PERBANDINGAN ALGORITMA *EXTREME GRADIENT BOOSTING* (XGBOOST) DAN ALGORITMA *EXTREME LEARNING MACHINE* (ELM) DALAM MEMPREDIKSI TINGKAT CURAH HUJAN DI KOTA MEDAN

SKRIPSI

TRIDINDA PERMATASARI
201401028

PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024

PERBANDINGAN ALGORITMA *EXTREME GRADIENT BOOSTING* (XGBOOST) DAN ALGORITMA *EXTREME LEARNING MACHINE* (ELM) DALAM MEMPREDIKSI TINGKAT CURAH HUJAN DI KOTA MEDAN

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana Ilmu Komputer

TRIDINDA PERMATASARI
201401028

PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024

i

PERSETUJUAN

Judul	: PERBANDINGAN ALGORITMA *EXTREME GRADIENT BOOSTING* (XGBOOST) DAN ALGORITMA *EXTREME LEARNING MACHINE* (ELM) DALAM MEMPREDIKSI TINGKAT CURAH HUJAN DI KOTA MEDAN
Kategori	: SKRIPSI
Nama	: TRIDINDA PERMATASARI
Nomor Induk Mahasiswa	: 201401028
Program Studi	: SARJANA (S-1) ILMU KOMPUTER
Fakultas	: ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

Diluluskan di
Medan, 11 Juni 2024

Komisi Pembimbing :

Pembimbing 2	Pembimbing 1
Dr. Maya Silvi Lydia, B.Sc., M.Sc.	Prof. Dr. Syahril Efendi, S.Si., M.I.T.
NIP.197401272002122001	NIP. 196711101996021001

Diketahui/Disetujui Oleh
Program Studi S-1 Ilmu Komputer
Ketua,

Dr. Amalia S.T., M.T.
NIP. 197812212014042001

PERNYATAAN

PERBANDINGAN ALGORITMA *EXTREME GRADIENT BOOSTING* (XGBOOST) DAN ALGORITMA *EXTREME LEARNING MACHINE* (ELM) DALAM MEMPREDIKSI TINGKAT CURAH HUJAN DI KOTA MEDAN

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil penelitian saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah dicantumkan sumbernya.

Medan, 11 Juni 2024

Tridinda Permatasari
201401028

UCAPAN TERIMA KASIH

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Penyayang, penulis mengucapkan segala puji dan syukur kepada-Nya atas rahmat dan hidayah yang telah diberikan, sehingga penulis dapat menyelesaikan masa perkuliahan dan menyusunan skripsi ini. Shalawat dan salam, semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW, yang telah menerangi hati nurani kita dan menjadi cahaya bagi segala perbuatan mulia.

Pada kesempatan ini juga penulis memberikan ucapan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bimbingan dan bantuan atas kekurangan yang dilakukan oleh penulis:

1. Bapak Prof. Dr. Muryanto Amin, S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara sekaligus Dosen Pembimbing II yang telah membimbing, memotivasi serta memberikan dukungan dan arahan kepada penulis sehingga penulis dapat menyelesaikan penyusunan skripsi ini.
3. Ibu Dr. Amalia, S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Sri Melvani Hardi S.Kom., M.Kom. selaku Sekretaris Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Prof. Dr. Syahril Efendi, S.Si., M.I.T. selaku Dosen Pembimbing I yang telah membimbing, memotivasi serta memberikan dukungan dan arahan kepada penulis sehingga penulis dapat menyelesaikan penyusunan skripsi ini.
6. Kedua orang tua penulis, Ibu Susi Syahlawati dan Bapak M. Irwan yang senantiasa memberikan do’a, kasih sayang, dan kebahagiaan kepada penulis.

7. Ibu Dr. Elviawaty Muisa Zamzami, S.T., M.T., MM., IPU, selaku Dosen Pembimbing Akademik yang telah membimbing penulis dari awal masa perkuliahan hingga penyusunan skripsi ini.
8. Seluruh Bapak dan Ibu Dosen Fasilkom-TI, khususnya Dosen Program Studi S-1 Ilmu Komputer yang telah mendidik, memberi wawasan serta moral yang berharga, dan memberikan waktu dan tenaga untuk mengajar dan membimbing sehingga penulis dapat sampai kepada tahap penyusunan skripsi ini.
9. Seluruh Staf Pegawai Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang telah banyak memberikan bantuan kepada penulis selama masa perkuliahan.
10. Sahabat dan teman-teman seperkuliahan penulis, Amru Khair Al Hakim, Syabrina Ramadhani Kamal, Rani Syafitri, Ayu Prasinta, Farrel Dwi Prayogo, Haryanda Fidi, Mhd. Habib Laits Yafa, Salsabila Humairah, dan seluruh teman-teman stambuk 2020 yang telah memberikan pengalaman belajar serta kepanitiaan bersama yang berharga kepada penulis.
11. Teman-teman dekat penulis, Putri Sakinah, Nur Assakinah, Shafira Putri dan abang-abang TOM yang telah memberikan semangat dan bantuan kepada penulis.
12. Sahabat penulis, Putri Amini Dalimunthe yang telah menemani, mendengarkan keluh kesah dan membantu penulis dalam menjalani kehidupan dari SMA sampai saat ini.

Serta kepada semua pihak yang telah membantu penulis yang belum dapat penulis sertakan satu-persatu. Semoga Allah SWT memberikan keberkahan atas kebaikan dan bantuan yang telah diberikan kepada penulis.

Medan, 11 Juni 2024

Tridinda Permatasari
201401028

ABSTRAK

Curah hujan adalah fenomena alam yang memiliki dampak signifikan pada berbagai aspek kehidupan manusia, termasuk pertanian, manajemen air, dan mitigasi bencana, khususnya di Kota Medan. Kota Medan merupakan salah satu kota yang rentan akan terjadinya banjir. Jika turun hujan dengan intensitas yang tinggi maka dapat dipastikan bahwa sebagian besar dari wilayah Medan akan digenangi banjir. Selain itu, iklim tropis dengan musim hujan yang intens juga meningkatkan risiko penyebaran penyakit seperti malaria. Penelitian ini bertujuan untuk membandingkan performa dua algoritma *machine learning*, yaitu *Extreme Gradient Boosting* (XGB) dan *Extreme Learning Machine* (ELM), dalam memprediksi tingkat curah hujan di Kota Medan. Data diambil dari *website* BMKG *online* pada Stasiun Balai Besar Meteorologi Klimatologi dan Geofisika Wilayah I di mulai dari tanggal tanggal 1 Januari 2022 – 31 Desember 2023 di Kota Medan. Situasi ini mendorong untuk membuat model prediksi curah hujan yang tepat agar dapat memberikan rekomendasi tindakan yang sesuai. Penelitian sebelumnya tentang prediksi curah hujan telah menggunakan model yang memiliki keterbatasan, sehingga menghasilkan kinerja yang kurang memuaskan. Berdasarkan hasil penelitian, nilai evaluasi MSE, RMSE, MAE yang diperoleh dari algoritma *Extreme Learning Machine* lebih rendah dan mendekati nilai 0 dibandingkan algoritma *Extreme Gradient Boosting*. Nilai akurasi yang di peroleh dari algoritma *Extreme Learning Machine* sebesar 95.75%, sedangkan nilai akurasi dari algoritma *Extreme Gradient Boosting* hanya 95.03 %. Penelitian ini menunjukkan algoritma ELM lebih baik digunakan dalam prediksi karena memiliki nilai evaluasi MSE, RMSE, MAE yang lebih kecil dan nilai akurasi yang tinggi dibandingkan dengan algoritma XGBoost.

Kata Kunci : Prediksi, Tingkat curah hujan, *Extreme Gradient Boosting*, *Extreme Learning Machine*.

COMPARISON OF THE EXTREME GRADIENT BOOSTING (XGBOOST) AND THE EXTREME LEARNING MACHINE (ELM) ALGORITHM IN PREDICTING RAINFALL LEVELS IN MEDAN CITY

ABSTRACT

Rainfall is a natural phenomenon that has a significant impact on various aspects of human life, including agriculture, water management and disaster mitigation, especially in the city of Medan. The city of Medan is one of the cities that is vulnerable to flooding. If it rains with high intensity, it is certain that most of the Medan area will be flooded. In addition, the tropical climate with intense rainy seasons also increases the risk of spreading diseases such as malaria. This research aims to compare the performance of two machine learning algorithms, namely Extreme Gradient Boosting (XGB) and Extreme Learning Machine (ELM), in predicting rainfall levels in Medan City. Data was taken from the BMKG online website at the Center for Meteorology, Climatology and Geophysics Region I starting from January 1 2022 – December 31 2023 in Medan City. This situation encourages the need to create appropriate rainfall prediction models in order to provide recommendations for appropriate action. Previous research on rainfall prediction has used models that have limitations, resulting in unsatisfactory performance. Based on the research results, the MSE, RMSE, MAE evaluation values obtained from the Extreme Learning Machine algorithm are lower and closer to 0 compared to the Extreme Gradient Boosting algorithm. The accuracy value obtained from the Extreme Learning Machine algorithm is 95.75%, while the accuracy value from the Extreme Gradient Boosting algorithm is only 95.03%. This research shows that the ELM algorithm is better used in predictions because it has smaller MSE, RMSE, MAE evaluation values and high accuracy values compared to the XGBoost algorithm.

Keywords: Prediction, Rainfall level, Extreme Gradient Boosting, Extreme Learning Machine.

DAFTAR ISI

PERSETUJUAN i
PERNYATAAN i
UCAPAN TERIMA KASIH iii
ABSTRAK v
ABSTRACT vi
DAFTAR ISI vii
DAFTAR TABEL x
DAFTAR GAMBAR xi
DAFTAR LAMPIRAN xiii
BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Rumusan Masalah 4
1.3. Batasan Masalah 4
1.4. Tujuan Penelitian 5
1.5. Manfaat Penelitian 5
1.6. Metodologi Penelitian 6
1.7. Sistematika Penulisan 7
BAB II LANDASAN TEORI 8
2.1. Pemanfaatan *Machine Learning* dalam Perkiraan Curah Hujan 8
2.2. *Machine Learning* 9
2.3. *Extreme Gradient Boosting* (XGBoost) 10
2.4. Jaringan Syaraf Tiruan 12
2.5. *Extreme Learning Machine* (ELM) 13
2.6. Pra-Pemrosesan Dataset 15
2.6.1. Analisis Data Eksplorasi 16
2.6.2. Normalisasi Dataset 16
2.7. Denormalisasi Dataset 16
2.8. Evaluasi 17
2.8.1. *Mean Square Error* (MSE) 17
2.8.2. *Root Mean Square Error* (RMSE) 17

2.8.3. *Mean Absolute Error* (MAE) 18
2.9. Penelitian Relevan 18
BAB III ANALISIS DAN PERANCANGAN SISTEM 21
3.1. Gambaran Umum Penelitian 21
3.2. Pengumpulan Data 22
3.3. Pra-Pemrosesan Dataset 23
3.3.1. Analisis Data Eksplorasi 23
3.3.2. Normalisasi Dataset 23
3.3.3. Pembagian Dataset 24
3.4. Proses Prediksi Tingkat Curah Hujan 24
3.5. Denormalisasi Dataset 27
3.6. Evaluasi 27
3.6.1. *Mean Square Error* (MSE) 27
3.6.2. *Root Mean Square Error* (RMSE) 27
3.6.3. *Mean Absolute Error* (MAE) 28
BAB IV IMPLEMENTASI DAN PENGUJIAN 29
4.1. Implementasi Sistem 29
4.2. Pra-Pemrosesan Dataset 29
4.2.1. Analisis Data Eksplorasi 29
4.2.2. *Handle Missing Values* 33
4.2.3. Imputasi nilai *Missing Values* 34
4.2.4. Normalisasi Dataset 35
4.2.5. Pembagian Dataset 36
4.3. Algoritma *Extreme Gradient Boosting* 36
4.3.1. Implementasi Algoritma *Extreme Gradient Boosting* (XGBoost) 37
4.3.2. Evaluasi Algoritma XGBoost 39
4.4. Algoritma *Extreme Learning Machine* 40
4.4.1. Implementasi Algoritma *Extreme Learning Machine* (ELM) 41
4.4.2. Evaluasi Algoritma ELM 45
4.5. Proses Prediksi dan Hasil Prediksi Tingkat Curah Hujan di Kota Medan 46
4.6. Hasil Perbandingan Algoritma *Extreme Gradient Boosting* dan Algoritma *Extreme Learning Machine* 51
BAB V KESIMPULAN DAN SARAN 54

5.1. Kesimpulan 54
5.2. Saran 55
DAFTAR PUSTAKA 56
LAMPIRAN 58

DAFTAR TABEL

Tabel 2.1 Kategori Curah Hujan 9
Tabel 3.1 Contoh Dataset BMKG Stasiun Balai Besar Meteorologi Klimatologi dan Geofisika Wilayah I 23
Tabel 4.1 Deskripsi Statistik 30
Tabel 4.2 Hasil pemeriksaan data.....34
Tabel 4.3 Hasil akhir pemeriksaan data.....35
Tabel 4.4 Contoh *True value Predict Value* serta *Error* dan *Accuracy Percentage* Algoritma XGBoost.....38
Tabel 4.5 Contoh *True value, Predict Value* serta *Error* dan *Accuracy Percentage* Algoritma ELM 43
Tabel 4.6 Percobaan nilai MSE dari *hidden layer*.....45
Tabel 4.7 Hasil prediksi tingkat curah hujan menggunakan Algoritma *Extreme Gradient Boosting* 49
Tabel 4.8 Hasil prediksi tingkat curah hujan menggunakan Algoritma *Extreme Learning Machine* 50
Tabel 4.9 Perbandingan Hasil nilai evaluasi dari Algoritma XGBoost dan Algoritma ELM.....52
Tabel 4.10 Perbandingan Hasil nilai akurasi dari Algoritma XGBoost dan Algoritma ELM.....52

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Jaringan *Artificial Neural Network* 13
Gambar 2.2 Arsitektur Jaringan *Extreme Learning Machine* 14
Gambar 3.1 Gambaran Umum Penelitian 22
Gambar 3.3 *Flowchart* penelitian prediksi tingkat curah hujan.....25
Gambar 4.1 Proses Deskripsi Statistik.....30
Gambar 4.2 Distribusi Curah Hujan 31
Gambar 4.3 Korelasi Antar Fitur.....32
Gambar 4.4 Visualisasi plot 33
Gambar 4.5 Proses *missing values* 33
Gambar 4.6 Imputasi nilai *missing values* 34
Gambar 4.7 Proses normalisasi dataset algoritma XGBoost 35
Gambar 4.8 Proses normalisasi dataset algoritma ELM.....36
Gambar 4.9 Proses pembagian dataset 36
Gambar 4.10 *Source code library* algoritma XGBoost 36
Gambar 4.11 Melatih model Algoritma XGBoost 37
Gambar 4.12 Proses prediksi model menggunakan Algoritma XGBoost 37
Gambar 4.13 Grafik *Predicted vs True Value* Algoritma XGBoost 39
Gambar 4.14 Proses dan hasil akurasi Algoritma XGBoost.....39
Gambar 4.15 Proses dan hasil MSE Algoritma XGBoost 40
Gambar 4.16 Proses dan hasil RMSE Algoritma XGBoost 40
Gambar 4.17 Proses dan hasil MAE Algoritma XGBoost 40
Gambar 4.18 *Source code library* Algoritma ELM.....41
Gambar 4.19 Mengkonversi data menjadi tensor *Pytorch*.....41
Gambar 4.20 Proses mendefinisikan model ELM.....42
Gambar 4.21 Proses pelatihan model ELM 42
Gambar 4.22 Proses prediksi model menggunakan Algoritma ELM.....43
Gambar 4.23 Grafik *Predicted vs True Value* Algoritma ELM.....44
Gambar 4.24 Proses dan hasil akurasi Algoritma ELM 44
Gambar 4.25 Proses dan nilai evaluasi MSE Algoritma ELM 46
Gambar 4.26 Proses dan nilai evaluasi RMSE Algoritma ELM.....46
Gambar 4.27 Proses dan nilai evaluasi MAE Algoritma ELM.....46

Gambar 4.28 Proses prediksi tingkat curah hujan menggunakan Algoritma XGBoost 47
Gambar 4.29 Proses prediksi tingkat curah hujan menggunakan Algoritma ELM.....47
Gambar 4.30 Proses denormalisasi.....48
Gambar 4.31 Proses klasifikasi tingkat curah hujan algoritma XGBoost 48
Gambar 4.32 Proses klasifikasi tingkat curah hujan algoritma ELM.....48
Gambar 4.33 Visualisasi hasil prediksi tingkat curah hujan menggunakan Algoritma XGBoost 50
Gambar 4.34 Visualisasi hasil prediksi tingkat curah hujan menggunakan Algoritma ELM.....51

DAFTAR LAMPIRAN

Lampiran 1 Dataset BMKG Periode 1 Januari 2022 - 31 Desember 2023.....58
Lampiran 2 Data Hasil *True Value, Predict Value, Error* dan *Accuracy Percentage* menggunakan Algoritma XGBOOST.....75
Lampiran 3 Data Hasil *True Value, Predict Value, Error* dan *Accuracy Percentage* menggunakan Algoritma ELM.....78

BAB I

PENDAHULUAN

1.1. Latar Belakang

Indonesia merupakan negara yang terletak di garis khatulistiwa dan memiliki dua musim, yaitu musim hujan dan musim kemarau. Kedua musim ini dapat mempengaruhi kehidupan mahluk hidup di wilayan Indonesia. Salah satu faktor yang menyebabkan adanya musim tersebut adalah curah penyebab musim tersebut adalah curah hujan.

Curah hujan adalah fenomena alam yang memiliki dampak signifikan pada berbagai aspek kehidupan manusia, termasuk pertanian, manajemen air, dan mitigasi bencana, khususnya di Kota Medan. Kota Medan merupakan salah satu kota yang yang sering dilanda berbagai bencana alam, salah satunya adalah banjir yang kerap terjadi akibat curah hujan yang tinggi. Jika turun hujan dengan intensitas yang tinggi maka dapat dipastikan bahwa sebagian besar dari wilayah Medan akan digenangi banjir.

Selain itu, iklim tropis dengan musim hujan yang intens juga meningkatkan risiko penyebaran penyakit seperti malaria. Malaria adalah salah satu penyakit menular yang tetap menjadi masalah signifikan di Indonesia, dalam level kontrol (Sulasmi et al., 2019).

Badan Meteorologi, Klimatologi dan Geofisika (BMKG) adalah lembaga yang mengelola informasi terkait data gempa, iklim, informasi cuaca, informasi suhu udara, kelembaban, lama penyinaran matahari, tekanan udara di seluruh wilayah Negara Kesatuan Republik Indonesia (NKRI). BMKG melakukan pencatatan dan pembaruan data harian serta dalam periode tertentu seperti prakiraan cuaca harian, mingguan, dan iklim, yang dapat digunakan untuk keperluan penelitian (Ikhwani, 2017).

Pemerintah daerah juga memerlukan prediksi curah hujan untuk merencanakan infrastruktur yang tahan cuaca, seperti *drainase*, bendungan, dan jaringan jalan raya. Dengan data yang akurat, mereka dapat memitigasi risiko kerusakan infrastruktur akibat cuaca ekstrem. Pemerintah daerah dan nasional juga menggunakan prediksi curah hujan untuk merencanakan respon terhadap bencana alam. Pemerintah dapat mengambil tindakan pencegahan seperti evakuasi dini, persediaan bantuan darurat dan mengalokasikan sumber daya untuk menangani dampak banjir, tanah longsor, atau kekeringan yang diakibatkan oleh curah hujan yang ekstrem. Kemampuan untuk memprediksi curah hujan dengan tepat adalah kunci untuk mengambil tindakan yang diperlukan dalam menghadapi perubahan cuaca dan mengurangi dampak negatifnya.

Saat ini Badan Meteorologi Klimatologi dan Geofisika (BMKG) mengoperasikan beberapa jenis peralatan untuk pengamatan data curah hujan, terdiri dari peralatan otomatis dan manual, baik yang bersifat konvensional maupun menggunakan teknologi canggih. Salah satu alatnya yaitu, ombrometer.

Selama ini penakar hujan ombrometer yang digunakan di BMKG memiliki beberapa kelemahan, diantaranya ombrometer merupakan alat ukur curah hujan yang dilakukan secara manual, sehingga memerlukan campur tangan manusia dalam pengelolaan data. Alat ini tidak terlalu efisien untuk representasi secara khusus. Alat ini juga kurang praktis dan efisien untuk diterapkan, karena membutuhkan waktu dan tenaga kerja secara rutin. Selain itu data yang diukur dari ombrometer terkadang tidak akurat karena adanya *human erorr* atau kesalahan manusia dalam penghitungan data curah hujan (Fenia, 2023).

Peramalan cuaca di masa depan terutama terkait curah hujan dapat menjadi fokus perhatian banyak peneliti, guna untuk mengurangi dampak negatif dari bencana alam serta membantu dalam pengendalian penyakit malaria yang timbul dari curah hujan. Curah hujan adalah isu utama dalam layanan meteorologi, khususnya layanan peringatan bahaya yang disebabkan oleh hujan.

Memprediksi curah hujan menjadi sangat penting. Prediksi yang akurat dapat memberikan informasi yang dibutuhkan untuk tindakan pencegahan dan mitigasi, sehingga masyarakat dapat lebih siap menghadapi tantangan yang ditimbulkan oleh kondisi cuaca yang ekstrem dan penyakit yang berhubungan dengan lingkungan.

Extreme Gradient Boosting, atau singkatnya XGBoost adalah implementasi sumber terbuka yang efisien dari algoritma peningkatan gradien yang dikembangkan oleh Tianqi Chen dan Carlos Guestrin (Jange, 2021). Metode XGBoost merupakan algoritma pengembangan dari *gradient tree boosting* yang berbasis pada algoritma *ensemble*. Secara efektif, XGBoost mampu menangani kasus *machine learning* yang berskala besar (Herni Yulianti et al., 2022).

Extreme Learning Machine (ELM) adalah sekumpulan metode pembelajaran yang diawasi. Metode ini adalah inovasi baru dari Jaringan Syaraf Tiruan atau *Artificial Neural Network* (ANN) yang pertama kali dicetuskan oleh Huang (2004). Algoritma ELM menggunakan dua proses utama, yaitu proses pelatihan (*training*) dan proses pengujian (*testing*) untuk dapat memperoleh hasil peramalan atau prediksi yang akurat (Dewi et al., 2018).

Prakiraan curah hujan menjadi salah satu tantangan ilmiah yang utama bagi teknologi di seluruh dunia pada abad terakhir. Oleh karena itu, diperlukannya teknologi otomatisasi untuk mengukur curah hujan secara langsung saat hujan terjadi, merekam data curah hujan secara *real time* tanpa harus melakukan pembacaan dan perhitungan secara manual. XGBoost menggunakan pendekatan *ensemble tree-based*, sementara ELM menggunakan pendekatan *neural network* dengan satu layer tersembunyi.

Dengan membandingkan XGBoost dan ELM secara mendalam, penelitian ini diharapkan dapat memberikan wawasan yang berharga bagi para pemangku kepentingan dalam pengembangan sistem prediksi curah hujan di Kota Medan. Hasil penelitian ini dapat menjadi landasan untuk memilih algoritma yang paling terbaik sesuai dengan karakteristik data cuaca kota ini, sehingga dapat meningkatkan kualitas prediksi dan dapat membantu dalam pengelolaan risiko dan pengambilan keputusan yang lebih baik terkait dampak cuaca.

Beberapa penelitian pernah dilakukan sebelumnya, yaitu penelitian yang dilakukan oleh M T Anwar, E Winarno, W Hadikurniawati dan M Novita pada tahun 2021 telah melakukan penelitian yang berjudul "*Rainfall prediction using Extreme Gradient Boosting*". Penelitian dilakukan di Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) Tanjung Mas, Kota Semarang.

Pada penelitian Rich Juniadi Domitri Simamora, Tibyani, dan Sutrisno pada tahun 2019 melakukan penelitian yang berjudul "Peramalan Curah Hujan Menggunakan Metode *Extreme Learning Machine*". Penelitian dilakukan di Badan Meteorologi Klimatologi dan Geofisika (BMKG) stasiun klimatologi Karangploso Malang pada daerah Poncokusumo.

Penelitian yang telah dilakukan memberikan wawasan tentang algoritma XGBoost dan *Extreme Learning Machine* (ELM) dalam konteks prediksi. XGBoost, sebagai model *boosting* yang kuat, menunjukkan kinerja yang sangat baik dalam menangani kompleksitas dan non-linearitas dalam data. Sedangkan *Extreme Learning Machine* (ELM) dengan kecepatan pelatihan yang tinggi dan sifatnya yang efisien dalam menangani dataset besar. ELM memberikan hasil yang memuaskan dengan pendekatan yang sederhana dan cepat dilatih.

Pada penelitian ini akan dirancang untuk membandingkan kinerja dari dua algoritma yaitu algoritma XGBoost dan algoritma ELM untuk mendapatkan hasil akhir pengujian yang paling akurat dalam memprediksi tingkat curah hujan di Kota Medan.

1.2. Rumusan Masalah

Berdasarkan latar belakang, maka dirumuskan masalah dari penelitian ini, yaitu : Pentingnya untuk mengetahui curah hujan yang akurat karena prediksi curah hujan memiliki dampak yang signifikan dalam berbagai aspek kehidupan manusia, lingkungan, dan ekonomi. Maka diperlukannya prediksi curah hujan di kota medan sehingga masyarakat dan pemerintah dapat mendeteksi jika terjadi curah hujan yang ekstrem yang dapat mengakibatkan bencana alam dan penyakit yang akan timbul serta mengetahui langkah selanjutnya.

1.3. Batasan Masalah

Batasan masalah yang ditentukan berdasarkan rumusan masalah yaitu :

1. Data diambil dari website BMKG (Badan Meteorologi, Klimatologi dan Geofisika) pada Stasiun Balai Besar Meteorologi Klimatologi dan Geofisika Wilayah I dengan data harian di mulai dari tanggal tanggal 1 Januari 2022 – 31 Desember 2023 di Kota Medan.

2. Khusus menggunakan beberapa data yaitu, temperatur minimum(°C), temperatur maksimum (°C), temperatur rata-rata (°C), kelembapan rata-rata (%), curah hujan (mm), lamanya penyinaran matahari (jam), kecepatan angin maksimum (m/s), arah angin saat kecepatan maksimum (°), kecepatan angin rata-rata (m/s).
3. Bahasa yang digunakan untuk membangun program dan model adalah bahasa pemrograman *Python*.
4. Penelitian ini menerapkan algoritma *Extreme Gradient Boosting* (XGBoost) dan algoritma *Extreme Learning Machine* (ELM) serta *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE) dan *Mean Absolute Error* (MAE) untuk mengevaluasi hasil.
5. Penelitian ini hanya membahas model dan hasil akurasi yang terbaik di antara algoritma *Extreme Gradient Boosting* (XGBoost) dan algoritma *Extreme Learning Machine* (ELM).

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah menerapkan metode algoritma *Extreme Gradient Boosting* (XGBoost) dan *Extreme Learning Machine* (ELM) untuk mengukur nilai kinerja dan melakukan prediksi curah hujan di Kota Medan serta membandingkan hasil nilai akurasi prediksi antara XGBoost dan ELM untuk mengidentifikasi algoritma yang lebih efektif dalam konteks prediksi curah hujan di Kota Medan.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

1. Dapat memberikan pengetahuan terkait metode *Extreme Gradient Boosting* (XGBoost) dan *Extreme Learning Machine* (ELM) dalam melakukan prediksi curah hujan.
2. Memprediksi curah hujan yang akurat dapat membantu dalam peringatan dini terhadap potensi bencana alam, seperti banjir.
3. Dapat memberikan informasi kepada masyarakat tentang curah hujan, seperti potensi peningkatan risiko penyakit malaria saat terjadinya periode curah hujan tinggi.
4. Mengetahui hasil kemampuan prediksi curah hujan dengan nilai akurasi yang baik.

1.6. Metodologi Penelitian

Tahapan-tahapan metodologi penelitian :

1. **Studi Literatur**

   Pada tahap studi literatur, penulis melakukan pengumpulan beberapa referensi yang diperlukan untuk penelitian ini. Tujuan dari langkah ini untuk mendapatkan informasi dan data yang relevan yang akan digunakan dalam penulisan skripsi ini. Referensi yang digunakan mencakup beberapa sumber seperti buku, jurnal, artikel, *paper*, makalah baik berupa media cetak maupun media internet yang berkaitan dengan topik Curah Hujan, Algoritma *Extreme Gradient Boosting* (XGBoost), *Extreme Learning Machine* (ELM), *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE), dan *Mean Absolute Error* (MAE).

2. **Analisa dan Perancangan**

   Pada tahap ini, penulis melakukan evaluasi terhadap kebutuhan yang diperlukan dalam penelitian yang akan dibangun. Evaluasi ini mencakup identifikasi elemen-elemen yang akan terlibat dalam penelitian, dan akan digambarkan dalam diagram arsitektur yang dirancang berdasarkan ruang lingkup penelitian.

3. **Implementasi**

   Pada tahap ini, bahasa pemrograman *Python* akan digunakan untuk mengimplementasikan hal-hal yang telah didefinisikan selama fase analisis dan perancangan untuk menghasilkan model.

4. **Pengujian**

   Pada tahap ini, dilakukan pengujian atau evaluasi menggunakan *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE), dan *Mean Absolute Error* (MAE).untuk memastikan bahwa model dari algoritma *Machine Learning* dengan *Extreme Gradient Boosting* (XGBoost) dan *Extreme Learning Machine* (ELM) dalam memprediksi curah hujan berfungsi sesuai dengan apa yang diharapkan.

5. **Dokumentasi**

   Pada tahap ini dilakukan penyusunan dokumentasi dan penulisan laporan serta kesimpulan hasil akhir dari analisa dan pengujian dalam bentuk skripsi.

1.7. Sistematika Penulisan

BAB 1 PENDAHULUAN

Bab ini merupakan tahapan awal penelitian yaitu dimulai dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, hingga sistematika penulisan. Bab ini juga memberikan gambaran komprehensif tentang kerangka dasar penelitian dan bagaimana penelitian tersebut akan dilaksanakan.

BAB 2 LANDASAN TEORI

Bab ini berisi penjelasan tentang berbagai teori dan konsep yang berkaitan dengan topik penelitian. Pada bab ini akan dijelaskan beberapa teori permasalahan yaitu Prediksi, Curah Hujan, *Machine Learning*, *Data Mining*, *Extreme Gradient Boosting* (XGBoost), algoritma *Extreme Learning Machine* (ELM), dan penelitian yang relevan.

BAB 3 ANALISIS DAN PERANCANGAN

Bab ini fokus pada analisis dan perancangan dari prediksi tingkat curah hujan di kota Medan menggunakan algoritma *Extreme Gradient Boosting* (XGBoost), algoritma *Extreme Learning Machine* (ELM) untuk mendapatkan algortima yang terbaik. Bab ini juga memuat analisis perancangan penelitian dan memberikan gambaran umum tentang perancangan sistem yang akan digunakan dalam penelitian.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Bab ini merupakan bagian penting dalam sebuah penelitian yang fokus pada penerapan dan evaluasi yang praktis dari sistem atau solusi yang telah dirancang. Dalam bab ini, peneliti menjelaskan tentang langkah-langkah sistem yang akan diimplementasikan berdasarkan analisis masalah dan kebutuhan sistem, skenario pengujian untuk sistem yang dikembangkan, dan diskusi tentang temuan pengujian sistem.

BAB 5 KESIMPULAN DAN SARAN

Bab ini mencakup kesimpulan dari keseluruhan hasil penelitian beserta saran yang diberikan dan diharapkan untuk dapat dikembangkan pada penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1. Pemanfaatan *Machine Learning* dalam Perkiraan Curah Hujan

Dalam beberapa tahun terakhir, perkembangan teknologi informasi dan komputasi telah membuka peluang baru dalam bidang prediksi cuaca, termasuk curah hujan. *Machine learning*, sebagai salah satu cabang dari kecerdasan buatan, menawarkan pendekatan baru yang lebih fleksibel dan adaptif dibandingkan dengan metode statistik tradisional. Algoritma *machine learning* memiliki kemampuan untuk memodelkan hubungan kompleks antara berbagai variabel cuaca dan menghasilkan prediksi yang lebih akurat.

Prediksi adalah proses memperkirakan secara sistematis hal-hal yang kemungkinan besar akan terjadi di masa depan berdasarkan informasi yang tersedia dari masa lalu dan sekarang, dengan tujuan meminimalkan kesalahan (selisih antara kejadian sebenarnya dan perkiraan). Prediksi tidak harus memberikan jawaban pasti mengenai kejadian yang akan terjadi, melainkan berupaya untuk mencari jawaban sedekat mungkin dengan kenyataan (Manalu, 2016).

Curah hujan harian adalah pengukuran jumlah hujan yang jatuh di suatu wilayah dalam waktu satu hari. Curah hujan dipengaruhi oleh berbagai faktor seperti suhu udara, tekanan udara, kelembaban, dan pola angin. Data ini sangat penting dalam memprediksi curah hujan di masa depan karena menunjukkan pola hujan dalam jangka waktu yang pendek. Dengan pemantauan rutin terhadap curah hujan harian, para ahli cuaca dapat mengidentifikasi tren jangka pendek dalam pola hujan, seperti periode kekeringan atau hujan lebat yang dapat menyebabkan banjir.

Prediksi curah hujan yang akurat sangat penting untuk berbagai keperluan, seperti perencanaan pertanian, manajemen sumber daya air, serta mitigasi bencana alam.

Selain itu prediksi curah hujan juga penting dalam hal untuk mencegah penyakit yang akan timbul ketika terjadi curah hujan tinggi atau periode hujan ekstrem. Hujan dapat menciptakan genangan air yang menjadi tempat berkembang biaknya nyamuk *Aedes aegypti* (penyebab demam berdarah) dan *Anopheles* (penyebab malaria). Dengan adanya prediksi yang akurat, pihak terkait dapat mengambil langkah-langkah antisipatif untuk mengurangi dampak negatif dari kejadian cuaca ekstrem.

Penelitian ini fokus pada pemanfaatan dua algoritma *machine learning*, yaitu *Extreme Gradient Boosting* (XGBoost) dan *Extreme Learning Machine* (ELM), dalam memprediksi tingkat curah hujan di Kota Medan. Kedua algoritma ini dipilih karena memiliki keunggulan masing-masing dalam menangani dataset yang besar dan kompleks.

Data curah hujan yang diperoleh ditulis dengan satuan milimeter (mm), dimana satu milimeter hujan berarti air hujan yang turun di wilayah seluas satu meter persegi akan memiliki ketinggian satu milimeter jika air hujan tidak meresap, mengalir, atau menguap. Ambang batas nilai yang digunakan untuk menentukan intensitas hujan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Kategori Curah Hujan

No.	Curah Hujan (mm)	Kategori
1	0 mm/hari	Berawan
2	0.5 – 20 mm/hari	Hujan ringan
3	20 – 50 mm/hari	Hujan sedang
4	50 – 100 mm/hari	Hujan lebat
5	100 – 150 mm/hari	Hujan sangat lebat
6	>150 mm/hari	Hujan ekstrem

2.2. *Machine Learning*

Machine Learning didefinisikan sebagai bentuk kecerdasan buatan (AI) yang melibatkan mesin-mesin yang dapat belajar dan berkembang tanpa perlu program secara eksplisit. Algoritma dalam *Machine Learning* digunakan untuk menyelesaikan berbagai jenis masalah dalam mengklasifikasikan data. Perkembangan teknologi

Machine Learning telah mengalami kemajuan pesat dalam beberapa tahun terakhir dan telah diadopsi luas dalam beragam bidang ilmiah. Algoritma *Machine Learning* secara umum dapat dikelompokkan menjadi tiga jenis yaitu pembelajaran terawasi, pembelajaran tanpa pengawasan, dan pembelajaran penguatan. Penerapan *Machine Learning* biasanya berfokus pada data dalam jumlah besar dengan proses perhitungan yang kompleks, seperti dalam pengolahan citra dan klasifikasi (Setiap gambar memiliki jutaan piksel, yaitu jutaan parameter). Jika hanya mengandalkan pada pendekatan statistik konvensional untuk melakukan perhitungan ini, akan muncul kendala besar yaitu jumlah perhitungan yang terlalu besar dan tidak memungkinkan untuk dijalankan. *Machine Learning* bergantung pada penggunaan data dalam jumlah besar untuk pelatihan dan pengembangannya sendiri, sehingga terdapat sedikit ketergantungan pada teori matematika, dan fokus utama *Machine Learning* adalah menyelesaikan permasalahan rekayasa yang melibatkan dataset yang sangat besar (Laird et al., 2017).

2.3. *Extreme Gradient Boosting* (XGBoost)

Extreme Gradient Boosting (XGBoost) adalah perluasan dari metode *Gradient Boosting*. Metode *Extreme Gradient Boosting* merupakan algoritma peningkatan gradien (*Gradient Boosting*) yang diusulkan oleh Dr. Tianqi Chen dari University of Washington pada tahun 2014. Algoritma *Gradient Boosting* merupakan pendekatan regresi dan klasifikasi yang menghasilkan model prediktif dari serangkaian pembelajaran yang lemah. Model dioptimalkan dengan mengevaluasi model menggunakan *loss function* (L (yi, pi)). Semakin kecil nilai *loss function* maka semakin baik performansi model tersebut. Pada setiap iterasi, algoritma XGBoost mencoba membangun pembelajar lemah yang dapat membuat prediksi lebih akurat daripada pembelajaran sebelumnya, sehingga model ini dapat diimplementasikan lebih cepat dan menghasilkan prediksi yang lebih akurat. Algoritma XGBoost memiliki fitur tambahan yang dapat mencegah *overfitting* dan meningkatkan efesiensi komputasi (Khumaidi et al., 2020).

Pada metode ini membutuhkan fungsi objektif yang berguna untuk menilai seberapa baik model yang didapatkan sesuai dengan data latih. Karakteristik yang terpenting dari fungsi objektif terdiri dari 2 bagian yaitu nilai pelatihan yang hilang dan nilai regularisasi seperti pada persamaan 1 berikut ini, (Herni Yulianti et al., 2022)

$$obj(\theta) = L(\theta) + \Omega(\theta) \quad (1)$$

Keterangan :

L = Fungsi pelatihan yang ilang

θ = Parameter model

Ω = Fungsi regularisasi

Secara umum fungsi pelatihan yang hilang dapat ditulis seperti persamaan 2 berikut,

$$L(\theta) = \sum_{i=0}^{n} l\,(y_{\mathrm{i}}, \hat{y}_i) \quad (2)$$

Keterangan :

y_i = nilai data sebenarnya yang dianggap benar

$\hat{y}_i$ = hasil nilai prediksi dari model

n = jumlah iterasi nilai dari model

XGBoost menerapkan teknik *ensemble learning*, di mana beberapa model sederhana digabungkan untuk membentuk model yang lebih kompleks. Algoritma XGBoost menggunakan banyak pohon keputusan yang dievaluasi secara berulang-ulang selama proses *training* untuk menciptakan model yang lebih akurat (Chen & Guestrin, 2016). Untuk lebih jelasnya, berikut ini adalah alur proses dari XGBoost.

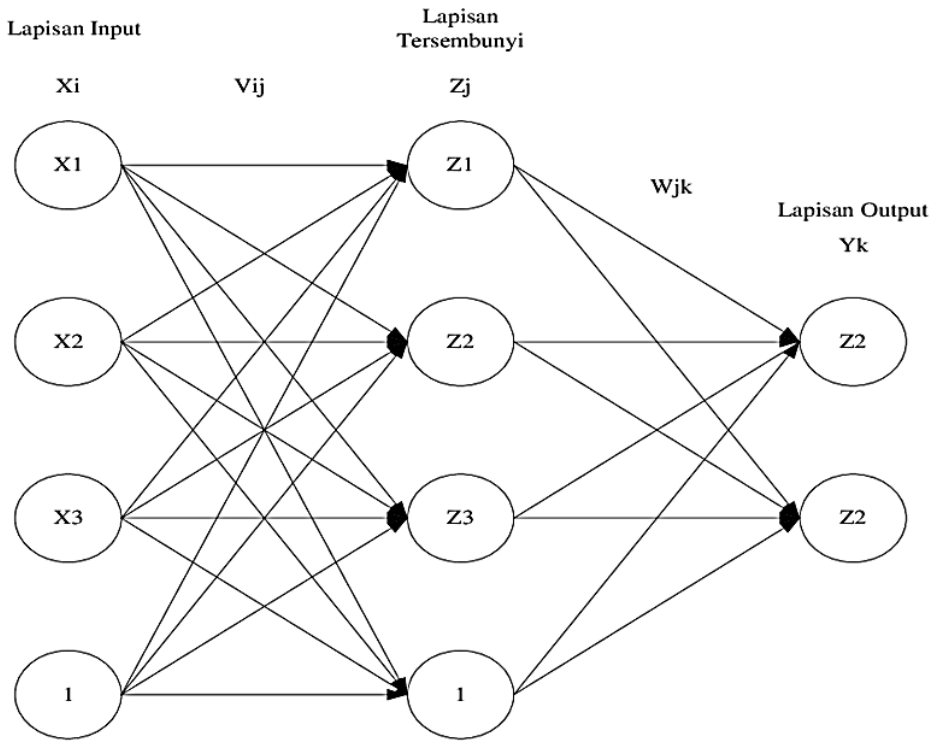
1. Memberikan bobot awal yang bernilai sama untuk setiap sampel.
2. Model membuat sebuah pohon keputusan sederhana untuk memprediksi target dari dataset.
3. Model akan menghitung kesalahan prediksi dari setiap sampel.
4. Menghitung *error gradient* dengan cara membandingkan nilai target aktual dengan hasil prediksi untuk setiap sampel.
5. XGBoost memperbarui setiap sampel menggunakan *gradient error* dari tahap sebelumnya sebagai faktor untuk menyesuaikan bobot.
6. Setelah memperbarui bobot, model membuat pohon baru dan mengulangi langkah 3 sampai 5.
7. Setelah membuat sejumlah pohon keputusan, XGBoost menggabungkan model-model tersebut untuk membuat model akhir.

8. Langkah terakhir yaitu melakukan evaluasi model pada data *testing* menggunakan metrik evaluasi.

XGBoost memiliki beberapa keunggulan, termasuk kemampuan untuk menghasilkan model yang akurat dan efisien dalam waktu yang relatif singkat. Selain itu, metode ini memiliki skalabilitas yang baik, memungkinkan penggunaannya dalam berbagai skenario, mulai dari data yang sedikit hingga data yang sangat besar, serta dapat dijalankan pada sistem terdistribusi untuk meningkatkan kecepatan training model (Chen & Guestrin, 2016).

2.4. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan atau yang lebih dikenal dengan istilah *Artificial Neural Network* (ANN) merupakan salah satu bagian dari *machine learning*. Jaringan Syaraf Tiruan ini terinspirasi oleh cara kerja sistem saraf biologis, mirip dengan proses informasi pada otak manusia. Komponen penting dari metode ini adalah struktur sistem pengolahan informasi yang terdiri dari banyak elemen pemrosesan yang terhubung satu sama lain (*neuron*). *Neuron-neuron* ini bekerja serentak untuk menyelesaikan masalah tertentu (Simamora et al., 2019). Arsitektur *Artificial Neural Network* (ANN) dapat dilihat pada Gambar 2.1.

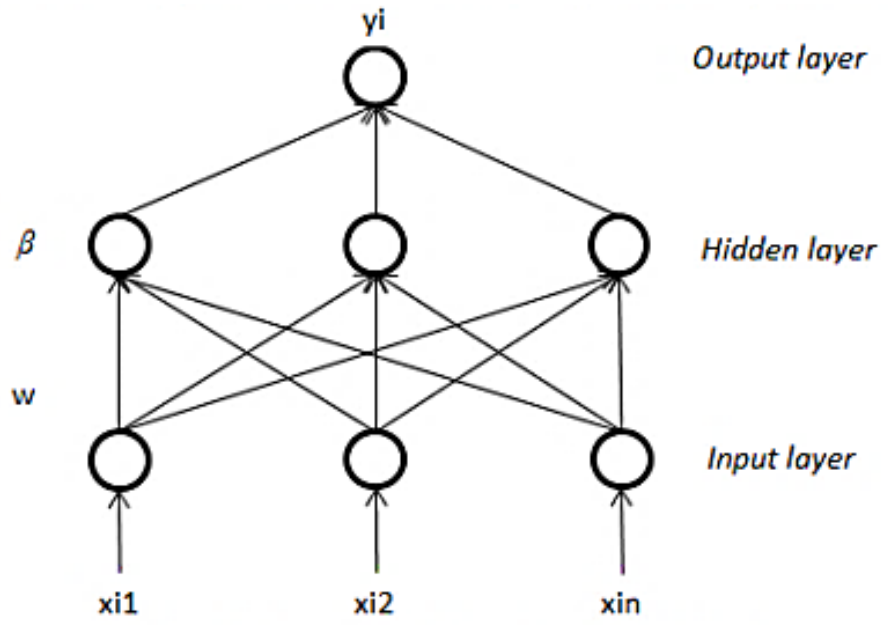


Gambar 2.1 Arsitektur Jaringan *Artificial Neural Network*

2.5. *Extreme Learning Machine* (ELM)

Extreme Learning Machine (ELM) adalah sekumpulan metode pembelajaran terawasi yang pertama kali diperkenalkan oleh Huang pada tahun 2004. *Extreme Learning Machine* merupakan pengembangan dari jaringan syaraf tiruan *feedforward* sederhana yang menggunakan satu *hidden layer* atau biasanya dikenal dengan *Single Hidden Layer Feedforward Neural Networks* (SLFNs). Dalam ELM, *layer Feedforward* parameter yang digunakan ditentukan secara manual seperti *input weight* dan *bias*. Nilai bobot *input* dan *bias* diambil secara acak dalam suatu rentang tertentu untuk menghindari hasil prediksi yang tidak stabil. ELM menggunakan dua proses, yaitu proses pelatihan (*training*) dan proses pengujian (*testing*) untuk mendapatkan hasil prediksi atau prediksi yang diinginkan (Dewi et al., 2018).

Dalam penerapan JST nilai x_1, x_2, hingga x_n, berfungsi sebagai *neuron* yang memiliki informasi. Untuk menghubungkan antar *neuron*, diperlukan *input weight* (*w*). Proses pelatihan dalam JST bertujuan untuk melatih jaringan pada setiap *input neuron*, sehingga saat prediksi dilakukan, jaringan mampu memberikan jawaban yang akurat. Secara umum model JST pada metode ELM sebagai metode pembelajaran dapat dilihat pada Gambar 2.2 berikut.



Gambar 2.2 Arsitektur Jaringan *Extreme Learning Machine*

Langkah-langkah perhitungan metode ELM dibagi menjadi dua proses yaitu proses pelatihan (*training*) dan proses pengujian(*testing*).

a. Proses *Training*

Proses *training* dilakukan untuk memperoleh *output weight* yang optimal atau memiliki tingkat kesalahan yang rendah. Tujuan dari proses *training* adalah untuk mengembangkan model ELM. Tahapan pada proses *training* adalah sebagai berikut:

1. Inisialisasi seluruh bobot (*input weight*) yang diatur dengan bilangan random kecil dari 0 hingga 1.
2. Keluaran di *hidden layer* dihitung dengan mengalikan masukan berupa data sejumlah N yang telah dinormalisasi dalam bentuk matriks (X) dengan matriks *input weight* $j \times k$ yang diperoleh secara *random*, dimana j adalah jumlah *hidden neuron* dan k adalah jumlah *input node*. Perhitungan keluaran di *hidden layer* dapat dilihat pada Persamaan 3.

$$Hinit = X.\, w^{T} \tag{3}$$

Keterangan:

$Hinit$: Matriks keluaran *hidden layer* x

X : Matriks normalisasi data tanpa target

w^{T} : Matriks *transpose input weight*

3. Setelah memperoleh keluaran *hidden layer* dengan fungsi aktivasi $((x)$, hitung H^{+} yang merupakan *Moore Penrose Generalized Invers* yaitu untuk menghitung nilai *invers* dari matriks dengan fungsi aktivasi. Berikut ini Persamaan 4 dalam menghitung nilai *output weight*.

$$\beta = H^{+}\, T \tag{4}$$

Keterangan :

β : Matriks *output weight*

H^{+} : Matriks *Moore Penrose Generalized Invers* dari matriks H

T : Matriks target

b. Proses *Testing*

Setelah proses *training*, diperoleh *output weight* yang optimal dari JST dengan metode ELM. Tujuan dari proses *testing* adalah mengevaluasi potensi metode ELM dalam hal prediksi. Tahapan pada proses *testing* adalah sebagai berikut:

1. Inisialisas*i input weight* yang telah diperoleh dari proses *training*
2. Semua keluaran di *hidden layer* dihitung dengan fungsi aktivasi menggunakan Persamaan 3.
3. Hasil prediksi yang dihitung menggunakan Persamaan 5.

$$y = H(x).\, \beta \tag{5}$$

Keterangan:

y : *Output layer* yang merupakan hasil prediksi

β : Matriks *output weight*

(x) : Keluaran di *hidden layer* dihitung dengan fungsi aktivasi

4. Hitung nilai *error* pada *output layer* untuk mengevaluasi performa model.

2.6. Pra-Pemrosesan Dataset

Pra-pemrosesan dataset atau *Pre-processing* adalah tahap awal dalam pengolahan data yang melibatkan serangkaian teknik untuk membersihkan dan mempersiapkan data mentah sebelum diproses lebih lanjut, seperti penghapusan data duplikat, penghapusan nilai yang hilang, dan transformasi data (Ahmed dan Kamal, 2020).

2.6.1. Analisis Data Eksplorasi

Analisis data eksplorasi proses investigasi awal yang bertujuan untuk mengidentifikasi pola, menemukan anomali, menguji hipotesis, dan memeriksa asumsi. Tujuan utama adalah untuk merumuskan hipotesis, mengidentifikasi pola-pola menarik, dan mendapatkan pemahaman yang lebih baik tentang struktur data tanpa membuat asumsi yang mendalam.

2.6.2. Normalisasi Dataset

Normalisasi data merupakan metode yang bertujuan untuk menyederhanakan proses dalam jaringan. Normalisasi data dilakukan untuk menstandarisasikan data agar *output* pelatihan sesuai dengan fungsi aktivasi. Berikut rumus untuk proses normalisasi dataset dapat dilihat pada Persamaan 6 :

$$x_i norm = \frac{xi - \min(x)}{\max(x) - \min(x)} \tag{6}$$

Keterangan :

$xinorm$: Hasil normalisasi data

xi : Data x pada indeks ke-i

$min(x)$: Data *minimum*

$max(x)$: Data *maximum*

2.7. Denormalisasi Dataset

Denormalisasi adalah proses mengembalikan nilai hasil normalisasi ke nilai aslinya dengan tujuan mendapat nilai yang sebenarnya. Dalam proses denormalisasi, data yang dinormalisasi yang diskalakan dari 0 ke 1 dikembalikan ke skala nilai aslinya dengan menggunakan Persamaan 7 berikut.

$$X = Xnorm * (Xmax - Xmin) + Xmin \tag{7}$$

Keterangan :

X : Denormalisasi dataset

X*norm* : Data normalisasi

X*max* : Data maksimum

X*min* : Data minimum

2.8. Evaluasi

Evaluasi dilakukan setelah algoritma XGBoost dan ELM diimplementasikan dan mendapatkan hasil prediksi. Evaluasi dilakukan untuk memperoleh pemahaman yang lebih baik tentang seberapa baik model XGBoost dan ELM dalam memprediksi angka dan tingkatan. Tujuan utama dari tahap evaluasi ini yaitu untuk menilai kinerja model yang telah dibangun dan mengukur seberapa baik model tersebut dapat memprediksi nilai target yang sebenarnya.

2.8.1. *Mean Square Error* (MSE)

Mean Square Error adalah rata-rata kuadrat selisih antara nilai yang diprediksi dan nilai yang sebenarnya. MSE dihitung dengan menjumlahkan kuadrat dari selisih antara setiap nilai prediksi dan nilai sebenarnya, kemudian membaginya dengan jumlah total data Semakin kecil nilai MSE, semakin baik model dalam memprediksi nilai yang sebenarnya.

MSE memiliki keuntungan bahwa akan memberikan bobot yang lebih besar pada *error* yang besar, sehingga dapat menyoroti perbedaan yang signifikan antara prediksi dan nilai yang sebenarnya. Kekurangannya terletak pada kurangnya keintuisian nilai karena outputnya dinyatakan dalam kuadrat satuan aslinya. Berikut rumus MSE dapat dilihat pada persamaan 8.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{8}$$

Keterangan :

MSE = *Mean Square Error*

y_i = target prediksi yang asli

$\hat{y}_i$ = nilai yang dihasilkan dari perhitungan pengujian

n = jumlah data latih

2.8.2. *Root Mean Square Error* (RMSE)

RMSE adalah akar kuadrat dari MSE. RMSE memberikan ukuran kesalahan rata-rata antara nilai sebenarnya dalam satuan aslinya dan prediksi. Keuntungan dari RMSE yaitu dapat memberikan ukuran kesalahan dalam satuan yang sama dengan variabel yang diukur. Kekurangannya yaitu nilainya lebih sulit untuk diinterpretasi daripada MSE atau MAE. Berikut rumus RMSE dapat dilihat pada persamaan 9.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \quad (9)$$

Keterangan :

RMSE = *Root Mean Square Error*

y_i = target prediksi yang asli

$\hat{y}_i$ = nilai yang dihasilkan dari perhitungan pengujian

n = jumlah data latih

2.8.3. *Mean Absolute Error* (MAE)

Mean Absolute Error adalah rata-rata dari selisih absolut antara nilai prediksi dan nilai sebenarnya. MAE memberikan ukuran kesalahan rata-rata tanpa memperhatikan arah kesalahan. Keuntungan dari MAE yaitu lebih mudah diinterpretasi karena nilainya berada dalam satuan yang sama dengan variabel yang diukur. Kekurangannya yaitu tidak sensitif terhadap kesalahan besar. Berikut rumus MAE dapat dilihat pada persamaan 10.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |yi - \hat{y}i| \quad (10)$$

Keterangan :

MAE = *Mean Absolute Error*

y_i = target prediksi yang asli

$\hat{y}_i$ = nilai yang dihasilkan dari perhitungan pengujian

n = jumlah data latih

2.9. Penelitian Relevan

Berikut beberapa penelitian yang relevan yang berkaitan dengan Perbandingan Algoritma *Extreme Gradient Boosting* (XGBoost) dan Algoritma *Extreme Learning Machine* (ELM) Dalam Memprediksi Tingkat Curah Hujan di Kota Medan.

1. Berdasarkan penelitian (Freecenta et al., 2022) yang membahas tentang prediksi curah hujan di Kab. Malang menggunakan LSTM, data yang digunakan yaitu data dari Januari 2010 s/d 25 Desember 2021. Pada penelitian ini dilakukan dengan menggunakan dua skenario uji coba. Pada skenario uji coba pertama dengan 4 *layer* LSTM dimana masing-masing *layer* terdapat 100

neuron dengan hasil akurasi sebesar 54,31%. Sedangkan skenario uji coba kedua dilakukan dengan 3 *layer* LSTM dimana masing-masing *layer* terdapat 50 *neuron* dengan hasil akurasi sebesar 58,54%. Hasil terbaik terdapat pada skenario 2 dengan akurasi model sebesar 58.54% dengan nilai *Mean Absolute Error* sebesar 7.90, *Root Mean Squared Error* sebesar 10.16, dan nilai *Mean Squared Error* sebesar 103.37.

2. Pada penelitian yang dilakukan oleh (Anwar et al., 2021) berjudul "*Rainfall prediction using Extreme Gradient Boosting*". Dataset pelatihan terdiri dari data cuaca harian tahun 2013 hingga 2019, sedangkan data pengujian berupa data cuaca harian tahun 2020. XGBoost mampu memberi peringkat pada atribut penting yang berkontribusi pada model. Saat diuji terhadap data cuaca pada tahun 2020, model tersebut memberikan MAE sebesar 8,8. Dari penelitian ini didapatkan hasil bahwa model mampu menghasilkan prediksi akurat dengan RMSE pelatihan sebesar 2.7 mm dan MAE pengujian sebesar 8.8 mm.
3. Pada penelitian yang dilakukan oleh (Simamora et al., 2019) dengan judul "Peramalan Curah Hujan Menggunakan Metode *Extreme Learning Machine*". Data yang digunakan adalah data curah hujan dalam berbentuk *time series* dan dalam dasarian (10 hari). Nilai dari rata-rata MAPE terkecil terlihat pada persentase sebesar 80% dengan nilai sebesar 6,9478% sedangkan nilai rata-rata MAPE terbesar pada persentase 50% dengan nilai 35,4518%. Rata-rata MAPE terkecil terletak pada jumlah neuron sebanyak 2 dengan nilai sebesar 6,0331% dan nilai rata-rata MAPE terbesar adalah sebanyak 10 neuron dengan nilai sebesar 29,7614%. Jumlah fitur sebanyak 4 menghasilkan nilai dari rata-rata MAPE yang terkecil dengan nilai 6,7568%. Nilai dengan rata-rata MAPE terbesar ada pada jumlah fitur sebanyak 2 dengan nilai 41,7843%. Berdasarkan pengujian tersebut dapat di perhatikan bahwa nilai rata-rata MAPE pada fitur sebanyak 2 memiliki nilai paling besar. Dari penelitian tersebut didapatkan hasil pengujian rentang waktu tahun 2002 sampai 2015 diperoleh dengan nilai MAPE terkecil sebesar 3,6852%, dengan banyak fiturnya 4, banyak *neuron* pada *hidden layer* sebanyak 2, dan *persentase* data *training* didapatkan sebanyak 80%.

4. Pada penelitian yang dilakukan oleh (Dewi et al., 2018) tentang "Prediksi Jumlah Kriminalitas menggunakan Metode *Extreme Learning Machine"*. Penelitian ini menggunakan data kriminalitas Polres Kabupaten Probolinggo pada tahun 2012 hingga 2017. Peneliti ini menemukan arsitektur optimal dengan fitur sebanyak 7 fitur, perbandingan rasio data yaitu 80%:20%, dan jumlah *neuron* pada *hidden layer* sebanyak 7 *neuron*, dan fungsi aktivasi *sigmoid biner*. Tingkat kesalahan dihitung menggunakan *Mean Square Error* (MSE) sebesar 0,037662.
5. Pada penelitian yang dilakukan (Pandika Pinata et al., 2020) dengan judul "Prediksi Kecelakaan Lalu Lintas menggunakan XGBoost". Diperoleh nilai *error* dari 4 kategori dengan masing-masing RMSE, RMSE 21,69 untuk model peramalan jumlah kejadian, RMSE 4,92 untuk model peramalan jumlah orang meninggal dunia, RMSE 4,11 untuk model peramalan jumlah orang yang mengalami luka berat dan RMSE 77,24 untuk model prediksi jumlah korban luka ringan.

BAB III

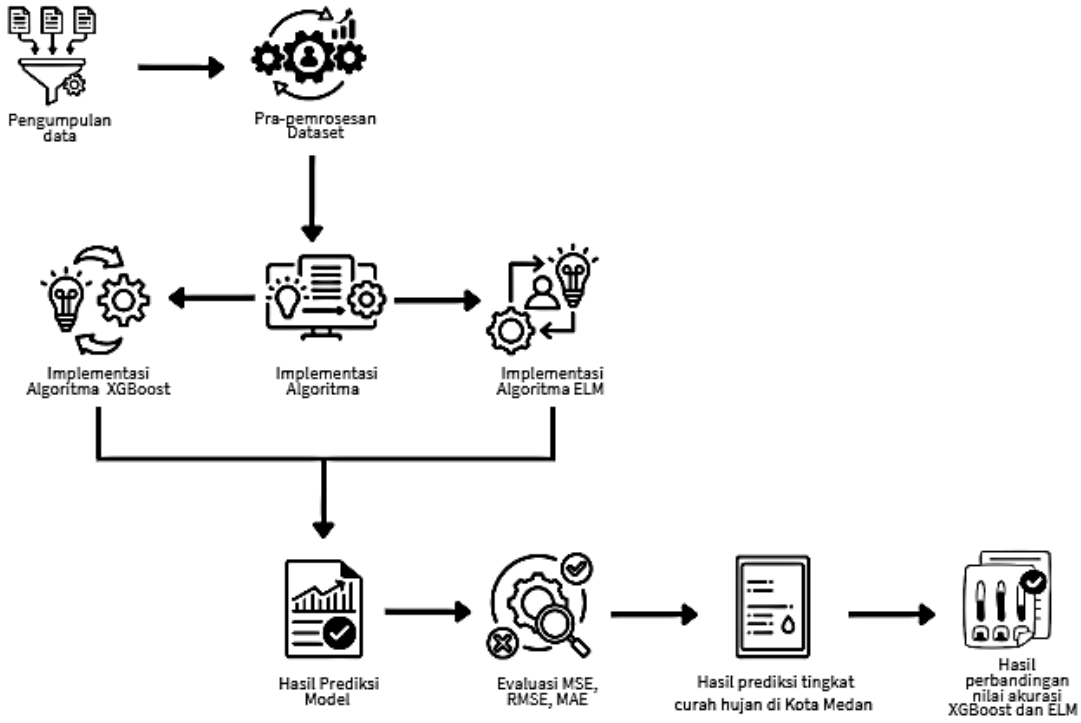
ANALISIS DAN PERANCANGAN SISTEM

3.1. Gambaran Umum Penelitian

Penelitian ini bertujuan untuk eksplorasi dan menyelidiki lebih dalam tentang prediksi curah hujan menggunakan algoritma XGBoost dan ELM, dengan fokus pada pemahaman yang menyeluruh serta analisis yang mendalam terhadap fenomena yang terjadi. Proses penelitian dapat dilihat pada Gambar 3.1 yang menunjukkan langkah-langkah yang akan dilakukan pada penelitian ini.

Tahapan penelitian dilakukan mulai dari pengumpulan data. Data yang dikumpulkan, yakni data yang diperoleh dari data harian cuaca yang diambil dari *website* BMKG (Badan Meteorologi, Klimatologi dan Geofisika) pada Stasiun Balai Besar Meteorologi Klimatologi dan Geofisika Wilayah I. Data kemudian disatukan dan disimpan dalam format .csv yang akan digunakan menjadi data *input* untuk melakukan prediksi. Tahap selanjutnya adalah tahap pra-pemrosesan dataset. Pra-pemrosesan dilakukan dengan beberapa tahap yaitu, Eksplorasi Data Analisis (EDA), normalisasi dataset, pembagian dataset. Tahap Eksplorasi Data Analisis (EDA) untuk memahami karakteristik data curah hujan yang tersedia secara mendalam sebelum melakukan analisis lebih lanjut. Setelah itu, dilakukan penanganan terhadap nilai-nilai yang hilang (*missing values*) dengan mengidentifikasi dan memutuskan cara untuk mengatasi nilai-nilai yang kosong tersebut. Jika ada nilai yang hilang, dilakukan imputasi untuk menggantikan nilai-nilai tersebut dengan nilai median yang sesuai. Tahap pra-pemrosesan dataset selanjutnya adalah normalisasi dataset yang dilakukan untuk memastikan setiap fitur memiliki skala yang serupa. Setelah itu dataset akan dibagi menjadi dua bagian yaitu, data *training* dan data *testing*. Setelah melakukan pra-pemrosesan dataset, tahap selanjutnya adalah melakukan implementasi algoritma masing-masing untuk mendapatkan nilai hasil prediksi.

Algoritma yang dimaksud yaitu Algoritma *Extreme Gradient Boosting* (XGBoost) dan Algoritma *Extreme Learning Machine* (ELM). Setelah mengimplementasikan algoritma, maka keluar hasil prediksi dari model dataset pelatihan dan nilai akurasi dari masing-masing algoritma. Selanjutnya tahap evaluasi yang dilakukan menggunakan perhitungan *Mean Square Error* (MSE), *Mean Absolute Error* (MAE), dan *Root Mean Square Error* (RMSE). Setelah mendapatkan nilai akurasi dari masing-masing algoritma, selanjutnya adalah memprediksi tingkat curah hujan. Hasil prediksi yang telah didapat akan melakukan tahap denormalisasi data. Denormalisasi dilakukan untuk mengembalikkan nilai kembali ke awal. Tahapan penelitian terakhir adalah membandingkan hasil akhir yang berguna untuk mengetahui algoritma mana yang lebih baik dari kedua algoritma yang digunakan dalam memprediksi tingkat curah hujan di Kota Medan.



Gambar 3.1 Gambaran Umum Penelitian

3.2. Pengumpulan Data

Data yang digunakan pada pengkajian ini merupakan data harian cuaca di Kota Medan. Adapun data diambil dari *website* BMKG (Badan Meteorologi, Klimatologi dan Geofisika) pada Stasiun Balai Besar Meteorologi Klimatologi dan Geofisika

Wilayah I dengan data harian di mulai dari tanggal tanggal 1 Januari 2022 – 31 Desember 2023 di Kota Medan. Data yang diambil yaitu, temperatur minimum (°C), temperatur maksimum (°C), temperatur rata-rata (°C), kelembapan rata-rata (%), curah hujan (mm), lamanya penyinaran matahari (jam), kecepatan angin maksimum (m/s), arah angin saat kecepatan maksimum (°), kecepatan angin rata-rata (m/s). Contoh data untuk temperatur minimum, temperatur maksimum, temperatur rata-rata, kelembapan rata-rata, curah hujan, lamanya penyinaran matahari, kecepatan angin maksimum, arah angin saat kecepatan maksimum, kecepatan angin rata-rata dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Contoh Dataset BMKG Stasiun Balai Besar Meteorologi Klimatologi dan Geofisika Wilayah I

Tanggal	Temperatur minimum (°C),	Temperatur maksimum (°C),	Temperatur rata-rata (°C)	Kelembapan rata-rata (%)	Curah Hujan (mm)	Lamanya penyinaran matahari (jam)	Kecepatan angin maksimum (m/s)	Arah angin saat kecepatan maksimum (°)	Kecepatan angin rata-rata (m/s)
01/01/2022	24	28.8	24.6	96	12.9	0.2	7	320	2
02/01/2022	23.5	26.6	24.8	94	26.5	1.5	4	300	2
03/01/2022	23.8	31.3	26.6	85	6.5	0	2	210	1
04/01/2022	23.8	31.2	26.8	88	3.2	3.1	3	290	1
05/01/2022	22.1	32	26.1	84	21.1	1.6	3	320	2

3.3. Pra-Pemrosesan Dataset

Tahap pra-pemrosesan dataset pada penelitian ini menggunakan Analisis data eksplorasi, normalisasi data, dan pembagian dataset.

3.3.1. Analisis Data Eksplorasi

Analisis Data Eksplorasi dilakukan agar dapat mengetahui karakteristik dari dataset dan melihat *outlier* pada dataset. Pada eksplorasi data ini dilakukan *input* dataset terlebih dahulu dengan menggunakan format csv (*comma separated values*).

3.3.2. Normalisasi Dataset

Setelah terbentuk pola data selanjutnya akan dilakukan normalisasi data. Normalisasi data bertujuan untuk menyederhanakan proses dalam jaringan. Contoh perhitungan

dataset menggunakan rumus persamaan 6 yang akan dinormalisasi pada dataset yang tertera pada Tabel 3.1 pada tanggal 1 Januari 2022:

Diketahui :

$xinorm$: Hasil normalisasi

xi : 12.9

$min(x)$: 0

$max(x)$: 128

$x_i norm = \frac{12.9-0}{128-0}$

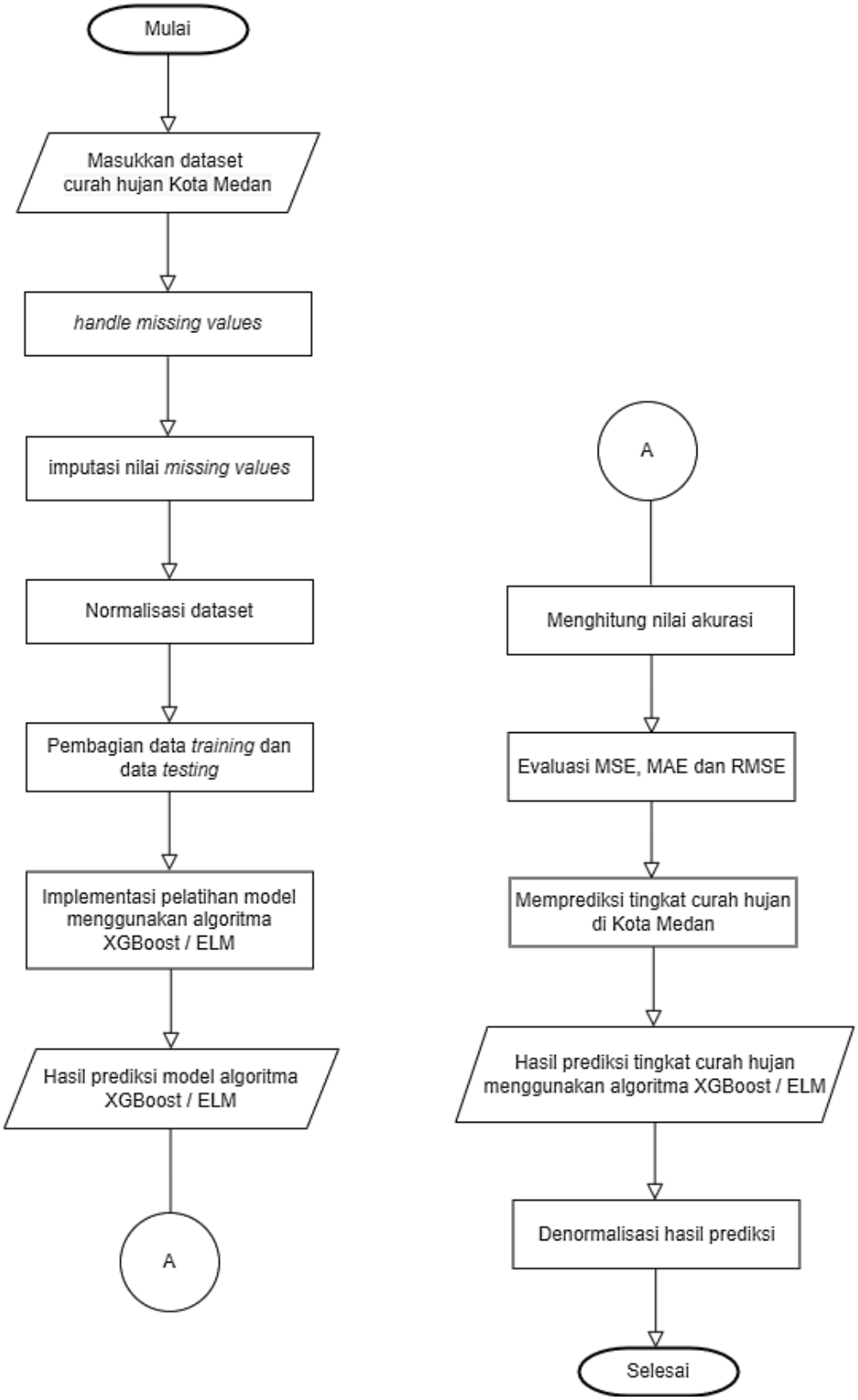
$x_i norm = 0.100781$

3.3.3. Pembagian Dataset

Data yang telah dinormalisasi kemudian dibagi menjadi dua bagian yaitu data pelatihan (*training)* dan data pengujian (*testing)*. Dalam penelitian terdapat total 730 data. Dataset ini dibagi menjadi 80% dataset *training* dan 20% dataset *testing*. Pembagian ini dilakukan dengan menggunakan data *training* untuk mengatur nilai bobot dan bias, sementara data *testing* digunakan untuk menguji keakuratan model yang telah diperoleh dari data *training*.

3.4. Proses Prediksi Tingkat Curah Hujan

Dapat dilihat pada Gambar 3.3 yang merupakan *flowchart* atau alur penelitian pada prediksi curah hujan.



Gambar 3.2 *Flowchart* penelitian prediksi tingkat curah hujan

Penjelasan proses prediksi pada penelitian ini adalah sebagai berikut :

a. Masukkan dataset yang telah dikumpulkan dalam bentuk file .csv kedalam sistem.
b. Melakukkan *handle missing values*.
c. Melakukan imputasi nilai *missing values.*
d. Melakukan normalisasi dataset.
e. Setelah menormalisasi dataset, selanjutnya adalah pembagian dataset. Pembagian dataset dibagi menjadi dua bagian, yaitu dataset *training* dan dataset *testing*.
f. Menerapkan implementasi dan pelatihan model menggunakan algoritma *Extreme Gradient Boosting* atau algoritma *Extreme Learning Machine*.
g. Mendapatkan hasil prediksi dari model yang telah dilatih menggunakan algoritma *Extreme Gradient Boosting* atau algoritma *Extreme Learning Machine*.
h. Menghitung nilai akurasi dari model algoritma *Extreme Gradient Boosting* atau algoritma *Extreme Learning Machine*.
i. Setelah mendapatkan hasil nilai akurasi, selanjutnya melakukan perhitungan nilai *Mean Square Error* (MSE), *Mean Absolute Error* (MAE), dan *Root Mean Square Error* (RMSE).
j. Melakukan prediksi tingkat curah hujan di Kota Medan menggunakan algoritma *Extreme Gradient Boosting* atau algoritma *Extreme Learning Machine*.
k. Melakukan denormalisasi pada hasil prediksi yang sudah didapatkan. Denormalisasi bertujuan untuk mengembalikan nilai ke awal.
l. Mendapatkan hasil prediksi tingkat curah hujan di Kota Medan menggunakan algoritma *Extreme Gradient Boosting* atau algoritma *Extreme Learning Machine*.

Alur penelitian dari algoritma *Extreme Gradient Boosting* dan algoritma *Extreme Learning Machine* memiliki tahapan yang sama.

3.5. Denormalisasi Dataset

Denormalisasi dilakukan pada tahap terakhir setelah mendapatkan data hasil prediksi. Dalam proses denormalisasi, data yang dinormalisasi yang diskalakan dari 0 ke 1 dikembalikan ke skala nilai aslinya. Berikut merupakan contoh perhitungan denormalisasi menggunakan rumus persamaan 7 :

Diketahui :

X : Denormalisasi dataset

X*norm* : 0.100781

X*max* : 128

X*min* : 0

X = 0.100781 * (128 - 0) + 0

X = 12.8

3.6. Evaluasi

Evaluasi dilakukan setelah algoritma XGBoost dan ELM diimplementasikan dan mendapatkan hasil prediksi. Evaluasi dilakukan untuk memperoleh pemahaman yang lebih baik tentang seberapa baik model XGBoost dan ELM dalam memprediksi. Dalam hal ini untuk mendapatkan prediksi dari model, penulis menggunakan evaluasi *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE), dan *Mean Absolute Error* (MAE).

3.6.1. *Mean Square Error* (MSE)

Mean Square Error (MSE) adalah metrik evaluasi yang mengukur rata-rata dari kuadrat kesalahan prediksi. MSE memberikan gambaran seberapa besar kesalahan prediksi rata-rata dalam bentuk kuadrat. Nilai MSE yang lebih kecil menunjukkan bahwa model memiliki kesalahan prediksi yang lebih rendah.

3.6.2. *Root Mean Square Error* (RMSE)

Root Mean Squared Error (RMSE) adalah akar kuadrat dari MSE. RMSE memberikan ukuran kesalahan prediksi dalam satuan yang sama dengan data asli. RMSE memberikan ukuran kesalahan prediksi yang lebih dapat diinterpretasikan karena berada dalam satuan yang sama dengan nilai asli data. Seperti MSE, nilai RMSE yang lebih kecil menunjukkan bahwa model memiliki kinerja yang lebih baik.

3.6.3. *Mean Absolute Error* (MAE)

Mean Absolute Error (MAE) adalah metrik evaluasi yang mengukur rata-rata dari nilai absolut kesalahan prediksi. MAE memberikan gambaran seberapa besar rata-rata kesalahan prediksi tanpa memperhitungkan arah kesalahan. Nilai MAE yang lebih kecil menunjukkan kinerja model yang lebih baik.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi Sistem

Untuk dapat mengimplementasi dan melakukan pengujian terhadap prediksi tingkat curah hujan di kota medan, perangkat keras dan lunak yang digunakan memiliki spesifikasi sebagai berikut:

1. *Processor* Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz (4CPUs), ~1.2GHz.
2. Memori RAM 4GB.
3. SSD dengan kapasitas 256 GB.
4. Sistem Operasi Windows 11 Home Single Language 64-bit.
5. Browser Google Chrome.
6. *Google Colaboratory*.
7. *Python*.

4.2. Pra-Pemrosesan Dataset

Pra-pemrosesan dataset atau *pre-processing* adalah bagian penting dari pemahaman data yang berpotensi mempengaruhi keakuratan data yang akan digunakan. Proses dari pra-pemrosesan dataset ini terdiri dari beberapa tahap, dimulai dari Eksplorasi Data Analisis (EDA), pembersihan data, normalisasi dataset sebelum masuk ke tahap pemodelan, serta pengelompokkan data menjadi data *training* dan data *testing*.

4.2.1. Analisis Data Eksplorasi

Eksplorasi data adalah langkah awal penyelidikan untuk memahami pola, karakteristik, dan informasi yang terkandung dalam data. Eksplorasi data ini tahapan penting dalam analisis data karena membantu dalam memberikan informasi yang relevan. Selama eksplorasi data, langkah pertama adalah memasukkan dataset dalam format csv (*comma separated values*).

Deskripsi Statistik Data untuk dapat memahami eksplorasi data dapat dilihat pada Gambar 4.1.

```
# Statistik deskriptif
print("\nStatistik Deskriptif:")
print(data.describe())
```

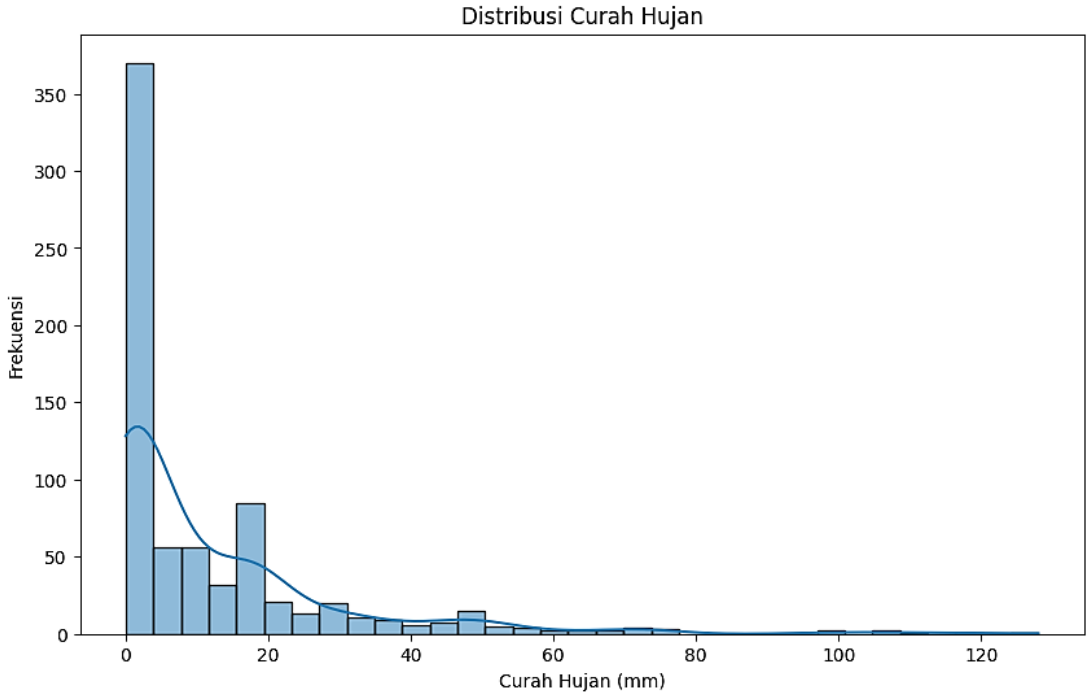
Gambar 4.1 Proses Deskripsi Statistik

Dapat dilihat pada Tabel 4.1 berikut yang merupakan contoh hasil dari proses deskripsi data yang telah dilakukan dari Gambar 4.1.

Tabel 4.1 Deskripsi Statistik

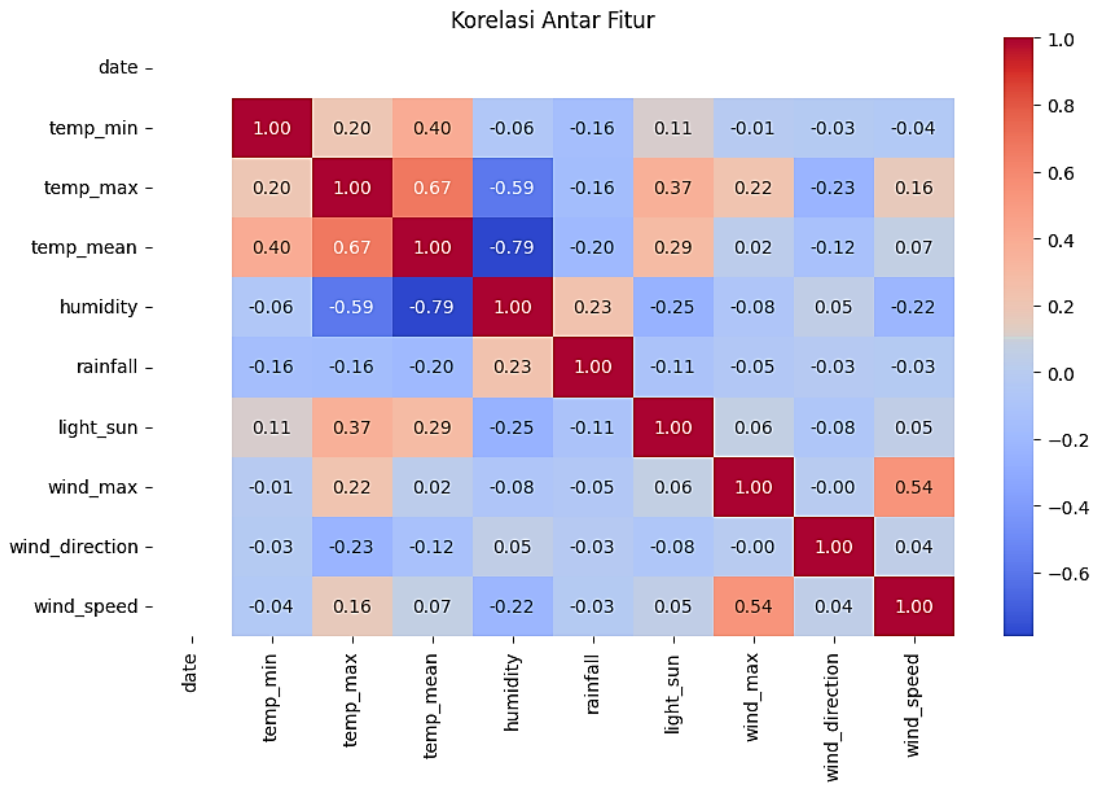
	Temperatur minimum (°C)	Temperatur maksimum (°C)	Temperatur rata-rata (°C)	Kelembapan rata-rata (%)	Curah Hujan (mm)	Lamanya penyinaran matahari (jam)	Kecepatan angin maksimum (m/s)	Arah angin saat kecepatan maksimum (°)	Kecepatan angin rata-rata (m/s)
count	730	730	730	730	730	730	730	730	730
mean	23.82	32.82	27.64	84.04	11.76	3.81	4.26	168.84	1.59
min	17.60	25.20	23.50	67.00	0.00	0.00	1.00	10.00	0.00
25%	23.40	31.80	26.90	81.00	0.20	1.70	3.00	50.00	1.00
50%	23.40	33.00	27.60	84.00	3.60	3.70	4.00	165.00	2.00
75%	24.40	34.20	28.30	87.00	18.00	5.52	5.00	280.00	2.00
max	26.40	37.00	32.20	96.00	128.00	11.00	16.00	360.00	5.00
std	1.05	1.81	1.20	4.44	17.83	2.52	1.67	114.86	0.65

Pada Gambar 4.2 menunjukkan hasil dari memvisualisasikan distribusi curah hujan dari kumpulan data yang tersedia. Dengan visualisasi ini, dapat dengan mudah melihat pola distribusi tingkat curah hujan dalam data, serta memperoleh pemahaman yang lebih baik tentang frekuensi kemunculan nilai-nilai curah hujan yang berbeda.



Gambar 4.2 Distribusi Curah Hujan

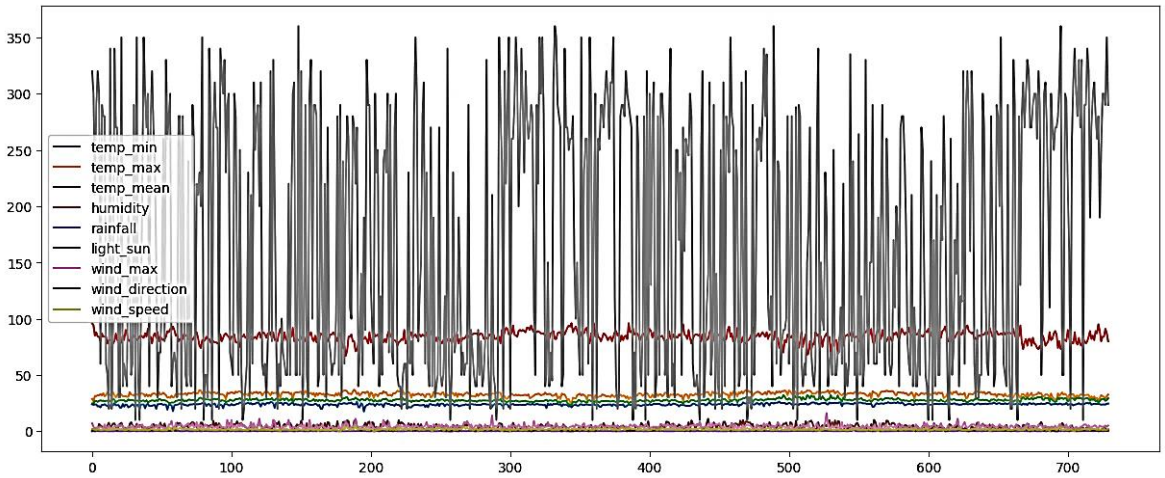
Pada Gambar 4.3 dapat dilihat hasil dari memvisualisasikan matriks korelasi antar fitur dari kumpulan data yang diberikan. Tujuan dari korelasi adalah untuk memberikan pemahaman tentang seberapa kuat hubungan antara setiap pasangan fitur dalam kumpulan data yang diberikan. Korelasi menggambarkan arah dan kekuatan hubungan antara variabel-variabel.



Gambar 4.3 Korelasi Antar Fitur

Korelasi positif menunjukkan bahwa jika nilai satu fitur meningkat, maka nilai fitur lainnya cenderung meningkat juga, sedangkan korelasi negatif menunjukkan bahwa jika nilai satu fitur meningkat, maka nilai fitur lainnya cenderung menurun. Jika korelasi nol menunjukkan bahwa tidak ada hubungan linier antara kedua fitur tersebut. Dengan memvisualisasikan matriks korelasi pembaca dapat dengan mudah melihat pola korelasi antar fitur, yang dapat membantu dalam pemahaman dan analisis data, serta dalam proses pemilihan fitur untuk pemodelan. Pada Gambar 4.3 dapat dilihat bahwa tidak adanya korelasi nol, maka dapat disimpulkan bahwa antara fitur satu dengan fitur lainnya saling berkaitan.

Pada Gambar 4.4 dapat dijelaskan bahwa garis plot menggambarkan beberapa fitur khusus dalam kumpulan data, di mana setiap fitur direpresentasikan oleh garis yang berbeda dalam plot. Hal ini berguna untuk visualisasi dan pemahaman lebih lanjut tentang tren atau pola dalam data. Visualisasi semua fitur yang dimasukkan sebagai *input* dan *output* agar dapat melihat rentang nilai dari masing-masing fitur.



Gambar 4.4 Visualisasi plot

4.2.2. *Handle Missing Values*

Pada tahap *handle missing values* ini melibatkan identifikasi dan penanganan nilai-nilai yang hilang dalam dataset. Dapat dilakukan dengan menghapus baris atau kolom yang mengandung nilai yang hilang, atau dengan menggunakan teknik imputasi untuk mengganti nilai yang hilang dengan estimasi yang sesuai. Identifikasi apakah ada nilai yang hilang dalam dataset, karena *missing values* dapat mempengaruhi kualitas prediksi. Proses dari *missing values* dapat dilihat pada Gambar 4.5 berikut.

```
# create a table with data missing
missing_values=data.isnull().sum() # missing values

percent_missing = data.isnull().sum()/data.shape[0]*100 # missing value %

value = {
    'missing_values ':missing_values,
    'percent_missing %':percent_missing ,
     'data type' : data.dtypes
}
frame=pd.DataFrame(value)
frame
```

Gambar 4.5 Proses *missing values*

Pada Tabel 4.2 hasil dari pemeriksaan data pada kolom tabel menunjukkan ada atau tidak adanya jumlah data yang hilang pada setiap variabelnya. Jika ada nilai yang hilang maka dapat dilakukan dengan nilai rata-rata dari variabel sebagai penggantian nilai.

Tabel 4.2 Hasil pemeriksaan data

Parameter	Jumlah Data Hilang
Tanggal	730
Temperatur minimum (°C)	0
Temparatur maksimum (°C)	0
Temperatur rata-rata (°C)	0
Kelembapan rata-rata (%)	0
Curah hujan (mm)	0
Lamanya penyinaran matahari (jam)	2
Kecepatan angin maksimum (m/s)	0
Arah angin saat kecepatan maksimum (°)	0
Kecepatan angin rata-rata (m/s)	0

4.2.3. Imputasi nilai *Missing Values*

Setelah melakukan tahap *handle missing values*, tahap selanjutnya adalah melakukan imputasi pada nilai *missing values* jika ada nilai yang hilang dalam dataset, pada tahap ini melibatkan untuk menggantikan nilai-nilai yang hilang dengan nilai yang sesuai. Teknik imputasi bisa beragam, seperti menggunakan nilai rata-rata, median, atau bahkan menggunakan model prediktif untuk memprediksi nilai yang hilang. Proses dari imputasi nilai *missing values* dapat dilihat pada Gambar 4.6 berikut.

```
# Imputasi nilai median pada kolom 'light_sun'
median_light_sun = data['light_sun'].median()
data['light_sun'] = data['light_sun'].fillna(median_light_sun)
```

Gambar 4.6 Imputasi nilai *missing values*

Setelah melakukan imputasi nilai median pada nilai yang hilang, dapat dilihat pada Tabel 4.3 hasil akhir dari pemeriksaan data pada kolom tabel yang menunjukkan tidak adanya jumlah data yang hilang pada setiap variabelnya.

Tabel 4.3 Hasil akhir pemeriksaan data

Parameter	Jumlah Data Hilang
Tanggal	730
Temperatur minimum (°C)	0
Temparatur maksimum (°C)	0
Temperatur rata-rata (°C)	0
Kelembapan rata-rata (%)	0
Curah hujan (mm)	0
Lamanya penyinaran matahari (jam)	0
Kecepatan angin maksimum (m/s)	0
Arah angin saat kecepatan maksimum (°)	0
Kecepatan angin rata-rata (m/s)	0

4.2.4. Normalisasi Dataset

Tahap selanjutnya adalah tahap normalisasi dataset. Normalisasi dilakukan menggunakan penyesuaian data ke dalam skala umum dilakukan agar menghasilkan hasil yang dapat dibandingkan antara nilai prediksi dan nilai aktual. Pada tahap ini menggunakan *library MinMaxScaler* berfungsi untuk menormalisasi data. Proses normalisasi ini dapat dilihat pada Gambar 4.7 dan Gambar 4.8 berikut.

```
# Inisialisasi MinMaxScaler
scaler = MinMaxScaler()

# Menentukan kolom yang akan dinormalisasi
rainfall_fix = ['rainfall']

# Melakukan normalisasi Min-Max
data[rainfall_fix] = scaler.fit_transform(data[rainfall_fix])

# Menampilkan lima baris pertama dataset setelah normalisasi
print(data.head())
```

Gambar 4.7 Proses normalisasi dataset algoritma XGBoost

```
# Normalisasi fitur menggunakan Min-Max Scaling
scaler = MinMaxScaler()
X_train_normalized = scaler.fit_transform(X_train)
X_test_normalized = scaler.transform(X_test)
```

Gambar 4.8 Proses normalisasi dataset algoritma ELM

4.2.5. Pembagian Dataset

Dataset dibagi menjadi dua bagian yaitu dataset *training* dan dataset *testing*. Dataset *training* digunakan untuk melatih model, sedangkan dataset *testing* digunakan sebagai data akhir untuk melihat keakuratan jaringan yang sudah dilatih. Dataset dibagi menjadi 80% data *training* dan 20% data *testing*. Pada Gambar 4.9 menunjukkan proses pembagian dataset. Dimana pada proses ini ditemukan hasilnya sebanyak 584 data *training* dan 146 data *testing*.

```
X = data[['temp_min', 'temp_max', 'temp_mean', 'humidity','light_sun','wind_max','wind_direction','wind_speed']]
y = data['rainfall']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 12)
```

Gambar 4.9 Proses pembagian dataset

4.3. Algoritma *Extreme Gradient Boosting*

Pada tahap pertama, dapat dilihat pada Gambar 4.10 merupakan *source code* untuk mengimport *library* yang digunakan.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import pandas as pd
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
```

Gambar 4.10 *Source code library* algoritma XGBoost

Fungsi-fungsi dari *library* di atas yaitu *numpy* berfungsi untuk memproses sebuah data berbentuk *array*. Pandas untuk mengolah *dataframe* seperti menambah atau menghapus data. XGBRegressor digunakan untuk pemodelan prediksi dan pembelajaran mesin. XGBRegressor ini merupakan implementasi dari algoritma

gradient boosting yang sangat efisien. *Sklearn* untuk *machine learning*-nya. Modul *model_selection* dari *sklearn* digunakan untuk pemisahan data menjadi set pelatihan dan pengujian. *Sklearn.metrics* digunakan untuk menyediakan berbagai metrik evaluasi untuk model pembelajaran mesin. *Matplotlib* berfungsi untuk memvisualisasikan data dalam bentuk grafik. *Seaborn* digunakan untuk memvisualisasikan data dalam analisis eksploratori dan pembuatan laporan. *MinMaxScaler* berfungsi untuk menormalisasi data.

4.3.1. Implementasi Algoritma *Extreme Gradient Boosting* (XGBoost)

Setelah dilakukan proses pra-pemrosesan, selanjutnya akan diimplementasikan algoritma XGBoost terhadap dataset. Implementasi dilakukan dengan menggunakan library *sklearn* untuk XGBoost serta menggunakan model XGB atau XGB Regressor. Dengan terbentuknya *training* model maka dapat dilakukan pengujian dengan menggunakan data uji berdasarkan *training* model yang telah dibangun. Proses ini dapat dilihat pada Gambar 4.11.

```
model = xgb.XGBRegressor()
model.fit(X_train,y_train)
model.save_model("xgboost_model.json")
```

Gambar 4.11 Melatih model Algoritma XGBoost

Setelah model dilatih menggunakan data *training*, selanjutnya melakukan prediksi menggunakan model yang telah dilatih. Proses prediksi model menggunakan algoritma XGBoost dapat dilihat pada Gambar 4.12.

```
pred = model.predict(X_test)
```

Gambar 4.12 Proses prediksi model menggunakan Algoritma XGBoost

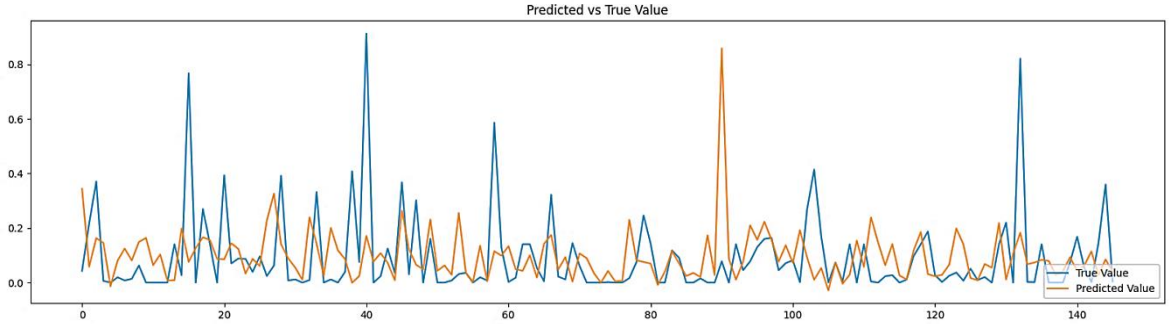
Hasil dari prediksi model menggunakan algoritma XGBoost. Hasil tersebut dapat dilihat pada Tabel 4.4.

Tabel 4.4 Contoh *True value Predict Value* serta *Error* dan *Accuracy Percentage* Algoritma XGBoost

True Value	***Predicted Value***	***Error***	***Accuracy Percentage***
0.042969	0.343434	0.300465	699.2645
0.216406	0.058532	0.157874	72.95268
0.370313	0.163136	0.207177	55.94647
0.00625	0.145447	0.139197	2227.145
0.5	-0.01343	0.513433	102.6865

Dapat dijelaskan pada Tabel 4.4. didalam situasi dimana curah hujan perlu diprediksi, *true value* merujuk pada jumlah curah hujan yang sebenarnya terjadi pada suatu titik waktu dan tempat tertentu, sementara *predict value* adalah nilai yang diperkirakan oleh model berdasarkan variabel cuaca yang dimasukkan ke dalamnya. Secara sederhana, *true value* mencerminkan data aktual yang diperoleh dari pengamatan langsung fenomena alam, sedangkan *predict value* merupakan hasil perkiraan atau estimasi dari model berdasarkan data cuaca yang telah di *input*. *Error* adalah selisih antara *true value* dan *predict value*. *Error* menunjukkan seberapa jauh prediksi model dari nilai sebenarnya. *Accuracy Percentage* memberikan ukuran kesalahan prediksi dalam bentuk persentase relatif terhadap *true value*. Ini menunjukkan seberapa besar *error* dibandingkan dengan nilai sebenarnya.

Dengan menggunakan aplikasi *google colab*, dapat dibuat plot perbandingan antara *predict value* dengan *true value* untuk menunjukkan kemiripan dan memperhitungan nilai kesalahan dalam hasil prediksi. Grafik perbandingan antara *predict value* dengan *true value* yaitu dapat dilihat pada Gambar 4.13.



Gambar 4.13 Grafik *Predicted vs True Value* Algoritma XGBoost

Setelah melakukan prediksi model, selanjutnya adalah menghitung rata-rata persentase akurasi yang didapatkan dari nilai *true value*. Proses dari perhitungan akurasi dapat dilihat pada Gambar 4.14.

```
# Menghitung persentase akurasi
df_results['Accuracy Percentage'] = ((df_results['Error'] / (df_results['True Value']))) * 100

# Opsional: Menampilkan rata-rata akurasi
average_accuracy = 100- (df_results['Accuracy Percentage'].mean()/100)
print(f"Accuracy: {average_accuracy:.2f}%")
```

```
Accuracy: 95.03%
```

Gambar 4.14 Proses dan hasil akurasi Algoritma XGBoost

Hasil akurasi yang didapat dengan menggunakan algoritma XGBoost sebesar 95.03%.

4.3.2. Evaluasi Algoritma XGBoost

Setelah proses implementasi algoritma XGBoost dan mendapatkan hasil prediksi, selanjutnya akan dilakukan proses evaluasi terhadap prediksi. Pada penelitian ini, digunakan *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE) untuk melakukan evaluasi.

1. ***Mean Square Error* (MSE)**

   Berdasarkan evaluasi *Mean Square Error* (MSE) pada gambar dapat dilihat bahwa tingkat akurasi yang dihasilkan dengan menggunakan algoritma XGBoost yakni sebesar 0.02947. Proses dan hasil MSE dapat dilihat pada Gambar 4.15.

```
# Menghitung evaluasi MSE
MSE = mean_squared_error(y_test, pred)
print("MSE: %.5f" % MSE)
```

```
MSE: 0.02947
```

Gambar 4.15 Proses dan hasil MSE Algoritma XGBoost

2. ***Root Mean Square Error* (RMSE)**

   Berdasarkan evaluasi *Root Mean Square Error* (RMSE) pada gambar dapat dilihat bahwa tingkat akurasi yang dihasilkan dengan menggunakan algoritma XGBoost yakni 0.17166. Proses dan hasil RMSE dapat dilihat pada Gambar 4.16.

```
RMSE = np.sqrt(mean_squared_error(y_test, pred))
print("RMSE: %.5f" % RMSE )
```

```
RMSE: 0.17166
```

Gambar 4.16 Proses dan hasil RMSE Algoritma XGBoost

3. ***Mean Absolute Error* (MAE)**

   Berdasarkan evaluasi *Mean Absolute Error* (MAE) pada gambar dapat dilihat bahwa tingkat akurasi yang dihasilkan dengan menggunakan algoritma XGBoost yakni 0.1086098584439765. Proses dan hasil MAE dapat dilihat pada Gambar 4.17.

```
# Menghitung Mean Absolute Error (MAE)
mae = np.mean(np.abs(y_test - pred))
print("MAE:", mae)
```

```
MAE: 0.1086098584439765
```

Gambar 4.17 Proses dan hasil MAE Algoritma XGBoost

4.4. Algoritma *Extreme Learning Machine*

Setelah selesai dalam melakukan proses implementasi menggunakan algoritma xgboost, maka selanjutnya adalah mengimplementasikan algoritma ELM. Selanjutnya dapat dilihat pada Gambar 4.18 merupakan *source code* untuk *import library* yang digunakan.

```
import torch
import torch.nn as nn
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
import numpy as np # linear algebra
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import mean_squared_log_error
```

Gambar 4.18 *Source code library* Algoritma ELM

Fungsi-fungsi dari *library* di atas yaitu *torch* digunakan untuk membangun dan melatih model *neural networks* dan modul *torch.nn* adalah bagian dari *library PyTorch* yang menyediakan berbagai fungsi dan kelas untuk membangun dan melatih *neural networks*. *Numpy* berfungsi untuk memproses sebuah data berbentuk *array*. Pandas untuk mengolah *dataframe* seperti menambah atau menghapus data. *Sklearn* untuk *machine learning*-nya. Modul *model_selection* dari *sklearn* digunakan untuk pemisahan data menjadi set pelatihan dan pengujian. *MinMaxScaler* berfungsi untuk menormalisasi data. *Sklearn.metrics* digunakan untuk menyediakan berbagai metrik evaluasi untuk model pembelajaran mesin. *Matplotlib* berfungsi untuk memvisualisasikan data dalam bentuk grafik. *Seaborn* digunakan untuk memvisualisasikan data dalam analisis eksploratori.

4.4.1. Implementasi Algoritma *Extreme Learning Machine* (ELM)

Setelah dilakukan proses pra-pemrosesan, selanjutnya akan diimplementasikan algoritma ELM terhadap dataset. Implementasi dari *Extreme Learning Machine* (ELM) menggunakan *library PyTorch* untuk pemodelan regresi. Data yang telah dinormalisasi kemudian dikonversi menjadi *tensor PyTorch* menggunakan *torch.tensor*. Ini diperlukan karena *PyTorch* membutuhkan *input* data dalam bentuk tensor untuk pemodelan. Proses dari mengkonversi data menjadi *tensor Pytorch* dapat dilihat pada Gambar 4.19.

```
# Mengkonversi data menjadi tensor PyTorch
X_train_tensor = torch.tensor(X_train_normalized, dtype=torch.float32)
y_train_tensor = torch.tensor(y_train.values, dtype=torch.float32).unsqueeze(1)
X_test_tensor = torch.tensor(X_test_normalized, dtype=torch.float32)
```

Gambar 4.19 Mengkonversi data menjadi tensor *Pytorch*

Implementasi dilakukan dengan mendefinisikan model ELM dan pelatihan model. Model ELM didefinisikan sebagai kelas turunan dari *nn.Module* dalam *PyTorch*. Model ini memiliki satu lapisan tersembunyi (*hidden layer*) dengan fungsi aktivasi *tanh* dan menggunakan *linear layer* untuk menghubungkan *input* dan lapisan tersembunyi serta parameter beta untuk lapisan tersembunyi. Proses ini dapat dilihat pada Gambar 4.20.

```
# Mendefinisikan model Extreme Learning Machine
class ELM(nn.Module):
    def __init__(self, input_size, hidden_size):
        super(ELM, self).__init__()
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.fc = nn.Linear(input_size, hidden_size)
        self.beta = nn.Parameter(torch.randn(hidden_size, 1))

    def forward(self, x):
        h = torch.tanh(self.fc(x))
        output = torch.mm(h, self.beta)
        return output

# Mendefinisikan model ELM
input_size = X_train.shape[1]
hidden_size = 5
model = ELM(input_size, hidden_size)
```

Gambar 4.20 Proses mendefinisikan model ELM

Dilakukan pelatihan model dengan melakukan iterasi sebanyak 750 kali, di mana pada setiap iterasi dilakukan *forward pass*, *input* diberikan ke model dan *output* dari model dihasilkan. Nilai epoch = 750 dipilih karena memberikan hasil akurasi tinggi, hal ini mengacu kepada penelitian sebelumnya (Adiba et al., 2023). Proses pelatihan model dapat dilihat pada Gambar 4.21 berikut.

```
# Pelatihan model
for epoch in range(750):
    outputs = model(X_train_tensor)
    loss = criterion(outputs, y_train_tensor)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

Gambar 4.21 Proses pelatihan model ELM

Setelah model dilatih menggunakan data *training*, selanjutnya melakukan prediksi menggunakan model yang telah dilatih. Proses prediksi model menggunakan algoritma ELM dapat dilihat pada Gambar 4.22.

```
y_pred = model(X_test_tensor).squeeze().numpy()
```

```
# Membuat DataFrame dengan nilai asli dan nilai prediksi
df_results = pd.DataFrame({'True Value': y_test, 'Predicted Value': y_pred})

# Mengubah nilai True Value yang 0 menjadi 0.5
df_results['True Value'] = df_results['True Value'].apply(lambda x: 0.5 if x == 0 else x)

# Menghitung error absolut
df_results['Error'] = np.abs(df_results['True Value'] - df_results['Predicted Value'])
print(df_results)
```

Gambar 4.22 Proses prediksi model menggunakan Algoritma ELM

Hasil dari prediksi model menggunakan algoritma ELM. Hasil tersebut dapat dilihat pada Tabel 4.5.

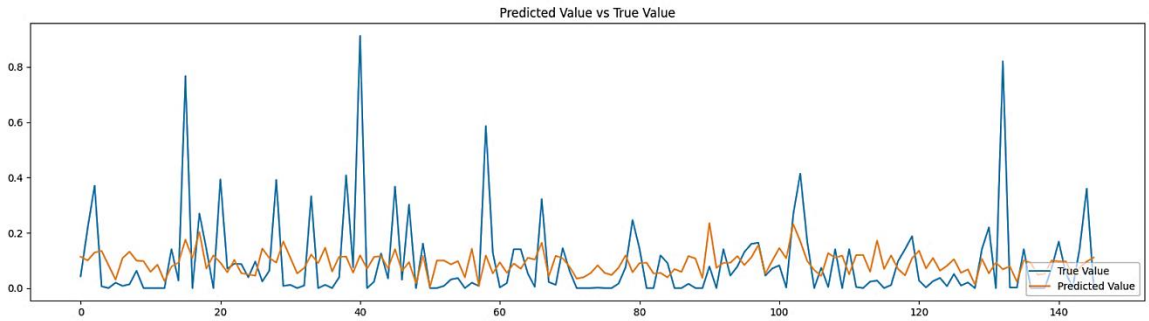
Tabel 4.5 Contoh *True value*, *Predict Value* serta *Error* dan *Accuracy Percentage* Algoritma ELM

True Value	***Predicted Value***	***Error***	***Accuracy Percentage***
0.042969	0.1134727	0.07050395	164.0819203
0.216406	0.10017097	0.11623528	53.71160955
0.370313	0.12886906	0.241443443	65.19991718
0.00625	0.1346043	0.128354305	2053.66888
0.5	0.082226634	0.417773366	83.55467319

Dapat dijelaskan pada Tabel 4.5, didalam situasi dimana curah hujan perlu diprediksi, *true value* merujuk pada jumlah curah hujan yang sebenarnya terjadi pada suatu titik waktu dan tempat tertentu, sementara *predict value* adalah nilai yang diperkirakan oleh model berdasarkan variabel cuaca yang dimasukkan ke dalamnya. Secara sederhana, *true value* mencerminkan data aktual yang diperoleh dari pengamatan langsung fenomena alam, sedangkan *predict value* merupakan hasil perkiraan atau estimasi dari model berdasarkan data cuaca yang telah di *input*.

Error adalah selisih antara *true value* dan *predict value*. *Error* menunjukkan seberapa jauh prediksi model dari nilai sebenarnya. *Accuracy Percentage* memberikan ukuran kesalahan prediksi dalam bentuk persentase relatif terhadap *true value*. Ini menunjukkan seberapa besar *error* dibandingkan dengan nilai sebenarnya.

Dengan menggunakan aplikasi *google colab* dapat dijelaskan menggunakan plot perbandingan antara *predict value* dengan *true value* untuk memperlihatkan kemiripan dan memperhitungan nilai kesalahan pada hasil peramalan. Grafik perbandingan antara *predict value* dengan *true value* yaitu dapat dilihat pada Gambar 4.23 berikut.



Gambar 4.23 Grafik *Predicted vs True Value* Algoritma ELM

Setelah melakukan prediksi model, selanjutnya adalah menghitung rata-rata persentase akurasi yang didapatkan dari nilai *true value*. Proses dari perhitungan akurasi dapat dilihat pada Gambar 4.24.

```
# Menghitung persentase akurasi
df_results['Accuracy Percentage'] = ((df_results['Error'] / (df_results['True Value']))) * 100

# Opsional: Menampilkan rata-rata akurasi
average_accuracy = 100- (df_results['Accuracy Percentage'].mean()/100)
print(f"Accuracy: {average_accuracy:.2f}%")
```

```
Accuracy: 95.75%
```

Gambar 4.24 Proses dan hasil akurasi Algoritma ELM

Hasil akurasi yang didapat dengan menggunakan algoritma ELM sebesar 95.75%.

4.4.2. Evaluasi Algoritma ELM

Setelah proses implementasi algoritma ELM, selanjutnya akan dilakukan proses evaluasi terhadap prediksi. Pada penelitian ini, digunakan *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE) untuk melakukan evaluasi.

1. ***Mean Square Error* (MSE)**

   Pada penelitian ini peneliti menggunakan percobaan jumlah *hidden layer* 5,10,20,30,40 dan 50 untuk membandingkan dan bertujuan untuk mendapatkan jumlah *neuron* terbaik yang didapatkan berdasarkan nilai MSE. Dapat dilihat pada Tabel 4.6 yang merupakan hasil dari percobaan jumlah *hidden layer* menggunakan perhitungan akurasi nilai MSE.

Tabel 4.6 Percobaan nilai MSE dari *hidden layer*

Jumlah *hidden neuron*	Nilai MSE
5	0.02151830504635791
10	0.02168109091082995
20	0.021682925717766723
30	0.02179986159314741
40	0.021686497509227534
50	0.022027857515132882

Dapat dilihat pada Tabel 4.6. uji coba jumlah *neuron* pada *hidden layer*, penelitian yang menggunakan jumlah *neuron* sebanyak 50 menghasilkan nilai MSE yaitu sebesar 0.022027857515132882, sedangkan untuk jumlah *neuron* sebanyak 5 menghasilkan nilai MSE 0.02151830504635791. Sehingga pada penelitian ini menggunakan jumlah *neuron* sebanyak 5 dikarenakan mempunyai MSE paling kecil yaitu 0.02151830504635791.

Berdasarkan evaluasi *Mean Square Error* (MSE) dapat disimpulkan bahwa nilai evaluasi yang dihasilkan dengan menggunakan jumlah *neuron* sebanyak 5 menggunakan algoritma ELM yakni sebesar 0.02151830504635791. Proses dan hasil MSE dapat dilihat pada Gambar 4.25.

```
# Evaluasi model
mse = mean_squared_error(y_test, y_pred)
print("MSE:", mse)
```

```
MSE: 0.02151830504635791
```

Gambar 4.25 Proses dan nilai evaluasi MSE Algoritma ELM

2. ***Root Mean Square Error* (RMSE)**

   Berdasarkan evaluasi *Root Mean Square Error* (RMSE) pada gambar dapat dilihat bahwa tingkat akurasi yang dihasilkan dengan menggunakan algoritma ELM yakni 0.14669. Proses dan hasil RMSE dapat dilihat pada Gambar 4.26.

```
RMSE = np.sqrt(mean_squared_error(y_test, y_pred))
print("RMSE: %.5f" % RMSE )
```

```
RMSE: 0.14669
```

Gambar 4.26 Proses dan nilai evaluasi RMSE Algoritma ELM

3. ***Mean Absolute Error* (MAE)**

   Berdasarkan evaluasi *Mean Absolute Error* (MAE) pada gambar dapat dilihat bahwa tingkat akurasi yang dihasilkan dengan menggunakan algoritma ELM yakni 0.09427524057969654. Proses dan hasil MAE dapat dilihat pada Gambar 4.27.

```
# Menghitung Mean Absolute Error (MAE)
mae = np.mean(np.abs(y_test - y_pred))
print("MAE:", mae)
```

```
MAE: 0.09427524057969654
```

Gambar 4.27 Proses dan nilai evaluasi MAE Algoritma ELM

4.5. Proses Prediksi dan Hasil Prediksi Tingkat Curah Hujan di Kota Medan

Setelah mendapatkan hasil prediksi dari model dan melakukan evaluasi, tahap terakhir pada penelitian ini adalah memprediksi tingkat curah hujan dan melakukan denormalisasi data prediksi. Proses prediksi tingkat curah hujan dari masing-masing algoritma dapat dilihat pada Gambar 4.28 dan Gambar 4.29.

```python
import datetime

# Tambahkan kolom tanggal sesuai dengan tanggal yang Anda tentukan
custom_dates = ['2024-07-1', '2024-07-2', '2024-07-3', '2024-07-4', '2024-07-5', '2024-07-6', '2024-07-7', '2024-07-8', '2024-07-9', '2024-07-10', '2024-07-11',

# Prediksi menggunakan model yang telah dilatih
data_input = pd.DataFrame({
    'temp_min': [20, 21, 22, 23, 24, 25, 26, 27, 26, 25, 24, 23],  # Contoh: Temperatur minimum
    'temp_max': [30, 31, 32, 33, 34, 35, 36, 37, 36, 35, 34, 33],  # Contoh: Temperatur maksimum
    'temp_mean': [25, 26, 27, 28, 29, 30, 31, 32, 31, 30, 29, 28],  # Contoh: Temperatur rata-rata
    'humidity': [70, 65, 60, 55, 50, 45, 40, 45, 50, 55, 60, 65],  # Contoh: Kelembaban udara
    'light_sun': [8, 9, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11],  # Contoh: Sinar matahari
    'wind_max': [20, 21, 22, 23, 24, 25, 26, 27, 28, 27, 26, 25],  # Contoh: Kecepatan angin maksimum
    'wind_direction': [90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200],  # Contoh: Arah angin
    'wind_speed': [10, 11, 12, 13, 14, 15, 16, 17, 18, 17, 16, 15],  # Contoh: Kecepatan angin

})
# Buang kolom 'rainfall' dari data input
data_input = data_input[['temp_min', 'temp_max', 'temp_mean', 'humidity', 'light_sun', 'wind_max', 'wind_direction', 'wind_speed']]

# Muat kembali model yang telah disimpan
loaded_model = xgb.XGBRegressor()
loaded_model.load_model("xgboost_model.json")

# Lakukan prediksi menggunakan model yang telah disimpan
preds = loaded_model.predict(data_input)
```

Gambar 4.28 Proses prediksi tingkat curah hujan menggunakan Algoritma XGBoost

```python
import pandas as pd
import torch

custom_dates = ['2024-07-1', '2024-07-2', '2024-07-3', '2024-07-4', '2024-07-5', '2024-07-6', '2024-07-7', '2024-07-8', '2024-07-9', '2024-07-10', '2024-07-11',

# Prediksi menggunakan model yang telah dilatih
data_input = pd.DataFrame({
    'temp_min': [20, 21, 22, 23, 24, 25, 26, 27, 26, 25, 24, 23],  # Contoh: Temperatur minimum
    'temp_max': [30, 31, 32, 33, 34, 35, 36, 37, 36, 35, 34, 33],  # Contoh: Temperatur maksimum
    'temp_mean': [25, 26, 27, 28, 29, 30, 31, 32, 31, 30, 29, 28],  # Contoh: Temperatur rata-rata
    'humidity': [70, 65, 60, 55, 50, 45, 40, 45, 50, 55, 60, 65],  # Contoh: Kelembaban udara
    'light_sun': [8, 9, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11],  # Contoh: Sinar matahari
    'wind_max': [20, 21, 22, 23, 24, 25, 26, 27, 28, 27, 26, 25],  # Contoh: Kecepatan angin maksimum
    'wind_direction': [90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200],  # Contoh: Arah angin
    'wind_speed': [10, 11, 12, 13, 14, 15, 16, 17, 18, 17, 16, 15],  # Contoh: Kecepatan angin
})
# Normalisasi data input menggunakan MinMaxScaler yang sama
data_input_normalized = scaler.fit_transform(data_input)

# Konversi data input menjadi tensor PyTorch
data_input_tensor = torch.tensor(data_input_normalized, dtype=torch.float32)

# Lakukan prediksi menggunakan model yang telah dimuat kembali
with torch.no_grad():
    preds_tensor = loaded_model(data_input_tensor)
    preds_numpy = preds_tensor.squeeze().numpy()  # Konversi hasil prediksi ke dalam bentuk array numpy
```

Gambar 4.29 Proses prediksi tingkat curah hujan menggunakan Algoritma ELM

Setelah memprediksi tingkat curah hujan, selanjutnya data prediksi melakukan tahap denormalisasi. Tujuan dari denormalisasi pada prediksi tingkat curah hujan adalah untuk mengembalikan skala data ke dalam rentang yang semula setelah proses normalisasi. Proses normalisasi dilakukan untuk memastikan bahwa setiap fitur memiliki skala yang serupa, yang membantu algoritma XGBoost dan ELM untuk konversi lebih cepat dan menghasilkan model yang lebih baik. Namun, setelah proses prediksi selesai, hasil prediksi yang diperoleh memiliki skala yang dinormalisasi. Oleh karena itu, denormalisasi dilakukan untuk mengembalikan hasil prediksi ke dalam satuan yang sesuai dengan data aslinya, sehingga hasil prediksi dapat lebih mudah dipahami dan diinterpretasikan oleh pengguna akhir. Proses ini penting dilakukan agar hasil prediksi dapat diterapkan secara praktis dalam konteks keputusan

dan tindakan yang diambil terkait dengan tingkat curah hujan yang diprediksi. Dapat dilihat proses dari denormalisasi pada Gambar 4.30 berikut.

```
# Denormalisasi hasil prediksi
denormalized_preds = preds_numpy * (rainfall_max - rainfall_min) + rainfall_min
```

Gambar 4.30 Proses denormalisasi

Selanjutnya dapat dilihat pada Gambar 4.31 dan Gambar 4.32 yang merupakan proses klasifikasi untuk hasil prediksi menggunakan algoritma XGBoost dan algoritma ELM berdasarkan nilai dan kategori tingkat curah hujan dari BMKG, kategori curah hujan yang dapat dilihat pada Tabel 3.2.

```
# Tambahkan kondisi elif untuk klasifikasi curah hujan
klasifikasi = []
for pred in preds_denormalized:
    if pred == 0:
        klasifikasi.append('Berawan')
    elif 0.5 <= pred <= 20:
        klasifikasi.append('Hujan Ringan')
    elif 20 < pred <= 50:
        klasifikasi.append('Hujan Sedang')
    elif 50 < pred <= 100:
        klasifikasi.append('Hujan Lebat')
    elif 100 < pred <= 150:
        klasifikasi.append('Hujan Sangat Lebat')
    else:
        klasifikasi.append('Hujan Ekstrem')
```

Gambar 4.31 Proses klasifikasi tingkat curah hujan algoritma XGBoost

```
# Tambahkan kolom klasifikasi berdasarkan kondisi yang diberikan
klasifikasi = pd.cut(denormalized_preds, bins=[-np.inf, 0.5, 20, 50, 100, 150, np.inf],
                     labels=['Berawan', 'Hujan Ringan', 'Hujan Sedang', 'Hujan Lebat', 'Hujan Sangat Lebat', 'Hujan Ekstrem'])
```

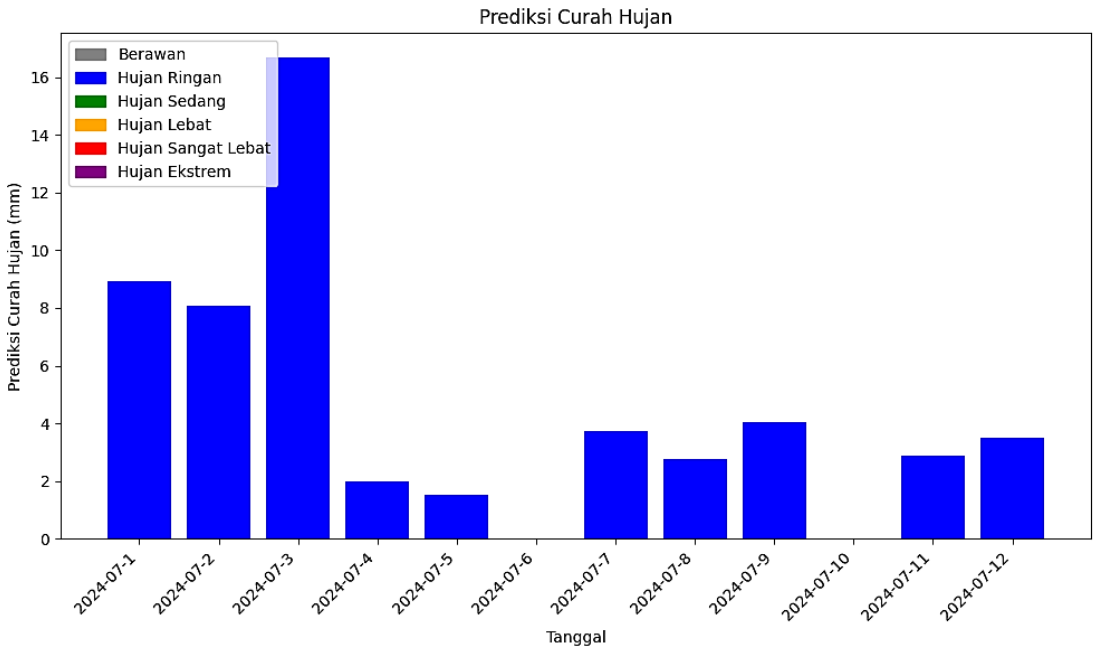
Gambar 4.32 Proses klasifikasi tingkat curah hujan algoritma ELM

Berikut ini pada Tabel 4.7 merupakan hasil dari prediksi data curah hujan Kota Medan dalam menggunakan Algoritma *Extreme Gradient Boosting* (XGBoost). Tabel 4.6 juga menjelaskan tentang perbandingan jumlah data yang ditunjukkan dengan skala angka dan klasifikasi berdasarkan kategori curah hujan dari BMKG.

Tabel 4.7 Hasil prediksi tingkat curah hujan menggunakan Algoritma *Extreme Gradient Boosting*

Tanggal	Hasil Prediksi	Denormalisasi	Kategori
1/07/2024	0.06985719	8.9	Hujan Ringan
2/07/2024	0.06317614	8.1	Hujan Ringan
3/07/2024	0.13040815	16.6	Hujan Ringan
4/07/2024	0.01545779	1.9	Hujan Ringan
5/07/2024	0.0119561	1.5	Hujan Ringan
6/07/2024	-0.00989455	0	Berawan
7/07/2024	0.02923567	3.7	Hujan Ringan
8/07/2024	0.02150253	2.7	Hujan Ringan
9/07/2024	0.0316697	4	Hujan Ringan
10/07/2024	-0.00746051	0	Berawan
11/07/2024	0.02256611	2.8	Hujan Ringan
12/07/2024	0.0274922	3.5	Hujan Ringan

Dapat dilihat pada Gambar 4.33 yang merupakan visual dari hasil prediksi yang dilakukan menggunakan algoritma XGBoost.



Gambar 4.33 Visualisasi hasil prediksi tingkat curah hujan menggunakan Algoritma XGBoost

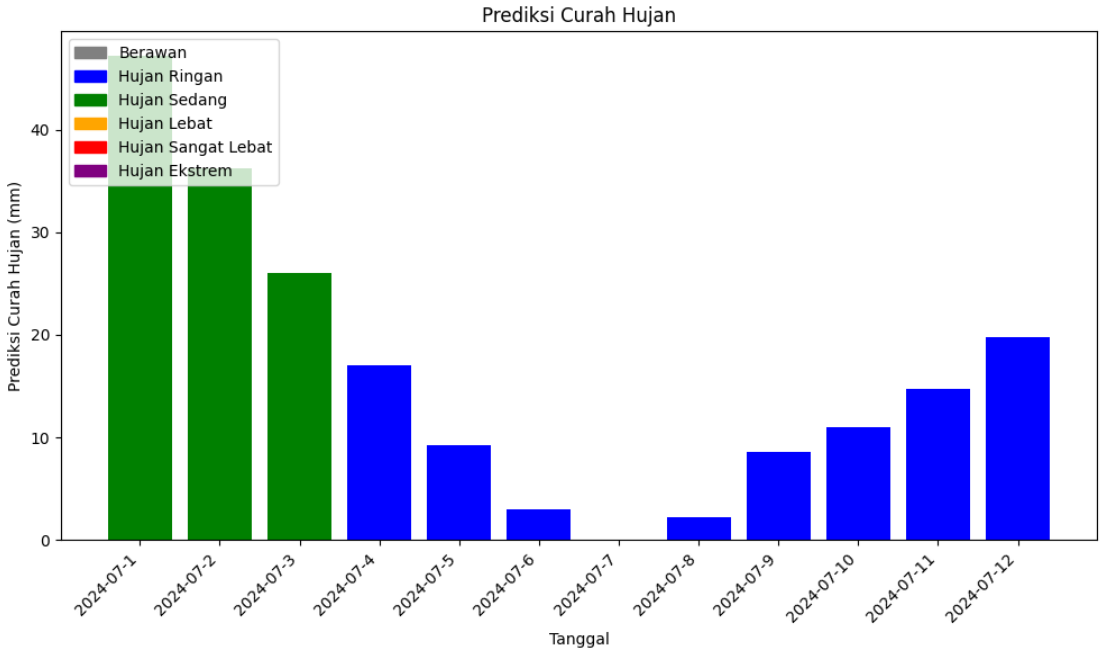
Hasil dari prediksi data curah hujan di Kota Medan dalam menggunakan Algoritma *Extreme Learning Machine* (ELM) dapat dilihat pada Tabel 4.8. Tabel 4.8 merupakan hasil dari prediksi data curah hujan Kota Medan dalam menggunakan Algoritma *Extreme Learning Machine* (ELM). Tabel 4.8 menjelaskan tentang perbandingan jumlah data yang ditunjukkan dengan skala angka dan klasifikasi berdasarkan kategori curah hujan.

Tabel 4.8 Hasil prediksi tingkat curah hujan menggunakan Algoritma *Extreme Learning Machine*

Tanggal	Hasil Prediksi	Denormalisasi	Kategori
1/07/2024	0.36884794	47.2	Hujan Sedang
2/07/2024	0.28312847	36.2	Hujan Sedang
3/07/2024	0.20351693	26	Hujan Sedang
4/07/2024	0.13275886	16.9	Hujan Ringan
5/07/2024	0.07249004	9.2	Hujan Ringan
6/07/2024	0.02326995	2.9	Berawan
7/07/2024	-0.01520205	0	Berawan

8/07/2024	0.01776278	2.2	Berawan
9/07/2024	0.0667752	8	Hujan Ringan
10/07/2024	0.0860191	11	Hujan Ringan
11/07/2024	0.11482966	14.6	Hujan Ringan
12/07/2024	0.15481007	19.8	Hujan Ringan

Dapat dilihat pada Gambar 4.34 yang merupakan visual dari hasil prediksi yang dilakukan algoritma ELM.



Gambar 4.34 Visualisasi hasil prediksi tingkat curah hujan menggunakan Algoritma ELM

4.6. Hasil Perbandingan Algoritma *Extreme Gradient Boosting* dan Algoritma *Extreme Learning Machine*

Pada tahap ini adalah proses membandingkan hasil nilai akurasi yang diperoleh dari Algoritma *Extreme Gradient Boosting* (XGBoost) dan *Extreme Learning Machine* (ELM) yang dilakukan dengan analisis dari nilai evaluasi dan nilai akurasi yang didapatkan pada setiap algoritma. Perbandingan nilai evaluasi prediksi dengan *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE), dan *Mean Absolute Error* (MAE) untuk mengetahui algoritma mana yang paling tepat yang menghasilkan nilai

tingkat kesalahan (*error*) yang paling kecil. Perbandingan Hasil Nilai Evaluasi algoritma *Extreme Gradient Boosting* (XGBoost) dan *Extreme Learning Machine* (ELM) dapat dilihat pada Tabel 4.9.

Tabel 4.9 Perbandingan Hasil nilai evaluasi dari Algoritma XGBoost dan Algoritma ELM

Model	**Evaluasi**		
	MSE	**RMSE**	**MAE**
XGBOOST	0.02947	0.17166	0.10860
ELM	0.02151	0.14669	0.09427
Model Terbaik	**ELM**	**ELM**	**ELM**

Dapat dilihat pada Tabel 4.9 menunjukkan bahwa perbandingan nilai MSE pada XGBoost didapatkan nilai sebesar 0.02947 sedangkan ELM nilai MSE sebesar 0.02151 dimana hasil menunjukkan bahwa ELM memiliki tingkat kesalahannya lebih kecil daripada XGBoost. Nilai RMSE pada XGBoost didapatkan nilai sebesar 0.17166 sedangkan ELM nilai RMSE sebesar 0.14669 dimana hasil menunjukkan bahwa ELM memiliki tingkat kesalahannya lebih kecil daripada XGBoost. Nilai MAE pada XGBoost didapatkan nilai sebesar 0.10860 sedangkan ELM nilai MAE sebesar 0.09427 dimana hasil menunjukkan bahwa ELM memiliki tingkat kesalahannya lebih kecil daripada XGBoost.

Perbandingan Hasil Nilai akurasi algoritma *Extreme Gradient Boosting* (XGBoost) dan *Extreme Learning Machine* (ELM) dapat dilihat pada Tabel 4.10.

Tabel 4.10 Perbandingan Hasil nilai akurasi dari Algoritma XGBoost dan Algoritma ELM

Model	**Nilai Akurasi**
XGBOOST	95.03%
ELM	95.75%
Model Terbaik	**ELM**

Pada Tabel 4.10 menunjukkan bahwa nilai akurasi yang didapatkan dari algoritma XGBoost sebesar 95.03%, sedangkan yang didapat dari algoritma ELM sebesar 95.75%.

Berdasarkan analisis pada Tabel 4.9 dan Tabel 4.10 menunjukkan algoritma *Extreme Learning Machine* lebih efektif digunakan dalam prediksi karena memiliki nilai evaluasi MSE, RMSE, MAE yang lebih kecil dan nilai akurasi yang tinggi dibandingkan dengan algoritma *Extreme Gradient Boosting*.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Adapun hasil yang diperoleh dari setiap tahapan analisis, perancangan dan pengujian sistem yang telah dilakukan pada penelitian ini dengan mengimplementasikan algoritma *extreme gradient boosting* dan algoritma *extreme learning machine* yang digunakan untuk memprediksi tingkat curah hujan di Kota Medan.

1. Hasil penelitian menunjukkan bahwa kedua algoritma tersebut mampu memberikan prediksi yang dapat diterima, namun dengan karakteristik yang berbeda. Pemilihan antara kedua algoritma ini dapat dipertimbangkan berdasarkan kebutuhan spesifik dari prediksi tingkat curah hujan, termasuk pertimbangan terkait dengan tingkat akurasi. Penelitian ini memberikan pemahaman yang lebih dalam tentang potensi dari masing-masing algoritma, serta arahan untuk pengembangan model prediksi tingkat curah hujan yang lebih baik di masa depan.
2. Hasil perbandingan algoritma *Extreme Gradient Boosting* (XGBoost) dan algoritma *Extreme Learning Machine* (ELM) jika dibandingkan dengan nilai *Mean Square Error* (MSE) pada algoritma XGBoost didapatkan nilai 0.02947, sedangkan pada algoritma ELM nilai MSE sebesar 0.02151 dengan jumlah *hidden layer* 5, dimana hasil menunjukkan bahwa algoritma ELM tingkat kesalahannya lebih kecil daripada algoritma XGBoost. Pada algoritma XGBoost nilai RMSE sebesar 0.17166, sedangkan pada ELM sebesar dimana 0.14669 hasil menunjukkan bahwa algoritma ELM tingkat kesalahannya lebih kecil daripada algoritma XGBoost. Pada algoritma XGBoost nilai MAE sebesar 0.10860, sedangkan pada ELM sebesar 0.09427. Nilai akurasi yang di peroleh dari algoritma *Extreme Learning Machine* sebesar 95.75%, sedangkan

nilai akurasi dari algoritma *Extreme Gradient Boosting* hanya 95.03%. algoritma XGBoost.

3. Berdasarkan hasil penelitian, nilai akurasi yang di peroleh dari algoritma *Extreme Learning Machine* lebih tinggi dibandingkan dengan algoritma *Extreme Gradient Boosting* serta nilai evaluasi yang diperoleh dari *Extreme Learning Machine* lebih rendah dan mendekati nilai 0 dibandingkan *Extreme Gradient Boosting*.

5.2. Saran

Adapun saran yang dapat dipertimbangkan untuk penelitian selanjutnya antara lain :

1. Dapat menambah jumlah data harian yang banyak dan dalam jangka waktu yang panjang untuk menambah akurasi prediksi yang lebih baik. Peneliti selanjutnya dapat menggunakan data bulanan maupun tahunan untuk mengamati angka prediksi dalam jangka pendek dan jangka panjang.
2. Dapat menggabungkan data cuaca dari berbagai sumber, termasuk data satelit, stasiun cuaca darat, dan sensor cuaca di udara, untuk meningkatkan akurasi dan cakupan prediksi tingkat curah hujan.
3. Peneliti selanjutnya dapat fokus pada pengembangan sistem peringatan dini yang memanfaatkan model prediksi curah hujan. Sistem ini bisa digunakan untuk memberikan peringatan kepada masyarakat dan pemerintah daerah mengenai potensi banjir atau cuaca ekstrem.

DAFTAR PUSTAKA

Anwar, M. T., Winarno, E., Hadikurniawati, W., & Novita, M. (2021). *Rainfall prediction using Extreme Gradient Boosting*. *Journal of Physics: Conference Series*, 1869(1). https://doi.org/10.1088/1742-6596/1869/1/012078

Chen, T., & Guestrin, C. (2016). XGBoost: *A scalable tree boosting system. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-*August*, 785–794. https://doi.org/10.1145/2939672.2939785

Dewi, S. N., Cholissodin, I., & Santoso, E. (2018). Prediksi Jumlah Kriminalitas Menggunakan Metode *Extreme Learning Machine* (Studi Kasus Di Kabupaten Probolinggo). Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya, 2(11), 4687–4693.

Freecenta, H. F., Yulia Puspaningrum, E., & Maulan, H. (2022). Prediksi Curah Hujan Di Kab. Malang Menggunakan LSTM (*Long Short Term Memory*). Jurnal Informatika Dan Sistem Informasi, 3(1), 51–55. https://doi.org/10.33005/jifosi.v3i1.448

Herni Yulianti, S. E., Oni Soesanto, & Yuana Sukmawaty. (2022). Penerapan Metode *Extreme Gradient Boosting* (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit. *Journal of Mathematics: Theory and Applications*, 4(1), 21–26. https://doi.org/10.31605/jomta.v4i1.1792

Ikhwani, Y. (2017). APLIKASI INFORMASI CUACA EKSTRIM DAN GEMPA BUMI Fakultas Teknologi Informasi Universitas Islam Kalimantan Muhammad Arsyad Al Banjari Banjarmasin Yusri Ikhwani. *Technologia*, 8(3), 191–195.

Jange, B. (2021). Prediksi Harga Saham Bank BCA Menggunakan XGBoost. *Journal of Trends Economics and Accounting …*, *2*(1), 1–5. https://doi.org/10.47065/arbitrase.v3i2.495

Khumaidi, A., Raafi'udin, R., & Solihin, I. P. (2020). Pengujian Algoritma *Long Short-Term Memory* untuk Prediksi Kualitas Udara dan Suhu Kota Bandung. Jurnal Telematika, 15(1), 13–18.

Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). *A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. AI Magazine*, 38(4), 13–26.

https://doi.org/10.1609/aimag.v38i4.2744

Manalu, M. T. P. (2016). Jaringan Syaraf Tiruan untuk Memprediksi Curah Hujan Sumatera Utara dengan Metode *Back Propagation* (Studi Kasus : BMKG Medan). Jurnal Riset Komputer, 3(1), 35–40.

Pandika Pinata, N. N., Sukarsa, I. M., & Dwi Rusjayanthi, N. K. (2020). Prediksi Kecelakaan Lalu Lintas di Bali dengan XGBoost pada *Python*. Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi), 8(3), 188. https://doi.org/10.24843/jim.2020.v08.i03.p04

Rizka Wiyossabhi Fenia. (2023). Ombrometer *:* Pengertian, Kelebihan, Kekurangan, dan Peran dalam Pertanian. Mertani Web. https://www.mertani.co.id/post/ombrometer-pengertian-kelebihan-kekurangan-dan-peran-dalam-pertanian

Simamora, R. J. D., Tibyani, & Sutrisno. (2019). Peramalan Curah Hujan Menggunakan Metode *Extreme Learning Machine*. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 3(10), 9670–9676. http://j-ptiik.ub.ac.id

Sulasmi, S., Setyaningtyas, D. E., Rosanji, A., & Rahayu, N. (2019). Pengaruh curah hujan, kelembaban, dan temperatur terhadap prevalensi Malaria di Kabupaten Tanah Bumbu Kalimantan Selatan. *Journal of Health Epidemiology and Communicable Diseases*, 3(1), 22–27. https://doi.org/10.22435/jhecds.v3i1.1794

LAMPIRAN

Lampiran 1 Dataset BMKG Periode 1 Januari 2022 - 31 Desember 2023

Tanggal	Tempe-ratur minimum (°C	Tempe-ratur maksimum (°C)	Tempe-ratur rata-rata (°C)	Kelem-bapan rata-rata (%)	Curah Hujan (mm)	Lamanya penyi-naran matahari (jam)	Kecepatan angin maksimum (m/s)	Arah angin saat kecepatan maksimum (°)	Kecepatan angin rata-rata (m/s)
1/1/2022	24	28.8	24.6	96	12.9	0.2	7	320	2
2/1/2022	23.5	26.6	24.8	94	26.5	1.5	4	300	2
3/1/2022	23.8	31.3	26.6	85	6.5	0	2	210	1
4/1/2022	23.8	31.2	26.8	88	3.2	3.1	3	290	1
5/1/2022	22.1	32	26.1	84	21.1	1.6	3	320	2
6/1/2022	22.3	31.8	27.1	84	0	6.4	4	300	2
7/1/2022	21.2	33	27.4	80	0	5.1	4	60	1
8/1/2022	23.4	31.8	27	87	0	4.1	3	290	1
9/1/2022	23.4	33.8	27.2	84	1.4	3.2	3	280	2
10/1/2022	24	31	26.4	84	18.5	3	6	280	2
11/1/2022	23	32	27.2	84	3.5	1	3	60	1
12/1/2022	22.4	34.8	27	78	0	5	3	50	1
13/1/2022	22.1	32.6	27.5	81	0	7.5	5	20	2
14/1/2022	23	32.2	26.7	81	3	3.5	2	340	1
15/1/2022	22.6	33	27.1	80	0	4.8	3	20	2
16/1/2022	23	34.6	28.7	78	0	6.7	3	90	2
17/1/2022	25	32.2	28.1	80	0	6.9	4	340	2
18/1/2022	23	29.8	25.7	87	26.2	4.5	5	270	2
19/1/2022	23.2	32.6	26.6	81	0.1	0.6	4	270	2
20/1/2022	23.2	33	27.4	79	0	5	6	30	2
21/1/2022	21	33.4	27.4	78	0.5	6.5	4	30	2
22/1/2022	21	32.6	27.6	79	2.5	7.4	6	350	2
23/1/2022	24.2	31	27.4	85	0	4.2	4	40	2
24/1/2022	21.8	31	27.6	81	5.3	2.3	3	60	1
25/1/2022	23.8	31.2	26.3	90	12.5	3.2	2	50	1
26/1/2022	23.4	30.6	26.8	87	51.5	0.4	3	40	1
27/1/2022	21.2	30.6	28.3	81	0	1.1	4	200	2
28/1/2022	21.9	34.8	27.5	80	0	6.5	4	130	2
29/1/2022	22.8	32.6	27.8	83	0	7.9	3	50	1
30/1/2022	24	32.2	27.8	83	18	5.2	5	60	2
31/1/2022	23	31.6	27.1	85	18	4.3	3	290	1
1/2/2022	21.4	33.4	27	83	0	1.3	2	10	1
2/2/2022	23.2	32.8	27.3	81	0	6.7	4	350	2
3/2/2022	23.4	32.3	27	82	0	3.8	6	50	1
4/2/2022	19.8	32.4	27.1	84	15.5	5.1	3	10	1

5/2/2022	23.6	33.6	26.6	85	0.3	5.5	5	100	1
6/2/2022	23.8	33.4	27.2	84	12	3.4	6	220	2
7/2/2022	23.8	33	27.4	83	0.3	2.8	3	350	1
8/2/2022	20.4	31	26.2	87	0	3.5	4	300	2
9/2/2022	23.6	32.8	27.6	84	3.8	1.6	3	270	1
10/2/2022	24.4	31	25.6	91	1.1	3.4	3	300	1
11/2/2022	22.4	32.3	26.3	82	9	0	4	40	1
12/2/2022	22.4	31	27.1	84	1.5	3.4	4	280	2
13/2/2022	22.6	33	27.4	78	0	0.7	6	320	2
14/2/2022	23.2	33.6	27.3	85	0	6.3	3	270	2
15/2/2022	23.2	33.4	26.7	87	22.9	4.3	3	240	1
16/2/2022	20	32.6	27.6	83	21.6	5.4	5	150	1
17/2/2022	24	30.8	26.9	88	9.6	3.7	3	30	1
18/2/2022	24	34.7	28.7	80	18	1.2	6	70	2
19/2/2022	24.2	32.7	28.3	84	0.6	7.6	4	70	1
20/2/2022	24.2	33	27.3	82	0	4	4	120	1
21/2/2022	23.4	34.2	27.7	81	9.4	4.4	3	260	1
22/2/2022	23.8	33	27.5	82	9	6.8	2	80	1
23/2/2022	23.6	32	27.1	84	0	2.6	3	330	1
24/2/2022	19.2	29.8	26.3	89	2.5	1.2	2	260	1
25/2/2022	23.4	30.2	25.1	88	9	0.5	6	280	4
26/2/2022	23.4	29.6	25	91	47.4	0.3	6	300	3
27/2/2022	22.8	31.6	25.6	90	1.6	2.4	9	40	1
28/2/2022	17.9	29.8	25.4	93	128	1.6	3	60	1
1/3/2022	23	30.6	25.7	90	41.2	1.2	3	70	2
2/3/2022	23.4	33.2	27.1	85	1.5	1.2	7	60	3
3/3/2022	23.6	32.8	27	86	46	7.6	5	30	2
4/3/2022	23.8	33.2	27.9	80	46	5.9	5	280	2
5/3/2022	25	33	27.6	82	5.2	7.3	6	280	3
6/3/2022	24	33.2	27.7	81	1.8	4.7	6	20	2
7/3/2022	24.6	33	28.2	82	0	3.4	4	280	2
8/3/2022	23.2	33	26.4	87	10.1	3.5	4	110	2
9/3/2022	23.4	33.8	27.3	85	18	4.6	5	50	2
10/3/2022	22.8	32	26.8	85	75.2	5.4	6	40	2
11/3/2022	22.8	34	28	82	0	2.8	7	240	2
12/3/2022	23.8	33.8	28.1	82	0	5.9	4	50	2
13/3/2022	23.8	33.4	27.6	84	0	7.2	5	40	2
14/3/2022	23.8	34	27.9	81	5.4	8.2	7	290	2
15/3/2022	24.2	34.8	27.8	83	0	7.3	3	260	1
16/3/2022	24	33.8	27.1	86	18	4.7	4	20	1
17/3/2022	24	33.6	26.9	88	2.3	4.2	8	220	2
18/3/2022	23.9	33.8	27.9	83	9.4	1.6	3	210	1
19/3/2022	24.2	36.6	29.6	76	0	1.3	6	230	3
20/3/2022	24.4	36.2	29.8	74	0	7.7	5	200	2
21/3/2022	25	33.2	28.7	81	0	7.5	3	350	1
22/3/2022	24.8	35	29.1	80	18	3.8	3	10	1

23/3/2022	25	35.2	28.7	81	14	5.7	6	50	2
24/3/2022	22.4	34.2	28.2	84	18.6	5.4	5	50	2
25/3/2022	24	34.2	28.2	82	0	4.8	3	30	1
26/3/2022	24.2	35	28.1	82	0	4.3	3	340	2
27/3/2022	22.2	33.6	27.7	79	1.7	5.5	4	280	1
28/3/2022	23.5	34.8	28.3	78	1.7	3	5	10	2
29/3/2022	24.2	34.5	28.5	80	0	9.2	3	270	1
30/3/2022	24.8	34.6	28.9	79	0	4.1	3	310	1
31/3/2022	21.7	33.2	28.1	79	0	7.5	4	270	1
1/4/2022	21.6	34.4	28.7	78	0	2.4	3	270	1
2/4/2022	25	33.4	28.5	78	0	7	5	20	2
3/4/2022	23.4	32.8	26.4	86	18.3	1.7	6	340	2
4/4/2022	23.2	33	27.6	80	18	2.6	3	320	1
5/4/2022	20.4	32.3	27.2	83	0.5	3.1	4	300	1
6/4/2022	24.2	31.6	27	86	1.5	2.5	3	330	1
7/4/2022	24	28	25.7	87	0.3	1.9	3	230	1
8/4/2022	24	34.2	26.5	85	0	0.8	10	290	2
9/4/2022	23	33.2	26.8	84	50.3	7.1	3	300	2
10/4/2022	23	34.2	28.1	81	0	2.9	5	70	1
11/4/2022	24.2	34.5	27.4	84	0	7.5	10	60	2
12/4/2022	23.2	34.2	27	84	24	4.8	3	50	1
13/4/2022	23.2	33.4	27.3	86	4.6	6.5	4	300	1
14/4/2022	23.8	35.2	28.5	79	3.2	6	5	280	2
15/4/2022	24.4	34.4	28.6	79	1	8.8	2	140	1
16/4/2022	24.4	35.2	28.7	78	0	6	6	50	2
17/4/2022	25.6	35.6	29.4	75	2	6.6	4	250	2
18/4/2022	25.4	34.2	28	84	0	7.5	7	60	2
19/4/2022	24	33.6	27.6	84	1.1	4.7	4	10	1
20/4/2022	24	32.2	27.2	86	24.8	5.2	5	20	1
21/4/2022	24.4	33.4	28.3	83	0.4	3.4	4	40	1
22/4/2022	24.8	32.8	28.7	83	0	3.2	3	60	1
23/4/2022	25.4	31.6	27.4	87	0	1.8	3	230	1
24/4/2022	24.3	34.8	28.7	80	0.4	0.2	4	30	2
25/4/2022	25	34.4	27.6	84	0	6.2	8	210	2
26/4/2022	23.8	33	28.6	81	16.4	3.5	3	60	1
27/4/2022	24.4	34.6	28.5	83	16.4	1.6	7	310	2
28/4/2022	24.4	34.2	28.5	84	2.5	4	3	250	2
29/4/2022	25.4	35	29.3	82	0	2.2	4	280	2
30/4/2022	25.4	33.6	27.6	86	0	6.2	5	290	2
1/5/2022	24.4	34.5	27	88	14.8	5.7	11	200	1
2/5/2022	22	34.4	27.5	83	13.4	5.8	4	310	2
3/5/2022	24.8	31.8	27.4	87	21.5	3.6	3	60	1
4/5/2022	24.7	33.4	28.2	88	0	0.4	3	50	1
5/5/2022	23.4	35	28.2	86	2.3	3.4	4	60	2
6/5/2022	24.8	35	28.4	83	8.5	5.4	6	30	1
7/5/2022	25.4	34.8	28.5	87	18	2.8	4	80	2

8/5/2022	24.6	33.6	28.4	86	18.8	3.9	4	90	1
9/5/2022	26	31.8	28.3	88	4	2.6	3	320	1
10/5/2022	25.8	34.8	29.6	82	0	0	4	80	1
11/5/2022	24.6	33.6	27.6	88	4.7	7.7	10	330	1
12/5/2022	20.8	34.2	27.5	81	1.9	3.7	9	210	2
13/5/2022	25.2	33.8	28.9	82	18	7.3	3	90	1
14/5/2022	25.2	35.2	29.2	80	18	6.2	4	20	1
15/5/2022	25	33.2	27.9	87	0	8.6	3	60	1
16/5/2022	25	29.2	27.1	91	6.3	2.6	2	40	0
17/5/2022	25.1	33.8	28.6	85	18	0	3	100	1
18/5/2022	23.4	35	29.1	80	0	4.8	4	70	2
19/5/2022	25.2	34.4	27.5	88	0	6.5	4	60	2
20/5/2022	25.2	34.4	29.1	82	0	4.8	4	50	2
21/5/2022	24.8	34.3	29	80	0	6.8	6	50	2
22/5/2022	23.2	34.6	26.9	86	3.8	6.1	10	220	1
23/5/2022	24	33	27.6	88	19.1	7.2	3	50	1
24/5/2022	25	34	29.1	84	3.3	5.2	3	200	1
25/5/2022	25.4	28.2	26	92	0	1.3	3	300	1
26/5/2022	23.2	33.8	27.4	80	3.5	0	4	310	1
27/5/2022	24.2	34.4	28.8	80	0	3.9	3	50	1
28/5/2022	23.6	34.6	28	82	8.6	6	2	50	1
29/5/2022	23.8	34.2	28.6	81	0	5.1	4	360	2
30/5/2022	23	35	28.7	81	18	4.4	5	50	2
31/5/2022	23	35	28	83	18	9.3	6	320	1
1/6/2022	24.6	32.4	28.1	85	5.8	4.8	2	10	1
2/6/2022	22.8	33	27.8	84	0	2.3	6	80	2
3/6/2022	25	36	29	80	0	0.5	3	290	1
4/6/2022	24.8	33.8	28.5	82	2.1	7.1	3	50	1
5/6/2022	23.8	34	27.2	83	2.7	3	2	30	1
6/6/2022	23.8	33.6	26.5	86	0	4	10	300	2
7/6/2022	23.2	34.2	27	88	48.8	4.2	4	330	2
8/6/2022	23.6	32.4	27.4	82	1	3.2	3	100	1
9/6/2022	23.6	34	28.6	80	0	5.8	3	290	1
10/6/2022	24.4	30.6	26.8	85	18	3.4	3	290	1
11/6/2022	24	33.8	27.8	79	18	0	6	280	2
12/6/2022	22.6	32	26.7	85	41.5	5.9	2	40	0
13/6/2022	23.4	33.9	27.8	81	0.5	2.2	4	50	1
14/6/2022	20	33.4	27	87	0	6.9	4	320	1
15/6/2022	19.8	32.8	27.8	86	14.1	5	6	120	2
16/6/2022	23.6	33.4	27.2	84	11	1.9	4	50	1
17/6/2022	23.6	31	26.6	87	18	7.2	3	220	1
18/6/2022	24	33.7	28.3	82	6.5	1.4	2	30	1
19/6/2022	24.2	33.2	27.8	80	18	8.1	3	270	1
20/6/2022	22.8	32.2	26.8	83	0.3	7.5	3	80	1
21/6/2022	24	32.2	27	86	0	1	4	70	1
22/6/2022	23.8	32	27.1	86	49.6	3.5	4	50	1

23/6/2022	23.4	31.8	26.5	88	110.5	4.9	3	60	2
24/6/2022	23.4	31.2	26.3	86	12.7	3.4	3	60	1
25/6/2022	23.4	33.6	27	83	0	2.6	3	100	2
26/6/2022	21.4	31.8	27.9	80	6.5	8.8	2	280	1
27/6/2022	24.8	34.2	27.8	87	0	2.8	5	50	0
28/6/2022	24.6	31.4	26.8	87	2.8	3.2	2	290	0
29/6/2022	24.2	34.6	28.1	80	2.8	0.7	3	30	1
30/6/2022	24.8	35.8	29.2	74	2.8	8.2	6	240	3
1/7/2022	24.9	36.1	29.2	76	0	8.3	4	60	1
2/7/2022	24.6	36.2	30	67	0		9	220	4
3/7/2022	24.4	35.3	30.2	72	0	9.7	6	280	3
4/7/2022	25	33.3	27.7	84	12.1	8.8	3	260	2
5/7/2022	24.8	33.4	27.9	84	1.4	3.4	3	110	1
6/7/2022	25.6	32.8	28.1	86	4.5	3.3	4	100	2
7/7/2022	24	35.1	28.6	80	4	3	4	280	2
8/7/2022	24	37	28.5	80	0	8.8	5	270	2
9/7/2022	23.8	35.6	29.9	71	0	10.2	11	220	5
10/7/2022	26.4	33	28.8	72	0	4.9	4	270	2
11/7/2022	21.4	34.2	28.2	78	0	1.7	4	40	2
12/7/2022	21.2	31.6	27	88	0	2.2	4	70	2
13/7/2022	24.9	33	28.4	82	2.6	0.3	4	140	2
14/7/2022	23.8	34.6	28.7	82	0	3.7	4	50	2
15/7/2022	17.6	34.6	27.6	82	10	4.6	4	190	2
16/7/2022	22	33.8	27.4	84	27.2	7.5	4	90	2
17/7/2022	23.2	35	29.1	81	0	4.2	3	330	1
18/7/2022	23.6	35	28.8	78	0	7.9	4	290	2
19/7/2022	24.8	30.2	27.1	88	0	8.5	3	290	1
20/7/2022	23.8	34.4	28.3	79	0	0.7	7	130	2
21/7/2022	24	33.6	27.7	80	15.1	8.1	4	90	2
22/7/2022	24	29.3	26.2	90	0	6.9	3	60	0
23/7/2022	22.4	32.8	27.7	84	8.6	0	2	300	0
24/7/2022	24.4	32.6	26.8	85	18	4	2	230	1
25/7/2022	24	31	26.3	86	9.5	0.8	4	250	1
26/7/2022	23.5	33.2	27.5	83	1	0.1	2	40	1
27/7/2022	24	33.8	26.7	86	8.3	6.3	3	120	2
28/7/2022	23.8	33.2	28	82	28.8	4.8	4	50	2
29/7/2022	21.4	35.6	28.7	79	16	4.5	10	240	2
30/7/2022	24.6	34.2	28.4	82	1.3	8.2	5	250	2
31/7/2022	25	34.6	29	79	0	5.9	5	230	2
1/8/2022	24.3	32	27.7	86	15.5	9.4	3	300	1
2/8/2022	24.3	32	26.7	89	18	0.7	8	70	3
3/8/2022	23.6	32.4	27.3	84	44.3	0	3	290	1
4/8/2022	23.4	31.2	26.5	86	1.7	2.9	3	70	2
5/8/2022	23.2	33.6	27.9	81	1.4	0	3	60	1
6/8/2022	22.6	33.6	27.8	81	14.5	2.7	4	270	2
7/8/2022	24.2	34.2	28.3	80	2.3	3.2	4	300	1

8/8/2022	23.4	36	28.3	76	2.3	4.2	6	50	3
9/8/2022	23.4	35.4	28.3	79	0	9.7	3	40	1
10/8/2022	22.2	35.4	28.6	80	1	6.4	5	40	3
11/8/2022	22.1	35	28.7	82	1	7.9	4	20	2
12/8/2022	23.8	36	29.6	81	0.9	7	4	50	2
13/8/2022	24.2	32.6	27.3	90	3	5.9	5	60	2
14/8/2022	24	33.9	28.6	78	18	2.2	2	60	1
15/8/2022	24	35	28.3	79	0	4.1	5	20	2
16/8/2022	21.4	34.9	27	83	97.7	7	5	230	2
17/8/2022	21.2	33.4	27.2	85	1	3.7	4	60	2
18/8/2022	21.2	29.6	25.8	88	98.2	6.6	3	80	1
19/8/2022	22.6	33.2	27.2	83	1.2	0.8	5	50	2
20/8/2022	23.2	31.8	26.7	85	7	6.7	3	280	2
21/8/2022	23.4	32.6	27.1	86	28.3	0.7	5	350	2
22/8/2022	23.4	32.8	27.3	83	68.5	2.9	5	300	2
23/8/2022	23.2	33.5	26.6	86	1	6.6	4	80	2
24/8/2022	23	32.8	27	86	2.5	1.3	2	120	1
25/8/2022	23	32	27.7	84	18	6.2	4	150	2
26/8/2022	23.1	33.1	27.9	86	1.8	3	4	280	2
27/8/2022	23.6	31.8	27.4	85	38.6	2	4	310	1
28/8/2022	21.5	30	26.5	84	27.8	1.8	5	60	2
29/8/2022	23.2	33	26.6	88	0	2.6	8	230	2
30/8/2022	21.4	33	27	84	54	1.9	4	60	2
31/8/2022	24	33.2	28	84	1.4	3.5	5	40	2
1/9/2022	24.2	29.8	26.6	89	7	4.6	3	270	2
2/9/2022	23.4	31	26.7	86	3.1	1.4	2	40	0
3/9/2022	23.4	34.8	26.9	85	3.1	1.3	6	190	2
4/9/2022	23	34.2	26.8	84	14	5.2	5	30	2
5/9/2022	23	33.8	28	80	0	6.3	4	40	2
6/9/2022	24.8	33.7	28.1	84	0	2.3	5	50	3
7/9/2022	22.3	33.4	27.1	82	30.1	3.2	6	270	3
8/9/2022	24.4	32.2	28.2	80	30.1	4.7	3	50	2
9/9/2022	24.6	34.2	28.3	78	0	0.5	6	40	2
10/9/2022	24.4	34.6	28.1	81	0	4.6	3	50	1
11/9/2022	22.2	31	26.8	83	3	3.9	3	120	1
12/9/2022	23.8	33.8	27.8	79	3	0	6	30	3
13/9/2022	24	31.8	27.8	82	3	4.8	4	340	2
14/9/2022	23	32	26.7	80	4.1	0	2	60	1
15/9/2022	23.2	31.8	27.1	82	4.1	0	4	10	2
16/9/2022	24	34	28.1	78	4.1	0	4	50	2
17/9/2022	24.3	34	27.2	81	0	5.6	5	230	2
18/9/2022	22.8	31.4	25.9	85	19.8	3.7	3	70	1
19/9/2022	23.4	34.1	27.6	79	0	0.7	5	80	2
20/9/2022	23.6	30.3	26.9	86	0	5.4	3	90	1
21/9/2022	22.1	33.4	27.7	78	0	0	5	190	2
22/9/2022	22.6	32.3	27.1	82	49.2	4.5	4	130	2

23/9/2022	23	33	27.7	80	0	1.9	4	50	2
24/9/2022	23.2	31.2	27.6	83	12.9	5.1	5	70	3
25/9/2022	23.2	32.6	27.2	82	11.7	3.4	8	50	4
26/9/2022	24	33.8	27.1	84	19.3	7	7	60	2
27/9/2022	23.5	34.6	26.5	86	5.8	4.8	9	60	3
28/9/2022	23.4	33.4	27.1	84	33.2	6.2	7	290	2
29/9/2022	24	33.2	28	83	1	5.1	3	20	2
30/9/2022	24	31	27.2	86	28.5	2	6	90	2
1/10/2022	24	31.4	27	86	72.5	1.7	3	70	1
2/10/2022	23.7	31	26.9	87	0.6	1.1	3	50	1
3/10/2022	23.8	30.6	26.8	87	0.3	0	5	40	2
4/10/2022	23.2	33.2	26.1	87	30	0.1	5	40	2
5/10/2022	23.4	31.2	27	85	8.1	3.8	4	90	0
6/10/2022	23.5	33	27.4	86	1.9	1.5	6	50	2
7/10/2022	23.4	33.6	27	87	20	5.2	3	80	1
8/10/2022	24.2	32.4	27.3	86	6.5	1.4	3	40	1
9/10/2022	24.2	31.6	27.4	82	3	3	3	50	2
10/10/2022	24.4	32.7	27.3	83	3	1.1	6	80	2
11/10/2022	23.8	33	27.3	83	0.3	4.2	4	330	2
12/10/2022	23.8	32.6	27.2	82	1	4.7	3	70	2
13/10/2022	23.6	30.8	27.1	86	23.2	4.4	5	30	2
14/10/2022	23.8	32	26.5	85	2	1.2	5	20	2
15/10/2022	21.6	33.6	26.5	86	4.5	1.3	14	210	3
16/10/2022	23.6	33.2	27.4	86	29.4		2	50	1
17/10/2022	24.2	33.8	27.7	82	2.5	1.6	5	40	2
18/10/2022	23.8	34.3	28.2	82	3.3	2.1	4	10	2
19/10/2022	24	34.2	27.8	84	48	2.2	5	30	2
20/10/2022	24.2	34.2	27.8	84	3.5	7	6	350	2
21/10/2022	24.3	34.2	28.5	78	1	6.8	4	290	2
22/10/2022	22.8	29	26.1	90	18	6.3	2	240	0
23/10/2022	22.8	31	27.2	84	0.2	0.8	3	20	1
24/10/2022	23.9	31	27.2	88	1.7	0.3	4	320	2
25/10/2022	24.2	28.8	25.9	92	9.2	0.4	2	220	1
26/10/2022	23.8	28.8	26	91	6.3	0.6	2	290	1
27/10/2022	21.3	31	26.8	86	2.8	0	5	350	1
28/10/2022	23.2	31	26.9	87	1	1.2	2	20	0
29/10/2022	23.8	33.4	26.8	87	27.7	0	6	260	2
30/10/2022	24.4	32.4	26.6	94	18.5	7.4	4	260	1
31/10/2022	23.2	31.6	26.7	86	29.3	4.8	5	290	2
1/11/2022	23.6	32.6	27.3	85	0	1.5	5	350	1
2/11/2022	23.2	31.8	26.8	84	13	2.2	3	320	2
3/11/2022	23.5	32.4	26.4	89	4.1	1.2	6	200	2
4/11/2022	23.6	31.7	26.3	90	14.3	2.5	4	260	1
5/11/2022	23.5	31.8	26.6	89	4.2	1.4	4	340	1
6/11/2022	23.4	31.8	26.7	90	1	2.6	2	290	1
7/11/2022	23.4	32.2	26.9	89	16	3.5	8	230	2

8/11/2022	23.4	32.4	27.1	88	29	4.2	2	320	1
9/11/2022	24.2	31.6	26.4	92	3	2.7	2	280	1
10/11/2022	24	32.8	26.5	92	10.3	1.7	5	260	2
11/11/2022	23.8	29.3	25.9	90	8.8	0.5	4	270	2
12/11/2022	23.7	29.8	26.4	94	7.5	0	3	290	1
13/11/2022	24.4	30.8	26	90	18	0.2	3	280	2
14/11/2022	23.6	31.8	25.8	88	42.5	0.6	9	110	2
15/11/2022	23.7	32.7	26.9	89	65.5	2	4	50	2
16/11/2022	24.4	32.8	27.2	88	18.2	2.3	4	290	2
17/11/2022	23.9	33.4	26.4	89	9.1	5.2	6	220	1
18/11/2022	23.6	33	26.8	88	25.5	4.4	2	350	1
19/11/2022	23.7	31.9	26.9	87	61.1	6.2	4	300	2
20/11/2022	23.6	32.4	26.9	87	3.5	5.4	4	350	2
21/11/2022	23.1	32.6	26.9	87	52.2	4.1	4	280	2
22/11/2022	23.4	34.3	27.5	85	1.2	4.1	4	40	1
23/11/2022	20.6	33	27.9	85	53	6.4	4	50	1
24/11/2022	24.7	32.8	26.6	90	3.7	4.8	8	150	2
25/11/2022	23.1	33.3	27.4	82	72.9	5.1	5	40	2
26/11/2022	24.2	32.8	27.4	87	0.8	5	4	70	2
27/11/2022	23.2	31.4	27.1	88	2	5	3	45	1
28/11/2022	24.3	31.6	27	89	0	0.7	4	310	1
29/11/2022	24	32.3	27.7	82	2	3	4	360	2
30/11/2022	24.3	33.4	27.8	82	1.5	4.6	7	350	2
1/12/2022	24.2	29	25.7	87	0	4.5	4	290	2
2/12/2022	23.6	30.9	26.5	88	20.5	2.8	3	280	1
3/12/2022	24	28.4	25.4	88	0.7	0.3	4	270	2
4/12/2022	22.2	33.4	26.5	84	0.8	0	4	40	2
5/12/2022	24	32.6	26.7	89	2	6	5	50	2
6/12/2022	23.4	32.6	27.1	86	16.4	4.5	4	340	1
7/12/2022	23.8	30.4	27	89	0	3.8	2	270	1
8/12/2022	23	27.6	25.3	92	0	0.4	3	270	2
9/12/2022	22	28.6	24	94	34.5	0	3	250	2
10/12/2022	22.6	25.9	23.8	93	8.2	0	4	260	1
11/12/2022	21.2	25.2	23.5	96	34.2	0	4	260	2
12/12/2022	22.6	29.8	25.8	88	10	0	5	280	2
13/12/2022	23.4	29.2	26.3	91	1	1.1	3	150	1
14/12/2022	23.4	29	25.2	93	15.6	0.6	6	100	2
15/12/2022	22.8	32	26.4	89	42.4	0.7	5	150	2
16/12/2022	22.7	32.6	27.7	85	1.8	3.1	5	80	2
17/12/2022	23.8	33.3	27.4	84	0	4.2	4	50	1
18/12/2022	24	31.6	26.1	88	18	1.3	3	350	2
19/12/2022	23.2	32.8	26.6	88	45.8	3.4	6	50	2
20/12/2022	23.2	32	26.6	87	4.4	4.2	7	230	2
21/12/2022	23.8	31	26	91	18	1.9	3	260	1
22/12/2022	23.7	28	25.3	92	18	0.8	4	260	1
23/12/2022	23.6	32	27.3	82	3.2	0	5	60	2

24/12/2022	23.2	31.9	27.6	78	0	3.7	5	350	2
25/12/2022	23.4	31.7	26.5	87	0.6	5.4	4	290	1
26/12/2022	23.3	31	27.3	86	14.5	2.2	4	10	2
27/12/2022	24.2	26.4	24.5	96	0.3	1.3	3	200	1
28/12/2022	23.2	32	26.7	84	23.6	0	5	300	2
29/12/2022	23.4	30.3	26.2	90	10.5	4	4	20	2
30/12/2022	23.5	30.3	26.5	84	4.2	3.8	3	260	2
31/12/2022	21.8	30.9	25.3	83	26	2.9	5	250	3
1/1/2023	21.7	31	26.5	80	18	2	6	290	2
2/1/2023	23.2	32.7	27	83	0	4.7	5	290	2
3/1/2023	23.4	32.2	26.7	87	0	7.1	4	270	2
4/1/2023	23.2	31.8	26.9	86	23.3	4.5	3	300	2
5/1/2023	23.2	30.7	27	84	0.2	3.4	5	320	2
6/1/2023	23.8	31.2	27	81	18	2.6	5	300	2
7/1/2023	23.6	30.8	26.5	83	0.7	3	4	260	3
8/1/2023	23.8	30.4	26.5	84	18	2.6	5	310	2
9/1/2023	22.4	32.5	26.6	80	0.3	2.1	4	310	2
10/1/2023	24	31.2	27.1	84	0.3	7.5	5	350	2
11/1/2023	23.6	30.2	26.4	88	11.3	1.1	3	120	1
12/1/2023	22.6	32.2	26	89	1.5	0	5	40	2
13/1/2023	23.1	31.6	26.9	85	8.1	5	4	20	1
14/1/2023	23.4	32	26.3	87	8.1	2.2	3	30	1
15/1/2023	23.4	32.6	27.1	80	8	4.3	3	280	2
16/1/2023	23.2	31.2	26.5	85	1	3.2	5	290	3
17/1/2023	24	31.6	26.9	81	18	5	4	290	3
18/1/2023	21.8	31.4	25.9	84	18	3.1	2	280	1
19/1/2023	22	29.4	26.2	85	0	4	4	320	2
20/1/2023	22.8	31	26.6	82	18	0	4	300	2
21/1/2023	23.8	28.4	24.2	96	21	1.5	3	290	2
22/1/2023	23.4	30.2	26	88	21.9	0	6	280	3
23/1/2023	23.4	29.4	26.2	88	18	2.3	5	270	2
24/1/2023	23.4	29.3	24.9	92	3.5	0.2	4	10	2
25/1/2023	22.2	32.8	26.3	86	49.5	0.2	6	70	2
26/1/2023	23.3	31.8	26.9	87	18	4.6	5	20	2
27/1/2023	23	31.6	26.4	87	4	4.1	4	280	1
28/1/2023	23.2	33.4	26.8	85	7.7	2.5	3	250	1
29/1/2023	22.8	33.6	27.7	81	0.5	3.4	5	60	3
30/1/2023	22.4	33.2	27.3	84	10	7.4	4	40	2
31/1/2023	23.2	31.3	27	86	0	5.8	4	90	1
1/2/2023	23.2	32.6	26.4	87	0	2.7	3	280	2
2/2/2023	23.6	33.8	27.6	81	0	5.6	6	10	2
3/2/2023	23.4	33.8	27.5	83	0	4.3	6	320	2
4/2/2023	23.4	32.6	27.3	86	1	8.8	3	50	1
5/2/2023	24.2	30.2	26.2	91	18	1	3	300	1
6/2/2023	23.2	29.8	26	89	24.5	0.5	3	130	1
7/2/2023	23.6	32.6	27.6	83	18	0.5	4	280	2

8/2/2023	24.6	30.2	26.8	88	18	3.7	4	310	2
9/2/2023	24	32.2	27.4	84	18	0.8	4	30	2
10/2/2023	24.2	33.4	27.7	86	11.3	3.5	4	50	1
11/2/2023	24.4	31.2	27.4	82	4.8	3.3	3	280	1
12/2/2023	24.3	32.8	27.6	84	0	0.8	4	50	2
13/2/2023	23.6	32.8	26.8	84	2.1	6.2	6	300	1
14/2/2023	23	31	26.5	85	32.8	5.1	6	300	3
15/2/2023	23.2	32	27.7	82	0	3	5	270	2
16/2/2023	22.8	34.2	27.3	80	0.4	3.4	6	50	2
17/2/2023	23	34.2	28.2	80	0.4	8.5	4	80	2
18/2/2023	23.4	33	27.6	82	2.5	8.2	6	30	2
19/2/2023	24.6	30	26.1	89	2.5	7.4	5	280	2
20/2/2023	23.2	33	27.2	82	4	3.3	4	340	2
21/2/2023	23.8	32.2	27.8	82	0	6.3	6	40	2
22/2/2023	24.6	32.6	27.3	86	0	6.7	3	60	2
23/2/2023	24.4	33.4	27.9	83	1.6	4	6	30	2
24/2/2023	23.8	32	27.1	84	35	4.4	3	250	1
25/2/2023	23.4	32.8	26.6	83	18	4.1	2	250	1
26/2/2023	22	32.4	26	86	18	6	3	170	2
27/2/2023	23.3	31	25.9	85	18	5	3	230	1
28/2/2023	23.2	31.8	25.9	88	8.4	2.7	5	50	2
1/3/2023	23.2	27.2	24.5	95	34.8	3.1	3	270	1
2/3/2023	23.1	31.9	26.3	86	16.2	0	4	160	1
3/3/2023	22.6	32.8	26.3	85	13	3.2	5	260	2
4/3/2023	22.6	32.2	26.9	82	0.5	3.6	4	250	2
5/3/2023	22.6	28.8	25.6	90	0.5	8	3	245	1
6/3/2023	22.6	32.6	26.8	83	9.2	0	5	300	2
7/3/2023	23.2	32.6	27.8	78	0	9.5	5	280	2
8/3/2023	24.4	31.7	27.8	82	0	6.7	5	60	2
9/3/2023	24.6	32.2	27.4	85	1.2	5.3	6	40	2
10/3/2023	23.6	33.6	26.4	86	0.5	1.9	5	40	2
11/3/2023	23	32.4	26.6	86	38.5	6.2	6	50	2
12/3/2023	23	33.8	27.3	85	20.6	5	5	30	2
13/3/2023	24.6	32.4	27.5	86	2	7.6	8	10	3
14/3/2023	24.2	32.4	27.3	80	18	3.8	4	250	2
15/3/2023	24.2	33.4	27.5	86	18	0.6	3	320	1
16/3/2023	24.4	32.4	27.4	87	17	2.4	4	30	1
17/3/2023	23	34.6	27.5	84	17	2.9	4	60	2
18/3/2023	23.2	34.8	28.3	81	1.7	7.4	4	110	3
19/3/2023	24.8	34.8	29	82	4	11	3	140	1
20/3/2023	25.6	32.4	26.9	88	0	4.2	7	310	2
21/3/2023	24.1	31.2	26.7	90	50.7	3.3	4	50	1
22/3/2023	23.6	35.2	28.1	79	0.5	2	7	70	2
23/3/2023	23.8	34.2	28.1	82	0	7.8	5	290	2
24/3/2023	23.6	32.6	26.9	89	19.6	5.9	6	20	2
25/3/2023	23.6	34.2	27.1	86	0.2	2.5	6	140	2

26/3/2023	24	33.2	27.8	86	2.6	5.2	6	40	2
27/3/2023	24.1	32.7	27.9	85	0.2	5.6	7	30	3
28/3/2023	25	31	26.9	89	0	3.7	5	250	1
29/3/2023	23.4	34.2	28	84	1	2.4	3	20	1
30/3/2023	23.3	34.4	28.5	81	1	8.3	4	310	2
31/3/2023	24.2	33.4	28.9	81	0	8.4	3	40	1
1/4/2023	24	32.6	27.3	82	0	4.4	3	280	2
2/4/2023	24.6	32.6	26.5	90	0	4	2	270	1
3/4/2023	23.6	28.4	25.3	93	47	4.9	3	230	1
4/4/2023	23.2	34.3	27.7	84	1.8	0	5	350	2
5/4/2023	24.6	32	26.7	87	1.8	7.3	7	280	2
6/4/2023	23.4	35.2	27.9	80	0	1	2	270	1
7/4/2023	23.3	34.8	28.3	80	0	6.7	6	30	2
8/4/2023	23.6	33.6	28	83	8.3	6.8	4	250	1
9/4/2023	24	31	25.7	92	0	6.2	2	220	1
10/4/2023	22.2	34.7	28	79	28.2	3.5	4	40	1
11/4/2023	23.6	33.4	28	84	0	9.4	3	50	1
12/4/2023	23.6	35.4	28.5	79	2.5	5.2	3	310	2
13/4/2023	25.7	33.6	29.2	84	0	9.9	5	150	2
14/4/2023	24.8	35.4	29.3	78	0.5	5.1	3	50	1
15/4/2023	24.6	36.5	30.1	78	0	8.6	5	320	2
16/4/2023	25.2	35.4	29.6	80	0.5	8.6	6	10	3
17/4/2023	24.8	35.9	29.1	81	0	7.6	5	30	2
18/4/2023	26.2	34.4	29.6	83	0	7.8	5	30	2
19/4/2023	25.2	35.3	28.1	86	26	5.7	4	20	1
20/4/2023	24	31.2	25.8	92	31.4	6.3	2	290	1
21/4/2023	23.6	33.4	27.9	86	5.8	0.3	3	50	1
22/4/2023	24.2	34.2	28.4	84	5.8	0.3	3	105	2
23/4/2023	25.2	35	28	85	5.8	5.1	6	260	2
24/4/2023	24.4	35	27.4	87	2.5	5	4	250	2
25/4/2023	23.6	34.3	28	84	15	4.3	4	270	1
26/4/2023	25.2	33.2	28.3	86	15	5.1	2	240	1
27/4/2023	24.7	32.6	27.6	88	8.5	4	3	280	1
28/4/2023	22.1	33.6	27	86	9.6	0.7	7	340	2
29/4/2023	24.8	34.8	28.2	84	9.6	4.9	5	280	2
30/4/2023	25.4	32.4	28	88	9	5.8	6	335	2
1/5/2023	24.6	35.6	29	82	18	3.8	6	110	3
2/5/2023	23	35.4	28.2	86	18	8	5	90	2
3/5/2023	25	33.1	28.5	86	0.7	4.8	2	120	1
4/5/2023	25.4	34.4	28.2	85	0	3.4	5	40	1
5/5/2023	24.6	34	28.9	81	0	4.5	3	360	1
6/5/2023	25.2	32.8	27.9	89	32.5	1.8	4	230	2
7/5/2023	25.4	35.4	29.2	83	2	3.1	5	60	1
8/5/2023	25.4	34.4	29.4	83	1.5	5.2	8	240	2
9/5/2023	26.4	35.4	30.3	74	0	5.7	6	230	3
10/5/2023	25.2	34.3	29	80	0	3.2	4	50	2

11/5/2023	25.1	35.6	29.5	77	0	4.4	4	80	2
12/5/2023	25	35.6	32.2	70	0	6.6	4	90	1
13/5/2023	25.2	36.2	30.7	81	0	6.5	4	90	2
14/5/2023	24	35.2	28.6	79	0	7.6	3	100	1
15/5/2023	24.4	32	28	86	0	4.2	4	280	2
16/5/2023	25.2	36.2	29.3	84	0	1.7	3	60	1
17/5/2023	24.8	35.4	28.5	83	2	8	5	150	2
18/5/2023	23.6	33	27.7	85	16.7	7	5	280	2
19/5/2023	23.6	32.6	29	81	5.1	1.7	2	20	1
20/5/2023	23.8	35.8	32	72	10.5	1.5	2	40	1
21/5/2023	24	33.6	29.8	79	33.5	9.3	3	288	1
22/5/2023	23.9	32.3	29.7	80	33.5	2	4	10	2
23/5/2023	23.4	33.7	30.5	74	12.3	1	4	290	1
24/5/2023	24.5	33.4	30.2	80	12.3	5.4	4	280	2
25/5/2023	24.2	35.4	27.7	88	5.5	2	4	60	1
26/5/2023	24.2	33.4	30.3	75	3.8	6.4	6	40	2
27/5/2023	24	34.6	28.4	82	18.9	3.1	4	60	1
28/5/2023	25.2	34.8	28.2	86	11.1	8.8	5	50	2
29/5/2023	23.8	36.2	32.1	68	3.5	5.9	4	150	1
30/5/2023	23.8	35.9	31.3	69	3.5	8.3	8	180	3
31/5/2023	24.3	36	29.1	80	15.5	8.6	2	270	1
1/6/2023	25.4	34.8	31.7	72	14	9.4	3	70	1
2/6/2023	25.6	33.2	30.5	78	0	6.8	3	50	1
3/6/2023	25.6	33.4	31	77	0	1.3	3	40	2
4/6/2023	24.6	33.3	28	88	14.5	1.8	5	60	1
5/6/2023	25	35	28.2	85	0	2.6	7	260	1
6/6/2023	24	35	28.1	86	36.1	7.9	4	340	1
7/6/2023	24.1	33.1	30.7	76	49.5	2	3	30	0
8/6/2023	24.8	34	28.3	84	49.5	3.7	3	150	2
9/6/2023	25	34.8	31.1	75	0	5.4	3	100	1
10/6/2023	25	34.4	29.3	82	48.1	3.7	4	90	2
11/6/2023	25.4	34.9	27.7	85	25.3	5.8	8	230	2
12/6/2023	24	35	27.1	87	25.1	4.3	16	190	2
13/6/2023	23.8	36.4	28.8	82	20.5	7.4	5	40	2
14/6/2023	25.6	33.4	28.6	84	4	9.9	3	60	1
15/6/2023	24.8	35.9	32.2	68	29.2	2.3	3	70	2
16/6/2023	25	35.4	29	78	0	8	5	50	1
17/6/2023	25	36.4	29.9	76	4	3.4	3	250	1
18/6/2023	25.4	33.4	28.6	81	0	9.6	4	50	1
19/6/2023	25	34.8	29	81	4.5	2	5	50	2
20/6/2023	24.9	35.3	31.9	69	0	9	4	70	2
21/6/2023	25.7	34	28.9	82	0	8.6	3	70	1
22/6/2023	25.4	32.6	28.4	86	0	2.2	3	110	1
23/6/2023	22.3	31.8	26.1	92	17.2	1.3	8	60	1
24/6/2023	24.7	33.3	30.1	76	50.2	0.4	3	220	1
25/6/2023	24.6	33.6	28	84	2	5	4	30	1

26/6/2023	24.8	32.2	28	85	0	3.5	3	60	1
27/6/2023	24.8	33.7	27.7	86	0	1.6	6	40	1
28/6/2023	24.7	33.6	28	84	0	3.2	4	60	1
29/6/2023	24.4	34	27.9	86	21.8	3.6	6	335	2
30/6/2023	24.3	33.9	27.2	89	1	5.3	3	80	1
1/7/2023	23.9	33.2	27.5	85	106	4.6	2	240	1
2/7/2023	25.2	34.8	28.6	83	105	3.6	3	50	2
3/7/2023	24.6	35	28.5	84	1	6.4	3	70	2
4/7/2023	24.8	33.4	28.9	84	2	7.5	3	10	2
5/7/2023	25.4	33.2	28.6	80	0	4.8	3	264	1
6/7/2023	25.6	34.4	28.9	81	0	3	3	50	1
7/7/2023	25.3	33.8	29.4	81	0	2.9	3	120	1
8/7/2023	24.8	33.8	27.6	84	2	4.9	5	40	2
9/7/2023	24.8	33.4	27.9	87	0.8	2.7	3	60	1
10/7/2023	24.7	32.6	27.2	88	0.4	3.6	4	330	1
11/7/2023	24.2	31.4	26.7	90	55.5	2.2	6	100	1
12/7/2023	24.2	31.3	27.3	87	46.3	0.9	3	60	2
13/7/2023	24.2	33.6	27.5	86	0.4	0.7	4	150	1
14/7/2023	24.4	28.8	25.9	91	10	6.3	3	150	1
15/7/2023	21.6	35.6	27.2	80	0.8	0	3	290	1
16/7/2023	21.8	34.4	28.6	77	0.9	10	5	60	2
17/7/2023	23.8	33.9	28.5	84	0	8.3	3	60	1
18/7/2023	25	33.2	28.2	88	18	3.9	3	60	1
19/7/2023	24.3	30.4	27.1	90	38	1.6	1	190	0
20/7/2023	24.4	32.8	28.2	86	18	0	7	70	2
21/7/2023	24.8	35	29	82	0.4	0.8	3	190	1
22/7/2023	24.6	35.2	27.5	90	4	6.9	2	290	1
23/7/2023	24.4	35.2	28.4	84	0	7	2	100	1
24/7/2023	25	35.4	29.3	81	0	3.8	5	50	2
25/7/2023	24.9	36	29.3	80	0	9.4	6	260	2
26/7/2023	25.8	35.6	29.6	83	0	9	2	210	1
27/7/2023	25.4	35.3	28.9	85	3.9	6.2	4	60	2
28/7/2023	24.9	35.4	29	82	2	5.1	5	50	2
29/7/2023	25	35.8	29.3	80	0	7.3	4	60	1
30/7/2023	25.4	35	28	87	4.5	7.3	3	170	1
31/7/2023	25.4	33	28.2	86	8.8	6	5	110	1
1/8/2023	25	31.7	28.1	88	8.8	2.9	3	200	1
2/8/2023	26	34.6	29.7	84	18	0.5	6	180	2
3/8/2023	25	34.5	28.9	82	1.2	6.7	6	40	2
4/8/2023	24.7	34.5	28	87	28.6	5.1	2	270	1
5/8/2023	25.6	34	27.1	91	47	5	5	280	1
6/8/2023	24.8	34.4	27.8	88	36	2.3	4	280	2
7/8/2023	23	33.4	27.1	86	54.7	5.2	4	240	2
8/8/2023	23.5	34	27.3	88	0	5.3	2	180	1
9/8/2023	24	32.4	27.1	87	47	3.6	3	80	2
10/8/2023	24.2	32	26.4	92	8.4	4.5	5	60	2

11/8/2023	24.5	33.3	26.9	86	4.5	1.6	5	50	2
12/8/2023	23.7	34.6	27.8	85	11	4.3	3	210	2
13/8/2023	23.2	32.4	27.8	84	41.5	5.5	3	40	1
14/8/2023	24.1	29	25.8	94	74.2	1.3	2	110	1
15/8/2023	24.1	33.8	27.9	86	0.8	0	6	40	3
16/8/2023	23.8	32.8	27.6	86	43.4	4.7	4	40	1
17/8/2023	24.2	31.8	27.2	88	3	2.8	3	90	1
18/8/2023	24.4	32.7	27.6	85	2	1.1	2	50	1
19/8/2023	24	34	27.5	85	19	1.8	3	270	2
20/8/2023	24	33.6	27.4	84	0.2	4.7	3	250	1
21/8/2023	24	32.8	27.6	86	0.2	2.8	8	140	2
22/8/2023	24.4	34	27	89	7.7	1.4	4	50	2
23/8/2023	24.4	33.8	27	88	13.3	3.4	7	40	3
24/8/2023	23.2	29.2	25.9	91	13.1	5	3	10	0
25/8/2023	23.6	33	27.8	84	0.2	0	3	90	1
26/8/2023	23.6	32.2	27.2	88	8	4.8	4	240	1
27/8/2023	23.7	33.7	28	84	3	2	5	10	1
28/8/2023	24.2	33.6	27.3	88	6.3	7.1	6	170	2
29/8/2023	24.2	34.4	27.7	86	71	3.7	5	40	2
30/8/2023	24.3	32.6	27.1	90	3	5.1	3	210	1
31/8/2023	24.1	34.2	27.1	86	38	2.6	4	50	2
1/9/2023	24	34.2	26.7	89	20.5	4	5	50	1
2/9/2023	24.6	33.6	28.2	83	20.5	2.5	7	200	2
3/9/2023	24.2	34.2	27	89	56.7	2.8	4	170	1
4/9/2023	24	29.4	26.5	90	116.8	3.3	2	280	1
5/9/2023	24.2	30.4	26	93	1	0	2	220	1
6/9/2023	24	31.4	26.9	88	11.6	0.2	2	100	1
7/9/2023	24.1	34.7	28.7	80	10	0.2	5	30	2
8/9/2023	25.5	34.4	29.2	80	11.6	6.2	4	10	1
9/9/2023	24.3	34.4	28.3	80	9	5.6	4	260	2
10/9/2023	24.2	32.6	27.8	85	0	3.9	4	50	2
11/9/2023	24.7	35	28.7	85	0	1.5	4	70	2
12/9/2023	24.8	34.4	28.3	83	0.9	5.2	5	170	2
13/9/2023	23.9	35.4	27.5	86	7.4	2.7	6	140	2
14/9/2023	24	33	27.5	88	28.9	6	11	220	2
15/9/2023	23.5	32.3	27.6	89	72.5	4.3	5	40	2
16/9/2023	23.7	33.4	27.6	86	75	1.7	3	120	2
17/9/2023	23.7	32.6	28	86	18	3.8	3	115	1
18/9/2023	24.4	33	26.7	90	9	1.7	6	320	1
19/9/2023	23.9	34	27.6	86	64.8	2.1	8	60	1
20/9/2023	24.6	29	26.4	93	6.8	4	4	290	1
21/9/2023	23.9	29.6	25.5	94	4.6	0.4	3	320	1
22/9/2023	24	33.2	26.5	89	5	0	4	280	1
23/9/2023	24.2	33.1	27.2	88	1.5	0.4	5	160	1
24/9/2023	24.6	33	27.6	87	18	4.3	6	320	2
25/9/2023	24	32.6	27	88	0.2	3.5	6	270	2

26/9/2023	23.5	34.2	28.2	82	69	0	6	60	2
27/9/2023	24	35.8	28.6	83	0	4.5	4	30	2
28/9/2023	24.6	35.4	29.5	81	0	7.5	3	90	2
29/9/2023	24.8	34.3	27.9	85	0	6.9	8	30	2
30/9/2023	24	34.4	27.9	85	45.8	2.4	3	50	1
1/10/2023	24.1	35.1	27.7	84	0	4.6	4	50	2
2/10/2023	24.2	34	28	86	1.2	6.6	6	300	2
3/10/2023	24.4	35	27.7	86	7.8	4.8	4	270	2
4/10/2023	24.6	33	27.9	88	29.8	5	3	130	1
5/10/2023	24	33.6	27.7	86	36.6	4	5	60	2
6/10/2023	24.4	34.5	27.1	88	18	2.5	5	50	2
7/10/2023	24.6	34.6	26.4	92	7.3	6	2	110	0
8/10/2023	24.6	32.8	28.1	88	2	1.1	4	280	1
9/10/2023	24.6	32.7	28	84	2	4.2	6	170	2
10/10/2023	25.6	32.3	28.4	86	3	4.8	4	40	2
11/10/2023	24.7	33.2	27.7	88	33.1	0.4	4	50	1
12/10/2023	24.2	30.3	26.9	89	8	2.3	4	290	2
13/10/2023	24.8	32.8	28.2	86	18	2.1	4	280	1
14/10/2023	25	32.8	27.9	86	18	3.9	6	200	2
15/10/2023	24.3	32.4	27.8	87	19.3	1.9	5	350	1
16/10/2023	24.4	32.5	27.4	86	19.3	4.3	4	170	2
17/10/2023	24	30.6	26.9	88	12.6	0.2	3	60	1
18/10/2023	25	33.3	28.5	86	18	0.4	5	50	2
19/10/2023	24.2	31.4	27.3	88	23.3	5.5	4	40	2
20/10/2023	24.2	30.8	27.4	87	1.6	2.4	2	290	1
21/10/2023	25	32	27.3	88	18	0.8	1	60	1
22/10/2023	24.6	30	26.9	87	1.5	0.3	5	20	1
23/10/2023	24	30	26.9	87	1.5	0.3	5	20	1
24/10/2023	24	29.7	26.1	91	37	0.8	3	330	1
25/10/2023	23.8	32.8	27.5	85	18	1.4	4	290	2
26/10/2023	24.6	33.9	27.2	83	0	5.6	6	50	2
27/10/2023	24.5	33.4	27.7	82	1.4	5.2	4	10	2
28/10/2023	24	31.6	27	85	1.4	4.8	3	280	1
29/10/2023	24.1	29.3	25.6	94	1.5	3.1	3	130	1
30/10/2023	23.4	32.8	29	79	31	0.8	3	280	1
31/10/2023	24.7	33	30.5	73	12.5	5.6	2	310	1
1/11/2023	23.8	32	28.9	79	7	5.5	4	270	2
2/11/2023	24.2	30.6	28.2	82	2.5	1.8	5	300	3
3/11/2023	23.4	34	30.1	76	2.5	0.9	7	330	2
4/11/2023	24.9	31	28.5	82	0.5	6.2	5	320	1
5/11/2023	25	32.4	28.6	82	1.2	0	2	270	1
6/11/2023	24	34.1	29.8	76	0.5	4.2	4	270	2
7/11/2023	24.8	31	28.1	83	0	4.6	4	290	1
8/11/2023	24.6	32.2	29.3	78	18	1.7	5	300	2
9/11/2023	23.6	32.6	29.1	77	5	1.3	4	300	2
10/11/2023	24.1	32.9	30	74	1	2.9	3	260	1

11/11/2023	24.2	34.4	30.3	73	1	4.8	6	320	2
12/11/2023	24.1	33.6	30.2	76	4	6.4	5	290	2
13/11/2023	24.8	33.5	30.4	72	0	8.1	6	50	1
14/11/2023	24.1	30.8	28	85	28.8	4.7	3	100	1
15/11/2023	24.4	31.2	28	82	31.5	0.4	4	300	2
16/11/2023	24.4	32.7	29.8	76	0.2	3.5	5	310	2
17/11/2023	25.2	32.6	29.4	78	0.2	3.4	3	270	1
18/11/2023	25	32.1	29.3	80	18	3.7	4	200	2
19/11/2023	24.8	29.2	26.6	88	6.2	2.5	2	110	0
20/11/2023	23.5	33.4	29.2	81	20.3	0	4	300	2
21/11/2023	24	30.6	27.9	82	12.5	2.7	4	270	2
22/11/2023	24	29.4	26.6	87	5.8	1	3	250	2
23/11/2023	24.3	28	26	89	21.6	0	5	270	2
24/11/2023	24.2	28.6	26.8	86	10	0	3	270	1
25/11/2023	23.8	33.6	29.9	74	10	0	3	280	2
26/11/2023	24.6	31.3	28.5	80	0	5.6	6	310	4
27/11/2023	24.4	32.8	29.7	73	0	2	6	360	3
28/11/2023	23.6	34.4	29.6	75	28.1	7.2	3	50	1
29/11/2023	24	32.4	28.6	82	0.9	5.6	4	50	1
30/11/2023	24	33.2	29.5	81	1.8	2.4	4	300	2
1/12/2023	25.2	32.3	28.9	80	0.2	2.3	6	295	2
2/12/2023	23.4	28.4	26.3	90	2.9	2.4	2	220	1
3/12/2023	24	32.4	27.7	83	3.8	0	4	160	1
4/12/2023	24	32.8	29.5	77	8.2	2.5	5	20	2
5/12/2023	24	32.8	29.3	79	18	5.1	3	260	1
6/12/2023	22.9	33	29	78	41.4	1.6	2	310	0
7/12/2023	23.6	29.4	27.6	88	18	2.5	2	340	0
8/12/2023	23.9	31.4	29.1	79	0.3	0.2	4	300	2
9/12/2023	24.2	31.7	29.8	77	0.3	1.7	3	280	1
10/12/2023	25	30.8	29.1	83	0.3	2.1	2	330	1
11/12/2023	23.9	32.6	29.6	77	0	0.6	4	270	1
12/12/2023	24	32.6	29.2	76	0	3	4	330	2
13/12/2023	24.3	32.3	28	83	0	3.3	5	10	2
14/12/2023	24.2	32	28.8	80	9.4	1	6	290	2
15/12/2023	25	32.2	29.3	78	18	2.6	6	290	2
16/12/2023	25.2	32.4	29.7	80	18	3.8	4	340	2
17/12/2023	24.6	32.3	28.5	84	0.7	4.9	5	320	2
18/12/2023	25.4	29.3	27.1	89	50.1	2.2	2	190	1
19/12/2023	23.6	31	28.5	82	56.5	0	4	270	2
20/12/2023	24	29.4	27.8	85	0	1.1	4	300	2
21/12/2023	24.5	30.9	28.7	81	2.8	1	5	310	2
22/12/2023	24.2	28.8	26.4	89	0	1.5	5	280	2
23/12/2023	24.4	29.8	27.5	87	3.5	0.6	3	260	1
24/12/2023	24.2	27.2	25.4	95	23	0	3	280	2
25/12/2023	24	29.6	26.7	89	61.7	0	5	190	1
26/12/2023	23.6	32	28.4	80	20.5	0.4	4	250	2

27/12/2023	23.8	31.1	28.4	84	0	2.1	4	300	1
28/12/2023	23.8	30.4	28.6	84	3	2.7	4	300	2
29/12/2023	23.7	29.2	27	91	9.6	2.6	5	290	1
30/12/2023	24.4	31.1	27.7	87	9.2	0.6	4	350	1
31/12/2023	24.6	32.6	29.3	80	19.7	1	5	290	2

Lampiran 2 Data Hasil *True Value*, *Predict Value*, *Error* dan *Accuracy Percentage* menggunakan Algoritma XGBOOST

True Value	***Predicted Value***	***Error***	***Accuracy Percentage***
0.042969	0.343434	0.300465	699.2645
0.216406	0.058532	0.157874	72.95268
0.370313	0.163136	0.207177	55.94647
0.00625	0.145447	0.139197	2227.145
0.5	-0.01343	0.513433	102.6865
0.019531	0.080522	0.060991	312.275
0.007813	0.124621	0.116809	1495.151
0.014063	0.081176	0.067114	477.253
0.0625	0.148892	0.086392	138.2264
0.5	0.163825	0.336175	67.23507
0.5	0.063822	0.436178	87.23567
0.5	0.103205	0.396795	79.35892
0.5	0.007927	0.492073	98.41462
0.140625	0.00881	0.131815	93.73485
0.027344	0.198405	0.171061	625.5956
0.767188	0.075976	0.691212	90.09682
0.5	0.126706	0.373294	74.6588
0.269531	0.166115	0.103416	38.36896
0.140625	0.156025	0.0154	10.95084
0.5	0.087035	0.412965	82.59306
0.392969	0.085315	0.307654	78.28971
0.070313	0.144251	0.073939	105.1576
0.088281	0.123124	0.034843	39.46791
0.086719	0.033058	0.053661	61.87888
0.039063	0.086999	0.047936	122.7166
0.096094	0.0608	0.035294	36.72871
0.024219	0.227656	0.203437	839.997
0.0625	0.32601	0.26351	421.6156
0.391406	0.14085	0.250556	64.01443
0.007813	0.088893	0.08108	1037.828
0.011719	0.054711	0.042992	366.8658
0.5	0.009968	0.490032	98.00638
0.009375	0.239517	0.230142	2454.844
0.332031	0.142046	0.189985	57.21899
0.5	0.025326	0.474674	94.93481
0.011719	0.200422	0.188703	1610.267
0.5	0.118113	0.381887	76.37738
0.039063	0.08613	0.047067	120.4927
0.407813	-0.00048	0.408289	100.1168
0.075	0.024026	0.050974	67.96561
0.9125	0.171143	0.741357	81.24462
0.5	0.076725	0.423275	84.65502
0.023438	0.10796	0.084523	360.6295
0.125	0.072491	0.052509	42.00748
0.035156	0.009043	0.026113	74.27767
0.367188	0.263646	0.103541	28.19841
0.030469	0.121853	0.091384	299.9266
0.301563	0.065325	0.236237	78.33768
0.5	0.04843	0.45157	90.31402
0.160938	0.231285	0.070348	43.71128
0.5	0.043152	0.456848	91.36965
0.5	0.063055	0.436945	87.38896
0.007813	0.028825	0.021013	268.961
0.03125	0.255109	0.223859	716.3489
0.035938	0.035845	9.20E-05	0.256022
0.5	0.002327	0.497673	99.53453
0.019531	0.135125	0.115593	591.8379
0.007813	0.005096	0.002717	34.77349
0.585938	0.114998	0.470939	80.37363
0.128125	0.098701	0.029424	22.96479
0.002344	0.133584	0.131241	5599.601

0.017969	0.049195	0.031226	173.779
0.140625	0.043327	0.097298	69.18966
0.140625	0.100397	0.040228	28.60632
0.053125	0.0185	0.034625	65.17627
0.004688	0.142303	0.137616	2935.806
0.321875	0.174684	0.147192	45.7294
0.021875	0.048073	0.026198	119.7634
0.011719	0.093303	0.081584	696.1875
0.144531	0.003744	0.140787	97.4095
0.060938	0.106859	0.045922	75.35869
0.5	0.089352	0.410648	82.12952
0.5	0.035465	0.464535	92.90694
0.5	-0.00028	0.500281	100.0561
0.001563	0.042719	0.041157	2634.046
0.5	0.004834	0.495166	99.03311
0.5	0.006329	0.493671	98.73414
0.016406	0.22989	0.213484	1301.234
0.075	0.081795	0.006795	9.060327
0.246094	0.075682	0.170412	69.24664
0.142188	0.069581	0.072607	51.06407
0.5	-0.00857	0.508574	101.7149
0.5	0.039883	0.460117	92.02333
0.117969	0.115487	0.002482	2.104024
0.090625	0.069923	0.020702	22.84309
0.5	0.024263	0.475737	95.14747
0.5	0.035434	0.464566	92.91315
0.015625	0.020535	0.00491	31.42607
0.5	0.173325	0.326675	65.335
0.5	0.027605	0.472395	94.47895
0.078125	0.857933	0.779808	998.1544
0.5	0.087726	0.412274	82.45489
0.140625	0.010837	0.129788	92.29379
0.045313	0.076291	0.030979	68.36711

0.078125	0.209709	0.131584	168.4277
0.130469	0.156658	0.026189	20.07334
0.160156	0.223054	0.062897	39.27248
0.164063	0.156988	0.007075	4.312206
0.045313	0.07649	0.031177	68.80451
0.071094	0.136792	0.065698	92.41088
0.082031	0.072196	0.009835	11.98904
0.001563	0.191884	0.190322	12180.58
0.267188	0.092459	0.174728	65.39532
0.414063	0.010287	0.403775	97.51557
0.16875	0.05362	0.11513	68.22494
0.5	-0.0286	0.528601	105.7201
0.073438	0.073086	0.000351	0.47813
0.003906	-0.00402	0.007931	203.0355
0.140625	0.029064	0.111561	79.33192
0.5	0.153843	0.346157	69.23144
0.140625	0.057693	0.082932	58.97383
0.003906	0.238809	0.234903	6013.51
0.5	0.146535	0.353465	70.69303
0.023438	0.063809	0.040371	172.2513
0.027344	0.141173	0.11383	416.2914
0.5	0.026312	0.473688	94.7376
0.010938	0.011754	0.000816	7.463385
0.097656	0.118037	0.020381	20.87009
0.140625	0.185191	0.044566	31.69172
0.1875	0.031336	0.156164	83.28733
0.027344	0.023117	0.004227	15.45951
0.002344	0.028748	0.026404	1126.588
0.025	0.065591	0.040591	162.3622
0.036719	0.198484	0.161766	440.5534
0.007031	0.139914	0.132883	1889.887
0.050781	0.016762	0.03402	66.99225
0.009375	0.008401	0.000974	10.38445

0.020313	0.068125	0.047813	235.3847
0.5	0.054656	0.445344	89.06883
0.140625	0.217952	0.077327	54.98785
0.219531	0.011132	0.208399	94.92907
0.001563	0.07356	0.071998	4607.848
0.140625	0.083466	0.057159	40.64609
0.5	0.079469	0.420531	84.10618
0.5	0.027367	0.472633	94.52659
0.5	0.041403	0.458597	91.71933
0.074219	0.093089	0.01887	25.42544
0.167969	0.042765	0.125203	74.5397
0.054688	0.062324	0.007637	13.96463
0.002344	0.114136	0.111792	4769.801
0.144531	0.028338	0.116194	80.39351
0.359375	0.084647	0.274728	76.44594
0.003906	0.040535	0.036629	937.69

0.5	0.10734	0.39266	78.53198
0.820313	0.182555	0.637758	77.74573
0.002344	0.06748	0.065136	2779.152

Lampiran 3 Data Hasil *True Value, Predict Value, Error* dan *Accuracy Percentage* menggunakan Algoritma ELM

True Value	***Predicted Value***	***Error***	***Accuracy Percentage***
0.042969	0.113473	0.070504	164.0819
0.216406	0.100171	0.116235	53.71161
0.370313	0.128869	0.241443	65.19992
0.00625	0.134604	0.128354	2053.669
0.5	0.082227	0.417773	83.55467
0.019531	0.030848	0.011317	57.94159
0.007813	0.108473	0.100661	1288.454
0.014063	0.131712	0.117649	836.6167
0.0625	0.099573	0.037073	59.31644
0.5	0.097433	0.402567	80.51341
0.5	0.0589	0.4411	88.21992
0.5	0.084596	0.415404	83.08084
0.5	0.025295	0.474705	94.94095
0.140625	0.077868	0.062757	44.62713
0.027344	0.092214	0.064871	237.2415
0.767188	0.175215	0.591973	77.16142
0.5	0.108646	0.391354	78.27077
0.269531	0.203033	0.066498	24.67167
0.140625	0.070721	0.069904	49.70983
0.5	0.118406	0.381594	76.3189
0.392969	0.092119	0.30085	76.5583
0.070313	0.057341	0.012972	18.44847
0.088281	0.101826	0.013545	15.3426
0.086719	0.054069	0.03265	37.65066
0.039063	0.049421	0.010359	26.51886
0.096094	0.045012	0.051082	53.15868
0.024219	0.143146	0.118927	491.0553
0.0625	0.109249	0.046749	74.79897
0.391406	0.092256	0.299151	76.42968
0.007813	0.168619	0.160807	2058.327
0.011719	0.111036	0.099317	847.5067
0.5	0.05299	0.44701	89.40198
0.009375	0.073025	0.06365	678.93
0.332031	0.121406	0.210625	63.43531
0.5	0.090344	0.409656	81.93123
0.011719	0.14595	0.134231	1145.438
0.5	0.060018	0.439982	87.99634
0.039063	0.113423	0.074361	190.3636
0.407813	0.113751	0.294061	72.10698
0.075	0.055984	0.019016	25.35456
0.9125	0.118603	0.793897	87.00236
0.5	0.070264	0.429736	85.9472
0.023438	0.112773	0.089335	381.164
0.125	0.115447	0.009553	7.642174
0.035156	0.07189	0.036734	104.4869
0.367188	0.140328	0.226859	61.783
0.030469	0.061798	0.031329	102.8247
0.301563	0.094036	0.207527	68.81716
0.5	0.017348	0.482652	96.53039
0.160938	0.117038	0.0439	27.27761
0.5	0.00303	0.49697	99.39402
0.5	0.099928	0.400072	80.01438
0.007813	0.099894	0.092082	1178.648
0.03125	0.085886	0.054636	174.8341
0.035938	0.097366	0.061428	170.9307
0.5	0.03937	0.46063	92.12596
0.019531	0.142346	0.122815	628.8116
0.007813	0.007809	3.34E-06	0.042725
0.585938	0.117777	0.46816	79.89931
0.128125	0.053382	0.074743	58.3363
0.002344	0.093202	0.090858	3876.601
0.017969	0.055255	0.037287	207.5087

0.140625	0.08893	0.051695	36.76109
0.140625	0.070064	0.070561	50.17654
0.053125	0.10979	0.056665	106.6637
0.004688	0.1032	0.098513	2101.605
0.321875	0.164211	0.157664	48.98292
0.021875	0.041976	0.020101	91.89099
0.011719	0.116738	0.105019	896.165
0.144531	0.107711	0.03682	25.47562
0.060938	0.07665	0.015712	25.78446
0.5	0.034295	0.465705	93.14102
0.5	0.039351	0.460649	92.1299
0.5	0.054292	0.445708	89.14164
0.001563	0.082302	0.08074	5167.353
0.5	0.055123	0.444877	88.97548
0.5	0.047379	0.452621	90.52418
0.016406	0.073839	0.057433	350.0656
0.075	0.118167	0.043167	57.55574
0.246094	0.057297	0.188796	76.71727
0.142188	0.090402	0.051786	36.4209
0.5	0.091738	0.408262	81.65236
0.5	0.052892	0.447108	89.42157
0.117969	0.055063	0.062906	53.32402
0.090625	0.038794	0.051831	57.19331
0.5	0.068774	0.431226	86.24519
0.5	0.058554	0.441446	88.28923
0.015625	0.115939	0.100314	642.0097
0.5	0.106684	0.393316	78.66312
0.5	0.036488	0.463512	92.70241
0.078125	0.234631	0.156506	200.3272
0.5	0.073436	0.426564	85.31275
0.140625	0.091254	0.049371	35.10852
0.045313	0.091727	0.046415	102.4324
0.078125	0.115738	0.037613	48.14514

0.130469	0.083274	0.047195	36.17322
0.160156	0.111207	0.048949	30.56321
0.164063	0.155485	0.008577	5.228115
0.045313	0.052003	0.00669	14.76504
0.071094	0.099373	0.02828	39.77797
0.082031	0.144805	0.062774	76.5245
0.001563	0.107863	0.106301	6803.24
0.267188	0.231185	0.036003	13.47467
0.414063	0.169665	0.244398	59.02439
0.16875	0.097996	0.070754	41.92802
0.5	0.065216	0.434784	86.9567
0.073438	0.044144	0.029294	39.88906
0.003906	0.126394	0.122488	3135.689
0.140625	0.109435	0.03119	22.17967
0.5	0.117637	0.382363	76.47251
0.140625	0.049824	0.090801	64.56939
0.003906	0.119433	0.115526	2957.474
0.5	0.119785	0.380215	76.04302
0.023438	0.059478	0.03604	153.7707
0.027344	0.171795	0.144451	528.2782
0.5	0.069334	0.430666	86.13325
0.010938	0.118057	0.10712	979.3822
0.097656	0.069115	0.028541	29.22583
0.140625	0.046036	0.094589	67.26354
0.1875	0.107676	0.079824	42.57269
0.027344	0.135131	0.107787	394.1927
0.002344	0.07131	0.068966	2942.544
0.025	0.108777	0.083777	335.1084
0.036719	0.063115	0.026396	71.88653
0.007031	0.080617	0.073585	1046.549
0.050781	0.104462	0.05368	105.7089
0.009375	0.055426	0.046051	491.21
0.020313	0.067723	0.04741	233.4034

0.5	0.013231	0.486769	97.35373
0.140625	0.105845	0.03478	24.73223
0.219531	0.053386	0.166146	75.68197
0.5	0.092584	0.407416	81.48321
0.820313	0.06789	0.752423	91.72392
0.002344	0.079142	0.076798	3276.734
0.001563	0.02347	0.021907	1402.048
0.140625	0.100668	0.039957	28.41365
0.5	0.091043	0.408957	81.79142
0.5	0.048283	0.451717	90.34337
0.5	0.052179	0.447821	89.56417
0.074219	0.100372	0.026154	35.23841
0.167969	0.096807	0.071161	42.36581
0.054688	0.097045	0.042358	77.45372
0.002344	0.046691	0.044348	1892.167
0.144531	0.073579	0.070952	49.09119
0.359375	0.095904	0.263471	73.31379
0.003906	0.110742	0.106836	2734.99