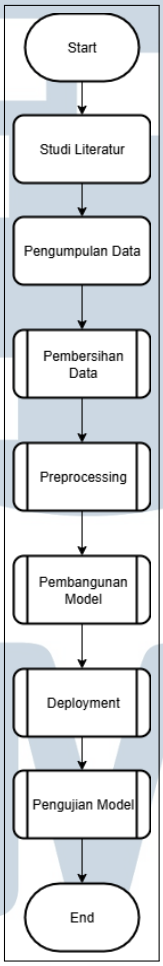
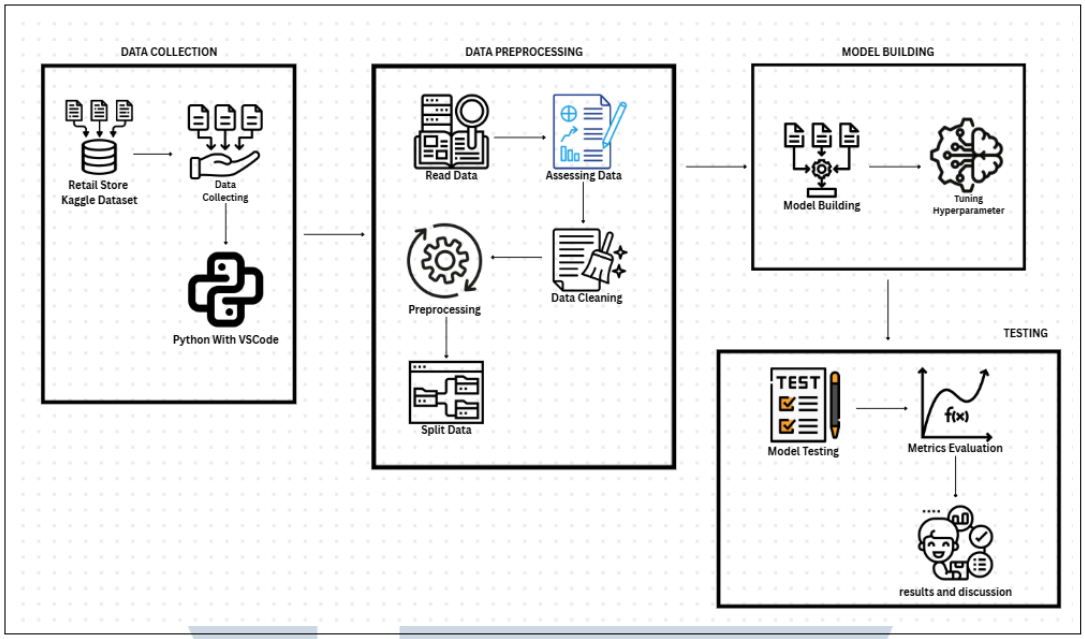
BAB 3
METODOLOGI PENELITIAN

Pada bagian ini tertulis langkah-langkah yang dilakukan dalam menyusun dan mengerjakan penelitian. Langkah-langkah penelitian yang digunakan, dimulai dari awal hingga akhir selesai seperti pada Gambar 3.1.



Gambar 3.1. Flow Metodologi Penelitian

Lebih lengkapnya, alur penelitian dapat dilihat pada Gambar 3.2. Pada gambar tersebut, dapat dilihat beberapa tahapan seperti pengambilan data dari kaggle, pembersihan data, preprocessing, dan pemisahan data pada set data latih dan uji.



Gambar 3.2. Gambar Alur Penelitian

Setelah data dipisah, selanjutnya model dibangun dengan data latih yang selanjutnya akan dilakukan hyperparameter tuning menggunakan RandomizedSearchCV untuk mendapatkan hyperparameter terbaik. Tahap terakhir ialah pengujian pada model dengan skenario berbeda dan hasilnya akan dievaluasi oleh metrik evaluasi seperti MAE, RMSE, dan MAPE.

3.1 Studi Literatur

Peneliti melakukan studi literatur sebagai langkah awal untuk menggali ide, informasi, serta referensi yang relevan dalam mendukung proses penelitian ini. Studi literatur merupakan metode yang digunakan untuk memahami dan menyelesaikan suatu permasalahan dengan melakukan reset berbagai hasil penelitian sebelumnya yang memiliki keterkaitan. Melalui metode ini, peneliti dapat memperoleh dasar teori, memahami perkembangan topik yang telah diteliti, serta mengidentifikasi celah yang dapat dijadikan ruang pengembangan penelitian selanjutnya.

Sumber informasi yang dijadikan referensi dalam studi literatur ini mencakup jurnal, skripsi, tesis, dan laporan penelitian ataupun proyek dari sumber terbuka. Selain itu, peneliti juga memanfaatkan sumber digital seperti YouTube dan berbagai situs edukatif atau teknologi untuk memperoleh solusi praktis dalam pengembangan model serta penulisan skripsi. Situs-situs seperti Google Scholar,

ResearchGate, Medium, GitHub, dan dokumentasi resmi dari alat atau pustaka yang digunakan turut dijadikan referensi tambahan yang memperkaya hasil studi literatur ini.

3.2 Pengumpulan Data

Data yang digunakan pada penelitian ini berasal dari Kaggle, platform online yang menyediakan beragam dataset publik. Dataset ini berjudul Retail Store Inventory Forecasting yang terediri dari 73.100 entri dan 15 kolom yang merekam aktivitas operasional ritel secara dinamis di berbagai toko dan wilayah. Dataset ini mencakup data berpenanda waktu terkait tingkat persediaan, jumlah unit terjual dan dipesan, serta perkiraan permintaan dan informasi harga. Faktor eksternal seperti diskon, harga kompetitor, kondisi cuaca, hari libur, dan tren musiman juga dicatat, sehingga memberikan konteks yang berharga terhadap fluktuasi penjualan dan persediaan. Dataset ini cocok digunakan untuk analisis ritel, peramalan permintaan, serta memahami pengaruh faktor eksternal dan promosi terhadap kinerja toko.

Data ini digunakan untuk training model dan melakukan regresi untuk melakukan prediksi penjualan. Data ini mencatat histori penjualan toko dari tahun 2022 hingga 2024. Setelah diakses, data tersebut telah siap untuk ke langkah selanjutnya yaitu pengolahan data. Detail terkait penamaan fitur dan informasinya bisa dilihat pada Gambar 3.3.

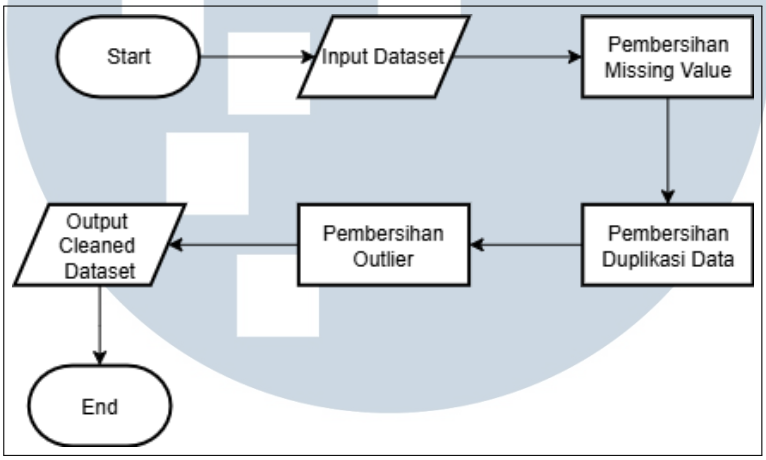
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Date	Store ID	Product ID	Category	Region	Inventory Level	Units Sold	Units Ordered	Demand	Price	Discount	Weather Condition	Holiday/Promotion	Competitor Pricing	Seasonality
2	01/01/2022	S001	P0001	Groceries	North	231	127	55	135.47	33.5	20	Rainy	0	29.69	Autumn
3	01/01/2022	S001	P0002	Toys	South	204	150	66	144.04	63.01	20	Sunny	0	66.16	Autumn
4	01/01/2022	S001	P0003	Toys	West	102	65	51	74.02	27.99	10	Sunny	1	31.32	Summer
5	01/01/2022	S001	P0004	Toys	North	469	61	164	62.18	32.72	10	Cloudy	1	34.74	Autumn
6	01/01/2022	S001	P0005	Electronics	East	166	14	135	9.26	73.64	0	Sunny	0	68.95	Summer
7	01/01/2022	S001	P0006	Groceries	South	138	128	102	139.82	76.83	10	Sunny	1	79.35	Winter
8	01/01/2022	S001	P0007	Furniture	East	359	97	167	108.92	34.16	10	Rainy	1	36.55	Winter
9	01/01/2022	S001	P0008	Clothing	North	380	312	54	329.73	97.99	5	Cloudy	0	100.09	Spring
10	01/01/2022	S001	P0009	Electronics	West	183	175	135	174.15	20.74	10	Cloudy	0	17.66	Autumn
11	01/01/2022	S001	P0010	Toys	South	108	28	196	24.47	59.99	0	Rainy	1	61.21	Winter
12	01/01/2022	S001	P0011	Furniture	South	258	150	153	152.74	58.53	10	Sunny	1	61.42	Spring
13	01/01/2022	S001	P0012	Clothing	West	66	24	70	26.75	58.25	20	Snowy	0	62.21	Spring
14	01/01/2022	S001	P0013	Toys	South	96	42	85	41.46	43.6	0	Cloudy	0	46.31	Spring
15	01/01/2022	S001	P0014	Clothing	West	193	12	187	6.8	78.11	0	Sunny	0	80.06	Spring
16	01/01/2022	S001	P0015	Clothing	North	379	369	154	363.46	92.99	15	Snowy	0	95.8	Winter
17	01/01/2022	S001	P0016	Electronics	North	363	255	69	255.74	21.9	5	Cloudy	1	20.27	Autumn
18	01/01/2022	S001	P0017	Toys	West	318	246	177	255.37	21.07	20	Sunny	0	16.49	Winter
19	01/01/2022	S001	P0018	Clothing	South	241	151	47	147.27	19.57	5	Cloudy	0	23.13	Autumn
20	01/01/2022	S001	P0019	Clothing	East	352	257	186	267.38	73.28	10	Cloudy	0	77.26	Winter
21	01/01/2022	S001	P0020	Toys	East	274	99	166	115.23	30.24	5	Cloudy	0	27.1	Spring
22	01/01/2022	S002	P0001	Groceries	South	343	104	144	112.55	32.8	20	Sunny	1	30.78	Spring
23	01/01/2022	S002	P0002	Clothing	West	373	141	151	135.62	46.16	10	Rainy	1	45.21	Winter
24	01/01/2022	S002	P0003	Furniture	East	445	182	119	176.31	20.46	20	Snowy	1	22.0	Spring
25	01/01/2022	S002	P0004	Toys	West	191	63	115	54.57	26.19	0	Cloudy	0	28.06	Autumn
26	01/01/2022	S002	P0005	Toys	West	281	156	25	150.9	83.29	20	Rainy	0	80.64	Autumn
27	01/01/2022	S002	P0006	Furniture	North	492	250	168	242.57	85.59	20	Cloudy	0	86.82	Winter
28	01/01/2022	S002	P0007	Groceries	West	460	393	70	401.48	91.13	5	Sunny	0	93.33	Summer
29	01/01/2022	S002	P0008	Electronics	West	304	29	99	35.54	66.96	20	Snowy	0	65.36	Summer

retail_store_inventory

Gambar 3.3. Retail Store Dataset Kaggle

3.3 Pembersihan Data

Pembersihan data merupakan tahap awal dalam preprocessing yang bertujuan untuk meningkatkan kualitas data sebelum digunakan dalam proses pelatihan model. Data yang bersih dan konsisten akan memberikan pengaruh signifikan terhadap akurasi dan keandalan model machine learning. Pada penelitian ini, proses pembersihan data dilakukan melalui beberapa langkah, yaitu penanganan missing value, penghapusan data duplikat, dan identifikasi serta penanganan outlier. Tahapan ini divisualisasikan dalam flowchart seperti pada Gambar 3.5.



Gambar 3.4. Flowchart Pembersihan Data

3.3.1 Pembersihan Missing Value

Missing value atau nilai kosong dalam dataset dapat menyebabkan gangguan pada proses pelatihan model, seperti error saat training atau hasil prediksi yang bias. Oleh karena itu, salah satu langkah yang dilakukan adalah mengidentifikasi dan menghapus baris data yang mengandung nilai kosong. Penghapusan missing value dipilih karena metode ini lebih sederhana dan efektif apabila proporsi data kosong tergolong kecil. Tujuan dari langkah ini adalah untuk menjaga konsistensi data dan menghindari kesalahan logika selama proses pembelajaran mesin.

3.3.2 Pembersihan Duplikasi Data

Data duplikat merujuk pada baris data yang identik dan tercatat lebih dari satu kali dalam dataset. Keberadaan data yang terduplikasi dapat memperbesar bobot pengaruh nilai tertentu, sehingga dapat menyebabkan model belajar dari

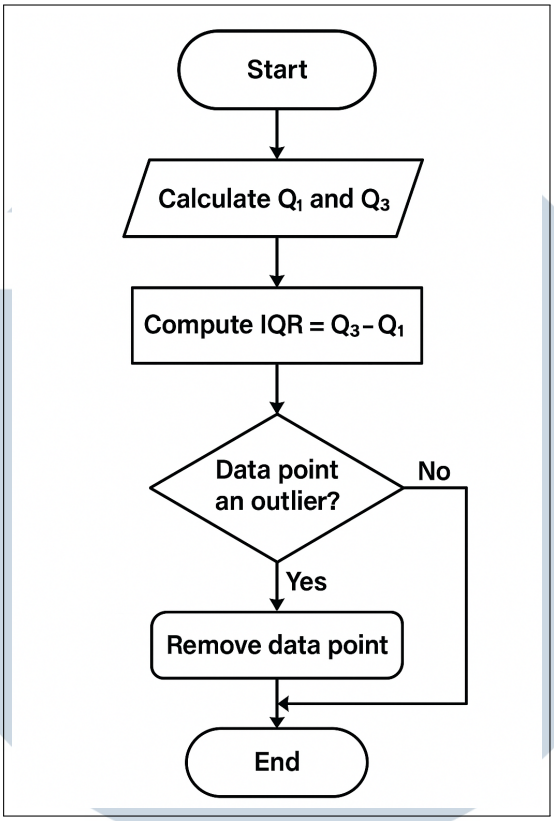
informasi yang tidak akurat. Oleh karena itu, pada tahap ini dilakukan proses identifikasi dan penghapusan baris data yang terduplikasi. Dengan menghapus data yang berulang, distribusi data menjadi lebih representatif dan model dapat dilatih dengan data yang lebih bervariasi dan tidak bias.

3.3.3 Pembersihan Outlier

Outlier merupakan nilai data yang secara signifikan berbeda dari sebagian besar nilai lainnya dalam suatu fitur. Keberadaan outlier dapat menyebabkan model machine learning menjadi bias, overfitting, atau memberikan prediksi yang tidak akurat. Hal ini karena model dapat belajar dari pola yang tidak representatif terhadap populasi data secara keseluruhan. Oleh karena itu, deteksi dan penghapusan outlier menjadi langkah penting dalam tahap pembersihan data.

Pada penelitian ini, metode yang digunakan untuk mengidentifikasi outlier adalah metode Interquartile Range (IQR). IQR dipilih karena merupakan pendekatan statistik yang sederhana namun efektif untuk mendeteksi nilai-nilai ekstrem dalam distribusi data, tanpa terlalu terpengaruh oleh skewness. Secara umum, IQR dihitung sebagai selisih antara kuartil ketiga (Q3) dan kuartil pertama (Q1), dan digunakan untuk menentukan batas bawah dan batas atas dari nilai yang dianggap normal. Nilai yang berada di luar rentang tersebut dianggap sebagai outlier dan akan dihapus dari dataset.

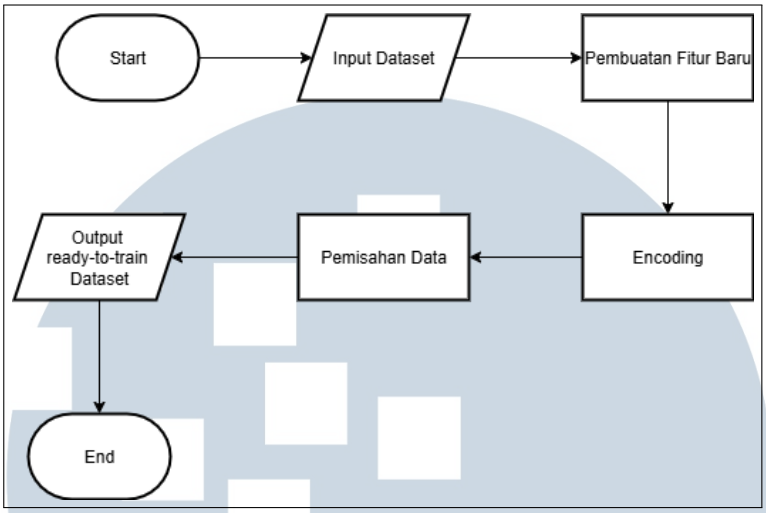
Alur proses pembersihan outlier dengan metode IQR ditunjukkan pada Gambar 3.6, yang menggambarkan tahapan identifikasi hingga penghapusan data yang terdeteksi sebagai outlier.



Gambar 3.5. Flowchart IQR

3.4 Preprocessing

Tahap preprocessing bertujuan untuk mempersiapkan data mentah menjadi dataset yang siap digunakan dalam pelatihan model. Proses ini dilakukan agar data berada dalam format yang sesuai dengan kebutuhan algoritma machine learning, serta meminimalisir potensi kesalahan yang dapat memengaruhi hasil prediksi. Beberapa langkah penting yang dilakukan pada tahap ini mencakup pembuatan fitur baru, encoding fitur kategorikal, dan pemisahan data, seperti yang dijelaskan pada gambar flowchart pada gambar 3.7.



Gambar 3.6. Flowchart Preprocessing

3.4.1 Pembuatan Fitur Baru

Dalam tahap ini, dilakukan pengembangan fitur dari atribut yang sudah ada untuk mengekstrak informasi yang lebih bermakna. Proses ini bertujuan meningkatkan kualitas data dengan menambahkan dimensi baru yang dapat memperkuat kemampuan prediktif model. Pembuatan fitur baru biasanya dilakukan jika ada atribut yang menyimpan informasi kompleks (seperti tanggal atau teks) yang masih perlu dipecah menjadi bagian-bagian yang lebih relevan secara kontekstual. Penambahan fitur yang relevan secara kontekstual berpotensi membantu model dalam mengenali pola yang sebelumnya tersembunyi dalam data asli.

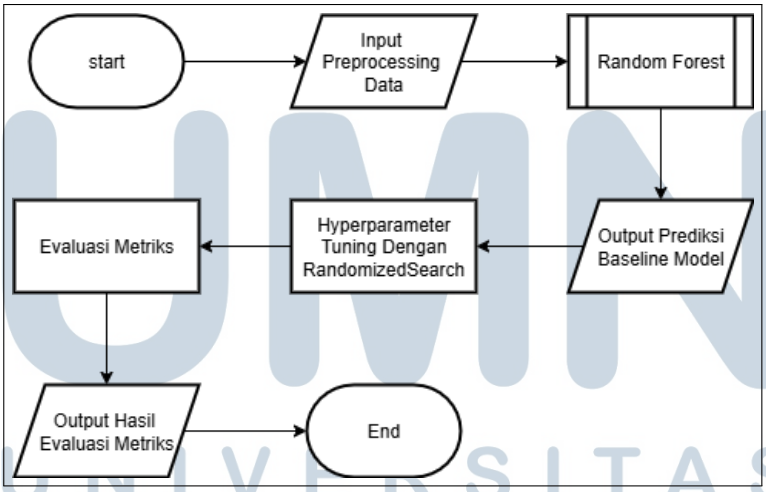
3.4.2 Encoding

Encoding merupakan proses konversi data kategorikal menjadi format numerik agar dapat diproses oleh algoritma machine learning. Sebagian besar algoritma hanya dapat bekerja dengan nilai numerik, sehingga fitur kategorikal perlu diubah ke bentuk angka. Dalam penelitian ini digunakan dua pendekatan, yaitu label encoding dan one-hot encoding. Label encoding digunakan ketika data kategorik bersifat ordinal (ada urutan), sedangkan one-hot encoding digunakan ketika fitur bersifat nominal (tanpa urutan) dan memiliki jumlah kategori yang tidak terlalu banyak. Pemilihan metode encoding disesuaikan dengan sifat data untuk menjaga relevansi informasi tanpa memperkenalkan bias pada model.

3.4.3 Pemisahan Data

Langkah terakhir dalam preprocessing adalah memisahkan dataset menjadi data latih (training) dan data uji (testing). Data latih digunakan untuk membangun model, sementara data uji digunakan untuk mengevaluasi kinerja model secara objektif. Pemisahan ini penting untuk menghindari overfitting dan mendapatkan estimasi performa yang lebih realistis. Pada penelitian ini digunakan rasio 80:20, artinya 80% data digunakan untuk pelatihan dan 20% sisanya untuk pengujian. Rasio ini dipilih karena merupakan praktik umum yang seimbang antara kebutuhan pelatihan model dan evaluasi performa. Salah satu cara untuk membagi data ini adalah dengan menggunakan fungsi train_test_split dari library sklearn.model_selection. Fungsi ini secara otomatis membagi dataset menjadi data latih dan data uji sesuai dengan proporsi yang ditentukan. Dengan cara ini, kita dapat memastikan bahwa model dapat memberikan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya.

3.5 Pembangunan Model



Gambar 3.7. Flowchart Pembangunan Model

Sold berdasarkan fitur-fitur yang telah dipersiapkan dari tahap preprocessing. Algoritma yang digunakan dalam penelitian ini adalah Random Forest Regressor, sebuah metode ensemble yang menggabungkan beberapa decision tree untuk meningkatkan akurasi prediksi dan mengurangi risiko overfitting. Pemilihan algoritma ini didasarkan pada kemampuannya dalam

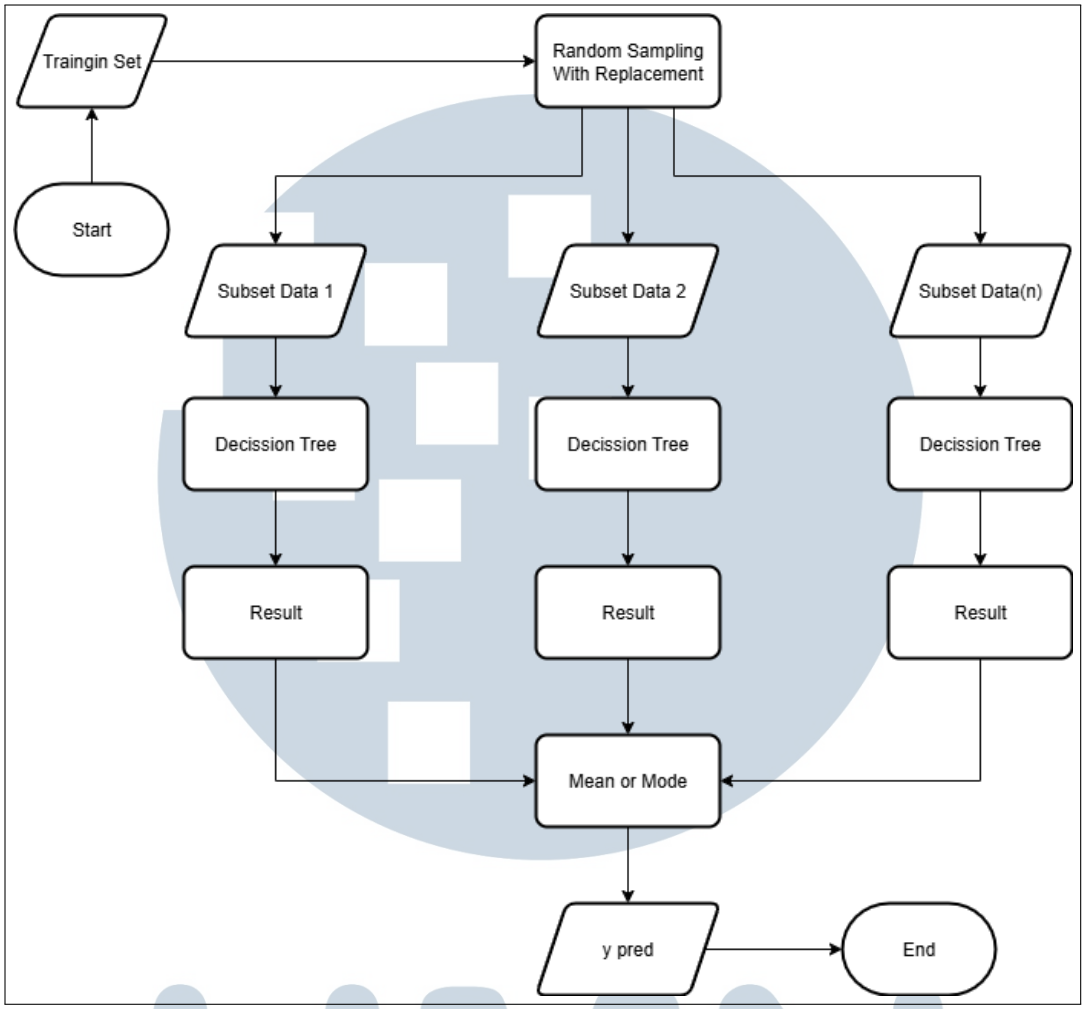
menangani data berdimensi tinggi, mengelola fitur-fitur numerik dan kategorik, serta stabil terhadap outlier dan noise.

Fokus utama dari proyek ini adalah membangun model regresi yang dapat memprediksi kuantitas penjualan produk pada toko retail berdasarkan berbagai informasi seperti inventaris, harga, diskon, kondisi cuaca, musiman, dan faktor eksternal lainnya. Hasil prediksi ini dapat digunakan untuk mendukung pengambilan keputusan dalam pengelolaan stok dan perencanaan permintaan.

Sebelum model dilatih, target variabel yaitu Units Sold terlebih dahulu ditransformasikan menggunakan fungsi logaritmik log1p(). Transformasi ini dilakukan untuk menstabilkan varians target dan mengurangi dampak nilai ekstrim (skewness), sehingga model dapat belajar dengan lebih baik pada data yang lebih terdistribusi normal. Setelah proses prediksi selesai, nilai output dikembalikan ke skala semula menggunakan fungsi eksponensial expm1().

Seperti yang dijelaskan pada gambar 3.8, Secara umum, proses pengembangan model melibatkan langkah-langkah berikut:

1. Menginput dataset hasil preprocessing.
2. Melatih model Random Forest pada data pelatihan.
3. Menghasilkan performa baseline model tanpa tuning hyperparameter.
4. Melakukan hyperparameter tuning menggunakan RandomizedSearchCV untuk mencari Hyperparameter dan performa terbaik.
5. Mengevaluasi performa model berdasarkan metrik MAE, RMSE, dan MAPE.



Gambar 3.8. Flowchart Random Forest

Gambaran cara kerja algoritma Random Forest ditampilkan pada Gambar 3.7. Setiap pohon keputusan dilatih pada subset data (bagging) dan subset fitur yang berbeda untuk menciptakan variasi model. Hasil prediksi dari seluruh pohon kemudian dirata-ratakan untuk mendapatkan nilai akhir. Proses ini menghasilkan model yang lebih akurat dan general dibanding satu decision tree tunggal.

3.6 Deployment

Setelah model prediksi terbaik diperoleh melalui proses pelatihan dan tuning hyperparameter, tahap selanjutnya adalah deployment, yaitu proses menyiapkan dan mempublikasikan aplikasi berbasis web agar dapat digunakan oleh pengguna secara luas. Deployment menjadi langkah penting untuk menerapkan hasil analisis dan model machine learning ke dalam bentuk aplikasi nyata yang dapat diakses melalui

internet.

Pada penelitian ini, deployment dilakukan dengan menggunakan framework *Streamlit*, yaitu framework berbasis Python yang memungkinkan pembuatan aplikasi web secara sederhana dan cepat khususnya untuk kebutuhan data science dan machine learning. Tahapan deployment ini mencakup beberapa langkah penting sebagai berikut:

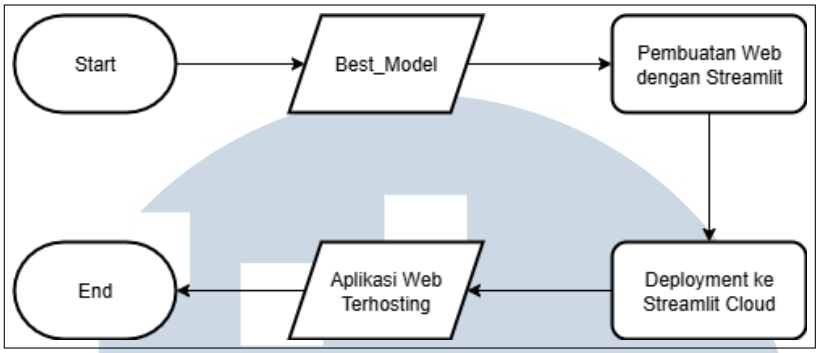
1. **Penyimpanan Model**
   Model prediksi yang telah melalui proses pelatihan dan tuning disimpan dalam format file `.pkl` menggunakan pustaka `joblib`. Penyimpanan ini dilakukan untuk memudahkan proses pemanggilan model saat digunakan pada aplikasi web tanpa perlu melatih ulang model dari awal.

2. **Pembuatan Aplikasi Web dengan Streamlit**
   Aplikasi web dibangun menggunakan `Streamlit`, yang menyediakan antarmuka pengguna (UI) interaktif untuk menginput data dan menampilkan hasil prediksi. Fitur-fitur yang diimplementasikan dalam aplikasi meliputi input numerik, dropdown pilihan kategori, slider diskon, serta tombol eksekusi prediksi.

3. **Integrasi Model ke dalam Aplikasi**
   Model yang telah disimpan dipanggil ke dalam aplikasi `Streamlit`, kemudian diintegrasikan dengan alur input pengguna. Input yang dimasukkan akan diolah dan diproses oleh model, lalu menghasilkan output berupa prediksi jumlah penjualan yang ditampilkan secara langsung.

4. **Deployment ke Streamlit Cloud**
   Repository GitHub yang telah disiapkan kemudian dihubungkan ke akun `Streamlit Cloud` untuk menjalankan aplikasi secara daring. Proses ini mencakup penyesuaian konfigurasi dependencies, pemilihan branch utama, dan penyesuaian file pendukung agar aplikasi dapat berjalan di lingkungan cloud.

Proses deployment ini digambarkan dalam Diagram Alur pada Gambar 3.9.



Gambar 3.9. Diagram alur proses deployment model ke aplikasi web

Tahapan deployment ini bertujuan untuk mengubah hasil penelitian menjadi bentuk aplikasi nyata yang dapat digunakan oleh pengguna secara praktis. Seluruh implementasi teknis dan hasil dari proses ini akan dijelaskan lebih lanjut pada Bab 4.

3.7 Pengujian Model

Tahap pengujian model bertujuan untuk mengevaluasi performa model Random Forest yang telah dibangun dengan kombinasi hyperparameter yang diperoleh dari proses tuning. Pengujian dilakukan secara sistematis melalui serangkaian skenario uji, di mana setiap skenario berfokus pada satu hyperparameter utama. Skema ini ditulis sebagai skenario f(n), yang berarti jumlah skenario uji disesuaikan dengan banyaknya hyperparameter yang dikaji.

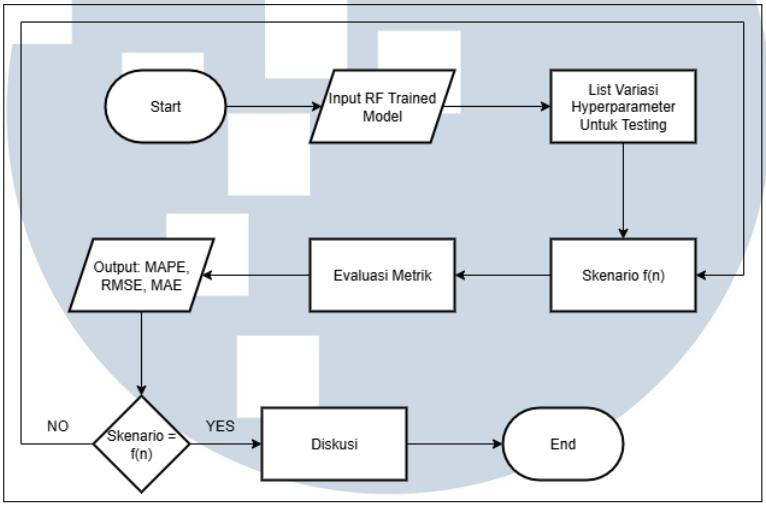
Adapun hyperparameter yang diuji dalam penelitian ini meliputi:

1. n_estimators: Jumlah pohon dalam hutan, n_estimators ini akan mengembangkan jumlah pohonnya sebanyak n
2. max_depth: Kedalaman maksimum pohon keputusan, akan membatasi seberapa jauh kedalaman pohon berdasarkan n(max_depth)
3. min_samples_leaf: Jumlah minimal sampel untuk berada pada simpul daun
4. min_samples_split: jumlah minimal sampel untuk membagi node internal

Setiap pengujian dilakukan dengan menyetel nilai salah satu hyperparameter secara bertahap, sementara hyperparameter lainnya menggunakan konfigurasi terbaik dari hasil RandomizedSearchCV. Model dilatih ulang berdasarkan kombinasi tersebut, kemudian dievaluasi menggunakan metrik performa regresi:

MAE (Mean Absolute Error), RMSE (Root Mean Square Error), dan MAPE (Mean Absolute Percentage Error).

Alur proses pengujian ini dirangkum dalam flowchart seperti pada Gambar 3.10, dimulai dari input model terlatih dan daftar variasi hyperparameter, dilanjutkan dengan eksekusi tiap skenario uji, evaluasi metrik, hingga menghasilkan simpulan pengaruh hyperparameter terhadap performa model. Hasil dari tahap ini akan menjadi dasar dalam menyusun diskusi performa model pada Bab 4.



Gambar 3.10. Flowchart Pengujian Model

3.8 Dokumentasi

Penulisan laporan dilakukan sebagai bentuk dokumentasi menyeluruh atas seluruh tahapan penelitian yang telah dilakukan dari awal hingga akhir. Laporan ini memuat penjabaran setiap proses yang dilalui, mulai dari tahapan penelitian, implementasi algoritma, hasil temuan penelitian, hingga dokumentasi lengkap terkait proses penelitian sesuai dengan standar yang berlaku.