

**ANALISIS KINERJA ALGORITMA LSTM DAN CNN UNTUK
FORECASTING KELEMBABAN UDARA DI KOTA MEDAN**

SKRIPSI

DISUSUN OLEH

MUHAMMAD ELFIKRY

NPM. 2009010121



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2024

**ANALISIS KINERJA ALGORITMA LSTM DAN CNN UNTUK
FORECASTING KELEMBABAN UDARA DI KOTA MEDAN**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
(S.Kom) dalam Program Studi Sistem Informasi pada Fakultas Ilmu Komputer
dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

MUHAMMAD ELFIKRY

NPM. 2009010121

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2024

LEMBAR PENGESAHAN

Judul Skripsi : ANALISIS KINERJA ALGORITMA LSTM DAN CNN
UNTUK FORECASTING KELEMBABAN UDARA DI
KOTA MEDAN
Nama Mahasiswa : MUHAMMAD ELFIKRY
NPM : 2009010121
Program Studi : SISTEM INFORMASI

Menyetujui
Komisi Pembimbing

Dr. Al-Khowarizmi, S.Kom., M.Kom
NIDN. 0127099201

Ketua Program Studi

Martiano S.Pd, S.Kom., M.Kom
NIDN. 0128029302

Dekan



Dr. Al-Khowarizmi, S.Kom., M.Kom.
NIDN. 0127099201

PERNYATAAN ORISINALITAS

**ANALISIS KINERJA ALGORITMA LSTM DAN CNN UNTUK
FORECASTING KELEMBABAN UDARA DI KOTA MEDAN**

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Agustus 2024

Yang membuat pernyataan



Muhammad Elfikry

NPM. 2009010121

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Muhammad Elfikry
NPM : 2009010121
Program Studi : Sistem Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

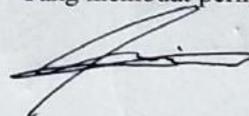
**ANALISIS KINERJA ALGORITMA LSTM DAN CNN UNTUK
FORECASTING KELEMBABAN UDARA DI KOTA MEDAN**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Agustus 2024

Yang membuat pernyataan



Muhammad Elfikry

NPM. 2009010121

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Muhammad Elfikry
Tempat dan Tanggal Lahir : Palembang, 06 Februari 2001
Alamat Rumah : Jl. Jalak 19 No. 505 Kota Medan
Telepon/Faks/HP : +6285156067809
E-mail : melfikry@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD : Taman Harapan Medan TAMAT: 2012
SMP : Negeri 12 Medan TAMAT: 2015
SMA : An – Nizam Medan TAMAT: 2018

KATA PENGANTAR



Puji syukur atas kehadiran Allah SWT, berkat limpahan rahmat, hidayah, dan karunianya, penulis dapat menyelesaikan skripsi dengan judul “**ANALISIS KINERJA ALGORITMA LSTM DAN CNN UNTUK FORECASTING KELEMBABAN UDARA DI KOTA MEDAN**”. Skripsi ini adalah salah satu dari beberapa persyaratan untuk menyelesaikan pendidikan dan memperoleh gelar sarjana pada program studi S1 Sistem Informasi di Universitas Muhammadiyah Sumatera Utara.

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Bapak Martiano, S.Pd., S.Kom., M.Kom Ketua Program Studi Sistem Informasi.
4. Ibu Yoshida Sary, S.E., S.Kom., M.Kom Sekretaris Program Studi Sistem Informasi.
5. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom selaku Dosen Pembimbing
6. Papa tercinta, bapak Ir. Elfian, terima kasih sebesar-besarnya penulis ucapkan karena telah menjadi panutan dalam mengenalkan arti kasih sayang dan pengorbanan. Kehadiran papa telah menjadi sumber semangat penulis dalam setiap langkah perjalanan akademik penulis.
7. Mama tercinta, ibu Dahlia MS, terima kasih sebesar-besarnya penulis ucapkan atas cinta, doa, dan dukungan tanpa batas, baik moral maupun material, yang selalu diberikan dalam setiap langkah saya.
8. Kepada abang dan kakak tersayang, Bang Ahmad Elfarisy, S.E dan Kak Elfina Nazlia, penulis sampaikan terima kasih Atas segala nasihat, semangat, serta dukungan yang tak henti-hentinya diberikan, yang telah menjadi inspirasi dan dorongan dalam menghadapi berbagai tantangan selama proses ini.

ANALISIS KINERJA ALGORITMA LSTM DAN CNN UNTUK FORECASTING KELEMBABAN UDARA DI KOTA MEDAN

ABSTRAK

Kelembaban udara adalah parameter krusial dalam cuaca dan iklim yang memengaruhi kesehatan, kenyamanan, serta sektor ekonomi seperti pertanian dan industri. Kota Medan, dengan variasi kelembaban yang signifikan, membutuhkan model prediksi yang akurat untuk mendukung perencanaan dan tindakan preventif. Penelitian ini bertujuan untuk membandingkan kinerja algoritma Long Short-Term Memory (LSTM) dan Convolutional Neural Network (CNN) dalam melakukan forecasting kelembaban udara di Kota Medan. Kedua model dilatih menggunakan data kelembaban udara historis, dengan evaluasi kinerja berdasarkan metrik Mean Squared Error (MSE) dan Root Mean Squared Error (RMSE). Hasil penelitian menunjukkan bahwa model LSTM unggul dalam menangkap pola jangka panjang kelembaban udara. Temuan ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem prediksi kelembaban udara yang lebih adaptif, yang bermanfaat untuk perencanaan manajemen sumber daya air, pertanian, kesehatan masyarakat, dan mitigasi bencana di Kota Medan.

Kata Kunci: LSTM, CNN, Kelembaban Udara, Forecasting, Deep Learning, Kota Medan.

PERFORMANCE ANALYSIS OF LSTM AND CNN ALGORITHMS FOR FORECASTING AIR HUMIDITY IN MEDAN CITY

ABSTRACT

Humidity is a crucial parameter in weather and climate that affects health, comfort, and economic sectors such as agriculture and industry. Medan City, with significant variations in humidity, requires an accurate prediction model to support planning and preventive actions. This research aims to compare the performance of Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) algorithms in forecasting air humidity in Medan City. Both models were trained using historical air humidity data, with performance evaluation based on Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics. The results show that the LSTM model excels in capturing long-term patterns of air humidity. The findings are expected to contribute to the development of a more adaptive air humidity prediction system, which is beneficial for water resources management planning, agriculture, public health, and disaster mitigation in Medan City.

Keywords: LSTM, CNN, Air Humidity, Forecasting, Deep Learning, Medan City

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	vii
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I.....	1
PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	5
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian	6
BAB II	7
LANDASAN TEORI.....	7
2.1. Landasan Teori	7
2.1.1. Kelembaban Udara.....	7
2.1.2. Time Series	7
2.1.3. Forecasting	9
2.1.4. Machine Learning	9
2.1.5. Artificial Neural Network.....	11
2.1.6. Deep Learning	13
2.1.7. Long Short-Term Memory	15
2.1.8. Convolutional Neural Network.....	17
2.1.9. Python	19
2.1.10. Tensorflow.....	19
2.1.11. Matplotlib	20
2.1.12. Google Colab.....	21
2.1.13. Visual Studio Code	22
2.1.14. Penelitian Terdahulu	23
BAB III.....	26
METODOLOGI PENELITIAN.....	26
3.1. Gambaran Umum Objek penelitian	26

3.2.	Data collection.....	28
3.3.	Exploratory Data Analysis (EDA).....	29
3.4.	Data Preprocessing	30
3.5.	Model Architecture	30
3.5.1.	Long Short-Term Memory Architecture	31
3.5.2.	Convolutional Neural Network Architecture.....	32
3.6.	Model Development	33
3.6.1.	Model Building.....	33
3.6.2.	Model Compilation	33
3.6.3.	Model Training dan Evaluation	34
3.7.	Model Testing.....	35
BAB IV	37
HASIL DAN PEMBAHASAN	37
4.1.	Pengolahan Data.....	37
4.4.	Implementasi Model LSTM dan CNN dengan Tensorflow	45
4.4.1.	Model LSTM	45
4.4.2.	Model CNN	46
4.5.	Proses Pelatihan Modal.....	47
4.6.	Analisis Performa dan Evaluasi Model.....	48
4.7.	Analisis dan Interpretasi Hasil.....	49
BAB V	51
PENUTUP	51
5.1.	Kesimpulan.....	51
5.2.	Saran	51
DAFTAR PUSTAKA	53
LAMPIRAN	57

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu.....	23
Table 4. 1 Ringkasan Model LSTM	45
Table 4. 2 Ringkasan Model CNN	46
Table 4. 3 Tabel Perbandingan Metrik Evaluasi	49

DAFTAR GAMBAR

Gambar 2. 1 Skema Artificial Intelligence dan Machine Learning	10
Gambar 2. 2 Arsitektur Artificial Neural Network.....	12
Gambar 2. 3 Pola Dasar Artificial Neural Networks	13
Gambar 2. 4 Taksonomi AI.....	14
Gambar 2. 5 Ilustrasi Deep Learning	15
Gambar 2. 6 Struktur LSTM	16
Gambar 2. 7 Struktur Convolutional Neural Network.....	18
Gambar 3. 1 Alur Penelitian.....	27
Gambar 3. 2 Dataset yang telah diunduh dari situs web BMKG	28
Gambar 3. 3 Visualisasi Box Plot Untuk Mendeteksi Outlier Pada Dataset	29
Gambar 3. 4 Layer Long Short-Term Memory	32
Gambar 3. 5 Layer Convolutional Neural Network	32
Gambar 4. 1 Plot Outlier pada Dataset Menggunakan Seaborn.....	38
Gambar 4. 2 Kode untuk Menyimpan dan Memplot Dataset yang Bebas dari Outlier	39
Gambar 4. 3 Plot Dataset Tanpa Outlier	39
Gambar 4. 4 Kode untuk tahapan standarisasi data	40
Gambar 4. 5 Kode Pembagian Dataset	40
Gambar 4. 6 Kode Teknik Windowing.....	41
Gambar 4. 7 Kode Untuk Membangun Arsitektur Model LSTM	46
Gambar 4. 8 Kode Untuk Membangun Arsitektur Model CNN	47
Gambar 4. 9 Kode Pelatihan Model LSTM	47
Gambar 4. 10 Kode Pelatihan Model CNN.....	48

Gambar 4. 11 Loss Pada Model LSTM & CNN	48
Gambar 4. 12 Plot Perbandingan Metrik Evaluasi MSE & RMSE	49
Gambar 4. 13 Plot Perbandingan data Aktual Dengan Hasil Forecasting Pada Model LSTM.....	50
Gambar 4. 14 Plot Perbandingan data Aktual Dengan Hasil Forecasting Pada Model CNN	50

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Kelembaban udara merupakan salah satu parameter penting dari cuaca dan iklim yang mempengaruhi berbagai aspek kehidupan manusia dan lingkungan. Secara umum, kelembaban udara dapat didefinisikan sebagai kandungan uap air yang ada dalam udara. Terdapat beberapa metrik untuk mengukur kelembaban udara, antara lain kelembaban mutlak, kelembaban nisbi (relatif), atau defisit tekanan uap air. Kelembaban udara menjadi faktor yang penting dalam menentukan kenyamanan iklim dan kesehatan manusia. Selain itu, kelembaban udara adalah faktor lingkungan yang sangat mempengaruhi penyebaran penyakit, termasuk penyakit endemik seperti Demam Berdarah (DBD). Tingkat kelembaban udara yang ekstrim juga sangat mempengaruhi kesejahteraan manusia, termasuk kualitas udara, sektor pertanian, industri makanan, dan potensi kebakaran hutan (Intan et al., 2021; Alizkan, 2017).

Kelembaban udara yang tinggi atau rendah dapat memberikan dampak yang signifikan terhadap kesehatan manusia. Misalnya, kelembaban udara yang tinggi dapat memperburuk kondisi asma dan meningkatkan risiko infeksi saluran pernapasan. Di sisi lain, kelembaban udara yang rendah dapat menyebabkan kulit menjadi kering dan iritasi. Selain itu, kelembaban udara juga berpengaruh pada kenyamanan termal, di mana kelembaban yang tinggi dapat meningkatkan perasaan panas dan tidak nyaman bagi manusia.

Dampak kelembaban udara tidak hanya terbatas pada individu, tetapi juga mencakup dampak yang lebih luas terhadap sektor-sektor penting dalam perekonomian dan lingkungan. Di sektor pertanian, kelembaban udara yang optimal sangat penting untuk pertumbuhan tanaman dan hasil panen yang baik. Sebaliknya, kelembaban udara yang tidak

sesuai dapat mengakibatkan penurunan produksi pertanian dan berdampak pada ketersediaan pangan. Di sektor industri, pengendalian kelembaban udara diperlukan untuk menjaga kualitas produk, seperti dalam industri makanan dan farmasi, serta untuk mendukung proses produksi yang efisien. Selain itu, kelembaban udara juga memiliki implikasi signifikan dalam konteks lingkungan dan keberlanjutan. Tingkat kelembaban udara yang ekstrem dapat mempengaruhi kualitas udara dan meningkatkan risiko kebakaran hutan (Intan et al., 2021).

Area kota menjadi salah satu objek yang cukup menarik karena sering terjadi perbedaan cuaca secara signifikan. Untuk masyarakat Kota Medan, yang mayoritas bekerja sebagai karyawan, supir transportasi, dan pedagang, dibutuhkan informasi yang akurat seperti Forecasting(peramalan) kelembaban udara, karena tidak ada yang tahu pasti apa yang akan terjadi di masa depan (Alfandi & Sihite, 2022). Kota Medan, yang menjadi fokus penelitian ini, merupakan contoh dari kota dengan variasi kelembaban udara yang signifikan sepanjang tahun. Variasi ini menuntut adanya model prediksi yang akurat untuk membantu masyarakat dan pemerintah dalam mengambil tindakan yang tepat. Forecasting dapat memudahkan dalam menyelenggarakan pendekatan analisis terhadap pola dan tingkah laku data pada masa lalu, untuk mendapatkan informasi kelembaban udara di masa depan dengan lebih akurat.

Forecasting atau peramalan kelembaban udara merupakan suatu proses penting dalam upaya memprediksi keadaan masa depan berdasarkan data historis yang ada. Dalam konteks perubahan iklim global dan peningkatan pola cuaca yang tidak menentu, prediksi yang akurat tentang kelembaban udara menjadi semakin penting. Teknik peramalan memungkinkan pemerintah, lembaga riset, dan sektor swasta untuk mengambil langkah-langkah preventif yang tepat, mempersiapkan infrastruktur yang diperlukan, serta merencanakan kegiatan ekonomi dengan lebih baik. Forecasting (peramalan) memiliki berbagai teknik yang sangat beragam untuk mendapatkan informasi di masa depan

berdasarkan data historis. Salah satu pendekatannya adalah penggunaan kecerdasan buatan (AI), khususnya dengan metode deep learning (Intan et al., 2021).

Deep learning memanfaatkan Jaringan Saraf Tiruan (Artificial Neural Network) yang kompleks dengan banyak lapisan untuk menganalisis data secara mendalam dan membuat prediksi yang akurat. Jaringan saraf tiruan sebagian besar telah cukup handal dalam pemecahan masalah selama beberapa tahun terakhir. Jaringan saraf tiruan terinspirasi oleh otak manusia, di mana neuron saling terkoneksi secara nonlinier, jaringan saraf tiruan membangun koneksi antar neuron dalam suatu jaringan. Paradigma pengolahan informasi ini meniru sistem saraf biologis, seperti yang terjadi dalam proses informasi pada otak manusia. Cara kerja jaringan saraf tiruan menyerupai cara kerja manusia, di mana proses pembelajaran terjadi melalui contoh.

Keuntungan dari jaringan saraf tiruan adalah kemampuannya untuk menangani data, mendeteksi tren, dan memprediksi pola yang tidak diberikan selama pelatihan. Oleh karena itu, dengan menggunakan deep learning yang memanfaatkan jaringan saraf tiruan, model forecasting dapat belajar dan mengekstraksi pola yang kompleks dari data historis, memungkinkan untuk melakukan peramalan yang lebih baik dan lebih adaptif terhadap perubahan cuaca (Andrian & Wayahdi, 2014).

LSTM adalah jenis jaringan saraf yang khusus dirancang untuk mengatasi masalah dalam pengolahan data deret waktu yang memiliki dependensi jangka panjang. Algoritma ini mampu mempertahankan informasi dalam jangka waktu yang panjang, sehingga sangat cocok untuk memprediksi data yang bersifat temporal, seperti kelembaban udara. LSTM bekerja dengan menggunakan memori internal untuk menyimpan informasi penting yang dapat digunakan dalam proses prediksi selanjutnya. Long Short-Term Memory (LSTM) merupakan metode untuk melakukan prediksi yang telah diuji dalam berbagai proses penelitian. LSTM merupakan arsitektur dari Recurrent Neural Network (RNN) yang dapat digunakan untuk memproses data sequential sehingga dapat digunakan untuk prediksi data yang bersifat time series (Lattifia et al., 2022).

Di sisi lain, CNN adalah jenis jaringan saraf yang sering digunakan untuk analisis citra dan data yang memiliki dimensi spasial. Meskipun CNN lebih dikenal dalam bidang pengolahan citra, algoritma ini juga telah diterapkan dalam berbagai aplikasi time series dengan hasil yang memuaskan. CNN mampu mengenali pola dan fitur penting dalam data melalui proses konvolusi, yang memungkinkan model untuk mempelajari representasi data yang lebih baik (Widiputra et al., 2021).

Penerapan LSTM dan CNN dalam forecasting kelembaban udara memungkinkan untuk memanfaatkan informasi historis dengan lebih efisien dan meningkatkan kemampuan prediksi terhadap perubahan cuaca yang cepat dan kompleks. Dengan membandingkan kinerja dari kedua algoritma ini, penelitian ini bertujuan untuk mengidentifikasi model yang paling efektif dalam memberikan prediksi kelembaban udara yang akurat.

Dengan membandingkan kinerja dari kedua algoritma tersebut, diharapkan dapat diperoleh model yang paling efektif dan akurat untuk digunakan dalam forecasting kelembaban udara di wilayah kota Medan. Prediksi kelembaban udara yang akurat dapat membantu pemerintah dan industri dalam mengambil tindakan preventif dan perencanaan yang lebih baik, seperti manajemen sumber daya air, pertanian, kesehatan masyarakat, dan mitigasi bencana.

Oleh karena itu, penelitian ini tidak hanya akan berkontribusi dalam bidang ilmu pengetahuan dan teknologi, tetapi juga memiliki implikasi praktis yang signifikan. Dengan menggunakan deep learning yang memanfaatkan jaringan saraf tiruan seperti LSTM dan CNN, diharapkan model forecasting kelembaban udara dapat belajar dan mengekstraksi pola yang kompleks dari data historis, sehingga memberikan prediksi yang lebih baik dan adaptif terhadap perubahan cuaca. Hal ini akan meningkatkan kemampuan masyarakat dan pemerintah untuk merespons perubahan lingkungan dengan lebih efektif dan efisien.

1.2. Rumusan Masalah

Rumusan masalah dalam penelitian ini yaitu bagaimana kinerja algoritma Long Short-Term Memory (LSTM) dan Convolutional Neural Network (CNN) dalam melakukan forecasting kelembaban udara di kota Medan, serta manakah di antara kedua algoritma tersebut yang memiliki kinerja terbaik untuk melakukan forecasting kelembaban udara di kota Medan?

1.3. Batasan Masalah

Terdapat beberapa batasan permasalahan dalam penelitian ini yaitu:

- a. Penelitian ini hanya membandingkan kinerja dari algoritma LSTM dan CNN dalam melakukan forecasting kelembaban udara di Kota Medan.
- b. Data yang digunakan terbatas pada data historis kelembaban udara di kota Medan.
- c. Evaluasi kinerja menggunakan metrik evaluasi MSE dan RMSE untuk melakukan perbandingan kinerja forecasting kelembaban udara.
- d. Penulis tidak memasukan faktor-faktor lain yang dapat mempengaruhi kelembaban udara seperti curah hujan atau suhu udara.

1.4. Tujuan Penelitian

- a. Membuat model *deep learning* menggunakan algoritma LSTM dan model deep learning menggunakan algoritma CNN untuk peramalan kelembaban udara di kota Medan.
- b. Menganalisis kinerja algoritma LSTM dalam memprediksi kelembaban udara di kota Medan.
- c. Menganalisis kinerja algoritma CNN dalam memprediksi kelembaban udara di kota Medan.

- d. Mengetahui Kinerja yang paling efektif antara model deep learning menggunakan algoritma LSTM dan model deep learning menggunakan algoritma CNN untuk forecasting kelembaban udara di Kota Medan.

1.5. Manfaat Penelitian

- a. Kontribusi Ilmiah: Memberikan kontribusi dalam pengembangan metodologi forecasting menggunakan teknik deep learning untuk keperluan prediksi kelembaban udara.
- b. Praktis: Memberikan rekomendasi tentang algoritma yang paling cocok dan akurat untuk digunakan oleh pemerintah daerah atau lembaga terkait dalam merencanakan kebijakan lingkungan, pertanian, dan kesehatan masyarakat di kota Medan.
- c. Pengembangan Teknologi: Mendorong pengembangan teknologi forecasting yang lebih canggih dan adaptif terhadap perubahan lingkungan dan cuaca.
- d. Peningkatan Responsif Masyarakat: Memberikan informasi yang lebih akurat dan tepat waktu kepada masyarakat mengenai kondisi kelembaban udara, sehingga meningkatkan kesadaran dan responsif terhadap potensi risiko lingkungan.

BAB II

LANDASAN TEORI

2.1. Landasan Teori

2.1.1. Kelembaban Udara

Kelembaban udara adalah jumlah dari uap air yang terkandung dalam udara atau atmosfer. Besarnya tingkat kelembaban udara tergantung pada seberapa banyak uap air yang masuk ke atmosfer, yang dihasilkan dari penguapan air yang ada di lautan, danau, sungai, dan air tanah. Selain itu, penguapan dari tumbuhan yang disebut proses transpirasi juga berkontribusi. Banyaknya air di dalam udara bergantung banyak faktor, antara lain ketersediaan air, sumber uap, suhu udara, tekanan udara, dan angin.

Kelembaban udara dapat diukur dengan berbagai cara, termasuk kelembaban mutlak, kelembaban relatif (nisbi), dan defisit tekanan uap air. Kelembaban udara nisbi memiliki pengertian sebagai nilai perbandingan antara jumlah dari uap air yang ada di udara saat pengukuran (e) dengan jumlah uap air maksimum (e_m) yang bisa ditampung udara pada suhu dan tekanan tersebut (Fadholi, 2013). Persamaan untuk kelembaban udara nisbi adalah sebagai berikut:

$$RH = \frac{e}{e_m} \times 100 \quad (1)$$

Dengan memahami kelembaban udara, kita dapat lebih baik memprediksi kondisi cuaca dan iklim, serta mengantisipasi perubahan lingkungan yang mungkin terjadi. Kelembaban yang cukup besar memberi petunjuk langsung bahwa udara memiliki kandungan uap air yang signifikan, yang penting untuk berbagai fenomena atmosfer (Alizkan, 2017).

2.1.2. Time Series

Time series adalah rangkaian data yang terdiri dari nilai-nilai pengamatan yang diukur selama kurun waktu tertentu dengan interval waktu yang sama. Analisis time series

memungkinkan untuk memahami kejadian di masa lalu dan memprediksi apa yang mungkin terjadi di masa depan. Time series bermanfaat di berbagai bidang, termasuk ekonomi, sains, bisnis, dan Teknik. Time series dapat mengidentifikasi tren, pola musiman, dan fluktuasi berulang yang membantu dalam pengambilan keputusan dan perencanaan strategis yang lebih efektif. Penerapan time series meliputi analisis pasar saham, peramalan cuaca, prediksi penjualan, dan pengelolaan rantai pasokan (Imaduddin et al., 2023).

Secara Umum time series terbagi menjadi dua kelompok besar yaitu univariate dan multivariate.

- a. *Univariate Time Series* adalah analisis data yang hanya melibatkan satu variabel yang diamati sepanjang waktu. Deret waktu ini hanya fokus pada satu aspek dimensi data, dan bagaimana nilainya berubah dari waktu ke waktu. Univariate digunakan ketika ingin membuat suatu prediksi dari satu variabel, terutama jika data historis tersedia untuk variabel tersebut. Teknik ini sangat umum dan mendasar karena sering diterapkan secara umum di berbagai bidang seperti ekonomi, keuangan, prakiraan cuaca, dan prakiraan permintaan dalam manajemen rantai pasokan. Tujuan dari analisis univariate adalah untuk memahami pola dan tren data, memprediksi nilai masa depan, dan menjelaskan faktor-faktor yang mempengaruhi perubahan data.
- b. *Multivariate Time Series* adalah analisis data yang melibatkan beberapa variabel yang diamati secara bersamaan sepanjang waktu. Setiap variabel tidak hanya bergantung pada nilai masa lalunya (historis), tetapi juga memiliki ketergantungan atau hubungan dengan variabel lainnya. Analisis deret waktu multivariate bertujuan untuk memahami hubungan antar variabel, memprediksi nilai masa depan beberapa variabel secara bersamaan, dan menjelaskan faktor-faktor yang mempengaruhi perubahan nilai variabel-variabel tersebut (Al Haromainy et al., 2021).

2.1.3. Forecasting

Forecasting adalah istilah usaha untuk memperkirakan, memprediksi, apa yang akan terjadi pada masa mendatang. Peramalan menjadi sangat penting karena masa depan selalu penuh dengan ketidakpastian, dan waktu antara kesadaran akan suatu perubahan dan pelaksanaan kebijakan untuk merespons perubahan tersebut seringkali memerlukan perencanaan yang matang. *forecasting* memudahkan dalam menganalisis pola dan perilaku data masa lalu, memberikan wawasan yang berharga untuk perencanaan masa depan. Pendekatan ini memungkinkan untuk menyusun dan menerapkan solusi yang lebih pragmatis dan sistematis.

Forecasting tidak hanya membantu dalam merencanakan dan mengelola sumber daya dengan lebih efisien, tetapi juga meningkatkan tingkat keyakinan terhadap keakuratan prediksi yang dibuat. Dengan memahami tren historis dan pola yang berulang, pelaku usaha dan pembuat kebijakan dapat mengambil keputusan yang lebih tepat, terinformasi, dan meminimalkan resiko (Alfandi & Sihite, 2022).

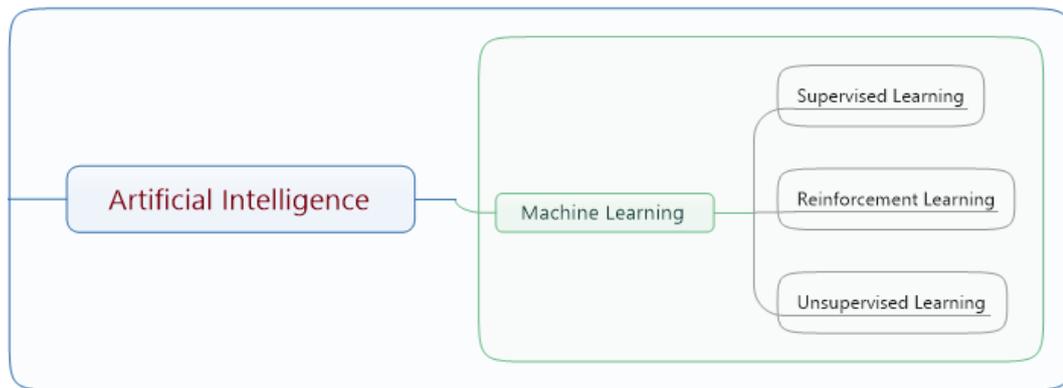
2.1.4. Machine Learning

Machine learning adalah bagian dari kecerdasan buatan (AI) yang memungkinkan sistem komputer menjadi cerdas tanpa aturan yang ditentukan langsung oleh manusia. *Machine learning* memungkinkan untuk mengenali pola dari data yang digunakan sebagai data latih sehingga bisa menghasilkan model yang dapat digunakan untuk melakukan prediksi (regresi) atau pengelompokan data (klasifikasi).

Machine learning membutuhkan data sebagai input untuk menganalisis kumpulan data besar dan menemukan pola tertentu. Dalam *machine learning*, data latih (*training data*) berfungsi untuk melatih algoritma dan data uji (*testing data*) berfungsi untuk mengevaluasi kinerja model saat menghadapi data baru yang belum pernah digunakan dalam pelatihan (Haris et al., 2022).

Machine learning adalah bidang dalam kecerdasan buatan yang berfokus pada pengembangan teknik pemrograman yang belajar dari pengalaman sebelumnya. Bidang ini sering terkait dengan ilmu statistik dan optimasi, karena menggunakan metode tersebut untuk meningkatkan kinerja dan akurasi model yang dibuat (Mujiasih, 2011).

Penelitian terkini mengungkapkan bahwa machine learning terbagi menjadi tiga kategori utama yaitu, Supervised Learning, Unsupervised Learning, dan Reinforcement Learning. Masing-masing kategori memiliki karakteristik, tujuan, dan pendekatan yang berbeda dalam memproses dan menganalisis data. Skema dari keterkaitan artificial intelligence (AI) dan machine Learning dijelaskan pada gambar berikut.



Gambar 2. 1 Skema Artificial Intelligence dan Machine Learning

- a. *Supervised Learning* adalah metode pembelajaran di mana model dilatih menggunakan kumpulan data yang telah diberi label. Data ini terdiri dari pasangan input-output, di mana setiap input memiliki output yang benar yang digunakan sebagai panduan untuk melatih model. Metode ini sangat efektif untuk tugas-tugas klasifikasi, di mana tujuannya adalah untuk mengidentifikasi kategori tertentu dari data, dan regresi, di mana tujuannya adalah untuk memprediksi nilai kontinu. Algoritma yang umum digunakan dalam supervised learning adalah *Linear Regression*, *Decision Trees*, *Random Forest*, *Support Vector Machines*, *k-Nearest Neighbor*, dan *Neural Networks*.

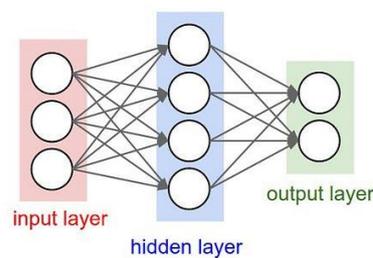
- b. *Unsupervised Learning* adalah metode pembelajaran di mana model dilatih menggunakan data yang tidak diberi label. Tujuan dari metode ini adalah untuk menemukan pola atau struktur tersembunyi dalam data. Teknik ini sangat berguna dalam tugas-tugas clustering, di mana data dikelompokkan berdasarkan kemiripan atau fitur Bersama, dan asosiasi, yang mencari aturan yang menggambarkan hubungan antara variabel dalam data. Algoritma yang sering digunakan dalam unsupervised learning termasuk k-means, DBSCAN, dan algoritma apriori.
- c. *Reinforcement Learning* adalah metode pembelajaran di mana model belajar untuk membuat keputusan dengan berinteraksi dengan lingkungan yang dinamis. Model ini belajar melalui pendekatan trial-and-error, di mana ia mencoba berbagai tindakan dan menerima umpan balik dalam bentuk reward atau penalti. Tujuannya adalah untuk memaksimalkan total reward sepanjang waktu. Pendekatan ini sering digunakan dalam situasi di mana keputusan harus dibuat dalam urutan dan hasil dari setiap keputusan mempengaruhi keadaan lingkungan selanjutnya. Algoritma umum dalam reinforcement learning termasuk Q-learning, SARSA, dan algoritma berbasis penguatan temporal difference. Pada umumnya reinforcement learning digunakan untuk memecahkan masalah optimisasi yang kompleks dan pengambilan keputusan yang adaptif (Roihan et al., 2020).

2.1.5. Artificial Neural Network

Artificial neural networks (ANN) adalah kumpulan elemen yang disusun menjadi sistem yang bekerja secara paralel. Fungsi sistem ini dipengaruhi oleh kekuatan hubungan antar elemen, struktur jaringan, dan proses komputasi dari setiap elemen atau node. Neural networks (NN) adalah struktur pemrosesan data yang bekerja secara paralel dan terdistribusi, terdiri dari elemen-elemen proses yang memiliki memori lokal dan mampu

beroperasi dengan informasi lokal. Setiap elemen proses ini memiliki koneksi keluar tunggal yang dapat bercabang ke beberapa koneksi lain menghasilkan sinyal yang sama dengan keluaran elemen tersebut. Artificial neural networks dirancang sebagai model sistem komputasi yang meniru cara kerja jaringan saraf biologis (Alfandi & Sihite, 2022).

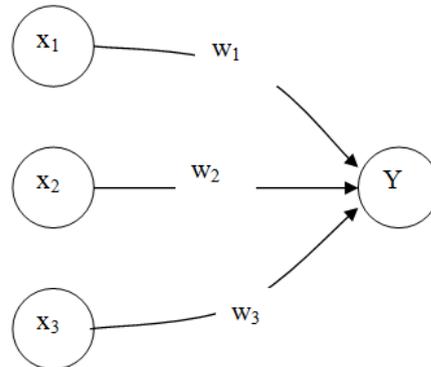
Artificial neural networks adalah model pengolahan informasi yang terinspirasi dari cara kerja sistem saraf biologis, seperti otak manusia. Elemen utama dalam ANN adalah struktur sistem pemrosesan informasi yang terdiri dari banyak elemen pemrosesan atau neuron yang saling terhubung dan bekerja bersama untuk menyelesaikan masalah tertentu. ANN bekerja seperti otak manusia, yaitu belajar melalui contoh. Struktur ANN terdiri dari tiga lapisan: lapisan input, lapisan tersembunyi, dan lapisan output. Pada dasarnya, ANN menerima input, memproses data, dan memberikan output yang terkait dengan input tersebut. Keuntungan ANN adalah kemampuannya untuk mengambil data, mendeteksi tren, dan memprediksi pola yang tidak terlihat selama pelatihan, yang dikenal sebagai generalisasi (Andrian & Wayahdi, 2014; Faraway & Chatfield, 1998).



Gambar 2. 2 Arsitektur Artificial Neural Network

Artificial neural networks memiliki tiga komponen utama yang menentukan cara kerjanya. Pertama adalah pola hubungan antar neuron yang dikenal sebagai arsitektur jaringan yang berfungsi untuk menentukan bagaimana neuron-neuron dalam jaringan terhubung satu sama lain. Kedua adalah metode untuk menentukan bobot penghubung, yang disebut metode pelatihan, berperan dalam mengatur ketebalan hubungan antar neuron berdasarkan data yang dipelajarinya. Ketiga adalah fungsi aktivasi yang bertanggung jawab

untuk mengubah input yang diterima neuron menjadi output yang digunakan untuk proses selanjutnya dalam jaringan. Ketiga elemen ini bekerja sama untuk membentuk dan mengoptimalkan jaringan syaraf tiruan agar mampu memecahkan dengan efisien.



Gambar 2. 3 Pola Dasar Artificial Neural Networks

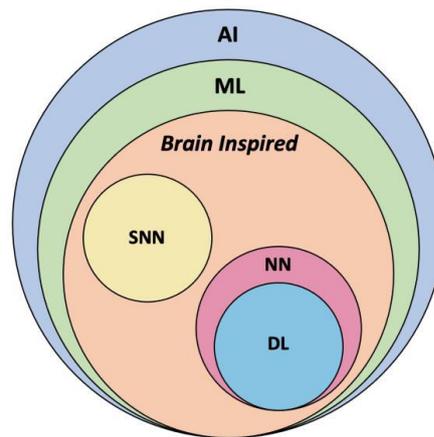
Output Y menerima input dari neuron X_1 , X_2 , dan X_3 dengan bobot hubungan masing-masing adalah W_1 , W_2 , dan W_3 . ketiga input tersebut dijumlahkan menjadi $net = X_1W_1 + X_2W_2 + X_3W_3$ (Riadi & Nurmahaludin, 2012).

2.1.6. Deep Learning

Sejak tahun 1950-an, salah satu cabang dari Artificial Intelligence (AI) yang disebut dengan Machine Learning (ML) telah mengalami perkembangan yang signifikan dan telah di implementasi di berbagai bidang. Neural Network (NN) adalah salah satu bentuk implementasi dari ML, sedangkan Deep Learning (DL) merupakan salah satu implementasi dari neural networks (Muhammad Haris Diponegoro et al., 2021).

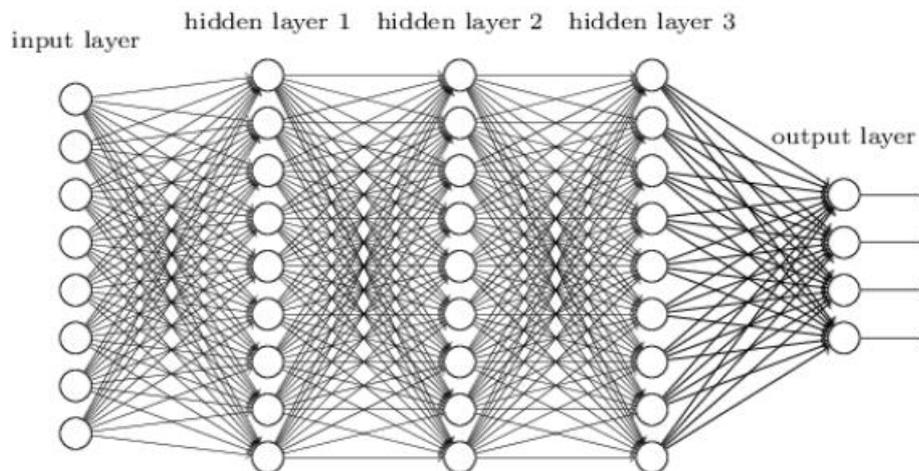
Deep Learning adalah bagian dari machine learning yang menggunakan algoritma yang terinspirasi dari struktur dan fungsi otak manusia yang dikenal sebagai artificial neural networks. Deep learning telah berkembang pesat di berbagai bidang seperti penglihatan komputer, audio, teks, dan banyak lagi. Hampir semua bidang dapat memanfaatkan kemampuan deep learning untuk menyelesaikan masalah yang kompleks (Karyadi, 2022).

Deep Learning adalah penerapan jaringan syaraf tiruan yang menggunakan *deep neural networks* untuk menyelesaikan masalah. Banyak masalah yang bisa diatasi dengan *deep learning*, dan terdapat berbagai metode yang bisa digunakan, seperti *backpropagation*, *Extreme Machine Learning (EML)*, dan *Long Short-Term Memory (LSTM)*. *Deep learning* pada umumnya diterapkan pada dataset besar dan melibatkan proses komputasi yang rumit (Diandra et al., 2022).



Gambar 2. 4 Taksonomi AI

Deep Learning terdiri dari beberapa lapisan yang masing-masing merupakan kumpulan node tempat perhitungan terjadi. Sebuah node input digabung dengan bobot (weight), dan setelah jumlahnya melewati fungsi aktivasi, sinyal akan berlanjut melalui jaringan, mempengaruhi hasil akhir. Perbedaan utama antara deep learning dan neural network adalah jumlah lapisan tersembunyi yang lebih banyak pada deep learning. Jika ada lebih dari tiga lapisan (termasuk input dan output), maka itu memenuhi syarat sebagai deep learning. Secara teknis, deep learning adalah machine learning yang memiliki lebih dari satu lapisan tersembunyi (Rizki et al., 2020).



Gambar 2. 5 Ilustrasi Deep Learning

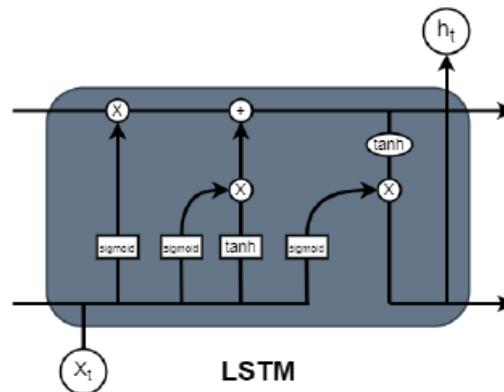
2.1.7. Long Short-Term Memory

Long short-term memory (LSTM) adalah jenis Recurrent Neural Network (RNN) yang telah dimodifikasi dengan menambahkan memory cell untuk menyimpan informasi dalam jangka waktu yang lama (Diandra et al., 2022). Keunggulan dari LSTM adalah kemampuan dalam mengingat sekuens data jangka Panjang yang sulit dicapai dengan teknik fitur tradisional (Rizki et al., 2020).

LSTM dirancang untuk mengatasi masalah vanishing gradient yang sering terjadi pada RNN saat memproses data sekuensial yang panjang dan kompleks (Diandra et al., 2022). LSTM memiliki tiga jenis gate: forget gate, input gate, dan output gate. Forget gate memutuskan informasi mana yang harus dihapus dari sel, input gate menentukan nilai input yang akan di update dalam state memori, dan output gate menentukan bagaimana menghasilkan output setelah memasuki memori sel (Karyadi, 2022).

LSTM dapat digunakan untuk menangani data sequential dan digunakan untuk prediksi data time series. LSTM memiliki kemampuan mendeteksi data yang perlu disimpan dan data yang tidak perlu, berkat tiga lapisan neuron yang disebut gates. Kelebihan utama LSTM dibandingkan RNN adalah kemampuannya mengingat data time

series dan informasi yang memiliki long-term dependency, serta menyimpan informasi terdahulu menggunakan sel-sel di dalam LSTM (Lattifia et al., 2022).



Gambar 2. 6 Struktur LSTM

Keterangan:

Forget Gate: Menggunakan fungsi aktivasi sigmoid untuk menghasilkan vector (forget gate) yang bernilai 0 dan 1. Nilai 0 menunjukkan bahwa informasi pada sel sebelumnya harus dilupakan sepenuhnya, sementara nilai 1 menunjukkan bahwa informasi tersebut harus dipertahankan tanpa perubahan.

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \quad (2)$$

Input Gate: Menerima masukan saat ini dan sebelumnya, kemudian menghasilkan vector (update vector) yang bernilai antara 0 dan 1. Nilai 0 menunjukkan elemen data masukan tidak akan diperbarui dan diabaikan, sedangkan nilai 1 menunjukkan elemen data masukan akan sepenuhnya diperbarui dan disimpan dalam sel memori.

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \quad (3)$$

Setelah melewati input gate, LSTM menggunakan aktivasi tanh untuk menghasilkan vector nilai baru yang akan disimpan dalam sel memori (memory cell). Fungsi ini berguna untuk membantu dalam mengontrol aliran informasi dalam sel memori dengan menghilangkan skala dari nilai yang dipertimbangkan.

$$\tilde{c}_t = \tanh(u_c x_t + w_c h_{t-1} + b_c) \quad (4)$$

Cell state: Proses ini memperbarui nilai pada memory cell sebelumnya dengan nilai baru tersebut. LSTM menggunakan operasi perkalian elemen-wise (element-wise multiplication) antara output dari forget gate dan input gate untuk menggabungkan informasi dari kedua gates tersebut untuk memperbarui nilai pada memory cell.

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (5)$$

Output Gate: Output gate pada LSTM menerima input saat ini dan sebelumnya, kemudian menggunakan fungsi aktivasi sigmoid untuk menghasilkan vector (output vector) yang bernilai antara 0 dan 1. Nilai 0 menunjukkan informasi dalam memory cell akan diabaikan, dan nilai 1 menunjukkan informasi tersebut secara sepenuhnya akan dikeluarkan.

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) \quad (6)$$

Dengan mengalikan Output gate dengan nilai memory cell state yang baru, LSTM akan memberikan hasil akhir dari proses tersebut (Kristiana & Miyanto, 2023).

$$h_t = o_t * \tanh(c_t) \quad (7)$$

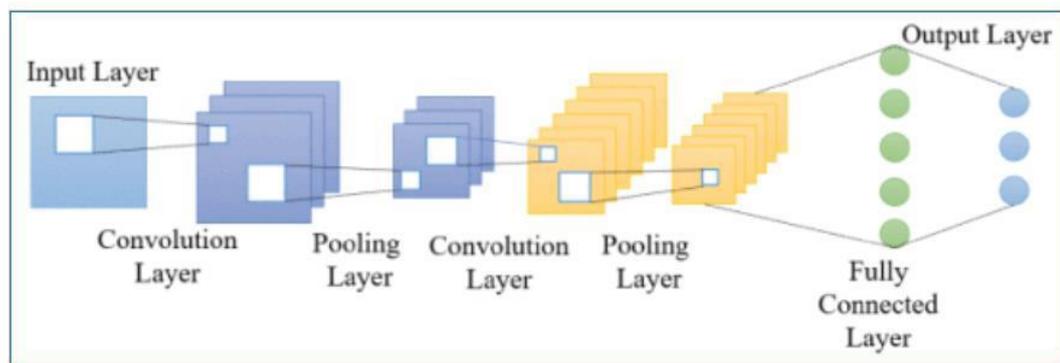
2.1.8. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan pengembangan dari Multilayer Perceptron (MLP) yang dirancang untuk mengolah data dua dimensi. CNN termasuk dalam jenis deep learning karena kedalaman jaringannya yang signifikan dan terdiri dari layer dengan susunan 3D (lebar, tinggi, dan kedalaman). Secara umum, CNN dibagi menjadi dua layer utama: layer ekstraksi dan layer klasifikasi. Layer ekstraksi terletak di awal arsitektur dan terdiri dari neuron yang saling terhubung dengan local region sebelumnya, sedangkan layer klasifikasi terdiri dari neuron yang terhubung penuh dengan layer lainnya (Ihsan, 2021).

CNN didefinisikan sebagai kelas neural network yang memiliki kemampuan khusus untuk memproses data dengan topologi seperti grid. CNN telah terbukti efektif dalam pengenalan gambar, pengelompokan video, analisis dokumen, dan klasifikasi serta prediksi data deret waktu seperti peramalan kelembaban udara. Dengan kemampuan untuk

mengenali dan mengekstrak fitur serta fitur local perception dan weight sharing yang meningkatkan efisiensi pembelajaran, CNN dapat mengatasi masalah komputasi yang kompleks (Widiputra et al., 2021).

Selain untuk deteksi dan pengenalan objek, CNN juga efektif dalam peramalan cuaca termasuk kelembaban udara karena kemampuannya dalam menangkap variasi lingkungan. Arsitektur konvolusinya yang sederhana dan jelas menyederhanakan klasifikasi tanpa memerlukan fitur yang direkayasa. CNN menggunakan operasi linier khusus yang dikenal sebagai konvolusi di setidaknya satu layer jaringannya. Kemampuannya untuk menangkap pemetaan non-linear antara berbagai area seperti ruang fitur dan ruang label menjadikannya sangat efektif dalam deskripsi dan klasifikasi gambar serta peramalan kelembaban udara (Kareem et al., 2021).



Gambar 2. 7 Struktur Convolutional Neural Network

Keterangan:

Input Layer: Lapisan ini memiliki $N \times k$ neurons, k menunjukkan jumlah variasi dari deret waktu input N menyatakan panjang dari setiap deret waktu univariat.

Convolutional Layer: Melakukan operasi konvolusi dengan filter konvolusi pada deret waktu lapisan sebelumnya. Beberapa parameter yang harus ditentukan adalah jumlah filter m , langkah konvolusi s , dan ukuran filter $k \times l$, di mana k menunjukkan jumlah variasi dari deret waktu di lapisan sebelumnya dan l sebagai panjang filter.

Pooling Layer: Peta fitur dibagi menjadi N segmen dengan panjang yang sama, kemudian setiap segmen direpresentasikan oleh nilai rata-rata atau maksimum. Kegunaan dari operasi pooling adalah menurunkan sampel output konvolusi, sehingga mengurangi variabilitas dalam aktivasi tersembunyi.

Feature Layer: Setelah beberapa operasi konvolusi dan pooling, seri waktu asli direpresentasikan oleh serangkaian peta fitur. Menghubungkan semua peta fitur untuk menghasilkan seri waktu panjang baru sebagai representasi akhir dari input asli dalam lapisan fitur.

Output layer: Lapisan output memiliki n neuron, sesuai dengan n kelas dari seri waktu. Lapisan ini terhubung sepenuhnya ke lapisan fitur (Zhao et al., 2017).

2.1.9. Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang menekankan keterbacaan kode. Bahasa ini mendukung berbagai paradigma pemrograman dan menawarkan banyak fitur, termasuk manajemen memori otomatis (Ihsan, 2021).

Python juga merupakan bahasa pemrograman berorientasi objek yang dinamis, cocok untuk berbagai jenis pengembangan perangkat lunak. Bahasa ini mendukung integrasi kuat dengan bahasa pemrograman lain dan alat bantu lainnya. Python dilengkapi dengan pustaka standar yang besar dan komprehensif, memungkinkan pengguna mempelajarinya dalam waktu singkat. Kombinasi kapabilitas dan sintaksis yang jelas menjadikan Python pilihan populer di kalangan pengembang (Nugroho et al., 2020).

2.1.10. Tensorflow

Tensorflow adalah framework machine learning open-source yang dikembangkan dan didukung oleh Google, bertujuan untuk mendukung penelitian dan pengembangan dalam bidang komputer cerdas. TensorFlow menyediakan antarmuka yang fleksibel,

memungkinkan pengguna untuk mengekspresikan algoritma machine learning dengan mudah dan menjalankannya di berbagai sistem operasi dan perangkat keras.

Salah satu keunggulan utama TensorFlow adalah kemampuannya untuk menangani berbagai macam tugas dalam data science, mulai dari pengenalan gambar dan suara hingga pemrosesan bahasa alami dan analisis prediktif. TensorFlow telah menjadi salah satu library paling populer dalam komunitas data science, didukung oleh sejumlah besar pengembang dan komunitas pengguna yang aktif.

Selain itu, TensorFlow terus diperbarui dengan fitur-fitur terbaru yang mendukung perkembangan teknologi machine learning, seperti kemampuan untuk melakukan training model pada skala besar dan integrasi dengan alat-alat lain dalam ekosistem data science. Keberadaan dokumentasi yang luas dan berbagai tutorial juga memudahkan para pengembang dan peneliti baru untuk memulai dan mengimplementasikan proyek-proyek menggunakan TensorFlow (Ihsan, 2021).

2.1.11. Matplotlib

Matplotlib adalah salah satu library visualisasi data yang paling populer digunakan dalam bahasa pemrograman Python. Library ini dibangun oleh John Hunter bersama dengan beberapa kontributor lainnya, yang telah menginvestasikan banyak waktu untuk mengembangkan perangkat lunak ini sehingga dapat digunakan oleh ilmuwan dan filosof di seluruh dunia. Matplotlib adalah library grafis yang integral dalam paket visualisasi data Python dan didukung dengan baik oleh NumPy, Pandas, dan library terkait lainnya (Sial et al., 2021).

Matplotlib adalah library visualisasi data yang terkenal dan banyak digunakan dalam Python. Library ini menyediakan berbagai macam plot 2D dan 3D yang dapat disesuaikan, termasuk scatter plot, line plot, histogram, dan banyak lagi. Matplotlib dibangun di atas array NumPy dan sangat kompatibel dengan library Python lainnya seperti Pandas, NumPy, dan scikit-learn. Selain itu, Matplotlib memiliki lingkungan interaktif

yang dapat digunakan di berbagai platform, yang memungkinkan peneliti untuk membuat visualisasi data yang kompleks dan berinteraksi dengan visualisasi tersebut secara real-time.

Matplotlib telah banyak digunakan dalam penelitian sebagai alat visualisasi data yang kuat, yang memungkinkan peneliti untuk mengkomunikasikan temuan mereka dengan efektif. Beberapa cara Matplotlib membantu dalam penelitian meliputi: memvisualisasikan data kompleks dengan berbagai opsi visualisasi seperti line plot, scatter plot, histogram, dan heatmap; menyediakan interaktivitas yang memungkinkan manipulasi visualisasi secara real-time; dan memastikan reproduksibilitas dengan visualisasi berkualitas publikasi yang dapat dengan mudah direproduksi dalam makalah penelitian, presentasi, dan materi lainnya. Integrasi yang mudah dengan alat dan paket Python lainnya, seperti NumPy, Pandas, dan SciPy, membuat Matplotlib menjadi alat yang sangat berguna bagi para peneliti untuk menganalisis dan mengkomunikasikan temuan mereka melalui visualisasi data yang efektif (Lavanya et al., 2023).

2.1.12. Google Colab

Google Colaboratory, atau yang lebih dikenal dengan Google Colab, adalah layanan open-source yang disediakan oleh Google untuk pengguna Gmail. Google Colab memberikan akses ke GPU untuk pengguna yang tidak memiliki sumber daya yang cukup. Layanan ini menyediakan 12.72 GB RAM dan 358.27 GB ruang hard disk dalam satu runtime yang berlangsung selama 12 jam sebelum direset. Pengguna harus memilih jenis runtime setiap kali membuka file Google Colab, dengan pilihan None (menggunakan CPU komputer pengguna), GPU, atau TPU (untuk pemrosesan tensor). Google Colab pada dasarnya adalah Jupyter Notebook online dengan semua fungsionalitas yang dimilikinya, termasuk kemampuan untuk mengunduh dataset langsung dari Google Drive atau Kaggle (Kanani & Padole, 2019).

Google Colaboratory, atau Colab, adalah proyek yang bertujuan untuk menyebarkan pendidikan dan penelitian machine learning. Notebook Colaboratory berbasis Jupyter dan bekerja seperti Google Docs, memungkinkan berbagi dan kolaborasi pada notebook yang sama. Google Colab menyediakan runtime Python 2 dan 3 yang telah dikonfigurasi dengan pustaka penting untuk machine learning dan kecerdasan buatan, seperti TensorFlow, Matplotlib, dan Keras. Mesin virtual (VM) di bawah runtime akan dinonaktifkan setelah beberapa waktu, menyebabkan data dan konfigurasi pengguna hilang, tetapi notebook tetap disimpan dan file dapat ditransfer dari hard disk VM ke akun Google Drive pengguna. Infrastruktur Google Colaboratory di hosting di platform Google Cloud (Carneiro et al., 2018).

2.1.13. Visual Studio Code

Visual Studio Code adalah editor kode sumber yang ringan namun sangat kuat, tersedia untuk berbagai sistem operasi termasuk Windows, macOS, dan Linux. Editor ini mendukung berbagai bahasa pemrograman secara bawaan, seperti JavaScript, TypeScript, dan Node.js, membuatnya ideal untuk pengembangan web dan aplikasi. Selain dukungan bawaan, Visual Studio Code memiliki ekosistem ekstensi yang sangat kaya. Pengguna dapat menambahkan ekstensi untuk berbagai bahasa pemrograman lainnya seperti C++, C#, Java, Python, PHP, dan Go, serta untuk berbagai runtime seperti .NET dan Unity.

Kemampuan untuk menyesuaikan editor dengan ekstensi membuat Visual Studio Code sangat fleksibel dan dapat disesuaikan dengan kebutuhan pengembang. Editor ini juga dilengkapi dengan fitur-fitur canggih seperti debugging, kontrol versi Git terintegrasi, dan berbagai alat pengembang lainnya yang mempermudah proses pengembangan. Antarmuka yang user-friendly dan performa yang cepat menjadikan Visual Studio Code pilihan populer di kalangan pengembang, baik pemula maupun profesional. Selain itu, komunitas pengguna dan pengembang yang aktif terus berkontribusi pada pengembangan

ekstensi dan peningkatan fitur, memastikan bahwa Visual Studio Code selalu up-to-date dengan teknologi terbaru (Romzi & Kurniawan, 2020).

2.1.14. Penelitian Terdahulu

Tabel 2. 1 Penelitian Terdahulu

No	Pengarang & Tahun	Judul Penelitian	Kesimpulan
1	(Widiputra et al., 2021)	Prediksi Indeks BEI dengan Ensemble Convolutional Neural Network dan Long Short-Term Memory.	Studi ini mengusulkan model CNN-LSTM untuk memprediksi nilai indeks saham esok hari dengan akurasi tertinggi dan RMSE terendah. Model ini menggabungkan CNN untuk ekstraksi fitur dan LSTM untuk prediksi, diuji dengan data Bursa Efek Indonesia, memberikan referensi bagi investor dalam pengambilan keputusan investasi dan penelitian keuangan.
2	(Alfandi & Sihite, 2022)	Penerapan Metode CNN-LSTM Dalam Memprediksi Hujan	Studi ini menyimpulkan bahwa metode CNN-LSTM efektif untuk

		<p>Pada Wilayah Kota Medan</p>	<p>memprediksi curah hujan di Medan. Algoritma CNN-LSTM menggunakan arsitektur dengan 2 lapisan tersembunyi dan dilatih dengan 400 epochs dan batch_size 6, menghasilkan prediksi yang semakin baik dari tahun ke tahun dengan nilai RMSE rendah sebesar 0.08, menunjukkan akurasi yang tinggi dalam memprediksi curah hujan.</p>
3	(M Devid Alam Carnegie & Chairani, 2023)	<p>Perbandingan Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU) Untuk Memprediksi Curah Hujan.</p>	<p>Analisis menggunakan LSTM dan GRU untuk prediksi curah hujan menunjukkan bahwa LSTM 1, dengan pembagian dataset 7:3, memiliki kinerja terbaik (RMSE 16.81, MSE 282.55, MAD 10.43). Model GRU 1 (7:3)</p>

			<p>terbaik untuk prediksi hujan/tidak hujan (akurasi 62%, presisi 58%, recall 66%, f1 score 62%).</p> <p>Disarankan untuk penelitian selanjutnya menambah jumlah data, mempertimbangkan fitur penguapan dan musim, serta menggunakan model kompleks seperti CNN atau ANN dengan epoch lebih banyak untuk prediksi yang lebih akurat dalam menghadapi kondisi cuaca yang kompleks.</p>
--	--	--	---

BAB III

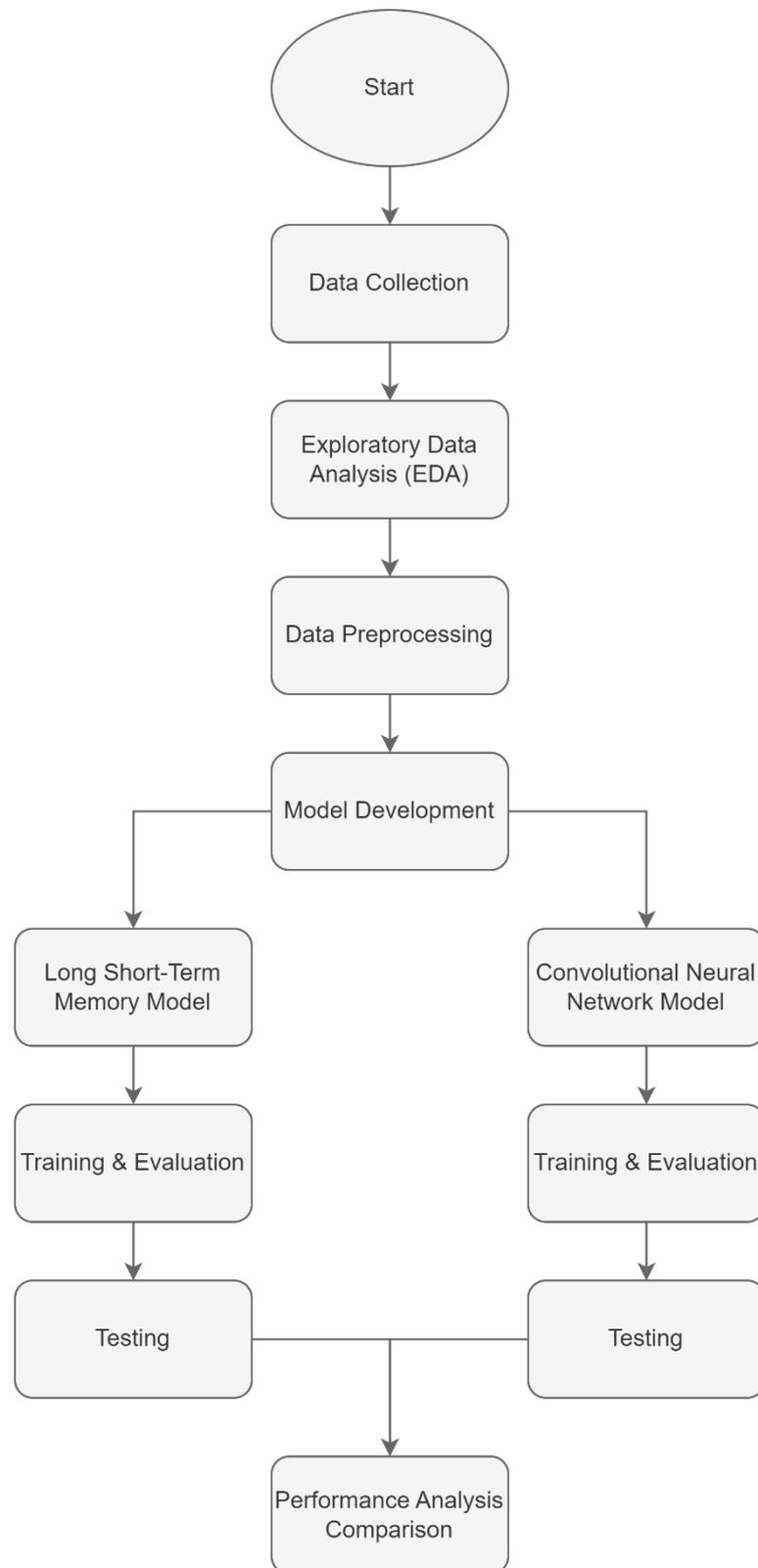
METODOLOGI PENELITIAN

3.1. Gambaran Umum Objek penelitian

Penelitian ini adalah sebuah studi eksperimental yang menggunakan dataset online yang dikumpulkan secara mandiri melalui situs web Badan Meteorologi Klimatologi dan Geofisika (BMKG). Karya ilmiah ini bertujuan untuk menganalisis kinerja algoritma deep learning dalam melakukan forecasting dengan tujuan menemukan algoritma terbaik dalam melakukan forecasting kelembaban udara, dengan harapan dapat menemukan algoritma terbaik untuk tujuan tersebut.

Langkah pertama dalam penelitian ini adalah pengumpulan data dari database online yang disediakan oleh Badan Meteorologi Klimatologi dan Geofisika (BMKG), UPT Balai Besar Meteorologi Klimatologi dan Geofisika Wilayah I. Untuk proses pembersihan data dan pembangunan model, penelitian ini menggunakan metode utama Tensorflow dan bahasa pemrograman Python. Selain itu, Google Colab dan Visual Studio Code digunakan sebagai lingkungan pengembangan (IDE) untuk analisis data.

Dengan demikian, penelitian ini tidak hanya fokus pada pengumpulan dan pembersihan data, tetapi juga pada pengembangan model deep learning yang efektif untuk peramalan kelembaban udara, memanfaatkan alat dan teknologi terbaru dalam bidang data science dan machine learning. Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pemahaman serta penerapan algoritma deep learning pada bidang meteorologi



Gambar 3. 1 Alur Penelitian

3.2. Data collection

Data kelembaban udara yang digunakan dalam penelitian ini diambil dari dataset online yang disediakan oleh Badan Meteorologi Klimatologi dan Geofisika (BMKG), UPT Balai Besar Meteorologi Klimatologi dan Geofisika Wilayah I. Dataset yang diperoleh merupakan data time series yang mencakup periode dari Januari 2010 hingga Februari 2024. Data ini sangat penting dalam analisis karena memberikan informasi kelembaban udara dalam jangka waktu yang panjang, memungkinkan untuk identifikasi tren jangka panjang dan pola musiman. Proses pengumpulan data ini melibatkan unduhan dari sumber resmi yang terpercaya untuk memastikan bahwa data yang digunakan adalah akurat dan dapat diandalkan. Pada tahapan selanjutnya data ini akan diproses dan dibersihkan untuk menghilangkan anomali atau nilai yang hilang sebelum digunakan dalam model. penggunaan data time series ini memungkinkan model untuk memprediksi kelembaban udara di masa depan berdasarkan pola yang telah teridentifikasi dari data historis.

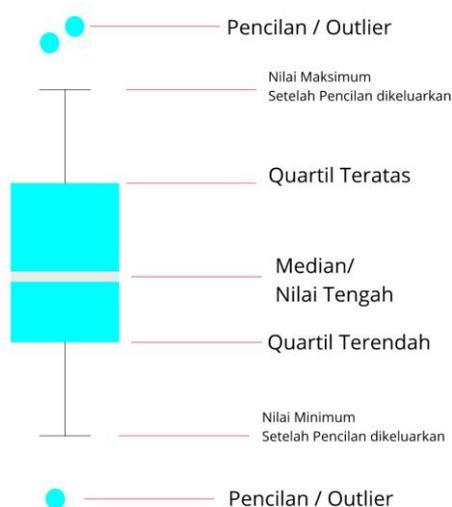
Tanggal	Tn	Tx	Tavg	RH_avg	RR	ss	ff_x	ddd_x	ff_avg	ddd_carr
01-01-2010	24	32,4	25,9	85	0	1,5				
02-01-2010	24	31,6	27,6	83	5	0,5				
03-01-2010	24	29,6	27	86	24	3,2				
04-01-2010	24	32	27,4	84	0	4,1				
05-01-2010	24	34,4	27,8	80	8	4,2				
....
25-02-2022	24,6	33,4	30,3	72	0	7,8	5	60	1	C
26-02-2022	24	32,2	29,8	73		2,6	6	100	2	C
27-02-2022	23,8	35	31,3	63		3,6	5	30	2	NE
28-02-2022	24,6	35,3	31,4	63		5,4	7	70	2	E
29-02-2024	24,6	32,2	29,3	79		6,3	5	30	1	C

Gambar 3. 2 Dataset yang telah diunduh dari situs web BMKG

3.3. Exploratory Data Analysis (EDA)

Exploratory data analysis atau EDA merupakan langkah awal yang penting dalam penelitian ini untuk menganalisis dan mengevaluasi data. Pada penelitian ini, EDA dilakukan pada dataset kelembaban udara yang diperoleh dari Badan Meteorologi Klimatologi dan Geofisika (BMKG), UPT Balai Besar Meteorologi Klimatologi dan Geofisika Wilaya I. Tujuan utama dari EDA adalah untuk memahami kualitas data yang akan digunakan, sehingga potensi kesalahan di kemudian hari dapat diminimalkan. Melalui EDA, peneliti dapat mengeksplorasi isi dan komponen penyusun data secara mendalam.

Proses ini melibatkan investigasi awal terhadap dataset kelembaban udara dimana penulis berusaha untuk menemukan pola, anomali, dan frekuensi yang ada. Teknik-teknik statistik serta representasi grafis digunakan secara ekstensif dalam EDA untuk memvisualisasikan data dengan lebih jelas. Pada tahapan EDA ini, penulis menggunakan Box Plot untuk mendeteksi outlier dan memahami sebaran data. Dengan demikian, EDA tidak hanya membantu penulis dalam memahami struktur data, tetapi juga memungkinkan identifikasi awal masalah yang mungkin timbul, seperti data yang hilang atau outlier yang signifikan. Melalui pendekatan ini, penulis dapat meningkatkan kualitas dan akurasi data secara signifikan.



Gambar 3. 3 Visualisasi Box Plot Untuk Mendeteksi Outlier Pada Dataset

3.4. Data Preprocessing

Data preprocessing adalah tahap di mana penulis mengolah data lebih lanjut sehingga menjadi siap pakai dalam pengembangan model. Pada proses ini penulis mengubah dan mentransformasi fitur-fitur data ke dalam bentuk yang mudah interpretasikan dan diproses oleh model. Proses data cleaning dan data transformation termasuk dalam tahapan data preprocessing yang dilakukan oleh penulis.

Tahapan yang dilakukan dalam proses data cleaning adalah penanganan missing value, data yang tidak konsisten, duplikasi data, dan ketidak seimbangan data. Sedangkan dalam proses transformasi data, penulis akan melakukan scaling atau merubah skala data agar sesuai dengan skala tertentu dengan standarisasi. Standarisasi adalah langkah dasar dalam machine learning sebelum pelatihan untuk menghapus rata-rata dari data. Metode standarisasi didasarkan pada rata-rata (mean) dan standar deviasi. Standarisasi suatu dataset melibatkan pengubahan skala distribusi nilai, sehingga nilai rata-rata (mean) menjadi 0 dan standar deviasi adalah 1.

$$x' = \frac{x - \mu}{\sigma} \quad (8)$$

Pada tahapan ini juga penulis akan melakukan proses train-test split untuk membagi data menjadi 70% data latih, 15% data validasi dan 15% data uji. Namun pembagian ini akan disesuaikan tergantung pada kebutuhan dan sifat data yang dimiliki pada proses pengembangan model.

$$\text{Data Latih} = \text{Jumlah Data} \times \frac{70}{100} \quad (9)$$

$$\text{Data Validasi} = \text{Jumlah Data} \times \frac{15}{100} \quad (10)$$

$$\text{Data Uji} = \text{Jumlah Data} \times \frac{15}{100} \quad (11)$$

3.5. Model Architecture

Pada tahapan ini, penulis akan melakukan pemilihan model yang akan digunakan dalam membangun model yang dapat melakukan forecasting kelembaban udara. Tujuan

utama dari tahapan ini adalah untuk menentukan model yang memiliki akurasi dan efektivitas terbaik dalam melakukan forecasting. Pada penelitian ini, penulis memilih dua jenis model untuk dikembangkan, yaitu Long Short-Term Memory (LSTM) dan Convolutional Neural Network (CNN).

LSTM dipilih karena kemampuannya yang unggul dalam menangani data deret waktu dengan pola-pola jangka panjang yang kompleks, berkat mekanisme memorinya yang memungkinkan informasi penting untuk dipertahankan dan informasi yang tidak relevan untuk dihapus. Sementara itu, CNN dipilih karena kehandalan model ini dalam mengenali pola-pola yang kompleks dan kemampuan dalam mengekstrak fitur-fitur penting dari data, yang dapat diaplikasikan secara efektif pada data deret waktu untuk mendeteksi tren dan variasi yang signifikan. Kedua model ini diharapkan dapat memberikan hasil yang optimal dalam forecasting kelembaban udara.

3.5.1. Long Short-Term Memory Architecture

Arsitektur model Long Short-Term Memory (LSTM) yang dirancang oleh penulis dalam penelitian ini bertujuan untuk membangun model peramalan kelembaban udara. LSTM adalah jenis jaringan saraf tiruan yang cocok untuk memproses data sekuensial dan menangkap hubungan temporal dalam data. Model ini dirancang dengan memperhatikan berbagai parameter penting, seperti jumlah layer LSTM, jumlah neuron dalam setiap layer, serta fungsi aktivasi yang digunakan, untuk menghasilkan prediksi kelembaban udara yang akurat. Berikut adalah komponen utama dari arsitektur yang digunakan:

- a. **Input Layer:** Lapisan input menerima data historis kelembaban udara yang telah di preproses.
- b. **LSTM Layers:** Menggunakan beberapa lapisan Longs Short-Term Memory untuk menangkap hubungan temporal dalam data historis kelembaban udara.
- c. **Dense Layers:** Merupakan lapisan fully connected untuk menggabungkan fitur dan membuat prediksi akhir.

- d. Output Layer: Lapisan output yang menghasilkan prediksi kelembaban udara untuk periode waktu tertentu di masa depan.



Gambar 3. 4 Layer Long Short-Term Memory

3.5.2. Convolutional Neural Network Architecture

Pada tahap selanjutnya, penulis melakukan desain arsitektur kedua menggunakan Convolutional Neural Network (CNN) yang akan digunakan untuk melakukan forecasting kelembaban udara. CNN adalah jenis jaringan saraf tiruan yang cocok untuk memproses data spasial dan juga dapat untuk data sequential dengan penyesuaian tertentu. Model ini dirancang dengan memperhatikan berbagai parameter penting seperti jumlah layer Convolutional, jumlah neuron dalam setiap layer, serta fungsi aktivasi yang digunakan, untuk menghasilkan prediksi kelembaban udara yang akurat. Berikut adalah komponen utama dari arsitektur yang digunakan:

- a. Input Layer: Lapisan input menerima data historis kelembaban udara yang telah di preproses.
- b. Convolutional Layers: Menggunakan beberapa lapisan konvolusi untuk mengekstraksi fitur dari data sekuensial kelembaban udara.
- c. Flatten Layer: lapisan yang meratakan output dari lapisan konvolusi sebelum memasuki lapisan fully connected.
- d. Dense Layer: Lapisan fully connected untuk menggabungkan fitur dan membuat prediksi akhir.
- e. Output Layer: Lapisan output yang menghasilkan prediksi kelembaban udara untuk periode waktu tertentu di masa depan.



Gambar 3. 5 Layer Convolutional Neural Network

3.6. Model Development

3.6.1. Model Building

Tahap pertama dalam pengembangan model adalah membangun arsitektur model menggunakan Tensorflow. Pada tahap ini, dua jenis model akan dikembangkan, yaitu Long Short-Term Memory (LSTM) dan Convolutional Neural Network (CNN). Arsitektur model yang digunakan mengacu pada bagian 3.4, Model Architecture, yang telah dijelaskan sebelumnya oleh penulis. Model ini akan dibangun menggunakan API Keras yang telah disediakan oleh Tensorflow, yang memungkinkan pembuatan model LSTM dan CNN secara intuitif dan efisien.

Proses pembangunan model dimulai dengan menggunakan menentukan struktur dasar dari masing-masing jenis model. Untuk model LSTM, langkah awal melibatkan penentuan jumlah lapisan LSTM, jumlah unit di setiap lapisan, dan parameter lainnya seperti dropout dan aktivasi. Sementara itu untuk model CNN, penulis akan menentukan jumlah lapisan konvolusi, fungsi aktivasi, serta jumlah dan ukuran filter.

3.6.2. Model Compilation

Setelah arsitektur model selesai dibangun, langkah selanjutnya adalah mengkompilasi model. Proses ini melibatkan pemilihan optimizer yang sesuai, seperti stochastic gradient descent (SGD), yang berfungsi untuk mengatur kecepatan pembelajaran dan menyesuaikan bobot model selama pelatihan. Selain itu, penentuan fungsi kerugian (loss function) yang tepat sangat penting, karena akan bergantung pada jenis masalah yang diselesaikan. Pada penelitian ini, penulis akan menggunakan dua jenis metrics evaluasi, yaitu root mean squared error (RMSE) dan mean squared error (MSE).

- a. Optimizer: Optimizer adalah algoritma yang digunakan untuk memperbarui bobot model selama proses training. Optimizer yang digunakan oleh penulis pada penelitian ini adalah *Stochastic Gradient Descent* (SGD). SGD adalah algoritma optimasi yang digunakan pada *machine learning*, khususnya pada

artificial neural network yang diterapkan pada *deep learning*. Algoritma ini bersifat iteratif, yang berarti membutuhkan proses pencarian untuk menentukan langkah diskrit (Hastomo et al., 2021).

- b. **Loss Function:** Loss function digunakan untuk mengukur seberapa baik model memprediksi output yang diinginkan. MAE dapat digunakan jika outlier merupakan bagian dari data yang rusak. MAE tidak terlalu memberikan penalti outlier pada data pelatihan (norma L_1 secara tidak langsung mengurangi semua kesalahan dari outlier), sehingga memberikan ukuran kinerja yang umum dan terbatas untuk model (Chicco et al., 2021).

$$MAE = \frac{1}{m} \sum_{i=1}^m |X_i - Y_i| \quad (12)$$

- c. **Metrics:** Metrics digunakan untuk mengevaluasi kinerja model selama training dan testing. Metrics yang digunakan oleh penulis pada penelitian ini adalah Root Mean Squared Error (RMSE) dan Mean Squared Error (MSE).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (13)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_1 - \hat{y}_1)^2 \quad (14)$$

3.6.3. Model Training dan Evaluation

Setelah model berhasil dikompilasi, penulis akan melanjutkan langkah selanjutnya yaitu melatih model menggunakan data latih yang telah dipersiapkan. Proses pelatihan melibatkan penentuan jumlah epoch dan batch size yang optimal untuk mengoptimalkan performa model.

- a. **Initialization:** Inialisasi bobot model secara acak atau menggunakan metode tertentu yang memastikan bobot awal memiliki nilai yang tepat.
- b. **Forward Pass:** Data input melewati seluruh lapisan model hingga menghasilkan output prediksi. Pada tahap ini, setiap lapisan dalam model melakukan perhitungan berdasarkan bobot dan input yang diberikan.

- c. **Loss Calculation:** Menghitung error atau loss antara output prediksi dengan nilai aktual menggunakan loss function yang telah ditentukan. Ini memberikan ukuran seberapa baik model memprediksi nilai yang diinginkan.
- d. **Backward Pass:** Menghitung gradien dari loss function terhadap setiap bobot dalam model dengan menggunakan algoritma backpropagation. Gradien ini menunjukkan arah dan seberapa besar perubahan yang harus dilakukan pada bobot untuk mengurangi error.
- e. **Weight Update:** Memperbarui bobot model berdasarkan gradien yang telah dihitung dan optimizer yang digunakan.
- f. **Epochs:** Mengulangi proses *forward pass* dan *backward pass* untuk sejumlah epoch yang telah ditentukan hingga model mencapai performa yang diinginkan. Setiap epoch terdiri dari satu kali iterasi melalui seluruh data latih.

Proses training ini akan diulang hingga model mencapai konvergensi atau hingga jumlah epoch yang telah ditentukan tercapai. Selama proses pelatihan, penulis akan secara berkala mengevaluasi model menggunakan data validasi untuk memantau kemungkinan overfitting dan melakukan penyesuaian parameter jika diperlukan. Hal ini penting untuk memastikan bahwa model tidak hanya dapat mengingat data latih dengan baik, tetapi juga dapat secara umum menggeneralisasi dan memberikan prediksi yang akurat terhadap data baru yang belum pernah dilihat sebelumnya.

3.7. Model Testing

Setelah model dilatih dan dievaluasi, langkah terakhir adalah menguji performa model menggunakan data uji yang tidak pernah dilihat sebelumnya oleh model. Data uji ini penting untuk mengukur seberapa baik model dapat melakukan prediksi terhadap data yang baru dan tidak terlibat dalam proses pelatihan atau validasi sebelumnya. Evaluasi dengan menggunakan data uji membantu memastikan bahwa model dapat

menggeneralisasi dengan baik pada data yang baru dan belum dikenal sebelumnya, serta menilai secara objektif kinerja keseluruhan dari model yang telah dikembangkan.

- a. **Prediction:** Memasukkan data uji ke dalam model yang telah dilatih untuk melakukan prediksi kelembaban udara.
- b. **Evaluation:** Mengukur kinerja model menggunakan metrics RMSE dan MSE untuk perbandingan antara nilai prediksi dengan nilai aktual dari data uji.
- c. **Analysis:** Menganalisis hasil evaluasi untuk memahami seberapa baik model mampu melakukan forecasting kelembaban udara. Evaluasi ini membantu dalam mengevaluasi keandalan dan akurasi model dalam menghadapi data baru.
- d. **Comparison Model:** Membandingkan kinerja model LSTM dengan CNN untuk menilai keunggulan dan kekurangan setiap model dalam tugas forecasting kelembaban udara di kota Medan.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Pengolahan Data

Pengolahan data dimulai dengan penggabungan dataset dari 1 Januari 2010 hingga 29 Februari 2024. Proses penggabungan ini bertujuan untuk membentuk satu dataset komprehensif yang mencakup seluruh periode yang akan dianalisis. Setelah penggabungan, data yang diperoleh kemudian melalui tahap pembersihan dan normalisasi. Pembersihan data meliputi identifikasi dan penanganan data yang hilang, kesalahan, serta anomali yang ada dalam dataset. Dalam menangani outlier, penulis menggunakan metode *Inter Quartile Range* (IQR) untuk mendeteksi dan menangani outlier dalam data. IQR menghitung rentang di mana sebagian besar nilai dalam dataset terletak. IQR adalah selisih antara kuartil ketiga (Q3) dan kuartil pertama (Q1) dalam sebuah data.

Rumus IQR: $IQR = Q3 - Q1$

Di mana:

Q1 (Kuartil Pertama): Nilai yang memisahkan 25% data terendah dari 75% data lainnya.

Q3 (Kuartil Ketiga): Nilai yang memisahkan 75% data terendah dari 25% data tertinggi.

Langkah Perhitungan:

Dataset kelembaban udara adalah [85.0, 83.0, 86.0, ..., 79.0]

- a. Langkah 1: Urutkan dataset
- b. Langkah 2: Tentukan nilai Q1 dan Q3

Di mana:

$$Q1 = 75.0$$

$$Q3 = 83.0$$

- c. Langkah 3: Hitung IQR

Di mana:

$$IQR = Q3 - Q1 = 83.0 - 75.0 = 8.0$$

- d. Langkah 4: Menghitung batas outlier

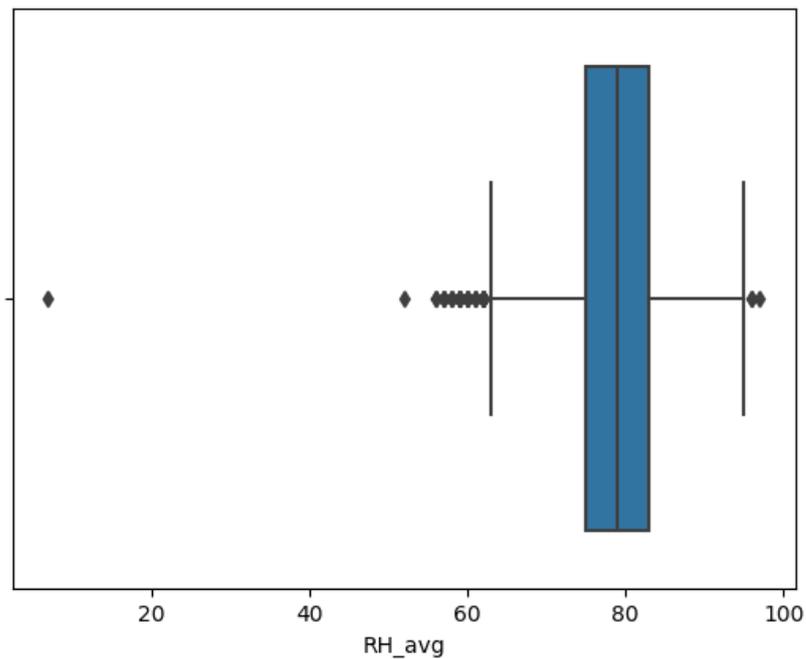
Di mana:

$$\text{Batas bawah } Q1 - 1.5 \times IQR = 75.0 - 1.5 \times 8.0 = -63$$

$$\text{Batas atas } Q3 + 1.5 \times IQR = 83.0 + 1.5 \times 8.0 = 95$$

- e. Langkah 5: Identifikasi Outlier

Nilai yang berada di bawah -63 atau di atas 95 dianggap sebagai outlier. Dalam dataset ini, terdapat beberapa nilai yang melampaui batas-batas tersebut, seperti yang ditunjukkan pada Gambar 4.1.



Gambar 4. 1 Plot Outlier pada Dataset Menggunakan Seaborn

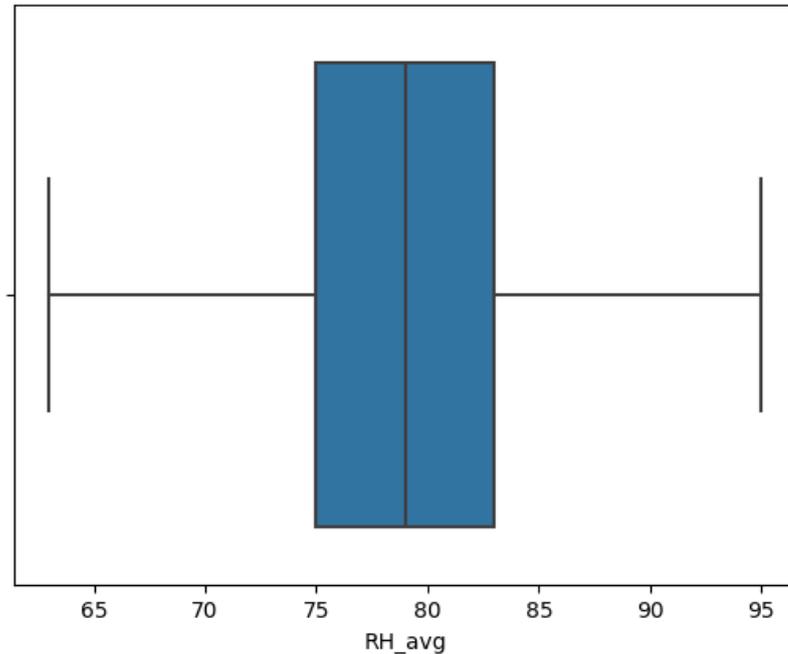
- f. Langkah 6: Penghapusan Outlier.

Proses penghapusan outlier pada dataset dilakukan menggunakan kode yang ditampilkan pada Gambar 4.2. Hasilnya, dataset yang telah dibersihkan dari outlier disimpan seperti yang ditunjukkan pada Gambar 4.3.

```
df = df[~((df['RH_avg'] < (Q1 - 1.5 * IQR)) | (df['RH_avg'] > (Q3 + 1.5 * IQR)))]
sns.boxplot(x=df["RH_avg"])
```

Gambar 4. 2 Kode untuk Menyimpan dan Memplot Dataset yang Bebas dari

Outlier



Gambar 4. 3 Plot Dataset Tanpa Outlier

Standarisasi dilakukan untuk mengubah skala fitur-fitur data agar berada dalam rentang yang seragam, sehingga dapat meningkatkan efisiensi dan kestabilan selama pelatihan model. Standarisasi dilakukan dengan cara mengurangi rata-rata (mean) pada fitur dan kemudian membagi dengan standar deviasi dari fitur tersebut. Rumus standarisasi untuk suatu fitur x adalah $z = \frac{x - \mu}{\sigma}$

Di mana:

z adalah nilai standar dari data.

x adalah nilai asli data.

μ adalah rata-rata dari semua nilai dalam fitur.

σ adalah standar deviasi dari semua nilai dalam fitur.

Proses standarisasi dilakukan melalui tahapan yang dijelaskan pada kode di Gambar 4.4.

```

import pandas as pd
from sklearn.preprocessing import StandardScaler

df = pd.read_csv('dataset.csv')
dataset = df["RH_avg"].values

scaler = StandardScaler()
dataset = scaler.fit_transform(dataset.reshape(-1, 1))

```

Gambar 4. 4 Kode untuk tahapan standarisasi data

Selanjutnya, dataset yang telah dibersihkan dan dinormalisasi dibagi menjadi tiga set yang berbeda untuk tujuan pelatihan, validasi, dan pengujian model. Pembagian ini dilakukan dengan proporsi 70% untuk set pelatihan, 15% untuk set validasi, dan 15% untuk set pengujian. Set pelatihan digunakan untuk melatih model, sedangkan set validasi digunakan untuk memantau dan menyesuaikan hyperparameter selama pelatihan untuk mencegah overfitting. Set pengujian, yang terpisah dari proses pelatihan dan validasi, digunakan untuk mengevaluasi kinerja akhir model.

$$Data\ latihan = 5110 \times \frac{70}{100}$$

$$Data\ Validasi = 5110 \times \frac{15}{100}$$

$$Data\ Uji = 5110 \times \frac{15}{100}$$

```

import pandas as pd

data = pd.read_csv('dataset.csv')
data = data["RH_avg"]

total_size = len(data)

train_size = int(total_size * 0.7)
validation_size = int(total_size * 0.15)
test_size = total_size - train_size - validation_size

train_data = data[:train_size]
validation_data = data[train_size:train_size+validation_size]
test_data = data[train_size+validation_size:]

```

Gambar 4. 5 Kode Pembagian Dataset

Dalam tahap persiapan model, data time series diolah menggunakan teknik windowing dengan ukuran jendela 90 timestep. Teknik windowing ini memecah data menjadi potongan-potongan waktu yang berukuran 90 hari, memungkinkan model untuk mempelajari pola dan tren yang terjadi dalam periode tersebut. Dengan demikian, model dapat memahami konteks temporal dari 90 hari sebelumnya untuk memprediksi nilai pada hari berikutnya. Proses ini sangat penting dalam forecasting time series, karena memungkinkan model untuk menangkap dinamika dan perubahan yang mungkin terjadi dalam jangka waktu tertentu.

```

window_size = 90
batch_size = 32
shuffle_buffer_size = 1000

def windowed_dataset(series, window_size, batch_size, shuffle_buffer):
    series = tf.expand_dims(series, axis=-1)
    ds = tf.data.Dataset.from_tensor_slices(series)
    ds = ds.window(window_size + 1, shift=1, drop_remainder=True)
    ds = ds.flat_map(lambda w: w.batch(window_size + 1))
    ds = ds.shuffle(shuffle_buffer)
    ds = ds.map(lambda w: (w[:-1], w[-1:]))
    return ds.batch(batch_size).prefetch(1)

```

Gambar 4. 6 Kode Teknik Windowing

4.2. Proses Perhitungan Model LSTM

LSTM bekerja dengan memproses data time series yang telah yang telah diproses seperti yang dijelaskan pada sub bab 4.1. pengolahan data. LSTM memproses dataset dengan window size 90 hari untuk memprediksi nilai kelembaban pada hari ke-90.

Dataset:

- a. Total data: 5049
- b. Window size: 90

Langkah-langkah perhitungan dengan LSTM:

- a. Inisialisasi Bobot dan Bias:
 1. Forgate Gate Weights (W_f): 0.5
 2. Input Gate Weights (W_i): 0.6

3. Output Gate Weights (W_o): 0.7
 4. Cell State Weights (W_c): 0.5
 5. Bias untuk Forget Gate (b_f): 0.1
 6. Bias untuk Input Gate (b_i): 0.1
 7. Bias untuk Output Gate (b_o): 0.1
 8. Bias untuk Cell State (b_c): 0.1
- b. Data untuk jendela pertama (t_1 hingga t_{90}):
1. $Data = [85.0, 83.0, 86.0, \dots, 84.0]$
- c. Inisialisasi Hidden State dan Cell State pada Timestep Pertama:
1. $h_{prev} = 0$
 2. $c_{prev} = 0$
- d. Perhitungan pada Timestep pertama:

1. Forget gate:

$$f_t = \sigma(W_f \cdot (h_{prev} + x_t) + b_f)$$

Di mana:

$$W_f = 0.5$$

$$x_t = [85.0, 83.0, 86.0, \dots, 84.0]$$

$$f_t = \sigma(0.5 \cdot (0 + avg(x_t)) + 0.1)$$

$$f_t = \sigma(0.5 \cdot (0 + 81.7) + 0.1) = \sigma(40.95) = 1.0$$

2. Input Gate:

$$i_t = \sigma(W_i \cdot (h_{prev} + x_t) + b_i)$$

Di mana:

$$W_i = 0.6$$

$$i_t = \sigma(0.6 \cdot (0 + 81.7) + 0.1) = \sigma(4412.8) = 1.0$$

3. Cell State Update:

$$C_t = f_t \cdot C_{prev} + i_t \cdot \tanh(W_c \cdot (h_{prev} + x_t) + b_c)$$

Di mana:

$$W_c = 0.5$$

$$\tanh(W_c \cdot (0 + 81.7) + 0.1) = \tanh(41.45) = 1.0$$

$$C_t = 1.0 \cdot 0 + 1.0 \cdot 1.0 = 1.0$$

4. Output Gate:

$$o_t = \sigma(W_o \cdot (h_{prev} + x_t) + b_o)$$

Di mana:

$$W_o = 0.7$$

$$o_t = \sigma(0.7 \cdot (0 + 81.7) + 0.1) = \sigma(57.29) = 1.0$$

5. Hidden State:

$$h_t = o_t \tanh(C_t)$$

Di mana:

$$h_t = 1.0 \cdot \tanh(1.0) = 1.0 \cdot 0.76 = 0.76$$

Hasil:

- a. Hidden State (h_t) pada timestep pertama: 0.76
- b. Cell State (c_t) pada timestep pertama: 1.0

Penggunaan pada Timestep Berikutnya:

- a. h_{prev} dan c_{prev} untuk timestep berikutnya adalah nilai h_t dan c_t dari timestep saat ini:

1. $h_{prev} = 0.76$

2. $C_{prev} = 1.0$

Pada timestep berikutnya, model akan menggunakan nilai ini sebagai bagian dari input untuk menghitung nilai baru dari hidden state dan cell state berdasarkan data baru yang diinputkan.

4.3. Proses Perhitungan Model CNN

Convolutional Neural Network (CNN) bekerja dengan menggeser filter ke kanan sebanyak satu langkah pada *window* data. Setiap langkah, CNN melakukan operasi perkalian antara nilai data pada *window* tersebut dengan bobot filter. Hasil perkalian ini kemudian dijumlahkan bersama nilai bias. Proses ini berlanjut secara berulang sampai seluruh data dalam *window* pertama telah diproses sepenuhnya.

Filter dan Window Input:

- a. Input Data Window Pertama (t_1 hingga t_{90}):
 1. $Data = [85.0, 83.0, 86.0, \dots, 84.0]$
- b. Filter: $[0.2, 0.5, 0.3]$
- c. Bias: 0
- d. Stride: 1

Proses Perhitungan Konvolusi:

- a. $Data = [85.0, 83.0, 86.0]$
- b. Perhitungan:

$$Output_1 = (85.0 \times 0.2) + (83.0 \times 0.5) + (86.0 \times 0.3) + 0$$

$$Output_1 = 17 + 41.5 + 25.8 = 84.3$$

Proses konvolusi terus berlanjut hingga filter telah bergerak melintasi seluruh *window* data, menghasilkan total 88 nilai output.

$$Output_{88} = (S_{88} \times 0.2) + (S_{89} \times 0.5) + (S_{90} \times 0.3) + 0$$

Flatten Layer: Setelah lapisan konvolusi selesai memproses data, output akan dikonversi menjadi vector 1D oleh Flatten Layer. Hasil output dari *window* pertama menghasilkan vector 1D dengan ukuran 88.

$$Flatten = [84.3, O_2, O_3, \dots, O_{88}]$$

Dense Layer: Vektor 1D dari Flatten Layer kemudian diteruskan ke Dense Layer untuk memprediksi suhu pada hari ke-90. Dense Layer memiliki satu neuron untuk prediksi, dengan bobot neuron $[W_1, W_2, W_3, \dots, W_{88}]$ dan bias b .

- a. Perhitungan dense layer:

$$\text{Output Neuron} = (W_1 \times X_1) + (W_2 \times X_2) + \dots + (W_{88} \times X_{88})$$

Di mana:

$$\text{Output Neuron} = (0.1 \times 84.3) + (W_2 \times X_2) + \dots + (W_{88} \times X_{88})$$

$$\text{Output Neuron} = 8.43 + (W_2 \times X_2) + \dots + (W_{88} \times X_{88})$$

$$= \text{Prediksi Kelembaban Udara Hari Ke } - 90$$

4.4. Implementasi Model LSTM dan CNN dengan Tensorflow

4.4.1. Model LSTM

Model Long Short-Term Memory (LSTM) dirancang untuk menangkap dependensi temporal jangka panjang yang relevan dalam data time series. Arsitektur yang diterapkan adalah sebagai berikut:

Table 4. 1 Ringkasan Model LSTM

Layer Type	Output Shape	Param
LSTM	(None, None, 64)	16,896
LSTM	(None, None, 64)	33,024
LSTM	(None, None, 64)	33,024
Dense	(None, 128)	8,320
Dense	(None, 64)	8,256
Dense	(None, 1)	65
Total		99,585

Penggunaan optimizer SGD dengan learning rate 0.001 dan fungsi loss Mean Absolute Error (MAE) memastikan konvergensi yang optimal selama proses pelatihan.

Model dilatih untuk mengoptimalkan prediksi berdasarkan data pelatihan dan divalidasi menggunakan data validasi untuk menghindari overfitting.

```
lstm_model = tf.keras.models.Sequential([
    tf.keras.layers.Bidirectional(LSTM(64, return_sequences=True)),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Bidirectional(LSTM(128, return_sequences=True)),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Bidirectional(LSTM(256)),
    Dense(32, activation='relu'),
    Dense(1)
])

optimizer = tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.9)
lstm_model.compile(loss=tf.keras.losses.MeanAbsoluteError(),
                  optimizer=optimizer,
                  metrics=[MeanSquaredError(), RootMeanSquaredError()])
```

Gambar 4. 7 Kode Untuk Membangun Arsitektur Model LSTM

4.4.2. Model CNN

Model Convolutional Neural Network (CNN) dirancang untuk memanfaatkan fitur spasial dari data time series melalui lapisan konvolusi. Arsitektur yang diterapkan adalah sebagai berikut:

Table 4. 2 Ringkasan Model CNN

Layer Type	Output Shape	Param
Conv1D	(None, 89, 64)	192
MaxPooling1D	(None, 44, 64)	0
Conv1D	(None, 43, 64)	8,256
MaxPooling1D	(None, 21, 64)	0
Flatten	(None, 1344)	0
Dense	(None, 128)	172,160
Dense	(None, 64)	8,256
Dense	(None, 1)	65
Total		188,929

Arsitektur CNN dioptimalkan dengan menggunakan optimizer SGD dengan learning rate 0.001 dan fungsi loss MSE. Model ini dilatih dengan pendekatan yang mirip dengan LSTM namun mengandalkan kemampuan konvolusi untuk mengidentifikasi pola spasial dalam data.

```
cnn_model = tf.keras.models.Sequential([
    Conv1D(64, kernel_size=2, activation="relu", input_shape=[window_size, 1]),
    MaxPooling1D(pool_size=2),
    Conv1D(64, kernel_size=2, activation="relu"),
    MaxPooling1D(pool_size=2),
    Flatten(),
    Dense(128, activation="relu"),
    Dense(64, activation="relu"),
    Dense(1),
])
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001, momentum=0.9)
lstm_model.compile(loss=tf.keras.losses.MeanAbsoluteError(),
                   optimizer=optimizer,
                   metrics=[MeanSquaredError(), RootMeanSquaredError()])
```

Gambar 4. 8 Kode Untuk Membangun Arsitektur Model CNN

4.5. Proses Pelatihan Modal

Kedua model, LSTM dan CNN menggunakan set pelatihan yang telah dibagi sebelumnya pada proses pengolahan data. Data ini dibagi menjadi 32 batch per epoch selama pelatihan model. Pada proses pelatihan ini, juga dilakukan tahapan evaluasi model menggunakan set evaluasi untuk memantau kinerja dari model. Jumlah epoch yang digunakan pada pada proses pelatihan model adalah 50 epoch, dengan penerapan Callbacks untuk mencegah overfitting.

```
history_lstm = lstm_model.fit(train_set,
                              validation_data=val_set,
                              callbacks=[callbacks]
                              epochs=50)
```

Gambar 4. 9 Kode Pelatihan Model LSTM

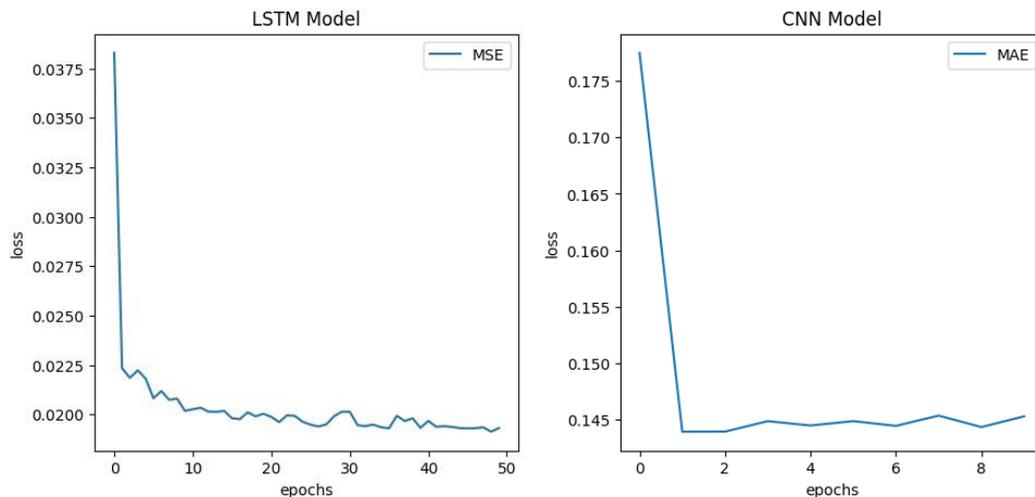
```

history_cnn = cnn_model.fit(train_Set,
                             validation_data=val_set,
                             callbacks=[callbacks],
                             epochs=50)

```

Gambar 4. 10 Kode Pelatihan Model CNN

Grafik konvergensi loss selama proses pelatihan ditampilkan pada Gambar 4.8, di mana kurva tersebut memperlihatkan tren penurunan nilai loss yang konsisten dan stabil, baik pada set pelatihan maupun validasi. Hal ini mengindikasikan bahwa model mampu belajar dengan baik dari data tanpa menunjukkan tanda-tanda overfitting, yang tercermin dari penurunan loss yang terus menerus seiring bertambahnya epoch.



Gambar 4. 11 Loss Pada Model LSTM & CNN

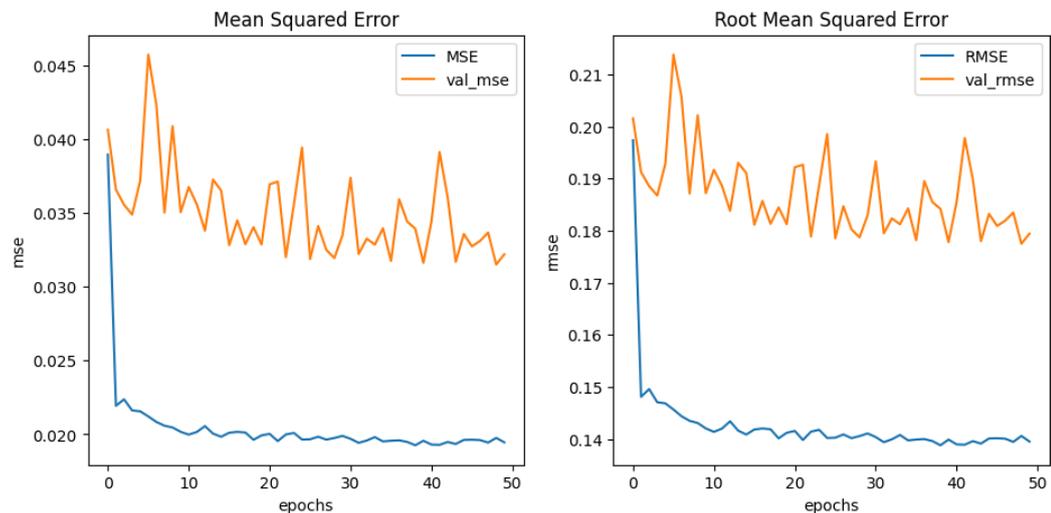
4.6. Analisis Performa dan Evaluasi Model

Evaluasi model dilakukan dengan mengukur Mean Squared Error (MSE) dan Root Mean Square Error (RMSE) pada set uji, dan hasilnya disajikan dalam Tabel 4.3. Metrik MSE dipilih karena kepekaannya terhadap kesalahan prediksi yang besar, sementara RMSE memberikan interpretasi yang lebih intuitif karena satuannya sama dengan target variabel, yaitu kelembaban udara.

Table 4. 3 Tabel Perbandingan Metrik Evaluasi

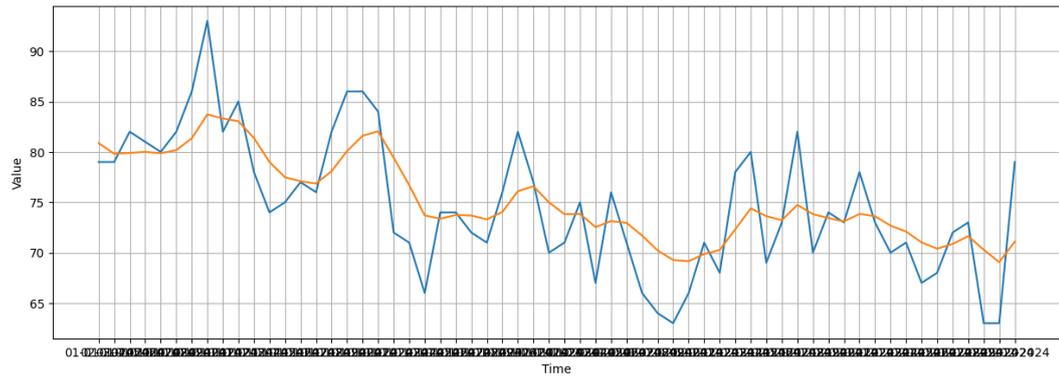
Model	MSE	RMSE	Val MSE	Val RMSE
LSTM	0.0277	0.1663	0.0312	0.1784
CNN	0.0321	0.1791	0.0321	0.1791

Hasil menunjukkan bahwa model LSTM memiliki nilai MSE dan RMSE yang lebih rendah dibandingkan model CNN, yang mengindikasikan performa yang lebih baik dalam hal akurasi prediksi. Grafik konvergensi metrik selama pengujian untuk kedua model ini dapat dilihat pada Gambar 4.12.

**Gambar 4. 12 Plot Perbandingan Metrik Evaluasi MSE & RMSE**

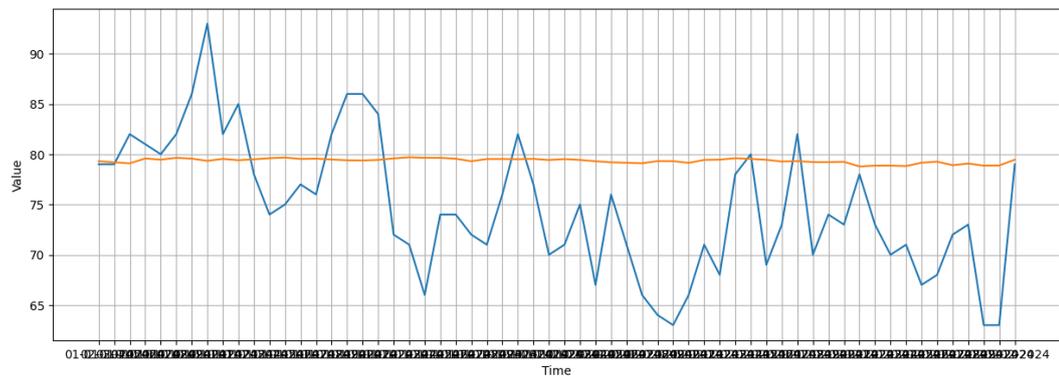
4.7. Analisis dan Interpretasi Hasil

Hasil analisis menunjukkan bahwa model LSTM lebih efektif dalam menangkap pola temporal jangka panjang dalam data kelembaban udara. Kemampuan LSTM untuk mengingat informasi historis dalam jangka waktu yang lebih lama memungkinkannya untuk lebih akurat dalam menangani fluktuasi musiman dan tren. Hal ini menjadikan LSTM sebagai pilihan yang lebih baik untuk memodelkan data yang memerlukan pemahaman mendalam terhadap perubahan yang terjadi dalam jangka panjang.



Gambar 4. 13 Plot Perbandingan data Aktual Dengan Hasil Forecasting Pada Model LSTM

Di sisi lain, model CNN unggul dalam efisiensi pelatihan, dengan kemampuan untuk menangkap pola jangka panjang meskipun tidak sekuat LSTM dalam mengingat informasi historis. CNN lebih cepat dalam proses pelatihan, menjadikannya pilihan yang lebih efisien ketika waktu komputasi menjadi faktor penting.



Gambar 4. 14 Plot Perbandingan data Aktual Dengan Hasil Forecasting Pada Model CNN

BAB V

PENUTUP

5.1. Kesimpulan

Penelitian ini bertujuan untuk menganalisis kinerja algoritma Long Short-Term Memory (LSTM) dan Convolutional Neural Network (CNN) dalam melakukan forecasting kelembaban udara di Kota Medan. Berdasarkan hasil implementasi dan evaluasi yang telah dilakukan, model LSTM Lebih Unggul dalam Akurasi. Hasil evaluasi menunjukkan bahwa model LSTM memiliki nilai Mean Squared Error (MSE) dan Root Mean Squared Error (RMSE) yang lebih rendah dibandingkan model CNN. Nilai MSE pada set uji untuk LSTM adalah 0.0277 dengan RMSE sebesar 0.1663, sedangkan CNN memiliki MSE sebesar 0.0321 dan RMSE sebesar 0.1791. Hal ini menandakan bahwa LSTM lebih efektif dalam menangkap pola temporal jangka panjang dalam data time series kelembaban udara. Meskipun demikian, Teknik windowing yang digunakan dalam penelitian ini, dengan ukuran jendela 90 timestep, terbukti efektif dalam membangun dataset time series yang dapat dimanfaatkan oleh kedua model untuk menangkap pola dalam data kelembaban udara.

5.2. Saran

Berdasarkan hasil penelitian ini, beberapa saran dapat diberikan untuk penelitian lanjutan dan pengembangan aplikasi forecasting kelembaban udara. Untuk meningkatkan akurasi prediksi, disarankan untuk menggunakan data dengan jangka waktu yang lebih panjang atau mengkombinasikan data kelembaban udara dengan variabel cuaca lainnya

seperti suhu, tekanan, dan kecepatan angin. Penelitian lanjutan juga dapat mengeksplorasi pengembangan model hybrid yang menggabungkan keunggulan LSTM dalam menangkap pola temporal dengan kemampuan CNN dalam ekstraksi fitur spasial, yang berpotensi menghasilkan prediksi yang lebih akurat. Dengan keunggulan model LSTM dalam akurasi prediksi, sangat disarankan untuk mengimplementasikan model ini pada sistem forecasting real-time yang dapat digunakan oleh pemerintah atau lembaga terkait dalam pengambilan keputusan terkait mitigasi dampak perubahan cuaca.

DAFTAR PUSTAKA

- Al Haromainy, M. M., Fatichah, C., & Saikhu, A. (2021). Prediksi Data Deret Waktu Multivariat Menggunakan Echo State Network dan Harmony Search. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 19(2), 111–119.
- Alfandi, M., & Sihite, A. M. H. (2022). Penerapan Metode CNN-LSTM Dalam Memprediksi Hujan Pada Wilayah Medan. *Nasional Teknologi Informasi Dan Komputer*, 6(1), 490–499. <https://doi.org/10.30865/komik.v6i1.5713>
- Alizkan, U. (2017). Analisis Korelasi Kelembaban Udara Terhadap Epidem Demam Berdarah Yang Terjai Di Kabupaten dan Kota Serang. *Gravity*, 3(1), 23–29.
- Andrian, Y., & Wayahdi, M. R. (2014). Analisis Kinerja Jaringan Saraf Tiruan Metode Backpropagation dalam Memprediksi Cuaca di Kota Medan. *Snikom*, 2014, 20–24.
- Carneiro, T., Da Nobrega, R. V. M., Nepomuceno, T., Bian, G. Bin, De Albuquerque, V. H. C., & Filho, P. P. R. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access*, 6, 61677–61685. <https://doi.org/10.1109/ACCESS.2018.2874767>
- Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7, 1–24. <https://doi.org/10.7717/PEERJ-CS.623>
- Diandra, D., Atsila, F., Hanif, R., Akhdan, S., & Yudistira, N. (2022). Prediksi Perubahan Iklim di Indonesia pada Tahun 2013-2014 Menggunakan LSTM. / *Jurnal Litbang Edusaintech*, 3(2), 101–106. <http://dx.doi.org/10.51402/jle.v3i2.49>
- Fadholi, A. (2013). Pemanfaatan Suhu Udara dan Kelembaban Udara dalam Persamaan Regresi untuk Simulasi Prediksi Total Hujan Bulanan di Pangkalpinang. *CAUCHY: Jurnal Matematika Murni Dan Aplikasi*, 3(1), 1–9. <https://doi.org/10.18860/ca.v3i1.2565>

- Faraway, J., & Chatfield, C. (1998). Time series forecasting with neural networks: A comparative study using the airline data. *Journal of the Royal Statistical Society. Series C: Applied Statistics*, 47(2), 231–250. <https://doi.org/10.1111/1467-9876.00109>
- Haris, M. S., Khudori, A. N., & Kusuma, W. T. (2022). Perbandingan Metode Supervised Machine Learning untuk Prediksi Prevalensi Stunting di Provinsi Jawa Timur. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 9(7), 1571. <https://doi.org/10.25126/jtiik.2022976744>
- Hastomo, W., Karno, A. S. B., Kalbuana, N., Nisfiani, E., & ETP, L. (2021). Optimasi Deep Learning untuk Prediksi Saham di Masa Pandemi Covid-19. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 7(2), 133. <https://doi.org/10.26418/jp.v7i2.47411>
- Ihsan, C. N. (2021). Klasifikasi Data Radar Menggunakan Algoritma Convolutional Neural Network (CNN). *DoubleClick: Journal of Computer and Information Technology*, 4(2), 115. <https://doi.org/10.25273/doubleclick.v4i2.8188>
- Imaduddin, M. A., Hamami, F., & Fa'Rifah, R. Y. (2023). Prediksi Cuaca Pada Data Time Series Menggunakan Backpropagation Neural Network (Bpnn). *E-Proceeding of Engineering*, 10(5), 4873. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/21333>
- Intan, I., Aminah Dinayati Ghani, S., Koswara, A. T., Dipa Makassar, U., Arsip Nasional Republik Indonesia, K., & Kemerdekaan, J. P. (2021). Analisis Performansi Prakiraan Cuaca Menggunakan Algoritma Machine Learning Performance Analysis of Weather Forecasting using Machine Learning Algorithms. *Jurnal_Pekommas_Vol._6_No, 2*, 1–8. <https://doi.org/10.30818/jpkm.2021.2060221>
- Kanani, P., & Padole, M. (2019). Deep learning to detect skin cancer using google colab. *International Journal of Engineering and Advanced Technology*, 8(6), 2176–2183.

<https://doi.org/10.35940/ijeat.F8587.088619>

- Kareem, S., Hamad, Z. J., & Askar, S. (2021). An evaluation of CNN and ANN in prediction weather forecasting: A review. *Sustainable Engineering and Innovation*, 3(2), 148–159. <https://doi.org/10.37868/sei.v3i2.id146>
- Karyadi, Y. (2022). Prediksi Kualitas Udara Dengan Metoda LSTM, Bidirectional LSTM, dan GRU. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 9(1), 671–684. <https://doi.org/10.35957/jatisi.v9i1.1588>
- Kristiana, L., & Miyanto, D. (2023). Penambahan Parameter PM2.5 dalam Prediksi Kualitas Udara : Long Short Term Memory. *Multimedia Artificial Intelligent Networking Database (MIND)*, 8(2), 188–202.
- Lattifia, T., Wira Buana, P., & Rusjyanthi, N. K. D. (2022). Model Prediksi Cuaca Menggunakan Metode LSTM. *JITTER Jurnal Ilmiah Teknologi Dan Komputer*, 3(1), 994–1000.
- Lavanya, A., Gaurav, L., Sindhuja, S., Seam, H., Joydeep, M., Uppalapati, V., Ali, W., & S.D, V. (2023). Assessing the Performance of Python Data Visualization Libraries: A Review. *International Journal of Computer Engineering in Research Trends*, 10(1), 28–39. <https://doi.org/10.22362/ijcert/2023/v10/i01/v10i0104>
- M Devid Alam Carnegie, & Chairani. (2023). Perbandingan Long Short Term Memory (LSTM) dan Gated Recurrent Unit (GRU) Untuk Memprediksi Curah Hujan. *Jurnal Media Informatika Budidarma*, 7(3), 1022–1032. <https://doi.org/10.30865/mib.v7i3.6213>
- Muhammad Haris Diponegoro, Sri Suning Kusumawardani, & Indriana Hidayah. (2021). Tinjauan Pustaka Sistematis: Implementasi Metode Deep Learning pada Prediksi Kinerja Murid. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 10(2), 131–138. <https://doi.org/10.22146/jnteti.v10i2.1417>
- Mujiasih, S. (2011). Pemanfaatan Data Mining Untuk Prakiraan Cuaca. *Jurnal Meteorologi Dan Geofisika*, 12(2). <https://doi.org/10.31172/jmg.v12i2.100>

- Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia. *Algor*, 2(1), 12–21.
- Riadi, J., & Nurmahaludin. (2012). Aplikasi Jaringan Syaraf Tiruan Multi Layer Perceptron Pada Aplikasi Prakiraan Cuaca. *Poros Teknik*, 4(1), 71–76.
- Rizki, M., Basuki, S., & Azhar, Y. (2020). Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory(LSTM) Untuk Prediksi Curah Hujan Kota Malang. *Jurnal Repositor*, 2(3), 331–338. <https://doi.org/10.22219/repositor.v2i3.470>
- Roihan, A., Sunarya, P. A., & Rafika, A. S. (2020). Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 5(1), 75–82. <https://doi.org/10.31294/ijcit.v5i1.7951>
- Romzi, M., & Kurniawan, B. (2020). Implementasi Pemrograman Python Menggunakan Visual Studio Code. *Jurnal Informatika Dan Komputer (JIK)*, 11(2), 1–9. www.python.org
- Sial, A. H., Yahya, S., & Rashdi, S. (2021). Comparative Analysis of Data Visualization Libraries Matplotlib and Seaborn in Python. *International Journal of Advanced Trends in Computer Science and Engineering*, 10(1), 277–281. <https://doi.org/10.30534/ijatcse/2021/391012021>
- Widiputra, H., Adele Mailangkay, & Elliana Gautama. (2021). Prediksi Indeks BEI dengan Ensemble Convolutional Neural Network dan Long Short-Term Memory. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(3), 456–465. <https://doi.org/10.29207/resti.v5i3.3111>
- Zhao, B., Lu, H., Chen, S., Liu, J., & Wu, D. (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1), 162–169. <https://doi.org/10.21629/JSEE.2017.01.1>

LAMPIRAN



UMSU
Unggul | Cerdas | Terpercaya

Bila menyalin surat ini agar disebutkan nomor dan tanggalnya

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

<https://fiki.umsu.ac.id> fiki@umsu.ac.id [umsumedan](#) [umsumedan](#) [umsumedan](#) [umsumedan](#)

PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 459/IL.3-AU/UMSU-09/F/2024

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi : Sistem Informasi
Pada tanggal : 14 Mei 2024

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama : Muhammad Elfikry
NPM : 2009010121
Semester : VIII (Delapan)
Program studi : Sistem Informasi
Judul Proposal / Skripsi : Analisis Kinerja Algoritma LSTM dan CNN untuk Forecasting Kelembapan Udara di Kota Medan

Dosen Pembimbing : Dr. Al-Khowarizmi, M.Kom

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi dinyatakan " BATAL "** bila tidak selesai sebelum Masa Kadalursa tanggal : **14 Mei 2025**
4. Revisi judul.....

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
Pada Tanggal : 06 Dzulqa'dah 1445 H
14 Mei 2024 M



a.n.Dekan
Wakil Dekan I

H. Maulana. S.T., M.Kom
NIDN : 0121119102

Cc. File



UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
 Nomor : 622/H.3-AU/UMSU-09/G/2024

UNDANGAN SEMINAR PROPOSAL

Fakultas : Ilmu Komputer dan Teknologi Informasi
 Program Studi : Sistem Informasi
 Hari/Tanggal : Kamis, 1 Agustus 2024
 Waktu /Tempat : 09.00/C
 Pemimpin Seminar : **Martiano, S.Pd., S.Kom., M.Kom**

No	NPM	NAMA MAHASISWA	Dosen Pembimbing	Dosen Pembahas	JUDUL PROPOSAL
5	2009010075	Herdawani Aldilla	Wilda Rina HSB S.T., M.Kom	Dr. Al-Khowarizmi, S.Kom., M.Kom	Analisis Dan Perbandingan Kinerja Algoritma K-Means Dan Algoritma X-Means Dalam Pengklasifikasian Rawat Inap Di RS. Mitra Medika
6	2009010133	Muhammad Farhan Syadwi	Halim Maulana, S.T., M.Kom	Fatma Sari Hutagalung, S.Kom., M.Kom	Analisis perbandingan kinerja algoritma cnn dan rnn dalam deteksi wajah
7	2009010121	Muhammad Elfikry	Dr. Al-Khowarizmi, S.Kom., M.Kom	Halim Maulana, S.T., M.Kom	Analisis Kinerja Algoritma LSTM dan CNN untuk Forecasting Kelembaban Udara di Kota Medan
8	2009010134	Raihan Oza Samudera Siregar	Farid Akbar Siregar, S.Kom., M.Kom	Martiano, S.Pd., S.Kom., M.Kom	Perancangan Chatbot menggunakan Metode Natural Language Processing (NLP) untuk Pelayanan Informasi pada Lazisnu Kota Medan menggunakan Artificial Intelligence



NB: - Laki-laki berbusana hitam putih dan memakai dasi
 Perempuan berbusana muslimah hitam putih

Medan, 23 Muharram 1446 H
 29 Juli 2024 M



Dean
 Al-Khowarizmi, M.Kom
 NIDN : 127099201



UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
Nomor : 726/IL.3-AU/UMSU-09/G/2024
HAL UJIAN MEJA HIJAU SARJANA (S1)



UMSU
 Berani Berprestasi

Fakultas : Ilmu Komputer dan Teknologi Informasi
 Program Studi : Sistem Informasi
 Hari/Tanggal : Sabtu, 03 September 2024
 Waktu /Tempat : 09:00-14:00 WIB/G

UNDANGAN
PANGGILAN

Kepada Yang Terhormat
 Bapak/Ibu Dosen Penguji Meja Hijau
 di
 Medan
 Catatan :
 *Harap datang tepat waktu karena ujian
 dalam bentuk tum (2 Orang) penguji I & II
 *Dosen Penguji yang terlambat 30 menit
 akan diganti

No	NAMA MAHASISWA	NPM	TOPIC/SKRIPSI	DOSEN PENGUJI I	DOSEN PENGUJI II	DOSEN PEMBIMBING/PENGUJI III
11	Raihan Oza Samudera Siregar	2009010134	Perancangan Chatbot menggunakan Metode Natural Language Processing (NLP) untuk Pelayanan Informasi pada Lazismu Kota Medan menggunakan Artificial Intelligence	Martiano, S.Pd., S.Kom., M.Kom	Halim Maulana, S.T., M.Kom	Farid Akbar Siregar, S.Kom., M.Kom
12	Muhammad Elfikry	2009010121	Analisis Kinerja Algoritma LSTM dan CNN untuk Forecasting Kelembaban Udara di Kota Medan	Halim Maulana, S.T., M.Kom	Yoshida Sary, S.Kom., M.Kom	Dr. Al-Khowarizmi, M.Kom
13	Mulhammad Farhan Syadwi	2009010133	Analisis perbandingan kinerja algoritma CNN dan RNN dalam deteksi wajah	Fatma Sari Hutagalung, S.Kom., M.Kom	Dr. Firaumi Rizky, M.Kom	Halim Maulana, S.T., M.Kom
14	M. Ardiansyah	2009010055	Prediksi Penurunan Harga Mobil Bekas Menggunakan Decision Pada Showroom Habib Mobil	Dr. Al-Khowarizmi, M.Kom	Halim Maulana, S.T., M.Kom	Mahardika Abdi Prawira, S.Kom., M.Kom

Asisten Pengambilan Berita Acara :
 1. Suvia Agustini S.I.Kom
 2. Andika Suras Saputra, S.M

Medan, 22 Shafar 1446 H
 27 Agustus 2024 M

Ditandatangani Oleh


 Prof. Dr. **MUHAMMAD ARIFIN, S.H., M. Hum**

Panitia Ujian


 Dr. **Al-Khowarizmi, M.Kom**

Sekretaris
 Wakil Dekan I


 Halim **MAULANA, S.T., M.Kom**