

TESIS
ANALISIS PENENTUAN HARGA JUAL DENGAN
ALGORITMA *GRADIENT BOOSTING* DI TOKO PASAR
TRADISIONAL



Disusun oleh:

BAGAS DWI NOVIANTO

23.55.2509

Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2025

TESIS
ANALISIS PENENTUAN HARGA JUAL DENGAN
ALGORITMA *GRADIENT BOOSTING* DI TOKO
TRADISIONAL

ANALYSIS OF SELLING PRICE DETERMINATION WITH
GRADIENT BOOSTING ALGORITHM TRADITIONAL MARKET
STORES

Diajukan untuk memenuhi salah satu syarat mencapai derajat Pascasarjana

Program Studi S2 PJJ Informatika



Disusun oleh:

BAGAS DWI NOVIANTO

23.55.2509

Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2025

HALAMAN PERSETUJUAN

**ANALISIS PENENTUAN HARGA JUAL DENGAN ALGORITMA
GRADIENT BOOSTING DI TOKO PASAR TRADISIONAL**

**ANALYSYS OF SELLING PRICE DETERMINATION WITH GRADIENT
BOOSTING ALGORITHM IN TRADITIONAL MARKET STORES**

yang disusun dan diajukan oleh

BAGAS DWI NOVIANTO

23.55.2509

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 01 Oktober 2025

Dosen Pembimbing,



Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PENGESAHAN

ANALISIS PENENTUAN HARGA JUAL DENGAN ALGORITMA
GRADIENT BOOSTING DI TOKO PASAR TRADISIONAL

ANALYSYS OF SELLING PRICE DETERMINATION WITH GRADIENT
BOOSTING ALGORITHM IN TRADITIONAL MARKET STORES

yang disusun dan diajukan oleh

BAGAS DWI NOVIANTO

23.55.2509

Telah dipertahankan di depan Dewan Penguji
pada tanggal 01 Oktober 2025

Susunan Dewan Penguji

Nama Penguji

Tanda Tangan

Dr. Ferry Wahyu Wibowo, M.Cs.
NIK. 190302235



Robert Marco, S.T., M.T., Ph.D.
NIK. 190302228



Prof. Dr. Kusriani, M.Kom.
NIK. 190302106



Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal 01 Oktober 2025

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : BAGAS DWI NOVIANTO
NIM : 23.55.2509
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:

ANALISIS PENENTUAN HARGA JUAL DENGAN ALGORITMA GRADIENT BOOSTING DI TOKO PASAR TRADISIONAL

Dosen Pembimbing : Prof. Dr. Kusrini, M.Kom,

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 01 Oktober 2025

Yang Menyatakan,



Bagas Dwi Novianto

KATA PENGANTAR

Puji dan syukur kita panjatkan kehadiran Allah SWT yang telah memberikan rahmat, hidayah dan kekuatan sehingga saya dapat menyelesaikan tesis yang berjudul "Analisis Penentuan Harga Jual dengan Algoritma Gradient Boosting di Toko Pasar Tradisional" dengan baik dan tepat waktu. Pada kesempatan ini penulis ingin menyampaikan ucapan terimakasih kepada :

1. Kedua Orang Tua saya, istri dan anak tercinta atas doa serta dukungannya selama ini.
2. Bapak Prof. Dr. M. Suyanto, M.M., selaku Rektor Universitas AMIKOM Yogyakarta.
3. Ibu Prof. Dr. Kusriani, M.Kom., selaku Dekan Fakultas Ilmu Komputer Universitas AMIKOM Yogyakarta dan juga sebagai dosen pembimbing utama yang telah membantu memberikan ilmu, masukan dan saran dalam penelitian ini.
4. Segenap Dosen dan Staff Magister Teknik Informatika Universitas AMIKOM Yogyakarta yang telah memberikan ilmu, wawasan, bantuan dan pengalaman baru pada penulis selama perkuliahan.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh karena itu penulis dengan sangat senang hati menerima kritik dan saran yang membangun dari pembaca. Akhir kata, penulis berharap semoga tesis ini dapat memberikan manfaat bagi yang membacanya.

Yogyakarta, 01 Oktober 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	1
HALAMAN PERSETUJUAN.....	2
HALAMAN PENGESAHAN.....	3
HALAMAN PERNYATAAN KEASLIAN TESIS.....	4
KATA PENGANTAR.....	5
DAFTAR ISI.....	6
DAFTAR TABEL.....	9
DAFTAR GAMBAR.....	10
DAFTAR LAMBANG DAN SINGKATAN.....	11
DAFTAR ISTILAH.....	12
INTISARI.....	13
<i>ABSTRACT</i>	14
BAB I PENDAHULUAN.....	15
1.1. Latar Belakang Masalah.....	15
1.2. Rumusan Masalah.....	21
1.3. Batasan Masalah.....	22
1.4. Tujuan Penelitian.....	22
1.5. Manfaat Penelitian.....	23
BAB II TINJAUAN PUSTAKA.....	24
2.1. Tinjauan Pustaka.....	24
2.2. Keaslian Penelitian.....	28

2.3. Landasan Teori.....	33
2.3.1 Pasar Tradisional	33
2.3.2 Harga Jual.....	33
2.3.3 <i>Gradient Boosting</i>	33
2.3.4 <i>XGBoost</i>	34
2.3.5 <i>GridSearchCV</i>	35
2.3.6 <i>SHAP (SHapley Additive exPlanations)</i>	35
2.3.7 <i>R² (Coefficient of Determination)</i>	36
2.3.8 <i>MAPE (Mean Absolute Percentage Error)</i>	36
2.3.9 <i>Python</i>	37
BAB III METODE PENELITIAN.....	38
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	38
3.1.1 Jenis Penelitian.....	38
3.1.2 Sifat Penelitian.....	38
3.1.3 Pendekatan Penelitian.....	39
3.2. Metode Pengumpulan Data.....	39
3.3. Metode Analisis Data.....	40
3.4. Alur Penelitian.....	41
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	45
4.1. Pengumpulan <i>Dataset</i>	45
4.1.1 Uji Asumsi Stasioneritas Data.....	50

4.2. Pra-pemrosesan Data	54
4.2.1. Uji Korelasi Antar Fitur.....	58
4.2.2. Alur Proses Pra-Pemrosesan Hingga Penentuan Harga Jual	61
4.3. Pembagian <i>Dataset</i>	64
4.4. Pengembangan Model.....	64
4.5. Pengujian Model.....	67
4.6. <i>Feature Importance</i>	69
4.6.1. <i>Feature Importance</i> pada <i>dataset</i> besar	69
4.6.2. <i>Feature Importance</i> pada <i>dataset</i> kecil.....	73
4.7. <i>SHAP Analysis</i>	76
4.7.1. <i>SHAP Analysis Dataset</i> Besar.....	77
4.7.2. <i>SHAP Analysis Dataset</i> Kecil.....	80
4.8. Analisis Skenario Pengujian.....	83
BAB V PENUTUP.....	85
5.1. Kesimpulan	85
5.2. Saran	87
DAFTAR PUSTAKA	88

DAFTAR TABEL

Tabel 2. 1. Matriks <i>Literature Review</i> Analisis Penentuan Harga Jual Dengan Algoritma <i>Gradient Boosting</i> di Toko Pasar Tradisional.....	28
Tabel 3. 1. Atribut <i>Dataset</i>	42
Tabel 4. 1. <i>Dataset</i> Besar.....	45
Tabel 4. 2. <i>Dataset</i> Kecil.....	46
Tabel 4. 3. Uji Stasioneritas <i>Dataset</i> Besar.....	53
Tabel 4. 4. Uji Stasioneritas <i>Dataset</i> Kecil.....	53
Tabel 4. 5. Parameter Kombinasi <i>GridSearchCV</i>	66
Tabel 4. 6. Hasil Evaluasi Algoritma pada <i>dataset</i> besar.....	68
Tabel 4. 7. Hasil Evaluasi Algoritma pada <i>dataset</i> kecil.....	68

DAFTAR GAMBAR

Gambar 3. 1. Alur Penelitian.....	41
Gambar 4. 1. <i>Labeling Dataset</i>	49
Gambar 4. 2. Kode Pengujian Stasioneritas Data.....	51
Gambar 4. 3. Kode Pra-pemrosesan Data.....	56
Gambar 4. 4. Kode Uji Korelasi.....	58
Gambar 4. 5. <i>Heatmap Dataset Besar</i>	59
Gambar 4. 6. <i>Heatmap Dataset Kecil</i>	60
Gambar 4. 7. Kode Pembagian <i>Dataset</i>	64
Gambar 4. 8. Kode <i>GridSearchCV Parameter</i>	65
Gambar 4. 9. Kode Pengujian Model <i>Gradient Boosting</i> dan <i>XGBoost</i>	68
Gambar 4. 10. Kode Visualisasi <i>Feature Importance Dataset Besar</i>	70
Gambar 4. 11. <i>Feature Importance Gradient Boosting Dataset Besar</i>	71
Gambar 4. 12. <i>Feature Importance XGBoost Dataset Besar</i>	72
Gambar 4. 13. Kode Visualisasi <i>Feature Importance Dataset Kecil</i>	73
Gambar 4. 14. <i>Feature Importance Gradient Boosting Dataset Kecil</i>	74
Gambar 4. 15. <i>Feature Importance XGBoost Dataset Kecil</i>	75
Gambar 4. 16. Kode Visualisasi <i>SHAP Analysis</i>	77
Gambar 4. 17. <i>SHAP Beeswarm Plot – Dataset Besar</i>	78
Gambar 4. 18. <i>SHAP Bar Plot – Mean Absolute SHAP Value Dataset Besar</i>	79
Gambar 4. 19. <i>SHAP Beeswarm Plot – Dataset Kecil</i>	80
Gambar 4. 20. <i>SHAP Bar Plot – Mean Absolute SHAP Value Dataset Kecil</i>	82

DAFTAR LAMBANG DAN SINGKATAN

XGBoost = Extreme Gradient Boosting

KNN = K-Nearest Neighbors

MAPE = Mean Absolute Percentage Error

MAE = Mean Absolute Error

MSE = Mean Squared Error

RMSE = Root Mean Squared Error

SVM = Support Vector Machine

TN = True Negative

TP = True Positive

FN = False Negative

FP = False Positive

ROC = Receiver Operating Characteristic

R^2 = Koefisien Determinasi

n = jumlah data

β = Koefisien regresi/parameter estimasi

θ = Parameter model yang dioptimalkan

η = Learning rate pada model boosting

DAFTAR ISTILAH

Overfitting = Model terlalu cocok dengan data pelatihan

Underfitting = Model terlalu sederhana dan kurang akurat.

Accuracy = Persentase prediksi yang benar

Precision = Proporsi data positif yang benar

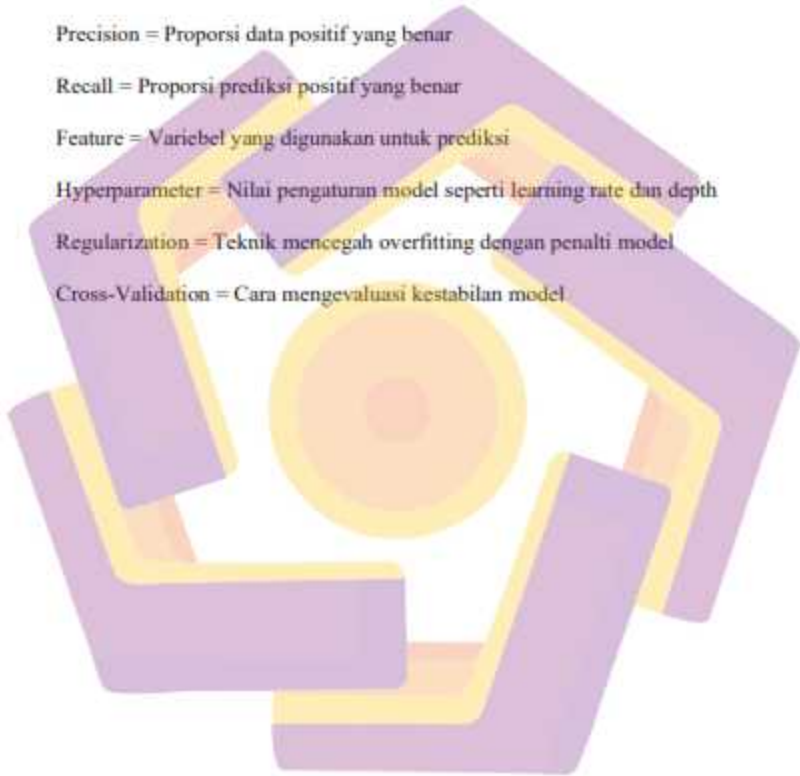
Recall = Proporsi prediksi positif yang benar

Feature = Variabel yang digunakan untuk prediksi

Hyperparameter = Nilai pengaturan model seperti learning rate dan depth

Regularization = Teknik mencegah overfitting dengan penalti model

Cross-Validation = Cara mengevaluasi kestabilan model



INTISARI

Penelitian ini bertujuan untuk menerapkan dan membandingkan algoritma *Gradient Boosting* dan *XGBoost* dalam menentukan harga jual optimal di toko pasar tradisional. Dalam penelitian ini digunakan dua skenario data, yaitu *dataset* skala besar yang terdiri dari data penjualan tahun 2024 dan *dataset* kecil yang terdiri dari data penjualan selama satu bulan di tahun yang sama. Penelitian ini memanfaatkan algoritma *Gradient Boosting* dan *XGBoost* untuk membangun model prediksi dalam penentuan harga jual yang optimal, dimana proses pelatihan model dilakukan melalui optimasi *hyperparameter* menggunakan *GridSearchCV* dan evaluasi kinerja menggunakan metrik regresi seperti R^2 dan *MAPE*. Selain itu, dilakukan pula analisis interpretabilitas model menggunakan *feature importance* dan *SHAP* (*SHaple Additive Explanations*) untuk mengidentifikasi faktor-faktor utama yang mempengaruhi harga jual. Hasil penelitian menunjukkan bahwa kedua algoritma memiliki kinerja prediksi yang sangat baik dengan nilai R^2 mencapai 0.9999 dan *MAPE* dibawah 2% pada semua skenario pengujian. Pada *dataset* besar, *Gradient Boosting* lebih unggul dengan nilai *MAPE* lebih rendah dibandingkan *XGBoost*, sedangkan pada *dataset* kecil *XGBoost* menunjukkan kinerja yang lebih baik karena mekanisme regularisasi yang mampu mencegah *overfitting*. Analisis *feature importance* dan *SHAP* mengidentifikasi bahwa harga beli, harga kompetitor 1, dan harga kompetitor 2 merupakan variabel yang paling berpengaruh dalam prediksi harga jual. Temuan ini dapat dimanfaatkan sebagai dasar dalam pengembangan sistem pendukung keputusan penentuan harga jual berbasis data di pasar tradisional, guna meningkatkan efisiensi, akurasi, dan objektivitas dalam proses penetapan harga.

Kata kunci: *Gradient Boosting*, *XGBoost*, Harga Jual, *SHAP*, *feature importance*.

ABSTRACT

This study aims to implement and compare the Gradient Boosting and XGBoost algorithms in determining the optimal selling price in traditional market stores. Two data scenarios are employed, namely a large-scale dataset consisting of 5,300 weekly sales records from 2024 and a small-scale dataset comprising one month of sales data from the same year. The models are trained through hyperparameter optimization using GridSearchCV and their performance is evaluated using the regression metrics R^2 and MAPE. In addition, model interpretability analysis is conducted using feature importance and SHAP (SHapley Additive Explanations) to identify the key factors influencing selling prices. The results show that both algorithms achieve highly accurate predictions, with R^2 reaching 0.9999 and MAPE below 2% across all testing scenarios. For the large dataset, Gradient Boosting performs slightly better with a lower MAPE compared to XGBoost, whereas for the small dataset, XGBoost demonstrates superior accuracy due to its regularization mechanism that effectively prevents overfitting. The feature importance and SHAP analyses reveal that purchase price, competitor 1 price, and competitor 2 price are the most influential variables in predicting selling prices. These findings can serve as a foundation for developing a data-driven decision support system for selling price determination in traditional markets, thereby improving efficiency, accuracy, and objectivity in the pricing process.

Keyword: Gradient Boosting, XGBoost, Selling Price, SHAP, Feature Importance

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pasar tradisional memiliki peran yang sangat penting bagi perekonomian masyarakat, terutama masyarakat lokal di daerah tersebut. Pasar tradisional menyediakan kebutuhan pokok bagi masyarakat dengan harga yang relatif terjangkau. pasar tradisional memainkan peran penting dalam meningkatkan pendapatan dan kesejahteraan masyarakat melalui transaksi jual beli antara pedagang dan pembeli [1]. Namun, di tengah berkembangnya persaingan dengan toko modern dan *e-commerce*, pedagang di pasar tradisional menghadapi tantangan yang besar dalam menentukan harga jual yang tepat untuk mempertahankan keberlangsungan usaha. Salah satu kendala yang dihadapi oleh pedagang di pasar tradisional adalah bagaimana menetapkan harga jual yang kompetitif dan sesuai dengan kondisi pasar.

Perkembangan pasar modern pada saat ini membawa dampak yang signifikan terhadap pasar tradisional, baik dari sisi pembeli maupun dari sisi pedagang. Pedagang pasar tradisional mulai merasakan dampak dari adanya pasar modern, dengan adanya pasar modern persaingan menjadi semakin ketat, seperti penurunan omset, dan berkurangnya pangsa pasar yang mengakibatkan berubahnya preferensi konsumen dan segmentasi pasar. Dengan kondisi ini mengharuskan pedagang untuk beradaptasi dengan perubahan zaman seperti melakukan pembaruan sistem usaha, dan memperkuat daya saing[2].

Pasar tradisional hingga kini masih sangat bergantung pada intuisi pedagang dalam menentukan harga jual, padahal perkembangan persaingan dengan ritel modern dan *e-commerce* menuntut adanya inovasi berbasis data. Isu penelitian yang diangkat pada penelitian ini adalah bagaimana pemanfaatan *Business Intelligence* dapat digunakan untuk membanung sistem estimasi harga jual yang optimal dengan memanfaatkan variabel historis dan kompetitif yang tersedia di pasar tradisional. Pendekatan ini penting karena sebagian besar penelitian terdahulu mengenai prediksi harga lebih banyak di fokuskan pada sektor ritel modern atau *e-commerce* yang memiliki infrastruktur digital memadai, sementara pasar tradisional relatif tertinggal dalam adopsi teknologi serupa. Dengan demikian, terdapat kesenjangan penelitian dalam penerapan model prediktif berbasis data untuk mendukung penetapan harga di pasar tradisional.

Penentuan harga jual suatu produk merupakan hal yang sangat penting dalam dunia usaha yang semakin kompetitif bersaing dengan pasar modern sehingga dapat mempertahankan pangsa pasar dan meningkatkan penjualan di tengah persaingan dengan *e-commerce* [3]. Banyak faktor yang mempengaruhi harga jual produk seperti promosi, kualitas produk, dan kualitas pelayanan. Pada pasar tradisional kualitas pelayanan merupakan peran penting dalam menciptakan hal positif terhadap konsumen, yang pada akhirnya meningkatkan loyalitas konsumen untuk tetap berbelanja di pasar tradisional [4]. Harga jual tidak hanya ditentukan oleh satu variabel Tunggal, tetapi dipengaruhi oleh berbagai faktor seperti harga modal, tingkat permintaan konsumen, ketersediaan barang, kondisi musim, hingga harga kompetitor. Kompleksitas ini menuntut pendekatan analitik

yang mampu mengidentifikasi dan mengukur kontribusi masing-masing faktor terhadap harga jual. Oleh karena itu, pemodelan berbasis algoritma *Gradient Boosting* dan *XGBoost* tidak hanya bertujuan untuk menghasilkan prediksi akurat, tetapi juga membantu mengungkap faktor-faktor paling berpengaruh dalam proses penetapan harga di lingkungan pasar tradisional. Dalam menentukan harga jual yang lebih kompetitif dapat dilakukan dengan mempertimbangkan beberapa faktor yang dapat mempengaruhi harga jual, untuk mengoptimalkan proses ini digunakanlah algoritma *Gradient Boosting*.

Kesulitan dalam menentukan harga jual yang tepat merupakan persoalan yang tidak hanya dialami oleh satu atau dua pedagang saja, melainkan telah menjadi fenomena umum di berbagai pasar tradisional di Indonesia. Penelitian yang dilakukan di Pasar Tradisional Cakke mengungkap bahwa pedagang cenderung menetapkan harga menggunakan pendekatan pasar yang berlaku (*going rate pricing*), tanpa mempertimbangkan secara menyeluruh faktor biaya operasional maupun permintaan konsumen. Akibatnya, harga yang ditetapkan sering kali tidak mencerminkan kondisi usaha yang sebenarnya dan menyebabkan keetidakstabilan harga [5]. Kasus ini menunjukkan bahwa sistem penentuan harga yang tidak terstruktur dan kurang berbasis data menjadi hambatan dalam menciptakan stabilitas harga dan efisiensi usaha di pasar tradisional. Oleh karena itu, penerapan solusi digital berbasis algoritma *machine learning* menjadi langkah strategis untuk membantu pedagang dalam menetapkan harga jual secara lebih objektif, akurat, dan adaptif terhadap perubahan pasar.

Model *Gradient Boosting* merupakan metode *machine learning* yang memiliki keunggulan dalam menganalisis data kompleks. *Gradient Boosting* adalah metode pembelajaran mesin yang membangun model prediktif melalui ekspansi aditif. Metode ini mengoptimalkan Teknik statistik dan probabilistik untuk menganalisis *dataset* yang besar, kompleks, dan tidak terstruktur. Sementara itu, *XGBoost* merupakan pengembangan dari metode *Gradient Boosting* yang memiliki keunggulan dalam hal regularisasi. *XGBoost* menerapkan dua regularisasi, yaitu *L1 (Lasso Regression)* dan *L2 (Ridge Regression)*, sedangkan *Gradient Boosting* konvensional hanya menggunakan regularisasi *L1* atau *Lasso Regression*. Penambahan teknik regularisasi ini dapat meningkatkan kemampuan model dalam generalisasi terhadap data baru. Regularisasi *L1* bekerja dengan menambahkan penalti berupa jumlah nilai absolut dari koefisien, sementara regularisasi *L2* menggunakan jumlah kuadrat dari bobot pada koneksi dalam jaringan saraf dari model pembelajaran mesin [6].

Pada penelitian ini menggunakan algoritma *Gradient Boosting* yang terdiri dari dua metode yaitu *Gradient Boosting* konvensional dan *XGBoost*. *Gradient Boosting* memiliki kinerja yang baik dalam menangani *dataset* yang kompleks dan efektivitasnya di berbagai skenario prediksi [7]. Sedangkan algoritma *XGBoost* memiliki keunggulan utamanya, termasuk akurasi, kecepatan, dan ketahanannya yang tinggi terhadap *overfitting*. Selain itu, model *Gradient Boosting* dan *XGBoost* biasa digunakan untuk estimasi waktu, masa pakai, harga, keuangan, dan biaya [8].

Pemilihan algoritma *Gradient Boosting* dan *XGBoost* dalam penelitian ini berdasarkan pada kemampuan kedua algoritma dalam menangani data yang

kompleks dan *non-linear*, yang umum ditemukan dalam dinamika harga jual di pasar tradisional. Algoritma ini unggul dibandingkan metode lain seperti regresi linear yang cenderung tidak mampu menangkap interaksi variabel yang kompleks, atau algoritma seperti *SVM* dan *KNN* yang kurang efisien pada data besar [9]. *XGBoost* sebagai pengembangan dari *Gradient Boosting*, memiliki keunggulan tambahan seperti regularisasi ganda (*L1* dan *L2*) untuk mencegah *overfitting*, efisiensi komputasi tinggi, serta fitur interpretasi *feature importance* yang membantu pedagang memahami faktor-faktor utama yang mempengaruhi penetapan harga jual. Sementara itu, *LightGBM* dan *CatBoost* meskipun memiliki kinerja komputasi lebih cepat dan efisien dalam skala industri besar, keduanya memiliki keterbatasan seperti sensitivitas terhadap parameter *tuning* dan prasyarat *preprocessing* data tertentu, yang tidak sejalan dengan keterbatasan sumber daya dan kebutuhan implementasi praktis di lingkungan pasar tradisional. Oleh karena itu, pemilihan *Gradient Boosting* dan *XGBoost* merupakan pilihan yang tepat dan strategis untuk mendukung pengambilan keputusan harga jual yang lebih objektif dan kompetitif di pasar tradisional.

Algoritma *LightGBM* dirancang sangat efisien untuk *dataset* berukuran besar, tetapi kinerjanya menurun ketika diterapkan pada *dataset* kecil. Dokumentasi model ini menyebutkan algoritma ini "*not ideal for small datasets*" karena pertumbuhan *tree* yang bersifat *leaf-wise* sering menghasilkan struktur pohon yang sangat dalam sehingga meningkatkan risiko *overfitting* apabila parameter tidak diatur dengan tepat. Selain itu, *LightGBM* mampu menangani data kategorikal, prosesnya tidak seotomatis *CatBoost* sehingga seringkali tetap memerlukan

encoding manual. Dari sisi interpretasi, *LightGBM* termasuk model *black box* yang sulit dipahami secara langsung tanpa dukungan metode interpretabilitas seperti *shap* atau *Lime* [10].

Sementara itu, *CatBoost* unggul dalam menangani variabel kategorikal secara *native* dan relatif stabil meskipun parameter *tuning* terbatas. Akan tetapi, *CatBoost* memiliki kelemahan berupa waktu *training* yang lebih lambat dibanding *LightGBM* ketika digunakan pada *dataset* yang besar. Keunggulan juga berkurang jika *dataset* didominasi oleh variabel numerik, karena dalam kondisi tersebut algoritma lain seperti *LightGBM* atau *XGBoost* justru dapat memberikan hasil lebih baik. Selain itu, *CatBoost* mengandalkan struktur data internal tertentu seperti *Pool* dan *Symmetric Tree*, yang dapat membatasi fleksibilitas integrasi dengan *pipeline* pemrosesan data yang kompleks. Walaupun mudah digunakan dengan parameter *default*, *CatBoost* relatif kurang dapat dikustomisasi bagi pengguna yang membutuhkan kontrol penuh atas *hyperparameter*. Sama seperti *LightGBM*, *CatBoost* juga sulit diinterpretasi tanpa bantuan metode tambahan [11].

Keputusan harga jual merupakan proses penentuan harga suatu produk atau barang yang akan ditawarkan kepada konsumen, dengan mempertimbangkan berbagai faktor seperti harga modal, margin keuntungan, permintaan keuntungan, tingkat persaingan dan kondisi pasar. Dalam konteks pasar tradisional, penentuan harga menjadi sangat penting karena pedagang menghadapi tantangan dalam menentukan harga agar tetap bersaing dengan kompetitor. Keputusan ini tidak hanya berdampak pada keberlangsungan usaha, tetapi juga pada persepsi konsumen terhadap nilai produk yang ditawarkan. Dengan menggunakan pendekatan berbasis

data seperti algoritma *Gradient Boosting*, proses penentuan harga jual dapat dioptimalkan untuk menghasilkan keputusan yang lebih objektif, strategis, dan sesuai dengan kondisi pasar yang dinamis.

Kontribusi utama penelitian ini adalah menghadirkan penerapan algoritma *Gradient Boosting* dan *XGBoost* sebagai solusi praktis dalam penentuan harga jual di pasar tradisional. Penelitian ini tidak hanya berfokus pada akurasi prediksi harga berbasis data historis penjualan, tetapi juga menekankan aspek interpretabilitas model melalui analisis *feature importance* dan *SHAP* (*Shapley Additive Explanations*). Dengan demikian, hasil prediksi tidak hanya memberikan pemahaman bagi pedagang mengenai faktor-faktor utama yang mempengaruhi penetapan harga. Kontribusi ini diharapkan dapat menjembatani kesenjangan literatur mengenai implementasi *Business Intelligence* di pasar tradisional, sekaligus menawarkan manfaat praktis berupa sistem pendukung keputusan harga yang adaptif dan dapat dijalankan pada sumber daya terbatas, sehingga pedagang tradisional mampu bersaing dengan pasar modern dan *e-commerce*.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, dapat diperoleh rumusan masalah yaitu:

- a. Bagaimana pengaruh variabel harga beli, harga kompetitor, permintaan pasar, ketersediaan barang, dan faktor musiman terhadap penentuan harga jual di toko pasar tradisional?

- b. Faktor-faktor apa saja yang paling berpengaruh terhadap penentuan harga jual berdasarkan hasil analisis *feature importance* dan *SHAP* pada model *Gradient Boosting* dan *XGBoost*?
- c. Bagaimana kinerja model *Gradient Boosting* dan *XGBoost* dalam memprediksi harga jual pada dua skenario dataset yang berbeda berdasarkan nilai R^2 dan MAPE.

1.3. Batasan Masalah

Berdasarkan latar belakang dan pernyataan rumusan masalah diatas, maka batasan masalah penelitian ini adalah:

- a. Penelitian ini dilakukan pada tiga toko yang berada di pasar tradisional, dengan satu toko dipilih sebagai fokus utama analisis, dan dua toko lainnya digunakan sebagai kompetitor.
- b. Data yang digunakan dalam penelitian ini terbatas pada data penjualan dan faktor-faktor yang mempengaruhi harga jual dalam beberapa bulan terakhir sebagai *dataset* kecil dan satu tahun terakhir sebagai *dataset* besar untuk menguji kinerja model pada skala data yang berbeda.
- c. Algoritma yang digunakan dalam penelitian ini menggunakan *Gradient Boosting Konvensional* dan *XGBoost* sebagai pembanding.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

- a. Menganalisis penerapan algoritma *Gradient Boosting* dalam menentukan harga jual di toko pasar tradisional.

- b. Mengidentifikasi faktor-faktor yang mempengaruhi keputusan harga jual di toko pasar tradisional.
- c. Mengevaluasi kinerja prediksi harga jual yang dihasilkan oleh algoritma *Gradient Boosting* dan *XGBoost* berdasarkan metrik R^2 dan *MAPE*

1.5. Manfaat Penelitian

Manfaat yang dihasilkan dari penelitian ini adalah:

- a. Memberikan wawasan bagi pemilik toko di pasar tradisional tentang penggunaan teknologi dalam menentukan harga jual.
- b. Menyediakan model prediksi harga jual yang dapat membantu pemilik toko dalam membuat keputusan harga yang lebih tepat.
- c. Menambah literatur ilmiah mengenai penerapan algoritma *Gradient Boosting* dalam.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Pada bab ini membahas tentang perbedaan dan persamaan dengan penelitian yang telah dilakukan sebelumnya, baik dari segi metode yang digunakan maupun komponen yang telah digunakan dalam penelitian. Kajian pustaka terhadap enam penelitian terdahulu dapat dijelaskan sebagai berikut :

Penelitian yang dilakukan oleh Yao Zao, Changcun Gao, Han Gao (2022) yang berjudul "*Business Failure Prediction Based on a Cost-Sensitive Extreme Gradient Boosting Machine*". Penelitian ini bertujuan untuk mengembangkan model prediksi menggunakan *XGBoost* melalui pendekatan *cost-sensitive* untuk mengatasi masalah ketidakseimbangan variabel *dataset* pada prediksi kegagalan bisnis. Model *XGBoost* memiliki fungsi objektif berbobot terbukti efektif dalam mengurangi *type-II error* atau kesalahan dalam mengklasifikasikan data Perusahaan. Model *XGBoost* memberikan hasil yang baik antara akurasi prediksi dan interpretabilitas. Penelitian selanjutnya diharapkan dapat mengeksplorasi dampak rasio non-keuangan terhadap kinerja *business failure prediction*. Sehingga dapat memberikan panduan yang lebih komprehensif bagi pembuat kebijakan untuk mengambil keputusan yang lebih baik [12].

Penelitian yang dilakukan oleh Biswanath Bhattacharjee, Sanjida Nowshin Mou, Md Shakhawot Hossain, Md Khalilur Rahman, Md Mehedi Hassan, Nabila Rahman, Refat Naznin, Md Ariful Islam Sarkar, Md Asaduzzaman, Md Sayem UI

Haque (2024) yang berjudul “*Machine Learning for Cost Estimation and Forecasting in Banking: A Comparative Analysis of Algorithm*”. Penelitian ini bertujuan untuk mengevaluasi berbagai kinerja *machine learning* yang digunakan untuk estimasi biaya dan prediksi di sektor perbankan. Penelitian ini menggunakan algoritma *Linear Regression*, *Ridge Regression*, *Random Forest*, *Gradient Boosting Machine (GBM)*, dan *Long Short Term Memory (LSTM)* untuk mengidentifikasi model yang paling optimal. Hasil penelitian menunjukkan bahwa model *Gradient Boosting* menunjukkan peningkatan signifikan dalam kinerja dibandingkan *Linear Regression*, dengan kemampuan lebih baik menangkap variabel kompleks. Penelitian selanjutnya diharapkan berfokus pada pendekatan *hybrid* yang menggabungkan beberapa model untuk mencapai kinerja optimal [13].

Penelitian yang dilakukan oleh Nadisadat Taghavi Razavizadeh, Maryam Salari, Mostafa Jafari, Ehsan Sabaghian, Vahid Ghavami (2023) yang berjudul “*Comparison of Two Methods, Gradient Boosting and Extreme Gradient Boosting to Predict Survival in Covid-19 Data*”. Penelitian ini bertujuan untuk membandingkan komprehensif antara algoritma *Gradient Boosting* dan *XGBoost* dalam memprediksi kelangsungan hidup pasien covid-19. Penelitian ini mengevaluasi kinerja kedua modal dalam tiga aspek yang meliputi akurasi prediksi, kalibrasi model, dan kecepatan komputasi. Kesimpulan dari penelitian ini yaitu kedua model memiliki karakteristik kinerja model yang serupa dalam hal akurasi dan kekuatan diskriminasi, namun model *XGBoost* menunjukkan hasil kalibrasi yang relatif lebih baik dibandingkan *Gradient Boosting*. Penelitian selanjutnya

disarankan menggunakan variasi data dan fitur yang lebih besar, serta mengeksplorasi metode kalibrasi yang lebih canggih [6].

Penelitian yang dilakukan oleh Dylan Norbert Gono, Herlina Napitupulu, Firdaniza (2023) yang berjudul "*Silver Price Forecasting Using Extreme Gradient Boosting (XGBoost) Method*". Penelitian ini bertujuan untuk memprediksi harga perak dengan Teknik *grid search* menggunakan model *XGBoost*. Penelitian ini menggunakan data deret waktu harga perak, emas, platinum, dan nilai tukar *euro-dollar* selama sepuluh tahun. Penelitian ini menghasilkan dua model terbaik berdasarkan metrik *Mean Absolute Percentage Error (MAPE)* dan *Root Mean Square Error (RMSE)*. Penelitian selanjutnya disarankan mengeksplorasi *hyperparameter* tambahan, Teknik pra-pemrosesan dan variabel tambahan untuk meningkatkan akurasi prediksi[14].

Penelitian yang dilakukan oleh Odey Alshboul, Ali Shehadeh, Ghassan Almasabha, Ali Saeed Almuflih (2022) yang berjudul "*Extreme Gradient Boosting-Based Machine Learning Approach for Green Building Cost Prediction*". Penelitian ini bertujuan untuk mengembangkan model prediksi biaya Pembangunan Gedung menggunakan algoritma *XGBoost*, *Deep Neural Network (DNN)*, dan *Random Forest (RF)*. Hasil dari penelitian ini menunjukkan bahwa model *XGBoost* memiliki kinerja terbaik, dengan nilai R^2 sebesar 0,96. Penelitian selanjutnya disarankan untuk memperluas *dataset*, mempertimbangkan dukungan eksternal, dan memperluas analisis biaya ke seluruh siklus hidup Gedung [15].

Penelitian yang dilakukan oleh Xiaolin Chen, Junting Hua, Chong Yin, Zhang Chen, Yangming Chen, Zhonghao Li (2023) yang berjudul "*A Precise Life*

Estimation Method for Retired Energy Storage Batteries Based on Energy Storage Batteries Attenuation Characteristics and XGBoost Algorithm". Penelitian ini bertujuan untuk mengembangkan model estimasi yang akurat untuk menilai umur baterai yang sudah habis masa pakainya dengan mempertimbangkan karakteristik atenuasi baterai yang mencerminkan kondisi operasi aktual. Algoritma *XGBoost* yang digunakan terbukti lebih efektif dibandingkan dengan *neural network conventional*. Metode ini hanya fokus pada segmen pengisian baterai untuk estimasi umur baterai, sehingga perlu eksplorasi lebih lanjut untuk integrasi data dari berbagai kondisi operasi [8].



2.2. Keaslian Penelitian

Tabel 2. 1. Matriks *Literature Review*
 Analisis Penentuan Harga Jual Dengan Algoritma *Gradient Boosting* di Toko Pasar Tradisional

No	Judul	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
1	<i>Business Failure Prediction Based on a Cost-Sensitive Extreme Gradient Boosting Machine</i>	Yao Zou, Changchun Gao, Han Gao IEEE 2022	<i>XGBoost</i>	Model <i>XGBoost</i> dengan fungsi objektif berbobot terbukti efektif dalam mengurangi kesalahan prediksi pada Perusahaan. Model <i>Cost Sensitive</i> yang diusulkan secara signifikan mengurangi <i>type-II error</i> yaitu kesalahan dalam mengklasifikasikan data Perusahaan yang gagal. Model yang digunakan memberikan hasil yang baik antara akurasi prediksi dan interpretabilitas, sehingga dapat menjadi alat yang baik untuk membantu investor dalam mengambil Keputusan investasi.	Keunggulan : Model <i>XGBoost cost-sensitive</i> mengurangi <i>Type-II error</i> dan memberikan keseimbangan baik antara akurasi dan interpretabilitas. Kekurangan : Terbatas pada interpretasi <i>post-hoc</i> , belum mempertimbangkan faktor non-keuangan, dan kurang diuji pada <i>dataset</i> lintas industri.	Penelitian ini menggunakan model <i>XGBoost</i> dengan pendekatan <i>cost-sensitive</i> untuk mengatasi ketidakseimbangan variabel dataset, sedangkan penelitian yang akan dilakukan menggunakan model <i>gradient boosting</i> dan <i>XGBoost</i> untuk menangani <i>dataset non-linear</i> yang digunakan untuk membantu menghitung estimasi harga jual.

No	Judul	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
2	<i>Machine Learning for Cost Estimation and Forecasting in Banking: A Comparative Analysis of Algorithms</i>	Biswanath Bhattarjee, Sanjida Nowadin Mৌ, Md Shakhawat Hossain, Md Khalilur Rahman, Md Mehedi Hassan, Nabila Rahman, Refat Naznin, Md Ariful Islam Sarkar, Md Asaduzzaman, Md Sayem Ul Haque Frontline Journal 2024	<i>Linear Regression, Ridge Regression, Random Forest, Gradient Boosting Machine (GBM), dan Long Short Term Memory (LSTM).</i>	Hasil penelitian ini menunjukkan bahwa model <i>LSTM</i> memberikan kinerja terbaik dengan nilai R^2 tertinggi serta nilai <i>MAE</i> dan <i>MSE</i> terendah dibandingkan dengan model lainnya. Sedangkan model <i>Gradient Boosting</i> menunjukkan peningkatan signifikan dalam kinerja dibandingkan model <i>Linear Regression</i> , dengan kemampuan lebih baik menangkap variabel kompleks, meskipun dengan biaya komputasi yang lebih tinggi.	Keunggulan : Model <i>Gradient Boosting</i> memberikan akurasi tinggi $R^2 = 0.84$ dengan kemampuan menangkap hubungan <i>non-linear</i> dan kompleks, stabil, adaptif, serta relatif interpretatif dibanding model <i>neural network</i> . Kekurangan : Memerlukan waktu komputasi tinggi, berisiko <i>overfitting</i> pada data terbatas, dan sangat bergantung pada <i>tuning</i> parameter yang kompleks.	Penelitian ini menggunakan berbagai <i>machine learning</i> untuk estimasi biaya dan prediksi di sektor perbankan salah satunya menggunakan algoritma <i>Gradient Boosting</i> untuk mengidentifikasi model yang paling optimal berdasarkan akurasi, efisiensi komputasi, dan interpretabilitas. Sedangkan penelitian yang akan dilakukan menggunakan algoritma <i>Gradient Boosting</i> dan <i>XGBoost</i> untuk estimasi harga jual suatu barang.
3	<i>Comparison of Two Methods, Gradient Boosting and Extreme Gradient Boosting to Predict Survival in Covid-19 Data</i>	Nadisadat Taghavi Razavizadeh, Maryam Salari, Mostafa	<i>Gradient Boosting dan XGBoost.</i>	Kesimpulan dari penelitian ini yaitu kedua model <i>machine learning</i> memiliki karakteristik kinerja yang serupa dalam hal	Keunggulan : <i>XGBoost</i> memiliki kalibrasi dan akurasi tinggi (<i>C-index</i> 0,894), efisien secara komputasi (47x lebih cepat dari	Penelitian ini menggunakan algoritma <i>machine learning</i> yaitu <i>Gradient Boosting</i> dan <i>XGBoost</i> untuk memprediksi kelangsungan hidup pasien <i>Covid-19</i> . Sedangkan

No	Judul	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
		Jafari, Ehsan Sabughian, Vahid Ghayami Journal of Biostatistic and Epidemiology 2023		akurasi prediksi dan kekuatan diskriminasi. Namun, terdapat perbedaan yang signifikan dimana model <i>XGBoost</i> menunjukkan kalibrasi yang relatif lebih baik dibandingkan dengan <i>gradient boosting</i> .	<i>Gradient Boosting</i> dan mampu mengatasi ketidakseimbangan data melalui data melalui regularisasi L1+L2. Kelemahan : <i>Gradient Boosting</i> kurang terkalibrasi dan sensitif terhadap outlier, <i>XGBoost</i> memerlukan <i>many</i> parameter kompleks dan sumber daya komputasi besar.	penelitian yang akan dilakukan menggunakan algoritma <i>machine learning</i> yaitu <i>Gradient Boosting</i> dan <i>XGBoost</i> untuk estimasi harga jual barang di toko tradisional.
4	<i>Silver Price Forecasting Using Extreme Gradient Boosting (XGBoost) Method</i>	Dylan Norbert Gono, Herlina Napitupuhu, Firdaniza MDPI 2023	<i>XGBoost</i>	Metodologi penelitian melibatkan penggunaan teknik <i>grid search</i> untuk mengoptimalkan empat <i>hyperparameter</i> : <i>learning rate</i> , <i>max_depth</i> , <i>n_estimators</i> , dan <i>gamma</i> . Setelah proses <i>tuning</i> , peneliti menghasilkan dua model terbaik yang dinilai berdasarkan metrik <i>MAPE (Mean Absolute Percentage Error)</i> dan <i>RMSE (Root</i>	Keunggulan : Akurasi tinggi (<i>MAPE</i> 5,98%; <i>RMSE</i> 1,69), validasi <i>5-fold cross-validation</i> , dan performanya lebih baik dari <i>CatBoost</i> dan <i>Random Forest</i> . Kelemahan : Belum menguji <i>hyperparameter</i> lanjutan, cakupan prediksi terbatas (6 hari), dan belum mengevaluasi interpretabilitas variabel model.	Penelitian ini menggunakan model <i>XGBoost</i> untuk memprediksi harga perak dengan <i>dataset</i> yang terdiri dari deret waktu, harga perak, emas, platinum dan nilai tukar <i>euro-dollar</i> . Sedangkan penelitian yang akan dilakukan menggunakan model <i>XGBoost</i> untuk estimasi harga barang dengan <i>dataset non-linear</i> seperti burung, tanggal, harga kompetitor. Harga modal, harga jual saat ini.

No	Judul	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
				<p><i>Mean Square Error</i>). Model A memiliki <i>MAPE</i> 5,98% dan <i>RMSE</i> 1,6998, dengan prediksi harga perak yang berfluktuasi naik-turun selama enam hari ke depan. Model B memiliki <i>MAPE</i> 6,06% dan <i>RMSE</i> 1,6967, dengan prediksi penurunan harga hingga hari ketiga dan kemudian peningkatan hingga hari keenam.</p>		
5	<i>Extreme Gradient Boosting-Based Machine Learning Approach for Green Building Cost Prediction</i>	Odey Alshboul, Ali Shehadeh, Ghaman Almasabbah, Ali Saeed Almuflih MDPI 2022	<i>XGBoost</i> , <i>Deep Neural Network(DNN)</i> , dan <i>Random Forest (RF)</i>	<p>Penelitian ini melibatkan pengumpulan data dari 283 proyek Gedung bersertifikat di <i>Amerika Utara</i>. Peneliti menggunakan Teknik pra-pemrosesan data, seleksi fitur, dan validasi untuk mengembangkan model prediksi. Empat metrik evaluasi (<i>MAE</i>, <i>RMSE</i>, <i>MAPE</i>, dan <i>R²</i>)</p>	<p>Keunggulan : Akurasi tinggi, regularisasi kuat, efisien, dan unggul dalam mengidentifikasi faktor biaya <i>green building</i>.</p> <p>Kekurangan : Data terbatas secara geografis, tanpa faktor sosial/lingkungan, belum mencakup <i>life cycle cost</i>, dan interpretabilitas terbatas.</p>	<p>Pada penelitian ini digunakan model <i>XGBoost</i>, <i>Deep Neural Network(DNN)</i>, dan <i>Random Forest (RF)</i> untuk mengatasi kesulitan dalam memperkirakan biaya konstruksi Gedung dengan mempertimbangkan beberapa atribut. Penelitian yang akan dilakukan menggunakan model <i>Gradient Boosting</i> dan <i>XGBoost</i> untuk memperkirakan harga jual</p>

No	Judul	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
				digunakan untuk menilai kinerja model. Hasil penelitian menunjukkan bahwa model <i>XGBoost</i> memiliki kinerja terbaik, dengan nilai R^2 sebesar 0,96 dan nilai terendah untuk <i>MAE</i> , <i>RMSE</i> , dan <i>MAPE</i> .		yang paling optimal dengan beberapa faktor pendukung.
6	<i>A Precise Life Estimation Method for Retired Energy Storage Batteries Based on Energy Storage Batteries Attenuation Characteristics and XGBoost Algorithm</i>	Xiuolin Chen, Junting Hua, Chong Yin, Zhang Chen, Yangming Chen, Zhonghao Li IEEE 2023	<i>XGBoost</i>	Algoritma <i>XGBoost</i> yang digunakan terbukti lebih efektif daripada jaringan saraf (<i>neural network</i>) konvensional, dengan peningkatan akurasi hingga 0,755% dalam <i>MAPE</i> (<i>Mean Absolute Percentage Error</i>).	Keunggulan : Akurasi tinggi, efisien, bebas model fisika, stabil, unggul dari <i>Neural Network</i> . Kekurangan : <i>Dataset</i> terbatas, hanya <i>lithium-ion</i> , kurang interpretabilitas fitur, tidak mencakup <i>discharge cycle</i> .	Tujuan penelitian ini untuk mengembangkan model estimasi yang akurat untuk menilai umur baterai yang telah habis pakai menggunakan model <i>XGBoost</i> . Sedangkan penelitian yang akan dilakukan menggunakan model <i>XGBoost</i> untuk estimasi harga jual barang.

2.3. Landasan Teori

2.3.1 Pasar Tradisional

Pasar tradisional merupakan sebuah tempat atau media berlangsungnya jual beli barang antara pedagang dan pembeli dan menjadi pusat perekonomian masyarakat setempat. Pasar tradisional sangat dibutuhkan masyarakat guna memenuhi kebutuhan *primer* maupun *sekunder* [16]. Pasar tradisional memiliki karakteristik yang berbeda dengan pasar modern contohnya seperti tawar-menawar, lebih mudah ditemukan, pasar tradisional menyediakan berbagai barang yang ditawarkan dengan harga yang lebih murah dan memiliki jam operasional yang pendek [17].

2.3.2 Harga Jual

Harga Jual adalah harga yang ditetapkan oleh penjual kepada pembeli untuk mendapatkan produk atau barang yang diinginkan. Penetapan harga jual didasarkan oleh beberapa aspek seperti keterjangkauan harga, kesesuaian harga dengan kualitas daya saing, dan kesesuaian harga dengan manfaat yang didapatkan. Harga jual yang kompetitif dapat memberikan keuntungan kepada pembeli maupun penjual sehingga tidak ada yang dirugikan dalam penentuan harga jual [18].

2.3.3 Gradient Boosting

Gradient Boosting adalah metode *machine learning* atau pembelajaran mesin yang membangun model prediktif dengan menggabungkan beberapa model lemah (*weak learners*) secara bertahap. Proses ini dilakukan melalui pendekatan yang disebut *boosting*, di mana model-model yang dihasilkan dalam setiap iterasi memperbaiki kesalahan model sebelumnya. Pada dasarnya, *Gradient Boosting*

memperbaiki dan mengoptimalkan model dengan menggunakan Teknik statistik dan probabilistik untuk memodelkan hubungan antara *variabel input (fitur)* dan *output (target)*. Setiap model yang dibangun *Gradient Boosting* relatif sederhana dan berupa pohon keputusan (*decision tree*), yang belajar dari kesalahan sebelumnya. Perbaruan model ini dilakukan dengan cara menambahkan model baru yang fokus pada kesalahan yang dihasilkan dari model sebelumnya. Hal ini memungkinkan algoritma ini untuk menangani data yang kompleks dengan baik [6].

2.3.4 XGBoost

XGBoost adalah pengembangan lebih lanjut dari algoritma *Gradient Boosting*. *XGBoost* menerapkan prinsip yang sama dengan *Gradient Boosting*, tetapi dengan beberapa perbaikan untuk meningkatkan akurasi, kecepatan, dan ketahanannya terhadap *overfitting*. *XGBoost* memiliki dua jenis regularisasi yang memperkuat kemampuannya untuk menggeneralisasi data, yaitu *L1 (Lasso Regression)* dan *L2 (Ridge Regression)*. Regularisasi ini menambahkan penalti pada model untuk menghindari *overfitting*, yaitu saat model terlalu cocok dengan data pelatihan dan gagal dalam memprediksi data baru yang belum terlihat sebelumnya. Dengan adanya regularisasi *L1* dan *L2*, *XGBoost* menjadi model algoritma yang lebih kuat dalam menangani data yang lebih kompleks dan mengurangi risiko *overfitting*, terutama dalam dataset yang besar dan tidak terstruktur [6].

2.3.5 *GridSearchCV*

GridSearchCV adalah metode *hyperparameter tuning* dalam *machine learning* yang biasa digunakan untuk mencari kombinasi parameter terbaik dari suatu algoritma untuk meningkatkan kinerja model. *GridSearchCV* dilakukan dengan membandingkan berbagai kombinasi parameter menggunakan Teknik *cross-validation* untuk memastikan model tidak hanya unggul pada data penelitian, tetapi juga memiliki kinerja generalisasi yang baik pada data baru. Penggunaan *GridSearchCV* pada algoritma *Gradient Boosting* dan *XGBoost* meliputi pemilihan nilai *hyperparameter* seperti *learning_rate*, *max_depth*, *n_estimators*, dan *subsample* sangat mempengaruhi akurasi dan ketahanan model terhadap *overfitting*. *GridSearchCV* memungkinkan eksplorasi sistematis terhadap ruang parameter tersebut, sehingga model yang dihasilkan memiliki kinerja prediksi yang lebih optimal [19].

2.3.6 *SHAP (SHapley Additive exPlanations)*

SHAP atau *Shapley Additive exPlanations* adalah metode interpretabilitas model *machine learning* yang berbasis pada teori permainan, khususnya konsep nilai *Shapley*. *SHAP* biasanya digunakan untuk mengukur kontribusi setiap fitur terhadap hasil prediksi model. Metode ini termasuk dalam kategori *additive feature attribution* yang dapat diaplikasikan pada berbagai model. *SHAP* menghitung rata-rata absolut dari kontribusi setiap fitur, lalu menyusunnya berdasarkan Tingkat kepentingan. Fitur dengan nilai *SHAP* yang tinggi dianggap paling berpengaruh terhadap *output* model. *SHAP* memberikan pemahaman mendalam tentang

bagaimana setiap fitur memengaruhi prediksi, namun membutuhkan komputasi tinggi, terutama untuk *dataset* besar [20].

2.3.7 R^2 (Coefficient of Determination)

R^2 atau *Coefficient of Determination* adalah metrik statistik yang digunakan untuk mengukur seberapa baik variable independen menjelaskan variasi dari variabel dependen dalam model regresi. Berikut persamaan 3 untuk mencari *Coefficient of Determination* atau R^2 :

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = \frac{SS_{reg}}{SS_{tot}} \quad (2.1)$$

Keterangan :

$$SS_{res} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \text{Residual sum of squares}$$

$$SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2 = \text{total sum of squares}$$

$$SS_{reg} = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 = \text{regression sum of squares}$$

Persamaan 3 digunakan untuk mencari nilai R^2 yang berada diantara 0 hingga 1. Nilai yang tinggi menunjukkan bahwa model memiliki variasi data yang lebih baik. Namun, R^2 sering disalahartikan sebagai persentase variasi aktual yang dijelaskan model [21].

2.3.8 MAPE (Mean Absolute Percentage Error)

MAPE atau *Mean Absolute Percentage Error* merupakan metrik evaluasi yang digunakan untuk mengukur akurasi model prediksi dalam bentuk persentase. *MAPE* menghitung rata-rata dari persentase kesalahan absolut antara nilai prediksi dan nilai actual. Berikut persamaan 4 untuk mencari *MAPE* :

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (2.2)$$

Keterangan :

y_i = nilai aktual ke- i

\hat{y}_i = nilai prediksi ke- i

n = jumlah data

MAPE memiliki keunggulan dalam memberikan interpretasi kesalahan prediksi dalam bentuk persentase, sehingga lebih intuitif digunakan dalam membandingkan kinerja model. Namun, *MAPE* bisa menjadi tidak stabil apabila terdapat nilai aktual yang sangat kecil atau mendekati nol, karena akan memperbesar error [22].

2.3.9 Python

Python merupakan Bahasa pemrograman Tingkat tinggi yang memiliki sifat interpreted, interaktif, dan berorientasi objek. *Python* dibuat agar mudah dipahami dan digunakan dengan sintaks yang mirip dengan bahasa inggris, serta lebih sedikit aturan penulisan dibandingkan dengan Bahasa lain. *Python* merupakan pemrograman yang fleksibel digunakan untuk pemrograman terstruktur, fungsional, dan *OOP (object-oriented programming)*. *Python* juga menyediakan banyak *library* atau Pustaka dan *framework* bawaan, serta mendukung manajemen memori otomatis atau *garbage collection* dan *dynamic typing* [23].

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

3.1.1 Jenis Penelitian

Penelitian ini menggunakan metode Penelitian Tindakan (*Action Research*), yang bertujuan untuk menerapkan dan mengevaluasi penerapan algoritma *Gradient Boosting* dan *XGBoost* beserta regularisasi dalam menentukan harga jual di toko tradisional. Penelitian ini dilakukan dengan cara mengamati dan menganalisis hasil dari penerapan metode tersebut menggunakan historis dari objek penelitian, yaitu toko di pasar tradisional. Data yang digunakan pada penelitian ini tidak dibuat langsung oleh peneliti melainkan mengambil histori atau data penjualan, wawancara dengan pengelola toko secara langsung untuk memastikan validitas dan akurasi.

3.1.2 Sifat Penelitian

Penelitian ini bersifat kausal komparatif, karena tujuan utama dalam penelitian ini yaitu untuk menganalisis hubungan sebab-akibat dari berbagai faktor yang mempengaruhi keputusan harga jual menggunakan algoritma *Gradient Boosting*. Penelitian ini dilaksanakan untuk memahami bagaimana variabel-variabel seperti harga modal, permintaan konsumen, dan ketersediaan barang memengaruhi harga jual yang ditetapkan. Dengan menggunakan pendekatan observasional, data historis akan dianalisis untuk mengidentifikasi hubungan kausal di antara variabel-variabel tersebut. Hasil analisis dari penelitian yang akan

dilakukan dapat memberikan wawasan yang signifikan mengenai pola hubungan antar variabel, yang nantinya dapat menjadi dasar untuk pengambilan keputusan harga yang lebih efektif.

3.1.3 Pendekatan Penelitian

Pendekatan yang digunakan pada penelitian ini adalah pendekatan kuantitatif. Pendekatan ini dipilih karena penelitian berfokus pada pengolahan data numerik dan pengukuran yang terukur untuk menganalisa hubungan antar variabel dalam penentuan harga jual. Penelitian ini melibatkan pengumpulan data historis penjualan, harga modal, dan faktor-faktor lain yang mempengaruhi harga jual di toko pasar tradisional.

Melalui pendekatan kuantitatif, penelitian ini menggunakan algoritma *Gradient Boosting* dan *XGBoost* sebagai alat analisis untuk mengevaluasi hubungan antar variabel *independen* (seperti harga modal, permintaan konsumen, ketersediaan barang, dan harga kompetitor) dengan variabel *dependen* (harga jual). Pendekatan ini memungkinkan penelitian menghasilkan model prediktif yang dapat diuji akurasi dengan metrik evaluasi yang terstruktur seperti R^2 dan *MAPE*.

3.2. Metode Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang dibutuhkan untuk penelitian yang akan dilakukan. Data yang digunakan berupa data numerik penjualan barang, sebelumnya akan dilakukan *labeling* dan normalisasi data kategori menjadi numerik. Data yang akan dilakukan *labeling* terdiri dari permintaan pasar, ketersediaan barang, dan musim. Permintaan pasar terdiri dari tiga kategori seperti tinggi, sedang, dan rendah. Ketersediaan barang terdiri dari dua

kategori yaitu mudah dan langka. Musim terdiri dari dua kategori yaitu hari biasa dan hari raya. Berikut metode pengumpulan data yang digunakan :

1) Wawancara (*Interview*)

Metode wawancara adalah pengumpulan data yang dilakukan melalui tanya jawab dengan pemilik toko yang dilakukan secara sistematis. Wawancara dilakukan untuk melengkapi informasi yang diperlukan seputar permasalahan yang mendalam sehingga diperoleh data yang akurat dan terpercaya karena dilakukan secara langsung tanpa perantara.

2) Observasi

Metode observasi adalah metode pengumpulan data melalui pengamatan secara langsung terhadap permasalahan yang ada. Observasi yang dilakukan adalah observasi secara langsung, yaitu untuk memperoleh data dari Pasar Tradisional.

3) Dokumentasi

Metode dokumentasi pada penelitian ini digunakan untuk mengumpulkan data historis penjualan yang didapatkan langsung dari pemilik toko.

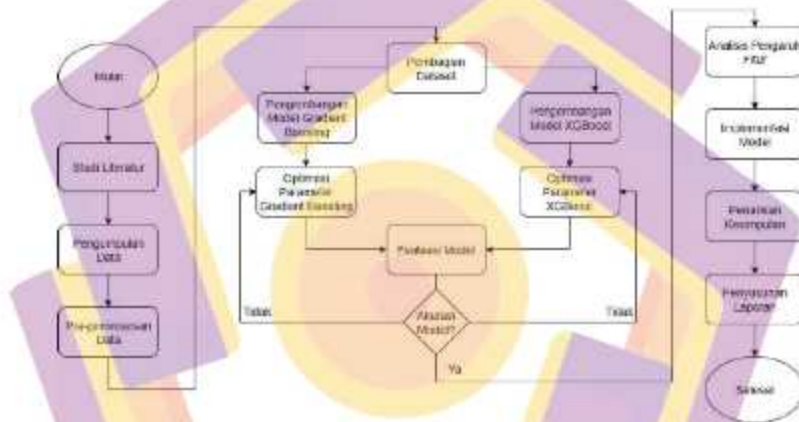
3.3. Metode Analisis Data

Metode analisis data pada penelitian ini dengan penerapan algoritma *Gradient Boosting* dan *XGBoost* dalam menentukan harga jual di toko pasar tradisional, analisis akan meliputi pra-pemrosesan data, implementasi kedua model dengan parameter optimal, dan evaluasi kinerja model. Selanjutnya, faktor-faktor yang mempengaruhi keputusan harga jual akan menggunakan perhitungan *feature importance*, analisis *SHAP*, dan analisis korelasi antar variabel untuk

mengidentifikasi kontribusi faktor yang paling signifikan. Evaluasi model akan mencakup perhitungan metrik evaluasi seperti R^2 dan $MAPE$.

3.4. Alur Penelitian

Penelitian ini dilaksanakan dengan mengikuti alur sistematis yang mencakup beberapa tahapan utama. Berikut adalah diagram alur penelitian yang menggambarkan keseluruhan proses penelitian :



Gambar 3. 1. Alur Penelitian

Pada gambar 3.1 menyajikan alur sistematis dari proses penelitian dan tahapan pengambilan data yang dilakukan. Diagram Alur Penelitian yang akan dijelaskan sebagai berikut :

1. Studi Literatur

Tahap ini peneliti menentukan mencari informasi dengan membaca artikel ilmiah, jurnal, dan buku yang memiliki kaitan dengan

algoritma estimasi yang digunakan yaitu *Gradient Boosting* dan *XGBoost*.

2. Pengumpulan data

Proses pengumpulan data merupakan tahap yang dilakukan dengan teknik wawancara bersama pemilik toko dan observasi yang dilakukan secara langsung di toko pasar tradisional. *Dataset* ini mencakup 12 atribut, berikut atribut yang digunakan pada penelitian ini

Tabel 3. 1. Atribut *Dataset*

No	Atribut	Deskripsi
1	Produk	Nama Produk
2	Tanggal	Tanggal Harga Produk
3	Harga Beli	Harga Beli Produk
4	Kompetitor 1	Harga Kompetitor 1
5	Kompetitor 2	Harga Kompetitor 2
6	Permintaan Pasar	Permintaan Pelanggan
7	Ketersediaan Barang	Ketersediaan Barang di Pasar
8	Musim	Musim Penjualan Berlangsung
9	Volume Pembelian	Jumlah pembelian pelanggan
10	Harga Jual	Harga Beli yang sudah ditambahkan margin
11	Margin	Keuntungan penjualan
12	Margin Persentase	Keuntungan Penjualan dalam bentuk persen

Tabel 3.1. menampilkan 12 atribut yang digunakan dalam dataset penelitian yang terdiri dari produk, tanggal, harga beli, kompetitor 1, kompetitor 2, permintaan pasar, ketersediaan barang, musim, volume pembelian, harga jual, margin, dan margin persentase. Pada atribut dataset selanjutnya dilakukan *labeling*. Setiap rekaman data diberi label dengan variabel tertentu untuk mendukung pelatihan

model, termasuk produk (V1), tanggal (V2), harga pembelian (V3), harga kompetitor 1 (V4), harga kompetitor 2 (V5), permintaan pasar (V6), ketersediaan barang (V7), musim (V8), volume pembelian (V9), harga jual (V10), margin (V11), dan persentase margin (V12).

3. Pra-pemrosesan Data

Pada tahap ini setelah data terkumpul selanjutnya dilakukan penanganan data yang hilang atau *missing value*, normalisasi data kategorikal menjadi numerik, dan seleksi fitur.

4. Pembagian Dataset

Pembagian dataset menjadi *training data* (80%) dan *testing data* (20%) yang nantinya akan digunakan untuk pengembangan model *Gradient Boosting* dan *XGBoost*.

5. Pengembangan Model

Pada penelitian ini digunakanlah model *Gradient Boosting* konvensional dan *XGBoost* yang dilengkapi dengan regularisasi *L1* dan *L2*.

6. Optimasi Model

Pada tahap ini untuk model *Gradient Boosting* maupun *XGBoost* menggunakan *tuning hyperparameter* dengan *grid search*.

7. Evaluasi Model

Kinerja kedua model akan dievaluasi menggunakan R^2 dan *MAPE*.

8. Akurasi Model

Pada tahap ini setelah dilakukan evaluasi model apabila hasil akurasi mencukupi maka selanjutnya dilakukan analisis pengaruh fitur, apabila tidak mencukupi hasilnya maka dilakukan optimasi Kembali.

9. Analisis Pengaruh Fitur

Analisis pengaruh fitur menggunakan *feature importance* untuk mengidentifikasi faktor-faktor yang paling berpengaruh dan analisis *SHAP* untuk interpretasi model.

10. Implementasi Model

Pada tahap ini dilakukan percobaan untuk menentukan harga jual.

11. Penarikan Kesimpulan

Pada tahap ini menarik Kesimpulan dari temuan utama penelitian.

12. Dokumentasi Penelitian

Penyusunan laporan tesis dan materi publikasi.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Pada bagian ini menggunakan algoritma *Gradient Boosting* dan *XGBoost* untuk menentukan hasil dari penelitian yang dilakukan. Pada tahap ini digunakan *tuning hyperparameter* menggunakan *GridSearchCV* untuk mendapatkan kombinasi parameter terbaik guna meningkatkan kinerja model. Kinerja kedua model akan dievaluasi menggunakan metrik evaluasi seperti *Koefisien Determinasi (R^2)* dan *Mean Absolute Percentage Error (MAPE)* untuk mengukur akurasi dan kemampuan generalisasi model terhadap data uji.

4.1. Pengumpulan *Dataset*

Dataset yang digunakan pada penelitian ini didapatkan melalui wawancara tanya jawab dengan pemilik toko yang dilakukan secara sistematis guna melengkapi informasi yang diperlukan seputar permasalahan yang mendalam sehingga diperoleh data yang akurat, kemudian dilakukan observasi secara langsung untuk memperoleh data dari pasar tradisional, dan dokumentasi yaitu mengumpulkan data historis penjualan yang didapatkan langsung dari pemilik toko. Berikut *dataset* besar pada penelitian ini :

Tabel 4. 1. *Dataset* Besar

No	Produk	Tanggal	Harga Beli	Kompetitor 1	Kompetitor 2
1	Mie Sukses Kuah	2024-01-01	78.100	84.000	79.500
2	Mie Sukses Kuah	2024-01-01	3.255	3.500	3.100
3	Saori 270 ml	2024-01-01	242.000	247.400	247.500
4	Saori 270 ml	2024-01-01	20.166	22.400	22.000
5	Roma Sandwich	2024-01-01	244.500	248.700	248.600
...
...
...
5296	Fresco Kopi Gula	2024-12-30	12.200	12.800	13.900
5297	Milku	2024-12-30	31.000	32.100	31.800

5298	Milku	2024-12-30	2.900	3.500	3.300
5299	Hemart 900 ml	2024-12-30	209.600	222.300	212.300
5300	Hemart 900 ml	2024-12-30	18.400	23.000	22.100

Tabel 4. 1. *Dataset Besar* (Lanjutan)

Permintaan Pasar	Ketersediaan Barang	Musim	Volume Pembelian	Harga Jual	Margin	Margin Persentase
Sedang	Mudah	Hari Biasa	24	81.000	2.900	3,58
Sedang	Mudah	Hari Biasa	1	3.500	245	7,00
Sedang	Mudah	Hari Biasa	12	243.500	1.500	0,62
Tinggi	Mudah	Hari Biasa	1	21.000	834	3,97
Rendah	Mudah	Hari Biasa	42	246.500	2.000	0,81
---	---	---	---	---	---	---
---	---	---	---	---	---	---
---	---	---	---	---	---	---
Tinggi	Mudah	Hari Raya	1	13.500	1.300	9,63
Rendah	Mudah	Hari Raya	12	33.300	2.300	6,91
Sedang	Mudah	Hari Raya	1	3.500	400	12,12
Rendah	Mudah	Hari Raya	12	219.000	9.400	4,29
Rendah	Mudah	Hari Raya	1	21.500	3.100	14,42

Tabel 4.1 menampilkan *dataset* skala besar yang terdiri dari 5.300 data penjualan mingguan yang dikumpulkan selama periode satu tahun tepatnya tahun 2024. Berikut *dataset* kecil sebagai pembanding dari *dataset* besar :

Tabel 4. 2. *Dataset Kecil*

No	Produk	Tanggal	Harga Beli	Kompetitor 1	Kompetitor 2
1	Mie Sukses Kuah	2024-01-01	78.100	84.000	79.500
2	Mie Sukses Kuah	2024-01-01	3.255	3.500	3.100
3	Saori 270 ml	2024-01-01	242.000	247.400	247.500
4	Saori 270 ml	2024-01-01	20.166	22.400	22.000
5	Roma Sandwich	2024-01-01	244.500	248.700	248.600
---	---	---	---	---	---
---	---	---	---	---	---
---	---	---	---	---	---
496	Fresco Kopi Gula	2024-01-29	12.135	12.500	13.200
497	Milku	2024-01-29	31.500	31.300	33.600
498	Milku	2024-01-29	2.625	2.500	2.700
499	Hemart 900 ml	2024-01-29	212.000	218.400	211.600
500	Hemart 900 ml	2024-01-29	17.666	20.500	21.400

Tabel 4. 2. *Dataset Kecil* (Lanjutan)

Permintaan Pasar	Ketersediaan Barang	Musim	Volume Pembeian	Harga Jual	Margin	Margin Persentase
Sedang	Mudah	Hari Biasa	24	81.000	2.900	3,58
Sedang	Mudah	Hari Biasa	1	3.500	245	7,00
Sedang	Mudah	Hari Biasa	12	243.500	1.500	0,62
Tinggi	Mudah	Hari Biasa	1	21.000	834	3,97
Rendah	Mudah	Hari Biasa	42	246.500	2.000	0,81
....
....
....
Rendah	Mudah	Hari Biasa	1	13.000	865	6,65
Sedang	Mudah	Hari Biasa	12	32.400	900	2,78
Rendah	Mudah	Hari Biasa	1	3.000	375	12,50
Sedang	Mudah	Hari Biasa	12	213.000	3.000	1,40
Rendah	Mudah	Hari Biasa	1	20.000	2.334	11,67

Pada tabel 4.1 dan tabel 4.2 merupakan *dataset* yang digunakan dalam penelitian ini terdiri dari beberapa fitur penting yang merepresentasikan kondisi penjualan di pasar tradisional. Adapun atribut atau kolom dalam *dataset* meliputi : Produk, Tanggal, Harga Beli, Harga Kompetitor 1, Harga Kompetitor 2, Permintaan Pasar, Ketersediaan Barang, Musim, Volume Pembeian, Harga Jual, Margin, dan Margin Persentase. Fitur produk mencatat nama barang yang dijual, sedangkan tanggal menunjukkan waktu pencatatan harga jual. Harga beli mencerminkan biaya perolehan barang, sementara kompetitor 1 dan kompetitor 2 mencatat harga jual barang yang sama di dua toko pesaing.

Fitur permintaan pasar dikategorikan berdasarkan stok dan volume penjualan: bernilai Tinggi apabila stok barang habis terjual, bernilai Sedang apabila barang terjual lebih dari 50% dari total persediaan, dan Rendah jika penjualan kurang dari 50% penjualan dari stok yang tersedia. Untuk Ketersediaan Barang, nilai Mudah diberikan apabila produk tersedia di ketiga toko yang menjadi objek observasi, sedangkan Langka apabila tidak tersedia di ketiga toko tersebut.

Selanjutnya, fitur Musim digunakan untuk membedakan konteks waktu penjualan, yaitu Hari Biasa untuk hari-hari normal dan Hari Raya untuk momen perayaan keagamaan yang biasanya berdampak pada permintaan pasar. Volume Pembelian merupakan jumlah barang yang dibeli konsumen, Harga Jual merupakan harga akhir yang ditetapkan toko, Margin merupakan selisih antara harga jual dan harga beli, dan margin persentase menunjukkan margin dalam bentuk persentase. Seluruh fitur tersebut membentuk fondasi analisis prediksi harga jual secara komprehensif menggunakan algoritma *machine learning*.

Penelitian ini menggunakan dua *dataset* yang berbeda yaitu *dataset* kecil yang terdiri dari data satu bulan penjualan dan *dataset* besar yang terdiri dari satu tahun penjualan. Pembagian *dataset* ini dilakukan untuk membandingkan algoritma *Gradient Boosting* dan *XGBoost* terhadap *dataset* yang berbeda. Setelah mendapatkan *dataset*, selanjutnya dilakukan labelling terhadap dua belas variabel pada *dataset*. Tidak hanya variabel, melainkan pada nilai kategorikal seperti permintaan pasar, ketersediaan barang, dan musim. Berikut kode yang digunakan untuk labelling *dataset*:

```

1 cat_nominal = [c for c in ['V1', 'V8'] if c in X_train.columns]
2 cat_ordinal = [c for c in ['V6', 'V7'] if c in X_train.columns]
3
4 if cat_ordinal:
5     categories_map = {
6         'V6': ['Rendah', 'Sedang', 'Tinggi'],
7         'V7': ['Mudah', 'Sulit']
8     }
9     ord_list = [categories_map[c] for c in cat_ordinal]
10    ord_enc = OrdinalEncoder(categories=ord_list)
11    X_train[cat_ordinal] = ord_enc.fit_transform(X_train[cat_ordinal])
12    X_test[cat_ordinal] = ord_enc.transform(X_test[cat_ordinal])
13
14 def safe_label_fit(series):
15     le = LabelEncoder()
16     le.fit(series.astype(str))
17     classes = {clstr for i, clstr in enumerate(le.classes_)}
18     return le, classes
19
20 def safe_label_transform(series, classes_dict):
21     return series.astype(str).map(classes_dict).fillna(-1).astype(int)
22
23 for c in cat_nominal:
24     le, classes_dict = safe_label_fit(X_train[c])
25     le.map[c] = classes_dict
26     X_train[c] = safe_label_transform(X_train[c], classes_dict)
27     X_test[c] = safe_label_transform(X_test[c], classes_dict)
28

```

Gambar 4. 1. *Labeling Dataset*

Gambar 4.1 menampilkan kode *python* yang digunakan untuk melakukan proses *labeling* pada atribut kategorikal. Pada tahap awal, *dataset* dipisahkan menjadi dua jenis kategori, yaitu nominal (V1 dan V8) serta *ordinal* (V6 dan V7). Kolom nominal tidak memiliki urutan tertentu, sedangkan kolom *ordinal* memiliki tingkatan yang jelas. Untuk kolom *ordinal*, kategori pada permintaan pasar (V6) didefinisikan berurutan dari Rendah, Sedang, hingga Tinggi, sedangkan pada

ketersediaan barang (V7), kategori Mudah diberi urutan lebih rendah dibanding Langka. Proses konversi dilakukan menggunakan *OrdinalEncoder* sehingga data latih dan data uji dikodekan secara konsisten. Selanjutnya, fungsi *safe_label_fit* dibuat untuk membangun kamus pemetaan setiap kelas ke dalam angka berdasarkan data latih, sehingga dapat menghindari data *leakage*. Fungsi *safe_label_transform* kemudian digunakan untuk mengubah kategori menjadi angka sesuai kamus, dengan perlakuan khusus apabila terdapat label baru di data uji yang tidak ada di data latih, yaitu secara otomatis diberi nilai -1. Setelah itu, label *encoding* diterapkan pada kolom nominal (V1 dan V8) dengan hasil pemetaan disimpan pada variabel *le_maps* agar dapat ditinjau kembali. Dengan proses ini, seluruh kolom kategorikal berhasil dikonversi ke bentuk numerik secara konsisten, sehingga dapat diproses lebih lanjut oleh algoritma *Gradient Boosting* dan *XGBoost* tanpa kehilangan informasi kategorikal yang relevan.

4.1.1 Uji Asumsi Stasioneritas Data

Penelitian ini menggunakan metode *Augmented Dickey-Fuller* untuk mengetahui apakah data deret waktu mengandung unit root atau tidak. Uji ini penting karena sebagian besar model deret waktu seperti *ARIMA* mensyaratkan data yang stasioner. Berikut kode yang digunakan dalam pengujian stasioneritas data pada *dataset* besar dan *dataset* kecil :

Pada gambar 4.2 Merupakan modul uji stasioneritas deret waktu berbasis *Augmented Dickey-Fuller* yang dijalankan pada dataset mentah sebelum tahap pemodelan. Pertama, peneliti mendefinisikan parameter utama berupa nama kolom waktu (tanggal), kolom pengelompokan produk (Produk), serta daftar variabel numerik yang akan diuji. Fungsi *load_raw_df()* memuat data langsung dari *Excel*, mengonversi kolom tanggal ke format waktu, menghapus nilai tanggal yang tidak valid, dan mengurutkan data secara kronologis. Inti pengujian ada pada fungsi *adf_result(series)* yang menjalankan *Augmented Dickey-Fuller* dengan *autolag='AIC'*, kemudian mengembalikan statistik uji, *p-value*, nilai kritis, serta keputusan. Data stasioner apabila $p\text{-value} < 0,05$ (H_0 ditolak), dan non-stasioner apabila $p\text{-value} \geq 0,05$ (gagal tolak H_0).

Untuk mengantisipasi kemungkinan non-stasioner, fungsi *transform_series()* menyiapkan rangkaian transformasi bertahap *level*, *log*, *difference orde 1*, kombinasi *log+diff1*, hingga *diff2* dan *log+diff2*. Setiap kandidat transformasi diuji *Augmented Dickey-Fuller* dan dicatat dalam *steps*; transformasi pertama yang menghasilkan keputusan stasioner dipilih sebagai transformasi terpilih (jika tidak ada yang stasioner, sistem mengambil kandidat terakhir sebagai upaya terbaik). Fungsi *run_checks(df, scope_name)* kemudian menjalankan dua hal untuk setiap variabel numerik yaitu *Augmented Dickey-Fuller* pada data mentah (memberi *raw_pvalue* dan *raw_decision*), dan pencarian transformasi sampai stasioner. Hasil ringkas transformasi terpilih dan keputusan akhir disusun ke dalam *summary* tabel, sedangkan rincian setiap langkah (statistik *Augmented Dickey-Fuller* per transformasi) disusun ke dalam detail tabel. Seluruh proses dieksekusi

dua tingkat yaitu agregat keseluruhan (*scope ALL*) dan (*opsional*) per produk (*RUN_PER_PRODUCT=True*). Berikut hasil pengujian pada *dataset* besar :

Tabel 4. 3. Uji Stasioneritas *Dataset* Besar

Variabel	<i>P-value</i>	Transformasi	Keputusan ADF
Harga Beli	0.000000	level	Stasioner (Tolak H0)
Kompetitor 1	0.000000	level	Stasioner (Tolak H0)
Kompetitor 2	0.000000	level	Stasioner (Tolak H0)
Harga Jual	0.000000	level	Stasioner (Tolak H0)
Margin	0.000015	level	Stasioner (Tolak H0)
Margin Persentase	-0.0000	level	Stasioner (Tolak H0)
Volume Pembelian	-0.0000	level	Stasioner (Tolak H0)

Berdasarkan tabel 4.3 pengujian menggunakan metode *Augmented Dickey-Fuller* pada *dataset* besar, seluruh variabel yang diuji menunjukkan nilai *p-value* lebih kecil dari 0,05. Hal ini berarti hipotesis nol (H_0) yang menyatakan bahwa data mengandung unit root ditolak, sehingga data dinyatakan stasioner. Selain itu, seluruh variabel telah memenuhi kriteria stasioneritas pada level awal (tanpa transformasi), yang terlihat dari kolom transformasi yang menunjukkan nilai "level" untuk setiap variabel. Dengan demikian tidak diperlukan proses transformasi tambahan seperti *logaritma* atau *differencing* untuk mencapai kondisi stasioner. Selanjutnya dibawah ini pengujian stasioneritas pada *dataset* kecil :

Tabel 4. 4. Uji Stasioneritas *Dataset* Kecil

Variabel	<i>P-value</i>	Transformasi	Keputusan ADF
Harga Beli	0.000000	level	Stasioner (Tolak H0)
Kompetitor 1	0.000000	level	Stasioner (Tolak H0)
Kompetitor 2	0.000000	level	Stasioner (Tolak H0)
Harga Jual	0.000000	level	Stasioner (Tolak H0)
Margin	0.000000	level	Stasioner (Tolak H0)
Margin Persentase	0.000000	level	Stasioner (Tolak H0)
Volume Pembelian	0.000000	level	Stasioner (Tolak H0)

Pada tabel 4.4 pengujian menggunakan metode *Augmented Dickey-Fuller* terhadap *dataset* kecil, seluruh variabel numerik menunjukkan nilai *p-value* lebih kecil dari 0,05, sehingga hipotesis nol (H_0) yang menyatakan adanya unit *root*

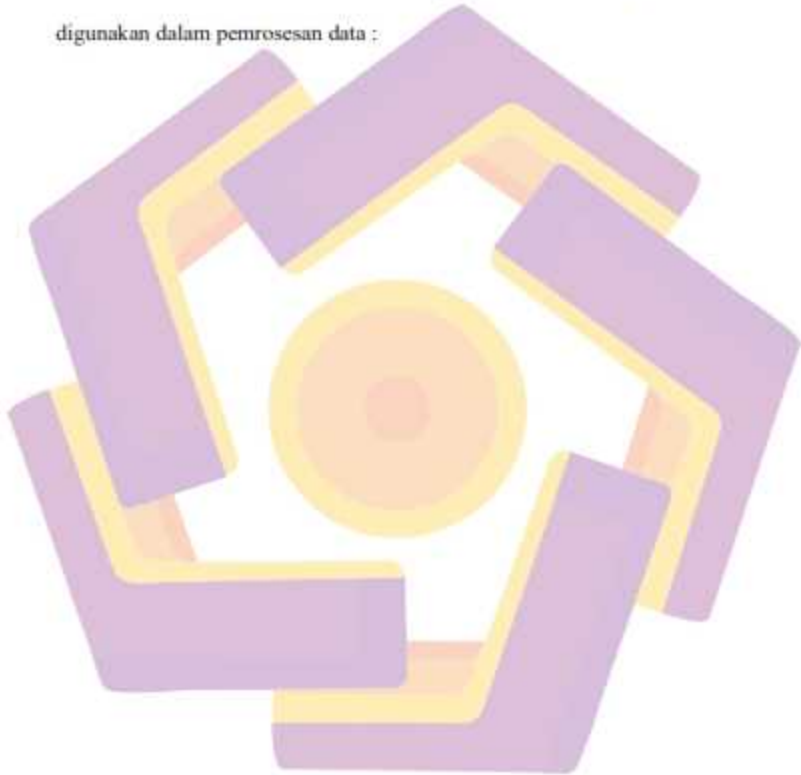
ditolak. Hal ini berarti semua variabel termasuk harga beli, kompetitor 1, kompetitor 2, harga jual dan margin telah memenuhi kriteria stasioneritas pada level awal (tanpa memerlukan transformasi *log* maupun *differencing*).

Pada penelitian ini, meskipun dataset yang digunakan merupakan data deret waktu yang memiliki elemen waktu dan urutan bertahap, hasil uji *Augmented Dickey-Fuller* menunjukkan bahwa seluruh variabel bersifat stasioner. Hal ini dapat terjadi karena keberadaan komponen waktu dalam data tidak selalu identik dengan tren atau pola non-stasioner. Suatu data deret-waktu dinyatakan non-stasioner hanya jika nilai rata-rata, variansi, atau kovariansnya berubah secara signifikan terhadap waktu. Dalam *dataset* ini, meskipun pencatatan data dilakukan secara periodik dan berurutan, fluktuasi nilai pada setiap variabel relatif stabil dan tidak menunjukkan tren naik atau turun yang kuat. Dengan kata lain, meskipun data dicatat berdasarkan waktu, perubahan nilainya bersifat acak dan berfluktuasi di sekitar rata-rata yang konstan, sehingga memenuhi kriteria stasioneritas. Stabilitas pola ini bisa disebabkan oleh faktor operasional yang relatif konsisten dari waktu ke waktu, misalnya harga dan margin produk yang tidak mengalami perubahan struktural besar. Oleh karena itu, keberadaan komponen waktu tidak secara otomatis menyebabkan data menjadi non-stasioner, sehingga hasil uji *Augmented Dickey-Fuller* menunjukkan data telah stasioner pada level awal tanpa transformasi tambahan.

4.2. Pra-pemrosesan Data

Pada tahap pra-pemrosesan data, dilakukan beberapa langkah penting untuk memastikan bahwa data siap untuk digunakan dalam melatih model. Pra-

pemrosesan data mencakup proses transformasi atau pengkodean data mentah ke dalam bentuk yang dapat dengan mudah dibaca dan dipahami oleh program. Pra-pemrosesan data dapat meningkatkan kualitas, validitas, dan keseragaman data yang digunakan dalam model *machine learning* [24]. Berikut kode yang digunakan dalam pemrosesan data :



Gambar 4.3 menampilkan tahapan penting yang dilakukan untuk menyiapkan *dataset* sebelum digunakan dalam proses pelatihan model. Pada bagian awal, data dimuat dari *file excel*, kolom yang tidak relevan seperti "No" dihapus, dan seluruh kolom dinormalisasi dengan nama standar V1 hingga V12 agar lebih konsisten. Kolom V1, V6, V7, dan V8 dinormalisasi teksnya, sedangkan kolom numerik seperti V2, V3, V4, V5, V9, V10, V11, dan V12 dipastikan bertipe numerik sehingga nilai yang tidak valid bisa diubah menjadi *NaN*. Setelah itu ditentukan variabel fitur (*features*) dan target (V10) yang digunakan dalam pemodelan, lalu dilakukan *train-test split* dengan proporsi 80% data latih dan 20% data uji untuk mencegah data *leakage*.

Selanjutnya dilakukan penanganan *missing value* menggunakan teknik imputasi. Untuk data numerik digunakan *median* agar lebih *robust* terhadap *outlier*, sedangkan untuk data kategorikal digunakan *modus* (frekuensi terbanyak). Nilai-nilai *outlier* kemudian ditangani dengan metode *Interquartile Range (IQR) clipping*, sehingga nilai ekstrem yang terlalu besar atau kecil dipangkas ke atas bawah dan batas atas yang lebih wajar. Tahap berikutnya adalah *encoding* data kategorikal. Untuk fitur *ordinal* (V6 dan V7), digunakan *OrdinalEncoder* dengan urutan kategori yang ditentukan, misalnya Rendah < Sedang < Tinggi dan Mudah < Langka. Sementara itu, untuk fitur nominal (V1 dan V8), digunakan *Label Encoding* dengan fungsi khusus yang aman terhadap label baru pada data uji. Jika ada kategori yang tidak dikenal, nilainya otomatis diberikan -1 agar *pipeline* tetap *robust*.

4.2.1. Uji Korelasi Antar Fitur

Untuk memahami hubungan antar variabel dalam *dataset*, dilakukan analisis korelasi menggunakan *heatmap*. Analisis korelasi ini bertujuan untuk melihat sejauh mana keterkaitan antara variabel *independen* (fitur) dengan variabel target, yaitu harga jual (V10). Dengan adanya *heatmap*, pola hubungan antar variabel dapat divisualisasikan secara lebih jelas melalui gradasi warna, dimana nilai mendekati 1 menunjukkan korelasi positif yang kuat, sedangkan nilai mendekati -1 menunjukkan korelasi negatif yang kuat. Berikut kode yang digunakan untuk uji korelasi :



```

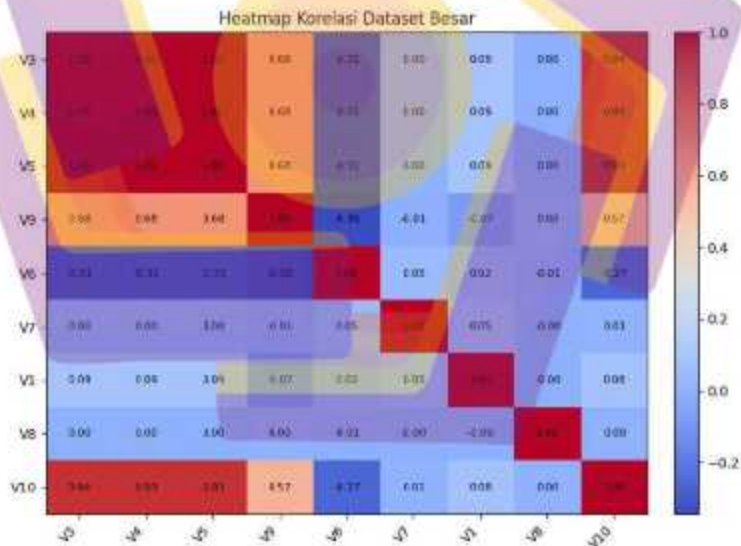
1 cols_for_corr = num_cols + cat_encoded + cat_encoded
2 corr_df = pd.DataFrame({'feature': [target], 'corr':corr(numeric_only=True)})
3
4 plt.figure(figsize=(8,6))
5 im = plt.imshow(corr_df, interpolation='nearest', aspect='equal', cmap='magma')
6 plt.title('Heatmap Korelasi Antar Fitur dan Target')
7 plt.colorbar(orientation='horizontal', label='Correlation')
8 plt.grid(True, linestyle='dotted', color='gray')
9 plt.xticks(rotation=45, labels=[corr_df.columns], labelsize=10, fontweight='bold')
10 plt.yticks(rotation=45, labels=[corr_df.index], labelsize=10, fontweight='bold')
11 fig = plt.gcf()
12 fig.subplots_adjust(bottom=0.1)
13 ax = fig.add_subplot(1,1,fig.gca().axisonline, width=10, height=10, frameon=False)
14 plt.tight_layout()
15 plt.show()

```

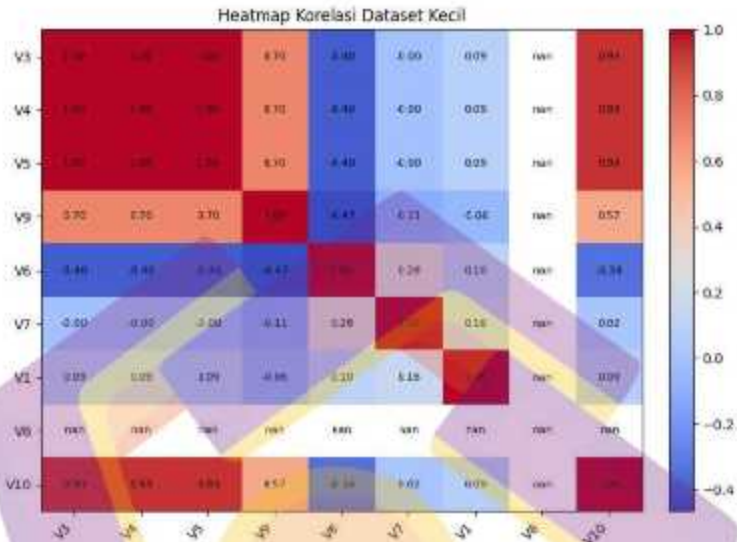
Gambar 4. 4. Kode Uji Korelasi

Pada gambar 4.4 digunakan untuk memvisualisasikan korelasi antar fitur menggunakan *library matplotlib*. Pertama, variabel *cols_for_corr* menyimpan daftar fitur numerik dan kategorikal (yang sudah diubah menjadi numerik melalui *encoding*) beserta target V10 (harga jual). Kemudian, fungsi *.corr(numeric_only=True)* menghitung matriks korelasi antar semua fitur numerik.

Nilai korelasi berkisar antara -1 sampai 1, dimana nilai positif menunjukkan hubungan searah, sementara nilai negatif menunjukkan hubungan berlawanan arah. Selanjutnya, fungsi `plt.imshow()` menampilkan matriks korelasi dalam bentuk *heatmap* dengan skema warna *coolwarm*. warna merah menunjukkan korelasi positif yang kuat, sedangkan warna biru menandakan korelasi negatif. Untuk memperjelas, setiap sel matriks diberi label angka dengan dua desimal yang menunjukkan nilai korelasi antar fitur. Melalui visualisasi ini, dapat diketahui fitur-fitur mana yang memiliki hubungan kuat dengan variabel target V10(harga jual), sehingga bisa menjadi indikasi awal fitur penting dalam pembentukan model prediksi. Berikut *heatmap* dari *dataset* besar dan *dataset* kecil :



Gambar 4. 5. *Heatmap Dataset Besar*



Gambar 4. 6. *Heatmap Dataset Kecil*

Gambar 4.5 *Heatmap Dataset Besar* dan Gambar 4.6 *Heatmap Dataset Kecil* menampilkan hasil uji korelasi antar fitur melalui tahap pra-pemrosesan. Pada *dataset besar* terlihat bahwa fitur Harga Beli(V3), Harga Kompetitor 1(V4) dan Harga Kompetitor 2(V5) memiliki korelasi yang sangat kuat dan positif terhadap Harga Jual(V10) dengan nilai diatas 0,90. Hal ini menunjukkan bahwa harga jual sangat dipengaruhi oleh harga beli serta harga dari kompetitor. Selain itu, Volume Pembelian (V9) juga menunjukkan korelasi sedang 0,57 terhadap harga jual, sementara variabel kategorikal seperti Permintaan Pasar(V6) dan Ketersediaan Barang(V7) memiliki korelasi lemah atau negatif, sehingga kontribusinya terhadap harga jual relatif kecil.

Pada *dataset* kecil pola yang ditunjukkan relatif sama, di mana fitur Harga Beli (V3), Harga Kompetitor 1(V4), dan Harga Kompetitor 2 (V5) tetap mendominasi dengan korelasi tinggi terhadap harga jual (>0,90). Namun, kekuatan korelasi dengan Volume Pembelian (V9) sedikit menurun (0,57) dibanding *dataset* besar, dan hubungan dengan Permintaan Pasar(V6) menjadi lebih negatif (-0,34). Sementara itu, beberapa variabel seperti musim (V8) tidak muncul dalam *heatmap dataset* kecil karena variasinya terbatas. Perbedaan pola ini mengindikasikan bahwa skala data memengaruhi kekuatan hubungan antar fitur, namun secara konsisten harga beli dan harga kompetitor tetap menjadi faktor dominan dalam menentukan harga jual.

4.2.2. Alur Proses Pra-Pemrosesan Hingga Penentuan Harga Jual

Proses penentuan harga jual pada penelitian ini diawali dengan serangkaian tahapan pra-pemrosesan data yang bertujuan untuk memastikan data dalam kondisi bersih, dan siap digunakan oleh algoritma *machine learning*. Tahapan ini melibatkan transformasi data mentah menjadi bentuk yang dapat dipahami oleh model *Gradient Boosting* dan *XGBoost*.

Langkah pertama dilakukan dengan menghapus atribut yang tidak berkontribusi terhadap proses prediksi, seperti kolom identitas (No).Selanjutnya, atribut produk dikonversi menggunakan teknik label *encoding* menjadi fitur numerik pada produk (V1) dan musim (V8) untuk mewakili kategori produk dalam bentuk angka. Proses serupa juga diterapkan pada atribut kategorikal lainnya seperti permintaan pasar (V6) dan ketersediaan barang(V7) dengan metode *ordinal*

encoding, sehingga seluruh fitur dapat diolah oleh algoritma secara konsisten dalam bentuk numerik.

Dua fitur kategorikal lain yang bersifat *ordinal* yaitu Permintaan Pasar dan Ketersediaan Barang juga diubah ke bentuk numerik berdasarkan aturan klasifikasi tertentu. Nilai Permintaan Pasar diberi label sebagai Tinggi apabila stok barang terjual habis, Sedang apabila lebih dari 50% stok terjual, dan Rendah apabila kurang dari 50% stok terjual. Ketersediaan Barang bernilai Mudah apabila barang tersedia di ketiga toko, dan Langka apabila tidak tersedia sama sekali. Adapun fitur Musim diberi nilai 0 untuk Hari Biasa dan 1 untuk Hari Raya.

Selanjutnya, dilakukan penanganan *outlier* pada fitur numerik seperti Harga Beli, Harga Kompetitor, dan Harga Jual dengan metode *clipping* berbasis *Interquartile Range (IQR)*, dimana nilai dibawah batas bawah ($Q1 - 1,5 \times IQR$) dan diatas batas atas ($Q3 + 1,5 \times IQR$) dianggap sebagai *outlier*. Pendekatan ini digunakan karena lebih adaptif terhadap distribusi data dibandingkan penggunaan ambang tetap (*fixed percentile*). Hal ini dilakukan untuk menghindari gangguan distribusi akibat nilai ekstrem yang terlalu besar atau kecil. Setelah data siap, proses pembagian data dilakukan menjadi data latih dan data uji dengan rasio 80:20. Dua skenario digunakan dalam penelitian ini, yaitu *dataset* besar sebanyak 5.300 data penjualan mingguan selama satu tahun dan *dataset* kecil sebanyak 500 data dalam satu bulan.

Model *Gradient Boosting* kemudian dilatih menggunakan parameter hasil dari *GridSearchCV*, yaitu $n_estimators = 200$, $learning_rate = 0,1$, dan $max_depth = 7$. Model ini bekerja dengan membangun pohon keputusan secara bertahap untuk

meminimalkan galat prediksi. Sementara itu, *XGBoost* sebagai pengembangan dari *Gradient Boosting*, dilatih dengan parameter terbaik yaitu $n_estimators = 200$, $learning_rate = 0,1$, $max_depth = 5$, $subsample = 1,0$, dan $colsample_bytree = 0,8$. Keunggulan *XGBoost* terletak pada efisiensi komputasi dan kemampuannya melakukan regularisasi untuk mencegah *overfitting*.

Sebagai ilustrasi, berikut adalah contoh kasus konkret yang diambil dari data aktual. Sebuah produk "Beras Premium" dijual dengan harga beli Rp11.500. Harga dari kompetitor 1 adalah Rp12.500 dan Kompetitor 2 adalah Rp12.300. Pada periode tersebut, stok barang habis terjual sehingga dikategorikan sebagai Permintaan Tinggi, barang tersedia di semua toko dikategorikan sebagai Mudah, dan hari penjualan termasuk ke dalam Hari Biasa. Setelah melalui tahap encoding dan normalisasi, data ini diproses oleh kedua algoritma.

Hasil prediksi yang diberikan oleh *Gradient Boosting* untuk harga jual optimal adalah Rp12.200, sedangkan *XGBoost* menghasilkan prediksi harga Rp12.100. Hasil ini menunjukkan bahwa kedua algoritma memberikan estimasi yang konsisten dan sangat dekat, namun dengan pendekatan struktur model yang berbeda. *Gradient Boosting* cenderung menyebarkan kontribusi prediksi ke beberapa fitur, sementara *XGBoost* secara agresif memfokuskan bobot prediksi pada fitur utama seperti Harga Beli dan Harga Kompetitor. Dengan proses ini, model dapat merekomendasikan harga jual optimal berdasarkan faktor historis dan kompetitif, memberikan solusi berbasis data yang lebih objektif dan akurat dalam mendukung pengambilan keputusan harga di toko pasar tradisional.

4.3. Pembagian Dataset

Penelitian ini *dataset* dibagi menjadi dua *subset*, yaitu data latih atau *training* data sebesar 80% dan data uji atau *testing* data sebesar 20%. Pembagian ini dilakukan untuk memastikan bahwa model dapat belajar dari Sebagian besar data yang tersedia, sekaligus memungkinkan evaluasi kinerja model secara objektif. Pendekatan ini bertujuan untuk mengukur kemampuan generalisasi model *Gradient Boosting* dan *XGBoost* dalam memprediksi harga jual pada data baru. Berikut kode yang digunakan dalam pembagian data :



```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

Gambar 4. 7. Kode Pembagian Dataset

Pada gambar 4.7 *dataset* dibagi menggunakan fungsi *train_test_split* dari Pustaka *scikit-learn*, yang memisahkan data menjadi dua bagian, yaitu data latih atau *training set* dan data uji atau *testing set*. Sebanyak 80% data digunakan untuk menguji akurasi model (*x_test* dan *y_test*). Parameter *test_size=0,2* menunjukkan proporsi data uji sebesar 20%, sementara *random_state=42* digunakan untuk menjamin reproduktibilitas hasil pembagian data, sehingga penelitian yang dilakukan dapat direplikasi dengan hasil yang konsisten.

4.4. Pengembangan Model

Pengembangan model pada penelitian ini diawali dengan penggunaan *GridSearchCV* untuk mendapatkan kombinasi parameter terbaik. *GridSearchCV* merupakan metode yang digunakan untuk pencarian sistematis terhadap kombinasi

hyperparameter yang dilakukan dengan Teknik *cross-validation*, sehingga meningkatkan kinerja dan akurasi dari model *Gradient Boosting* yang digunakan dalam penelitian ini. Dibawah ini merupakan penggunaan *GridSearchCV* pada penelitian ini :

```

1 param_gbr = {
2     'n_estimators': [100, 200],
3     'learning_rate': [0.05, 0.1],
4     'max_depth': [3, 5]
5 }
6 gbr_grid = GridSearchCV(
7     GradientBoostingRegressor(random_state=42),
8     param_grid=param_gbr,
9     cv=3,
10    scoring='r2',
11    n_jobs=-1
12 )
13 gbr_grid.fit(X_train, y_train)
14 gbr_best = gbr_grid.best_estimator_
15 print("Model Gradient Boosting terbaik (berdasarkan R2 CV):", gbr_best)
16
17 param_xgb = {
18     'n_estimators': [100, 200],
19     'learning_rate': [0.05, 0.1],
20     'max_depth': [3, 5, 7],
21     'subsample': [0.8, 1.0],
22     'colsample_bytree': [0.8, 1.0]
23 }
24 xgb_grid = GridSearchCV(
25     XGBRegressor(objective='reg:squarederror', random_state=42, verbosity=0),
26     param_grid=param_xgb,
27     cv=3,
28     scoring='r2',
29     n_jobs=-1
30 )
31 xgb_grid.fit(X_train, y_train)
32 xgb_best = xgb_grid.best_estimator_
33 print("Model XGBoost terbaik:", xgb_best)
34

```

Gambar 4. 8. Kode *GridSearchCV* Parameter

Pada gambar 4.8 untuk mencari kombinasi parameter terbaik digunakan *GridSearchCV* dengan parameter pada tabel berikut :

Tabel 4. 5. Parameter Kombinasi *GridSearchCV*

No	Model	<i>Hyperparameter Combination</i>
1	<i>Gradient Boosting</i>	<i>n_estimators</i> = [100, 200] <i>learning_rate</i> = [0,05, 0,1] <i>max_depth</i> = [3, 5]
2	<i>XGBoost</i>	<i>n_estimators</i> = [100, 200] <i>learning_rate</i> = [0,05, 0,1] <i>max_depth</i> = [3, 5, 7] <i>subsample</i> = [0,8, 1,0] <i>colsample_bytree</i> = [0,8, 1,0]

Pada tabel 4.5 memuat kombinasi *hyperparameter* untuk model *Gradient Boosting* dan *XGBoost*. Model *Gradient Boosting* menggunakan parameter *n_estimators* dengan nilai [100, 200], *learning_rate* dengan nilai [0,05, 0,1], serta *max_depth* dengan nilai [3, 5]. Parameter *n_estimators* berfungsi mengatur jumlah iterasi *boosting* yang digunakan, dimana nilai yang lebih tinggi memungkinkan model membangun lebih banyak pohon untuk meningkatkan akurasi, tetapi juga berisiko menimbulkan *overfitting*. Parameter *learning_rate* mengontrol seberapa besar kontribusi tiap pohon baru dalam memperbaiki kesalahan model sebelumnya, sehingga nilai yang lebih kecil membuat proses pembelajaran lebih hati-hati namun memerlukan lebih banyak pohon, sedangkan nilai yang lebih besar mempercepat pembelajaran tetapi rentan terhadap *overfitting*. Sementara itu, *max_depth* berfungsi mengendalikan kompleksitas tiap pohon, dimana nilai kedalaman kecil menghasilkan model yang lebih sederhana, sedangkan nilai yang lebih besar memungkinkan model menangkap interaksi variabel yang lebih kompleks.

Sedangkan pada model *XGBoost*, ruang pencarian parameter lebih luas dengan *n_estimators* [100, 200], *learning_rate* [0,05, 0,1], *max_depth* [3, 5, 7], *subsample* [0,8, 1,0], serta *colsample_bytree* [0,8, 1,0]. Parameter *subsample* mengatur proporsi data latih yang digunakan pada setiap iterasi *boosting*, sehingga

dapat membantu mengurangi *overfitting* dengan cara memperkenalkan keragaman data pada setiap pohon. Sementara itu, *colsample_bytree* menentukan proporsi fitur yang digunakan ketika membangun setiap pohon, yang juga berfungsi sebagai teknik regularisasi agar model tidak hanya bergantung pada *subset* fitur tertentu. Kombinasi parameter ini membuat *XGBoost* lebih fleksibel dan mampu menghasilkan model yang lebih stabil terhadap *overfitting*, terutama pada dataset yang kompleks.

4.5. Pengujian Model

Setelah model *Gradient Boosting* dan *XGBoost* dikembangkan dengan menggunakan parameter terbaik hasil optimasi menggunakan *GridSearchCV*, tahap selanjutnya yaitu melakukan pengujian model untuk menilai kinerja dan kemampuan generalisasi terhadap data yang belum pernah diuji sebelumnya. Berikut kode yang digunakan dalam pengujian model *Gradient Boosting* dan *XGBoost* pada *dataset* besar maupun *dataset* kecil :

```

1 def mape(y_true, y_pred, eps=1e-8):
2     return np.mean(np.abs(y_true - y_pred) / (np.abs(y_true) + eps)) * 100
3
4 y_pred_gbr = gbr_best.predict(X_test)
5 y_pred_xgb = xgb_best.predict(X_test)
6
7 print("\n\n Evaluasi MAPE & R2 :")
8 print("Gradient Boosting:")
9 print(f" R2 : {r2_score(y_test, y_pred_gbr):.4f}")
10 print(f" MAPE : {mape(y_test, y_pred_gbr):.1f}%")
11
12 print("\n\n XGBoost:")
13 print(f" R2 : {r2_score(y_test, y_pred_xgb):.4f}")
14 print(f" MAPE : {mape(y_test, y_pred_xgb):.1f}%")
15

```

Gambar 4. 9. Kode Pengujian Model *Gradient Boosting* dan *XGBoost*

Pada gambar 4.9 menampilkan kode *python* yang digunakan untuk mengevaluasi kinerja algoritma *Gradient Boosting* dan *XGBoost* pada data uji. Pertama, dibuat fungsi *mape()* yang menghitung *Mean Absolute Percentage Error (MAPE)* untuk mengukur rata-rata kesalahan prediksi dalam bentuk persentase. Fungsi ini berguna karena memberikan gambaran intuitif mengenai tingkat deviasi prediksi dari nilai aktual. Selanjutnya, model terbaik hasil *GridSearchCV*, yaitu *gbr_test* untuk *Gradient Boosting* dan *xgb_best* untuk *XGBoost*, digunakan untuk melakukan prediksi terhadap data uji yang telah dipisahkan sebelumnya (*X_test*). Hasil prediksi tersebut disimpan pada variabel *y_pred_gbr* dan *y_pred_xgb*. Dari penggunaan kode diatas terhadap model *Gradient Boosting* dan *XGBoost* pada *dataset* besar dan *dataset* kecil maka hasilnya sebagai berikut :

Tabel 4. 6. Hasil Evaluasi Algoritma pada *dataset* besar

Model	R^2	MAPE
<i>Gradient Boosting</i>	0,9999	1,82%
<i>XGBoost</i>	0,9999	1,85%

Tabel 4. 7. Hasil Evaluasi Algoritma pada *dataset* kecil

Model	R^2	MAPE
<i>Gradient Boosting</i>	0,9999	1,07%
<i>XGBoost</i>	0,9999	1,05%

Pada tabel 4.6 dan 4.7 memperlihatkan bahwa baik algoritma *Gradient Boosting* maupun *XGBoost* memiliki kinerja yang sangat baik pada *dataset* besar maupun *dataset* kecil. Pada *dataset* besar, kedua algoritma menghasilkan nilai R^2 sebesar 0,9999, yang menandakan bahwa hampir seluruh variasi pada variabel target dapat dijelaskan oleh model. Nilai *MAPE* yang diperoleh juga relatif rendah,

yaitu 1,82% untuk *Gradient Boosting* dan *XGBoost* 1,85%, yang menunjukkan tingkat kesalahan prediksi rata-rata kurang dari 2%. Hasil ini menegaskan bahwa kedua model mampu melakukan prediksi harga dengan akurasi yang sangat tinggi.

Sementara itu, pada dataset kecil juga diperoleh hasil evaluasi yang konsisten, bahkan dengan tingkat kesalahan yang lebih rendah. Nilai R^2 tetap berada pada 0,9999 untuk kedua algoritma, sementara nilai *MAPE* mencapai 1,07% pada *Gradient Boosting* dan 1,05% pada *XGBoost*. Hal ini mengindikasikan bahwa meskipun ukuran *dataset* lebih kecil, kedua algoritma tetap mampu melakukan generalisasi dengan sangat baik. Dari hasil ini dapat disimpulkan bahwa baik *Gradient Boosting* dan *XGBoost* sama-sama efektif digunakan untuk estimasi harga, dengan keunggulan *XGBoost* yang sedikit lebih baik pada *dataset* kecil dan *Gradient Boosting* yang sedikit lebih unggul pada *dataset* besar.

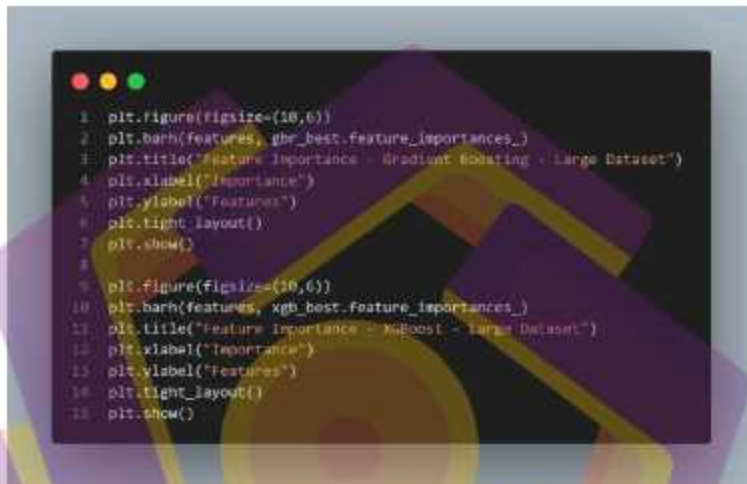
4.6. Feature Importance

Feature Importance adalah teknik dalam pembelajaran mesin atau *machine learning* yang digunakan untuk mengukur sejauh mana setiap fitur atau *variable input* memberikan kontribusi terhadap prediksi yang dihasilkan oleh model algoritma. Pada penelitian ini *feature importance* diaplikasikan pada kedua algoritma yaitu *Gradient Boosting* dan *XGBoost* pada *dataset* besar dan *dataset* kecil.

4.6.1. Feature Importance pada dataset besar

Analisis *feature importance* pada *dataset* besar dilakukan untuk mengevaluasi kontribusi masing-masing fitur terhadap prediksi model pada *dataset* besar. Penggunaan teknik ini untuk membantu proses interpretasi dan pemilihan

fitur yang paling relevan, serta meningkatkan transparansi dalam pengambilan Keputusan berbasis data. Berikut kode yang digunakan untuk memvisualisasikan *feature importance* algoritma *Gradient Boosting* dan *XGBoost* :



```

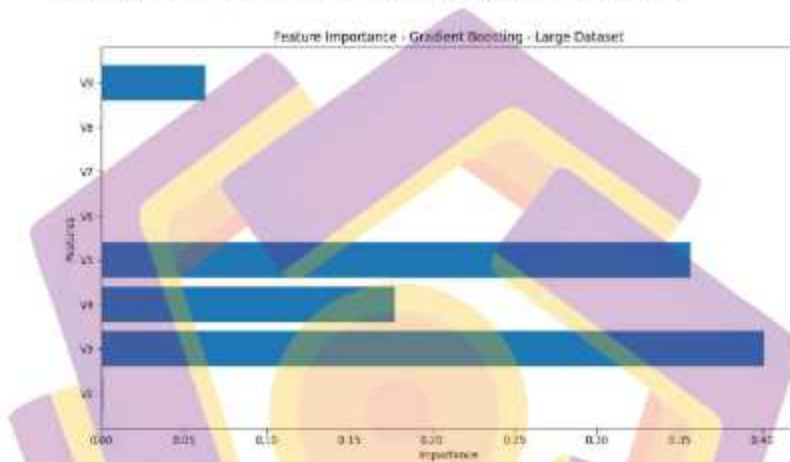
1 plt.figure(figsize=(10,6))
2 plt.barh(features, gbr_best.feature_importances_)
3 plt.title("Feature Importance - Gradient Boosting - Large Dataset")
4 plt.xlabel("Importance")
5 plt.ylabel("Features")
6 plt.tight_layout()
7 plt.show()
8
9 plt.figure(figsize=(10,6))
10 plt.barh(features, xgb_best.feature_importances_)
11 plt.title("Feature Importance - XGBoost - Large Dataset")
12 plt.xlabel("Importance")
13 plt.ylabel("Features")
14 plt.tight_layout()
15 plt.show()

```

Gambar 4. 10. Kode Visualisasi *Feature Importance* Dataset Besar

Pada gambar 4.10 kode tersebut digunakan untuk memvisualisasikan *feature importance* algoritma *Gradient Boosting* dan *XGBoost* pada *dataset* besar. Proses visualisasi ini dilakukan menjadi dua bagian. Bagian pertama menghasilkan grafik *feature importance* dari model *Gradient Boosting*. Dimana `plt.barh(features,gbr_best.feature_importance_)` memetakan daftar fitur terhadap nilai kepentingannya berdasarkan hasil pelatihan model terbaik (`gbr_best`) yang diperoleh dari *GridSearchCV*. Label sumbu *x* dan *y* masing-masing menunjukkan nilai "*importance*" dan nama-nama fitur, sementara fungsi `tight_layout()` digunakan untuk merapikan tampilan grafik agar tidak terpotong. Bagian kedua dari kode

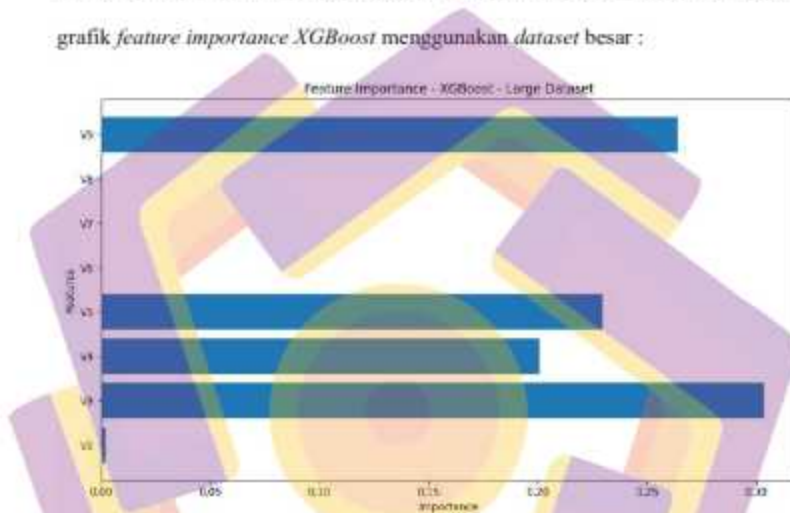
tersebut menjalankan visualisasi serupa untuk model *XGBoost*. Dengan menggunakan objek `xgb_best.feature_importances_`, grafik menampilkan kontribusi fitur pada model *XGBoost* terbaik yang telah melalui proses tuning parameter. Berikut hasil visualisasi *feature importance dataset* besar :



Gambar 4. 11. *Feature Importance Gradient Boosting Dataset Besar*

Berdasarkan grafik pada gambar 4.11 menunjukkan urutan kontribusi fitur terhadap kinerja prediksi model *Gradient Boosting*. Dari hasil visualisasi terlihat bahwa fitur harga beli (V3) dan harga kompetitor 2 (V5) merupakan variabel dengan tingkat kepentingan paling tinggi, diikuti oleh harga kompetitor 1 (V4). Ketiga fitur ini memberikan pengaruh dominan terhadap penentuan nilai target karena secara langsung berkaitan dengan aspek harga dalam data penjualan. Sementara itu, fitur volume pembelian (V9) memberikan kontribusi yang relatif kecil, dan fitur kategorikal seperti permintaan pasar (V6), ketersediaan barang (V7), musim (V8), serta produk (V1) memiliki pengaruh yang sangat minim terhadap

prediksi. Temuan ini konsisten dengan teori bahwa variabel harga lebih signifikan dalam membentuk model peramalan harga dibanding faktor eksternal non-harga, sehingga dapat disimpulkan bahwa model *Gradient Boosting* lebih banyak mengandalkan informasi harga dalam menghasilkan prediksi yang akurat. Berikut grafik *feature importance XGBoost* menggunakan *dataset* besar :



Gambar 4. 12. *Feature Importance XGBoost Dataset Besar*

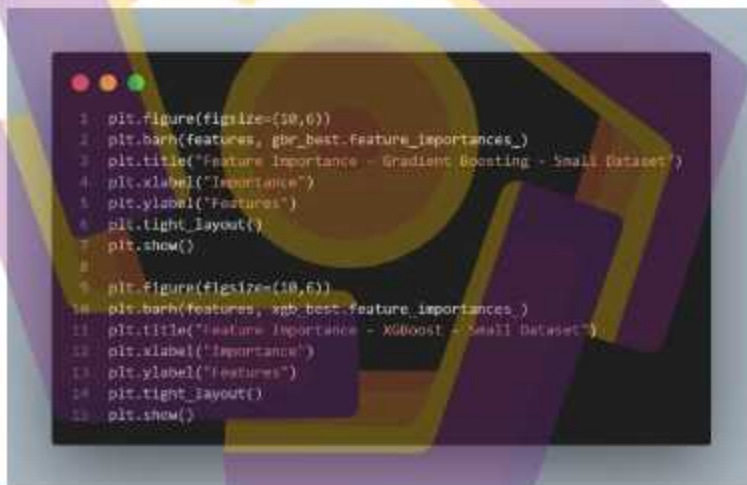
Pada gambar 4.12 menunjukkan bahwa distribusi kontribusi masing-masing fitur terhadap kinerja model *XGBoost*. Hasil evaluasi menunjukkan bahwa fitur

Harga Beli (V3) dan volume pembelian (V9) menjadi dua variabel dengan pengaruh paling dominan dalam memprediksi target. Selanjutnya, fitur harga kompetitor 2(V5) berada pada tingkat menengah. Disisi lain, fitur kategorikal seperti produk (V1), ketersediaan barang (V7), dan musim (V8) memberikan kontribusi yang sangat kecil terhadap kinerja model. Temuan ini mengindikasikan bahwa *XGBoost* lebih menekankan pada variabel harga serta volume pembelian

dalam menghasilkan prediksi yang akurat, sehingga faktor-faktor eksternal non-harga memiliki peranan yang relatif minor dalam struktur model.

4.6.2. *Feature Importance* pada *dataset* kecil

Analisis *feature importance* juga dilakukan pada *dataset* kecil untuk melihat bagaimana distribusi kontribusi fitur berubah ketika jumlah data lebih terbatas. Visualisasi dilakukan menggunakan model terbaik dari algoritma *Gradient Boosting* dan *XGBoost* setelah proses *GridSearchCV*. Berikut kode *python* yang digunakan untuk memvisualisasikan *feature importance* pada algoritma *Gradient Boosting* dan *XGBoost* menggunakan *dataset* kecil :



```

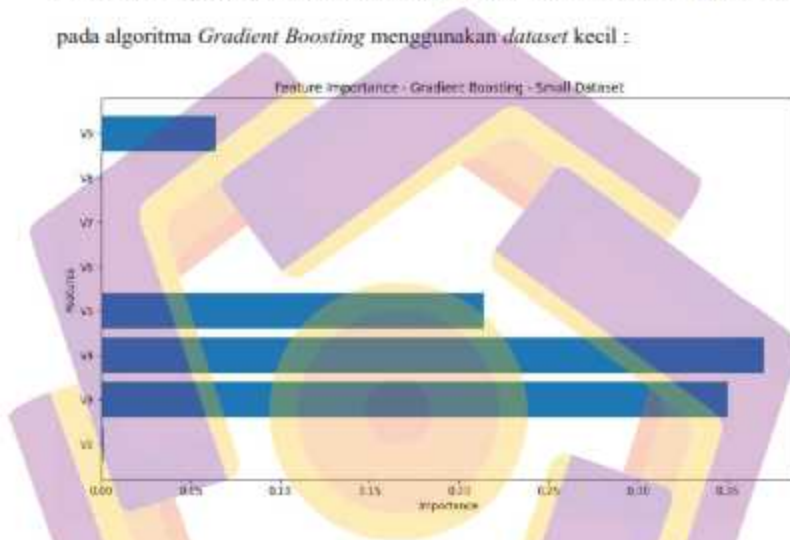
1 plt.figure(figsize=(10,6))
2 plt.barh(features, gbr_best.feature_importances_)
3 plt.title("Feature Importance - Gradient Boosting - Small Dataset")
4 plt.xlabel("Importance")
5 plt.ylabel("Features")
6 plt.tight_layout()
7 plt.show()
8
9 plt.figure(figsize=(10,6))
10 plt.barh(features, xgb_best.feature_importances_)
11 plt.title("Feature Importance - XGBoost - Small Dataset")
12 plt.xlabel("Importance")
13 plt.ylabel("Features")
14 plt.tight_layout()
15 plt.show()

```

Gambar 4. 13. Kode Visualisasi *Feature Importance* *Dataset* Kecil

Pada gambar 4.13 *skrip python* menggunakan dua blok utama. Blok pertama memvisualisasikan nilai *feature importance* dari model *Gradient Boosting*, sedangkan blok kedua untuk model *XGBoost*. Fungsi *plt.barh()* digunakan untuk

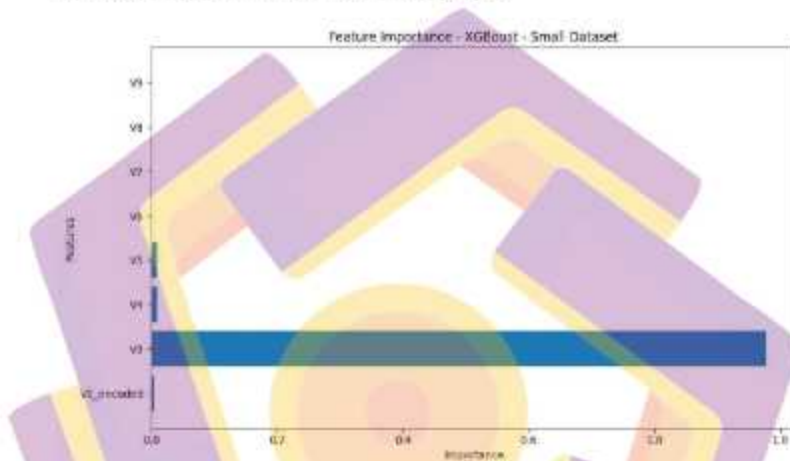
membuat grafik batang *horizontal* dari daftar fitur (*features*) terhadap skor kepentingan (*features_importance*) yang dihasilkan model terbaik (*gbr_best* dan *xgb_best*). Ukuran grafik ditentukan dengan *figsize=(10,6)* untuk memastikan visual dapat dibaca dengan jelas. Berikut gambar hasil visualisasi dari gambar 4.14 pada algoritma *Gradient Boosting* menggunakan *dataset* kecil :



Gambar 4.14: *Feature Importance Gradient Boosting Dataset* Kecil

Pada gambar 4.14 menunjukkan bahwa model *Gradient Boosting* terlihat bahwa fitur harga kompetitor 1(V4) dan harga beli(V3) menjadi faktor paling dominan dengan nilai kepentingan yang tinggi, diikuti oleh harga kompetitor 2 (V5). Ketiga fitur ini berperan besar dalam membentuk kinerja prediksi karena secara langsung berkaitan dengan aspek penentuan harga. Selanjutnya, fitur volume pembelian (V9) menunjukkan kontribusi yang relatif rendah, sementara fitur kategorikal seperti produk(V1), permintaan pasar(V6), ketersediaan barang(V7), dan musim(V8) hampir tidak memberikan pengaruh yang signifikan terhadap

model. Temuan ini memperkuat hasil pada dataset besar, bahwa variabel harga tetap menjadi penentu utama dalam proses estimasi, sedangkan faktor non-harga cenderung memiliki peranan yang sangat terbatas. Berikut visualisasi *feature importance XGBoost* menggunakan *dataset* kecil :




Gambar 4. 15. *Feature Importance XGBoost Dataset* Kecil

Pada gambar 4.15 menunjukkan bahwa model *XGBoost* terlihat bahwa fitur harga beli(V3) menjadi variabel yang paling dominan dengan tingkat kepentingan yang jauh lebih tinggi dibandingkan fitur lainnya. Selanjutnya, harga kompetitor 1 (V4) dan volume pembelian (V9) juga memberikan kontribusi yang cukup berarti meskipun tidak sebesar harga beli (V3). Sementara itu, fitur harga kompetitor 2(V5) hanya memiliki pengaruh yang sangat kecil, sedangkan fitur kategorikal seperti produk (V1), permintaan pasar(V6), ketersediaan barang(V7), dan musim(V8) hampir tidak memberikan kontribusi terhadap model. Hasil ini menegaskan bahwa pada *dataset* kecil, *XGBoost* sangat mengandalkan variabel harga terutama harga

beli dalam menghasilkan prediksi, sedangkan faktor eksternal non-harga hanya berperan minor.

4.7. SHAP Analysis

SHAP Analysis digunakan untuk meningkatkan interpretabilitas dari model pembelajaran mesin yang telah dibangun, khususnya dalam konteks penentuan harga jual, penelitian ini menggunakan pendekatan interpretasi model berbasis *SHAP (SHapley Additive Explanatio)*. *SHAP* adalah metode yang secara sistematis konsisten dan teoritis kuat untuk menjelaskan kontribusi setiap fitur terhadap prediksi individual yang dihasilkan oleh model. Metode *SHAP* didasarkan pada konsep nilai *Shapley* dalam teori permainan kooperatif, yang bertujuan menghitung kontribusi marginal dari setiap fitur dengan mempertimbangkan semua kemungkinan kombinasi fitur lainnya. Nilai *SHAP* memberikan gambaran kuantitatif mengenai sejauh mana masing-masing fitur memengaruhi hasil prediksi akhir. Jika nilai *SHAP* suatu fitur bernilai positif terhadap prediksi, maka fitur tersebut mendorong prediksi menjunilai yang lebih tinggi. Berikut kode *python* yang digunakan untuk menampilkan hasil visualisasi *SHAP* Analysis algoritma *Gradient Boosting* dan *XGBoost* pada *dataset* besar maupun kecil :



```

1 explainer = shap.Explainer(xgb_best, X_train)
2 shap_values = explainer(X_test)
3
4 shap.plots.beeswarm(shap_values)
5 shap.plots.bar(shap_values)

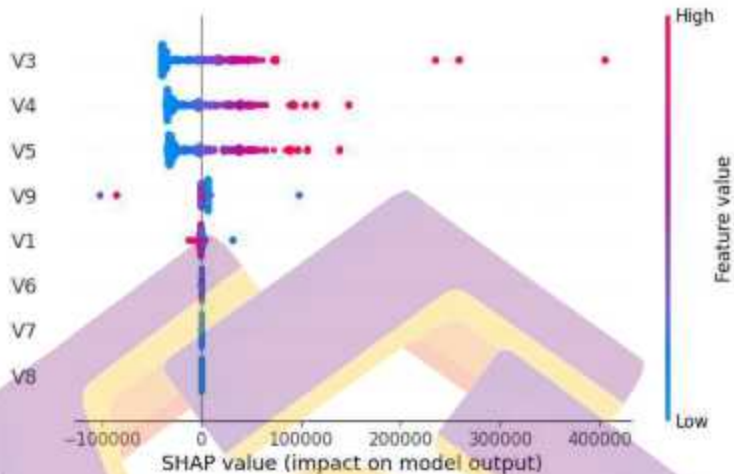
```

Gambar 4. 16. Kode Visualisasi *SHAP Analysis*

Pada gambar 4.16 memanfaatkan fungsi *shap.Explainer* untuk membangun objek *explainer* berdasarkan model *XGBoost* terbaik (*xgb_best*) dan data pelatihan (*X_train*). Objek tersebut digunakan untuk menghitung nilai *SHAP* dari data uji (*X_test*). Selanjutnya, hasil *shap_values* divisualisasikan menggunakan dua jenis grafik yaitu *shap.plots.beeswarm()* dan *shap.plots.bar()*. Grafik *beeswarm* menunjukkan distribusi pengaruh fitur terhadap prediksi model pada setiap sampel, sementara bar menyajikan rata-rata absolut kontribusi setiap fitur dalam model.

4.7.1. *SHAP Analysis Dataset Besar*

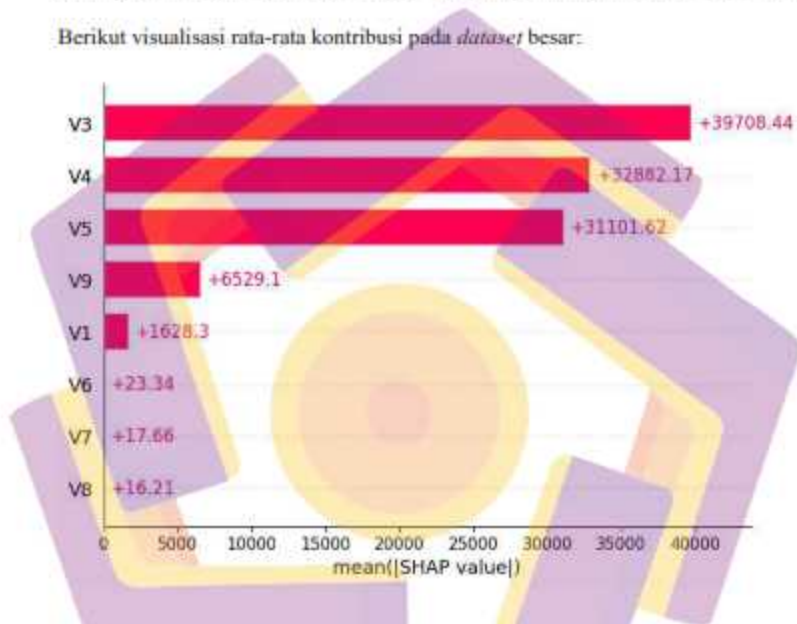
SHAP Analysis pada *dataset* besar digunakan untuk mendapatkan pemahaman mendalam mengenai kontribusi setiap fitur terhadap prediksi harga jual yang dihasilkan oleh model *XGBoost*. Analisis ini memanfaatkan dua bentuk visualisasi. Berikut hasil dari visualisasi *SHAP Analysis* pada *dataset* besar :



Gambar 4. 17. *SHAP Beeswarm Plot – Dataset Besar*

Pada gambar 4.17 menampilkan analisis interpretabilitas model *XGBoost* menggunakan *SHAP* (*Shapley Additive exPlanations*). Visualisasi ini memperlihatkan seberapa besar kontribusi setiap fitur terhadap output prediksi model, dengan sumbu *horizontal* menunjukkan nilai *SHAP* dan warna mewakili nilai fitur dari rendah (biru) hingga tinggi (merah). Dari hasil analisis terlihat bahwa fitur harga beli(V3), harga kompetitor 1(V4), dan harga kompetitor 2(V5) memberikan pengaruh paling dominan terhadap prediksi, baik dalam meningkatkan maupun menurunkan nilai target. Titik merah pada ketiga fitur tersebut menunjukkan bahwa nilai tinggi cenderung memberikan kontribusi positif besar terhadap output model. Sementara itu, fitur volume pembelian(V9) juga memberikan pengaruh, namun relatif lebih kecil dibandingkan ketiga variabel harga. Sebaliknya, fitur kategorikal seperti produk(V1), permintaan pasar(V6),

ketersediaan barang(V7), dan musim(V8) hanya memiliki dampak minimal terhadap prediksi. Hasil ini menegaskan bahwa model terutama mengandalkan variabel harga dalam menghasilkan prediksi, dengan dukungan tambahan dari volume pembelian, sedangkan faktor non-harga kurang berperan signifikan. Berikut visualisasi rata-rata kontribusi pada *dataset* besar:



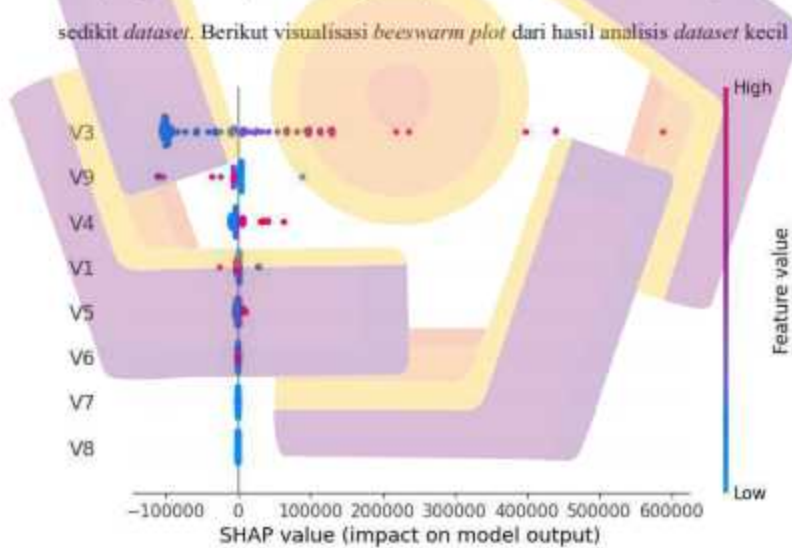
Gambar 4. 18. *SHAP Bar Plot – Mean Absolute SHAP Value Dataset Besar*

Gambar 4.18 menampilkan bahwa harga beli (V3) memiliki pengaruh paling besar dengan rata-rata nilai *SHAP* sebesar 39,708,44, yang berarti perubahan pada harga beli memberikan dampak paling signifikan terhadap hasil prediksi. Selanjutnya, harga kompetitor 1 (V4) juga menunjukkan kontribusi tinggi dengan rata-rata nilai *SHAP* 32,882,17, diikuti oleh harga kompetitor 2(V5) dengan rata-rata 31,101,62. Fitur volume pembelian (V9) memberikan pengaruh menengah dengan rata-rata 6,529,1, sementara produk(V1) hanya menyumbang rata-rata

1,628,3. Adapun fitur lainnya seperti permintaan pasar(V6), ketersediaan barang(V7), dan musim(V8) hanya memiliki pengaruh sangat kecil dengan rata-rata nilai SHAP masing-masing 23,34,17,66, dan 16,21. Temuan ini menunjukkan bahwa variabel harga (V3, V4, V5) menjadi faktor dominan dalam menentukan prediksi, sedangkan faktor non-harga hanya memberikan kontribusi minimal.

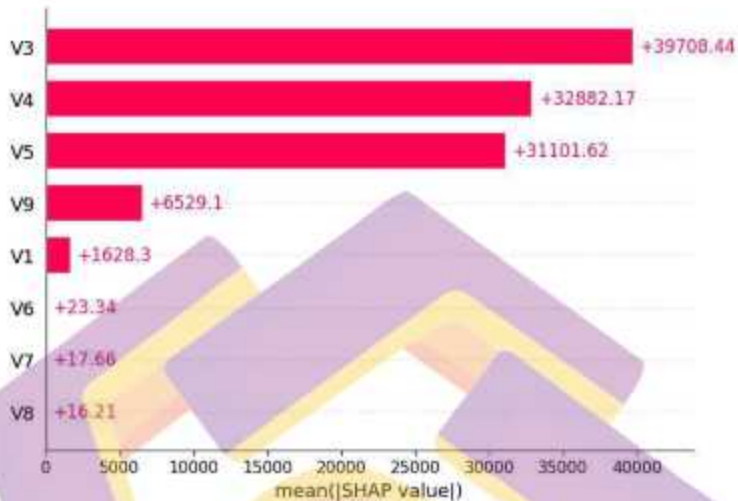
4.7.2. SHAP Analysis Dataset Kecil

SHAP analysis juga diterapkan pada dataset kecil untuk mengevaluasi seberapa besar pengaruh masing-masing fitur dalam kondisi data yang terbatas. Tujuan dari analisis ini untuk mengidentifikasi fitur dominan yang paling mempengaruhi hasil prediksi harga jual pada model XGBoost ketika hanya tersedia sedikit dataset. Berikut visualisasi beeswarm plot dari hasil analisis dataset kecil :



Gambar 4. 19. SHAP Beeswarm Plot – Dataset Kecil

Berdasarkan gambar 4.19 menampilkan bahwa harga beli(V3) menjadi fitur paling berpengaruh, dimana nilai *SHAP* yang tinggi secara konsisten memberikan kontribusi positif besar terhadap prediksi, sedangkan nilai rendah cenderung mengurangi prediksi. Selanjutnya, volume pembelian(V9) juga memberikan dampak yang cukup signifikan, meskipun kontribusinya lebih rendah dibandingkan harga beli(V3). Fitur harga kompetitor 1(V4) menempati posisi berikutnya dengan pengaruh yang masih terlihat jelas, sementara produk(V1) hanya memberikan dampak minor terhadap *output* model. Adapun fitur lain seperti harga kompetitor 2(V5), permintaan pasar(V6), ketersediaan barang(V7), dan musim(V8) hampir tidak menunjukkan kontribusi berarti. Secara keseluruhan, *plot* ini menegaskan bahwa pada dataset kecil, model *XGBoost* sangat bergantung pada variabel harga dan volume, terutama harga beli, sebagai faktor penentu utama prediksi, sementara faktor non-harga memiliki pengaruh yang sangat terbatas. Berikut visualisasi rata-rata kontribusi setiap fitur pada *dataset* kecil :



Gambar 4. 20. *SHAP Bar Plot – Mean Absolute SHAP Value Dataset Kecil*

Gambar 4.20 memperlihatkan bahwa harga beli(V3) memiliki pengaruh paling dominan dengan nilai rata-rata *SHAP* tertinggi yaitu sekitar 39,708,44, diikuti oleh harga kompetitor 1(V4) dengan nilai 32,882,17, serta harga kompetitor 2(V5) dengan nilai 31,101,62. Ketiga fitur harga ini memberikan dampak yang signifikan terhadap prediksi model, menegaskan bahwa aspek harga menjadi faktor utama dalam menentukan hasil prediksi. Selanjutnya, volume pembelian(V9) menunjukkan kontribusi menengah dengan rata-rata *SHAP* sebesar 6,529,1, sementara produk(V1) hanya memiliki pengaruh kecil dengan nilai 1,628,3. Adapun fitur lainnya, yaitu permintaan pasar(V6), ketersediaan barang(V7), dan musim(V8), hampir tidak berpengaruh dengan nilai *SHAP* rata-rata masing-masing 23,34, 17,66, dan 16,21. Dengan demikian, hasil ini menegaskan bahwa pada

dataset kecil, variabel harga tetap menjadi penentu utama prediksi, sedangkan faktor non-harga memiliki kontribusi yang sangat terbatas.

4.8. Analisis Skenario Pengujian

Untuk memperoleh pemahaman yang lebih mendalam terkait kondisi kinerja masing-masing algoritma, dilakukan pengujian terhadap dua skenario *dataset* yang berbeda, yaitu *dataset* besar dan *dataset* kecil. *Dataset* besar terdiri dari 5.300 data mingguan dalam satu tahun, sedangkan *dataset* kecil mencakup data selama satu bulan.

Hasil Pengujian menunjukkan bahwa algoritma *Gradient Boosting* maupun *XGBoost* sama-sama mampu memberikan kinerja prediksi yang sangat baik dengan nilai R^2 sebesar 0,9999 pada kedua skenario *dataset*. Perbedaan utama terletak pada nilai *MAPE* yang menunjukkan tingkat kesalahan relatif. Pada *dataset* besar, *Gradient Boosting* memperoleh nilai *MAPE* sebesar 1,82%, sedikit lebih baik dibandingkan *XGBoost* dengan *MAPE* 1,85%. Sebaliknya, pada *dataset* kecil, *XGBoost* menghasilkan nilai *MAPE* 1,05%, sedikit lebih rendah daripada *Gradient Boosting* yang memiliki *MAPE* 1,07%. Perbedaan ini relatif kecil, namun tetap menunjukkan kecenderungan bahwa *Gradient Boosting* lebih stabil pada *dataset* berukuran besar, sedangkan *XGBoost* lebih adaptif pada *dataset* kecil.

Fenomena ini dapat dijelaskan dari karakteristik masing-masing algoritma. *Gradient Boosting* menggunakan mekanisme *boosting* bertahap, dimana model dibangun secara iteratif dengan memperbaiki kesalahan model sebelumnya. Dengan ukuran data yang lebih besar, mekanisme ini mampu menyerap kompleksitas pola data secara lebih efektif sehingga menghasilkan prediksi yang

stabil dan akurat. Sementara itu, *XGBoost* memiliki keunggulan dalam hal regularisasi melalui parameter *reg_alpha(L1)* dan *reg_lambda(L2)*, yang dapat mengendalikan kompleksitas model serta cegah *overfitting*. Keunggulan ini membuat *XGBoost* lebih unggul ketika diaplikasikan pada *dataset* berukuran kecil, karena mampu mempertahankan kinerja prediksi yang stabil terhadap variasi data yang terbatas.

Hasil pengujian ini sejalan dengan temuan pada penelitian yang dilakukan oleh [6] yang membandingkan *Gradient Boosting* dan *XGBoost* dalam prediksi survival COVID-19 dan menemukan bahwa *XGBoost* lebih stabil pada *dataset* kecil, sedangkan *Gradient Boosting* menunjukkan keunggulan pada *dataset* yang lebih besar dan kompleks. Selain itu, penelitian oleh [25] yang menggunakan *XGBoost* untuk analisis risiko kredit juga menekankan efektifitas regularisasi *XGBoost* dalam mengatasi *overfitting* pada *dataset* berukuran terbatas.

Dari skenario ini dapat disimpulkan bahwa pemilihan algoritma ideal sangat bergantung pada karakteristik data yang digunakan. *Gradient Boosting* direkomendasikan untuk data berukuran besar dan kompleks, sedangkan *XGBoost* lebih cocok digunakan pada data yang terbatas namun tetap mengandung variasi signifikan antar fitur.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan algoritma *Gradient Boosting* dan *XGBoost* dalam menentukan harga jual optimal di toko pasar tradisional memberikan hasil yang akurat dan dapat diandalkan. Model yang dibangun mampu memanfaatkan data historis penjualan dan faktor eksternal seperti harga kompetitor serta kondisi pasar untuk menghasilkan prediksi harga jual yang objektif dan adaptif. Adapun kesimpulan yang menjawab setiap pertanyaan dalam rumusan masalah adalah sebagai berikut:

1. Berdasarkan hasil analisis menggunakan algoritma *Gradient Boosting* dan *XGBoost*, diperoleh bahwa variabel harga beli, harga kompetitor 1, dan harga kompetitor 2 merupakan faktor yang paling dominan dalam menentukan harga jual di toko pasar tradisional.
 - Pada hasil *feature importance* model *Gradient Boosting* menggunakan dataset besar, ketiga variabel tersebut memiliki bobot *feature importance* tertinggi, masing-masing sebesar 0,31 (harga beli), 0,29(harga kompetitor 1), dan 0,27(harga kompetitor 2).
 - Hasil analisis *SHAP* juga menunjukkan bahwa harga beli (V3) memiliki rata-rata nilai *SHAP* tertinggi sebesar 39,708 diikuti oleh harga kompetitor 1 sebesar 32,882 dan harga kompetitor 2 sebesar 31,101.

- Sebaliknya, variabel non-harga seperti permintaan pasar, ketersediaan barang, dan faktor musiman memiliki nilai *feature importance* dan nilai *SHAP* jauh lebih kecil masing-masing dibawah 1% kontribusi total model.

Temuan ini menegaskan bahwa komponen harga merupakan faktor utama yang menentukan nilai jual produk di toko pasar tradisional, sedangkan faktor eksternal non-harga hanya memberikan variasi kecil terhadap perubahan harga. Dengan demikian, strategi penentuan harga di pasar tradisional sebaiknya difokuskan pada analisis terstruktur biaya dan perbandingan harga kompetitor.

2. Faktor utama yang memengaruhi harga jual berdasarkan hasil *feature importance* dan analisis *SHAP* adalah harga beli (V3), harga kompetitor 1(V4), harga kompetitor 2(V5). Ketiga variabel ini secara konsisten muncul sebagai kontributor terbesar dalam pembentukan *output* prediksi pada kedua algoritma, sementara variabel non-harga seperti permintaan pasar, ketersediaan barang, musim, dan nama produk hanya memberikan pengaruh yang relatif lebih kecil.
3. Kinerja model pada dua skenario *dataset* menunjukkan bahwa baik *Gradient Boosting* maupun *XGBoost* sama-sama memiliki kemampuan prediksi yang sangat baik dengan nilai R^2 mencapai 0,9999 dan *MAPE* dibawah 2%. Pada *dataset* besar, *Gradient Boosting* sedikit lebih unggul dengan *MAPE* 1,82% dibanding *XGBoost* 1,85%, sehingga lebih sesuai untuk data berukuran besar dan kompleks. Sebaliknya, pada *dataset* kecil, *XGBoost* menunjukkan akurasi

yang lebih baik dengan *MAPE* 1,05% dibanding *Gradient Boosting* 1,07%, karena keunggulan regularisasi ganda yang mampu mencegah *overfitting*. Hal ini membuktikan bahwa pemilihan algoritma terbaik sangat bergantung pada karakteristik ukuran dan kompleksitas data yang digunakan.

5.2. Saran

Berdasarkan hasil temuan dan keterbatasan yang dihadapi selama penelitian, disarankan agar penelitian selanjutnya dapat memperluas cakupan data dengan melibatkan lebih banyak toko dan variabel tambahan, seperti biaya operasional dan perilaku konsumen. Selain itu, integrasi sistem prediksi harga jual ke dalam platform berbasis aplikasi atau sistem informasi berbasis web dapat dikembangkan untuk membantu pedagang dalam mengambil keputusan harga secara *real-time* dan lebih praktis. Penelitian lanjutan juga dapat mengeksplorasi penggunaan pendekatan *ensemble learning* lainnya seperti *LightGBM* atau *CatBoost*, serta melakukan perbandingan kinerja algoritma dalam konteks pasar dengan karakteristik yang berbeda. Adapun dalam konteks implementasi praktis, diperlukan pelatihan dan pendampingan bagi pedagang agar mereka mampu memanfaatkan teknologi prediktif ini secara optimal dalam pengelolaan usaha.

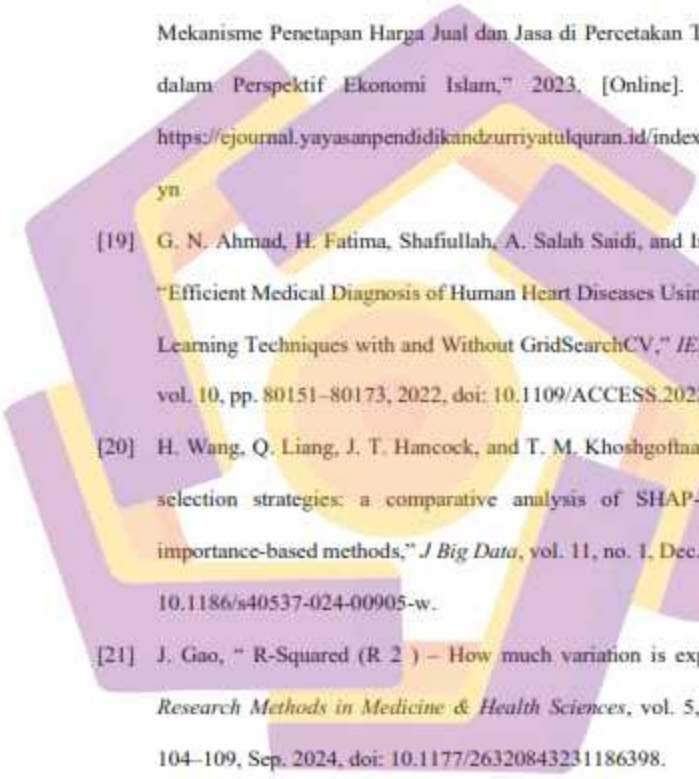
DAFTAR PUSTAKA

- [1] A. Syukria, "Potensi Pasar Tradisional dalam Peningkatan Ekonomi Masyarakat," *Jurnal Informatika Ekonomi Bisnis*, pp. 306–311, Jun. 2023, doi: 10.37034/infec.v5i2.221.
- [2] F. Wibowo, A. U. Khasanah, I. Fitroh, and S. Putra, "Analisis Dampak Kehadiran Pasar Modern terhadap Kinerja Pemasaran Pasar Tradisional Berbasis Perspektif Pedagang dan Konsumen di Kabupaten Wonogiri," *Benefit: Jurnal Manajemen dan Bisnis*, vol. 7, no. 1, pp. 53–65, 2022, Accessed: Jun. 22, 2025. [Online]. Available: <https://doi.org/10.23917/benefit.v7i1.16057>
- [3] F. Sampe, R. Ardianto, M. Yusuf, and U. Sultan Aji Muhammad Idris Samarinda, "The Impact Of Brand Image And Price Online Product Purchase Decisions At Shopee," *Asian Journal of Management Entrepreneurship and Social Science*, 2023, [Online]. Available: <https://ajmesc.com/index.php/ajmesc>
- [4] E. Permatasari, H. Luthfiana, N. A. Pratama, and H. Ali, "Faktor-faktor yang Mempengaruhi Pembelian Ulang: Promosi, Harga dan Produk," *Jurnal Ilmu Manajemen Terapan*, vol. 3, no. 5, 2022, doi: 10.31933/jimt.v3i5.
- [5] A. Rusman Salch, M. Hasan, and T. Supatminingsih, "Analisis Faktor Penyebab Turunnya Harga Jual Beli Sayuran di Pasar Cakke Kecamatan Anggeraja Kabupaten Enrekang," *INTELEKTIVA : JURNAL EKONOMI, SOSIAL DAN HUMANIORA*, May 2022,

- Accessed: Jun. 22, 2025. [Online]. Available: <https://www.jurnalintelektiva.com/index.php/jurnal/article/view/741>.
- [6] N. T. Razavizadeh, M. Salari, M. Jafari, E. Sabaghian, and V. Ghavami, "Comparison of Two Methods, Gradient Boosting and Extreme Gradient Boosting to Predict Survival in Covid-19 Data," 2023. Accessed: Jun. 22, 2025. [Online]. Available: <https://doi.org/10.18502/jbe.v9i3.15450>
- [7] I. Maulita, C. R. A. Widinwati, and A. M. Wahid, "Analisis Komparatif Linear Regression, Random Forest, dan Gradient Boosting untuk Prediksi Banjir," *Jurnal Pendidikan dan Teknologi Indonesia (JPTI)*, Aug. 2024, doi: DOI:10.52436/1.jpti.599.
- [8] X. Chen, J. Hua, C. Yin, Z. Chen, Y. Chen, and Z. Li, "A Precise Life Estimation Method for Retired Energy Storage Batteries Based on Energy Storage Batteries Attenuation Characteristics and XGBoost Algorithm," *IEEE Access*, vol. 11, pp. 135968–135978, 2023, doi: 10.1109/ACCESS.2023.3336813.
- [9] P. Anandhi and Dr. E. Nathiya, "Application of linear regression with their advantages, disadvantages, assumption and limitations," *International Journal of Statistics and Applied Mathematics*, vol. 8, no. 6, pp. 133–137, Nov. 2023, doi: 10.22271/math.2023.v8.i6b.1463.
- [10] İ. Levent, G. Şahin, G. Işık, and W. G. J. H. M. van Sark, "Comparative Analysis of Advanced Machine Learning Regression Models with Advanced Artificial Intelligence Techniques to Predict Rooftop PV

Solar Power Plant Efficiency Using Indoor Solar Panel Parameters,” *Applied Sciences (Switzerland)*, vol. 15, no. 6, Mar. 2025, doi: 10.3390/app15063320.

- [11] N. Nguyen and D. Ngo, “Comparative analysis of boosting algorithms for predicting personal default,” *Cogent Economics and Finance*, vol. 13, no. 1, 2025, doi:10.1080/23322039.2025.2465971.
- [12] Y. Zou, C. Gao, and H. Gao, “Business Failure Prediction Based on a Cost-Sensitive Extreme Gradient Boosting Machine,” *IEEE Access*, vol. 10, pp. 42623–42639, 2022, doi: 10.1109/ACCESS.2022.3168857.
- [13] B. Bhattacharjee *et al.*, “Machine Learning For Cost Estimation and Forecasting in Banking: A Comparative Analysis of Algorithms,” *Frontline Marketing, Management and Economics Journal*, vol. 04, no. 12, pp. 66–83, Dec. 2024, doi: 10.37547/marketing-fimmej-04-12-06.
- [14] D. N. Gono, H. Napitupulu, and Firdaniza, “Silver Price Forecasting Using Extreme Gradient Boosting (XGBoost) Method,” *Mathematics*, vol. 11, no. 18, Sep. 2023, doi: 10.3390/math11183813.
- [15] O. Alshboul, A. Shehadeh, G. Almasabha, and A. S. Almuflih, “Extreme Gradient Boosting-Based Machine Learning Approach for Green Building Cost Prediction,” *Sustainability (Switzerland)*, vol. 14, no. 11, Jun. 2022, doi: 10.3390/su14116651.
- [16] Siti Haniatunnisa, “Implementasi Hukum Ekonomi Syariah pada Pasar Tradisional Siti Haniatunnisa,” 2022.

- 
- [17] A. N. Ilmi, M. Mahmud, and A. S. Hardiani, "Characterization of Modern Markets and Traditional Markets (Case Study In Grandmall and Maros Central Market)," 2022.
- [18] H. Lisa, A. Syukur, and S. Auliaurasyidin Tembilahan, "Analisis Mekanisme Penetapan Harga Jual dan Jasa di Percetakan Tembilahan dalam Perspektif Ekonomi Islam," 2023. [Online]. Available: <https://ejournal.yayasanpendidikandzurriyatulquran.id/index.php/AIZayn>.
- [19] G. N. Ahmad, H. Fatima, Shafuallah, A. Salah Saidi, and Imdadullah, "Efficient Medical Diagnosis of Human Heart Diseases Using Machine Learning Techniques with and Without GridSearchCV," *IEEE Access*, vol. 10, pp. 80151–80173, 2022, doi: 10.1109/ACCESS.2022.3165792.
- [20] H. Wang, Q. Liang, J. T. Hancock, and T. M. Khoshgoftar, "Feature selection strategies: a comparative analysis of SHAP-value and importance-based methods," *J Big Data*, vol. 11, no. 1, Dec. 2024, doi: 10.1186/s40537-024-00905-w.
- [21] J. Gao, "R-Squared (R^2) – How much variation is explained?," *Research Methods in Medicine & Health Sciences*, vol. 5, no. 4, pp. 104–109, Sep. 2024, doi: 10.1177/26320843231186398.
- [22] J. Khatti and K. S. Grover, "Prediction of compaction parameters for fine-grained soil: Critical comparison of the deep learning and standalone models," *Journal of Rock Mechanics and Geotechnical*

Engineering, vol. 15, no. 11, pp. 3010–3038, Nov. 2023, doi: 10.1016/j.jrmge.2022.12.034.

[23] S. Peta, "Python- An Appetite for the Software Industry," *International Journal of Programming Languages and Applications*, vol. 12, no. 4, pp. 1–14, Oct. 2022, doi: 10.5121/ijpla.2022.12401.

[24] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltip.2022.04.020.

[25] A. A. Saputra, B. N. Sari, and C. Rozikin, "Penerapan Algoritma Extreme Gradient Boosting (Xgboost) Untuk Analisis Risiko Kredit," *Jurnal Ilmiah Wahana Pendidikan*, vol. 10, no. 7, pp. 27–36, 2024, doi: 10.5281/zenodo.10960080.