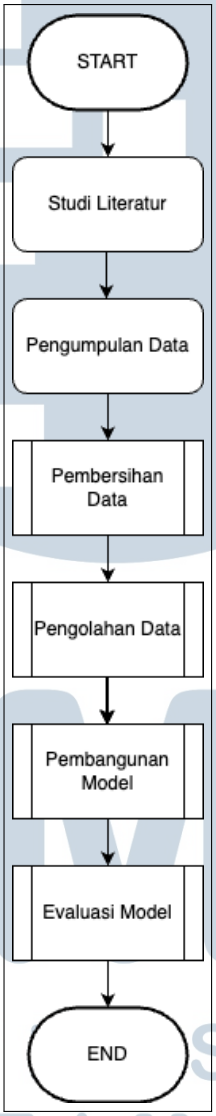
BAB 3
METODOLOGI PENELITIAN

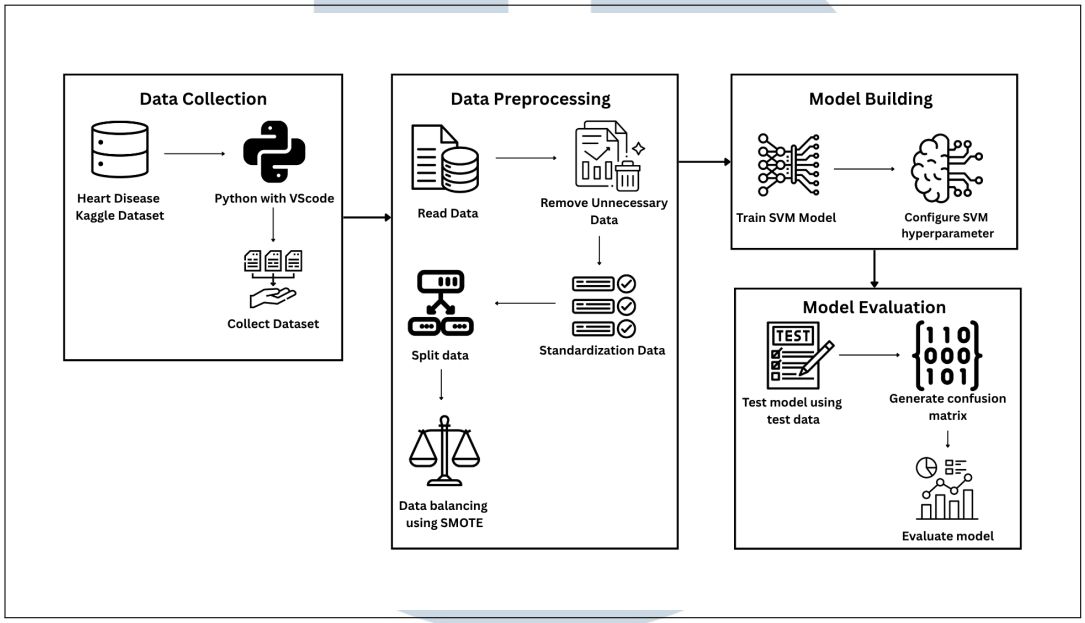
Pada metodologi penelitian berisi alur yang dilakukan selama penelitian berlangsung. Flowchart alur Penelitian yang dilakukan dapat pada Gambar 3.1.



Gambar 3.1. Alur Penelitian

Alur penelitian secara lengkap dapat dilihat pada Gambar 3.2. Pada gambar tersebut terdapat beberapa tahapan seperti pengumpulan data dari Kaggle, selanjutnya dilakukan pengolahan data dengan beberapa teknik yaitu Single Imputation, SMOTE, pembagian data latih dan uji, dan standardization data.

Setelah itu dilakukan hyperparameter tuning menggunakan GridSearch untuk menemukan model algoritma Super Vector Machine (SVM) yang terbaik. Tahap terakhir adalah evaluasi model dengan data uji, dengan mencari evaluasi metrik seperti accuracy, precision, recall, f1-score, dan AUC-ROC.



Gambar 3.2. Gambaran keseluruhan penelitian

3.1 Studi Literatur

Peneliti akan melakukan studi literatur untuk memperoleh ide, informasi, dan referensi yang relevan dalam mendukung penelitian ini. Studi literatur merupakan metode yang digunakan untuk menyelesaikan suatu persoalan dengan menelusuri hasil-hasil penelitian sebelumnya yang berkaitan. Melalui pendekatan ini, peneliti dapat menemukan landasan teori, mengetahui perkembangan penelitian terdahulu, serta mengidentifikasi celah penelitian yang bisa dikembangkan lebih lanjut.

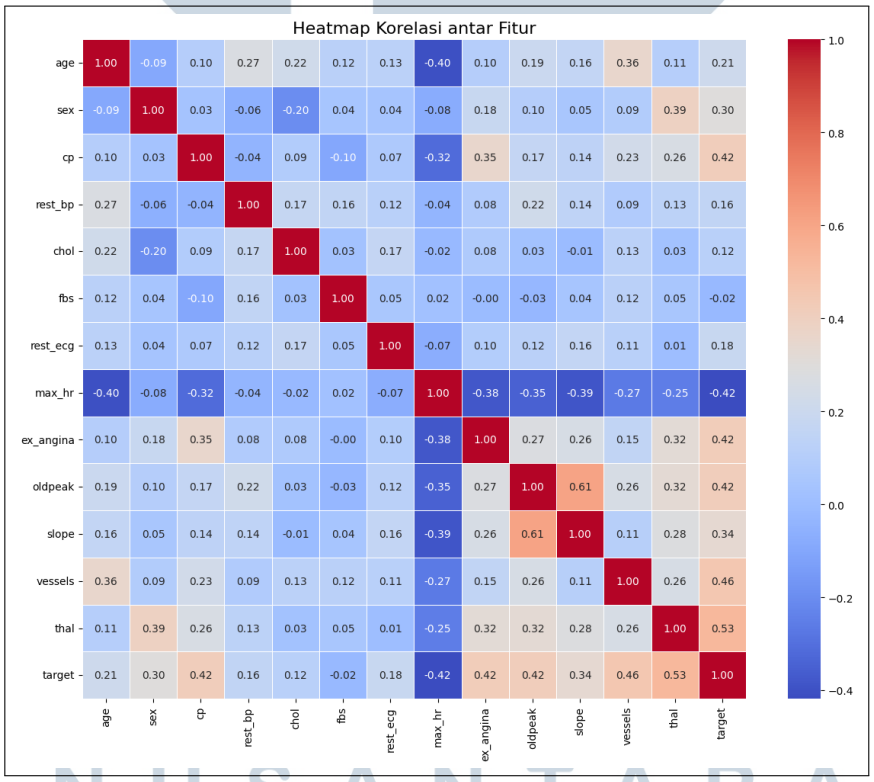
Sumber yang digunakan dalam studi literatur meliputi buku-buku ilmiah, jurnal terakreditasi, skripsi, tesis, prosiding, dan laporan penelitian. Selain itu, peneliti juga memanfaatkan sumber digital seperti YouTube dan berbagai website edukatif atau teknologi untuk mendapatkan solusi praktis dalam pengembangan model dan penulisan skripsi. Situs seperti Google Scholar, ResearchGate, Medium, GitHub, dan dokumentasi resmi dari tools yang digunakan menjadi referensi tambahan yang memperkaya hasil studi literatur ini.

3.2 Pengumpulan Data

Data yang digunakan pada Gambar 3.3 berasal dari UCI Machine Learning Repository, dataset tersebut berisi berbagai indikator kesehatan dan faktor risiko yang berkaitan dengan penyakit jantung. Data ini memiliki informasi terkait pasien yaitu age (umur), sex (jenis kelamin), chest pain type (jenis nyeri dada), resting blood pressure (tekanan darah saat istirahat), serum cholestoral (kadar kolesterol dalam darah), fasting blood sugar (kadar gula darah saat puasa), resting electrocardiographic results (hasil elektrokardiogram saat istirahat), maximum heart rate achieved (detak jantung maksimum yang dicapai), exercise induced angina (nyeri dada akibat olahraga), oldpeak (depresi ST yang diinduksi oleh olahraga dibandingkan saat istirahat), slope (kemiringan segmen ST pada EKG), number of major vessels (jumlah pembuluh darah utama yang terlihat melalui fluoroskopi), dan thal (hasil tes thalium scan jantung). Adapun fitur target merupakan label dari data tersebut, yang menunjukkan kondisi pasien: nilai 1 menunjukkan pasien tidak memiliki penyakit jantung (negatif), sedangkan nilai 2 menunjukkan pasien positif menderita penyakit jantung.. Data ini yang digunakan untuk training model dan melakukan klasifikasi untuk mendeteksi penyakit jantung. Data ini dapat diakses pada situs web https://archive.ics.uci.edu/dataset/145/statlog+heart . Data ini berasal dari tahun 1988 dan terdiri dari empat basis data: Cleveland, Hungaria, Swiss, dan Long Beach V. Setelah diakses, data tersebut telah siap untuk ke langkah selanjutnya yaitu pengolahan data. Dataset tersebut terdiri dari 270 row dan 14 features. Korelasi pada setiap fitur digambarkan menggunakan heatmap yang dapat dilihat pada Gambar 3.4. Korelasi setiap fitur pada target label cenderung sangat berpengaruh secara positif maupun negatif. Warna merah menunjukkan hubungan secara positif dan warna biru menggambarkan hubungan negatif antar fitur.

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
58	0	3	150	283	1	0	162	0	1	2	0	2	1

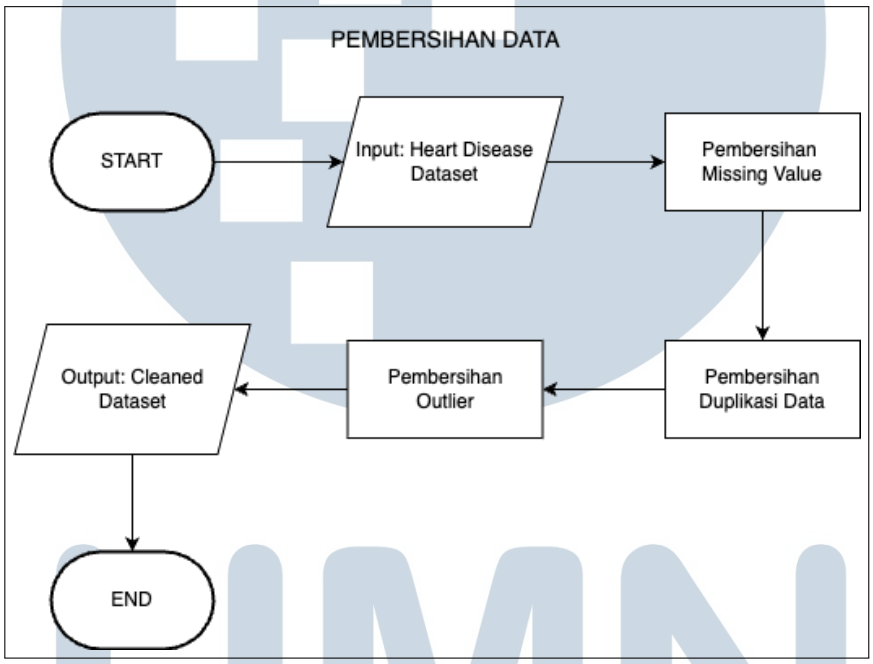
Gambar 3.3. Dataset Heart Disease Kaggle



Gambar 3.4. Korelasi per fitur

3.3 Pembersihan Data

Pada Gambar 3.5 menampilkan flowchart yang menggambarkan tahapan-tahapan dalam proses pembersihan data. Proses ini bertujuan untuk pembersihan data untuk memastikan kualitas dan reliabilitas data yang akan digunakan dalam analisis. Pembersihan data dilakukan melalui tiga tahapan utama, yaitu deteksi missing value, deteksi duplikasi data, dan deteksi outlier. Ketiga tahapan ini bertujuan untuk mengidentifikasi dan menangani potensi masalah dalam data yang dapat memengaruhi hasil model atau analisis.



Gambar 3.5. Flowchart pembersihan data

3.3.1 Pembersihan Missing Value

Missing value atau data yang hilang merupakan permasalahan umum dalam proses pengolahan data yang dapat memengaruhi validitas hasil analisis. Oleh karena itu, langkah pertama dalam pembersihan data adalah mengidentifikasi nilai-nilai yang hilang pada dataset. Proses deteksi missing value dilakukan dengan mengevaluasi setiap atribut untuk mengetahui keberadaan nilai kosong, null, atau nilai yang tidak terisi secara semestinya.

Hasil dari deteksi ini akan menjadi dasar untuk menentukan perlakuan selanjutnya terhadap data yang hilang, seperti imputasi, penghapusan baris, atau metode lain yang sesuai dengan karakteristik dataset dan tujuan analisis.

3.3.2 Pembersihan Duplikasi Data

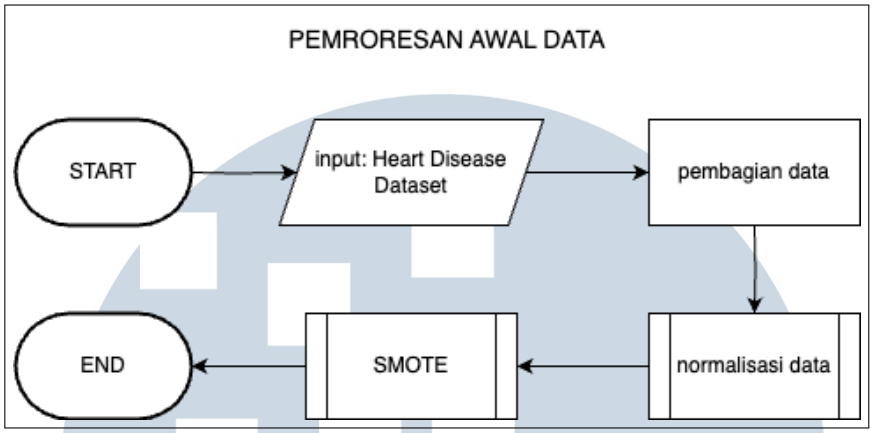
Duplikasi data terjadi ketika terdapat baris atau entri yang identik lebih dari satu kali dalam dataset. Keberadaan data duplikat dapat menyebabkan distorsi dalam analisis, menghasilkan bobot berlebih pada informasi yang seharusnya hanya dihitung satu kali, dan menurunkan akurasi model yang dibangun. Deteksi dilakukan secara menyeluruh terhadap dataset untuk memastikan bahwa data yang digunakan bersih dari pengulangan yang tidak diinginkan sebelum masuk ke tahap pemodelan atau analisis lanjutan.

3.3.3 Pembersihan Outiler

Outlier merupakan data yang nilainya sangat berbeda dari sebagian besar data lainnya. Nilai-nilai yang menyimpang ini bisa menyebabkan hasil analisis menjadi tidak akurat karena dapat memengaruhi perhitungan statistik dan kinerja model. Jika tidak ditangani dengan baik, outlier bisa membuat model salah memahami pola dalam data. Oleh karena itu, mendeteksi dan menangani outlier adalah langkah penting agar hasil analisis lebih terpercaya dan representatif. Proses ini bertujuan untuk memastikan bahwa data yang dianalisis mewakili pola umum dan tidak terdistorsi oleh nilai yang tidak wajar. Penelitian ini memakai *interquartile range* untuk metode pembersigan outlier.

3.4 Pemrosesan Awal Data

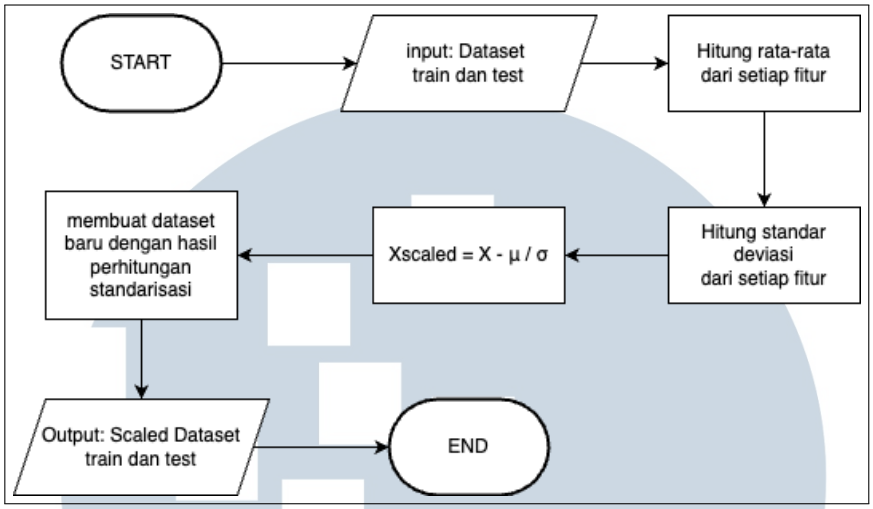
Pada Gambar 3.6 menampilkan flowchart yang menggambarkan tahapan-tahapan dalam proses pengolahan data. Tahapan tersebut meliputi pemilahan fitur yang relevan dari dataset, pembagian data menjadi data latih dan data uji, proses normalisasi (scaling) untuk menyamakan skala fitur, serta penerapan metode SMOTE untuk mengatasi ketidakseimbangan kelas. Pengolahan data (preprocessing) ini merupakan langkah krusial guna meningkatkan kualitas dan akurasi dalam pembangunan model.



Gambar 3.6. *Flowchart* pemrosesan awal data

3.4.1 Standarisasi Data

Pada Gambar 3.7 menampilkan flowchart yang menggambarkan tahapan-tahapan dalam proses standarisasi data. Standarisasi data adalah proses transformasi data agar memiliki skala yang seragam, biasanya dengan cara mengubah setiap fitur agar memiliki nilai rata-rata nol dan deviasi standar satu. Tujuan utama dari standarisasi adalah untuk menghindari dominasi fitur dengan skala besar terhadap model, terutama pada algoritma yang sensitif terhadap skala data seperti K-Nearest Neighbors (KNN) dan Support Vector Machines (SVM). Salah satu cara untuk melakukan standarisasi adalah dengan menggunakan StandardScaler, yang merupakan metode yang mengubah setiap nilai data pada fitur dengan cara mengurangi nilai rata-rata (mean) dan membaginya dengan deviasi standar (standard deviation) dari fitur tersebut. Dengan menggunakan StandardScaler, setiap fitur akan memiliki rata-rata nol dan deviasi standar satu, sehingga tidak ada fitur yang mendominasi model berdasarkan skala data yang lebih besar. Hal ini memungkinkan model untuk mempelajari pola dari data dengan lebih efisien, meningkatkan akurasi prediksi, dan memastikan bahwa model dapat menangani data dengan berbagai skala secara adil.



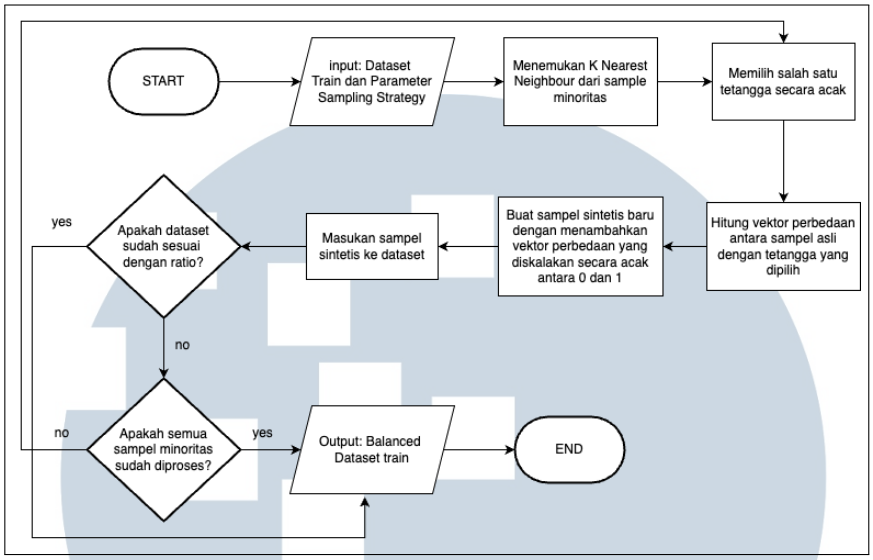
Gambar 3.7. *Flowchart* standarisasi data

3.4.2 Pembagian Data

Tahapan pembagian data adalah proses memisahkan dataset menjadi dua bagian utama yaitu data latih (training data) dan data uji (test data). Data latih digunakan untuk melatih model, sementara data uji digunakan untuk mengevaluasi performa model setelah dilatih. Pembagian data ini bertujuan untuk menghindari overfitting, yaitu ketika model terlalu menyesuaikan diri dengan data latih sehingga tidak dapat menggeneralisasi dengan baik pada data baru. Biasanya, data dibagi dengan proporsi tertentu, seperti 80% untuk data latih dan 20% untuk data uji. 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, dapat memastikan bahwa model dapat memberikan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya.

3.4.3 SMOTE(Synthetic Minority Oversampling Technique)

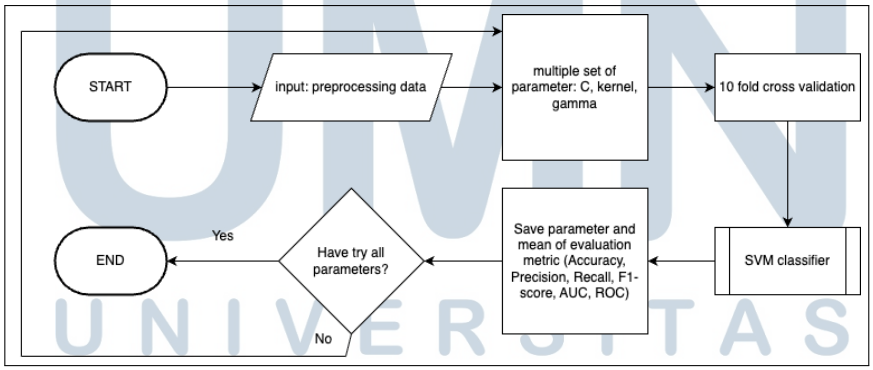
Pada gambar 3.8 merupakan flowchart yang menggambar cara kerja SMOTE. SMOTE (Synthetic Minority Oversampling Technique) adalah suatu teknik yang membuat kelas minoritas menjadi seimbang dengan kelas mayoritas. Teknik ini membuat data sintetis baru dari tetangga terdekat menggunakan euclidean distance.



Gambar 3.8. *Flowchart* SMOTE

3.5 Pembangunan Model

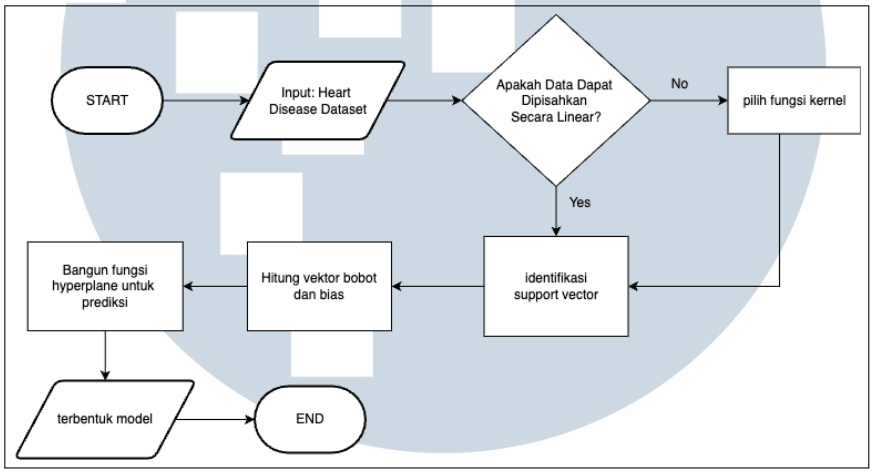
Padagambar 3.9 merupakan flowchart untuk pembangunan model dari algoritma SVM. Algoritma yang dipakai adalah SVC (Support Vector Classifier) yang diimplementasikan bertujuan untuk melakukan prediksi untuk mendeteksi penyakit jantung berdasarkan dataset statlog(Heart) UCI Machine Learning Repository.



Gambar 3.9. *Flowchart* pembangunan model

Pada langkah pertama, perlu diberikan inisiasi beberapa hyperparameter seperti C, kernel, dan gamma. Parameter C akan mengontrol tingkat penalti terhadap kesalahan klasifikasi, kernel menentukan jenis fungsi yang digunakan untuk memetakan data ke ruang fitur yang lebih tinggi, dan gamma mengatur seberapa jauh pengaruh satu titik data terhadap lainnya. Setelah inisiasi

parameter, dilakukan 10-fold cross-validation. Cross-validation sangat penting untuk mengevaluasi model secara lebih robust, terutama ketika dataset yang digunakan tidak terlalu besar. Setelah itu, parameter-parameter tersebut dicoba satu per satu menggunakan GridSearchCV untuk menemukan kombinasi terbaik dari C, kernel, dan gamma, yang dikenal dengan istilah hyperparameter tuning. Setelah mendapatkan model SVM dengan parameter terbaik, model ini digunakan untuk memprediksi data testing dan mengukur performanya dalam mendeteksi penyakit jantung.



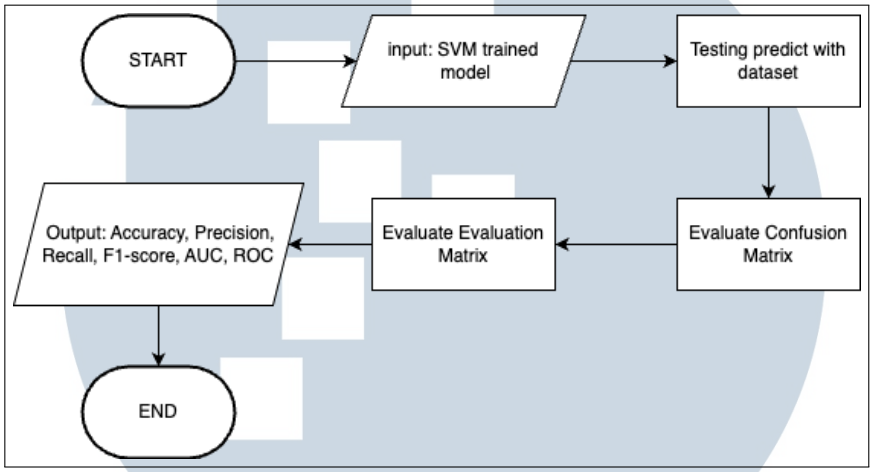
Gambar 3.10. *Flowchart* algoritma Support Vector Machine

Proses dimulai dengan memasukkan dataset penyakit jantung sebagai data latih. Setelah itu, sistem mengevaluasi apakah data dapat dipisahkan secara linear. Jika data tidak linear, maka digunakan fungsi kernel untuk mengubah data ke dalam ruang berdimensi lebih tinggi agar dapat dipisahkan secara optimal. Langkah berikutnya adalah mengidentifikasi titik-titik data penting yang disebut support vector, lalu menghitung parameter bobot dan bias yang menentukan garis atau bidang pemisah antar kelas. Berdasarkan parameter tersebut, sistem membentuk fungsi pemisah yang digunakan untuk melakukan prediksi. Setelah model selesai dibangun, proses pelatihan berakhir dan model siap digunakan untuk klasifikasi data baru.

3.6 Evaluasi Model

Pada gambar 3.11 merupakan flowchart evaluasi model. Flowchart tersebut berisi tahapan untuk menguji kinerja dari model yang telah di train sebelumnya. Evaluasi model yang dilakukan untuk mengetahui performa model tersebut adalah

confusion matrix, accuracy, precision, recall, F1-score, Area Under Curve (AUC) dan Receiver Operating Characteristic (ROC). Performa yang diukur berdasarkan hasil dari klasifikasi model terhadap data test yang dibandingkan dengan data aslinya.



Gambar 3.11. *Flowchart* evaluasi model

3.7 Dokumentasi

Dokumentasi yang dilakukan berupa penulisan laporan. Penulisan laporan berfungsi untuk mendokumentasikan semua tahap penelitian dari awal hingga akhir dengan mematuhi standar dan temuan yang diperoleh. Laporan juga mencakup rincian mengenai tahapan penelitian, implementasi algoritma, hasil temuan dari penelitian, serta dokumentasi lengkap tentang proses penelitian yang telah dilakukan.