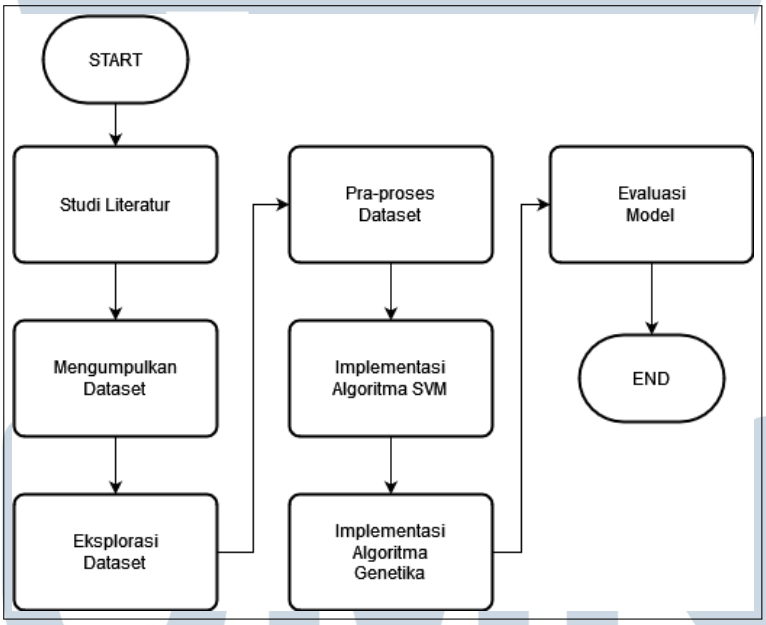
BAB 3
METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Gambar 3.1 menunjukkan tahapan-tahapan yang dilakukan dalam penelitian. Tahapan-tahapan tersebut adalah studi literatur mengenai topik yang bersangkutan dengan penelitian, mengumpulkan data yang akan digunakan, mengolah atau melakukan praproses serta eksplorasi data, mengimplementasi algoritma *Support Vector Machine*, mengimplementasi algoritma genetika dan yang terakhir adalah melakukan evaluasi tingkat akurasi model.



Gambar 3.1. *Flowchart* Metodologi Penelitian

3.1.1 Studi Literatur

Pada tahapan ini akan dilakukan studi literatur dengen mempelajari dan mencari informasi-informasi yang terkait dengan algoritma *Support Vector Machine*, algoritma genetika dari berbagai sumber seperti jurnal ilmiah, buku, dokumentasi dan lainnya. Tahapan ini berguna untuk membantu mengimplementasikan model pembelajaran mesin yang digunakan agar hasil dan tingkat akurasi yang maksimal dapat dicapai.

3.1.2 Pengumpulan Dataset

Dataset yang digunakan di dalam penelitian ini adalah dataset *Heart Disease* yang berasal dari situs *website UCI Machine Learning Repository*. Dataset merupakan hasil kompilasi data dari beberapa negara seperti Amerika Serikat, Switzerland, dan Hungary dan dibuat oleh Andras Janosi, M.D. dari *Hungarian Institute of Cardiology* di Budapest, William Steinbrunn, M.D. dari *University Hospital* di Zurich, Matthias Pfisterer, M.D. dari *University Hospital* di Basel dan Robert Detrano, M.D., Ph.D. dari *V.A. Medical Center, Long Beach and Cleveland Clinic Foundation* yang lalu didonasikan ke *University of California Irvine* untuk digunakan sebagai bahan penelitian di bidang pembelajaran mesin dan sudah dikutip serta digunakan di beragam penelitian [23].

Dataset yang digunakan memiliki 14 kolom *attribute* dan 242 jumlah data dengan 129 data yang positif memliki penyakit jantung dan 113 yang negatif memiliki penyakit jantung. Gambar 3.2 menunjukkan potongan dataset yang digunakan yang terdiri dari umur, jenis kelamin, *chest pain type*, *resting blood pressure*, *serum cholestoral in mg/dl*, *fasting blood sugar* $>$ *120 mg/dl*, *resting electrocardiographic results*, *maximum heart rate achieved*, *exercise induced angina*, *oldpeak = ST depression induced by exercise relative to rest*, *the slope of the peak exercise ST segment*, *number of major vessels (0-3) colored by flourosopy*, *thal*, dan *target* yang menunjukkan klasifikasi penyakit jantung dan menjadi kolom yang akan diprediksi dengan menggunakan model pembelajaran mesin yang dikembangkan.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	69	1	0	160	234	1	2	131	0	0.1	1	1	0	0
1	69	0	0	140	239	0	0	151	0	1.8	0	2	0	0
2	66	0	0	150	226	0	0	114	0	2.6	2	0	0	0
3	65	1	0	138	282	1	2	174	0	1.4	1	1	0	1
4	64	1	0	110	211	0	2	144	1	1.8	1	0	0	0

Gambar 3.2. Potongan Dataset *Heart Disease*

3.1.3 Eksplorasi Data

Pada tahapan ini dilakukan eksplorasi data untuk mendapatkan wawasan dan mengidentifikasi pola pada data dengan melakukan visualisasi dan menghitung beragam statistik. Eksplorasi data juga dapat mengungkap anomali atau *outlier* dalam data dan dapat membantu mengidentifikasi hubungan antara variabel sehingga meningkatkan pemahaman mengenai data yang akan diprediksi.

3.2 Spesifikasi Sistem

Untuk melakukan proses pelatihan model algoritma *Support Vector Machine*, spesifikasi perangkat yang digunakan adalah sebagai berikut:

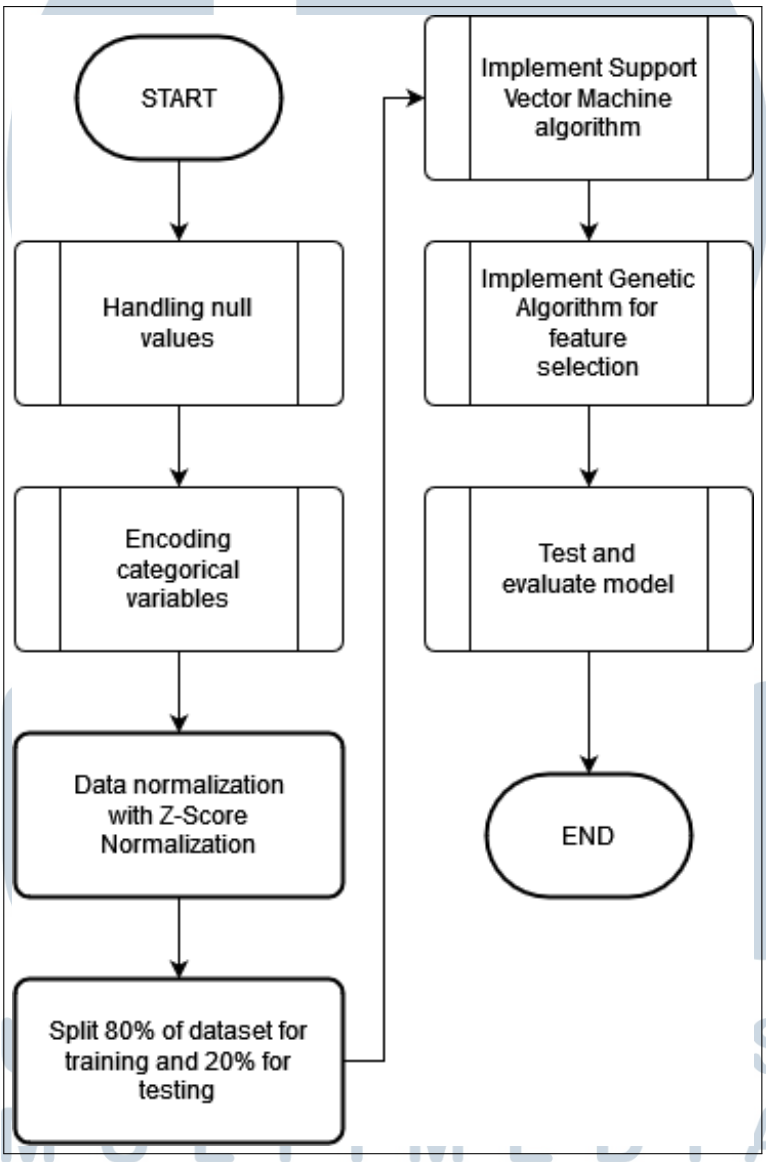
- CPU: AMD Ryzen 5 3600
- GPU: NVIDIA RTX 2070 Super
- RAM: 16GB DDR4

Adapun *tools* atau *software* yang digunakan dalam penelitian ini adalah:

- Google Collab
- Python (Scikit-learn, Matplotlib, Numpy, Pandas)

3.3 Flowchart Sistem

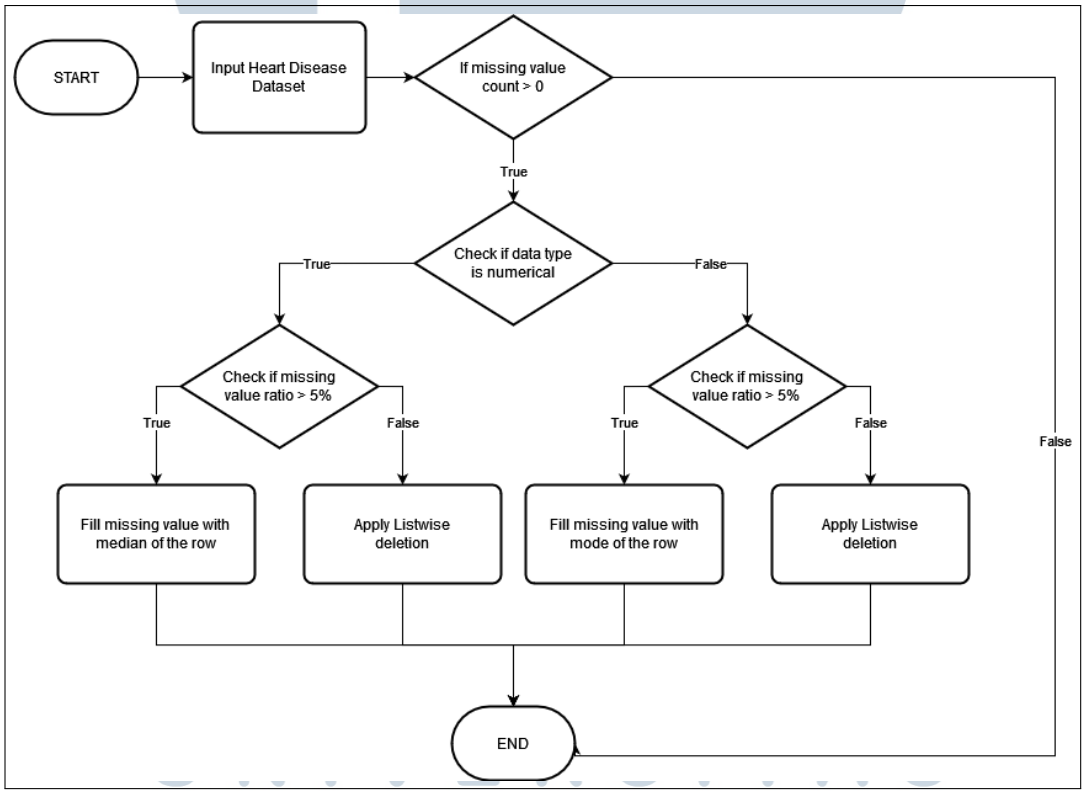
Gambar 3.3 menunjukkan keseluruhan tahapan yang digunakan untuk mengimplementasi dan mengembangkan model pembelajaran mesin *Support Vector Machine* dengan seleksi fitur algoritma genetika untuk mendeteksi penyakit jantung.



Gambar 3.3. *Flowchart* Sistem

3.3.1 Menangani Null Values

Nilai yang hilang atau *null values* umumnya terjadi dalam fase pengumpulan atau observasi data karena terdapat kesalahan ketika memasukkan data secara manual, kesalahan pengukuran data atau data tidak tersedia untuk pengamatan tertentu [36][37]. Metode yang digunakan untuk menangani nilai yang hilang adalah dengan menghapus data yang tidak lengkap. Metode tersebut dapat digunakan apabila jumlah data yang mengandung nilai yang hilang tidak melebihi 5%, apabila jumlah data sudah melebihi itu maka akan lebih baik apabila nilai yang hilang di imputasi atau diisi dengan nilai estimasi berdasarkan median data apabila tipe data adalah numerikal atau berdasarkan mode data apabila tipe data adalah kategorikal [36] seperti yang digambarkan pada Gambar 3.4 .

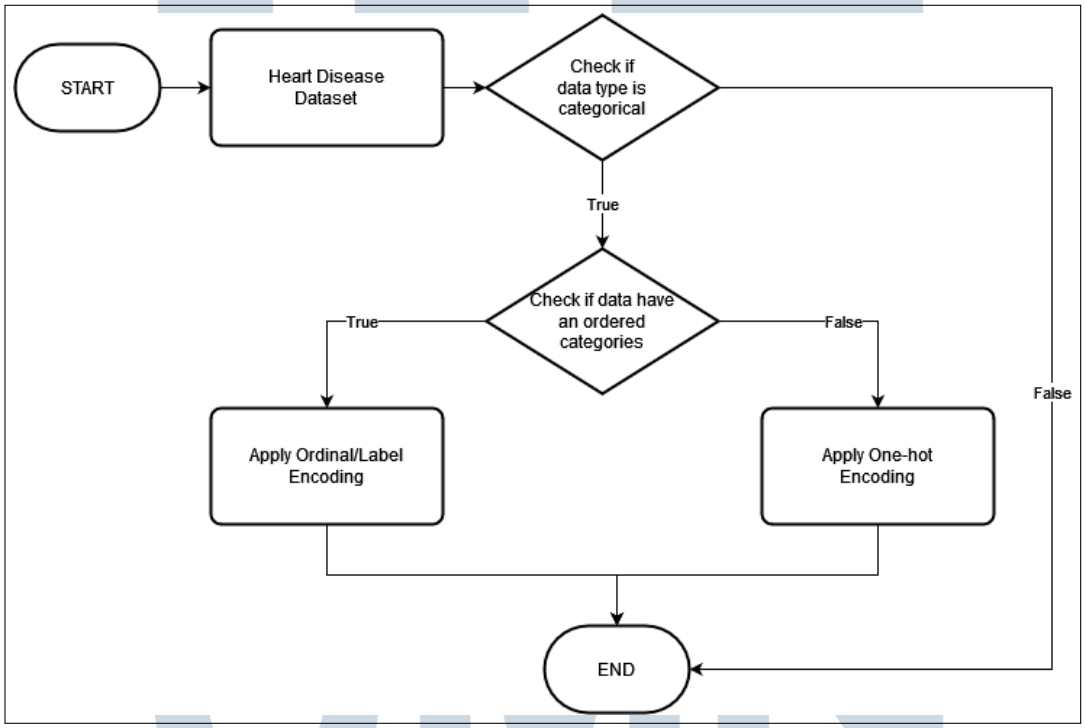


Gambar 3.4. *Flowchart Handling Null Values*

3.3.2 Encoding Data Kategorikal

Data kategorikal secara umum dapat dikategorikan menjadi data nominal atau data ordinal. Data nominal adalah data yang tidak memiliki nilai kuantitatif dan hanya untuk menunjukkan kategori atau klasifikasi seperti misalnya dalam kasus

dataset yang digunakan adalah jenis nyeri dada. Di sisi lain, data ordinal adalah data yang mengacu pada urutan dalam pengukuran dan menunjukkan peringkat atau arah [38]. Data-data tersebut perlu dirubah menjadi data numerik atau di *encode* karena model pembelajaran mesin tidak dapat menerima nilai kategorikal sebagai nilai input. Metode yang digunakan untuk mengubah nilai kategorikal menjadi numerik adalah dengan menggunakan teknik *Ordinal Encoding* atau *Label Encoding* apabila tipe data kategorikal adalah data ordinal dan menggunakan teknik *One-hot Encoding* apabila tipe data kategorikal adalah data nominal [39] seperti yang digambarkan pada Gambar 3.5.



Gambar 3.5. *Flowchart Encoding Categorical Variables*

3.3.3 Normalisasi Data

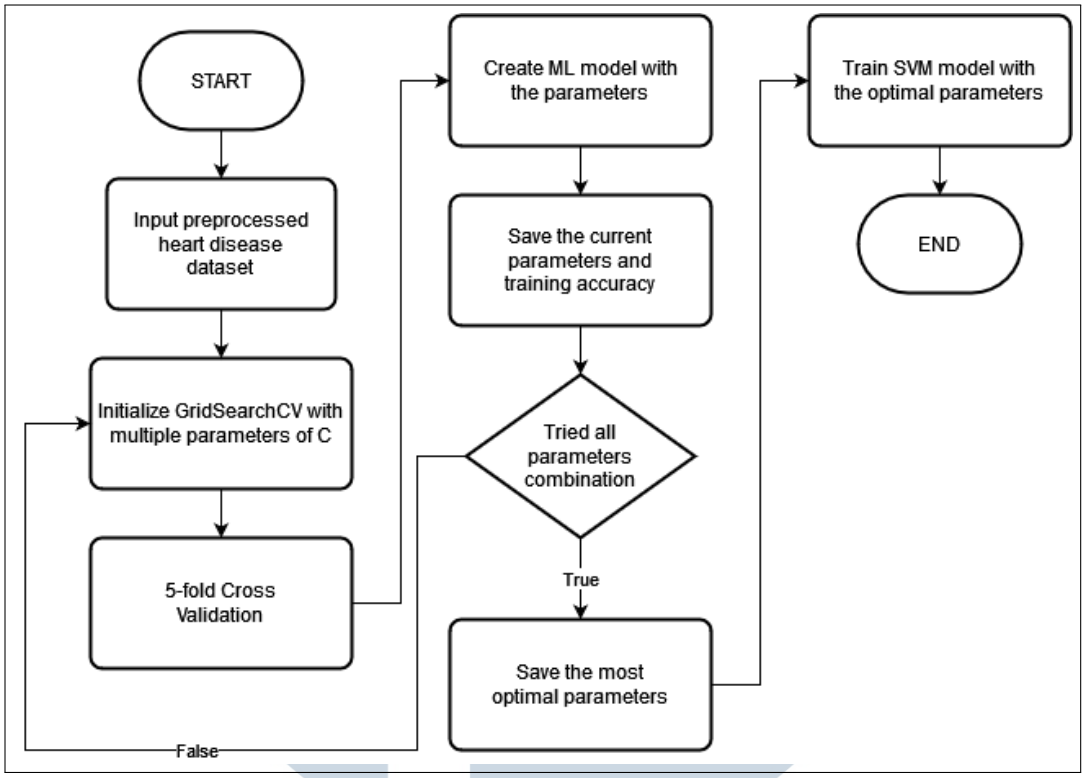
Normalisasi data adalah tahapan yang mentransformasi atau mengubah skala data sehingga setiap fitur memiliki kontribusi yang seragam dan nilai fitur numerik yang lebih besar tidak dapat mendominasi nilai fitur numerik yang lebih kecil [40]. Metode yang digunakan untuk normalisasi data adalah dengan teknik *Z-Score Normalization* yang menggunakan rata-rata dan standar deviasi untuk mengubah skala data sehingga fitur memiliki rata-rata nol dan standar deviasi sama dengan satu [41].

3.3.4 Pembagian Data

Sebelum membangun model pembelajaran mesin, dataset yang ada perlu dibagi terlebih dahulu menjadi data latih dan data uji. Data latih akan digunakan untuk melatih algoritma pembelajaran mesin yang digunakan sedangkan data uji digunakan untuk menguji dan mengevaluasi hasil tingkat akurasi dari model pembelajaran mesin dalam mendeteksi potensi penyakit jantung. Rasio pembagian data uji dan data latih adalah 80% dari keseluruhan data akan menjadi data latih sedangkan 20% akan menjadi data uji yang diambil dari prinsip Pareto yang menyatakan bahwa 80% dari output atau reaksi dihasilkan dari 20% aksi yang dilakukan [42] dan hal ini juga didukung oleh penelitian terdahulu yang menyatakan rasio pembagian 80/20 menghasilkan performa yang paling maksimal terutama untuk dataset yang besar dan untuk masalah klasifikasi. [43]

3.3.5 Implementasi Algoritma Support Vector Machine

Gambar 3.6 menunjukkan tahapan yang digunakan untuk menghasilkan model pembelajaran mesin dengan algoritma *Support Vector Machine*. Tahapan pertama adalah menginisialisasi fungsi GridSearchCV yang disediakan oleh *library* Scikit-learn dengan beberapa nilai C dan gamma yang akan digunakan. Parameter C digunakan untuk mengatur batas *hyperplane* atau batas keputusan. Semakin besar nilai C, nilai penalti untuk misklasifikasi juga akan besar sehingga batas keputusan dengan margin terkecil akan digunakan untuk mengurangi misklasifikasi. Begitu juga sebaliknya, semakin kecil nilai C, nilai penalti untuk misklasifikasi juga akan semakin kecil sehingga batasan keputusan dengan margin terbesar akan digunakan. Sedangkan gamma menentukan ketajaman kurva dari *hyperplane* atau menentukan seberapa jauh jangkauan pengaruh sebuah titik. Nilai gamma yang tinggi berarti hanya titik terdekat dengan batas keputusan yang akan memiliki bobot atau pengaruh. Di sisi lain, nilai gamma yang rendah berarti bahkan titik-titik yang jauh dari batas keputusan juga memiliki bobot atau pengaruh.



Gambar 3.6. *Flowchart Implement Support Vector Machine Algorithm*

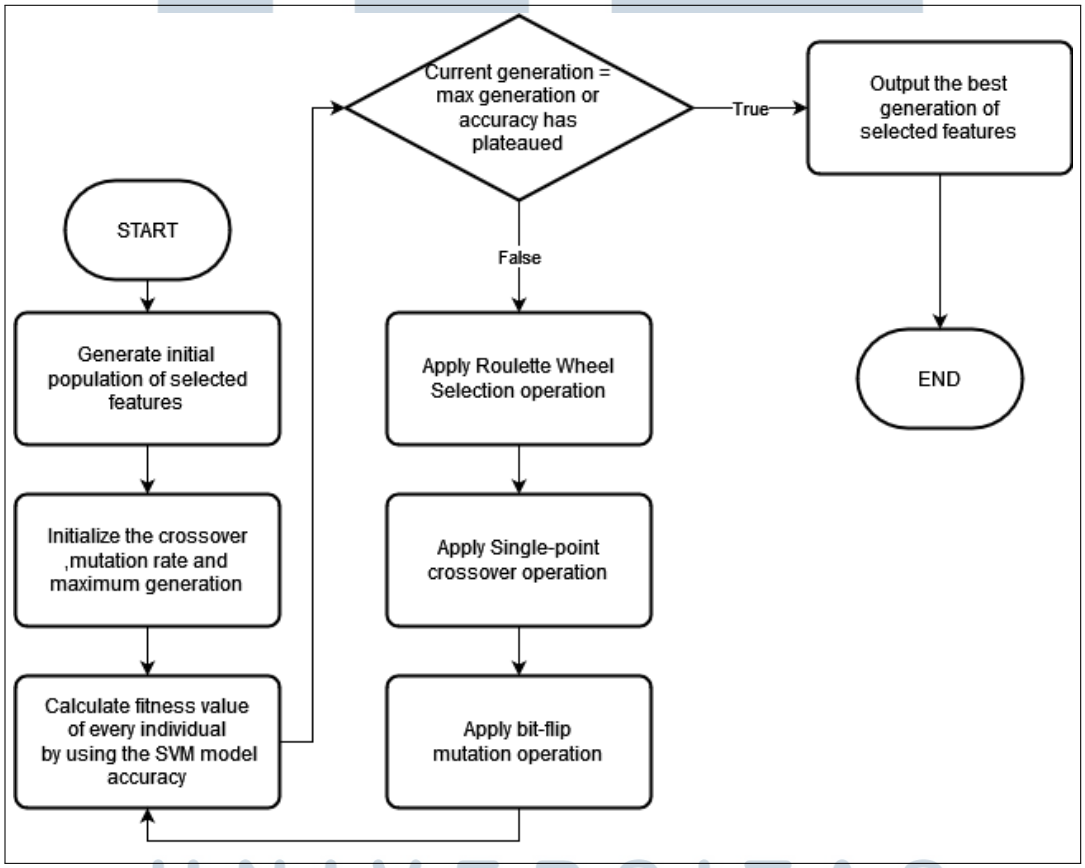
Fungsi GridSearchCV akan melakukan 5-*fold Cross Validation* untuk setiap nilai C yang diberikan sebagai parameter dan membuat model pembelajaran mesin dengan parameter C yang dipilih. Proses tersebut akan berulang sampai parameter terbaik didapatkan yang dapat menghasilkan tingkat akurasi yang terbaik.

3.3.6 Implementasi Algoritma Genetika

Gambar 3.7 menunjukkan tahapan yang digunakan untuk menseleksi fitur dengan menggunakan algoritma genetika. Tahapan awal adalah membuat populasi awal secara acak yang menentukan fitur apa saja yang digunakan dan menginisialisasi variabel yang menentukan seberapa sering operasi *crossover* dan operasi mutasi terjadi serta menginisialisasi jumlah generasi maksium yang akan digunakan untuk kriteria pemberhentian. Setelah itu setiap individu akan dikalkulasi nilai *fitness*nya dengan menggunakan *fitness function* yang menghitung tingkat akurasi model *Support Vector Machine* dengan seleksi fitur yang didefinisikan oleh kromosom yang sedang di evaluasi. Apabila tingkat akurasi sudah memenuhi kriteria pemberhentian yaitu ketika akurasi model tidak mengalami peningkatan yang stabil atau *plateaued* dan atau ketika algoritma

genetika sudah mencapai batasan generasi yang sudah diinisialisasi sebelumnya maka proses algoritma genetika akan terhentikan.

Apabila kriteria pemberhentian belum terpenuhi maka beberapa individu akan diseleksi dengan metode *Roulette Wheel* yang menseleksi setiap individu berdasarkan nilai probabilitas yang proposional dengan nilai *fitness*. Lalu operasi *Single-point crossover* dan operasi *Bit-flip mutation* akan dilakukan dengan probabilitas yang sudah diinisialisasikan sebelumnya. Proses tersebut akan berulang sampai kriteria pemberhentian terpenuhi dan generasi dengan akurasi terbaik dapat dihasilkan.

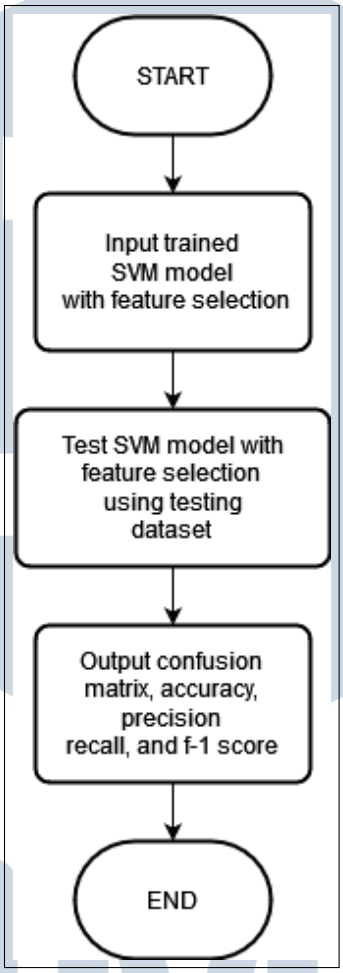


Gambar 3.7. *Flowchart Implement Genetic Algorithm for Feature Selection*

3.4 Evaluasi Model

Pada tahapan ini model pembelajaran mesin yang dihasilkan akan diuji dan dievaluasi dengan menggunakan dataset uji yang sebelumnya sudah dibagi di tahapan praproses data. Setelah itu, perhitungan seperti *precision*, *accuracy*, *f-1 score*, *weighted average*, *macro average* beserta dengan *confusion matrix* akan

digunakan untuk menentukan tingkat akurasi model dalam mendeteksi potensi penyakit jantung seperti yang digambarkan pada Gambar 3.8.



Gambar 3.8. *Flowchart Test and Evaluate Model*