

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Tabel 2. 1 Tabel Penelitian Terdahulu

No	Sumber	Tahun	Penulis	Perfoma Model
1	A Machine Learning Approach in Predicting Student's Academic Performance Using Artificial Neural Network [11]	2023	Rufai Aliyu Yauri, Hassan Umar Suru, James Afrifa, Hannatu G. Moses	(ANN) Akurasi: 97.36%,
2	CatBoost: An Ensemble Machine Learning Model for Prediction and Classification of Student Academic Performance [12]	2021	Abhisht Joshi, Pranay Saggar, Rajat Jain, Moolchand Sharma	(Catboost) Akurasi: 92.27%
3	Predicting Academic Success of College Students Using Machine Learning Techniques [13]	2024	Jorge Humberto Guanin-Fajardo, Javier Guaña-Moya, Jorge Casillas	(XGBoost) : Akurasi: 79.49%, Sensitivitas: 0.8425, Presisi: 0.8214 (Random Forest) : Akurasi: 79.25%, Sensitivitas: 0.8444, Persis: 0.8205
4	Factors Influencing Academic Performance and Dropout Rates [14]	2024	Ádám Kocsis, Gyöngyvér Molnár	(SVM) : Akurasi: 94.69 (Random Forest) : Akurasi: 79.25%, (PART) : Akurasi: 92.3

5	K-means Clustering for Predicting Student Performance [15]	2021	Ray Mondow Sagala	(K-Means) Akurasi: 93%, Presisi: 96%, Recall: 92%
6	A Deep Learning Model to Predict Student Learning Outcomes in LMS Using CNN and LSTM [16]	2021	Mohammed M. Ali, Iman M. Yousaf, Adel Alkhali	(CNN-LSTM) Akuras: 95.96%, RMSE: 27.56, 29.92%
7	Application Decision Tree Predicting Student Performance in Online Learning [17]	2021	Amir Shahzad, Aliya Ijaz, Nida Hassan	(BayesNet-D) Cross-validation: 90,24%
8	Using Data Mining Techniques to Predict Student Performance [18]	2020	Hanan Abdullah Mengash	(ANN) Accuracy: 79.22%
9	Student-Performulator: Student Academic Performance Using Hybrid Deep Neural Network [19]	2021	Bashir Khan Yousafzai, Sher Afzal Khan, Taj Rahman, Inayat Khan, Inam Ullah, Ateeq Ur Rehman, Mohammed Baz, Habib Hamam, Omar Cheikhrouhou	(Random Forest) Accuracy: 85.33% (CNN) Accuracy: 85.33% (LSTM) Accuracy: 86,16%

10	Analysis of Factors of Academic Readiness Related to Socio-Economic and Integrated Support Towards Academic Achievement for Students18 ANALYSIS FACTORS OF[20]	2020	Ade Maulia Firdani, Lutfi Rachman, Marindra Firmansyah	(CNN) F1 Score for Aspect Classification: 92.23% F1 Score for Sentiment Classification: 95.13%
----	--	------	--	---

Penelitian-penelitian sebelumnya telah menunjukkan pentingnya integrasi teknologi data dan pemahaman terhadap faktor eksternal dalam mengevaluasi performa akademik mahasiswa. Penerapan Big Data Analytics dalam konteks pendidikan tinggi terbukti membantu institusi dalam merancang system intervensi berbasis data untuk meningkatkan capaian akademik [1] [4]. Pendekatan ini memungkinkan pengambilan Keputusan berbasis bukti melalui pemantauan data akademik secara real-time [21].

Selain itu, studi lain juga menyoroti bahwa faktor eksternal seperti kondisi sosial-ekonomi, dukungan keluarga, dan lingkungan belajar sangat berpengaruh terhadap performa akademik mahasiswa [2] [3]. Penelitian-penelitian ini menunjukkan bahwa tekanan ekonomi, keterbatasan fasilitas pendukung, serta kurangnya lingkungan belajar yang kondusif dapat meningkatkan risiko dropout dan menurunkan motivasi belajar [11].

Prediksi nilai akhir juga telah dieksplorasi lebih lanjut melalui penggunaan Neural Network dan Support Vector Machine, yang keduanya terbukti memiliki akurasi tinggi dalam berbagai eksperimen [7] [22]. Studi terbaru juga mendukung hal ini dengan hasil validasi akurasi model prediksi berbasis Neural Network dan Support Vector Machine [11].

Di sisi lain, aspek kehadiran mahasiswa sebagai indicator performa akademik juga menjadi aspek utama dalam penelitian. Tingkat kehadiran ditemukan memiliki korelasi positif terhadap capaian akademik, dan dapat dimanfaatkan sebagai fitur penting dalam pengembangan system prediksi akademik

[9]. Penelitian tambahan menyebutkan bahwa data kehadiran dapat berfungsi sebagai input awal dalam model evaluasi kinerja [23].

Perkembangan dalam bidang Explainable Artificial Intelligence memberikan kontribusi terutama untuk menjelaskan bagaimana sistem prediktif menghasilkan Keputusan dalam konteks pendidikan. Kerangka kerja seperti LIME memungkinkan visualisasi dan interpretasi logis terhadap klasifikasi model yang penting untuk menjamin transparansi dan akuntabilitas algoritma yang digunakan [10] [24] [1].

Meskipun penelitian-penelitian terdahulu telah mengeksplorasi berbagai pendekatan prediktif untuk menilai performa akademik mahasiswa, masih terdapat beberapa kekurangan yang belum banyak dikaji. Beberapa studi hanya menggunakan satu atau dua algoritma tanpa membandingkan performa antar model secara sistematis [15][16][18][19].

. Selain itu, belum banyak penelitian yang menerapkan strategi tuning hyperparameter secara komprehensif untuk mengoptimalkan hasil prediksi, terutama dengan mempertimbangkan keterbatasan sumber daya komputasi.

Oleh karena itu, penelitian ini bertujuan untuk mengisi gap tersebut dengan membandingkan performa beberapa algoritma Machine Learning dan Deep Learning (XGBoost, Random Forest, ANN, dan LSTM), serta menerapkan teknik hyperparameter tuning (Optuna dan Hyperband) guna memperoleh model prediksi akademik yang optimal dan efisien. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi baru dalam hal keakuratan, efisiensi, dan keterbandingan model prediktif berbasis data akademik dan faktor kehadiran.

2.2 Tinjauan Teori

2.2.1 Machine Learning dalam Pendidikan

Machine Learning merupakan salah satu cabang dari kecerdasan buatan yang mampu mempelajari pola dari data dan membuat prediksi tanpa pemrograman eksplisit. Dalam dunia pendidikan, Machine Learning telah dimanfaatkan untuk menganalisis performa mahasiswa, mengklasifikasikan tingkat risiko dropout, hingga memprediksi nilai akademik berdasarkan

variabel historis dan perilaku belajar [6] [7]. Machine Learning digunakan dalam berbagai metode seperti Decision Tree, Support Vector Machine, hingga Artificial Neural Network yang memungkinkan pengolahan data kompleks dari sistem informasi akademik [6]. Keunggulan Machine Learning terletak pada kemampuannya menyaring pola tersembunyi dalam dataset pendidikan yang sangat besar dan beragam [5].

2.2.2 Explainable Artificial Intelligence

Seiring meningkatnya penggunaan algoritma kompleks seperti deep learning, muncul kebutuhan untuk memastikan sistem tersebut dapat dipahami manusia. Explainable Artificial Intelligence adalah pendekatan yang bertujuan menjelaskan proses pengambilan keputusan dari sistem Artificial Intelligence yang bersifat black box, terutama dalam konteks yang mempengaruhi kehidupan manusia seperti pendidikan [7].

Pendekatan Explainable Artificial Intelligence memastikan bahwa prediksi yang dihasilkan oleh sistem Artificial Intelligence dapat dijelaskan secara logis, akuntabel, dan transparan bagi pengguna akhir seperti dosen, mahasiswa, maupun lembaga pendidikan [10].

2.2.3 Big Data Analytics dalam Pendidikan Tinggi

Seiring Big Data Analytics adalah proses mengekstraksi informasi dan pengetahuan dari kumpulan data besar dan kompleks. Dalam konteks pendidikan tinggi, pendekatan ini digunakan untuk menganalisis perilaku mahasiswa, mengevaluasi efektivitas pembelajaran, serta mendukung pengambilan keputusan institusional [1] [21].

Penerapan Data dalam pendidikan memungkinkan visualisasi performa akademik mahasiswa secara real-time dan membantu institusi mengidentifikasi pola risiko akademik seperti penurunan nilai atau potensi dropout [4]. Strategi ini juga mendukung integrasi sistem digital seperti Learning Management System dengan data kehadiran, nilai, dan aktivitas mahasiswa [10].

2.2.4 Faktor Internal Yang Mempengaruhi Prestasi Akademik

Seiring Prestasi akademik mahasiswa tidak hanya dipengaruhi oleh faktor internal seperti motivasi dan kemampuan kognitif, tetapi juga oleh faktor eksternal seperti latar belakang sosial-ekonomi, dukungan keluarga, fasilitas kampus, dan lingkungan sosial [2] [3] [25].

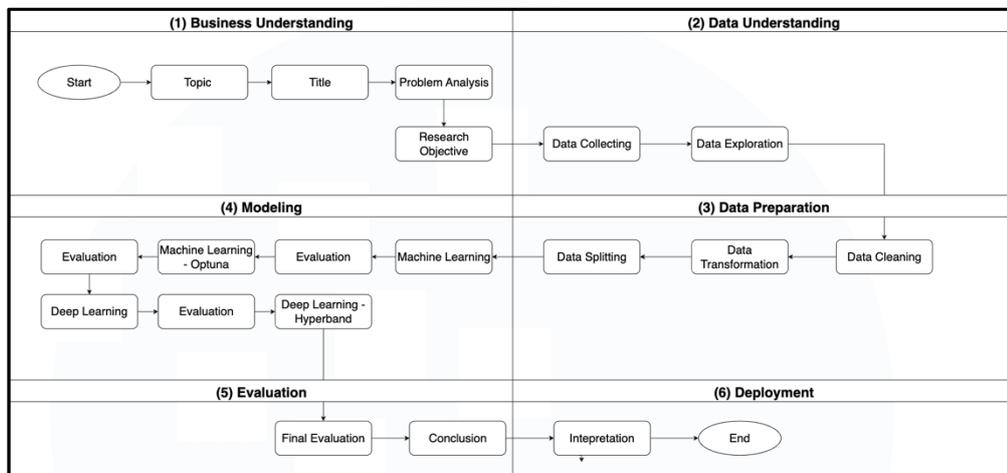
Kondisi ekonomi keluarga, misalnya, dapat mempengaruhi kemampuan mahasiswa dalam mengakses sumber daya belajar, termasuk perangkat teknologi dan koneksi internet yang memadai untuk pembelajaran daring. Selain itu, kehadiran di kelas juga terbukti berperan signifikan dalam menentukan capaian akademik mahasiswa [9].

2.3 Teori Framework & Algoritma yang digunakan

2.3.1 CRISP-DM

Seiring meningkatnya penggunaan algoritma kompleks seperti deep learning. CRISP-DM adalah framework yang seringkali digunakan dalam proses data mining dan machine learning, yang menawarkan pendekatan sistematis dan terstruktur untuk menangani proyek data dengan tujuan memaksimalkan efisiensi dan hasil dari proses analisis data. Framework ini dibagi menjadi tiga bagian utama: pre-processing steps, processing steps, dan post-processing steps. Pada bagian pertama, pre-processing steps mencakup Business Understanding, Data Understanding, dan Data Preparation, yang bertujuan untuk memahami masalah bisnis, menganalisis data secara eksploratif, dan mempersiapkan data agar siap digunakan dalam pemodelan. Bagian kedua, processing steps, terdiri dari tahap Modeling dan Evaluation, yang fokus pada penerapan teknik pemodelan untuk menghasilkan model prediktif dan mengevaluasi kinerja model tersebut untuk memastikan bahwa hasilnya sesuai dengan tujuan proyek. Bagian terakhir, post-processing steps, berfokus pada penerapan hasil evaluasi model dalam konteks bisnis untuk mengembangkan strategi dan kebijakan yang lebih efektif sesuai dengan

kebutuhan



Gambar 2. 1 Kerangka kerja CRISP-DM

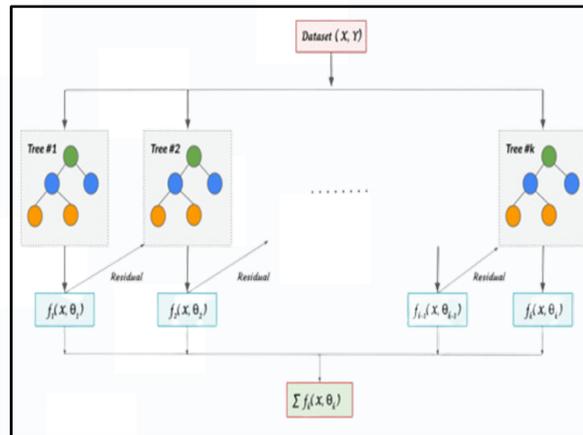
bisnis. Dengan penggunaan yang terstruktur ini, peneliti akan lebih mudah untuk menguraikan proses analisis yang dilakukan. Setiap tahapan dalam CRISP-DM memberikan panduan yang jelas mengenai aktivitas yang perlu dilakukan, tujuan dari setiap aktivitas, serta keterkaitannya dengan tahap sebelumnya maupun tahap berikutnya.

2.3.2 Extreme Gradient Boost

Extreme Gradient Boosting atau yang dikenal dengan XGBoost adalah algoritma machine learning yang sangat kuat dan efisien, banyak digunakan dalam tugas regresi maupun klasifikasi. Algoritma ini merupakan pengembangan dari metode Gradient Boosting, yang menggabungkan banyak pohon Keputusan [15] secara bertahap untuk meningkatkan akurasi prediksi secara keseluruhan

XGBoost dikenal karena efisiensinya dalam komputasi, dukungan terhadap regularisasi L1 dan L2, serta kemampuan pemrosesan paralel, yang membuatnya unggul dalam menangani dataset besar dan kompleks. Selain itu, XGBoost memiliki fitur pruning dan teknik pengurangan overfitting yang menjadikannya salah satu algoritma paling andal dalam berbagai kompetisi data science [26].

Pertama kali diperkenalkan oleh Tianqi Chen dan Carlos Guestrin, XGBoost telah menjadi standar de facto dalam pemodelan prediktif berbasis ensemble karena kinerja dan fleksibilitasnya [23].



Gambar 2. 2 Arsitektur Extreme Gradient Boost

sumber: [26]

Proses XGBoost dimulai dengan sebuah dataset (X, Y) , di mana pohon pertama dalam ensemble memprediksi nilai target. Kesalahan prediksi residual dari pohon pertama ini dihitung dan dijadikan target untuk pohon berikutnya. Pohon-pohon berikutnya terus mempelajari residual dari model sebelumnya, sehingga setiap pohon yang ditambahkan memperbaiki akurasi model secara bertahap. Setelah beberapa pohon terbentuk, prediksi akhir diperoleh dengan menjumlahkan semua prediksi dari setiap pohon, menghasilkan model yang semakin akurat dalam memprediksi target.

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

Formula 2. 1 Rumus Extreme Gradient Boost

Sumber: [26]

Deskripsi symbol:

1. \hat{y} = Jumlah prediksi akhir dari setiap pohon

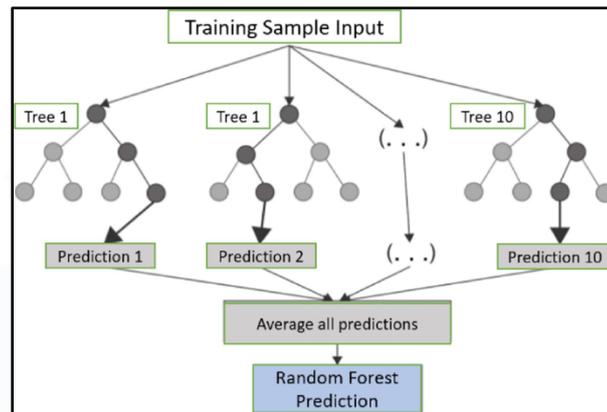
2. Σ = Penjumlahan (kontribusi setiap pohon)
3. k = Indeks pohon ke- k (adanya penambahan untuk setiap pohon)
4. f_k = Fungsi prediksi hasil dari pohon ke- k (individual)
5. F = Ruang fungsi (mengandung semua pohon yang sudah dilatih)
6. x_i = Vektor fitur dari samle data ke- i (data yang diprediksi)

2.3.3 Random Forest

Seiring meningkatnya penggunaan algoritma kompleks seperti deep learning. Random Forest adalah metode pembelajaran ensemble yang menggunakan algoritma yang sama atau berbeda secara berulang, kemudian digabungkan ke dalam satu model yang lebih kuat dibandingkan model aslinya. Pendekatan ini memungkinkan prediksi dibuat berdasarkan jumlah bobot dari berbagai pohon keputusan, sehingga hasil akhir menjadi lebih akurat dibandingkan hanya menggunakan satu pohon saja.

Keunggulan utama dari Random Forest adalah ketahanannya terhadap perubahan pada dataset. Jika suatu perubahan berdampak pada satu pohon, hal ini tidak akan memengaruhi keseluruhan hutan pohon yang telah dibentuk. Random Forest terdiri dari lima langkah utama: (1) memilih sejumlah K data secara acak dari dataset pelatihan, (2) membangun satu pohon keputusan berdasarkan K data tersebut, (3) mengulangi proses ini sebanyak N kali untuk membentuk N pohon keputusan, (4) setiap pohon akan memberikan prediksi nilai Y untuk data baru, dan (5) rata-rata dari seluruh prediksi pohon akan menghasilkan prediksi akhir. Rata-rata prediksi ini terbukti lebih stabil dan mengurangi risiko overfitting yang sering terjadi pada model pohon tunggal.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 2. 3 Arsitektur Random Forest
Sumber: [27]

Gambar 2.2 menunjukkan ilustrasi arsitektur Random Forest yang terdiri atas sejumlah decision tree yang dilatih secara paralel dengan subset data pelatihan yang berbeda-beda. Setiap pohon menghasilkan prediksi secara independen, lalu hasil dari semua pohon tersebut dirata-ratakan untuk memperoleh prediksi akhir

$$H(x) = \frac{1}{M} \sum_{m=1}^M h_m(x)$$

Formula 2. 2 Rumus Random Forest
Sumber: [27]

Deskripsi symbol:

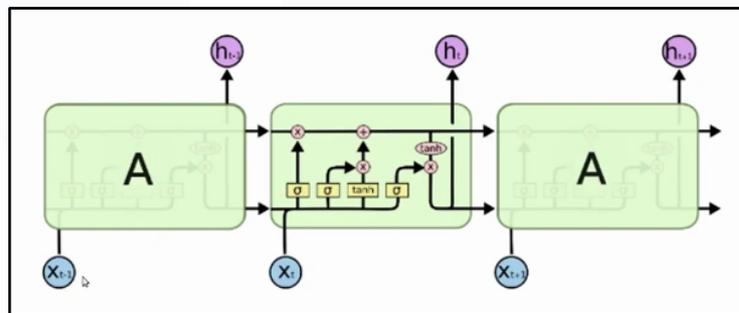
1. $H(x)$ = Prediksi akhir dari model Random Forest
2. M = Jumlah total pohon (decision trees) dalam ensemble
3. \sum = Simbol penjumlahan dari semua pohon f_k = Fungsi prediksi hasil dari pohon ke-k (individual)
4. m = Indeks pohon ke- m (dari 1 sampai M)
5. $h_m(x)$ = Fungsi prediksi dari pohon ke- m terhadap input x

2.3.4 Long Short Time Memory

LSTM banyak digunakan dalam pemrosesan bahasa alami seperti analisis teks dan pengenalan suara karena kemampuannya dalam mengingat dan

mempertahankan informasi jangka panjang. Salah satu kekuatan utama LSTM adalah adanya sel memori internal yang memungkinkan model untuk menyimpan informasi penting dari langkah sebelumnya dalam urutan data.

Belakangan ini, LSTM semakin banyak digunakan dalam prediksi data deret waktu, seperti peramalan cuaca, harga saham, dan data sensor. Dalam arsitekturnya, LSTM terdiri dari tiga gerbang utama yang mengatur aliran informasi, yaitu: (1) forget gate untuk melupakan informasi yang tidak relevan, (2) input gate untuk menerima informasi baru, dan (3) output gate untuk menghasilkan keluaran berdasarkan memori sel yang diperbarui.



Gambar 2. 4 Arsitektur Long Short-Term Memory

Sumber: [28]

Setiap unit LSTM menggunakan perhitungan matematis untuk mengatur informasi yang mengalir melalui jaringan. Proses internal dari LSTM dapat dijelaskan melalui rumus berikut:

$$h_t = o_t \cdot \tanh \tanh (C_t)$$

Formula 2. 3 Rumus Long Short-Term Memory

Sumber: [28]

Deskripsi symbol:

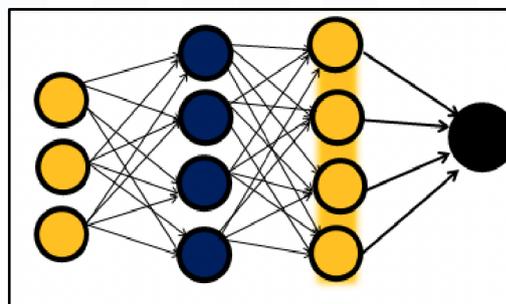
1. h_t = Hidden state saat waktu ke-t. Ini adalah output utama dan dikirim ke layer selanjutnya atau timestep berikutnya.

2. o_t = Output gate saat waktu ke- t . Ini adalah vektor bernilai antara 0 dan 1 yang menentukan seberapa besar informasi dari memori akan dikeluarkan.
3. C_t = Cell state di waktu $-t$, menyimpan memori jangka panjang.
4. Tanh = Fungsi aktivasi untuk menormalkan nilai C_t antara -1 dan 1.

2.3.5 Artificial Neural Network

Artificial Neural Network merupakan salah satu teknik yang populer dalam bidang Educational Data Mining. ANN dirancang untuk meniru cara kerja otak manusia dalam menyelesaikan masalah kompleks. Model ini terdiri dari sekumpulan unit yang menerima masukan berbobot, kemudian mengolahnya dan menghasilkan keluaran. Struktur ini memungkinkan ANN untuk mempelajari pola dan hubungan yang tidak secara eksplisit didefinisikan dalam data.

ANN juga digunakan karena kemampuannya dalam mendeteksi semua kemungkinan interaksi antar variabel serta belajar dari jumlah data yang terbatas. Dalam sebuah studi sebelumnya, model ANN terbukti mengungguli beberapa teknik klasifikasi konvensional dalam mengklasifikasikan calon mahasiswa yang diterima dan tidak diterima. Selain itu, algoritma pembelajaran berbasis ANN dilaporkan memiliki akurasi prediksi yang lebih baik dibandingkan model regresi linier klasik, terutama dalam memprediksi nilai akademik siswa sekolah menengah.



Gambar 2. 5 Arsitektur Artificial Neural Network

Sumber: [28]

Proses kerja ANN 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 ANN dapat dinyatakan sebagai berikut:

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

Formula 2. 4 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

2.3.6 Tuning

Tuning atau penyesuaian hyperparameter adalah proses pencarian nilai optimal untuk parameter-parameter yang mengontrol proses pembelajaran dalam model machine learning atau deep learning. Proses ini bertujuan untuk mengoptimalkan performa model dengan cara mengatur parameter-parameter penting, seperti laju pembelajaran, jumlah lapisan tersembunyi, atau ukuran batch, untuk mencapai akurasi prediksi yang lebih baik [29]. Tuning hyperparameter dilakukan dengan cara yang sistematis, seringkali menggunakan teknik seperti grid search atau random search, untuk menemukan kombinasi parameter yang menghasilkan performa terbaik

berdasarkan metrik evaluasi yang digunakan. Meskipun demikian, proses ini memerlukan perhatian khusus karena parameter yang dipilih harus sesuai dengan karakteristik data dan tujuan analisis yang ingin dicapai, agar model benar-benar teroptimasi dengan baik [30]. Dalam penelitian ini, terdapat 2 teknik tuning yang akan dilakukan, yaitu:

2.3.6.1 Optuna

Optuna adalah metode optimasi hyperparameter otomatis yang dirancang untuk menyederhanakan pencarian hyperparameter optimal pada model machine learning dan deep learning. Metode ini bekerja dengan ruang pencarian yang fleksibel dan adaptif, memungkinkan cakupan rentang yang lebih luas dalam pencarian hyperparameter. Optuna menerapkan strategi pencarian berbasis Tree-structured Parzen Estimator yang secara iteratif mengevaluasi dan mengadaptasi kombinasi hyperparameter untuk meminimalkan kesalahan pada setiap langkah, sehingga meningkatkan akurasi prediksi model [44]. Dengan dukungan integrasi dengan pustaka visualisasi seperti Matplotlib dan Seaborn, Optuna memfasilitasi pemantauan kinerja model serta analisis data secara lebih komprehensif.

2.3.6.2 Hyperband

Hyperband adalah metode optimasi hyperparameter yang dirancang untuk meningkatkan efisiensi dan akurasi pencarian konfigurasi terbaik dalam model machine learning. Berbeda dengan metode pencarian acak atau pencarian berbasis grid, Hyperband menggunakan strategi pengalokasian sumber daya adaptif berdasarkan jumlah iterasi yang berbeda, memungkinkan evaluasi cepat pada berbagai konfigurasi awal. Dengan menggunakan pendekatan sampling yang lebih selektif dan evaluasi iteratif, Hyperband mengurangi biaya komputasi dan waktu yang dibutuhkan untuk menemukan nilai hyperparameter optimal, terutama pada dataset yang besar [31].

Hasil penelitian menunjukkan bahwa Hyperband dapat meningkatkan performa model dibandingkan dengan sampling seragam, tanpa bergantung pada konfigurasi spesifik saat dijalankan. Penggunaan prior berbasis data, yaitu mengacu pada nilai hyperparameter yang baik di pengujian sebelumnya, dapat membantu dalam menemukan konfigurasi yang lebih baik secara konsisten [31]. Skor peringkat yang tinggi pada Hyperband mengindikasikan kinerja yang lebih rendah dibandingkan konfigurasi lain, menunjukkan pentingnya tuning pada hyperparameter tertentu dalam meningkatkan performa model secara keseluruhan [29] [31]

2.4 Teori tentang Tools/ Software yang digunakan

2.4.1 Jupyter Notebook

Jupyter Notebooks adalah alat yang sangat efektif untuk pengembangan dan eksplorasi kode, terutama dalam bahasa pemrograman Python [11]. Dengan kemampuannya mengintegrasikan kode, visualisasi, dan penjelasan dalam satu dokumen, Jupyter Notebooks memungkinkan pengguna untuk menjalankan kode secara langsung, memodifikasi analisis, dan melihat hasilnya secara real-time. Alat ini mendukung proses iteratif dan eksploratif, yang memungkinkan pengguna untuk dengan mudah menguji dan memvisualisasikan berbagai solusi dalam satu lingkungan yang terorganisir. Jupyter Notebooks juga mendukung berbagai Bahasa pemrograman selain Python, seperti R dan Julia, menjadikannya platform yang fleksibel untuk berbagai kebutuhan analisis dan pemrograman [32].

2.4.2 Python

Ilmu data, mencerminkan meningkatnya adopsi pendekatan berbasis data dan pentingnya analisis, manipulasi, serta interpretasi data. Popularitas Python terus meningkat, melampaui bahasa-bahasa dominan sebelumnya seperti C, C++, dan C#, menjadikannya pilihan utama untuk berbagai tugas terkait data [33]. Selain itu, munculnya banyak pustaka sumber terbuka seperti NumPy, Pandas, dan Matplotlib telah mempermudah sekaligus menstandarkan proses pengolahan dan analisis data. Dalam beberapa tahun terakhir, Python telah

menjadi standar de facto untuk aplikasi ilmu data dan pembelajaran mesin [11] [32] [33].

2.4.3 Metrik Evaluasi

2.4.3.1 MAE

MAE 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 MAE, semakin baik model dalam memprediksi.

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

Formula 2. 5 MAE [34]

2.4.3.2 MAPE

MAPE mengukur seberapa besar kesalahan relatif antara nilai prediksi dan nilai aktual dalam bentuk persentase. MAPE sering digunakan dalam karena memberikan gambaran tentang akurasi prediksi dalam konteks relatif.

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

Formula 2. 6 MAPE [35]

2.4.3.3 RMSE

RMSE adalah akar kuadrat dari MSE 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.

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

Formula 2. 7 RMSE [35]

2.4.3.4 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. Nilai R² yang tinggi menunjukkan bahwa model dapat menjelaskan sebagian besar variasi dalam data

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

Formula 2. 8 R² [35]

