

PREDIKSI TINGKAT TEMPERATUR KOTA SEMARANG MENGUNAKAN METODE *LONG SHORT-TERM MEMORY* (LSTM)

Rahmatul Akbar^{1*}, Rukun Santoso², Budi Warsito³

^{1,2,3}Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro

*e-mail: akb4rrmt@gmail.com

DOI: 10.14710/j.gauss.11.4.572-579

Article Info:

Received: 2022-08-05

Accepted: 2022-10-17

Available Online: 2023-02-25

Keywords:

Temperature; Long Short-Term Memory; Hyperparameters

Abstract: Temperature is one of the most important attributes of climate, temperature affects life in many different ways such as in agriculture, aviation, energy, and life in general. Temperature prediction is needed to make the right step to prevent the negative impact of climate change. Long Short-Term Memory (LSTM) is the method that can predict time series data, using the unique design of neural networks, LSTM can help to prevent vanishing gradient from happening which allows LSTM model to use more data from the past to predict the future. Hyperparameters like LSTM unit, epochs, and batch size are used to make the best model, the best model is the one with the lowest loss function. This research used climate data from 1 January 2019 until 31 December 2021 consist of 1096 data in total. The best prediction in this research is made by the model with 70% training data, 0,009 learning rate, 128 LSTM unit, 16 batch size, and 100 epochs with the lowest loss function of 0,013, this model gives MAPE value of 1,896016% and RMSE value of 0,725.

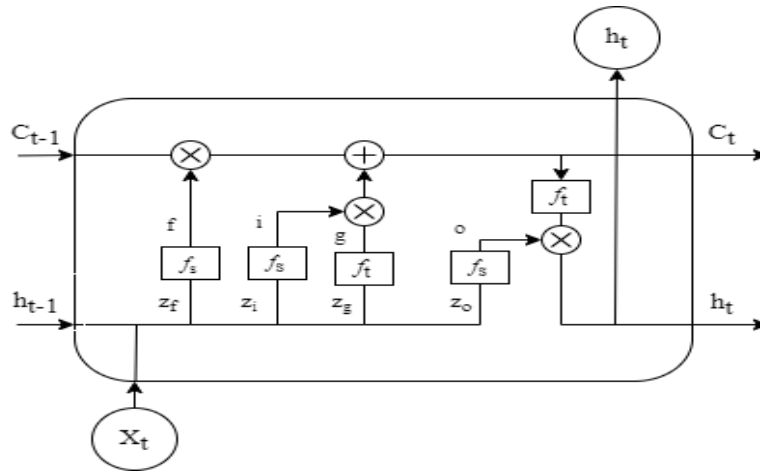
1. PENDAHULUAN

Iklim dapat didefinisikan sebagai kondisi rata-rata suhu udara, curah hujan, tekanan udara, arah angin, kelembapan udara, dan parameter iklim lainnya dalam jangka waktu yang panjang (Tjasyono, 2004). Perubahan iklim regional dan global menjadi pusat perhatian dalam beberapa dekade terakhir. Menurut BMKG tren suhu maksimum di Jakarta telah meningkat signifikan sebesar $2,12^{\circ}\text{C}$ per 100 tahun. Demikian pula pada lebih dari 80 stasiun BMKG untuk pengamatan suhu udara di Indonesia dalam periode 30 tahun terakhir (Supari dkk, 2017). Semarang merupakan salah satu kota terpanas yang ada di Indonesia. Pada tahun 2019 tercatat suhu tertinggi sejak tahun 2002 yaitu sebesar $39,4$ derajat Celsius. Tingginya suhu udara cenderung menyulitkan masyarakat dalam beraktivitas sehari-hari.

Prediksi data *time series* memiliki banyak manfaat yang sangat dirasakan pada masa sekarang. Instansi pemerintah dan perusahaan swasta sangat banyak menggunakan hasil prediksi sebagai salah satu acuan dalam mengambil keputusan yang tepat. *Deep Learning* menjadi salah satu metode yang banyak digunakan untuk melakukan prediksi data *time series* karena memiliki performa yang baik dalam menemukan pola data. Salah satu metode *deep learning* yang dapat digunakan untuk melakukan prediksi data *time series* adalah *long short-term memory* (LSTM) yang merupakan bentuk pengembangan dari *recurrent neural network* (RNN). LSTM memiliki kelebihan untuk dapat mengatasi masalah ketergantungan jangka panjang. Diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997, model LSTM terdiri dari serangkaian sel memori unik yang menggantikan *neuron* lapisan tersembunyi dari RNN. RNN merupakan bentuk pengembangan dari *artificial neural network* (ANN). Pengembangan tersebut dilakukan karena ANN tidak mampu melakukan proses pelatihan secara cepat pada data yang banyak. *Recurrent neural network* (RNN) dirancang untuk memproses data bersambung atau berurutan, serta biasa digunakan untuk menyelesaikan masalah dengan data *time series*. Tapi RNN tidak mampu mengingat data historis yang terlalu lama (Kinanti, 2021).

2. TINJAUAN PUSTAKA

LSTM diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997 yang merupakan bentuk khusus dari RNN, didesain untuk mengatasi masalah ketergantungan jangka panjang. *Long Short-Term Memory* mampu mengingat informasi untuk periode yang panjang dengan mudah karena memiliki struktur jaringan berulang yang unik. LSTM memiliki tiga jenis *gate* yang memiliki fungsi masing-masing dalam mengatur informasi pada *cell state*.



Gambar 1. Sel LSTM

Langkah pertama dari LSTM adalah menentukan informasi mana yang akan dibuang dari *cell state*, hal ini diputuskan oleh lapisan sigmoid yang disebut dengan *forget gate*. Pada gambar dapat dilihat sel LSTM menerima *output* h_{t-1} dari sel sebelumnya dan informasi baru x_t dari data sekarang sebagai *input* yang dikombinasikan sebagai vektor $[h_{t-1}, x_t]$. Langkah kedua dari *cell* LSTM adalah menentukan seberapa besar informasi baru yang akan ditambahkan ke dalam *cell state*. Proses ini memiliki dua bagian, pertama lapisan sigmoid menentukan nilai yang akan diperbarui dari *cell state*, lalu bagian kedua lapisan tanh membuat vektor dari kandidat baru, lalu keduanya dikombinasikan untuk melakukan pembaruan pada *cell state*. *Cell state* berfungsi untuk membawa informasi dari sel di belakang ke sel-sel LSTM selanjutnya, pada setiap *timestep cell state* akan diperbarui dengan menggunakan *forget gate* dan *input gate* untuk menentukan informasi yang akan dibuang ataupun ditambahkan ke dalam *cell state*. *output gate* berguna untuk menentukan *output* dari *cell state* sekarang. Pertama, lapisan sigmoid menentukan bagian dari *cell state* yang menjadi *output*. Lalu, lapisan tanh akan mengubah nilai *cell state* menjadi antara -1 dan 1, kemudian nilai dari lapisan sigmoid dan lapisan tanh dikalikan (Olah, 2015). Berikut persamaan-persamaan yang digunakan pada sel LSTM:

$$z_f = W_{xf} x_t + W_{hf} h_{t-1} + b_f \quad (1)$$

$$f_t = f_s(z_f) \quad (2)$$

$$z_g = W_{xg} x_t + W_{hg} h_{t-1} + b_g \quad (3)$$

$$g_t = f_t(z_g) \quad (4)$$

$$z_i = W_{xi} x_t + W_{hi} h_{t-1} + b_i \quad (5)$$

$$i_t = f_s(z_i) \quad (6)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ g_t \quad (7)$$

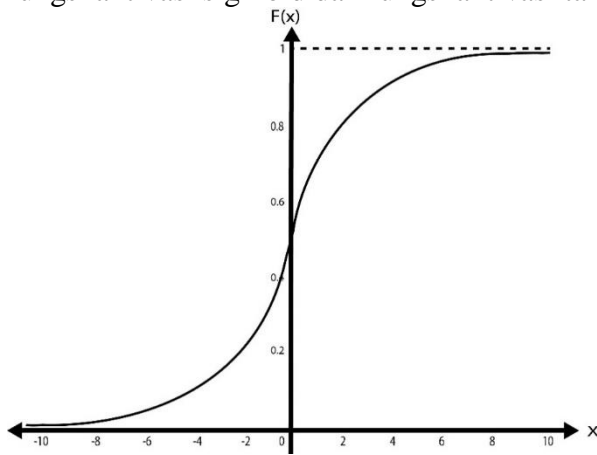
$$\mathbf{z}_o = \mathbf{W}_{x_o} \mathbf{x}_t + \mathbf{W}_{h_o} \mathbf{h}_{t-1} + \mathbf{b}_o \quad (8)$$

$$\mathbf{O}_t = f_s(\mathbf{z}_o) \quad (9)$$

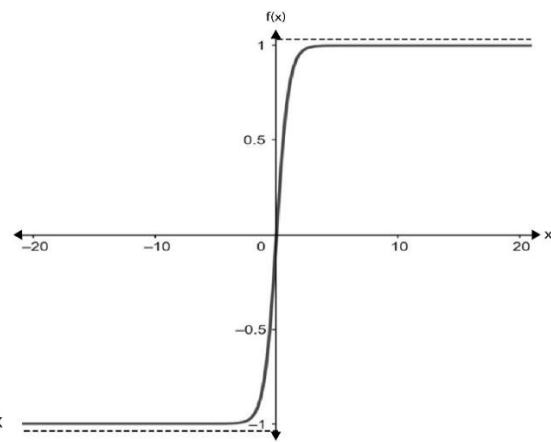
$$\mathbf{h}_t = \mathbf{O}_t \circ f_t(\mathbf{C}_t) \quad (10)$$

Simbol \mathbf{f}_t merupakan *forget gate*, \mathbf{i}_t merupakan lapisan sigmoid, \mathbf{g}_t merupakan *candidate state*. \mathbf{C}_t merupakan *cell state* merupakan, \mathbf{O}_t *output gate*, \mathbf{h}_t merupakan *hidden state*, \mathbf{W} merupakan matrik bobot, \mathbf{h}_{t-1} merupakan *hidden state* sebelumnya, \mathbf{x}_t merupakan data baru, \mathbf{b} merupakan bias, f_s merupakan fungsi aktivasi sigmoid, dan f_t merupakan fungsi aktivasi tanh.

Sel LSTM menggunakan fungsi aktivasi sigmoid dan fungsi aktivasi tanh. Fungsi aktivasi sigmoid merubah nilai *input* ke dalam interval [0,1] dan fungsi aktivasi tanh merubah nilai *input* ke interval [-1,1]. Gambar 2 dan 3 masing-masing menunjukkan grafik fungsi aktivasi sigmoid dan fungsi aktivasi tanh.



Gambar 2. Fungsi Aktivasi Sigmoid



Gambar 3. Fungsi Aktivasi tanh

Persamaan 11 dan 12 adalah persamaan fungsi aktivasi sigmoid dan fungsi aktivasi tanh.

$$f_s(x) = \frac{1}{1+e^{-x}}, -\infty < x < \infty \quad (11)$$

$$f_t(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, -\infty < x < \infty \quad (12)$$

3. METODE PENELITIAN

Jenis data yang digunakan dalam penelitian ini adalah data sekunder yang didapat dari website BMKG (http://dataonline.bmkg.go.id/akses_data). Data yang dipakai adalah data harian dari tanggal 1 Januari 2019 sampai dengan tanggal 31 Desember 2021 yang berjumlah 1096 data. Data cuaca ini merupakan hasil pengamatan yang dilakukan oleh Stasiun Klimatologi Semarang. Variabel yang digunakan dalam penelitian ini terbagi menjadi dua, yaitu variabel bebas dan variabel terikat. Variabel terikat merupakan tingkat temperatur dan variabel bebas berjumlah 8 variabel yaitu lama penyinaran matahari, curah hujan, kecepatan angin rata-rata, kelembapan rata-rata, temperatur minimum, temperatur maksimum, kecepatan angin maksimum, dan arah angin. Data yang telah diperoleh kemudian diolah dengan langkah-langkah sebagai berikut:

1. Menginput data ke *google colab*
2. Melakukan *preprocessing* data
3. Melakukan normalisasi data dengan metode *MinMaxScaler*

4. Validitas variabel *input* dengan metode RFE
5. Mengubah struktur data dengan metode *sliding windows*
6. Membagi data latih dan data uji
7. Mengubah bentuk *input* data menjadi *samples, timestep, dan features*
8. Membuat model LSTM
9. Melakukan pelatihan data
10. Jika nilai *loss function* berubah maka proses pelatihan dilanjutkan
11. Jika nilai *loss function* tidak berubah maka proses pelatihan berhenti
12. Evaluasi model
13. Prediksi

4. HASIL DAN PEMBAHASAN

Data dinormalisasi agar semua variabel memiliki pengaruh terhadap model yang dibuat, nilai data diubah ke dalam interval $[0,1]$. Metode yang digunakan adalah metode *min-max scaling*. Algoritma RFE digunakan untuk mengambil lima variabel terbaik berdasarkan nilai *feature importance* dari model *decision tree*. Tabel 1 memperlihatkan lima variabel yang menghasilkan model terbaik.

Tabel 1. Lima Variabel Terbaik

Kolom	Selected True/False
0	Selected True
1	Selected False
2	Selected True
3	Selected False
4	Selected True
5	Selected False
6	Selected True
7	Selected True

Variabel terbaik adalah variabel yang berada pada kolom 0,2,4,6, dan 7 pada dataset yaitu kelembapan rata-rata, lama penyinaran matahari, arah angin maksimum, temperatur minimum dan temperatur maksimum.

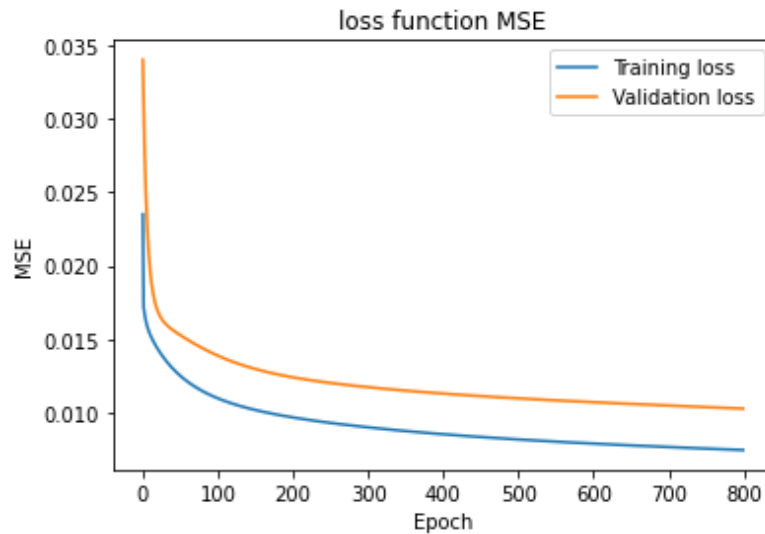
Model LSTM yang akan dibentuk adalah model dengan menggunakan metode optimasi *stochastic gradient descent* (SGD). Pemilihan model terbaik akan dilihat dari nilai *loss function* terkecil. pada penelitian kali ini akan dicobakan tiga proporsi pembagian data, data dibagi menjadi 60% data latih dan 40% data uji, 70% data latih dan 30% data uji dan terakhir 80% data latih dan 20% data uji. Untuk menemukan model terbaik maka perlu dicoba kombinasi dari beberapa parameter sehingga didapat model dengan nilai *loss function* terkecil, parameter yang akan digunakan meliputi *learning rate*, LSTM unit, jumlah *epochs*, dan *batch size*, tabel 2 merupakan rata-rata *loss function* dari masing-masing model.

Tabel 2. *Loss function*

<i>LSTM Unit</i>	<i>Batch size</i>	<i>Epochs</i>	<i>Learning Rate</i>								
			0,009			0,01			0,02		
			Data Latih	Data Latih	Data Latih	Data Latih	Data Latih	Data Latih	Data Latih	Data Latih	Data Latih
			60%	70%	80%	60%	70%	80%	60%	70%	80%

		100	0,026	0,016	0,022	0,030	0,016	0,020	0,028	0,022	0,024
		200	0,021	0,018	0,019	0,018	0,015	0,021	0,062	0,018	0,026
	4	400	0,023	0,018	0,015	0,024	0,017	0,019	0,027	0,026	0,021
		800	0,022	0,020	0,021	0,025	0,018	0,016	0,026	0,025	0,023
		100	0,022	0,015	0,019	0,023	0,020	0,019	0,022	0,020	0,018
	8	200	0,023	0,019	0,022	0,021	0,024	0,017	0,030	0,022	0,019
		400	0,027	0,016	0,014	0,028	0,016	0,017	0,025	0,021	0,016
		800	0,021	0,020	0,024	0,020	0,016	0,020	0,021	0,019	0,017
		100	0,020	0,015	0,016	0,020	0,018	0,016	0,020	0,016	0,018
	16	200	0,019	0,018	0,016	0,024	0,016	0,016	0,020	0,016	0,018
		400	0,019	0,015	0,014	0,021	0,015	0,017	0,023	0,016	0,018
		800	0,022	0,017	0,017	0,019	0,015	0,017	0,023	0,017	0,020
		100	0,040	0,019	0,025	0,021	0,020	0,021	0,034	0,023	0,025
	4	200	0,024	0,017	0,016	0,029	0,016	0,018	0,031	0,025	0,032
		400	0,025	0,015	0,022	0,025	0,016	0,023	0,041	0,024	0,028
		800	0,023	0,018	0,022	0,022	0,019	0,020	0,034	0,026	0,036
		100	0,018	0,015	0,016	0,023	0,017	0,021	0,018	0,021	0,018
	8	200	0,029	0,017	0,021	0,023	0,016	0,020	0,031	0,020	0,018
		400	0,016	0,020	0,018	0,020	0,016	0,023	0,025	0,020	0,018
		800	0,019	0,018	0,019	0,021	0,016	0,019	0,030	0,017	0,018
		100	0,022	0,018	0,014	0,017	0,016	0,015	0,022	0,016	0,016
	16	200	0,017	0,016	0,014	0,021	0,016	0,017	0,023	0,017	0,018
		400	0,020	0,017	0,014	0,019	0,014	0,021	0,022	0,019	0,018
		800	0,017	0,015	0,016	0,021	0,014	0,018	0,020	0,015	0,014
		100	0,023	0,017	0,018	0,039	0,019	0,023	0,036	0,026	0,033
	4	200	0,027	0,021	0,016	0,038	0,024	0,023	0,036	0,025	0,036
		400	0,034	0,019	0,020	0,030	0,019	0,018	0,040	0,025	0,035
		800	0,032	0,019	0,025	0,027	0,021	0,021	0,032	0,027	0,035
		100	0,020	0,017	0,017	0,027	0,014	0,019	0,022	0,023	0,022
	8	200	0,021	0,015	0,022	0,024	0,015	0,024	0,021	0,024	0,024
		400	0,016	0,017	0,017	0,023	0,015	0,017	0,027	0,024	0,020
		800	0,026	0,015	0,019	0,021	0,016	0,026	0,021	0,019	0,019
		100	0,020	0,016	0,014	0,020	0,015	0,016	0,021	0,016	0,020
	16	200	0,017	0,015	0,018	0,023	0,016	0,020	0,020	0,017	0,016
		400	0,015	0,015	0,018	0,019	0,016	0,019	0,028	0,016	0,023
		800	0,018	0,013	0,019	0,017	0,015	0,019	0,022	0,016	0,023

Tabel 2 menunjukkan model terbaik dengan *loss function* terkecil 0,013 diperoleh oleh model dengan konfigurasi 70% data latih, 0,009 *learning rate*, 128 LSTM unit, 16 *batch size*, dan 800 *epochs*.



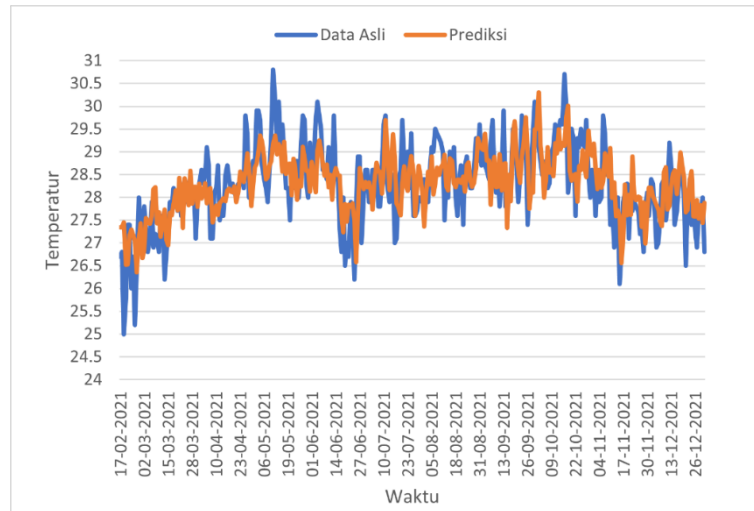
Gambar 3. Grafik *Loss Function*

Gambar 4 memperlihatkan bahwa nilai loss function dari data latih dan data validasi mengalami penurunan seiring pertambahan epochs. Ini menandakan bahwa proses pelatihan model telah berlangsung dengan baik. Model terbaik digunakan untuk melakukan prediksi terhadap data uji, dimana hasilnya digunakan untuk melihat keakuratan dari model. Tabel 3 menunjukkan prediksi data uji untuk lima data awal dan lima data terakhir.

Tabel 3. Prediksi Data Uji

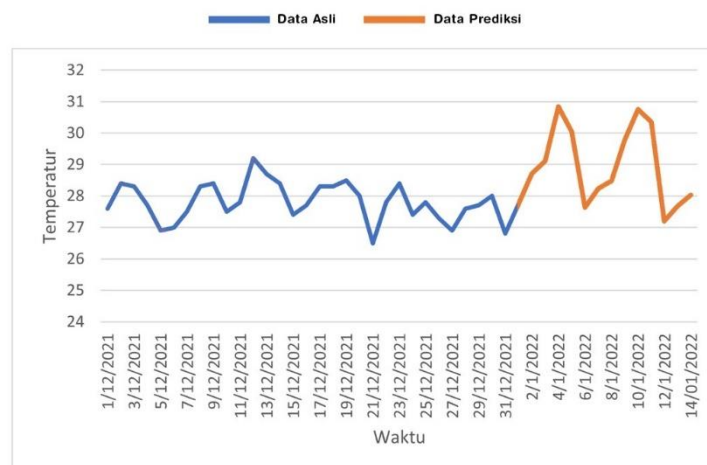
Waktu	Prediksi	Data Asli
17-02-2021	27,33856	26,7
18-02-2021	27,34163	26,8
19-02-2021	27,44883	25,0
20-02-2021	26,51868	25,8
21-02-2021	26,52719	27,4
...
27-12-2021	27,94138	26,9
28-12-2021	27,54134	27,6
29-12-2021	27,83398	27,7
30-12-2021	27,44327	28,0
31-12-2021	27,87586	26,8

Gambar 5 menunjukkan hasil prediksi sudah dapat mengikuti pola data asli dengan cukup baik, hal ini menunjukkan bahwa model sudah memiliki performa yang baik dalam melakukan prediksi. tetapi hasil model yang dibentuk terlihat belum bisa memprediksi data ekstrem dengan optimal, hal ini dapat disebabkan oleh konfigurasi model yang belum maksimal ataupun variabel bebas yang digunakan tidak lengkap.



Gambar 4. Grafik Prediksi Temperatur Kota Semarang

Dari hasil prediksi dan data asli didapat nilai MAPE sebesar 1,896016 %. Nilai ini menunjukkan nilai rata-rata variansi dari *error* sebesar 1,896016 %, nilai MAPE yang kecil menunjukkan model yang didapat memiliki kemampuan prediksi yang baik digunakan untuk melakukan prediksi temperatur Kota Semarang. Nilai *root mean square error* (RMSE) yang diperoleh untuk hasil prediksi adalah sebesar 0,725, yang berarti standar deviasi dari *error* adalah sebesar 0,725. RMSE dapat dibandingkan dengan nilai interval data uji yang sebesar 5,8, nilai RMSE masih kecil jika dibandingkan dengan interval data, maka dapat disimpulkan bahwa hasil prediksi cukup baik.



Gambar 5. Grafik Data Asli dan Prediksi

Gambar 6 menunjukkan grafik data sebulan terakhir dan 14 data hasil prediksi. hasil prediksi terlihat masih berada dalam nilai wajar, ini menunjukkan bahwa model cukup bagus dalam melakukan prediksi temperatur untuk 14 hari ke depan di Kota Semarang.

5. KESIMPULAN

Kemampuan LSTM dalam memprediksi data dipengaruhi oleh kombinasi rancangan dan parameter di dalam model, model dengan 70% data latih memberikan hasil yang lebih baik daripada model dengan 60% dan 80% data latih, hal ini terjadi karena data latih yang sedikit dapat menyebabkan *over fitting* pada model dan data latih yang terlalu banyak membuat model susah dalam mempelajari data. Jumlah *epochs* yang tinggi akan memberikan nilai *loss function* yang lebih rendah tetapi membutuhkan waktu latih yang lebih lama, Parameter *batch size* membantu mempercepat proses pelatihan, semakin besar jumlah *batch size* maka semakin cepat proses pelatihan. Model terbaik adalah model dengan *loss function* terkecil 0,013 diperoleh oleh model dengan konfigurasi 70% data latih, 0,009 *learning rate*, 128 LSTM unit, 16 *batch size*, dan 800 *epochs*. Model memiliki kelemahan dalam memprediksi data ekstrem yang dapat diakibatkan oleh konfigurasi model yang belum maksimal ataupun variabel bebas yang belum lengkap.

DAFTAR PUSTAKA

- BMKG. 2015. *Data Online Pusat Database-BMKG*. Diakses pada 11 Oktober 2021, dari http://dataonline.bmkg.go.id/akses_data
- Hardesty, L. 2017. *Explained: Neural Networks Ballyhooed Artificial-Intelligence Technique Known as "Deep Learning" Revives 70-Year-Old Idea*. Diakses pada 21 Januari 2022, dari <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>
- Khumaidi, A., Raafi'udin, R., & Solihin, I. (2020). *Pengujian Algoritma Long Short-Term Memory untuk Prediksi Kualitas Udara dan Suhu Kota Bandung*. *Jurnal Telematika*, 15(1), 13-18. Retrieved from <https://journal.ithb.ac.id/telematika/article/view/340>.
- Larasati, K. D. and Primandari, A. H. 2021. *Forecasting Bitcoin Price Based on Blockchain Information Using Long-Short Term Method*, *Parameter: Journal of Statistics*, 1(1), pp. 1-6. doi: 10.22487/27765660.2021.v1.i1.15389.
- Olah, C. 2015. *Understanding LSTM Networks*. Diakses pada 20 November 2021, dari <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- Purnomo, H., Suyono, H., and Hasanah, R. N., 2021. *Peramalan Beban Jangka Pendek Sistem Kelistrikan Kota Batu Menggunakan Deep Learning Long Short-Term Memory*. *Transmisi: Jurnal Ilmiah Teknik Elektro*, [Online] Volume 23(3), pp. 97-102. <https://doi.org/10.14710/transmisi.23.3.97-102>
- Rahim dkk. 2016. *Karakteristik Data Temperatur Udara dan Kenyamanan Termal di Makasar*. *Temu Ilmiah*, 075-078
- Sahishanu. 2021. *LSTM – Derivation of Back propagation through time*. Diakses pada 5 Januari 2021, dari <https://www.geeksforgeeks.org/lstm-derivation-of-back-propagation-through-time/>.
- Supari. 2017. *Observed changes in exstream temperature and precipitation over indonesia* *International Journal of Climatology*. 37 (4), 1979–1997.
- Sutanto, T. 2021. *Pengenalan Long Short-Term Memory (LSTM)*. Diakses pada 10 Februari 2022, dari <https://tau-data.id/lstm/>.
- Tjasyono, Bayong. 2004. *Klimatologi*. ITB.
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P. and Gao, R.X. 2019. *Deep Learning and Its Applications to Machine Health Monitoring*. *Mechanical Systems and Signal Processing*, Vol. 115, doi: 10.1016/j.ymsp.2018.05.050.