

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Investasi

Investasi menjadi salah satu faktor penting yang memengaruhi laju pertumbuhan ekonomi suatu negara. Secara umum, investasi diartikan sebagai proses mobilisasi sumber daya untuk menciptakan atau meningkatkan kapasitas produksi maupun pendapatan di masa mendatang (Millia *et al.* 2024). Tujuan utama investasi adalah menggantikan bagian penyediaan modal yang kurang produktif serta menambah ketersediaan modal yang telah ada. Investasi juga dipahami sebagai kegiatan penanaman modal, baik dalam bentuk uang maupun aset lainnya, dengan tujuan memperoleh keuntungan dalam suatu periode tertentu. Namun, investasi memiliki unsur ketidakpastian (*uncertainty*) atau risiko, sehingga investor tidak dapat memastikan secara tepat besarnya keuntungan maupun potensi kerugian yang mungkin timbul dari investasi tersebut (Paningrum, 2022). Keputusan dalam berinvestasi tidak terlepas dari pengaruh berbagai faktor ekonomi, baik berskala makro maupun mikro. Beberapa faktor utama yang memengaruhi keputusan investasi meliputi inflasi, suku bunga, nilai tukar, stabilitas ekonomi, dan kondisi pasar. Faktor-faktor ini saling berinteraksi dalam memengaruhi preferensi masyarakat terhadap berbagai instrumen investasi, termasuk emas (Siagian, 2025).

2.1.2 Harga Emas Dunia

Menurut Philip Kotler harga adalah suatu unsur bauran pemasaran yang berfungsi sebagai elemen yang menghasilkan pendapatan bagi perusahaan. Dalam konteks komoditas, salah satu barang yang memiliki nilai ekonomi tinggi dan diperdagangkan secara luas adalah emas. Emas merupakan logam padat yang bersifat lembut, berkilau, dan dikenal sebagai logam paling lentur dibandingkan dengan logam lainnya. Emas memiliki beberapa kelebihan, yaitu tidak mudah berubah warna, tidak berkarat, tidak memudar meskipun disimpan dalam jangka

waktu yang lama, dan memiliki daya tarik tinggi sebagai simbol kemewahan dan kekayaan.

Menurut Mohammad (2014), emas merupakan salah satu komoditas penting di dunia yang dapat berfungsi sebagai alat tukar atau alat pembayaran. Dalam pasar global, harga emas dunia berperan sebagai acuan internasional dalam perdagangan logam mulia. Harga ini biasanya dinyatakan dalam satuan USD per *troy ounce* (USD/oz) dan ditetapkan melalui aktivitas perdagangan di pasar komoditas internasional seperti *London Bullion Market Association* (LBMA) dan *Commodity Exchange* (COMEX). Standar yang digunakan sebagai patokan untuk harga emas di seluruh dunia adalah harga standar emas pasar London atau disebut sebagai *London Gold Fixing* (LGF). Standar ini mengatur sistem penentuan harga emas yang dilakukan dua kali pada setiap hari kerja di pasar London (Candra et al. 2014). Waktu yang ditetapkan untuk menentukan harga emas tersebut adalah pada pukul 10.30 GMT (*London Gold AM Fix*) dan pukul 15.00 GMT (*London Gold PM Fix*). Harga emas ini ditentukan oleh Presiden LGF yang kemudian diteruskan kepada lima anggota pasar LGF, yaitu *Bank of Nova Scotia*, *Barclaya Capital*, *Deutsche Bank*, *Hongkong and Shanghai Banking Corporation (HSBC)*, dan *Societe Generale*. Kemudian kelima anggota ini akan berhubungan dengan *dealer* secara langsung dengan pembeli untuk menetapkan harga dari emas yang mengacu pada permintaan dan penawaran dari pasar.

Keuntungan investasi emas:

1. Emas merupakan simbol kebanggaan, karena memiliki prestise tersendiri saat digunakan sebagai aksesoris (salah satu tujuan investasi emas adalah untuk perhiasan atau aksesoris).
2. Harga emas cenderung mengalami fluktuasi karena permintaannya selalu lebih besar daripada jumlah yang tersedia.
3. Emas dapat digunakan untuk mempertahankan kekayaan karena ketika disimpan nilai tukarnya akan tetap sama, bahkan cenderung naik bila diperlukan dengan mata uang lain.
4. Emas tersedia dalam bentuk yang beragam (perhiasan berbagai bentuk, batang ataupun koin).

5. Emas dapat dijadikan jaminan saat darurat karena diterima oleh banyak lembaga keuangan seperti pegadaian, bank syariah, maupun koperasi simpan pinjam.

2.1.3 Prediksi

Prediksi (*forecasting*) adalah suatu upaya perkiraan yang dilakukan untuk mendapatkan sebuah dugaan peristiwa yang akan terjadi di masa depan. Prediksi merupakan hal yang penting bagi setiap bidang dan organisasi bisnis untuk setiap pengambilan keputusan manajemen yang sangat signifikan. Prediksi dapat mempengaruhi pengambilan keputusan dalam perencanaan jangka panjang. Prediksi selalu melibatkan data masa lalu dan menampilkannya ke masa depan dengan menggunakan model matematika (Nasution, 2019). Oleh karena itu, metode prediksi ini mengasumsikan bahwa kondisi yang dihasilkan data masa lalu tidak berbeda dengan kondisi masa depan, kecuali variabel-variabel yang secara eksplisit digunakan pada periode tersebut. Prediksi bagi manajemen merupakan suatu sistem yang sistematis. Dengan kata lain, prediksi tidak boleh dilihat sebagai sesuatu yang permanen atau statis (Yulian *et al.* 2020).

Prediksi diperlukan dalam kehidupan bermasyarakat karena segala sesuatu yang tidak pasti sukar diperkirakan dengan tepat dan akurat. Tujuan dari prediksi adalah untuk mendapatkan prediksi yang memiliki nilai terkecil pada kesalahan prediksi (*forecast error*), yang dapat diukur dengan *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE), *Root Mean Squared Error* (RMSE) dan sebagainya (Layakana *et al.* 2020). Saat membuat prediksi, penting untuk diingat bahwa pada dasarnya tidak ada yang tahu apa yang akan terjadi di masa depan. Wajar jika setiap prediksi pasti memiliki eror atau kesalahan. Namun, harus diusahakan agar kesalahan dalam prediksi itu sekecil mungkin. Penelitian ini menggunakan data *time series* atau biasa disebut juga data deret waktu.

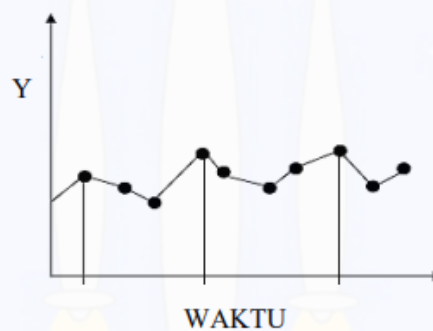
2.1.4 Deret Waktu (*Time Series*)

Menurut Cryer (1986), *time series* adalah serangkaian data pengamatan yang disusun menurut urutan waktu, di mana pengamatan tersebut bersifat acak dan

saling berhubungan secara statistika. Menurut Box dan Jenkins (1976) *time series* adalah sekelompok nilai-nilai pengamatan yang diperoleh pada titik waktu yang berbeda dengan selang waktu yang sama dan barisan data diasumsikan saling berhubungan satu sama lain. Dengan demikian data *time series* dapat didefinisikan sebagai sekumpulan data hasil observasi secara teratur dari waktu ke waktu (Indah *et al.* 2018.). Menurut (Hamirsa *et al.* 2022) data *time series* memiliki beberapa jenis pola, yang masing-masing digambarkan secara ilustratif pada Gambar 2.1.

1. Pola Horizontal

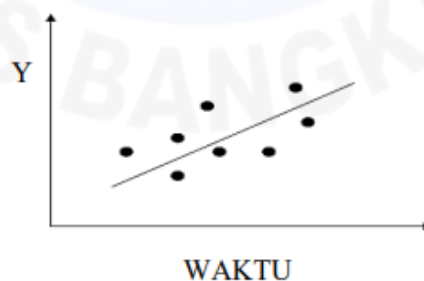
Pola horizontal terjadi apabila nilai data yang berfluktuasi di sekitar nilai rata-rata yang relatif konstan atau stasioner. Adapun Gambar 2.1 menggambarkan pola data horizontal.



Gambar 2. 1 Pola Horizontal

2. Pola Tren (*Trend*)

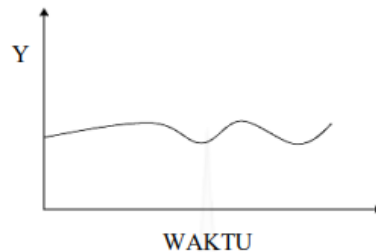
Pola tren terjadi apabila terdapat kenaikan atau penurunan jangka panjang pada data deret waktu. Adapun Gambar 2.2 menggambarkan pola data tren.



Gambar 2. 2 Pola Data Tren

3. Pola Siklis (*Cyclical*)

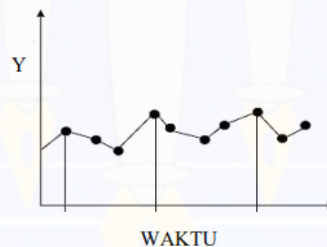
Pola siklis terjadi apabila datanya dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis. Adapun Gambar 2.3 menggambarkan pola data siklis.



Gambar 2. 3 Pola Data Siklis

4. Pola Musiman (*Seasonal*)

Pola musiman terjadi ketika suatu deret waktu dipengaruhi oleh faktor musiman, misalnya kuartal tertentu dalam satu tahun, periode bulanan, atau hari-hari tertentu dalam minggu. Adapun Gambar 2.4 menggambarkan pola data musiman.



Gambar 2. 4 Pola Data Musiman

2.1.5 *Machine Learning*

Machine Learning atau Pembelajaran Mesin adalah teknik pendekatan dari *Artificial Intelligent* (AI) yang digunakan untuk meniru untuk menggantikan peran manusia dalam melakukan aktivitas untuk memecahkan masalah. Menurut Arthur Samuel, seorang perintis Amerika di bidang permainan komputer dan kecerdasan buatan, *machine learning* merupakan cabang ilmu yang mempelajari bagaimana memberi komputer kemampuan untuk belajar tanpa diprogram secara eksplisit. Pembelajaran Mesin dapat melakukan hal tersebut jika didasarkan pada ide-ide yang diperoleh dari data sebelumnya, mengidentifikasi pola dan membuat keputusan dengan sedikit intervensi manusia atau penggunaannya. Secara garis besar,

algoritma *machine learning* dibagi menjadi tiga jenis, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. *Supervised learning* memanfaatkan data berlabel untuk melatih model sehingga mampu memprediksi *output* berdasarkan *input* tertentu, biasanya digunakan dalam masalah klasifikasi dan regresi. Sementara itu, *unsupervised learning* bekerja tanpa data berlabel untuk menemukan pola atau struktur tersembunyi dalam data, seperti pada teknik *clustering*. Adapun *reinforcement learning* merupakan metode yang melibatkan agen yang belajar melalui interaksi dengan lingkungan dengan memperoleh *feedback* berupa *reward* dan *punishment*, sehingga secara bertahap dapat menemukan strategi yang memaksimalkan hasil (Wijoyo *et al.* 2024).

2.1.6 Deep Learning

Deep learning merupakan salah satu bidang dari *machine learning* yang memanfaatkan banyak layer pengolahan informasi non linier untuk melakukan ekstraksi fitur, pengenalan pola, dan klasifikasi (Ahmad *et al.* 2019). Menurut (Rochmawati *et al.* 2021). *Deep learning* adalah bagian dari *machine learning* yang berkaitan dengan algoritma di mana cara kerjanya meniru struktur dan fungsi otak yang disebut jaringan saraf tiruan. Sedangkan (Goodfellow *et al.* 2016) menyebutkan bahwa *deep learning* adalah pendekatan dalam penyelesaian masalah pada sistem pembelajaran komputer menggunakan konsep yang kompleks dengan cara menggabungkan konsep-konsep yang sederhana.

Algoritma *deep learning* merupakan jaringan neural dengan cara kerja meniru otak manusia yang di mana memiliki banyak lapisan neuron buatan yang saling terhubung yang bekerja sama untuk mempelajari dan memproses informasi di dalam komputer. *Neuron* buatan adalah modul perangkat lunak yang berguna untuk memproses data menggunakan perhitungan matematis. Teknik *deep learning* memberikan tampilan kerja yang sangat baik dengan menambahkan lebih banyak lapisan neuron memungkinkan model merepresentasikan data gambar berlabel dengan lebih baik. *Deep learning* mencakup teknik untuk mengekstraksi fitur dari data pelatihan dan algoritma khusus untuk mengklasifikasikan gambar dan mendeteksi audio. (Nurhakiki *et al.* 2024). Pada penelitian ini menggunakan pendekatan metode *deep learning* dengan metode EEMD-LSTM.

2.1.7 *Preprocessing*

Preprocessing data pada metode *deep learning* adalah tahapan awal dalam proses analisis data yang bertujuan untuk mempersiapkan data mentah agar layak digunakan dalam proses pemodelan. Proses *preprocessing* ini dilakukan dengan tujuan untuk meningkatkan kualitas data serta meminimalkan risiko terjadinya *overfitting* atau *underfitting* yang dapat disebabkan oleh data yang terlalu banyak atau terlalu sedikit sehingga model tidak mampu menangkap pola yang terdapat pada data dengan baik. pada model (Muhammad *et al.* 2025). Adapun tahapan *preprocessing* yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. *Data Cleaning*

Proses pembersihan data, atau yang dikenal sebagai *Data Cleaning*, merupakan serangkaian langkah untuk membersihkan data, seperti mengidentifikasi dan mengisi nilai yang hilang (*missing value*) pada data, menghapus data yang tidak konsisten (*noisy/outlier*), mencari dan menghapus data yang duplikat, serta menangani data yang tidak lengkap (Gori *et al.* 2024).

2. Pembagian data

Pembagian data menjadi data *training* dan data *testing* dengan tujuan untuk melatih dan menguji model (Cahyani *et al.* 2023). Salah satu pendekatan yang digunakan pada pembagian data *training* dan data *testing* adalah metode “*Pareto Principle*” yang dikenal sebagai hukum 80:20 (Islam, 2006.). Sehingga jumlah data *training* atau data *testing* yang dipakai pada penelitian ini adalah 80% dari jumlah data yang ada dan sisanya digunakan untuk data *testing* 20%. Pada proses *training* dilakukan dengan melatih setiap model menggunakan data *training*. Sedangkan pada proses *testing* dilakukan menggunakan data *testing* yang kemudian dilakukan pengujian dari hasil model *training* sebelumnya untuk mengetahui efektivitas model yang telah dibuat.

3. Normalisasi Data

Normalisasi data bertujuan untuk proses mentransformasi sehingga menghindari situasi di mana nilai dengan rentang besar terhadap nilai dengan rentang kecil. Normalisasi dapat dilakukan menggunakan metode seperti *min-max*

scaler dengan batas rangenya dari 0 sampai dengan 1. Tahapan normalisasi data menggunakan metode *min-max scaler* yang merupakan metode normalisasi di mana transformasi linier dari data asli dilakukan untuk menciptakan keseimbangan nilai perbandingan antara data awal dan akhir setelah proses transformasi dilakukan dengan persamaan sebagai berikut (Nasution, 2019).

$$x' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \dots\dots\dots [1]$$

2.1.8 Metode EMD (*Empirical Mode Decomposition*)

Empirical Mode Decomposition (EMD) adalah metode adaptif untuk menganalisis sinyal nyata dengan skala yang beragam, non linier dan non-stasioner. EMD adalah metode analisis data yang memecah sinyal menjadi sejumlah *Intrinsic Mode Function* (IMF) dan komponen sisa. Jika seluruh IMF dan komponen sisa tersebut dijumlahkan kembali, akan diperoleh data asli secara utuh tanpa kehilangan informasi maupun menimbulkan kesalahan (Siregar *et al.* 2021). Tujuan pertama dari algoritma ini adalah untuk menguraikan sinyal *univariat*. EMD menguraikan serangkaian waktu menjadi sejumlah IMF yang memenuhi kondisi:

1. Jumlah titik ekstrim (*ekstrem point*) dan jumlah sinyal yang melewati sumbu horizontal (nilai nol) atau disebut dengan *zero crossing* harus sama atau berbeda salah satunya.
2. Nilai rata-rata kurva halus (*envelope*) didefinisikan oleh *maksimum* dan *minimum* harus sama dengan nol pada setiap titik.

Proses dalam EMD dikenal sebagai *shifting process*, yaitu perhitungan yang dilakukan secara berulang hingga diperoleh nilai IMF. Adapun tahapan deteksi sinyal menggunakan EMD adalah sebagai berikut (Siregar *et al.* 2021):

1. Mengidentifikasi nilai *ekstrem point* dari data $z(t)$, termasuk *envelope* atas (bukit) dan *envelope* bawah (lembah). Lalu, menghubungkan *envelope* atas dan *envelope* bawah menggunakan *cubic spline* merupakan teknik interpolasi yang umum digunakan dalam proses *Empirical Mode Decomposition* (EMD). Teknik ini digunakan untuk membentuk *envelope* atas dan bawah dari sinyal

berdasarkan titik-titik ekstrem, dengan tujuan menghindari terjadinya *overshooting* dan *undershooting*, serta tetap mempertahankan kehalusan kurva.

Persamaan *cubic spline* (Anggreni *et al.* 2025) :

$$S_i(x) = ax^3 + bx^2 + cx + d \dots\dots\dots [2]$$

2. Menghitung data rata-rata antara *envelope* atas dan *envelope* bawah yang didefinisikan sebagai $b(t)$.

$$b(t) = \frac{E_{max} - E_{min}(t)}{2} \dots\dots\dots [3]$$

3. Mengekstraksi nilai rata-rata $b(t)$ dari sinyal untuk memperoleh mode osilasi $d(t)$. Nilai $d(t)$ ini ditentukan melalui persamaan (4):

$$d(t) = z(t) - b(t) \dots\dots\dots [4]$$

4. Apabila $d(t)$ telah memenuhi kriteria IMF, maka proses dihentikan sementara dan $d(t)$ ditetapkan sebagai IMF pertama, yang dilambangkan dengan $c_i(t)$. Selanjutnya, apabila $d(t)$ memenuhi kriteria IMF, maka proses dilanjutkan kembali ke langkah pertama dengan mengganti $z(t)$ dengan $d(t)$.

5. Memisahkan $c_i(t)$ dari data residu yang didefinisikan sebagai $r(t)$, dimana $r(t)$ ditentukan menggunakan persamaan (5):

$$r(t) = z(t) - c_i(t) \dots\dots\dots [5]$$

6. Selanjutnya, $z(t)$ diganti dengan $r(t)$ dan langkah 1 hingga 5 diulangi untuk memperoleh IMF kedua ($c_2(t)$). Untuk mendapatkan IMF ke- n , prosedur yang sama dilakukan sebanyak n iterasi. Proses dekomposisi dihentikan apabila $r(t)$ telah menjadi fungsi monoton.

Proses ekstraksi IMF dihentikan Ketika residu $r(t)$ tidak lagi menunjukkan adanya osilasi. Dengan demikian, apabila seluruh IMF $c_i(t)$ dijumlahkan, maka data yang asli Z_t dapat direkonstruksi kembali (Dwi Candra *et al.* 2018). Algoritma EMD menguraikan sinyal asli menjadi IMF dan residu dapat diperoleh melalui persamaan (6):

$$z(t) = \sum_{i=1}^n c_i(t) + r(t) \dots\dots\dots [6]$$

2.2.0 Metode EEMD (*Ensemble Empirical Mode Decomposition*)

EEMD merupakan salah satu metode analisis data yang dikembangkan dari *Empirical Mode Decomposition* (EMD). Kelemahan utama EMD adalah terjadinya *mode mixing*, yaitu ketika satu IMF memuat sinyal dengan skala berbeda atau sinyal serupa tersebar pada beberapa IMF, sehingga pola sinyal menjadi tidak stabil dan kurang konsisten saat dianalisis. Untuk mengatasi kelemahan *mode mixing* pada EMD, diajukan suatu uji yang bersifat tidak tetap dan subjektif, meskipun pengaruhnya tidak dapat diprediksi. Selanjutnya, Wu dan Huang (2009) mengusulkan konsep *Ensemble EMD* (EEMD) sebagai solusi untuk mengatasi kelemahan EMD yang tidak mampu menanggulangi terjadinya *mode mixing* (Siregar *et al.* 2021). Metode *ensemble EMD* yang diajukan merupakan sebuah metode analisis baru dengan bantuan *noise*, untuk menghilangkan fenomena *mode mixing* dan mendapatkan distribusi frekuensi sebenarnya dari sinyal asli.

Prinsip sederhana dari EEMD yaitu menambahkan *white noise* ke data, kemudian mendistribusikan secara merata ke seluruh ruang frekuensi, sinyal bit dari skala yang berbeda dapat dirancang secara otomatis ke skala referensi yang tepat yang ditentukan oleh *white noise*. Adanya *mode mixing* dapat teridentifikasi apabila suatu IMF memuat sinyal dengan perbedaan skala yang signifikan, atau ketika sinyal dengan skala yang relatif sama muncul pada beberapa IMF yang berbeda (Dwi Candra *et al.* 2018). Prosedur *ensemble EMD* dapat dilihat sebagai berikut (Siregar *et al.* 2021).

1. Menginisialisasi jumlah *ensemble* M serta *amplitudo* sinyal *white noise*, dengan $m=1$.
2. Menambahkan *white noise* $n_m(t)$ ke dalam data $z(t)$, sehingga diperoleh persamaan (7)

$$Z_m(t) = z(t) + n_m(t) \dots\dots\dots [7]$$

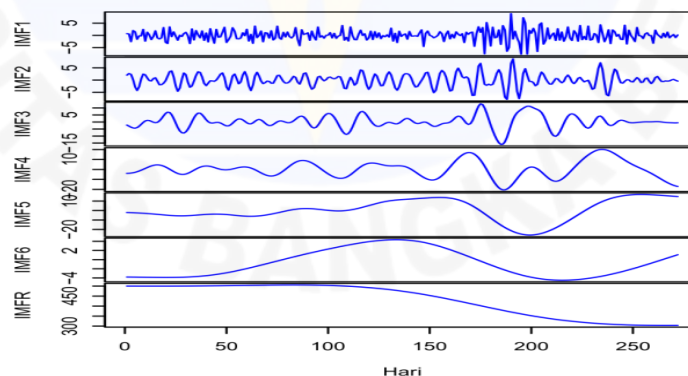
$n_m(t)$ merepresentasikan penambahan *white noise* ke- m , sedangkan $Z_m(t)$ menunjukkan hasil sinyal setelah diberikan *white noise* ke- m

3. Mendekomposisi $Z_m(t)$ menjadi I IMF_s $C_{i,m}$ ($i=1,2,3,\dots,I$) dengan algoritma EMD. $C_{i,m}$ menyatakan IMF ke- i dari m percobaan dan I menyatakan jumlah IMF_s.
4. Apabila $m < M$, kembali ke langkah (2) dengan mengganti $m=m+1$. Langkah 1 dan 2 diulan sebanyak M kali, dengan *white noise* yang berbeda pada setiap iterasi. Proses ini dihentikan ketika nilai m sama dengan M .
5. Menghitung rata-rata *ensemble* dari seluruh komponen IMF dan residu yang dihasilkan pada langkah sebelumnya, dengan menggunakan persamaan (8) dan (9) berikut:

$$\overline{c_i} = \frac{1}{M} \sum_{m=1}^M c_{i,m} \dots \dots \dots [8]$$

$$\overline{r_n} = \frac{1}{M} \sum_{m=1}^M r_{n,m} \dots \dots \dots [9]$$

Pada penelitian (Siregar *et al.* 2021), Zhang dinyatakan bahwa iterasi *ensemble* dapat dilakukan sebanyak 100 kali percobaan, dengan nilai simpangan baku yang dihasilkan berada rentang 0.1 dan 0.2. Hal ini dilakukan untuk melihat visualisasi data pada metode EEMD, sehingga pola sinyal yang tersembunyi dapat terungkap dengan lebih jelas melalui proses dekomposisi yang stabil dan menghasilkan visualisasi yang lebih akurat, yang dapat dilihat dari Gambar 2.5 berikut.



Gambar 2. 5 Contoh Visualisasi Data Hasil EEMD

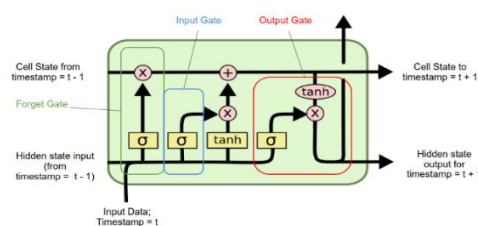
2.2.1 Long-Short Term Memory (LSTM)

Long-Short Term Memory (LSTM) merupakan varian dari *Recurrent Neural Network* (RNN) yang digunakan dalam bidang *Deep Learning* yang mampu untuk mengatasi keterbatasan memori jangka pendek (Hochreiter, 1997). Varian ini dinamakan dengan *Long-Short Term Memory* karena arsitektur LSTM Arsitektur dibangun dengan mekanisme yang mengubah memori jangka pendek menjadi memori jangka panjang. Pada konteks ini, memori jangka panjang merujuk pada bobot yang diperoleh selama proses pembelajaran, sedangkan memori jangka pendek direpresentasikan oleh *cell state* dalam *gates* yang terus diperbarui pada setiap langkah waktu t . (Lasijan *et al.* 2023).

LSTM dirancang untuk mampu menangkap ketergantungan jangka panjang pada data sekuensial, sesuatu yang sulit dicapai oleh RNN konvensional. Data sekuensial adalah data yang tersusun dalam urutan tertentu, di mana setiap elemen saling berkaitan dengan elemen sebelumnya dan sesudahnya, sehingga urutan data menjadi penting untuk dipelajari. Semakin panjang urutan data, semakin sulit bagi RNN konvensional dalam memperbarui gradien, sehingga sering terjadi masalah *vanishing gradient* (gradien mengecil hingga mendekati nol) atau *exploding gradient* (gradien membesar hingga tak terhingga) selama proses training. Keterbatasan ini dapat diatasi oleh arsitektur LSTM, yang dirancang khusus untuk menangani permasalahan tersebut. LSTM sangat efektif digunakan dalam klasifikasi, pemrosesan, maupun prediksi data runtun waktu, karena unit di dalamnya memungkinkan gradien mengalir secara stabil tanpa mudah menghilang saat dilakukan *backpropagation* (Goodfellow *et al.* 2018).

2.2.2 Struktur Jaringan Model LSTM

Struktur jaringan LSTM terdiri dari tiga proses *gates* dan satu proses pada *cell state* yang dapat dilihat dari Gambar 2.6 berikut:



Gambar 2. 6 Struktur Jaringan LSTM

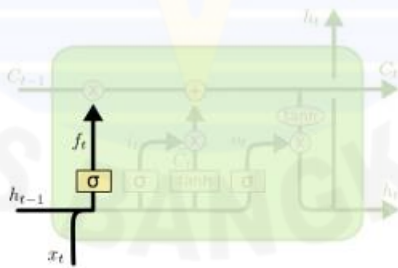
Komponen utama dalam metode LSTM adalah *Cell state* (C_t), yang digambarkan sebagai garis horizontal di bagian atas diagram sel dan berfungsi menghubungkan keluaran dari lapisan sebelumnya. Setiap *gates* dalam jaringan LSTM terdiri dari lapisan sigmoid serta fungsi tangen hiperbolik. Ketiga *gates* tersebut memiliki peran untuk menambahkan dan meneruskan informasi, maupun menghapus dan menghentikannya. Lapisan sigmoid berfungsi untuk mengecilkan hasil *output* ke dalam rentang $[0,1]$, jika nilai 0 maka informasi akan dihentikan, sedangkan jika bernilai 1 maka informasi akan diteruskan. Adapun persamaan sigmoid dapat dituliskan sebagai berikut:

$$\sigma(x) = \frac{1}{1+e^{-x}} \dots\dots\dots [10]$$

Sedangkan fungsi tangen hiperbolik memiliki rentang -1 sampai 1 dan merupakan fungsi alternatif dari lapisan sigmoid. Fungsi ini memiliki persamaan sebagai berikut:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \dots\dots\dots [11]$$

Tahap awal LSTM adalah *forget gate*, yang berfungsi untuk menentukan informasi yang akan dihapus. Proses ini terjadi melalui lapisan sigmoid yang menghasilkan *output* berupa nilai antara 0 hingga 1. Alur *forget gate* pada LSTM dapat dilihat pada Gambar 2.7 berikut:

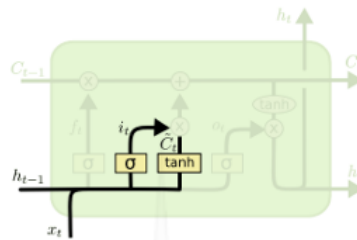


Gambar 2. 7 Alur *Forget Gate* pada LSTM

Perhitungan pada *forget gate* dilakukan dengan memanfaatkan output data sebelumnya (h_{t-1}) dan data input saat ini x_t dengan persamaan berikut:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \dots\dots\dots [12]$$

Tahap kedua adalah *input gate*, yaitu proses pengolahan informasi untuk menentukan data mana yang diperbarui dan kemudian disimpan ke dalam *cell state*. Alur *input gate* pada LSTM dapat dilihat pada Gambar 2.8 berikut:



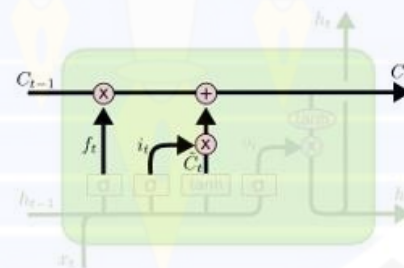
Gambar 2. 8 Alur *Input Gate* pada LSTM

Persamaan fungsi sigmoid dan tangen hiperbolik dari *input gate* ditulis menggunakan persamaan berikut:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \dots\dots\dots [13]$$

$$\hat{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \dots\dots\dots [14]$$

Tahap ketiga melibatkan pembaruan *cell state* yang dapat dilihat pada Gambar 2.9 berikut:

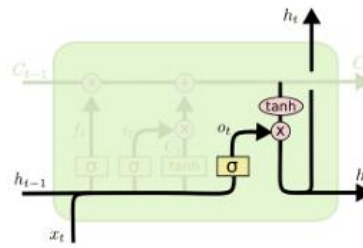


Gambar 2. 9 Alur Pembaruan *Cell State* pada LSTM

Pembaruan *cell state* diperoleh dari hasil perkalian *forget gate* (f_t) dengan *cell state* sebelumnya (C_{t-1}), kemudian ditambahkan dengan *input gate* ($i_t \hat{C}_t$). Persamaan *cell state* tersebut ditampilkan pada persamaan berikut:

$$C_t = f_t C_{t-1} + i_t \hat{C}_t \dots\dots\dots [15]$$

Proses terakhir dalam metode LSTM ini adalah menentukan hasil *output* dengan jaringan yang dapat dilihat pada Gambar 2.10 berikut:



Gambar 2. 10 Alur *Output Gate* pada LSTM

Lapisan sigmoid berfungsi untuk menentukan bagian dari *cell state* yang akan dijadikan *output*, dengan persamaan sebagai berikut:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \dots\dots\dots [16]$$

Setelah nilai dari *output gate* diperoleh, tahap berikutnya adalah memasukkan *cell state* melalui fungsi aktivasi tangen hiperbolik sehingga menghasilkan nilai dalam rentang -1 hingga 1. Nilai tersebut kemudian dikalikan dengan *output gate* dari lapisan sigmoid. Persamaan untuk nilai *output* pada orde t dapat diperoleh dari persamaan berikut:

$$h_t = o_t \tanh(C_t) \dots\dots\dots [17]$$

Dengan h_t mempresentasikan *hidden state* baru, o_t adalah output dari *output gate*, \tanh adalah fungsi aktivasi *tanh*, dan C_t menunjukkan *cell state*. Yang telah diperbarui. Kemudian tahapan terakhir dari analisis metode LSTM, yaitu denormalisasi data untuk mengembalikan data ke dalam data dengan skala aktual.

2.2.3 Denormalisasi Data

Denormalisasi data merupakan proses untuk mengembalikan data yang sebelumnya dinormalisasi ke dalam skala data aslinya. Tujuan dari proses ini adalah untuk mengembalikan hasil keluaran yang masih dalam bentuk *range* 0-1 menjadi sesuai dengan nilai aktual dari data. Denormalisasi data dapat diperoleh menggunakan persamaan berikut:

$$x_{dnorm} = \hat{Y}_t(x_{max} - x_{min}) + x_{min} \dots\dots\dots [18]$$

2.2.4 Evaluasi Metrik

Evaluasi model dilakukan untuk mengukur kinerja model. Kinerja model diukur dengan menggunakan hasil perhitungan RMSE (*Root Mean Square Error*). RMSE dan MAPE (*Mean Absolute Percentage Error*). RMSE adalah metode evaluasi yang digunakan untuk mengukur tingkat akurasi suatu model regresi dengan cara membandingkan nilai prediksi dan nilai aktual. Berdasarkan kriteria yang dikemukakan oleh (Fadhillah *et al.* 2025), jika nilai RMSE kurang dari 10% dari rata-rata data maka model dikategorikan sangat baik. RMSE erat kaitannya dengan MSE (Mean Square Error), yaitu rata-rata kuadrat error, hanya saja RMSE lebih mudah diinterpretasikan karena kembali pada skala asli data. Selain itu, RMSE juga memberikan bobot yang lebih besar pada kesalahan yang signifikan (Anwar, 2023).

MAPE (*Mean Absolute Percentage Error*) merupakan salah satu metrik evaluasi yang umum digunakan dalam peramalan untuk mengukur akurasi suatu model peramalan dalam memprediksi nilai-nilai tertentu. MAPE mengukur rata-rata persentase kesalahan antara nilai aktual dan nilai prediksi dalam suatu dataset. MAPE memberikan gambaran tentang seberapa besar kesalahan rata-rata dalam prediksi sebagai persentase dari nilai aktual. Semakin kecil nilai MAPE, semakin baik model peramalan. Zainun dan Majid yang dikutip oleh (Amalia *et al.* 2017) mengatakan bahwa suatu model mempunyai kinerja sangat baik jika nilai MAPE berada di bawah 10%, dan mempunyai kinerja baik jika nilai MAPE berada di antara 10-20% (Siregar *et al.* 2021). Dengan demikian, RMSE dan MAPE memiliki hubungan yang saling melengkapi. RMSE memberikan informasi mengenai besarnya kesalahan prediksi dalam skala asli data, sedangkan MAPE menunjukkan besarnya kesalahan dalam bentuk persentase. Secara umum, model yang baik ditunjukkan oleh nilai RMSE dan MAPE yang sama-sama kecil.

Perhitungan RMSE dan MAPE dapat dilihat dari persamaan berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \dots\dots\dots [19]$$

$$MAPE = \left(\frac{1}{n} \sum_{t=1}^n \left| \frac{y(t) - y'(t)}{y(t)} \right| \right) \times 100\% \dots\dots\dots [20]$$

Kriteria nilai MAPE dan RMSE ditunjukkan pada Tabel 2.1 dan Tabel 2.2

Tabel 2. 1 Kriteria MAPE

Nilai MAPE (persen)	Kriteria
$< 10 \%$	Kemampuan prediksi sangat baik
$10\% \leq \text{MAPE} < 20\%$	Kemampuan prediksi baik
$20\% \leq \text{MAPE} < 50\%$	Kemampuan prediksi cukup
$\geq 50 \%$	Kemampuan prediksi buruk

Tabel 2. 2 Kriteria RMSE

Nilai MAPE (persen)	Kriteria
$< 10 \%$	Kemampuan prediksi sangat baik
$10\% \leq \text{MAPE} < 20\%$	Kemampuan prediksi baik
$20\% \leq \text{MAPE} < 30\%$	Kemampuan prediksi cukup
$\geq 30 \%$	Kemampuan prediksi buruk

2.2 Penelitian Terdahulu

Dalam penulisan skripsi ini, mencari informasi dari penelitian-penelitian terdahulu sebagai bahan perbandingan, baik dari segi kekurangan maupun kelebihan yang sudah ada. Berikut ini terdapat beberapa penelitian terdahulu yang masih terkait dengan tema yang akan diteliti:

Tabel 2. 3 Penelitian Terdahulu

No	Uraian	Keterangan
1	Peneliti (Tahun)	Zengshun Chen, Chenfeng Yuan, Haofan Wu, Likai Zhang, Ke Li, Xuanyi Xue, Lei Wu (2022)
	Judul Penelitian	<i>An Improved Method Based on EEMD-LSTM to Predict Missing Measured Data of Structural Sensors</i>
	Hasil Penelitian	Hasil Penelitian menunjukkan bahwa metode hibrida berbasis <i>Ensemble Empirical Mode Decomposition</i> (EEMD) dan <i>Long Short-Term Memory</i> (LSTM) terbukti efektif dalam melengkapi data sensor yang

		<p>hilang. Studi ini menunjukkan bahwa metode EEMD-LSTM memiliki akurasi lebih tinggi dibandingkan metode <i>deep learning</i> tradisional seperti <i>Deep Neural Networks</i> (DNN), <i>Gated Recurrent Unit</i> (GRU), dan <i>Bidirectional Gated Recurrent Unit</i> (BiGRU).</p>
2	Peneliti (Tahun)	Nam Nguyen Vu Nhat, Duc Nguyen Huu, Thu Nguyen Thi Hoai (2023)
	Judul Penelitian	<i>Evaluating the EEMD-LSTM Model for Short-Term Forecasting of Industrial Power Load: A Case Study in Vietnam</i>
	Hasil Penelitian	<p>Kesimpulan dari jurnal ini adalah bahwa model <i>Ensemble Empirical Mode Decomposition-Long Short-Term Memory</i> (EEMD-LSTM) terbukti efektif untuk peramalan beban listrik industri jangka pendek. Model ini menunjukkan akurasi yang lebih tinggi dibandingkan model lain seperti <i>Linear Regression</i> (LR), <i>Artificial Neural Network</i> (ANN), dan <i>LSTM</i> biasa, dengan nilai kesalahan yang lebih rendah (RMSE, n-RMSE, dan MAPE) untuk prediksi 1 langkah dan 24 langkah ke depan.</p> <p>Keunggulan utama model EEMD-LSTM adalah kemampuannya menangkap pola kompleks dan non-linear dalam data konsumsi listrik industri, yang sulit ditangani oleh model konvensional. Penelitian ini menegaskan bahwa metode kombinasi <i>decomposition</i> dan <i>deep learning</i> dapat meningkatkan akurasi peramalan beban listrik, sehingga dapat membantu pengambilan keputusan dalam pengelolaan energi industri secara lebih efisien.</p>

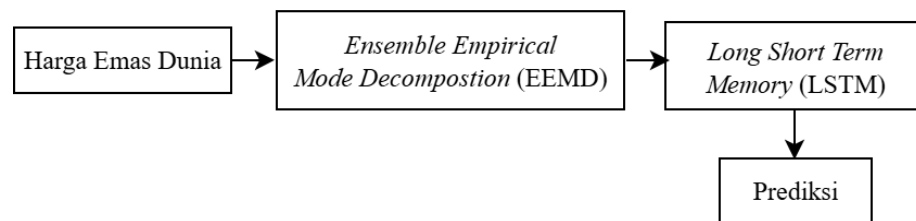
3	Peneliti (Tahun)	Tania Giovani Lasijan, Rukun Santoso, dan Arief Rachman Hakim (2023)
	Judul Penelitian	Prediksi Harga Emas Dunia Menggunakan Metode <i>Long-Short Term Memory</i>
	Hasil Penelitian	Penelitian mengenai prediksi harga emas dengan metode Long Short-Term Memory (LSTM) dilakukan melalui pendekatan <i>trial and error</i> untuk menentukan proporsi data <i>training</i> dan <i>testing</i> serta pemilihan <i>hyperparameter</i> . Dari proses ini dihasilkan 18 model percobaan LSTM. Setelah dilakukan pelatihan pada seluruh model, diperoleh bahwa model terbaik adalah Trial 2, dengan nilai MAPE sebesar 2,70601. Model ini menggunakan <i>hyperparameter</i> berupa batch size = 1, units = 1, optimizer = AdaGrad, learning rate = 0,1, serta jumlah <i>epochs</i> sebanyak 500.
4	Peneliti (Tahun)	Fatkhurokhman Fauzi, Syifa Aulia, Ahmad Reyhan Syaifullah, dan Tiani Wahyu Utami (2024)
	Judul Penelitian	Peramalan Harga Emas Menggunakan Pendekatan <i>Long-Short Term Memory</i> (LSTM)
	Hasil Penelitian	Hasil penelitian menunjukkan bahwa bahwa model <i>Long Short-Term Memory</i> (LSTM) berhasil digunakan untuk meramalkan harga emas dengan akurasi yang baik. Model terbaik yang diperoleh menggunakan parameter learning rate 0,01, 10 neuron hidden, dan 100 epoch, menghasilkan nilai <i>Mean Absolute Percentage Error</i> (MAPE) sebesar 3,499% untuk data training dan 5,311% untuk data testing. model LSTM mampu menangkap pola harga emas yang fluktuatif tanpa mengalami <i>overfitting</i> atau <i>underfitting</i> . Prediksi harga emas untuk 30 periode ke depan menunjukkan pola yang tetap

		fluktuatif, dengan kenaikan signifikan pada periode ke-24.
5	Peneliti (Tahun)	Sri Herawati, Firmansyah Adiputra, M. Latif, Aeri Rachmad (2017)
	Judul Penelitian	<i>Implementing Method of Ensemble Empirical Mode Decomposition and Recurrent Neural Network for Gold Price Forecasting</i>
	Hasil Penelitian	Hasil penelitian menunjukkan bahwa kombinasi EEMD-RNN menghasilkan akurasi yang lebih baik dibandingkan metode EEMD-Feedforward Neural Network (FNN), dengan nilai MSE sebesar 0,003460 dan RMSE sebesar 0,058826 pada arsitektur terbaik 6-9-1. Keunggulan utama metode ini terletak pada kemampuan EEMD dalam mendekomposisi data harga emas menjadi beberapa <i>Intrinsic Mode Functions (IMF)</i> sehingga tren lebih mudah dianalisis, sementara RNN mampu menangkap pola fluktuasi harga yang kompleks dengan memori internalnya. Penelitian ini menyimpulkan bahwa EEMD-RNN lebih efektif dalam memprediksi harga emas dibandingkan metode tradisional, serta merekomendasikan pengembangan model lebih lanjut menggunakan LSTM atau GRU untuk meningkatkan akurasi prediksi dalam berbagai kondisi pasar

2.3 Kerangka Pemikiran

Kerangka pemikiran adalah model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting, kerangka pemikiran ini merupakan penjelasan sementara terhadap gejala-gejala yang menjadi objek permasalahan (Sugiyono, 2017). Kerangka pemikiran

dalam memprediksi Harga Emas dengan menggunakan EEMD-LSTM ditunjukkan pada Gambar 2.11.



Gambar 2. 11 Kerangka Pemikiran

Penelitian ini menggunakan *Ensemble Empirical Mode Decomposition* (EEMD) untuk mendekomposisikan atau menguraikan data harga emas. Proses dekomposisi data menghasilkan sejumlah *Intrinsic Mode Function* (IMF) dan sisaan IMF (residu) yang selanjutnya akan digunakan sebagai input untuk model *Long Short-Term Memory* (LSTM). Setiap komponen IMF dan residu akan diproses dalam model LSTM untuk melakukan pelatihan, optimasi, dan evaluasi hingga diperoleh model terbaik untuk masing-masing komponen. Hasil prediksi dari setiap komponen IMF dan residu kemudian digabungkan untuk menghasilkan prediksi akhir harga emas yang lebih akurat. Dengan pendekatan ini, diharapkan dapat memberikan gambaran arah pergerakan harga di masa depan, sehingga menjadi acuan bagi investor dalam mengambil keputusan yang tepat serta mengantisipasi risiko pasar. Oleh karena itu, diperlukan metode yang mampu menangkap sifat non linier dan non stasioner dari data harga emas untuk menghasilkan prediksi yang lebih akurat.