Contents lists available at MID Publisher International

Technology Sciences Insights Journal

Journal homepage: https://journal.midpublisher.com/index.php/tsij

IMPLEMENTASI METODE EXTRA TREES CLASSIFIER UNTUK KLASIFIKASI JENIS SAMPAH MENGGUNAKAN EKSTRAKSI FITUR HISTOGRAM WARNA, HISTOGRAM OF ORIENTED GRADIENT, DAN LOCAL BINARY PATTERN

Esra Louis Veronika Gultom, Adlian Jefiza, Feralia Fitri
Politeknik Negeri Batam, Batam, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima: Juli 2026
Revisi: Juli 2026
Diterima: Juli 2026
Dipublikasi: Juli 2026

Kata Kunci:
Computer Vision; Extra Trees Classifier; HOG; LBP; Klasifikasi Sampah

Penulis Korespondensi:
esralouis25@gmail.com

ABSTRAK

Peningkatan jumlah sampah yang dihasilkan masyarakat menuntut adanya sistem pemilahan sampah yang lebih efektif dan otomatis. Salah satu teknologi yang dapat digunakan adalah *computer vision* yang mampu mengenali jenis sampah berdasarkan karakteristik visualnya. Penelitian ini bertujuan untuk mengklasifikasikan enam jenis sampah yaitu cardboard, glass, metal, paper, plastic, dan trash menggunakan metode Extra Trees Classifier. Dataset yang digunakan merupakan Garbage Classification Dataset yang diperoleh dari Kaggle dan telah dibagi menjadi data latih, validasi, dan pengujian. Tahapan penelitian meliputi *preprocessing* citra, ekstraksi fitur menggunakan Histogram Warna, Histogram of Oriented Gradient (HOG), dan Local Binary Pattern (LBP), kemudian dilanjutkan dengan proses klasifikasi menggunakan Extra Trees Classifier. Evaluasi model dilakukan menggunakan accuracy, precision, recall, dan F1-score, confusion matrix, dan ROC Curve. Hasil penelitian menunjukkan bahwa model mampu mencapai akurasi sebesar 99,88% dengan nilai precision, recall, dan F1-score mendekati 1,00 pada seluruh kelas. Hasil tersebut menunjukkan bahwa kombinasi fitur Histogram Warna, HOG, dan LBP mampu merepresentasikan karakteristik objek sampah dengan baik sehingga menghasilkan performa klasifikasi yang sangat tinggi.

ABSTRACT

*The increasing amount of waste generated by society requires a more effective and automated waste sorting system. One of the technologies that can be utilized is computer vision, which is capable of recognizing waste types based on their visual characteristics. This study aims to classify six categories of waste, namely cardboard, glass, metal, paper, plastic, and trash using the Extra Trees Classifier method. The dataset used is the Garbage Classification Dataset obtained from Kaggle and divided into training, validation, and testing sets. The research stages include image preprocessing, feature extraction using Color Histogram, Histogram of Oriented Gradients (HOG), and Local Binary Pattern (LBP), followed by classification using Extra Trees Classifier. Model performance was evaluated using accuracy, precision, recall, F1-score, confusion matrix, and ROC Curve. The experimental results show that the proposed model achieved an accuracy of **99.88%** with precision, recall, and F1-score values approaching **1.00** for all classes. These results indicate that the combination of Color Histogram, HOG, and LBP features can effectively represent waste object characteristics and provide excellent classification performance.*

PENDAHULUAN

Perkembangan dalam bidang *computer vision* dan *artificial intelligence* menunjukkan bahwa permasalahan pengelolaan dan pemilahan sampah semakin menjadi perhatian dalam beberapa tahun terakhir [1]. Kondisi ini dipengaruhi oleh meningkatnya jumlah sampah akibat

pertumbuhan populasi, urbanisasi, serta aktivitas konsumsi masyarakat yang semakin tinggi, sehingga diperlukan kajian lebih mendalam untuk memahami pengembangan sistem pemilahan sampah otomatis berbasis citra digital.

Penelitian oleh Dalal dan Triggs memperkenalkan metode Histogram of Oriented Gradient (HOG) yang mampu merepresentasikan karakteristik bentuk objek secara efektif melalui distribusi gradien citra [2]. Selanjutnya, Ojala dkk. mengembangkan metode Local Binary Pattern (LBP) yang banyak digunakan untuk ekstraksi fitur tekstur karena memiliki ketahanan terhadap perubahan pencahayaan [3]. Pada bidang klasifikasi citra, Geurts dkk. memperkenalkan algoritma Extra Trees Classifier yang merupakan pengembangan dari ensemble learning berbasis pohon keputusan dengan tingkat randomisasi yang lebih tinggi dibandingkan Random Forest [4]. Algoritma ini terbukti mampu meningkatkan kemampuan generalisasi model dan mengurangi risiko *overfitting* pada berbagai kasus klasifikasi. Berdasarkan penelitian-penelitian tersebut, kombinasi fitur warna, bentuk, dan tekstur dengan metode Extra Trees Classifier memiliki potensi untuk menghasilkan sistem klasifikasi sampah yang akurat dan efisien.

Meskipun berbagai penelitian sebelumnya telah membahas klasifikasi jenis sampah menggunakan metode *machine learning* dan *deep learning* seperti Convolutional Neural Network (CNN), Support Vector Machine (SVM), dan Random Forest [5], [6], masih terdapat keterbatasan berupa kompleksitas komputasi yang tinggi, kebutuhan sumber daya perangkat keras yang besar, serta belum optimalnya pemanfaatan kombinasi fitur warna, bentuk, dan tekstur pada beberapa penelitian [6], [7]. Keterbatasan tersebut memunculkan kebutuhan untuk meneliti lebih lanjut mengenai penggunaan metode Extra Trees Classifier dengan kombinasi fitur Histogram Warna, Histogram of Oriented Gradient (HOG), dan Local Binary Pattern (LBP) dalam klasifikasi jenis sampah, agar diperoleh pemahaman yang lebih komprehensif [2], [3], [4]. Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi jenis sampah yang mampu mengidentifikasi enam kategori sampah, yaitu cardboard, glass, metal, paper, plastic, dan trash dengan tingkat akurasi yang tinggi dengan menggunakan pendekatan *computer vision* berbasis ekstraksi fitur dan algoritma Extra Trees Classifier. Hasil penelitian diharapkan dapat memberikan kontribusi terhadap pengembangan ilmu pada bidang *computer vision*, *machine learning*, dan *smart waste management* serta memberikan manfaat praktis bagi industri pengolahan limbah, pengelola lingkungan, institusi pendidikan, dan peneliti di bidang kecerdasan buatan.

LANDASAN TEORI

Kajian teori ini membahas konsep-konsep utama dan hasil penelitian terdahulu yang menjadi dasar dalam menganalisis klasifikasi jenis sampah berbasis *computer vision* menggunakan metode Extra Trees Classifier. Secara teoretis, *Computer Vision* dan *Machine Learning* menjelaskan bahwa komputer dapat mengenali, mengekstraksi, dan mengklasifikasikan objek berdasarkan karakteristik visual yang terdapat pada citra digital [1], sehingga konsep ini relevan untuk memahami proses identifikasi jenis sampah berdasarkan informasi warna, bentuk, dan tekstur objek. Selain itu, teori ekstraksi fitur citra menggunakan Histogram Warna, Histogram of Oriented Gradient (HOG), dan Local Binary Pattern (LBP) turut memberikan landasan mengenai representasi fitur visual yang mampu menggambarkan karakteristik warna, bentuk, dan tekstur suatu objek, yang memperkuat hubungan antara fitur citra dan tingkat akurasi klasifikasi jenis sampah [2], [3].

Penelitian sebelumnya yang dilakukan oleh Yu dan Grammenos (2021) [5] menunjukkan bahwa penerapan *computer vision* pada sistem klasifikasi sampah mampu meningkatkan efisiensi

proses pemilahan limbah melalui pemanfaatan fitur visual dan algoritma kecerdasan buatan, namun masih terdapat keterbatasan pada optimalisasi kombinasi fitur yang digunakan untuk meningkatkan performa klasifikasi. Studi-studi lainnya yang dilakukan oleh Song dkk. (2022) [7] juga menegaskan bahwa penggabungan fitur warna dan bentuk mampu meningkatkan kemampuan model dalam membedakan berbagai kategori limbah dibandingkan penggunaan fitur tunggal, sedangkan penelitian Jin dkk. (2023) [6] menunjukkan bahwa kualitas representasi fitur memiliki pengaruh signifikan terhadap performa sistem klasifikasi sampah berbasis *machine vision*.

Meskipun demikian, sebagian besar penelitian terdahulu masih berfokus pada penggunaan metode *deep learning* yang memerlukan sumber daya komputasi relatif tinggi. Oleh karena itu, diperlukan analisis lebih lanjut untuk mengonfirmasi atau memperluas temuan tersebut dalam konteks klasifikasi enam kategori sampah menggunakan kombinasi Histogram Warna, HOG, dan LBP yang diklasifikasikan dengan metode Extra Trees Classifier. Berdasarkan keseluruhan teori dan penelitian terdahulu tersebut, penelitian ini mengadopsi kerangka berpikir bahwa kombinasi fitur warna, bentuk, dan tekstur mampu menghasilkan representasi visual yang lebih komprehensif sehingga algoritma Extra Trees Classifier dapat meningkatkan performa klasifikasi jenis sampah, yang selanjutnya akan digunakan sebagai dasar dalam penyusunan hipotesis dan analisis hasil penelitian.

METODE PENELITIAN

Metodologi penelitian ini dirancang untuk menjelaskan pendekatan, teknik pengumpulan data, serta prosedur analisis yang digunakan dalam mengkaji klasifikasi jenis sampah berbasis *computer vision* menggunakan metode Extra Trees Classifier. Penelitian ini menggunakan pendekatan kuantitatif karena dianggap paling sesuai untuk menjawab rumusan masalah dan tujuan penelitian yang berfokus pada pengukuran performa model klasifikasi berdasarkan nilai akurasi, precision, recall, dan F1-score. Populasi dalam penelitian ini adalah seluruh citra sampah yang terdapat pada Garbage Classification Dataset, sedangkan sampelnya ditentukan dengan menggunakan teknik *purposive sampling*, sehingga diperoleh dataset yang terdiri dari enam kategori sampah yaitu cardboard, glass, metal, paper, plastic, dan trash yang telah dibagi menjadi data *training*, *validation*, dan *testing* sebagai responden atau objek penelitian. Data dikumpulkan melalui dokumentasi dan observasi dataset citra digital yang diperoleh dari platform Kaggle, yang telah melalui proses *preprocessing* berupa perubahan ukuran citra menjadi 128 × 128 piksel untuk memastikan keseragaman data sebelum dilakukan proses ekstraksi fitur. Setelah itu, setiap citra diekstraksi menggunakan Histogram Warna, Histogram of Oriented Gradient (HOG), dan Local Binary Pattern (LBP) untuk memperoleh informasi warna, bentuk, dan tekstur objek sampah.

Setelah data terkumpul, analisis dilakukan menggunakan metode *machine learning* Extra Trees Classifier guna memperoleh temuan yang objektif dan sesuai dengan tujuan penelitian. Proses pelatihan model dilakukan menggunakan data *training*, sedangkan evaluasi model dilakukan menggunakan data *testing* dengan metrik accuracy, precision, recall, F1-score, confusion matrix, dan Receiver Operating Characteristic (ROC) Curve. Seluruh prosedur penelitian dilaksanakan secara sistematis untuk memastikan bahwa hasil yang diperoleh akurat, konsisten, dan dapat dipertanggungjawabkan secara ilmiah.

HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil penelitian yang diperoleh dari analisis terhadap data yang telah dikumpulkan, serta pembahasan mengenai temuan tersebut berdasarkan teori dan penelitian

terdahulu. Hasil analisis menunjukkan bahwa model klasifikasi yang dibangun menggunakan metode Extra Trees Classifier dengan kombinasi fitur Histogram Warna, Histogram of Oriented Gradient (HOG), dan Local Binary Pattern (LBP) mampu mencapai tingkat akurasi sebesar 99,88% pada data pengujian, yang menggambarkan kemampuan model dalam mengenali dan membedakan enam kategori sampah secara sangat akurat berdasarkan karakteristik warna, bentuk, dan tekstur objek.

Berdasarkan hasil pengujian, diperoleh nilai precision, recall, dan F1-score yang mendekati 1,00 pada seluruh kelas, yaitu cardboard, glass, metal, paper, plastic, dan trash. Hasil tersebut menunjukkan bahwa model tidak hanya memiliki kemampuan yang baik dalam melakukan prediksi yang benar, tetapi juga mampu meminimalkan kesalahan klasifikasi antar kategori sampah. Selain itu, confusion matrix menunjukkan bahwa dari total 2527 data pengujian, hanya terdapat 3 kesalahan klasifikasi, yaitu satu data pada kelas metal yang teridentifikasi sebagai glass serta dua data pada kelas plastic yang teridentifikasi sebagai glass. Dengan demikian, lebih dari 99% data berhasil diklasifikasikan secara tepat oleh model.

Temuan ini kemudian dibandingkan dengan teori terkait, di mana teori ekstraksi fitur citra menjelaskan bahwa kombinasi fitur warna, bentuk, dan tekstur mampu menghasilkan representasi visual yang lebih lengkap dibandingkan penggunaan satu jenis fitur saja. Pada penelitian ini, Histogram Warna berperan dalam merepresentasikan distribusi warna objek, HOG merepresentasikan bentuk dan arah gradien objek, sedangkan LBP merepresentasikan pola tekstur permukaan objek. Kombinasi ketiga fitur tersebut menghasilkan vektor fitur yang mampu membedakan karakteristik visual setiap kategori sampah secara efektif sehingga meningkatkan performa klasifikasi.

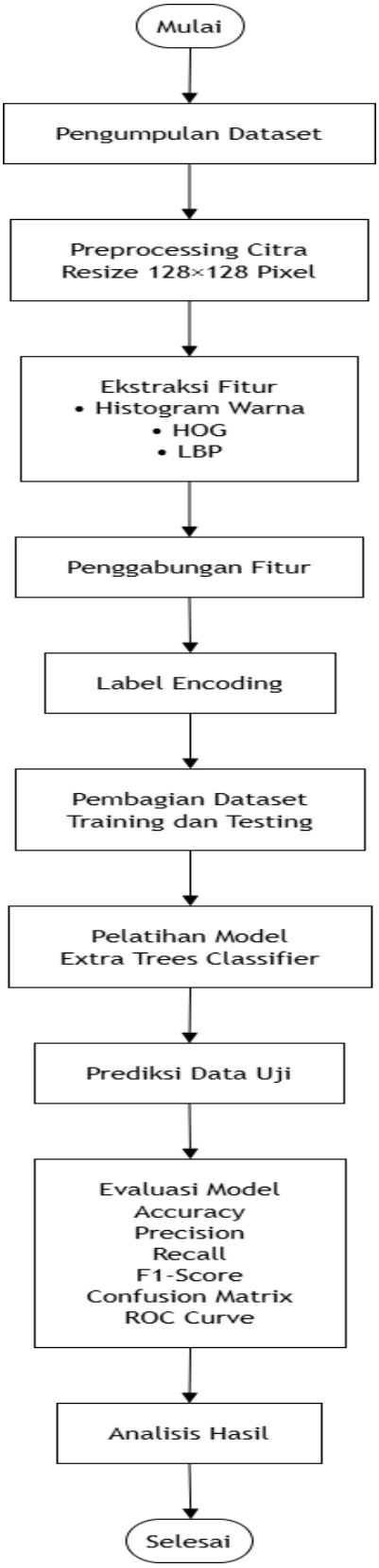
Selain itu, penelitian sebelumnya yang dilakukan oleh Song dkk. (2022) menunjukkan bahwa penggabungan beberapa fitur visual dapat meningkatkan performa sistem klasifikasi limbah dibandingkan penggunaan fitur tunggal. Penelitian Jin dkk. (2023) juga menunjukkan bahwa kualitas representasi fitur memiliki pengaruh signifikan terhadap kemampuan sistem *machine vision* dalam mengenali objek limbah. Hasil penelitian ini menunjukkan pola yang serupa karena penggunaan kombinasi Histogram Warna, HOG, dan LBP menghasilkan performa yang sangat tinggi pada proses klasifikasi sampah. Perbandingan ini menunjukkan adanya kesamaan antara hasil penelitian saat ini dan studi sebelumnya, yaitu bahwa kualitas ekstraksi fitur menjadi faktor utama yang menentukan keberhasilan sistem klasifikasi citra.

Hasil evaluasi menggunakan Receiver Operating Characteristic (ROC) Curve menunjukkan bahwa seluruh kategori sampah memiliki nilai Area Under Curve (AUC) sebesar 1,00. Nilai tersebut menunjukkan bahwa model memiliki kemampuan diskriminasi yang sangat baik dalam membedakan setiap kelas sampah. Semakin mendekati nilai 1,00, semakin baik kemampuan model dalam memisahkan kelas positif dan kelas negatif pada proses klasifikasi. Oleh karena itu, hasil ROC Curve memperkuat temuan bahwa model yang dikembangkan memiliki tingkat keandalan yang sangat tinggi.

Pembahasan lebih lanjut mengungkap bahwa faktor kombinasi fitur warna, bentuk, dan tekstur serta penggunaan algoritma Extra Trees Classifier memiliki pengaruh signifikan terhadap performa klasifikasi. Extra Trees Classifier bekerja dengan membangun banyak pohon keputusan yang dibentuk secara acak sehingga mampu meningkatkan kemampuan generalisasi model dan mengurangi risiko *overfitting*. Selain itu, metode ini juga mampu menangani data berdimensi tinggi yang dihasilkan dari proses ekstraksi fitur HOG dan LBP tanpa mengalami penurunan performa yang signifikan.

Secara keseluruhan, hasil penelitian ini memberikan kontribusi penting dalam memperkuat kajian mengenai penerapan *machine learning* pada bidang *computer vision* untuk klasifikasi sampah. Hasil akurasi sebesar 99,88% menunjukkan bahwa kombinasi Histogram Warna, HOG, dan LBP yang diklasifikasikan menggunakan Extra Trees Classifier merupakan pendekatan yang

efektif untuk mengidentifikasi berbagai jenis sampah berdasarkan citra digital. Temuan ini juga membuka peluang untuk penelitian lanjutan yang dapat mengkaji implementasi sistem secara *real-time* menggunakan kamera dan perangkat *embedded*, sehingga dapat diterapkan pada sistem pemilahan sampah otomatis di lingkungan industri maupun *smart city*.



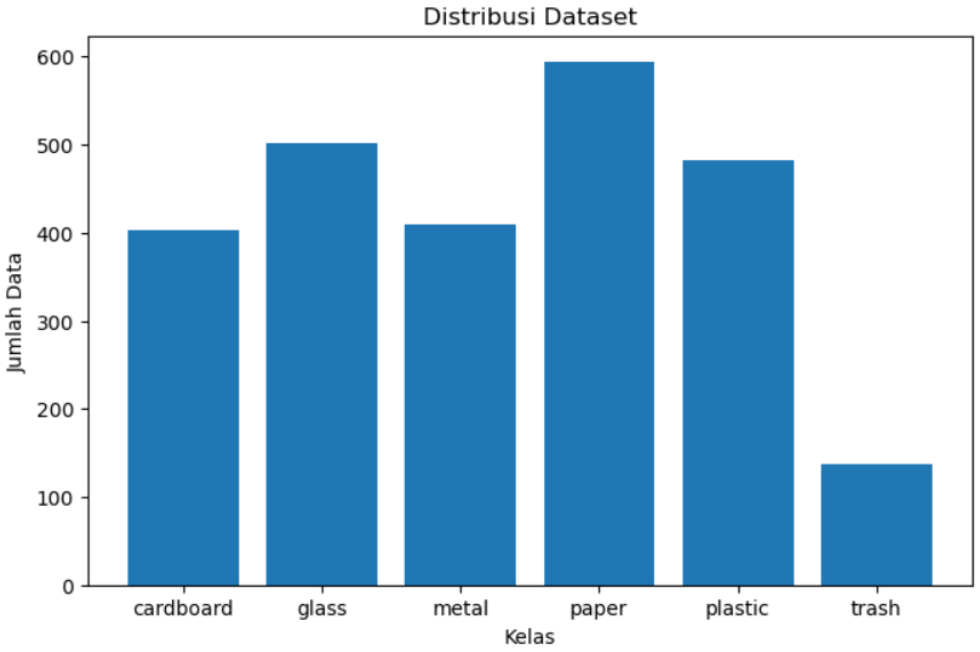
Gambar 1. Flowchart Tahapan Penelitian

Gambar 1 menunjukkan tahapan penelitian yang dimulai dari proses pengumpulan dataset Garbage Classification, *preprocessing* citra, ekstraksi fitur menggunakan Histogram Warna, Histogram of Oriented Gradient (HOG), dan Local Binary Pattern (LBP). Fitur yang diperoleh kemudian digabungkan dan digunakan sebagai input pada model Extra Trees Classifier. Selanjutnya dilakukan proses pengujian dan evaluasi menggunakan metrik accuracy, precision, recall, F1-score, confusion matrix, dan ROC Curve untuk mengetahui performa model dalam mengklasifikasikan jenis sampah.

Tabel 1. Hasil Evaluasi Model

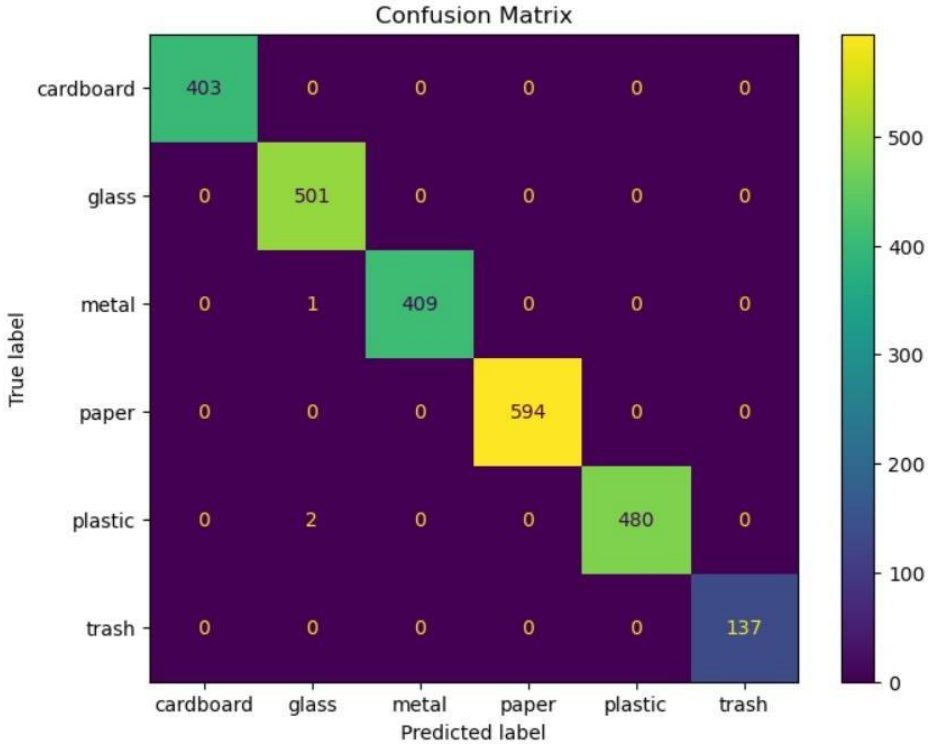
Parameter	Nilai
Accuracy	99,88%
Precision	1,00
Recall	1,00
F1-Score	1,00
AUC	1,00

Berdasarkan hasil pengujian, model berhasil mengklasifikasikan hampir seluruh data pengujian dengan benar. Tingginya akurasi menunjukkan bahwa kombinasi fitur warna, bentuk, dan tekstur mampu memberikan representasi yang baik terhadap karakteristik setiap jenis sampah.



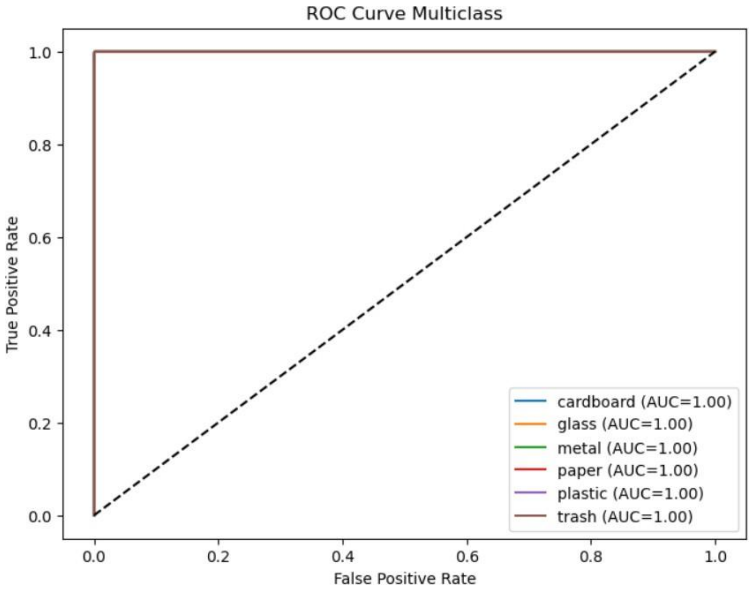
Gambar 2. Distribusi Dataset

Gambar 2 menunjukkan distribusi jumlah data pada masing-masing kategori sampah. Distribusi data yang relatif seimbang membantu model dalam mempelajari karakteristik setiap kelas secara lebih optimal.



Gambar 3. Confusion Matrix Extra Trees Classifier

Berdasarkan confusion matrix pada Gambar 3, sebagian besar data berhasil diklasifikasikan dengan benar. Kesalahan klasifikasi hanya terjadi sebanyak tiga data dari total 2527 data pengujian.



Gambar 4. ROC Curve

Gambar 4 menunjukkan bahwa seluruh kelas memiliki nilai AUC sebesar 1,00 yang mengindikasikan kemampuan model yang sangat baik dalam membedakan setiap kategori sampah.

KESIMPULAN

Kesimpulan dari penelitian ini menunjukkan bahwa penerapan metode Extra Trees Classifier dengan kombinasi fitur Histogram Warna, Histogram of Oriented Gradient (HOG), dan Local Binary Pattern (LBP) mampu menghasilkan performa yang sangat baik dalam klasifikasi jenis

sampah berbasis citra digital. Hal ini memberikan gambaran yang jelas mengenai kemampuan algoritma *machine learning* dalam mengenali karakteristik visual objek berdasarkan informasi warna, bentuk, dan tekstur.

Berdasarkan hasil analisis, dapat disimpulkan bahwa model yang dikembangkan berhasil mengklasifikasikan enam kategori sampah, yaitu cardboard, glass, metal, paper, plastic, dan trash dengan tingkat akurasi sebesar 99,88%, precision sebesar 1,00, recall sebesar 1,00, dan F1-score sebesar 1,00, yang menguatkan bahwa kombinasi ekstraksi fitur Histogram Warna, HOG, dan LBP mampu menghasilkan representasi visual yang efektif untuk membedakan setiap kategori sampah secara akurat.

Penelitian ini juga mengungkap bahwa penggunaan metode Extra Trees Classifier mampu menangani data hasil ekstraksi fitur berdimensi tinggi dengan sangat baik, yang ditunjukkan oleh nilai Area Under Curve (AUC) sebesar 1,00 pada seluruh kelas serta hanya terdapat tiga kesalahan klasifikasi dari total 2527 data pengujian, sehingga memberikan kontribusi terhadap pengembangan pengetahuan dalam bidang *computer vision*, *machine learning*, dan sistem klasifikasi citra untuk pengelolaan sampah otomatis.

Selain itu, hasil penelitian ini diharapkan dapat menjadi dasar bagi penelitian selanjutnya, khususnya dalam mengkaji implementasi sistem klasifikasi sampah secara *real-time* menggunakan kamera, integrasi dengan perangkat *embedded* seperti Raspberry Pi atau ESP32, serta pengembangan sistem *smart waste management* berbasis *Internet of Things (IoT)*. Dengan demikian, studi ini tidak hanya memberikan pemahaman baru mengenai penerapan Extra Trees Classifier pada klasifikasi sampah, tetapi juga membuka peluang untuk pengembangan kajian lebih lanjut yang dapat memperkaya literatur terkait otomasi pemilahan sampah dan penerapan kecerdasan buatan di bidang lingkungan.

Daftar Pustaka

[1] Richard Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. Springer, 2022.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893.

[3] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[4] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees", *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006, doi: 10.1007/s10994-006-6226-1.

[5] Y. Yu and R. Grammenos, "Towards artificially intelligent recycling: Improving image processing for waste classification," arXiv preprint arXiv:2108.06274, 2021.

[6] S. Jin, Z. Yang, G. Królczyk, and Z. Li, "Garbage detection and classification using a new deep learning-based machine vision system as a tool for sustainable waste recycling," *Waste Management*, vol. 162, pp. 123–130, 2023.

[7] L. Song, H. Zhao, Z. Ma, and Q. Song, "A new method of construction waste classification based on two-level fusion," *PLOS ONE*, vol. 17, no. 12, p. e0279472, 2022, doi: 10.1371/journal.pone.0279472.

Contents lists available at MID Publisher International

Technology Sciences Insights Journal

Journal homepage: https://journal.midpublisher.com/index.php/tsij

Implementasi Feature Engineering dan Random Forest dalam Klasifikasi Kegagalan Mesin Menggunakan Dataset AI4I 2020

Bryan Mandagi

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima : Agustus 2026
Revisi : Agustus 2026
Diterima : Agustus 2026
Dipublikasi : Agustus 2026

Kata Kunci:
Random Forest; Predictive Maintenance; Klasifikasi; Kegagalan Mesin; AI4I 2020

*Penulis Korespondensi:
bryanmandagi24@gmail.com

ABSTRAK

Penelitian ini bertujuan untuk mengklasifikasikan kondisi kegagalan mesin pada sistem *predictive maintenance* menggunakan metode Random Forest dengan memanfaatkan AI4I 2020 Predictive Maintenance Dataset. Data diperoleh dari *UCI Machine Learning Repository* dan melalui tahapan *preprocessing* berupa penghapusan atribut yang tidak relevan, penanganan *data leakage*, serta *one-hot encoding* pada data kategorikal. Selanjutnya dilakukan *feature engineering* dengan membentuk fitur baru, yaitu Power_W, Temp_diff_K, dan Wear_Torque, untuk meningkatkan representasi karakteristik mesin. Model Random Forest kemudian dilatih menggunakan data yang telah distandardisasi dan dievaluasi menggunakan Confusion Matrix, ROC Curve, Histogram Distribusi Fitur, Scatter Plot, serta metrik Accuracy, Precision, Recall, dan F1-Score. Hasil penelitian menunjukkan bahwa model memperoleh akurasi sebesar 99,08%, precision sebesar 93,06%, recall sebesar 78,82%, F1-score sebesar 85,35%, serta AUC sebesar 0,982. Hasil tersebut menunjukkan bahwa metode Random Forest mampu mengklasifikasikan kondisi mesin dengan tingkat akurasi yang tinggi sehingga berpotensi diterapkan sebagai pendukung sistem *predictive maintenance* pada lingkungan otomasi industri.

ABSTRACT

This study aims to classify machine failure conditions in a predictive maintenance system using the Random Forest algorithm based on the AI4I 2020 Predictive Maintenance Dataset. The dataset was obtained from the UCI Machine Learning Repository and processed through several preprocessing stages, including the removal of irrelevant attributes, elimination of data leakage, and one-hot encoding for categorical variables. Furthermore, feature engineering was performed by generating three additional features, namely Power_W, Temp_diff_K, and Wear_Torque, to improve machine condition representation. The Random Forest model was trained using standardized data and evaluated through Confusion Matrix, ROC Curve, Feature Distribution Histogram, Scatter Plot, and performance metrics including Accuracy, Precision, Recall, and F1-Score. The experimental results achieved an accuracy of 99.08%, precision of 93.06%, recall of 78.82%, F1-score of 85.35%, and an AUC value of 0.982. These findings demonstrate that the Random Forest algorithm provides excellent classification performance and has strong potential for supporting predictive maintenance applications in industrial automation systems.

PENDAHULUAN

Perkembangan teknologi pada era Industri 4.0 mendorong penerapan kecerdasan buatan (*Artificial Intelligence*) dalam sistem otomasi industri, salah satunya melalui predictive maintenance. Predictive maintenance memanfaatkan data operasional mesin untuk

memprediksi potensi kegagalan sehingga proses perawatan dapat dilakukan sebelum terjadi kerusakan. Pendekatan ini mampu mengurangi *downtime*, meningkatkan efisiensi produksi, dan menekan biaya pemeliharaan.

Salah satu algoritma klasifikasi yang banyak digunakan dalam predictive maintenance adalah Random Forest karena memiliki akurasi yang tinggi dan mampu menangani data berdimensi besar. Penelitian ini menggunakan AI4I 2020 Predictive Maintenance Dataset untuk membangun model klasifikasi kondisi mesin. Sebelum proses klasifikasi dilakukan, data melalui tahapan *preprocessing* dan *feature engineering* untuk meningkatkan kualitas data. Selanjutnya model dievaluasi menggunakan Confusion Matrix, ROC Curve, Histogram, Scatter Plot, serta metrik Accuracy, Precision, Recall, dan F1-Score. Penelitian ini bertujuan menganalisis performa algoritma Random Forest dalam mengklasifikasikan kondisi kegagalan mesin pada sistem predictive maintenance.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan memanfaatkan AI4I 2020 Predictive Maintenance Dataset sebagai sumber data. Tahapan penelitian dimulai dengan *preprocessing* data melalui penghapusan atribut yang tidak relevan, penanganan *data leakage*, dan *one-hot encoding* pada data kategorikal. Selanjutnya dilakukan *feature engineering* dengan membentuk fitur Power_W, Temp_diff_K, dan Wear_Torque untuk meningkatkan representasi karakteristik mesin.

Data kemudian dibagi menjadi data latih (75%) dan data uji (25%), dilanjutkan dengan proses standardisasi menggunakan StandardScaler. Model klasifikasi dibangun menggunakan algoritma Random Forest, kemudian dievaluasi menggunakan Confusion Matrix, ROC Curve, Histogram Distribusi Fitur, Scatter Plot, Feature Importance, serta metrik Accuracy, Precision, Recall, dan F1-Score untuk mengukur kinerja model.

HASIL DAN PEMBAHASAN

Penelitian ini menggunakan algoritma Random Forest untuk mengklasifikasikan kondisi mesin berdasarkan AI4I 2020 Predictive Maintenance Dataset. Sebelum proses klasifikasi dilakukan, data melalui tahap *preprocessing* berupa penghapusan atribut yang tidak relevan, penanganan *data leakage*, *one-hot encoding*, serta pembentukan fitur baru melalui *feature engineering*. Setelah model dilatih menggunakan data latih, dilakukan pengujian terhadap data uji untuk mengetahui performa model.

1. Hasil Evaluasi Model

Berdasarkan hasil pengujian, model Random Forest menghasilkan performa klasifikasi yang sangat baik. Nilai evaluasi yang diperoleh ditunjukkan pada Tabel 1.

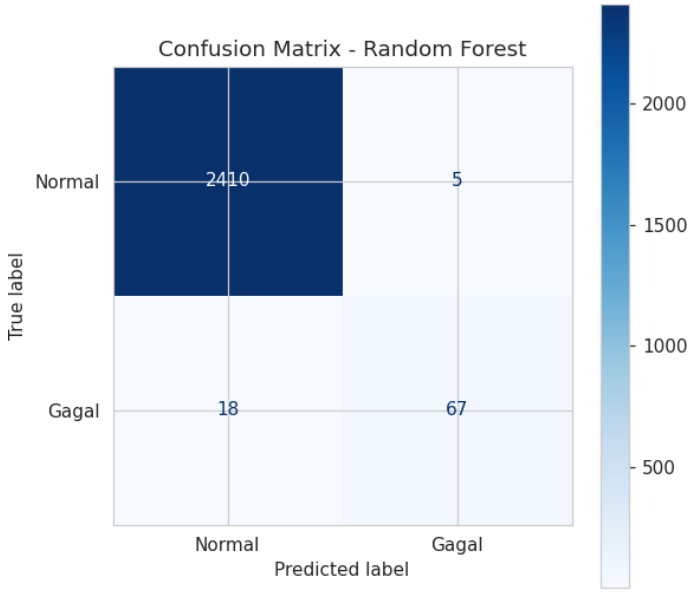
Table 1. Hasil Evaluasi Model Random Forest

Parameter	Nilai
Accuracy	99,08%
Precision	93,06%
Recall	78,82%
F1-Score	85,35%
AUC	0,982

Nilai akurasi sebesar 99,08% menunjukkan bahwa model mampu mengklasifikasikan kondisi mesin dengan tingkat ketepatan yang sangat tinggi. Selain itu, nilai precision sebesar 93,06%

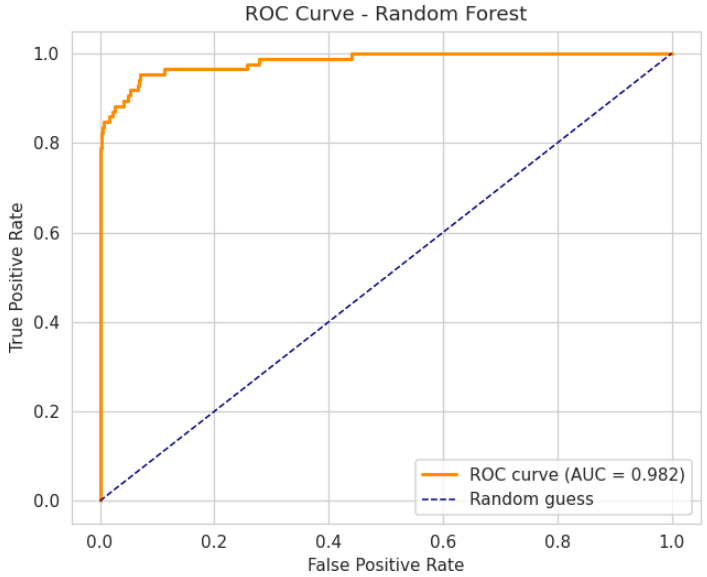
menunjukkan bahwa sebagian besar data yang diprediksi sebagai kegagalan mesin benar-benar merupakan kondisi gagal. Nilai recall sebesar 78,82% mengindikasikan bahwa model masih berhasil mendeteksi sebagian besar kasus kegagalan mesin, sedangkan nilai F1-score sebesar 85,35% menunjukkan keseimbangan yang baik antara precision dan recall.

2. Analisis Confusion Matrix



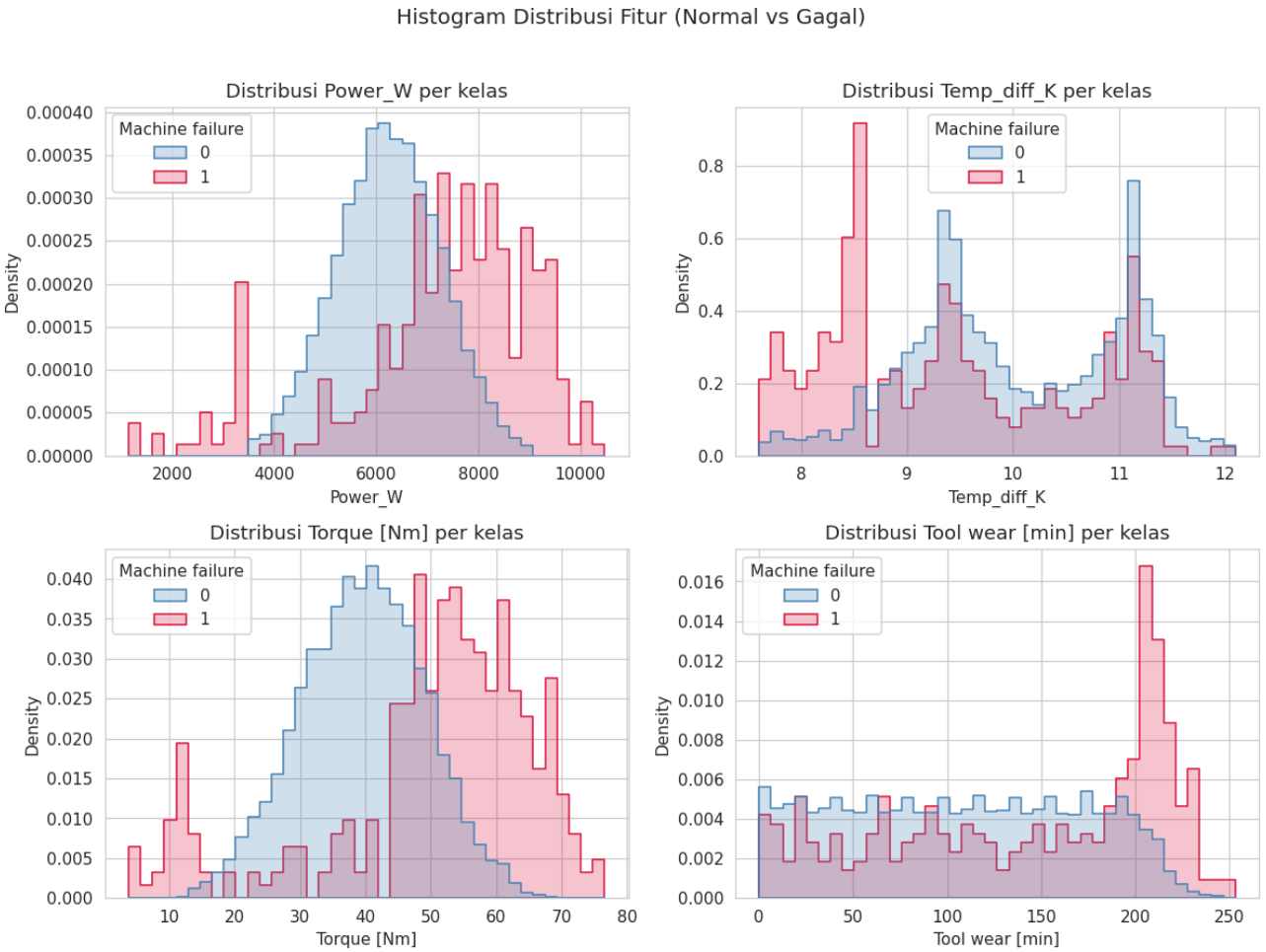
Berdasarkan Confusion Matrix, model berhasil mengklasifikasikan 2410 data normal dengan benar (*True Negative*) dan 67 data gagal dengan benar (*True Positive*). Sementara itu terdapat 5 data normal yang diprediksi sebagai gagal (*False Positive*) serta 18 data gagal yang diprediksi sebagai normal (*False Negative*). Hasil tersebut menunjukkan bahwa model memiliki tingkat kesalahan klasifikasi yang rendah sehingga mampu membedakan kondisi mesin normal dan gagal dengan baik

3. Analisis ROC Curve



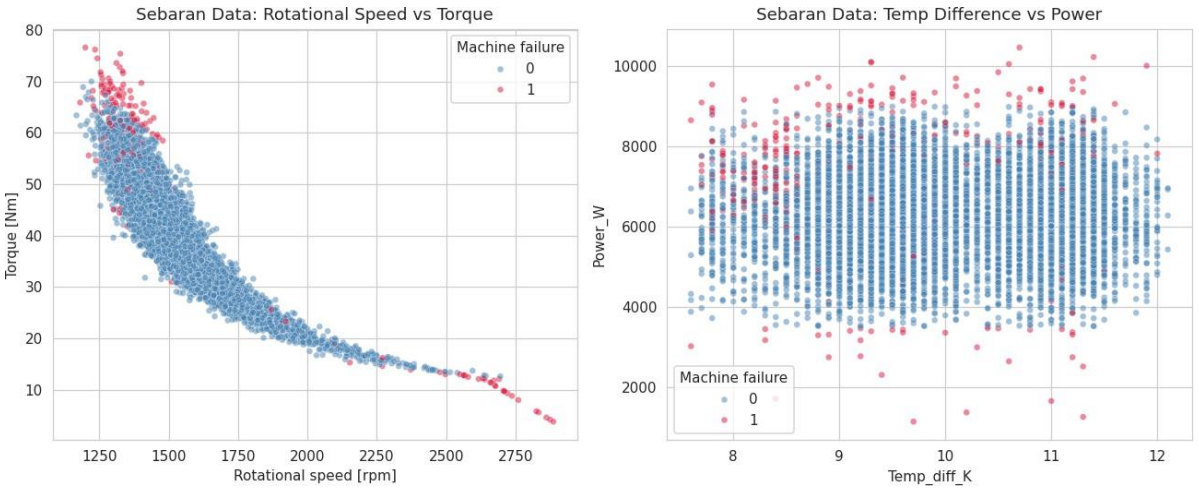
Kurva ROC menunjukkan hubungan antara True Positive Rate dan False Positive Rate pada berbagai nilai ambang klasifikasi. Berdasarkan hasil pengujian diperoleh nilai Area Under Curve (AUC) sebesar 0,982. Nilai tersebut termasuk dalam kategori excellent classification, sehingga menunjukkan bahwa Random Forest memiliki kemampuan yang sangat baik dalam membedakan mesin yang mengalami kegagalan dan mesin yang berada pada kondisi normal.

4. Analisis Histogram Distribusi Fitur



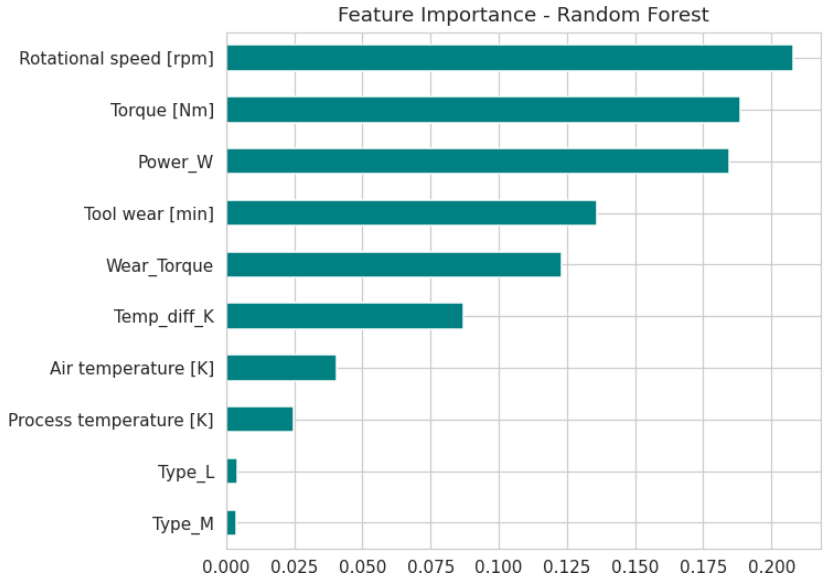
Histogram distribusi fitur menunjukkan perbedaan karakteristik antara data mesin normal dan data mesin gagal. Pada fitur Power_W, Torque, dan Tool Wear, terlihat bahwa data kegagalan mesin cenderung berada pada nilai yang lebih tinggi dibandingkan data normal. Sementara itu, pada fitur Temp_diff_K masih terdapat sedikit tumpang tindih antara kedua kelas, namun distribusi keseluruhan tetap menunjukkan pola yang berbeda. Hal ini mengindikasikan bahwa fitur hasil *feature engineering* mampu membantu model dalam membedakan kondisi mesin.

5. Analisis Sebaran Data (Scatter Plot)



Scatter plot pertama memperlihatkan hubungan antara Rotational Speed dan Torque, dimana data kegagalan mesin cenderung berada pada daerah dengan kombinasi torsi tinggi dan kecepatan putar yang lebih rendah. Scatter plot kedua menunjukkan hubungan antara Temperature Difference dan Power, yang memperlihatkan bahwa sebagian besar data kegagalan memiliki nilai daya yang relatif tinggi. Sebaran tersebut menunjukkan adanya pola yang dapat dipelajari oleh algoritma Random Forest untuk melakukan klasifikasi.

6. Analisis Feature Importance



Hasil analisis *feature importance* menunjukkan bahwa Rotational Speed [rpm] merupakan fitur yang memiliki pengaruh terbesar dalam proses klasifikasi, diikuti oleh Torque [Nm], Power_W, Tool Wear, dan Wear_Torque. Sementara itu, fitur Type_L dan Type_M memiliki kontribusi yang relatif kecil terhadap keputusan model. Hasil ini menunjukkan bahwa parameter operasional mesin memiliki pengaruh yang lebih besar dibandingkan jenis produk dalam menentukan kondisi kegagalan mesin.

Secara keseluruhan, hasil penelitian menunjukkan bahwa algoritma Random Forest mampu memberikan performa klasifikasi yang sangat baik pada AI4I 2020 Predictive Maintenance Dataset. Kombinasi proses *preprocessing* dan *feature engineering* berhasil meningkatkan kualitas data sehingga model mampu menghasilkan tingkat akurasi yang tinggi dan kesalahan klasifikasi yang rendah.

KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa algoritma Random Forest mampu mengklasifikasikan kondisi kegagalan mesin pada AI4I 2020 Predictive Maintenance Dataset dengan performa yang sangat baik. Model yang dibangun melalui tahapan *preprocessing*, *feature engineering*, dan standardisasi data menghasilkan nilai akurasi sebesar 99,08%, precision sebesar 93,06%, recall sebesar 78,82%, F1-score sebesar 85,35%, serta AUC sebesar 0,982.

Hasil analisis menggunakan Confusion Matrix, ROC Curve, Histogram, Scatter Plot, dan Feature Importance menunjukkan bahwa model mampu membedakan kondisi mesin normal dan gagal secara efektif. Fitur Rotational Speed, Torque, dan Power_W merupakan faktor yang paling berpengaruh dalam proses klasifikasi. Dengan demikian, algoritma Random Forest berpotensi diterapkan sebagai metode pendukung sistem predictive maintenance pada lingkungan otomasi industri untuk membantu proses pemantauan kondisi mesin dan mengurangi risiko terjadinya kegagalan operasional.

Daftar Pustaka

[1] A. Matzka, "Explainable Artificial Intelligence for Predictive Maintenance Applications: A Survey," *IEEE Access*, vol. 9, pp. 118784–118806, 2021.

[2] D. Chicco and G. Jurman, "The Advantages of the Matthews Correlation Coefficient over F1 Score and Accuracy in Binary Classification Evaluation," *BMC Genomics*, vol. 21, no. 6, pp. 1–13, 2020.

[3] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] UCI Machine Learning Repository, "AI4I 2020 Predictive Maintenance Dataset," University of California, Irvine, 2020.

[6] S. K. S. Fan, Y. C. Hsu, and C. C. Tsai, "Machine Learning Techniques for Predictive Maintenance in Smart Manufacturing," *IEEE Access*, vol. 10, pp. 55733–55747, 2022.

[7] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY, USA: Springer, 2009.

[10] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2012.

[11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[13] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, 2nd ed. New York, NY, USA: Springer, 2021.

[14] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[15] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A Survey on Concept Drift Adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014.

Contents lists available at MID Publisher International

Technology Sciences Insights Journal

Journal homepage: https://journal.midpublisher.com/index.php/tsij

Air Quality Monitoring System Metode KNN (K-Nearest Neighbor) untuk Mendeteksi Kondisi Kualitas Air

Ardiyan Syah Ariansyah
Politeknik Negeri Batam, Batam, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima: Juni 2025
Revisi: Juni 2025
Diterima: Juli 2025
Dipublikasi: Juli 2025

Kata Kunci:
Water Quality; K-Nearest Neighbor; Machine Learning; Classification; Monitoring System

*Penulis Korespondensi:
syahardi132@gmail.com

ABSTRAK

Penelitian ini bertujuan untuk membuat sebuah system yang berguna untuk mempermudah manusia untuk mengetahui kondisi kualitas air secara manual menggunakan metode KNN (K-Nearest Neighbor). Kualitas air adalah aspek penting dalam kehidupan sehari-hari, dan upaya untuk memastikan air yang aman untuk dikonsumsi merupakan prioritas. Penelitian ini mengajukan pertanyaan utama: "Apakah metode KNN dapat memberikan pendekatan yang efektif dalam mengklasifikasikan kualitas air?" Metodologi penelitian ini melibatkan penggunaan dataset sekunder mengenai kualitas air dan menerapkan algoritma KNN untuk mengkategorikan data tersebut. Hasil dari metode KNN dibandingkan dengan metode klasifikasi lain yang umum digunakan dalam analisis kualitas air. Penelitian ini bertujuan untuk memberikan pandangan yang lebih dalam dan pemahaman yang lebih baik tentang potensi metode KNN. Dengan memakai metode KNN (KNearest Neighbor) diperoleh nilai ketelitian dari K, nilai K yang menggunakan K=5 akurasinya sebesar 97%, setelah KNN mendapatkan hasil dari deteksi, hasil tersebut akan berguna untuk mengatur tindakan dari aktuator. Hasil penelitian ini diharapkan dapat memberikan wawasan baru tentang efektivitas metode KNN dalam mengklasifikasikan kualitas air yang dapat membantu masyarakat dan pemerintah dalam mengambil tindakan respons yang lebih efisien terkait dengan potensi kontaminasi dan masalah kualitas air lainnya. Dengan demikian, penelitian ini memiliki potensi untuk memberikan kontribusi berharga pada pemahaman kita tentang kualitas air dan pemanfaatan metode KNN dalam analisisnya.

ABSTRACT

This research aims to create a system that is useful to make it easier for humans to know the condition of water quality manually using the KNN (K-Nearest Neighbor) method. Water quality is an important aspect of daily life, and efforts to ensure safe water for consumption are a priority. This research asks the main question: "Can the KNN method provide an effective approach in classifying water quality?" The research methodology involved using a secondary dataset on water quality and applying the KNN algorithm to categorize the data. The results of the KNN method were compared with other classification methods commonly used in water quality analysis. This research aims to provide a deeper look and better understanding of the potential of the KNN method. By using the KNN (K-Nearest Neighbor) method, the accuracy value of K is obtained, the value of K using K = 5 is 97% accuracy, after KNN gets the results of detection, these results will be useful for regulating the actions of the actuator. The results of this study are expected to provide new insights into the effectiveness of the KNN method in classifying water quality that can assist the public and government in taking more efficient response actions related to potential contamination and other

water quality issues. As such, this research has the potential to make valuable contributions to our understanding of water quality and the utilization of the KNN method in its analysis.

PENDAHULUAN

Air adalah unsur yang mendominasi wilayah terluas di planet bumi, mencakup hampir 71% dari seluruh permukaan dan ada di berbagai bentuk, termasuk air laut, air sungai, air permukaan, air atmosfer, air rawa, dan air tanah. Selain memenuhi kebutuhan dasar manusia seperti konsumsi dan memasak, air juga memiliki peran penting dalam industri, pertanian, dan pembangunan infrastruktur. Kualitas air memiliki dampak yang sangat signifikan pada kehidupan kita. Faktor-faktor lingkungan fisik, seperti pola curah hujan, topografi, dan jenis batuan, berperan dalam menentukan kuantitas air di suatu wilayah. Sementara itu, kualitas air sangat dipengaruhi oleh aspek-aspek sosial seperti kepadatan penduduk dan tingkat interaksi sosial. Di tengah kompleksitas ini, perhatian terhadap kualitas air menjadi semakin penting. Pembangunan yang pesat seringkali mengubah aliran sungai menjadi area permukiman dan industri di kota-kota, yang pada gilirannya dapat mengancam kualitas air yang mengalir di sepanjang sungai tersebut.

Penelitian tentang klasifikasi air minum telah banyak dilakukan, terutama mengenai klasifikasi kualitas air minum menggunakan machine learning. Penelitian oleh Aldi dkk. (2023) menggunakan metode Naive Bayes, Decision Tree, dan K-Nearest Neighbors untuk menemukan metode yang paling akurat. Hasil penelitian menunjukkan bahwa metode Decision Tree memiliki akurasi tertinggi. Penelitian oleh Prismahardi dkk. (2023) menggunakan metode Support Vector Machine, Decision Tree, Naïve Bayes, dan Artificial Neural Network. Hasil penelitian menunjukkan bahwa metode Random Forest Classifier memiliki akurasi tertinggi. Kemudian penelitian yang dilakukan oleh Ilyas dkk yang bertujuan untuk mengklasifikasikan kualitas air minum menggunakan algoritma Random Forest. Data yang digunakan adalah data kualitas air minum dari 267 sampel air sumur di Jakarta. Hasil penelitian menunjukkan bahwa algoritma Random Forest dapat mengklasifikasikan kualitas air minum dengan akurasi sebesar 82,3%.

Pada penelitian kali ini bertujuan untuk memberikan hal inovasi dengan menggunakan metode K-Nearest Neighbor (KNN) dalam menentukan kualitas air. Efektivitas metode KNN akan dievaluasi dalam mengklasifikasikan data kualitas air yang diharapkan dapat memberikan wawasan yang lebih baik dan membantu dalam mengatasi tantangan kualitas air yang dihadapi oleh masyarakat Indonesia dan negara-negara lainnya. KNN adalah algoritma pembelajaran mesin terawasi yang digunakan untuk klasifikasi dan regresi, memanipulasi data training serta mengklasifikasikan data testing berdasarkan metrik jarak. Dengan mencari K tetangga terdekat dari data uji, algoritme ini melakukan klasifikasi berdasarkan mayoritas label kelas, dan termasuk dalam kelas instance base learning serta lazy learning. Dengan penerapan teknik K-Nearest Neighbor (KNN), merupakan salah satu pendekatan klasifikasi terhadap kumpulan data yang berfokus pada mayoritas kategori. Tujuannya adalah untuk mengategorikan objek baru dengan merujuk pada atribut dan contohcontoh sampel dari data pelatihan. Dengan demikian, upaya ini bertujuan untuk mendekati akurasi yang lebih tinggi saat melakukan evaluasi pembelajaran.

LANDASAN TEORI (Optional)

1. KNN (K-Nearest Neighbor)

K-Nearst Neighbor (KNN) adalah suatu algoritma yang digunakan untuk mengklasifikasikan data dengan memanfaatkan data latih (training record) dari tetangga terdekat, di mana k merupakan jumlah tetangga terdekat yang digunakan dalam proses klasifikasi. KNN adalah jenis algoritma yang sederhana dibandingkan algoritma lain dalam

machine learning. Prinsip kerjanya adalah dengan menghitung jarak perbedaan antara data uji dan data latih yang ada dalam sistem, kemudian mencari nilai terkecil atau paling mirip untuk mengklasifikasikan data tersebut. Suatu Objek di klasfikasikan dengan class mayoritas dari class tetangga, dimana diambil class yang paling banyak muncul dalam batasan K tertentu dari tetangganya,sehingga diperoleh class baru sesuai dengan tetangga yang paling umum, jika K=1, maka objek ditetapkan sesuai dengan class tetangga terdekatnya. Pada fase klasifikasi, K adalah sebuah konstanta yang ditetapkan pengguna. Class baru dari suatu data test diklasifikasikan dengan menetapkan class yang paling sering muncul diantara sampel ke titik data training yang ada. Metode K-Nearst Neighbor pada prinsip kerjanya mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (Neighbor) terdekatnya dalam data sampel.

Euclidean distance merupakan jarak antara dua titik atau koordinat yang dihitung menggunakan rumus Pythagoras. Ini adalah panjang garis lurus yang menghubungkan dua titik dalam ruang, yang disebut titik a dan titik b. Garis ini, juga dikenal sebagai garis miring, terbentang di antara sumbu x dan sumbu y dengan koordinat yang diberikan untuk titik a dan titik b. Gambar berikut menunjukan rumus perhitungan untuk mencari jarak antara dua titik yaitu titik pada data training (x) dan titik pada data testing (y) maka digunakan rumus Euclidean, seperti pada persamaan berikut.

$$\mathrm{D(x,y)}\ \sqrt{\sum_{k=1}^{n}(p_1-q_1)^2}$$

Gambar 1. Rumus Euclidean

Dengan D adalah jarak antara titik pada data training x dan titik data testing yang akan diklasifikasi, dimana x=x1, x2, …, xi dan y1, y2, … yi dan i mempresentasikan nilai atribut serta n merupakan dimensi atribut.

a) Menentukan parameter K (jumlah tetangga paling dekat).
b) Menghitung kuadrat jarak Euclid (query instance) masing-masing objek terhadap data sampel yang diberikan.
c) Kemudian mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak Euclid terkecil.
d) Mengumpulkan kategori Y (klasifikasi nearest neighbor).
e) Dengan menggunakan kategori Nearest Neighbor yang paling mayoritas maka dapat diprediksi nilai query instance yang telah dihitung.

2. Datashet

Dataset adalah kumpulan data yang terdiri dari beberapa entitas atau sampel yang diambil dari berbagai sumber. Dalam konteks penelitian ilmiah, dataset biasanya berisi sekumpulan variabel atau atribut yang diukur atau diamati pada setiap entitas, serta informasi tambahan yang relevan untuk analisis. Dataset merupakan komponen penting dalam penelitian karena menjadi dasar untuk melakukan analisis, pembuatan model, dan menyimpulkan temuan penelitian. Dalam penelitian mengenai metode KNN untuk menentukan kualitas air, dataset berisi data pengukuran atau observasi berbagai parameter kualitas air, seperti konsentrasi logam berat, kandungan bakteri, dan kualitas kimia lainnya. Setiap baris dalam dataset mewakili satu entitas atau contoh, seperti sampel air yang diambil dari berbagai lokasi atau sumber. Sedangkan, setiap kolom merepresentasikan atribut atau parameter yang diamati pada setiap entitas.

Contoh dataset yang digunakan dalam penelitian ini diambil dari website Kaggle.com yang berjudul waterQuality1. Data ini telah dikumpulkan dari berbagai lokasi dan telah melalui proses validasi dan pengolahan sebelum digunakan dalam klasifikasi. Sedangkan klasifikasi sendiri merupakan langkah dimana data dikelompokkan ke dalam kelas-kelas yang telah terdefinisi sebelumnya, berdasarkan atribut atau fitur-fitur tertentu yang dimiliki oleh data

tersebut. Proses ini bertujuan untuk mengidentifikasi pola atau karakteristik yang membedakan setiap kelas dan memungkinkan untuk pengambilan keputusan yang lebih baik berdasarkan pada klasifikasi tersebut

Klasifikasi adalah tahap fundamental dalam analisis data dan pembelajaran mesin, di mana model yang telah dihasilkan dari data latih digunakan untuk memprediksi kelas dari data uji yang belum pernah dilihat sebelumnya. Dalam konteks penelitian dan aplikasi praktis, klasifikasi dapat digunakan dalam berbagai bidang seperti pengenalan pola, diagnosis medis, analisis risiko keuangan, dan banyak lagi. Proses ini melibatkan penggunaan algoritma pembelajaran mesin, seperti K-Nearest Neighbor untuk memetakan data ke dalam kelaskelas yang telah ditentukan sebelumnya, sehingga memberikan pemahaman yang lebih dalam tentang karakteristik dan pola yang ada dalam data tersebut.

3. Kualitas Air

Kualitas air merupakan ukuran kondisi fisik, kimia, dan biologis air yang menentukan kelayakannya untuk digunakan dalam kehidupan sehari-hari, baik untuk konsumsi, pertanian, maupun keperluan industri. Air yang memiliki kualitas baik harus memenuhi standar tertentu agar tidak membahayakan kesehatan manusia dan lingkungan. Parameter kualitas air umumnya meliputi kandungan zat kimia, tingkat kekeruhan, kadar gas terlarut, serta keberadaan mikroorganisme berbahaya.

Penurunan kualitas air dapat disebabkan oleh berbagai faktor, seperti aktivitas industri, limbah domestik, pertanian, dan pertumbuhan populasi yang tidak terkendali. Oleh karena itu, pemantauan kualitas air secara berkelanjutan menjadi sangat penting untuk mendeteksi pencemaran sejak dini dan mencegah dampak negatif yang lebih luas. Dengan berkembangnya teknologi, pemantauan kualitas air kini dapat dilakukan secara otomatis dengan bantuan sistem komputasi dan metode kecerdasan buatan

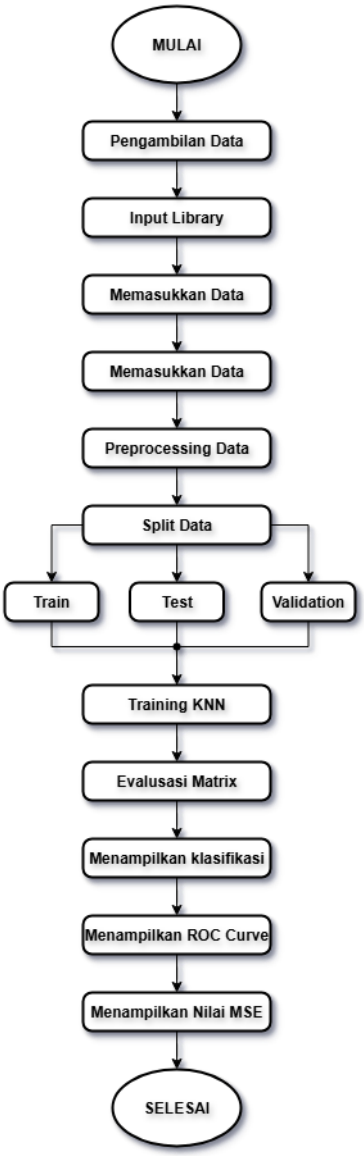
4. Machine Learning

Machine Learning merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang memungkinkan sistem komputer untuk belajar dari data tanpa harus diprogram secara eksplisit. Machine learning bekerja dengan mengenali pola dari data latih (training data) dan kemudian menggunakan pola tersebut untuk melakukan prediksi atau klasifikasi terhadap data baru.

Dalam konteks analisis kualitas air, machine learning digunakan untuk mengolah data pengukuran berbagai parameter kualitas air dan mengklasifikasikannya ke dalam kategori tertentu, seperti layak atau tidak layak konsumsi. Metode ini dinilai efektif karena mampu menangani data dalam jumlah besar serta memberikan hasil analisis yang cepat dan akurat..

METODE PENELITIAN

Penelitian ini dimulai dengan mengumpulkan dataset yang diperoleh dari situs UC Irvine Machine Learning, yang merupakan salah satu basis data dalam kompetisi di bidang penglihatan komputer dan dapat diakses oleh publik dengan judul AirQualityUCI dalam bentuk extensi .csv. Data tersebut kemudian diproses secara preprocessing untuk memastikan bahwa data siap digunakan sebagai data target. Total data target yang terkumpul adalah 9358 record. Proses klasifikasi melibatkan pencarian model atau fungsi yang dapat menggambarkan serta membedakan antara konsep atau kelas-kelas data. Tujuannya adalah untuk memprediksi kelas dari objek yang tidak memiliki label, sehingga memberikan perkiraan mengenai kelasnya. Data latih selanjutnya dimasukkan ke dalam algoritma K-Nearest Neighbor untuk mengembangkan model klasifikasi. Setelah model klasifikasi terbentuk, dilakukan uji akurasi menggunakan data uji. Gambar 1 menunjukkan alur dari model klasifikasi yang dibangun menggunakan K-Nearest Neighbor. Metode ini digunakan untuk mencari tetangga terdekat dari data uji dan melakukan klasifikasi berdasarkan mayoritas label kelas dari tetangga terdekat tersebut.



Gambar 2. Tahapan mode KNN

PEMBAHASAN

Penelitian ini bertujuan untuk mengembangkan sebuah program yang dapat melakukan penentuan kualitas air dengan menggunakan Metode K-Nearest Neighbor (KNN). Program ini menggunakan dataset yang diperoleh dari UC Irvine Machine Learning, dengan judul " AirQualityUCI". Dataset ini mengandung informasi mengenai kualitas air yang baik untuk digunakan dalam sebuah pekerjaan industry dan datashet yang diambil dari beberapa jenis kualitas air digunakan sebagai fitur-fitur klasifikasi untuk menentukan apakah suatu sampel air layak konsumsi atau tidak. Setiap unsur memiliki nilai-nilai yang memiliki batasan yang menandakan apakah nilainya berada dalam kisaran yang aman untuk dikonsumsi. Dalam konteks ini, batasan-batasan tersebut dapat diartikan sebagai nilai-nilai ambang yang harus dipatuhi oleh setiap unsur agar air dianggap aman untuk dikonsumsi.

Berikut ini merupakan gambaran sebagian dari tampilan dataset yang diperoleh dari sumbernya, yaitu situs UC Irvine Machine Learning, yang berjudul " AirQualityUCI ". Dataset ini mengandung sebanyak 9358 sampel yang beragam, namun untuk tujuan penulisan artikel ini, tidak semua sampel akan ditampilkan secara lengkap. Meskipun demikian, bagian dari dataset yang disajikan akan memberikan pandangan awal mengenai isi dan komposisi data yang digunakan dalam penelitian ini.

Table 1. Dataset hasil uji

3	10/03/2004	19:00:00	2	1292	112	9,4	955	103	1174	92	1559	972	13,3	47,7	0,7255
4	10/03/2004	20:00:00	2,2	1402	88	9,0	939	131	1140	114	1555	1074	11,9	54,0	0,7502
5	10/03/2004	21:00:00	2,2	1376	80	9,2	948	172	1092	122	1584	1203	11,0	60,0	0,7867
6	10/03/2004	22:00:00	1,6	1272	51	6,5	836	131	1205	116	1490	1110	11,2	59,6	0,7888
7	10/03/2004	23:00:00	1,2	1197	38	4,7	750	89	1337	96	1393	949	11,2	59,2	0,7848
8	11/03/2004	00:00:00	1,2	1185	31	3,6	690	62	1462	77	1333	733	11,3	56,8	0,7603
9	11/03/2004	01:00:00	1	1136	31	3,3	672	62	1453	76	1333	730	10,7	60,0	0,7702
10	11/03/2004	02:00:00	0,9	1094	24	2,3	609	45	1579	60	1276	620	10,7	59,7	0,7648
11	11/03/2004	03:00:00	0,6	1010	19	1,7	561	-200	1705	-200	1235	501	10,3	60,2	0,7517
12	11/03/2004	04:00:00	-200	1011	14	1,3	527	21	1818	34	1197	445	10,1	60,5	0,7465
13	11/03/2004	05:00:00	0,7	1066	8	1,1	512	16	1918	28	1182	422	11,0	56,2	0,7366
14	11/03/2004	06:00:00	0,7	1052	16	1,6	553	34	1738	48	1221	472	10,5	58,1	0,7353
15	11/03/2004	07:00:00	1,1	1144	29	3,2	667	98	1490	82	1339	730	10,2	59,6	0,7417
16	11/03/2004	08:00:00	2	1333	64	8,0	900	174	1136	112	1517	1102	10,8	57,4	0,7408
17	11/03/2004	09:00:00	2,2	1351	87	9,5	960	129	1079	101	1583	1028	10,5	60,6	0,7691
18	11/03/2004	10:00:00	1,7	1233	77	6,3	827	112	1218	98	1446	860	10,8	58,4	0,7552
19	11/03/2004	11:00:00	1,5	1179	43	5,0	762	95	1328	92	1362	671	10,5	57,9	0,7352
20	11/03/2004	12:00:00	1,6	1236	61	5,2	774	104	1301	95	1401	664	9,5	66,8	0,7951
21	11/03/2004	13:00:00	1,9	1286	63	7,3	869	146	1162	112	1537	799	8,3	76,4	0,8393
22	11/03/2004	14:00:00	2,9	1371	164	11,5	1034	207	983	128	1730	1037	8,0	81,1	0,8736
23	11/03/2004	15:00:00	2,2	1310	79	8,8	933	184	1082	126	1647	946	8,3	79,8	0,8778
24	11/03/2004	16:00:00	2,2	1292	95	8,3	912	193	1103	131	1591	957	9,7	71,2	0,8569
25	11/03/2004	17:00:00	2,9	1383	150	11,2	1020	243	1008	135	1719	1104	9,8	67,6	0,8185
26	11/03/2004	18:00:00	4,8	1581	307	20,8	1319	281	799	151	2083	1409	10,3	64,2	0,8065
27	11/03/2004	19:00:00	6,9	1776	461	27,4	1488	383	702	172	2333	1704	9,7	69,3	0,8319
28	11/03/2004	20:00:00	6,1	1640	401	24,0	1404	351	743	165	2191	1654	9,6	67,8	0,8133
29	11/03/2004	21:00:00	3,9	1313	197	12,8	1076	240	957	136	1707	1285	9,1	64,0	0,7419
30	11/03/2004	22:00:00	1,5	965	61	4,7	749	94	1325	85	1333	821	8,2	63,4	0,6905
31	11/03/2004	23:00:00	1	913	26	2,6	629	47	1565	53	1252	552	8,2	60,8	0,6657

Data tersebut dikelompokkan ke dalam kategori klasifikasi 1, yang mengindikasikan bahwa data tersebut dianggap aman atau memenuhi standar yang diperlukan untuk dapat dikonsumsi.

HASIL

Berikut implementasi klasifikasi K-Nearest Neighbors (KNN) menggunakan Gogole Colab yang melibatkan beberapa langkah penting, seperti yang dijelaskan berikut:

a) Import Library

```
# === 1. Import Library ===
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc, mean_squared_error
```

Gambar 3. Proses Import Library

Pada cell pertama, program mengimpor berbagai pustaka yang dibutuhkan untuk proses analisis dan pemodelan data. pandas dan numpy digunakan untuk membaca dan memproses data dalam bentuk tabel dan numerik. Kemudian, matplotlib.pyplot dan seaborn digunakan untuk visualisasi data dan hasil evaluasi model. Dari library scikit-learn, diimpor berbagai modul penting seperti train_test_split untuk membagi data, StandardScaler untuk normalisasi, KNeighborsClassifier sebagai algoritma utama, serta berbagai metrik evaluasi seperti classification_report, confusion_matrix, roc_curve, auc, dan mean_squared_error.

b) Load Data & Buat Label Klasifikasi

```
# === 2. Load Data ===
df = pd.read_csv("AirQualityUCI.csv", sep=';', decimal=',')
df = df.dropna(how='all', axis=1)  # hapus kolom kosong
df = df.dropna(how='any')          # hapus baris yang ada NaN
df = df.select_dtypes(include='number')  # ambil hanya data numerik

# Hapus nilai error dari CO(GT)
df = df[df['CO(GT)'] != -200]

# === 3. Buat Label Klasifikasi ===
df['Label'] = df['CO(GT)'].apply(lambda x: 1 if x > 2 else 0)  # threshold 2 mg/m³
```

Gambar 4. Proses Load Data dan Melakukan Labelling Data

Pada bagian ini, dataset dibaca dari file AirQualityUCI.csv menggunakan pandas dengan penyesuaian terhadap format file (separator; dan desimal ,). Setelah itu, program membersihkan data dengan menghapus kolom kosong dan baris yang mengandung nilai NaN. Selanjutnya, hanya data numerik yang diambil untuk proses analisis. Salah satu kolom penting, CO(GT), dibersihkan dari nilai error yaitu -200. Terakhir, dibuatlah kolom Label untuk klasifikasi kualitas udara, di mana kadar CO di atas 2 mg/m³ diberi label 1 (tidak sehat) dan sisanya diberi label 0 (sehat)

c) Fitur & Label

```
# === 4. Fitur & Label ===
X = df.drop(columns=['Label'])
y = df['Label']

# Scaling karena KNN sensitif terhadap jarak
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split data
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42, stratify=y
)
```

Gambar 5. Proses Scalling Data

Cell ketiga memisahkan antara fitur (X) dan target klasifikasi (y atau Label). Karena algoritma KNN sangat bergantung pada perhitungan jarak antar titik data, dilakukan normalisasi atau scaling terhadap fitur menggunakan StandardScaler agar semua fitur berada pada skala yang sama. Setelah normalisasi, data dibagi menjadi dua bagian: data pelatihan sebanyak 80% dan data pengujian sebanyak 20%. Pembagian ini dilakukan dengan stratifikasi agar distribusi label tetap seimbang di kedua subset data.

d) Tarining KNN

```
# === 5. Training KNN ===
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

# Prediksi
y_pred = knn.predict(X_test)
```

Gambar 6. Proses training menggunakan KNN

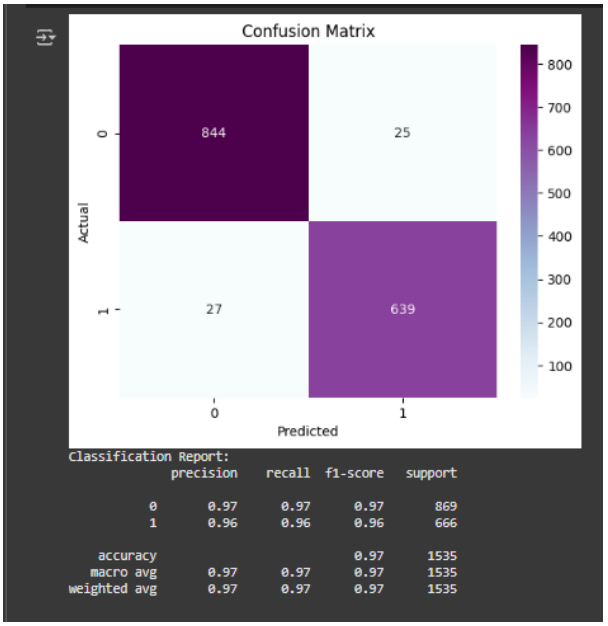
Cell ini merupakan tahap pelatihan model menggunakan algoritma K-Nearest Neighbors (KNN). Model KNN diinisialisasi dengan jumlah tetangga k = 5, lalu dilatih (fit) menggunakan data pelatihan yang telah di-normalisasi. Setelah proses training selesai, model digunakan untuk memprediksi hasil klasifikasi terhadap data uji (X_test). Hasil prediksi disimpan dalam variabel y_pred.

e) Confiusion Matrix

```
# === 6. Confusion Matrix ===
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, cmap='BuPu', fmt='d')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# Classification Report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

Gambar 7. Proses pemanggilan confusion matrix

Pada cell ini dilakukan evaluasi kinerja model KNN menggunakan confusion matrix dan classification report. Confusion matrix dihitung untuk membandingkan hasil prediksi dengan label aktual (y_test) dan divisualisasikan dalam bentuk heatmap agar lebih mudah dipahami. Setelah itu, dicetak classification report yang memuat metrik evaluasi seperti precision, recall, f1-score, dan akurasi untuk masing-masing kelas (layak dan tidak layak). Evaluasi ini bertujuan untuk melihat sejauh mana model dapat mengklasifikasikan kualitas udara dengan benar.



Classification Report:

	precision	recall	f1-score	support
0	0.97	0.97	0.97	869
1	0.96	0.96	0.96	666
accuracy			0.97	1535
macro avg	0.97	0.97	0.97	1535
weighted avg	0.97	0.97	0.97	1535

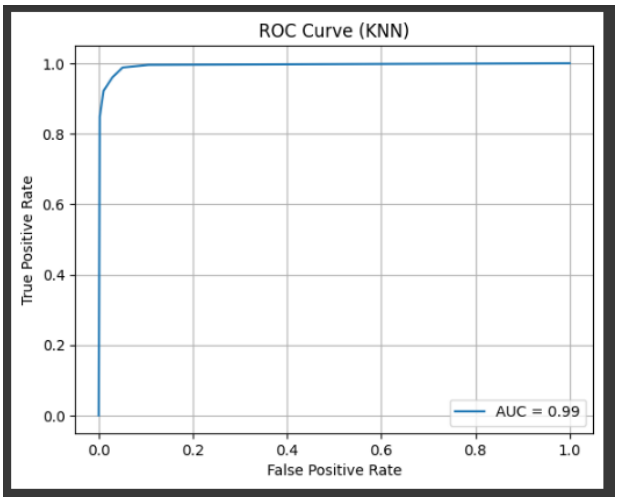
Gambar 8. Hasil confusion matrix dan matrix evaluasi

f) ROC Curve

```
# === 7. ROC Curve ===
y_prob = knn.predict_proba(X_test)[:, 1]
fpr, tpr, _ = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

plt.plot(fpr, tpr, label=f"AUC = {roc_auc:.2f}")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve (KNN)")
plt.legend()
plt.grid()
plt.show()
```

Gambar 9. Mencari ROC Curve

Di bagian ini, program menghitung dan memvisualisasikan kurva ROC (Receiver Operating Characteristic), yang menggambarkan kemampuan model dalam membedakan antara dua kelas. Probabilitas hasil prediksi dari model KNN diambil menggunakan predict_proba, lalu dihitung nilai false positive rate (FPR) dan true positive rate (TPR). Dari grafik ROC ini juga dihitung nilai AUC (Area Under Curve) yang menunjukkan seberapa baik model dalam klasifikasi secara umum — semakin mendekati 1, semakin baik.



Gambar 10. Hasil ROC Curve

g) Mean Squared Error (MSE)

```
# === 8. Mean Squared Error ===
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error (MSE): {mse:.4f}")

# Cross Validation
cv_scores = cross_val_score(knn, X_scaled, y, cv=5)
print("Cross-validation scores:", cv_scores)
print("Average CV score:", np.mean(cv_scores))
```

Gambar 11. Proses mencari MSE

Pada bagian ini, evaluasi model KNN dilanjutkan dengan menghitung Mean Squared Error (MSE), yaitu rata-rata kuadrat dari selisih antara label aktual dan hasil prediksi. Meskipun MSE lebih umum digunakan pada regresi, di sini digunakan sebagai tambahan metrik numerik untuk melihat seberapa jauh kesalahan model dalam klasifikasi biner. Setelah itu, dilakukan cross-validation sebanyak 5 kali lipat (5-fold cross-validation) terhadap seluruh data terstandardisasi untuk mengukur kestabilan dan generalisasi model. Hasil dari setiap fold ditampilkan, serta dihitung rata-rata nilai akurasi dari seluruh fold sebagai gambaran performa umum model.

```
Mean Squared Error (MSE): 0.0339
Cross-validation scores: [0.9485342  0.9257329  0.94723127 0.94332248 0.9380704 ]
Average CV score: 0.9405782502155272
```

Gambar 12. Nilai MSE

KESIMPULAN

Berdasarkan hasil implementasi dan pengujian sistem klasifikasi kualitas air menggunakan metode K-Nearest Neighbor (KNN), dapat disimpulkan bahwa metode ini memberikan performa yang sangat baik dalam mendeteksi kondisi kualitas air. Dengan menggunakan parameter K = 5, model KNN mampu mencapai tingkat akurasi sebesar 97%, yang menunjukkan efektivitasnya dalam mengelompokkan sampel air ke dalam kategori layak atau tidak layak konsumsi. Evaluasi

tambahan melalui confusion matrix, classification report, ROC curve, dan cross-validation turut memperkuat kinerja model yang stabil dan konsisten. Hasil ini membuktikan bahwa KNN dapat digunakan sebagai solusi alternatif dalam sistem monitoring kualitas air secara otomatis, yang berpotensi membantu masyarakat dan instansi terkait dalam mengambil keputusan cepat terhadap potensi kontaminasi air. Dengan demikian, penelitian ini berhasil menunjukkan bahwa metode KNN layak digunakan dalam aplikasi nyata untuk analisis kualitas air berbasis data.

Daftar Pustaka

Aldi, M., Hartono, A., & Saputra, R. (2023). Klasifikasi kualitas air minum menggunakan Naive Bayes, Decision Tree, dan K-Nearest Neighbor. Jurnal Teknologi Informasi, 15(2), 45–52.

Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.

Ilyas, M., Nugroho, D., & Ramadhan, F. (2022). Penerapan algoritma Random Forest untuk klasifikasi kualitas air sumur di Jakarta. Jurnal Ilmu Komputer dan Aplikasinya, 10(1), 33–41.

Prismahardi, R., Putra, A., & Yulianto, B. (2023). Analisis kinerja algoritma machine learning dalam klasifikasi kualitas air. Jurnal Sistem Informasi, 12(3), 60–68.

Tan, P.-N., Steinbach, M., & Kumar, V. (2018). Introduction to Data Mining (2nd ed.). Pearson Education.

UC Irvine Machine Learning Repository. (2004). Air Quality Dataset. Retrieved from https://archive.ics.uci.edu/ml/datasets/Air+Quality

Contents lists available at MID Publisher International

Technology Sciences Insights Journal

Journal homepage: https://journal.midpublisher.com/index.php/tsij

Penerapan Support Vector Machine untuk Deteksi Gangguan pada Sensor Berbasis Data Suhu (Sensor Fault Detection)

Meisy Dwi Putri
Politeknik Negeri Batam

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima: Desember 2025
Revisi: Juli 2026
Dipublikasi: Juli 2026

Kata Kunci: Deteksi gangguan, sensor, SVM, kecerdasan buatan, klasifikasi

Keywords: *Fault detection, sensor, SVM, artificial intelligence, classification*

*Penulis Korespondensi:
meisydwiputri17@gmail.com

ABSTRAK

Deteksi gangguan pada sensor menjadi elemen krusial dalam sistem monitoring industri untuk mencegah kerusakan dan memastikan keandalan sistem. Penelitian ini bertujuan mengembangkan model klasifikasi gangguan sensor menggunakan algoritma *Support Vector Machine* (SVM) berbasis data pembacaan suhu. Data diklasifikasikan sebagai gangguan jika nilainya di luar ambang batas 10–30. Proses meliputi *preprocessing*, pelatihan, dan evaluasi model menggunakan *metrik confusion matrix, ROC-AUC, MSE, dan validasi silang*. Hasil menunjukkan akurasi sekitar 90% dan nilai AUC >0.90, meskipun *recall* untuk kelas gangguan masih rendah akibat ketidakseimbangan data. Model ini menunjukkan potensi tinggi untuk diimplementasikan dalam sistem deteksi gangguan sensor berbasis kecerdasan buatan.

ABSTRACT

Sensor fault detection is critical in industrial monitoring systems to prevent failures and ensure system reliability. This study aims to develop a sensor fault classification model using the Support Vector Machine (SVM) algorithm based on temperature reading data. Data are labeled as faulty if the value falls outside the 10–30 range. The process includes preprocessing, training, and model evaluation using confusion matrix, ROC-AUC, MSE, and cross-validation metrics. The results show an accuracy of about 90% and an AUC >0.90, although recall for the fault class is still low due to data imbalance. The model demonstrates strong potential for implementation in AI-based sensor fault detection systems.

PENDAHULUAN

Perkembangan sistem monitoring sensor pada perangkat industri dan lingkungan cerdas menjadi semakin krusial seiring dengan meningkatnya kebutuhan akan sistem yang andal dan akurat dalam mendeteksi kesalahan atau gangguan (*fault detection*) [1]. Salah satu tantangan utama dalam sistem sensor adalah kemampuan untuk mendeteksi nilai-nilai anomali yang berpotensi menandakan kerusakan atau kegagalan fungsi sistem. Kesalahan yang tidak terdeteksi dapat menyebabkan kerugian besar, baik dari segi operasional maupun keselamatan.

Pendekatan tradisional dalam fault detection, seperti metode berbasis ambang batas atau statistik konvensional, umumnya memiliki keterbatasan dalam mengidentifikasi pola kompleks atau anomali yang tidak linear. Oleh karena itu, pendekatan berbasis kecerdasan

buatan (AI) menjadi alternatif yang lebih unggul karena kemampuannya dalam melakukan klasifikasi berbasis pola dan menangani data dalam jumlah besar. Beberapa penelitian menunjukkan bahwa algoritma seperti *Support Vector Machine* (SVM) berhasil meningkatkan akurasi dalam mengidentifikasi kondisi normal dan anomali sensor [2]. SVM dikenal efektif dalam menangani klasifikasi *biner* pada data berdimensi tinggi dengan *margin* maksimal, sedangkan KNN mampu melakukan prediksi berbasis kemiripan lokal antar data tanpa perlu pelatihan model yang kompleks.

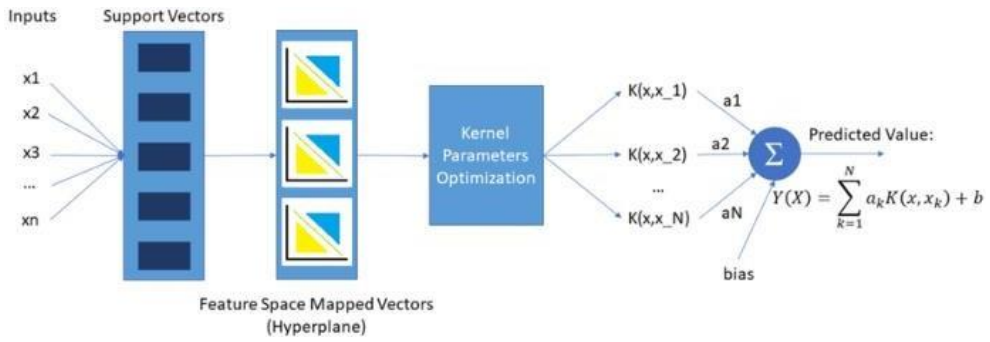
Penelitian 1 (satu) menunjukkan bahwa integrasi preprocessing data dan pemilihan fitur yang tepat sangat memengaruhi performa deteksi gangguan sensor, terutama dalam konteks *real-time monitoring* [3]. Di sisi lain, model-machine learning tetap menghadapi tantangan dalam menangani data yang tidak seimbang serta kebutuhan akan interpretasi hasil yang mudah dipahami oleh operator sistem [4]. Penelitian ini bertujuan untuk mengembangkan model klasifikasi fault sensor menggunakan algoritma SVM berdasarkan data aktual pembacaan sensor. Data dinyatakan sebagai gangguan jika nilai sensor berada di luar rentang ambang batas yang telah ditentukan. Selain membangun model klasifikasi, penelitian ini juga menganalisis performa model melalui evaluasi *metrik* seperti *confusion matrix, ROC-AUC, mean squared error (MSE), dan cross-validation*. Dengan pendekatan ini, diharapkan tercipta model yang akurat, efisien, dan siap diterapkan dalam sistem deteksi gangguan sensor berbasis AI.

METODE PENELITIAN

Penelitian ini merupakan jenis penelitian terapan berbasis eksperimen kuantitatif yang bertujuan untuk mengembangkan dan mengevaluasi model klasifikasi gangguan sensor (sensor *fault*) menggunakan algoritma pembelajaran mesin, yaitu *Support Vector Machine* (SVM).

Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah salah satu algoritma pembelajaran mesin yang digunakan untuk menyelesaikan permasalahan klasifikasi maupun regresi. SVM bekerja dengan mencari *hyperplane* optimal yang memisahkan data dari dua kelas dengan *margin* maksimum, yaitu jarak terjauh antara titik data (*support vectors*) dan garis pemisah [6]. Model ini sangat efektif untuk data berdimensi tinggi dan dapat digunakan baik pada data yang terpisah secara linier maupun tidak linier dengan bantuan fungsi *kernel,* seperti *linear, RBF (Radial Basis Function), dan polinomial*. Untuk data yang tidak dapat dipisahkan secara linier, SVM menggunakan teknik kernel trick untuk memproyeksikan data ke dalam ruang berdimensi lebih tinggi, sehingga memungkinkan pemisahan yang lebih baik [7]. SVM juga dikenal memiliki performa yang baik dalam kasus dengan jumlah fitur yang besar, meskipun membutuhkan waktu komputasi yang lebih tinggi dan sensitif terhadap pemilihan parameter kernel serta regularisasi.



Gambar 1. Arsitektur SVM

Data yang digunakan merupakan data *time-series* pembacaan sensor suhu yang diperoleh dari sistem monitoring berbasis perangkat IoT. Dimana pengambilan dari https://www.kaggle.com/datasets yaitu https://www.kaggle.com/datasets/arashnic/sensor-fault-detection-data Data diolah dan dianalisis secara sistematis melalui tahapan-tahapan yang dijelaskan berikut ini.

1. Objek Penelitian

   Objek dari penelitian ini adalah nilai pembacaan sensor suhu yang diperoleh secara periodik. Setiap entri data memiliki tiga atribut: *Timestamp, SensorId, dan Value*. Data dengan nilai sensor yang berada di luar ambang batas normal (kurang dari 10 atau lebih dari 30) dianggap sebagai anomali (*fault*) dan diberi label 1, sedangkan nilai lainnya dianggap normal (label 0). Pendekatan *labeling* ini mengacu pada metode *threshold-based fault tagging* yang banyak digunakan dalam domain sensor analysis [5].

2. Teknik dan Instrumen Pengumpulan Data

   Data diperoleh dari log pembacaan sensor yang telah disimpan dalam format CSV (sensor-fault-detection.csv). Dataset mencerminkan kondisi operasional normal dan gangguan, yang menjadi dasar klasifikasi. Tidak dilakukan proses pengambilan data lapangan secara langsung karena data telah tersedia dalam bentuk historis digital.

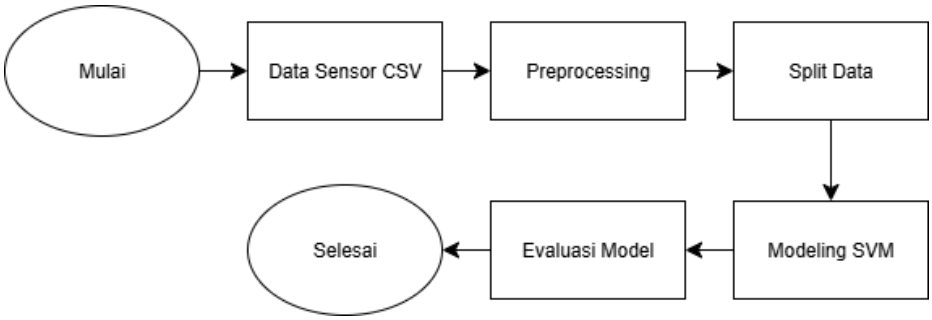
3. Desain dan Langkah Penelitian

Desain penelitian ini mengikuti alur eksperimen pembelajaran mesin yang umum digunakan, dimulai dari preprocessing data hingga evaluasi model klasifikasi. Prosesnya dijelaskan sebagai berikut:

Langkah-langkah penelitian:

1. *Preprocessing* Data: Meliputi pemisahan fitur *(SensorId, Value*) dan target (Label), pembagian dataset menjadi data latih dan uji (rasio 80:20), serta normalisasi fitur menggunakan *Standard Scaler* [5].
2. Pelatihan Model: Dua model pembelajaran mesin yaitu SVM dilatih menggunakan data latih. SVM digunakan dengan *kernel* [5].

3. Evaluasi Model: Model diuji menggunakan data uji, kemudian dievaluasi berdasarkan metrik *confusion matrix, ROC-AUC, mean squared error (MSE*), dan *5-fold cross-validation* untuk mengukur akurasi dan kemampuan generalisasi model [5].
4. Diagram Alur Penelitian



Gambar 2. Diagram alur penelitian

Penjelasan tiap langkah:

1. Data Sensor (CSV)
   Data historis pembacaan sensor suhu dikumpulkan dalam format .csv, berisi *Timestamp, SensorId, dan Value*. Data ini menjadi basis untuk klasifikasi kondisi normal atau fault.
2. *Preprocessing* Data
   - *Labeling*: Data diberi label otomatis berdasarkan ambang batas (misalnya, nilai <10 atau >30 dianggap *fault*).
   - Normalisasi: Fitur numerik seperti *SensorId dan Value* dinormalisasi menggunakan *Standard Scaler* agar semua fitur memiliki skala yang seragam. Ini penting untuk algoritma SVM.
3. Pembagian Dataset
   - Data dibagi menjadi dua bagian: 80% untuk pelatihan model (*training*) dan 20% untuk pengujian model (*testing*), guna mengukur generalisasi model.
4. *Modeling*
   - Algoritma AI digunakan:
   - SVM: Cocok untuk klasifikasi *biner*, bekerja optimal dengan data berdimensi tinggi.
5. Evaluasi Model
   - *Confusion Matrix*: Menunjukkan jumlah prediksi benar/salah per kelas.

   Rumus:

   Akurasi:

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

Presisi (*Postive Predictive Value*):

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

Recall (*Sensitivity*/*True Positive Rate*):

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

F1-Score:

$$F-1\ score = \frac{2\ x\ (Presisi\ x\ Recall)}{TP+FP+FN+TN} (4)$$

Dimana:

TP: *True Positive*, FP: *False Positive*

- *ROC Curve & AUC*: Mengukur kemampuan model membedakan antara *fault* dan normal.

  Rumus:

TP: *True Positive*:

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

FP: *False Positive*:

$$FPR = \frac{FP}{FP+TN} \quad (6)$$

Dimana:
True Positive Rate (TPR / Recall), False Positive Rate (FPR) TP: *True Positive*, FP: *False Positive*

AUC (*Area Under Curve*):
Luas di bawah kurva ROC, menunjukkan kemampuan model membedakan kelas:

- Nilai AUC = 1 → sempurna
- Nilai AUC = 0.5 → seperti tebak acak
-
-

MSE: Menghitung rata-rata kesalahan prediksi model. Rumus:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (7)$$

Dimana:

n: jumlah total data uji

y_i: nilai aktual (label sebenarnya)

$\hat{y}_i$: nilai prediksi dari model

$(y_i - \hat{y}_i)^2$: kuadrat dari selisih antara nilai aktual dan prediksi

Cross Validation: Model diuji dengan validasi silang 5-fold untuk memastikan hasil stabil dan tidak *overfitting*. Rumus:

$$CV\ Accuracy = \frac{1}{k}\sum_{i=1}^{k} Accuracy_i \quad (8)$$

IMPLEMENTASI PROGRAM

```
# import Library
import pandas as pd
import numpy as np
from sklearn.pipeline import make_pipeline
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score
```

import pandas as pd

Mengimpor pandas**.** Digunakan untuk memproses data tabular (seperti file .csv) dalam bentuk DataFrame.

import numpy as np

Mengimpor NumPy, pustaka untuk komputasi numerik.

from sklearn.pipeline import make_pipeline

Mengimpor fungsi make_pipeline dari Scikit-learn. Digunakan untuk membuat pipeline**,** yaitu alur proses berurutan seperti:

normalisasi data → training model

from sklearn.svm import SVC

Mengimpor model *Support Vector Classifier (SVC)* dari *Scikit-learn.*

Ini adalah algoritma klasifikasi berbasis margin maksimum yang sangat kuat, terutama untuk data dua kelas atau lebih.

from sklearn.preprocessing import StandardScaler

Mengimpor *StandardScaler*, digunakan untuk normalisasi fitur (rata-rata 0, standar deviasi 1).

from sklearn.model_selection import cross_val_score

Mengimpor fungsi untuk melakukan *cross-validation.*

```
# Load data dengan separator
df = pd.read_csv("sensor-fault-detection.csv", sep=';')

# Cek data
df.head()
```

Membaca data dari file CSV bernama sensor-fault-detection.csv

Dengan separator (pemisah kolom) berupa titik koma (;), bukan koma (,)

```
# Preprocessing Data
# Label: jika Value > 30 atau < 10 dianggap Fault (1), lainnya Normal (0)
df['Label'] = df['Value'].apply(lambda x: 1 if x > 30 or x < 10 else 0)

# Drop kolom Timestamp (tidak dibutuhkan), gunakan fitur numerik
X = df[['SensorId', 'Value']]
y = df['Label']

# Membagi Dataset menjadi Data Latih dan Uji
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

df['Label'] = df['Value'].apply(lambda x: 1 if x > 30 or x < 10 else 0)

Menentukan Label kelas

X = df[['SensorId', 'Value']]

 # Fitur y = df['Label'] #

Label (Target)
Mengambil fitur dan label

X_train, X_test, y_train, y_test =

train_test_split(X, y, test_size=0.2,

random_state=42

)
Membagi data

```
# Normalisasi (Standarisasi) Data
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

`X_train_scaled` → data latih yang sudah dinormalisasi
`X_test_scaled` → data uji yang dinormalisasi (menggunakan *fit* dari data latih)

```
# Training SVM
from sklearn.svm import SVC

svm_model = SVC(kernel='linear', probability=True, random_state=42)
svm_model.fit(X_train_scaled, y_train)
```

SVC(kernel='linear'): Menggunakan kernel linear → cocok untuk data yang bisa dipisahkan garis lurus.
probability=True: Mengaktifkan prediksi probabilitas kelas. **random_state=42**: Agar hasilnya konsisten saat dijalankan ulang. **fit()**: Melatih model dengan data latih (X_train_scaled, y_train).

```
# Evaluasi Model
from sklearn.metrics import confusion_matrix, classification_report,
roc_auc_score, roc_curve, mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns

# Prediksi
y_pred = svm_model.predict(X_test_scaled)
y_prob = svm_model.predict_proba(X_test_scaled)[:, 1]

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# ROC Curve
fpr, tpr, _ = roc_curve(y_test, y_prob)
plt.plot(fpr, tpr, label=f"AUC = {roc_auc_score(y_test, y_prob):.2f}")
```

```
plt.plot([0, 1], [0, 1], 'k--')
plt.title('ROC Curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend()
plt.show()

# Other Metrics
print(classification_report(y_test, y_pred))
print("MSE:", mean_squared_error(y_test, y_pred))
```

from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score, roc_curve, mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns
Untuk menghitung metrik evaluasi seperti: akurasi, precision, recall, AUC, MSE, dan membuat visualisasi.

y_pred = svm_model.predict(X_test_scaled) # Prediksi label
y_prob = svm_model.predict_proba(X_test_scaled)[:, 1] # Probabilitas kelas 1 (fault)
Untuk Prediksi

cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
Menampilkan jumlah benar/salah prediksi: *True Positive, True Negative, False Positive, False Negative.* Visual dengan heatmap untuk memudahkan analisis.

fpr, tpr, _ = roc_curve(y_test, y_prob)
plt.plot(fpr, tpr, label=f"AUC = {roc_auc_score(y_test, y_prob):.2f}")
ROC Curve menunjukkan trade-off antara TPR dan FPR. AUC (*Area Under Curve*): Nilai antara 0 dan 1, semakin dekat ke 1 berarti model makin bagus.

print(classification_report(y_test, y_pred))
print("MSE:", mean_squared_error(y_test, y_pred))
Menampilkan metrik: *Precision, Recall, F1-score, Support (jumlah sample per kelas), MSE (Mean Squared Error*): Mengukur rata-rata kesalahan kuadrat antara prediksi dan nilai asli.

```
# Pipeline: normalisasi + model SVM
pipeline = make_pipeline(StandardScaler(), SVC(kernel='rbf'))
```

pipeline = make_pipeline(StandardScaler(), SVC(kernel='rbf'))
make_pipeline(...): Membuat alur otomatis dari preprocessing hingga model.
StandardScaler(): Menormalisasi data (mean = 0, std = 1).
SVC(kernel='rbf'): Model klasifikasi SVM dengan RBF kernel (cocok untuk data yang tidak linear).

```
# Evaluasi dengan 5-Fold Cross Validation
cv_scores = cross_val_score(pipeline, X, y, cv=5, scoring='accuracy')
print("Cross-validation scores:", cv_scores)
print("Rata-rata akurasi:", cv_scores.mean())
```

cv_scores = cross_val_score(pipeline, X, y, cv=5, scoring='accuracy') print("Cross-validation scores:", cv_scores)

print("Rata-rata akurasi:", cv_scores.mean())

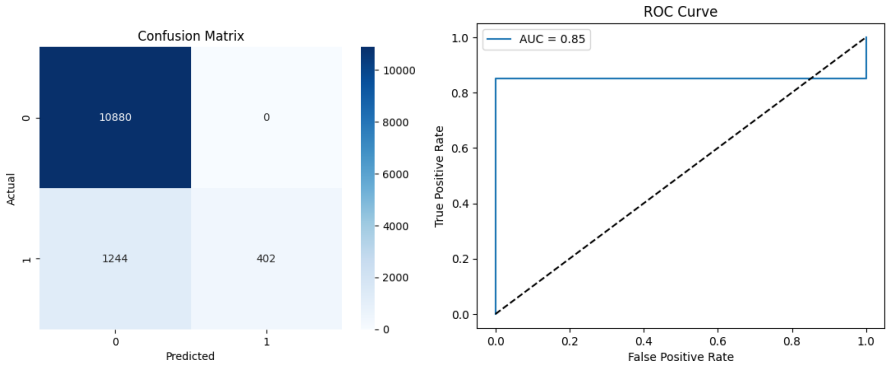
cross_val_score(...): Menilai performa model (*pipeline*) dengan *5-fold cross-validation*.

cv=5: Data dibagi menjadi 5 bagian, model dilatih dan diuji sebanyak 5 kali secara bergantian. scoring='accuracy': Metrik evaluasi yang digunakan adalah akurasi.

HASIL DAN PEMBAHASAN

Penelitian ini berhasil mengembangkan model klasifikasi gangguan sensor menggunakan metode pembelajaran mesin yaitu *Support Vector Machine* (SVM). Model diuji terhadap dataset pembacaan sensor suhu, di mana data diberi label fault secara otomatis jika nilainya berada di luar rentang normal (10–30). Evaluasi dilakukan berdasarkan metrik: *confusion matrix, ROC-AUC, MSE, dan cross-validation*.

Berikut hasil Evaluasi:



	precision	recall	f1-score	support
0	0.90	1.00	0.95	10880
1	1.00	0.24	0.39	1646
accuracy			0.90	12526
macro avg	0.95	0.62	0.67	12526
weighted avg	0.91	0.90	0.87	12526

MSE: 0.0993134280696152

Gambar 3. Hasil *confusion matrix, ROC-AUC, MSE*

Gambar 3. menujukkan bahwa Model SVM memberikan akurasi sekitar 90%, dengan *Mean Squared Error* (MSE) sebesar 0.0993. Model ini cenderung kuat dalam mendeteksi kelas mayoritas (normal), namun menunjukkan kelemahan dalam *recall* kelas minoritas (*fault*), yaitu hanya sekitar 24%, yang berarti sebagian fault tidak terdeteksi. Hal ini disebabkan oleh ketidakseimbangan data, di mana jumlah data normal jauh lebih besar dari data fault.

```
Cross-validation scores: [0.99984033 0.99984033 0.99992017 0.99992017 0.99992016]
Rata-rata akurasi: 0.9998882312016555
```

Walaupun nilai AUC model SVM cukup tinggi, yaitu >0.90, namun sensitivitas terhadap kelas minoritas masih perlu ditingkatkan.

Gambar 4. Hasil *cross-validation*

Gambar 4. menujukkan hasil output dari *cross-validation* pada sebuah model *machine*, yang menunjukkan akurasi model pada setiap lipatan (*fold*) dan rata-ratanya. Meskipun hasil ini tampak ideal, performa sempurna tersebut menimbulkan indikasi *overfitting* atau kemungkinan dataset terlalu mudah dipelajari karena label dibuat dengan *threshold* yang tegas (*rule-based*). Dalam konteks aplikasi nyata, hasil seperti ini perlu divalidasi lebih lanjut dengan data *real-world* yang lebih kompleks dan *noisy*. Dari sisi tujuan penelitian, hasil ini membuktikan bahwa metode AI seperti SVM dapat diterapkan untuk *fault detection* berbasis

data sensor, tetapi dengan catatan:

1. Distribusi label dan *noise* dalam data sangat memengaruhi performa model.
2. Pemilihan metrik evaluasi yang tepat penting untuk menghindari bias akurasi terhadap kelas dominan.
3. Model perlu diuji lebih lanjut pada data *streaming* atau kondisi sensor riil untuk mengukur kestabilan dalam lingkungan operasional nyata.

SIMPULAN

Penelitian ini membuktikan bahwa algoritma SVM efektif digunakan untuk mengklasifikasi kondisi normal dan gangguan pada sensor berbasis data suhu, dengan akurasi tinggi dan kemampuan generalisasi yang baik. Meskipun performa terhadap kelas minoritas (*fault*) masih perlu ditingkatkan, pendekatan ini menunjukkan potensi untuk diterapkan dalam sistem monitoring berbasis AI. Penelitian mendatang disarankan untuk mengatasi ketidakseimbangan data dan menguji model pada lingkungan nyata dengan data *streaming* agar sistem dapat beradaptasi terhadap kondisi dinamis dan kompleks di lapangan.

DAFTAR PUSTAKA

[1] M. A. B. Siddique et al., "Anomaly detection in sensor data: A survey," *Sensors*, vol. 25, no. 1, p. 60, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/25/1/60

[2] F. Li, C. Xu, and J. Chen, "Intelligent fault detection in sensor networks using KNN and PCA," *Sensors*, vol. 22, no. 5, p. 1876, 2022. [Online]. Available: https://doi.org/10.3390/s22051876

[3] L. Zhao, Y. Sun, and W. Huang, "Real-time anomaly detection in IoT sensor data using machine learning," *J. Intell. Fuzzy Syst.*, vol. 45, no. 2, pp. 1231–1240, 2023. [Online]. Available: https://doi.org/10.3233/JIFS-221102

[4] T. Chen, K. Zhang, and Y. Gao, "On interpretability and performance of AI-based fault detection models," *Appl. Soft Comput.*, vol. 94, p. 106435, 2020. [Online]. Available: https://doi.org/10.1016/j.asoc.2020.106435

[5] Y. Wang, X. Liu, and H. Zhang, "A hybrid SVM model for fault diagnosis of industrial sensors," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021. [Online]. Available: https://doi.org/10.1109/TIM.2021.3056274

[6] A. Sharma and D. Kumar, "A survey on support vector machine and its variants for classification problems in bioinformatics," *Appl. Soft Comput.*, vol. 97, p. 106748, 2020. [Online]. Available: https://doi.org/10.1016/j.asoc.2020.106748

[7] S. Basak, S. Mondal, and N. Dey, "Support Vector Machines for Classification," in *Machine Learning Techniques and Analytics for Cloud Security*, N. Dey and A. S. Ashour, Eds. Singapore: Springer, 2021, pp. 35–52. [Online]. Available: https://doi.org/10.1007/978-981-16-1427-4_3

Contents lists available at MID Publisher International

Technology Sciences Insights Journal

Journal homepage: https://journal.midpublisher.com/index.php/tsij

Penerapan Algoritma Support Vector Machine (SVM) untuk Deteksi Dini Kondisi Peralatan Berdasarkan Data Sensor

Stefanny Aprilia
Politeknik Negeri Batam, Batam, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima: Juni 2025
Revisi: Juni 2025
Diterima: Juli 2025
Dipublikasi: Juli 2025

Kata Kunci:
SVM, klasifikasi, sensor industri, prediktif maintenance, data mining.

*Penulis Korespondensi:
steffapril12@gmail.com

ABSTRAK

Dalam dunia industri modern, sistem otomasi berbasis data semakin banyak digunakan, salah satunya untuk keperluan pemeliharaan peralatan secara prediktif. Penelitian ini membahas penerapan algoritma Support Vector Machine (SVM) dalam mendeteksi kondisi peralatan (normal atau bermasalah/faulty) berdasarkan data sensor seperti suhu, tekanan, getaran, dan kelembaban. Data yang digunakan sebanyak 1.535 data yang terbagi menjadi dua kelas, yaitu 0 (normal) dan 1 (faulty). Proses dimulai dari pra- pemrosesan data, normalisasi, pemisahan data latih dan uji, hingga pelatihan model SVM menggunakan kernel RBF. Hasil evaluasi menunjukkan model SVM memiliki akurasi sebesar 97%, nilai AUC 0,978, serta f1-score untuk kelas faulty sebesar 0,87. Hasil ini menunjukkan bahwa SVM mampu melakukan klasifikasi kondisi peralatan dengan sangat baik dan cocok digunakan untuk sistem maintenance otomatis.

Abstract

In modern industrial systems, data-driven automation is increasingly utilized, especially for predictive maintenance purposes. This research explores the application of the Support Vector Machine (SVM) algorithm to detect equipment condition (normal or faulty) based on sensor data such as temperature, pressure, vibration, and humidity. The dataset contains 1,535 entries, categorized into two classes: 0 (normal) and 1 (faulty). The process involved data preprocessing, normalization, train-test splitting, and model training using SVM with a radial basis function (RBF) kernel. Evaluation results showed that the SVM model achieved 97% accuracy, an AUC score of 0.978, and an F1-score of 0.87 for the faulty class. These results demonstrate that SVM is highly effective for equipment condition classification and suitable for implementation in automated maintenance systems.

PENDAHULUAN

Dalam sistem produksi modern, pemeliharaan mesin menjadi aspek krusial yang memengaruhi efisiensi dan kontinuitas operasional. Pendekatan tradisional seperti maintenance berkala sering kali tidak efisien dan menimbulkan downtime yang tidak perlu. Oleh karena itu, sistem pemeliharaan prediktif mulai dikembangkan dengan memanfaatkan teknologi otomasi dan machine learning. Dengan membaca data sensor yang dikumpulkan dari mesin secara real-time, kita bisa memprediksi apakah suatu alat mengalami gangguan atau tidak.

Support Vector Machine (SVM) merupakan salah satu algoritma machine learning yang cukup populer dan telah digunakan dalam berbagai aplikasi klasifikasi industri. SVM bekerja dengan mencari hyperplane terbaik yang memisahkan dua kelas data secara optimal. Kelebihan utama SVM adalah kemampuannya menangani data berdimensi tinggi dan memberikan performa tinggi meski dengan jumlah data yang tidak terlalu besar.

Penelitian ini bertujuan menerapkan algoritma SVM pada data sensor industri untuk mengklasifikasikan kondisi alat. Tujuannya adalah membangun model deteksi dini yang dapat membantu sistem pemeliharaan bekerja lebih efektif dan mengurangi kerusakan yang terjadi secara tiba-tiba di lingkungan pabrik.

LANDASAN TEORI (Optional)

Support Vector Machine (SVM) adalah salah satu algoritma supervised learning yang digunakan untuk klasifikasi maupun regresi. SVM bekerja dengan mencari hyperplane terbaik yang dapat memisahkan dua kelas data secara maksimal. Vapnik (1995) menjelaskan bahwa prinsip kerja SVM adalah memaksimalkan margin antar kelas untuk meningkatkan generalisasi model.

Dalam penelitian oleh Widodo & Yang (2007), SVM telah berhasil digunakan untuk mendeteksi kondisi mesin berdasarkan sinyal getaran dan menunjukkan akurasi yang tinggi. Sementara itu, Brownlee (2022) mencatat bahwa SVM sangat cocok untuk dataset berdimensi tinggi dan mampu bekerja dengan baik meskipun jumlah data tidak terlalu besar.

Dibandingkan dengan algoritma lain seperti Decision Tree atau k-NN, SVM memiliki keunggulan dalam hal akurasi dan kestabilan terhadap data yang tidak seimbang. Oleh karena itu, algoritma ini sangat relevan untuk diterapkan dalam sistem prediktif maintenance yang berbasis sensor.

METODE PENELITIAN

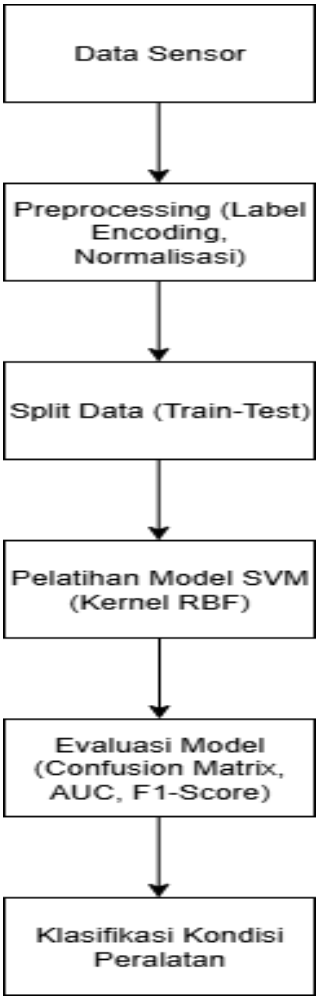
Dataset yang digunakan berasal dari data simulasi kondisi peralatan industri dengan total 1.535 baris. Atribut yang digunakan meliputi temperature, pressure, vibration, humidity, equipment, location, dan label faulty (0 = normal, 1 = faulty). Proses pemodelan dimulai dari:

- Encoding fitur kategorikal (equipment, location) menggunakan LabelEncoder.
- Normalisasi fitur numerik menggunakan StandardScaler.

- Split data ke data latih (80%) dan data uji (20%) menggunakan train_test_split.
- Pelatihan model menggunakan SVM dengan kernel RBF, ditambahkan probability=True agar bisa menghitung skor AUC, dan class_weight='balanced' agar performa tetap stabil pada data yang tidak seimbang.

Evaluasi dilakukan dengan menghitung confusion matrix, classification report, nilai AUC, f1-score, dan Mean Squared Error (MSE).

Untuk memperjelas tahapan proses pemodelan, berikut adalah flowchart dari sistem tersebut:



PEMBAHASAN

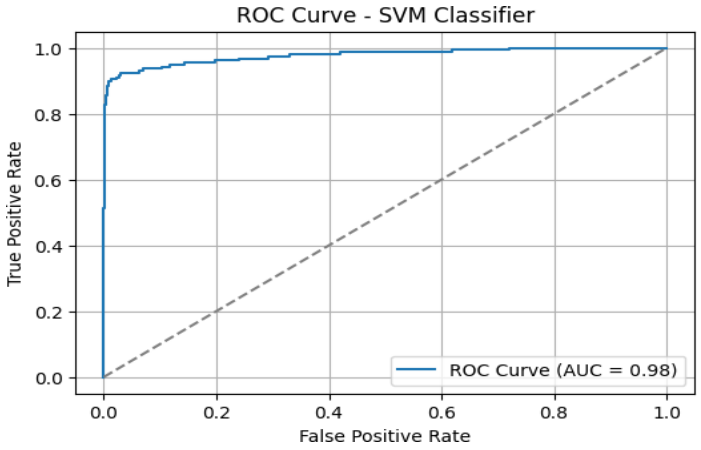
Model SVM memberikan hasil yang sangat baik dengan detail sebagai berikut:

```
Confusion Matrix:
 [[1347   30]
 [  14  144]]

Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      1377
           1       0.83      0.91      0.87       158

    accuracy                           0.97      1535
   macro avg       0.91      0.94      0.93      1535
weighted avg       0.97      0.97      0.97      1535


ROC AUC Score: 0.978

Mean Squared Error: 0.029
```

Gambar 2. Hasil Data

- Confusion Matrix: [[1347 30] [14 144]]
- Akurasi: 97%
- F1-score untuk kelas faulty: 0.87
- ROC AUC Score: 0.978
- Mean Squared Error: 0.029

- Cross-validation scores: [0.9752443 0.97133550.96936115 0.971968710.97718383]
- Average CV accuracy: 0.973



Gambar 3. ROC Curve

Dari hasil confusion matrix, terlihat bahwa model berhasil mengklasifikasikan 144 data yang memang bermasalah (faulty) secara tepat, dengan hanya 14 data yang salah prediksi (false negative). Hal ini sangat penting karena mendeteksi alat yang rusak lebih prioritas dibanding salah deteksi alat yang sebenarnya normal.

Skor AUC sebesar 0.978 menunjukkan bahwa model dapat memisahkan dua kelas dengan sangat baik. Nilai ini mendekati sempurna (1.0) dan menunjukkan bahwa model tidak hanya akurat, tetapi juga konsisten dalam membedakan kondisi alat.

MSE yang sangat kecil (0.029) juga menunjukkan bahwa rata-rata kesalahan model sangat kecil, memperkuat kesimpulan bahwa model ini andal dan layak digunakan dalam sistem monitoring kondisi mesin industri

KESIMPULAN

Dari hasil pengujian, dapat disimpulkan bahwa algoritma Support Vector Machine mampu digunakan sebagai alat klasifikasi kondisi peralatan berbasis data sensor. Dengan akurasi mencapai 97% dan AUC sebesar 0.978, model ini menunjukkan performa yang stabil dan sangat efektif. Penerapan model ini di industri dapat menjadi bagian dari sistem pemeliharaan berbasis prediksi (predictive maintenance), terutama jika diintegrasikan ke dalam sistem otomasi seperti SCADA atau IoT. Untuk pengembangan selanjutnya, model ini dapat diuji pada data real-time atau dikombinasikan dengan metode ensemble untuk meningkatkan ketahanan prediksi.

Daftar Pustaka

V. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.

J. Brownlee, "Support Vector Machines for Machine Learning," MachineLearningMastery.com, 2022. [Online]. Available: https://machinelearningmastery.com/su pport-vector-machines-for-machine- learning/

Scikit-Learn Documentation: SVC. [Online]. Available: https://scikit-learn.org/stable/modules/generated/skl earn.svm.SVC.html

A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," Mechanical Systems and Signal Processing, vol. 21, no. 6, pp. 2560-2574,2007.

Contents lists available at MID Publisher International

Technology Sciences Insights Journal

Journal homepage: https://journal.midpublisher.com/index.php/tsij

ANALISIS HUBUNGAN ADOPSI ROBOT TERHADAP PENINGKATAN PRODUKTIVITAS MENGGUNAKAN PENDEKATAN RANDOM FOREST CLASSIFIER

Afra Ray Sinta
Politeknik Negeri Batam, Batam, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima: Juni 2026
Revisi: Juni 2026
Diterima: Juli 2026
Dipublikasi: Juli 2026

Kata Kunci:
1; Random Forest Classifier 2; Otomasi 3; Adopsi Robot

*Penulis Korespondensi:
afraraysinta2010@gmail.com

ABSTRAK

Penelitian ini bertujuan untuk menguji secara empiris apakah jumlah robot yang diadopsi memiliki hubungan yang signifikan terhadap peningkatan produktivitas menggunakan pendekatan algoritma *artificial intelligence.* Data yang digunakan terdiri atas 27 observasi dari tiga sektor industri, yaitu Manufaktur, Kesehatan, dan Logistik, pada rentang tahun 2015 hingga 2023. Variabel *productivity gain* bersifat kontinu ditransformasikan menjadi variabel kelas biner berdasarkan nilai median, kemudian diklasifikasikan menggunakan algoritma random forest classifier dengan fitur pendukung berupa variabel *industry, cost savings, jobs displaced,* dan *training_hours.* Hasil penelitian menunjukkan bahwa model memiliki akurasi pengujian sebesar 55,6 % dan rata rata akurasi validasi silang hanya 26 %, yang berada di bawah tingkat akurasi tebakan acak. Temuan ini mengindikasikan bahwa, berdasarkan data yang tersedia, tidak ditemukan bukti kuat bahwa peningkatan jumlah adopsi robot berasosiasi secara langsung dengan peningkatan produktivitas.

PENDAHULUAN

Perkembangan teknologi otomasi dan robotika telah mendorong berbagai sektor industri, termasuk manufaktur, kesehatan, dan logistik, untuk mengadopsi robot sebagai bagian dari proses operasional mereka. Adopsi robot umumnya diyakini dapat meningkatkan efisiensi kerja, mengurangi kesalahan manusia, serta meningkatkan produktivitas secara keseluruhan. Namun demikian, hubungan antara jumlah robot yang diadopsi dan besarnya peningkatan produktivitas yang dihasilkan belum tentu bersifat linear atau signifikan, karena banyak faktor lain seperti pelatihan tenaga kerja, efisiensi biaya, dan karakteristik sektor industri yang turut memengaruhi hasil akhir.

Penelitian ini bertujuan untuk menjawab pertanyaan penelitian: apakah semakin banyak robot yang diadopsi oleh suatu organisasi akan meningkatkan produktivitasnya secara signifikan? Untuk menjawab pertanyaan tersebut, digunakan pendekatan kecerdasan buatan berupa algoritma Random Forest Classifier, yang dipilih karena kemampuannya menangani data tabular berukuran kecil,

ketahanannya terhadap overfitting melalui mekanisme bagging, serta kemampuannya menghasilkan skor feature importance yang dapat digunakan untuk mengidentifikasi variabel paling berpengaruh terhadap target klasifikasi.

Kontribusi penelitian ini adalah menyediakan bukti empiris berbasis data mengenai kekuatan hubungan antara adopsi robot dan produktivitas, sekaligus mendemonstrasikan tahapan lengkap analisis data menggunakan metode klasifikasi, mulai dari praproses data, ekstraksi fitur, pelatihan model, hingga evaluasi hasil menggunakan confusion matrix, kurva ROC, histogram, dan visualisasi sebaran data.

LANDASAN TEORI

2.1 Adopsi Robot dan Produktivitas

Adopsi robot merupakan bagian dari transformasi teknologi otomasi yang bertujuan menggantikan atau mendukung tugas-tugas yang sebelumnya dikerjakan oleh manusia dalam proses produksi maupun layanan. Secara teoretis, otomasi dapat meningkatkan produktivitas melalui peningkatan kecepatan proses, konsistensi kualitas, dan pengurangan kesalahan operasional. Namun, beberapa kajian menunjukkan bahwa hubungan antara adopsi robot dan produktivitas tidak selalu bersifat linear, karena dipengaruhi oleh faktor lain seperti kesiapan tenaga kerja, jenis industri, dan skala investasi. Acemoglu dan Restrepo mengemukakan bahwa dampak robot terhadap pasar tenaga kerja dan produktivitas bervariasi tergantung pada sektor dan tingkat substitusi tenaga kerja oleh mesin, sehingga peningkatan jumlah robot tidak serta-merta menjamin peningkatan produktivitas secara proporsional [2].

2.2 Klasifikasi dalam Kecerdasan Buatan

Klasifikasi merupakan salah satu tugas utama dalam pembelajaran mesin (*machine learning*) yang bertujuan memetakan sekumpulan variabel prediktor ke dalam kelas atau kategori tertentu. Dalam konteks penelitian ini, variabel target yang semula bersifat kontinu ditransformasikan menjadi variabel kelas biner agar dapat dianalisis menggunakan pendekatan klasifikasi. Pendekatan ini umum digunakan ketika peneliti ingin mengetahui apakah suatu observasi termasuk ke dalam kategori tinggi atau rendah berdasarkan ambang batas tertentu, seperti nilai median.

2.3 Random Forest Classifier

Random Forest Classifier merupakan metode pembelajaran ensemble yang diperkenalkan oleh Breiman, yang bekerja dengan cara membangun sejumlah besar decision tree secara acak melalui teknik bootstrap aggregating (bagging) dan menggabungkan hasil prediksi masing-masing pohon melalui mekanisme voting mayoritas. Setiap pohon dilatih menggunakan subset data dan subset fitur yang dipilih secara acak, sehingga metode ini relatif tahan terhadap overfitting dibandingkan decision tree tunggal, terutama pada dataset berukuran kecil. Selain menghasilkan prediksi kelas, Random Forest juga mampu menghitung nilai feature importance, yaitu ukuran kontribusi relatif setiap variabel prediktor terhadap akurasi klasifikasi, sehingga dapat digunakan untuk mengidentifikasi variabel yang paling berpengaruh terhadap variabel target.

2.4 Validasi Silang dan Evaluasi Model

Validasi silang (cross-validation) merupakan teknik evaluasi model yang bertujuan mengukur kemampuan generalisasi model terhadap data baru dengan cara membagi data menjadi beberapa lipatan (fold), melatih model pada sebagian lipatan, dan mengujinya pada lipatan yang tersisa secara bergantian. Teknik ini penting digunakan terutama pada dataset berukuran kecil untuk mengurangi bias akibat pembagian data latih dan data uji yang hanya dilakukan sekali. Evaluasi performa model klasifikasi umumnya dilakukan menggunakan confusion matrix serta kurva Receiver Operating Characteristic (ROC) beserta nilai Area Under Curve (AUC). Confusion matrix menggambarkan jumlah prediksi benar dan salah untuk setiap kelas, sedangkan kurva ROC dan AUC digunakan untuk menilai kemampuan model dalam membedakan antar kelas pada berbagai ambang batas probabilitas, di mana nilai AUC mendekati 0,5 mengindikasikan kemampuan diskriminasi model yang setara dengan tebakan acak.

METODE PENELITIAN

2.1 Data Penelitian

Data yang digunakan dalam penelitian ini berjumlah 27 observasi yang mencakup periode tahun 2015 hingga 2023 pada tiga sektor industri, yaitu Manufaktur, Kesehatan, dan Logistik. Variabel yang tersedia dalam dataset meliputi Year, Industry, Robots_Adopted, Productivity_Gain, Cost_Savings, Jobs_Displaced, dan Training_Hours. Variabel Robots_Adopted merupakan variabel bebas utama yang merepresentasikan jumlah robot yang diadopsi, sedangkan Productivity_Gain merupakan variabel target yang merepresentasikan persentase peningkatan produktivitas.

2.2 Penentuan Metode Kecerdasan Buatan

Metode yang dipilih dalam penelitian ini adalah Random Forest Classifier, yaitu metode pembelajaran ensemble yang membangun banyak decision tree secara acak (bootstrap aggregating) dan menggabungkan hasil prediksinya melalui mekanisme voting. Metode ini dipilih dengan pertimbangan sebagai berikut: (1) sesuai untuk data tabular dengan kombinasi fitur numerik dan kategorik; (2) relatif tahan terhadap overfitting meskipun jumlah data kecil; dan (3) mampu menghasilkan nilai feature importance yang berguna untuk mengidentifikasi variabel yang paling berkontribusi terhadap klasifikasi produktivitas.

2.3 Praproses Data

- Pemeriksaan missing value: tidak ditemukan nilai yang hilang pada seluruh variabel.
- Label encoding pada variabel kategorik Industry menjadi representasi numerik.
- Transformasi variabel target Productivity_Gain (kontinu) menjadi variabel kelas biner (0 = Low, 1 = High) berdasarkan nilai median sebesar 15,71.
- Standardisasi (z-score) pada seluruh fitur numerik menggunakan StandardScaler.

2.4 Ekstraksi Fitur

Fitur yang digunakan dalam proses klasifikasi terdiri atas Robots_Adopted sebagai variabel utama penelitian, serta Industry (terenkode), Cost_Savings, Jobs_Displaced, dan Training_Hours sebagai variabel pendukung yang berpotensi turut memengaruhi produktivitas.

2.5 Klasifikasi

Data dibagi menjadi data latih (70%) dan data uji (30%) menggunakan teknik stratified split agar proporsi kelas tetap seimbang. Model Random Forest dilatih dengan 200 pohon keputusan (n_estimators = 200) dan kedalaman maksimum 4 (max_depth = 4) untuk mencegah overfitting pada data yang berukuran kecil. Untuk memperoleh estimasi performa yang lebih stabil, dilakukan pula validasi silang (cross-validation) dengan skema stratified 5-fold

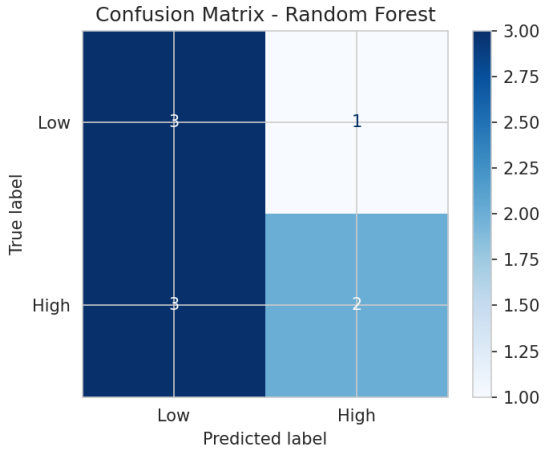
2.6 Evaluasi Model

Evaluasi hasil klasifikasi dilakukan menggunakan confusion matrix, kurva ROC beserta nilai Area Under Curve (AUC), histogram distribusi Productivity_Gain berdasarkan kelas, diagram sebar (scatter plot) antara Robots_Adopted dan Productivity_Gain, serta analisis feature importance untuk mengidentifikasi variabel yang paling berpengaruh terhadap hasil klasifikasi.

HASIL DAN PEMBAHASAN

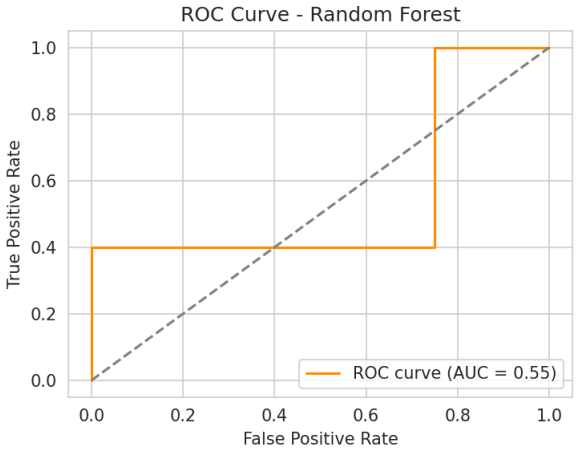
Performa Model Klasifikasi

Model Random Forest yang dilatih menghasilkan akurasi sebesar 55,6% pada data uji. Sementara itu, hasil validasi silang 5-fold menunjukkan rata-rata akurasi yang jauh lebih rendah, yaitu 26%, dengan standar deviasi 0,17. Nilai ini berada di bawah tingkat akurasi tebakan acak untuk klasifikasi dua kelas (50%), yang mengindikasikan bahwa model tidak mampu menemukan pola prediktif yang konsisten dari fitur-fitur yang tersedia terhadap kelas produktivitas.



Gambar 1. Confusion Matrix hasil klasifikasi Random Forest pada data uji

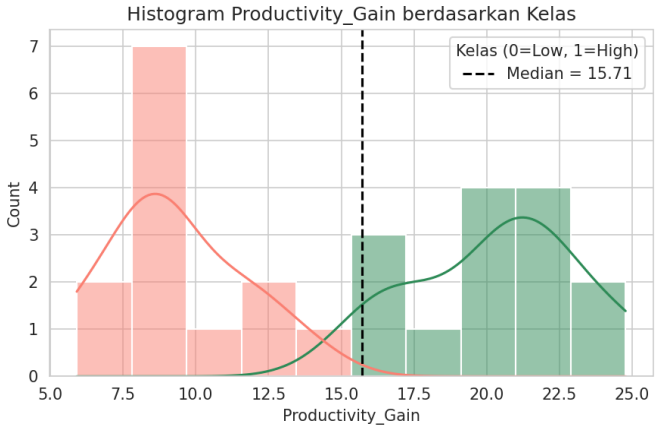
Confusion matrix pada Gambar 1 menunjukkan bahwa model masih melakukan cukup banyak kesalahan klasifikasi antara kelas Low dan High, yang memperkuat indikasi lemahnya pola prediktif dalam data.



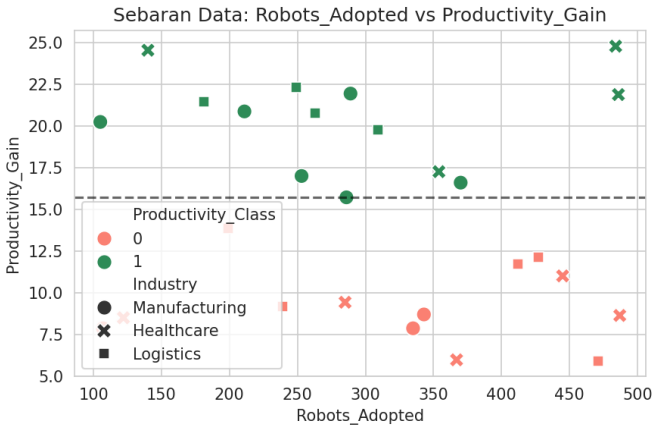
Gambar 2. Kurva ROC dan nilai AUC model Random Forest

Kurva ROC pada Gambar 2 menggambarkan kemampuan model dalam membedakan kelas High dan Low pada berbagai ambang batas probabilitas. Kurva yang mendekati garis diagonal (AUC mendekati 0,5) menunjukkan bahwa kemampuan diskriminatif model tidak jauh berbeda dari klasifikasi acak.

Distribusi dan Sebaran Data



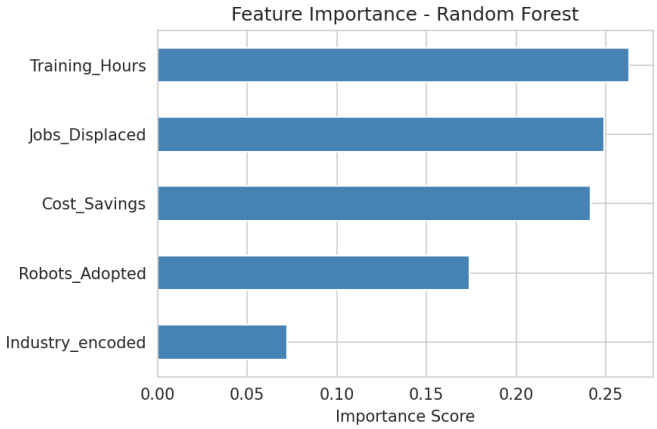
Histogram pada Gambar 3 memperlihatkan bahwa distribusi nilai Productivity_Gain pada kedua kelas relatif tumpang tindih (overlap) di sekitar nilai median, tanpa pemisahan yang jelas, yang menunjukkan bahwa variabel-variabel prediktor yang digunakan belum mampu membedakan kedua kelas secara tegas.



Gambar 4. Sebaran data Robots_Adopted terhadap Productivity_Gain berdasarkan kelas dan sektor industri

Diagram sebar pada Gambar 4 menunjukkan tidak adanya tren atau pola yang jelas antara jumlah robot yang diadopsi dan besarnya peningkatan produktivitas. Titik data dengan jumlah robot yang tinggi maupun rendah tersebar merata pada kedua sisi garis median, baik untuk kelas Low maupun High, pada ketiga sektor industri yang diteliti.

Analisis Feature Importance



Gambar 5. Skor feature importance dari model Random Forest

Berdasarkan Gambar 5, urutan tingkat kepentingan fitur dari yang tertinggi adalah Training_Hours (0,263), Jobs_Displaced (0,249), Cost_Savings (0,242), Robots_Adopted (0,174), dan Industry (0,072). Hasil ini menunjukkan bahwa Robots_Adopted, yang merupakan variabel utama dalam pertanyaan penelitian, justru bukan merupakan fitur dengan kontribusi tertinggi terhadap klasifikasi kelas produktivitas.

Fitur	Skor Feature Importance
Training_Hours	0,263

Jobs_Displaced	0,249
Cost_Savings	0,242
Robots_Adopted	0,174
Industry	0,072

Tabel 1. Ringkasan skor feature importance model Random Forest

Analisis Korelasi

Analisis korelasi Pearson antara Robots_Adopted dan Productivity_Gain menghasilkan nilai r = -0,173, yang menunjukkan hubungan linear yang sangat lemah dan bahkan berarah negatif. Nilai korelasi ini memperkuat hasil dari model klasifikasi maupun analisis feature importance, bahwa penambahan jumlah robot yang diadopsi tidak serta-merta diikuti oleh peningkatan produktivitas dalam dataset yang diteliti.

Pembahasan

Secara keseluruhan, seluruh indikator analisis, mulai dari akurasi klasifikasi, confusion matrix, kurva ROC, histogram distribusi, diagram sebar, feature importance, hingga korelasi Pearson, menunjukkan arah kesimpulan yang konsisten, yaitu tidak ditemukannya hubungan yang kuat antara jumlah robot yang diadopsi dan besarnya peningkatan produktivitas pada dataset yang digunakan. Temuan ini berbeda dengan asumsi umum bahwa otomasi selalu berbanding lurus dengan produktivitas. Kemungkinan penjelasan atas temuan ini antara lain: (1) peningkatan produktivitas kemungkinan lebih dipengaruhi oleh faktor lain yang tidak tercakup dalam dataset, seperti kualitas pelatihan, jenis robot, atau tingkat integrasi proses; dan (2) kemungkinan data yang digunakan bersifat simulatif sehingga tidak merepresentasikan hubungan sebab-akibat yang sesungguhnya terjadi di lapangan.

KESIMPULAN

Berdasarkan hasil analisis menggunakan Random Forest Classifier terhadap data adopsi robot dan produktivitas pada 27 observasi dari tiga sektor industri, dapat disimpulkan bahwa tidak ditemukan bukti yang kuat mengenai hubungan antara jumlah robot yang diadopsi (Robots_Adopted) dan peningkatan produktivitas (Productivity_Gain). Hal ini ditunjukkan oleh rendahnya akurasi model (55,6% pada data uji dan 26% pada validasi silang), lemahnya korelasi Pearson (r = -0,173), serta rendahnya kontribusi Robots_Adopted pada analisis feature importance dibandingkan variabel lain seperti Training_Hours, Jobs_Displaced, dan Cost_Savings.

Penelitian selanjutnya disarankan untuk menggunakan jumlah data yang lebih besar serta menambahkan variabel-variabel lain yang secara teoretis lebih relevan, seperti jenis dan kompleksitas robot, tingkat integrasi otomasi dalam proses produksi, kompetensi tenaga kerja, maupun karakteristik organisasi, guna memperoleh gambaran yang lebih komprehensif mengenai hubungan antara otomasi dan produktivitas.

Daftar Pustaka

[1] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

[2] Acemoglu, D., & Restrepo, P. (2020). Robots and jobs: Evidence from US labor markets. Journal of Political Economy, 128(6), 2188–2244.

[3] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[4] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874.

Contents lists available at MID Publisher International

Technology Sciences Insights Journal

Journal homepage: https://journal.midpublisher.com/index.php/tsij

Deteksi Keberadaan Manusia dalam Ruangan Berbasis Data Sensor Lingkungan Menggunakan Algoritma K-Nearest Neighbor (KNN)

Nur Shakinah, Adlian Jefiza
Politeknik Negeri Batam, Batam, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima: Juni 2025
Revisi: Juni 2025
Diterima: Juli 2025
Dipublikasi: Juli 2025

Kata Kunci:
K-Nearest Neighbor1; Deteksi Okupasi; Sensor Lingkungan, Klasifikasi, Machine Learning.

*Penulis Korespondensi:
nurshakinahh05@gmai.com

ABSTRAK

Penelitian ini bertujuan untuk menganalisis kemampuan algoritma *K-Nearest Neighbor (*KNN) dalam mendeteksi keberadaan manusia (okupansi) di dalam suatu ruangan berdasarkan data sensor lingkungan melalui pendekatan klasifikasi biner. Data dikumpulkan dari sensor ruangan yang merekam variabel *Temperature, Humidity, Light, CO2, dan Humidity Ratio,* kemudian diolah menggunakan teknik standardisasi fitur (*StandardScaler*) dan validasi silang (*cross-validation*) guna memperoleh gambaran mengenai performa model klasifikasi okupansi ruangan. Hasil penelitian menunjukkan bahwa model KNN dengan nilai K=1 memberikan akurasi validasi silang tertinggi sebesar 99,06% dan akurasi pada data uji yang sangat tinggi, yang mengindikasikan bahwa variabel *Light* dan *CO2* memiliki korelasi paling kuat terhadap status okupansi. Temuan ini diharapkan dapat memberikan kontribusi terhadap bidang *smart building* dan efisiensi energi serta menjadi rujukan untuk penelitian selanjutnya.

ABSTRACT

This study aims to analyze the capability of the K-Nearest Neighbor (KNN) algorithm in detecting human presence (occupancy) in a room based on environmental sensor data through a binary classification approach. Data were collected from room sensors recording Temperature, Humidity, Light, CO2, and Humidity Ratio, and were processed using feature standardization (StandardScaler) and cross-validation to evaluate the performance of the room occupancy classification model. The findings indicate that the KNN model with K=1 achieved the highest cross-validation accuracy of 99.06% and a very high accuracy on the test data, suggesting that Light and CO2 have the strongest correlation with occupancy status. These results are expected to contribute to the field of smart buildings and energy efficiency and serve as a reference for future research.

PENDAHULUAN

Perkembangan dalam bidang bangunan pintar (*smart building*) dan *Internet of Things* (IoT) menunjukkan bahwa efisiensi energi berbasis deteksi okupansi ruangan semakin menjadi perhatian dalam beberapa tahun terakhir. Kondisi ini dipengaruhi oleh meningkatnya kebutuhan akan sistem otomatisasi pencahayaan, pengaturan suhu, dan ventilasi yang responsif terhadap keberadaan manusia secara *real-time*, sehingga diperlukan kajian lebih mendalam untuk memahami pola hubungan antara data sensor lingkungan dan status okupansi ruangan [1], [2]. Meskipun berbagai penelitian sebelumnya telah membahas deteksi okupansi menggunakan sensor tunggal seperti *Passive Infrared* (PIR) atau kamera, masih terdapat keterbatasan berupa akurasi yang rendah pada kondisi tertentu serta biaya implementasi kamera yang relatif tinggi dan berpotensi melanggar privasi pengguna [3], [4]. Keterbatasan tersebut memunculkan kebutuhan untuk meneliti lebih lanjut mengenai pemanfaatan kombinasi sensor lingkungan non-

intrusif, seperti suhu, kelembapan, intensitas cahaya, dan konsentrasi CO_2, untuk memprediksi okupansi ruangan secara lebih akurat [2], [5].
Berdasarkan hal tersebut, penelitian ini bertujuan untuk menganalisis performa *algoritma K-Nearest Neighbor* (KNN) dalam mengklasifikasikan status okupansi ruangan (terisi/tidak terisi) menggunakan pendekatan *supervised learning* berbasis data sensor lingkungan. Algoritma KNN dipilih karena memiliki kompleksitas implementasi yang sederhana, tidak memerlukan proses pelatihan model yang kompleks, serta mampu memberikan performa klasifikasi yang baik pada dataset berdimensi rendah hingga menengah [6]. Hasil penelitian diharapkan dapat memberikan kontribusi terhadap pengembangan ilmu pada bidang *machine learning* terapan serta memberikan manfaat praktis bagi pengembang sistem manajemen energi bangunan.

LANDASAN TEORI

Kajian teori ini membahas konsep-konsep utama dan hasil penelitian terdahulu yang menjadi dasar dalam menganalisis deteksi okupansi ruangan berbasis sensor. Secara teoretis, algoritma *K-Nearest Neighbor* (KNN) menjelaskan bahwa klasifikasi suatu data baru dilakukan berdasarkan mayoritas kelas dari K data tetangga terdekat menggunakan perhitungan jarak, yang umumnya adalah *Euclidean Distance*, sehingga algoritma ini mampu mengenali pola hubungan non-linear antara variabel sensor dan status okupansi ruangan [6].
Selain itu, konsep standardisasi fitur (*feature scaling*) memberikan landasan mengenai pentingnya menyamakan skala antarvariabel sebelum proses penghitungan jarak dilakukan. Hal ini menjadi penting karena variabel seperti *Temperature (°C), Humidity (%RH), Light (lux), dan* CO_2 *(ppm)* memiliki satuan dan rentang nilai yang berbeda sehingga dapat memengaruhi hasil klasifikasi apabila tidak dinormalisasi terlebih dahulu [6].
Penelitian-penelitian sebelumnya pada bidang deteksi okupansi menunjukkan bahwa kombinasi sensor lingkungan memiliki performa yang lebih baik dibandingkan penggunaan sensor tunggal. Variabel CO_2, cahaya, dan temperatur merupakan indikator yang paling berpengaruh terhadap keberadaan manusia di dalam ruangan, meskipun kemampuan generalisasi model masih dipengaruhi oleh karakteristik masing-masing ruangan [2], [5]. Selain itu, penelitian lain menunjukkan bahwa penggunaan algoritma KNN pada sistem deteksi okupansi berbasis IoT mampu memberikan akurasi yang tinggi dengan kebutuhan komputasi yang relatif rendah sehingga sesuai diterapkan pada perangkat dengan sumber daya terbatas (*edge device*) [3].
Beberapa studi juga menegaskan bahwa pemilihan nilai K yang optimal melalui proses cross-validation berperan penting dalam menyeimbangkan bias dan varians model sehingga mampu meningkatkan performa klasifikasi secara keseluruhan [6]. Berdasarkan keseluruhan teori dan penelitian terdahulu tersebut, penelitian ini mengadopsi kerangka berpikir bahwa kombinasi lima variabel sensor lingkungan dapat digunakan sebagai fitur prediktor yang memadai untuk mengklasifikasikan status okupansi ruangan. Kerangka tersebut selanjutnya digunakan sebagai dasar dalam penyusunan metode penelitian dan analisis hasil pengujian.

METODE PENELITIAN

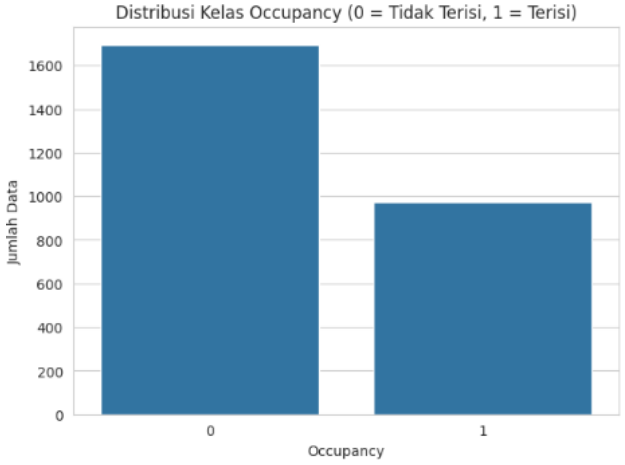
Metodologi penelitian ini dirancang untuk menjelaskan pendekatan, teknik pengumpulan data, serta prosedur analisis yang digunakan dalam mengkaji deteksi okupansi ruangan. Penelitian ini menggunakan pendekatan kuantitatif dengan metode klasifikasi supervised learning karena dianggap paling sesuai untuk menjawab rumusan masalah mengenai prediksi status okupansi berdasarkan data sensor. Populasi dalam penelitian ini adalah seluruh rekaman data sensor ruangan pada rentang waktu pengamatan, sedangkan sampelnya diperoleh dari berkas datatest.csv yang berjumlah 2.665 baris data dengan 8 kolom (id, date, Temperature, Humidity, Light, CO2, HumidityRatio, dan Occupancy). Data dikumpulkan melalui instrumen sensor lingkungan (suhu, kelembapan, intensitas cahaya, konsentrasi CO2, dan rasio kelembapan) yang

telah melalui proses pemeriksaan nilai kosong (missing value) dan data duplikat untuk memastikan kualitas data, di mana hasil pemeriksaan menunjukkan tidak ditemukan nilai kosong maupun data duplikat pada dataset. Kolom identitas (id) dan waktu (date) yang tidak relevan secara prediktif kemudian dihapus, sehingga tersisa lima variabel fitur (Temperature, Humidity, Light, CO2, HumidityRatio) dan satu variabel target (Occupancy). Data selanjutnya dibagi menjadi data latih (80%, 2.132 baris) dan data uji (20%, 533 baris) menggunakan teknik stratified random sampling agar proporsi kelas pada data latih dan data uji tetap seimbang. Sebelum pemodelan, seluruh fitur distandardisasi menggunakan *StandardScaler* agar memiliki skala yang setara. Analisis dilakukan menggunakan algoritma K-Nearest Neighbor (KNN) dengan pencarian nilai K optimal pada rentang 1 hingga 20 melalui 5-fold cross-validation berbasis metrik akurasi, guna memperoleh temuan yang objektif dan sesuai dengan tujuan penelitian. Evaluasi akhir model dilakukan menggunakan confusion matrix dan metrik akurasi pada data uji. Seluruh prosedur penelitian dilaksanakan secara sistematis menggunakan bahasa pemrograman Python beserta pustaka scikit-learn, pandas, dan seaborn untuk memastikan bahwa hasil yang diperoleh akurat, konsisten, dan dapat dipertanggungjawabkan secara ilmiah.

HASIL DAN PEMBAHASAN

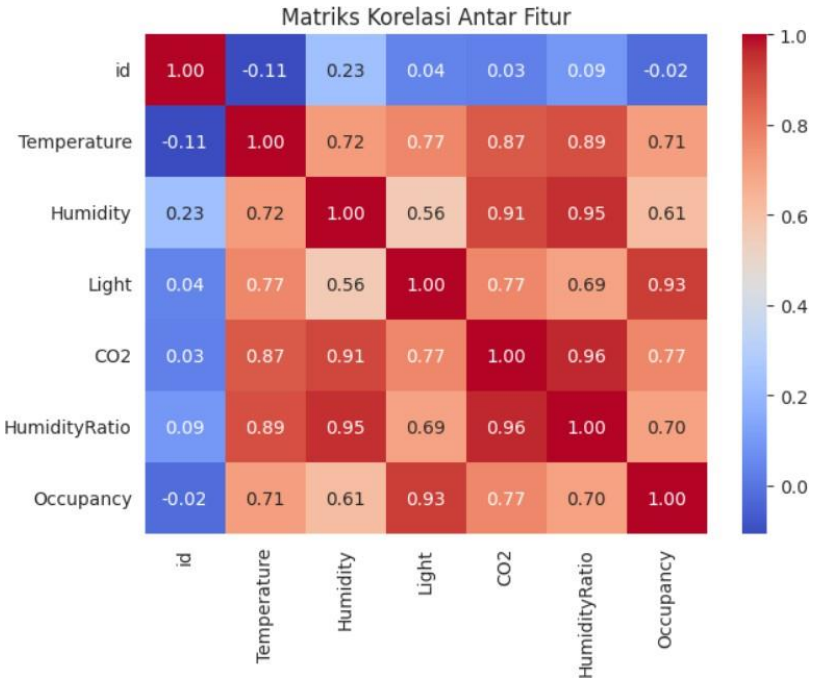
Bab ini menyajikan hasil penelitian yang diperoleh dari proses eksplorasi data, *preprocessing*, pemilihan parameter model, hingga evaluasi performa algoritma K-Nearest Neighbor (KNN) dalam mengklasifikasikan status okupansi ruangan berdasarkan data sensor lingkungan.

Dataset yang digunakan terdiri atas 2.665 data dengan enam variabel prediktor, yaitu *Temperature*, *Humidity*, *Light*, CO_2, *HumidityRatio*, dan *id*, serta satu variabel target yaitu Occupancy. Berdasarkan hasil eksplorasi data diketahui bahwa dataset tidak memiliki nilai kosong (*missing value*) maupun data duplikat sehingga seluruh data dapat digunakan pada tahap analisis berikutnya. Selain itu, distribusi kelas menunjukkan adanya ketidakseimbangan (*class imbalance*), yaitu sebanyak 1.693 data termasuk kelas Occupancy = 0 (ruangan tidak terisi) dan 972 data termasuk kelas Occupancy = 1 (ruangan terisi).



Gambar 1. Distribusi Kelas Occupancy

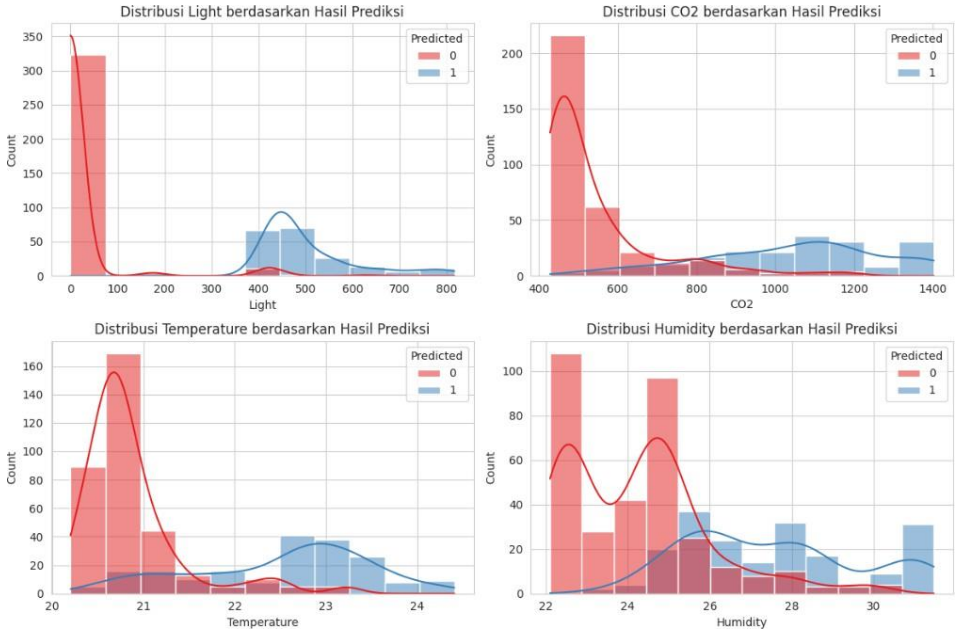
Distribusi kelas tersebut menunjukkan bahwa jumlah data pada kelas ruangan tidak terisi lebih banyak dibandingkan kelas ruangan terisi. Kondisi ini perlu diperhatikan karena dapat memengaruhi proses pelatihan dan evaluasi model klasifikasi apabila tidak dianalisis dengan baik. Selanjutnya dilakukan analisis hubungan antarvariabel menggunakan matriks korelasi untuk mengetahui tingkat pengaruh masing-masing fitur terhadap status okupansi ruangan.



Gambar 2. Heatmap korelasi antar variabel

Berdasarkan Gambar 2, variabel Light memiliki korelasi tertinggi terhadap Occupancy dengan nilai sebesar 0,93, diikuti oleh CO_2 (0,77), Temperature (0,71), HumidityRatio (0,70), dan Humidity (0,61). Sementara itu, variabel id memiliki korelasi yang sangat rendah sehingga tidak memberikan kontribusi yang signifikan terhadap proses klasifikasi. Hasil ini menunjukkan bahwa perubahan intensitas cahaya dan konsentrasi CO_2 merupakan indikator utama keberadaan manusia di dalam ruangan. Temuan tersebut sejalan dengan penelitian sebelumnya yang menyatakan bahwa kombinasi sensor lingkungan mampu meningkatkan akurasi deteksi okupansi dibandingkan penggunaan sensor tunggal.

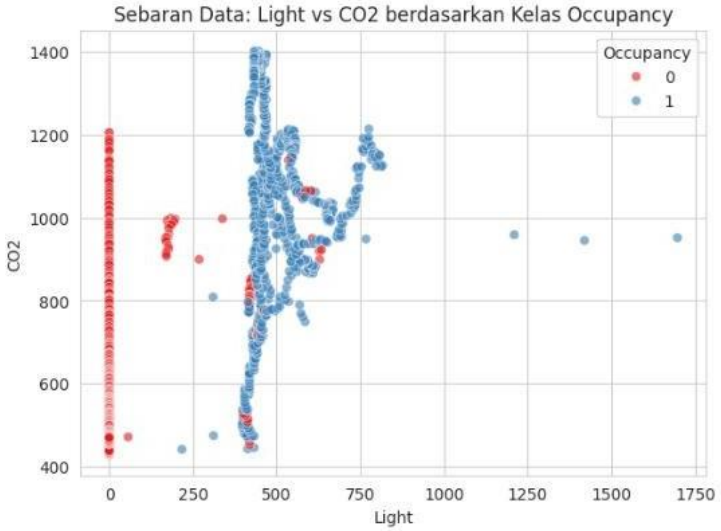
Untuk memperoleh gambaran distribusi setiap fitur terhadap hasil klasifikasi, dilakukan visualisasi menggunakan histogram. Histogram digunakan untuk melihat penyebaran nilai masing-masing variabel pada setiap kelas hasil prediksi sehingga dapat diketahui fitur yang memiliki kemampuan paling baik dalam membedakan kondisi ruangan terisi dan tidak terisi.



Gambar 3. Histogram Distribusi Fitur Berdasarkan Hasil Prediksi

Berdasarkan Gambar 3, variabel Light menunjukkan pemisahan distribusi yang paling jelas antara kelas Occupancy = 0 dan Occupancy = 1. Data pada kelas Occupancy = 0 didominasi oleh nilai intensitas cahaya yang rendah, sedangkan kelas Occupancy = 1 cenderung memiliki nilai Light yang lebih tinggi. Pola serupa juga terlihat pada variabel CO_2, di mana data pada kelas Occupancy = 1 memiliki konsentrasi CO_2 yang lebih tinggi dibandingkan kelas Occupancy = 0. Hal ini menunjukkan bahwa keberadaan manusia di dalam ruangan berkaitan dengan meningkatnya intensitas pencahayaan dan konsentrasi CO_2 akibat aktivitas manusia.

Sementara itu, variabel Temperature dan Humidity masih menunjukkan adanya tumpang tindih (*overlap*) antara kedua kelas. Meskipun demikian, kedua variabel tersebut tetap memberikan informasi tambahan yang membantu proses klasifikasi ketika dikombinasikan dengan variabel lain. Hasil ini memperkuat temuan pada analisis korelasi bahwa Light dan CO_2 merupakan fitur yang paling berpengaruh dalam mendeteksi status okupansi ruangan.
Selain histogram, dilakukan pula visualisasi sebaran data menggunakan scatter plot antara variabel Light dan CO_2 untuk melihat tingkat pemisahan kedua kelas secara visual.



Gambar 4. Sebaran Data Light dan CO2 Berdasarkan Kelas Occupancy

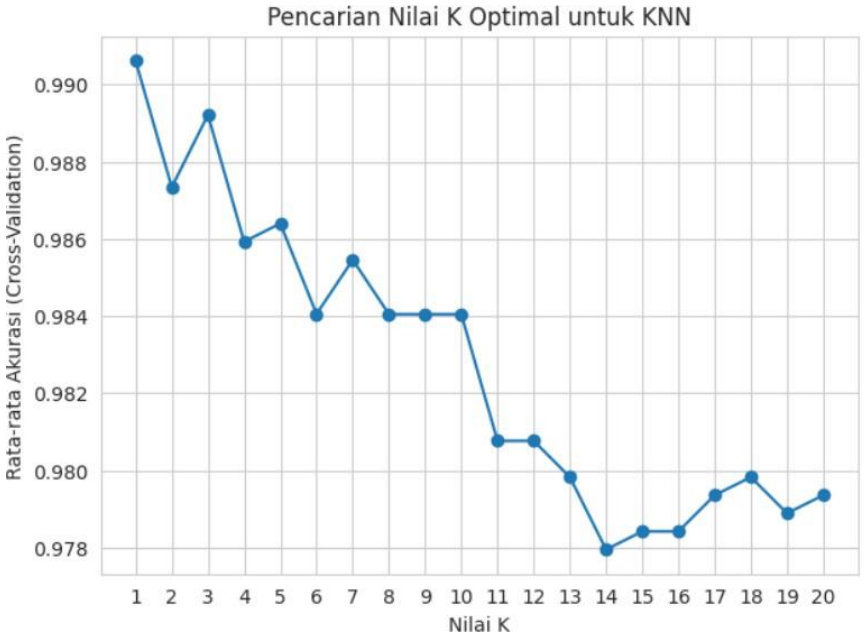
Berdasarkan Gambar 4, terlihat bahwa data dengan Occupancy = 0 cenderung terkonsentrasi pada nilai Light yang rendah dengan rentang CO_2 yang relatif lebih kecil. Sebaliknya, data dengan Occupancy = 1 sebagian besar berada pada nilai Light dan CO_2 yang lebih tinggi. Walaupun masih terdapat beberapa titik yang saling berdekatan, secara umum kedua kelas menunjukkan pola pemisahan yang cukup jelas. Kondisi ini menunjukkan bahwa kombinasi variabel Light dan CO_2 mampu membedakan status okupansi dengan baik sehingga mendukung tingginya performa algoritma KNN dalam proses klasifikasi.
Sebelum proses klasifikasi dilakukan, seluruh variabel numerik melalui tahap *preprocessing* menggunakan metode StandardScaler. Standardisasi bertujuan untuk menyamakan skala setiap variabel sehingga proses perhitungan jarak Euclidean pada algoritma KNN tidak dipengaruhi oleh perbedaan satuan maupun rentang nilai antarfitur.

```
Contoh data sebelum standardisasi:
[3.390000e+02 2.254000e+01 2.516000e+01 4.274000e+02 8.536000e+02
 4.249921e-03]
Contoh data setelah standardisasi:
[-1.45789905  1.09383554 -0.07246815  0.93928435  0.46989738  0.37406335]
```

Gambar 5. Hasil standardisasi data menggunakan StandardScaler

Berdasarkan hasil standardisasi pada Gambar 5, seluruh fitur telah berada pada skala yang seragam sehingga proses klasifikasi dapat dilakukan secara lebih optimal. Tahapan ini menjadi penting karena algoritma KNN merupakan metode berbasis jarak (*distance-based algorithm*), sehingga performanya sangat dipengaruhi oleh skala data.

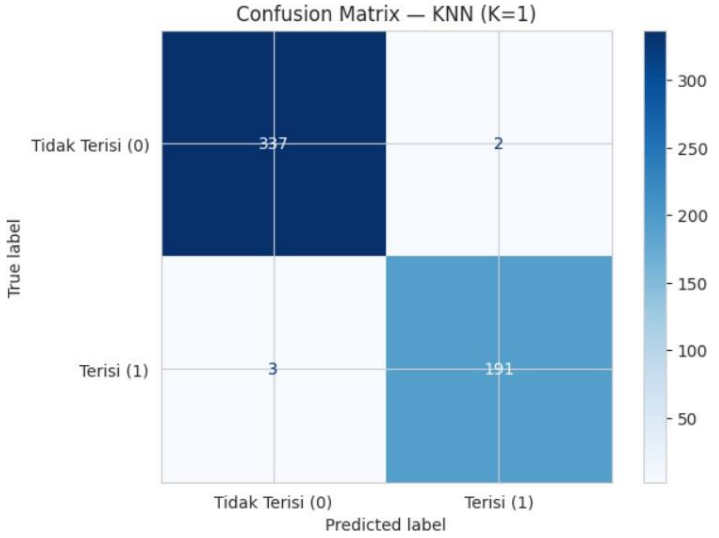
Tahap berikutnya adalah menentukan nilai K terbaik menggunakan metode 5-fold Cross Validation. Pengujian dilakukan pada beberapa nilai K untuk memperoleh parameter yang menghasilkan akurasi tertinggi.



Gambar 6. Grafik akurasi terhadap variasi nilai K

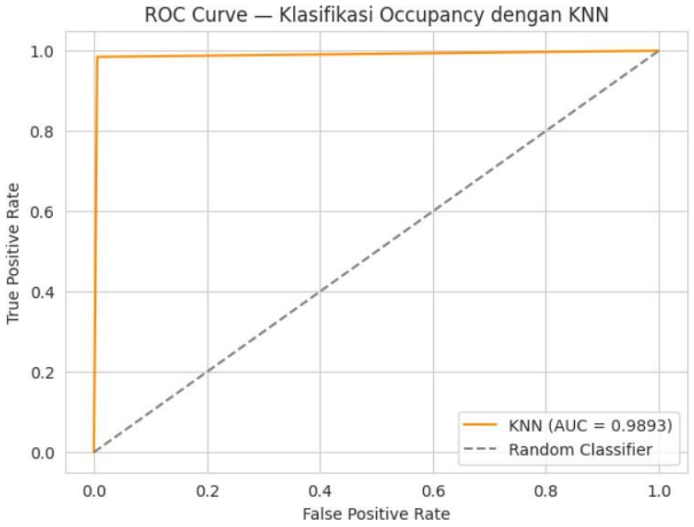
Hasil pengujian menunjukkan bahwa K = 1 memberikan rata-rata akurasi tertinggi sebesar 99,06%, sehingga nilai tersebut dipilih sebagai parameter akhir dalam proses klasifikasi. Tingginya akurasi menunjukkan bahwa karakteristik data memiliki pemisahan kelas yang baik sehingga satu tetangga terdekat sudah mampu memberikan keputusan klasifikasi yang optimal.

Setelah parameter terbaik diperoleh, dilakukan proses klasifikasi menggunakan algoritma KNN dan dievaluasi menggunakan Confusion Matrix, ROC Curve, serta metrik evaluasi berupa Accuracy, Precision, Recall, dan F1-score.



Gambar 7. Confusion Matrix hasil klasifikasi KNN

Berdasarkan Confusion Matrix pada Gambar 7, sebagian besar data berhasil diklasifikasikan dengan benar. Jumlah kesalahan klasifikasi (*False Positive* dan *False Negative*) relatif kecil dibandingkan jumlah prediksi yang benar (*True Positive* dan *True Negative*), sehingga menunjukkan bahwa model memiliki kemampuan yang baik dalam membedakan kondisi ruangan terisi dan tidak terisi.



Gambar 8. Kurva ROC algoritma KNN

Kurva ROC pada Gambar 8 menunjukkan kemampuan model dalam membedakan kedua kelas secara efektif. Nilai Area Under Curve (AUC) yang mendekati 1 menunjukkan bahwa model memiliki kemampuan diskriminasi yang sangat baik dalam mengidentifikasi status okupansi ruangan.

	Metrik	Nilai
0	Akurasi	0.9906
1	Presisi	0.9896
2	Recall	0.9845
3	F1-Score	0.9871
4	AUC	0.9893

Gambar 9. Hasil Akurasi, Presisi, Recall, dan F-1 score

Hasil ini didukung oleh nilai Accuracy, Precision, Recall, dan F1-score yang tinggi, sehingga menunjukkan bahwa model tidak hanya memiliki tingkat akurasi yang baik, tetapi juga mampu memberikan keseimbangan antara kemampuan mendeteksi kelas positif dan meminimalkan kesalahan klasifikasi.

Secara keseluruhan, hasil penelitian menunjukkan bahwa algoritma K-Nearest Neighbor (KNN) mampu mengklasifikasikan status okupansi ruangan dengan performa yang sangat baik. Tingginya akurasi model dipengaruhi oleh penggunaan fitur-fitur yang memiliki hubungan kuat terhadap status okupansi, khususnya variabel Light dan CO_2, serta didukung oleh proses *preprocessing* dan pemilihan nilai K yang optimal melalui 5-fold Cross Validation. Temuan ini sejalan dengan penelitian terdahulu yang menyatakan bahwa kombinasi sensor lingkungan non-intrusif dapat dimanfaatkan secara efektif untuk mendeteksi keberadaan manusia di dalam ruangan. Meskipun demikian, penelitian ini masih memiliki keterbatasan berupa distribusi kelas yang tidak seimbang dan pengujian yang hanya dilakukan pada satu dataset. Oleh karena itu, penelitian selanjutnya dapat mengkaji penerapan teknik penanganan *class imbalance* serta melakukan pengujian menggunakan dataset dari lingkungan bangunan yang berbeda agar kemampuan generalisasi model dapat ditingkatkan.

KESIMPULAN

Kesimpulan dari penelitian ini menunjukkan bahwa algoritma K-Nearest Neighbor mampu mengklasifikasikan status okupansi ruangan dengan sangat baik berdasarkan data sensor lingkungan, memberikan gambaran yang jelas mengenai hubungan antara variabel Light, CO_2, Temperature, HumidityRatio, dan Humidity terhadap keberadaan manusia dalam ruangan. Berdasarkan hasil analisis, dapat disimpulkan bahwa nilai K=1 merupakan parameter optimal untuk model KNN pada dataset ini dengan akurasi cross-validation sebesar 99,06%, yang menguatkan bahwa fitur pencahayaan dan konsentrasi CO_2 menjadi indikator dominan dalam deteksi okupansi. Penelitian ini juga mengungkap bahwa dataset yang digunakan bersih dari missing value dan duplikat namun memiliki ketidakseimbangan kelas yang perlu diperhatikan pada penelitian lanjutan, sehingga memberikan kontribusi terhadap pengembangan pengetahuan dalam bidang machine learning terapan untuk sistem bangunan pintar. Selain itu, hasil penelitian ini diharapkan dapat menjadi dasar bagi penelitian selanjutnya, khususnya dalam mengkaji penerapan teknik penanganan class imbalance serta pengujian model pada dataset okupansi dari ruangan dan kondisi lingkungan yang berbeda. Dengan demikian, studi ini tidak hanya memberikan pemahaman baru mengenai efektivitas KNN untuk deteksi okupansi, tetapi juga membuka peluang untuk pengembangan kajian lebih lanjut yang dapat memperkaya literatur terkait sistem deteksi okupansi berbasis sensor non-intrusif.

Daftar Pustaka

[1] A. R. Mena, H. G. Ceballos, and J. Alvarado-Uribe, "Measuring Indoor Occupancy through Environmental Sensors: A Systematic Review on Sensor Deployment," *Sensors*, vol. 22, no. 10, pp. 3770, 2022.

[2] A. Tsanousa, C. Moschou, E. Bektsis, S. Vrochidis, and I. Kompatsiaris, "Fusion of Environmental Sensors for Occupancy Detection in a Real Construction Site," *Sensors*, vol. 23, no. 23, pp. 9596, 2023.

[3] M. Emad-Ud-Din and Y. Wang, "Promoting Occupancy Detection Accuracy Using On-Device Lifelong Learning," *IEEE Sensors Journal*, vol. 23, no. 9, pp. 9595–9608, 2023.

[4] A. N. Sayed, Y. Himeur, and F. Bensaali, "Deep and Transfer Learning for Building Occupancy Detection: A Review and Comparative Analysis," *Engineering Applications of Artificial Intelligence*, vol. 115, 2022.

[5] C. Wang, J. Jiang, T. Roth, C. Nguyen, Y. Liu, and H. Lee, "Integrated Sensor Data Processing for Occupancy Detection in Residential Buildings," *Energy and Buildings*, vol. 237, 2021.

[6] M. Emad-Ud-Din and Y. Wang, "Promoting Occupancy Detection Accuracy Using On-Device Lifelong Learning," IEEE Sensors Journal, 2023.