

**IMPLEMENTASI MODEL *GATED RECURRENT UNIT* (GRU)
ATAU *EXTREME GREDIENT BOOSTING* (XGBOOST)
UNTUK PREDIKSI HARGA *CRYPTOCURRENCY* ETHEREUM**

LAPORAN TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat memperoleh
Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



Disusun Oleh :
MUHAMMAD FAKHRUL REZA
32602100075

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025**

**IMPLEMENTATION OF THE GATE RECURRENT UNIT (GRU)
OR EXTREME GREEN BOOSTING (XGBOOST) MODEL FOR
ETHEREUM CRYPTOCURRENCY PRICE PREDICTION**

FINAL PROJECT

*Proposed to complete the requirement to obtain a bachelor's degree (S1) at
Informatics Engineering Departement of Industrial Technology Faculty Sultan
Agung Islamic University*



ARRANGED BY :

MUHAMMAD FAKHRUL REZA

32602100075

**INFORMATICS ENGINEERING STUDY PROGRAM
FACULTY OF INDUSTRIAL TECHNOLOGY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG**

2025

**LEMBAR PENGESAHAN
TUGAS AKHIR**

**IMPLEMENTASI MODEL *GATED RECURRENT UNIT* (GRU) ATAU
EXTREME GRADIENT BOOSTING (XGBOOST) UNTUK PREDIKSI
HARGA *CRYPTOCURRENCY* ETHEREUM**

**MUHAMMAD FAKHRUL REZA
NIM 32602100075**

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 4 Agustus 2025

TIM PENGUJI UJIAN SARJANA :

**Sam Farisa Chaerul
Haviana, ST., M. Kom**
NIK. 210615046
(Penguji 1)

21 Agustus 2025

Moch. Taufik, ST., MIT
NIK. 210604034
(Penguji 2)

21 Agustus 2025

Ghufron, ST., M. Kom
NIK. 210622056
(Pembimbing)

21 Agustus 2025

Semarang, 21 Agustus 2025
Mengetahui,



NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Muhammad Fakhrol Reza

NIM : 32602100075

Judul Tugas Akhir : IMPLEMENTASI MODEL *GATED RECURRENT UNIT* (GRU) ATAU *EXTREME GRADIENT BOOSTING* (XGBOOST) UNTUK PREDIKSI HARGA *CRYPTOCURRENCY* ETHEREUM

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 25 Agustus 2025

UNISSULA
جامعة سلطان أبوبوع الإسلامية



Muhammad Fakhrol Reza

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Fakhrlul Reza

NIM : 32602100075

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Alamat Asal : Pemalang

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Implementasi Model *Gated Recurrent Unit* (GRU) Atau *Extreme Gradient Boosting* (XGBoost) Untuk Prediksi Harga *Cryptocurrency* Ethereum.

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 25 Agustus 2025


Muhammad Fakhrlul Reza

KATA PENGANTAR

Puji syukur penulis panjatkan kepada ALLAH SWT, yang telah memberikan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Implementasi Model *Gated Recurrent Unit* (GRU) Atau *Extreme Gradient Boosting* (XGBoost) Untuk Prediksi Harga *Cryptocurrency* Ethereum.” ini dengan baik. Dengan penuh rasa hormat, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, SH., M.Hum yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T
3. Dosen pembimbing penulis Bapak Ghufro, ST., M.kom yang telah memberikan arahan, bimbingan, dan saran yang berarti dalam penyelesaian tugas akhir ini.
4. Seluruh dosen Program Studi Teknik Informatika, Fakultas Teknologi Industri UNISSULA yang telah memberikan ilmunya kepada penulis.
5. Orang tua penulis, Bapak Syefrudin dan Ibu Wilastris serta kedua adik penulis yang selalu memberikan segala doa, dukungan, dan motivasi dengan penuh kasih sayang sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik.
6. Teman-teman seperjuangan atas kebersamaannya yang telah bekerja keras serta semangat dalam proses penyelesaian tugas akhir ini.

Semarang, 25 Agustus 2025



Muhammad Fakhru Reza

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN TUGAS AKHIR	iii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR	iv
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
ABSTRAK	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	6
2.2.1 Sistem Prediksi	6
2.2.2 <i>Blockchain</i>	7
2.2.3 <i>Cryptocurrency</i>	13
2.2.4 <i>Ethereum</i>	16
2.2.5 <i>Machine Learning</i>	20
2.2.6 <i>Gated Recurrent Unit (GRU)</i>	20
2.2.7 <i>Extreme Gradient Boosting (XGBoost)</i>	22
BAB III METODE PENELITIAN	25
3.1 Metode Penelitian	25
3.2 Pengumpulan Data	26
3.3 <i>Preprocessing Data</i>	27
3.4 Pembuatan Model	29
3.4.1 Pembuatan Model GRU	30
3.4.2 Pembuatan Model XGBoost	32
3.5 Evaluasi Model	34

3.6	Visualisasi Hasil	35
3.7	Perancangan Sistem.....	35
3.7.1	<i>Wireframe</i> Halaman Awal.....	37
3.7.2	<i>Wireframe</i> Halaman Hasil Prediksi.....	38
3.8	<i>Deployment</i>	39
3.9	Bahasa pemrograman yang digunakan.....	39
3.10	<i>Software</i> yang digunakan	40
3.11	<i>Library</i> yang digunakan	41
BAB IV HASIL DAN ANALISIS PENELITIAN		45
4.1	Hasil Penelitian.....	45
4.1.1	Pengumpulan data	45
4.1.2	<i>Preprocessing data</i>	46
4.1.3	Pembuatan Model <i>Gated Recurrent Unit</i> (GRU) dan <i>Extreme Gradient Boosting</i> (XGBoost)	48
4.1.4	Evaluasi Model.....	50
4.1.5	Visualisasi Hasil.....	52
4.2.1	Visualisasi hasil model GRU	52
4.2.2	Visualisasi hasil model XGBoost.....	53
4.1.6	Hasil <i>Deployment</i>	54
BAB V KESIMPULAN.....		60
5.1	Kesimpulan.....	60
5.2	Saran	60
DAFTAR PUSTAKA.....		62
LAMPIRAN.....		64

DAFTAR GAMBAR

Gambar 2. 1 Cara kerja bolckchain.....	7
Gambar 2. 2 Alur transaksi cryptocurrency	13
Gambar 2. 3 Ethereum	17
Gambar 2. 4 Arsitektur GRU	21
Gambar 2. 5 Arsitektur XGboost	23
Gambar 3. 1 Flowchart Penelitian.....	25
Gambar 3. 2 Data Historis Harga Ethereum di Yahoo Finance	26
Gambar 3. 3 Flowchart pembuatan model GRU.....	30
Gambar 3. 4 Flowchart pembuatan model XGBoost	32
Gambar 3. 5 Flowchart Perancangan Sistem	36
Gambar 3. 6 Wireframe Halaman Awal	37
Gambar 3. 7 Wireframe Halaman Hasil Prediksi.....	38
Gambar 4. 1 Hasil grafik prediksi GRU.....	52
Gambar 4. 2 Hasil grafik prediksi XGBoost.....	53
Gambar 4. 3 Halaman Awal.....	55
Gambar 4. 5 Halaman hasil prediksi dan grafik prediksi model GRU	56
Gambar 4. 6 Halaman hasil prediksi dan grafik prediksi model XGBoost.....	57
Gambar 4. 7 Tabel hasil prediksi model GRU	58
Gambar 4. 8 Hasil prediksi model XGBoost	59



DAFTAR TABEL

Tabel 4. 1 Dataset harga ethereum.....	45
Tabel 4. 2 Handling Missing Value	46
Tabel 4. 3 Mengecek duplikasi data.....	47
Tabel 4. 4 Hasil Splitting Data Pada model GRU.....	48
Tabel 4. 5 Arsitektur model GRU	48
Tabel 4. 6 Hyperparameter model GRU	49
Tabel 4. 7 Hyperparameter model XGBoost.....	50
Tabel 4. 8 Hasil evaluasi model GRU.....	51
Tabel 4. 9 Nilai evaluasi model XGBoost	52



ABSTRAK

Ethereum, sebagai salah satu aset kripto utama, memiliki volatilitas harga yang tinggi, sehingga menciptakan kebutuhan akan model prediksi yang akurat untuk membantu pengambilan keputusan investasi. Penelitian ini bertujuan untuk mengimplementasikan kinerja dua model *machine learning* populer, yaitu *Gated Recurrent Unit* (GRU) yang merupakan model *deep learning* untuk data sekuensial, dan *Extreme Gradient Boosting* (XGBoost) yang merupakan model ensemble. Data yang digunakan adalah data historis harga harian Ethereum yang mencakup fitur *Open, High, Low, Close, Volume* (OHLCV). Metode penelitian meliputi tahap pra-pemrosesan data seperti normalisasi *Min-Max Scaler* dan pembagian data dengan rasio 80% data latih dan 20% data uji. Evaluasi kinerja kedua model diukur menggunakan metrik *Root Mean Squared Error* (RMSE) dan *R-squared* (R^2). Hasil pengujian menunjukkan bahwa model GRU menghasilkan prediksi yang lebih baik, mencapai nilai RMSE 101.37 dan R^2 0.9718, sedangkan model XGBoost memperoleh nilai RMSE 435.68 dan R^2 0.4433. Hal ini mengindikasikan bahwa kemampuan GRU dalam menangkap pola dan dependensi temporal pada data deret waktu lebih unggul untuk kasus prediksi harga Ethereum. Kesimpulan dari penelitian ini adalah model GRU lebih efektif dan dapat diandalkan untuk memprediksi harga Ethereum dibandingkan XGBoost dalam penelitian ini.

Kata Kunci: Prediksi Harga, Ethereum, *Gated Recurrent Unit* (GRU), XGBoost, *Root Mean Squared Error* (RMSE).

ABSTRACT

As a leading cryptocurrency, Ethereum exhibits high price volatility, creating a need for accurate prediction models to aid in investment decision-making. This study aims to implement and compare the performance of two popular machine learning models: the *Gated Recurrent Unit* (GRU), a deep learning model for sequential data, and *Extreme Gradient Boosting* (XGBoost), an ensemble model. The research utilizes historical daily Ethereum price data, including *Open, High, Low, Close, and Volume* (OHLCV) features. The methodology includes a data preprocessing phase with *Min-Max Scaler* normalization and an 80% training and 20% testing data split. The performance of both models was evaluated using the *Root Mean Squared Error* (RMSE) and *R-squared* (R^2) metrics. The test results show that the GRU model yields better predictions, achieving an RMSE of 101.37 and an R^2 of 0.9718, whereas the XGBoost model obtained an RMSE of 435.68 and an R^2 of 0.4433. This indicates that the GRU's ability to capture patterns and temporal dependencies in time-series data is superior for the case of Ethereum price prediction. The study concludes that the GRU model is more effective and reliable for predicting Ethereum prices compared to XGBoost within this research framework.

Keywords: Price Prediction, Ethereum, *Gated Recurrent Unit* (GRU), XGBoost, *Root Mean Squared Error* (RMSE).

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi semakin menghasilkan inovasi yang canggih. Salah satunya adalah teknologi *blockchain*. Teknologi *Blockchain* adalah teknologi inovatif yang memungkinkan pengguna untuk berkomunikasi tanpa memerlukan perantara yang tepercaya. Teknologi *Blockchain* menarik minat komunitas penelitian pada tahun 2008 ketika sebuah *white paper* diterbitkan oleh Satoshi Nakamoto tentang masalah pengeluaran ganda dalam jaringan desentralisasi *peer-to-peer* (Kushwaha dkk., 2022).

Dalam dunia investasi, teknologi blockchain ini juga digunakan dalam aset digital berupa *cryptocurrency*. *Cryptocurrency* ini memiliki keunggulan, seperti sistem pengelolaan yang terdesentralisasi, tingkat keamanan tinggi yang menjaga kerahasiaan, dan kemudahan dalam proses transaksi (Satria Andromeda & Anisa Sri Winarsih, 2025). Berdasarkan Peraturan Badan Pengawas Perdagangan Berjangka Komoditi (Bappebti) Nomor 7 Tahun 2020, perdagangan aset kripto telah memperoleh legalitas di Indonesia, di mana peraturan tersebut menetapkan daftar aset kripto yang sah untuk ditransaksikan di pasar fisik (Arta Pangaribuan dkk., 2023).

Pergerakan harga *cryptocurrency* yang fluktuatif, ditambah dengan potensinya untuk terus meningkat, telah menjadi daya tarik utama yang mendorong banyak orang untuk berinvestasi di dalamnya. Khususnya Bitcoin dan Ethereum yang menduduki posisi teratas dalam kapitalisasi pasar *cryptocurrency* sebagai instrumen investasi. Ethereum merupakan salah satu aset digital paling populer setelah Bitcoin dan telah menarik perhatian banyak investor di seluruh dunia. Ethereum adalah sebuah platform terdesentralisasi yang dibangun di atas teknologi *blockchain* dengan berbasis *smart contract* memungkinkan berbagai layanan transaksi pertukaran nilai secara daring (Pradana Ananda Raharja, 2021). Volatilitas harga yang tinggi pada aset kripto seperti Ethereum menjadikannya peluang sekaligus tantangan bagi para investor ataupun *trader* kripto. Fluktuasi harga yang tajam dapat menghasilkan keuntungan besar, tetapi di sisi lain juga mengandung risiko

kerugian yang sangat tinggi, terutama bagi investor pemula. Konteks risiko ini menjadi semakin relevan di Indonesia, di mana minat masyarakat terhadap investasi aset kripto terus meningkat tajam. Menurut data dari Badan Pengawas Perdagangan Berjangka Komoditi (Bappebti), jumlah investor kripto di Indonesia telah melampaui 22,1 juta orang per November 2024. Besarnya jumlah investor ritel ini, yang sebagian besar mungkin belum memiliki pemahaman mendalam tentang analisis pasar, menciptakan urgensi akan adanya alat bantu analisis prediktif. Tanpa alat bantu yang akurat, investor menjadi sangat rentan dalam mengambil keputusan, sehingga berisiko kehilangan modal secara signifikan di tengah dinamika pasar yang ekstrim (Moch Farryz Rizkilloh & Sri Widiyanesti, 2022).

Untuk menjawab tantangan tersebut, penelitian ini memanfaatkan *machine learning* dengan menguji dua pendekatan yang berbeda. Pendekatan pertama adalah *Deep Learning* melalui model *Gated Recurrent Unit* (GRU), sebagai bagian dari keluarga *Recurrent Neural Network* (RNN), unggul dalam menangkap pola data *time series* dalam data historis. Pendekatan kedua adalah *Ensemble Learning* melalui *Extreme Gradient Boosting* (XGBoost), yang unggul menangani data tabular dalam memodelkan hubungan kompleks dari fitur-fitur yang direkayasa secara eksplisit (Nusaiba Yulisa dkk., 2023).

Implementasi model GRU dan XGBoost untuk memprediksi harga Ethereum diharapkan dapat memberikan gambaran tren harga di masa depan secara lebih akurat. Hasil prediksi ini dapat dimanfaatkan sebagai alat bantu dalam pengambilan keputusan investasi, khususnya bagi investor atau trader kripto yang ingin meminimalkan risiko dan memaksimalkan peluang pada pasar *cryptocurrency* yang sangat dinamis (Moch Farryz Rizkilloh & Sri Widiyanesti, 2022).

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengimplementasikan model GRU dan XGBoost dalam memprediksi harga Ethereum, sehingga dapat memberikan sebuah informasi yang bermanfaat bagi para investor ataupun trader kripto dalam mengambil keputusan strategi investasi yang lebih baik.

1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah :

1. Bagaimana implementasi model GRU untuk memprediksi harga ethereum dalam USD?
2. Bagaimana implementasi model XGBoost untuk memprediksi harga ethereum dalam USD?
3. Bagaimana hasil evaluasi model GRU dan XGBoost untuk memprediksi harga ethereum dalam USD?

1.3 Batasan Masalah

Untuk menjaga fokus dan ruang lingkup penelitian, terdapat beberapa batasan yang ditetapkan yaitu :

1. Penelitian ini hanya memfokuskan pada prediksi harga Ethereum (ETH-USD) dengan menggunakan data historis harian harga Ethereum dalam USD yang diperoleh dari yahoo finance.
2. Dataset yang digunakan berisi data *date*, *open*, *high*, *low*, dan *volume*.
3. Model yang akan dipakai dalam penelitian ini adalah model GRU dan XGBoost saja tanpa membahas secara mendalam model lain.
4. Tujuan utama penelitian ini adalah implementasi dan evaluasi masing-masing model.
5. Penelitian ini fokus pada prediksi harga penutupan (*close*) harian Ethereum dalam USD dan tidak mempertimbangkan faktor-faktor lain seperti volume perdagangan, berita pasar, sentimen sosial media, atau kejadian-kejadian spesifik yang dapat mempengaruhi harga.
6. Prediksi yang dilakukan dalam penelitian ini yaitu untuk 15 hari kedepan pada masing-masing model yaitu GRU dan XGBoost.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengimplementasikan kinerja model GRU dan XGBoost dalam memprediksi harga ethereum di masa depan yang dapat digunakan untuk mendukung pengambilan keputusan dalam investasi aset kripto khususnya pada ethereum.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan wawasan kepada investor dan trader kripto mengenai potensi pemanfaatan model prediktif berbasis AI sebagai alat bantu dalam menganalisis pergerakan harga Ethereum. Meskipun bukan merupakan saran finansial, hasil analisis dapat membantu dalam perumusan strategi investasi atau manajemen risiko yang lebih terinformasi.

1.6 Sistematika Penulisan

Struktur penulisan yang akan diterapkan dalam pembuatan laporan tugas akhir adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada BAB I menjelaskan tentang latar belakang, pemilihan judul, rumusan masalah, Batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penelitian

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Pada BAB II memuat tentang penelitian terdahulu dan landasan teori yang berkaitan untuk membantu memahami konsep algoritma GRU dan XGBoost untuk melengkapi penelitian ini.

BAB III : METODE PENELITIAN

Pada BAB III menjelaskan proses penelitian yang dimulai dari pengumpulan data hingga evaluasi hasil prediksi.

BAB IV : HASIL DAN ANALISIS PENELITIAN

Pada BAB IV berisi tentang pemaparan hasil penelitian yang dimulai dari pembuatan sistem sampai dengan proses deployment.

BAB V : KESIMPULAN DAN SARAN

Pada BAB V merangkum keseluruhan proses penelitian dari awal sampai akhir.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada penelitian sebelumnya memperlihatkan bahwa *Gated Recurrent Unit* (GRU) secara konsisten memberikan performa yang lebih baik dibandingkan dengan *Long Short-Term Memory* (LSTM) dalam melakukan prediksi harga *cryptocurrency* seperti Bitcoin (BTC) dan Ethereum (ETH). GRU menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) dan *Root Mean Square Error* (RMSE) yang lebih rendah pada sebagian besar konfigurasi *hyperparameter*, yang menunjukkan bahwa prediksinya lebih akurat dan stabil. Sebagai ilustrasi, pada prediksi harga BTC, GRU mencatat MAPE terendah sebesar 0,38%, sementara LSTM hanya mampu mencapai 0,46%. Untuk prediksi ETH, GRU memperoleh MAPE serendah 0,44%, secara signifikan lebih baik daripada LSTM yang mencatat 1,64% pada konfigurasi tertentu. Selain itu, pola prediksi GRU lebih konsisten dan mendekati nilai aktual, sedangkan LSTM cenderung lebih sensitif terhadap perubahan *hyperparameter* dan menghasilkan fluktuasi prediksi yang lebih tinggi (Satria Andromeda & Anisa Sri Winarsih, 2025).

Penelitian sebelumnya tentang penggunaan metode GRU untuk memprediksi harga saham Coca-Cola dengan data harga penutupan menunjukkan hasil yang baik. Penelitian ini menggunakan *window-width* 3, jumlah neuron 64, *batch size* 32, dan *epoch* sebanyak 30, dengan rasio data latih dan uji 80:20. Model dikompilasi menggunakan *optimizer Adam*, *loss function mean squared error* (MSE), dan evaluasi menggunakan *mean absolute error* (MAE). Hasilnya, model mencapai nilai MAE sebesar 0,42, RMSE 0,64, dan MSE 0,40 (Silalahi & Muljono, 2024).

Pada penelitian lain tentang optimasi model XGBoost terhadap data harga saham PT United Tractors Tbk. menunjukkan bahwa model yang dihasilkan memiliki tingkat akurasi yang baik. Model terbaik diperoleh dengan kombinasi *hyperparameter* *gamma* sebesar 0,01, *learning rate* sebesar 0,05, *max_depth* sebesar 15, dan jumlah estimasi pohon (*n_estimators*) sebanyak 200. Berdasarkan hasil eksperimen, model ini menghasilkan nilai *Mean Absolute Percentage Error*

(MAPE) sebesar 3,89%. Dengan nilai tersebut, model dapat dikategorikan sebagai model yang akurat dalam melakukan prediksi (Astutiningsih dkk., 2023).

Dalam penelitian lain teknik data mining yang diterapkan melalui algoritma *Random Forest* dan *XGBoost* menunjukkan hasil prediksi yang akurat dan konsisten. Penelitian ini menggunakan dataset klinis dan biokimia yang diperoleh dari situs Kaggle, terdiri dari 768 data dengan 9 indikator. Setelah melalui proses *preprocessing* termasuk penanganan data hilang, penanganan *outlier*, dan normalisasi diperoleh 688 data yang layak untuk dianalisis. Selanjutnya, dilakukan proses pelatihan dan pengujian menggunakan teknik *Cross Validation*, serta pencarian parameter terbaik untuk masing-masing model. Evaluasi kinerja model dilakukan menggunakan metrik akurasi, presisi, *recall*, dan *F1-score*. Hasil evaluasi menunjukkan bahwa model memiliki performa yang baik, dengan tingkat akurasi sebesar 74% untuk *Random Forest* dan 76% untuk *XGBoost* (Salsabil Muhammad dkk., 2024).

Pada penelitian lain hasil dari evaluasi kinerja model *Vector Autoregressive* (VAR) dalam memprediksi harga Ethereum terhadap Dolar AS, yang diukur menggunakan *Root Mean Squared Error* (RMSE), menunjukkan adanya selisih yang cukup besar antara nilai prediksi dan nilai aktual. Skor RMSE yang dihasilkan untuk harga pembukaan adalah 890,29, harga tertinggi 930,50, harga terendah 1.164,37, dan harga penutupan 978,37. Tingginya skor RMSE pada setiap variabel ini menandakan bahwa akurasi prediktif model tersebut masih belum dapat dianggap optimal (Pradana Ananda Raharja, 2021).

2.2 Dasar Teori

2.2.1 Sistem Prediksi

Sebuah sistem prediksi adalah suatu sistem terintegrasi yang terdiri dari serangkaian metodologi, algoritma, dan teknologi yang bekerja sama untuk menganalisis data historis dan saat ini guna meramalkan kejadian atau nilai di masa depan. Lebih dari sekadar sebuah model, sistem prediksi mencakup keseluruhan alur kerja dari pengumpulan data hingga penyajian hasil kepada pengguna akhir. Tujuannya adalah untuk memberikan informasi proaktif yang dapat digunakan sebagai dasar untuk pengambilan keputusan berbasis data (Kusuma dkk., 2021).

Dalam konteks *machine learning*, sebuah sistem prediksi mengotomatiskan proses pembuatan peramalan dengan memanfaatkan model matematis yang telah "belajar" dari pola data masa lalu untuk membuat prediksi pada data baru yang belum pernah dilihat.

2.2.2 Blockchain

2.2.2.1 Pengertian Blockchain

Blockchain adalah sebuah teknologi basis data terdistribusi (*distributed ledger technology*) yang menyimpan catatan transaksi secara permanen, transparan, dan aman dalam bentuk blok-blok yang saling terhubung. Setiap blok berisi sekumpulan data transaksi yang diverifikasi dan disimpan secara permanen di jaringan (Gupta, 2022). Blok dapat diibaratkan halaman dalam buku besar digital, dimana setiap halaman mencatat sejumlah transaksi dan semua halaman dihubungkan secara berurutan sehingga membentuk rantai (*chain*) yang tidak dapat diubah tanpa persetujuan mayoritas jaringan.

Blockchain pertama kali diperkenalkan melalui protokol Bitcoin sebagai buku besar digital publik yang bersifat *decentralized* tanpa memerlukan otoritas pusat. Teknologi ini memungkinkan dua pihak atau lebih melakukan transaksi secara langsung (*peer-to-peer*) dengan tingkat keamanan tinggi dan kepercayaan yang terjamin melalui mekanisme konsensus (Dhea Larasati & Primandari, 2021).



Gambar 2. 1 Cara kerja blockchain

Pada gambar 2.1 merupakan alur transaksi pada teknologi *blockchain* yang dimulai dari permohonan transaksi oleh pengguna. Transaksi ini kemudian disiarkan ke jaringan *peer-to-peer* (P2P) untuk divalidasi oleh para peserta jaringan

(*nodes*). Setelah dinyatakan sah, transaksi tersebut akan digabungkan dengan transaksi lain yang juga telah terverifikasi untuk membentuk sebuah blok baru. Blok baru ini selanjutnya ditambahkan secara permanen ke dalam rantai *blockchain* yang ada. Proses diakhiri dengan status transaksi berhasil, yang menandakan bahwa catatan transaksi telah final dan tidak dapat diubah.

2.2.2.2 Karakteristik *Blockchain*

Blockchain memiliki beberapa karakteristik utama yang membedakannya dari basis data tradisional, antara lain:

1. Desentralisasi (*Decentralization*)

Tidak ada pihak tunggal yang mengontrol jaringan. Semua node dalam jaringan memiliki salinan lengkap dari buku besar, sehingga mengurangi risiko kegagalan tunggal (*single point of failure*).

2. Transparansi (*Transparency*)

Semua transaksi yang terjadi dapat dilihat oleh seluruh peserta jaringan. Meskipun data dapat dilihat publik, identitas pengguna tetap anonim melalui penggunaan alamat kriptografi.

3. *Immutability* (Tidak Dapat Diubah)

Setelah data atau transaksi dicatat di dalam *blockchain* dan divalidasi, data tersebut tidak dapat diubah atau dihapus. Hal ini dicapai melalui penerapan algoritma *hash* kriptografi.

4. Keamanan (*Security*)

Blockchain menggunakan teknik kriptografi seperti *hash function*, tanda tangan digital (*digital signature*), dan algoritma enkripsi untuk melindungi integritas dan autentikasi data.

5. Distribusi (*Distributed Ledger*)

Salinan buku besar disimpan di banyak node di seluruh jaringan, sehingga membuat sistem lebih tahan terhadap serangan atau kerusakan pada sebagian node.

2.2.2.3 Komponen Utama *Blockchain*

Blockchain dibangun dari beberapa komponen utama sebagai berikut :

1. *Block* (Blok)

Block adalah unit dasar dari *blockchain* yang berfungsi sebagai wadah untuk menyimpan data transaksi yang telah diverifikasi oleh jaringan. Setiap *block* dapat diibaratkan seperti sebuah “halaman buku besar digital” yang mencatat semua aktivitas atau transaksi yang terjadi dalam kurun waktu tertentu. Data yang tersimpan di dalam *block* meliputi informasi transaksi (misalnya pengirim, penerima, jumlah aset), timestamp (waktu pencatatan), serta *hash* unik yang menjadi identitas *block* tersebut. Selain itu, setiap *block* juga menyimpan *hash* dari *block* sebelumnya, sehingga membentuk sebuah rantai yang saling terhubung dan sulit diubah.

Struktur *block* pada umumnya terdiri dari dua bagian utama, yaitu *block header* dan *block body*. *Block header* berisi informasi penting seperti *hash* *block* sebelumnya, *timestamp*, *nonce* (bilangan acak yang digunakan pada mekanisme konsensus), serta *Merkle root* (*hash* dari seluruh transaksi dalam *block*). Sedangkan *block body* berisi kumpulan transaksi yang sudah divalidasi oleh node atau validator dalam jaringan.

2. *Hash*

Hash dalam konteks *blockchain* adalah hasil keluaran dari sebuah fungsi *hash* kriptografi (*cryptographic hash function*), yaitu algoritma matematika yang mengubah data berukuran berapa pun menjadi *string* atau kode unik dengan panjang tetap.

Kode *hash* ini bersifat unik untuk setiap data masukan (*input*). Artinya, jika mengubah satu huruf atau bahkan satu bit saja pada data awal, hasil *hash*-nya akan berubah total secara drastis (*avalanche effect*). Dalam *blockchain*, *hash* biasanya ditulis dalam format heksadesimal (angka 0–9 dan huruf a–f).

Dalam dunia *cryptocurrency*, ada banyak jenis algoritma *hash* yang digunakan, dan setiap *blockchain* biasanya memilih algoritma yang sesuai dengan tujuannya. Berikut adalah jenis-jenis *hash* yang umum digunakan :

Tabel 2. 1 jenis algoritma hash

Algoritma Hash	Pengguna Utama	Panjang Output	Karakteristik Utama	Kelebihan	Kekurangan
SHA-256	Bitcoin, Bitcoin Cash	256-bit (64 hex)	Digunakan untuk menghasilkan <i>hash block header</i> , mengamankan alamat <i>wallet</i>	Keamanan tinggi, standar industri	Boros daya, mining terpusat di ASIC
Keccak-256 (SHA-3)	Ethereum	256-bit (64 hex)	Varian SHA-3, digunakan untuk alamat dan tanda tangan digital	Aman, efisien, struktur berbeda dari SHA-2	Tidak sepopuler SHA-256 di mining
Scrypt	Litecoin, Dogecoin	256-bit	Memory-hard, awalnya anti-ASIC	Lebih ramah CPU/GPU awalnya	ASIC Scrypt kini tersedia
Ethash	Ethereum (PoW)	256-bit	Dirancang ASIC-resistant, gunakan DAG besar	Mendorong desentralisasi mining	Memori besar dibutuhkan, tidak ramah perangkat lemah
X11	Dash	256-bit	Menggabungkan 11 algoritma hash berurutan	Lebih hemat daya awalnya, keamanan berlapis	ASIC X11 kini ada
Blake2b / Blake2s	Siacoin (B2b), Decred (B256)	256-bit	Lebih cepat dari SHA-2, efisien daya	Performa tinggi, efisiensi energi	Belum sepopuler SHA-256
CryptoNight	Monero	256-bit	Memory-hard, ramah CPU, mendukung privasi	Anti-ASIC relatif lama, privasi kuat	ASIC CryptoNight mulai bermunculan
Equihash	Zcash, Komodo	Variatif (umum 192-bit)	Proof-of-space, memori besar	Anti-ASIC awalnya, aman	Ukuran memori besar, ASIC sudah tersedia

3. Node

Node adalah komputer atau perangkat yang terhubung ke jaringan *blockchain* dan berperan dalam menyimpan, menyebarkan, serta memverifikasi data transaksi. Setiap node menjalankan perangkat lunak (*software*) *blockchain* yang sesuai dengan protokol jaringan, misalnya Bitcoin Core untuk Bitcoin atau Geth/Nethermind untuk Ethereum.

Karena *blockchain* bersifat terdesentralisasi, tidak ada server pusat. Semua informasi transaksi dan blok disimpan di node-node yang tersebar di

seluruh dunia. Dengan demikian, node menjadi komponen penting yang menjaga keamanan, transparansi, dan konsistensi jaringan *blockchain*.

4. Mekanisme Konsensus

Mekanisme konsensus adalah inti dari teknologi *blockchain* yang bertujuan memastikan semua node di jaringan terdistribusi memiliki salinan buku besar (*ledger*) yang sama dan valid. Berikut ini adalah beberapa jenis mekanisme konsensus yang umum digunakan:

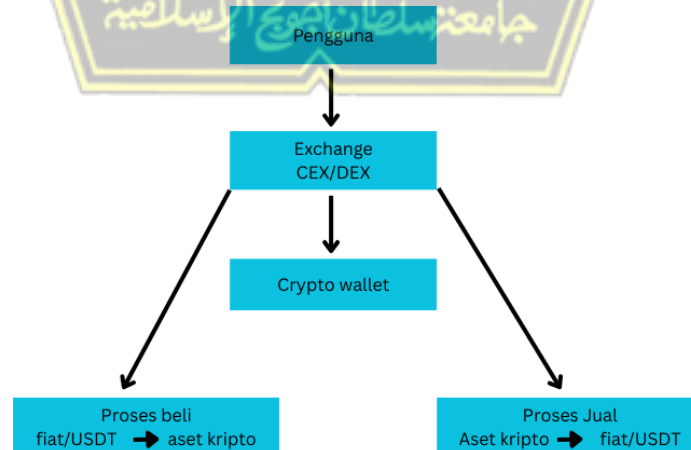
- a. *Proof of Work* (PoW) yaitu mekanisme di mana penambang (*miner*) bersaing memecahkan teka-teki kriptografi yang kompleks untuk memvalidasi transaksi dan menambahkan blok baru ke *blockchain*. Prosesnya dimulai dengan pengumpulan transaksi yang belum terverifikasi, kemudian penambang mencoba menemukan nilai *nonce* yang menghasilkan *hash* sesuai tingkat kesulitan (*difficulty*) jaringan. Penambang pertama yang berhasil akan mengumumkan hasilnya dan setelah diverifikasi oleh node lain, blok tersebut ditambahkan ke *blockchain*. Keunggulan PoW adalah keamanannya yang tinggi dan telah terbukti efektif sejak digunakan oleh Bitcoin pada tahun 2009, namun kekurangannya adalah konsumsi energi yang sangat besar dan kecepatan transaksi yang rendah. Contoh implementasinya adalah Bitcoin, Litecoin, dan Ethereum sebelum *Merge*.
- b. *Proof of Stake* (PoS) yaitu mekanisme konsensus yang memilih validator berdasarkan jumlah koin yang mereka *stake* atau taruh sebagai jaminan. Calon validator akan mengunci sejumlah koin, kemudian algoritma secara acak memilih validator berdasarkan jumlah dan lama *stake*. Validator yang terpilih bertugas memverifikasi transaksi dan membuat blok baru. PoS memiliki keunggulan dalam efisiensi energi dan kecepatan transaksi yang lebih tinggi dibanding PoW, namun kelemahannya adalah potensi sentralisasi jika sebagian besar koin dikuasai oleh segelintir pihak. Contoh *blockchain* yang menggunakan PoS adalah Ethereum (setelah *Merge*), Cardano, dan Polkadot.

- c. *Delegated Proof of Stake* (DPoS) merupakan varian dari PoS di mana pemilik koin melakukan voting untuk memilih sekelompok delegasi (*delegates*) yang bertugas memvalidasi transaksi. Delegasi yang terpilih akan memproses blok secara bergiliran, dan hadiah blok akan dibagi antara delegasi dan pemilihnya. Kelebihan DPoS adalah kecepatan dan efisiensi yang lebih tinggi dibanding PoW dan PoS murni, serta konsumsi energi yang rendah. Namun, kelemahannya adalah tingkat sentralisasi yang lebih tinggi karena hanya segelintir node yang berhak memvalidasi blok. Contoh penerapannya dapat ditemukan pada EOS, TRON, dan Steem.
- d. *Proof of Authority* (PoA) yaitu mekanisme yang mengandalkan validator dengan identitas dan reputasi yang telah diverifikasi, sehingga lebih sering digunakan pada *blockchain* privat atau konsorsium. Dalam PoA, hanya validator yang dipercaya yang dapat membuat blok baru dan memvalidasi transaksi berdasarkan aturan jaringan. Keunggulannya adalah kecepatan dan efisiensi yang sangat tinggi, namun kelemahannya terletak pada sifatnya yang sangat terpusat dan memerlukan kepercayaan pada pihak validator. Contoh implementasinya adalah VeChain.
- e. *Byzantine Fault Tolerance* (BFT) dan variannya, merupakan konsensus yang mampu mencapai kesepakatan meskipun terdapat sebagian node yang gagal atau berperilaku jahat. Mekanisme ini bekerja dengan node yang saling bertukar pesan untuk memverifikasi transaksi, dan jika mayoritas node setuju, transaksi dianggap valid. Beberapa varian BFT adalah *Practical Byzantine Fault Tolerance* (PBFT) yang efisien untuk jaringan kecil hingga menengah, dan *Tendermint* BFT yang digunakan pada Cosmos. Kelebihan BFT adalah toleransinya terhadap kesalahan node dan kecepatan transaksi yang tinggi pada jaringan kecil, sedangkan kelemahannya adalah ketidakefisiennya jika digunakan pada jaringan publik berskala besar. Contoh penerapan BFT dapat ditemukan pada Hyperledger Fabric dan Cosmos.

2.2.3 Cryptocurrency

Cryptocurrency merupakan sebuah bentuk mata uang digital yang tercipta melalui teknologi *blockchain*, terdiri dari rangkaian kode kriptografi yang diatur sedemikian rupa agar dapat disimpan dalam perangkat komputer dan dipindahkan seperti surat elektronik. Kriptografi ini bertujuan untuk mengamankan transaksi, mengontrol penciptaan unit baru, dan memverifikasi transfer aset (Pourpourides, 2025).

Berbeda dengan mata uang tradisional (fiat) yang dikeluarkan dan diatur oleh otoritas pusat seperti bank sentral, sebagian besar *cryptocurrency* beroperasi secara terdesentralisasi di atas teknologi yang dikenal sebagai *blockchain* (rantai blok). Teknologi ini memungkinkan setiap transaksi untuk dicatat dalam sebuah buku besar yang terdistribusi secara terbuka dan terdesentralisasi (Putri dkk., 2022). Mata uang kripto ini memungkinkan penggunaannya sebagai alat pembayaran yang praktis. Dengan peredaran yang pesat di seluruh dunia, kemunculan *cryptocurrency* memiliki potensi besar untuk berdampak signifikan pada perekonomian global, memberikan indikasi bahwa fenomena ini dapat memengaruhi dinamika perekonomian dunia (Futri, 2024). Proses ini berjalan melalui serangkaian tahapan sistematis yang diamankan oleh kriptografi untuk memastikan validitas dan keamanan tanpa memerlukan perantara terpusat. Berikut adalah alur sistematis dari sebuah transaksi *cryptocurrency*.



Gambar 2. 2 Alur transaksi cryptocurrency

Pada gambar 2.2 merupakan alur jual-beli *cryptocurrency* menunjukkan bagaimana interaksi terjadi antara pengguna, *exchange*, dan *crypto wallet*. Pengguna adalah pihak yang melakukan transaksi jual atau beli aset digital. *Exchange* berperan sebagai tempat pertukaran yang terbagi menjadi dua jenis, yaitu *Centralized Exchange* (CEX) seperti Binance, Indodax, dan Coinbase yang dikelola pihak ketiga, serta *Decentralized Exchange* (DEX) seperti Uniswap dan PancakeSwap yang berjalan otomatis melalui *smart contract*. Dompet kripto digunakan untuk menyimpan aset digital, yang terbagi menjadi *hot wallet* (terhubung internet) dan *cold wallet* (offline dengan keamanan lebih tinggi). Pada proses beli, pengguna menukarkan uang fiat (misalnya Rupiah, USD) atau stablecoin (seperti USDT) melalui *exchange* untuk mendapatkan aset kripto seperti ETH atau BTC, yang kemudian dapat disimpan di *exchange* atau ditransfer ke *crypto wallet* pribadi. Sebaliknya, pada proses jual, pengguna mengirimkan aset kripto dari *crypto wallet* atau *exchange* untuk ditukar menjadi fiat atau stablecoin, lalu hasilnya dapat ditarik ke rekening bank (melalui CEX) atau langsung dipertukarkan di *blockchain* (melalui DEX). Dengan demikian, ilustrasi ini menegaskan bahwa pengguna bertindak sebagai pengendali, *exchange* sebagai tempat pertukaran, dan *crypto wallet* sebagai penyimpanan aset.

Perkembangan *cryptocurrency* tidak hanya berhenti pada kehadiran Bitcoin sebagai uang digital pertama di dunia, tetapi terus berevolusi seiring dengan meningkatnya kebutuhan teknologi dan kompleksitas sistem keuangan modern. Dari awalnya hanya berfungsi sebagai alat tukar digital, *cryptocurrency* berkembang menjadi platform yang mampu menjalankan aplikasi terdesentralisasi, hingga menghadirkan solusi untuk masalah skalabilitas, interoperabilitas, dan efisiensi energi. Berdasarkan tahap perkembangannya, *cryptocurrency* umumnya dibagi ke dalam tiga generasi utama yaitu :

1. Generasi pertama, yang dikenal sebagai *Cryptocurrency 1.0*, berfokus pada fungsi utamanya sebagai alat pembayaran digital terdesentralisasi tanpa memerlukan pihak ketiga seperti bank atau pemerintah. Contoh paling terkenal dari generasi ini adalah Bitcoin (BTC) yang diperkenalkan oleh Satoshi Nakamoto pada tahun 2009. Fokus utama dari Bitcoin adalah sebagai *store of*

value atau penyimpan nilai layaknya emas digital, serta menyediakan transaksi *peer-to-peer* yang aman. Sistem konsensus yang digunakan adalah *Proof of Work* (PoW) melalui proses mining. Meskipun aman dan terdesentralisasi, generasi pertama ini memiliki kekurangan, yakni skalabilitas terbatas karena jumlah transaksi per detik yang rendah dan tidak mendukung aplikasi kompleks di luar transfer nilai. Dengan demikian, generasi pertama dapat dikatakan sebagai mata uang digital murni.

2. Generasi kedua, yaitu *Cryptocurrency 2.0* yang hadir untuk mengatasi keterbatasan Bitcoin. Generasi ini tidak hanya berfungsi sebagai uang digital, tetapi juga sebagai platform aplikasi terdesentralisasi (DApps) yang didukung oleh *smart contract*. Contoh utama dari generasi ini adalah Ethereum (ETH) yang dirilis pada tahun 2015. Ethereum memungkinkan pengembang untuk membangun aplikasi berbasis *blockchain* dengan logika bisnis tertentu yang berjalan secara otomatis. Selain itu, Ethereum juga melahirkan berbagai standar token seperti ERC-20 untuk token fungible dan ERC-721 untuk NFT. Dari sini berkembanglah ekosistem DeFi, NFT, dan DAO. Keunggulan dari generasi kedua ini adalah fleksibilitasnya yang jauh lebih besar dibanding Bitcoin, karena mampu mengeksekusi berbagai logika bisnis di atas *blockchain*. Namun, kelemahannya terletak pada biaya transaksi (*gas fee*) yang tinggi dan masih adanya keterbatasan dalam hal skalabilitas. Oleh karena itu, generasi kedua dapat dipandang sebagai *blockchain* yang berfungsi bukan hanya sebagai uang digital, melainkan juga sebagai platform aplikasi terdesentralisasi.
3. Generasi ketiga, yaitu berkembang *Cryptocurrency 3.0* yang bertujuan untuk memperbaiki kelemahan dari generasi sebelumnya, khususnya dalam hal skalabilitas, kecepatan transaksi, biaya, dan efisiensi energi. Beberapa contoh dari generasi ini adalah Cardano (ADA), Polkadot (DOT), Solana (SOL), dan Avalanche (AVAX). Fokus utama dari generasi ketiga ini adalah pada skalabilitas, yakni kemampuan untuk memproses ribuan hingga jutaan transaksi per detik, interoperabilitas yang memungkinkan *blockchain* saling terhubung (*cross-chain*), serta sustainability dengan penggunaan konsensus yang lebih hemat energi seperti *Proof of Stake* (PoS). Kelebihannya terletak pada transaksi

yang lebih cepat, biaya yang lebih murah, dan sifatnya yang ramah lingkungan sehingga lebih sesuai untuk penggunaan massal, baik oleh perusahaan, pemerintah, maupun integrasi dengan *Internet of Things* (IoT). Namun, kekurangan dari generasi ini adalah sebagian besar proyek masih dalam tahap pengembangan dan tingkat adopsinya belum sebesar Bitcoin maupun Ethereum.

Cryptocurrency memiliki beberapa karakteristik ekonomi yang membedakannya dari aset tradisional:

- Pasokan Terbatas (*Fixed Supply*): Banyak *cryptocurrency*, seperti Bitcoin, memiliki jumlah pasokan maksimum yang telah ditentukan dalam kodenya (21 juta untuk Bitcoin). Sifat kelangkaan ini adalah salah satu faktor yang dapat mendorong nilainya.
- Volatilitas Tinggi: Harga *cryptocurrency* dikenal sangat fluktuatif. Nilainya dapat mengalami perubahan puluhan persen dalam satu hari. Volatilitas ini disebabkan oleh berbagai faktor, termasuk sentimen pasar, berita regulasi, adopsi teknologi, dan sifat spekulatifnya.
- Pasar 24/7: Berbeda dengan pasar saham tradisional, pasar *cryptocurrency* tidak pernah tutup. Perdagangan terjadi selama 24 jam sehari, 7 hari seminggu, di seluruh dunia.

2.2.4 Ethereum

Ethereum merupakan sebuah platform digital berbasis teknologi *blockchain* terdesentralisasi yang menggunakan *smart contract* untuk layanan transaksi nilai secara online. Sistem *peer-to-peer* yang diadopsi oleh Ethereum memungkinkan para penggunanya untuk melakukan transaksi secara langsung melalui internet, sehingga tidak bergantung pada institusi keuangan tradisional seperti bank sentral. (Pradana Ananda Raharja, 2021).



Gambar 2. 3 Ethereum

Salah satu pendiri Ethereum, Vitalik Buterin yang saat itu baru berusia 27 tahun mengakui bahwa biayanya tinggi menjadi tantangan signifikan bagi aset digital. Ethereum pertama kali diciptakan pada Juli 2015 dengan harga 2,83 USD per koin. (Arta Pangaribuan dkk., 2023). Perbedaan fundamental antara Ethereum dan Bitcoin adalah fungsionalitasnya. Jika Bitcoin dirancang utamanya sebagai sistem uang elektronik *peer-to-peer*, Ethereum memperkenalkan konsep *blockchain* yang dapat diprogram (Ikhlasse dkk., 2022).

Ekosistem Ethereum ditopang oleh beberapa komponen teknologi yang saling bekerja sama.

1. *Ether* (ETH)

Ether (ETH) adalah aset kripto asli dari jaringan Ethereum. ETH memiliki dua fungsi krusial:

- Sebagai Aset Investasi: ETH diperdagangkan di berbagai bursa aset kripto di seluruh dunia dan merupakan *cryptocurrency* terbesar kedua berdasarkan kapitalisasi pasar setelah Bitcoin.
- Sebagai "Gas": ETH digunakan untuk membayar biaya transaksi dan layanan komputasi di dalam jaringan Ethereum. Biaya ini, yang disebut "gas," berfungsi untuk memberi insentif kepada validator agar memproses transaksi dan mengamankan jaringan, sekaligus mencegah spam transaksi di jaringan.

2. *Smart Contract*

Smart contract adalah program komputer yang disimpan di dalam jaringan *blockchain* Ethereum. Program ini otomatis menjalankan perintah sesuai dengan logika yang sudah ditulis di dalam kode, tanpa perlu campur tangan pihak ketiga (seperti bank, notaris, atau perantara lain). Kontrak ini mengeksekusi logika bisnis (misal transfer token, aturan lelang) ketika dipanggil melalui transaksi, tanpa perantara pihak ketiga. Sifatnya otomatis, terbuka (kode *bytecode*-nya publik di *blockchain*), dan tidak dapat diubah (*immutable*) setelah dideploy.

Kalau dianalogikan, *smart contract* itu seperti mesin penjual otomatis (*vending machine*). Saat kita memasukkan uang dan memilih produk, mesin akan langsung mengeluarkan barang tanpa negosiasi atau izin dari orang lain. Begitu juga dengan *smart contract*, jika syarat terpenuhi, kontrak akan langsung mengeksekusi perintahnya.

Proses pembuatan dan eksekusi *smart contract* di Ethereum seperti berikut:

- 1) **Penulisan Kode:** Pengembang menulis logika kontrak menggunakan bahasa seperti Solidity. Kode ini mendefinisikan aturan, fungsi, dan kondisi kontrak. Contoh sederhananya adalah kontrak untuk penggalangan dana (*crowdfunding*): "Jika target dana sebesar 100 ETH tercapai dalam 30 hari, kirimkan semua dana ke alamat dompet proyek. Jika tidak, kembalikan dana ke masing-masing donatur."
- 2) **Kompilasi dan Penyebaran (*Deployment*):** Ditulis dengan kode Solidity kemudian dikompilasi menjadi *bytecode* EVM. Pengembang kemudian menyebarkan (*deploy*) *bytecode* ini ke *blockchain* Ethereum melalui sebuah transaksi. Setelah disebarkan, *smart contract* mendapatkan alamat unik di *blockchain* dan menjadi bagian permanen dari buku besar yang tidak dapat diubah.
- 3) **Eksekusi (Interaksi):** Siapa pun dapat berinteraksi dengan *smart contract* dengan cara mengirimkan transaksi ke alamat kontrak tersebut. Transaksi ini akan memicu fungsi yang ada di dalam kontrak. Misalnya, seorang donatur mengirimkan 1 ETH ke alamat kontrak *crowdfunding*.

- 4) Eksekusi oleh EVM dan Konsep "Gas": Ketika sebuah fungsi dipanggil, setiap *node* di jaringan Ethereum akan mengeksekusi kode kontrak tersebut melalui EVM mereka untuk memverifikasi hasilnya. Proses komputasi ini tidak gratis. Untuk mencegah *spam* dan komputasi tak terbatas, Ethereum menggunakan konsep "Gas". Gas adalah unit yang mengukur jumlah kerja komputasi yang diperlukan untuk menjalankan sebuah operasi. Pengguna yang memicu eksekusi kontrak harus membayar biaya transaksi (*gas fee*) dalam bentuk Ether (ETH) kepada para *miner* atau validator yang memproses transaksi tersebut.
- 5) Perubahan Status di *Blockchain*: Jika eksekusi berhasil, status internal *smart contract* (misalnya, jumlah dana terkumpul) akan diperbarui. Perubahan ini kemudian dicatat secara permanen di *blockchain* Ethereum dan disebarkan ke seluruh jaringan.

3. *Ethereum Virtual Machine* (EVM)

EVM adalah lingkungan komputasi terisolasi yang tertanam di dalam setiap node Ethereum. EVM berfungsi sebagai "mesin" yang menjalankan dan mengeksekusi logika dari kontrak pintar. Setiap node di jaringan menjalankan EVM untuk memastikan semua peserta mencapai konsensus tentang hasil dari setiap eksekusi, menjaga integritas status *blockchain*.

Pada awalnya, Ethereum menggunakan mekanisme konsensus *Proof of Work* (PoW) yang mirip dengan Bitcoin, di mana *miner* bersaing memecahkan teka-teki kriptografi untuk menambahkan blok baru. Namun, PoW dinilai boros energi dan tidak efisien untuk skala besar. Oleh karena itu, pada tahun 2022 Ethereum melakukan pembaruan besar bernama *The Merge*, yang mengubah mekanisme konsensus menjadi *Proof of Stake* (PoS). Dalam PoS, validasi blok tidak lagi ditentukan oleh kekuatan komputasi, melainkan oleh jumlah ETH yang di-*stake* oleh validator. Perubahan ini membuat Ethereum menjadi lebih hemat energi, efisien, dan ramah lingkungan, sekaligus meningkatkan skalabilitas untuk mendukung transaksi dalam jumlah besar

2.2.5 *Machine Learning*

Machine Learning (ML) adalah cabang dari kecerdasan buatan (AI) dan ilmu komputer yang berfokus pada penggunaan data dan algoritma untuk meniru cara manusia belajar, dengan secara bertahap meningkatkan akurasi. ML memungkinkan sistem komputer untuk "belajar" dari data historis, mengidentifikasi pola, dan membuat keputusan dengan intervensi manusia yang minimal (Krittanawong dkk., 2020). Berbeda dengan pemrograman tradisional di mana aturan dan logika dibuat secara eksplisit oleh pengembang, model ML membangun model matematisnya sendiri berdasarkan data sampel, yang dikenal sebagai "data latih" (*training data*), untuk membuat prediksi atau keputusan.

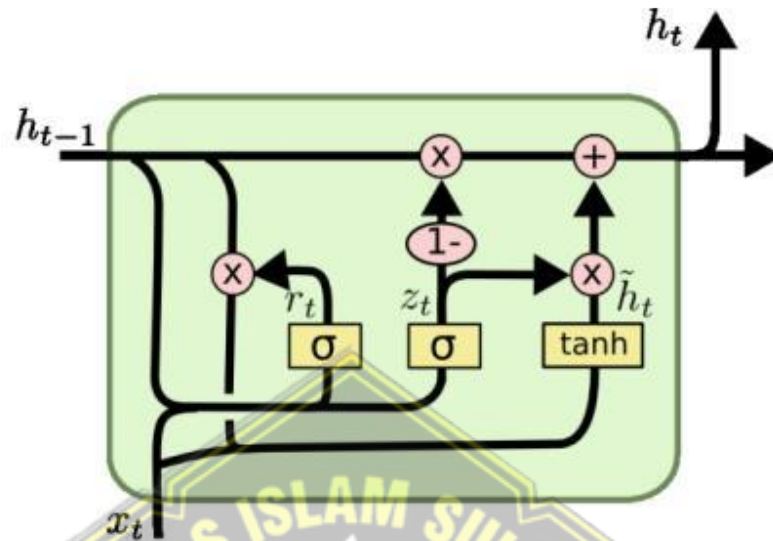
Tujuan utama dari *Machine Learning* adalah untuk mengembangkan teknik yang memungkinkan mesin dapat belajar dan beradaptasi secara otomatis. Kemampuan ini sangat penting untuk menangani masalah yang kompleks di mana solusi analitis tradisional tidak efisien atau tidak mungkin untuk dirumuskan, sebuah prinsip yang relevansinya terus meningkat seiring dengan ledakan volume data di era digital saat ini.

2.2.6 *Gated Recurrent Unit (GRU)*

Gated Recurrent Unit (GRU) adalah hasil pengembangan dari model RNN yang memiliki dua *gate*, yaitu *update gate* dan *reset gate*. (Aryati dkk., 2024b) selain memiliki 2 *gate* GRU juga memiliki 3 fungsi aktivasi. Struktur *gate* yang lebih sederhana dalam GRU memungkinkan proses pengolahan data menjadi lebih cepat, terutama saat menangani dataset yang berukuran besar. Dengan desain yang lebih efisien (Meriani & Rahmatulloh, 2024). *Gated Recurrent Unit* (GRU) memfungsikan *gate*-nya ini untuk memutuskan informasi yang dapat diteruskan menjadi *output*. Arsitektur ini dapat dilatih untuk menyimpan informasi terlebih dahulu tanpa harus menghapus informasi yang tidak berhubungan dengan prediksi yang akan dibuat (Novela Waroi & Setyanto, 2024).

Kelebihan yang dimiliki oleh metode GRU adalah proses komputasi yang lebih sederhana, namun tetap mampu mencapai akurasi yang baik. Selain itu, arsitektur GRU cukup efektif dalam mengatasi permasalahan hilangnya gradien

(*vanishing gradient*), sehingga dapat meningkatkan stabilitas dan efisiensi dalam pelatihan model (Prayogi dkk., 2024).



Gambar 2. 4 Arsitektur GRU

Dalam gambar 4.1 menampilkan arsitektur *Gated Recurrent Unit* (GRU). GRU memiliki dua gerbang utamayaitu *reset gate* (r_t) dan *update gate* (z_t), yang keduanya memodifikasi *hidden state* (h_t).

Reset gate (r_t) ini berfungsi untuk menentukan seberapa banyak informasi dari *hidden state* sebelumnya (h_{t-1}) yang akan diabaikan. Persamaannya adalah:

$$r_t = \sigma(W_r \cdot [h_{t-1}, X_t] + b_r) \dots \dots \dots (01)$$

Di sini, σ adalah fungsi aktivasi sigmoid. Jika hasil dari r mendekati 0, GRU cenderung mengabaikan informasi lama. Sebaliknya, jika hasil dari r mendekati 1, maka GRU akan mempertahankan sebagian besar informasi dari *hidden state* sebelumnya.

Tahap selanjutnya yaitu *update gate* (z). Gerbang ini mengatur seberapa banyak *hidden state* sebelumnya (h_{t-1}) yang akan diteruskan ke *hidden state* saat ini (h_t), dan seberapa banyak informasi baru dari input saat ini (x_t) akan diperbarui. Persamaannya adalah:

$$z_t = \sigma(W_z \cdot [h_{t-1}, X_t] + b_z) \dots \dots \dots (02)$$

Dalam *update gate* ini digunakan untuk mengontrol seberapa banyak *hidden state* sebelumnya yang akan dibawa ke *hidden state* saat ini. Jika hasil nilai z mendekati 1, maka GRU akan mempertahankan sebagian besar *hidden state* lama.

Namun jika hasil nilai z mendekati 0, maka GRU akan lebih banyak menggunakan informasi baru dan mengabaikan *hidden state* lama.

Tahapan selanjutnya adalah kandidat *hidden state*. Tahap ini menghitung kandidat *hidden state* baru yang potensial, dengan mempertimbangkan *input* saat ini (x_t) dan *hidden state* sebelumnya (h_{t-1}) yang telah difilter oleh *reset gate*. Persamaan sebagai berikut:

$$\tilde{h} = \tanh(W_h \cdot X_t + r * U_h \cdot h_{(t-1)} + b_h) \dots \dots \dots (03)$$

Dimana :

$\tilde{h}$: sebagai nilai sementara *hidden state* baru

r : sebagai *reset gate*

Tahapan ini digunakan untuk menghitung *hidden state* kandidat berdasarkan input baru dan *hidden state* lama yang sudah di filter oleh *reset gate*. Dimana jika hasil r_t mendekati 0 maka *hidden state* lama tidak diperhitungkan dan sebaliknya jika mendekati 1 maka *hidden state* lama akan digunakan dalam perhitungan *hidden state* kandidat.

Tahap selanjutnya adalah menghitung *hidden state* final yang akan mengkombinasikan antara *hidden state* lama dengan *hidden state* kandidat berdasarkan hasil dari nilai *update gate*. Proses ini menggunakan persamaan sebagai berikut :

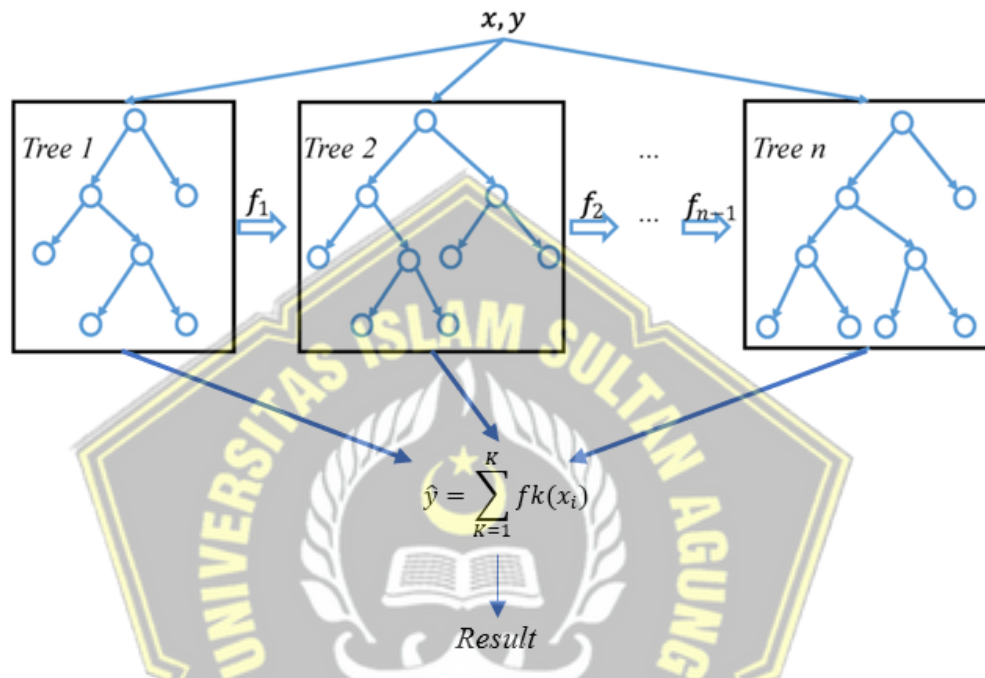
$$h = z * h_{(t-1)} + (1 - z) * \tilde{h} \dots \dots \dots (04)$$

hidden state final ini akan berperan sebagai pengontrol memori yang akan digunakan untuk menentukan seberapa banyak *hidden state* lama yang akan digunakan yang dimana apabila nilai z mendekati 1 maka *hidden state* lama lebih dominan sehingga GRU lebih banyak mengingat informasi sebelumnya. Jika nilai z mendekati 0 maka *hidden state* baru lebih dominan dan GRU akan lebih banyak belajar dari informasi baru dan jika nilai z berada diantara 0 dan 1 maka *hidden state* akan menggunakan kombinasi dari memori lama dan informasi yang baru.

2.2.7 Extreme Gradient Boosting (XGBoost)

XGBoost yang merupakan singkatan dari *Extreme Gradient Boosting* merupakan penyempurnaan dari algoritma *Gradient Boosting* yang mempunyai keunggulan dalam hal kecepatan komputasi dan akurasi prediksi. Sebagai bagian

dari metode *ensemble learning*, XGBoost membangun modelnya dengan menggabungkan sejumlah pohon keputusan (*decision tree*) dan secara iteratif mengoreksi kesalahan yang dibuat oleh gabungan pohon sebelumnya. Karena kemampuannya ini, XGBoost terbukti sangat andal untuk berbagai tugas, baik regresi maupun klasifikasi (Astutiningsih dkk., 2023).



Gambar 2. 5 Arsitektur XGboost

Pada gambar 2.2 mengilustrasikan cara kerja XGBoost (*Extreme Gradient Boosting*), yang merupakan sebuah algoritma *ensemble learning*. Ide utamanya adalah menggabungkan beberapa model sederhana (disebut *weak learners*), dalam hal ini adalah pohon keputusan (*decision trees*), untuk menciptakan satu model prediksi yang kuat.

Tahapan prosesnya sebagai berikut :

1. Input Data (x, y) Proses dimulai dengan data training Anda, yang terdiri dari fitur (x) dan target atau label yang ingin diprediksi (y).
2. Pembangunan Pohon Secara Sekuensial (*Boosting*) Tidak seperti algoritma lain yang membangun model secara paralel (seperti Random Forest), XGBoost membangun pohon keputusan secara sekuensial atau berurutan.

- *Tree 1*: Pohon pertama dibuat untuk mempelajari pola dari data asli dan membuat prediksi awal. Tentu saja, prediksi ini belum sempurna dan memiliki kesalahan (*error*).
 - *Tree 2*: Pohon kedua tidak belajar dari data asli, melainkan belajar untuk memperbaiki kesalahan yang dibuat oleh *Tree 1*.
 - *Tree n*: Proses ini berlanjut. Setiap pohon baru (f_k) dibangun untuk memperbaiki sisa kesalahan dari gabungan semua pohon sebelumnya.
- Ini adalah inti dari konsep "*boosting*".

3. Penjumlahan Hasil (*Agregasi*) Setelah semua pohon (sebanyak K) selesai dibangun, proses prediksi untuk data baru (x_i) dilakukan. Caranya adalah dengan memasukkan data tersebut ke setiap pohon untuk mendapatkan skor prediksi dari masing-masing pohon ($f_k(x_i)$). Seperti yang ditunjukkan oleh rumus :

$$\hat{y} = \sum_{k=1}^K f_k(x_i) \dots \dots \dots (05)$$

dengan:

$\hat{y}$ adalah *output* atau hasil prediksi dari model untuk data ke- i ,

K adalah jumlah total pohon (*round boosting*),

f_k merupakan fungsi dari pohon keputusan ke- k yang memetakan input x_i ke nilai prediksi

x_i adalah vektor fitur dari data ke- i .

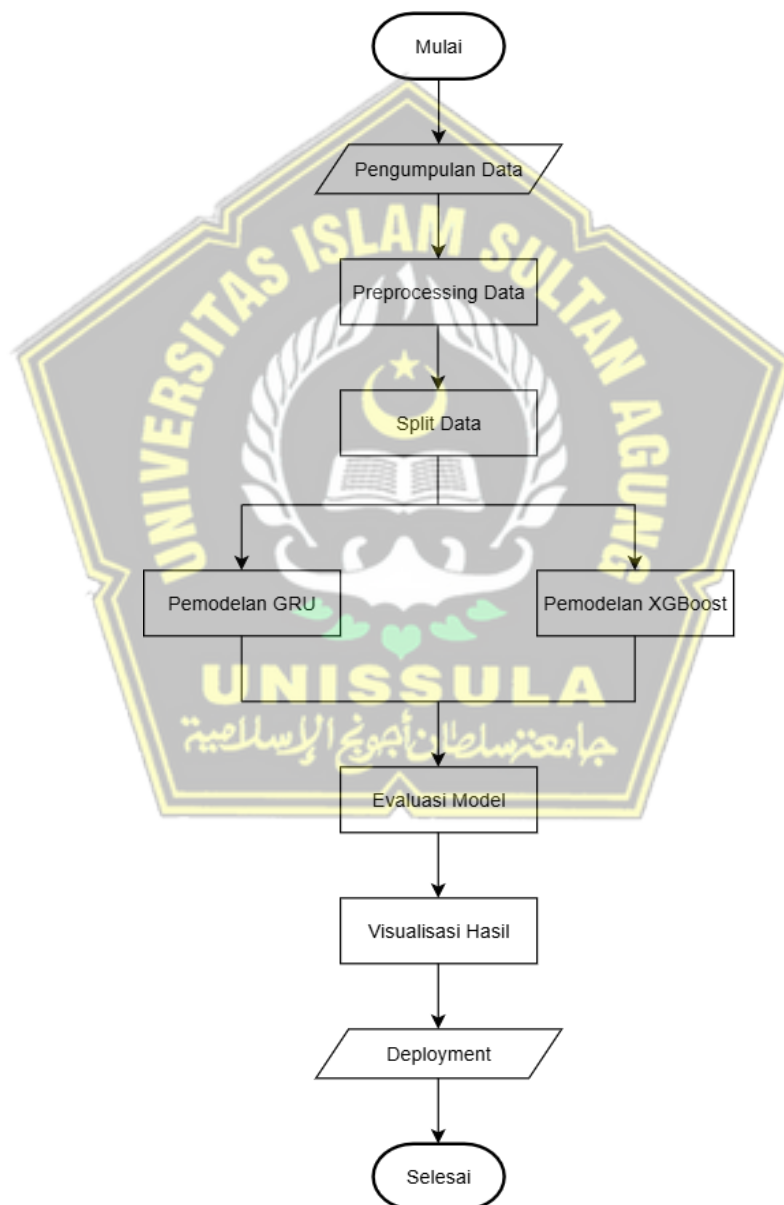
Artinya, prediksi akhir ($\hat{y}$) adalah jumlah dari skor prediksi yang diberikan oleh semua pohon (*Tree 1 + Tree 2 + ... + Tree n*).

BAB III

MATODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian digunakan untuk merencanakan, melaksanakan, dan menganalisis penelitian. Metode penelitian ini dapat membantu dalam merancang prosedur yang tepat untuk menyusun data bermanfaat, ini adalah beberapa metode penelitian yang digunakan:



Gambar 3. 1 *Flowchart* Penelitian

Berdasarkan gambar 3.1 langkah -langkah yang akan dilakukan pada tahapan ini yaitu memulai dengan pengumpulan dan pengolahan data, di tahap kedua adanya *preprocessing data* dan dilanjutkan pemodelan atau pembuatan model, tahap ke empat yaitu pengujian atau evaluasi model, kemudian perancangan sistem, lalu dilanjutkan dengan tahap deployment, dan diakhiri dengan penarikan kesimpulan sebagai hasil akhir.

3.2 Pengumpulan Data

Pada proses ini adalah pengumpulan data yang diperlukan oleh peneliti. Sumber data pada penelitian ini menggunakan data historis harga ethereum yang diperoleh dari sumber *yahoo finance* atau bisa diakses di <https://finance.yahoo.com>. Data historis yang digunakan berisi informasi berupa data *date*, *open*, *high*, *low*, *close*, dan *volume*.



ETH-USD	Date	Open	High	Low	Close (O)	Adj Close (O)	Volume
Ethereum USD	Jun 3, 2025	2,607.60	2,650.65	2,591.78	2,621.77	2,621.77	18,695,559,168
	Jun 2, 2025	2,536.18	2,615.52	2,477.29	2,607.10	2,607.10	16,744,168,824
2,628.37 +3.53%	Jun 1, 2025	2,529.40	2,547.77	2,472.54	2,538.27	2,538.27	13,287,852,482
	May 31, 2025	2,529.83	2,550.48	2,484.22	2,529.09	2,529.09	13,992,843,765
Summary	May 30, 2025	2,632.84	2,648.26	2,510.16	2,529.94	2,529.94	24,599,403,943
News	May 29, 2025	2,681.84	2,784.75	2,620.98	2,632.65	2,632.65	27,567,479,811
Chart	May 28, 2025	2,663.01	2,688.74	2,611.16	2,682.21	2,682.21	19,087,366,068
Conversations	May 27, 2025	2,564.14	2,712.29	2,512.59	2,663.07	2,663.07	28,284,200,536
Historical Data	May 26, 2025	2,551.38	2,598.57	2,530.32	2,564.14	2,564.14	14,936,610,009
Profile	May 25, 2025	2,530.89	2,553.87	2,467.88	2,551.76	2,551.76	14,556,431,364
	May 24, 2025	2,526.39	2,575.13	2,516.02	2,530.65	2,530.65	11,380,899,856
	May 23, 2025	2,664.08	2,731.22	2,504.87	2,528.44	2,528.44	30,536,501,878
	May 22, 2025	2,552.77	2,691.10	2,546.43	2,664.16	2,664.16	26,596,978,352
	May 21, 2025	2,524.16	2,614.06	2,454.54	2,552.35	2,552.35	31,952,904,125
	May 20, 2025	2,529.14	2,585.62	2,446.45	2,524.17	2,524.17	23,453,322,578

Gambar 3. 2 Data Historis Harga Ethereum USD di Yahoo Finance

Pada gambar 3.2 merupakan tampilan antarmuka dari website *yahoo finance*. Seperti yang ditunjukkan pada gambar, halaman yang diakses adalah bagian "*Historical Data*", yang menyajikan data deret waktu harian. Setiap baris pada tabel mewakili satu hari perdagangan dan terdiri dari beberapa kolom (fitur) esensial yang digunakan dalam penelitian ini, yaitu *Date*, *Open*, *High*, *Low*, dan *Volume*.

Tabel 3. 1 Data historis harga ethereum

Date	Open	High	Low	Close	Volume
	ETH-USD	ETH-USD	ETH-USD	ETH-USD	ETH-USD
01/01/2021	737.7	749.2	719.79	730.36	13652004358
02/01/2021	730.4	786.79	718.1	774.53	19740771179
03/01/2021	774.51	1006.56	771.56	975.5	45200463368
04/01/2021	977.05	1153.18	912.3	1040.23	56945985763
05/01/2021	1041.49	1129.37	986.81	1100.0	41535932781
06/01/2021	1101.0	1209.42	1064.23	1207.11	44699914188
07/01/2021	1208.07	1282.57	1167.44	1225.67	40468027280
08/01/2021	1225.96	1273.82	1076.08	1224.19	44334826666

Pada tabel 3.1 merupakan tampilan dari dataset harga historis ethereum, data tersebut memuat fitur yang akan digunakan yaitu *Date*, *Open*, *High*, *Low*, *Close*, dan *Volume*. Jumlah data yang akan digunakan yaitu sebanyak 1.642 data yang diambil dari 1 Januari 2021 sampai 1 Juli 2025. Data tersebut nantinya akan dibagi untuk data *training* 80% dan data *testing* 20%.

3.3 Preprocessing Data

Tahapan ini adalah proses mengolah data mentah menjadi data yang siap digunakan untuk pemodelan dengan tujuan untuk menghasilkan data yang lebih akurat. Data *preprocessing* meliputi tahapan sebagai berikut :

3.2.1 Data Cleaning

Data cleaning adalah salah satu tahap paling krusial dalam siklus hidup proyek *machine learning*. Tahap ini merujuk pada proses mendeteksi, memperbaiki, atau menghapus data yang korup, tidak akurat, tidak konsisten, atau tidak lengkap dari sebuah dataset. Fungsi utama dari *data cleaning* adalah untuk memastikan kualitas dan keandalan data sebelum digunakan untuk analisis atau pemodelan. Berikut merupakan tahapan pada *data cleaning* yaitu :

1. Penanganan Nilai Yang Hilang (*Missing Value*)

Penanganan nilai yang hilang adalah langkah pertama dan salah satu yang paling fundamental dalam tahap *data cleaning*. Keberadaan data yang kosong atau tidak lengkap (NaN/null) dapat menyebabkan kesalahan komputasi selama pelatihan model dan menghasilkan analisis yang bias serta prediksi

yang tidak akurat. Oleh karena itu, identifikasi dan penanganan nilai yang hilang menjadi wajib dilakukan. Untuk mengidentifikasi keberadaan nilai yang hilang secara sistematis, penelitian ini menggunakan pustaka *pandas* pada Python. Prosesnya adalah dengan menerapkan fungsi `.isnull().sum()`. Fungsi ini akan memindai seluruh *DataFrame* dan menghasilkan sebuah objek baru dengan ukuran yang sama, di mana setiap sel yang berisi nilai hilang akan ditandai sebagai `True`, dan yang berisi data akan ditandai sebagai `False`. Hasilnya adalah sebuah daftar yang menunjukkan nama setiap kolom beserta jumlah total nilai yang hilang di dalamnya.

2. Penanganan Data Duplikat

Tujuan dari tahap penanganan data duplikat ini adalah untuk memastikan bahwa setiap baris data dalam dataset bersifat unik. Keberadaan data duplikat dapat menyebabkan bias pada model *machine learning*, karena model akan memberikan bobot yang tidak semestinya pada data yang berulang. Hal ini juga dapat mengarah pada hasil evaluasi yang terlalu optimis dan tidak mencerminkan kinerja model di dunia nyata. Implementasi untuk menangani data duplikat dilakukan menggunakan pustaka *pandas* dengan fungsi `.duplicated().sum()`. Fungsi ini mengidentifikasi apakah terdapat baris data yang identik. Dalam konteks data deret waktu, ini biasanya berarti memeriksa apakah ada catatan dengan stempel waktu (*timestamp*) dan nilai-nilai fitur yang sama persis. Jika hasil dari langkah pertama menunjukkan adanya data duplikat (lebih dari 0), maka tindakan penghapusan akan dilakukan.

3.2.2 Normalisasi Data

Normalisasi data merupakan teknik krusial dalam tahap pra-pemrosesan, terutama karena setiap variabel dalam dataset umumnya memiliki skala nilai yang berbeda. Tujuan dari normalisasi adalah untuk mengubah skala nilai dari berbagai fitur ke dalam rentang yang seragam. Dalam penelitian ini, peneliti menerapkan metode *min-max scaler* untuk melakukan normalisasi, yaitu dengan mengubah nilai-nilai data ke dalam rentang tertentu umumnya antara 0 hingga 1 agar setiap variabel berada pada skala yang sebanding dan dapat dibandingkan secara

seimbang (Salsabil Muhammad dkk., 2024). Transformasi ini dilakukan dengan menggunakan rumus matematis sebagai berikut:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \dots \dots \dots (06)$$

Dimana :

X_{scaled} adalah nilai data setelah dinormalisasi.

X adalah nilai data asli.

X_{min} adalah nilai minimum dari seluruh data pada fitur tersebut.

X_{max} adalah nilai maksimum dari seluruh data pada fitur tersebut.

3.2.3 Split Data

Setelah data dibersihkan dan dinormalisasi, langkah selanjutnya adalah membagi dataset menjadi dua bagian terpisah data latih (*training data*) dan data uji (*testing data*). Tahap ini merupakan prosedur fundamental dalam pengembangan model *machine learning* untuk mengevaluasi kinerja model secara objektif. Tujuan utama dari pembagian data adalah untuk melatih model pada satu set data (data latih) dan kemudian menguji seberapa baik model tersebut dapat melakukan generalisasi pada data baru yang belum pernah dilihat sebelumnya (data uji). Untuk data deret waktu (*time series*) seperti harga Ethereum, pembagian data tidak boleh dilakukan secara acak. Pembagian harus dilakukan secara kronologis untuk menjaga dependensi temporal data. Hal ini mensimulasikan skenario dunia nyata di mana kita menggunakan data masa lalu untuk memprediksi data di masa depan. Dalam penelitian ini, rasio pembagian yang digunakan adalah 80:20, dengan rincian yaitu 80% sebagai data latih (*training data*) dan 20% sebagai data uji (*testing data*).

3.4 Pembuatan Model

Tahap pembuatan model merupakan inti dari penelitian ini karena pada bagian ini dilakukan implementasi dua pendekatan pemodelan berbeda, yaitu model GRU (*Gated Recurrent Unit*) yang berbasis deep learning dan model XGBoost (*Extreme Gradient Boosting*) yang merupakan algoritma machine learning berbasis pohon keputusan. Tujuan dari tahap ini adalah membangun kedua model dengan arsitektur dan parameter yang sesuai.

3.4.1 Pembuatan Model GRU

Pada tahap ini merupakan langkah untuk pembuatan model *Gated Recurrent Unit* (GRU) setelah proses preprocessing data. Berikut merupakan langkah pembuatan model GRU yang terlihat pada flowchart seperti gambar 3.3 :



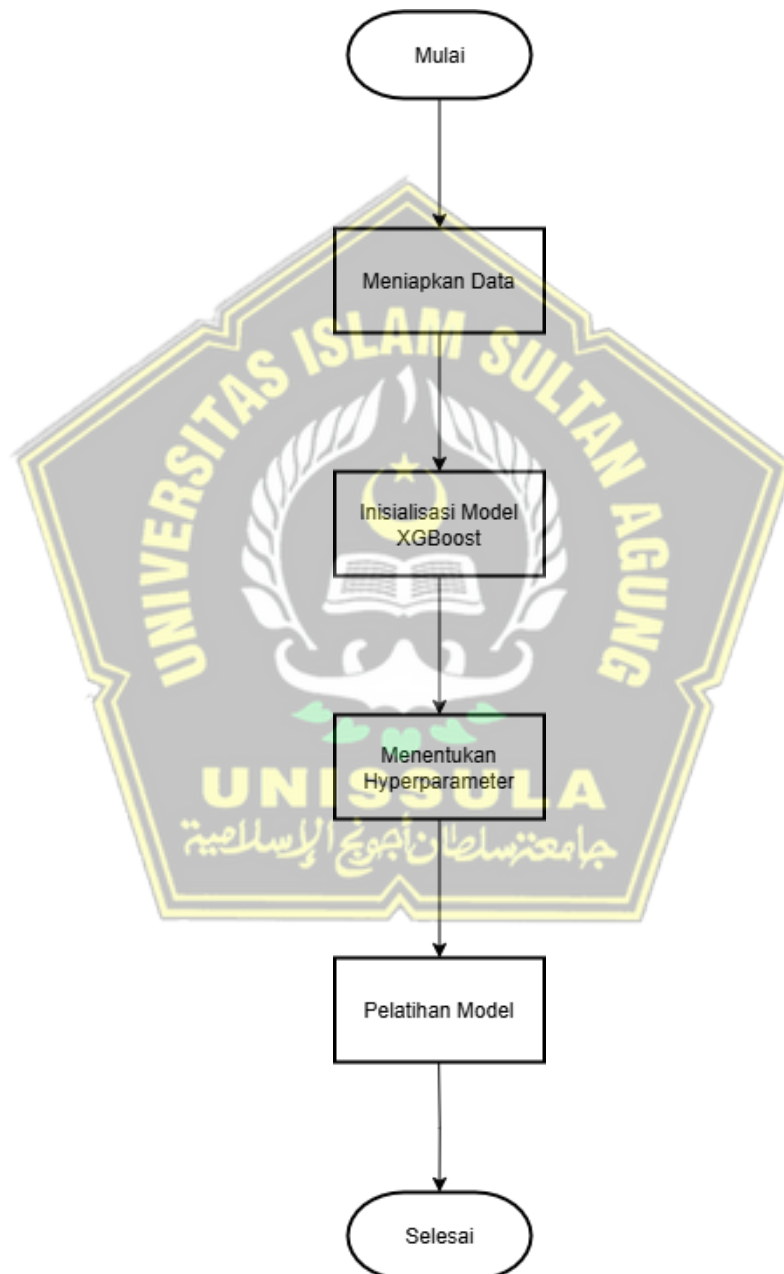
Gambar 3. 3 Flowchart pembuatan model GRU

Langkah pembuatan model GRU :

1. Membangun Arsitektur Model Arsitektur model dibangun secara sekuensial menggunakan library seperti Keras. Urutan lapisannya adalah sebagai berikut:
 - Lapisan GRU (*GRU Layer*): Sebagai lapisan inti, lapisan ini bertugas untuk mempelajari pola dan dependensi temporal dari data sekuens. Jumlah unit (neuron) di dalamnya ditentukan sebagai *hyperparameter*.
 - Lapisan *Dropout* (*Dropout Layer*): Lapisan ini sering ditambahkan setelah lapisan GRU untuk regularisasi. Tujuannya adalah mencegah *overfitting* dengan menonaktifkan sebagian neuron secara acak selama pelatihan.
 - Lapisan *Dense* (*Dense Layer*): Ini adalah lapisan output yang menghasilkan nilai prediksi akhir. Karena tujuannya adalah memprediksi satu nilai harga, lapisan ini hanya memiliki 1 unit neuron.
2. Kompilasi Model Setelah arsitektur di bangun, model perlu dikompilasi. Proses ini melibatkan penentuan dua komponen penting:
 - *Optimizer*: Algoritma yang digunakan untuk memperbarui bobot model berdasarkan data kesalahan. Adam adalah pilihan yang populer dan efisien.
 - *Loss Function*: Fungsi yang mengukur seberapa besar kesalahan model. Untuk masalah regresi seperti prediksi harga, *Mean Squared Error* (MSE) adalah pilihan yang umum digunakan.
3. Melatih Model (*Training*) Langkah terakhir adalah melakukan *training* model menggunakan data *training* yang telah disiapkan. Proses ini dilakukan dengan memanggil fungsi `.fit()`, di mana beberapa parameter ditentukan:
 - *Epochs*: Jumlah berapa kali seluruh dataset latih akan dilewatkan melalui model.
 - *Batch Size*: Jumlah sampel data yang diproses sebelum model memperbarui bobotnya.

3.4.2 Pembuatan Model XGBoost

Pada tahap ini merupakan langkah untuk pembuatan model Extreme Gradient Boosting (XGBoost) setelah proses preprocessing data. Berikut merupakan langkah pembuatan model XGBoost yang terlihat pada flowchart seperti gambar 3.4 :



Gambar 3. 4 Flowchart pembuatan model XGBoost

Langkah pembuatan model XGBoost :

1. Persiapan Data untuk XGBoost: Berbeda dengan GRU yang memproses data sekuensial, XGBoost memerlukan data dalam format tabular (baris dan kolom). Oleh karena itu, data deret waktu perlu diubah menjadi format *supervised learning*. Proses ini biasanya melibatkan pembuatan fitur (*feature engineering*) dari data masa lalu, seperti menggunakan harga penutupan hari sebelumnya ($t-1$, $t-2$, dst.) dan nilai indikator teknikal sebagai fitur input untuk memprediksi harga hari ini (t).
2. Inisialisasi Model XGBoost: Model XGBoost tidak dibangun lapis demi lapis seperti jaringan saraf. Sebagai gantinya, sebuah objek model diinisialisasi, yaitu `XGBRegressor`, karena tugasnya adalah prediksi regresi (memprediksi nilai kontinu).
3. Menentukan *Hyperparameter*: Sejumlah *hyperparameter* utama ditentukan untuk mengontrol kinerja dan kompleksitas model. Beberapa di antaranya adalah:
 - *n_estimators*: Jumlah total pohon keputusan (*decision tree*) yang akan dibangun.
 - *learning_rate*: Mengontrol laju belajar model untuk mencegah *overfitting*.
 - *max_depth*: Kedalaman maksimum dari setiap pohon untuk membatasi kompleksitas.
 - *objective*: Fungsi tujuan yang dioptimalkan, misalnya 'reg:squarederror' untuk masalah regresi.
4. Melatih model: Model dilatih menggunakan fungsi `.fit()` pada data latih. Seringkali, teknik *Early Stopping* digunakan selama pelatihan, di mana performa model dipantau pada data validasi. Pelatihan akan berhenti jika tidak ada peningkatan performa setelah beberapa iterasi tertentu untuk mendapatkan model yang optimal.

3.5 Evaluasi Model

Tahap evaluasi model merupakan bagian krusial dalam proses penelitian karena di sinilah efektivitas kedua model, GRU dan XGBoost, benar-benar diuji dalam memberikan prediksi harga ethereum. Evaluasi ini tidak hanya bertujuan untuk mengukur akurasi dari masing-masing model, tetapi juga untuk memahami bagaimana kemampuan masing-masing algoritma dalam menangani pola-pola kompleks pada data keuangan yang bersifat time-series dan sangat fluktuatif. Metrik evaluasi yang digunakan untuk mengukur seberapa baik model tersebut digunakan dalam memprediksi yaitu dengan menggunakan RMSE dan R²Score. Dari hasil pengujian ini maka akan diberikan kesimpulan untuk masing-masing model yang di uji.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \dots \dots \dots (07)$$

Dimana :

y_i : nilai aktual

$\hat{y}_i$: nilai prediksi

$\frac{1}{n}$: menghitung rata-rata dari kuadrat eror

$\sum$: menjumlahkan semua eror dari semua data

Dari perhitungan tersebut memberikan pengertian bahwa semakin kecil RMSE yang dihasilkan maka akurasi model yang digunakan semakin tinggi dan sebaliknya apabila hasil RMSE semakin besar maka nilai akurasi pada model semakin kecil. RMSE ini dipilih karena keunggulan yang dimiliki oleh RMSE dibandingkan dengan perhitungan akurasi yang lain. RMSE dinilai cocok untuk digunakan dalam pengolahan data yang besar.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \dots \dots \dots (08)$$

Dimana :

n jumlah total data

y_i adalah nilai harga aktual pada data ke- i

$\hat{y}_i$ adalah nilai harga prediksi yang dihasilkan oleh model pada data ke- i

$\bar{y}$ adalah nilai rata-rata dari seluruh harga aktual

$\sum_{i=1}^n (y_i - \hat{y}_i)^2$ adalah jumlah kuadrat selisih antara nilai aktual dan prediksi (sama seperti pada perhitungan MSE)

$\sum_{i=1}^n (y_i - \bar{y}_i)^2$ adalah jumlah kuadrat selisih total dari data aktual terhadap nilai rata-ratanya

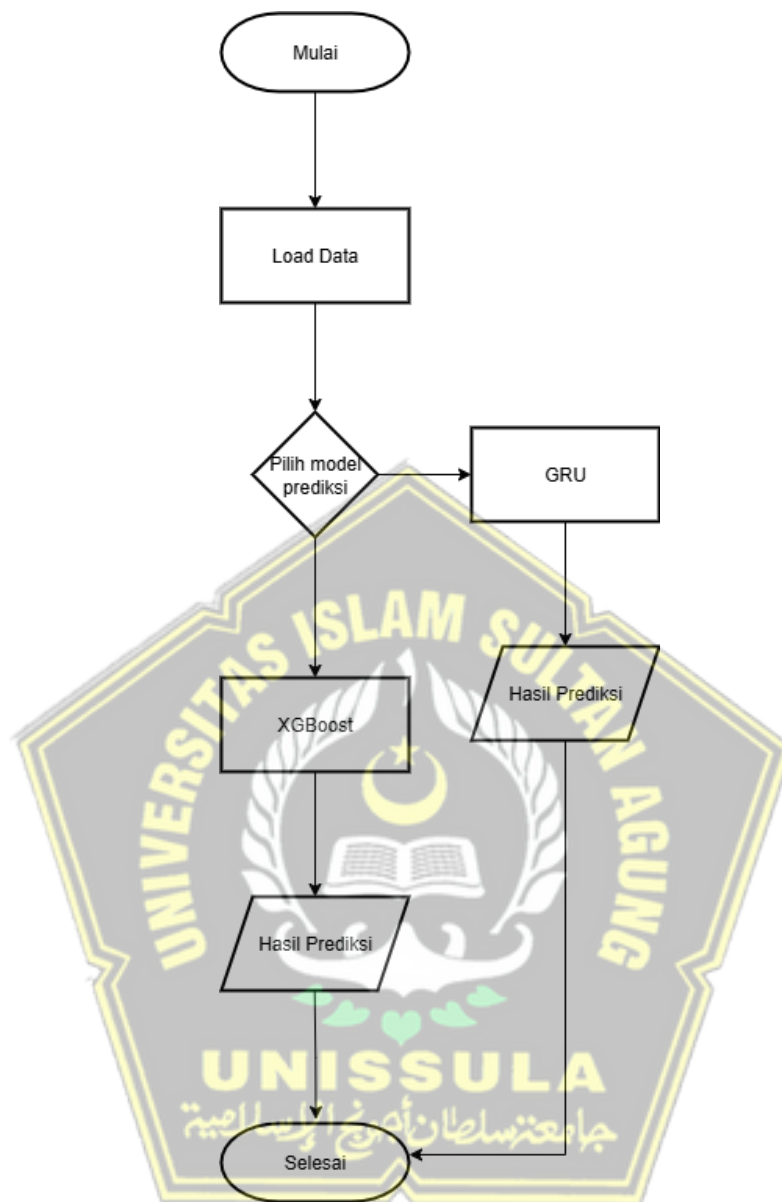
Selain RMSE, penelitian ini juga menggunakan metrik *R-squared* (R^2) atau Koefisien Determinasi untuk mengukur kecocokan model (*goodness-of-fit*). Metrik ini berfungsi untuk mengukur seberapa besar proporsi variasi dari harga aktual Ethereum yang berhasil dijelaskan oleh model prediksi. Nilai R^2 berkisar antara 0 hingga 1 (atau 0% hingga 100%), di mana skor yang lebih tinggi menunjukkan performa yang lebih baik. Secara matematis, R^2 dihitung dengan membandingkan jumlah kuadrat kesalahan prediksi model terhadap jumlah kuadrat total dari data, sehingga memberikan gambaran komprehensif tentang kemampuan prediktif model.

3.6 Visualisasi Hasil

Tahapan visualisasi hasil berfungsi sebagai analisis kualitatif untuk memberikan pemahaman yang intuitif mengenai kinerja model, yang melengkapi metrik kuantitatif seperti RMSE. Implementasinya adalah dengan membuat sebuah grafik garis menggunakan pustaka seperti Matplotlib, yang membandingkan secara langsung antara pergerakan harga aktual dengan garis prediksi historis yang dihasilkan model pada data uji. Grafik ini juga secara khusus menandai titik prediksi untuk 15 hari kedepan agar memberikan konteks visual. Dilengkapi dengan tampilan yang jelas, visualisasi ini secara efektif mempermudah interpretasi akurasi model dan membantu mengidentifikasi di mana model berhasil atau gagal.

3.7 Perancangan Sistem

Pada tahap ini peneliti akan menentukan alur kerja sistem berupa *flowchart* yang akan memberikan gambaran dari alur sistemnya. Dimana model yang dipilih nantinya akan melakukan prediksi harga ethereum, dan ada beberapa tahapan yang perlu dilakukan dalam penelitian ini, seperti yang terlihat pada gambar 3.5 :



Gambar 3. 5 Flowchart Perancangan Sistem

Berdasarkan *flowchart* yang telah dirancang seperti pada gambar 3.5 secara visual menggambarkan urutan alur kerja sistem prediksi dari awal hingga akhir, alur kerja sistem dimulai saat pengguna membuka aplikasi, yang secara otomatis memicu proses pemuatan (*load*) data harga historis Ethereum. Setelah data siap, sistem akan menampilkan antarmuka utama yang mengharuskan pengguna untuk membuat keputusan, yaitu memilih model prediksi yang akan digunakan. Pengguna dapat memilih antara model GRU atau XGBoost sebagai mesin prediksinya. Berdasarkan pilihan tersebut, sistem akan mengeksekusi algoritma yang relevan

untuk memproses data dan menghasilkan output. Langkah terakhir dari alur ini adalah penyajian hasil prediksi kepada pengguna, di mana sistem menampilkan nilai harga yang telah diramalkan sebelum proses dianggap selesai. Alur ini menunjukkan sebuah proses yang terstruktur untuk memastikan pengguna dapat dengan mudah mendapatkan hasil prediksi dari model yang diinginkan.

3.7.1 Wireframe Halaman Awal

Panel Kontrol

Gunakan panel ini untuk memilih model dan memulai prediksi harga Ethereum.

Pilih model prediksi

☐ GRU
Prediksi 1 hari

☐ XGBoost
Prediksi 1 hari

Buat Prediksi

Panduan pengguna

Sistem Prediksi Harga Ethereum (ETH-USD)

Analisis data historis dan prediksi harga ethereum menggunakan model Machine Learning.

Notifikasi Data Ethereum berhasil diunduh

Metrik Utama

Harga Terakhir (USD)	Tertinggi 24j	Terendah 24j	Volume 24j
\$0000	\$0000	\$0000	0000

Tentang model yang digunakan

Tampilkan data historis

Peringatan

Gambar 3. 6 Wireframe Halaman Awal

Wireframe pada Gambar 3.6 menyajikan rancangan antarmuka pengguna (UI) untuk halaman utama sistem, yang secara fundamental dirancang dengan struktur dua panel untuk menciptakan pengalaman pengguna yang terintegrasi dan efisien. Panel sisi kiri didedikasikan sepenuhnya sebagai "Panel Kontrol", yang berfungsi sebagai pusat kendali interaktif di mana pengguna dapat secara langsung memilih model yang akan digunakan, antara XGBoost atau GRU, dan kemudian mengeksekusi proses peramalan dengan menekan tombol "Buat Prediksi". Sementara itu, panel utama di sisi kanan berperan sebagai pusat informasi yang komprehensif, menampilkan judul sistem, area notifikasi untuk umpan balik, ringkasan "Metrik Utama" pasar terkini seperti harga terakhir, harga tertinggi, harga terendah dan volume, serta menyediakan fitur tentang informasi model dan data historis ethereum untuk menambah literatur pengguna. Kombinasi antara panel

kontrol yang fungsional dan panel informasi yang kaya ini bertujuan untuk menciptakan sebuah dashboard yang intuitif, di mana pengguna dapat melakukan prediksi sambil tetap memantau data pasar yang relevan dalam satu tampilan.

3.7.2 Wireframe Halaman Hasil Prediksi

Panel Kontrol

Gunakan panel ini untuk memilih model dan memulai prediksi harga Ethereum.

Pilih model prediksi

☐ GRU
Prediksi 1 hari

☐ XGBoost
Prediksi 1 hari

[Panduan pengguna](#)

Hasil Prediksi

Prediksi harga untuk hari berikutnya : \$0000

Analisis Grafik & Prediksi Harga

Analisis Harga Ethereum & Prediksi Model

Harga (USD)

Tanggal

Tabel Prediksi

Gambar 3. 7 Wireframe Halaman Hasil Prediksi

Wireframe pada Gambar 3.7 secara detail merancang antarmuka untuk halaman hasil prediksi, yang akan ditampilkan setelah pengguna mengeksekusi salah satu model. Tata letak halaman ini dirancang secara strategis untuk menyajikan informasi secara hierarkis dan komprehensif. Pada bagian paling atas, terdapat sebuah kotak notifikasi khusus yang berfungsi untuk menampilkan *output* utama dan paling penting, yaitu nilai numerik dari prediksi harga untuk hari berikutnya. Tepat di bawahnya, area yang lebih besar dialokasikan untuk "Analisis Grafik & Prediksi Harga", di mana sebuah grafik deret waktu akan divisualisasikan untuk memberikan konteks visual yang mendalam. Grafik ini akan memplot pergerakan harga historis dan menandai titik prediksi masa depan sehingga pengguna dapat secara intuitif menilai hasil prediksi terhadap tren sebelumnya. Sebagai pelengkap, bagian bawah halaman menyediakan ruang untuk sebuah "Tabel Prediksi" yang akan menyajikan rangkuman data secara tabular, memungkinkan perbandingan langsung antara nilai-nilai kunci seperti harga aktual terakhir dengan nilai prediksi.

Kombinasi dari ketiga komponen ini nilai tunggal, visualisasi grafik, dan rangkuman tabel bertujuan untuk menyajikan hasil prediksi dari berbagai perspektif, sehingga memastikan pengguna mendapatkan pemahaman yang jelas dan detail.

3.8 Deployment

Deployment adalah fase akhir dari penelitian ini, di mana model *machine learning* yang telah dilatih dan dievaluasi diimplementasikan menjadi sebuah sistem atau aplikasi yang fungsional. Tujuan utama dari tahap ini adalah untuk menyajikan hasil penelitian dalam bentuk sistem yang dapat diakses dan digunakan oleh pengguna akhir. Daripada hanya menyajikan kode dan angka, *deployment* menciptakan sebuah alat bantu praktis yang memungkinkan pengguna untuk berinteraksi langsung dengan model prediksi tanpa memerlukan pengetahuan teknis. Sistem ini di-deploy sebagai aplikasi web menggunakan streamlit, sebuah kerangka kerja (*framework*) *open source* berbasis python. Streamlit dipilih karena kemudahannya dalam mengubah skrip analisis data menjadi aplikasi web yang interaktif dengan cepat.

3.9 Bahasa pemrograman yang digunakan

Dalam penelitian ini bahasa pemrograman yang digunakan adalah :

1. Python 3.11.9

Python adalah bahasa pemrograman tingkat tinggi yang bersifat interpretatif, interaktif, dan berorientasi objek. Bahasa ini menjadi pilihan utama dalam penelitian ini karena ekosistemnya yang kaya akan pustaka (*library*) untuk komputasi ilmiah dan analisis data.

Dalam penelitian ini, *Python* versi 3.11.9 digunakan sebagai bahasa utama untuk seluruh proses. Penggunaan versi spesifik ini bertujuan untuk memastikan kompatibilitas antar pustaka yang digunakan serta memanfaatkan peningkatan performa yang ditawarkan. Beberapa pustaka kunci yang digunakan bersama *Python* antara lain *Pandas*, *NumPy*, *Scikit-learn*, *TensorFlow*, dan *XGBoost*.

3.10 *Software yang digunakan*

Dalam penelitian ini menggunakan beberapa *software* yang akan digunakan untuk mengolah data, pembuatan model serta evaluasi hasil. Berikut daftar *software* yang akan digunakan :

1. *Google Colaboratory*

Google Colab salah satu platform berbasis cloud dari Google ini dipilih karena menyediakan lingkungan *notebook Jupyter* yang dapat diakses langsung melalui peramban web, sehingga menghilangkan kebutuhan instalasi lokal dan yang terpenting, memberikan akses gratis ke sumber daya komputasi berkinerja tinggi. Secara spesifik, ketersediaan GPU (*Graphics Processing Unit*) di Colab menjadi fungsi krusial yang dimanfaatkan untuk secara signifikan mempercepat waktu pelatihan model *deep learning* GRU yang bersifat komputasional intensif. Selain itu, Colab juga mempermudah manajemen pustaka *python* serta memungkinkan penyimpanan dan akses kerja yang fleksibel melalui integrasinya dengan Google Drive, menjadikannya platform yang sangat efektif dan efisien untuk melaksanakan penelitian ini.

2. *Visual Studio Code*

Visual Studio Code merupakan sebuah *software* kode editor modern yang dikembangkan oleh Microsoft. Perangkat lunak ini bersifat gratis (*open-source*), dan dapat dijalankan di berbagai sistem operasi, termasuk Windows, macOS, dan Linux. Dalam penelitian ini, VS Code digunakan sebagai lingkungan pengembangan utama untuk proses pengkodean. Penggunaan VS Code dipilih karena fleksibilitasnya, dukungan ekstensi yang lengkap untuk pengembangan Python, dan integrasi terminal yang mempercepat alur kerja dari penulisan kode hingga proses *deployment* aplikasi.

3. *Streamlit*

Streamlit adalah *framework python open-source* yang dalam penelitian ini berfungsi sebagai alat utama untuk tahap *deployment*, yaitu proses mengubah model prediksi yang telah dilatih menjadi sebuah aplikasi web yang fungsional dan interaktif. Penggunaannya meliputi perancangan dan pembangunan seluruh antarmuka pengguna (UI). Secara fungsional, streamlit menjembatani

interaksi antara pengguna dan model yang berjalan di latar belakang, skripnya bertanggung jawab untuk menerima input pilihan model (GRU atau XGBoost), memuat file model yang sesuai, memicu proses prediksi, dan kemudian menyajikan outputnya secara komprehensif. Hasil yang disajikan mencakup nilai prediksi numerik dan juga visualisasi grafik yang dinamis, sehingga secara efektif mengubah hasil penelitian dari sekadar kode menjadi sebuah produk aplikasi yang dapat didemonstrasikan.

3.11 *Library yang digunakan*

Dalam penelitian ini menggunakan beberapa *library python* yang akan digunakan. Berikut daftar *library python* yang akan digunakan :

1. *yfinance*

yfinance adalah sebuah pustaka (*library*) Python *open-source* yang andal untuk mengakses dan mengunduh data pasar finansial dari *yahoo finance*. Pustaka ini berfungsi sebagai jembatan yang memungkinkan skrip python untuk mengambil data historis harga aset secara terprogram. Dalam penelitian ini, *yfinance* digunakan pada tahap pengumpulan data. Secara spesifik, pustaka ini berfungsi untuk mengunduh seluruh data historis harian untuk aset ethereum (ETH-USD) sesuai dengan rentang waktu yang telah ditentukan. Penggunaan *yfinance* memastikan proses pengumpulan data mentah yang menjadi dasar penelitian ini dapat dilakukan secara efisien dan akurat.

2. *Pandas*

Dalam *library Python*, *Pandas* merupakan pustaka esensial yang dirancang untuk keperluan analisis dan manipulasi data, dengan *DataFrame* sebagai struktur datanya yang paling utama. *DataFrame* sendiri adalah sebuah objek tabular dua dimensi berkinerja tinggi yang terdiri dari sumbu baris dan kolom yang dapat diberi label.

3. *NumPy*

NumPy (Numerical Python) adalah pustaka fundamental untuk komputasi numerik dalam *python*, yang menyediakan objek utama berupa array multidimensi berkinerja tinggi beserta serangkaian fungsi untuk operasi

matematis. Dalam penelitian ini, peran utamanya adalah pada tahap persiapan data untuk pemodelan, di mana *NumPy* digunakan untuk mentransformasikan data dari struktur pandas *DataFrame* menjadi format array numerik murni. Transformasi ini merupakan langkah krusial, terutama untuk model GRU, karena model *deep learning* memerlukan *input* dalam bentuk array dengan dimensi yang spesifik. *NumPy* digunakan untuk melakukan manipulasi bentuk (*reshaping*) pada array tersebut untuk menciptakan sekuens data yang sesuai dengan arsitektur *input* model. Selain penggunaan langsung, *NumPy* juga berfungsi sebagai fondasi bagi pustaka lain seperti *pandas* dan *TensorFlow*, sehingga perannya sangat esensial dalam keseluruhan alur kerja komputasi.

4. *Matplotlib/Seaborn*

Matplotlib/seaborn adalah pustaka visualisasi data yang fundamental dalam ekosistem *Python*, yang dalam penelitian ini memegang peranan krusial pada tahap evaluasi model dan *deployment*. Secara spesifik, pustaka ini digunakan untuk membuat grafik garis (*line chart*) yang membandingkan secara visual antara data deret waktu harga aktual ethereum dengan data harga prediksi yang dihasilkan oleh model GRU dan XGBoost. Visualisasi ini berfungsi sebagai alat analisis kualitatif yang penting untuk mengamati seberapa baik setiap model mampu menangkap tren dan volatilitas pasar. Grafik yang telah dihasilkan ini kemudian diintegrasikan dan ditampilkan di dalam antarmuka aplikasi streamlit untuk menyajikan hasil prediksi kepada pengguna akhir. Dengan demikian, *matplotlib* berfungsi sebagai jembatan untuk mengubah hasil numerik yang kompleks menjadi representasi visual yang intuitif dan mudah diinterpretasikan.

5. *TensorFlow*

TensorFlow ini digunakan untuk membangun, mengonfigurasi, dan melatih model *deep learning Gated Recurrent Unit* (GRU). Platform ini menyediakan kerangka kerja yang kuat dan terstruktur untuk merancang arsitektur jaringan saraf sekuensial lapis demi lapis, mengonfigurasi proses pembelajarannya dengan mendefinisikan *optimizer* dan *loss function* melalui

proses kompilasi, serta menjalankan seluruh siklus pelatihan pada data latih secara efisien.

6. *Scikit-learn*

Scikit-learn adalah pustaka machine learning yang sangat fundamental dan populer dalam ekosistem *Python*, yang menyediakan serangkaian alat komprehensif untuk tugas-tugas analisis data prediktif. Meskipun tidak digunakan secara langsung untuk membangun model inti GRU atau XGBoost, pustaka ini memainkan peran pendukung yang sangat penting dalam dua tahapan krusial penelitian ini. Pertama, pada tahap pra-pemrosesan data, kelas *MinMaxScaler* dari modul *sklearn.preprocessing* dimanfaatkan untuk melakukan normalisasi, yaitu mengubah skala seluruh fitur ke dalam rentang seragam. Kedua, pada tahap evaluasi model, fungsi-fungsi dari modul *sklearn.metrics*, seperti *mean_squared_error* digunakan untuk menghitung skor kuantitatif RMSE secara akurat. Dengan demikian, *Scikit-learn* berfungsi sebagai perangkat esensial yang menyediakan fungsi-fungsi standar industri untuk persiapan data dan validasi model, memastikan alur kerja penelitian ini dapat diandalkan.

7. OS (*Operating System*)

Pustaka os adalah modul bawaan (*built-in*) dalam *Python* yang berfungsi sebagai jembatan untuk berinteraksi dengan fungsionalitas sistem operasi, yang dalam penelitian ini peranannya bukan untuk analisis data, melainkan untuk manajemen file dan direktori secara terprogram. Secara spesifik, pustaka ini digunakan untuk mengelola struktur proyek dengan membuat direktori baru secara otomatis, misalnya folder untuk menyimpan model GRU dan XGBoost yang telah dilatih, sehingga memisahkan aset dari kode sumber utama. Salah satu fungsi krusial yang dimanfaatkan adalah `os.path.join()` untuk membangun path file yang portabel, memastikan skrip dapat berjalan secara konsisten di berbagai sistem operasi tanpa perlu mengubah format path secara manual. Selain itu, pustaka os juga digunakan untuk memeriksa keberadaan file atau direktori sebelum melakukan operasi tulis atau baca, yang merupakan praktik penting untuk mencegah terjadinya kesalahan program.

8. Joblib

Joblib adalah sebuah pustaka Python yang dirancang untuk menyediakan pipelining yang ringan, dengan salah satu fungsi utamanya adalah untuk menyimpan dan memuat objek Python secara efisien. Dalam alur kerja machine learning, sangat penting untuk menyimpan model yang telah dilatih agar dapat digunakan kembali tanpa perlu mengulang proses pelatihan yang memakan waktu. Dalam penelitian ini, joblib secara spesifik digunakan untuk proses serialisasi atau penyimpanan model XGBoost yang telah final ke dalam sebuah file. Pustaka ini dipilih karena sangat efisien dalam menangani objek yang berisi array NumPy berukuran besar, yang merupakan komponen inti dari model seperti XGBoost. Setelah pelatihan selesai, fungsi `joblib.dump()` digunakan untuk menyimpan model.

9. Xgboost

XGBoost (*Extreme Gradient Boosting*) adalah sebuah pustaka (*library*) *machine learning* yang bersifat *open-source* dan sangat populer. Pustaka ini menyediakan implementasi yang sangat efisien dan teroptimisasi dari algoritma *gradient boosting*, sebuah teknik *ensemble* yang membangun model secara sekuensial, di mana setiap model baru mencoba untuk memperbaiki kesalahan dari model sebelumnya.

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil Penelitian

4.1.1 Pengumpulan data

Tahap awal dalam penelitian ini adalah pengumpulan data historis yang akan digunakan untuk melatih dan menguji model prediksi. Data yang relevan dan berkualitas merupakan fondasi utama untuk membangun model *machine learning* yang akurat.

Data historis harga ethereum (ETH-USD) yang digunakan dalam penelitian ini diperoleh dari *Yahoo Finance*. Sumber ini dipilih karena merupakan penyedia data finansial publik yang tepercaya, mudah diakses, dan menyediakan data historis yang lengkap.

Data yang dikumpulkan mencakup rentang waktu dari tanggal 1 Januari 2021 hingga 1 Juli 2025. Rentang waktu yang panjang ini dipilih untuk memastikan model dapat mempelajari berbagai kondisi pasar, termasuk periode tren naik (*bull market*), tren turun (*bear market*), dan periode konsolidasi. Jumlah data yang digunakan adalah sebanyak 1.642 data yang nantinya akan digunakan sebagai data training dan testing.

Setiap baris data mewakili satu hari dan terdiri dari beberapa atribut atau fitur utama yang disajikan pada Tabel 4.1.

Tabel 4. 1 Dataset harga ethereum

Date	Open	High	Low	Close	Volume
	ETH-USD	ETH-USD	ETH-USD	ETH-USD	ETH-USD
01/01/2021	737.7	749.2	719.79	730.36	13652004358
02/01/2021	730.4	786.79	718.1	774.53	19740771179
03/01/2021	774.51	1006.56	771.56	975.5	45200463368
04/01/2021	977.05	1153.18	912.3	1040.23	56945985763
05/01/2021	1041.49	1129.37	986.81	1100.0	41535932781
06/01/2021	1101.0	1209.42	1064.23	1207.11	44699914188
07/01/2021	1208.07	1282.57	1167.44	1225.67	40468027280
08/01/2021	1225.96	1273.82	1076.08	1224.19	44334826666

Fitur-fitur yang diambil meliputi:

- *Open*: Harga pembukaan Ethereum pada hari tersebut.
- *High*: Harga tertinggi pada hari tersebut.
- *Low*: Harga terendah pada hari tersebut.
- *Close*: Harga penutupan pada hari tersebut, yang akan menjadi target prediksi utama.
- *Volume*: Jumlah total Ethereum yang diperdagangkan pada hari tersebut.

4.1.2 *Preprocessing data*

Pada tahap ini data yang digunakan akan diolah untuk membersihkan, mengubah, dan menyiapkan data mentah sebelum digunakan untuk analisis atau dimasukkan ke dalam model *machine learning*. Tujuannya adalah untuk meningkatkan kualitas data sehingga model dapat belajar secara efektif.

4.1.2.1 *Melihat Missing Value*

Tahap pertama dalam proses *data cleaning* adalah melakukan pengecekan terhadap nilai yang hilang (*missing value*) pada dataset harga Ethereum yang telah diunduh. Tujuan dari langkah ini adalah untuk mengidentifikasi apakah terdapat data yang tidak lengkap atau kosong pada setiap kolom fitur. Data yang tidak lengkap dapat mengganggu proses pelatihan dan menghasilkan model yang tidak akurat.

Pengecekan dilakukan dengan menghitung jumlah nilai kosong (NaN atau null) pada setiap kolom fitur yang digunakan, yaitu *Open*, *High*, *Low*, *Close*, dan *Volume*. Hasil dari proses pengecekan disajikan pada tabel 4.2 di bawah ini.

Tabel 4. 2 *Handling Missing Value*

<i>Price</i>	<i>Ticker</i>	<i>Handling Missing Value</i>
<i>Close</i>	ETH-USD	0
<i>High</i>	ETH-USD	0
<i>Low</i>	ETH-USD	0
<i>Open</i>	ETH-USD	0
<i>Volume</i>	ETH-USD	0

Berdasarkan Tabel 4.2, dapat dilihat bahwa hasil pengecekan menunjukkan angka 0 untuk semua kolom fitur. Hal ini mengindikasikan bahwa tidak ditemukan adanya nilai yang hilang dalam dataset penelitian. Dengan demikian, tidak

diperlukan tindakan lebih lanjut seperti imputasi atau penghapusan data, dan dataset dapat dianggap lengkap serta siap untuk tahap pra-pemrosesan.

4.1.2.2 Mengecek Duplikasi Data

Tabel 4. 3 Mengecek duplikasi data

<i>Price</i>	<i>Ticker</i>	<i>Duplicate Data</i>
<i>Close</i>	ETH-USD	0
<i>High</i>	ETH-USD	0
<i>Low</i>	ETH-USD	0
<i>Open</i>	ETH-USD	0
<i>Volume</i>	ETH-USD	0

Berdasarkan tabel 4.3 pada tahap ini proses untuk melihat apakah ada duplikasi data dari dataset yang akan digunakan. Tahap ini menghasilkan bahwa data yang akan digunakan tidak terdapat duplikasi data sehingga data tersebut bisa digunakan untuk proses selanjutnya.

4.1.2.3 Normalisasi data

Tahap ini merupakan bagian krusial dalam *preprocessing data*, terutama untuk model yang sensitif terhadap skala nilai fitur seperti GRU (*Gated Recurrent Unit*). Tujuan utama dilakukannya normalisasi adalah untuk mengubah skala nilai dari berbagai fitur ke dalam rentang yang seragam.

Dalam penelitian ini, metode *Min-Max Scaler* digunakan dalam proses normalisasi data pada model GRU. Metode ini melakukan penskalaan ulang data ke dalam rentang nilai tertentu, umumnya antara 0 dan 1. Implementasi teknisnya menggunakan fungsi *MinMaxScaler* dari *library Scikit-learn* di *Python*. Penting untuk dicatat bahwa proses fit (mempelajari nilai X_{min} dan X_{max}) hanya dilakukan pada data latih (*training data*). Hal ini dilakukan untuk mencegah kebocoran informasi (*data leakage*) dari data validasi dan data uji ke dalam model. Selanjutnya, *scaler* yang sama digunakan untuk mentransformasi data validasi dan data uji.

4.1.2.4 Pembagian data (*data splitting*)

Pada tahap ini setelah data berhasil dibersihkan dan dinormalisasi, langkah selanjutnya adalah membagi dataset menjadi dua bagian terpisah yaitu data latih

(*training data*) dan data uji (*testing data*). Tahap ini merupakan prosedur standar dalam pengembangan model *machine learning* yang bertujuan untuk mengevaluasi kinerja model secara objektif pada data yang belum pernah dilihat sebelumnya (*unseen data*).

Tabel 4. 4 Hasil *Splitting Data* Pada model GRU

Ticker	Jumlah Data	Splitting Data	
		Data Training	Data Testing
ETH-USD	1.642	1.313	329

Pada tabel 4.6 menunjukkan hasil pembagian data dengan perbandingan 80% untuk data latih dan 20% untuk data uji. Pembagian ini tidak dilakukan secara acak, melainkan berdasarkan urutan waktu. Pendekatan ini sangat penting untuk data deret waktu (*time series*) seperti harga *ethereum*, karena menjaga dependensi temporal data. Model dilatih menggunakan data dari masa lalu dan diuji kemampuannya untuk memprediksi data di masa depan, yang mensimulasikan skenario penggunaan di dunia nyata. Secara spesifik, 80% data pertama dalam urutan kronologis dialokasikan sebagai data latih, sementara 20% data sisanya yang merupakan data paling akhir digunakan sebagai data uji.

4.1.3 Pembuatan Model *Gated Recurrent Unit* (GRU) dan *Extreme Gradient Boosting* (XGBoost)

4.1.3.1 Pembuatan Model *Gated Recurrent Unit* (GRU)

Setelah data siap digunakan, langkah selanjutnya adalah membangun arsitektur model *Gated Recurrent Unit* (GRU) untuk melakukan prediksi harga *Ethereum*. Model ini dirancang menggunakan *library Keras* dari *TensorFlow*.

Tabel 4. 5 Arsitektur model GRU

MODEL	Arsitektur Model		
	Neuron	Drop out	Dense
<i>Gated Recurrent Unit</i> (GRU)	50	0.2	1

Arsitektur model GRU yang dibuat dalam penelitian ini disusun secara *sekuensial*, dimulai dengan lapisan inti GRU yang terdiri dari 50 *unit neuron*. Lapisan fundamental ini dirancang untuk memproses data deret waktu dan secara efektif menangkap pola serta dependensi temporal yang kompleks dari data sekuens

harga sebelumnya, di mana kapasitas 50 unit dipilih untuk memberikan kemampuan belajar yang memadai tanpa menjadi terlalu rumit. Untuk memastikan model tidak mengalami *overfitting* dan mampu melakukan generalisasi dengan baik pada data baru, setelah lapisan GRU ditambahkan sebuah lapisan *Dropout* dengan laju (*rate*) 0.2. Lapisan regularisasi ini bekerja dengan cara menonaktifkan secara acak 20% *neuron* dari lapisan GRU pada setiap iterasi pelatihan, sehingga memaksa jaringan untuk belajar representasi fitur yang lebih kuat. Sebagai tahap akhir, arsitektur ini ditutup dengan sebuah lapisan *Dense* yang berfungsi sebagai lapisan output. Lapisan ini hanya memiliki 1 *unit neuron*, sesuai dengan tujuan penelitian yaitu untuk menghasilkan satu nilai prediksi tunggal yang merepresentasikan harga penutupan (*Close Price*) Ethereum untuk 15 hari kedepan.

Tabel 4. 6 Hyperparameter model GRU

MODEL	Hyperparameter		
	Optimizer	Batch Size	Verbose
GATED RECURRENT UNIT (GRU)	Adam	32	1

Proses pelatihan model dilakukan dengan memanggil fungsi *fit()* pada arsitektur yang telah dikompilasi, dengan konfigurasi parameter yang telah ditentukan secara cermat. Model dilatih selama 70 *epoch*, yang berarti keseluruhan data latih dilewatkan melalui jaringan saraf sebanyak 70 kali untuk memungkinkan model belajar secara iteratif dan menyempurnakan bobot internalnya guna meminimalkan kesalahan prediksi. Dalam setiap *epoch*, proses pembelajaran ini dibagi ke dalam *batch* dengan ukuran 32, artinya pembaruan bobot model dilakukan setelah setiap 32 sampel data diproses, yang membuat prosesnya lebih efisien secara komputasi. Untuk memantau jalannya pelatihan secara langsung, parameter *verbose* diatur ke 1, yang berfungsi untuk menampilkan *progress bar* di setiap *epoch* agar kemajuan dan penurunan nilai *loss* dapat diamati secara *real-time*.

4.1.3.2 Pembuatan Model *Extreme Gradient Boosting* (XGBoost)

Setelah model GRU, model kedua yang dibangun adalah XGBoost (*Extreme Gradient Boosting*). Berbeda dengan GRU yang merupakan model *deep learning*, XGBoost adalah model *ensemble* berbasis pohon (*tree-based*) yang dikenal dengan performa dan efisiensinya.

Tabel 4. 7 Hyperparameter model XGBoost

MODEL	Hyperparameter						
	<i>objective</i>	<i>n_estimators</i>	<i>learning_rate</i>	<i>max_depth</i>	<i>sub_sample</i> dan <i>colsample_bytree</i>	<i>random_state</i>	<i>n_jobs</i>
XGBoost	<i>reg:squarederror</i>	1000	0.01	5	0.8	42	1

Model XGBoost diinisialisasi sebagai *XGBRegressor* menggunakan serangkaian *hyperparameter* untuk mengoptimalkan kinerjanya dalam tugas regresi. Fungsi tujuan (*objective*) diatur ke *reg:squarederror* untuk secara eksplisit mengarahkan model agar meminimalkan rata-rata kuadrat kesalahan. Model ini dibangun dari 1000 pohon keputusan (*n_estimators*) dengan tingkat belajar (*learning_rate*) yang diatur rendah pada 0.01 untuk memastikan proses pembelajaran yang lebih stabil dan robust. Untuk mencegah *overfitting*, kedalaman maksimum (*max_depth*) setiap pohon dibatasi hingga 5, dan regularisasi lebih lanjut diterapkan dengan hanya menggunakan 80% dari sampel data (*subsample*) dan 80% dari fitur (*colsample_bytree*) untuk setiap pohon. Terakhir, *random_state* ditetapkan ke 42 untuk memastikan bahwa hasil pelatihan dapat diulang kembali dengan konsisten, sementara *n_jobs* diatur ke 1 untuk menjalankan proses pada satu *core* prosesor. proses pelatihan dimulai menggunakan perintah `model.fit()`. Selama pelatihan, kinerja model akan terus dievaluasi tidak hanya pada data latih, tetapi juga pada data uji (*eval_set*), yang memungkinkan kita untuk memantau bagaimana model belajar dan mengidentifikasi jika terjadi *overfitting*.

4.1.4 Evaluasi Model

Setelah kedua model, yaitu GRU dan XGBoost, berhasil dilatih, tahap selanjutnya adalah melakukan evaluasi untuk mengukur kinerja dan akurasi prediksi dari masing-masing model. Tujuan utama dari tahap evaluasi ini adalah untuk mengukur seberapa dekat nilai prediksi yang dihasilkan oleh model dengan nilai harga aktual. Proses ini dilakukan dengan menggunakan data uji (*test data*),

yaitu data yang sama sekali belum pernah dilihat oleh model selama proses pelatihan. Dengan demikian, hasil evaluasi dapat memberikan gambaran yang tidak bias mengenai kemampuan generalisasi model pada data baru.

Dalam penelitian ini, metrik evaluasi yang digunakan untuk mengevaluasi kinerja model adalah *Root Mean Squared Error* (RMSE) dan *R-squared* (R^2). RMSE adalah salah satu metrik yang paling umum digunakan untuk masalah regresi karena memberikan gambaran tentang besaran rata-rata kesalahan prediksi. RMSE dihitung dengan mengambil akar kuadrat dari rata-rata selisih kuadrat antara nilai aktual dan nilai prediksi.

Selain RMSE, metrik evaluasi lain yang digunakan adalah *R-squared* (R^2) atau Koefisien Determinasi. Metrik ini berfungsi untuk mengukur seberapa baik model prediksi mampu menjelaskan variasi dari data aktual. Nilai R^2 berkisar antara 0 hingga 1, di mana nilai yang semakin mendekati 1 menunjukkan bahwa model semakin baik dalam menangkap pergerakan data. Berbeda dengan RMSE yang mengukur besaran rata-rata kesalahan, R^2 memberikan gambaran tentang persentase kecocokan (*goodness-of-fit*) model.

4.1.4.1 Evaluasi Model GRU

Tabel 4. 8 Hasil evaluasi model GRU

Model	Ticker	RMSE	R^2
GRU	ETH-USD	101.37	0.9718

Berdasarkan hasil evaluasi pada data uji, model GRU menunjukkan tingkat performa yang sangat tinggi dan akurat dalam memprediksi harga ethereum (ETH-USD). Model ini berhasil mencapai skor *Root Mean Squared Error* (RMSE) sebesar 101.37, yang mengindikasikan bahwa rata-rata kesalahan atau selisih antara harga prediksi dengan harga aktual adalah sekitar \$101.37. Kinerja yang kuat ini dipertegas oleh skor *R-squared* (R^2) yang mencapai 0.9718, yang berarti model mampu menjelaskan sekitar 97.18% dari variasi pergerakan harga aktual. Kombinasi dari nilai RMSE yang relatif rendah (m mengingat rentang harga Ethereum) dan nilai R^2 yang sangat mendekati 1 secara bersamaan menunjukkan bahwa model GRU tidak hanya akurat dalam besaran prediksi, tetapi juga sangat

andal dalam menangkap pola dan tren data, sehingga membuktikan efektivitasnya dalam prediksi ini.

4.1.4.2 Hasil evaluasi model XGBoost

Tabel 4. 9 Nilai evaluasi model XGBoost

Model	Ticker	RMSE	R ²
XGBoost	ETH-USD	435.68	0.4433

Berdasarkan hasil evaluasi model pada data uji, dapat disimpulkan bahwa performa model masih belum optimal. Hal ini ditunjukkan oleh skor *Root Mean Squared Error* (RMSE) yang mencapai \$435.68, mengindikasikan bahwa rata-rata kesalahan prediksi yang dihasilkan cukup signifikan jika dibandingkan dengan harga aktual. Kinerja ini dipertegas oleh nilai R-squared (R²) yang hanya sebesar 0.4433, yang berarti model hanya mampu menjelaskan sekitar 44.33% dari variasi pergerakan harga. Kombinasi antara tingkat kesalahan (RMSE) yang tinggi dan skor R² yang jauh dari 1 menandakan bahwa model masih kesulitan untuk menangkap pola kompleks dari data harga, sehingga tingkat akurasi masih perlu ditingkatkan.

4.1.5 Visualisasi Hasil

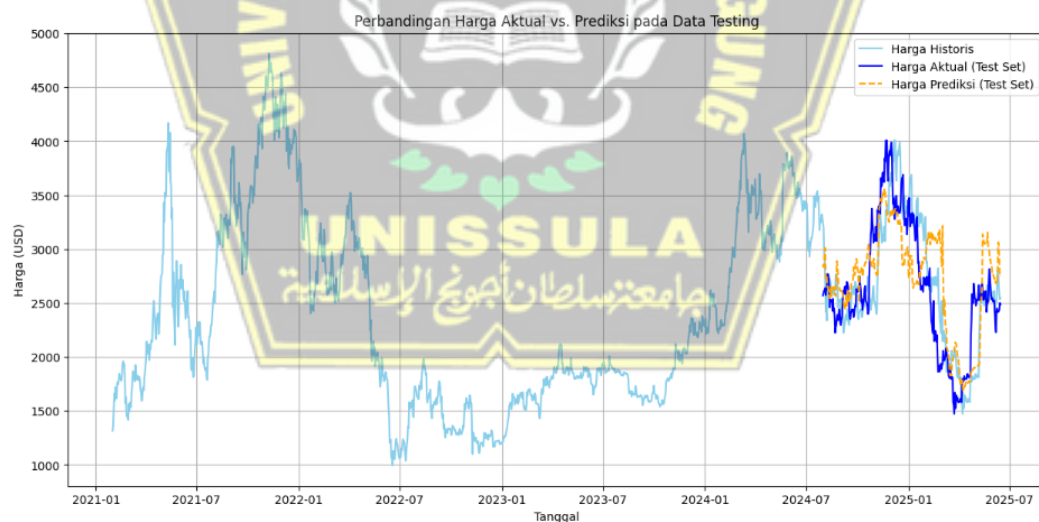
4.2.1 Visualisasi hasil model GRU



Gambar 4. 1 Hasil grafik prediksi GRU

Gambar 4.1 di atas menyajikan perbandingan visual antara harga penutupan aktual Ethereum (garis hijau) dengan harga yang diprediksi oleh model GRU (garis merah) pada data uji. Secara keseluruhan, grafik tersebut menunjukkan kemampuan model GRU yang sangat baik dalam menangkap dan mengikuti tren pergerakan harga, di mana garis prediksi (merah) secara konsisten bergerak sangat dekat mengikuti pola umum dari garis aktual (hijau), baik saat pasar naik, turun, maupun menyamping, yang mengindikasikan tingkat akurasi yang tinggi. Bahkan pada periode volatilitas tinggi dengan fluktuasi yang tajam, model mampu beradaptasi dengan baik, meskipun pada beberapa puncak dan lembah yang ekstrim dapat diamati adanya sedikit keterlambatan atau *lag* yang wajar terjadi pada model peramalan deret waktu. Visualisasi ini secara efektif mengonfirmasi bahwa model GRU telah berhasil mempelajari dependensi temporal yang kompleks dari data historis, sehingga mampu menghasilkan prediksi yang sangat mendekati pergerakan harga di dunia nyata.

4.2.2 Visualisasi hasil model XGBoost



Gambar 4. 2 Hasil grafik prediksi XGBoost

Gambar 4.3 menampilkan hasil grafik untuk memvisualisasikan kinerja model XGBoost dengan membandingkan harga historis (biru muda), harga aktual pada data uji (biru solid), dan harga prediksi (oranye putus-putus). Secara umum, garis prediksi oranye berhasil mengikuti arah tren utama dari harga aktual, menunjukkan bahwa model mampu menangkap pola pergerakan pasar. Meskipun demikian,

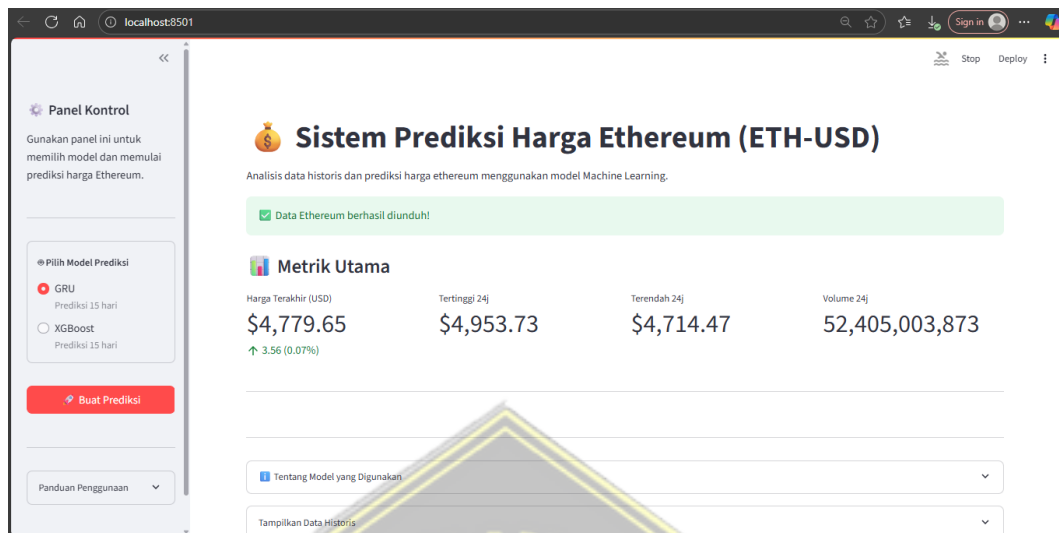
terdapat selisih yang cukup jelas antara garis prediksi dan garis aktual di banyak titik, di mana prediksi sering kali melebihi atau kurang dari harga sebenarnya, terutama saat terjadi fluktuasi tajam. Hal ini secara visual mengonfirmasi bahwa model memiliki tingkat kesalahan yang perlu diperhatikan.

4.1.6 Hasil Deployment

Sebagai tahap akhir implementasi, model prediksi GRU dan XGBoost yang telah dilatih dan dievaluasi di-*deploy* menjadi sebuah aplikasi web interaktif. Proses ini bertujuan untuk mengubah model dari sekadar kode menjadi sebuah sistem fungsional yang dapat diakses dan digunakan dengan mudah. *Deployment* dilakukan dengan menggunakan *framework Streamlit*, sebuah pustaka Python yang memungkinkan pembuatan aplikasi web untuk data sains secara cepat dan efisien.

Proses ini diawali dengan menyimpan kedua model final ke dalam file terpisah agar dapat dimuat kembali saat dibutuhkan. Selanjutnya, sebuah skrip aplikasi utama dengan nama format *app.py* dikembangkan untuk merancang seluruh antarmuka pengguna. Logika sistem ini mencakup fungsi untuk memuat model yang dipilih pengguna (GRU atau XGBoost), memproses data yang relevan, menjalankan fungsi prediksi, dan kemudian menyajikan output berupa nilai prediksi dan visualisasi kembali ke antarmuka. Akhirnya, aplikasi dijalankan secara lokal melalui perintah terminal `streamlit run app.py`, yang mengubah penelitian ini menjadi sebuah sistem praktis yang dapat diakses melalui peramban web yang berjalan di localhost. Hasil dari *deployment* ini adalah sebuah Sistem Prediksi Harga Ethereum berbasis web seperti berikut :

1. Halaman Awal



Gambar 4. 3 Halaman Awal

Pada gambar 4.3 menampilkan halaman utama dari sistem prediksi harga ethereum yang telah berhasil di-deploy menjadi sebuah aplikasi web interaktif menggunakan *framework* streamlit, sistem ini dirancang dengan antarmuka dua panel untuk fungsionalitas yang terstruktur. Pada halaman awal ini menampilkan antarmuka pengguna (UI) dari sistem prediksi harga Ethereum. Pada bagian atas terdapat judul dari sistem yang telah dibuat, kemudian di sisi kiri, "Panel Kontrol" berfungsi sebagai pusat interaksi utama, dimana pengguna dapat memilih model prediksi yang ingin digunakan antara GRU atau XGBoost lalu kemudian memicu proses prediksi dengan menekan tombol "Buat Prediksi". Panel utama di sisi kanan menyajikan informasi secara komprehensif, diawali dengan notifikasi keberhasilan pengunduhan data, lalu diikuti oleh bagian "Metrik Utama" yang menampilkan ringkasan kondisi pasar terkini seperti harga terakhir, tertinggi, terendah, dan volume perdagangan dalam 24 jam. Di bawahnya, terdapat menu tambahan untuk menampilkan detail model yang berisi informasi mengenai model yang digunakan yaitu GRU dan XGBoost. Kemudian juga terdapat menu data historis harga ethereum, serta sebuah kotak peringatan penting yang menegaskan bahwa hasil prediksi murni berdasarkan data historis dan tidak boleh dianggap sebagai saran finansial, sehingga mendorong pengguna untuk melakukan riset mandiri.

2. Halaman hasil prediksi



Gambar 4. 4 Halaman hasil prediksi dan grafik prediksi model GRU

Pada gambar 4.5 menampilkan halaman hasil akhir dari sistem setelah pengguna memilih model GRU sebagai pilihan modelnya dan menjalankan prediksi. Halaman ini secara detail menyajikan *output* prediksi dalam dua format utama yaitu numerik dan visual. Di bagian atas, pada bagian "Hasil Prediksi", sistem secara jelas menampilkan hasil prediksi untuk hari berikutnya, yaitu sebesar \$4,632.82. Di bawahnya, pada bagian "Analisis Grafik & Prediksi Harga", disajikan sebuah grafik deret waktu yang komprehensif. Grafik ini memplot perbandingan antara harga historis ethereum dengan hasil prediksi GRU pada data uji untuk 15 hari ke depan, yang memungkinkan pengguna untuk secara kualitatif menilai akurasi model di masa lalu. Bagian terpenting pada visualisasi ini adalah penanda berwarna kuning di ujung kanan grafik, yang secara spesifik merepresentasikan nilai prediksi 15 hari ke depan, sehingga memberikan konteks visual yang jelas tentang di mana posisi harga prediksi tersebut berada relatif terhadap tren pergerakan harga sebelumnya.



Gambar 4. 5 Halaman hasil prediksi dan grafik prediksi model XGBoost

Pada gambar 4.6 menampilkan proses setelah pengguna memilih model XGBoost sebagai model yang dipilih dan mengeksekusi prediksinya, sistem menyajikan halaman hasil yang merangkum *outputnya* secara numerik dan visual. Informasi utama yang ditampilkan adalah nilai prediksi harga untuk hari berikutnya, yaitu sebesar \$3,063.75. Untuk memberikan pemahaman kontekstual terhadap angka tersebut, sebuah analisis grafik disajikan di bawahnya, yang memetakan pergerakan Harga Historis dan membandingkannya dengan Prediksi XGBoost Historis untuk menunjukkan kinerja model di masa lalu. Posisi prediksi masa depan ini ditandai secara visual dengan simbol kuning pada akhir grafik, yang secara spesifik merepresentasikan nilai prediksi 15 hari ke depan. Lalu memungkinkan pengguna untuk secara cepat menilai di mana prediksi tersebut berada dalam kaitannya dengan tren harga terkini.

3. Bagian tabel prediksi

Tabel Prediksi

Tanggal	Prediksi Harga (USD)
2025-08-25 00:00:00	\$4,632.82
2025-08-26 00:00:00	\$4,547.57
2025-08-27 00:00:00	\$4,478.73
2025-08-28 00:00:00	\$4,419.08
2025-08-29 00:00:00	\$4,362.90
2025-08-30 00:00:00	\$4,308.61
2025-08-31 00:00:00	\$4,256.23
2025-09-01 00:00:00	\$4,205.71
2025-09-02 00:00:00	\$4,156.96
2025-09-03 00:00:00	\$4,109.88

 Unduh Prediksi (CSV)

Gambar 4. 6 Tabel hasil prediksi model GRU

Pada gambar 4.7 menampilkan komponen "Tabel Prediksi" pada antarmuka sistem, yang muncul setelah pengguna memilih model GRU dan menjalankan proses prediksi. Tabel ini berfungsi untuk menyajikan *output* akhir, di mana secara spesifik menunjukkan tanggal prediksinya untuk hasil prediksi 15 hari kedepan. Di bawah tabel, sistem juga menyediakan fitur tambahan melalui tombol "Unduh Prediksi (CSV)", yang memungkinkan pengguna untuk menyimpan dan mengeksport data hasil prediksi tersebut ke dalam format file CSV untuk keperluan analisis atau pencatatan lebih lanjut. Secara keseluruhan, komponen ini dirancang untuk memberikan hasil prediksi dengan cara yang mudah dipahami serta menyediakan opsi praktis untuk ekspor data.

Tabel Prediksi

Tanggal	Prediksi Harga (USD)
2025-08-25 00:00:00	\$3,063.75
2025-08-26 00:00:00	\$3,030.86
2025-08-27 00:00:00	\$3,015.77
2025-08-28 00:00:00	\$3,073.46
2025-08-29 00:00:00	\$3,049.66
2025-08-30 00:00:00	\$3,091.11
2025-08-31 00:00:00	\$3,072.87
2025-09-01 00:00:00	\$3,123.82
2025-09-02 00:00:00	\$3,218.92
2025-09-03 00:00:00	\$3,179.06

 Unduh Prediksi (CSV)

Gambar 4. 7 Hasil prediksi model XGBoost

Pada gambar 4.8 menampilkan bagian "Tabel Prediksi" dari sistem setelah pengguna memilih model XGBoost sebagai model yang sudah dipilih dan menjalankan prediksinya. Komponen ini menyajikan *output* final dari model secara ringkas, di mana tabel secara spesifik menunjukkan tanggal prediksinya untuk hasil prediksi 15 hari kedepan. Di bawah tabel, sistem menyediakan fitur tambahan melalui tombol "Unduh Prediksi (CSV)", yang memungkinkan pengguna untuk menyimpan hasil prediksi ini ke dalam format file CSV untuk keperluan pencatatan atau analisis lebih lanjut.

BAB V

KESIMPULAN

5.1 Kesimpulan

Dari hasil penelitian ini dapat memberikan beberapa kesimpulan yaitu :

1. Kedua model, GRU (*Gated Recurrent Unit*) dan XGBoost (*Extreme Gradient Boosting*), berhasil diimplementasikan dan mampu melakukan prediksi harga Ethereum untuk 15 hari kedepan dengan mengikuti tren pergerakan harga aktual secara baik, sebagaimana ditunjukkan pada visualisasi hasil prediksi pada kedua model.
2. Berdasarkan hasil evaluasi menggunakan *metrik Root Mean Squared Error* (RMSE), terdapat perbedaan performa yang jelas antara kedua model. Model GRU menghasilkan nilai RMSE sebesar 101.37 dan nilai R^2 sebesar 0.9718, sementara model XGBoost menghasilkan nilai RMSE sebesar 435.68 dan nilai R^2 sebesar 0.4433.
3. Nilai RMSE yang lebih rendah pada model GRU menunjukkan bahwa model ini memiliki tingkat akurasi yang lebih tinggi dan rata-rata kesalahan prediksi yang lebih kecil dibandingkan model XGBoost. Sementara itu hasil evaluasi dengan menggunakan nilai R^2 pada model GRU juga lebih tinggi untuk mendekati 1 dibandingkan dengan model XGBoost, yang berarti akurasi dari model GRU lebih bagus berdasarkan nilai R^2 Score. Dengan demikian, dapat disimpulkan bahwa model GRU lebih unggul dan efektif untuk memprediksi harga Ethereum dalam kerangka penelitian ini. Keunggulan ini kemungkinan besar disebabkan oleh kemampuan arsitektur GRU yang lebih baik dalam menangkap pola dan dependensi temporal jangka panjang pada data deret waktu.

5.2 Saran

Untuk meningkatkan akurasi prediksi harga ethereum dalam penelitian selanjutnya, disarankan untuk:

1. Penggunaan Model Hibrida : Menggabungkan kekuatan kedua model (misalnya, menggunakan output dari GRU sebagai salah satu fitur input untuk

model XGBoost) dapat dieksplorasi untuk melihat apakah model hibrida mampu memberikan hasil yang lebih akurat.

2. Penambahan faktor eksternal : Penelitian selanjutnya dapat menambahkan fitur-fitur eksternal seperti data sentimen dari media sosial (misalnya Twitter/X) atau portal berita, serta data *on-chain* dari *blockchain* Ethereum.
3. Optimasi Hyperparameter yang Lebih Mendalam: Melakukan pencarian *hyperparameter* yang lebih ekstensif menggunakan teknik seperti *Bayesian Optimization* atau *Randomized Search CV* dengan rentang pencarian yang lebih luas dapat membantu menemukan konfigurasi parameter yang lebih optimal untuk kedua model.



DAFTAR PUSTAKA

- Arta Pangaribuan, R., Ufrida Rahmi, N., & prima Indonesia, U. (2023). Risk and Return Investment Analysis on Bitcoin, Ethereum, Dogecoin, Litecoin, XRP and LQ45 Share After The Covid-19 Pandemic Analisis Risk Dan Return Investasi Pada Bitcoin, Ethereum, Dogecoin, Litecoin, XRP Dan Saham Saham LQ45 Setelah Pandemi Covid-19. Dalam *Management Studies and Entrepreneurship Journal* (Vol. 4, Nomor 3). <http://journal.yrpiiku.com/index.php/msej>
- Astutiningsih, T., Saputro, D. R. S., & Sutanto. (2023). Optimasi Algoritme Xtreme Gradient Boosting (XGBoost) pada Harga Saham PT. United Tractors Tbk. *SPECTA Journal of Technology*, 7(3), 632–641. <https://doi.org/10.35718/specta.v7i3.1031>
- Dhea Larasati, K., & Primandari, A. H. (2021). FORECASTING BITCOIN PRICE BASED ON BLOCKCHAIN INFORMATION USING LONG-SHORT TERM METHOD. Dalam *Parameter: Journal of Statistics* (Vol. 1). www.blockchain.com
- Futri, D. A. (2024). Feby Alfaraby 3) , Moh Azral Fathurrazaq 4) , Fikri Maulana 5) , Somantri 6) , Gina Purnama Insany 7) 1, 2,3,4,5,6,7) Fakultas Teknik Komputer dan Desain. Dalam *Universitas Nusa Putra*.
- Gupta, B. (2022). Understanding Blockchain Technology: How It Works and What It Can Do. *Metaverse Basic and Applied Research*, 1, 18. <https://doi.org/10.56294/mr202218>
- Ikhlasse, H., Benjamin, D., Vincent, C., & Hicham, M. (2022). Recent implications towards sustainable and energy efficient AI and big data implementations in cloud-fog systems: A newsworthy inquiry. Dalam *Journal of King Saud University - Computer and Information Sciences* (Vol. 34, Nomor 10, hlm. 8867–8887). King Saud bin Abdulaziz University. <https://doi.org/10.1016/j.jksuci.2021.11.002>
- Krittanawong, C., Virk, H. U. H., Bangalore, S., Wang, Z., Johnson, K. W., Pinotti, R., Zhang, H. J., Kaplin, S., Narasimhan, B., Kitai, T., Baber, U., Halperin, J. L., & Tang, W. H. W. (2020). Machine learning prediction in cardiovascular diseases: a meta-analysis. *Scientific Reports*, 10(1). <https://doi.org/10.1038/s41598-020-72685-1>
- Kushwaha, S. S., Joshi, S., Singh, D., Kaur, M., & Lee, H. N. (2022). Ethereum Smart Contract Analysis Tools: A Systematic Review. Dalam *IEEE Access* (Vol. 10, hlm. 57037–57062). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ACCESS.2022.3169902>
- Kusuma, F., Ahsan, M., Teknik Informatika, J., Sains dan Teknologi, F., PGRI Kanjuruhan Malang Jl Supriadi No, U. S., Malang, K., & Timur, J. (2021). *Jurnal Informatika dan Rekayasa Perangkat Lunak Prediksi Jumlah Penduduk Miskin Indonesia menggunakan Metode Single Moving Average dan Double Moving Average*. 3(2), 105–109.
- Meriani, A. P., & Rahmatulloh, A. (2024). Perbandingan Gated Recurrent Unit (GRU) dan algoritma Long Short Term Memory (LSTM) linear refression dalam prediksi harga emas menggunakan model time series. *Jurnal*

- Informatika dan Teknik Elektro Terapan*, 12(1).
<https://doi.org/10.23960/jitet.v12i1.3808>
- Meri Aryati, N. W., Wiguna, I. K. A. G., Putri, N. W. S., Widiartha, I. K. K., & Ginantra, N. L. W. S. R. (2024). Komparasi Metode LSTM dan GRU dalam Memprediksi Harga Saham. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 8(2), 1131. <https://doi.org/10.30865/mib.v8i2.7342>
- Moch Farryz Rizkilloh, & Sri Widiyanesti. (2022). Prediksi Harga Cryptocurrency Menggunakan Algoritma Long Short Term Memory (LSTM). *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 6(1), 25–31. <https://doi.org/10.29207/resti.v6i1.3630>
- Novela Waroi, E., & Setyanto, A. (t.t.). *408 Elsin Novela Prediksi Harga Laptop Menggunakan Algoritma GRU dan BILSTM*.
- Nusaiba Yulisa, P., Al Haris, M., & Rismawati Arum, P. (2023). Peramalan Nilai Ekspor Migas di Indonesia dengan Model Long Short Term Memory (LSTM) dan Gated Recurrent Unit (GRU). Dalam *J Statistika* (Vol. 16, Nomor 1).
- Pourpourides, P. M. (2025). Long-term nexus of macroeconomic and financial fundamentals with cryptocurrencies. *Frontiers in Blockchain*, 8. <https://doi.org/10.3389/fbloc.2025.1550720>
- Pradana Ananda Raharja, P. (2021). Prediksi Harga Ethereum Menggunakan Metode Vector Autoregressive. *Journal of Informatics, Information System, Software Engineering and Applications*, 3(2), 71–79. <https://doi.org/10.20895/INISTA.V3I2>
- Prayogi, K., Gata, W., & Kussanti, D. P. (t.t.). *Prediksi Harga Saham Bank Central Asia Menggunakan Algoritma Deep Learning GRU*.
- Putri, R. N. M. H., Primasari, N. S., & Khusnah, H. (2022). Return Analisis Teknikal Moving Average, Bollinger Band, dan Relative Strength Index pada Cryptocurrency. *Jurnal Ilmiah Akuntansi dan Keuangan*, 11(1), 21–30. <https://doi.org/10.32639/jiak.v11i1.25>
- Salsabil Muhammad, Azizah Lutvi Nuril, & Ade Eviyanti. (2024). Implementasi Data Mining Dalam Melakukan Prediksi Penyakit Diabetes Menggunakan Metode Random Forest Dan Xgboost. *Jurnal Ilmiah Komputasi*, 23(1). <https://doi.org/10.32409/jikstik.23.1.3507>
- Satria Andromeda, R., & Anisa Sri Winarsih, N. (2025). *Sistemasi: Jurnal Sistem Informasi Perbandingan Kinerja Metode LSTM dan GRU dalam Prediksi Harga Close Cryptocurrency Performance Comparison of LSTM and GRU Methods in Predicting Cryptocurrency Closing Prices*. <http://sistemasi.ftik.unisi.ac.id>
- Silalahi, R. N., & Muljono, M. (2024). Perbandingan Kinerja Metode Linear Regression, LSTM dan GRU Untuk Prediksi Harga Penutupan Saham Coca-Cola. *Komputika : Jurnal Sistem Komputer*, 13(2), 201–211. <https://doi.org/10.34010/komputika.v13i2.12265>