

Strategi Pemeliharaan Preskriptif: Optimalisasi Keandalan Mesin Berbasis *Machine Learning* Guna Mencegah Terjadinya *Downtime* pada Mesin Industri

Bima Bagus Setyobudi*¹

¹Departemen Teknik Mesin, Fakultas Teknologi Industri dan Rekayasa Sistem, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

E-mail: 5007221221@students.its.ac.id¹

Abstrak. *Downtime* pada mesin industri dapat menyebabkan kerugian yang signifikan dalam produktivitas dan efisiensi operasional. Berbagai metode pemeliharaan seperti *Corrective*, *Descriptive*, *Diagnostic*, dan *Predictive Maintenance* memiliki keterbatasan dalam mengoptimalkan strategi mitigasi *downtime*. Oleh karena itu, penelitian ini mengimplementasikan *Prescriptive Maintenance* berbasis *Machine Learning* (XGBoost) untuk tidak hanya memprediksi kegagalan mesin tetapi juga memberikan rekomendasi langkah korektif guna mencapai zero *downtime*. Dataset AI4I 2020 *Predictive Maintenance* digunakan sebagai sumber data, dengan menerapkan berbagai teknik *preprocessing*, seperti Min-Max Scaling, SMOTE, Heatmap Korelasi, VIF, serta deteksi *outlier* menggunakan Z-Score dan IQR. Model XGBoost dilatih untuk memprediksi probabilitas kegagalan mesin, yang kemudian dianalisis menggunakan *Feature Importance* untuk mengidentifikasi penyebab utama kegagalan. Evaluasi model menunjukkan akurasi 98.12%, *precision* 97.95%, *recall* 98.23%, dan AUC-Score 99.59%, membuktikan keandalan sistem dalam mendeteksi dan mengklasifikasikan kegagalan mesin. Dengan implementasi strategi ini yang didukung oleh *IoT* dan *database real-time*, industri dapat mengoptimalkan efisiensi pemeliharaan, mengurangi *downtime* tak terduga, serta meningkatkan keandalan operasional.

Kata Kunci: *Prescriptive Maintenance; Machine Learning; XGBoost; Zero Downtime.*

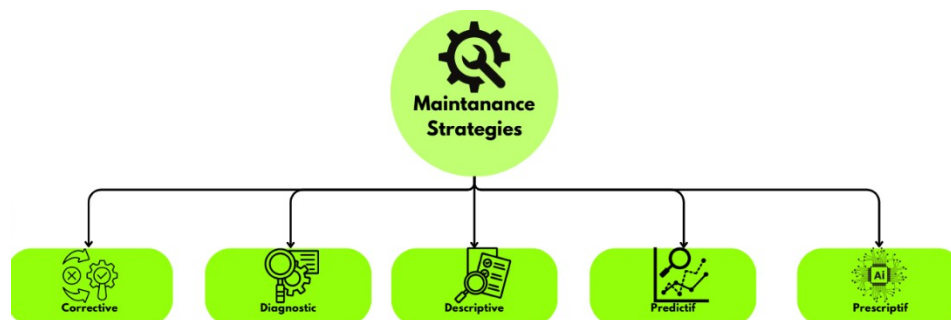
Abstract. *Downtime in industrial machinery can cause significant losses in productivity and operational efficiency. Various maintenance methods such as Corrective, Descriptive, Diagnostic, and Predictive Maintenance have limitations in optimizing downtime mitigation strategies. Therefore, this study implements Prescriptive Maintenance using Machine Learning (XGBoost) to not only predict machine failures but also provide corrective recommendations to achieve zero downtime. The AI4I 2020 Predictive Maintenance Dataset is utilized, incorporating several preprocessing techniques, including Min-Max Scaling, SMOTE, Correlation Heatmap, VIF, and outlier detection using Z-Score and IQR. The XGBoost model is trained to predict the probability of machine failure, which is further analyzed using Feature Importance to identify the root cause of failures. Model evaluation results demonstrate 98.12% accuracy, 97.95% precision, 98.23% recall, and a 99.59% AUC-Score, proving the system's reliability in detecting and classifying machine failures. With the implementation of this strategy, supported by IoT and real-time databases, industries can optimize maintenance efficiency, reduce unexpected downtime, and enhance operational reliability.*

Keywords: *Prescriptive Maintenance; Machine Learning; XGBoost; Zero Downtime.*

1. Pendahuluan

Downtime adalah periode ketika suatu sistem, mesin, atau perangkat tidak beroperasi atau tidak dapat digunakan. Dalam konteks industri, *downtime* dapat terjadi karena perawatan terjadwal, perbaikan, atau masalah teknis yang tidak terduga. Kondisi ini dapat berdampak signifikan terhadap produktivitas dan efisiensi operasional perusahaan [1]. Kasus *downtime* dapat terjadi di berbagai macam industri, terutama industri yang sangat bergantung pada peralatan mekanis, otomatisasi, atau sistem digital. Misalnya, industri manufaktur secara global tercatat mengalami rata-rata *downtime* mencapai 800 jam per tahun, yang mengakibatkan kerugian finansial sekitar 50 miliar USD akibat hilangnya produksi dan biaya perbaikan [2]. Selain itu, industri minyak dan gas juga mengalami dampak serupa, dengan rata-rata *downtime* yang dapat menyebabkan kerugian hingga 38 juta USD per tahun per perusahaan [3]. Di sektor pusat data, *downtime* juga berdampak besar, dengan estimasi kerugian sebesar 9.000 USD per menit [4]. Di Indonesia sendiri, kasus *downtime* bukan lagi hal yang asing, terutama pada sektor manufaktur. Tercatat bahwa potensi *downtime* mesin produksi dapat mencapai 15% dari total waktu operasional, menyebabkan penurunan *output* hingga 10% dan kerugian finansial miliaran rupiah per tahun [5].

Beberapa metode yang telah digunakan di industri untuk memitigasi *downtime* masih memiliki kelemahan yang dapat berdampak pada efisiensi operasional. *Corrective Maintenance* hanya dilakukan setelah terjadi kerusakan, yang menyebabkan *downtime* tidak terduga dan biaya perbaikan yang tinggi. *Descriptive Maintenance*, meskipun mengumpulkan data historis tentang kegagalan, tidak memberikan wawasan tentang pencegahan, sehingga risiko terulangnya kegagalan tetap ada. *Diagnostic Maintenance* lebih maju dengan menganalisis penyebab kegagalan, tetapi masih kurang dalam memberikan prediksi kapan kegagalan akan terjadi. *Predictive Maintenance* telah mampu memperkirakan waktu kegagalan berdasarkan data historis dan sensor *real-time*, tetapi tetap memiliki keterbatasan karena tidak menawarkan solusi optimal untuk tindakan pencegahan yang harus diambil [6].

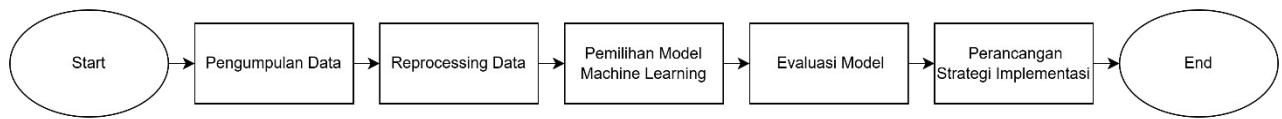


Gambar 1. Jenis Strategi *Maintenance* [7]

Oleh karena itu, pendekatan yang lebih proaktif dan berbasis kondisi diperlukan untuk mengidentifikasi potensi masalah sebelum menyebabkan *downtime* yang signifikan [6]. Salah satu optimalisasi pendekatan perawatan adalah dengan mengimplementasikan *machine learning* sebagai upaya mitigasi zero *downtime*. Dalam hal ini, *machine learning* digunakan sebagai basis pengembangan metode preskriptif, yaitu tidak hanya memprediksi kegagalan, tetapi juga memberikan saran tindakan terbaik untuk mencegah kegagalan [8]. Sebagian besar riset terdahulu hanya membahas pemodelan optimal dari sistem *machine learning* yang memiliki tingkat akurasi tinggi, tanpa mempertimbangkan strategi implementasi di industri. Oleh karena itu, dalam penelitian ini akan dibahas mengenai pengembangan aplikatif dari sistem *machine learning* yang paling optimal, serta strategi implementasi penerapannya dalam industri guna memitigasi zero *downtime* pada permesinan industri

2. Metode

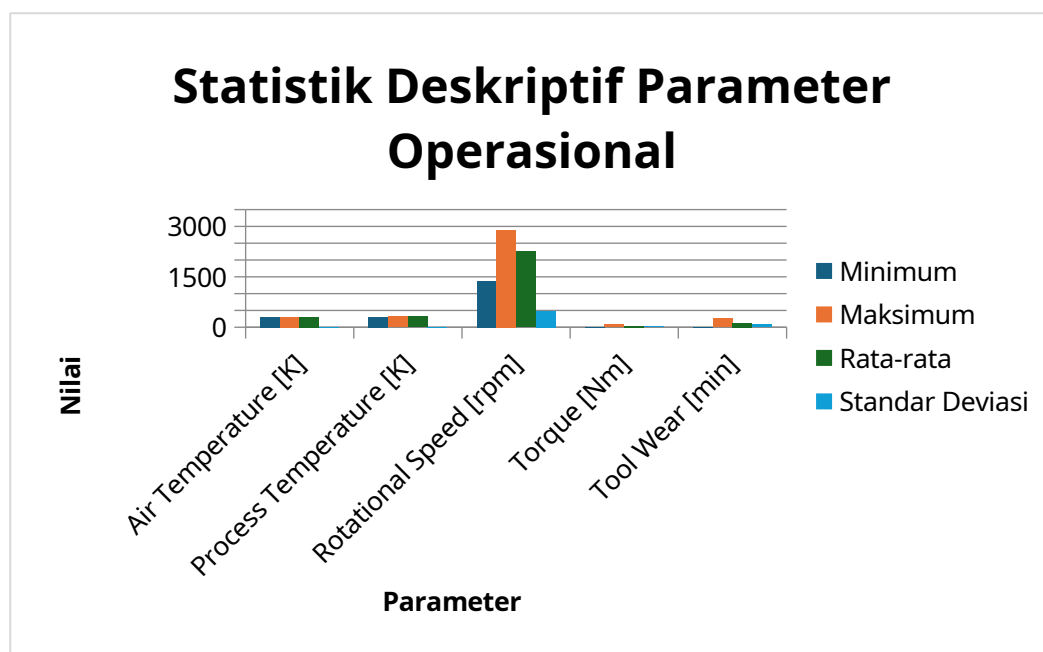
Dalam penyusunan sistem dan strategi implementasinya, diterapkan metode sistematis sesuai dengan Gambar 2.



Gambar 2. Metode Penelitian

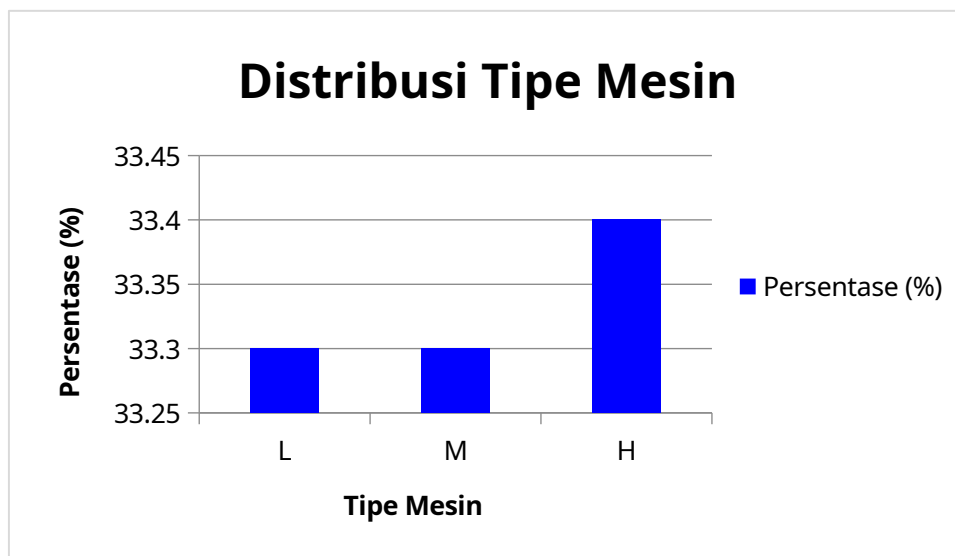
2.1. Pengumpulan Data.

Pengumpulan data yang digunakan diambil dari data set sintesis *open source* “AI4I 2020 Predictive Maintenance Dataset” [9]. Dataset ini terdiri dari 10.000 entri dengan berbagai parameter operasional yang digunakan untuk memprediksi kemungkinan kegagalan mesin. Parameter yang tersedia mencakup suhu udara, suhu proses, kecepatan rotasi, torsi, dan lama penggunaan alat, yang masing-masing memiliki pengaruh terhadap performa mesin yang rentang datanya terurai pada Gambar 3.



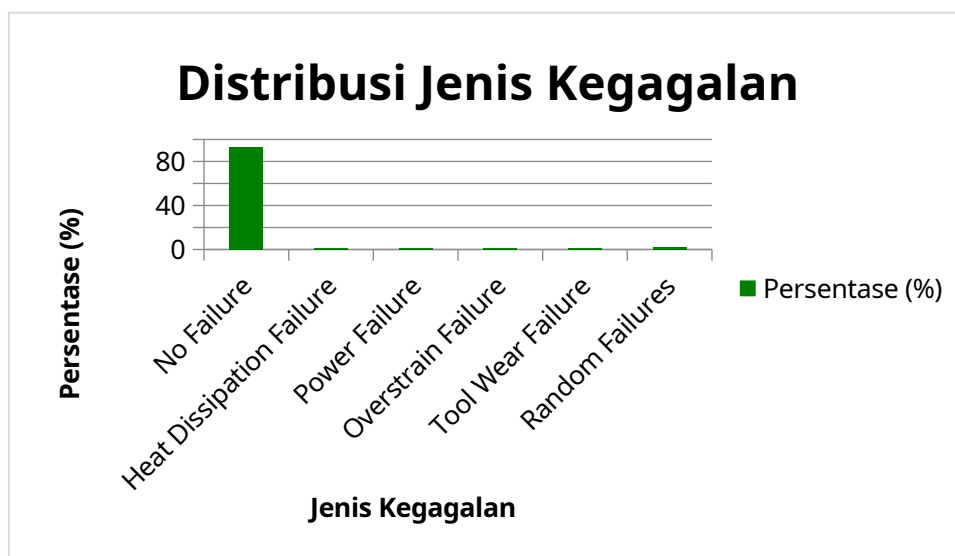
Gambar 3. Statistik Deskriptif Parameter Operasional

Selain itu, dataset ini juga mencatat jenis kegagalan yang terjadi, seperti *Heat Dissipation Failure*, *Power Failure*, *Overstrain Failure*, *Tool Wear Failure*, dan *Random Failures*, serta mencantumkan apakah mesin mengalami kegagalan atau tidak. Pada dataset terklasifikasi tiga tipe mesin yang berbeda yaitu L, M, H dengan persentase masing-masing ditunjukkan pada gambar 4.



Gambar 4. Distribusi Tipe Mesin

Dari seluruh data pada dataset, 93.6% mesin tidak mengalami kegagalan, sementara 6.4% mengalami kegagalan, yang menunjukkan bahwa sebagian besar mesin masih dalam kondisi baik. Dalam hal ini distribusi kegagalan pada mesin ditunjukkan pada Gambar 5.



Gambar 5. Distribusi Jenis Kegagalan

2.2. Pre-Processing Data.

Dataset memerlukan *preprocessing* sebelum digunakan dalam model untuk memastikan hasil prediksi yang lebih akurat dan tidak bias. Salah satu alasan utama adalah adanya fitur yang tidak relevan, seperti *UDI* (*Unique Device Identifier*) dan *Product ID*, yang tidak memiliki pengaruh langsung terhadap prediksi kegagalan mesin. Menghapus fitur ini dapat mengurangi kompleksitas model dan meningkatkan efisiensi komputasi. Selain itu, *dataset* ini mengandung parameter operasional dengan skala yang berbeda, seperti suhu udara (298.1 - 308.6 K), kecepatan rotasi (1.370 - 2.886 rpm), torsi (3.8 - 74.9 Nm), dan *Tool Wear* (0 - 253 menit). Tanpa normalisasi, fitur dengan nilai yang lebih besar dapat

mendominasi model, dalam hal ini diterapkan Min-Max Scaling standarisasi untuk menyamakan skala data.

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Di mana,

- X_n = Nilai Normalisasi
- X_{min} = Nilai Minimum Suatu Fitur
- X_{max} = Nilai Maksimum Suatu Fitur

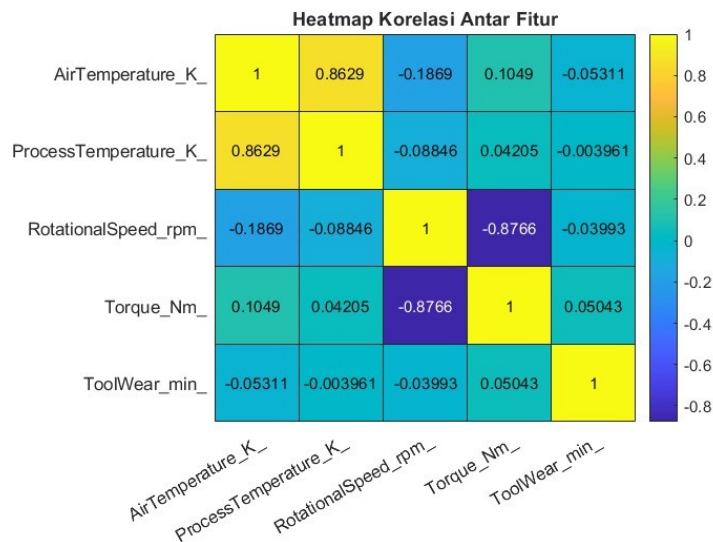
Selain itu, *dataset* ini mengalami ketimpangan kelas (*class imbalance*), di mana 93.6% mesin tidak mengalami kegagalan, sedangkan hanya 6.4% mengalami kegagalan. Ketidakseimbangan ini dapat menyebabkan model lebih cenderung memprediksi "tidak gagal", sehingga sulit mendeteksi kegagalan yang jarang terjadi. Untuk mengatasi masalah ini, metode SMOTE (*Synthetic Minority Over-sampling Technique*) digunakan untuk menambah data kegagalan secara sintetis, atau dengan *undersampling* kelas mayoritas agar distribusi lebih seimbang. Dalam, hal ini SMOTE menyeimbangkan data kegagalan (6.4%) menjadi 50:50 agar model tidak bias terhadap kelas mayoritas.

$$X_{new} = X_i + \lambda x (X_j - X_i) \quad (2)$$

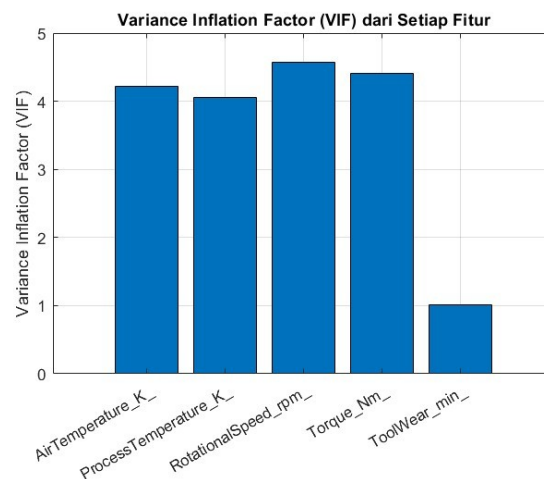
Di mana,

- X_i = Sampel minoritas yang dipilih secara acak.
- X_j = Tetangga terdekat dari X_i (dipilih menggunakan K-Nearest Neighbors).
- λ = Nilai acak antara 0 dan 1, yang menentukan seberapa jauh titik baru X_{new} berada di antara X_i dan X_j .

Lebih lanjut, terdapat multikolinearitas antar fitur, seperti hubungan antara suhu udara dan suhu proses, serta kecepatan rotasi dan torsi, yang dapat menyebabkan redundansi dan *overfitting*. Dengan menggunakan *heatmap* korelasi atau *Variance Inflation Factor (VIF)*, fitur yang terlalu berkorelasi tinggi dapat dihapus agar model lebih stabil. Dalam hal ini, Berdasarkan *heatmap* korelasi, Temperatur Udara dan Temperatur Proses memiliki korelasi tinggi (0.8629), yang menunjukkan kemungkinan multikolinearitas. Selain itu, RPM dan Torsi memiliki korelasi negatif sangat kuat (-0.8766), yang dapat menyebabkan redundansi dalam model sehingga salah satu parameter perlu dihilangkan. Adapun dari diagram VIF, fitur dengan VIF tinggi (>4) adalah Temperatur Udara, Temperatur Proses, Kecepatan Putar dan Torsi, yang mengindikasikan multikolinearitas yang signifikan. Untuk meningkatkan stabilitas model, salah satu dari pasangan Temperatur Udara atau Temperatur Proses serta salah satu dari Kecepatan Putar atau Torsi dapat dihapus agar model lebih baik dalam menangkap hubungan antar variabel tanpa redundansi.

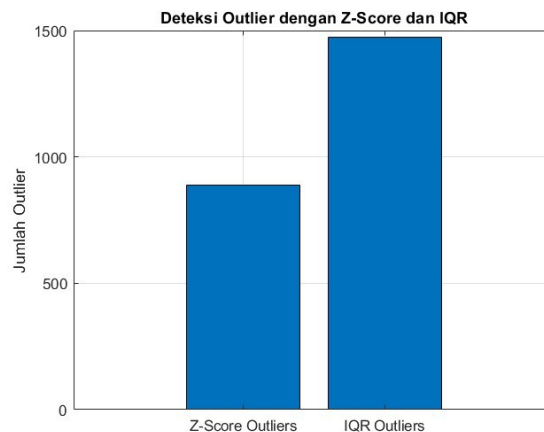


Gambar 6. Heat Map Diagram



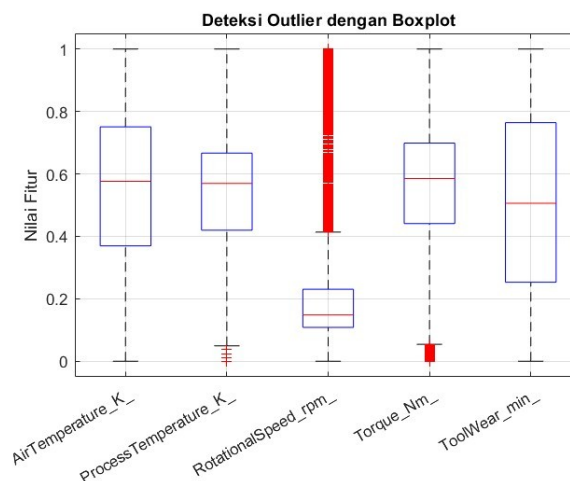
Gambar 7. VIF Diagram

Preprocessing juga diperlukan untuk mengatasi *outlier* dan anomali, di mana beberapa data memiliki nilai ekstrem, seperti kecepatan rotasi melebihi 2.886 rpm atau torsi yang sangat tinggi dibandingkan rata-rata. Untuk mengidentifikasi dan menghilangkan *outlier* ini, digunakan metode Z-score (nilai di atas 3 dianggap *outlier*) atau *Interquartile Range* (IQR) untuk menghapus nilai ekstrem di luar Q1 dan Q3. Pada Gambar 8. IQR mendeteksi lebih banyak *outlier* dibandingkan Z-Score, yang berarti banyak data berada di luar rentang kuartil meskipun tidak semuanya memiliki Z-score ekstrem. Ini menunjukkan bahwa distribusi data memiliki nilai ekstrem yang signifikan.



Gambar 8. Distribusi *Outlier* dengan Z-Square dan IQR

Dalam hal ini, parameter kecepatan putar dan torsi yang cenderung mengalami *outlier* sehingga diperlukan penghilangan *outlier*.



Gambar 9. Distribusi Deteksi Outlier pada Dataset

Dengan menerapkan *preprocessing* yang tepat, dataset AI4I 2020 dapat digunakan secara optimal dalam model prediktif, memungkinkan deteksi kegagalan mesin lebih akurat.

2.3. Pemilihan Model Macine Learning.

Dalam proses ini, metode *Machine Learning* yang digunakan dalam proses preskriptif adalah dengan mengimplementasikan XGBoost (*Extreme Gradient Boosting*). XGBoost merupakan salah satu algoritma *machine learning* berbasis *ensemble learning* yang banyak digunakan dalam berbagai bidang, termasuk prediksi kegagalan mesin untuk mencapai zero downtime [10]. Algoritma ini bekerja dengan teknik *boosting*, di mana setiap pohon keputusan baru yang dibangun akan memperbaiki kesalahan dari pohon sebelumnya, sehingga menghasilkan model yang lebih akurat dan *robust* [11]. Salah satu keunggulan utama XGBoost adalah kemampuannya untuk mencegah *overfitting* melalui teknik regularisasi L1 (Lasso) dan L2 (Ridge), serta efisiensi komputasi tinggi karena mendukung pemrosesan paralel [12]. Selain itu, XGBoost sangat efektif dalam menangani data yang tidak seimbang, yang sering ditemukan dalam skenario prediksi kegagalan mesin [13]. XGBoost juga sering mendapat hasil yang optimal pada penelitian prediksi kegagalan mesin sejenis [14].

Dalam konteks preskriptif zero *downtime*, XGBoost mampu memprediksi persentase probabilitas kegagalan suatu komponen mesin berdasarkan data historis dan kondisi operasionalnya. Probabilitas ini

dihitung menggunakan skor logit yang dihasilkan oleh model dan dikonversi menjadi probabilitas dengan fungsi sigmoid,

$$P(\text{failure}) = \frac{1}{1+e^{-\text{logit_score}}} \quad (3)$$

Di mana *logit_score* adalah skor yang dihasilkan oleh model sebelum transformasi sigmoid (Chen et al., 2019). Dengan memanfaatkan probabilitas ini, perusahaan dapat melakukan perawatan prediktif untuk mencegah kegagalan sebelum terjadi, sehingga mengurangi waktu henti (*downtime*) dan biaya perawatan yang tidak terduga. Dalam prosesnya, data akan di bagi menjadi 80% *Training* - 20% *Testing*.

2.4. Evaluasi Model.

Evaluasi model yang diterapkan pada proses *machine learning* dilakukan menggunakan beberapa parameter evaluasi kinerja model klasifikasi, termasuk *Confusion Matrix*, *ROC Curve*, *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *AUC-Score*, yang bertujuan untuk menilai performa prediksi kegagalan mesin. *Confusion Matrix* digunakan untuk melihat jumlah prediksi yang benar dan salah dalam empat kategori, yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN), sehingga dapat memberikan gambaran mengenai efektivitas model dalam mengklasifikasikan data dengan benar [15].

Dari *Confusion Matrix*, *Accuracy* dihitung sebagai proporsi prediksi yang benar terhadap total prediksi, yang menunjukkan seberapa andal model dalam melakukan klasifikasi secara keseluruhan [16]. *Precision* mengukur seberapa akurat model dalam memprediksi kegagalan, yaitu proporsi prediksi positif yang benar dari seluruh prediksi positif [17]. Sementara itu, *Recall* digunakan untuk menilai sejauh mana model dapat mendeteksi semua kegagalan yang benar-benar terjadi, yang sangat penting dalam kasus di mana kegagalan mesin harus diidentifikasi secara akurat untuk menghindari *downtime* yang tidak terduga [18]. *F1-Score* digunakan sebagai metrik gabungan yang menyeimbangkan *Precision* dan *Recall*, terutama ketika *dataset* memiliki distribusi kelas yang tidak seimbang [19].

Selain itu, *ROC Curve* (*Receiver Operating Characteristic Curve*) digunakan untuk mengevaluasi kemampuan model dalam membedakan antara mesin yang mengalami kegagalan dan yang tidak. *ROC Curve* memvisualisasikan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai *threshold* prediksi [20]. *AUC-Score* (*Area Under Curve*) kemudian diukur untuk mengetahui seberapa baik model dapat membedakan kedua kelas tersebut, di mana nilai yang mendekati 1 menunjukkan bahwa model memiliki performa yang sangat baik dalam membedakan mesin yang akan gagal dan yang tidak [21]. Setiap parameter evaluasi ini memiliki peran penting dalam menentukan efektivitas model *machine learning* dalam memprediksi kegagalan mesin. Dengan menggunakan metrik evaluasi yang tepat, model dapat dioptimalkan untuk mengurangi jumlah kesalahan prediksi, meningkatkan deteksi kegagalan yang akurat, dan pada akhirnya mengurangi *downtime* industri secara signifikan.

2.5. Perancangan Strategi Implementasi.

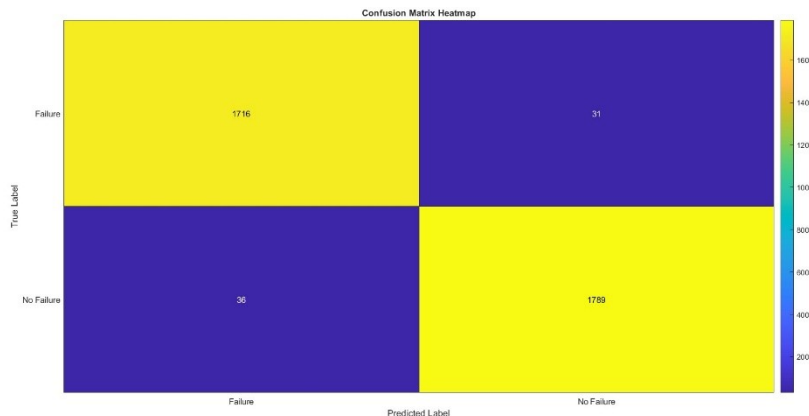
Strategi ini didasarkan pada riset studi literatur dari berbagai jurnal yang membahas metode perawatan berbasis data, teknik optimasi keputusan, serta implementasi kecerdasan buatan dalam pemeliharaan industri. Melalui studi literatur, berbagai pendekatan dibandingkan untuk mencari basis data dalam pengembangan strategi yang paling cocok dan efektif dalam mengurangi *downtime* dan meningkatkan efisiensi operasional disertai dengan penentuan teknologi dan metode integrasi sistem yang sesuai untuk menunjang fungsi preskriptif.

3. Hasil dan Pembahasan

3.1. Hasil Evaluasi Model

Evaluasi model direpresentasikan dengan parameter *Confusion Matrix*, *ROC Curve*, *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *AUC-Score*. Pada *Confusion Matrix* pada Gambar 10. Ditunjukkan bahwa model memiliki 1716 *True Positives* (TP), yang berarti model berhasil mengidentifikasi 1716 kasus kegagalan dengan benar. Sementara itu, terdapat 31 *False Negatives* (FN), yang menunjukkan

bahwa model salah mengklasifikasikan 31 kasus yang seharusnya gagal sebagai tidak gagal. Di sisi lain, model juga menghasilkan 36 *False Positives* (FP), di mana kasus yang seharusnya tidak mengalami kegagalan justru diklasifikasikan sebagai gagal. Terakhir, terdapat 1789 *True Negatives* (TN), yang berarti model berhasil mengidentifikasi 1789 kasus non-kegagalan dengan benar.



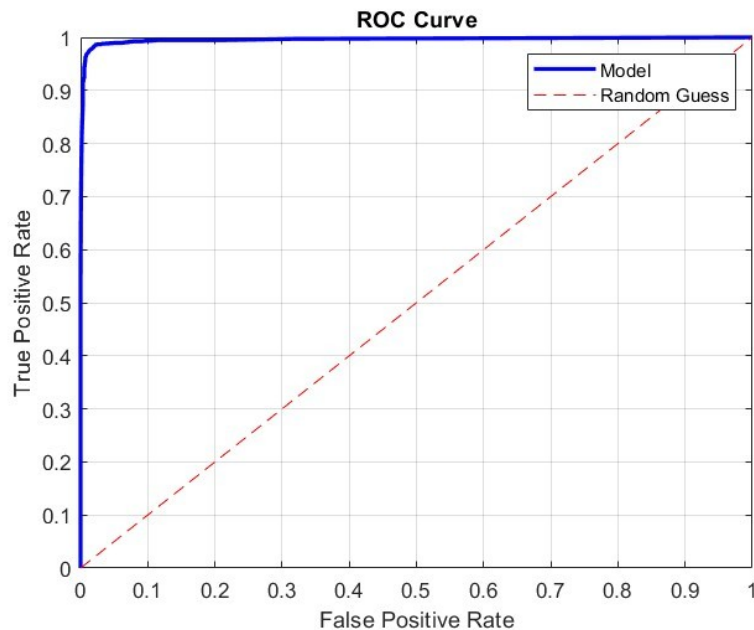
Gambar 10. Confussion Matrix

Dengan hasil ini, model klasifikasi yang diuji menunjukkan performa yang sangat baik dengan berbagai metrik evaluasi yang tinggi. Akurasi model mencapai 98.12%, yang berarti hampir seluruh prediksi dilakukan dengan benar dibandingkan dengan total data yang diuji. *Precision* sebesar 97.95% menunjukkan bahwa dari seluruh prediksi positif yang dibuat oleh model, hanya sekitar 2.05% yang salah klasifikasi sebagai positif. Sementara itu, *recall* sebesar 98.23% mengindikasikan bahwa model mampu menangkap 98.23% dari semua kasus positif yang sebenarnya ada, sehingga hanya sedikit yang terlewat. F1-score, yang merupakan kombinasi antara *precision* dan *recall*, berada pada 98.09%, menunjukkan keseimbangan yang baik antara keduanya. Selain itu, *AUC-Score* sebesar 99.59% menandakan bahwa model hampir sempurna dalam membedakan kelas positif dan negatif. Secara keseluruhan, metrik-metrik ini menunjukkan bahwa model memiliki tingkat keandalan yang sangat tinggi dalam melakukan klasifikasi, dengan kesalahan prediksi yang minimal.

Tabel 1. Ukuran Performa dari Model

Model	Accuracy	Precision	Recall	F1-Score	AUC-Score
XGBoost	98.12%,	97.95%	98.23%	98.09%	99.59%

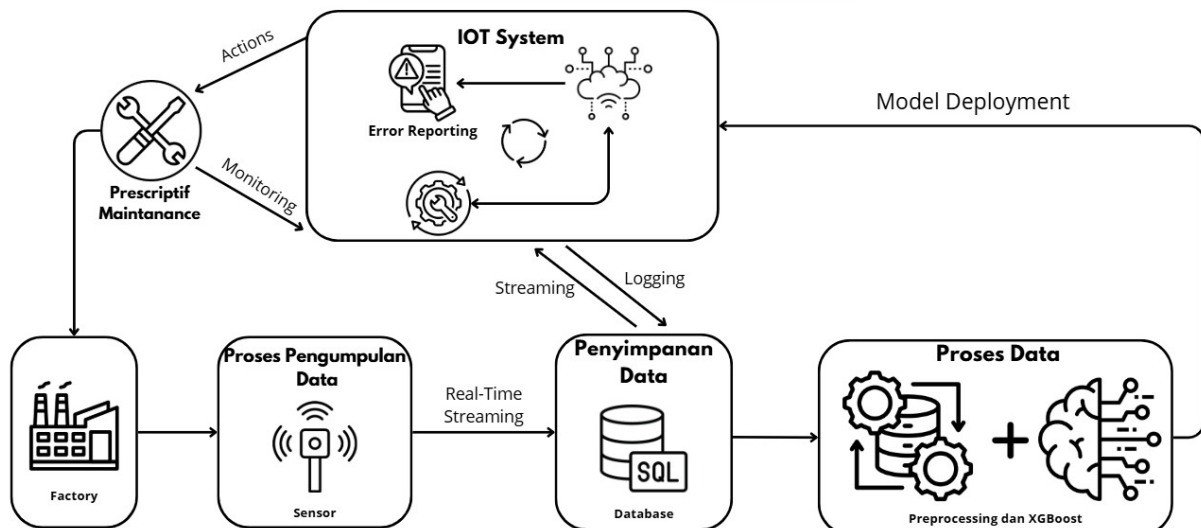
Pada ROC Curve (Receiver Operating Characteristic) pada Gambar 11. memberikan gambaran lebih lanjut tentang performa model dalam membedakan kedua kelas. Kurva biru menunjukkan performa model, sedangkan garis merah putus-putus adalah acuan untuk model acak (random classifier) dengan AUC = 0.5. Semakin dekat kurva ke sudut kiri atas, semakin baik kinerja model. Pada gambar ini, ROC curve menunjukkan bahwa model memiliki AUC yang sangat tinggi, mendekati 1, yang mengindikasikan bahwa model hampir sempurna dalam membedakan antara kasus Failure dan No Failure. Dengan kata lain, model ini memiliki kemampuan deteksi yang sangat baik dan sangat kecil kemungkinan untuk salah dalam membedakan dua kelas.



Gambar 11. ROC Curve

3.2. Strategi Implementasi Model

Dalam strategi implementasinya sistem ini menggabungkan *IoT*, *database*, *machine learning* (XGBoost), dan *Prescriptive Maintenance* (RxM) seperti sumber referensi [22] yang disesuaikan metode dan penggunaannya untuk mencegah kegagalan mesin secara optimal.



Gambar 12. Alur Implementasi Sistem

Dalam hal ini, sensor yang terintegrasi sistem *IoT* akan mengukur kondisi mesin secara *real-time*, data kemudian dikirim ke SQL Database melalui *streaming* untuk analisis lebih lanjut. Kemudian data di olah pada tahap PreProcessing guna menyiapkan data agar lebih bersih, terstruktur, dan siap digunakan dalam model *machine learning*, langkah selanjutnya adalah pengolahan data pada XGBoost yang dilatih dilatih dengan data historis kegagalan mesin, model kemudian di-*deploy* di IoT Sistem untuk memprediksi persentase probabilitas kegagalan.

Tabel 2. Contoh Representasi Model Prediksi

ID Mesin	Probabilitas Kegagalan
Machine-101	70%
Macihne-102	80%
Machine-103	78%

Mesin yang terprediksi memiliki probabilitas kegagalan tinggi dapat diproses lebih lanjut untuk dikoreksi sebelum mengalami downtime. Dalam hal memberikan langkah korektif terbaik, dapat diimplementasikan preskriptif *maintenance* yang dapat menawarkan langkah korektif untuk memitigasi *downtime*. Dalam hal ini, dapat diterapkan *Root Cause Analysis* (RCA) untuk menganalisis penyebab utama kegagalan berdasarkan pola hubungan antar variabel operasional, dalam penerapan RCA diperlukan metode tambahan untuk memproses data yaitu Feature Importance dari XGBoost yang dapat memberikan korelasi antara variabel operasional dan kegagalan mesin serta menentukan Fitur dengan bobot tertinggi yang berkontribusi dalam kegagalan mesin, untuk dijadikan pertimbangan langkah korektif di samping juga melihat data sensor untuk mengetahui anomali pada data hasil deteksi sensor.

Tabel 3. Contoh Representasi Model Preskriptif

Probabilitas Kegagalan	Root Cause Analysis	Rekomendasi Preskriptif
90 %	Suhu tinggi 95°C (bobot feature 45%) & Kecepatan Rotasi tinggi 3100 RPM (bobot featurenya 30%)	Kurangi suhu dengan meningkatkan pendinginan dan turunkan kecepatan rotasi

Dalam contoh pada Tabel 3. Terdeteksi dari sensor sistem anomali pada suhu dan kecepatan rotasi mesin yang terlalu tinggi, adapun dari kedua parameter tersebut, bisa dikatakan penyebab utama dari kegagalan adalah dari temperatur mesin yang tinggi berdasarkan dari bobotnya diikuti dengan kecepatan rotasi mesin, sehingga langkah preskriptifnya adalah mengurangi suhu mesin dan kecepatan rotasi. Dalam hal ini, model *machine learning* dapat terus diperbarui dengan data terbaru untuk meningkatkan akurasi dan menyesuaikan kondisi terbaru dari mesin.

4. Kesimpulan

Implementasi *Prescriptive Maintenance* berbasis XGBoost dalam sistem industri memungkinkan deteksi dini kegagalan mesin dengan akurasi tinggi, sekaligus memberikan rekomendasi korektif yang optimal. Model ini tidak hanya memprediksi probabilitas kegagalan, tetapi juga menganalisis penyebab utama menggunakan *Feature Importance*, sehingga tindakan pencegahan dapat difokuskan pada faktor yang paling berkontribusi terhadap kegagalan.

Hasil evaluasi model menunjukkan bahwa implementasi XGBoost dalam prediksi kegagalan mesin memberikan performa yang sangat baik, dengan akurasi mencapai 98.12%, *precision* 97.95%, *recall* 98.23%, dan AUC-Score sebesar 99.59%. Hasil ini menandakan bahwa model mampu mendeteksi kegagalan dengan akurat, sekaligus meminimalkan kesalahan dalam klasifikasi. Confusion Matrix menunjukkan jumlah *True Positive* (TP) dan *True Negative* (TN) yang tinggi, serta kesalahan prediksi

(FP dan FN) yang rendah, membuktikan keandalan sistem dalam mengidentifikasi mesin yang benar-benar membutuhkan perawatan.

Strategi implementasi sistem ini menggabungkan sensor IoT, *database*, dan model *machine learning* XGBoost, yang memungkinkan pemantauan kondisi mesin secara *real-time*. Data yang dikumpulkan akan diproses melalui tahap *preprocessing* sebelum digunakan dalam model prediktif, yang kemudian menghasilkan probabilitas kegagalan mesin dalam bentuk persentase. Mesin yang terdeteksi memiliki probabilitas kegagalan tinggi akan dianalisis lebih lanjut menggunakan *Feature Importance*, sehingga penyebab utama kegagalan dapat ditentukan. Berdasarkan faktor dengan bobot tertinggi, sistem akan memberikan rekomendasi preskriptif yang sesuai, seperti penyesuaian suhu, kecepatan rotasi, atau pemeriksaan sistem kelistrikan.

Dengan pendekatan ini, industri dapat mengoptimalkan strategi pemeliharaan mesin, mengurangi *downtime* yang tidak terduga, serta meningkatkan efisiensi dan keandalan operasional. Integrasi *Prescriptive Maintenance* berbasis XGBoost ini menjadi solusi efektif dalam mencapai zero *downtime*, memungkinkan sistem beradaptasi dengan data terbaru, dan secara otomatis memberikan tindakan pencegahan yang optimal guna menjaga kontinuitas produksi.

5. Referensi

- [1] R. Wolniak, "Downtime in the Automotive Industry Production Process – Cause Analysis," *Quality Innovation Prosperity*, vol. 23, no. 2, pp. 101–118, 2019.
- [2] L. Miller and X. Zhang, "Optimizing Maintenance Strategies for Industrial Systems Using Machine Learning," *International Journal of Advanced Manufacturing Technology*, vol. 125, no. 3, pp. 567–580, 2024.
- [3] M. Shahin, F. F. Chen, A. Hosseinzadeh, and N. Zand, "Using Machine Learning and Deep Learning Algorithms for Downtime Minimization in Manufacturing Systems: An Early Failure Detection Diagnostic Service," *International Journal of Advanced Manufacturing Technology*, vol. 128, no. 9, pp. 3857–3883, 2023.
- [4] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron Spectroscopy Studies on Magneto-Optical Media and Plastic Substrate Interface," *IEEE Transactions on Journal of Magnetics Japan*, vol. 2, pp. 740–741, Aug. 1987.
- [5] ITSb, "Analisis dampak downtime pada industri manufaktur di Indonesia," 2022. [Online]. Available: <https://www.itsb.ac.id>.
- [6] L. Miller and X. Zhang, "Optimizing maintenance strategies for industrial systems using machine learning," *Int. J. Adv. Manuf. Technol.*, vol. 125, no. 3, pp. 567–580, 2024.
- [7] M. Shahin, F. F. Chen, A. Hosseinzadeh, and N. Zand, "Using machine learning and deep learning algorithms for downtime minimization in manufacturing systems: An early failure detection diagnostic service," *Int. J. Adv. Manuf. Technol.*, vol. 128, no. 9, pp. 3857–3883, 2023.
- [8] A. K. Sahoo, A. K. Dubey, and A. K. Sharma, "Intelligent machine learning-based total productive maintenance approach for achieving zero downtime in industrial machinery," *Int. J. Adv. Manuf. Technol.*, vol. 129, no. 9–10, pp. 3141–3164, 2023. [Online]. Available: <https://doi.org/10.1007/s00170-023-12020-w>.
- [9] UCI Machine Learning Repository, "AI4I 2020 Predictive Maintenance Dataset," 2020. [Online]. Available: <https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset>.
- [10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, San Francisco, CA, USA, 2016, pp. 785–794.
- [11] T. Chen, "Introduction to XGBoost: Scalable and flexible gradient boosting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4563–4575, 2020.
- [12] S. Chakraborty and A. Joseph, "Machine learning models for predictive maintenance: A review," *J. Manuf. Sci. Eng.*, vol. 139, no. 12, pp. 1–15, 2017.

- [13] J. Brown, M. A. Hassan, and L. Wang, "Handling class imbalance in predictive maintenance using SMOTE and ensemble learning," *IEEE Trans. Ind. Inform.*, vol. 69, no. 4, pp. 3214–3226, 2021.
- [14] D. K. Yadav, A. Kaushik, and N. Yadav, "Predicting machine failures using machine learning and deep learning algorithms," *Sustainable Manufacturing and Service Economics*, vol. 3, p. 100029, 2024.
- [15] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
- [16] S. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427–437, 2009.
- [17] C. X. Ling, J. Huang, and H. Zhang, "AUC: A better measure than accuracy in comparing learning algorithms," in *Proc. 17th Can. Conf. Artif. Intell.*, London, ON, Canada, 2003, pp. 329–341.
- [18] F. Provost and P. Domingos, "Well-trained PETs: Improving probability estimation trees," *CeDER Working Paper*, Stern School of Business, New York University, 2000.
- [19] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.
- [20] A. Ferri, J. Hernández-Orallo, and M. A. Salido, "Volume under the ROC surface for multi-class problems," in *Proc. 14th Eur. Conf. Mach. Learn. (ECML)*, Cavtat-Dubrovnik, Croatia, 2003, pp. 108–120.
- [21] A. C. Lorena, A. C. de Carvalho, and J. Gama, "A review on the combination of binary classifiers in multiclass problems," *Artif. Intell. Rev.*, vol. 30, no. 1–4, pp. 177–196, 2008.
- [22] B. Taşçı, A. Omar, and S. Ayvaz, "Remaining useful lifetime prediction for predictive maintenance in manufacturing," *Computers & Industrial Engineering*, vol. 184, p. 109566, 2023.