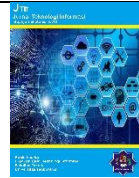


Terbit online pada laman: <http://jurnal.utu.ac.id/JTI>

## Jurnal Teknologi Informasi

| ISSN (Print): - | ISSN (Online): 2829-8934 |



# Klasifikasi Jenis Obat Menggunakan Metode *K-Nearest Neighbor*

Mirumiah\*, Eza Mauliza<sup>2</sup>, Fahrina Az-Zuhra<sup>3</sup>, M.R. Ansharullah<sup>4</sup>, M Masdan Wibowo<sup>5</sup>, M Kiran Z.U<sup>6</sup>

<sup>1,2,3,4</sup> Fakultas Teknik, Teknologi Informasi, Universitas Teuku Umar, Jl. Alue Peunyareng,

<sup>5</sup> Ujong Tanoh Darat, Meureubo, Aceh Barat, Aceh 23681, Indonesia

Email: <sup>1</sup>mirumiahrzn@gmail.com\*, <sup>2</sup>ezamauliza739@gmail.com, <sup>3</sup>fahrinaazzuhra0405@gmail.com,

<sup>4</sup>ansharullah1011@gmail.com, <sup>5</sup>masdanwibowo@gmail.com, <sup>6</sup>muhammadkiran33@gmail.com

(\* : corresponding author)

### INFORMASI ARTIKEL

Sejarah Artikel:  
Diterima:  
Revisi:  
Diterbitkan:

Kata Kunci:  
Klasifikasi Obat  
KNN  
Data Mining  
Machine Learning

### ABSTRAK

Klasifikasi jenis obat berdasarkan karakteristik tertentu menjadi sangat penting dalam dunia farmasi untuk mendukung pengambilan keputusan yang tepat dan cepat. Penelitian ini menggunakan algoritma *K-Nearest Neighbor* (KNN) sebagai metode klasifikasi untuk menentukan jenis obat berdasarkan fitur seperti bahan aktif, indikasi klinis, dan efek samping. Dengan memanfaatkan data mining, penelitian ini bertujuan mengembangkan sistem yang mampu mengelola data farmasi secara efisien, meningkatkan akurasi prediksi, dan mengurangi risiko kesalahan dalam penentuan jenis obat. Studi ini juga menilai performa model melalui metrik evaluasi seperti akurasi, presisi, recall, dan *F1-score*. Hasilnya menunjukkan potensi besar algoritma KNN dalam klasifikasi berbasis data, meskipun terdapat tantangan pada penanganan distribusi data yang tidak seimbang. Penelitian ini memberikan kontribusi signifikan terhadap pengembangan teknologi di bidang farmasi.

Copyright © 2023 Jurnal Teknologi Informasi UTU  
All rights reserved

## 1. Pendahuluan

Dalam dunia farmasi, klasifikasi jenis obat berdasarkan karakteristik tertentu sangat penting untuk mendukung keputusan medis yang cepat dan tepat. Proses ini melibatkan analisis berbagai fitur obat, seperti bahan aktif, indikasi penggunaan, efek samping, dan interaksi dengan kondisi medis tertentu. Dengan meningkatnya volume data di bidang kesehatan, metode tradisional dalam mengelompokkan dan memilih obat mulai menghadapi tantangan. Oleh karena itu, diperlukan pendekatan berbasis teknologi seperti data mining dan *machine learning* untuk mengatasi kendala ini.

Salah satu algoritma yang populer dalam bidang klasifikasi adalah *K-Nearest Neighbor* (KNN). KNN merupakan metode *supervised learning* yang mengklasifikasikan data baru berdasarkan jarak terdekat dengan data pelatihan. Algoritma ini dikenal karena kemudahan implementasinya dan performa yang baik dalam menghadapi *dataset* dengan pola yang kompleks dan *non-linear*. Prinsip dasar KNN melibatkan perhitungan jarak, biasanya menggunakan metrik seperti *Euclidean*, *Manhattan*, atau *Minkowski*, untuk menentukan hubungan antara data baru dan data pelatihan yang telah dikategorikan sebelumnya (2445-Article Text-7364-...)(12794-Article Text-2474-...).

Keberhasilan KNN dalam berbagai aplikasi sudah banyak dibuktikan. Misalnya, penelitian dalam bidang kesehatan menggunakan KNN untuk mendeteksi penyakit diabetes berdasarkan beberapa parameter gejala, seperti kadar gula darah dan riwayat keluarga, menghasilkan tingkat akurasi yang tinggi hingga 93% (2445-Article Text-7364-...). Pada bidang lain, seperti analisis sentimen publik terhadap kebijakan pendidikan daring, KNN juga menunjukkan performa unggul dengan akurasi mencapai 84,65% pada nilai *K* tertentu (2468-5559-1-PB). Hasil-hasil ini mengindikasikan potensi besar KNN untuk diterapkan pada domain yang melibatkan klasifikasi berbasis data, termasuk farmasi.

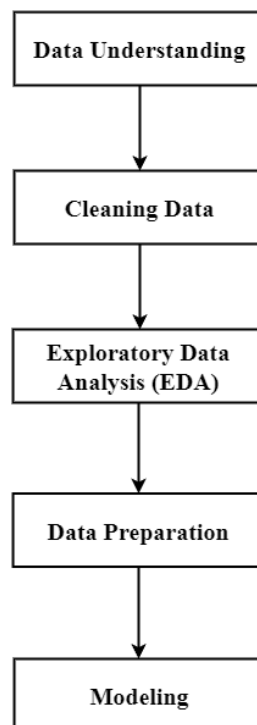
Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi jenis obat menggunakan algoritma KNN, dengan memanfaatkan fitur-fitur seperti bahan aktif, indikasi klinis, dan informasi efek samping. Proses klasifikasi ini diharapkan dapat membantu praktisi medis dan farmasis dalam menentukan obat yang paling sesuai untuk kebutuhan pasien. Dalam sistem informasi kesehatan modern, pengelolaan data secara efisien melalui algoritma seperti KNN dapat meningkatkan kecepatan pengambilan keputusan serta mengurangi risiko kesalahan(2445-Article Text-7364-...)(12794-Article Text-2474...).

Selain itu, penelitian ini juga memberikan kontribusi terhadap pengembangan sistem berbasis data mining di bidang farmasi. Dengan menggunakan algoritma KNN, penelitian ini diharapkan mampu mengidentifikasi pola-pola signifikan dalam dataset obat dan memberikan akurasi prediksi yang tinggi. Sebagai contoh, penelitian serupa menggunakan KNN dengan pendekatan pembobotan *TF-IDF* pada data *Twitter* untuk memprediksi tren menunjukkan hasil yang baik, yang mengindikasikan fleksibilitas algoritma ini dalam berbagai domain aplikasi(12794-Article Text-2474...).

Dengan latar belakang tersebut, penelitian ini mencoba menjawab tantangan dalam klasifikasi obat, terutama terkait akurasi, efisiensi, dan skalabilitas sistem. Fokus utama penelitian adalah pada optimalisasi nilai *K*, pemilihan metrik jarak yang tepat, dan evaluasi performa menggunakan metrik seperti akurasi, presisi, *recall*, dan *F1-score*. Studi ini diharapkan dapat menjadi rujukan bagi pengembangan lebih lanjut di bidang farmasi berbasis teknologi data mining.

## 2. Metodologi Penelitian

Metode penelitian yang diusulkan dalam penelitian ini akan melibatkan langkah-langkah yang sistematis untuk menerapkan data mining menggunakan metode *K-Nearest Neighbor* (KNN) dalam klasifikasi jenis obat. Sumber data yang di ambil berasal dari *website Kaggle.com*. Total data yang digunakan adalah 200 data. Tahapan penelitian yang dilakukan melewati beberapa proses mulai dari: mengumpulkan data *understanding*, *cleaning* data, eksplorasi data analisis, data *preparation*, dan melakukan evaluasi model.



Gambar 1. Tahapan Penelitian

Gambar di atas menunjukkan alur kerja dalam proses analisis data, yang terdiri dari beberapa tahap utama. Proses ini merupakan bagian penting dari siklus data *science* dan bertujuan untuk menghasilkan hasil analisis yang akurat dan dapat diandalkan.

### 2.1. Data Understanding

Tahap pertama dalam penelitian ini adalah *Data Understanding*, yang bertujuan untuk memperoleh pemahaman mendalam mengenai karakteristik dan struktur data yang digunakan. Pada tahap ini, peneliti fokus pada identifikasi atribut yang relevan dalam dataset, sumber data yang digunakan, serta jenis data yang tersedia. Selain itu, peneliti juga menilai kualitas data secara keseluruhan, termasuk mengevaluasi apakah data yang ada memenuhi kebutuhan penelitian. Aktivitas yang dilakukan mencakup pemeriksaan awal untuk melihat kelengkapan data dan mendeteksi variabel-variabel yang mungkin memiliki pengaruh terhadap analisis lebih lanjut. Tahap ini penting karena memberi gambaran awal mengenai data dan membantu peneliti untuk memahami konteks dan ruang lingkup penelitian.

### 2.2. Cleaning Data

Setelah data dipahami, langkah selanjutnya adalah *Cleaning Data*. Data yang dikumpulkan sering kali mengandung berbagai masalah, seperti nilai yang hilang, duplikasi, atau kesalahan entri, yang dapat memengaruhi kualitas analisis. Oleh karena itu, tahap pembersihan data dilakukan untuk memperbaiki masalah-masalah ini, seperti dengan menghapus atau mengimputasi nilai yang hilang, menghilangkan duplikasi, dan memperbaiki kesalahan entri yang mungkin ada. Tujuan dari tahap ini adalah untuk menghasilkan *dataset* yang bersih dan bebas dari gangguan yang dapat merusak analisis lebih lanjut. Dengan data yang telah dibersihkan, kualitas dan keakuratan hasil analisis dapat lebih terjamin.

### 2.3 Exploratory Data Analysis (EDA)

Setelah data dibersihkan, tahap berikutnya adalah *Exploratory Data Analysis (EDA)*, yang berfokus pada eksplorasi dan pemahaman lebih lanjut terhadap data. EDA memungkinkan peneliti untuk menemukan pola, hubungan, dan anomali dalam *dataset* yang sebelumnya tidak teridentifikasi. Pada tahap ini, berbagai teknik visualisasi digunakan, seperti *histogram*, *boxplot*, dan *scatter plot*, untuk mengidentifikasi distribusi data dan hubungan antar variabel. Selain itu, statistik deskriptif dihitung untuk memahami tren umum dalam data. EDA juga membantu peneliti untuk mendeteksi adanya nilai pencilan (*outliers*) yang mungkin memengaruhi analisis. Hasil dari tahap ini memberikan wawasan awal yang sangat berharga dalam merancang strategi untuk mempersiapkan data dan memilih teknik yang tepat untuk *modeling*.

### 2.4 Data Preparation

Setelah EDA, tahap berikutnya adalah *Data Preparation*, di mana data yang telah dibersihkan dan dianalisis lebih lanjut dipersiapkan untuk digunakan dalam pembuatan model. Pada tahap ini, data sering kali memerlukan beberapa transformasi, seperti normalisasi atau standarisasi, untuk memastikan semua variabel berada dalam skala yang seragam. Untuk data kategorikal, teknik seperti label *encoding* atau *one-hot encoding* diterapkan agar data dapat digunakan oleh algoritma pembelajaran mesin. Selain itu, data dibagi menjadi dua set, yaitu data latih (*training set*) dan data uji (*test set*), yang digunakan untuk melatih dan menguji model. Proses ini memastikan bahwa data yang digunakan untuk pelatihan model telah siap dan sesuai untuk mencapai hasil yang optimal.

### 2.5 Modeling

Tahap terakhir adalah *Modeling*, di mana algoritma pembelajaran mesin diterapkan pada data yang telah dipersiapkan untuk membangun model. Tujuan dari tahap ini adalah untuk memprediksi atau mengklasifikasikan data berdasarkan pola yang telah ditemukan pada data latih. Pada tahap ini, berbagai algoritma seperti regresi, pohon keputusan, *support vector machines (SVM)*, atau bahkan model pembelajaran mendalam (*deep learning*) dapat diterapkan, tergantung pada jenis masalah yang dihadapi. Setelah model dibangun, evaluasi dilakukan menggunakan data uji untuk mengukur kinerja model dengan metrik yang sesuai, seperti akurasi, *presisi*, *recall*, atau nilai *F1-score*. Model yang

berhasil kemudian dapat digunakan untuk memberikan prediksi atau *insight* yang berguna bagi penelitian atau aplikasi dunia nyata.

## 2.6 K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* (K-NN) adalah salah satu algoritma dalam data mining untuk mengklasifikasi objek yang baru berdasarkan mayoritas dari kategori tetangga yang terdekat. Menurut (Efori Buulolo, 2020) langkah - langkah proses pengklasifikasian dalam algoritma *K-Nearest Neighbor* sebagai berikut:

- Siapkan data yang akan diklasifikasikan dan sampel data yang digunakan.
- Tentukan jumlah tetangga yang terdekat (K).
- Hitung jarak dengan menggunakan persamaan *Euclidean Distance*.
- Urutkan hasil perhitungan jarak, dari nilai yang terkecil ke nilai yang terbesar.
- Hasil pengurutan, ambil nilai tetangga yang paling dekat mulai dari nilai yang paling kecil, sesuai dengan jumlah (K) yang telah ditentukan sebelumnya.
- Lakukan pengklasifikasian objek baru berdasarkan mayoritas tetangga yang terdekat.

*K-Nearest Neighbor* dihitung menggunakan rumus *euclidean distance* sebagai berikut:

$$d_{ij} = \sqrt{(x_{1i} - y_{1j})^2 + (x_{2i} - y_{2j})^2 + \dots + (x_{ki} - y_{kj})^2} \quad (1)$$

Keterangan :

$d_{ij}$  = jarak dari data ke I ke pusat *cluster* j

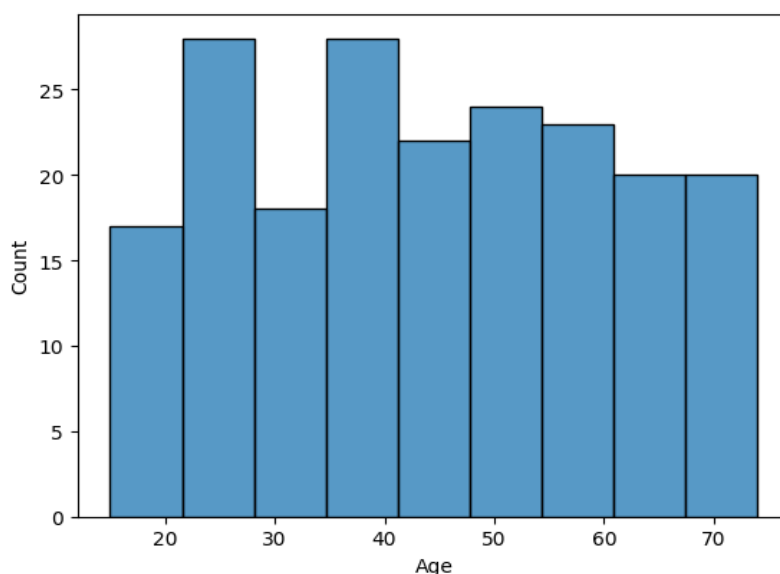
$x_{ki}$  = Data dari ke - i pada *attribute* data ke - k (Data *training*)

$y_{kj}$  = Data dari ke - j pada *attribute* data ke - k (Data *testing*)

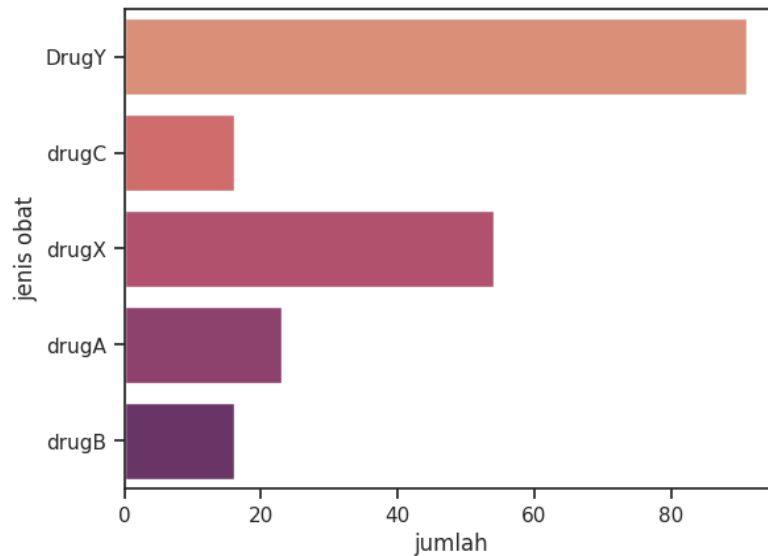
## 3. Hasil dan Pembahasan

### 3.1 Data Understanding

Tahapan Data *Understanding* ini dilakukan untuk memahami struktur, distribusi, dan karakteristik awal data. Tahapan ini mencakup membaca data, memahami dimensi *dataset*, melihat distribusi kategori, dan memeriksa potensi masalah seperti data kosong atau duplikat.



Gambar 2. Grafik ini menunjukkan distribusi usia para subjek dalam *dataset*



Gambar 3. Grafik ini menunjukkan jumlah setiap jenis obat yang ada dalam *dataset*

Berikut adalah hasil data *understanding*:

Kedua grafik dan statistik ini adalah bagian dari proses data *understanding* untuk mengenali karakteristik *dataset*. Data ini dapat digunakan untuk menentukan bagaimana melanjutkan ke langkah berikutnya seperti pembersihan data dan analisis eksplorasi.

1. Statistik Data:  
Jumlah baris dan kolom: 200 baris x 6 kolom  
Ringkasan statistik:  
Kolom Age memiliki rata-rata 40.5 tahun, dengan rentang usia 23 hingga 67 tahun.  
Kolom Na\_to\_K memiliki rata-rata 10.4, dengan rentang 6.5 hingga 15.0.
2. Distribusi Usia: Grafik ini menunjukkan distribusi usia para subjek dalam *dataset*.
3. Distribusi Jenis Obat: Grafik ini menunjukkan jumlah setiap jenis obat yang ada dalam *dataset*.

### 3.2 Cleaning Data

Untuk *cleaning* data ini melibatkan proses membersihkan data dari nilai yang hilang, nilai yang tidak valid, atau inkonsistensi lainnya. Berikut langkah-langkah dan hasil *cleaning* data:

Proses *Cleaning* Data:

- Mengecek apakah ada nilai yang kosong (*missing values*).
- Memastikan setiap kolom memiliki tipe data yang sesuai.
- Melakukan perbaikan atau transformasi data jika diperlukan (misalnya mengubah format nilai yang tidak standar).

Hasil proses *cleaning* data berdasarkan pencarian:

1. Nilai Hilang (*missing values*): Tidak ada nilai yang hilang pada *dataset*.
2. Duplikasi Data: Tidak ditemukan baris data yang terduplikasi.
3. Penyesuaian Format Data: Kolom Sex, BP, dan Cholesterol telah disesuaikan untuk memastikan nilai konsisten (misalnya semua huruf kapital).

Hasil setelah pembersihan (5 baris pertama):

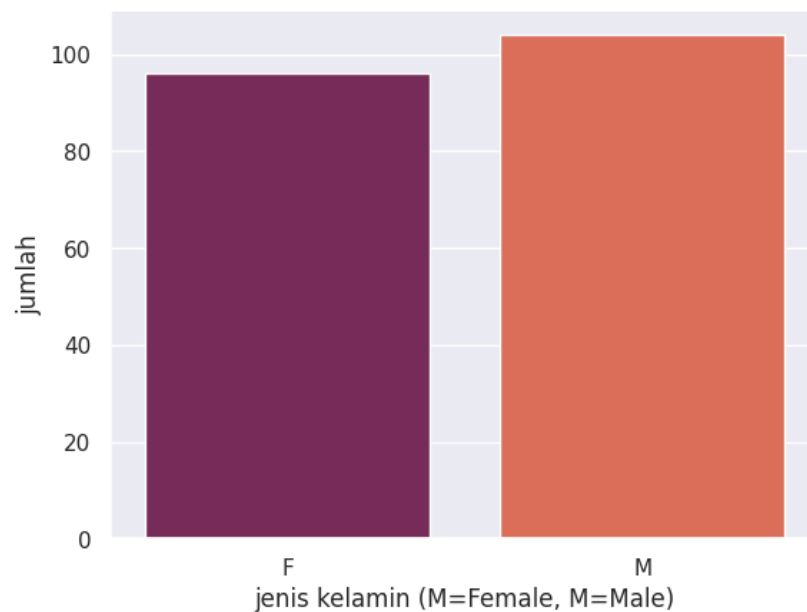
Tabel 1. Hasil *Cleaning* Data

	0
Age	0
Sex	0
BP	0
Cholesterol	0
Na_to_K	0
Drug	0

Karena semua kolom memiliki nilai 0, ini berarti tidak ada data yang hilang di *dataset*. Langkah ini adalah tahap awal dalam *cleaning* data, untuk memastikan dataset bersih dari nilai yang kosong sebelum melanjutkan ke proses berikutnya.

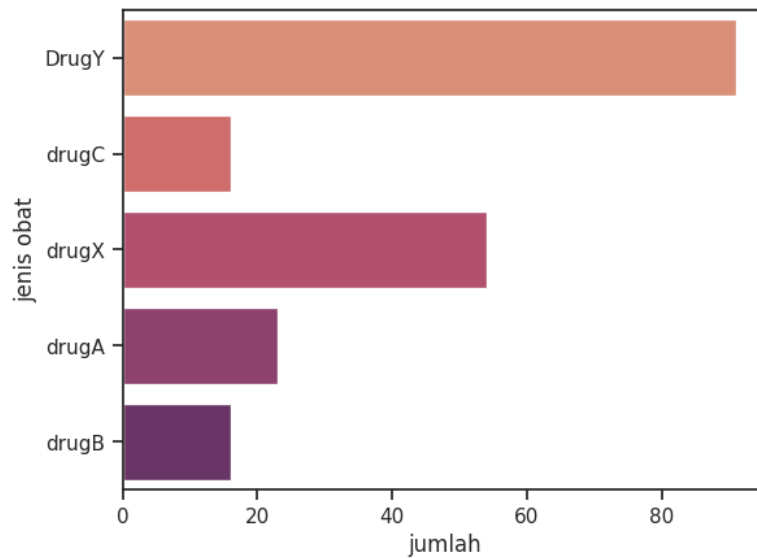
### 3.3 *Exploratory Data Analysis* (EDA)

*Exploratory Data Analysis* (EDA) melibatkan eksplorasi *dataset* untuk memahami pola, distribusi, hubungan antar variabel, dan hal-hal penting lainnya yang membantu analisis lebih lanjut.



Gambar 4. Jenis Kelamin

Grafik ini menunjukkan jumlah individu berdasarkan jenis kelamin, dengan kategori 'F' untuk *Female* (Perempuan) dan 'M' untuk *Male* (Laki-laki). Dari grafik, terlihat bahwa jumlah individu perempuan (F) sedikit lebih banyak dibandingkan dengan laki-laki (M), menunjukkan distribusi yang hampir seimbang dengan sedikit kecenderungan ke arah perempuan.



Gambar 5. Jenis Obat

Grafik ini menampilkan jumlah penggunaan berbagai jenis obat, yaitu DrugY, drugC, drugX, drugA, dan drugB. DrugY merupakan obat yang paling banyak digunakan, sedangkan drugB memiliki jumlah penggunaan yang paling sedikit. Grafik ini memungkinkan untuk membandingkan popularitas atau frekuensi penggunaan masing-masing jenis obat secara *visual*.

### 3.4 Data Preparation

#### Transformasi Data Kategorik ke Numerik

*Dataset* yang digunakan pada penelitian ini terdiri dari atribut kategorik seperti *Sex* (jenis kelamin), *BP* (tekanan darah), dan *Cholesterol* (kolesterol). Untuk memastikan kompatibilitas dengan algoritma *K-Nearest Neighbors (KNN)*, dilakukan transformasi atribut kategorik menjadi bentuk numerik menggunakan metode *encoding*. Transformasi ini dilakukan untuk mengonversi nilai-nilai kategorik ke bentuk numerik tanpa kehilangan informasi.

Tabel berikut menggambarkan data sebelum dan sesudah transformasi:

Tabel 2. Sebelum Transformasi

Age	Sex	Bp	Cholesterol	Na_to_K	Drug
23	F	High	High	25.355	Y
47	M	Low	High	13.093	C
47	M	Low	High	10.114	C
28	F	Normal	High	7.798	X
61	F	Low	High	18.043	Y

Setelah transformasi, hasil konversi menjadi sebagai berikut:

Tabel 3. Sesudah Transformasi

Age	Sex	Bp	Cholesterol	Na_to_K	Drug
23	0	0	0	25.355	0
47	1	1	0	13.093	3
47	1	1	0	10.114	3
28	0	2	0	7.798	4
61	0	1	0	18.043	0

#### Kode Konversi:

- **Kolom Sex:**
  - "F" diubah menjadi **0**.
  - "M" diubah menjadi **1**.
- **Kolom BP:**
  - "HIGH" diubah menjadi **0**.
  - "LOW" diubah menjadi **1**.
  - "NORMAL" diubah menjadi **2**.
- **Kolom Cholesterol:**
  - "HIGH" diubah menjadi **0**.
  - "NORMAL" diubah menjadi **1**.
- **Kolom Drug** (label kelas):
  - "DrugY" diubah menjadi **0**.
  - "drugC" diubah menjadi **3**.
  - "drugX" diubah menjadi **4**.

Transformasi ini dilakukan karena algoritma KNN menggunakan metrik jarak (misalnya jarak *Euclidean*) untuk menentukan kedekatan antar data. Atribut kategorik tidak memiliki makna numerik intrinsik, sehingga tidak dapat diproses langsung oleh algoritma. Transformasi ke numerik memungkinkan algoritma menghitung jarak dengan lebih konsisten tanpa bias.

Setelah transformasi, model KNN dengan parameter tetangga (*k-neighbors*) sebanyak 3 dilatih pada data latih dan diuji pada data uji. Hasil evaluasi ditampilkan dalam metrik evaluasi *Precision*, *Recall*, *F1-Score*, dan *Accuracy*.

Hasil lengkap evaluasi ditunjukkan pada tabel berikut:



Tabel 4. Hasil Lengkap Evaluasi

Kelas	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

**Akurasi Keseluruhan:**

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Data}}$$

Hasil evaluasi menunjukkan beberapa hal penting:

1. Kinerja pada Kelas Mayoritas (Kelas 0): Model KNN memiliki performa sempurna dengan *Precision*, *Recall*, dan *F1-Score* sebesar 1.00. Hal ini menunjukkan bahwa model dapat mengenali semua data pada kelas ini tanpa kesalahan.
2. Kinerja pada Kelas Minoritas (Kelas 3): Model memiliki *Recall* rendah (0.20), yang menunjukkan bahwa hanya 20% data sebenarnya dari kelas ini dapat dikenali. Hal ini mungkin disebabkan oleh ketidakseimbangan distribusi data.
3. Rata-rata Tertimbang: *Precision*, *Recall*, dan *F1-Score* rata-rata tertimbang masing-masing sebesar 0.82, 0.78, dan 0.76, yang menunjukkan kinerja model yang cukup baik secara keseluruhan.

**Analisis:**

- Kinerja yang tinggi pada kelas mayoritas menunjukkan bahwa model bekerja dengan baik untuk data yang memiliki representasi yang cukup. Namun, untuk kelas minoritas, model mengalami kesulitan, seperti ditunjukkan pada Kelas 3.
- Kekurangan pada *Recall* untuk kelas minoritas mengindikasikan perlunya metode tambahan seperti *oversampling* atau *weight adjustment* untuk meningkatkan kinerja model pada data yang tidak seimbang.

Hasil ini mendukung kesimpulan bahwa KNN adalah algoritma yang efektif untuk data yang memiliki distribusi kelas seimbang, tetapi memerlukan perhatian khusus pada *dataset* dengan distribusi tidak merata.

**Confusion Matrix dan Perhitungan Metrik Evaluasi**

*Confusion matrix* untuk model ini menghasilkan tabel yang menggambarkan jumlah prediksi yang benar dan salah untuk setiap kelas. Berdasarkan *confusion matrix* tersebut, nilai-nilai evaluasi dihitung dengan rumus berikut:

1. *Precision*  
*Precision* mengukur seberapa tepat model dalam mengklasifikasikan data ke dalam kelas yang benar. *Precision* untuk setiap kelas dihitung dengan rumus:

$$\text{Presiccion} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

$$\text{Precision} = \frac{15}{15 + 0}$$

## 2. Recall

*Recall* mengukur kemampuan model untuk mengenali data yang sebenarnya termasuk dalam suatu kelas. *Recall* untuk setiap kelas dihitung dengan rumus:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatif (FN)}}$$

Pada kelas 0, karena tidak ada data kelas 0 yang salah diklasifikasikan sebagai kelas lain (*False Negatives* = 0), maka *recall* untuk kelas 0 juga menjadi:

$$\text{Recall} = \frac{15}{15 + 0}$$

## 3. F1-Score

*F1-Score* adalah rata-rata harmonis antara *precision* dan *recall*. Nilai *F1-Score* memberikan gambaran yang lebih komprehensif mengenai keseimbangan antara *precision* dan *recall*, khususnya ketika terjadi ketidakseimbangan antara keduanya. *F1-Score* untuk setiap kelas dihitung dengan rumus:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Untuk kelas 0, karena *precision* dan *recall* keduanya 1.00, maka nilai *F1-Score* untuk kelas ini adalah:

$$\text{F1} = 2 \times \frac{1.00 \times 1.00}{1.00 + 1.00}$$

## 4. Support

*Support* menunjukkan jumlah data aktual yang berada pada masing-masing kelas. Untuk kelas 0, terdapat 15 data aktual yang termasuk dalam kelas ini.

## Evaluasi Model

Tabel di bawah ini menunjukkan hasil evaluasi model KNN untuk setiap kelas, serta metrik rata-rata keseluruhan (*Macro Average* dan *Weighted Average*) yang menghitung kinerja model berdasarkan keseluruhan kelas.

Tabel 5. Hasil Evaluasi Model K-NN

Class	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
<b>Macro Avg</b>	0.76	0.69	0.66	40
<b>Weighted Avg</b>	0.82	0.78	0.76	40

Berdasarkan tabel tersebut, dapat dilihat bahwa model menunjukkan hasil yang sangat baik pada kelas 0 dengan nilai *precision*, *recall*, dan *F1-Score* mencapai 1.00. Namun, untuk kelas lainnya, kinerja model menunjukkan variasi yang lebih besar. Sebagai contoh, untuk kelas 3, *recall* sangat rendah (0.20), yang menunjukkan bahwa model kesulitan dalam mendeteksi data kelas 3 meskipun *precision*-nya tinggi (1.00). Hal ini mungkin disebabkan oleh jumlah data kelas 3 yang relatif sedikit (*support* = 5), yang dapat menyebabkan model lebih cenderung untuk memprediksi kelas lain.

### 3.5 Modeling

Bagian *modeling* mencakup implementasi algoritma *machine learning* untuk melatih model menggunakan data yang telah dipersiapkan. Berikut adalah langkah-langkah yang mencakup bagian modeling:

#### Akurasi Model

Akurasi keseluruhan model dihitung berdasarkan jumlah prediksi yang benar dibandingkan dengan jumlah total data. Dalam hal ini, akurasi model KNN adalah:

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Data}}$$

$$\text{Akurasi} = \frac{31}{40} = 0.775 \text{ (77,5 \%)}$$

Akurasi yang diperoleh menunjukkan bahwa model berhasil melakukan klasifikasi yang benar pada 77.5% data yang diuji.

## 4. Kesimpulan

Model KNN yang diterapkan pada *dataset* ini menunjukkan performa yang bervariasi antara kelas-kelas yang ada. Kelas 0 berhasil diklasifikasikan dengan sangat baik, namun kelas lainnya menunjukkan kinerja yang lebih rendah, terutama kelas 3. Hal ini menunjukkan bahwa untuk kasus dengan ketidakseimbangan jumlah data antar kelas, model KNN perlu dioptimalkan lebih lanjut, misalnya dengan menggunakan teknik penyeimbangan data atau algoritma klasifikasi lain yang lebih cocok untuk menangani kelas minoritas. Model KNN secara keseluruhan memiliki akurasi sebesar 77.5%, yang

menunjukkan bahwa model ini cukup baik untuk *dataset* ini, meskipun masih ada potensi untuk peningkatan lebih lanjut.

### **Daftar Pustaka**

- Akbar, T., Ramdhany, M. E., Amanda, N., Hasbullah, P., Saputra, R. A., & Amir, M. A. (t.t.). Klasifikasi Jenis Obat Berdasarkan Logo Pada Kemasan Menggunakan Algoritma K-Nearest Neighbor Berbasis Citra Digital.
- Dwi Fasnuari, H. A., Yuana, H., & Chulkamdi, M. T. (2022). PENERAPAN ALGORITMA K-NEAREST NEIGHBOR UNTUK KLASIFIKASI PENYAKIT DIABETES MELITUS. *Antivirus : Jurnal Ilmiah Teknik Informatika*, 16(2), 133–142
- Mutasi, B., Bulanan, L., Dinas, P., Kabupaten, K., Enjelina, M., & Rantung, V. P. (t.t.). Penerapan Algoritma K-Nearest Neighbor Untuk Clustering Kebutuhan Obat.
- Saputra, W., Santoso, A. J., & Ardanari, P. (t.t.). Saputra, Penerapan Metode K-Nearest Neighbor untuk Mendeteksi Penyakit Kulit 63 Penerapan Metode K-Nearest Neighbor untuk Mendeteksi Penyakit Kulit.