

Prediksi Harga Saham Di Indonesia Menggunakan Algoritma Long Short-Term Memory

Adhib Arfan¹ dan Lussiana ETP.²

⁽¹⁾Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok, Jawa Barat 16424

⁽²⁾STMIK Jakarta STI&K

Jl. BRI No.17, Radio Dalam Kebayoran Baru Jakarta Selatan 12140

adhib.arfan@gmail.com, lussiana@jak-stik.ac.id

ABSTRAK

Saham adalah hak yang dimiliki seseorang terhadap perusahaan dikarenakan penyerahan modal baik itu sebagian maupun keseluruhan. Para investor melakukan investasi saham dan berupaya untuk mendapatkan hasil secara maksimal, akan tetapi banyak investor masih ragu dengan risiko dalam berinvestasi. Telah banyak dikembangkan metode untuk memperkirakan harga saham yang akan datang namun masih memiliki keterbatasan diantaranya adalah ketergantungan jangka panjang. Pada tahun 1997 mulai dikembangkan model baru yang bekerja untuk menangani masalah ketergantungan jangka panjang dan dikenal dengan Long Short-Term memory (LSTM). Tujuan penelitian yang ingin dicapai adalah menghasilkan model peramalan harga saham yang lebih efektif dan memberikan hasil yang akurat. Tahapan yang dilakukan terdiri dari pengumpulan data, preprocessing data, pembagian data, perancangan LSTM, training LSTM dan melakukan pengujian. Parameter yang digunakan untuk melakukan prediksi harga saham dengan LSTM adalah data yang digunakan, hidden layer, neuron, epoch dan batch size. Pengujian tersebut mendapatkan hasil terbaik menggunakan data latih 2014-2016 dan data uji 2017-2019 yaitu dengan menggunakan parameter 2 hidden layer, 125 neuron, 10 epoch, 200 batch size dengan hasil waktu training 18.2s dengan hasil MSE sebesar 0.0015. Berdasarkan hasil pengujian, LSTM mampu memprediksi harga saham pada tahun 2017-2019 dengan performa yang baik dan tingkat kesalahan yang relatif kecil, sedangkan pengujian menggunakan metode Support Vector Regression (SVR) hanya mampu memprediksi 30 hari ke depan. Dengan demikian dapat disimpulkan bahwa LSTM mampu menanggulangi ketergantungan jangka panjang dan mampu memprediksi harga saham dengan hasil yang akurat.

Kata Kunci : prediksi, saham, jaringan syaraf tiruan, long short term memory

PENDAHULUAN

Dewasa ini dalam melakukan investasi, tersedia banyak pilihan bagi investor. Investasi bisa berbagai macam bentuknya, salah satu diantaranya adalah investasi saham. Saham adalah hak yang dimiliki seseorang terhadap perusahaan karena penyerahan modal baik itu sebagian maupun keseluruhan[1]. Para investor melakukan investasi saham dan berupaya untuk mendapatkan hasil yang diinvestasikan secara maksimal, akan tetapi banyak investor masih ragu dengan risiko dalam berinvestasi, khawatir yang diinvestasikan tidak sesuai dengan harapan [2]. Keraguan calon investor dalam berinvestasi dikarenakan fluktuasi indeks harga saham dalam waktu yang singkat. Dalam penelitian [3], tentang dampak nilai kurs dan suku bunga dijelaskan bahwa nilai kurs maupun tingkat suku bunga memiliki pengaruh negatif terhadap pasar saham.

Dalam rangka meyakinkan calon investor terhadap keuntungan yang didapat, telah banyak dikembangkan metode untuk memperkirakan harga saham yang akan datang. Metode *Back Propagation* (BP) yang digunakan dalam pemodelannya[4]. Hasilnya menyatakan bahwa hasil prediksi mendekati kenyataan, sehingga tidak jauh berbeda dengan tren indeks harga saham Bursa Efek Indonesia (BEI), namun hasil tersebut masih terdapat *error* yang besar. Selanjutnyadikembangkan model dengan menggabungkan metode BP dengan *Particle Swarm Optimization* (PSO) untuk peramalan indeks harga saham BEI[5], hasilnya dapat memperkecil waktu komputasi dan *error* dibandingkan dengan model BP saja, tetapi tidak dijelaskan periode saham yang digunakan. Kemudian penelitian tentang penggabungan *Support Vector Regression* (SVR) dan Jaringan Syaraf Tiruan (JST) untuk memprediksi indeks harga saham[6]. Hasil pengembangan ini dapat menunjukkan

performa yang cukup baik, tingkat kesalahan metode SVR-JST pada hasil MAPE dan RMSE lebih kecil dibandingkan dengan metode JST pada 3, 7, 10, 15, dan 30 hari kedepan, namun semakin besar nilai n-hari yang diprediksi menghasilkan tingkat kesalahan semakin besar.

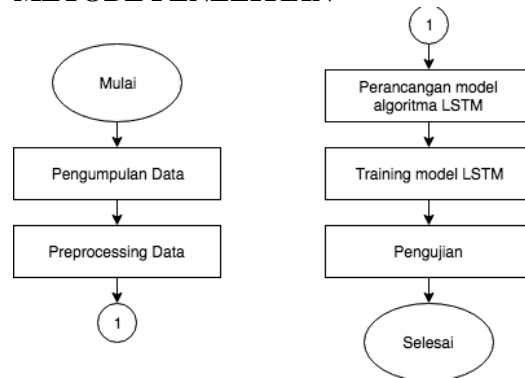
Seiring dengan pengembangan pemodelan dengan menggunakan JST, pada tahun 1997 mulai dikembangkan model baru yang bekerja untuk menangani masalah ketergantungan jangka panjang dan dikenal dengan *Long Short-Term memory (LSTM)* yang merupakan unit spesial dari *Recurrent Neural Network (RNN)*. Mengacu pada informasi bahwa LSTM dapat menanggulangi permasalahan jangka panjang[7]. Tujuan penelitian ini melakukan peramalan indeks harga saham dengan menggunakan LSTM, tujuan khusus yang ingin dicapai adalah menghasilkan model peramalan harga saham yang lebih efektif dan memberikan hasil yang akurat.

Long Sort Term Memory

Long Short Term Memory networks (LSTM) merupakan sebuah evolusi dari arsitektur RNN, dimana pertama kali diperkenalkan oleh Hochreiter & Schmidhuber (1997). Hingga penelitian ini dilakukan banyak para peneliti yang terus mengembangkan arsitektur LSTM di berbagai bidang seperti dalam bidang *speech recognition* dan *forecasting*. [8]

RNN memiliki kekurangan, kekurangan itu dapat dilihat pada inputan x_0 , Xx_1 memiliki rentang informasi yang sangat besar dengan Xx_t , x_{t+1} sehingga ketika h memerlukan informasi yang relevan dengan x_0 , x_1 . RNN tidak dapat untuk belajar menghubungkan informasi karena memori lama yang tersimpan akan semakin tidak berguna dengan seiringnya waktu berjalan karena tertimpa atau tergantikan dengan memori baru, permasalahan ini ditemukan oleh Bengio, et al. (1994). Berbeda dengan RNN, LSTM tidak memiliki kekurangan tersebut karena LSTM dapat mengatur memori pada setiap masukannya dengan menggunakan *memory cells* dan *gate units*

METODE PENELITIAN



Gambar 1. Tahapan penelitian

Gambar 1 merupakan skema tahapan penelitian yang dilakukan. Secara umum terdiri dari identifikasi masalah, pengumpulan data, *preprocessing data*, pembagian data, perancangan LSTM, *training LSTM* dan melakukan pengujian. Dalam sistem yang dibangun, dataset terdiri dari data saham dan kurs Dollar terhadap Rupiah yang diolah terlebih dahulu menggunakan normalisasi *min max scaling*. Selanjutnya dilakukan uji pada model yang telah didapatkan dari proses *training* terhadap data *testing*. Proses tersebut diulang hingga mendapatkan *output* dengan akurasi yang dapat diterima.

Pengumpulan Data

Penelitian ini menggunakan dua data sekunder yaitu data indeks harga saham dan informasi kurs USD terhadap IDR. Data indeks harga saham yang digunakan adalah data yang diperoleh dari situs *Yahoo ! Finance*, sedangkan data informasi kurs yang digunakan adalah data yang diperoleh dari OFX (<https://www.ofx.com>).

Preprocessing Data

Data yang diperoleh pada tahapan pengumpulan data masih berupa data mentah, artinya data tersebut belum bisa diolah karena ada beberapa data yang tidak menampilkan tanggal tertentu. sehingga diperlukan tahapan *preprocessing data*. *Preprocessing data* ini meliputi:

1. Join data

Proses *join data* merupakan proses penggabungan antara harga saham dengan *currency*

2. *Fill null value*
 Setelah proses *join data*, dilanjutkan dengan proses *fill null value*, yaitu dengan menggunakan *forward fill*, *forward fill* merupakan proses yang dilakukan dengan mengganti nilai *null* dengan nilai pada tanggal sebelumnya
3. *Split data*
Split data atau pembagian data digunakan untuk membedakan antara data yang digunakan untuk proses pelatihan dan data yang digunakan untuk proses pengujian. Tahapan pembagian data dibagi menjadi 2 yaitu data latih dan data uji.
4. *Normalize data*
 Untuk meminimalkan *error*, dilakukan normalisasi pada dataset dengan mengubah data aktual menjadi nilai dengan *rangeinterval* [0,1]. Teknik normalisasi yang digunakan menggunakan *min-max scaling*. Adapun untuk rumus normalisasi *min-max scaling* adalah :

$$x' = \frac{(x - \min_x)}{(\max_x - \min_x)} \quad (1)$$

keterangan :

- x : data yang dinormalisasi
- x' : data setelah dinormalisasi
- \min : nilai minimum dari keseluruhan data
- \max : nilai maksimum dari keseluruhan data

5. Pembuatan dataset
 Dataset merupakan objek yang merepresentasikan data dan relasinya dengan mengelompokan *input* dan target *output*. Dataset adalah data yang digunakan/diolah. Langkah dalam membuat dataset dijelaskan dengan *pseudo coded* dibawah ini:
 Input: data, look_back
 1. $i <- 0$
 2. $data_x <- array$
 3. $data_y <- array$
 4. for $i \rightarrow (\text{len}(\text{data}) - \text{look_back} - 1)$
 do
 5. $a <- \text{data}[i \text{ to } (i + \text{look_back})]$
 6. $data_x <- \text{append}(a)$

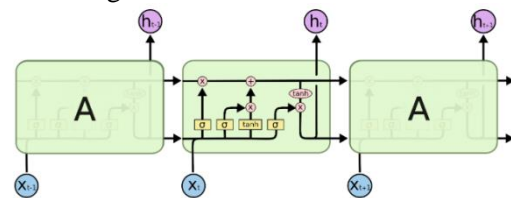
```
7. data_y <- append(data[i +
    look_back, 0])
```

Penjelasan:

nilai *look_back* = nilai rentang data. Misalnya nilai dari *look_back* adalah 1, artinya data 1 hari yang digunakan untuk prediksi hari berikutnya.

1. Lakukan perulangan untuk i sampai nilai $(\text{len}(\text{data}) - \text{look_back} - 1)$
2. a = nilai data pada index ke i sampai $i + \text{look_back}$
3. simpan nilai a ke array $data_x$
4. simpan nilai data pada index $[i + \text{look_back}, 0]$.

Perancangan model LSTM



Gambar 2. Alur kerja LSTM

Pada gambar 2 menjelaskan bagaimana alur kerja *memory cells* pada setiap *neurons* LSTM bekerja. Terdapat empat proses fungsi aktivasi pada setiap masukan pada *neurons* yang selanjutnya disebut sebagai *gates units*. *Gates units* tersebut ialah *forget gates*, *input gates*, *cell gates*, dan *output gates*.

Pada *forget gates* informasi pada setiap data masukan diolah dan dipilih data mana saja yang akan disimpan atau dibuang pada *memory cells*. Fungsi aktivasi yang digunakan pada *forget gates* ini adalah fungsi aktivasi sigmoid. Dimana hasil keluarannya antara 0 dan 1. Jika keluarannya adalah 1 maka semua data akan disimpan dan sebaliknya jika keluarannya 0 maka semua data akan dibuang. Dengan rumus sebagai berikut[9]:

$$f_t = \sigma (w_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

Pada *input gates* terdapat dua *gates* yang dilaksanakan, pertama diputuskan nilai mana yang diperbarui

menggunakan fungsi aktivasi *sigmoid*. Selanjutnya fungsi aktivasi *tanh* membuat *vector* nilai baru yang disimpan pada *memory cell*, dari kedua bagian tersebut digabungkan untuk membuat pembaharuan *memory cell*. Dengan rumus sebagai berikut[9]:

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$$

$$C'_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

Pada *cell gates* mengganti nilai pada *memory cell* sebelumnya dengan nilai *memory cell* yang baru. Dimana nilai ini didapatkan dari menggabungkan nilai yang terdapat pada *forget gate* dan *input gate*. Dengan rumus sebagai berikut[9] :

$$C_t = f_t * C_{t-1} + i_t * C'_t \quad (4)$$

Pada *output gates* terdapat dua *gates* yang dilaksanakan, pertama diputuskan nilai pada bagian *memory cell* mana yang dikeluarkan dengan menggunakan fungsi *aktivasi sigmoid*. Selanjutnya ditempatkan nilai pada *memory cell* dengan menggunakan fungsi aktivasi *tanh*. Terakhir kedua *gates* tersebut di dikalikan sehingga menghasilkan nilai yang dikeluarkan. Dengan rumus sebagai berikut [9]:

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t) \quad (5)$$

Training model LSTM

Beberapa tahapan dalam proses *training* model LSTM adalah :

1. Inisialisasi bobot awal dengan menggunakan *Xavier/Glorot Initialization*
2. *Input data training*
3. Perhitungan LSTM pada setiap *input* yaitu dimulai dengan *forget gates*, fungsi *inputgates*, fungsi *cell gates* dan yang terakhir fungsi *output gates*.
4. Perhitungan *mean squared error* (MSE) untuk mendapatkan nilai selisih antara nilai LSTM dengan target *output*. Berikut ini adalah rumus dari MSE.

$$MSE = \frac{1}{n} \sum_{i=1}^n (f_i - y_i)^2 \quad (6)$$

5. Perhitungan *gradient* untuk menentukan nilai bobot supaya hasil *loss* mendekati 0 dengan menggunakan *Backpropagation Through Time*(BTTP)
6. Setelah mendapatkan nilai *gradient*, maka dilanjutkan dengan optimasi RMSProp dan *update* bobot.
7. Kembali ke langkah dua sebanyak *epoch* yang telah ditentukan.

Pengujian

Pengujian ini dengan mengambil data testing kemudian dibandingkan dengan data yang dihasilkan dengan metode LSTM pada rentang waktu yang ditentukan dengan metode akurasi yang digunakan menggunakan MSE.

HASIL DAN PEMBAHASAN

Dataset

Berikut ini adalah komposisi data yang digunakan:

Tabel 1. Komposisi data yang digunakan

No	Data Latih	Data Uji
1	09 Jun 2014 - 30 Desember 2016	02 Januari 2017 - 06 Juni 2019

Skenario Pengujian

Pengujian yang dilakukan adalah dengan menggunakan parameter sebagai berikut :

1. Data yang digunakan adalah data pada tabel 1.
2. Jumlah *hidden layer* : 2, 3, 4, 5, 6
3. Jumlah *Neuron* pada *hidden layer* : 20, 30, 40, 50, 60
4. *epoch* = 10, 20, 30, 40, 50
5. *batch size* = 50, 100, 150, 200, 250

Untuk setiap parameter dikombinasikan dan dilakukan observasi sebanyak 5 kali. Hal ini dilakukan karena bobot awal dihasilkan secara *random*, sehingga tidak cukup dilakukan observasi hanya sekali saja, karena hasil yang

didapatkan bisa saja kebetulan baik ataupun buruk.

Menganalisis parameter dengan jumlah *hidden layer*

Parameter jumlah *hidden layer* yang digunakan adalah 2, 3, 4, 5, dan 6.

Tabel 2. Kombinasi parameter *hidden layer*

No	<i>hidden layer</i>	<i>neuron</i>	<i>epoch</i>	<i>batch size</i>	waktu training	<i>Loss</i>
1.	2	50	10	200	12.4s	0.0106
2.	3	50	10	200	14.4s	0.0104
3.	4	50	10	200	15.6s	0.0127
4.	5	50	10	200	21.9s	0.0062
5.	6	50	10	200	23.4s	0.0053

Berdasarkan lampiran Tabel 2, menunjukkan bahwa *hidden layer* 6 memiliki nilai *loss* yang paling kecil namun memiliki waktu *training* yang paling tinggi. Begitu juga dengan *hidden layer* 2 yang memiliki nilai *loss* terbesar dengan waktu *training* yang paling kecil. Semakin banyak *hidden layer* yang digunakan maka semakin lama waktu *training* yang dibutuhkan.

Menganalisis parameter dengan jumlah *neuron*

Parameter jumlah *neuron* yang digunakan adalah 25, 50, 75, 100 dan 125 *neuron*

Tabel 3. Kombinasi parameter *neuron*

No	<i>Hidden layer</i>	<i>Neuron</i>	<i>epoch</i>	<i>Batch size</i>	<i>time</i>	<i>Loss</i>
1.	2	25	10	200	14.5s	0.0298
2.	2	50	10	200	16.5s	0.0125
3.	2	75	10	200	15.4s	0.0062
4.	2	100	10	200	16.9s	0.0046
5.	2	125	10	200	18.2s	0.0037

Berdasarkan lampiran Tabel 3, menunjukkan bahwa jumlah *neuron* 125 mempunyai waktu *training* yang paling lama namun memiliki nilai *loss* yang paling kecil. Sedangkan *neuron* 25 memiliki waktu *training* paling cepat namun memiliki nilai *loss* yang paling tinggi. Nilai *neuron* sebanding dengan nilai *hidden layer*, semakin besar nilai yang digunakan maka semakin lama waktu *training* yang dibutuhkan.

Menganalisis parameter dengan jumlah *epoch*

Parameter *epoch* yang digunakan adalah 10, 20, 35, 40 dan 50 *epoch*

Tabel 4. Kombinasi parameter *epoch*

No	<i>Hidden layer</i>	<i>Neuron</i>	<i>epoch</i>	<i>Batch size</i>	<i>time</i>	<i>Loss</i>
1.	2	125	10	200	18.2s	0.0037
2.	2	125	20	200	21.6s	0.0042
3.	2	125	30	200	24.1s	0.0027
4.	2	125	40	200	26.2s	0.0027
5.	2	125	50	200	28.1s	0.0026

Berdasarkan Tabel 4, menunjukkan bahwa *epoch* 50 memiliki nilai *loss* yang paling kecil namun waktu *training* yang digunakan sangat tinggi, begitu juga dengan *epoch* 10 yang memiliki waktu *training* rendah namun memiliki nilai *loss* yang paling tinggi. *Epoch* tidak boleh terlalu besar dan terlalu kecil untuk menemukan solusi optimum, tidak ada aturan yang pasti mengenai besarnya *epoch* yang paling optimal sehingga nilai *epoch* yang didapatkan dengan melakukan percobaan.

Menganalisis parameter dengan jumlah *batch size*

Parameter *batch size* yang digunakan adalah 50, 100, 150, 200, 250

Tabel 5. Komposisi *batch size* yang digunakan

No	<i>Hidden layer</i>	<i>Neuron</i>	<i>epoch</i>	<i>Batch size</i>	<i>time</i>	<i>Loss</i>
1.	2	125	10	50	25.5s	0.0031
2.	2	125	10	100	24.2	0.0032
3.	2	125	10	150	24.1s	0.0035
4.	2	125	10	200	23.8	0.0037
5.	2	125	10	250	23.5	0.0038

Berdasarkan tabel 5, menunjukkan bahwa *batch size* 250 memiliki waktu *training* paling rendah, namun nilai *loss* yang dihasilkan paling tinggi. Semakin besar nilai *batch size* yang digunakan maka waktu *training* semakin rendah.

Prediksi harga saham dengan menggunakan algoritma LSTM dengan menggunakan data latih 2014-2016 dan data uji 2017-2019 dengan pengujian parameter menggunakan *hidden layer, neuron* pada *hidden layer, epoch* dan *batch size*. Hasil yang terbaik dari peramalan harga saham tidak tergantung dari banyaknya nilai pada setiap parameter, karena setiap parameter memiliki ketergantungan satu sama lain. Pengujian tersebut mendapatkan hasil terbaik yaitu dengan menggunakan parameter 2 *hidden layer*, 125 *neuron*, 10 *epoch*, 200 *batch size* dengan hasil waktu

training 18.2s dengan hasil MSE sebesar 0.0015. Berikut ini adalah perbandingan grafik data asli dengan grafik data prediksi.



Gambar 3. Hasil prediksi harga saham dengan LSTM

Pada gambar 3 terdapat dua warna grafik yaitu grafik warna biru dan jingga. Grafik warna biru adalah grafik data asli sedangkan grafik warna jingga adalah grafik data prediksi. Kedua grafik tersebut memiliki pola yang sama, dengan hasil MSE sebesar 0.0015 terhadap data asli.

PENUTUP

Berdasarkan hasil pengujian, LSTM mampu memprediksi harga saham pada tahun 2017-2019 dengan performa yang baik, hal ini ditunjukkan dengan tingkat kesalahan yang kecil. Dengan demikian dapat disimpulkan bahwa LSTM mampu menanggulangi ketergantungan jangka panjang dan mampu memprediksi harga saham dengan hasil yang akurat.

DAFTAR PUSTAKA

- [1]. Trippi, R. R. and Desieno, "Trading Equity Indeks Futures With A Neural Networks". *The Journal of Portfolio Management*. 1992
- [2]. Downes, J., Goodman, J. E., and Budhidarmo, S. "Kamus Istilah Keuangan dan Investasi", Elex Media Komputindo, 1999.
- [3]. Abd. Majid, M. S. "Dampak Nilai Kurs dan Suku Bunga Terhadap Harga Saham di Indonesia", *Jurnal Ekonomi dan Bisnis*, 8:51–57, 2014.
- [4]. Bachtiar Purnama, R. Perancangan prediksi untuk menentukan indeks harga saham menggunakan jaringan syaraf tiruan. *Kinetik*, 2:125, 2017.
- [5]. Wartati, D. and Aini Masruroh, N. "Aplikasi Jaringan Saraf Tiruan dan Particle Swarm Optimization Untuk Peramalan Indeks Harga Saham Bursa Efek Indonesia", *Jurnal Teknosains*, 6:22, 2017
- [6]. Evalina Siahaan, L. "Prediksi indeks harga saham dengan metode gabungan support vector regression dan jaringan syaraf tiruan", *Indonesian Journal on Computing (Indo-JC)*, 2:21, 2017
- [7]. Hochreiter, S. and Schmidhuber, J. "Long Short-Term Memory". *Neural Comput.*, 9(8):1735–1780. 1997
- [8]. Character Recognition Dengan Menggunakan Jaringan Syaraf Tiruan. *Jurnal TIMES*, Vol. III, No. 2, Pp. 1-5.
- [9]. Roondiwala, M., Patel, H., and Varma, S. "Predicting stock prices using lstm". *International Journal of Science and Research (IJSR)*, 6, 2017