JURNAL DIGITAL TEKNOLOGI INFORMASI
Vol. 07, No.02, September 2024, pp. 14-20
E-ISSN 2714-9706 | P-ISSN 2686-4185
Terakreditas Sinta 5 oleh Dirjen Pendidikan Tinggi, Riset, dan Teknologi No. 164/E/KPT/2021

Prediksi penyakit jantung berbasis mesin learning dengan menggunakan metode k-nn

Lukman Hakim [a,1,*]; Ahmad Sobri [a,2]; Lukman Sunardi [a,3]; Deni Nurdiansyah [a,4]
[a] *Universitas Bina Insan, Jl HM Soeharto KM. 13 Kel. Lubuk Kupang, Kota Lubuklinggau Sumatera Selatan 31626, Indonesia*
[1] *lukman_hakim@univbinainsan.ac.id;* [2] *ahmadsobri506@gmail.com;* [3] *lukmansunardi@univbinainsan.ac.id;*
[4] *deninurdiansyah@univbinainsan.ac.id*
** Corresponding author*

Artikel Histori: Diterima 03/02/2025; Revisi 05/02/2025; Terbit 06/02/2025

Abstrak

Permasalahan dalam penelitian ini adalah belum adanya sistem prediksi penyakit jantung yang dapat memudahkan proses penanganan dan identifikasi gejala penyakit jantung. Oleh karena itu, dilakukan proses ekstraksi pola gejala penyakit jantung berdasarkan data medis menggunakan model K-Nearest Neighbor (KNN). Proses pengolahan data penyakit jantung dengan algoritma KNN ini bertujuan untuk menganalisis data penyakit jantung berbasis machine learning. Dalam prediksi penyakit jantung, diperlukan sistem yang dapat membantu mengelola data terkait pengolahan citra medis. Tujuan dari penelitian ini adalah untuk membantu melakukan klasifikasi jenis penyakit jantung, sehingga memudahkan proses penanganan medis. Analisis dilakukan menggunakan sistem machine learning dengan tools Python, di mana dalam pengolahan data terdapat tahapan yang berhubungan dengan pembagian data (splitting data) untuk menghasilkan akurasi yang optimal.

Kata Kunci: Penyakit Jantung, Machine Learning, KNN

Pendahuluan

Penyakit jantung merupakan salah satu penyakit yang sangat ditakuti oleh masyarakat luas karena dapat menyebabkan kematian, dan dalam menjalankan terapi penyakit jantung akan memakan waktu yang lama. Selain itu, penyakit jantung juga dipengaruhi oleh faktor keturunan dan pola makan yang dapat meningkatkan pergerakan sel darah yang berhubungan dengan aktivitas organ tubuh lainnya [1].

Data mining adalah proses analisis data untuk menemukan pola atau informasi tersembunyi yang tidak terduga, serta merangkum data dengan cara atau metode baru yang dapat dimengerti dan bermanfaat bagi pemilik data. Dalam konteks ini, data mining berperan penting dalam knowledge discovery in databases (KDD), yang melibatkan teknik-teknik untuk menggali informasi tersembunyi dalam jumlah besar dan kompleks, sehingga menghasilkan output berupa karakteristik atau pola dari data tersebut [2].

Implementasi bermuara pada aktivitas atau kegiatan pencapaian tujuan, adanya hasil kegiatan, hasil sebagai produk, dan hasil dari akibat. Ungkapan mekanisme mengandung arti bahwa implementasi bukan sekedar aktivitas tetapi suatu kegiatan yang terencana dan dilakukan secara sungguh-sungguh berdasarkan acuan norma tertentu untuk mencapai tujuan kegiatan [3].

Model machine learning KNN ini memungkinkan mesin untuk melakukan pembelajaran terhadap kumpulan citra digital yang disebut dengan dataset. KNN sendiri merupakan algoritma dalam pengolahan citra yang digunakan sebagai pengolah data citra digital 2 dimensi [4]. Secara fungsional, KNN bekerja dengan mengklasifikasikan data berdasarkan kedekatannya dengan data lain dalam ruang fitur, tanpa melibatkan bobot, bias, atau fungsi aktivasi seperti pada jaringan saraf tiruan.

Python merupakan bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, software/aplikasi, mengotomatiskan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Artinya, ia bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja [5].

Untuk menerapkan metode data mining ini, penulis menggunakan model K-Nearest Neighbors (KNN). Metode KNN merupakan algoritma klasifikasi yang bekerja dengan mengklasifikasikan data berdasarkan kedekatannya dengan data lain dalam ruang fitur. Algoritma ini termasuk dalam supervised learning, di mana hasil query instance yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam data latih. Setelah mengumpulkan KNN, kemudian diambil mayoritas dari KNN untuk dijadikan prediksi dari sample uji [6].

https://doi.org/10.32502/digital.v7i2.9429

Metode Penelitian

Metode penelitian merupakan serangkaian kegiatan dalam mencari kebenaran suatu studi penelitian, yang diawali dengan suatu pemikiran yang membentuk rumusan masalah sehingga menimbulkan hipotesis awal, dengan dibantu dan persepsi penelitian terdahulu, sehingga penelitian bisa diolah dan dianalisis yang akhirnya membentuk suatu kesimpulan. Metode penelitian atau ilmiah merupakan langkah dalam mendapatkan pengetahuan ilmiah. Metode penelitian merupakan suatu penyelidikan terstruktur dan kritis dalam mengungkap fakta [7].

a. Metode Pengujian dan Pengolahan Data

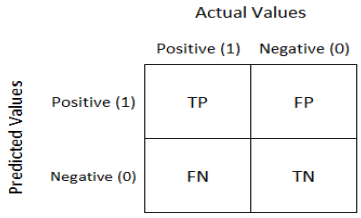
1. Metode Pengujian

(a) Confusion Matrix

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi kinerja suatu model klasifikasi. Ini membandingkan hasil prediksi model dengan nilai sebenarnya dalam dataset pengujian. Confusion matrix memiliki 4 sel utama [8]:

(1) True Positive (TP) : Kasus di mana model memprediksi kelas positif dengan benar.
(2) True Negative (TN) : Kasus di mana model memprediksi kelas negatif dengan benar.
(3) False Positive (FP) : Kasus di mana model memprediksi kelas positif padahal sebenarnya negatif.
(4) False Negative (FN) : Kasus di mana model memprediksi kelas negatif padahal sebenarnya positif.

Dari confusion matrix, kita dapat menghitung berbagai metrik evaluasi seperti akurasi, presisi, recall, dan lainnya [9].



Gambar 1. Confusion Matrix

(1) Akurasi (Accuracy) : Akurasi adalah proporsi total prediksi yang benar (baik positif maupun negatif) dibandingkan dengan total jumlah data. Secara matematis, akurasi dihitung dengan rumus:

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

(2) Presisi (Precision) : Presisi adalah proporsi dari prediksi positif yang benar terhadap total jumlah prediksi positif. Ini mengukur seberapa banyak dari hasil yang diprediksi sebagai positif yang sebenarnya benar positif. Secara matematis, presisi dihitung dengan rumus:

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2)$$

(3) Recall (Sensitivity atau True Positive Rate):Recall adalah proporsi dari kelas positif yang diprediksi dengan benar dibandingkan dengan total jumlah kelas positif yang sebenarnya. Ini mengukur seberapa baik model mengidentifikasi semua kasus positif. Secara matematis, recall dihitung dengan rumus:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

(4) F-1 Score : F-1 score adalah rata-rata harmonik dari presisi dan recall. Ini memberikan keseimbangan antara kedua metrik ini. F-1 score berguna ketika kelas memiliki distribusi yang tidak seimbang. Secara matematis, F-1 score dihitung dengan rumus:

$$\text{F-1 Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (4)$$

P-ISSN : 2686-4185
E-ISSN : 2714-9706

b. Pengolahan Data

1. Persiapan Data

Data yang digunakan adalah data pasien penyakit jantung yang melakukan pengobatan dengan jumlah sebanyak 300 orang. Data ini merupakan data mentah yang sudah berlabel. Kemudian data tersebut disimpan dalam format .csv untuk dapat diolah dalam proses berikutnya. Adapun fitur-fitur yang digunakan dalam menentukan seseorang berpotensi penyakit jantung adalah: umur, anamia (kurang darah), kadar creatinine phosphokinase, potensi diabetes, fraksi ejeksi, tekanan darah tinggi, kadar trombosit, serum creatinine, serum sodium, jenis kelamin, perokok, dan kejadian meninggal.

2. Pemrosesan Awal [10]

Pemrosesan awal digunakan untuk bagaimana data bisa diproses sehingga dapat dilakukan klasifikasi dengan menggunakan model pembelajaran mesin. Agar dapat diproses maka diperlukan pelabelan dari data tersebut. pelabelan berguna untuk menentukan prediksi yang akan dipakai dalam proses mendeteksi penyakit jantung [11].

Hasil dan Pembahasan

a. Hasil

Mengimport library

Library yang digunakan dalam kode ini mencakup berbagai fungsi yang sangat penting untuk analisis data dan pembuatan model machine learning. NumPy dan Pandas digunakan untuk manipulasi data, Matplotlib dan Seaborn digunakan untuk visualisasi data, sedangkan Scikit-learn digunakan untuk pembuatan model machine learning dan evaluasi kinerjanya. Handling warnings juga dilakukan untuk mengabaikan peringatan yang tidak perlu. Gambar 2 berikut menunjukkan library yang digunakan.

```
# Importing necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score

import warnings

# Handling warnings
warnings.filterwarnings('ignore')
```

Gambar 2. Import library

Setelah melakukan import library, langkah selanjutnya adalah melakukan pembacaan dataset. Fungsi pd.read_csv() digunakan untuk membaca data dari file CSV. Parameter yang diberikan adalah nama file CSV yang ingin dibaca, yaitu 'dataset_penyakit_jantung.csv'. Hasil pembacaan data ini akan disimpan dalam sebuah Data Frame yang dinamai df. Gambar menyajikan cara melakukan pembacaan terhadap dataset penyakit jantung.

```
# Load Data
df = pd.read_csv('dataset_penyakit_jantung.csv')
```

```
#data exploration and cleaning
df.head(10)
```

	umur	anaemia	kadar_creatinine_phosphokinase	potensi_diabetes	fraksi_ejeksi	tekanan_darah_tinggi	kadar_trombosit	serum_creatinine	serum_sodium	jenis_kelami
0	75.0	0	582	0	20	1	265000.00	1.9	130	
1	55.0	0	7861	0	38	0	263358.03	1.1	136	
2	65.0	0	146	0	20	0	162000.00	1.3	129	
3	50.0	1	111	0	20	0	210000.00	1.9	137	
4	65.0	1	160	1	20	0	327000.00	2.7	116	
5	90.0	1	47	0	40	1	204000.00	2.1	132	
6	75.0	1	246	0	15	0	127000.00	1.2	137	
7	60.0	1	315	1	60	0	454000.00	1.1	131	
8	65.0	0	157	0	65	0	263358.03	1.5	138	
9	80.0	1	123	0	35	1	388000.00	9.4	133	

Gambar 3. Loading Dataset

Splitting Dataset

Data splitting adalah proses membagi dataset menjadi tiga bagian utama: training set, validation set, dan testing set. Ini dilakukan untuk memastikan bahwa model pembelajaran mesin dipelajari dengan baik

pada data pelatihan, diuji dengan data validasi untuk menyesuaikan parameter model, dan diukur kinerjanya dengan data tes. Gambar 4 menyajikan proses membagi dataset.

```
# Split the dataset into train, validation, and test sets
X = df.drop('meninggal', axis=1)
y = df['meninggal']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=7)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.05, random_state=7)

# Standardize the features
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_val = sc.transform(X_val)
X_test = sc.transform(X_test)

# Print the shapes of the resulting datasets
print("Training set:", X_train.shape, y_train.shape)
print("Validation set:", X_val.shape, y_val.shape)
print("Test set:", X_test.shape, y_test.shape)
```

```
Training set: (255, 12) (255,)
Validation set: (14, 12) (14,)
Test set: (30, 12) (30,)
```

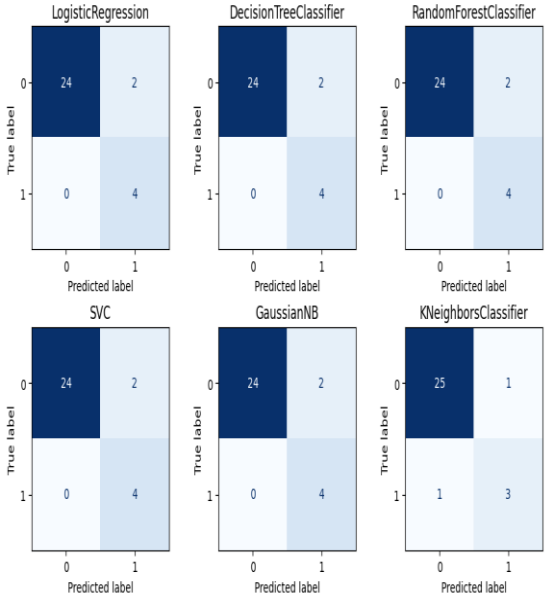
Gambar 4. Proses Splitting Dataset

Membagi Dataset: Fungsi train_test_split digunakan untuk membagi dataset menjadi set pelatihan (X_train, y_train) dan set tes (X_test, y_test). Parameter test_size=0.1 menentukan bahwa 10% dari dataset akan digunakan sebagai set tes, sedangkan 90% akan digunakan sebagai set pelatihan.

Membagi Set Pelatihan: Kemudian, set pelatihan dibagi lagi menjadi set pelatihan (X_train, y_train) dan set validasi (X_val, y_val). Parameter test_size=0.05 menentukan bahwa 5% dari set pelatihan akan digunakan sebagai set validasi, sedangkan 95% akan tetap sebagai set pelatihan.

Evaluasi Model

Gambar dibawah ini menyajikan hasil evaluasi dengan menggunakan confusion matrix.



Gambar 5. Proses Klasifikasi Dengan CM Pada Masing-Masing Algoritma

```
Classification Report for LogisticRegression:
              precision    recall  f1-score   support

           0       1.00      0.92      0.96        26
           1       0.67      1.00      0.80         4

    accuracy                           0.93        30
   macro avg       0.83      0.96      0.88        30
weighted avg       0.96      0.93      0.94        30
```

Gambar 6. Classification report untuk logistic regression

P-ISSN : 2686-4185
E-ISSN : 2714-9706

Classification Report for KNeighborsClassifier:

	precision	recall	f1-score	support
0	0.96	0.96	0.96	26
1	0.75	0.75	0.75	4
accuracy			0.93	30
macro avg	0.86	0.86	0.86	30
weighted avg	0.93	0.93	0.93	30

Gambar 7. Classification report untuk KNN

b. Pembahasan

Dari hasil yang telah dijabarkan, dapat dilihat bahwa KNN dapat memprediksi dengan baik dan memilah barang yang ‘favorit’ maupun yang ‘tidak favorit’. Berikut beberapa hal yang dapat dijabarkan:

(1) Precision:

Precision untuk kelas "favorit" adalah 1.00, yang berarti dari semua sampel yang diprediksi sebagai "favorit", semuanya benar-benar "favorit".Precision untuk kelas "tidak_favorit" adalah 0.88, yang berarti dari semua sampel yang diprediksi sebagai "tidak_favorit", 88% benar-benar "tidak_favorit".

(2) Recall:

Recall untuk kelas "favorit" adalah 0.67, yang berarti 67% dari semua sampel "favorit" berhasil diprediksi dengan benar oleh model.Recall untuk kelas "tidak_favorit" adalah 1.00, yang berarti 100% dari semua sampel "tidak_favorit" berhasil diprediksi dengan benar oleh model.

(3) F1-score:

F1-score untuk kelas "favorit" adalah 0.80, yang merupakan harmonic mean dari precision dan recall untuk kelas "favorit". Ini menunjukkan keseimbangan antara precision dan recall untuk kelas "favorit".

F1-score untuk kelas "tidak_favorit" adalah 0.93, yang merupakan harmonic mean dari precision dan recall untuk kelas "tidak_favorit". Ini menunjukkan keseimbangan antara precision dan recall untuk kelas "tidak_favorit".

(4) Accuracy:

Akurasi model adalah 0.90, yang menunjukkan bahwa model berhasil memprediksi dengan benar 90% dari semua sampel dalam data uji.

(5) Macro Average:

Macro average dari precision, recall, dan F1-score adalah rata-rata dari metrik-metrik tersebut dihitung untuk setiap kelas secara terpisah, tanpa mempertimbangkan ketidakseimbangan kelas. Nilainya adalah 0.94 untuk precision, 0.83 untuk recall, dan 0.87 untuk F1-score.

(6) Weighted Average:

Weighted average dari precision, recall, dan F1-score adalah rata-rata dari metrik-metrik tersebut dihitung untuk setiap kelas, dengan bobot yang mempertimbangkan jumlah sampel dalam setiap kelas. Ini memberikan gambaran tentang kinerja keseluruhan model dengan memperhitungkan ketidakseimbangan kelas. Nilainya adalah 0.91 untuk precision, 0.90 untuk recall, dan 0.89 untuk F1-score.

Dengan melihat classification report ini, kita dapat memahami secara lebih mendalam bagaimana model klasifikasi melakukan prediksi terhadap setiap kelas dan mengevaluasi kinerjanya secara menyeluruh

Berdasarkan hasil yang telah dijelaskan, model KNN menunjukkan kemampuan yang baik dalam memprediksi dan mengklasifikasikan barang sebagai 'favorit' atau 'tidak favorit'. Berikut adalah analisis metrik evaluasi model:

1. Precision:
   a. Kelas 'favorit': Precision sebesar 1.00 menunjukkan bahwa semua sampel yang diprediksi sebagai 'favorit' benar-benar termasuk dalam kategori tersebut.
   b. Kelas 'tidak_favorit': Precision sebesar 0.88 berarti 88% dari sampel yang diprediksi sebagai 'tidak_favorit' benar-benar termasuk dalam kategori tersebut.
2. Recall:
   a. Kelas 'favorit': Recall sebesar 0.67 menunjukkan bahwa 67% dari semua sampel 'favorit' berhasil diprediksi dengan benar oleh model.
   b. Kelas 'tidak_favorit': Recall sebesar 1.00 berarti 100% dari semua sampel 'tidak_favorit' berhasil diprediksi dengan benar oleh model.

3. F1-Score:
   a. Kelas 'favorit': F1-Score sebesar 0.80 menunjukkan keseimbangan antara precision dan recall untuk kelas 'favorit'.
   b. Kelas 'tidak_favorit': F1-Score sebesar 0.93 menunjukkan keseimbangan antara precision dan recall untuk kelas 'tidak_favorit'.
4. Akurasi:

   Akurasi model sebesar 0.90 menunjukkan bahwa model berhasil memprediksi dengan benar 90% dari semua sampel dalam data uji.
5. Macro Average:
   a. Precision: 0.94
   b. Recall: 0.83
   c. F1-Score: 0.87

   Macro Average menghitung rata-rata metrik untuk setiap kelas secara terpisah tanpa mempertimbangkan ketidakseimbangan kelas.
6. Weighted Average:
   a. Precision: 0.91
   b. Recall: 0.90
   c. F1-Score: 0.89

   Weighted Average menghitung rata-rata metrik untuk setiap kelas dengan mempertimbangkan jumlah sampel dalam setiap kelas, memberikan gambaran tentang kinerja keseluruhan model dengan memperhitungkan ketidakseimbangan kelas.

Dengan menganalisis classification report ini, kita dapat memahami secara lebih mendalam bagaimana model klasifikasi melakukan prediksi terhadap setiap kelas dan mengevaluasi kinerjanya secara menyeluruh.

Simpulan

Semua model klasifikasi menunjukkan akurasi yang tinggi baik pada validasi maupun tes, dengan model SVC dan KNeighbors Classifier memiliki nilai akurasi tertinggi, yaitu 0,93. Meskipun demikian, presisi dan recall dapat bervariasi tergantung pada data dan kelas yang digunakan. Oleh karena itu, pemilihan model yang tepat harus mempertimbangkan tujuan analisis dan evaluasi.

1. Model SVC: Dengan akurasi tertinggi pada kedua dataset (validasi dan tes), model SVC dapat menjadi pilihan utama untuk aplikasi yang memprioritaskan akurasi tinggi. SVC unggul dalam memberikan prediksi yang lebih tepat secara keseluruhan.
2. Model KNeighbors Classifier: Meskipun memiliki F1-score sedikit lebih rendah dibandingkan SVC, KNeighbors Classifier menunjukkan presisi dan recall yang lebih baik pada data tes. Ini menjadikannya pilihan alternatif yang baik, terutama jika presisi dan recall lebih penting daripada hanya akurasi. Model ini lebih handal dalam memprediksi kelas-kelas tertentu dengan ketepatan tinggi.

Secara keseluruhan, keputusan pemilihan model bergantung pada prioritas analisis: apakah lebih mengutamakan akurasi keseluruhan atau keseimbangan antara presisi dan recall dalam prediksi setiap kelas.

Daftar Pustaka

[1] I. Sumadikarta and L. Andrayani, “Implementasi Data Mining Untuk Clustering Makanan dan Minuman Favorit Dengan Menggunakan Algortima K-Means,” J. Ilm. Fak. Tek. LIMIT’S, vol. 15, no. 1, pp. 40–49, 2019.

[2] Y. R. Amalia, “Penerapan Data Mining Untuk Prediksi Penjualan Produk Elektronik Terlaris Menggunakan Metode K-Nearest Neighbor (Studi Kasus : PT.Bintang Multi Sarana Palembang),” Skripsi, pp. i–90, 2018, [Online]. Available: http://eprints.radenfatah.ac.id/id/eprint/3302%0A

[3] A. Muhajir Haris and E. Priyo Purnomo, “Implementasi Csr (Corporate Social Responsibility) Pt. Agung Perdana Dalam Mengurangi Dampak Kerusakan Lingkungan,” J. Gov. Public Policy, vol. 3, no. 2, pp. 203–225, 2016, doi: 10.18196/jgpp.2016.0056.

[4] I. P. Putri, “Analisis Performa Metode K- Nearest Neighbor (KNN) dan Crossvalidation pada Data Penyakit Cardiovascular,” Indones. J. Data Sci., vol. 2, no. 1, pp. 21–28, 2021, doi: 10.33096/ijodas.v2i1.25.

[5] W. Lestari, F. Fatoni, and H. Hutrianto, “Implementasi Data Mining Untuk Kartu Indonesia Sehat Bagi Masyarakat Kurang Mampu Menggunakan Metode Clustering Pada Dinas Sosial Kota Palembang,” J. Nas. Ilmu Komput., vol. 1, no. 4, pp. 169–174, 2020, doi: 10.47747/jurnalnik.v1i4.163.

[6] D. Cahyanti, A. Rahmayani, and S. A. Husniar, “Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara,” Indones. J. Data Sci., vol. 1, no. 2, pp. 39–43, 2020, doi: 10.33096/ijodas.v1i2.13.

P-ISSN : 2686-4185
E-ISSN : 2714-9706

[7] E. P. W. Mandala, D. E. Putri, and R. Permana, “Penerapan Data Mining untuk Klasifikasi Hasil Panen Jamur Tiram Menggunakan Algoritma K-Nearest Neighbor,” J. Media Inform. Budidarma, vol. 7, no. 1, p. 223, 2023, doi: 10.30865/mib.v7i1.5252.
[8] H. P. Herlambang, F. Saputra, M. H. Prasetiyo, D. Puspitasari, and D. Nurlaela, “Perbandingan Klasifikasi Tingkat Penjualan Buah di Supermarket dengan Pendekatan Algoritma Decision Tree, Naive Bayes dan K-Nearest Neighbor,” J. Insa. - J. Inf. Syst. Manag. Innov., vol. 3, no. 1, pp. 21–28, 2023, doi: 10.31294/jinsan.v3i1.2097.
[9] E. Karyadiputra, . A., and . H., “Penerapan Data Mining Untuk Klasifikasi Spesies Ikan Di Lingkungan Akuatik Air Tawar,” Technol. J. Ilm., vol. 13, no. 3, p. 265, 2022, doi: 10.31602/tji.v13i3.6877.
[10] F. Liantoni, “Klasifikasi Daun Dengan Perbaikan Fitur Citra Menggunakan Metode K-Nearest Neighbor,” J. Ultim., vol. 7, no. 2, pp. 98–104, 2016, doi: 10.31937/ti.v7i2.356.
[11] Y. Miftahuddin, S. Umaroh, and F. R. Karim, “Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, Dan Manhattan Dalam Penentuan Posisi Karyawan,” J. Tekno Insentif, vol. 14, no. 2, pp. 69–77, 2020, doi: 10.36787/jti.v14i2.270.