



Prediksi IHSG dengan Backpropagation Neural Network

Andy Santoso¹, Seng Hansun²

^{1,2}Program Studi Informatika, Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara

¹andy@student.umn.ac.id, ²hansun@umn.ac.id

Abstract

IDX Composite is a combination of all common stock and preferred stock which registered on Bursa Efek Indonesia (BEI). IDX Composite is often used by investor to predict the stock price to get profit. But, to predict the stock price is not easy, hence it yields a high risk to investor. This study offers the usage of backpropagation algorithm to minimize the risk. Backpropagation is a supervised algorithm and will be made in Python programming language, in this case, backpropagation will use and learn the past 5 days data to predict the outcome. Also, this study shows that backpropagation have a high accuracy which reflects in Mean Square Error Testing value of 320.49865083640924 to predict IDX Composite using 0.3 learning rate and 3000 epoch.

Keywords: backpropagation, IDX composite, investor, prediction, Python

Abstrak

Indeks Harga Saham Gabungan (IHSG) merupakan gabungan dari seluruh saham biasa dan saham preferen yang tercatat di Bursa Efek Indonesia (BEI). IHSG dapat dijadikan acuan oleh para investor untuk meramalkan harga saham sehingga mendapatkan keuntungan. Tetapi, untuk memprediksi harga saham kedepannya merupakan hal yang cukup sulit sehingga dapat menjadi suatu resiko bagi para investor, sehingga diperlukan suatu metode yang dapat meramalkan data IHSG untuk mengurangi resiko tersebut. Dalam penelitian ini, akan digunakan algoritma *backpropagation* yang merupakan algoritma pembelajaran terawasi yang dibangun dengan menggunakan bahasa pemrograman Python. Algoritma ini akan mempelajari data 5 (lima) hari sebelumnya dan akan digunakan untuk memprediksi harga kedepannya. Penelitian ini memiliki tingkat akurasi berupa MSE *testing* sebesar 320,49865083640924 dengan menggunakan *learning rate* 0,3 dan 3.000 *epoch*.

Kata kunci: *backpropagation*, IHSG, investor, prediksi, Python.

© 2019 Jurnal RESTI

1. Pendahuluan

Berikut Menurut Darmadji dan Fakhrudin [1], saham (*stock*) merupakan tanda penyertaan atau kepemilikan seseorang atau badan dalam suatu perusahaan atau perseroan terbatas. Banyak golongan masyarakat membeli ataupun menanamkan modalnya ke dalam pasar saham berharap untuk mendapatkan keuntungan. Hal ini senada dengan pendapat Widoatmodjo [2] yang menyatakan bahwa beberapa keuntungan membeli saham adalah *Capital Gain*, deviden, dan saham juga dapat dijaminkan ke bank sebagai agunan untuk memperoleh kredit. Untuk mendapatkan keuntungan tentunya diperlukan suatu analisis untuk memprediksi arah maupun harga saham. Secara garis besar analisis

tersebut dapat digolongkan menjadi dua, yaitu analisis fundamental dan analisis teknikal.

Analisis fundamental adalah teknik analisis yang digunakan oleh investor jangka panjang, yang didasarkan pada faktor-faktor ekonomi seperti suku bunga acuan (BI 7-Day Repo Rate), *yield* obligasi, dan faktor-faktor ekonomi lainnya. Menurut Suad [3] analisis fundamental dalam memperkirakan harga saham di masa yang akan datang dilakukan dengan menggunakan nilai faktor-faktor fundamental, meliputi juga kinerja perusahaan, misalnya Earning Per Share (EPS), Debt to Equity Ratio (DER), Return on Equity (ROE), dan lainnya. Akan tetapi, analisis fundamental tidak mampu untuk menentukan arah dari harga saham. Analisis fundamental hanya dapat memberikan

penilaian terhadap suatu saham apakah saham tersebut layak atau tidak layak untuk dijadikan tempat investasi.

Menurut Rahardjo [4], analisis teknikal adalah suatu metodologi peramalan fluktuasi harga saham yang datanya diambil dari data perdagangan saham yang terjadi di pasar saham (bursa efek). Jenis data bisa berbentuk informasi harga saham, jumlah *volume* dan nilai transaksi perdagangan, harga tertinggi dan terendah pada perdagangan setiap hari, atau berbagai informasi lain yang terkait dengan transaksi saham yang terwujud dalam bentuk tren harga saham; bisa dalam bentuk grafik atau sejenisnya. Dalam analisis teknikal digunakan berbagai macam indikator seperti Moving Average (MA), Stochastic, Fibonacci, dan indikator-indikator lainnya. Indikator tersebut didasarkan pada keyakinan bahwa sejarah akan berulang. Hal ini juga dibuktikan pada penelitian yang dilakukan oleh Abdillah [5] yang menyatakan bahwa harga saham masa lalu berpengaruh secara signifikan terhadap harga saham. Adapun kelemahan dari analisis teknikal adalah bias dari pengguna indikator tersebut, karena dalam menggunakan indikator yang terdapat pada analisis teknikal membutuhkan pemahaman dari pengguna itu sendiri.

Didasari pada kekurangan dari teknik-teknik di atas, peneliti menggunakan Jaringan Saraf Tiruan dengan algoritma pelatihan *backpropagation*. *Backpropagation* merupakan algoritma pelatihan yang terawasi (*supervised*) dan hasil dari algoritma ini mempunyai tingkat akurasi yang tinggi, seperti yang sudah dibuktikan pada penelitian Latifah [6] yang meramalkan Indeks Saham Syariah, Novita [7] yang memprediksi harga Bank BCA, dan Malyadi [8] yang memprediksi harga perusahaan Ace Hardware. Ketiga penelitian tersebut menggunakan algoritma *backpropagation* untuk memprediksi harga saham yang berbeda-beda. Selain itu *backpropagation* tidak mengharuskan penggunanya paham akan indikator-indikator maupun faktor ekonomi yang dibutuhkan dalam analisis teknikal dan analisis fundamental.

2. Metode Penelitian

Beberapa metode dan algoritma yang digunakan dalam penelitian dibahas di sini, meliputi jaringan saraf tiruan, fungsi aktivasi, *backpropagation*, *mean square error*, dan indeks harga saham gabungan.

2.1. Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (*artificial neural network*) atau disingkat JST adalah sebuah sistem yang terinspirasi dari sel saraf otak manusia [9]. Dalam sel saraf biologis *dendrit*, *soma* dan *axon* merupakan komponen penyusun yang saling bekerja sama untuk mengolah sinyal-sinyal informasi.

Jaringan Saraf Tiruan tersusun dari beberapa *neuron* yang saling berhubungan. *Neuron* tersebut akan

mentransmisikan informasi yang diterima, menuju *neuron-neuron* yang lain. *Neuron* yang terdapat pada Jaringan Saraf Tiruan akan dikumpulkan pada suatu *layer* dan *layer* tersebut akan dihubungkan dengan *layer-layer* sebelum dan sesudahnya. Umumnya, *layer* pada Jaringan Saraf Tiruan tersusun dari *Input Layer*, *Hidden Layer*, dan *Output Layer*.

2.2. Fungsi Aktivasi Sigmoid Biner

Fungsi ini sering digunakan untuk jaringan saraf yang dilatih menggunakan metode *backpropagation* [10]. *Range* nilai dari fungsi ini berkisar dari 0 sampai 1.

Fungsi sigmoid biner dapat dirumuskan:

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}} \quad (1)$$

dimana y adalah nilai fungsi sigmoid biner yang diperoleh dari suatu input nilai *neuron* x .

2.3. Backpropagation

Backpropagation adalah sebuah algoritma pembelajaran yang terawasi (*supervised learning*) dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot terhubung dengan *neuron* yang ada pada lapisan tersembunyinya (*hidden layer*). Algoritma pelatihan *backpropagation* pada dasarnya terdiri dari tiga tahapan [10], yaitu:

1. Input nilai data pelatihan sehingga diperoleh nilai *output* (tahap *Feedforward*).
2. Propagasi balik dari nilai *error* yang diperoleh (tahap *Backpropagation*).
3. Penyesuaian bobot koneksi untuk meminimalkan nilai *error*.

Ketiga tahapan tersebut diulangi terus-menerus sampai mendapatkan nilai *error* yang diinginkan. Setelah *training* selesai dilakukan, hanya tahap pertama yang diperlukan untuk memanfaatkan Jaringan Saraf Tiruan tersebut. Proses pelaksanaan dari algoritma *backpropagation* secara lebih detail adalah sebagai berikut [11, 12]:

1. Inisialisasi bobot (ambil bobot awal dengan nilai acak yang cukup kecil).
2. Kerjakan langkah-langkah berikut ini selama kondisi bernilai *false*.

Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:

Fase *Feedforward*

- a. Tiap-tiap input ($x_i, i = 1, 2, 3, \dots, n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (*hidden layer*).
- b. Tiap-tiap unit tersembunyi ($z_j, j = 1, 2, 3, \dots, p$) menjumlahkan sinyal-sinyal input terbobot.

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya.

$$z_j = f(z_{in_j}) \quad (3)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *hidden*).

- c. Tiap-tiap unit *hidden* ($y_k, k = 1, 2, 3, \dots, m$) menjumlahkan sinyal-sinyal *input* terbobot.

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (4)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya

$$y_k = f(y_{in_k}) \quad (5)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output*).

Fase Backpropagation

- a. Tiap-tiap unit *output* ($y_k, k = 1, 2, 3, \dots, m$) menerima target pola yang berhubungan dengan pola input pembelajarannya, hitung informasi *error*-nya:

$$\delta_k = (t_k - y_k) f'(y_k) \quad (6)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk}):

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (7)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai w_{0k}):

$$\Delta w_{0k} = \alpha \delta_k \quad (8)$$

Kirimkan δ_k ini ke unit-unit yang ada di lapisan bawahnya.

- b. Tiap-tiap unit tersembunyi ($z_j, j = 1, 2, 3, \dots, p$) menjumlahkan input-nya (dari unit-unit yang berada pada lapisan di atasnya):

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (9)$$

kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*:

$$\delta_j = \delta_{in_j} f'(z_j) \quad (10)$$

Kemudian hitung koreksi bobot

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (11)$$

Hitung juga koreksi bias

$$\Delta v_{0j} = \alpha \delta_j \quad (12)$$

- c. Tiap-tiap unit *output* ($y_k, k = 1, 2, 3, \dots, m$) memperbaiki bias dan bobotnya ($j = 0, 1, 2, 3, \dots, p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (13)$$

Tiap-tiap unit tersembunyi ($z_j, j = 1, 2, 3, \dots, p$) memperbaiki bias dan bobotnya ($i = 0, 1, 2, 3, \dots, n$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (14)$$

Tes kondisi berhenti ketika sudah memenuhi jumlah *epoch* atau *Mean Square Error* sudah lebih kecil daripada target.

2.4. Mean Square Error

Mean Square Error (MSE) adalah fungsi kinerja yang umumnya digunakan untuk *backpropagation* dimana fungsi ini akan mengambil rata-rata kuadrat *error* yang terjadi antara *output* jaringan dan target.

Mean Square Error dihitung sebagai berikut:

1. Hitung keluaran jaringan saraf untuk masukan pertama aktivasi prediksi
2. Hitung selisih antara nilai target dengan nilai keluaran prediksi
3. Kuadratkan setiap selisih tersebut
4. Jumlahkan semua kuadrat selisih dari tiap-tiap data pembelajaran dalam satu *epoch*.
5. Bagi hasil penjumlahan tersebut dengan jumlah data pembelajaran

Rumus dari *Mean Square Error*:

$$MSE = \frac{\sum_{i=1}^n e_i^2}{n} \quad (15)$$

dimana e_i adalah selisih nilai target dengan nilai keluaran prediksi dan n adalah jumlah data pembelajaran yang digunakan.

2.5. Indeks Harga Saham Gabungan

Indeks Harga Saham Gabungan adalah suatu alat ukur yang digunakan untuk mengukur kinerja seluruh saham yang tercatat pada suatu bursa efek dengan menggunakan semua saham yang tercatat di Bursa Efek Jakarta sebagai komponen perhitungan indeks. Indeks Harga Saham Gabungan diperkenalkan pertama kali pada tanggal 1 April 1983. Menurut Sunariyah [13] Indeks Harga Saham Gabungan menggambarkan suatu rangkaian informasi historis mengenai pergerakan harga saham gabungan seluruh saham, sampai pada tanggal tertentu.

3. Hasil dan Pembahasan

3.1. Normalisasi

Data yang akan dibaca dan diproses menggunakan fungsi aktivasi *sigmoid* biner, data akan ditransformasikan terlebih dahulu karena *range* keluaran fungsi aktivasi *sigmoid* adalah [0,1]. Tapi akan lebih baik jika ditransformasikan ke interval yang lebih kecil, misal pada interval [0.1, 0.9]. Ini

mengingat fungsi *sigmoid* merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 maupun 1 [14]. Data saham akan dinormalisasi terlebih dahulu dengan menggunakan rumus normalisasi berikut [16].

$$x' = \frac{0.8(x - \text{data minimum})}{(\text{data maksimum} - \text{data minimum})} + 0.1 \quad (16)$$

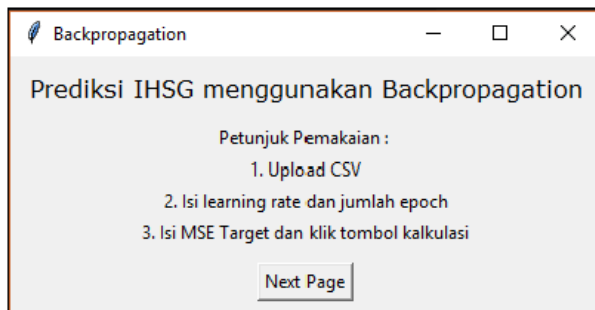
3.2. Arsitektur Jaringan Saraf Tiruan

Berikut ini adalah tahap-tahap perancangan jaringan saraf tiruan yang akan digunakan dalam aplikasi ini:

1. Jaringan saraf tiruan terdiri dari 3 (tiga) buah *layer* yaitu *Input layer*, *Hidden layer*, dan *Output layer*
2. Jumlah masukan pada *input layer* pada aplikasi agar dapat memprediksi harga saham ke depan ditetapkan sebanyak 5 (lima) hari sebelumnya
3. *Hidden layer* yang dicoba berjumlah satu *layer* dan jumlah *neuron* pada *hidden layer* adalah 4 (empat)
4. Bobot awal dan bias pada awalnya akan dipilih secara acak antara -1 sampai dengan 1
5. Jumlah keluaran yang ada pada jaringan terdapat 1 (satu) buah
6. Jumlah data total yang ada terdapat 234 buah, dimana akan dibagi menjadi dua dengan rasio 7:3, 164 data untuk *training* dan sisanya untuk *testing*

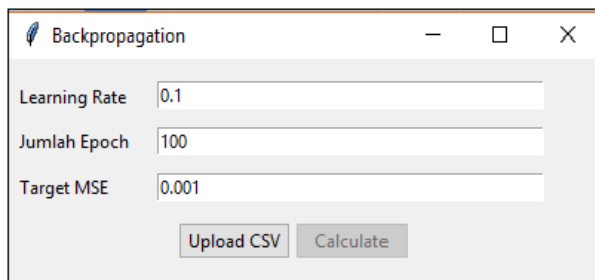
3.3. Hasil

Hasil pembangunan aplikasi dengan menggunakan bahasa Python dan *library* matplotlib, pandas, dan NumPy diperlihatkan pada gambar-gambar berikut.



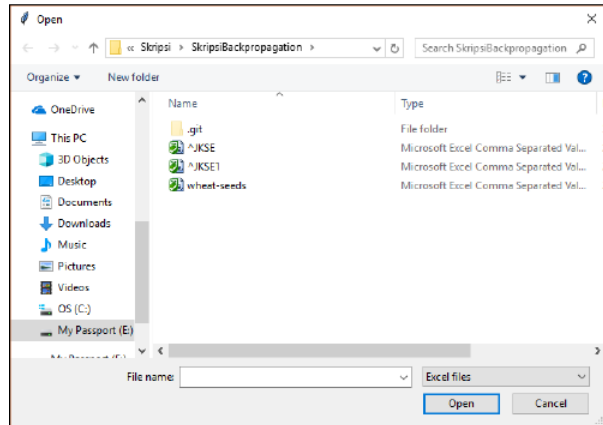
Gambar 1. Halaman Instruksi

Gambar 1 merupakan halaman yang pertama kali muncul saat aplikasi berjalan. Halaman ini berisi instruksi singkat pemakaian aplikasi.



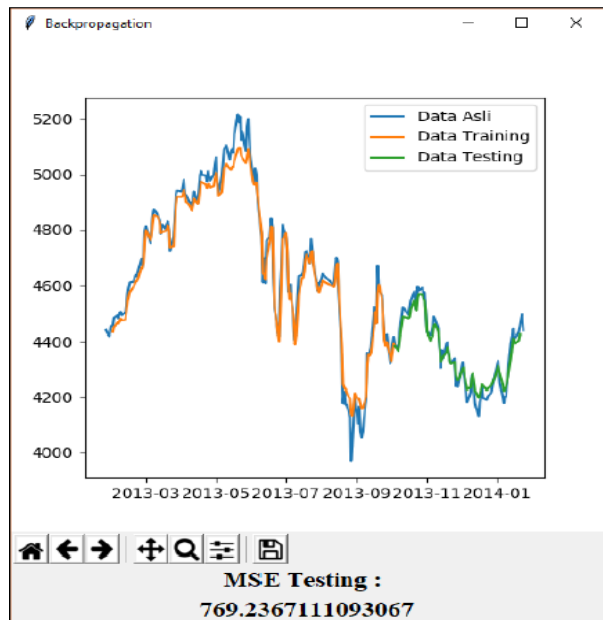
Gambar 2. Halaman Input

Gambar 2 merupakan halaman yang disajikan ketika *user* mengklik tombol *next page* dari halaman instruksi. Di halaman ini terdapat *entry* untuk mengisi *learning rate*, jumlah *epoch*, dan target *Mean Square Error* (MSE). Untuk mengisi *learning rate* mempunyai rentang nilai di atas 0 (nol) sampai dengan 1 (satu). Selain itu terdapat tombol *upload CSV* dan kalkulasi yang mana tombol kalkulasi tidak dapat ditekan sampai tombol *upload* telah dipilih.



Gambar 3. Halaman Upload CSV

Gambar 3 memperlihatkan halaman *upload CSV* dari direktori pada komputer yang ada. File yang dipilih hanya boleh berekstensi CSV dan mempunyai kolom tanggal dan *close*.



Gambar 4. Halaman Grafik

Halaman yang disajikan pada Gambar 4 merupakan halaman yang ditampilkan setelah *user* mengisi semua kolom yang terdapat di halaman *input* dan sudah melakukan *upload CSV* dengan kriteria yang

ditentukan. Grafik yang ditampilkan berupa *line graph* dan terdapat MSE *testing* di bawah gambar grafik.

3.4. Uji coba

Melalui cara dan langkah-langkah yang sudah dijelaskan di atas, algoritma *backpropagation* diterapkan pada data-data IHSg. Jumlah data yang digunakan dari tanggal 25 Januari 2013 sampai dengan 24 Januari 2014 secara harian.

Data-data tersebut akan dibagi dengan rasio 7:3, dimana 164 data pertama akan digunakan untuk *training* jaringan dan sisanya akan digunakan untuk *testing*. Data hasil kalkulasi akan ditampilkan dan dibandingkan dengan data aktual, MSE kemudian akan digunakan untuk menghitung akurasi dari keluaran jaringan.

Uji coba akan dilakukan menggunakan tiga *learning rate* yang berbeda-beda dan *epoch* yang digunakan bernilai 100 sampai dengan 3000 dengan kelipatan 100. Untuk bobot dan bias awal akan digunakan angka yang sama untuk percobaan, yakni dengan menggunakan perintah *seed* pada Python. Hasil percobaan akan ditampilkan pada tabel-tabel di bawah, dengan menampilkan jumlah *epoch*, batas kesalahan, dan MSE *testing*. MSE *Testing* adalah perhitungan MSE antara data *testing* dengan data aktual, dan data tersebut tidak dinormalisasi, sehingga angka yang dihasilkan akan cukup besar dibandingkan dengan batas kesalahan. Adapun hasil uji coba yang dilakukan dengan beberapa *learning rate* berbeda diperlihatkan pada Tabel 1, 2, dan 3.

Tabel 1. Hasil Uji Coba dengan Learning Rate 0,3

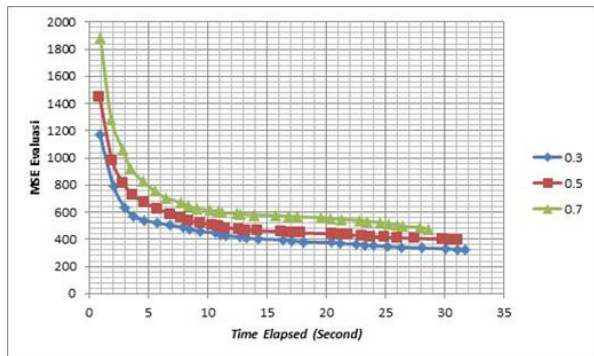
Epoch	Batas kesalahan	MSE Testing
100	0,003271442834591454	1166,8839979180486
200	0,0026669194265550055	788,0117284605611
300	0,0024045349314345647	635,250635708825
400	0,0022685588622996435	570,7205251534083
500	0,002197575514646256	540,4191730538487
600	0,0021587380388849125	520,8409155588647
700	0,0021356069744182175	503,8193403844934
800	0,0021205244380337867	487,68527127412744
900	0,0021098941270993716	472,6805943812739
...	...	...
3000	0,0020063706062967003	320,49865083640924

Tabel 2. Hasil Uji Coba dengan Learning Rate 0,5

Epoch	Batas kesalahan	MSE Testing
100	0,0029754933066231904	1447,1058682900211
200	0,0024697183669309384	981,8245066185858
300	0,0022751239265209754	813,7907560341896
400	0,0021961199027015215	729,8201986679272
500	0,0021596031232051682	670,9459527058315
600	0,0021395699524901176	624,3844478134395
700	0,002126772000455218	587,659089809317
800	0,0021174272334717774	559,1865398662194
900	0,0021097750923134836	537,1442161158928
...	...	...
3000	0,0019569183752354766	392,48181998191757

Tabel 3. Hasil Uji Coba dengan Learning Rate 0,7

Epoch	Batas kesalahan	MSE Testing
100	0,0028331755641423746	1884,9881008271586
200	0,002383707694316713	1281,384951173774
300	0,0022365641087077207	1057,8227218494478
400	0,0021829142301134368	924,2834922830403
500	0,00215885930040156	827,0556855825902
600	0,0021456462206517644	756,1458746652139
700	0,00213667152864732	705,8341268332885
800	0,002129240215274034	670,5221861328962
900	0,0021221687789155434	645,6116848053956
...	...	...
3000	0,001941792774800371	477,14234672813257



Gambar 5. Visualisasi Hasil Percobaan

Berdasarkan hasil percobaan dan visualisasi gambar di atas, *learning rate* 0,7 dapat menghasilkan *output* lebih cepat dibandingkan *learning rate* lainnya dengan total waktu 28,58247423171997 detik dengan menggunakan 3000 *epoch*. Tetapi, untuk tingkat akurasi, *learning rate* 0,3 menghasilkan hasil yang lebih baik dengan Mean Square Error evaluasi sebesar 320,49865083640924 dengan menggunakan 3000 *epoch*.

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut.

1. Metode peramalan *backpropagation* memberikan hasil peramalan yang cukup baik, hal ini dapat dilihat dengan hasil MSE *testing* sebesar 320,49865083640924 dengan *learning rate* yang digunakan adalah 0,3 dan *epoch* sejumlah 3000.
2. Dari ketiga *learning rate* yang digunakan, 0,3 merupakan yang terbaik untuk data ini, dengan nilai MSE *testing* yang paling kecil.

Melalui penelitian yang telah dilakukan, beberapa saran dapat diberikan agar penelitian yang lebih lanjut lebih baik adalah sebagai berikut:

1. Penelitian selanjutnya dapat dibuat sebagai aplikasi yang sudah terpisah (*stand-alone*) sehingga tidak diperlukan *text editor* maupun Integrated Development Environment (IDE) tertentu dalam menjalankan aplikasi.

2. Karena perkembangan teknologi yang semakin pesat dan menjamurnya *smartphone*, maka penelitian selanjutnya dapat dibuat dalam versi *mobile* untuk meningkatkan aksesibilitas.

Ucapan Terimakasih

Penulis mengucapkan terima kasih atas dukungan dan fasilitas yang diberikan oleh komunitas Lab Artificial Intelligence dan Lab Mobile Development di Universitas Multimedia Nusantara.

Daftar Rujukan

- [1] Darmadji, T. dan Fakhruddin, H.M. 2012. *Pasar modal di Indonesia: Pendekatan tanya jawab*. Jakarta: Salemba Empat.
- [2] Widoatmodjo, S. 2009. *Pasar modal Indonesia*. Ciawi, Bogor: Ghalia Indonesia.
- [3] Suad, H. 2003. *Dasar-dasar Teori Portofolio dan Analisis Sekuritas*. Yogyakarta: UPP AMP YKPN.
- [4] Raharjo, S. 2006. *Kiat Membangun Aset Kekayaan*. Jakarta: PT. Gramedia.
- [5] Abdillah, H. 2011. *Pengaruh Variabel Fundamental dan Teknikal Terhadap Harga Saham Perbankan yang Go Public di Bursa Efek Indonesia Periode 2007-2009*. Skripsi. Universitas Pembangunan Nasional Jawa Timur.
- [6] Latifah, L.N. 2016. *Jaringan Syaraf Tiruan Algoritma Backpropagation Untuk Peramalan Harga Index Saham Syariah Pada Bursa Efek Indonesia*. Skripsi. Fakultas Sains dan Teknologi. Universitas Sunan Kalijaga, Yogyakarta.
- [7] Novita, A. 2016. *Prediksi Pergerakan Harga Saham Pada Bank Terbesar di Indonesia dengan Metode Backpropagation Neural Network*. JUTISI, 5(1), pp.965-972.
- [8] Malyadi, M., Novawati, N.R., dan Purnama, R.B. 2017. *Perancangan Prediksi Untuk menentukan Indeks Harga Saham Menggunakan Jaringan Syaraf Tiruan*. Jurnal KINETIK, 2(2).
- [9] Kristanto, A. 2004. *Jaringan Syaraf Tiruan (Konsep Dasar, Algoritma dan Aplikasi)*. 1st ed. Yogyakarta: Gava Media.
- [10] Fausett, L.V. 2004. *Fundamentals of neural networks: Architectures, algorithms, and applications*. Delhi, India: Pearson Education.
- [11] Hansun, S. 2013. *Peramalan Data IHSG Menggunakan Metode Backpropagation*. ULTIMATICS, 5(1), pp.26-30.
- [12] Hansun, S. 2013. *Jakarta Stock Exchange Forecasting using Backpropagation Neural Networks*. Prosiding 2013 IEEE International Conference on Electronics Technology and Industrial Development. 23-24 Oktober 2013, Bali, Indonesia.
- [13] Sunariyah. 2006. *Pengantar pengetahuan pasar modal*. Yogyakarta: UPP Akademi Manajemen Perusahaan YKPN.
- [14] Siang, J. 2009. *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*. Yogyakarta: Penerbit Andi.