

Peramalan Harga Saham Menggunakan Metode *Support Vector Regression* (SVR) Dengan *Particle Swarm Optimization* (PSO)

Vera Rusmalawati¹, M. Tanzil Furqon², Indriati³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹verarusmalawati@gmail.com, ²m.tanzil.furqon@gmail.com, ³indriati.tif@ub.ac.id

Abstrak

Salah satu keuntungan dari investasi pada saham adalah *capital gain*, yaitu keuntungan dari perdagangan saham. Dengan adanya peramalan harga saham akan meningkatkan keuntungan dari transaksi jual beli saham karena pemilik saham dapat mengetahui kapan waktu yang tepat untuk menjual atau membeli saham tertentu. *Support Vector Regression* (SVR) merupakan salah satu metode yang digunakan untuk peramalan karena dapat mengenali pola dari data *time series* dan dapat memberikan hasil peramalan yang baik bila parameter pentingnya dapat ditentukan secara baik pula. Sehingga diperlukan metode optimasi untuk menentukan parameter SVR sehingga SVR dapat secara optimal diterapkan dalam peramalan harga saham. Salah satu algoritma optimasi yang dapat digunakan adalah *Particle Swarm Optimization* (PSO). Peramalan harga saham menggunakan SVR dengan optimasi PSO ini menggunakan metode MAPE untuk mengevaluasi hasil peramalan. Berdasarkan pengujian yang telah dilakukan, nilai MAPE yang didapatkan adalah 0,8195% dengan *fitness* sebesar 0,5496 dengan parameter optimal yang didapatkan adalah jumlah partikel 40, iterasi PSO 40, iterasi SVR 1000, rentang parameter C 100 – 500, rentang parameter ϵ 0,0001 – 0,001, rentang parameter σ 0,001 – 2, rentang parameter γ 0,00001 – 0,001, rentang parameter λ 0,001 – 0,1, dan perbandingan data latih dan data uji dari data saham Bank BCA tahun 2016 yaitu 90%:10%.

Kata kunci: Peramalan, Saham, SVR, PSO, MAPE

Abstract

One of the advantages of investing in stocks is capital gains, which is the benefits from stock trading. The use of stock price forecasting will increase profits from stock sale and purchase transactions because shareholders can know when the right time to sell or buy a particular stock. Support Vector Regression (SVR) is one of the methods used for forecasting, it can recognize patterns of time series data and can provide good forecasting results when the parameters of importance can be determined well as well. So we need an optimization method to determine SVR parameters so that SVR can be optimally applied in stock price forecasting. One of the optimization algorithms that can be used is Particle Swarm Optimization (PSO). Stock price forecasting using SVR with PSO optimization uses MAPE method to evaluate forecasting results. Based on the test that has been done, the value of MAPE obtained is 0.8195% with fitness of 0.5496 with the optimal parameters obtained is the number of particles 40, iteration PSO 40, iteration SVR 1000, parameter range of C 100 - 500, the parameter range of ϵ 0, 0001 - 0,001, parameter range of σ 0,001 - 2, parameter range of γ 0,00001 - 0,001, parameter range of λ 0,001 - 0,1, and comparison of training data and test data from 2016 BCA Bank share data is 90%: 10%.

Keywords: Forecasting, Stocks, SVR, PSO, MAPE

1. PENDAHULUAN

Investasi dalam bentuk saham pada pasar modal di Indonesia mengalami peningkatan ditandai dengan pencapaian jumlah tertinggi perdagangan saham pada Februari 2017 semenjak pasar saham Indonesia didirikan

menurut catatan Bursa Efek Indonesia (BEI) serta kenaikan Indeks Harga Saham Indonesia (IHSG) sebesar 15% (Pasopati, 2017). Hal ini membuat saham menjadi salah satu objek investasi yang menjanjikan. Sedangkan berdasar data operasional BEI dan data Kustodian Sentra Efek Indonesia (KSEI)

jumlah investor baru pasar modal Indonesia meningkat 23,47 % sepanjang tahun 2016 (Ayu, 2017).

Tujuan utama dari investasi dalam bentuk saham adalah untuk memaksimalkan *return* dan meminimalkan resiko yang didapatkan dari *return* (Peng, 2001). *Return* saham dapat berupa laba perusahaan yang dibagikan (dividen) atau hasil jual beli saham, namun harga jual dapat berbeda dari harga belinya sehingga ada potensi keuntungan dan kerugian dari transaksi jual-beli saham (Zubir, 2013). Sehingga investasi pada saham memerlukan analisis yang mendalam agar investor mengetahui kapan harus menjual atau membeli saham tertentu sehingga dapat meningkatkan keuntungan, salah satu caranya yaitu dengan meramalkan harga saham yang akan datang. Ketersediaan data harga saham dari waktu ke waktu pada Bursa Efek dapat dimanfaatkan untuk melakukan analisis teknis sehingga harga saham yang akan datang bisa diramalkan untuk meningkatkan keuntungan investor.

Penelitian yang menggunakan data historis saham telah banyak dilakukan seperti penelitian pembuatan model regresi prediksi harga saham menggunakan Algoritma Genetika. Pada penelitian yang dilakukan oleh Rahmi (Rahmi, 2015) berhasil membuat model regresi untuk prediksi harga saham dan hasil prediksi lebih baik dibandingkan aplikasi miniTab namun nilai MSE dari algoritma ini masih cukup tinggi yaitu 54,5023. Metode lain yang dapat diterapkan dalam peramalan data *time series* yaitu *Support Vector Regression* (SVR) yang merupakan model regresi dari algoritma *Support Vector Machine* (SVM). Seperti penelitian yang dilakukan oleh Meesad & Rasel (2013) untuk memprediksi harga saham, SVR merupakan algoritma *machine learning* yang memiliki performa yang baik untuk mengenali pola dari data *time series* serta dapat memberikan hasil prediksi yang baik jika nilai dari parameter pentingnya dapat ditentukan secara baik pula (Meesad & Rasel, 2013).

Parameter SVR biasanya dipilih berdasarkan pengalaman dan *grid search* (GS), pengalaman membutuhkan pemahaman yang mendalam mengenai teori SVR sedangkan komputasi pada metode GS besar sehingga kedua metode tersebut tidak dapat menjamin tercapainya solusi optimal (Pei-you & Lu, 2013). Maka dari itu diperlukan metode

optimasi untuk menentukan parameter SVR sehingga SVR dapat secara optimal diterapkan dalam peramalan harga saham. Optimasi parameter SVR telah banyak dilakukan seperti penelitian mengenai prediksi volume penjualan mobil dengan SVR-PSO (Lu & Geng, 2011) yang menyatakan bahwa PSO-SVR lebih baik daripada GA-SVR dalam tingkat efisiensi dan akurasi prediksi.

PSO merupakan metode dengan pendekatan stokastik berbasis populasi untuk memecahkan masalah optimasi diskrit dan kontinyu, yang bila dibandingkan dengan Algoritma Genetika (GA) dan algoritma heuristik lainnya, lebih mudah untuk diterapkan dan tidak banyak parameter yang perlu ditentukan (Hu, et al., 2015). Penelitian lain yaitu mengenai peramalan *short term solar radiation* menggunakan SVR dan memberikan hasil bahwa SVR yang dioptimasi parameternya menggunakan PSO memberikan hasil ramalan yang lebih baik dibandingkan dengan SVR biasa dan SVR-GA (Sreekumar, et al., 2016). Dari penelitian-penelitian tersebut metode SVR dipilih untuk meramalkan harga saham dengan optimasi PSO pada parameternya. Dengan tujuan PSO-SVR dapat menghasilkan kombinasi parameter SVR yang optimal sehingga dapat meramalkan harga saham dengan tingkat akurasi yang baik pula.

2. TINJAUAN PUSTAKA

2.1. Saham

Saham merupakan surat bukti kepemilikan bagian modal perseroan terbatas yang memberi hak atas dividen dan lain-lain menurut besar kecilnya modal yang disetor. Ada dua macam keuntungan yang dapat kita peroleh dari investasi saham yaitu *dividen* dan *capital gain*. *Dividen* merupakan laba yang akan dibagikan kepada pemegang saham sesuai porsi kepemilikannya (Zubir, 2013).

Selain *dividen*, keuntungan lain yang bisa didapatkan oleh pemilik saham yaitu *capital gain*. *Capital gain* adalah keuntungan yang kita dapatkan dengan memperdagangkan saham. Berbeda dengan *dividen* yang dipengaruhi oleh kinerja perusahaan, keuntungan yang didapat dari *capital gain* sangat bergantung pada harga saham dipasar modal sendiri. Data harga saham ini terekam berdasarkan kurun waktu tertentu dan dapat diakses oleh umum melalui media surat kabar maupun internet seperti *yahoo*

finance atau google finance. Dalam penelitian ini data yang digunakan untuk meramalkan harga saham adalah *close price* karena *close price* menggambarkan semua aktifitas dari indeks saham pada perdagangan hari itu (Adebiyi, et al., 2014).

2.2. Support Vector Regression (SVR)

Support Vector Regression (SVR) merupakan versi regresi dari *Support Vector Machine (SVM)* yang diperkenalkan oleh Vapnik, Steven Golowich dan Alex Smola pada tahun 1997 (Vapnik, et al., 1997). Dasar pemikiran dari SVR adalah untuk memetakan set data ke ruang fitur dimensi tinggi non-linear dan menyelesaikan permasalahan regresi dalam ruang fitur dimensi ini (Lu & Geng, 2011).

Jika terdapat data set $\{(x_1, y_1), \dots, (x_i, y_i)\} x, y \in R$, dimana x menyatakan variabel input dan y merupakan nilai aktual (target). Tujuan dari SVR adalah mencari fungsi $f(x)$ yang dapat meramalkan nilai target dengan nilai deviasi ϵ (*epsilon*) untuk setiap nilai masukan, dan sebisa mungkin fungsi $f(x)$ mengikuti pola naik turunnya data (Bi, et al., 2011). Sedangkan nilai ϵ dan variabel *slack* akan mempengaruhi akurasi, semakin kecil nilai ϵ dan semakin tinggi variabel *slack* maka akurasi akan semakin tinggi dan sebaliknya.

Penyelesaian masalah non-linear dapat diselesaikan menggunakan algoritma *Sequential Learning SVR*. Sebelum masuk ke dalam algoritma tersebut, dilakukan proses normalisasi data terlebih dahulu dengan dengan tujuan meningkatkan akurasi regresi dengan mengurangi tingkat *error* dalam komputasi. Proses normalisasi dilakukan sebelum melakukan pelatihan menggunakan SVR dengan menggunakan rumus:

$$x'_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{1}$$

Keterangan:

- x'_i = hasil normalisasi data ke-i
- x_i = nilai data ke-i
- x_{min} = nilai data minimum dari keseluruhan
- x_{max} = nilai data maksimum dari keseluruhan

Langkah dari algoritma *Sequential Learning SVR* adalah sebagai berikut (Vijayakumar, 1999):

1. Inisialisasi parameter SVR yaitu λ (variabel skalar), γ (*learning rate*), C (kompleksitas), ϵ (nilai kerugian) dan iterasi maksimum
2. Inisialisasi awal α_i^* dan α_i masing-masing bernilai 0 tiap data ke- i dan menghitung matriks *Hessian* dengan rumus sebagai berikut.

$$R_{ij} = K(x_i, x_j) + \lambda^2 \text{ untuk indeks } i, j = 1, 2, \dots, l \tag{2}$$

Keterangan:

- R_{ij} = Matriks *Hessian*
- K = *Kernel* yang digunakan
- x_i = data ke- i
- x_j = data ke- j
- λ^2 = variabel skalar
- l = banyaknya data

3. Untuk tiap data latih indeks 1 hingga l , lakukan proses perhitungan dengan rumus berikut secara berurutan.

- a. Hitung error tiap data latih $E_i = y_i - \sum_{j=1}^l (\alpha_j^* - \alpha_j) R_{ij}$ (3)

Keterangan:

- y_i = nilai target data latih ke- i
- E_i = nilai error data ke- i
- α_j^*, α_j = *Lagrange Multipliers*

- b. Menghitung variabel tunggal $\delta\alpha^*$ dan $\delta\alpha$ (4)
 $\delta\alpha_i^* = \min\{\max[\gamma(E_i - \epsilon), -\alpha_i^*], C - \alpha_i^*\}$ (5)
 $\delta\alpha_i = \min\{\max[\gamma(-E_i - \epsilon), -\alpha_i], C - \alpha_i\}$

Keterangan:

$\delta\alpha_i^*, \delta\alpha_i$ = bukan perkalian antara δ dan α_i atau δ dan α_i^* melainkan kesatuan variabel

ϵ = nilai kerugian

C = kompleksitas

γ = Learning rate (LR)

- c. Memperbarui nilai α_i dan α_i^* (5)
 $\alpha_i^* = \alpha_i^* + \delta\alpha_i^*$ (6)
 $\alpha_i = \alpha_i + \delta\alpha_i$

4. Kembali ke langkah ketiga hingga iterasi maksimum tercapai atau $\max(|\delta\alpha_i|) < \epsilon$ dan $\max(|\delta\alpha_i^*|) < \epsilon$

5. Data disebut sebagai *support vector* jika memenuhi kondisi $(\alpha_i^* - \alpha_i) \neq 0$, berarti kedua variabel memiliki nilai berbeda

6. Fungsi peramalan digunakan untuk meramalkan nilai target pada data uji

$$f(x) = \sum_{i=1}^l (\alpha_i^* - \alpha_i) (K(x_i, x) + \lambda^2) \quad (7)$$

α_i^*, α_j = Lagrange Multiplier

$K(x_i, x)$ = fungsi kernel yang digunakan

λ = variabel skalar

l = banyaknya data uji

7. Selesai

Pada penelitian ini kernel yang digunakan adalah Kernel Gaussian atau RBF (*Radial Basis Function*) karena memberikan performa yang terbaik untuk meramalkan muatan dibandingkan kernel yang lain (Hong, 2009). Berikut adalah persamaan dari kernel Gaussian atau RBF:

$$K(x, x_i) = \exp\left(-\frac{\|x-x_i\|^2}{2\sigma^2}\right) \quad (8)$$

Keterangan:

x = nilai dari fitur data yang digunakan untuk peramalan

x_i = nilai dari fitur data latih ke- i

σ = nilai dari radial basis

2.3. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) merupakan algoritma evolusioner dengan optimasi global yang pertama kali diperkenalkan oleh Eberhart dan Kennedy pada tahun 1995, yang terinspirasi dari perilaku kawanan burung dan ikan. Setiap burung digambarkan sebagai partikel yang merepresentasikan solusi permasalahan dan yang memiliki posisi (x) dan kecepatan (v). PSO memiliki dua fungsi utama yaitu *update* kecepatan dan *update* posisi partikel yang digunakan untuk memperbarui kecepatan dan posisi partikel sehingga partikel akan terus dipercepat mendekati posisi partikel terbaik sebelumnya dan terbaik global sampai kondisi minimum yang error tercapai (Pei-you & Lu, 2013). Persamaan dari *update* kecepatan dan posisi partikel adalah sebagai berikut (Pei-you & Lu, 2013).

$$v'_{id} = \omega * v_{id} + c_1 * r_1 * (p_{id} - x_{id}) + c_2 * r_2 * (p_{gd} - x_{id}) \quad (9)$$

$$x'_{id} = x_{id} + v'_{id} \quad (10)$$

dengan $i = 1, 2, \dots, n, d = 1, 2, \dots, D$

Dimana c_1 dan c_2 merupakan koefisien akselerasi, r_1 dan r_2 adalah angka acak yang dipilih dari rentang [0-1] dan ω merupakan bobot inersia. Sedangkan $x_{id}, p_{id}, v_{id}, p_{gd}$ secara berurutan merupakan posisi, posisi terbaik sebelumnya, kecepatan sebelumnya dari tiap partikel serta posisi partikel yang paling

baik diantara seluruh partikel pada dimensi D dari ruang pencarian .

Untuk mengatasi masalah pergerakan partikel yang hanya terjadi pada area solusi lokal dan global, perlu adanya pengurangan secara linear nilai ω dari nilai maksimum ke nilai minimum setiap iterasi bertambah (Lu & Geng, 2011).

$$\omega = \omega_{max} - iter * \frac{\omega_{max} - \omega_{min}}{itermax} \quad (11)$$

Iter merupakan iterasi sekarang sedangkan *itermax* menunjukkan maksimum iterasi, ω_{max} dan ω_{min} biasanya diatur pada nilai 0,9 dan 0,1 (Lu & Geng, 2011). Selain pengurangan secara linear pada nilai ω dilakukan pula perubahan pada nilai c_1 dan c_2 berdasarkan waktu atau disebut *Time-Varying Acceleration Coefficients (TVAC)*. Tujuannya agar dengan perubahan koefisien akan mempengaruhi komponen kognitif dan sosial sehingga pada awal PSO partikel akan bergerak lebih leluasa dalam ruang pencarian dibandingkan bergerak hanya menuju populasi terbaik, dan semakin menuju akhir optimasi dengan perubahan koefisien akan memungkinkan partikel akan berkumpul dalam global optimal (Ratnaweera, et al., 2004). Berikut merupakan persamaan perubahan nilai c_1 dan c_2 :

$$c_1 = (c_{1f} - c_{1i}) * \frac{iter}{itermax} + c_{1i} \quad (12)$$

$$c_2 = (c_{2f} - c_{2i}) * \frac{iter}{itermax} + c_{2i} \quad (13)$$

Dimana nilai c_{1i}, c_{1f}, c_{2i} , dan c_{2f} adalah konstan dengan *iter* merupakan nilai iterasi sekarang dan *itermax* adalah maksimum iterasi. Nilai c_{1i}, c_{1f}, c_{2i} , dan c_{2f} ditetapkan pada 2,5, 0,5, 0,5 dan 2,5 (Ratnaweera, et al., 2004).

Langkah dari algoritma PSO untuk pemilihan parameter SVR adalah (Cao & Ahmed, 2010):

1. Inisialisasi awal posisi dan kecepatan tiap partikel
2. Menghitung nilai fitness tiap partikel
3. Memperbarui kecepatan dan posisi tiap partikel
4. Proses (1-3) diulangi hingga maksimal iterasi tercapai

Berdasarkan pengkodean pada partikelnya PSO terdiri atas beberapa macam diantaranya *Binary Code PSO, Real Code PSO*, dan *Hybrid Discrete PSO*. Penentuan pengkodean PSO disesuaikan dengan permasalahan yang akan dihadapi. Untuk pemilihan parameter SVR,

pengkodean *real code* dipilih karena memuat banyak variabel yang harus dioptimasi dan variabel tersebut adalah bilangan desimal (*real*).

Pada *update* kecepatan untuk *real code PSO* digunakan teknik pembatasan atau disebut *velocity clamping* yang ditentukan berdasarkan nilai minimum dan maksimum setiap dimensi dari representasi solusi partikel ($x_{ij}=[x_{min_j} ; x_{max_j}]$) seperti pada persamaan dibawah ini.

$$v_{max_j} = k \frac{(x_{max_j} - x_{min_j})}{2} \quad k \in (0,1] \quad (\text{Marini \& Walczak, 2015}) \quad (14)$$

Batasan kecepatan atau *threshold* yang digunakan adalah seperti persamaan dibawah ini (Marini & Walczak, 2015).

Jika $v_{ij}^{t+1} > v_{max_j}$ maka $v_{ij}^{t+1} = v_{max_j}$ (15)
 Jika $v_{ij}^{t+1} < -v_{max_j}$ maka $v_{ij}^{t+1} = -v_{max_j}$

Setiap partikel memiliki nilai *fitness* dimana semakin besar nilai *fitness* maka semakin baik partikel tersebut untuk dijadikan solusi. Nilai *fitness* untuk optimasi permasalahan SVR didapatkan dari perhitungan evaluasi model SVR. Nilai *fitness* dapat diperoleh menggunakan tingkat error dari proses evaluasi yang dilakukan seperti pada persamaan berikut (Rodriguez, et al., 2009).

$$fitness = \frac{1}{1+MAPE} \quad (17)$$

Keterangan:

MAPE = *Mean Absolute Percentage Error* (tingkat nilai *error rate*)

Metode yang digunakan untuk melakukan penghitungan nilai *error rate* adalah *Mean Absolute Percentage Error* (MAPE), dengan rumus seperti pada persamaan 2.13 (Lu & Geng, 2011).

$$f = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (18)$$

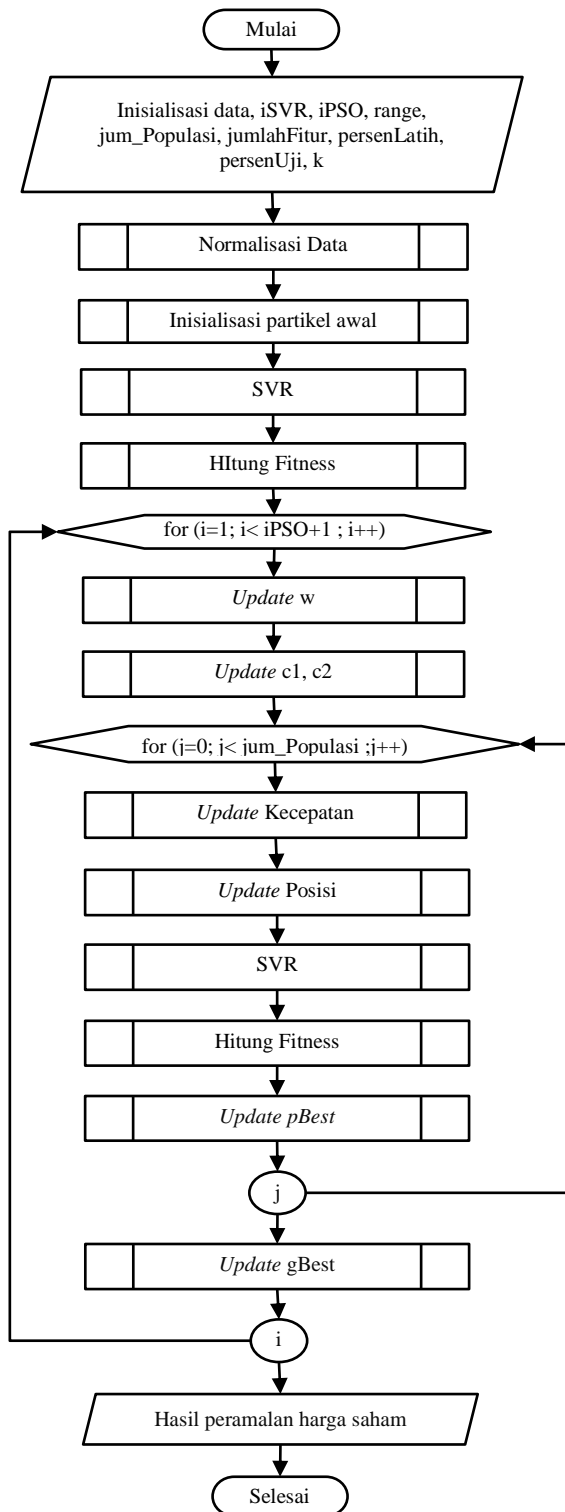
Dimana y_i dan \hat{y}_i merupakan nilai aktual dan nilai target, sedangkan n adalah jumlah data yang diramalkan.

3. SVR DENGAN PSO UNTUK PERAMALAN SAHAM

Optimasi SVR dengan menggunakan PSO bertujuan untuk menghasilkan tingkat error paling rendah untuk peramalan harga saham. Diagram alir optimasi SVR menggunakan PSO seperti pada Gambar 1. Penjelasan setiap langkah SVRPSO adalah sebagai berikut.

Normalisasi Data: normalisasi data dilakukan diawal dengan harapan akan mengurangi waktu komputasi program. Sehingga data dinormalisasi terlebih dahulu sebelum masuk dalam optimasi peramalan.

Inisialisasi Partikel Awal: partikel untuk permasalahan optimasi SVR untuk peramalan harga saham berbentuk *real code* dengan panjang dimensi adalah 5, masing dimensi menampung nilai parameter yang akan dioptimasi.



Gambar 1 Diagram Alir SVRPSO

SVR: setiap partikel yang diinisialisasi akan digunakan untuk membentuk model peramalan melalui proses *sequential learning* SVR yang selanjutnya dievaluasi menggunakan MAPE.

Hitung fitness: setelah nilai MAPE didapat kemudian dihitung *fitness*, ini digunakan untuk menginisialisasi Pbest dan Gbest awal.

Update w: dilakukan untuk memperbarui nilai w sehingga partikel tidak terjebak dalam local optimum.

Update c1 dan c2: dilakukan untuk memperbarui nilai koefisien akselerasi.

Update Kecepatan: memperbarui kecepatan tiap partikel sehingga partikel mampu bergerak mendekati solusi optimum, dengan tambahan penggunaan *velocity clamping* sehingga kecepatan partikel tidak terlalu besar dan kecil diluar jangkauan.

Update Posisi: memperbarui posisi dari partikel menggunakan kecepatan yang telah diperbarui kemudian hasil posisi baru harus diperiksa kembali agar tidak keluar dari rentang awal yang ditentukan.

Setelah *update* posisi dilakukan kemudian dihitung kembali nilai *fitness* tiap partikel yang baru sehingga dapat dilakukan pemilihan partikel lokal terbaik.

Update Pbest: untuk memperbarui partikel lokal setelah posisi diperbarui menggunakan *fitness* yang didapatkan dari SVR.

Update Gbest: untuk memperbarui partikel global sehingga didapatkan partikel terbaik dari partikel terbaik lokal (Pbest)

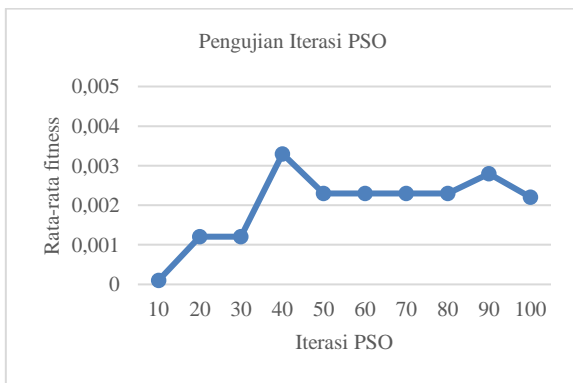
4. PENGUJIAN DAN ANALISIS

Terdapat 10 pengujian yang dilakukan dalam penelitian ini. Pengujian yang dilakukan diantaranya pengujian jumlah partikel, iterasi PSO, rentang parameter yang dioptimasi, iterasi SVR, jumlah fitur, dan perbandingan data latih dan data uji yang digunakan.

Untuk setiap pengujian diulang sebanyak 10 kali untuk mendapatkan *fitness* rata-rata. Dalam pengujian digunakan parameter awal sebagai berikut:

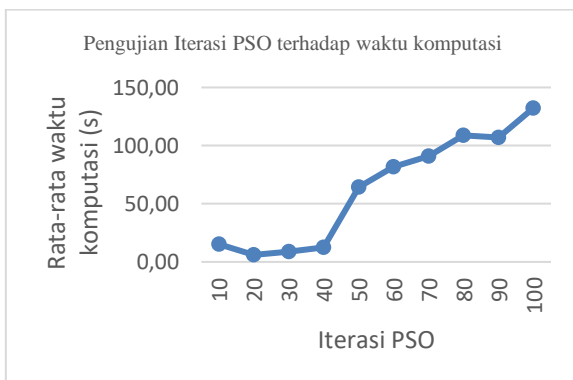
1. C : 1 – 200 (Cholissodin, et al., 2016)
2. ϵ : 0,00001 – 0,1 (Chai, 2011)
3. σ : 0,0001 – 2 (Cholissodin, et al., 2016)
4. γ : 0,1 – 1 (Gao, et al., 2016)
5. λ : 1 – 67 (Cholissodin, et al., 2016)
6. Jumlah fitur : 4
7. Iterasi SVR : 1000
8. Jumlah Partikel : 20 (Hseih, et al., 2011)
9. Iterasi PSO : 20
10. Data latih : data uji : 80% : 20%

4.1 Pengujian Iterasi PSO



Gambar 2 Grafik Hasil Pengujian Iterasi PSO

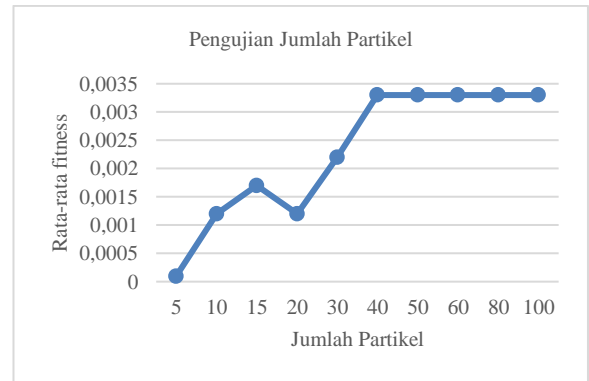
Berdasarkan Gambar 2 semakin banyak iterasi pada PSO maka semakin bagus nilai fitness yang didapatkan namun pada jumlah iterasi tertentu nilai fitness akan mengalami konvergensi. Pada penelitian ini fitness terbaik dicapai pada iterasi PSO = 40. Sedangkan berdasar grafik pengaruh iterasi terhadap lama waktu komputasi pada Gambar 3 menunjukkan bahwa semakin banyak iterasi yang digunakan maka akan menyebabkan waktu komputasi yang lama.



Gambar 3 Grafik Pengaruh Iterasi PSO terhadap Waktu Komputasi (s)

4.2 Pengujian jumlah partikel PSO

Berdasarkan Gambar 4 semakin besar jumlah partikel PSO akan menghasilkan nilai fitness yang semakin besar karena lingkup ruang solusi yang lebih besar. Namun setelah jumlah partikel tertentu terjadinya konvergensi dimana solusi maksimum tercapai dan tidak ada pergerakan *fitness* yang signifikan. Pada penelitian ini jumlah partikel yang menghasilkan solusi optimal pada jumlah partikel sebesar 40.

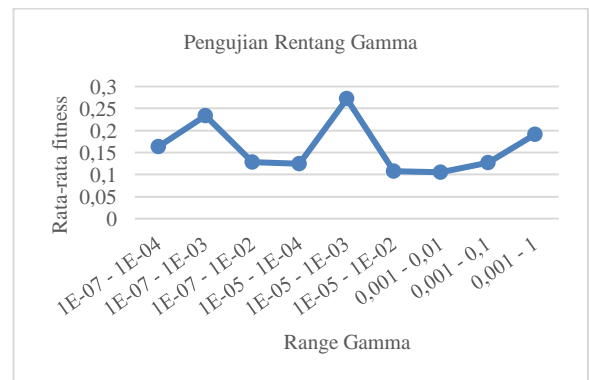


Gambar 4 Grafik Hasil Pengujian Jumlah Partikel

4.3 Hasil dan Analisa Rentang Pencarian Dimensi Parameter

Pengujian rentang pencarian parameter SVR dilakukan terhadap sejumlah parameter SVR yang akan dioptimasi yaitu C, ϵ (epsilon), gamma, lamda, dan sigma atau radial (σ).

4.3.1 Pengujian Rentang Pencarian Parameter Gamma (γ)



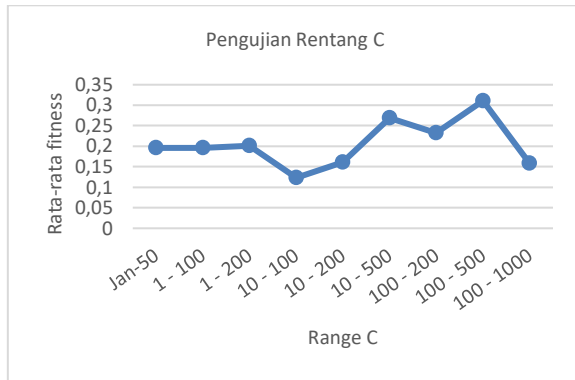
Gambar 5 Hasil Pengujian Rentang Pencarian Parameter Gamma

Berdasarkan Gambar 5 semakin tinggi nilai gamma maka nilai fitness yang dihasilkan akan semakin bagus. Nilai gamma memiliki pengaruh terhadap laju pembelajaran, jika γ bernilai kecil maka laju pembelajaran akan berjalan lebih lama sedangkan bila nilai γ besar maka laju pembelajaran akan berjalan lebih cepat namun akan cepat mencapai konvergensi pula. Pada penelitian ini rentang gamma optimal adalah pada 0.00001 – 0.001.

4.3.2 Pengujian Rentang Pencarian Parameter C

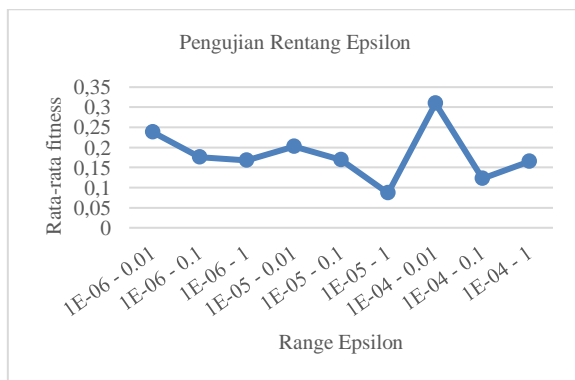
Berdasarkan grafik pada Gambar 6 semakin tinggi nilai C maka nilai fitness cenderung makin besar. Nilai C merupakan konstanta yang menampung nilai untuk penalti dari error untuk

model regresi yang dihasilkan, nilai C yang baik adalah nilai C yang tidak terlalu kecil sehingga menjamin error yang dihasilkan model regresi tidak terlalu besar (Chai, 2011). Untuk penelitian ini nilai C yang optimal didapatkan pada rentang 100 – 500.



Gambar 6 Grafik Hasil Pengujian Rentang Pencarian Parameter C

4.3.3 Pengujian Rentang Pencarian Parameter Epsilon (ϵ)



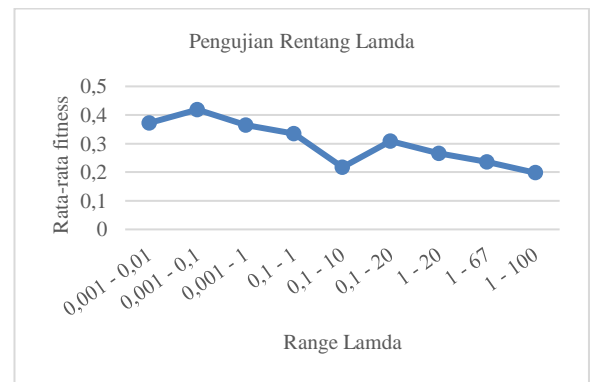
Gambar 7 Grafik Hasil Pengujian Rentang Pencarian Parameter Epsilon

Berdasarkan grafik pada gambar Gambar 7 semakin kecil batas bawah dari rentang pencarian ϵ akan menghasilkan nilai fitness yang semakin baik. Nilai ϵ merupakan nilai kerugian yang digunakan untuk mengontrol jumlah vektor pendukung, jika nilai ϵ semakin besar maka presisi dari suatu pola regresi akan semakin rendah dan semakin kecil support vectornya (Chai, 2011) untuk itu nilai ϵ semakin kecil maka akan semakin baik. Pada penelitian ini, rentang epsilon terbaik dicapai pada 0,0001 – 0,01.

4.3.4 Pengujian Rentang Pencarian Parameter Lamda (λ)

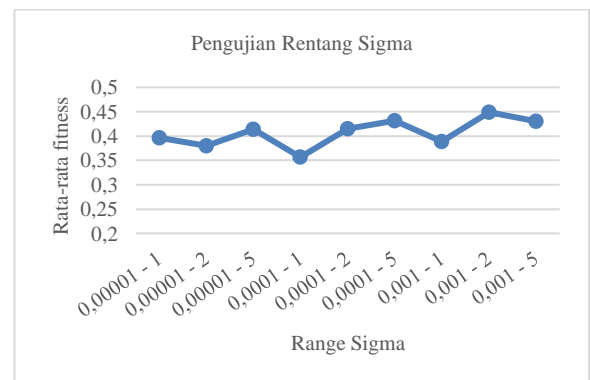
Berdasarkan grafik pada Gambar 8, semakin kecil nilai λ akan mengakibatkan nilai

fitness yang dihasilkan semakin bagus. λ merupakan learning rate dari kernel, semakin besar seharusnya semakin bagus tapi jika terlalu besar akan membutuhkan waktu komputasi yang lebih lama dan proses learning yang tidak stabil. Tapi saat menggunakan kernel non-linear, efek dari λ tidak dapat dilihat secara langsung karena bergantung pada kernel yang digunakan dan untuk permasalahan non-linear nilai λ mendekati 0 akan semakin bagus (Vijayakumar & Wu, 1999). Pada penelitian ini fitness optimal dicapai pada rentang λ 0,001 – 0,1.



Gambar 8 Grafik Hasil Pengujian Rentang Pencarian Parameter Lamda

4.3.5 Pengujian Rentang Parameter Sigma (σ)

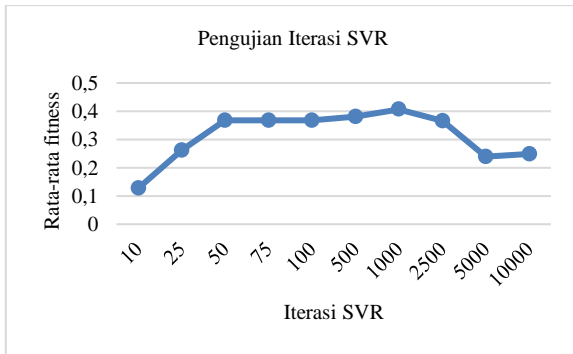


Gambar 9 Grafik Hasil Pengujian Rentang Pencarian Parameter Sigma

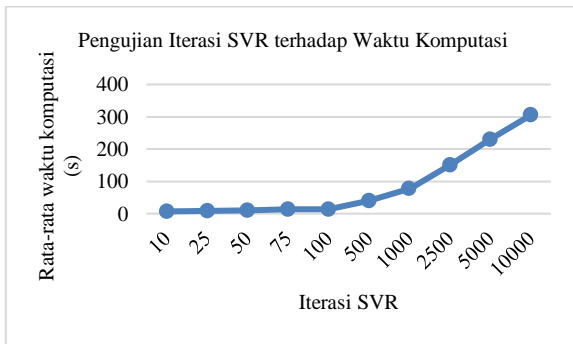
Berdasarkan grafik pada gambar Gambar 9 semakin tinggi nilai σ akan menyebabkan semakin baik fitness yang didapatkan namun dengan rentang yang semakin pendek. Sigma (σ) memiliki pengaruh dalam SVR sebagai parameter kernel RBF yang digunakan, jika σ bernilai terlalu kecil akan membuat terjadinya overfit pada data latih dan jika bernilai besar akan membuat komputasi berjalan lebih lama untuk fungsi yang lebih kompleks. Rentang σ

yang menghasilkan nilai fitness paling optimal pada penelitian ini pada 0,001 – 2.

4.4 Hasil dan Analisa Pengujian Iterasi SVR



Gambar 10 Grafik Hasil Pengujian Jumlah Iterasi SVR



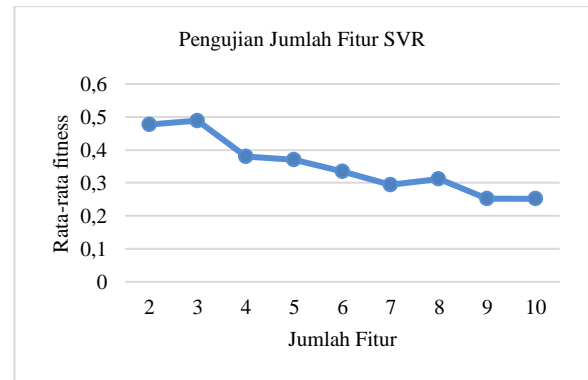
Gambar 11 Hasil Pengujian Iterasi SVR terhadap Waktu Komputasi

Berdasarkan Gambar 10 rata-rata fitness terbaik dicapai pada iterasi SVR ke-1000, dengan pada awalnya semakin tinggi iterasi SVR maka rata-rata fitness yang dicapai akan semakin tinggi pula. Namun setelah mencapai titik tertinggi dan konvergensi tercapai, semakin tinggi iterasi justru membuat rata-rata fitness turun. Pengaruh jumlah iterasi SVR terhadap lama waktu komputasi program berdasarkan grafik pada Gambar 11 yakni semakin tinggi iterasi SVR maka semakin lama waktu komputasi dari program.

4.5 Hasil dan Analisa Pengujian Jumlah Fitur SVR

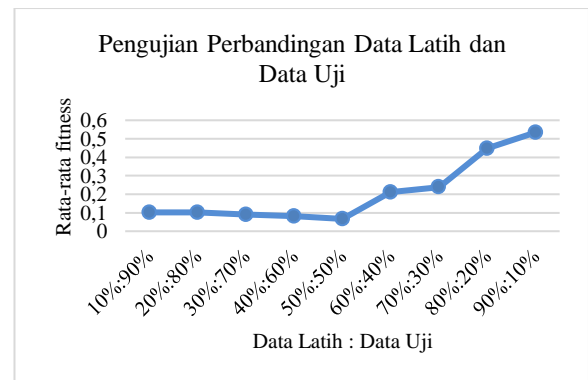
Berdasarkan grafik hasil pengujian pada Gambar 12 rata-rata fitness tertinggi didapatkan pada fitur berjumlah 3. Semakin banyak jumlah fitur yang digunakan rata-rata fitness cenderung menurun, jumlah fitur yang digunakan tidak ada ketetapan yang pasti, tiap data memiliki pola yang berbeda dan jumlah fitur yang digunakan pun berbeda. Untuk data saham pada penelitian ini, semakin sedikit semakin baik membuktikan

bahwa peramalan untuk harga saham Bank BCA hanya cocok untuk peramalan jangka pendek.



Gambar 12 Grafik Hasil Pengujian Jumlah Fitur SVR

4.6 Pengujian perbandingan data *training* dan data *testing*



Gambar 13 Grafik Hasil Pengujian Perbandingan Data Latih dan Data Uji

Berdasarkan grafik hasil pengujian pada Gambar 12 menunjukkan rata-rata fitness terbesar didapatkan pada perbandingan data latih dan data uji 90%:10%. Hasil pengujian juga menunjukkan bahwa semakin sedikit rasio data latih akan menyebabkan nilai fitness akan semakin kecil. Hal ini dikarenakan SVR membutuhkan proses *learning* menggunakan data latih untuk menghasilkan model regresi peramalan untuk meramalkan data uji. Sehingga semakin banyak data latih yang digunakan maka pola data yang dihasilkan akan semakin akurat untuk meramalkan data uji.

5. KESIMPULAN

Berdasarkan hasil penelitian peramalan harga saham menggunakan *Support Vector Regression* yang dioptimasi menggunakan *Particle Swarm Optimization* (PSO) didapatkan nilai MAPE sebesar 0,8195% dengan fitness

0,5496. Nilai MAPE tersebut berdasarkan Lewis (1982) dapat disimpulkan bahwa hasil peramalan menggunakan SVRPSO memiliki akurasi tinggi karena kurang dari 10%.

6. DAFTAR PUSTAKA

- Adebiyi, A. A., Adewumi, A. O. & Ayo, C. K., 2014. Stock Price Prediction Using the ARIMA Model. *UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pp. 105-111.
- Ayu, D., 2017. *Tak Butuh Duit Banyak Buat Investasi di Pasar Modal*. [Online] Available at: <http://bisnis.liputan6.com/read/2865270/tak-butuh-duit-banyak-buat-investasi-di-pasar-modal> [Accessed 22 Februari 2017].
- Bi, L., Tsimhoni, O. & Liu, Y., 2011. Using the Support Vector Regression Approach to Model Human Performance. *IEEE Transaction on Systems, Man, and Cybernetics - Part A: Systems and Humans*, pp. 410-417.
- Chai, Y., 2011. A Coal Mine Gas Concentration Prediction Method based on Particle Swarm Optimization Support Vector Regression. *IEEE*, pp. 334-337.
- Cholissodin, I., Farisuddin, F. & Santoso, E., 2016. Klasifikasi Tingkat Resiko Stroke Menggunakan Improved Particle Swarm Optimization dan Support Vector Machine. *Konferensi Nasional Sistem & Informasi 2016*, pp. 1-5.
- Gao, P., Hu, J., Zhou, H. & Yi, Z., 2016. Travel Time Prediction with Immune Genetic Algorithm and Support Vector Regression. *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, pp. 987-992.
- Hong, W. C., 2009. Electric Load Forecasting by Support Vector Model. *Aplied Mathematical Modelling*, pp. 2444-2454.
- Hsieh, H.-I., Lee, T.-P. & Lee, T.-S., 2011. A Hybrid Particle Swarm Optimization and Support Vector Regression Model for Financial Time Series Forecasting. *International Journal of Business Administration*, Volume 2, pp. 48-56.
- Hu, W., Yan, L., Liu, K. & Wang, H., 2015. PSO-SVR: A Hybrid Short-term Traffic Flow Forecasting Method. *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 553-561.
- Lewis, C. D., 1982. *International and Business Forecasting*. London: Butterworths.
- Lu, X. & Geng, X., 2011. Car Sales Volume Prediction Based on Particle Swarm Optimization Algorithm and Support Vector Regression. *Fourth International Conference on Intelligent Computation Technology and Automation*, pp. 72-74.
- Marini, F. & Walczak, B., 2015. Particle Swarm Optimization (PSO). A Tutorial. *Chemometrics and Intelligent Laboratory Systems*, Volume 149, pp. 153-165.
- Meesad, P. & Rasel, R. I., 2013. Predicting Stock Market Price Using Support Vector Regression. *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, pp. 1-6.
- Pasopati, G., 2017. *Frekuensi Transaksi Saham Indonesia Cetak Rekord Baru*. [Online] Available at: <http://www.cnnindonesia.com/ekonomi/20170209112717-92-192302/frekuensi-transaksi-saham-indonesia-cetak-rekord-baru/> [Accessed 22 Februari 2017].
- Pei-you, C. & Lu, L., 2013. Study on Coal Logistics Demand Forecast Based on PSO-SVR. *2013 10th International Conference on Service Systems and Service Management*, pp. 130-133.
- Peng, C. S. P., 2001. *Profit from Online Stock Investment*. Singapura: Mc-Graw-Hill Education.
- Rahmi, A., 2015. *Prediksi Harga Saham Berdasarkan Data Historis Menggunakan Model Regresi yang Dibangun dengan Algoritma Genetika*, Malang: Fakultas Ilmu Komputer Universitas Brawijaya.
- Ratnaweera, A., Halgamuge, S. K. & Watson, H. C., 2004. Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, Volume 8, pp. 240-255.
- Rodriguez, A., Neto, P. & Ferreira, T., 2009. A Prime Step in The Time Series Forecasting With Hybrid Methods: The Fitness Function Choice. *Proceedings of International Joint Conference on Neural Network, Atlanta, Georgia, USA*.
- Sreekumar, S., Sharma, K. C. & Bhakar, R., 2016. Optimized Support Vector Regression Models for Short Term Solar

- Radiation in Smart Environment. *2016 IEE Region 10 Conference (TENCON)*, pp. 1929-1932.
- Vapnik, V., Golowich, S. & Smola, A., 1997. *Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing*. Cambridge: MIT Press.
- Vijayakumar, S. & Wu, S., 1999. Sequential Support Vector Classifiers and Regression. *Proc. International Conference on Soft Computing (SOCO'99)*, pp. 610-619.
- Zubir, Z., 2013. *Manajemen Portofolio Penerapannya dalam Investasi Saham*. Jakarta: Salemba Empat.