Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 1, November 2023, Page 10-19
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i1.4511

Perbandingan Jarak Metrik pada Klasifikasi Jamur Beracun Menggunakan Algoritma K-Nearest Neighbor (K-NN)

Andre Suarisman, Alwis Nazir*, Fadhilah Syafria, Liza Afriyanti

Fakultas Sains dan Teknologi, Program Studi Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim, Pekanbaru, Indonesia
Email: [1]11950111673@students.uin-suska.ac.id, [2,*]alwis.nazir@uin-suska.ac.id, [3]fadhilah.syafria@uin-suska.ac.id, [4]liza.afriyanti@uin-suska.ac.id
Email Penulis Korespondensi: alwis.nazir@uin-suska.ac.id
Submitted: 31/10/2023; Accepted: 27/11/2023; Published: 28/11/2023

Abstrak−Jamur adalah organisme dari kingdom fungi yang memiliki struktur tubuh berdaging dan bisa dikonsumsi, tetapi ada beberapa spesies jamur yang tidak aman untuk dimakan dan memiliki ciri-ciri khusus, Maka dalam membedakan antara jamur yang dapat dimakan dan beracun bisa jadi rumit karena tampilan yang hampir identik dari berbagai spesies jamur. Kesalahan dalam mengidentifikasi jamur yang dapat dimakan dapat berdampak pada kesehatan konsumen yang mengkonsumsi jamur tersebut. Evaluasi performa berbagai metode pada dataset adalah langkah kunci dalam menentukan metode klasifikasi yang paling sesuai. Penelitian ini mengangkat tentang bagaimana mengukur performa metode klasifikasi pada dataset jamur beracun dengan menggunakan algoritma K-Nearest Neighbor dengan beberapa metrik seperti *euclidean, manhattan* dan *minkowski*, dimana merupakan sebuah metode untuk mengklasifikasikan data baru berdasarkan kedekatan dengan data pelatihan yang ada. Hasil yang didapat dalam penelitian ini dengan beberapa jarak metrik dapat disimpulkan bahwa nilai akurasi metrik *manhattan* lebih baik dari metrik *euclidean* dan *minkowski*. Karena metrik *manhattan* mendapatkan hasil akurasi tertinggi yaitu 99% dengan K = 100 dan terendah 82% dengan K = 3000 , sedangkan metrik *euclidean* mendapatkan hasil akurasi dengan nilai 98% dengan K = 100 dan 72% dengan K = 3000, dan pada metrik *minkowski* mendapatkan hasil akurasi dengan nilai 96% pada K = 100 dan 64% pada K = 3000.

Kata Kunci: Klasifikasi; Metrik; K-Nearest Neighbor; Python; Jamur

Abstract− Mushrooms are organisms from the kingdom fungi that have a fleshy body structure and can be consumed, but there are some species of mushrooms that are not safe to eat and have specific characteristics, so distinguishing between edible and poisonous mushrooms can be tricky due to the almost identical appearance of various mushroom species. Errors in identifying edible mushrooms can impact the health of consumers who consume the mushrooms. Evaluating the performance of various methods on a dataset is a key step in determining the most suitable classification method. This research is about how to measure the performance of classification methods on toxic mushroom datasets using the K-Nearest Neighbor algorithm with several metrics such as euclidean, manhattan and minkowski, which is a method for classifying new data based on proximity to existing training data. The results obtained in this study with several distance metrics can be concluded that the accuracy value of the manhattan metric is better than the euclidean and minkowski metrics. Because the manhattan metric gets the highest accuracy result of 99% with K = 100 and the lowest 82% with K = 3000, while the euclidean metric gets accuracy results with a value of 98% with K = 100 and 72% with K = 3000, and the minkowski metric gets accuracy results with a value of 96% at K = 100 and 64% at K = 3000.

Keywords: Classification; Metric; K-Nearest Neighbor; Python; Mushroom

1. PENDAHULUAN

Jamur adalah organisme dari kingdom fungi yang memiliki struktur tubuh berdaging dan bisa dikonsumsi, tetapi perlu diingat bahwa ada beberapa spesies jamur yang tidak aman untuk dimakan dan memiliki ciri-ciri khusus[1]. Jamur memiliki banyak jenis dengan bentuk, ukuran, dan warna yang beragam. Walaupun demikian, terdapat kesamaan dalam beberapa hal seperti ukuran yang bervariasi dari yang kecil hingga besar, bentuk yang menyerupai payung terbuka ke atas, dan warna yang dapat beragam dari gelap hingga terang[2]. Maka dalam membedakan antara jamur yang dapat dimakan dan beracun bisa jadi rumit karena tampilan yang hampir identik dari berbagai spesies jamur. Kesalahan dalam mengidentifikasi jamur yang dapat dimakan dapat berdampak pada kesehatan konsumen yang mengkonsumsi jamur tersebut[3].

Pada tahun 2010-2020, Indonesia telah mencatat 76 kasus keracunan akibat mengonsumsi jamur liar. Sebanyak 550 orang menjadi korban dan 9 diantaranya meninggal dunia akibat keracunan tersebut. Puncak kasus keracunan terjadi pada tahun 2014, dan kasus-kasus keacunan tercatat berasal dari beberapa daerah besar yaitu Sumatra, Jawa, Kalimantan, Sulawesi, dan Nusa Tenggara (NTB dan NTT)[4][5]. Selama periode tahun 2010 sampai dengan tahun 2020 di Indonesia terdapat 7 kasus keracunan yang disebabkan oleh salah mengidentifikasi jenis jamur Inocybe sebagai Termitomyces. Dari 31 korban, satu orang meninggal dengan rentang usia 2-80 tahun. Penyebaran kasus keracunan hanya terjadi di 5 provinsi, dan sebagian besar kasus terjadi di Pulau Jawa[5]. Berdasarkan permasalahan yang ada maka dirasa perlu untuk melakukan penelitian yang bisa mengklasifikasikan jamur beracun dan tidak beracun dengan memanfaatkan teknologi data mining.

Data mining adalah suatu proses penggalian data yang melibatkan kecerdasan buatan, teknik matematika, statistik, dan macine learning pada database yang besar untuk menentukan informasi yang bermanfaat serta pengetahuan terkait. Data mining dapat didefinisikan sebagai proses pencarian korelasi atau pola yang

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 1, November 2023, Page 10-19
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i1.4511

bermanfaat dari ratusan atau ribuan dalam database yang sangat besar[6]. Dalam penelitian ini akan digunakan salah satu teknik analisis data mining yaitu klasifikasi.

Klasifikasi adalah sebuah metode pengelompokan atau penempatan data ke dalam kelas atau kategori yang telah ditentukan sebelumnya[7]. Ada tiga tahapan dalam klasifikasi, yaitu pembuatan model, aplikasi model, dan evaluasi. Pada tahap pengembangan model, model dibuat menggunakan data pelatihan yang sudah memiliki atribut dan kelas. Selanjutnya, data tersebut digunakan untuk menerapkan klasifikasi pada data atau objek baru. Setelah itu, dilakukan evaluasi terhadap data untuk mengetahui akurasi pengembangan dan penerapan model terhadap data baru[8]. Dalam penelitian ini, algoritma yang akan diterapkan adalah algoritma *K-Nearest Neighbor*.

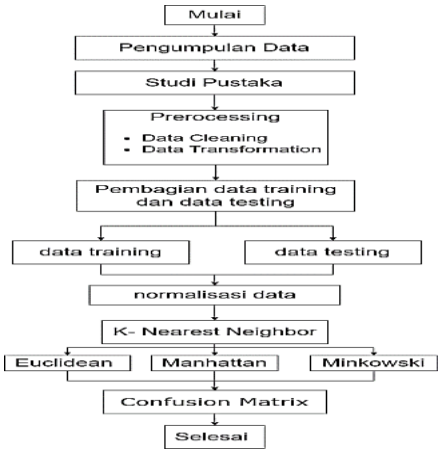
Algoritma K-Nearest Neighbor adalah algoritma klasifikasi yang digunakan untuk mengklasifikasikan data berdasarkan klasifikasi yang sudah diketahui sebelumnya[9]. Algoritma k-nearest neighbor termasuk dalam kategori instance-base learning dan termasuk dalam teknik lazy learning[6]. KNN bekerja dengan mencari K objek terdekat dalam data latihan yang memiliki kesamaan dengan objek pada data uji. pendekatan untuk mencari kasus dengan memperhitungkan kesamaan bobot dari beberapa atribut yang ada dengan menghitung jarak antara data latihan (x) dan data uji (y), didefinisikan menggunakan rumus jarak metrik *Euclidean*, *Manhattan* dan *Minkowski*[9][10]. Ada salah satu penelitian yang telah dilakukan menggunakan algoritma K-NN dalam menyelesaikan masalah seperti salah satu penelitian yang dilakukan Esty Purwaningsih dan Ela Nurelasari dengan judul "Penerapan K-*Nearest Neighbor* untuk Klasifikasi Tingkat Kelulusan Pada Siswa". Pada penelitian ini diterangkan bahwa algoritma K-NN dapat memprediksi tingkat keberhasilan siswa dalam kelulusan berdasarkan dari kinerja siswa. Hasil dari penelitian tersebut bahwa prediksi tingkat kelulusan siswa didapatkan dengan rata- rata akurasi sebesar 96,49%[11]. Ada beberapa penelitian yang telah melakukan penelitian terkait kasus jamur seperti salah satu penelitian yang dilakukan Yohannes1 dkk tahun 2022 dengan judul "Klasifikasi Jenis Jamur Menggunakan SVM dengan Fitur HSV dan HOG" dan mendapatkan hasil bahwa SVM mampu mengklasifikasikan jenis jamur pada citra dengan fitur HSV dan HOG dengan akurasi 82,69% dengan jenis jamur yang mendapatkan hasil terbaik, yaitu jamur boletus dengan akurasi 89,69%[2]. Penelitian yang dilakukan Aisha Alfani W dkk dengan judul "Prediksi penjualan Produk Unilever Menggunakan Metode K-Nearest Neighbor" tahun 2021dan mendapatkan hasil penjualan tertinggi dengan akurasi 86,66% dan akurasi terendah 40%[12]. Penelitian dari Aida Indriani dengan judul "Analisa Perbandingan Metode Naïve Bayes Classifier dan K- Nearest Neighbor terhadap Klasifikasi Data" tahun 2020. Pada penelitian ini mendapatkan hasil berdasarkan nilai efektifitas klasifikasi bahwa metode K-NN lebih baik dengan nilai 80% sedangkan NBC mendapatkan nilai 73%[13].

Pada penelitian ini data yang dipakai adalah data jamuryang diperoleh dari situs Kaggle.com. Dataset ini mencakup 23 kategori dan terdapat dua kelas utama, yakni jamur yang aman dikonsumsi (edible) dan jamur beracun (poison). Terdapat 4208 sampel data untuk kelas jamur yang dapat dimakan dan 3916 sampel data untuk kelas jamur beracun. Dengan begitu, total data yang digunakan mencapai 8124 sampel.

Dalam penelitian ini akan memanfaatkan algoritma K-Nearest Neighbors (KNN) untuk mengukur akurasi, presisi, recall, dan F1-score menggunakan *pyhton*. Ini dilakukan dengan menguji berbagai nilai K dan menggunakan berbagai rumus jarak metrik dalam metodenya. Penelitian ini juga melibatkan beberapa tahap, seperti pra-pemrosesan data, transformasi data, pembagian data menjadi data pelatihan dan data uji, penerapan algoritma KNN, serta analisis kinerja metode yang diuji.

2. METODOLOGI PENELITIAN

Panduan langkah-langkah yang digunakan dalam penelitian ini dapat diacu pada ilustrasi yang terdapat dalam Gambar 1.



Gambar 1. Kerangka Metodologi Penelitian

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 1, November 2023, Page 10-19
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i1.4511

2.1 Pengumpulan Data

Dalam penelitian ini, digunakan sebuah dataset yang bersumber dari situs Kaggle.com, yang terdiri dari 23 kategori jamur diantaranya adalah *cap-shape*, *cap-surface*, *cap-color*, *bruise%3f*, *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*, *spore-print-color*, *population*, *habitat*, dan *class*. Dataset ini tersegmentasi menjadi dua kelas utama, yaitu kelas jamur yang aman untuk dikonsumsi (edible) dan kelas jamur yang bersifat beracun (poison). Dataset ini terdiri dari 4208 sampel data untuk kelas jamur yang dapat dimakan dan 3916 sampel data untuk kelas jamur beracun, sehingga menjadikan total jumlah data yang digunakan sebanyak 8124 sampel dimana akan dibagi menjadi data latih dan data uji yaitu 20% data uji dan 80% data latih.

	cap-shape	cap-surface	cap-color	bruises%3F	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	...	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat	class
0	convex	smooth	brown	True	pungent	free	close	narrow	black	enlarging	...	white	white	partial	white	one	pendant	black	scattered	urban	poison
1	convex	smooth	yellow	True	almond	free	close	broad	black	enlarging	...	white	white	partial	white	one	pendant	brown	numerous	gresses	edible
2	bell	smooth	white	True	anise	free	close	broad	brown	enlarging	...	white	white	partial	white	one	pendant	brown	numerous	meadow	edible
3	convex	scaly	white	True	pungent	free	close	narrow	brown	enlarging	...	white	white	partial	white	one	pendant	black	scattered	urban	poison
4	convex	smooth	gray	False	none	free	crowded	broad	black	tapering	...	white	white	partial	white	one	evanescent	brown	abudant	gresses	edible
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8119	knibbed	smooth	brown	False	none	attached	close	broad	yellow	enlarging	...	orange	orange	partial	orange	one	pendant	buff	cluseted	leaves	edible
8120	convex	smooth	brown	False	none	attached	close	broad	yellow	enlarging	...	orange	orange	partial	brown	one	pendant	buff	several	leaves	edible
8121	flat	smooth	brown	False	none	attached	close	broad	brown	enlarging	...	orange	orange	partial	orange	one	pendant	buff	cluseted	leaves	edible
8122	knibbed	scaly	brown	False	fishy	free	close	narrow	buff	tapering	...	white	white	partial	white	one	evanescent	white	several	leaves	poison
8123	convex	smooth	brown	False	none	attached	close	broad	yellow	enlarging	...	orange	orange	partial	orange	one	pendant	orange	cluseted	leaves	edible

8124 rows × 23 columns

Gambar 2. Dataset Penelitian

2.2 *Pre-Processing Data*

Pre-processing data merupakan langkah awal yang berfokus pada transformasi data ke dalam format yang lebih sederhana dan efisien, yang dapat meningkatkan akurasi, mengurangi waktu komputasi, dan mengompresi ukuran data tanpa mengorbankan informasi yang ada[14]. Dalam penelitian ini, tahap pre-processing data melibatkan pembersihan data, yang bertujuan untuk mengenali serta mengatasi masalah seperti nilai-nilai yang hilang, data-data ekstrem, dan ketidaksesuaian data untuk memastikan kualitas data yang lebih baik. Selain itu, dilakukan transformasi data seperti normalisasi atau standarisasi agar data sesuai dengan kebutuhan analisis yang spesifik.

2.3 Normalisasi

Normalisasi data adalah suatu teknik penting dalam praproses data yang melibatkan penyesuaian ulang nilai-nilai dalam dataset. Tujuannya adalah untuk menyamakan skala atau rentang nilai di antara atribut-atribut dataset. Hal ini diperlukan karena perbedaan besar dalam rentang nilai antar atribut dapat mengganggu kinerja optimal atribut dalam analisis data. Oleh karena itu, normalisasi data menjadi suatu langkah krusial untuk memastikan data yang diolah lebih konsisten dan akurat[15]. Dalam penelitian ini dilakukan normalisasi menggunakan metode normalisasi Z-Score untuk mengolah data.

$$Z = \frac{x-\mu}{\sigma} \quad (1)$$

Keterangan:
Z = nilai Z-Score yang dihasilkan.
X = nilai dari data yang akan dinormalisasi.
μ = rata-rata (mean) dari seluruh data.
σ = simpangan baku (standard deviation) dari seluruh data.

2.4 *Klasifikasi K Nearest Neighbor*

K-Nearest Neighbors (KNN) adalah salah satu teknik dalam *supervised learning* yang memanfaatkan data latih untuk mengklasifikasikan data baru berdasarkan kedekatan dengan data pelatihan yang ada[13]. Algoritma KNN mengelompokkan data baru berdasarkan seberapa dekatnya dengan data pelatihan, menggunakan berbagai metrik jarak metrik seperti jarak metrik *Euclidean*, jarak metrik *Manhattan*, dan metrik jarak *Minkowski*. Ini berarti bahwa KNN memanfaatkan informasi mengenai tetangga terdekat untuk mengkategorikan data baru[15].

Algoritma KNN sangat sederhana karena bekerja berdasarkan jarak terpendek antara instance query dan sampel data training untuk menentukan tetangganya. Jarak dekat atau jauh dari tetangga biasanya dihitung menggunakan jarak *Euclidean*, yang sering digunakan untuk mengukur kedekatan antara dua objek[16]. Dalam

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 1, November 2023, Page 10-19
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i1.4511

penelitian ini dilakukan menggunakan jarak *Euclidean*, jarak *Manhattan*, dan jarak *Minkowski*. Rumus jarak *Euclidean* dinyatakan sebagai berikut.

$$euclidean = \sqrt{(\sum_{i=1}^{n}(x_i - y_i)^2)} \quad (3)$$

Keterangan :
xi = data latihan
yi = data uji
i = variable data
n = dimensi data
Rumus jarak *Manhattan* dinyatakan sebagai berikut.

$$manhattan = \sum_{i=1}^{n}|xi - yi| \quad (4)$$

Keterangan:
xi = data training
yi = data test
i = variable data
n = dimensi data
Rumus jarak *Minkowski* dinyatakan sebagai berikut.

$$manhattan = (\sum_{i=1}^{n}|xi - yi|^p)^{\frac{1}{p}} \quad (5)$$

Keterangan :
xi = data training
yi = data test
i = variable data
n = dimensi data
p = parameter order
Berikut proses-proses yang terlibat dalam penerapan metode klasifikasi KNN :

a. Langkah awal adalah memilih nilai k. Proses pemilihan k ini sangat fleksibel dan bisa disesuaikan dengan kebutuhan analisis. Dalam penelitian ini, kami menerapkan berbagai nilai k yang berbeda untuk melihat bagaimana pengaruhnya terhadap hasil klasifikasi, sehingga memungkinkan perbandingan antar-nilai k yang berbeda.
b. Langkah selanjutnya adalah mengukur jarak antara data uji dan data latih. Dalam penelitian ini, metode yang digunakan untuk menghitung jarak adalah *Euclidean*, *Manhattan*, dan *Minkowski*.
c. Langkah selanjutnya adalah menentukan nilai tetangga terdekat dari perhitungan jarak. Lalu kita dapat melakukan klasifikasi data uji dengan memperhatikan kelas mayoritas dari nilai-nilai tetangga terdekat. Sebagai contoh, jika mayoritas tetangga terdekat memiliki kelas 'jamur beracun', maka data uji yang belum diketahui akan diklasifikasikan sebagai 'jamur beracun'.

Setelah melakukan klasifikasi menggunakan K-Nearest Neighbor, langkah selanjutnya adalah mengukur performansi hasil klasifikasi dengan menghitung nilai akurasi, presisi, recal dan f1-score menggunakan *confusion matrix*[17][18].

2.5 Confusion Matrix

Confusion matrix adalah sebuah metode untuk mengevaluasi performa sebuah model klasifikasi yang mampu mengklasifikasikan data dengan benar dan mengidentifikasikan kesalahan yang dilakukan[13][19]. Contoh confusion matrix dapat dilihat pada tabel 1 berikut.

Table 1. Evaluasi *Confusion Matrix*

		Kelas prediksi	
		Positif	Negatif
Kelas sebenarnya	Positif	TP	FN
	Negatif	FP	TN

Keterangan :
TP (True Positive) = jumlah data yang termasuk dalam kategori positif pada kelas sebernya dan jugadiprediksi sebagai positif.
FN (False Negative) = jumlah data yang termasuk dalam kategori positif pada kelas sebenarnya, tetapi diprediksi sebagai negatif.
FP (False Positive) = jumlah data yang termasuk dalam kategori negatif pada kelas sebenarnya, namun diprediksi sebagai positif.

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 1, November 2023, Page 10-19
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i1.4511

TN (True Negative) = jumlah data yang termasuk dalam kategori negatif pada kelas sebenarnya dan juga diprediksi sebagai negatif.

Beberapa pengujian detail akan dilakukan untuk mengevaluasi akurasi, presisi, dan recall. Akurasi adalah pendekatan pengujian yang mengukur sejauh mana prediksi cocok dengan nilai sebenarnya secara menyeluruh. Rumus untuk menghitung akurasi terlihat dalam persamaan berikut.

$$\text{Akurasi} = \frac{TP+TN}{TP+FN+FP+TN} \quad (5)$$

Presisi adalah mengukur sejauh mana hasil positif yang diprediksi benar. Rumus untuk menghitung presisi terlihat dalam persamaan berikut.

$$Presisi = \frac{TP}{TP+FP} \quad (6)$$

Recal adalah mengukur sejauh mana data sebenarnya positif diidentifikasi dengan benar. Rumus untuk menghitung recal dapat dilihat dalam persamaan berikut.

$$Recal = \frac{TP}{TP+FN} \quad (7)$$

F1 Score adalah ukuran gabungan dari presisi dan recal. Rumus untuk menghitung f1 score dapat dilihat dalam persamaan berikut.

$$F1 - Scorre = \frac{2 \times Presisi \times Recal}{Presisi+Recal} \quad (8)$$

3. HASIL DAN PEMBAHASAN

Pada bagian ini mencakup hasil-hasil pengujian sistem terhadap berbagai macam percobaan nilai k dengan tujuan untuk mendapatkan nilai akurasi terbaik.

3.1 Implementasi pada K-NN menggunakan *Python*

Adapun langkah-langkah utaman dalam implementasi kode program python melibatkanurutan berikut secara garis besar:

a. *Import library* yang diperlukan dalam *python*
Pada langkah-langkah pemrograman menggunakan Python, langkah awal yang krusial adalah mengimpor berbagai library yang diperlukan dalam pembuatan program. Ini merupakan fondasi dari pengembangan perangkat lunak yang efisien dan efektif. Beberapa dari library-library kunci yang sering digunakan dalam pengembangan perangkat lunak dengan Python termasuk:
1. *NumPy*, *Library* ini memiliki peran penting dalam melakukan operasi matematis dan ilmiah.
2. *Pandas*, Untuk keperluan membaca, mengolah, dan memanipulasi data, khususnya dalam berbagai format seperti *file .txt, .csv, .xlsx,* dan lainnya.
3. *Matplotlib dan Seaborn,Library* ini digunakan untuk menciptakan *visualisasi* dan grafik.
4. *Scikit-learn (Sklearn),Sklearn* memiliki berbagai algoritma dan metode yang memfasilitasi pengembangan model *machine learning.* Dalam pengembangan program *machine learning* ini, perlu memanggil berbagai algoritma seperti *KNeighborsClassifier,* melakukan pemisahan data dengan *train-test split,* standarisasi data dengan *StandardScaler,* dan menganalisis performa model dengan *confusion matrix,* serta berbagai fungsi penting lainnya.

Untuk melihat script Python yang digunakan untuk mengimpor library, dapat merujuk ke gambar 3.

```
[39] import numpy as np
     import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.metrics import confusion_matrix, accuracy_score, precision_recall_fscore_support, classification_report
     import matplotlib.pyplot as plt
     import seaborn as sns
```

Gambar 3. Script *Python Import Library*

b. *Import* dataset
Langkah selanjutnya dalam penelitian ini adalah mengakses dataset yang tersimpan di Google Drive dan memverifikasi apakah program mampu membacanya dengan benar. Untuk script python yang digunakan untuk mengimpor dataset, dapat merujuk gambar 4.

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 1, November 2023, Page 10-19
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i1.4511

```
from google.colab import drive
drive.mount('/content/drive')

#masukkan dataset ke dalam variable
data_excel = "/content/drive/MyDrive/Colab Notebooks/data_jamur_test1.xlsx"

#membaca dataset
datajamur = pd.read_excel(data_excel)
datajamur
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call driv
```

Gambar 4. Script Python *Import* Dataset

c. Transformasi data menggunakan *label encoding*
Langkah selanjutnya dalam penelitian ini, dataset akan mengalami transformasi menggunakan metode label encoding. Proses ini bertujuan untuk mengubah data non-nominal menjadi data nominal, mempermudah operasi data oleh program. Untuk script python yang digunakan untuk mentranformasi dataset dan hasilnya, dapat merujuk gambar 5 dan 6.

```
# Menghubungkan Google Colab dengan Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Impor library yang dibutuhkan
import pandas as pd
from sklearn.preprocessing import LabelEncoder

#masukkan dataset ke dalam variable
excel_path = "/content/drive/MyDrive/Colab Notebooks/data_jamur_test1.xlsx"

# Baca data jamur Anda (pastikan ganti 'nama_file.xlsx' dengan nama file yang sesuai)
data = pd.read_excel(excel_path)

# Daftar kolom yang ingin Anda lakukan label encoding
kolom_kategorikal = ['cap-shape','cap-surface','cap-color','bruises%3F','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','habitat']
# Inisialisasi objek LabelEncoder
label_encoder = LabelEncoder()

# Lakukan label encoding untuk setiap kolom kategorikal
for kolom in kolom_kategorikal:
    data[kolom] = label_encoder.fit_transform(data[kolom])

# Simpan data yang telah di-label encoding ke dalam file Excel
data.to_excel('data_label_encoded.xlsx', index=False)
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Gambar 5. Script Python Transformasi Dataset

```
setdata = pd.read_excel('data_label_encoded.xlsx')
setdata.head()
```

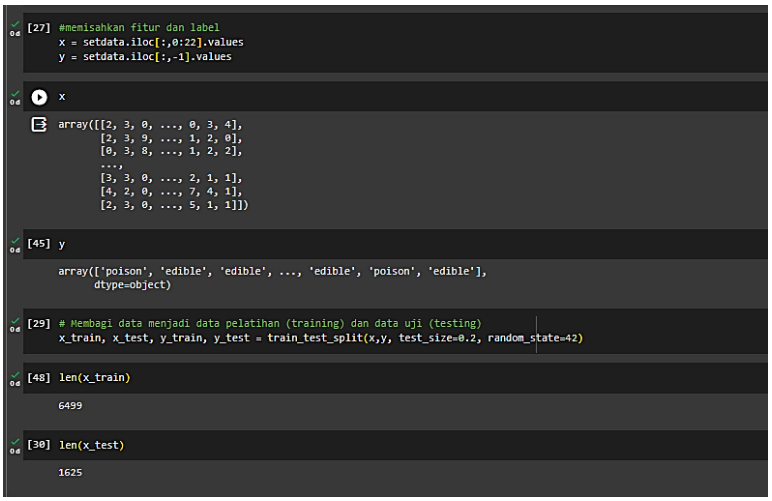
	cap-shape	cap-surface	cap-color	bruises%3F	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	...	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat	class
0	2	3	0	1	7	1	0	1	0	0	...	7	7	0	2	1	4	0	3	4	poison
1	2	3	9	1	0	1	0	0	0	0	...	7	7	0	2	1	4	1	2	0	edible
2	0	3	8	1	1	1	0	0	1	0	...	7	7	0	2	1	4	1	2	2	edible
3	2	2	8	1	7	1	0	1	1	0	...	7	7	0	2	1	4	0	3	4	poison
4	2	3	3	0	6	1	1	0	0	1	...	7	7	0	2	1	0	1	0	0	edible

5 rows × 23 columns

Gambar 6. Hasil Transformasi Dataset

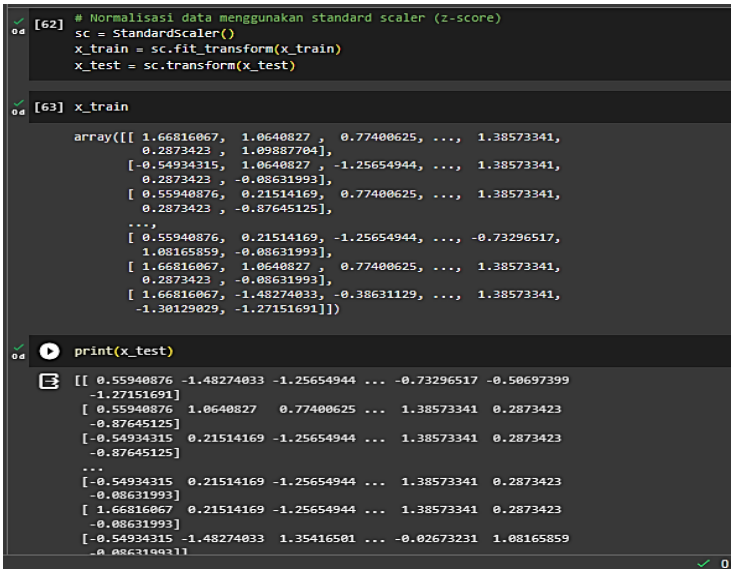
d. Membagi dataset
Proses selanjutnya dalam penelitian ini adalah pemisahan data. Yang pertama adalah pemisahan label dari dataset. Kemudian, dataset akan dibagi menjadi dua bagian, dengan 80% data digunakan sebagai data latih dan 20% sisanya sebagai data uji. Pemisahan data ini dilakukan berdasarkan label, yaitu antara 'edible' dan 'poison'. Untuk script python yang digunakan untuk membagi dataset dan hasilnya, dapat merujuk pada gambar 7.

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 1, November 2023, Page 10-19
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i1.4511



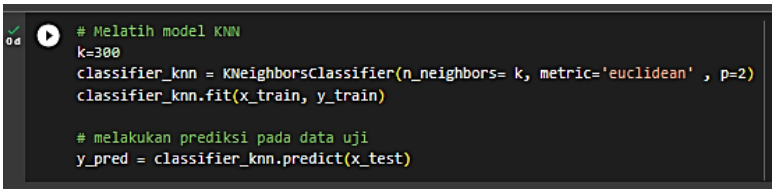
Gambar 7. Script Python dan Hasil Membagi Dataset

e. Melakukan normalisasi menggunakan *standard scaler* (*z-score*)
Langkah selanjutnya, dalam penelitian ini, kami melakukan normalisasi data dengan menggunakan metode *StandardScaler (z-score)*. Dalam proses ini, setiap atribut dalam data disesuaikan sehingga rata-rata atribut menjadi nol dan deviasi standar menjadi satu. Untuk script *python* yang digunakan untuk normalisasi dan hasilnya dapat dilihat pada gambar 8.



Gambar 8. Script Python dan Hasil Normalisasi *Standard Scale*

f. Melakukan pelatihan model K-NN
Langkah berikutnya dalam penelitian ini adalah melatih model *K-Nearest Neighbors*. Dalam penelitian ini melakukan pelatihan dengan menggunakan beberapa nilai k yang berbeda dan beberapa metrik jarak yang beragam. Hal ini bertujuan untuk memahami bagaimana performa model dipengaruhi oleh pilihan nilai k dan metrik jarak yang berbeda. Untuk script *python* yang digunkan untuk melatih model K-NN dapat dilihat pada gambar 9.

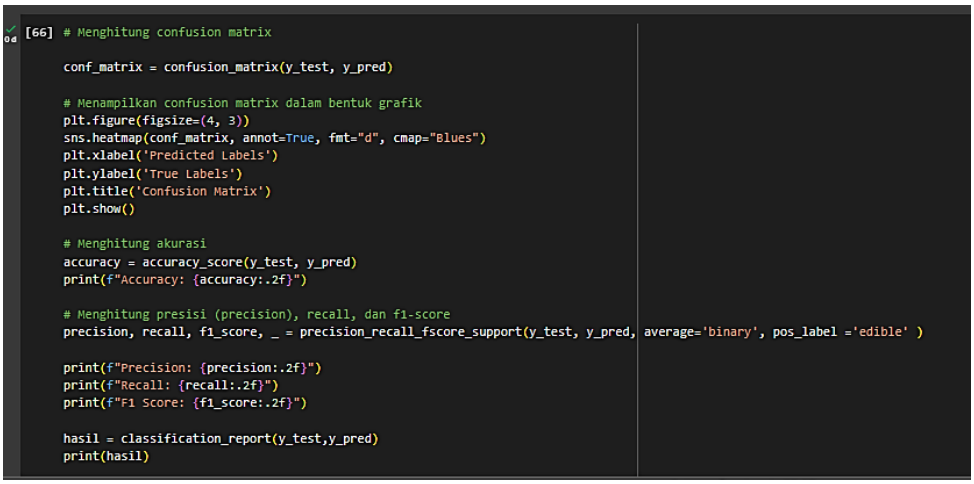


Gambar 9. Script Python Melatih Model K-NN

g. mengevaluasi model dengan metode *confusion matrix*
Langkah selanjutnya adalah melakukan evaluasi model dengan menggunakan metode *confusion matrix*. Dalam penelitian ini, kami menguji model KNN untuk mengevaluasi akurasi, presisi, recall, dan f1-score.

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 1, November 2023, Page 10-19
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i1.4511

Hasil evaluasi ini disajikan dalam bentuk grafik untuk memberikan gambaran yang lebih jelas tentang performa model K-NN. Untuk script *python* yang digunakan untuk evaluasi model dengan metode *confusion matrix* dapat dilihat pada gambar 10.



```
[66] # Menghitung confusion matrix

     conf_matrix = confusion_matrix(y_test, y_pred)

     # Menampilkan confusion matrix dalam bentuk grafik
     plt.figure(figsize=(4, 3))
     sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues")
     plt.xlabel('Predicted Labels')
     plt.ylabel('True Labels')
     plt.title('Confusion Matrix')
     plt.show()

     # Menghitung akurasi
     accuracy = accuracy_score(y_test, y_pred)
     print(f"Accuracy: {accuracy:.2f}")

     # Menghitung presisi (precision), recall, dan f1-score
     precision, recall, f1_score, _ = precision_recall_fscore_support(y_test, y_pred, average='binary', pos_label ='edible' )

     print(f"Precision: {precision:.2f}")
     print(f"Recall: {recall:.2f}")
     print(f"F1 Score: {f1_score:.2f}")

     hasil = classification_report(y_test,y_pred)
     print(hasil)
```

Gambar 10. Script Python untuk Evaluasi Model Dengan *Confusion Matrix*

3.2 Evaluasi Hasil

Tentang Hasil dari evaluasi menggunakan empat indikator, yakni akurasi, presisi, *recall*, dan *f1-score* perbandingan hasil keempatnyanya dengan jarak metrik *eulidean*, *manhattan*, *minkowski* dan nilai K=100, K=300, K=500, K=800, K=1000, K=2000, K=3000. Pada hasil evaluasi telah ditampilkan pada tabel 2,3,4 dan pada gambar 11 dan 14 telah ditamplkan grafik dari hasil evaluasi.

Table 2. Hasil evaluasi nilai K, Akurasi, Presisi, Recall,dan F1-Score dari Metrik *Euclidean*

Nilai K	Akurasi	Presisi	*Recall*	*F1-Score*
100	0,976	0,963	0,987	0,975
300	0,941	0,923	0,957	0,940
500	0,881	0,895	0,854	0,874
800	0,874	0,903	0,827	0,863
1000	0,865	0,889	0,822	0,854
2000	0,779	0,747	0,815	0,780
3000	0,724	0,678	0,815	0,740

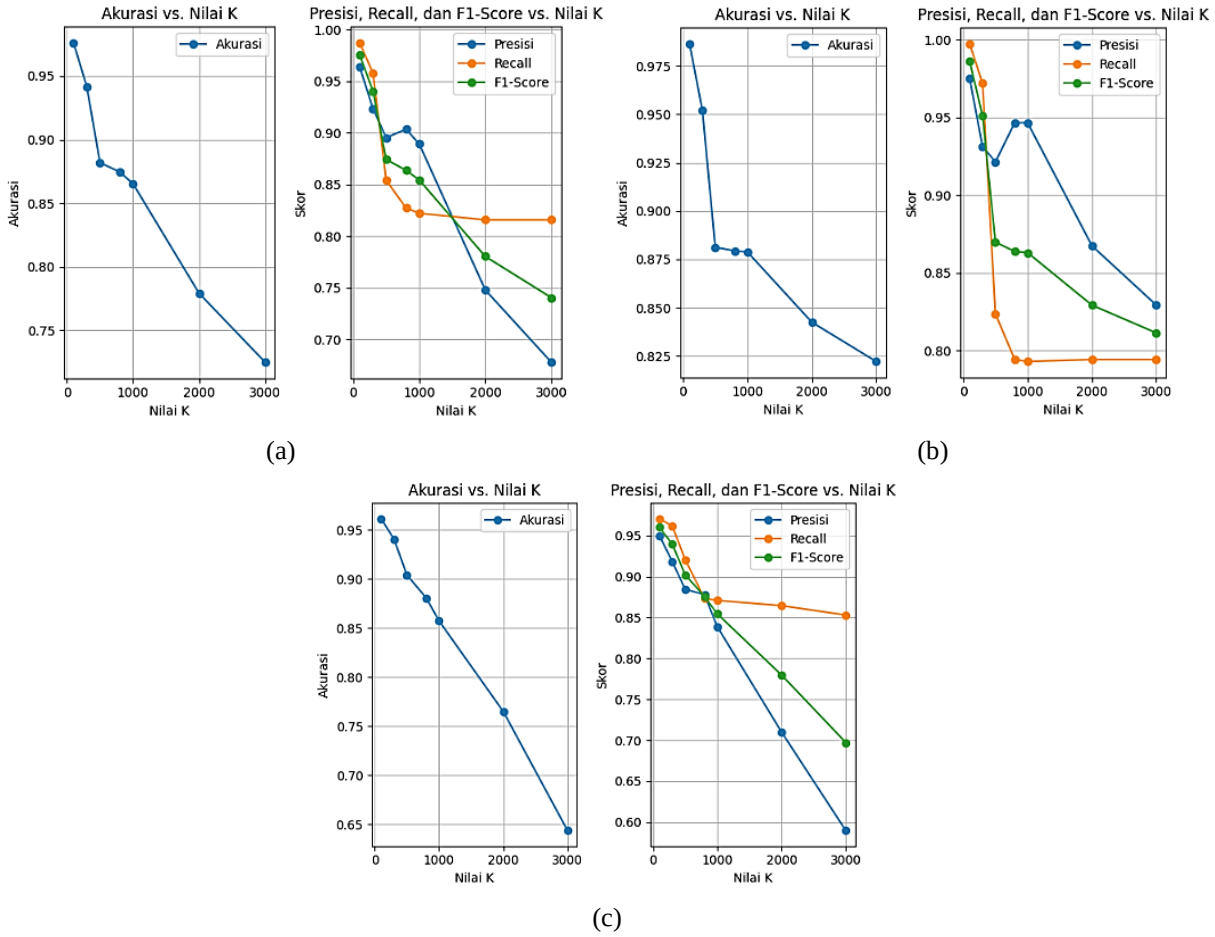
Table 3. Hasil evaluasi Nilai K, Akurasi, Presisi, Recall, dan F1-Score dari Metrik Manhattan

Nilai K	Akurasi	Presisi	*Recall*	*F1-Score*
100	0,986	0,975	0,997	0,986
300	0,952	0,931	0,971	0,951
500	0,881	0,921	0,823	0,869
800	0,879	0,9466	0,794	0,863
1000	0,878	0,9465	0,792	0,862
2000	0,842	0,867	0,794	0,829
3000	0,822	0,829	0,794	0,811

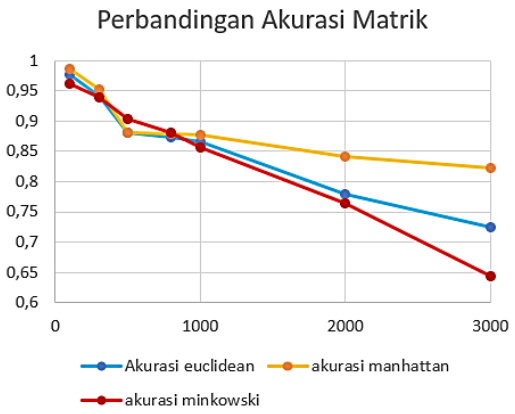
Table 3. Hasil evaluasi nilai K, Akurasi, Presisi, Recall, dan F1-Score dari Metrik *Minkowski* Parameter 3

Nilai K	Akurasi	Presisi	*Recal*	*F1-Score*
100	0,961	0,949	0,970	0,960
300	0,940	0,918	0,961	0,939
500	0,904	0,884	0,920	0,902
800	0,880	0,877	0,873	0,875
1000	0,857	0,838	0,870	0,854
2000	0,764	0,710	0,864	0,779
3000	0,643	0,589	0,852	0,697

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 1, November 2023, Page 10-19
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i1.4511



Gambar 11. (a) Hasil evaluasi Dengan Jarak Metrik Euclidean, (b) Hasil Evaluasi Dengan Jarak Metrik Manhattan, (c)



Gambar 11. Hasil Perbandingan Akurasi dari Metrik *Eulidean*, *Manhattan*, dan *Minkowski*

4. KESIMPULAN

Dari penelitian ini dapat disimpulkan beberapa hal dalam klasifikasi jamur beracun menggunakan algoritma k-*nearest neighbor* adalah dari hasil evaluasi dengan beberapa jarak metrik dapat disimpulkan bahwa nilai akurasi metrik *manhattan* lebih baik dari metrik *euclidean* dan *minkowski*. Karena metrik *manhattan* mendapatkan hasil akurasi tertinggi yaitu 99% dengan K = 100 dan terendah 82% dengan K = 3000 , sedangkan metrik *euclidean* mendapatkan hasil akurasi dengan nilai 98% dengan K = 100 dan 72% dengan K = 3000, dan pada metrik *minkowski* mendapatkan hasil akurasi dengan nilai 96% pada K = 100 dan 64% pada K = 3000. Pada nilai presisi metrik *manhattan* mendapatkan nilai presisi lebih baik yaitu 98% dengan K = 100. Sedangkan metrik *euclidean* mendapatkan nilai presisi 96% dan metrik *minkowski* mendapatkan nilai presisi 95%. Pada nilai *recall* metrik *manhattan* mendapatkan nilai recall lebih baik yaitu 99%. Sedangkan metrik *euclidean* mendapatkan nilai *recall*

ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 5, No. 1, November 2023, Page 10-19
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v5i1.4511

96% dan metrik *minkowski* mendapatkan nilai *recall* 95%. Pada nilai *f1-score* metrik *manhattan* mendapatkan nilai *f1-score* lebih baik yaitu 98%. Sedangkan metrik *euclidean* mendapatkan nilai *f1-score* 97% dan metrik *minkowski* mendapatkan nilai *f1-score* 96%. Berdasarkan evaluasi pengujian beberapa nilai K, kesimpulan yang dapat ditarik adalah bahwa semakin meningkatnya nilai K, akan mengakibatkan penurunan persentase yang diperoleh.

REFERENCES

[1] 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.

[2] Y. Yohannes, D. Udjulawa, and T. Ivan Sariyo, "Klasifikasi Jenis Jamur Menggunakan SVM dengan Fitur HSV dan HOG," *Petir*, vol. 15, no. 1, pp. 113–120, 2022, doi: 10.33322/petir.v15i1.1101.

[3] E. Praja, W. Mandala, D. E. Putri, and R. Permana, "Penerapan Data Mining untuk Klasifikasi Hasil Panen Jamur Tiram Menggunakan Algoritma K-Nearest Neighbors," *J. Media Inform. Budidarma*, vol. 7, no. 1, pp. 223–230, 2023, doi: 10.30865/mib.v7i1.5252.

[4] Putra Ivan Permana, "KASUS-KASUS KERACUNAN JAMUR LIAR DI INDONESIA Poisoning Cases of Wild Edible Mushrooms in Indonesia," *J. Ekol. Kesehat.* , vol. 20, pp. 215–230, 2021, [Online]. Available: https://doi.org/10.22435/jek.v20i3.4943

[5] I. P. Putra, "Kasus keracunan Inocybe sp. di Indonesia," *Pros. Semin. Nas. Biol.*, no. September, pp. 148–153, 2020.

[6] H. Rofiq, K. C. Pelangi, and Y. Lasena, "Penerapan Data Mining Untuk Menentukan Potensi Hujan Harian Dengan Menggunakan Algoritma 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

[7] A. M. Argina, "Penerapan Metode Klasifikasi K-Nearest Neigbor pada Dataset Penderita Penyakit Diabetes," *Indones. J. Data Sci.*, vol. 1, no. 2, pp. 29–33, 2020, doi: 10.33096/ijodas.v1i2.11.

[8] D. A. Nasution, H. H. Khotimah, and N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," *Comput. Eng. Sci. Syst. J.*, vol. 4, no. 1, p. 78, 2019, doi: 10.24114/cess.v4i1.11458.

[9] D. N. Anisa and Jumanto, "Klasifikasi Penyakit Diabetes Menggunakan Algoritma Naive Bayes," *Din. Inform.*, vol. 14, no. 1, pp. 33–42, 2022.

[10] A. Setiawan, "Perbandingan Penggunaan Jarak Manhattan, Jarak Euclid, dan Jarak Minkowski dalam Klasifikasi Menggunakan Metode KNN pada Data Iris," *J. Sains dan Edukasi Sains*, vol. 5, no. 1, pp. 28–37, 2022, doi: 10.24246/juses.v5i1p28-37.

[11] E. Purwaningsih and E. Nurelasari, "Penerapan K-Nearest Neighbor Untuk Klasifikasi Tingkat Kelulusan Pada Siswa," *Syntax J. Inform.*, vol. 10, no. 01, pp. 46–56, 2021, doi: 10.35706/syji.v10i01.5173.

[12] M. K. Neighbor, "Prediksi penjualan produk unilever menggunakan metode k-nearest neighbor," vol. 06, pp. 155–160, 2021.

[13] A. Indriani, "Analisa Perbandingan Metode Naïve Bayes Classifier Dan K-Nearest Neighbor Terhadap Klasifikasi Data," *Sebatik*, vol. 24, no. 1, pp. 1–7, 2020, doi: 10.46984/sebatik.v24i1.909.

[14] F. Alghifari and D. Juardi, "Penerapan Data Mining Pada Penjualan Makanan Dan Minuman Menggunakan Metode Algoritma Naïve Bayes," 2021.

[15] R. G. Whendasmoro and J. Joseph, "Analisis Penerapan Normalisasi Data Dengan Menggunakan Z-Score Pada Kinerja Algoritma K-NN," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 4, p. 872, 2022, doi: 10.30865/jurikom.v9i4.4526.

[16] A. S. REZKI, "Klasifikasi Emosi Pada Twitter Dengan Metode K-Nearest Neighbor (Knn) Tugas Akhir," 2021.

[17] I. Loelianto, M. S. S. Thayf, and H. Angriani, "Implementasi Teori Naive Bayes Dalam Klasifikasi Calon Mahasiswa Baru Stmik Kharisma Makassar," *SINTECH (Science Inf. Technol. J.*, vol. 3, no. 2, pp. 110–117, 2020, doi: 10.31598/sintechjournal.v3i2.651.

[18] K. Neighbor *et al.*, "Klasifikasi Penentuan Pengajuan Kartu Kredit Menggunakan K-Nearest Neighbor," vol. 22, no. 1, pp. 73–82, 2020.

[19] L. Farokhah, "Implementasi K-Nearest Neighbor untuk Klasifikasi Bunga Dengan Ekstraksi Fitur Warna RGB," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 6, pp. 1129–1136, 2020, doi: 10.25126/jtiik.2020722608.