P-ISSN: 2723-8202
E-ISSN: 2503-1171

Volume 6, No 2, Juni 2021, Page: **93-99**

JURNAL TIKA
Fakultas Ilmu Komputer Universitas Almuslim
http://www.journal.umuslim.ac.id/index.php/tika/index

doi.org/10.51179/tika.v6i02.486

MODEL FOURIER UNTUK PREDIKSI HARGA SAHAM ASTRAZENECA MENGGUNAKAN ALGORITMA LEVENBERG-MARQUARDT

Hafizh Al Kautsar Aidilof [1]
[1] Program Studi Informatika, Fakultas Teknik, Universitas Malikussaleh
e-mail: hafizh@unimal.ac.id

Abstract

[Fourier Model For Astrazeneca Stock Price Prediction Using Levenberg-Marquardt Algorithm] The soaring cases of covid-19 prompted some countries to find solutions to save their people. One of the steps that is currently being taken is with vaccines. Several leading companies in the world that produce drugs are known to have produced vaccines for covid-19, one of which is AstraZeneca. AstraZeneca vaccine is known as the most widely used vaccine in all countries in the world. Interesting thing to research is how the development of the company's stock engaged in the medical field, especially companies that produce vaccines for covid-19. This study used Fourier's approach to modeling its curve fittings. As for the prediction process using levenberg-marquardt algorithm which is known to be reliable to perform the prediction process. Levenberg-Marquardt's algorithm has the advantage of a fast training process and reliable accuracy due to its work that combines Gauss-Newton and Steepest Descent. The result is root mean square error value from the test result that was lower than the Root Mean Square Error value in the training process. This indicates that the prediction went well.

Keywords: Levenberg-Marquardt; Fourier; AstraZeneca.

Abstrak

Melonjaknya kasus covid-19 mendorong beberapa negara untuk mencari solusi guna menyelamatkan rakyatnya. Salah satu langkah yang saat ini gencar dilakukan adalah dengan vaksin. Beberapa perusahaan terkemuka di dunia yang memproduksi obat-obatan diketahui sudah memproduksi vaksin untuk covid-19, salah satunya adalah AstraZeneca. Vaksin AstraZeneca diketahui sebagai vaksin yang paling banyak digunakan di seluruh negara di dunia. Hal yang menarik untuk diteliti adalah bagaimana perkembangan saham perusahaan yang bergerak di bidang medis terutama perusahaan yang memproduksi vaksin untuk covid-19. Penelitian ini menggunakan pendekatan Fourier untuk permodelan fitting kurvanya. Sedangkan untuk melakukan proses prediksi menggunakan algoritma Levenberg-Marquardt yang dikenal handal untuk melakukan proses prediksi. Algoritma Levenberg-Marquardt memiliki kelebihan berupa proses training yang cepat dan akurasi yang handal dikarenakan cara kerjanya yang menggabungkan *Gauss-Newton* dan *Steepest Descent*. Hasil yang diperoleh adalah nilai Root Mean Square Error dari hasil pengujian yang dilakukan lebih rendah dari nilai Root Mean Square Error pada proses pelatihan. Hal ini menunjukkan bahwa prediksi berjalan dengan baik.

Kata Kunci: Levenberg-Marquardt; Fourier; AstraZeneca.

1. Pendahuluan

Investasi merupakan hal yang berkembang pesat di perekonomian masyarakat. Investasi adalah penanaman modal yang dilakukan investor dengan tujuan untuk memperoleh keuntungan. Potensi memperoleh keuntungan yang besar dalam waktu yang singkat menjadi daya tarik terbesar bagi investor.

Contoh investasi yang paling banyak diperdagangkan di pasar modal adalah saham. Saham merupakan salah satu pilihan investasi yang menarik karena dapat memperoleh untung yang besar dibandingkan dengan usaha lainnya. Dalam hal ini, penulis mengambil data saham AstraZeneca. AstraZeneca adalah perusahaan multinasioanl yang bergerak di bidang farmasi. Nama AstraZeneca itu sendiri sudah terkenal di seluruh dunia, khususnya di Indonesia sejak antusiasme masyarakat Indonesia akan hal vaksin tumbuh.

Keuntungan yang diperoleh dari investasi saham ini dapat dilihat dari nilai *return*, dimana nilai *return* dipengaruhi oleh perubahan harga saham. *Return* memiliki dua komponen yaitu *current income* dan *capital gain*. Karena pergerakan harga saham pada dasarnya tidak dapat diprediksi secara pasti, maka diperlukan model matematis tentang pergerakan harga saham tersebut. Salah satu metode yang digunakan untuk memprediksi harga saham di masa yang akan datang adalah dengan algoritma *Levenberg Marquardt*. Algorima *Levenberg Marquardt* merupakan metode optimasi nonlinier yang digunakan pada saat koreksi error *backpropagation* untuk menemukan bobot yang disesuaikan. Algoritma *Levenberg Marquardt* memiliki kelebihan mengoptimasi akurasi sehingga diusulkan digunakan untuk membantu masalah prediksi harga saham.

Data yang telah dioptimasi dengan Algoritma *Levenberg Marquardt* nantinya akan disajikan dalam bentuk grafik dengan menggunakan *Transformasi Fast Fourier*. *Transformasi Fast Fourier* adalah suatu model transformasi yang memindahkan domain spasial atau domain waktu menjadi domain frekuensi. Dengan *Transformasi Fast Fourier* yang digunakan untuk analisis *spectral*, memiliki tujuan agar sinyal dari domain waktu bisa menjadi sinyal dalam domain frekuensi. Hal ini dilakukan agar perhitungan menjadi lebih mudah, apabila masih dalam domain frekuensi maka perhitungannya akan lebih sulit

2. Metode

A. Transformasi Fast Fourier

Fast Fourier Transform (FFT) adalah teknik perhitungan operasi matematika yang digunakan untuk mentrasformasi sinyal analog menjadi sinyal digital berbasis frekuensi. *Fast Fourier Transform* (FFT) membagi sebuah sinyal menjadi frekuensi yang berbeda-beda dalam fungsi eksponensial yang kompleks. *Fast Fourier Transform* (FFT) menghitung transformasi *fourier* diskrit dengan cepat dan efisien karena sinyal-sinyal dalam sistem komunikasi bersifat kontinyu, sehingga hasilnya dapat digunakan untuk transformasi fourier. Fourier Transform dapat didefinisikan dengan rumus:

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt$$

dimana :

$S(f)$	= sinyal dalam domain frekuensi (frequency domain)
$s(t)$	= sinyal dalam domain waktu (time domain)
$s(t)e^{-j2\pi ft}$	= konstanta nilai sebuah sinyal
f	= frekuensi
t	= waktu

Dari persamaan integral di atas dapat dilihat bahwa *Fast Fourier Transform* (FFT) dapat digunakan untuk menghitung nilai frekuensi, amplitudo dan fase dari suatu gelombang sinyal. Sementara untuk menghitung spektrum frekuensi sinyal pada komputer digital membutuhkan algoritma *Discrete Fourier Transform* (DFT). *Discrete Fourier Transform* (DFT) mengubah sinyal domain waktu menjadi sinyal domain frekuensi. Berikut adalah persamaan *Discrete Fourier Transform* (DFT):

$$F(u) = \frac{1}{N} \sum_{X=0}^{X=N-1} f(x) \exp\left[-2j\pi ux / N\right]$$

$$F(u) = \frac{1}{N} \sum_{X=0}^{X=N-1} f(x) \left(\cos\left(\frac{2\pi ux}{N}\right) - j \sin\left(\frac{2\pi ux}{N}\right) \right)$$

dimana :

N = jumlah sampel yang diambil

B. Levenberg-Marquardt

Algoritma *Levenberg-Marquardt* dikembangkan oleh *Kenneth Levenberg* dan *Donald Marquardt*. Ide dasar algoritma *Levenberg-Marquardt* adalah memadukan kestabilan algoritma *Steepest Descent* dan kecepatan *Gauss Newton*. Pada dasarnya algoritma ini bekerja dengan cara melakukan pencarian nilai minimum berdasarkan jumlah kuadrat terendah. *Levenberg-Marquardt* melakukan proses training gabungan. Pada wilayah yang memiliki kelengkungan yang kompleks algoritma ini beralih ke *Steepest Descent* hingga kelengkungannya tepat untuk membuat pendekatan kuadrat dimana pendekatan tersebut menggunakan algoritma *Gauss-Newton* untuk mempercepat konvergensi.

Levenberg-Marquardt adalah pengembangan dari algoritma JST *backpropagation,* pada proses *training Levenberg-Marquardt* membutuhkan jumlah iterasi lebih sedikit dibandingkan metode *training backpropagation*. Perbedaan algoritma *Levenberg-Marquardt* dengan *backpropagation* terletak pada proses update bobot [3]. Pada II-11 algoritma *backpropagation*. Proses update bobot pada pada *Levenberg-Marquardt* berdasarkan pendekatan Matriks Hessian [4]. Algoritma ini merupakan metode tercepat untuk pelatihan feedforward neural *network* berukuran besar bahkan sampai ratusan *weight*. Metode *training Levenberg-Marquardt* membutuhkan jumlah iterasi yang lebih sedikit dibandingkan metode training *Backpropagation Algorithm* (BPA) dalam mencapai *error* minimum. Hal ini dikarenakan metode BPA memerlukan *training rate* yang kecil untuk menghindari osilasi dan sering kali terlalu lambat untuk masalah praktis. Matriks Hessian merupakan turunan kedua dari fungsi kinerja terhadap setiap komponen bobot dan bias. Penentuan matriks Hessian merupakan langkah dasar pada *Levenberg-Marquardt* untuk mencari bobot-bobot dan bias koneksi yang digunakan. Agar mempermudah proses komputasi, selama algoritma pelatihan berjalan matriks Hessian diubah dengan pendekatan secara iteratif di setiap *epoch*. Fungsi gradien berguna sebagai proses perubahannya. jika berbentuk MSE (jumlah kuadrat *error*) fungsi kinerja yang digunakan, maka persamaan estimasi Matriks Hessian dapat berupa persamaan berikut:

$$H = J^T.J$$

Sedangkan gradient dapat dihitung dengan persamaan

$$g = J^T.e$$

Selanjutnya, perhitungan perubahan bobot pada Algoritma *Levenberg-Marquardt* dapat dihitung dengan persamaan:

$$\Delta w = \left[J^T J + \mu I\right]^{-1}.J^T e$$

$$J = \left[\frac{\partial e_{pj}}{\partial w_{kj}}\right]$$

dimana:

e_{pj} = error pelatihan pada output ke-j ketika menerapkan pola *n*

w_{kj} = bobot antara hidden ke-m sampai output ke-*j*

Matriks Jacobian dapat dijabarkan dengan persamaan sebagai berikut:

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \cdots & \frac{\partial e_1}{\partial w_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_n}{\partial w_1} & \cdots & \frac{\partial e_n}{\partial w_n} \end{bmatrix} \quad e = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}$$

dimana:

∂w : update bobot LM

e : vektor pada output jaringan untuk menyatakan setiap error.

Matriks Hessian dapat dijabarkan sebagai persamaan

$$H = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1 \partial w_1} & \cdots & \frac{\partial^2 E}{\partial w_1 \partial w_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_n \partial w_1} & \cdots & \frac{\partial^2 E}{\partial w_n \partial w_n} \end{bmatrix}$$

dimana:

n = jumlah bobot

C. Pelatihan Algoritma *Levenberg Marquardt*

Berikut merupakan penjabaran algoritma pelatihan *Levenberg-Marquardt*:

1. Inisialiasasikan bobot dan bias awal dengan bilangan acak kecil, target minimal, dan maksimum *epoch*. Target dihitung menggunakan *Root Mean of Squared Error* (R-MSE).
2. Tentukan parameter yang dibutuhkan, diantaranya:
   a. Inisialisasi *epoch* =0
   b. Parameter *Levenberg-Marquardt* (μ>0). Parameter harus besar dari nol.
   c. Parameter faktor *Tau* (τ), parameter ini yang berguna untuk dibagikan atau dikalikan dengan parameter *Marquardt*.
3. Perhitungan *Feedforward* untuk setiap unit *input* $(xi, i = 1,2,3, \dots, n)$ menerima sinyal masukan dan diteruskan ke seluruh unit pada lapisan tersembunyi. Masing-masing unit lapisan tersembunyi $(zj, j = 1,2,3, \dots, n)$ menjumlahkan sinyal-sinyal *input* berbobot $\left(V_{ij}\right)$ dan bias $\left(b1_j\right)$.

$$z_{in\ j} = b1_j + \sum_{i=1}^{n} x_i v_{ij}$$

   Untuk menghitung sinyal *output* maka gunakan fungsi aktivasi.

$$z_j = f\left(z_{in\ j}\right)$$

dimana:

$b1_j$ = Sinyal *input* yang masuk ke *hidden layer* $b1_j$

x = *Input* ke i = 1, 2, 3, . . , n

V_{ij} = Bobot awal dari *input layer* ke *hidden layer*

   Selanjutnya sinyal tersebut dikirimkan ke seluruh unit pada lapisan atasnya.

4. Setiap unit lapisan *output* (Yk, k = 1, 2, 3, . . , m) menjumlahkan sinyal-sinyal *input* berbobot $\left(W_{jk}\right)$ dan bias $(b2_k)$.

$$y_{in\ k} = b2_k + \sum_{i}^{p} z_j\, w_{jk}$$

   Untuk menghitung sinyal *output* maka gunakan fungsi aktivasi.

$$y_k = f(y_{in\ k})$$

   dimana:

   $y_{in\ k}$ = sinyal *input* yang masuk ke lapisan *output*

   $b2_k$ = bobot awal dari bias ke *output layer*

   z_j = *output* dari setiap *neuron hidden layer*

   w_{jk} = bobot awal dari *hidden layer* ke *output layer*

   y_k = fungsi aktivasi pada *output layer*

   Selanjutnya sinyal tersebut dikirimkan ke seluruh unit pada lapisan atasnya.

5. Hitung *error* dan MSE.

   Rumus menghitung *error*:

$$e_r = t_r y_r$$

$$e = [t_1 - y_1, t_2 - y_2 \dots t_r - y_r]^T$$

   dimana:

   r = *input* ke -*r*

   t_r = target keluaran yang diharapkan

   e_r = kesalahan pada unit keluaran (*output*)

   y_r = keluaran *actual*

   Rumus menghitung MSE:

$$MSE = \frac{\sum_{i=1}^{n} er^2}{n}$$

6. Untuk tiap unit lapisan *output, h*itung *error neuron* (Yk, k = 1,2,3, . . . , m)

$$\delta 2_k = (t_r - y_r) f' y_{in\ k}$$

$$\varphi 2_{jk} = \varphi 2_k z_j$$

$$\beta 2_k = \delta 2_k$$

Kemudian hitung koreksi bobot untuk memperbaiki nilai w_{jk}

$$\Delta w_{jk} = \varphi 2_{jk}$$

Untuk memperbaiki nilai $\beta 2_k$ maka hitung koreksi bias

$$\Delta b2_k = \beta 2_k$$

dimana:

$\delta 2_k$ = *Error neuron* utk tiap lapisan keluaran (*output*)
$\varphi 2_{jk}$ = Menghitung bobot pada *output*
$\beta 2_k$ = Menghitung bias pada *output*
Δw_{jk} = Koreksi bobot untuk memperbaiki nilai w_{jk}
$\Delta b2_k$ = Nilai koreksi bias untuk memperbaiki $\beta 2_k$

7. Kemudian *error neuron* dihitung untuk setiap unit lapisan tersembunyi ($z_j j = 1,2,3, \ldots, p$)

$$\delta_{in\ j} = \sum_{k=1}^{m} \delta 2_k w_{jk}$$

$$\delta 1_j = \delta_{in\ j} f'(z_{in\ j}) 1 - (z_{in\ j})$$

$$\varphi 1_{ij} = \delta 1_j x_j$$

$$\beta 1_j = \delta 1_j$$

8. Bentuk matriks *Jacobian* J(x). X yaitu matriks yang berisikan nilai koreksi bobot dan bias dari seluruh jaringan.

$$J = [\varphi 1_{11} \ldots \varphi 1_{np} \beta 2_1 \beta 2_m]$$

9. Menghitung bobot baru

$$W_{baru} = W_{lama} - [J^T J + \mu I] - J^T e$$

dimana:

J = Matriks Jacobian
J^T = Transpos Matriks Jacobian
I = Matriks identitas
e = nilai error

10. Hitung MSE

Jika MSE_{baru}<=MSE_{lama} maka

   a. $\mu = \frac{\mu}{\tau}$
   b. *epoch = epoch+1*
   c. kembali ke langkah 3
   d. Jika $MSE_{baru} > MSE_{lama}$ maka
      a. $\mu = \mu . \tau$
      b. Kembali ke langkah 9

      dimana:
      μ =Parameter *Marquardt*
      τ =Parameter faktor *Tau*

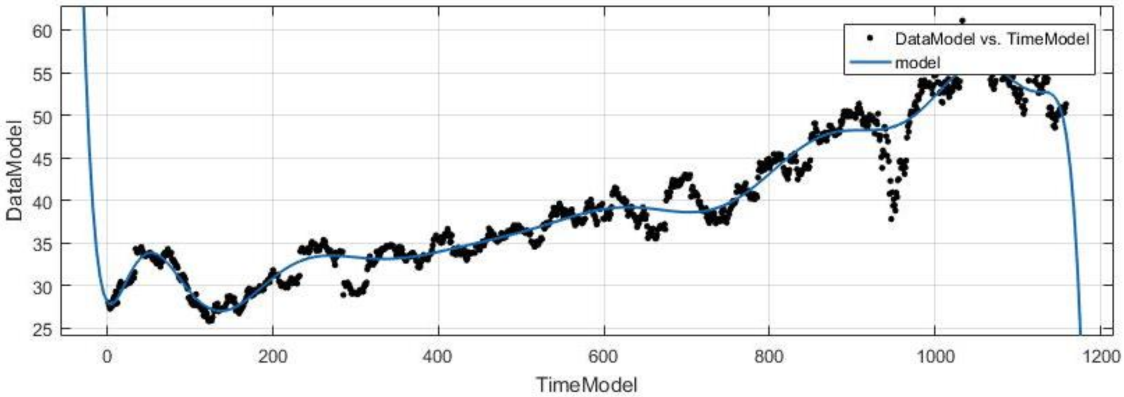
11. Proses pelatihan berhenti jika *error* = target *error* ataupun *epoch* = *epoch* maksimal

D. AstraZeneca

Sejak munculnya pandemi Covid-19 hingga antusiasme akan vaksin oleh masyarakat dunia menggema, nama AstraZeneca mulai dikenal luas di masyarakat umum. AstraZeneca adalah perusahaan multinasioanl yang bergerak di bidang farmasi. AstraZeneca berdiri pada tahun 1999 dan merpakan penggabungan dari perusahaan Astra AB dari Swedia dan Zeneca Grup dari Britania Raya. Riset dan pengembagan perusahaan ini dilakukan di Cambridge (Inggris), Gaithersburg (Amerika Serikat), dan Gothenburg (Swedia). AstraZeneca melantai di Bursa Saham London, Nasdaq, Bursa Saham Mumbai, National Stock Exchange of India dan OMX.

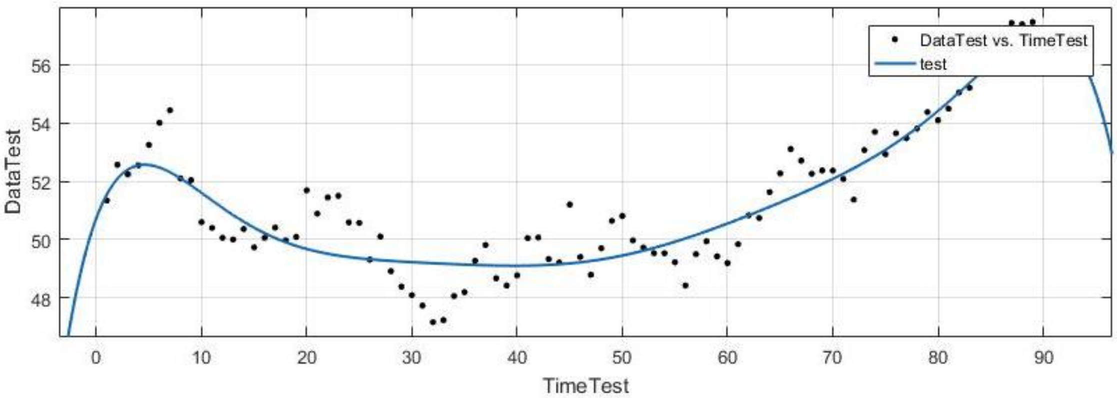
3. Hasil dan Pembahasan

Penelitian ini mengambil data saham AstraZeneca selama 5 tahun terakhir mulai dari 31 Mei 2016 sampai 31 Mei 2021. Total data yang digunakan adaalah 1520 data. Data untuk proses *training* diambil dari tanggal 31 Mei 2016 sampai 31 Desember 2020 dengan jumlah data sebanyak 1158 dan berasumsi bahwa pada akhir Desember 2020 pembicaraan tentang vaksin mulai hangat. Sedangkan untuk proses *testing* diambil dari tanggal 1 Januari 2021 sampai 31 Mei 2021 dengan jumlah data sebanyak 92. Model prediksi yang dibentuk menggunakan Fast Fourier Transfor dapat dilihat pada gambar berikut:



Gambar 1. Model Prediksi

Sedangkan untuk hasil prediksi dapat dilihat pada gambar berikut:



Gambar 2. Hasil Prediksi

Perbandingan hasil R-MSE yang didapat antara Model dan Testing sesuai periode pada Fourier dapat dilihat pada tabel berikut:

Tabel 1. Perbandingan R-MSE

Periode	R-MSE Model	R-MSE Testing
1	0,4327	0,9369
2	0,4519	0,9446
3	0,5256	0,9264
4	1,0077e+28	0,6217
5	0,6822	0,7141
6	0,7353	0,6611
7	0,7778	0,6338

8	0,8382	0,5852

4. Kesimpulan

Kesimpulan dari peelitian ini adalah proses prediksi menggunakan algoritma Levenberg-Marquardt menghasilkan nilai Root MSE yang lebih rendah daripada model dimulai dari model periode Fourier ke 6. Sedangkan nilai Root Mean Square Error terbaik yang didapat adalah 0,5852 yang didapat pada periode Fourier ke 8.

Daftar Pustaka

Kusuma, Tiara Dine. 2021. Fast Fourier Transform (FFT) Dalam Transformasi Sinyal Frekuensi Suara Sebagai Upaya Perolehan Average Energy (AE) Musik. Jurnal Pengkajian dan Penerapan Teknik Informatika Institut Teknologi PLN, 14(1), pp. 2655-5018.

Sunarauw, S. J. A. 2018. Algoritma Pelatihan Levenberg-Marquardt Backpropagation Artificial Neural Network Untuk Data Time Series. Jurnal Sains dan Teknologi Universitas Negeri Manado, 1 (2), pp. 213-221.

Hidayat, R. N., Isnanto, R. R., dan Nurhayati, O. D. (2013). Implementasi Jaringan Syaraf Tiruan Perambatan Balik untuk Memprediksi Harga Logam Mulia Emas Menggunakan Algoritma Levenberg Marquardt. Jurnal Teknologi Dan Sistem Komputer, 1(2), 49–55.

Rudyatmoko, C., dan Sugiantoro, B. (2017). Konsep Kriptografi Menggunakan Jaringan Saraf Tiruan Backpropagation dan Levenberg-Marquardt, (x), 1–7.

Lisa, Yasinta. 2015. Implementasi Algoritma Levenberg-Marquardt dan Regularisasi Bayes untuk Prediksi Curah Hujan. Vox Edukasi, 6 (2), pp. 201-210.