

Predictive Maintenance Berbasis Machine Learning dalam Smart Manufacturing

Haris Rafi^{1 (*)}

¹ Universitas Bakrie, Jakarta

Abstract

The concept of predictive maintenance represents a significant change in traditional maintenance methods. The use of machine learning in manufacturing machine maintenance has the potential to offer unprecedented opportunities for predicting problems by uncovering hidden patterns in vast data sets. This study aims to examine four machine learning models in classifying maintenance needs in a smart manufacturing environment. Machine learning models such as Logistic Regression, Random Forest, XGBoost, and Multi-layer Perceptron (MLP) are trained with 5-fold cross-validation. The dataset used is a public dataset from the kaggle website, which consists of 10000 rows and 13 features with the maintenance_required feature as the target feature. The model training results are evaluated using various metrics, such as accuracy, precision, recall, f1-score, and ROC-AUC. The test results show that Random Forest provides the best performance with an accuracy of 98.37%, precision of 99.97%, recall of 91.72%, f1-score of 95.67%, and ROC-AUC of 95.95%. The tree-based ensemble method Random Forest is able to capture patterns in the data better than linear and neural models. This indicates that Random Forest is a reliable model for detecting machine maintenance requirements. Further research can consider increasing dataset capacity, integration with deep learning techniques, examining the perspective of multivariate time-series structures

Kata Kunci: Classification, Machine Learning, Smart Manufacturing, Predictive Maintenance.

Informasi Artikel:

Dikirim : 30 Oktober 2025
Direvisi : -
Diterima : 22 Januari 2026
Diterbitkan : 28 Januari 2026

Juli – Desember 2025, Vol 6 (2) : hlm 101 – 111
Institut Teknologi dan Bisnis Ahmad Dahlan

(*) Korespondensi: hariss.rafi@gmail.com (Haris Rafi)

PENDAHULUAN

Sektor manufaktur telah mengalami perubahan substansial sebagai hasil dari munculnya revolusi industri keempat, khususnya di bidang analisis data besar, kecerdasan buatan (AI), dan teknologi Internet of Things (IoT). Hal ini terkait erat dengan integrasi sistem fisik dan digital dalam lingkungan produksi. Keterkaitan ini memungkinkan pengumpulan data dalam jumlah besar dari berbagai peralatan yang terletak di berbagai area produksi (Carvalho et al., 2019). Berbagai mesin dan peralatan produksi dapat dipantau secara real-time menggunakan sensor pintar dan sistem pemantauan berbasis data. Oleh karena itu, gagasan “smart manufacturing” menjadi lebih penting karena memanfaatkan data yang dihasilkan oleh sistem yang saling terhubung untuk memfasilitasi proses produksi yang lebih efektif, fleksibel, dan berkelanjutan. Istilah “smart manufacturing” mengacu pada penggunaan teknologi digital dan konsep industri 4.0 dalam proses manufaktur untuk meningkatkan efisiensi (Sufian, Abdullah and Miller, 2025).

Pemeliharaan peralatan adalah salah satu komponen terpenting dari smart manufacturing. Dalam beberapa tahun terakhir, mayoritas industri cenderung menerapkan tindakan pemeliharaan preventif atau reaktif. Praktik memperbaiki komponen atau mesin hanya setelah mengalami malfungsi atau digunakan hingga rusak dikenal sebagai pemeliharaan reaktif. Sementara itu, untuk mencegah kerusakan tak terduga dan waktu henti (downtime) serta biaya yang diakibatkannya, pemeliharaan preventif, juga dikenal sebagai pemeliharaan terencana, mencakup pelaksanaan tugas pemeliharaan saat peralatan beroperasi normal (Carvalho et al., 2019; Phuyal, Bista and Bista, 2020; Prometheus Group, 2022).

Predictive maintenance (PdM) muncul sebagai solusi untuk masalah ini, dengan memanfaatkan algoritma machine learning untuk memprediksi kegagalan sebelum terjadi (Kalusivalingam et al., 2020). PdM mengantisipasi kebutuhan pemeliharaan untuk menghemat biaya akibat downtime yang tidak terduga. Dengan menghubungkan ke perangkat dan memantau data yang dihasilkan perangkat, pola yang menyebabkan potensi masalah atau kegagalan dapat segera diidentifikasi (Hrnjica and Softic, 2020). Machine learning (ML), dengan kapasitasnya untuk mengungkap pola tersembunyi dalam kumpulan data besar, menghadirkan kemungkinan yang sebelumnya tidak pernah terdengar untuk prediksi kegagalan peralatan dan remaining useful life (RUL). Pada saat yang sama, teknologi IoT memungkinkan pemantauan dan pengumpulan data secara terus-menerus dari peralatan produksi, menciptakan kumpulan data terperinci dan komprehensif yang menjadi dasar analisis prediktif (Kalusivalingam et al., 2020). Kombinasi ML dan IoT menawarkan solusi pemeliharaan yang real-time, terukur, dan adaptif yang disesuaikan dengan persyaratan pengaturan produksi kontemporer.

Isu penerapan PdM dalam mendukung smart manufacturing bukanlah hal baru. Pada (Hossan and Sultana, 2025), integrasi smart manufacturing dengan berbagai sistem seperti machine learning, deep learning, dan data analytics dilakukan. Pada penelitian ini, model LSTM mendapatkan tingkat akurasi tertinggi sebesar 88%.

Selain itu, model ML seperti Random Forest, Support Vector Machines, Gradient Boosting, dan Recurrent Neural Networks (RNN) diuji pada (Kalusivalingam et al., 2020). Dari penelitian ini RNN memberikan hasil yang memuaskan, yaitu 98% akurasi. Model XGBoost juga ternyata dapat memberikan hasil yang memuaskan untuk klasifikasi kondisi mesin produksi dengan akurasi sebesar 99.07% (Nazara, 2022). Penerapan ML untuk PdM mesin-mesin industri juga dilakukan oleh (Gaurkar, Kotalwar and Gabale, 2021). Penelitian ini juga mendapatkan model LSTM sebagai pengklasifikasi yang baik.

Studi ini menggunakan kumpulan data Smart Manufacturing IoT-Cloud Monitoring untuk menilai dan membandingkan beberapa teknik ML, seperti Logistic Regression, Random Forest, XGBoost, dan Multi-Layer Perceptron (MLP), untuk mengklasifikasikan persyaratan pemeliharaan mesin. Untuk mengatasi masalah ketidakseimbangan data, The Synthetic Minority Oversampling Technique (SMOTE) digunakan dalam penelitian ini untuk menyeimbangkan kumpulan data dan meningkatkan generalisasi model. Hasil dari penelitian ini diharapkan dapat membantu menciptakan sistem manajemen pemeliharaan yang berbasis data, mudah dipahami, dan efektif untuk sektor industri yang cerdas.

METODE

Penelitian ini bertujuan untuk mengembangkan dan membandingkan berbagai model ML untuk memprediksi apakah mesin membutuhkan perawatan atau tidak. Studi ini mengambil pendekatan kuantitatif karena meneliti hubungan antara karakteristik sensor mesin dan kebutuhan pemeliharaan menggunakan data numerik dan metode pembelajaran statistik. Untuk mengidentifikasi model PdM yang berkinerja terbaik, penelitian ini menggunakan metodologi eksperimental-komparatif di mana beberapa model ML dilatih, dioptimalkan, dan dievaluasi. Berikut tahap utama dari proses metodologi seperti yang digambarkan pada Gambar 1.

1. Pengumpulan Data

Data yang digunakan dalam studi ini, Smart Manufacturing IoT-Cloud Monitoring Dataset yang didapatkan dari situs web Kaggle (Ziya, 2025). Data ini berisi beberapa pembacaan sensor IoT seperti temperature, humidity, vibration, pressure, and energy consumption, di samping status operasional mesin dan label pemeliharaan. Apakah suatu mesin memerlukan pemeliharaan (1) atau tidak (0) ditunjukkan oleh variabel target, *maintenance_required*. Selain itu, *Exploratory Data Analysis* (EDA) dilakukan untuk memahami distribusi fitur, kualitas data, dan memeriksa korelasi antar fitur.

2. Prapemrosesan Data

Berikut adalah alur prapemrosesan:

- a. Imputasi Nilai yang Hilang

Rata-rata digunakan untuk mengganti nilai numerik yang hilang, sedangkan modus digunakan untuk mengganti nilai kategoris yang hilang.

- b. *Feature Encoding*

One-Hot Encoding (OHE) digunakan untuk mentransformasi fitur kategoris.

c. *Feature Scaling*

Teknik *feature scaling* digunakan untuk menormalisasikan distribusi fitur dan menstandarisasi fitur numerik.

3. Pembagian Data

Untuk mempertahankan distribusi kelas variabel target, pemisahan acak digunakan untuk memisahkan data menjadi 80% data latih dan 20% data uji. Subset uji hanya digunakan untuk evaluasi model, sedangkan subset pelatihan digunakan untuk penyetelan *hyperparameter*, *resampling* SMOTE, dan pelatihan model.

4. Pelatihan Model dan Optimasi *Hyperparameter*

Ada empat algoritma ML yang digunakan:

a. Regresi Logistik (RL)

Dalam RL, kita mempelajari fungsi yang mengubah data menjadi nilai antara 0 dan 1. Metode ini digunakan untuk klasifikasi, di mana hasil dari fungsi tersebut dapat diartikan sebagai peluang bahwa suatu data memiliki label 1. Teknik ini diterapkan pada klasifikasi, di mana keluaran fungsinya dapat dilihat sebagai kemungkinan bahwa data memiliki label 1 (Shai Salev-Shwarts and Shai Ben-David, 2014).

b. Random Forest (RF)

Teknik klasifikasi yang disebut Random Forest menggunakan sejumlah besar *decision tree*. Data pelatihan dan komponen acak tambahan yang dipilih dari distribusi independen digunakan untuk membangun setiap pohon (Shai Salev-Shwarts and Shai Ben-David, 2014).

c. XGBoost (XGB)

The gradient-boosted decision trees (GBDT) dijalankan menggunakan kerangka kerja XGBoost, berisi metode untuk menangani data fitur yang jarang atau “missing/zero” dan peningkatan teknologi lebih lanjut untuk bekerja dengan cepat dan efisien pada kumpulan data yang besar (Chen and Guestrin, 2016).

d. Multi-layer Perceptron (MLP)

Salah satu variasi model *Perceptron* asli Rosenblatt, yang pertama kali disarankan pada tahun 1950, adalah *Multilayer Perceptron*. *Neuron* tersusun berlapis-lapis, koneksinya selalu dari lapisan bawah ke lapisan atas, dan *neuron-neuron* di lapisan yang sama tidak terhubung. *Neuron* juga memiliki satu atau lebih lapisan tersembunyi di antara lapisan masukan dan keluarannya (Ramchoun *et al.*, 2016).

Optimasi *hyperparameter* untuk setiap model dilakukan dengan *5-fold cross-validation*, di mana skor F1 digunakan sebagai metrik utama untuk menangani ketidakseimbangan kelas secara efektif. Selain itu, untuk menangani *imbalanced data* pada penelitian ini, SMOTE diterapkan pada data pelatihan di dalam *pipeline*. Ini untuk memastikan tidak ada kebocoran data ke dalam set data pengujian (Chawla *et al.*, 2002).

5. Evaluasi Model

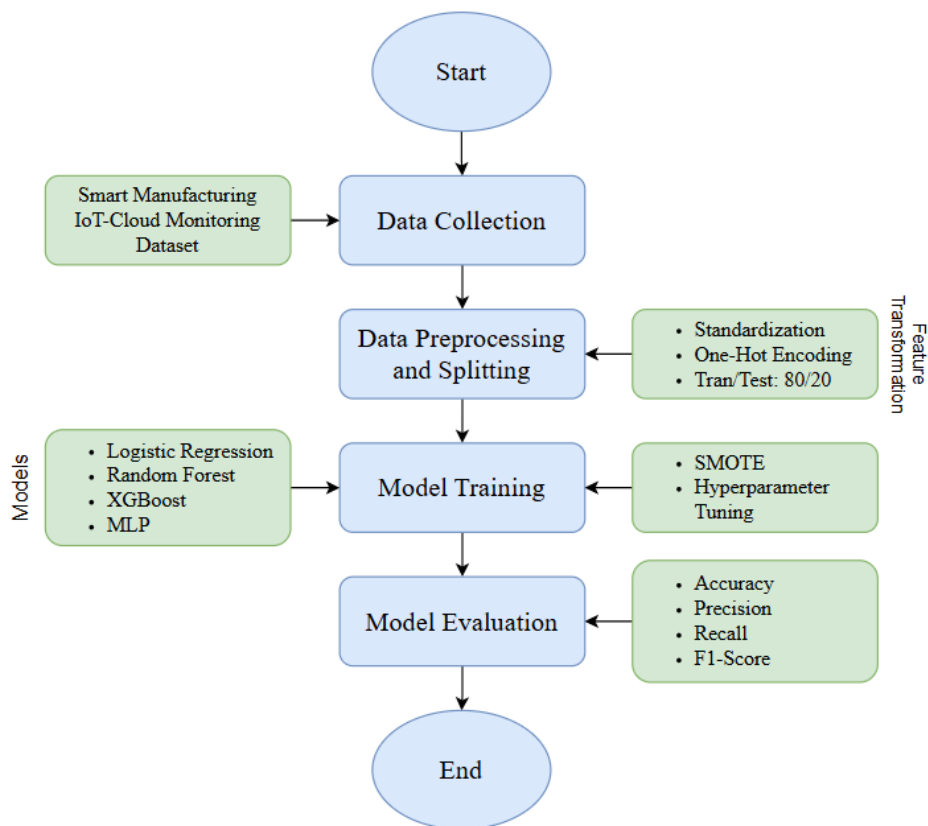
Metrik berikut digunakan untuk mengevaluasi setiap model yang disetel pada set pengujian:

- a. *Accuracy*
- b. *Precision*
- c. *Recall*
- d. *F1-Score*
- e. *ROC-AUC*

Pengklasifikasi PdM ditentukan berdasarkan akurasi dan ROC-AUC terbaik.

6. Tools

Semua proses penelitian dilakukan menggunakan Python pada platform Visual Studio Code. Beberapa *library* yang digunakan antara lain Pandas, NumPy, Scikit-learn, dan Matplotlib/Seaborn untuk analisis data, pemodelan, dan visualisasi.



Gambar 1. Metode Penelitian

HASIL DAN PEMBAHASAN

Temuan eksperimental dan analisis kerangka kerja PdM berbasis ML ditunjukkan di bagian ini.

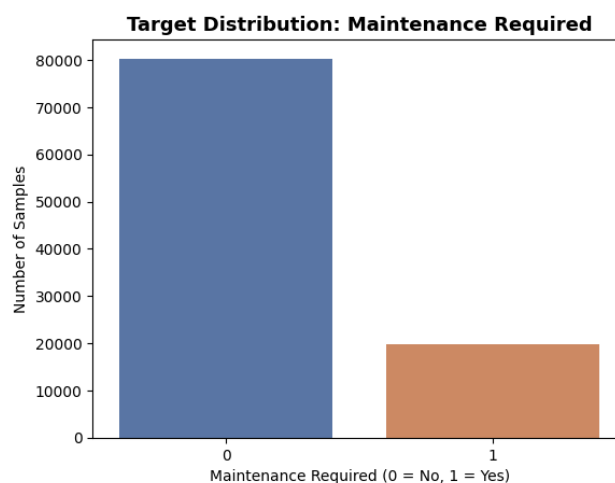
Deskripsi Data

Data yang digunakan dalam studi ini adalah data terbuka dari situs web Kaggle, yang berisi data sensor IoT dari mesin industri. Data ini terdiri dari 10.000 baris dan 13 kolom (Ziya, 2025). Data yang digunakan pada penelitian ini tidak seimbang. Untuk mengatasi masalah tersebut, SMOTE digunakan untuk melakukan *oversampling* hanya pada data latih dengan cara membuat data sintesis kelas minoritas. Tabel 1 merupakan deskripsi data yang digunakan pada penelitian ini.

Tabel 1. Deskripsi Data

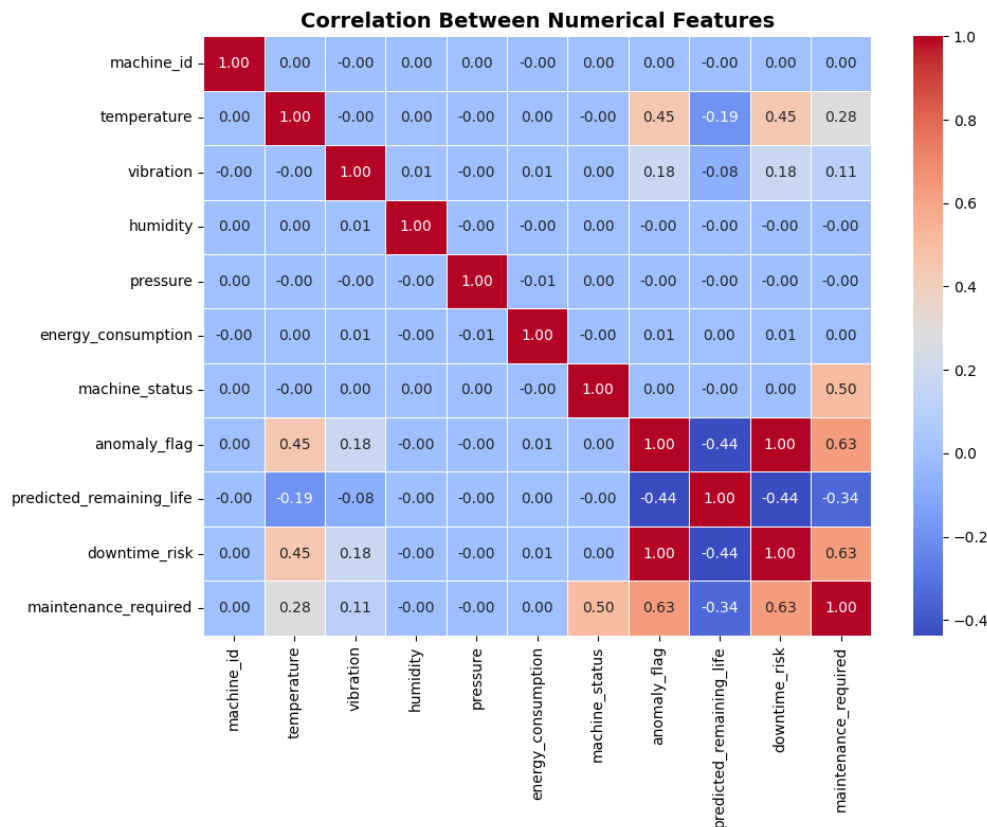
Fitur	Deskripsi
timestamp	Waktu saat pembacaan sensor direkam
machine_id	Pengidentifikasi unik untuk setiap mesin
temperature	Nilai suhu yang diukur oleh sensor di lingkungan produksi atau mesin
vibration	Nilai getaran yang direkam oleh sensor
humidity	Kelembaban lingkungan di sekitar sensor/mesin produksi
pressure	Tekanan yang diukur pada sistem mesin atau komponen dalam proses produksi
energy_consumption	Konsumsi energi dari sistem produksi atau mesin
machine_status	Menunjukkan apakah mesin dalam keadaan <i>Idle</i> , <i>Running</i> , atau <i>Failure</i>
anomaly_flag	Mengidentifikasi nilai ekstrim dalam suhu dan getaran
predicted_remaining_life	Perkiraan waktu sebelum pemeliharaan diperlukan
failure_type	Alasan kegagalan mesin
downtime_risk	Kemungkinan kerusakan mesin
maintenance_required	Kolom target (0 = Tidak, 1 = Ya)

Gambar 2 menampilkan distribusi dari fitur target (*maintenance_required*) yang tidak seimbang. Seperti yang dapat diamati, ada lebih banyak sampel dengan label 0 daripada yang berlabel 1. Karena sebagian besar mesin berfungsi dengan baik dan hanya sebagian kecil yang memerlukan pemeliharaan, ini menunjukkan bahwa data tidak seimbang.



Gambar 2. Distribusi Fitur Target (*maintenance_required*)

Gambar 3 merupakan korelasi *heatmap* yang menunjukkan kekuatan hubungan antar fitur dalam dataset. Fitur *anomaly_flag*, *downtime_risk*, dan *maintenance_required* memiliki korelasi cukup kuat satu sama lain. Hal ini masuk akal karena mesin yang mengalami anomali cenderung memerlukan pemeliharaan lebih sering dan berisiko lebih besar mengalami *downtime*. Selain itu, *temperature* memiliki korelasi sedang dengan *anomaly_flag* dan *downtime_risk*, sehingga suhu menjadi parameter yang cukup berpengaruh terhadap kondisi mesin. Sebagai hasil dari *heatmap* ini, fitur-fitur dalam model prediksi yang memiliki hubungan paling signifikan dengan label target dapat diidentifikasi.

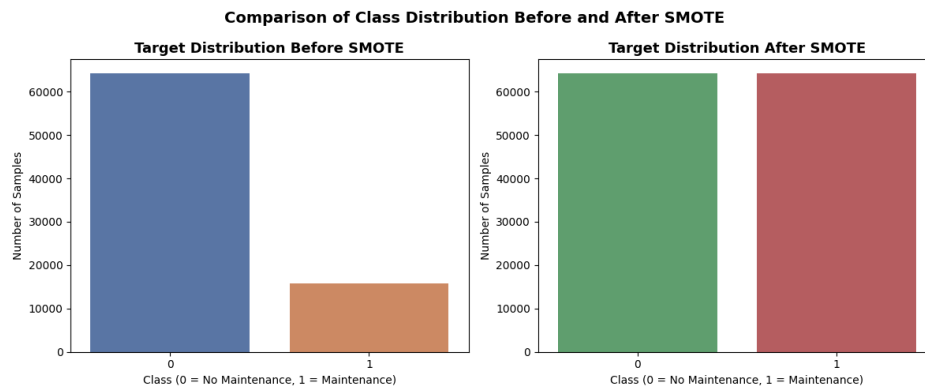


Gambar 3. *Heatmap* Korelasi

Hasil Prapemrosesan Data

Pada tahap ini, beberapa prosedur pemrosesan data digunakan untuk menjamin kesesuaian dan kualitas data sebelum prosedur pelatihan model. Pertama, dilakukan *missing values imputation* untuk menangani keberadaan nilai yang hilang pada data. Nilai kosong fitur numerik diisi menggunakan nilai rata-rata (*mean*), sementara nilai kosong fitur kategoris diisi dengan kategori yang paling sering muncul. Setelah itu, setiap kategori direpresentasikan sebagai kolom biner melalui OHE. Selanjutnya, fitur numerik distandarisasi dengan metode standardisasi, yang melibatkan perubahan skala data menjadi satu standar deviasi dan rata-rata nol. Tujuan standardisasi adalah untuk mencegah fitur dengan skala nilai lebar mendominasi selama proses pembelajaran model.

Seperti yang sudah disinggung sebelumnya, untuk mengatasi ketidakseimbangan kelas target, penelitian ini menerapkan teknik penyeimbangan kelas menggunakan SMOTE. SMOTE hanya digunakan pada data pelatihan. SMOTE tidak boleh menyentuh data uji, karena hal itu akan mengubah representasi data terhadap kondisi nyata. Metode ini mengatasi bias model terhadap kelas mayoritas dan meningkatkan kemampuan generalisasi model (Chawla *et al.*, 2002). Gambar 4 menunjukkan perbandingan distribusi kelas sebelum dan sesudah penerapan SMOTE. Setelah diterapkan SMOTE, distribusinya adalah 50:50, yang memungkinkan model mempelajari dua kelas secara lebih merata.



Gambar 4. Distribusi Target Sebelum dan Sesudah SMOTE

Fitur yang digunakan dalam penelitian ini difokuskan pada variabel sensor seperti temperature, vibration, humidity, pressure, dan energy_consumption yang secara langsung mencerminkan keadaan fisik mesin. Metrik operasi utama termasuk kenaikan suhu, getaran berlebihan, perubahan tekanan, atau penggunaan energi yang tidak biasa tercermin dalam lima aspek ini, yang mungkin merupakan indikator awal penurunan kinerja. Di sisi lain, karena berisi informasi yang terhubung langsung atau tidak langsung dengan label target, beberapa fitur seperti machine_status, anomaly_flag, predict_remaining_life, downtime_risk, dan failure_type, dikecualikan dari pemodelan untuk menghindari kebocoran data. Agar hasil prediksi dapat secara akurat mencerminkan kapasitas sistem dalam mengidentifikasi persyaratan PdM berdasarkan kondisi operasi aktual, model perlu didasarkan pada indikator sensorik kausal dan realistis.

Hasil Pelatihan Model

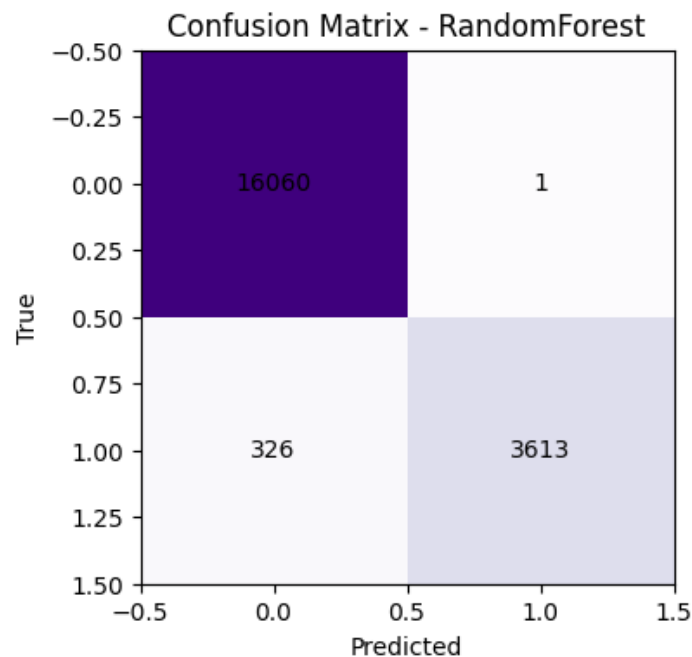
Performa dari pelatihan model dapat dilihat pada Tabel 2. Berdasarkan hasil tersebut, Random Forest mencapai akurasi tertinggi, yaitu 98.37%, mengungguli model lainnya. Hasil ini juga mengindikasikan bahwa metode *ensemble* berbasis pohon menangkap ketergantungan non-linier antara fitur sensor lebih baik daripada model linier dan *neural*.

Tabel 2. Hasil Pelatihan Model

Metrik	LR	RF	XGB	MLP
Accuracy	93.59%	98.37%	98.28%	98.28%
Precision	95.33%	99.97%	99.89%	99.86%
Recall	70.93%	91.72%	91.39%	91.42%

F1-Score	81.34%	95.67%	95.45%	95.45%
ROC-AUC	93.16%	95.95%	96.15%	96.22%

Berdasarkan hasil pelatihan pada Tabel 2, terlihat bahwa seluruh algoritma memiliki performa yang cukup tinggi, namun terdapat perbedaan signifikan pada kemampuan masing-masing dalam mendeteksi kebutuhan pemeliharaan. Dibandingkan dengan model lain, RF mencapai *accuracy*, *precision*, *f1-score*, dan ROC-AUC tertinggi. Hal ini menunjukkan bahwa RF mampu menjaga keseimbangan saat mengidentifikasi kelas mayoritas dan minoritas sambil menangkap pola rumit dalam data. Selain itu, XGB dan MLP juga menunjukkan performa yang kompetitif dengan nilai metrik yang hampir mendekati RF. Dengan kata lain, XGB dan MLP layak dipertimbangkan dalam skenario yang memerlukan model alternatif. Logistic Regression memiliki metrik *recall* terendah, yang menunjukkan bahwa model linier ini gagal mendeteksi sebagian besar sampel yang memerlukan maintenance. Secara keseluruhan, RF dianggap sebagai model paling efektif untuk masalah PdM ini karena memberikan keseimbangan yang baik antara evaluasi dan identifikasi mesin yang berisiko gagal.



Gambar 5. Confusion Matrix Hasil RF

Confusion matrix pada Gambar 5 menunjukkan bahwa model RF mampu mengklasifikasikan data dengan sangat baik. Dalam kasus mesin yang tidak memerlukan pemeliharaan, hampir semuanya diprediksi dengan benar, dan hanya satu yang terdeteksi secara tidak tepat. Mayoritas mesin yang memerlukan pemeliharaan juga dikenali secara efektif, meskipun beberapa masih terabaikan.

KESIMPULAN

Penelitian ini berhasil menerapkan pendekatan berbasis *machine learning* (ML) untuk PdM di lingkungan *smart manufacturing*. Beberapa model ML diuji, yaitu

Regresi Logistik, Random Forest, XGBoost, dan MLP. Model-model tersebut dilatih dengan metode *5-fold cross-validation* untuk mencari model terbaik dalam mengklasifikasikan kebutuhan perbaikan mesin manufaktur. Untuk mengatasi masalah kelas data yang tidak seimbang, teknik *oversampling* SMOTE diterapkan. Hasilnya, Random Forest menunjukkan kinerja terbaik dengan akurasi 98%, *precision* 99.97%, *recall* 91.72%, *f1-score* 95.67%, dan ROC-AUC 95.95%.

Walaupun hasilnya cukup menjanjikan, penelitian ini tetap memiliki batasan. Perilaku sensor tidak ditangkap secara *real-time* dan pengaruh temporal tidak tersedia pada data. Penelitian ini juga tidak menyajikan penjelasan bagaimana model ML menentukan pilihan prediksinya. Oleh karena itu, penelitian lebih lanjut dapat mempertimbangkan mengintegrasikan teknik *Explainable AI* (XAI) agar keputusan dari sistem cerdas dapat dipahami dan diinterpretasikan dengan mudah oleh pengguna. Penelitian selanjutnya juga dapat mempertimbangkan pendekatan berbasis *deep learning*, seperti LSTM atau Temporal Convolutional Networks untuk meningkatkan performa prediktif model.

DAFTAR PUSTAKA

- Carvalho, T.P. *et al.* (2019) “A systematic literature review of machine learning methods applied to predictive maintenance,” *Computers and Industrial Engineering*, 137. Available at: <https://doi.org/10.1016/j.cie.2019.106024>.
- Chawla, N. V *et al.* (2002) *SMOTE: Synthetic Minority Over-sampling Technique*, *Journal of Artificial Intelligence Research*.
- Chen, T. and Guestrin, C. (2016) “XGBoost: A Scalable Tree Boosting System.” Available at: <https://doi.org/10.1145/2939672.2939785>.
- Gaurkar, S., Kotalwar, A. and Gabale, S. (2021) “Predictive Maintenance of Industrial Machines using Machine Learning,” *International Research Journal of Engineering and Technology* [Preprint]. Available at: www.irjet.net.
- Hossan, M.Z. and Sultana, T. (2025) “AI for Predictive Maintenance in Smart Manufacturing,” *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 17(03), pp. 25–33. Available at: <https://doi.org/10.18090/samriddhi.v17i03.03>.
- Hrnjica, B. and Softic, S. (2020) “Explainable AI in Manufacturing: A Predictive Maintenance Case Study,” in *IFIP Advances in Information and Communication Technology*. Springer, pp. 66–73. Available at: https://doi.org/10.1007/978-3-030-57997-5_8.
- Kalusivalingam, K. *et al.* (2020) “Enhancing Predictive Maintenance in Manufacturing Using Machine Learning Algorithms and IoT-Driven Data Analytics,” *International Journal of AI and ML*, 1(3).
- Nazara, K.Y. (2022) “Perancangan Smart Predictive Maintenance untuk Mesin Produksi,” in *Seminar Nasional Official Statistics*, pp. 691–702. Available at: <https://doi.org/https://doi.org/10.34123/semnasoffstat.v2022i1.1575>.

- Phuyal, S., Bista, D. and Bista, R. (2020) “Challenges, Opportunities and Future Directions of Smart Manufacturing: A State of Art Review,” *Sustainable Futures*. Elsevier Ltd. Available at: <https://doi.org/10.1016/j.sftr.2020.100023>.
- Prometheus Group (2022) *Reactive Vs. Preventive Vs. Predictive Maintenance: What’s the Difference?* Available at: <https://www.prometheusgroup.com/resources/posts/reactive-vs-preventive-vs-predictive-maintenance> (Accessed: October 24, 2025).
- Ramchoun, H. *et al.* (2016) “Multilayer Perceptron: Architecture Optimization and Training,” *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(1), p. 26. Available at: <https://doi.org/10.9781/ijimai.2016.415>.
- Shai Salev-Shwarts and Shai Ben-David (2014) *Understanding Machine Learning: From Theory to Algorithms*. New York: Cambridge University Press. Available at: <http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning>.
- Sufian, A.T., Abdullah, B.M. and Miller, O.J. (2025) “Smart Manufacturing Application in Precision Manufacturing,” *Applied Sciences (Switzerland)*, 15(2). Available at: <https://doi.org/10.3390/app15020915>.
- Ziya (2025) *Smart Manufacturing IoT-Cloud Monitoring Dataset*. Available at: <https://www.kaggle.com/datasets/ziya07/smart-manufacturing-iot-cloud-monitoring-dataset> (Accessed: October 24, 2025).