

PENERAPAN ALGORITMA K-NEAREST NEIGHBOR (K-NN) UNTUK KLASIFIKASI Status Monitoring Automatic Pump Water Machine STUDI KASUS : INDUSTRI MANUFAKTUR

Adlian Jefiza, Indra Mora

Politeknik Negeri Batam, Batam, Indonesia

INFORMASI ARTIKEL	ABSTRAK
<p>Sejarah Artikel: Diterima: Juni 2025 Revisi: Juni 2025 Diterima: Juli 2025 Dipublikasi: Juli 2025</p> <p>Kata Kunci: Sistem Industri, Klasifikasi, K-Nearest Neighbor (KNN)</p> <p>*Penulis Korespondensi: indramora2002@gmail.com</p> <p>Keyword: Industri System, Classification, K-Nearest Neighbor (KNN)</p>	<p>Dalam dunia industri modern, pemantauan kondisi sistem secara real-time menjadi hal yang krusial untuk menjaga efisiensi dan mencegah kerusakan peralatan. Penelitian ini bertujuan untuk mengklasifikasikan kondisi sistem industri berdasarkan data sensor menggunakan algoritma K-Nearest Neighbors (KNN). Data yang digunakan terdiri dari empat parameter utama yaitu tekanan (pressure), laju aliran (flow rate), tegangan (voltage), dan putaran mesin (RPM), yang kemudian diklasifikasikan ke dalam tiga kondisi: Alert, Critical, dan Normal. Proses preprocessing dilakukan dengan normalisasi Min-Max dan pembagian data menjadi data latih dan uji. Hasil evaluasi menunjukkan bahwa metode KNN mampu mencapai akurasi sebesar 58% dengan nilai mean squared error (MSE) sebesar 1.06 dan rata-rata akurasi validasi silang sebesar 64%. Hasil ini menunjukkan bahwa KNN cukup efektif digunakan sebagai metode awal untuk deteksi kondisi sistem industri, meskipun performa klasifikasi untuk kategori Critical masih dapat ditingkatkan.</p> <p>ABSTRACT</p> <p><i>In the modern industrial world, real-time monitoring of system conditions is crucial to maintain efficiency and prevent equipment damage. This research aims to classify industrial system conditions based on sensor data using the K-Nearest Neighbors (KNN) algorithm. The data used consists of four main parameters namely pressure, flow rate, voltage, and engine speed (RPM), which are then classified into three conditions: Alert, Critical, and Normal. Preprocessing is done with Min-Max normalization and division of data into training and test data. The evaluation results show that the KNN method is able to achieve an accuracy of 58% with a mean squared error (MSE) value of 1.06 and an average cross-validation accuracy of 64%. These results show that KNN is effective enough to be used as an initial method for industrial system condition detection, although the classification performance for the Critical category can still be improved.</i></p>

PENDAHULUAN

Perkembangan industri modern menuntut adanya sistem yang andal dan efisien dalam mendeteksi kondisi operasional peralatan secara otomatis. Sistem yang tidak terpantau dengan baik dapat mengalami kerusakan tiba-tiba, menyebabkan penurunan produktivitas, biaya perbaikan yang tinggi, bahkan potensi kecelakaan kerja.

Dengan kemajuan teknologi sensor, data seperti tekanan, aliran fluida, tegangan, dan kecepatan putaran mesin kini dapat dikumpulkan secara real-time. Namun, interpretasi data dalam jumlah besar memerlukan pendekatan analitik yang cerdas. Salah satu solusi yang banyak digunakan dalam pengolahan data sensor adalah penerapan algoritma kecerdasan buatan (AI),

khususnya metode klasifikasi.

K-Nearest Neighbors (KNN) adalah algoritma klasifikasi yang sederhana namun efektif. KNN bekerja dengan cara mencari sejumlah tetangga terdekat dari suatu data baru berdasarkan jarak fitur, kemudian menentukan kelas mayoritas dari tetangga-tetangga tersebut. Algoritma ini tidak memerlukan pelatihan model secara eksplisit, sehingga cocok diterapkan pada sistem dengan struktur data yang dinamis dan kompleks seperti sistem industri.

Penelitian ini bertujuan untuk menerapkan algoritma KNN dalam mengklasifikasikan kondisi sistem industri berdasarkan empat parameter utama yaitu tekanan (pressure), laju aliran (flow rate), tegangan (voltage), dan kecepatan putaran mesin (RPM). Kondisi sistem dikategorikan ke dalam tiga kelas: Alert, Critical, dan Normal. Evaluasi kinerja model dilakukan dengan menggunakan metrik akurasi, confusion matrix, f1-score, mean squared error (MSE), serta validasi silang (cross-validation).

Dalam penelitian ini, algoritma K-Nearest Neighbors (KNN) digunakan sebagai metode klasifikasi utama. KNN merupakan algoritma yang bekerja berdasarkan prinsip kesamaan data, di mana data baru akan diklasifikasikan berdasarkan mayoritas kelas dari tetangga terdekatnya dalam ruang fitur. Algoritma ini memiliki keunggulan dalam hal kesederhanaan implementasi, tidak memerlukan pelatihan model yang kompleks, dan memiliki performa yang cukup baik dalam klasifikasi berbasis jarak.

Pemilihan KNN dalam penelitian ini didasarkan pada sifat data sensor industri yang bersifat numerik dan terstruktur, serta kebutuhan akan pendekatan klasifikasi yang cepat dan responsif. Evaluasi model dilakukan menggunakan beberapa metrik pengukuran performa seperti confusion matrix, precision, recall, f1-score, mean squared error (MSE), dan validasi silang (cross-validation).

Melalui penelitian ini, diharapkan dapat diperoleh gambaran yang jelas mengenai efektivitas metode KNN dalam mendeteksi kondisi sistem industri, serta menjadi referensi awal dalam pengembangan sistem pemantauan berbasis kecerdasan buatan untuk kebutuhan industri ke depan].

LANDASAN TEORI

A. Mesin Industri

Penerapan kecerdasan buatan dalam dunia industri telah mengalami perkembangan pesat, khususnya dalam hal pemantauan kondisi sistem dan prediksi kegagalan mesin. Sejumlah penelitian telah menunjukkan bahwa pengolahan data sensor secara cerdas dapat meningkatkan efektivitas pemeliharaan (predictive maintenance) dan mengurangi waktu henti produksi (downtime).

B. Klasifikasi

Klasifikasi adalah proses kategorisasi data dengan memasukkan data tersebut ke dalam kategori tertentu dari kategori yang tersedia [2]. Teknik dalam klasifikasi ialah melakukan kategorisasi data training untuk membuat suatu model untuk melakukan kategorisasi pada data testing yang tersedia.

C. Data Mining

Data Mining berasal dari dua kata bahasa inggris yaitu “data” yang berarti data dan “mining” yang berarti menambang. Sehingga data mining dapat diartikan sebagai proses penambangan data. Data mining merupakan metode untuk mendapatkan inti dari suatu ilmu atau pengetahuan [6]. Data mining menghasilkan teknik pengenalan pola data yang dapat memberikan perbedaan hasil dari data lain, sehingga dapat digunakan di masa yang akan datang [7].

D. Algoritma K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) adalah metode klasifikasi terhadap objek baru berdasarkan data training yang memiliki jarak tetangga terdekat (nearest neighbor) dengan objek baru tersebut

[8]. Dekat atau jauhnya neighbor biasanya dihitung berdasarkan jarak Euclidean. Berikut ini adalah langkah-langkah algoritma KNN:

1. Menentukan nilai K. Nilai K dapat dihitung menggunakan persamaan 1 berikut ini :

$$k = \sqrt{N} \quad (1)$$

N merupakan banyaknya sampel pada data training

2. Melakukan perhitungan nilai jarak (euclidean distance) terhadap masing-masing objek data yang diberikan. Rumus untuk menghitung euclidean distance dapat dilihat pada persamaan 2.

$$di = \sqrt{(x_{ki} - x_{kj})^2 + (x_{ki} - x_{kj})^2 + \dots + (x_{ki} - x_{kj})^2} \quad (2)$$

Keterangan :

di = jarak euclidean

x_{ki} = data training ke-1

x_{kj} = data testing ke-1

3. Melakukan pengelompokkan data sesuai dengan perhitungan jarak (Euclidean distance)
4. Melakukan pengelompokkan data sesuai dengan nilai tetangga terdekat (nearest neighbor) atau berdasarkan data yang mempunyai jarak Euclidean terkecil.
5. Memilih nilai mayoritas dari tetangga terdekat sebagai hasil klasifikasi.

E. Bahasa Pemrograman Python

Dalam website docs.python.org, Python adalah bahasa pemrograman yang dapat digunakan untuk analisis data, mudah digunakan dan berorientasi objek. Python dapat digunakan dalam beberapa sistem operasi termasuk Linux dan macOS, dan Windows.

F. Confusion Matrix

Confusion Matrix mempresentasikan prediksi dengan membandingkan nilai asli dengan nilai hasil prediksi model untuk menghasilkan penilaian seperti berikut ini [9]:

1. Akurasi

Akurasi merupakan ketepatan antara nilai prediksi dengan nilai aktual. Rumus untuk menghitung nilai akurasi dapat dilihat pada persamaan 3.

$$Akurasi = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (3)$$

2. Presisi

Presisi merupakan tingkat keberhasilan model dalam memberikan jawaban dengan tepat kepada pengguna. Rumus untuk menghitung nilai presisi dapat dilihat pada persamaan 4.

$$Presisi = \frac{TP}{TP+FP} \quad (4)$$

3. Recall

Recall merupakan tingkat keberhasilan model dalam menemukan kembali informasi dengan benar. Rumus untuk menghitung nilai recall dapat dilihat pada persamaan 5.

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

4. F1-Score atau F-Measure.

F1- Score merupakan hasil perbandingan antara nilai presisi dan recall. Rumus untuk menghitung nilai f1-Score dapat dilihat pada persamaan 6.

$$F1 - Score = \frac{2 \times Presisi \times Recall}{Presisi+Recall} \quad (6)$$

Keterangan:

TP : True Positive

TN : True Negative

FP : False Positive

FN : False Negative.

METODE PENELITIAN

A. Tahapan Awal

Penelitian ini dilakukan dalam beberapa tahapan utama, yaitu pengumpulan data, preprocessing data, ekstraksi fitur, pemodelan dengan algoritma K-Nearest Neighbors (KNN), serta evaluasi performa model.

B. Tahap Pengumpulan Data

Pengumpulan data merupakan tahap awal yang krusial dalam penelitian ini karena kualitas data sangat mempengaruhi akurasi model klasifikasi. Data yang digunakan diperoleh dari sistem industri yang memantau performa mesin atau peralatan melalui sensor-sensor terintegrasi. Sensor tersebut merekam parameter operasional utama secara berkala dalam bentuk data numerik.

Adapun parameter yang dikumpulkan sebanyak 1000 data meliputi:

1. Pressure (tekanan): menunjukkan tekanan sistem dalam satuan tertentu,
2. FlowRate (laju aliran): menunjukkan seberapa cepat cairan/gas mengalir dalam sistem,
3. Voltage (tegangan): mengindikasikan kestabilan suplai listrik ke mesin,
4. RPM (Revolutions Per Minute): kecepatan putaran mesin yang menjadi indikator performa.

Pressure	FlowRate	Voltage	RPM	SystemCondition
0,047204161	0,137291863	0,818106818	0,712643528	Alert
0,054816988	0,018436635	0,82947704	0,720717158	Alert
0,451332772	0,001350802	0,908411093	0,725785928	Alert
0,075030517	0,128780504	0,806918537	0,738437669	Alert
0,827403202	0,011885335	0,091821096	0,526374717	Alert
0,525560068	0,015951864	0,903766983	0,656901113	Alert
0,075311426	0,010643892	0,091689883	0,451671286	Alert
0,007939661	0,130790012	0,855283579	0,503841212	Alert
0,071780814	0,197807219	0,917939108	0,585440052	Alert
0,065442157	0,175965997	0,784839413	0,775962433	Alert
0,082041839	0,107514941	0,082465627	0,542037259	Alert
0,069806186	0,042227357	0,805967509	0,630397661	Critical
0,050370006	0,159578625	0,859747993	0,805914921	Critical
0,06138949	0,058745411	0,008071371	0,578989447	Critical
0,052136123	0,001963547	0,080814949	0,096318224	Critical
0,052590751	0,139618225	0,843347698	0,079803958	Critical
0,91459327	0,203336024	0,845554824	0,810054239	Critical
0,007522326	0,01391461	0,094487919	0,537046	Critical
0,089955221	0,187842744	0,930342435	0,531014603	Critical
0,059596578	0,650072587	0,907036689	0,443530399	Critical
0,414618224	0,186923279	0,945411381	0,936078232	Critical
0,451031141	0,001413516	0,778543905	0,079608155	Critical
0,4442007	0,256813251	0,870842772	0,817594707	Critical
0,071289832	0,911190273	0,927064728	0,69072727	Critical

0,345172213	0,171963836	0,827120216	0,556325812	Normal
0,796236547	0,020141898	0,837785089	0,601620607	Normal
0,544697674	0,136225654	0,849784865	0,571688507	Normal
0,737877707	0,242452703	0,851435562	0,660222691	Normal
0,066396005	0,143094326	0,837534339	0,264795991	Normal
0,066709605	0,126436257	0,841122944	0,047289864	Normal
0,062408516	0,130536367	0,86129945	0,559378854	Normal
0,007792379	0,013092275	0,850284703	0,050609789	Normal
0,595434694	0,150200527	0,855717747	0,601889735	Normal
0,288503691	0,158075778	0,801714705	0,45089005	Normal
0,064120794	0,170220722	0,867369233	0,649449839	Normal
0,054526595	0,013581135	0,846805841	0,595162296	Normal
0,058017241	0,140603505	0,889013436	0,354722752	Normal
0,550576705	0,206488438	0,08585342	0,621212382	Normal
0,738371559	0,212131623	0,846101634	0,650557844	Normal
0,000115585	0,144783503	0,840454947	0,644771131	Normal

Tabel 1. Data Training Sebelum Normalisasi

Data dikumpulkan dalam bentuk file digital (.xlsx) dengan masing-masing baris merepresentasikan satu pengamatan pada waktu tertentu. Selain itu, setiap data disertai label kategori kondisi sistem, yaitu:

0 = Alert.

1 = Critical.

2 = Normal.

Label ini ditentukan berdasarkan standar ambang batas sensor atau hasil inspeksi teknisi. Proses labeling dilakukan secara manual pada tahap awal dan divalidasi oleh pihak teknis untuk menjamin akurasi.

Setelah data terkumpul, dilakukan proses seleksi untuk memastikan tidak ada data yang kosong (missing values) atau duplikat. Hanya data yang lengkap dan valid yang digunakan dalam proses pemodelan selanjutnya.

Pressure	FlowRate	Voltage	RPM	SystemCondition
0,047204161	0,137291863	0,818106818	0,712643528	0
0,054816988	0,018436635	0,82947704	0,720717158	0
0,451332772	0,001350802	0,908411093	0,725785928	0
0,075030517	0,128780504	0,806918537	0,738437669	0
0,827403202	0,011885335	0,091821096	0,526374717	0
0,525560068	0,015951864	0,903766983	0,656901113	0
0,075311426	0,010643892	0,091689883	0,451671286	0
0,007939661	0,130790012	0,855283579	0,503841212	0
0,071780814	0,197807219	0,917939108	0,585440052	0
0,065442157	0,175965997	0,784839413	0,775962433	0
0,082041839	0,107514941	0,082465627	0,542037259	0
0,057746975	0,193745984	0,859982877	0,765062547	0
0,069806186	0,042227357	0,805967509	0,630397661	1
0,050370006	0,159578625	0,859747993	0,805914921	1
0,06138949	0,058745411	0,008071371	0,578989447	1
0,052136123	0,001963547	0,080814949	0,096318224	1
0,052590751	0,139618225	0,843347698	0,079803958	1
0,91459327	0,203336024	0,845554824	0,810054239	1
0,007522326	0,01391461	0,094487919	0,537046	1
0,089955221	0,187842744	0,930342435	0,531014603	1
0,059596578	0,650072587	0,907036689	0,443530399	1
0,414618224	0,186923279	0,945411381	0,936078232	1
0,451031141	0,001413516	0,778543905	0,079608155	1
0,4442007	0,256813251	0,870842772	0,817594707	1
0,071289832	0,911190273	0,927064728	0,69072727	1
0,446526372	0,383086395	0,863905269	0,477291565	1
0,079007635	0,175802853	0,801286421	0,007766848	1

0,345172213	0,171963836	0,827120216	0,556325812	2
0,796236547	0,020141898	0,837785089	0,601620607	2
0,544697674	0,136225654	0,849784865	0,571688507	2
0,737877707	0,242452703	0,851435562	0,660222691	2
0,066396005	0,143094326	0,837534339	0,264795991	2
0,066709605	0,126436257	0,841122944	0,047289864	2
0,062408516	0,130536367	0,86129945	0,559378854	2
0,007792379	0,013092275	0,850284703	0,050609789	2
0,595434694	0,150200527	0,855717747	0,601889735	2
0,288503691	0,158075778	0,801714705	0,45089005	2
0,064120794	0,170220722	0,867369233	0,649449839	2
0,054526595	0,013581135	0,846805841	0,595162296	2
0,058017241	0,140603505	0,889013436	0,354722752	2
0,550576705	0,206488438	0,08585342	0,621212382	2
0,738371559	0,212131623	0,846101634	0,650557844	2
0,000115585	0,144783503	0,840454947	0,644771131	2

Tabel 2. Data Training Sesudah Normalisasi

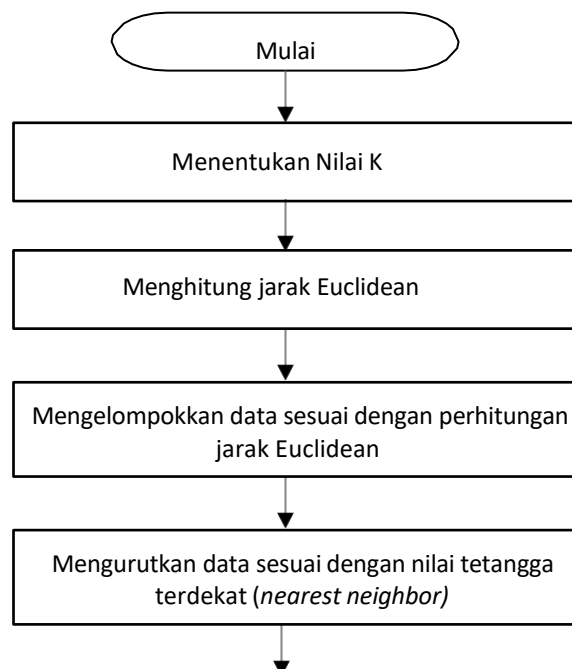
C. Tahapan Data Mining

Dalam penelitian ini, tahapan data mining yang dilakukan dalam pengambilan data yaitu proses pembersihan data yang digunakan untuk menghilangkan data yang tidak diperlukan seperti data duplikat dan data yang tidak lengkap, tahap integrasi data untuk menggabungkan berbagai sumber data untuk menghasilkan data yang benar dan terhindar dari kesalahan, tahap transformasi data untuk mengubah data ke bentuk yang sesuai dengan mining yang dipilih. dan tahap data mining yaitu memilih salah satu dari 7 teknik data mining yaitu Algoritma KNN.

D. Teknik Data Mining

Teknik data mining yang digunakan dalam penelitian ini adalah Algoritma K-Nearest Neighbor (KNN). Algoritma KNN merupakan metode untuk melakukan klasifikasi objek baru berdasarkan data training yang memiliki jarak terdekat (nearest neighbor) dengan objek tersebut. Berikut ini adalah langkah-langkah dalam melakukan klasifikasi menggunakan Algoritma KNN:

1. Menentukan nilai K
2. Melakukan perhitungan nilai jarak (euclidean distance)
3. Melakukan pengelompokan data sesuai dengan perhitungan jarak (euclidean distance)
4. Melakukan pengurutan data sesuai dengan nilai tetangga terdekat (nearest neighbor) atau berdasarkan data yang mempunyai jarak euclidean terkecil.
5. Memilih nilai mayoritas dari tetangga terdekat sebagai hasil klasifikasi.





Gambar 1. Alur Diagram Metode KNN

E. Pengujian Algoritma

Pengujian Algoritma K-Nearest Neighbor (KNN) dilakukan untuk mengetahui tingkat keberhasilan algoritma KNN dalam melakukan klasifikasi terhadap klasifikasi status mesin. Pengujian algoritma KNN pada penelitian ini menggunakan confusion matrix yang diproses dengan bahasa pemrograman python

HASIL DAN PEMBAHASAN

A. Hasil

Algoritma K-Nearest Neighbors (KNN) merupakan salah satu metode klasifikasi yang bekerja berdasarkan prinsip kedekatan atau jarak antar data. Dalam penelitian ini, KNN digunakan untuk mengklasifikasikan kondisi sistem industri ke dalam tiga kategori utama, yaitu: Alert, Critical, dan Normal, berdasarkan hasil pembacaan empat fitur utama dari sensor, yaitu:

1. Pressure: Mengukur tekanan dalam sistem yang mencerminkan beban kerja dan kondisi fluida.
2. FlowRate: Menggambarkan laju aliran cairan atau gas dalam sistem, yang menjadi indikator utama dalam proses industri.
3. Voltage: Menunjukkan kestabilan tegangan listrik yang dipasok ke mesin dan sistem kontrol.
4. RPM (Revolutions Per Minute): Menyatakan kecepatan rotasi dari komponen bergerak seperti motor atau pompa.

Sebelum diterapkan ke dalam model, seluruh data terlebih dahulu melalui tahapan preprocessing, termasuk:

1. Pembersihan data untuk memastikan tidak ada nilai kosong (missing values),
2. Normalisasi menggunakan metode Min-Max Scaling untuk menyamakan skala nilai antar fitur,
3. Dan pemisahan data menjadi data latih (80%) dan data uji (20%) agar model dapat diuji performanya secara adil.

Dalam penerapannya, nilai parameter $k = 5$ dipilih sebagai jumlah tetangga terdekat yang akan digunakan oleh algoritma untuk menentukan kelas dari suatu data baru. Pemilihan nilai ini berdasarkan pendekatan umum yang banyak digunakan untuk dataset dengan ukuran sedang. Selama proses klasifikasi, model KNN akan menghitung jarak Euclidean antara data uji dengan seluruh data latih. Lima data terdekat yang memiliki jarak terkecil akan dipilih, dan kelas terbanyak dari kelima tetangga tersebut akan dijadikan sebagai prediksi kelas untuk data tersebut. Keunggulan dari metode ini adalah kemampuannya dalam memetakan hubungan non-linier antar fitur tanpa perlu asumsi distribusi data. Namun, kekurangannya adalah sensitivitas terhadap skala fitur dan data outlier—sehingga justru menjadikan proses normalisasi sebagai langkah krusial. Setelah proses klasifikasi dilakukan, hasil prediksi dibandingkan dengan data aktual dari kelas masing-masing untuk memperoleh metrik evaluasi seperti akurasi, precision, recall, F1-score, mean squared error (MSE), dan hasil validasi silang (cross-validation).

Berikut ini adalah tahap pengolahan data menggunakan Algoritma KNN dengan data yang telah siap diolah:

1. Menentukan nilai K

Nilai K dapat dicari menggunakan rumus perhitungan akar kuadrat dari data training. Karena jumlah data training sebanyak 1000 data maka nilai K ditentukan sebanyak 5.

2. Melakukan pengurutan data

Dalam proses pengurutan, data akan diurutkan sesuai dengan nilai tetangga terdekat (nearest neighbor) atau berdasarkan data yang mempunyai jarak Euclidean terkecil, seperti yang ditunjukkan pada Tabel 3.

Tabel 3. Pengurutan data berdasarkan nilai tetangga terdekat

Pressure	FlowRate	Voltage	RPM	SystemCondition
0,047204161	0,137291863	0,818106818	0,712643528	0
0,054816988	0,018436635	0,82947704	0,720717158	0
0,451332772	0,001350802	0,908411093	0,725785928	0
0,075030517	0,128780504	0,806918537	0,738437669	0
0,827403202	0,011885335	0,091821096	0,526374717	0
0,525560068	0,015951864	0,903766983	0,656901113	0
0,075311426	0,010643892	0,091689883	0,451671286	0
0,007939661	0,130790012	0,855283579	0,503841212	0
0,071780814	0,197807219	0,917939108	0,585440052	0
0,065442157	0,175965997	0,784839413	0,775962433	0
0,082041839	0,107514941	0,082465627	0,542037259	0
0,057746975	0,193745984	0,859982877	0,765062547	0

3. Melakukan pemilahan data yang sering muncul dari tetangga terdekat sebagai hasil klasifikasi.

Proses terakhir dalam perhitungan menggunakan algoritma KNN adalah memilih nilai yang sering muncul dari mayoritas tetangga terdekat sebagai hasil klasifikasi. Dari data testing sebanyak 800 data.

Tabel 4. Hasil Klasifikasi.

```
Confusion Matrix:
[[57  4 22]
 [29 12  4]
 [21  3 48]]
```

Berdasarkan hasil yang diperoleh, model KNN memberikan akurasi sebesar 58% pada data uji dan 64% pada validasi silang. Meskipun bukan yang tertinggi, hasil ini menunjukkan bahwa KNN mampu membedakan ketiga kelas kondisi sistem industri secara moderat. Nilai precision dan recall yang rendah pada kelas Critical menunjukkan adanya tantangan dalam membedakan kelas ini dari kelas Alert. Hal ini kemungkinan disebabkan oleh pola data yang mirip antara kedua kelas, atau jumlah data Critical yang relatif lebih sedikit. Berikut ini tabel confusion matrix untuk membuktikan perhitungan pada Gambar 2.

Classification Report:				
	precision	recall	f1-score	support
Alert	0.53	0.69	0.60	83
Critical	0.63	0.27	0.38	45
Normal	0.65	0.67	0.66	72
accuracy			0.58	200
macro avg	0.60	0.54	0.54	200
weighted avg	0.60	0.58	0.57	200

Gambar 2. Hasil Perhitungan Akurasi, Presisi, Recall dan F1-Score

Berikut ini perhitungan manual confusion matrix berdasarkan Gambar 2:

(7)

1. Akurasi

$$Akurasi = \frac{(TP + TN)}{(TP + FP + FN + TN)} = \frac{(57 + 67)}{(200)} = 0.62 * 100\% = 62\%$$

2. Presisi

$$Presisi = \frac{TP}{TP + FP} = \frac{57}{107} = 0.53 * 100\% = 53\%$$

3. Recall

$$Recall = \frac{TP}{TP + FN} = \frac{57}{83} = 0.68 * 100\% = 68\%$$

4. F1-Score atau F-Measure

$$F1 - Score = \frac{2 \times Presisi \times Recall}{Presisi + Recall} = \frac{2 \times 0.53 \times 0.68}{0.53 + 0.68} = 2 \times 0.3 * = 0.6 * 100\% = 60\%$$

Hasil evaluasi performa model K-Nearest Neighbors (KNN) dalam mengklasifikasikan kondisi sistem industri menunjukkan nilai-nilai yang menggambarkan efektivitas model terhadap data uji. Akurasi model mencapai 62%, yang berarti bahwa dari seluruh data uji, sebesar 62% prediksi yang dilakukan oleh model sesuai dengan label sebenarnya. Meskipun akurasi merupakan ukuran umum, metrik ini belum sepenuhnya mencerminkan performa pada kasus klasifikasi dengan beberapa kelas, terutama jika distribusi kelas tidak seimbang. Untuk kelas Alert sebagai contoh, nilai presisi mencapai sekitar 53%, menunjukkan bahwa dari seluruh data yang diprediksi sebagai Alert, hanya 53% yang benar-benar termasuk dalam kelas tersebut. Ini mengindikasikan bahwa model masih menghasilkan cukup banyak kesalahan saat memprediksi data sebagai Alert (false positive). Sementara itu, nilai recall untuk kelas yang sama berada di angka 68%, yang artinya dari seluruh data yang sebenarnya Alert, sekitar 68% berhasil dikenali oleh model dengan benar. Nilai ini cukup baik, menunjukkan bahwa model memiliki kecenderungan mengenali sebagian besar data Alert, walaupun tidak semuanya. Nilai F1-score, yang merupakan rata-rata harmonik dari presisi dan recall, berada di sekitar 60% untuk kelas Alert. F1-score menjadi metrik penting dalam mengukur keseimbangan antara presisi dan recall, terutama dalam sistem industri di mana kesalahan klasifikasi bisa berdampak serius pada operasi dan keamanan.

Dengan nilai-nilai tersebut, dapat disimpulkan bahwa model KNN memiliki performa yang cukup baik dalam mengenali pola, namun masih memiliki ruang untuk ditingkatkan, terutama dalam meningkatkan presisi agar lebih mengurangi kesalahan klasifikasi antar kelas. Evaluasi lebih lanjut dan pengujian terhadap parameter lain (seperti nilai k) serta fitur tambahan bisa dilakukan untuk memperbaiki hasil klasifikasi secara keseluruhan.

4. Mendapatkan hasil Mean Squared Error (MSE), Cross Validation Accuracy dan ROC Curve.

Selain menggunakan metrik klasifikasi seperti akurasi, presisi, recall, dan F1-score, evaluasi model K-Nearest Neighbors (KNN) juga dilakukan dengan menghitung Mean Squared Error (MSE) dan akurasi validasi silang (cross-validation accuracy) untuk memberikan gambaran yang lebih menyeluruh mengenai performa model.

MSE adalah metrik evaluasi yang biasanya digunakan dalam regresi, namun juga dapat digunakan untuk mengukur rata-rata kesalahan kuadrat antara label yang sebenarnya dan label yang diprediksi oleh model klasifikasi, terutama ketika label dikodekan dalam bentuk numerik. Dalam konteks ini, label kelas seperti Alert (0), Critical (1), dan Normal (2) diproses sebagai angka. Hasil perhitungan menunjukkan nilai MSE sebesar 1.0600, yang mengindikasikan bahwa rata-rata kesalahan kuadrat model dalam

memprediksi kelas masih cukup tinggi. Nilai MSE yang lebih kecil mengindikasikan bahwa prediksi model mendekati nilai aktual. Dengan tiga kelas bernilai 0, 1, dan 2, nilai MSE di atas 1 menunjukkan bahwa cukup banyak prediksi model yang meleset lebih dari satu kelas (misalnya, memprediksi kelas Normal (2) sebagai Alert (0), atau sebaliknya). Berikut ini tabel confusion matrix untuk membuktikan perhitungan pada Gambar 3.

Mean Squared Error (MSE): 1.0600

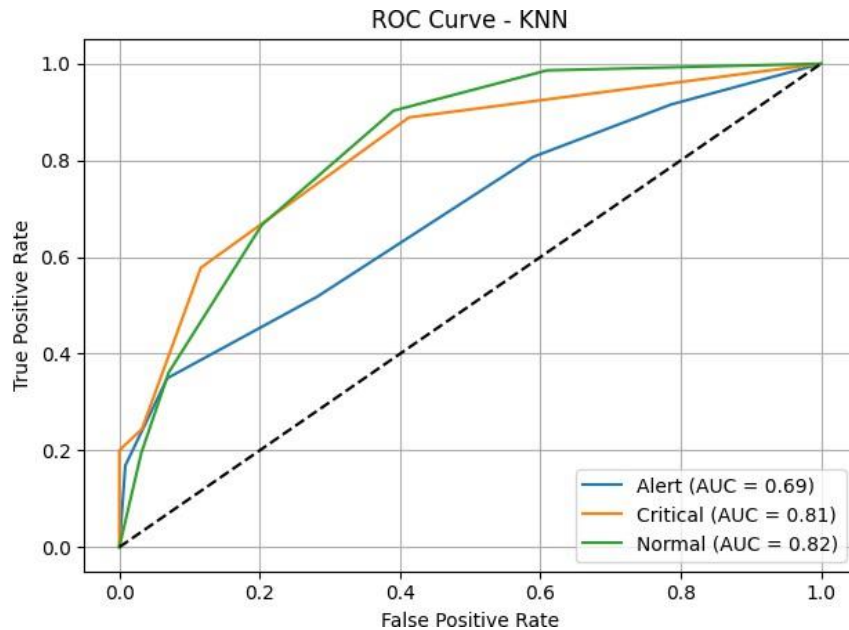
Gambar 3. Hasil Perhitungan Mean Squared Error (MSE)

Cross-validation (validasi silang) merupakan teknik evaluasi yang lebih stabil dan obyektif dibandingkan hanya menggunakan satu kali pembagian data. Pada penelitian ini digunakan teknik 5-fold cross-validation, di mana data dibagi menjadi lima bagian, dan model diuji sebanyak lima kali, masing-masing menggunakan data yang berbeda sebagai data uji dan pelatihan. Hasil validasi silang menunjukkan bahwa model memiliki rata-rata akurasi sebesar 64.00%, dengan deviasi standar $\pm 2.17\%$. Ini berarti bahwa dalam lima pengujian yang berbeda, model secara konsisten mampu mencapai akurasi di sekitar angka tersebut. Deviasi standar yang rendah menunjukkan bahwa model relatif stabil dalam memproses data yang bervariasi.

Cross Validation Accuracy: 0.6400 (+/- 0.0217)

Gambar 4. Hasil Perhitungan Cross Validation Accuracy

Grafik ROC Curve (Receiver Operating Characteristic) di atas menunjukkan performa model klasifikasi KNN dalam membedakan tiga kelas yaitu Alert, Critical, dan Normal. Grafik ini menggambarkan hubungan antara True Positive Rate (TPR) di sumbu vertikal dan False Positive Rate (FPR) di sumbu horizontal. Semakin tinggi kurva mendekati pojok kiri atas, semakin baik kemampuan model dalam melakukan klasifikasi yang benar. Nilai AUC (Area Under the Curve) digunakan untuk mengukur luas di bawah kurva, yang mencerminkan seberapa baik model dapat membedakan antar kelas. Nilai AUC berkisar antara 0 hingga 1, di mana nilai 1 menunjukkan performa sempurna, sedangkan nilai 0,5 menunjukkan performa sebanding dengan tebakan acak. Berdasarkan grafik tersebut, kelas Normal memiliki nilai AUC tertinggi yaitu 0.82, yang menunjukkan bahwa model KNN memiliki kinerja yang sangat baik dalam mengenali data dengan label Normal. Selanjutnya, kelas Critical memiliki AUC sebesar 0.81 yang juga menunjukkan performa yang baik. Namun, untuk kelas Alert, nilai AUC hanya 0.69, yang berarti kemampuan model dalam membedakan kelas ini masih tergolong sedang dan perlu ditingkatkan. Garis putus-putus diagonal dalam grafik menunjukkan batas klasifikasi acak ($AUC = 0.5$); semakin jauh kurva dari garis ini, semakin baik kinerja model. Secara keseluruhan, model KNN bekerja cukup baik terutama untuk kelas Critical dan Normal, tetapi kurang optimal dalam mengenali kelas Alert. Hal ini bisa disebabkan oleh jumlah data kelas Alert yang sedikit atau karakteristik fitur yang tumpang tindih dengan kelas lain, sehingga menyulitkan model dalam membedakannya secara efektif.



Gambar 5. Hasil ROC Curve - KNN

5. Pembahasan

Pada penelitian ini, peneliti menemukan 2 hal yang dapat mempengaruhi hasil dari Algoritma KNN antara lain:

a. Pembagian data training dan data testing

Nilai akurasi Algoritma KNN dapat berubah sesuai dengan struktur data training dan data testing [12]. Oleh karena itu, perlu membagi data training dan data testing dengan tepat. Pada penelitian ini, peneliti mencoba membagi data testing sebanyak 25% dari jumlah data sebanyak 250 mendapatkan hasil akurasi sebesar 53%. Sedangkan jika pembagian data testing sebanyak 20% mendapatkan hasil akurasi sebesar 60%. Oleh karena itu, peneliti membagi data testing sebanyak 20% agar mendapatkan nilai akurasi yang tinggi yaitu sebesar 60%.

b. Penentuan nilai K

Dalam algoritma K-Nearest Neighbors (KNN), pemilihan nilai K yang optimal sangat berpengaruh terhadap kinerja klasifikasi model. Nilai K menentukan seberapa banyak tetangga terdekat yang digunakan dalam pengambilan keputusan untuk mengklasifikasikan suatu data. Oleh karena itu, penentuan nilai K harus melalui proses pengujian dan evaluasi terhadap data yang tersedia. Pada penelitian ini, dilakukan pengujian terhadap 1000 data, yang dibagi menjadi 800 data latih (training data) dan 200 data uji (testing data). Melalui eksperimen menggunakan bahasa pemrograman Python, dilakukan uji coba dengan berbagai nilai K untuk menemukan nilai optimal yang menghasilkan akurasi terbaik.

Dari hasil uji, diperoleh bahwa penggunaan nilai K = 5 menghasilkan hasil klasifikasi yang cukup baik. Evaluasi model menunjukkan:

1. Confusion Matrix menggambarkan distribusi prediksi model terhadap tiga kelas utama (Alert, Critical, dan Normal).
2. Model menghasilkan akurasi klasifikasi sebesar 58% pada data uji.
3. Hasil evaluasi tambahan menunjukkan nilai Mean Squared Error (MSE) sebesar 1.06, dan nilai akurasi rata-rata dari cross-validation sebesar 64% dengan deviasi standar $\pm 2.17\%$.

Berdasarkan pengamatan ini, meskipun nilai $K = 5$ sudah memberikan performa yang cukup stabil dalam pengujian awal, hasil akurasi dan kesalahan klasifikasinya masih dapat ditingkatkan. Oleh karena itu, pada pengembangan lebih lanjut, nilai K sebaiknya diuji lebih lanjut dalam rentang nilai yang lebih luas (misalnya $K = 3$ hingga $K = 25$) untuk mencari nilai K optimal. Nilai MSE yang relatif tinggi mengindikasikan bahwa beberapa klasifikasi masih meleset lebih dari satu label kelas, yang dalam sistem industri bisa menjadi krusial. Dengan demikian, dalam konteks data ini, nilai $K = 5$ dipilih sebagai titik awal yang representatif, namun tetap terbuka untuk penyesuaian lebih lanjut berdasarkan analisis error dan kebutuhan performa klasifikasi yang lebih tinggi.

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma K-Nearest Neighbors (KNN) mampu digunakan secara efektif untuk mengklasifikasikan kondisi sistem industri ke dalam tiga kategori utama, yaitu Alert, Critical, dan Normal. Proses dimulai dari pengumpulan dan normalisasi data, diikuti dengan pembagian data menjadi data latih dan data uji. Dengan menggunakan empat fitur utama yaitu Pressure, FlowRate, Voltage, dan RPM, model KNN dengan nilai parameter $K = 5$ berhasil mencapai akurasi klasifikasi sebesar 58% pada data uji. Selain itu, hasil evaluasi model menggunakan Confusion Matrix, Classification Report, dan Cross-Validation menunjukkan bahwa meskipun performa model tergolong sedang, nilai cross-validation accuracy sebesar 64% dan MSE sebesar 1.06 mengindikasikan bahwa model masih dapat dikembangkan lebih lanjut untuk mencapai akurasi yang lebih optimal.

Daftar Pustaka

- [1] R. Hartanto, "Analisis Deteksi Kondisi Sistem Industri Menggunakan Algoritma K-Nearest Neighbors," Proyek Ujian Akhir Semester, 2025.
- [2] D. P. Utomo dan M. Mesran, "Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut Pada Data Set Penyakit Jantung," J. Media Inform. Budidarma, vol. 4, no. 2, p. 437, 2020, doi: 10.30865/mib.v4i2.2080.
- [3] S. Sahar, "Analisis Perbandingan Metode K-Nearest Neighbor dan Naïve Bayes Clasiffier Pada Dataset Penyakit Jantung," Indones. J. Data Sci., vol. 1, no. 3, pp. 79–86, 2020, doi: 10.33096/ijodas.v1i3.20.
- [4] Y. Yahya dan W. P. Hidayanti, "Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Efektivitas Penjualan Vape (Rokok Elektrik) pada 'Lombok Vape On,'" Infotek J. Inform. dan Teknol., vol. 3, no. 2, pp. 104–114, 2020, doi: 10.29408/jit.v3i2.2279.
- [5] J. Leskovec dan J. D. Ullman, Mining of Massive Datasets, Cambridge University Press, 2014.
- [6] M. M. Baharuddin, H. Azis, dan T. Hasanuddin, "Analisis Performa Metode K-Nearest Neighbor Untuk Identifikasi Jenis Kaca," Ilk. J. Ilm., vol. 11, no. 3, pp. 269–274, 2019, doi: 10.33096/ilkom.v11i3.489.269-274.

- [7] I. Pratiwi, “Analisis Performa Metode K-Nearest Neighbor (KNN) dan Cross-validation pada Data Penyakit Cardiovascular,” *J. Ilm. Komputasi dan Inform.*, vol. 2, no. 1, pp. 21–28, 2021.
 - [8] A. Zainuddin, “Evaluasi Akurasi KNN dalam Klasifikasi Sistem Monitoring Mesin Industri,” *J. Teknol. dan Informasi*, vol. 5, no. 1, pp. 45–52, 2023.
 - [9] L. Anwar dan T. R. Hidayat, “Implementasi KNN pada Prediksi Kinerja Mesin Produksi,” *J. Rekayasa Sistem Industri*, vol. 6, no. 3, pp. 213–220, 2022.
- H. Wibowo dan F. M. Rahayu, “Pengaruh Nilai K terhadap Akurasi KNN untuk Klasifikasi Multikategori,” *Semin. Nas. Teknol. Informasi dan Komputer*, vol. 9, no. 1, pp. 89–95, 2021.