

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pasar saham, yang umumnya dipandang sebagai tempat untuk berinvestasi dan meraih potensi keuntungan (atau kerugian) bagi masyarakat umum, didasari oleh bidang analisis pasar yang kompleks (Szydlowski & Chudziak, 2024). Dalam pasar saham terdapat istilah investasi. Investasi adalah suatu cara atau proses penundaan konsumsi dari saat ini untuk masa depan yang akan datang yang mana terdapat risiko ketidakpastian. Saham merupakan surat berharga atau instrumen pasar keuangan yang menjadi bukti kepemilikan seseorang atas suatu perusahaan. Semakin besar saham yang dimiliki, maka semakin besar kekuasaannya di perusahaan tersebut (Idham et al., 2022).

Prediksi harga saham merupakan topik penting dalam bidang keuangan (Chen, 2025). Prediksi yang akurat sangat penting bagi investor dan pedagang dalam membuat keputusan yang tepat saat membeli dan menjual saham. Selain itu, peramalan yang tepat juga mendukung perusahaan dalam pengambilan keputusan dan penyusunan strategi investasi (Cebi & Bahsoun, 2024).

Upaya untuk meramalkan harga saham secara akurat telah menjadi tantangan bagi banyak pihak, karena sifat pasar saham yang dinamis membuat prediksi pergerakan di masa depan sulit dilakukan secara konsisten dan tepat (Cebi & Bahsoun, 2024). Mengingat ketidakpastian dari variabilitas yang melekat dalam pasar saham, meramalkan tren pasar secara akurat merupakan tantangan besar. Ketidakpastian ini berasal dari berbagai faktor, seperti indikator ekonomi, peristiwa geopolitik, sentimen pasar, dan perilaku investor, yang seluruhnya memperumit proses pemodelan pergerakan harga saham (Xiao et al., 2024).

Algoritma pembelajaran mesin (*machine learning*) canggih mampu mengatasi tantangan ini dengan memproses data historis maupun *real-time* dalam jumlah besar, mengungkap pola-pola tersembunyi, dan memberikan prediksi yang lebih akurat dibandingkan metode tradisional. Pemanfaatan teknik-teknik canggih ini dapat meningkatkan akurasi prediksi dan mendukung pengambilan keputusan yang lebih baik dalam lingkungan pasar saham yang dinamis (Xiao et al., 2024).

Belakangan ini, model Transformer menjadi sangat populer. Awalnya, model ini dirancang untuk tugas pemrosesan bahasa alami (*Natural Language Processing/NLP*), namun kini juga digunakan dalam peramalan deret waktu. NLP adalah bidang yang berfokus pada interaksi antara komputer dan bahasa manusia (alami). NLP memungkinkan mesin untuk memahami, menafsirkan, dan menghasilkan bahasa manusia. Tujuan utama dari NLP mencakup pengenalan suara, penerjemahan bahasa, dan analisis sentimen (Song et al., 2025).

Model Transformer menggunakan mekanisme perhatian (*attention mechanism*), yang membantu model untuk fokus pada bagian-bagian penting dari urutan input. Banyak penelitian menunjukkan bahwa Transformer dapat mengungguli model tradisional seperti LSTM (*Long Short-Term Memory*) dalam permasalahan peramalan. Oleh karena itu, semakin banyak pihak yang menerapkan Transformer untuk prediksi harga saham (Chen, 2025).

Tesla merupakan perusahaan teknologi yang mengalami pertumbuhan pesat dalam beberapa tahun terakhir dan memiliki kekuatan serta sumber daya yang unggul di bidang energi baru dan inovasi otomotif. Hal ini menjadikan Tesla memiliki pangsa pasar saham yang signifikan (Jiang, 2023). Prediksi harga saham Tesla menjadi kasus menarik karena perusahaan ini merupakan salah satu pemimpin dalam industri kendaraan listrik dan energi terbarukan. Saham Tesla mencerminkan dinamika pasar yang kompleks dan dipengaruhi oleh tren teknologi global (Saputra et al., 2024).

Meskipun telah banyak penelitian yang mengkaji prediksi harga saham Tesla menggunakan berbagai metode seperti LSTM, KNN (*K-Nearest Neighbour*), Regresi Linier (Li, 2024), ARIMA, XGBoost, maupun metode hybrid lainnya (Jiang, 2023; Saputra et al., 2024), sejauh ini belum ditemukan kajian yang secara eksplisit dan komprehensif menggunakan algoritma Transformer dalam memprediksi harga saham Tesla. Padahal, model Transformer telah menunjukkan kinerja yang unggul dalam berbagai tugas peramalan deret waktu dibandingkan dengan model tradisional seperti LSTM dan GRU (Chen, 2025). Kurangnya kajian ini menciptakan kesenjangan literatur, khususnya dalam konteks pengaplikasian Transformer untuk saham sektor energi baru seperti Tesla, yang memiliki volatilitas tinggi berdasarkan grafik.

Berdasarkan uraian latar belakang dan penelitian sebelumnya, peneliti tertarik untuk melakukan penelitian yang berjudul **"Analisis Algoritma Transformer Dalam Prediksi Harga Saham Tesla"**. Penelitian ini diharapkan dapat berkontribusi dalam pengembangan model prediksi saham yang lebih akurat, serta memberikan wawasan bagi investor dalam pengambilan keputusan investasi yang lebih baik.

1.2 Rumusan Masalah

Rumus masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana kinerja algoritma Transformer dalam memprediksi harga saham Tesla berdasarkan data historis, jika dievaluasi menggunakan metrik *Root Mean Square Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE)?
2. Apakah algoritma Transformer mampu memberikan performa prediksi yang lebih unggul dibandingkan pendekatan-pendekatan *machine learning* yang telah digunakan dalam peramalan harga saham Tesla?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Menganalisis kinerja algoritma Transformer dalam memprediksi harga saham Tesla berdasarkan data historis, dengan melakukan evaluasi menggunakan metrik *Root Mean Square Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE).
2. Membandingkan performa prediksi algoritma Transformer dengan beberapa pendekatan *machine learning* lain yang telah digunakan dalam peramalan harga saham Tesla, untuk mengetahui apakah Transformer mampu memberikan hasil yang lebih unggul.

1.4 Landasan Teori

1.4.1 Saham

Saham merupakan salah satu instrumen paling populer di pasar keuangan. Saham dapat didefinisikan sebagai tanda penyertaan modal seseorang atau pihak (badan usaha) dalam suatu perusahaan atau perseroan terbatas. Dengan menyertakan modal tersebut, maka pihak tersebut memiliki klaim atas pendapatan perusahaan, klaim atas aset perusahaan, dan berhak hadir dalam Rapat Umum Pemegang Saham (RUPS) (Juarna et al., 2025). Harga saham yang ditawarkan ditentukan oleh perusahaan tersebut untuk para investor yang ingin memiliki hak kepemilikan saham (Syah et al., 2025). Proses jual beli saham dilakukan melalui pasar modal, tempat dimana emiten dan investor bertemu untuk mendapatkan modal (Kwanda et al., 2024).

1.4.2 Tesla

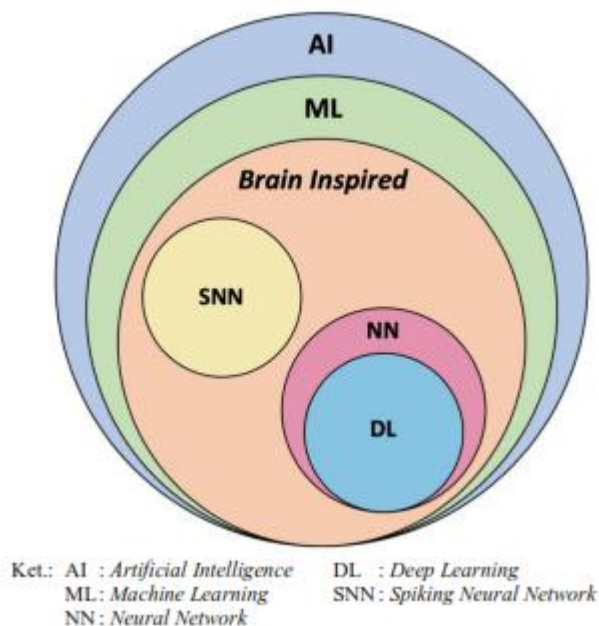
Tesla, Inc. adalah perusahaan otomotif dan energi yang didirikan pada tahun 2003 dengan visi mempercepat transisi dunia ke energi berkelanjutan. Tesla dikenal sebagai pelopor kendaraan listrik (*electric vehicles* atau EV) dan solusi energi ramah lingkungan, seperti Powerwall dan panel surya. Di bawah kepemimpinan Elon Musk, Tesla mendorong inovasi dalam baterai, pengisian daya, dan kecerdasan buatan, serta menerapkan strategi distribusi langsung tanpa *dealer*. Perusahaan ini berperan penting dalam mendorong adopsi EV secara global dan meningkatkan persaingan di industri kendaraan ramah lingkungan (Saputra et al., 2024).

1.4.3 Machine Learning

Machine Learning (ML) merupakan cabang dari Kecerdasan Buatan (*Artificial Intelligence*). Jika *Artificial Intelligence* (AI) adalah tujuan akhir untuk menciptakan mesin yang cerdas dan mampu meniru perilaku manusia, maka *machine learning* adalah salah satu jalan menuju tujuan tersebut. Secara umum, *machine learning* dapat didefinisikan sebagai suatu bidang studi yang berfokus pada pelatihan

komputer agar mampu belajar dari data dan pengalaman masa lalu untuk mengenali pola, membuat prediksi, dan mengambil keputusan. Teknologi ini menjadi fondasi dari berbagai inovasi modern, seperti pengenalan wajah, kendaraan otonom, asisten virtual, dan sistem rekomendasi (Bintoro et al., 2024).

1.4.4 Deep Learning



Gambar 1. Taksonomi Artificial Intelligence (Sumber: Diponegoro et al., 2021)

Deep Learning adalah teknik pembelajaran mesin yang berbasis *artificial neural network* (jaringan saraf tiruan) dengan banyak lapisan tersembunyi (Nurhakiki & Yahfizham, 2024). *Deep Learning* disebut sebagai *deep* karena banyaknya *hidden layer* maupun *neuron* di dalamnya (Hariz et al., 2022). Setiap lapisan dalam *Deep Learning* menerima *output* dari lapisan sebelumnya sebagai *input*, memungkinkan terjadinya pembelajaran hierarkis yang kuat (Diponegoro et al., 2021).

1.4.5 Prediksi

Prediksi adalah upaya untuk meramalkan masa depan. Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Kwanda et al., 2024).

1.4.6 Runtun Waktu (*Time Series*)

Data time series merupakan serangkaian pengamatan yang terurut berdasarkan waktu dengan jarak yang sama. Jenis data ini sering ditemui dalam keseharian karena data tersebut dikumpulkan melalui waktu interval, yaitu harian, mingguan atau bulanan. Dari data yang terkumpul dapat dilihat suatu pola di dalamnya. Pola gerakan data atau nilai-nilai variabel dari data *time series* yang diketahui dapat dijadikan sebagai dasar untuk pembuatan keputusan saat ini, peramalan keadaan pada masa yang akan datang, dan perencanaan kegiatan untuk masa depan (Anbiya dan Garini, 2022).

1.4.7 Data Preprocessing

Pada tahap data preprocessing adalah tahapan dimana melibatkan persiapan dataset sebelum digunakan untuk membangun model. Proses ini mencakup pembagian data menjadi *train set* dan *test set*, lalu kemudian normalisasi data menggunakan skala fitur untuk memastikan bahwa semua variabel berada dalam rentang yang sama (Marwa et al., 2025). Tahap *preprocessing* dilakukan untuk menghasilkan data berkualitas atau data yang bebas dari duplikasi data dan data tidak lengkap untuk menunjang pemodelan prediktif (Perwitasari, 2023).

1.4.9.1 Data Splitting

Data *splitting* adalah proses membagi dataset menjadi data pelatihan (*training*) dan data pengujian (*testing*) dengan proporsi tertentu (Putri et al., 2022). Pada data time series, pembagian dilakukan secara berurutan tanpa pengacakan. Data *training* digunakan untuk membangun model, sedangkan data *testing* digunakan untuk mengevaluasi performanya (Janastu & Wutsqa, 2024). Pembagian ini penting agar evaluasi model menjadi konsisten dan hasil prediksi dapat diandalkan (Putri et al., 2022).

1.4.9.2 Data Normalization

Normalisasi dilakukan untuk menskalakan nilai data dalam rentang nilai 0 sampai 1, untuk mempercepat konvergensi model selama proses pelatihan (Perwitasari, 2023; Marwa et al., 2025). Salah satu metode normalisasi adalah *min-max normalization* atau *min-max scaler*, yang mengubah data menjadi rentang nilai [0, 1]. Rumus *min-max scaler* (Wahyudi et al., 2024) ditunjukkan pada persamaan (1).

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Dimana:

x_i' = Hasil normalisasi
 x_i = Data ke- i

x_{max} = Data dengan nilai terkecil
 x_{min} = Data dengan nilai terbesar

1.4.8 Transformer

Transformer adalah model yang pertama kali dikenalkan oleh A. Vaswani dan rekan-rekannya melalui makalah terkenal "*Attention is All You Need*". Model ini membawa dampak besar dalam bidang *Natural Language Processing* (NLP), karena berhasil menggantikan pendekatan lama seperti *Recurrent Neural Network* (RNN) (Cebi & Bahsoun, 2024). Tidak seperti RNN atau *Convolutional Neural Network* (CNN) yang memproses data secara berurutan, arsitektur Transformer menggunakan mekanisme *self-attention* untuk memahami hubungan antar elemen dalam data urutan secara lebih cepat dan efisien (Yan & Fan, 2025).

Transformer merupakan model *neural network* berbasis *attention* yang sangat efektif dan telah banyak digunakan dalam berbagai tugas NLP sejak pertama kali diperkenalkan pada tahun 2017. Model ini sangat populer karena tingkat akurasi yang tinggi dalam pemodelan data berurutan (*sequence data*). Hal ini telah memungkinkan model Transformer untuk digunakan dalam permasalahan analisis deret waktu, seperti prediksi harga saham. Deret waktu sendiri merupakan urutan data numerik yang tercatat secara berurutan dari waktu ke waktu (Muhammad et al., 2022).

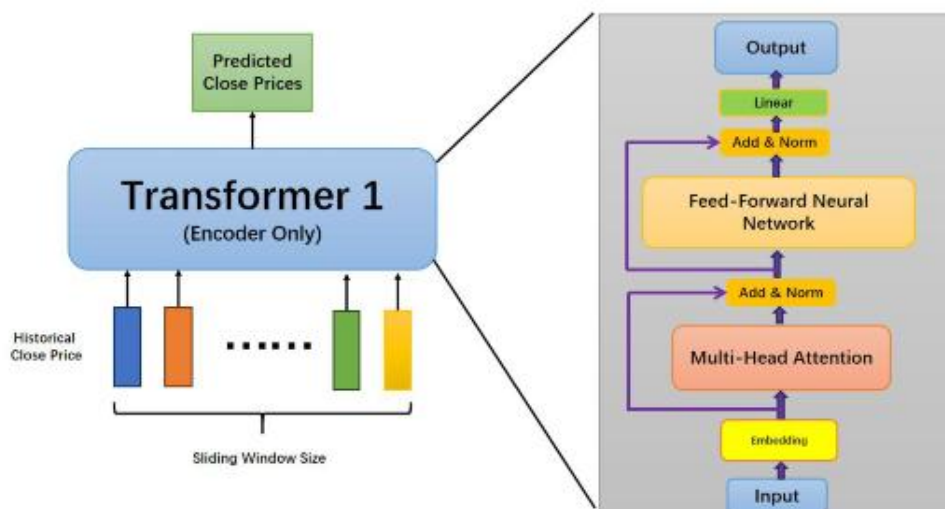
NLP sendiri merupakan salah satu subbidang dari kecerdasan buatan yang memungkinkan komputer untuk memahami dan menghasilkan bahasa manusia. NLP terdiri atas dua komponen utama, yaitu *Natural Language Understanding* (NLU) dan *Natural Language Generation* (NLG). Komponen pertama, NLU, mencakup seluruh konsep yang terlibat dalam proses pemahaman bahasa alami oleh komputer. NLU memungkinkan sistem untuk memahami informasi kontekstual dari berbagai bentuk data dan bahasa. Sementara itu, NLG, merujuk pada proses menghasilkan data berupa frasa, kalimat, hingga paragraf berdasarkan representasi internal (Tucudean et al., 2024).

Language model (LM) adalah salah satu bagian dari NLP. *Language model* telah hadir sebagai metode untuk memodelkan sifat-sifat bahasa alami secara statistik. Dengan menggunakan kumpulan teks sebagai masukan, LM menghitung properti statistik dari bahasa tersebut, seperti frekuensi dan probabilitas kata serta konteks sekitarnya, yang kemudian dapat digunakan untuk berbagai tujuan, termasuk NLU, NLG, dan secara lebih umum dalam NLP. Fitur *autogenerate* di *Google Search* adalah salah satu contoh LM. (Zubiaga, 2024).

Model Transformer memiliki beberapa komponen utama, yaitu mekanisme *self-attention*, jaringan saraf *feed-forward*, *residual connection*, dan normalisasi lapisan (Yan & Fan, 2025). Di antara semua komponen tersebut, bagian paling penting dari Transformer adalah mekanisme *attention*. Selain itu, karena proses ini dapat

dilakukan secara paralel, Transformer menjadi sangat efisien saat dilatih pada dataset yang besar (Cebi & Bahsoun, 2024). Penjelasan lebih rinci mengenai desain setiap bagian dari model ini akan dijabarkan pada bagian selanjutnya.

1.4.9 Transformer (*Encoder-only*)



Gambar 2. Struktur Transformer 1 (Sumber: Chen, 2025)

Transformer biasanya digunakan dalam arsitektur *encoder-decoder* penuh untuk tugas *sequence-to-sequence*. Namun, dalam konteks prediksi harga saham, beberapa peneliti memilih untuk hanya menggunakan bagian *encoder* dari Transformer, karena tujuannya biasanya untuk meramalkan nilai berikutnya dalam suatu urutan daripada menghasilkan urutan yang sepenuhnya baru. Sebagai contoh, Brugiére dan Turinici menggunakan Transformer *encoder-only* untuk meramalkan indeks saham S&P 500.

Seperti yang ditunjukkan pada Gambar 2, model ini hanya menggunakan komponen *encoder* dari arsitektur Transformer. Model ini memproses urutan *input*, seperti data historis harga saham, dan menghasilkan representasi urutan yang digunakan untuk memprediksi harga saham di masa mendatang (Chen, 2025).

1.4.9.1 Encoder

Encoder merupakan bagian utama dari model Transformer yang berfungsi untuk mengambil informasi penting dari data *input* dan memahami hubungan antar data dalam urutan waktu. Secara umum, *encoder* terdiri dari beberapa lapisan (*Encoder Layers*) yang disusun secara bertumpuk. Di dalam setiap lapisan tersebut terdapat komponen utama bernama *self-attention* yang berperan menangkap pola hubungan antar elemen dalam data (Yan & Fan, 2025).

1.4.9.2 *Embedding*

Embedding data merujuk pada proses transformasi *input* mentah menjadi representasi vektor berdimensi tinggi yang dapat dipelajari oleh model. Menghapus lapisan *embedding* data dari banyak model peramalan terkini tidak menurunkan kinerja peramalan, dalam banyak kasus, hal ini justru meningkatkan akurasi peramalan dan efisiensi komputasi (Nematirad et al., 2025).

1.4.9.3 *Attention*

Encoder merupakan bagian utama dari model Transformer yang berfungsi untuk mengambil informasi penting dari data *input* dan memahami hubungan antar data dalam urutan waktu. Secara umum, *encoder* terdiri dari beberapa lapisan (*Encoder Layers*) yang disusun secara bertumpuk. Di dalam setiap lapisan tersebut terdapat komponen utama bernama *self-attention* yang berperan menangkap pola hubungan antar elemen dalam data (Yan & Fan, 2025).

Mekanisme *self-attention* menghitung penjumlahan berbobot dari seluruh elemen dalam urutan *input*, dimana bobot ditentukan secara dinamis berdasarkan isi dari *input* itu sendiri. Ini memungkinkan model untuk memfokuskan perhatian pada bagian-bagian *input* yang paling relevan dengan konteks saat ini. Dalam arsitektur Transformer seperti yang diperkenalkan oleh Vaswani et al. (2017), *self-attention* diimplementasikan melalui mekanisme *scaled dot-product attention*, yang secara efisien menghitung bobot perhatian berdasarkan kesesuaian antara *query* dan *key*. Fungsi attention (Cebi & Bahsoun, 2024) didefinisikan sebagai:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

Dimana:

Q = Query

K = Key

V = Value

d_k = Dimensi dari key

1.4.9.4 *Multi-Head Attention*

Dalam mekanisme *multi-head attention*, beberapa skor *attention* dihitung secara paralel melalui proyeksi linear. Hasil dari setiap *head* ini kemudian digabungkan dan diproyeksikan kembali untuk menghasilkan *output* akhir. Pendekatan ini memungkinkan model untuk secara bersamaan menangkap berbagai karakteristik dari urutan *input* melalui representasi dari subruang yang berbeda.

Untuk lebih menyempurnakan kapasitas model dalam memperhatikan informasi dari subruang representasi yang berbeda pada posisi yang berbeda, Transformer memanfaatkan mekanisme *multi-head attention* (Anh et al., 2024).

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (3)$$

dimana

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (4)$$

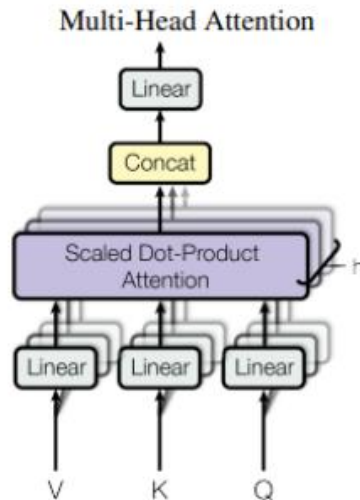
Dimana:

W_i^Q, W_i^K, W_i^V = Matriks parameter untuk *head* ke- i

W^O = Matriks parameter *output*

h = Jumlah *head*

Dalam ekspresi ini, matriks parameter W^O, W_i^Q, W_i^K, W_i^V adalah elemen yang mentransformasikan *input* ke dalam mekanisme *attention*. Skor *attention* dihitung dalam setiap *head* melalui *scaled dot-product attention*. Dengan kata lain, setiap *head*, i , dalam mekanisme *multi-head attention* menerapkan fungsi *attention* pada proyeksi *query*, *key*, dan *value*, memungkinkan model untuk menangkap rentang informasi yang lebih luas dari urutan *input* (Cebi & Bahsoun, 2024).



Gambar 3. *Multi-Head Attention* terdiri dari beberapa lapisan perhatian (*attention*) yang berjalan secara paralel (Sumber: Cebi & Bahsoun, 2024)

1.4.9.5 Add & Norm Layer

Sesuai dengan nama lapisannya, lapisan ini melakukan dua operasi. Langkah pertama adalah bagian "Add" yang menjalankan kontrol aliran melalui *Residual Connections*. Langkah kedua adalah "Norm" yang melakukan *Layer Normalization* (Nanal, 2022).

1.4.9.5.1 Residual Connections

Sesuai dengan nama lapisannya, lapisan ini melakukan dua operasi. Langkah pertama adalah bagian "Add" yang menjalankan kontrol aliran melalui *Residual Connections*. Langkah kedua adalah "Norm" yang melakukan *Layer Normalization* (Nanal, 2022).

Menambahkan input ke output dari lapisan menengah untuk mengurangi hilangnya gradien dan membantu pelatihan model yang lebih dalam (Sarkar & Vadivu, 2025):

$$\text{Residual Output} = \text{Layer Output} + \text{Input} \quad (5)$$

1.4.9.5.2 Layer Normalization

Layer Normalization memiliki fungsi untuk menstabilkan pelatihan dengan menormalkan keluaran dari lapisan-lapisan menengah, meningkatkan konvergensi dan mengurangi sensitivitas terhadap inisialisasi.

Proses ini secara matematis dijelaskan sebagai berikut (Anh et al., 2024):

$$\text{LayerNorm}(x) = \frac{x - E[x]}{\sqrt{\text{Var}[x] + \epsilon}} \odot \gamma + \beta \quad (6)$$

Dimana:

x = Vektor *input* untuk *layer normalization*

$E[x]$ = Rata-rata yang dihitung atas dimensi fitur

$\text{Var}[x]$ = Varians yang dihitung atas dimensi fitur

ϵ = Konstanta kecil yang ditambahkan untuk stabilitas numerik, biasanya pada orde 1e-5 hingga 1e-6

γ = Parameter skala yang dipelajari selama pelatihan

β = Parameter pergerakan yang dipelajari selama pelatihan

1.4.9.6 Feed-Forward Network

Setelah mekanisme *attention*, setiap lapisan dalam Transformer mengandung jaringan *feed-forward position-wise*, yang menerapkan transformasi linear pada setiap posisi secara terpisah dan identik (Anh et al., 2024):

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2 \quad (7)$$

Dimana:

x = Input ke lapisan *feed-forward*

W_1, b_1 = Bobot dan bias untuk transformasi linear pertama

W_2, b_2 = Bobot dan bias untuk transformasi linear kedua

1.4.10 Hyperparameter Tuning

Hyperparameter adalah parameter yang nilainya ditentukan sebelum pelatihan. Penyetelan *hyperparameter* (*hyperparameter tuning*) adalah proses untuk memilih kombinasi *hyperparameter* terbaik agar model memberikan hasil optimal, mencegah *underfitting* maupun *overfitting*, dan mempercepat konvergensi (Hariz et al., 2022). Dalam penelitian kali ini *hyperparameter* yang digunakan adalah *Optuna*.

1.4.10.1 Optuna

Optuna adalah *framework* optimisasi *hyperparameter* berbasis *Bayesian Optimization* yang mengeksplorasi ruang pencarian secara adaptif dan efisien. Berbeda dari *GridSearch* yang mengevaluasi seluruh kombinasi secara *brute-force*, *Optuna* menggunakan hasil percobaan sebelumnya untuk memandu pencarian menuju parameter optimal dengan biaya komputasi yang lebih rendah (Riski et al., 2025).

1.4.11 Data Denormalization

Denormalisasi dilakukan untuk mengembalikan menjadi data aktual dari data yang telah dinormalisasi sebelumnya. Nilai prediksi yang di dapat dari perhitungan dilakukan proses denormalisasi terlebih dahulu sebelum menghitung akurasi hasil prediksi. Perhitungan proses denormalisasi (Yunizar et al., 2023) dapat dilihat pada persamaan (8):

$$x_i = x_i'(x_{max} - x_{min}) + x_{min} \quad (8)$$

Dimana:

x_i = Data ke- i

x_i' = Hasil normalisasi

x_{max} = Data dengan nilai terbesar

x_{min} = Data dengan nilai terkecil

1.4.12 Evaluasi

Metriks evaluasi digunakan untuk mengukur performa dari prediksi model. Pada model *time series*, evaluasi dapat dilakukan dengan menghitung error yang dibuat pada model (Meriani & Rahmatulloh, 2024). Metrik evaluasi yang digunakan dalam evaluasi model penelitian kali ini adalah *Root Mean Square Error* (RMSE) dan *Mean Percentage Absolute Error* (MAPE).

1.4.12.1 Root Mean Square Error (RMSE)

Ketika model *machine learning* diterapkan untuk prediksi saham, *Root Mean Square Error* (RMSE) umumnya digunakan untuk mengukur perbedaan antara hasil prediksi model dan nilai aktual yang diamati (Li, 2024). Berikut adalah interpretasi matematis (Karnisih et al., 2025). dari RMSE, yang merupakan ukuran kesalahan yang umum digunakan.

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

Dimana:

$\hat{y}_i$ = Nilai prediksi pada data ke- i

y_i = Nilai aktual pada data ke- i

n = Jumlah data

Alasan penggunaan RMSE sebagai ukuran untuk membandingkan kinerja berbagai model dalam prediksi harga saham adalah karena RMSE mengukur kesalahan prediksi secara kuantitatif, menyoroti perbedaan antara nilai prediksi dan nilai aktual secara objektif. RMSE menekankan error besar, memberikan wawasan tentang prediksi nilai ekstrem. Konsistensi unitnya membantu pemahaman cepat oleh praktisi dan peneliti, memfasilitasi pengambilan keputusan dan pemilihan model (Li, 2024).

1.4.12.2 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) merupakan nilai absolut dari persentase error data terhadap nilai rata-rata. MAPE adalah perhitungan yang digunakan untuk mengukur akurasi dari sistem prediksi. Perhitungan MAPE dapat dilihat pada persamaan 10.

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

Dimana:

$\hat{y}_i$ = Nilai prediksi pada data ke- i

y_i = Nilai aktual pada data ke- i

n = Jumlah data

Nilai MAPE dikatakan baik dalam suatu prediksi apabila memiliki hasil yang rendah atau persentasenya mendekati nol. Semakin kecil nilai MAPE maka semakin akurat sebuah model dalam melakukan prediksi (Yunizar et al., 2023).

BAB II

METODOLOGI PENELITIAN

2.1 Pendekatan dan Jenis Penelitian

Pendekatan yang digunakan dalam penelitian ini adalah kuantitatif deskriptif dengan metode penelitian eksperimental bersifat prediktif. Pendekatan kuantitatif deskriptif dipilih karena penelitian ini berfokus pada pengolahan data numerik berupa harga penutupan harian saham, yang dianalisis menggunakan algoritma *machine learning* untuk menggambarkan pola serta karakteristik data secara matematis. Penelitian ini juga bersifat eksperimental prediktif, karena melibatkan pengujian model prediksi Transformer, untuk memperkirakan harga saham masa depan berdasarkan data historis.

Data yang digunakan dalam penelitian ini adalah data kuantitatif berupa data sekunder harga penutupan saham harian (*close*) selama hampir sepuluh tahun terakhir, yaitu dari Juni 2010 hingga Februari 2020. Proses pengolahan dan analisis data dilakukan menggunakan perangkat lunak Google Colaboratory untuk mendukung pemodelan, analisis, serta prediksi harga saham berbasis *machine learning* (Transformer). Kombinasi metode dan alat ini diharapkan dapat memberikan hasil yang komprehensif dalam membandingkan pendekatan tersebut.

Metode pengumpulan data dilakukan melalui studi literatur dan pengunduhan data harga penutupan saham harian dari situs <https://www.kaggle.com/datasets/timoboz/tesla-stock-data-from-2010-to-2020>. Selain itu, analisis dan pengembangan model didukung oleh berbagai referensi, termasuk buku, jurnal, skripsi, serta artikel yang membahas model Transformer dan prediksi harga saham dengan pendekatan *machine learning*.

Penelitian ini membandingkan performa model Transformer dengan model-model *machine learning* lainnya yang telah dikaji dalam artikel *Comparison of Machine Learning Models for Stock Prediction* karya Siyuan Li (2024), seperti *Linear Regression*, *K-Nearest Neighbor* (KNN), dan *Long Short-Term Memory* (LSTM), yang dipublikasikan dalam *Proceedings of the 2nd International Conference on Management Research and Economic Development*.

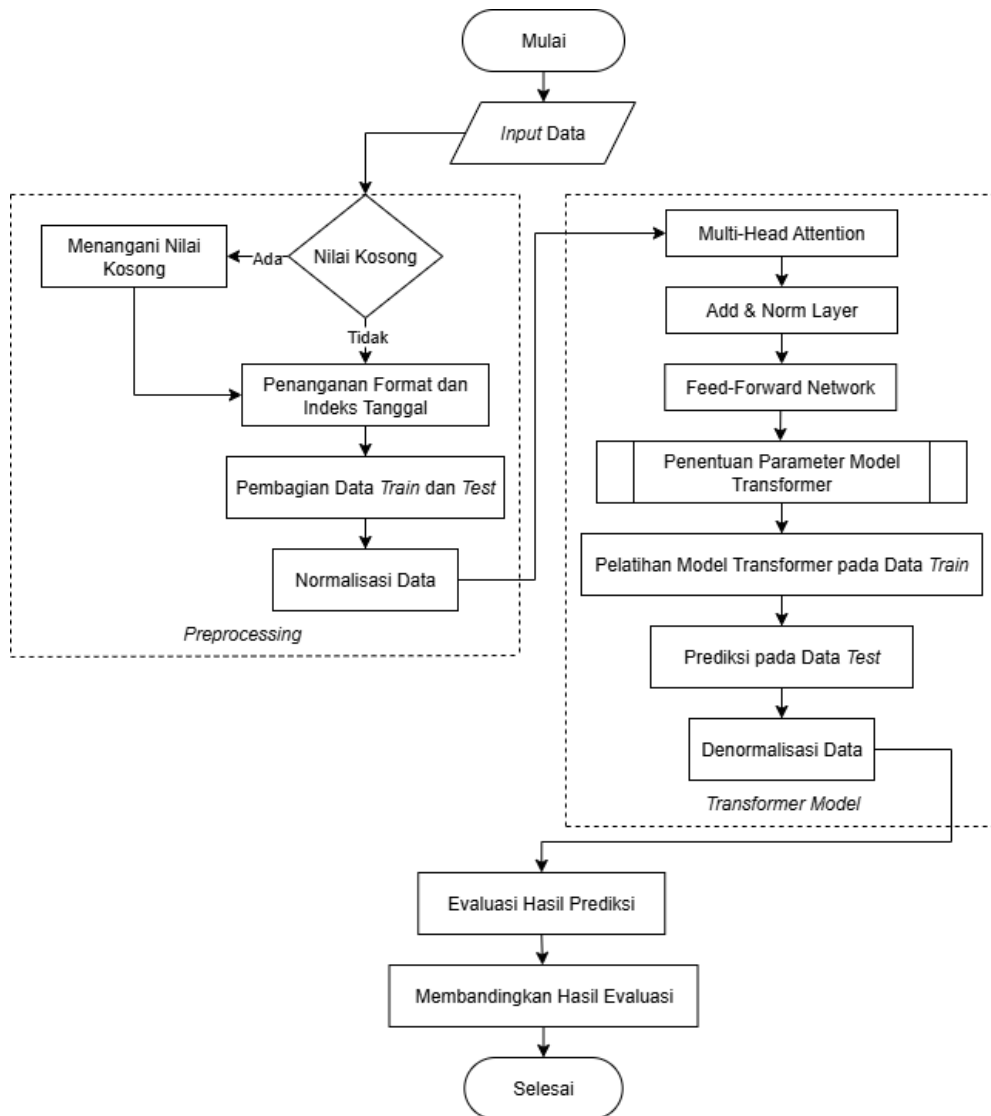
2.2 Metode Machine Learning (Transformer)

Penelitian ini dilaksanakan melalui serangkaian tahapan yang disusun secara sistematis guna memperoleh hasil prediksi harga saham secara ilmiah. Adapun alur penelitian yang ditempuh dalam studi ini dapat dijabarkan sebagai berikut:

1. Melakukan pengumpulan data dari situs Kaggle.
2. Membaca dataset saham, fokus penelitian ini adalah pada kolom "*Close*".
3. Memeriksa *missing value*.
4. Memeriksa tipe data dan menyesuaikannya dengan tipe yang dibutuhkan.
5. Melakukan pembagian data *train* dan data *test*.

6. Melakukan normalisasi data.
7. Menentukan *timesteps* pada data.
8. Menentukan parameter untuk *hyperparameter tuning*.
9. Membangun model Transformer.
10. Melatih model Transfomer pada data *train*.
11. Melakukan prediksi data *test* dengan model Transformer.
12. Mengevaluasi model Transformer dengan metrik RMSE dan MAPE.
13. Menyimpan hasil prediksi harga saham.
14. Melakukan perbandingan hasil evaluasi model Transformer dengan hasil evaluasi model-model *machine learning* lain yang dikaji dalam artikel *Comparison of Machine Learning Models for Stock Prediction* karya Siyuan Li (2024), yang dipublikasikan dalam *Proceedings of the 2nd International Conference on Management Research and Economic Development*.

2.3 Alur kerja



Gambar 4. Alur Kerja Pendekatan *Machine Learning* Transformer