

BAB III METODE PENELITIAN

3.1 Objek Penelitian

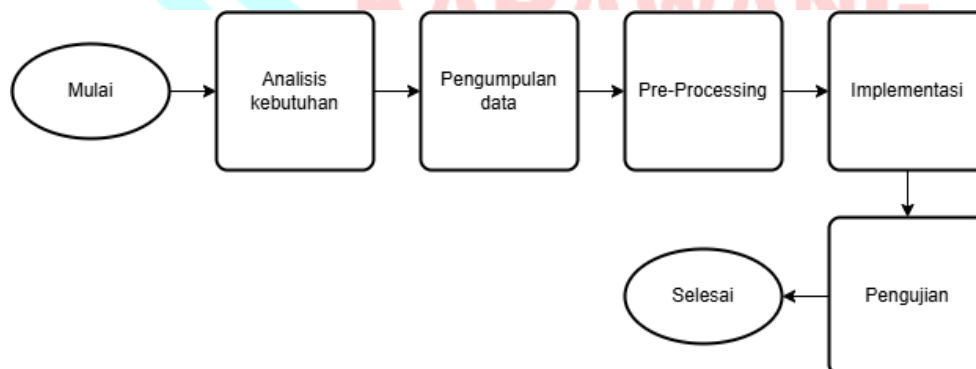
Objek penelitian ini adalah tentang prediksi harga penutupan saham bank BCA, yang dikenal dengan kode BBKA. Dataset yang digunakan data historis yang diambil melalui *website id.investing.com* dengan periode bulan Januari tahun 2020 hingga tahun bulan Desember 2024. Algoritma yang digunakan untuk melakukan prediksi ini, yaitu *Random Forest* dan *Support Vector Regression*.

3.2 Waktu Penelitian

Waktu penelitian ini berlangsung selama empat bulan, dimulai dari November 2024 hingga Februari 2025. Rangkaian kegiatan penelitian dirancang secara sistematis untuk memastikan semua tahapan penelitian dapat dilaksanakan sesuai dengan tujuan yang telah ditetapkan. Berikut adalah rangkaian waktu penelitian.

3.3 Prosedur Penelitian

Pada prosedur penelitian ini menggambarkan secara umum proses penelitian yang akan dilakukan, mencakup langkah langkah yang digambarkan pada alur prosedur penelitian di bawah ini.



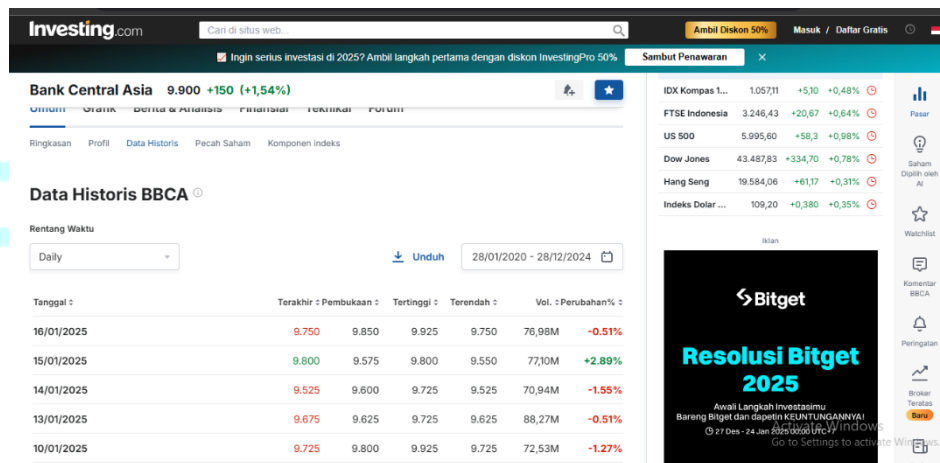
Gambar 3.1 Prosedur Penelitian

3.4 Analisis kebutuhan

Pada tahap ini, dilakukan analisis terlebih dahulu mengenai topik prediksi harga saham yang telah dilakukan melalui *Google Scholar*. Melihat urgensi apa yang akan di angkat berdasarkan studi literatur yang telah dilakukan.

3.5 Pengumpulan data

Pengumpulan dilakukan dengan cara mengunduh melalui *website id.investing.com*, dengan mengambil periode tahun 2020 bulan Januari hingga tahun 2024 bulan Desember. Sumber <https://id.investing.com/equities/bnk-central-as-historical-data>.



Gambar 3.2 Data Historis BBCA

3.6 Pre-Processing

Pada *pre-processing* dilakukan 5 tahap pengerjaan, yaitu Konversi tanggal ke format *datetime*, penyaringan kolom numerik, menangani *missing values*, pemisahan fitur (x) dan fitur (y), dan melatih data *training* dan data *testing*, dapat dijelaskan di bawah ini :

1. Konversi tanggal ke format *datetime*

Data saham memiliki dimensi waktu (*time series*). Oleh karena itu, kolom tanggal harus dikonversi ke format *datetime* agar dapat diurutkan dan digunakan dalam analisis yang bergantung pada waktu.

2. Konversi Desimal untuk Analisis Time Series Harga Saham

Setelah dilakukan konversi angka desimal dari format lokal yang menggunakan koma (,) menjadi titik (.), nilai-nilai pada kolom numerik seperti Terakhir, Pembukaan, Tertinggi, dan Terendah ditampilkan dalam format *float*.

Pseudocode Konversi desimal untuk data *timeseries* :

Mulai

1. Inisialisasi daftar kolom harga yang akan di proses, yaitu: ['Terakhir', 'Pembukaan', 'Tertinggi', 'Terendah'].

2. Lakukan iterasi untuk setiap kolom dalam daftar kolom harga:
 - a) ubah tipe data setiap nilai pada kolom menjadi string.
 - b) Ganti setiap tanda koma (,) pada nilai menjadi titik (.).
 - c) Konversi kembali nilai pada kolom menjadi tipe data float.
 3. Tampilkan lima baris pertama dari data hasil pra-pemrosesan.
- Selesai

2. Menangani Missing Values

Missing values digunakan untuk memeriksa jumlah data yang hilang dalam setiap kolom *dataset*. Dalam analisis data, penting untuk mengetahui apakah terdapat nilai kosong (NaN / *null values*) dalam dataset sebelum melakukan pemrosesan lebih lanjut.

Pseudocode Cek Missing Values:

Mulai

1. Hitung jumlah nilai yang hilang (*missing values*) pada setiap kolom dalam dataset.
2. Simpan hasil perhitungan ke dalam sebuah variabel.
3. Tampilkan hasil jumlah nilai yang hilang dalam bentuk tabel.

Selesai

4. Melatih data *training* dan data *testing*

Pada tahap ini, *dataset* dibagi menjadi 2 bagian, yaitu data latih dan data uji. rasio 70% untuk latih dan 30% untuk data uji, memastikan bahwa model memiliki cukup data untuk belajar tanpa kehilangan kemampuan generalisasi saat diuji dengan data baru. Sebanyak 70% data digunakan untuk latih agar model dapat menangkap pola historis dalam harga saham, sementara 30% data uji digunakan untuk mengukur performa model terhadap data yang belum pernah dilihat sebelumnya. Rasio ini dipilih untuk menghindari *overfitting*.

Pseudocode split data:

Mulai

1. Tentukan proporsi data uji sebesar 30%, dan data latih sebesar 70%.
2. Lakukan pembagian data menjadi dua bagian:
 - a) data latih dan target latih sebanyak 70% dari data awal.

b) Data uji dan target uji sebanyak 30% dari data awal.

3. Pembagian dilakukan berdasarkan urutan waktu (*time series*) tanpa pengacak

4. Simpan juga pembagian tanggal sesuai dengan data latih dan data uji.

Selesai

5. Rekayasa Fitur (Feature Engineering)

pada tahap ini dilakukan tahap rekayasa fitur untuk membantu mendapatkan informasi yang dapat digunakan oleh model dalam mempelajari pola dari data historis harga saham. Moving average dengan periode 3, 5, dan 10 hari memiliki pengaruh yang signifikan terhadap return saham perbankan, di mana semakin pendek periode MA maka semakin sensitif terhadap pergerakan harga dan memberikan sinyal tren yang lebih cepat. (Majiah & Asnawi, 2024) Sehingga model dapat memahami tren serta perubahan yang terjadi dari waktu ke waktu.

Pseudocode Rekayasa Fitur

Mulai

1. Hitung rata-rata bergerak 5 hari dari kolom 'Terakhir', lalu simpan sebagai fitur baru 'MA5'.
2. Hitung rata-rata bergerak 10 hari dari kolom 'Terakhir', lalu simpan sebagai fitur baru 'MA10'.
3. Buat fitur lag satu hari dari kolom 'Terakhir', yaitu nilai dari hari sebelumnya, lalu simpan sebagai 'Lag1'.
4. Hitung persentase perubahan selama lima hari terakhir dari kolom 'Terakhir', simpan sebagai fitur 'Return_5d'.
5. Hitung volatilitas dalam bentuk simpangan baku selama lima hari terakhir dari kolom 'Terakhir', simpan sebagai fitur 'Volatility_5d'.
6. Hapus seluruh baris data yang mengandung nilai kosong akibat proses perhitungan rolling dan shifting.
7. Tampilkan lima baris pertama dari data yang telah direkayasa fiturnya.

Selesai

6. Normalisasi Data

Standarisasi merupakan tahap penting terutama ketika menggunakan algoritma yang sensitif terhadap skala, seperti *Support Vector Regression (SVR)*.

Pseudocode Normalisasi :

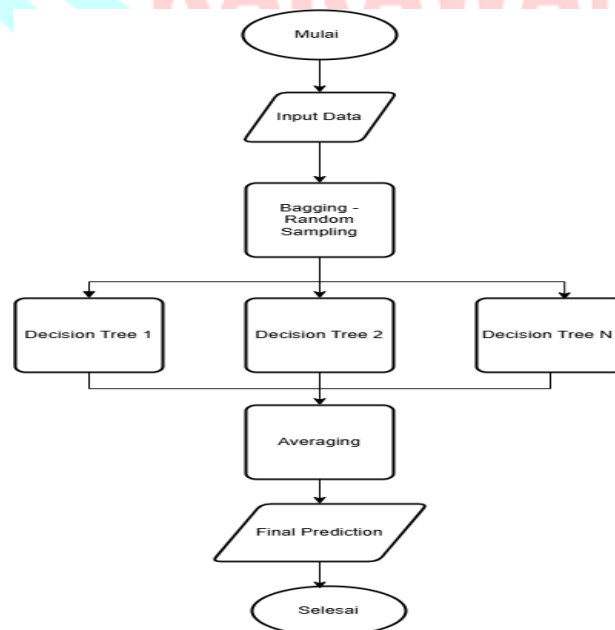
Mulai

1. Inisialisasi alat normalisasi menggunakan metode StandardScaler.
2. Lakukan pelatihan dan transformasi terhadap data latih agar memiliki nilai rata-rata 0 dan standar deviasi 1.
3. Gunakan scaler yang telah dilatih dari data latih untuk mentransformasi data uji.
4. Simpan hasil transformasi sebagai data latih terstandarisasi dan data uji terstandarisasi.

Selesai

3.7 Implementasi Algoritma *Random Forest*

Berdasarkan jurnal (Dikan Ismafillah *et al.*, 2023). Alur dari algoritma *Random forest* dijelaskan dalam bentuk visual flowchart seperti di bawah ini :



Gambar 3.3 *Flowchart Random Forest*

Penjelasan dan *flowchart* pada gambar 3.3 adalah sebagai berikut :

1. Tahap awal, melakukan input data histori saham BBKA yang sudah di dapatkan di *website id,investing.com*.
2. Tahap kedua, lalu melakukan *bagging – random sampling*, yaitu teknik yang bekerja dengan cara membangun model (*decision tree*) dari subset data yang dipilih secara acak.
3. Tahap ketiga, melatih beberapa *decision tree* dengan subset data yang didapat dari hasil *bagging*.
4. Tahap keempat, melakukan tahap *averaging*, untuk menghitung *mean* dari prediksi seluruh *decision tree*.
5. Tahap terakhir, yaitu *final prediction*. Menghasilkan prediksi akhir berdasarkan nilai rata rata dari semua *decision tree*.

Berdasarkan jurnal (Ishlah et al., 2023) Alur dari algoritma Support Vector Regression dijelaskan dalam bentuk visual flowchart seperti di bawah ini :

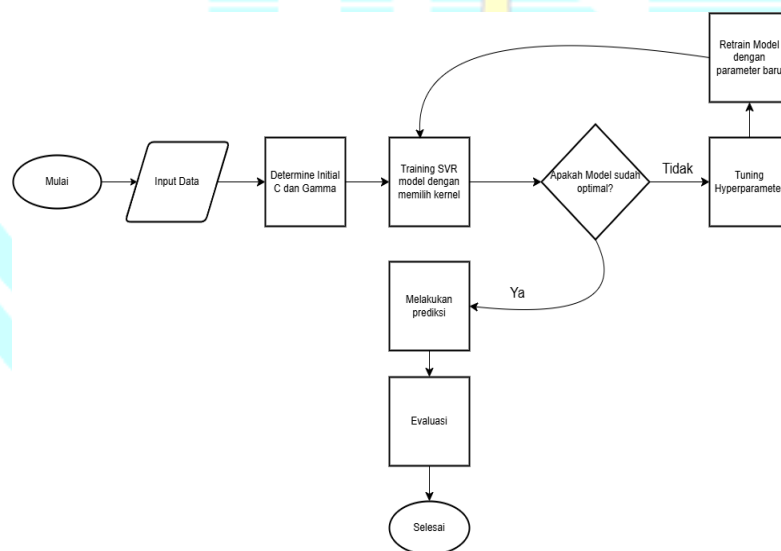
3.7.1 Pseudocode Random Forest

Mulai

1. Tentukan rentang nilai untuk parameter Random Forest, yaitu:
 - a) Jumlah pohon (*n_estimators*): 50, 100, 200, 250.
 - b) Kedalaman maksimum pohon (*max_depth*): 3, 5, 6.
 - c) Minimum sampel untuk pemisahan (*min_samples_split*): 20, 50.
 - d) Minimum sampel pada daun (*min_samples_leaf*): 10, 20, 30.
2. Siapkan skema validasi silang menggunakan *TimeSeriesSplit* dengan jumlah pembagian sebanyak 4.
3. Inisialisasi model Random Forest Regressor dengan parameter:
 - a) *random_state* diset agar hasil replikasi konsisten.
 - b) *max_features* diset ke '*sqrt*' untuk mencegah *overfitting*.
4. Lakukan pencarian parameter terbaik menggunakan *Grid Search*

- a) Model dasar: Random Forest
 - b) Parameter yang dicoba: seperti pada langkah 1
 - c) Validasi silang: TimeSeriesSplit
 - d) Skor evaluasi: nilai negatif dari Mean Squared Error
5. Latih model dengan data latih yang telah dinormalisasi.
 6. Ambil model terbaik berdasarkan hasil pencarian parameter.
 7. Gunakan model terbaik untuk melakukan prediksi terhadap data latih.
 8. Gunakan model yang sama untuk melakukan prediksi terhadap data uji selesai.

3.8 Implementasi Algoritma SVR



Gambar 3.4 *Flowchart SVR*

Penjelasan dan flowchart pada gambar 3.3 adalah sebagai berikut :

1. Tahap awal, melakukan input data histori saham BBCA yang sudah di dapatkan di *website id,investing.com*.
2. Tahap kedua, lalu melakukan *Determine initial C dan Gamma*, yaitu untuk mengontrol keseimbangan antara kompleksitas model
3. Tahap ketiga, melatih model SVR dengan memilih *kernel*, *linear kernel* untuk data yang bersifat *linear*, *polynomial kernel* untuk hubungan *non-linear*

dengan *polinomial* derajat tertentu, *radial basis function kernel*, untuk kernel paling umum untuk menangani data non-linear dan *sigmoid kernel*, terkadang digunakan untuk data dengan distribusi tertentu.

4. Tahap keempat melakukan *decision*, apakah model sudah optimal? Jika model sudah optimal bisa dilanjutkan ke tahap selanjutnya, jika tidak lanjut ke proses *Tuning Hyperparameter*.
5. Tahap kelima melakukan *tuning hyperparameter* jika model tidak optimal, pada *hyperparameter* dilakukan beberapa metode seperti *grid search*, *random search*, *bayesian optimization*, dan *genetic algorithm*
6. Tahap ketujuh jika sudah melakukan tuning, yaitu tahap retrain model dengan parameter baru dari kombinasi *hyperparameter* yang lebih baik
7. Tahap terakhir melakukan prediksi dengan model yang sudah optimal, dengan data uji atau data baru yang sudah di latih ulang pada *tuning hyperparameter*.

3.8.1 Pseudocode SVR

Mulai

1. Tentukan rentang parameter yang akan dieksplorasi, meliputi:
 - a) Nilai C: 0.1, 1, 10, 100, 150, 200.
 - b) Nilai epsilon: 0.001, 0.01, 0.05, 0.1, 0.2
 - c) Nilai gamma: 'scale', 'auto', 0.001, 0.01, 0.1
2. Siapkan validasi silang berbasis urutan waktu (Time Series Cross Validation) dengan jumlah pembagian sebanyak 4.
3. Inisialisasi pencarian parameter menggunakan Grid Search dengan konfigurasi:
 - a) Model dasar: Support Vector Regression (SVR) dengan kernel radial basis function (RBF)
 - b) Parameter yang diuji: sesuai pada langkah 1
 - c) Metode evaluasi: nilai negatif dari Mean Squared Error
 - d) Skema validasi: Time Series Split
4. Latih model SVR menggunakan data latih yang telah dinormalisasi.

5. Simpan model terbaik berdasarkan kombinasi parameter yang menghasilkan performa terbaik.
6. Lakukan prediksi terhadap data uji menggunakan model SVR terbaik yang telah diperoleh.

Selesai

3.9 Pengujian

Pada tahap ini, selanjutnya, dilakukan pengujian pada dataset yang dimiliki dengan algoritma yang akan di gunakan, pada kasus ini, algoritma *Random Forest* dan *Support*.

Tabel 3.1 Tabel Pengujian

No	Algoritma	Metode Pengujian	Parameter Utama	Matriks Evaluasi	Hasil Evaluasi
1	Random Forest	Train – Test split (70:30)	n_estimators=100, random_state=42	MAE, MSE, SE, R ² Score	
2	Support Vector Regression	Train – Test split (70:30)	kernel='rbf', C=1.0, gamma='scale'	MAE, MSE, SE, R ² Score	

Vector Machine yang akan digunakan. Setelah model dilatih, dilakukan pengujian menggunakan data testing untuk mengevaluasi kinerja prediksi berdasarkan beberapa metrik evaluasi, seperti

Mean Absolute Error (MAE), *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan *R-Squared* (R²). Hasil dari kedua algoritma akan dibandingkan untuk menentukan model yang lebih optimal dalam memprediksi harga saham BBCA.

Pada tabel pengujian, dilakukan implementasi dua algoritma, yaitu *Random Forest* dan *Support Vector Regression* (SVR), dengan metode *Train-Test Split* (70:30), di mana 70% data digunakan untuk melatih model dan 30%.

untuk pengujian. Untuk mengevaluasi kinerja model, digunakan metrik *Mean Absolute Error* (MAE), *Mean*

Squared Error (MSE), dan *R² Score*, di mana MAE mengukur rata-rata kesalahan absolut, MSE memberikan penalti lebih besar terhadap kesalahan yang lebih jauh, dan *R² Score* mengukur seberapa baik model dapat menjelaskan variabilitas data

