BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam sistem keuangan masa kini, pasar modal memegang peranan kursial sebagai mekanisme untuk mengumpulkan dan mengalokasikan dana dari mereka yang memiliki surplus modal ke pihak-pihak yang memerlukan pendanaan (Hidriani et al., 2025). Dalam konteks perekonomian Indonesia, pergerakan saham memiliki peran strategis sebagai barometer stabilitas ekonomi sekaligus indikator arah kebijakan moneter dan fiskal. Di antara berbagai instrumen investasi yang tersedia, saham sering kali menjadi opsi utama bagi investor karena proses imbal hasilnya yang lebih tinggi jika dibandingkan dengan obligasi atau deposito. Meski demikian, harga saham cenderung berubah-ubah secara drastis, tidak stabil dalam waktu panjang, dan terpengaruh oleh faktor internal seperti performa perusahaan serta eksternal seperti tingkat inflasi, suku bunga, dan peristiwa global (Azkiya et al., 2020; Hisam, 2024).

Ketidakpastian ini mendorong para peniliti untuk mengembangkan teknik prediksi yang bisa menangkap pola-pola non-linear dan hubungan dinamis dalam data keuangan. Metode tradisional seperti regresi linear sering kali gagal karena asumsi-asumsi stasioneritas dan linearitas jarang terpenuhi dalam data pasar saham. (Sapanji et al., 2023; Wibawa, 2021). Hal ini kemudian membuka peluang bagi pendekatan berbasis *machine learning* yang lebih fleksibel, karena teknologi ini mampu mengenali pola tersembunyi dan hubungan non-linear dalam data keuangan (Hutomo, 2025). Seiring perkembangan teknologi, algoritma pohon seperti *Extreme Gradient Boosting* (XGBoost), *Light Gradient Boosting Machine* (LightGBM), dan Random Forest telah banyak diaplikasikan dalam analisis pasar keuangan. (Yunita et al*., 2023*). Ketiga model ini unggul dalam menangani hubungan non-linear, mengelola variabel dengan skala yang berbeda, serta mengurangi risiko *overfitting* melalui teknik boosting dan bagging (Ananda Surya et al., 2025) .

Meskipun demikian, masing-masing model masih memiliki kelemahan ketika digunakan secara mandiri. Model tunggal dapat memberikan performa yang baik pada kondisi tertentu, tetapi tidak selalu konsisten di seluruh horizon prediksi (Fauziah, 2025). Untuk meningkatkan stabilitas dan akurasi, penilitian modern banyak mengadopsi pendekatan *ensemble,* khususnya *fusion ensemble,* yaitu teknik penggabungan model-model dasar untuk memperoleh hasil prediksi yang kuat. Strategi penggabungan seperti *Equal Weighted, Ridge Stacking, Constrained Wolf Optimize, dan Lasso Fusion* (Chubanov, 2023; Ge et al., dirancang untuk mengoptimalkan kontribusi antar-model agar ıanya akurat pada horizon jangka pendek (t+1), tetapi juga ngka menengah (t + 2 dan t +3) (Kayit & Ismail, 2025).

Sebagai objek penelitian, digunakan dua saham sektor pertambangan yaitu PT Aneka Tambang Tbk (ANTM) dan PT Adaro Energy Indonesia Tbk (ADRO). Kedua saham ini dipilih karena memiliki likuiditas tinggi dan volatilitas yang mencerminkan dinamika sektor pertambangan nasional yang dipengaruhi fluktuasi harga komoditas global seperti nikel dan batu bara. Sifat volatil dan non-stasioner pada kedua saham tersebut memerlukan metode prediksi yang lebih adaptif.Untuk menstabilkan variansi dan mengatasi karakteristik data keuangan yang non-stasioner, harga saham umum diubah menjadi *log-return* sehinggan perubahan relatif antar waktu dapat dianalisis lebih konsisten (Carlos et al., 2025).

Berbagai penelitian sebelumnya telah menerapkan model *machine learing* tunggal pada data keuangan, namun pemanfaatan *Hybrid Multi-Model Fusion Ensenmble* yang menggabungkan model XGBoost, LightGBM, dan Random Forest secara simultan masih terbatas, terutama untuk prediksi multi-step pada horizon menengah (t+2 dan t+3). Selain itu, penelitian terdahulu umumnya hanya berfokus pada perbandingan akurasi tanpa menguji signifikansi statistik perbedaan performa antar-model. Padahal, pengujian seperti Uji Diebold-Mariano (DM-test) penting untuk memastikan apakah peningkatan akurasi benar-benar signifikan secara statistik (Hyndman & Athanasopoulos, 2021).

Dengan rancangan tersebut, penelitian "**Prediksi Multi-Step Harga Saham ANTM dan ADRO Menggunakan Hybrid *Fusion Ensemble* XGBoost, LightGBM, dan Random Forest**" diharapkan mampu memberikan kontribusi metodologis dalam pengembangan sistem prediksi harga saham di Indonesia. Pendekatan *fusion ensemble* yang digunakan tidak hanya berfokus pada peningkatan akurasi, tetapi juga kestabilan dan efisiensi prediksi dalam menghadapi volatilitas tinggi pada saham-saham sektor pertambangan. Hasil yang diperoleh nantinya diharapkan dapat menjadi pijakan analitis untuk membangun model prediksi yang lebih adaptif dan relevan di masa mendatang.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian adalah:

1. Bagaimana kinerja model dasar XGBoost, LightGBM, dan Random Forest dalam memprediksi harga saham ANTM dan ADRO secara *multi-step* (t + 1, t + 2, t + 3) ?
2. Bagaimana efektivitas pendekatan Hybrid Multi-Model Fusion Ensemble (F0–F4) dalam meningkatkan akurasi, kestabilan, dan efisiensi hasil ndingkan pendekatan sederhana?

1.3 Tujuan dan Manfaat Penelitian

1.3.1 Tujuan Penelitian

Sejalan dengan rumusan masalah yang telah dikemukakan, maka tujuan dari penelitian adalah:

1. Mengevaluasi performa tiga model dasar XGBoost, LightGBM, dan Random Forest dalam melakukan prediksi multi-step harga saham ANTM dan ADRO.
2. Menganalisis efektivitas lima strategi penggabungan model (Fusion F0–F4) dalam meningkatkan akurasi, kestabilan, serta efisiensi hasil prediksi dibandingkan pendekatan sederhana.

1.3.2 Manfaat Penelitian

Penelitian yang dilakukan diharapkan memberikan manfaat baik secara akademis maupun praktis. Secara akademis, penelitian yang dilakukan memperluas literatur mengenai penerapan *machine learning* ensemble untuk prediksi harga saham di pasar modal Indonesia, khususnya melalui pendekatan Hybrid Multi-Model Fusion Ensemble (F0–F4) yang menggabungkan tiga algoritma utama. Sedangkan secara praktis, hasil penelitian yang dilakukan dapat menjadi referensi bagi investor, analis, dan pembuat kebijakan dalam memahami potensi penggunaan *ensemble learning* untuk meningkatkan akurasi dan efisiensi prediksi harga saham, sekaligus mendukung pengambilan keputusan berbasis data di sektor keuangan.

1.4 Landasan Teori

1.4.1 Saham

Saham merupakan salah satu instrumen keuangan yang paling diminati karena memberikan kesempatan bagi investor untuk memperoleh keuntungan dari pertumbuhan nilai perusahaan dan pembagian dividen (Budiharto, 2021). Saham dapat diartikan sebagai tanda kepemilikan terhadap sebagian nilai suatu perusahaan, yang memberikan hak kepada pemegangnya untuk memperoleh sebagian keuntungan atau capital gain apabila harga saham meningkat di pasar. Namun, fluktuasi harga yang tinggi membuat investasi saham mengandung risiko yang signifikan akibat pengaruh kondisi ekonomi, kinerja perusahaan, serta faktor eksternal lainnya (Vuong et al., 2024).

Harga saham pada dasarnya terbentuk melalui mekanisme permintaan dan penawaran (*supply and demand*) di pasar. Ketika permintaan meningkat karena prospek positif suatu perusahaan, harga saham akan naik; sebaliknya, tekanan jual akan menurunkan harga. Pergerakan ini mencerminkan reaksi pasar terhadap aru, baik fundamental seperti laporan keuangan, suku bunga, konomi, maupun non-fundamental seperti sentimen publik, u isu global (Vuong et al., 2024). Dengan demikian, harga asil interaksi dinamis antara rasionalitas ekonomi dan faktor

Namun, pola pergerakan harga saham bersifat non-linear, fluktuatif, dan sulit diprediksi karena banyak faktor saling memengaruhi dalam waktu yang bersamaan. Pendekatan tradisional seperti analisis regresi linier sering kali gagal menangkap pola kompleks tersebut (Kayit & Ismail, 2025). Oleh karena itu, penggunaan metode berbasis *machine learning* menjadi alternatif yang lebih adaptif dan efisien, karena mampu mempelajari hubungan non-linear antarvariabel dan menyesuaikan diri terhadap dinamika pasar yang cepat (Amellia Kharis et al., 2025). Dalam penelitian yang dilakukan, saham dipilih sebagai objek utama karena pergerakannya yang sensitif terhadap perubahan informasi dan dapat merepresentasikan kondisi ekonomi yang sesungguhnya.

1.4.2 *Log Return*

Perubahan harga saham umumnya bersifat non-stasioner, sehingga perlu ditransformasikan agar model prediksi dapat bekerja secara stabil. Salah satu transformasi yang umum digunakan dalam analisis keuangan adalah log return, yaitu perubahan relatif harga yang diukur dengan logaritma natural dari rasio harga saat ini terhadap harga sebelumnya (Hudson & Gregoriou, 2015).

$$r_t = \ln\left(\frac{C_t}{C_{t-1}}\right) \tag{1}$$

Keterangan:

r_t : Log-return pada waktu ke-t

$C_{t,}$: Harga penutupan waktu ke-t

C_{t-1} : Harga penutupan waktu ke-$(t-1)$

ln : Logaritma natural

Transformasi ini menjadikan data lebih mendekati sifat stasioner dan menghilangkan efek skala harga, sehingga perbedaan antarperiode dapat dibandingkan secara proporsional. Dengan kata lain, log return tidak hanya menggambarkan perubahan nominal harga, tetapi juga mencerminkan tingkat pertumbuhan relatif dari nilai saham.

Setelah proses prediksi selesai, hasil log return dapat dikonversi kembali ke bentuk
nakan persamaan eksponensial kumulatif berikut:

$$\hat{C}_{t+h} = C_t \times e^{\sum_{k=1}^{h} r_{t+k}} \tag{2}$$

Keterangan:

r_{t+k} : Hasil prediksi log-return pada horizon ke-k

C_t : Harga aktual pada waktu ke-t (harga awal sebelum prediksi)

h : Jumlah langkah prediksi ke depan (multi-step horizon)

C_{t+h} : Hasil prediksi harga kumulatif setelah h langkah

Persamaan 2 digunakan untuk merekonstruksi harga saham aktual berdasarkan hasil prediksi log-return, sehingga hasilnya dapat diinterpretasikan kembali dalam satuan harga pasar .

1.4.3 Prediksi Multi-Step

Prediksi multi-step merupakan pendekatan dalam peramalan deret waktu (*time series forecasting*) yang bertujuan untuk memperkirakan nilai variabel target pada beberapa periode mendatang secara bersamaan, misalnya t + 1, t + 2, dan t + 3. Pendekatan ini banyak digunakan dalam konteks pasar saham karena perubahan harga sering kali dipengaruhi oleh dinamika jangka pendek maupun menengah yang saling bertaut (Hewamalage et al., 2020). Berbeda dengan *single-step forecasting* yang hanya memperkirakan satu langkah ke depan, *multi-step forecasting* mempertimbangkan ketergantungan antar-periode untuk meminimalkan bias prediksi pada horizon yang lebih panjang.

Secara umum prediksi *multi-step* dapat dituliskan sebagai:

$$\hat{y}_{t+h} = f(y_t, y_{t-1}, \dots, y_{t-p+1}), h = 1,2,3 \quad (3)$$

Keterangan:

$\hat{y}_{t+h}$: Prediksi harga saham pada horizon ke-*h*

y_t : Nilai aktual pada waktu ke-*t*

p : Panjang *lag* yang digunakan sebagai input

h : Panjang horizon prediksi

Dalam konteks penelitian yang dilakukan, fungsi pemetaan $f(y_t, y_{t-1}, \dots, y_{t-p+1})$ eh model machine learning berbasis ensemble (XGBoost, om Forest) yang berfungsi untuk memodelkan hubungan non-toris dan nilai masa depan secara simultan.

ıa pada *multi-step forecasting* adalah akumulasi error. pada horizon awal dapat terbawa ke langkah berikutnya erung meningkat seiring bertambahnya horizon (Hewamalage

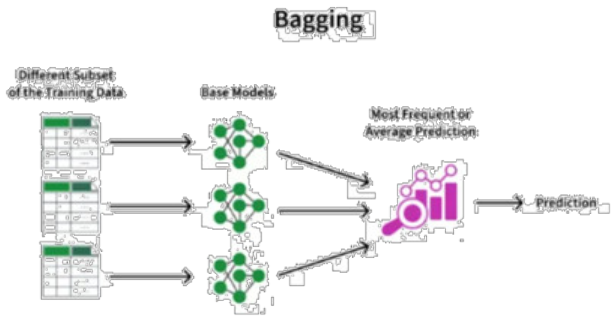
et al., 2020). Oleh karena itu, pemilihan metode yang mampu menjaga stabilitas prediksi pada horizon lebih panjang menjadi krusial. Oleh karena itu, pemilihan metode yang mampu menjaga kestabilan hasil prediksi antar-horizon menjadi penting dalam konteks *multi-step forecasting*.

1.4.4 Ensemble Learning

Ensemble learning adalah pendekatan yang menggabungkan beberapa model dasar (*base learners*) untuk memperoleh hasil prediksi yang lebih akurat dan stabil dibandingkan model tunggal (Liu, 2021). Model ensemble umumnya dibangun melalui dua mekanisme utama, yaitu *bagging dan boosting*, yang masing-masing memanfaatkan varians data dan penguatan bertahap ntuk meningkatkan kinerja prediksi. Selain kedua pendekatan tersebut, terdapat pula teknik *meta-ensemble* seperti *stacking* yang mengombinasikan keluaran dari beberapa model dasar untuk menghasilkan prediksi yang lebih optimal (Wu & He, 2022).

1.4.4.1 *Bagging*

Bagging (*Bootstrap Aggregating*) berfungsi mengurangi variansi dan risiko *overfitting* dengan melatih beberapa model pada subset data pelatihan yang berbeda . Prediksi dari masing-masing model digabung menggunakan rata-rata atau *voting* untuk menghasilkan prediksi akhir yang lebih stabil. Metode ini diimplementasikan melalui algoritma Random Forest, yang membangun banyak pohon keputusan secara paralel untuk memperkuat generalisasi model (Liu, 2021)

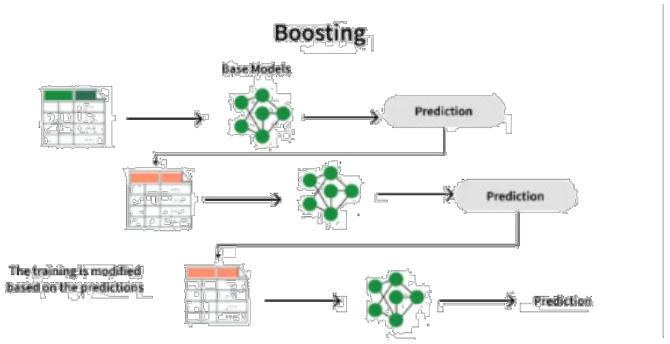


Gambar 1. Ilustrasi Konsep Bagging

Sumber: *GeeksforGeeks, 2025, https://www.geeksforgeeks.org/machine-learning/a-comprehensive-guide-to-ensemble-learning/*

an metode ensemble yang membangun model secara
ode ini, setiap model baru berfokus memperbaiki kesalahan
del sebelumnya. Pendekatan ini cocok untuk mempelajari pola
pleks dan menghasilkan akurasi prediksi yang tinggi (Liu,

2021). Dua algoritma boosting yang digunakan adalah Extreme Gradient Boosting (XGBoost) dan Light Gradient Boosting Machine (LightGBM), keduanya dikenal efisien untuk data besar dan mampu mengatasi masalah non-stasioner.

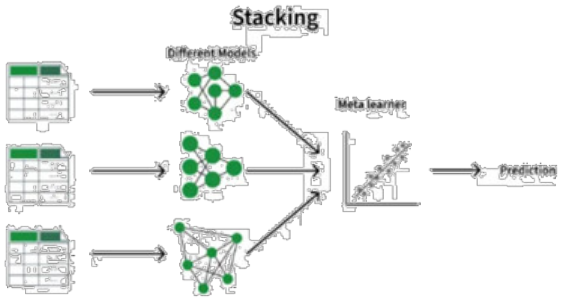


Gambar 2. Ilustrasi Konsep Boosting

Sumber: *GeeksforGeeks, 2025, https://www.geeksforgeeks.org/machine-learning/a-comprehensive-guide-to-ensemble-learning/*

1.4.4.3 *Stacking*

Stacking adalah metode *meta-leaner* yang menggabungkan hasil prediksi dari beberapa model dasar menggunakan model tingkat meta lanjutan (*meta-learner*). Teknik ini bertujuan untuk menemukan kombinasi bobot paling optimal. Dalam berbagai literatur, menunjukkan bahwa stacking dapat diterapkan menggunakan beragam bentuk fungsi pemboboran atau regresi, seperti penggabungan, seperti *Ridge Stacking*, *Lasso Stacking* maupun metode pembobotan lainnya .



Gambar 3. Ilustrasi Konsep Stacking

eks, 2025, https://www.geeksforgeeks.org/machine-learning/a-
-to-ensemble-learning/

1.4.5 Model Dasar

1.4.5.1 XGBoost (Extreme Gradient Boosting)

XGBoost merupakan pengembangan dari metode *Gradient Boosting Decision Tree* (GBDT) yang bekerja secara iteratif dengan menambahkan pohon keputusan (decision tree) baru di setiap langkah untuk memperbaiki kesalahan model sebelumnya (Liu, 2021). Prinsip dasarnya adalah menggabungkan banyak weak learners menjadi satu strong learner yang lebih akurat dan stabil terhadap data kompleks.

Secara matematis, prediksi akhir model XGBoost dapat dituliskan sebagai:

$$\hat{y}_i = \sum_{k=1}^{K} f_k\,(x_i) \tag{4}$$

Keterangan:

$\hat{y}_i$: Hasil prediksi akhir untuk data ke-i

$f_k(x_i)$: Hasil prediksi dari pohon ke-k

K : Jumlah total pohon keputusan dalam ensemble

Rumus ini menunjukkan bahwa model XGBoost dibentuk dari penjumlahan kontribusi setiap pohon keputusan yang dilatih secara berurutan.

Pada proses pelatihannya, XGBoost tidak membangun semua pohon sekaligus, melainkan menambah satu pohon baru di setiap iterasi untuk memperbaiki kesalahan prediksi sebelumnya (Liu, 2021). Hubungan antar iterasi ditunjukkan dengan persamaan:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_k(x_i) \tag{5}$$

Keterangan:

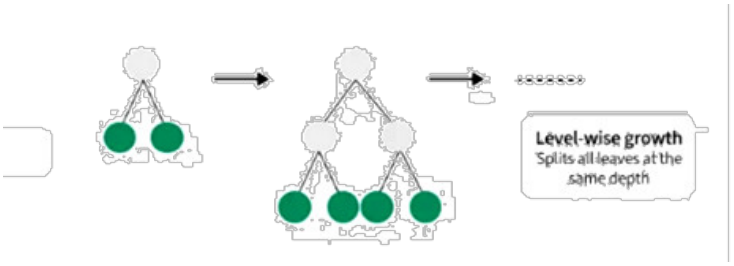
$\hat{y}_i^{(t-1)}$: Hasil prediksi dari iterasi sebelumnya

$f_t(x_i)$: Pohon keputusan ke-k (pohon baru yang dibagun pada interasi ke-t)

laskan bahwa setiap model baru berperan sebagai *corrector* yang tersisa dari iterasi sebelumnya, Proses ini berlangsung iterasi memperbaiki kesalahan model sebelumnya.

GBoost menambah pohon baru dengan cara level-wise, yaitu le pada kedalaman yang sama secara bersamaan. Cara ini

membuat struktur pohon tetap seimbang dan membantu mencegah overfitting, seperti ditunjukkan pada Gambar 4.



Gambar 4. Proses *level-wise growth*

Sumber:GeeksforGeeks,2025,https://www.geeksforgeeks.org/machine-learning/lightgbm-light-gradient-boosting-machine/

Fungsi objektif pada XGBoost terdiri atas dua komponen, yaitu fungsi *loss* dan *regularization term* yang mengatur kompleksitas model:

$$Obj\,(\theta) = \sum_{i-1}^{n} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{6}$$

dengan regularisasi didefinisikan sebagai:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda_x \sum_{j=1}^{T} t_j^2 \tag{7}$$

Keterangan:

$\ell(y_i, \hat{y}_i)$: Fungsi loss

T : Jumlah daun dalam satu pohon

t_j : Bobot daun dalam satu pohon

γ, λ_1 : Parameter regulasi untuk mengontrol kompleksitas model

Parameter γ (gamma) berfungsi sebagai ambang batas minimal agar node dapat
an λ (lambda) berfungsi menahan bobot daun agar tidak

ntukan cabang pohon, XGBoost menghitung information gain
k pemisah (split point) terbaik:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda_x} + \frac{G_R^2}{H_R + \lambda_x} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda_x}\right] - \gamma \quad (8)$$

Keterangan:

G_L, G_R : Jumlah gradient pada node kiri dan kanan

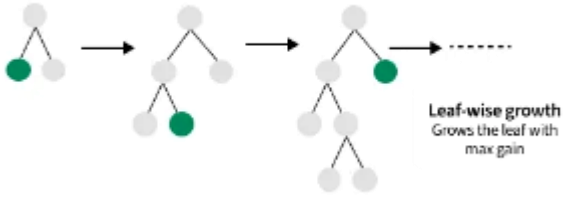
H_L, H_R : Jumlah *Hessian* (turunan kedua loss) pada node kiri dan kanan

Semakin besar nilai Gain, semakin baik split tersebut dalam mengurangi error. Dengan demikian, XGBoost secara otomatis memilih pembagian data yang paling efisien untuk memperbaiki akurasi model (Zhang et al., 2023)

1.4.5.2 LightGBM (*Extreme Gradient Boosting*)

Light Gradient Boosting Machine (LightGBM) merupakan pengembangan dari metode *Gradient Boosting Decision Tree* (GBDT) yang dirancang untuk meningkatkan kecepatan pelatihan dan efisiensi memori tanpa mengorbankan akurasi model *(Id et al., 2025)*. Sama seperti XGBoost, LightGBM bekerja secara iteratif dengan menambahkan pohon keputusan baru untuk memperbaiki kesalahan dari model sebelumnya.

Berbeda dengan XGBoost yang menggunakan pembentukan pohon secara *level-wise*, LightGBM menggunakan pendekatan *leaf-wise*, yaitu memperluas daun yang memiliki nilai gain tertinggi terlebih dahulu. Pendekatan ini membuat model lebih efisien dalam menurunkan error dan mempercepat proses pembelajaran, seperti terlihat pada Gambar 5.



Gambar 5. Proses *leaf-wise*

Sumber: Geeks for Geeks,2025,https://www.geeksforgeeks.org/machine-learning/light-gradient-boosting-machine/

wise growth pada LightGBM, di mana pertumbuhan pohon ıgan gain tertinggi untuk meningkatkan akurasi secara efisien. si total model dinyatakan sebagai:

$$\hat{y}_i = \sum_{k=1}^{K} f_k\,(x_i) \tag{9}$$

dengan $f_t(x_i)$ adalah pohon keputusan ke–t yang ditambahkan untuk meminimalkan fungsi kehilangan (*loss function*) sebagai berikut:

$$Obj\,(\theta) = \sum_{i-1}^{n} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{10}$$

dengan:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda_l \sum_{j=1}^{T} t_j^2 \tag{11}$$

dimana γ adalah penalti terhadap jumlah daun dalam satu pohon, sedangkan λ parameter regularisasi (L2) yang digunakan untuk mengontrol besar bobot t agar model tidak overfitting (Setu et al., 2025).

Selain itu, LightGBM menghitung gain menggunakan pendekatan *histogram-based approximation*, bukan data mentah . Rumus *gain* yang digunakan adalah:

$$Gain_{bin} = \frac{1}{2}\left(\frac{G_{bin}^2}{H_{bin} + \lambda_l}\right) \tag{12}$$

dengan :

$Gain_{bin}$: Jumlah gradient (turunan pertama loss) dalam bin fitur

H_{bin} : Jumlah *Hessian* (turunan kedua loss)

λ : Paramater regulasi

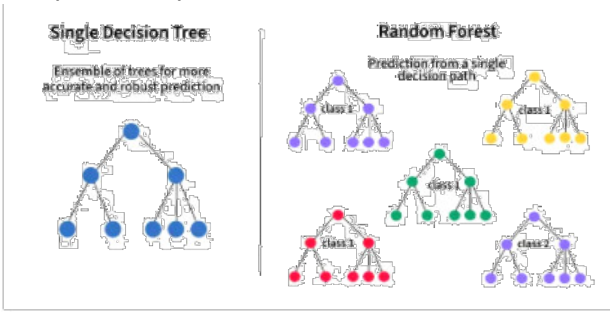
Pendekatan ini membuat LightGBM mampu mencapai hasil yang serupa dengan metode *exact greedy* milik XGBoost, namun dengan waktu komputasi yang jauh lebih cepat karena proses perhitungan dilakukan berdasarkan agregat histogram, bukan data mentah.

est

ah algoritma *ensemble learning* yang menggabungkan banyak *ecision trees*) secara paralel menggunakan teknik *bagging ing*). Tujuan utamanya adalah mengurangi variansi dan

meningkatkan stabilitas model dengan memanfaatkan keberagaman data maupun fitur (Yacoubi & Moussaoui, n.d.) .

Berbeda dengan pohon keputusan tunggal, Random Forest membentuk kumpulan beberapa pohon yang dilatih secara independen, lalu hasilnya digabungkan untuk menghasilkan prediksi yang lebih akurat dan stabil. Perbandingannya dapat dilihat pada Gambar 6.



Gambar 6. *Single Decision Tree* dan *Random Forest*

Sumber:*GeeksforGeeks,2025,https://www.geeksforgeeks.org/machine learning/random-forest-algorithm-in-machine-learning/*

Setiap pohon keputusan dibangun dari subset data pelatihan yang diambil secara acak dengan pengembalian (*bootstrap sampling*). Selain itu, pada setiap percabangan (*split*) algoritma hanya mempertimbangkan sebagian fitur yang dipilih secara acak melalui mekanisme ini disebut *feature randomness*. Kombinasi kedua proses tersebut membuat pohon-pohon yang terbentuk bersifat tidak berkorelasi tinggi satu sama lain, sehingga hasil agregasinya lebih stabil dan general. Secara matematis, hasil prediksi akhir dari Random Forest untuk observasi ke–i dapat dinyatakan sebagai:

$$\hat{y}_i = \frac{1}{K}\sum_{k=1}^{K} f_k(x_i) \tag{13}$$

Keterangan:

$\hat{y}_i$: Hasil prediksi akhir untuk data ke–i,

total pohon keputusan dalam ensemble

ediksi dari pohon ke–k

unjukkan bahwa hasil akhir Random Forest diperoleh dari rata-
pohon untuk regresi, atau *majority voting* untuk klasifikasi.

Dengan demikian, hasil agregasi dari sejumlah pohon keputusan membentuk model yang lebih stabil dan memiliki kemampuan generalisasi yang lebih baik terhadap data baru.

1.4.6. Fusion Ensemble

Pendekatan fusion ensemble merupakan tahap penggabungan hasil prediksi dari beberapa model dasar (*base learners*) untuk memperoleh hasil akhir yang lebih akurat dan stabil. Prinsip utama *ensemble regression* adalah membentuk kombinasi linier dari sejumlah model yang telah dilatih secara terpisah, agar hasil prediksi lebih stabil dan tidak mudah berubah terhadap data baru. Secara umum, formulasi matematisnya dinyatakan sebagai:

$$\hat{y}_i \sum_{j=1}^{M} w_j f_j(x_i) \tag{14}$$

Keterangan:

$\hat{y}_i$	:	Nilai prediksi gabungan untuk observasi ke-i
M	:	Jumlah model dasar yang digunakan dalam ensemble
w_j	:	Bobot model dasar ke-j yang merepresentasikan kontribusinya terhadap hasil akhir
$f_j(x_i)$	:	Hasil prediksi dari model dasar ke-j terhadap observasi ke-i

Persamaan 14 menunjukkan bahwa hasil prediksi akhir diperoleh dengan menjumlahkan hasil prediksi dari beberapa model dasar berdasarkan bobot tertentu. Bobot w_j berfungsi merepresentasikan kontribusi realtif setiap model dasar terhadap hasil akhir. Dalam konteks umum *ensemble regression,* seluruh keluaran model dasar disusun dalam sebuah matriks prediksi $Z \in \mathbb{R}^{N\times M}$, sedangkan bobot *ensemble direpresentasikan* sebagai vektor $w = [w_1, w_2, w_3]^T$ kemudian digunakan untuk menghasilkan estimasi gabungan:

$$\hat{y} = Zw \tag{15}$$

umum dan mendasari berbagai metode pebbotan dalam menggunakan rata-rata sederhana, regularisasi, pembatasan ekatan metaheuristik (Zhou, 2012; Moreira et al., 2012).
rediksi multi-step, rumus ini diterapkan secara terpisah untuk si, sehingga matriks prediksi dan vektor bobot bersifat spesifik

terhadap horizon h. Dengan demikian, estimasi gabungan pada horizon ke-h diperoleh dari matriks prediksi $Z_h$dan bobot w_h yang bersesuaian .

1.4.6.1 Proses Pembangunan Model Hybrid Fusion Ensemble

1.4.6.1.1 Penggabungan Hasil Prediksi dalam Peramalan

Penggabungan hasil prediksi (*forecast combination*) merupakan pendekatan yang digunakan untuk meningkatkan akurasi dan kestabilan prediksi dengan mengombinasikan keluaran dari beberapa model prediksi yang dilatih secara independen. Pendekatan ini didasarkan pada fakta bahwa model cenderung menghasilkan pola kesalahan yang berbeda. (Bates & Clive, 1969)

Dalam kerangka *ensemble*, prediksi akhir diperoleh melalui fungsi penggabung (*fusion function*) yang mengombinasikan keluaran dari beberapa model dasar (base learners).

Pendekatan *hybrid fusion ensemble* juga sejalan dengan konsep *stacked generalization* (*stacking*), yaitu pembelajaran tingkat-dua (*meta-learner*) yang menggunakan prediksi model tingkat-satu sebagai masukan untuk meminimalkan kesalahan generalisasi. (Wolpert, 1992)

1.4.6.1.2 *Stacking* dan Prediksi pada Data Runtun Waktu

Pada stacking, *meta-learner* tidak disarankan dilatih menggunakan prediksi model dasar yang dibentuk dari data pelatihan yang sama, karena dapat menimbulkan optimisme berlebih akibat kebocoran informasi. (Wolpert, 1992)

Oleh karena itu, prediksi yang digunakan sebagai masukan fusion umumnya dibentuk menggunakan prediksi validasi silang (*out-of-fold predictions*) atau prediksi yang dihasilkan oleh model dasar yang tidak digunakan saat proses pelatihan model tersebut agar mencerminkan performa model pada data yang tidak digunakan saat pelatihan. (Wolpert, 1992)

Pada data runtun waktu, prosedur validasi silang perlu menjaga urutan kornologis untuk menghindari *look-ahead bias*, sehingga skema seperti *time-series split/blocked cross-validation/forward chaining* lebih sesuai dibanding k-fold acak. (Bergmeir, 2015)

Secara notasi, misalkan terdapat *M model dasar* $f_1, f_{2,\dots} f_M$ dan target y_t. Prediksi *out-of-fold* dari model ke-m pada waktu ke-t dinyatakan sebagai $\hat{y}_{t,OOF}^{(m)}$. Seluruh prediksi OOF tersebut kemudian disusun dalam vektor pediksi level-1 sebagai berikut:

$$Z_t = \begin{bmatrix} \hat{y}_{t,OOF}^{(1)} \\ \hat{y}_{t,OOF}^{(2)} \\ \vdots \\ \hat{y}_{t,OOF}^{(M)} \end{bmatrix}$$

berperan sebagai fitur masukan bagi tahap *fusion* atau *meta-*ni merupakan representasi umum dari *stacking,* di mana

keluaran model dasar diperlakukan sebagai variabel penjelasan pada tingkat meta (Wolpert, 1992)

1.4.6.1.3 Prediksi Multi-Step dan Implikasinya terhadap *Fusion Ensemble*

Dalam prediksi multi-step horizon $h = 1, \dots, H,$ salah satu strategi yang umum digunakan adalah *direct strategy,* yaitu membangun model terpisah untuk setiap horizon sehingga setiap horizon memiliki fungsi prediksi dan karakter kesalahan yang berbeda. (Taieb & Hyndman, 2012)

Implikasinya, proses *fusion/stacking* dapat dilakukan secara terpisah untuk setiap horizon karena bobot optimal atau *meta-leaner* yang paling sesuai untuk $t + 1$ belum tentu sama untuk $t + 2$ dan $t + 3$. (Taieb & Hyndman, 2012)

1.4.6.1.4 Tahap *Fusion*

Misalkan untuk horizon terntentu diperoleh matriks prediksi level-1:

$$Z_h = \begin{bmatrix} \hat{y}_{1,h}^{(1)} & \hat{y}_{1,h}^{(2)} & \cdots & \hat{y}_{1,h}^{(M)} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{N,h}^{(1)} & \hat{y}_{N,h}^{(2)} & \cdots & \hat{y}_{N,h}^{(M)} \end{bmatrix}$$

Dengan $\hat{y}_{t,h}^{(m)}$ merupakan prediksi model ke-m untuk horizon $t + h$. Fusion bertujuan membentuk $\hat{y}_{t,h}^{(fusion)}$ dari Z_h.

1.4.6.2 *Equal Weighted Fusion*

Metode ini menggunakan pembobotan rata-rata sederhana di mana setiap model dasar dianggap memiliki kontribusi yang sama terhadap hasil akhir. Pendekatan ini umum digunakan sebagai baseline (Rokach, 2010) dan diformulasikan sebagai:

$$\hat{y}_i = \frac{1}{M} \sum_{j=1}^{M} f_j(x_i) \tag{16}$$

Metode *equal weighted* memiliki keunggulan pada kesederhanaan dan ketahanan terhadap overfitting, meskipun tidak mempertimbangkan perbedaan akurasi antar-model.

1.4.6.3 *Ridge Stacking*

akan regresi *Ridge* untuk menentukan bobot optimal dengan
ı hasil prediksi *ensemble* terhadap target sebenarnya dan
ti terhadap bobot besar agar model tidak mudah overfitting.
70). Fungsi objektifnya dituliskan sebagai:

$$\min_{\mathrm{w}} \|y - Zw\|_2^2 + \lambda_{\mathrm{r}} \|w\|_2^2, \tag{17}$$

dengan parameter regularisasi $\lambda_2 > 0$ yang dikalibrasi menggunakan *cross-validation*. Solusi tertutupnya diberikan oleh:

$$w^* = (Z^T Z + \lambda_{\mathrm{r}} I)^{-1} Z^T y \tag{18}$$

1.4.6.4 *Constrained Least Squares*

Pendekatan *Constrained Least Squares* membatasi bobot agar bernilai non-negatif dan jumlah totalnya sama dengan 1 (Ahrens et al., 2023) . Tujuannya memastikan kontribusi setiap model bersifat proporsional dan tidak negatif sehingga hasil ensemble tetap stabil. Formulasi optimasinya ialah:

$$\min_{\mathrm{w}} \|y - Zw\|_2^2, \mathrm{s.t}\, w_j \geq 0, \sum_{j=1}^{M} w_j = 1 \tag{19}$$

Pendekatan ini menjaga agar bobot tidak ekstrem dan tetap berada dalam rentang yang stabil.

1.4.6.5 *Grey Wolf Optimizer*

Algoritma *Grey Wolf Optimizer* diperkenalkan oleh (Mirjalili et al., 2014) sebagai metode meta-heuristik yang meniru perilaku sosial serigala abu-abu dalam berburu. Dalam konteks *ensemble*, Algoritma Grey Wolf (GWO) digunakan sebagai optimizer untuk menentukan bobot w yang meminimalkan fungsi kesalahan kuadrat. Fungsi objektifnya ialah:

$$\min_{\mathrm{w}} J(w) = \|y - Zw\|_2^2 \tag{20}$$

Tiap serigala pada populasi merepresentasikan kandidat bobot w_i, sementara tiga serigala terbaik alpha, beta, dan delta menjadi acuan pembaruan posisi bobot lainnya melalui persamaan:

$$\frac{1}{3}\left[(w_\alpha - A_1|C_1 w_\alpha - w_i|) + \left(w_\beta - A_2|C_2 w_\beta - w_i|\right) + (w_\delta - A_3|C_3 w_\delta - w_i|)\right] \tag{21}$$

- $a, C_k = 2r_{2k}$ dan $\alpha = 2\left(1 - \frac{g}{g_{max}}\right)$. Nilai r_{1k} dan r_{2k} adalah ang menjaga keseimbangan antara eksplorasi dan eksploitasi.

Setelah iterasi selesai, bobot optimal w^* dari serigfala alpha digunakan sebagai bobot akhir pada model *fusion ensembel.*

1.4.6.6 *Lasso Fusion*

Metode ini menggunakan regresi *Lasso* dengan penalti L1 untuk menghasilkan bobot yang bersifat *sparse*, yaitu sebagian bobot dapat menjadi nol sehingga hanya model dengan kontribusi paling signifikan yang dipertahankan (Tibshirani, 1996). Fungsi objektifnya ditulis sebagai:

$$\min_{\mathrm{w}} \|y - Zw\|_2^2 + \lambda_{\mathrm{L}} \|\mathrm{w}\|_1 \tag{22}$$

Pendekatan ini efektif dalam melakukan seleksi model otomatis serta menghindari multikolinearitas antar-prediksi dasar.

1.4.7 Evaluasi Model Prediksi

Evaluasi model prediksi bertujuan untuk menilai sejauh mana hasil peramalan mendekati nilai aktual, baik dari segi akurasi angka maupun arah pergerakan harga. Dalam konteks penelitian yang dilakukan, evaluasi dilakukan terhadap hasil prediksi multi-step harga saham yang dihasilkan oleh model ensemble (XGBoost, LightGBM, dan Random Forest) beserta lima skema penggabungannya. Beberapa ukuran yang digunakan mencakup *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE), *Symmetric MAPE* (sMAPE), *Directional Accuracy* (DA), dan *Theil's U*. Keenam metrik ini dipilih karena mampu menggambarkan performa model dari sisi numerik, stabilitas, hingga efisiensi ekonomis.

1. RMSE (*Root Mean Square Error*)

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_i - y_i)^2} \tag{23}$$

RMSE mengukur seberapa jauh rata-rata hasil prediksi menyimpang dari nilai aktual dengan memberikan penalti lebih besar pada kesalahan yang ekstrem. Secara matematis, RMSE merupakan akar dari rata-rata kuadrat kesalahan ($e_i =$

lebih kecil menunjukkan prediksi model semakin mendekati
RMSE yang lebih kecil menunjukkan prediksi model lebih
ıal (Chai & Draxler, 2014).

)lute Error)

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|\hat{y}_i - y_i| \tag{24}$$

MAE mengukur rata-rata besarnya kesalahan tanpa memperhatikan arah (positif atau negatif). Metrik ini memberikan gambaran sederhana dan intuitif tentang seberapa besar perbedaan antara nilai prediksi dan nilai aktual, sehingga cocok digunakan saat membandingkan performa antar-model secara umum (Willmott & Matsuura, 2005) .

3. MAPE (Mean Absolute Percentage Error)

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{25}$$

MAPE digunakan untuk menilai kesalahan relatif dalam bentuk persentase. Nilai MAPE yang kecil menandakan model mampu memberikan prediksi dengan proporsi kesalahan yang rendah terhadap nilai sebenarnya. Meskipun mudah diinterpretasikan, metrik ini cenderung bias jika data memiliki nilai aktual yang mendekati nol .

4. sMAPE (Symmetric MAPE)

$$sMAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{|\hat{y}_i| + |y_i|/2}\right| \tag{26}$$

sMAPE merupakan versi simetris dari MAPE yang menormalkan kesalahan terhadap rata-rata nilai aktual dan prediksi. Pendekatan ini mencegah pembagian dengan nol serta memberikan hasil yang lebih stabil pada data keuangan yang fluktuatif (Makridakis, 1993).

5. Directional Accuracy (DA)

$$DA = \frac{1}{n}\sum_{i=2}^{n} I[(\hat{y}_i - y_{i-1})(y_i - y_{i-1}) > 0] \times 100\% \tag{27}$$

digunakan untuk menilai seberapa sering model berhasil
rubahan harga (naik atau turun) dengan benar. Meskipun tidak
alahan, DA penting dalam konteks investasi karena membantu

menilai kemampuan model dalam menangkap tren pergerakan pasar (Thompson, 2020).

6. Theil's U Statistic

$$U = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} {y_i}^2} + \sqrt{\frac{1}{n}\sum_{i=1}^{n} {\hat{y}_i}^2}} \tag{28}$$

Theil's U digunakan untuk mengukur efisiensi relatif suatu model dibandingkan dengan pendekatan naïf,yaitu prediksi harga sama dengan harga sebelumnya. Nilai U < 1 menunjukkan bahwa model prediksi lebih efisien daripada metode naïf, sedangkan U > 1 berarti model kurang efektif (Theil, 1966).

1.4.8 Analisis Korelasi Antar-Model

Korelasi digunakan untuk mengukur kekuatan dan arah hubungan linear antara dua variabel atau hasil prediksi model. Dalam konteks yang dilakukan, analisis korelasi dilakukan untuk menilai sejauh mana prediksi XGBoost, LightGBM, dan Random Forest saling berhubungan. Nilai korelasi yang tidak terlalu tinggi mengindikasikan adanya keragaman (*diversity*) di antara model dasar. *Ensemble* yang baik terbentuk ketika model-model penyusunnya memiliki pola prediksi yang berbeda, karena keragaman tersebut menurunkan tingkat kesalahan yang saling berkorelasi dan pada akhirnya meningkatkan akurasi hasil gabungan (Wood et al., 2023)

Hubungan antar-dua model dapat dihitung menggunakan koefisien korelasi Pearson sebagai berikut (James et al., 2021) :

$$r_{jk} = \frac{\sum_{t=1}^{T}\left(\hat{y}_t^{(j)} - \bar{y}^{(j)}\right)\left(\hat{y}_t^{(k)} - \bar{y}^{(k)}\right)}{\sqrt{\sum_{t=1}^{T}\left(\hat{y}_t^{(j)} - \bar{y}^{(j)}\right)^2}\sqrt{\sum_{t=1}^{T}\left(\hat{y}_t^{(k)} - \bar{y}^{(k)}\right)^2}} \tag{29}$$

dengan x_t dan y_t merupakan hasil prediksi dua model pada waktu ke-t, serta $\bar{x}$ dan $\bar{y}$ adalah nilai rata-ratanya. Koefisien r_{xy} bernilai antara −1 hingga +1. Nilai positif enandakan hubungan linear positif yang kuat, sedangkan nilai ukkan hubungan yang lemah. Dalam *ensemble learning*, nilai at atau rendah di antara model dasar menandakan adanya sehingga setiap model mampu memberikan kontribusi unik iksi gabungan memberikan kontribusi unik terhadap hasil Wood et al., 2023).

Keterangan:

r_{xy} : Koefisien korelasi Pearson antara dua model

x_t, y_t : Hasil prediksi dua model pada waktu ke-t

$\bar{x}, \bar{y}$: Rata-rata hasil prediksi masing-masing model

1.4.9 Uji Signifikansi Model

Dalam proses evaluasi model prediksi, selain menilai tingkat kesalahan menggunakan metrik seperti RMSE, MAE, MAPE, dan Theil's U, digunakan pula pengujian statistik untuk memastikan apakah perbedaan akurasi antar-model bersifat signifikan secara statistik. Salah sa tu metode yang banyak digunakan dalam literatur modern adalah Uji Diebold–Mariano (DM).

Secara matematis, selisih tingkat kesalahan antar-model dinotasikan sebagai *differential loss* (Zaiontz, 2025):

$$d_t = L(e_{1,t}) - L(e_{2,t}) \tag{30}$$

dengan $L(e_{1,t})$ merupakan fungsi kerugian model ke – i pada waktu t. Nilai rata-ratanya ditulis sebagai:

$$\bar{d} = \frac{1}{T}\sum_{t=1}^{T} d_t \tag{31}$$

Statistik ujinya dirumuskan sebagai:

$$DM = \frac{\bar{d}}{\sqrt{\widehat{Var}(\bar{d})}} \tag{32}$$

Dalam teori DM-test estimasi varians $\widehat{Var}(\bar{d})$ biasanya dihitung melalui pendekatan Heteroskedasticity and Autocorrelation Consistent (HAC) karena deret d_t pada data time series berpotensi mengandung autokorelasi dan heteroskedastisitas. Pada itian ini, pendekatan tersebut disederhanakan dengan ngan baku sampel dari deret d_t dibagi akar jumlah observasi. mum digunakan pada horizon prediksi pendek dan efisiensi komputasi. Fungsi kerugian $L(e_{1,t})$ direalisasikan at, yaitu $L(e_{1,t}) = \left(y_t - \hat{y}_t^{(i)}\right)^2$. Sehingga selisih kerugian d_t ariano merepresentasikan perbedaan antara dua nilaisquare

error pada waktu yang sama. Metode ini telah diterapkan secars luas dan terus dikembangkan untuk pengujian *forecast accuracy* di bidang ekonomi dan keuangan (Buturac, 2022; Coroneo & Profumo, 2023)
Keterangan:

$L(e_{1,t})$	:	Fungsi kerugian model ke – i
$\bar{d}$	:	Rata-rata selisih kerugian antar model
d_t	:	Selisih kerugian antar model
$\widehat{Var}(\bar{d})$	:	Estimasi varians dengan penyesuaian HAC
DM	:	Nilai statistik uji Diebold–Mariano
H_0	:	Tidak ada perbedaan akurasi antar-model.

BAB II
METODOLOGI PENELITIAN

2.1 Pendekatan dan Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian kuantitatif dengan pendekatan eksperimen komputasional. Pendekatan ini digunakan untuk menganalisis data harga saham dan menguji kinerja model prediksi berbasis *machine learning*. Penelitian yang dilakukan termasuk dalam kategori penelitian terapan (*applied research*) karena berfokus pada penerapan model Hybrid Multi-Model Fusion Ensemble untuk meningkatkan kemampuan prediksi harga saham.

Data yang digunakan merupakan data sekunder berupa harga saham harian PT Aneka Tambang Tbk (ANTM) dan PT Adaro Energy Indonesia Tbk (ADRO) periode 2020–2024, yang diambil dari situs *Yahoo Finance* melalui pustaka *yfinance* pada Python. Atribut yang digunakan meliputi Open, High, Low, Close, dan Volume. Kedua saham ini dipilih karena mewakili sektor pertambangan dengan tingkat volatilitas yang tinggi.

Pembagian data latih dan data uji pada penelitian menggunakan pendekatan *Time Series Cross-Validation* dengan skema *TimeSeriesSplit* sebanyak 5 fold. Pada setiap fold, bagian awal deret waktu digunakan sebagai data latih, sedangkan segmen waktu berikutnya digunakan sebagai data uji.

Seluruh proses analisis dilakukan di Google Colab menggunakan pustaka *pandas, numpy, scikit-learn, xgboost, lightgbm*, dan *matplotlib*. Tahapan utama mencakup pembersihan data, transformasi log-return, pembuatan fitur teknikal, pelatihan model dasar, penggabungan model, evaluasi, uji statistik, dan visualisasi hasil.

Adapun tahapan analisis data dalam penelitian yang dilakukan adalah sebagai berikut:

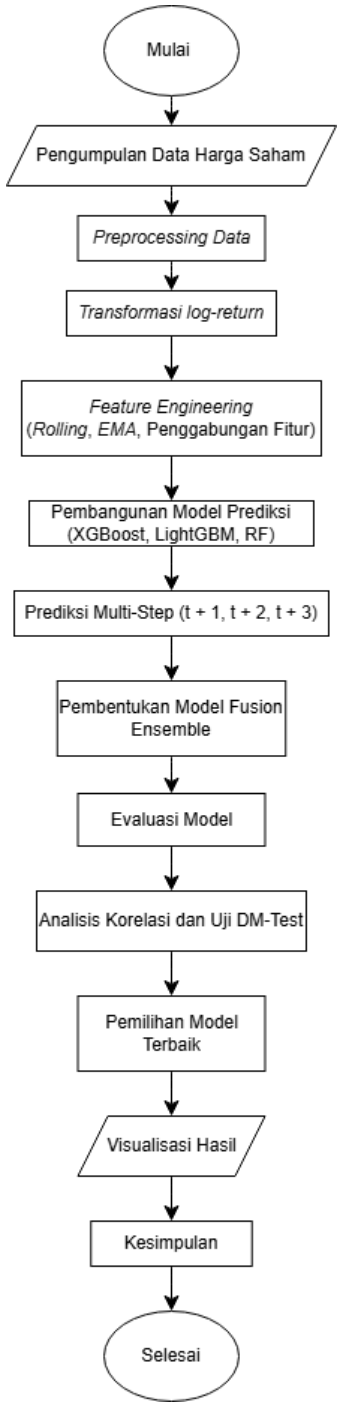
1. Mengumpulkan data historis saham ANTM dan ADRO, Data diambil dari *Yahoo Finance* melalui pustaka *yfinance* dengan periode 2020 – 2024
2. Melakukan praproses data dengan memeriksa kelengkapan dataset dan membersihkan nilai yang hilang (missing values).
3. Melakukan transformasi log-return untuk menstabilkan variansi data dan mengurangi efek skala harga.
4. Membangun fitur turunan (feature engineering), meliputi perhitungan rolling mean, rolling standard deviation, dan exponential moving average (EMA), kemudian menggabungkannya ke dalam satu dataframe pelatihan.

del dasar (base learners), melatih tiga model utama – BM, dan Random Forest – untuk masing-masing horizon (t+1,

el *Hybrid Multi-Model Fusion Ensemble,* hasil prediksi dari tiga bungkan melalui lima skema penggabungan (F0–F4)

7. Menghitung nilai evaluasi model, menghitung metrik evaluasi: RMSE, MAE, MAPE, sMAPE, DA, dan Theil's U.
8. Menganalisis korelasi antar model dasar, menghitung korelasi antara XGBoost, LightGBM, dan Random Forest untuk menilai hubungan antarhasil prediksi.
9. Melakukan uji statistik Diebold – Mariano (DM Test), Menguji perbedaan kinerja antara model fusion (F1–F4) dengan baseline (F0). Nilai t-statistik negatif dan signifikan menunjukkan bahwa model fusion menghasilkan kesalahan prediksi yang lebih rendah dibanding model dasar.
10. Menentukan model terbaik, berdasarkan nilai RMSE terendah yang didukung oleh konsistensi metrik lain serta pola prediksi yang paling mendekati data aktual.
11. Membuat visualisasi hasil prediksi, berupa grafik perbandingan antara harga aktual dan prediksi untuk setiap model fusion, serta visualisasi model terbaik.

2.2 Alur Kerja



Gambar 7. Alur Kerja