNN)

Algoritma	Akurasi	Presisi	Recall	F1-Score
C4.5	98,52%	98,55%	98,52%	98,51%

Algoritma	Akurasi	Presisi	Recall	F1-Score
MKNN	96,62%	96,69%	96,62%	96,57%

Berdasarkan perbandingan performa, terlihat bahwa algoritma C4.5 mencapai nilai akurasi yang lebih tinggi, yakni sekitar 98,52%, sedangkan Algoritma Modified K-Nearest Neighbor (MKNN) memperoleh nilai akurasi sekitar 96,62%. Dalam hal presisi, algoritma C4.5 juga menunjukkan performa yang lebih baik dengan nilai sekitar 98,55%, sementara Algoritma Modified K-Nearest Neighbor (MKNN) mencapai presisi sekitar 96,69%. Pada recall, algoritma C4.5 memperoleh nilai sekitar 98,52%, sedangkan Algoritma Modified K-Nearest Neighbor (MKNN) mendapatkan recall sekitar 96,62%. Sementara pada f1-score, algoritma C4.5 meraih nilai sekitar 98,51%, sedangkan Algoritma Modified K-Nearest Neighbor (MKNN) memperoleh f1-score sekitar 96,57%.

4. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa perbandingan kinerja antara algoritma C4.5 dan Modified K-Nearest Neighbor (MKNN) dalam melakukan klasifikasi jamur menunjukkan bahwa algoritma C4.5 memiliki performa lebih unggul. Penilaian kinerja dilakukan dengan membandingkan akurasi, presisi, recall, dan f1-score dari kedua algoritma tersebut. Meskipun kedua algoritma mampu menghasilkan model klasifikasi dengan tingkat akurasi di atas 90%, algoritma C4.5 memperoleh hasil yang lebih baik dibandingkan dengan Modified K-Nearest Neighbor (MKNN). Dalam penelitian ini, algoritma C4.5 berhasil memperoleh tingkat akurasi sebesar 98,52%, presisi sebesar 98,55%, recall sebesar 98,52%, dan f1-score sebesar 98,51%. Sebaliknya, algoritma Modified K-Nearest Neighbor (MKNN) dengan menggunakan nilai K=10 meraih tingkat akurasi sebesar 96,62%, presisi sebesar 96,69%, recall sebesar 96,62%, dan nilai f1-score sebesar 96,57%. Dari total 1625 data jamur beracun (poison) dan jamur tidak beracun (edible), algoritma C4.5 berhasil mengklasifikasikan 1602 data dengan benar, sedangkan algoritma Modified K-Nearest Neighbor (MKNN) berhasil mengklasifikasikan 1570 data dengan benar. Algoritma C4.5 juga berhasil mengurangi jumlah atribut dari 23 menjadi 5. Hal ini menyebabkan model klasifikasi algoritma C4.5 menjadi lebih sederhana dalam hal kompleksitas perhitungan. Sebaliknya, Modified K-Nearest Neighbor (MKNN) mencapai kinerja yang bagus dengan memakai nilai K=10. Dari hasil penelitian ini, dapat disimpulkan bahwa algoritma C4.5 lebih efektif untuk melakukan klasifikasi data jamur.

REFERENCES

- [1] Y. Suryani, O. Taupiqurrahman, and Y. Kulsum, Mikologi, 1st ed. Padang: PT. Freeline Cipta Granesia, 2020.
- [2] K. Tangah et al., "Identifikasi Keanekaragaman Jamur Makroskopis di Daerah Lubuk," no. 2006, pp. 832–841, 2023.
- [3] K. Nurfitri, A. D. Pradana, and I. Widaningrum, "PENERAPAN ALGORITMA PRINCIPAL COMPONENT ANALYSIS (PCA) DAN K-NEAREST NEIGHBORS (KNN) PADA KLASIFIKASI," vol. x, pp. 1–19, 2021.
- [4] R. Hayami, Soni, and I. Gunawan, "Klasifikasi Jamur Menggunakan Algoritma Naïve Bayes," J. CoSciTech (Computer Sci. Inf. Technol., vol. 3, no. 1, pp. 28–33, 2022, doi: 10.37859/coscitech.v3i1.3685.
- [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," vol. 6, no. 8, pp. 3948–3955, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [6] I. P. Putra, "Kasus keracunan Inocybe sp. di Indonesia," Pros. Semin. Nas. Biol., no. September, pp. 148–153, 2020.
- [7] I. P. Putra, "Kasus-Kasus Keracunan Jamur Liar Di Indonesia," J. Ekol. Kesehat., vol. 20, no. 3, pp. 215–230, 2022, doi: 10.22435/jek.v20i3.4943.
- [8] Yuli Mardi, "Data Mining : Klasifikasi Menggunakan Algoritma C4.5," J. Edik Inform., vol. 2, no. 2, pp. 213–219, 2019.
- [9] H. Rofiq, K. C. Pelangi, and Y. Lasena, "Penerapan Data Mining Untuk Menentukan Potensi Hujan Harian Dengan Menggunakan Algoritma K Nearest Neighbor (KNN)," J. Manaj. Inform. dan Sist. Inf., vol. 3, no. 1, pp. 8–15, 2020, [Online]. Available: <http://mahasiswa.dinus.ac.id/docs/skripsi/jurnal/19417.pdf>
- [10] D. Ardiansyah, "Algoritma C4.5 Untuk Klasifikasi Calon Peserta Lomba Cerdas Cermat Siswa Smp Dengan Menggunakan Aplikasi Rapid Miner," J. Inkofar, vol. 1, no. 2, pp. 5–12, 2019, doi: 10.46846/jurnalinkofar.v1i2.29.
- [11] P. B. N. Setio, D. R. S. Saputro, and Bowo Winarno, "Klasifikasi Dengan Pohon Keputusan Berbasis Algoritme C4.5," Prism. Pros. Semin. Nas. Mat., vol. 3, pp. 64–71, 2020.
- [12] D. Marlina and M. Bakri, "Penerapan Data Mining Untuk Memprediksi Transaksi Nasabah Dengan Algoritma C4.5," J. Teknol. dan Sist. Inf., vol. 2, no. 1, pp. 23–28, 2021.
- [13] I. Agustin, Y. Novia Nasution, and D. Wasono, "Klasifikasi Batubara Berdasarkan Jenis Kalori Dengan Menggunakan Algoritma Modified K-Nearest Neighbor (Studi Kasus : PT.Pancaran Surya Abadi) Classification Of Coal Based On Calorie Type Using Modified K-Nearest Neighbor Algorithm (Case Study: PT.Pancara)," J. EKSPONENSIAL, vol. 10, no. 1, pp. 73–80, 2019.
- [14] E. F. Ginting, K. Ibnutama, and S. Kusnasari, "Aplikasi Cerdas Membantu Orang Tua Meningkatkan Kecerdasan Anak Dalam Pembelajaran Online Pandemi Covid -19 Dengan Algoritma C4.5.," J. Sist. Komput. dan Inform., vol. 3, no. 2, p. 147, 2021, doi: 10.30865/json.v3i2.3620.
- [15] K. Aidi Saputra, J. Tata Hardinata, M. Ridwan Lubis, S. Retno Andani, and I. Syahputra Saragih, "Klasifikasi Algoritma C4.5 Dalam Penerapan Tingkat Kepuasan Siswa Terhadap Media Pembelajaran Online," Media Online), vol. 1, no. 3, pp. 113–118, 2020, [Online]. Available: <https://djournals.com/klik>
- [16] A. Nirwana, A. Siregar, and R. Rahmat, "Klasifikasi Permasalahan Kredit Macet Pada Bank Menggunakan Algoritma Decision Tree C4. 5," Sci. Student J. Information, Technol. Sci., vol. 3, no. 1, pp. 43–50, 2022, [Online]. Available:

- <https://journal.ubpkarawang.ac.id/mahasiswa/index.php/ssj/article/view/420>
- [17] N. Attamami, A. Triayudi, and R. T. Aldisa, “Analisis Performa Algoritma Klasifikasi Naive Bayes dan C4.5 untuk Prediksi Penerima Bantuan Jaminan Kesehatan,” *J. JTIC (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 7, no. 2, pp. 262–269, 2023, doi: 10.35870/jtik.v7i2.756.
 - [18] F. Insani and S. Nissa, “Penerapan Genetic Modified k-Nearest Neighbor Pada Prediksi PM10 di Pekanbaru,” *Komputika J. Sist. Komput.*, vol. 10, no. 2, pp. 153–161, 2021, doi: 10.34010/komputika.v10i2.4404.
 - [19] M. R. W. A. Saputra, “Identifikasi Kerusakan Mesin Pada Mobil Matic Menggunakan Metode Modified K-Nearest Neighbor (MKNN),” *J. Smart Teknol.*, vol. 1, no. 1, pp. 2774–1702, 2021, [Online]. Available: <http://jurnal.unmuhjember.ac.id/index.php/JST>
 - [20] R. Chaidir, H. Oktavianto, and R. Umilasari, “Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) Untuk Klasifikasi Penerimaan Dana Batuan COVID-19,” vol. 5, no. 1, pp. 38–44, 2023.
 - [21] H. D. A. Hamid and Nurul Hidayat, “Diagnosis Penyakit Tanaman Cabai Menggunakan Metode Modified K-Nearest Neighbor (MKNN),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 3, pp. 2881–2886, 2019.
 - [22] B. Erizal, M. Bettiza, and A. Uperiati, “Perbandingan Algoritma Modified K-Nearest Neighbor Dan Naive Bayes Untuk Klasifikasi Data Rekam Medis Berdasarkan Icd-10 (Studi Kasus: Rsud Raja Ahmad Tabib Kota Tanjungpinang),” vol. 10, pp. 407–415, 2021.
 - [23] I. S. Muallif et al., “Penerapan Data Mining untuk Prediksi Pergerakan Harga Saham Menggunakan Algoritma K-Nearest Neighbor,” vol. 1, no. 1, pp. 1–5, 2023.
 - [24] S. Ulfah, “Komparasi Algoritma C4.5 Dengan Algoritma Naive Bayes Classifier Untuk Klasifikasi Status Gizi Balita,” 2022, doi: 10.34001/jdpt.v12i2.
 - [25] Suryani, D. Rahmadani, A. A. Muzafar, A. Hamid, R. Annisa, and Mustakim, “Comparative Analysis of C4.5 and CART Algorithms for Classification of Stroke,” *SENTIMAS Semin. Nas. Penelit. dan Pengabd. Masy.*, vol. 1, no. 1, pp. 197–206, 2022.
 - [26] F. M. Hana, “Klasifikasi Penderita Penyakit Diabetes Menggunakan Algoritma Decision Tree C4.5,” *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 4, no. 1, pp. 32–39, 2020, doi: 10.47970/siskom-kb.v4i1.173.