

# Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin

Muhammad Wildan Putra Aldi<sup>1</sup>, Jondri<sup>2</sup>, Annisa Aditsania<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>wildanputra@students.telkomuniversity.ac.id, <sup>2</sup>jondri@telkomuniversity.ac.id,

<sup>3</sup>aaditsania@telkomuniversity.ac.id

## Abstrak

Bitcoin adalah salah satu *cryptocurrency* yang sedang diminati untuk menjadi media investasi dalam meraih keuntungan secara finansial. Meskipun sedang digemari, investasi menggunakan bitcoin masih memiliki resiko yang cukup besar. Agar dapat mengantisipasi resiko yang didapat dalam berinvestasi menggunakan Bitcoin, diperlukan suatu sistem prediksi yang dapat memprediksi pergerakan kurs nilai tukar Bitcoin. Untuk memprediksi harga Bitcoin maka data historis harga Bitcoin akan dipelajari hingga mengenali pola-pola tertentu.

Jaringan syaraf tiruan merupakan salah satu metode yang memiliki kemampuan untuk mempelajari pola-pola dari suatu data. Sistem yang dibangun pada penelitian ini adalah menggunakan metode jaringan syaraf tiruan yaitu dengan menggunakan arsitektur *Long Short Term Memory Neural Networks*. Namun teknik ini memerlukan parameter yang tepat untuk mendapatkan hasil prediksi yang akurat. Dalam tugas akhir ini menganalisis beberapa parameter seperti jumlah pola time series, jumlah neuron hidden, max epoch, dan komposisi data latih dan uji terhadap akurasi prediksi yang didapatkan.

Hasil analisis menunjukkan bahwa sistem yang dibangun mampu memprediksi harga Bitcoin dengan baik, dengan rata-rata tingkat akurasi sebesar 93.5% terhadap data testing.

**Kata kunci :** bitcoin, prediksi, jaringan syaraf tiruan, long short term memory

## Abstract

Bitcoin is one of the cryptocurrencies that are in demand to become a medium of investment in achieving financial benefits. While it is popular, investments using bitcoin still have considerable risk. In order to anticipate the risks involved in investing using Bitcoin, a prediction system is needed that can predict the movement of the Bitcoin exchange rate. To predict the price of Bitcoin, Bitcoin price historical data will be studied to recognize certain patterns.

Artificial neural networks are one method that has the ability to study patterns of data. The system built on this research is using artificial neural network method by using Long Short Term Memory Neural Networks architecture. But this technique needs the right parameters to get accurate prediction results. In this final project analyze the number of neurons in the input and hidden layer to the prediction accuracy obtained. The results of the analysis show that the built system is able to predict the Bitcoin price well, with an accuracy of 95.12% to the data testing.

**Keywords:** bitcoin, prediction, artificial neural network, long short term memory

## 1. Pendahuluan

### Latar Belakang

Bitcoin merupakan salah satu *cryptocurrency* yang menggunakan *peer-to-peer* untuk proses transaksi [1]. Banyak orang yang melakukan perdagangan jual beli Bitcoin sebagai salah satu investasi. Akibatnya nilai tukar Bitcoin terhadap mata uang yang lain seperti Dolar Amerika semakin meningkat. Salah satu strategi melakukan investasi adalah memprediksi harga di masa yang akan datang. Untuk memprediksi harga Bitcoin maka data historis harga Bitcoin akan dipelajari hingga mengenali pola-pola tertentu.

Jaringan syaraf tiruan (JST) merupakan salah satu metode yang memiliki kemampuan untuk mempelajari pola-pola dari suatu data [4]. Terdapat beberapa struktur JST yaitu diantaranya *feedforward* dan *recurrent*. Dalam tugas akhir ini, struktur JST yang akan digunakan merupakan *recurrent neural network* (RNN). RNN cocok digunakan untuk mempelajari pola-pola dalam suatu data karena arsitektur LSTM menyimpan beberapa informasi mengenai pola data dalam arsitektur jaringannya [9].

*Long Short Term Memory Neural Network* (LSTM) merupakan salah satu jenis RNN. LSTM menyimpan informasi terhadap pola-pola pada data. LSTM dapat mempelajari data mana saja yang akan disimpan dan data mana saja yang akan dibuang, karena pada setiap neuron LSTM memiliki beberapa *gates* yang mengatur memori pada setiap neuron itu sendiri. LSTM banyak digunakan untuk pemrosesan teks, video, dan data *time series*.

## Perumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, didapatkan rumusan masalah yang diangkat pada tugas akhir ini, yaitu bagaimana mengimplementasikan *Long Short Term Memory* untuk prediksi harga Bitcoin dan bagaimana performansi arsitektur *Long Short Term memory* tersebut dalam memprediksi harga bitcoin.

## Tujuan

Tujuan dari Tugas Akhir ini adalah mencari kombinasi parameter terbaik, meliputi komposisi data latih, jumlah pola *time series*, jumlah neuron hidden, dan max epoch serta menganalisis performansi prediksi pada model jaringan syaraf tiruan dengan arsitektur *Long Short Term Memory*.

## Organisasi Tulisan

Tugas Akhir ini disusun dengan struktur sebagai berikut. Setelah dijelaskan pendahuluan pada bagian pertama, dijelaskan studi terkait pada bagian kedua. Selanjutnya, dijelaskan pemodelan sistem pada bagian ketiga. Setelah itu, evaluasi performansi sistem terhadap sistem yang dibangun pada bagian ketiga. Bagian terakhir, dijelaskan kesimpulan dan saran untuk penelitian selanjutnya pada bagian keempat.

## 2. Studi Terkait

### 2.1 Penelitian Terkait

Terdapat beberapa karya tulis yang terkait dengan tulisan ini seperti "Analisis dan Implementasi Jordan Recurrent Neural Network untuk Prediksi Data Time Series Study Kasus Harga Minyak" [6], "Analisis dan Implementasi Elman Recurrent Neural Network dan Tabu Search Pada Prediksi Harga Perak" [7], dan "*Application of the Artificial Neural Network in predicting the direction of stock market index*" [5]. Karya tulis tersebut membahas tentang prediksi data time series dengan metode jaringan syaraf tiruan.

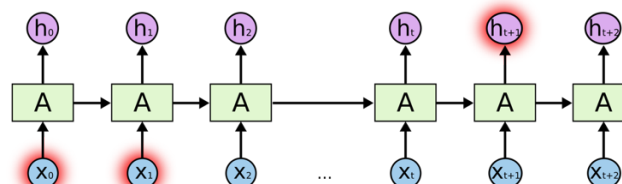
Pada karya yang berjudul "Analisis dan Implementasi Jordan Recurrent Neural Network untuk Prediksi Data Time Series Study Kasus Harga Minyak" [6], penelitian tersebut membahas tentang metode Jordan Recurrent Neural Network dalam memprediksi data time series pada kasus harga minyak. Dalam karya tulis tersebut, dikatakan bahwa dengan menggunakan data historis minyak mentah dari Januari 1986 sampai dengan Desember 2011 didapatkan rata-rata akurasi prediksi terbaik sebesar 91.87% untuk data testing.

Pada karya yang berjudul "Analisis dan Implementasi Elman Recurrent Neural Network dan Tabu Search Pada Prediksi Harga Perak" [7], penelitian tersebut membahas tentang metode Elman Recurrent Neural Network dan Tabu Search dalam memprediksi data time series pada kasus harga perak. Dalam karya tulis tersebut, dikatakan bahwa dengan menggunakan data historis harga perak dari tanggal 2 Januari 2008 sampai 14 Mei 2010 didapatkan rata-rata akurasi prediksi terbaik sebesar 97.19% untuk data testing.

Pada karya yang berjudul "*Application of the Artificial Neural Network in predicting the direction of stock market index*" [5], penelitian tersebut membahas tentang metode Artificial Neural Network dengan optimasi Genetic Algorithms. Dalam karya tulis tersebut, dikatakan bahwa dengan menggunakan teknikal analisis sebagai input pada sistem jaringan didapatkan rata-rata akurasi prediksi terbaik sebesar 86.39%

### 2.2 Long Short 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 [8] dan forecasting [2].



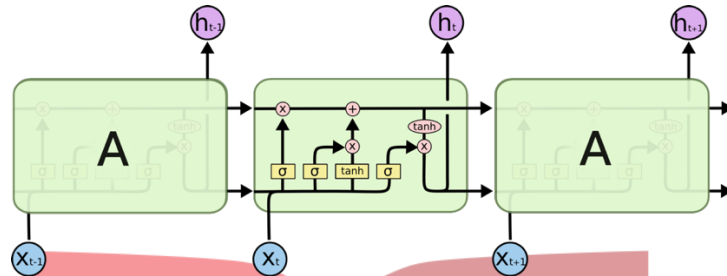
Gambar 0.1 Memori pada RNN [11]

Pada gambar 2.1 menjelaskan RNN memiliki kekurangan, kekurangan itu dapat dilihat pada inputan  $X_0$ ,  $X_1$  memiliki rentang informasi yang sangat besar dengan  $X_t$ ,  $X_{t+1}$  sehingga ketika  $h_{t+1}$  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*

### 2.2.1 Memory cells dan gate units

Seperti yang telah dijelaskan pada sub bab sebelumnya, dimana LSTM memiliki *memory cell* dan *gate units* pada setiap *neurons* nya yang berfungsi untuk mengatur memori dalam setiap *neurons* .



Gambar 0.2 flowchart memory cells LSTM [11]

Pada gambar 2.2 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 akan 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 :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

Pada *input gates* terdapat dua gates yang akan dilaksanakan, pertama akan diputuskan nilai mana yang akan diperbarui menggunakan fungsi aktivasi sigmoid. Selanjutnya fungsi aktivasi tanh akan membuat vector nilai baru yang akan disimpan pada *memory cell*. Dengan rumus sebagai berikut :

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$\check{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.3)$$

Pada *cell gates* akan 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 :

$$c_t = f_t * c_{t-1} + i_t * \check{c}_t \quad (2.4)$$

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

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.5)$$

$$h_t = o_t \tanh(c_t) \quad (2.6)$$

### 2.3 Adaptive Moment Estimation optimization (Adam)

Algoritma optimasi Adaptive moment estimation (Adam) adalah perluasan untuk Stochastic gradient descent yang baru-baru ini telah digunakan sebagai pembelajaran yang mendalam dalam *computer vision* dan *natural language processing*. Pertama kali diperkenalkan oleh Diederik Kingma dari OpenAI dan Jimmy Ba dari Universitas Toronto (2015) [10].

Adam merupakan algoritma optimasi yang mengembangkan dengan memanfaatkan kelebihan dari algoritma Adaptive Gradient (AdaGrad) dan Root Mean Square Propagation (RMSProp).

Alih-alih mengadaptasi tingkat pembelajaran parameter berdasarkan rata-rata pertama (mean) seperti dalam RMSProp, Adam juga menggunakan rata-rata kedua dari gradien (varians uncentered). Algoritma

menghitung rata-rata pergerakan eksponensial dari gradien dan gradien kuadratnya, dan parameter beta 1 dan beta 2 mengontrol tingkat peluruhan rata-rata pergerakan . [10]

Langkah algoritma optimasi Adam dapat dijelaskan dengan *pseudo code* dibawah ini :

```

 $\alpha$  : learning rate
 $\beta_1, \beta_2 \in [0,1)$  : Exponential decay rates for the moment estimates
 $f(\theta)$  : fungsi stokastik dengan parameter  $\theta$ 
 $\theta_0$  : Inisial parameter vector
 $m_0 \leftarrow 0$  (Inisialisasi moment vector pertama)
 $v_0 \leftarrow 0$  (Inisialisasi moment vector kedua)
 $t \leftarrow 0$  (inisialisasi timestep)
while  $\theta_t$  not converged do
     $t \leftarrow t + 1$ 
     $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$  (Mendapatkan gradient dari fungsi stokastik pada timestep t)
     $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$  (Update bias estimasi moment pertama)
     $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$  (Update bias estimasi moment kedua)
     $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$  (Compute bias-corrected first moment estimate)
     $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$  (Compute bias-corrected second raw moment estimate)
     $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$  (Update parameters)
End while
Return  $\theta_t$  (Hasil parameter yang didapatkan)
  
```

## 2.4 Mean Square Error (MSE)

Performansi sistem dilakukan dengan menggunakan perhitungan Mean Square Error (MSE). Model terbaik adalah model dengan nilai MSE terkecil. Rumus MSE dapat ditulis seperti dibawah ini:

$$MSE = \frac{1}{N} \sum_{i=1}^N (f_i - y_i)^2 \quad (2.7)$$

Keterangan :

$N$  = Jumlah data

$f_i$  = nilai yang didapat dari model

$y_i$  = nilai data sebenarnya

## 3. Sistem yang Dibangun

Secara umum langkah-langkah membangun sistem yang dibuat terdiri dari *Preprocessing Data*, inisialisasi parameter, training LSTM Network, dan melakukan uji terhadap data testing. Dalam sistem yang dibangun, dataset yang didapatkan diolah terlebih dahulu dengan menggunakan teknik normalisasi *min max scaling*. Dilakukan inisialisasi pada setiap parameter, setelah itu dilakukan training pada jaringan yang dibuat sesuai dengan parameter yang telah ditentukan. Selanjutnya dilakukan uji pada model yang telah didapatkan dari proses training terhadap data testing. Proses tersebut terus diulang hingga mendapatkan model dengan akurasi yang dapat diterima.

### 3.1 Preprocessing Data

Untuk meminimalkan error, dilakukan normalisasi pada dataset dengan mengubah data aktual menjadi nilai dengan range interval  $[0,1]$ . Teknik normalisasi yang digunakan menggunakan *min-max scaling*. Adapun untuk rumus normalisasi *min-max scaling* adalah :

$$X' = \frac{(x - \min_x)}{(\max_x - \min_x)} \quad (3.1)$$

Keterangan :

$X$  : Data yang akan dinormalisasikan

$X'$  : Data setelah dinormalisasikan

$\min_x$  : Nilai minimum dari keseluruhan data

$\max_x$  : Nilai maksimum dari keseluruhan data

### 3.2 Inisialisasi Parameter

Setelah dataset dilakukan *preprocessing*, selanjutnya menentukan inisialisasi parameter-parameter dasar yang dibutuhkan antara lain :

- Nilai *Learning rate*
- Jumlah *Hidden layer*
- Jumlah *Neuron* pada *Hidden Layer*
- Target *Error* yang berupa *Mean Square Error* (MSE)
- Epoch* Maksimum

### 3.3 Training LSTM Network

Berikutnya adalah penjelasan dari proses training pada jaringan LSTM Network yang diusulkan.

- Hitung semua fungsi gates unit pada setiap neurons. Dengan berurut fungsi gates yang akan dihitung adalah *forget gates* dengan persamaan (2.1), fungsi *input gates* dengan persamaan (2.2) dan (2.3), fungsi *cell gates* dengan persamaan (2.4), dan yang terakhir fungsi *output gates* dengan persamaan (2.5) dan (2.6)
- Menghitung fungsi aktivasi linear pada *output layer* dengan rumus :

$$\varphi(x) = x \quad (3.2)$$

- Jika telah melakukan perulangan sebanyak *epoch* yang telah ditentukan, maka berhenti. Jika belum, akan dilakukan optimasi dengan optimasi Adam dan memperbarui bobot dan bias pada sistem, kemudian kembali ke langkah dua.

### 3.4 Testing

Model yang telah didapatkan pada proses training akan diuji dengan menggunakan data testing yang telah didapat dari *preprocessing* data, dengan metode akurasi yang digunakan menggunakan MSE dengan persamaan (2.7).

## 4. Evaluasi

### 4.1 Dataset

Pada penelitian ini dataset yang digunakan menggunakan data jual beli indeks harga Bitcoin terhadap US Dollar yang diambil dari [blockchain.info](https://blockchain.info/charts/market-price?timespan=2years) (dapat diakses pada <https://blockchain.info/charts/market-price?timespan=2years>), data yang digunakan pada tanggal 22 Mei 2016 – 19 Mei 2018. Dataset yang telah ada dilakukan *preprocessing data* dengan menggunakan teknik normalisasi dengan persamaan (3.1). Setelah itu, dibagi menjadi data training dan data testing. Berbagai komposisi data akan dilakukan observasi. Berikut ini adalah berbagai komposisi data yang digunakan :

**Table 4.1 Kombinasi Data**

No.	Data Latih	Data Uji
1	50%	50%
	364 Data Mei 2016 – April 2017	364 Data April 2017 – Mei 2018
2	70%	30%
	510 Data Mei 2016 – September 2017	218 Data Oktober 2017 – Mei 2018

### 4.2 Skenario Pengujian

Skenario pengujian dalam Tugas Akhir ini adalah menganalisa dampak parameter terhadap akurasi yang didapatkan. Parameter yang diuji berupa komposisi data, jumlah pola *time series*, jumlah neuron *hidden*, dan besarnya max epoch untuk menghasilkan bobot LSTM yang optimal. Berikut adalah nilai parameter yang akan diuji :

Kombinasi data : 50% data latih:50% data uji , 70% data latih:30% data uji  
 Jumlah pola *time series* : 1,2,3,4,5  
 Jumlah neuron *hidden* : 5, 10, 15, 25, 30  
 Max Epoch : 100, 500, 1000



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.

#### 4.3 Menganalisis parameter jumlah pola *time series*

Parameter jumlah pola *time series* yang digunakan pada tugas akhir ini adalah 1,2,3,4, dan 5 pola *time series*.

**Table 4.2 Kombinasi Parameter Jumlah Pola Time Series**

Pola Time Series	Neuron hidden	Max epoch	Rata-rata Akurasi Training	Rata-rata Akurasi Testing
1	5	100	95.72 %	90.88 %
2	5	100	95.68 %	88.83 %
3	5	100	94.11 %	88.22 %
4	5	100	95.92 %	83.74 %
5	5	100	95.49 %	86.55 %

Berdasarkan Tabel 4.2, menunjukkan bahwa jumlah pola *time series* 1 menghasilkan hasil yang lebih optimal dibandingkan dengan pola *time series* lainnya. Walaupun pola *time series* 1 dan *time series* 2 memiliki selisih 2.05 % tetapi dalam prediksi itu memiliki nilai yang besar karena 2.05 % dikalikan dengan harga Bitcoin per US Dolar nya. Jumlah pola *time series* merepresentasikan banyaknya pola yang akan dipelajari LSTM. Tidak ada aturan pasti mengenai jumlah banyaknya pola *time series* yang paling optimal dalam memprediksi *time series*. Sehingga jumlah pola *time series* yang didapatkan melalui percobaan.

#### 4.4 Menganalisis parameter jumlah neuron hidden

Parameter jumlah hidden neuron yang digunakan pada tugas akhir ini adalah 5,10,15,20,25, dan 30 hidden neuron.

**Table 4.3 Kombinasi Parameter Neuron Hidden**

Pola Time Series	Neuron hidden	Epoch	Akurasi Rata-rata Training	Akurasi Rata-rata Testing
1	5	100	96.77 %	91.85 %
2	10	100	93.15 %	93.22 %
3	15	100	95.93 %	93.03 %
4	20	100	94.02 %	92.38 %
5	25	100	95.36 %	93.5 %
6	30	100	94.6 %	93.21 %

Berdasarkan lampiran Tabel 4.3, menunjukkan bahwa jumlah neuron hidden 25 menghasilkan hasil yang lebih optimal dibandingkan dengan jumlah neuron hidden lainnya. Walaupun jumlah neuron hidden 25 dan neuron hidden 10 memiliki selisih 0.28 % tetapi dalam prediksi itu memiliki nilai yang besar karena 0.28 % dikalikan dengan harga Bitcoin per US Dolar nya. Hidden neuron mengolah nilai input dan menghubungkannya dengan neuron output, sehingga jumlah hidden neuron akan menentukan nilai output yang dihasilkan oleh LSTM. Tidak ada aturan pasti mengenai jumlah hidden neuron yang paling optimal dalam memprediksi *time series*. Sehingga jumlah hidden neuron yang didapatkan melalui percobaan.

#### 4.5 Menganalisis parameter max epoch

Parameter max epoch yang digunakan pada tugas akhir ini adalah 50, 100, 500, dan 1000 epoch.

**Table 4.4 Kombinasi Parameter Max epoch**

Pola Time Series	Neuron hidden	Epoch	Akurasi Rata-rata Training	Akurasi Rata-rata Testing
1	25	50	95.28 %	93.3 %
1	25	100	94.44 %	93.75 %
1	25	500	94.37 %	86.67 %
1	25	1000	96.79 %	82.57 %

Berdasarkan lampiran Tabel 4.4, menunjukan bahwa max epoch 100 menghasilkan hasil yang lebih optimal dibandingkan dengan max epoch lainnya. Walaupun max epoch 100 dan max epoch 50 memiliki selisih 0.45 % tetapi dalam prediksi itu memiliki nilai yang besar karena 0.45 % dikalikan dengan harga Bitcoin per US Dolar. Max epoch tidak boleh terlalu besar dan terlalu kecil untuk menemukan solusi optimum. Tidak ada aturan pasti mengenai besarnya max epoch yang paling optimal dalam memprediksi time series. Sehingga besarnya max epoch yang didapatkan melalui percobaan.

#### 4.6 Menganalisis parameter komposisi data

Parameter komposisi data yang digunakan pada tugas akhir ini adalah menggunakan dua skema komposisi data.

**Table 4.5 Kombinasi Parameter Komposisi Data**

No	Data	Pola Time Series	Neuron hidden	Epoch	Akurasi Rata-rata Training	Akurasi Rata-rata Testing
1	50% Data Latih : 50% Data Testing	1	25	100	97.39 %	92.27 %
2	70% Data Latih : 30% Data Testing	1	25	100	95.36 %	93.5 %

Berdasarkan lampiran Tabel 4.5, menunjukan bahwa komposisi data yang baik digunakan adalah skema 2. Hal ini dikarenakan skema 2 memiliki rata-rata tingkat akurasi yang baik pada data uji. Walaupun skema 2 dan skema 1 memiliki selisih 1.23 % tetapi dalam prediksi itu memiliki nilai yang besar karena 1.23 % dikalikan dengan harga Bitcoin per US Dolar. Tidak ada aturan pasti mengenai komposisi data latih dan uji yang paling optimal dalam memprediksi time series. Sehingga komposisi data yang didapatkan melalui percobaan.

## 5. Kesimpulan

Pada Tugas Akhir ini, dibangun model LSTM untuk memprediksi harga Bitcoin dengan pengujian parameter komposisi data, jumlah pola *time series*, jumlah hidden neuron dan max epoch. Pada pengujian tersebut didapatkan hasil yang terbaik yaitu dengan komposisi data latih 70% dan data uji 30%, parameter 1 pola *time series*, jumlah 25 neuron hidden, dan max epoch adalah 100 dengan akurasi rata-rata pada data latih 95.36% dan data testing 93.5%.

Saran untuk penelitian selanjutnya, yaitu dapat membangun model LSTM dengan menambahkan fitur baru dalam *memory cell* dan proses optimasi baru yang digunakan.

## 6. Daftar Pustaka

- [1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2009.
- [2] A. Fariza, "Peramalan Time Series".
- [3] A. S. Perdana, "Perbandingan Metode Times Series Regression dan Arimax Pada Permodelan Data Penjualan Pakaian di Boyolali," 2010.
- [4] Suyanto, Artificial Intelligence, Bandung, Jawa Barat: Informatika.
- [5] QIU Mingyue, LI Cheng, SONG Yu, "Application of the Artificial Neural Network in predicting the direction of stock market index", 2016.
- [6] Prabowo Satryo, " Analisis dan Implementasi Jordan Recurrent Neural Network untuk Prediksi Data Time Series Study Kasus Harga Minyak " 2012.
- [7] T. H. Imam, "Analisis dan Implementasi Elman Recurrent Neural Network dan Tabu Search Pada Prediksi Harga Perak", 2014.
- [8] M. K. B. C. T. M. and J. W. , "Speaker-Independent Silent Speech Recognition From Flesh-Point Articulatory," 2017.

- [9] C. Olah, "Understanding LSTM Networks," 2015. [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed December 2017].
- [10] D. P. Kingma and J. L. Ba, "ADAM: A Method for Stochastic Optimization," 2015.

## Lampiran

