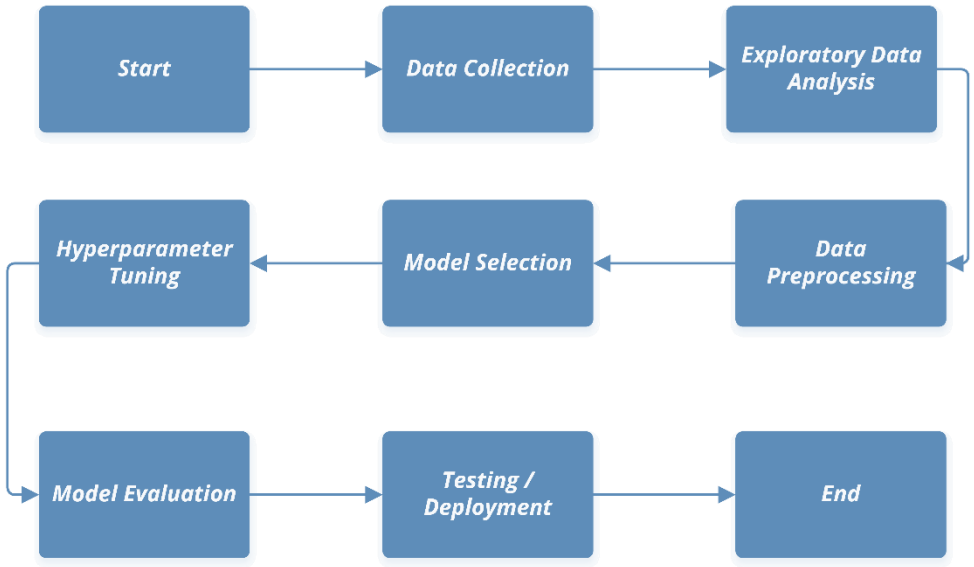
BAB III
METODOLOGI PENELITIAN

3.1 Alur Penelitian

Adapun tahapan penelitian yang dilakukan untuk mengembangkan model *machine learning* yang dibutuhkan menurut Aurélien Géron [16] adalah sebagai berikut.



Gambar 3. 1 Alur Penelitian

3.2 Pengumpulan Data

Penelitian ini mengunakan dataset yang diambil dari *Kaggle*, dengan set data yang berisi 3.855 data pasien penderita kardiovaskular. Set data ini berisi berbagai atribut seperti usia, jenis kelamin, tekanan darah diastolik, tekanan darah sistolik, ras, sel darah merah, Tingkat sedimentasi, Kolesterol, Zat besi, Magnesium, Protein, TIBC, TS, Sel darah putih, BMI, Tekanan nadi dan variabel target berupa status kelangsungan hidup (*survival*).

3.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) adalah proses menganalisis dan menampilkan data dengan tujuan meningkatkan pemahaman yang lebih baik tentang wawasan dari data. Proses ini dimulai dengan menganalisis data yang telah dikumpulkan untuk mengidentifikasi dan menangani nilai yang hilang, outlier, atau tidak konsisten [18].

1) Missng *Value* bisa muncul karena berbagai alasan diantaranya yaitu Kesalahan Pengumpulan Data, Kesalahan Pengolahan Data, Data Tidak Relevan dan lainnya. Ada beberapa cara untuk Penanganan *Missing Values* yaitu menghapus baris dengan *missing values* jika terjadi pada sejumlah kecil baris dan tidak mempengaruhi analisis atau mengimputasi data dengan mengisi *missing values* dengan nilai statistic seperti mean, median, mode dan lainnyaa.
2) *Outlier* dapat merusak visualisasi data, membuat pola yang sebenarnya dalam data menjadi sulit dilihat sehingga mengarahkan pada kesimpulan yang salah. Model juga dapat menjadi *overfitting* atau *underfitting* jika outlier tidak ditangani dengan benar sehingga mempengaruhi performa model *machine learning*. Maka perlu dilakukan pengecekan *outlier* dengan mengidentifikasi dan menangani *outlier*, sehingga data yang digunakan dalam analisis atau modelisasi lebih bersih, akurat, dan representatif, dan menghasilkan hasil yang lebih dapat diandalkan.
3) Distribusi Data adalah konsep dasar dalam statistik yang menggambarkan bagaimana nilai dari variabel tertentu tersebar dalam dataset. Distribusi data menggambarkan frekuensi atau proporsi kemunculan nilai-nilai berbeda dalam dataset. Ini membantu dalam memahami pola, kecenderungan, dan variabilitas data. Untuk memeriksa distribusi data dapat menggunakan *histogram* ataupun *boxplot*.
4) *Corelation Analysis* adalah teknik statistik yang digunakan untuk mengukur dan menganalisis kekuatan serta arah hubungan antara dua variabel atau lebih. Korelasi membantu dalam memahami bagaimana variabel saling berhubungan dan seberapa kuat hubungan tersebut. Analisis korelasi dibagi menjadi dua yaitu *Unvariate analysis* adalah teknik statistik yang digunakan

untuk memahami karakteristik dasar dan *Multivariate Analysis* adalah tahapan untuk memahami hubungan kompleks antara beberapa variable. Dalam analisis korelasi, confusion matrix dapat digunakan untuk mengevaluasi hubungan antara variabel prediktif dan variabel target dengan menemukan pola kesalahan klasifikasi yang terjadi. Selain itu, dapat diukur kekuatan dan arah korelasi berdasarkan distribusi prediksi yang benar dan salah di antara berbagai kelas.

3.4 Data Preprocessing

Preprocessing adalah proses mengubah data agar lebih representatif dan memastikan bahwa data yang digunakan dalam analisis atau pembelajaran mesin bersih, konsisten, dan dalam format yang sesuai. Sebelum data dilatih, dataset akan diproses terlebih dahulu ke dalam format tertentu untuk meningkatkan kinerja model. Pada penelitian ini dilakukan *preprocessing* berupa [17]:

1) *Split test* atau pembagian data ke dalam beberapa bagian. Umumnya, data splitting memisahkan dua bagian, bagian pertama digunakan untuk mengevaluasi atau uji data dan data lainnya digunakan untuk melatih model. Tidak ada standar atau metrik yang jelas untuk membagi data splitting. Prosesnya dapat bergantung pada ukuran dataset asli atau jumlah prediktor dalam model prediktif. Data splitting memisahkan dataset dengan jumlah rasio data tertinggi dipakai untuk training. Rasio yang tepat tergantung pada data, namun secara umum rasio training-test 70:30 atau 80:20 merupakan rasio paling optimal untuk dataset berukuran kecil [24]. *Dataset* ini akan dibagi untuk *training data* dan *test data* dengan rasio sebesar 80:20. Tujuan utama dari pembagian ini adalah untuk memastikan model yang dibangun dapat generalisasi dengan baik ke data baru, menghindari *overfitting*, dan memberikan evaluasi yang akurat terhadap performa model.
2) *Encoding* fitur kategori adalah proses penting dalam analisis data dan *machine learning* yang mengubah data kategori (data yang berbentuk label atau kategori) menjadi format numerik atau kolom biner baru dengan nilai 1 atau 0, yang bisa diproses oleh algoritma *machine learning*. Model *machine learning* umumnya memerlukan data numerik, sehingga

encoding kategori diperlukan untuk membuat data kategorikal dapat digunakan dalam model.

3) Normalisasi data adalah proses penting dalam *preprocessing* data yang bertujuan untuk mengubah skala data menjadi rentang tertentu, biasanya antara 0 dan 1. Ini membantu algoritma *machine learning* bekerja lebih efisien dan efektif, terutama ketika fitur-fitur dalam dataset memiliki skala yang berbeda. Normalisasi data merupakan teknik untuk menskalakan fitur atau variabel sehingga berada dalam rentang yang sama atau memiliki distribusi yang seragam. Ini mempermudah proses pelatihan model dan memastikan bahwa fitur dengan skala besar tidak mendominasi fitur dengan skala kecil.
4) *Dimensional reduction* (reduksi dimensi) adalah teknik dalam analisis data dalam *machine learning* yang digunakan untuk mengurangi jumlah fitur (dimensi) dalam dataset dengan mempertahankan informasi yang penting. Teknik ini sangat berguna dalam situasi di mana data memiliki banyak fitur, yang sering kali menyebabkan masalah seperti *overfitting*, peningkatan waktu komputasi, dan kesulitan dalam visualisasi. Adapun salah satu metode reduksi dimensi ini yaitu PCA dengan mengubah fitur asli menjadi sejumlah fitur baru (*principal components*) yang merupakan kombinasi linier dari fitur asli. Komponen utama ini diurutkan berdasarkan variansi mereka, dengan komponen pertama memiliki variansi terbesar.

3.5 Model Selection

Tahap selanjutnya adalah memilih model. Model yang digunakan adalah algoritma *Random Forest*, karena *Random Forest* memiliki kemampuan dalam menangani data yang kompleks dan keberagaman fitur. sangat efektif dalam mengklasifikasikan kasus penyakit jantung. Kelebihannya termasuk kemampuan untuk menangani data yang sangat besar dan *noise* dan *missing value* [9]. Dengan kualitas algoritma yang dipilih, *Random Forest* diharapkan dapat menghasilkan hasil prediksi yang aktual, akurat, dan dapat diandalkan untuk memprediksi penyakit gagal jantung. *Random Forest* adalah algoritma pembelajaran mesin yang

diawasi yang banyak digunakan dalam masalah regresi dan klasifikasi. Sebagian besar waktu, bahkan tanpa menyetel *hyperparameter*, ia menghasilkan hasil yang sangat baik. Ini mungkin algoritma yang paling banyak digunakan karena kesederhanaannya. Ini membangun sejumlah pohon keputusan pada sampel yang berbeda dan kemudian mengambil suara mayoritas jika itu merupakan masalah klasifikasi [25]. Kualitas hebat lainnya dari algoritma luar biasa ini adalah ia juga dapat digunakan untuk pemilihan fitur. Kita dapat menggunakannya untuk mengetahui pentingnya fitur tersebut. Salah satu manfaat terbesar dari algoritma *random forest* adalah fleksibilitasnya. Kita dapat menggunakan algoritma ini untuk masalah regresi dan klasifikasi. Ini dapat dianggap sebagai algoritma yang berguna karena memberikan hasil yang lebih baik bahkan tanpa penyetelan *hyperparameter*. Selain itu, parameternya cukup jelas, mudah dipahami, dan jumlahnya juga tidak banyak [17] .

3.6 Hyperparameter Tuning

Penetapan *hyperparameter* sangat penting untuk mengoptimalkan kinerja algoritma *machine learning* (ML) apa pun. Hal ini disebabkan oleh fakta bahwa kinerja *random forest* sangat bergantung pada penetapan *hyperparameter*, sehingga meningkatkan kinerja model pembelajaran mesin. *Hyperparameter* merupakan variabel yang memengaruhi output dari sebuah model [18]. Proses *hyperparameter tunning* dengan *GridSearch* dapat membantu dalam mencari kombinasi *hyperparameter* yang optimal untuk model. *Hyperparameter* yang akan dipakai yaitu jumlah pohon (n_*estimators*), kedalaman maksimum (*max_depth*), dan jumlah minimum (*min_samples_leaf*).

3.7 Model Evaluation

C-Index adalah sebuah metrik yang digunakan untuk menilai seberapa baik kinerja model dalam memprediksi risiko atau *prognosis* pasien. Metrik ini membandingkan prediksi yang dibuat oleh model dengan waktu sebenarnya pasien bertahan hidup. Dalam analisis kelangsungan hidup, *C-Index* membantu untuk memahami seberapa baik model dalam mengurutkan pasien berdasarkan risiko

mereka. Jika model hanya menebak secara acak, hanya akan benar setengah dari waktu dan memiliki nilai *C-Index* sebesar 0.5. Namun, jika model sangat akurat dan selalu benar dalam prediksinya, nilai *C-Index*-nya akan menjadi 1.

3.8 Testing/Deployment

Dengan bantuan *library streamlit*, model yang telah dievaluasi akan diuji untuk memprediksi data baru secara langsung.