

# PREDIKSI PARAMETER CUACA MENGGUNAKAN DEEP LEARNING LONG-SHORT TERM MEMORY (LSTM)

## WEATHER PARAMETERS PREDICTION USING DEEP LEARNING LONG-SHORT TERM MEMORY (LSTM)

Eko Supriyadi

Pusat Meteorologi Maritim, BMKG, Jl. Angkasa No2. Kemayoran, Jakarta, 10720

\*E-mail: eko.supriyadi@bmkg.go.id

Naskah masuk: 20 Mei 2019      Naskah diperbaiki: 29 September 2019      Naskah diterima: 16 Oktober 2019

### ABSTRAK

Saat ini metode *deep learning* dapat diaplikasikan untuk memprediksi suatu kejadian, seperti memprediksi cuaca suatu wilayah. Salah satu contoh *deep learning* yang cocok digunakan pada jenis data *time series* adalah LSTM. Penelitian ini menerapkan metode *deep learning* LSTM dengan jumlah layer 200, perbandingan data *training* dengan data *test* sebesar 9:1, serta mengukur nilai RMSE dan RMSE *update* hasil validasi dan prediksi beberapa hari ke depan. Data yang digunakan terdiri dari pengukuran suhu udara, kelembaban udara, kecepatan angin, dan tekanan udara selama bulan Januari dan Februari 2019. Data bulan Januari digunakan sebagai data *training* dan *test* untuk melakukan validasi prakiraan, sedangkan data bulan Februari digunakan sebagai pembandingan dari hasil prediksi *deep learning* LSTM. Hasil penelitian menunjukkan RMSE seluruh validasi parameter cuaca nilainya semakin baik ketika menggunakan LSTM dengan *update*. Diperoleh RMSE *update* untuk parameter suhu, kelembaban, kecepatan angin, dan tekanan udara masing-masing bernilai 0,576; 2,8687; 2,1963; dan 1,0647. Sedangkan prediksi suhu udara, kelembaban, kecepatan angin, dan tekanan udara untuk 1 hari ke depan (1 Februari 2019) masing-masing sebesar 1,0337; 6,3413; 2,8934; dan 1,4313. Dari parameter tersebut hanya parameter suhu dan kelembaban udara yang mengalami pertambahan RMSE seiring bertambahnya waktu. Sedangkan parameter kecepatan angin dan tekanan udara mengalami penurunan di hari ketiga dan meningkat secara kontinu hingga satu bulan ke depan.

**Kata kunci:** prediksi, parameter cuaca, *deep learning*, LSTM

### ABSTRACT

Today, *deep learning* can be applied to predict any events, such as predict the weather of a region. One of them is LSTM which is suitable for use in time series data types. This study conducted the *deep learning* LSTM with the number of 200 layers, ratio training with test data of 9:1, measuring the value of validation RMSE and RMSE *update* and also predictions some weather parameters in a few days later. The data used consisted of measurements of air temperature, humidity, wind speed, and air pressure during January and February 2019. The data January were used as training and test data to conduct forecast validation, while the data February was used as a comparison of the results predicted for *deep learning* LSTM. The result shows that the forecast RMSE for all-weather parameters is better when using LSTM with an *update*. Obtained for temperature, relative humidity, wind speed, and air pressure have RMSE with *update* are 0,576; 2,8687; 2,1963; and 1,0647, respectively. While the prediction of air temperature, wind speed, and air pressure for one day later (1 February 2019) is 1.0337; 6.3413; 2,8934; and 1.4313, respectively. From the all-weather parameters only temperature and humidity parameter that increase in RMSE over time. While the parameters such as wind speed and air pressure decreased on the third day and increased continuously for the next one month.

**Keywords:** prediction, weather parameters, *deep learning*, LSTM

## 1. Pendahuluan

Semakin hari minat dan kebutuhan masyarakat Indonesia terhadap kebutuhan informasi cuaca dan iklim semakin tinggi. Hal ini dapat dibuktikan dengan meningkatnya jumlah *follower* akun twitter #infoBMKG (Badan Meteorologi, Klimatologi, dan Geofisika) yang mencapai 4,03 juta (per Maret 2019),

laporan Survey Kepuasan Masyarakat (SKM 2018) BMKG yang mengalami kenaikan dari 3,38 di tahun 2017 menjadi 3,41 di tahun 2018 [1], dan terakhir dukungan pemerintah untuk BMKG melalui alokasi APBN 2018 mencapai 1,7 T, naik 7% dibandingkan tahun sebelumnya [2].

Agar layanan masyarakat tentang informasi cuaca dan iklim tetap berjalan tentunya dibutuhkan prakiraan cuaca yang tepat. Namun hingga saat ini proses pembuatan prakiraan cuaca dan iklim masih terbatas dari masukan data pemodelan. Di mana dalam pemodelan tersebut masih memiliki kekurangan seperti penggunaan jumlah parameter, asumsi-asumsi matematis, dan formulasi persamaan yang cenderung rumit. Alasan tersebut dapat dipahami sebab untuk menghasilkan hasil model yang mendekati kenyataan dibutuhkan banyak masukan parameter dan pemenuhan asumsi yang terkadang sangat sulit untuk dilakukan.

Mengatasi hal tersebut maka dikembangkan kemampuan belajar, menganalisa, dan menarik kesimpulan pada komputer yang dikenal sebagai *deep learning* dengan prinsip kerjanya mirip jaringan syaraf tiruan yang selama ini kita kenal. Bedanya kalau jaringan syaraf tiruan hanya mengandalkan 1 *layer*, sedangkan *deep learning* lebih dari 1 *layer* [3]. Ada berbagai definisi yang berbeda mengenai batasan “*deep*” itu sendiri. Hinton [4] menjelaskan *deep* itu sendiri sebagai jaringan syaraf yang memiliki tiga *layer*, Simonyan [5] dengan batasan 16 *layer*, dan He [6] yang menggolongkan *deep* sebanyak 1000 *layer*. Semakin tinggi *layer*, *learning* komputer makin lama.

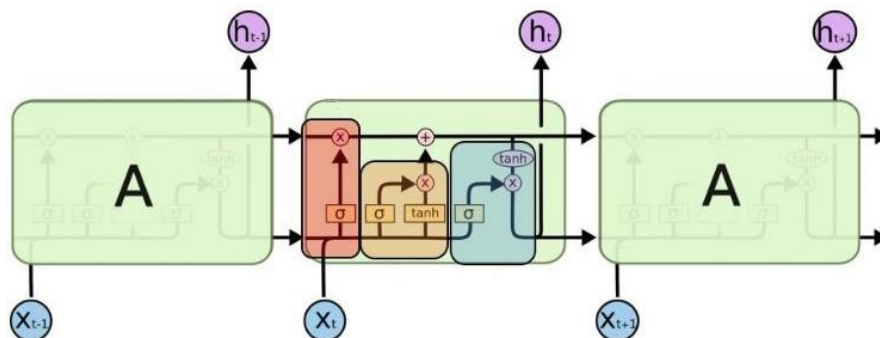
*Deep Learning* sendiri merupakan bagian dari *machine learning* yang bertugas mempelajari data yang tersedia melalui algoritma yang ada. Proses pembelajaran ini dapat dilakukan melalui tiga pilihan, yaitu secara terarah, semi terarah atau tak terarah [7,8,9]. Pembelajaran secara terarah mengijinkan algoritma untuk belajar berdasarkan data yang tersedia. Sebaliknya pada pembelajaran semi terarah, algoritma belajar berdasarkan data yang tersedia tetapi tidak mencukupi. Sedangkan pembelajaran tak terarah mengandalkan algoritma untuk belajar tanpa ada masukan data satupun. Di sini, algoritma dituntut untuk menemukan pola suatu masalah yang tersembunyi.

Lebih lanjut, *layer* yang dimaksudkan di atas tadi merupakan kesamaan pada sistem syaraf manusia yang disebut sebagai neuron. *Layer* merupakan fungsi

matematika dalam bentuk vektor yang menghasilkan nilai berupa *gradient descent* [10,11]. Fungsi matematika yang sering digunakan dalam proses *deep learning* adalah fungsi *sigmoid* ( $\sigma$ ). Fungsi *sigmoid* digunakan sebab hasil turunan pertamanya adalah fungsi *sigmoid* dengan selisih satu fungsi *sigmoid* itu sendiri. Di mana bentuk ini jauh lebih membutuhkan perhitungan yang relatif mudah dan cepat. Selain itu fungsi *sigmoid* dapat diinterpretasikan sebagai nilai peluang sebab nilainya di antara 0 dan 1 [12].

Umumnya perhitungan *deep learning* menggunakan *layer* lebih dari satu. Semakin banyak *layer* yang digunakan proses pengolahan semakin lambat. Sehingga *deep learning* sangat cocok bila diolah menggunakan GPU (*Graphic Processing Units*) yang memiliki *core* lebih banyak ketimbang CPU (*Central Processing Unit*) [13]. Contoh penggunaan *deep learning* untuk data *time series* yang banyak dihasilkan dari pengamatan cuaca adalah *Long-Short Term Memory* (LSTM). LSTM sendiri diciptakan oleh Hochreiter dan Schmidhuber pada tahun 1997 [14]. Jaringan ini terdiri dari *layer* LSTM dengan proses berulang. Di dalam satu modul LSTM terdapat beberapa bagian komputasi seperti operasi penambahan, perkalian, penggabungan, duplikasi vektor dan fungsi matematika seperti *tanh* dan *sigmoid* seperti dijelaskan sebelumnya (Gambar 1).

Percobaan *deep learning* LSTM di bidang cuaca dan iklim hingga saat ini sangat minim ditemui. Namun aplikasi lainnya lebih banyak ditemui pada bidang agronomi [15,16,17,18,19], kualitas udara [20,21], perkembangan bahasa [22,23,24,25] bahkan *e-commerce* yang sedang berkembang pesat saat ini [26,27]. Penelitian yang dilakukan oleh Shi [28] menemukan LSTM dapat menghitung jumlah curah hujan secara spasial dan temporal dengan korelasi 0,908. Zaytar [29] menggunakan LSTM untuk memprediksi parameter suhu, kelembaban, dan kecepatan angin pada sembilan kota di Monako. Diperoleh akurasi yang baik dengan nilai MSE (*Mean Squared Error*) berkisar 0,005 – 0,01. Nilai MSE makin dekat nol semakin baik hasil prakiraan.



**Gambar 1.** Diagram LSTM. Perhatikan dalam modul LSTM terdapat 4 proses perhitungan interaktif (ikuti alur panah hitam kotak tengah). Gambar diadaptasi dari Wiest [30].

Selain metode LSTM, metode *deep learning* lainnya adalah CNN (*Convolutional Neural Networks*) dan RNN (*Recurrent Neural Networks*). Perbandingan ketiga metode tersebut disajikan Tabel 1. Metode CNN umumnya sering digunakan dalam pengenalan wajah dan klasifikasi objek. CNN mempunyai 4 *layer* yang terdiri dari *Convolution layer*, *ReLu layer*, *pooling* dan *fully connected layer* [31]. Penelitian yang dilakukan oleh Klein [32] menunjukkan CNN dengan metode *patch based dynamic convolution network* memiliki nilai *error* yang lebih rendah dibandingkan *last frame* dalam memprediksi liputan awan hujan dari radar cuaca. Lebih lanjut penelitian yang dilakukan Arrofiqoh [33] menemukan metode CNN memiliki akurasi sebesar 90,12% dalam mengklasifikasikan jenis bangunan dari citra satelit.

Sementara pada RNN bekerja dengan menyimpan *layer* dan mengeluarkannya kembali sebagai masukan dalam memprediksi hasil akhirnya. Perbedaannya dengan CNN adalah pada RNN dapat menghitung barisan data, sedangkan CNN tidak. Ketika memikirkan data cuaca terdapat kesan bahwa data cuaca terkini hingga 24 jam ke depan lebih penting. Namun ketika terdapat kasus data lama harus dimanfaatkan untuk mengenali *trend* dan pergerakan umum beberapa hari ke depan RNN gagal melakukan tugasnya. Masalah ini disebut sebagai ketergantungan jangka panjang-pendek. Mengatasi hal tersebut *deep learning* metode LSTM cocok digunakan sebagai prakiraan cuaca.

Berkaca dari penjelasan sebelumnya maka sangat dimungkinkan penggunaan *deep learning* LSTM untuk mendukung kegiatan prakiraan cuaca. Sebab data pengamatan meteorologi umumnya berbentuk vektor dan *time series*. Oleh sebab itu, penelitian ini akan menjelaskan kegunaan *deep learning* LSTM untuk memperkirakan parameter cuaca seperti suhu, kelembaban, kecepatan angin dan tekanan udara, membandingkan nilai prakiraan dan pengamatan pada tahap validasi dengan melihat nilai RMSE-nya serta menguji kehandalan *deep learning* LSTM dalam memprediksi suhu, kelembaban, kecepatan angin dan tekanan udara beberapa hari kedepan. Diharapkan penggunaan konsep *deep learning* LSTM ini dapat menjadi salah satu jurus andalan BMKG meningkatkan kualitas pelayanan cuaca dan iklim masyarakat Indonesia.

**Tabel 1. Perbandingan karakteristik utama Deep Learning CNN, RNN, dan LSTM.**

	CNN	RNN	LSTM
<b>Penggunaan</b>	<i>image data</i> klasifikasi	<i>time series</i> regresi	<i>time series</i>
<b>Fungsi matematika</b>	-	<i>tanh</i>	<i>tanh</i> dan <i>sigmoid</i>
<b>Proses perhitungan</b>	input saat ini	input saat ini dan lalu	input saat ini dan lalu
<b>Layer</b>	1	≥1	≥1

## 2. Metode Penelitian

**Data.** Data yang digunakan dalam penelitian ini berasal dari pengamatan sinoptik Stasiun Meteorologi Maritim Tanjung Priok. Data tersebut terdiri dari pengukuran suhu udara, kelembaban udara, kecepatan angin, dan tekanan udara pada bulan Januari dan Februari 2019. Data bulan Januari digunakan sebagai data *training* dan *test*, sedangkan data bulan Februari digunakan sebagai pembandingan dari hasil prakiraan *deep learning* LSTM yang dihasilkan dari bulan Januari. Tidak ada kriteria dalam pemilihan waktu sebab uji *deep learning* LSTM berlaku pada semua waktu asalkan datanya berupa *time series*. Keseluruhan data dapat diakses melalui <http://bmkgsoft.database.bmkg.go.id>.

**Metode.** Setelah memperoleh kedua data di atas, langkah selanjutnya melakukan interpolasi pada data bulan Januari 2019 agar tidak terdapat data kosong. *Deep learning* LSTM tidak akan bekerja bila terdapat data kosong. Persamaan yang digunakan menggunakan pendekatan interpolasi linier:

$$f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \quad (1)$$

di mana:

$x$  dan  $f_1(x)$  adalah titik yang ingin dicari melalui interpolasi

$x_0$  dan  $f(x_0)$  adalah titik pertama yang diketahui

$x_1$  dan  $f(x_1)$  adalah titik kedua yang diketahui

Setelah itu data dibagi menjadi dua bagian, yaitu data *training* dan data *test*. Pembagian data menjadi dua ini merupakan inti dari teori *deep learning* LSTM [34]. Pembagiannya sendiri menggunakan rasio 9:1 di mana nilai 1 (satu) untuk *test* data sudah mewakili sifat dari keseluruhan data.

$$\text{data\_train} = 0,9 \cdot \text{data} \quad (2)$$

$$\text{data\_test} = 0,1 \cdot \text{data} \quad (3)$$

Untuk menghasilkan kesesuaian prakiraan yang baik dan mencegah penyimpangan dari pengolahan *training* data, maka dilakukan standarisasi agar memiliki *mean* dan *varians nol*. Persamaan yang digunakan adalah:

$$\sigma_{\text{train\_std}} = \frac{\text{data\_train} - \mu_{\text{train}}}{\sigma_{\text{train}}} \quad (4)$$

di mana:

$\sigma_{\text{train\_std}}$  adalah data *train* yang sudah distandarisasi

$\mu_{\text{train}}$  adalah rata-rata data *train*

$\sigma_{\text{train}}$  adalah standar deviasi data *train*

Untuk memprakirakan nilai selanjutnya pada data *time series*, tentukan yang akan menjadi respon dari

data *train* yang sudah distandarisasi tadi dengan selisih satu data terakhir. Pada langkah ini LSTM mempelajari prediksi nilai selanjutnya. Selanjutnya tentukan juga yang menjadi prediktor dari data *train* yang sudah distandarisasi tetapi tanpa menggunakan data terakhir.

$$x_{\text{train}} = \sigma_{\text{train std}(1)}, \dots, \sigma_{\text{train std}(n-1)} \quad (5)$$

$$y_{\text{train}} = \sigma_{\text{train std}(2)}, \dots, \sigma_{\text{train std}(n)} \quad (6)$$

di mana:

$x_{\text{train}}$  adalah respon

$y_{\text{train}}$  adalah prediktor

$\sigma_{\text{train std}(n)}$ , adalah respon ke- $n$ , dengan  $n = 1, 2, 3, \dots$  dst

Lakukan hal yang sama untuk data *test*:

$$\sigma_{\text{test\_std}} = \frac{\text{data\_test} - \mu_{\text{train}}}{\sigma_{\text{train}}} \quad (7)$$

$$x_{\text{test}} = \sigma_{\text{test std}(1)}, \dots, \sigma_{\text{test std}(n-1)} \quad (8)$$

$$y_{\text{test}} = \sigma_{\text{test std}(2)}, \dots, \sigma_{\text{test std}(n)} \quad (9)$$

Standarisasi data *test* dilakukan dengan menggunakan *mean* dan standar deviasi data *training*. Sampai di sini langkah kita selesai dalam membuat masukan LSTM. Gunakan respon dan prediktor di atas sebagai masukan LSTM. Selanjutnya masuk ke dalam *deep learning* LSTM.

Ada empat hal yang harus dipersiapkan dalam *deep learning*, yaitu menyiapkan *layer*, parameter *training*-nya, latih jaringan LSTM, dan prakiraan data. Hasil akhirnya nanti menghasilkan nilai  $y$

prediksi ( $y_{\text{pred}}$ ). Patut diingat variabel  $y_{\text{pred}}$  masih berupa nilai standarisasi untuk itu perlu di standarisasi balik menggunakan persamaan:

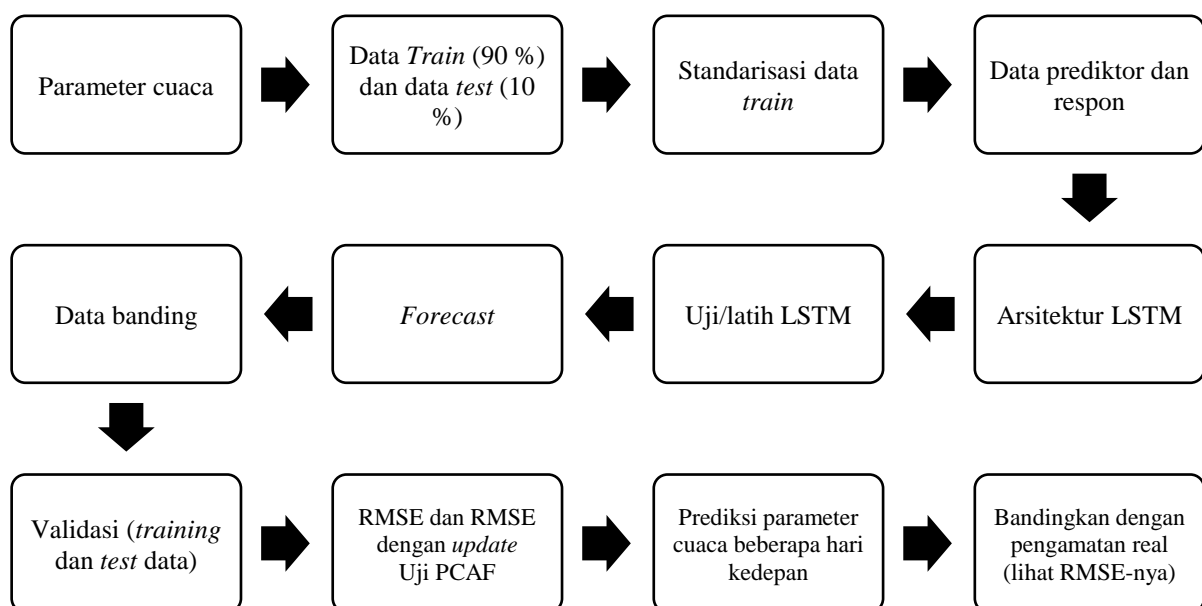
$$y_{\text{pred}} = \sigma_{\text{train}} \cdot y_{\text{pred}} + \mu_{\text{train}} \quad (10)$$

Selanjutnya nilai  $y_{\text{pred}}$  dan  $y_{\text{test}}$  ini akan dibandingkan selisihnya menggunakan RMSE (*Root Mean Square Error*):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_{\text{pred}} - y_{\text{test}})^2}{n}} \quad (11)$$

Persamaan 11 di atas selanjutnya digunakan sebagai prediksi parameter cuaca beberapa hari ke depan yaitu: 1 hari, 3 hari, 1 minggu, dan 1 bulan. Hasilnya akan dibandingkan dengan data bulan Februari 2019 untuk melihat performa RMSE-nya. Untuk memahami metode yang digunakan dalam kajian ini, perhatikan diagram alir pada Gambar 2 di bawah ini.

Terakhir untuk menguji apakah terdapat hubungan antar nilai di dalam data pengamatan atau model itu sendiri digunakan *autocorrelation function* (ACF). Secara tidak langsung ACF akan menguji apakah nilai bergantung pada waktu atau tidak. Bila bergantung terhadap waktu pengamatan maka nilai yang dihasilkan cenderung berfluktuatif atau memiliki *trend*. Namun bila tidak berpengaruh terhadap waktu nilai yang dihasilkan cenderung konstan. Dalam statistika jenis data tersebut masing-masing dikenal sebagai data tidak stasioner dan stasioner [35].



Gambar 2. Alur *deep learning* LSTM dalam prediksi parameter cuaca.

Nilai autokorelasi menggunakan perbandingan kovarian terhadap varians data *time series* dalam perhitungannya [36]:

$$\rho_k = \frac{E[(z_t - \mu)(z_{t+k} - \mu)]}{\sqrt{E[(z_t - \mu)^2] E[(z_{t+k} - \mu)^2]}} \quad (12)$$

dimana:

$\rho_k$  = nilai ACF pada *lag* ke-k, dengan  $k = 0, 1, 2, \dots$

$z_t$  = data ke t, dengan  $t = 1, 2, 3, \dots$

$z_{t+k}$  = data ke t dengan *lag* ke-k

$\mu$  = *mean* data keseluruhan

Terkadang dalam proses ACF kita tidak dapat mengetahui seberapa besar data menyebar dari posisi awal. Sehingga digunakan teknik lanjutan berupa *partial autocorrelation function* (PACF) dengan persamaan [36]:

$$\phi_{kk} = \text{corr}[z_t - \hat{z}_t, z_{t-k} - \hat{z}_{t-k}] \quad (13)$$

Persamaan 13 merupakan pengembangan dari Pers. 12 di mana pada Pers. 13 yang dicari adalah nilai korelasi ACF antar *lag*. Semakin besar nilai *lag*-nya nilai PACF mendekati nol. Selanjutnya karena PACF merupakan korelasi antar nilai pengamatan maka diperlukan selang kepercayaan untuk meyakinkan berapa banyak data tersebut berkorelasi. Sehingga digunakan standar deviasi dengan masukan banyak data ( $n$ ) dan selang kepercayaan (SK 95 %) [36]:

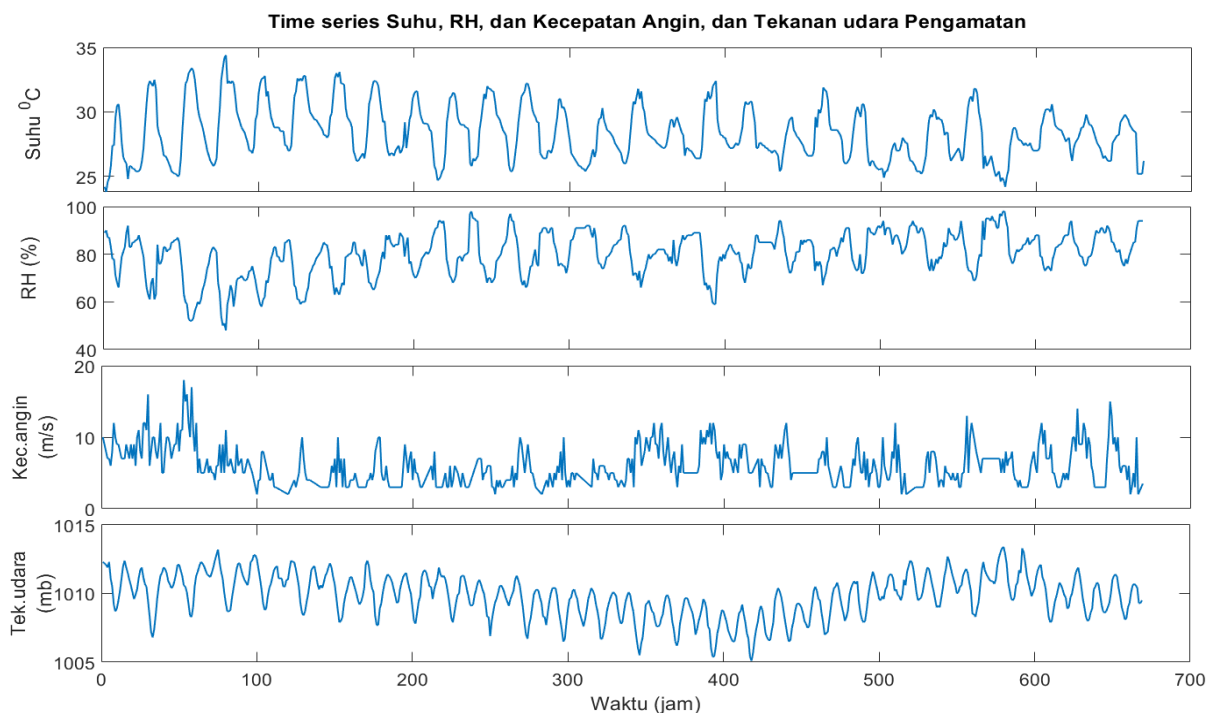
$$SE[\hat{\phi}_{kk}] \cong \pm 1,96 \frac{1}{\sqrt{n}} \quad (14)$$

Nilai ACF dan PACF disajikan dalam grafik *correlogram* [36]. Dalam penelitian dilakukan analisis PACF sebab data jenis *time series* umumnya menghasilkan jenis data tidak stasioner. Hal ini disebabkan pengaruh skala waktu yang berbeda-beda baik skala hari hingga tahun.

### 3. Hasil dan Pembahasan

#### *Training dan testing data deep learning LSTM.*

Sebelum mendapatkan hasil prediksi cuaca dengan *deep learning LSTM*, terlebih dahulu dilakukan proses *training* data di komputer sebagai *machine learning*. *Machine learning* (ML) adalah kumpulan algoritma komputer untuk mengolah data. Pada penggunaannya, ML dapat tersusun dari satu atau lebih algoritma. Selain itu, ML juga mampu bekerja dengan atau tanpa bantuan manusia [37]. *Machine learning* dapat diaplikasikan kepada berbagai jenis komputer, server, *cloud*, dan ponsel. Setiap perangkat tersebut tentunya memiliki kemampuan yang berbeda dalam menyelesaikan proses *training* data tergantung jenis spesifikasi *hardware*-nya. Agar lebih memahami analisis tentang *deep learning* disajikan lebih dahulu grafik *time series* pengamatan suhu, kelembaban, kecepatan angin, dan tekanan udara untuk mendapat pola dan gambaran umum data (Gambar 3).



**Gambar 3.** Grafik *time series* pengamatan suhu, kelembaban, kecepatan angin dan tekanan udara yang akan digunakan sebagai uji *deep learning LSTM*. Perhatikan seluruh data sudah dilakukan interpolasi (Pers. 1) untuk mempermudah pengujian.

Terlihat pada Gambar 3, grafik suhu, kelembaban, dan tekanan udara angin memiliki pola *sinusoidal*. Nilai suhu udara berada pada rentang nilai 23,8 – 34,4<sup>o</sup> C, kelembaban udara pada rentang nilai 48 – 100 %, dan tekanan udara pada rentang nilai 1005 – 1015 mb. Untuk grafik kecepatan angin sendiri tidak memiliki pola beraturan dengan rentang nilai 0 – 20 m/s. Pada subbab berikutnya akan dijelaskan apakah tiap parameter memiliki korelasi kuat terhadap waktu pengamatan itu sendiri atau tidak.

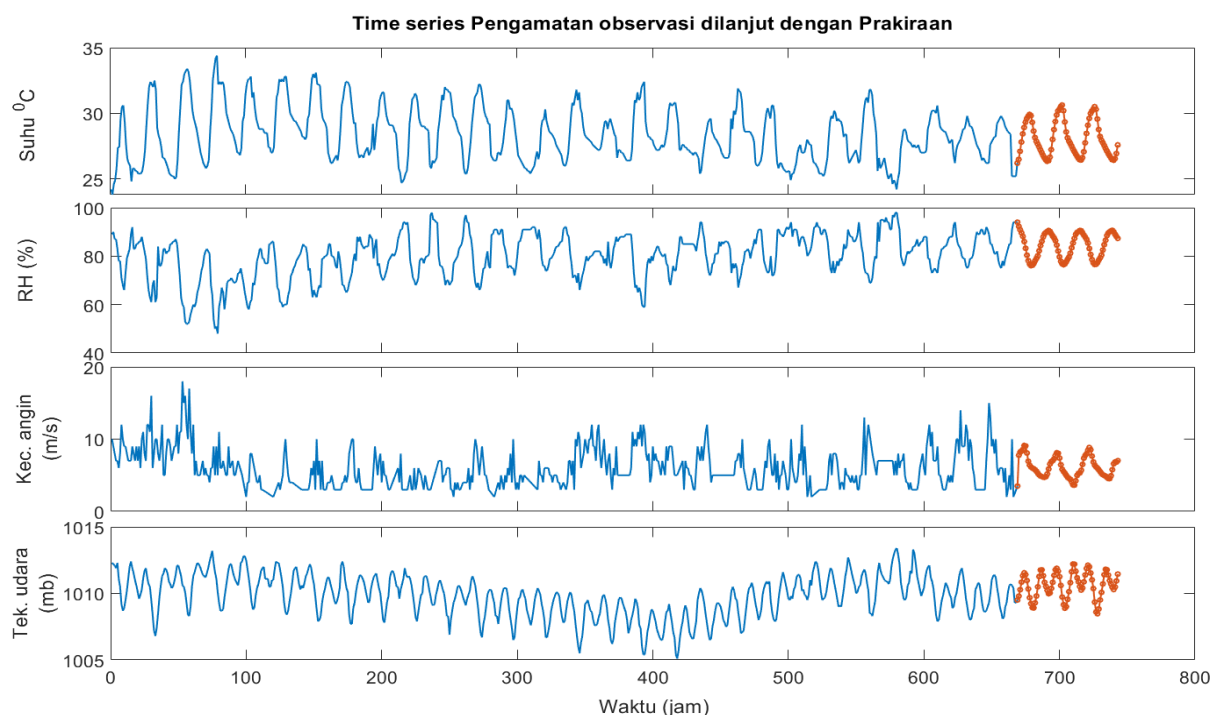
Pola yang dihasilkan dari grafik suhu, kelembaban, dan tekanan udara secara langsung dipengaruhi oleh penyinaran matahari terutama dalam skala waktu diurnal. Suhu udara maksimum terjadi pada siang dan minimum di malam hari. Sedangkan kelembaban dan tekanan udara cenderung maksimum menjelang pagi dan minimum menjelang malam hari [38]. Hal berbeda terjadi pada kecepatan angin, di mana hasilnya secara tidak langsung dipengaruhi oleh penyinaran matahari. Namun waktu kejadian suhu, kelembaban, tekanan udara dan kecepatan angin dapat tergantung dari faktor lainnya seperti tutupan awan dan kejadian hujan [39].

Waktu pada sumbu x ke empat grafik dibuat dari jam 0 hingga ke jam 670 selama bulan Januari 2019. Hal ini dilakukan untuk menerapkan rasio 9:1, yaitu 9 untuk *training* data dan 1 untuk *test* data pada *deep learning LSTM*. Tidak ada batasan khusus dalam penentuan rasio ini. Namun yang jelas nilai rasio pada *training* data harus lebih besar dari *test* data.

Beberapa penelitian terdahulu dalam membagi *training* dan *test* data ada yang menerapkan prinsip *pareto* dengan rasio 8:2 [40, 41]. Gholami [42] menggunakan rasio 7:3 untuk menghitung ketinggian air tanah menggunakan jaringan syaraf tiruan dan model *dendrochronology*.

Hasil *training* untuk parameter suhu, kelembaban, kecepatan angin, dan tekanan udara ditampilkan pada Gambar 4. Grafik berwarna biru merupakan hasil data observasi, sedangkan yang berwarna coklat hasil prakiraan *training deep learning LSTM* untuk 200 *layer*. Terlihat hasil prakiraan suhu, kelembaban, dan tekanan udara dari hasil *training* memiliki pola *sinusoidal* dari jam 670 hingga 744 di bulan Januari 2019. Sedangkan kecepatan angin berbeda dengan pola-pola sebelumnya

Bila diperhatikan secara seksama terdapat tiga puncak pada grafik prakiraan suhu udara (panel pertama Gambar 4). Nilai ini setara dengan 74 jam pengamatan, yaitu selisih dari jam 744 dengan 670). Hasil tersebut merupakan nilai yang sama untuk tiga hari data observasi, artinya 1 puncak untuk 1 hari. Ini sesuai dengan keadaan pengamatan sebenarnya di mana akan terdapat 1 suhu maksimal untuk 1 hari. Menurut Wikandari [43] suhu udara maksimum biasanya terjadi pada pukul 12 – 14. Namun hal ini berlainan dengan suhu udara maksimal yang dihasilkan dari grafik prakiraan (coklat). Waktu suhu maksimum prakiraan justru terjadi berturut-turut terjadi pada pukul 12:00, 11:00, dan 10:00 UTC.



**Gambar 4.** Grafik *time series* pengamatan observasi suhu (biru) dilanjutkan dengan grafik prakiraan hasil *training deep learning LSTM* (coklat).



Serupa dengan suhu, hasil *training data* kelembaban udara memiliki empat puncak nilai (panel kedua Gambar 4). Nilai maksimumnya berturut-turut terjadi pada pukul 02:00, 05:00, 07:00, dan 04:00 UTC. Sedangkan untuk kecepatan angin tetap memiliki tiga puncak nilai yang jelas. Hanya saja pola ini agak berlainan dengan pola jam sebelumnya. Polanya cenderung sinusoidal meskipun sudah dilakukan *training* dengan rasio 9:1. Terakhir untuk nilai prakiraan tekanan udara, hasilnya cenderung memiliki pola yang sama dengan pola jam sebelumnya. Di mana dalam tiga hari terdapat 2 nilai tekanan udara maksimum, masing-masing berkisar pada pukul 07:00 dan 20:00; 06:00 dan 19:00; serta 06:00 dan 19:00 UTC.

Hasil prakiraan yang diperoleh selanjutnya dibandingkan dengan pengamatan observasi tersedia untuk mengetahui seberapa jauh atau dekatnya hasil prakiraan dengan data observasi (validasi). Hasilnya ditunjukkan pada Gambar 5 dan 6. Terlihat pada kedua gambar memiliki hubungan yang cukup baik antara data observasi dengan hasil prakiraan.

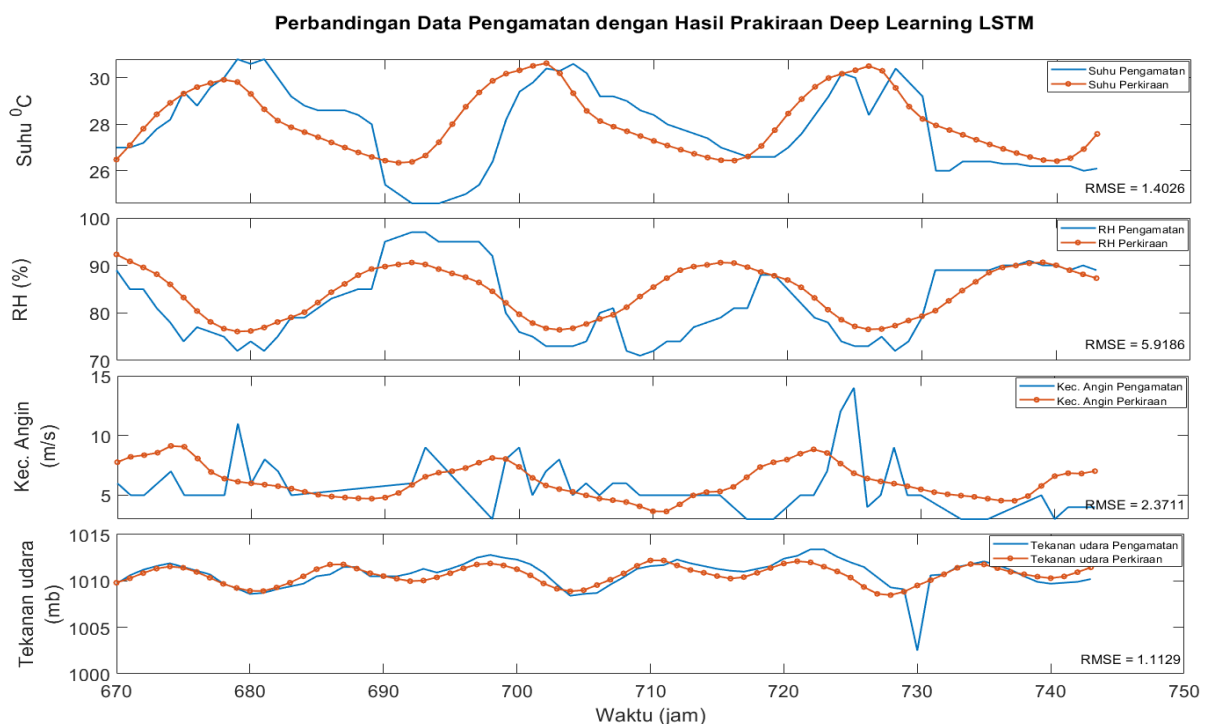
Gambar 5 menyajikan perbandingan hasil prakiraan antara data observasi dengan metode *deep learning* LSTM. Sedangkan Gambar 6 melalui metode *deep learning* LSTM dengan *update*. Keduanya memakai 200 *layer*. Perbandingan kedua gambar dimulai dari jam ke 670, yaitu 10% dari urutan data terakhir atau *test data*. Perhatikan pula RMSE-nya untuk semua

unsur pengamatan, terlihat nilainya semakin kecil ketika menggunakan LSTM dengan *update* dibandingkan LSTM tanpa *update*. Semakin rendah nilai RMSE semakin baik hasil prakiraannya.

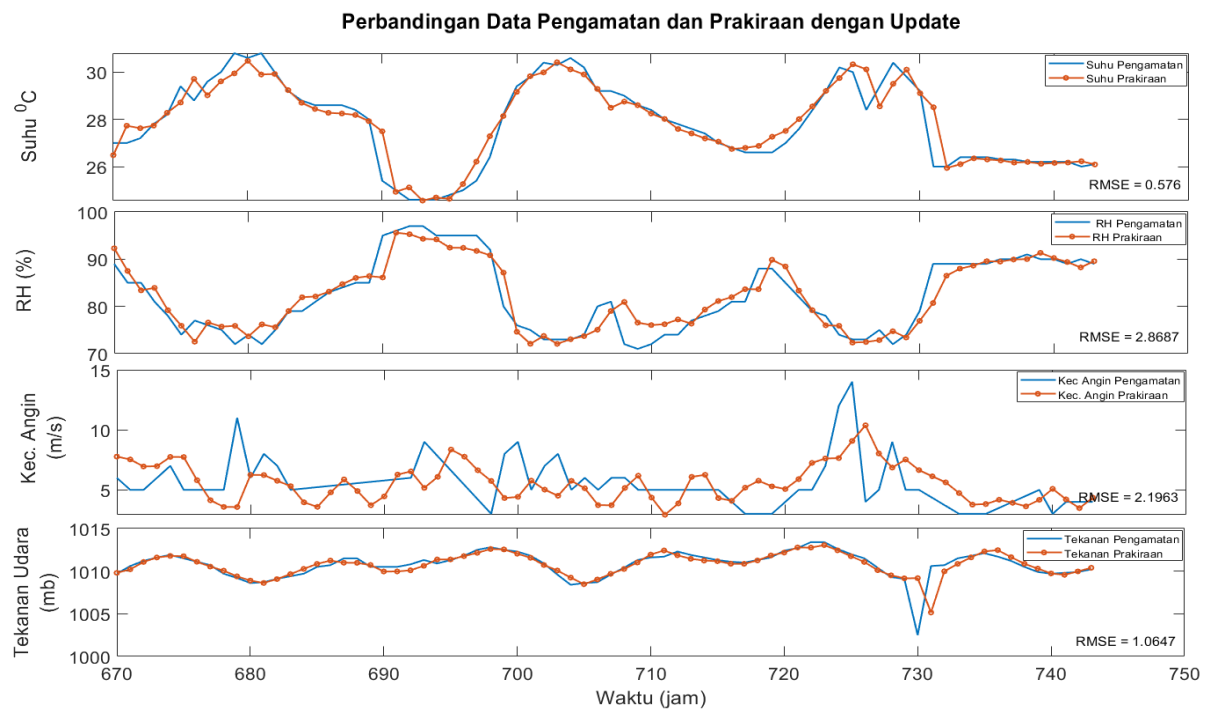
Bila diperhatikan seksama, hasil prakiraan dengan metode *deep learning* LSTM *update* menunjukkan pola yang lebih serupa dengan data observasi dibandingkan tanpa *update*. Pengertian *update* di sini adalah bila di antara waktu prakiraan kita memiliki nilai observasi yang aktual maka dapat dengan mudah ditentukan hasil prakiraan yang lebih tepat. Terlihat masing-masing parameter memiliki nilai RMSE 0,576; 2,8687; 2,1963; dan 1,0647 untuk suhu, kelembaban, kecepatan angin, dan tekanan udara.

Lebih lanjut, prakiraan LSTM pada parameter suhu dan tekanan udara mengalami faktor keterdahuluan untuk metode tanpa *update* terhadap data pengamatan (Gambar 5 panel 1 dan 4). Tetapi parameter lain seperti RH dan kecepatan angin faktor keterdahuluan ini tidak terlihat jelas. Kebalikannya pada keluaran LSTM dengan *update* justru mengalami keterlambatan disemua parameter.

Berdasarkan Gambar 6 dari empat parameter hanya kecepatan angin yang tidak memiliki kesamaan pola yang baik dibandingkan tiga parameter lainnya. Hal ini disebabkan nilai yang diperoleh dalam waktu yang berdekatan cenderung berfluktuatif. Sehingga mengakibatkan jangkauan nilai yang begitu besar.



**Gambar 5. Grafik perbandingan dan hubungan keeratan pengamatan observasi dengan prakiraan melalui *deep training* LSTM.**



**Gambar 6.** Serupa dengan Gambar 5 hanya saja dengan tambahan (*update*) data observasi.

Kajian yang dilakukan oleh Keskar [44] mengungkapkan sekumpulan data dengan jangkaun yang besar dapat menyebabkan perbedaan hasil prakiraan dengan hasil pengamatan. Di mana hasil prakiraan cenderung berkumpul pada satu titik minimum yang tajam (*sharp minima*).

Beberapa penelitian terdahulu dengan metode dan nilai rasio yang sama menunjukkan akurasi dan keeratan yang baik. Misalnya pada penelitian Tato [43] yang mengukur variasi emosi manusia dengan LSTM, diperoleh akurasi sebesar 67,2 % untuk rasio *training* dan *test* 9:1. Selanjutnya Hua [46] juga menggunakan LSTM untuk menghitung lalu lintas jaringan telekomunikasi dengan rasio yang sama. Didapatkan grafik RMSE melonjak dari 0,05 ke 0,11 ketika persentase perhitungan baru mencapai 20% dan relatif tetap sampai perhitungan selesai.

Bila dalam proses perbandingan pola grafik pengamatan dengan prakiraan menghasilkan nilai yang terlalu jauh maka ada beberapa hal yang mempengaruhi. Hoffer [47] menjelaskan penggunaan data dalam jumlah besar dapat mengakibatkan perbedaan hasil prakiraan dengan pengamatan dan bisa diperbaiki dengan menambah jumlah data aktual pada prakiraan untuk *update*.

Alasan lainnya juga dikemukakan oleh Reed [48] yang menyebutkan data dengan gangguan (*noise*) yang banyak mampu menurunkan akurasi prakiraan hingga 15 %. Selain itu penggunaan data yang tidak distandarisasi dan terdapat data bukan bilangan dalam *time series* (*not a number* atau *nan*) dapat

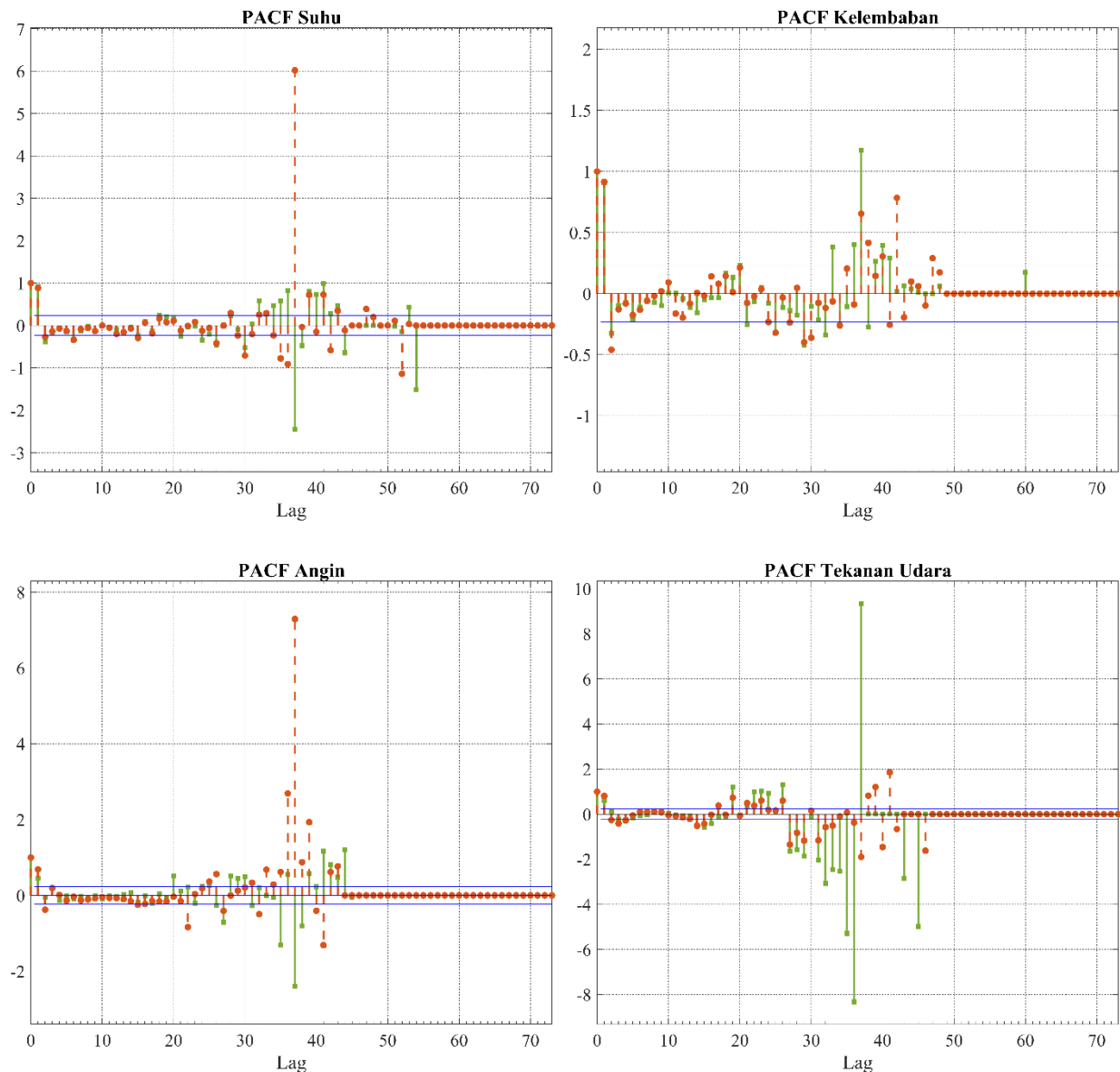
mengakibatkan perbedaan hasil prakiraan dengan pengamatan.

**Uji PACF.** Bila pada pembahasan sebelumnya kita hanya melihat data dalam skala waktu maka pada bagian ini akan diuji apakah data setiap parameter memiliki keterikatan terhadap waktu pengamatan atau tidak. Untuk itu dilakukan uji PACF dengan batas  $SK_{95\%}$ . Perlu dicatat batas nilai kepercayaan ini ditentukan banyak data. Sehingga hasil berbeda mungkin diperoleh ketika memakai jumlah data yang tidak sama dengan penelitian ini. Hasilnya disajikan pada Gambar 7.

Terlihat pada Gambar 7 seluruh parameter memiliki nilai PACF unik pada *lag* ke - 0 sebesar 1. Hal ini wajar mengingat pada tahap ini semua variabel akan menghitung nilai parameter pertama sebagai *lag* ke - 0. Disusul *lag* ke - 1 yang masih melewati  $SK_{95\%}$ . Selain itu pada *lag* ke - 37 (30 Januari 2019 pukul 09.00 UTC) seluruh parameter memiliki nilai PACF melebihi  $SK_{95\%}$ . Ini menunjukkan pada waktu tersebut data tidak stasioner. Bila di perhatikan seksama pada *lag* ke - 20 sampai 50 umumnya data bersifat tidak stasioner. Namun begitu memasuki *lag* ke - 50 untuk semua parameter nilai PACF berada pada  $SK_{95\%}$ . Ini menandakan data pada *lag* tersebut sudah stasioner.

Penelitian uji PACF juga yang dilakukan oleh Alsharif [49] pada data radiasi matahari bulanan selama 37 tahun. Diperoleh nilai PACF meluruh pada *lag*-ke 4 dari uji 60 data yang disediakan dengan kecenderungan data stasioner memasuki data ke 50.



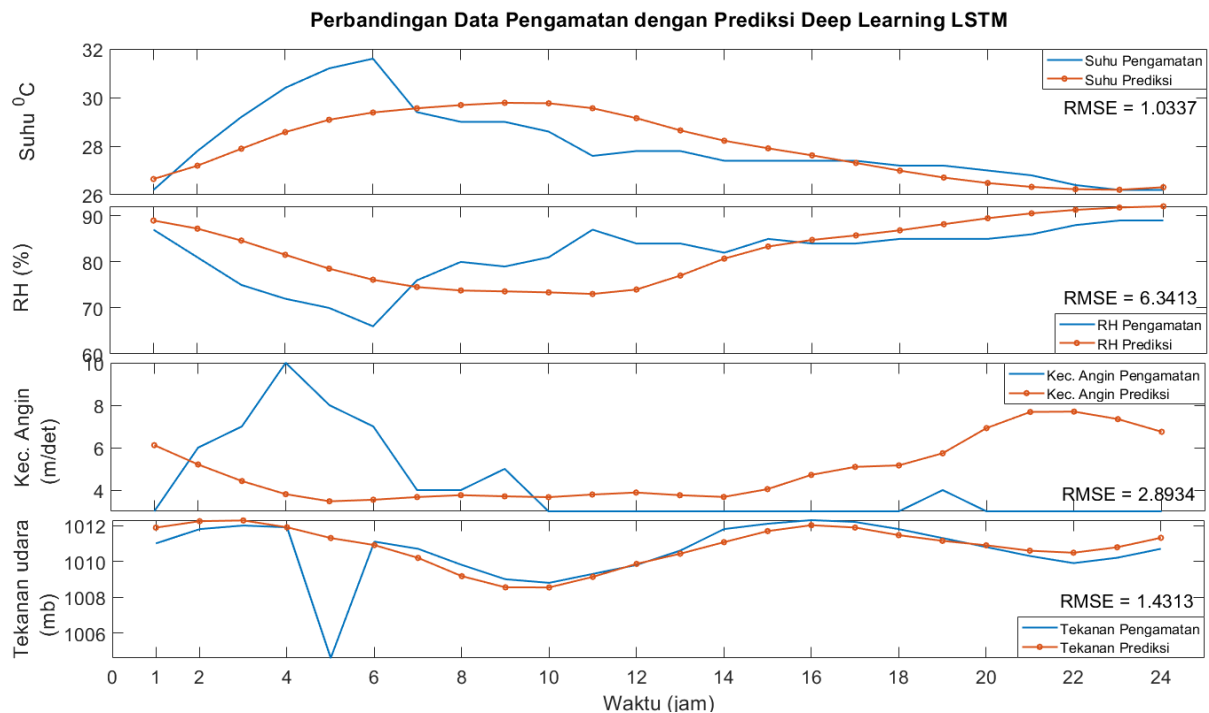


**Gambar 7.** *Correlogram* PACF untuk tiap parameter cuaca. Garis coklat putus – putus menyatakan data prakiraan, sedangkan garis hijau solid menyatakan data pengamatan. Garis biru horizontal menyatakan batas selang kepercayaan 95 % (Pers. 14). Data prakiraan yang digunakan seluruhnya dari hasil *deep learning* LSTM dengan *update*.

Lebih lanjut, kajian yang dilakukan Chawsheen [50] dalam mengolah data suhu bulanan di Irak selama 32 tahun mendapatkan nilai ACF dan PACF yang memiliki siklus setiap 12 bulannya. Terutama pada uji ACF terlihat jelas membentuk pola sinusodal dibandingkan PACF yang cenderung muncul sebagai *spikes*. Pola ACF yang dihasilkan sesuai dengan penelitian ini. Namun tidak bagi PACF yang cenderung muncul *spikes* menyebar di antara *lag* – 20 sampai *lag* – 50.

Dalam perspektif meteorologi, keberadaan autokorelasi dapat mengukur kekuatan gangguan dari skala ruang dan waktu. Ketika autokorelasi mendekati nol gangguan tidak akan muncul atau berulang lagi. Selain itu gangguan dapat muncul [51].

**Prediksi dengan *deep learning* LSTM.** Pada bagian ini akan diterapkan kemampuan *deep learning* LSTM untuk memprediksi parameter cuaca, seperti suhu udara, kelembaban, kecepatan angin, dan tekanan udara. Hasil prediksi hingga 1 hari ke depan, yaitu 1 Februari 2019 disajikan pada Gambar 8. Terlihat prediksi seluruh parameter cuaca memiliki pola yang hampir mirip dengan data pengamatan. Didapat masing-masing parameter memiliki nilai RMSE sebesar 1,0337; 6,3413; 2,8934; dan 1,4313 untuk suhu, kelembaban, kecepatan angin, dan tekanan udara. Dari keempat parameter hanya kelembaban udara yang memiliki hubungan jauh lebih besar. Padahal pada tahap validasi sebelumnya (Gambar 6), kelembaban hanya memiliki RMSE sebesar 2,8687. Nilai tersebut naik mencapai hampir 2,2 kali lipat.



**Gambar 8.** Grafik perbandingan dan hubungan keceratan data pengamatan (biru) selama 24 jam ke depan terhadap prediksi hasil *deep learning* LSTM (coklat).

Hal tersebut menandakan metode *deep learning* LSTM kurang cocok untuk memprediksi nilai kelembapan udara pada jenis data yang digunakan dalam penelitian ini. Namun demikian tetap diperlukan kajian mendalam menggunakan lokasi berbeda dan rentang waktu yang lebih panjang untuk melihat fenomena ini.

Dari seluruh prediksi parameter cuaca di Gambar 8 terdapat satu kesamaan yang berlaku untuk seluruh parameter, yaitu tidak dapat memprediksi secara tepat nilai fluktuasi yang terjadi dalam rentang waktu singkat. Perhatikan panel pertama Gambar 6, hasil prediksi tidak mampu memprediksi nilai fluktuatif data pengamatan di pukul 6 UTC. Pada panel kedua, hasil prediksi tidak mampu juga memprediksi nilai fluktuatif data pengamatan pukul 6 dan 11 UTC. Pada panel ketiga, hasil prediksi tidak mampu memprediksi nilai fluktuatif yang terjadi pada pukul 4 UTC. Sedangkan pada panel keempat kedua grafik mampu berdampingan dengan baik, namun masih gagal memprediksi fluktuatif di pukul 5 UTC.

Prediksi 1 hari ke depan dirasa masih belum cukup untuk melengkapi kebutuhan prakiraan cuaca BMKG. Sehingga pada penjelasan berikutnya kita akan melihat sejauh mana kehandalan *deep learning* LSTM dalam memprediksi parameter cuaca dalam jangka waktu 3 hari, 1 minggu, dan 1 bulan kedepan (Tabel 2). Pembanding yang digunakan tetap sama, yaitu data bulan Februari 2019.

**Tabel 2.** *Timestep* RMSE prediksi parameter cuaca beberapa hari kedepan (akurat hingga 4 angka dibelakang koma).

Parameter	24 jam	3 hari	7 hari	1 bulan
Suhu udara	1,0337	1,0443	1,1229	1,3631
RH	6,3413	6,1320	6,9393	10,8324
Kec. angin	2,8934	2,6297	2,7984	3,0342
Tekanan	1,4313	0,9191	0,9770	2,1709

Pemilihan waktu 24 jam, 3 hari, dan 7 hari disesuaikan dengan produk prakiraan cuaca BMKG yang telah berjalan selama ini, sedangkan waktu 1 bulan dipilih untuk melihat sejauh mana kehandalan metode *deep learning* LSTM bila diterapkan dalam jangka waktu lama. Terlihat pada Tabel 2 hanya parameter suhu udara dan RH yang mengalami pertambahan RMSE seiring dengan bertambahnya waktu. Sedangkan untuk parameter kecepatan angin dan tekanan udara awalnya mengalami penurunan dihari ke tiga, namun meningkat lagi pada prediksi hari ke 7 hingga 1 bulan ke depan.

*Timestep* RMSE kecepatan angin cenderung rendah di hari ketiga disebabkan pola *time series* yang terbentuk sebelumnya bersifat acak (panel ketiga Gambar 3). Sedangkan untuk *timestep* tekanan udara, bila merujuk pada pola tekanan (panel keempat Gambar 3) hasilnya cenderung beraturan. Akan tetapi *timestep* RMSE yang dihasilkan cenderung rendah di hari ke tiga. Seharusnya nilainya terus meningkat bersamaan dengan bertambahnya waktu prediksi. Hal ini perlu membutuhkan penelitian lanjutan.

Idealnya bila merujuk Tabel 2, hasil prediksi hanya bisa digunakan maksimal sampai tiga hari saja. Bila waktu prediksi terus ditambah dikhawatirkan rentan muncul bias tinggi sebagai akibat tingginya nilai RMSE. Hal ini berkaitan dengan rasio pembagian data 9:1 sebagai data *training* dan data *test* yang digunakan dalam penelitian ini. Walaupun hasil yang diperoleh Tabel 2 bukan sebagai data *test*, namun pemanfaatan hasil jaringan (*net*) data *training*-nya berperan sebagai proses pembelajaran *deep learning* LSTM. Sehingga bila ingin memprediksi sampai 7 hari dan 1 bulan ke depan, data *time series* yang digunakan idealnya masing-masing minimal 78 dan 334 hari berurut.

Namun demikian, penentuan prediksi hingga 1 bulan mendatang sangat jarang terjadi sebab menurut Kirtman [52], prediksi yang dilakukan semakin jauh dari waktu awal justru menghasilkan nilai hubungan yang semakin buruk. Hal ini disebabkan pengaruh eksternal yang terkadang tidak tertangkap dalam proses *training* data. Posisi wilayah pengamatan turut mempengaruhi performa suatu prediksi cuaca. Pada wilayah lintang menengah prediksi cuaca dapat dilakukan hingga 2 minggu kedepan [53] dan 4 hari pada wilayah tropis [54].

#### 4. Kesimpulan

Pada penelitian ini telah dilakukan penggunaan metode *deep learning* LSTM untuk memprediksi parameter cuaca, seperti suhu udara, kelembaban, kecepatan angin, dan tekanan udara. Metode ini bekerja dengan memanfaatkan fungsi matematika seperti fungsi *tanh* dan *sigmoid* yang berada dalam *layer* LSTM. Adapun jumlah *layer* yang digunakan sebanyak 200 buah. Sedangkan jumlah datanya dibagi dua menjadi *training* data dan *test* data dengan rasio 9:1 pada bulan Januari 2019.

Setelah dilakukan *training* data maka dilanjutkan proses pengolahan data prakiraan untuk dibandingkan dengan data observasi. Diperoleh RMSE parameter suhu udara, kelembaban, kecepatan angin, dan tekanan udara nilainya semakin baik ketika menggunakan *deep learning* LSTM dengan *update* dibandingkan LSTM tanpa *update*. Sebab pada penggunaan LSTM dengan *update* terdapat nilai observasi aktual yang dapat meningkatkan hasil prakiraan.

Dari hasil uji PACF menunjukkan data prakiraan LSTM dan pengamatan pada seluruh parameter cuaca mengalami ketidak-stasioner pada *lag* ke - 20 hingga 50 dan stasioner pada *lag* ke - 50 hingga 73 pada  $SK_{95\%}$ . Selain itu terlihat pada *lag* ke - 37 seluruh parameter cuaca nilai PACF sangat berfluktuatif.

Setelah dilakukan *training* dan *test* data selanjutnya dilakukan prediksi selama 1 hari, 3 hari, 7 hari, dan 1 bulan ke depan di bulan Februari 2019 menggunakan hasil *training* yang sudah dilakukan di bulan Januari 2019. Diperoleh hasil prediksi suhu udara, kelembaban, kecepatan angin, dan tekanan udara 1 hari ke depan memiliki RMSE yang baik.

Dari parameter cuaca tersebut hanya parameter suhu dan kelembaban udara yang mengalami pertambahan RMSE seiring bertambahnya waktu. Sedangkan parameter kecepatan angin dan tekanan udara mengalami penurunan di hari ketiga dan meningkat secara kontinu hingga 1 bulan ke depan.

**Saran.** Penelitian ini masih dapat dikembangkan dengan menambahkan jumlah *layer* yang digunakan untuk melihat perubahan nilai RMSE yang paling kecil, deret *time series* data yang lebih panjang, dan ujicoba di lokasi lain untuk mengetahui kehandalan *deep learning* LSTM. Namun hal tersebut dapat dilakukan dengan baik bila menggunakan komponen CPU dan GPU berspesifikasi tinggi.

#### Ucapan Terima Kasih

Penulis mengucapkan rasa terima kasih setinggi-tingginya kepada Nurul Dwi Chandra yang telah memberikan dukungan dana dalam penelitian ini, rekan-rekan Pusat Meteorologi Maritim BMKG dan *anonymous reviewer* yang telah menelaah naskah ini.

#### Daftar Pustaka

- [1] BMKG. Laporan Indeks Kepuasan Masyarakat 2018, Pusat Meteorologi Publik BMKG.
- [2] Kemenkeu. Daftar Isian Pelaksanaan Anggaran (DIPA) Induk BMKG Tahun Anggaran 2018.
- [3] G. Hinton, *et al*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, pp.82-97, 2012.
- [4] G.E. Hinton, S. Osindero, and Y-W Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Journal Neural Computation*, vol. 18(7), 1527 - 1554, 2006.
- [5] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ICLR*, 2015.
- [6] K. He, X. Zhang, S. en, and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

- [7] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.35 (8), pp. 1798–1828, 2013.
- [8] J. Schmidhuber, "Deep Learning in Neural Networks: an Overview," *Neural Networks*, vol. 61 pp. 85–117, 2015.
- [9] Y. Bengio, Y. LeCun, G. Hinton, "Deep Learning," *Nature*, vol. 521 (7553), pp. 436–444, 2015.
- [10] G. Chen, "A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation," Internet: <https://arxiv.org/pdf/1610.02583>, diakses 16 Maret 2019.
- [11] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling", *Interspeech*, 2014.
- [12] The Clever Machine, "Derivation: Derivatives for Common Neural Network Activation Functions," Internet: <https://theclevermachine.wordpress.com/2014/09/08/derivation-derivatives-for-common-neural-network-activation-functions/>, diakses 22 Maret 2019.
- [13] Nvidia, "Long-Short Term Memory (LSTM)". Internet <https://developer.nvidia.com/discover/lstm>, diakses 16 Maret 2019.
- [14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9(8), pp. 1735-1780, 1997.
- [15] S.A. Haider *et al*, "LSTM Neural Network Based Forecasting Model for Wheat Production in Pakistan," *Agronomy*, vol. 9(2), 2019.
- [16] D. S. Reddy and P. R. C. Prasad, "Prediction of Vegetation Dynamics Using NDVI Time Series Data and LSTM," *Modelling earth systems and Environment*, vol. 4(1), pp. 409 - 419, 2018.
- [17] J. Zhang, Y. Zhu, X. Zhanf, M. Ye, and J. Yang, "Developing a Long Short-Term Memory (LSTM) Based Model for Predicting Water Table in Agricultural Areas," *Journal of Hydrology*, vol. 561, pp. 918 -929, 2018.
- [18] Z. Jiang, *et al*, "Predicting County Level Corn Yields Using Deep Long Short Term Memory Models," Internet: <https://arxiv.org/abs/1805.12044>, diakses 17 Maret 2018.
- [19] J. Shook, L. Wu, T. Gangopadhyay, B. Ganapathysubramanian, and S. Sarkar, "Integrating Genotype And Weather Variables for Soybean Yield Prediction Using Deep Learning," Iowa State University, 2018.
- [20] X. Li, "Pollutant Concentration Predictions: Method Development And Evaluation," *Environmental Pollution*, vol. 231, pp. 997-1004, 2017.
- [21] C-J. Huang and P.H.Kuo, "A Deep CNN-LSTM Model for Particulate Matter (PM<sub>2.5</sub>) Forecasting in Smart Cities", *Sensors* vol.18(7), 2018.
- [22] M.Sundermeyer, R. Schluter, and H. Ney, "LSTM Neural Networks for Language Modelling," *Interspeech 2012*, 2012.
- [23] M.Sudermeyer, H. Ney, and R. Schluter, "From Feedforward to Recurrent LSTM Neural Networks for Language Modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23(3), 2015.
- [24] O. Vinyals, et al, "Grammar as Foreign Language," *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.
- [25] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-Aware Neural Language Models," *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016.
- [26] H. Bakir, G. Chniti, and H. Zaher, "E-Commerce Price Forecasting Using LSTM Neural Networks," *International Journal of Machine Learning and Computing*, vol.18 (2), pp. 169 - 174, 2018.
- [27] G. Chniti, H. Bakir, and H. Zaher, "E-commerce Time Series Forecasting using LSTM Neural Network and Support Vector Regression," *Proceedings of the International Conference on Big Data and Internet of Thing*, pp. 80 - 84, 2017.
- [28] X. Shi, Z. Chen, H. Wang, and D-Y. Yeung, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," Internet: <https://arxiv.org/abs/1506.04214>, diakses 17 Maret 2019.
- [29] M.A Zaytar and C. E. Amrani, "Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks," *International Journal of Computer Applications*, vol. 143(11), pp. 7-11, 2016.
- [30] L. Wiest, "Recurrent Neural Networks Combination of RNN and CNN," Internet: <https://wiki.tum.de/display/lfdv/Recurrent+Neural+Networks+-+Combination+of+RNN+and+CNN>, diakses 20 April 2019.
- [31] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Computation*, vol. 29, pp. 2352-2449, 2017.

- [32] B. Klein, L. Wolf, and Y. Afek, "A Dynamic Convolutional Layer for Short range Weather Prediction," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4840 - 4848, 2015.
- [33] E. Nour Arrofiqoh., "Otomatisasi ekstraksi fitur bangunan dengan memanfaatkan metode deep learning untuk interpretasi citra satelit resolusi tinggi," Tesis, Fakultas Teknik, Universitas Gadjah Mada, Yogyakarta, 2018.
- [34] Z. Reitermanova, "Data Splitting," *WDS'S Proceedings of Contributed Papers, part I*, pp. 31-36, 2010.
- [35] P.J. Brockwell dan R.A. Davis, "Introduction Time Series and Forecasting", Second Edition. Springer, USA, 2002.
- [36] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, dan G. M. Ljung. "Time Series Analysis: Forecasting and Control", Fifth Edition. John Wiley & Sons, USA, 2015.
- [37] A. Dey, "Machine Learning Algorithms: A Review," *International Journal of Computer Science and Information Technologies*, vol. 7(3), pp. 1174-1179, 2016.
- [38] I.I. Tika, "Variasi Suhu dan RH di Taman Suropati dan Sekitarnya," Skripsi, Fakultas MIPA: Universitas Indonesia, 2010.
- [39] R. Anggreni, Muliadi, and R. Adriat, "Analisis Pengaruh Tutupan Awan terhadap Radiasi Matahari di Kota Pontianak," *Prisma Fisika*, vol. 3, pp.214-219, 2018.
- [40] G.E.P. Box and R.D. Meyer, "An Analysis for Unreplicated Fractional Factorials," *Technometrics*, vol. 28(1), 1986.
- [41] M.E.J Newman, "Power Laws, Pareto Distributions and Zipf's Law," Internet: [https://arxiv.org/PS\\_cache/cond-mat/pdf/0412/0412004v3.pdf](https://arxiv.org/PS_cache/cond-mat/pdf/0412/0412004v3.pdf), diakses 21 Maret 2019.
- [42] V. Gholami, K.W. Chau, F. Fadaee, J. Tokaman, and A. Ghaffari, "Modeling Of Groundwater Level Fluctuations Using Dendrochronology in Alluvia Aquifers," *Journal of Hydrology*, vol. 529(3), pp. 1060-1069, 2015.
- [43] N.P.I. Wikandari, H. Harsa, A.F. Assomadi, dan R. Boedisantoso, "Penentuan Korelasi Perubahan Kecepatan Angin dan Kekuatan Radiasi terhadap Ketinggian Lapisan inversi dan Hubungannya dengan Kualitas Udara Ambien Kota Surabaya," *Jurnal Teknik ITS*, vol. 4(1), pp. F120-F124, 2015.
- [44] N.S. Keskar, J. Nocedal, P.T.P. Tang, D. Mudigere, and M. Smelyanskiy, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," Internet: <https://arxiv.org/abs/1609.04836>, diakses 24 Maret 2019.
- [45] A. Tato, R. Nkambou, and C. Frasson, "Predicting from Multimodal User Data using a Deep Long Term Memory," *UMAP*, 2018, Internet: [https://www.researchgate.net/publication/326212084\\_Predicting\\_Emotions\\_From\\_Multimodal\\_Users'\\_Data](https://www.researchgate.net/publication/326212084_Predicting_Emotions_From_Multimodal_Users'_Data), diakses 24 Maret 2019.
- [46] Y. Hua, *et al*, "Deep Learning with Long Short-Term Memory for Time Series Prediction," Internet: <https://arxiv.org/pdf/1810.10161>, diakses 25 Maret 2019.
- [47] E. Hoffer, I. Hubara, and D. Soudry, "Train Longer, generalize better: Closing the Generalization Gap in Large Batch Training of Neural Networks," Internet: <https://arxiv.org/abs/1705.08741>, diakses 25 Maret 2019.
- [48] S.E. Reed *et al*, "Training Deep Neural Network on noisy Labels with Bootstrapping," Internet: <https://arxiv.org/abs/1412.6596>, diakses 24 Maret 2019.
- [49] M.H. Alsharif, M.K. Younes, and J.Kim, "Time Series ARIMA Model for Prediction of Daily and Monthly Average Global Solar Radiation: The Case Study of Seoul, South Korea," *Symmetry*, vol. 11(240), pp.1-17, 2019.
- [50] T.A. Chawsheen and M. Broom, "Seasonal time-series modelling and forecasting of monthly mean temperature for decision making in the Kurdistan Region of Iraq," *Journal of Statistical Theory and Practice*, vol. 11(4), pp. 604-633, 2017.
- [51] R.B. Stull., "An introduction to Boundary Layer Meteorology", First edition, Kluwer Academic Publishers, Netherlands, 1988.
- [52] Kirtman, *et al*, "Near-term Climate Change: Projections and Predictability," Cambridge University Press.
- [53] Zhang, *et al*, "What Is the Predictability Limit of Midlatitude Weather," *Journal of the Atmospheric Sciences*, vol. 76, pp. 1077 - 1091, 2019.
- [54] Zhu, *et al.*, "Seamless Precipitation Prediction Skill in the Tropics and Extratropics from a Global Model," *Monthly Weather Review*, vol. 142, pp. 1556 - 1569, 2014.