

Perbandingan Performa Model *Long Short-Term Memory* dan *Bidirectional* untuk Prediksi Kabut

<http://dx.doi.org/10.28932/jutisi.v7i2.10588>

Riwayat Artikel

Received: 17 Desember 2024 | Final Revision: 22 Juli 2025 | Accepted: 22 Juli 2025

Creative Commons License 4.0 (CC BY – NC)



Atri Wiujianna^{✉#1}, Sunarno^{#2}, Iqbal^{*3}

[#] Fisika, Universitas Negeri Semarang
Sekaran Gunung Pati, Kota Semarang, 50229, Indonesia

*Badan Meteorologi, Klimatologi dan Geofisika

¹atriwiujianna@students.unnes.ac.id

²Sunarno@mail.unnes.ac.id

³Iqbal@bmkgo.go.id

✉Corresponding author: atriwiujianna@students.unnes.ac.id

Abstrak — Kabut merupakan fenomena cuaca yang dapat menurunkan jarak pandang secara signifikan dan berdampak pada keselamatan transportasi serta aktivitas masyarakat. Wilayah Citeko, Bogor, yang berada di dataran tinggi, memiliki frekuensi kejadian kabut yang cukup tinggi, terutama pada pagi hari dan musim hujan. Penelitian ini bertujuan untuk membangun dan membandingkan performa model prediksi kabut menggunakan algoritma *Long Short-Term Memory* (LSTM) dan *Bidirectional LSTM* (BiLSTM) berdasarkan data cuaca historis selama periode 2013–2023. Data yang digunakan berasal dari Stasiun Meteorologi Citeko dan mencakup parameter cuaca seperti suhu bola kering, suhu bola basah, titik embun, jarak pandang, kelembapan relatif, tutupan awan, arah dan kecepatan angin, serta kondisi cuaca per jam. Data mengalami beberapa tahap pra-pemrosesan, termasuk interpolasi nilai hilang, klasifikasi kabut berdasarkan parameter cuaca, normalisasi, dan pembagian data menjadi data latih dan uji (80:20). Model LSTM dan BiLSTM kemudian dilatih menggunakan pendekatan *deep learning*, baik dengan maupun tanpa *early stopping*. Hasil menunjukkan BiLSTM dengan *early stopping* mencapai kinerja terbaik: akurasi 99,93%, *precision* 96,53%, *recall* 98,81%, dan *F1-score* 97,66%, dengan hanya 9 *false positive* dan 3 *false negative*. Penelitian ini memberikan kontribusi dalam pengembangan sistem prediksi kabut berbasis kecerdasan buatan.

Kata kunci— BiLSTM; Data Cuaca; Kabut; LSTM.

Comparison of Long Short-Term Memory and Bidirectional Models for Fog Prediction

Abstract — Fog is a weather phenomenon that can significantly reduce visibility and impact transportation safety as well as public activities. The Citeko region in Bogor, located in a highland area, experiences a relatively high frequency of fog events, especially during the morning and rainy seasons. This study aims to develop and compare the performance of fog prediction models using Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) algorithms based on historical weather data from 2013 to 2023. The data, obtained from the Citeko Meteorological Station, includes weather parameters such as dry-bulb temperature, wet-bulb temperature, dew point, visibility, relative humidity, cloud cover, wind direction and speed, and hourly weather conditions. The data underwent several preprocessing steps, including missing value interpolation, fog classification based on weather parameters, normalization, and splitting into training and testing sets (80:20 ratio). The LSTM and BiLSTM models were then trained using a deep learning approach, both with and without early stopping. The results show that BiLSTM with early stopping achieved the best

performance: 99.93% accuracy, 96.53% precision, 98.81% recall, and an F1-score of 97.66%, with only 9 false positives and 3 false negatives. This study contributes to the development of fog prediction systems based on artificial intelligence.

Keywords— BiLSTM; Fog; LSTM; Weather Data.

I. PENDAHULUAN

Menurut *World Meteorological Organization* (WMO) Kabut adalah fenomena cuaca yang terjadi ketika suspensi tetesan air yang sangat kecil biasanya mikroskopis melayang di udara dan mengakibatkan berkurangnya jarak pandang di permukaan bumi [1]. Kondisi ini sering kali mengakibatkan gangguan pada layanan transportasi udara, kereta api, dan jalan raya karena jarak pandang yang terbatas, sehingga meningkatkan risiko kecelakaan. Meskipun sama-sama mengurangi jarak pandang, polusi udara (asap atau partikulat) tidak tergolong fenomena cuaca. Berdasarkan kode *ww* (*present weather*) kabut dan asap mempunyai klasifikasi kode yang berbeda.

Kabut lebih sering terjadi di daerah pegunungan dan dataran tinggi, termasuk di wilayah Citeko, Bogor yang terkenal dengan kondisi cuaca lembab dan kejadian kabut terutama pada pagi hari dan selama musim hujan. Citeko sebagai salah satu jalur destinasi pariwisata menuju puncak Bogor, kabut yang terjadi di wilayah ini dapat mengganggu transportasi dan berpotensi meningkatkan risiko kecelakaan akibat penurunan jarak pandang. Oleh karena itu, kebutuhan akan prediksi kabut yang akurat sangat penting untuk meminimalisir dampak negatif terhadap keselamatan dan aktivitas masyarakat serta wisatawan di wilayah ini.

Kabut terbentuk melalui proses kondensasi uap air menjadi tetesan cairan atau kristal es sebagai hasil dari pendinginan udara, peningkatan kelembaban atau pencampuran massa udara yang berbeda [2]. Partikel aerosol di udara memainkan peran penting sebagai inti kondensasi awan, yang dapat mempengaruhi sifat mikrofisik dan radiatif kabut [3]. Menurut penelitian Roach, 1995, bentuk kabut yang paling umum terjadi ketika pembentukannya di daratan melibatkan pendinginan radiasi pada malam hari di bawah kondisi angin rendah [2].

Namun, meskipun pembentukannya dapat dipahami, prediksi kabut sulit dilakukan karena sifatnya yang tidak menentu. Prediksi kabut tergantung pada pencampuran vertikal udara atmosfer dengan kelembaban permukaan yang dapat berubah dalam waktu yang sangat singkat, bahkan dalam hitungan detik [4].

Seiring perkembangan teknologi, metode prediksi cuaca telah mengalami banyak kemajuan pesat. Salah satu pendekatan yang kini berkembang adalah kecerdasan buatan (*Artificial Intelligence*, AI) untuk meningkatkan akurasi dan efisiensi prediksi cuaca, termasuk fenomena kabut. Di antara berbagai Teknik dalam AI, *Machine Learning* (ML) dapat memprediksi karena kemampuannya untuk memproses data dalam jumlah besar, mengungkap pola-pola tersembunyi, dan menghasilkan prediksi lebih cepat dibanding metode tradisional.

Beberapa penelitian tentang prediksi kabut diantaranya model prakiraan kabut jangka pendek menggunakan model klasifikasi biner, multi-kelas dan regresi dengan hasil terbaik masing-masing diperoleh dari *Multi-Layer Perceptron* (MPL), *Gradient Boosting Classifier* (GBC) dan LSTM satu lapis [5]. Pendekatan ML lainnya diterapkan untuk memprediksi kabut dan kabut tebal [6]. Berikutnya penelitian menggunakan *decision tree* yang hasilnya bahwa *decision tree* mampu melakukan prediksi jangka pendek dengan tingkat akurasi 66% [7]. Penelitian lain membandingkan model *logistic regression* (LR) dan *random forest* (RF) yang menunjukkan model LR mempunyai sensitivitas yang lebih tinggi dibandingkan model RF pada awal malam, tetapi secara keseluruhan kinerja model RF lebih baik daripada model LR [8]. Pendekatan yang lebih kompleks menggunakan model BiSLTM-CNN berbasis *attention* untuk memprediksi jarak pandang jalan raya [9].

Berdasarkan tinjauan literatur yang dilakukan, belum ditemukan penelitian penelitian yang memprediksi kejadian kabut di wilayah Citeko, Bogor. Meskipun beberapa studi telah mengembangkan model prediksi kabut menggunakan berbagai metode *machine learning* dan *deep learning*. Penelitian-penelitian tersebut dilakukan di lokasi yang berbeda.

Dalam prediksi kabut, salah satu metode yang digunakan adalah *Long Short-Term memory* (LSTM), sebuah varian dari arsitektur *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi *vanishing gradient*. LSTM mampu mengingat informasi dalam waktu yang lebih lama berkat adanya *memory cell* [10]. *Bidirectional Long Short-Term Memory* (BiLSTM) adalah pengembangan lebih lanjut dari LSTM yang terdiri dari dua lapisan LSTM yang berjalan berlawanan arah. Lapisan maju memproses data dari awal ke akhir, sedangkan lapisan mundur memproses data dari akhir ke awal [11]. Hal ini memungkinkan BiLSTM untuk menangkap informasi kontekstual dari kedua arah.

Pengembangan penelitian mengenai AI dan prediksi kabut menjadi tantangan penting dalam dunia meteorologi. Oleh karena itu, penelitian ini bertujuan untuk membangun model prediksi kabut menggunakan metode LSTM dan BiLSTM. Model yang dikembangkan akan diuji untuk memprediksi kejadian kabut berdasarkan data historis yang relevan. Selain itu, penelitian ini juga berfokus pada perbandingan performa antara metode LSTM dan BiLSTM, dengan mengacu pada hasil evaluasi model untuk menentukan metode yang lebih efektif dalam memprediksi kabut. Evaluasi performa dilakukan menggunakan sejumlah metrik akurasi, sehingga hasil penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam penerapan metode *deep learning* untuk prediksi kabut di masa mendatang.

II. METODE

A. Data Penelitian

Penelitian ini menggunakan data cuaca yang diambil dari Stasiun Meteorologi Citeko Bogor yang terletak pada koordinat 6.70 °LS dan 106.85 °BT dengan ketinggian 920 mdpl. Pemilihan variabel dalam penelitian ini didasarkan pada kajian dari penelitian sebelumnya yang menunjukkan bahwa variabel-variabel cuaca memiliki pengaruh signifikan terhadap fenomena kabut [12], mengidentifikasi bahwa variabel-variabel cuaca yang digunakan dalam penelitiannya berperan penting dalam memodelkan fenomena kabut. Berdasarkan temuan tersebut, penelitian ini juga menggunakan variabel cuaca yang sama, seperti tercantum pada Tabel 1.

TABEL 1
VARIABLE CUACA

Parameter	Kode
Suhu Bola Kering	tt
Suhu Bola Basah	tw
Titik Embun	td
Jarak Pandang	vv
Kelembaban	rh
Tutupan Awan	n
Arah Angin	dd
Kecepatan Angin	ff
Kondisi Cuaca	ww

Proses pembentukan kabut memerlukan kondisi udara yang stabil, angin yang tenang dan kelembaban relatif yang tinggi. Arah angin dipertimbangkan oleh faktor cuaca lokal. Langit cerah juga merupakan faktor pembentuk kabut terutama kabut radiasi. Jarak pandang rendah adalah dampak dari fenomena kabut sehingga variabel ini perlu dipantau [12].

Variabel-variabel yang digunakan dalam penelitian ini direkam setiap jam selama periode 1 Januari 2013 hingga 31 Desember 2023. Selama periode tersebut, data yang terkumpul mencakup 86904 baris dan 10 kolom, yang merepresentasikan berbagai parameter cuaca secara terperinci untuk analisis lebih lanjut. Jumlah data dari beberapa parameter cuaca terdapat pada Tabel 2.

TABEL 2
JUMLAH DATA

Parameter	Jumlah Data
Date	86903
Suhu Bola Kering	86852
Suhu Bola Basah	86811
Titik Embun	86655
Jarak Pandang	86838
Kelembaban	86653
Tutupan Awan	86758
Arah Angin	86844
Kecepatan Angin	86772
Kondisi Cuaca	86833

Penelitian ini menggunakan model *Long Short-Term memory* (LSTM) dan *Bidirectional Long Short-Term Memory* (BiLSTM). Model ini diimplementasikan di Google Colab dengan menggunakan bahasa pemrograman Python versi 3.10.12.

B. Data Preprocessing

Tahapan preprocessing data dilakukan sebelum membangun model LSTM dan BiLSTM. Proses pra-pemrosesan meliputi *data cleaning*, penambahan kolom, *statistik deskriptif*, korelasi data, normalisasi data dan pembagian data.

Tahap pertama dalam pra-pemrosesan data adalah pembersihan data (*data cleaning*), yang bertujuan untuk memperbaiki data yang memiliki nilai hilang (*missing value*). Penanganan terhadap *missing value* dilakukan melalui dua pendekatan yaitu untuk data pada kolom *date* yang hilang, baris data tersebut dihapus, sedangkan untuk kolom lainnya, nilai yang hilang diisi menggunakan metode interpolasi. Metode interpolasi ini mengisi nilai-nilai yang hilang dengan perhitungan berdasarkan data diantara dua titik yang diketahui. Setelah dilakukan *data cleaning*, dataset menjadi 86385 data untuk setiap parameter cuaca yang digunakan.

Tahap kedua, dilakukan penambahan kolom untuk mengelompokkan data menjadi dua kategori, yaitu *fog* dan tidak *fog*. Klasifikasi ini didasarkan pada data *present weather* yang menunjukkan kondisi cuaca saat pengamatan. Berdasarkan peraturan yang ditetapkan oleh BMKG, fenomena *fog* atau kabut diidentifikasi menggunakan kode tertentu pada *present weather* yang menunjukkan kabut selama pengamatan [13].

Tahap ketiga yaitu melakukan analisis statistik deskriptif untuk memberikan gambaran umum mengenai distribusi dan karakteristik data, kemudian dilakukan visualisasi distribusi data untuk mengetahui pola distribusi dari setiap fitur yang dipilih.

Selanjutnya tahap keempat adalah analisis korelasi data, yang bertujuan untuk menghitung hubungan antar fitur numerik dalam dataset. Hasilnya disajikan dalam bentuk matriks korelasi, dimana setiap elemen matriks menunjukkan koefisien korelasi Pearson antara dua fitur. Matriks korelasi ini menunjukkan informasi tentang seberapa kuat hubungan antara fitur-fitur numerik di dalam dataset.

Tahap kelima adalah normalisasi data, yang merupakan suatu teknik menskalakan atau memetakan data ke dalam rentang tertentu agar lebih mudah diolah oleh model. Salah satu teknik normalisasi data yang umum digunakan adalah *min-max normalization*. Teknik ini mengubah nilai data ke dalam rentang 0 hingga 1 [14], sehingga semua fitur memiliki skala yang sama. Proses normalisasi ini dilakukan menggunakan persamaan (1).

$$x_{norm,i} = \frac{x_i - x_{min}}{x_{max} - x_{min}} ; i = 1, 2, 3, \dots, t \quad (1)$$

Dimana x_{norm} merupakan nilai hasil normalisasi, x_{max} adalah nilai maksimum dari keseluruhan data, dan x_{min} adalah nilai minimum dari keseluruhan data.

Setelah menyelesaikan proses *data cleaning*, penambahan kolom, analisis statistik deskriptif, korelasi data dan normalisasi data, langkah selanjutnya adalah proses *data split* atau pembagian data. Pada tahap ini, data dibagi menjadi dua bagian, yaitu data *training* untuk melatih model dan data *testing* untuk menguji performa model.

C. Training dan Testing Setiap Model

Dataset dibagi menjadi dua bagian yaitu data pelatihan (*data training*) 80% dan data pengujian (*data testing*) 20%, seperti yang ditunjukkan pada Tabel 3. Data pelatihan digunakan untuk melatih model, sedangkan data pengujian digunakan untuk mengevaluasi kinerja model yang telah dilatih.

TABEL 3
PEMBAGIAN DATA TRAINING DAN TESTING

Persentase	Data Training	Data Testing
80% : 20%	69108	17277

Model dilatih menggunakan dua algoritma yaitu LSTM dan BiLSTM, dengan dua skenario pelatihan yaitu: tanpa *early stopping* dimana pelatihan dilakukan hingga mencapai jumlah *epoch* maksimum (50 *epoch*), dan dengan *early stopping* dimana teknik ini diterapkan untuk menghentikan pelatihan secara otomatis ketika kinerja model pada data validasi tidak membaik dalam beberapa *epoch*. Proses pelatihan melibatkan penggunaan data pelatihan untuk mengajarkan model mengenali pola dalam data. Proses pengujian dilakukan dengan memberikan data pengujian yang belum pernah dilihat oleh model sebelumnya. Hasil pengujian kedua skenario (dengan dan tanpa *early stopping*) dibandingkan untuk menilai seberapa baik model dapat memprediksi atau mengklasifikasikan data baru.

D. Evaluasi Model

Dalam penelitian ini, evaluasi kinerja model yang digunakan adalah *confusion matrix*, *accuracy*, *precision*, *recall* dan *F1-Score*. *Confusion matrix* yang digunakan adalah *confusion matrix biner*, dimana hanya ada dua kelas yaitu kelas positif dan kelas negatif [15]. Nilai *accuracy*, *precision*, *recall* dan *F1-Score* dapat dihitung menggunakan persamaan berikut

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$precision = \frac{TP}{TP+FP} \quad (3)$$

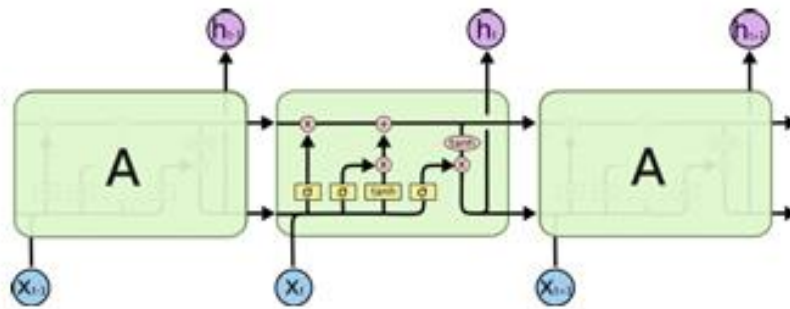
$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F1 = 2 * \frac{P*R}{P+R} \quad (5)$$

Dimana TP adalah *true positive*, TN adalah *true negative*, FP adalah *false positive*, FN adalah *false negative*, P adalah *precision* dan R adalah *recall*.

E. Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM), pertama kali diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997, adalah varian dari *Recurrent Neural Network* (RNN) [16], yang dirancang khusus untuk mengatasi masalah hilangnya informasi pada lapisan-lapisan awal jaringan saraf saat memproses data urutan yang panjang. Dengan adanya *memory cell*, LSTM dapat mengingat informasi dari waktu sebelumnya, sehingga mampu memproses dependensi jangka panjang dalam data sekuensial [10]. Arsitektur LSTM [16] dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur LSTM (Sumber: [16])

Arsitektur LSTM menggunakan tiga jenis gate untuk mengontrol aliran informasi antara lain *forget gate* (f_t), *input gate* (i_t), dan *output gate* (O_t) [17]. *Forget gate* (f_t) untuk memutuskan informasi mana yang tidak relevan, *Input gate* (i_t) untuk menentukan informasi baru yang akan disimpan dalam *memory state* dan *Output State* (O_t) untuk menghasilkan *output* berdasarkan *memory cell*. Berikut persamaan yang digunakan dalam proses LSTM [18].

$$f_t = \sigma(W_{fh}[h_{t-1}], W_{fx}[x_t], b_f) \quad (6)$$

$$i_t = \sigma(W_{ih}[h_{t-1}], W_{ix}[x_t], b_i) \quad (7)$$

$$\tilde{C}_t = \tanh(W_{ch}[h_{t-1}], W_{cx}[x_t], b_c) \quad (8)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (9)$$

$$O_t = \sigma(W_{oh}[h_{t-1}], W_{ox}[x_t], b_o) \quad (10)$$

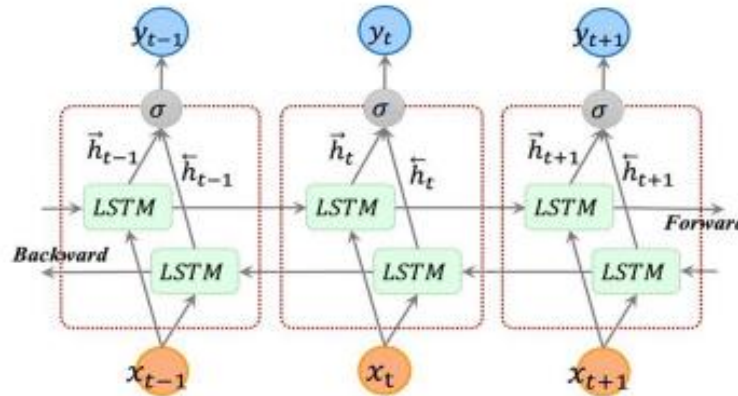
$$h_t = O_t * \tanh(C_t) \quad (11)$$

Persamaan (6) hingga (11) menunjukkan perhitungan matematis untuk setiap *gate*, dimana f_t adalah *forget gate*, i_t adalah *input gate*, \tilde{C}_t adalah nilai baru yang dapat ditambahkan pada *cell state*, C_t adalah *cell state*, O_t adalah *output gate*, h_t merupakan *output* akhir, σ adalah fungsi sigmoid, \tanh adalah fungsi tanh, h_{t-1} adalah nilai keluaran sebelum orde ke t, x_t adalah nilai input pada orde ke t. W_{fh} , W_{fx} , W_{ih} , W_{ix} , W_{ch} , W_{cx} , W_{oh} , W_{ox} adalah nilai bobot. b_f , b_i , b_c , b_o adalah nilai bias.

F. Bidirectional Long Short Term Memory (Bi-LSTM)

Bidirectional Long Short Term Memory (BiLSTM) merupakan pengembangan dari model *Long Short Term Memory* (LSTM) yang terdiri dari dua jaringan saraf LSTM yang bekerja secara berlawanan arah yaitu satu memproses data dari awal

ke akhir (*forward*) dan yang lainnya memproses data dari akhir ke awal (*backward*) [11]. Output kedua lapisan ini kemudian digabungkan pada setiap langkah waktu untuk menghasilkan *output* akhir dari model BiLSTM [19]. Dengan adanya kedua lapisan yang bekerja secara berlawanan arah, model BiLSTM mampu menangkap informasi baik dari masa lalu maupun masa depan dalam suatu urutan data, sehingga dapat memahami konteks yang lebih lengkap [20]. Ilustrasi dari arsitektur BiLSTM [21] dapat dilihat pada Gambar 2.



Gambar 2. Arsitektur BiLSTM (Sumber: [22])

Persamaan yang digunakan pada proses BiLSTM [23]

$$\vec{h}_t = \text{LSTM}(x_t, h_{t-1}) \quad (12)$$

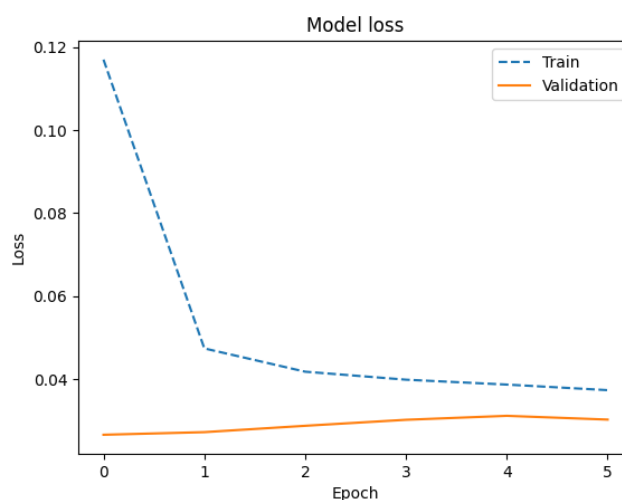
$$\overleftarrow{h}_t = \text{LSTM}(x_t, h_{t+1}) \quad (13)$$

$$h_t = [\vec{h}_t, \overleftarrow{h}_t] \quad (14)$$

Persamaan (12) merupakan proses *forward* LSTM, Persamaan (13) merupakan proses *backward* LSTM dan Persamaan (14) adalah output akhir BiLSTM yang merupakan gabungan dari kedua lapisan tersebut. Dimana h_t adalah *output gate* LSTM dua arah, \vec{h}_t adalah nilai keluaran LSTM maju, \overleftarrow{h}_t adalah nilai keluaran LSTM mundur, x_t adalah input pada waktu ke- t , h_{t-1} adalah hidden state pada waktu sebelumnya, h_{t+1} adalah hidden state pada waktu setelahnya.

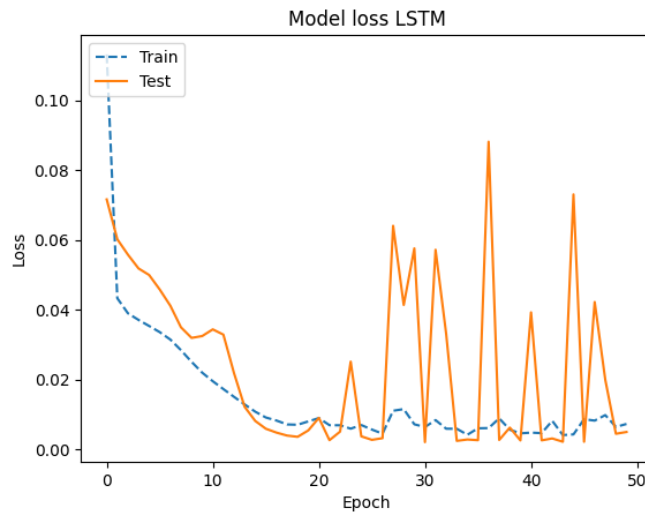
III. HASIL DAN PEMBAHASAN

Setelah dilakukan tahapan pra pemrosesan (*preprocessing*) seperti normalisasi data dan pembagian dataset menjadi data *training* dan data *testing*, proses selanjutnya adalah pelatihan model menggunakan algoritma LSTM dan BiLSTM. Performa model selama pelatihan dievaluasi menggunakan metrik *loss* untuk data pelatihan dan validasi, sebagaimana ditampilkan dalam gambar 3 sampai gambar 6.



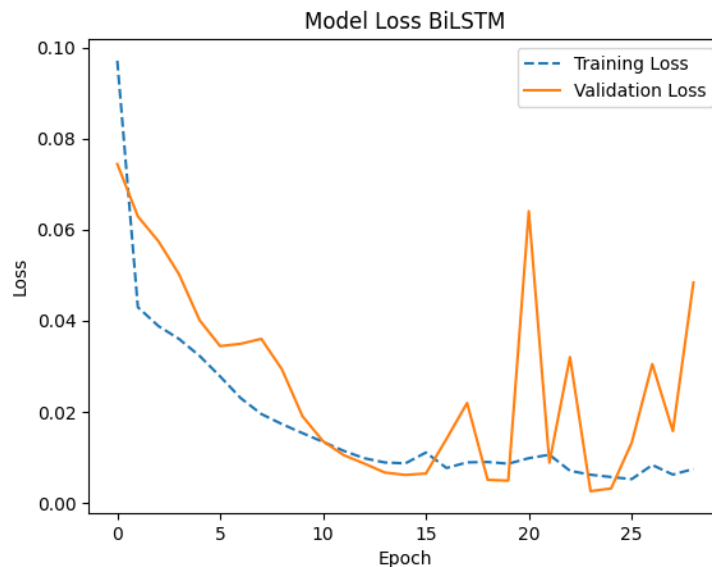
Gambar 1. Grafik Training Loss dan Validation Loss Model LSTM dengan Early Stopping

Gambar 3 menampilkan grafik perbandingan nilai *training loss* dan *validation loss* selama proses pelatihan model LSTM dengan penambahan *early stopping*. Pada grafik tersebut, terlihat bahwa nilai *training loss* mengalami penurunan yang cukup signifikan dari *epoch* ke-0 hingga *epoch* ke-1, dan terus menurun secara bertahap hingga *epoch* ke-5. Sementara itu, *validation loss* cenderung meningkat perlahan dari *epoch* ke-1 hingga ke-4, sebelum akhirnya sedikit menurun di *epoch* ke-5.



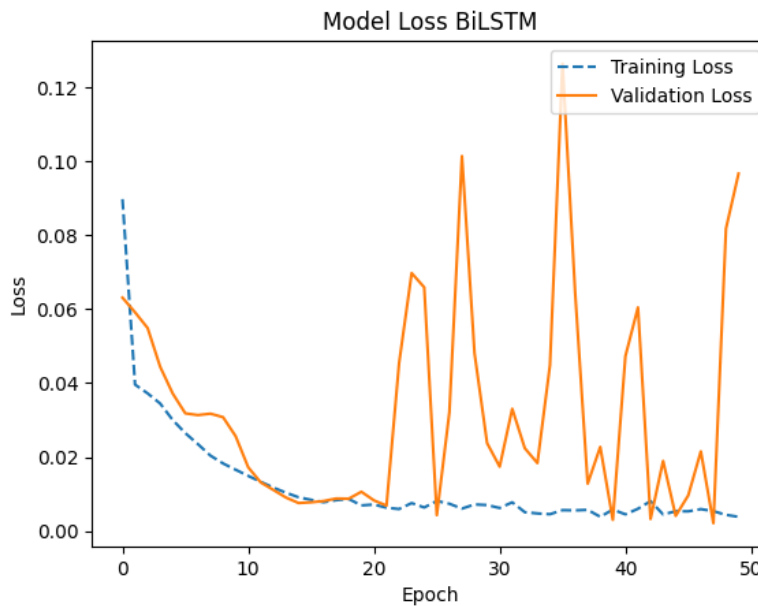
Gambar 2. Grafik Training Loss dan Validation Loss Model LSTM tanpa Early Stopping

Gambar 4 menunjukkan kurva *loss* dari model LSTM tanpa penambahan *early stopping* selama 50 *epoch* pelatihan, baik untuk data pelatihan (*train*) maupun data pengujian (*test*). Kurva *train loss* memperlihatkan penurunan yang konsisten dan stabil dari *epoch* ke-0 hingga mendekati *epoch* ke-50. Namun, pada *test loss* terlihat sangat fluktuatif, terutama setelah *epoch* ke-20. Meskipun pada awal pelatihan (sekitar *epoch* ke-0 hingga *epoch* ke-15), *test loss* masih mengikuti tren penurunan dari *train loss*, setelah itu *test loss* mulai menunjukkan ketidakstabilan yang tajam dengan lonjakan-lonjakan besar di berbagai titik *epoch*. Fluktuasi besar ini mengindikasikan terjadinya *overfitting* dimana model terlalu menyesuaikan diri terhadap data pelatihan hingga kehilangan kemampuan generalisasi terhadap data uji yang belum pernah dilihat sebelumnya.



Gambar 3. Grafik Training Loss dan Validation Loss Model BiLSTM dengan Early Stopping

Gambar 5 merupakan grafik Model Loss BiLSTM dengan *early stopping* yang menunjukkan model BiLSTM selama proses pelatihan terhadap data pelatihan dan data validasi. Pada awal pelatihan (sekitar *epoch* ke-0 hingga ke-10), baik nilai *training loss* maupun *validation loss* mengalami penurunan yang signifikan. Namun, setelah melewati *epoch* ke-15, terlihat bahwa nilai *validation loss* mulai mengalami fluktuasi dan menjadi tidak stabil, meskipun nilai *training loss* terus menurun dan menjadi sangat kecil. Ketidakstabilan ini bisa disebabkan oleh kompleksitas arsitektur BiLSMT yang tinggi, variasi distribusi data validasi, atau pengaturan *batch size* yang kurang optimal. Dengan menggunakan *early stopping*, proses pelatihan secara otomatis dihentikan ketika tidak terjadi perbaikan pada nilai *validation loss* dalam jumlah *epoch* tertentu. Berdasarkan grafik, pelatihan dihentikan pada sekitar *epoch* ke-29, yaitu sebelum *validation loss* meningkat lebih tinggi, sehingga *overfitting* dapat dicegah.



Gambar 4. Grafik Training Loss dan Validation Loss Model BiLSTM tanpa Early Stopping

Gambar 6 grafik Model Loss BiLSTM tanpa penggunaan *early stopping* menunjukkan nilai *training loss* dan *validation loss* berubah selama proses pelatihan hingga 50 *epoch*. Terlihat bahwa *training loss* secara konsisten menurun dan cenderung stabil mendekati nol. Namun, pola yang terlihat pada *validation loss* berbeda. Pada awal pelatihan (sekitar *epoch* 0 hingga 15), *validation loss* juga mengalami penurunan, tetapi setelah itu grafik *validation loss* mulai menunjukkan fluktuasi besar dan tidak stabil, bahkan mengalami kenaikan tajam di beberapa titik. Hal ini mengindikasikan model telah memasuki fase *overfitting*. Karena tidak menggunakan *early stopping*, pelatihan tetap dilanjutkan hingga *epoch* ke-50 meskipun *validation loss* sudah menunjukkan pola tidak stabil sejak pertengahan pelatihan.

TABEL 4
 PERBANDINGAN LSTM DAN BiLSTM

Model Type	Training Mode	True Negative (TN)	False Positive (FP)	False Negative (FN)	True Positive (TP)	Accuracy	Precision	Recall	F1-Score
LSTM	With ES	16,750	274	160	93	98.41%	47.91%	99.60%	64.70%
LSTM	Without ES	16,993	31	3	250	99.80%	88.97%	98.81%	93.63%
BiLSTM	With ES	17,015	9	3	250	99.93%	96.53%	98.81%	97.66%
BiLSTM	Without ES	16,799	225	1	252	98.69%	52.83%	99.60%	69.04%

Tabel 4 menunjukkan perbandingan evaluasi kinerja model LSTM dan BiLSTM yang dilatih dengan dan tanpa *Early Stopping* (ES) untuk klasifikasi kejadian kabut. Dari hasil yang diperoleh, terlihat bahwa setiap model memiliki kelebihan dan kekurangannya masing-masing. Model LSTM dengan *early stopping* menunjukkan kemampuan sangat tinggi dalam mengenali kejadian kabut, ditandai dengan nilai *recall* sebesar 99,60%. Namun, model ini memiliki *precision* yang rendah (47,91%), yang mengindikasikan banyak prediksi kabut yang sebenarnya tidak terjadi (*false positive* tinggi). Hal ini menyebabkan nilai *F1-Score* model hanya mencapai 64,70%. Sebaliknya, LSTM tanpa *early stopping* mampu memberikan hasil yang jauh lebih seimbang, dengan *precision* sebesar 88,97%, *recall* 98,81%, dan *F1-Score* 93,63%.

Sementara itu, BiLSTM dengan *early stopping* tampil sebagai model dengan performa terbaik di antara keempatnya. Model ini mencatatkan akurasi 99,93%, *precision* 96,53%, *recall* 98,81%, dan *F1-Score* 97,66%. Hasil ini menunjukkan bahwa penggunaan BiLSTM dengan *early stopping* tidak hanya efektif dalam mencegah *overfitting*, tetapi juga mampu mempertahankan performa klasifikasi yang sangat tinggi dan seimbang. Di sisi lain, BiLSTM tanpa *early stopping* justru menunjukkan penurunan performa. Meskipun *recall*-nya tetap tinggi (99,60%), *precision*-nya turun drastis menjadi 52,83%, dengan *F1-Score* 69,04%, yang mengindikasikan bahwa model menjadi terlalu agresif dalam memprediksi kabut dan menghasilkan banyak *false positive*.

IV. SIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa penggunaan metode *Bidirectional Long Short-Term Memory* (BiLSTM) dengan penerapan *early stopping* merupakan pendekatan terbaik dalam melakukan klasifikasi kejadian kabut di Citeko Bogor. Model ini menunjukkan performa tertinggi dengan akurasi sebesar 99,93%, *precision* 96,53%, *recall* 98,81%, dan *F1-Score* 97,66%, yang menunjukkan bahwa model mampu mendeteksi kejadian kabut dengan baik sekaligus mengurangi kesalahan dalam prediksi. Selain itu, hasil penelitian juga menunjukkan bahwa *early stopping* mempunyai peran penting dalam mencegah *overfitting*, terutama pada model BiLSTM. Model yang tidak menggunakan *early stopping* cenderung mengalami fluktuasi pada *validation loss*, yang berdampak pada menurunnya *precision* dan *F1-Score* meskipun nilai *recall* tetap tinggi. Sementara itu, model LSTM tanpa *early stopping* juga menunjukkan performa yang baik dan seimbang, meskipun masih berada di bawah BiLSTM dengan *early stopping*. Di sisi lain, model LSTM dan BiLSTM dengan *early stopping* memang unggul dalam hal *recall*, tetapi *precision* yang rendah menunjukkan kecenderungan menghasilkan banyak *false positive*. Namun demikian, penelitian ini memiliki beberapa keterbatasan. Pertama, model hanya diuji menggunakan data dari satu stasiun meteorologi Citeko Bogor, sehingga belum dapat disimpulkan apakah model ini akan bekerja sebaik di lokasi lain dengan karakteristik cuaca yang berbeda. Kedua, evaluasi model dilakukan dengan pendekatan *train-test split* (80:20) yang meskipun sudah representatif untuk data besar, belum seobjektif pendekatan *k-fold cross validation*. Ketiga, proses tuning *hyperparameter* dilakukan secara manual tanpa optimisasi otomatis. Untuk pengembangan lebih lanjut, disarankan agar penelitian dilakukan dengan cakupan lokasi yang lebih luas, menggunakan *cross-validation* untuk evaluasi model dan menerapkan teknik optimisasi otomatis dalam proses tuning parameter.

DAFTAR PUSTAKA

- [1] WMO, "Definition: Fog: A suspension of very small, usually microscopic water droplets in the air, reducing visibility at the Earth's surface.," World Meteorological Organization. [Online]. Available: <https://cloudatlas.wmo.int/en/definitions-and-descriptions-of-meteors.html>
- [2] R. D. O. Colabone, A. L. Ferrari, F. A. da S. Vecchia, and A. R. B. Tech, "Application of artificial neural networks for fog forecast," *J. Aerosp. Technol. Manag.*, vol. 7, no. 2, pp. 240–246, 2015.
- [3] D. K. E. Smith *et al.*, "Forecasts of fog events in northern India dramatically improve when weather prediction models include irrigation effects," *Commun. Earth Environ.*, vol. 5, no. 1, 2024.
- [4] C. Castillo-Botón *et al.*, "Machine learning regression and classification methods for fog events prediction," *Atmos. Res.*, vol. 272, 2022.
- [5] S. Sharma, K. Bajaj, P. Deshpande, A. Bhattacharya, and S. Tripathi, "Short-Term Fog Forecasting using Meteorological Observations at Airports in North India," in *ACM International Conference Proceeding Series*, 2024.
- [6] A. Shankar and B. C. Sahana, "Early warning of low visibility using the ensembling of machine learning approaches for aviation services at Jay Prakash Narayan International (JPNI) Airport Patna," *SN Appl. Sci.*, vol. 5, no. 5, 2023.
- [7] N. G. Dhangar, A. N. Parde, R. Ahmed, D. S. V. V. D. Prasad, and D. M. Lal, "Fog nowcasting over the IGI airport, New Delhi, India using decision tree," *Mausam*, vol. 73, no. 4, 2022.
- [8] S. Kim, C. Rickard, J. Hernandez-Vazquez, and D. Fernandez, "Early Night Fog Prediction Using Liquid Water Content Measurement in the Monterey Bay Area," *Atmosphere (Basel)*, vol. 13, no. 8, 2022.
- [9] W. Li, X. Yang, G. Yuan, and D. Xu, "ABCNet: A comprehensive highway visibility prediction model based on attention, Bi-LSTM and CNN," *Math. Biosci. Eng.*, vol. 21, no. 3, 2024.
- [10] R. S. Pontoh *et al.*, "Jakarta Pandemic to Endemic Transition: Forecasting COVID-19 Using NNAR and LSTM," *Appl. Sci.*, vol. 12, no. 12, 2022.
- [11] N. Afrianto, D. H. Fudholi, and S. Rani, "Prediksi Harga Saham Menggunakan BiLSTM dengan Faktor Sentimen Publik," *J. RESTI (Rekayasa Sist. dan Teknol. Informatika)*, vol. 6, no. 1, pp. 41–46, 2022.
- [12] R. Dewi, Prawito, and H. Harsa, "Fog prediction using artificial intelligence: A case study in Wamena Airport," *J. Phys. Conf. Ser.*, vol. 1528, no. 1, 2020.
- [13] BMKG, "Peraturan Kepala BMKG Nomor SK.38/KT.104/KB/BMG-06 Tahun 2006 tentang Tata Cara Tetap Pelaksanaan Pengamatan, Penyandian, Pelaporan dan Pengarsipan Data Meteorologi Permukaan." [Online]. Available: [https://jdih.bmkg.go.id/common/dokumen/LAMPIRAN II SK.38 TAHUN 2006.pdf](https://jdih.bmkg.go.id/common/dokumen/LAMPIRAN%20II%20SK.38%20TAHUN%202006.pdf)
- [14] M. Yang and J. Wang, "Adaptability of Financial Time Series Prediction Based on BiLSTM," *Procedia Comput. Sci.*, vol. 199, pp. 18–25, 2021.

- [15] S. Ruuska, W. Hämäläinen, S. Kajava, M. Mughal, P. Matilainen, and J. Mononen, "Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle," *Behav. Processes*, vol. 148, pp. 56–62, 2018.
- [16] C. Olah, "Understanding LSTM Networks." [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [17] H. Hartono, "BUMI Stock Price Prediction Using Long Short Term Memory (LSTM) with Three Hyperparameter Tuning Regression," *Quant. Econ. Manag. Stud.*, vol. 3, no. 6, pp. 1008–1017, 2022.
- [18] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," *Proc. - 2019 IEEE Int. Conf. Big Data, Big Data 2019*, pp. 3285–3292, 2019.
- [19] K. U. Jaseena and B. C. Kovoov, "Decomposition-based hybrid wind speed forecasting model using deep bidirectional LSTM networks," *Energy Convers. Manag.*, vol. 234, no. 2020, p. 113944, 2021.
- [20] Y. Karyadi, "Prediksi Kualitas Udara Dengan Metoda LSTM, Bidirectional LSTM, dan GRU," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 1, pp. 671–684, 2022.
- [21] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values," *Transp. Res. Part C Emerg. Technol.*, vol. 118, no. 2014, 2020.
- [22] M. Gabor, "Bidirectional LSTM (BiLSTM) Model." [Online]. Available: [https://www.gabormelli.com/RKB/Bidirectional_LSTM_\(BiLSTM\)_Model](https://www.gabormelli.com/RKB/Bidirectional_LSTM_(BiLSTM)_Model)
- [23] Z. Hameed and B. Garcia-Zapirain, "Sentiment Classification Using a Single-Layered BiLSTM Model," *IEEE Access*, vol. 8, pp. 73992–74001, 2020.