JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 12 No. 2, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v12i2.4191

PREDIKSI KUALITAS AIR SUNGAI DI JAKARTA MENGGUNAKAN KNN YANG DIOPTIMALISASI DENGAN PSO

Iustisia Natalia Simbolon[1*], Hari D.S.J Siburian[2], Wybren Agung Manik[3]

[1,2,3] Informatika; Institut Teknologi Del; Jl. Sisingamangaraja Sitoluama, Laguboti, Toba, Sumatera Utara, Indonesia, 22381

Riwayat artikel:

Received: 8 Maret 2024
Accepted: 30 Maret 2024
Published: 2 April 2024

Keywords:
River water quality, Particle swarm optimization, K-Nearest neighbors, feature selection.

Corespondent Email:
iustisia.simbolon@del.ac.id

Abstrak. Kualitas air sungai merupakan isu lingkungan penting bagi masyarakat dan pemerintah. Penelitian ini fokus pada kualitas air sungai di Jakarta dengan 21 atribut yang berbeda dari standar KLHK. Peneliti menganalisis pengaruh seluruh atribut tersebut menggunakan algoritma *K-Nearest neighbor* (KNN) yang dioptimalisasi *dengan Particle swarm optimization* (PSO) untuk memprediksi kualitas air sungai di Jakarta. Penelitian ini bertujuan mengembangkan model prediksi kualitas air sungai yang dioptimasi menggunakan algoritma PSO-KNN serta membangun prototipe aplikasi web berbasis *Flask. Preprocessing* data dilakukan dalam tiga tahap utama: *data cleaning, data transformation*, dan *balancing data* untuk mengatasi masalah *missing value, outlier*, dan ketidakseimbangan data. Kemudian dilakukan seleksi fitur untuk mengidentifikasi atribut paling berpengaruh dimana dari total 21 atribut didapatkan 8 atribut paling berpengaruh terhadap kualitas air sungai di Jakarta. Hasil penelitian menunjukkan bahwa model PSO-KNN mencapai akurasi 95,8%, lebih baik daripada model KNN tanpa optimasi yang hanya mencapai 77,9%. Seleksi fitur membantu mengidentifikasi atribut-atribut yang paling berpengaruh dalam memprediksi kualitas air sungai di Jakarta.

Abstract. *The quality of river water is an important environmental issue for both the public and the government. This research focuses on the river water quality in Jakarta, utilizing 21 different attributes from the KLHK standard. The researchers analyze the influence of all these attributes using the K-Nearest Neighbor (KNN) algorithm optimized with Particle Swarm Optimization (PSO) to predict the river water quality in Jakarta. The aim of this study is to develop a predictive model for river water quality that is optimized using the PSO-KNN algorithm, as well as to build a web application prototype based on Flask. Data preprocessing is carried out in three main stages: data cleaning, data transformation, and data balancing to address issues such as missing values, outliers, and data imbalance. Feature selection is then conducted to identify the most influential attributes on river water quality in Jakarta, where out of a total of 21 attributes, 8 most influential attributes are obtained. The research results show that the PSO-KNN model achieves an accuracy of 95.8%, which is better than the non-optimized KNN model that only reaches 77.9% accuracy. Feature selection helps identify the attributes that have the most impact on predicting river water quality in Jakarta.*

1. PENDAHULUAN

Air adalah sumber daya alam yang sangat penting dan diperlukan untuk kegiatan dan kelangsungan hidup makhluk hidup, termasuk manusia, hewan, dan tumbuhan. Sungai adalah salah satu pilihan sumber air yang bisa diolah untuk kebutuhan manusia. Sungai berfungsi sebagai sumber air terdekat bagi penduduk di daerah pedesaan dan perkotaan, serta habitat bagi berbagai ekosistem air. Namun, dengan meningkatnya jumlah penduduk, pertumbuhan industri, dan perkembangan ekonomi, kualitas air sungai mengalami penurunan. Hal ini terlihat dari kualitas air yang mengalir di sungai yang tercemar. [1]

Penelitian ini bertujuan untuk membangun sistem prediksi kualitas air sungai di Jakarta dengan menggunakan algoritma *K-Nearest neighbor* (KNN) berbasis *Particle swarm optimization* (PSO). Air sebagai sumber daya alam yang penting menghadapi tantangan serius dalam menjaga kualitasnya akibat pertumbuhan populasi, industri, dan ekonomi Sungai Jakarta memiliki atribut yang berbeda dengan standar Kementerian Lingkungan Hidup dan Kehutanan Republik Indonesia (KLHK), terdapat 21 atribut yang mempengaruhi kualitas air sungai dijakarta. [2]

Penelitian ini mencoba untuk menganalisis atribut-atribut yang paling berpengaruh dalam menentukan kualitas air sungai di Jakarta untuk memberikan pemahaman yang lebih mendalam tentang kondisi sungai dan memungkinkan pengembangan solusi dan prototipe aplikasi yang berfokus pada pengelolaan sungai dan pemulihan lingkungan.

Pendekatan menggunakan *machine learning* dengan algoritma KNN dipilih karena telah terbukti memiliki akurasi yang tinggi. Hal ini dibuktikan oleh penelitian berjudul "Analisis Perbandingan Klasifikasi Prediksi Penyakit Hepatitis dengan Menggunakan Algoritma KNN, Naive Bayes dan Neural Network" dimana dalam penelitian tersebut algoritma KNN dibandingkan dengan Naive Bayes dan Neural Network untuk memprediksi penyakit hepatitis, dimana hasilnya menunjukkan KNN memiliki akurasi paling tinggi yaitu 93%, naive bayes 76,92% dan Neural Network 82,97% [3]. Namun KNN juga memiliki kelemahan yaitu dalam konteks seleksi fitur, KNN tidak secara otomatis memilih subset fitur yang paling berpengaruh atau relevan Semua fitur yang ada akan digunakan dalam perhitungan jarak, termasuk fitur-fitur yang mungkin tidak memberikan kontribusi signifikan terhadap hasil prediksi. Hal ini dapat mengakibatkan *overfitting*, dimana model KNN terlalu fokus pada fitur-fitur yang tidak relevan dan mengabaikan fitur-fitur yang penting. Oleh karena itu, peneliti menggunakan algoritma optimasi PSO yang dapat meningkatkan akurasi model KNN dengan memilih subset fitur yang lebih relevan [4].

Keunikan dari penelitian ini yaitu mengidentifikasi dan memilih fitur-fitur (variabel-variabel) yang paling relevan dan berpengaruh terhadap kualitas air sungai di Jakarta. Penggunaan PSO dalam algoritma KNN membantu mengoptimalkan pemilihan fitur-fitur tersebut, sehingga didapatkan kombinasi fitur terbaik untuk meningkatkan akurasi sistem prediksi. Peneliti memanfaatkan kemampuan PSO untuk melakukan pencarian ruang fitur secara adaptif dan menemukan kombinasi fitur yang paling optimal Dengan identifikasi fitur-fitur berpengaruh yang lebih akurat dan efisien, hasil penelitian ini diharapkan dapat memberikan manfaat pengelolaan lingkungan dan pemulihan kualitas air sungai di Jakarta.

2. TINJAUAN PUSTAKA

2.1 Preprocessing Data

Preprocessing data adalah langkah untuk pembersihan data, contohnya menghapus *noise* dan data yang kurang konsisten, integrasi data yang mana beberapa sumber data bisa digabungkan menjadi satu bagian, transformasi data dimana data diubah dan disatukan ke dalam bentuk yang sesuai untuk tugas operasi agregasi dan reduksi data, termasuk seleksi dan ekstraksi fitur. Dalam penelitian ini, metode *preprocessing* yang digunakan terdiri atas 3 yaitu [5] :

1. Data cleaning

Data cleaning yaitu teknik pembersihan dalam data yang biasanya digunakan untuk menangani *missing value* pada data. Terdapat beberapa cara yang dilakukan dalam menangani *missing value* pada dataset, antara lain [6] :

- *Delete Strategy*, yaitu penghapusan nilai *missing value*
- *Ignored value strategy*, yaitu mengabaikan nilai *missing value*
- *Predicted value strategy*, yaitu strategi untuk melengkapi *missing value* dengan nilai yang seharusnya.
- *Distributed Value Strategy*, yaitu memprediksi distribusi atas semua nilai yang mungkin.

2. *Data Transformation*

Transformasi data melibatkan proses mengubah data mentah menjadi format yang lebih cocok untuk analisis atau pemodelan dengan memodifikasi skala, bentuk, atau distribusi data. Tujuannya adalah untuk memenuhi asumsi model statistik, mengurangi pengaruh *outlier*, atau meningkatkan pemahaman terhadap data [7].

Outlier adalah nilai data yang signifikan atau ekstrem yang sangat berbeda dengan pola umum dari data tersebut. Jika tidak ditangani dengan baik, *outlier* dapat menyebabkan distorsi pada analisis statistik dan model *machine learning* sehingga perlu ditangani dengan transformasi data. *Outlier* terlebih dahulu harus dideteksi, salah satunya dengan metode *Z-score*.

Metode *Z-score*, juga dikenal sebagai metode skor standar, digunakan untuk mengidentifikasi *outlier* dalam suatu dataset dengan mengukur jarak antara sebuah titik data dan rata-rata dalam satuan standar deviasi. Dalam metode ini, nilai *Z-score* dihitung dengan membandingkan nilai titik data dengan rata-rata dan standar deviasi dari dataset. Persamaan dalam metode *Z-score* ditunjukkan melalui persamaan (1) :

$$\boldsymbol{Z = \frac{X-\mu}{\sigma}} \quad \textbf{(1)}$$

Keterangan :

Z : Nilai *Z-score* untuk titik data tertentu.

X : Nilai titik data.

M : Rata-rata dari dataset.

Σ : Standar deviasi dari dataset.

Setelah *outlier* terdeteksi, dilakukan transformasi data, salah satu tekniknya yaitu tranformasi *box-cox.* Transformasi data *Box-cox* adalah sebuah metode yang digunakan untuk mengubah distribusi data menjadi lebih normal dengan memperhitungkan atribut lambda (λ). Metode ini membantu dalam penanganan *outlier* dalam data.

Dalam proses transformasi data *Box-cox*, rumus yang digunakan ditunjukkan melalui persamaan (2):

$$Y(\lambda) = \begin{cases} (X^{\lambda} - 1) / \lambda, & \text{jika } \lambda \neq 0 \\ \log(X), & \text{jika } \lambda = 0 \end{cases} \quad \textbf{(2)}$$

Keterangan:

Y(λ) : Data yang telah ditransformasi

X : Data mentah

Λ : Parameter transformasi

3. *Balancing data*

Balancing data adalah proses untuk menyeimbangkan jumlah data antara kelas mayoritas dan kelas minoritas dalam sebuah dataset. Ketidakseimbangan data dapat terjadi ketika satu kelas memiliki jumlah data yang jauh lebih banyak daripada kelas lainnya, sehingga dapat mempengaruhi kinerja model yang dibangun.

Beberapa teknik yang dapat digunakan untuk *balancing data* antara lain [8]:

- *Undersampling*: mengurangi jumlah sampel dari kelas mayoritas sehingga jumlah sampel dari kedua kelas menjadi seimbang.
- *Oversampling*: menambah jumlah sampel dari kelas minoritas sehingga jumlah sampel dari kedua kelas menjadi seimbang. Salah satu teknik *oversampling* yang populer adalah SMOTE *(Synthetic Minority Oversampling Technique).*
- Kombinasi dari undersampling dan *oversampling.*

Adapun dalam penelitian ini akan menggunakan teknik *oversampling* SMOTE. SMOTE *(Synthetic Minority Oversampling Technique)* adalah teknik *oversampling* yang digunakan untuk menyeimbangkan dataset yang tidak seimbang. Teknik ini menghasilkan sampel sintetis baru dari kelas minoritas dengan cara menggabungkan sampel yang ada dan membuat sampel sintetis baru di antara sampel-sampel tersebut.

2.2 K-Fold Cross validation

Cross-Validation (CV) adalah teknik statistik yang digunakan untuk mengevaluasi kinerja atau model algoritma dengan membagi dataset menjadi dua bagian, yaitu bagian pelatihan dan bagian validasi. Bagian pelatihan digunakan untuk melatih model atau algoritma, sedangkan bagian validasi digunakan untuk menguji performa model atau algoritma. Jenis *Cross-Validation* yang digunakan dapat dipilih berdasarkan ukuran dataset. Metode *Cross-Validation* umumnya dipilih karena dapat mengurangi waktu komputasi dan masih menghasilkan estimasi performa yang akurat [9].

2.3 K-Nearest neighbor (KNN)

KNN merupakan sebuah algoritma klasifikasi yang menggunakan nilai K untuk menentukan kelas dari data baru yang akan diklasifikasikan. Algoritma ini bekerja dengan cara mencari data yang memiliki kesamaan atau kedekatan dengan data yang akan diklasifikasikan. Dalam algoritma KNN, data yang berdekatan dengan data yang akan diklasifikasikan disebut sebagai tetangga (*neighbor*). Tujuan dari algoritma ini adalah untuk melakukan klasifikasi pada objek baru dengan membandingkan nilai atributnya dengan data latih yang ada. Salah satu metode dalam algoritma KNN yaitu *Euclidean distance. Euclidean distance* adalah jarak antara dua titik untuk menemukan dua titik pada bidang, panjang segmen yang menghubungkan dua titik diukur. Seperti namanya memberikan jarak antara dua titik atau jarak garis lurus [10]. Berikut adalah rumus *Euclidean distance* yang ditunjukkan pada persamaan (3):

$$\mathbf{d} = \sqrt{\sum_{i-1}^{n}(xi - yi)^2} \quad \textbf{(3)}$$

Keterangan :
d = Jarak Euclidean antara x dan y
x = Data uji
y = Data training
I = Setiap data
n = Jumlah data

2.4 Particle swarm optimization (PSO)

PSO adalah suatu algoritma yang menggunakan populasi untuk mengoptimalkan solusi dengan cara mengeksploitasi individu dalam populasi untuk menuju daerah penyelesaian yang terdapat dalam daerah pencarian. Dalam PSO, populasi dijuluki sebagai "swarm" dan individu disebut sebagai "*Particle*". Setiap partikel dalam swarm bergerak dengan kecepatan yang disesuaikan dengan daerah pencarian, dan menyimpan posisi terbaik yang pernah dicapai sebagai referensi [11].

2.5 Confusion matrix

Confusion matrix adalah sebuah tabel yang digunakan untuk mengukur kinerja dari model klasifikasi dalam *machine learning* Tabel ini menggambarkan jumlah data yang diklasifikasikan dengan benar maupun salah. Terdapat empat nilai yang dihasilkan dalam tabel *confusion matrix*, yaitu [12]:

- *True Positive* (TP): Jumlah data yang bernilai positif dan diprediksi benar sebagai positif.
- *False Positive* (FP): Jumlah data yang bernilai negatif tetapi diprediksi sebagai positif.
- *False Negative* (FN): Jumlah data yang bernilai positif tetapi diprediksi sebagai negatif.
- *True Negative* (TN): Jumlah data yang bernilai negatif dan diprediksi benar sebagai negatif.

Adapun persamaan-persamaan yang digunakan dalam *confusion matrix* yaitu :

1. Akurasi adalah tingkat tingkat kedekatan antara nilai prediksi dengan nilai aktual.Semakin tinggi nilai akurasi maka semakin mampu model tersebut dalam memprediksi label dari sebuah data .
2. Ketepatan *(Precision*) adalah tingkat ketepatan data yang sukses diprediksi positif, dibandingkan dengan seluruh data yang diprediksi positif, yang kenyataannya benar dan tidak benar .
3. Sensitivitas (*Recall*) adalah tingkat keberhasilan data saat diprediksisebagai positif dibandingkan dengan seluruh data yang pada kenyataannya positif.
4. *F-Score* adalah penggabungan nilai presisi dan *recall* untuk memberikan keseimbangan pada *precision* dan *recall*.

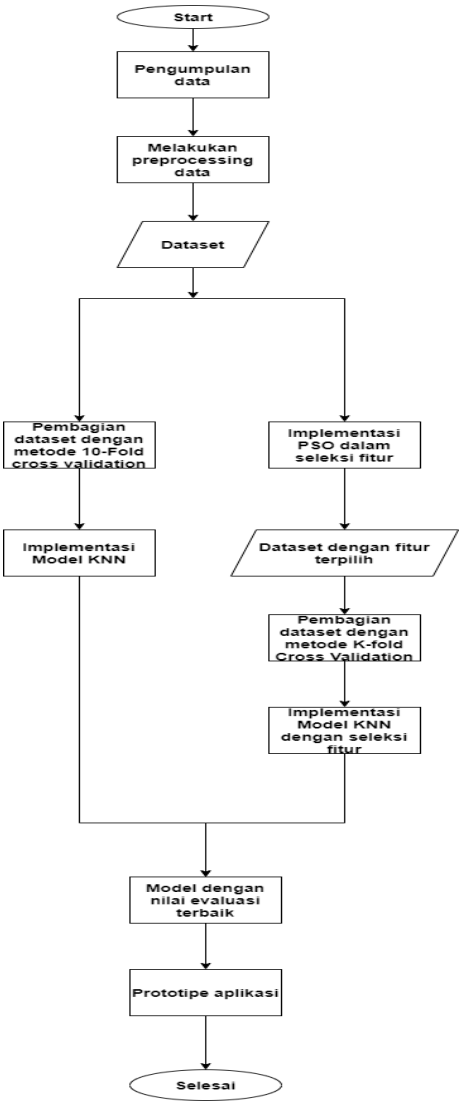
3. METODE PENELITIAN

Dalam penelitian ini, diuraikan metode

penelitian yang dilakukan. Metode penelitian dapat dijelaskan melalui Gambar 1.

3.1. Pengumpulan Data

Semua data yang digunakan dalam penelitian ini merupakan data kualitas air sungai dalam kurun waktu 2018 hingga 2020[13]. Dataset terbuka tersebut berjumlah 700 baris yang terdiri dari 21 atribut diantaranya Zat padat tersuspensi, BOD, COD, Total fosfat, Nitrat, Kadmium, Crom hexavalen, Tembaga, Merkuri, Seng, Flourida, Nitrit, Klorin bebas, Sulfida, Minyak dan Lemak, Senyawa aktif biru metilen, Fenol, Bakteri koli, Bakteri koli tinja, Indeks pencemar, dan Kategori. Adapun atribut kategori yang menjadi label target dengan 3 kategori target yaitu cemar ringan, cemar sedang dan cemar berat



Gambar 1. Metode Penelitian

3.2. Preprocessing Data

Preprocessing Data adalah tahap penelitian untuk membersihkan, mengubah, dan mengorganisasi data sehingga dapat diinterpretasikan dan diproses dengan benar oleh algoritma *machine learning*. Adapun tahapan *preprocessing data* dalam penelitian ini yaitu [14].

1. *Data Cleaning*
   Data cleaning pada penelitian ini bertujuan untuk membersihkan data dari *missing value.* Penanganan pada *missing value* yang dilakukan pada tahap ini yaitu melalui *delete strategy*, yaitu penghapusan baris data yang mengandung nilai *missing value.*
2. *Data Transformation*
   Data transformation pada penelitian ini bertujuan untuk mengurangi efek nilai *outlier* pada data. *Outlier* terlebih dahulu diidentifikasi dengan *Z-Score method* kemudian ditangan dengan *box cox transformation*
3. *Balancing data*
   Balancing data pada penelitian ini bertujuan untuk menyeimbangkan jumlah kelas dalam data. Hal ini dilakukan agar saat pembuatan model prediksi yang didapatkan seimbang sesuai dengan jumlah kelas data.

3.3 Pembagian dataset ke dalam 10-fold cross validation

Dalam 10-*Fold Cross Validation*, data dibagi menjadi 10 fold dengan ukuran yang sama, sehingga terdapat 10 subset data yang digunakan untuk mengevaluasi kinerja model atau algoritma. Setiap subset data diuji menggunakan 1 kali uji dan dilatih menggunakan 9 kali latih. Kedua model yang akan dibangun terlebih dahulu harus melewati tahap ini.

3.4 Implementasi model KNN (21 atribut)

Pembuatan model KNN dilakukan melalui langkah-langkah berikut [15]:

1. Menentukan nilai K, dalam penelitian ini K yang akan diuji coba yaitu K= 3,5,7,9,11,13,15,17,19.
2. Menghitung jarak antara data uji dan data latih menggunakan metrik jarak *Euclidean*.

3. Proses KNN dilakukan dengan mencari K tetangga terdekat dari setiap data uji berdasarkan jarak yang telah dihitung.
4. Kemudian, kita akan melakukan voting mayoritas dari tetangga terdekat untuk menentukan label prediksi dari data uji.
5. Setelah proses klasifikasi selesai, langkah selanjutnya adalah melakukan evaluasi kinerja model.
6. Hasil kinerja model akan dikembalikan ke dalam 10-fold cross validation, dimana nilai evaluasi model akan dimuat dalam 10 fold yang dirata-ratakan untuk menjadi nilai evaluasi model.

3.5 Implementasi PSO-KNN

Pembuatan model PSO-KNN dilakukan melalui Langkah-langkah berikut.

1. Inisialisasi partikel-partikel dengan posisi dan kecepatan awal secara acak.
2. Melakukan iterasi PSO untuk menemukan fitur-fitur terbaik dengan memaksimalkan nilai akurasi dari model KNN yang telah diperoleh
3. Pada setiap iterasi, kita mengupdate posisi partikel dan menyimpan skor F1-score terbaik dari setiap partikel (Pbest) serta skor F1-score terbaik dari seluruh populasi partikel (Gbest).
4. Setelah proses seleksi fitur selesai, kita melakukan evaluasi metrik kinerja model pada setiap fold dari validasi silang (cross-validation).
5. Fitur-fitur yang didapat kemudian menjadi model KNN yang baru dengan nilai K dari model awal terbaik
6. Tahapan Kembali ke prose KNN, dimana hasil evaluasi didapatkan dari rata-rata 10 fold selama iterasi yang dilakukan.
7. Rata-rata keseluruhan nilai akurasi akan menjadi nilai performa model PSO-KNN.

4. HASIL DAN PEMBAHASAN

4.1. Preprocessing Data

Pada penelitian ini, proses *preprocessing* data dilakukan dalam tiga tahap utama, yaitu *data cleaning*, data *transformation*, dan *balancing data*.

1. Data cleaning

Tahap *data cleaning* dilakukan untuk mengatasi masalah nilai *missing value*. *Missing value* adalah nilai yang tidak ada atau tidak terdefinisi dalam dataset. Jumlah *missing value* yang ditemukan sebanyak 277. Contoh dari data yang *missing value* dapat dilihat pada Gambar 2.

Zat padat tersuspensi (TSS)	BOD (20°C. 5 hari)	COD (dichromat)	\
24	NaN	7.0	
14	NaN	11.0	
25	NaN	10.0	
41	NaN	29.0	
147	NaN	26.0	
...	...	...	
29	NaN	4.0	
30	NaN	4.0	
13	NaN	35.0	
55	NaN	4.0	
10	NaN	8.0	

Gambar 2. Data *Missing value*

Untuk menangani masalah ini, dipilih pendekatan penghapusan baris data yang mengandung *missing value*. Pilihan ini didasarkan pada pertimbangan bahwa data time series sulit untuk diimputasi dengan benar karena tidak mungkin mengambil informasi dari waktu sebelumnya atau setelahnya untuk mengisi nilai yang hilang. Dengan menghapus baris data yang mengandung *missing value*, dataset akan menjadi lengkap dan siap untuk analisis lebih lanjut.

2. *Data Transformation*

Pada tahap data *transformation*, perhatian diberikan pada nilai *outlier* dalam dataset. *Outlier* adalah nilai ekstrim yang jauh berbeda dengan sebagian besar data lainnya. Metode *z-score* digunakan untuk mengidentifikasi *outlier* dengan batasan *z-score* lebih dari 3, yang didasarkan pada konvensi umum bahwa nilai yang berjarak tiga deviasi standar atau lebih dari rata-rata cenderung merupakan *outlier*. Hasilnya ditidentifikasi sebanyak 52 data termasuk ke dalam *outlier. C*ontoh nilai *outlier* pada data dapat dilihat pada Gambar 3.

Bakteri Koli Tinja	Bakteri Koli	Indeks Pencemar	Kategori
32000000	25000000	17.48	Cemar Berat
38000000	30000000	17.75	Cemar Berat
39000000	30000000	18.10	Cemar Berat
110000	240000	8.54	Cemar Sedang
41000000000	33000000000	29.15	Cemar Berat
1100000000	780000000	23.31	Cemar Berat
5900000000	3200000000	25.95	Cemar Berat
39000000	79000000	18.05	Cemar Berat
320000000	2900000000	21.54	Cemar Berat
140000	220000	8.54	Cemar Ringan
340000000	610000000	21.46	Cemar Ringan
430000000	1300000000	21.90	Cemar Ringan
58000000000	75000000000	29.00	Cemar Berat
6600000000	8600000000	26.00	Cemar Berat
68000000000	88000000000	29.00	Cemar Berat
82000000000	92000000000	29.00	Cemar Berat
77000000000	84000000000	30.00	Cemar Berat
70000000000	97000000000	30.00	Cemar Berat
76000000000	95000000000	30.00	Cemar Berat
69000000000	86000000000	29.00	Cemar Berat
68000000000	84000000000	29.00	Cemar Berat
68000000000	98000000000	29.00	Cemar Berat

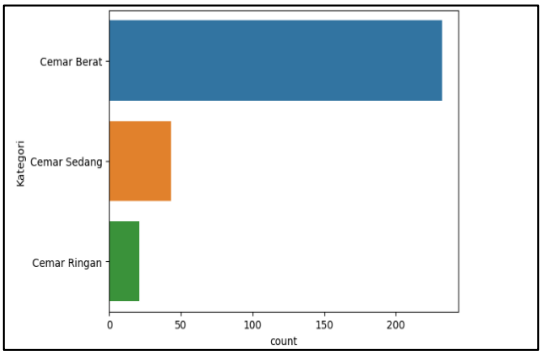
Gambar 3. Data *outlier*

Setelah identifikasi *outlier*, dilakukan transformasi *Box-cox* pada data. Transformasi

ini digunakan untuk mengubah distribusi data mendekati normal dan mengurangi pengaruh nilai-nilai *outlier* yang dapat menyebabkan distorsi pada analisis atau model. Hasilnya nilai *outlier* berkurang menjadi 18 data *outlier*.

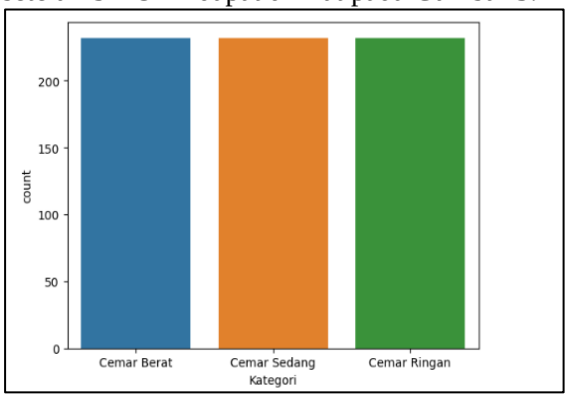
3. *Balancing data*

Tahap *balancing data* dilakukan untuk mengatasi masalah ketidakseimbangan jumlah kelas dalam dataset. Ketidakseimbangan ini menyebabkan Model cenderung memiliki bias terhadap kelas mayoritas dan performanya kurang optimal pada kelas minoritas. Visualiasi jumlah persebaran kelas data dapat dilihat pada Gambar 4.



Gambar 4. Persebaran Jumlah Kelas Data

Untuk menyeimbangkan jumlah data di setiap kelas, digunakan metode *oversampling* dengan algoritma SMOTE (*Synthetic Minority Over-sampling Technique*). Metode ini menghasilkan data sintetis pada kelas minoritas dengan mempertahankan informasi asli kelas tersebut. Hasil persebaran jumlah kelas data setelah SMOTE dapat dilihat pada Gambar 5.



Gambar 5. Persebaran Jumlah Kelas data setelah SMOTE

Penggunaan metode SMOTE membuat variasi data dalam kelas minoritas diperkaya, dan model dapat belajar pola yang lebih baik serta menghasilkan prediksi yang lebih akurat.

Hasil akhir dari proses *preprocessing* adalah dataset yang lengkap tanpa *missing value*, telah ditransformasi untuk mengatasi *outlier*, dan jumlah kelas data telah seimbang. Dataset yang sudah melalui tahap ini berjumlah 577 data, yang siap digunakan untuk melatih model prediksi yang lebih akurat dan tidak terpengaruh oleh masalah ketidakseimbangan atau nilai-nilai *outlier*.

4.2 Model KNN

Dalam penelitian ini, pembuatan model KNN dilakukan dengan pengujian nilai K secara acak. Adapun nilai K yang diuji yaitu K= 3,5,7,9,11,13,15,17,19. Setelah nilai K dipilih, langkah berikutnya adalah menghitung jarak antara data uji dengan data pada dataset pelatihan. Jarak ini dapat dihitung menggunakan berbagai metrik, seperti *Euclidean distance* atau *Manhattan distance*. Pada KNN, umumnya digunakan *Euclidean distance* untuk mengukur jarak antara dua data dalam ruang fitur. Setelah model KNN dibuat dengan nilai K tertentu, langkah terakhir adalah melakukan evaluasi untuk menilai kinerja model. Evaluasi dapat menggunakan berbagai metrik, seperti akurasi, presisi, *recall*, dan lain-lain [16]. Pada penelitian ini, fokus pada nilai akurasi sebagai metrik evaluasi. Akurasi mengukur seberapa tepat model dalam memprediksi kelas data uji. Model KNN kemudian diuji dengan menggunakan dataset uji atau dengan menggunakan teknik *cross-validation* untuk menghindari *overfitting*. Proses ini dilakukan pada setiap nilai K yang diuji. Setelah itu, nilai akurasi masing-masing model dihitung dan dicatat. Akhirnya, nilai rata-rata akurasi untuk setiap nilai K dihitung dan digunakan untuk memilih nilai K terbaik yang memberikan performa tertinggi pada dataset yang diberikan. Nilai Akurasi yang didapatkan pada percobaan setiap nilai K dapat dilihat pada Tabel 1.

Berdasarakan Tabel 1 percobaan, hasil pengujian menunjukkan bahwa nilai rata-rata akurasi terbaik diperoleh pada K=3 dengan akurasi sebesar 77,9%. Nilai K yang dipilih ini memberikan performa yang lebih baik dibandingkan dengan nilai K lainnya yang diuji.

Tabel 1 Nilai Akurasi Tiap Fold

K	Nilai Akurasi tiap Fold										Rata-rata Akurasi (%)
	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Fold-6	Fold-7	Fold-8	Fold-9	Fold-10	
3	0.603	0.724	0.827	0.810	0.741	0.844	0.789	0.807	0.754	0.736	**0.779**
5	0.637	0.655	0.810	0.793	0.706	0.862	0.701	0.789	0.736	0.736	0.743
7	0.603	0.620	0.741	0.775	0.706	0.793	0.701	0.789	0.701	0.754	0.718
9	0.586	0.603	0.741	0.810	0.706	0.793	0.649	0.771	0.719	0.754	0.713
11	0.586	0.603	0.775	0.758	0.689	0.775	0.631	0.719	0.719	0.736	0.699
13	0.517	0.603	0.793	0.741	0.689	0.758	0.614	0.701	0.701	0.754	0.687
15	0.5	0.551	0.775	0.724	0.689	0.758	0.631	0.684	0.684	0.771	0.677
17	0.482	0.5	0.724	0.724	0.672	0.724	0.596	0.736	0.666	0.736	0.656
19	0.5	0.5	0.724	0.672	0.689	0.741	0.596	0.701	0.649	0.771	0.654

4.3 Model PSO-KNN

Pada penelitian ini, model PSO-KNN dibuat dengan tujuan untuk melakukan seleksi fitur (*feature selection*) pada dataset yang terdiri dari 21 atribut (fitur). Proses seleksi fitur dilakukan dengan menggunakan algoritma *Particle swarm optimization* (PSO). Pada awalnya, algoritma PSO akan menginisialisasi sejumlah partikel secara acak dalam ruang pencarian yang mewakili solusi. Setiap partikel akan mewakili sebuah himpunan atribut yang berfungsi sebagai kandidat fitur untuk digunakan dalam model KNN. Setelah partikel diinisialisasi langkah selanjutnya adalah mengevaluasi setiap partikel berdasarkan performa model KNN dengan fitur yang terpilih. Evaluasi dilakukan menggunakan 10-fold *cross validation*, di mana dataset akan dibagi menjadi 10 subset (*fold*) dengan ukuran yang seimbang. Model KNN akan dilatih dan diuji pada setiap fold, dan nilai akurasi pada tiap fold akan dihitung.

Setelah evaluasi dilakukan, setiap partikel akan memperbarui posisinya berdasarkan hasil evaluasi. Setiap partikel akan mencoba mencari solusi yang lebih baik dengan bergerak di sekitar posisi saat ini. Partikel akan bergerak menuju posisi yang memiliki performa model KNN yang lebih baik. Selain memperbarui posisi individu, algoritma PSO juga akan mencatat posisi terbaik yang pernah ditemukan oleh seluruh partikel dalam ruang pencarian. Posisi global terbaik ini mewakili solusi terbaik yang ditemukan selama proses iterasi. Proses ini dilakukan hingga iterasi yang ditentukan, dimana dalam penelitian ini dilakukan sebanyak 50 iterasi. Setelah 50 iterasi, proses PSO akan menghasilkan partikel terbaik yang mewakili solusi terbaik untuk seleksi fitur. Atribut yang terpilih dalam partikel terbaik merupakan atribut-atribut optimal yang memberikan performa model KNN dengan akurasi yang paling baik. Dalam penelitian ini, didapatkan 8 atribut terpilih dari total 21 atribut yang ada. Adapun kedelapan atribut terpilih tersebut dapat dilihat pada Tabel 2.

Tabel 2 Atribut Terpilih

NO	Atribut Terpilih
1.	BOD (20°C. 5 hari)
2.	COD (dichromat)
3.	Nitrat
4.	Crom Hexavalen (Cr6+)
5.	Tembaga (Cu)
6.	Flourida
7.	Bakteri Koli
8.	Indeks Pencemar

Hasil dari proses pembuatan model PSO-KNN ini adalah model KNN yang menggunakan 8 atribut terpilih untuk melakukan klasifikasi data. Nilai akurasi dari dapat dilihat pada Tabel 3.

Tabel 3 Nilai Iterasi

Nilai K-Fold	Rata-rata sepanjang iterasi
1	**0.7931034482758621**
2	**0.9137931034482759**
3	**1.0**
4	**0.9827586206896551**
5	**1.0**
6	**0.9655172413793104**
7	**0.9655172413793104**
8	**1.0**
9	**0.9649122807017544**
10	**0.9824561403508771**
Rata-rata akurasi model (%)	**0.9584996975196611**

Berdasarkan tabel di atas, Model ini mencapai rata-rata akurasi sebesar 95,8% berdasarkan evaluasi menggunakan 10-fold *cross validation*.

4.4 Perbandingan Model KNN dan PSO-KNN

Perbandingan akurasi model KNN dan PSO-KNN dapat dilhat pada Tabel 4.

Tabel 4 Perbandinga nilai KNN dan KNN-PSO

	Model KNN	**Model PSO-KNN**
Nilai Akurasi Model	77,9%	95,8%

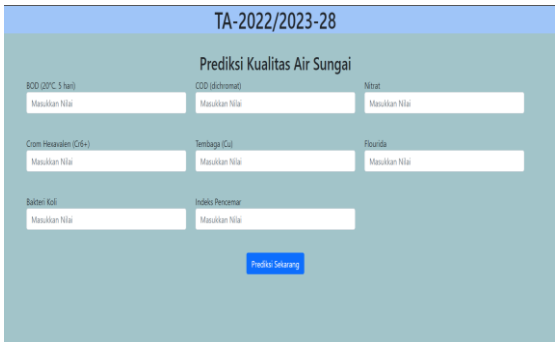
Perbandingan nilai akurasi model KNN dan model PSO-KNN pada Tabel 4 , terlihat bahwa model PSO-KNN memiliki akurasi yang jauh lebih tinggi (95,8%) dibandingkan dengan model KNN (77,9%). Perbedaan akurasi yang signifikan ini menunjukkan bahwa penggunaan algoritma PSO untuk melakukan seleksi fitur dalam KNN telah memberikan peningkatan kinerja pada model klasifikasi. Model KNN menggunakan seluruh fitur yang ada dalam dataset untuk melakukan klasifikasi. Hal ini dapat menyebabkan beberapa fitur yang kurang relevan atau noise turut mempengaruhi proses klasifikasi, sehingga akurasi model menjadi terbatas. Sementara itu, model PSO-KNN telah melakukan seleksi fitur menggunakan algoritma PSO. Algoritma ini telah mencari kombinasi fitur yang paling optimal untuk digunakan dalam model KNN. Hasilnya, hanya 8 atribut terpilih dari 21 atribut total yang dianggap paling relevan dan memberikan kontribusi signifikan dalam klasifikasi. Dengan demikian, penggunaan fitur yang lebih relevan dan informatif dalam PSO-KNN membantu meningkatkan kemampuan model untuk memahami pola yang lebih baik, sehingga akurasi prediksi menjadi lebih tinggi. Selain itu dalam model KNN, semakin banyak fitur yang digunakan, semakin tinggi kompleksitas komputasinya. Jika jumlah fitur sangat besar, proses klasifikasi menjadi lebih lambat dan memerlukan sumber daya komputasi yang lebih besar. Dalam PSO-KNN, karena hanya menggunakan fitur-fitur terpilih, kompleksitas komputasi menjadi lebih rendah. Proses seleksi fitur dengan PSO juga memiliki tingkat efisiensi yang cukup baik, sehingga memungkinkan pengolahan data menjadi lebih cepat dan efisien.

Adanya perbedaan akurasi yang signifikan dan efisiensi yang lebih tinggi, model PSO-KNN dapat dianggap sebagai pilihan yang lebih baik dibandingkan dengan model KNN konvensional dalam konteks penelitian ini Penggunaan algoritma PSO untuk seleksi fitur telah membantu meningkatkan performa dan kinerja model, sehingga dapat memberikan hasil prediksi yang lebih akurat dan lebih handal untuk aplikasi klasifikasi pada dataset yang diberikan

4.5 Model Terbaik sebagai Prototipe Aplikasi

Setelah perbandingan antara kedua model, maka model PSO-KNN yang akan dikembangkan menjadi prototipe aplikasi prediksi kualitas air sungai. Dalam pembutan aplikasi, tampilan aplikasi dibuat dengan menampilkan 8 atribut terpilih yang akan diisi oleh pengguna. Aplikasi menyediakan 8 atribut field yang harus diisi pengguna untuk melakukan prediksi. Model PSO-KNN akan didefinisikan dalam aplikasi *flask*. Model ini akan menerima data masukan, melakukan klasifikasi berdasarkan atribut terpilih, dan menghasilkan prediksi mengenai kualitas air sungai. Hasil prediksi yang diperoleh dari model PSO-KNN akan dikirim kembali ke tampilan HTML dan ditampilkan ke pengguna. Pengguna akan melihat hasil prediksi berupa label atau kelas yang menggambarkan kualitas air sungai berdasarkan atribut terpilih yang dimasukkan. Tampilan aplikasi prediksi

kualitas air sungai dapat dilihat pada Gambar 6 dan Gambar 7.



Gambar 6. Tampilan awal aplikasi

Gambar 7. Tampilan Hasil Prediksi

5. KESIMPULAN

Pada penelitian ini implementasi algoritma KNN dengan optimasi PSO berhasil mendapatkan atribut optimal sebanyak 8 atribut dari total 21 atribut yang digunakan dalam model prediksi kualitas air sungai di jakarta. Hasil dari optimasi PSO ini membantu menentukan atribut yang yang paling berpengaruh dari masing masing atribut yang ada dalam dataset yang digunakan. Dimana hal ini dapat meningkatkan kinerja dan akurasi model KNN dalam menyelesaikan tugas klasifikasi atau prediksi. Perbandingan antara model KNN dengan model KNN berbasis PSO dalam penelitian ini menunjukkan bahwa model PSO-KNN memberikan kinerja yang lebih baik. Hal ini dapat dilihat dari nilai evaluasi model KNN diperoleh sebesar 77,9 % dan pada model PSO-KNN diperoleh sebesar 95,8%. Dari perbandingan tersebut, terlihat bahwa nilai akurasi PSO-KNN lebih besar dibandingkan dengan model KNN. Model PSO-KNN mampu mengoptimalkan seleksi fitur dan meningkatkan akurasi prediksi kualitas air sungai.

DAFTAR PUSTAKA

[1] T. Hartono, S. Wang, Q. Ma, and Z. Zhu, "Layer structured graphite oxide as a novel adsorbent for humic acid removal from aqueous solution," Journal of Colloid and Interface Science, vol. 333, no. 1, pp. 114–119, Feb. 2009. [Online]. Available: https://doi.org/10.1016/j.jcis.2009.02.005

[2] Zeleňák, K., Krajina, A., Meyer, L., Fiehler, J., Behme, D., Bulja, D., Caroff, J., & Chotai, A. A. (2021). How to Improve the Management of Acute Ischemic Stroke by Modern Technologies, Artificial Intelligence, and New Treatment Methods. *Life*, *11*(6). https://doi.org/10.3390/life1106048

[3] S. Hermayeni, "Penerapan Metode Modified *K-Nearest neighbor*," Universitas Islam Negeri Sultan Syarif Kasim Riau, 2021.

[4] Yusra, R. N., Sitompul, O. S., & Sawaluddin. (2021). Kombinasi K-Nearest Neighbor (KNN) dan Relief-F Untuk Meningkatkan Akurasi Pada Klasifikasi Data. *InfoTekJar: Jurnal Nasional Informatika Dan Teknologi Jaringan*, *1*, 0–5.

[5] García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: methods and prospects. *Big Data Analytics*, *1*(1), 1–22. https://doi.org/10.1186/s41044-016-0014-0

[6] García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: methods and prospects. *Big Data Analytics*, *1*(1), 1–22. https://doi.org/10.1186/s41044-016-0014-0

[7] Vierheller, J. (2014). Exploratory data analysis. *Communications in Computer and Information Science*, *500*, 110–126. https://doi.org/10.1007/978-3-662-45006-2_9

[8] Fatiya. (2022). Pengaruh Synthetic Minority Oversampling Technique pada Analisis Sentimen Menggunakan Algoritma K-Nearest Neighbors. *Jlk*, *5*(1), 7–12. https://github.com/riochr17/Analisis-Sentimen-ID

[9] Anguita. (2012). The 'K' in K-fold cross validation. *ESANN 2012 Proceedings, 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, *April*, 441–446.

[10] Saifur, R., Handayani, T., Prathivi, R., & Ardianita, T. (2021). Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa. *IJCIT (Indonesian Journal on Computer and Information Technology)*, *6*(2), 118–127.

[11] Eberhart. (2004). Feature Article Particle Swarm Optimization Feature Article (Cont .). *Neural Networks*, *February*, 69–73.

[12] Townsend. (1971). Erratum to: Theoretical analysis of an alphabetic confusion matrix. *Perception & Psychophysics*, *10*(4), 256. https://doi.org/10.3758/BF03212817

[13] Fadillah, I. J., & Muchlisoh, S. (2017). Perbandingan Metode Hot-Deck Imputation dan Metode KNN dalam Mengatasi Missing Value Penerapan Pada Data Susenas Maret Tahun 2017. *Seminar Nasional Official Statistics.*, *2017*(March), 275–285.

[14] Fadillah, I. J., & Muchlisoh, S. (2017). Perbandingan Metode Hot-Deck Imputation dan Metode KNN dalam Mengatasi Missing Value Penerapan Pada Data Susenas Maret Tahun 2017. Seminar Nasional Official Statistics., 2017(March), 275–285.

[15] Rosyidi, A., Ginardi, R. V. H., & Munif, A. (2017). Implementasi Metode K-Nearest Neighbor Untuk Penentuan Lokasi Pos Hujan Terdekat Dengan Titik Rute Perjalanan Pada Aplikasi Clearroute. Jurnal Teknik ITS, 6(2), 1–4. https://doi.org/10.12962/j23373539.v6i2.23581

[16] Putra, P., M. H. Pardede, A., & Syahputra, S. (2022). Analisis Metode K-Nearest Neighbour dalam Klasifikasi Data Iris Bunga. *Jurnal Teknik Informatika Kaputama (JTIK)*, 6(1), 297–305.