HOAQ: JURNAL TEKNOLOGI INFORMASI, Volume 17 Nomor 2, Desember 2026, p351-361
p-ISSN: 2337-5280, e-ISSN: 26207427
DOI: https://doi.org/10.52972/hoaq.vol16no2

PERBANDINGAN METODE DECISION TREE CART DAN K-NEAREST NEIGHBORS UNTUK KLASIFIKASI KUALITAS KISMIS PADA RAISIN DATASET

Melly Agustin[1], Fildzah Syarafina Sumantri[2], Laras Ayu Pradanti[3], Alfan Rizaldy Pratama[4]
[1-4] Program Studi Sains Data, Universitas Pembangunan Nasional Veteran Jawa Timur
Jl. Raya Rungkut Madya, Gunung Anyar, Surabaya – Jawa Timur, Indonesia
Email: [1]24083010103@student.upnjatim.ac.id, [2]24083010053@student.upnjatim.ac.id,
[3]24083010008@student.upnjatim.ac.id, [4]alfan.fasilkom@upnjatim.ac.id

ABSTRAK

Kismis merupakan salah satu produk pertanian yang memiliki nilai ekonomis tinggi, sehingga klasifikasi kualitasnya menjadi aspek penting dalam proses seleksi dan distribusi. Penelitian ini bertujuan untuk membandingkan kinerja metode *Decision Tree CART* dan *K-Nearest Neighbors (KNN)* dalam mengklasifikasikan jenis kismis pada raisin *dataset* yang terdiri dari 900 data dengan dua kelas yaitu, besni dan kecimen. *Dataset* memiliki tujuh fitur morfologi, meliputi area, major axis length, minor axis length, eccentricity, convex area, extent, dan perimeter. Tahap *preprocessing* mencakup penanganan *outlier* menggunakan metode capping IQR, encoding label, standarisasi fitur dengan *StandardScaler*, serta pembagian data latih dan uji dengan rasio 80:20. Hasil evaluasi menunjukkan bahwa metode KKN dengan nilai k optimal sebesar 16 menghasilkan akurasi 88,33%, presisi 84,16%, recall 94,44%, dan FI-Score 89,01%. Sementara itu, *Decision Tree CART* memperoleh akurasi 84,44%, presisi 80,39%, recall 91,11%, dan F1-Score 85,42%. Berdasarkan perbandingan keempat metrik evaluasi tersebut, KNN terbukti unggul dibandingkan *Decision Tree CART* dalam mengklasifikasikan kualitas kismis pada *dataset* ini. KKN direkomendasikan sebagai metode yang lebih sesuai untuk klasifikasi kismis dengan karakteristik *dataset* serupa.

Kata kunci: *Decision Tree*, CART, *K-Nearest Neighbors*, Klasifikasi, Raisin *Dataset*

ABSTRACT

Raisins are one of the high-value agricultural products, making quality classification a critical aspect of their selection and distribution process. This study aims to compare the performance of the Decision Tree CART and K-Nearest Neighbors (KKN) methods in classifying raisin varieties using the raisin dataset, which consists of 900 instances with two classes: besni and kecimen. The dataset includes seven morphological features: area, major axis length, minor axis length, eccentricity, convex area, extent, and perimeter. Preprocessing steps included outlier handling using IQR capping, label encoding, feature standardization with StandardScaler, and an 80:20 train-test split. Evaluation results show that KNN with an optimal K value of 16 achieved an accuracy of 88,33%, precision of 84,16%, recall of 94,44%, and F1-Score of 89,01%. Decision Tree CART obtained an accuracy of 84,44%, precision of 80,39%, recall of 91,11%, and F1-Score of 85,42%. Based on all four evaluation metrics, KNN outperformed Decision Tree CART in classifying raisin quality on this dataset. KKN is therefore recommended as the more suitable method for raisin classification tasks with similar dataset characteristics.

Keywords: Decision Tree, CART, K-Nearest Neighbors, Classification, Raisin Dataset

1. PENDAHULUAN

Kismis merupakan salah satu hasil olahan anggur yang populer di pasaran, diproduksi melalui proses pengeringan alami maupun mekanis. Warna dan kualitas kismis umumnya dipengaruhi oleh varietas anggur yang digunakan serta metode pengeringannya. Penilaian kualitas secara manual oleh manusia dinilai kurang efisien karena membutuhkan waktu yang lama dan rentan terhadap kesalahan akibat human *error* [1]. Berdasarkan data dari konferensi negara-negara penghasil kismis tanpa biji ke-63, Turki mendominasi produksi dan ekspor kismis dunia untuk musim 2022/2023 [2]. Tingginya volume produksi ini menegaskan kebutuhan mendesak terhadap sistem klasifikasi otomatis yang cepat, konsisten, dan akurat.

Data mining merupakan sebuah proses penggalian hubungan, pola, atau tren bermakna dari kumpulan data tersimpan dalam suatu repositori dengan memanfaatkan teknik pengenalan pola seperti statistik atau matematika [1]. Penerapan data *mining* di sektor pertanian dan industri pangan telah terbukti mampu meningkatkan efisiensi proses identifikasi dan klasifikasi produk secara signifikan [3]. Klasifikasi merupakan salah satu teknik data *mining* yang membangun model dari sampel untuk mengelompokkannya ke dalam kelas-kelas yang telah ditentukan. Teknik ini dibagi menjadi lima kategori berdasarkan pendekatan matematikanya yaitu berbasis

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

statistik, jarak, pohon keputusan, jaringan syaraf, dan *rule*. Di antara berbagai algoritma yang tersedia *decision tree* dan KNN (*K-Nearest Neighbors*) merupakan dua algoritma yang paling banyak digunakan dalam klasifikasi data [4].

Decision tree merupakan algoritma yang umum digunakan dalam pengambilan keputusan, di mana setiap pohon memiliki cabang yang mewakili suatu atribut yang harus dipenuhi hingga berakhir disimpul daun [1]. Secara khusus, varian *decision tree* CART (*Classification and Regression Tree*) menggunakan kriteria impuritas Gini dalam proses pemisahan data pada setiap *node* untuk menghasilkan pohon keputusan biner [5]. Algoritma KNN bekerja dengan mengklasifikasikan data baru berdasarkan kedekatan jaraknya terhadap sejumlah K data tetangga terdekat dalam data latih [4]. Pemilihan kedua algoritma ini didasarkan pada beberapa pertimbangan. Pertama, *decision tree* CART dan KNN mewakili dua paradigma klasifikasi yang berbeda, yaitu pendekatan berbasis aturan/pohon keputusan (*rule-based*) pada CART dan pendekatan berbasis kemiripan jarak (*instance-based*) pada KNN, sehingga perbandingan keduanya dapat menunjukkan paradigma mana yang lebih sesuai dengan karakteristik fitur morfologi pada raisin *dataset*. Kedua, CART bersifat interpretatif karena aturan klasifikasinya dapat divisualisasikan dalam bentuk pohon keputusan dan menghasilkan nilai *feature importance* yang menunjukkan kontribusi tiap atribut morfologi terhadap keputusan klasifikasi, sedangkan KNN bersifat non-parametrik dan tidak memerlukan asumsi terhadap distribusi data, sehingga cocok diterapkan pada data numerik hasil ekstraksi citra yang pola sebarannya belum tentu mengikuti distribusi tertentu. Ketiga, kedua algoritma ini relatif ringan secara komputasi dan tidak memerlukan data berjumlah besar untuk pelatihan, sehingga sesuai dengan ukuran raisin *dataset* yang tergolong moderat (900 data), berbeda dengan pendekatan *deep learning* yang umumnya memerlukan volume data jauh lebih besar untuk mencapai performa optimal. Kedua algoritma ini dikenal memiliki performa yang kompetitif namun dengan karakteristik yang berbeda, sehingga perbandingan keduanya relevan untuk menentukan algoritma yang paling sesuai pada kasus klasifikasi kismis berdasarkan fitur morfologi [6].

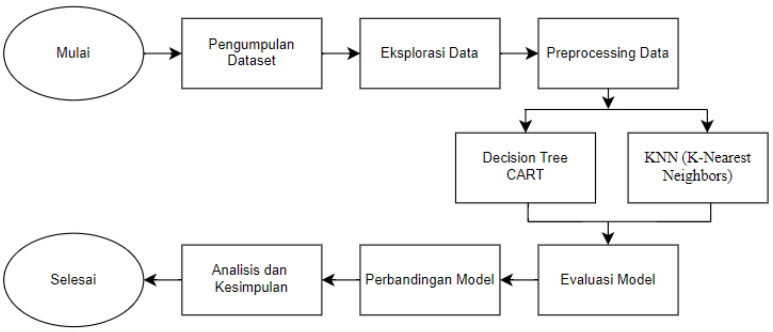
Beberapa penelitian terdahulu telah mengkaji klasifikasi kismis maupun perbandingan metode klasifikasi. [1] membandingkan lima algoritma klasifikasi pada *raisin dataset* yaitu *naïve bayes, decision tree*, *random forest*, *neural network*, dan SVM (*Support Vector Machine*) , dan menemukan bahwa neural network menghasilkan akurasi tertinggi sebesar 86,81%, namun evaluasi yang dilakukan hanya berfokus pada akurasi tanpa mempertimbangkan metrik lain seperti presisi, *recall*, dan f1-*score*, serta belum membahas potensi *overfitting* pada model yang dibangun. [5] menerapkan KNN dengan dua metrik jarak yang berbeda untuk mengklasifikasikan kismis besni dan kecimen, dengan hasil akurasi terbaik sebesar 86,7% menggunakan metrik *manhattan*, namun nilai K yang diuji terbatas pada rentang kecil (3,5,7,9) sehingga belum tentu merepresentasikan K optimal secara keseluruhan. Tangkelayuk dan Mailoa membandingkan KNN, *naïve bayes*, dan *decision tree* dengan akurasi sebesar 80,84%, namun belum melakukan evluasi menyeluruh menggunakan presisi, *recall*, dan f1-*score*, serta belum mengoptimalkan parameter K pada KNN [4]. Secara eksplisit, terdapat tiga celah penelitian (*research gap*) yang teridentifikasi dari kajian tersebut: (1) evaluasi performa model umumnya masih berfokus pada akurasi saja tanpa metrik pelengkap seperti presisi, *recall*, dan f1-*score*; (2) optimasi parameter K pada KNN belum dilakukan secara sistematis pada rentang nilai yang memadai; dan (3) belum ada penelitian yang secara spesifik membandingkan *decision tree* CART dan KNN pada raisin *dataset* sekaligus menelaah indikasi *overfitting* pada masing-masing model. Ketiga celah tersebut menjadi urgensi penting mengingat performa kedua algoritma dapat bervariasi tergantung karakteristik data yang digunakan, sehingga tanpa evaluasi yang komprehensif, pemilihan metode klasifikasi kismis yang optimal sulit dipastikan.

Berdasarkan celah tersebut, penelitian ini memberikan kontribusi dalam tiga hal. Pertama, penelitian ini melakukan evaluasi performa *decision tree* CART dan KNN secara komprehensif menggunakan empat metrik evaluasi sekaligus, yaitu akurasi, presisi, *recall*, dan f1-*score*, sehingga perbandingan kedua metode menjadi lebih menyeluruh dan tidak bias terhadap satu metrik saja. Kedua, penelitian ini melakukan optimasi nilai K pada KNN secara sistematis pada rentang K=1 hingga K=20 untuk memperoleh nilai K optimal, alih-alih menggunakan rentang nilai terbatas seperti pada penelitian sebelumnya. Ketiga, penelitian ini menyertakan analisis indikasi *overfitting* pada model CART melalui perbandingan *train accuracy* dan *test accuracy*, yang belum dibahas secara eksplisit pada penelitian-penelitian terdahulu. Dengan demikian, penelitian ini bertujuan untuk membandingkan kinerja metode decision tree CART dan KNN dalam mengklasifikasikan dua varietas kismis pada raisin *dataset* berdasarkan tujuh fitur morfologi. Manfaat yang diharapkan adalah memberikan rekomendasi algoritma yang lebih tepat untuk digunakan dalam sistem klasifikasi kismis otomatis, sekaligus berkontribusi pada pengembangan aplikasi data *mining* di bidang pertanian dan industri pangan.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental komparatif untuk membandingkan kinerja dua algoritma klasifikasi, yaitu decision tree CART dan KNN (*K-Nearest Neighbors*), dalam mengklasifikasikan varietas kismis pada raisin *dataset*. Tahapan penelitian meliputi pengumpulan data, *preprocessing*, pemodelan, dan evaluasi model.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi



Gambar 1. *Research Pipeline* Penelitian

Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah raisin dataset yang bersumber dari UCI *Machine Learning Repository* [7]. *Dataset* ini terdiri dari 900 data dengan dua kelas yaitu kecimen dan besni, masing-masing sebanyak 450 data. Terdapat tujuh fitur morfologi yang digunakan sebagai atribut input sebagaimana ditampilkan pada tabel 1.

Tabel 1. Atribut raisin dataset

No	Atribut	Deskripsi
1	Area	Jumlah piksel dalam batas area kismis
2	Major Axis Length	Panjang sumbu utama kismis
3	Minor Axis Length	Panjang sumbu minor kismis
4	Eccentricity	Ukuran eksentrisitas elips kismis
5	Convex Area	Jumlah piksel pada cangkang cembung kecil
6	Extent	Rasio wilayah kismis terhadap kontak pembatas
7	Perimeter	Panjang keliling batas kismis
8	Class	Kelas target: kecimen atau besni

Eksplorasi Data

Tahap eksplorasi data berfungsi memahami karakteristik dataset sebelum dianalisis lebih lanjut. Proses ini terdiri dari pengecekan tipe data, identifikasi missing value, pengecekan data duplikat, dan deteksi *outlier*.

***Preprocessing* Data**

Tahap *preprocessing* merupakan langkah penting sebelum pemodelan untuk memastikan kualitas data yang digunakan [8]. Pada penelitian ini, tahap preprocessing mencakup empat proses utama. Pertama ada penanganan outlier dengan menggunakan *capping* IQR. Pemilihan metode IQR untuk penanganan outlier didasarkan pada karakteristik fitur pada raisin *dataset* yang seluruhnya berupa fitur morfologi hasil ekstraksi citra (seperti luas, perimeter, panjang sumbu mayor dan minor, *eccentricity*, *convex area*, dan *extent*), yang secara alami cenderung memiliki distribusi tidak normal (*skewed*) akibat variasi bentuk dan ukuran biji kismis. Metode deteksi *outlier* berbasis z-*score* kurang tepat digunakan pada kondisi ini karena mengasumsikan data berdistribusi normal, sedangkan IQR bersifat non-parametrik sehingga lebih *robust* terhadap distribusi data yang tidak normal maupun skewed. Selain itu, teknik *capping* dipilih dibandingkan penghapusan (*removal*) data *outlier* agar jumlah data tetap terjaga mengingat ukuran dataset yang relatif terbatas (900 data), sehingga informasi pada data tidak hilang namun pengaruh nilai ekstrem terhadap proses pemodelan tetap dapat diminimalkan. *Outlier* merupakan nilai yang menyimpang jauh dari distribusi umum data dan dapat mempengaruhi performa model secara signifikan [8]. Penanganan *outlier* dilakukan menggunakan metode *capping* berbasis IQR (*Interquartile Range*). Nilai IQR dihitung sebagai selisih antara kuartil ketiga (Q3) dan kuartil pertama (Q1), sebagaimana dirumuskan pada persamaan (1) berikut:

$$IQR = Q3 - Q1 \quad (1)$$

Keterangan:

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

IQR = selisih interkuartil
$Q3$ = kuartil ketiga
IQR = kuartil pertama
Batas bawah dan atas ditentukan menggunakan persamaan (2) dan (3)berikut:

$$Batas\ bawah = Q1 - 1{,}5 \times IQR \quad (2)$$

$$Batas\ atas = Q3 + 1{,}5 \times IQR \quad (3)$$

Nilai yang berada di bawah batas bawah diganti dengan nilai batas bawah, sedangkan nilai yang melebihi batas atas akan diganti dengan nilai batas atas, sehingga informasi tetap terjaga. Kedua ada label encoding di mana variabel target yang semua berupa data kategorikal (besni dan kecimen) dikonversi menjadi representasi numerik (0 dan 1). Label encoding dipilih dibandingkan *one-hot encoding* karena variabel target bersifat biner (hanya terdiri dari dua kelas), sehingga representasi numerik tunggal sudah cukup merepresentasikan kedua kelas tanpa menimbulkan dimensi tambahan maupun risiko misinterpretasi hubungan ordinal antarkelas, sekaligus memungkinkan data diproses oleh algoritma data *mining*[9]. Ketiga ada standarisasi fitur dengan StandardScaler. Dilakukan menggunakan metode *z-score normalization* (StandardScaler) agar setiap fitur memiliki skala yang setara dengan rata-rata nol dan standar deviasi satu[8]. Proses ini sangat penting terutama pada algoritma berbasis jarak seperti KNN, di mana fitur berskala besar akan mendominasi perhitungan jarak apabila tidak distandarisasi. Standarisasi dilakukan menggunakan persamaan (4) berikut:

$$z = \frac{x-\mu}{\sigma} \quad (4)$$

Keterangan:
z = nilai hasil standarisasi
x = nilai asli futur
μ = rata-rata fitur
σ = standar deviasi fitu

Terakhir ada pembagian data. *Dataset* dibagi menjadi data latih dan data uji dengan rasio 80:20 menggunakan *stratified splitting* untuk menjaga proporsi kelas tetap seimbang. Rasio 80:20 dipilih karena merupakan proporsi umum yang cukup untuk menyediakan data latih yang memadai bagi proses pembelajaran model, sekaligus menyisakan jumlah data uji yang representatif untuk mengevaluasi performa model secara andal, terutama mengingat ukuran *dataset* yang relatif terbatas (900 data). Teknik *stratified splitting* digunakan agar proporsi kelas tetap seimbang pada data latih maupun data uji, mengingat distribusi kelas pada raisin *dataset* yang seimbang (450:450) perlu dipertahankan agar evaluasi model tidak bias terhadap salah satu kelas. Dari total 900 data, diperoleh 720 data latih dan 180 data uji.

Decision Tree CART

Decision tree merupakan algoritma klasifikasi yang membangun struktur pohon keputusan melalui pemisahan data secara rekursif berdasarkan atribut yang paling informatif. Secara khusus, varian CART (*Classification and Regression Tree*) menghasilkan pohon keputusan biner, dimana setiap node internal hanya dapat memiliki dua cabang[10]. Berbeda dari algoritma pohon keputusan lainnya seperti ID3 atau C4.5 yang menggunakan information gain atau gain ratio, CART menggunakan indeks *gini* (*Gini Impurity*) sebagai kriteria pemilihan atribut pada setiap pemisahan[10]. Indeks *gini* untuk suatu node t dihitung menggunakan persamaan (5) berikut:

$$Gini\ (t) = 1 - \sum_{k-1}^{K} \quad [P(C_K \mid t)\ ]^2 \quad (5)$$

Keterangan:
$Gini\ (t)$ = nilai impuritas Gini pada node t
K = jumlah kelas
$P(C_K \mid t)$ = proporsi sampel kelas c_k pada node t
Nilai *gini*=0 menunjukkan node murni, sedangkan nilai mendekati 0,5 menandakan tingkat ketidakmurnian tertinggi pada klasifikasi biner. Pemilihan pemisahan terbaik dilakukan dengan memaksimalkan penurunan impuritas menggunakan persamaan (6) berikut:

$$\Delta Gini = Gini(t) - \left[\frac{N_L}{N}.Gini(t_L) + \frac{N_R}{N}.Gini(t_R)\right] \quad (6)$$

Keterangan:
$\Delta Gini$ = penurunan impuritas Gini
$Gini(t)$ = nilai impuritas node induk
N = jumlah total sampel pada node induk
N_L = jumlah sampel pada node anak kiri

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

N_R = jumlah sampel pada node anak kanan
$Gini(t_L)$ = nilai impuritas node anak kiri
$Gini(t_R)$ = nilai impuritas node anak kanan

Proses ini diulang secara rekursif pada setiap *node* hingga memenuhi kriteria berhenti, seperti pencapaian kedalaman pohon maksimum atau batas minimum jumlah sampel per *node*. Abedinia dan Seydi[11] menjelaskan bahwa kualitas model *decision tree* sangat dipengaruhi oleh kompleksitas struktur pohon yang terbentuk selama proses pelatihan. Pohon yang terlalu dalam mampu menangkap pola yang sangat spesifik pada data latih, tetapi berpotensi menurunkan kemampuan model dalam melakukan generalisasi terhadap data baru. Halabaku dan Bytyçi[12] juga menyatakan bahwa Decision Tree merupakan algoritma yang rentan mengalami *overfitting* apabila pertumbuhan pohon tidak dibatasi. Risiko tersebut dapat dikurangi melalui teknik *pre-pruning*, *post-pruning*, maupun pengaturan hyperparameter seperti *max_depth*, *min_samples_split*, dan *min_samples_leaf* sehingga model memiliki kemampuan generalisasi yang lebih baik.

K-Nearest Neighbors (KNN)

KNN merupakan algoritma klaifikasi non-parametrik berbasis *instance* yang mengklasifikasikan data baru berdasarkan mayoritas kelas dari k tetangga terdekatnya dalam ruang fitur pelatihan[13], [14]. KNN tidak membangun model eksplisit selama pelatihan (*lazy training*), melainkan menyimpan seluruh data latih dan melakukan komputasi pada saat fase prediksi[4]. Menurut Halder et al.[15] performa algoritma *K-Nearest Neighbors* sangat dipengaruhi oleh pemilihan nilai *K*, metrik jarak, serta skala fitur yang digunakan. Oleh karena itu, proses standarisasi data menjadi tahapan penting agar setiap atribut memberikan kontribusi yang seimbang dalam perhitungan jarak sehingga akurasi klasifikasi dapat ditingkatkan. Perhitungan kedekatan jarak antar data menggunakan *euclidean distance* menggunakan persamaan (7) berikut:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} \quad (x_i - y_i)^2} \tag{7}$$

Keterangan:
$d(x,y)$ = jarak *euclidean* antara dua titik data
x_i = nilai fitur ke-i dari data x
y_i = nilai fitur ke-i dari data y
n = jumlah fitur
Setelah k tetangga terdekat ditemukan, kelas data baru ditentukan berdasarkan mayoritas voting menggunakan persamaan (8) berikut:

$$\hat{y} = arg\ max_c \sum_{i=1}^{k} \quad 1(y_i = c) \tag{8}$$

Keterangan:
$\hat{y}$ = kelas hasil prediksi
c = kelas yang dievaluasi
k = jumlah tetangga terdekat
y_i = kelas tetangga ke-1
$1(y_i = c)$ = fungsi indikator bernilai 1 jika $y_i = c$, dan 0 jika sebaliknya
Penentuan nilai k optimal dilakukan melalui eksperimen dengan berbagai nilai k dan dipilih berdasarkan akurasi tertinggi pada data uji.

Evaluasi Model

Kinerja kedua model dievaluasi menggunakan empat metrik berbasis *confusion matrix*[16]. *Confusion matrix* merupakan tabel yang merangkum hasil prediksi model dibandingkan nilai sebenarnya, yang terdiri dari *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Keempat metrik evaluasi yang digunakan yaitu *accuracy*, *precision*, *recall*, dan f1-*score*, dirumuskan pada persamaan (9) hingga persamaan (12) berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

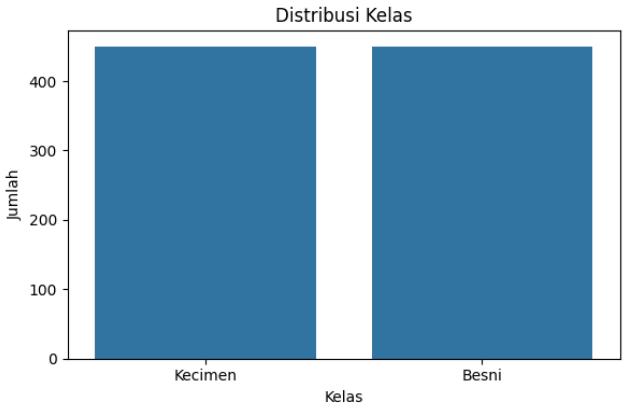
Keterangan:
TP = prediksi positif yang benar
TN = prediksi negatif yang benar
FP = prediksi positif yang salah
FN = prediksi negatif yang salah

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil pengujian dan analisis perbandingan metode *decision tree* CART dan KNN (*K-Nearest Neighbors*) dalam mengklasifikasikan varietas kismis (besni dan kecimen) pada raisin *dataset*. Pengujian dilakukan melalui serangkaian tahapan yang meliputi eksplorasi data, *preprocessing*, pemodelan, dan evaluasi model menggunakan metrik, akurasi, presisi, *recall*, dan *f1-score*.

Eksplorasi Data

Dataset yang digunakan terdiri dari 900 data dengan 8 kolom, yaitu 7 fitur numerik dan 1 kolom target (*class*). Berdasarkan hasil pengecekan menggunakan df.isnull().sum(), seluruh kolom tidak memiliki *missing value* (0 nilai kosong pada semua atribut). Pengecekan data duplikat menggunakan df.duplicated().sum() juga menunjukkan hasil o, sehingga tidak ditemukan baris yang identik. Distribusi kelas pada *dataset* bersifat seimbang (*balanced*) dengan masing-masing kelas besni dan kecimen berjumlah 450 data, sebagaimana ditampilkan pada gambar 1 berikut.



Gambar 1. Distribusi Kelas Raisin *Dataset*

Deteksi *outlier* dilakukan menggunakan metode IQR pada setiap fitur numerik. Tabel 2 berikut menampilkan jumlah *outlier* yang ditemukan pada masing-masing fitur sebelum dilakukan penanganan.

Tabel 2. Jumlah *Outlier* per Fitur Sebelum Penanganan

No	Atribut	Sebelum	Sesudah
1	Area	41	0
2	Major Axis Length	17	0
3	Minor Axis Length	26	0
4	Eccentricity	43	0
5	Convex Area	42	0
6	Extent	21	0
7	Perimeter	17	0

Berdasarkan Tabel 2, atribut *eccentricity* memiliki jumlah *outlier* terbanyak yaitu 43 data. Setelah dilakukan penanganan outlier menggunakan *capping* IQR, seluruh fitur menunjukkan 0 *outlier*.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

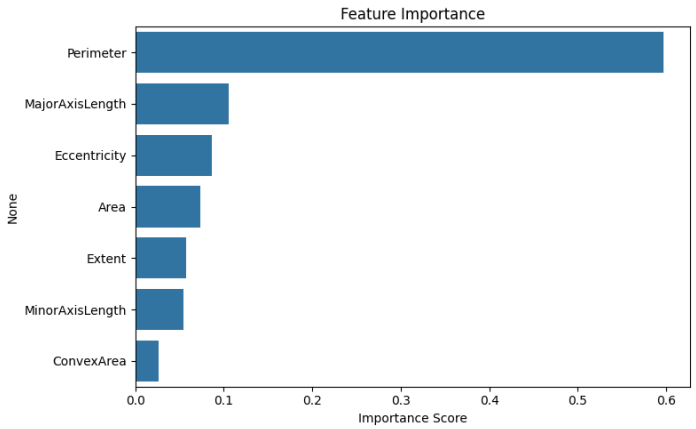
***Preprocessing* Data**

Setelah penanganan *outlier*, atribut target (*class*) dikonversi menggunakan label *encoding* dengan hasil besni=0 dan kecimen=1. Distribusi kelas setelah *encoding* tetap seimbang dengan masing-masing 450 data. Selanjutnya seluruh fitur numerik distandarisasi menggunakan StandardScaler agar memiliki skala seragam dengan rata-rata 0 dan standar deviasi 1. *Dataset* kemudian dibagi dengan rasio 80:20 menggunakan *stratified splitting*, menghasilkan 720 data latih dan 180 data uji dengan distribusi kelas yang seimbang yaitu 90 data besni dan 90 data kecimen pada data uji.

Pemodelan dan Evaluasi *Decision Tree* CART

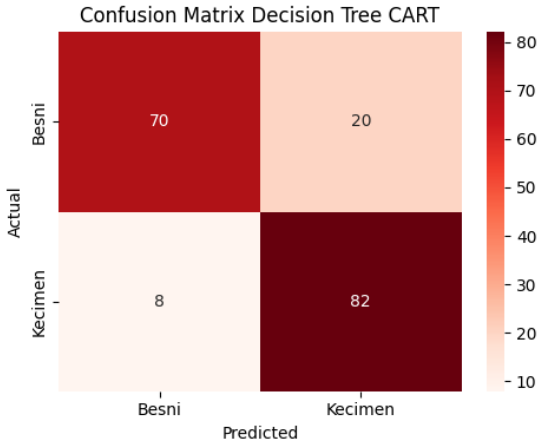
Model *decision tree* CART dibangun menggunakan parameter criterion='gini' dan random_state=42. Hasil pelatihan menunjukkan *train accuracy* sebesar 100% dan *test accuracy* sebesar 84,44%. Selisih yang signifikan antara akurasi latih dan akurasi uji ini mengindikasikan adanya *overfitting*, yaitu kondisi di mana model terlalu menyesuaikan diri dengan data latih sehingga kurang mampu menggeneralisasi pada data baru. Hal ini merupakan salah satu kelemahan umum dari algoritma *decision tree* tanpa pembatasan kedalaman pohon. *Overfitting* pada model CART terjadi karena proses pembentukan pohon keputusan dilakukan secara rekursif hingga setiap node menjadi semakin homogen terhadap data latih. Tanpa pembatasan kompleksitas pohon, seperti pengaturan *max_depth*, *min_samples_split*, atau *min_samples_leaf*, model akan terus membentuk cabang baru untuk mengakomodasi pola-pola yang sangat spesifik, termasuk *noise* yang terdapat pada data pelatihan. Akibatnya, model memiliki performa yang sangat tinggi pada data latih, tetapi mengalami penurunan kemampuan generalisasi ketika dihadapkan pada data uji. Kondisi ini tercermin dari perbedaan yang cukup besar antara *train accuracy* sebesar 100% dan *test accuracy* sebesar 84,44%.

Hasil perhitungan *feature importance* menunjukkan kontribusi masing-masing atribut dalam proses klasifikasi CART, sebagaimana ditampilkan pada Gambar 2 berikut.



Gambar 2. *Feature Importance Decision Tree* CART

Berdasarkan Gambar 2, atribut perimeter merupakan atribut paling dominan. Tingginya kontribusi atribut perimeter menunjukkan bahwa ukuran keliling kismis merupakan pembeda utama antara varietas besni dan kecimen. Gambar 3 berikut merangkum hasil prediksi model CART pada 180 data uji dalam bentuk confusion matrix.



Gambar 3. *Confusion Matrix Decision Tree* CART

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

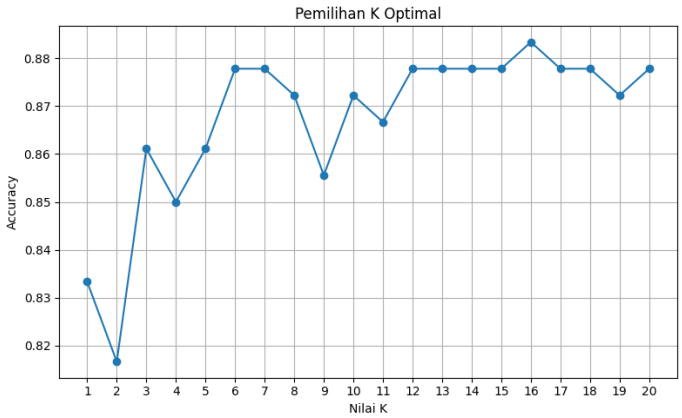
Berdasarkan Gambar 3, model CART berhasil mengklasifikasikan 70 data besni dan 82 data kecimen secara benar. Terdapat 20 data besni yang salah diklasifikasikan sebagai kecimen (*false positive*) dan 8 data kecimen yang salah diklasifikasikan sebagai besni (*false negative*). Hasil evaluasi lengkap model CART disajikan pada tabel 4 berikut.

Tabel 4. Hasil Evaluasi *Decision Tree* CART

Metrik	Nilai
Akurasi	0.844444
Presisi	0.803922
Recall	0.911111
F1-Score	0.854167

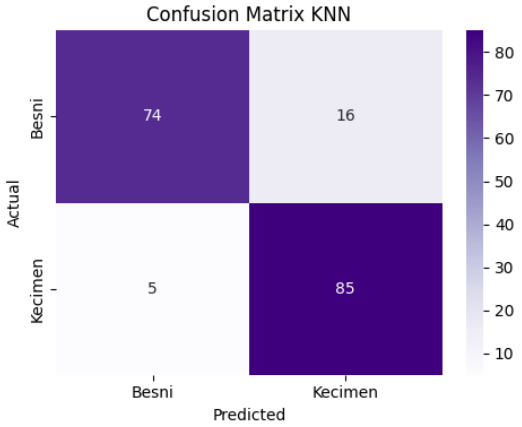
Pemodelan dan Evaluasi *K-Nearest Neighbors*

Sebelum membangun model KNN, dilakukan pengujian nilai K dari 1 hingga 20 untuk menentukan K yang menghasilkan akurasi tertinggi. Grafik hasil pengujian nilai K pada Gambar 4 berikut.



Gambar 4. Grafik Pemilihan K Optimal

Berdasarkan Gambar 4, nilai K=16 menghasilkan akurasi tertinggi sebesar 0,8833 atau 88,33% sehingga ditetapkan sebagai K optimal. Model KNN kemudian dibangun menggunakan K=16. Berbeda dengan CART, KNN menunjukkan *train accuracy* sebesar 86,11% dan test accuracy sebesar 88,33%. Nilai *test accuracy* yang lebih tinggi dari *train accuracy* menunjukkan bahwa model KNN tidak mengalami *overfitting* dan mampu menggeneralisasi dengan baik pada data baru. Gambar 5 berikut merangkum hasil prediksi model KNN pada 180 data uji dalam bentuk *confusion matrix.*



Gambar 5. *Confusion Matrix K-Nearest Neighbors* (K=16)

Berdasarkan Gambar 5, model KNN berhasil mengklasifikasikan 74 data besni dan 85 data kecimen secara benar. Terdapat 16 data besni yang salah diklasifikasikan sebagai kecimen dan 5 data kecimen yang salah

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

diklasifikasikan sebagai besni. Dibandingkan CART, KNN menghasilkan jumlah kesalahan prediksi lebih sedikit pada kedua kelas. Hasil evaluasi lengkap model KNN disajikan pada Tabel 5 berikut.

Tabel 5. Hasil Evaluasi *K-Nearest Neighbors* (K=16)

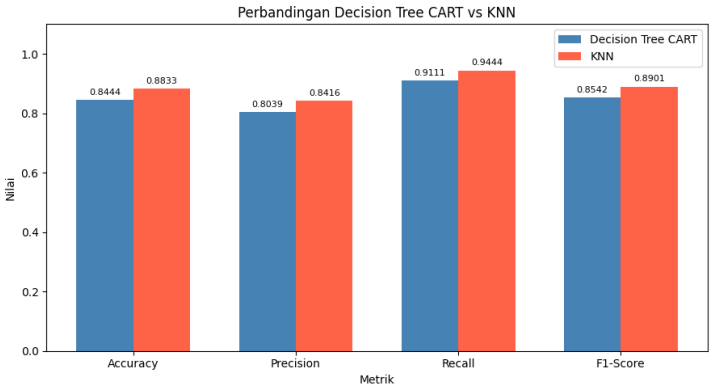
Metrik	Nilai
Akurasi	0.883333
Presisi	0.841584
Recall	0.944444
F1-Score	0.890052

Perbandingan *Decision Tree* CART dan KKN

Perbandingan kinerja kedua model secara keseluruhan disajikan pada tabel 6 berikut, dengan visualisasi perbandingan ditampilkan pada gambar 6.

Tabel 6. Perbandingan Hasil Evaluasi *Decision Tree* CART dan KNN

Metrik	Decision Tree CART	KNN(K=16)
Akurasi	0.844444	0.883333
Presisi	0.803922	0.841584
Recall	0.911111	0.944444
F1-Score	0.854167	0.890052



Gambar 6. Grafik Perbandingan *Decision Tree* CART vs KNN

Berdasarkan Tabel 6 dan Gambar 6, KNN dengan K=16 mengungguli decision tree CART pada seluruh metrik evaluasi. KKN mencapai akurasi 88,33% lebih tinggi 3,89 % dibandingkan CART sebesar 84,44%. Dari sisi presisi, KNN menghasilkan 84,16% berbanding 80,39% pada CART, menunjukkan bahwa KNN lebih sedikit menghasilkan prediksi positif yang keliru. Nilai *recall* KNN sebesar 94,44% lebih tinggi dari CART sebesar 91,11%, yang berarti KNN lebih baik dalam mendeteksi seluruh data kelas kecimen yang sebenarnya. F1-*score* KNN sebesar 89,01% dibandingkan 85,42% pada CART mengkonfirmasi keunggulan KNN secara keseluruhan dalam menyeimbangkan presisi dan *recall*.

Keunggulan KNN pada *dataset* ini dapat dijelaskan oleh beberapa faktor. Pertama, distribusi kelas yang seimbang (450:450) sangat mendukung kinerja KNN karena algoritma ini sensitif terhadap ketidakseimbangan kelas. Kedua, standarisasi fitur yang dilakukan sebelum pemodelan memaksimalkan kemampuan KNN dalam menghitung jarak *euclidean* secara akurat. Ketiga, nilai K=16 yang cukup besar membantu meredam pengaruh *noise* pada data sehingga menghasilkan batas keputusan yang lebih stabil. Sebaliknya, *decision tree* CART menunjukkan gejala *overfitting* yang ditandai dengan *train accuracy* 100% namun *test accuracy* hanya 84,44% yang berarti pohon yang terbentuk terlalu kompleks untuk data latih namun kurang mampu menggeneralisasikan pada data uji.

Selain faktor-faktor tersebut, karakteristik data juga berpengaruh terhadap keunggulan KNN dibandingkan CART. Raisin *dataset* terdiri atas tujuh fitur morfologi kontinu yang telah distandarisasi menggunakan

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

StandardScaler, sehingga hubungan kedekatan antar objek dapat direpresentasikan dengan lebih baik melalui perhitungan jarak *euclidean*. KNN memanfaatkan informasi dari beberapa tetangga terdekat (K=16) sehingga keputusan klasifikasi tidak bergantung pada satu batas pemisah tertentu, melainkan pada pola lokal di sekitar data uji. Sebaliknya, CART membentuk aturan klasifikasi berdasarkan pemisahan biner pada setiap node. Apabila terdapat variasi nilai fitur yang tinggi atau batas antar kelas saling tumpang tindih, struktur pohon yang terbentuk cenderung semakin kompleks untuk menyesuaikan data latih. Kondisi tersebut menyebabkan model memiliki kemampuan generalisasi yang lebih rendah dibandingkan KNN, sebagaimana ditunjukkan oleh perbedaan akurasi data latih dan data uji. Dengan demikian, pada dataset ini KNN lebih mampu menangkap pola kemiripan antar sampel, sedangkan CART lebih rentan membentuk batas keputusan yang terlalu spesifik terhadap data pelatihan. Temuan ini juga sejalan dengan penelitian [5] yang menunjukkan bahwa KNN memberikan performa yang baik pada klasifikasi raisin *dataset* karena karakteristik fitur morfologi lebih mudah dibedakan menggunakan pendekatan berbasis jarak daripada aturan pemisahan yang kaku pada pohon keputusan.

Meskipun demikian, performa *decision tree* CART masih berpotensi ditingkatkan melalui optimasi model. *Overfitting* yang teridentifikasi pada penelitian ini menunjukkan bahwa struktur pohon yang terbentuk terlalu kompleks sehingga mampu menghafal pola pada data latih, tetapi kurang mampu menggeneralisasi data baru. Kondisi tersebut dapat diminimalkan dengan menerapkan teknik *pre-pruning*, misalnya melalui pengaturan parameter *max_depth*, *min_samples_split*, dan *min_samples_leaf*, atau *post-pruning* untuk memangkas cabang yang tidak memberikan peningkatan performa yang signifikan. Selain itu, optimasi *hyperparameter* menggunakan *grid search* atau *random search* yang dipadukan dengan *cross-validation* juga berpotensi menghasilkan model CART yang lebih sederhana dengan kemampuan generalisasi yang lebih baik.

Perbandingan dengan penelitian terdahulu menunjukkan bahwa performa model pada penelitian ini kompetitif dengan studi sejenis. [1] yang mengklasifikasikan varietas kismis menggunakan lima algoritma data *mining* melaporkan akurasi tertinggi sebesar 86,81% dengan *neural network*, sementara [5] yang menerapkan KNN dengan metrik *euclidean* pada *dataset* kismis yang sama memperoleh akurasi tertinggi sekitar 85% pada k=7 dan k=9. Model KNN pada penelitian ini menghasilkan akurasi 88,33% pada k=16, sedikit lebih tinggi dibandingkan kedua penelitian tersebut. Peningkatan ini kemungkinan disebabkan oleh proses praproses data yang lebih matang, khususnya penanganan *outlier* menggunakan *capping* IQR dan standarisasi fitur sebelum pemodelan, sehingga *noise* pada data berkurang dan estimasi jarak antar-tetangga menjadi lebih akurat. Perbandingan ini menegaskan bahwa kombinasi praproses data yang tepat dengan pemilihan nilai K optimal mampu menghasilkan performa yang setara bahkan lebih baik dibandingkan pendekatan lain pada permasalahan klasifikasi kismis.

Meskipun demikian, penelitian ini memiliki beberapa limitasi. Pertama, *dataset* yang digunakan berjumlah 900 data dan hanya mencakup dua varietas kismis (besni dan kecimen), sehingga generalisasi model terhadap varietas lain atau ukuran dataset yang lebih besar belum dapat dipastikan. Kedua, evaluasi model hanya dilakukan dengan skema pembagian data latih-uji sederhana tanpa validasi silang (*cross-validation*), sehingga stabilitas performa model pada berbagai pembagian data belum teruji sepenuhnya. Ketiga, *decision tree* CART pada penelitian ini menunjukkan gejala *overfitting* (*train accuracy* 100% vs *test accuracy* 84,44%) sehingga performanya belum optimal; penerapan teknik *pruning* atau *tuning hyperparameter* seperti *max_depth* dan *min_samples_leaf* berpotensi memperbaiki hasil evaluasinya. Limitasi ini menjadi peluang bagi penelitian selanjutnya untuk memperluas *dataset*, menerapkan validasi silang, dan mengeksplorasi optimasi *hyperparameter* secara lebih sistematis.

4. SIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan metode *decision tree* CART dengan parameter criterion'gini' menghasilkan akurasi 84,44%, presisi 80,39%, *recall* 91,11%, dan *f1-score* 85,42% dalam mengklasifikasikan varietas kismis besni dan kecimen pada raisin dataset, namun model ini menunjukkan gejala *overfitting* yang ditandai dengan *train accuracy* 100% sementara *test accuracy* hanya 84,44%. Sebaliknya metode *k-nearest neighbors* dengan nilai K optimal K=16 menghasilkan akurasi 88,33%, presisi 84,16%, *recall* 94,44%, dan *f1-score* 89,01%, tanpa menunjukkan gejala *overfitting* dengan *train accuracy* 86,11% dan *test accuracy* 88,33% relatif seimbang.

Berdasarkan perbandingan keempat metrik evaluasi tersebut, KNN terbukti lebih unggul dibandingkan *decision tree* CART dengan selisih akurasi 3,89%, presisi 3,77%, *recall* 3,33%, dan *f1-score* 3,59%, sehingga KNN direkomendasikan sebagai metode yang lebih sesuai untuk klasifikasi kualitas kismis pada *dataset* dengan karakteristik distribusi kelas seimbang dan fitur morfologi seperti raisin *dataset*. Penelitian ini masih memiliki beberapa limitasi, di antaranya jumlah data yang terbatas (900 data dengan dua varietas kismis), evaluasi model yang belum menggunakan validasi silang (*cross-validation*), serta belum diterapkannya teknik *pruning* atau *tuning hyperparameter* pada model CART untuk mengurangi *overfitting*. Untuk penelitian selanjutnya, disarankan memperluas ukuran dan varietas *dataset* (mencakup varietas kismis lain atau sumber data dari wilayah berbeda), menerapkan skema validasi silang untuk menguji stabilitas model secara lebih menyeluruh, mengeksplorasi teknik optimasi *hyperparameter* (misalnya *grid search* atau *random search*) pada CART dan KNN, serta mencoba algoritma klasifikasi lain seperti *random forest* atau *support vector machine* sebagai pembanding tambahan terhadap SOTA.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

DAFTAR PUSTAKA

[1] R. A. Anggraini, Ratningsih, Y. Apriyani, M. W. Pertiwi, M. Kusmira, dan S. Bahri, "Klasifikasi Jenis Kismis Menggunakan Teknik Data Mining," *Jurnal Kajian Ilmiah*, vol. 24, no. 1, pp. 45–56, Jan 2024, doi: 10.31599/ryvqk945.

[2] "Statistical-Yearbook-2022-2023.pdf." Diakses: 16 Agustus 2026. [Daring]. Tersedia pada: https://inc.nutfruit.org/wp-content/uploads/2023/05/Statistical-Yearbook-2022-2023.pdf.

[3] L. N. Aziza, R. Y. Astuti, B. A. Maulana, dan N. Hidayati, "Penerapan Algoritma K-Nearest Neighbor untuk Klasifikasi Ketahanan Pangan di Provinsi Jawa Tengah: Application of The K-Nearest Neighbor Algorithm for Food Security Classification In Central Java Province," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 404–412, Feb 2024, doi: 10.57152/malcom.v4i2.1201.

[4] A. Tangkelayuk, "The Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes, dan Decision Tree | JATISI," Jun 2022, Diakses: 9 Juni 2026. [Daring]. Tersedia pada: https://jurnal.mdp.ac.id/index.php/jatisi/article/view/2048.

[5] M. A. Aprihartha, T. N. Alam, dan M. Husniyadi, "Perbandingan Metrik Euclidean dan Metrik Manhattan untuk K-Nearest Neighbors dalam Klasifikasi Kismis," *Jurnal Ilmu Komputer dan Informatika*, vol. 4, no. 1, pp. 21–30, Jun 2024, doi: 10.54082/jiki.126.

[6] A. S. Biyantoro dan B. Prasetyo, "Penerapan Decision Tree untuk Klasifikasi Status Kesehatan dengan perbandingan KNN dan Naive Bayes: Application of Decision Tree for Health Status Classification, Compared to KNN and Naive Bayes," *Indonesian Journal of Informatic Research and Software Engineering (IJIRSE)*, vol. 4, no. 1, pp. 47–55, Mar 2024, doi: 10.57152/ijirse.v4i1.1342.

[7] "Raisin Dataset." Diakses: 6 Agustus 2026. [Daring]. Tersedia pada: https://www.kaggle.com/datasets/muratkokludataset/raisin-dataset.

[8] T. Gori, A. Sunyoto, dan H. A. Fatta, "Preprocessing Data dan Klasifikasi untuk Prediksi Kinerja Akademik Siswa," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 1, pp. 215–224, Feb 2024, doi: 10.25126/jtiik.20241118074.

[9] C. Herdian, A. Kamila, dan I. G. A. M. Budidarma, "Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi," *Technologia : Jurnal Ilmiah*, vol. 15, no. 1, pp. 93–108, Jan 2024, doi: 10.31602/tji.v15i1.13457.

[10] W. Loh, "Classification and regression trees," *WIREs Data Min & Knowl*, vol. 1, no. 1, pp. 14–23, Jan 2011, doi: 10.1002/widm.8.

[11] A. Abedinia dan V. Seydi, "Building semi-supervised decision trees with semi-cart algorithm," *Int. J. Mach. Learn. & Cyber.*, vol. 15, no. 10, pp. 4493–4510, Okt 2024, doi: 10.1007/s13042-024-02161-z.

[12] E. Halabaku dan E. Bytyçi, "Overfitting in Machine Learning: A Comparative Analysis of Decision Trees and Random Forests," *IASC*, vol. 39, no. 6, pp. 987–1006, 2024, doi: 10.32604/iasc.2024.059429.

[13] I. Riadi, R. Umar, dan R. Anggara, "Prediksi Kelulusan Tepat Waktu Berdasarkan Riwayat Akademik Menggunakan Metode K-Nearest Neighbor," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 2, pp. 249–256, Agu 2024, doi: 10.25126/jtiik.20241127330.

[14] A. Razaki, Y. H. Chrisnanto, dan M. Melina, "Penanganan Outlier Pada Metode Algoritma K- Nearest Neighbors (KNN) Dengan Metode Kernel Density Estimation Pada Kasus Penyakit Diabetes," *INTECOMS: Journal of Information Technology and Computer Science*, vol. 7, no. 4, pp. 1177–1188, Jul 2024, doi: 10.31539/intecoms.v7i4.10866.

[15] R. K. Halder, M. N. Uddin, Md. A. Uddin, S. Aryal, dan A. Khraisat, "Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications," *J Big Data*, vol. 11, no. 1, pp. 113, Agu 2024, doi: 10.1186/s40537-024-00973-y.

[16] L. Kurniawati, D. Priyanto, N. S. Ningsih, M. Syahrir, dan R. Rismayati, "Perbandingan Metode Berbasis Decision Tree dalam Deteksi Penyakit Paru-Paru," *BITe*, vol. 7, no. 1, pp. 51–62, Jun 2025, doi: 10.30812/bite.v7i1.4909.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi