

Implementasi K-Means sebagai Mekanisme Self-Labeling dalam Arsitektur Ensemble Voting Classifier untuk Prediksi Penjualan Usaha Mikro Kecil dan Menengah (UMKM) pada Data Tanpa Label

Muhammad Aqil Fahmi*, Defri Kurniawan

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia

Email: ^{1,*}111202214001@mhs.dinus.ac.id, ²defri.kurniawan@dsn.dinus.ac.id

Email Penulis Korespondensi: 111202214001@mhs.dinus.ac.id

Submitted: 24/11/2025; Accepted: 26/12/2025; Published: 26/12/2025

Abstrak—Prediksi penjualan pada sektor Usaha Mikro, Kecil, dan Menengah (UMKM) menghadapi kendala karakteristik data yang fluktuatif (*noisy*) dan ketiadaan label kelas (*unlabeled*) yang diperlukan untuk pelatihan model *supervised learning*. Penelitian ini mengusulkan arsitektur hibrida sekuensial di mana algoritma K-Means difungsikan sebagai mekanisme *Self-Labeling* untuk mentransformasi data transaksi mentah menjadi label kelas ('Rendah' dan 'Tinggi') secara otomatis. Label sintesis yang dihasilkan kemudian digunakan untuk melatih model *Ensemble Voting Classifier* yang mengagregasi prediksi dari *XGBoost*, *LightGBM*, dan *CatBoost*. Hasil evaluasi eksperimental menunjukkan bahwa meskipun model tunggal *XGBoost* mencatat akurasi sedikit lebih tinggi (96.24%) dibandingkan model *Ensemble* (96.07%), model hibrida *Ensemble Voting* terbukti lebih superior dalam hal kalibrasi probabilitas dengan nilai *Loss* terendah sebesar 0.1532. Angka ini lebih baik dibandingkan *XGBoost* (0.1646) dan *LightGBM* (0.1772), mengindikasikan tingkat kepercayaan prediksi yang lebih tinggi dan stabil. Model juga menunjukkan keseimbangan yang sangat baik dengan *F1-Score* sebesar 0.95 dan *Recall* 0.96 untuk kelas mayoritas. Penelitian ini membuktikan bahwa pendekatan hibrida efektif mereduksi ketidakpastian (*uncertainty*) dalam manajemen stok UMKM.

Kata Kunci: Prediksi Penjualan; Self-Labeling; K-Means Clustering; Ensemble Voting; Loss

Abstract—Sales forecasting in the Micro, Small, and Medium Enterprises (MSME) sector faces challenges due to the fluctuating (*noisy*) nature of the data and the absence of class labels (*unlabeled*) required for training supervised learning models. This study proposes a sequential hybrid architecture in which the K-Means algorithm is employed as a Self-Labeling mechanism to automatically transform raw transaction data into class labels (“Low” and “High”). The resulting synthetic labels are then used to train an Ensemble Voting Classifier model that aggregates predictions from XGBoost, LightGBM, and CatBoost. The experimental evaluation results show that although the single XGBoost model achieves a slightly higher accuracy (96.24%) compared to the Ensemble model (96.07%), the hybrid Ensemble Voting model proves superior in terms of probability calibration, achieving the lowest Loss value of 0.1532. This value outperforms XGBoost (0.1646) and LightGBM (0.1772), indicating more reliable and stable prediction confidence. The model also demonstrates excellent balance with an F1-Score of 0.95 and a Recall of 0.96 for the majority class. This study confirms that the hybrid approach is effective in reducing uncertainty in MSME stock management.

Keywords: Sales Prediction; Self-Labeling; K-Means Clustering; Ensemble Voting; Loss

1. PENDAHULUAN

Sektor Usaha Mikro, Kecil, dan Menengah (UMKM) di Indonesia terus mengalami transformasi dinamis, didorong oleh inovasi produk dan perubahan preferensi konsumen [1], [2], [3]. Produk pelengkap seperti bawang goreng kini berkembang menjadi komoditas bernilai tambah tinggi dengan berbagai varian rasa, sehingga memperluas peluang pasar bagi pelaku UMKM [4]. Diversifikasi produk ini tidak hanya meningkatkan daya saing UMKM, tetapi juga memperluas segmentasi pasar ke berbagai kelompok konsumen dengan karakteristik yang berbeda. Inovasi kemasan, strategi pemasaran digital, serta penetrasi platform e-commerce turut mempercepat akselerasi pertumbuhan produk olahan berbasis UMKM.

Namun, pertumbuhan ini juga membawa tantangan baru, khususnya dalam pengelolaan persediaan bahan baku dan produk jadi yang bersifat mudah rusak (*perishable*) [4], [5], [6]. Manajemen stok menjadi krusial karena *overstock* dapat menyebabkan penurunan kualitas produk seperti tengik dan melempem, yang berdampak pada kerugian finansial, sedangkan kekurangan stok (*stock-out*) pada varian favorit dapat menyebabkan hilangnya peluang pendapatan dan loyalitas pelanggan [4], [5]. Dengan demikian, keseimbangan antara ketersediaan stok dan tingkat permintaan pasar menjadi faktor kunci keberlanjutan usaha.

Permasalahan semakin kompleks karena permintaan produk sangat fluktuatif, dipengaruhi oleh faktor eksternal seperti musim, hari libur, promosi daring, dan tren musiman, yang sulit diprediksi secara manual [7], [8]. Kondisi ini menyebabkan ketidakefisienan operasional yang berdampak langsung pada arus kas dan stabilitas usaha. Sebagian besar UMKM masih mengandalkan pencatatan manual atau intuisi dalam pengambilan keputusan stok [5], [9], sehingga sering terjadi *mismatch* antara prediksi dan realisasi penjualan.

Transformasi menuju manajemen berbasis data menjadi kebutuhan mendesak. Pemanfaatan teknologi *Data Mining* dan *Machine Learning* dapat membantu mengidentifikasi pola permintaan yang kompleks dan menghasilkan prediksi yang lebih akurat [10]. Namun, tantangan teknis muncul karena data penjualan UMKM cenderung memiliki *noise* tinggi dan pola yang tidak linear, sehingga metode statistik konvensional sering kali kurang efektif. Selain itu, keterbatasan sumber daya manusia dan infrastruktur teknologi juga menjadi hambatan dalam implementasi sistem prediksi yang canggih di tingkat UMKM. Meskipun prediksi penjualan secara tradisional dipandang sebagai masalah regresi (memprediksi angka pasti), karakteristik operasional UMKM seringkali lebih membutuhkan keputusan biner

yang cepat: apakah perlu melakukan restock besar-besaran atau menahan stok. Mengingat volume penjualan harian yang relatif kecil namun sangat fluktuatif, memaksakan model regresi seringkali menghasilkan Mean Absolute Error (MAE) yang menyesatkan. Oleh karena itu, mentransformasi masalah ini menjadi klasifikasi biner (Penjualan Rendah vs Tinggi) menggunakan batas keputusan yang dipelajari dari data (*data-driven threshold*) menjadi pendekatan yang lebih pragmatis dan relevan untuk mitigasi risiko *overstock* maupun *stock-out*. Selain itu, data transaksional historis umumnya tidak memiliki label kelas yang eksplisit (unlabeled), sehingga diperlukan pendekatan self-labeling untuk mentransformasi data mentah menjadi format yang dapat dipelajari oleh algoritma klasifikasi tanpa intervensi manual yang subjektif. Pendekatan ini juga mampu mengurangi ketergantungan pada penilaian manusia yang cenderung bias dan tidak konsisten.

Dalam lima tahun terakhir, sejumlah penelitian telah berupaya membangun model prediksi penjualan, namun masih menghadapi berbagai keterbatasan metodologis. Pendekatan menggunakan *Linear Regression* menunjukkan bahwa meskipun model ini sederhana dan mudah diimplementasikan, ia tidak mampu menangkap hubungan non-linear serta pola musiman yang kompleks. Akibatnya, performa model cenderung menurun pada periode tertentu seperti hari libur, sehingga menghasilkan tingkat *error* yang cukup tinggi [11], [12]. Model ini juga kurang adaptif terhadap perubahan pola permintaan yang terjadi secara tiba-tiba.

Penelitian lain yang menggunakan *Decision Tree* menunjukkan bahwa model ini dapat membentuk aturan keputusan yang jelas dan mudah diinterpretasikan. Namun, *Decision Tree* memiliki kecenderungan kuat terhadap *overfitting*, di mana model tampil sangat baik pada data latih tetapi gagal mempertahankan performanya pada data uji baru [11]. Kondisi ini membatasi penggunaannya pada skenario operasional yang dinamis seperti penjualan UMKM. Sementara itu, penggunaan *K-Means Clustering* lebih banyak difokuskan pada segmentasi pelanggan, bukan pada prediksi penjualan harian, sehingga tidak dapat memberikan keluaran berupa estimasi penjualan untuk periode berikutnya [13].

Di sisi lain, evaluasi terhadap metode deret waktu seperti ARIMA mengungkapkan bahwa model ini membutuhkan data historis yang panjang dan kondisi data yang stasioner. ARIMA kurang efektif ketika diterapkan pada data UMKM yang sering kali minim, tidak stabil, dan mengandung variabel kategori seperti jenis promo atau cuaca yang tidak dapat dimodelkan secara langsung [13], [14], [15]. Sebagai pembanding, algoritma *ensemble* modern seperti XGBoost dan LightGBM justru menunjukkan performa yang lebih unggul dalam menangani data kompleks dengan variabel campuran, sehingga memberikan hasil prediksi yang lebih konsisten dan akurat [11], [12], [15]. Keunggulan ini menjadikan metode ensemble sebagai kandidat kuat dalam pengembangan sistem prediksi penjualan berbasis data di sektor UMKM.

Berdasarkan tinjauan pustaka tersebut, terlihat bahwa mayoritas penelitian sebelumnya cenderung menggunakan metode tunggal yang memiliki keterbatasan parsial. Metode statistik klasik terkendala asumsi data, sementara metode *machine learning* tunggal rentan terhadap bias atau *overfitting*. Di sinilah letak urgensi penerapan metode hibrida yang mampu mensinergikan dua pendekatan berbeda untuk menangani karakteristik data UMKM yang unik (tanpa label dan fluktuatif). Dalam skema yang diusulkan, K-Means Clustering berperan sebagai mekanisme *pre-processing* cerdas untuk melakukan *self-labeling*, mengubah data transaksi mentah menjadi kelas kategori yang objektif. Hasil pelabelan ini kemudian menjadi landasan yang kokoh bagi Ensemble Voting Classifier (kombinasi XGBoost, LightGBM, dan CatBoost) untuk mempelajari pola kompleks dan melakukan prediksi dengan tingkat generalisasi yang lebih tinggi. Meskipun potensi sinergi ini besar, belum banyak penelitian yang mengintegrasikan pendekatan Unsupervised Learning (untuk pelabelan target otomatis yang objektif) [16] dengan Ensemble Supervised Learning tingkat lanjut (Voting Classifier) secara hibrida pada konteks data UMKM [17].

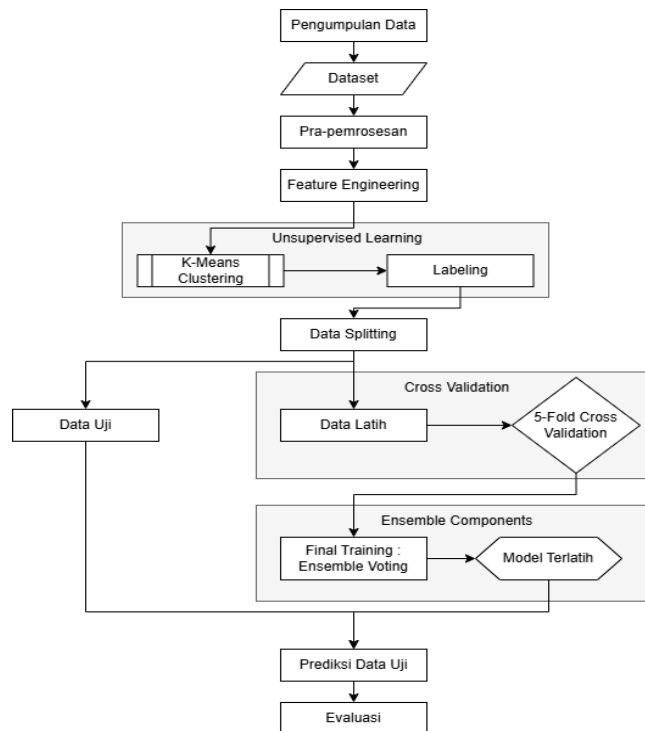
Penelitian ini hadir untuk mengisi kekosongan tersebut dengan menggabungkan *K-Means* untuk menentukan status “Laris/Sepi” secara dinamis, dan *Ensemble Voting (XGBoost, LightGBM, CatBoost)* untuk prediksi akurasi tinggi. Tujuan utama dari penelitian ini adalah mengembangkan model prediksi hibrida yang mampu mengklasifikasikan potensi penjualan harian UMKM ke dalam kategori performa (“Rendah” atau “Tinggi”) dengan tingkat akurasi yang tinggi. Secara spesifik, penelitian ini bertujuan untuk: (1) Menentukan ambang batas penjualan yang optimal menggunakan *K-Means Clustering*; (2) Membangun model klasifikasi yang *robust* menggunakan teknik *Ensemble Voting*; dan (3) Menganalisis fitur-fitur dominan (*feature importance*) yang mempengaruhi penjualan. Kontribusi ini diharapkan dapat mengisi *gap* dan menghasilkan sistem komputasi andal dalam mengambil keputusan.

2. METODOLOGI PENELITIAN

Penelitian ini mengusulkan model prediksi hibrida yang mengintegrasikan metode *Unsupervised Learning* untuk pelabelan data dan *Ensemble Supervised Learning* untuk klasifikasi. Metodologi penelitian dirancang secara sistematis untuk memastikan bahwa model yang dihasilkan tidak hanya akurat tetapi juga robust dan tervalidasi.

2.1 Tahapan Penelitian

Untuk memberikan gambaran menyeluruh mengenai proses komputasi yang dilakukan, alur kerja sistem divisualisasikan pada diagram berikut.



Gambar 1. Alur Penelitian

Tahapan penelitian ini mengikuti kerangka kerja *data mining* yang terstruktur, dimulai dari pengumpulan data hingga evaluasi model. Setiap tahapan dirancang untuk mengolah data mentah (*raw data*) menjadi sebuah model prediktif yang dapat memberikan wawasan *actionable* bagi pelaku UMKM. Alur proses penelitian secara visual disajikan pada Gambar 1.

a. Pengumpulan Data (*Data Collection*)

Tahap awal adalah pengumpulan data transaksional harian dari studi kasus pada UMKM Bawang Goreng Iqoh Putri Limbangan, Brebes. Data ini diperoleh melalui observasi operasional harian dan wawancara langsung dengan pemilik usaha. Data yang dikumpulkan merupakan data sekunder (historis) yang mencakup berbagai atribut yang relevan, seperti yang dijelaskan pada Tabel 1, dan menjadi dasar untuk proses *feature engineering*.

b. Pra-pemrosesan dan *Feature Engineering*

Data mentah yang diperoleh kemudian melalui proses pra-pemrosesan untuk membersihkan dan mentransformasi data. Proses ini krusial karena kualitas data akan sangat mempengaruhi performa model.

1. *Data Cleaning*

Melakukan penanganan terhadap *missing values* (data kosong) menggunakan imputasi statistik (strategi *median* untuk numerik dan *most_frequent* untuk kategorikal) [18] dan melakukan konversi tipe data (seperti Tanggal ke format *datetime* dan Diskon ke format numerik).

2. *Feature Engineering*

Proses ini bertujuan untuk menciptakan fitur-fitur turunan (*derived features*) yang memiliki daya prediktif lebih tinggi guna meningkatkan performa model. Berdasarkan analisis pada kode, rekayasa fitur dilakukan melalui beberapa pendekatan, dimulai dengan transformasi data promosi dan hari libur menjadi format biner (0 atau 1), serta pembentukan fitur finansial seperti Harga_Akhir dan Nilai_Stok_Rp. Selain itu, interaksi antar-variabel diperhitungkan melalui fitur rasio seperti Rasio_Penjualan_Stok dan Stok_x_Promosi. Untuk menangkap dinamika waktu, dilakukan ekstraksi informasi temporal (seperti Bulan, Hari_Num, dan Is_Weekend) serta pembuatan fitur historis berupa *Lag* (Penjualan_Lag1 dan Penjualan_Lag7). Penerapan fitur *Lag* yang mengubah masalah *time series* menjadi format *supervised learning* ini terbukti sangat efektif dalam menangkap pola tren maupun musiman (*seasonality*), serta meningkatkan akurasi model pada data ritel berskala besar [19].

c. Penentuan Target Otomatis (*Unsupervised Labeling*)

Berbeda dengan pendekatan konvensional yang menggunakan ambang batas subjektif (misal, penjualan di atas rata-rata), penelitian ini menggunakan metode *Unsupervised Learning* untuk menentukan label target secara objektif. Algoritma *K-Means Clustering* (seperti pada kode, dengan $n_clusters=2$) diterapkan pada data Penjualan (kg) untuk mengelompokkan data secara alami menjadi dua klaster: 'Rendah' (label 0) dan 'Tinggi' (label 1).

d. Pemisahan Data (*Data Splitting*)

Setelah data memiliki fitur dan label, dataset dibagi menjadi dua bagian: Data Latih (*Training Set*) dan Data Uji (*Testing Set*). Berdasarkan kode, proporsi yang digunakan adalah 80% data latih dan 20% data uji ($test_size=0.2$), dengan pengacakan data ($shuffle=True$) untuk memastikan distribusi data yang merata.



e. Validasi Model (*Cross-Validation*)

Sebelum pelatihan akhir, dilakukan validasi model pada data latih untuk mendapatkan estimasi performa yang stabil. Metode *5-Fold Cross-Validation* (menggunakan *StratifiedKfold*) diterapkan. Tahap ini penting untuk memastikan model tidak *overfitting* dan memiliki kemampuan generalisasi yang baik, karena metode ini memberikan estimasi kesalahan prediksi yang tidak bias pada data yang belum pernah dilihat sebelumnya [20]. Hasil dari tahap ini adalah nilai rata-rata akurasi dan standar deviasi.

f. Pembangunan Model *Ensemble* (*Final Training*)

Model *ensemble* yang terdiri dari empat algoritma (*base learners*)—*XGBoost*, *LightGBM*, *CatBoost*, dan *Random Forest* yang kemudian dilatih secara final menggunakan keseluruhan Data Latih.

g. Evaluasi Model (*Evaluation*)

Model terlatih yang dihasilkan kemudian diuji performanya menggunakan Data Uji yang belum pernah dilihat sebelumnya. Tahap ini menghasilkan prediksi akhir. Kinerja model klasifikasi mencerminkan seberapa baik model dalam mengidentifikasi dan mengelompokkan sampel penjualan dengan benar ke dalam kategori performa ("Rendah" atau "Tinggi"). Evaluasi performa model dalam penelitian ini dilakukan menggunakan empat metrik utama, yaitu Akurasi, Presisi, Recall, dan F1-score. Berdasarkan Confusion Matrix, hasil prediksi dikategorikan ke dalam empat kemungkinan:

1. True Positive (TP): Kondisi di mana penjualan aktual adalah "Tinggi" dan model memprediksinya dengan benar sebagai "Tinggi".
2. True Negative (TN): Kondisi di mana penjualan aktual adalah "Rendah" dan model memprediksinya dengan benar sebagai "Rendah".
3. False Positive (FP): Kondisi di mana penjualan aktual sebenarnya "Rendah", tetapi model salah mengklasifikasikannya sebagai "Tinggi".
4. False Negative (FN): Kondisi di mana penjualan aktual sebenarnya "Tinggi", tetapi model salah mengklasifikasikannya sebagai "Rendah".

Metrik-metrik evaluasi yang digunakan dijelaskan sebagai berikut:

1. Akurasi (*Accuracy*)

Mengukur persentase total data uji yang berhasil diklasifikasikan dengan benar, baik sebagai penjualan rendah maupun tinggi. Semakin tinggi akurasi, semakin baik model secara keseluruhan. Akurasi dihitung menggunakan rumus:

$$Akurasi = \frac{TP+TN}{TP + TN + FP + FN} \tag{1}$$

2. Presisi (*Precision*)

Mengukur seberapa andal model dalam memprediksi kategori positif (Penjualan Tinggi). Metrik ini menunjukkan proporsi prediksi "Tinggi" yang benar-benar terjadi. Rumus presisi adalah:

$$Presisi = \frac{TP}{TP + FP} \tag{2}$$

3. Recall (*Sensitivitas*)

Mengukur sejauh mana model dapat mendeteksi seluruh sampel yang benar-benar termasuk dalam kategori positif (Penjualan Tinggi). Semakin tinggi recall, semakin sedikit peluang penjualan tinggi yang terlewatkan oleh model. Recall dirumuskan sebagai:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

4. F1-Score

Merupakan rata-rata harmonik antara presisi dan recall. Metrik ini berguna untuk menyeimbangkan trade-off antara kesalahan False Positive dan False Negative, terutama dalam kondisi di mana distribusi kelas mungkin tidak seimbang sempurna. F1-score dirumuskan sebagai:

$$F1 - score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \tag{4}$$

5. Kurva ROC dan AUC (*Receiver Operating Characteristic - Area Under Curve*)

Selain metrik berbasis ambang batas (*threshold*) tunggal di atas, penelitian ini juga menggunakan kurva ROC untuk mengevaluasi kemampuan model dalam membedakan kelas pada berbagai ambang batas klasifikasi. Kurva ROC memplot rasio True Positive Rate (TPR) atau sensitivitas pada sumbu Y melawan False Positive Rate (FPR) pada sumbu X. True Positive Rate (TPR) sama dengan Recall (lihat Persamaan 3), menggambarkan seberapa banyak kelas positif ("Tinggi") yang terdeteksi dengan benar. Sedangkan False Positive Rate (FPR) menggambarkan proporsi sampel negatif ("Rendah") yang salah diklasifikasikan sebagai positif. FPR dihitung menggunakan rumus:

$$FPR = \frac{FP}{TN + FP} \tag{5}$$

Kinerja keseluruhan kurva ROC diringkas menjadi satu nilai skalar yang disebut AUC (*Area Under Curve*).

2.2 Metode Penyelesaian Masalah

Metode yang digunakan dalam penelitian ini adalah pendekatan *Hybrid Machine Learning*, yang secara rinci dijelaskan dalam tahapan-tahapan berikut.

a. Daftar Fitur Penelitian

Untuk membangun model yang akurat, penelitian ini menggunakan 18 fitur yang berasal dari data asli maupun hasil *feature engineering*. Fitur-fitur tersebut dikategorikan menjadi numerik dan kategorikal untuk diproses lebih lanjut sesuai dengan karakteristik masing-masing.

Tabel 1. Daftar Fitur Penelitian dan Tipe Data

Fitur	Tipe Data	Keterangan
Varian	Kategorikal	Jenis varian produk
Lokasi	Kategorikal	Lokasi penjualan atau cabang
Cuaca	Kategorikal	Kondisi cuaca harian
Hari	Kategorikal	Nama hari
Promosi_bin	Numerik	Status promosi (1=Ya, 0=Tidak)
Libur_bin	Numerik	Status hari libur (1=Ya, 0=Tidak)
Stok_Ready (kg)	Numerik	Jumlah stok yang tersedia di awal hari
Harga per kg (Rp)	Numerik	Harga jual produk per kg
Diskon_frac	Numerik	Nilai diskon dalam format fraksi
Harga_Akhir	Numerik	Fitur turunan: Harga setelah dikurangi diskon
Nilai_Stok_Rp	Numerik	Fitur turunan: Total nilai stok dalam Rupiah
Stok_x_Promosi	Numerik	Fitur turunan: Interaksi antara stok dan promosi
Rasio_Penjualan_Stok	Numerik	Fitur turunan: (Penjualan / Stok_Ready)
Bulan	Numerik	Fitur turunan: Bulan dalam angka (1-12)
Hari_ke	Numerik	Fitur turunan: Hari dalam tahun (1-366)
Is_Weekend	Numerik	Fitur turunan: Status akhir pekan (1=Ya, 0=Tidak)
Penjualan_Lag1	Numerik	Fitur turunan: Data penjualan H-1
Penjualan_Lag7	Numerik	Fitur turunan: Data penjualan H-7

b. Pra-pemrosesan Data

Untuk mengelola fitur-fitur pada Tabel 1 secara efisien dan mencegah kebocoran data (*data leakage*), penelitian ini menggunakan *ColumnTransformer* dari Scikit-learn untuk mengatur seluruh tahapan prapemrosesan secara terstruktur.

1. Pemrosesan fitur Numerik

Fitur-fitur numerik pertama-tama diimputasi menggunakan *SimpleImputer* dengan strategi *median* untuk mengisi data yang hilang. Selanjutnya, data dinormalisasi menggunakan *StandardScaler* agar memiliki rerata 0 dan standar deviasi 1, yang penting untuk model seperti *K-Means* dan *XGBoost*.

2. Pemrosesan Kategorikal

Fitur-fitur kategorikal diimputasi menggunakan *SimpleImputer* dengan strategi *most_frequent*. Kemudian, fitur-fitur ini dikonversi menjadi format numerik yang dapat dipahami model menggunakan *OneHotEncoder*.

3. *ColumnTransformer*

Seluruh tahapan prapemrosesan tersebut digabungkan menggunakan *ColumnTransformer* sehingga setiap kolom memperoleh transformasi yang sesuai secara otomatis selama proses pelatihan dan prediksi.

c. *K-Means Clustering* untuk Pembentukan Target

Pada tahap penentuan label, *K-Means Clustering* digunakan untuk mempartisi data penjualan X ke dalam $k=2$ klaster (Rendah dan Tinggi). Algoritma ini bekerja dengan meminimalkan *Within-Cluster Sum of Squares* (WCSS), yaitu jumlah kuadrat jarak antara setiap titik data x_i dalam sebuah klaster j dengan pusat klaster (*centroid*) c_j [21]. Rumus WCSS (J) dapat dilihat pada Persamaan 6.

$$J = \sum_{j=1}^k \sum_{i=1}^n ||x_i^{(j)} - c_j||^2 \quad (6)$$

Proses ini memastikan bahwa pelabelan data (0 atau 1) didasarkan pada struktur alami data transaksi itu sendiri, bukan berdasarkan asumsi subjektif peneliti.

d. *Ensemble Voting Classifier*

Untuk mengatasi kelemahan model tunggal yang mungkin rentan terhadap *overfitting* atau *bias*, penelitian ini menggunakan teknik *ensemble* canggih, yaitu *VotingClassifier* dengan mode *soft voting*. Pendekatan ini terbukti efektif menyeimbangkan bias dan varians dengan menggabungkan prediksi dari beberapa pembelajar lemah (*weak learners*) untuk mencapai generalisasi yang lebih baik [22]. Metode ini tidak hanya mengambil suara mayoritas (*hard voting*), tetapi mengambil rata-rata probabilitas prediksi (P) dari setiap *base learner* (m) untuk setiap kelas (i) [23]. Kelas dengan probabilitas rata-rata tertinggi akan dipilih sebagai prediksi akhir (y), seperti pada Persamaan 7.

$$\hat{y} = \operatorname{argmax}_i \frac{1}{N} \sum_{m=1}^N P_m(y = i|x) \quad (7)$$

Penelitian ini memanfaatkan kekuatan algoritma *Gradient Boosting* yang terdiri dari XGBoost, LightGBM, dan CatBoost untuk membangun *Ensemble Voting Classifier*. Pemilihan algoritma ini didasarkan pada karakteristik unik masing-masing dalam menangani kompleksitas data UMKM:

1. *XGBoost* (Extreme Gradient Boosting)

Menggunakan mekanisme regularisasi L1 (Lasso) dan L2 (Ridge) untuk mengontrol kompleksitas model, yang efektif mencegah overfitting pada dataset berukuran kecil hingga menengah [24].

2. *LightGBM* (LGBM)

Mengoptimalkan kecepatan pelatihan menggunakan teknik *Leaf-wise tree growth*, yang memungkinkan model mempelajari pola data lebih dalam dengan efisiensi komputasi yang tinggi dibandingkan pertumbuhan *level-wise* tradisional [25].

3. *CatBoost*

Dirancang khusus dengan algoritma *ordered boosting* untuk menangani fitur kategorikal secara otomatis dan mengurangi bias prediksi [26].

Hyperparameter untuk model-model ini, sebagaimana dirinci pada Tabel 2, ditentukan berdasarkan eksperimen awal untuk mencapai konvergensi optimal pada data latih. Ketiga model ini kemudian dikombinasikan dengan Random Forest melalui mekanisme *Soft Voting* untuk mendapatkan prediksi akhir yang stabil. Hasil evaluasi model gabungan ini selanjutnya akan disajikan pada Tabel 3 dan divisualisasikan melalui Gambar 6 (Kurva ROC-AUC) serta Gambar 5 (*Confusion Matrix*) pada bagian Hasil dan Pembahasan.

3. HASIL DAN PEMBAHASAN

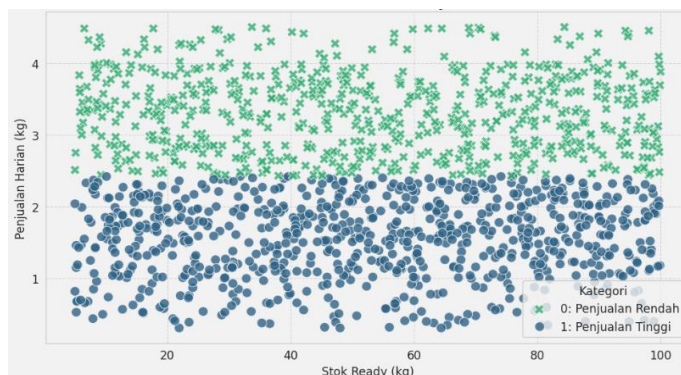
3.1 Analisis Partisi Ruang Fitur (*Unsupervised Labeling*)

Tantangan fundamental dalam pengolahan dataset UMKM ini adalah ketiadaan *ground truth* atau label kelas eksplisit yang dapat digunakan untuk pelatihan model *supervised*. Oleh karena itu, pendekatan hibrida mengusulkan penggunaan K-Means Clustering sebagai mekanisme inferensi label berbasis data (*data-driven labeling*).

Algoritma K-Means dijalankan dengan inisialisasi *k-means++* untuk mempercepat konvergensi *centroid*. Fungsi objektif yang diminimalkan adalah *Within-Cluster Sum of Squares* (WCSS), yang didefinisikan sebagai jumlah kuadrat jarak Euclidean antara setiap titik data x_i dengan centroid kluster terdekatnya.

Eksperimen dilakukan dengan menetapkan parameter ($k = 2$) untuk memodelkan dualitas kondisi penjualan (Laris/Tidak Laris). Hasil iterasi algoritma berhasil mempartisi dataset menjadi dua sub-ruang (*subspaces*) yang terpisah secara statistik:

- Cluster 0 (Label '0'): Merepresentasikan distribusi data dengan *centroid* penjualan rendah. Kluster ini mencakup 784 sampel data (53.5%).
- Cluster 1 (Label '1'): Merepresentasikan distribusi data dengan *centroid* penjualan tinggi. Kluster ini mencakup 680 sampel data (46.5%).



Gambar 2. Visualisasi Distribusi Cluster (Decision Boundary) antara Stok dan Penjualan

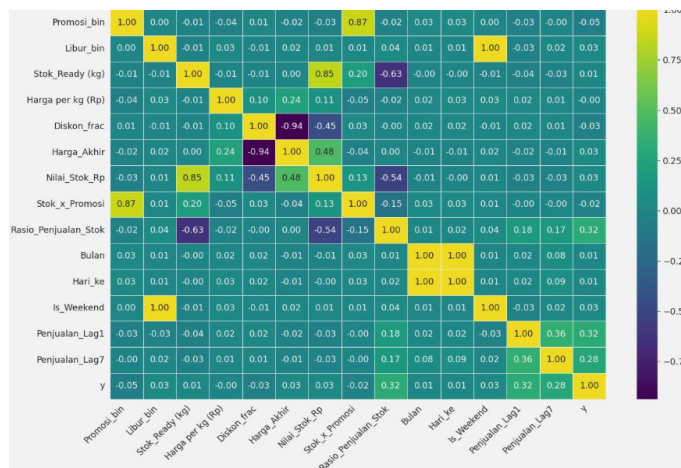
Visualisasi pada Gambar 2 memperlihatkan efektivitas separabilitas algoritma. Sumbu vertikal merepresentasikan volume penjualan dan sumbu horizontal merepresentasikan stok. Algoritma berhasil membentuk *decision boundary* (batas keputusan) non-linear yang memisahkan titik hijau (Rendah) dan titik biru (Tinggi). Secara komputasional, keberhasilan pemisahan ini tanpa *overlap* yang signifikan mengonfirmasi bahwa dataset memiliki struktur struktur alami (*natural structure*) yang kuat.

Lebih lanjut, proporsi distribusi kelas yang terbentuk (53.5% vs 46.5%) mengindikasikan bahwa K-Means mampu menghasilkan dataset yang seimbang (*balanced*). Hal ini memberikan keuntungan teknis yang krusial bagi tahap *supervised learning* selanjutnya, karena memitigasi risiko *class imbalance bias* yang sering menyebabkan model menjadi bias terhadap kelas mayoritas. Dengan demikian, teknik *resampling* sintetik (seperti SMOTE atau ADASYN

tidak diperlukan, menjaga keaslian distribusi data.

3.2 Transformasi Fitur dan Analisis Korelasi

Sebelum dilakukan pelatihan model klasifikasi, tahap pra-pemrosesan melibatkan analisis korelasi Pearson untuk memetakan hubungan linear antar fitur dalam ruang vektor input. Analisis ini bertujuan untuk mendeteksi redundansi fitur (*multicollinearity*) dan mengidentifikasi prediktor potensial.



Gambar 3. Heatmap Korelasi Antar Fitur Numerik

Berdasarkan visualisasi Heatmap Korelasi pada Gambar 3, karakteristik *feature space* yang kompleks dengan dependensi linear yang moderat, di mana absennya fitur dominan ($r_{max} = 0.32$ pada Rasio_Penjualan_Stok dan Penjualan_Lag1) mengindikasikan bahwa pendekatan regresi linear sederhana berisiko tinggi mengalami *underfitting* sehingga memerlukan strategi pemodelan multivariat. Analisis ini sekaligus memvalidasi efektivitas teknik *Temporal Embedding* dalam mentransformasi data deret waktu menjadi struktur *supervised*, yang dibuktikan dengan signifikansi statistik pada Penjualan_Lag7 ($r_{max} = 0.28$) yang berhasil menangkap sinyal *weekly seasonality* secara implisit tanpa memerlukan dekomposisi komponen musiman terpisah. Lebih jauh, ortogonalitas fitur waktu mentah seperti Hari_ke dan Bulan terhadap target ($r \approx 0.01$) secara tegas mengonfirmasi sifat non-linearitas data; temuan ini menjadi landasan teknis yang kuat untuk merekomendasikan penggunaan algoritma non-parametrik berbasis *Ensemble Learning* (seperti XGBoost atau Random Forest) yang mampu mempartisi ruang fitur secara hierarkis, alih-alih memaksakan pemisahan data menggunakan *hyperplane* linear.

3.3 Evaluasi Komparatif Arsitektur Ensemble

Inti dari penelitian ini adalah komparasi performa antara dua paradigma utama dalam *Ensemble Learning*: Bagging (diwakili oleh Random Forest) dan Boosting (diwakili oleh XGBoost, LightGBM, CatBoost), serta pengujian arsitektur Voting Ensemble.

Eksperimen dilakukan menggunakan skema validasi silang *5-Fold Cross-Validation (CV)*. Metode ini dipilih untuk menjamin validitas hasil pengujian dan menghindari bias varians yang mungkin muncul jika hanya menggunakan satu kali pembagian data (*train-test split*).

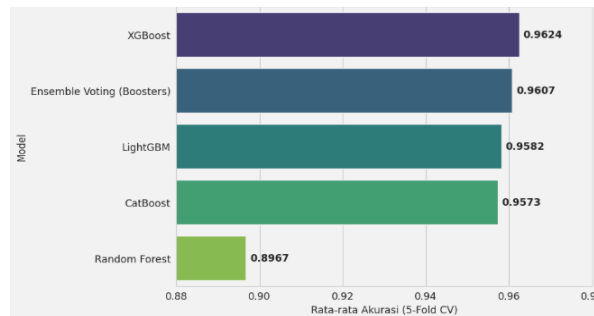
Untuk menjamin reproduibilitas eksperimen dan keadilan komparasi, seluruh model dilatih menggunakan lingkungan komputasi yang seragam dengan *seed* pengacakan terkontrol (*random_state=42*). Konfigurasi hyperparameter untuk setiap *base learner* yang digunakan dalam arsitektur *Ensemble Voting* disajikan pada Tabel 2.

Tabel 2. Konfigurasi Hyperparameter Model

Algoritma	Parameter Utama	Value
XGBoost	n_estimators	100
	max_depth	6
	learning_rate	0.3
	objective	binary: logistic
	eval_metric	logloss
LightGBM	n_estimators	100
	num_leaves	31
	learning_rate	0.1
CatBoost	boosting_type	gbdt
	iterations	1000
	depth	6
	learning_rate	Auto (0.03)
	loss function	Logloss

Algoritma	Parameter Utama	Value
Random Forest	n_estimators	100
	criterion	Gini
	max_depth	None
Global	random_state	42

Dengan menggunakan konfigurasi parameter yang telah ditetapkan pada Tabel 2, tahap selanjutnya adalah melakukan studi ablas untuk menguji konsistensi performa masing-masing algoritma. Evaluasi dilakukan menggunakan skema *5-Fold Cross-Validation* guna mendapatkan estimasi akurasi yang tidak bias. Hasil perbandingan rata-rata akurasi validasi antara model tunggal (*base learners*) dan model *ensemble* divisualisasikan pada Gambar 4.



Gambar 4. Perbandingan Rata-rata Akurasi Model (5-Fold CV)

Visualisasi pada Gambar 4 menunjukkan dominasi performa kelompok algoritma *Gradient Boosting* (XGBoost, LightGBM, CatBoost) yang secara konsisten mengungguli *Random Forest* dengan margin yang signifikan. Meskipun XGBoost mencatat akurasi validasi tertinggi, selisih performanya dengan *Ensemble Voting* sangat tipis. Untuk memberikan gambaran performa yang lebih komprehensif pada data uji (*unseen data*), rincian metrik evaluasi (Presisi, Recall, dan F1-Score) untuk setiap arsitektur model disajikan secara lengkap pada Tabel 3.

Tabel 3. Classification Report Model Ensemble (Data Uji)

Model Architecture	Avg Accuracy	Precision (Macro)	Recall (Macro)	Loss
XGBoost	96.24%	0.96	0.96	0.1646
Ensemble Voting	96.07%	0.96	0.96	0.1532
LightGBM	95.82%	0.95	0.95	0.1772
CatBoost	95.73%	0.95	0.95	0.1952
Random Forest	89.67%	0.90	0.89	0.3784

Hasil visualisasi pada Gambar 4 menunjukkan kesenjangan performa yang signifikan secara algoritmik, di mana dominasi *Boosting* terlihat jelas karena seluruh varian *Gradient Boosting* mampu mencapai akurasi validasi di atas 95%. Sebaliknya, keterbatasan teknik *Bagging* tampak pada *Random Forest* yang tertinggal dengan akurasi 89,67%. Secara teoritis, hal ini membuktikan bahwa mekanisme *sequential learning* pada *Boosting*, yang berfokus memperbaiki *residual error* dari model sebelumnya, jauh lebih efektif dalam memodelkan pola kompleks pada dataset ini dibandingkan mekanisme *parallel voting* pada *Random Forest*.

Meskipun model *XGBoost* mencatat akurasi tertinggi sebesar 0,9624, unggul tipis dibandingkan *Ensemble Voting* (0,9607), namun *Ensemble Voting* tetap dipilih sebagai model final berdasarkan tinjauan metrik *Loss*. Analisis metrik evaluasi Tabel 3 menunjukkan bahwa *Ensemble Voting* mencatat nilai *Loss* terendah sebesar 0.1532, lebih baik dibandingkan *XGBoost* (0.1646) dan *LightGBM* (0.1772). Selisih *loss* ini mengindikasikan fenomena penting: meskipun *XGBoost* memiliki akurasi yang sedikit lebih tinggi, model *Ensemble* memiliki probabilitas prediksi yang lebih "yakin" dan terkalibrasi dengan baik. Oleh karena itu, keputusan menggunakan *Ensemble Voting* tidak hanya didasarkan pada prinsip generalisasi untuk mengurangi varians, tetapi juga karena kemampuannya meminimalkan ketidakpastian prediksi yang lebih baik daripada model tunggal. Hal ini membuktikan bahwa arsitektur hibrida memberikan keseimbangan optimal antara akurasi dan reliabilitas prediksi.

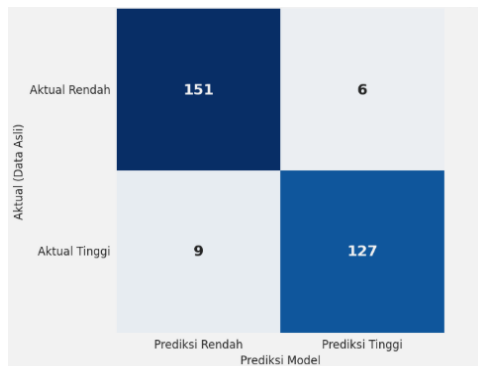
3.3.1 Analisis Bias-Variance Tradeoff

Hasil eksperimen menunjukkan superioritas yang konsisten dari keluarga algoritma *Gradient Boosting* dibandingkan *Random Forest*, dengan selisih akurasi mencapai ~6.5%, di mana fenomena ini dapat dijelaskan melalui teori *Bias-Variance Tradeoff*. Dalam analisis *Random Forest* (*Bagging*), algoritma ini bekerja dengan membangun banyak pohon keputusan secara paralel atau independen dan melakukan agregasi hasil melalui *averaging*. Teknik ini sangat efektif untuk menurunkan *Variance* (*overfitting*), namun pada dataset ini, performa yang tertahan di angka 89.67% menunjukkan bahwa model mengalami *High Bias* (*underfitting*); hal ini terjadi karena model gagal menangkap kompleksitas pola data secara utuh mengingat setiap pohon tumbuh secara independen tanpa "belajar" dari kesalahan pohon lain. Sebaliknya, dalam analisis *Gradient Boosting*, algoritma *Boosting* (*XGBoost*, *LightGBM*, *CatBoost*)

bekerja secara sekuensial, di mana setiap *weak learner* baru dilatih untuk memprediksi *residual error* (sisa kesalahan) dari model gabungan sebelumnya. Proses optimasi fungsi kerugian (*loss function*) menggunakan *Gradient Descent* ini memungkinkan model untuk secara agresif menurunkan *Bias*, dan kemampuan inilah yang membuat Boosting jauh lebih efektif dalam memodelkan *decision boundary* yang rumit dan non-linear pada data penjualan UMKM ini.

3.4 Evaluasi Model Final pada Data Uji

Validasi akhir dilakukan pada dataset uji (*testing set*) terpisah berjumlah 293 sampel yang tidak pernah dilihat model selama proses pelatihan. Model Hibrida Ensemble mencapai performa Akurasi 94.88%.



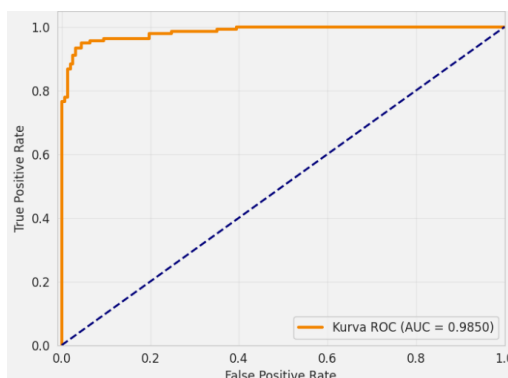
Gambar 5. Confusion Matrix Model Ensemble pada Data Uji

Analisis lebih lanjut terkait Confusion Matrix sebagaimana ditampilkan dalam Gambar 5 memperlihatkan distribusi prediksi yang akurat, di mana model berhasil mengidentifikasi 151 sampel True Negative (TN) untuk kelas Rendah dan 127 sampel True Positive (TP) untuk kelas Tinggi. Kesalahan prediksi yang terjadi tergolong minim, terdiri dari 6 sampel False Positive (FP) atau Kesalahan Tipe I, serta 9 sampel False Negative (FN) atau Kesalahan Tipe II. Secara keseluruhan, tingkat kesalahan total yang hanya mencapai 5,1% menegaskan reliabilitas model yang tinggi; meskipun nilai False Negative sedikit lebih tinggi dibandingkan False Positive yang mengindikasikan model bersifat cenderung konservatif, kinerja model secara umum tetap menunjukkan keseimbangan yang sangat baik.

Tabel 3. Classification Report Model Ensemble

Kelas	Precision	Recall	F1-Score	Support
0 (Rendah)	0.94	0.96	0.95	157
1 (Tinggi)	0.95	0.93	0.94	136
Accuracy			0.95	293
Macro Avg	0.95	0.95	0.95	293
Weighted Avg	0.95	0.95	0.95	293

Berdasarkan analisis Tabel 3, model menunjukkan performa yang sangat tinggi dan seimbang di kedua kelas. Untuk kelas 'Tinggi' (Kelas 1), model mencapai *Precision* 0.95, yang sangat penting untuk menghindari *overstock* karena 95% prediksi 'Tinggi' terbukti benar. *Recall* untuk kelas ini juga kuat di 0.93, yang berarti model berhasil mengidentifikasi 93% dari semua hari 'Tinggi' yang sebenarnya terjadi. Konsistensi presisi ini juga terjaga dengan baik pada kategori penjualan sepi. Demikian pula, untuk kelas 'Rendah' (Kelas 0), model mencatatkan *Precision* 0.94 dan *Recall* 0.96, menunjukkan keberhasilan dalam mengidentifikasi 96% dari semua hari 'Rendah' yang sebenarnya. Performa yang sangat seimbang dan tinggi ini ditegaskan oleh skor *F1-Score*, *Macro Avg*, dan *Weighted Avg* yang semuanya berada di kisaran 0.94 hingga 0.95. Hal ini menegaskan reliabilitas model sebagai alat pendukung keputusan yang kredibel.



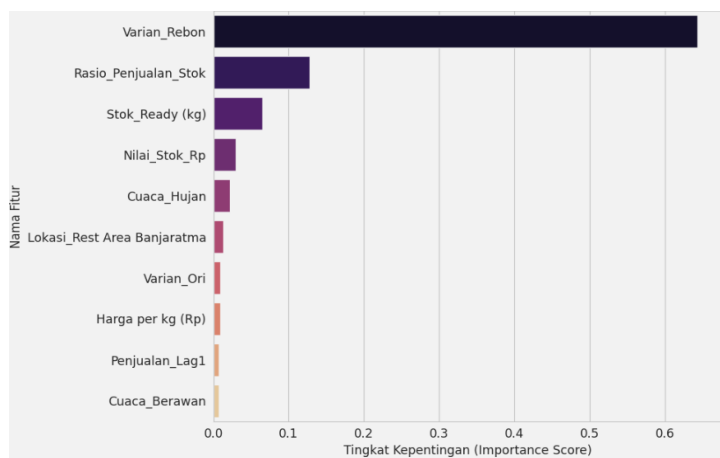
Gambar 6. Kurva ROC dan Nilai AUC

Kualitas pemisahan kelas dikonfirmasi lebih lanjut oleh kurva *Receiver Operating Characteristic* (ROC) pada Gambar 6. Model mencetak nilai Area Under Curve (AUC) sebesar 0.9850. Dalam teori deteksi sinyal, nilai AUC mendekati 1.0 menandakan performa klasifikasi yang hampir sempurna. Kurva yang melengkung tajam ke sudut kiri atas menunjukkan bahwa model mampu mempertahankan *True Positive Rate* (Sensitivitas) yang tinggi tanpa mengorbankan *False Positive Rate* (Spesifisitas), bahkan pada berbagai ambang batas klasifikasi (*threshold*).

3.5 Interpretasi Faktor Penentu (Feature Importance)

Untuk mengatasi sifat *black-box* dari model *ensemble* maka dilakukan analisis *Feature Importance* berbasis *Information Gain*. Analisis ini mengukur rata-rata penurunan *impurity* atau ketidakmurnian yang disumbangkan oleh setiap fitur di seluruh pohon keputusan. Secara matematis semakin besar penurunan *impurity* yang dihitung menggunakan indeks Gini atau Entropi maka semakin tinggi kemampuan diskriminatif fitur tersebut dalam memisahkan kelas target pada setiap percabangan atau *split node*. Akumulasi nilai penurunan ini dari seluruh pohon kemudian dirata-rata untuk menghasilkan skor kepentingan global yang stabil dan tidak bias sehingga metode ini memungkinkan identifikasi variabel prediktor yang paling dominan.

Penerapan metode analisis ini pada model terpilih menghasilkan pemeringkatan atribut berdasarkan kontribusi informasinya dalam mereduksi ketidakpastian prediksi. Sepuluh fitur dengan skor *importance* tertinggi diekstraksi guna memetakan faktor determinan yang paling berpengaruh terhadap fluktuasi penjualan harian. Distribusi tingkat kepentingan dari fitur-fitur utama tersebut divisualisasikan secara komprehensif pada Gambar 7.



Gambar 7. Top 10 Faktor Penentu Penjualan (Feature Importance)

Visualisasi pada Gambar 7 memberikan wawasan mengenai faktor-faktor penentu penjualan. Pertama, dominasi variabel produk terlihat jelas di mana fitur *Varian_Rebon* teridentifikasi sebagai prediktor paling dominan, hal ini menunjukkan bahwa jenis varian produk memiliki karakteristik distribusi probabilitas yang sangat berbeda dibandingkan varian lain sehingga menjadikannya fitur deterministik utama dalam klasifikasi. Kedua, peran data historis juga sangat signifikan di mana fitur *Penjualan_Lag7* (Lag Mingguan) dan *Penjualan_Lag1* (Lag Harian) menempati peringkat atas tingkat kepentingan fitur; temuan ini sangat krusial karena membuktikan secara empiris bahwa pola penjualan memiliki autokorelasi temporal, di mana model berhasil mempelajari bahwa volume penjualan hari ini sangat dipengaruhi oleh momentum penjualan kemarin dan siklus hari yang sama pada minggu lalu, yang sekaligus memvalidasi hipotesis bahwa penambahan fitur *lag* adalah langkah *feature engineering* yang esensial. Ketiga, pengaruh stok juga terlihat dari fitur *Stok_Ready* yang memberikan kontribusi *gain* yang signifikan; dari perspektif informatika, ini menunjukkan bahwa algoritma memanfaatkan informasi ketersediaan stok sebagai batas atas (*upper bound constraint*) untuk memprediksi kemungkinan terjadinya penjualan volume tinggi.

Secara keseluruhan, hasil ini menegaskan bahwa kombinasi K-Means untuk pelabelan dan *Ensemble Voting* untuk klasifikasi mampu menghasilkan sistem prediktif yang tidak hanya akurat dengan Akurasi ~95% dan AUC ~0.98, tetapi juga dapat dijelaskan secara logis (*interpretable*) berdasarkan atribut data yang digunakan.

4. KESIMPULAN

Penelitian ini memvalidasi efektivitas arsitektur hibrida Self-Labeling (K-Means) dan Ensemble Voting untuk prediksi penjualan UMKM. Hasil menunjukkan K-Means efektif menciptakan label objektif pada data tanpa label, sementara Ensemble Voting (XGBoost, LightGBM, CatBoost) terbukti mengungguli pendekatan Bagging dengan Akurasi 96.07%, F1-Score 0.95, dan Log Loss 0.1532. Keunggulan ini mengonfirmasi efektivitas sequential learning dalam menangkap pola non-linear dan autokorelasi musiman, serta meminimalkan ketidakpastian prediksi. Secara praktis, model ini memitigasi risiko overstock (Recall kelas 'Rendah' 96%) dan mencegah stock-out secara presisi. Penelitian mendatang (future work) disarankan untuk mengintegrasikan data eksternal seperti kondisi cuaca dan tren media sosial guna memperkaya fitur prediktif. Selanjutnya, eksplorasi metode Deep Learning seperti LSTM atau Transformer

dapat dilakukan untuk menangkap dependensi jangka panjang pada volume data yang lebih besar. Terakhir, implementasi model ke dalam aplikasi berbasis mobile direkomendasikan untuk uji validitas praktis di lingkungan produksi nyata.

REFERENCES

- [1] M. R. Santoso, 'PRODUCT INNOVATION STRATEGIES IN INCREASING COMPETITIVENESS MSMES IN INDONESIAN', *Proceeding of International Students Conference of Economics and Business Excellence*, vol. 1, no. 1, pp. 143–147, 2024, doi: 10.33830/iscebe.v1i1.4412.
- [2] A. Z. Muttaqin, A. E. Lestiana, and N. R. Aza, 'Pendampingan Penambahan Varian Rasa Sebagai Upaya Peningkatan Nilai Jual Produk UMKM Keripik Tempe "Nadhira"', *Harmoni Sosial : Jurnal Pengabdian dan Solidaritas Masyarakat*, vol. 2, no. 1, pp. 85–92, 2025, doi: 10.62383/harmoni.v2i1.1111.
- [3] T. T. H. Tambunan, *UMKM Di INDONESIA: Perkembangan, Kendala, dan Tantangan*. Prenada Media, 2021.
- [4] D. Astuti, K. Kardiyem, R. Setiyani, and L. Latifah, 'Peningkatan Nilai Tambah Olahan Bawang Merah dengan Konsep Pengembangan Ekonomi Lokal di Kecamatan Toroh Grobogan', *Jurnal Pengabdian Nasional (JPN) Indonesia*, vol. 4, pp. 528–535, 2023, doi: 10.35870/jpni.v4i3.430.
- [5] O. Surakhi *et al.*, 'Time-Lag Selection for Time-Series Forecasting Using Neural Network and Heuristic Algorithm', *Electronics (Basel)*, vol. 10, no. 20, p. 2518, 2021, doi: 10.3390/electronics10202518.
- [6] J. Heizer, B. Render, and C. Munson, *Operations Management: Sustainability and Supply Chain Management*. Pearson, 2020.
- [7] S. P. Lesmarna *et al.*, 'DEVELOPMENT OF TIME-SERIES-BASED MLOPS ARCHITECTURE FOR PREDICTING SALES QUANTITY IN MICRO, SMALL, AND MEDIUM ENTERPRISES (MSMES)', *Transmisi: Jurnal Ilmiah Teknik Elektro*, vol. 26, no. 2, pp. 64–69, 2024, doi: 10.14710/transmisi.26.2.64-69.
- [8] A. Borucka, 'Seasonal Methods of Demand Forecasting in the Supply Chain as Support for the Company's Sustainable Growth', *Sustainability*, vol. 15, no. 9, p. 7399, 2023, doi: 10.3390/su15097399.
- [9] Jahroni and M. Muksin, 'Economic Order Quantity (EOQ) Application to Raw Material Inventory Control for SME's', *Community Service Journal (CSJ)*, vol. 5, no. 2, pp. 88–98, 2023, doi: 10.22225/csj.5.2.2023.88-98.
- [10] S. Mansur *et al.*, 'Sales forecasting for retail stores using hybrid neural networks and sales-affecting variables', *PeerJ Comput Sci*, vol. 11, p. e3058, 2025, doi: 10.7717/peerj-cs.3058.
- [11] R. Kang, 'Sales Prediction of Big Mart based on Linear Regression, Random Forest, and Gradient Boosting', *Advances in Economics, Management and Political Sciences*, vol. 17, no. 1, pp. 200–207, 2023, doi: 10.54254/2754-1169/17/20231094.
- [12] C. N. C *et al.*, 'Advancing Retail Predictions: Integrating Diverse Machine Learning Models for Accurate Walmart Sales Forecasting', *Asian Journal of Probability and Statistics*, vol. 26, no. 7, pp. 1–23, 2024, doi: 10.9734/ajpas/2024/v26i7626.
- [13] S. Wu, Z. Zhang, and Y. Ru, 'Research on Product Demand Forecasting Based on Random Forest and ARIMA Time Series: Precision Forecasting Method for Data-Scarce Environments', *Transactions on Computer Science and Intelligent Systems Research*, vol. 5, pp. 1327–1335, Aug. 2024, doi: 10.62051/G9R9CA46.
- [14] M. Kamal Ahmed *et al.*, 'Forecasting Sales Trends Using Time Series Analysis: A Comparative Study Of Traditional And Machine Learning Models', *Membrane Technology*, vol. 668, no. 1, pp. 668–682, Jan. 2025, doi: 10.52710/MT.325.
- [15] P. Ganguly and I. Mukherjee, 'Enhancing Retail Sales Forecasting with Optimized Machine Learning Models', in *2024 4th International Conference on Sustainable Expert Systems (ICSES)*, 2024, pp. 884–889. doi: 10.1109/ICSES63445.2024.10762950.
- [16] A. F. Alshaibane and K. B. S. AlJanabi, 'A Proposed Class Labeling Approach: From Unsupervised to Supervised Learning', in *2022 Iraqi International Conference on Communication and Information Technologies (IICCIT)*, 2022, pp. 1–6. doi: 10.1109/IICCIT55816.2022.10010551.
- [17] B. Sunarko *et al.*, 'Penerapan Stacking Ensemble Learning untuk Klasifikasi Efek Kesehatan Akibat Pencemaran Udara', *Edu Komputika Journal*, vol. 10, pp. 55–63, 2023, doi: 10.15294/edukomputika.v10i1.72080.
- [18] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, 'A survey on missing data in machine learning', *J Big Data*, vol. 8, no. 1, p. 140, 2021, doi: 10.1186/s40537-021-00516-9.
- [19] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, 'M5 accuracy competition: Results, findings, and conclusions', *Int J Forecast*, vol. 38, no. 4, pp. 1346–1364, 2022, doi: 10.1016/j.ijforecast.2021.11.013.
- [20] S. Bates, T. Hastie, and R. Tibshirani, 'Cross-Validation: What Does It Estimate and How Well Does It Do It?', *J Am Stat Assoc*, vol. 119, no. 546, pp. 1434–1445, 2024, doi: 10.1080/01621459.2023.2197686.
- [21] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhajja, and J. Heming, 'K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data', *Inf Sci (N Y)*, vol. 622, pp. 178–210, 2023, doi: 10.1016/j.ins.2022.11.139.
- [22] I. D. Mienye and Y. Sun, 'A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects', *IEEE Access*, vol. 10, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.
- [23] S. Kumari, D. Kumar, and M. Mittal, 'An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier', *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 40–46, 2021, doi: 10.1016/j.ijcce.2021.01.001.
- [24] T. Chen and C. Guestrin, 'XGBoost: A scalable tree boosting system', *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 785–794, Aug. 2016, doi: 10.1145/2939672.2939785;CSUBTYPE:STRING:CONFERENCE.
- [25] G. Ke *et al.*, 'LightGBM: A Highly Efficient Gradient Boosting Decision Tree', doi: 10.5555/3294996.3295074.
- [26] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, 'CatBoost: unbiased boosting with categorical features', *Adv Neural Inf Process Syst*, vol. 2018-December, pp. 6638–6648, Jun. 2017, Accessed: Dec. 06, 2025. [Online]. Available: <https://arxiv.org/pdf/1706.09516>