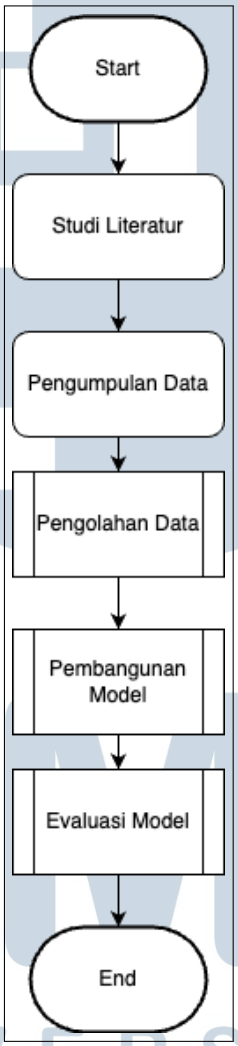
BAB 3
METODOLOGI PENELITIAN

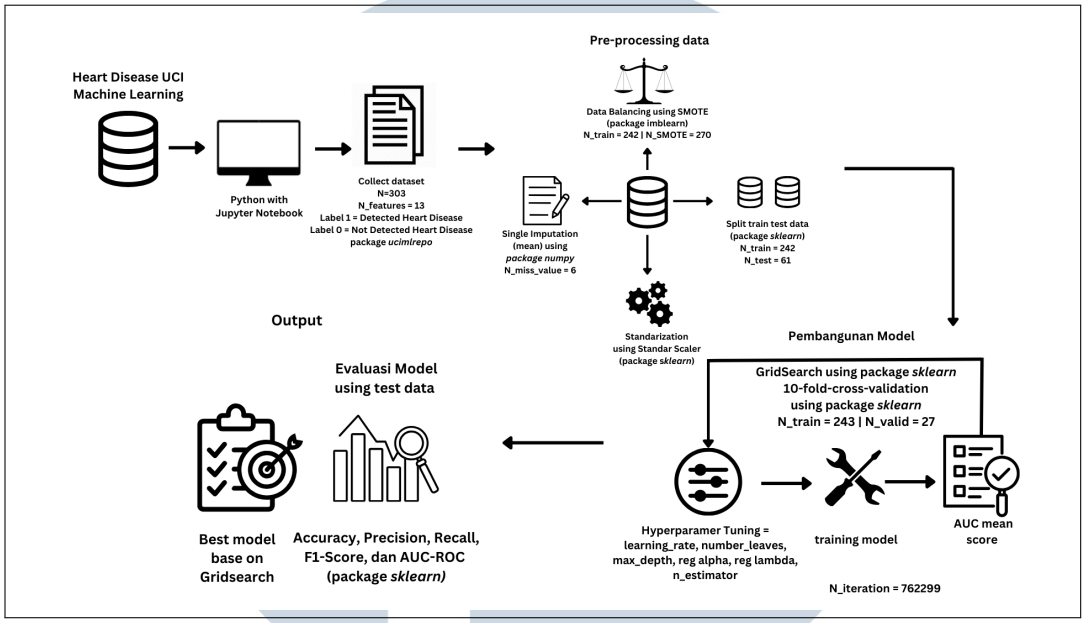
Pada metodologi penelitian berisi alur yang dilakukan selama penelitian berlangsung. Flowchart alur Penelitian yang dilakukan dapat pada Gambar 3.1.



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 *UCI Machine Learning*, 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 LightGBM yang terbaik. Tahap terakhir adalah

evaluasi model dengan data uji, dengan mencari avaluasi 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 mendapatkan berbagai ide atau sumber referensi dalam penelitian. Studi literatur adalah sebuah cara untuk menyelesaikan sebuah persoalan dengan menelusuri penelitian yang sudah pernah dibuat sebelumnya. Beberapa hal yang layak digunakan pada studi literatur yaitu buku-buku terpercaya, jurnal terakreditasi, skripsi, dan lain sebagainya.

3.2 Pengumpulan Data

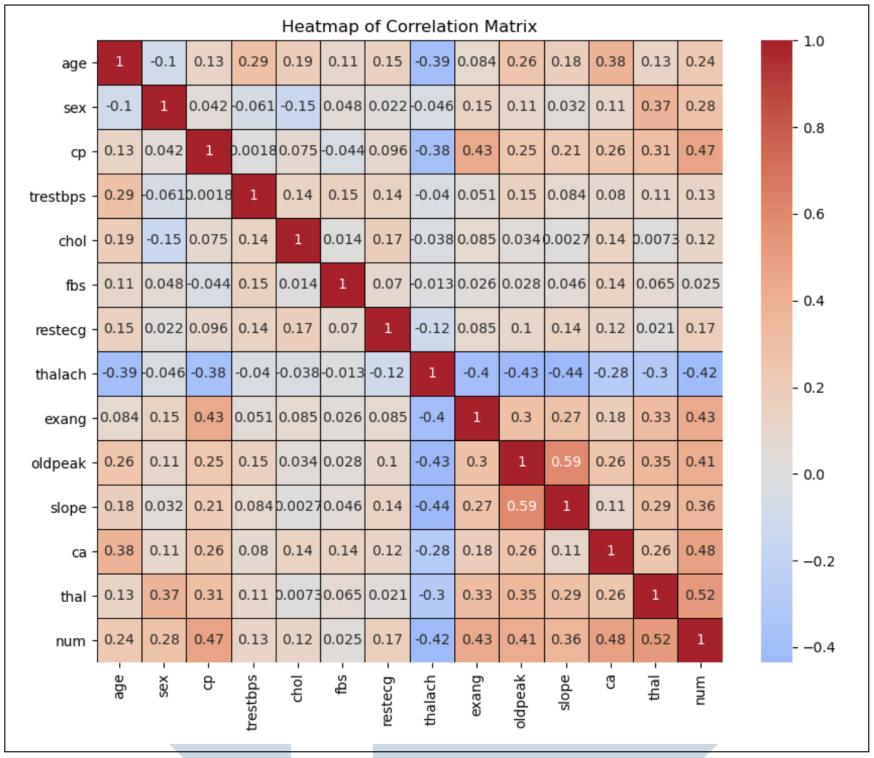
Data yang digunakan seperti pada gambar 3.3 berasal dari *UCI Repository*, data tersebut berisi pasien yang diperiksa untuk mendeteksi penyakit jantung pada rumah sakit di Cleveland. Data ini memiliki informasi terkait pasien yaitu umur, jenis kelamin, tipe nyeri dada, tekanan darah istirahat, kadar kolesterol, kadar gula darah puasa, elektrokardiogram istirahat, detak jantung maksimal, angina yang diinduksi oleh olahraga, depresi ST setelah olahraga, kemiringan segmen ST, jumlah pembuluh darah utama yang diwarnai dengan *flourosopy*, dan jenis kelainan *thalassemia*. Data ini yang digunakan untuk

training model dan melakukan klasifikasi untuk mendeteksi penyakit jantung. Data ini dapat diakses pada situs web *UCI Machine Learning Repository* yaitu https://archive.ics.uci.edu/dataset/45/heart+disease. Data tersebut dapat diakses dengan melakukan *install package* ***ucimlrepo*** lalu tinggal gunakan dengan **fetch_ucirepo(id=45)** pada code. Data tersebut terakhir diakses pada bulan Mei 2024. Setelah diakses, data tersebut telah siap untuk ke langkah selanjutnya yaitu pengolahan data. Dataset tersebut terdiri dari 303 row dan 13 *features*. 13 *features* tersebut, sangat umum digunakan dalam penelitian untuk mendeteksi dini penyakit jantung [10, 11, 12]. 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. Dataset yang telah diproses hanya terdapat pada 1 sumber yaitu Cleveland sehingga terdapat limitasi pada jumlah data.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	63	1	1	145	233	1	2	150	0	2.3	3	0.0	6.0	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3.0	3.0	2
2	67	1	4	120	229	0	2	129	1	2.6	2	2.0	7.0	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0.0	3.0	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0.0	3.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	45	1	1	110	264	0	0	132	0	1.2	2	0.0	7.0	1
299	68	1	4	144	193	1	0	141	0	3.4	2	2.0	7.0	2
300	57	1	4	130	131	0	0	115	1	1.2	2	1.0	7.0	3
301	57	0	2	130	236	0	2	174	0	0.0	2	1.0	3.0	1
302	38	1	3	138	175	0	0	173	0	0.0	1	NaN	3.0	0

303 rows × 14 columns

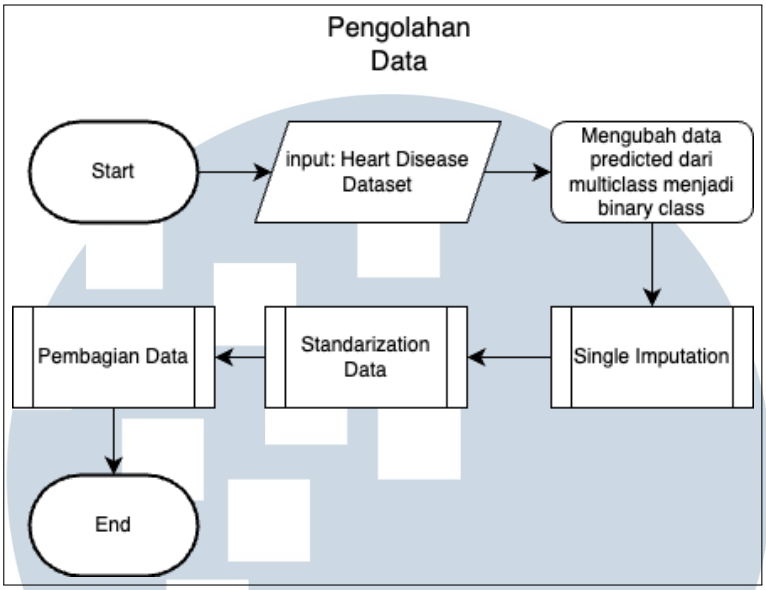
Gambar 3.3. Dataset Heart Disease UCI Machine Learning Repository



Gambar 3.4. *Corellation* per *feature*

3.3 Pengolahan Data

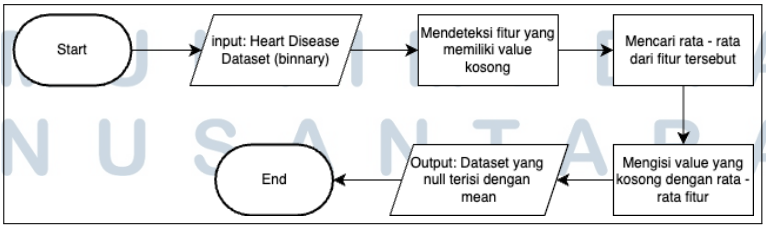
Pada gambar 3.5 merupakan *flowchart* yang berisi tahapan untuk melakukan pengolahan terhadap data. Tahapan tersebut meliputi pengisian nilai yang *null*, standarisasi pada *features* di dataset, dan pembagian data. Pengolahan data (*Pre-processing data*) sangat penting untuk membuat data menjadi lebih akurat dalam perancangan model. Langkah - langkah *pre-processing* sebagai berikut.



Gambar 3.5. *Flowchart* Pengolahan Data

3.3.1 *Single Imputation*

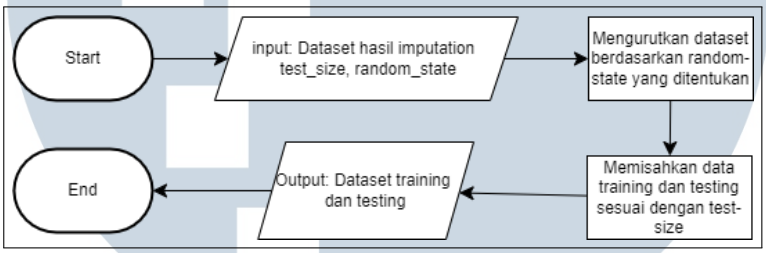
Pada gambar 3.6 merupakan *flowchart* yang menggambarkan tahapan dalam mengisi value yang null pada dataset. Pengisian value tersebut menggunakan teknik *single imputation*. Teknik *single imputation* dipilih untuk menghasilkan angka spesifik. Pada *single imputation* menggunakan informasi sebanyak mungkin dari data observasi untuk memprediksi nilai yang hilang. Metode *single imputation* yang dipilih adalah *mean imputation*. *Mean imputation* dipilih karena memiliki keakuratan dan error yang lebih baik dibandingkan metode seperti median dan mode. *Mean imputation* tersebut mengisi nilai yang null dengan nilai perhitungan rata-rata dari nilai tersebut [35, 36]. Perhitungan *mean* tersebut berdasarkan masing - masing per *features*, metode tersebut lebih baik dibandingkan menggunakan *mean* secara global [37].



Gambar 3.6. *Flowchart Single Imputation*

3.3.2 Pembagian Data

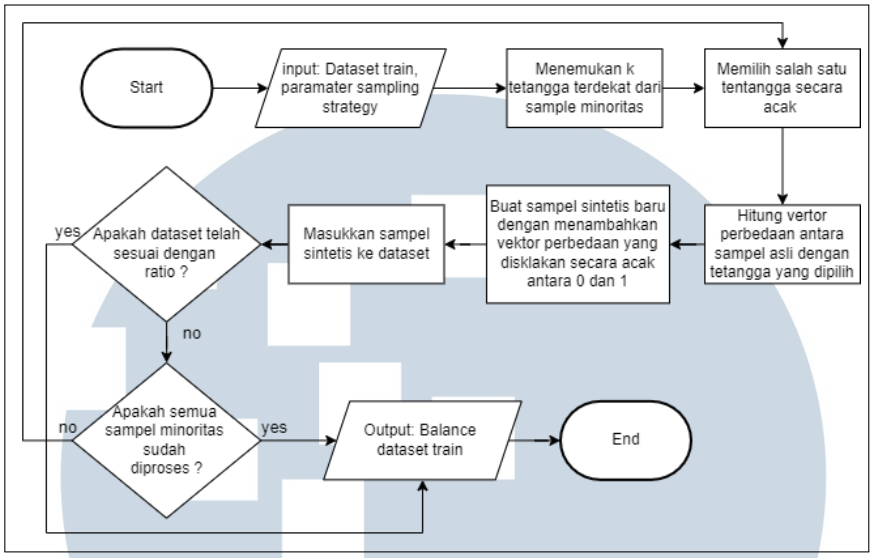
Pada gambar 3.7 merupakan *flowchart* yang menggambarkan langkah-langkah pembagian data. Dataset dibagi menjadi dua subset yang berbeda yaitu set pelatihan dan set pengujian, dengan alokasi proporsional masing-masing sebesar 80% dan 20%. Pembagian dataset ini sangat penting untuk mengevaluasi kinerja model dan memastikan bahwa hasilnya kuat dan signifikan secara statistik. Set pelatihan digunakan untuk melatih model dan set pengujian digunakan untuk mengevaluasi kinerja model [38]. Parameter random seed yang digunakan saat melakukan split data adalah 42.



Gambar 3.7. *Flowchart* Pembagian Data

3.3.3 SMOTE (*Synthetic Minority Oversampling Technique*)

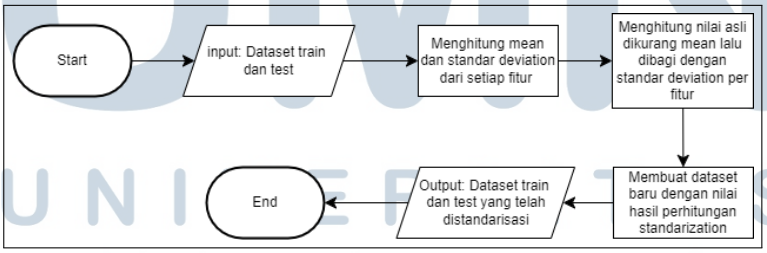
Pada gambar 3.8 merupakan *flowchart* yang menggambar cara kerja SMOTE. SMOTE (Synthetic Minority Oversampling Technique) adalah suatu teknik yang membuat kelas minoritas menjadi seimbang dengan kelas mayoritas. Teknik ini membuat data sintetis baru dari tetangga terdekat menggunakan *euclidean distance* [13].



Gambar 3.8. *Flowchart* SMOTE

3.3.4 Standarisasi Data

Pada gambar 3.9 merupakan *flowchart* yang menampilkan alur dalam melakukan standarisasi dataset. Standarisasi dataset adalah suatu teknik untuk membuat data - data individu tidak lebih kurang mirip menjadi data yang berdistribusi normal standar. Standarisasi yang diterapkan menggunakan teknik *Z-Score Standardization*. *Z-Score Standardization* merupakan perhitungan dari nilai asli dari fitur dalam dataset dikurangi dengan rata-rata dari sampel kemudian dibagi dengan standar deviasi dari sampel tersebut[39].

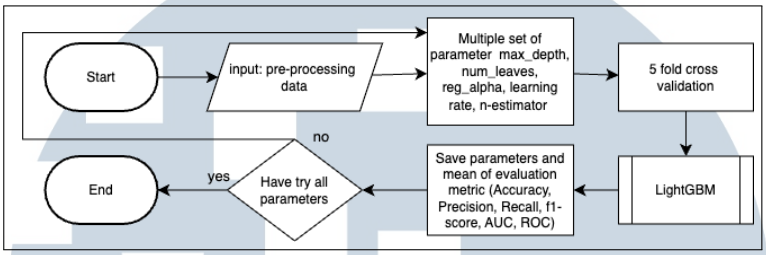


Gambar 3.9. *Flowchart* Standarisasi Data

3.4 Pembangunan Model

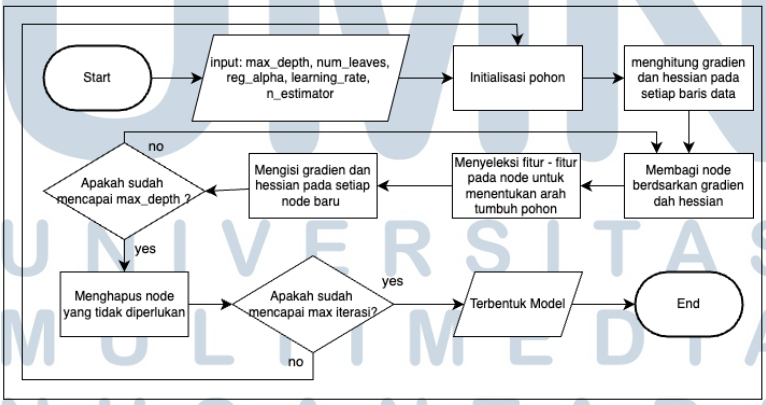
Pada gambar 3.10 merupakan *flowchart* untuk pembangunan model dari algoritma LightGBM. Algoritma *Light Gradient Boosting Machine (LightGBM)*

classifier yang diimplementasikan bertujuan untuk melakukan prediksi untuk mendeteksi penyakit jantung berdasarkan dataset *heart disease UCI Machine Learning*.



Gambar 3.10. *Flowchart* Pembangunan Model

Pada langkah pertama, perlu diberikan inisiasi beberapa *hyperparameter* seperti *max_depth, num_leaves, reg_alpha, learning_rate, dan n_estimator. Parameter max_depth* akan menentukan berapa kedalaman maksimal dalam sebuah pohon dan n_estimator untuk menentukan berapa banyak iterasi pembentukan pohon. Setelah inisiasi parameter maka akan melakukan *k-fold 10 cross validation*. *Cross validation* sangat diperlukan dalam proses training dengan dataset yang kecil [25]. Setelah itu, parameter tersebut dicoba satu per satu sehingga mendapatkan parameter terbaik yang disebut *hyperparameter tuning*. Metode tersebut dilakukan menggunakan *Grid Search*. Setelah mendapat model algoritma LightGBM dengan parameter terbaik maka dilakukan *predict* terhadap data testing untuk melihat performa model dalam prediksi untuk mendeteksi penyakit jantung.



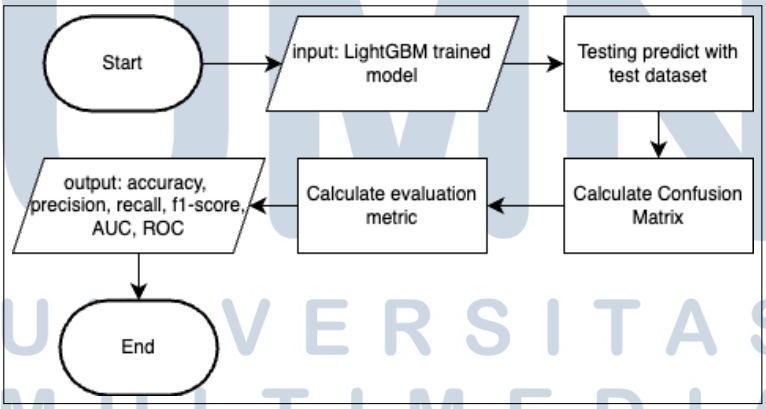
Gambar 3.11. *Flowchart* Algoritma LightGBM

Pada gambar 3.11 merupakan *flowchart* algoritma LightGBM. Langkah awal yang dilakukan pada algoritma LightGBM adalah melakukan inisialisasi

pohon. Setelah itu akan menghitung gradien dan *hessian* pada setiap baris data. Kemudian tahap selanjutnya dilakukan pembagian *node* berdasarkan hasil perhitungan tersebut. Menyeleksi fitur - fitur yang paling berpengaruh untuk menentukan arah tumbuh pohon tersebut. Dengan mengetahui arahnya maka dapat dibentuk *node* baru berdasarkan gradien dan *hessian*. *Node* tersebut akan terus tumbuh dalam pohon hingga mencapai *max_depth* yang ditentukan saat inisialisasi. Kemudian setelah itu dihapus *node* yang tidak diperlukan sehingga dapat menjadi acuan yang lebih efektif dalam melakukan pembentukan pohon baru. Iterasi pembentukan pohon baru sesuai dengan *hyperparameter* n_estimator yang telah diinisiasi di awal. Setelah iterasi tersebut selesai maka terbentuk model dari algoritma LightGBM.

3.5 Evaluasi Model

Pada gambar 3.12 merupakan *flowchart* evaluasi model. Flowchart tersebut berisi tahapan untuk menguji kinerja dari model yang telah kita training 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[34].



Gambar 3.12. *Flowchart* Evaluasi Model

3.6 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.