Klasifikasi Jenis Obat Berdasarkan Gejala yang Dimiliki Pasien Menggunakan Metode K-Nearest Neighbors (KNN)

Ngakan Putu Bagus Ananta Wijaya[a1], Ida Ayu Gde Suwiprabayanti Putra[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]anantawijaya994@gmail.com
[2]iagsuwiprabyantiputra@unud.ac.id

Abstract

This research applies the K-Nearest Neighbors (KNN) algorithm to classify medicine types based on patient symptoms using a dataset from Kaggle with 200 rows and 6 columns. After preprocessing steps such as handling missing values, encoding categorical variables, and splitting data into training and testing sets, exploratory data analysis (EDA) was performed to understand the dataset's structure. The KNN model was evaluated with k values of 1, 2, and 3, finding the optimal k to be 3, achieving an accuracy of 77.50% with average precision of 0.76, recall of 0.69, and f1-score of 0.66. Lower accuracy was observed for k=2 (65.00%) and k=1 (67.50%), indicating that k=3 is the most effective for this dataset. These results suggest that while KNN is a viable method for classifying medicine types based on symptoms, larger datasets are recommended for improved accuracy.

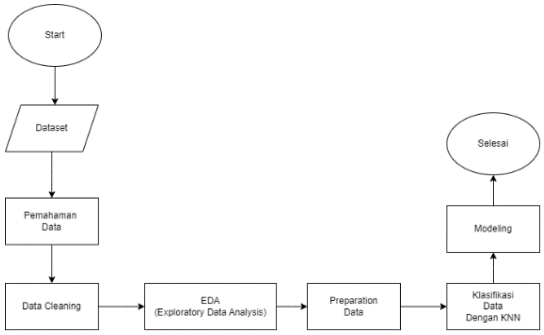
Keywords: *K-Nearest Neighbors (KNN), classify, medicine, exploratory data analysis (EDA), preprocessing*

1. Pendahuluan

Penggunaan teknologi dalam Kesehatan telah meningkatkan efisiensi dan akurasi dalam diagnosa dan pengobatan penyakit. Salah satu metode yang sangat popular dalam klasifikasi jenis obat berdasarkan gejala yang dimiliki pasien adalah menggunakan algoritma *K-Nearest Neighbors.* Algoritma ini merupakan metode pembelajaran mesin yang digunakan untuk memprediski kelas objek berdasarkan kelas objek yang paling dekat dengannya dan sangat cocok untuk digunakan dalam klasifikasi jenis obat berdasarkan gejala pasien [1]. Algoritma KNN adalah algoritma yang relatif sederhana dan mudah digunakan, tetapi memiliki beberapa kelebihan yang membuatnya sangat populer[2]. Salah satu kelebihan utama adalah kemampuan KNN untuk tidak memerlukan asumsi tentang distribusi data, yang membuatnya sangat efektif dalam menghadapi data yang tidak berdistribusi normal[3]. Selain itu, KNN juga dapat digunakan untuk mengklasifikasikan data yang memiliki atribut numerik dan kategorikal, membuatnya sangat fleksibel dalam aplikasi kesehatan yang beragam[4]. Jenis obat akan melalui proses klasifikasi berdasarkan data-data pasien seperti umur pasien, kandungan kolesterol, tekanan darah dan rasio natrium. Data-data tersebut akan dikumpulkan dan akan diolah melalui beberapa tahap yaitu tahap pemahaman data, preprocessing data, *Exploratory Data Analysis* (EDA), *preparation* data, dan evaluasi dengan algoritma KNN. Penelitian ini menggunakan python dan librarynya untuk mengolah data.

2. Metode Penelitian

Studi penelitan ini memanfaatkan algoritma pembelajaran mesin,yaitu metode klasifikasi yang menggunakan algoritma *K-Nearest Neighbors* (KNN). Diagram alir pada Gambar 1 menunjukkan beberapa tahapan penelitian ini.



Gambar 1. Diagram Alir

2.1. Pemahaman data

Studi ini menggunakan data sekunder dari situs *Kaggle,* yang datanya berjumlah 200 baris dan 6 kolom yang berisi informasi tentang umur, jenis kelamin, kolesterol, tekanan darah, rasio natrium, dan jenis obat. Tabel 1 menunjukkan data yang terdapat pada dataset.

Tabel 1. Tabel Dataset

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY
…	…	…	…	…	…	…

Pada tahap ini, program menggunakan library python (pandas, numpy, seaborn dan matplotlib) untuk memuat statistik distribusi jumlah data yang terdapat pada kolom-kolom dataset, mengecek tipe data dan statistik rata rata jumlah kolom, serta memvisualisasikan data dengan histogram.

2.2. Preprocessing data

Sebelum melanjutkan ke tahap berikutnya, dataset harus melalui tahap preprocessing data untuk menghilangkan data yang salah atau tidak sesuai. Tahapan preprocessing data termasuk tahapan berikut:

a. Pengecekan *missing values*
   Dataset harus diperiksa terlebih dahulu apakah terdapat nilai-nilai data yang hilang pada dataset. Pada dataset yang digunakan saat ini tidak terdapat nilai-nilai data yang hilang

b. Pengecekan duplikasi
   Dataset yang digunakan juga harus diperiksa apakah terdapat data yang duplikasi atau tidak. Pada dataset yang digunakan saat ini juga tidak terdapat data yang duplikasi

c. *Encoding*
   Karena beberapa fitur dalam dataset berupa data kategorikal, seperti pada kolom 'Sex', 'BP', 'Cholesterol', dan 'Drug', harus dilakukan proses encoding menggunakan LabelEncoder untuk mengkonversi nilai kategorikal menjadi numerik.

d. Pemisahan fitur dan target
   Fitur (X) dan target (y) dipisahkan untuk diproses lebih lanjut. Fitur 'Drug' dihapus dari dataset X karena merupakan target data.

2.3. *Exploratory Data Analysis* (EDA)

EDA merupakan metode yang digunakan untuk memahami dan menjelajahi data sebelum memulai analisis yang lebih mendalam[5]. Pada tahap ini dataset yang digunakan akan dieksplorasi dengan visualisasi dan statistik untuk memahami hubungan dataset. Ringkasan statistik dan visualisasi distribusi digunakan untuk memahami variabel numerik, sementara plot hubungan antar variabel membantu mengidentifikasi korelasi. Variabel kategorikal divisualisasikan untuk melihat distribusinya. Tahap ini juga mencakup penanganan missing values dan pemahaman tentang target jika ada.

2.4. *Preparation* data

Pada tahap *preparation* data, data dimodifikasi agar sesuai dengan kebutuhan analisis atau pemodelan. Langkah-langkah melibatkan pemisahan fitur dan target, *encoding* variabel kategorikal, penanganan nilai yang hilang, dan penyusunan data *training* dan *testing*. Untuk memastikan bahwa skala data konsisten, standarisasi atau normalisasi juga dapat dilakukan. Persiapan data yang baik akan dapat membuat data menjadi lebih siap digunakan dan akan memicu hasil yang akurat dan relevan

2.5. Klasifikasi dengan Algoritma KNN

Algoritma ini memiliki kemampuan untuk melakukan klasifikasi dengan mencari jarak terdekat antara data dengan K-Nearest Neighbors [1]. Pada dataset ini, jenis obat akan diklasifikasi berdasarkan gejala pasien yang ada di dalamnya. Adapun rumus dari algoritma KNN ini:

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

Euclidean distance = jarak ketetanggaan

xi = *data train*
yi = *data test*
I = variable
n = dimensi data

3. Hasil dan Diskusi

3.1. Pemahaman data

Pada tahap awal ini, dilakukan pemeriksaan pada dataset untuk memahami struktur dataset. Dataset memiliki ukuran 200 baris dan 6 kolom yang mencakup umur, jenis kelamin, kolesterol, tekanan darah, rasio natrium dan jenis obat. Program selanjutnya akan memeriksa tipe data kolom, memeriksa statistik deskriptif yang menunjukkan rata-rata data per kolom, dan memvisualisasikan histogram untuk menunjukkan distribusi dan frekuensi data-data per kolom pada dataset.

```
#PEMAHAMAN DATA

print("ukuran : ", df.shape)

ukuran :  (200, 6)

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Age          200 non-null    int64
 1   Sex          200 non-null    object
 2   BP           200 non-null    object
 3   Cholesterol  200 non-null    object
 4   Na_to_K      200 non-null    float64
 5   Drug         200 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

Gambar 2. Program Proses Pemahaman Data

p-ISSN: 2986-3929
e-ISSN: 3032-1948

```
df.describe()
```

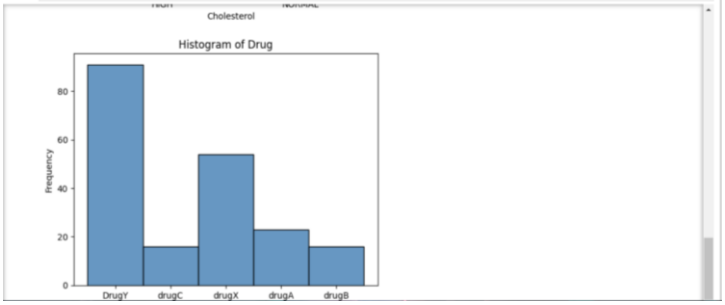
	Age	Na_to_K
count	200.000000	200.000000
mean	44.315000	16.084485
std	16.544315	7.223956
min	15.000000	6.269000
25%	31.000000	10.445500
50%	45.000000	13.936500
75%	58.000000	19.380000
max	74.000000	38.247000

```
df.Drug.value_counts()
```

```
Drug
DrugY    91
drugX    54
drugA    23
drugC    16
drugB    16
Name: count, dtype: int64
```

Gambar 3. Program Proses Pemahaman Data

```
# Histogram untuk variabel Drug
plt.figure()
sns.histplot(df['Drug'])
plt.title('Histogram of Drug')
plt.xlabel('Drug')
plt.ylabel('Frequency')
plt.show()
```



Gambar 4. Program Proses Pemahaman Data

3.2. Data Cleaning

Tahap selanjutnya adalah data cleaning yang bertujuan untuk penanganan nilai yang hilang serta pengecekan duplikat data. Pemeriksaan yang dilakukan menunjukkan bahwa tidak ada nilai yang hilang di seluruh kolom dataset dan data yang duplikat juga tidak ditemukan. Oleh karena itu, dataset ini bersifat konsisten dan siap untuk melanjutkan tahap analisis EDA.

```
#DATA CLEANING
```

```
df.isnull().sum()
```

```
Age            0
Sex            0
BP             0
Cholesterol    0
Na_to_K        0
Drug           0
dtype: int64
```

```
df.duplicated().sum()
```

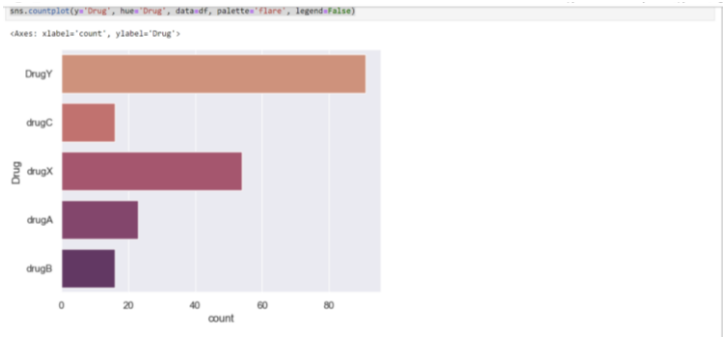
```
0
```

Gambar 5. Program Proses Data Cleaning

3.3. *Exploratory Data Analysis* (EDA)

EDA dilakukan untuk lebih dalam memahami struktur dan karakteristik yang terdapat pada dataset. Grafik-grafik yang dibuat memperlihatkan distribusi jumlah data pada kolom yang ditampilkan. Gambar 6 menunjukkan sebaran jumlah data dari kategori jenis obat. Informasi ini membantu pemahaman tentang dataset yang digunakan.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 6. Program Proses EDA

Gambar 7. Program Proses EDA

3.4. *Preparation* data

Dataset akan melalui proses encoding variabel kategorikal pada kolom ‘Sex’, ‘BP’, ‘Cholesterol’, dan ‘Drug’ dikarenakan data-data pada kolom yang disebutkan masih bertipe *string* dan perlu diubah menjadi tipe *integer* agar bisa diproses. Kemudian, dataset akan dibagi menjadi fitur (X) dan target (y), dan dibagi lagi menjadi *data training* dan *data testing* dengan rasio 80:20, dimana 80% untuk *training* dan 20% untuk *testing*. Setelah pembagian data, output ukuran pembagian *data training* dan *data testing* akan dicetak seperti pada Gambar 8.

Gambar 8. Program Proses *Preparation* Data

p-ISSN: 2986-3929
e-ISSN: 3032-1948

3.5. Modeling

Tahap terakhir adalah modeling dengan menggunakan algoritma KNN sebagai model. Model KNN diinisialisasi dengan jumlah tetangga yang dipakai ada (3), (2), (1).

a. KNN dengan k=3
Hasil akurasi KNN jika menggunakan k=3 sudah cukup bagus dikarenakan berhasil memperoleh akurasi sebesar 77.50% yang menandakan bahwa k yang digunakan cukup optimal. Model KNN juga menunjukkan hasil akurasi rata rata dari *precision* sebesar 0.76, *recall* sebesar 0.69 dan *f1-score* sebesar 0.66.

b. KNN dengan k=2
Hasil akurasi KNN jika menggunakan k=2, mengalami penurunan menjadi 65.00% yang menandakan bahwa k yang digunakan masih belum cukup optimal. Model KNN juga menunjukkan hasil akurasi rata rata dari *precision* sebesar 0.50, *recall* sebesar 0.59 dan *f1-score* sebesar 0.49 yang menandakan penurunan dari k = 3.

c. KNN dengan k=1
Hasil akurasi KNN jika menggunakan k=1, mengalami kenaikan sedikit hanya menjadi 67.50% yang menandakan bahwa k yang digunakan juga masih belum cukup optimal. Model KNN juga menunjukkan hasil akurasi rata rata dari *precision* sebesar 0.58, *recall* sebesar 0.60 dan *f1-score* sebesar 0.58 yang menandakan sedikit kenaikan dari k=2.

```
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train, y_train)

y_pred = knn.predict(x_test)
KNN_acc = accuracy_score(y_pred, y_test)

print(classification_report(y_test, y_pred))
print('Akurasi KNN = {:.2F}%'.format(KNN_acc*100))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        15
           1       0.56      0.83      0.67         6
           2       0.50      0.67      0.57         3
           3       1.00      0.20      0.33         5
           4       0.73      0.73      0.73        11

    accuracy                           0.78        40
   macro avg       0.76      0.69      0.66        40
weighted avg       0.82      0.78      0.76        40

Akurasi KNN = 77.50%
```

Gambar 10. Program *Modeling*

```
testing = {'Age': [35],
        'Sex': [1],
        'BP': [2],
        'Cholesterol': [0],
        'Na_to_K': [4.5]}

testing = pd.DataFrame(testing)
testing
```

	Age	Sex	BP	Cholesterol	Na_to_K
0	35	1	2	0	4.5

```
pred_coba = knn.predict(testing)
print("Hasil Prediksi Pasien Baru = ")
print(pred_coba)
```

```
Hasil Prediksi Pasien Baru =
[4]
```

Gambar 11. Program *Modeling*

4. Kesimpulan

Berdasarkan hasil studi penelitian yang sudah dilakukan, dapat diketahui bahwa k = 3 adalah k yang paling cocok dan optimal untuk digunakan dengan akurasi 77,50%, sedangkan k = 1 dan k = 2 lebih buruk dengan akurasi hanya 65,00% dan 67,50%, masing-masing. Karena jumlah data yang terbatas dalam dataset yang digunakan, akurasi tidak dapat mencapai 100%. Oleh karena itu, algoritma KNN dapat dengan mudah dan optimal mengklasifikasikan obat berdasarkan gejala

pasien. Oleh karena itu, peneliti menyarankan untuk menggunakan dataset dengan jumlah data yang lebih besar untuk mendapatkan hasil yang lebih akurat dan menghasilkan model terbaik.

Daftar Pustaka

[1] Hasran. (2020). *Indonesian Journal of Data and Science Klasifikasi Penyakit Jantung Menggunakan Metode K-Nearest Neighbor*. *1*(1), 6–10. http://bit.ly/datasetcardio.

[2] Novitasari, H. B., Hadianto, N., Sfenrianto, Rahmawati, A., Prasetyo, R., Miharja, J., & Gata, W. (2019). K-nearest neighbor analysis to predict the accuracy of product delivery using administration of raw material model in the cosmetic industry (PT Cedefindo). *Journal of Physics: Conference Series*, *1367*(1). https://doi.org/10.1088/1742-6596/1367/1/012008

[3] Azis, A., Zy, A. T., & Sunge, A. S. (2024). Prediksi Penjualan Obat Dan Alat Kesehatan Terlaris Menggunakan Algoritma K-Nearest Neighbor. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, *6*(1), 117–124. https://doi.org/10.47233/jteksis.v6i1.1078

[4] andiani, L., Ilmu Komputer, J., & Palupi Rini, D. (2019). Analisis Penyakit Jantung Menggunakan Metode KNN Dan Random Forest. In *Prosiding Annual Research Seminar* (Vol. 5, Issue 1).

[5] Sagala, N. T. M., & Aryatama, F. Y. (2022). *SISTEMASI: Jurnal Sistem Informasi Exploratory Data Analysis (EDA): A Study of Olympic Medallist Analisis Data Eksplorasi (EDA): Studi Peraih Medali Olimpiade* (Vol. 11, Issue 3). http://sistemasi.ftik.unisi.ac.id

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong