

Prediksi Harga Saham menggunakan *Support Vector Regression* dan *Firefly Algorithm*

Stock Market Price Prediction using Support Vector Regression and Firefly Algorithm

Alfredo¹, Jondri², Rita Rismala³

^{1,2,3}Departemen Informatika, Universitas Telkom

Jalan Telekomunikasi No. 1, Dayeuh Kolot, Bandung 40257

djurisatu@gmail.com¹, @telkomuuniversity.ac.id², @gmail.com³

Abstrak

Dalam dunia investasi, saham merupakan salah satu instrumen pasar keuangan yang paling populer karena menjanjikan keuntungan yang lebih besar dari instrumen konvensional lain seperti deposito ataupun emas. Keuntungan tersebut didapat dari *dividen* (keuntungan dari hasil pembagian laba perusahaan) maupun *capital gain* (keuntungan yang diperoleh dari kelebihan nilai jual terhadap nilai beli saham). Akan tetapi harga suatu saham dapat berubah secara cepat dari waktu ke waktu dan para investor diharapkan untuk segera memutuskan kapan sebaiknya saham dijual atau tetap dipertahankan. Oleh karena itu dibutuhkan sistem yang dapat memprediksi pergerakan harga saham tersebut untuk membantu para investor dalam melakukan analisis dan tindakan yang tepat sehingga resiko dapat diminimalisir dan keuntungan dapat dioptimalkan.

Dalam Tugas Akhir ini, akan dibangun sebuah sistem yang melakukan prediksi terhadap harga saham menggunakan analisis teknikal yang diimplementasikan menggunakan *Support Vector Regression* dan *Firefly Algorithm*. *Support Vector Regression* (SVR) merupakan pengembangan dari metode support vector machine untuk kasus regresi. Metode ini mampu mengatasi overfitting serta mampu menunjukkan performa yang bagus. Akan tetapi terdapat kelemahan pada SVR dalam menentukan nilai parameter yang paling optimal untuk digunakan. Untuk mengatasi kelemahan tersebut digunakanlah algoritma optimasi *Firefly Algorithm* untuk mencari nilai parameter SVR yang paling optimal.

Database yang digunakan pada Tugas Akhir ini menggunakan data historis pergerakan harga empat saham *blue chip* yang mengacu pada *finance.yahoo.com* periode 2010 - 2014. Hasil penelitian menunjukkan bahwa SVR dan FA dapat diimplementasikan sebagai metode untuk memprediksi harga saham dengan *error* kurang dari 5%.

Kata kunci: Prediksi harga saham, Time Series, Support Vector Regression (SVR), Firefly Algorithm (FA).

Abstract

On the investing world, holdings/stock is one of the most popular financial market instruments because of it benefits to get better profits rather than other conventional instruments such as deposits or gold. The profits comes from dividends (profits from the distribution of company profits) and capital gains (profits on the sale value of the excess value of shares).Basically in the stock market, the price of a stock can change rapidly over time and the investor must decide when the stock should be sold or retained. Because of that we need a system that can give prediction of the stock price to help investors in taking the right action so that the risk can be minimalized.

In this final project , a software will be built to predict stock price movements using technical analysis which implemented using *Support Vector Regression* (SVR) and *Firefly Algorithm* (FA). SVR is a regression method which developed from Support Vector Machine and able to overcome the overfitting and demonstrate a good performance. However, SVR have weakness in determining parameter value that fits for it. To overcome the weakness, *Firefly Algorithm* (FA) is used to determine the optimal value for SVR parameters.

By using the dataset from *finance.yahoo.com* , which is four *bluechip* stock price from period 2010-2014, this final project gets a result that shows that SVR and FA algorithm can be applied in the stock price prediction with the error rate below 4%.

Keywords: Stock market prediction, Support Vector Regression (SVR), Firefly Algorithm (FA), financial time series.

1 PENDAHULUAN

Di era yang serba maju sekarang ini, mencari uang sudah tidak terfokus lagi dari pekerjaan yang dimiliki oleh seseorang. Banyak hal yang dapat dilakukan untuk mencari tambahan uang selain dari pekerjaan tetap seseorang, salah satunya adalah investasi. Investasi dapat dilakukan dalam berbagai bentuk, salah satunya adalah investasi dalam bentuk saham. Saham dikeluarkan oleh perusahaan untuk mendapatkan modal tambahan dengan cara menjualnya kepada investor. Investor membeli saham dengan tujuan mendapat keuntungan dari *dividen* (keuntungan dari pembagian laba perusahaan) ataupun *capital gain* (keuntungan dari kelebihan nilai jual terhadap nilai beli saham). Di dalam dunia investasi, khususnya saham, peramalan/prediksi merupakan suatu hal yang penting bagi para investor, karena dalam setiap transaksi perdagangan saham investor dihadapkan pada pilihan untuk membeli atau menjual. Setiap kesalahan dalam mengambil keputusan akan menimbulkan kerugian pada investor itu sendiri. Oleh sebab itu dibutuhkan suatu sistem yang dapat menganalisa dan memprediksi harga saham untuk menjadi bahan pertimbangan bagi para investor untuk mengambil keputusan yang tepat.

Menurut teori *efficient market hypothesis*, harga saham lampau telah merepresentasikan kondisi saat itu, serta banyak penelitian yang menggunakan data historis karena lebih mudah diperoleh dan tidak serumit menggunakan data fundamental. Oleh karena itu pada tugas akhir ini dibuat sebuah sistem untuk memprediksi harga saham dengan menggunakan data historis yang mengimplementasikan *Support Vector Regression* dengan optimasi algoritma *Firefly*. SVR merupakan metode regresi dari SVM (*Support Vector Machine*) yang biasa digunakan untuk mengatasi *overfitting* dan memiliki performansi yang baik untuk kasus regresi. Akan tetapi, untuk mendapatkan performansi yang baik dibutuhkan pula parameter yang tepat. Selain membutuhkan parameter yang tepat, penentuan fitur adalah salah satu hal yang penting untuk melatih SVR. Maka dari itu dibutuhkan suatu metode optimasi untuk mencapai performa model SVR yang akurat.

Firefly Algorithm merupakan salah satu metode yang dapat digunakan untuk memberikan solusi yang optimal untuk kasus optimasi. Solusi yang dihasilkan lebih baik daripada algoritma optimasi lainnya seperti GA (*Genetic Algorithm*) dan PSO (*Particle Swarm Optimization*) berdasarkan kasus *Travelling Salesman Problem*. Maka pada tugas akhir ini akan dibuat sistem berdasarkan 2 metode diatas dan mengukur performansinya untuk kasus prediksi saham.

2 LANDASAN TEORI

2.1 Peramalan

Peramalan adalah proses memperkirakan sesuatu yang akan terjadi di masa yang akan datang menggunakan informasi lampau dan saat ini. Peramalan memberikan kondisi yang belum pasti tentang apa yang akan terjadi, akan tetapi biasanya hasil dari peramalan digunakan sebagai acuan agar kesalahan dapat diperkecil.

Peramalan dibagi menjadi dua bagian yaitu [8]:

1. Peramalan Kualitatif

Peramalan kualitatif didasarkan atas data kualitatif masa lalu. Peramalan ini digunakan jika data dari masa lalu dari variabel yang akan diprediksi tidak ada. Hasil prediksi sangat bergantung pada orang yang menyusunnya karena hasil peramalan ini ditentukan berdasarkan pemikiran yang bersifat *judgement* atau opini, pengetahuan dan pengalaman dari penyusun.

2. Peramalan Kuantitatif

Peramalan kuantitatif didasarkan atas data kuantitatif masa lalu. Hasil peramalan sangat bergantung pada metode yang digunakan dalam peramalan. Penggunaan metode yang berbeda akan menghasilkan hasil yang berbeda pula. Metode yang baik adalah metode yang memberikan nilai perbedaan yang mungkin. Peramalan kuantitatif hanya dapat digunakan apabila terdapat tiga kondisi berikut:

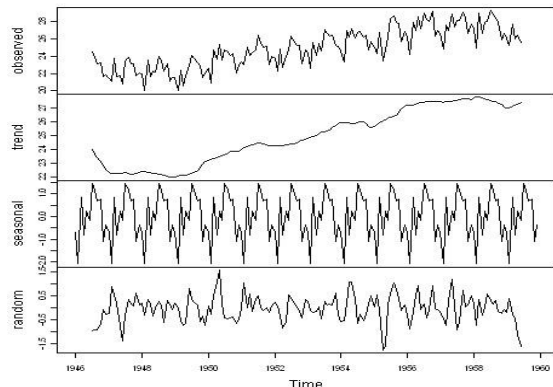
- a) Adanya informasi tentang keadaan yang lain
- b) Informasi tersebut dapat dikuantifikasikan dalam bentuk data numeric
- c) Diasumsikan pola masa lalu akan berkelanjutan pada masa yang akan datang.

2.2 Time Series

Prediksi secara *time series* merupakan peramalan yang dilakukan dengan menggunakan data tertentu yang disebut data *time series*. Data *time series* adalah nilai suatu variabel yang disusun berdasarkan urutan/deret waktu bisa dalam satuan harian, mingguan, bulanan, atau tahunan tergantung keperluan terhadap perilaku data yang akan diobservasi. Data *time series* yang digunakan adalah data historis yang diukur berdasarkan suatu pengamatan tertentu. Karena data yang digunakan adalah data yang terukur, maka peramalan secara *time series*, termasuk ke dalam peramalan kuantitatif [8]. Metode prediksi *time series* beranggapan bahwa data atau kejadian masa lalu akan cenderung berulang dimasa yang akan datang [9]. Fokus prediksi secara *time series* adalah apa yang akan terjadi, bukan mengapa hal itu terjadi. Hal yang harus diperhatikan sebelum melakukan prediksi *time series* adalah :

1. Adanya ketergantungan masa depan dengan masa lampau.
2. Aktivitas masa yang akan datang mengikuti pola masa lalu.

3. Hubungan keterkaitan masa lalu dan masa kini dapat ditentukan melalui pengamatan.



Gambar 2.1 Pola data time series

Berikut empat pola data time series [9]:

1. Pola Data Konstan
Pola data konstan terjadi apabila data berfluktuasi secara stabil. Pola data konstan berupa garis horizontal. Pola data konstan umumnya terdapat pada data yang memiliki periode waktu jangka pendek atau menengah.
2. Pola Data Trend
Pola data trend terjadi apabila data cenderung meningkat atau menurun dari waktu ke waktu dalam jangka panjang. Pola ini disebabkan antara lain oleh bertambahnya populasi, perubahan pendapat, dan pengaruh budaya.
3. Pola Data Musiman
Pola data musiman terjadi apabila data berfluktuasi berulang-ulang secara teratur dalam setiap periode tentu, misalnya tahunan, semesteran, kuartalan, bulanan atau mingguan.

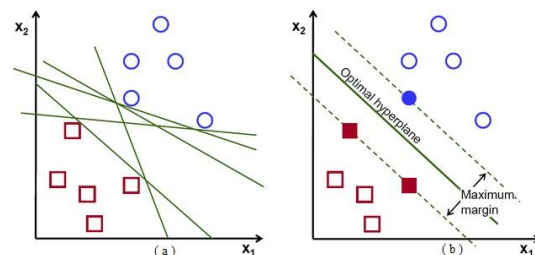
Pola ini berhubungan dengan faktor iklim / cuaca atau faktor yang dibuat manusia seperti liburan.
4. Pola Data Siklus
Pola data siklus terjadi apabila data dipengaruhi oleh fluktuasi ekonomi jangka panjang, seperti daur hidup bisnis. Perbedaan utama antara pola data musiman dan siklus adalah pola musiman mempunyai panjang gelombang yang tetap dan terjadi pada jarak waktu tetap, sedangkan pola siklus memiliki durasi yang lebih panjang dan bervariasi.
5. Pola Data Residual atau Variasi Acak
Pola data residu terjadi apabila fluktuasi data tidak teratur sama sekali. Data yang bersifat residu tidak dapat digambarkan.

2.3 Support Vector Regression

SVR merupakan penerapan konsep *Support Vector Machine* (SVM) untuk kasus regresi. Output berupa bilangan riil atau kontinu dalam kasus

regresi. SVR merupakan metode yang dapat

kinerja yang bagus. SVR digunakan untuk menemukan suatu fungsi $f(x)$ yang mempunyai deviasi ϵ paling besar dari nilai aktual y , untuk semua data *training*. Jika nilai $\epsilon = 0$, maka kita dapatkan suatu regresi yang sempurna (Santosa 2007). *Support Vector Machine* (SVM) pertama kali diperkenalkan oleh Boser, Guyon, dan Vapnik pada tahun 1992. Konsep dasar SVM merupakan kombinasi dari teori-teori komputasi yang telah ada, seperti *margin hyperplane* oleh Duda dan Hart pada tahun 1973, *kernel* yang diperkenalkan oleh Aronszajn pada tahun 1950, dan konsep pendukung lainnya. Menurut Nugroho *et al.* (2003), prinsip dasar SVM adalah *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada masalah non-linear, dengan memasukkan konsep *kernel trick*. Gambar 2-2a memperlihatkan beberapa pola yang merupakan anggota dari dua buah kelas : +1 dan -1. Pola yang tergabung pada kelas -1 disimbolkan dengan kotak, sedangkan pola pada kelas +1 disimbolkan dengan lingkaran. Masalah klasifikasi ini dapat diselesaikan dengan usaha menemukan garis (*hyperplane*) maksimum yang memisahkan antara kedua kelas tersebut (Nugroho *et al.* 2003) (Gambar 2-2b).



Gambar 2.2 Hyperplane

Misal terdapat fungsi berikut sebagai garis regresi:

$$f(x) = w^T \phi(x) + b \tag{2.1}$$

Dimana $\phi(x)$ menunjukkan suatu titik didalam

feature space F hasil pemetaan x di dalam input space. Koefisien w dan b diestimasi dengan cara meminimalkan fungsi resiko (*risk function*) yang didefinisikan dalam persamaan:

$$\min \frac{1}{2} \|w\|^2 + C \frac{1}{\lambda} \sum_{i=1}^{\lambda} L_{\epsilon}(y_i, f(x_i)) \tag{2.2}$$

Subject to

$$\begin{aligned} y_i - w\phi(x_i) - b &\leq \epsilon \\ w\phi(x_i) - y_i + b &\leq \epsilon \\ i &= 1, 2, \dots, \lambda \end{aligned}$$

Dimana

$$L_{\epsilon}(y_i, f(x_i)) = |y_i - f(x_i)| - \epsilon \quad |y_i - f(x_i)| \geq 0 \tag{2.3}$$

Faktor $\|w\|^2$ adalah regualisasi. Agar

mengatasi *overfitting*, sehingga akan menghasilkan

kapasitas fungsi dapat dikontrol maka fungsi harus

dibuat setipis mungkin dengan cara meminimalkan $\|w\|^2$ Faktor kedua dalam fungsi tujuan adalah kesalahan empirik (*empirical error*) yang diukur dengan ϵ -insensitive lossfunction. Dengan menggunakan ide ϵ -insensitive lossfunction norm

dari w harus diminimalkan agar mendapatkan generalisasi yang baik untuk fungsi regresi f . Karena itu diperlukannya menyelesaikan problem

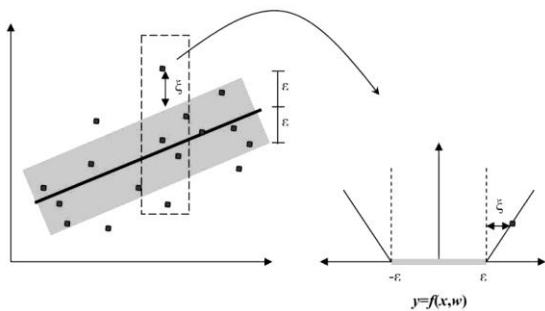
estimasi berikut:

$$\min \frac{1}{2} \|w\|^2 \tag{2.4}$$

Subject to

$$\begin{aligned} y_i - w\phi(x_i) - b &\leq \epsilon \\ w\phi(x_i) - y_i + b &\leq \epsilon \\ i &= 1, 2, \dots, \lambda \end{aligned}$$

$$\begin{aligned} \sum e^2 &= \sum (y - \hat{y})^2 \\ &= \sum (y - (b_0 + b_1 x))^2 \end{aligned}$$



Gambar 2.3 ϵ -insensitive loss function pada SVR

Asumsikan bahwa ada suatu fungsi f yang dapat mengaproksimasi semua titik (x_i, y_i) dengan presisi ϵ . Diasumsikan bahwa semua titik ada dalam rentang $f \pm \epsilon$ atau disebut *feasible*. Sedangkan *infeasible* merupakan kondisi dimana

ada beberapa titik yang keluar dari rentang $f \pm \epsilon$, titik yang *infeasible* ini bisa ditambahkan variabel

slack ξ, ξ^* untuk mengatasi masalah pembatas

yang tidak layak (*infeasible constrain*) dalam masalah optimasi. Selanjutnya masalah optimasi diatas bisa diformulasikan sebagai berikut:

$$\min \frac{1}{2} \|w\|^2 + C \frac{1}{2} \sum (\xi_i, \xi_i^*) \tag{2.5}$$

deviasi lebih besar daripada ϵ akan dikenakan penalti sebesar C . Dalam SVR, ϵ ekuivalen dengan akurasi dari aproksimasi terhadap data training. nilai ϵ yang kecil akan dikaitkan dengan nilai yang tinggi pada variabel slack (*)

ξ_i dan akurasi aproksimasi yang tinggi. Sebaliknya, nilai yang tinggi untuk ϵ berkaitan dengan nilai ξ_i^*

yang kecil dan aproksimasi yang rendah. Menurut persamaan (2.5) nilai yang tinggi untuk variabel slack akan membuat kesalahan empirik mempunyai pengaruh yang besar terhadap faktor regulasi. Dalam SVR, *support vector* adalah data training yang terletak pada dan diluar batas f dari fungsi keputusan, karena itu jumlah *support vector* menurun dengan naiknya ϵ .

Dalam formulasi dual, problem SVR adalah sebagai berikut:

$$\begin{aligned} \max - \frac{1}{2} & \sum_{i=1}^{\lambda} (\alpha_i - \alpha_i^*) (\alpha_i - \alpha_i^*) K(x_i, x_j) + \sum_{i=1}^{\lambda} (\alpha_i - \alpha_i^*) y_i - \epsilon \sum_{i=1}^{\lambda} (\alpha_i - \alpha_i^*) \end{aligned} \tag{2.6}$$

Subject to

$$\begin{aligned} \sum_{i=1}^{\lambda} (\alpha_i - \alpha_i^*) &= 0 \\ 0 \leq \alpha_i &\leq C, i = 1, 2, \dots, \lambda \\ 0 \leq \alpha_i^* &\leq C, i = 1, 2, \dots, \lambda \end{aligned}$$

Dimana C didefinisikan oleh user, $K(x_i, x_j)$ adalah dot-product kernel yang

didefinikan sebagai $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$

Dengan menggunakan *lagrange multiplier* dan kondisi optimalitas, fungsi regresi secara eksplisit dirumuskan sebagai berikut:

$$f(x) = \sum_{i=1}^{\lambda} (\alpha_i - \alpha_i^*) K(x_i, x_j) + b \tag{2.7}$$

2.4 Firefly Algorithm

Algoritma Kunang-kunang adalah sebuah algoritma metaheuristik yang terinspirasi dari perilaku berkedip kunang-kunang untuk berkomunikasi satu sama lain. Algoritma ini biasanya digunakan untuk

Subject to

$\lambda_{i=1}$

kasu
s
opti
masi
kare
na
jauh
lebih
seder
hana
baik
d
a
l
a
m
k
o
n
s
e
p
m
a
u
p
u
n
i
m
p
l
e
m
e
n
t
a
s
i
,
s
e
r
t
a
m
a
m
p
u
m
e
n
g
h
a
s
i
l
k
a
n
s
o
l
u

si optimal baik dan benar.

$$y_i - w_i^T \phi(x_i) - b - \xi_i \leq \epsilon, i = 1, 2, \dots, \lambda$$

$$w_i^T \phi(x_i) - y_i + b - \xi_i^* \leq \epsilon, i = 1, 2, \dots, \lambda$$

$$\xi_i, \xi_i^* \geq 0$$

Konstanta C menentukan tawar-menawar (*trade off*) antara ketipisan fungsi f dan batas atas deviasi lebih dari ϵ yang masih ditoleransi. Semua

Algoritma yang terinspirasi dari kunang-kunang ini dibuat oleh *Xin-She Yang* pada tahun 2008. FA memiliki aturan sebagai berikut [10]:

1. Semua kunang-kunang adalah *unisex* yang berarti setiap kunang-kunang akan tertarik satu sama lain.

2. Tingkat daya tarik (*attractiveness*) kunang-kunang proposional dengan tingkat kecerahan atau intensitas cahaya (*light intensity*) yang dimilikinya. Kunang-kunang yang memiliki cahaya yang kurang terang akan tertarik dan bergerak ke kunang-kunang yang memiliki cahaya lebih terang. Jika tidak ada kunang-kunang yang memiliki cahaya yang lebih terang dari dirinya, maka kunang-kunang akan bergerak

secara *random* (acak).

3. Kecerahan atau intensitas cahaya kunang-

kunang ditentukan oleh nilai fungsi objektif dari masalah yang diberikan. Untuk masalah maksimisasi, intensitas cahaya sebanding dengan nilai fungsi objektif.

Parameter yang terdapat dalam algoritma firefly adalah:

1. Intensitas cahaya (*I*)

Intensitas cahaya menyatakan tingkat kecerahan dari kunang-kunang. Semakin cerah seekor kunang-kunang, maka dikatakan semakin bagus solusi kunang-kunang tersebut. Intensitas cahaya proposional dengan fungsi objektif. Intensitas cahaya dalam kasus ini adalah nilai *error* masing-masing kunang-kunang.

2. *Attractiveness*

Daya tarik yang diindikasikan dengan cerah tidaknya cahaya yang dihasilkan oleh kunang-kunang. Besar kecilnya daya tarik bersifat relatif yang tergantung pada seberapa jauh kunang-kunang. Dalam FA, fungsi ini merupakan fungsi paling utama yang dapat dirumuskan sebagai berikut:

$$a_{ij} = \frac{1}{m+1} \quad \text{dimana } (m \geq 1) \quad (2.7)$$

3. Jarak

Semakin jauh jarak antar kunang-kunang, semakin redup cahaya dari kunang-kunang yang berkedip. Jarak antara dua kunang-kunang *i* dan *j* pada x_i dan x_j adalah jarak kartesian dengan rumus sebagai berikut:

$$r_{ij} = \sqrt{\sum_{k=1}^D (x_{ik} - x_{jk})^2} \quad (2.8)$$

4. *Movement*

Pergerakan kunang-kunang, kunang-kunang *i* bergerak menuju kunang-kunang *j* yang lebih terang dengan rumus sebagai berikut:

$$x_i = x_i + \text{rand}() \cdot a_{ij} \cdot r_{ij} \cdot (x_j - x_i) + \text{rand}() \cdot \frac{1}{2} \cdot (x_i - x_i) \quad (2.9)$$

2.5 Pengukuran hasil prediksi

Untuk mengetahui seberapa besar tingkat kesalahan / *error* pada tugas akhir ini, digunakan MAPE (*Mean Absolute Percentage Error*). MAPE merupakan ukuran akurasi yang menunjukkan seberapa besar kesalahan prediksi dibandingkan dengan nilai sebenarnya. Akurasi dari MAPE diekspresikan dalam bentuk persentase. Semakin kecil nilai MAPE yang dihasilkan maka semakin baik akurasi performansi dari sistem prediksi yang dibangun. Berikut rumu perhitungan MAPE :

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{D_i - P_i}{D_i} \right| \quad (2.10)$$

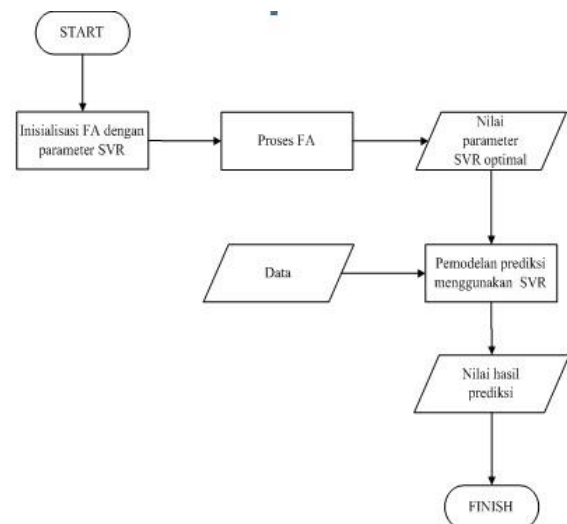
dimana D_i merupakan nilai aktual dan P_i merupakan nilai hasil prediksi.

3. IMPLEMENTASI SISTEM

Sistem ini menggunakan data historis 4 saham bluechip dengan rincian sebagai berikut:

Data Historis Saham				Periode dan jenis data	
Nike (NKE)	Data Harian	Data mingguan	Data Bulanan	2010-2012	Data latih
NVIDIA (NVIDIA)				2013-2014	Data uji
AMD (AMD)					
IBM (IBM)					

Adapun deskripsi sistem secara garis besar digambarkan pada flowchart pada gambar 3.1

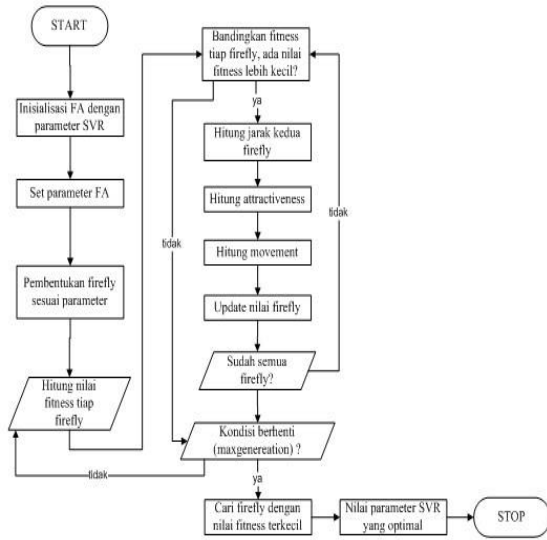


Gambar 3.1 Flowchart sistem

Pada proses training, sistem menggunakan FA untuk mendapatkan nilai parameter optimal dari SVR. Nilai paling optimal diambil dari firefly dengan nilai fitness paling baik. Pada proses pengujian, nilai optimal hasil keluaran dari FA dijadikan sebagai inputan pada SVR dan diuji untuk diketahui performansinya dalam

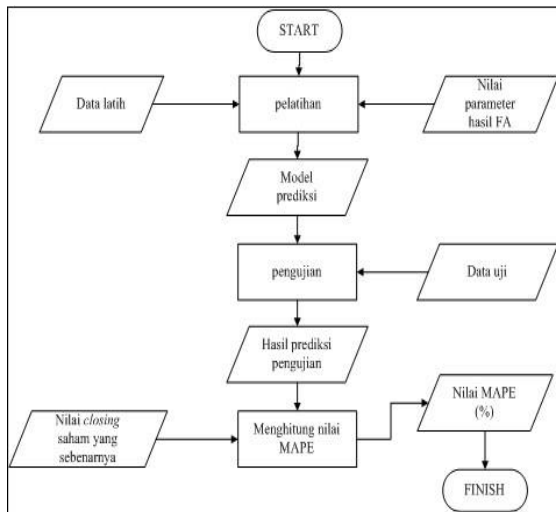
melakukan prediksi nilai *closing* dari indeks saham yang digunakan. MAPE digunakan untuk

mengukur performansi dari sistem prediksi yang telah dibangun.



Gambar 3.2 Proses FA

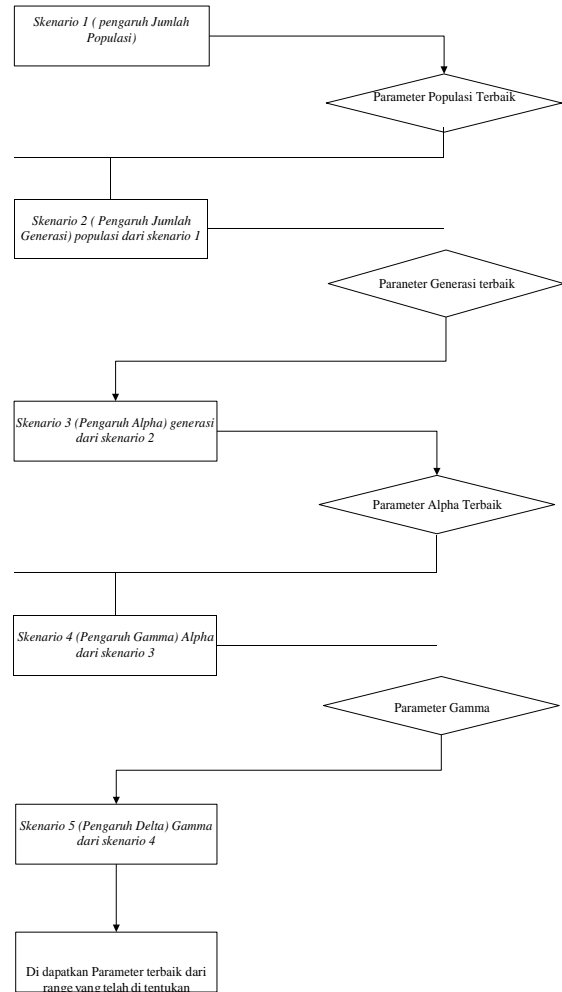
Setelah didapatkan hasil dari proses FA yang berupa parameter nilai C dan epsilon, dilakukan pelatihan dan pengujian menggunakan SVR dengan flowchart seperti gambar 3.3



Gambar 3.4 Flowchart pelatihan dan pengujian pada SVR

4 PENGUJIAN DAN ANALISIS

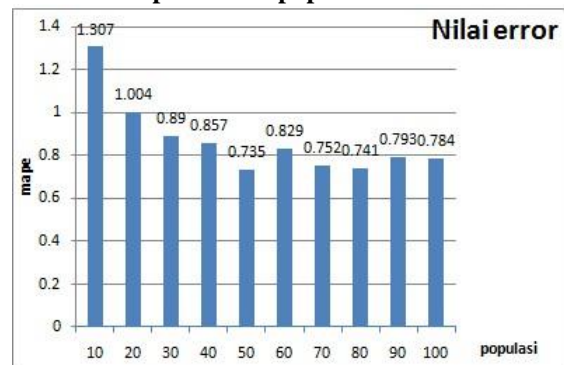
Strategi pengujian menggunakan 5 skenario untuk proses FA yang berkorelasi satu dengan yang lainnya seperti pada gambar dibawah ini



Gambar 4.1 Skenario pengujian

Sedangkan pada proses SVR, hanya dilakukan pengujian untuk memprediksi harga saham dengan memakai nilai output dari proses FA.

4.1 Analisis parameter populasi

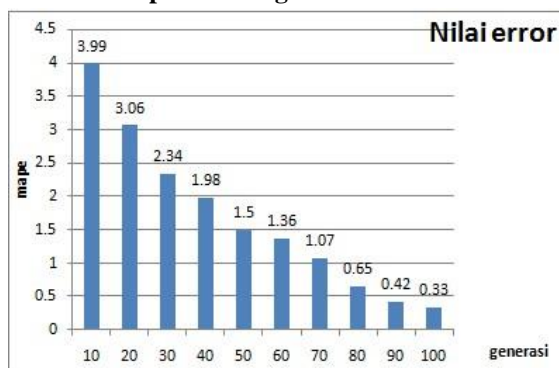


Suatu firefly pada Firefly Algorithm merepresentasikan suatu kandidat solusi. Jumlah firefly merepresentasikan sekumpulan firefly yang terdapat dalam suatu waktu atau yang biasa dikenal dengan istilah populasi. Firefly Algorithm mempunyai cara kerja yang parallel yang artinya pada setiap generasi, Firefly Algorithm akan membangkitkan, menguji dan berusaha memperbaiki sekumpulan firefly tersebut sampai

ditemukan satu solusi yang dapat diterima. Semakin banyak jumlah firefly atau ukuran populasi yang diberikan, maka semakin banyak pula pada setiap generasi terjadi proses evolusi dari suatu firefly menjadi firefly yang lebih baik. Semakin banyak sebuah firefly berevolusi, hal ini akan mengakibatkan tingkat error yang dimiliki oleh firefly tersebut semakin kecil. Pada akhirnya, semakin besar jumlah firefly yang ditentukan maka semakin kecil pula hasil error yang akan didapat. Namun, kerugiannya jika jumlah firefly yang diberikan besar maka waktu selesai program akan semakin lama. Dalam kondisi terburuk, komputasi yang terjadi pada suatu generasi pada Firefly Algorithm dapat sebanyak $O(n^2)$. Hal ini menunjukkan dalam kondisi terburuk, lama waktu komputasi yang dilakukan akan meningkat dengan tajam. Selain itu, pemberian jumlah firefly yang tidak tepat akan menyebabkan kondisi overfit.

Pada Tabel diatas, jumlah firefly yang baik terdapat pada pengujian ke-2 dengan nilai 20. Seperti yang telah disinggung sebelumnya, bahwa semakin besar jumlah firefly maka semakin kecil pula error yang akan dicapai. Namun, perlu diperhatikan bahwa apakah solusi suatu firefly yang didapat merupakan solusi yang baik untuk data latih maupun data uji. Pada observasi kali ini, didapat kesimpulan bahwa penggunaan jumlah firefly sebanyak 20 akan menghasilkan solusi yang memiliki nilai error yang lebih kecil jika dibandingkan dengan error firefly lain.

4.2 Analisis parameter generasi

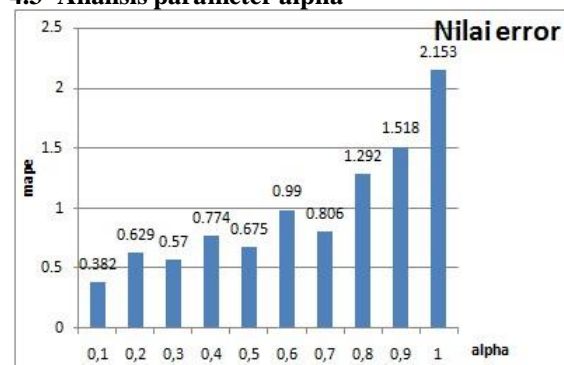


Jumlah generasi/iterasi merepresentasikan lamanya proses pembelajaran yang dilakukan terhadap jaringan yang sedang diobservasi. Jumlah generasi/iterasi menentukan kapan proses pembelajaran dihentikan. Semakin besar nilai jumlah generasi/iterasi, maka semakin lama pula proses pembelajaran berlangsung. Begitu juga sebaliknya. Jumlah generasi/iterasi yang terlalu sedikit, mengakibatkan jaringan yang terbentuk bersifat terlalu general/umum. Artinya kemampuan jaringan dalam mengenali pola terlalu sedikit atau bahkan tidak ada sama sekali. Kondisi ini dinamakan overweight. Overweight akan

mengakibatkan jaringan yang terbentuk akan menganggap setiap data yang dimasukkan bersifat sama sehingga output yang dihasilkan untuk setiap data akan sama. Dengan kata lain, jaringan yang mengalami kondisi overweight akan kehilangan kemampuannya dalam mengenali pola dari data yang diberikan. Jumlah generasi/iterasi yang terlalu banyak, akan mengakibatkan jaringan mengalami kondisi overfit. Kondisi overfit mempunyai arti bahwa jaringan bersifat terlalu spesifik terhadap data pelatihan. Akibat dari overfit adalah ketika proses pengujian berlangsung, jaringan akan mengalami kegagalan dalam menghasilkan output yang sesuai terhadap data pengujian dikarenakan jaringan kehilangan kemampuan generalisasinya. Oleh karena itu, dikatakan bahwa jaringan yang mengalami kondisi overfit akan bagus dalam mengenali pola data pelatihan namun buruk dalam mengenali pola dari data pengujian. Jumlah generasi/iterasi yang tepat akan menentukan kemampuan akhir jaringan dalam melakukan memorialisasi dan generalisasi. Memorialisasi adalah kemampuan jaringan dalam mengenali pola yang telah dilatihkan sedangkan generalisasi adalah kemampuan jaringan dalam mengenali pola yang belum pernah dilatihkan.

Pada Tabel diatas, jumlah generasi terbaik terdapat pada pengujian ke-10 dengan jumlah generasi sebanyak 100.

4.3 Analisis parameter alpha

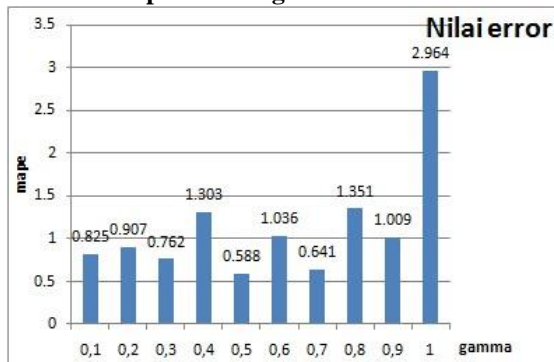


Konstanta alpha pada Firefly Algorithm berpengaruh pada pergerakan kunang-kunang atau movement. Nilai alpha berguna untuk melakukan randomisasi seberapa besar suatu kunang-kunang bergerak. Konstanta alpha dapat diibaratkan seperti halnya learning rate pada jaringan syaraf tiruan. Learning rate dalam jaringan syaraf tiruan harus mempunyai besaran yang tidak terlalu kecil ataupun tidak terlalu besar. Jika learning rate bernilai terlalu besar, proses pembelajaran akan mengalami keadaan "stagnan". Artinya fungsi pembatas yang terbentuk tidak dapat memetakan data dengan baik. Hal ini dikarenakan proses belajarnya yang terlalu cepat sehingga diakhir epoch, fungsi pembatas tidak dapat melakukan pembelajaran lagi yang mengakibatkan nilai error diakhir epoch bernilai besar. Jika nilai learning rate

terlalu kecil, error pada akhir epoch pun akan bernilai besar. Hal ini dikarenakan proses pembelajaran yang terlalu lambat sehingga dibutuhkan waktu yang lama untuk mencapai solusi yang diharapkan. Biasanya jika nilai learning rate kecil, maka jumlah epoch harus bernilai besar untuk mengimbangi antara banyaknya pembelajaran dengan lamanya pembelajaran.

Pada Tabel diatas, nilai alpha yang terbaik terdapat pada pengujian ke-1 yang bernilai 0.1. Jika nilai alpha bernilai 1 maka kunang-kunang akan bergerak secara bebas dan tidak teratur. Namun, jika alpha 0.1, kunang-kunang akan bergerak secara bebas namun teratur.

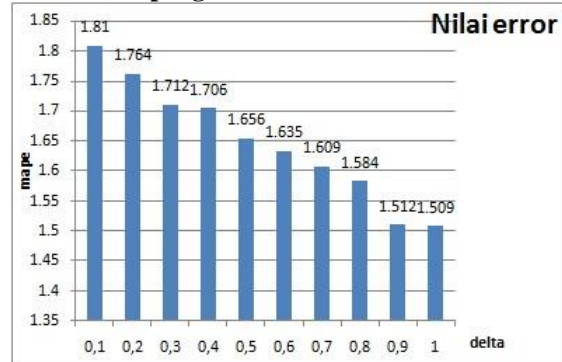
4.4 Analisis parameter gamma



Konstanta gamma pada Firefly Algorithm memiliki pengaruh pada nilai attractiveness. Besar kecilnya nilai gamma akan berpengaruh pada daya tarik antar kunang-kunang yang pada akhirnya akan berpengaruh pada pergerakan kunang-kunang. Konstanta gamma sendiri memiliki definisi sebagai koefisien absorpsi cahaya. Pada dasarnya, dapat dikatakan bahwa udara dapat menyerap cahaya. Hal ini dapat dilihat bahwa pada kondisi udara yang baik, cahaya akan terlihat. Akan tetapi, saat kondisi udara berkabut, intensitas cahaya akan berkurang sehingga cahaya akan terlihat semakin redup. Jika nilai gamma bernilai terlalu kecil bahkan sama dengan 0, maka yang terjadi adalah nilai attractiveness mendekati atau bahkan sama dengan

1. Hal ini berarti dalam kondisi apapun suatu firefly akan tertarik dengan firefly lainnya tidak bergantung pada perbedaan jarak yang dimilikinya. Selain itu, jika nilai attractiveness sama dengan 1, maka akan mengakibatkan nilai suatu bobot akan bernilai lebih dari atau minimal 1. Padahal rentang nilai bobot bernilai (-1,1). Pada hasil analisis tabel diatas, nilai konstanta gamma yang terbaik terdapat pada pengujian ke-5 yang bernilai 0.5.

4.5 Analisis pengaruh konstanta m



Konstanta m merupakan sebuah konstanta yang dipakai pada persamaan (2.8). Konstanta ini merepresentasikan pengaruh seberapa besar jarak terhadap nilai attractiveness. Attractiveness merupakan daya tarik yang diindikasikan dengan cerah tidaknya cahaya yang dihasilkan oleh kunang-kunang. Nilai attractiveness ini akan berpengaruh pada penentuan movement pada persamaan (2.9). Semakin besar nilai konstanta m yang ditentukan, semakin kecil nilai attractiveness antara kunang-kunang yang satu dengan kunang-kunang yang lain. Jika nilai attractiveness kecil, maka besarnya perpindahan atau nilai movement akan semakin pula. Hal ini mengakibatkan program akan memakan waktu yang lama untuk mencapai solusi yang diinginkan. Jika nilai attractiveness besar, terdapat kemungkinan bahwa program tidak akan konvergen atau dengan kata lain tidak dapat menemukan solusi yang optimal.

Pada Tabel diatas, nilai konstanta m yang baik adalah bernilai 1 pada saat pengujian ke-10. Hal ini dikarenakan penggunaan nilai konstanta m bernilai lain akan menurunkan secara drastis nilai attractiveness yang mengakibatkan program dalam jumlah generasi/iterasi dan jumlah firefly yang sama, akan memiliki solusi yang lebih buruk daripada program dengan nilai konstanta 1.

Dari proses pengujian parameter FA diatas, didapat nilai parameter terbaik sebagai berikut:

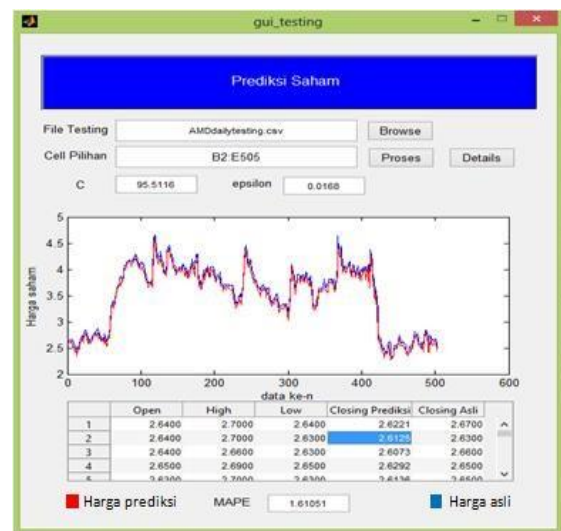
Jumlah Populasi	Jumlah Generasi	Alpha	Gamma	Konstanta m
20	100	0,1	0,5	1

Nilai parameter diatas kemudian diinputkan kedalam FA dan mendapatkan parameter SVR dengan nilai C sebesar 95,5116 dan epsilon sebesar 0,0168.

4.6 Hasil uji coba dengan SVR



Gambar 4.2 Hasil prediksi dengan data saham IBM harian



Gambar 4.5 Hasil prediksi dengan data saham AMD harian



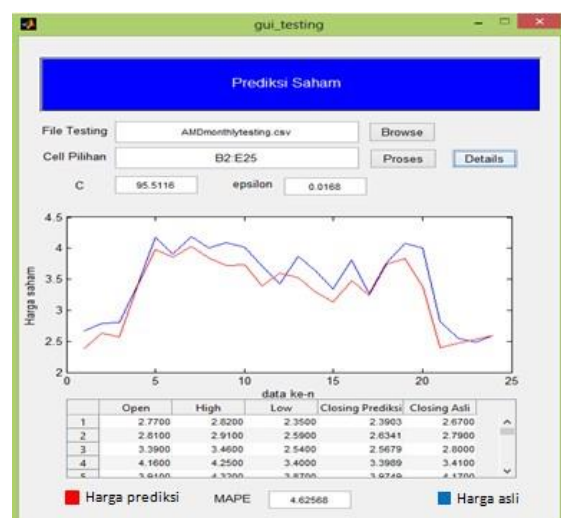
Gambar 4.3 Hasil prediksi dengan data saham IBM mingguan



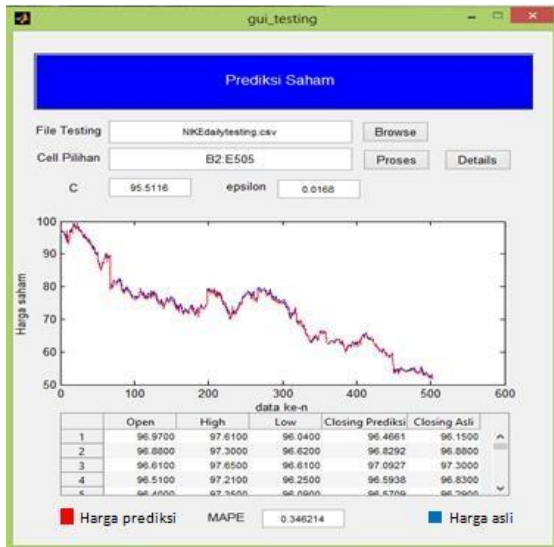
Gambar 4.6 Hasil prediksi dengan data saham AMD mingguan



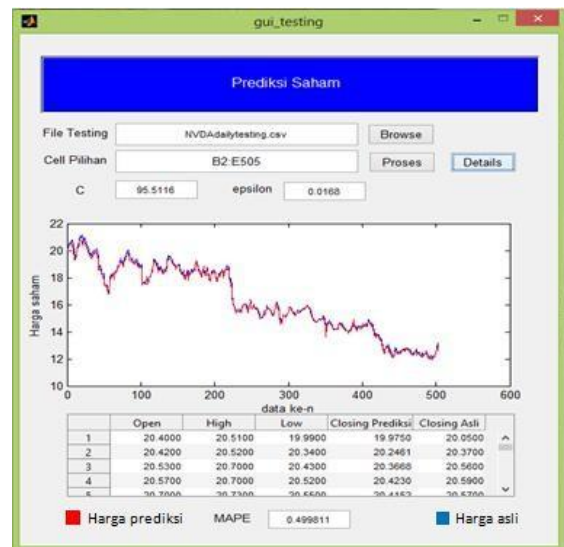
Gambar 4.4 Hasil prediksi dengan data saham IBM bulanan



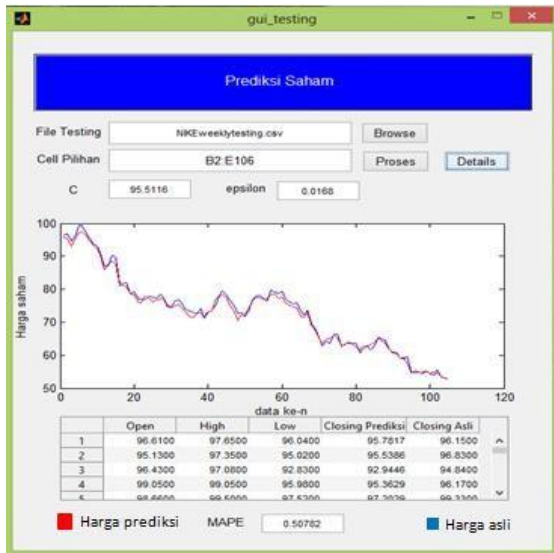
Gambar 4.7 Hasil prediksi dengan data saham AMD bulanan



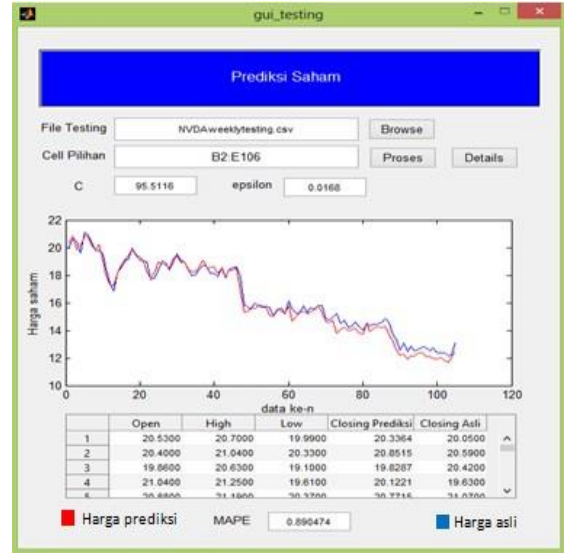
Gambar 4.8 Hasil prediksi dengan data saham NIKE harian



Gambar 4.11 Hasil prediksi dengan data saham NVIDIA harian



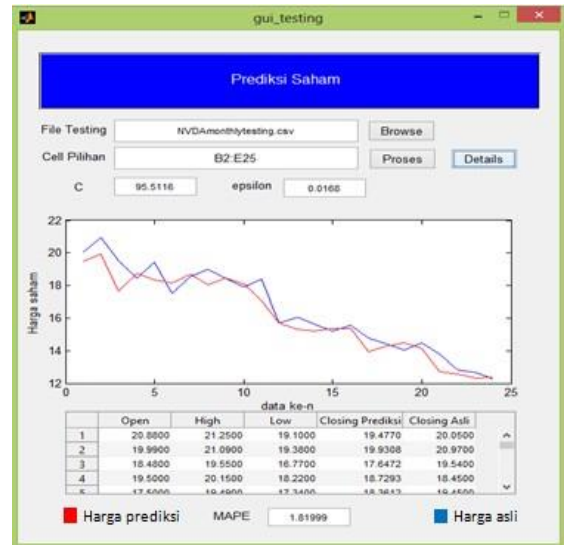
Gambar 4.9 Hasil prediksi dengan data saham NIKE mingguan



Gambar 4.12 Hasil prediksi dengan data saham NVIDIA mingguan



Gambar 4.10 Hasil prediksi dengan data saham NIKE bulanan



Gambar 4.13 Hasil prediksi dengan data saham NVIDIA bulanan

5 KESIMPULAN DAN SARAN

Setelah penelitian dan analisis dilakukan secara mendalam, dapat ditarik beberapa poin kesimpulan anatara lain :

1. Penggunaan metode *Firefly Algorithm* (FA) dan *Support Vector Regression* (SVR) pada data saham menghasilkan sistem prediksi dengan akurasi yang cukup bagus, dengan MAPE rata-rata kurang dari 5%.
2. Kombinasi terbaik yang didapat pada kasus kali ini adalah kombinasi dengan parameter jumlah generasi/iterasi yaitu 100, jumlah populasi (firefly) yaitu 20, nilai konstanta m yaitu 1, konstanta α yaitu 0.1, dan nilai konstanta γ yaitu 0.5 yang menghasilkan nilai C sebesar 95,5116 dan epsilon sebesar 0,0168.
3. Banyaknya Jumlah Gen mempengaruhi performansi FA dalam mencari solusi optimal. Semakin banyak jumlah gen maka nilai eror semakin membesar dan nilai akurasi akan menurun.

Nilai konstanta α , γ , dan konstanta m sangat menentukan pergerakan kunang-kunang menuju kunang-kunang yang terbaik

Berikut ini adalah beberapa saran yang dapat dijadikan pedoman untuk melakukan pengembangan tugas akhir ini, antara lain :

1. Jumlah dan nilai variasi kombinasi parameter hendaknya diperbanyak untuk mengeksplorasi lebih dalam lagi hasil yang didapat.
2. Mencoba mencari nilai parameter SVR dengan algoritma optimasi yang lain.
3. Mencoba menggunakan data lain seperti data curah hujan, harga listrik,dll.

6. DAFTAR PUSTAKA

- [1] A.Kazem et al, “*Support Vector Regression with chaos-based firefly algorithm for stock market price forecasting*”, *Applied Soft Computing* 13 (2013) 947-958.
- [2] Huang, Sian-Chang,2010, “Chaos-based support vector regression for exchange rate forecasting”, *Expert System with Application* 37 (2010) 8590-8598.
- [3] Kavousi-Fard, Abdollah, 2014, “*A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short Term Load Forecasting*”, *Expert System with Applications* 41 (2014) 6047-6056.
- [4] Santosa,Budi, “Tutorial Support Vector Machine”. ITS,Surabaya

- [5] Sari, Diana Puspita,2009, “Analisis performansi *Support Vector Regression* dalam memprediksi bonus tahunan karyawan”, *J@ti Undip* vol 4.
- [6] Capital Market Education, 2008, “Pengertian Data Mining”, <http://coki002.wordpress.com/pengertian-saham-dan-jenis-jenis-saham/>, diakses tanggal 23 Maret 2015.
- [7] Darmadji dan Fakhruddin,2001.” *Pasar modal di Indonesia*”. Salemba Empat, Jakarta.
- [8] Herawati, S., (2013), “Peramalan Harga Saham Menggunakan Integrasi Empirical Mode Decomposition dan Jaringan Syaraf Tiruan”,*JURNAL ILMIAH MIKROTE.*, 1, 23-28.
- [9] Digensia,2012,“Analisa Time Series”, <http://digensia.wordpress.com/2012/08/24/analisa-time-series/>, diakses tanggal 15 Januari 2015.
- [10] Xin-She Yang and Xingshi He, (2013). “Firefly Algorithm: Recent Advances and Applications”, *Int. J. Swarm Intelligence*, Vol. 1, No. 1, pp. 36–50.
- [11] C. Sudheer, S. K. Sohani, D. Kumar et al., “A support vector machine-firefly algorithm based forecasting model to determine malaria transmission,” *Neurocomputing*, vol. 129, pp. 279–288, 2014.
- [12] L.J. Cao. Support Vector Machines Experts for Time Series Forecasting. *Neurocomputing* 51 (2003) 321-339.
- [13] MO Yuan-bin et al, Optimal Choice of Parameters for Firefly Algorithm, 2013 Fourth International Conference on Digital Manufacturing & Automation.
- [14] Drucker, H., Burges, C., Smola, A., Kaufmann, L., & Vapnik, V. (1996). Support Vector Regression Machines. *Advances in Neural Information Processing Systems*, 155-161.
- [15] Furi R.P.,2014 ”Prediksi Financial Time Series Menggunakan *Independent Component Analysis* dan *Support Vector Regression*”, Bandung: Universitas Telkom.