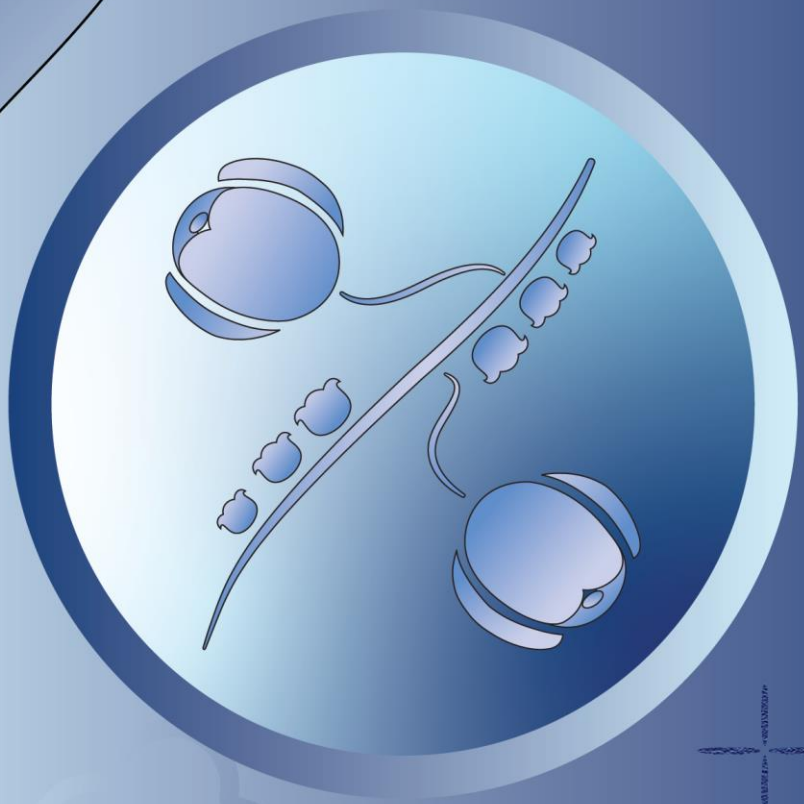
Ajang Pengenalan Statistika dan Festival Data #19

Makalah



DATAVERS

NAMA TIM

Inf-duo stg

NOMOR PESERTA

DVS0000343

1. PENDAHULUAN

A. Latar Belakang

Seiring berkembangnya waktu, perkembangan teknologi informasi terjadi secara signifikan, baik dalam lingkup domestik maupun internasional, kemajuan teknologi juga terus berkembang di berbagai bidang salah satunya di bidang industri dan jasa [1]. *Internet of Things* (IoT) memainkan peran penting dalam mengaktifkan kecerdasan buatan yang mengarah pada penciptaan produk dan layanan yang inovatif [2]. Perusahaan yang menerapkan pendekatan ini menghadapi persaingan yang semakin ketat dalam lingkungan pasar yang dinamis, dimana peningkatan kapasitas produksi saja tidak menjamin kesuksesan [3]. Meskipun terdapat berbagai macam tantangan industri, menambahkan ilmu pengetahuan ke dalam model yang dapat dipahami masih menjadi suatu tantangan [4]. Sistem pendukung keputusan yang memanfaatkan algoritma *machine learning* atau pembelajaran mesin dapat membantu dalam memfasilitasi pembuatan kebijakan yang efektif, sehingga jika terdapat suatu kerusakan dan kegagalan di industri terutama pada bagian mesin dapat berpotensi dilakukan pemulihan yang lebih cepat [5]. Selain itu, pemanfaatan data dalam jumlah besar (*big data*), terutama dalam memprediksi kegagalan mesin (*machine breakdowns*) memungkinkan industri untuk meningkatkan kinerja dan mengelola kebutuhan produk secara mandiri [6].

Dengan meningkatnya data yang dihasilkan dalam industri, algoritma *machine learning* memainkan peran penting dalam mengambil wawasan untuk pemahaman yang baik [7]. *Machine learning* dapat digunakan untuk mendiagnosis dan mengklasifikasikan masalah [8]. Dalam beberapa kasus, mesin terkadang menunjukkan tanda-tanda kerusakan dan gejala sebelum mengalami hal tersebut [9]. Penerapan prediktif penanganan gejala dan kerusakan dapat meningkatkan produktivitas, efisiensi sumber daya, dan pengurangan kesalahan sistem [10].

Terdapat penelitian terdahulu yang dilakukan terkait penerapan *machine learning* pada klasifikasi dan prediksi kerusakan mesin. Penelitian oleh [11] membahas tentang prediksi kesalahan mesin menggunakan *machine learning* dan *deep learning* dengan menghasilkan nilai *f1-score* tertinggi bernilai 0,977 menggunakan model *XGBoost* dan *random forest* dengan nilai 0,946. Pada

penelitian akan mengimplementasikan berbagai model *machine learning* untuk menyelesaikan masalah klasifikasi pada kegagalan mesin (*machine breakdowns*) dan menyelesaikan kompetisi DATAVERS ANAVA UGM 2025.

B. Rumusan Masalah

Rumusan masalah yang dapat diambil dari penelitian ini adalah:

1. Bagaimana penerapan model *machine learning* dalam melakukan klasifikasi kerusakan mesin?
2. Bagaimana performa tiap model *machine learning* dalam melakukan klasifikasi kerusakan mesin?

C. Pembatasan Masalah

Diperoleh batasan masalah sebagai berikut:

1. Data yang diambil adalah data kompetisi DATAVERS ANAVA UGM 2025 dengan sumber kaggle.com.
2. Analisis data menggunakan model machine learning *logistic regression, k-nearest neighbor, decision tree, random forest, xgboost,* dan *adaboost.*

D. Tujuan Penelitian

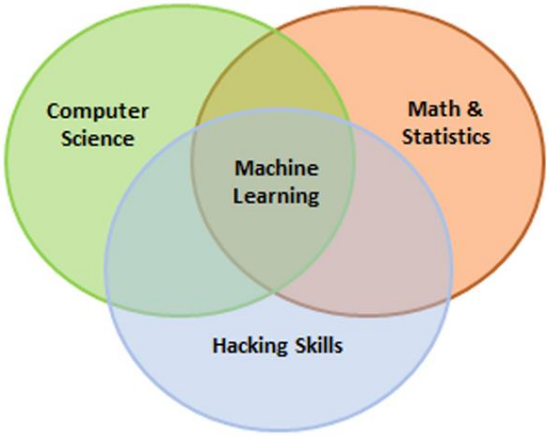
Berdasarkan rumusan masalah yang telah diuraikan, tujuan penelitian ini adalah:

1. Mengetahui penerapan model *machine learning* dalam melakukan klasifikasi kerusakan mesin.
2. Mengetahui performa tiap model *machine learning* dalam melakukan klasifikasi kerusakan mesin.

2. LANDASAN TEORI

A. *Machine Learning*

Machine learning adalah cabang ilmu komputer yang memanfaatkan data di masa lalu untuk belajar dan menggunakan pengetahuannya dalam membuat keputusan di masa depan [12]. *Machine learning* berada pada pertemuan antara ilmu komputer, teknik, statistika, dan matematika. Tujuan *machine learning* adalah untuk mengetahui pola yang dan menciptakan aturan yang belum diketahui dari contoh-contoh yang diberikan. *Machine learning* merupakan salah satu cabang ilmu kecerdasan buatan (*Artificial Intelligence*) yang menyelesaikan masalah klasifikasi, regresi, klastering, dan *anomaly detection* pada berbagai bidang sehingga dapat diatasi lebih efisien. Secara umum, *machine learning* diklasifikasikan ke dalam tiga kategori yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning* [12].



Gambar 1. Konsep *Machine Learning*

B. Klasifikasi

Klasifikasi dalam konsep *machine learning* adalah salah satu jenis *supervised learning* yang bertujuan untuk memprediksi label kategori dari data baru berdasarkan pola yang dipelajari dari data yang sudah dilabeli sebelumnya [12]. Dalam klasifikasi, model dilatih menggunakan dataset yang terdiri dari fitur *(input)* dan label *(output)* kategori, dengan tujuan untuk mengelompokkan data baru ke dalam salah satu kategori yang sudah ditentukan sebelumnya.

C. *Data Prepocessing*

Data preprocessing adalah langkah awal dalam analisis data atau pembangunan model *machine learning* yang bertujuan untuk mempersiapkan data mentah agar menjadi lebih bersih, terstruktur, dan siap digunakan oleh algoritma machine learning. Berikut merupakan tahapan data *prepocessing* dalam penelitian ini:

1. *Data Engineering*

   Data engineering adalah proses merancang, membangun, dan mengelola infrastruktur dan alur kerja *(pipelines)* yang memungkinkan pengumpulan, penyimpanan, pengolahan, dan analisis data dalam jumlah besar. *Data engineering* sangat penting dalam menyiapkan data yang bersih, terstruktur, dan dapat diakses untuk mendukung analisis dan penggunaan *machine learning*.

2. *Missing Value*

   Penanganan *missing value* adalah proses mengatasi nilai yang hilang dalam dataset, *missing value* dalam analisis ini menggunakan metode *complete case analysis (listwise deletion)*, di mana setiap pengamatan yang memiliki nilai hilang akan dihapus dari dataset.

$$m_{ij} = \begin{cases} 1 & jika\ x_{ij} tersedia (tidak\ hilang) \\ 0 & jika\ x_{ij}\ hilang \end{cases} \quad (1)$$

$$c_i = \begin{cases} 1 & jika \sum_{j=1}^{p} m_{ij} = p \\ 0 & jika \sum_{j=1}^{p} m_{ij} < p \end{cases} \quad (2)$$

3. *Outlier*

   Outlier adalah data atau observasi yang memiliki nilai yang jauh berbeda atau menyimpang secara signifikan dari sebagian besar data lainnya dalam dataset. *Outlier* dapat muncul dalam satu variabel *(univariate)* atau dalam hubungan antara beberapa variabel (*multivariate*). Penanganan *outlier* dilakukan pada kolom numerik menggunakan metode *winsorization*, di mana

nilai *outlier* diganti dengan *threshold* atas dan bawah berdasarkan visualisasi boxplot untuk setiap kolom. Misal $X = \{x_1, x_2, \dots, x_n\}$.

a. Menentukan batas (*threshold*)

$$IQR = Q_3 - Q_1$$
$$Lower\ bound\ (L) = Q_1 - 1{,}5 \times IQR \quad (3)$$
$$Upperbound\ (U) = Q_3 + 1{,}5 \times IQR$$

b. Fungsi *winsorization*

$$W_{(x)}: W_{(x)} - \{L, if\ x < Lx, if\ L \leq x \leq U\ U, if\ x > U\} \quad (4)$$

c. Setelah *winsorization*

$$W_{(x)}: W_{(x)} - \{L, if\ x < Lx, if\ L \leq x \leq U\ U, if\ x > U\} \quad (5)$$

4. *Undersampling-Oversampling*

Untuk mengatasi ketidakseimbangan kelas dalam klasifikasi multikelas, digunakan pendekatan *hybrid sampling* yang menggabungkan teknik *undersampling* dan *oversampling*. *Undersampling* akan diterapkan pada kelas mayoritas untuk mengurangi dominasinya, sementara *oversampling* akan dilakukan pada kelas-kelas minoritas untuk meningkatkan representasinya.

$$N_{new\_mayoritas} = \min\ (N_{mayoritas}, N_{minoritas} \quad (6)$$

$$N_{new\_minoritas} = \max\ (N_{minoritas} + N_{mayoritas}) \quad (7)$$

5. *Feature Selection*

Feature selection merupakan metode dalam memilih suatu fitur/variabel, sehingga dapat dilakukan untuk mengoptimalkan pemilihan kolom/fitur yang relevan. Metode yang digunakan adalah *Logistic Regression Multinomial* dengan *L1 regularization* (Lasso).

$$P(y = k|X) = \frac{\exp\ (w_k X)}{\Sigma_{j=1}^{C} \exp\ (w_j X)} \quad (8)$$

6. *Normalization Robust Scaling*

Normalization Robust Scaling adalah metode *preprocessing* data yang dirancang untuk membuat data lebih tahan terhadap *outlier* atau data ekstrem yang dapat mempengaruhi distribusi data secara signifikan. Ini sering digunakan dalam berbagai teknik *machine learning* untuk memastikan bahwa skala fitur tidak dipengaruhi oleh data yang berada jauh di luar distribusi normal.

$$X_{scaled} = \frac{X - median(x)}{k \times MAD} \quad (9)$$

D. Logistic Regression

Logistic regression adalah masalah di mana hasil yang didapat berupa kelas-kelas diskrit daripada nilai kontinu. Dalam metodologi statistik, regresi logistik menggunakan metode kemungkinan maksimum (*maximum likelihood*) untuk menghitung parameter dari masing-masing variabel [12]. Sebaliknya, dalam metodologi *machine learning*, *log loss* akan diminimalkan sehubungan dengan koefisien β (juga dikenal sebagai bobot). Regresi logistik memiliki bias yang tinggi dan kesalahan varians yang rendah.

$$P(y = k|x) = \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)} \quad (10)$$

dengan,

$$z_k = \beta_{k0} + \beta_{k1x1} + \beta_{k1x2} + \cdots + \beta_{kpxp} = \beta_{k0} + \beta_k x \quad (11)$$

E. *K-nearest Neighbors* (KNN)

K-nearest neighbors (KNN) merupakan model *machine learning* non-parametrik di mana model mengingat observasi pelatihan untuk mengklasifikasikan data uji yang belum pernah dilihat. Model ini mulai bekerja hanya selama fase pengujian/evaluasi untuk membandingkan observasi uji yang diberikan dengan observasi pelatihan terdekat, yang akan memakan waktu signifikan dalam membandingkan setiap titik data uji [12].

Metrik jarak, yaitu menghitung jarak antara x dengan setiap data dalam D, dengan n adalah jumlah fitur.

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} \left(x_j - x_i, j\right)^2} \quad 12$$

Pemilihan *K* tetangga terdekat dengan mengurutkan dataset *D* berdasarkan jarak $d(x, x_i)$ dalam urutan naik. Selanjutnya dilakukan pengambilan K terdekat $\left(x_{(1)}, x_{(2)}, \dots, x_K\right)$. Untuk setiap kelas $c \in C$di mana C adalah himpunan semua kelas, hitung jumlah tetangga dalam K terdekat yang memiliki label c.

$$n_c = \sum_{I=1}^{K} \mathbb{I}\left(Y_{(i)} = c\right) \quad 13$$

dengan $\mathbb{I}$ adalah fungsi indikator:

$$\mathbb{I}(y_{(i)} = c = \begin{cases} 1, & jika\ y_{(i)} = c \\ 0, & lainnya \end{cases} \quad 14$$

Selanjutnya adalah melakukan prediksi kelas $\hat{y}$ untuk x dengan kelas jumlah tetangga terbanyak.

$$\hat{y} = \arg \underset{c \in C}{max}\, n_c \quad 15$$

F. *Decision Tree*

Decision tree merupakan algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi maupun regresi berbasis *tree structure*. Cara kerja algoritma ini dengan mempartisi dataset ke dalam subset-subset secara berulang berdasarkan nilai fitur untuk membuat keputusan [13]. Untuk *classification tree*, pemisahan data dilakukan dengan memaksimalkan kemurnian (*purity*) dalam node. Kriteria yang digunakan adalah *gini impurity.*

$$G = \sum_{k=1}^{K} \hat{p}_{mk}\left(1 - \hat{p}_{mk}\right) \quad 16$$

G. *Random Forest*

Random forest adalah salah satu algoritma *ensemble learning* yang digunakan untuk klasifikasi, regresi, dan tugas lainnya. Algoritma ini bekerja dengan menggabungkan banyak *decision tree* yang dilatih pada subset data yang berbeda untuk meningkatkan akurasi dan mengurangi risiko *overfitting* [14]. Konsep utama dari Random Forest adalah membangun hutan yang terdiri dari banyak pohon keputusan yang saling tidak berkorelasi (*decorrelated trees*).

$$\hat{y} = \underset{y}{argmax} \left(\sum_{t=1}^{T} \mathbb{I}(h_t(x) = y) \right) \quad 17$$

dengan

$\hat{y}$: Kelas yang diprediksi oleh random forest.

T : Jumlah pohon dalam hutan.

$h_t(x)$: Prediksi kelas oleh pohon t untuk input x.

$\mathbb{I}$: Fungsi indikator, bernilai 1 jika $h_t(x) = y$, dan 0 jika tidak

$argmax$: Operator yang memilih nilai y dengan jumlah voting terbesar.

H. *Extreme Gradient Boosting* (XGBoost)

Extreme Gradient Boosting (XGBoost) adalah algoritma berbasis *boosting* yang dioptimalkan untuk kinerja tinggi dan efisiensi. Algoritma ini merupakan pengembangan dari algoritma *Gradient Boosting* yang dirancang untuk mengatasi masalah kecepatan, efisiensi, dan kinerja model. XGBoost menggunakan pendekatan berbasis pohon keputusan dan mengintegrasikan teknik seperti *regularisasi*, *subsampling*, dan *parallel computing* [15].

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t \times h_t(x) \right) \quad 18$$

I. *Adaptive Boosting* (Adaboost)

Adaptive boosting bekerja dengan menggabungkan banyak model lemah (pohon keputusan dengan kedalaman rendah atau *decision stumps*) menjadi sebuah model kuat. Algoritma ini menyesuaikan bobot data secara adaptif untuk meningkatkan kinerja model lemah pada sampel yang sulit diklasifikasikan [16].

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t \times h_t(x)\right) \quad (19)$$

J. Ketepatan Klasifikasi

Pengukuran ketepatan dilakukan untuk melihat performa klasifikasi yang telah dilakukan [12]. Dalam mengukur ketepatan klasifikasi, perlu diketahui jumlah data pada setiap kelas aktual yang terdiri dari TP (*True Positive*) yaitu label positif yang tepat diklasifikasi kedalam kelas positif, TN (*True Negative*) yaitu label yang tepat diklasifikasi dalam kelas negatif, FP (*False Positive*) adalah label negatif yang terklasifikasi kedalam kelas positif, dan FN (*False Negative*) yaitu label positif yang terklasifikasi kedalam kelas negatif [17]. Pada penelitian ini menggunakan *F1-Score.*

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (20)$$

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (21)$$

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative(FN)} \quad (22)$$

3. PROSES ANALISIS

A. Persiapan Data

Tahapan persiapan data bertujuan untuk menyiapkan data latih dan uji yang sesuai. Pada tahap ini dilakukan penggabungan *(join)* data dari berbagai data ke data latih dan uji.

B. Praproses Data

1) Penanganan *Missing Value*

Penanganan *missing value* dalam analisis ini menggunakan metode *complete case analysis (listwise deletion)*, di mana setiap pengamatan yang memiliki nilai hilang akan dihapus dari dataset. Meski berpotensi mengurangi jumlah data, metode ini dipilih karena sesuai dengan karakteristik dataset besar yang dimiliki. Penelitian oleh [18] dan [19] menyatakan bahwa *listwise deletion* dapat diterima pada dataset berukuran besar selama pengurangan data tidak mempengaruhi representasi populasi.

2) Penanganan *Outlier*

Penanganan *outlier* dilakukan pada kolom numerik menggunakan metode *winsorization*, di mana nilai *outlier* diganti dengan *threshold* atas dan bawah berdasarkan visualisasi boxplot untuk setiap kolom. Pendekatan ini didukung oleh [20] yang menyatakan bahwa *winsorization* lebih efektif dibanding penghapusan *outlier* karena mempertahankan ukuran sampel dan struktur data.

3) *Feature Engineering*

Beberapa *feature engineering* yang dilakukan meliputi pembuatan fitur/kolom baru, *bining, encoding*, dan *feature extraction.* Pembuatan kolom baru digunakan untuk menghasilkan kolom yang lebih relevan dari kolom yang ada. *Binning* dilakukan untuk mengekstrak informasi temporal yang lebih spesifik dari data waktu pemeriksaan mesin. *Encoding* digunakan untuk mengubah variabel kategorikal menjadi bentuk numerik yang dapat diproses oleh model *machine learning*. Sedangkan *feature extraction* dilakukan untuk membuat kolom baru dari kombinasi kolom-kolom yang ada.

4) Normalisasi Data

Normalisasi data dilakukan pada kolom numerik non-kategorikal menggunakan metode *Robust Scaler*. Pemilihan *Robust Scaler* didasarkan pada

kemampuannya menangani data dengan *outlier*, karena lebih tahan terhadap pengaruh nilai ekstrim [21]. Sementara itu, kolom kategorikal akan dibiarkan dalam bentuk aslinya karena telah melalui proses *encoding* sebelumnya dan tidak memerlukan normalisasi lebih lanjut.

5) Pembagian Data

Data hasil normalisasi kemudian dibagi menjadi subset pelatihan dan subset validasi dengan proporsi 80:20. Proporsi 80:20 sering digunakan sebagai standar karena memberikan keseimbangan antara jumlah data yang cukup untuk melatih model dan data yang cukup untuk mengevaluasi performa model.

6) Penanganan Kelas Tidak Seimbang

Untuk mengatasi ketidakseimbangan kelas dalam klasifikasi multikelas, analisis ini akan menggunakan pendekatan *hybrid sampling* yang menggabungkan teknik *undersampling* dan *oversampling*. *Undersampling* akan diterapkan pada kelas mayoritas untuk mengurangi dominasinya, sementara *oversampling* akan dilakukan pada kelas-kelas minoritas untuk meningkatkan representasinya. Penanganan kelas tidak seimbang dilakukan pada subset data pelatihan.

7) *Feature Selection*

Feature selection dilakukan untuk mengoptimalkan pemilihan kolom/fitur yang relevan. Metode yang digunakan adalah *Logistic Regression Multinomial* dengan *L1 regularization* (Lasso). Menurut [22] dalam penelitiannya tentang klasifikasi multikelas menggunakan Lasso, metode ini memiliki dua keunggulan utama. Pertama, Lasso dapat melakukan seleksi fitur secara otomatis dengan mendorong koefisien fitur yang kurang penting mendekati nol. Kedua, pendekatan ini menghasilkan model yang lebih *interpretable* dengan tetap mempertahankan akurasi prediksi yang baik.

C. Modeling

Modeling adalah tahap di mana data yang telah diproses dan fitur-fitur yang telah dipilih digunakan untuk melatih algoritma *machine learning*. Pada tahap ini, subset pelatihan digunakan untuk membangun model, sementara subset validasi digunakan untuk mengevaluasi performanya. Beberapa metode

machine learning diterapkan dalam proses ini, yaitu regresi logistik multinomial, *K-Nearest Neighbors* (KNN), *decision tree* (DT), *random forest* (RF), XGBoost, dan AdaBoost. Evaluasi performa model akan mengutamakan *f1-score* sebagai metrik utama, mengingat pentingnya keseimbangan antara *precision* dan *recall* dalam konteks klasifikasi multikelas.

4. HASIL ANALISIS

A. Persiapan Data

Informasi penggabungan *(join)* data yang dilakukan pada data latih dan uji ditunjukkan pada Tabel 1.

Tabel 1. Informasi *Join* Data

Kolom	Data Sumber	Kolom Penghubung
Priority, dan Area	Area-List	ID_Area
Age, Last Maintenance, Status Sparepart, dan Country Machine	Machine-Area	ID_Area dan ID_Mesin

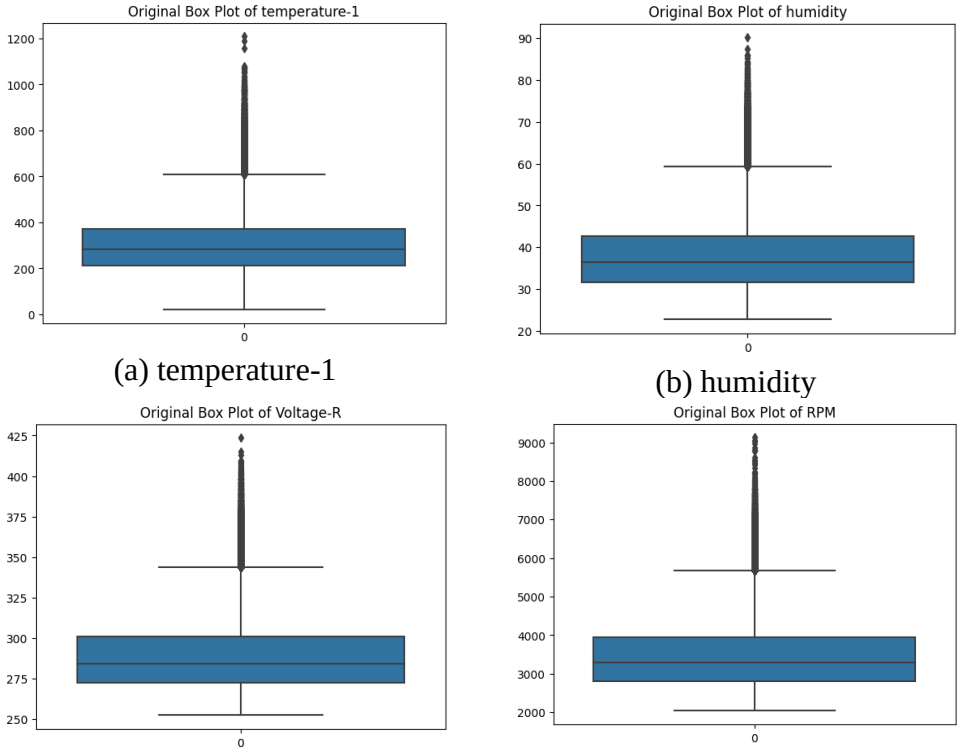
B. Eksplorasi Data

Eksplorasi data dilakukan untuk memahami karakteristik dataset, dengan fokus pada tiga aspek utama, yakni analisis *missing value*, identifikasi *outlier*, dan distribusi kelas target. Sebaran *missing value* data latih hasil penggabungan ditunjukkan pada Tabel 2.

Tabel 2. Informasi *Missing Value* Data Latih

Kolom	Jumlah *Missing Value*	Persentase
Country Machine	4.031.795	31,01%
Last Maintenance	4.031.795	31,01%
⋮	⋮	⋮
Current-R	182.444	1,40%
Current-M	141.165	1,08%

Analisis *outlier* pada kolom numerik dilakukan menggunakan visualisasi boxplot. Hasil visualisasi menunjukkan keberadaan *outlier* yang signifikan pada beberapa variabel, seperti terlihat pada Gambar 2 yang menampilkan boxplot untuk temperature-1, humidity, Voltage-R, dan RPM.



(a) temperature-1 (b) humidity

(c) Voltage-R (d) RPM

Gambar 2. Sampel visualisasi boxplot (a) temperature-1, (b) humidity, (c) Voltage-R, dan (d) RPM

Selanjutnya akan ditunjukkan distribusi kolom target seperti pada Tabel 3. Hasil tersebut menunjukkan bahwa distribusi target tidak seimbang, sehingga perlu dilakukan penanganan kelas tidak seimbang pada tahapan analisis selajutnya.

Tabel 3. Sebaran Kelas Kolom Status

Status	Jumlah	Persentase
Normal	7.986.989	0,61%
Warning	2.550.024	0,20%
Breakdown	2.462.987	0,19%

C. Praproses Data

1) Penanganan *Missing Value*

Penanganan *missing value* menggunakan *complete case analysis (listwise deletion)* menghasilkan pengurangan data dari 13 juta menjadi 429.841 pengamatan. Sebaran kelas target sebelum dan sesudah penanganan *missing value* ditunjukkan pada Tabel 4.

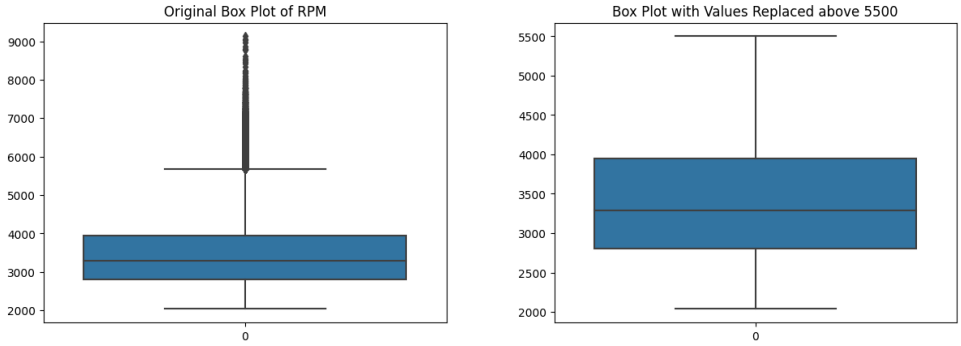
Tabel 4. Perbandingan Sebaran Kelas Target

Status	Sebelum Penanganan		Sesudah Penanganan	
	Jumlah	Persentase	Jumlah	Persentase
Normal	7.986.989	0,61%	264.014	0,61%
Warning	2.550.024	0,20%	84.339	0,20%
Breakdown	2.462.987	0,19%	81.488	0,19%

Hasil pada Tabel 4 menunjukkan bahwa meskipun terjadi pengurangan jumlah data yang signifikan, proporsi masing-masing kelas target tetap terjaga. Hal ini memperkuat keputusan penggunaan *complete case analysis* sebagai metode penanganan *missing value* yang tepat untuk dataset ini.

2) Penanganan *Outlier*

Visualisasi sampel boxplot perbandingan sebelum dan sesudah penanganan *outlier* ditunjukkan pada Gambar 3.



(a) Sebelum penanganan *outlier* (b) Sesudah penanganan *outlier*

Gambar 3. Visualisasi boxplot RPM

3) *Feature Engineering*

Informasi pembuatan kolom baru yang dilakukan dalam analisis ini ditampilkan pada Tabel 5.

Tabel 5. Informasi Pembuatan Kolom Baru

Kolom Asal	Kolom Baru
Timestamp	jam, hari, dan bulan pemeriksaan
Last Maintenance	jam, hari, bulan *maintenance*
Last Maintence dan timestamp	jarak dari *maintenace* (bulan)

Tiga jenis *binning* diterapkan untuk menghasilkan variabel kategorikal baru seperti ditunjukkan pada Tabel 6.

Tabel 6. Informasi *Binning*

Kolom Asal	*Binning*
Hari	Apakah akhir pekan/tidak
Jam	Kategori hari
Hari	Apakah akhir bulan/tidak

Encoding yang dilakukan terdiri atas *label encoding, one hot encoding,* dan *ordinal encoding* seperti yang ditunjukkan pada Tabel 7.

Tabel 7. Informasi *Encoding* Kolom

Kolom	Jenis *Encoding*
Power Backup, Weekend, End Month, Time of Day	*Label Encoding*
Country Machine, Status Sparepart, Priority	*One Hot Encoding*
Status	*Ordinal Encoding*

Informasi *feature extraction* yang dilakukan dalam analisis ini ditampilkan pada Tabel 8.

Tabel 8. Informasi *Feature Extraction*

Kolom Asal	Kolom Baru
Voltage-L, Voltage-R, Voltage-M	Average Voltage
Current-M, Current-R, Current-T	Average Current
Vibration-1, Vibration-2	Total Vibration, Average Vibration
Temperature-1, Temperature-2, Temperature-3	Average Temperature
Temperature_10H_max, Temperature_10H_min	Temeperature Range
RPM-1, RPM-2, RPM-3	Total RPM, Average RPM

4) Normalisasi Data

Perbandingan data kolom numerik non-kategorikal sebelum dan sesudah normalisasi ditunjukkan pada Tabel 9.

Tabel 9. Sampel Hasil Normalisasi Data

Sebelum Normalisasi		Sesudah Normalisasi	
temperature-1	temperature-2	temperature-1	temperature-2
117,3227	22,2565	-1,0411	-0,1832
165,3943	37,8129	-0,7394	0,4991
465,5929	37,7589	1,1442	0,4968

5) Pembagian Data

Informasi jumlah data latih dan validasi setelah pembagian ditunjukkan pada Tabel 10.

Tabel 10. Informasi Pembagian Data

Subset	Proporsi	Jumlah Data
Data Latih	80%	343.872
Data Validasi	20%	85.969

6) Penanganan Kelas Tidak Seimbang

Penanganan kelas tidak seimbang dilakukan pada data latih. Perbandingan sebaran kelas sebelum dan sesudah penanganan kelas tidak seimbang ditampilkan pada Tabel 11.

Tabel 11. Perbandingan Kelas Sebelum dan Sesudah Penanganan

Sebelum Penanganan		Sesudah Penanganan	
Kelas	Jumlah	Kelas	Jumlah
Normal	264.014	Normal	100.000
Warning	84.339	Warning	100.000
Breakdown	81.488	Breakdown	100.000

7) Feature Selection

Hasil *feature selection* menggunakan *Logistic Regression Multinomial* dengan *L1 regularization* (Lasso) ditunjukkan pada Tabel 12.

Tabel 12. Daftar Kolom Hasil *Feature Selection*

Kategori	Kolom Terpilih
Kolom Asli	temperature-2, apparent_temperature_max, apparent_temperature_min, Voltage-M, RPM, Power, Power_Backup, Age
Hasil *Feature Engineering*	hour, day, day_maintenance, month_maintenance, weekend, end_month, time_of_day, Country_Machine_1, Country_Machine_2, Country_Machine_4, Status_Sparepart_0, Status_Sparepart_2, Status_Sparepart_3, Status_Sparepart_4, Status_Sparepart_5, Priority_1, Priority_2

Hasil Tabel 12 menunjukkan bahwa proses *feature engineering* berhasil menambahkan kolom-kolom baru yang relevan untuk meningkatkan performa model.

D. Modeling

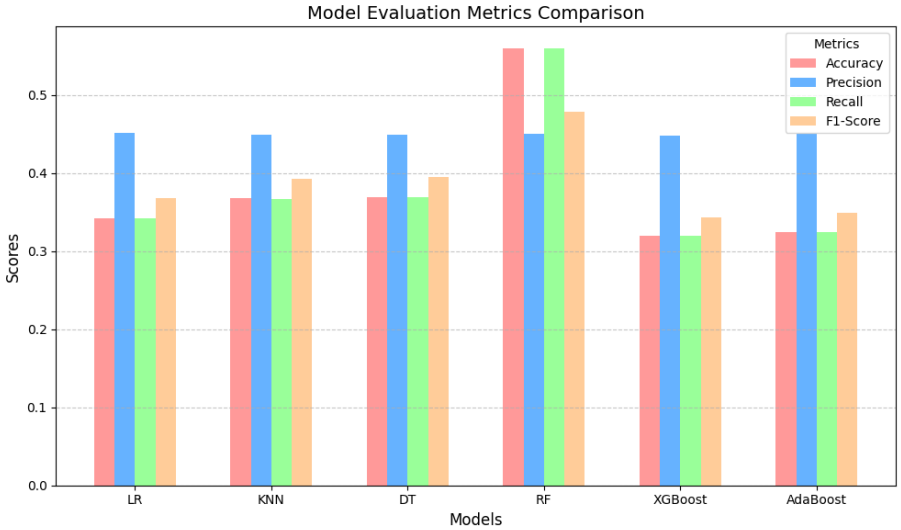
Perbandingan evaluasi model *machine learning* pada data validasi ditunjukkan pada Tabel 13.

Tabel 13. Perbandingan Evaluasi Model

Model	*Accuracy*	*Precision*	*Recall*	*F1-Score*
LR	0,3416	**0,4512**	0,3416	0,3681
KNN	0,3678	0,4487	0,3670	0,3931
DT	0,3688	0,4485	0,3688	0,3944
RF	**0,5595**	0,4506	**0,5595**	**0,4781**
XGBoost	0,3196	0,4483	0,3196	0,3431
AdaBoost	0,3241	0,4503	0,3241	0,3488

Hasil evaluasi menunjukkan bahwa *random forest* (RF) unggul hampir di semua metrik evaluasi, dengan nilai *f1-score* tertinggi sebesar 0,4781. Model *decision tree* (DT) dan *k-nearest neighbors* (KNN) mendekati performa RF, terutama dalam *f1-score*. Sebaliknya, model XGBoost dan AdaBoost memiliki

performa yang lebih rendah dibandingkan model lainnya. Visualisasi evaluasi perbandingan model ditampilkan pada Gambar 4.



Gambar 4. Visualisasi Perbandingan Evaluasi Model

5. KESIMPULAN DAN REKOMENDASI

a. Kesimpulan

Penerapan model *machine learning* dalam klasifikasi kerusakan mesin sangat penting untuk mendeteksi dan mengidentifikasi masalah mesin secara dini. Dengan menggunakan teknik seperti klasifikasi berbasis kecerdasan buatan, seperti algoritma klasifikasi berbasis pohon keputusan, regresi logistik, dan boosting dapat dipantau secara *real-time* untuk mendeteksi kerusakan. Pada penelitian ini model *machine learning* mampu melakukan deteksi klasifikasi kerusakan mesin mesin tersebut.

Dari hasil evaluasi performa berbagai model *machine learning*, terlihat bahwa *random forest* (RF) unggul hampir di semua metrik evaluasi, dengan nilai f1-score tertinggi sebesar 0,4781. Model *decision tree* (DT) dan *k-nearest neighbors* (KNN) menunjukkan performa yang cukup dekat dengan RF, terutama dalam metrik *f1-score*. Di sisi lain, model XGBoost dan AdaBoost memiliki performa yang lebih rendah dibandingkan model lainnya.

Penerapan teknik *data engineering*, seperti melakukan *join* antara tabel yang terpisah, serta penerapan proses *data preprocessing* seperti penghapusan nilai kosong, pembatasan *outlier*, normalisasi data, dan penyeimbangan kelas melalui metode *undersampling* dan *oversampling*, mampu meningkatkan performa model secara signifikan. Selain itu, penggunaan *feature engineering* dan *feature selection* membantu memperbaiki kualitas model dengan menghilangkan variabel yang kurang relevan, sehingga menghasilkan model yang lebih efisien dan akurat.

b. Rekomendasi

Berdsarkan hasil analisis yang diperoleh, maka dapat dituliskan beberapa rekomendasi sebagimana terdapat pada Tabel 14.

Tabel 14. Rekomendasi

Aspek	Rekomendasi
Penanganan *Missing Value*	Menggunakan metode imputasi berbasis model seperti KNN atau MICE.

Feature Selection	Menggunakan *feature importance* atau *Recursive Feature Elimination* (RFE).
Penanganan Kelas Tidak Seimbang	Menggunkan metode lain seperti SMOTE atau ADASYN.
Modeling	Melakukan *tuning hyperparameter* model menggunakan *Grid Search* dan *Cross Validation*. Selain itu penggunaan model berbasis *Neural Network* juga bisa dicoba.

DAFTAR PUSTAKA DAN LAMPIRAN

[1] N. Aznawati, "Analisa Algoritma C4.5 Untuk Memprediksi Penjualan Motor Pada Pt. Capella Dinamik Nusantara Cabang Muka Kuning ," *Jurnal Ilmiah Ilmu Komputer*, vol. 13, no. 1, hlm. 1–6, Feb 2018.

[2] R. Kunst, L. Avila, A. Binotto, E. Pignaton, S. Bampi, dan J. Rochol, "Improving devices communication in Industry 4.0 wireless networks," *Eng Appl Artif Intell*, vol. 83, hlm. 1–12, Agu 2019, doi: 10.1016/j.engappai.2019.04.014.

[3] V. Tessoni dan M. Amoretti, "Advanced statistical and machine learning methods for multi-step multivariate time series forecasting in predictive maintenance," *Procedia Comput Sci*, vol. 200, hlm. 748–757, 2022, doi: 10.1016/j.procs.2022.01.273.

[4] S. Vollert, M. Atzmueller, dan A. Theissler, "Interpretable Machine Learning: A brief survey from the predictive maintenance perspective," dalam *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, Sep 2021, hlm. 01–08. doi: 10.1109/ETFA45728.2021.9613467.

[5] P. O'Donovan, C. Gallagher, K. Leahy, dan D. T. J. O'Sullivan, "A comparison of fog and cloud computing cyber-physical interfaces for Industry 4.0 real-time embedded machine learning engineering applications," *Comput Ind*, vol. 110, hlm. 12–35, Sep 2019, doi: 10.1016/j.compind.2019.04.016.

[6] H. Boyes, B. Hallaq, J. Cunningham, dan T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Comput Ind*, vol. 101, hlm. 1–12, Okt 2018, doi: 10.1016/j.compind.2018.04.015.

[7] A. Kaushik dan D. K. Yadav, "Analysing Failure Prediction for a Manufacturing Firm Using Machine Learning Algorithms," 2023, hlm. 457–463. doi: 10.1007/978-981-19-9285-8_44.

[8] C. Zhou dan C.-K. Tham, "GraphEL: A Graph-Based Ensemble Learning Method for Distributed Diagnostics and Prognostics in the Industrial Internet of Things," dalam *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, Des 2018, hlm. 903–909. doi: 10.1109/PADSW.2018.8644943.

[9] Z. Liu, H. Wang, Y. Zhou, J. Liu, X. Zhang, dan G. Hu, "Effect of microwave chlorine depleted pyrolyzate on the combustion characteristics of refuse derived fuel derived from package waste," *Waste Management*, vol. 82, hlm. 1–8, Des 2018, doi: 10.1016/j.wasman.2018.09.053.

[10] C. M. Carbery, R. Woods, dan A. H. Marshall, "A Bayesian network based learning system for modelling faults in large-scale manufacturing," dalam *2018 IEEE International Conference on Industrial Technology (ICIT)*, IEEE, Feb 2018, hlm. 1357–1362. doi: 10.1109/ICIT.2018.8352377.

[11] D. K. Yadav, A. Kaushik, dan N. Yadav, "Predicting machine failures using machine learning and deep learning algorithms," *Sustainable Manufacturing and Service Economics*, vol. 3, hlm. 100029, 2024, doi: 10.1016/j.smse.2024.100029.

[12] Y. Heryadi dan T. Wahyono, *Machine Learning: Konsep dan Implementasi*. Yogyakarta: Gava Media Yogyakarta, 2020.

[13] G. James, D. Witten, T. Hastie, dan R. Tibshirani, "Tree-Based Methods," 2021, hlm. 327–365. doi: 10.1007/978-1-0716-1418-1_8.

[14] R. Genuer dan J.-M. Poggi, "Random Forests," 2020, hlm. 33–55. doi: 10.1007/978-3-030-56485-8_3.

[15] T. Chen dan C. Guestrin, "XGBoost," dalam *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Agu 2016, hlm. 785–794. doi: 10.1145/2939672.2939785.

[16] R. E. Schapire, “Explaining AdaBoost,” dalam *Empirical Inference*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, hlm. 37–52. doi: 10.1007/978-3-642-41136-6_5.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, dan W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *Journal of Artificial Intelligence Research*, vol. 16, hlm. 321–357, Jun 2002, doi: 10.1613/jair.953.

[18] F. Arteaga dan A. J. Ferrer-Riquelme, “Missing Data,” dalam *Comprehensive Chemometrics*, Elsevier, 2009, hlm. 285–314. doi: 10.1016/B978-044452701-1.00125-3.

[19] H. Kang, “The prevention and handling of the missing data,” *Korean J Anesthesiol*, vol. 64, no. 5, hlm. 402, 2013, doi: 10.4097/kjae.2013.64.5.402.

[20] S. K. Kwak dan J. H. Kim, “Statistical data preparation: management of missing values and outliers,” *Korean J Anesthesiol*, vol. 70, no. 4, hlm. 407, 2017, doi: 10.4097/kjae.2017.70.4.407.

[21] V. Sharma, “A Study on Data Scaling Methods for Machine Learning,” *International Journal for Global Academic & Scientific Research*, vol. 1, no. 1, Feb 2022, doi: 10.55938/ijgasr.v1i1.4.

[22] A. Robles-Guerrero, T. Saucedo-Anaya, E. González-Ramírez, dan J. I. De la Rosa-Vargas, “Analysis of a multiclass classification problem by Lasso Logistic Regression and Singular Value Decomposition to identify sound patterns in queenless bee colonies,” *Comput Electron Agric*, vol. 159, hlm. 69–74, Apr 2019, doi: 10.1016/j.compag.2019.02.024.