

**IMPLEMENTASI METODE *GAUSSIAN NAÏVE BAYES* UNTUK  
MEMBANGUN MODEL PREDIKSI SERANGAN JANTUNG**

**SKRIPSI**

**Oleh:**  
**MUHAMMAD IMAM GHOZALI**  
**NIM. 200605110154**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2024**

**IMPLEMENTASI METODE *GAUSSIAN NAÏVE BAYES* UNTUK  
MEMBANGUN MODEL PREDIKSI SERANGAN JANTUNG**

**SKRIPSI**

Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:  
**MUHAMMAD IMAM GHOZALI**  
NIM. 200605110154

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2024**

HALAMAN PERSETUJUAN

IMPLEMENTASI METODE *GAUSSIAN NAÏVE BAYES* UNTUK  
MEMBANGUN MODEL PREDIKSI SERANGAN JANTUNG

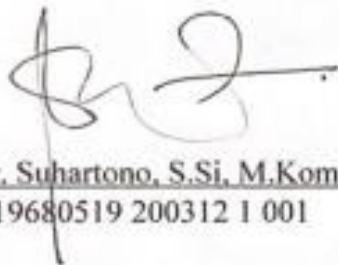
SKRIPSI

Oleh:

MUHAMMAD IMAM GHOZALI  
NIM. 200605110154

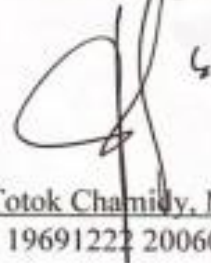
Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 6 Juni 2024

Pembimbing I,



Prof. Dr. Suhartono, S.Si, M.Kom  
NIP. 19680519 200312 1 001

Pembimbing II,




Dr. Totok Chamidy, M.Kom  
NIP. 19691222 200604 1 001

Mengetahui,

Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Taalim Kurniawan, M.MT, IPM  
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

IMPLEMENTASI METODE *GAUSSIAN NAÏVE BAYES* UNTUK  
MEMBANGUN MODEL PREDIKSI SERANGAN JANTUNG

SKRIPSI

Oleh:  
**MUHAMMAD IMAM GHOZALI**  
NIM. 200605110154

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Tanggal: 6 Juni 2024

Susunan Dewan Penguji

Ketua Penguji : Fatchurrochman, M.Kom  
NIP. 19700731 200501 1 002

Anggota Penguji I : Okta Qomaruddin Aziz, M.Kom  
NIP. 19911019 201903 1 013

Anggota Penguji II : Prof. Dr. Suhartono, S. Si., M.Kom  
NIP. 1968051 9200312 1 001

Anggota Penguji III : Dr. Totok Chamidy, M.Kom  
NIP. 19691222 200604 1 001

(  )  
(  )  
(  )  
(  )

Mengetahui dan Mengesahkan,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
**Dr. Fachrul Kurniawan, M.MT, IPM**  
NIP. 19771020 200912 1 001

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Imam Ghozali  
NIM : 200605110154  
Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika  
Judul Skripsi : Implementasi Metode *Gaussian Naïve Bayes* Untuk  
Membangun Model Prediksi Serangan Jantung

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 6 Juni 2024  
Yang membuat pernyataan,



Muhammad Imam Ghozali  
NIM. 200605110154

**HALAMAN MOTTO**

*“Don’t despair of the mercy of Allah”*

*“Bersungguh-sungguhlah engkau dalam menuntut ilmu,  
jauhilah kemalasan dan kebosanan karena jika tidak  
demikian engkau akan berada dalam bahaya kesesatan”*

*(Imam Al-Ghozali)*

## **HALAMAN PERSEMBAHAN**

Puji syukur kehadiran Allah SWT, atas limpahan rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi ini dengan lancar. Skripsi ini saya persembahkan untuk kedua orang tua saya yang selalu memberikan penuh kasih sayang, kesabaran, dan pengorbanan membesarkan saya hingga mencapai titik ini. Ibu Lutfiah dan Bapak Anang Afandy yang selalu memberikan dukungan dalam mencapai cita-cita saya dan mendoakan tiada henti yang menjadi sumber kekuatan saya dalam menyelesaikan studi ini. Kakak Diah Dina Aminata, Kakak Afifuddin dan Kakak Fahmi Fathoni yang sudah memberikan semangat, nasihat, dan kesabaran dalam memberikan arah. Seluruh keluarga yang tiada henti memberikan doa agar terselesaikan skripsi ini. Semoga Allah SWT selalu memberikan hal-hal baik kepada mereka.

## KATA PENGANTAR

*Assalamualaikum Warahmatullahi Wabarakatuh.*

Dengan penuh rasa syukur, penulis panjatkan puji syukur kehadiran Allah SWT atas limpahan rahmat dan karunia-Nya, serta shalawat dan salam semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW. Berkat berkah-Nya, penulis dapat menyelesaikan penulisan skripsi ini dengan judul “Implementasi Metode *Gaussian Naïve Bayes* Untuk Membangun Model Prediksi Serangan Jantung”. Skripsi ini disusun sebagai salah satu syarat untuk mencapai gelar sarjana di Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Ucapan rasa syukur dan terima kasih penulis sampaikan kepada seluruh pihak yang telah membantu berupa kritik dan saran agar terlesaikannya skripsi ini. Dengan rasa hormat penulis mengucapkan terima kasih kepada :

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT., IPM, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Prof. Dr. Suhartono, S. Si., M.Kom selaku dosen pembimbing I dan Dr. Totok Chamidy, M.Kom selaku dosen pembimbing II yang telah membimbing serta




memberikan bantuan dan arahan, sehingga penulis dapat menyelesaikan skripsi ini.

5. Fatchurrochman, M.Kom selaku dosen penguji I dan Okta Qomaruddin Aziz, M.Kom selaku dosen penguji II yang telah memberikan kritik dan saran, sehingga dapat menyelesaikan skripsi ini.
6. Segenap dosen dan jajaran staff Program Studi Teknik Informatika yang telah memberikan dukungan selama pengerjaan skripsi ini.
7. Kedua orang tua penulis yang saya cintai, serta kedua kakak saya yang tiada henti memberikan motivasi dan doa untuk menyelesaikan skripsi ini dengan baik.
8. Teman - teman saya yaitu Minoy (Almarhum Helmi), Azhar, Ulil, Khalid, Zaky, Priam, Yuss, Muzaki, Idris, Wisnu, Nanda Afiq, Nadia, Rila, Anisa, Alya yang memberikan dukungan dan motivasi untuk lulus skripsi tepat waktu serta menjadi teman yang baik selama saya di tanah perantauan.
9. Teman – Teman Integer Teknik Informatika 2020 yang selalu memberikan semangat dan doa kepada penulis.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan, dari keilmuan maupun penulisan. Maka dari itu, kritik dan saran yang membangun sangat diharapkan agar lebih baik lagi kedepannya. Semoga dengan penyusunan skripsi ini bisa memberikan manfaat bagi banyak pihak.

Malang, 7 Juni 2024



Muhammad Imam Ghozali  
NIM. 200605110154

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	ii
<b>HALAMAN PERSETUJUAN</b> .....	iii
<b>HALAMAN PENGESAHAN</b> .....	iv
<b>PERNYATAAN KEASLIAN TULISAN</b> .....	v
<b>HALAMAN MOTTO</b> .....	vi
<b>HALAMAN PERSEMBAHAN</b> .....	vii
<b>KATA PENGANTAR</b> .....	viii
<b>DAFTAR ISI</b> .....	x
<b>DAFTAR TABEL</b> .....	xii
<b>DAFTAR GAMBAR</b> .....	xiii
<b>ABSTRAK</b> .....	xiv
<b>ABSTRACT</b> .....	xv
مستخلص البحث .....	xvi
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	6
1.3 Batasan Masalah .....	6
1.4 Tujuan Penelitian .....	6
1.5 Manfaat Penelitian .....	6
<b>BAB II STUDI PUSTAKA</b> .....	7
2.1 Penelitian Terdahulu .....	7
2.2 Serangan Jantung .....	14
2.3 <i>Machine Learning</i> .....	15
2.4 Naïve Bayes .....	16
2.5 Gaussian Naïve Bayes .....	16
2.6 <i>K-fold cross validation</i> .....	17
2.7 <i>Confusion matrix</i> .....	18
<b>BAB III DESAIN DAN IMPLEMENTASI</b> .....	20
3.1 Desain Sistem .....	20
3.2 Pengumpulan Data .....	21
3.3 <i>Preprocessing Data</i> .....	23
3.3.1 Transformasi Data .....	23
3.3.2 <i>Split Data</i> .....	25
3.3.3 Normalisasi Data .....	26
3.4 <i>GridSearchCV</i> .....	30
3.5 Implementasi <i>Gaussian Naïve Bayes</i> .....	32
3.6 Evaluasi Performa .....	39
3.7 Skenario Pengujian .....	41
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	44
4.1 Langkah-Langkah Pengujian .....	44
4.2 Uji Coba <i>Split Data</i> .....	45
4.2.1 Hasil Pengujian Model A .....	46
4.2.2 Hasil Pengujian Model B .....	49

4.2.3 Hasil Pengujian Model C .....	51
4.2.4 Hasil Pengujian Model D.....	53
4.3 Uji Coba K-fold cross validation .....	55
4.3.1 K-fold cross validation K-5 .....	56
4.3.2 K-fold cross validation K-10 .....	57
4.3.3 K-fold cross validation K-15 .....	58
4.3.4 K-fold cross validation K-20 .....	59
4.4 Pembahasan.....	60
4.5 Integrasi Islam.....	64
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>69</b>
5.1 Kesimpulan.....	69
5.2 Saran	70
<b>DAFTAR PUSTAKA .....</b>	<b>72</b>

## DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu .....	10
Tabel 2. 2 <i>K-fold cross validation</i> .....	17
Tabel 2. 3 Contoh <i>Confusion Matrix</i> .....	19
Tabel 3. 1 Penjelasan Dataset.....	22
Tabel 3. 2 Contoh Dataset.....	22
Tabel 3. 3 Contoh dataset yang belum menjalani proses transformasi data .....	25
Tabel 3. 4 Contoh dataset yang telah menjalani proses transformasi data.....	25
Tabel 3. 5 Contoh dataset yang belum menjalani normalisasi.....	28
Tabel 3. 6 Contoh dataset yang sudah menjalani proses normalisasi .....	29
Tabel 3. 7 Contoh Confusion matrix.....	40
Tabel 3. 8 Pembagian data latih dan data uji .....	41
Tabel 3. 9 Contoh proses 5-fold cross validation.....	42
Tabel 4. 1 Hasil pengujian model A.....	47
Tabel 4. 2 Hasil nilai aktual dan prediksi model A.....	48
Tabel 4. 3 Jumlah nilai aktual dan prediksi model A.....	48
Tabel 4. 4 Hasil pengujian model B.....	49
Tabel 4. 5 Hasil nilai aktual dan prediksi model B .....	50
Tabel 4. 6 Jumlah nilai aktual dan prediksi model B.....	50
Tabel 4. 7 Hasil Pengujian Model C .....	51
Tabel 4. 8 Hasil nilai aktual dan prediksi model C .....	52
Tabel 4. 9 Jumlah nilai aktual dan prediksi model C.....	53
Tabel 4. 10 Hasil Pengujian Model D.....	53
Tabel 4. 11 Hasil nilai aktual dan prediksi model D.....	54
Tabel 4. 12 Jumlah nilai aktual dan prediksi model D.....	55
Tabel 4. 13 Hasil Accuracy 5-fold cross validation.....	56
Tabel 4. 14 Hasil Accuracy 10-fold cross validation.....	57
Tabel 4. 15 Hasil Accuracy 15-fold cross validation.....	58
Tabel 4. 16 Hasil Accuracy 20-fold cross validation.....	59
Tabel 4. 17 Pembahasan hasil skenario A.....	61
Tabel 4. 18 Pembahasan hasil skenario B.....	62
Tabel 4. 19 Pembahasan hasil skenario C.....	62
Tabel 4. 20 Pembahasan hasil skenario D.....	62

## DAFTAR GAMBAR

Gambar 3. 1 Desain Sistem.....	20
Gambar 3. 2 Implementasi Tahap Transformasi Data .....	24
Gambar 3. 3 Proses Implementasi Tahap <i>Splitdata</i> .....	26
Gambar 3. 4 Proses Implementasi Tahap Normalisasi Data.....	27
Gambar 3. 5 Proses Implementasi Tahap <i>GridSearchCV</i> .....	31
Gambar 3. 6 Proses Implementasi <i>Gaussian Naïve Bayes</i> .....	38
Gambar 4. 1 Hasil Confusion Matrix Model A.....	47
Gambar 4. 2 Hasil Confusion Matrix Model B.....	50
Gambar 4. 3 Hasil Confusion Matrix Model C.....	52
Gambar 4. 4 Hasil Confusion Matrix Model D.....	54
Gambar 4. 5 Grafik rata-rata akurasi K-fold cross-validation dengan Outliers....	63

## ABSTRAK

Ghozali, Muhammad Imam. 2024. **Implementasi Metode *Gaussian Naïve Bayes* Untuk Membangun Model Prediksi Serangan Jantung** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Suhartono, S. Si., M.Kom. (II) Dr. Totok Chamidy, M.Kom.

**Kata kunci:** Model Prediksi, *Gaussian Naïve Bayes*, Serangan Jantung, *K-Fold Cross Validation*

Penyakit jantung merupakan penyebab utama kematian di dunia, termasuk di Indonesia. Data dari WHO tahun 2015 menunjukkan 17 juta kematian global akibat penyakit ini, sementara di Indonesia sekitar 12,9% kematian disebabkan oleh penyakit jantung berdasarkan data RISKESDAS 2018 dan SRS 2014. Hal ini menunjukkan bahwa masih banyak orang yang tidak serius menanggapi penyebab penyakit jantung, sehingga banyak kasus ditemukan di stadium lanjut setelah pemeriksaan kesehatan. Cara mengatasi penyakit jantung dengan kondisi serupa meliputi operasi, penyinaran, dan kemoterapi, namun pencegahan tetap menjadi pilihan terbaik melalui gaya hidup sehat dan pemeriksaan rutin. Penelitian ini bertujuan untuk memprediksi serangan jantung menggunakan metode Gaussian Naive Bayes dengan memanfaatkan dataset rekam medis pasien dari Elsevier Mendeley Data Repository yang terdiri dari 1.319 data. Setelah melalui tahapan preprocessing seperti Label Encoding dan normalisasi data, dataset dibagi menjadi data latih dan data uji dengan rasio berbeda (90:10, 80:20, 70:30, 65:35). Evaluasi model menggunakan Confusion Matrix menunjukkan akurasi prediksi bervariasi dari 70% hingga 92%. Pengujian tambahan dengan K-fold cross validation (K=5, 10, 15, 20) menunjukkan akurasi tertinggi 88,26% pada K=5 dan terendah 57,95% pada K=15, dengan rata-rata akurasi tertinggi 70,43% pada K=5. Hasil penelitian ini menunjukkan bahwa model Gaussian Naive Bayes dapat memprediksi serangan jantung dengan baik, meskipun terdapat variasi akurasi berdasarkan pembagian data dan metode validasi yang digunakan.

## ABSTRACT

Ghozali, Muhammad Imam. 2024. **Implementation of the Gaussian Naïve Bayes Method for Building a Heart Attack Prediction Model**. Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Promotor: (I) Prof. Dr. Suhartono, S. Si., M.Kom. (II) Dr. Totok Chamidy, M.Kom.

**Keywords:** Prediction Model, *Gaussian Naïve Bayes*, Heart Attack, *K-Fold Cross Validation*.

Heart disease is the leading cause of death worldwide, including in Indonesia. WHO data from 2015 shows 17 million global deaths due to this disease, while in Indonesia, around 12.9% of deaths are caused by heart disease according to RISKESDAS 2018 and SRS 2014 data. This indicates that many people do not take the causes of heart disease seriously, resulting in many cases being discovered at an advanced stage after health examinations. Methods to address heart disease in similar conditions include surgery, radiation, and chemotherapy, but prevention remains the best option through a healthy lifestyle and regular check-ups. This study aims to predict heart attacks using the Gaussian Naive Bayes method by utilizing a medical record dataset from the Elsevier Mendeley Data Repository consisting of 1,319 data points. After preprocessing steps such as Label Encoding and data normalization, the dataset was divided into training and test data with different ratios (90:10, 80:20, 70:30, 65:35). Model evaluation using a Confusion Matrix showed prediction accuracy varying from 70% to 92%. Additional testing with K-fold cross validation (K=5, 10, 15, 20) showed the highest accuracy of 88.26% at K=5 and the lowest at 57.95% at K=15, with the highest average accuracy of 70.43% at K=5. The results of this study indicate that the Gaussian Naive Bayes model can predict heart attacks well, although there is variability in accuracy based on data partitioning and validation methods used.

## مستخلص البحث

غزالي، محمد إمام. 2024. تنفيذ طريقة غاوسيان ساذج بايز لبناء نموذج أطروحة للتنبؤ بالنوبات القلبية. برنامج دراسة هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية، مالانج. المشرف: (1) الأستاذ الدكتور سوهارتونو الماجستير، (2) الدكتور توتوك حامدي الماجستير

الكلمات المفتاحية: نموذج التنبؤ، غاوسيان ساذج بايز، نوبة قلبية، التحقق من صحة الصليب **K-fold**

مرض القلب هو السبب الرئيسي للوفاة في العالم، بما في ذلك في إندونيسيا. أظهرت بيانات منظمة الصحة العالمية في عام 2015 وجدت 17 مليون حالة وفاة عالمية بسبب هذا المرض، بينما في إندونيسيا حوالي 12.9% من الوفيات كانت ناجمة عن أمراض القلب بناء على بيانات **2018 RISKESDAS** و **2014 SRS**. هذا يدل على أنه لا يزال هناك الكثير من الأشخاص الذين لا يأخذون سبب مرض القلب على محمل الجد، لذلك يتم العثور على العديد من الحالات في مرحلة متقدمة بعد الفحص الطبي. تشمل طرق التعامل مع أمراض القلب ذات الحالات المماثلة الجراحة والإشعاع والعلاج الكيميائي، لكن الوقاية تظل الخيار الأفضل من خلال نمط حياة صحي وفحوصات منتظمة. تهدف هذه الدراسة إلى التنبؤ بالنوبة القلبية باستخدام طريقة **Gaussian Naive Bayes** من خلال استخدام مجموعة بيانات من السجلات الطبية للمرضى من مستودع بيانات **Mendeley Elsevier** الذي يتكون من 1,319 بيانات. بعد المرور بمراحل المعالجة المسبقة مثل ترميز المصنقات وتطبيع البيانات، تم تقسيم مجموعة البيانات إلى بيانات تدريب وبيانات اختبار بنسب مختلفة (10:90، 20:80، 30:70، 35:65). أظهر تقييم النموذج باستخدام مصفوفة الارتباك أن دقة التنبؤ تراوحت من 70% إلى 92%. أظهرت الاختبارات الإضافية مع التحقق المتقاطع **K-fold** (5، 10، 15، 20) أعلى دقة بنسبة 88.26% عند  $K = 5$  وأدنى دقة بنسبة 57.95% عند  $K = 15$ ، مع أعلى متوسط دقة 70.43% عند  $K = 5$ . تظهر نتائج هذه الدراسة أن نموذج **Gaussian Naive Bayes** يمكنه التنبؤ بالنوبات القلبية بشكل جيد، على الرغم من وجود اختلافات في الدقة بناء على مشاركة البيانات وطرق التحقق المستخدمة.



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Jantung adalah otot yang melakukan banyak hal penting untuk sistem peredaran darah manusia, Jantung memompa darah ke seluruh tubuh melalui kontraksi berirama yang terus-menerus, Hal ini berfungsi untuk menyediakan oksigen dan nutrisi untuk sel tubuh dan juga membantu mengeluarkan sisa metabolisme (Yunanto Setyaji dkk., 2018). Jantung sendiri merupakan salah satu organ tubuh yang paling penting karena peranannya yang sangat penting untuk menjaga kehidupan. Jika terdapat kelainan atau gangguan pada jantung dapat memiliki dampak yang cukup serius bahkan berbahaya hingga menyebabkan kematian (Harbanu H Mariyono, 2007).

ثُمَّ لَقَطَعْنَا مِنْهُ الْوَتِينَ ۗ

*“Kemudian, benar-benar kami potong urat tali jantungnya.” (Q.S Al-Haqqah : 46)*

Menurut Tafsir dari Ibnu Katsir, Pembuluh nadi dalam hal ini merupakan pembuluh darah besar atau dalam dunia medis disebut aorta yang tekanannya langsung dari kontraksi jantung dan volume darah yang besar, tersumbat atau terpotongnya aorta dapat mengakibatkan syok bahkan kematian (Tafsir Ibnu Katsir, n.d.). Ayat di atas menunjukkan pentingnya pemahaman tentang fungsi dan kerentanan jantung dalam konteks risiko serangan jantung. Dalam penelitian ini, ayat di atas dihubungkan dengan pemahaman akan akibat yang cukup serius dari gangguan pada jantung..

Penyakit jantung merupakan tantangan besar dalam dunia kesehatan global. Menurut data dari *World Health Organization* (WHO) Jumlah kematian karena penyakit jantung dan kerusakan pembuluh darah mencapai lebih dari 17 juta orang di seluruh dunia pada tahun 2015, yang setara dengan 31% total kematian di dunia dalam tahun 2015. Selain itu, sekitar 8,7 juta orang meninggal karena penyakit jantung, yang merupakan masalah kesehatan yang signifikan bagi masyarakat (Riani et al., 2019). Fenomena ini terutama terlihat di negara-negara berkembang, di mana rata-rata populasi memiliki penghasilan menengah ke bawah. Hal ini menunjukkan bahwa kondisi sosioekonomi dan prevalensi penyakit jantung berkorelasi. Dalam munculnya masalah kesehatan ini, hal-hal seperti akses terhadap layanan kesehatan, pola makan, gaya hidup, dan lingkungan harus sangat diperhatikan (Nursita & Pratiwi, 2020).

Di Indonesia, situasi terkait penyakit jantung juga mengkhawatirkan. Penyakit ini tidak lagi hanya terjadi pada usia lanjut, tetapi juga telah menjangkiti populasi yang lebih muda. Berdasarkan data yang telah dikumpulkan oleh Riset Kesehatan Dasar (RISKESDAS) di tahun 2018, sekitar 15 dari 1.000 orang di Indonesia mengalami penyakit jantung. Selain itu, *Survei Sample Registration System* (SRS) pada 2014 menunjukkan penyakit jantung menyumbang 12,9% kematian di Indonesia (*Laporan Risesdas 2018 Nasional.Pdf*, n.d.).

Berdasarkan data diatas, Bisa kita simpulkan bahwa masih terdapat banyak orang yang tidak serius menanggapi penyebab penyakit jantung, yang menyebabkan banyak kasus ditemukan di stadium lanjut setelah pemeriksaan kesehatan. cara mengatasi penyakit jantung dengan kondisi serupa terdapat pilihan

pengobatan yaitu operasi, penyinaran, dan khemoterapi. Namun, Pencegahan tetap menjadi pilihan terbaik dengan cara menjaga gaya hidup sehat dan pemeriksaan rutin untuk deteksi dini. Seperti yang sudah diterangkan dalam Al-qur'an (QS.Yunus Ayat 57) yang berbunyi :

يَا أَيُّهَا النَّاسُ قَدْ جَاءَكُمْ مَوْعِظَةٌ مِنْ رَبِّكُمْ وَشِفَاءٌ لِمَا فِي الصُّدُورِ وَهُدًى وَرَحْمَةٌ لِّلْمُؤْمِنِينَ

*“Wahai manusia, sungguh telah datang kepadamu pelajaran (Al-Qur'an) dari Tuhanmu, penyembuh bagi sesuatu (penyakit) yang terdapat dalam dada, dan petunjuk serta rahmat bagi orang-orang mukmin.” (QS.Yunus : 57)*

Menurut tafsir kitab Al-imam Ibnu Katsir, setelah diingatkan bahwa semua yang hidup pasti akan mati dan akan kembali kepada Allah, manusia diingatkan, "Wahai manusia! Sungguh, telah datang kepadamu pelajaran berupa Kitab Suci Al-Qur'an dari Tuhanmu, obat untuk penyakit yang ada di dalam hatimu, petunjuk menuju kebenaran, dan rahmat yang besar bagi orang yang benar-benar beriman." Ini adalah peringatan untuk perbuatan buruk.

Kemudian dalam Al-Qur'an juga dijelaskan ayat yang berhubungan dengan klasifikasi pada surah Al-An'am ayat 141 :

وَهُوَ الَّذِي أَنشَأَ جَنَّاتٍ مَّعْرُوشَاتٍ وَغَيْرَ مَعْرُوشَاتٍ وَالنَّخْلَ وَالزَّرْعَ مُخْتَلِفًا أَكْلُهُ ۗ وَالزَّيْتُونَ وَالرَّيْحَانَ مُتَشَابِهًا ۗ وَغَيْرَ مُتَشَابِهٍ ۗ كُلُوا مِنْ ثَمَرِهِ إِذَا أَثْمَرَ وَآتُوا حَقَّهُ ۗ وَلَا تَسْرِفُوا ۚ إِنَّهُ لَا يُحِبُّ الْمُسْرِفِينَ

*“Dan dialah yang menumbuhkan tanaman-tanaman yang merambat dan yang tidak merambat, pohon kurma, tanaman yang beraneka ragam rasanya, serta zaitun dan delima yang serupa (bentuk dan warnanya) dan tidak serupa (rasanya). Makanlah buahnya apabila ia berbuah dan berikanlah haknya (zakatnya) pada waktu memetik hasilnya. Akan tetapi, janganlah berlebih-lebihan. Sesungguhnya Allah tidak menyukai orang-orang yang berlebih-lebihan”(QS. Al-An'am :141).*

Menurut tafsir kitab al-imam Ibnu Katsir, Allah SWT menyatakan dalam firman-Nya bahwa Dia adalah Yang menciptakan semua tanaman, buah-buahan, dan binatang ternak. Orang-orang musyrik memperlakukan ternak mereka dengan cara yang salah sesuai dengan keyakinan mereka yang salah. Mereka membaginya menjadi beberapa kategori dan membaginya menjadi haram dan halal. "Dan Dialah yang menciptakan kebun-kebon yang berjunjung dan yang tidak berjunjung," kata Allah. Menurut riwayat dari sahabat Ali bin Abu Talhah (Al-An'am: 141), "ma'rusyatin" berarti yang merambat. Dalam riwayat lain, "ma'rusyatin" berarti tanaman yang ditanam oleh manusia. Namun, "ghairu marusyatin" berarti tanaman berbuah yang tumbuh sendiri di bukit-bukit dan hutan. Menurut Atha' Al-Khurasani, "ma'rusyatin" berarti tanaman anggur yang ditanam, sedangkan "ghairu ma'rusyatin" berarti tanaman anggur yang tidak ditanam. As-Suddi juga mengatakan hal yang sama. Sehubungan dengan makna firman-Nya, "Yang serupa dan yang tidak serupa" (Al-An'am: 141), Ibnu Juraij mengatakan bahwa itu berarti yang serupa secara bentuk tetapi tidak sama secara rasa. Sehubungan dengan makna firman-Nya: "Makanlah buahnya bila berbuah" (Al-An'am: 141), Muhammad bin Ka'b mengatakan bahwa ini mengacu pada buah kurma dan anggur.

Dengan perkembangan teknologi yang sudah canggih ini, terdapat salah satu solusi yang bisa diterapkan yaitu membuat sebuah sistem prediksi penyakit serangan jantung yang akan berguna untuk memberikan informasi kepada masyarakat. serta dapat melakukan pengecekan prediksi secara dini terkait penyakit serangan jantung yang diderita oleh pasien (Alhamad et al., 2019). Pada proses untuk membuat sebuah *system* prediksi penyakit serangan jantung terdapat

beberapa metode dalam *Machine learning* yang secara umum dipakai diantaranya adalah *Naive bayes*, *Random forest*, *Support Vector Machine (SVM)*, *KNN (K-Nearest Neighbor)*. Pada penelitian ini, penulis menggunakan salah satu metode tersebut yaitu *Naïve Bayes* dengan tipe *Gaussian*.

Secara umum, *Gaussian naive bayes* (GNB) adalah salah satu varian dari algoritma *Naive bayes* Untuk klasifikasi data. Meskipun metode ini cukup sederhana, itu seringkali berhasil menyelesaikan berbagai masalah klasifikasi, terutama jika fiturnya kontinu atau numerik yang biasanya digunakan oleh GNB untuk mengkategorikan instance data ke dalam kelas atau kategori yang telah ditetapkan sebelumnya. Namun, GNB juga memiliki asumsi bahwa fitur-fitur dalam data diambil dari distribusi normal (*Gaussian*). Ini berarti bahwa nilai-nilai fitur berada dalam distribusi *Gaussian*, atau dalam kata lain, mengikuti kurva lonceng. Dengan asumsi ini, GNB dapat menggunakan distribusi *Gaussian* untuk menghitung probabilitas dari setiap fitur terhadap setiap kelas. Bersama dengan probabilitas prior dari setiap kelas dan fitur-fiturnya, GNB dapat menghitung

prediksi dini penyakit serangan jantung, peneliti memakai metode *Machine learning* yaitu *Gaussian naive bayes*. Berdasarkan faktor-faktor risiko dari variabel yang berpotensi dan berkontribusi pada penyakit tersebut, akan digunakan untuk memprediksi kemungkinan terjadinya serangan jantung dengan menggunakan metode *Gaussian Naïve Bayes*. penelitian ini bertujuan untuk menguji seberapa baik akurasi pada metode *Gaussian naive bayes* dalam sistem prediksi penyakit serangan jantung.

## 1.2 Rumusan Masalah

Seberapa tinggi tingkat *accuracy*, *Precision*, *Recall*, *F1-score* metode *Gaussian Naïve Bayes* dalam memprediksi penyakit serangan jantung ?

## 1.3 Batasan Masalah

1. Menggunakan data publik yang diperoleh melalui Elsevier Mendeley Data Repository: *Heart Attack Dataset* yang terdiri dari 1319 data pasien serta memiliki 9 fitur yaitu *Age*, *Gender*, *Heart rate*, *Systolic BP*, *Diastolic BP*, *Blood Sugar*, *CK-MB*, *Troponin*, *Result* dan hanya 8 fitur yang digunakan.
2. Model ini dilakukan hanya bertujuan untuk memprediksi risiko positif dan negatif terjadinya serangan jantung berdasarkan fitur dataset.

## 1.4 Tujuan Penelitian

Tujuan penelitian dilakukan untuk mengukur tingkat akurasi metode *Gaussian naive bayes* pada sistem prediksi serangan jantung berdasarkan faktor resiko.

## 1.5 Manfaat Penelitian

1. Diharapkan dapat memberikan manfaat untuk prediksi resiko penyakit serangan jantung dengan tingkat akurasi yang sesuai.
2. Dapat memberikan pengetahuan dalam implementasi metode *Gaussian Naïve Bayes* untuk prediksi penyakit serangan jantung.
3. Dapat dijadikan landasan bagi peneliti selanjutnya.

## BAB II

### STUDI PUSTAKA

#### 2.1 Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Quswatun hasanah dkk, (2022), Penelitian ini menggunakan *dataset Kaggle* yang terdiri dari tiga puluh variabel yang dikumpulkan dari data 1000 pasien yang menderita gagal jantung. Untuk tahap skenario uji coba, *Cross Fold Validation* digunakan, dengan nilai  $k = 2, 4, 5,$  dan 10. Untuk menilai hasil klasifikasi, matriks confusion digunakan terhadap output data awal, atau nilai aktual. Setelah menggunakan algoritma *Gaussian naive bayes* pada data pasien yang menderita gagal jantung, hasil validasi terbaik dicapai pada *fold* 10 tahap 9 dengan skenario 4, dengan akurasi 69%. Pada *fold* 4 tahap 4 dengan skenario 2, hasil presisi terbaik mencapai 65,73%, dan nilai *Recall* tertinggi dicapai pada *fold* 10 tahap 4 dengan skenario 4, dengan nilai 95,91%.

Berdasarkan penelitian yang dilakukan oleh (Rizkia, 2023), Dengan menggunakan metode (GNB) *Gaussian Naïve Bayes*. Tujuan penelitian ini adalah mengembangkan sistem untuk mengidentifikasi jenis tanaman obat berdasarkan gambar daun, dengan metode deteksi tepi dan Gaussian Naïve Bayes (GNB) Parameter rata-rata dan varian fitur yang digunakan penelitian ini akan diidentifikasi menggunakan metode GNB. Hasil pengujian menunjukkan performa terbaik pada pembagian data, dengan rasio 90:10 untuk 6 kelas daun, Memiliki *accuracy* 90%, *Precision* 92,46%, *Recall* 90%, *f-measure* 89,69%. Namun, pada uji coba dengan 16 kelas daun, akurasi turun menjadi 57,50%. Penurunan ini disebabkan oleh rentang nilai fitur-fitur yang sangat dekat di antara

jenis daun. Oleh karena itu, pilihan jenis fitur bentuk dan jumlah kelas sangat memengaruhi kinerja sistem.

Berdasarkan Penelitian yang dilakukan oleh Kamel Al-tuwaijari, (2019), Metode *Gaussian Naïve Bayes* yang berguna untuk membantu dalam melakukan proses diagnosis pada penyakit kanker. Serta bertujuan untuk melihat hasil klasifikasi dengan meimplementasikan metode *Gaussian Naïve Bayes* pada pembuatan sistem klasifikasi penyakit kanker. Dan juga untuk mengetahui nilai akurasi dari atribut yang sesuai dengan *dataset* yang dipakai. Terdapat dua *dataset* digunakan. Yang pertama adalah *Wisconsin Breast Cancer (WBCD)*, dan yang kedua adalah kumpulan data kanker paru-paru. Hasil evaluasi menunjukkan bahwa algoritma yang diusulkan memiliki tingkat akurasi yang tinggi dalam memprediksi kanker, dengan hasil mencapai 98% untuk memprediksi kanker payudara dan memiliki hasil 90% untuk memprediksi kanker paru-paru, hal ini menunjukkan bahwa algoritma tersebut memiliki kinerja yang baik dalam memprediksi kedua jenis kanker berdasarkan data yang diberikan.

Berdasarkan penelitian yang dilakukan oleh Ulfatul dkk, (2022), Dalam studi ini dilakukan perbandingan dua metode klasifikasi yaitu *algoritma K-Nearest Neighbor* dan *Gaussian naive bayes* Sebagai metode untuk membuat sistem prediksi penyakit stroke. Hasil menunjukkan bahwa algoritma *K-Nearest Neighbor* memiliki *accuracy* 68,30%, *Precision* 67,20%, dan *Recall* 73,34%, sementara algoritma *Gaussian naive bayes* memiliki *accuracy* 74,45%, *Precision* 74,01%, dan *Recall* 75,71%. Dari sini kesimpulannya penggunaan metode *Gaussian Naïve Bayes* terbukti memberikan hasil yang lebih akurat.



Berdasarkan penelitian yang dilakukan oleh Octaviary, (2022), Penggunaan metode *Naïve Bayes* dilakukan untuk mengukur tingkat akurasi dalam mendeteksi awal penyakit gagal jantung berdasarkan faktor resiko. Penyakit gagal jantung sendiri berarti jantung tidak memompa darah dengan baik. Sedangkan Serangan jantung terjadi pada saat aliran darah menuju jantung terkena hambatan, merusak jaringannya. Data yang digunakan diperoleh melalui situs *Kaggle*, Data diproses dengan melakukan perubahan jenis data kategori menjadi numerik, meningkatkan, dan memisahkan. Untuk memisahkan data latih dan uji, ada empat model rasio: model A memiliki rasio 90:10, model B memiliki rasio 80:20, model C memiliki rasio 75:25, dan model D memiliki rasio 70:30. Rasio yang paling tepat untuk memprediksi adalah model A dengan rasio 80:10 yaitu menghasilkan nilai 75.26% yang dikategorikan cukup. Sedangkan, hasil nilai akurasi dari proses *10-fold cross validation* dengan nilai k sama dengan 10, sementara hasil nilai akurasi dari proses *cross-validation* sepuluh kali dengan penentuan nilai k 10, yang berarti sebanyak sepuluh kali proses iterasi, menghasilkan nilai ketepatan prediksi data optimal dengan nilai 84,37%.

Berdasarkan penelitian yang dilakukan oleh Manikandan dkk, (2017), untuk mengukur tingkat akurasi dari penggunaan metode *Gaussian Naïve Bayes* dalam pembuatan sistem prediksi penyakit serangan jantung. Data mengambil *dataset* dari *UCI's (Irvine's Machine learning repository)* dengan kumpulan data yang terdiri dari total 14 atribut, dan 13 variabel yang digunakan. Hasil dari penggunaan metode *Gaussian Naïve Bayes* dalam penelitian ini mendapatkan hasil akurasi terbaik sebesar 81,25%.

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti (tahun)	Judul	Metode Penelitian	Hasil Penelitian	Perbedaan
1.	Quswatun hasanah dkk, (2022)	Analisis Algoritma <i>Gaussian naive bayes</i> Terhadap Klasifikasi Data Pasien Penderita Gagal Jantung	<i>Gaussian naive bayes</i>	Hasil dari penggunaan metode <i>Gaussian Naive Bayes</i> untuk klasifikasi pasien penderit gagal jantung ini mendapatkan hasil validasi terbaik dicapai pada <i>fold</i> 10 tahap 9 dengan skenario 4, dengan akurasi 69%. Pada <i>fold</i> 4 tahap 4 dengan skenario 2, hasil presisi terbaik mencapai 65,73%, dan nilai <i>Recall</i> tertinggi dicapai pada <i>fold</i> 10 tahap 4 dengan skenario 4, dengan nilai 95,91%.	Dalam jurnal ini peneliti menggunakan objek berbeda yang terdiri dari data penderita pasien gagal jantung pada tahun 2020.
2.	Rizkia, (2023)	Identifikasi Jenis Tanaman Obat Indonesia Berdasarkan Bentuk Pada	Metode Deteksi Tepi dan <i>Gaussian Naive Bayes</i>	Parameter rata-rata dan varian fitur ini akan diidentifikasi menggunakan	Dalam penelitian ini menggunakan studi kasus yang berbeda yaitu

		Citra Daun Menggunakan Metode Deteksi Tepi dan <i>Gaussian Naïve Bayes</i>		n metode GN B. Hasil pengujian menunjukkan performan terbaik pada pembagian data, dengan rasio 90:10 untuk 6 kelas daun, Memiliki akurasi 90%, presisi 92,46%, <i>Recall</i> 90%, f-measure 89,69%. Namun, pada uji coba dengan 16 kelas daun, akurasi turun menjadi 57,50%. Penurunan ini disebabkan oleh rentang nilai fitur-fitur yang sangat dekat di antara jenis daun.	mengidentifikasi jenis tanaman obat berdasarkan bentuk citra daun dan menggunakan deteksi tepi. <i>Dataset</i> terdiri dari 16 jenis daun, masing-masing jenis memiliki jumlah 50 data gambar. Setiap gambar memiliki fitur bentuk seperti kerampingan, panjang, lebar, luas, dan keliling daun
3.	Kamel Al-tuwaijari, (2019)	Cancer Classification Using <i>Gaussian naive bayes</i> Algorithm	<i>Gaussian naive bayes</i>	Algoritma yang diusulkan mendapat tingkat akurasi tinggi, mencapai 98% untuk kanker payudara dan 90% untuk	Dalam jurnal ini peneliti menggunakan objek berbeda, yaitu menggunakan data pasien penderita penyakit kanker dan mengklasifikasi 2

				kanker paru-paru. Ini menandakan kinerja yang baik dalam memprediksi kedua jenis kanker berdasarkan data yang tersedia.	jenis penderita penyakit kanker payudara dan kanker paru paru.
4.	Ulfatul dkk, (2022)	Perbandingan Metode K-Nearest Neighbor Dan <i>Gaussian naive bayes</i> Untuk Klasifikasi Penyakit Stroke	Metode K-Nearest Neighbor Dan <i>Gaussian naive bayes</i>	Dalam studi ini membandingkan dua metode klasifikasi yaitu <i>algoritma K-Nearest Neighbor dan Gaussian naive bayes</i> . Hasil menunjukkan bahwa algoritma K-Nearest Neighbor memiliki akurasi 68,30%, presisi 67,20%, dan <i>Recall</i> 73,34%, sementara algoritma <i>Gaussian naive bayes</i> memiliki akurasi 74,45%, presisi 74,01%, dan	Dalam jurnal ini menggunakan objek berbeda, yaitu menggunakan data dari pasien penderita penyakit stroke dan bertujuan untuk membandingkan 2 metode, yaitu K-Nearest Neighbor dan <i>Gaussian Naïve Bayes</i> .

				<i>Recall</i> 75,71%.	
5.	Octaviary, (2022)	Deteksi awal penyakit gagal jantung berdasarkan faktor resiko menggunakan metode <i>Naïve Bayes</i> .	<i>Naïve Bayes</i>	Dalam penelitian ini digunakan 4 model scenario ujicoba dan prediksi data yang terbaik didapatkan dari model B dengan rasio 80:20 menghasilkan nilai 75.26% yang dikategorikan cukup. Sedangkan, hasil nilai akurasi dari proses 10- <i>fold cross validation</i> dengan penentuan nilai k 10 yang berarti sebanyak 10 kali proses iterasi. Dari proses iterasi kesepuluh kali menghasilkan nilai ketepatan prediksi data yang optimal dihasilkan dari nilai k 9 dengan nilai sebesar 84,37%.	Perbedaan dalam penelitian ini data yang digunakan merupakan <i>dataset</i> gagal jantung yang diperoleh dari kaggle. Dan metode yang digunakan adalah <i>Naïve Bayes</i> yang digunakan secara umum yang tidak mengasumsikan distribusi khusus.
6.	Manikandan, (2017)	<i>Heart attack prediction system</i>	<i>Gaussian Naïve Bayes</i>	Hasil dari penggunaan metode <i>Gaussian Naïve Bayes</i> dalam penelitian ini mendapatkan hasil	Dalam jurnal ini studinya sama namun data yang dipakai berbeda yaitu diambil dari UCI's (Irvine's

				akurasi terbaik sebesar 81,25%.	<i>Machine learning repository</i> ) dengan kumpulan data terdiri dari total 14 atribut, dan 13 variabel yang digunakan.
--	--	--	--	---------------------------------	--

## 2.2 Serangan Jantung

Serangan jantung, atau infark miokard akut (IMA), terjadi ketika aliran darah ke otot jantung terhenti secara tiba-tiba, menyebabkan kekurangan oksigen yang signifikan, yang dapat menyebabkan kematian atau kerusakan jaringan otot jantung. Penanganan segera dan tepat diperlukan karena serangan jantung adalah kondisi medis darurat (Ketut et al., 2022). Ketika plak aterosklerosis atau gumpalan darah tersumbat pada arteri koroner yang mengalirkan darah ke jantung, aliran darah ke jaringan otot jantung terhenti. Sel-sel jantung cepat mati tanpa oksigen. Ini dapat menyebabkan berbagai gejala seperti sesak napas, mual, muntah, nyeri dada yang parah, atau angina, dan rasa tidak nyaman di bagian tubuh lainnya (Puspa Wardhani, 2016).

Faktor resiko utama dari penyakit serangan jantung diantaranya adalah Merokok, kolesterol tinggi, tekanan darah tinggi, diabetes, obesitas, kurangnya aktivitas fisik, dan stress. Merokok dapat mengakibatkan rusaknya pembuluh darah dan akan menyebabkan pembentukan plak aterosklerosis. kadar kolesterol tinggi dalam darah juga menyebabkan meningkatnya risiko penumpukan plak di arteri. dan diabetes merusak pembuluh darah (Kasus et al., 2021). Kesehatan

jantung dan pembuluh darah juga terpengaruh oleh obesitas, Serta dengan kurangnya Aktivitas fisik juga dapat mengurangi kebugaran jantung dan faktor risiko lainnya. Selain itu, stres jangka panjang dapat meningkatkan tekanan darah dan menyebabkan reaksi biokimia yang berbahaya bagi jantung. Risiko seseorang terkena serangan jantung juga dipengaruhi oleh genetika, usia, dan jenis kelamin. Untuk menghindari serangan jantung dan menjaga kesehatan jantung yang baik, sangat penting untuk menyadari faktor risiko ini dan mengelolanya melalui perubahan gaya hidup dan pengawasan medis yang tepat (Amrullah et al., 2022).

### **2.3 *Machine Learning***

*Machine learning* adalah cabang dari kecerdasan buatan yang mengacu pada pengembangan sistem komputer yang dapat belajar dari data, mengidentifikasi pola, dan membuat keputusan dengan minimal intervensi manusia. Tujuan utama *Machine learning* adalah untuk mengembangkan algoritma yang dapat mengenali pola dalam data, membuat prediksi atau keputusan berdasarkan pola tersebut, dan secara otomatis meningkatkan kinerjanya seiring dengan lebih banyak data. Adapun *Machine learning* memiliki beberapa proses pembelajaran dengan melibatkan Pengumpulan data, pemrosesan, pembuatan model, dan evaluasi kinerja model (Wahyudi, 2023). *Machine learning* dalam bidang kesehatan atau medis dapat mempermudah dalam mengerjakan sesuatu sebagai contoh dokter bisa mendiagnosa penyakit jantung dengan cepat tanpa membutuhkan waktu lama. Selain itu, para medis juga memiliki keuntungan besar yang dapat memudahkan pekerjaannya sebagai petugas medis dalam menghadapi pasien (Dodo et al., 2019).

## 2.4 *Naïve Bayes*

*Naive bayes* adalah metode klasifikasi yang populer dan umum digunakan dalam pembelajaran mesin. Itu sering digunakan untuk klasifikasi teks dan telah terbukti efektif dalam banyak bidang. *Naive bayes* baik karena mudah digunakan dan efektif. Keunggulan utama *Naive bayes* adalah mudah digunakan dan efektif dalam banyak situasi. Untuk melakukan klasifikasi, *Naive bayes* menggunakan teorema Bayes, yang menentukan kemungkinan suatu kelas diperkirakan berdasarkan kemungkinan fitur yang diamati. Meskipun demikian, perlu diingat bahwa *Naive bayes* sangat memperhatikan pemilihan fitur. Terlalu banyak fitur dapat menyebabkan waktu komputasi yang lebih lama dan akurasi klasifikasi yang lebih rendah. Oleh karena itu, untuk mendapatkan hasil terbaik dari penggunaan *Naive bayes*, pemilihan fitur yang tepat sangat penting (Susana & Suarna, 2022).

## 2.5 *Gaussian Naïve Bayes*

Metode *Gaussian Naïve Bayes* digunakan untuk klasifikasi. Ini mengasumsikan distribusi normal (*Gaussian*) apabila ditemukan data kontinu. Data dibagi menjadi kelas selama proses pelatihan, dan rata-rata dan *standard deviation* fitur kontinu untuk setiap kelas dihitung (Azizah & Goejantoro, 2019). Selanjutnya, ketika model *Gaussian naive bayes* digunakan untuk memprediksi kelas data baru, algoritma ini akan menggunakan distribusi *Gaussian* yang sudah dihitung sebelumnya untuk mengestimasi probabilitas data kontinu tersebut berasal dari setiap kelas. Dengan menggunakan rata-rata dan *standard deviation* yang telah diketahui, algoritma menghitung nilai probabilitas data baru berasal dari setiap kelas. Untuk klasifikasi data baru, probabilitas untuk setiap kelas dibandingkan, dan



kelas dengan probabilitas tertinggi dianggap sebagai kelas yang paling mungkin untuk data tersebut. Dengan menggunakan distribusi *Gaussian* dari fitur-fitur kontinu dan asumsi independensi antar-fitur yang ada dalam algoritma *Naïve Bayes*, metode *Gaussian Naïve Bayes* memungkinkan klasifikasi data baru berdasarkan estimasi probabilitas.

## 2.6 *K-fold cross validation*

Definisi validasi silang (*K-fold cross validation*) adalah teknik validasi silang yang umum digunakan untuk menentukan nilai  $k$  yang optimal dalam prediksi dan melakukan kegiatan memprediksi model dan estimasi tingkat akurat model ketika sedang dijalankan. Data awal digunakan sebagai bahan prediksi untuk metode validasi data ini, dan dibagi menjadi  $k$  *subset* dengan ukuran yang sesuai dengan persyaratan dan dilakukan secara acak. Selama proses ini, melakukan pelatihan dan pengujian sebanyak  $k$  kali. Pada iterasi  $i$ , bagian tempat dimasukkan ke dalam data uji dan partisi yang tersisa dimasukkan ke dalam data pelatihan (Octaviary, 2022).

Tabel 2. 2 *K-fold cross validation*

<i>Fold</i>	<i>5-fold cross validation</i>				
<b>1</b>	1 (test)	2 (train)	3 (train)	4(train)	5 (train)
<b>2</b>	1 (train)	2 (test)	3 (train)	4(train)	5 (train)
<b>3</b>	1 (train)	2 (train)	3 (test)	4(train)	5 (train)
<b>4</b>	1 (train)	2 (train)	3 (train)	4 (test)	5 (train)
<b>5</b>	1 (train)	2 (train)	3 (train)	4(train)	5 (test)

Terlihat dari tabel 2.2 nilai  $k$  pada proses ini menggunakan nilai 5 dan data dijalankan sebanyak 5 kali untuk setiap *subset* data yang dapat digunakan sebagai

data latih dan data uji. Untuk pengujian dimana data dipecah dan perulangan dilakukan 5 kali dengan posisi data uji yang berbeda pada setiap iterasinya. Iterasi pertama data uji berada di posisi awal, iterasi kedua posisi data uji berada di posisi kedua, demikian untuk iterasi kelima. Dari penerapan nilai parameter  $k$  sebagai juga akan menghasilkan akurasi.

## 2.7 *Confusion matrix*

Tingkat kesesuaian antara nilai prediksi dan nilai aktual, atau dengan kata lain, seberapa baik model memprediksi kelas data, disebut akurasi. Presisi, juga disebut sensitivitas, mengukur tingkat ketepatan atau ketelitian dalam pengklasifikasian kelas, yaitu seberapa banyak dari data yang diprediksi sebagai positif yang benar-benar positif. *Recall*, juga disebut sensitivitas, mengukur proporsi positif aktual yang benar diidentifikasi oleh modifikasi.

Untuk mengukur nilai dari akurasi, *Recall*, dan presisi. Biasanya menggunakan *Confusion matrix*. *Confusion Matrix* adalah alat ukur berbentuk matriks yang digunakan untuk menggambarkan jumlah ketepatan klasifikasi terhadap kelas dengan algoritma yang digunakan. Alat ini biasanya digunakan untuk mengukur akurasi, presisi, dan *Recall*. Setiap baris dalam matrix confusion menunjukkan kelas sebenarnya, sedangkan setiap kolom menunjukkan kelas yang diprediksi oleh model. Dan dapat menghitung akurasi, presisi, dan *Recall* dari model klasifikasi yang digunakan dengan memeriksa elemen *Confusion Matrix* seperti *True positive*, *True negative*, *False negative*, dan *False positive* (Qadrini, 2021).

Tabel 2. 3 Contoh Confusion Matrix

Nilai Prediksi	Nilai Aktual	
	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	<i>TP</i>	<i>FN</i>
<i>Negative</i>	<i>FP</i>	<i>TN</i>

Keterangan :

TP (*True Positive*) = data diprediksikan positif, data aktual positif.

TN (*True Negative*) = data diprediksikan negatif, data aktual negatif.

FP (*False Positive*) = data diprediksikan positif, data aktual negatif.

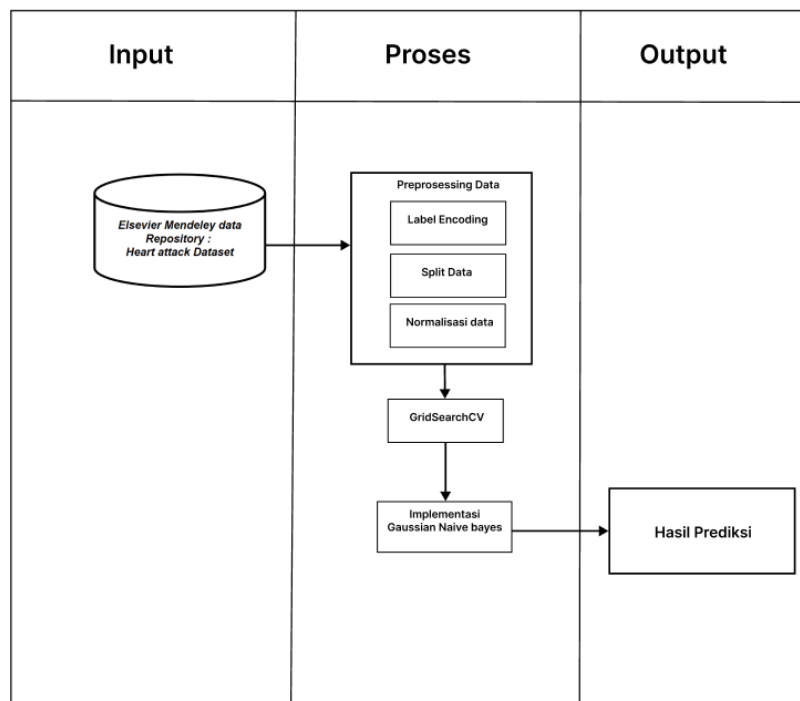
FN (*False Negative*) = data diprediksikan negatif, data aktual positif.

## BAB III

### DESAIN DAN IMPLEMENTASI

#### 3.1 Desain Sistem

Desain sistem dimulai dengan tahapan pengumpulan *dataset* dan *preprocessing* dan mengimplementasikan analisis data dan pengembangan model klasifikasi sebagai meprediksi serangan jantung. Dalam proses untuk membangun model prediksi serangan jantung diperlukan sebuah data, *preprocessing* dan implementasi *Gaussian Naïve Bayes* dan evaluasi.



Gambar 3. 1 Desain Sistem

Penelitian ini menggunakan algoritma *Gaussian Naïve Bayes* yang memiliki dua tahapan yaitu pelatihan dan pengujian. Dalam algoritma *Gaussian naive bayes*, tahap pelatihan melibatkan menghitung rata-rata dan variansi setiap fitur untuk setiap kelas yang ada dalam *dataset* pelatihan. Kemudian, algoritma ini

mengestimasi parameter distribusi *Gaussian* untuk setiap kelas dengan menggunakan data pelatihan.

Tahap pengujian dimulai setelah tahap pelatihan selesai. Tujuan dari tahap pengujian ini adalah untuk mengevaluasi kemampuan model *Gaussian naive bayes* dalam mengklasifikasikan data baru. Pada titik ini, algoritma menggunakan model yang telah dipelajari untuk memprediksi kelas dari data uji yang tidak memiliki label. Berdasarkan distribusi *Gaussian* yang telah ditetapkan selama tahap pelatihan, algoritma ini menghitung kemungkinan bahwa data uji tersebut termasuk dalam setiap kelas. Kelas dengan kemungkinan tertinggi kemudian dianggap sebagai prediksi kelas untuk data uji tersebut.

### **3.2 Pengumpulan Data**

Dalam tahapan ini akan dilakukan sebuah pengumpulan data menggunakan data skunder. Data yang akan digunakan dalam penelitian ini yaitu *dataset “Heart attack dataset”* yang bersumber dari *Elsevier Mendeley Data Repository*, Data skunder yang dipakai mangacu pada informasi yang dikumpulkan tidak langsung oleh peneliti, lebih tepatnya dikumpulkan oleh peneliti lain. Dari data skunder yang sudah dikumpulkan ini kemudian akan dimanfaatkan sebagai data penelitian untuk menguji model metode *Gaussian Naïve Bayes* dalam memperoleh hasil prediksi.

Dalam *dataset* ini berisikan data rekam medis dari pasien penderita serangan jantung yang terdiri dari 1319 data pasien serta memiliki 9 fitur dan hanya 8 fitur yang akan digunakan. Untuk penjelasan mengenai *dataset* terdapat pada table berikut :

Tabel 3. 1 Penjelasan Dataset

No	Fitur	Keterangan
1	<i>Age</i>	Umur
2	<i>Gender</i>	Jenis Kelamin (Pria "1" Wanita "0")
3	<i>Heart rate</i>	Detak jantung
4	<i>Systolic BP</i>	Tekanan darah sistolik adalah tekanan darah saat jantung berkontraksi dan memompa darah ke dalam jenis pembuluh darah arteri biasanya ditunjukkan dengan angka pertama dalam pembacaan tekanan darah, seperti "120/80 mmHg", di mana 120 adalah tekanan darah sistolik.
5	<i>Diastolic BP</i>	Tekanan darah diastolik adalah tekanan darah saat jantung beristirahat di antara kontraksi. Misalnya, dalam pembacaan "120/80 mmHg", angka 80 adalah tekanan darah diastolik.
6	<i>Blood Sugar</i>	Gula darah adalah kadar glukosa dalam darah, yang merupakan sumber energi utama untuk tubuh.
7	<i>CK-MB</i>	Kadar <i>CK-MB</i> dalam darah dapat digunakan sebagai petunjuk bahwa ada kerusakan pada jantung, seperti serangan jantung atau penyakit jantung lainnya. <i>CK-MB</i> adalah enzim kreatin kinase (CK) yang unik untuk otot jantung.
8	<i>Troponin</i>	<i>Troponin</i> adalah protein dalam otot jantung. Ketika terjadi kerusakan jantung, <i>Troponin</i> dilepaskan ke dalam darah. Pengukuran kadar <i>Troponin</i> digunakan untuk mendiagnosis serangan jantung dan menilai kerusakan jaringan otot jantung.
9	<i>Result</i>	Hasil yang akan ditunjukkan dapat berupa positif (serangan jantung) dan <i>negative</i> (serangan jantung)

Tabel 3. 2 Contoh Dataset

<i>Age</i>	<i>Gender</i>	<i>Heart rate</i>	<i>Systolic blood pressure</i>	<i>Diastolic blood pressure</i>	<i>Blood Sugar</i>	<i>CK-MB</i>	<i>Troponin</i>	<i>Result</i>
64	1	66	160	83	160.0	1.80	0.012	<i>negative</i>
21	1	94	98	46	296.0	6.75	1.060	<i>positive</i>
55	1	64	160	77	270.0	1.99	0.003	<i>Negative</i>
64	1	70	120	55	270.0	13.87	0.122	<i>Positive</i>
55	1	64	112	65	300.0	1.08	0.003	<i>Negative</i>
58	0	61	112	58	87.0	1.83	0.004	<i>Negative</i>

32	0	40	179	68	102.0	0.71	0.003	<i>negative</i>
63	1	60	214	82	87.0	300.0	2.370	<i>positive</i>
44	0	60	154	81	135.0	2.35	0.004	<i>Negative</i>
67	1	61	160	95	100.0	2.84	0.011	<i>negative</i>

### 3.3 *Preprocessing Data*

Untuk menyediakan data untuk penelitian, tahap prapemrosesan dilakukan. Prapemrosesan data dalam penelitian ini akan dilakukan dalam beberapa langkah. Pertama, data diubah pada atribut "*Result*" dengan menggunakan *LabelEncoder* untuk mengubah tipe datanya. Selanjutnya, Pengubahan kolom kategorikal menjadi tipe data kategori. langkah berikutnya Data *split* atau pembagian data dengan melakukan pemisahan fitur (x) dan target (y), kemudian dilakukan pembagian data latih dan data uji. Lalu, *StandardScaler* digunakan untuk melakukan penskalaan fitur (*feature scaling*). Penskalaan fitur adalah proses normalisasi atau standarisasi nilai-nilai dari atribut dalam *dataset*.

#### 3.3.1 Transformasi Data

Mengubah variabel kategorikal menggunakan *label encoder* sangat penting untuk langkah transformasi data karena *dataset* yang menggunakan *label encoder* secara otomatis memberikan nilai unik dalam kolom *Result* dan mengubah variabel kategorikal menjadi format numerik. Pada awalnya, nilai "Positif" dan "Negatif" dimasukkan ke dalam kolom atau atribut "*Result*" dalam bentuk string atau objek. Kemudian nilai-nilai ini diubah menjadi integer, dengan nilai "Negatif" adalah 0 dan "Positif" adalah 1. Teknik *LabelEncoder* mengubah data kategorikal menjadi

data numerik, membuatnya lebih mudah dipahami dan diakses oleh model pembelajaran mesin. Proses implementasi tahap transformasi data ditunjukkan pada gambar 3.2.

```
label_encoder = LabelEncoder()
label_encoder.fit(df["Result"])
class_mapping = dict(zip(label_encoder.classes_,
                        label_encoder.transform(label_encoder.classes_)))
print(class_mapping)
df["Result"] = label_encoder.transform(df["Result"])
```

Gambar 3. 2 Implementasi Tahap Transformasi Data

Disini *LabelEncoder* digunakan untuk mengubah label kategorial, kemudian *LabelEncoder* dilatih (*fit*) pada data, kita bisa menggunakannya untuk mengubah nilai-nilai dalam kolom "Result" menjadi angka. Pada baris 'dictionary' digunakan untuk memetakan kelas asli (nilai unik dalam kolom "Result") ke nilai numerik yang sesuai. Kemudian kolom 'label\_encoder.classes\_' adalah array yang berisi semua kelas unik yang dipelajari oleh *LabelEncoder*.

Kemudian 'label\_encoder.transform(label\_encoder.classes\_)' digunakan untuk mengubah semua kelas tersebut menjadi angka sesuai dengan urutan yang telah dipelajari. Terakhir pada baris *df["Result"] = ...* Baris ini berfungsi untuk mengubah semua nilai dalam kolom "Result" dari *DataFrame* menjadi angka menggunakan *LabelEncoder* yang telah dilatih sebelumnya. Hasilnya adalah kolom "Result" yang tadinya berisi nilai kategorikal sekarang berisi nilai numerik 'negative': 0 dan 'positive': 1. Berikut contoh *dataset* yang belum menjalani proses tranformasi data dengan menggunakan teknik *LabelEncoder* :



Tabel 3. 3 Contoh dataset yang belum menjalani proses transformasi data

<i>Systolic BP</i>	<i>Diastolic BP</i>	<i>Blood Sugar</i>	<i>CK-MB</i>	<i>Troponin</i>	<i>Result</i>
160	83	160.0	1.80	0.012	<i>negative</i>
214	82	87.0	300.0	2.370	<i>positive</i>
160	77	270.0	1.99	0.003	<i>Negative</i>

Dalam tabel 3.3 diatas terlihat dalam kolom atribut “*Result*” masih dalam bentuk string atau objek. Hal ini yang kemudian yang nantinya akan diubah menjadi numerik dengan menggunakan teknik *LabelEncoder* . Berikut adalah contoh *dataset* yang telah menjalani proses transformasi data menggunakan *LabelEncoder*:

Tabel 3. 4 Contoh dataset yang telah menjalani proses transformasi data

<i>Systolic BP</i>	<i>Diastolic BP</i>	<i>Blood Sugar</i>	<i>CK-MB</i>	<i>Troponin</i>	<i>Result</i>
160	83	160.0	1.80	0.012	0
214	82	87.0	300.0	2.370	1
160	77	270.0	1.99	0.003	0

Dalam tabel 3.4 diatas terlihat dalam kolom atribut “*Result*” yang sudah menjalankan tranformasi data dan merubah kolom “*Result*” menjadi numerik dengan menggunakan teknik *LabelEncoder* .

### 3.3.2 Split Data

*Split Data* adalah proses membagi *dataset* menjadi dua *subset* yang terpisah, yang umumnya disebut sebagai data (*train data*) latih dan (*test data*) data uji (Tan, n.d.). Pada langkah ini dilakukan langkah penting dengan memisahkan data menjadi fitur (X) dan target (y). Dalam konteks ini, fitur (X) adalah variabel input yang

digunakan untuk memprediksi nilai target, sementara target (y) adalah variabel output yang ingin diprediksi oleh model. Dengan memisahkan kolom "Result" sebagai target (y) yang ingin diprediksi. Proses implementasi tahap *Splitdata* ditunjukkan pada gambar 3.3.

```
# Menentukan fitur (x) dan target (y)
x = df.drop('Result', axis=1) # Fitur
y = df['Result'] # Target

# Melakukan split data
# Model A (90:10)
x_train_A, x_test_A, y_train_A, y_test_A = train_test_split(x, y, test_size=0.1, random_state=57)
print("Model A:")
print("Jumlah data train:", len(x_train_A))
print("Jumlah data test:", len(x_test_A))
```

Gambar 3. 3 Proses Implementasi Tahap Splitdata

Setelah pemisahan fitur (X) dan target (y), data dibagi menjadi dua set terpisah: data latih dan data uji menggunakan *train-test Split*. Proses ini penting karena memungkinkan kita untuk menguji kinerja model pada data yang tidak pernah dilihat sebelumnya. Proporsi data uji dapat ditentukan dengan *test\_size*, yang biasanya merupakan persentase dari keseluruhan *dataset*. Hal ini memungkinkan kita untuk mengalokasikan sebagian data untuk pengujian model, sementara sisa data digunakan untuk melatih model. Ini membantu dalam menyusun data dalam format yang sesuai untuk proses pembelajaran mesin, karena model memerlukan input dalam bentuk fitur dan output yang ingin diprediksi.

### 3.3.3 Normalisasi Data

Dalam analisis data dan pembelajaran mesin, fitur scaling merupakan langkah penting dalam praproses data. Tujuan utamanya adalah untuk menstandarisasi atau menormalkan fitur numerik *dataset* sehingga mereka memiliki skala yang

sebanding. Proses ini membantu memastikan bahwa perbedaan skala antara fitur tidak menyebabkan bias dalam model *machine learning*.

Dalam hal ini, *StandardScaler* adalah metode yang paling umum digunakan untuk scaling. Proses ini mengubah setiap fitur sehingga memiliki deviasi standar satu dan rata-rata nol. Dengan cara menghitung nilai rata-rata (*mean*) dan *standard deviation*(*standard deviation*) dari setiap fitur pada data latih. lalu nilai masing-masing fitur kemudian akan diubah dengan mengurangi nilai rata-ratanya dan membaginya dengan deviasi standar. Dengan cara ini, semua fitur akan memiliki rentang yang sebanding, yang membuat proses algoritma pembelajaran mesin lebih mudah. Proses implementasi tahap normalisasi data ditunjukkan pada gambar 3.4.

```
# Melakukan standard scaling terhadap fitur-fitur numerik
scaler = StandardScaler()
x_train_A = scaler.fit_transform(x_train_A)
x_test_A = scaler.transform(x_test_A)
x_train_B = scaler.fit_transform(x_train_B)
x_test_B = scaler.transform(x_test_B)
x_train_C = scaler.fit_transform(x_train_C)
x_test_C = scaler.transform(x_test_C)
x_train_D = scaler.fit_transform(x_train_D)
x_test_D = scaler.transform(x_test_D)
```

Gambar 3. 4 Proses Implementasi Tahap Normalisasi Data

Pada baris *fit*: *scaler.fit(X\_train)* digunakan untuk menghitung *mean* dan *standard deviation* dari setiap fitur dalam *X\_train*. Ini memungkinkan *StandardScaler* untuk mengetahui bagaimana data harus diubah sehingga setiap fitur memiliki *mean* 0 dan *standard deviation* 1.

Selanjutnya dalam baris *transform*: *scaler.transform(X\_train)* kemudian menggunakan *mean* dan *standard deviation* yang telah dihitung untuk menstandarkan data *X\_train*. Setiap fitur dalam *X\_train* diubah menjadi  $(x - \text{mean}) / \text{std}$ , di mana  $x$  adalah nilai asli, *mean* adalah rata-rata fitur tersebut, dan *std* adalah

*standard deviation* fitur tersebut. Gabungan kedua langkah ini (*fit* dan *transform*) dilakukan dengan satu metode *fit\_transform* yang menghitung statistik yang diperlukan dari data pelatihan dan langsung menerapkannya untuk menstandarkan data tersebut. Berikut adalah rumus dari *StandardScaler* yang terdapat pada persamaan (3.1)

$$Z_i = \frac{X_i - \bar{X}}{\sigma} \quad (3.1)$$

Keterangan :

Simbol  $Z_i$  = *Z-score* (ukuran seberapa jauh sebuah nilai dalam *dataset* berbeda dari rata-rata dalam satuan deviasi standar)

$Z_i$  = *Z-score* dari nilai individual dalam *dataset*

$X_i$  = Nilai individual dalam *dataset*.

$\bar{X}$  = Rata-rata dari *dataset*.

$\sigma$  = Deviasi standar dari *dataset*.

Berikut ini adalah contoh *dataset* yang belum menjalani normalisasi dengan menggunakan *StandardScaler*. Terdapat 5 data pada setiap fiturnya.

Tabel 3. 5 Contoh *dataset* yang belum menjalani normalisasi

<i>Systolic BP</i>	<i>Diastolic BP</i>	<i>Blood Sugar</i>	<i>CK-MB</i>	<i>Troponin</i>
160	83	160.0	1.80	0.012
112	58	87.0	183.0	0.004
160	77	270.0	1.99	0.003
117	68	110.0	1.04	0.810

Langkah pertama dalam mengimplementasikan persamaan (3.1) yang tercantum pada Tabel 3.5 adalah menghitung rata-rata dan *standard deviation* dari masing-masing atribut. rata-rata bisa dihitung dengan rumus persamaan (3.2).

$$\mu = \frac{\text{Jumlah seluruh nilai}}{\text{Jumlah sampel data}} \quad (3.2)$$

Cara untuk mengubah nilai pada atribut *Systolic BP* adalah dengan menghitung rata-rata dari atribut tersebut. Dari data yang sudah ada, jumlah seluruh nilai rata-rata *Systolic BP* adalah 167.738 dibagi jumlah sampel data 1319, hasilnya sekitar 127.171. Kemudian menghitung rata-rata (*mean*) dan *Standard deviation* dapat dihitung dengan menggunakan rumus persamaan (3.3).

$$\sigma = \sqrt{\frac{\sum(x - \mu)^2}{N}} \quad (3.3)$$

Keterangan :

- $X$  = nilai fitur asli
- $\mu$  = nilai rata-rata dari semua data
- $N$  = jumlah data
- $\sigma$  = Deviasi standar dari *dataset*.

Hasilnya dari  $\sigma$  yang didapatkan adalah :  $\sqrt{(160 - 127.171)^2 + (98 - 127.171)^2 + (160 - 127.171)^2 + (120 - 127.171)^2 + (112 - 127.171)^2 + \dots} / (1319) = 26.1227$ .

Nilai *mean* dan *standard deviation* dari masing-masing atribut kemudian ditambahkan ke persamaan (3.1) dan didapatkan  $\sigma = (160-127.171)/26.1227 = 1.25672307993$  atau 1.26. Dalam Tabel 3.6, Berikut ini adalah contoh *dataset* yang sudah menjalani normalisasi dengan menggunakan *StandardScaler*.

Tabel 3. 6 Contoh *dataset* yang sudah menjalani proses normalisasi

<i>Systolic BP</i>	<i>Diastolic BP</i>	<i>Blood Sugar</i>	<i>CK-MB</i>	<i>Troponin</i>
1.26	0.76	0.11	-0.59	-0.40
-12.07	-18.65	-516.43	-1041.81	185.47
1.26	0.34	1.65	-0.29	-0.31

-0.38	-0.31	-0.47	-0.41	-0.97
-------	-------	-------	-------	-------

Nilai *Z-score* bisa positif atau negatif tergantung apakah nilai dari fitur lebih tinggi atau lebih rendah dari rata-rata populasi. Nilai *Z-score* negatif berarti nilai sifat lebih rendah dari rata-rata populasi, sedangkan nilai *Z-score* positif menunjukkan nilai sifat lebih tinggi dari rata-rata populasi. Nilai negatif bukan berarti fitur tersebut salah atau tidak berguna, hanya menunjukkan posisi relatif fitur tersebut dalam distribusi populasi. Nilai *Z-score* juga dapat digunakan untuk mengukur seberapa jauh nilai fitur dari *mean* (rata-rata) populasi dalam satuan standar deviasi. Semakin besar nilai *Z-score* maka semakin jauh nilai fitur dari *mean* populasi.

### 3.4 *GridSearchCV*

*GridSearchCV* atau *Grid Search Cross-Validation* adalah metode yang digunakan dalam *machine learning* untuk mengoptimalkan *hyperparameter* model dengan cara yang sistematis dan efisien. *Hyperparameter* adalah parameter yang tidak dapat dipelajari dari data secara langsung dan perlu ditentukan sebelum pelatihan model, seperti tingkat *learning rate* dalam algoritma pembelajaran. *GridSearchCV* melakukan pencarian secara menyeluruh di seluruh ruang *hyperparameter* yang telah ditentukan oleh pengguna dalam bentuk *grid*, yang terdiri dari berbagai kombinasi nilai *hyperparameter* yang mungkin. Untuk setiap kombinasi *hyperparameter*, model dilatih dan divalidasi menggunakan *cross-validation* pada data pelatihan, yang melibatkan pembagian data menjadi beberapa

*subset (folds)* untuk memastikan bahwa model tidak *overfitting* pada *subset* tertentu dari data.

```
# Menggunakan GridSearchCV untuk mencari parameter terbaik untuk Model A
param_grid_A = {'var_smoothing': np.logspace(0,-9, num=100)}
nb_model_A = GaussianNB()
grid_search_A = GridSearchCV(nb_model_A, param_grid_A, cv=5)
grid_search_A.fit(x_train_A, y_train_A)

# Menampilkan parameter terbaik Model A
print("Parameter terbaik Model A:", grid_search_A.best_params_)

# Menampilkan skor validasi silang terbaik Model A
print("Skor validasi silang terbaik Model A:", grid_search_A.best_score_)

# Menampilkan model terbaik Model A
best_model_A = grid_search_A.best_estimator_
print("Model terbaik Model A:")
print(best_model_A)
```

Gambar 3. 5 Proses Implementasi Tahap GridSearchCV

Pada kode program diatas *GridSearchCV* bekerja dengan terlebih dahulu mendefinisikan rentang nilai *hyperparameter* yang ingin diuji, seperti *param\_grid\_A = {'var\_smoothing': np.logspace(0, -9, num=100)}*, yang mencakup 100 nilai dari 1 hingga  $10^{-9}$ . Kemudian model *Gaussian Naive Bayes* diinisialisasi dengan *nb\_model\_A = GaussianNB()*, kemudian *GridSearchCV* diatur menggunakan model tersebut dengan *5-fold cross-validation* melalui *grid\_search\_A = GridSearchCV(nb\_model\_A, param\_grid\_A, cv=5)*. Proses pencarian dimulai dengan membagi data latih menjadi 5 *subset*, selanjutnya melatih model pada 4 subset dan menguji pada subset ke-5, diulangi 5 kali untuk setiap kombinasi *hyperparameter*. Kemudian rata-rata skor validasi untuk setiap kombinasi dihitung, dan kombinasi dengan skor tertinggi dipilih sebagai yang terbaik. Hasilnya ditampilkan dengan menggunakan *\_search\_A.best\_params\_* untuk parameter terbaik, *grid\_search\_A.best\_score\_* untuk menampilkan skor validasi tertinggi, dan *grid\_search\_A.best\_estimator\_* untuk menampilkan model

terbaik. Proses ini memastikan bahwa model yang dihasilkan adalah yang paling optimal berdasarkan evaluasi menyeluruh.

### 3.5 Implementasi *Gaussian Naïve Bayes*

Klasifikasi *Naïve bayes* adalah pendekatan klasifikasi yang didasarkan pada probabilitas, dengan asumsi bahwa setiap atribut atau fitur objek memiliki sifat independen satu sama lain. Karena cukup sederhana dan seringkali menunjukkan kinerjanya yang cukup baik, metode ini masih menjadi subjek yang menarik dalam pengembangan sistem klasifikasi. Contoh hitungan *Naïve Bayes* secara umum dapat dilihat dalam persamaan (3.4).

$$P(C_j|X) = \frac{P(X|C_j) \cdot P(C_j)}{P(X)} \quad (3.4)$$

Keterangan

$P(C_j|X)$  = Probabilitas posterior dari kelas  $C_j$  setelah mengamati fitur  $X$ .

$P(X|C_j)$  = Probabilitas likelihood dari fitur  $X$  dalam kelas .

$P(C_j)$  = Probabilitas prior dari kelas  $C_j$ .

$P(X)$  = Probabilitas dari fitur  $X$ , yang sering dianggap sebagai faktor skalar dan tidak dihitung secara eksplisit dalam banyak kasus klasifikasi.

Metode *Naïve Bayes* biasanya menganggap bahwa fungsi kepadatan peluang untuk setiap atribut berdistribusi secara normal (*Gaussian*). Meskipun sederhana, asumsi ini dapat berdampak signifikan pada pembentukan model klasifikasi *Naïve bayes*. Adapun langkah pertama, menggunakan *Gaussian Naïve Bayes* adalah dengan menghitung probabilitas prior untuk setiap kelas. Probabilitas prior adalah probabilitas dari setiap kelas sebelum kita melihat data. Dalam konteks klasifikasi,



ini mengacu pada seberapa sering setiap kelas muncul dalam *dataset* pelatihan. Rumus untuk menghitung probabilitas prior dapat dilihat dalam persamaan (3.5).

$$P(C_j) = \frac{\text{Jumlah data dari kelas } C_j}{\text{Total data}} \quad (3.5)$$

Kita menghitung jumlah data yang memiliki kelas tertentu, kemudian dibagi dengan total jumlah data dalam *dataset* untuk mendapatkan probabilitas prior dari setiap kelas. Selanjutnya mengestimasi parameter distribusi *Gaussian* untuk setiap fitur dalam setiap kelas. Ini melibatkan perhitungan rata-rata ( $\hat{\mu}_{jk}$ ) dan deviasi standar ( $\hat{\sigma}_{jk}$ ) dari setiap fitur dalam setiap kelas. Misalnya, jika kita memiliki dua kelas (positif dan negatif) dan lima fitur (*Systolic BP*, *Diastolic BP*, *Blood Sugar*, *CK-MB* dan *Troponin*), maka kita perlu menghitung nilai rata-rata dan deviasi standar untuk setiap fitur dalam kedua kelas.

Langkah selanjutnya adalah menghitung probabilitas likelihood dari seluruh fitur ( $X$ ). Misalnya, Ketika fitur-fitur objek,  $X = \{x_1 = \text{Systolic BP}, x_2 = \text{Diastolic BP}, x_3 = \text{Blood Sugar}, x_4 = \text{CK-MB dan } x_5 = \text{Troponin}\}$  diketahui, dan setiap fitur diasumsikan berdistribusi normal (*Gaussian*), maka peluang fitur atau atribut dengan memperhatikan kelas ke- $j$  ( $C_j$ ). Rumus untuk menghitung probabilitas likelihood dapat dilihat pada persamaan (3.6).

$$P(X|C_j) = \prod_{k=1}^d P(X_k|C_j) = \prod_{k=1}^d N(x_k; \hat{\mu}_{jk}, \hat{\sigma}_{jk}) \quad (3.6)$$

Keterangan :

$P(X|C_j)$  = Probabilitas likelihood keseluruhan dari semua fitur  $X$  dalam kelas  $C_j$ .

$d$  = Jumlah total fitur

$X_k$  = Nilai fitur ke- $k$  dalam data  $X$

$N(x_k; \hat{\mu}_{jk}, \hat{\sigma}_{jk})$  = probabilitas likelihood dari fitur ke- $k$  dalam kelas  $C_j$  yang dihitung menggunakan rumus distribusi *Gaussian* (normal) seperti contoh pada persamaan (3.7).

$P(X/C_j)$  mengindikasikan probabilitas dari seluruh vektor fitur  $X$  (*Systolic BP*, *Diastolic BP*, *Blood Sugar*, *CK-MB* dan *Troponin*) dalam kelas  $C_j$ . Ini dihitung dengan mengalikan probabilitas dari setiap fitur individual  $X_k$  dalam kelas  $C_j$ , di mana  $k$  adalah indeks fitur dan  $d$  adalah jumlah total fitur.

$P(X_k/C_j)$  adalah probabilitas dari fitur individual  $X_k$  dalam kelas  $C_j$ . Ini dihitung menggunakan distribusi *Gaussian*, yang dinyatakan sebagai  $N(x_k; \mu^{jk}, \sigma^{jk})$ . Artinya, probabilitas ini adalah probabilitas bahwa nilai fitur  $X_k$  berasal dari distribusi normal dengan rata-rata  $\mu^{jk}$  dan deviasi standar  $\sigma^{jk}$  dalam kelas  $C_j$ . Rumus ini berguna untuk menggambarkan bagaimana kita menghitung probabilitas dari seluruh vektor fitur  $\bar{X}$  dalam kelas tertentu  $\bar{C}_j$ , dengan mempertimbangkan probabilitas distribusi normal dari setiap fitur individual  $\bar{X}_k$  dalam kelas tersebut.

Perhitungan probabilitas likelihood menggunakan distribusi *Gaussian* dilakukan dengan menggunakan rumus untuk distribusi *Gaussian* dapat dilihat dalam contoh persamaan (3.7) dan (3.8).

$$\mathcal{N}(x_k; \hat{\mu}_{jk}, \hat{\sigma}_{jk}) = \frac{1}{\sqrt{2\pi\hat{\sigma}_{jk}^2}} \exp\left(-\frac{(x_k - \hat{\mu}_{jk})^2}{2\hat{\sigma}_{jk}^2}\right) \quad (3.7)$$

Keterangan :

$x_k$  = adalah nilai fitur ke- $k$  yang diamati.

$\mu^{jk}$  = adalah rata-rata fitur ke- $k$  dalam kelas  $C_j$ .

$\sigma^{jk}$  = adalah deviasi standar fitur ke- $k$  dalam kelas  $C_j$ .

Rumus ini menggambarkan distribusi *Gaussian* atau distribusi normal dari nilai fitur  $X_k$  dalam kelas, di mana probabilitas kemunculan nilai  $x_k$  dihitung berdasarkan jaraknya dari rata-rata  $\mu^j_k$  dalam satuan deviasi standar  $\sigma^j_k$ . Semakin dekat nilai  $x_k$  dengan rata-rata  $\mu^j_k$ , semakin tinggi probabilitasnya, dan sebaliknya. Contoh persamaan untuk menghitung probabilitas likelihood ( $1.26|C1$ ) dari nilai *Systolic BP* yang diberikan (1.26) dalam kelas positif  $C1$ .

$$P(1.26|C1) = \frac{1}{\sqrt{2\pi} \times 26.1227} \times e^{-\frac{(1.26-127.171)^2}{2 \times (26.1227)^2}}$$

$$P(1.26|C1) \approx 0.036 \quad (3.8)$$

Keterangan :

- $C1$  adalah probabilitas prior dari kelas positif.
- $P(1.26|C1)$  adalah probabilitas likelihood dari nilai fitur *Systolic BP* sebesar 1.26 dalam kelas positif ( $C1$ ).
- Pertama, kita hitung bagian dalam ekspresi eksponensial:  $e^{-\frac{(1.26-127.171)^2}{2 \times (26.1227)^2}}$
- Kemudian, kita hitung nilai eksponensialnya menggunakan fungsi eksponensial  $e$ .
- Selanjutnya, kita hitung bagian pembagi ekspresi tersebut:  $2\pi \times 26.1227$
- Terakhir, kita bagi hasil dari langkah 2 dengan hasil dari langkah 3 untuk mendapatkan nilai  $P(1.26|C1)$  yang menunjukkan seberapa mungkin kita akan melihat nilai 1.26 dalam fitur *Systolic BP* jika data berasal dari kelas  $C1$ .
- Dalam contoh ini, nilai yang diperoleh adalah “sekitar” 0.036, yang menunjukkan probabilitas likelihood dari nilai fitur 1.26 dalam kelas  $C1$ .

Lakukan perhitungan yang sama pada variable selanjutnya berikut adalah hasil *Diastolic BP*, *Blood Sugar*, *CK-MB*, *Troponin* Untuk kelas  $C1$  (positif) :

- $P(1.26|C1) \approx 0.036$  (seperti yang telah kita hitung sebelumnya)
- $P(0.76|C1) \approx 4.2432 \times 10^{-7}$ .
- $P(0.11|C1) \approx 0.0404$
- $P(-0.51|C1) \approx 0.0167$
- $P(-0.41|C1) \approx 0.0489$

Selanjutnya, lakukan perhitungan probabilitas likelihood menggunakan distribusi *Gaussian* pada setiap variabel sama seperti sebelumnya pada kelas *negative* :

$$P(-12.06|C1) = \frac{1}{\sqrt{2\pi} \times 26.1227} \times e^{-\frac{(-12.06-127.171)^2}{2 \times (26.1227)^2}}$$

Langkah-langkahnya adalah sebagai berikut:

- Hitung nilai dari pembilang:

$$\frac{1}{2\pi \times 26.1227} = \frac{1}{52.2454} \approx 0.0191$$

- Hitung nilai dari penyebut eksponensial:

$$2 \times (26.1227)^2 = 3417.5711$$

- Hitung nilai eksponensial:

$$(-12.06-127.171)^2 = (-139.231)^2 = 19365.5626$$

- Bagi hasil dari langkah 3 dengan hasil dari langkah 2:

$$\frac{19365.5626}{3417.5711} \approx 5.6636$$

- Hitung nilai eksponensial dari hasil pembagian:

$$e^{-5.6636} \approx 0.0034$$

- Kalikan hasil dari langkah 1 dengan hasil dari langkah 5:

$$0.0191 \times 0.0034 \approx 0.0001$$

Jadi, nilai dari  $P(-12.06|C0)$  adalah sekitar 0.0001. kemudian berikut adalah hasil *Diastolic BP, Blood Sugar, CK-MB, Troponin* Untuk kelas C0 (negatif).

- $P(-12.06|C0) \approx 0,0001$
- $P(-18.654|C0) \approx 3.9924 \times 10^{-12}$ .
- $P(-516.428|C0) \approx 5.1954 \times 10^{-13}$ .
- $P(-1041.811|C0) \approx 0,0095$
- $P(185.467|C0) \approx 2.083 \times 10^{-16}$ .

Langkah selanjutnya menghitung menggunakan probabilitas posterior. Probabilitas posterior adalah probabilitas dari suatu kelas setelah mengamati data atau fitur tertentu. Dalam konteks *Gaussian Naïve Bayes*, probabilitas posterior  $P(C_j|X)$  adalah probabilitas bahwa suatu sampel data termasuk dalam kelas tertentu ( $C_j$ ) setelah mengamati nilai fitur ( $X$ ). Ini dihitung menggunakan teorema Bayes dan dikalikan dengan probabilitas prior  $P(C_j)$  dan probabilitas likelihood  $P(X|C_j)$ . Rumus untuk menghitung probabilitas posterior dapat dilihat pada persamaan (3.9).

$$P(C_j|X) = P(C_j)P(X|C_j) = P(C_j) \prod_{k=1}^d N(x_k; \hat{\mu}_{jk}, \hat{\sigma}_{jk}) \quad (3.9)$$

Keterangan :

- $P(C_j|X)$  = Probabilitas posterior dari kelas  $C_j$  setelah mengamati fitur  $X$ . Ini adalah probabilitas kelas yang kita prediksi setelah mengamati nilai fitur dari data yang diamati.
- $P(C_j)$  = Probabilitas prior dari kelas  $C_j$ , yaitu probabilitas kelas  $C_j$  sebelum kita mengamati data. Ini dapat dilihat sebagai probabilitas "asumsi" sebelum kita melihat nilai fitur dari data.
- $P(X|C_j)$  = Probabilitas likelihood dari fitur  $X$  dalam kelas  $C_j$ , yang menyatakan seberapa mungkin kita akan melihat nilai fitur  $X$  jika data berasal dari kelas  $C_j$ . Ini dihitung menggunakan distribusi *Gaussian* untuk setiap fitur dalam setiap kelas.
- $\prod_{k=1}^d N(x_k; \hat{\mu}_{jk}, \hat{\sigma}_{jk})$  = Hasil perkalian dari probabilitas likelihood untuk setiap fitur  $X_k$  dalam kelas  $C_j$ . Ini adalah probabilitas likelihood gabungan untuk semua fitur dalam kelas  $C_j$ , di mana  $\hat{\mu}_{jk}$  adalah rata-rata dan  $\hat{\sigma}_{jk}$  adalah deviasi standar fitur  $X_k$  dalam kelas  $C_j$ . Mari kita lanjutkan perhitungan probabilitas posterior untuk kelas positif :
  - $P(C1|X) = P(C1) \times P(1.26 | C1) \times P(0.76 | C1) \times P(0.18 | C1) \times P(-0.29 | C1) \times P(-0.30 | C1)$
  - $P(C1 | X) = 0.036 \times 4.2432 \times 0.0404 \times 0.0167 \times 0.0489$
  - $P(C1 | X) \approx 0.036 \times 4.2432 \times 0.0404 \times 0.0167 \times 0.0489 \times 0.0489$
  - $P(C1|X) \approx 1.3933 \times 10^{-8}$
- Berikut adalah hasil perhitungan probabilitas posterior untuk kelas *negative*:
  - $P(C0|X)=P(C0) \times P(-12.06|C0) \times P(-18.654|C0) \times P(-516.428|C0) \times P(-1041.811|C0) \times P(185.467|C0)$

- $P(C0|X) = 0.0001 \times 0.00000000000039924 \times 0.000000000000051954 \times 0.0095 \times 0.0000000000000000208$
- $P(C0|X) \approx 4.2432 \times 10^{-16} \times 0.0404 \times 0.0167 \times 0.0489 \times 0.0489$
- $P(C0|X) \approx 4.2432 \times 10^{-16} \times 0.00002707228$
- $P(C0|X) \approx 1.147042816 \times 10^{-21}$

Jadi, probabilitas posterior  $P(C0|X)$  adalah sekitar  $1.147042816 \times 10^{-21}$

Selanjutnya, kita membandingkan probabilitas posterior untuk kelas positif  $P(C1|X)$  dengan probabilitas posterior untuk kelas *negative*  $P(C0|X)$ . kemudian Kita akan memilih kelas yang memiliki probabilitas posterior tertinggi sebagai prediksi kita.

- $P(C1|X) \approx 1.3933 \times 10^{-8}$
- $P(C0|X) \approx 1.147042816 \times 10^{-21}$

Kita dapat melihat bahwa  $P(C1|X)$  lebih besar dari  $P(C0|X)$ . Oleh karena itu, kesimpulan menggunakan rumus argmax adalah bahwa prediksi kelas untuk sampel ini adalah kelas positif C1, karena probabilitas posterior untuk kelas positif lebih tinggi daripada probabilitas posterior untuk kelas negatif.

Implementasi *Gaussian naive bayes* bersama *GridSearchCV* dilakukan untuk hyperparameter tuning. Kombinasi ini memungkinkan kita untuk mencari nilai terbaik dari parameter `var_smoothing` dengan menggunakan cross-validation, memastikan model yang lebih robust dan siap untuk generalisasi pada data baru. *Gaussian Naïve Bayes* dalam program dapat dilihat pada gambar (3.5).

```
param_grid = {'var_smoothing': np.logspace(0,-9, num=100)}
nb_model = GaussianNB()
grid_search = GridSearchCV(nb_model, param_grid, cv=5)
grid_search.fit(x_train, y_train)
```

Gambar 3. 6 Proses Implementasi *Gaussian Naïve Bayes*

Jadi, Proses klasifikasi menggunakan metode *Gaussian Naïve Bayes* dimulai dengan memasukkan nilai pada setiap fitur yang digunakan. Selanjutnya, nilai  $(\hat{\mu}^jk)$  dan  $(\hat{\sigma}^jk)$  dihitung untuk setiap fitur dalam setiap kelas. Kemudian, dilakukan perhitungan *Gaussian* menggunakan rumus distribusi *Gaussian* untuk setiap fitur dalam setiap kelas. Perhitungan posterior dilakukan dengan mengalikan nilai prior (probabilitas kelas) dengan nilai hasil perhitungan *Gaussian*. Setelah menghitung probabilitas posterior, langkah selanjutnya adalah memilih kelas yang memiliki probabilitas posterior tertinggi untuk digunakan sebagai prediksi akhir. Kemudian Setelah melakukan prediksi untuk data tertentu kita dapat mengevaluasi kinerja model, Ini biasanya dilakukan dengan menggunakan metrik evaluasi seperti *accuracy*, *Precision*, *Recall*, dan *F1-score*.

### 3.6 Evaluasi Performa

Tahapan terakhir dari penelitian ini adalah dengan mengevaluasi lebih lanjut hasil prediksi yang sudah didapatkan dari pelatihan model. Terdapat beberapa tahapan Dalam proses evaluasi yang pertama menggunakan *Confusion matrix*. Tabel *Confusion Matrix* digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan nilai prediksi model dengan nilai *dataset* sebenarnya. Proses ini melibatkan perhitungan *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Dimana *True Positive* (TP) dan *True Negative* (TN) menggambarkan keakuratan prediksi yang sesuai dengan hasil yang sebenarnya. sedangkan *False Positive* (FP) dan *False Negative* (FN) terjadi Ketika hasil yang diprediksi tidak sesuai dengan kondisi sebenarnya.

Tabel 3. 7 Contoh *Confusion matrix*

Nilai Prediksi	Nilai Aktual	
	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	<i>TP</i>	<i>FN</i>
<i>Negative</i>	<i>FP</i>	<i>TN</i>

Selanjutnya, Untuk menghasilkan perhitungan akurasi, hasil matriks confusion dapat digunakan. Akurasi adalah gambaran hasil dari proses yang teliti, tepat, dan cermat yang menghasilkan nilai atau perhitungan perbandingan jumlah data yang diprediksi yang benar dengan jumlah data secara keseluruhan (Kamil et al., 2022). Nilai akurasi dapat dilakukan dengan rumus berikut (3.10).

$$Akurasi = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (3.10)$$

Selanjutnya presisi, presisi adalah salah satu metrik evaluasi yang digunakan untuk mengukur seberapa banyak dari semua prediksi positif yang dibuat oleh model yang benar-benar positif. Metrik ini bermanfaat dalam situasi di mana penting untuk mengurangi jumlah *False positive*. Nilai presisi dapat dihitung menggunakan rumus berikut (3.11).

$$Precision = \frac{TP}{(TP + FP)} \quad (3.11)$$

Keterangan :

- TP adalah jumlah *True Positive*, yaitu jumlah kasus positif yang benar-benar diprediksi dengan benar oleh model.
- FP adalah jumlah *False Positive*, yaitu jumlah kasus negatif yang salah diprediksi sebagai positif oleh model.

Selanjutnya *Recall*, metrik *Recall* adalah sebuah cara untuk mengetahui seberapa berhasil model menemukan informasi yang benar. Ini dapat dilakukan



dengan membandingkan jumlah prediksi yang benar positif dengan total data yang sebenarnya positif. Nilai *Recall* dapat dihitung dengan rumus berikut (3.12)

$$Recall = \frac{TP}{(TP + FN)} \quad (3.12)$$

Keterangan :

- TP adalah jumlah *True Positive*, yaitu jumlah kasus positif yang benar-benar diprediksi dengan benar oleh model.
- FN adalah jumlah *False Negative*, yaitu jumlah kasus positif yang salah diprediksi sebagai negatif oleh model.

Selanjutnya F1 score, yang merupakan metrik kombinasi yang menilai keseimbangan antara *Recall* dan presisi. Nilai dari F1 score bisa dihitung dengan menggunakan rumus berikut (3.13)

$$F1\ Score = 2x \frac{Presisi \times Recall}{Presisi + Recall} \quad (3.13)$$

### 3.7 Skenario Pengujian

Pertama, data dibagi menjadi dua komponen utama yaitu: fitur (X) dan target (y). Ini dilakukan untuk mempersiapkan data untuk analisis lebih lanjut. Selanjutnya, label di kolom "*Result*" dienkodkan menggunakan *LabelEncoder* untuk memastikan bahwa nilai target memiliki sifat kategorikal dalam format numerik agar dapat diproses oleh model. kemudian untuk melakukan uji coba, Data dibagi dengan menggunakan *train-test Split*.

Presentase dari data pelatihan dan data pengujian dalam setiap scenario pengujian dapat dilihat pada Tabel 3.8 berikut.

Tabel 3. 8 Pembagian data latih dan data uji

Skenario	Presentase data latih	Presentase data uji
A	90%	10%

B	80%	20%
C	70%	30%
D	65%	35%

Model pertama (Model A) menggunakan total 1319 data, 1187 data latih dan 132 data uji. sementara (Model B) menggunakan total 1319 data dengan pembagian 1055 data latih dan 264 data uji. (Model C) menggunakan total 1319 dengan pembagian 923 data latih dan 396 data uji, sedangkan (Model D) menggunakan total 1319 dengan pembagian 857 data latih dan 462 data uji. Pembagian data ini membantu dalam pelatihan model dengan menggunakan data latih (*train*) dan menguji performa model menggunakan data uji (*test*). Data uji (*test*) digunakan untuk menguji performa model dengan metrik evaluasi menggunakan *accuracy*, *Precision*, *Recall*, dan *F1-score*.

Kasus uji yang akan digunakan pada penelitian ini akan melibatkan metode *Cross-validation* atau validasi silang untuk mengevaluasi model secara lebih detail.

Berikut ilustrasi dari proses *5-fold cross validation* pada tabel 3.9

Tabel 3. 9 Contoh proses *5-fold cross validation*

<i>Fold</i>	<i>5-fold cross validation</i>				
<b>1</b>	1 (test)	2 (train)	3 (train)	4(train)	5 (train)
<b>2</b>	1 (train)	2 (test)	3 (train)	4(train)	5 (train)
<b>3</b>	1 (train)	2 (train)	3 (test)	4(train)	5 (train)
<b>4</b>	1 (train)	2 (train)	3 (train)	4 (test)	5 (train)
<b>5</b>	1 (train)	2 (train)	3 (train)	4(train)	5 (test)

*Cross-validation* yang akan digunakan yaitu *5 fold*, *10 fold*, *15 fold*, *20 fold* atau dengan menggunakan  $cv=5$ ,  $cv=10$ ,  $cv=15$ ,  $cv=20$ . Metrik evaluasi yang digunakan adalah akurasi. Hasil *cross-validation* ditampilkan sebagai array nilai akurasi pada setiap *fold*, dan rata-rata akurasi dari seluruh *fold* juga ditampilkan.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Langkah-Langkah Pengujian

Pada bagian ini menjabarkan tentang analisis hasil pengujian sistem berdasarkan scenario pengujian yang telah dijelaskan pada sub bab 3.6 untuk mengetahui performa metode *Gaussian Naïve Bayes* dalam memprediksi risiko serangan jantung. Adapun langkah-langkah yang dilakukan untuk pengujian diantaranya :

1. *Dataset* yang digunakan berasal dari *dataset* publik yang diambil melalui Elsevier Mendeley Data Repository: *Heart Attack Dataset* yang terdiri dari 1319 data pasien serta memiliki 9 fitur yaitu *Age*, *Gender*, *Heart rate*, *Systolic BP*, *Diastolic BP*, *Blood Sugar*, *CK-MB*, *Troponin*, dan *Result*.
2. Kemudian, kita menggunakan *LabelEncoder* untuk mengubah nilai kategorikal dalam kolom "*Result*" setelah pemanggilan *fit* dan *transform*, *class\_mapping* akan menjadi `{'negative': 0, 'positive': 1}`, yang menunjukkan bahwa nilai '*negative*' akan diubah menjadi 0 dan nilai '*positive*' akan diubah menjadi 1.
3. *Dataset* kemudian dipisahkan menjadi dua bagian, yaitu data latih dan data uji, sesuai dengan skenario pembagian *dataset* yang dijelaskan pada subbab 3.6. Pembagian *dataset* dilakukan menggunakan metode *train\_test\_Split()*.
4. Kemudian dilakukan normalisasi data menggunakan menggunakan *StandardScaler* dari library *scikit-learn*.

5. Implementasi *Gaussian naive bayes* diikuti dengan pencarian parameter terbaik menggunakan *GridSearchCV* untuk mencari parameter *var\_smoothing*. Kemudian model terbaik disimpan dalam variabel *best\_model* untuk evaluasi lebih lanjut terhadap data uji guna mengukur performanya dalam melakukan klasifikasi.
6. Setelah mencari parameter terbaik dan melakukan pembagian data, data *testing* digunakan untuk mengevaluasi performa model. Pada penelitian ini data dibagi dengan skenario perbandingan 90:10, 80:30, 70:20, dan 65:35 dengan 65% data *training* dan 35% data *testing*. Metode *K-fold cross validation* digunakan untuk mengevaluasi performa model dengan jumlah lipatan yang berbeda untuk mendapatkan estimasi yang lebih stabil terhadap performa model. Selanjutnya, sistem yang telah dibangun dievaluasi untuk mengetahui kinerja metode dengan menggunakan *Confusion matrix*, *Accuracy*, *Precision*, *Recall*, dan *F1-score*.

#### **4.2 Uji Coba *Split Data***

Pada subbab ini sistem prediksi serangan jantung diuji dengan menggunakan metode *Gaussian naive bayes*. Metode ini merupakan metode pembelajaran mesin yang dapat mengolah dan mengklasifikasikan data berdasarkan karakteristik yang telah ditentukan. Tingkat keberhasilan metode *Gaussian naive bayes* pada sistem ini dievaluasi berdasarkan perbandingan antara nilai kelas sebenarnya ( $y_{test}$ ) dengan nilai kelas yang diprediksi oleh model ( $y_{pred}$ ). Setelah memperoleh nilai  $y_{test}$  dan  $y_{pred}$  dari model yang dilatih dengan data pelatihan, dilakukan

pengujian menggunakan *cross validation* untuk mendapatkan estimasi performa model yang lebih stabil.

Pada penjelasan bab sebelumnya skenario uji coba dalam pengujian sistem dilakukan dua metode, yaitu dengan *Split Data* dan *K-fold cross validation*. *Split Data* metode yang digunakan dengan membagi data menjadi dua bagian, yaitu data *training* dan data *testing*. Data *training* digunakan untuk melatih model, sedangkan data *testing* digunakan untuk mengevaluasi performa model. Pada penelitian ini data dibagi dengan skenario perbandingan 90:10, 80:30, 70:20, dan 65:35 dengan 65% data *training* dan 35% data *testing*. Metode *K-fold cross validation* digunakan untuk mengevaluasi performa model dengan menggunakan data yang sama. Teknik dari *K-fold cross validation* membagi data menjadi 'k' bagian set data dengan ukuran yang sama. *Training* data dan *testing* data dilakukan sebanyak jumlah k yang telah ditentukan. Penelitian ini membagi jumlah nilai k dengan 5, 10, 15, dan 20 (Riza, rizqi robbi arisandi, 2022).

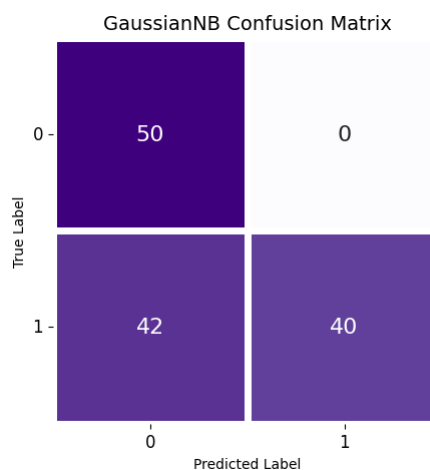
#### **4.2.1 Hasil Pengujian Model A**

Sebanyak 1319 data pasien telah disiapkan dan dibagi menjadi dua kategori, yaitu data pelatihan dan data pengujian, dengan rasio 90:10. Dengan demikian, terdapat 1187 data pelatihan dan 132 data pengujian. Dari hasil pengujian yang telah dilakukan didapatkan nilai *accuracy*, *Precision*, *Recall*, dan *F-1 score* yang ditunjukkan pada tabel 4.1.

Tabel 4. 1 Hasil pengujian model A

Kriteria	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-1 score</i>	Jumlah data pengujian
0		0.54	1.00	0.70	50
1		1.00	0.49	0.66	82
	0.68				132

Pada tabel 4.1 memberikan hasil evaluasi yang menyeluruh terhadap kinerja model klasifikasi. Dalam konteks ini, model telah diuji terhadap dua kelas yang berbeda, diidentifikasi sebagai kelas 0 dan kelas 1. Ini bisa merepresentasikan dua kategori yang berbeda dalam *dataset* yaitu *negative* (0) dan positif (1). Untuk kelas 0 model menunjukkan nilai *Precision* sebesar 0.54, *Recall* 1.00, dan *F-1 score* 0.70 dengan 50 jumlah data pengujian. Untuk kelas 1 model menunjukkan nilai *Precision* sebesar 1.00, *Recall* 0.49, dan *F-1 score* 0.66 dengan 82 jumlah data pengujian. Terakhir *accuracy* didapatkan sebesar 0.68 dengan 132 jumlah data pengujian.

Gambar 4. 1 Hasil *Confusion Matrix* Model A

Berikut adalah hasil dari nilai aktual dan prediksi pada pengujian model A.

Tabel 4. 2 Hasil nilai aktual dan prediksi model A

<i>ID</i>	<i>Actual</i>	<i>Predicted</i>
677	1	1
1046	0	0
610	0	0
49	0	0
1284	1	0
...	...	...
898	1	1
963	1	0
932	1	0
578	1	1
409	0	0

Jumlah yang dihasilkan nilai aktual dan yang sudah diprediksi, bisa dilihat dalam tabel berikut nilai yang salah pada sebanyak 42.

Tabel 4. 3 Jumlah nilai aktual dan prediksi model A

Class	Actual	Predicted
1	82	40
0	50	92
Jumlah nilai yang salah prediksi ada 42		

Hasil pengujian model *Gaussian Naïve Bayes* memprediksi 50 data yang berhasil diprediksi *True Negative* (TN) dengan benar, 40 data yang berhasil diprediksi *True Positive* (TP) dengan benar, 42 data yang salah diprediksi *negative* (FN), dan 0 data yang salah diprediksi *positive* (FP). Hasil perhitungan *accuracy*, *Precision*, *Recall*, dan *F-1 score* dari model A sebagai berikut :

$$Accuracy = \frac{40+50}{40+50+42+0} = 0.68$$

$$Precision = \frac{40}{40+0} = 1.00$$

$$Recall = \frac{40}{40+42} = 0.48$$

$$F-1 \text{ score} = \frac{2 \times 1.00 \times 0.48}{1.00 + 0.48} = 0.65$$



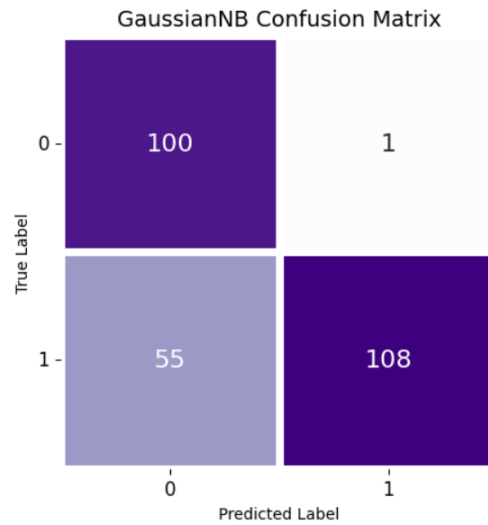
#### 4.2.2 Hasil Pengujian Model B

Sebanyak 1319 data pasien telah disiapkan dan dibagi menjadi dua kategori, yaitu data pelatihan dan data pengujian, dengan rasio 80:30. Dengan demikian, terdapat 1055 data pelatihan dan 264 data pengujian. Dari hasil pengujian yang telah dilakukan didapatkan nilai *accuracy*, *Precision*, *Recall*, dan *F-1 score* yang ditunjukkan pada tabel 4.2.

Tabel 4. 4 Hasil pengujian model B

Kriteria	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-1 score</i>	Jumlah data pengujian
0		0.65	0.99	0.78	101
1		0.99	0.66	0.79	163
	0.79				264

Pada tabel 4.2 memberikan hasil evaluasi yang menyeluruh terhadap kinerja model klasifikasi. Dalam konteks ini, model telah diuji terhadap dua kelas yang berbeda, diidentifikasi sebagai kelas 0 dan kelas 1. Ini bisa merepresentasikan dua kategori yang berbeda dalam *dataset* yaitu *negative* (0) dan *positif* (1). Untuk kelas 0 model menunjukkan nilai *Precision* sebesar 0.65, *Recall* 0.99, dan *F-1 score* 0.78 dengan 101 jumlah data pengujian. Untuk kelas 1 model menunjukkan nilai *Precision* sebesar 0.65, *Recall* 0.66, dan *F-1 score* 0.79 dengan 163 jumlah data pengujian. Terakhir *accuracy* didapatkan sebesar 0.79 dengan 264 jumlah data pengujian.

Gambar 4. 2 Hasil *Confusion Matrix* Model B

Berikut adalah hasil dari nilai aktual dan prediksi pada pengujian model B.

Tabel 4. 5 Hasil nilai aktual dan prediksi model B

<i>ID</i>	<i>Actual</i>	<i>Predicted</i>
677	1	1
1046	0	0
610	0	0
49	0	0
1284	1	0
...	...	...
1176	0	0
1002	0	0
1159	1	0
542	1	1
170	1	0

Jumlah yang dihasilkan nilai aktual dan yang sudah diprediksi bisa dilihat dalam tabel berikut.

Tabel 4. 6 Jumlah nilai aktual dan prediksi model B

Class	Actual	Predicted
1	163	109
0	101	155

Dari tabel diatas terlihat jumlah nilai yang salah pada sebanyak 56.

Hasil pengujian model *Gaussian Naïve Bayes* memprediksi 100 data yang berhasil diprediksi *True Negative* (TN) dengan benar, 108 data yang berhasil

diprediksi *True Positive* (TP) dengan benar, 55 data yang salah diprediksi *negative* (FN), dan 1 data yang salah diprediksi *positive* (FP). Hasil perhitungan *accuracy*, *Precision*, *Recall*, dan *F-1 score* dari model B sebagai berikut :

$$Accuracy = \frac{100+108}{100+108+55+1} = 0.79$$

$$Precision = \frac{100}{100+55} = 0.64$$

$$Recall = \frac{100}{100+1} = 0.99$$

$$F-1 \text{ score} = \frac{2 \times 0.64 \times 0.99}{0.64 + 0.99} = 0.78$$

#### 4.2.3 Hasil Pengujian Model C

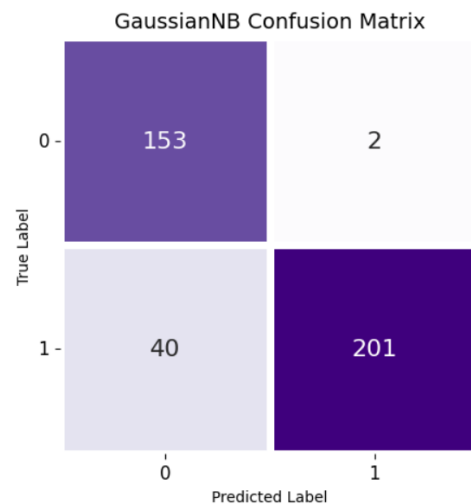
Sebanyak 1319 data pasien telah disiapkan dan dibagi menjadi dua kategori, yaitu data pelatihan dan data pengujian, dengan rasio 70:20. Dengan demikian, terdapat 923 data latih dan 396 data pengujian. Dari hasil pengujian yang telah dilakukan didapatkan nilai *accuracy*, *Precision*, *Recall*, dan *F-1 score* yang ditunjukkan pada tabel 4.3.

Tabel 4. 7 Hasil Pengujian Model C

Kriteria	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-1 score</i>	Jumlah data pengujian
0		0.83	0.99	0.90	155
1		0.99	0.87	0.92	241
	0.91				396

Pada tabel 4.3 memberikan hasil evaluasi yang menyeluruh terhadap kinerja model klasifikasi. Dalam konteks ini, model telah diuji terhadap dua kelas yang berbeda, diidentifikasi sebagai kelas 0 dan kelas 1. Ini bisa merepresentasikan dua kategori yang berbeda dalam *dataset* yaitu *negative* (0) dan *positive* (1). Untuk kelas 0 model menunjukkan nilai *Precision* sebesar 0.83, *Recall* 0.99, dan *F-1 score* 0.90

dengan 155 jumlah data pengujian. Untuk kelas 1 model menunjukkan nilai *Precision* sebesar 0.99, *Recall* 0.87, dan *F-1 score* 0.92 dengan 241 jumlah data pengujian. Terakhir *accuracy* didapatkan sebesar 0.91 dengan 396 jumlah data pengujian.



Gambar 4. 3 Hasil Confusion Matrix Model C

Berikut adalah hasil dari nilai aktual dan prediksi pada pengujian model C.

Tabel 4. 8 Hasil nilai aktual dan prediksi model C

<i>ID</i>	<i>Actual</i>	<i>Predicted</i>
677	1	1
1046	0	0
610	0	0
49	0	0
1284	1	0
...	...	...
141	0	0
1169	1	1
613	1	1
543	1	1
139	0	0

Jumlah yang dihasilkan nilai aktual dan yang sudah diprediksi bisa dilihat dalam tabel berikut.

Tabel 4. 9 Jumlah nilai aktual dan prediksi model C

Class	Actual	Predicted
1	241	203
0	193	193
Dari tabel diatas terlihat jumlah nilai yang salah pada sebanyak 42.		

Hasil pengujian model *Gaussian Naïve Bayes* memprediksi 153 data yang berhasil diprediksi *True Negative* (TN) dengan benar, 201 data yang berhasil diprediksi *True positive* (TP) dengan benar, 40 data yang salah diprediksi *negative* (FN), dan 2 data yang salah diprediksi *positive* (FP). Hasil perhitungan *accuracy*, *Precision*, *Recall*, dan *F-1 score* dari model B sebagai berikut :

$$Accuracy = \frac{153+201}{153+201+40+2} = 0.89$$

$$Precision = \frac{153}{153+40} = 0.79$$

$$Recall = \frac{153}{153+2} = 0.99$$

$$F-1\ score = \frac{2 \times 0.79 \times 0.99}{0.79 + 0.99} = 0.88$$

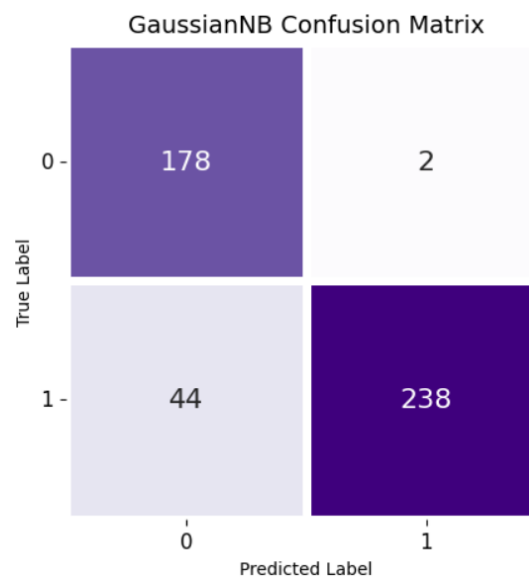
#### 4.2.4 Hasil Pengujian Model D

Sebanyak 1319 data pasien telah disiapkan dan dibagi menjadi dua kategori, yaitu data pelatihan dan data pengujian, dengan rasio 65:35. Dengan demikian, terdapat 857 data latih dan 462 data pengujian. Dari hasil pengujian yang telah dilakukan didapatkan nilai *accuracy*, *Precision*, *Recall*, dan *F-1 score* yang ditunjukkan pada tabel 4.4.

Tabel 4. 10 Hasil Pengujian Model D

Kriteria	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-1 score</i>	Jumlah data pengujian
0		0.83	0.99	0.90	180
1		0.99	0.87	0.93	282
	0.92				462

Pada tabel 4.4 memberikan hasil evaluasi yang menyeluruh terhadap kinerja model klasifikasi. Dalam konteks ini, model telah diuji terhadap dua kelas yang berbeda, diidentifikasi sebagai kelas 0 dan kelas 1. Ini bisa merepresentasikan dua kategori yang berbeda dalam *dataset* yaitu *negative* (0) dan *positif* (1). Untuk kelas 0 model menunjukkan nilai *Precision* sebesar 0.83, *Recall* 0.99, dan *F-1 score* 0.90 dengan 180 jumlah data pengujian. Untuk kelas 1 model menunjukkan nilai *Precision* sebesar 0.99, *Recall* 0.87, dan *F-1 score* 0.93 dengan 282 jumlah data pengujian. Terakhir *accuracy* didapatkan sebesar 0.92 dengan 462 jumlah data pengujian.



Gambar 4. 4 Hasil *Confusion Matrix* Model D

Berikut adalah hasil dari nilai aktual dan prediksi pada pengujian model D.

Tabel 4. 11 Hasil nilai aktual dan prediksi model D

<i>ID</i>	<i>Actual</i>	<i>Predicted</i>
677	1	1
1046	0	0
610	0	0
49	0	0
1284	1	0
...	...	...
893	1	1

5	0	0
100	1	1
424	1	1
772	0	0

Jumlah yang dihasilkan nilai aktual dan yang sudah diprediksi bisa dilihat dalam tabel berikut.

Tabel 4. 12 Jumlah nilai aktual dan prediksi model D

Class	Actual	Predicted
1	282	240
0	180	222
Dari tabel diatas terlihat jumlah nilai yang salah pada sebanyak 46		

Hasil pengujian model *Gaussian Naïve Bayes* memprediksi 178 data yang berhasil diprediksi *True Negative* (TN) dengan benar, 238 data yang berhasil diprediksi *True Positive* (TP) dengan benar, 44 data yang salah diprediksi *Negative* (FN), dan 2 data yang salah diprediksi *Positive* (FP). Hasil perhitungan *accuracy*, *Precision*, *Recall*, dan *F-1 score* dari model B sebagai berikut :

$$Accuracy = \frac{178+238}{178+238+44+2} = 0.90$$

$$Precision = \frac{178}{178+44} = 0.80$$

$$Recall = \frac{178}{178+2} = 0.99$$

$$F-1\ score = \frac{2 \times 0.79 \times 0.99}{0.79 + 0.99} = 0.88$$

### 4.3 Uji Coba *K-fold cross validation*

Pengujian menggunakan metode *k-fold cross-validation* secara keseluruhan dengan nilai k-5, k-10, k-15, dan k-20 dilakukan untuk mengevaluasi akurasi model terhadap *dataset*. Setelah melakukan pengujian, akurasi dari masing-masing pemilihan nilai k dievaluasi untuk menentukan nilai k yang memberikan hasil

terbaik. Pemilihan nilai  $k$  yang optimal dapat bervariasi tergantung pada ukuran *dataset* dan kompleksitas model yang digunakan. Setelah membandingkan hasil pengujian, nilai  $k$  yang memberikan akurasi terbaik dapat dipilih untuk digunakan dalam pembangunan model.

#### 4.3.1 *K-fold cross validation K-5*

Pengujian dengan  $k=5$  maka akan terjadi percobaan *5 fold* dari total keseluruhan data. Pada setiap *fold*, *dataset* akan dibagi menjadi 5 bagian yang sama besar. Setiap bagian akan digunakan sebagai data *testing* satu kali, sedangkan bagian-bagian lainnya akan digunakan sebagai data *training*. Hasil pengujian mendapatkan nilai dari masing-masing setiap *fold* yang ditunjukkan pada tabel 4.12.

Tabel 4. 13 Hasil *Accuracy 5-fold cross validation*

<i>Fold</i>	Hasil <i>Accuracy 5-fold cross validation</i>
1	88.26%
2	68.18%
3	68.56%
4	61.74%
5	65.40%
<b>Rata-rata</b>	70.43%

Dalam hasil *5-fold cross-validation* pada tabel 4.12, akurasi model bervariasi di setiap *fold*. pada *fold* pertama menunjukkan akurasi tertinggi sebesar 88.26%, yang menunjukkan bahwa model memiliki performa yang baik dalam menggeneralisasi data yang ada pada *fold* ini. Namun pada *fold* kedua, akurasinya turun drastis menjadi 68.18%, dan kemudian turun lagi menjadi 68.56% pada *fold* ketiga. Penurunan ini mungkin disebabkan oleh perbedaan distribusi atau pola data di setiap lipatan, yang dapat memengaruhi kemampuan model untuk



mempelajari dan menggeneralisasi pola dengan baik. Dan mendapatkan nilai terendah pada *fold* ke empat dengan akurasi 61.74%. Meskipun terdapat variasi dalam akurasi di setiap lipatan, rata-rata akurasi keseluruhan sekitar 70.43% menunjukkan bahwa model memiliki kemampuan yang cukup baik untuk memprediksi secara konsisten dalam kasus ini.

#### 4.3.2 K-fold cross validation K-10

Pengujian dengan k-10 maka akan terjadi percobaan 10 *fold* dari total keseluruhan data. Pada setiap *fold*, *dataset* akan dibagi menjadi 10 bagian yang sama besar. Setiap bagian akan digunakan sebagai data *testing* satu kali, sedangkan bagian-bagian lainnya akan digunakan sebagai data *training*. Hasil pengujian mendapatkan nilai dari masing-masing setiap *fold* yang ditunjukkan pada tabel 4.13.

Tabel 4. 14 Hasil Accuracy 10-fold cross validation

<i>Fold</i>	Hasil Accuracy 10-fold cross validation
1	77.27%
2	64.39%
3	63.64%
4	72.73%
5	69.70%
6	71.21%
7	61.36%
8	62.88%
9	67.42%
10	67.18%
<b>Rata-rata</b>	<b>67.78%</b>

Dalam hasil *10-fold cross-validation*, akurasi model menunjukkan variasi di setiap *fold*. *Fold* pertama memiliki akurasi tertinggi sebesar 77.27%, sementara *fold* ketujuh mencatat akurasi terendah sebesar 61.36%. Fluktuasi ini menggambarkan perbedaan dalam performa model saat diuji pada *subset* data yang berbeda.

Meskipun ada variasi, rata-rata akurasi model dari keseluruhan *fold* adalah 67.78%, menunjukkan bahwa secara keseluruhan, model memiliki kemampuan yang cukup baik untuk memprediksi data dengan konsistensi yang cukup tinggi.

#### 4.3.3 K-fold cross validation K-15

Pengujian dengan k-15 maka akan terjadi percobaan 15 *fold* dari total keseluruhan data. Pada setiap *fold*, *dataset* akan dibagi menjadi 15 bagian yang sama besar. Setiap bagian akan digunakan sebagai data *testing* satu kali, sedangkan bagian-bagian lainnya akan digunakan sebagai data *training*. Hasil pengujian mendapatkan nilai dari masing-masing setiap *fold* yang ditunjukkan pada tabel 4.14.

Tabel 4. 15 Hasil Accuracy 15-fold cross validation

<i>Fold</i>	Hasil Accuracy 15-fold cross validation
1	75.00%
2	73.86%
3	62.50%
4	63.64%
5	69.32%
6	71.59%
7	72.73%
8	68.18%
9	68.18%
10	63.64%
11	57.95%
12	65.91%
13	68.18%
14	65.91%
15	68.97%
<b>Rata-rata</b>	<b>67.70%</b>

Dalam hasil *15-fold cross-validation*, akurasi model menunjukkan variasi yang signifikan di setiap *fold*. *Fold* pertama memiliki akurasi tertinggi sebesar 70.00%, sementara *fold* kesebelas memiliki akurasi terendah hanya sebesar 57.95%. Fluktuasi ini mencerminkan perbedaan dalam performa model saat diuji pada *subset* data yang berbeda. Meskipun terjadi variasi, rata-rata akurasi model dari

keseluruhan *fold* adalah 67.70%, menunjukkan bahwa secara keseluruhan, model memiliki kemampuan yang cukup baik untuk memprediksi data dengan konsistensi yang relatif tinggi.

#### 4.3.4 K-fold cross validation K-20

Pengujian dengan k-20 maka akan terjadi percobaan 20 *fold* dari total keseluruhan data. Pada setiap *fold*, *dataset* akan dibagi menjadi 20 bagian yang sama besar. Setiap bagian akan digunakan sebagai data *testing* satu kali, sedangkan bagian-bagian lainnya akan digunakan sebagai data *training*. Hasil pengujian mendapatkan nilai dari masing-masing setiap *fold* yang ditunjukkan pada tabel 4.15.

Tabel 4. 16 Hasil Accuracy 20-fold cross validation

<i>Fold</i>	Hasil Accuracy 20-fold cross validation
1	75.76%,
2	71.21%,
3	69.70%,
4	63.64%,
5	60.61%,
6	66.67%,
7	75.76%,
8	69.70%,
9	71.21%,
10	69.70%,
11	71.21%,
12	65.15%,
13	66.67%,
14	59.09%,
15	59.09%,
16	65.15%,
17	68.18%,
18	66.67%,
19	71.21%,
20	63.08%
<b>Rata-rata</b>	67.47%

Dalam hasil 20-fold cross-validation, terlihat variasi yang cukup signifikan dalam akurasi model antara setiap *fold*. Misalnya, *Fold 1* dan *Fold 7* menunjukkan

akurasi tertinggi, masing-masing sebesar 75.76%, sementara *Fold* 14 dan *Fold* 15 memiliki akurasi terendah, masing-masing hanya 59.09%.

Perbedaan dalam akurasi ini mencerminkan variasi dalam performa model ketika diuji pada *subset* data yang berbeda. Namun, meskipun terjadi fluktuasi dalam akurasi di setiap *fold*, rata-rata akurasi model dari seluruh *fold* adalah sekitar 67.47%. Ini menunjukkan bahwa secara keseluruhan, model memiliki kemampuan yang cukup baik untuk memprediksi data dengan konsistensi yang relatif tinggi, meskipun terdapat variasi dalam performa pada *subset* data tertentu.

#### 4.4 Pembahasan

Dalam penelitian ini, penulis menggunakan 1.319 data rekam medis pasien yang berasal dari Elsevier Mendeley Data Repository: *Heart Attack Dataset* sebagai *dataset*. Data tersebut perlu melalui tahap *preprocessing* sebelum dapat digunakan oleh sistem. Tahap *preprocessing* meliputi *Label Encoding* dan normalisasi data menggunakan *feature scaling* yang bertujuan untuk mengubah data kategorikal menjadi numerik dan menormalisasi rentang nilai data. Tahap ini sangat penting untuk meningkatkan performa sistem dalam mengolah data. Setelah data siap, penulis membagi data menjadi data latih dan data uji dengan proporsi yang berbeda. Kemudian Dilakukan pencarian parameter terbaik menggunakan *GridSearchCV*, di mana kita mencari parameter terbaik untuk *var\_smoothing* atau parameter pada *Gaussian naive bayes* dengan menggunakan nilai *logspace* antara 0 hingga -9. Kemudian, *GridSearchCV* akan membagi data latih ( $x_{train}$ ,  $y_{train}$ ) menjadi beberapa lipatan (*fold*), di mana setiap lipatan digunakan secara bergantian sebagai data validasi sementara yang digunakan untuk mengevaluasi model yang dilatih

pada data yang tersisa. Dalam kasus ini, penulis menggunakan 'cv=5' jumlah lipatan (*fold*). Setelah semua proses pemodelan telah dilakukan, maka masuk ke tahap akhir yaitu evaluasi model dengan menggunakan *Confusion Matrix* untuk mendapatkan nilai *accuracy*, *Precision*, *Recall*, dan *F1-score*.

Pada skenario A, *Dataset* dibagi menjadi data latih dan data uji dengan rasio 90:10. Pengujian dilakukan dengan cara mengambil data secara acak dengan nilai *random state* sebesar 42. Tujuannya adalah untuk melihat *accuracy* model saat digunakan pada data baru yang belum pernah dilihat sebelumnya. Pada skenario ini hasil menunjukkan bahwa model memiliki akurasi sebesar 70%. Kemudian digunakan *Confusion Matrix* untuk mendapatkan metrik-metrik lain seperti *accuracy*, *Precision*, *Recall*, dan *F1-score* dan hasilnya cukup baik.

Tabel 4. 17 Pembahasan hasil skenario A

Kriteria	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-1 score</i>	Jumlah data pengujian
0		0.54	1.00	0.70	50
1		1.00	0.49	0.66	82
	0.68				132

Pada skenario B, evaluasi dilakukan dengan menggunakan model *Gaussian naive bayes* pada *dataset* yang dibagi menjadi data latih dan data uji dengan rasio 80:20. Pengujian dilakukan dengan mengambil data secara acak menggunakan nilai *random state* sebesar 42, tujuannya adalah untuk melihat seberapa baik model dapat memprediksi data baru yang belum pernah dilihat sebelumnya. Hasil evaluasi menunjukkan bahwa model memiliki akurasi sebesar 79%. kemudian digunakan *Confusion Matrix* untuk mendapatkan metrik-metrik lain seperti *accuracy*, *Precision*, *Recall*, dan *F1-score* dan hasilnya cukup baik.

Tabel 4. 18 Pembahasan hasil skenario B

Kriteria	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-1 score</i>	Jumlah data pengujian
0		0.65	0.99	0.78	101
1		0.99	0.66	0.79	163
	0.79				264

Pada skenario C, *Dataset* dibagi menjadi data latih dan data uji dengan rasio 70:30. Pengujian dilakukan dengan mengambil data secara acak menggunakan nilai *random state* sebesar 42, tujuannya adalah untuk melihat seberapa baik model dapat memprediksi data baru yang belum pernah dilihat sebelumnya. Hasil evaluasi menunjukkan bahwa model memiliki akurasi sebesar 91%. kemudian digunakan *Confusion Matrix* untuk mendapatkan metrik-metrik lain seperti *accuracy*, *Precision*, *Recall*, dan *F1-score* dan hasilnya cukup baik.

Tabel 4. 19 Pembahasan hasil skenario C

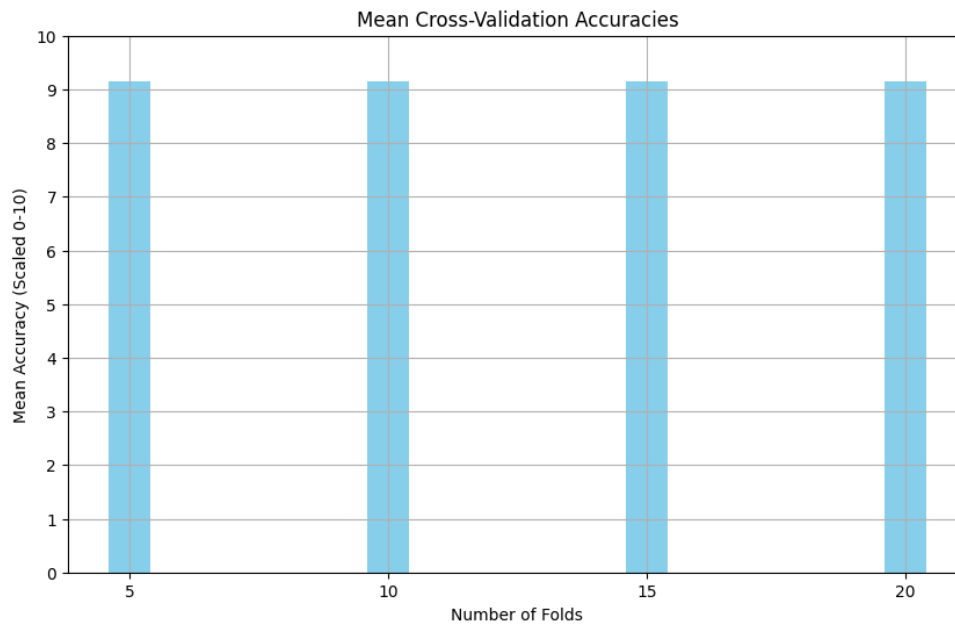
Kriteria	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-1 score</i>	Jumlah data pengujian
0		0.83	0.99	0.90	155
1		0.99	0.87	0.92	241
	0.91				396

Pada skenario 65:35, *Dataset* dibagi menjadi data latih dan data uji dengan rasio 65:35. Pengujian dilakukan dengan mengambil data secara acak menggunakan nilai *random state* sebesar 42, dengan tujuan untuk melihat seberapa baik model dapat memprediksi data baru yang belum pernah dilihat sebelumnya. Hasil evaluasi menunjukkan bahwa model memiliki akurasi sebesar 92%. Kemudian digunakan *Confusion Matrix* untuk mendapatkan metrik-metrik lain seperti *accuracy*, *Precision*, *Recall*, dan *F1-score* dan hasilnya cukup baik.

Tabel 4. 20 Pembahasan hasil skenario D

Kriteria	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-1 score</i>	Jumlah data pengujian
0		0.83	0.99	0.90	180
1		0.99	0.87	0.93	282
	0.92				462

Kemudian, Setiap pengujian *K-fold cross validation* dengan nilai K yang berbeda, seperti K=5, 10, 15, dan 20, menghasilkan variasi dalam hasil akurasi pada setiap lipatan.



Gambar 4. 5 Grafik rata-rata akurasi *K-fold cross-validation* dengan *Outliers*

Misalnya, ketika melakukan *K-fold cross validation* dengan K=5, percobaan terdiri dari 5 lipatan dari total data yang ada. Masing-masing lipatan digunakan sebagai data uji satu kali, sementara lipatan lainnya sebagai data latih. Hasil akurasi dari setiap lipatan menunjukkan variasi, di mana lipatan pertama memiliki akurasi tertinggi 88,26% , namun akurasi pada lipatan berikutnya dapat turun secara signifikan. Penurunan ini mungkin disebabkan oleh perbedaan dalam distribusi atau pola data di setiap lipatan, yang memengaruhi kemampuan model untuk belajar dan menggeneralisasi pola dengan baik.

Variasi dalam akurasi antara setiap lipatan (*fold*) dalam *cross-validation* bisa disebabkan oleh beberapa faktor yang mungkin terjadi. Seperti adanya perbedaan dalam distribusi data antara lipatan-lipatan dapat menjadi penyebab utama variasi dalam akurasi. Fenomena serupa terjadi pada pengujian dengan  $K=10$ ,  $K=15$ , dan  $K=20$ , di mana variasi dalam akurasi antar lipatan dapat diamati. Meskipun terjadi fluktuasi dalam akurasi di setiap lipatan, rata-rata akurasi dari seluruh lipatan menunjukkan bahwa model memiliki kemampuan yang cukup baik untuk memprediksi data secara konsisten dan rata-rata tertinggi didapatkan pada  $K=5$  yaitu sebesar 70.43%.

#### 4.5 Integrasi Islam

Dalam Al-qur'an telah dijelaskan ayat-ayat tentang (Alqolbu) jantung, seperti pada surah Al-haqqah ayat 46 :

ثُمَّ لَقَطَعْنَا مِنْهُ الْوَتِينَ ۗ

“Kemudian, Kami benar-benar memotong urat nadinya.” (Q.S Al-Haqqah : 46)

Menurut Tafsir dari Ibnu Katsir, Ibnu Abbas mengatakan bahwa al-wafin artinya urat tali jantungnya. Hal yang semisal dikatakan oleh Ikrimah, Sa'id ibnu Jubair, Al-Hakam, Qatadah, Adh-Dhahhak, Muslim Al-Batin, dan Abu Sakhr alias Humaid ibnu Ziad. Menurut Muhammad ibnu Ka'b, makna yang dimaksud ialah jantung dan semua uratnya serta semua bagian yang berada di dekatnya.

Kemudian dalam Al-Qur'an juga dijelaskan ayat-ayat yang berhubungan dengan klasifikasi berbagai aspek kehidupan. Ayat-ayat tersebut memberikan panduan dalam memisahkan antara yang baik dan yang buruk, yang halal dan yang



haram, serta yang benar dan yang salah. Seperti yang terdapat pada surah Al-An'am ayat 141 :

وَهُوَ الَّذِي أَنشَأَ جَنَّاتٍ مَّعْرُوشَاتٍ وَغَيْرَ مَعْرُوشَاتٍ وَالنَّخْلَ وَالزَّرْعَ مُخْتَلِفًا أَكْلُهُمُ وَالزَّيْتُونَ وَالرُّمَانَ مِثْلَهَا  
 وَغَيْرَ مِثْلَهَا كُلُوا مِنْ ثَمَرِهِ إِذَا أَثْمَرَ وَآتُوا حَقَّهُ يَوْمَ حَصَادِهِ وَلَا تُسْرِفُوا إِنَّهُ لَا يُحِبُّ  
 الْمُسْرِفِينَ

*“Dan dialah yang menumbuhkan tanaman-tanaman yang merambat dan yang tidak merambat, pohon kurma, tanaman yang beraneka ragam rasanya, serta zaitun dan delima yang serupa (bentuk dan warnanya) dan tidak serupa (rasanya). Makanlah buahnya apabila ia berbuah dan berikanlah haknya (zakatnya) pada waktu memetik hasilnya. Akan tetapi, janganlah berlebih-lebihan. Sesungguhnya Allah tidak menyukai orang-orang yang berlebih-lebihan” (QS. Al-An'am :141).*

Menurut tafsir dalam kitab al-imam ibnu katsir, Allah SWT dalam firman-Nya menjelaskan bahwa Dia adalah Yang menciptakan segala tanaman, buah-buahan, dan binatang ternak, yang semua itu diperlakukan oleh orang-orang musyrik dengan berbuat sekehendak hatinya terhadap ternak-ternak mereka berdasarkan pemikiran mereka yang sesat. Mereka menjadikannya ke dalam beberapa bagian dan pengkategorian, lalu mereka menjadikan sebagiannya haram dan sebagian yang lainnya halal. Untuk itu Allah berfirman: “Dan Dialah yang menciptakan kebun-kebun yang berjunjung dan yang tidak berjunjung.” (*Al-Anam: 141*) dari sahabat Ali ibnu Abu Talhah meriwayatkan dari Ibnu Abbas, bahwa makna ma'rusyatin ialah yang merambat Menurut riwayat yang lain, “ma'rusyat” artinya tanaman yang ditanam oleh manusia. Sedangkan “ghairo marusyat” artinya tanam-tanaman berbuah yang tumbuh dengan sendirinya di hutan-hutan dan bukit-bukit. ‘Atha’ Al-Khurasani meriwayatkan dari Ibnu Abbas, bahwa makna “ma'rusyat” ialah tanaman anggur yang dirambatkan, sedangkan “ghaira ma'rusyat”

ialah tanaman anggur yang tidak dirambatkan. Hal yang sama dikatakan oleh As-Suddi. Ibnu Juraij mengatakan sehubungan dengan makna firman-Nya: “Yang serupa dan yang tidak serupa.” (*Al-An'am: 141*) Maksudnya, yang serupa bentuknya, tetapi tidak sama rasanya. Muhammad ibnu Ka'b mengatakan sehubungan dengan makna firman-Nya: “Makanlah buahnya bila berbuah.” (*Al-An'am: 141*) Yaitu buah kurma dan buah anggurnya.

Selain itu, dalam surah Al-Hujurat ayat 13, Allah menjelaskan bahwa manusia diciptakan berbangsa-bangsa dan bersuku-suku agar saling mengenal, menandakan pentingnya perbedaan dan keberagaman dalam kehidupan sosial.

يَا أَيُّهَا النَّاسُ إِنَّا خَلَقْنَاكُمْ مِنْ ذَكَرٍ وَأُنْثَىٰ وَجَعَلْنَاكُمْ شُعُوبًا وَقَبَائِلَ لِتَعَارَفُوا ۗ إِنَّ أَكْرَمَكُمْ عِنْدَ اللَّهِ أَتْقَىٰكُمْ ۚ إِنَّ اللَّهَ عَلِيمٌ خَبِيرٌ

*“Wahai manusia, sesungguhnya Kami telah menciptakan kamu dari seorang laki-laki dan perempuan. Kemudian, Kami menjadikan kamu berbangsa-bangsa dan bersuku-suku agar kamu saling mengenal. Sesungguhnya yang paling mulia di antara kamu di sisi Allah adalah orang yang paling bertakwa. Sesungguhnya Allah Maha Mengetahui lagi Mahateliti”* (Q.S Al-Hujurat : 13).

Menurut tafsir dalam kitab al-imam ibnu katsir, “Hai manusia, sesungguhnya Kami menciptakan kamu dari seorang laki-laki dan seorang perempuan dan menjadikan kamu berbangsa-bangsa dan bersuku-suku supaya kamu saling kenal-mengenal. Sesungguhnya orang yang paling mulia di antara kamu di sisi Allah ialah orang yang paling bertakwa di antara kamu. Sesungguhnya Allah Maha Mengetahui lagi Maha Mengenal”. Allah SWT menceritakan kepada manusia bahwa Dia telah menciptakan mereka dari diri yang satu dan darinya Allah menciptakan istrinya, yaitu Adam dan Hawa, kemudian Dia menjadikan mereka berbangsa-bangsa.

Pengertian bangsa dalam bahasa Arab adalah sya 'bun yang artinya lebih besar daripada kabilah, sesudah kabilah terdapat tingkatan-tingkatan lainnya yang lebih kecil seperti fasa-il (puak), 'asya-ir (Bani), 'ama-ir, Afkhad, dan lain sebagainya. Menurut suatu pendapat, yang dimaksud dengan syu'ub ialah kabilah-kabilah yang non-Arab. Sedangkan yang dimaksud dengan kabilah-kabilah ialah khusus untuk bangsa Arab, seperti halnya kabilah Bani Israil disebut Asbat. Hal ini juga sudah dijelaskan dalam mukadimah kitab yang berjudul Al-Qasdu wal Umam fi Ma'rifati Ansabil Arab wal 'Ajam.

Pada garis besarnya semua manusia bila ditinjau dari unsur kejadiannya yaitu tanah liat sampai dengan Adam dan Hawa a.s. sama saja. Sesungguhnya perbedaan keutamaan di antara mereka karena perkara agama, yaitu ketaatannya kepada Allah dan Rasul-Nya. Karena itulah sesudah melarang perbuatan menggunjing dan menghina orang lain, Allah ﷻ berfirman mengingatkan mereka, bahwa mereka adalah manusia yang mempunyai martabat yang sama: Hai manusia, sesungguhnya Kami menciptakan kamu dari seorang laki-laki dan seorang perempuan dan menjadikan kamu berbangsa-bangsa dan bersuku-suku supaya kamu saling kenal-mengenal. (Al-Hujurat: 13) Agar mereka saling mengenal di antara sesamanya, masing-masing dinisbatkan kepada kabilah (suku atau bangsa) nya.

الْمِ ۙ غَلَبَتِ الرُّومَ ۚ فِي ۙ اَدْنَى الْاَرْضِ ۚ وَهُمْ مِّنْ ۙ بَعْدِ غَلَبِهِمْ سَيَغْلِبُونَ ۚ فِي ۙ بَضْعِ سِنِينَ ۚ ۙ لِّلّٰهِ الْاَمْرُ مِّنْ قَبْلُ  
وَمِنْ ۙ بَعْدِ وَيَوْمَئِذٍ ۙ يَفْرَحُ الْمُؤْمِنُونَ ۙ

*Alif Lam Mim. Telah dikalahkan bangsa Romawi di negeri yang terdekat dan mereka sesudah dikalahkan itu akan menang dalam beberapa tahun (lagi). Bagi Allah-lah urusan sebelum dan sesudah (mereka menang). Dan di hari (kemenangan bangsa Romawi) itu bergembiralah orang-orang yang beriman. (Q.S Ar-Rum: 1-4)*

Menurut tafsir Ibnu Katsir, Ayat-ayat ini diturunkan ketika Sabur (Raja Persia) berhasil mengalahkan tentara Romawi dan berhasil merebut negeri-negeri Syam serta bagian lainnya yang termasuk ke dalam wilayah kerajaan Romawi dari tanah Jazirah Arabia, juga sebagian besar wilayah kerajaan Romawi, sehingga Kaisar Romawi Heraklius terpaksa mundur dan mengungsi ke kota Konstantinopel. Ia dikepung oleh Raja Sabur dan bala tentaranya di kota Konstantinopel dalam waktu yang cukup lama, tetapi pada akhirnya kawasan kerajaan Romawi berhasil direbut kembali oleh Heraklius dari tangan orang-orang Persia.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Dalam penelitian ini, penulis menggunakan *dataset* rekam medis pasien dari Elsevier Mendeley Data Repository: *Heart Attack Dataset* yang terdiri dari 1.319 data. Data ini harus melalui tahap *preprocessing* seperti *Label Encoding* dan normalisasi data menggunakan *feature scaling* untuk mengubah data kategorikal menjadi numerik dan menormalisasi rentang nilai data. Setelah data siap, dilakukan pembagian data menjadi data latih dan data uji dengan proporsi yang berbeda, diikuti dengan pencarian parameter terbaik menggunakan *GridSearchCV*. Proses evaluasi model dilakukan dengan menggunakan *Confusion Matrix* untuk mendapatkan nilai *accuracy*, *Precision*, *Recall*, dan *F1-score*.

Pada tiga skenario yang berbeda, penulis membagi *dataset* dengan rasio 90:10, 80:20, 70:30, dan 65:35 untuk data latih dan data uji. Hasil evaluasi menunjukkan bahwa model memiliki akurasi yang bervariasi, mulai dari 70% hingga 92%, tergantung pada proporsi pembagian data. Pada setiap skenario, metrik evaluasi seperti *accuracy*, *Precision*, *Recall*, dan *F1-score* dievaluasi dengan menggunakan *Confusion Matrix* dan menunjukkan hasil yang cukup baik.

Selain itu, penulis juga melakukan pengujian menggunakan *K-fold cross validation* (dengan  $K=5, 10, 15,$  dan  $20$ ), terdapat variasi signifikan dalam akurasi model antar fold. Dalam percobaan *K-fold cross validation*, hasil paling bagus dicapai pada  $K=5$  dengan akurasi tertinggi sebesar 88.26%. Sedangkan hasil terendahnya terjadi pada  $K=15$ , dengan akurasi terendah sebesar 57.95%. Rata-rata

akurasi tertinggi dari semua percobaan *K-fold cross validation* adalah 70.43%, yang terjadi pada K=5. Sedangkan rata-rata akurasi terendah adalah 67.47%, yang terjadi pada K=20.

Dalam analisis penulis, variasi ini dipengaruhi oleh perbedaan dalam distribusi data serta ukuran sampel dalam proses pengambilan data training dan testing disetiap fold, untuk mengatasinya penulis melakukan analisis lebih lanjut dengan melakukan penghapusan *Outliers*. Dan berhasil mendapatkan rata-rata akurasi yang lebih baik dan stabil.

## 5.2 Saran

Berdasarkan dengan penelitian yang sudah dilakukan, penulis menyadari penelitian ini masih belum sempurna. Oleh karena itu, penulis memberikan saran untuk peneliti selanjutnya agar mendapatkan hasil performa sistem yang lebih baik berdasarkan hasil yang telah diterima.

1. Perdalam *Preprocessing Data*: Coba untuk melakukan cek missing value dan penghapusan *outliers* agar rata-rata pada pengujian K-fold cross Validation menjadi lebih baik.
2. Eksplorasi Metode Pengujian: Coba variasi metode pengujian selain *K-fold cross validation*, seperti *Leave-One-Out Cross Validation* atau *Bootstrap Cross Validation*, untuk memahami keandalan model dengan lebih baik.
3. Lakukan analisis terhadap variasi model dalam akurasi model antar *fold* dalam *K-fold cross validation*. Gunakan teknik analisis data lanjutan atau model tambahan untuk memahami penyebab variasi tersebut dengan lebih baik.

4. Coba mengganti representasi numerik pada kelas 0 dan 1 menjadi dengan 0 (negatif), 100 (positif). atau untuk menentukan laki dan perempuan tidak harus 0 dan 1 bisa juga diganti dengan yang lain.

## DAFTAR PUSTAKA

- Alhamad, A., Azis, A. I. S., Santoso, B., & Taliki, S. (2019). *Prediksi Penyakit Jantung Menggunakan Metode-Metode Machine learning Berbasis Ensemble – Weighted Vote*. 5(3), 352–360.
- Amrullah, S., Rosjidi, C. H., Dhessa, D. B., Wurjatmiko, A. T., Wurjatmiko, A. T., Kendari, K., Akut, I. M., & Infarction, A. M. (2022). *Jurnal ilmiah karya kesehatan*. 02.
- Azizah, N., & Goejantoro, R. (2019). *METODE NAIVE BAYES DENGAN PENDEKATAN*. 8–14.
- Dodo, F., Hulu, P., Nadeak, T. Z., Lumbantong, R. R., Dharma, A., Teknologi, F., Studi, P., Informatika, T., Teknologi, F., Komputer, I., Studi, P., Informatika, T., & Teknologi, F. (2019). *Penggunaan Machine Learning*. 2, 391–399.
- harbanu H Mariyono, A. S. (2007). *gagal jantung*. 85–94.
- Kamel, H., & Al-tuwajjari, J. M. (2019). Cancer Classification Using *Gaussian naive bayes* Algorithm. *2019 International Engineering Conference (IEC)*, 165–170.
- Kamil, M., Endra, T., & Tju, E. (2022). *Naive Bayes dan Confusion Matrix untuk Efisiensi Analisa Intrusion Detection System Alert*. 02, 81–88.
- Kasus, S., Johannes, P. W. Z., Naomi, W. S., Picauly, I., & Toy, S. M. (2021). *Media Kesehatan Masyarakat FAKTOR RISIKO KEJADIAN PENYAKIT JANTUNG KORONER Media Kesehatan Masyarakat*. 3(1), 99–107.
- Ketut, S. I., Kiki, W. P., Anak, Y., Gede, A., & Pratama, W. (2022). *INFARK MIOKARD AKUT DENGAN ELEVASI SEGMENT ( IMA-EST ) ANTERIOR EKSTENSIF : LAPORAN KASUS*. 2(1), 22–32.
- Laporan Riskesdas 2018 Nasional.pdf*. (n.d.).
- Manikandan, S. (2017). Heart Attack Prediction System. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 817–820.
- Nursita, H., & Pratiwi, A. (2020). *Peningkatan Kualitas Hidup pada Pasien Gagal Jantung : A Narrative Review Article*. 13(1), 10–21.
- Octaviary, shafa risqi. (2022). *DETEKSI AWAL PENYAKIT GAGAL JANTUNG BERDASARKAN FAKTOR RISIKO MENGGUNAKAN METODE NAIVE BAYES*.
- Puspa Wardhani, M. Z. A. (2016). *CLINICAL PATHOLOGY AND CLINICAL PATHOLOGY AND*. 22(1).
- Qadrini, L. (2021). *DECISION TREE DAN ADABOOST PADA KLASIFIKAS*



*PENERIMA PROGRAM BANTUAN SOSIAL. 2(7).*

- Quswatun hasanah, hardian oktaviano, yeni dwi rahayu. (2022). *Analisis Algoritma Gaussian naive bayes Terhadap Klasifikasi Data Pasien Penderita Gagal Jantung Gaussian naive bayes Algorithm Analysis Of Data Classification Of Heart Failure Patiens Jurnal Smart Teknologi. 3(4), 382–389.*
- Riani, A., Susianto, Y., Rahman, N., & Ali, U. D. (2019). *Implementasi Data Mining Untuk Memprediksi Penyakit Jantung Menggunakan Metode Naive bayes Data Mining Implementation to Predict Heart Disease using Naive bayes Method. 1(01), 25–34. <https://doi.org/10.35970/jinita.v1i01.64>*
- Riza, rizqi robbi arisandi, budi warsito. (2022). *Aplikasi Naive Bayes classifier (nbc) pada klasifikasi status gizi balita stunting dengan pengujian K-fold cross validation 1,2,3. 11, 130–139.*
- Rizkia, ayu putri. (2023). *IDENTIFIKASI JENIS TANAMAN OBAT INDONESIA BERDASARKAN BENTUK CITRA DAUN MENGGUNAKAN METODE DETEKSI TEPI DAN GAUSSIAN NAÏVE BAYES.*
- Susana, H., & Suarna, N. (2022). *PENERAPAN MODEL KLASIFIKASI METODE NAIVE BAYES. 4(1), 2–9.*
- Tan, J. (n.d.). *A critical look at the current train / test Split in Machine learning arXiv : 2106 . 04525v1 [ cs . LG ] 8 Jun 2021.*
- Ulfatul, D., Rachmad, M., Oktavianto, H., & Rahman, M. (2022). *Perbandingan Metode K-Nearest Neighbor Dan Gaussian naive bayes Untuk Klasifikasi Penyakit Stroke Comparison Of K-Nearest Neighbor And Gaussian naive bayes Methods For Stroke Disease Classification. 3(4), 405–412.*
- Wahyudi, M. A., Jiddan, A. N., & Rohman, R. S. (2023). *Penerapan Algoritma Naive bayes Untuk Prediksi Penyakit Diare Pada Balita : Data dengan class yang belum diketahui : Hipotesis data X merupakan suatu class spesifik  $P(H | X)$  : Probabilitas probability )  $P(H)$  probability ) berdasar kondisi pada hipotesis  $H P(X)$ . 2(2).*