

Peramalan Crude Palm Oil (CPO) Menggunakan *Support Vector Regression* Kernel Radial Basis

Rezzy Eko Caraka

School Of Mathematics, Faculty of Science and Technology, The National University of Malaysia, Bangi, Putra Jaya, Malaysia
Bioinformatics and Data Science Research Center, Bina Nusantara University, Anggrek Campus Room 700, Jakarta, Indonesia
e-mail: rezzyekocaraka@gmail.com / rezzy.caraka@binus.edu

Hasbi Yasin

Departemen Statistika Universitas Diponegoro, Semarang

Adi Waridi Basyiruddin

Departemen Statistika Universitas Diponegoro, Semarang

Abstract: Recently, instead of selecting a kernel has been proposed which uses SVR, where the weight of each kernel is optimized during training. Along this line of research, many pioneering kernel learning algorithms have been proposed. The use of kernels provides a powerful and principled approach to modeling nonlinear patterns through linear patterns in a feature space. Another benefit is that the design of kernels and linear methods can be decoupled, which greatly facilitates the modularity of machine learning methods. We perform experiments on real data sets crude palm oil prices for application and better illustration using kernel radial basis. We see that evaluation gives a good to fit prediction and actual also good values showing the validity and accuracy of the realized model based on MAPE and R^2 .

Keywords: Crude Palm Oil; Forecasting; SVR; Radial Basis; Kernel

1. Pendahuluan

Data sering memiliki beberapa keteraturan intrinsik yang jika terungkap, dapat memudahkan untuk memahami data tersebut atau bisa membuat prediksi tentang data baru dari sumber yang sama. Keteraturan ini disebut *patern* atau analisis pola, yang telah dipelajari secara luas seperti dalam statistik, arti dari konsep kecerdasan buatan atau (*artificial intelligence*) berkaitan dengan pendeteksian otomatis pada pola dalam data. Konsep kecerdasan buatan (*artificial intelligence*) berhubungan dengan *machine learning* dimana *machine learning* sebagai alat statistika (*statistical tools*) yang

digunakan dalam analisis data. Pada konsep ini peneliti telah banyak melakukan pembangunan sebuah metode baru yang dapat di aplikasikan di berbagai permasalahan. *Machine Learning* juga dikenal sebagai teori belajar berbasis komputasi, bertujuan untuk memahami prinsip-prinsip dasar pembelajaran sebagai proses komputasi. Bidang ini berusaha untuk memahami pada tingkat matematika dan statistika yang tepat berdasarkan kemampuan mendapatkan informasi secara fundamental mempelajari berbagai jenis Metode yang cepat, efisien, dan untuk memahami prinsip-prinsip algoritma dasar yang terlibat dalam mendapatkan informasi dari data dan untuk meningkatkan kinerja dari Metode tersebut. Tujuan dari teori ini untuk membantu dalam desain metode pembelajaran yang lebih baik dan memahami masalah-masalah mendasar dalam proses belajar itu sendiri Blum [4]. Pemodelan berbasis waktu (*time series*) menjadi populer dalam beberapa tahun terakhir. Menggunakan informasi historis untuk mengkarakteristik dan melakukan peramalan gerakan seri itu sendiri. Ada berbagai alasan belajar *time series*.

Prediksi masa depan berdasarkan masa lalu, pengendalian proses memproduksi seri data, Pemahaman dari mekanisme menghasilkan seri tersebut. Konsumsi kelapa sawit terus meningkat jika dibandingkan minyak nabati lainnya beberapa peneliti telah melakukan prediksi pada harga CPO dengan menggunakan metode yang berbeda. Mahmudah [11] memprediksi harga CPO menggunakan kombinasi Fuzzy Mamdani dan Evolution Strategis menghasilkan akurasi sebesar 95% pada TRD dan 91% pada CHD. Arshad [2] melakukan analisis CPO dengan menggunakan metode klasik Box-Jenkins ARIMA dan dilakukan uji asumsi pada metode tersebut didapatkan error yang cukup besar. Oleh karena itu dibutuhkan sebuah metode yang mampu untuk digunakan sebagai peramalan dan menghasilkan akurasi yang baik. Peramalan *time series* ini dimungkinkan karena masa depan tergantung pada masa lalu atau analog karena ada hubungan antara masa depan dan masa lalu. Namun hubungan ini tidak deterministik dan hampir tidak dapat ditulis dalam bentuk analitis Caraka [6] [7]. SVR adalah bagian dari teknik mesin belajar (*machine learning*). Secara umum, *support vector machine* membangun sebuah *hyperplane* atau seperangkat *hyperplanes* dalam dimensi ruang yang tinggi atau terbatas, yang dapat digunakan untuk klasifikasi, regresi, atau tugas-tugas lainnya.

Secara intuitif, sebuah pemisahan yang baik dicapai oleh *hyperplane* yang memiliki jarak terbesar dengan titik data *training* terdekat setiap kelas (disebut margin fungsional), karena pada umumnya semakin besar margin maka kesalahan classifier akan rendah. Algoritma SVM diusulkan oleh Vapnik dan Corinna Cortes pada tahun 1995. SVM merupakan metode yang memiliki keunggulan dalam optimasi sistem pengenalan pola dengan kemampuan generalisasi yang baik. SVM juga dapat digunakan untuk meramalkan suatu nilai, seperti yang telah diperkenalkan oleh Vapnik akhir tahun 1960 Santosa [12]. SVM berusaha menemukan *hyperplane* (garis pemisah)

terbaik antar kelas. Metode SVM dapat digunakan dalam kasus regresi dan data non-linier dengan jumlah input data yang besar, dalam kasus ini metode SVM yang digunakan adalah *Support Vector Regression* (SVR) dengan output berupa data kontinu. Konsep *loss function* pada SVR dapat digunakan untuk kasus regresi. Keuntungan dari SVR adalah: (a) SVR melatih secara signifikan lebih cepat karena hanya memecahkan set persamaan linear, maka lebih cocok untuk dataset dimensi tinggi, (b) SVR menghindari masalah minima lokal dari optimasi nonlinier, dan (c) ϵ -insensitive yang merupakan norm error yang digunakan oleh SVR untuk melakukan penalti outlier dalam data pelatihan. Schölkopf *et al* [13] [14] [15] menjelaskan *Support Vector* (SV) adalah sebuah metode, berdasarkan teori belajar statistik, yang memiliki konsep seperti jaringan saraf, dan fungsi radial basis (RBF) sebagai sebuah jaringan kasus khusus. Dalam kasus RBF, algoritma SV otomatis menentukan pusat, bobot dan *threshold* seperti untuk meminimalkan batas atas kesalahan pengujian yang diharapkan. Namun demikian, hasil yang sangat baik telah diperoleh dengan menggunakan SVR dalam waktu masalah seri SVR. Yasin *et al* [18] menggunakan SVR pada harga minyak mentah dengan algoritma *grid search-cross validation*. Pada penerapan kernel pada SVR yang dilakukan oleh Schölkopf *et al* [13] [14] [15] didapat sebuah hasil empiris, mereka mengklaim bahwa kernel RBF tidak hanya lebih mudah untuk diterapkan, tetapi juga mampu diaplikasikan dalam kasus *non-linear* memetakan data pelatihan menjadi berhingga pada dimensi ruang; dengan demikian, sangat cocok untuk menangani masalah hubungan *nonlinear*.

Oleh karena itu pada jurnal ini dilakukan penerapan kembali pada kernel radial basis untuk peramalan menggunakan SVR. Pada jurnal ini akan dijelaskan sebagai berikut. Dalam Bagian 2, meninjau secara singkat pada SVR serta kernel radial basis, fungsi kerugian (*loss function*), *Quadratic Programming Standard*, dan bagaimana mencari α . Dalam Bagian 3, kami menggunakan SVR-Radial Basis yang terintegrasi dan evaluasi kinerja hasil studi empiris. Bagian 4 merangkum dan menyimpulkan. Dalam tulisan ini kami menggunakan SVR dengan kernel radial basis untuk memperkirakan harga minyak sawit mentah.

2. *Support Vector Regression* (SVR)

SVM adalah sebuah metode yang digunakan untuk kasus klasifikasi, namun prinsip metode tersebut dapat dikembangkan ke dalam regresi dan metode peramalan (*Time series*). misalkan terdapat l data *training*, (x_i, y_i) , $i = 1, \dots, l$ dengan data input $x = \{x_1, \dots, x_l\} \subseteq \mathfrak{R}^N$ dan $y = \{y_1, \dots, y_l\} \subseteq \mathfrak{R}$ dan l adalah banyaknya data training. Dengan metode SVR didapat fungsi regresi sebagai berikut:

$$f(x) = w^T \varphi(x) + b \quad (1)$$

dengan:

\mathbf{w} = vektor pembobot

$\varphi(\mathbf{x})$ = fungsi yang memetakan \mathbf{x} dalam suatu dimensi

b = bias

Agar mendapatkan generalisasi yang baik untuk fungsi regresi $f(\mathbf{x})$, dapat dilakukan dengan cara meminimalkan norm dari \mathbf{w} . Oleh karena itu perlu adanya penyelesaian problem optimasi:

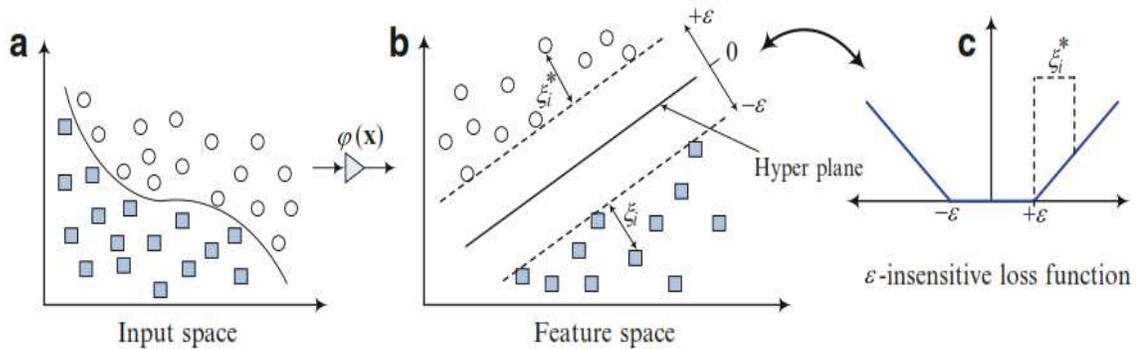
$$\min \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\} \quad (2)$$

dengan syarat:

$$y_i - \mathbf{w}^T \varphi(\mathbf{x}_i) - b \leq \varepsilon \quad (3)$$

$$\mathbf{w}^T \varphi(\mathbf{x}_i) - y_i + b \leq \varepsilon, i = 1, 2, \dots, l$$

Diasumsikan bahwa ada suatu fungsi $f(\mathbf{x})$ yang dapat mengaproksimasi semua titik (\mathbf{x}_i, y_i) dengan presisi ε . Dalam kasus ini diasumsikan bahwa semua titik ada dalam rentang $f(\mathbf{x}) \pm \varepsilon$ (*feasible*). Dalam hal ketidaklayakan (*infeasible*), dimana ada beberapa titik yang mungkin keluar dari rentang $f(\mathbf{x}) \pm \varepsilon$, sehingga dapat ditambahkan variabel *slack* t, t^* untuk mengatasi masalah pembatas yang tidak layak dalam problem optimasi.



Gambar 1. Ilustasi SVR (Hong, 2013)

Gambar 1 memperlihatkan situasi ini secara grafis, hanya titik-titik di luar area yang mempunyai kontribusi terhadap pinalti. Selanjutnya problem optimasi di atas dapat diformulasikan sebagai berikut:

$$\min \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (t_i + t_i^*) \right\} \quad (4)$$

dengan syarat :

$$y_i - \mathbf{w}^T \varphi(\mathbf{x}_i) - b - t_i \leq \varepsilon, i = 1, 2, \dots, l \quad (5)$$

$$\mathbf{w}^T \varphi(\mathbf{x}_i) - y_i + b - t_i^* \leq \varepsilon, i = 1, 2, \dots, l$$

$$t_i, t_i^* \geq 0$$

Menurut Scholkopf dan Smola [13] [14] solusi optimal untuk persamaan (4) dengan pembatas yang tertera pada persamaan (5) adalah dengan fungsi *lagrange* sebagai berikut:

$$L(w, b, t, t^*, \alpha, \alpha^*, \eta, \eta^*) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (t_i + t_i^*) - \sum_{i=1}^l \alpha_i (\varepsilon + t_i - y_i + w^T \varphi(x_i) + b) - \sum_{i=1}^l \alpha_i^* (\varepsilon + t_i^* + y_i - w^T \varphi(x_i) - b) - \sum_{i=1}^l (\eta_i t_i + \eta_i^* t_i^*) \quad (6)$$

Dimana C didefinisikan oleh peneliti dan $\alpha_i, \alpha_i^*, \eta_i, \eta_i^*$ adalah *lagrange multiplier*. Untuk mendapatkan solusi yang optimal, maka dilakukan turunan parsial dari L terhadap w, b, t , dan t^* .

$$\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0 \quad (7)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^l (\alpha_i^* - \alpha_i) = 0 \quad (8)$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^l (\alpha_i - \alpha_i^*) \varphi(x_i) = 0 \quad (9)$$

$$\frac{\partial L}{\partial t_i} = C - \alpha_i - \eta_i = 0 \quad (10)$$

$$\frac{\partial L}{\partial t_i^*} = C - \alpha_i^* - \eta_i^* = 0 \quad (11)$$

Dari turunan parsial tersebut dihasilkan $w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \varphi(x_i)$, sehingga fungsi regresi secara eksplisit dirumuskan sebagai berikut:

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \mathbf{K}(x_i, x) + b \quad (12)$$

Dimana $\mathbf{K}(x_i, x)$ merupakan fungsi kernel yang memiliki nilai inner product dari dua vektor x_i dan x .

Dimana selisih antara α_i dan α_i^* menghasilkan nilai beta dan b adalah bias. Menurut Scholkopf dan Smola [13] solusi optimal untuk bias (b) dapat dihitung menggunakan KKT (Karush-Kuhn-Tucker) sebagai berikut:

$$\alpha_i (\varepsilon + t_i - y_i + w^T \varphi(x_i) + b) = 0 \quad (13)$$

$$\alpha_i^* (\varepsilon + t_i^* + y_i - w^T \varphi(x_i) - b) = 0 \quad (14)$$

$$(C - \alpha_i) t_i = 0 \quad (15)$$

$$(C - \alpha_i^*) t_i^* = 0 \quad (16)$$

Sehingga didapat:

$$b = y_i - w^T \varphi(x_i) - \varepsilon \quad \text{untuk } 0 < \alpha_i < C \quad (17)$$

$$b = y_i - w^T \varphi(x_i) + \varepsilon \quad \text{untuk } 0 < \alpha_i^* < C \quad (18)$$

2.1 Kernel Radial Basis

Pendekatan metode kernel memiliki empat aspek utama: (i) Data yang masuk ke dalam *euclidean feature space*; (ii) hubungan linier yang dicari di *feature space*; (iii) Algoritma diimplementasikan sehingga hanya *inner* produk antara vektor dalam *feature space* yang diperlukan; (iv) Produk dapat langsung dihitung dari data asli oleh efisien "short-cut" atau yang dikenal sebagai fungsi kernel. Hal ini juga dikenal sebagai *kernel trick*. Ide untuk menggunakan fungsi kernel sebagai inner produk dalam *feature space* diperkenalkan pada teknik *machine learning* pada tahun 1964 dengan metode fungsi

potensial. Melalui teknik ini Schölkopf *et al* [13] [14] melakukan kombinasi dengan hyperplanes margin pada teknik SVM. Banyak teknik data mining yang dikembangkan dengan pendekatan linier, sehingga algoritma yang dihasilkan terbatas pada kasus linear. secara umum, kasus-kasus di dunia nyata adalah kasus yang tidak linier sehingga dapat mengatasi data yang tidak linier yang sering terjadi pada kasus nyata, dapat diterapkan untuk fungsi kernel. Dengan fungsi kernel dari data dalam ruang input dipetakan ke ruang fitur dengan dimensi yang lebih tinggi melalui ϕ dinyatakan sebagai:

$$\begin{aligned}
 \phi : \mathbf{x} \rightarrow \phi(\mathbf{x}) \quad K_{\text{RBF}}(\mathbf{x}, \mathbf{x}') &= \exp\left[-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right] & (19) \\
 &= \exp\left[-\frac{1}{2}\langle \mathbf{x} - \mathbf{x}', \mathbf{x} - \mathbf{x}' \rangle\right] \\
 &= \exp\left[-\frac{1}{2}\|\mathbf{x}\|^2 - \frac{1}{2}\|\mathbf{x}'\|^2\right] \exp\left[-\frac{1}{2}2\langle \mathbf{x}, \mathbf{x}' \rangle\right] \\
 C &:= \exp\left[-\frac{1}{2}\|\mathbf{x}\|^2 - \frac{1}{2}\|\mathbf{x}'\|^2\right] \\
 &= C \exp\langle \mathbf{x}, \mathbf{x}' \rangle \\
 K_c(\mathbf{x}, \mathbf{x}') &:= K_a(\mathbf{x}, \mathbf{x}') + K_b(\mathbf{x}, \mathbf{x}')
 \end{aligned}$$

Melakukan implikasi dengan ψ_c sehingga didapatkan

$$\begin{aligned}
 \psi_c(\mathbf{x}) &:= (\psi_a(\mathbf{x}), \psi_b(\mathbf{x})) \\
 \langle \psi_c(\mathbf{x}), \psi_c(\mathbf{x}') \rangle &:= \langle \psi_a(\mathbf{x}), \psi_a(\mathbf{x}') \rangle + \langle \psi_b(\mathbf{x}), \psi_b(\mathbf{x}') \rangle
 \end{aligned}$$

Penggunaan kernel yang berbeda terdapat kelebihan dan kekurangan. Radial basis kernel memiliki sejumlah property yang menarik yang mampu menghubungkan pendekatan non-linier dan melakukan interpolasi Vapnik [16]

2.2 Loss Function

Menurut Gunn [8] *loss function* adalah fungsi yang menunjukkan hubungan antara error dengan bagaimana error ini dikenai pinalti. Perbedaan *loss function* akan menghasilkan formulasi SVR yang berbeda. Ada 2 jenis *loss function* yang digunakan dalam penelitian ini, yaitu ε -insensitive dan *quadratic loss function*. Berikut ini adalah formulasi matematika untuk ε -insensitive *loss function*:

$$L(\mathbf{y}, f(\mathbf{x})) = \begin{cases} 0, & \text{untuk } |f(\mathbf{x}) - \mathbf{y}| < \varepsilon \\ |f(\mathbf{x}) - \mathbf{y}| - \varepsilon, & \text{untuk yang lain} \end{cases} \quad (20)$$

Solusi yang diberikan adalah:

$$\max \left\{ -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_{i=1}^l \alpha_i (y_i - \varepsilon) - \alpha_i^* (y_i + \varepsilon) \right\} \quad (21)$$

atau dapat disederhanakan menjadi:

$$\max \left\{ -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \beta_i \beta_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \sum_{i=1}^l \beta_i y_i \right\} \quad (22)$$

dengan syarat :

$$-C \leq \beta_i \leq C, i = 1, \dots, l \quad (23)$$

$$\sum_{i=1}^l \beta_i = 0$$

dimana $\beta_i = \alpha_i - \alpha_i^*$ dan $\beta_j = \alpha_j - \alpha_j^*$

Sedangkan *quadratic loss function*:

$$L(\mathbf{y}, f(\mathbf{x})) = (f(\mathbf{x}) - \mathbf{y})^2 \quad (24)$$

menghasilkan solusi :

$$\max \left\{ -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i - \frac{1}{2C} \sum_{i=1}^l (\alpha_i^2 + \alpha_i^{*2}) \right\} \quad (25)$$

atau dapat disederhanakan menjadi:

$$\max \left\{ -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \beta_i \beta_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_{i=1}^l \beta_i y_i - \frac{1}{2C} \sum_{i=1}^l \beta_i^2 \right\} \quad (26)$$

dengan syarat:

$$\sum_{i=1}^l \beta_i = 0 \quad (27)$$

dimana $\beta_i = \alpha_i - \alpha_i^*$ dan $\beta_j = \alpha_j - \alpha_j^*$

2.3 Formulation Support Vector Regression in Quadratic Programming Standard

Pada penerapan SVR untuk kasus regresi, dapat digunakan konsep *loss function* dengan pendekatan quadratic. Menurut Ancona [3] untuk memenuhi bentuk standar *quadratic programming* sehingga dapat diselesaikan dengan *solver quadratic programming* komersial, persamaan (20) perlu dikonversi menjadi:

$$\begin{aligned} & \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \mathbf{x}_i \cdot \mathbf{x}_j = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \left(\sum_{j=1}^l \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{j=1}^l \alpha_j^* \mathbf{x}_i \cdot \mathbf{x}_j \right) \\ & = \sum_{i=1}^l \alpha_i \sum_{j=1}^l \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i=1}^l \alpha_i \sum_{j=1}^l \alpha_j^* \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i=1}^l \alpha_i^* \sum_{j=1}^l \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^l \alpha_i^* \sum_{j=1}^l \alpha_j^* \mathbf{x}_i \cdot \mathbf{x}_j \\ & = \sum_{i=1}^l \alpha_i \left(\sum_{j=1}^l \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{j=1}^l \alpha_j^* (-\mathbf{x}_i \cdot \mathbf{x}_j) \right) \\ & \quad + \sum_{i=1}^l \alpha_i^* \left(\sum_{j=1}^l \alpha_j (-\mathbf{x}_i \cdot \mathbf{x}_j) + \sum_{j=1}^l \alpha_j^* \mathbf{x}_i \cdot \mathbf{x}_j \right) \end{aligned} \quad (28)$$

didefinisikan vektor baru \mathbf{a} dengan $2l$ komponen sehingga:

$$\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_l, \alpha_1^*, \alpha_2^*, \dots, \alpha_l^*)$$

dan \mathbf{D} adalah matriks $2l \times 2l$ yang didefinisikan sebagai :

$$D_{ij} = \begin{cases} x_i \cdot x_j & i = 1, 2, \dots, l \\ & j = 1, 2, \dots, l \\ -x_i \cdot x_{j-l} & i = 1, 2, \dots, l \\ & j = l + 1, l + 2, \dots, 2l \\ -x_{i-l} \cdot x_j & i = l + 1, l + 2, \dots, 2l \\ & j = 1, 2, \dots, l \\ x_{i-l} \cdot x_{j-l} & i = l + 1, l + 2, \dots, 2l \end{cases}$$

sehingga:

$$\begin{aligned} & \sum_{i=1}^l a_i (\sum_{j=1}^l a_j D_{ij} + \sum_{j=l+1}^{2l} a_j D_{ij}) + \sum_{i=l+1}^{2l} a_i (\sum_{j=1}^l a_j D_{ij} + \sum_{j=l+1}^{2l} a_j D_{ij}) \\ &= \sum_{i=1}^l a_i \sum_{j=1}^{2l} a_j D_{ij} + \sum_{i=l+1}^{2l} a_i \sum_{j=1}^{2l} a_j D_{ij} \\ &= \sum_{i=1}^{2l} a_i \sum_{j=1}^{2l} a_j D_{ij} = \mathbf{a} \cdot \mathbf{D} \mathbf{a} \end{aligned} \quad (29)$$

$$\begin{aligned} & \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) = \sum_{i=1}^l \alpha_i y_i + \sum_{i=1}^l \alpha_i^* (-y_i) + \sum_{i=1}^l \alpha_i (-\varepsilon) \\ & + \sum_{i=1}^l \alpha_i^* (-\varepsilon) = \sum_{i=1}^l \alpha_i (y_i - \varepsilon) + \sum_{i=1}^l \alpha_i^* (-y_i - \varepsilon) \end{aligned} \quad (30)$$

didefinisikan vektor lain \mathbf{c} dengan ukuran $2l$

$$\mathbf{c} = (y_1 - \varepsilon, y_2 - \varepsilon, \dots, y_l - \varepsilon, -y_1 - \varepsilon, -y_2 - \varepsilon, \dots, -y_l - \varepsilon)$$

sehingga:

$$\sum_{i=1}^l a_i c_i + \sum_{i=l+1}^{2l} a_i c_i = \sum_{i=1}^{2l} a_i c_i = \mathbf{a} \cdot \mathbf{c} \quad (31)$$

didefinisikan vektor \mathbf{b} dengan ukuran $2l$

$$\mathbf{b} = (1, 1, \dots, 1, -1, -1, \dots, -1)$$

sehingga pembatas liniernya menjadi :

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = \sum_{i=1}^{2l} a_i b_i = \mathbf{a} \cdot \mathbf{b} = 0 \quad (32)$$

dari variabel-variabel baru diatas, bentuk standar dari *Quadratic Programming* dapat ditulis menjadi :

$$\max \left\{ -\frac{1}{2} \mathbf{a} \mathbf{D} \mathbf{a} + \mathbf{a} \mathbf{c} \right\} \quad (33)$$

dengan syarat :

$$\mathbf{a} \mathbf{b} = 0$$

$$\mathbf{0} \leq \mathbf{a} \leq \mathbf{c}$$

Menurut Gunn (1998) cara mencari α adalah sebagai berikut:

$$\min \left\{ \frac{1}{2} \mathbf{a}^T \mathbf{H} \mathbf{a} + \mathbf{c}^T \mathbf{a} \right\} \quad (34)$$

dengan pembatas:

$$\mathbf{a} \cdot (1, \dots, 1, -1, \dots, -1) = 0 \quad (35)$$

$$a_i, a_i^* \geq 0, i = 1, \dots, l$$

dimana :

$$\mathbf{H} = \begin{bmatrix} \mathbf{xx}^T & -\mathbf{xx}^T \\ -\mathbf{xx}^T & \mathbf{xx}^T \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \boldsymbol{\varepsilon} + \mathbf{y} \\ \boldsymbol{\varepsilon} - \mathbf{y} \end{bmatrix} \text{ dan } \mathbf{a} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^* \end{bmatrix}$$

α dapat dihitung dengan cara:

$$\begin{bmatrix} -(H+I) & A^T \\ A & 1 \end{bmatrix} \begin{bmatrix} a \\ \lambda \end{bmatrix} = \begin{bmatrix} c \\ b \end{bmatrix} \quad (36)$$

dimana $A = [1, \dots, 1, -1, \dots, -1]$

Banyak cara untuk mengukur akurasi perkiraan, dan jawaban atas pertanyaan-pertanyaan ini tergantung pada apa yang sedang diteliti, apa ukuran akurasi yang digunakan, dan apa set data yang digunakan untuk menghitung ukuran akurasi. MAPE (*mean absolute percentage error*) adalah ukuran kesalahan relatif yang menggunakan nilai absolut. MAPE memiliki dua keuntungan. Pertama, nilai absolut menjaga kesalahan positif dan negatif dari perhitungan. Kedua, karena kesalahan relatif tidak tergantung pada skala variabel dependen, ukuran ini memungkinkan peneliti membandingkan akurasi perkiraan antara data *time series* berbeda skala Caraka [7]. Dimana \hat{y}_i adalah nilai perkiraan dan y_i adalah nilai sebenarnya. Sehingga, dapat ditulis:

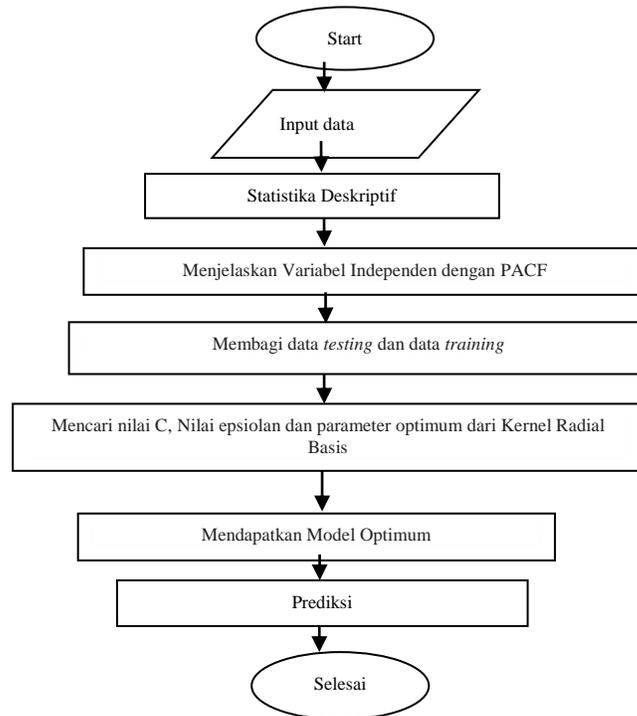
$$\text{MAPE} = \frac{(\sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i}) \times 100}{n} \quad (37)$$

Koefisien determinasi (R^2) dapat digunakan sebagai ukuran akurasi model yang menjelaskan seberapa besar variabel dependent Caraka [7]. Koefisien determinasi memiliki range dari 0 sampai 1 atau 0% sampai 100%.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \bar{y})^2}{\sum_{i=1}^n (y_i^2 - (\bar{y})^2)} \quad (38)$$

3. Analisis dan Pembahasan

Dalam jurnal ini, disajikan beberapa langkah untuk menggunakan metode SVR kernel radial basis. Menentukan variabel independen dengan menggunakan *Partial Autocorrelation Function* (PACF) dan membagi data training dan testing. Jika set pembelajaran memiliki lebih dari 1000 kasus data, menggunakan 5×2 cross-validasi untuk menghasilkan 10 pelatihan dan validasi set dengan stratifikasi, jika menggunakan 12-fold cross-validasi. Validasi set n digunakan untuk mengoptimalkan hiperparameter C (mencoba nilai 0,1, 1, 10, dan 100). Parameter hiper konfigurasi yang terbaik yang memiliki akurasi rata-rata tertinggi pada *fold* validasi) digunakan untuk melatih pada pelatihan *fold*. Untuk memecahkan masalah optimasi SVR atau masalah optimasi yang lebih kompleks diukur; pada jurnal ini digunakan mean data dan standar deviasi. Data yang digunakan dalam penelitian ini adalah data sekunder, yaitu data historis yang diambil melalui situs (<http://www.investing.com/commodities/crude-palm-oil-historical-data>) data historis harga minyak mentah kelapa sawit mulai 01 Januari 2015 hingga 5 September 2016. Untuk lebih jelasnya dapat dilihat pada Gambar.2



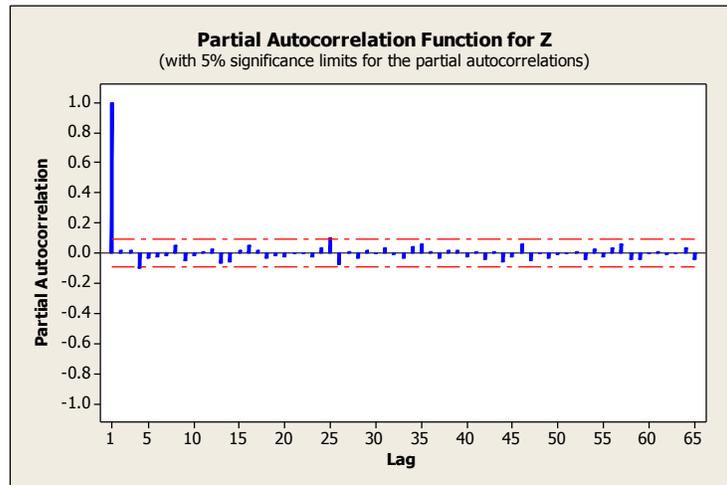
Gambar 2. Flowchart

Untuk melihat karakteristik data dapat melihat tabel statistika deskriptif yang disajikan pada Table 1.

Table 1. Statistika Deskriptif

| | Mean | StdDev | Minimum | Maximum |
|-----------------------|---------|--------|---------|---------|
| Crude Palm Oil Prices | 460.944 | 54.31 | 352.2 | 573.6 |

Berdasarkan Table 1. Dapat dilihat bahwa harga minimum kelapa sawit adalah \$352.2 setelah melakukan pengecekan pada dataset diketahui bahwa nilai minimum tersebut terjadi pada 26 August, 2015 dan nilai maksimum sebesar \$573.6 terjadi pada 22 April 2016. Harga minyak kelapa sawit juga memiliki keragaman yang rendah dengan nilai varian σ (stddev) \$54.31. Identifikasi lag sebagai komponen input didasarkan pada lag-lag yang memiliki nilai PACF terbesar. Hal ini dikarenakan karakteristik persamaan model tersebut memiliki kesamaan dengan model AR. Yasin *et al* [18]. Berdasarkan gambar 3. Dapat dilihat bahwa lag keluar dari garis stasioner pada lag ke 1 oleh karena itu dalam jurnal ini kami menggunakan lag 1 atau X_{t-1} .



Gambar 3. PACF

Analisis selanjutnya adalah melakukan estimasi parameter dan mendapatkan parameter terbaik dengan memiliki nilai epsilon dan error terkecil Yasin, *et al* [18]. Pada jurnal ini digunakan software  dengan *package* e1071

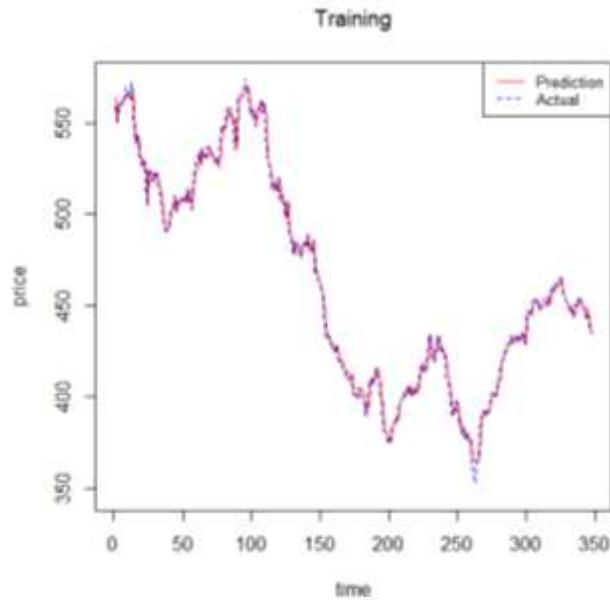
Table 2. Parameter Estimation

| Cost (C) | Epsilon (ϵ) | Error | Dispersion |
|------------|------------------------|------------------|------------------|
| 0.1 | 0.1000 | 68.17083 | 211.8879 |
| 1 | 0.1000 | 29.97035 | 62.0149 |
| 10 | 0.1000 | 28.9281 | 62.33006 |
| 100 | 0.1000 | 28.70373 | 60.60751 |
| 0.1 | 0.0100 | 63.07452 | 192.088 |
| 1 | 0.0100 | 29.51551 | 61.01827 |
| 10 | 0.0100 | 28.04148 | 58.58143 |
| 100 | 0.0100 | 27.83702 | 58.20656 |
| 0.1 | 0.0010 | 63.98349 | 196.3066 |
| 1 | 0.0010 | 29.39629 | 61.71205 |
| 10 | 0.0010 | 27.54666 | 58.0491 |
| 100 | 0.0010 | 28.06313 | 57.91202 |
| 0.1 | 0.0001 | 64.10834 | 196.5439 |
| 1 | 0.0001 | 29.3608 | 61.77654 |
| 10* | 0.0001* | 27.40893* | 58.09817* |
| 100 | 0.0001 | 28.0534 | 58.02616 |

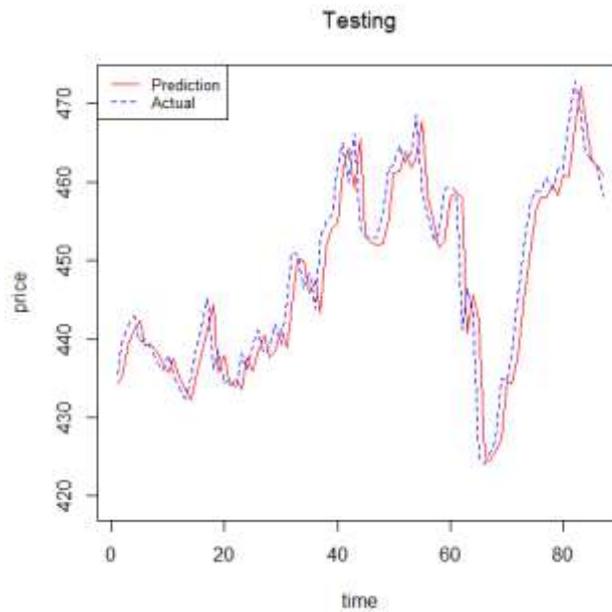
*Best Parameter

Berdasarkan simulasi didapat nilai parameter terbaik dengan nilai $C = 10$ dan epsilon 0.0001 dengan nilai error terkecil 27.40893. Untuk melihat perbandingan nilai target (*prediction*) dan *actual* dapat dilihat pada Gambar.4 and Gambar.5 SVR Kernel radial

basis terindikasi memiliki performa yang sangat baik dalam melakukan pengujian dan Pelatihan model (indikasi dari garis biru dan merah)



Gambar 4. Training SVR Radial Basis



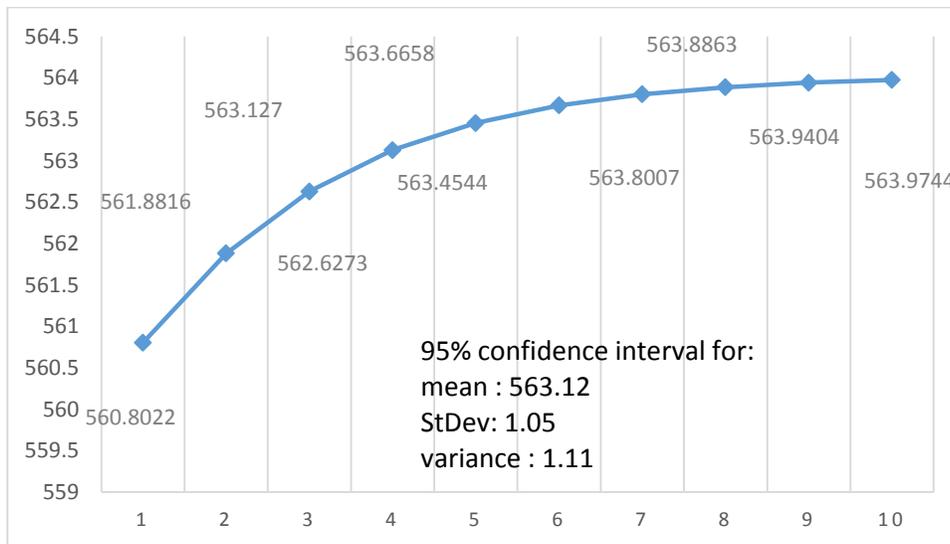
Gambar 5. Testing SVR Radial Basis

Pada jurnal ini dipilih sebesar 80% dari data untuk dilakukan pelatihan (*training*) dan pengujian (*testing*) 20% untuk evaluasi model. SVR kernel tersebut memiliki akurasi yang sangat baik dengan nilai R^2 yang sangat baik dan nilai error (MAPE) dengan sangat kecil. Untuk lebih lengkap nya dapat dilihat pada Table 3.

Tabel 3. Accuracy

| Data | Accuracy | |
|----------------|-----------|-----------|
| | R^2 | MAPE (%) |
| Training (80%) | 0.9871738 | 0.9191906 |
| Testing (20%) | 0.8345659 | 0.8761705 |

Langkah selanjutnya adalah dilakukan peramalan dengan nilai parameter terbaik dan kernel radial basis untuk enam langkah kedepan dapat dilihat pada Gambar.6 bahwa harga minyak mentah akan selalu meningkat dengan nilai mean \$ 563.12 dan stddev (σ) 1.05 dengan 95% confidence interval



Gambar 6. Forecasting

4. Kesimpulan

Artikel ini menyajikan penerapan SVR dengan kernel radial basis untuk peramalan harga minyak mentah kelapa sawit (*crude palm oil prices*) pada ukuran akurasi digunakan MAPE dan R^2 dapat disimpulkan bahwa evaluasi model tersebut memberikan nilai yang baik dan menunjukkan validitas, akurasi yang baik dari model yang telah direalisasikan dengan nilai R^2 training sebesar 98,71738% dn R^2 testing sebesar 83,45659%. Untuk penelitian selanjutnya dimungkinkan untuk melakukan optimasi yang berbeda dalam pemilihan parameter yang optimal pada SVR atau dapat menggunakan kernel yang berbeda atau melakukan dua atau tiga kombinasi kernel yang berbeda (*multiple kernel*). Penggunaan kernel yang berbeda memungkinkan untuk

menggunakan input yang berasal dari representasi yang berbeda dan modilitas yang berbeda.

Ucapan Terima Kasih

Peneliti mengucapkan terimakasih kepada *reviewer* yang telah memberikan masukan terhadap penulisan. Penelitian ini merupakan signifikan diskusi dengan Bapak Hasbi Yasin sebagai *co-author* dan memberikan kajian terhadap teori dan teknis analisis. Peneliti mengucapkan terimakasih kepada Kadi Mey Ismail karena telah menjadi sahabat yang baik dengan memberikan dukungan moril kepada peneliti pertama dan Bapak Wawan Sugiyarto *Directorate of Islamic Financing Directorate General of Financing and Risk Management The Ministry of Finance of the Republic of Indonesia*. Semoga semesta ramah dengan semua impian dapat menjadi peneliti yang bermanfaat untuk Indonesia

Daftar Pustaka

- [1] be, S. Support Vector Machine for Pattern Classification. Springer-Verlag. London.2005
- [2] Arshad,F.M., Ghaffar,R.A. Crude Palm Oil Price Forecasting : Box-Jenkins Approach.PERTANIKA Vol.9.No.3,1986.
- [3] Ancona,V.A., Vargaz,J.D. Quadratic Function Fields with Exponent Two Ideal Class Group. Cornell University Library. Number Theory(math.NT). [arXiv:math/0504581.2005](https://arxiv.org/abs/math/0504581)
- [4] Blum,A. [A Discriminative Model for Semi-Supervised Learning](#). With Nina Balcan. *JACM* Vol 57, Issue 3.2010
- [5] C Cortes, V Vapnik. Support-vector networks. *Machine learning*. 20: 273-297.1995
- [6] Caraka, R.E., Yasin, H., Sugiyarto, W., Sugiaro and Ismail, K.E. Time Series Analysis Using Copula Gauss and AR(1)-N.GARCH(1,1). Vol.9 No.1; Jornal of Media Statistika, Universitas Diponegoro. DOI:10.14710/medstat.9.1.1-13. pp 01-13. 2016
- [7] Caraka,R.E., Devi,A.R. 2016. Application Of Non Parametric Basis Spline (Bspline) In Temperature Forecasting. *Statistika*, Vol. 4, No. 2. Pp.68-73
- [8] Gunn, S. Support Vector Machines for Classification and Regression. Technical Report, ISIS. 1998
- [9] Hamel, L. Knowledge Discovery with Support Vector Machine. United State Of America :A John Wiley &Sons, INC., Publication. 2009

- [10] Hong, W.C. Intelligent Energy Demand Forecasting, Lecture Notes in Energy 10, DOI 10.1007/978-1-4471-4968-2_2.2013
- [11] Mahmudah, E. Prediksi Harga Minyak Kelapa Sawit (CPO) Menggunakan Kombinasi Metode Fuzzy-Mamdani dan Evolution Strategies. ITTELKOM Repository. 2010
- [12] Santosa, B. Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis. Yogyakarta: Graha Ilmu. 2007
- [13] Schölkopf B., and Smola A. A Tutorial on Support Vector Regression. 2003
- [14] Schölkopf, B. and Smola, A.J.. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press. 2002
- [15] Schölkopf, *et al.* Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers A.I. Memo No. 1599 C.B.C.L. Paper No. 142. 1996
- [16] VN Vapnik. *Statistical learning theory*. Wiley, New York, 1998.
- [17] Yao, Y, *et al.* An Improved Grid Search Algorithm and Its Application in PCA and SVM Based Face Recognition, Journal of Computational Information Systems. Vol. 10, No. 3:1219–1229. 2014
- [18] Yasin, H., Caraka, R.E., Tarno, and Hoyyi, A. Prediction Of Crude Oil Prices Using Support Vector Regression (SVR) with grid search–cross validation algorithm. Vol.12 No.4; August. Global Journal of Pure and Applied Mathematics. Print ISSN : 0973-1768 Online ISSN: 0973-9750. pp. 3009–3020. 2016