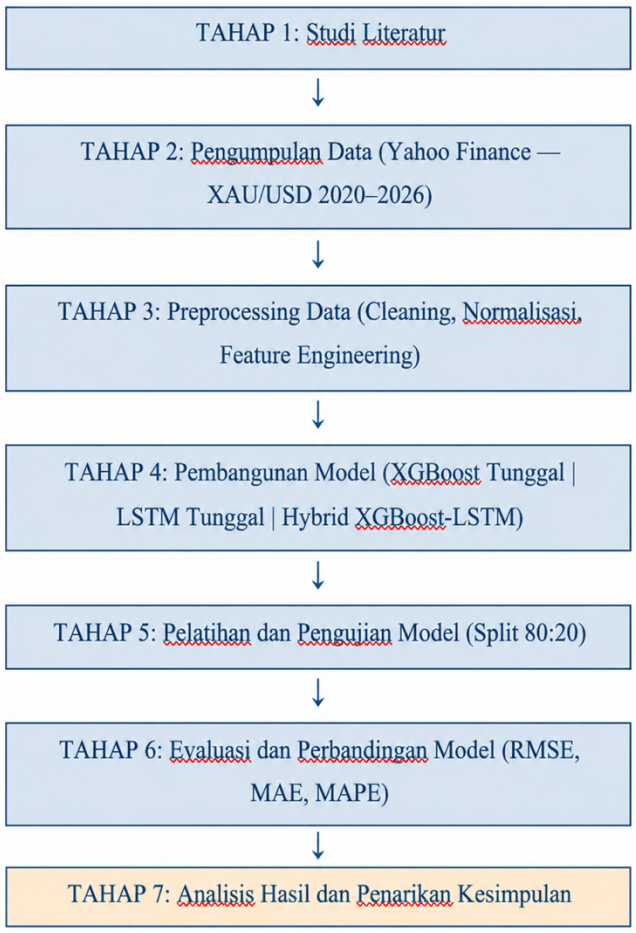
BAB III
METODOLOGI PENELITIAN

3.1 Alur Penelitian

Penelitian ini dilaksanakan secara sistematis mengikuti alur yang terdiri dari tujuh tahapan utama. Setiap tahapan saling bergantung dan harus diselesaikan secara berurutan sebelum melanjutkan ke tahapan berikutnya. Secara keseluruhan, alur penelitian ini mengikuti siklus standar pengembangan model machine learning: dari pengumpulan data, preprocessing, pembangunan model, evaluasi, hingga penarikan kesimpulan.



Gambar 3. 1 Alur Penelitian

3.2 Studi Literatur

Tahap pertama penelitian adalah studi literatur yang bertujuan membangun landasan teori yang kuat dan mengidentifikasi research gap yang hendak diisi oleh penelitian ini. Studi literatur dilakukan dengan mengkaji jurnal-jurnal ilmiah yang relevan, khususnya jurnal terakreditasi SINTA di Indonesia. Hasil studi literatur menjadi dasar penentuan metode, arsitektur model, dan metrik evaluasi yang digunakan dalam penelitian ini.

Studi literatur mencakup kajian terhadap delapan jurnal terakreditasi SINTA yang membahas prediksi harga emas, algoritma LSTM, algoritma XGBoost, dan model hybrid. Dari tinjauan tersebut disimpulkan bahwa belum terdapat penelitian yang mengimplementasikan model hybrid XGBoost-LSTM untuk memprediksi harga emas dunia (XAU/USD), inilah celah penelitian yang menjadi justifikasi utama penelitian ini.

3.3 Pengumpulan Data

3.3.1 Sumber Data

Data yang digunakan dalam penelitian ini adalah data harga emas dunia (XAU/USD) harian yang diperoleh dari Yahoo Finance (https://finance.yahoo.com) dengan ticker simbol GC=F (Gold Futures). Yahoo Finance dipilih sebagai sumber data karena menyediakan data historis yang lengkap, konsisten, dapat diakses secara gratis, dan telah banyak digunakan dalam penelitian-penelitian serupa di bidang machine learning untuk prediksi harga komoditas.

Selain data harga emas, penelitian ini juga mengumpulkan data indikator makroekonomi sebagai fitur tambahan yang diperoleh dari sumber-sumber berikut: data indeks Dolar AS (DXY) dari Yahoo Finance dengan ticker DX-Y.NYB; data harga minyak mentah WTI dari Yahoo Finance dengan ticker CL=F; dan data yield obligasi pemerintah AS tenor 10 tahun (US 10-Year Treasury Yield) dari Yahoo Finance dengan ticker %5ETNX.

3.3.2 Rentang Data

Data dikumpulkan untuk rentang waktu 25 tahun, yaitu periode 1 Januari 2000 hingga 20 Februari 2026. Rentang waktu ini dipilih karena beberapa alasan:

1. Mencakup berbagai kondisi pasar yang ekstrem, mulai dari pandemi COVID-19 pada 2020, pemulihan ekonomi 2021- 2022, tekanan inflasi 2022–2023, hingga lonjakan harga emas 2024.
2. Menghasilkan volume data yang cukup besar (±1.260 data harian) untuk melatih model LSTM secara optimal
3. Sesuai dengan praktik yang umum digunakan dalam penelitian prediksi harga emas berbasis machine learning, seperti yang dilakukan oleh Ramadhan et al. (2025) yang menggunakan 4.545 data harian.

3.3.3 Variabel Penelitian

Variabel yang digunakan dalam penelitian ini terdiri dari satu variabel dependen (target) dan beberapa variabel independen (fitur input). Variabel dependen adalah harga penutupan (Close Price) XAU/USD harian. Variabel independen terdiri dari data OHLCV (Open, High, Low, Close, Volume) harga emas harian, indikator teknikal turunan, dan variabel makroekonomi pendukung.

Tabel 3. 1 Variabel Penelitian

No	Variabel	Jenis	Keterangan	Sumber
1	Close Price XAU/USD	Dependen (Target)	Harga penutupan emas harian variabel yang diprediksi	Yahoo Finance (GC=F)
2	Open Price	Independen	Harga pembukaan emas harian	Yahoo Finance (GC=F)
3	High Price	Independen	Harga tertinggi emas harian	Yahoo Finance (GC=F)

4	Low Price	Independen	Harga terendah emas harian	Yahoo Finance (GC=F)
5	Volume	Independen	Volume perdagangan emas harian	Yahoo Finance (GC=F)
6	MA-7, MA-14, MA-30	Independen	Moving Average periode 7, 14, 30 hari	Turunan dari Close Price
7	RSI-14	Independen	Relative Strength Index periode 14 hari	Turunan dari Close Price
8	Bollinger Bands	Independen	Upper Band, Middle Band, Lower Band	Turunan dari Close Price
9	MACD	Independen	Moving Average Convergence Divergence	Turunan dari Close Price
10	Indeks Dolar AS (DXY)	Independen	Nilai tukar Dolar AS terhadap keranjang mata uang	Yahoo Finance (DX-Y.NYB)
11	Harga Minyak WTI	Independen	Harga penutupan minyak mentah WTI harian	Yahoo Finance (CL=F)
12	US 10-Year Treasury Yield	Independen	Yield obligasi pemerintah AS tenor 10 tahun	Yahoo Finance (%5ETNX)

3.4 Preprocessing Data

Preprocessing data merupakan tahapan yang sangat kritis untuk memastikan kualitas data yang digunakan dalam pelatihan model. Data mentah dari Yahoo Finance seringkali mengandung inkonsistensi yang perlu dibersihkan sebelum dapat diproses oleh algoritma. Tahapan preprocessing dalam penelitian ini terdiri dari empat langkah berurutan.

3.4.1 Data Cleaning

Langkah pertama adalah membersihkan data dari nilai yang hilang (missing value) dan duplikasi. Data harga emas pada hari libur bursa (weekend dan public holiday) secara otomatis tidak tersedia di Yahoo Finance sehingga tidak perlu dihapus secara manual. Namun, jika terdapat hari perdagangan aktif yang memiliki nilai kosong akibat inkonsistensi sumber data, penanganan dilakukan dengan metode forward fill, yaitu mengisi nilai yang hilang dengan nilai hari perdagangan terakhir yang tersedia. Setelah pembersihan, dilakukan pengecekan konsistensi data untuk memastikan tidak ada nilai negatif atau nilai nol pada kolom harga yang secara logis tidak mungkin terjadi.

3.4.2 Penggabungan Data Multi-Variabel

Data dari berbagai sumber (harga emas, DXY, minyak WTI, dan Treasury yield) digabungkan berdasarkan kolom tanggal menggunakan operasi inner join, sehingga hanya baris dengan tanggal yang tersedia di semua sumber yang digunakan. Hal ini penting untuk memastikan konsistensi temporal antar variabel. Setelah penggabungan, dilakukan pengecekan kembali untuk memastikan tidak ada baris yang memiliki missing value pada kolom mana pun.

3.4.3 Feature Engineering

Feature engineering dilakukan untuk menciptakan fitur-fitur teknikal yang bermakna dari data harga historis. Fitur-fitur ini diharapkan dapat membantu model menangkap pola pergerakan harga yang tidak langsung terlihat dari data OHLCV mentah. Fitur teknikal yang dihitung adalah sebagai berikut:

1. Moving Average (MA): MA-7, MA-14, dan MA-30 dihitung sebagai rata-rata bergerak Close Price dalam 7, 14, dan 30 hari terakhir. MA berfungsi menghaluskan fluktuasi jangka pendek dan mengidentifikasi tren jangka menengah-panjang.
2. Relative Strength Index (RSI-14): RSI dihitung pada periode 14 hari menggunakan rumus RSI = 100 - (100 / (1 + RS)), di mana RS = rata-rata kenaikan / rata-rata penurunan dalam 14 hari. RSI mengukur kekuatan momentum pergerakan harga dengan nilai antara 0–100.

3. Bollinger Bands: Upper Band = MA-20 + (2 × standar deviasi 20 hari), Lower Band = MA-20 - (2 × standar deviasi 20 hari). Bollinger Bands mengukur volatilitas harga dan memberikan sinyal overbought/oversold.
4. MACD: Dihitung sebagai selisih antara EMA-12 dan EMA-26, dengan Signal Line berupa EMA-9 dari MACD. MACD digunakan sebagai indikator momentum dan arah tren.
5. Fitur Lag: Nilai Close Price pada t-1, t-2, t-3, t-5, dan t-10 digunakan sebagai fitur historis eksplisit untuk membantu model memahami konteks pergerakan harga terbaru.

3.4.4 Normalisasi Data

Normalisasi dilakukan menggunakan metode Min-Max Normalization dengan rumus x' = (x - x_min) / (x_max - x_min) sehingga seluruh nilai fitur berada dalam rentang [0, 1]. Normalisasi sangat penting untuk proses pelatihan LSTM karena algoritma ini sensitif terhadap perbedaan skala antar fitur. Perlu dicatat bahwa nilai x_min dan x_max dihitung hanya dari data training — bukan dari keseluruhan data — untuk menghindari terjadinya data leakage. Setelah prediksi dihasilkan oleh model, nilai hasil prediksi didenormalisasi kembali ke skala asli menggunakan transformasi invers dari Min-Max Normalization.

3.4.5 Pembagian Data Training dan Testing

Data dibagi secara kronologis dengan rasio 80:20, di mana 80% data awal digunakan untuk training dan 20% data terbaru digunakan untuk testing. Dari total 6.391 data harian (periode 2000–2026), sekitar 5.112 data digunakan untuk training dan 1.278 data (kurang lebih satu tahun terakhir) digunakan untuk testing. Pembagian kronologis ini memastikan bahwa model tidak pernah melihat data masa depan selama pelatihan, sehingga evaluasi performa yang dihasilkan bersifat realistis dan dapat dipercaya.

3.5 Pembangunan Model

Penelitian ini membangun tiga model yang akan dibandingkan performanya: model XGBoost tunggal sebagai baseline pertama, model LSTM tunggal sebagai baseline kedua, dan model hybrid XGBoost-LSTM sebagai model yang diusulkan. Perbandingan ketiga model ini memungkinkan penilaian yang objektif terhadap kontribusi pendekatan hybrid dibandingkan model individual.

3.5.1 Model XGBoost Tunggal

Model XGBoost dibangun menggunakan library XGBoost versi terbaru pada Python. Pada model ini, seluruh variabel input (fitur OHLCV, indikator teknikal, dan variabel makroekonomi) digunakan langsung sebagai fitur untuk memprediksi Close Price XAU/USD pada hari berikutnya (t+1). Hyperparameter yang akan dioptimasi meliputi: n_estimators (jumlah pohon), max_depth (kedalaman maksimum pohon), learning_rate (laju pembelajaran), subsample (proporsi sampel per pohon), colsample_bytree (proporsi fitur per pohon), serta parameter regularisasi alpha (L1) dan lambda (L2). Optimasi hyperparameter dilakukan menggunakan Grid Search Cross Validation dengan time series split untuk memastikan tidak ada data leakage dalam proses validasi.

3.5.2 Model LSTM Tunggal

Model LSTM dibangun menggunakan library TensorFlow/Keras pada Python. Arsitektur LSTM yang digunakan terdiri dari dua lapisan LSTM bertumpuk (stacked LSTM) dengan lapisan Dropout di antara keduanya untuk mencegah overfitting, diikuti oleh satu lapisan Dense sebagai output layer. Input model berupa sequence data time series dengan window size (lookback period) yang ditentukan melalui eksperimen. Parameter utama yang dioptimasi meliputi: jumlah unit LSTM pada setiap lapisan, dropout rate, batch size, jumlah epoch, learning rate optimizer Adam, dan panjang window sequence. Model dikompilasi menggunakan fungsi loss Mean Squared Error (MSE) dan optimizer Adam.

Tabel 3. 2 Konfigurasi Arsitektur LSTM

Layer	Tipe	Parameter	Fungsi
Input Layer	Input	Shape: (window_size, n_features)	Menerima data sekuensial
Layer 1	LSTM	Units: 128, return_sequences=True	Menangkap pola temporal jangka panjang
Layer 2	Dropout	Rate: 0.2	Mencegah overfitting
Layer 3	LSTM	Units: 64, return_sequences=False	Memperhalus representasi temporal
Layer 4	Dropout	Rate: 0.2	Mencegah overfitting
Output Layer	Dense	Units: 1, activation: linear	Menghasilkan nilai prediksi Close Price

3.5.3 Model Hybrid XGBoost-LSTM

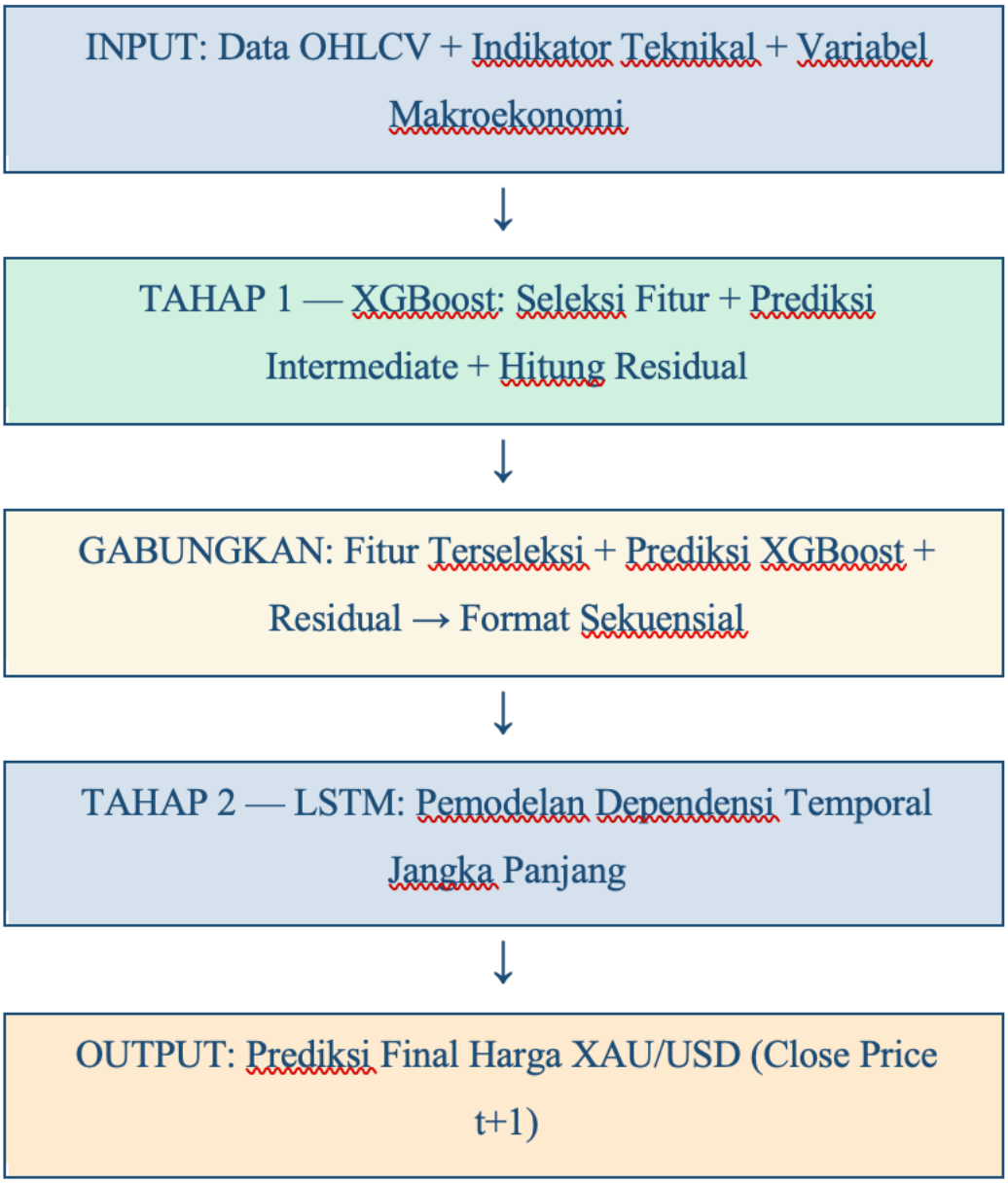
Model hybrid dirancang dengan mekanisme dua tahap yang saling melengkapi, memanfaatkan kekuatan XGBoost dalam seleksi fitur dan kekuatan LSTM dalam pemodelan temporal.

Tahap 1 XGBoost sebagai Feature Extractor:

Pada tahap pertama, XGBoost dilatih menggunakan seluruh variabel input untuk menghasilkan dua output: (1) prediksi harga awal (intermediate prediction) yang merepresentasikan estimasi berbasis fitur terstruktur, dan (2) skor feature importance yang mengidentifikasi variabel-variabel paling berpengaruh. Residual atau selisih antara prediksi XGBoost dan nilai aktual kemudian dihitung sebagai representasi komponen non-linear yang belum dapat ditangkap oleh XGBoost sendiri.

Tahap 2 LSTM sebagai Temporal Modeler:

Pada tahap kedua, LSTM menerima input berupa gabungan dari: (1) fitur-fitur yang telah diseleksi dan dibobot berdasarkan feature importance dari XGBoost, (2) prediksi intermediate dari XGBoost, dan (3) residual XGBoost. Data gabungan ini disusun dalam format sekuensial dengan window size tertentu dan diproses oleh LSTM untuk memodelkan dependensi temporal jangka panjang yang tidak tertangkap oleh XGBoost. Output dari LSTM adalah prediksi final harga XAU/USD.



Gambar 3. 2 Arsitektur Model Hybrid XGBoost-LSTM

3.6 Pelatihan dan Pengujian Model

3.6.1 Proses Pelatihan

Proses pelatihan ketiga model dilakukan secara terpisah menggunakan data training (80% data awal, periode 2000–2026). Untuk model XGBoost, pelatihan

dilakukan menggunakan metode boosting iteratif dengan early stopping untuk mencegah overfitting. Untuk model LSTM dan model hybrid, pelatihan dilakukan selama maksimum 200 epoch dengan callback *EarlyStopping* yang memantau validation loss pelatihan dihentikan secara otomatis jika validation loss tidak membaik dalam 20 epoch berturut-turut. Untuk model *hybrid*, pelatihan dilakukan secara sekuensial: XGBoost dilatih terlebih dahulu, kemudian hasil outputnya digunakan sebagai input tambahan untuk melatih LSTM.

3.6.2 Optimasi Hyperparameter

Optimasi hyperparameter dilakukan untuk menemukan konfigurasi terbaik dari setiap model. Metode yang digunakan adalah Grid Search dengan Time Series Cross Validation (5-fold). Time Series Cross Validation digunakan karena berbeda dengan k-fold cross validation biasa, metode ini mempertahankan urutan kronologis data sehingga tidak ada kebocoran informasi dari data masa depan ke data masa lalu selama proses validasi.

Tabel 3. 3 Rentang Hyperparameter yang Dioptimasi

Model	Hyperparameter	Rentang Nilai	Keterangan
XGBoost	n_estimators	50, 100	Jumlah pohon keputusan (*decision trees*) yang dibangun secara sekuensial.
	max_depth	3, 5	Batas kedalaman pohon (dibuat dangkal agar bertindak sebagai *weak learner*).
	learning_rate	0.05, 0.1	Kecepatan penyesuaian bobot (koreksi error) antar pohon.
	subsample	0.8, 1.0	Rasio acak observasi data baris untuk melatih setiap pohon.

	colsample_bytree	0.8, 1.0	Rasio acak kolom fitur penjelas untuk melatih setiap pohon.
LSTM	lstm_units_1	64, 128	Jumlah memori sel pada lapisan LSTM pertama (pola jangka panjang).
	lstm_units_2	32, 64	Jumlah memori sel pada lapisan LSTM kedua (pemadatan fitur).
	dropout_rate	0.1, 0.2	Persentase pemadaman neuron untuk mencegah *overfitting*.
	learning_rate	0.001, 0.002	Kecepatan konvergensi algoritma optimisasi
	batch_size	16, 32	Jumlah sampel sekuens data yang dieksekusi per satu propagasi maju/mundur.

3.6.3 Pengujian Model

Setelah proses pelatihan dan optimasi hyperparameter selesai, ketiga model diuji menggunakan data testing (20% data terbaru, periode 2026). Pada tahap pengujian, model hanya menerima data input tanpa melihat nilai target sepenuhnya mensimulasikan kondisi prediksi dunia nyata. Hasil prediksi dari ketiga model kemudian didenormalisasi kembali ke skala harga asli (USD per troy ons) sebelum dievaluasi menggunakan metrik RMSE, MAE, dan MAPE.

3.7 Evaluasi dan Perbandingan Model

Evaluasi dilakukan dengan menghitung tiga metrik kesalahan prediksi untuk masing-masing model: RMSE (*Root Mean Squared Error*), MAE (*Mean Absolute Error*), dan MAPE (Mean Absolute Percentage Error). Ketiga metrik ini dipilih karena saling melengkapi: RMSE sensitif terhadap kesalahan besar (outlier), MAE

memberikan gambaran kesalahan rata-rata yang lebih stabil, dan MAPE memberikan interpretasi dalam persentase yang mudah dipahami secara praktis.

Perbandingan performa dilakukan secara kuantitatif menggunakan tabel komparasi nilai RMSE, MAE, dan MAPE dari ketiga model. Model dengan nilai RMSE, MAE, dan MAPE terkecil secara keseluruhan ditetapkan sebagai model terbaik. Selain evaluasi kuantitatif, dilakukan juga analisis visual dengan memplot grafik perbandingan antara harga aktual dan hasil prediksi dari ketiga model pada data testing untuk memberikan gambaran visual kualitas prediksi.

3.8 Alat dan Lingkungan Penelitian

Implementasi seluruh model dalam penelitian ini dilakukan menggunakan bahasa pemrograman Python. Pemilihan Python didasarkan pada ketersediaan ekosistem library machine learning yang sangat lengkap, dokumentasi yang komprehensif, dan penggunaannya yang sudah menjadi standar de facto dalam komunitas penelitian machine learning global.

Tabel 3. 4 Spesifikasi Alat dan Library yang Digunakan

Komponen	Nama / Versi	Fungsi
Bahasa Pemrograman	Python 3.10+	Bahasa implementasi utama seluruh model
Library XGBoost	xgboost ≥ 2.0	Implementasi model XGBoost
Library Deep Learning	TensorFlow / Keras ≥ 2.12	Implementasi model LSTM dan model hybrid
Library Data Processing	pandas ≥ 2.0, numpy ≥ 1.24	Manipulasi dan preprocessing data
Library Visualisasi	matplotlib ≥ 3.7, seaborn ≥ 0.12	Visualisasi data dan hasil prediksi
Library Teknikal Indikator	ta-lib atau pandas-ta	Perhitungan indikator teknikal (RSI, MACD, Bollinger)
Library Machine Learning	scikit-learn ≥ 1.3	Preprocessing, evaluasi, dan hyperparameter tuning
Sumber Data	yfinance ≥ 0.2	Mengunduh data historis dari Yahoo Finance secara otomatis

Lingkungan Pengembangan	Jupyter Notebook / Google Colab	Lingkungan eksekusi kode interaktif
Hardware (Minimum)	RAM 8 GB, CPU Intel Core i5 / GPU	Spesifikasi minimum yang direkomendasikan