

OPTIMASI HYPERPARAMETER XGBOOST UNTUK PERAMALAN PERMINTAAN BERBASIS DATA *TIME SERIES*

Ria Amelia Shinta Putricia Hendra, Anggraini Puspita Sari, Yisti Vita Via

Informatika, UPN Veteran Jawa Timur

Jl. Raya Rungkut Madya, Surabaya, Jawa Timur, Indonesia

anggraini.puspita.if@upnjatim.ac.id

ABSTRAK

Peramalan permintaan merupakan aspek penting dalam pengelolaan penjualan ritel karena membantu perusahaan menyesuaikan ketersediaan barang dengan kebutuhan pasar. Data penjualan ritel umumnya bersifat fluktuatif dan dinamis, sehingga ketidakakuratan peramalan dapat menyebabkan kelebihan atau kekurangan persediaan serta menurunkan efisiensi operasional. Penelitian ini bertujuan untuk meningkatkan akurasi peramalan permintaan berbasis data *time series* menggunakan algoritma XGBoost melalui optimasi *hyperparameter*. Metode penelitian meliputi *preprocessing* data, pemeriksaan *missing value* dan duplikasi, pembentukan fitur *lag* serta variabel diskon, pembagian data berdasarkan urutan waktu, dan pengujian enam kombinasi *hyperparameter* yang terdiri dari *learning rate*, *max depth*, dan *n_estimators*. Evaluasi model dilakukan menggunakan MAE, RMSE, dan MAPE untuk mengukur tingkat kesalahan prediksi. Hasil penelitian menunjukkan bahwa kombinasi terbaik diperoleh pada *learning rate* 0,10, *max depth* 2, dan *n_estimators* 50, dengan MAE sebesar 1,7269, RMSE sebesar 2,1112, dan MAPE sebesar 9,5985%. Model mampu mengikuti pola penjualan secara umum, meskipun masih terdapat selisih pada periode dengan fluktuasi tinggi. Hasil ini menunjukkan bahwa optimasi *hyperparameter* XGBoost efektif digunakan untuk meningkatkan akurasi peramalan permintaan berbasis *time series* serta mendukung proses pengambilan keputusan persediaan.

Kata kunci : peramalan permintaan, XGBoost, optimasi *hyperparameter*, *time series*, *machine learning*

1. PENDAHULUAN

Peramalan permintaan merupakan komponen penting dalam pengelolaan penjualan, khususnya pada sektor ritel yang memiliki karakteristik data dinamis dan fluktuatif. Variasi permintaan dapat dipengaruhi oleh faktor musiman, perilaku konsumen, dan aktivitas promosi, sehingga proses peramalan menjadi lebih kompleks [1].

Peramalan yang tidak akurat dapat menyebabkan ketidakseimbangan antara jumlah persediaan dan kebutuhan pasar, baik berupa kelebihan maupun kekurangan barang. Kondisi tersebut dapat berdampak pada peningkatan biaya operasional serta penurunan tingkat pelayanan kepada konsumen [2].

Oleh karena itu, diperlukan metode peramalan yang mampu menangkap pola data penjualan secara adaptif dan akurat [3].

Permasalahan utama dalam penelitian ini adalah sulitnya menghasilkan prediksi permintaan yang akurat pada data penjualan ritel yang bersifat fluktuatif. Pendekatan berbasis *time series* seperti ARIMA banyak digunakan karena mampu memodelkan pola linier dan tren historis [4].

Namun, metode tersebut memiliki keterbatasan dalam merepresentasikan hubungan nonlinier dan interaksi kompleks antarvariabel [5], sehingga kurang optimal pada data ritel yang memiliki perubahan permintaan cukup dinamis [6] [7].

Kondisi ini menunjukkan perlunya pendekatan lain yang lebih fleksibel dalam memodelkan pola nonlinier pada data penjualan.

Perkembangan *machine learning* memberikan alternatif yang lebih fleksibel dalam memodelkan pola

data yang kompleks dan nonlinier, karena metode ini mampu mempelajari pola dari data serta menyesuaikan proses pemodelan berdasarkan karakteristik data yang digunakan [8] [9].

Salah satu algoritma *machine learning* yang dapat digunakan dalam peramalan adalah XGBoost, yaitu algoritma *gradient boosting* berbasis pohon keputusan yang mampu meningkatkan akurasi prediksi secara iteratif dan stabil [8] [10], serta menunjukkan performa yang kompetitif dalam berbagai studi peramalan [11] [12] [13].

Kinerja XGBoost sangat dipengaruhi oleh pemilihan *hyperparameter* seperti *learning rate*, *max depth*, dan *n_estimators*. Pemilihan parameter yang tidak tepat dapat menyebabkan model mengalami *underfitting* atau *overfitting* [14], sehingga optimasi *hyperparameter* serta penggunaan fitur *lag* menjadi penting dalam meningkatkan akurasi prediksi pada data *time series* [1] [8] [15].

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk meningkatkan akurasi peramalan permintaan berbasis data *time series* menggunakan algoritma XGBoost melalui optimasi *hyperparameter*. Rencana penyelesaian masalah dilakukan melalui beberapa tahapan, yaitu *preprocessing* data, pembentukan fitur *lag* dan variabel diskon, pembagian data berdasarkan urutan waktu, serta pengujian enam kombinasi *hyperparameter* yang terdiri dari *learning rate*, *max depth*, dan *n_estimators*. Hasil prediksi dari setiap skenario kemudian dievaluasi menggunakan metrik MAE, RMSE, dan MAPE untuk menentukan model dengan performa terbaik. Dengan pendekatan tersebut, penelitian ini diharapkan dapat menghasilkan

model peramalan yang lebih akurat dalam memprediksi permintaan penjualan ritel untuk mendukung pengelolaan persediaan.

2. TINJAUAN PUSTAKA

Tinjauan pustaka pada penelitian ini membahas konsep dan penelitian terdahulu yang berkaitan dengan peramalan permintaan berbasis data *time series*. Pembahasan mencakup peramalan permintaan, karakteristik data *time series*, penerapan *machine learning* khususnya XGBoost, optimasi *hyperparameter*, serta metrik evaluasi model. Penelitian terdahulu digunakan sebagai dasar untuk menunjukkan relevansi penggunaan XGBoost dalam peramalan penjualan dan permintaan.

2.1. Penelitian Terdahulu

Beberapa penelitian terdahulu telah membahas penerapan metode statistik maupun *machine learning* dalam peramalan penjualan dan permintaan. Penelitian sebelumnya membandingkan pendekatan klasik dan *machine learning* dalam peramalan serta perencanaan persediaan [4].

Penelitian lain membahas penggunaan algoritma *machine learning* untuk peramalan penjualan [5].

Selanjutnya, penggunaan XGBoost dengan *feature engineering* terbukti dapat meningkatkan hasil peramalan penjualan [8].

XGBoost juga telah diterapkan untuk peramalan permintaan menggunakan pendekatan *machine learning* [10].

Selain itu, XGBoost dapat digunakan pada peramalan volume penjualan berbasis *time series* [11], serta dikembangkan dalam model SSA-XGBoost untuk peramalan permintaan [13].

Berdasarkan penelitian tersebut, XGBoost memiliki potensi dalam peramalan permintaan, namun performanya tetap dipengaruhi oleh pembentukan fitur dan pemilihan *hyperparameter*, sehingga penelitian ini berfokus pada optimasi *hyperparameter* XGBoost menggunakan fitur *lag* dan variabel diskon.

2.2. Peramalan Permintaan dan Time Series

Peramalan permintaan merupakan proses estimasi nilai di masa mendatang berdasarkan data historis untuk mendukung pengambilan keputusan dalam pengelolaan penjualan [2] [4].

Pada sektor ritel, data penjualan umumnya bersifat fluktuatif karena dipengaruhi oleh faktor musiman, tren, serta aktivitas promosi [16] [17].

Pendekatan berbasis *time series* banyak digunakan karena mampu memanfaatkan urutan data historis untuk mengenali pola tren dan musiman pada penjualan [3] [18].

Kemampuan tersebut menjadikan metode *time series* relevan untuk analisis data penjualan ritel, tetapi metode statistik seperti ARIMA masih memiliki keterbatasan dalam menangani hubungan nonlinier dan kompleksitas data [1] [5] [7].

Oleh karena itu, diperlukan pendekatan yang lebih fleksibel agar model dapat merepresentasikan pola permintaan yang dinamis.

2.3. Machine Learning dan XGBoost

Machine learning menawarkan pendekatan yang lebih fleksibel dalam memodelkan hubungan nonlinier dan pola kompleks pada data [1] [5] [19].

Metode ini mampu memanfaatkan interaksi antar variabel secara lebih efektif serta menunjukkan peningkatan performa melalui pendekatan *ensemble* dan *hybrid* dalam berbagai kasus peramalan [20] [21].

Oleh karena itu, penggunaan *machine learning* menjadi semakin relevan dalam peramalan permintaan, terutama pada data penjualan yang bersifat fluktuatif dan dinamis.

XGBoost merupakan algoritma *ensemble learning* berbasis *gradient boosting* yang mampu meningkatkan akurasi model secara iteratif melalui penambahan *decision tree* pada setiap tahap pembelajaran [8] [10].

Algoritma ini dirancang untuk menghasilkan model yang akurat dan efisien dalam komputasi serta mampu menangani data dalam skala besar [22], serta didukung oleh penerapan teknik regularisasi untuk mengontrol kompleksitas model dan mengurangi risiko *overfitting* pada data yang kompleks [15].

Berbagai penelitian juga menunjukkan bahwa XGBoost memiliki performa yang kompetitif dibandingkan metode lain dalam peramalan [11] [13]. Dalam proses pelatihannya, XGBoost menggunakan fungsi objektif yang terdiri dari fungsi kerugian dan fungsi regularisasi untuk mengoptimalkan model, di mana fungsi ini bekerja dengan meminimalkan kesalahan prediksi sekaligus mengontrol kompleksitas model yang dibangun. Secara matematis, fungsi objektif XGBoost dinyatakan sebagai berikut:

$$Obj^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^t \Omega(f_k) \quad (1)$$

dengan:

- $l(y_i, \hat{y}_i^{(t)})$ = fungsi kerugian antara nilai aktual dan prediksi
- $\Omega(f_k)$ = fungsi regularisasi untuk mengontrol kompleksitas model
- f_k = pohon keputusan ke- k
- t = iterasi ke- t
- n = jumlah data yang digunakan dalam proses penelitian

Melalui kombinasi fungsi kerugian dan regularisasi, XGBoost mampu menghasilkan model yang akurat dan stabil dalam menghindari *overfitting*. Pendekatan ini efektif dalam menangani data dengan pola kompleks dan nonlinier. Selain itu, proses pembelajaran bertahap memungkinkan model memperbaiki kesalahan prediksi pada setiap iterasi sehingga meningkatkan performa secara keseluruhan.

2.4. Optimasi Hyperparameter

Optimasi *hyperparameter* merupakan proses penyesuaian parameter model yang ditentukan sebelum proses pelatihan untuk meningkatkan performa model. Pada algoritma XGBoost, parameter seperti *learning rate*, *max depth*, dan *n_estimators* digunakan untuk mengontrol proses pembelajaran, di mana *learning rate* mengatur kontribusi pembaruan model, *max depth* mengontrol kompleksitas *decision tree*, dan *n_estimators* menentukan jumlah iterasi dalam proses *boosting* [13] [14] [15].

Pemilihan parameter yang tidak tepat dapat menyebabkan model menjadi tidak stabil atau mengalami *overfitting*, sehingga diperlukan pengujian beberapa kombinasi parameter untuk memperoleh hasil yang optimal. Penelitian menunjukkan bahwa optimasi *hyperparameter* pada XGBoost dapat meningkatkan akurasi prediksi secara signifikan, terutama pada data yang kompleks dan fluktuatif [8] [19].

Selain itu, pada data *time series*, fitur berbasis *lag* juga sering digunakan untuk merepresentasikan nilai historis sebagai input model dalam proses peramalan [3] [8].

2.5. Evaluasi Model

Evaluasi model merupakan tahap penting dalam proses peramalan untuk mengukur tingkat akurasi hasil prediksi terhadap data aktual. Proses ini bertujuan untuk mengetahui sejauh mana model mampu merepresentasikan pola data yang sebenarnya. Dalam penelitian ini, metrik evaluasi yang digunakan meliputi MAE, RMSE, dan MAPE yang masing-masing memberikan sudut pandang berbeda terhadap kesalahan prediksi [4] [23].

MAE digunakan untuk menghitung rata-rata selisih absolut antara nilai aktual dan nilai prediksi sehingga mudah diinterpretasikan [4].

Metrik ini tidak dipengaruhi oleh arah kesalahan sehingga memberikan gambaran umum terhadap besarnya kesalahan prediksi, di mana nilai MAE yang lebih kecil menunjukkan bahwa hasil prediksi semakin mendekati nilai aktual. Secara matematis, MAE dinyatakan sebagai berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

dengan:

- y_i = nilai aktual pada periode ke- i
- $\hat{y}_i$ = nilai prediksi pada periode ke- i
- n = jumlah data

RMSE digunakan untuk mengukur kesalahan prediksi dengan memberikan penalti yang lebih besar terhadap kesalahan yang tinggi [4].

Metrik ini menghitung akar dari rata-rata kuadrat selisih antara nilai aktual dan nilai prediksi sehingga lebih sensitif terhadap penyimpangan besar dalam data, di mana nilai RMSE yang lebih kecil menunjukkan performa model yang lebih baik dalam

memprediksi data. Secara matematis, RMSE dinyatakan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3)$$

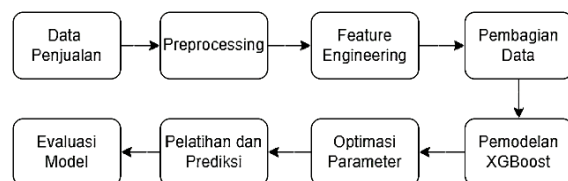
MAPE digunakan untuk mengukur kesalahan prediksi dalam bentuk persentase terhadap nilai aktual sehingga lebih mudah diinterpretasikan [23].

Metrik ini memberikan gambaran kesalahan relatif terhadap skala data sehingga memudahkan perbandingan antar model, di mana nilai MAPE yang lebih kecil menunjukkan tingkat akurasi yang lebih baik. Secara matematis, MAPE dinyatakan sebagai berikut:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (4)$$

3. METODE PENELITIAN

Metode penelitian ini digunakan untuk membangun model peramalan permintaan menggunakan algoritma XGBoost melalui tahapan yang terstruktur. Proses dimulai dari pengolahan data penjualan, dilanjutkan dengan *preprocessing* dan *feature engineering* untuk menyesuaikan data dengan kebutuhan model, serta pembagian data dan pembangunan model dengan beberapa skenario *hyperparameter*. Model yang dihasilkan kemudian digunakan untuk proses pelatihan dan prediksi, yang selanjutnya dievaluasi menggunakan metrik MAE, RMSE, dan MAPE. Alur metode penelitian ditunjukkan pada Gambar 1.



Gambar 1. Alur metode penelitian

Gambar 1 menunjukkan tahapan penelitian secara berurutan mulai dari pengolahan data hingga evaluasi model. Setiap tahap dilakukan untuk memastikan data telah melalui proses yang sesuai sebelum digunakan dalam pemodelan. Tahap *preprocessing* dan *feature engineering* menghasilkan data yang siap digunakan sebagai input model, yang kemudian digunakan untuk proses prediksi dan evaluasi.

3.1. Data Penelitian

Data yang digunakan dalam penelitian ini berupa data penjualan mingguan yang disusun dalam bentuk *time series*. Data mencakup beberapa variabel sebagai input pemodelan, yaitu tanggal, jumlah unit terjual, status diskon, persentase diskon, serta fitur *lag*. Data kemudian dibagi menjadi data pelatihan dan data pengujian berdasarkan urutan waktu, dengan data

tahun 2025 digunakan sebagai data pengujian. Contoh struktur data ditunjukkan pada Tabel 1.

Tabel 1. Contoh data penjualan mingguan tahun 2025

tgl	ut	id	dp	l1	l2	l3	l4
06/01/2025	22	1	15	22	20	19	18
13/01/2025	21	1	15	22	22	20	19
20/01/2025	24	1	15	21	22	22	20
27/01/2025	24	1	15	24	21	22	22
03/02/2025	22	0	0	24	24	21	22

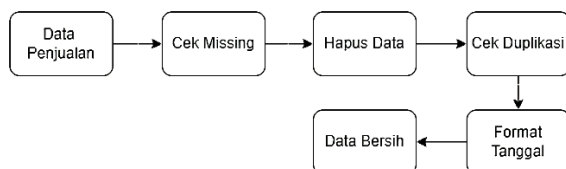
Keterangan:

- tgl = tanggal minggu pengamatan
- ut = unit_terjual
- id = is_diskon
- dp = diskon_persen
- l1–l4 = lag_1 hingga lag_4

Tabel 1 menampilkan sebagian data penjualan mingguan yang digunakan dalam penelitian. Setiap baris merepresentasikan satu periode minggu dengan variabel target berupa jumlah unit terjual serta variabel pendukung seperti diskon dan fitur *lag*. Fitur *lag* merepresentasikan nilai penjualan pada periode sebelumnya sehingga membantu model dalam memahami pola historis dan menghasilkan prediksi pada periode berikutnya.

3.2. Preprocessing Data

Tahap *preprocessing* dilakukan untuk mempersiapkan data agar sesuai digunakan dalam proses pemodelan. Tahapan ini meliputi identifikasi dan penghapusan *missing value*, pemeriksaan data duplikat, serta penyesuaian format variabel tanggal agar dapat diproses dengan baik. Alur *preprocessing* yang dilakukan ditunjukkan pada Gambar 2.



Gambar 2. Alur *preprocessing* data

Gambar 2 menunjukkan tahapan *preprocessing* yang dilakukan secara berurutan hingga diperoleh data yang bersih dan siap digunakan. Proses ini memastikan tidak terdapat nilai kosong maupun data duplikat yang dapat mempengaruhi hasil pelatihan model. Dengan demikian, data yang dihasilkan telah memenuhi kebutuhan untuk digunakan dalam proses pemodelan.

3.3. Feature Engineering

Tahap *feature engineering* dilakukan untuk membentuk variabel input yang digunakan dalam proses pemodelan. Fitur yang digunakan terdiri dari fitur *lag* serta variabel terkait diskon yang diperoleh dari data hasil *preprocessing*. Fitur *lag* dibentuk dari nilai penjualan pada periode sebelumnya (lag-1 hingga lag-4) untuk merepresentasikan pola historis,

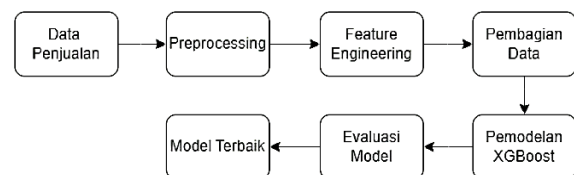
sedangkan variabel *is_diskon* dan *diskon_persen* digunakan untuk merepresentasikan aktivitas promosi yang dapat mempengaruhi jumlah penjualan.

3.4. Pembagian Data

Pembagian data dilakukan berdasarkan urutan waktu sesuai karakteristik *time series*. Data penjualan tahun 2023–2024 digunakan sebagai data *training*, sedangkan data tahun 2025 digunakan sebagai data *testing*. Pembagian dilakukan tanpa pengacakan agar urutan data tetap kronologis dan pola historis tetap terjaga selama proses pemodelan.

3.5. Pemodelan XGBoost

Pemodelan dalam penelitian ini menggunakan algoritma XGBoost untuk melakukan peramalan jumlah penjualan. Model dibangun dengan memanfaatkan fitur yang telah dibentuk pada tahap sebelumnya, yaitu fitur *lag* serta variabel terkait diskon, dan dilatih menggunakan data *training* untuk menghasilkan model prediksi. XGBoost digunakan karena kemampuannya dalam menangani hubungan nonlinier dan meningkatkan akurasi prediksi secara iteratif, dengan mekanisme pembelajaran yang mengikuti fungsi objektif pada Persamaan (1). Alur pemodelan XGBoost dalam penelitian ini ditunjukkan pada Gambar 3.



Gambar 3. Alur pemodelan XGBoost

Gambar 3 menunjukkan alur pemodelan XGBoost yang dimulai dari data penjualan, proses *preprocessing*, *feature engineering*, pembagian data, pemodelan menggunakan algoritma XGBoost, hingga evaluasi model untuk memperoleh model terbaik. Tahapan tersebut dilakukan secara berurutan agar model dapat mempelajari pola historis data dan menghasilkan prediksi yang optimal. Model yang telah dilatih kemudian digunakan untuk melakukan prediksi pada data *testing*, di mana parameter model ditentukan melalui beberapa skenario *hyperparameter* yang diuji pada tahap berikutnya. Hasil prediksi selanjutnya dibandingkan dengan nilai aktual menggunakan metrik evaluasi untuk mengetahui performa model dalam melakukan peramalan penjualan.

3.6. Skenario Optimasi Hyperparameter

Optimasi *hyperparameter* dilakukan untuk memperoleh kombinasi parameter yang menghasilkan performa model terbaik. Parameter yang digunakan meliputi *learning rate*, *max depth*, dan *n_estimators* yang mengatur proses pembelajaran model XGBoost. Berdasarkan kombinasi nilai parameter tersebut, ditentukan enam skenario pengujian untuk

menghasilkan variasi model. Skenario optimasi yang digunakan ditunjukkan pada Tabel 2.

Tabel 2. Skenario optimasi hyperparameter XGBoost

Skenario	Learning Rate	Max Depth	N Estimators
1	0,01	3	150
2	0,01	4	300
3	0,03	3	100
4	0,03	3	200
5	0,05	4	150
6	0,10	2	50

Tabel 2 menunjukkan enam kombinasi parameter yang digunakan dalam proses optimasi model. Perbedaan nilai parameter mempengaruhi kemampuan model dalam menangkap pola data serta tingkat kompleksitas model. Melalui pengujian skenario tersebut, dipilih model dengan performa terbaik berdasarkan hasil evaluasi.

3.7. Proses Pelatihan dan Prediksi

Proses pelatihan dilakukan menggunakan data *training* yang telah dibagi berdasarkan urutan waktu. Model XGBoost dilatih pada setiap skenario *hyperparameter* untuk mempelajari hubungan antara fitur input, yaitu fitur *lag* dan variabel diskon, terhadap jumlah unit terjual. Setelah proses pelatihan, model digunakan untuk memprediksi data *testing* tahun 2025. Hasil prediksi kemudian dibandingkan dengan nilai aktual menggunakan metrik evaluasi untuk mengetahui performa setiap skenario dan menentukan model terbaik.

3.8. Evaluasi Model

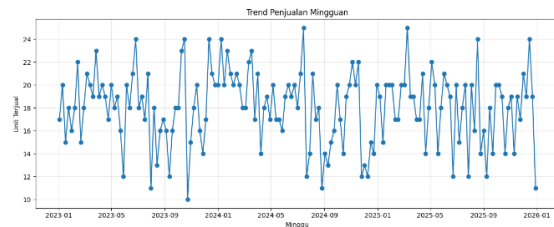
Evaluasi model dilakukan untuk mengukur tingkat kesalahan prediksi yang dihasilkan oleh setiap skenario *hyperparameter*. Hasil prediksi pada data *testing* dibandingkan dengan nilai aktual menggunakan metrik MAE, RMSE, dan MAPE. MAE digunakan untuk melihat rata-rata kesalahan absolut, RMSE digunakan untuk memberikan penalti lebih besar pada kesalahan yang tinggi, sedangkan MAPE digunakan untuk melihat kesalahan prediksi dalam bentuk persentase. Skenario dengan nilai kesalahan paling rendah dipilih sebagai model terbaik dalam penelitian ini.

4. HASIL DAN PEMBAHASAN

Hasil penelitian diperoleh melalui proses pemodelan yang telah dilakukan pada tahap sebelumnya. Hasil yang ditampilkan meliputi gambaran data penjualan, hasil prediksi model, serta evaluasi performa model berdasarkan beberapa skenario *hyperparameter*. Pembahasan dilakukan untuk melihat kemampuan model dalam menangkap pola data penjualan.

4.1. Gambaran Data

Untuk memahami karakteristik data, dilakukan visualisasi terhadap data penjualan mingguan. Visualisasi ini digunakan untuk melihat pola perubahan penjualan dari waktu ke waktu serta mengidentifikasi karakteristik utama dari data. Hasil visualisasi ditunjukkan pada Gambar 4.



Gambar 4. Tren penjualan mingguan

Gambar 4 menunjukkan bahwa data penjualan memiliki pola yang fluktuatif dengan variasi nilai yang cukup tinggi pada setiap periode. Terdapat peningkatan dan penurunan penjualan pada waktu tertentu yang menunjukkan perubahan permintaan. Pola ini menunjukkan bahwa data tidak bersifat linier dan dipengaruhi oleh beberapa faktor.

Selain visualisasi, dilakukan pemeriksaan terhadap kualitas data yang meliputi jumlah data, jumlah kolom, serta keberadaan data duplikat dan nilai kosong. Hasil pemeriksaan ditunjukkan pada Tabel 3. Pemeriksaan ini dilakukan untuk memastikan data dalam kondisi yang dapat digunakan dalam proses pemodelan.

Tabel 3. Hasil pemeriksaan data

Pemeriksaan	Hasil
Jumlah baris	157
Jumlah kolom	8
Jumlah data duplikat	0
Jumlah nilai kosong	0

Tabel 3 menunjukkan bahwa *dataset* terdiri dari 157 baris dan 8 kolom tanpa data duplikat maupun nilai kosong. Pemeriksaan pada setiap variabel, yaitu minggu, *unit_terjual*, *is_diskon*, *diskon_persen*, serta *lag_1* hingga *lag_4*, juga menunjukkan bahwa tidak terdapat nilai kosong. Dengan kondisi tersebut, data dapat digunakan langsung dalam proses pemodelan tanpa memerlukan penanganan *missing value*.

4.2. Hasil Evaluasi Model

Sebelum evaluasi dilakukan, data dibagi menjadi data *training* dan data *testing* berdasarkan urutan waktu. Data *training* digunakan untuk membangun model, sedangkan data *testing* digunakan untuk menguji performa model pada data yang belum pernah digunakan sebelumnya. Ringkasan pembagian data ditunjukkan pada Tabel 4.

Tabel 4. Pembagian data *training* dan *testing*

Data	Periode Awal	Periode Akhir	Jumlah Data	Rata-rata Unit Terjual
Training	02/01/2023	30/12/2024	105	18,1333
Testing	06/01/2025	29/12/2025	52	17,9038

Tabel 4 menunjukkan bahwa jumlah data *training* lebih besar dibandingkan data *testing*, sehingga model memiliki cukup data untuk mempelajari pola historis. Rata-rata unit terjual pada kedua data juga tidak berbeda jauh, sehingga distribusi data tetap terjaga. Dengan kondisi tersebut, data dapat digunakan untuk proses pelatihan dan evaluasi model.

Berdasarkan pembagian tersebut, model dilatih dan diuji menggunakan beberapa skenario *hyperparameter*. Evaluasi dilakukan dengan membandingkan hasil prediksi terhadap nilai aktual pada data *testing* menggunakan metrik MAE, RMSE, dan MAPE. Hasil evaluasi untuk setiap skenario ditunjukkan pada Tabel 5.

Tabel 5. Hasil evaluasi model pada setiap skenario

Sk	LR	MD	NE	MAE	RMSE	MAPE (%)
1	0.01	3	150	1,7518	2,2198	9,7068
2	0.01	4	300	1,8358	2,2795	10,1881
3	0.03	3	100	1,7426	2,1744	9,6850
4	0.03	3	200	1,8201	2,2548	10,0954
5	0.05	4	150	1,9719	2,4434	10,8863
6	0.10	2	50	1,7269	2,1112	9,5985

Keterangan:

- Sk = skenario
- LR = *learning rate*
- MD = *max depth*
- NE = *n_estimators*

Tabel 5 menunjukkan bahwa setiap skenario menghasilkan nilai kesalahan yang berbeda. Berdasarkan nilai evaluasi yang diperoleh, skenario 6 menempati performa terbaik, diikuti oleh skenario 3, skenario 1, skenario 4, skenario 2, dan skenario 5. Skenario terbaik diperoleh pada skenario 6 dengan nilai MAE sebesar 1,7269, RMSE sebesar 2,1112, dan MAPE sebesar 9,5985%. Hasil tersebut menunjukkan bahwa model mampu menghasilkan prediksi dengan tingkat kesalahan yang relatif lebih kecil dibandingkan skenario lainnya. Selain itu, peningkatan kompleksitas parameter tidak selalu menghasilkan performa yang lebih baik, sehingga pemilihan *hyperparameter* menjadi faktor penting dalam proses pemodelan.

4.3. Analisis Perbandingan Skenario

Analisis perbandingan skenario dilakukan untuk memahami perbedaan performa model yang dihasilkan dari setiap kombinasi *hyperparameter*. Perbandingan ini didasarkan pada nilai evaluasi, yaitu MAE, RMSE, dan MAPE, yang telah diperoleh pada tahap sebelumnya. Setiap skenario memiliki kombinasi parameter yang berbeda sehingga menghasilkan tingkat akurasi yang tidak sama. Oleh

karena itu, analisis ini bertujuan untuk mengidentifikasi kombinasi parameter yang paling sesuai dengan karakteristik data penjualan.

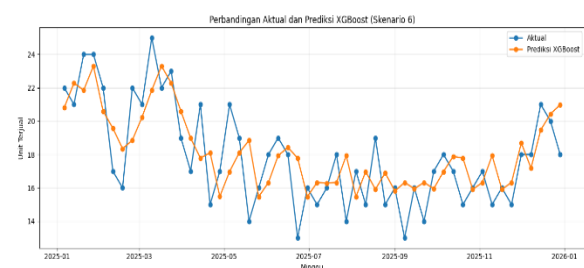
Berdasarkan hasil evaluasi pada Tabel 5, skenario 6 menghasilkan nilai kesalahan yang paling rendah dibandingkan skenario lainnya. Nilai MAE sebesar 1.7269, RMSE sebesar 2.1112, dan MAPE sebesar 9.5985% menunjukkan bahwa model mampu menghasilkan prediksi yang lebih mendekati nilai aktual dibandingkan model lainnya. Hal ini mengindikasikan bahwa kombinasi parameter pada skenario 6 memberikan keseimbangan yang baik antara akurasi dan stabilitas model.

Jika dibandingkan dengan skenario lainnya, peningkatan kompleksitas model tidak selalu menghasilkan performa yang lebih baik. Skenario dengan nilai *n_estimators* dan *max_depth* yang lebih besar justru tidak selalu memberikan kesalahan yang lebih kecil. Hal ini menunjukkan bahwa model yang terlalu kompleks berpotensi mengalami *overfitting*, sehingga kurang optimal dalam melakukan generalisasi pada data pengujian. Sebaliknya, parameter yang lebih sederhana dapat menghasilkan performa yang lebih stabil.

Berdasarkan analisis tersebut, skenario 6 dipilih sebagai model terbaik dalam penelitian ini karena memiliki nilai kesalahan paling rendah dibandingkan skenario lainnya. Kombinasi parameter berupa *learning rate* sebesar 0.10, *max_depth* sebesar 2, dan *n_estimators* sebanyak 50 menghasilkan performa yang lebih optimal dalam proses peramalan. Hasil tersebut menunjukkan bahwa keseimbangan antara kompleksitas model dan kemampuan generalisasi menjadi faktor penting dalam menentukan performa model XGBoost.

4.4. Hasil Prediksi Model Terbaik

Setelah model terbaik diperoleh berdasarkan hasil evaluasi pada tahap sebelumnya, model tersebut digunakan untuk menghasilkan prediksi pada data pengujian. Model yang digunakan merupakan hasil dari skenario 6 dengan kombinasi *hyperparameter* yang telah ditentukan. Proses prediksi dilakukan untuk melihat kemampuan model dalam merepresentasikan pola data penjualan yang belum pernah dilihat sebelumnya. Hasil perbandingan antara nilai aktual dan nilai prediksi pada periode pengujian minggu pertama Januari 2025 hingga minggu terakhir Desember 2025 ditunjukkan pada Gambar 5.



Gambar 5. Perbandingan nilai aktual dan prediksi XGBoost (Skenario 6)

Gambar 5 menunjukkan bahwa hasil prediksi model secara umum mampu mengikuti pola pergerakan data aktual. Pada sebagian besar periode, nilai prediksi berada cukup dekat dengan nilai aktual, sehingga menunjukkan bahwa model mampu menangkap pola historis dengan baik. Meskipun demikian, terdapat beberapa titik di mana selisih antara nilai prediksi dan nilai aktual cukup besar. Perbedaan tersebut umumnya terjadi pada perubahan nilai yang bersifat drastis, yang menunjukkan keterbatasan model dalam menangkap fluktuasi ekstrem. Namun, secara keseluruhan model tetap mampu mempertahankan pola tren penjualan sehingga hasil prediksi yang dihasilkan masih tergolong baik.

Sebagai ilustrasi numerik, sebagian hasil prediksi pada beberapa periode awal data pengujian ditunjukkan pada Tabel 6. Tabel 6 menggunakan contoh lima minggu pertama dari data *testing* tahun 2025 untuk memberikan gambaran perbandingan langsung antara nilai aktual dan hasil prediksi model. Penyajian tabel ini bertujuan untuk melengkapi visualisasi hasil prediksi yang telah ditampilkan pada Gambar 5.

Tabel 6. Contoh hasil prediksi model terbaik

Minggu	Aktual	Prediksi
06/01/2025	22	20,81
13/01/2025	21	22,28
20/01/2025	24	21,85
27/01/2025	24	23,30
03/02/2025	22	20,59

Tabel 6 menunjukkan bahwa hasil prediksi model memiliki selisih yang bervariasi terhadap nilai aktual. Pada beberapa periode, nilai prediksi mendekati nilai aktual, yang menunjukkan performa model yang cukup baik. Namun, pada beberapa periode lainnya masih terdapat perbedaan yang cukup signifikan. Secara keseluruhan, model mampu mengikuti tren utama dari data penjualan meskipun masih terdapat kesalahan pada beberapa titik tertentu.

4.5. Analisis Hasil Prediksi

Hasil prediksi yang ditunjukkan pada Gambar 5 menunjukkan bahwa model mampu mengikuti pola pergerakan data penjualan secara umum. Nilai prediksi cenderung berada dekat dengan nilai aktual pada sebagian besar periode, sehingga menunjukkan bahwa model dapat menangkap tren utama dari data penjualan. Kemampuan ini menunjukkan bahwa model XGBoost yang digunakan memiliki performa yang cukup baik dalam mempelajari pola historis data. Dengan demikian, model dapat digunakan untuk memberikan estimasi penjualan yang cukup representatif terhadap kondisi sebenarnya.

Meskipun demikian, masih terdapat beberapa periode di mana selisih antara nilai prediksi dan nilai aktual cukup besar. Perbedaan ini umumnya terjadi pada saat terjadi perubahan nilai yang signifikan dalam waktu singkat. Hal ini menunjukkan bahwa model

masih memiliki keterbatasan dalam menangkap fluktuasi ekstrem pada data penjualan. Kondisi tersebut dapat disebabkan oleh adanya faktor eksternal yang tidak sepenuhnya direpresentasikan dalam fitur yang digunakan dalam model.

Selain itu, penggunaan fitur *lag* dan variabel diskon dalam model memberikan kontribusi dalam membantu model memahami pola historis penjualan. Fitur *lag* memungkinkan model untuk memanfaatkan informasi dari periode sebelumnya, sedangkan variabel diskon membantu menangkap pengaruh promosi terhadap peningkatan penjualan. Namun, pengaruh faktor lain di luar variabel yang digunakan masih dapat menyebabkan ketidaksesuaian antara nilai prediksi dan aktual. Oleh karena itu, pengembangan model dengan penambahan fitur lain dapat menjadi langkah yang dapat dilakukan untuk meningkatkan akurasi prediksi di masa mendatang.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah dilakukan, model XGBoost dengan optimasi *hyperparameter* mampu menghasilkan prediksi penjualan yang cukup akurat pada data *time series*. Pengujian terhadap enam skenario parameter menunjukkan bahwa skenario terbaik diperoleh pada kombinasi *learning rate* sebesar 0.10, *max depth* sebesar 2, dan *n_estimators* sebanyak 50, dengan nilai MAPE sebesar 9.5985%. Hasil prediksi menunjukkan bahwa model mampu mengikuti pola tren penjualan secara umum, meskipun masih terdapat perbedaan pada beberapa periode dengan fluktuasi yang tinggi. Penggunaan fitur *lag* dan variabel diskon berperan dalam membantu model memahami pola historis dan pengaruh promosi terhadap penjualan. Oleh karena itu, model ini dapat digunakan sebagai pendekatan yang efektif dalam peramalan permintaan. Untuk penelitian selanjutnya, disarankan untuk menambahkan variabel eksternal lain serta mengeksplorasi metode atau model yang lebih kompleks guna meningkatkan akurasi prediksi, khususnya dalam menangkap fluktuasi ekstrem.

DAFTAR PUSTAKA

- [1] K. Douaioui, R. Oucheikh, O. Benmoussa, and C. Mabrouki, "Machine Learning and Deep Learning Models for Demand Forecasting in Supply Chain Management: A Critical Review," *Appl. Syst. Innov.*, vol. 7, no. 5, 2024, doi: 10.3390/asi7050093.
- [2] R. Bandaru, "Impact of AI-Driven Demand Forecasting on Retail Inventory Efficiency," *Int. J. Sci. Res. Arch.*, vol. 18, no. 1, pp. 394–400, 2026, doi: 10.30574/ijrsra.2026.18.1.0062.
- [3] P. Praveen Kumar Ashok, "Time-Series Forecasting in Inventory and Sales Using SAP Analytics Cloud," *Int. J. Sci. Res.*, vol. 12, no. 4, pp. 1994–1998, 2025, doi: 10.21275/sr23431091015.
- [4] H. J. Wahedi, M. Heltoft, G. J. Christophersen,

- T. Severinsen, S. Saha, and I. E. Nielsen, "Forecasting and Inventory Planning: An Empirical Investigation of Classical and Machine Learning Approaches for Svanehøj's Future Software Consolidation," *Appl. Sci.*, vol. 13, no. 15, pp. 1–21, 2023, doi: 10.3390/app13158581.
- [5] E. Martins and N. V. Galegale, "Sales forecasting using machine learning algorithms," *Rev. Gestão e Secr. (Management Adm. Prof. Rev.)*, vol. 14, no. 7, pp. 11294–11308, 2023, doi: 10.7769/gesec.v14i7.1670.
- [6] A. P. Sari, R. Arifuddin, and D. A. Prasetya, "Short-Term Wind Speed and Direction Forecasting by 3DCNN and Deep Convolutional LSTM," *IEEEJ Trans. Electr. Electron. Eng.*, 2023, doi: 10.1002/tee.23669.
- [7] A. C. Tu, T. Januschowski, Y. Wang, and A. T. Cemgil, "Forecasting intermittent and sparse time series: A unified probabilistic framework via deep renewal processes," *PLoS One*, pp. 1–26, 2021, doi: 10.1371/journal.pone.0259764.
- [8] G. R. Kumar, G. Thippanna, and S. B. Naik, "Enhanced Sales Forecasting Using XGBoost and Feature Engineering: A Comparative Evaluation," *J. Knowl. Data Sci. Inf. Manag.*, vol. 2, no. 3, 2025.
- [9] Y. V. Via, I. Y. Purbasari, and A. P. Pratama, "Analisa Algoritma Convolution Neural Network (Cnn) Pada Klasifikasi Genre Musik Berdasar Durasi Waktu," *Scan J. Teknol. Inf. dan Komun.*, vol. 17, no. 1, pp. 35–41, 2022, doi: 10.33005/scan.v17i1.3251.
- [10] S. Nagadevi, G. Abirami, R. Vidhya, S. Selvakumar, and C. Vijayalakshmi, "XGBoost-Based Demand Forecasting in Supply Chain Management Using Machine Learning Algorithm," *Int. J. Interact. Mob. Technol.*, vol. 19, no. 18, pp. 161–174, Sep. 2025, doi: 10.3991/ijim.v19i18.57253.
- [11] L. Zhang, W. Bian, W. Qu, L. Tuo, and Y. Wang, "Time series forecast of sales volume based on XGBoost Time series forecast of sales volume based on XGBoost," *J. Phys. Conf. Ser.*, 2021, doi: 10.1088/1742-6596/1873/1/012067.
- [12] M. S. Rahman, A. H. Chowdhury, and M. Amrin, "Accuracy comparison of ARIMA and XGBoost forecasting models in predicting the incidence of COVID-19 in Bangladesh," *PLOS Glob. Public Heal.*, vol. 2, no. 5, pp. 1–13, 2022, doi: 10.1371/journal.pgph.0000495.
- [13] S. Ni, Y. Peng, K. Peng, and Z. Liu, "Supply Chain Demand Forecast Based on SSA-XGBoost Model," *J. Comput. Commun.*, vol. 10, no. 12, pp. 71–83, 2022, doi: 10.4236/jcc.2022.1012006.
- [14] Z. Qinghe, X. Wen, H. Boyan, W. Jong, and F. Junlong, "Optimised extreme gradient boosting model for short term electric load demand forecasting of regional grid system," *Sci. Rep.*, vol. 12, no. 1, pp. 1–13, 2022, doi: 10.1038/s41598-022-22024-3.
- [15] A. Massaro, A. Panarese, D. Giannone, and A. Galiano, "Augmented data and Xgboost improvement for sales forecasting in the large-scale retail sector," *Appl. Sci.*, vol. 11, no. 17, 2021, doi: 10.3390/app11177793.
- [16] R. E. Van Ruitenbeek, G. M. Koole, and S. Bhulai, "A hierarchical agglomerative clustering for product sales forecasting," *Decis. Anal. J.*, vol. 8, no. September, p. 100318, 2023, doi: 10.1016/j.dajour.2023.100318.
- [17] E. Erjiang, X. Tian, Y. Tao, and H. Sun, "An ES-based model with contemporaneous and temporal aggregation for forecasting intermittent and lumpy retail demand," *Procedia Comput. Sci.*, vol. 242, pp. 264–271, 2024, doi: 10.1016/j.procs.2024.08.239.
- [18] A. Kaltsounis, E. Spiliotis, and V. Assimakopoulos, "Conditional Temporal Aggregation for Time Series Forecasting Using Feature-Based Meta-Learning," *MDPI Algorithms*, 2023.
- [19] I. M. Hammam, A. K. El-Kharbotly, and Y. M. Sadek, "Adaptive demand forecasting framework with weighted ensemble of regression and machine learning models along life cycle variability," *Sci. Rep.*, vol. 15, no. 1, pp. 1–16, 2025, doi: 10.1038/s41598-025-23352-w.
- [20] L. B. Sina, C. A. Secco, M. Blazevic, and K. Nazemi, "Hybrid Forecasting Methods—A Systematic Review," May 01, 2023, *MDPI*. doi: 10.3390/electronics12092019.
- [21] L. Ni, Z. Huang, and N. Fu, "A Stacking-Based Fusion Framework for Dynamic Demand Forecasting in E-Commerce," *Mathematics*, vol. 13, no. 21, pp. 1–16, 2025, doi: 10.3390/math13213436.
- [22] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Aug, no. March, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.
- [23] Y. M. Tang, K. Y. Chau, Y. Y. Lau, and Z. Zheng, "Data-Intensive Inventory Forecasting with Artificial Intelligence Models for Cross-Border E-Commerce Service Automation," *Appl. Sci.*, vol. 13, no. 5, 2023, doi: 10.3390/app13053051.