

LAPORAN AKHIR

PREDIKSI RISIKO PENYAKIT JANTUNG MENGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE



Anggota Kelompok A:

Ayu Febriana Lingga	00000057105
Nayasha Clarisa Dwisutrisna	00000056883
Salwa Putri Riswana	00000057092
Wilcoustine Qhristmas Pniel Wijaya	00000056960

**KELAS IF540-E
SEMESTER GASAL 2023-2024**

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG**

2023

PREDIKSI RISIKO PENYAKIT JANTUNG MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE

ABSTRAK

Penyakit jantung adalah sebuah situasi dimana terdapat gangguan terhadap fungsi kerja jantung. Penyakit jantung sendiri memiliki banyak jenis misalnya kardi-vaskuler, jantung koroner dan serangan jantung secara tiba-tiba. Penyakit jantung menjadi salah satu penyakit yang kasusnya sering terjadi di khalayak umum tanpa memandang usia, jenis kelamin dan gaya hidup. Penyakit jantung sendiri dapat disebabkan oleh banyak hal, misalnya penyumbatan terhadap pembuluh darah, peradangan pada sistem, infeksi pada jantung, ataupun kelainan lainnya. Berdasarkan *WHO (World Health Organization)* yang merupakan sebuah organisasi kesehatan dunia menuliskan bahwa penyakit jantung adalah salah satu penyebab utama kematian di negara Inggris, Amerika Serikat, Kanada dan Australia. Hingga saat ini jumlah orang dewasa yang didiagnosis dengan penyakit jantung mencapai hingga 26,6 Juta Jiwa atau setara dengan 11,3% dari populasi orang dewasa. Dengan demikian, diagnosa dini terhadap penyakit jantung sangat penting untuk dilakukan. Namun, prediksi secara manual dapat dikatakan kurang efektif akibat kurangnya keahlian staff medis yang dapat menghasilkan prediksi yang salah. Seiring dengan berkembangnya waktu, teknologi yang semakin maju juga sangat membantu dalam bidang kesehatan terlebih lagi untuk menangani berbagai penyakit. Pada sistem prediksi resiko penyakit jantung yang disusun oleh penulis, akan digunakannya metode *SVM (Support Vector Machine)* yang menjadi bagian dari *machine learning*. Data yang digunakan oleh penulis merupakan dataset dengan judul "HEART DISEASE" yang memiliki 14 fitur dan mencakup 4 wilayah yaitu Cleveland, Hungaria, Switzerland, dan Long Beach V.

Kata kunci: *Heart Disease, Machine Learning, SVM (Support Vector Machine), Dataset, Prediction*

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

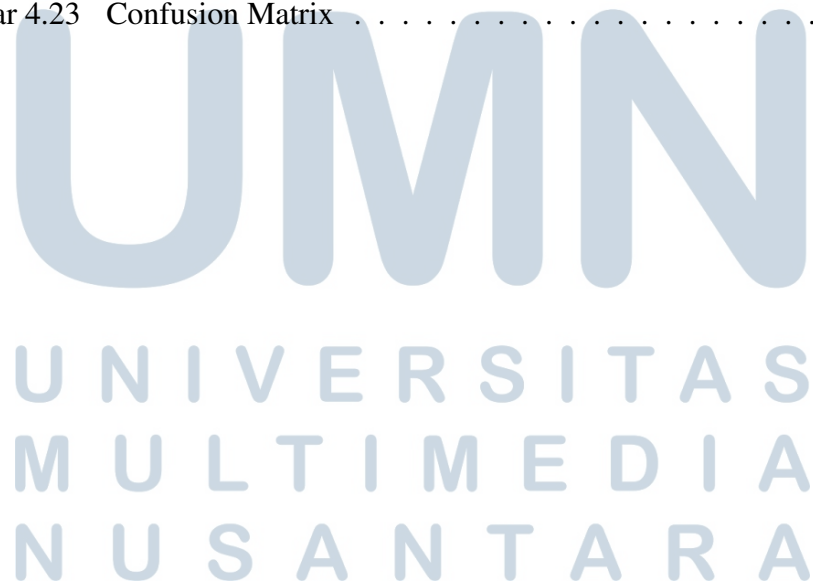
DAFTAR ISI

ABSTRAK	ii
DAFTAR ISI	iii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Permasalahan	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB 2 LANDASAN TEORI	5
2.1 Penyakit Jantung	5
2.2 <i>Machine Learning</i>	6
2.3 <i>Support Vector Machine</i>	7
2.3.1 <i>Kernel</i>	7
BAB 3 METODOLOGI PENELITIAN	9
3.1 Tahapan Penelitian	9
3.2 Dataset	10
3.3 <i>Preprocessing</i>	10
3.3.1 Visualisasi Data	11
3.3.2 <i>Data Scaling</i>	11
3.3.3 <i>Feature Selection</i>	11
3.3.4 <i>Outlier & Null Value Checking</i>	11
3.4 <i>Processing</i>	11
3.4.1 <i>Data Splitting</i>	12
3.4.2 Proses klasifikasi dengan metode SVM	12
3.4.3 Model SVM	12
3.4.4 <i>Underfit / Overfit Checking</i>	12
3.5 <i>Tuning Model</i>	13
3.6 Evaluasi	13
3.7 Dokumentasi Hasil	13
BAB 4 HASIL DAN DISKUSI	14
BAB 5 SIMPULAN DAN SARAN	25
5.1 Simpulan	25
5.2 Saran	25
DAFTAR PUSTAKA	26

UNIVERSITAS
MULTIMEDIA
NUSANTARA

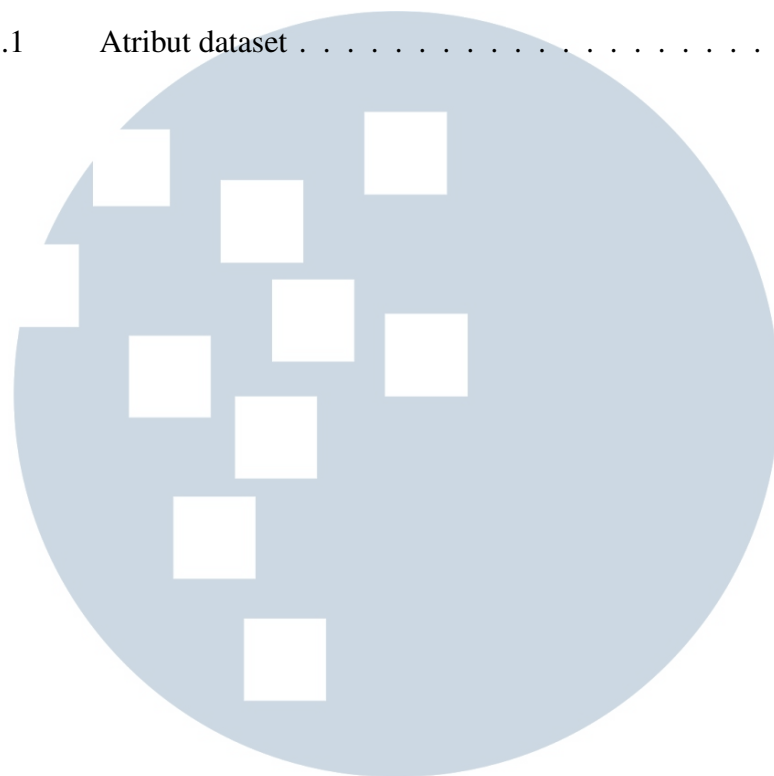
DAFTAR GAMBAR

Gambar 3.1	Flowchart <i>Data Preprocessing</i>	9
Gambar 3.2	Flowchart <i>Data Processing</i>	9
Gambar 3.3	<i>Bar Chart</i> kolom Target dalam dataset	10
Gambar 3.4	Contoh <i>output</i>	12
Gambar 4.1	Import Library	14
Gambar 4.2	Import Dataset	14
Gambar 4.3	Informasi Data	15
Gambar 4.4	Visualisasi Data 1	15
Gambar 4.5	Visualisasi Data 2	15
Gambar 4.6	Visualisasi Data 3	16
Gambar 4.7	Visualisasi Data 4	16
Gambar 4.8	Visualisasi Data 5	16
Gambar 4.9	Visualisasi Heatmap	17
Gambar 4.10	Pengecekan Outlier	18
Gambar 4.11	Cek nilai Null atau NaN	18
Gambar 4.12	Visualisasi Bar Chart Frekuensi Usia	19
Gambar 4.13	Split Data	19
Gambar 4.14	Pengecekan Overfit/Underfit Data	20
Gambar 4.15	Cek Overfit/Underfit 1	20
Gambar 4.16	Cek Overfit/Underfit 1	20
Gambar 4.17	Cek Overfit/Underfit 1	21
Gambar 4.18	Cek Overfit/Underfit 1	21
Gambar 4.19	Hasil Perhitungan dalam Tabel	22
Gambar 4.20	Hasil Perhitungan menggunakan Input User	22
Gambar 4.21	Akurasi, Precision, Recall, dan F1 Score Model SVM	23
Gambar 4.22	Tuning Data	23
Gambar 4.23	Confusion Matrix	23



DAFTAR TABEL

Tabel 3.1	Atribut dataset	10
-----------	---------------------------	----



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

BAB 1 PENDAHULUAN

1.1 Latar Belakang Masalah

Penyakit jantung adalah sebuah situasi di mana terdapat gangguan terhadap fungsi kerja jantung. Penyakit jantung sendiri memiliki banyak jenis misalnya kardiovaskuler, jantung koroner dan serangan jantung secara tiba-tiba. Penyakit jantung menjadi salah satu penyakit yang kasusnya sering terjadi di khalayak umum tanpa memandang usia, jenis kelamin dan gaya hidup [1]. Penyakit jantung sendiri dapat disebabkan oleh banyak hal, misalnya penyumbatan terhadap pembuluh darah, peradangan pada sistem, infeksi pada jantung, atau pun kelainan lainnya. Adapun penyebab seseorang dapat terkena penyakit jantung melibatkan banyak faktor, misalnya terlalu sering mengonsumsi rokok, riwayat keluarga atau keturunan, pola makan yang tidak teratur, tekanan darah yang cukup tinggi, kadar kolesterol yang tinggi, diabetes, tidak menjaga kebersihan tubuh, usia, dan juga jenis kelamin [2]. Melalui faktor di atas, terdapat beberapa hal yang biasanya dirasakan oleh pengidap penyakit jantung yaitu sesak napas, mual, keringat dingin, jantung berdebar, nyeri dada hingga hilangnya kesadaran atau pingsan [3].

Berdasarkan *WHO (World Health Organization)* yang merupakan sebuah organisasi kesehatan dunia menuliskan bahwa penyakit jantung adalah salah satu penyebab utama kematian di negara Inggris, Amerika Serikat, Kanada dan Australia. Hingga saat ini jumlah orang dewasa yang di diagnosis dengan penyakit jantung mencapai hingga 26,6 Juta Jiwa atau setara dengan 11,3% dari populasi orang dewasa [1]. Penyakit jantung merupakan salah satu penyumbang kematian terbanyak di dunia setiap tahunnya, angkanya mencapai sekitar 17 juta orang yang meninggal akibat penyakit jantung [4]. Dengan demikian, diagnosa dini terhadap penyakit jantung sangat penting untuk dilakukan. Namun, prediksi secara manual dapat dikatakan kurang efektif akibat kurangnya keahlian *staff* medis yang dapat menghasilkan prediksi yang salah [5].

Seiring dengan berkembangnya waktu, teknologi yang semakin maju juga sangat membantu dalam bidang kesehatan terlebih lagi untuk menangani berbagai macam penyakit. Salah satunya adalah sebuah teknologi yang telah berkembangnya disebut juga sebagai kecerdasan buatan. Teknologi ini merupakan sebuah pengembangan sistem atau mesin dalam melakukan hal-hal yang biasanya dilakukan oleh

manusia. Maka dari itu, melalui penelitian ini penulis akan memanfaatkan sebuah teknologi kecerdasan buatan yaitu *machine learning*.

Machine learning, merupakan sebuah pendekatan pengolahan data yang terus belajar dan berkembang melalui pengalaman dan data yang diterima. Pada sistem prediksi risiko penyakit jantung yang disusun oleh penulis, akan di gunakan-nya metode *SVM (Support Vector Machine)* yang menjadi bagian dari *machine learning*. *Support Vector Machine* merupakan salah satu bagian dari algoritma *machine learning* yang termasuk ke dalam kategori *supervised learning*. Metode ini biasanya digunakan untuk kasus klasifikasi dan regresi [6].

Penulis memutuskan untuk menggunakan metode *SVM* yang diduga dapat dengan baik memproses data yang kompleks dengan fitur yang banyak. Data yang digunakan oleh penulis merupakan dataset dengan judul "*HEART DISEASE*" yang memiliki 14 fitur dan mencakup 4 wilayah yaitu Cleveland, Hungaria, Switzerland, dan Long Beach V. Di mana keempat wilayah tersebut merupakan wilayah yang berada di Eropa dan Amerika Serikat.

Meneliti lebih lanjut, terdapat sebuah studi komprehensif di setiap kabupaten di Eropa dan Amerika Serikat. Melalui studi tersebut, diketahui bahwa terdapat 58% dari 1.391 daerah di Eropa memiliki tingkat kematian yang lebih banyak dibandingkan tingkat kelahiran pada dekade pertama abad ke-21 yaitu dalam rentang waktu 2000 hingga 2009. Hasil tersebut lebih banyak dibandingkan dengan hasil dari Amerika Serikat yang angka kematiannya hanya 28% dari 3.141 kabupaten [7].

Hal yang cukup mengejutkan mendengar banyaknya dan tingginya angka kematian dari negara Eropa dan juga Amerika Serikat, saat ternyata berdasarkan *Bloomberg Healthiest Country Index* edisi 2019, banyaknya wilayah bagian Eropa dan Amerika Serikat menjadi wilayah dengan tingkat kesehatan yang tinggi bahkan Swiss yang berada di wilayah Switzerland pun masuk ke dalam urutan kelima [8]. Dengan pola hidup yang ternyata produktif dan juga teratur, ternyata tidak menutup kemungkinan untuk sebuah negara memiliki angka kematian yang besar akibat penyakit jantung.

Berdasarkan penjabaran latar belakang di atas, penulis ingin membuat sebuah sistem prediksi risiko penyakit jantung untuk keempat wilayah di Eropa dan Amerika Serikat tersebut untuk membantu menurunkan angka kematian dengan cara pencegahan secara dini atas penyakit jantung.

1.2 Rumusan Masalah

Melalui latar belakang dan juga permasalahan yang sudah dijabarkan, terdapat rumusan masalah yang akan ditekuni oleh penulis, sebagai berikut:

1. Bagaimana implementasi sistem prediksi risiko penyakit jantung menggunakan *SVM* pada masyarakat Eropa dan Amerika Serikat terkhususnya wilayah Cleveland, Hungaria, Switzerland, dan Long Beach V?

1.3 Batasan Permasalahan

Adapun beberapa batasan yang diperlukan agar penelitian ini memiliki fokus yang sama dari awal hingga akhirnya, sehingga tidak terjadinya penyimpangan saat penelitian dilaksanakan.

1. Penelitian ini dilakukan menggunakan algoritma *SVM*
2. Penelitian ini mengambil *sample* dari dataset *HEART DISEASE* yang mencakup 14 fitur untuk diolah dalam model *SVM* yang penulis gunakan.
3. Batasan wilayah yang tercakup dalam dataset adalah Cleveland, Hungaria, Switzerland, dan Long Beach V.
4. Penelitian ini berfokus pada pembentukan sebuah sistem prediksi risiko penyakit jantung yang efektif dengan tingkat akurasi yang tinggi.

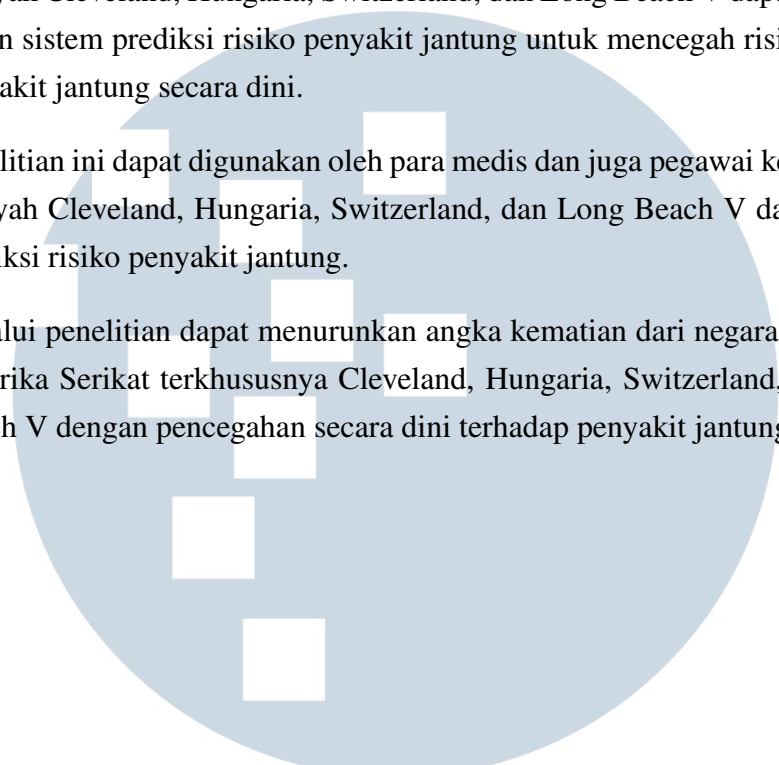
1.4 Tujuan Penelitian

Melalui penelitian, terdapat sebuah hal yang ingin dicapai oleh penulis, di mana hal tersebut ialah:

1. Mengetahui implementasi sistem prediksi risiko penyakit jantung menggunakan *SVM* pada masyarakat Eropa dan Amerika Serikat terkhususnya wilayah Cleveland, Hungaria, Switzerland, dan Long Beach V.

1.5 Manfaat Penelitian

Pastinya, melalui penelitian ini, terdapat beberapa harapan penulis agar penelitian ini dapat dipergunakan. Yang akan dijabarkan, seperti berikut:

- 
1. Melalui penelitian ini masyarakat Eropa dan Amerika Serikat terkhusus pada wilayah Cleveland, Hungaria, Switzerland, dan Long Beach V dapat menggunakan sistem prediksi risiko penyakit jantung untuk mencegah risiko terkena penyakit jantung secara dini.
 2. Penelitian ini dapat digunakan oleh para medis dan juga pegawai kesehatan di wilayah Cleveland, Hungaria, Switzerland, dan Long Beach V dalam memprediksi risiko penyakit jantung.
 3. Melalui penelitian dapat menurunkan angka kematian dari negara Eropa dan Amerika Serikat terkhususnya Cleveland, Hungaria, Switzerland, dan Long Beach V dengan pencegahan secara dini terhadap penyakit jantung.



BAB 2

LANDASAN TEORI

2.1 Penyakit Jantung

Jantung merupakan salah satu organ penting yang terdapat didalam tubuh dan memiliki tanggung jawab untuk memompa darah dan menyuplai oksigen ke seluruh tubuh [9]. Apabila kinerja jantung terganggu, jantung tidak dapat melakukan tugasnya dengan baik, sehingga pompa darah dan oksigen tidak optimal. Hal ini dapat terjadi karena sebagian otot jantung mati yang disebabkan oleh penyempitan arteri [10] atau terdapat penumpukan plak di arteri koroner [11]. Menurut World Health Organization atau WHO pada tahun 2021 [12], penyakit jantung terdapat beberapa jenis, diantaranya adalah sebagai berikut [13].

1. Penyakit jantung koroner merupakan salah satu penyakit jantung yang diakibatkan oleh menumpuknya plak pada pembuluh darah dan menyebabkan jantung tidak dapat memompa darah dengan baik.
2. Penyakit serebrovaskular merupakan salah satu penyakit jantung yang diakibatkan oleh adanya penyumbatan yang mengakibatkan pasokan ke otak menjadi terganggu.
3. Penyakit arteri perifer merupakan salah satu penyakit jantung yang diakibatkan oleh penyempitan pembuluh darah yang menyebabkan penyumbatan sehingga pasokan darah ke lengan dan kaki menjadi terganggu.
4. Penyakit jantung rematik merupakan salah satu penyakit jantung yang disebabkan oleh bakteri streptokokus sehingga menyebabkan terjadinya kerusakan pada otot dan katup jantung.
5. Penyakit jantung bawaan merupakan salah satu penyakit jantung yang disebabkan oleh kelainan struktur jantung sejak lahir.
6. Trombosis vena dalam dan emboli paru merupakan salah satu penyakit jantung yang disebabkan oleh gumpalan darah di pembuluh darah dan dapat berpindah ke jantung dan paru-paru.

Pada umumnya, penyakit jantung dapat disebabkan oleh dua faktor, yaitu faktor yang tidak dapat diubah dan faktor yang dapat diubah. Faktor yang tidak

dapat diubah adalah faktor yang sudah terjadi dan tidak dapat diubah dengan cara apapun. Faktor yang tidak dapat diubah meliputi umur, jenis kelamin, dan faktor genetik. Sedangkan faktor yang dapat diubah merupakan faktor yang sudah terjadi tetapi masih dapat diperbaiki. Faktor yang dapat diubah meliputi gaya hidup, berat badan, hingga kurangnya aktivitas fisik [10].

2.2 *Machine Learning*

Machine learning atau pembelajaran mesin merupakan jenis pengaplikasian dari kecerdasan buatan atau *artificial intelligence* yang memiliki kemampuan untuk belajar dari penginputan data [14]. Dalam hal ini, data historis dapat digunakan pada *machine learning* sebagai input untuk melakukan prediksi masa depan [15]. Pada umumnya, *machine learning* dapat digunakan untuk berbagai bidang dengan masalah yang berbeda-beda seperti prediksi risiko hingga pengenalan pola [16]. *Machine learning* memiliki empat kategori pembelajaran, diantaranya sebagai berikut [17].

1. *Supervised learning* atau pembelajaran terbimbing merupakan salah satu jenis pembelajaran yang terdapat pada *machine learning* dan digunakan untuk melakukan penyelesaian masalah klasifikasi serta regresi. Untuk menyelesaikan permasalahan klasifikasi dan regresi, *supervised learning* menggunakan data input yang memiliki label sehingga dapat membuat proses deteksi dan pengambilan keputusan menjadi lebih efektif dan cepat.
2. *Semi-supervised learning* atau pembelajaran semi terbimbing merupakan salah satu kategori pembelajaran yang terdapat pada *machine learning* dan sifatnya mirip dengan *supervised learning*, namun yang membedakan antara *supervised learning* dengan *semi-supervised learning* ada pada proses pelabelan data. Umumnya pada *semi-supervised learning* menggunakan data yang memiliki label lebih sedikit dibandingkan dengan data yang tidak memiliki label untuk melatih model.
3. *Unsupervised learning* atau pembelajaran tidak terbimbing merupakan salah satu jenis pembelajaran yang terdapat pada *machine learning* dan digunakan untuk melatih sistem. Untuk melatih sistem, *unsupervised learning* menggunakan data yang tidak memiliki label sehingga harus menemukan struktur yang tersembunyi dari data input yang tidak memiliki label dan melakukan pengelompokan berdasarkan pada kesamaan satu sama lain.

4. *Reinforcement learning* merupakan salah satu kategori pembelajaran yang terdapat pada *machine learning* yang digunakan untuk melatih agen untuk belajar melalui *trial and error* dengan menggunakan sistem *reward and punishment* [18].

2.3 Support Vector Machine

Algoritma *support vector machine* merupakan salah satu jenis algoritma dari *supervised learning* yang digunakan untuk melakukan perbandingan standar nilai diskrit pada parameter serta mengambil salah satu nilai pada parameter yang memiliki akurasi klasifikasi terbaik [19]. Persamaan untuk *decision function* dari algoritma *support vector machine* adalah sebagai berikut [20].

$$w \cdot x + b = 0 \quad (2.1)$$

Keterangan :

w = parameter *hyperplane* yang dicari

x = titik data *input support vector machine*

b = nilai bias

Pada umumnya, algoritma *support vector machine* digunakan untuk menyelesaikan masalah klasifikasi *linear*. Namun seiring berkembangnya waktu, algoritma *support vector machine* dapat melakukan klasifikasi *non-linear* [21]. Selain menyelesaikan masalah klasifikasi *linear* dan *non-linear*, algoritma *support vector machine* juga dapat digunakan untuk menyelesaikan masalah regresi.

2.3.1 Kernel

Dalam menyelesaikan masalah klasifikasi dan regresi, algoritma *support vector machine* dapat menggunakan *kernel* yang merupakan pemisah antara kelas satu dengan kelas yang lain. Pada umumnya, algoritma *support vector machine* memiliki tiga jenis *kernel*, yang diantaranya sebagai berikut [22].

1. *Linear* merupakan salah satu *kernel* yang dapat digunakan dalam algoritma *support vector machine* dengan cara menggunakan garis lurus sebagai *hyperplane* untuk antar kelas. Pada umumnya, *kernel linear* membutuhkan dua jenis variabel yaitu x_i dan x_j . Untuk melakukan perhitungan *support vector machine* dengan menggunakan *kernel linear*, nilai dari x_i akan dilakukan

transpose dan kemudian dikalikan dengan x_j .

$$k(x_i, x_j) = x_i^T x_j \quad (2.2)$$

Keterangan:

x_i = input vektor i

x_j = input vektor j

x_i^T = transpose dari vektor x_i

2. RBF atau *Radial Basis Function* merupakan salah satu *kernel* yang dapat digunakan dalam algoritma *support vector machine* dengan cara menggunakan parameter gamma dan C. Dalam hal ini, gamma memiliki fungsi untuk menjadi batas keputusan dan wilayah keputusan. Nilai gamma yang digunakan harus lebih dari angka 0 sehingga pada umumnya, nilai yang digunakan berkisar dari 0.0001 hingga 10. Sedangkan C memiliki fungsi untuk menjadi penalti terhadap kesalahan ketika melakukan klasifikasi.

$$\exp(-\gamma \|x_i - x\|^2), \gamma > 0 \quad (2.3)$$

Keterangan:

$-\gamma$ = konstanta yang mempengaruhi sensitivitas kernel

$\|x_i - x\|^2$ = jarak euclidean antara x_i dan x

3. *Polinomial* merupakan salah satu *kernel* yang dapat digunakan dalam algoritma *support vector machine* dengan cara menggunakan dua parameter yang berbeda dari *kernel linear* dan *kernel RBF*. *Kernel polinomial* menggunakan parameter r yang merupakan parameter bebas dan parameter d yang merupakan derajat atau kuadrat.

$$k(x_i, x) = (y \cdot x_i^T x + r)^d \quad (2.4)$$

Keterangan:

x_i dan x = input vektor

y = label kelas dari data input

r = parameter bebas

d = derajat polinomial

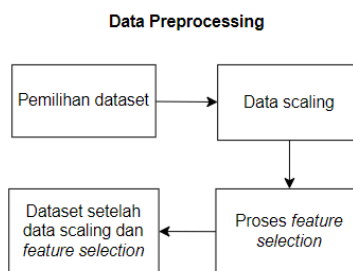
BAB 3

METODOLOGI PENELITIAN

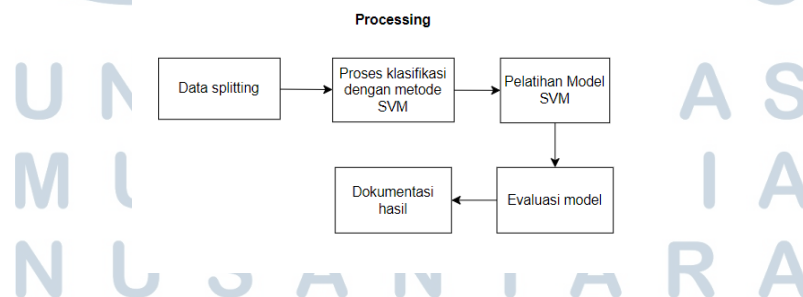
Pada bagian ini dijabarkan metodologi yang diterapkan dalam penelitian. Berikut adalah penjabaran alurnya.

3.1 Tahapan Penelitian

Pada sub-bab ini terdapat gambaran secara keseluruhan dari alur penelitian yang dilakukan dari awal hingga akhir. Tahapan penelitian yang dilakukan dalam penelitian dijabarkan dalam bentuk *flowchart*. Seluruh tahapan penelitian dilakukan menggunakan bahasa pemrograman python. Berikut adalah gambar dari *flowchart* tahapan penelitian. [23]



Gambar 3.1. Flowchart *Data Preprocessing*



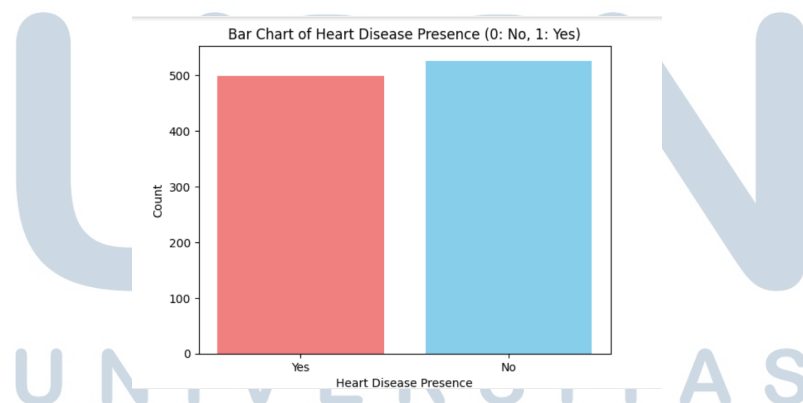
Gambar 3.2. Flowchart *Data Processing*

3.2 Dataset

Dataset yang digunakan bersumber dari Kaggle <https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset> yang di dalamnya terdapat 14 fitur yang mencakup usia, jenis kelamin, *chest pain* dan data pendukung prediksi lainnya.

Atribut	Keterangan
<i>Age</i>	Usia
<i>Sex</i>	Jenis Kelamin
<i>Chest Pain Type</i>	Nyeri dada
<i>Cholesterol</i>	Kadar Kolesterol
<i>FastingBS</i>	Kadar Gula Darah
<i>RestingECG</i>	Hasil <i>Electrocardiographic</i> Istirahat
<i>MaxHR</i>	Tingkat Detak Jantung Maksimum
<i>ExerciseAngina</i>	Induksi Angina
<i>Oldpeak</i>	Tingkat Depresi
<i>Slope</i>	kemiringan segmen ST
<i>ca (Vessels)</i>	Jumlah pembuluh darah utama yang terlihat dalam pemeriksaan flouroskopi
<i>Thal</i>	Hasil penggambaran talium dalam konteks medis
<i>Target</i>	Klasifikasi data

Tabel 3.1. Atribut dataset



Gambar 3.3. Bar Chart kolom Target dalam dataset

3.3 Preprocessing

Tahap awal dalam penelitian adalah *preprocessing*. Tahap ini dilakukan untuk memodifikasi data yang akan digunakan untuk pelatihan model *machine learning*.

3.3.1 Visualisasi Data

Visualisasi data adalah proses mewakili data secara grafis, baik dalam bentuk grafik, diagram, atau plot, dengan tujuan untuk membuat data lebih dapat dimengerti, mengidentifikasi pola, dan mengekspresikan informasi yang tersembunyi.

3.3.2 Data Scaling

Data scaling adalah proses mengubah rentang nilai (skala) dari suatu variabel dalam dataset sehingga variabel-variabel tersebut memiliki skala yang seragam. Tujuan utama dari *data scaling* adalah untuk memastikan bahwa variabel-variabel tersebut berada pada skala yang setara, sehingga perbandingan atau perhitungan jarak antara variabel-variabel tersebut dapat dilakukan dengan benar.

3.3.3 Feature Selection

Feature selection adalah proses pemilihan subset fitur dari suatu dataset yang paling relevan atau signifikan untuk digunakan dalam analisis atau model prediksi. Tujuan utama dari *feature selection* adalah untuk meningkatkan kinerja model dengan mengurangi dimensi dataset, sehingga hanya fitur-fitur yang paling informatif atau penting yang digunakan. Di dalam alur penelitian, penulis melakukan seleksi fitur dengan menggunakan *Heatmap Correlation*.

3.3.4 Outlier & Null Value Checking

Pengecekan *outlier* adalah langkah untuk mengidentifikasi nilai-nilai yang signifikan atau ekstrem dalam dataset. *Outlier* adalah data yang secara signifikan berbeda dari mayoritas data. Sedangkan, Pengecekan nilai *null* atau *missing values* adalah proses untuk mengidentifikasi apakah dataset mengandung *missing values* atau *null*.

3.4 Processing

Setelah dilakukan tahap data *preprocessing*, selanjutnya akan dimulai tahap *processing* dimana disini dilakukan beberapa hal inti dalam penelitian.

3.4.1 Data Splitting

Data splitting adalah proses membagi dataset menjadi beberapa subset yang berbeda untuk digunakan pada berbagai tahap dalam pembangunan dan evaluasi model *machine learning*. Pemisahan data ini penting untuk mengukur kinerja model secara objektif dan memastikan bahwa model dapat melakukan generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

3.4.2 Proses klasifikasi dengan metode SVM

Dalam proses ini, data yang sudah dilakukan *splitting* dan sudah melewati berbagai proses *preprocessing* digunakan untuk proses klasifikasi deteksi penyakit jantung menggunakan algoritma SVM. Klasifikasi ini bertujuan agar *output* yang dihasilkan oleh mesin nantinya akan sesuai dengan kriteria yang ingin ditampilkan. Pada penelitian ini, penulis mengklasifikasikan data menjadi 5 tipe *output*, yang terdiri dari risiko rendah, risiko sedang, risiko tinggi, risiko sangat tinggi dan risiko berat.

3.4.3 Model SVM

Model SVM yang penulis kembangkan memiliki fitur yang dapat memberi *output* sesuai input dari pengguna, contohnya jika pengguna ingin mengecek data index ke-10 dalam dataset, maka model SVM akan mencari dan mengklasifikasikan apakah pasien dengan index ke-10 tersebut memiliki risiko penyakit jantung di tingkat apa. Contohnya seperti yang ada di gambar berikut.

```
Input index ke (ketik 'selesai' untuk keluar): 10
Prediksi probabilitas untuk indeks 10:
Probabilitas kelas 0 (Tidak terkena penyakit jantung): 0.0207
Probabilitas kelas 1 (Terkena penyakit jantung): 0.9793
Resiko berat
```

Gambar 3.4. Contoh *output*

3.4.4 Underfit / Overfit Checking

1. **Underfitting** merupakan keadaan dimana model tidak memiliki kemampuan yang maksimal pada saat melakukan proses pada data pelatihan atau data

train. Jika ini terjadi, maka model akan menghasilkan kinerja buruk pada data *train*. [24]

2. **Overfitting** terjadi karena model memiliki kompleksitas yang lebih. Misalnya, model terlalu menghafal pola data sebelumnya dan karena itu, model tidak "mempelajari" pola baru. [24]

3.5 Tuning Model

Tuning model merupakan proses menyesuaikan parameter atau konfigurasi model untuk meningkatkan kinerja pada set data tertentu. Prosesnya melibatkan pemilihan parameter yang optimal, seperti tingkat pembelajaran, jumlah lapisan dan unit dalam jaringan saraf, atau parameter lainnya, untuk meningkatkan kemampuan model untuk mempelajari pola yang mewakili data dengan baik.

3.6 Evaluasi

Setelah melakukan *training* model SVM, penulis melakukan evaluasi model yang mencakup akurasi model, nilai *Precision*, nilai *Recall* dan *F1-Score*. Berikut adalah penjelasan lebih lanjut dalam proses evaluasi model yang sudah disebutkan.

1. **Akurasi Model SVM** berfungsi mengukur sejauh mana model dapat memprediksi dengan benar dari semua kelas.
2. ***Precision* SVM** berfungsi mengukur sejauh mana prediksi positif model benar, atau seberapa banyak dari yang diprediksi sebagai positif yang sebenarnya positif.
3. ***Recall*** berfungsi mengukur sejauh mana model dapat mendeteksi semua *instance* yang benar dari suatu kelas.
4. ***F1-Score*** merupakan nilai rata-rata harmonik antara presisi dan *recall*. Kedua metrik tersebut memberikan keseimbangan antara kedua metrik tersebut.

3.7 Dokumentasi Hasil

Setelah semua proses *machine learning* selesai, penulis melakukan dokumentasi hasil yang akan dijabarkan dalam bab selanjutnya.

BAB 4

HASIL DAN DISKUSI

Dalam melakukan pembangunan sebuah model, pada penelitian ini yaitu model *Support Vector Machine* dibutuhkan *import library* bagi penulis untuk melakukan prediksi risiko penyakit jantung menggunakan algoritma *Support Vector Machine*. Berikut penulis sertakan potongan kode dari *import library* yang diperlukan dalam penelitian.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from matplotlib import pyplot
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
from sklearn.decomposition import PCA as RandomizedPCA
from sklearn.pipeline import make_pipeline
from sklearn.metrics import confusion_matrix
```

Gambar 4.1. Import Library

Tahap selanjutnya setelah penulis melakukan *import library* adalah melakukan *import* dataset yang dibutuhkan untuk melakukan penelitian. Dataset yang diimport merupakan dataset sekunder yang berasal dari Kaggle. Dataset tersebut terdiri dari 14 fitur yang menjadi faktor pendukung dalam penelitian. Berikut penulis lampirkan potongan kode ketika mengimport dataset.

```
heart = pd.read_csv('heart.csv')
heart.head(20)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
5	58	0	0	100	248	0	0	122	0	1.0	1	0	2	1
6	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
7	55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
8	46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
9	54	1	0	122	286	0	0	116	1	3.2	1	2	2	0
10	71	0	0	112	149	0	1	125	0	1.6	1	0	2	1
11	43	0	0	132	341	1	0	136	1	3.0	1	0	3	0
12	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
13	51	1	0	140	298	0	1	122	1	4.2	1	3	3	0
14	52	1	0	128	204	1	1	156	1	1.0	1	0	0	0
15	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
16	51	0	2	140	308	0	0	142	0	1.5	2	1	2	1
17	54	1	0	124	266	0	0	109	1	2.2	1	1	3	0
18	50	0	1	120	244	0	1	162	0	1.1	2	0	2	1
19	58	1	2	140	211	1	0	165	0	0.0	2	0	2	1

Gambar 4.2. Import Dataset

Setelah melakukan *import* dataset, tahap berikutnya yaitu menampilkan informasi dari DataFrame 'heart' seperti jumlah baris, jumlah kolom, tipe data setiap kolom, Informasi DataFrame tersebut akan mempermudah penulis untuk

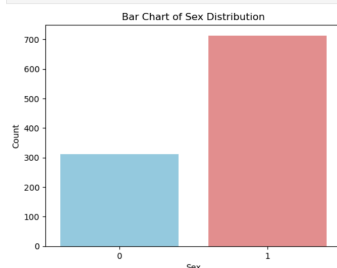
melakukan penelitian karena memberikan gambaran lebih untuk eksplorasi data serta pemahaman awal dari atribut yang ada.

```
heart.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   age         1025 non-null    int64
 1   sex         1025 non-null    int64
 2   cp          1025 non-null    int64
 3   trestbps    1025 non-null    int64
 4   chol        1025 non-null    int64
 5   fbs         1025 non-null    int64
 6   restecg     1025 non-null    int64
 7   thalach     1025 non-null    int64
 8   exang       1025 non-null    int64
 9   oldpeak     1025 non-null    float64
10   slope       1025 non-null    int64
11   ca          1025 non-null    int64
12   thal        1025 non-null    int64
13   target      1025 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

Gambar 4.3. Informasi Data

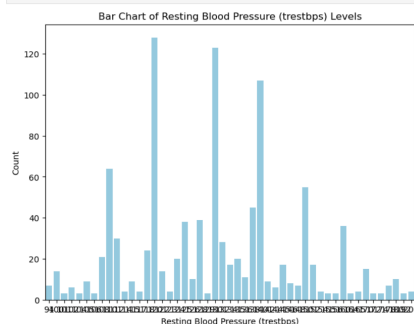
Setelah melakukan cetak informasi dari dataset dan mengetahui informasi dari setiap fitur yang ada dalam dataset. Penulis melakukan visualisasi data dari beberapa fitur yang cukup penting untuk memberikan persebaran gambaran mengenai fitur terkait. Berikut penulis sertakan code dan output dari visualisasi data beberapa fitur.

```
sex_counts = heart['sex'].value_counts()
sns.barplot(x=sex_counts.index, y=sex_counts.values, palette=['skyblue', 'lightcoral'])
plt.title('Bar Chart of Sex Distribution')
plt.xlabel('Sex')
plt.ylabel('Count')
plt.show()
```

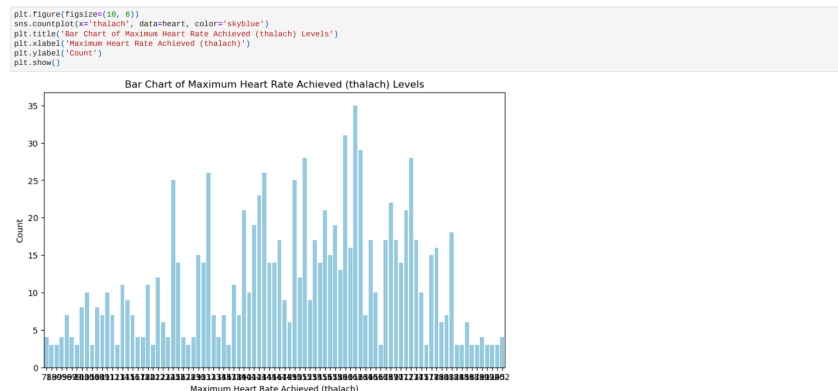


Gambar 4.4. Visualisasi Data 1

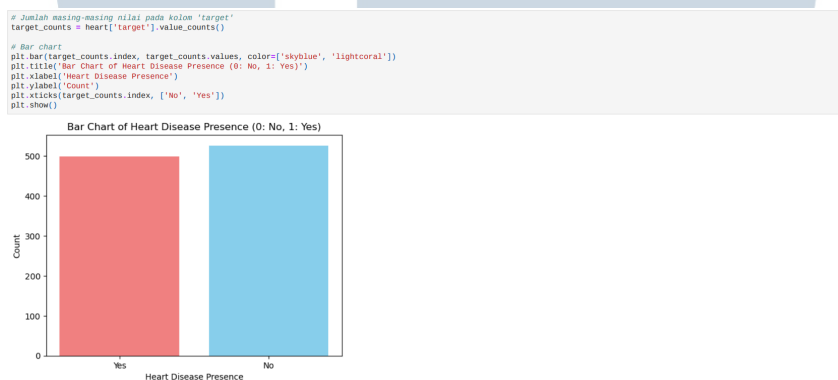
```
plt.figure(figsize=(8, 6))
sns.countplot(x='trestbps', data=heart, color='skyblue')
plt.title('Bar Chart of Resting Blood Pressure (trestbps) Levels')
plt.xlabel('Resting Blood Pressure (trestbps)')
plt.ylabel('Count')
plt.show()
```



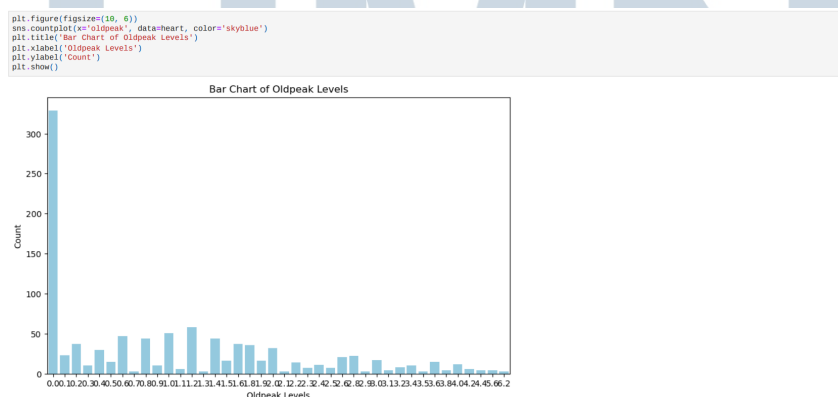
Gambar 4.5. Visualisasi Data 2



Gambar 4.6. Visualisasi Data 3



Gambar 4.7. Visualisasi Data 4



Gambar 4.8. Visualisasi Data 5

Setelah melakukan visualisasi data dari beberapa fitur dataset, penulis melakukan visualisasi *heatmap*. Visualisasi *heatmap* ini untuk menampilkan korelasi antar fitur dalam dataset 'heart'. Hal ini dapat mengukur sejauh mana dua

variabel berkaitan satu sama lain yang ditunjukkan dengan nilai -1, 1, dan 0. Visualisasi *heatmap* ini memberikan gambaran tentang korelasi fitur-fitur dalam dataset untuk membantu memberi informasi lebih lanjut tentang hubungan antar variabel dalam menganalisa data. Berikut penulis sertakan *code* dan hasil dari visualisasi *heatmap*.

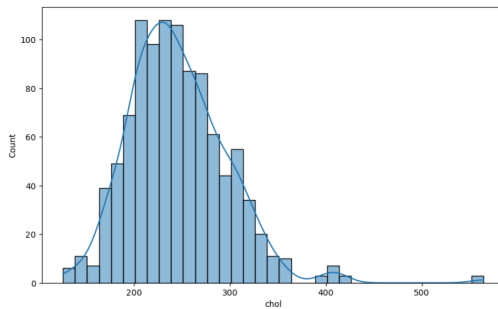


Gambar 4.9. Visualisasi Heatmap

Setelah melakukan visualisasi *heatmap*, penulis melakukan pengecekan untuk outlier terhadap fitur 'chol'. fitur 'chol' merupakan kadar serum kolestrol yang berada di dalam darah. Kolestrol ini merupakan lemak yang berada di dalam sel tubuh dan berbagai makanan yang dikonsumsi. Sehingga, kadar kolesterol menjadi faktor penting untuk menentukan risiko masalah kesehatan penyakit jantung. Berikut merupakan hasil cek outlier dari fitur 'chol'. Hasil outlier menunjukkan bahwa fitur chol datanya telah seimbang dilihat dari grafik gaussian yang sempurna.

UNIVERSITAS
MULTIMEDIA
NUSANTARA


```
plt.figure(figsize=(10, 6))
sns.histplot(heart['chol'], kde=True)
<AxesSubplot: xlabel='chol', ylabel='Count'>
```



Gambar 4.10. Pengecekan Outlier

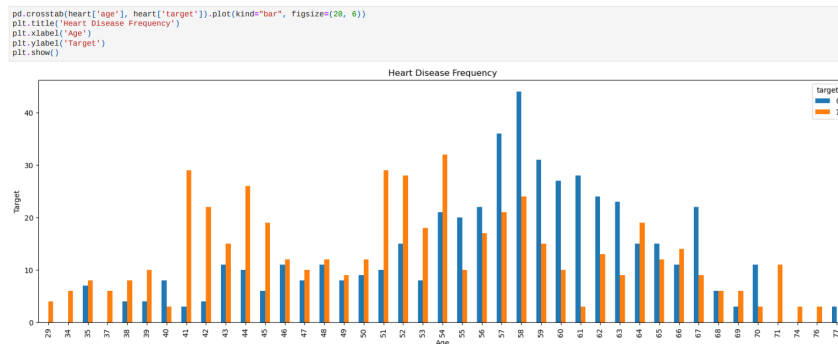
Setelah melakukan cek *outlier*, tahap selanjutnya yaitu menghitung jumlah nilai *null* atau *missing values* dalam setiap kolom DataFrame 'heart'. Dari kode dan output berikut menunjukkan bahwa DataFrame 'heart' tidak ada nilai *null* dalam setiap kolom karena ditampilkan bahwa jumlah nilai *null* untuk setiap kolom adalah 0.

```
heart.isnull().sum()
age      0
sex      0
cp       0
trestbps 0
chol     0
fbs      0
restecg  0
thalach  0
exang    0
oldpeak  0
slope    0
ca       0
thal     0
target   0
dtype: int64
```

Gambar 4.11. Cek nilai Null atau NaN

Setelah melakukan pengecekan *missing values*, penulis melakukan visualisasi berupa diagram batang dengan memberikan gambaran frekuensi penyakit jantung berdasarkan usia dalam dataset. Dilakukan pembuatan tabel silang terlebih dahulu antara dua tabel untuk menunjukkan distribusi frekuensi dari kedua variabel. Ditampilkan dalam bentuk diagram batang dengan sumbu x sebagai nilai usia dan sumbu y sebagai perwakilan frekuensi masing masing nilai usia untuk tiap nilai target.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 4.12. Visualisasi Bar Chart Frekuensi Usia

Setelah dilakukan visualisasi data mengenai distribusi frekuensi penyakit jantung, dilakukan pemisahan dataset menjadi data *train* dan *test*. Pada pemisahan dataset menjadi set *train* dan *test*, setelah itu dilakukan normalisasi dan dilakukan pelatihan model dari SVM. Pelatihan ini dilanjutkan dengan prediksi probabilitas pada dataset *test*. Setelah itu, disajikan hasil berdasarkan kategori yang telah dibuat pada kode tersebut.

```
# Memisahkan fitur (X) dan label (y)
X = heart.drop('target', axis=1)
y = heart['target']

# Memisahkan dataset menjadi set pelatihan dan set pengujian
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.098, random_state=42)

# Normalisasi fitur menggunakan StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Membuat dan melatih model SVM
svm_model = SVC(probability=True)
svm_model.fit(X_train_scaled, y_train)

# Melakukan prediksi probabilitas pada set pengujian
probabilities_all_svm = svm_model.predict_proba(X_test_scaled)

# Menampilkan seluruh data dalam bentuk tabel tanpa elipsis
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)

result_df_svm = pd.DataFrame({
    'Probabilitas Kelas 0': probabilities_all_svm[:, 0],
    'Probabilitas Kelas 1': probabilities_all_svm[:, 1],
    'Probabilitas Kelas 1': probabilities_all_svm[:, 1],
    'Resiko': pd.cut(probabilities_all_svm[:, 1], bins=[float('inf'), 0.1, 0.2, 0.3, 0.4, float('inf')],
        labels=['Resiko Rendah', 'Resiko Sedang', 'Resiko Tinggi', 'Resiko Sangat Tinggi', 'Resiko Berat'])
})
```

Gambar 4.13. Split Data

Setelah dilakukan pemisahan dataset menjadi data *train* dan *test*, penulis melakukan pengecekan *Overfit* ataupun *Underfit*. Pertama dilakukan penghapusan kolom target dari dataset dan menunjukkan bahwa dataset terdiri dari 1025 sampel dan tiap sampel memiliki 13 fitur, serta vektor target y memiliki 1025 elemen (sesuai dengan jumlah sampel). Kemudian, dilakukan split data menggunakan *library scikit-learn* pada dataset menjadi data *train* dan data *test*. Dari pemisahan data, digunakan sebesar 9.8 persen sebagai data untuk data uji dan sisanya sebagai data latih. Hasil menunjukkan bahwa data latih memiliki 924 sampel dengan 13 fitur dan data uji memiliki 101 sampel dengan 13 fitur. Target dari vektor tersebut untuk data train memiliki 924 elemen dan vektor target untuk data test memiliki 101 elemen. Berikut penulis sertakan potongan kode beserta outputnya.

```
# Menentukan dataset
X = heart.drop('target', axis=1)
y = heart['target']
print(X.shape, y.shape)

(1025, 13) (1025,)

# Melakukan split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.098, random_state=42)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

(924, 13) (101, 13) (924,) (101,)
```

Gambar 4.14. Pengecekan Overfit/Underfit Data

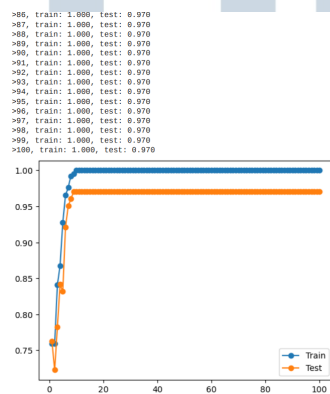
Kemudian, penulis melakukan pengujian data terjadi *overfit* atau *underfit*. Output dan grafik menunjukkan bahwa dataset yang digunakan mengalami sedikit *overfitting*. Hal ini ditunjukkan dari grafik yang menggambarkan bahwa akurasi data *train* lebih tinggi sedikit dibandingkan dengan grafik akurasi data *test*. Berikut penulis sertakan potongan kode dan hasil dari grafik pengecekan *underfit/overfit* data.

```
# Menentukan list untuk mengumpulkan skor
train_scores, test_scores = list(), list()
# Menentukan kedalaman tree yang akan dievaluasi
values = [i for i in range(1, 101)]
# Mengevaluasi decision tree dari setiap kedalaman
for i in values:
    model = DecisionTreeClassifier(max_depth=i)
    model.fit(X_train, y_train)
    train_yhat = model.predict(X_train)
    train_acc = accuracy_score(y_train, train_yhat)
    train_scores.append(train_acc)
    test_yhat = model.predict(X_test)
    test_acc = accuracy_score(y_test, test_yhat)
    test_scores.append(test_acc)
    print(">{0}, train: {1:.3f}, test: {2:.3f} % (i, train_acc, test_acc)")

# Plotting skor train dan test dengan kedalaman tree
pyplot.plot(values, train_scores, "-o", label="Train")
pyplot.plot(values, test_scores, "-o", label="Test")
pyplot.legend()
pyplot.show()

>1, train: 0.769, test: 0.762
>2, train: 0.769, test: 0.723
>3, train: 0.841, test: 0.782
>4, train: 0.868, test: 0.842
>5, train: 0.927, test: 0.832
>6, train: 0.965, test: 0.921
>7, train: 0.976, test: 0.958
>8, train: 0.992, test: 0.968
>9, train: 0.996, test: 0.979
>10, train: 1.000, test: 0.970
>11, train: 1.000, test: 0.970
>12, train: 1.000, test: 0.970
>13, train: 1.000, test: 0.970
>14, train: 1.000, test: 0.970
>15, train: 1.000, test: 0.970
>16, train: 1.000, test: 0.970
>17, train: 1.000, test: 0.970
>18, train: 1.000, test: 0.970
>19, train: 1.000, test: 0.970
```

Gambar 4.15. Cek Overfit/Underfit 1



Gambar 4.16. Cek Overfit/Underfit 1

```

# Menentukan kedalaman pohon yang akan dievaluasi
values = [1 for i in range(1, 101)]

# Menginisialisasi list untuk simpan skor
train_scores = []
test_scores = []

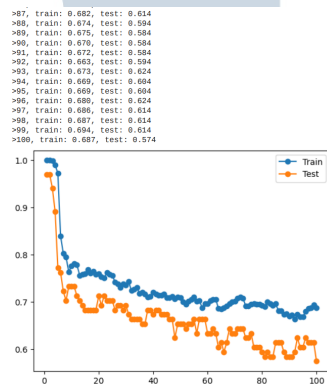
# Mengevaluasi decision tree dari setiap kedalaman
for i in values:
    model = KNeighborsClassifier(n_neighbors=i)
    model.fit(X_train, y_train)
    train_yhat = model.predict(X_train)
    train_acc = accuracy_score(y_train, train_yhat)
    train_scores.append(train_acc)
    test_yhat = model.predict(X_test)
    test_acc = accuracy_score(y_test, test_yhat)
    test_scores.append(test_acc)
    print('>%d, train: %.3f, test: %.3f' % (i, train_acc, test_acc))

# Plotting skor train dan test dengan kedalaman tree
plt.plot(values, train_scores, '-o', label='train')
plt.plot(values, test_scores, '-o', label='test')
plt.legend()
plt.show()

```

>1, train: 1.000, test: 0.979
>2, train: 1.000, test: 0.979
>3, train: 0.999, test: 0.941
>4, train: 0.999, test: 0.891
>5, train: 0.973, test: 0.772
>6, train: 0.849, test: 0.762
>7, train: 0.883, test: 0.723
>8, train: 0.795, test: 0.703
>9, train: 0.754, test: 0.733
>10, train: 0.776, test: 0.733
>11, train: 0.781, test: 0.733
>12, train: 0.779, test: 0.713
>13, train: 0.756, test: 0.703
>14, train: 0.759, test: 0.693
>15, train: 0.769, test: 0.683
>16, train: 0.768, test: 0.683

Gambar 4.17. Cek Overfit/Underfit 1



Gambar 4.18. Cek Overfit/Underfit 1

Setelah melakukan setiap proses dalam penelitian, penulis menampilkan hasil dari perhitungan yang telah dilakukan. Hasil yang ditampilkan merupakan persentase risiko orang terkait mengenai terkena penyakit jantung dan tidak terkena serta klasifikasi dari risiko tersebut. Probabilitas 0 adalah probabilitas seseorang tidak terkena penyakit jantung dan probabilitas 1 adalah probabilitas seseorang terkena penyakit jantung. Berikut penulis sertakan beberapa hasil output dari setiap index serta besar persentasenya dan klasifikasinya.

```
print(result_df_svm)
```

	Probabilitas Kelas 0	Probabilitas Kelas 1	Risiko
0	0.886472	0.993528	Risiko Berat
1	0.993501	0.996499	Risiko Berat
2	0.999644	0.880356	Risiko Rendah
3	0.992306	0.997694	Risiko Berat
4	0.946792	0.953208	Risiko Rendah
5	0.963938	0.936062	Risiko Berat
6	0.942914	0.957086	Risiko Rendah
7	0.996206	0.883794	Risiko Rendah
8	0.827297	0.972703	Risiko Berat
9	0.986255	0.913745	Risiko Rendah
10	0.823467	0.876533	Risiko Berat
11	0.976625	0.923375	Risiko Rendah
12	0.946678	0.953322	Risiko Berat
13	0.948022	0.951978	Risiko Berat
14	0.948828	0.859172	Risiko Rendah
15	0.148987	0.859013	Risiko Berat
16	0.984519	0.915481	Risiko Rendah
17	0.618962	0.981038	Risiko Berat
18	0.983965	0.916035	Risiko Berat
19	0.989618	0.810382	Risiko Rendah
20	0.227384	0.772616	Risiko Berat
21	0.969632	0.930368	Risiko Rendah
22	0.264294	0.735706	Risiko Berat
23	0.996862	0.903138	Risiko Rendah
24	0.227384	0.772616	Risiko Berat
25	0.864848	0.935152	Risiko Berat
26	0.614189	0.985811	Risiko Berat
27	0.948799	0.959201	Risiko Rendah
28	0.948799	0.959201	Risiko Rendah
29	0.833803	0.966197	Risiko Berat
30	0.948799	0.959201	Risiko Rendah
31	0.864848	0.935152	Risiko Berat
32	0.874133	0.125867	Risiko Sedang
33	0.983965	0.916035	Risiko Berat
34	0.836895	0.963105	Risiko Berat

Gambar 4.19. Hasil Perhitungan dalam Tabel

Setelah dilakukan pencetakan hasil output untuk memprediksi dari setiap index dalam bentuk tabel, pada tahap selanjutnya yaitu hanya berubah format penampilan dari cara menampilkan risiko. Pada tahap ini, user dapat melakukan input sesuai yang ingin dicari dengan melakukan input nomor index dan program akan mengkalkulasi dan menentukan untuk persentase serta klasifikasi dari risiko index yang diinput oleh user. Berikut penulis sertakan potongan kode dan hasil keluarannya.

```
# Melakukan prediksi probabilitas pada set pengujian
probs_all_svm = svm_model.predict_proba(X_test_scaled)

while True:
    input_index = input("Input index ke (ketik 'selesai' untuk keluar): ")
    if input_index.lower() == 'selesai':
        break
    try:
        index = int(input_index)
        if 0 <= index < len(probs_all_svm):
            prob_class_1 = probs_all_svm[index, 1]

            # Menampilkan hasil prediksi probabilitas untuk indeks yang diinput
            print(f"Prediksi probabilitas untuk indeks (index):")
            print(f"Probabilitas kelas 0 (Tidak terkena penyakit jantung): {1 - prob_class_1:.4f}")
            print(f"Probabilitas kelas 1 (Terkena penyakit jantung): {prob_class_1:.4f}")

            # Menampilkan berdasarkan kategori risiko
            if prob_class_1 < 0.10:
                print("Risiko rendah")
            elif prob_class_1 < 0.20:
                print("Risiko sedang")
            elif prob_class_1 < 0.30:
                print("Risiko tinggi")
            elif prob_class_1 < 0.40:
                print("Risiko sangat tinggi")
            else:
                print("Risiko berat")
            except ValueError:
                print("Indeks tidak valid. Silakan masukkan indeks yang sesuai.")
            except ValueError:
                print("Input harus berupa bilangan bulat.")

print("Terima kasih! semoga sehat selalu!")

Prediksi probabilitas untuk indeks 1:
Probabilitas kelas 0 (Tidak terkena penyakit jantung): 0.8649
Probabilitas kelas 1 (Terkena penyakit jantung): 0.9360
Risiko berat
Terima kasih! semoga sehat selalu!
```

Gambar 4.20. Hasil Perhitungan menggunakan Input User

Setelah mencetak hasil dari input user, selanjutnya yaitu menghitung nilai akurasi, presisi, *f1 score*, dan *recall* dari model SVM yang digunakan. Didapatkan hasil akurasi dari model SVM pada prediksi penyakit jantung adalah sebesar 91 persen. Berikut penulis sertakan potongan kode dan hasil dari akurasi, presisi, *f1 score*, dan *recall* SVM pada prediksi penyakit jantung.

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.098, random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

svm_model = SVC(probability=True)
svm_model.fit(X_train_scaled, y_train)
predict_svm = svm_model.predict(X_test_scaled)

accuracy_svm = accuracy_score(y_test, predict_svm)
precision_svm = precision_score(y_test, predict_svm)
recall_svm = recall_score(y_test, predict_svm)
f1_svm = f1_score(y_test, predict_svm)

print("Akurasi model SVM: (accuracy_svm:.4f)")
print("Precision SVM: (precision_svm:.4f)")
print("Recall SVM: (recall_svm:.4f)")
print("F1 Score SVM: (f1_svm:.4f)")

Akurasi model SVM: 0.9189
Precision SVM: 0.8669
Recall SVM: 0.9558
F1 Score SVM: 0.8953

```

Gambar 4.21. Akurasi, Precision, Recall, dan F1 Score Model SVM

Tahap ini dilakukan *tuning* data. Kemudian dilakukan PCA untuk mereduksi dimensi dan SVM dalam klasifikasi menggunakan *library scikit-learn*. Selanjutnya menggunakan GridSearchCV untuk melakukan pencarian grid pada parameter *hyperparameter* model.

```

pca = RandomizedPCA(n_components=10, whiten=True, random_state=42)
svc = SVC(class_weight='balanced')
model = make_pipeline(pca, svc)

from sklearn.model_selection import GridSearchCV
param_grid = {'svc_C': [1, 5, 10, 50],
              'svc_gamma': [0.0001, 0.0005, 0.001, 0.005]}

grid = GridSearchCV(model, param_grid, scoring='accuracy')
grid.fit(X_train, y_train)
print(grid.best_params_)

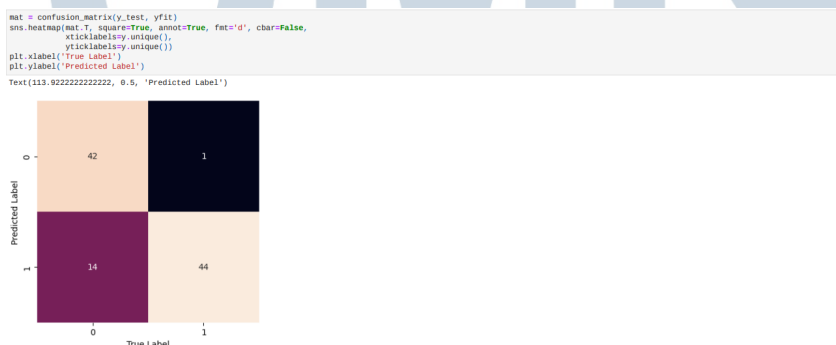
('svc_C': 50, 'svc_gamma': 0.005)

model = grid.best_estimator_
model.fit(X_train, y_train)
yfit = model.predict(X_test)

```

Gambar 4.22. Tuning Data

Selanjutnya yaitu melakukan pengukuran kinerja dalam model SVM yang digunakan dalam memprediksi jantung menggunakan *confusion matrix*. Hasil menunjukkan bahwa pada dataset yang bernilai *false* dan diprediksi *false* sebanyak 42. yang bernilai *false* namun diprediksi *true* sebesar 14. Kemudian, yang bernilai *true* dan diprediksi *true* sebesar 44 dan yang bernilai *true* serta diprediksi *false* sebesar 1.



Gambar 4.23. Confusion Matrix

Model *Support Vector Machine* yang penulis kembangkan memiliki output yang menentukan pasien tersebut memiliki probabilitas 0 (tidak terkena penyakit

jantung) atau prob. 1 (terkena penyakit jantung), output didapat dari hasil kalkulasi sesuai index dataset yang dipilih. Contoh pasien index ke-1 dalam dataset memiliki Probabilitas 1 lebih tinggi sebesar 0.9169 dan Probabilitas 0 sebesar 0.0831, maka pasien index ke-1 ini memiliki risiko terkena penyakit jantung. Dalam output penelitian juga ditampilkan tingkat risiko berdasarkan nilai output pada probabilitas 1 dan 0 dengan contoh pasien index ke-1 termasuk dalam kategori “risiko Berat”.

Pada penelitian prediksi penyakit jantung menggunakan algoritma *Support Vector Machine* ini dihasilkan tingkat akurasi model *Support Vector Machine* yang digunakan sebesar 91.09 persen, presisi sebesar 86 persen, recall sebesar 95.56 persen, dan F1 score sebesar 90.53 persen. Dari hasil perhitungan metrik evaluasi tersebut, penulis dapat mengatakan bahwa penelitian ini cocok untuk melakukan prediksi penyakit jantung dengan algoritma dan dataset yang telah digunakan.

Penulis melakukan perbandingan dengan algoritma lain yang penelitiannya dilakukan oleh Victor Chang, Vallabhanent Rupa Bhavani, Qianwen Xu, dan Alamgir Hosain yang berjudul “An artificial intelligence model for heart disease detection using machine learning algorithms”, dihasilkan akurasi dari model algoritma yang berbeda, dataset dan ukuran tes yang sama yaitu sebanyak 100 data. Penelitian yang mereka lakukan memiliki hasil akurasi sebesar 87 persen untuk algoritma kNN, 79 persen untuk algoritma *decision tree*, 84 persen untuk algoritma *random forest*, dan sebesar 83 persen untuk algoritma SVM [25].

Perbandingan dengan penelitian yang dilakukan oleh Perbandingan dengan penelitian yang dilakukan oleh Dwi Sidik Permana dan Astried Silvanie dalam jurnal yang berjudul “Prediksi Penyakit Jantung menggunakan Support Vector Machine dan Python pada Basis Data Pasien di Cleveland” dihasilkan bahwa akurasi penelitian menunjukkan angka sebesar 90.11 persen [26].

Dari beberapa hasil penelitian, menunjukkan keberagaman hasil akurasi dari berbagai algoritma yang digunakan dalam penelitian. Kebanyakan dari hasil akurasi penelitian penelitian tersebut menunjukkan bahwa model algoritma SVM cocok karena menunjukkan bahwa angka akurasi tersebut selalu tinggi.

BAB 5

SIMPULAN DAN SARAN

5.1 Simpulan

Melalui penelitian penulis, penggunaan algoritma SVM menggunakan dataset “Heart Disease Dataset” dari situs kaggle, didapatkan dengan adanya 14 fitur yang berada dalam dataset. Fitur-fitur tersebut ialah *age*, *sex*, *cp*, *trestbps*, *chol*, *fbs*, *restecg*, *thalach*, *exang*, *oldpeak*, *slope*, *ca*, *thal*, dan *target*. 14 fitur inilah yang dapat mempengaruhi akurasi serta penentuan prediksi risiko penyakit jantung yang dimiliki. Pada penelitian ini penulis membuat 5 tingkatan skala risiko, yaitu, ‘Risiko Rendah’, ‘Risiko Sedang’, ‘Risiko Tinggi’, ‘Risiko Sangat Tinggi’, dan ‘Risiko Berat’.

Setelah melakukan perhitungan menggunakan metode SVM, dari 100 orang yang di tes melalui riwayat kesehatannya, terdapat 51 yang memiliki ‘Risiko Berat’, 2 yang memiliki ‘Risiko Sangat Tinggi’, 2 yang memiliki ‘Risiko Tinggi’, 3 yang memiliki ‘Risiko Sedang’, dan 43 yang memiliki ‘Risiko Rendah’.

Setelah melakukan penentuan risiko, akhirnya penulis mengukur akurasi dari algoritma SVM ini menggunakan dataset yang sama. Dimana hasil akurasinya mencapai hingga angka 91

Melihat dari hasil keseluruhan, dengan adanya 51 orang yang memiliki risiko berat, 51 orang tersebut dapat mencegah secara dini, agar risiko terkena penyakit jantung semakin menurun. Sehingga, tujuan penelitian penulis dapat dinyatakan berhasil.

5.2 Saran

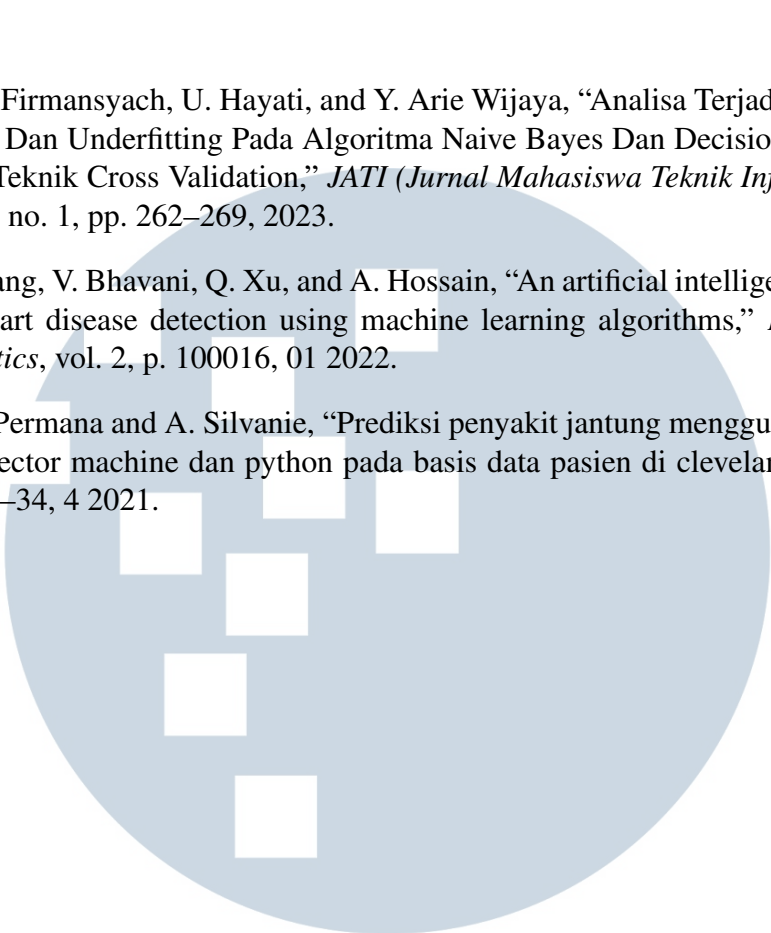
Setelah penulis selesai melakukan penelitian, terdapat beberapa hal yang ingin penulis sarankan agar penelitian dapat diteliti lebih baik kedepannya:

1. Menggunakan algoritma lain yang dapat memberikan hasil akurasi yang lebih tinggi apabila menggunakan data yang besar.
2. Menggunakan lebih banyak dataset agar dapat menghasilkan model yang lebih stabil dan akurat.

DAFTAR PUSTAKA

- [1] D. P. Utomo and M. Mesran, “Analisis komparasi metode klasifikasi data mining dan reduksi atribut pada data set penyakit jantung,” *Jurnal Media Informatika Budidarma*, 2020.
- [2] dr. Pittara, “Penyakit jantung,” 16 Maret 2023. [Online]. Available: <https://www.alodokter.com/penyakit-jantung/penyebab>
- [3] dr. Sienny Agustin, “Ciri-ciri sakit jantung yang harus diwaspadai,” 8 Mei 2022. [Online]. Available: <https://www.alodokter.com/Kenali-Ciri-Ciri-Sakit-Jantung>
- [4] W. H. Organization, “Who director-general’s opening remarks at the second annual gathering for the global ncd compact,” 21 September 2023. [Online]. Available: <https://www.who.int/director-general/speeches/detail/>
- [5] V. S. S. G. D. P. G. A. H. . G. J. Pouriyeh, S., “A comprehensive investigation and comparison of machine learning techniques in the domain of heart disease,” 2017.
- [6] I. R. I. A. Ary Putranto¹, Nuril Lutvi Azizah, “Sistem prediksi penyakit jantung berbasis web menggunakan metode svm dan framework streamlit,” *Jurnal Penerapan Sistem Informasi (Komputer Manajemen)*, p. 11, 2023.
- [7] Q. R. . W. D. Putri, “Angka kematian di eropa lebih banyak dibanding angka kelahiran,” 19 Jan 2016.
- [8] L. J. Miller and W. Lu, “These are the world’s healthiest nations,” *Bloomberg Journal*, 2019.
- [9] B. P. SANTOSO, “Heart rate monitor,” 10 2018.
- [10] D. G. Pradana, M. L. Alghifari, M. F. Juna, and S. D. Palaguna, “Klasifikasi penyakit jantung menggunakan metode artificial neural network,” *Indonesian Journal of Data and Science (IJODAS)*, vol. 3, pp. 55–60, 2022.
- [11] L. Ghani, M. D. Susilawati, and H. Novriani, “Faktor risiko dominan penyakit jantung koroner di indonesia,” *Buletin Penelitian Kesehatan*, vol. 44, 12 2016.
- [12] “Cardiovascular diseases (cvds).” [Online]. Available: [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
- [13] N. W. J. NARYADI, “Hubungan tingkat pengetahuan, tingkat dukungan keluarga dan tingkat kepatuhan diet pasien jantung pasca rawat inap di rumah sakit umum bangli,” 2019.

- [14] A. S. RAHARJO, "Perbandingan klasifikasi serangan jaringan distributed denial of service (ddos) menggunakan algoritma decision tree, k-nearest neighbors, bayesian network dan oner," 2020.
- [15] M. James and R. Dennis, "Machine-learning and statistical methods for ddos attack detection and defense system in software defined networks," 5 2021. [Online]. Available: <https://www.semanticscholar.org/paper/Machine-learning-and-statistical-methods-for-DDoS-Dennis/033a589571860de1b499ceab4a81a1c8da646de4>
- [16] A. G. Budianto and A. Syarief, "Analisis pengaruh pengurangan dimensi data pada keakuratan prediksi penyakit jantung dengan menggunakan svm linear," *Jurnal Taguchi : Jurnal Ilmiah Teknik dan Manajemen Industri*, vol. 3, pp. 80–91, 7 2023. [Online]. Available: <https://taguchi.lppmbinabangsa.id/index.php/home/article/view/58>
- [17] A. Hermawan, "Konsep dasar machine learning dan neural network dalam studi literatur — pdf." [Online]. Available: <https://id.scribd.com/document/627739208/44>
- [18] J. Andreanus, A. Kurniawan, K. M. V. D. Maitreya, S. Panas, and K. R. Indonesia, "Sejarah, teori dasar dan penerapan reinforcement learning: Sebuah tinjauan pustaka," *Jurnal Telematika*, vol. 12.
- [19] F. F. Handayanna, "Penerapan metode support vector machine menggunakan optimasi genetic algorithm untuk prediksi penyakit diabetes," *Jurnal Teknik Informatika*, vol. 1, pp. 139–147, 2015. [Online]. Available: <https://www.neliti.com/publications/495039/>
- [20] S. Y. Pangestu, Y. Astuti, and L. D. Farida, "Algoritma support vector machine untuk klasifikasi sikap politik terhadap partai politik indonesia," *Jurnal Mantik Penusa*, vol. 3, pp. 236–241, 2019. [Online]. Available: <https://t.co/eF>
- [21] E. Tasia, R. Zaid, I. Z. Ismail, S. Kenia, P. Loka, Y. Ikhsani, and R. Ocviani, "Sentimas: Seminar nasional penelitian dan pengabdian masyarakat classification of heart failure disease using supervised learning klasifikasi penyakit gagal jantung menggunakan supervised learning." [Online]. Available: <https://journal.irpi.or.id/index.php/sentimas>
- [22] A. S. Nugraha and K. K. Purnamasari, "Penerapan metode support vector machine pada part of speech tag bahasa indonesia."
- [23] A. Putranto, N. L. Azizah, I. Ratna, I. Astutik, F. Sains, and D. Teknologi, "Sistem Prediksi Penyakit Jantung Berbasis Web Menggunakan Metode SVM dan Framework Streamlit," *Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, vol. 4, no. 2, pp. 442–452, 2023. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/heart+disease>

- 
- [24] W. A. Firmansyach, U. Hayati, and Y. Arie Wijaya, “Analisa Terjadinya Overfitting Dan Underfitting Pada Algoritma Naive Bayes Dan Decision Tree Dengan Teknik Cross Validation,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 1, pp. 262–269, 2023.
- [25] V. Chang, V. Bhavani, Q. Xu, and A. Hossain, “An artificial intelligence model for heart disease detection using machine learning algorithms,” *Healthcare Analytics*, vol. 2, p. 100016, 01 2022.
- [26] D. S. Permana and A. Silvanie, “Prediksi penyakit jantung menggunakan support vector machine dan python pada basis data pasien di cleveland,” vol. 2, pp. 29–34, 4 2021.

