

Oil Demand Forecasting With the Integration of Support Vector Regression (SVR) and Particle Swarm Optimization (PSO)

[Peramalan Permintaan Oli Dengan Integrasi Support Vector Regression (SVR) Dan Particle Swarm Optimization (PSO)]

Aulia Putri Ramdhani ¹⁾, Tedjo Sukmono ^{*,2)}

¹⁾ Program Studi Teknik Industri, Universitas Muhammadiyah Sidoarjo, Indonesia

²⁾ Program Studi Teknik Industri, Universitas Muhammadiyah Sidoarjo, Indonesia

*Email Penulis Korespondensi: thedjoss@umsida.ac.id

Abstract. Inventory control is a problem often faced by companies, such as PT.XYZ, which has not established a safety stock system for its oil inventory. This is due to fluctuations in oil demand each month, which averages 53%. This study aims to determine the results of oil forecasting for the next period and produce forecasts with a minimum error rate. The method used in this study is to integrate machine learning, namely Support Vector Regression (SVR) with Particle Swarm Optimization (PSO), assisted by the Radial Basis Function (RBF) kernel. This study produced forecast results with a MAPE value of 18% with optimal parameters $C = 199.9993$, $\varepsilon = 0.0744$, $\sigma = 0.2487$, $cLR = 0.1954$, and $\lambda = 0.003$. The results of this study can then be used by companies as a good reference in determining safe stock and the optimal number of oil orders.

Keywords - Forecasting; Support Vector Regression; Particle Swarm Optimization; Safety Stock

Abstrak. Pengendalian persediaan (inventory control) adalah permasalahan yang sering dihadapi oleh perusahaan, seperti pada PT. XYZ yang belum menetapkan sistem stok aman (safety stock) pada persediaan oli. Hal ini karena adanya fluktuasi permintaan oli setiap bulannya yang rata-rata sebesar 53%. Penelitian ini bertujuan untuk mengetahui hasil peramalan oli pada periode berikutnya dan menghasilkan peramalan dengan tingkat kesalahan minimum. Metode yang digunakan pada penelitian ini adalah dengan mengintegrasikan machine learning yaitu Support Vector Regression (SVR) dengan Particle Swarm Optimization (PSO) yang dibantu oleh kernel Radial Basis Function (RBF). Dari penelitian ini didapatkan hasil peramalan dengan nilai MAPE = 18% dengan parameter optimum $C = 199,9993$, $\varepsilon = 0,0744$, $\sigma = 0,2487$, $cLR = 0,1954$ dan $\lambda = 0,003$. Hasil penelitian ini selanjutnya dapat digunakan oleh perusahaan sebagai acuan yang baik dalam menentukan stok aman dan jumlah pemesanan optimum oli.

Kata Kunci – Peramalan; Support Vector Regression; Particle Swarm Optimization; Stok Aman

I. PENDAHULUAN

Perencanaan dan pengendalian produksi merupakan salah satu permasalahan penting yang pasti dihadapi oleh perusahaan, termasuk dalam hal ini adalah pengendalian persediaan terhadap berbagai macam peralatan dan material pendukung. Hal ini bertujuan untuk mempertahankan pertumbuhan dan perkembangan perusahaan[1]. Perencanaan disusun atas dasar perkiraan berdasarkan data masa lalu, salah satu metode yang tepat digunakan yaitu peramalan. Peramalan adalah proses untuk memperkirakan kebutuhan di masa datang untuk memenuhi dari segi kuantitas, kualitas maupun waktu[2]. Kesuksesan peramalan didukung dengan ketersediaan data historis yang jumlahnya besar atau biasa disebut dengan *big data*[3]. Pada penelitian ini dilakukan peramalan menggunakan bantuan *machine learning*, yaitu sebuah tipe kecerdasan buatan (AI) yang menyediakan komputer dengan kemampuan untuk belajar dari data tanpa harus diprogram terlebih dahulu[4].

PT. XYZ adalah perusahaan yang bergerak di bidang manufaktur, memproduksi beberapa jenis beton seperti beton siap pakai, beton pracetak, dan beton *masonry*. Di tengah kegiatan produksinya, perusahaan ini juga mengalami permasalahan terkait pengendalian persediaan (*inventory*) pada gudang *spare part*. Salah satu nya adalah perusahaan belum menetapkan sistem stok aman (*safety stock*) pada seluruh *spare part* dan bahan habis pakai (*consumable part*) di gudang dikarenakan permintaannya yang tidak menentu setiap bulannya. Hal ini dibuktikan dengan adanya fluktuasi permintaan oli yang termasuk dalam jenis bahan habis pakai (*consumable part*). Pada salah satu gudangnya diketahui fluktuasi permintaan oli rata-rata sekitar 53% setiap bulannya.

Oleh karena itu pada penelitian ini akan dilakukan peramalan untuk mengetahui permintaan oli pada periode selanjutnya sehingga dapat digunakan sebagai dasar menentukan kuantitas pembelian oli dan stok aman di gudang. Metode peramalan yang digunakan adalah *Support Vector Regression* (SVR). Metode *Support Vector Regression* adalah modifikasi metode *support vector machine learning* yang digunakan dalam pembangunan *hyperlane* pada kasus regresi[5]. Namun, dikarenakan data pada kasus dalam penelitian ini bukan merupakan data linier, maka peramalan ditambahkan dengan bantuan fungsi kernel[6]. Jenis kernel yang dipilih adalah kernel *Radial Basis Function* (RBF), yaitu fungsi kernel yang digunakan ketika data tidak dapat terpisahkan secara linier[7]. Selanjutnya,

Copyright © Universitas Muhammadiyah Sidoarjo. This preprint is protected by copyright held by Universitas Muhammadiyah Sidoarjo and is distributed under the Creative Commons Attribution License (CC BY). Users may share, distribute, or reproduce the work as long as the original author(s) and copyright holder are credited, and the preprint server is cited per academic standards.

Authors retain the right to publish their work in academic journals where copyright remains with them. Any use, distribution, or reproduction that does not comply with these terms is not permitted.

untuk menghitung akurasi hasil peramalan dengan MAPE dilakukan dengan menggunakan algoritma *Particle Swarm Optimization* (PSO) yaitu algoritma pencarian heuristik yang didasarkan pada algoritma evolusi dan kehidupan buatan. Parameter algoritma ini dapat dengan mudah disesuaikan, sehingga memungkinkan penerapannya pada berbagai masalah praktis[8]. Metode peramalan *Support Vector Regression* (SVR) dipilih karena jauh lebih efisien untuk diterapkan pada jenis data non-linier dibandingkan dengan metode peramalan tradisional lain seperti *Autoregressive Moving Average* (ARIMA) yang merupakan metode peramalan tradisional yang salah satu aspek penting penerapannya adalah linearitas kumpulan data yang ingin diprediksi[9].

Penelitian terkait pernah dilakukan oleh Angela untuk meramalkan penjualan produk pakaian dengan tujuan membantu dalam pengambilan keputusan pada pengendalian persediaan dan rencana strategis perusahaan[10]. Penelitian lain dilakukan oleh Umiyati untuk melakukan peramalan harga batu bara acuan di masa mendatang dengan tujuan membantu perusahaan dalam menentukan alokasi dana yang tepat sehingga keuntungan dapat dimaksimalkan dan biaya produksi dapat ditekan[11].

Penelitian ini bertujuan untuk mengetahui hasil peramalan oli pada periode berikutnya menggunakan metode *Support Vector Regression* (SVR) dan *Particle Swarm Optimization* (PSO) sehingga dapat digunakan sebagai dasar menentukan jumlah stok aman (*safety stock*) dan jumlah pemesanan optimum. Selain itu, untuk menghasilkan peramalan dengan tingkat kesalahan minimum sehingga dapat digunakan sebagai acuan yang baik bagi perusahaan.

II. METODE

A. Metode Penelitian

Metode yang digunakan untuk melakukan peramalan pada penelitian ini adalah *Support Vector Regression* (SVR) dan *Particle Swarm Optimization* (PSO). Namun, sebelum dilakukan seluruh rangkaian pengolahan data menggunakan metode tersebut terlebih dahulu perlu dilakukan normalisasi data. Normalisasi data adalah proses membuat variabel memiliki rentang nilai yang sama sehingga data menjadi seragam, tidak ada data yang terlalu besar maupun terlalu kecil[12]. Tujuan dari normalisasi data adalah untuk membentuk data dalam posisi nilai dengan rentang yang sama[13]. Beberapa metode yang dapat digunakan untuk melakukan normalisasi data yaitu *Min-Max Normalization*, *Z-Score Normalization*, dan *Decimal Scaling Normalization*[12]. Pada penelitian ini akan dilakukan uji normalisasi data pada rentang 0 – 1 dengan tujuan untuk meningkatkan akurasi regresi dengan mengurangi tingkat error [10][14], dengan rumus sebagai berikut:

$$X_{scaled} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Sumber: [15]

Keterangan :

X_{scaled} = Uji normalisasi
 X_i = Data yang akan dinormalisasi
 X_{min} = Batas bawah data
 X_{max} = Batas atas data

Kemudian pada proses denormalisasi, yaitu sebuah proses pengembalian data hasil peramalan ke dalam angka sebenarnya dapat dilakukan menggunakan persamaan sebagai berikut[16]:

$$\text{Denormalisasi} = X_i' \times (X_{max} - X_{min}) + X_{min} \quad (2)$$

Sumber: [16][17][18]

Keterangan :

X_i' = Data hasil normalisasi
 X_{min} = Batas bawah data
 X_{max} = Batas atas data

1. *Support Vector Regression* (SVR)

Support Vector Regression adalah kembangan model regresi dari *Support Vector Machine* (SVM) yang biasanya diterapkan dalam konteks masalah klasifikasi[14][19]. Sebagai model regresi, SVR bertujuan untuk memperkirakan fungsi $f(x)$ dan mengikuti prinsip minimalisasi risiko struktural sehingga dapat meningkatkan akurasi prediksi[20]. Dalam regresi oleh SVR, hubungan non-linier dan rumit antara *input* dan *output* dapat diperkirakan SVR tersusun dari analisis prediksi, dan kinerjanya dipengaruhi oleh tiga parameter, yaitu C , γ , dan ϵ [21]. Tingkat akurasi prediksi SVR sangat dipengaruhi oleh fungsi kernel yang digunakan[22]. Oleh karena itu, pada penelitian ini akan juga digunakan fungsi kernel *Radial Basis Function* (RBF), yaitu fungsi kernel yang paling sering digunakan karena memiliki beberapa parameter penyesuaian[23]. Berikut adalah langkah-langkah dalam melakukan peramalan menggunakan *Support Vector Regression*[10][14][11]:

- a. Inisialisasi parameter. Dalam metode SVR digunakan beberapa parameter yaitu ϵ dan C , yang berpengaruh dalam menentukan toleransi kesalahan, CLR sebagai penentu kecepatan proses pembelajaran, σ sebagai konstanta yang memengaruhi distribusi dimensi data, dan λ sebagai penentu skala dimensi pemetaan kernel SVR.

- b. Pembentukan Matriks *Hessian* yang dikombinasikan menggunakan kernel *Radial Basis Function* (RBF) menggunakan persamaan berikut.

$$[R]_{ij} = \exp \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) + \lambda^2 \quad (3)$$

Sumber: [10][11][14][24][25]

Keterangan :

$[R]_{ij}$ = Matriks Hessian
 x_i = Data ke-i
 σ = Konstanta dimensi
 λ = Variabel skalar

- c. Menghitung nilai *error*, dengan menginisialisasi nilai *lagrange multiplier* α_i dan α_i^* menjadi 0, menghitung nilai galat, menghitung perubahan *lagrange multiplier*, dan pembaharuan nilai *lagrange multiplier*. Menghitung nilai *error* dapat dilakukan menggunakan persamaan berikut.

$$E_i = Y_i - \sum (\alpha_j^* - \alpha_j) R_{ij} \quad (4)$$

Sumber: [11][14][24]

Keterangan :

E_i = Nilai eror pada posisi ke-i
 Y_i = Nilai aktual data latih ke-i
 $\alpha_j^* - \alpha_j$ = Nilai *Lagrange Multiplier* terakhir
 R_{ij} = Matriks Hessian

Selanjutnya perhitungan perubahan nilai *lagrange multiplier* menggunakan persamaan sebagai berikut:

$$\delta\alpha_i^* = \min \{ \max[\gamma (E_i - \varepsilon), -\alpha_i^*], C - \alpha_i^* \} \quad (5)$$

$$\delta\alpha_i = \min \{ \max[\gamma (-E_i - \varepsilon), -\alpha_i], C - \alpha_i \} \quad (6)$$

Sumber : [11][14][26]

Keterangan :

$\delta\alpha_i$ = Perubahan nilai dari α_i
 γ = *Learning rate*
 ε = Nilai toleransi *error*
 C = Nilai kompleksitas

Dan terakhir pembaharuan nilai *lagrange multiplier* dapat dihitung menggunakan persamaan sebagai berikut:

$$\alpha_i^* = \alpha_i^* + \delta\alpha_i^* \quad (7)$$

$$\alpha_i = \alpha_i + \delta\alpha_i \quad (8)$$

Sumber : [11][14][26]

- d. Proses iterasi, yaitu mengulang langkah ketiga hingga salah satu kondisi berikut terpenuhi: (1) Iterasi mencapai batas iterasi maksimum yang telah ditentukan, (2) Tidak ada perubahan nilai pada *lagrange multiplier* (konvergensi), (3) Perubahan nilai telah memenuhi persyaratan perubahan nilai *lagrange multiplier* tidak lebih dari konstanta epsilon ($\max(|\delta\alpha_i|) < \varepsilon$ dan $\max(|\delta\alpha_i^*|) < \varepsilon$).
- e. Menghitung hasil peramalan. Nilai hasil peramalan yang didapatkan setelah perhitungan regresi dirumuskan pada persamaan berikut.

$$f(x) = \sum_{i=1}^n (\alpha_j^* - \alpha_j) (K(x_i, x_j) + \lambda^2) \quad (9)$$

Sumber: [10][11][14][21][22]

Keterangan :

$f(x)$ = Hasil peramalan
 $\alpha_j^* - \alpha_j$ = Nilai *Lagrange Multiplier*
 $K(x_i, x_j)$ = Fungsi Kernel
 λ = Variabel skalar

2. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) adalah pendekatan optimasi stokastik yang meniru perilaku sosial kawanan burung[4]. Pada PSO, *particle* merujuk pada individu, dan *swarm* merujuk pada populasi[22]. Segerombolan partikel akan mencari ruang dalam dimensi tertentu dan menemukan solusi terbaik yang mengoptimalkan masalah yang tengah dihadapi[27]. Dalam algoritma ini juga partikel mewakili solusi masalah pengoptimalan yang ditandai dengan menginisialisasi posisi dan kecepatan di ruang pencarian, dan nantinya algoritma ini akan secara berulang mencari solusi optimal berdasarkan nilai fungsi partikel *fitness*-nya[25]. Berikut ini adalah langkah-langkah dalam melakukan optimasi menggunakan algoritma *Particle Swarm Optimization*[10][11][28]:

- a. Menghitung *K-Fold Cross Validation*, yaitu proses mempersiapkan data yang akan digunakan untuk melatih SVR dengan cara mengelompokkan antara data latih dan data uji dari keseluruhan data. Pada tahap ini dilakukan penentuan Data Latih (*Data Training*) dan Data Uji (*Data Testing*). Tahap ini dilakukan dengan membagi proporsi

data ke dalam dua bagian, dengan perbandingan data uji dan data latih 20:80. Dipilihnya proporsi *split data* tersebut dengan alasan untuk menghasilkan nilai akurasi peramalan yang tinggi seperti yang telah di uji coba pada penelitian sebelumnya[29][30].

- b. Inisialisasi posisi partikel untuk menutupi ruang pencarian secara seragam menggunakan rumus sebagai berikut.

$$x(0) = x_{\min,j} + r_j (x_{\max,j} - x_{\min,j}) \quad (10)$$

Sumber: [31]

Keterangan :

$x(0)$ = Posisi partikel

$x_{\min}$ = Batas bawah setiap dimensi

$x_{\max}$ = Batas atas setiap dimensi

r_j = Nilai *random* dalam *range* (0,1)

- c. Mengevaluasi Fungsi *Fitness* dengan pelatihan metode SVR, yaitu dengan memilih fitur, *sequential learning*, menguji model regresi dan perhitungan *error*.
- d. Mencari nilai *pBest*, yaitu posisi terbaik yang pernah dicapai partikel dengan cara membandingkan nilai *cost pBest* dengan partikel iterasi saat ini.
- e. Mencari nilai *gBest* dan nilai Bobot Inersia Baru. *gBest* (*global best*) merupakan posisi terbaik partikel yang dapat dicari dengan membandingkan nilai *cost gBest* dengan *pBest*. Sedangkan menghitung nilai bobot inersia baru dapat dilakukan menggunakan rumus:

$$w = w_{\max} + \frac{w_{\max} - w_{\min}}{t_{\max}} \times t \quad (11)$$

Sumber: [10][11][32]

Keterangan :

w = Bobot inersia

$w_{\max}$ = Batas atas bobot inersia

$w_{\min}$ = Batas bawah bobot inersia

$t_{\max}$ = Iterasi maksimal

t = Iterasi saat ini

- f. Memperbarui Kecepatan Partikel
- g. Memperbarui Posisi Partikel
- h. Mengulang langkah 1-8 hingga didapat kondisi konvergen (maksimal iterasi)
- i. Menentukan hasil optimal parameter SVR dan jumlah fitur pilihan dari nilai *gBest*

3. Akurasi Hasil Peramalan

Ukuran akurasi hasil peramalan pada penelitian ini menggunakan *Mean Absolute Percentage Error* (MAPE), yaitu metrik yang digunakan dalam analisis peramalan untuk mengukur sejauh mana peramalan atau prediksi deviasi dan nilai aktual dalam bentuk persentase rata-rata[33]. MAPE diaplikasikan untuk menghitung keakuratan nilai prediksi atau peramalan suatu data, dengan memperoleh selisih nilai aktual dan nilai prediksi[34]. Semakin kecil nilai MAPE menunjukkan semakin akurat hasil peramalan[33]. Apabila nilai MAPE menunjukkan angka kurang dari 20% maka peramalan dinyatakan baik sehingga dapat dilanjutkan untuk dilakukan penarikan kesimpulan[35][34]. Berikut adalah kriteria nilai MAPE yang ditunjukkan pada Tabel 1.

Tabel 1. Kriteria Nilai MAPE [35]

Nilai MAPE	Interpretasi
< 10%	Perkiraan yang sangat akurat
10 – 20 %	Perkiraan yang baik
20 – 50 %	Perkiraan yang masuk akal
> 50%	Perkiraan yang tidak akurat

Untuk mengukur nilai kesalahan dengan menggunakan *Mean Absolute Percentage Error* (MAPE) dapat dihitung dengan menggunakan persamaan sebagai berikut.

$$MAPE = \left(\frac{100}{n} \right) \sum \left| A_t - \frac{F_t}{A_t} \right| \quad (12)$$

Sumber: [2]

Keterangan :

MAPE = *Mean Absolute Percentage Error*

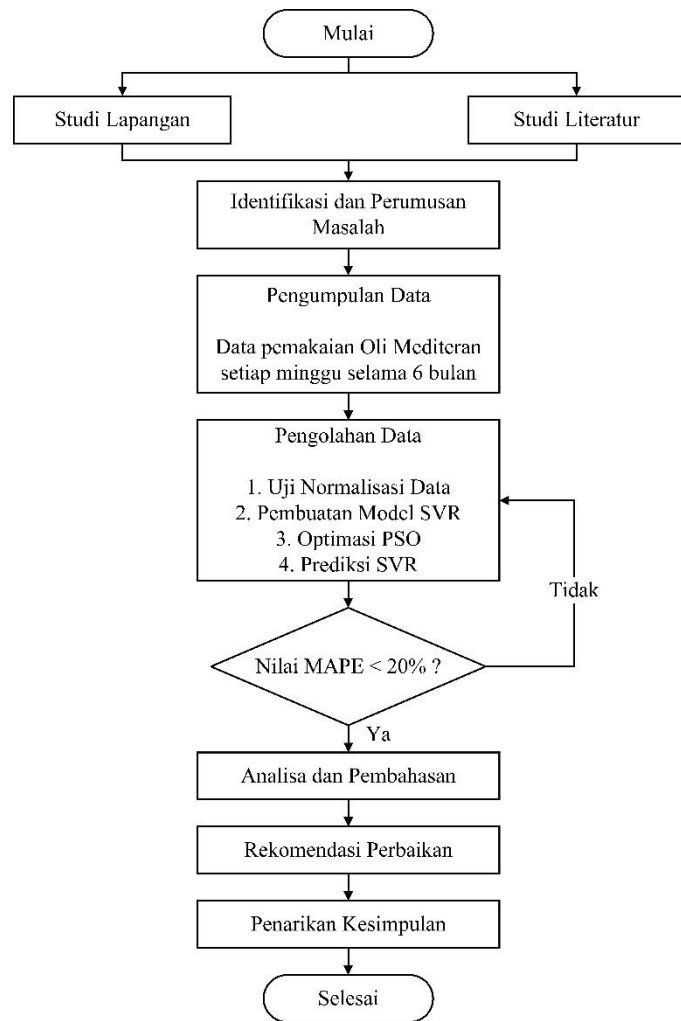
n = Jumlah data

A_t = Permintaan aktual

F_t = Permintaan *forecast*

B. Alur Penelitian

Berikut ini merupakan langkah-langkah penelitian yang digambarkan pada diagram alir yang ditunjukkan pada Gambar 1.

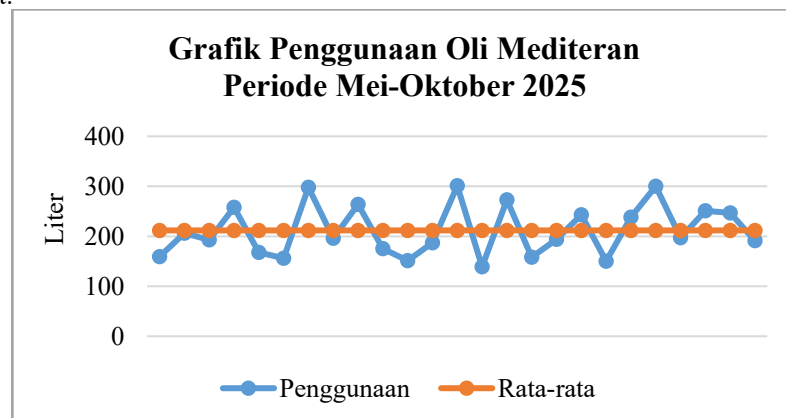


Gambar 1. Diagram Alir Penelitian

III. HASIL DAN PEMBAHASAN

A. Uji Normalisasi Data

Dalam penelitian ini menggunakan data sekunder yang didapatkan dari *data base* perusahaan. Data tersebut yaitu data pemakaian oli setiap minggunya di salah satu gudang milik PT. XYZ dalam kurun waktu 6 (enam) bulan yang dimulai dari bulan Mei hingga Oktober tahun 2025. Rincian data tersebut ditunjukkan dalam grafik penggunaan oli pada Gambar 2 berikut.



Gambar 2. Grafik Penggunaan Oli Mediteran Periode Mei-Oktober 2025

Dari Gambar 2 di dapat diketahui bahwa penggunaan oli selalu mengalami fluktuasi pada setiap minggunya. Data yang digunakan untuk melakukan peramalan pada penelitian ini berjumlah 25 data, yang diuraikan menjadi empat bagian yaitu data T-3 yang menunjukkan data tiga minggu sebelumnya, data T-2 sebagai data dua minggu sebelumnya, data T-1 sebagai data satu minggu sebelumnya dan data aktual yaitu data yang digunakan saat ini. Sebelum dilakukan seluruh rangkaian perhitungan peramalan, terlebih dahulu dilakukan normalisasi data menggunakan persamaan 1. Berikut adalah hasil normalisasi data yang dapat dilihat pada Tabel 2, di mana untuk kolom F1, F2, dan F3 merupakan atribut peramalan dan kolom aktual adalah hasil normalisasi data untuk data aktual.

Tabel 2. Uji Normalisasi Data

Data	Periode	F1	F2	F3	Aktual
1	Mei Minggu ke-1	0,9630	0,2099	0,2840	0,1235
2	Mei Minggu ke-2	0,2099	0,2840	0,1235	0,4136
3	Mei Minggu ke-3	0,2840	0,1235	0,4136	0,3333
...	...	...	...	...	...
25	November Minggu ke-1	0,3580	0,6914	0,6667	0,3210

B. Pembuatan Model SVR

Pada tahap ini dilakukan pembuatan model peramalan *Support Vector Regression* (SVR) dengan langkah-langkah sebagai berikut:

1. *K-Fold Cross Validation*

Selanjutnya yaitu dilakukan perhitungan *k-fold cross validation*, yaitu proses mempersiapkan data yang akan digunakan untuk melatih SVR dengan cara mengelompokkan antara data latih dan data uji dari keseluruhan data. Pada proses ini dilakukan pendefinisian data dengan membagi data latih (*data training*) dan data uji (*data testing*) sebanyak K dan mengulangi proses latihan sebanyak K. Berikut adalah hasil pembagian data yang disajikan pada Tabel 3, di mana data untuk pengolahan telah dibagi menjadi 5 *fold* (skenario) dengan indeks masing-masing yang berbeda.

Tabel 3. Pembagian Data Uji dan Data Latih

Skenario	Data Testing	Data Training
K = 1	Indeks 0-4	Indeks 5-24
K = 2	Indeks 5-9	Indeks 0-4 dan 10-24
K = 3	Indeks 10-14	Indeks 0-9 dan 15-24
K = 4	Indeks 15-19	Indeks 0-14 dan 20-24
K = 5	Indeks 20-24	Indeks 0-19

Berdasarkan Tabel 3 di atas dapat dilihat bahwa data uji (*data testing*) dan data latih (*data training*) dibagi menjadi dua proporsi yang berbeda. Diketahui dari tabel tersebut bahwa jumlah data uji lebih sedikit dari jumlah data latih dengan proporsi data uji sebesar 20% dan data latih sebesar 80%. Hal ini dikarenakan data uji merupakan data yang akan digunakan sebagai acuan dalam menentukan peramalan, sedangkan data latih adalah data yang digunakan untuk mempelajari pola dari data historis yang telah ada, terlebih lagi untuk kasus data non-linier seperti pada penelitian ini.

2. Penetapan Nilai Partikel dan Parameter

Sebelum melanjutkan pada proses inialisasi partikel, terlebih dahulu perlu dilakukan penetapan nilai sebagai batas dari pengolahan data yang akan dilakukan. Penetapan nilai pertama ini dilakukan dengan mengacu pada penelitian Angela (2024) yang telah memperoleh hasil peramalan yang sangat akurat.

1. Jumlah partikel = 3
2. Jumlah iterasi maksimum = 50
3. Nilai bobot inersia batas bawah ($w_{\min}$) = 0,4 dan batas atas ($w_{\max}$) = 0,8
4. Nilai C1 dan C2 = 1
5. Nilai batas bawah dan batas atas ruang pencarian tiap dimensi partikel:
 - a. Dimensi C ($x_{\min}, 1$) = 0,01 dan ($x_{\max}, 1$) = 1000
 - b. Dimensi ε ($x_{\min}, 2$) = 0,01 dan ($x_{\max}, 2$) = 0,09
 - c. Dimensi σ ($x_{\min}, 3$) = 0,01 dan ($x_{\max}, 3$) = 10
 - d. Dimensi cLR ($x_{\min}, 4$) = 0,01 dan ($x_{\max}, 4$) = 2
 - e. Dimensi λ ($x_{\min}, 5$) = 0,01 dan ($x_{\max}, 5$) = 5

3. Inialisasi Partikel

Proses inialisasi posisi partikel yaitu proses mewakili setiap partikel dengan nilai C, ε , σ , cLR, dan λ yang dihitung berdasarkan batas bawah dan batas atas ruang pencarian tiap dimensi partikel yang telah ditentukan pada proses penetapan nilai. Proses ini dilakukan sebanyak jumlah partikel yaitu 3 kali. Pada proses ini dimensi partikel terbagi menjadi dua, yaitu dimensi kontinu yang berjumlah 5 (C, ε , σ , cLR, dan λ) yang mewakili solusi sebagai nilai parameter optimal dan dimensi biner berjumlah 3 (F1, F2, dan F3) yang selanjutnya mewakili solusi sebagai fitur

yang dipilih. Berikut adalah perhitungan inisialisasi partikel yang telah dilakukan menggunakan persamaan 10 yang disajikan pada Tabel 4.

Tabel 4. Inisialisasi Partikel

X	C	ε	σ	cLR	λ	F1	F2	F3
1	328,3279	0,0399	5,5198	1,7243	0,6502	1	1	0
2	961,2476	0,0133	2,0424	3,9132	0,3644	0	1	1
3	849,0079	0,0564	2,2829	2,2771	1,5674	1	1	0

$$x(C_1) = x_{\min,C} + r_1 (x_{\max,C} - x_{\min,C}) = 0,01 + 0,32832 (1000 - 0,01) = 328,3279$$

Pada Tabel 4 nilai X menunjukkan jumlah partikel dan seluruh hasil pada dimensi kontinu didapat dari perhitungan menggunakan persamaan tersebut. Diketahui *particle* (individu) diwakili oleh parameter dimensi kontinu yaitu C, ε , σ , cLR, dan λ . Sedangkan *swarm* (populasi) ditunjukkan oleh nilai X1, X2 dan X3 sehingga terdapat tiga populasi pada perhitungan. Nilai r_1 pada persamaan di atas didapat dari angka *random* pada *range* 0 – 1. Dimensi biner diinisialisasi dengan nilai 0 dan 1, yang diartikan bahwa variabel dengan nilai 1 adalah variabel yang dipilih untuk dilakukan pengolahan selanjutnya dan variabel dengan nilai 0 adalah variabel yang tidak dipilih. Pada tahap ini inisialisasi dimensi biner masih dilakukan secara acak (*random*).

4. Perhitungan Sequential Learning

Proses *sequential learning* bertujuan mencari nilai untuk pengujian pada model regresi. Proses ini dimulai dengan menginisialisasi nilai α_i dan α^*_i dengan 0, pembentukan matriks X, membangun model regresi, menghitung *learning rate*, menghitung nilai *error*, menghitung $\delta\alpha_i$ dan $\delta\alpha^*_i$, dan memperbarui nilai α_i dan α^*_i . Proses ini berhenti ketika jumlah iterasi mencapai jumlah maksimal yang telah didefinisikan atau *error* lebih kecil dari nilai ε . Berikut adalah hasil dari pembentukan matriks X pada partikel ke-1 iterasi 0 dan *fold* K = 1 yang disajikan pada Tabel 5.

Tabel 5. Jarak Antar Data Latih

X	1	2	3	...	20
1	0,0000	0,4696	0,4202	...	0,0025
2	0,4696	0,0000	0,3141	...	0,4043
3	0,4202	0,3141	0,0000	...	0,3759
...	...	...	...	...	...
20	0,0025	0,4043	0,3759	...	0,0000

$$\|x_1 - x_1\|^2 = (F1_{1,5} - F1_{1,5})^2 + (F2_{1,5} - F2_{1,5})^2 = (0,333 - 0,333)^2 + (0,735 - 0,735)^2 = 0,000$$

Pada Tabel 5 menunjukkan hasil pembentukan matriks X yaitu matriks yang berisi perhitungan jarak antar data latih. Kemudian isi matriks tersebut digunakan untuk membangun model regresi (matriks *hessian*) yang dikombinasikan dengan Kernel *Radial Basis Function* (RBF) menggunakan persamaan 3. Disajikan pada Tabel 6 sebagai berikut.

Tabel 6. Pembangunan Model Regresi

X	1	2	3	...	20
1	1,422721	1,415044	1,415848	...	1,422680
2	1,415044	1,422721	1,417579	...	1,416108
3	1,415848	1,417579	1,422721	...	1,416571
...	...	...	...	...	...
20	1,422680	1,416108	1,416570	...	1,422720

$$[R]_{11} = \exp\left(-\frac{\|x_1 - x_1\|^2}{2\sigma^2}\right) + \lambda^2 = \exp\left(-\frac{0,000}{2(5,5198)^2}\right) + 0,6502^2 = 1,422721$$

Pada Tabel 6 menunjukkan hasil perhitungan matriks model regresi menggunakan persamaan yang telah dicantumkan di atas. Nilai $\|x_1 - x_1\|^2$ menunjukkan hasil perhitungan jarak antar data latih, yaitu hasil dari matriks X pada Tabel 5. Selanjutnya dilakukan perhitungan *learning rate* menggunakan persamaan sebagai berikut.

$$\begin{aligned} \gamma &= \frac{\text{cLR}}{\text{MAX (diagonal matriks regresi)}} \\ &= \frac{1,7243}{\text{MAX (1,422721, 1,415044, ..., 1,422720)}} \\ &= 1,2120 \end{aligned}$$

Selanjutnya dilakukan perhitungan nilai *error* menggunakan persamaan 4 pada partikel ke-1 iterasi 0 dan K = 1 yang hasil perhitungannya dapat dilihat pada Tabel 7.

Tabel 7. Nilai Error

Data ke-	1	2	3	...	20
E _i	0,1049	0,9815	0,3519	...	0,3210

$$\begin{aligned}
 E_1 &= Y_1 - \sum_{j=1}^n (\alpha_j^* - \alpha_j) R_{1j} \\
 &= 0,1049 - (((0-0)*1,422721) + ((0-0)*1,415044) + ((0-0)*1,415848) + \dots + ((0-0)*1,422680)) \\
 &= 0,1049
 \end{aligned}$$

Pada Tabel 7 seluruh perhitungan dilakukan menggunakan persamaan tersebut hingga pada data ke-20. Nilai α_j^* dan α_j selalu bernilai 0 dikarenakan hasil inisialisasi awal seperti yang telah dijelaskan sebelumnya. Nilai Y merupakan nilai data aktual pada data latih yang telah dinormalisasi. Selanjutnya yaitu menghitung nilai $\delta\alpha_i$ dan $\delta\alpha_i^*$ menggunakan persamaan sebagai berikut pada perhitungan partikel ke-1 iterasi 0 dan *fold* K = 1 yang disajikan pada Tabel 8.

Tabel 8. Nilai $\delta\alpha_i$ dan $\delta\alpha_i^*$

Data ke-	1	2	3	...	20
$\delta\alpha^*$	0,0788	1,1412	0,3781	...	0,3407
$\delta\alpha$	0,0000	0,0000	0,0000	...	0,0000

$$\begin{aligned}
 \delta\alpha_1^* &= \min\{\max[\gamma(E_1 - \varepsilon), -\alpha_1^*], C - \alpha_1^*\} \\
 &= \min\{\max[1,2120(0,1049 - 0,0399), -0], 328,3279 - 0\} \\
 &= 0,0788 \\
 \delta\alpha_1 &= \min\{\max[\gamma(-E_1 - \varepsilon), -\alpha_1], C - \alpha_1\} \\
 &= \min\{\max[1,2120(-0,1049 - 0,0399), -0], 328,3279 - 0\} \\
 &= 0
 \end{aligned}$$

Hasil perhitungan pada Tabel 8 selanjutnya digunakan untuk melakukan perhitungan pembaharuan nilai α_i dan α_i^* menggunakan persamaan sebagai berikut pada perhitungan partikel ke-1 iterasi 0 dan *fold* K = 1 yang disajikan pada Tabel 9.

Tabel 9. Nilai α_i dan α_i^*

Data ke-	1	2	3	...	20
α_i^*	0,0788	1,1412	0,3781	...	0,3407
α_i	0,0000	0,0000	0,0000	...	0,0000

$$\begin{aligned}
 \alpha_1^* &= \alpha_1^* + \delta\alpha_1^* = 0 + 0,0788 = 0,0788 \\
 \alpha_1 &= \alpha_1 + \delta\alpha_1 = 0 + 0 = 0
 \end{aligned}$$

Tabel 9 merupakan hasil pembaharuan nilai α_i^* dan α_i yang sebelumnya masih diinisialisasikan dengan nilai 0. Dari hasil perhitungan pada Tabel 9 diasumsikan bahwa pada tahap ini telah mencapai kondisi terminasi, sehingga nilai hasil perhitungan α_i^* dan α_i dapat digunakan dalam menguji model regresi.

5. Pengujian Model Regresi

Sebelum dilakukan pengujian model regresi maka terlebih dahulu dilakukan perhitungan jarak data latih dengan data uji dan pembentukan matriks kernel berdasarkan matriks jarak data uji-latih. Berikut adalah matriks jarak data uji latih pada perhitungan partikel ke-1 iterasi 0 dan *fold* K = 1 yang disajikan pada Tabel 10.

Tabel 10. Jarak Data Uji dan Data Latih

Uji/Latih	1	2	3	...	20
1	0,6717	0,0531	0,6256	...	0,5978
2	0,2183	0,2863	0,0330	...	0,1879
3	0,3759	0,2061	0,0114	...	0,3280
4	0,1471	0,4285	0,0983	...	0,1322
5	0,1674	0,1268	0,1072	...	0,1313

Tabel 10 menunjukkan hasil perhitungan data uji dan data latih dengan menggunakan cara yang sama seperti pada perhitungan jarak antara data latih. Selanjutnya dari hasil tersebut dapat digunakan untuk membuat matriks kernel. Berikut adalah matriks kernel pada perhitungan partikel ke-1 iterasi 0 dan *fold* K = 1 yang disajikan pada Tabel 11.

Tabel 11. Matriks Kernel

X	1	2	3	...	20
1	1,4118	1,4218	1,4125	...	1,4130
2	1,4191	1,4180	1,4222	...	1,4196

3	1,4166	1,4193	1,4225	...	1,4174
4	1,4203	1,4157	1,4211	...	1,4206
5	1,4200	1,4206	1,4210	...	1,4206

Tabel 11 menunjukkan hasil perhitungan matriks kernel dengan cara yang sama seperti pada perhitungan pembaguan matriks model regresi menggunakan persamaan 3. Selanjutnya hasil tersebut digunakan untuk melakukan pengujian regresi menggunakan persamaan 2. Berikut adalah hasil dari perhitungan regresi, denormalisasi dan nilai aktual partikel ke-1 iterasi 0 dan *fold* K = 1 yang disajikan pada Tabel 12.

Tabel 12. Nilai F(x), Denormalisasi dan Data Aktual

Uji	F(x)	Denormalisasi	Aktual
1	14,8774	2549,144	159
2	14,9088	2554,226	206
3	14,9008	2552,929	193
4	14,9043	2553,500	258
5	14,9235	2556,603	168

$$\begin{aligned}
 \text{Uji 1} &= (X_1' * (X_{\max} - X_{\min}) + X_{\min}) \\
 &= (14,8774 * (301 - 139) + 139) \\
 &= 2549,144
 \end{aligned}$$

Tabel 12 menunjukkan bahwa hasil denormalisasi sangat menjauhi nilai aktual, hal ini dapat terjadi karena parameter SVR belum mencapai posisi optimum. Sehingga nilai MAPE yang dihasilkan menjadi sangat tinggi. Oleh karena itu selanjutnya perlu dilakukan pencarian posisi terbaik partikel menggunakan Algoritma PSO. Namun sebelum itu dilakukan perhitungan nilai *error* pada masing-masing *K-fold* yang disajikan pada Tabel 13 sebagai berikut.

Tabel 13. Nilai Cost

<i>Fold</i>	<i>Error</i>
1	11,77
2	10,07
3	11,14
4	11,92
5	9,60
<i>Cost</i>	10,90

Dari Tabel 13 nilai *cost* didapatkan dari hasil rata-rata *error* dari *fold* ke-1 hingga *fold* ke-5. Dari tabel tersebut juga dapat diketahui bahwa nilai *cost* yang dihasilkan dari perhitungan SVR menggunakan posisi partikel acak menghasilkan nilai yang sangat tinggi. Nilai *cost* ini selanjutnya akan digunakan sebagai *input* pada tabel posisi terbaik individu yang kemudian dibandingkan dengan nilai dari partikel lain.

6. Pembaruan Posisi dan Kecepatan Partikel

Langkah pertama yang dilakukan ialah pencarian posisi individu terbaik (*pBest*) dan posisi terbaik global (*gBest*). Parameter yang disajikan adalah hasil inisialisasi partikel pada perhitungan sebelumnya. Berikut adalah hasil posisi terbaik individu beserta nilai *cost* yang dihasilkan dan disajikan pada Tabel 14.

Tabel 14. Posisi Terbaik Individu Iterasi 0

<i>pBest</i>	<i>C</i>	ϵ	σ	<i>cLR</i>	λ	<i>F1</i>	<i>F2</i>	<i>F3</i>	<i>Cost</i>
1	328,3279	0,0399	5,5198	1,7243	0,6502	1	1	0	10,90
2	961,2476	0,0133	2,0424	3,9132	0,3644	0	1	1	24,78
3	849,0079	0,0564	2,2829	2,2771	1,5674	1	1	0	13,33

Pada Tabel 14 diketahui bahwa partikel 1 memiliki nilai *cost* yang paling rendah di antara partikel yang lain yaitu sebesar 10,90, sehingga dengan demikian partikel 1 dinyatakan sebagai posisi terbaik global pada iterasi 0. Selanjutnya dilakukan perhitungan bobot inersia iterasi ke-0 menggunakan persamaan 11 dan didapatkan hasil sebesar 0,8. Kemudian dilakukan perhitungan kecepatan partikel menggunakan persamaan berikut.

$$\begin{aligned}
 V_{1C}(1) &= WV_{11}(0) + c_1 r_1(0) [y_{11}(0) - x_{11}(0)] + c_2 r_2(0) [y_{11}(0) - x_{11}(0)] \\
 &= 0,8 \times 0 + 1 \times 0,6054 [328,3279 - 328,3279] + 1 \times 0,5098 [328,3279 - 328,3279] \\
 &= 0
 \end{aligned}$$

Dari persama tersebut nilai r_1 dan r_2 merupakan angka acak yang dipilih pada *range* 0 – 1. Selanjutnya berikut adalah hasil pembaharuan kecepatan partikel yang disajikan pada Tabel 15.

Tabel 15. Kecepatan Baru Partikel Iterasi 0

X	C	ε	σ	cLR	λ	F1	F2	F3
1	0,0000	0,0000	0,0000	0,0000	0,0000	0,5000	0,5000	0,5000
2	-705,8320	0,0297	3,8780	-2,4411	0,3187	0,7310	0,5000	0,2689
3	-580,6624	-0,0184	3,6097	-0,6165	-1,0228	0,5000	0,5000	0,5000

Pada Tabel 15 menunjukkan nilai kecepatan baru partikel yang selanjutnya akan digunakan dalam pembaruan posisi partikel dengan memperbarui posisi pada dimensi partikel yang bernilai kontinu dengan menggunakan persamaan sebagai berikut.

$$X_1(1) = X_1(0) + V_1(1) = 328,3279 + 0 = 328,3279$$

Selanjutnya dilakukan pengecekan apakah posisi baru partikel berada dalam ruang pencarian. Proses ini dilakukan dengan cara membandingkan nilai posisi baru partikel dengan nilai batas bawah dan atas dimensi partikel. Berdasarkan perhitungan yang dilakukan diketahui bahwa posisi baru partikel telah berada dalam batas ruang pencarian. Sedangkan untuk pembaharuan partikel bernilai diskrit (dimensi biner) terlebih dahulu dilakukan pembangkitan nilai *random* dalam *range* 0 – 1. Sehingga berikut adalah posisi baru dimensi partikel yang disajikan pada Tabel 16.

Tabel 16. Posisi Baru Partikel Iterasi 0

X	C	ε	σ	cLR	λ	F1	F2	F3	Cost
1	328,3279	0,0399	5,5198	1,7243	0,6502	1	0	1	10,90
2	360,7334	0,0385	5,3417	1,8363	0,6355	1	1	0	24,78
3	354,9867	0,0407	5,3541	1,7526	0,6971	1	0	1	13,33

Tabel 16 menunjukkan posisi baru partikel dengan nilai variabel diskrit atau dimensi biner yang berbeda dari hasil inisialisasi sebelumnya. Seluruh tahap di atas diulang sebanyak *max* iterasi. Berdasarkan tabel tersebut menunjukkan bahwa partikel 1 merupakan posisi terbaik global yang didapatkan dari perhitungan dengan inisialisasi partikel acak dengan nilai *cost* paling minimum di antara ketiga partikel.

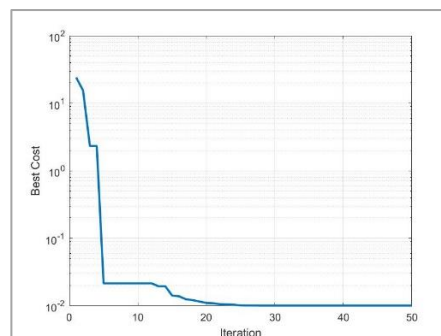
C. Optimasi PSO

Untuk mendukung terbentuknya nilai *cost* yang minimum dari penyelesaian *gBest*, maka digunakan *software* MATLAB R2016a sebagai skenario pengujian algoritma PSO. Dalam penyelesaiannya menggunakan *input coding* yang menghasilkan *output* nilai *cost* dari iterasi 1 hingga 50 yang hasil perhitungannya disajikan pada Tabel 17 berikut.

Tabel 17. Nilai *Cost* Iterasi 1 hingga 50

<i>gBest</i>	<i>Cost</i>
1	24,03988
2	15,42977
3	2,32748
...	...
50	0,01006

Pada Tabel 17, nilai *cost* pada setiap iterasi diperoleh dari hasil evaluasi fungsi *fitness* yang dihitung berdasarkan tingkat kesalahan prediksi model SVR pada iterasi 0 dan parameter (C, ε , σ , cLR, λ) yang dibawa oleh setiap partikel digunakan untuk membangun model SVR. Berdasarkan hasil perhitungan yang telah dilakukan pada Tabel 17, diketahui bahwa kombinasi parameter SVR yang dihasilkan nilai *cost* yang mengalami perubahan. Nilai *cost* yang dihasilkan menggunakan optimasi PSO saat iterasi terakhir adalah 0,01006 yang mana menunjukkan bahwa telah didapat nilai *cost* paling minimum. Berikut adalah grafik *output* hasil optimasi Algoritma PSO menggunakan MATLAB.

**Gambar 3.** Grafik Optimasi PSO - MATLAB

Dalam *output* grafik pada Gambar 3 dapat dilihat bahwa partikel menghasilkan nilai *cost* yang stabil pada iterasi 30 hingga 50. Ini menunjukkan bahwa model regresi yang terbentuk telah cukup akurat untuk memprediksi kebutuhan oli. Berikut adalah posisi terbaik global partikel terbaru yang dihasilkan dari optimasi PSO menggunakan MATLAB yang disajikan pada Tabel 18.

Tabel 18. Posisi Terbaik Global Optimum

X	C	ε	σ	cLR	λ	F1	F2	F3
1	199,9993	0,0744	0,2487	0,1954	0,003	1	1	0

Pada Tabel 18, nilai parameter optimum didapatkan dari hasil optimasi PSO menggunakan *software* MATLAB yang kemudian akan digunakan pada perhitungan prediksi SVR sebagai acuan posisi partikel pada proses *sequential learning* selanjutnya. Dengan nilai partikel terbaru ini diharapkan peramalan dapat menghasilkan nilai MAPE yang kecil yaitu di bawah 20%.

D. Prediksi SVR

Pada proses ini dibentuk ulang data uji dan data latih dengan jumlah proporsi data uji sebesar 40% dan data latih sebesar 60% dari total 25 data yang telah dinormalisasi. Berikut adalah data uji yang akan digunakan pada peramalan SVR setelah optimasi PSO pada perhitungan *fold* K = 1 yang disajikan pada Tabel 19.

Tabel 19. Data Uji (Data Testing)

Indeks	F1	F2	F3	Aktual
0	0,9630	0,2099	0,2840	0,1235
1	0,2099	0,2840	0,1235	0,4136
2	0,2840	0,1235	0,4136	0,3333
...	...	...	...	...
9	0,9815	0,3519	0,7716	0,2222

Pada Tabel 19 diketahui bahwa jumlah data yang menjadi data uji adalah sebanyak 10 data dari indeks 0 hingga indeks 9. Selanjutnya berikut adalah data latih yang digunakan untuk peramalan SVR-PSO pada perhitungan *fold* K=1 yang disajikan pada Tabel 20.

Tabel 20. Data Latih (Data Training)

Indeks	F1	F2	F3	Aktual
10	0,3519	0,7716	0,2222	0,0741
11	0,7716	0,2222	0,0741	0,2963
12	0,2222	0,0741	0,2963	1,0000
...	...	...	...	...
24	0,3580	0,6914	0,6667	0,3210

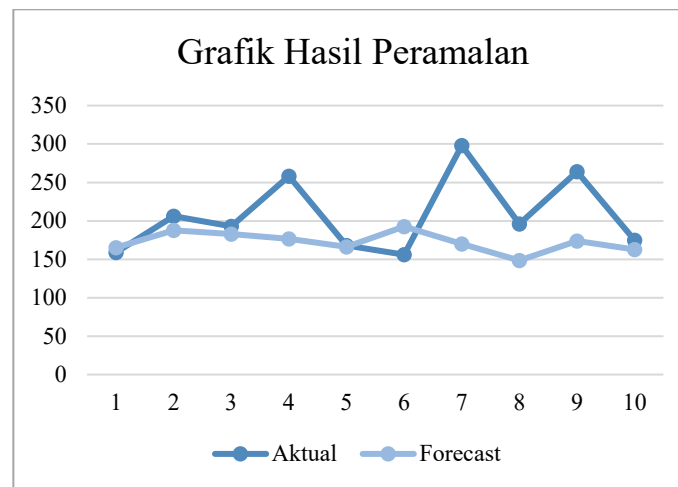
Pada Tabel 20 diketahui bahwa jumlah data yang dipilih sebagai data latih adalah sebanyak 15 data dari indeks 10 hingga indeks 24. Selanjutnya dilakukan prediksi atau peramalan dengan langkah yang sama seperti pada langkah sebelumnya, mulai dari proses *sequential learning* hingga perhitungan MAPE. Peramalan dilakukan menggunakan parameter yang telah didapat dari optimasi PSO pada langkah sebelumnya, yaitu parameter C = 199,9993, ε = 0,0744, σ = 0,2487, cLR = 0,1954 dan λ = 0,003. Berikut adalah hasil peramalan yang didapatkan setelah dilakukan optimasi menggunakan algoritma PSO yang ditunjukkan pada Tabel 21 di bawah ini.

Tabel 21. Hasil Peramalan PSO-SVR

Uji	Aktual	Forecast
1	159	165,19
2	206	187,67
3	193	182,87
4	258	176,59
5	168	166,27
6	156	192,61
7	298	170,15
8	196	148,46
9	264	173,84
10	175	162,74

Tabel 21 menunjukkan hasil peramalan yang didapatkan menggunakan metode SVR-PSO dengan posisi parameter terbaru. Diketahui pada gambar tersebut hasil peramalan cenderung mendekati nilai aktual dan bersifat fluktuatif atau

mengalami kenaikan dan penurunan nilai setiap periodenya seperti pada kondisi aktual suatu permintaan (*demand*) oli pada perusahaan. Dengan demikian, hasil peramalan ini dapat digunakan untuk mengetahui dan memperkirakan jumlah kebutuhan oli di periode mendatang sebagai dasar dalam perencanaan persediaan perusahaan. Berikut adalah grafik yang menunjukkan perbandingan data aktual dan data hasil peramalan yang telah dilakukan disajikan pada Gambar 4.



Gambar 4. Grafik Hasil Peramalan

Gambar 4 menunjukkan perbandingan nilai aktual dan hasil peramalan (*forecast*) pada 10 pengujian, di mana pola pergerakan data ramalan secara umum mengikuti tren data aktual meskipun terdapat selisih pada beberapa periode tertentu seperti periode 4 dan 6 yang belum sepenuhnya tertangkap oleh model. Namun secara keseluruhan kedekatan antara kedua pola mengindikasikan bahwa metode SVR-PSO mampu merepresentasikan pola fluktuasi permintaan oli dengan cukup baik. Selanjutnya adalah perhitungan nilai *Mean Absolute Percentage Error* (MAPE) yang didapatkan menggunakan persamaan 12 sebagai berikut.

$$\begin{aligned}
 \text{MAPE} &= \left(\frac{100}{n} \right) \sum \left| A_t - \frac{F_t}{A_t} \right| \\
 &= \left(\frac{100}{10} \right) \left[\left| 159 - \frac{165}{159} \right| + \dots + \left| 175 - \frac{163}{175} \right| \right] \\
 &= 18,24 \%
 \end{aligned}$$

Dari hasil MAPE pada perhitungan di atas dapat dilihat bahwa tingkat kesalahan telah mencapai kondisi minimum yaitu di bawah 20%. Berdasarkan interpretasi nilai MAPE pada Tabel 1 menunjukkan bahwa peramalan yang dilakukan dengan hasil nilai MAPE tersebut dinyatakan cukup baik, terlebih untuk kasus data non linier seperti pada penelitian ini. Dengan demikian, hasil peramalan yang diperoleh tidak hanya menghasilkan tingkat kesalahan yang minimum, tetapi juga dapat memberikan prediksi yang cukup akurat untuk dijadikan sebagai dasar penentuan jumlah stok aman (*safety stock*) dan jumlah pemesanan optimum bagi perusahaan.

E. Rekomendasi Perbaikan

Berdasarkan penelitian yang telah dilakukan, berikut merupakan beberapa rekomendasi perbaikan yang diberikan kepada perusahaan untuk meningkatkan kualitas pengambilan keputusan dalam pengelolaan persediaan di gudang.

1. Menetapkan jumlah stok aman (*safety stock*) di gudang berdasarkan hasil perhitungan peramalan yang telah dilakukan. Perusahaan juga dapat menetapkan *safety stock* pada item barang lain yang ada di gudang selain oli, baik barang jenis *spare part* maupun *consumable part* seperti *bearing*, kawat las, filter, kampas rem dan *v-belt*.
2. Menjadikan data hasil peramalan sebagai dasar pertimbangan dalam penyusunan Purchase Order (PO) atau perencanaan pembelian oli mingguan, sehingga proses pengadaan dapat dilakukan secara terukur dan sesuai dengan estimasi kebutuhan pada periode mendatang.
3. Melakukan pembaruan dan pelatihan model secara berkala dengan data terbaru agar model adaptif terhadap tren baru dan akurasi tetap terjaga. Selain itu juga perlu dilakukan evaluasi secara berkala berdasarkan metrik *error* (MAPE, MSE atau RMSE) untuk perbaikan berkelanjutan.

IV. SIMPULAN

Berdasarkan peramalan yang telah dilakukan menggunakan *Support Vector Regression* (SVR) didapatkan nilai MAPE sebesar 18,24% yang menunjukkan bahwa peramalan yang dilakukan cukup baik. Hasil peramalan menunjukkan kuantitas permintaan oli pada 10 periode berikutnya yaitu 165, 188, 183, 177, 166, 193, 170, 148, 174 dan 163 dalam satuan liter. Nilai MAPE yang minimum ini didapatkan dari optimasi algoritma *Particle Swarm Optimization* (PSO)

menggunakan bantuan *software* MATLAB R2016a untuk mencari posisi terbaik partikel agar didapatkan *cost* paling minimum. Dari hasil optimasi menggunakan PSO didapatkan parameter optimal yaitu $C = 199,9993$, $\varepsilon = 0,0744$, $\sigma = 0,2487$, $cLR = 0,1954$ dan $\lambda = 0,003$. Peramalan akhir ini menggunakan pilihan fitur F1 dan F3 dengan proporsi data uji sebesar 40% dan data latih sebesar 60% yang mana pada pilihan fitur dan proporsi ini setelah dilakukan perhitungan secara keseluruhan dapat menghasilkan nilai MAPE paling minimum. Dengan penelitian ini diharapkan dapat digunakan oleh perusahaan sebagai acuan yang baik dalam menentukan stok aman dan jumlah pemesanan optimum oli.

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada Universitas Muhammadiyah Sidoarjo dan perusahaan terkait yang telah bersedia menjadi objek penelitian ini.

REFERENSI

- [1] R. N. Aini and E. Aryanny, "Policy Making in Optimizing Inventory Control with Continous and Periodic Review Method at PT . XYZ," *Prozima*, vol. 5, no. 2, pp. 93–99, 2021.
- [2] A. H. Nasution and Y. Prasetyawan, *Perencanaan dan Pengendalian Produksi*. Yogyakarta: Graha Ilmu, 2008.
- [3] R. K. Dash, T. N. Nguyen, K. Cengiz, and A. Sharma, "Fine-Tuned Support Vector Regression Model for Stock Predictions," *Neural Comput. Appl.*, vol. 35, no. 32, 2021.
- [4] W. Budiharto, *Machine Learning & Computational Intelligence*. Yogyakarta: CV Andi, 2016.
- [5] A. B. Raharjo, Z. Z. Dinanto, D. Sunaryono, and D. Purwitasari, "Prediksi Akumulasi Kasus Terkonfirmasi Covid-19 Di Indonesia Menggunakan Support Vector Regression," *Techno.Com*, vol. 20, no. 3, pp. 372–381, 2021.
- [6] B. Santosa, *Data Mining: Teknik Pemanfaatan Data Untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu, 2007.
- [7] N. Afifah *et al.*, "Perbandingan Kinerja Kernel RBF dan Linear pada Algoritma Support Vector Machine (SVM) untuk Prediksi Serangan Ransomware Locker," vol. 15, no. 1, pp. 3118–3123, 2023.
- [8] J. Zhu, J. Liu, Y. Chen, and X. Xue, "Binary Restructuring Particle Swarm Optimization and Its Application," vol. 2, no. 2, pp. 1–17, 2023.
- [9] V. I. Kontopoulou, A. D. Panagopoulos, and I. Kakkos, "A Review of ARIMA vs . Machine Learning Approaches for Time Series Forecasting in Data Driven Networks," vol. 15, no. 8, pp. 1–31, 2023.
- [10] F. C. Angela and T. Sukmono, "Fusing SVR with PSO Improves E-commerce Sales Prediction with 8.98% MAPE," *Indones. J. Innov. Stud.*, vol. 25, no. 2, pp. 1–14, 2024.
- [11] A. Umiyati, D. Dasari, and F. Agustina, "Peramalan Harga Batubara Acuan Menggunakan Metode PSOSVR dan IPSOSVR," *J. EurekaMatika*, vol. 9, no. 1, pp. 69–94, 2021.
- [12] M. R. Kusnaldi, T. Gulo, and S. Aripin, "Penerapan Normalisasi Data Dalam Mengelompokkan Data Mahasiswa Dengan Menggunakan Metode K-Means Untuk Menentukan Prioritas Bantuan Uang Kuliah Tunggal," *J. Comput. Syst. Informatics*, vol. 3, no. 4, pp. 330–338, 2022.
- [13] A. Harmain, Paiman, H. Kurniawan, Kusrini, and D. Maulina, "Normalisasi Data Untuk Efisiensi K-Means Pada Pengelompokan Wilayah Berpotensi Kebakaran Hutan Dan Lahan Berdasarkan Sebaran Titik Panas," *TEKNIMEDIA*, vol. 2, no. 2, pp. 83–89, 2021.
- [14] D. Haryadi, A. R. Hakim, D. M. U. Atmaja, and S. N. Yutia, "Implementation of Support Vector Regression for Polkadot Cryptocurrency Price Prediction," *Int. J. Informatics Vis.*, vol. 6, no. 1–2, pp. 201–207, 2022.
- [15] Jagadish, *Essential Concepts and Techniques of AI & ML*. Bopal: AGPHBooks, 2024.
- [16] S. Herawati and N. Prastiti, "Implementasi Metode Long Short Term Memory Untuk Peramalan Pertumbuhan Jumlah UMKM," *J. Simantec*, vol. 12, no. 2, pp. 63–70, 2024.
- [17] A. I. Sakti, D. Y. Dalimunthe, and I. Fahria, "Implementasi Elm Menggunakan Sigmoid Biner Untuk Prediksi Harga Cabai Rawit Di Provinsi Kepulauan Bangka Belitung," *J. Ilm. Mat.*, vol. 13, no. 02, pp. 367–373, 2025.
- [18] S. K. Wati, N. Imro'ah, and E. Sulistianingsih, "Peramalan Harga Penutupan Indeks Saham LQ45 Dengan Metode Generalized Regression Neural Network (GRNN)," *Bimaster*, vol. 14, no. 2, pp. 217–226, 2025.
- [19] P. Fiszeder and W. Orzeszko, "Covariance matrix forecasting using support vector regression," *Appl. Intell.*, vol. 51, pp. 7029–7042, 2021.
- [20] Z. Liao, S. Dai, and T. Kuosmanen, "Convex Support Vector Regression," *Eur. J. Oper. Res.*, vol. 313, no. 3, pp. 858–870, 2024.
- [21] S. I. Abba *et al.*, "Emerging Harris Hawks Optimization based load demand forecasting and optimal sizing of stand-alone hybrid renewable energy systems – A case study of Kano and Abuja , Nigeria," *Results Eng.*, vol. 12, no. 12, 2021.
- [22] B. Musa, N. Yimen, S. I. Abba, H. H. Adun, and M. Dagbasi, "Multi-State Load Demand Forecasting Using

- Hybridized Support Vector Regression Integrated with Optimal Design of Off-Grid Energy Systems — A Metaheuristic Approach,” *Processes*, vol. 9, no. 7, 2021.
- [23] R. T. Gorji, S. M. Hosseini, A. A. Abdoos, and A. Ebadi, “A Hybrid Intelligent Method for Compensation of Current Transformers Saturation Based on PSO-SVR,” *Period. Polytech. Electr. Eng. Comput. Sci.*, vol. 65, no. 1, pp. 53–61, 2021.
- [24] A. M. Hadi, W. Witanti, and Melina, “Prediksi Pergerakan Harga Emas Menggunakan Metode Genetic Support Vector Regression,” *J. Inform. Teknol. dan Sains*, vol. 6, no. 3, pp. 486–496, 2024.
- [25] P. Zhang, S. Mei, C. Shi, R. Xie, Y. Zhuo, and Y. Wang, “Forecasting DO of the river-type reservoirs using input variable selection and machine learning techniques - taking Shuikou reservoir in the Minjiang River as an example,” *Ecol. Indic.*, vol. 155, no. 6, 2023.
- [26] C. Puspitasari, N. Rokhman, and Wahyono, “Prediction Of Ozone (O3) Values Using Support Vector Regression Method,” *J. Inform. Polinema*, vol. 7, no. 4, pp. 81–88, 2021.
- [27] T. M. Shami, A. A. El-saleh, M. Alswaiti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, “Particle Swarm Optimization : A Comprehensive Survey,” *IEEE Access*, vol. 10, pp. 10031–10061, 2022.
- [28] J. B. E. Putry, A. T. Sasongko, and W. Hadikristanto, “Optimization of Decision Tree Using Particle Swarm Optimization for Credit Risk of KMG Bank DKI,” *Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 4, pp. 1403–1410, 2024.
- [29] B. N. Azmi, A. Hermawan, and D. Avianto, “Analisis Pengaruh Komposisi Data Training dan Data Testing pada Penggunaan PCA dan Algoritma Decision Tree untuk Klasifikasi Penderita Penyakit Liver,” *J. Teknol. Inf. dan Multimed.*, vol. 4, no. 4, pp. 281–290, 2023.
- [30] N. F. Fahrudin, K. R. Putra, S. Umaroh, and G. B. Lautan, “Influence of Data Scaling and Train / Test Split Ratios on LightGBM Efficacy for Obesity Rate Prediction,” *Multimed. Artif. Intell. Netw. Database J.*, vol. 9, no. 2, pp. 220–234, 2024.
- [31] A. P. Engelbrecht, *Computational Intelligence : An Introduction*. Chichester: John Wiley & Sons, Ltd, 2007.
- [32] X. Jiang, Y. Yue, Y. Min, Q. Zhang, and X. Gui, “Particle Swarm Optimization with Multiple Adaptive Subswarms,” *J. Phys. Conf. Ser.*, vol. 1757, 2021.
- [33] Z. Ngabidin, A. Sanwidi, and E. R. Arini, “Implementasi Metode Double Exponential Smoothing Brown Untuk Meramalkan Jumlah Penduduk Miskin,” vol. 11, no. 2, pp. 328–338, 2023.
- [34] R. N. Puspita, “Perbandingan Metode Double Exponential Smoothing Dan Triple Exponential Smoothing Pada Peramalan Nilai Ekspor Di Indonesia,” *Jembura J. Probab. Stat.*, vol. 3, no. 2, 2022.
- [35] C. D. Lewis, *Industrial and Business Forecasting Methods*. Kent: Butterworth & Co, Ltd, 1982.

Conflict of Interest Statement:

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.