

**Jurnal Politeknik Caltex Riau**Terbit Online pada laman <https://jurnal.pcr.ac.id/index.php/jkt/>

| e- ISSN : 2460-5255 (Online) | p- ISSN : 2443-4159 (Print) |

Prediksi Pendapatan Kargo Menggunakan Arsitektur Long Short Term Memory

Bagas Aji Aprian¹, Yufis Azhar² dan Vinna Rahmayanti Setya Nastiti³¹Universitas Muhammadiyah Malang, Teknik Informatika, email: bagasajiaprian@yahoo.com²Universitas Muhammadiyah Malang, Teknik Informatika, email: yufis.az@gmail.com³Universitas Muhammadiyah Malang, Teknik Informatika, email: vinna.nastiti@gmail.com

Abstrak

Angkutan kargo udara Indonesia saat ini mengalami perkembangan yang cukup signifikan. Salah satu layanan kargo yang terdapat di Indonesia yaitu Garuda Indonesia Cargo dan memiliki beberapa kantor cabang. Adanya suatu model perkiraan pendapatan diharapkan dapat memberikan wawasan pada suatu kantor cabang. Penelitian ini mengajukan sebuah prediksi pendapatan menggunakan algoritma Deep Learning yaitu Long Short Term Memory (LSTM). LSTM digunakan karena data yang akan diolah adalah data time series. Hasil akurasi pengujian diukur menggunakan Root Mean Squared Error (RMSE). Data pada penelitian ini adalah pendapatan dari salah satu kantor cabang yaitu Cargo Service Center (CSC) Tangerang City. Data berisi kumpulan-kumpulan transaksi pengiriman barang tiap hari. Data melewati 4 proses preprocessing yaitu subtotal, deteksi outlier, difference dan scaling. Hasil penelitian ini menunjukkan hasil prediksi terbaik yaitu pada komposisi data train 90% dan data test 10% dengan nilai RMSE sebesar data train 641375.70 dan data test 594197.70.

Kata kunci: Long Short Term Memory (LSTM), Prediksi, Deep Learning

Abstract

Indonesian air cargo transportation is currently experiencing quite significant development. One of the cargo services in Indonesia is Garuda Indonesia Cargo and has several branch offices. The existence of a revenue forecast model is expected to provide insight into a branch office. This research proposes an income prediction using the Deep Learning algorithm, Long Short Term Memory (LSTM). LSTM is used because the data to be processed is time series data. The results of testing accuracy are measured using Root Mean Squared Error (RMSE). The data in this study are income from one branch office, the Cargo Service Center (CSC) Tangerang City. Data contains collections of goods delivery transactions every day. The data goes through 4 preprocessing processes, namely subtotal, outlier detection, difference, and scaling. The results of this study show the best prediction results, namely the composition of the 90% train data and 10% test data with RMSE values of train data 641375.70 and test data 594197.70.

Keywords: Long Short Term Memory (LSTM), Prediction, Deep Learning

1. Pendahuluan

Saat ini angkutan kargo udara Indonesia mengalami perkembangan yang cukup signifikan. Kebutuhan manusia yang juga semakin bertambah menyebabkan layanan kargo di Indonesia sangat dibutuhkan. Indonesia adalah negara yang memiliki banyak pulau sehingga dengan adanya layanan kargo udara dapat memberikan kemudahan dalam hal pengiriman barang terutama antar pulau. Salah satu layanan kargo yang terdapat di Indonesia yaitu Garuda Indonesia Cargo. Garuda Indonesia Cargo sendiri telah memiliki beberapa kantor cabang yang tersebar di kota-kota besar Indonesia. Setiap kantor cabang memiliki pendapatan yang berbeda dengan yang lainnya. Adanya suatu model perkiraan pendapatan diharapkan dapat memberikan wawasan pada kantor cabang.

Artificial Intelligence (AI) atau kecerdasan buatan adalah kemampuan mesin untuk meniru kecerdasan layaknya seperti manusia. Machine Learning adalah bagian dari AI. Machine Learning memberi kemampuan pada komputer untuk belajar dengan pemrograman secara tidak langsung. Salah satu kekurangan Machine Learning adalah tidak dapat mengatasi input dan output dalam skala yang besar.

Deep Learning adalah bagian dari Machine Learning. Deep Learning mampu mengatasi kelemahan dari Machine Learning. Deep Learning melewati langkah-langkah manual dari mengekstraksi fitur pada Machine Learning, sehingga Deep Learning dapat langsung memberi input berupa gambar ke algoritma Deep Learning karena algoritma akan secara otomatis menentukan fitur yang berkaitan. Deep Learning menggunakan jaringan saraf tiruan yang terinspirasi oleh struktur dan fungsi otak manusia. Deep learning memungkinkan mempelajari representasi data dengan berbagai tingkat abstraksi menggunakan model komputasi yang terdiri dari beberapa processing layer [1]. Deep Learning adalah tentang bagaimana memberikan nilai secara akurat pada banyak tahapan tersebut [2]. Deep Learning dapat digunakan pada banyak tujuan yaitu supervised learning, unsupervised learning, dan semi-supervised learning karena algoritma serta cara kerjanya sehingga dapat digunakan pada klasifikasi teks, pengenalan citra, dan pengenalan suara [3].

Long Short Term Memory (LSTM) termasuk dalam jenis Recurrent Neural Network (RNN), karena dengan tautan umpan balik yang melekat pada beberapa lapisan jaringan [4]. LSTM pertama kali dikenalkan oleh Hochreiter dan Schmidhuber dan kemudian dikembangkan pada tahun 2000 oleh Schmidhuber. RNN menggunakan koneksi berulang dalam loop, yang memungkinkan informasi tetap ada [5]. Tetapi, LSTM dapat melakukan pelatihan dan gradien kepunahan yang menjadi kesulitan pada RNN [6]. LSTM diciptakan dengan tujuan mengatasi masalah hidden layer [7].

Terdapat beberapa penelitian sebelumnya yang menggunakan arsitektur LSTM. Pada penelitian yang dilakukan oleh M. Rizki [8] memprediksi curah hujan kota Malang menggunakan LSTM. Hasil dari penelitian tersebut mendapatkan nilai error RMSE yang paling rendah yaitu sebesar 11.429 menggunakan komposisi hyperparameter yang paling optimal yaitu 128 neuron hidden layer, 500 epoch, dan komposisi data train dan data test yaitu dengan komposisi data train 80% dan data test 20%. Penelitian lainnya yang dilakukan oleh M. Wildan [9] memprediksi harga bitcoin menggunakan LSTM mendapatkan hasil yang paling optimal dengan komposisi data latih 70% dan data uji 30%, parameter 1 pola time series, 25 neuron hidden layer, dan max epoch 100 dengan akurasi rata-rata pada data training 95.36% dan data testing 93.5%. A. Ubrani [10] memprediksi harga pembukaan pasar pada pasar listrik India yang dideregulasi menggunakan LSTM dan GRU dengan hasil nilai error MAPE untuk nilai data mingguan LSTM sebesar 4%, GRU sebesar 4.5%, untuk nilai data harian LSTM sebesar 6.9%, GRU sebesar 7.1%, dan model LSTM lebih baik daripada model GRU untuk data mingguan dan harian.

Penelitian ini berbeda dengan penelitian terdahulu karena studi kasus yang digunakan baru dari studi kasus yang sebelumnya yaitu prediksi pendapatan suatu perusahaan menggunakan LSTM. Data yang akan diuji adalah pendapatan dari salah satu kantor cabang Garuda Indonesia Cargo yaitu Cargo Service Center (CSC) Tangerang City. Data akan melewati tahapan preprocessing terlebih dahulu agar data yang digunakan dalam kondisi baik dan siap digunakan. Pengujian LSTM dilakukan beberapa kali untuk mendapatkan hasil yang optimal.

2. Metode

2.1 Pengumpulan Data

Data diperoleh dari data transaksi kantor cabang CSC Tangerang City. Data berisi kumpulan-kumpulan transaksi pengiriman barang tiap hari. Data yang diterima berupa hasil rekapan tiap bulan. Data yang diperoleh dengan rentang waktu antara Januari 2016 - September 2019.

2.2 Preprocessing Data

Data yang telah diperoleh kemudian melakukan tahapan preprocessing. Data mentah sebelum digunakan akan melalui preprocessing agar data yang digunakan dapat bekerja dengan optimal. Data akan melewati 4 proses yaitu subtotal pada persiapan data, deteksi outlier, difference dan scaling.

2.2.1 Persiapan Data

Data mentah yang sebelumnya dikumpulkan berupa kumpulan transaksi pengiriman barang tiap hari. Tetapi, transaksi yang dilakukan pada satu hari tidak hanya sekali dan dapat lebih. Sehingga, untuk merubah data menjadi per hari memiliki satu nilai maka dilakukan subtotal pada transaksi yang dilakukan tiap harinya. Selain itu juga menghapus sebanyak 7 kolom yang tidak digunakan dalam penelitian ini. Data subtotal kemudian dikumpulkan dan yang akan digunakan pada penelitian ini hanya variable terima.

2.2.2 Deteksi Outlier

Data yang digunakan terlebih dahulu melakukan deteksi outlier. Apabila sebuah data memiliki outlier dapat membuat hasil suatu pengujian menjadi tidak optimal. Cara untuk mengatasi data outlier yang ada pada data dengan menggunakan boxplot pada Microsoft Excel. Boxplot dapat juga digunakan untuk mendeteksi adanya outlier selain untuk mendeskripsikan data numerik. Data outlier kemudian dihapus sehingga data yang digunakan tidak memiliki outlier.

2.2.3 Difference

Data yang tidak stasioner dapat menjadi stasioner dengan melakukan transformasi difference pada data. Data difference mengurangi 1 dari jumlah data sebelum transformasi karena data pertama tidak memiliki nilai sehingga baris pertama dihapus.

2.2.4 Scaling

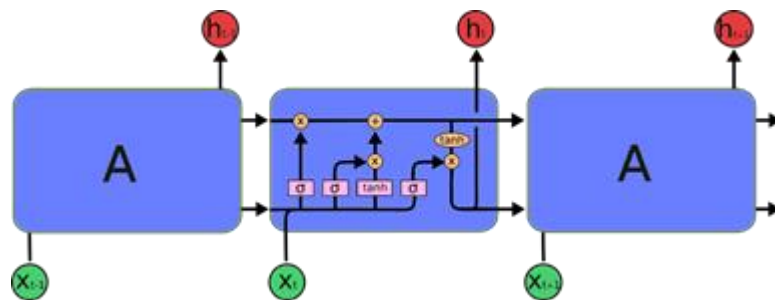
Data kemudian melakukan scaling atau menyesuaikan data numerik. Metode scaling yang akan digunakan adalah MinMaxScaler. MinMaxScaler melakukan scaling data dengan range tertentu berdasarkan nilai minimum hingga nilai maksimum, membuat data berada pada rentang 0 – 1.

$$x_s = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Keterangan rumus: x_s adalah data hasil scaling, x adalah data asli, x_{min} adalah data minimum, x_{max} adalah data maksimum.

2.3 Implementasi Long Short Term Memory (LSTM)

Long Short Term Memory adalah varian dari Recurrent Neural Network (RNN). LSTM dapat mempelajari pola panjang dari data berurut karena mencegah situasi vanishing gradient. Tetapi, LSTM tetap memiliki prinsip yang sama dengan RNN dan yang membedakan dengan RNN yaitu isi cellnya. LSTM digunakan karena data yang akan diolah adalah data rentang waktu dan LSTM sangat cocok untuk memprediksi rentang waktu ketika ada langkah-langkah waktu dengan ukuran yang bebas [4]. Pada penelitian ini menggunakan 4 layer LSTM yaitu 1 input layer, 2 hidden layer, dan 1 output layer. Proses dari implementasi model LSTM yaitu data akan masuk pada input layer pada model, kemudian terjadi proses perhitungan dalam sel LSTM. Berikut pada gambar 1 adalah contoh struktur LSTM.



Gambar 1. Struktur LSTM [4]

Pada gambar 1 terdapat persegi panjang berwarna biru dengan huruf A adalah cell LSTM dengan lingkaran hijau X_t adalah input dan lingkaran merah h_t adalah output. Persegi panjang merah muda adalah layer neuron dengan σ adalah fungsi aktivasi sigmoid dan tanh adalah fungsi aktivasi tanh. Lingkaran berwarna kuning adalah operasi element-wise. Panah hitam adalah aliran informasi di dalam dan antar cell.

Proses yang terjadi di dalam sel LSTM yaitu, pertama data akan masuk ke dalam forget gate yang berfungsi untuk mengidentifikasi informasi yang tidak dibutuhkan dan akan melupakannya dari cell state menggunakan layer sigmoid. Kemudian data akan masuk ke dalam input gate yang berfungsi untuk menentukan informasi baru yang akan disimpan pada cell state menggunakan layer sigmoid dan layer tanh. Selanjutnya memperbarui cell state yang lama dengan mengkalikan dengan hasil forget gate kemudian menambahkan hasil kali dari proses input gate yang telah melewati layer sigmoid dan layer tanh. Terakhir, data akan masuk ke dalam output gate yang berfungsi untuk memutuskan bagian mana dari cell state yang akan digunakan sebagai output melalui layer sigmoid kemudian mengkalikan dengan cell state yang telah melewati layer tanh dan mendapatkan bagian cell state sesuai dengan keputusan sebagai nilai output. Setelah mendapatkan nilai output, maka perhitungan pada sel LSTM tersebut selesai.

Selanjutnya masuk pada hidden layer pertama pada model dan melakukan perhitungan pada sel LSTM yang sama pada layer pertama. Proses yang sama akan terjadi pada layer selanjutnya yaitu hidden layer kedua dan sampai output layer. Proses tersebut akan bekerja sampai mendapatkan hasil yang paling optimal.

Rumus-rumus Long Short Term Memory pada setiap sesi.

Forget Gate [11]:

$$f_t = \sigma(W_{fx}X_t + W_{fh}h_{t-1} + W_{fc}C_{t-1} + b_f) \quad (2)$$

Input Gate [11]:

$$i_t = \sigma(W_{ix}X_t + W_{ih}h_{t-1} + W_{ic}C_{t-1} + b_i) * \tanh(W_{cx}X_t + W_{ch}h_{t-1} + b_i) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t \quad (4)$$

Output Gate [11]:

$$O_t = \sigma(W_{ox}X_t + W_{oh}h_{t-1} + W_{oc}C_{t-1} + b_o) \quad (5)$$

$$h_t = O_t * \tanh(C_t) \quad (6)$$

Keterangan rumus: f_t adalah Forget Gate, i_t adalah Input Gate, C_t adalah Update Cell State, O_t adalah Output Gate, h_t adalah Nilai Output, X_t adalah Nilai Input, h_{t-1} adalah Nilai Output Cell sebelumnya, C_{t-1} adalah Cell State sebelumnya, b adalah Bias, W adalah Weight, σ adalah Sigmoid.

2.4 Uji Validitas RMSE

Uji Validitas menggunakan Root Mean Squared Error (RMSE) dilakukan untuk mengukur hasil akurasi pengujian. RMSE merupakan salah satu contoh parameter yang biasa digunakan sebagai indikator untuk mengukur dan membandingkan kemiripan hasil prediksi dan data asli. RMSE merupakan nilai rata-rata dari jumlah kuadrat kesalahan.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2} \quad (7)$$

Keterangan rumus: n adalah jumlah data, x_i adalah data asli pada waktu i , y_i adalah data prediksi pada waktu i .

3. Hasil dan Pembahasan

Pengujian pada penelitian ini menggunakan IDE Jupyter Notebook untuk melakukan prediksi pendapatan menggunakan metode LSTM. Pengujian dilakukan menggunakan 1 unit laptop dengan spesifikasi Hardware: Processor Intel Core i5-6200U CPU @ 2.30 GHz, Tipe Sistem 64-bit, Memory 4 GB, dan Sistem Operasi Windows 10 Pro. Setiap pengujian dilakukan dengan komposisi data train dan data test yang berbeda, kemudian mencari jumlah neuron hidden layer dan epoch yang paling optimal. Setelah mendapatkan hyperparameter yang optimal, melakukan pengujian pada komposisi data train dan data test yang telah ditetapkan sekali lagi.

Data yang digunakan telah melewati 4 proses preprocessing yaitu subtotal pada persiapan data, deteksi outlier, difference dan scaling dengan tujuan data siap digunakan serta mendapatkan hasil yang lebih baik.

3.1 Hasil Pengujian Jumlah Neuron Hidden Layer

Pengujian jumlah neuron hidden layer dengan komposisi data dan jumlah neuron hidden layer yang berbeda-beda untuk mendapatkan hasil yang paling optimal.

Tabel 1. Kombinasi Jumlah Neuron Hidden Layer

No	Komposisi Data	Neuron Hidden Layer	RMSE Train Data	RMSE Test Data
1	data train 50% dan data test 50%	16	829739.08	836596.13

2	data train 50% dan data test 50%	32	663686.33	662954.32
3	data train 50% dan data test 50%	64	652326.78	650307.98
4	data train 50% dan data test 50%	128	648466.19	648115.79
5	data train 50% dan data test 50%	256	671472.16	667542.49
6	data train 50% dan data test 50%	512	835291.95	841932.47
7	data train 60% dan data test 40%	16	842007.35	802968.55
8	data train 60% dan data test 40%	32	689369.81	650965.41
9	data train 60% dan data test 40%	64	663039.68	624376.94
10	data train 60% dan data test 40%	128	662768.99	621805.69
11	data train 60% dan data test 40%	256	664147.06	623471.56
12	data train 60% dan data test 40%	512	853183.17	811925.42
13	data train 70% dan data test 30%	16	726309.35	675763.78
14	data train 70% dan data test 30%	32	671473.14	615210.79
15	data train 70% dan data test 30%	64	664263.86	608313.83
16	data train 70% dan data test 30%	128	666300.75	613582.33
17	data train 70% dan data test 30%	256	686107.51	634269.87
18	data train 70% dan data test 30%	512	852227.04	795759.85
19	data train 80% dan data test 20%	16	836121.90	802440.97
20	data train 80% dan data test 20%	32	677658.16	636603.26
21	data train 80% dan data test 20%	64	653162.59	613362.11
22	data train 80% dan data test 20%	128	653858.38	610696.59
23	data train 80% dan data test 20%	256	661585.48	615755.12
24	data train 80% dan data test 20%	512	840367.68	805776.17
25	data train 90% dan data test 10%	16	749122.09	685113.87
26	data train 90% dan data test 10%	32	666843.22	599600.12

27	data train 90% dan data test 10%	64	650099.21	594401.29
28	data train 90% dan data test 10%	128	655327.92	595058.44
29	data train 90% dan data test 10%	256	651650.83	587911.37
30	data train 90% dan data test 10%	512	839514.28	759988.16

Berdasarkan tabel 1, menunjukan bahwa 128 neuron hidden layer pada data train 50% dan data test 50% memiliki nilai yang paling optimal dibandingkan dengan jumlah neuron hidden layer dan komposisi yang lainnya dengan nilai RMSE data train sebesar 648466.19. Pada prediksi data time series, tidak ada aturan pasti jumlah neuron hidden layer yang optimal karena tergantung dengan data yang digunakan pada percobaan tersebut. Sehingga untuk mendapatkan hasil yang optimal dapat dilakukan dengan percobaan.

3.2 Hasil Pengujian Jumlah Epoch

Pengujian jumlah epoch menggunakan 128 neuron hidden layer dengan komposisi data train 50% dan data test 50%.

Tabel 2. Kombinasi Jumlah Epoch

No	Neuron Hidden Layer	Epoch	Train Loss	Test Loss
1	128	100	0.027042	0.028801
2	128	200	0.024540	0.028489
3	128	300	0.024134	0.028478
4	128	400	0.023547	0.028261
5	128	500	0.021477	0.023836
6	128	600	0.017006	0.017719
7	128	700	0.016869	0.017520
8	128	800	0.015518	0.017930
9	128	900	0.015492	0.017585
10	128	1000	0.015998	0.017115

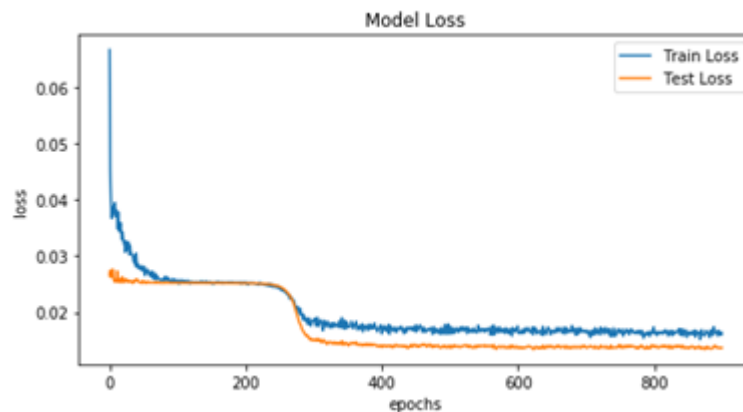
Berdasarkan tabel 2, menunjukan bahwa jumlah 900 epoch memiliki hasil train loss yang paling optimal dengan nilai train loss pada model sebesar 0.015492. Pada prediksi data time series, tidak ada aturan pasti jumlah epoch yang optimal karena tergantung dengan data yang digunakan pada percobaan tersebut. Sehingga untuk mendapatkan hasil yang optimal dapat dilakukan dengan percobaan.

Setelah mendapatkan jumlah neuron hidden layer dan epoch yang paling optimal, melakukan pengujian sekali lagi dengan kombinasi hyperparameter optimal tersebut.

Tabel 3. Kombinasi Hyperparameter Optimal

No	Komposisi Data	Neuron Hidden Layer	Epoch	RMSE Train Data	RMSE Test Data
----	----------------	---------------------	-------	-----------------	----------------

1	data train 50% dan data test 50%	128	900	648408.28	649151.41
2	data train 60% dan data test 40%	128	900	654693.06	618770.27
3	data train 70% dan data test 30%	128	900	655382.25	603286.15
4	data train 80% dan data test 20%	128	900	644197.87	606569.65
5	data train 90% dan data test 10%	128	900	641375.70	594197.70

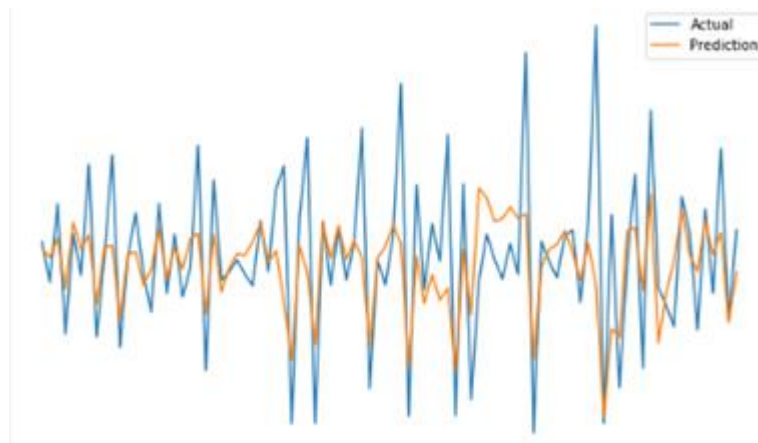


Gambar 2. Grafik Model Loss Data Train 90% dan Data Test 10%

Berdasarkan tabel 3, menunjukan bahwa hasil yang paling optimal adalah dengan hyperparameter 128 neuron hidden layer, 900 epoch, dan komposisi data train 90% dan data test 10% dengan nilai RMSE sebesar data train 641375.70 dan data test 594197.70.

Terlihat pada gambar 2 adalah grafik model loss dari kombinasi hyperparameter yang optimal tersebut menunjukan bahwa grafik model loss yang dihasilkan cukup bagus karena nilai loss data train dan data test menurun seiring dengan epoch, serta loss data test yang tidak melebihi data train pada akhir yang menunjukkan bahwa model tidak overfit.

Hasil prediksi perlu dilakukan perbandingan dengan data asli untuk mengukur kesalahan dari hasil prediksi. Pada gambar 3 adalah grafik perbandingan data asli dengan data hasil prediksi.



Gambar 3. Grafik Data Asli dan Data Prediksi

Berdasarkan gambar 3, menunjukkan bahwa hasil prediksi memiliki akurasi cukup baik mendekati nilai aslinya meskipun sedikit dan masih banyak data prediksi yang jauh dari nilai aslinya. Tetapi, data prediksi memiliki nilai yang turun dan naik sama dengan nilai asli cukup banyak. Dengan tingkat akurasi kesalahan 16.67% pada nilai penurunan dan kenaikan data prediksi. Sebanyak 15 data dari 90 perbandingan data yang mengalami kesalahan.

4. Kesimpulan

Setelah melakukan rancangan, implementasi, pengujian dan melakukan analisis pada prediksi pendapatan CSC Tangerang City menggunakan arsitektur Long Short Term Memory (LSTM), dapat disimpulkan bahwa:

1. Jumlah neuron hidden layer yang paling optimal adalah sebanyak 128 neuron pada pengujian komposisi data train 50% dan data test 50% dengan nilai RMSE data train sebesar 648466.19.
2. Jumlah epoch yang paling optimal yaitu 900 epoch dengan nilai train loss pada model sebesar 0.015492.
3. Pengujian dengan hyperparameter optimal mendapatkan hasil terbaik yaitu pada komposisi data train 90% dan data test 10% dengan nilai RMSE sebesar data train 641375.70 dan data test 594197.70.
4. Penelitian ini menunjukkan hasil prediksi tergolong cukup baik terlihat pada gambar 3. Model LSTM dapat memprediksi pendapatan dengan cukup optimal, hanya saja memiliki kekurangan pada akurasinya dengan akurasi kesalahan sebesar 16.67% pada nilai penurunan dan kenaikan data prediksi.
5. Nilai RMSE yang dihasilkan besar disebabkan karena beberapa alasan, yaitu pengujian RMSE dilakukan setelah model prediksi mengembalikan nilai dari scaling (range nilai 0-1) menjadi nilai normal sebelum scaling dan tergantung pada data yang digunakan. Data memiliki range nilai minimum -1633033 dan nilai maksimum 2166061. Selain itu data yang digunakan tidak cukup banyak dan memiliki nilai variasi yang besar sehingga data yang digunakan tidak memiliki pola dan tidak ada data memiliki nilai yang sama pada satu variabel. Sehingga, menyulitkan model pada proses learning dan prediksi.

Berdasarkan kesimpulan dan hasil yang telah didapat, adapun beberapa saran adalah sebagai berikut:

1. Diperlukan data perusahaan yang lebih banyak untuk mendapatkan hasil prediksi yang lebih baik.
2. Menggunakan komputer dengan spesifikasi tinggi untuk mempercepat proses learning pada arsitektur LSTM terutama ketika menggunakan data yang sangat banyak.
3. Perlu melakukan pengkajian ulang dengan hyperparameter yang berbeda dan menambahkan parameter baru yang dapat mempengaruhi hasil prediksi kedepannya untuk mendapatkan hasil yang optimal.

Daftar Pustaka

- [1] C. Hu, Q. Wu, H. Li, S. Jian, N. Li, and Z. Lou, "Deep learning with a long short-term memory networks approach for rainfall-runoff simulation," *Water (Switzerland)*, vol. 10, no. 11, pp. 1–16, 2018, doi: 10.3390/w10111543.
- [2] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.
- [3] G. Ghazali, Jondri, and M. Si, "Prediksi saham menggunakan dbn (deep belief network) stock prediction using dbn (deep belief network)," *eProceedings Eng.*, vol. 4, no. 1, pp. 1258–1273, 2017.
- [4] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and longshort term memory," *J. Cheminform.*, vol. 24, no. 4, pp. 1–11, 2018, doi: 10.6084/m9.figshare.5028110.
- [5] S. Li, H. Fang, and B. Shi, "Multi-Step-Ahead Prediction with Long Short Term Memory Networks and Support Vector Regression," *Chinese Control Conf. CCC*, vol. 2018-July, pp. 8104–8109, 2018, doi: 10.23919/ChiCC.2018.8484066.
- [6] Z. Chen, Y. Liu, and S. Liu, "Mechanical state prediction based on LSTM neural network," *Chinese Control Conf. CCC*, pp. 3876–3881, 2017, doi: 10.23919/ChiCC.2017.8027963.
- [7] M. A. D. Suyudi, "Prediksi Harga Saham menggunakan Metode Recurrent Neural Network," *Wikipedia*, vol. 052, no. 735, pp. 3–6, 2015.
- [8] M. Rizki, "Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory(Lstm) Untuk Prediksi Curah Hujan Kota Malang," vol. 2, no. 3, pp. 331–338, 2019.
- [9] M. Wildan, P. Aldi, and A. Aditsania, "Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin," *e-Proceeding Eng.*, vol. 5, no. 2, pp. 3548–3555, 2018.
- [10] A. Ubrani and S. Motwani, "LSTM- and GRU-based time series models for market clearing price forecasting of Indian deregulated electricity markets," *Adv. Intell. Syst. Comput.*, vol. 898, pp. 693–700, 2019, doi: 10.1007/978-981-13-3393-4_70.
- [11] Q. Wang, Y. Guo, L. Yu, and P. Li, "Earthquake Prediction based on Spatio-Temporal Data Mining: An LSTM Network Approach," *IEEE Trans. Emerg. Top. Comput.*, vol. 6750, no. c, pp. 1–1, 2017, doi: 10.1109/tetc.2017.2699169.