

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Tabel 2. 1 Penelitian Terdahulu

No	Penulis / Tahun	Judul Jurnal	Model	Metric	Masalah
1	Yara Kayyali Elalem, et al., 2023 [1]	A machine learning-based framework for forecasting sales of new products with short life cycles using deep neural networks	ARIMAX MASE: 24.32%	ARIMAX terbaik untuk prediksi penjualan; DNN lebih robust terhadap noise.	Perusahaan semakin sering meluncurkan produk baru dengan siklus hidup pendek, sehingga memerlukan peramalan permintaan.
2	Md. Parvezur Rahman Mahina, et al., 2025 [2]		Voting Regressor MAE: 0.62, RMSE: 1.54, R ² : 0.88	Voting Regressor outperform Random Forest & Linear Regression	Dibutuhkan model prediksi yang lebih robust dan efisien untuk mengurangi error sekaligus menjaga kinerja komputasi.
3	Sudha Ka, et al., 2024 [3]	Leveraging Machine Learning for Customer Intelligence: An Experimental Analysis Learning Classifiers	C4.5 Accuracy: 89.87% Random Tree 87.36% Logistic Regression Accuracy: 91.32%	C4.5 terbaik untuk akurasi; Naive Bayes cepat & efisien	Ketiadaan kerangka sistematis untuk memilih classifier terbaik sesuai dataset pelanggan.
4	Vincent Adomat, et al., 2023 [4]	A Machine Learning Approach for Revenue Management in Cloud Manufacturing	LONG SHORT-TERM MEMORY MAE: 14.65, RMSE: 19.82 Autoregression MAE: 32.13, RMSE: 58.25	LONG SHORT-TERM MEMORY outperform model tradisional dalam prediksi biaya dan	Keterbatasan model deterministik tradisional yang tidak cukup akurat menghadapi kompleksitas tersebut.

No	Penulis / Tahun	Judul Jurnal	Model	Metric	Masalah
				optimasi revenue	
5	Hamed GhorbanTanhaei, et al., 2024 [7]	Predictive analytics in customer behavior: Anticipating trends and preferences	Logistic Regression Accuracy: 82.6%, F1: 0.766, ROC-AUC: 0.878 Random Forest Accuracy: 80.6%, Precision: 0.62, Recall: 1.0 Gradient Boosting Accuracy: 82.3%	RF & LR efektif untuk prediksi perilaku pelanggan	Sulit memprediksi perilaku pelanggan dengan model tradisional dan kurangnya pemanfaatan machine learning untuk mendukung keputusan portofolio pelanggan.
6	C.V. Shakila & Khadar Babu, 2025 [8]	Evaluating and contrasting machine learning and statistical techniques for time series forecasting with hyperparameter optimization	Random Forest Accuracy: 85%, RMSE: 0.4411, MAE: 0.3022, R ² : 0.8565 Artificial Neural Network Accuracy: 83% ARIMA Accuracy: 80%	Bayesian Optimization meningkatkan akurasi RFR	ARIMA dan Artificial Neural Network kurang akurat; RFR butuh optimasi hyperparameter.
7	Usha R, Vijiyalakshmi V, Shenbaha S, 2025 [20]	Enhancing Business Profitability Through Regression-Based Sales Forecasting	Polynomial Regression R ² = 0.9588 Multi-Linear Regression → R ² = 0.89	Polynomial Regression paling akurat dalam memprediksi pendapatan harian	Kebutuhan model prediksi yang mampu mengoptimalkan profit sekaligus strategi pemasaran.
8	Paul Tunsch, Nils Becker, Berthold Schlecht, 2024 [13]	Development of a workflow to build optimal machine learning models for stress concentration factor regression	Optimized Artificial Neural Network MSE = 0.0001, MAPE = 0.37%	Optimized Artificial Neural Network paling akurat dan cepat	Analitik cepat tapi kurang akurat; belum ada gabungan FEM-efisien.
9	Muhammad Nadeem Gul, et al., 2025 [16]	Data-driven decisions in education using a comprehensive	CatBoost Accuracy = 87.46% Gradient	CatBoost terbaik untuk prediksi	Butuh Machine Learning berbasis data

No	Penulis / Tahun	Judul Jurnal	Model	Metric	Masalah
		machine learning framework for student performance prediction	Boosting 87.28% Decision Tree 82.42%	performa siswa	siswa; sistem prediksi ini masih lemah.
10	Suan Lee, et al., 2023 [9]	A deep learning model for predicting the number of stores and average sales in commercial district	TF MSE: 0.00401, RMSE: 0.00445, R ² : -0.63709 Linear MSE: 0.00569, RMSE: 0.00564, R ² : -1.07328	DNN outperform model lainnya dalam prediksi jumlah toko & penjualan	Prediksi toko dan penjualan sulit; model akurat masih kurang.

Beberapa penelitian terdahulu telah menunjukkan bahwa model statistik tradisional masih dapat digunakan sebagai baseline dalam peramalan penjualan, namun performanya umumnya dilaporkan lebih rendah dibandingkan metode Machine Learning dan Deep Learning. Pada studi[1] ARIMAX dan DNN digunakan untuk memprediksi penjualan produk dengan siklus hidup pendek. Dalam penelitian tersebut, ARIMAX ditemukan cukup efektif, tetapi Deep Neural Network dinyatakan lebih robust terhadap noise, sehingga keterbatasan pendekatan tradisional semakin ditegaskan.

Pendekatan ensemble juga telah dilaporkan memberikan hasil yang lebih baik. Studi [2] menunjukkan bahwa Voting Regressor diterapkan pada konteks rantai pasok berkelanjutan dan menghasilkan nilai R² yang cukup tinggi (0.88), sehingga model tersebut dinyatakan mengungguli model tunggal. Temuan serupa diperoleh pada penelitian [3] di mana boosting algorithms diuji, dan CatBoost serta Gradient Boosting dilaporkan mencapai akurasi 87%, sedangkan Extreme Gradient Boosting mencapai 91.42%, sehingga metode boosting dianggap mendominasi dalam penanganan data tabular.

Sementara itu, pendekatan berbasis klasifikasi digunakan pada penelitian[5], di mana C4.5, Random Tree, dan Logistic Regression dibandingkan menggunakan data pelanggan. Hasil yang diperoleh bervariasi: C4.5 dilaporkan unggul pada akurasi, sedangkan Logistic Regression menunjukkan specificity yang lebih tinggi.

Pada penelitian [9], Logistic Regression, Random Forest, dan Gradient Boosting dibandingkan untuk memprediksi perilaku pelanggan, dan hasil studi tersebut menunjukkan bahwa setiap algoritma memiliki keunggulan metrik yang berbeda sehingga tidak ada satu model pun yang secara konsisten lebih baik pada seluruh indikator.

Penelitian [6] menunjukkan bahwa Long Short-Term Memory terbukti mengungguli autoregression dalam prediksi revenue berbasis time series, karena MAE dan RMSE yang diperoleh jauh lebih rendah. Konsistensi temuan ini diperlihatkan juga pada studi, di mana Deep Neural Network dilaporkan mengungguli model linear dalam memprediksi jumlah toko dan penjualan di distrik komersial. Pada studi [11] Optimized Artificial Neural Network bahkan dilaporkan mencapai MSE 0.0001 dan MAPE 0.37%, sehingga model tersebut dinyatakan sangat efisien sekaligus akurat.

Selain itu, penelitian [10] menekankan bahwa hyperparameter optimization berpengaruh signifikan terhadap performa model. Dengan Bayesian Optimization, performa Random Forest dilaporkan meningkat (R^2 0.8565), sehingga model tersebut mengungguli ARIMA dan Artificial Neural Network. Studi [22] menambahkan perspektif berbeda, di mana regresi polinomial sederhana ditemukan lebih akurat dibandingkan regresi linier berganda dalam memprediksi pendapatan harian, dengan R^2 mencapai 0.9588.

Secara keseluruhan, literatur terdahulu dapat disintesis menjadi tiga poin utama: (1) model tradisional tetap digunakan sebagai baseline namun dilaporkan memiliki performa lebih rendah dibandingkan metode Machine Learning dan Deep Learning modern, (2) model-model seperti Voting Regressor, CatBoost, dan Extreme Gradient Boosting secara konsisten dilaporkan memberikan performa yang lebih baik, dan (3) metode Deep Learning seperti Long Short-Term Memory, Artificial Neural Network, dan Deep Neural Network dinilai lebih unggul untuk pola data non-linear atau sekuensial.

Sebagian besar studi diarahkan langsung pada pengembangan hybrid atau ensemble, namun pembahasan yang fokus mengenai strategi hyperparameter tuning

masih jarang ditemukan. Penelitian [8] menunjukkan bahwa hyperparameter optimization dengan Bayesian Optimization dapat meningkatkan performa secara signifikan (R^2 0.8565), bahkan mengungguli Artificial Neural Network pada dataset terbatas. Sementara itu, penelitian [13] membuktikan bahwa Artificial Neural Network yang dioptimasi dapat mencapai akurasi sangat tinggi (MAPE 0.37%) pada dataset yang cukup besar. Penelitian lain juga [16] [17] secara konsisten melaporkan bahwa Extreme Gradient Boosting merupakan model paling akurat untuk forecasting pada education teknologi dibanding model lainnya.

Berdasarkan gap dan temuan literatur tersebut, penelitian ini difokuskan pada perbandingan dua model utama: Extreme Gradient Boosting yang terbukti superior untuk forecasting [16] [17], dan Artificial Neural Network yang efektif pada dataset besar [13] namun diperkirakan underperform pada data terbatas [8]. Kedua model mewakili kategori Machine Learning dan Deep Learning, dengan penambahan strategi tuning Optuna, Bayesian optimization untuk Extreme Gradient Boosting dan Hyperband sehingga diharapkan dapat dihasilkan model prediktif yang lebih akurat dan efisien tanpa memerlukan penggunaan resource yang sangat besar hingga tahap stacking, kombinasi ataupun hybrid model yang memerlukan komputasi dan resource yang sangat amat besar.

2.2 Teori yang berkaitan / Konsep Dasar

2.2.1 Teori Prediksi

Prediksi merupakan proses memperkirakan suatu nilai atau kejadian di masa depan berdasarkan pola historis maupun variabel yang relevan. Dalam konteks data mining, prediksi tidak hanya berfungsi sebagai estimasi, tetapi juga sebagai sarana memahami hubungan antar variabel yang memengaruhi hasil [16].

Seiring perkembangan teknologi, predictive analytics digunakan untuk beralih dari model statistik eksplanatif menuju pendekatan komputasional. Hal ini menekankan performa model dalam menghasilkan estimasi yang akurat dan reliabel, sehingga dapat mendukung pengambilan keputusan berbasis data di berbagai bidang [23]

2.2.2 Deep Learning

Deep Learning merupakan cabang dari Machine Learning yang menggunakan arsitektur jaringan saraf tiruan berlapis untuk mempelajari representasi data yang kompleks. DL mampu mengekstraksi fitur secara otomatis dari data besar tanpa memerlukan rekayasa fitur manual, sehingga sangat efektif dalam menangani masalah prediksi dan klasifikasi yang melibatkan data berukuran besar maupun tidak terstruktur [18].

Perkembangan Deep Learning semakin didukung oleh ketersediaan framework open source seperti TensorFlow dan PyTorch yang mempermudah implementasi model canggih dalam berbagai bidang [2]. Teknik Deep Learning telah banyak diterapkan, mulai dari navigasi otonom, analisis data dalam jumlah besar, hingga optimasi performa sistem melalui prediksi kegagalan peralatan. Dengan kemampuan generalisasi yang kuat, Deep Learning menjadi metode penting dalam meningkatkan efektivitas sistem berbasis kecerdasan buatan di berbagai domain [24].

2.2.3 Machine Learning

Machine Learning merupakan cabang dari Artificial Intelligence yang memungkinkan sistem belajar dari data dan meningkatkan kinerjanya secara bertahap tanpa diprogram secara eksplisit [1]. Melalui algoritma pembelajaran, Machine Learning mampu mengenali pola, membuat prediksi, dan mengambil keputusan berbasis data. Salah satu penerapan penting Machine Learning adalah predictive, yaitu strategi yang menganalisis data untuk mendeteksi potensi kerusakan sebelum terjadi, sehingga dapat mengurangi biaya operasional dan meningkatkan efisiensi [24].

Selain itu, Machine Learning juga digunakan dalam berbagai konteks lain seperti navigasi kapal, di mana algoritma dapat belajar dari pengalaman historis untuk memperbaiki navigasi secara real-time [1]. Teknik Machine Learning juga sering digunakan dalam pembuatan data sintetis guna memperkaya dataset pelatihan. Pendekatan ini membantu meningkatkan performa algoritma

peramalan dengan menambah variasi data yang tersedia untuk proses pembelajaran [25].

2.2.4 Hyperparameter Tuning

Hyperparameter optimization merupakan proses penting dalam pengembangan model machine learning karena secara langsung memengaruhi kompleksitas model, kemampuan generalisasi, serta efisiensi komputasi [10]. Pemilihan hyperparameter yang tidak tepat dapat menyebabkan performa model menjadi suboptimal, meningkatkan variansi, atau bahkan menambah biaya komputasi yang tidak perlu. Oleh karena itu, optimasi hyperparameter yang tepat sangat krusial untuk mendukung pengambilan keputusan berbasis data [10].

Metode tradisional seperti Grid Search dan Random Search digunakan secara luas dalam menemukan kombinasi hyperparameter terbaik. Grid Search mengevaluasi seluruh kombinasi secara sistematis, sedangkan Random Search memilih kombinasi secara acak sehingga lebih efisien meskipun tidak menjamin optimalitas penuh [10]. Seiring perkembangan, Bayesian Optimization diperkenalkan sebagai metode yang lebih adaptif dengan membangun model probabilistik dari fungsi objektif, lalu memilih nilai hyperparameter berdasarkan ekspektasi perbaikan. Hasil penelitian menunjukkan bahwa Bayesian Optimization mampu melampaui Grid Search dan Random Search baik dalam akurasi maupun efisiensi komputasi [10].

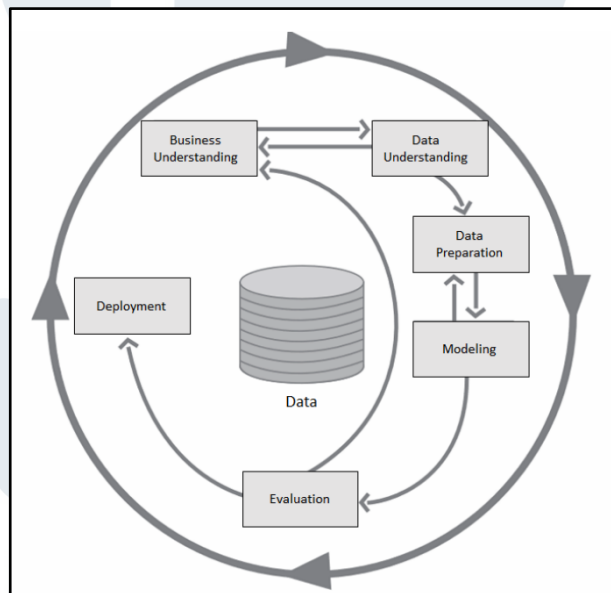
Selain itu, hyperparameter optimization juga diakui sebagai aspek vital untuk mengendalikan perilaku model dengan menemukan konfigurasi terbaik selama proses pembelajaran [26]. Namun, eksplorasi ruang hyperparameter secara menyeluruh sering kali memakan waktu, mahal, dan tidak efisien, sehingga diperlukan metode optimasi adaptif yang dapat mempercepat konvergensi tanpa mengorbankan performa [26].

2.3 Framework/Algoritma yang digunakan

2.3.1 CRISP-DM

CRISP-DM (Cross Industry Standard Process for Data Mining) merupakan salah satu metodologi yang paling banyak digunakan dalam penelitian maupun implementasi industri karena sifatnya yang generik dan fleksibel. Metodologi ini menyediakan kerangka kerja yang terstruktur mulai dari tahap pemahaman bisnis hingga validasi model, sekaligus memberikan panduan untuk proses persiapan data, pemodelan, dan evaluasi. Keunggulan CRISP-DM terletak pada kemampuannya untuk disesuaikan dengan berbagai konteks aplikasi, sehingga dapat menjadi pedoman standar dalam merancang serta mengimplementasikan penelitian berbasis data mining [27].

Sebagai sebuah model proses yang bersifat independen terhadap domain, CRISP-DM terdiri dari enam fase utama yang bersifat iteratif, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*.



Gambar 2. 1 Metodologi CRISP DM – Project

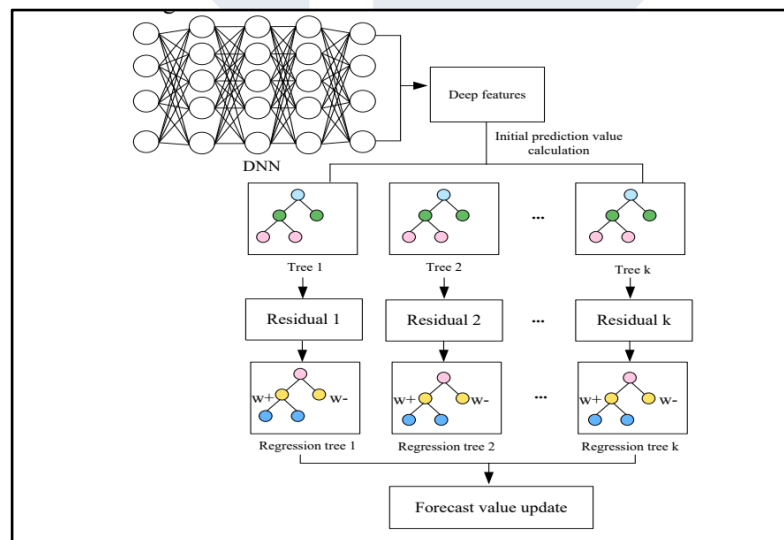
Sumber: [28]

Sifat iteratif ini memungkinkan peneliti maupun praktisi untuk kembali ke fase sebelumnya apabila ditemukan perbaikan yang diperlukan, sehingga kualitas model dapat ditingkatkan secara berkesinambungan. Oleh karena itu,

CRISP-DM menjadi metodologi yang kokoh dan teruji dalam mendukung proses analitik data secara sistematis dan terstruktur [29].

2.3.2 Extreme Gradient Boosting

Extreme Gradient Boost merupakan salah satu algoritma *ensemble learning* yang dikembangkan untuk meningkatkan kinerja prediksi terutama pada data terstruktur. Algoritma ini dikenal memiliki kemampuan yang kuat dalam menangani masalah ketidakseimbangan kelas (*class imbalance*), serta efisien dalam proses komputasi. Dengan menggabungkan pembelajaran fitur mendalam (*deep feature expression learning*) dan integrasi banyak *decision tree*, Extreme Gradient Boost tidak hanya mampu meningkatkan akurasi klasifikasi, tetapi juga memberikan landasan praktis yang penting bagi perusahaan dalam memahami perilaku konsumen secara lebih komprehensif [30] [31]. Hal ini menjadikan *Extreme Gradient Boost* relevan dalam mendukung strategi pemasaran perusahaan agar lebih tepat sasaran dan kompetitif.



Gambar 2. 2 Arsitektur Extreme Gradient Boost

Selain itu, Extreme Gradient Boost juga memungkinkan eksplorasi karakteristik multidimensi melalui analisis bobot fitur (*feature importance*). Dengan menggunakan model signifikansi fitur, variabel yang memiliki bobot terbesar dapat diidentifikasi sebagai faktor paling berpengaruh dalam

pembentukan model. Pendekatan ini tidak hanya meningkatkan transparansi model, tetapi juga memberikan wawasan baru bagi pengambilan keputusan berbasis data, khususnya dalam memahami preferensi pelanggan dan mengoptimalkan keputusan bisnis [30].

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in F$$

Formula 2. 1 Rumus Extreme Gradient Boost

Deskripsi symbol:

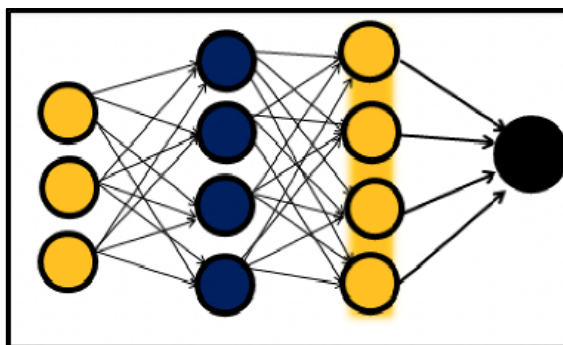
1. \hat{y} = Jumlah prediksi akhir dari setiap pohon
2. \sum = Penjumlahan (kontribusi setiap pohon)
3. k = Indeks pohon ke-k (adanya penambahan untuk setiap pohon)
4. f_k = Fungsi prediksi hasil dari pohon ke-k (individual)
5. F = Ruang fungsi (mengandung semua pohon yang sudah dilatih)
6. x_i = Vektor fitur dari sampel data ke-i (data yang diprediksi)

2.3.3 Artificial Neural Network

Artificial Neural Network merupakan model komputasi yang terinspirasi dari struktur dan fungsi otak manusia. Artificial Neural Network terdiri dari lapisan-lapisan neuron buatan yang saling terhubung, di mana setiap neuron berfungsi untuk menerima input, melakukan transformasi matematis terhadap input tersebut, lalu menghasilkan output yang diteruskan ke neuron pada lapisan berikutnya. Struktur jaringan ini memungkinkan Artificial Neural Network untuk mempelajari representasi data yang kompleks melalui proses propagasi informasi antar lapisan [29].

Fleksibilitas Artificial Neural Network terletak pada kemampuannya menyesuaikan bobot (weights) dan bias selama proses pelatihan (training). Dalam tahap ini, jaringan diberikan pasangan data input dan output, kemudian melakukan penyesuaian parameter untuk meminimalkan selisih antara prediksi model dan nilai aktual. Mekanisme inilah yang memungkinkan Artificial Neural Network untuk mempelajari pola non-linear yang sulit ditangkap oleh

model statistik tradisional, sehingga menjadikannya salah satu algoritma yang banyak digunakan dalam berbagai bidang, termasuk prediksi penjualan, pengenalan pola, dan pengolahan bahasa alami [29] [31].



Gambar 2. 3 Arsitektur Artificial Neural Network

Proses kerja Artificial Neural Network dapat dijelaskan secara matematis melalui fungsi aktivasi dan penjumlahan berbobot dari input. Setiap neuron dalam jaringan menerima input dari neuron sebelumnya, mengalikannya dengan bobot tertentu, menjumlahkannya, dan melewatkannya ke fungsi aktivasi untuk menghasilkan output. Rumus dasar dari satu unit neuron dalam Artificial Neural Network dapat dinyatakan sebagai berikut.

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

Formula 2. 2 Rumus Artificial Neural Network

Sumber: [28]

Deskripsi symbol:

1. y = Keluaran dari neuron (output)
2. f = Fungsi aktivasi (misalnya sigmoid, ReLU, tanh)
3. x_i = Nilai input ke-I
4. w_i = Bobot yang dikaitkan dengan input ke-i
5. b = Bias dari neuron
6. n = Jumlah total input ke neuron

2.3.4 Tuning

Tuning merupakan komponen fundamental dalam analitik prediktif karena performa model sangat dipengaruhi oleh konfigurasi arsitektural dan parameter dinamis yang ditentukan sebelum proses pelatihan dimulai. Dalam konteks reservoir computing maupun model Machine Learning / Deep Learning lainnya, pemilihan nilai hyperparameter seperti learning rate, jumlah pohon pada ensemble methods, jumlah neuron pada Artificial Neural Network, hingga panjang urutan pada LONG SHORT-TERM MEMORY, berperan penting dalam menentukan akurasi serta efisiensi model. Teknik konvensional seperti grid search dan random search banyak digunakan karena sederhana, namun keduanya cenderung tidak efisien dan memerlukan biaya komputasi tinggi ketika ruang pencarian parameter sangat besar atau menunjukkan karakteristik non-linear [17].

Untuk mengatasi keterbatasan tersebut, penelitian terkini mulai mengadopsi pendekatan optimasi yang lebih adaptif, seperti Bayesian optimization dalam tugas regresi dan prediksi berbasis data kompleks. Pendekatan ini memungkinkan eksplorasi ruang hyperparameter secara lebih terarah dengan memanfaatkan fungsi probabilistik untuk menilai kandidat terbaik. Lebih lanjut, strategi multi-objektif juga telah digunakan untuk menyeimbangkan beberapa kriteria performa sekaligus, misalnya antara akurasi prediksi dengan efisiensi komputasi, sehingga menghasilkan konfigurasi model yang lebih optimal dalam berbagai kondisi [32].

2.3.6.1 Optuna

Teknik pencarian hyperparameter tradisional seperti grid search dan random search memiliki keterbatasan karena mengeksplorasi ruang parameter secara menyeluruh atau acak, sehingga memerlukan biaya komputasi yang tinggi dan cenderung menghasilkan performa kurang optimal, khususnya pada model yang kompleks seperti Gradient Boosting dan Extreme Gradient Boosting. Untuk mengatasi hal tersebut, pendekatan

modern berbasis Bayesian optimization mulai digunakan karena mampu mengeksplorasi ruang hyperparameter dengan lebih cerdas dan adaptif [33].

Optuna merupakan salah satu kerangka kerja *hyperparameter optimization* yang dirancang untuk melakukan pencarian parameter secara otomatis dan efisien. Optuna menggabungkan Bayesian Optimization dengan metode optimasi canggih lainnya untuk menentukan konfigurasi hyperparameter terbaik pada suatu model. Kerangka ini mampu melakukan eksplorasi adaptif, sehingga menghasilkan konvergensi lebih cepat dan model dengan akurasi lebih tinggi dibandingkan strategi optimasi konvensional. Penelitian menunjukkan bahwa Optuna secara konsisten mengungguli metode pencarian tradisional baik dari sisi efisiensi maupun kualitas hasil prediksi [33].

2.3.6.2 Hyperband

Hyperband merupakan salah satu metode optimasi hyperparameter yang dirancang untuk meningkatkan efisiensi serta akurasi dalam pencarian konfigurasi terbaik pada model machine learning. Tidak seperti metode random search atau grid search, Hyperband menggunakan strategi pengalokasian sumber daya secara adaptif berdasarkan jumlah iterasi yang berbeda. Pendekatan ini memungkinkan evaluasi cepat pada banyak konfigurasi awal tanpa harus menghabiskan biaya komputasi secara berlebihan, sehingga mempercepat proses pencarian nilai hyperparameter yang optimal [34].

Meskipun Hyperband dikenal memiliki waktu eksekusi relatif singkat, kelemahan utamanya terletak pada potensi pemilihan konfigurasi hyperparameter akhir yang kurang optimal. Hal ini terjadi karena mekanisme seleksi konfigurasi dilakukan secara acak pada tahap awal. Namun, dengan menggabungkan fleksibilitas jumlah konfigurasi awal yang dievaluasi serta strategi sampling yang lebih selektif, Hyperband tetap mampu mengurangi waktu pelatihan sekaligus menekan biaya komputasi dibandingkan metode konvensional. Oleh karena itu, Hyperband menjadi

salah satu pendekatan yang banyak digunakan dalam penelitian modern sebagai alternatif optimasi hyperparameter yang lebih efisien [34].

2.4 Tools/software yang digunakan

2.4.1 Jupyter Notebook

Jupyter Notebook merupakan salah satu perangkat yang banyak digunakan dalam penelitian modern karena kemampuannya mengintegrasikan kode imperatif dan metadata deklaratif dalam satu format yang unik. Notebook ini memungkinkan pemisahan yang jelas antara lapisan aplikasi dan koordinasi, sekaligus menyediakan antarmuka terpadu untuk keduanya. Dengan demikian, Jupyter Notebook menempati posisi tengah di antara dua kelas alat pemodelan alur kerja, yaitu bahasa koordinasi tingkat tinggi dan pustaka komputasi terdistribusi tingkat rendah [35].

Selain keunggulan tersebut, Jupyter Notebook juga dilengkapi dengan antarmuka berbasis web yang kaya fitur serta ramah pengguna. Hal ini menjadikannya standar de facto di berbagai bidang penelitian, karena banyak pakar domain yang telah terbiasa menggunakan platform ini dalam kegiatan analitik data maupun pemodelan. Dengan fleksibilitas dan adopsi luas tersebut, Jupyter Notebook berperan penting dalam mempercepat kolaborasi penelitian sekaligus meningkatkan reproduibilitas eksperimen [35].

2.4.2 Python

Python merupakan bahasa pemrograman open source serbaguna yang banyak digunakan dalam pengembangan perangkat lunak, analisis data, hingga riset akademik. Sebagai bahasa umum, Python memiliki kemampuan analisis variabel dan liveness checking untuk mendeteksi potensi ketidakstabilan yang dapat muncul akibat eksekusi ulang sel dalam lingkungan seperti Jupyter Notebook. Fitur ini menjadikan Python relevan tidak hanya untuk kebutuhan pemrograman dasar, tetapi juga dalam konteks penelitian berbasis notebook interaktif yang membutuhkan keterandalan dan keakuratan dalam pengelolaan data [35].

Kelebihan Python terletak pada kesederhanaan sintaks yang mudah dipahami, menjadikannya pilihan ideal baik bagi pemula maupun pengguna tingkat lanjut. Ekosistem pustaka yang luas seperti NumPy, SciPy, Pandas, dan OpenCV mendukung berbagai kebutuhan mulai dari pengolahan data, *machine learning*, hingga pemrosesan citra [36]. Dukungan komunitas yang kuat, ketersediaan dokumentasi serta *framework* pihak ketiga, dan kemampuannya berintegrasi dengan bahasa lain semakin memperluas cakupan penggunaan Python. Fleksibilitas ini menjadikan Python sebagai bahasa yang sangat penting di bidang akademik maupun industri, serta berkontribusi pada adopsinya secara luas di berbagai disiplin ilmu [36].

2.4.3 Github

GitHub merupakan platform berbasis cloud yang mendukung pengelolaan kode sumber secara kolaboratif dengan memanfaatkan sistem version control Git. Platform ini memungkinkan penyimpanan, pelacakan, dan pengelolaan versi program Python maupun bahasa pemrograman lainnya dalam satu lingkungan yang homogen, sehingga memudahkan peneliti untuk melakukan eksperimen secara terstruktur. Repositori GitHub berfungsi sebagai pusat penyimpanan kode yang dapat diakses secara publik, serta dapat diintegrasikan dengan berbagai layanan tambahan. Dengan demikian, GitHub menjadi sarana penting untuk menyimpan, membagikan, sekaligus mendokumentasikan hasil penelitian berbasis data dan pemrograman [37].

Selain sebagai repositori publik, GitHub juga berperan dalam membangun keterhubungan antara publikasi akademik dengan implementasi teknis. Namun, praktik pengelolaan keterkaitan antara repositori GitHub dengan artikel ilmiah masih relatif belum terstandarisasi. Studi menunjukkan bahwa topik yang paling sering muncul dalam repositori akademik adalah machine learning, dan repositori tersebut sebagian besar dimiliki oleh individu yang berafiliasi dengan institusi pendidikan [38]. Dalam implementasi praktis, repositori Python yang tersimpan di GitHub juga dapat langsung digunakan untuk proses deployment aplikasi berbasis Streamlit, sehingga memungkinkan

hasil penelitian tidak hanya terdokumentasi secara ilmiah tetapi juga dapat diakses secara interaktif oleh pengguna akhir.

2.4.4 Streamlit

Streamlit merupakan kerangka kerja open-source yang dirancang untuk membangun aplikasi web interaktif berbasis data dengan cepat menggunakan bahasa pemrograman Python. Salah satu keunggulan utama Streamlit adalah dukungan bawaan terhadap berbagai pustaka visualisasi populer seperti Matplotlib, Plotly, dan pustaka pemrosesan data seperti Pandas. Dengan antarmuka yang sederhana, peneliti maupun praktisi dapat mengubah skrip Python menjadi aplikasi web hanya dengan sedikit modifikasi kode. Hal ini menjadikan Streamlit sangat populer baik bagi pengguna yang baru mengenal pemrograman maupun bagi pengembang berpengalaman yang ingin membangun prototipe aplikasi data secara efisien [39].

Selain itu, Streamlit dilengkapi dengan berbagai elemen antarmuka yang telah terdefinisi sebelumnya, seperti *sidebar*, *input widgets*, hingga dukungan untuk aplikasi multipage. Integrasi erat dengan pustaka Python memungkinkan pembuatan tabel interaktif, grafik dinamis, serta visualisasi data kompleks tanpa memerlukan pengetahuan mendalam tentang pengembangan web tradisional. Dengan komunitas yang aktif dan pengembangan berkelanjutan, Streamlit menjadi salah satu alat penting dalam mendukung penelitian maupun implementasi industri yang membutuhkan aplikasi analitik berbasis web secara cepat dan interaktif [39].

2.5 Metrik Evaluasi

2.5.1 Mean Absolute Error

Mean Absolute Error mengukur rata-rata perbedaan atau selisih absolut antara nilai prediksi dan nilai aktual. Metrik ini memberikan gambaran tentang seberapa jauh prediksi dari nilai yang sebenarnya tanpa mempertimbangkan arah kesalahan. Semakin kecil nilai Mean Absolute Error, semakin baik model dalam memprediksi [40].

$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Formula 2. 3 Rumus Mean Absolute Error

[40]

2.5.2 Mean Absolute Percentage Error

Mean Absolute Percentage Error mengukur seberapa besar kesalahan relatif antara nilai prediksi dan nilai aktual dalam bentuk persentase [41]. Mean Absolute Percentage Error sering digunakan dalam karena memberikan gambaran tentang akurasi prediksi dalam konteks relative [40].

$$\frac{1}{m} \sum_{i=1}^m \left| \frac{Y_i - X_i}{Y_i} \right|$$

Formula 2. 4 Rumus Mean Absolute Percentage Error

[40]

2.5.3 Root Mean Square Error

Root Mean Square Error adalah akar kuadrat dari Mean Square Error c dan memberikan ukuran seberapa besar kesalahan prediksi dalam unit yang sama dengan data asli, sehingga sering digunakan ketika penting untuk menghindari prediksi yang sangat salah [40].

$$\sqrt{\frac{1}{m} \sum_{i=1}^m (X_i - Y_i)^2}$$

Formula 2. 5 Rumus Root Mean Square Error

[40]

2.5.4 R²

R² mengukur sejauh mana variansi dalam data aktual dapat dijelaskan oleh model. Nilai R² berkisar antara 0 hingga 1, dengan nilai lebih tinggi menunjukkan model yang lebih baik [41]. Nilai R² yang tinggi menunjukkan bahwa model dapat menjelaskan sebagian besar variasi dalam data [40].

$$1 - \frac{\sum_{i=1}^m (X_i - Y_i)^2}{\sum_{i=1}^m (X_i - \bar{Y})^2}$$

Formula 2. 6 Rumus R^2

[40]



UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA