

PENERAPAN ALGORITMA RANDOM FOREST REGRESSION DALAM PREDIKSI HARGA SAHAM BBRI

Kevin Winardi¹, Yulianus Febry Tri Nugroho², Johannes³, Hendrik Fery Herdiatmoko⁴

¹⁻⁴Program Studi Informatika, Universitas Katolik Musi Charitas

Jalan Bangau No. 60, Kecamatan Ilir Timur II, Palembang.

Email: ¹kevinwinardi192005@gmail.com, ²frytn13@gmail.com, ³johanneswork2024@gmail.com, ⁴hendrik@ukmc.ac.id

ABSTRAK

Pergerakan harga saham dipengaruhi oleh berbagai faktor dan bersifat fluktuatif, sehingga diperlukan metode prediksi yang mampu menangkap pola data yang kompleks. Penelitian ini bertujuan untuk memprediksi harga saham menggunakan metode *Random Forest Regression*. Data yang digunakan dibagi menjadi data pelatihan dan data pengujian untuk mengevaluasi kinerja model. Kinerja model dievaluasi menggunakan beberapa metrik, yaitu *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), dan koefisien determinasi (R^2). Hasil penelitian menunjukkan bahwa model *Random Forest Regression* setelah optimasi menghasilkan nilai MAE sebesar 64,02, RMSE sebesar 84,51, dan R^2 sebesar 0,8484. Nilai-nilai tersebut mengindikasikan bahwa model memiliki tingkat kesalahan prediksi yang rendah dan mampu menjelaskan 84,84% variasi pada data harga saham. Berdasarkan hasil tersebut, dapat disimpulkan bahwa Random Forest memiliki kinerja yang baik dan cukup andal dalam memprediksi harga saham.

Kata Kunci: Harga Saham, *Machine Learning*, Random Forest Regression, *Data Mining*

ABSTRACT

Stock price movements are influenced by various factors and exhibit high volatility, making accurate prediction a challenging task. This study aims to predict stock prices using the Random Forest Regression method. The dataset is divided into training and testing sets to evaluate the model's performance. The performance of the model is assessed using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). The results show that the Random Forest Regression model after optimization achieves an MAE of 64,02, an RMSE of 84,51, and an R^2 value of 0.8484. These results indicate a low prediction error and demonstrate that the model is able to explain 84.84% of the variance in stock price data. Therefore, it can be concluded that Random Forest is an effective and reliable method for stock price prediction.

Key Words: Stock Price, Machine Learning, Random Forest Regression, Data Mining

1. PENDAHULUAN

Pergerakan pertumbuhan ekonomi pada setiap negara pada umumnya berbeda-beda [1]. Pasar modal menjadi salah satu indikator utama ekspansi ekonomi suatu negara dan saham adalah contoh produk keuangan yang paling banyak digunakan karena potensi keuntungannya yang tinggi [2][3]. Bagi investor pemula tentunya ingin mendapatkan keuntungan semaksimal mungkin dengan resiko yang kecil [4]. Namun, kebanyakan investor pemula cenderung mengambil keputusan berdasarkan saran dari orang lain, dibandingkan analisis mendalam. Dalam hal analisis saham, perlu memperhatikan fluktuasi harga saham dimana dipengaruhi oleh berbagai faktor, seperti *Current Ratio* dan *Return on Asset* [5].

Untuk membantu prediksi harga saham dengan akurat, maka perlu didorong adanya adopsi pendekatan *machine learning* untuk [6]. Salah satu algoritma yang banyak diaplikasikan dalam prediksi data deret waktu (*time series*) adalah *Random Forest Regression*, yang dikenal efektif dalam menangani data berdimensi tinggi, mengurangi risiko overfitting, serta memberikan performa prediksi yang stabil [7].

Pada penelitian [8] yang membahas mengenai prediksi harga 3 saham, yaitu PT Indofood Sukses Makmur Tbk, PT Telekomunikasi Indonesia Tbk, dan PT Kimia Farma Tbk. Adapun untuk algoritma yang digunakan seperti *Random Forest Regression* memiliki nilai *Root Mean Squared Error* (RMSE) cukup kecil yaitu 0.012 dan tertinggi sebesar 0.018. Hal ini menunjukkan kalau algoritma *Random Forest Regression* yang diterapkan sudah memiliki tingkat *error* yang cukup rendah, hanya sekitar 1,2% dari rentang data. Hal ini juga dibuktikan dari perbandingan algoritma yang dilakukan oleh peneliti [9], didapat kalau algoritma Random Forest memiliki akurasi yang lebih baik dibandingkan algoritma *Random Forest Regression* dan *Support Vector Regression* (SVR) dalam hal prediksi harga saham. Tidak hanya itu, algoritma *Random Forest Regression* juga dinilai lebih baik dibandingkan dengan algoritma prediksi lainnya dalam beberapa penelitian berbeda [1][10][11][12]. Berdasarkan

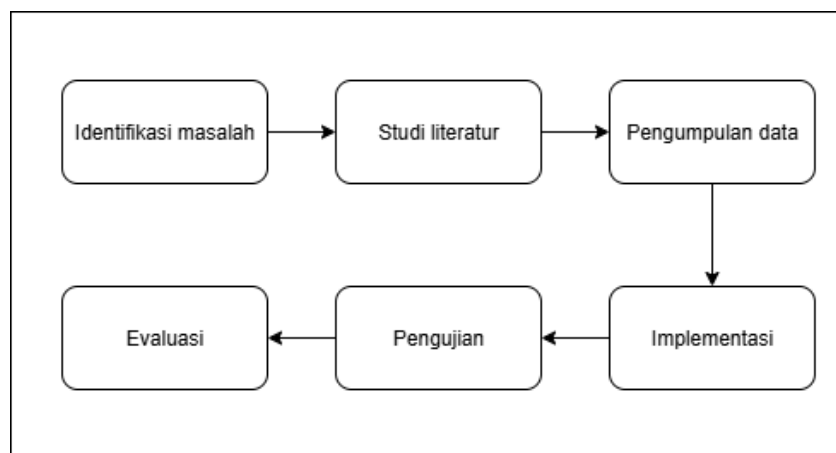


penelitian [13], didapat bahwa penggunaan algoritma *Random Forest* dengan optimasi parameter *Grid Search*, mampu meningkatkan hasil prediksi. Hal ini menjadi bukti kalau penggunaan algoritma *Random Forest Regression* untuk data yang sifatnya *time series* seperti saham sangat cocok digunakan dengan dikombinasi optimasi parameter lanjutan.

Berdasarkan penelitian [14], prediksi harga saham dilakukan menggunakan metode statistik seperti Linear Regression dan Bayesian Regression pada saham PT Bank Rakyat Indonesia (Persero) Tbk (BBRI). Saham PT Bank Rakyat Indonesia (Persero) Tbk (BBRI) merupakan salah satu saham perbankan dengan tingkat fluktuasi yang tinggi dan memiliki data historis jangka panjang sejak tahun 2003 [15], sehingga membutuhkan model prediksi yang mampu menangani kompleksitas tersebut secara lebih optimal. Oleh karena itu, penelitian ini bertujuan untuk menerapkan *Random Forest Regression* yang mampu menangani data kompleks serta melakukan optimasi *hyperparameter* menggunakan *Grid Search* agar meningkatkan akurasi prediksi harga saham BBRI.

2. METODE PENELITIAN

Penelitian ini akan dilakukan dengan beberapa tahapan terdiri dari identifikasi masalah, studi literatur, pengumpulan data, implementasi, pengujian, dan evaluasi. Adapun untuk mempermudah disajikan urutan penelitian pada Gambar 1.



Gambar 1. Alur Penelitian

Identifikasi Masalah

Pada tahap awal, dilakukan identifikasi masalah untuk merumuskan masalah yang ingin diselesaikan pada penelitian. Berlandaskan pada bagian Pendahuluan sebelumnya, didapat masalah yang perlu diteliti yaitu prediksi harga saham. Dimana salah satu saham yang digunakan sebagai objek penelitian adalah saham BBRI.

Studi Literatur

Studi literatur menjadi pondasi untuk membangun penelitian yang berbobot. Pada tahap ini dilakukan pengumpulan referensi jurnal ilmiah yang berkaitan dengan prediksi harga saham dan penggunaan algoritma *Random Forest*.

1. Saham

Saham merupakan indikator kesehatan perusahaan yang berbentuk surat berharga [6]. Harga saham bersifat fluktuatif karena dipengaruhi oleh berbagai faktor, seperti kinerja perusahaan, kondisi ekonomi makro, sentimen pasar, serta faktor eksternal lainnya.

2. *Random Forest Regression*

Random Forest Regression sendiri membangun banyak pohon regresi lalu menghitung nilai rata-rata hasil prediksi peubah respon dari semua pohon regresi tersebut [8]. Metode ini mampu menangani hubungan non-linier dan data berdimensi tinggi dengan baik, serta relatif tahan terhadap *overfitting*. Dalam konteks prediksi harga saham, *Random Forest* efektif digunakan karena mampu menangkap pola kompleks dalam data historis dan menghasilkan prediksi yang lebih *robust* dibandingkan metode tunggal.

3. *Grid Search*

Grid Search merupakan proses menentukan nilai parameter terbaik dalam subproses operator [13]. Metode ini bekerja dengan mencari kombinasi parameter terbaik pada suatu model dan menguji kombinasi nilai parameter yang telah ditentukan.

Pengumpulan Data



Data yang digunakan dalam penelitian ini merupakan data sekunder berupa harga saham historis yang diperoleh secara daring (dalam jaringan) menggunakan *library* Python *yfinance*. Pustaka ini menyediakan akses ke data pasar keuangan yang bersumber dari Yahoo Finance dan banyak digunakan dalam penelitian akademik maupun praktis untuk analisis pasar saham. Data diambil selama 6 tahun dengan rentang periode harian dari tanggal 1 Januari 2020 hingga 1 Januari 2026.

Implementasi

Tahap implementasi dilakukan dengan membangun dan menerapkan algoritma *Random Forest Regression* menggunakan platform Google Colab. Pada tahap ini, model dilatih menggunakan data latih yang telah melalui proses *pre-processing* dan *feature engineering*. Implementasi dilakukan dengan memanfaatkan bahasa pemrograman Python dan *library machine learning* yang relevan.

Pengujian

Setelah proses pelatihan model selesai, dilakukan tahap pengujian dengan memberikan data testing yang belum pernah digunakan pada proses pelatihan. Tahap pengujian bertujuan untuk memperoleh hasil prediksi harga saham di hari berikutnya.

Evaluasi

Tahap evaluasi dilakukan untuk menilai kinerja model dalam memprediksi harga saham. Evaluasi dilakukan dengan membandingkan hasil prediksi model terhadap data aktual pada data testing. Kinerja model diukur menggunakan beberapa metrik evaluasi, yaitu *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan koefisien determinasi (R^2).

MAE mengukur rata-rata besar kesalahan prediksi tanpa memperhatikan arah kesalahan. Semakin kecil MAE, maka model semakin akurat.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \dots (1)$$

RMSE memberikan penalti lebih besar pada kesalahan besar (*outlier*). Rumusnya mirip seperti MAE, hanya saja perlu diakar kuadratkan.

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

R^2 menunjukkan proporsi variasi data yang dapat dijelaskan oleh model. Ada 3 interpretasi yang digunakan, seperti yang disajikan di Tabel 1.

Tabel 1. Interpretasi Nilai R^2

Nilai R^2	Interpretasi
1	Prediksi sempurna
0	Sama dengan model rata-rata
< 0	Lebih buruk dari model rata-rata

3. HASIL DAN PEMBAHASAN

Pada tahap ini menyajikan hasil implementasi algoritma *Random Forest Regression* untuk prediksi harga saham BBRI.

Persiapan Data

Dataset dikumpulkan dan disimpan sebagai variabel *df*. Hasil dari *yf.download* berupa *dataframe pandas* sehingga bisa langsung digunakan tanpa harus konversi. Kemudian dilakukan *df.reset_index()* untuk menghapus posisi indeks awal kolom *Date*.

```
import yfinance as yf

df = yf.download("BBRI.JK", start="2020-01-01", end="2026-01-01", interval="1d")
df = df.reset_index()
df.head()

/tmp/ipython-input-1198744173.py:3: FutureWarning: YF.download() has changed argument auto_adjust default to True
df = yf.download("BBRI.JK", start="2020-01-01", end="2026-01-01", interval="1d")
[*****100%*****] 1 of 1 completed
```

Price	Date	Close	High	Low	Open	Volume
Ticker	BBRI.JK	BBRI.JK	BBRI.JK	BBRI.JK	BBRI.JK	BBRI.JK
0	2020-01-02	2853.532227	2853.532227	2821.179185	2847.061583	45886302
1	2020-01-03	2860.002930	2872.944217	2840.591173	2860.002930	91189705
2	2020-01-06	2827.650146	2840.591260	2795.297101	2821.179503	48648450
3	2020-01-07	2847.061768	2853.532411	2834.120654	2853.532411	114344885
4	2020-01-08	2834.120605	2847.061718	2808.238206	2834.120605	188929583

Gambar 2. Preview Dataset di Google Colab

Untuk mengetahui gambaran awal data mentah yang diperoleh dari Yahoo Finance, dilakukan pengecekan informasi *dataset* menggunakan perintah `df.info()`. Dari Gambar 3, bisa didapat informasi *dataset* yang diperoleh berjumlah 1.449 baris dengan semua kolom sama terisi tanpa adanya nilai kosong.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1449 entries, 0 to 1448
Data columns (total 6 columns):
#   Column              Non-Null Count  Dtype
---  ---             
0   (Date, )            1449 non-null   datetime64[ns]
1   (Close, BBRI.JK)   1449 non-null   float64
2   (High, BBRI.JK)   1449 non-null   float64
3   (Low, BBRI.JK)    1449 non-null   float64
4   (Open, BBRI.JK)   1449 non-null   float64
5   (Volume, BBRI.JK) 1449 non-null   int64
dtypes: datetime64[ns](1), float64(4), int64(1)
memory usage: 68.1 KB
```

Gambar 3. Informasi Dataset

Pre-processing Data

Untuk memudahkan pemanggilan nama kolom, maka dilakukan proses mengubah nama kolom yang sebelumnya *Multi Index* menjadi *Single Index* pada *dataframe*. Kemudian kolom harga penutupan yaitu *Close* diganti menjadi *Price*.

```
df.columns = ['Date', 'Price', 'High', 'Low', 'Open', 'Volume']
df.head()
```

	Date	Price	High	Low	Open	Volume
0	2020-01-02	2853.532471	2853.532471	2821.179426	2847.061827	45886302
1	2020-01-03	2860.002930	2872.944217	2840.591173	2860.002930	91189705
2	2020-01-06	2827.649902	2840.591015	2795.296860	2821.179259	48648450
3	2020-01-07	2847.061768	2853.532411	2834.120654	2853.532411	114344885
4	2020-01-08	2834.120605	2847.061718	2808.238206	2834.120605	188929583

Gambar 4. Nama Kolom Baru DataFrame

Untuk memperkaya fitur dan membantu model dalam mempelajari pola informasi lebih dalam terkait harga saham, maka dilakukan *feature engineering*. Pada tahap ini dibuat beberapa kolom baru dengan beberapa pendekatan. Adapun untuk setiap fitur baru yang dibuat dijelaskan dalam bentuk Tabel 2.

Tabel 2. Penerapan Feature Engineering

Fitur	Deskripsi	Rekonstruksi
<i>Return_Id</i>	Persentase perubahan harga dari hari sebelumnya	Rumus:
		$\frac{Price_t - Price_{t-1}}{Price_{t-1}}$
		Python:

Fitur	Deskripsi	Rekonstruksi
		<code>df['Price'].pct_change()</code>
<i>Volume_Change_1d</i>	Persentase perubahan volume perdagangan dari hari sebelumnya	Rumus: $\frac{Volume_{t-1} - Volume_{t-1}}{Volume_{t-1}}$ Python: <code>df['Volume'].pct_change()</code>
<i>HL_Pct</i>	<i>High-Low Spread</i> relative terhadap harga pembukaan, mencerminkan <i>volatilitas intraday</i>	Rumus: $\frac{High - Low}{Open}$ Python: <code>(df['High'] - df['Low']) / df['Open']</code>
<i>CO_Pct</i>	Perubahan harga penutupan relatif terhadap harga pembukaan	Rumus: $\frac{Close - Open}{Open}$ Python: <code>(df['Price'] - df['Open']) / df['Open']</code>
<i>Price_lag_L</i>	Harga saham beberapa hari sebelumnya (<i>lag</i>), menangkap momentum / autokorelasi	<code>df['Price'].shift(L)</code>
<i>Return_lag_L</i>	Return beberapa hari sebelumnya, menangkap tren jangka pendek	<code>df['Return_1d'].shift(L)</code>
<i>Volume_lag_L</i>	Volume beberapa hari sebelumnya, menangkap aktivitas pasar historis	<code>df['Volume'].shift(L)</code>
<i>SMA_w</i>	<i>Simple Moving Average</i> , rata-rata harga selama “w” hari, menangkap tren harga	<code>df['Price'].rolling(w).mean()</code>
<i>Volatility_w</i>	Volatilitas <i>rolling</i> berdasarkan <i>return</i> , ukuran risiko	<code>df['Return_1d'].rolling(w).std()</code>
<i>Day_of_week</i>	Hari dalam minggu, menangkap efek musiman (Senin-Jumat)	<code>df['Date'].dt.dayofweek</code>
<i>Month</i>	Bulan dalam tahun, menangkap pola musiman bulanan	<code>df['Date'].dt.month</code>

Setelah melewati tahap *pre-processing*, data saat ini memiliki banyak fitur seperti yang ditampilkan pada gambar di bawah.

	Date	Price	High	Low	Open	Volume	Return_1d	Volume_Change_1d	HL_Pct	CO_Pct	Price_lag_1	Return_1d
0	2020-01-02	2853.532471	2853.532471	2821.179426	2847.061827	45886302	NaN	NaN	NaN	NaN	NaN	NaN
1	2020-01-03	2860.002930	2872.944217	2840.591173	2860.002930	91189705	0.002268	0.987297	0.011364	0.002273	2853.532471	
2	2020-01-06	2827.649902	2840.591015	2795.296860	2821.179259	48648450	-0.011312	-0.466514	0.011312	0.000000	2860.002930	0.00
3	2020-01-07	2847.061768	2853.532411	2834.120654	2853.532411	114344885	0.006865	1.350432	0.016055	0.002294	2827.649902	-0.01
4	2020-01-08	2834.120605	2847.061718	2808.238206	2834.120605	188929583	-0.004545	0.652278	0.006803	-0.002268	2847.061768	0.00

Gambar 5. *Preview Data* Setelah Pre-Processing

Akibat penggunaan operasi *shift*, *lag*, dan perhitungan *rolling* pada tahap sebelumnya, terdapat sejumlah baris data yang mengandung nilai kosong (*NaN*). Hal ini didasarkan pada baris tersebut tidak memiliki jumlah baris sebelumnya yang mencukupi. Misalnya indeks ke-0 tidak memiliki 1 hari sebelumnya sehingga nilai *Return_1d* didapat nilai kosong, begitupun pada kolom lainnya yang bergantung dengan data historis hari sebelumnya. Oleh

karena itu, seluruh baris yang mengandung nilai *NaN* dihapus agar data yang digunakan dalam pemodelan bersifat lengkap dan konsisten.

Split Data

Pada tahap ini dilakukan persiapan data akhir sebelum proses pelatihan dan pengujian model. Data yang telah melalui tahap *pre-processing* selanjutnya diproses untuk membentuk variabel target serta memisahkan fitur dan target yang akan digunakan dalam pemodelan.

Selanjutnya, dilakukan pemilahan antara variabel fitur (X) dan variabel target (y). Variabel target saat ini dibuat menggunakan nilai harga saham di hari berikutnya ($Price_{t+1}$). Dikarenakan menggunakan data target hari selanjutnya, indeks terakhir pada data akan memiliki nilai *NaN*, sehingga bisa dihapus untuk memastikan model belajar dengan data yang terisi lengkap saja. Variabel fitur akan menggunakan semua kolom kecuali Date dan variabel target (y).

```
def prepare_model_data(df: pd.DataFrame):
    df = df.copy()

    df['Price_next'] = df['Price'].shift(-1)
    df = df.dropna().reset_index(drop=True)

    drop_cols = ['Date', 'Price_next']
    feature_cols = [c for c in df.columns if c not in drop_cols]

    X = df[feature_cols]
    y = df['Price_next']

    return df, X, y, feature_cols

df_model, X, y, feature_cols = prepare_model_data(df_feat)

print("Feature:\n", X.columns)

Feature:
Index(['Price', 'High', 'Low', 'Open', 'Volume', 'Return_1d',
       'Volume_Change_1d', 'HL_Pct', 'CO_Pct', 'Price_lag_1', 'Return_lag_1',
       'Volume_lag_1', 'Price_lag_2', 'Return_lag_2', 'Volume_lag_2',
       'Price_lag_3', 'Return_lag_3', 'Volume_lag_3', 'Price_lag_5',
       'Return_lag_5', 'Volume_lag_5', 'Price_lag_10', 'Return_lag_10',
       'Volume_lag_10', 'SMA_5', 'Volatility_5', 'SMA_10', 'Volatility_10',
       'SMA_20', 'Volatility_20', 'Day_of_week', 'Month'],
      dtype='object')
```

Gambar 6. Persiapan Data

Setelah proses tersebut, diperoleh *dataset* akhir yang terdiri dari data fitur (X) dan *target* (y) yang siap digunakan untuk proses pemodelan. *Dataset* ini kemudian dibagi menjadi data latih dan data uji dengan rasio 80% untuk data latih dan 20% untuk data uji. Pembagian dilakukan secara berurutan sesuai urutan waktu (*time series split*) untuk menjaga konsistensi temporal data dan menghindari kebocoran informasi dari data masa depan ke data masa lalu.

```
def time_series_train_test_split(X, y, train_ratio=0.8):
    n = len(X)
    split_idx = int(n * train_ratio)

    X_train = X.iloc[:split_idx]
    y_train = y.iloc[:split_idx]
    X_test = X.iloc[split_idx:]
    y_test = y.iloc[split_idx:]

    return X_train, X_test, y_train, y_test

X_train, X_test, y_train, y_test = time_series_train_test_split(X, y, train_ratio=0.8)
print(f"Jumlah data pelatihan (fitur): {len(X_train)}")
print(f"Jumlah data pelatihan (target): {len(y_train)}")
print()
print(f"Jumlah data uji (fitur): {len(X_test)}")
print(f"Jumlah data uji (target): {len(y_test)}")

Jumlah data pelatihan (fitur): 1142
Jumlah data pelatihan (target): 1142

Jumlah data uji (fitur): 286
Jumlah data uji (target): 286
```

Gambar 7. Split Data

Modeling

Pada tahap *modeling* dibuat dua model *Random Forest Regression* yaitu model pertama tanpa optimasi *hyperparameter*, sedangkan model kedua setelah melewati optimasi *hyperparameter*. Model pertama menggunakan parameter manual yang dipilih sebagai berikut:



1. `n_estimators = 400`
Menentukan jumlah pohon keputusan yang dibangun dalam model. Jumlah 400 pohon dipilih untuk meningkatkan kestabilan dan performa prediksi model.
2. `max_depth = 8`
Membatasi kedalaman maksimum setiap pohon keputusan untuk mencegah model menjadi terlalu kompleks dan mengalami *overfitting*.
3. `min_samples_split = 20`
Menentukan jumlah minimum sampel yang diperlukan untuk melakukan pemisahan pada suatu *node* untuk membantu mengontrol kompleksitas model.
4. `min_samples_leaf = 10`
Menetapkan jumlah minimum sampel pada setiap daun (*leaf node*), sehingga model lebih general dan tidak terlalu sensitif terhadap data ekstrem.
5. `max_features = "sqrt"`
Menentukan jumlah fitur yang dipertimbangkan saat melakukan pemisahan *node*, yaitu akar kuadrat dari total fitur. Strategi ini meningkatkan keberagaman antar pohon dalam *Random Forest*.
6. `random_state = 42`
Digunakan untuk memastikan hasil pelatihan model dapat direproduksi (*reproducibility*).
7. `n_jobs = -1`
Memanfaatkan seluruh inti prosesor yang tersedia untuk mempercepat proses pelatihan model.
8. `oob_score = True`
Mengaktifkan perhitungan *Out-of-Bag* (OOB) *Score* sebagai estimasi performa model tanpa memerlukan data validasi terpisah.

Setelah seluruh parameter ditentukan, model dilatih menggunakan data pelatihan dengan metode `fit()`.

```
def train_random_forest(X_train, y_train):
    rf = RandomForestRegressor(
        n_estimators=400,
        max_depth=8,
        min_samples_split=20,
        min_samples_leaf=10, |
        max_features="sqrt",
        random_state=42,
        n_jobs=-1,
        oob_score=True
    )
    rf.fit(X_train, y_train)
    return rf

rf_model = train_random_forest(X_train, y_train)
```

Gambar 8. Proses Konfigurasi Model

Untuk model kedua, dilakukan proses optimasi *hyperparameter* untuk melakukan perbandingan kinerja dengan model pertama. Penelitian ini menggunakan teknik *Grid Search* dengan menggunakan distribusi nilai parameter sebagaimana ditunjukkan pada Tabel 3.

Tabel 3. Distribusi Nilai *Hyperparameter*

Parameter	Distribusi Nilai
<code>n_estimators</code>	[300, 500, 800]
<code>max_depth</code>	[5, 10, None]
<code>min_samples_split</code>	[5, 10, 20]
<code>min_samples_leaf</code>	[5, 10, 15]
<code>max_features</code>	["sqrt", "log2", 0.5]

Berdasarkan hasil proses *Grid Search*, diperoleh konfigurasi *hyperparameter* terbaik sebagaimana ditunjukkan pada Tabel 4.

Tabel 4. Nilai *Hyperparameter* Terbaik

Parameter	Nilai
n_estimators	300
max_depth	10
min_samples_split	5
min_samples_leaf	5
max_features	0.5

Pengujian dan Evaluasi

Model yang telah dilatih kemudian diuji untuk memprediksi data baru yang belum pernah dilihat sebelumnya (*data testing*). Setelah itu dilakukan evaluasi dengan memperhatikan indikator MAE, RMSE, dan R^2 . Didapatkan hasil yang disajikan dalam Tabel 5 dan Tabel 6.

Tabel 5. Evaluasi Model Sebelum *Grid Search*

Indikator	Train	Test
MAE	43,77	72,84
RMSE	60,77	94,55
R^2	0,9952	0,8103

Tabel 6. Evaluasi Model Setelah *Grid Search*

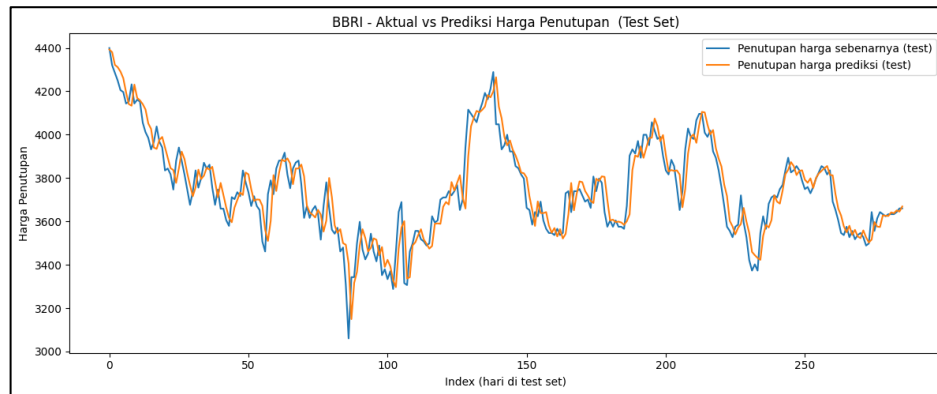
Indikator	Train	Test
MAE	31,07	64,02
RMSE	43,71	84,51
R^2	0,9975	0,8484

Berdasarkan Tabel 5 dan Tabel 6, dapat diketahui bahwa model kedua yang telah melalui proses *Grid Search* menunjukkan performa yang lebih baik dibandingkan dengan model pertama yang menggunakan parameter manual. Peningkatan performa ini terlihat dari penurunan nilai *Mean Absolute Error* (MAE) pada data uji dari 72,84 menjadi 64,02, serta penurunan nilai *Root Mean Squared Error* (RMSE) dari 94,55 menjadi 84,51. Penurunan nilai MAE dan RMSE tersebut mengindikasikan bahwa rata-rata kesalahan prediksi model terhadap nilai aktual menjadi lebih kecil setelah dilakukan optimasi parameter.

Selain itu, nilai koefisien determinasi (R^2) pada data uji juga mengalami peningkatan dari 0,8103 pada model pertama menjadi 0,8484 pada model kedua. Hal ini menunjukkan bahwa model hasil *Grid Search* mampu menjelaskan variasi harga saham dengan lebih baik dibandingkan model awal. Dengan demikian, proses pencarian parameter terbukti efektif dalam meningkatkan akurasi dan kemampuan generalisasi model prediksi harga saham.

Ditinjau dari potensi *overfitting*, kedua model menunjukkan perbedaan nilai evaluasi antara data pelatihan dan data pengujian terutama pada nilai R^2 yang sangat tinggi pada data latih (di atas 0,99). Hal ini mengindikasikan adanya kecenderungan model untuk menyesuaikan diri secara sangat baik terhadap data pelatihan. Namun demikian, peningkatan performa pada data uji setelah *Grid Search* yang ditunjukkan oleh penurunan MAE dan RMSE serta peningkatan R^2 , menunjukkan bahwa *overfitting* yang terjadi tidak bersifat berat. Sebaliknya, model kedua justru memiliki kemampuan generalisasi yang lebih baik dibandingkan model pertama.

Adapun grafik perbandingan antara harga saham aktual (sebenarnya) dan hasil prediksi model kedua yang disajikan pada Gambar 9 memperkuat hasil evaluasi kuantitatif yang telah diperoleh sebelumnya. Grafik tersebut menunjukkan bahwa pola prediksi model cenderung mengikuti pergerakan data aktual, baik pada saat harga mengalami kenaikan maupun penurunan. Kesesuaian pola ini mengindikasikan bahwa model mampu menangkap tren dan dinamika pergerakan harga saham secara umum, meskipun masih terdapat selisih kecil pada beberapa titik pengamatan. Secara keseluruhan, hasil ini menunjukkan bahwa model memiliki performa yang cukup baik dalam memprediksi harga saham dan berpotensi digunakan sebagai alat bantu analisis dalam pengambilan keputusan.



Gambar 9. Grafik Sebenarnya vs Prediksi Penutupan Harga Saham BBRI

4. SIMPULAN

Berdasarkan hasil penelitian, model *Random Forest Regression* yang dioptimasi menggunakan *Grid Search* menunjukkan kinerja yang baik dalam memprediksi harga saham. Hal ini dibuktikan oleh nilai *Mean Absolute Error* (MAE) sebesar 64,07 dan *Root Mean Squared Error* (RMSE) sebesar 84,76, serta nilai Koefisien Determinasi (R^2) sebesar 0,8475 yang menunjukkan bahwa model mampu menjelaskan 84,75% variasi data harga saham secara efektif. Meskipun optimasi *Grid Search* meningkatkan performa model, penelitian ini masih memiliki keterbatasan belum mempertimbangkan faktor eksternal seperti kondisi makroekonomi dan sentimen pasar. Oleh karena itu, penelitian selanjutnya disarankan untuk memperluas teknik optimasi *hyperparameter*, mengombinasikan *Random Forest Regression* dengan model lain, serta menambahkan variabel *input* yang lebih beragam agar akurasi dan kemampuan generalisasi model dapat ditingkatkan.

DAFTAR PUSTAKA

- [1] E. Fitri, "Analisis Perbandingan Metode Regresi Linier, Random Forest Regression dan Gradient Boosted Trees Regression Method untuk Prediksi Harga Rumah," *Journal of Applied Computer Science and Technology (JACOST)*, vol. 4, no. 1, pp. 58-64, Jan. 2023, doi: [10.52158/jacost.v4i1.491](https://doi.org/10.52158/jacost.v4i1.491).
- [2] D. Kurniawan, M. Rahmah, and Z. B. Azzahra, "Analisis Pengaruh Investasi Pasar Saham terhadap Pertumbuhan Ekonomi di Indonesia," *Journal of Capital Market and Banking (JCMB)*, vol. 13, no. 3, pp. 153–165, Aug. 2025, doi: [10.63607/jcmb.v13i3.37](https://doi.org/10.63607/jcmb.v13i3.37).
- [3] Pandaya, P. D. Julianti, and I. Suprpta, "Pengaruh Faktor Fundamental Terhadap Return Saham," *Jurnal Akuntansi*, vol. 9 no. 2, pp. 233-243, Nov. 2020, doi: [10.37932/ja.v9i2.156](https://doi.org/10.37932/ja.v9i2.156).
- [4] A. Rasyid, A. B. Muharam, and A. Solichin, "Prediksi Harga Saham Syariah Indonesia Berdasarkan Analisis Fundamental, Teknikal dan Bandarmology Menggunakan Metode Random Forest," *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, vol. 10, no. 2, pp. 1663-1667, Jun. 2025, doi: [10.29100/jipi.v10i2.7855](https://doi.org/10.29100/jipi.v10i2.7855).
- [5] T. Handayani, R. Hasibuan, S. Rizkiyanti, A. Suharmanto, and M. Berlianti, "Faktor-faktor yang Mempengaruhi Return Saham Pada Perusahaan Manufaktur di Bursa Efek Indonesia (BEI) Tahun 2019-2022," *Jurnal Mutiara Akuntansi*, vol. 8, no. 2, pp. 1-18, Dec. 2023, doi: [10.51544/jma.v8i2.4708](https://doi.org/10.51544/jma.v8i2.4708).
- [6] M. Jannah, "Tinjauan Algoritma Machine Learning untuk Prediksi Harga Saham: Studi Literatur," *Jurnal Penelitian Multidisiplin Bangsa*, vol. 1, no. 7, pp. 608–613, Dec. 2024, doi: [10.59837/jpnmb.v1i7.121](https://doi.org/10.59837/jpnmb.v1i7.121).
- [7] A. Fauzi, N. Maulidah, R. Supriyadi, H. Nalatisifa, and S. Diantika, "Prediksi Harga Properti di Indonesia Menggunakan Algoritma Random Forest," *Journal of Artificial Intelligence and Digital Business (RIGGS)*, vol. 4, no. 1, pp. 43-49, Apr. 2025, doi: [10.59837/jpnmb.v1i7.121](https://doi.org/10.59837/jpnmb.v1i7.121).
- [8] E. Fitri, and D. Riana, "Analisa Perbandingan Model Prediction Dalam Prediksi Harga Saham Menggunakan Metode Linear Regression, Random Forest Regression dan Multilayer Perceptron," *Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, vol. 6, no. 1, pp. 69-78, Apr. 2022, doi: [10.46880/jmika.Vol6No1.pp69-78](https://doi.org/10.46880/jmika.Vol6No1.pp69-78).
- [9] M. Y. Urrochman, H. Asy'ari, and F.A. Hizham, "Performance Comparison of Random Forest Regression, SVR Models in Stock Price Prediction," *Jurnal of Computing and Information System*, vol. 21, no. 1, pp. 16-23, Mar. 2025, doi: [10.33480/pilar.v21i1.6072](https://doi.org/10.33480/pilar.v21i1.6072).
- [10] S. A. N. Cahyo and M. Y. T. Sulistyono, "Comparison of Multiple Linear Regression and Random Forest Methods for Predicting National Rice Production in Indonesia," *Journal of Applied Informatics and Computing (JAIC)*, vol. 9, no. 6, pp. 3479-3489, Dec. 2025, doi: [10.30871/jaic.v9i6.11398](https://doi.org/10.30871/jaic.v9i6.11398).

- [11] Suyatna, A. J. Satrio, S. P. Pratama, M. Indarto, F. Syarif, and F. Amsury “Analisis Perbandingan Kinerja Model machine Learning dalam Prediksi Harga Saham TLKM,” *Journal of Artificial Intelligence and Digital Business (RIGGS)*, vol. 4, no. 4, pp. 829-837, Nov. 2025, doi: [10.31004/riggs.v4i4.3476](https://doi.org/10.31004/riggs.v4i4.3476).
- [12] A. R. Fadillah, M. N. Fauzan, and N. S. Fathonah, “Analisis Perbandingan Linear Regression dan Random Forest Regression Untuk Prediksi Batas Kredit: Pendekatan Optimasi Hyperparameter,” *Jurnal Informatika Polinema*, vol. 10, no. 4, pp. 543-549, Aug. 2024, doi: [10.33795/jip.v10i4.5700](https://doi.org/10.33795/jip.v10i4.5700).
- [13] D. Z. H. Iskandar and Y. Ramadhan, “Optimasi Parameter Random Forest Menggunakan Grid Search Untuk Analisis Time Series,” *Fountain of Informatics*, vol. 8, no. 2, pp. 67-74, Oct. 2025, doi: [10.21111/fij.v8i2.10532](https://doi.org/10.21111/fij.v8i2.10532).
- [14] I. M. Tirta, A. Riski, and N. Sholikhah, “Prediksi Pergerakan Saham BBRI ditengah Issue Ancaman Resesi 2023 dengan Pendekatan Machine Learning”, *Jurnal Teknologi dan Manajemen Informatika*, vol. 9, no. 1, pp. 20-27, Jun. 2023, doi: [10.26905/jtmi.v9i1.9135](https://doi.org/10.26905/jtmi.v9i1.9135).
- [15] W. C. Utomo, “Prediksi Harga Saham PT Bank Rakyat Indonesia Tbk Menggunakan AUTOML H2O,” *Jurnal Ilmiah KOMPUTASI*, vol. 23, no. 3, pp. 397-403, Sep. 2024, doi: [10.32409/jikstik.23.3.3624](https://doi.org/10.32409/jikstik.23.3.3624).