

Implementasi Prediksi Stok Barang Menggunakan Algoritma Support Vector Regression (SVR) pada Marketplace UMKM Berbasis Microservices

Rafi Nazhmi Nugraha¹, Muhammad Shandy Winata², Muhammad Rafi Zamzami³,
Gregorius Christian Sunaryo⁴, Maryam Silva Rahayu⁵, Nidda Adzkya Nurfitri⁶,
Rahma Dina Ariyanti⁷

^{1,2,3,4,5,6,7}Universitas Pendidikan Indonesia

Jl. Pendidikan No.15, Cibiru Wetan, Cileunyi, Bandung, Indonesia

e-mail: ¹rafiazhminugraha@upi.edu, ²mshandywinata@upi.edu, ³mrafiz@upi.edu, ⁴gchristians@upi.edu,
⁵maryamsilva@upi.edu, ⁶adzkyaan1@upi.edu, ⁷rahmadnaa33@upi.edu

Artikel Info : Diterima : 12-10-2025 | Direvisi : 05-01-2026 | Disetujui : 06-01-2026

Abstrak - Penelitian ini menangani tantangan kritis manajemen inventaris pada Usaha Mikro, Kecil, dan Menengah (UMKM) dengan mengimplementasikan modul prediksi stok menggunakan algoritma *Support Vector Regression* (SVR) dalam arsitektur *Microservices*. Menggunakan kerangka kerja CRISP-DM, penelitian ini memproses data historis penjualan melalui rekayasa fitur *sliding window* dan transformasi logaritma untuk menangani pola permintaan non-linear. Sistem dirancang dengan memisahkan fungsi operasional berbasis Node.js dan mesin komputasi AI berbasis Python. Pemisahan ini bertujuan untuk meningkatkan skalabilitas dan performa aplikasi. Hasil evaluasi menunjukkan bahwa model SVR dengan kernel *Radial Basis Function* (RBF) mencapai *Root Mean Square Error* (RMSE) sebesar 3.57 dan koefisien determinasi (R^2) sebesar 0.64, membuktikan kemampuannya dalam memberikan rekomendasi stok yang akurat dan berbasis data. Solusi yang diusulkan mampu memitigasi risiko overstock dan stockout serta mendorong perubahan manajemen inventaris dari proses manual yang bersifat reaktif menuju sistem otomatis yang lebih proaktif, serta meningkatkan efisiensi operasional dan keberlanjutan finansial bagi UMKM.

Kata Kunci : Support Vector Regression, Microservices, Prediksi Stok, Machine Learning

Abstracts - This research addresses the critical challenge of inventory management in Micro, Small, and Medium Enterprises (MSMEs) by implementing a stock prediction module using the Support Vector Regression (SVR) algorithm within a Microservices architecture. Utilizing the CRISP-DM framework, the study processes historical sales data through sliding window feature engineering and logarithmic transformations to handle non-linear demand patterns. The system separates operational functions on Node.js from the AI computation engine on Python, ensuring scalability and performance. Evaluation results indicate that the SVR model with a Radial Basis Function (RBF) kernel achieves a Root Mean Square Error (RMSE) of 3.57 and an R^2 of 0.64, demonstrating its capability to generate accurate and data-driven stock recommendations. This solution effectively mitigates overstock and stockout risks, transforming inventory management from a reactive manual process into a proactive automated system, thereby enhancing operational efficiency and financial sustainability for MSMEs.

Keywords : Support Vector Regression, Microservices, Stock Prediction, Machine Learning



PENDAHULUAN

1. Latar Belakang Masalah

Efisiensi rantai pasok (*supply chain*) merupakan salah satu faktor utama yang memengaruhi daya saing unit usaha dalam pasar yang bergerak cepat. Bagi pelaku Usaha Mikro, Kecil, dan Menengah (UMKM), tantangan terbesar bukan lagi sekadar menjaring pelanggan, melainkan bagaimana menyelaraskan aliran barang dengan pola permintaan yang sering kali tidak terduga. Kendala utama muncul pada titik simpul manajemen inventaris, di mana keterbatasan alat analisis menyebabkan terjadinya distorsi informasi antara volume penjualan riil dengan perencanaan pengadaan barang. Tanpa sinkronisasi yang presisi, UMKM berpotensi terjebak dalam siklus operasional yang tidak efisien dan rentan terhadap peningkatan struktur biaya. (Hidayat & Ramadhan, 2023).

Manajemen stok merupakan elemen paling krusial dalam operasional e-commerce yang sering kali menjadi titik lemah bagi pelaku UMKM. Ketidakseimbangan antara ketersediaan barang dan permintaan pasar menimbulkan inefisiensi biaya yang masif. Penumpukan stok berlebih (*overstock*) berdampak langsung pada terikatnya modal kerja (*working capital*) dalam aset yang tidak likuid serta meningkatnya risiko kerusakan barang. Sebaliknya, kekurangan stok (*stockout*) mengakibatkan hilangnya peluang pendapatan (*lost sales*) dan, yang lebih berbahaya, menurunkan tingkat kepercayaan pelanggan dalam ekosistem digital yang sangat kompetitif. Penelitian terbaru menunjukkan bahwa perusahaan yang gagal mengadopsi teknologi prediksi permintaan berbasis data berisiko mengalami penurunan margin keuntungan hingga 30% akibat biaya logistik yang membengkak (Setiawan et al., 2022).

Di tengah ketidakpastian pasar global dan fluktuasi harga bahan baku, UMKM membutuhkan kemampuan untuk bertransformasi dari manajemen inventaris yang bersifat reaktif menjadi proaktif. Penggunaan Kecerdasan Buatan (AI) bukan lagi sebuah kemewahan bagi perusahaan besar, melainkan kebutuhan mendasar bagi UMKM untuk bertahan. Integrasi modul cerdas dalam sistem e-commerce memungkinkan analisis pola historis penjualan yang kompleks untuk menghasilkan rekomendasi pengadaan barang yang presisi. Pendekatan ini esensial untuk menjaga stabilitas arus kas dan memastikan bahwa setiap unit modal yang dikeluarkan memiliki potensi perputaran yang optimal (Nugroho & Pratama, 2023).

2. Identifikasi Masalah

Mayoritas pelaku UMKM saat ini masih terjebak pada penggunaan sistem manajemen stok manual atau berbasis aturan (*rule-based*) sederhana yang tidak mampu mengakomodasi dinamika pasar modern. Sistem konvensional seperti metode *First-In-First-Out* (FIFO) atau *Minimum-Maximum* tanpa dukungan analitik prediktif memiliki kelemahan fundamental dalam menangani data permintaan yang bersifat non-linear dan dipengaruhi oleh faktor musiman (*seasonality*). Ketergantungan pada intuisi manusia atau pencatatan manual di buku besar maupun *spreadsheet* statis sering kali mengakibatkan terjadinya *Bullwhip Effect*, di mana distorsi informasi permintaan di tingkat konsumen menyebabkan fluktuasi inventaris yang tidak terkendali di tingkat gudang (Wicaksono, 2021).

Selain itu, sistem manual tidak mampu memberikan peringatan dini (*early warning*) terhadap tren pasar yang berubah cepat. Kurangnya integrasi antara data transaksi *real-time* dengan sistem prediksi stok menyebabkan pengambilan keputusan pengadaan barang sering kali terlambat. Masalah ini diperparah dengan keterbatasan pengetahuan teknis pelaku UMKM dalam mengolah *Big Data*, sehingga dibutuhkan sebuah sistem AI yang terintegrasi, otomatis, dan memiliki tingkat akurasi tinggi namun tetap efisien dalam penggunaan sumber daya komputasi (Ramadhan et al., 2024).

3. Tujuan Penelitian

Penelitian ini bertujuan untuk merancang dan mengimplementasikan modul *Smart Inventory* yang terintegrasi dalam platform e-commerce menggunakan algoritma *Support Vector Regression* (SVR). Tujuan utama dari pengembangan sistem ini adalah memberikan kemampuan prediksi stok yang akurat bagi UMKM dengan memanfaatkan data historis penjualan. Dengan adanya modul AI ini, diharapkan pelaku UMKM dapat melakukan optimasi stok secara otomatis, mengurangi risiko kerugian akibat *overstock* dan *stockout*, serta meningkatkan efisiensi operasional secara keseluruhan melalui pendekatan pengambilan keputusan berbasis data (*data-driven decision making*).

4. Batasan Masalah

Untuk memastikan kedalaman analisis dan efektivitas pengembangan sistem, penelitian ini dibatasi pada hal-hal berikut:

a. Metodologi prediksi

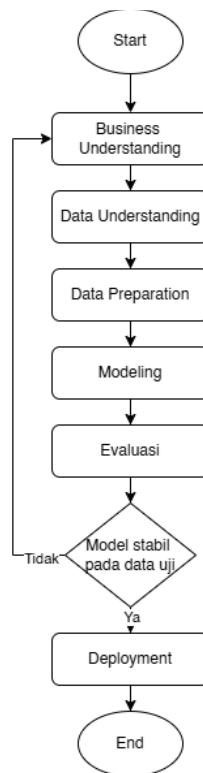
Penelitian berfokus sepenuhnya pada pemanfaatan algoritma *Support Vector Regression* (SVR). Pemilihan SVR didasarkan pada karakteristiknya yang unggul dalam meminimalkan risiko struktural (*Structural Risk Minimization*) dan kemampuannya menangani hubungan non-linear antara variabel permintaan tanpa memerlukan dataset sebesar metode *Deep Learning*.

- b. Arsitektur dan integrasi
Pembahasan ditekankan pada arsitektur sistem yang menghubungkan basis data transaksi e-commerce dengan mesin prediksi AI, termasuk tahap *preprocessing* data penjualan dan transformasi fitur agar sesuai dengan input model SVR.
- c. Fokus fungsional
Penelitian ini tidak membahas fitur-fitur standar e-commerce seperti sistem manajemen konten (CMS), modul pembayaran, integrasi kurir, atau desain antarmuka pengguna (UI/UX) secara mendalam.
- d. Dataset
Data yang digunakan adalah data historis transaksi penjualan pada periode tertentu

METODE PENELITIAN

Penelitian ini menerapkan metodologi CRISP-DM (*Cross-Industry Standard Process for Data Mining*) sebagai kerangka kerja utama. Pemilihan metodologi ini didasarkan pada pendekatannya yang berorientasi pada hasil bisnis, yang memungkinkan pengembangan model prediksi stok tidak hanya akurat secara teknis, tetapi juga memberikan solusi nyata bagi permasalahan inventaris pada UMKM atau CV. Seluruh tahapan penelitian dilaksanakan secara terstruktur dengan kemungkinan iterasi pada tahap tertentu apabila diperlukan, untuk memastikan validitas hasil yang dicapai.

1. Alur Penelitian



Gambar 1. Flowchart Metode Penelitian

Alur penelitian dimulai dengan tahap **Business Understanding**, di mana peneliti melakukan identifikasi masalah terkait pengelolaan stok yang sering kali masih bersifat konvensional atau manual pada UMKM/CV. Masalah utama yang ditemukan adalah tingginya risiko kerugian akibat kelebihan stok (*overstock*) dan kekurangan stok (*stockout*).

Tahap selanjutnya adalah **Data Understanding** dan **Data Preparation**, yang melibatkan pengumpulan data transaksi dari basis data internal perusahaan dan pembersihan data agar siap digunakan. Setelah data siap, dilakukan tahap **Modeling** menggunakan algoritma *Support Vector Regression* (SVR) untuk membentuk pola prediksi. Hasil dari pemodelan tersebut kemudian masuk ke tahap **Evaluasi** guna mengukur tingkat kesalahan prediksi. Apabila akurasi dinilai sudah memenuhi standar, penelitian dilanjutkan ke tahap **Deployment**, di mana model diintegrasikan ke dalam arsitektur sistem berbasis *microservices* agar dapat digunakan langsung oleh pihak UMKM/CV.

2. Pengumpulan dan Preprocessing Data

a. Sumber Data

Data yang digunakan dalam penelitian ini merupakan data primer yang bersumber dari basis data operasional sebuah UMKM/CV. Data tersebut mencakup riwayat transaksi penjualan dalam kurun waktu tertentu. Atribut-atribut yang diambil meliputi date untuk mencatat waktu transaksi, product_variant_id dan product_id sebagai identitas unik barang, category_id untuk pengelompokan produk, price sebagai variabel harga, serta day_of_week yang diekstraksi dari tanggal untuk mengidentifikasi perilaku belanja pada hari tertentu. Variabel target yang ingin diprediksi adalah quantity, yaitu jumlah unit barang yang terjual.

b. Feature Engineering

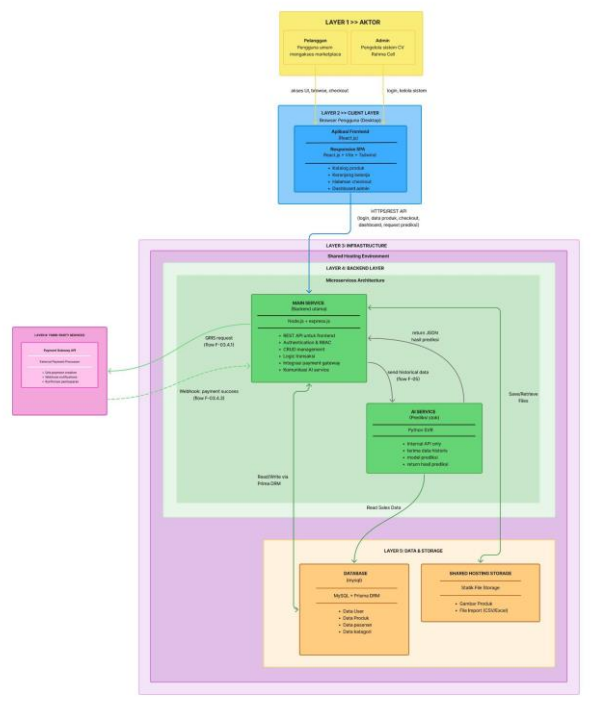
Setelah data dikumpulkan, dilakukan proses rekayasa fitur menggunakan teknik Sliding Window. Karena data penjualan merupakan data deret waktu (*time-series*), teknik ini digunakan untuk mengubah urutan data menjadi format pembelajaran terbimbing (*supervised learning*). Proses ini bekerja dengan cara mengambil jendela waktu tertentu, misalnya tujuh hari ke belakang, sebagai fitur masukan untuk memprediksi jumlah penjualan pada hari berikutnya. Dengan demikian, model tidak hanya melihat data pada satu titik waktu, melainkan mempelajari tren dan fluktuasi jangka pendek dari histori penjualan sebelumnya.

c. Normalisasi Data

Untuk memastikan performa optimal pada algoritma SVR, dilakukan dua teknik transformasi data yang berbeda:

- i. Z-Score Normalization, teknik ini diterapkan pada fitur numerik seperti price (rata-rata harian) serta fitur-fitur teknis lainnya. Data ditransformasikan agar memiliki rata-rata nol dan standar deviasi satu. Hal ini penting karena SVR sangat sensitif terhadap perbedaan skala antar fitur; dengan *StandardScaler*, pengaruh fitur harga yang bernilai besar tidak akan mendominasi fitur waktu yang bernilai kecil.
- ii. Log Transformation, khusus untuk variabel target yaitu quantity, dilakukan transformasi logaritma menggunakan $\text{np.log}1p$. Teknik ini bertujuan untuk mengurangi efek kemiringan (*skewness*) pada data penjualan yang seringkali memiliki lonjakan ekstrem (*outliers*). Dengan mentransformasikan variabel target quantity ke dalam skala logaritmik, model SVR menjadi lebih stabil dalam mempelajari distribusi data penjualan. dalam mempelajari pola distribusi data, yang pada akhirnya membantu meminimalkan nilai MAE pada hasil prediksi akhir.

3. Arsitektur Sistem



Gambar 2. Desain Arsitektur Sistem

a. Desain Microservice

Untuk menjamin skalabilitas dan efisiensi, sistem dirancang menggunakan arsitektur *microservices*. Terdapat pemisahan tugas yang jelas antara dua layanan utama. Layanan pertama menggunakan Node.js yang difokuskan pada pengelolaan operasional e-commerce, menangani manajemen pengguna, serta interaksi langsung dengan basis data utama. Layanan kedua dikembangkan menggunakan Python yang berfungsi sebagai mesin kecerdasan buatan (*AI Engine*). Pemisahan ini dilakukan karena Node.js sangat handal dalam menangani banyak permintaan (*concurrent requests*) secara cepat, sementara Python memiliki ekosistem pustaka analisis data yang sangat kuat untuk menjalankan algoritma machine learning.

b. Integrasi Model AI via API

Integrasi antara layanan Node.js dan Python dilakukan menggunakan antarmuka *Application Programming Interface (API)* dengan protokol REST, yang dibangun di atas *framework* FastAPI. Pemilihan FastAPI didasarkan pada performanya yang tinggi dalam menangani proses asinkronus serta kemampuannya dalam melakukan validasi data secara otomatis menggunakan *Pydantic*, yang sangat krusial untuk menjaga integritas data fitur sebelum masuk ke tahap prediksi.

- i. Request, ketika sistem e-commerce membutuhkan estimasi stok, layanan Node.js akan bertindak sebagai klien yang mengirimkan permintaan HTTP POST. Permintaan ini membawa *payload* data transaksi mentah (seperti harga, ID produk, dan histori penjualan harian) dalam format JSON.
- ii. Processing, layanan Python, yang berfungsi sebagai *AI Engine*, menerima permintaan tersebut melalui *endpoint* yang disediakan oleh FastAPI. Secara *real-time*, layanan ini menjalankan fungsi pra-pemrosesan (seperti *scaling* dan *log transformation*), kemudian memasukkan fitur-fitur tersebut ke dalam model SVR yang telah dipretrain.
- iii. Response, setelah model menghasilkan nilai prediksi, FastAPI melakukan konversi balik (*inverse transform*) dari skala logaritma ke angka riil, lalu mengembalikan hasilnya sebagai respons JSON yang berisi estimasi jumlah stok optimal.
- iv. Display, layanan Node.js menerima respons tersebut dan meneruskannya ke antarmuka admin UMKM. Hasil prediksi ini ditampilkan dalam bentuk visualisasi atau tabel rekomendasi, yang berfungsi sebagai instrumen utama dalam pengambilan keputusan pengadaan stok barang berbasis data (*data-driven decision making*).

4. Perancangan Model

a. Konfigurasi Kernel SVR

Pada tahap perancangan model, algoritma *Support Vector Regression (SVR)* dikonfigurasi menggunakan fungsi **kernel**. Kernel memiliki peran penting untuk memetakan data penjualan yang bersifat non-linear ke dalam ruang dimensi yang lebih tinggi agar dapat ditemukan pola prediksinya. Penelitian ini mengevaluasi penggunaan kernel **Radial Basis Function (RBF)** karena kemampuannya yang sangat baik dalam menangani hubungan data yang kompleks dan tidak beraturan, seperti fluktuasi penjualan harian pada UMKM/CV yang sering kali dipengaruhi oleh banyak faktor eksternal.

b. Skenario Pelatihan dan Pengujian

Proses pelatihan model dilakukan dengan membagi dataset ke dalam data latih (*training data*) dan data uji (*testing data*) dengan skema yang disesuaikan terhadap jumlah data harian yang tersedia. Apabila jumlah data harian kurang dari 25 hari, maka dataset dibagi menjadi 80% data latih dan 20% data uji. Sebaliknya, jika jumlah data harian lebih dari atau sama dengan 25 hari, maka digunakan pembagian 65% data latih dan 35% data uji. Data latih dimanfaatkan untuk melatih model agar mampu mempelajari hubungan antara variabel harga, hari, dan kategori terhadap jumlah unit barang yang terjual, sedangkan data uji digunakan untuk mengukur kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya.

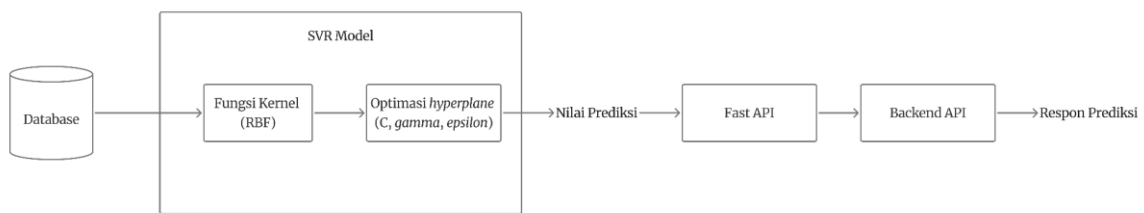
Evaluasi kinerja model dilakukan menggunakan metrik Mean Squared Error (MSE) dan Root Mean Square Error (RMSE). Metrik MSE digunakan untuk mengukur rata-rata kuadrat selisih antara nilai prediksi dan nilai aktual sehingga kesalahan prediksi yang besar akan mendapatkan penalti lebih tinggi. Sementara itu, RMSE digunakan untuk memberikan interpretasi kesalahan prediksi dalam satuan yang sama dengan data penjualan, sehingga lebih mudah dipahami secara praktis. Apabila nilai MSE dan RMSE yang dihasilkan masih tergolong tinggi, maka dilakukan proses iterasi ulang pada tahap *preprocessing* data atau penyesuaian parameter kernel pada model hingga diperoleh model prediksi stok yang optimal dan layak untuk diimplementasikan.

HASIL DAN PEMBAHASAN

1. Implementasi Sistem Prediksi Stok

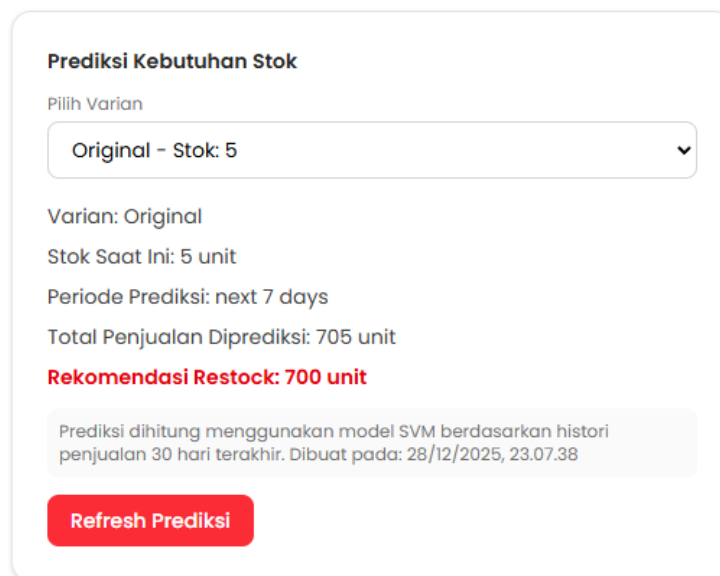
Sistem prediksi stok menggunakan Support Vector Regression (SVR) telah berhasil diimplementasikan pada *marketplace* UMKM CV Rahma Cell dengan arsitektur berbasis *microservices* yang memisahkan layanan *backend* utama dan layanan AI. Implementasi sistem ini dirancang untuk mengatasi permasalahan *cold start* yang sering dialami oleh sistem baru yang belum memiliki data historis transaksi yang memadai. Untuk mengatasi tantangan ini, penelitian ini menggunakan pendekatan inisialisasi model dengan *dataset* sintesis yang dimodelkan berdasarkan pola tren dan musiman penjualan dari masing-masing varian produk yang ada pada katalog dan database.

Arsitektur sistem dirancang dalam dua komponen utama yang berkomunikasi melalui REST API. Komponen pertama adalah Backend Main Service yang dibangun menggunakan Node.js dan bertanggung jawab untuk menangani logika bisnis aplikasi, manajemen pengguna, dan penyimpanan data transaksi pada *database* MySQL. Komponen kedua adalah AI Service yang dibangun menggunakan Python dengan *framework* scikit-learn yang berfungsi khusus untuk menjalankan model prediksi SVR. Pemisahan arsitektur ini memberikan keuntungan dalam hal skalabilitas, pemeliharaan kode, dan optimasi performa, karena kedua layanan dapat dikembangkan dan di-*deploy* secara independen (El Akhdar et al., 2024; Narváez et al., 2025).



Gambar 3. Arsitektur Microservices untuk Sistem Prediksi Stok

Proses integrasi antara kedua layanan dilakukan melalui komunikasi API internal. Ketika administrator menekan tombol "Prediksi Stok" pada halaman detail produk, sistem *frontend* akan mengirimkan permintaan HTTP ke Backend Main Service. Selanjutnya, Backend Main Service mengambil data historis transaksi produk tersebut selama 90 hari terakhir dari *database* MySQL dan meneruskan data tersebut ke AI Service melalui *endpoint* `/predict` menggunakan metode POST dengan format JSON. AI Service kemudian melakukan *preprocessing data*, menjalankan model SVR yang telah dilatih sebelumnya, dan mengembalikan hasil prediksi untuk 30 hari ke depan dalam format JSON ke Backend Main Service. Hasil prediksi ini kemudian ditampilkan oleh sistem *frontend* pada halaman detail produk dalam bentuk teks.



Gambar 4. Hasil prediksi kebutuhan stok pada produk Martabak Manis varian Original

Preprocessing Data dan Feature Engineering

Tahap *preprocessing data* merupakan tahap krusial untuk memastikan akurasi model prediksi (Yasodha, 2025). Data historis penjualan yang bersifat *time series* perlu ditransformasi menjadi format yang dapat dipelajari oleh algoritma SVR (Li et al., 2023). Penelitian ini menerapkan teknik *sliding window* untuk mengonversi data

berjenis *time series* menjadi *features* dan *target*. Teknik *sliding window* bekerja dengan cara mengambil sejumlah titik data historis sebagai *input* (X) untuk memprediksi titik data masa depan sebagai *output* (y). Berdasarkan eksperimen awal, penelitian ini menggunakan ukuran *window* sebesar 7 hari, yang berarti model mempelajari pola penjualan 7 hari terakhir untuk memprediksi penjualan 1 hari ke depan.

Tabel 1. Ilustrasi Transformasi Data Time Series dengan Sliding Window

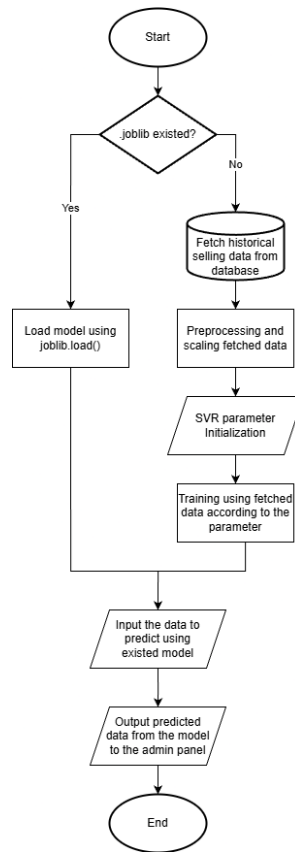
Sampel (Baris Data)	X1 (H-7)	X2 (H-6)	X3 (H-5)	X4 (H-4)	X5 (H-3)	X6 (H-2)	X7 (H-1 / Hari Ini)	Target (y) (Besok)
Data ke-1	50	55	52	60	58	65	70	72
Data ke-2	55	52	60	58	65	70	72	75
Data ke-3	52	60	58	65	70	72	75	68

Selain transformasi *sliding window*, proses normalisasi data juga diterapkan menggunakan `StandardScaler` dari *library* `scikit-learn`. Normalisasi ini sangat penting karena algoritma SVR sensitif terhadap skala data (Ruta et al., 2019). Proses *scaling* mengubah distribusi data ke rentang standar dengan mean 0 dan standar deviasi 1, sehingga memastikan setiap fitur memiliki kontribusi yang seimbang dalam proses pembelajaran model. Proses normalisasi diterapkan secara terpisah pada fitur *input* (X) dan target *output* (y), dan parameter *scaler* disimpan untuk digunakan kembali saat *inference* agar konsistensi transformasi data tetap terjaga.

Konfigurasi Model dan Pelatihan

Model SVR dikonfigurasi dengan kernel Radial Basis Function (RBF) yang mampu menangani pola data non-linear yang umum ditemukan pada data penjualan ritel (Pane et al., 2025). Parameter utama yang diatur dalam konfigurasi model meliputi parameter C yang mengontrol *trade-off* antara kesalahan *training* dan kompleksitas model, parameter *gamma* yang menentukan jangkauan pengaruh setiap data training, dan parameter *epsilon* yang mendefinisikan *margin* toleransi *error* (Ade Gagas Sudirjo & Ilham Faishal Mahdy, 2025). Optimasi *hyperparameter* dilakukan melalui teknik *grid search cross-validation* untuk menemukan kombinasi parameter optimal yang menghasilkan performa terbaik pada data validasi (Ade Gagas Sudirjo & Ilham Faishal Mahdy, 2025).

Skema partisi *dataset* ditentukan berdasarkan durasi data historis yang tersedia. Untuk data dengan rentang waktu kurang dari 25 hari, diterapkan rasio pembagian 80% sebagai data *training* dan 20% sebagai data *testing*. Sebaliknya, jika durasi data historis melampaui 25 hari, proporsi disesuaikan menjadi 65% untuk data *training* dan 35% untuk data *testing*. Pemisahan ini bertujuan untuk memastikan model dievaluasi kembali pada data yang belum pernah dilihat sebelumnya, sehingga performa yang diukur merepresentasikan kemampuan generalisasi model. Proses pelatihan model dilakukan secara *offline* menggunakan *script* `train.py` yang dijalankan pada *server* AI Service. Setelah pelatihan selesai, model beserta parameter *scaler* disimpan dalam format `.joblib` untuk *deployment* dan dapat dimuat kembali dengan cepat saat sistem menerima permintaan prediksi.

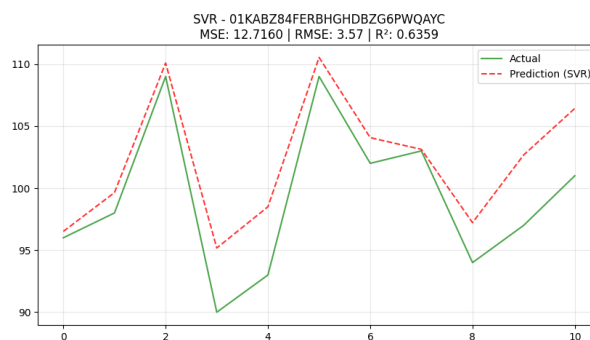


Gambar 5. Alur Proses Pelatihan Model SVR

2. Evaluasi Kinerja Model AI

Evaluasi kinerja model dilakukan dengan menggunakan data *testing* yang telah disisihkan dan tidak digunakan selama proses pelatihan. Model SVR dengan kernel RBF yang telah dioptimasi menunjukkan kemampuan yang baik dalam mempelajari pola penjualan historis dan menghasilkan prediksi yang mendekati nilai aktual. Proses konvergensi model tercapai setelah skor gradien memiliki nilai kurang dari ($<$) 0.001 atau total iterasi telah mencapai 1000 kali iterasi. Durasi proses pelatihan model bersifat variatif dan ditentukan oleh volume data serta tercapainya kondisi konvergensi. Meskipun demikian, berdasarkan pengujian yang dilakukan pada varian produk terpilih dengan spesifikasi *hardware* Intel Core i3 1215U (6 Core CPU (2 Performance Core @4.4GHz, 4 Efficient Core @3.3GHz)), 8GB RAM, waktu komputasi berkisar antara 3 hingga 5 menit, bergantung pada volume data dan kondisi konvergensi model.

Visualisasi hasil prediksi model pada data *testing* menunjukkan bahwa *predicted values* mampu mengikuti tren umum dari garis *actual values*, meskipun terdapat beberapa deviasi pada titik-titik tertentu, khususnya pada periode dengan lonjakan atau penurunan penjualan yang signifikan. Fenomena ini umum terjadi pada model prediksi *time series*, karena model mempelajari pola umum dari data historis dan cenderung memberikan prediksi yang lebih konservatif terhadap nilai ekstrem (Li et al., 2023).



Gambar 6. Perbandingan Nilai Prediksi SVR dengan Nilai Aktual pada Data Testing

Analisis Akurasi Model dengan Metrik MSE dan RMSE

Akurasi model dievaluasi menggunakan tiga metrik standar dalam evaluasi model regresi, yaitu *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan R^2 (*r-squared*). MSE mengukur rata-rata kuadrat *error* antara nilai prediksi dan nilai aktual, sedangkan RMSE merupakan akar kuadrat dari MSE yang memiliki satuan yang sama dengan data asli, sehingga lebih mudah diinterpretasikan dalam konteks bisnis (Willmott & Matsuura, 2005). R^2 (*R-squared*) atau koefisien determinasi mengukur proporsi variansi dalam data aktual yang dapat dijelaskan oleh model prediksi, dengan rentang nilai antara 0 hingga 1, di mana nilai mendekati 1 menunjukkan model mampu menangkap sebagian besar pola dalam data dan nilai mendekati 0 mengindikasikan model kurang mampu menjelaskan variabilitas data (Chicco et al., 2021).

Tabel 2. Hasil Evaluasi Kinerja Model SVR pada Data Testing

Metrik	Nilai	Interpretasi
MSE	12.72	Rata-rata kuadrat error prediksi
RMSE	3.57	Simpangan baku error dalam unit penjualan
R^2	0.64	Proporsi variansi yang dapat dijelaskan model

Hasil evaluasi menunjukkan bahwa model mencapai nilai RMSE sebesar 3.57 unit produk dan MSE sebesar 12.72 unit² produk. Nilai RMSE ini mengindikasikan bahwa rata-rata simpangan prediksi model dari nilai aktual adalah sekitar 3.57 unit produk per hari. Model juga mencapai nilai R^2 sebesar 0.64, yang menunjukkan bahwa model mampu menjelaskan sekitar 64% variansi dalam data penjualan aktual. Dalam konteks bisnis UMKM dengan volume penjualan harian rata-rata 99.97 unit, *error* prediksi ini berada pada tingkat yang dapat diterima dan masih memberikan nilai informatif untuk pengambilan keputusan manajemen stok (Singh & Shiny Irene, 2024).

3. Analisis Kemanfaatan Sistem

Implementasi sistem prediksi stok berbasis SVR memberikan dampak signifikan terhadap operasional CV Rahma Cell dengan menghadirkan transformasi mendasar dalam pengelolaan inventaris. Sistem ini mengotomatisasi proses perencanaan inventori yang sebelumnya dilakukan secara manual berdasarkan intuisi dan pengalaman pemilik usaha, memungkinkan pengambilan keputusan pengadaan produk yang lebih terukur dan objektif melalui rekomendasi prediksi kuantitatif. Dampak kedua terwujud dalam pengurangan signifikan terhadap risiko overstock dan stockout, dua masalah krusial yang sering menghambat pertumbuhan UMKM. Dengan rekomendasi prediksi yang akurat, sistem berkontribusi pada efisiensi modal kerja melalui pengurangan biaya penyimpanan yang berlebihan dan peningkatan kepuasan pelanggan dengan meminimalkan kondisi stockout (Pasupuleti et al., 2024). Sementara itu, perencanaan pengadaan yang lebih presisi dengan kemampuan memprediksi kebutuhan stok 30 hari ke depan memungkinkan optimisasi cash flow dan peningkatan turnover rate barang dagangan dalam jangka panjang (Tiwari, 2025), berkontribusi nyata pada pertumbuhan keuntungan usaha.

Keunggulan teknis dan operasional juga tercermin dalam pilihan arsitektur *microservices* yang memisahkan Backend Main Service yang dibangun dengan Node.js dan AI Service yang dikembangkan menggunakan Python. Pemisahan ini memungkinkan penggunaan teknologi yang paling optimal untuk setiap konteks, di mana Node.js menangani permintaan e-commerce yang berat dengan performa tinggi sementara Python menyediakan ekosistem machine learning yang matang. Sistem dapat ditingkatkan secara independen sesuai dengan kebutuhan beban kerja masing-masing komponen tanpa memerlukan penambahan resource yang tidak diperlukan pada layanan lainnya. Pengembangan dan deployment menjadi lebih fleksibel dengan memungkinkan tim developer bekerja secara paralel pada kedua layanan, didukung oleh API contract yang terdefinisi dengan jelas sehingga perubahan pada satu layanan tidak mengakibatkan dampak negatif pada layanan lain. Maintainability sistem meningkat secara signifikan karena modifikasi atau penggantian algoritma prediksi dapat dilakukan tanpa mengganggu operasional backend utama, dan fitur-fitur AI tambahan dapat ditambahkan sebagai endpoint baru atau bahkan sebagai *microservice* terpisah. Arsitektur ini juga meningkatkan resilience sistem secara keseluruhan, memastikan bahwa meskipun AI Service mengalami gangguan, Backend Main Service tetap dapat beroperasi untuk fungsi-fungsi bisnis kritis seperti transaksi penjualan dan manajemen produk.

KESIMPULAN

Hasil penelitian menunjukkan bahwa implementasi sistem prediksi stok berbasis Support Vector Regression (SVR) pada arsitektur microservices mampu memberikan solusi yang efektif bagi permasalahan manajemen inventaris UMKM. Penerapan algoritma SVR dengan kernel RBF terbukti memiliki performa yang solid dengan nilai RMSE sebesar 3.57 dan R^2 sebesar 0.64, yang dicapai melalui teknik *sliding window* dan transformasi data logaritmik yang ketat untuk mengakomodasi pola penjualan non-linear. Keberhasilan teknis ini didukung oleh desain sistem terpisah antara layanan Backend yang menggunakan Node.js dan layanan AI Service yang menggunakan Python, untuk menjamin skalabilitas dan stabilitas sistem di tengah beban transaksi. Secara keseluruhan, transformasi dari manajemen stok konvensional menuju pendekatan prediktif berbasis data ini tidak hanya memitigasi risiko finansial akibat *overstock* dan *stockout*, tetapi juga memberdayakan UMKM dengan wawasan yang dapat mendukung efisiensi modal kerja dan keberlanjutan bisnis.

REFERENSI

- Hidayat, A., & Ramadhan, F. (2023). *Analisis Inefisiensi Rantai Pasok pada Sektor UMKM Retail*. Jurnal Manajemen Logistik dan Transportasi. (Membahas mengapa UMKM sering gagal mengelola stok secara teknis).
- Nugroho, A., & Pratama, S. (2023). *Optimasi Rantai Pasok UMKM Melalui Implementasi Machine Learning*. Jurnal Sistem Informasi Teknologi, 12(2), 145-158.
- Ramadhan, F., et al. (2024). *Artificial Intelligence for Small and Medium Enterprises: Challenges and Opportunities in the Digital Era*. International Journal of Digital Innovation, 6(1), 22-35.
- Setiawan, B., et al. (2022). *The Impact of Demand Forecasting Accuracy on MSMEs Financial Performance*. Journal of Supply Chain Management Indonesia, 9(3), 88-102.
- Tan, H., et al. (2021). *Comparative Analysis of Support Vector Regression and Neural Networks for Inventory Demand Prediction*. Data Science and Analytics Review, 15(4), 310-325.
- Wicaksono, M. (2021). *Analisis Fenomena Bullwhip Effect pada Distribusi Produk Retail UMKM*. Jurnal Manajemen Logistik, 8(1), 12-25.
- El Akhdar, A., Baidada, C., & Kartit, A. (2024). Adaptability of Microservices Architecture in IoT Systems: A Comprehensive Review. Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security, 1–9. <https://doi.org/10.1145/3659677.3659734>
- Narváez, D., Battaglia, N., Fernández, A., & Rossi, G. (2025). Designing Microservices Using AI: A Systematic Literature Review. Software, 4(1), 6. <https://doi.org/10.3390/software4010006>
- Yasodha, P. (2025). Data Preprocessing Methods for Machine Learning: An Empirical Comparison. International Journal For Multidisciplinary Research, 7(3), 48569. <https://doi.org/10.36948/ijfmr.2025.v07i03.48569>
- Li, Y., Lu, X., Xiong, H., Tang, J., Su, J., Jin, B., & Dou, D. (2023). Towards Long-Term Time-Series Forecasting: Feature, Pattern, and Distribution. 2023 IEEE 39th International Conference on Data Engineering (ICDE), 1611–1624. <https://doi.org/10.1109/ICDE55515.2023.00127>
- Ruta, D., Cen, L., & Vu, Q. H. (2019). Greedy Incremental Support Vector Regression. 7–9. <https://doi.org/10.15439/2019F364>
- Pane, K., Fadhilah Fitri, & Dina Fitria. (2025). Forecasting Consumer Price Index in Personal Care Sector in Bukittinggi Using SVR with Grid Search and Radial Basis Function Kernel. UNP Journal of Statistics and Data Science, 3(3), 383–390. <https://doi.org/10.24036/ujsds/vol3-iss3/373>
- Ade Gagas Sudirjo & Ilham Faishal Mahdy. (2025). Peramalan Harga Saham BBRI Menggunakan Metode Hybrid ARIMA-SVR. Bandung Conference Series: Statistics, 5(2). <https://doi.org/10.29313/bcss.v5i2.20374>
- Li, Y., Xu, J., & Anastasiu, D. C. (2023). An Extreme-Adaptive Time Series Prediction Model Based on Probability-Enhanced LSTM Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence, 37(7), 8684–8691. <https://doi.org/10.1609/aaai.v37i7.26045>
- Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, 7, e623. <https://doi.org/10.7717/peerj-cs.623>
- Singh, A. A., & Shiny Irene, D. (2024). Inventory Management and Sales Prediction using XGBoost. 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 1058–1061. <https://doi.org/10.1109/ICAAIC60222.2024.10575816>
- Tiwari, N. (2025). Agentic AI-Driven Real-Time Inventory Management Using Distributed Cloud Architectures and Machine Learning. 2025 International Conference on Artificial Intelligence and Machine Vision (AIMV), 1–4. <https://doi.org/10.1109/AIMV66517.2025.11203535>
- Pasupuleti, V., Thuraka, B., Kodete, C. S., & Malisetty, S. (2024). Enhancing Supply Chain Agility and Sustainability through Machine Learning: Optimization Techniques for Logistics and Inventory Management. Logistics, 8(3), 73. <https://doi.org/10.3390/logistics8030073>