Klasifikasi *Maintenance* Perangkat TI Menggunakan Algoritma Random Forest dan XGBOOST Classifier

Gabriella Mustika Anugraheni*, Woro Isti Rahayu, Fatia Amalia Maresti
Sekolah Teknologi Informasi, Universitas Logistik dan Bisnis Internasional

Received: 24 July 2026 Accepted: 25 August 2026 Published: 1 December 2026

*gabriellamshn5@gmail.com

Kata Kunci: *Machine Learning,* Random Forest, XGBoost, *Maintenance,* Klasifikasi.

DSI: Jurnal Data Science Indonesia is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

Abstrak : Efektivitas tata kelola pemeliharaan infrastruktur TI sangat krusial untuk meminimalkan *downtime*, namun metode identifikasi manual kerap menimbulkan subjektivitas dan keterlambatan keputusan. Penelitian ini mengembangkan model otomatis berbasis *machine learning* untuk mengategorikan pemeliharaan perangkat TI menjadi *Preventive Maintenance* (PM) dan *Corrective Maintenance* (CM) pada proyek X-System di PT XYZ. Algoritma Random Forest dan XGBoost dievaluasi menggunakan 15.204 baris *dataset* dengan lima fitur operasional (brand, kategori, umur perangkat, waktu tunggu, dan log keterangan), serta dioptimasi menggunakan SMOTE untuk mengatasi ketidakseimbangan kelas. Analisis komparatif menemukan bahwa pelibatan fitur tekstual 'keterangan' memicu kebocoran data (*data leakage*) yang menghasilkan akurasi semu 100% (Random Forest) dan 99,97% (XGBoost). Saat dieksklusi, akurasi global menjadi lebih realistis di angka 84,21% dan 85,89%. Analisis *Feature Importance* mengonfirmasi terjadinya *overfitting* ekstrem yang didorong oleh dominasi log keterangan tersebut. Oleh karena itu, untuk membangun model yang *robust* dan dapat digeneralisasi, direkomendasikan mengeksklusi fitur 'keterangan' agar sistem belajar murni dari atribut fisik operasional perangkat keras.

PENDAHULUAN

Dalam ekosistem industri, efektivitas tata kelola pemeliharaan (*maintenance*) infrastruktur teknologi informasi (TI) merupakan determinan krusial dalam menjamin keberlanjutan operasional, memperpanjang estimasi masa pakai aset, serta meminimalkan risiko kerugian akibat waktu henti (*downtime*)[1]. Secara konseptual, aktivitas pemeliharaan mencakup segenap upaya terstruktur untuk mempertahankan suatu aset agar senantiasa berada dalam kondisi siap operasi secara prima[2]. Strategi ini terbagi menjadi dua pendekatan utama, *Preventive Maintenance* (PM) yang berorientasi pada tindakan preventif terjadwal secara berkala guna mengontrol keandalan perangkat, dan *Corrective Maintenance* (CM) yang bersifat reaktif untuk memulihkan fungsi aset pasca terjadinya kerusakan atau kegagalan signifikan[3].

Dari perspektif manajemen operasional, urgensi penerapan strategi pemeliharaan preventif yang terstruktur serta tata kelola suku cadang yang terintegrasi penuh merupakan pilar utama dalam meningkatkan produktivitas[4]. Praktik *preventive maintenance* yang konsisten terbukti secara empiris mampu menurunkan waktu henti mesin secara signifikan[5]. Selaras dengan temuan tersebut, transisi dari strategi *corrective* menuju *preventive maintenance* dinilai jauh lebih efektif dalam mendongkrak keandalan peralatan dibandingkan mempertahankan strategi perawatan reaktif[3]. Di lingkup industri manufaktur Indonesia, implementasi metode *Usage-Based Maintenance* terbukti mampu mereduksi *downtime* secara optimal pada komponen kritis[6], sedangkan metode *Age Replacement* berhasil dimanfaatkan untuk menjadwalkan penggantian komponen secara tepat waktu sekaligus menghemat pengeluaran biaya

operasional akibat kegagalan mendadak[7].

Meskipun urgensi standardisasi pemeliharaan preventif telah ditekankan, implementasi di lapangan mutakhir ini masih kerap menghadapi kendala operasional. Ketergantungan pada proses identifikasi tindakan pemeliharaan yang bersifat konvensional atau manual sering kali memicu tingginya subjektivitas pelabelan, inkonsistensi pencatatan data log teknis, serta kelambatan respons dalam pengambilan keputusan strategis [8]. Keterbatasan mendasar dari proses manual ini tidak hanya memperpanjang birokrasi penanganan gangguan, tetapi juga berpotensi menurunkan produktivitas perusahaan secara drastis akibat kegagalan deteksi kerusakan perangkat secara *real-time*[5].

Efektivitas pemodelan klasifikasi telah banyak dibuktikan melalui sebagian besar literatur yang mengasumsikan data log parameter yang digunakan sudah ideal dan terbebas dari bias kualitatif teks bebas di lapangan. Penelitian ini mengisi celah tersebut dengan berfokus pada studi kasus pemantauan inventarisasi perangkat IT proyek X-System di PT XYZ dengan *dataset* operasional riil sebanyak 15.204 baris dan melibatkan lima fitur operasional (*brand*, kategori perangkat, umur perangkat, waktu tunggu, dan keterangan teknis), pemodelan ini dioptimalisasi menggunakan metode SMOTE untuk mitigasi ketidakseimbangan kelas. Secara spesifik, analisis perbandingan performa dalam penelitian ini bertujuan menginvestigasi dampak dominasi bias teks lapangan pada fitur 'keterangan' yang berpotensi memicu tingkat *overfitting* semu. Melalui audit kebocoran data tersebut, penelitian ini diharapkan mampu merumuskan arsitektur klasifikasi berbasis atribut perangkat keras murni yang lebih tangguh, interpretatif, serta memiliki kemampuan generalisasi yang valid terhadap data operasional baru di masa depan.

TINJAUAN LITERATUR

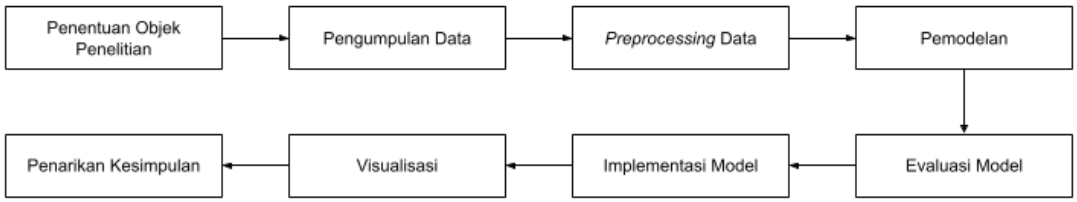
Sebagai upaya mentransformasi metodologi konvensional yang rentan terhadap kesalahan *human error,* integrasi pendekatan *data science* berbasis *machine learning* membuka peluang eksponensial dalam membangun sistem klasifikasi otomatis yang objektif dan adaptif. Dalam ranah pemodelan prediktif tabular, algoritma Random Forest menawarkan stabilitas tinggi melalui pendekatan *ensemble learning* berbasis pohon keputusan yang kokoh terhadap risiko *overfitting*[9]. Disisi lain, algoritma XGBoost Classifier memberikan keunggulan komparatif dari segi efisiensi komputasi, kecepatan eksekusi, serta fleksibilitas pemetaan batas keputusan[10]. Sebagai langkah untuk mengoptimalisasi karakteristik data operasional yang tidak seimbang, metode *Synthetic Minority Over-sampling Technique* (SMOTE) diterapkan guna meningkatkan representasi data kelas minoritas secara sintetis sebelum tahapan pelatihan model berjalan[11]

Eksplorasi terhadap kapabilitas arsitektur *machine learning* dalam optimasi strategi pemeliharaan telah dikaji secara luas oleh beberapa penelitian terdahulu. Evaluasi komparatif terhadap algoritma SVR, Decision Tree, Random Forest, dan XGBoost menunjukkan bahwa XGBoost terbukti memiliki efisiensi waktu pemrosesan tercepat[12]. Efektivitas XGBoost dalam lingkungan distribusi digital berskala besar juga telah divalidasi, di mana integrasi algoritma tersebut bersama arsitektur data *time series* berhasil menurunkan waktu henti operasional secara signifikan[13]. Keandalan arsitektur *boosting* dalam memprediksi potensi kerusakan mesin secara akurat guna mendukung pengambilan keputusan perawatan preventif juga telah dikonfirmasi secara empiris[14]. Sementara itu, evaluasi performa pada kondisi ketidakseimbangan data tabular membuktikan bahwa XGBoost mampu mempertahankan stabilitas prediksi meskipun dihadapkan pada rasio kelas yang timpang[15]. Melengkapi ketangguhan arsitektur *boosting* tersebut, keandalan arsitektur murni berbasis pohon (*tree-based models*) untuk deteksi gangguan infrastruktur juga telah dibuktikan dengan keberhasilan mencapai akurasi hingga 100% dalam klasifikasi level kondisi komponen industri[16]. Keunggulan model *ensemble* ini juga dikonfirmasi melalui *Systematic Literature Review* secara komprehensif, di mana model berbasis pohon keputusan secara konsisten unggul dibandingkan algoritma tunggal pada berbagai domain klasifikasi tabular, sebuah pola yang relevan untuk mendasari pemilihan Random Forest dan XGBoost sebagai algoritma utama klasifikasi maintenance TI pada penelitian[17]. Lebih lanjut, penerapan metode seleksi fitur menggunakan Random Forest terbukti berhasil memetakan dan mengisolasi atribut operasional yang paling berpengaruh secara akurat[18].

Aspek lain yang krusial untuk melandasi kontribusi utama artikel ini adalah konsep *data leakage* dalam *machine learning*. Secara definitif, *data leakage* merujuk pada masuknya informasi terkait variabel target ke dalam proses pelatihan model yang seharusnya tidak tersedia secara sah pada saat prediksi dilakukan, sehingga performa model yang terukur pada data uji menjadi terlalu optimistis dan tidak merepresentasikan kemampuan generalisasi yang sesungguhnya[19]. Bentuk kebocoran ini kerap muncul secara implisit, misalnya ketika fitur prediktor secara tidak sengaja merekam jejak proses yang terjadi setelah label target ditetapkan, sebagaimana ditemukan pada ratusan studi ilmiah berbasis *machine learning* yang tersebar di 17 bidang ilmiah berbeda, mulai dari kedokteran hingga ilmu sosial komputasional[19]. Dalam konteks klasifikasi *maintenance* perangkat TI, risiko ini menjadi relevan ketika fitur tekstual yang dicatat oleh teknisi berpotensi merekam ulang kondisi yang sama dengan yang digunakan untuk menentukan label PM/CM, sehingga model berisiko "menghafal" pola pelabelan alih-alih mempelajari hubungan kausal antara kondisi fisik perangkat dan kebutuhan pemeliharaan. Aspek inilah yang menjadi landasan konseptual bagi desain eksperimen dua skenario komparatif.

METODE PENELITIAN

Penelitian menggunakan pendekatan kuantitatif terapan yang berfokus pada pengembangan model prediktif berbasis *machine learning* untuk mengklasifikasikan jenis pemeliharaan perangkat TI. Ruang lingkup dan objek penelitian dibatasi pada pemantauan inventarisasi *hardware* proyek X-System di PT XYZ, dengan fokus operasional wilayah Zona 3. Guna memastikan pemecahan masalah dilakukan secara sistematis dan terukur, tahapan pelaksanaan penelitian dirancang ke dalam serangkaian alur kerja (*pipeline*) metodologi.



Gambar 1. Alur Diagram Metode Penelitian

1. Pengumpulan Data

   Data yang digunakan dalam penelitian ini didapatkan dari dokumen pengadaan perangkat dan laporan catatan lapangan mengenai kondisi perangkat saat dilakukan pengecekan *maintenance* yang diperoleh dari tim teknis proyek X-System PT XYZ di wilayah Zona 3.

   Data ini fokus pada data lapangan wilayah Zona 3 (Banjarmasin, Makassar, Manado, dan Jayapura) dengan total data siap pakai sebanyak 15.204 baris dan 22 kolom parameter. Berikut rincian data *maintenance* perangkat TI PT XYZ di wilayah Zona 3 dalam Tabel 1.

Tabel 1. *Dataset Maintenance* Perangkat TI PT XYZ di Wilayah Zona 3

no	kanwil	kanca	unit_kerja	...	status_ evaluasi	gap_ kuartal	klasifikasi _manual
0	Banjarmasin	Kanwil Banjarmasin	Kanwil Banjarmasin	...	Terpenuhi Tepat Waktu	Kuartal 1 Bulan 2 (from Kuartal 1 Bulan 2)	PM
1	Banjarmasin	Kanwil Banjarmasin	Kanwil Banjarmasin	...	Terpenuhi - Overdue	Kuartal 4 Bulan 2 (from Kuartal 4 Bulan 1)	PM
2	Banjarmasin	Kanwil Banjarmasin	Kanwil Banjarmasin	...	Terpenuhi - Overdue	Kuartal 3 Bulan 3 (from Kuartal 3 Bulan 2)	PM
3	Banjarmasin	Kanwil	Kanwil	...	Terpenuhi	Kuartal 2 Bulan 3	PM

		Banjarmasin	Banjarmasin		Tepat Waktu	(from Kuartal 2 Bulan 3)	
4	Banjarmasin	Kanwil Banjarmasin	Kanwil Banjarmasin	...	Terpenuhi Tepat Waktu	Kuartal 2 Bulan 3 (from Kuartal 2 Bulan 3)	PM
..	...	...	...	...	...	...	...
15202	MANADO	KC TOLI TOLI	KC TOLI TOLI	...	Terpenuhi Tepat Waktu	Kuartal 1 Bulan 3 (from Kuartal 1 Bulan 2)	CM

Ruang fitur prediktor (x) mengisolasi lima parameter utama: *brand*, kategori perangkat, umur perangkat, waktu tunggu, dan keterangan teknis lapangan. Variabel target kategorikal (y) mengklasifikasikan setiap entri data ke dalam dua jenis pemeliharaan: *Preventive Maintenance* (PM) atau *Corrective Maintenance* (CM). Berikut rangkuman karakteristik parameter operasional yang digunakan disajikan pada Tabel 2.

Tabel 2. Karakteristik Jenis Variabel Parameter *Dataset*

Nama Fitur	Jenis Variabel	Deskripsi	Contoh Data
brand	Kategorikal/Nominal	Merek produsen dari perangkat TI	HP, Lenovo, ASUS, Epson
kategori_perangkat	Kategorikal/Nominal	Jenis pengelompokkan *hardware*	PC, Printer, Scanner, Laptop
umur_perangkat	Numerikal/Diskrit	Lama masa pakai perangkat di lapangan (tahun)	1,3,5,8
waktu_tunggu	Numerikal/Diskrit	Durasi respon pemeliharaan operasional (hari)	1,2,3,4,5
keterangan	Tekstual/Kategorikal	Log catatan kondisi teknis riil lapangan oleh teknisi	"Layar bergaris", "Rusak"
klasifikasi_manual	Kategorikal/Nominal	Variabel target/*Output* kelas klasifikasi	PM, CM

2. *Preprocessing* Data

2.1. *Feature Selection*

Untuk mereduksi dimensi data dan menghindari *noise*, dilakukan seleksi fitur dengan mengisolasi 5 kolom parameter utama yang memiliki relevansi operasional paling kuat sebagai variabel prediktor (x), yaitu: *brand, kategori_perangkat, umur_perangkat, waktu_tunggu,* dan *keterangan*. Setelah dilakukan proses eliminasi data duplikat dan baris kosong pada fitur-fitur terpilih tersebut, diperoleh total 15.204 baris data yang siap diproses lebih lanjut.

2.2. *Data Labelling*

Dataset mentah hasil ekstraksi dari sistem PT XYZ belum memiliki label kelas biner yang bersih. Oleh karena itu, pelabelan variabel target (y) dibentuk secara terprogram melalui pendekatan berbasis aturan (*rule-based filtering*) terhadap log historis aktivitas perawatan menjadi variabel *klasifikasi_manual*. Log dipilah ke dalam dua kelas kategori:

a. *Preventive Maintenance* (PM): Ditetapkan untuk aktivitas pengecekan rutin berkala, perawatan sistem terjadwal, dan optimalisasi perangkat sebelum terjadinya kerusakan. Perangkat TI diklasifikasikan sebagai PM jika umur perangkat ≤ 5 tahun dan keterangan teknisi yang tidak menyebutkan hal terkait kerusakan atau penggantian.

b. *Corrective Maintenance* (CM): Ditetapkan untuk tindakan reaktif yang dipicu oleh laporan keluhan kendala teknis, kerusakan komponen, perangkat mati total (*breakdown*), atau penggantian suku cadang pasca-insiden. Perangkat TI diklasifikasikan sebagai CM jika umur perangkat >5 tahun dan keterangan kondisi perangkat "Tidak Digunakan", "Di Gudang", "Tidak Ada di Lokasi", "Rusak", "Tidak Nyala."

Perlu ditekankan bahwa skema pelabelan pada tahap ini memiliki implikasi metodologis yang lebih mendasar dibandingkan *data leakage* konvensional. Aturan pelabelan CM secara eksplisit

menyertakan kemunculan kata kunci tertentu (misalnya "Rusak", "Tidak Nyala") pada fitur *keterangan* sebagai salah satu syarat penentu label. Dengan kata lain, fitur *keterangan* bukan sekadar berkorelasi dengan label *klasifikasi_manual*, melainkan menjadi bagian definitif dari aturan yang membentuk label itu sendiri. Akibatnya, model yang dilatih dengan menyertakan fitur *keterangan* (Skenario A) pada dasarnya mempelajari ulang aturan pelabelan yang telah di-*hardcode*, bukan mempelajari hubungan kausal antara kondisi fisik perangkat dan kebutuhan pemeliharaan. Kondisi ini melampaui definisi *data leakage* yang umum dipahami dalam literatur [19]. Secara lebih spesifik, kondisi ini berkesesuaian dengan kategori yang belakangan diformalkan dalam literatur sebagai *feature-to-target leakage*, yaitu kondisi ketika label target dikonstruksi langsung dari fitur yang sama dengan yang digunakan untuk prediksi [20], sebuah bentuk kebocoran yang lebih parah dibandingkan leakage konvensional karena model tidak sekadar berkorelasi dengan informasi yang bocor, melainkan dilatih untuk mereproduksi ulang aturan pelabelan yang telah di-*hardcode*. Oleh karena itu, penelitian ini secara eksplisit memisahkan Skenario A (dengan fitur *keterangan*) sebagai ilustrasi kegagalan tersebut dan Skenario B (tanpa fitur *keterangan*) sebagai arsitektur yang merepresentasikan kondisi prediksi operasional yang sesungguhnya.

2.3. Anonimisasi dan Pembersihan Data

Kolom informasi internal perusahaan yang bersifat sensitif dan tidak memiliki korelasi dengan performa perangkat (nama pengguna perangkat, jabatan, dan nomor personal karyawan) dihapus secara eksplisit guna menegakkan privasi data korporasi. Selain itu, dilakukan standardisasi penulisan kapitalisasi teks (*case folding*) untuk menghindari duplikasi makna pada fitur tekstual. Pada basis data mentah sebelum diproses, kolom umur perangkat sempat memiliki variasi penulisan yang tidak konsisten pada sebagian entri (misalnya "<5th", "<5 TAHUN", dan ">5 tahun"). Untuk menghindari ambiguitas sekaligus menjamin konsistensi numerik secara menyeluruh, variabel umur perangkat pada dataset final tidak diperoleh melalui konversi tekstual, melainkan dihitung secara langsung dan sistematis dari kolom tahun pengadaan yang bersifat numerik dan tersedia untuk seluruh entri, yaitu umur perangkat=tahun referensi pengumpulan data (2025)−tahun pengadaan. Pendekatan berbasis perhitungan ini menghasilkan variabel umur perangkat dengan rentang nilai riil 0 hingga 21 tahun (median=7 tahun, rata-rata=6,9 tahun, simpangan baku=2,49 tahun berdasarkan 15.204 baris data), sehingga terhindar dari potensi bias akibat pembulatan atau pembentukan kategori (bucketing) nilai secara arbiter.

2.4. *Data Splitting*

Pada tahap ini, *dataset* yang telah dibersihkan dipisahkan menjadi fitur prediktor (x) dan variabel target (y) yang direpresentasikan oleh kolom *klasifikasi_manual* (terdiri dari 13.219 baris kelas PM dan 1.985 baris kelas CM). *Dataset* berlabel yang berjumlah 15.204 baris ini dipisah secara acak menggunakan proporsi rasio 80% sebagai data latih (*training set*) untuk proses penyesuaian bobot algoritma dan 20% sebagai data uji (*testing set*) yang diisolasi khusus untuk pengujian performa akhir yang objektif.

2.5. *Data Imbalance Mitigation*

Berdasarkan analisis distribusi pada variabel target (y) *klasifikasi_manual*, ditemukan ketimpangan kelas yang sangat mencolok. PM mendominasi sebesar 86,9% dan CM hanya sebesar 13,1%. Untuk mengantisipasi bias komputasi dengan model cenderung condong ke kelas mayoritas secara berlebihan, metode SMOTE (*Synthetic Minority Over-sampling Technique*) diintegrasikan secara langsung ke dalam arsitektur *pipeline* pemodelan. SMOTE dikonfigurasi untuk bekerja secara otomatis hanya pada fase *training* di setiap arsitektur model *Random Forest* dan XGBoost dan menyamakan proporsi kelas minoritas (CM) terhadap kelas mayoritas (PM) menjadi rasio seimbang (50:50) sebelum pohon keputusan terbentuk. Sebaliknya, data uji (*testing set*) pada *pipeline* diisolasi penuh dari intervensi SMOTE agar evaluasi metrik performa akhir tetap mencerminkan karakteristik data riil di lapangan.

3. Pemodelan *Machine Learning*

3.1. Random Forest

Random Forest adalah algoritma *ensemble learning* berbasis pohon keputusan yang efektif mereduksi risiko *overfitting* guna menghasilkan prediksi yang jauh lebih stabil dan akurat, terutama dalam memproses data berskala besar dengan variabel yang kompleks. Adapun rumus Random Forest dalam Persamaan (1)[21].

$$Gini(S_i) = 1 - \sum_{i=0}^{c-1} p_i^2 \quad \text{..... (1)}$$

Indeks Gini digunakan sebagai kriteria penentu dalam pemilihan fitur terbaik di setiap simpul internal. Kualitas pemisahan (*split*) pada fitur ke-*k* dievaluasi dengan meninjau distribusi jumlah sampel dari setiap kelas i (i = 1, . . . , c) yang terbentuk dalam subset hasil pemisahan. Dengan mengukur kemurnian data melalui Indeks Gini, algoritma dapat menentukan fitur yang memberikan pemisahan paling optimal. Nilai Indeks Gini yang lebih rendah mengindikasikan tingkat kemurnian subset yang lebih tinggi, yang berarti pembagian data pada titik tersebut lebih efektif.

3.2. XGBoost

XGBoost (*Extreme Gradient Boosting*) adalah algoritma *ensemble* iteratif yang meminimalkan tingkat eror dengan memetakan titik pemisah (*split point*) terbaik guna menargetkan sisa galat (residu) dari pohon-pohon sebelumnya. XGBoost meningkatkan akurasi prediksi model *ensemble* melalui optimasi fungsi kerugian (*loss function*). Penentuan arah dan besaran pembaruan bobot dilakukan dengan memanfaatkan *Gradien* (turunan pertama) dan matriks *Hessian* (turunan kedua), sebagaimana dirumuskan pada Persamaan (2) dan (3)[10].

$$g_i = \frac{\partial L}{\partial p_i} = p_i - y_i \text{.....(2)}$$

$$h_i = \frac{\partial^2 L}{\partial p_i} = p_i(1 - p_i) \text{.....(3)}$$

Variabel g_i dalam persamaan ini merepresentasikan *Gradien*, yaitu besaran sisa *error* pada setiap sampel data. Komponen h_i berperan sebagai *Hessian* yang mengukur kelengkungan (*curvature*) fungsi kerugian untuk memandu proses optimasi. Sementara itu, p_i menyatakan nilai probabilitas prediksi, dan *L* melambangkan fungsi kerugian yang menjadi acuan utama dalam tahapan optimasi.

4. Skenario Eksperimen Pemodelan

Pemodelan *machine learning* untuk klasifikasi PM dan CM hanya menggunakan 5 kolom dari 22 kolom pada data *maintenance* yang digunakan sebagai variabel dengan fitur prediktor (x) yakni kolom *brand*, *kategori_perangkat, umur_perangkat, waktu_tunggu, keterangan*, dan variabel target (y) yang direpresentasikan oleh kolom *klasifikasi_manual* sebagai label klasifikasi. Untuk mendukung proses klasifikasi jenis *maintenance* perangkat, digunakan pendekatan *supervised learning* dengan dua algoritma, yaitu Random Forest dan XGBoost Classifier. Pemodelan dilakukan dalam dua skenario, yaitu:

4.1 Pemodelan Skenario A

Skenario dengan fitur keterangan teknisi, menggunakan variabel dari *brand, kategori_perangkat, umur_perangkat, waktu_tunggu,* dan *keterangan*,

4.2 Pemodelan Skenario B

Skenario tanpa fitur keterangan teknisi, menggunakan variabel dari *brand, kategori_perangkat, umur_perangkat,* dan *waktu_tunggu*.

Sehubungan dengan konfigurasi model, penelitian ini secara eksplisit menggunakan hyperparameter default dari pustaka yang digunakan tanpa proses *tunning* tambahan (GridSearchCV maupun RandomizedSearchCV tidak diterapkan). Model Random Forest diinisialisasi melalui RandomForestClassifier() dari Scikit-Learn dengan seluruh parameter pada nilai *default* (di antaranya n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2), kecuali parameter random_state=42 yang ditetapkan untuk menjamin reproduktibilitas hasil. Model XGBoost diinisialisasi melalui XGBClassifier() dari pustaka XGBoost dengan seluruh parameter pada nilai *default* (di antaranya n_estimators=100, max_depth=6, learning_rate=0,3), kecuali parameter eval_metric='logloss' yang ditetapkan secara eksplisit untuk fungsi evaluasi pelatihan. Penggunaan hyperparameter default ini

merupakan keterbatasan metodologis yang diakui; penelitian lanjutan disarankan untuk mengeksplorasi ruang pencarian hyperparameter guna mengoptimalkan performa kedua model.

5. Analisis dan Evaluasi Model

Evaluasi ini dilakukan untuk mengukur kemampuan model mengklasifikasikan perangkat TI ke dalam dua jenis pemeliharaan PM dan CM. Performa model diukur menggunakan matriks evaluasi *Confusion Matrix*. Matriks ini tersusun atas empat elemen dasar, yaitu *True Positive* (TP) yakni kondisi ketika model memprediksi kelas positif dengan benar; *True Negative* (TN) yakni kondisi ketika model memprediksi kelas negatif dengan benar; *False Positive* (FP) yakni kondisi ketika model keliru memprediksi data aktual negatif sebagai kelas positif; serta *False Negative* (FN) yakni kondisi ketika model keliru memprediksi data aktual positif sebagai kelas negatif. Keempat elemen tersebut digunakan untuk menghitung berbagai metrik evaluasi, sebagai berikut [22].

5.1. *Accuracy*

Akurasi merupakan metrik evaluasi fundamental yang mengukur persentase ketepatan prediksi model secara keseluruhan terhadap total data uji. Rumus disajikan pada Persamaan (4).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{.....(4)}$$

5.2. *Precision*

Presisi mengukur rasio ketepatan prediksi pada kelas positif, yakni merepresentasikan seberapa banyak data yang benar-benar positif dari keseluruhan data yang ditebak positif oleh model. Rumus disajikan pada Persamaan (5).

$$Precision = \frac{TP}{TP+FP} \quad \text{.....(5)}$$

5.3. *Recall*

Recall (sensitivitas) mengukur kemampuan atau kepekaan model dalam mendeteksi dan menemukan kembali seluruh data yang aktualnya memang bernilai positif. Rumus disajikan pada Persamaan (6).

$$Recall = \frac{TP}{TP+FN} \quad \text{.....(6)}$$

5.4. *F1-Score*

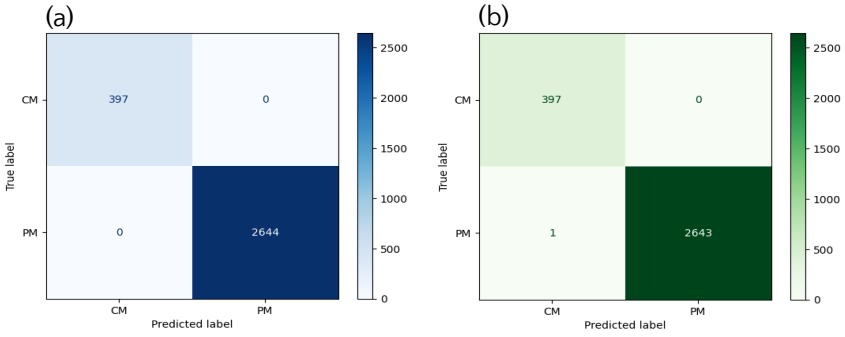
F1-score adalah metrik yang merepresentasikan nilai rata-rata harmonik antara presisi dan *recall*, yang sangat ideal digunakan untuk menyeimbangkan evaluasi performa pada *dataset* dengan distribusi kelas yang timpang. Rumus disajikan pada Persamaan (7).

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \text{.....(7)}$$

HASIL PENELITIAN

1. Evaluasi Performa Model Skenario A

Hasil eksperimen pada Skenario A menunjukkan tingkat akurasi yang luar biasa tinggi pada kedua algoritma. Klasifikasi menggunakan Random Forest menghasilkan akurasi sempurna sebesar 100% (1,00). Performa ini diikuti secara ketat oleh XGBoost *Classifier* yang menghasilkan akurasi sebesar 99,97% (0,9997). Melalui representasi *Confusion Matrix*, terlihat bahwa inklusi data log deskripsi dari teknisi memudahkan algoritma dalam memisahkan batas keputusan antara kelas PM dan CM secara mutlak dengan tingkat kesalahan prediksi yang mendekati nol.

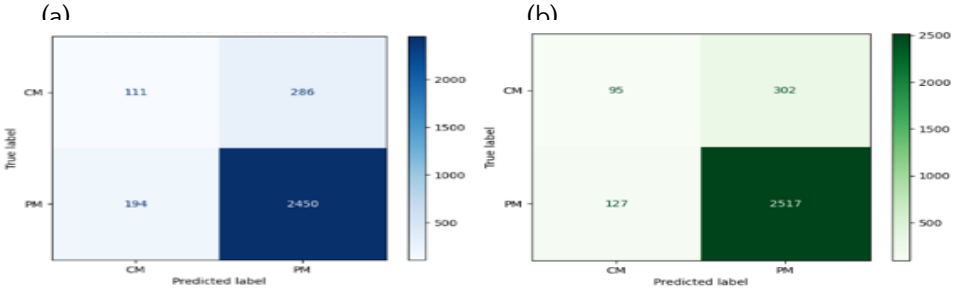


Gambar 2. *Confusion Matrix* PM dan CM Skenario A Random Forest (a), XGBoost (b)

Secara makro, visualisasi *Confusion Matrix* pada Skenario A dalam Gambar 2 memperlihatkan pemisahan batas keputusan (*decision boundary*) yang sangat mutlak dan terlampau sempurna. Model arsitektur Random Forest berhasil mengisolasi 397 sampel data uji aktual kelas CM dan 2.644 sampel aktual kelas PM secara absolut tanpa ada satupun kesalahan klasifikasi silang (*zero misclassification*). Tren performa yang nyaris identik ditunjukkan oleh algoritma XGBoost yang mampu memetakan seluruh 397 sampel kelas CM secara tepat dan hanya mencatatkan 1 margin kesalahan pada prediksi kelas PM (keliru memprediksi 1 kelas aktual PM sebagai CM).

2. Evaluasi Performa Model Skenario B

Eksperimen pada Skenario B dilakukan dengan mengeksklusi atribut teks log teknisi guna menyimulasikan kondisi prediksi operasional yang sesungguhnya di lapangan. Tanpa adanya fitur yang memicu *data leakage* tersebut, hasil pengujian menunjukkan koreksi performa yang cukup signifikan pada kedua algoritma. Akurasi model kembali pada tingkat yang lebih realistis dan terbebas dari anomali *overfitting*. Melalui representasi *Confusion Matrix*, terlihat bahwa hilangnya variabel teks deterministik membuat algoritma tidak lagi mampu memisahkan batas keputusan secara absolut sehingga mulai muncul persentase kesalahan klasifikasi silang (*misclassification*) pada pemetaan kelas PM maupun CM.



Gambar 3. *Confusion Matrix* PM dan CM Skenario B Random Forest (a), XGBoost (b)

Secara makro, visualisasi *Confusion Matrix* pada Skenario B dalam Gambar 3 menyoroti tantangan utama model dalam mengisolasi kelas minoritas, CM. Penurunan performa paling kritis terdeteksi pada nilai sensitivitas (*recall*) model terhadap kelas tersebut. Pada Random Forest, model hanya mampu memprediksi 111 kasus CM dengan benar dan keliru mengklasifikasikan 286 kasus lainnya sebagai PM (ekuivalen dengan kegagalan mendeteksi ~72% kasus CM aktual). Tren serupa dialami oleh XGBoost, algoritma ini hanya berhasil memetakan 95 kasus CM secara tepat dan mengalami misklasifikasi pada 302 baris data aktual CM (gagal mengenali ~76% kasus riil). Fenomena ini mengindikasikan bahwa ketika pemodelan murni bergantung pada parameter fisik perangkat keras, algoritma *machine learning* cenderung mengalami bias mayoritas dan kesulitan menangkap pola spesifik dari kerusakan perangkat yang bersifat korektif.

3. Evaluasi Performa Komparatif Model

Rangkuman hasil perbandingan performa komparatif seluruh skenario disajikan secara terintegrasi pada Tabel 3.

Tabel 3. Evaluasi Keseluruhan Performa Komparatif Model

Model	Algoritma	Kelas	Sampel	Prediksi Benar	Prediksi Salah	Prec.	Recall	F1-Score	Acc.
Skenario A	Random Forest	CM	397	397	0	1,00	1,00	1,00	1,00
		PM	2644	2644	0	1,00	1,00	1,00	
	XGBoost	CM	397	397	0	0,99	1,00	0,99	0,9997
		PM	2644	2643	1	1,00	0,99	0,99	
Skenario B	Random Forest	CM	397	111	286	0,36	0,28	0,32	0,8421
		PM	2644	2450	194	0,90	0,93	0,91	
	XGBoost	CM	397	95	302	0,43	0,24	0,31	0,8589

		PM	2644	2517	127	0,89	0,95	0,92	

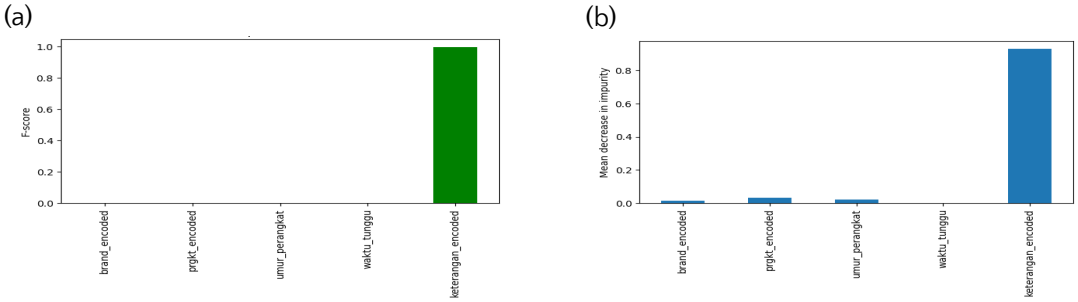
Berdasarkan Tabel 3, pengujian Skenario A menghasilkan metrik performa semu yang nyaris sempurna dengan akurasi 100% pada arsitektur *Random Forest* dan 99,97% pada *XGBoost*. Tingkat sensitivitas (*recall*) yang mencapai angka sempurna pada kedua model ini mengindikasikan bahwa pola *string* kualitatif yang terekam pada log deskripsi teknisi memiliki korelasi linear yang sangat deterministik terhadap variabel target. Fenomena ini membuktikan bahwa arsitektur berbasis pohon (*tree-based models*) tidak belajar dari parameter fisik operasional perangkat, melainkan mengeksploitasi fitur teks tersebut layaknya "kunci jawaban" langsung, yang secara nyata memvalidasi terjadinya kebocoran data (*data leakage*) secara struktural. Sebaliknya, pada Skenario B yang merepresentasikan kondisi operasional aktual, performa prediksi model terkoreksi menjadi lebih valid dengan tingkat akurasi global berada di angka 84,21% untuk Random Forest dan 85,89% untuk XGBoost.

Secara keseluruhan, XGBoost terbukti sedikit lebih unggul dalam akurasi global. Namun, jika meninjau metrik *F1-Score* sebagai penyeimbang ketimpangan kelas, kedua algoritma terlihat menghadapi tantangan berat dalam mengklasifikasikan kelas minoritas (CM). Hal ini dibuktikan dengan anjloknya nilai *recall* kelas CM secara drastis dari angka sempurna, 1,00, menjadi hanya 0,28 pada Random Forest dan 0,24 pada XGBoost. Penurunan signifikan ini menegaskan bahwa tanpa adanya intervensi teks deskripsi teknisi, fitur perangkat keras operasional belum cukup memadai untuk memetakan pola kerusakan korektif secara optimal.

Evaluasi pada penelitian ini menggunakan satu kali pembagian data latih dan data uji (*single train-test split* 80:20). Sebagai pemeriksaan tambahan, dilakukan pengujian 5-Fold Cross-Validation pada arsitektur Skenario A (Random Forest dengan fitur keterangan), yang menghasilkan rata-rata akurasi sebesar 1,00 pada seluruh lipatan (*fold*) — temuan ini semakin memperkuat bukti bahwa performa sempurna pada Skenario A bersifat struktural akibat kebocoran data, bukan artefak dari satu kali pembagian data uji tertentu. Pengujian setara juga dilakukan untuk Skenario B menggunakan 5-Fold Stratified Cross-Validation pada kedua algoritma, dengan SMOTE diterapkan secara eksklusif pada setiap fold data latih agar data uji setiap lipatan tetap bebas dari intervensi *resampling*. Hasilnya, Random Forest memperoleh rata-rata akurasi 83,75% (±0,56%), *precision* kelas CM 35,87% (±2,69%), *recall* kelas CM 31,23% (±3,23%), dan *F1-Score* kelas CM 33,37% (±2,94%) antarlipatan. Sementara itu, XGBoost memperoleh rata-rata akurasi 86,43% (±0,61%), *precision* kelas CM 45,50% (±5,29%), *recall* kelas CM 20,86% (±3,35%), dan *F1-Score* kelas CM 28,57% (±4,18%) antarlipatan. Simpangan baku yang relatif kecil pada seluruh metrik (di bawah 1% untuk akurasi) menunjukkan bahwa performa Skenario B yang dilaporkan pada Tabel 3 bersifat stabil dan bukan artefak dari satu kali pembagian data uji tertentu, sehingga memperkuat validitas klaim generalisasi model.

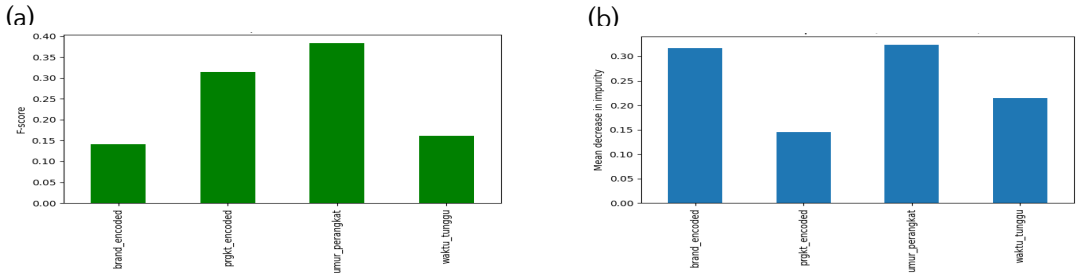
PEMBAHASAN

Kesenjangan performa yang masif antarskenario dianalisis lebih lanjut menggunakan ekstraksi nilai penting fitur (*Feature Importance*). Tahapan ini bertujuan untuk menginterpretasikan bagaimana model "mengambil keputusan" dengan cara memetakan kontribusi atau bobot masing-masing variabel prediktor terhadap hasil akhir klasifikasi.



Gambar 4. *Feature Importance* Skenario A Random Forest (a), XGBoost (b)

Berdasarkan nilai bobot variabel Skenario A pada Gambar 4, ditemukan struktur anomali berupa dominasi bias ekstrem. Pada arsitektur Random Forest, fitur *keterangan_encoded* menyerap bobot kepentingan hingga ~0,90, sementara model XGBoost mencatatkan nilai absolut sebesar 1,00. Angka ini secara teoretis memvalidasi terjadinya fenomena kebocoran data (*data leakage)*. Kebocoran ini terjadi karena kalimat deskriptif yang ditulis oleh teknisi di lapangan (misalnya kalimat "penggantian komponen rusak") di-*input* pasca-tindakan pemeliharaan selesai dieksekusi sehingga teks tersebut secara tidak langsung sudah mengandung representasi implisit dari label target (kunci jawaban).



Gambar 5. *Feature Importance* Skenario B Random Forest (a), XGBoost (b)

Sebaliknya, ketika fitur bias tersebut dieksklusi pada Skenario B, berdasarkan Gambar 5 distribusi bobot fitur bergeser menjadi lebih objektif dan natural. Variabel *umur_perangkat* secara konsisten mengemuka sebagai prediktor utama, dengan nilai kepentingan sebesar 0,32 pada Random Forest dalam dan 0,38 pada XGBoost. Posisi selanjutnya diikuti oleh *brand_encoded* sebesar 0,31 pada Random Forest dan *prgkt_encoded* 0,31 pada XGBoost. Hal ini mengonfirmasi bahwa tanpa intervensi teks keterangan, model memvalidasi faktor usia *hardware* sebagai indikator operasional paling krusial dalam menentukan keputusan pemeliharaan.

Melalui visualisasi ini, dapat dibuktikan secara ilmiah bahwa performa tinggi pada Skenario A merupakan bentuk *overfitting* semu. Oleh karena itu, arsitektur Skenario B ditetapkan sebagai model final operasional perusahaan karena merepresentasikan proses pembelajaran mesin yang valid, realistis terhadap data baru, serta terbebas dari bias retrospektif teks lapangan.

Dibandingkan dengan penelitian terdahulu yang menjadi rujukan literatur, temuan ini memberikan konteks penting mengenai realistisnya performa yang dilaporkan. Studi [16] melaporkan bahwa model berbasis pohon keputusan mampu mencapai akurasi hingga 100% dalam mengklasifikasikan level kondisi komponen industri, sebuah angka yang secara struktural mirip dengan hasil Skenario A pada penelitian ini sebelum audit kebocoran data dilakukan. Sebaliknya, studi [12] yang membandingkan SVR, Decision Tree Regression, Random Forest Regression, dan XGBoost pada kasus prediksi konsumsi bahan bakar kendaraan berat melaporkan bahwa performa XGBoost (MSE = 3,857) justru berada di bawah SVR (MSE = 0,26), mengindikasikan bahwa keunggulan algoritma *ensemble* tidak selalu konsisten antarkonteks tugas dan *dataset*. Dengan mempertimbangkan kedua temuan tersebut, akurasi 84,21% (Random Forest) dan 85,89% (XGBoost) yang diperoleh pada Skenario B penelitian ini berada pada rentang yang secara metodologis lebih dapat dipertanggungjawabkan dibandingkan klaim akurasi mendekati sempurna yang umum dilaporkan pada studi *maintenance* berbasis fitur tekstual yang tidak diaudit kebocorannya, sekaligus menegaskan kontribusi utama penelitian ini dalam mengidentifikasi dan mengoreksi bias tersebut.

Selain keunggulan metodologis tersebut, nilai *recall* kelas CM pada Skenario B yang masih rendah (0,28 untuk Random Forest dan 0,24 untuk XGBoost) merupakan keterbatasan signifikan yang perlu dicermati dari sisi implikasi operasional. Dalam konteks perencanaan *maintenance*, kegagalan mendeteksi kasus *corrective maintenance* yang sesungguhnya (*False Negative*) berpotensi menimbulkan konsekuensi biaya yang jauh lebih besar dibandingkan *False Positive*, karena keterlambatan penanganan kerusakan aktual dapat memicu *downtime* tak terjadwal. Oleh karena itu, penerapan model pada praktik operasional PT XYZ sebaiknya tidak hanya mengandalkan label kelas hasil prediksi baku, melainkan mempertimbangkan penyesuaian ambang keputusan (*threshold*) probabilitas ke arah yang lebih sensitif terhadap kelas CM, penerapan skema pembelajaran sensitif-biaya *(cost-sensitive learning*) yang membebankan penalti lebih besar pada kesalahan klasifikasi CM, atau kombinasi SMOTE dengan teknik penyeimbangan data lanjutan

seperti ADASYN maupun pendekatan *ensemble* bertingkat (*stacking*). Ketiga strategi tersebut berpotensi meningkatkan sensitivitas model terhadap kelas minoritas tanpa mengorbankan validitas evaluasi yang telah dicapai melalui eksklusi fitur 'keterangan'.

KESIMPULAN

Penelitian berhasil merancang struktur *Machine Learning Pipeline* terintegrasi berbasis algoritma SMOTE untuk klasifikasi otomatis jenis pemeliharaan perangkat IT pada proyek X-System di PT XYZ. Hasil analisis komparatif menemukan adanya anomali *data leakage* ekstrem akibat pelibatan fitur kualitatif teks log lapangan ('keterangan') yang memicu akurasi semu hingga 100%. Eksklusi fitur tersebut pada model Skenario B mengembalikan kapabilitas sistem ke tingkat akurasi yang lebih realistis, yakni 84,21% pada *Random Forest* dan 85,89% pada XGBoost. Arsitektur Skenario B direkomendasikan sebagai luaran akhir operasional karena terbukti lebih objektif, tangguh (*robust*), serta memiliki kemampuan generalisasi yang valid terhadap sebaran data operasional di masa depan.

Beberapa keterbatasan penelitian ini perlu dikemukakan secara eksplisit. Pertama, validasi model hanya dilakukan pada data operasional wilayah Zona 3 (Banjarmasin, Makassar, Manado, dan Jayapura), sehingga generalisasi hasil ke wilayah operasional PT XYZ lainnya maupun ke organisasi lain memerlukan pengujian lanjutan. Kedua, variabel target (klasifikasi_manual) dihasilkan melalui *rule-based filtering* terhadap log historis, bukan berasal dari label lapangan aktual yang diverifikasi langsung oleh teknisi maupun manajemen aset, sehingga performa model pada dasarnya mengukur konsistensi terhadap aturan pelabelan tersebut dan bukan terhadap kebenaran lapangan (ground truth) yang independen. Ketiga, meskipun eksklusi fitur 'keterangan' pada Skenario B berhasil menghilangkan kebocoran data, nilai recall kelas CM yang dihasilkan masih tergolong rendah (0,28 pada Random Forest dan 0,24 pada XGBoost), sehingga model pada kondisi saat ini belum cukup andal untuk menggantikan penilaian manual teknisi secara penuh dan lebih tepat diposisikan sebagai alat bantu pendukung keputusan (*decision support*) yang perlu digunakan bersama pertimbangan domain.

Untuk pengembangan riset selanjutnya, disarankan melakukan rekayasa fitur lanjutan atau *feature engineering* pada variabel teks bebas menggunakan metode *Natural Language Processing* (NLP) yang terstandardisasi atau ekstraksi berbasis *Deep Learning* seperti LSTM untuk membedakan anomali sebelum dan sesudah tindakan tanpa memicu kebocoran data. Selain itu, manajemen perusahaan disarankan menetapkan regulasi standardisasi penulisan log deskripsi guna meminimalkan subjektivitas input data masa depan.

REFERENCES

[1] K. N. . Dervitsiotis, *Operations Management*. McGraw-Hill, 1981.

[2] R. Keith. Mobley, *An introduction to predictive maintenance*. Butterworth-Heinemann, 2002.

[3] G. Ewin and M. Wilson, "Transitioning from Corrective to Preventive Maintenance Strategies: Enhancing Equipment Reliability and Efficiency," Mar. 2025.

[4] F. Mustaqim, W. Kosasih, and A. Ahnad, "Pemeliharaan Mesin Hydraulic Shear Menggunakan Pendekatan Reliability Centered Maintenance dan Manajemen Suku Cadang," *Jurnal Rekayasa Sistem Industri*, vol. 9, no. 3, pp. 153–162, Oct. 2020, doi: 10.26593/jrsi.v9i3.4023.153-162.

[5] J. O. Iwarue, "Impact of Preventive Maintenance Practices on Productivity," *British Journal of Management and Marketing Studies*, vol. 8, no. 1, pp. 118–137, Feb. 2025, doi: 10.52589/bjmms-qrmkxg9r.

[6] I. Ramadhan and R. Fitriani, "Optimalisasi Efektivitas Preventive Maintenance Berbasis Usage-Based Maintenance untuk Mengurangi Downtime di PT PQR," *Journal of Integrated System*, vol. 7, no. 2, pp. 166–183, Jan. 2025, doi: 10.28932/jis.v7i2.9760.

[7] M. I. Haq and D. Riandadari, "PENENTUAN PENJADWALAN PREVENTIVE MAINTENANCE PADA KOMPONEN MESIN CALLENDER DI PT. KARET NGAGEL SURABAYA WIRA JATIM," *JPTM*, vol. 09, no.

1, pp. 8–16, 2019.

[8] C. T. Nauli Siregar, P. Kindangen, and I. Debbie Palandeng, "EVALUATION OF MAINTENANCE ON PRODUCTION MACHINERY AND EQUIPMENT AT PT. MULTI NABATI SULAWESI BITUNG CITY," MNS, 2022.

[9] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: https://doi.org/10.1023/A:1010933404324.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[11] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," 2002.

[12] H. A. Parhusip, L. Lea, S. Trihandaru, D. B. Nugroho, P. P. Santosa, and A. H. Hariadi, "Learning Algorithms of SVR, DTR, RFR, and XGBoost (Case Study: Predictive Maintenance of Fuel Consumption)," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 14, no. 2, pp. 250–261, Jul. 2025, doi: 10.23887/janapati.v14i2.85657.

[13] N. B. Taş, A. Başgümüş, and M. Namdar, "Predictive Maintenance in E-Commerce Distribution Systems," *All Science Academy*, pp. 1–45, May 2025, doi: 10.5281/zenodo.15549040.

[14] G. Hema, S. Rani, B. Vyshnavi, K. Narendra, M. Yamini, and S. Chandana, "Strategic Predictive Maintenance Using XG Boost," *IJIRT 164243 INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY*, vol. 495, 2024.

[15] G. Velarde, A. Sudhir, S. Deshmane, A. Deshmunkh, K. Sharma, and V. Joshi, "Evaluating XGBoost for Balanced and Imbalanced Data: Application to Fraud Detection," *The Conference for the Era of AI and the Metaverse*, pp. 1–17, Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.15218

[16] M. Wu *et al.*, "An intelligent predictive maintenance system based on random forest for addressing industrial conveyor belt challenges," *Front. Mech. Eng.*, vol. 10, 2024, doi: 10.3389/fmech.2024.1383202.

[17] N. Nazifah, C. Prianto, and C. A. Id, "Analisis Perbandingan Decision Tree Algoritma C4.5 dengan algoritma lainnya: Sistematic Literature Review," *Jurnal Informatika dan Teknologi Komputer*, vol. 04, no. 02, pp. 57–64, Oct. 2023, doi: https://doi.org/10.55377/j-icom.v5i2.

[18] A. W. Santana, M. G. Arifyanto, and M. Ikbal, "Feature Selection for Predictive Asset Health Management in Power Plants Using Random Forest," *ITB Graduate School Conference*, vol. 4, no. 1, Jan. 2025, [Online]. Available: https://gcs.itb.ac.id/proceeding-igsc/index.php/igsc/article/view/224

[19] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, Sep. 2023, doi: 10.1016/J.PATTER.2023.100804.

[20] L. Sasse *et al.*, "Overview of leakage scenarios in supervised machine learning," *Journal of Big Data 2025 12:1*, vol. 12, no. 1, pp. 135-, May 2025, doi: 10.1186/S40537-025-01193-8.

[21] A. Sari, M. Arifin, and E. Darmanto, "PREDIKSI KEBUTUHAN STOK BARANG MENGGUNAKAN ALGORITMA RANDOM FOREST UNTUK MENINGKATKAN EFISIENSI PENJUALAN," *Journal Of Information Systems And Informatics Engineering)*, vol. 9, no. 2, pp. 339–351, 2025.

[22] K. M. Sujon, R. Hassan, K. Choi, and M. A. Samad, "Accuracy, precision, recall, f1-score, or MCC? empirical evidence from advanced statistics, ML, and XAI for evaluating business predictive models," *Journal of Big Data 2025 12:1*, vol. 12, no. 1, pp. 268-, Dec. 2025, doi: 10.1186/S40537-025-01313-4.