

PREDIKSI HARGA MATA UANG KRIPTO MENGGUNAKAN ALGORITMA *TEMPORAL CONVOLUTIONAL NETWORK*

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 :
Aris Munandar
NIM 32602100034

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

FINAL PROJECT
CRYPTOCURRENCY PRICE PREDICTION USING THE
TEMPORAL CONVOLUTIONAL NETWORK ALGORITHM

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



Arranged By :

Aris Munandar

NIM 32602100034

MAJORING OF INFORMATIC ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG
2025

**LEMBAR PENGESAHAN
TUGAS AKHIR**

**PREDIKSI HARGA MATA UANG KRIPTO MENGGUNAKAN
ALGORITMA *TEMPORAL CONVOLUTIONAL NETWORK***

**ARIS MUNANDAR
NIM 32602100034**

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 11 Juni 2025

TIM PENGUJI UJIAN SARJANA :

Bagus Satrio Waluyo Poetro,
S.Kom., M.Cs

NIK. 210616051

(Penguji 1)

Mustafa, ST., MM., M.Kom

NIK. 2106100040

(Penguji 2)

Ghufron, ST., M.Kom

NIK. 210622056

(Pembimbing)

13 Juni 2025

13 Juni 2025

17 Juni 2025

Semarang, 19 Juni 2025

Mengetahui,

Kaprodi Teknik Informatika

Universitas Islam Sultan Agung

Moch. Taufik, ST., MIT

NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Aris Munandar

NIM : 32602100034

Judul Tugas Akhir : Prediksi Harga Mata Uang Kripto Menggunakan Algoritma
Temporal Convolutional Network

Bahwa dengan 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, 11 Juni 2025

Yang menyatakan,



Aris Munandar

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Aris Munandar
NIM : 32602100034
Program Studi : Teknik Informatika
Fakultas : Teknologi industri
Alamat Asal : Semarang

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Prediksi Harga Mata Uang Kripto Menggunakan Algoritma *Temporal Convolutional Network*. Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan di internet 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, 11 Juni 2025

Yang menyatakan,



Aris Munandar

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 “Prediksi Harga Mata Uang Kripto Menggunakan Algoritma *Temporal Convolutional Network*.” 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 Riswad dan Ibu Dairoh yang selalu memberikan segala doa, dukungan, dan motivasi dengan penuh limpahan kasih sayang sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik.
6. Kakak penulis, Darwanto, Suhari, Rustopa, Sakuri, Sakup, Nurhayati, Sukmawati, yang selalu memberikan dukungan moril sehingga penulis mampu menyelesaikan studinya.
7. Untuk Diri Sendiri yang tidak pernah menyerah dan berhenti mencoba sesulit apapun rintangan kuliah dan selama proses penyusunan laporan ini.
8. Rekan seperjuangan, yang telah memberikan dukungan moral, motivasi, serta semangat dalam proses penyelesaian tugas akhir ini.
9. Dan kepada seluruh pihak yang tidak dapat saya sebutkan satu persatu.

Semarang, 11 Juni 2025

Aris Munandar

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN TUGAS AKHIR	ii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
ABSTRAK	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Pembatasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori.....	7
2.2.1 <i>Cryptocurrency</i>	7
2.2.2 <i>Ethereum</i>	7
2.2.3 <i>Machine Learning</i>	8
2.2.4 <i>Deep learning</i>	8
2.2.5 <i>Time Series</i>	8
2.2.6 <i>Temporal Convolutional Network (TCN)</i>	9
2.2.7 <i>One Dimensional Convolutional Network</i>	10
2.2.8 <i>Causal convolutions</i>	11
2.2.9 <i>Dilated convolutions</i>	12
BAB III METODOLOGI PENELITIAN	15

3.1	Metode Penilitan	15
3.1.1	Studi Literatur	15
3.1.2	Pengumpulan Data	16
3.1.3	Pra-pemrosesan Data.....	16
3.1.3.1	<i>Data Cleaning</i>	16
3.1.3.2	Normalisasi Data.....	16
3.1.3.3	Pembagian Data	17
3.1.4	Perancangan Model.....	18
3.2	Analisis kebutuhan.....	22
3.3	Perancangan sistem	25
BAB IV HASIL DAN ANALISIS PENELITIAN.....		26
4.1	Hasil penelitian.....	26
4.1.1	Pengumpulan data.....	26
4.1.2	Pra-pemrosesan Data.....	28
4.1.2.1	<i>Data Cleaning</i>	28
4.1.2.2	Pembagian data	29
4.1.2.3	Normalisasi data.....	29
4.1.3	Perancangan model TCN	31
4.1.4	Denormalisasi.....	33
4.1.5	Pengujian model.....	34
4.1.6	Evaluasi model.....	42
4.2	Hasil Prediksi	43
4.3	Dashboard Streamlit.....	45
BAB V KESIMPULAN DAN SARAN		49
5.1	Kesimpulan	49
5.2	Saran.....	49
DAFTAR PUSTAKA		

DAFTAR TABEL

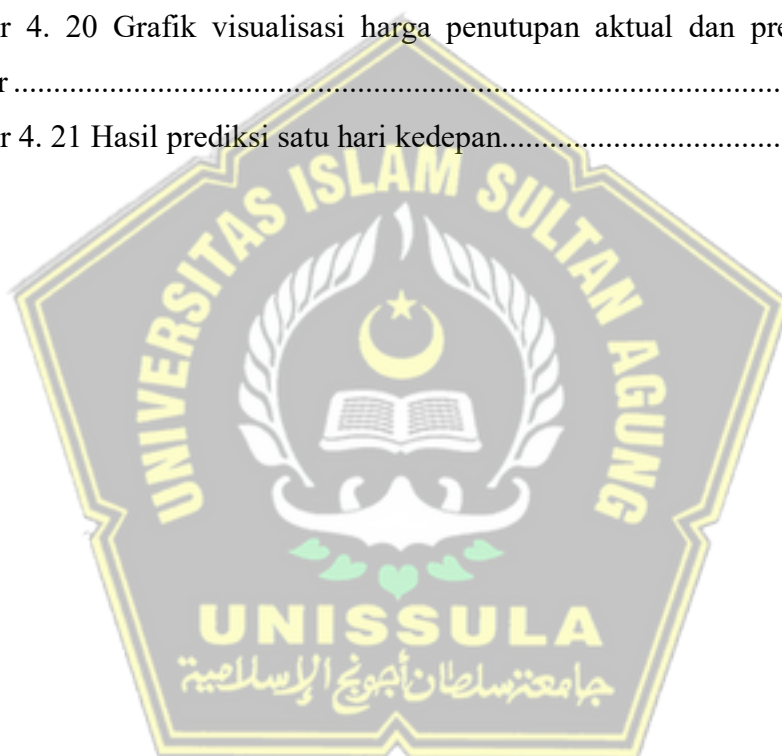
Tabel 3. 1 Skala nilai akurasi MAPE	21
Tabel 4. 1 Data Ethereum.....	26
Tabel 4. 2 Statistik deskriptif harga penutupan Ethereum	27
Tabel 4. 3 Hasil <i>missinv value</i>	28
Tabel 4. 4 Hasil data duplikat.....	29
Tabel 4. 6 Tabel jumlah data latih dan uji	29
Tabel 4. 5 Tabel harga penutupan Ethereum aktual dan <i>scale</i>	30
Tabel 4. 7 Tabel pengujian model	33
Tabel 4. 8 Tabel hasil evaluasi	42
Tabel 4. 9 Hasil prediksi 1 hari selanjutnya.....	45



DAFTAR GAMBAR

Gambar 2. 1 Model TCN (Adel Musallam dkk., 2019)	9
Gambar 2. 2 Proses <i>1D convolution</i> (Dai dkk., 2023)	10
Gambar 2. 3 <i>Non Causal convolutions</i> (Lara-Benítez dkk., 2020).....	11
Gambar 2. 4 <i>Causal convolutions</i> (Lara-Benítez dkk., 2020).....	11
Gambar 2. 5 <i>Zero padding</i> (Mo dkk., 2023)	12
Gambar 2. 6 <i>Dilated Convolution</i> (Lara-Benítez dkk., 2020).....	13
Gambar 2. 7 <i>Dilated causal convolution</i> (Hao & Liu, 2024).....	14
Gambar 3. 1 Tahapan penelitian.....	15
Gambar 3. 2 <i>Flowchart</i> model TCN	18
Gambar 3. 3 <i>Flowchart</i> sistem.....	25
Gambar 4. 1 Grafik harga penutupan harian ethereum 2018-2025.....	27
Gambar 4. 2 Model sequential TCN	31
Gambar 4. 3 Ringkasan arsitektur model sequential TCN.....	32
Gambar 4. 4 Pelatihan TCN <i>patience</i> 10, <i>epochs</i> 30, dan <i>batch size</i> 32.....	34
Gambar 4. 5 Grafik loss pelatihan TCN <i>patience</i> 10, <i>epochs</i> 30, dan <i>batch size</i> 32	35
Gambar 4. 6 Pelatihan TCN <i>patience</i> 10, <i>epochs</i> 45, dan <i>batch size</i> 32.....	36
Gambar 4. 7 Grafik loss pelatihan TCN <i>patience</i> 10, <i>epochs</i> 45, dan <i>batch size</i> 32	36
Gambar 4. 8 Pelatihan TCN <i>patience</i> 10, <i>epochs</i> 60, dan <i>batch size</i> 32.....	37
Gambar 4. 9 Grafik loss pelatihan TCN <i>patience</i> 10, <i>epochs</i> 60, dan <i>batch size</i> 32	37
Gambar 4. 10 Pelatihan TCN <i>patience</i> 15, <i>epochs</i> 90, dan <i>batch size</i> 64.....	38
Gambar 4. 11 Grafik loss pelatihan TCN <i>patience</i> 15, <i>epochs</i> 90, dan <i>batch size</i> 64	39
Gambar 4. 12 Pelatihan TCN <i>patience</i> 15, <i>epochs</i> 100, dan <i>batch size</i> 64.....	40
Gambar 4. 13 Grafik loss pelatihan TCN <i>patience</i> 15, <i>epochs</i> 100, dan <i>batch size</i> 64.....	40
Gambar 4. 14 Pelatihan TCN <i>patience</i> 20, <i>epochs</i> 120, dan <i>batch size</i> 64.....	41

Gambar 4. 15 Grafik loss pelatihan TCN <i>patience</i> 15, <i>epochs</i> 120, dan <i>batch size</i> 64.....	41
Gambar 4. 16 Grafik perbandingan aktual dan prediksi harga penutupan harian ethereum.....	44
Gambar 4. 17 Tampilan utama sistem.....	45
Gambar 4. 18 Tampilan data 30 hari terakhir.....	46
Gambar 4. 19 Tabel perbandingan harga penutupan aktual dan prediksi 30 hari terakhir.....	47
Gambar 4. 20 Grafik visualisasi harga penutupan aktual dan prediksi 30 hari terakhir.....	47
Gambar 4. 21 Hasil prediksi satu hari kedepan.....	48



ABSTRAK

Perkembangan pesat mata uang kripto sebagai aset digital telah menimbulkan kebutuhan akan sistem prediksi harga yang akurat guna mendukung pengambilan keputusan investasi. Untuk membangun sistem prediksi yang andal, diperlukan model yang mampu menangani karakteristik data deret waktu secara efektif. *Temporal Convolutional Network* (TCN) merupakan arsitektur jaringan saraf mendalam yang dirancang khusus untuk pemrosesan data deret waktu dengan memanfaatkan *causal convolution* dan *dilated convolution*, sehingga mampu menangkap pola jangka panjang tanpa kehilangan informasi urutan waktu. Penelitian ini menerapkan algoritma TCN untuk memprediksi harga penutupan harian Ethereum berdasarkan data historis. Evaluasi performa model dilakukan menggunakan metrik *Mean Absolute Error* (MAE) dan *Mean Absolute Percentage Error* (MAPE). Hasil evaluasi menunjukkan bahwa model TCN mampu memberikan performa prediksi yang baik dengan tingkat kesalahan yang relatif rendah dengan MAE sebesar 75,70 dan MAPE sebesar 2,67%. Visualisasi hasil prediksi juga memperlihatkan bahwa TCN dapat menangkap pola tren harga Ethereum secara akurat. Temuan ini menunjukkan bahwa TCN memiliki potensi sebagai pendekatan yang efektif dalam memodelkan dan memprediksi pergerakan harga mata uang kripto, khususnya Ethereum.

Kata kunci: *Cryptocurrency, Ethereum, Dilated causal convolution, Temporal Convolutional Network*

ABSTRACT

The rapid development of cryptocurrencies as digital assets has led to the need for accurate price prediction systems to support investment decision-making. To build a reliable prediction system, a model that can effectively handle the characteristics of time series data is required. Temporal Convolutional Network (TCN) is a deep neural network architecture specifically designed for time series data processing by utilizing causal convolution and dilated convolution, thus capturing long-term patterns without losing time sequence information. This research applies the TCN algorithm to predict the daily closing price of Ethereum based on historical data. Model performance evaluation is performed using the Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) metrics. The evaluation results show that the TCN model is able to provide good prediction performance with a relatively low error rate with an MAE of 75.70 and a MAPE of 2.67%. Visualization of the prediction results also shows that TCN can accurately capture the trend pattern of the Ethereum price. These findings suggest that TCN has the potential to be an effective approach in modeling and predicting cryptocurrency price movements, especially Ethereum.

Keywords: Cryptocurrency, Ethereum, Dilated causal convolution, Temporal Convolutional Network

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi informasi telah menyebabkan dampak yang cukup berpengaruh terhadap berbagai aspek kehidupan, termasuk sektor keuangan (Aysa, 2021). Transformasi digital yang terjadi secara masif telah mendorong terciptanya berbagai inovasi keuangan berbasis teknologi, salah satunya adalah kehadiran *cryptocurrency* atau mata uang kripto. *Cryptocurrency* merupakan bentuk mata uang digital atau virtual yang dirancang sebagai alat tukar yang menggunakan kriptografi untuk menjamin keamanan transaksi dan memverifikasi transfer aset secara terdesentralisasi (Atmaja & Hakim, 2022). Tidak seperti mata uang konvensional yang dikendalikan oleh bank sentral, mata uang kripto beroperasi melalui teknologi *blockchain* yang bersifat terbuka dan tersebar, memungkinkan transparansi dan kepercayaan tanpa memerlukan pihak ketiga (Zainudin Hasan dkk., 2024).

Mata uang kripto telah berkembang menjadi lebih dari sekadar alat tukar, tetapi juga sebagai instrumen investasi yang populer di kalangan masyarakat global (Abdurohim & Irfan, 2024). Ethereum merupakan salah satu jenis kripto dengan kapitalisasi pasar terbesar, yang telah menunjukkan lonjakan nilai signifikan dalam periode tertentu, sehingga menarik perhatian investor, baik individu maupun institusional (Rolando dkk., 2024). Dalam konteks investasi, mata uang kripto dianggap sebagai alternatif terhadap aset-aset konvensional seperti saham, obligasi, dan emas, karena menawarkan potensi keuntungan yang tinggi, meskipun dengan risiko yang tidak kalah besar akibat fluktuasi harga yang ekstrem (Duta Pratama & Rahmawati Gustini, 2023).

Fluktuasi harga yang tinggi pada mata uang kripto menjadi salah satu hambatan utama dalam pengambilan keputusan investasi (Wardhana, 2024). Untuk mengatasi hal tersebut, dibutuhkan sistem yang mampu menangkap karakteristik data yang kompleks dan bersifat nonlinier guna menghasilkan prediksi harga yang lebih presisi dan andal. Dalam hal ini, teknologi kecerdasan buatan dan *deep learning* yang merupakan cabang dari *machine learning* memiliki peran krusial,

khususnya dalam pemodelan prediksi berbasis deret waktu atau *time series forecasting* (Setiyani dkk., 2023).

Salah satu algoritma pembelajaran mesin yang dinilai efektif dalam menangani data deret waktu dengan volatilitas tinggi adalah *Temporal Convolutional Network* (TCN). TCN merupakan pengembangan dari arsitektur *Convolutional Neural Network* (CNN) yang dirancang khusus untuk menyelesaikan permasalahan deret waktu (Zuo dkk., 2023). Keunggulan utama TCN terletak pada strukturnya yang unik, yaitu *dilated convolution* dan *causal convolution* yang bekerja bersama untuk mengekstraksi pola jangka panjang dari data historis secara efisien (Aisyah dkk., 2024).

Berdasarkan latar belakang tersebut, penelitian ini difokuskan pada penerapan algoritma *Temporal Convolutional Network* untuk memprediksi harga penutupan harian Ethereum. Melalui penelitian ini, diharapkan dapat diketahui seberapa baik performa TCN dalam menangkap pola pergerakan harga kripto dan menghasilkan prediksi yang akurat, sehingga dapat memberikan kontribusi terhadap pengembangan sistem prediksi berbasis kecerdasan buatan di bidang *Cryptocurrency*.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah dalam penelitian ini adalah :

1. Implementasi model menggunakan algoritma *Temporal Convolution Networks* dalam memprediksi harga mata uang kripto Ethereum.
2. Bagaimana performa algoritma *Temporal Convolutional* dalam memprediksi harga mata uang kripto Ethereum.

1.3 Pembatasan Masalah

Berdasarkan rumusan masalah tersebut maka batasan masalah yang akan dibuat adalah sebagai berikut:

1. Prediksi difokuskan pada jenis mata uang kripto Ethereum sebagai objek utama penelitian.
2. Data yang digunakan untuk prediksi merupakan data harga penutupan harian (*closing price*) Ethereum.

3. Penelitian ini hanya membahas aspek teknis dalam pembangunan dan evaluasi model TCN, tidak mencakup analisis fundamental atau sentimen pasar terhadap harga mata uang kripto.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu :

1. Mengimplementasikan algoritma *Temporal Convolutional Network* (TCN) untuk melakukan prediksi harga penutupan harian Ethereum.
2. Mengetahui performa TCN dalam melakukan prediksi harga penutupan harian Ethereum.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut

1. Penelitian ini dapat memberikan kontribusi dalam pengembangan literatur terkait *Temporal Convolutional Network* (TCN) dalam predikis harga mata uang kripto khususnya harga penutupan harian Ethereum.
2. Menyediakan model prediksi harga yang dapat membantu investor, analis, dalam mengantisipasi fluktuasi harga Ethereum, sehingga dapat mendukung pengambilan keputusan investasi.

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 ulasan teori penelitian terdahulu yang berkaitan untuk membantu memahami konsep algoritma TCN dan pasar kripto khususnya Ethereum 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

Dalam pemodelan deret waktu, penelitian sebelumnya yang membandingkan TCN dengan arsitektur jaringan syaraf tiruan berulang seperti LSTM dan GRU dalam pemodelan deret waktu menunjukkan bahwa TCN dapat menangkap ketergantungan jangka panjang lebih baik dengan menggunakan arsitektur konvolusional yang mencakup *causal convolution*, *dilated convolution*, dan koneksi residual. Selain itu, karena TCN tidak bergantung pada perhitungan berurutan, TCN memiliki arsitektur yang lebih sederhana, lebih stabil selama pelatihan, dan menyimpulkan informasi dengan lebih cepat (Bai dkk., 2018).

Penelitian yang lain mengenai prediksi harga penutupan saham PT Bank Central Asia Tbk menggunakan algoritma *Temporal Convolutional Network* (TCN) menunjukkan bahwa hasil terbaik diperoleh dengan kombinasi parameter yang ideal, yaitu ukuran *batch* kecil dan jumlah *epoch* yang terbatas. Penggunaan optimasi Adam turut berkontribusi dalam meningkatkan efisiensi pelatihan model. Model ini menghasilkan nilai rata-rata terbaik MAE sebesar 49, MSE sebesar 6213, dan RMSE sebesar 78. Penelitian ini menegaskan bahwa TCN dapat menjadi metode yang berguna untuk memprediksi harga saham, serta membuka peluang penelitian masa depan dalam mengeksplorasi variasi parameter atau menggunakan data lain untuk membandingkan performa TCN dalam konteks yang sama (Wathani dkk., 2025).

Penelitian lainnya terkait perbandingan model TCN, *Transformers* dan *Recurrent Neural Network* (RNN) dalam prediksi kenaikan permukaan air di Pesisir Surabaya. Pada penelitian tersebut data pelatihan dibagi menjadi tiga skenario yaitu 3 bulan, 6 bulan, dan 8 bulan. Hasil penelitian menunjukkan bahwa skenario data pelatihan 8 bulan memberikan hasil terbaik dan secara keseluruhan, TCN menunjukkan performa prediksi yang lebih konsisten dibandingkan model lainnya pada berbagai interval waktu peramalan dengan nilai R^2 sebesar 0,9950 dan nilai RMSE sebesar 0,0487, sedangkan *Transformers* dan RNN memiliki nilai masing-

masing R^2 0,9727 dan RMSE 0,1135 untuk Transformers, serta R^2 0,9688 dan RMSE 0,1213 untuk RNN (Saepudin dkk., 2024).

Penelitian lain mengenai prediksi pasar saham mengusulkan model yang menggabungkan *Temporal Convolutional Networks* (TCN) dengan fitur emosi yang diekstraksi dari data teks Twitter menggunakan klasifikator emosi multi-label berbasis BERT. Dengan menerapkan seleksi fitur berdasarkan korelasi korelasi dan penggabungan prediksi melalui rata-rata sederhana, metode ini menunjukkan performa yang lebih unggul dibandingkan berbagai metode individu maupun ensemble pada prediksi jangka pendek hingga menengah dengan nilai rata-rata MAE 4,158 dan MAPE 0,082. (Liapis & Kotsiantis, 2023).

Penelitian lain yang membandingkan *Temporal Convolutional Networks* (TCN) dengan model deret waktu tradisional seperti CNN-LSTM dalam konteks prediksi harga emas. Hasil dari penelitian ini menunjukkan bahwa TCN secara konsisten memberikan performa prediksi yang lebih baik. Hal ini dibuktikan melalui nilai metrik evaluasi yang diperoleh, dalam hal ini model TCN menghasilkan nilai *Root Mean Squared Error* (RMSE) sebesar 15,26 dan *Mean Absolute Error* (MAE) sebesar 10,05. Sebaliknya, model CNN-LSTM yang digunakan sebagai pembanding menghasilkan nilai RMSE sebesar 23,5 dan MAE sebesar 15,4. Perbedaan ini menunjukkan bahwa TCN mampu mengurangi kesalahan prediksi secara signifikan (Fajou & McCarren, 2021).

Penelitian lainnya tentang prediksi harga mata uang kripto Ethereum. Penggunaan algoritma *Support Vector Regression* (SVR) yang memanfaatkan fitur teknikal berupa *Moving Average* (MA) dan *Exponential Moving Average* (EMA) dengan periode 10 hari sebagai variabel input utama. Hasil implementasi model SVR menunjukkan kinerja prediksi yang sangat baik, dengan nilai koefisien determinasi (R^2) sebesar 0,9985, yang mengindikasikan bahwa model mampu menjelaskan lebih dari 99% variasi harga Ethereum. Selain itu, nilai *Mean Squared Error* (MSE) yang diperoleh juga tergolong rendah, yaitu sebesar 2137,97, yang menunjukkan bahwa rata-rata selisih kuadrat antara harga aktual dan hasil prediksi cukup kecil (Fatah dkk., 2025).

Berdasarkan beberapa penelitian diatas yang dilakukan terhadap objek harga Ethereum, terlihat bahwa algoritma *Temporal Convolutional Network* belum digunakan sehingga sangat layak diuji lebih lanjut. Penerapan TCN terbukti efisien dalam memprediksi dan memberikan hasil prediksi yang baik. Dengan kemampuannya dalam mengenali pola jangka panjang secara efisien, TCN memiliki potensi untuk menghasilkan prediksi harga Ethereum yang lebih akurat.

2.2 Dasar Teori

2.2.1 Cryptocurrency

Cryptocurrency adalah sebuah mata uang digital atau virtual yang dibuat sebagai alat tukar. *Cryptocurrency* merupakan mata uang digital yang diciptakan dari rangkaian *code* atau disebut *blockchain* (Rizkilloh & Widiyanesti, 2022). *Cryptocurrency* juga bisa disebut dengan seperangkat teknologi berbasis kriptografi dan algoritma, yang secara matematis akan menyusun beberapa kode dan sandi untuk mencetak *cryptocurrencies*. Mata uang kripto dianggap memiliki kelebihan dibandingkan mata uang konvensional diantaranya adalah fleksibel bisa digunakan dimana saja, transparan, cepat, dan biaya transaksi yang rendah (Baharani dkk., 2024). *Cryptocurrency* dapat menjadi salah satu alternatif dalam berinvestasi. Investasi dalam bentuk uang virtual khususnya kripto dapat membantu masyarakat dalam mempersiapkan finansial untuk masa yang akan datang (Atmaja & Hakim, 2022).

2.2.2 Ethereum

Ethereum merupakan salah satu jenis mata uang kripto yang cukup *popular* setelah bitcoin. Ethereum juga disebut *platform blockchain* terdesentralisasi berbasis *smart contract* yang berfungsi untuk mewujudkan layanan pertukaran secara online (Goa Mere, 2023). Ethereum pertama kali diperdagangkan pada 7 Agustus 2015 dengan nilai awal sebesar US\$2,83 dan mencapai puncak harga tertinggi pada 8 November 2021, yaitu sebesar US\$4.822,97. Ethereum memungkinkan pengguna untuk melakukan transaksi melalui jaringan internet tanpa perlu menggunakan kartu kredit atau melibatkan bank sentral, sehingga prosesnya menjadi lebih mudah dan efisien (Bastian Sianturi dkk., 2023).

2.2.3 Machine Learning

Machine Learning adalah salah satu bagian dari bidang ilmu AI (*Artificial intelligence*) yang memungkinkan mesin dapat belajar sendiri menggunakan data-data tanpa harus berulang kali diprogram manusia (Alfarizi dkk., 2023). Untuk bisa mengembangkan sistem *Artificial intelligence* supaya dapat menjalankan tugas tanpa harus diprogram berulang kali, aspek terpenting adalah bagaimana sistem tersebut dapat belajar sehingga mendapatkan informasi yang dibutuhkan untuk melakukan tugasnya (Arnolus Juantri dkk., 2023). Konsep dari *machine learning* adalah melatih mesin menggunakan model algoritma matematika melalui data-data inputan dengan tujuan memperoleh informasi (*knowledge discovery/discovery unknown structure*) dan memprediksi pola (*pattern discovery*) secara mandiri (Purwono dkk., 2022). Proses dalam pembelajaran mesin untuk memperoleh kecerdasan melalui dua tahapan antara lain tahapan latihan (*training*) dan pengujian (*testing*) (Kanedi & Jumadi, 2024). Semakin sering mesin dilatih dan diuji, maka mesin akan semakin cerdas dan kemampuan untuk mengenali informasi juga semakin akurat.

2.2.4 Deep learning

Deep learning (Pembelajaran Mendalam) atau *Deep Structured Learning* (Pembelajaran Struktural Mendalam) merupakan bidang turunan dari *machine learning* yang memanfaatkan *Artificial Neural Network* (Jaringan Syaraf Tiruan). Konsep dari *deep learning* adalah menambahkan banyak lapisan pada model sehingga proses belajar menjadi lebih kompleks dan mendalam (Nurhakiki & Yahfizham, 2024). Hal yang membedakan *deep learning* dengan *machine learning* adalah data yang digunakan merupakan data yang sangat banyak. Dengan begitu diperlukan pula proses komputasi yang membutuhkan spesifikasi *hardware* yang mumpuni, biasanya digunakan *Graphics Processing Unit* (GPU) (Digo Saputra & Hatta Fudholi, 2023).

2.2.5 Time Series

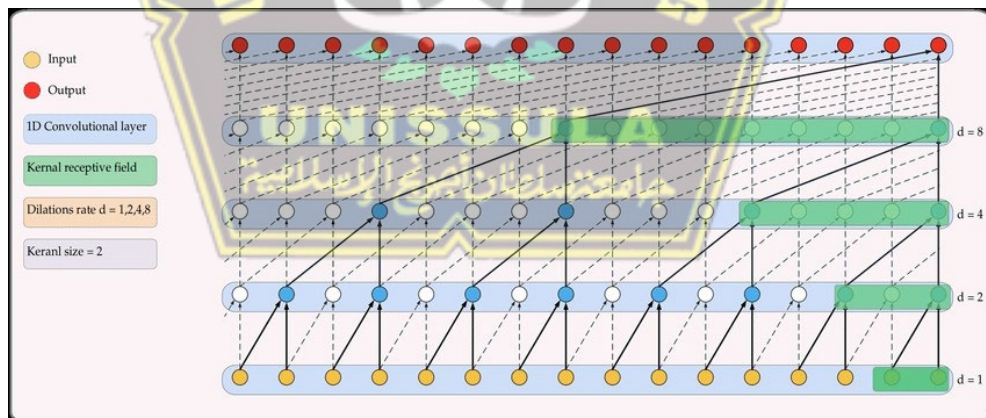
Time Series merupakan serangkaian beberapa nilai variabel yang tersusun berdasarkan waktu, Analisis *time series* mempelajari pola gerakan nilai-nilai *variabel* pada satu interval waktu seperti mingguan, bulanan, tahunan yang teratur.

Dengan data *input* sebagai berikut $X = (x_0, x_1, x_2, \dots, x_T)$ dan mencoba memprediksi output sebagai berikut $Y = (y_0, y_1, y_2, \dots, y_T)$. untuk memprediksi *output* y_T , hanya dapat menggunakan data input yang telah diamati sebelumnya.. Manfaat dari analisis *time series* dapat diperoleh ukuran-ukuran yang dapat digunakan untuk membuat keputusan pada saat ini, untuk peramalan dan untuk merencanakan masa depan (Fungki Wahyu & Billy Hendrik, 2023).

2.2.6 Temporal Convolutional Network (TCN)

Temporal Convolutional Network (TCN) merupakan algoritma yang dibuat untuk menangani permasalahan pada data sekuensial atau deret waktu dengan memanfaatkan konvolusi satu dimensi (*1D convolution*) secara mendalam. Algoritma tersebut sering digunakan dalam tugas segmentasi tindakan dan prediksi deret waktu (Aisyah dkk., 2024). TCN menggunakan pendekatan konvolusional yang memungkinkan pemrosesan data secara paralel dan efisien. Struktur TCN menggabungkan beberapa komponen penting seperti *causal convolutions*, *dilated convolutions* dan *residual connections*, yang meningkatkan efektivitasnya (Saepudin dkk., 2024).

TCN memiliki struktur jaringan seperti gambar berikut :



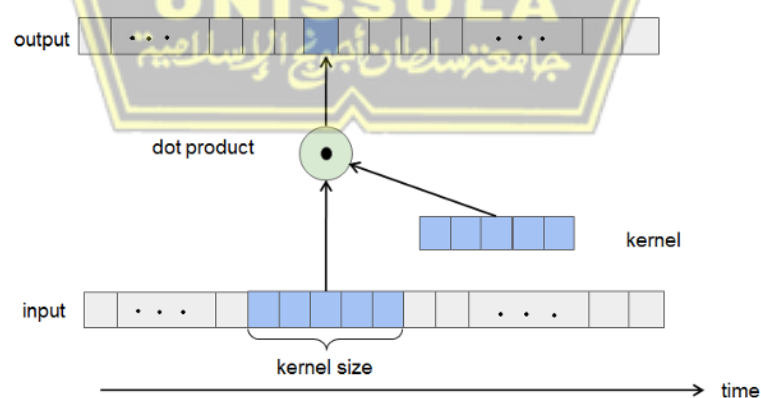
Gambar 2. 1 Model TCN (Adel Musallam dkk., 2019)

Gambar 2.1 menunjukkan menggambarkan arsitektur *Temporal Convolutional Network* (TCN) untuk pemrosesan data deret waktu. TCN menggunakan beberapa lapisan *1D convolutional* dan menerapkan *dilated convolutions* secara bertingkat dengan faktor dilasi yang meningkat secara eksponensial ($d = 1, 2, 4, 8$). Setiap titik berwarna mewakili *node*, kuning untuk *input*, biru dan putih untuk neuron di lapisan

konvolusi, dan merah untuk output. Dilasi memungkinkan model menangkap dependensi jangka panjang tanpa memperbesar ukuran jaringan secara signifikan, karena setiap lapisan melompat lebih jauh ke belakang dalam waktu. Area hijau menunjukkan *receptive field kernel*, yang memperluas cakupan input yang dapat dipengaruhi oleh output akhir. Pendekatan ini menjadikan TCN unggul dalam mempelajari pola temporal dibandingkan jaringan konvolusional biasa.

2.2.7 One Dimensional Convolutional Network

One Dimensional Convolutional Network (1D ConvNet) merupakan jenis jaringan saraf konvolusional yang dirancang khusus untuk memproses data sekuensial satu dimensi, seperti sinyal waktu, deret waktu, dan teks. Dalam arsitektur *Temporal Convolutional Network (TCN)*, *1D ConvNet* berperan sebagai komponen utama yang digunakan untuk mengekstraksi pola dari data sekuensial. Konvolusi satu dimensi dalam TCN bekerja dengan cara menggeser *filter* sepanjang sumbu waktu, sehingga setiap output dipengaruhi oleh sejumlah nilai *input* sebelumnya dalam jendela waktu tertentu. Data input *1D ConvNet* biasanya berbentuk tensor dengan konfigurasi (*batch_size*, *input_length*, *input_size*). Setelah melewati proses konvolusi, output-nya berbentuk tensor dengan format (*batch_size*, *input_length*, *output_size*). Pada gambar 6.2 menunjukkan bagaimana proses *1 dimensional convolution* yang mengubah *input* menjadi *output*.



Gambar 2. 2 Proses *1D convolution* (Dai dkk., 2023)

Setiap proses konvolusi akan berlangsung di setiap data pada seluruh *input tensor* tersebut. Pada gambar 6.2, digunakan *kernel size* yaitu lima, artinya kernel akan mengambil lima data dari *input* lalu mengalikan tiap nilai itu dengan bobot

dalam *kernel*, kemudian hasilnya dijumlahkan yang selanjutnya disimpan di *output tensor* sebagai hasil akhir. *Kernel* memiliki bobot yang sama pada setiap proses yang berjalan. Proses berlangsung sama dengan *kernel* akan mulai bergerak ke arah data selanjutnya setiap satu data dan hasilnya disimpan lagi pada *output*.

2.2.8 Causal convolutions

Dalam arsitektur TCN, model tidak menggunakan informasi dari masa depan saat memproses data, sehingga digunakan *causal convolution*, yaitu proses konvolusi yang hanya menggunakan data saat ini dan data sebelumnya (Liu et al., 2019). Untuk memastikan bahwa panjang *output* tetap sama dengan *input* dan tetap menjaga kausalitas, maka dilakukan penambahan padding nol (*zero-padding*) di bagian awal data *input*. (Wang et al., 2020).

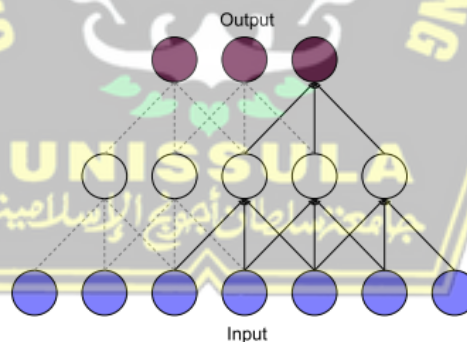
Untuk menghitung jumlah *padding*, digunakan rumus (1):

$$P = k - 1 \quad (1)$$

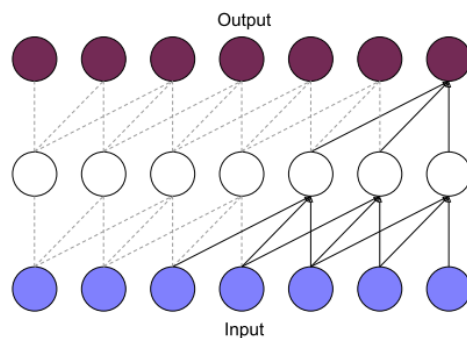
Keterangan:

P = Jumlah *padding*

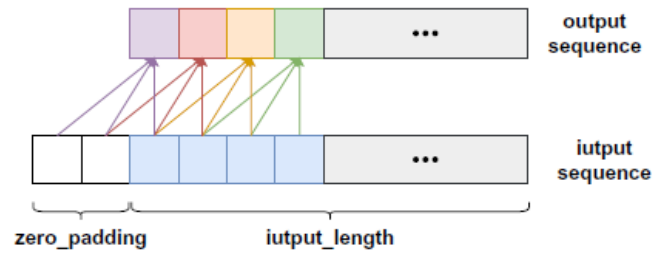
k = Ukuran *kernel*



Gambar 2. 3 Non Causal convolutions (Lara-Benítez dkk., 2020)



Gambar 2. 4 Causal convolutions (Lara-Benítez dkk., 2020)



Gambar 2. 5 Zero padding (Mo dkk., 2023)

Pada gambar 6.3 merupakan *non causal convolution* dengan yang menunjukkan setiap *output* dipengaruhi oleh data *input* dari masa lalu, sekarang, dan masa depan. Sedangkan gambar 6.4 merupakan *causal convolution* dengan yang menunjukkan setiap *output* dipengaruhi oleh data *input* dari masa lalu dan sekarang saja. Pada gambar 6.5 terdapat empat data *input* dengan *kernel_size* 3, jadi jumlah *padding* yang harus ditambahkan adalah *kernel size* – 1 yaitu 2. Proses konvolusi pada data pertama menambahkan 2 data tambahan yang disebut *padding*, hal tersebut dilakukan untuk mendapatkan hasil *output* dari data pertama, berlaku hal yang sama pada data selanjutnya sehingga data *output tensor* memiliki panjang yang sama dengan data *input*.

2.2.9 Dilated convolutions

Dalam proses konvolusi, *output* yang dihasilkan bergantung pada *kernel* yang digunakan serta nilai-nilai *input* sebelumnya. Misalnya, jika panjang data *input* adalah 4 dan ukuran *kernel size* 3, maka nilai *output* pada posisi ke-4 sangat dipengaruhi oleh data pada posisi ke-2 dan ke-3. Untuk mengurangi ketergantungan yang terlalu dekat dan memungkinkan model menangkap pola jangka panjang, digunakanlah *dilated convolutions*.

Dilated convolutions adalah teknik konvolusi yang memperluas jangkauan *kernel* dengan menyisipkan jarak atau dilasi antar elemen *input* yang digunakan, sehingga dapat menjangkau informasi dari masa lalu yang lebih jauh tanpa menambah jumlah parameter secara signifikan. Ini dilakukan dengan menambahkan lapisan (*layer*) yang menggunakan kernel dengan dilasi yang semakin besar di setiap *layer*-nya. *Dilated convolution* secara matematis dapat dinyatakan dalam rumus (2)

$$F(s) = (x_d f) = \sum_{i=0}^n f(i) \cdot x_{s-d.i} \quad (2)$$

Keterangan:

$F(s)$: hasil akhir pada posisi s

x_d : data x dengan kelipatan d

f : *filter* atau kernel dengan panjang $n+1$

$\sum_i^n$: penjumlahan dari indeks $i=0$ sampai $i=n$

k : adalah *kernel size*

$f(i) \cdot x_{s-dxi}$: kalikan nilai filter $f(i)$ dengan data x_{s-dxi}

Untuk menghitung jumlah layer (n) yang dibutuhkan agar dapat menjangkau panjang input tertentu (l), serta untuk menghitung *receptive field* (r), digunakan rumus 3 dan 4.

$$n = \frac{(l-1)}{(k-1)} \quad (3)$$

$$r = 1 + n(k-1) \quad (4)$$

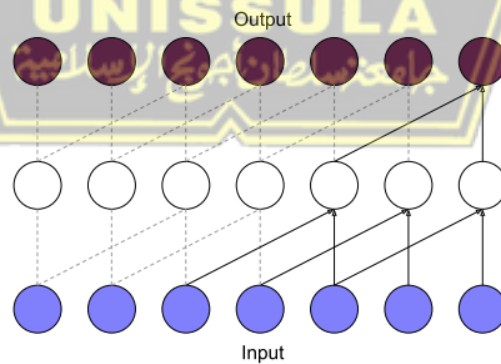
Dimana:

n : Jumlah *layers*

l : Panjang data

k : Ukuran *kernel*

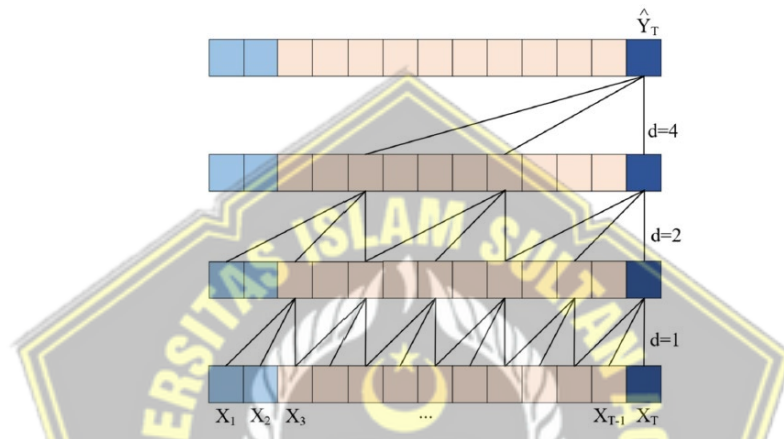
r : Rentang data yang dilalui proses dilasi



Gambar 2. 6 *Dilated Convolution* (Lara-Benítez dkk., 2020)

Pada gambar 2.6 ditunjukkan contoh penggunaan *dilated convolution* dengan panjang data sebesar 7, ukuran kernel 2, nilai dilasi (*dilation*) 2, dan rentang data yang dijangkau adalah 3. Jika tidak menggunakan dilasi, maka rentang data yang dilalui hanya sebesar ukuran *kernel*, yaitu 2. Namun, ketika dilasi diterapkan,

rentang ini meluas menjadi 3 karena adanya jarak antar titik yang diproses oleh *kernel*. Namun, jika nilai dilasi tetap dipertahankan pada setiap layer, maka akan dibutuhkan data input yang sangat panjang di layer-layer berikutnya. Untuk mengatasi hal ini, nilai dilasi biasanya ditingkatkan secara eksponensial pada setiap lapisan, seperti yang ditunjukkan pada gambar 6.7. Nilai dasar eksponensial ini disebut sebagai *dilation base*, dan disimbolkan dengan b .



Gambar 2. 7 Dilated causal convolution (Hao & Liu, 2024)

Pada Gambar 2.7, digunakan panjang data sebesar 13, ukuran *kernel* 3, *dilation base* 2 (*dilation factors* 1,2,4), dan jumlah lapisan sebanyak 3. Dalam praktiknya, nilai *kernel* harus lebih besar dari nilai *dilation base*. Selain itu, karena peningkatan nilai dilasi per *layer*, maka jumlah *padding* yang diperlukan juga bertambah di setiap lapisan. Oleh karena itu, digunakan rumus 5 untuk menghitung nilai dilasi dan rumus 6 untuk menghitung *receptive field* :

$$d = b^i \quad (5)$$

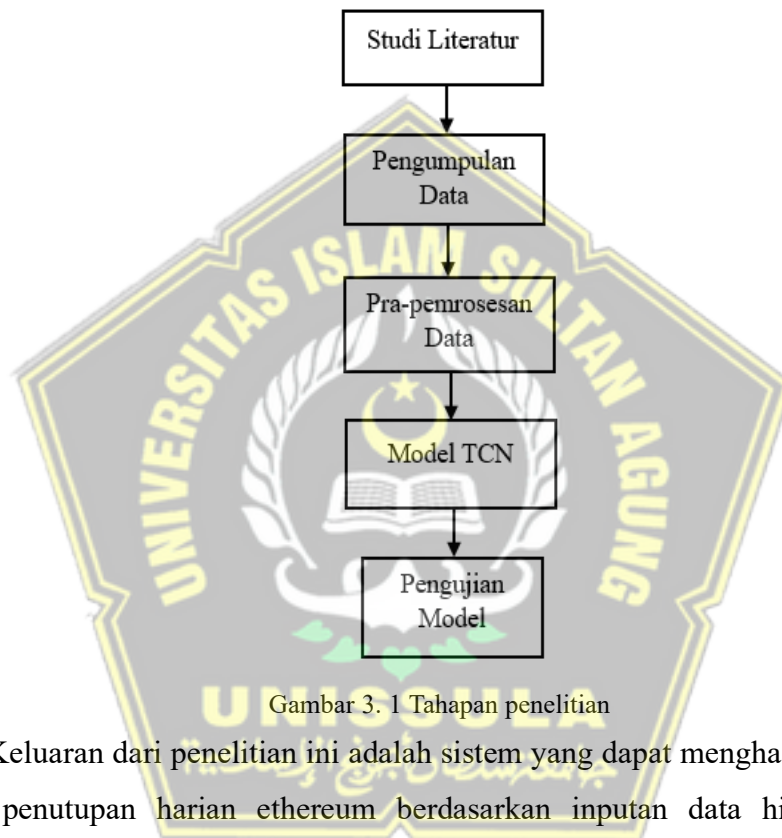
$$d = b^i \cdot (k - 1) \quad (6)$$

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penilitan

Pada bagian ini, akan membahas langkah-langkah yang akan digunakan dalam proses prediksi harga penutupan harian ethereum menggunakan algoritma TCN. Berikut merupakan tahapan yang akan dilakukan :



Gambar 3. 1 Tahapan penelitian

Keluaran dari penelitian ini adalah sistem yang dapat menghasilkan prediksi harga penutupan harian ethereum berdasarkan inputan data historis dengan menggunakan algoritma *Temporal Convolutional Network*.

3.1.1 Studi Literatur

Dalam proses penelitian ini peneliti melakukan pencarian dari berbagai sumber informasi yang terpercaya mengenai teori yang relevan dengan penelitian yang dilakukan untuk memperdalam pemahaman. Berbagai teori yang dipelajari meliputi *Machine Learning*, *Data Mining*, dan *Temporal Convolutional Network*. Sumber-sumber yang dijadikan referensi meliputi artikel ilmiah, jurnal akademik, laporan tugas akhir mahasiswa, hingga informasi yang diperoleh dari situs-situs *website* kredibel yang tersedia di internet.

3.1.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data historis harga penutupan harian aset kripto Ethereum. Data tersebut diperoleh dari sumber resmi, yaitu melalui situs *website* Yahoo Finance, yang dikenal luas sebagai penyedia informasi pasar keuangan yang terpercaya. Periode data yang dikumpulkan mencakup rentang waktu dari tanggal 1 Januari 2018 hingga 28 Mei 2025. Dalam kurun waktu tersebut, dataset yang dihasilkan terdiri dari total 2.705 baris data, di mana setiap baris mewakili informasi harga pada satu hari perdagangan. Dataset ini memuat sejumlah kolom atau label penting seperti *Date*, *Open*, *High*, *Low*, *Close*, dan *Volume*, yang semuanya memberikan gambaran menyeluruh tentang pergerakan harga Ethereum dari waktu ke waktu.

3.1.3 Pra-pemrosesan Data

Tahapan pra-pemrosesan data yang dilakukan dalam penelitian ini mencakup sejumlah langkah penting yang dirancang untuk memastikan bahwa data yang digunakan memiliki kualitas yang baik, bersih, dan siap untuk dimasukkan ke dalam proses pelatihan model prediksi. Berikut ini adalah beberapa tahapan pra-pemrosesan data yang diterapkan dalam penelitian ini.

3.1.3.1 Data Cleaning

Data cleaning atau pembersihan data, yang memiliki peran penting dalam memastikan kualitas data sebelum digunakan dalam pelatihan model. Pada tahap ini, peneliti secara cermat menangani adanya nilai-nilai yang hilang (*missing values*) atau tidak lengkap, serta menghapus data yang bersifat duplikat. Langkah ini dilakukan agar data yang digunakan benar-benar merepresentasikan kondisi sebenarnya tanpa adanya gangguan dari informasi yang tidak valid, tidak akurat, atau bersifat bias. Dengan demikian, model yang dibangun nantinya dapat belajar dari data yang bersih dan terpercaya, sehingga hasil prediksi yang dihasilkan pun menjadi lebih akurat.

3.1.3.2 Normalisasi Data

Normalisasi data yaitu mengubah skala data numerik ke dalam rentang yang seragam, yakni antara 0 hingga 1. Normalisasi ini bertujuan untuk mencegah dominasi fitur tertentu akibat perbedaan skala nilai, serta mempercepat proses

pelatihan model. Teknik normalisasi yang digunakan adalah *min-max scaler*, yang dirumuskan pada rumus(7).

$$x' = \frac{(x - \min_x)}{(\max_x - \min_x)} \quad (7)$$

keterangan :

x : data yang dinormalisasi

x' : data setelah dinormalisasi

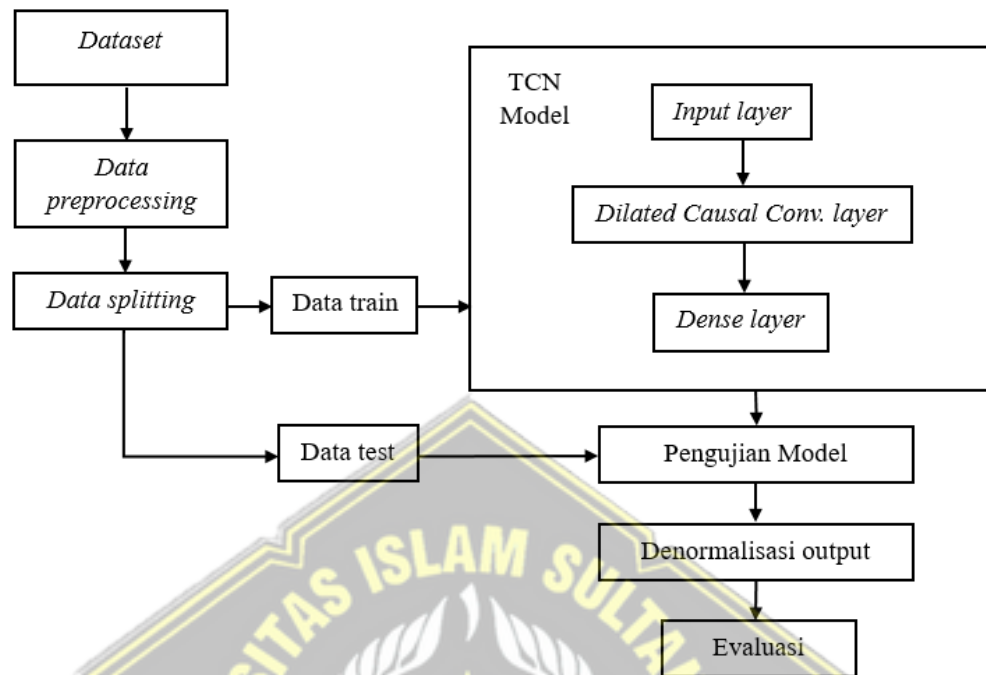
$\min$: nilai minimum dari keseluruhan data

$\max$: nilai maksimum dari keseluruhan data.

3.1.3.3 Pembagian Data

Tahapan selanjutnya adalah proses pembagian data (*data splitting*), di mana dataset dibagi menjadi dua bagian utama, yaitu data pelatihan (*training data*) dan data pengujian (*testing data*). Tujuan dari pembagian ini adalah untuk memberikan kesempatan bagi model untuk belajar dari sebagian data yang tersedia, yaitu data pelatihan, guna membentuk pola dan memahami hubungan antar variabel dalam dataset. Setelah model selesai dilatih, performa dan kemampuannya untuk melakukan generalisasi akan diuji menggunakan data pengujian—yakni data yang sebelumnya tidak pernah dilihat oleh model. Dengan cara ini, peneliti dapat menilai sejauh mana model mampu bekerja dengan baik pada data baru di luar data yang digunakan saat pelatihan, yang menjadi indikator penting dalam mengukur akurasi model.

3.1.4 Perancangan Model



Gambar 3. 2 Flowchart model TCN

Berikut adalah alur proses prediksi harga Ethereum menggunakan model *Temporal Convolutional Network* (TCN) yang ditunjukkan pada gambar 3.2 :

1. *Dataset*

Dataset yang digunakan adalah kolom *close* atau data harga penutupan harian Ethereum

2. *Data preprocessing*

Kemudian dilakukan pemrosesan data dengan melakukan pengecekan data yang nilainya kosong atau nilai duplikat sebelum diuji pada model

3. *Data splitting*

Selanjutnya dilakukan pemisahan data menjadi data latih dan data uji yang nantinya akan akan dilatih dan diuji menggunakan model TCN.

4. Tahap pelatihan

Data latih digunakan untuk melatih arsitektur model TCN agar dapat mengenali pola dari data historis harga penutupan Ethereum. Sebelum dimasukkan ke dalam model, data latih telah melalui proses normalisasi menggunakan metode *Min-Max Scaler* sehingga seluruh nilai berada dalam rentang antara 0 hingga 1.

Dalam pelatihan model, digunakan teknik *early stopping*, yaitu suatu metode regulasi untuk mencegah model mengalami *overfitting* terhadap data latih. *Early stopping* bekerja dengan cara memantau nilai *loss* pada data validasi dan secara otomatis menghentikan proses pelatihan jika tidak terdapat perbaikan performa dalam beberapa *epoch* berturut-turut. Selain itu, digunakan juga opsi *restore_best_weights=True*, yang memastikan bobot model dikembalikan ke kondisi terbaiknya saat nilai *validation loss* paling rendah tercapai.

5. Tahap didalam model TCN

Model TCN yang digunakan dalam penelitian ini terdiri dari beberapa layer:

a. *Input layer*

Layer pertama adalah *input layer* yang berfungsi untuk menerima data yang sudah diproses dalam bentuk sekuens dengan format tiga dimensi: (jumlah sampel, panjang *time_step*, jumlah fitur). Dalam penelitian ini, *time_step* ditetapkan sebanyak 30 hari, dan fitur yang digunakan hanya satu, yaitu harga penutupan harian Ethereum.

b. *Dilated causal convolution layer*

Komponen inti dari TCN adalah *dilated causal convolution*, yang bertujuan untuk menangkap dependensi jangka panjang dalam data deret waktu. Pada lapisan ini digunakan:

- *nb_filters*=64, yaitu jumlah filter konvolusi sebanyak 64 buah.
- *kernel_size*=3, yaitu ukuran jendela konvolusi yang memproses tiga nilai sekaligus.
- *dilations*=[1, 2, 4, 8], yaitu faktor dilasi yang mengatur seberapa jauh jaringan melompat dalam membaca urutan waktu, sehingga memungkinkan model mengenali pola jangka panjang tanpa meningkatkan jumlah parameter secara signifikan.
- *return_sequences=False*, artinya hanya output terakhir dari urutan yang dikembalikan, karena model ini hanya memprediksi satu nilai ke depan.
- *use_skip_connections=True*, yang memungkinkan penggunaan koneksi residual untuk membantu aliran gradien selama pelatihan, serta menjaga kestabilan pelatihan model yang dalam.

c. *Dropout*

Setelah lapisan konvolusi, ditambahkan *Dropout* dengan rasio 0,2. Hal ini berfungsi sebagai teknik regularisasi untuk mencegah *overfitting* dengan cara mengabaikan 20% neuron secara acak selama pelatihan, sehingga model tidak terlalu bergantung pada pola tertentu dalam data latih.

d. *Dense layer*

Layer terakhir adalah *dense layer* dengan satu neuron *output*, yang berfungsi menghasilkan prediksi akhir, yaitu harga Ethereum untuk hari ke-31 berdasarkan 30 hari data sebelumnya. *Layer* ini juga bertugas mengubah output dari format *multi-channel* (hasil konvolusi) menjadi satu nilai prediktif yang bisa dibandingkan langsung dengan nilai target.

e. *Model compiler*

Model kemudian dikompilasi menggunakan *optimizer* Adam, yang merupakan metode optimasi adaptif yang efisien untuk data *time series*, serta fungsi kerugian (*loss function*) *Mean Squared Error* (MSE), yang sesuai digunakan untuk masalah regresi seperti prediksi harga.

6. Tahap pengujian model

Setelah proses pelatihan model selesai dilakukan menggunakan data latih, langkah selanjutnya adalah pengujian model untuk mengevaluasi performa model terhadap data yang belum pernah dilihat sebelumnya, yaitu data uji (*test data*). Dalam tahap ini, model yang telah dilatih digunakan untuk melakukan prediksi terhadap input data uji.

7. Denormalisasi

Denormalisasi merupakan Proses ini dilakukan untuk mengubah data data prediksi yang awalnya berada dalam rentang interval 0-1 kembali ke nilai aktual harga saham (Bode, 2017).

$$d = \hat{d}(\max - \min) + \min \quad (8)$$

8. Evaluasi model

Evaluasi model bertujuan untuk mengukur akurasi dari model prediksi yang telah dikembangkan. Metrik evaluasi yang digunakan untuk mengukur seberapa baik model tersebut digunakan dalam memprediksi yaitu dengan menggunakan

Mean Absolute Error (MAE) dan *Mean Absolute Percentage Error* MAPE. MAE menghitung rata-rata dari selisih absolut antara nilai prediksi dan nilai aktual. Metrik ini memberikan gambaran seberapa besar kesalahan rata-rata dalam satuan yang sama dengan data aslinya. Nilai MAE yang lebih rendah menunjukkan bahwa prediksi model semakin mendekati nilai sebenarnya. (9).

$$MAE = \frac{1}{N} \sum_{i=0}^N |y_i - \hat{y}_i| \quad (9)$$

Keterangan:

y_i : nilai aktual data ke-i

$\hat{y}_i$: nilai prediksi data ke-i

n : jumlah total data

Sementara itu, MAPE mengukur kesalahan prediksi dalam bentuk persentase terhadap nilai aktual. MAPE berguna untuk mengetahui seberapa besar kesalahan relatif yang dilakukan model dalam melakukan prediksi, tidak peduli apakah kesalahannya negatif atau positif. Nilai MAPE yang lebih rendah menunjukkan bahwa model memiliki tingkat akurasi yang lebih tinggi. Rumus MAPE ditulis pada rumus (10).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (10)$$

Keterangan:

y_i : nilai aktual data ke-i

$\hat{y}_i$: nilai prediksi data ke-i

n : jumlah total data

Tingkat akurasi model berdasarkan nilai MAPE diklasifikasikan dalam empat kategori (Budiprasetyo dkk., 2023), seperti yang ditunjukkan pada Tabel 3.1 berikut :

Tabel 3. 1 Skala nilai akurasi MAPE

MAPE	Category
<10	Prediksi akurat
10-20	Prediksi baik
20-50	Prediksi layak
21-50	Prediksi tidak akurat

3.2 Analisis kebutuhan

Dalam penelitian ini, digunakan berbagai kebutuhan pendukung untuk membangun sistem agar dapat berfungsi sesuai dengan tujuan yang diharapkan. Adapun analisis kebutuhan yang digunakan antara lain:

1. *Python 3.10.11*

Python adalah bahasa pemrograman tingkat tinggi yang dirancang untuk memberikan kemudahan dan fleksibilitas dalam pengembangan perangkat lunak. Dalam penelitian ini digunakan *Python* versi 3.10.11, karena versi tersebut telah mendukung berbagai pustaka serta fitur terkini yang sangat mendukung pengembangan sistem berbasis *machine learning*, *data science*, dan *deep learning*.

2. *TensorFlow/Keras*

TensorFlow/Keras merupakan pustaka *Open-source* yang digunakan untuk membangun serta melatih model *deep learning*. *Keras* dirancang dengan antarmuka yang sederhana dan efisien, sehingga memudahkan pengembangan model neural network secara cepat. Salah satu aplikasi penting *Keras* adalah dalam prediksi *time series* seperti harga kripto, melalui implementasi arsitektur jaringan saraf seperti TCN yang mampu mengenali pola dalam data temporal.

3. *Pandas*

Pandas adalah pustaka *Python* yang sangat umum digunakan untuk keperluan manipulasi dan analisis data. Pustaka ini menyediakan struktur data seperti *DataFrame* dan *Series*, yang memungkinkan pengguna melakukan berbagai operasi kompleks, seperti pembersihan data, transformasi, hingga analisis statistik. *Pandas* juga mendukung impor data dari beragam format, termasuk CSV, Excel, SQL, dan JSON.

4. *NumPy*

NumPy merupakan pustaka fundamental dalam komputasi numerik di *Python*. Dengan menyediakan objek *array* multidimensi, *NumPy* memungkinkan operasi matematis dan manipulasi vektor/matriks dilakukan secara efisien. Selain itu, *NumPy* memiliki beragam fungsi statistik dan matematis, serta dapat diintegrasikan

dengan pustaka lain seperti *Pandas* dan *Matplotlib* untuk keperluan analisis maupun visualisasi data.

5. *Matplotlib*

Matplotlib adalah pustaka visualisasi data di Python yang berguna untuk membuat berbagai jenis grafik seperti plot garis, *bar chart*, dan *scatter plot*. Sedangkan *Seaborn* adalah pustaka berbasis *Matplotlib* yang memudahkan pembuatan visualisasi statistik yang menarik dan informatif, seperti *heatmap*, *boxplot*, dan visualisasi distribusi data lainnya

6. *Scikit-learn*

Scikit-learn adalah pustaka Python yang banyak digunakan dalam implementasi algoritma machine learning. *Scikit-learn* menyediakan berbagai alat untuk preprocessing data seperti normalisasi, *split data*, serta mendukung banyak metode klasifikasi, regresi, dan *clustering*. Selain itu, pustaka ini juga menyediakan fitur validasi silang dan *tuning hyperparameter* untuk meningkatkan performa model secara optimal.

7. *Cycle*

Cycle merupakan teknik dalam analisis deret waktu (*time series*) yang digunakan untuk mengenali pola berulang jangka panjang yang tidak terkait dengan musiman. Pola siklus ini biasanya terjadi karena faktor ekonomi, bisnis, atau fenomena eksternal lainnya, dan penting untuk dianalisis dalam konteks prediksi jangka panjang.

8. *Plotly*

Plotly adalah pustaka visualisasi interaktif berbasis web yang memungkinkan pembuatan grafik dinamis dan responsif secara mudah. *Plotly* mendukung berbagai jenis grafik seperti *line chart*, *bar chart*, *scatter plot*, dan *heatmap*, serta dapat diintegrasikan dengan *Google Colab* maupun *Streamlit*. *Plotly* sangat bermanfaat dalam eksplorasi data karena pengguna dapat memperbesar, menggeser, dan menelusuri grafik secara interaktif.

9. *Google Colab*

Google Colab (Colaboratory) adalah platform berbasis cloud yang disediakan oleh Google untuk menjalankan kode Python melalui notebook interaktif langsung

di browser, tanpa perlu instalasi lokal. Colab mendukung eksekusi kode secara real-time, visualisasi data, dan dokumentasi dalam satu antarmuka, mirip dengan Jupyter Notebook. *Platform* ini sangat populer di kalangan peneliti, pelajar, dan praktisi data science karena menyediakan akses gratis ke GPU dan TPU, serta integrasi mudah dengan Google Drive untuk penyimpanan data dan proyek.

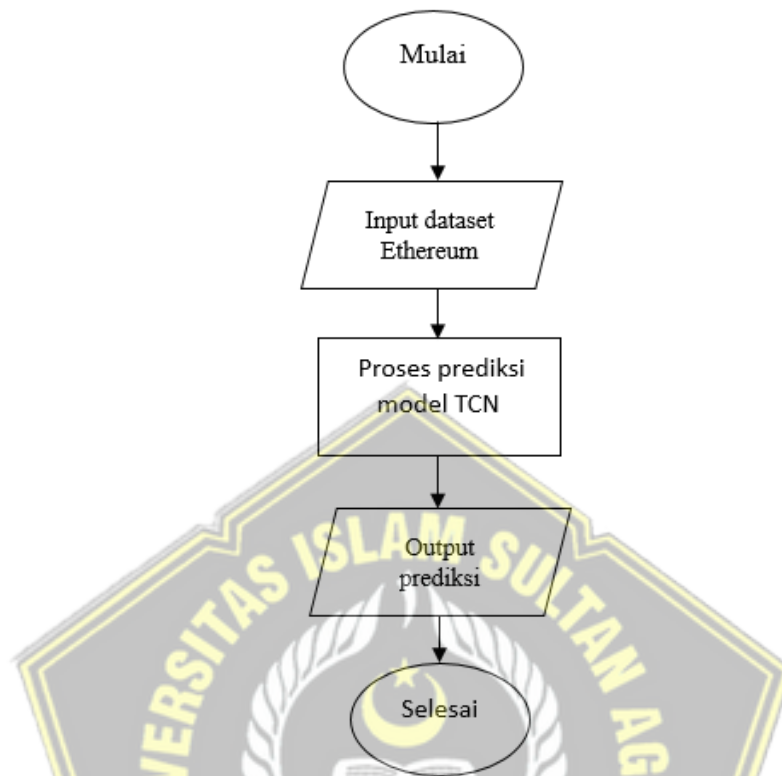
10. Visual Studio Code

Visual Studio Code (VS Code) adalah editor kode sumber gratis yang dikembangkan oleh Microsoft dan banyak digunakan oleh pengembang perangkat lunak. Editor ini mendukung berbagai bahasa pemrograman, seperti Python, JavaScript, dan C++, serta dilengkapi fitur unggulan seperti IntelliSense (saran kode otomatis), terminal bawaan, integrasi Git, dan dukungan ekstensi yang sangat luas. Dengan tampilan yang ringan namun fungsional, VS Code menjadi pilihan populer untuk pengembangan aplikasi baik oleh pemula maupun profesional.

11. Streamlit

Streamlit adalah sebuah *framework open-source* berbasis Python yang digunakan untuk membangun aplikasi web interaktif secara cepat dan mudah, terutama untuk kebutuhan *data science* dan *machine learning*. Dengan sintaks yang sederhana, pengguna dapat membuat antarmuka visual dari *script* Python hanya dengan beberapa baris kode, tanpa perlu keahlian dalam pengembangan web. Streamlit sangat cocok digunakan untuk membuat *dashboard*, visualisasi data, dan demo model AI.

3.3 Perancangan sistem



Gambar 3. 3 Flowchart sistem

Gambar 3.3 tersebut merupakan *flowchart* yang menggambarkan proses prediksi harga Ethereum menggunakan model *Temporal Convolutional Network* (TCN). Proses dimulai ketika user membuka sistem, kemudian memasukkan dataset Ethereum dalam bentuk .csv sebagai input utama. Setelah data dimasukkan, proses dilanjutkan ke tahap inti, yaitu melakukan prediksi menggunakan model TCN. Hasil dari proses prediksi tersebut kemudian ditampilkan atau dikeluarkan sebagai *output* prediksi. Terakhir, proses diakhiri dengan langkah "Selesai". Diagram ini memberikan gambaran ringkas dan terstruktur mengenai alur kerja sistem prediksi yang dikembangkan.

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil penelitian

4.1.1 Pengumpulan data

Pada tahap pengumpulan data, penelitian ini menggunakan dataset dari data historis harga Ethereum dalam kurs USD (ETH/USD) dari) Januari tahun 2018 hingga Mei 2025, terdiri dari kolom *Date*, *Open*, *High*, *Low*, *Close*, dan *Volume*. Dataset ini disimpan dalam bentuk file CSV. Berikut data historis dari etehreum yang terdapat pada tabel 4.1

Tabel 4. 1 Data Ethereum

<i>Date</i>	<i>Open</i>	<i>High</i>	<i>Low</i>	<i>Close</i>	<i>Volume</i>
2018-01-01	755,757019	782,530029	742,004028	772,640991	2595760128
2018-01-02	772,346008	914.830017	772,346008	884,443970	5783349760
2018-01-03	886,000000	974,471008	868,450989	962,719971	5093159936
2018-01-04	961,713013	1045,079956	946,085999	980,921997	6502859776
2018-01-05	975,750000	1075,390015	956,325012	997,719971	6683149824
...	...	...	...	...	...
2025-05-24	2526,387939	2575,131348	2516,018066	2530,646240	11380899856
2025-05-25	2530,890137	2553,869141	2467,880127	2551,763916	14556431364
2025-05-26	2551,380859	2598,566650	2530,321533	2564,138428	14936610009
2025-05-27	2564,140869	2712,285156	2512,588623	2663,069824	26264200536
2025-05-28	2663,010254	2688,742188	2611,160889	2682,212891	19087366068

Tabel 4.1 menunjukkan data historis harga harian Ethereum (ETH) terhadap USD dari tanggal 1 Januari 2018 hingga 28 Mei 2025. Setiap baris merepresentasikan satu hari perdagangan dan mencakup lima kolom utama yaitu *Date*, *Open*, *High*, *Low*, *Close*, dan *Volume*. Kolom *Open* adalah harga saat pasar dibuka, *High* adalah harga tertinggi harian, *low* adalah harga terendah, *close* adalah harga penutupan, dan kolom *volume* menunjukkan total nilai transaksi Ethereum dalam USD pada hari tersebut. Dataset tersebut berjumlah 2705 baris data yang nantinya akan dibagi menjadi data latih sebesar 80% dan data uji sebesar 20%. Penelitian ini akan fokus untuk memprediksi harga penutupan harian Ethereum. Untuk memberikan gambaran yang lebih jelas berikut adalah grafik pergerakan

harga penutupan harian Ethereum dari tahun 2018 sampai tahun 2025 yang dapat dilihat pada gambar 4.1



Gambar 4. 1 Grafik harga penutupan harian ethereum 2018-2025

Gambar 4.1 merupakan grafik harga penutupan harian Ethereum yang menunjukkan fluktuasi harga selama periode 1 Januari 2018 hingga 21 Mei 2025. Terlihat bahwa harga Ethereum mengalami kenaikan yang sangat signifikan pada tahun 2021, yang kemudian diikuti oleh penurunan tajam pada tahun 2022. Setelah itu, harga bergerak dinamis dengan pola naik-turun yang mencerminkan volatilitas pasar kripto. Untuk memperjelas deskripsi dari data *close* yang ditunjukkan pada gambar 4.1, berikut merupakan tampilan ringkasan statistiknya yang dijelaskan:

Tabel 4. 2 Statistik deskriptif harga penutupan Ethereum

Deskripsi	<i>Close</i>
Count	2705
Mean	1.560,67
Std	1.222,57
Min	84,3
25%	282,96
50%	1.579,73
70%	2.509,09
max	4.812,08

Pada tabel 4.2 menunjukkan statistik deskriptif pada kolom *close* yang menunjukkan bahwa data harga penutupan harian Ethereum terdiri dari 2.705 data. Rata-rata harga penutupan selama periode tersebut adalah sebesar 1.560,67 USD, dengan standar deviasi sebesar 1.222,57 USD yang menandakan tingkat fluktuasi harga yang cukup tinggi. Harga penutupan terendah tercatat sebesar 84,30 USD, sedangkan harga tertinggi mencapai 4.812,08 USD. Nilai kuartil pertama (25%) berada pada 282,96 USD, yang menunjukkan bahwa 25% data berada di bawah harga tersebut. Median harga penutupan adalah 1.579,73 USD, menandakan bahwa setengah dari total data memiliki harga di bawah nilai tersebut. Sementara itu, persentil ke-70 berada pada 2.509,09 USD, artinya hanya 30% data yang memiliki harga di atas angka tersebut. Rentang harga yang sangat luas ini menggambarkan tingginya volatilitas pasar Ethereum selama periode 2018 hingga 2025.

4.1.2 Pra-pemrosesan Data

Pada tahapan ini ada beberapa yang perlu di perhatikan dalam pemrosesan data yang nantinya akan digunakan untuk proses penelitian yaitu dengan pembersihan data, normalisasi data, dan pembagian data menjadi data latih dan uji.

4.1.2.1 Data *Cleaning*

Tahap pembersihan data dilakukan untuk memastikan bahwa dataset yang digunakan dalam penelitian bebas dari nilai yang hilang (*missing values*) dan tidak ada data duplikat.

A. Mengecek *missing value*

Tabel 4. 3 Hasil *missing value*

Kolom	<i>Missing value</i>
<i>Close</i>	0

Tabel 4.3 menunjukkan hasil pemeriksaan nilai yang hilang (*missing values*) pada kolom *Close*. Hasilnya menunjukkan bahwa semua kolom memiliki nol (0) nilai yang hilang, yang berarti tidak ada data yang kosong atau tidak tersedia dalam dataset tersebut. Ini menunjukkan bahwa dataset bersih dan lengkap, sehingga dapat digunakan langsung untuk analisis tanpa perlu proses imputasi atau pembersihan data tambahan.

B. Mengecek data duplikat

Tabel 4. 4 Hasil data duplikat

Kolom	Data duplikat
<i>Close</i>	0

Tabel 4.4 menunjukkan hasil dari pemeriksaan data duplikat pada dataset, pada kolom *close* dalam dataset tersebut tidak terdapat baris yang identik secara keseluruhan, sehingga tidak perlu dilakukan penghapusan atau penanganan duplikasi. Ini mengindikasikan bahwa data bersih dari pengulangan baris yang sama dan siap untuk analisis lebih lanjut.

4.1.2.2 Pembagian data

Tahapan ini digunakan untuk memisahkan data *training* dan data *cleaning* dengan tujuan untuk menghindari *overfitting* yang akan mengganggu pemrosesan data dalam model TCN.

Tabel 4. 5 Tabel jumlah data latih dan uji

Mata Uang Kripto	Jumlah Data	Data <i>Splitting</i>	
		Data <i>Train</i>	Data <i>Test</i>
Ethereum	2698	2164	541

Tabel 4.6 tersebut menunjukkan hasil dari proses pembagian data (*data split*) ke dalam dua subset, yaitu data latih (*data training*) dan data uji (*data testing*). Untuk pembagian data yang akan digunakan dalam pemrosesan data yaitu dengan rasio 80:20 dimana 80% merupakan data *training* dan 20% digunakan untuk data *testing*. Dalam dataset yang ada di setiap perusahaan berjumlah 2698 data sehingga menghasilkan sebanyak 2164 data digunakan untuk *training* dan 541 data digunakan untuk *testing*.

4.1.2.3 Normalisasi data

Proses normalisasi dilakukan untuk mengubah nilai dataset ke dalam skala 0 hingga 1 dengan tujuan untuk meminimalkan error saat pelatihan model. Normalisasi data dilakukan menggunakan metode *Min-Max Scaler* dari library *MinMaxScaler*, yang mengubah rentang nilai data menjadi antara 0 dan 1. Proses ini sangat penting karena dapat mempercepat konvergensi dan menjaga stabilitas

perhitungan saat menggunakan model *Temporal Convolutional Network* (TCN). Normalisasi membantu model TCN bekerja lebih efisien dalam menangkap pola deret waktu tanpa dipengaruhi oleh perbedaan skala antar fitur. Pada tahap ini, hanya data latih (*train data*) yang digunakan untuk menghitung nilai minimum (x_{min}) dan maksimum (x_{max}). Selanjutnya, parameter tersebut juga digunakan untuk mentransformasikan data uji (*test data*) agar tidak terjadi data leakage, yaitu kondisi di mana informasi dari data uji mempengaruhi proses pelatihan model. Tabel 4.5 berikut menyajikan contoh hasil normalisasi data harga penutupan Ethereum:

Tabel 4. 6 Tabel harga penutupan Ethereum aktual dan *scale*

<i>Date</i>	<i>Close</i>	<i>Scaled Close</i>
2018-01-01	772,640991	0.145593
2018-01-02	884,443970	0.169241
2018-01-03	962,719971	0.185798
2018-01-04	980,921997	0.189648
2018-01-05	997,719971	0.193201
...	...	...
2025-05-24	2530,646240	0.517439
2025-05-25	2551,763916	0.521906
2025-05-26	2564,138428	0.524523
2025-05-27	2663,069824	0.545449
2025-05-28	2682,212891	0.549498

Tabel 4.5 menyajikan data harga penutupan harian Ethereum sebelum dan sesudah dinormalisasi menggunakan metode MinMax Scaler. Kolom *Date* menunjukkan tanggal pencatatan harga, *close* adalah harga penutupan asli dalam satuan USD, dan *Scaled close* merupakan hasil normalisasi nilai *close* ke dalam rentang antara 0 hingga 1. Sebagai contoh, pada tanggal 1 Januari 2018, harga penutupan Ethereum adalah 772,64 USD yang setelah dinormalisasi menjadi 0,145593. Sedangkan pada tanggal 28 Mei 2025, harga penutupan sebesar 2.682,21 USD dikonversi menjadi 0,549498.

4.1.3 Perancangan model TCN

Setelah dilakukan normalisasi data dan pembagian menjadi data latih dan data uji, langkah selanjutnya adalah membentuk dataset dalam format yang sesuai untuk pelatihan model TCN. Proses ini dimulai dengan mengubah data deret waktu menjadi format input-output menggunakan sliding window, dalam hal ini setiap window atau time step terdiri dari 30 hari data historis sebagai input dan 1 nilai target yaitu harga penutupan pada hari ke-31 sebagai output. Selanjutnya data input dibentuk ulang (*reshape*) menjadi format 3 dimensi yaitu [*samples, time step, features*] agar sesuai dengan kebutuhan arsitektur TCN. Dalam kasus ini, input memiliki dimensi 2134 sampel dari data latih, 30 langkah waktu (*time step*) dan 1 fitur target, yaitu target predikis harga penutupan Ethereum. Setelah proses pembentukan dataset selesai, selanjutnya adalah membuat model TCN.



```
# buat fungsi build_tcn untuk membangun model TCN
def build_tcn(input_shape):
    model = Sequential()
    model.add(Input(shape=input_shape))
    model.add(TCN(
        nb_filters=64,
        kernel_size=3,
        dilations=[1, 2, 4, 8],
        return_sequences=False,
        use_skip_connections=True
    ))
    model.add(Dropout(0.2))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mse')
    return model

# memanggil fungsi build_tcn
model = build_tcn(input_shape=(X_train.shape[1], X_train.shape[2]))
model.summary()
```

Gambar 4. 2 Model sequential TCN

Pada gambar 4.2 model dibangun menggunakan fungsi `build_tcn`, yang dirancang untuk menginisialisasi arsitektur *Temporal Convolutional Network* (TCN) dalam kerangka *Sequential Keras*. Fungsi tersebut menerima parameter `input_shape`, yaitu bentuk input data tiga dimensi [*samples, time steps, features*], yang dalam hal ini berukuran (2134, 30, 1). Layer pertama pada model adalah *input*, yang mendefinisikan bentuk input sesuai dengan data pelatihan. Selanjutnya, *layer* utama adalah TCN, yang menggunakan 64 buah filter konvolusional dan ukuran

kernel sebesar 3 untuk mengekstraksi pola jangka pendek dan panjang dari data historis. Skema dilasi [1, 2, 4, 8] digunakan untuk memperluas *receptive field*, memungkinkan model mengamati urutan hingga ± 30 hari sebelumnya tanpa kehilangan informasi temporal. Parameter `return_sequences=False` digunakan karena model hanya memprediksi satu nilai di masa depan, yaitu harga penutupan pada hari berikutnya. *Skip connection* diaktifkan untuk membantu aliran gradien saat pelatihan. Sebagai bentuk regularisasi, ditambahkan lapisan *Dropout* dengan rasio 0.2 guna mengurangi risiko *overfitting*. Terakhir, lapisan *Dense* dengan satu neuron digunakan sebagai output layer untuk menghasilkan prediksi harga target. Model ini dikompilasi dengan *optimizer* Adam dan fungsi kerugian *Mean Squared Error* (MSE), yang sesuai untuk masalah regresi deret waktu. Model yang telah dirancang ini siap dilatih menggunakan data yang telah diproses sebelumnya. Berikut adalah model summary dari model TCN.

Model: "sequential_3"

Layer (type)	Output Shape	Param #
tcn_3 (TCN)	(None, 64)	86,848
dropout_3 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

Total params: 86,913 (339.50 KB)
 Trainable params: 86,913 (339.50 KB)
 Non-trainable params: 0 (0.00 B)

Gambar 4. 3 Ringkasan arsitektur model sequential TCN

Gambar tersebut menunjukkan ringkasan arsitektur model Sequential yang digunakan untuk prediksi harga Ethereum menggunakan *Temporal Convolutional Network* (TCN). Arsitektur ini terdiri dari tiga lapisan utama, yaitu lapisan TCN, *Dropout*, dan *Dense*. Lapisan TCN merupakan inti dari model yang berfungsi menangkap pola temporal dari data historis, dan menghasilkan 64 unit keluaran dengan total 86.848 parameter yang dapat dilatih. Selanjutnya, lapisan *Dropout* digunakan sebagai mekanisme regularisasi untuk mencegah *overfitting*, tanpa memiliki parameter yang dilatih. Terakhir, lapisan *Dense* berperan sebagai output layer yang menghasilkan satu nilai prediksi harga penutupan, dengan jumlah

parameter sebesar 65. Secara keseluruhan, model ini memiliki total 86.913 parameter.

Selama proses pelatihan model, data uji digunakan sebagai *validation data* untuk memantau kinerja model secara berkala terhadap data yang tidak dilibatkan dalam pelatihan langsung. Tujuan dari penggunaan data validasi ini adalah untuk mengetahui apakah model belajar secara general dan tidak hanya menghafal data latih (*overfitting*). Untuk mencegah *overfitting*, digunakan mekanisme *EarlyStopping*, yaitu metode yang secara otomatis menghentikan pelatihan jika nilai *validation loss* (*val_loss*) tidak mengalami penurunan selama beberapa *epoch* berturut-turut. Dalam eksperimen ini, nilai *patience* yang digunakan adalah 10 dan 15. Artinya, pelatihan akan dihentikan lebih awal jika dalam 10 atau 15 *epoch* terakhir tidak terjadi peningkatan kinerja model pada data validasi. Selain *patience*, eksperimen juga dilakukan dengan berbagai konfigurasi jumlah *epoch* (30, 60, 90, dan 120) dan ukuran *batch* (32 dan 64). Tujuan variasi ini adalah untuk mengeksplorasi kombinasi pelatihan terbaik yang memberikan hasil prediksi paling optimal. Berikut adalah tabel konfigurasi yang digunakan selama pelatihan model:

Tabel 4. 7 Tabel pengujian model

<i>Patience</i>	<i>Epoch</i>	<i>Batch_size</i>
10	30	32
10	45	32
10	60	32
15	90	64
15	100	64
15	120	64

4.1.4 Denormalisasi

Setelah model melakukan prediksi pada data uji (x_{test}) yang sebelumnya telah dinormalisasi menggunakan *minmax scaler*. Langkah selanjutnya adalah mengembalikan nilai-nilai hasil prediksi ($test_predict$) dan nilai target sebenarnya (y_{test}) ke skala asli. Hal ini dilakukan karena model belajar dan memprediksi data dalam skala 0 dan 1, sehingga hasil yang keluar tidak dapat langsung diartikan sebagai harga aslinya. Agar hasil prediksi dapat diinterpretasikan dalam konteks

nilai harga Ethereum yang sebenarnya, maka nilai-nilai tersebut perlu dikembalikan ke skala aslinya. Proses ini disebut denormalisasi. Denormalisasi dilakukan dengan menggunakan fungsi `inverse_transform()` dari objek `MinMaxScaler` yang telah digunakan saat normalisasi. Fungsi ini akan mengembalikan nilai dari skala `[0, 1]` ke skala harga aktual sesuai data awal. Langkah-langkah denormalisasi dilakukan sebagai berikut:

```
test_predict_actual = scaler.inverse_transform(test_predict)
y_test_actual = scaler.inverse_transform(y_test.reshape(-1, 1))
```

Dengan menggunakan metode ini, kita dapat membandingkan secara langsung antara harga prediksi dan harga aktual dalam satuan yang sama (USD).

4.1.5 Pengujian model

Pada prose pengujian model menerapkan *EarlyStopping* dari *library* Keras sebagai metode untuk mencegah *overfitting*. *EarlyStopping* berfungsi untuk menghentikan proses pelatihan secara otomatis apabila nilai *validation loss* (*val_loss*) tidak mengalami penurunan selama beberapa *epoch* atau *patience* tertentu dan memastikan bobot model akan dikembalikan ke kondisi terbaiknya menggunakan `restore_best_weights=True`. Model dilatih menggunakan data latih (*X_train*, *y_train*) dan divalidasi menggunakan data uji (*X_test*, *y_test*) selama beberapa *epoch*.

```
early_stop = EarlyStopping(monitor='val_loss',
                           patience=10,
                           restore_best_weights=True)

history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=30,
    batch_size=32,
    callbacks=[early_stop],
    verbose=1
)
```

Gambar 4. 4 Pelatihan TCN *patience* 10, *epochs* 30, dan *batch size* 32

Pada gambar 4.4 digunakan nilai *patience* sebesar 10, artinya proses pelatihan akan berhenti secara otomatis apabila nilai *validation loss* tidak mengalami perbaikan selama 10 *epoch* berturut-turut. Proses pelatihan dilakukan selama

maksimum 30 *epoch* dengan *batch_size* sebesar 32. Berikut adalah grafik *loss* selama pelatihan model yang ditunjukkan pada gambar 4.5.



Gambar 4. 5 Grafik loss pelatihan TCN *patience* 10, *epochs* 30, dan *batch size* 32

Gambar 4.5 menunjukkan grafik *loss* selama proses pelatihan model dengan *patience* 10, *epochs* 30, dan *batch size* 32. Dari grafik terlihat bahwa *Train Loss* mengalami penurunan tajam di awal pelatihan dan kemudian menurun secara perlahan hingga mendekati nol. Sementara itu, *Val Loss* tetap stabil dan rendah sepanjang proses pelatihan. Pola ini menunjukkan bahwa model mampu belajar dengan baik dari data latih dan tidak mengalami *overfitting*, karena performa terhadap data validasi tetap konsisten. Pelatihan berhenti pada *epoch* ke-29 artinya model tidak menunjukkan peningkatan performa (penurunan *val_loss*) yang signifikan selama 10 *epoch* berturut-turut setelahnya.

```

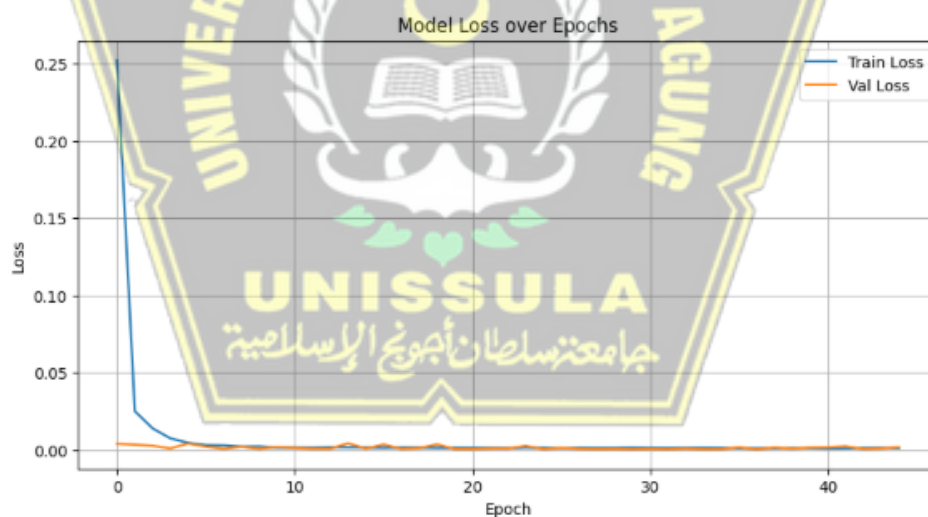
early_stop = EarlyStopping(monitor='val_loss',
                           patience=10,
                           restore_best_weights=True)

history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=45,
    batch_size=32,
    callbacks=[early_stop],
    verbose=1
)

```

Gambar 4. 6 Pelatihan TCN *patience* 10, *epochs* 45, dan *batch size* 32

Pada gambar 4.6 digunakan nilai *patience* sebesar 10, artinya proses pelatihan akan berhenti secara otomatis apabila nilai *validation loss* tidak mengalami perbaikan selama 10 *epoch* berturut-turut. Proses pelatihan dilakukan selama maksimum 45 *epoch* dengan *batch_size* sebesar 32. Berikut adalah grafik *loss* selama pelatihan model yang ditunjukkan pada gambar 4.7.



Gambar 4. 7 Grafik *loss* pelatihan TCN *patience* 10, *epochs* 45, dan *batch size* 32

Gambar 4.7 menunjukkan grafik *loss* selama proses pelatihan model dengan *patience* 10, *epochs* 45, dan *batch size* 32. Dari grafik terlihat bahwa *Train Loss* mengalami penurunan tajam di awal pelatihan dan kemudian menurun secara perlahan hingga mendekati nol. Sementara itu, *Val Loss* tetap stabil dan rendah sepanjang proses pelatihan. Pola ini menunjukkan bahwa model mampu belajar dengan baik dari data latih dan tidak mengalami *overfitting*, karena performa

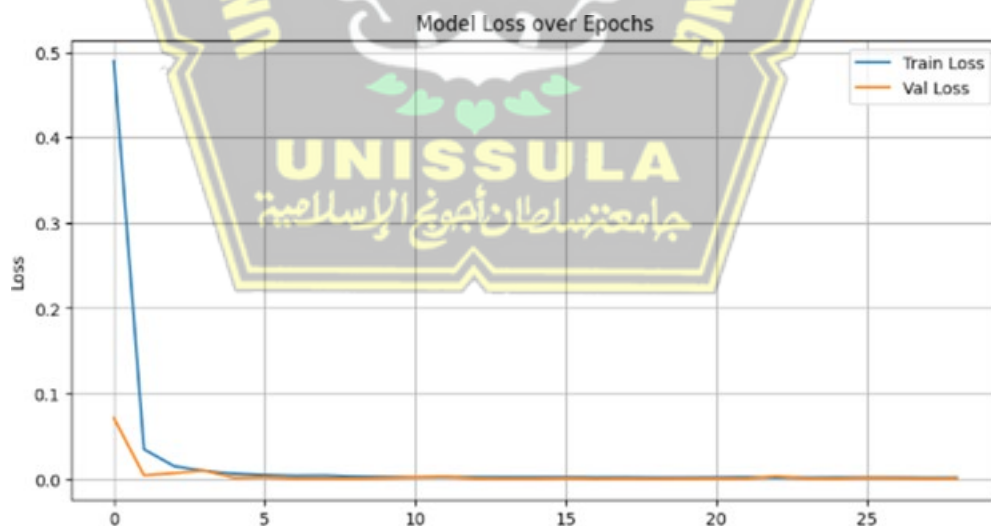
terhadap data validasi tetap konsisten. Pelatihan berjalan hingga *epoch* terakhir yaitu *epoch* ke-45.

```
early_stop = EarlyStopping(monitor='val_loss',
                           patience=10,
                           restore_best_weights=True)

history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=60,
    batch_size=32,
    callbacks=[early_stop],
    verbose=1
)
```

Gambar 4. 8 Pelatihan TCN *patience* 10, *epochs* 60, dan *batch size* 32

Pada gambar 4.8 digunakan nilai *patience* sebesar 10, artinya proses pelatihan akan berhenti secara otomatis apabila nilai *validation loss* tidak mengalami perbaikan selama 10 *epoch* berturut-turut. Proses pelatihan dilakukan selama maksimum 60 *epoch* dengan *batch_size* sebesar 32. Berikut adalah grafik *loss* selama pelatihan model yang ditunjukkan pada gambar 4.9.



Gambar 4. 9 Grafik *loss* pelatihan TCN *patience* 10, *epochs* 60, dan *batch size* 32

Gambar 4.9 menunjukkan grafik *loss* selama proses pelatihan model dengan *patience* 10, *epochs* 60, dan *batch size* 32. Dari grafik terlihat bahwa *Train Loss* mengalami penurunan tajam di awal pelatihan dan kemudian menurun secara perlahan hingga mendekati nol. Sementara itu, *Val Loss* tetap stabil dan rendah

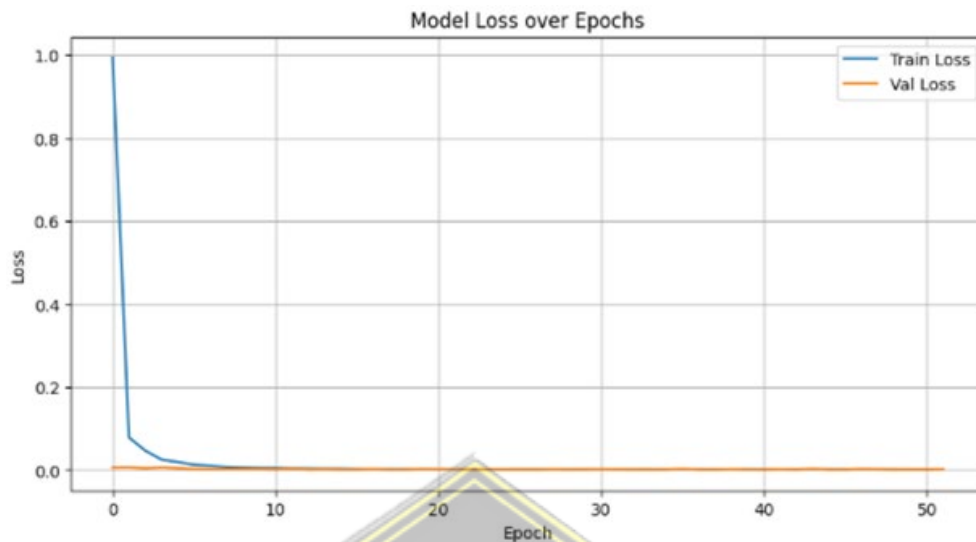
sepanjang proses pelatihan. Pola ini menunjukkan bahwa model mampu belajar dengan baik dari data latih dan tidak mengalami *overfitting*, karena performa terhadap data validasi tetap konsisten. Pelatihan berhenti pada *epoch* ke-29 artinya model tidak menunjukkan peningkatan performa (penurunan *val_loss*) yang signifikan selama 10 *epoch* berturut-turut setelahnya.

```
early_stop = EarlyStopping(monitor='val_loss',
                           patience=15,
                           restore_best_weights=True)

history = model.fit(
    x_train, y_train,
    validation_data=(x_test, y_test),
    epochs=90,
    batch_size=64,
    callbacks=[early_stop],
    verbose=1
)
```

Gambar 4. 10 Pelatihan TCN *patience* 15, *epochs* 90, dan *batch size* 64

Pada gambar 4.10 digunakan nilai *patience* sebesar 15, artinya proses pelatihan akan berhenti secara otomatis apabila nilai *validation loss* tidak mengalami perbaikan selama 15 *epoch* berturut-turut. Proses pelatihan dilakukan selama maksimum 90 *epoch* dengan *batch_size* sebesar 64. Berikut adalah grafik *loss* selama pelatihan model yang ditunjukkan pada gambar 4.11.



Gambar 4. 11 Grafik loss pelatihan TCN *patience* 15, *epochs* 90, dan *batch size* 64

Gambar 4.11 menunjukkan grafik loss selama proses pelatihan model dengan *patience* 15, *epochs* 90, dan *batch size* 64. Dari grafik terlihat bahwa *Train Loss* mengalami penurunan tajam di awal pelatihan dan kemudian menurun secara perlahan hingga mendekati nol. Sementara itu, *Val Loss* tetap stabil dan rendah sepanjang proses pelatihan. Pola ini menunjukkan bahwa model mampu belajar dengan baik dari data latih dan tidak mengalami *overfitting*, karena performa terhadap data validasi tetap konsisten. Pelatihan berhenti pada *epoch* ke-51 artinya model tidak menunjukkan peningkatan performa (penurunan *val_loss*) yang signifikan selama 15 *epoch* berturut-turut setelahnya.

```

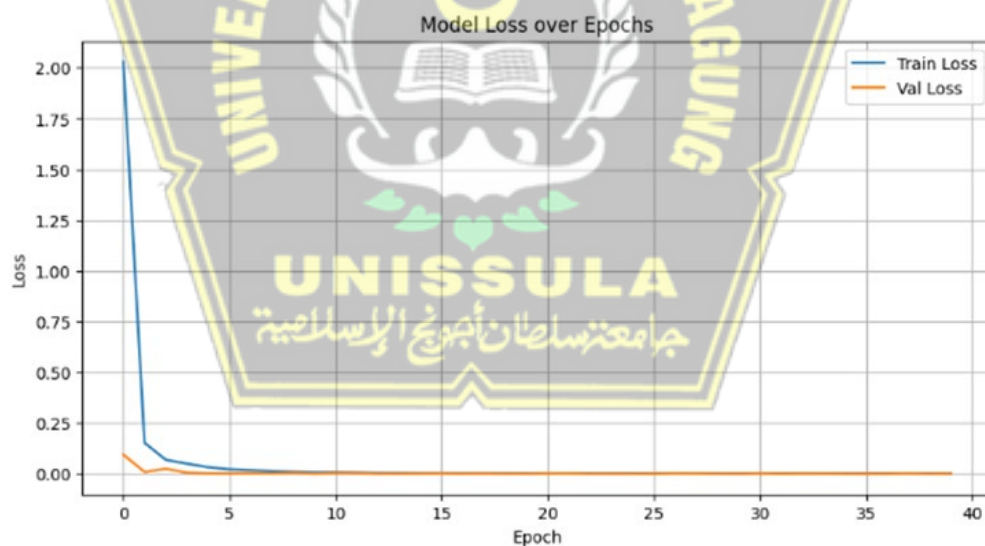
early_stop = EarlyStopping(monitor='val_loss',
                           patience=15,
                           restore_best_weights=True)

history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=100,
    batch_size=64,
    callbacks=[early_stop],
    verbose=1
)

```

Gambar 4. 12 Pelatihan TCN *patience* 15, *epochs* 100, dan *batch size* 64

Pada gambar 4.12 digunakan nilai *patience* sebesar 15, artinya proses pelatihan akan berhenti secara otomatis apabila nilai *validation loss* tidak mengalami perbaikan selama 15 *epoch* berturut-turut. Proses pelatihan dilakukan selama maksimum 100 *epoch* dengan *batch_size* sebesar 64. Berikut adalah grafik *loss* selama pelatihan model yang ditunjukkan pada gambar 4.5.



Gambar 4. 13 Grafik loss pelatihan TCN *patience* 15, *epochs* 100, dan *batch size* 64

Gambar 4.13 menunjukkan grafik loss selama proses pelatihan model dengan *patience* 15, *epochs* 100, dan *batch size* 64. Dari grafik terlihat bahwa *Train Loss* mengalami penurunan tajam di awal pelatihan dan kemudian menurun secara perlahan hingga mendekati nol. Sementara itu, *Val Loss* tetap stabil dan rendah sepanjang proses pelatihan. Pola ini menunjukkan bahwa model mampu belajar dengan baik dari data latih dan tidak mengalami *overfitting*, karena performa

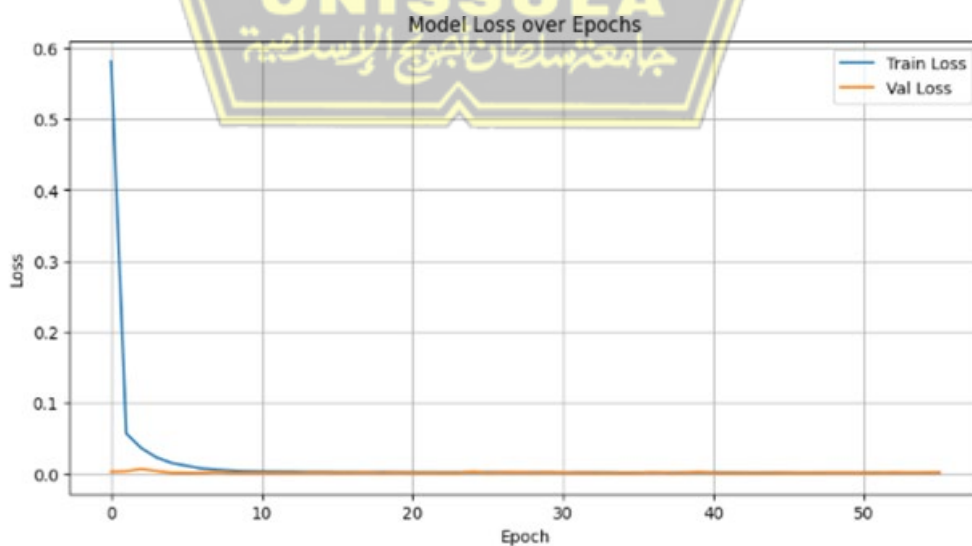
terhadap data validasi tetap konsisten. Pelatihan berhenti pada *epoch* ke-39 artinya model tidak menunjukkan peningkatan performa (penurunan *val_loss*) yang signifikan selama 15 *epoch* berturut-turut setelahnya.

```
early_stop = EarlyStopping(monitor='val_loss',
                           patience=20,
                           restore_best_weights=True)

history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=120,
    batch_size=64,
    callbacks=[early_stop],
    verbose=1)
```

Gambar 4. 14 Pelatihan TCN *patience* 20, *epochs* 120, dan *batch size* 64

Pada gambar 4.9 digunakan nilai *patience* sebesar 20, artinya proses pelatihan akan berhenti secara otomatis apabila nilai *validation loss* tidak mengalami perbaikan selama 20 *epoch* berturut-turut. Proses pelatihan dilakukan selama maksimum 120 *epoch* dengan *batch_size* sebesar 64. Berikut adalah grafik *loss* selama pelatihan model yang ditunjukkan pada gambar 4.5.



Gambar 4. 15 Grafik *loss* pelatihan TCN *patience* 15, *epochs* 120, dan *batch size* 64

Gambar 4.5 menunjukkan grafik *loss* selama proses pelatihan model dengan *patience* 20, *epochs* 120, dan *batch size* 64. Dari grafik terlihat bahwa *Train Loss* mengalami penurunan tajam di awal pelatihan dan kemudian menurun secara perlahan hingga mendekati nol. Sementara itu, *Val Loss* tetap stabil dan rendah sepanjang proses pelatihan. Pola ini menunjukkan bahwa model mampu belajar dengan baik dari data latih dan tidak mengalami *overfitting*, karena performa terhadap data validasi tetap konsisten. Pelatihan berhenti pada *epoch* ke-53 artinya model tidak menunjukkan peningkatan performa (penurunan *val_loss*) yang signifikan selama 20 *epoch* berturut-turut setelahnya.

4.1.6 Evaluasi model

Evaluasi model dilakukan dengan menguji berbagai kombinasi parameter pelatihan, yaitu nilai *patience*, jumlah maksimum *epoch*, serta *batch size*, dengan tujuan untuk mengetahui pengaruh konfigurasi tersebut terhadap kinerja model prediksi harga. Evaluasi dilakukan menggunakan dua metrik utama yaitu *Mean Absolute Error* (MAE) dan *Mean Absolute Percentage Error* (MAPE). Setelah dilakukan beberapa pengujian yang sudah dilakukan, didapatkan hasil berikut:

Tabel 4. 8 Tabel hasil evaluasi

<i>Patience</i>	<i>Epoch</i>	<i>Early Stop</i>	<i>Batch_size</i>	MAE	MAPE
10	30	29	32	80,92	2,85 %
10	45	43	32	75,70	2,67%
10	60	23	32	84,56	2,98 %
15	90	51	64	78,61	2,78 %
15	100	39	64	89,72	3,19 %
15	120	53	64	78,76	2,78 %

Berdasarkan tabel 4.8 diketahui bahwa model dengan konfigurasi *patience* 10, *epoch* maksimum 45, dan *batch size* 32 memberikan hasil terbaik, dengan MAE sebesar 75,70 dan MAPE sebesar 2,67%. Model ini berhenti pada *epoch* ke-45 atau *epoch* terakhir. Hasil ini menunjukkan bahwa model mampu belajar secara efektif dan efisien dengan melatih *epoch* sampai batas maksimum.

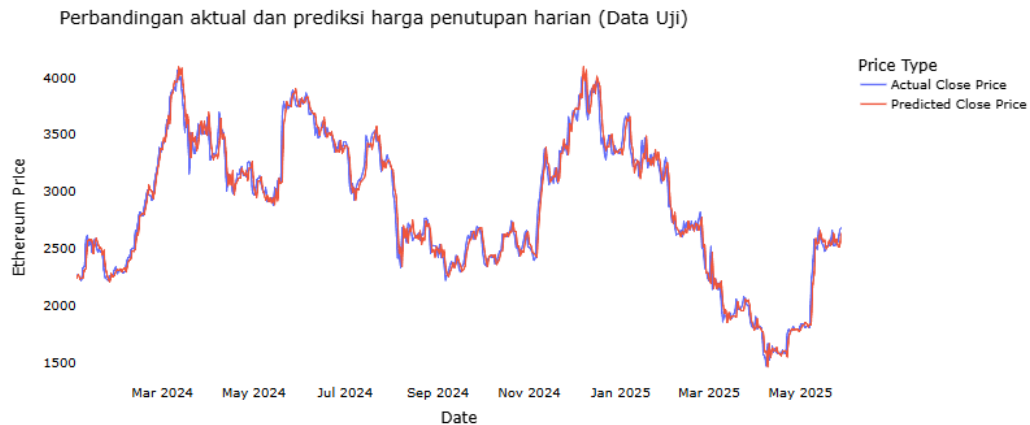
Pada konfigurasi *epoch* maksimum 30, model berhenti pada *epoch* ke-29 dengan MAE sebesar 80,92 dan MAPE 2,85%, masih menunjukkan performa yang baik, namun sedikit lebih rendah dibandingkan konfigurasi sebelumnya. Sementara pada konfigurasi *epoch* 60, pelatihan justru berhenti lebih awal pada *epoch* ke-23, dengan MAE yang meningkat menjadi 84,56 dan MAPE 2,98%, mengindikasikan bahwa peningkatan jumlah *epoch* tidak selalu meningkatkan performa model.

Untuk pengujian dengan *patience* 15 dan *batch size* 64, hasil menunjukkan bahwa model mampu mencapai MAE dan MAPE yang cukup stabil. Konfigurasi dengan *epoch* maksimum 90 dan 120 masing-masing berhenti pada *epoch* ke-51 dan ke-53, menghasilkan MAE sebesar 78,61 dan 78,76, serta MAPE sebesar 2,78% untuk keduanya. Sementara konfigurasi dengan *epoch* maksimum 100 justru menghasilkan MAE tertinggi 89,72 dan MAPE 3,19%, yang menunjukkan bahwa penambahan *epoch* tanpa peningkatan validasi justru menyebabkan performa memburuk.

Secara keseluruhan, hasil evaluasi menunjukkan bahwa pengaturan parameter pelatihan berpengaruh terhadap performa model. Penggunaan *early stopping* terbukti efektif dalam mencegah *overfitting* dan menghemat waktu pelatihan. Konfigurasi dengan *patience* 10, *epoch* 45, dan *batch size* 32 dapat disimpulkan sebagai pengaturan optimal pada eksperimen ini karena menghasilkan kesalahan prediksi terendah berdasarkan metrik evaluasi.

4.2 Hasil Prediksi

Berikut merupakan grafik dari hasil prediksi harga penutupan harian ethereum menggunakan model TCN.



Gambar 4. 16 Grafik perbandingan aktual dan prediksi harga penutupan harian ethereum

Gambar 4.16 tersebut menampilkan grafik perbandingan antara harga penutupan aktual dan hasil prediksi model terhadap data uji harian Ethereum. Grafik menunjukkan dua garis, yaitu garis biru yang merepresentasikan harga penutupan aktual dan garis merah yang menunjukkan harga penutupan hasil prediksi oleh model TCN. Dari visualisasi tersebut dapat dilihat bahwa kedua garis memiliki pola yang sangat mirip, mengikuti tren naik-turun yang serupa sepanjang rentang waktu dari awal Maret 2024 hingga Mei 2025. Hal ini menunjukkan bahwa model memiliki kemampuan yang baik dalam menangkap pola temporal dari data historis dan mampu melakukan prediksi dengan akurasi yang tinggi. Tidak terdapat deviasi signifikan antara hasil prediksi dan nilai aktual, yang memperkuat kesimpulan bahwa model telah terlatih dengan baik dan tidak mengalami *overfitting*. Keseluruhan grafik ini menjadi bukti visual atas performa model dalam merepresentasikan dinamika harga Ethereum secara realistis.

Setelah model TCN selesai dilatih, dilakukan proses prediksi untuk memproyeksikan harga Ethereum satu hari ke depan. Prediksi dilakukan dengan mengambil window data terakhir dari data uji sebanyak `time_step` sebagai input awal. Data tersebut kemudian dirapikan dalam bentuk tiga dimensi (1, `time_step`, 1) sesuai dengan bentuk input yang dibutuhkan oleh arsitektur TCN. Proses prediksi dilakukan secara iteratif, namun dalam kasus ini hanya dilakukan untuk satu hari (`pred_days = 1`). Model TCN yang telah dilatih digunakan untuk memprediksi harga pada hari berikutnya, kemudian hasilnya disimpan ke dalam list `lst_output`. Nilai hasil prediksi ini masih berada dalam bentuk yang ternormalisasi (rentang 0–1),

sehingga dilakukan proses `inverse_transform` menggunakan objek `MinMaxScaler` untuk mengembalikannya ke skala harga sebenarnya. Setelah itu, ditentukan tanggal prediksi berdasarkan tanggal terakhir dari dataset asli, kemudian dibuat sebuah `DataFrame` yang memuat tanggal dan nilai prediksi harga untuk 1 hari ke depan.

Adapun hasil prediksi ditampilkan pada Tabel 4., yang menunjukkan nilai prediksi harga penutupan Ethereum untuk tanggal [tanggal hasil prediksi], misalnya sebesar 2.679,98 USD.

Tabel 4. 9 Hasil prediksi 1 hari selanjutnya

Tanggal	Hasil prediksi Ethereum
2025-05-29	2679.98

4.3 Dashboard Streamlit

Berikut merupakan tampilan sistem untuk memprediksi harga penutupan ethereum menggunakan model yang sudah diuji.



Gambar 4. 17 Tampilan utama sistem

Gambar 4.17 merupakan tampilan memungkinkan pengguna untuk melakukan prediksi harga penutupan ethereum menggunakan model *Temporal Convolutional Network*. Pengguna dapat mengunggah file CSV berisi data historis harga Ethereum yang akan digunakan sebagai input untuk proses prediksi. Setelah file berhasil diunggah, sistem akan membaca dan memvalidasi struktur data,

memastikan kolom-kolom yang dibutuhkan seperti *Date*, *Open*, *High*, *Low*, *Close*, dan *volume* tersedia.

Setelah dataset divalidasi sistem, selanjutnya akan ditampilkan data 30 hari terakhir dari dataset seperti yang ditunjukkan pada gambar 4.18

Unggah file CSV yang berisi data harga kripto Ethereum. Harga harus dalam satuan USD.

Upload file CSV Anda

Drag and drop file here
Limit 200MB per file • CSV

Browse files

eth_usd_2018_2025.csv
246.8KB

Dataset berhasil dimuat.

Data 30 Hari Terakhir

	Date	Open	High	Low	Close	Volume
2675	2025-04-29 00:00:00	1798.9122	1842.0331	1782.2813	1799.1757	14734257744
2676	2025-04-30 00:00:00	1799.2091	1816.6899	1736.1429	1793.7754	14810431362
2677	2025-05-01 00:00:00	1794.041	1872.9396	1793.1511	1839.2216	15861442721
2678	2025-05-02 00:00:00	1839.1748	1869.53	1814.7305	1842.7079	13570171872
2679	2025-05-03 00:00:00	1842.7181	1848.2468	1812.8157	1833.8397	8344452620
2680	2025-05-04 00:00:00	1833.6472	1849.9985	1804.4951	1808.5942	8918979887
2681	2025-05-05 00:00:00	1808.705	1832.0508	1783.3148	1819.6975	11389336489
2682	2025-05-06 00:00:00	1819.734	1820.7966	1753.3154	1815.0875	13206516310
2683	2025-05-07 00:00:00	1815.0189	1849.6584	1788.6874	1811.6067	50540755039
2684	2025-05-08 00:00:00	1811.5464	2222.2056	1807.4569	2206.5146	37777245241

Gambar 4. 18 Tampilan data 30 hari terakhir

Gambar 4.18 memperlihatkan antarmuka setelah pengguna berhasil mengunggah file dataset berformat CSV yang berisi data historis harga Ethereum. Tampilan ini menampilkan Data 30 Hari Terakhir dalam bentuk tabel yang memungkinkan membantu pengguna untuk memverifikasi bahwa data telah dimuat dengan benar dan memungkinkan pengguna untuk melihat sekilas dinamika harga Ethereum dalam periode terbaru. Tabel ini menampilkan informasi harga Ethereum selama 30 hari terakhir dari dataset, dengan kolom sebagai berikut:

- *Date*: Tanggal.
- *Open*: Harga pembukaan Ethereum pada hari tersebut.
- *High*: Harga tertinggi Ethereum dalam satu hari.
- *Low*: Harga terendah dalam satu hari.
- *Close*: Harga penutupan Ethereum pada hari tersebut.
- *Volume*: Jumlah *volume* perdagangan Ethereum dalam USD.



Gambar 4. 19 Tabel perbandingan harga penutupan aktual dan prediksi 30 hari terakhir

Gambar 4.19 menampilkan hasil keluaran dari model prediksi harga penutupan Ethereum. Setelah proses pelatihan dan pengujian model selesai, sistem menampilkan tabel perbandingan antara harga aktual dan harga prediksi pada 30 hari terakhir, dengan fokus pada kolom *close price* (harga penutupan).



Gambar 4. 20 Grafik visualisasi harga penutupan aktual dan prediksi 30 hari terakhir

Gambar 4.20 memperlihatkan tampilan tabel hasil prediksi harga penutupan Ethereum (*Close Price*) yang dihasilkan oleh model, dibandingkan dengan harga aktual pada 30 hari terakhir dari dataset.

Prediksi Harga 1 Hari ke Depan [↗](#)

	Tanggal	Harga Prediksi (USD)
0	2025-05-29 00:00:00	2693.3

Gambar 4. 21 Hasil prediksi satu hari kedepan

Gambar 4.21 menampilkan hasil prediksi harga penutupan Ethereum (Close Price) untuk 1 hari ke depan menggunakan model *Temporal Convolutional Network* (TCN) yang telah dilatih sebelumnya.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penerapan algoritma *Temporal Convolutional Network* (TCN) dalam memprediksi harga penutupan harian mata uang kripto Ethereum, dapat disimpulkan bahwa algoritma TCN mampu memberikan performa yang baik dalam memprediksi harga penutupan harian Ethereum. Hal ini dibuktikan melalui hasil evaluasi model menggunakan metrik *Mean Absolute Error* (MAE) dan *Mean Absolute Percentage Error* (MAPE), dengan nilai terbaik yaitu MAE sebesar 75,70 dan MAPE sebesar 2,67%.

Hasil terbaik tersebut diperoleh pada konfigurasi parameter pelatihan dengan *patience* 10, *epoch* 45, dan *batch size* 32. Secara keseluruhan, evaluasi menunjukkan bahwa pengaturan parameter pelatihan memiliki pengaruh yang signifikan terhadap kinerja model. Selain itu, penggunaan teknik *early stopping* terbukti efektif dalam mencegah *overfitting* serta menghemat waktu pelatihan, dengan tetap mempertahankan akurasi yang optimal. Oleh karena itu, konfigurasi tersebut dapat disimpulkan sebagai pengaturan paling optimal dalam eksperimen ini karena menghasilkan tingkat kesalahan prediksi yang paling rendah.

5.2 Saran

Berdasarkan hasil penelitian, disarankan agar pengembangan model prediksi harga Ethereum di masa mendatang mempertimbangkan berbagai faktor yang memengaruhi tren harga mata uang kripto. Selain itu, eksplorasi lebih lanjut terhadap teknik konfigurasi parameter dapat dilakukan untuk meningkatkan performa model. Penggabungan model TCN dengan model lain, dengan memanfaatkan *dilated causal convolution* sebagai inti arsitektur TCN, juga dapat menjadi pendekatan yang potensial.

DAFTAR PUSTAKA

- Abdurohim, & Irfan, M. (2024). *Cryptocurrency dan Stabilitas Sistem Keuangan : Tinjauan Literatur*. 21, 64–94.
- Adel Musallam, M., Baptista, R., Al Ismaeil, K., & Aouada, D. (2019). Temporal 3D human pose estimation for action recognition from arbitrary viewpoints. *Proceedings - 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019, December*, 253–258. <https://doi.org/10.1109/CSCI49370.2019.00052>
- Aisyah, S., Angraini, Y., Sadik, K., Sartono, B., & Dito, G. A. (2024). *Technical Analysis of the Indonesian Stock Market with Gated Recurrent Unit and Temporal Convolutional Network*. 12(2), 187–196.
- Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning. *Karya Ilmiah Mahasiswa Bertauhid (KARIMAH TAUHID)*, 2(1), 1–6.
- Arnolus Juantri E. Oktavianus, Lamhot Naibaho, & Djoys Anneke Rantung. (2023). Pemanfaatan Artificial Intelligence pada Pembelajaran dan Asesmen di Era Digitalisasi. *Jurnal Kridatama Sains Dan Teknologi*, 05(2), 473–476.
- Atmaja, D. M. U., & Hakim, A. R. (2022). Peramalan Harga Mata Uang Kripto Solana Menggunakan Metode Support Vector Regression (Svr). *Jurnal Media Elektro*, XI(2), 97–104. <https://doi.org/10.35508/jme.v0i0.8117>
- Aysa, I. R. (2021). Tantangan Transformasi Digital Bagi Kemajuan Perekonomian Indonesia. *Jurnal At-Tamwil: Kajian Ekonomi Syariah*, 3(2), 140–153. <https://doi.org/10.33367/at.v2i3.1458>
- Baharani, A. W., Apriza, Z., Mutmaina, N. A., & Sutabri, T. (2024). IJM: Indonesian Journal of Multidisciplinary Perbandingan Kinerja Mata Uang Kripto Utama: Bitcoin vs Ethereum. *IJM: Indonesian Journal of Multidisciplinary*, 2, 138–145. <https://journal.csspublishing/index.php/ijm>
- Bai, S., Kolter, J. Z., & Koltun, V. (2018). *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. <http://arxiv.org/abs/1803.01271>
- Bastian Sianturi, T., Cholissodin, I., & Yudistira, N. (2023). Penerapan Algoritma Long Short-Term Memory (LSTM) berbasis Multi Fungsi Aktivasi Terbobot dalam Prediksi Harga Ethereum. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7(3), 1101–1107. <http://j-ptiik.ub.ac.id>
- Budiprasetyo, G., Hani'ah, M., & Aflah, D. Z. (2023). Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM). *Jurnal Nasional Teknologi dan Sistem Informasi*, 8(3), 164–172. <https://doi.org/10.25077/teknosi.v8i3.2022.164-172>

- Dai, S., Song, K. F., Wang, Y. L., & Zhang, P. J. (2023). Two-Dimensional Space Turntable Pitch Axis Trajectory Prediction Method Based on Sun Vector and CNN-LSTM Model. *Applied Sciences (Switzerland)*, 13(8). <https://doi.org/10.3390/app13084939>
- Digo Saputra, R., & Hatta Fudholi, D. (2023). *Model Mobile untuk Deteksi Objek pada On-Shelf Availability Produk Retail*.
- Duta Pratama, F., & Rahmawati Gustini, D. (2023). 23-Moderasi-0101-464 (1). 2022, 1–17. <https://doi.org/10.11111/nusantara.xxxxxxx>
- Fajou, J., & McCarren, A. (2021). Forecasting Gold Prices Using Temporal Convolutional Networks. *CEUR Workshop Proceedings*, 3105, 248–259.
- Fatah, M. A., Dikananda, A. R., & Rifai, A. (2025). *Support Vector Regression to Improve Ethereum Price Prediction for Trading Strategies*. 4(2).
- Fungki Wahyu, & Billy Hendrik. (2023). Perbandingan Algoritma Time Series Dan Fuzzy Inference System Dalam Analisis Data Deret Waktu. *Jurnal Penelitian Teknologi Informasi dan Sains*, 1(3), 16–24. <https://doi.org/10.54066/jptis.v1i3.711>
- Goa Mere, J. H. B. (2023). *Pembuatan dan pengujian token crypto pada jaringan mainnet menggunakan smart contract binance smart chain (bsc) dan remix.ethereum*. 14(c), 82–89.
- Hao, J., & Liu, F. (2024). Improving long-term multivariate time series forecasting with a seasonal-trend decomposition-based 2-dimensional temporal convolution dense network. *Scientific Reports*, 14(1), 1–14. <https://doi.org/10.1038/s41598-024-52240-y>
- Kanedi, I., & Jumadi, J. (2024). *Implementasi Machine Learning Untuk Prediksi Penjualan Oli Shell Pada CV. Harapan Karya Mandiri Bengkulu*. 20(2), 534–541.
- Lara-Benítez, P., Carranza-García, M., Luna-Romera, J. M., & Riquelme, J. C. (2020). Temporal convolutional networks applied to energy-related time series forecasting. *Applied Sciences (Switzerland)*, 10(7), 1–17. <https://doi.org/10.3390/app10072322>
- Liapis, C. M., & Kotsiantis, S. (2023). Temporal Convolutional Networks and BERT-Based Multi-Label Emotion Analysis for Financial Forecasting. *Information (Switzerland)*, 14(11). <https://doi.org/10.3390/info14110596>
- Mo, J., Wang, R., Cao, M., Yang, K., Yang, X., & Zhang, T. (2023). A hybrid temporal convolutional network and Prophet model for power load forecasting. *Complex and Intelligent Systems*, 9(4), 4249–4261. <https://doi.org/10.1007/s40747-022-00952-x>
- Nurhakiki, J., & Yahfizham, Y. (2024). Studi Kepustakaan: Pengenalan 4 Algoritma Pada Pembelajaran Deep Learning Beserta Implikasinya. *Jurnal*

- Purwono, P., Dewi, P., Wibisono, S. K., & Dewa, B. P. (2022). Model Prediksi Otomatis Jenis Penyakit Hipertensi dengan Pemanfaatan Algoritma Machine Learning Artificial Neural Network. *Insect (Informatics and Security): Jurnal Teknik Informatika*, 7(2), 82–90. <https://doi.org/10.33506/insect.v7i2.1828>
- Rizkillloh, M. F., & Widiyanesti, S. (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>
- Rolando, B., Amin, A., Rahmat, Zuwardi, & Izmuhammad, I. (2024). *Memahami Nilai Tukar Kripto dalam Ekonomi Digital: Pendekatan Investasi di Masa Kini*. 1(6), 560–571.
- Saepudin, D., Egi Shidqi Rabbani, Dio Navialdy, & Didit Adytia. (2024). Water Level Time Series Forecasting Using TCN Study Case in Surabaya. *Jurnal Online Informatika*, 9(1), 61–69. <https://doi.org/10.15575/join.v9i1.1312>
- Setiyani, L., Indahsari, A. N., & Roestam, R. (2023). Analisis Prediksi Level Obesitas Menggunakan Perbandingan Algoritma Machine Learning dan Deep Learning. *JTERA (Jurnal Teknologi Rekayasa)*, 8(1), 139. <https://doi.org/10.31544/jtera.v8.i1.2022.139-146>
- Wardhana, C. S. (2024). Eksplorasi Fundamental Cryptocurrency dalam Volatilitas Harga. *Jurnal Syntax Admiration*, 5(4), 1040–1053. <https://doi.org/10.46799/jsa.v5i4.1094>
- Wathani, M. N., Bagja, A., Rodi, M., Amri, Z., & Zulkipli. (2025). *Penerapan Temporal Convolution Network (TCN) dalam Memprediksi Harga Saham PT Bank Central Asia Tbk*. 6(1).
- Zainudin Hasan, Wiryadi Wiryadi, Arkaan Fadhlurrahman, Muhammad Dimas, & Ronald Dzaky Al Jabbar. (2024). Regulasi Penggunaan Teknologi Blockchain Dan Mata Uang Kripto Sebagai Tantangan Di Masa Depan Dalam Hukum Siber. *Birokrasi: JURNAL ILMU HUKUM DAN TATA NEGARA*, 2(2), 55–69. <https://doi.org/10.55606/birokrasi.v2i2.1163>
- Zuo, C., Wang, J., Liu, M., Deng, S., & Wang, Q. (2023). An Ensemble Framework for Short-Term Load Forecasting Based on TimesNet and TCN. *Energies*, 16(14). <https://doi.org/10.3390/en16145330>