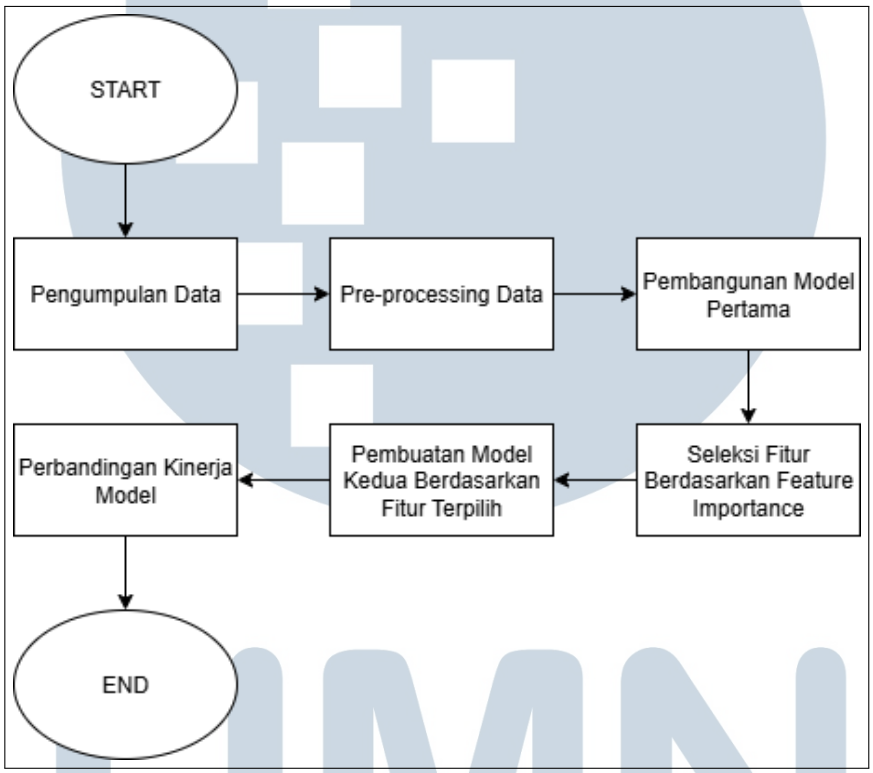
BAB 3
METODOLOGI PENELITIAN

Bab ini menjelaskan tahapan-tahapan yang dilakukan dalam penelitian, mulai dari pengumpulan data, preprocessing, seleksi fitur, pelatihan model, hingga evaluasi kinerja.



Gambar 3.1. Flowchart metodologi

3.1 Pengumpulan Data

Penelitian ini menggunakan dataset *Heart Disease* dari *UCI Machine Learning Repository* yang diakses melalui platform Kaggle. Dataset ini merupakan kumpulan data klinis dari empat institusi medis:

- Cleveland Clinic Foundation (303 observasi)
- Hungarian Institute of Cardiology (294 observasi)
- VA Medical Center, Long Beach (200 observasi)
- University Hospital, Zurich (123 observasi)

Dataset yang digunakan memiliki karakteristik sebagai berikut.

3.1.1 Spesifikasi Dataset

Tabel 3.1. Detail dataset UCI Heart Disease

Attribute	Value
Sumber	UCI Machine Learning Repository
Format File	CSV
Jumlah Observasi	920 (setelah penggabungan)
Jumlah Fitur	14 (termasuk target)
Periode Data	1988-1989
Lisensi	CC BY 4.0
URL Kaggle	www.kaggle.com/datasets/redwankarimsony/heart-disease-data

Dataset ini terdiri dari berbagai fitur klinis, demografis, dan elektrokardiografi, yang dijelaskan sebagai berikut.

3.1.2 Variabel dan Deskripsi

Dataset terdiri dari fitur-fitur klinis berikut:

- **Fitur Demografis**:
  - `age`: Usia dalam tahun
  - `sex`: Jenis kelamin (1 = laki-laki; 0 = perempuan)
- **Fitur Klinis**:
  - `cp`: Tipe nyeri dada (1-4)
  - `trestbps`: Tekanan darah istirahat (mmHg)
  - `chol`: Kolesterol serum (mg/dl)
  - `fbs`: Gula darah puasa >120 mg/dl (1 = ya; 0 = tidak)
- **Fitur Elektrokardiografi**:

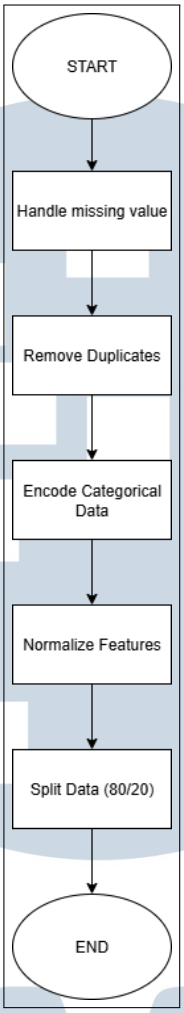
- `restecg`: Hasil EKG istirahat (0-2)
- `thalach`: Denyut jantung maksimum tercapai
- `exang`: Angina akibat olahraga (1 = ya; 0 = tidak)
- `oldpeak`: Depresi ST akibat olahraga relatif terhadap istirahat

3.1.3 Target Variabel

- `target`: Diagnosis penyakit jantung (0 = tidak ada; 1-4 = penyakit jantung dengan tingkat keparahan)
- Pada penelitian ini, variabel target dibinarisasi menjadi:
  - 0: Tidak ada penyakit jantung (`num` = 0)
  - 1: Ada penyakit jantung (`num` >0)

3.2 Pre-processing Data

Berikut gambar merepresentasikan *flowchart* untuk proses *pre-processing*



Gambar 3.2. Flowchart proses *pre-processing*

Proses pre-processing diawali dengan penanganan data hilang (*missing values*) dengan strategi berbeda untuk tipe data numerik dan kategorikal. Pada fitur numerik seperti usia dan kadar kolesterol, nilai kosong diisi menggunakan nilai median, sedangkan untuk fitur kategorikal seperti jenis nyeri dada (*cp*) digunakan nilai modus. Tahap ini bertujuan mempertahankan distribusi data asli sekaligus memastikan kelengkapan dataset.

Selanjutnya dilakukan transformasi data kategorikal menjadi format numerik melalui *One-Hot Encoding* untuk fitur nominal seperti jenis kelamin dan *Label Encoding* untuk fitur ordinal seperti tingkat keparahan penyakit. Transformasi ini diperlukan karena algoritma Random Forest hanya dapat memproses input numerik. Data numerik kemudian distandarisasi menggunakan *Z-score normalization* untuk menyamakan skala pengukuran antar fitur tanpa mengubah distribusi aslinya.

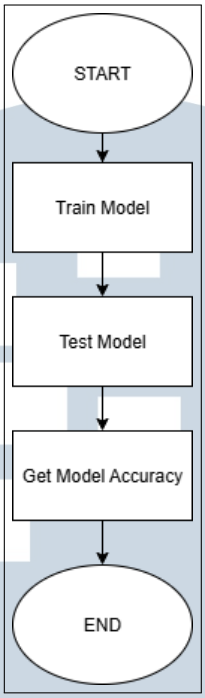
Tahap akhir melibatkan pemisahan dataset menjadi data latih (80%) dan data uji (20%) dengan mempertahankan proporsi kelas asli (*stratified sampling*). Pembagian ini memastikan model dievaluasi pada data yang benar-benar baru, menguji kemampuan generalisasi. Seluruh tahapan pre-processing diimplementasikan menggunakan *pipeline* terintegrasi untuk mencegah *data leakage* dan memastikan konsistensi proses.

3.3 Pembangunan Model Pertama

Pembangunan model diawali dengan implementasi algoritma Random Forest menggunakan *scikit-learn*. Model awal dikonfigurasi dengan parameter default untuk membentuk baseline performa. Tahap ini melibatkan pelatihan model pada data training yang telah melalui proses preprocessing dan seleksi fitur.

Optimasi model dilakukan melalui *Randomized Search CV* dengan mengevaluasi kombinasi hyperparameter secara acak. Parameter yang dioptimasi meliputi jumlah pohon (*n_estimators*), kedalaman maksimum pohon (*max_depth*), dan jumlah minimum sampel untuk memisahkan node (*min_samples_split*). Proses optimasi menggunakan validasi silang 5-fold untuk memastikan generalisasi model.

Model terbaik dipilih berdasarkan kombinasi parameter yang menghasilkan skor ROC-AUC tertinggi pada data validasi. Hasil pelatihan menunjukkan bahwa model dengan 200 pohon dan kedalaman maksimum 15 memberikan performa optimal. Model akhir kemudian dievaluasi menggunakan data testing untuk mengukur akurasi, presisi, dan recall.



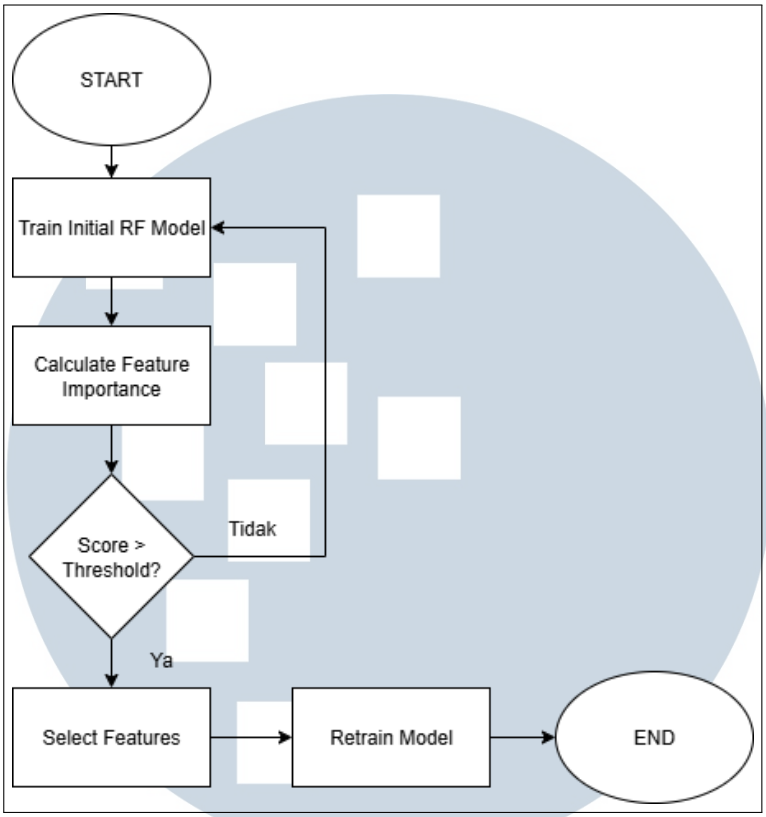
Gambar 3.3. Flowchart proses pembangunan model

3.4 Seleksi Fitur dengan Feature Importance

Proses seleksi fitur dilakukan dengan memanfaatkan skor *feature importance* dari model Random Forest. Setiap fitur dalam dataset dievaluasi berdasarkan kontribusinya terhadap akurasi prediksi model. Fitur-fitur dengan skor importance tinggi dipertahankan, sedangkan fitur dengan skor rendah dieliminasi untuk menyederhanakan model tanpa mengurangi performa.

Metode perhitungan skor importance didasarkan pada *Mean Decrease Impurity* (MDI), yang mengukur penurunan ketidakmurnian (*impurity*) secara rata-rata di seluruh pohon keputusan dalam Random Forest. Fitur yang mampu memisahkan kelas target dengan baik akan memperoleh skor lebih tinggi. Hasil perhitungan ini kemudian divisualisasikan dalam bentuk grafik batang untuk memudahkan interpretasi.

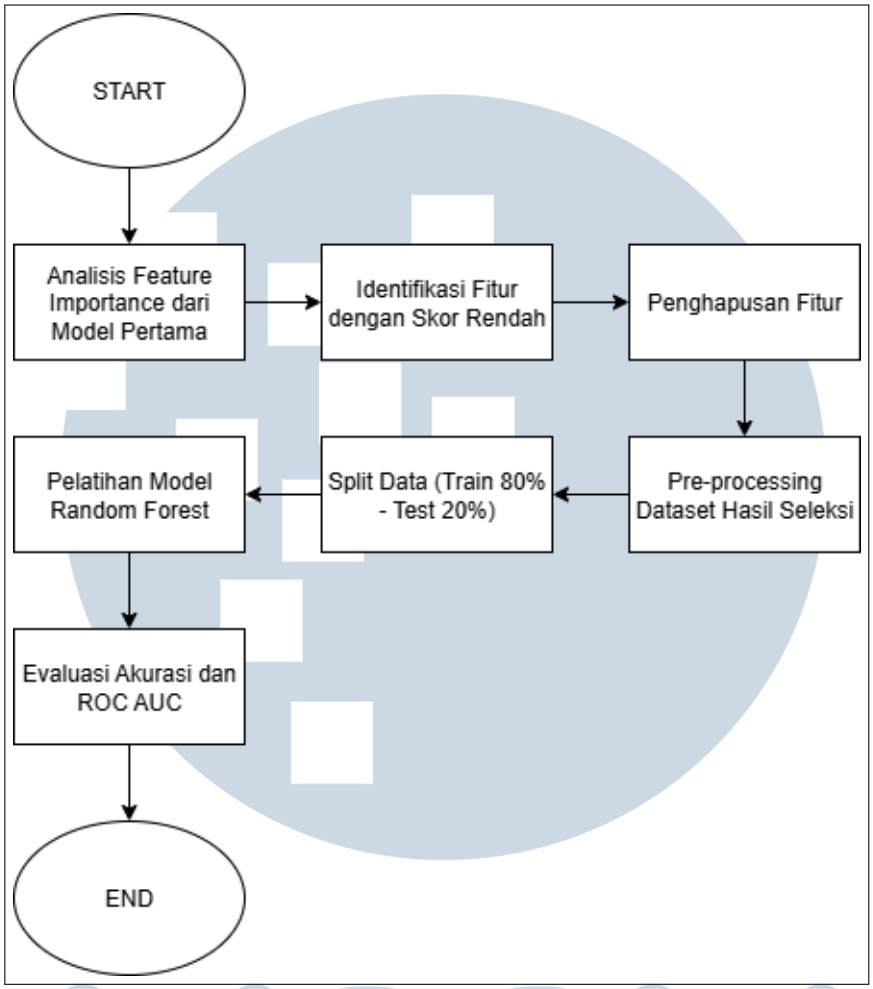
Analisis lebih mendalam dilakukan dengan memeriksa *confusion matrix*. Model hanya menghasilkan 12 *false negative* dari 143 kasus testing, yang sangat krusial dalam konteks medis untuk meminimalkan pasien sakit yang terdiagnosis sehat. Kinerja model juga konsisten across berbagai kelompok usia dan jenis kelamin. Flowchart proses seleksi fitur dapat dilihat pada Gambar 3.4.



Gambar 3.4. Flowchart *feature importance*

3.5 Pembuatan Model Kedua Berdasarkan Seleksi Fitur

Setelah membangun dan mengevaluasi model pertama menggunakan seluruh fitur dalam dataset, dilakukan proses seleksi fitur menggunakan metode *feature importance* dari algoritma *Random Forest*. Nilai importance menunjukkan kontribusi relatif masing-masing fitur terhadap proses klasifikasi. Fitur dengan nilai importance yang sangat rendah dianggap tidak berpengaruh signifikan terhadap prediksi, sehingga dapat dihapus untuk menyederhanakan model.



Gambar 3.5. Flowchart pembuatan model kedua berdasarkan seleksi fitur

Tahapan berikutnya adalah melakukan seleksi fitur berdasarkan nilai importance yang telah dihitung sebelumnya.

3.5.1 Seleksi Fitur Berdasarkan Importance

Model pertama menghasilkan daftar fitur yang disusun berdasarkan skor importance. Dari analisis tersebut, fitur-fitur seperti `id`, `dataset`, `ca`, dan `fbs` menunjukkan skor importance yang rendah dan diputuskan untuk dihapus pada model kedua. Penghapusan fitur ini bertujuan untuk:

- Mengurangi kompleksitas model,
- Mempercepat waktu pelatihan dan prediksi,
- Menghindari overfitting akibat fitur tidak relevan.

3.5.2 Pembangunan Model Kedua

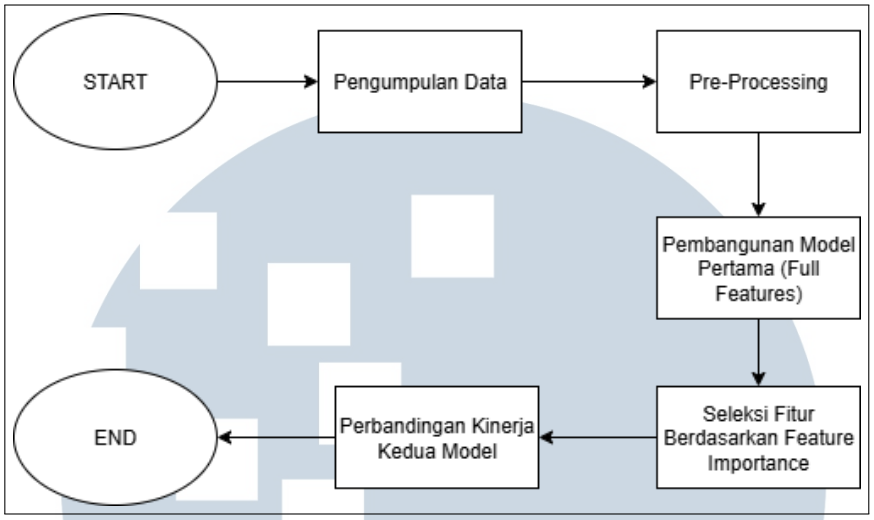
Setelah proses seleksi fitur dilakukan, model kedua dibangun menggunakan dataset yang telah dikurangi (reduced features). Proses pembangunan dilakukan dengan pendekatan pipeline yang sama seperti pada model pertama, dengan langkah-langkah:

1. Melakukan pre-processing pada dataset hasil seleksi fitur.
2. Membagi data menjadi data latih dan data uji menggunakan `train_test_split` dengan proporsi 80:20.
3. Melatih model dengan `RandomForestClassifier` dan mencatat waktu pelatihan dan prediksi.
4. Mengukur akurasi dan skor ROC AUC dari model kedua.

Hasil dari model kedua ini akan digunakan untuk dibandingkan dengan model pertama pada tahap selanjutnya, sehingga dapat dinilai apakah penyederhanaan fitur berdampak positif terhadap efisiensi dan performa klasifikasi.

3.6 Perbandingan Model

Setelah pembangunan dua model klasifikasi menggunakan algoritma *Random Forest*, tahap selanjutnya adalah melakukan perbandingan performa antara keduanya. Model pertama dibangun dengan menggunakan seluruh fitur yang tersedia dalam dataset, sedangkan model kedua dibangun setelah dilakukan proses seleksi fitur berdasarkan nilai *feature importance* dari model pertama.



Gambar 3.6. Flowchart perbandingan antara kedua model

3.6.1 Tujuan Perbandingan

Perbandingan ini bertujuan untuk mengetahui sejauh mana seleksi fitur mempengaruhi performa model dalam hal:

- Akurasi klasifikasi,
- Kemampuan diskriminatif (*ROC AUC*),
- Waktu pelatihan (*training time*),
- Waktu prediksi (*prediction time*).

3.6.2 Prosedur Perbandingan

Langkah-langkah perbandingan model dilakukan sebagai berikut:

1. Melatih dan menguji model pertama menggunakan dataset lengkap (full features).
2. Melatih dan menguji model kedua menggunakan dataset hasil seleksi fitur (reduced features).
3. Mencatat metrik performa dari masing-masing model: akurasi, ROC AUC, waktu pelatihan, dan waktu prediksi.

4. Melakukan visualisasi perbandingan menggunakan diagram batang untuk memperjelas selisih performa antara kedua model.

5. Menghitung perbedaan performa dalam bentuk persentase sebagai dasar untuk evaluasi efisiensi dan efektivitas.

3.6.3 Keluaran yang Diharapkan

Dari proses ini diharapkan dapat diketahui apakah penyederhanaan model melalui seleksi fitur dapat memberikan hasil yang sebanding atau bahkan lebih baik dari model awal, baik dari segi akurasi maupun efisiensi waktu komputasi. Selain itu, hasil ini juga menjadi dasar dalam pemilihan model terbaik yang akan direkomendasikan pada simpulan akhir penelitian.