Prediksi Kunjungan Wisatawan Taman Nasional Gunung Merbabu dengan *Time Series Forecasting* dan *LSTM*

Josua Manullang[1], Albertus J. Santoso[2], Andi W. R. Emanuel[3*]
[1,2,3] Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta
[1]josua.manullang@outlook.com, [2]joko.santoso@uajy.ac.id, [3*]andi.emanuel@uajy.ac.id

Abstract. *Prediction of tourist visits of Mount Merbabu National Park (TNGMb) needs to be done to control the number of visitors and to preserve the national park. The combination of time series forecasting (TSF) and deep learning methods has become a new alternative for prediction. This case study was conducted to implement several methods combination of TSF and Long-Short Term Memory (LSTM) to predict the visits. In this case study, there are 18 modelling scenarios as research objects to determine the best model by utilizing tourist visits data from 2013 to 2018. The results show that the model applying the lag time method can improve the model's ability to capture patterns on time series data. The error value is measured using the root mean square error (RMSE), with the smallest value of 3.7 in the LSTM architecture, using seven lags as a feature and one lag as a label.*
Keywords: *Tourist Visit, Taman Nasional Gunung Merbabu, Prediction, Recurrent Neural Network, Long-Short Term Memory*

Abstrak. *Prediksi kunjungan wisatawan Taman Nasional Gunung Merbabu (TNGMb) perlu dilakukan untul pengendalian jumlah pengunjung dan menjaga kelestarian taman nasional. Gabungan metode antara time series forecasting (TSF) dan deep learning telah menjadi alternatif baru untuk melakukan prediksi. Studi kasus ini dilakukan untuk mengimplementasi gabungan dari beberapa macam metode antara TSF dan Long-Short Term Memory (LSTM) untuk memprediksi kunjungan pada TNGMb. Pada studi kasus ini, terdapat 18 skenario pemodelan sebagai objek penelitian untuk menentukan model terbaik, dengan memanfaatkan data jumlah kunjungan wisatawan di TNGMb mulai dari tahun 2013 sampai dengan tahun 2018. Hasil prediksi menunjukkan pemodelan dengan menerapkan metode lag time dapat meningkatkan kemampuan model untuk menangkap pola pada data deret waktu. Besar nilai kesalahan diukur menggunakan root mean square error (RMSE), dengan nilai terkecil sebesar 3,7 pada arsitektur LSTM, menggunakan tujuh lag sebagai feature dan satu lag sebagai label.*
Kata Kunci: *Kunjungan Wisatawan, Taman Nasional Gunung Merbabu, Prediksi, Recurrent Neural Network, Long-Short Term Memory*

1. Pendahuluan

Industri pariwisata di Indonesia memiliki kontribusi yang besar dalam mendatangkan wisatawan baik nusantara maupun mancanegara, dan salah satunya adalah pariwisata alam. Pariwisata alam yang cukup terkenal di Indonesia adalah kawasan konservasi TNGMb [1]. TNGMb menyediakan jasa pariwisata dalam bentuk ekowisata, didalamnya terdapat kegiatan aktivitas luar ruangan, seperti *hiking* ataupun *trekking* [2] melalui lima jalur pendakian resmi jika ingin sampai di puncak TNGMb dengan ketinggian 3142 meter diatas permukaan laut. Dalam pengelolaanya, TNGMb dapat di katakan berhasil untuk menarik minat wisata para wisatawan. Namun, meningkatnya jumlah kunjungan para wisatawan ke TNGMb juga menjadi tantangan bagi pihak pengelola taman nasional tersebut. Jika jumlah wisatawan tidak dikendalikan dengan baik akan muncul beberapa masalah seperti meningkatnya resiko kecelakaan bagi wisatawan, jumlah wisatawan yang melebihi batas dan optimalisasi pada kawasan konservasi sebagai potensi objek wisata menjadi terganggu.

Pihak pengelola TNGMb mampu menghasilkan data yaitu jumlah kunjungan wisatawan bulanan [3], yang dapat di analisis menggunakan beberapa metode dari *time series forecasting* dan menerapkannya sebagai metode tambahan dalam pemodelan. Data jumlah wisatawan yang berkunjung ke TNGMb dapat dikategorikan data *time series* dan dapat di proses secara sekuensial dengan menggunakan *Recurrent Neural Network* (RNN). Penerapan sistem komputasi dengan menggabungkan metode analisis *time series* pada data dan metode pemodelan RNN untuk memprediksi dapat meningkatkan peforma model dan akurasi hasil prediksi yang lebih baik.

2. Tinjauan Pustaka

Time Series Forecasting merupakan metode untuk bisa mendapatkan nilai di masa mendatang dari suatu deret waktu berdasarkan data di masa lampau. *Time series forecasting* telah dipergunakan dalam beberapa kasus penggunaan seperti dalam kondisi liburan [4], konsumsi energi untuk bangunan [5], prediksi finansial [6], dan produksi minyak bumi [7]. Penelitian ini bermanfaat dalam membantu pengelola TNGMb dalam memprediksi jumlah kunjungan wisatawan di Gunung Merbabu dengan lebih baik seiring dengan tren peningkatan jumlah kunjuangan wisatawan di taman nasional tersebut. Dengan prediksi yang lebih baik, maka antisipasi dan layanan dari pengelola taman nasional tersebut dapat menjadi lebih baik dengan tetap mengedepankan konservasi lingkungan.

3. Metode Penelitian

Data yang berhasil dikumpulkan akan melalui proses analisis dan olah data *resampling* untuk menjadi *data sample* pada proses pelatihan dan proses pengujian. Pada tahap pemodelan berguna untuk menentukan skenario dari variasi antara data dan model ke bentuk pengkodean sebagai objek bahan penelitian studi kasus ini. Kemudian model yang telah di bangun akan melalui proses pelatihan dan proses pengujian. Hasil implementasi model akan digunakan pada tahap evaluasi.

3.1. *Pre-processing* Data Deret Waktu

Dasar untuk setiap analisis deret waktu (*time series*) adalah data deret waktu yang stasioner. Deret waktu yang di pengaruhi tren atau musiman, merupakan non-stasioner. Untuk mendapatkan kondisi stasioner dapat menerapkan metode transformasi data deret waktu yaitu *differencing* seperti yang ditunjukkan pada Persamaan 1 berikut ini.

$$\Delta y_t = y_t - y_{t-1} \tag{1}$$

Data deret waktu yang non-stasioner akan melalui tranformasi *differencing* dengan menghitung perubahan nilai dari waktu $t-1$ hingga ke waktu t dari seri waktu y_t [8]. Salah satu jenis pengujian dapat menggunakan *Augmented-Dickey Fuller* (ADF) *test*. Normalisasi atau *normalization* adalah metode untuk mengubah ukuran data dari rentang asli sehingga semua nilai berada dalam kisaran pada rentang tertentu. Pada penelitian ini akan dilakukan normalisasi dengan nilai interval - 1 dan 1 [9] seperti yang ditunjukkan pada Persaman 2.

$$n_i = \frac{2(x_i - x_{min})}{x_{max} - x_{min}} - 1 \tag{2}$$

Dengan x_{min} sebagai nilai minimum, x_{max} sebagai nilai maksimum, x_i data pada indeks ke-i.

3.2. *Backpropagation Through Time*

Backpropagation Through Time (BPTT) merupakan metode pelatihan spesifik untuk RNN, yang merupakan pengembangan dari metode pelatihan *backpropagation*. Tujuan BPTT adalah untuk menghitung turunan parsial dari *loss* untuk setiap bobot sinaptik yang terhubung, yang dikenal sebagai *gradients*. RNN memiliki $[w_{xh}, w_{hh}, w_{hy}, b_h, b_y]$ dimana dapat di notasikan

θ sebagai rangkaian *Learnable parameter* atau parameter yang akan dipelajari. *Learnable parameter* juga merupakan elemen pada RNN yang akan di modifikasi [10].

3.3. Arsitektur LSTM

RNN pada dasarnya sulit untuk di latih, terutama pada masalah dengan ketergantungan jangka panjang (*long-range dependencies*). Alasan mengapa hal ini terjadi adalah sulit untuk menangkap dependensi jangka panjang karena gradien multiplikasi yang dapat menurun atau meningkat secara eksponensial yang berhubungan dengan jumlah lapisan. Hal ini dikenal dengan *the butterfly-effect* [10], apabila gradien untuk w_{hh} lebih besar dari satu, nilai baru yang di gunakan akan sangat besar (*exploding gradient*). Apabila gradien untuk w_{hh} lebih kecil dari satu, nilai baru yang di gunakan akan sangat kecil (*vanishing gradient*) [11]. *Long Short Term Memory* (LSTM) adalah salah satu pengembangan dari arsitektur RNN yang juga mengatasi kekurangan pada RNN [11].

4. Hasil dan Diskusi

Implementasi penelitian ini akan dilakukan dengan menggunakan bahasa pemograman *python*. *File script* dari kode program dan eksperimen dilakukan pada *Jupyter Notebook*. *Package library* utama untuk data akan dianalisa dan diproses menggunakan *Pandas* dan *Numpy*. Untuk proses komputasi dan dan pemodelan akan memanfaatkan *package library* utama yaitu *Keras* dengan *back-end Tensorflow*.

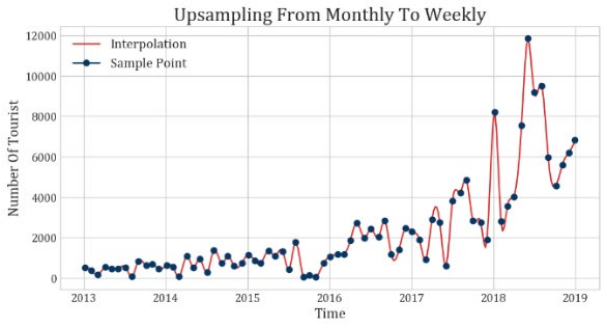
4.1. *Dataset*

Data yang telah dikumpulkan dan diperoleh merupakan data langsung yang dipublikasikan dari pihak Balai TNGMb [3]. Dataset memiliki observasi selama enam tahun, berupa data bulanan seperti yang ditunjukkan pada Gambar 1. Data ini akan di gunakan sebagai *dataset* untuk *training sample* dan *testing sample* pada model. *Sample* akan dipisah dengan menggunakan 80% sebagai *training* dan 20% sebagai *testing*.

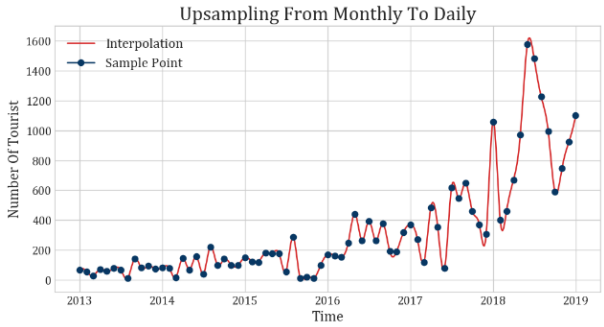
Gambar 1. Grafik jumlah wisatawan tahun 2013 - 2018

4.2. *Upsampling* Data Deret Waktu

Karena data bulanan selama enam tahun masih dianggap terlalu sedikit untuk pengelolahan data, maka pada proses ini terdapat dua *dataset* baru hasil *resampling* yang akan dihasilkan [8], berupa *dataset* mingguan dan *dataset* harian yang ditunjukkan pada Gambar 2 dan Gambar 3. Selanjutnya perlu di lakukan penentuan spasi antar tanggal, untuk data *point* mingguan jarak antar indeks waktu adalah 6 hari. Untuk data *point* harian jarak antar waktu dapat berbeda-beda (27, 28, 29, dan 30 hari). Dilanjutkan dengan melakukan dekomposisi, yang berguna untuk menentukan nilai pada data *point* dan interval nilai kosong (`NaN`).



Gambar 2. Visualisasi *upsampling* ke mingguan



Gambar 3. Visualisasi *upsampling* ke harian

4.3. Tranformasi Data deret Waktu

Pada ketiga *dataset* terdapat sifat non-stasioner, secara visual, terdapat komponen tren yaitu jumlah kunjungan wisatawan meningkat tiap tahunnya seperti yang ditunjukkan pada Gambar 4. Untuk membuat data deret waktu menjadi stasioner akan diterapkan Persamaan 1 pada setiap *data point*. Hasil ADF *test* dapat dijadikan landasan bahwa data deret waktu sudah menunjukkan sifat stasioner dengan menggunakan parameter *orde* pertama dengan satu *lag.*

Gambar 4. Tren peningkatan kunjungan wisatawan dengan rata-rata per 12 bulan

4.4. Pemodelan

Tahap ini akan di inisialisasi dengan menentukan jumlah model, agar dapat dibuat sebuah eksperimen, pada tahap sebelumnya terdapat tiga *dataset* yang telah di hasilkan. Terdapat tiga variasi bentuk *input* dan *output* dengan menerapakan metode *lag time*. Terdapat dua model arsitektur RNN yang akan digunakan yaitu arsitektur RNN dan arsitektur LSTM. Sehingga perancangan akan menghasilkan 18 skenario dengan komposisi pengaturan yang bervariasi, pada Tabel 1.

4.5. Pelatihan

Pada tahap ini akan di terapkan salah satu variasi metode pelatihan pada RNN yaitu *online learning* atau *real time recurrent learning*. *Online learning* merupakan salah satu pengembangan dari BPTT yang bertujuan untuk belajar secara langsung pada setiap *time step*, untuk setiap *time step* bobot dan bias akan di perbarui dan hasil pembaruan akan menjadi inisialisasi bobot dan bias untuk *time step* berikutnya [10]. Setelah melalui beberapa percobaan proses pelatihan, maka proses pelatihan yang optimal ditemukan pada 3000 epoch. Pengaturan *default* pada setiap model ditunjukkan pada Tabel 2 dengan deskripsi hasil implementasi proses pelatihan ditunjukkan pada Tabel 3.

Tabel 1. Deskripsi skenario pada setiap pemodelan

Skenario	Dataset	Arsitekur	Lag Time	Feature	Label	Training	Testing
1	Bulanan	RNN	1	1	1	59	12
2	Bulanan	RNN	4	3	1	56	12
3	Bulanan	RNN	3	3	3	56	10
4	Mingguan	RNN	1	1	1	260	52
5	Mingguan	RNN	5	4	1	256	52
6	Mingguan	RNN	4	4	4	256	49
7	Harian	RNN	1	1	1	1825	365
8	Harian	RNN	8	7	1	1819	365
9	Harian	RNN	7	7	1	1819	358
10	Bulanan	LSTM	1	1	1	59	12
11	Bulanan	LSTM	4	3	1	56	12
12	Bulanan	LSTM	3	3	3	56	10
13	Mingguan	LSTM	1	1	1	260	52
14	Mingguan	LSTM	5	4	1	256	52
15	Mingguan	LSTM	4	4	4	256	49
16	Harian	LSTM	1	1	1	1825	365
17	Harian	LSTM	8	7	1	1819	365
18	Harian	LSTM	7	7	1	1819	358

Tabel 2. Deskripsi pengaturan *default* pada setiap model

Input Shape	Output Shape	Epoch	Batch Size	Optimizer	Loss	Layer	Monitor	Mode	Patience
[1,1,feature]	[1,label]	3000	1	Adam	MSE	3	Val_loss	auto	2

Tabel 3. Deskripsi hasil implementasi proses pelatihan pada model

Skenario	Unit In Layer	Epoch	Waktu
1	1–1–1	3000	30 menit
2	1–3–1	3000	28 menit
3	1–3–3	3000	28 menit
4	1–1–1	1734	74 menit
5	1–4–1	74	3 menit
6	1–4–4	2380	104 menit
7	1–1–1	3	1 menit
8	1–7–1	4	1 menit
9	1–7–7	5	2 menit
10	1–1–1	3000	43 menit
11	1–3–1	3000	46 menit
12	1–3–3	3000	44 menit
13	1–1–1	3000	155 menit
14	1–4–1	337	19 menit
15	1–4–4	1930	98 menit
16	1–1–1	20	8 menit
17	1–7–1	16	7 menit
18	1–7–7	10	4 menit

***** Baris biru merupakan aktifnya fungsi *early stopping* saat pelatihan

Untuk analisis berdasarkan ukuran data pada dan jumlah observasi pada *training sample*, menunjukkan semakin besar ukuran sampel semakin besar juga pengaruhnya pada proses pembelajaran. Jika di bandingkan, LSTM mampu menyelesaikan pelatihan untuk jumlah *epoch* yang banyak dengan jumlah *sample* yang banyak pula.

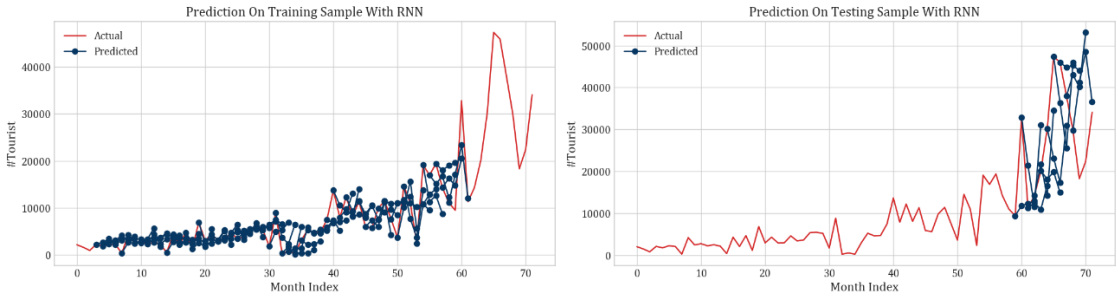
4.6. Pengujian

Proses pengujian di mulai dengan memberikan satu baris observasi untuk satu langkah waktu dari seluruh baris observasi yang ada. Sehingga model akan membuat prediksi untuk setiap satu baris observasi yang di berikan. Proses ini meniru skenario dunia nyata di mana observasi baru untuk saat ini akan di gunakan dalam prediksi untuk waktu berikutnya. Variasi pelatihan ini dapat mempercepat proses belajar pada model dan juga secara langsung menerapakan prediksi yang dikenal dengan *walk-forward validation.* Setelah model menghasilkan nilai prediksi, hasil prediksi akan di buat untuk kembali ke *range* data asli agar dapat di lihat dan di bandingkan secara nyata antara nilai hasil prediksi dan nilai aktual yang diharapkan. Terdapat dua fungsi yang diterapkan untuk mengembalikan nilai pada bentuk aslinya yaitu transformasi pengembalian *differencing* (*inverting*) dan tranformasi pengembalian skala data (*denormalization*) [9], yaitu seperti pada Persamaan 3 dan 4 berikut:

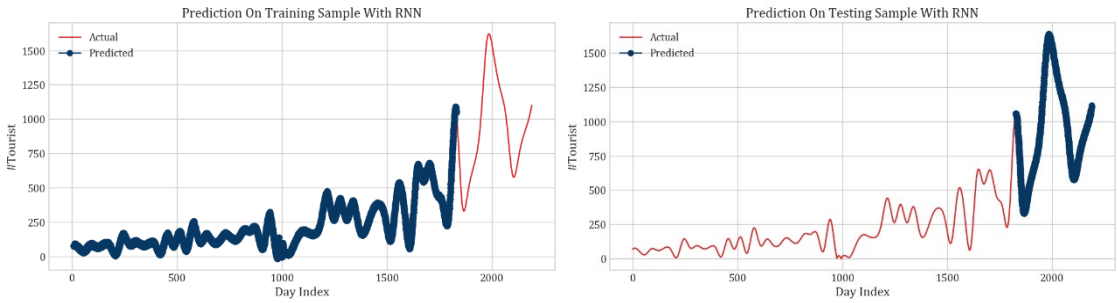
$$dn_i = \left((n_i + 1)(x_{max} - x_{min})\right) + \left(\frac{2(x_{min})}{2}\right) \quad (3)$$

$$y_t = \Delta y_t + y_{t-1} \quad (4)$$

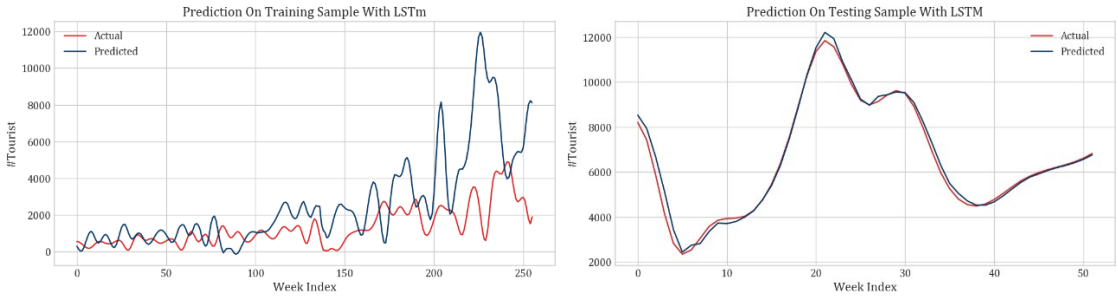
Pada Persamaan 3 akan dilakukan penjumlahan pengamatan hasil prediksi y pada waktu $t-1$ dengan nilai Δy pada waktu t, dimana Δy adalah nilai hasil *differencing,* pada Persamaan (1). Pada Persamaan 4 diterapkan untuk mengembalikan skala data pada hasil prediksi ke *range* yang sebanding dengan skala data aktual dengan n_i merupakan nilai hasil hasil normalisasi pada Persamaan 2. Nilai hasil prediksi akan di bandingkan satu sama lain berdasarkan arsitektur RNN dan arsitektur LSTM pada model. Hasil prediksi pada *training sample* dan pada *testing sample* pada beberapa skenario ditunjukkan pada Gambar 5, 6, 7, dan 8.



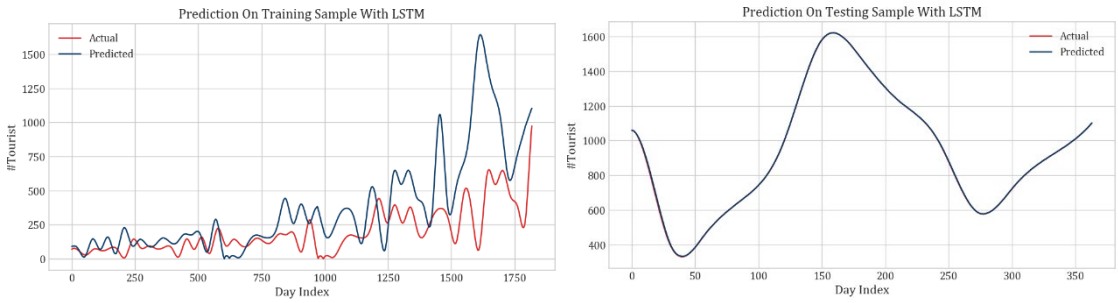
Gambar 5. Hasil prediksi pada *training sample* dan *testing sample* skenario 3



Gambar 6. Hasil prediksi pada *training sample* dan *testing sample* skenario 9



Gambar 7. Hasil prediksi pada *training sample* dan *testing* sample skenario 14



Gambar 8. Hasil prediksi pada *training sample* dan *testing sample* skenario 17

4.7. Evaluasi

Setelah proses transformasi pengembalian *range* nilai, pada setiap *sample*, hasil prediksi akan di kumpulkan dan di hitung skor kesalahan dengan menggunakan *root mean square erorr* (RMSE) [7], yaitu seperti pada Persamaan 5 berikut:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - y_i)^2} \quad (5)$$

Dimana n adalah jumlah data, a_i adalah hasil yang diharapkan dan y_i adalah hasil prediksi dari masing-masing indeks atau baris.

Tabel 4. Deskripsi perhitungan *error* menggunakan *training sample*

Skenario	RMSE Training Sample						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	9590,7						
2	12327,8						
3	3265,9	3479,8	3861,4				
4	2216,8						
5	2447,9						
6	66,6	182,9	351,4	557,9			
7	315,7						
8	324,6						
9	1,9	4,9	6,9	9,8	13,3	16,9	19,9
10	9647,9						
11	12421,7						
12	2996,1	3117,9	3664,9				
13	2216,4						
14	2455,9						
15	61,7	180,8	349,1	543,6			
16	315,9						
17	324,7						
18	2,41	4,8	7,16	9,4	11,8	14,1	16,2

Tabel 5. Deskripsi perhitungan *error* menggunakan *testing sample*

Skenario	RMSE Testing Sample						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	13769,9						
2	13987,8						
3	14549,7	18646,1	18310,3				
4	406,3						
5	310,3						
6	262,8	643,8	1119,6	1647,6			
7	4,4						
8	4,2						
9	2,4	5,9	8,4	11,8	15,2	19,2	22,9
10	13842,8						
11	16861,9						
12	15697,9	15400,0	17634,6				
13	408,7						
14	262,3						
15	170,7	435,4	858,6	1400,6			
16	4,4						
17	3,7						
18	5,8	11,6	17,5	23,4	29,02	33,9	39,6

*Baris berwarna biru adalah nilai RMSE *Testing* < RMSE *Training*

Pada Tabel 4, untuk proses prediksi lebih dari satu keluaran (*multi label*), nilai RMSE pada *testing sample* akan lebih besar dibandingkan dengan *training sample*. Juga untuk setiap langkah waktu nilai RMSE akan bertambah. Sehingga model lebih banyak mengingat di bandingkan mempelajari pola pada data. Saat model memprediksi untuk lebih dari satu *output,* maka kemungkinan nilai *error* akan bertambah untuk setiap satu langkah waktu kedepan pada hasil prediksi, ini menandakan penggunaan lebih dari satu *unit* pada *output layer* tidak terlau efektif untuk menurunkan nilai RMSE. Sehingga menambah lebih dari satu *unit* pada *output layer* dapat menyebabkan model mengalami *overfitting* dan tidak menunjukkan hasil yang optimal.

Sedangkan pada Tabel 5, jika menggunakan *dataset* bulanan hasil prediksi menunjukkan nilai RMSE pada *training sample* lebih kecil dari pada nilai RMSE pada *testing sample.* Hal ini menunjukkan bahwa model mengalami *overfitting* saat melakukan prediksi sehingga model memerlukan data observasi yang lebih banyak lagi pada *dataset*. Serta, pada seluruh model yang menggunakan *dataset* bulanan menujukkan nilai RMSE yang lebih besar di bandingkan dengan *dataset* mingguan dan *dataset* harian.

Evaluasi hasil proses prediksi juga menunjukkan bahwa model dapat memprediksi namun belum sepenuhnya akurat karena nilai RMSE masih di anggap cukup besar (tidak lebih kecil dari nol). Penerapan metode *lag time* juga memberikan pengaruh pada model, untuk mempelajari pola pada data deret waktu, untuk itu jumlah *feature* lebih dari satu dapat memberi peluang pada model untuk menangkap pola pada data deret waktu. Pengaturan lebih dari satu *unit* pada *hidden layer* juga memiliki pengaruh, dapat meningkatkan akurasi dan memperkecil kesalahan. Sehingga model dapat meningkatkan kapasitas dan kemampuan untuk belajar, namun untuk *outpur layer* jika *unit* lebih dari satu dapat menambah peluang pada model menenemui permasalahan *overfiting* dan meningkatkan nilai *error*.

5. Kesimpulan dan Saran

Pada penelitian studi kasus ini telah dilakukan prosedur pemodelan dan mengidentikasikasi performansi pada model untuk memprediksi. Dengan membentuk skenario berdasarkan variasi antara *dataset* dan arsitektur pada masing-masing model. Dataset kunjungan ke TNGMb dari 2013 sampai 2018 dapat di-*upsampling* menjadi data mingguan dan bulanan. Data mingguan dan bulanan tersebut dapat dijadikan dataset untuk membuat model prediksi kunjungan wisatawan di masa mendatang.

Proses pelatihan model dilakukan dengan menggunakan arsitektur LSTM, yang memiliki pola pembelajaran lebih stabil dibandingkan dengan arsitektur RNN. Hasil prediksi menunjukkan

proses prediksi pada skenario 4, 5, 7, 8, 13, 14, 16, dan 17 menghasilkan nilai prediksi yang lebih baik dan tidak menemui *overfitting.* Nilai RMSE yang terkecil yang didapatkan yaitu 3,7 pada model LSTM dan 4.2 pada model RNN menggunakan *dataset* harian. Menambahkan lebih dari satu *unit* pada *hidden layer* dapat meningkatkan kemampuan belajar pada model. Model juga memiliki potensi menemui *overfitting* jika menambahkan lebih dari satu *unit* pada *output layer*. Penerapan metode *lag time* pada *sample* data dapat meningkatkan kemampuan pada model, dengan *lag time* terbaik yaitu tujuh *lag* pada *feature* dan satu *lag* pada *label*.

Saran yang dapat di berikan untuk pengembangan lebih lanjut membuat analisis *multivariate time series*, dengan menambah jumlah variabel terhadap deret waktu yang lebih banyak lagi.

Referensi

[1] Direktorat Jendral KSDAE, "Laporan kinerja 2018 direktorat jenderal konservasi sumber daya alam dan ekosistem kementrian lingkungan hidup dan kehutanan," Kemen. Lingkungan Hidup dan Kehutanan, Jakarta, Indonesia, Dec. 2018.

[2] Z. Pololikashvili, Walking Tourism Promoting Regional Development, Madrid, Spain: World Tourism Organization (WTO), 2019, doi: https://doi.org/10.18111/9789284420346

[3] B. TNGMb, "Statistika Balai Taman Nasional Gunung Merbabu tahun 2013 - 2018," B. TNGMb, Magelang, Indonesia, Jul. 2019.

[4] N. Laptev, J. Yosinski, L. E. Li and S. Smyl, "Time-series extreme event forecasting with Neural Networks at Uber," in *ICML 2017 Time Series Workshop*, Sydney, Australia, 2017.

[5] C. Deb, F. Zhang, J. Yang, S. E. Lee and K. W. Shah, "A review on time series forecasting techniques for building energy consumption," *Renewable and Sustainable Energy Reviews,* vol. 74, pp. 902-924, Jul. 2017.

[6] J. Cao, Z. Li and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," *Physica A: Statistical Mechanical and its Applications,* vol. 519, pp. 127 - 139, Apr. 2019.

[7] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing,* vol. 323, pp. 203-213, Jan. 2019.

[8] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles And Practice, Melbourne, Australia: Otexts, 2018.

[9] A. A. Rizal and S. Soraya, "Multi Time Steps Prediction dengan Recurrent Neural Network long Short Term Memory," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.,* vol. 18, no. 1, pp. 115 - 124, Nov. 2018.

[10] I. Sutskever, Training Recurrent Neural Networks, Toronto, Kanada: University of Toronto, 2013.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.,* vol. 9, no. 8, pp. 1735-1780, Dec. 1997.