HOAQ: JURNAL TEKNOLOGI INFORMASI, Volume 17 Nomor 2, Desember 2026, p193-202
p-ISSN: 2337-5280, e-ISSN: 26207427
DOI: https://doi.org/10.52972/hoaq.vol17no2

PENERAPAN ALGORITMA *DECISION TREE REGRESSION* DALAM PREDIKSI PENJUALAN BARANG UNTUK REKOMENDASI STOK MASUK PADA CV ABC PALEMBANG

Johannes[1], Aji Pragoyo[2], I Komang Darmawan[3], Latius Hermawan[4]

[1,2,3]Program Studi Informatika, Universitas Katolik Musi Charitas
Jl. Bangau No. 60, Palembang – Sumatera Selatan, Indonesia
Email: [1]johanneswork2024@gmail.com, [2] stagama42@gmail.com, [3]kdarmawan2004@gmail.com, [4]tiuz.hermawan@ukmc.ac.id

ABSTRAK

Pengelolaan stok barang pada toko bangunan yang tidak berbasis data historis dapat menyebabkan risiko kekurangan atau kelebihan persediaan. Penelitian ini dilakukan pada CV ABC Palembang, yaitu nama samaran dari salah satu toko bangunan di Palembang. Tahapan penelitian diawali dengan identifikasi masalah, studi literatur, pengumpulan data, pre-processing dan pembentukan target, feature engineering, pemodelan, pengujian, evaluasi, serta implementasi sistem web berbasis Django. Data yang digunakan merupakan transaksi penjualan dua produk semen, yaitu Semen Dynamix 40 KG dan Semen Merdeka 40 KG, pada periode 1 Januari 2025 sampai 30 Desember 2025. *Dataset* mentah berjumlah 243 baris, kemudian melalui proses pembersihan menjadi 241 baris transaksi valid. Target model dibentuk menggunakan variabel Jumlah_Penjualan_Berikutnya sehingga data siap model berjumlah 239 baris. Data dibagi secara berurutan berdasarkan waktu dengan rasio 80% untuk data latih dan 20% untuk data uji. Setelah dilakukan penyesuaian hyperparameter DecisionTreeRegressor, evaluasi model menghasilkan MAE sebesar 5,86 dan RMSE sebesar 7,70. Hasil tersebut menunjukkan bahwa model mampu menghasilkan estimasi awal dalam satuan jumlah penjualan, sedangkan sistem web dapat memberikan rekomendasi stok masuk berdasarkan *safety stock* dan stok akhir saat ini.

Kata kunci: *decision tree regression, machine learning*, prediksi penjualan, stok barang, django

ABSTRACT

Inventory management in a building materials store that is not supported by historical data may increase the risk of stock shortages or excess inventory. This study was conducted at CV ABC Palembang, a pseudonym for a building materials store in Palembang, to apply the Decision Tree Regression algorithm to predict product sales in the next period and generate incoming stock recommendations. The dataset consists of sales transaction data for two cement products, namely Semen Dynamix 40 KG and Semen Merdeka 40 KG. The raw dataset contained 243 rows and was cleaned into 241 valid transaction rows. The model target was constructed using the Jumlah_Penjualan_Berikutnya variable, resulting in 239 rows of modeling data. The research stages include problem identification, literature review, data collection, preprocessing, feature engineering, model training, testing, evaluation, and web-based system implementation using Django. The dataset was split sequentially based on time using an 80% training and 20% testing ratio. After adjusting the DecisionTreeRegressor hyperparameters, the model evaluation produced an MAE of 5.86 and an RMSE of 7.70. The results indicate that the model can provide an initial sales prediction in unit form, while the web system can generate incoming stock recommendations based on safety stock and current ending inventory.

Keywords: decision tree regression, machine learning, sales prediction, inventory stock, django

1. PENDAHULUAN

Pengelolaan persediaan barang merupakan salah satu aspek penting dalam kegiatan operasional penjualan, termasuk pada CV ABC Palembang yang merupakan nama samaran dari salah satu toko bangunan di Palembang. Toko bangunan tersebut memiliki aktivitas transaksi penjualan bahan bangunan, salah satunya produk semen, yang membutuhkan pengelolaan stok secara tepat agar ketersediaan barang tetap terjaga. Ketersediaan stok yang terlalu sedikit dapat menyebabkan kehilangan peluang penjualan, sedangkan stok yang terlalu banyak dapat menimbulkan penumpukan barang dan biaya penyimpanan. Permasalahan ini semakin terasa ketika pengambilan keputusan restock masih dilakukan secara manual karena keputusan sering bergantung pada perkiraan subjektif, bukan pada pola historis penjualan. Literatur *forecasting* menjelaskan bahwa prediksi permintaan berperan penting dalam proses perencanaan dan pengambilan keputusan karena membantu organisasi mengurangi ketidakpastian masa

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

depan [1]. Dalam konteks ritel, prediksi pada level produk dapat mendukung keputusan operasional seperti pemesanan ulang dan pengendalian persediaan [2].

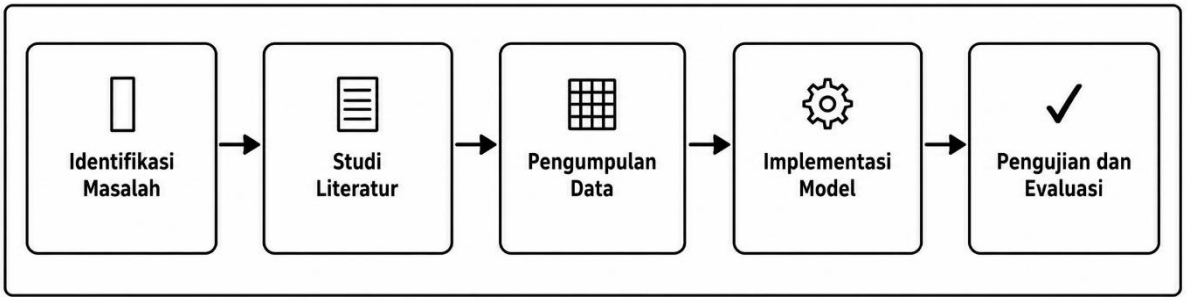
Pemanfaatan *machine learning* menjadi salah satu pendekatan yang dapat digunakan untuk mempelajari pola penjualan dari data historis. Penelitian sebelumnya menunjukkan bahwa metode pembelajaran mesin telah digunakan dalam berbagai kasus *demand forecasting* dan *inventory forecasting*, baik menggunakan model *tree-based, ensemble learning*, maupun *deep learning* [3], [4]. Beberapa penelitian di Indonesia juga mulai menerapkan model *machine learning* untuk prediksi stok dan penjualan, seperti forecasting persediaan produk barang pokok, optimalisasi *overstock* dan *stockout* menggunakan *Random Forest*, dan prediksi penjualan produk tertentu [5]-[7]. Kajian terbaru juga menegaskan bahwa penerapan *machine learning* pada demand *forecasting* dan *inventory control* sangat dipengaruhi oleh kualitas data historis, pemilihan fitur, dan karakteristik permintaan [8], [9]. Hal ini menunjukkan bahwa pengolahan data transaksi penjualan dapat dimanfaatkan sebagai dasar dalam membangun sistem pendukung keputusan stok.

Penelitian ini berfokus pada prediksi jumlah penjualan barang pada periode berikutnya menggunakan algoritma *Decision Tree Regression*. Pemilihan target berupa jumlah penjualan periode berikutnya dilakukan untuk menghindari kebocoran data (*data leakage*) yang dapat terjadi apabila target dibuat dari status stok saat ini, sementara status tersebut sebenarnya dapat dihitung secara langsung dari stok awal, stok masuk, dan jumlah penjualan. Dengan memprediksi jumlah penjualan periode berikutnya, sistem tidak hanya memberikan label kondisi stok, tetapi juga membantu penjual memperkirakan kebutuhan restock ke supplier secara lebih praktis.

Berdasarkan permasalahan tersebut, penelitian ini diarahkan untuk menerapkan algoritma *Decision Tree Regression* dalam memprediksi jumlah penjualan barang pada periode berikutnya, mengevaluasi kinerja model menggunakan metrik regresi yang sesuai, serta mengimplementasikan hasil prediksi ke dalam sistem web berbasis Django. Melalui penelitian ini, diharapkan sistem yang dibangun dapat menjadi prototipe alat bantu bagi CV ABC Palembang dalam memperkirakan kebutuhan penjualan dan menyusun rekomendasi stok masuk berdasarkan data transaksi historis.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimen berbasis machine learning. Tahapan penelitian dilakukan secara berurutan, dimulai dari identifikasi masalah, studi literatur, pengumpulan data, *pre-processing* dan pembentukan target, *feature engineering*, analisis *feature importance*, skenario pengujian, pemodelan, evaluasi model, serta implementasi sistem web. Tahapan tersebut disusun agar proses penelitian dapat menggambarkan alur penyelesaian masalah secara runtut, mulai dari pemahaman kebutuhan studi kasus hingga penerapan model ke dalam sistem web. Alur penelitian disajikan pada Gambar 1.



Gambar 1. Alur Penelitian

Identifikasi Masalah

Tahap identifikasi masalah dilakukan untuk merumuskan kebutuhan sistem pada CV ABC Palembang, yaitu membantu penjual memperkirakan penjualan barang pada periode berikutnya dan memberikan rekomendasi stok masuk. Permasalahan utama yang diangkat adalah belum adanya alat bantu berbasis data yang dapat mengolah pola transaksi penjualan menjadi informasi prediktif untuk pengambilan keputusan restock.

Studi Literatur

Studi literatur dilakukan dengan menelaah penelitian terkait *demand forecasting, inventory forecasting, machine learning*, dan *Decision Tree Regression. Forecasting* secara umum digunakan untuk memperkirakan nilai masa depan berdasarkan data masa lalu [1]. Pada sektor ritel, forecasting dapat membantu perencanaan persediaan pada level produk dan toko [2]. Model berbasis pohon keputusan memiliki kemampuan membentuk aturan nonlinier dari data dan dapat digunakan pada tugas regresi untuk memprediksi nilai numerik [10], [11].

Pengumpulan data

Data yang digunakan merupakan dataset transaksi penjualan barang pada CV ABC Palembang. Nama CV ABC digunakan sebagai nama samaran untuk menjaga kerahasiaan identitas tempat studi kasus. Data transaksi mencakup periode 1 Januari 2025 sampai 30 Desember 2025. Atribut awal yang tersedia meliputi tanggal, jenis produk, jumlah penjualan, stok awal, stok masuk, dan stok akhir. Produk yang terdapat pada dataset adalah Semen

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Dynamix 40 KG dan Semen Merdeka 40 KG. *Dataset* mentah berjumlah 243 baris. Setelah dilakukan pembersihan data, diperoleh 241 baris transaksi valid. Dua baris data dihapus karena merupakan baris non-transaksi/rekap pada *file dataset*, sehingga tidak merepresentasikan transaksi penjualan dan tidak dapat digunakan pada proses pemodelan. Pada tahap pembentukan target periode berikutnya, satu baris terakhir dari masing-masing produk tidak memiliki target lanjutan sehingga data siap model menjadi 239 baris. Sehingga data siap model menjadi 239 baris.

Deskripsi atribut data yang digunakan dalam penelitian disajikan pada Tabel 1.

Tabel 1. Deskripsi Atribut Data

Atribut	Deskripsi
Jumlah_Penjualan	Jumlah produk yang terjual pada periode transaksi
Tanggal	Tanggal terjadinya transaksi penjualan barang
Jenis_Produk	Nama produk yang dijual, yaitu Semen Dynamix 40 KG dan Semen Merdeka 40 KG
Stok_Awal	Jumlah stok produk sebelum transaksi pada periode tersebut
Stok_Masuk	Jumlah stok tambahan yang masuk pada periode transaksi. Nilai kosong diisi 0 karena tidak semua transaksi memiliki stok masuk
Stok_Akhir	Jumlah stok setelah transaksi berlangsung berdasarkan data sumber
Stok_Akhir_Hitung	Validasi stok akhir yang dihitung kembali dari Stok_Awal + Stok_Masuk - Jumlah_Penjualan
Jumlah_Penjualan_Berikutnya	Target model yang dibentuk dari jumlah penjualan pada periode berikutnya untuk produk yang sama

Tabel 2. Ringkasan Dataset

Keterangan	Jumlah / Isi
Jumlah data mentah	243 baris
Jumlah data bersih	241 baris
Jumlah data model	239 baris
Jumlah produk	2 produk
Target model	Jumlah_Penjualan_Berikutnya
Algoritma	DecisionTreeRegressor

***Pre-processing* dan pembentukan target**

Tahap *pre-processing* dilakukan untuk memastikan data transaksi siap digunakan dalam proses pemodelan. Tahap pertama adalah menghapus kolom atau baris yang tidak merepresentasikan transaksi penjualan. Tahap kedua adalah mengonversi kolom Tanggal ke format *datetime* agar dapat digunakan untuk membentuk fitur waktu seperti bulan, hari, hari dalam minggu, minggu ke-, dan kuartal. Tahap ketiga adalah mengonversi atribut numerik seperti Stok_Awal, Stok_Masuk, Jumlah_Penjualan, dan Stok_Akhir ke tipe data numerik agar dapat diproses oleh model. Nilai kosong pada atribut Stok_Masuk diisi dengan 0 karena tidak semua transaksi memiliki penambahan stok masuk. Tahap keempat adalah memvalidasi nilai stok akhir dengan menghitung kembali stok akhir berdasarkan Stok_Awal, Stok_Masuk, dan Jumlah_Penjualan. Dampak dari tahap *pre-processing* adalah berkurangnya jumlah data dari 243 baris menjadi 241 baris transaksi valid. Selain itu, data yang semula belum sepenuhnya siap digunakan dalam model menjadi data numerik dan terstruktur, sehingga dapat digunakan untuk proses feature engineering dan pelatihan model. Stok akhir dihitung menggunakan persamaan (1).

$$Stok_Akhir_t = Stok_Awal_t + Stok_Masuk_t - Jumlah_Penjualan_t \quad \ldots (1)$$

Keterangan: t menunjukkan periode saat ini, Stok_Akhir_t adalah jumlah stok setelah transaksi pada periode t, Stok_Awal_t adalah stok sebelum transaksi, Stok_Masuk_t adalah jumlah barang yang masuk, dan Jumlah_Penjualan_t adalah jumlah barang yang terjual pada periode yang sama.

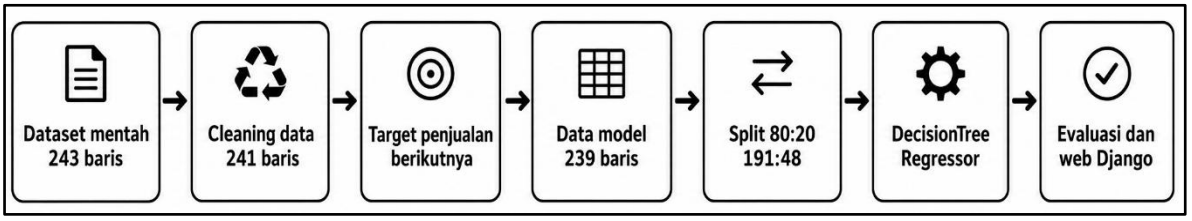
Target model dibentuk menggunakan data penjualan pada periode berikutnya untuk produk yang sama, sebagaimana ditunjukkan pada persamaan (2).

$$Jumlah_Penjualan_Berikutnya = Jumlah_Penjualan_(t+1) \quad \ldots (2)$$

Keterangan: Jumlah_Penjualan_(t+1) adalah jumlah penjualan pada periode berikutnya untuk produk yang sama. Target ini digunakan agar model tidak memprediksi nilai yang dapat dihitung langsung dari variabel pada periode yang sama.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Alur pembentukan data model dari dataset mentah sampai siap digunakan dalam sistem web ditunjukkan pada Gambar 2.



Gambar 2. Alur *Pre-processing* dan Pemodelan

Feature engineering

Feature engineering dilakukan untuk membentuk fitur yang dapat membantu model mempelajari pola transaksi. Fitur yang digunakan terdiri dari fitur waktu, fitur stok, fitur penjualan saat ini, serta fitur historis seperti lag dan rolling average. Fitur yang digunakan sebagian besar berasal dari data transaksi dan stok yang saling berkaitan. Oleh karena itu, proses feature engineering dilakukan dengan memperhatikan urutan waktu agar tidak terjadi kebocoran data. Fitur seperti Stok_Akhir_Hitung digunakan untuk merepresentasikan kondisi stok pada periode berjalan, sedangkan Penjualan_Lag_1, Penjualan_Rolling_3, Penjualan_Rolling_5, dan Stok_Akhir_Lag_1 dibentuk dari data periode sebelumnya untuk produk yang sama. Dengan demikian, fitur historis yang digunakan dalam model hanya berasal dari data saat ini atau data sebelumnya, bukan dari data setelah target prediksi. Fitur yang secara langsung merepresentasikan nilai target pada periode berikutnya tidak digunakan sebagai input model.

Tabel 3. Penerapan *Feature engineering*

Fitur	Deskripsi
Bulan, Hari, Hari_Dalam_Minggu, Minggu_Ke, Kuartal	Fitur waktu yang menggambarkan informasi kalender transaksi.
Akhir_Pekan	Penanda apakah transaksi berada pada hari Sabtu atau Minggu.
Stok_Awal, Stok_Masuk, Jumlah_Penjualan	Fitur utama kondisi stok dan transaksi pada periode saat ini.
Stok_Akhir_Hitung	Hasil perhitungan stok akhir saat ini berdasarkan rumus stok.
Penjualan_Lag_1	Jumlah penjualan pada transaksi sebelumnya untuk produk yang sama.
Penjualan_*Rolling*_3 dan Penjualan_*Rolling*_5	Rata-rata penjualan beberapa transaksi terakhir untuk menangkap kecenderungan historis.
Stok_Akhir_Lag_1	Stok akhir pada transaksi sebelumnya untuk produk yang sama.

Analisis *feature importance* dilakukan untuk mengetahui kontribusi relatif setiap fitur terhadap hasil prediksi model Dec*isionTreeRegressor*. Nilai *feature importance* diperoleh dari atribut *feature_importances_* pada model scikit-learn. Nilai tersebut dihitung berdasarkan total penurunan impurity atau error yang diberikan oleh suatu fitur ketika digunakan sebagai pemisah pada node pohon keputusan. Semakin besar nilai *feature importance*, semakin besar kontribusi fitur tersebut dalam proses pembentukan keputusan model. Analisis ini digunakan sebagai alat interpretasi model dan sebagai dasar penyusunan skenario pengujian fitur secara incremental, bukan sebagai satu-satunya dasar penghapusan fitur.

Skenario pengujian

Pengujian model dilakukan menggunakan pembagian data secara berurutan berdasarkan waktu dengan rasio 80% data awal sebagai data latih dan 20% data akhir sebagai data uji. Pembagian berurutan dipilih karena target yang digunakan adalah Jumlah_Penjualan_Berikutnya, sehingga urutan data perlu dipertahankan agar model tidak memperoleh informasi dari periode setelahnya pada proses pelatihan. Dengan pendekatan ini, model dilatih menggunakan data transaksi pada periode sebelumnya dan diuji pada data yang berada pada urutan waktu berikutnya. Skenario pengujian ini bertujuan untuk menilai kemampuan model dalam melakukan prediksi terhadap data baru yang belum digunakan pada proses pelatihan.

Modeling

Model yang digunakan adalah *Decision Tree Regression* dengan implementasi menggunakan *library scikit-learn* [12]. *Decision Tree Regressor* memecah data berdasarkan fitur tertentu untuk menghasilkan prediksi numerik pada node daun. Pada revisi ini, hyperparameter model disesuaikan untuk mengurangi kompleksitas pohon dan menurunkan tingkat kesalahan prediksi. Hyperparameter utama yang digunakan adalah *criterion squared_error*,

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

max_depth sebesar 3, min_samples_split sebesar 2, min_samples_leaf sebesar 10, dan random_state sebesar 42. Nilai random_state digunakan agar hasil pelatihan dapat direproduksi.

Tabel 4. *Hyperparameter Model DecisionTreeRegressor*

Hyparameter	Nilai	Keterangan
criterion	*squared_error*	Mengukur kualitas pemisahan node berdasarkan pengurangan varians.
max_depth	3	Membatasi kedalaman pohon agar tidak terlalu kompleks dan mengurangi overfitting.
min_samples_split	2	Jumlah minimum sampel untuk memecah node.
min_samples_leaf	10	Jumlah minimum sampel pada setiap daun pohon untuk membantu generalisasi.
random_state	42	Menjaga hasil pelatihan tetap konsisten.

Pengujian dan evaluasi

Data dibagi secara berurutan berdasarkan waktu dengan komposisi 80% data awal sebagai data latih dan 20% data akhir sebagai data uji. Pembagian berurutan digunakan agar model tidak menggunakan informasi dari masa depan pada proses pelatihan. Evaluasi difokuskan pada *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE) karena keduanya menunjukkan besar kesalahan prediksi dalam satuan jumlah penjualan sehingga lebih mudah diinterpretasikan pada kasus stok barang. MAE dan RMSE merupakan metrik umum dalam evaluasi model regresi, bukan kontribusi baru dari penelitian ini. Kedua metrik tersebut digunakan untuk mengukur selisih antara nilai aktual dan nilai prediksi [10], [12]. Kontribusi penelitian ini terletak pada penerapan DecisionTreeRegressor untuk prediksi jumlah penjualan periode berikutnya dan integrasinya ke dalam sistem rekomendasi stok masuk.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}| \quad \dots (3)$$

Keterangan: n adalah jumlah data pengujian, y_i adalah nilai aktual jumlah penjualan berikutnya, dan ŷ_i adalah nilai hasil prediksi model. MAE menunjukkan rata-rata selisih absolut antara nilai aktual dan nilai prediksi.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \quad \dots (4)$$

Keterangan: RMSE menunjukkan akar rata-rata kesalahan kuadrat antara nilai aktual dan nilai prediksi. Nilai RMSE memberikan penalti lebih besar terhadap kesalahan prediksi yang tinggi.

Rekomendasi stok masuk

Setelah model menghasilkan prediksi jumlah penjualan periode berikutnya, sistem menghitung rekomendasi stok masuk menggunakan persamaan (5). *Safety stock* digunakan sebagai stok pengaman agar sistem tidak hanya memenuhi kebutuhan prediksi, tetapi juga memberikan cadangan minimal.

Rekomendasi_Stok_Masuk = max(0, ceil(Prediksi_Penjualan_Berikutnya + Safety_Stock - Stok_Akhir_Saat_Ini)) … (5))

Keterangan: Rekomendasi_Stok_Masuk adalah jumlah stok yang disarankan untuk ditambahkan, Prediksi_Penjualan_Berikutnya adalah hasil prediksi model, Safety_Stock adalah stok pengaman, dan Stok_Akhir_Saat_Ini adalah stok yang tersedia setelah transaksi pada periode berjalan.

3. HASIL DAN PEMBAHASAN

Hasil *pre-processing data*

Berdasarkan hasil pengolahan data transaksi CV ABC Palembang, dataset mentah berjumlah 243 baris dan 7 kolom. Kolom kosong tidak digunakan, dua baris tidak valid di bagian akhir dataset dihapus, dan nilai kosong pada Stok_Masuk diisi dengan 0. Setelah proses tersebut, diperoleh 241 baris transaksi bersih. Selanjutnya, target Jumlah_Penjualan_Berikutnya dibentuk dengan fungsi shift berdasarkan kelompok jenis produk. Karena terdapat dua produk, baris terakhir dari masing-masing produk tidak memiliki target periode berikutnya sehingga data model menjadi 239 baris.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Tabel 5. Hasil Pre-processing Data

Tahap	Jumlah Baris	Keterangan
Data mentah	243	Data awal dari dataset transaksi.
Data bersih	241	Baris kosong dan kolom tidak relevan dibuang.
Data model	239	Target Jumlah_Penjualan_Berikutnya berhasil dibentuk.
Data latih	191	80% data awal berdasarkan urutan waktu.
Data uji	48	20% data akhir berdasarkan urutan waktu.

Distribusi data dan target

Status stok saat ini tetap dihitung menggunakan aturan berbasis stok akhir, tetapi tidak digunakan sebagai target utama model. Status ini hanya digunakan untuk memberikan informasi kondisi stok kepada pengguna sistem. Status stok ditentukan berdasarkan nilai Stok_Akhir_Hitung. Apabila Stok_Akhir_Hitung kurang dari atau sama dengan 0, status stok dikategorikan sebagai HABIS. Apabila Stok_Akhir_Hitung lebih dari 0 dan kurang dari atau sama dengan 25, status stok dikategorikan sebagai KRITIS. Apabila Stok_Akhir_Hitung lebih dari 25 dan kurang dari atau sama dengan 50, status stok dikategorikan sebagai AMAN. Apabila Stok_Akhir_Hitung lebih dari 50, status stok dikategorikan sebagai BERLEBIH.

Tabel 6. Aturan Penentuan Status Stok Saat Ini

Kondisi Stok Akhir	Status Stok
Stok_Akhir <= 0	HABIS
0 < Stok_Akhir <= 25	KRITIS
25 < Stok_Akhir <= 50	AMAN
Stok_Akhir > 50	BERLEBIH

Tabel 7. Distribusi Status Stok Saat Ini

Status Stok	Jumlah Data
BERLEBIH	153
KRITIS	42
AMAN	32
HABIS	14

Hasil Evaluasi Model

Model *DecisionTreeRegressor* dilatih menggunakan 191 baris data latih dan diuji menggunakan 48 baris data uji. Setelah penyesuaian *hyperparameter*, hasil evaluasi menunjukkan nilai MAE sebesar 5,86 dan RMSE sebesar 7,70. Nilai MAE sebesar 5,86 berarti bahwa rata-rata selisih absolut antara hasil prediksi dan data aktual berada pada kisaran 5 sampai 6 unit penjualan. Sementara itu, nilai RMSE sebesar 7,70 menunjukkan bahwa ketika terdapat kesalahan prediksi yang lebih besar pada beberapa data uji, kesalahan tersebut diberikan penalti lebih tinggi. Agar nilai tersebut dapat dibaca secara lebih proporsional, penelitian ini menambahkan ringkasan statistik data uji dan perbandingan model awal dengan model hasil penyesuaian hyperparameter.

Tabel 8. Hasil Evaluasi Model

Metrik	Nilai	Interpretasi
MAE	5.86	Rata-rata kesalahan absolut sekitar 5 sampai 6 unit penjualan.
RMSE	7.70	Kesalahan prediksi yang lebih besar diberi penalti lebih tinggi; nilai ini menunjukkan tingkat error keseluruhan sekitar 7 sampai 8 unit.

Berdasarkan hasil evaluasi pada Tabel 8, nilai MAE dan RMSE digunakan sebagai metrik utama karena keduanya lebih langsung menunjukkan besar kesalahan prediksi dalam satuan jumlah penjualan. Pada data transaksi CV ABC Palembang, beberapa nilai penjualan aktual relatif kecil dan fluktuatif sehingga metrik persentase seperti MAPE dapat menjadi kurang stabil. Oleh karena itu, pembahasan evaluasi difokuskan pada MAE dan RMSE agar interpretasi hasil lebih sesuai dengan kebutuhan sistem rekomendasi stok. Hasil ini menunjukkan bahwa model telah mampu memberikan estimasi awal terhadap jumlah penjualan periode berikutnya, meskipun peningkatan data historis dan pengujian algoritma pembanding tetap diperlukan untuk meningkatkan akurasi.

Berdasarkan ringkasan statistik data uji, nilai MAE sebesar 5,86 perlu dibaca dalam konteks sebaran jumlah penjualan aktual yang berada pada rentang 1 sampai 35 unit. Perbandingan model awal dan model revisi menunjukkan bahwa penyesuaian hyperparameter menurunkan MAE dari 9,50 menjadi 5,86 dan RMSE dari 12,89 menjadi 7,70. Dengan demikian, model revisi memberikan error yang lebih rendah, meskipun masih perlu pengembangan karena pola penjualan aktual masih fluktuatif.

Perbandingan hasil evaluasi model awal dan model setelah penyesuaian *hyperparameter*.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Tabel 9. Perbandingan hasil evaluasi model

Skenario Model	MAE	RMSE	Keterangan
Model awal	9,50	12,89	Sebelum penyesuaian hyperparameter
Model revisi	5,86	7,70	Setelah max_depth dibatasi 3 dan min_samples_leaf ditetapkan 10

Ringkasan statistik data uji untuk membaca konteks nilai *error*.

Tabel 10. Ringkasan Statistik Data Uji

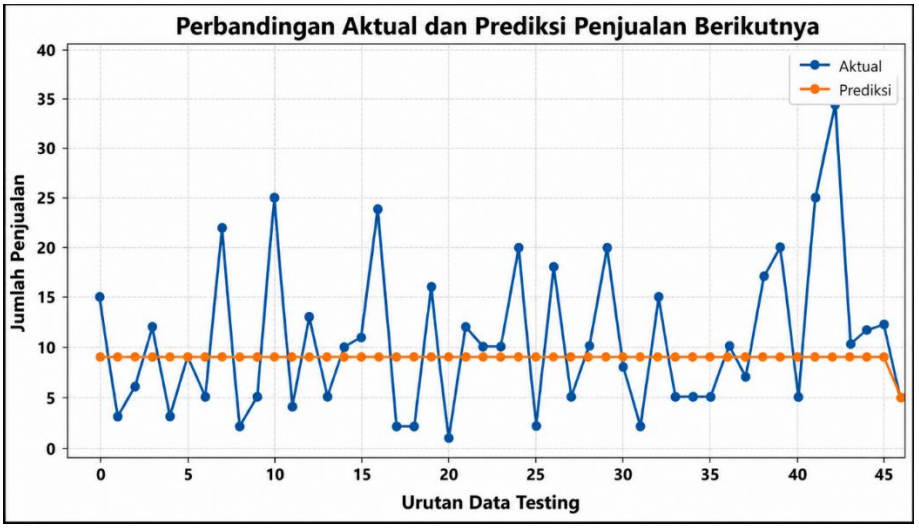
Statistik Data Uji	Nilai
Jumlah data uji	48
Minimum penjualan aktual	1 unit
Maksimum penjualan aktual	35 unit
Rata-rata penjualan aktual	10,52 unit
Median penjualan aktual	10 unit
MAE model revisi	5,86
RMSE model revisi	7,70

Untuk menanggapi pola prediksi yang cenderung datar, dilakukan pengujian tambahan secara *incremental*. Pengujian dilakukan dengan membandingkan performa model berdasarkan jumlah fitur yang digunakan, mulai dari fitur dengan nilai *importance* tertinggi, kombinasi beberapa fitur dominan, hingga seluruh fitur terpilih. Pengujian ini bertujuan untuk melihat apakah model cukup responsif terhadap perubahan fitur input dan untuk memastikan bahwa prediksi tidak hanya mengikuti nilai rata-rata data latih. Hasil pengujian *incremental* menunjukkan bahwa penambahan fitur dominan dapat menurunkan nilai *error* dari skenario satu fitur ke skenario lima fitur, tetapi pola aktual yang fluktuatif tetap belum sepenuhnya dapat diikuti oleh model pohon tunggal. Oleh karena itu, model digunakan sebagai prototipe awal dan pengembangan selanjutnya disarankan menambah periode data serta membandingkan algoritma lain seperti *Random Forest Regression*, *Gradient Boosting*, atau XGBoost.

Hasil pengujian incremental fitur dan variasi split data.

Tabel 11. Hasil pengujian incremental fitur dan variasi split data

Skenario	Jumlah Fitur	Split	MAE	RMSE	Keterangan
S1	1 fitur	80:20	6,44	8,40	Fitur Penjualan_Rolling_3 saja
S2	2 fitur	80:20	6,22	8,32	Penjualan_Rolling_3 dan Minggu_Ke
S3	3 fitur	80:20	6,00	7,76	Menambahkan fitur jenis produk
S4	5 fitur	80:20	5,86	7,70	Lima fitur dominan; digunakan sebagai model revisi
S5	Semua fitur	80:20	5,86	7,70	Performa sama dengan lima fitur dominan
S6	Semua fitur	85:15	5,79	7,78	MAE sedikit lebih rendah, tetapi RMSE lebih tinggi
S7	Semua fitur	90:10	6,17	8,38	Data uji lebih sedikit dan error meningkat

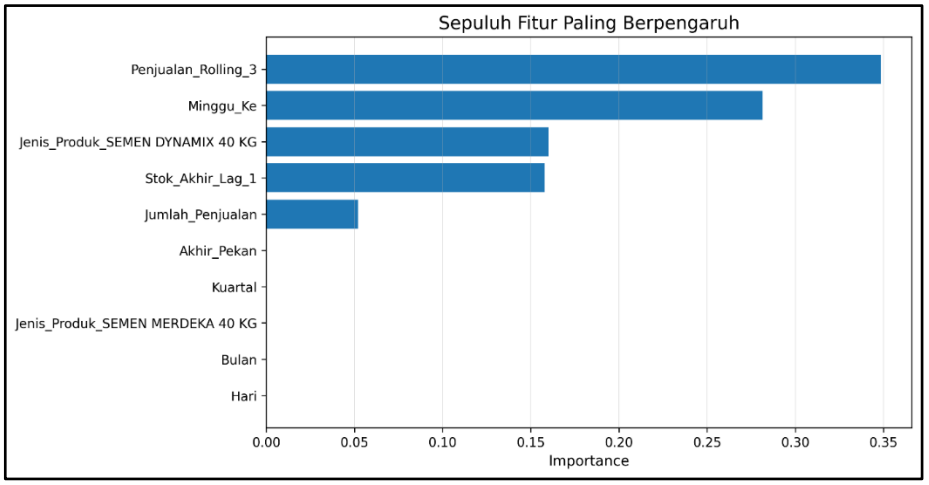


Gambar 3. Grafik Aktual dan Prediksi Jumlah Penjualan Berikutnya

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Feature importance

Analisis *feature importance* pada bagian hasil digunakan untuk menafsirkan fitur yang paling banyak berkontribusi terhadap keputusan model. Nilai importance diperoleh dari atribut feature_importances_ milik DecisionTreeRegressor pada scikit-learn, yaitu berdasarkan penurunan impurity/error ketika fitur digunakan sebagai pemisah pada pohon keputusan. Berdasarkan hasil model setelah penyesuaian hyperparameter, fitur Penjualan_Rolling_3, Minggu_Ke, Jenis_Produk, Stok_Akhir_Lag_1, dan Jumlah_Penjualan menjadi fitur yang berpengaruh terhadap hasil prediksi. Hal ini menunjukkan bahwa pola rata-rata penjualan sebelumnya, waktu transaksi, jenis produk, serta kondisi stok historis berkontribusi dalam memprediksi jumlah penjualan periode berikutnya. Fitur dengan nilai importance 0 tidak berarti data tersebut tidak ada, tetapi menunjukkan bahwa fitur tersebut tidak digunakan sebagai node pemisah pada pohon akhir.



Gambar 4. *Feature Importance DecisionTreeRegressor*

Implementasi Sistem Web

Model hasil pelatihan disimpan dalam format joblib dan diimplementasikan ke dalam sistem web berbasis Django. Implementasi sistem prediksi stok berbasis web sejalan dengan penelitian sebelumnya yang menerapkan model *machine learning* untuk prediksi stok berbasis web dan membandingkan performa beberapa model *machine learning* pada *demand forecasting* [13], [14]. Django dipilih karena merupakan *framework* web Python yang mendukung pengembangan aplikasi *server-side* dan dapat diintegrasikan dengan model *machine learning* berbasis Python [15]. Sistem menyediakan form input yang terdiri dari tanggal, jenis produk, stok awal, stok masuk, jumlah penjualan, dan *safety stock*. Setelah pengguna mengisi data, sistem menghitung stok akhir saat ini, menentukan status stok saat ini, memprediksi jumlah penjualan periode berikutnya, menghitung rekomendasi stok masuk, dan menampilkan risiko stok periode berikutnya.

Alur kerja sistem web dimulai ketika pengguna mengisi form input yang terdiri dari tanggal transaksi, jenis produk, stok awal, stok masuk, jumlah penjualan, dan *safety stock*. Setelah data dimasukkan, sistem terlebih dahulu menghitung stok akhir saat ini menggunakan rumus stok awal ditambah stok masuk dikurangi jumlah penjualan. Nilai stok akhir tersebut digunakan untuk menentukan status stok saat ini, seperti HABIS, KRITIS, AMAN, atau BERLEBIH. Selanjutnya, data input diproses oleh model D*ecisionTreeRegressor* yang telah disimpan dalam format joblib untuk menghasilkan prediksi jumlah penjualan pada periode berikutnya. Hasil prediksi tersebut kemudian digunakan untuk menghitung rekomendasi stok masuk berdasarkan prediksi penjualan berikutnya, *safety stock*, dan stok akhir saat ini. Dengan alur tersebut, sistem web tidak hanya menampilkan hasil prediksi, tetapi juga memberikan rekomendasi operasional yang dapat membantu pengguna dalam menentukan kebutuhan *restock* ke *supplier*.

Tabel 12. *Output* Sistem Web

Output	Keterangan
Stok Akhir Saat Ini	Hasil perhitungan stok awal + stok masuk - jumlah penjualan
Status Stok Saat Ini	Label kondisi stok berdasarkan nilai stok akhir
Prediksi Penjualan Berikutnya	Hasil prediksi model D*ecisionTreeRegressor*
Rekomendasi Stok Masuk	Jumlah stok yang disarankan untuk diminta ke *supplier*
Risiko Stok Berikutnya	Indikasi risiko stok apabila tidak dilakukan *restock*

Dengan mekanisme tersebut, sistem tidak menjadikan status stok saat ini sebagai target prediksi, sehingga mengurangi risiko *data leakage*. Status stok saat ini tetap dihitung secara matematis, sedangkan model *machine learning* berfokus pada prediksi jumlah penjualan periode berikutnya. Pendekatan ini lebih sesuai dengan kebutuhan bisnis karena pengguna tidak hanya mengetahui kondisi stok, tetapi juga memperoleh estimasi jumlah permintaan dan rekomendasi *restock*.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Pembahasan Hasil

Hasil evaluasi menunjukkan bahwa penerapan *DecisionTreeRegressor* pada *dataset* ini sudah dapat menghasilkan prediksi awal jumlah penjualan periode berikutnya, tetapi masih memiliki keterbatasan. Nilai MAE sebesar 5,86 dan RMSE sebesar 7,70 menunjukkan bahwa masih terdapat selisih antara hasil prediksi dan nilai aktual. Pengujian incremental memperlihatkan bahwa model memberi respons ketika fitur dominan ditambahkan, tetapi pola prediksi masih belum sepenuhnya mengikuti fluktuasi aktual pada seluruh data uji. Penyebab utama kondisi ini diperkirakan berasal dari jumlah data yang relatif kecil, tanggal transaksi yang tidak sepenuhnya berurutan harian, jumlah produk yang hanya dua jenis, dan fluktuasi penjualan yang cukup tinggi. Selain itu, model *Decision Tree* tunggal cenderung sensitif terhadap variasi data dan dapat menghasilkan generalisasi yang kurang stabil jika data latih terbatas. Temuan ini sejalan dengan penelitian yang menyatakan bahwa model forecasting sangat bergantung pada kualitas data, pemilihan fitur, dan karakteristik permintaan [1], [2], [8], [9].

Meskipun performa prediksi masih dapat ditingkatkan, penelitian ini memberikan kontribusi dalam bentuk prototipe sistem yang menghubungkan proses machine learning dengan rekomendasi stok masuk pada studi kasus CV ABC Palembang. Sistem ini dapat dikembangkan lebih lanjut dengan menambah data historis, memperbaiki granularitas data menjadi harian yang konsisten, menambahkan variabel eksternal seperti hari libur dan promosi, serta membandingkan *Decision Tree Regression* dengan *Random Forest Regression*, *Gradient Boosting*, XGBoost, atau LSTM. Penambahan data dan model pembanding diharapkan dapat meningkatkan akurasi prediksi dan kemampuan generalisasi model.

4. SIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa algoritma *Decision Tree Regression* berhasil diterapkan untuk membangun sistem prediksi jumlah penjualan barang pada periode berikutnya dan rekomendasi stok masuk berbasis web pada studi kasus CV ABC Palembang. Dataset mentah berjumlah 243 baris berhasil dibersihkan menjadi 241 baris transaksi valid dan dibentuk menjadi 239 baris data model dengan target Jumlah_Penjualan_Berikutnya. Sistem yang dibangun mampu menghitung stok akhir saat ini, menentukan status stok saat ini, memprediksi penjualan periode berikutnya, serta memberikan rekomendasi stok masuk berdasarkan *safety stock*.

Hasil evaluasi model setelah penyesuaian hyperparameter memperoleh nilai MAE sebesar 5,86 dan RMSE sebesar 7,70. Nilai tersebut menunjukkan bahwa model masih memiliki kesalahan prediksi, tetapi dapat digunakan sebagai prototipe awal untuk mengintegrasikan proses prediksi dan rekomendasi stok. Pengujian incremental fitur menunjukkan bahwa beberapa fitur dominan, terutama Penjualan_Rolling_3, Minggu_Ke, jenis produk, Stok_Akhir_Lag_1, dan Jumlah_Penjualan, berkontribusi terhadap penurunan *error*. Penelitian selanjutnya disarankan untuk menambah jumlah data historis, menggunakan data harian yang lebih konsisten, menambahkan faktor eksternal seperti hari libur dan promosi, serta membandingkan beberapa algoritma regresi lain agar performa prediksi dapat ditingkatkan.

DAFTAR PUSTAKA

[1] F. Petropoulos, D. Apiletti, V. Assimakopoulos, M. Z. Baba, "Forecasting: Theory and Practice," International Journal of Forecasting, vol. 38, no. 3, pp. 705-871, Jul. 2022, doi: 10.1016/j.ijforecast.2021.11.001.

[2] R. Fildes, S. Ma, and S. Kolassa, "Retail Forecasting: Research and Practice," International Journal of Forecasting, vol. 38, no. 4, pp. 1283-1318, 2022, doi: 10.1016/j.ijforecast.2019.06.004.

[3] S. Punia, K. Nikolopoulos, S. P. Singh, J. K. Madaan, and K. Litsiou, "Deep Learning with Long Short-Term Memory Networks and Random Forests for Demand Forecasting in MULTI-CHANNEL RETAIL," International Journal of Production Research, vol. 58, no. 16, pp. 4964-4979, 2020, doi: 10.1080/00207543.2020.1735666.

[4] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward," PLOS ONE, vol. 13, no. 3, p. e0194889, Mar. 2018, doi: 10.1371/journal.pone.0194889.

[5] Avinash, A. Widjaja, and O. Karnalim, "Analisis Perbandingan Algoritma Machine Learning untuk Forecasting Persediaan Produk Barang Pokok," Jurnal Teknik Informatika dan Sistem Informasi, vol. 10, no. 2, pp. 361-378, Aug. 2024, doi: 10.28932/jutisi.v10i2.9357.

[6] N. R. Sari and A. Alfin, "Optimalisasi Rencana Produksi untuk Mengurangi Overstock dan Stockout di Divisi PPIC Menggunakan Random Forest," Jurnal Pustaka AI, vol. 5, no. 2, pp. 149-165, 2025, doi: 10.55382/jurnalpustakaai.v5i2.1040.

[7] D. Septari, I. Kanedi, and J. Jumadi, "Implementasi Machine Learning Untuk Prediksi Penjualan Oli Shell Pada CV. Harapan Karya Mandiri Bengkulu," Jurnal Media Infotama, vol. 20, no. 2, pp. 534-541, 2024, doi: 10.37676/jmi.v20i2.6513.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

[8] K. Douaioui, R. Oucheikh, O. Benmoussa, and C. Mabrouki, “Machine Learning and Deep Learning Models for Demand Forecasting in Supply Chain Management: A Critical Review,” Applied System Innovation, vol. 7, no. 5, p. 93, 2024, doi: 10.3390/asi7050093.

[9] R. Bergsma, C. de Ruijt, and S. Bhulai, “A Systematic Review of Machine Learning Approaches in Inventory Control Optimization,” Operations Research Perspectives, vol. 15, p. 100367, 2025, doi: 10.1016/j.orp.2025.100367.

[10] Scikit-learn Developers, “DecisionTreeRegressor,” Scikit-Learn Documentation. Accessed: Jun. 2, 2026. [Online]. Available: https://scikit-learn.org/stable/modules/ generated/sklearn.tree.DecisionTreeRegressor.html.

[11] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, An Introduction to Statistical Learning with Applications in Python. Cham: Springer, 2023, doi: 10.1007/978-3-031-38747-0.

[12] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011, doi: 10.48550/arXiv.1201.0490.

[13] A. Nafi’uzzahidi, G. W. N. Wibowo, and Sarwido, “Peningkatan Akurasi Prediksi Stok Bahan Baku Furnitur Menggunakan Algoritma Random Forest Regressor Berbasis Web,” Journal of Information System Research (JOSH), vol. 7, no. 2, pp. 318-329, Jan. 2026, doi: 10.47065/josh.v7i2.9095.

[14] M. S. Shak et al., “Optimizing Retail Demand Forecasting: A Performance Evaluation of Machine Learning Models Including LSTM and Gradient Boosting,” The American Journal of Engineering and Technology, vol. 6, no. 09, pp. 67-80, Sep. 2024, doi: 10.37547/tajet/Volume06Issue09-09.

[15] Django Software Foundation, “Django documentation,” Accessed: Jun. 4, 2026. [Online]. Available: https://docs.djangoproject.com/en/6.0/.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi