IMPLEMENTASI *EXTREME GRADIENT BOOSTING* (XGBOOST) DENGAN *RECURSIVE FEATURE ELIMINATION* (RFE) UNTUK KLASIFIKASI PENYAKIT SERANGAN JANTUNG

SKRIPSI

Oleh :
AWANG SIREGAR LENGGAH PERMEI
NIM. 210605110145

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026

IMPLEMENTASI *EXTREME GRADIENT BOOSTING* (XGBOOST) DENGAN *RECURSIVE FEATURE ELIMINATION* (RFE) UNTUK KLASIFIKASI PENYAKIT 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 :
AWANG SIREGAR LENGGAH PERMEI
NIM. 210605110145

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026

HALAMAN PERSETUJUAN

IMPLEMENTASI *EXTREME GRADIENT BOOSTING* (XGBOOST) DENGAN *RECURSIVE FEATURE ELIMINATION* (RFE) UNTUK KLASIFIKASI PENYAKIT SERANGAN JANTUNG

SKRIPSI

Oleh :
AWANG SIREGAR LENGGAH PERMEI
NIM. 210605110145

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 5 Juni 2026

Pembimbing I,	Pembimbing II,
Okta Qomaruddin Aziz, M.Kom	Dr. Irwan Budi Santoso, M.Kom
NIP. 199110192019031013	NIP. 19770103 201101 1 004

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

Supriyono, M. Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

IMPLEMENTASI *EXTREME GRADIENT BOOSTING* (XGBOOST) DENGAN *RECURSIVE FEATURE ELIMINATION* (RFE) UNTUK KLASIFIKASI PENYAKIT SERANGAN JANTUNG

SKRIPSI

Oleh :
AWANG SIREGAR LENGGAH PERMEI
NIM. 210605110145

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 12 Juni 2026

Susunan Dewan Penguji

Ketua Penguji	: Supriyono, M. Kom NIP. 19841010 201903 1 012	( )
Anggota Penguji I	: Fajar Rohman Hariri, M.Kom NIP. 19890515 201801 1 001	( )
Anggota Penguji II	: Okta Qomaruddin Aziz, M.Kom NIP. 199110192019031013	( )
Anggota Penguji III	: Dr. Irwan Budi Santoso, M.Kom NIP. 19770103 201101 1 004	( )

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

Supriyono, M. Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Awang Siregar Lenggah Permei
NIM	: 210605110145
Fakultas / Program Studi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Implementasi *Extreme Gradient Boosting* (XGBoost) dengan *Recursive Feature Elimination* (RFE) untuk Klasifikasi Penyakit 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, 12 Juni 2026
Yang membuat pernyataan,

Awang Siregar Lenggah Permei
NIM. 210605110145

MOTTO

“Karya ini adalah langkah awal penulis untuk tumbuh menjadi pribadi yang lebih baik”

“Berani untuk memulai, berkomitmen untuk menuntaskan.”

“Jangan menyerah, terus melangkah dan selalu bersemangat untuk masa depan.”

HALAMAN PERSEMBAHAN

Alhamdulillahi rabbil 'alamin segala puji bagi Allah SWT yang telah melimpahkan rahmat, hidayah, serta kemudahan sehingga penulis dapat menyelesaikan skripsi ini dengan baik.

Skripsi ini adalah wujud terima kasih terdalam peneliti yang dipersembahkan kepada Bapak Sutikno, Ibu Maryati, dan Kakak Tata Sutrafia Armeyntan, serta segenap keluarga besar. Kehadiran mereka yang selalu mengalirkan doa, harapan dan semangat menjadikan motivasi utama bagi peneliti untuk terus bertahan dan melangkah maju dalam setiap proses akademis ini.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillahi Rabbil 'Alamin, Segala puji dan syukur kehadirat Allah SWT yang telah memberikan nikmat dan anugerah-Nya sehingga skripsi yang berjudul "Implementasi *Extreme Gradient Boosting* (XGBoost) dengan *Recursive Feature Elimination* (RFE) untuk Klasifikasi Penyakit Serangan Jantung" dapat terselesaikan dengan baik. Sholawat serta salam semoga selalu tercurahkan kepada Nabi Muhammad SAW yang syafaatnya kita nanti-nantikan pada hari akhir kelak.

Peneliti mengucapkan terima kasih kepada semua pihak yang telah membimbing serta memberi arahan dalam penulisan skripsi ini. Ucapan terima kasih peneliti sampaikan kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si, CAHRM, CRMP., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom., selaku ketua Program Studi Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Okta Qomaruddin Aziz, M.Kom, selaku pembimbing I, yang telah memberikan pendampingan secara berkesinambungan melalui bimbingan, arahan, serta masukan yang sangat berarti selama proses penyusunan skripsi ini. Dengan dukungan tersebut, penulis dapat menyelesaikan penelitian ini hingga tuntas.

5. Dr. Irwan Budi Santoso, M.Kom selaku pembimbing II, yang dengan penuh ketulusan senantiasa memberikan bimbingan serta dengan cermat mengoreksi berbagai kesalahan dalam proses penyusunan skripsi ini.
6. Supriyono, M.Kom selaku Ketua Penguji yang senantiasa memberikan saran, kritik, serta arahan yang membangun kepada penulis, mulai dari seminar proposal hingga pelaksanaan sidang skripsi.
7. Fajar Rohman Hariri, M.Kom yang senantiasa memberikan saran, kritik, dan masukan yang membangun sepanjang proses penyusunan skripsi, dari seminar proposal hingga sidang skripsi.
8. Nia Faricha, S.Si selaku admin Program Studi Teknik Informatika yang telah dengan sabar memberikan informasi, bantuan, serta arahan selama masa perkuliahan hingga proses penyusunan skripsi ini.
9. Segenap Seluruh dosen dan staf akademik Program Studi Teknik Informatika UIN Maulana Malik Ibrahim Malang, yang telah memberikan ilmu, pengalaman, serta pelayanan akademik selama masa studi penulis.
10. Kepada teman-teman Kelas E dan segenap keluarga besar Teknik Informatika khususnya angkatan 2021 "Aster". atas kebersamaan, dukungan, dan semangat yang telah diberikan.
11. Skripsi ini didedikasikan sepenuhnya untuk kedua orang tua, kakak, serta segenap keluarga besar. Terima kasih atas doa-doa tulus yang selalu dipanjatkan, restu yang melapangkan jalan, serta nasihat yang tak pernah henti diberikan demi mendampingi proses pertumbuhan dan perjuangan penulis selama ini.

Penulis menyadari bahwa skripsi ini masih memiliki berbagai keterbatasan dan kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan sebagai bahan perbaikan di masa mendatang. Penulis berharap skripsi ini dapat memberikan manfaat serta menambah wawasan bagi para pembaca.

Malang, 12 Juni 2026

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Permei, Awang Siregar Lenggah. 2026. **Implementasi Extreme Gradient Boosting (XGBoost) dengan Recursive Feature Elimination (RFE) untuk Klasifikasi Penyakit Serangan Jantung**. Skripsi. Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Okta Qomaruddin Aziz, S.Si., M.Kom (II) Dr. Irwan Budi Santoso, M.Kom.

Kata Kunci: XGBoost, Penyakit Serangan Jantung, *Recursive Feature Elimination* (RFE), *Random Undersampling*, Klasifikasi.

Pemilihan topik skripsi ini dilatarbelakangi oleh tingginya angka kematian global akibat penyakit kardiovaskular, khususnya serangan jantung, sehingga sistem deteksi dini yang akurat sangat dibutuhkan. Namun, pengembangan model klasifikasi sering kali terkendala oleh masalah data yang tidak seimbang (i*mbalanced* data) dan karakteristik klinis pasien yang saling tumpang tindih (*overlapping features*). Oleh karena itu, penelitian ini bertujuan untuk mengoptimalkan performa algoritma *Extreme Gradient Boosting* (XGBoost) dalam memprediksi risiko penyakit jantung. Penelitian ini menggunakan dataset medis dari Kaggle yang terdiri atas 4.238 data pengamatan dengan 15 variabel input. Pengujian dilakukan secara komprehensif melalui sembilan skenario eksperimen utama, yang mencakup dua belas kombinasi pengujian dengan mengombinasikan metode penyeimbangan data (*Random Undersampling* dan SMOTE) serta teknik seleksi fitur (*Recursive Feature Elimination* dan *Thresholding*). Hasil penelitian menunjukkan bahwa penerapan *Random Undersampling* mampu meningkatkan sensitivitas model secara signifikan jika dibandingkan dengan metode dasarnya (*baseline*). Pada skenario dasar (*Pure+Full*), model memang menghasilkan akurasi sebesar 83,88%, tetapi nilai *recall*-nya sangat rendah, yakni hanya 14,29%, yang membuktikan bahwa model masih sangat bias terhadap kelas mayoritas. Sebaliknya, skenario RUS+*Threshold* (0,07) berhasil memperbaiki kelemahan tersebut dengan mencapai nilai recalltertinggi sebesar 75,00% serta skor ROC-AUC sebesar 71,58%. Selain itu, berdasarkan analisis feature importance, fitur jenis kelamin, usia, dan kadar glukosa teridentifikasi sebagai indikator klinis yang paling menonjol. Secara keseluruhan, temuan ini membuktikan bahwa strategi undersampling sangat efektif untuk meningkatkan objektivitas model dalam mendeteksi risiko penyakit jantung secara dini, meskipun adanya karakteristik data yang tumpang tindih cukup membatasi capaian maksimal dari nilai *F1-Score*.

ABSTRACT

Permei, Awang Siregar Lenggah. 2026. **Implementation of Extreme Gradient Boosting (XGBoost) with Recursive Feature Elimination (RFE)** for Heart Attack Classification. Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Advisors: (I) Okta Qomaruddin Aziz, S.Si., M.Kom (II) Dr. Irwan Budi Santoso, M.Kom.

Keywords: XGBoost, Heart Attack, Recursive Feature Elimination (RFE), Random Undersampling, Classification.

This research is motivated by the high global mortality rate caused by cardiovascular diseases, particularly heart attacks, which underscores the urgent need for an accurate early detection system. However, the development of classification models is frequently hampered by imbalanced data and overlapping clinical features among patients. Therefore, this study aims to optimize the performance of the Extreme Gradient Boosting (XGBoost) algorithm in predicting heart disease risk. The study utilizes a medical dataset from Kaggle, comprising 4,238 observations with 15 input variables. Comprehensive testing was conducted through nine main experimental scenarios, encompassing twelve test combinations by integrating data balancing methods (Random Undersampling and SMOTE) and feature selection techniques (Recursive Feature Elimination and Thresholding). The results indicated that the implementation of Random Undersampling significantly increased the model's sensitivity compared to the baseline method. In the baseline scenario (Pure+Full), the model achieved an accuracy of 83.88%, but its recall was considerably low at only 14.29%, proving that the model remained highly biased toward the majority class. In contrast, the RUS+Threshold (0.07) scenario successfully addressed this limitation by achieving the highest recall of 75.00% and an ROC-AUC score of 71.58%. Furthermore, based on the feature importance analysis, gender, age, and glucose levels were identified as the most prominent clinical indicators. Overall, these findings demonstrate that the undersampling strategy is highly effective in enhancing the model's objectivity for the early detection of heart disease risk, although overlapping data characteristics still limit the maximum achievable F1-Score.

الملخص

بيرمي، أوانغ سيريجار لينغاه. 2026. تطبيق خوارزمية تعزيز التدرج المتطرف (XGBoost) مع حذف الميزات المتكرر (RFE) لتصنيف أمراض النوبات القلبية. أطروحة. برنامج دراسة هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانغ. المشرفان: (1) أوكتا قمر الدين عزيز، بكالوريوس علوم، ماجستير علوم حاسوب (2) د. إروان بودي سانتوسو، ماجستير علوم حاسوب.

الكلمات المفتاحية: XGBoost، مرض النوبات القلبية، إزالة الميزة العودية (RFE)، الاختزال العشوائي، التصنيف.

ينبع اختيار موضوع هذه الأطروحة من الارتفاع الكبير في معدلات الوفيات العالمية الناجمة عن أمراض القلب والأوعية الدموية، ولا سيما النوبات القلبية، مما يبرز الحاجة الماسة إلى نظام دقيق للكشف المبكر. ومع ذلك، غالباً ما يعوق تطوير نماذج التصنيف تحديات تتمثل في عدم توازن البيانات وتداخل الخصائص السريرية للمرضى. بناءً على ذلك، تهدف هذه الدراسة إلى تحسين أداء خوارزمية التعزيز المتدرج الفائق (XGBoost) في التنبؤ بمخاطر الإصابة بأمراض القلب. وتعتمد الدراسة على مجموعة بيانات طبية من منصة Kaggle، تضم 4238 حالة مراقبة بـ 15 متغيراً مدخلاً. أُجريت التقييمات بشكل شامل عبر تسعة سيناريوهات تجريبية رئيسية، تضمنت اثنتي عشرة تركيبة اختبار تدمج بين طرق موازنة البيانات (مثل أخذ العينات العشوائية الناقصة وتقنية SMOTE) وتقنيات اختيار الميزات (مثل الحذف المتكرر للميزات وتطبيق العتبة). وأظهرت النتائج أن تطبيق أخذ العينات العشوائية الناقصة ساهم في تعزيز حساسية النموذج بشكل ملحوظ مقارنة بالنهج الأساسي. ففي السيناريو الأساسي (Pure+Full)، حقق النموذج دقة بلغت 83.88%، غير أن قيمة الاستدعاء كانت متدنية للغاية عند 14.29% فقط، مما يؤكد الانحياز الشديد للنموذج نحو الفئة الأغلبية. وعلى النقيض من ذلك، نجح سيناريو RUS+Threshold بنسبة 0.07 في تدارك هذا القصور، محققاً أعلى قيمة استدعاء بنسبة 75.00% ودرجة ROC-AUC بلغت 71.58%. علاوة على ذلك، واستناداً إلى تحليل أهمية الميزات، تبين أن الجنس والعمر ومستويات الجلوكوز هي المؤشرات السريرية الأكثر تأثيراً. وفي المجمل، تبرهن هذه النتائج على الفعالية العالية لاستراتيجية أخذ العينات الناقصة في إضفاء المزيد من الموضوعية على النموذج عند الكشف المبكر عن مخاطر أمراض القلب، رغم أن تداخل البيانات لا يزال يشكل قيداً يحد من بلوغ الحد الأقصى لقيمة F1-Score.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Jantung adalah organ penting dalam tubuh manusia yang memiliki peran utama dalam sistem peredaran darah. Menjaga kesehatan jantung sangatlah krusial karena perannya yang besar dalam menunjang keberlangsungan hidup (Nafisah et al., 2024). Namun, gaya hidup masyarakat modern saat ini menjadi pemicu utama munculnya berbagai penyakit serius, seperti hipertensi, diabetes, dan serangan jantung. Pola makan yang buruk serta rendahnya aktivitas fisik sehari-hari turut memperparah risiko gangguan pada organ tersebut (Purnomo et al., 2020). Risiko yang dihasilkan dari gaya hidup tersebut salah satunya adalah serangan jantung. Serangan jantung merupakan kondisi saat pembuluh darah yang menuju ke jantung tersumbat atau menyempit secara mendadak, sehingga aliran darah menjadi terhambat dan merusak otot jantung (Darmawan & Dianta, 2023).

Pada tahun 2022, *World Health Organization* (WHO) memperkirakan bahwa sekitar 19,8 juta kematian disebabkan oleh penyakit kardiovaskular yang setara dengan kurang lebih 32% dari keseluruhan angka kemtian di seluruh dunia. Berdasarkan data tersebut, sekitar 85% kematian atau kurang lebih 16,8 juta jiwa terjadi akibat serangan jantung dan stroke (World Health Organization, 2025). Di Indonesia, hasil Riset Kesehatan Dasar (Riskesdas) tahun 2018 menunjukkan tren serupa, di mana dari setiap 1.000 penduduk terdapat 15 orang yang menderita penyakit jantung, atau sekitar 2,78 juta jiwa secara nasional. Berdasarkan hasil

diagnosis dokter, presentasi prevalensi penyakit jantung mengalami peningkatan yaitu dari 0,5% pada tahun 2013 menjadi 1,5% pada tahun 2018 (Sari & Ikbal, 2024).

Faktor risiko penyakit serangan jantung sangat beragam. Beberapa faktor yang menjadi penyebab utama dalam peningkatan risiko serangan jantung diantaranya adalah usia, gaya hidup, riwayat kesehatan keuarga, serta kebiasaan merokok (Dhany et al., 2024). Nafisah et al. (2024) dalam penelitiannya menyatakan bahwa zat tembakau dalam rokok memiliki efek trombotik yang menyebabkan darah lebih mudah menggumpal, sehingga berkontribusi terhadap terbentuknya aterosklerosis yang menjadi penyebab meningkatnya risiko serangan jantung. Berdasarkan atas kondisi tersebut, penerapan gaya hidup sehat serta deteksi dini faktor risiko menjadi langkah pencegahan yang krusial untuk dilakukan secara menyeluruh.

Dari sudut pandang Islam, menjaga kesehatan merupakan amanah yang harus dijaga oleh setiap individu. Rasulullah ﷺ bersabda:

إِنَّ لِجَسَدِكَ عَلَيْكَ حَقًّا

“Sesungguhnya tubuhmu memiliki hak atasmu." (HR. Bukhari, no. 5199).

Al-Qur’an juga menegaskan pentingnya menghindari hal-hal yang membahayakan diri, sebagaimana firman Allah ﷻ:

وَلَا تُلْقُوا بِأَيْدِيكُمْ إِلَى التَّهْلُكَةِ ۛ وَأَحْسِنُوا ۛ إِنَّ اللَّهَ يُحِبُّ الْمُحْسِنِينَ

“Dan janganlah kamu menjatuhkan dirimu sendiri ke dalam kebinasaan. Dan berbuat baiklah, sesungguhnya Allah menyukai orang-orang yang berbuat baik." (QS. Al-Baqarah: 195).

Ayat dan hadits ini menjadi landasan bahwa upaya menjaga kesehatan, termasuk deteksi dini terhadap serangan jantung, serta pelaksanaan tanggung jawab seorang hamba dalam memelihara kesehatan tubuh yang diberikan oleh Allah. Oleh karena itu, pemanfaatan teknologi dalam pencegahan dan deteksi dini kondisi ini bukan hanya relevan pada pandangan ilmiah saja, namun juga sejalan pada nilai-nilai syariat Islam.

Pendeteksian dini faktor risiko serangan jantung sangat krusial untuk mencegah komplikasi serius seperti gagal jantung, stroke, maupun gangguan ginjal. Serangan jantung sendiri termasuk dalam keadaan medis yang darurat dan diperlukan penanganan yang cepat dan tepat agar kerusakan pada jantung tidak semakin meluas (Aniamarta et al., 2022). Diagnosis dan intervensi dini secara signifikan menurunkan risiko kematian mendadak. Namun demikian, proses identifikasi risiko saat ini sebagian besar masih mengandalkan prosedur manual melalui wawancara medis dan observasi klinis konvensional. Metode ini memiliki keterbatasan dalam hal objektivitas dan kecepatan, serta rentan terhadap bias subjektif yang dapat mengakibatkan keterlambatan penanganan. Di sisi lain, meskipun upaya edukasi kesehatan terus ditingkatkan, kompleksitas data medis yang besar menuntut adanya instrumen analisis yang lebih presisi dan otomatis guna mengoptimalkan hasil deteksi dini secara akurat.

Tingginya angka kejadian serta urgensi deteksi dini penyakit jantung menegaskan pentingnya integrasi kemajuan teknologi untuk mendukung tenaga medis dalam melakukan diagnosis secara lebih presisi dan efektif (Sausan et al.,

2024). Kemajuan teknologi dalam dunia medis salah satunya adalah pemanfaatan algoritma pembelajaran mesin (*machine learning*), khususnya XGBoost (*Extreme Gradient Boosting*). Algoritma ini telah terbukti memberikan hasil yang menjanjikan dalam mengklasifikasikan data kesehatan secara efisien, dengan kecepatan pemrosesan dan performa prediksi yang lebih unggul dibandingkan teknik pembelajaran mesin konvensional (Shrivastava et al., 2024). Algoritma ini dirancang untuk membangun pohon secara bertahap dengan cara yang lebih efisien dan dapat dijalankan secara paralel, sehingga prosesnya lebih cepat dan optimal (Liu et al., 2021). Secara prinsip, sekumpulan pohon keputusan dibangun melalui sistem kerja XGBoost secara bertahap. Prediksi yang salah dari pohon sebelumnya akan diperbaiki melalui pembentukan pohon baru. Pendekatan bertahap ini memungkinkan model untuk terus meningkatkan akurasi, sekaligus menjaga efisiensi komputasi melalui proses paralelisasi dan pengaturan bobot fitur secara adaptif.

Dalam penelitian ini, algoritma XGBoost dipadukan dengan metode *Recursive Feature Elimination* (RFE). RFE diperuntukkan sebagai penyeleksi variabel-variabel paling relevan dari suatu dataset. Secara sederhana, RFE akan mendeteksi fitur dengan pengaruh terbesar pada variabel target yang akan diprediksi (Herza et al., 2024). Fitur terpenting yang akan dipertahankan menjadi tujuan pendekatan ini agar model menjadi lebih ringkas, efisien serta tidak mudah mengalami *overfitting*.

Penelitian ini diharapkan dapat memberikan kontribusi nyata dalam pengembangan sistem deteksi risiko serangan jantung berbasis *machine learning*

yang lebih akurat, efisien, serta mudah diterapkan pada layanan kesehatan. Dengan memanfaatkan kombinasi XGBoost dan RFE, model yang dihasilkan tidak hanya mampu menangani kompleksitas data medis dan ketidakseimbangan kelas, tetapi juga dapat mendukung tenaga medis dalam menentukan keputusan secara objektif dan efisien. Selain itu, penerapan pendekatan ini sejalan dengan nilai-nilai Islam yang menekankan pentingnya menjaga kesehatan sebagai bentuk tanggung jawab atas amanah jasmani yang diberikan oleh Allah SWT. Penelitian ini diharapkan dapat menjadi dasar pengembangan teknologi prediksi yang lebih efektif dalam upaya pencegahan penyakit jantung, sehingga dapat menurunkan risiko kematian serta meningkatkan kualitas hidup masyarakat.

1.2 Rumusan Masalah

Bagaimana performa kombinasi metode *Extreme Gradient Boosting* (XGBoost) dengan *Recursive Feature Elimination* (RFE) dalam mengklasifikasi populasi yang berisiko serangan jantung?

1.3 Batasan Masalah

a. Dataset yang digunakan adalah dataset "Heart Attack" dari Kaggle dengan jumlah 4.238 baris data dan memiliki 16 atribut, yaitu: *male*, *age*, *education*, *currentSmoker*, *cigsPerDay*, *BPMeds*, *prevalentStroke*, *prevalentHyp*, *diabetes*, *totChol*, *sysBP*, *diaBP*, *BMI*, *heartRate*, *glucose*, dan *TenYearCHD*.
b. Menggunakan bahasa pemrograman Python.
c. Evaluasi performa model menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, dan AUC-ROC.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membangun serta mengevaluasi model prediksi penyakit serangan jantung melalui kombinasi algoritma XGBoost dengan metode RFE, serta melakukan akurasi dan efektivitas model yang berhasil diperoleh dalam deteksi dini risiko serangan jantung.

1.5 Manfaat Penelitian

a. Memberikan kontribusi dalam pengembangan sistem deteksi dini risiko serangan jantung yang memiliki tingkat akurasi dan efisiensi yang lebih baik.
b. Memberikan wawasan tentang implementasi kombinasi metode XGBoost dan RFE dalam bidang Kesehatan.
c. Penelitian ini dapat dijadikan dasar dalam pengembangan sistem pendukung keputusan di bidang kesehatan, sekaligus menjadi acuan untuk penelitian lanjutan pada bidang prediksi penyakit serangan jantung.

BAB II

STUDI PUSTAKA

2.1 Klasifikasi Penyakit Serangan Jantung

Serangan jantung merupakan gangguan kesehatan yang terjadi ketika aliran darah pembawa oksigen menuju otot jantung tersumbat, sehingga menyebabkan kekurangan oksigen dan berujung pada kerusakan jaringan otot jantung (Khalisatifa et al., 2024). Untuk membantu mengenali atau memprediksi kondisi tersebut, dapat digunakan metode klasifikasi yang digunakan untuk mengelompokkan objek yang memiliki karakteristik serupa ke dalam beberapa kelas (Azhari et al., 2021). Klasifikasi dalam data *mining* bertujuan mengelompokkan data ke dalam kategori yang sudah ditentukan melalui analisis pada pola data yang sudah ada (Rahayu et al., 2023). Prinsip dasarnya adalah melakukan prediksi terhadap label kelas yang bersifat kategorikal berdasarkan data yang tersedia, sehingga setiap data dapat diklasifikasikan ke dalam kelas yang sudah ditentukan sebelumnya (Barata et al., 2023).

Penelitian oleh Khalisatifa et al. (2024) membahas tentang klasifikasi penyakit serangan jantung sebagai upaya untuk mendeteksi dan mencegah penyakit jantung sejak dini. *Dataset* yang digunakan berjumlah 303 data pasien dengan berbagai atribut numerik dan nominal yang berkaitan dengan kondisi kesehatan jantung. Melalui proses klasifikasi, diperoleh hasil akurasi sebesar 83,98%, yang menunjukkan kemampuan model untuk mendeteksi individu yang berisiko tinggi terhadap serangan jantung.

Selanjutnya, penelitian oleh Pradana et al. (2022) menerangkan klasifikasi penyakit serangan jantung untuk mendukung upaya identifikasi kondisi jantung secara lebih akurat. Proses klasifikasi yang dilakukan pada peneliitian ini menerapkan metode *Support Vector Machine* (SVM) dan *Naive Bayes* guna menentukan model terbaik dalam mengenali pola penyakit jantung. Pengujian ini menghasilkan akurasi 87% yang membuktikan bahwa pendekatan klasifikasi berbasis *machine learning* efektif dalam menganalisis penyakit serangan jantung.

Penelitian oleh Laili et al. (2025) berfokus pada klasifikasi penyakit serangan jantung untuk membandingkan tingkat akurasi antara algoritma *Decision Tree* dan SVM. Algoritma dari *Decision Tree* memperoleh hasil akurasi 98,11%, akurasi tersebut melampaui performa SVM yang hanya memperoleh akurasi 92,80%. Temuan ini menegaskan bahwa penerapan metode klasifikasi dapat menjadi pendekatan efektif dalam menganalisis dan mengidentifikasi penyakit serangan jantung secara lebih akurat.

Berdasarkan berbagai penelitian tersebut, metode klasifikasi terbukti efektif dalam membantu proses deteksi dini penyakit jantung. Namun, untuk memperoleh hasil yang lebih optimal, dibutuhkan algoritma yang mampu menangani kompleksitas data medis secara lebih baik, seperti *Extreme Gradient Boosting* (XGBoost) yang dibahas pada subbab berikutnya.

2.2 Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost**)** merupakan salah satu algoritma *machine learning* yang umum digunakan dalam proses klasifikasi. Algoritma ini memiliki berbagai kelebihan dibandingkan metode klasifikasi lainnya, di antaranya

adalah kemampuannya dalam menangani *outlier* yang lebih baik, waktu pemrosesan yang relatif cepat, serta menunjukkan performa akurasi prediksi yang tinggi (Syukron et al., 2020). Beberapa penelitian telah membuktikan efektivitas XGBoost dalam klasifikasi penyakit serangan jantung dengan hasil yang sangat baik.

Pagrut et al. (2023) menerapkan algoritma XGBoost pada dataset UCI dengan 918 sampel pasien dan 12 atribut medis. Model XGBoost dioptimalkan dengan *hyperparameter* seperti *learning rate*, *max_depth*, dan jumlah *boosting rounds*. Akurasi sebesar 96% berhasil didapatkan oleh XGBoost dalam penelitian ini. Nilai tersebut tercatat lebih tinggi jika dibandingkan dengan performa yang ditunjukkan oleh KNN dan *Random Forest*. Studi ini membuktikan bahwa XGBoost efektif dalam mendeteksi pasien dengan risiko penyakit jantung, serta dapat membantu tenaga medis mendiagnosis secara lebih akurat dan efisien.

Yang & Guan (2022) mengimplementasikan XGBoost pada *Heart Disease Dataset* dengan 4232 sampel pasien yang memiliki 37 fitur klinis. Setelah proses seleksi fitur berbasis *information gain*, dataset direduksi menjadi 15 fitur utama yang paling relevan. Model dilatih dengan metode 5-*fold cross validation* untuk menjaga generalisasi. Optimasi *hyperparameter* dilakukan dengan mengatur *learning rate*, *max_depth*, dan jumlah *estimators*. Hasil penelitian mencapai akurasi sebesar 93,44%, *precision* 92,66%, *recall* 97,16%, dan *F1-score* 94,86%, dengan nilai AUC 92,44%. Dibandingkan dengan lima algoritma *baseline* (*Random Forest*, KNN, *Logistic Regression*, *Decision Tree*, dan *Naive Bayes*), XGBoost konsisten memberikan performa terbaik di semua metrik evaluasi. *Recall* sebesar 97,16%

sangat penting dalam konteks medis untuk menunjukkan kemampuan model dalam mendeteksi mayoritas pasien yang berisiko, meminimalkan *false negative* yang dapat berakibat fatal.

Mayang Pratiwi & Mufidah (2024) menerapkan XGBoost *Classifier* pada dataset UCI *Machine Learning Repository* dengan 920 data observasi dan 14 fitur untuk klasifikasi risiko penyakit jantung. Berdasarkan hasil penelitian, XGBoost menghasilkan akurasi sebesar 93%, melampaui kinerja *Decision Tree Classifier* dengan akurasi 90%. Dapat disimpulkan pada hasil penelitian ini XGBoost lebih efektif dalam mendukung klasifikasi dan diagnosis dini penyakit jantung secara tepat dan efisien.

Dullah et al. (2025) melakukan penelitian untuk menciptakan model algoritma machine learning yang dapat mengkategorikan penyakit jantung dengan akurasi tinggi. Penelitian ini berfokus dalam mengatasi keterbatasan akurasi dalam proses diagnosis dengan mengevaluasi model yang mampu memberikan performa terbaik dalam mengolah serta menganalisis data kesehatan. Model XGBoost berhasil melampaui akurasi metode terdahulu setelah mencatatkan nilai tertinggi sebesar 99% dalam penelitian ini. Tingginya tingkat akurasi ini menunjukkan XGBoost sebagai algoritma yang efektif untuk meningkatkan akurasi deteksi dan klasifikasi penyakit jantung di masa mendatang.

Hakim et al (2025) menerapkan XGBoost dalam klasifikasi penyakit Diabetes Mellitus menggunakan dataset Diabetes *Health Indicators* dari CDC dengan 253.680 sampel dan 21 fitur. Hasilnya XGBoost menunjukkan performa hingga mencapai akurasi sebesar 90,78%. Meskipun dalam studi ini metode

Stacking Ensemble memberikan akurasi tertinggi sebesar 91,79% dan LightGBM mencapai 91,29%, XGBoost tetap menunjukkan performa yang kompetitif dalam menangani dataset berskala besar.

2.3 Keunggulan XGBoost untuk Klasifikasi Medis

Keunggulan utama XGBoost yang membuatnya efektif dalam klasifikasi penyakit serangan jantung meliputi:

1. **Regularisasi**

   XGBoost menerapkan teknik regularisasi (L1 dan L2) yang berfungsi mencegah *overfitting* dan kemampuan generalisasi model meningkat (Al Aziz & Santoso, 2025). sehingga metode ini sebagai pengembangan lebih lanjut dari *gradient boosting*, memiliki menawarkan efisiensi waktu pelatihan dan kemampuan regularisasi yang lebih baik (Sah et al., 2025).

2. **Penanganan *Missing Values***

   Dalam tugas-tugas imputasi data yang hilang, XGBoost memanfaatkan *ensemble* beberapa pohon untuk memprediksi nilai yang hilang secara tepat, mengatasi keterbatasan metode tradisional saat menangani data kompleks (Jinbo et al., 2025). Hal ini yang diterapkan pada penelitian Yang & Guan (2022), penelitian tersebut menerapkan strategi penanganan *missing values* dengan menghapus fitur yang memiliki lebih dari 70% data hilang dan melakukan imputasi dengan mean untuk fitur lainnya sebelum training model XGBoost, sehingga menghasilkan dataset yang lebih berkualitas.

3. ***Parallel Processing***

XGBoost mendukung pemrosesan paralel, yang memungkinkan pelatihan efisien pada kumpulan data besar dalam waktu yang relatif singkat (Putra et al., 2025). Selain itu Sistem ini berjalan lebih dari sepuluh kali lebih cepat daripada solusi populer yang ada pada satu mesin dan dapat diskalakan hingga miliaran contoh dalam pengaturan terdistribusi atau dengan memori terbatas. Komputasi paralel dan terdistribusi mempercepat pembelajaran yang memungkinkan eksplorasi model lebih cepat (Chen & Guestrin, 2016).

4. ***Cross-Validation***

XGBoost memiliki *built-in cross-validation* yang memudahkan dalam evaluasi dan tuning model. Yang & Guan (2022) mengimplementasikan 5-*fold cross validation* sebagai penjaga generalisasi model dan memastikan performa yang konsisten, yang menghasilkan model yang memiliki tingkat keandalan tinggi dalam klasifikasi penyakit jantung.

2.3.1 *Preprocessing* dan Optimasi Model

Tahapan *preprocessing* memiliki peran penting dalam memaksimalkan performa algoritma XGBoost. Berbagai penelitian sebelumnya menunjukkan bahwa tahap ini berpengaruh signifikan terhadap hasil akhir model.

Yang & Guan (2022) melakukan *preprocessing* yang komprehensif, meliputi penanganan *missing values*, normalisasi data menggunakan metode *min-max*, serta penyeimbangan data dengan algoritma SMOTE-ENN. Penelitian tersebut menghadapi distribusi data yang sangat tidak seimbang (rasio 9:1), dan

setelah proses *preprocessing*, dataset yang awalnya terdiri dari 4.232 sampel dengan 37 fitur direduksi menjadi 3.527 sampel dengan 15 fitur utama. Penerapan SMOTE-ENN terbukti efektif dalam meningkatkan nilai *recall* hingga 97,16%, yang menunjukkan kemampuan model dalam mendeteksi hampir seluruh pasien berisiko tinggi.

Pagrut et al. (2023) juga menerapkan tahapan *preprocessing* dengan menyiapkan variabel medis yang relevan, serta melakukan pembagian dataset menjadi data latih dan data uji sebelum melatih model XGBoost dengan optimasi *hyperparameter*. Sementara itu, Sausan et al. (2024) melakukan *encoding* untuk mengubah variabel kategorikal serta menerapkan teknik *oversampling* sebelum data dipisahkan menjadi data latih dan data uji, yang membantu XGBoost memperoleh akurasi hingga 93%.

Sejalan dengan penelitian tersebut, Hakim et al. (2025) juga melakukan normalisasi data dan penyeimbangan kelas menggunakan SMOTE pada *dataset Diabetes Mellitus*, dengan pembagian data latih sebesar 80% dan data uji sebesar 20%. Pendekatan ini memungkinkan evaluasi performa XGBoost dilakukan secara objektif dan konsisten terhadap data yang seimbang dan terstandarisasi.

2.3.2 Fungsi Objektif XGBoost

Fungsi objektif XGBoost dapat dinyatakan dalam persamaan (2.1).

$$Obj(\theta) = L(\theta) + \Omega(\theta) \quad (2.\ 1)$$

Kombinasi antara *loss function* dan *regularization* term dalam persamaan (2.1) membuat model yang dibentuk XGBoost menghasilkan *data training* yang akurat serta mampu melakukan generalisasi secara efektif terhadap data yang belum pernah digunakan sebelumnya, seperti yang dibuktikan dalam berbagai penelitian prediksi penyakit jantung.

2.3.3 Integrasi XGBoost dengan *Feature Selection*

Salah satu kekuatan XGBoost adalah kemampuannya untuk diintegrasikan dengan teknik *feature selection* seperti RFE. XGBoost memiliki *built-in feature importance* yang dapat digunakan sebagai dasar untuk ranking fitur dalam proses RFE. Ketika XGBoost digunakan sebagai *base estimator* dalam RFE, algoritma akan secara iteratif mengeliminasi fitur yang memiliki *importance score* terendah, sehingga diperoleh subset fitur yang optimal guna memaksimalkan performa klasifikasi. Kombinasi XGBoost dan RFE telah menunjukkan efektivitas yang baik dalam melakukan peningkatan akurasi sekaligus efisiensi model untuk klasifikasi penyakit serangan jantung, seperti yang ditunjukkan oleh berbagai penelitian yang menggunakan seleksi fitur sebelum atau bersamaan dengan *training* model XGBoost.

2.4 *Recursive Feature Elimination* (RFE)

Metode yang dipilih sebagai pemilihan fitur adalah RFE. *Recursive Feature Elimination* (RFE) merupakan teknik yang secara bertahap menyeleksi fitur dan menghapus fitur-fitur yang dianggap kurang relevan atau memiliki kontribusi lemah. Dilakukan proses ini secara berulang kali sampai kumpulan fitur yang paling

informatif untuk digunakan dalam pembangunan model (Pratama et al., 2022). Tujuan RFE adalah melakukan pemilihan subset untuk fitur paling relevan dan memiliki daya prediksi tertinggi, sehingga dimensionalitasnya pada dataset berkurang dan berpotensi meningkatkan kinerja model. RFE telah berkembang menjadi salah satu pendekatan yang banyak diadopsi untuk seleksi fitur dalam beragam konteks pembelajaran mesin, termasuk pada tugas klasifikasi maupun prediksi (Priyatno & Widiyaningtyas, 2024). RFE sangat efektif untuk mengurangi dimensionalitas data dan meningkatkan interpretabilitas model, terutama ketika dikombinasikan dengan algoritma klasifikasi yang kuat seperti XGBoost untuk klasifikasi penyakit serangan jantung.

2.4.1 Penerapan *Feature Selection* dalam Klasifikasi Penyakit Jantung

Yang & Guan (2022) menerapkan seleksi fitur berbasis *information gain* pada *Heart Disease Dataset* yang awalnya memiliki 4232 sampel dengan 37 fitur klinis. Proses seleksi fitur berhasil mereduksi dataset menjadi 3527 sampel dengan 15 fitur utama yang paling relevan untuk klasifikasi penyakit jantung. Reduksi fitur dari 37 menjadi 15 ini tidak menurunkan performa model XGBoost, bahkan justru meningkatkan efisiensi komputasi dan interpretabilitas model. Model XGBoost yang dilatih dengan 15 fitur terpilih berhasil mencapai akurasi 93,44%, *precision* 92,66%, *recall* 97,16%, *F1-score* 94,86%, dan AUC 92,44%. Hasil ini menunjukkan bahwa seleksi fitur yang tepat dapat mempertahankan atau bahkan memperbaiki peningkatan performa klasifikasi dengan mengurangi kompleksitas model.

Dalam penelitian Zhang et al. (2023) pendekatan RFE diperuntukkan sebagai penyaring relevan terhadap fitur yang berpengaruh untuk hasil klasifikasi penyakit. RFE bekerja dengan mengeliminasi atribut yang kurang signifikan secara bertahap hingga diperoleh kombinasi fitur terbaik. Penelitian tersebut juga menggabungkan RFE dengan algoritma *Extreme Gradient Boosting* (XGBoost), menghasilkan model klasifikasi SMOTE-RFE-XGBoost dengan akurasi mencapai 97,56%. Kombinasi ini membuktikan bahwa RFE mampu meningkatkan kinerja XGBoost melalui pemilihan fitur optimal, sehingga pendekatan serupa berpotensi diterapkan dalam klasifikasi penyakit serangan jantung untuk memperoleh hasil prediksi yang lebih akurat dan efisien.

Menurut Anita et al. (2025), RFE efektif digunakan dalam hal mengoptimalkan akurasi klasifikasi penyakit jantung. RFE melakukan pemilihan fitur paling relevan agar model klasifikasi dapat dengan tepat mengidentifikasi kondisi pasien. Hasil penelitiannya didapatkan bahwa RFE mampu melakukan peningkatan dalam hal performa klasifikasi penyakit jantung, menjadikannya metode yang penting dalam pengembangan sistem pendukung keputusan medis.

2.4.2 Algoritma *Recursive Feature Elimination* (RFE)

Penelitian oleh Fathima et al. (2023) menjelaskan langkah dari algoritma RFE adakah sebagai berikut.

1. Menentukan nilai ambang batas (*threshold*) atau parameter evaluasi awal sesuai dengan kebutuhan analisis.
2. Menyatakan hipotesis utama yang meliputi hipotesis 0 serta hipotesis alternatif yang akan diuji untuk setiap fitur.

3. Memuat dataset Ds yang berisi seluruh fitur masukan, seperti jenis kelamin, usia, status merokok, tekanan darah, kolesterol, kadar glukosa, dan variabel medis lainnya**.**

4. Melatih model pembelajaran pada dataset lengkap untuk menghitung skor kepentingan (*importance score*) atau nilai statistik dari setiap fitur.

5. Menghapus fitur yang tidak signifikan atau memiliki skor terendah sesuai dengan kriteria seleksi yang telah ditetapkan.

6. Mengulangi langkah 4 dan 5 secara bertahap hingga diperoleh fitur-fitur yang signifikan atau relevan terhadap model.

Proses iteratif ini menyisakan hanya fitur yang memiliki kontribusi signifikan terhadap klasifikasi penyakit jantung yang dipertahankan dalam model final. Dalam klasifikasi penyakit jantung, fitur-fitur seperti usia, tekanan darah, kolesterol, kadar gula darah, dan tipe nyeri dada biasanya memiliki *importance score* yang tinggi karena hubungannya yang kuat dengan risiko kardiovaskular.

Ringkasan dari berbagai penelitian yang telah diulas ditunjukkan pada Tabel 2.1. Tabel tersebut merangkum perbandingan antarstudi berdasarkan peneliti, judul penelitian, metode yang digunakan, serta temuan utamanya. Selain itu, tabel juga dilengkapi dengan informasi mengenai kelebihan dan keterbatasan pada setiap penelitian untuk memperoleh pemahaman yang lebih jelas terkait kontribusi dan batasan dari setiap studi.

Tabel 2.1 Penelitian Tentang Klasifikasi Penyakit Serangan Jantung

No	Peneliti	Judul	Metode	Hasil	Kelebihan	Keterbatasan
1.	Khalisatifa et al. (2024)	Klasifikasi Risiko Penyakit Serangan Jantung	Algoritma C4.5 untuk klasifikasi penyakit jantung	Akurasi 83,98%.	C4.5 mencapai akurasi 8398% dan menemukan 5 faktor risiko	Data sedikit dan metode masih terbatas pada C4.5

No	Peneliti	Judul	Metode	Hasil	Kelebihan	Keterbatasan
		dengan Menggunakan Algoritma C4.5			utama serangan jantung.	tanpa optimasi lanjutan.
2.	Pradana et al. (2022)	Komparasi Metode *Support Vector Machine* dan *Naïve Bayes* dalam Klasifikasi Peluang Penyakit Serangan Jantung	*SUPPORT VECTOR MACHINE* DAN *NAÏVE BAYES*	Hasil akurasi SVM 87% dan *Naïve Bayes* 84%	SVM mencapai akurasi tertinggi 87% dalam klasifikasi risiko serangan jantung, mengungguli *Naïve Bayes*.	Tidak memberikan detail tentang bagaimana data diproses (seperti normalisasi atau *feature scaling*.
3.	Pagrut et al. (2023)	Heartcare: Heart Disease Detection using Machine Learning	XGBoost, *Random Forest Classifier*, *K-Nearest Neighbors*	Akurasi XGBoost mencapai 96%, lebih tinggi dibandingkan KNN dan *Random Forest*	Mencapai akurasi tertinggi (98%) menggunakan algoritma XGBoost (96%), mengalahkan model sebelumnya.	Tidak merinci metrik performa (*Precision, Recall*, *F1-score*) untuk setiap algoritma yang diuji.
4.	Laili et al. (2025)	Komparasi Algoritma *Decision Tree* dan *Support Vector Machine* (SVM) dalam Klasifikasi Serangan Jantung	*DECISION TREE* DAN *SUPPORT VECTOR MACHINE* (SVM)	*Decision Tree* (98.11%) memiliki akurasi 5.31% lebih tinggi daripada SVM (92.80%) dalam klasifikasi serangan jantung.	*Decision Tree* mencapai akurasi sangat tinggi menjadikannya model terbaik untuk klasifikasi serangan jantung dalam studi ini.	Hanya menggunakan satu rasio *splitting* data (60:40), yang mengurangi validitas pengujian dan berisiko *overfitting.*
5.	Yang & Guan et al. (2022)	A Heart Disease Prediction Model Based on Feature Optimization and Smote-Xgboost Algorithm	XGBoost	Tingkat akurasi yang dihasilkan cukup tinggi yaitu dengan akurasi: 93,44%	*Smote*-XGBoost dengan optimasi fitur memiliki performa terbaik pada semua metrik evaluasi (Akurasi, Presisi, *Recall*, *F1-score*, dan AUC).	Dataset tidak dapat diakses publik untuk replikasi, sehingga validitas dan generalisasi hasilnya sulit diverifikasi oleh peneliti lain.

No	Peneliti	Judul	Metode	Hasil	Kelebihan	Keterbatasan
6.	Pratiwi & Mufidah (2024)	Perbandingan Metode *Decision Tree Classifier* dan XGBoost *Classifier* Dalam Memprediksi Penyakit Jantung	*Decision Tree Classifier* dan XGBoost *Classifier*	XGBoost mencapai akurasi 93%, lebih tinggi dari *Decision Tree* 90%, sehingga lebih efektif mendeteksi risiko penyakit jantung	Memiliki metode yang lebih unggul didukung dengan kemampuanny a menangani *overfitting* dan *imbalance* fitur.	tidak mengevaluasi aspek komputasi seperti waktu pemrosesan dan penggunaan memori (sebagaimana disebutkan di kesimpulan).
7.	Dullah et al. (2025)	Extreme Gradient Boosting Model with SMOTE for Heart Disease Classification	XGBoost, *Random Forest*, dan *Decision Tree*	Model XGBoost menunjukkan performa terbaik dengan akurasi sebesar 99%,	Model XGBoost yang dioptimalkan dengan SMOTE mencapai akurasi luar biasa sebesar 99%, menunjukkan potensi diagnostik yang sangat tinggi untuk klasifikasi penyakit jantung.	Penelitian ini tidak menjelaskan bagaimana *hyperparamet er* XGBoost (seperti *learning rate* atau *max depth*) dioptimalkan
8.	Anita et al. (2025)	KLASIFIKASI FAKTOR RISIKO PENYAKIT JANTUNG MENGGUNAKAN MACHINE LEARNING	SVM, *Random Forest*, KNN, *Naive Bayes*, *Decision Tree*, *Logistic Regression* dengan seleksi fitur (RFE)	Model terbaik adalah SVM dengan akurasi 83,91% dan fitur tipe nyeri dada, jumlah pembuluh darah, serta jenis talasemia paling berkontribusi.	Penelitian ini unggul dalam evaluasi komprehensif dengan RFE dan perbandingan enam algoritma, menawarkan solusi yang lebih mendalam dari studi terdahulu.	Model ini belum menerapkan penanganan data tak seimbang dan tidak melakukan analisis interpretabilitas lanjutan (seperti SHAP/LIME)
9.	(Zhang et al., 2023)	Classification and prediction of spinal disease	SMOTE-RFE-XGBoost	Hasil akurasi mencapai 97,56%	Model SMOTE-RFE XGBoost unggul karena mengintegrasik	kelemahannya adalah kompleksitas tinggi yang menjadikanny

No	Peneliti	Judul	Metode	Hasil	Kelebihan	Keterbatasan
		based on the SMOTE-RFE XGBoost model			an seleksi fitur dan penanganan data tak seimbang dengan XGBoost untuk klasifikasi yang kuat	a sulit diinterpretasi dan rentan *overfitting*.

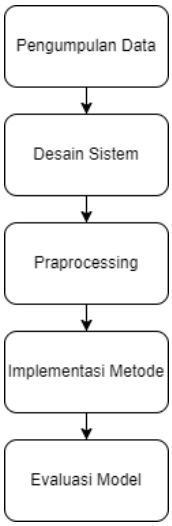
Berdasarkan Tabel 2.1, terlihat adanya perkembangan signifikan dalam penggunaan algoritma *Machine Learning* untuk klasifikasi penyakit jantung antara tahun 2022 hingga 2025, di mana algoritma XGBoost secara konsisten menunjukkan performa unggul dengan akurasi mencapai 93% hingga 99%, terutama bila dikombinasikan dengan teknik optimasi seperti SMOTE dan seleksi fitur RFE. Meskipun algoritma lain seperti SVM dan *Decision Tree* masih memberikan hasil yang kompetitif, sebagian besar studi masih memiliki keterbatasan pada aspek transparansi *preprocessing*, penggunaan dataset yang terbatas, serta kurangnya analisis mengenai efisiensi komputasi dan interpretabilitas model. Dapat disimpulkan, tren penelitian saat ini memperlihatkan integrasi metode penanganan data tidak seimbang dan pemilihan fitur kunci menjadi faktor penentu utama dalam meningkatkan akurasi diagnostik serangan jantung.

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Penelitian ini dilaksanakan melalui lima tahapan sistematis sebagaimana ditunjukkan pada Gambar 3.1 yang menggambarkan alur penelitian.



Gambar 3.1 Alur Penelitian

Pada Gambar 3.1, tahap pertama adalah pengumpulan data, yaitu proses memperoleh dataset risiko penyakit jantung dari Kaggle sebagai sumber data utama yang digunakan dalam analisis. Tahap kedua adalah perancangan sistem, yang mencakup penyusunan arsitektur alur penelitian serta analisis awal terhadap karakteristik data untuk memahami struktur, distribusi, dan potensi permasalahan yang terdapat dalam dataset. Selanjutnya, tahap ketiga adalah preprocessing data. Pada tahap ini dilakukan pembersihan data, penanganan data yang tidak seimbang, pemilihan fitur yang relevan, serta pembagian dataset menjadi data latih dan data

uji. Selanjutnya, tahap keempat yaitu implementasi metode yang mencakup proses pelatihan model klasifikasi dan optimasi parameter guna menghasilkan performa model yang optimal. Evaluasi menjadi tahapan terakhir sebagai tahapan untuk mengukur performa model menggunakan beragam metrik evaluasi sehingga dapat diketahui tingkat efektivitas model dalam memprediksi risiko penyakit jantung.

3.2 Pengumpulan Data

Pada penelitian ini mempergunakan dataset sekunder berjudul "*Heart Attack*" yang dipublikasikan oleh (Zaiem Mustaqiem, 2021) di *platform* Kaggle dengan lisensi akses terbuka. Dipilihnya dataset ini dikarenakan terdapatnya data medis yang lengkap dan relevan untuk mendukung deteksi dini risiko penyakit jantung. Dataset "*Heart Attack*" berisi data medis pasien yang digunakan untuk mengklasifikasikan individu yang berisiko mengalami serangan jantung dalam kurun waktu sepuluh tahun ke depan. Secara keseluruhan, dataset ini terdiri atas 4.238 baris data dan 16 atribut yang mencakup aspek demografis**,** riwayat kesehatan**,** gaya hidup**,** dan hasil pemeriksaan medis pasien.

Dari total 16 atribut tersebut, 15 atribut berfungsi sebagai fitur *input* dan 1 atribut berperan sebagai label target (variabel dependen)**,** yaitu *TenYearCHD***,** yang menunjukkan risiko seseorang mengalami penyakit jantung dalam sepuluh tahun ke depan. Nilai 1 menandakan individu berisiko**,** sedangkan nilai 0 menunjukkan individu tidak berisiko**.** Rincian lengkap mengenai masing-masing atribut disajikan pada Tabel 3.1.

Tabel 3.1 Detail Atribut Dataset

No.	Fitur	Keterangan
1.	*male*	Jenis kelamin responden (1 = laki-laki, 0 = perempuan)
2.	*age*	Usia responden dalam tahun

No.	Fitur	Keterangan
3.	*education*	Tingkat pendidikan responden (kode numerik, misalnya 1 = rendah, 4 = tinggi)
4.	*currentSmoker*	Status perokok saat ini (1 = ya, 0 = tidak)
5.	*cigsPerDay*	Jumlah batang rokok yang dihisap per hari
6.	*BPMeds*	Penggunaan obat tekanan darah (1 = ya, 0 = tidak)
7.	*prevalentStroke*	Riwayat stroke sebelumnya (1 = pernah, 0 = tidak)
8.	*prevalentHyp*	Kondisi hipertensi (1 = ya, 0 = tidak)
9.	*Diabetes*	Kondisi diabetes responden (1 = ya, 0 = tidak)
10.	*totChol*	Total kadar kolesterol dalam darah (mg/dL)
11.	*sysBP*	Tekanan darah sistolik (mmHg)
12.	*diaBP*	Tekanan darah diastolik (mmHg)
13.	*BMI*	Indeks Massa Tubuh (kg/m²)
14.	*heartRate*	Detak jantung per menit (bpm)
15.	*Glucose*	Kadar gula darah (mg/dL)
16.	*TenYearCHD*	Label target: kemungkinan terkena penyakit jantung dalam 10 tahun (1 = ya, 0 = tidak)

Berdasarkan rincian yang disajikan dalam Tabel 3.1, dataset ini terdiri dari 16 atribut yang mencerminkan profil kesehatan responden secara menyeluruh. Atribut *TenYearCHD* merupakan variabel dependen yang bersifat kategorikal *biner*, yang menentukan *output* klasifikasi apakah seorang pasien memiliki risiko terkena penyakit jantung dalam rentang waktu sepuluh tahun ke depan atau tidak. Sementara itu, 15 atribut lainnya bertindak sebagai prediktor yang memberikan informasi latar belakang dari berbagai sudut pandang klinis. Jika ditinjau lebih mendalam, variabel-variabel tersebut merepresentasikan keterkaitan antara faktor internal dan eksternal pasien. Atribut medis seperti *totChol, sysBP, diaBP,* dan *glucose* menyediakan data kuantitatif mengenai kondisi fisiologis pasien yang sering kali menjadi indikator utama dalam diagnosis medis. Di sisi lain, atribut seperti BMI dan *heartRate* memberikan gambaran mengenai kebugaran fisik dan kinerja jantung saat istirahat.

Selain faktor medis objektif, keterlibatan fitur perilaku seperti kebiasaan merokok (*currentSmoker* dan *cigsPerDay*) memberikan dimensi tambahan bagi

model untuk mempelajari bagaimana gaya hidup berkontribusi terhadap kesehatan jantung. Penggabungan antara faktor demografis (usia dan jenis kelamin), riwayat medis (stroke dan diabetes), serta hasil laboratorium ini menciptakan sebuah dataset yang komprehensif. Keanekaragaman tipe data dan satuan dalam tabel ini mulai dari data biner (0 dan 1) hingga data numerik dengan rentang nilai yang besar, menuntut adanya proses standarisasi pada tahap berikutnya agar model *machine learning* dapat memberikan bobot yang adil terhadap setiap atribut dalam melakukan prediksi.

Tabel 3.2 menampilkan 10 contoh data (*sample*) dari dataset penelitian yang digunakan untuk memprediksi risiko penyakit jantung dalam 10 tahun. Setiap baris mewakili satu responden, sedangkan kolom berisi informasi dasar, kondisi kesehatan, serta hasil targetnya (*TenYearCHD*).

Tabel 3.2 Sample 10 Dataset

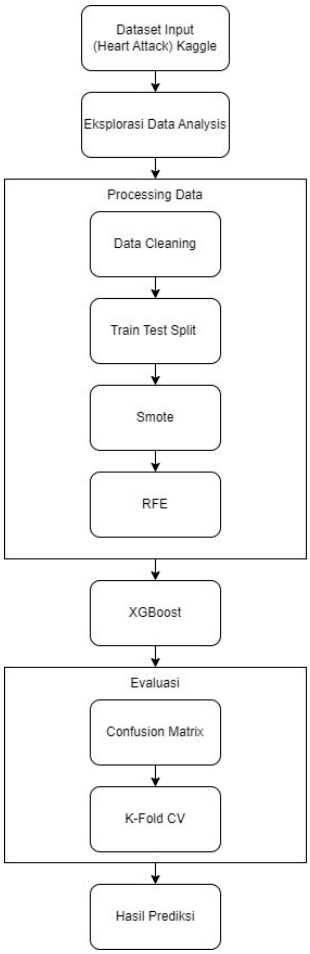
male	***age***	***education***	***currentSmoker***	***cigsPerDay***	***BPMeds***	***prevalentStrok***	***prevalentHyp***	***diabetes***	***totChol***	***sysBP***	***diaBP***	***BMI***	***heartRate***	***glucose***	***TenYearCHD***
1	61	NA	1	5	0	0	0	0	175	134	825	18.59	72	75	1
1	54	1	1	20	0	0	1	0	214	147	74	24.71	96	87	0
1	37	2	0	0	0	0	1	0	225	1245	925	38.53	95	83	0
1	56	NA	0	0	0	0	0	0	257	1535	102	28.09	72	75	0
1	52	1	0	0	0	0	1	1	178	160	98	40.11	75	225	0
0	42	1	1	1	0	0	1	0	233	153	101	28.93	60	90	0
1	36	3	0	0	0	0	0	0	180	111	73	27.78	71	80	0
0	43	2	1	10	0	0	0	0	243	1165	80	26.87	68	78	0
0	41	2	1	1	0	0	0	0	237	122	78	23.28	75	74	0
0	52	1	0	0	1	0	1	0	NA	148	92	25.09	70	NA	1

Tabel 3.2 di atas menyajikan sampel dari dataset sekunder yang diperoleh melalui platform Kaggle, yang mencakup informasi klinis dan perilaku responden

untuk prediksi risiko penyakit serangan jantung. Dataset ini terdiri dari berbagai atribut yang dikelompokkan ke dalam variabel demografi (usia, jenis kelamin, pendidikan), riwayat perilaku (status merokok, konsumsi rokok harian), dan parameter medis (tekanan darah, kadar kolesterol, BMI, denyut jantung, dan kadar glukosa). Setiap baris data merepresentasikan profil kesehatan individu yang diakhiri dengan variabel target *TenYearCHD* sebagai label biner. Representasi sampel ini menunjukkan kompleksitas fitur yang digunakan untuk melatih model klasifikasi dalam memprediksi risiko kesehatan jangka panjang.

3.3 Desain Sistem

Penelitian ini membangun model klasifikasi dengan menerapkan algoritma XGBoost serta RFE untuk memprediksi kemungkinan terjadinya penyakit serangan jantung dalam kurun waktu sepuluh tahun mendatang. Arsitektur sistem penelitian dirancang secara sistematis dan terdiri dari tujuh tahapan utama yang dijalankan secara sekuensial untuk menghasilkan model klasifikasi risiko penyakit jantung yang optimal. Untuk mencapai tujuan tersebut, penelitian ini menggunakan arsitektur sistem yang terdiri atas tujuh tahapan utama yang disusun secara sistematis dan dijalankan secara sekuensial sehingga menghasilkan model klasifikasi risiko penyakit jantung yang optimal. Arsitektur sistem penelitian secara keseluruhan ditunjukkan pada Gambar 3.2.



Gambar 3.2 Desain Sistem

Gambar 3.2 menunjukkan alur sistem yang digunakan dalam penelitian mulai dari data masuk sampai menghasilkan prediksi. Proses dimulai dari pengambilan dataset *Heart Attack* yang diperoleh dari Kaggle. Setelah data diperoleh, dilakukan eksplorasi awal untuk melihat kondisi data dan memahami pola-pola dasar yang ada. Selanjutnya, data memasuki tahap pemrosesan. Pada tahap ini dilakukan beberapa langkah, yaitu pembersihan data yang bertujuan mengatasi nilai yang hilang atau tidak sesuai dengan cara dihapus, pembagian data menjadi data latih dan data uji, penyeimbangan distribusi kelas menggunakan

SMOTE, serta pemilihan fitur penting dengan metode RFE. Tahapan ini dilakukan agar data siap dipergunakan sebagai pembagunan model prediksi.

Setelah data dipersiapkan, model dibangun menggunakan algoritma XGBoost. Model ini kemudian dievaluasi menggunakan *confusion matrix* dan metode *K-Fold Cross Validation* untuk memastikan performanya akurat dan konsisten. Pada akhir alur, sistem menghasilkan output berupa hasil prediksi terhadap data yang diuji. Secara keseluruhan, gambar ini menggambarkan urutan proses dari data awal sampai memperoleh prediksi terakhir melalui serangkaian tahap pengolahan dan evaluasi.

3.4 *Exploratory Data Analysis* (EDA)

Exploratory Data Analysis (EDA) adalah langkah krusial untuk memahami karakteristik dataset sebelum digunakan untuk membangun model. EDA bertujuan untuk mengidentifikasi struktur data, mendeteksi *missing values*, menganalisis distribusi data, serta memahami hubungan antar variabel yang dapat memengaruhi klasifikasi risiko penyakit jantung. Hasil EDA menjadi dasar untuk menentukan teknik preprocessing yang tepat. Tahap awal EDA dilakukan dengan memeriksa dimensi dataset menggunakan fungsi *shape* dan *info*() untuk mengetahui jumlah baris, kolom, serta tipe data setiap atribut. Penelitian ini menggunakan sebanyak 4.238 sampel data dengan 16 atribut yaitu 15 fitur dan 1 kelas, dengan kategori fitur biner dan numerik yang disajikan pada Tabel 3.3.

Tabel 3.3 Tabel Klasifikasi Jenis Fitur pada Dataset

Fitur Kategorik Biner	Fitur Numerik
male, currentSmoker, BPMeds, prevalentStroke, prevalentHyp, diabetes.	*age, education, cigsPerDay, totChol, sysBP, diaBP, BMI, heartRate, glucose*

Tabel 3.3 menampilkan pengelompokan fitur berdasarkan jenis datanya, yaitu fitur kategorik biner dan fitur numerik. Berdasarkan hasil identifikasi, terdapat 6 fitur kategorik biner yang direpresentasikan dengan nilai 0 dan 1, yaitu *male, currentSmoker, BPMeds, prevalentStroke, prevalentHyp,* dan diabetes. Sementara itu, 9 fitur lainnya merupakan fitur numerik seperti *age, education, cigsPerDay, totChol, sysBP, diaBP, BMI, heartRate, dan glucose*. Pemeriksaan ini dilakukan untuk memastikan bahwa dataset telah termuat dengan benar dan tidak terdapat inkonsistensi tipe data yang dapat menghambat proses *preprocessing* dan *modeling*. Pada penelitian ini dilakukan penghapusan *missing values. Missing values* adalah data yang tidak lengkap atau kosong (ditandai dengan NA/NaN). Pemeriksaan dilakukan menggunakan fungsi *isnull().sum()* untuk mendeteksi jumlah nilai kosong pada setiap atribut.

Tabel 3.4 menampilkan contoh baris data yang mengandung nilai NA, yaitu nilai yang hilang atau tidak tersedia pada dataset. Tabel ini menunjukkan bahwa ketidaklengkapan data tidak hanya muncul pada satu kolom saja, tetapi tersebar pada beberapa fitur penting. Setiap baris menggambarkan satu individu, dan titik-titik (…) di bagian atas dan bawah menandakan bahwa hanya sebagian baris yang ditampilkan sebagai sampel dari keseluruhan dataset.

Tabel 3.4 Data NA yang Tersebar pada Dataset

male	***age***	***education***	***currentSmoker***	***cigsPerDay***	***BPMeds***	***prevalentStroke***	***prevalentHyp***	***diabetes***	***totChol***	***sysBP***	***diaBP***	***BMI***	***heartRate***	***glucose***	***TenYearCHD***
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
0	39	2	1	9	0	0	0	0	226	114	64	22.35	85	NA	0
0	40	2	1	20	0	0	0	1	NA	114	65	21.19	61	NA	1

male	*age*	*education*	*currentSmoker*	*cigsPerDay*	*BPMeds*	*prevalentStrok*	*prevalentHyp*	*diabetes*	*totChol*	*sysBP*	*diaBP*	*BMI*	*heartRate*	*glucose*	*TenYearCHD*
0	45	NA	1	30	0	0	0	0	203	131	85	23.47	94	70	0
0	65	2	0	0	NA	0	1	0	270	165	98	21.66	62	92	1
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

Tabel 3.4 mendemonstrasikan kondisi ketidakteraturan data berupa nilai hilang yang terdapat pada fitur di dalam dataset. Notasi NA yang tersebar pada kolom *education*, *BPMeds*, *totChol*, hingga *glucose* menunjukkan bahwa fenomena data hilang ini bersifat acak dan tidak hanya terpusat pada satu variabel tertentu saja. Tanpa penanganan yang sesuai, keberadaan nilai yang tidak lengkap pada indikator klinis penting ini dapat memengaruhi performa algoritma klasifikasi. Jadi, representasi data pada tabel ini mendasari perlunya dilakukan tahap pembersihan data (*data cleaning*) dan strategi imputasi guna menjaga integritas informasi sebelum dataset digunakan untuk pelatihan model.

Hasil pemeriksaan menunjukkan bahwa terdapat *missing values* pada beberapa atribut *dataset*. *Missing values* ini akan ditangani pada tahap *preprocessing* menggunakan teknik *Data Cleaning* pada sub-bab berikutnya. Jumlah *missing value* terdapat pada beberapa fitur dalam dataset yang kemudian disajikan pada Tabel 3.5.

Tabel 3.5 Jumlah *Missing Value* pada Keseluruhan Dataset

Nama Fitur	**Jumlah *Missing Value***
education	105
cigsPerDay	29
BPMeds	53
totChol	50
BMI	19
heartRate	1

Nama Fitur	Jumlah *Missing Value*
glucose	388
Total	645

Tabel 3.5 menunjukkan jumlah nilai yang hilang (*missing value*) pada setiap fitur dalam keseluruhan *dataset*. Informasi ini penting untuk memahami tingkat kelengkapan data dan menentukan langkah yang tepat dalam proses *data cleaning*. Dari tabel terlihat bahwa beberapa fitur memiliki jumlah *missing value* yang cukup tinggi, sementara fitur lainnya hanya memiliki sedikit data yang tidak terisi. Fitur dengan jumlah *missing* terbesar adalah *glucose*, yaitu sebanyak 388 nilai hilang, yang menunjukkan bahwa hampir sebagian besar data glukosa tidak tercatat. Hal ini menjadikan fitur *glucose* salah satu fitur dengan masalah kelengkapan data yang paling serius dalam dataset.

Fitur lainnya yang juga memiliki jumlah missing value besar adalah *education* dengan 105 nilai hilang, serta *totChol* dan *BPMeds* yang masing-masing memiliki 50 dan 53 nilai hilang. Hilangnya nilai pada fitur-fitur ini dapat memengaruhi analisis karena ketiganya merupakan fitur yang cukup penting dalam memprofilkan kondisi pasien. Sementara itu, fitur seperti *cigsPerDay* (29 *missing*), serta BMI (19 *missing*) memiliki jumlah nilai hilang yang lebih sedikit namun tetap perlu diperhatikan dalam proses imputasi. Yang paling sedikit *missing*-nya adalah *heartRate*, hanya 1 data yang tidak terisi. Secara total terdapat 645 *missing value* pada seluruh dataset. Jumlah ini cukup besar sehingga penanganan *missing value* menjadi langkah penting pada tahap *preprocessing*. Keputusan yang diambil apakah melakukan imputasi, penghapusan sebagian baris, atau teknik lain akan sangat berpengaruh pada kualitas model yang akan dibangun.

3.5 *Preprocessing*

Preprocessing merupakan tahap penting dalam mempersiapkan data agar layak digunakan pada proses pembangunan model. Melalui tahap ini, data dipersiapkan agar memiliki kualitas yang memadai serta siap digunakan dalam proses pembelajaran algoritma *machine learning*. Sesuai dengan metodologi penelitian, *preprocessing* dalam penelitian ini meliputi tiga tahapan utama yang meliputi data *cleaning*, SMOTE, dan RFE. Pada Tabel 3.6 disajikan contoh *sample* dataset yang akan digunakan pada proses *processing.*

Tabel 3.6 Contoh *Sample* Dataset

No	*male*	*age*	*education*	*currentSmoker*	*cigsPerDay*	*BPMeds*	*prevalentStrok*	*prevalentHyp*	*diabetes*	*totChol*	*sysBP*	*diaBP*	*BMI*	*heartRate*	*glucose*	*TenYearCHD*
1	0	60	1	0	0	0	0	0	0	247	130	88	30.36	72	74	0
2	1	61	NA	1	5	0	0	0	0	175	134	825	18.59	72	75	1
3	1	36	4	1	35	0	0	0	0	295	102	68	28.15	60	63	0
4	0	52	1	0	0	1	0	1	0	NA	148	92	25.09	70	NA	1
5	1	37	2	0	0	0	0	1	0	225	1245	925	38.53	95	83	0
6	0	43	2	1	10	0	0	0	0	243	1165	80	26.87	68	78	0
7	0	59	1	0	0	0	0	1	0	209	150	85	20.77	90	88	1
8	1	56	NA	0	0	0	0	0	0	257	1535	102	28.09	72	75	0
9	1	52	1	0	0	0	0	1	1	178	160	98	40.11	75	225	0
10	0	42	1	1	1	0	0	1	0	233	153	101	28.93	60	90	0
11	0	41	2	1	1	0	0	0	0	237	122	78	23.28	75	74	0
12	1	54	2	0	0	0	0	0	0	195	132	835	26.21	75	100	0
13	0	40	2	0	0	0	0	0	0	205	100	60	NA	60	72	1

No	*male*	*age*	*education*	*currentSmoker*	*cigsPerDay*	*BPMeds*	*prevalentStrok*	*prevalentHyp*	*diabetes*	*totChol*	*sysBP*	*diaBP*	*BMI*	*heartRate*	*glucose*	*TenYearCHD*
14	0	63	2	1	3	0	0	1	0	267	1565	925	27.1	60	79	1
15	0	60	3	0	0	0	0	1	0	325	182	106	27.61	80	77	1

Tabel 3.6 berisi contoh data mentah yang digunakan pada penelitian ini. Setiap baris menunjukkan satu pasien, sedangkan setiap kolom berisi fitur atau informasi terkait kondisi kesehatan, seperti jenis kelamin, usia, status perokok, tekanan darah, kolesterol, BMI, denyut jantung, dan kadar glukosa. Kolom terakhir, *TenYearCHD*, merupakan label yang menunjukkan apakah pasien mengalami penyakit jantung dalam 10 tahun ke depan (1 = ya, 0 = tidak). Pada sampel ini juga terlihat adanya beberapa nilai hilang (*NA*) yang nantinya akan diperbaiki pada tahap pembersihan data.

3.5.1 *Data Cleaning*

Data cleaning dilakukan untuk menjamin kualitas data agar berada dalam kondisi yang optimal sebelum digunakan pada proses analisis dan pelatihan model. Tahapan penting dalam penelitian ini salah satunya adalah penanganan *missing values*, yaitu menangani data hilang serta tidak lengkap dalam dataset. *Missing values* umumnya ditandai dengan nilai seperti *NA*, *NaN*, *Null*, atau *None*. *Missing values* dapat memengaruhi proses pelatihan model dengan menimbulkan error, menghasilkan bias pada prediksi, menurunkan performa model, serta membuat hasil analisis kurang konsisten.

Penelitian ini menerapkan metode *Listwise Deletion*, yaitu penghapusan seluruh baris pada data kosong dengan fungsi *dropna*(). Metode ini dipilih karena proporsi *missing values* pada dataset relatif kecil (kurang dari 10% dari total data), sehingga proses penghapusan tidak mengurangi ukuran dataset secara signifikan. Metode ini juga dapat menghindari bias yang dapat timbul karena proses *imputation*, serta menjaga kualitas dan keaslian data agar tidak terdapat nilai yang bersifat buatan.

Hasil pengamatan menunjukkan bahwa 5 nilai kosong tersebut tidak tersebar secara acak, melainkan terkonsentrasi pada sejumlah baris yang sama-sama memiliki lebih dari satu *missing value*. Misalnya, satu baris data bisa saja memiliki nilai kosong pada fitur *totChol* dan *glucose*. Karena itu, meskipun terdapat 5 nilai yang hilang, total baris yang terdampak hanya sebanyak 4 baris yaitu pada bari nomor 2, 4, 8, dan 13. Jumlah baris data yang dihapus dihitung menggunakan persamaan (3.1).

$$N_{\text{hapus}} = N_{\text{awal}} - N_{\text{akhir}} \quad (3.\ 1)$$

Persamaan (3.1) digunakan untuk menghitung berapa banyak baris data yang hilang atau dibuang selama proses pembersihan dataset. Berdasarkan hasil proses *data cleaning* terhadap data yang memiliki nilai kosong (*missing values*), diperoleh sebanyak 4 baris data yang teridentifikasi mengandung nilai NA. Melalui substitusi nilai ke dalam rumus, jumlah data awal yang semula sebanyak 4.238 sampel berkurang menjadi 4.234 sampel data akhir. Hasil dari pembersihan data tersebut ditunjukkan secara detail pada Tabel 3.7.

Tabel 3.7 Contoh Data Setelah Dilakukan *Cleaning* pada Data NA

No	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStrok	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
1	0	60	1	0	0	0	0	0	0	247	130	88	30.36	72	74	0
3	1	36	4	1	35	0	0	0	0	295	102	68	28.15	60	63	0
5	1	37	2	0	0	0	0	1	0	225	1245	925	38.53	95	83	0
6	0	43	2	1	10	0	0	0	0	243	1165	80	26.87	68	78	0
7	0	59	1	0	0	0	0	1	0	209	150	85	20.77	90	88	1
9	1	52	1	0	0	0	0	1	1	178	160	98	40.11	75	225	0
10	0	42	1	1	1	0	0	1	0	233	153	101	28.93	60	90	0
11	0	41	2	1	1	0	0	0	0	237	122	78	23.28	75	74	0
12	1	54	2	0	0	0	0	0	0	195	132	835	26.21	75	100	0
14	0	63	2	1	3	0	0	1	0	267	1565	925	27.1	60	79	1
15	0	60	3	0	0	0	0	1	0	325	182	106	27.61	80	77	1

Tabel 3.7 menunjukkan contoh data yang telah dilakukan *data cleaning*, khususnya untuk menangani nilai hilang (*missing values* atau NA). Pada tahap ini, baris-baris data yang mengandung NA telah dihapus sehingga hanya tersisa *record* yang benar-benar lengkap.

3.5.2 SMOTE

Imbalanced class adalah kondisi ketidakseimbangan distribusi kelas dalam dataset, ketika jumlah data pada kelas mayoritas secara signifikan lebih banyak dibandingkan kelas minoritas, sehingga dapat menyebabkan berbagai kendala dalam proses pelatihan model klasifikasi, seperti dominasi prediksi pada kelas

mayoritas sehingga nilai *recall* yang diperoleh oleh kelas minoritas menjadi rendah, akurasi yang terlihat tinggi namun menyesatkan (*misleading*), serta kemampuan generalisasi yang buruk terhadap data dunia nyata.

Dataset *TenYearCHD* hasil *cleaning* memiliki total 11 sampe**l**, dengan distribusi kelas 0 (tidak mengalami CHD dalam 10 tahun) adalah 8 sampel dan kelas 1 (mengalami CHD dalam 10 tahun) yaitu 3 sampel. Ketimpangan jumlah sampel antar kelas berpotensi membuat model lebih condong pada kelas mayoritas, yang berdampak pada menurunnya performa prediksi terhadap kelas minoritas. Oleh karena itu, diterapkan metode SMOTE sebagai pendekatan untuk menyeimbangkan data.ss

Metode SMOTE menghasilkan data sintetis baru untuk kelas minoritas melalui proses interpolasi linear antara suatu sampel minoritas $x_i$dan salah satu tetangganya x_{zi}, sebagaimana dinyatakan pada Persamaan (3.2).

$$x_{\{new\}} = x_i + \lambda \times \left(x_{\{zi\}} - x_i\right) \quad (3.\ 2)$$

Persamaan (3.2) merupakan rumus yang biasanya digunakan dalam proses *oversampling*, khususnya pada metode SMOTE. Tujuan rumus ini adalah membuat data sintetis (data baru) berdasarkan sampel asli. λ (lambda) adalah nilai acak yang berada dalam rentang 0 hingga 1. Jika *random oversampling* hanya melakukan duplikasi terhadap data yang telah tersedia, SMOTE membentuk sampel baru secara sintetis sehingga variasi data menjadi lebih tinggi, memperluas representasi kelas minoritas, dan mengurangi risiko *overfitting*.

Proses penerapan SMOTE dilakukan melalui beberapa tahapan berikut:

1. Menentukan data sampel dari kelas minoritas secara acak

2. Menentukan *k-nearest neig*hbors dari sampel tersebut (umumnya k=5) menggunakan jarak *Euclidean* yang dihitung berdasarkan Persamaan (3.3):

$$d(x_i, x_j) = \sqrt{\left\{\Sigma_f \left(x_{\{i,f\}} - x_{\{j,f\}}\right)^2\right\}} \quad (3.\ 3)$$

Persamaan (3.3) merupakan rumus *Euclidean distance* yang digunakan untuk mengukur tingkat kemiripan atau jarak antara dua sampel data, yaitu data asli $x_i$dan data tetangganya x_j. Perhitungan dilakukan dengan menjumlahkan selisih kuadrat pada setiap fitur (f), kemudian diakarkan. Semakin kecil nilai jarak $d(x_i, x_j)$, maka semakin dekat hubungan antara kedua sampel data tersebut. Semakin kecil nilai jarak d(xi, xj) yang dihasilkan, semakin dekat hubungan ketetanggaan antara kedua data tersebut. Oleh karena itu, data tetangga tersebut lebih layak dipilih sebagai acuan dalam proses pembentukan sampel sintetis baru pada algoritma SMOTE.

3. Memilih salah satu *neighbor* secara acak.
4. Membuat sampel sintetis baru melalui interpolasi linear.

Untuk menjelaskan proses SMOTE, digunakan tiga sampel data yang berasal dari hasil *data cleaning* dan ditampilkan pada Tabel 3.8 dengan menunjukkan bahwa sampel tersebut mewakili kelas minoritas dari dataset dan digunakan untuk menggambarkan proses pembangkitan data sintetis oleh algoritma SMOTE.

Tabel 3.8 Perhitungan SMOTE Dilakukan Menggunakan Beberapa Baris Sampel Minoritas

No	*male*	*age*	*education*	*currentSmoker*	*cigsPerDay*	*BPMeds*	*prevalentStroke*	*prevalentHyp*	*diabetes*	*totChol*	*sysBP*	*diaBP*	*BMI*	*heartRate*	*glucose*	*TenYearCHD*
7	0	59	1	0	0	0	0	1	0	209	150	85	20.77	90	88	1
14	0	63	2	1	3	0	0	1	0	267	1565	925	27.1	60	79	1
15	0	60	3	0	0	0	0	1	0	325	182	106	27.61	80	77	1

Sampel data yang disajikan pada Tabel 3.8 berasal dari kelas minoritas (*TenYearCHD* = 1) dan telah melalui proses pembersihan data. Data tersebut digunakan untuk menjelaskan mekanisme kerja SMOTE dalam menghasilkan sampel sintetis baru melalui proses interpolasi dengan tetangga terdekatnya. Dengan menggunakan fitur-fitur pada tabel ini sebagai parameter masukan, pembentukan sampel sintetis yang merepresentasikan karakteristik kelas minoritas, SMOTE digunakan untuk menangani masalah ketidakseimbangan kelas dalam dataset, sehingga model klasifikasi yang dihasilkan tidak menjadi bias akibat dominasi kelas mayoritas.

Sebagai contoh, jika dipilih sampel minoritas No 7 [59,209] serta salah satu *neighbor*-nya, yaitu sampel No. 14 dengan nilai [63,267] untuk fitur [*age,totChol*], Dengan menetapkan $\lambda = 0.5$, perhitungan sampel sintetis dilakukan menggunakan persamaan (3.4):

$$x_{\{new\}} = x_i + \lambda \times \left(x_{\{zi\}} - x_i\right) \quad (3.\ 4)$$

Perhitungan setiap fitur dengan menggunakan persamaan (3.4) adalah sebagai berikut:

$$age_{\{new\}} = 59 + 0.5 \times (63 - 59) = 61$$

$$totChol_{\{new\}} = 209 + 0.5 \times (267 - 209) = 238$$

Sehingga diperoleh sampel sintetis baru dengan nilai [*age*, *totChol*] = [61, 238]. Proses interpolasi tersebut diterapkan pada seluruh fitur numerik dalam dataset. Parameter λ ditetapkan memiliki batas nilai yang berada pada rentang 0 dan 1. Nilai λ dipilih secara acak pada rentang tersebut untuk menentukan posisi titik sintetis di antara sampel asli dan *k-nearest neighbor*-nya. Setiap sampel minoritas dapat dipasangkan dengan lebih dari satu *neighbor*, sehingga nilai λ yang digunakan pada setiap pembentukan sampel sintetis bervariasi. Sebagai contoh tambahan, sampel No. 7 dipasangkan dengan *neighbor* No. 15 yang memiliki nilai [60, 325] dengan λ = 0,3. Perhitungannya menggunakan persamaan (3.4) adalah sebagai berikut:

$$age_{\{new\}} = 59 + 0.3 \times (60 - 59) = 59.3 = 59$$

$$totChol_{\{new\}} = 209 + 0.3 \times (325 - 209) = 249$$

Sehingga diperoleh sampel sintetis dengan nilai $[age, totChol] = [59{,}249]$. Nilai dilakukan pembulatan karena fitur *age* dan *totChol* merupakan data diskrit/bilangan bulat sehingga hasil interpolasi disesuaikan kembali ke format data awal. Perbedaan nilai λ (*lambda*) pada setiap proses pembentukan data sintetis disebabkan oleh sifat acak (*randomness*) dari algoritma SMOTE. Pada setiap iterasi, satu sampel data minoritas dapat dipasangkan dengan salah satu *k-nearest neighbor*-nya secara acak. Setelah itu, nilai λ dipilih secara acak dalam rentang 0 sampai 1 sebagai faktor yang menentukan posisi data sintetis antara kedua sampel.

Dengan demikian, meskipun berasal dari sampel dasar yang sama, setiap pasangan data minoritas dan tetangganya dapat menghasilkan nilai λ yang berbeda. Langkah ini dilakukan agar posisi sampel sintetis dapat tersebar secara proporsional

di antara data asli dan sampel tetangga terdekatnya, sehingga distribusi kelas minoritas menjadi lebih bervariasi dan tidak menumpuk pada satu titik tertentu.

Sebagaimana ditunjukkan pada Tabel 3.9, dua contoh proses interpolasi ditampilkan untuk sampel minoritas No. 7 dengan pasangan *neighbor* dan nilai λ yang berbeda.

Tabel 3.9 Dua Contoh Hasil Interpolasi dari Sampel Minoritas

Sampel Asli	*Neighbor*	λ	*age_new*	*totChol_new*
No 7 (59,209)	No 14 (63,267)	0.5	61	238
No 7 (59,209)	No 15 (60,325)	0.3	59	249

Pada Tabel 3.9 ditunjukkan proses pembentukan sampel sintetis menggunakan metode SMOTE melalui interpolasi antar sampel minoritas. Dalam proses ini, setiap sampel minoritas (No. 7, 14, dan 15) dipasangkan dengan salah satu tetangga terdekatnya yang juga berasal dari kelas minoritas. Selanjutnya, nilai λ ditentukan secara acak antara 0 dan 1 sebagai faktor yang mengatur posisi data sintetis pada garis yang menghubungkan antara sampel asli dan *neighbor*-nya. Semakin besar nilai λ, semakin dekat posisi sampel sintetis terhadap neighbor tersebut.

Seluruh fitur numerik dihitung menggunakan rumus interpolasi SMOTE, sedangkan fitur kategorikal dibulatkan ke nilai integer agar sesuai dengan tipe datanya. Variasi pasangan sampel dan nilai λ menghasilkan perbedaan nilai pada seluruh fitur numerik, yang terlihat jelas pada setiap baris dalam tabel. Hasil tersebut memperlihatkan bahwa SMOTE tidak hanya melakukan penambahan data secara kuantitatif, tetapi juga menyebarkan sampel sintetis secara lebih merata pada ruang fitur. Oleh karena itu, kelas minoritas memiliki sebaran data yang lebih

seimbang serta tidak terfokus pada satu titik tertentu. Hasil dari proses interpolasi SMOTE ini disajikan pada Tabel 3.10.

Tabel 3.10 Hasil Interpolasi SMOTE

Sampel Asli	***Neighbor***	**λ**	***age_new***	***totChol_new***	***sysBP_new***	***diaBP_new***	***BMI_new***	***heartRate_new***	***glucose_new***
No 7	No 14	0.5	59 + 0.5(63-59)=61	209 + 0.5(267-209)=238	150 + 0.5(1565 - 150)=857.5	85 + 0.5(92.5 - 85)=88.75	20.77 + 0.5(27.1 - 20.77)= 23.935= 23.94	90 + 0.5(60-90)=75	88 + 0.5(79-88)=83.5
No 7	No 15	0.3	59 + 0.3(60-59)=59.3=59	209 + 0.3(325-209)=248.8=249	150 + 0.3(182-150)=159.6=160	85 + 0.3(106 - 85)=91.3=91	20.77 + 0.3(27.61- 20.77)= 22.96≈23	90 + 0.3(80-90)=87	88 + 0.3(77-88)=84.9=85
No 14	No 7	0.5	63 + 0.5(59-63)=61	267 + 0.5(209-267)=238	1565 + 0.5(150-1565)=857.5	92.5 + 0.5(85-92.5)=88.75	27.1 + 0.5(20.77- 27.1)=23.935=23.94	60 + 0.5(90-60)=75	79 + 0.5(88-79)=83.5
No 14	No 15	0.4	63 + 0.4(60-63)=61.8=62	267 + 0.4(325-267)=291.2=291	1565 + 0.4(182-1565)=923.4	92.5 + 0.4(106 - 92.5)=98.9=99	27.1 + 0.4(27.61- 27.1)=27.36	60 + 0.4(80-60)=68	79 + 0.4(77-79)=78.2=78

Keterangan:
1. Sampel minoritas = No 7, 14, 15
2. *Neighbor* dipilih dari minoritas lainnya untuk interpolasi
3. λ dipilih acak antara 0 dan 1
4. Semua fitur numerik dihitung dengan rumus SMOTE
5. Fitur kategorikal dibulatkan ke integer

Tabel 3.10 menyajikan hasil simulasi perhitungan interpolasi SMOTE yang diterapkan pada sampel minoritas. Tabel ini merinci bagaimana data sintetis baru dibentuk dengan mengombinasikan sampel asli dan tetangga terdekatnya (*neighbor*) menggunakan parameter bobot acak (*lambda*) antara 0 dan 1. Melalui proses matematis ini, nilai-nilai baru pada fitur kontinu seperti *age*, *totChol*, hingga

glucose dihasilkan guna menambah jumlah observasi pada kelas minoritas tanpa menghilangkan karakteristik statistik data asli. Hasil dari perhitungan ini menunjukkan bahwa setiap fitur numerik mengalami penyesuaian nilai berdasarkan jarak antar titik data, yang secara kolektif berfungsi untuk memperkuat distribusi kelas minoritas sehingga diperoleh keseimbangan data yang lebih proporsional untuk tahap pemodelan klasifikasi. Contoh hasil SMOTE ditampilkan pada Tabel 3.11.

Tabel 3.11 Contoh Data Hasil SMOTE

No.	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStrok	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
1	0	60	1	0	0	0	0	0	0	247	130	88	30.36	72	74	0
2	1	36	4	1	35	0	0	0	0	295	102	68	28.15	60	63	0
3	1	37	2	0	0	0	0	1	0	225	1245	925	38.53	95	83	0
4	0	43	2	1	10	0	0	0	0	243	1165	80	26.87	68	78	0
5	1	52	1	0	0	0	0	1	1	178	160	98	40.11	75	225	0
6	0	42	1	1	1	0	0	1	0	233	153	101	28.93	60	90	0
7	0	41	2	1	1	0	0	0	0	237	122	78	23.28	75	74	0
8	1	54	2	0	0	0	0	0	0	195	132	835	26.21	75	100	0
9	0	59	1	0	0	0	0	1	0	209	150	85	20.77	90	88	1
10	0	63	2	1	3	0	0	1	0	267	1565	92.5	27.1	60	79	1
11	0	60	3	0	0	0	0	1	0	325	182	106	27.61	80	77	1
12	0	61	1.5	0.5	1.5	0	0	1	0	238	857.5	88.75	23.94	75	83.5	1
13	0	59	2	0	0	0	0	1	0	249	160	91	23	87	85	1
14	0	61	1.5	0.5	1.5	0	0	1	0	238	857.5	88.75	23.94	75	83.5	1
15	0	62	2.5	0.7	1.5	0	0	1	0	291	923.4	99	27.36	68	78	1

No.	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStrok	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
16	0	60	2	0	0	0	0	1	0	286	173	100	25	83	81	1

Hasil proses SMOTE pada kelas minoritas (*TenYearCHD* = 1) ditampilkan pada Tabel 3.11. Proses SMOTE menghasilkan sejumlah sampel sintetis baru dengan karakteristik yang menyerupai data asli, namun memiliki nilai fitur yang bervariasi sesuai dengan hasil interpolasi antara data minoritas dan *neighbor*-nya.

Sebagai contoh, pada tabel terlihat bahwa data dengan No. 12 dan No. 13 merupakan hasil sintesis dari sampel minoritas No. 7 yang dipasangkan dengan *neighbor* No. 14 dan No. 15. Nilai beberapa fitur numerik seperti *age*, *totChol*, *sysBP*, dan *diaBP* mengalami perubahan sesuai dengan nilai λ yang digunakan pada proses interpolasi.

Perubahan tersebut menunjukkan bahwa SMOTE tidak menyalin data secara langsung, melainkan menciptakan titik data baru di antara dua sampel minoritas yang berdekatan. Kondisi tersebut membuat distribusi kelas minoritas menjadi lebih seimbang, sehingga model pembelajaran mesin seperti XGBoost dapat mempelajari pola data dengan lebih optimal tanpa terpengaruh bias akibat ketidakseimbangan kelas.

Sebelum SMOTE diterapkan, dilakukan terlebih dahulu proses penghapusan data yang mengandung nilai hilang (*NaN*), karena perhitungan SMOTE bergantung pada kelengkapan data numerik dalam pengukuran jarak

Euclidean. Setelah itu, data dipisahkan menjadi *training set* dan *testing set* dengan rasio 80:20 guna mendukung proses pelatihan serta evaluasi model.

SMOTE hanya diterapkan pada data latih untuk mencegah terjadinya *data leakage* serta menjaga agar data uji tetap mencerminkan distribusi asli kelas. Implementasi SMOTE dilakukan menggunakan *library imbalanced-learn (imblearn)* dengan parameter:

1. *k_neighbors* = 5
2. *sampling_strategy='auto'* (menyeimbangkan kelas secara otomatis menjadi rasio 1:1)

Setelah distribusi kelas berhasil diseimbangkan, tahap selanjutnya adalah melakukan seleksi fitur menggunakan metode RFE untuk menentukan atribut yang paling berkontribusi terhadap prediksi risiko penyakit jantung. Fitur-fitur yang kurang relevan diabaikan agar model menjadi lebih efisien, akurat, dan terhindar dari risiko *overfitting*.

3.5.3 *Random Undersampling*

Random Undersampling merupakan teknik penyeimbangan data yang dilakukan dengan mengurangi jumlah sampel pada kelas mayoritas secara acak hingga jumlahnya setara dengan kelas minoritas. Metode ini digunakan untuk mengatasi ketidakseimbangan kelas yang dapat menyebabkan model lebih condong pada kelas mayoritas dalam proses prediksi.

Sebagai ilustrasi penerapan metode *Random Undersampling*, digunakan distribusi kelas pada dataset penelitian yang terdiri atas 4.238 data. Berdasarkan variabel target (*TenYearCHD*), kelas 0 (tidak berisiko penyakit jantung) berjumlah

3.596 data, sedangkan kelas 1 (berisiko penyakit jantung) berjumlah 642 data. Perbedaan jumlah data yang cukup besar menunjukkan bahwa dataset berada dalam kondisi tidak seimbang (*imbalanced dataset*), sehingga diperlukan proses *Random Undersampling*.

Pada metode ini, jumlah sampel dari kelas mayoritas dikurangi secara acak hingga menyamai jumlah sampel pada kelas minoritas. Dengan demikian, sebanyak 642 data pada kelas mayoritas dipertahankan sehingga kedua kelas memiliki jumlah yang sama, yaitu masing-masing 642 data. Setelah proses *undersampling* selesai, total data yang digunakan menjadi 1.284 sampel dengan distribusi kelas yang seimbang, yakni 50% kelas 0 dan 50% kelas 1. Selanjutnya, dataset hasil *undersampling* dibagi menjadi *training set* dan *testing set* dengan rasio 80:20. Dari total 1.284 sampel, sebanyak 1.028 data digunakan sebagai data latih, sedangkan 256 data lainnya digunakan sebagai data uji.

Distribusi kelas pada data latih tetap dijaga agar seimbang, yaitu masing-masing 514 sampel untuk kelas 0 dan kelas 1. Hal serupa juga berlaku pada data uji, dengan masing-masing 128 sampel per kelas. Keseimbangan ini dilakukan agar proses pelatihan dan pengujian model berlangsung secara adil, sehingga model dapat mempelajari pola dari setiap kelas secara proporsional serta dievaluasi secara objektif tanpa dipengaruhi ketimpangan distribusi kelas.

3.5.4 *Recursive Feature Elimination*

Setelah SMOTE diterapkan, langkah selanjutnya adalah melakukan seleksi fitur dengan metode RFE. Proses ini bertujuan untuk memilih fitur yang paling berpengaruh dalam memprediksi risiko penyakit jantung (*TenYearCHD*) sekaligus

mengeliminasi fitur yang tidak relevan, sehingga model menjadi lebih sederhana, efisien, dan lebih tahan terhadap *overfitting*.

RFE bertujuan untuk:

1. Menyeleksi fitur yang memiliki tingkat pengaruh tertinggi terhadap target (*TenYearCHD*).
2. Mengeliminasi fitur yang kurang relevan terhadap model
3. Mengoptimalkan efisiensi model
4. Mengurangi risiko *overfitting*

RFE sangat efektif jika digabungkan dengan algoritma berbasis *gradient boosting* seperti XGBoost, karena metode tersebut memiliki kemampuan alami untuk menghitung *feature importance* dari setiap variabel prediktor.

Berikut langkah penerapan RFE pada dataset hasil SMOTE yang terdiri dari 16 data dan 15 variabel prediktor:

1. Normalisasi dan Penentuan Target

   Variabel target adalah *TenYearCHD* (0 = tidak berisiko, 1 = berisiko). Semua fitur numerik (*age, sysBP, diaBP, totChol*, dsb.) dinormalisasi agar skala antar fitur seimbang.

2. Penghitungan Korelasi Awal

   Untuk mempercepat proses eliminasi, dilakukan perhitungan korelasi Pearson antara setiap fitur dan target dengan korelasi awal menggunakan persamaan (3.5).

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (3.\ 5)$$

Pada persamaan (3.5), r_{xy} merupakan koefisien korelasi Pearson, $x_i$dan $y_i$adalah nilai pengamatan ke-i, $\bar{x}$dan $\bar{y}$ masing-masing merupakan nilai rata-rata variabel x dan y, sedangkan $\sum$ menyatakan operasi penjumlahan seluruh data.

Untuk menjelaskan besarnya hubungan yang melekat dalam korelasi linier ketat antara variabel klinis dan kemungkinan mengembangkan penyakit jantung (sepuluh tahunCHD), penelitian ini menggunakan standardisasi tradisional koefisien korelasi seperti yang digambarkan oleh (Schober & Schwarte, 2018) pada tabel 3.12.

Tabel 3.12 Kategori Interpretasi Koefisien Korelasi Pearson

No	Nilai Absolut Koefisien Korelasi	Interpretasi
1	0.00 – 0.10	Korelasi sangat lemah (*Negligible correlation*)
2	0.10 – 0.39	Korelasi lemah (*Weak correlation*)
3	0.40 – 0.69	Korelasi sedang (*Moderate correlation*)
4	0.70 – 0.89	Korelasi kuat (*Strong correlation*)
5	0.90 – 1.00	Korelasi sangat kuat (*Very strong correlation*)
4	0.70 – 0.89	Korelasi kuat (*Strong correlation*)
5	0.90 – 1.00	Korelasi sangat kuat (*Very strong correlation*)

Pada Tabel 3.12, nilai korelasi pada rentang 0,00–0,10 menunjukkan bahwa hubungan antar variabel sangat kecil atau hampir tidak ada. Nilai 0,10–0,39 menggambarkan adanya hubungan antarvariabel, namun tingkat hubungannya masih rendah. Rentang 0,40–0,69 menunjukkan adanya hubungan yang cukup kuat, yang mengindikasikan bahwa variabel prediktor mulai memiliki keterkaitan yang lebih jelas terhadap variabel target.

Selanjutnya, contoh hasil pemetaan korelasi awal pada dataset sampel berdasarkan standar (Schober & Schwarte, 2018) disajikan pada Tabel 3.13.

Tabel 3.13 Contoh Perhitungan Dataset pada Tabel

Variabel	r(*TenYearCHD*)	Tingkat Korelasi
age	0,78	Korelasi kuat
prevalentHyp	0,67	Korelasi sedang
male	0,58	Korelasi sedang
BMI	0,54	Korelasi sedang
totChol	0,42	Korelasi sedang
diaBP	0,36	Korelasi lemah
cigsPerDay	0,29	Korelasi lemah
diabetes	0,26	Korelasi lemah
heartRate	0,23	Korelasi lemah
glucose	0,23	Korelasi lemah
sysBP	0,21	Korelasi lemah
currentSmoker	0,18	Korelasi lemah
education	0,04	Korelasi sangat lemah
BPMeds	0,00	Korelasi sangat lemah
prevalentStroke	0,00	Korelasi sangat lemah

Tabel 3.13 memperlihatkan hasil koefisien korelasi Pearson (r) yang digunakan untuk menilai hubungan antara atribut prediktor dengan variabel target *TenYearCHD*. Nilai korelasi yang digunakan pada analisis ini adalah nilai absolut (|r|), sehingga arah hubungan (positif atau negatif) tidak diperhatikan, melainkan hanyas kekuatan hubungannya saja.

Koefisien korelasi menunjukkan tingkat kekuatan hubungan linier antara setiap fitur dengan risiko penyakit jantung dalam periode sepuluh tahun. Semakin mendekati nilai 1, maka hubungan yang terbentuk semakin kuat, sedangkan nilai yang mendekati 0 menunjukkan hubungan yang semakin lemah.

Hasil perhitungan menunjukkan bahwa variabel *age* memiliki nilai korelasi sebesar 0,79 yang termasuk dalam kategori korelasi kuat. Variabel *prevalentHyp* (0,67), *male* (0,58), BMI (0,54), dan *totChol* (0,42) berada

pada kategori korelasi sedang, yang menunjukkan adanya hubungan yang cukup signifikan terhadap *TenYearCHD*.

Selanjutnya, variabel *diaBP* (0,36), *cigsPerDay* (0,29), diabetes (0,26), *heartRate* (0,23), *glucose* (0,23), *sysBP* (0,21), dan *currentSmoker* (0,18) termasuk dalam kategori korelasi lemah, yang menunjukkan bahwa hubungan liniernya terhadap variabel target masih relatif rendah.

Sementara itu, variabel *education* (0,04), *BPMeds* (0,00), dan *prevalentStroke* (0,00) termasuk dalam kategori korelasi sangat lemah, sehingga kontribusinya terhadap variabel target dapat dianggap sangat kecil dalam hubungan linier.

Hasil analisis korelasi ini digunakan sebagai tahap awal dalam proses seleksi fitur, di mana variabel dengan nilai korelasi lebih tinggi dianggap lebih berpotensi berkontribusi dalam pembentukan model prediksi penyakit jantung.

Dalam penelitian ini, perhitungan korelasi Pearson dilakukan dengan menggunakan fitur *age* sebagai variabel x dan *TenYearCHD* sebagai variabel y, berdasarkan 16 sampel data yang digunakan. Perhitungan ini mengacu pada Penghitungan Korelasi Awal sebagaimana ditunjukkan pada Persamaan 3.5. Persamaan 3.5 memiliki tiga komponen utama yang harus dihitung secara terpisah, yaitu $\sum(x_i - \bar{x})(y_i - \bar{y})$ sebagai pembilang, serta $\sum(x_i - \bar{x})^2$ dan $\sum(y_i - \bar{y})^2$ sebagai komponen dalam akar penyebut. Masing-masing komponen tersebut dihitung secara bertahap pada persamaan-persamaan berikut :

Langkah pertama adalah menghitung rata-rata fitur *age*. Data yang digunakan adalah x = [60, 36, 37, 43, 52, 42, 41, 54, 59, 63, 60, 61, 59, 61, 62, 60] dengan jumlah total $\sum x$ = 870. Rata-rata nilai *age* dihitung menggunakan persamaan (3.6).

$$\bar{x} = \frac{\sum x}{n} \tag{3. 6}$$

$$\bar{x} = \frac{870}{16} = 54.375$$

Berdasarkan persamaan (3.6), diperoleh nilai rata-rata fitur *age* sebesar $\bar{x}$ = 54,375 tahun. Rata-rata ini selanjutnya dijadikan acuan untuk menghitung deviasi atau selisih masing-masing data terhadap nilai pusat.

Langkah kedua adalah menghitung rata-rata variabel target *TenYearCHD*. Variabel ini bersifat biner dengan nilai y = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1], di mana 0 berarti tidak berisiko dan 1 berarti berisiko mengalami penyakit serangan jantung. Jumlah total nilai target adalah $\sum y$ = 8, sehingga rata-ratanya dihitung menggunakan persamaan (3.7).

$$\bar{y} = \frac{\sum y}{n} \tag{3. 7}$$

$$\bar{y} = \frac{8}{16} = 0.5$$

Hasil Persamaan 3.7 menunjukkan bahwa rata-rata variabel target adalah $\bar{y}$ = 0,5. Hal ini mencerminkan proporsi yang seimbang antara sampel positif dan negatif dalam data yang digunakan, karena tepat setengah dari 16 sampel memiliki nilai target 1.

Setelah rata-rata kedua variabel diperoleh, langkah berikutnya adalah menghitung selisih setiap nilai data terhadap rata-ratanya masing-masing. Proses ini menggunakan persamaan 3.8 dan persamaan 3.9 di bawah ini.

$$(x_i - \bar{x}) \quad (3.\ 8)$$

$$(y_i - \bar{y}) \quad (3.\ 9)$$

Sebagai ilustrasi penerapan persamaan 3.8 dan persamaan 3.9, beberapa contoh perhitungan selisih untuk variabel *age* adalah sebagai berikut: nilai pertama menghasilkan (60 − 54,375) = 5,625, nilai kedua menghasilkan (36 − 54,375) = −18,375, dan nilai ke-10 menghasilkan (63 − 54,375) = 8,625. Sementara itu, untuk variabel target yang hanya bernilai 0 atau 1, selisih yang mungkin hanya ada dua, yaitu (0 − 0,5) = −0,5 untuk delapan data pertama dan (1 − 0,5) = 0,5 untuk delapan data terakhir.

Setelah seluruh selisih dihitung menggunakan Persamaan 3.8 dan 3.9, langkah selanjutnya adalah mengalikan setiap pasangan selisih antara variabel *x* dan variabel *y*, kemudian menjumlahkan seluruh hasil perkalian tersebut. Proses ini dinyatakan dalam persamaan 3.10.

$$\sum(x_i - \bar{x})\,(y_i - \bar{y}) \quad (3.\ 10)$$

Penerapan Persamaan 3.10 pada 16 sampel data dilakukan dengan mempertimbangkan bahwa delapan data pertama memiliki $(y_i - \bar{y}) = -0{,}5$ dan delapan data terakhir memiliki $(y_i - \bar{y}) = 0{,}5$, sehingga perhitungannya adalah sebagai berikut.

$\sum(x_i - \bar{x})(y_i - \bar{y}) = (5.625)(-0.5) + (-18.375)(-0.5) + (-17.375)(-0.5) + (-11.375)(-0.5) + (-2.375)(-0.5) + (-12.375)(-0.5) + (-13.375)(-0.5) + (-0.375)(-0.5) + (4.625)(0.5) + (8.625)(0.5) + (5.625)(0.5) + (6.625)(0.5) + (4.625)(0.5) + (6.625)(0.5) + (7.625)(0.5) + (5.625)(0.5)$

$= (-2.8125 + 9.1875 + 8.6875 + 5.6875 + 1.1875 + 6.1875 + 6.6875 + 0.1875) + (2.3125 + 4.3125 + 2.8125 + 3.3125 + 2.3125 + 3.3125 + 3.8125 + 2.8125)$

$\sum(x_i - \bar{x})(y_i - \bar{y}) = 60.00$

Berdasarkan perhitungan pada Persamaan 3.10, diperoleh nilai $\sum(x_i - \bar{x})(y_i - \bar{y})$ = 60,00. Nilai positif ini menunjukkan adanya kecenderungan bahwa semakin tinggi usia seseorang, semakin besar pula kemungkinan untuk masuk ke dalam kategori berisiko terkena penyakit serangan jantung.

Komponen berikutnya yang diperlukan dalam Persamaan 3.5 adalah jumlah kuadrat selisih variabel *x* terhadap rata-ratanya. Nilai ini menggambarkan variasi total dari fitur *age* pada keseluruhan sampel dan dihitung menggunakan Persamaan 3.11.

$$\sum(x_i - \bar{x})^2 \qquad (3.\ 11)$$

Dengan menerapkan Persamaan 3.11 pada seluruh 16 sampel data, setiap selisih dikuadratkan lalu dijumlahkan sehingga diperoleh perhitungan sebagai berikut.

$\sum(x_i - \bar{x})^2 =$

$(5.6252)+(-18.3752)+(-17.3752)+(-11.3752)+(-2.3752)+(-12.3752)+(-13.3752)+(-0.3752)+(4.6252)+(8.6252)+(5.6252)+(6.6252)+(4.6252)+(6.6252)+(7.6252)+(5.6252) = 1464.75$

Hasil perhitungan pada Persamaan 3.11 menunjukkan bahwa nilai $\sum(x_i - \bar{x})^2 = 1.464{,}75$. Nilai ini mencerminkan seberapa besar variasi yang terdapat pada fitur *age* di antara seluruh sampel yang digunakan dalam perhitungan.

Komponen terakhir yang diperlukan dalam Persamaan 3.5 adalah jumlah kuadrat selisih variabel target *y* terhadap rata-ratanya. Perhitungan ini dilakukan menggunakan Persamaan 3.12 sebagai berikut.

$$\sum(y_i - \bar{y})^2 \tag{3. 12}$$

Karena variabel *TenYearCHD* hanya bernilai 0 atau 1, penerapan Persamaan 3.12 dapat disederhanakan. Dari 16 sampel, terdapat 8 data dengan $(y_i - \bar{y}) = -0{,}5$ dan 8 data dengan $(y_i - \bar{y}) = 0{,}5$, sehingga perhitungannya adalah sebagai berikut.

$$\sum(y_i - \bar{y})^2 = 8\,(-0.5)^2 + 8(0.5)^2 = 8 \times 0.25 + 8 \times 0.25 = 4$$

Hasil perhitungan pada Persamaan 3.12 diperoleh nilai $\sum(y_i - \bar{y})^2 = 4$. Nilai ini relatif kecil karena variabel target bersifat biner dan hanya memiliki dua kemungkinan selisih terhadap rata-ratanya.

Setelah ketiga komponen diperoleh, yaitu $\sum(x_i - \bar{x})(y_i - \bar{y}) = 60{,}00$ berdasarkan Persamaan 3.10, $\sum(x_i - \bar{x})^2 = 1.464{,}75$ berdasarkan Persamaan 3.11, dan $\sum(y_i - \bar{y})^2 = 4$ berdasarkan Persamaan 3.12, seluruh nilai tersebut disubstitusikan ke dalam Persamaan 3.5 untuk mendapatkan nilai korelasi Pearson akhir.

$$r_{age,CHD} = \frac{60.00}{sqrt(1464.75 \times 4)}$$

$$= \frac{60.00}{sqrt(5859)}$$

$$= \frac{60.00}{76.54}$$

$$= 0.784 = 0.78$$

Berdasarkan hasil substitusi pada Persamaan 3.5 di atas, diperoleh nilai korelasi Pearson antara fitur *age* dan variabel target *TenYearCHD* sebesar *r*(age, CHD) = 0,78. Nilai tersebut tergolong dalam kategori korelasi kuat dan bersifat positif, yang berarti semakin tinggi usia seseorang maka semakin tinggi pula risiko terjadinya penyakit jantung dalam kurun waktu sepuluh tahun ke depan. Dengan demikian, fitur *age* memiliki korelasi yang signifikan terhadap variabel target dan dinilai layak untuk dipertahankan sebagai salah satu fitur dalam pembangunan model prediksi.

3. Eliminasi Fitur Bertahap (Iteratif)

RFE bekerja dengan menghapus fitur dengan kontribusi paling kecil berdasarkan *importance* (hasil *gain* rata-rata).

Prosesnya:

a. Model awal dibangun dengan semua fitur (15 variabel).

b. Hitung *importance score* dari masing-masing fitur.

c. Hapus fitur dengan nilai terkecil.

d. Ulangi langkah sampai tersisa fitur paling signifikan.

Hasil iterasi RFE dengan contoh hasil akhir setelah 7 iterasi ditunjukkan pada Tabel 3.14.

Tabel 3.14 Contoh Perhitungan Hasil Akhir

Iterasi	Fitur Dihapus	Alasan	Sisa Fitur
1	*BPMeds*	Nilai *gain* = 0.000	14

Iterasi	Fitur Dihapus	Alasan	Sisa Fitur
2	*prevalentStroke*	Nilai *gain* = 0.000	13
3	diabetes	Nilai *gain* = 0.002	12
4	*currentSmoker*	Nilai *gain* = 0.038	11
5	*cigsPerDay*	Nilai *gain* = 0.020	10
6	*education*	Nilai *gain* = 0.020	9
7	*heartRate*	Nilai *gain* = 0.035	8

Tabel 3.14 menampilkan hasil seleksi fitur menggunakan metode RFE berdasarkan nilai gain yang diperoleh dari model awal. Proses ini dilakukan secara iteratif dengan mengeliminasi satu fitur pada setiap tahap, yaitu fitur yang memiliki nilai gain terendah atau kontribusi paling kecil terhadap peningkatan kinerja model.

Pada iterasi pertama, fitur *BPMeds* dihapus karena memiliki nilai *gain* paling rendah sebesar 0.000. Proses kemudian dilanjutkan dengan penghapusan fitur *prevalentStroke* (*gain* = 0.000), *diabetes* (*gain* = 0.002), *currentSmoker* (*gain* = 0.038), *cigsPerDay* (*gain* = 0.020), *education* (*gain* = 0.020), dan *heartRate* (*gain* = 0.035). Setelah tujuh kali iterasi, diperoleh delapan fitur tersisa yang memiliki nilai *gain* lebih tinggi dan dianggap paling berkontribusi dalam proses prediksi risiko penyakit jantung.

Proses ini dilakukan untuk menyederhanakan model melalui pemilihan fitur-fitur yang paling relevan, sehingga dapat meningkatkan efisiensi, mengurangi kompleksitas, serta menekan risiko *overfitting* tanpa menurunkan tingkat akurasi. Nilai *Gain* pada algoritma XGBoost dihitung menggunakan Persamaan (3.13). Nilai tersebut digunakan untuk mengukur peningkatan kualitas pemisahan (*split*) yang dihasilkan oleh suatu fitur,

sehingga fitur dengan nilai *Gain* yang lebih besar dianggap memberikan kontribusi lebih tinggi terhadap proses pembentukan pohon keputusan.

$$Gain_{split} = \frac{1}{2}\left(\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right) - \gamma \quad (3.\ 13)$$

Pada Persamaan (3.13), $G_L$dan $G_R$menyatakan jumlah gradien pada cabang kiri dan kanan, sedangkan $H_L$dan H_R merupakan jumlah nilai *Hessian* pada masing-masing cabang. Parameter λ berfungsi sebagai regularisasi untuk mengendalikan kompleksitas model, sedangkan γmerupakan penalti terhadap penambahan *split* baru. Semakin besar nilai *Gain* yang dihasilkan, semakin baik kualitas pemisahan data oleh fitur tersebut sehingga fitur tersebut memiliki kontribusi yang lebih besar dalam pembentukan model XGBoost.

4. Penentuan Fitur Terpilih

Setelah dilakukan proses eliminasi berulang, diperoleh 8 fitur terbaik berdasarkan nilai *importance* tertinggi yang disajikan pada Tabel 3.15.

Tabel 3.15 Hasil Perolehan 8 Fitur Terbaik

No	Nama Fitur	*Importance*	Keterangan
1	*age*	0.312927	Dipertahankan
2	*prevalentHyp*	0.158555	Dipertahankan
3	*BMI*	0.104366	Dipertahankan
4	*sysBP*	0.095716	Dipertahankan
5	*totChol*	0.093975	Dipertahankan
6	*male*	0.080867	Dipertahankan
7	*diaBP*	0.079759	Dipertahankan
8	*glucose*	0.073835	Dipertahankan

Hasil seleksi fitur menggunakan RFE yang menghasilkan delapan fitur terbaik disajikan pada Tabel 3.15. *Importance* merepresentasikan tingkat kontribusi masing-masing fitur terhadap performa model, dengan

nilai yang lebih besar menandakan pengaruh yang lebih dominan dalam proses prediksi risiko penyakit jantung.

Berdasarkan tabel, fitur *age* memiliki nilai *importance* tertinggi sebesar 0.312927, yang menunjukkan bahwa faktor usia merupakan variabel paling berpengaruh dalam menentukan kemungkinan terjadinya penyakit jantung dalam sepuluh tahun ke depan. Fitur *prevalentHyp* (0.158555) dan *BMI* (0.104366) juga menunjukkan kontribusi yang cukup besar, yang menandakan bahwa kondisi hipertensi dan indeks massa tubuh merupakan faktor penting dalam model prediksi. Fitur *sysBP* (0.095716) dan *totChol* (0.093975) berada pada kategori pengaruh menengah, yang menunjukkan bahwa tekanan darah sistolik dan kadar kolesterol turut memberikan kontribusi terhadap prediksi risiko penyakit jantung. Adapun fitur *male* (0.080867), *diaBP* (0.079759), dan *glucose* (0.073835) menunjukkan pengaruh yang relatif kecil, tetapi masih memiliki kontribusi dalam meningkatkan kinerja model secara keseluruhan.

Hasil tersebut menunjukkan bahwa model lebih peka terhadap faktor fisiologis utama seperti usia, hipertensi, indeks massa tubuh, dan tekanan darah dibandingkan variabel lainnya. Oleh karena itu, fitur-fitur tersebut dapat dimanfaatkan untuk mendukung analisis medis dalam pengambilan keputusan terkait pencegahan penyakit jantung.

Hasil proses seleksi fitur menunjukkan bahwa delapan variabel berikut merupakan fitur paling berpengaruh terhadap risiko penyakit jantung 10 tahun (*TenYearCHD*), yaitu: *age, prevalentHyp, BMI, sysBP, totChol, male, diaBP, glucose.*

Kedelapan fitur ini kemudian dirangkum pada Tabel 3.15 sebagai fitur terpilih yang akan digunakan dalam tahap pembangunan model XGBoost. Pemilihan fitur tersebut didasarkan pada nilai *feature importance* tertinggi yang diperoleh dari proses seleksi, sehingga hanya variabel yang benar-benar relevan dan berkontribusi signifikan terhadap hasil prediksi yang dipertahankan.

Meskipun RFE menghasilkan delapan fitur terpilih yang dipertahankan dalam model, hasil tersebut tidak dapat diartikan bahwa fitur yang dieliminasi tidak memiliki hubungan dengan risiko penyakit jantung. Dalam konteks *machine learning*, RFE berfokus pada pemilihan fitur yang memiliki kontribusi paling signifikan terhadap peningkatan performa model pada dataset yang digunakan. Oleh karena itu, fitur yang tidak terpilih bukan berarti tidak penting, melainkan kontribusinya relatif lebih kecil dibandingkan fitur lainnya dalam proses klasifikasi.

Berdasarkan hasil seleksi fitur, model memanfaatkan delapan atribut utama yaitu *age, prevalentHyp, BMI, sysBP, totChol, male, diaBP,* dan *glucose* dalam membedakan kelas risiko penyakit jantung. Sementara itu, beberapa fitur lainnya seperti *diabetes, currentSmoker, education, BPMeds,* dan *prevalentStroke* tidak terpilih dalam proses seleksi karena memiliki nilai *importance* yang lebih rendah atau kontribusi yang tidak signifikan terhadap kinerja model. Hal tersebut dapat dipengaruhi oleh karakteristik dataset, distribusi data, serta adanya hubungan antar fitur (*multicollinearity*), di mana sebagian informasi dari fitur yang dieliminasi telah direpresentasikan oleh fitur lain yang memiliki kontribusi lebih dominan dalam proses prediksi.

Dari perspektif medis, hasil seleksi fitur tidak dapat dijadikan satu-satunya dasar untuk menilai penting atau tidaknya suatu faktor risiko. Menurut World Health Organization (WHO), beberapa faktor seperti hipertensi, kadar kolesterol, indeks massa tubuh (BMI), kebiasaan merokok, dan diabetes merupakan faktor risiko utama penyakit jantung. Oleh karena itu, meskipun beberapa variabel tidak terpilih dalam proses seleksi fitur, atribut tersebut tetap memiliki nilai klinis yang penting dalam penilaian risiko penyakit jantung. Dengan demikian, hasil seleksi fitur pada penelitian ini perlu dipahami sebagai upaya optimasi model klasifikasi untuk meningkatkan performa prediksi, bukan sebagai penilaian absolut terhadap signifikansi medis suatu variabel.

Dengan demikian, hasil seleksi fitur pada penelitian ini perlu dipahami sebagai upaya optimasi model klasifikasi untuk memperoleh performa prediksi yang lebih baik, bukan sebagai dasar untuk menentukan penting atau tidaknya suatu faktor risiko secara medis. Penafsiran hasil penelitian harus mempertimbangkan baik aspek komputasional maupun aspek klinis agar kesimpulan yang diperoleh tetap relevan dengan konteks kesehatan dan karakteristik dataset yang digunakan.

Dengan adanya seleksi fitur ini, model yang dibangun menjadi lebih efisien karena hanya memproses atribut penting, lebih stabil terhadap variasi data, serta memiliki interpretasi yang lebih jelas terhadap faktor-faktor risiko penyakit jantung. Tabel 3.16 menyajikan contoh hasil pengurangan fitur setelah dilakukan proses RFE.

Tabel 3.16 Hasil Pengurangan Fitur Setelah Dilakukan Proses RFE

No.	*male*	*age*	*prevalentHyp*	*totChol*	*sysBP*	*diaBP*	*BMI*	*glucose*	*TenYearCHD*
1	0	60	0	247	130	88	30.36	74	0
2	1	36	0	295	102	68	28.15	63	0
3	1	37	1	225	1245	925	38.53	83	0
4	0	43	0	243	1165	80	26.87	78	0
5	1	52	1	178	160	98	40.11	225	0
6	0	42	1	233	153	101	28.93	90	0
7	0	41	0	237	122	78	23.28	74	0
8	1	54	0	195	132	835	26.21	100	0
9	0	59	1	209	150	85	20.77	88	1
10	0	63	1	267	1565	92.5	27.1	79	1
11	0	60	1	325	182	106	27.61	77	1
12	0	61	1	238	857.5	88.75	23.94	83.5	1
13	0	59	1	249	160	91	23	85	1
14	0	61	1	238	857.5	88.75	23.94	83.5	1
15	0	62	1	291	923.4	99	27.36	78	1
16	0	60	1	286	173	100	25	81	1

Dari Tabel 3.16, setelah dilakukan proses *Recursive Feature Elimination* (RFE), dataset direduksi menjadi delapan fitur prediktor utama, yaitu *male, age, prevalentHyp, totChol, sysBP, diaBP, BMI,* dan *glucose*, beserta variabel target *TenYearCHD*. Pengurangan fitur dilakukan melalui eliminasi variabel yang menunjukkan kontribusi rendah terhadap performa model pada tahap seleksi sebelumnya. Oleh karena itu, hanya fitur yang memiliki tingkat signifikansi lebih tinggi dalam memprediksi risiko penyakit jantung yang dipertahankan.

Secara keseluruhan, hasil ini menunjukkan bahwa variabel seperti *age,* tekanan darah *(sysBP dan diaBP),* indeks massa tubuh (BMI)*,* kadar kolesterol *(totChol),* serta kadar glukosa *(glucose)* memiliki peran penting dalam membentuk

model prediksi. Sementara itu, variabel yang dieliminasi pada proses RFE dianggap memiliki pengaruh yang lebih kecil terhadap performa model. Oleh karena itu, penggunaan fitur yang lebih ringkas dilakukan untuk meningkatkan efisiensi komputasi, menekan kompleksitas model, dan meminimalkan risiko *overfitting* tanpa mengurangi informasi esensial yang dimiliki dataset.

3.6 Implementasi Metode XGBoost

XGBoost (*Extreme Gradient Boosting*) merupakan algoritma *ensemble learning* berbasis *decision tree* yang menerapkan teknik *gradient boosting*. Algoritma ini diperkenalkan oleh Tianqi Chen pada tahun 2014 dan dikenal luas sebagai salah satu metode yang paling banyak digunakan dalam bidang *machine learning* karena memiliki tingkat akurasi yang tinggi serta efisiensi komputasi yang baik.

3.6.1 Prinsip *Gradient Boosting*

Pada metode *gradient boosting*, pembentukan model dilakukan secara sekuensial dengan menambahkan *tree* baru yang bertujuan untuk mengoreksi kesalahan prediksi dari model sebelumnya. Prediksi akhir kemudian diperoleh melalui kombinasi seluruh model yang telah terbentuk. Proses ini secara matematis ditunjukkan pada Persamaan (3.14).

$$\widehat{\{y\}}_{\iota} = \sum_{\{k=1\}}^{\{K\}} fk(x_i) \qquad (3.\ 14)$$

Pada Persamaan (3.14), $\hat{y}_i$ merupakan nilai prediksi untuk data ke-i, $f_k(x_i)$ menyatakan fungsi prediksi yang dihasilkan oleh *tree* ke-k, sedangkan K

menunjukkan jumlah keseluruhan *tree* yang dibangun dalam model. Persamaan ini menunjukkan bahwa prediksi akhir diperoleh dari akumulasi kontribusi setiap *tree*.

Untuk memperoleh model yang optimal, setiap *tree* dilatih dengan meminimalkan fungsi objektif yang terdiri atas komponen *training loss* dan *regularization term*. Bentuk fungsi objektif tersebut ditunjukkan pada Persamaan (3.15).

$$L(\theta) = \sum_{\{i=1\}}^{\{n\}} l(y_i, \hat{y}_i) + \sum_{\{k=1\}}^{\{K\}} \Omega(f_k) \qquad (3.\ 15)$$

Dalam Persamaan (3.15), $l(y_i, \hat{y}_i)$ berfungsi sebagai *loss function* yang menghitung selisih antara nilai sebenarnya dan nilai prediksi model. Di sisi lain, $\Omega\,(f_i)$ merupakan komponen regularisasi yang bertujuan untuk membatasi kompleksitas *tree* yang dibangun. Integrasi kedua komponen tersebut membuat XGBoost mampu mencapai performa prediksi yang tinggi serta menjaga kemampuan generalisasi model agar terhindar dari *overfitting*.

Regularisasi ini berfungsi untuk mengurangi kompleksitas model, sehingga dapat meningkatkan kemampuan generalisasi dan mencegah *overfitting*. Perhitungan manual XGBoost bertujuan untuk memahami secara mendalam mekanisme kerja algoritma dalam memprediksi risiko penyakit jantung (*Coronary Heart Disease*/CHD) dalam 10 tahun ke depan. Perhitungan dilakukan langkah demi langkah (*step-by-step*) menggunakan sampel data yang telah difilter melalui *Recursive Feature Elimination* (RFE).

3.6.2 Perhitungan Manual XGBoost

Untuk memperjelas mekanisme kerja algoritma XGBoost dalam menghasilkan prediksi, dilakukan perhitungan secara manual. Proses perhitungan tersebut tidak melibatkan seluruh data dan fitur yang tersedia, tetapi menggunakan sejumlah sampel dan fitur terpilih sehingga tahapan perhitungan dapat dijelaskan dengan lebih sederhana dan mudah dipahami. Oleh karena itu, peneliti hanya menggunakan empat sampel data dari hasil seleksi fitur (RFE) dan memfokuskan perhitungan pada satu fitur utama, yaitu usia (*age*), yang secara klinis merupakan faktor penting dalam penentuan risiko penyakit jantung (CHD).

Pemilihan empat sampel data ini dilakukan agar proses penjelasan tahapan XGBoost seperti inisialisasi model, pembentukan prediksi awal, perhitungan *error* (selisih antara nilai aktual dengan prediksi), hingga pembentukan *tree* pertama dapat dijelaskan secara runtut dan tidak terlalu kompleks. Dengan menggunakan fitur age, proses seperti pembentukan aturan pemisahan (*split*), perhitungan nilai *gain, gradient, hessian*, serta penentuan nilai *leaf* pada pohon keputusan dapat disajikan secara lebih jelas dan sistematis. Meskipun disederhanakan, pendekatan ini tetap mewakili cara kerja asli XGBoost dalam membangun model prediksi.

Tabel 3.17 Contoh 4 Sampel Dataset yang Diambil dari Tabel 3.16

No.	***male***	***age***	***prevalentHyp***	***totChol***	***sysBP***	***diaBP***	***BMI***	***glucose***	***TenYearCHD***
1	0	60	0	247	130	88	30.36	74	0
2	1	36	0	295	102	68	28.15	63	0
9	0	59	1	209	150	85	20.77	88	1
10	0	63	1	267	1565	92.5	27.1	79	1

Dari tabel 3.17, terdapat 2 pasien dengan label negatif (*TenYearCHD* = 0) yang artinya tidak berisiko mengalami CHD dalam 10 tahun, dan 2 pasien dengan label positif (*TenYearCHD* = 1) yang berisiko CHD. Untuk perhitungan manual, fokus utama adalah pada fitur *age* (usia) karena merupakan prediktor kuat untuk risiko penyakit jantung. Pada tahap inisialisasi (iterasi ke-0), model XGBoost tidak menggunakan fitur prediktor, melainkan hanya berdasarkan distribusi kelas pada data. Oleh karena itu, probabilitas awal dihitung menggunakan Persamaan (3.16):

$$p_i = \frac{n_i}{N} \tag{3. 16}$$

Berdasarkan Persamaan (3.16), jumlah data pada kelas positif sebanyak 2 sampel dari total 4 sampel, sehingga diperoleh probabilitas awal sebesar:

$$p = \frac{2}{4} = 0.5$$

Berdasarkan hasil perhitungan pada Persamaan (3.16), diperoleh nilai probabilitas sebesar 0,5 yang mengindikasikan bahwa jumlah data pada kelas positif dan kelas negatif memiliki proporsi yang sama, yaitu masing-masing sebesar 50%. Nilai probabilitas ini kemudian digunakan untuk menghitung prediksi awal dalam bentuk *log-odds* menggunakan Persamaan (3.17). Transformasi ke bentuk *log-odds* dilakukan karena algoritma XGBoost mengoptimalkan fungsi objektif pada ruang *log-odds* sehingga proses pembelajaran model menjadi lebih efektif.

$$\widehat{(y)}^0 = \ln\left(\frac{p}{1-p}\right) \tag{3. 17}$$

Berdasarkan Persamaan (3.17), dengan nilai probabilitas awal sebesar 0,5, maka prediksi awal dihitung sebagai berikut.

$$\widehat{(y)}^0 = \log\left(\frac{0.5}{1-0.5}\right) = \log(1) = 0$$

Nilai 0 pada *log-odds* pada hasil persamaan 3.17 menunjukkan bahwa model belum memiliki kecenderungan (*bias*) terhadap salah satu kelas. Dengan kata lain, model memberikan peluang awal yang sama untuk kelas positif maupun kelas negatif. Tahap berikutnya adalah mengonversi nilai *log-odds* menjadi nilai probabilitas melalui Persamaan (3.18).

$$\hat{p}^{(i)} = \frac{1}{1 + e^{0-\hat{y}^{(1)(0)}}} \quad (3.\ 18)$$

Berdasarkan Persamaan (3.18), dengan nilai prediksi awal $\hat{y}^{(0)} = 0$, maka probabilitas awal dihitung sebagai berikut.

$$\hat{p}^{(i)} = \frac{1}{1 + e^{0-\hat{y}^{(1)(0)}}} = \frac{1}{2} = 0.5$$

Berdasarkan hasil perhitungan pada Persamaan (3.18), diperoleh probabilitas awal sebesar 0,5. Nilai tersebut menunjukkan bahwa setiap sampel memiliki peluang awal sebesar 50% untuk termasuk ke dalam kelas berisiko maupun tidak berisiko penyakit jantung. Hasil inisialisasi prediksi model XGBoost pada iterasi ke-0 ditunjukkan pada Tabel 3.18.

Tabel 3.18 Inisialisasi Prediksi Model XGBoost (Iterasi ke-0)

No	*TenYearCHD*	$\widehat{y^{(0)}}$	*pred*(0)
1	0	0	0.5
2	0	0	0.5
9	1	0	0.5
10	1	0	0.5

Tabel 3.18 menunjukkan tahap inisialisasi awal (Iterasi ke-0) pada model XGBoost sebelum proses pembelajaran dimulai. Pada tahap ini, model menetapkan

nilai prediksi awal yang seragam untuk seluruh sampel sebagai titik acuan. Nilai 0 pada kolom *F0(x)* merepresentasikan nilai *log-odds* awal, yang jika ditransformasikan menggunakan fungsi *sigmoid* akan menghasilkan probabilitas sebesar 0.5 atau 50%. Tahap inisialisasi ini bersifat netral dan berperan sebagai dasar bagi algoritma untuk menghitung residual antara nilai sebenarnya dan hasil prediksi. Residual tersebut kemudian dikoreksi secara bertahap melalui pembentukan *decision tree* pada iterasi selanjutnya.

Pada iterasi 0, semua prediksi identik karena model belum belajar pola apapun dari data. Langkah selanjutnya adalah membangun pohon pertama untuk memperbaiki prediksi ini.

1. Iterasi Awal : Menghitung *Gradient* dan *Hessian*

   Untuk klasifikasi biner dengan *binary:logistic loss function*, nilai gradien digunakan untuk mengukur arah dan besar kesalahan prediksi pada setiap sampel. Perhitungan gradien dilakukan menggunakan Persamaan (3.19).

$$(Gradien\ g_i = \text{pred}_i - y_i) \qquad (3.\ 19)$$

   Berdasarkan persamaan (3.19), Gradien menunjukkan arah dan besar kesalahan prediksi. Jika nilai prediksi lebih besar dari label aktual (misalnya 0.5 saat labelnya 0), maka gradien bernilai positif (0.5), artinya model perlu menurunkan prediksi. Sebaliknya, jika prediksi lebih kecil dari label (misalnya 0.5 saat labelnya 1), *gradient* bernilai negatif (-0.5), yang berarti model harus meningkatkan prediksi untuk mendekati nilai aktual.

Sementara itu, nilai *Hessian* digunakan untuk mengukur tingkat kelengkungan (*curvature*) fungsi *loss* pada setiap sampel. Perhitungan nilai Hessian dilakukan menggunakan Persamaan (3.20).

$$(h_i = \text{pred}_i \times (1 - \text{pred}_i)) \quad (3.\ 20)$$

Berdasarkan Persamaan (3.20), nilai *Hessian* diperoleh dari hasil perkalian probabilitas prediksi $(\hat{p}_i)$ dengan komplemennya $(1-\hat{p}_i)$. Pada iterasi awal, seluruh sampel memiliki probabilitas prediksi sebesar 0,5, sehingga nilai *Hessian* untuk setiap sampel dihitung sebagai berikut.

$$h_i = 0.5(1 - 0.5) = 0.5 \times 0.5 = 0.25$$

Berdasarkan hasil perhitungan pada Persamaan (3.20), diperoleh nilai *Hessian* sebesar 0,25 untuk setiap sampel. Nilai ini menunjukkan bahwa seluruh sampel memiliki tingkat kelengkungan fungsi *loss* yang sama pada iterasi awal. Hasil perhitungan *gradient* dan *Hessian* pada setiap sampel disajikan pada Tabel 3.19.

Tabel 3.19 Perhitungan *Gradient* dan *Hessian*

No	y_i	$pred_i$	$g_i = pred_i - y_i$	$h_i = pred_i(1 - pred_i)$
1	0	0.5	0.5	0.25
2	0	0.5	0.5	0.25
9	1	0.5	-0.5	0.25
10	1	0.5	-0.5	0.25

Tabel 3.19 menunjukkan contoh perhitungan nilai *gradient* (g_i) dan *hessian* (h_i) pada empat data sampel awal yang digunakan dalam tahap pertama pelatihan model XGBoost. Nilai y_i merupakan label aktual (0 atau 1), sedangkan $pred_i$ menunjukkan nilai prediksi awal model. Pada iterasi awal, prediksi setiap sampel biasanya dimulai dari nilai 0.5, karena model

belum melakukan pembelajaran dan menganggap probabilitas kedua kelas (positif dan negatif) masih seimbang.

2. Mencari *Split* Terbaik Berdasarkan *Age*

Pada proses pembangunan pohon keputusan dalam XGBoost, algoritma mencoba berbagai nilai *threshold* untuk menemukan titik pemisahan (*split*) yang paling optimal. Dalam contoh ini, dicoba split pada nilai *age* = 58, yang membagi data menjadi dua kelompok, yaitu pasien muda dengan *age* < 58 dan pasien tua dengan *age* ≥ 58. Hasil pembagian data berdasarkan threshold tersebut adalah sebagai berikut.

Left Node (*age* < 58): Sampel 2 → *age* = 36 tahun

Right Node (*age* ≥ 58): Sampel 1, 9, 10 → *age* = (60, 59, 63) tahun

Setelah data dibagi ke dalam *Left Node* dan *Right Node*, langkah selanjutnya adalah menghitung jumlah gradien dan *Hessian* pada setiap *node*. Total gradien diperoleh dengan menjumlahkan seluruh nilai gradien dari sampel yang berada pada node tersebut menggunakan Persamaan (3.21).

$$G_j = \sum_{i \in I_j} g_i \tag{3. 21}$$

Jumlah gradien pada masing-masing *node* dihitung menggunakan persamaan 3.21.

Left Node (*age* < 58):

$$G_L = \sum_{i \in Left} g_i$$

$$G_L = 0.5$$

Right Node (*age* ≥ 58):

$$G_R = \sum_{i \in Right} g_i$$

$$G_R = 0.5 + (-0.5) + (-0.5)$$

$$G_R = -0.5$$

Selanjutnya, total nilai *Hessian* pada setiap node dihitung dengan menjumlahkan seluruh nilai *Hessian* dari sampel yang berada pada *node* tersebut menggunakan Persamaan (3.22).

$$H_j = \sum_{i \in I_j} h_i \quad (3.\ 22)$$

Berdasarkan Persamaan (3.22), jumlah *Hessian* pada masing-masing *node* dihitung sebagai berikut.

Left Node (*age* < 58):

$$H_L = \sum_{i \in Left} h_i$$

$$H_L = 0.25$$

Right Node (*age* ≥ 58):

$$H_R = \sum_{i \in Right} h_i$$

$$H_R = 0.25 + 0.25 + 0.25$$

$$H_R = 0.75$$

Berdasarkan hasil persamaan (3.21), diperoleh total gradien pada *Left Node* sebesar 0,5 dan pada *Right Node* sebesar −0,5. Sementara itu, total *Hessian* pada *Left Node* pada persamaan (3.22) sebesar 0,25 dan pada *Right Node* sebesar 0,75. Nilai gradien dan *Hessian* tersebut selanjutnya digunakan untuk menghitung nilai *Gain*.

3. Menghitung *Gain* untuk Evaluasi *Split*

Setelah diperoleh total gradien dan *Hessian* pada masing-masing *node*, langkah berikutnya adalah mengevaluasi kualitas pemisahan (*split*) menggunakan nilai *Gain*. Besarnya nilai *Gain* menunjukkan tingkat pengurangan *loss function* yang dihasilkan oleh suatu pemisahan data (*split*). Nilai *Gain* yang semakin besar menandakan bahwa kualitas pemisahan data yang dilakukan semakin baik. Perhitungan *Gain* dilakukan menggunakan Persamaan (3.23).

$$(Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma) \quad (3.\ 23)$$

Dengan menggunakan hasil perhitungan sebelumnya, yaitu G_L = 0,5, H_L = 0,25, G_R = -0,5, H_R = 0,75, serta parameter regularisasi λ = 1 dan γ = 0, maka dilakukan substitusi ke dalam Persamaan (3.23) sebagai berikut.

$$(Gain = \frac{1}{2}\left[\frac{(0.5)^2}{0.25 + 1} + \frac{(-0.5)^2}{0.75 + 1} - \frac{(0.5 - 0.5)^2}{0.25 + 0.75 + 1}\right] - 0)$$

Perhitungan detail:

$$(Gain = \frac{1}{2}[0.2 + 0.143 - 0] = \frac{1}{2}(0.343) = 0.1715)$$

Berdasarkan hasil perhitungan pada Persamaan (3.23), diperoleh nilai *Gain* sebesar 0,1715. Karena nilai *Gain* bernilai positif, maka pemisahan data pada *age* = 58 mampu mengurangi *loss function*, sehingga *split* tersebut layak dipilih sebagai kandidat pemisahan pada pohon keputusan XGBoost.

4. Menghitung Bobot Optimal untuk Setiap *Leaf*

Setelah diperoleh *split* terbaik berdasarkan nilai Gain, langkah selanjutnya adalah menghitung bobot (*weight*) optimal pada setiap *leaf*. Bobot ini digunakan sebagai nilai koreksi yang akan ditambahkan pada prediksi setiap sampel yang berada pada *leaf* tersebut. Perhitungan bobot dilakukan menggunakan Persamaan (3.24).

$$(w_j = -\frac{G_j}{H_j + \lambda}) \tag{3. 24}$$

Berdasarkan Persamaan (3.24), bobot pada masing-masing *leaf* dihitung sebagai berikut.

Leaf Kiri (sampel 2):

$$(w_L = -\frac{0.5}{0.25 + 1} = -\frac{0.5}{1.25} = -0.4)$$

Bobot negatif (-0.4) berarti pasien di *leaf* ini perlu prediksi lebih rendah (risiko lebih kecil), sesuai dengan sampel 2 yang merupakan pasien muda (36 tahun) tanpa risiko CHD.

Leaf Kanan (sampel 1, 9, 10):

$$(w_R = -\frac{-0.5}{0.75 + 1} = \frac{0.5}{1.75} = 0.286)$$

Bobot sebesar 0,286 menunjukkan bahwa prediksi pada *leaf* ini akan dikoreksi ke arah yang lebih tinggi, sehingga mengindikasikan risiko penyakit jantung yang lebih besar. Berdasarkan hasil perhitungan dari persamaan 3.24, diperoleh bobot −0,4 pada *Left Leaf* dan 0,286 pada *Right Leaf*.

5. Update Prediksi dengan *Learning Rate*

Setelah diperoleh bobot optimal pada setiap *leaf*, langkah berikutnya adalah memperbarui nilai prediksi (*prediction update*) pada iterasi pertama. Pembaruan prediksi dilakukan dengan menambahkan hasil perkalian antara bobot *leaf* dan *learning rate* (η) terhadap prediksi sebelumnya. Perhitungan pembaruan prediksi dilakukan menggunakan Persamaan (3.25).

Prediksi diupdate menggunakan formula:

$$\hat{y}_i^{(1)} = \hat{y}_i^{(0)} + \eta w_j \qquad (3.\ 25)$$

Pada penelitian ini, digunakan nilai *learning rate* sebesar $\eta = 0{,}3$. Berdasarkan Persamaan (3.25), nilai prediksi pada masing-masing *leaf* dihitung sebagai berikut. Untuk sampel 2 (*Left Leaf*):

$$\hat{y}_2^{(1)} = \hat{y}_2^{(0)} + \eta w_L$$

$$\hat{y}_2^{(1)} = 0 + 0.3 \times (-0.4) = -0.12$$

Untuk Sampel 1, 9, 10 (*Right Leaf*):

$$\hat{y}_{1,9,10}^{(1)} = \hat{y}_{1,9,10}^{(0)} + \eta w_L$$

$$\hat{y}_{1,9,10}^{(1)} = 0 + 0.3 \times 0.286 = 0.086$$

Hasil perhitungan pada persamaan 3.25 menunjukkan nilai prediksi sebesar −0,12 untuk *Left Leaf* dan 0,086 untuk *Right Leaf*. Nilai ini masih berada pada skala *log-odds* sehingga belum dapat diinterpretasikan sebagai probabilitas. Selanjutnya, nilai tersebut dikonversi ke dalam bentuk probabilitas melalui fungsi *sigmoid*.

6. Mengkonversi ke Probabilitas

Nilai prediksi yang diperoleh pada iterasi pertama masih berada dalam bentuk *log-odds*. Untuk memudahkan interpretasi hasil prediksi, nilai tersebut selanjutnya ditransformasikan menjadi probabilitas dengan menggunakan fungsi *sigmoid*. Perhitungan probabilitas dilakukan menggunakan Persamaan (3.26).

$$p = \sigma(\hat{y}) = \frac{1}{1 + e^{-\hat{y}}} \tag{3. 26}$$

Berdasarkan Persamaan (3.26), probabilitas untuk masing-masing sampel dihitung sebagai berikut.

Sampel 2 (*Left Leaf*):

$$(p_2 = \frac{1}{1 + e^{-(-0.12)}} = \frac{1}{1 + e^{0.12}} = \frac{1}{1 + 1.1275} = 0.470)$$

Nilai probabilitas sebesar 0,470 menunjukkan bahwa peluang sampel 2 mengalami penyakit jantung sebesar 47,0%. Nilai tersebut lebih rendah dibandingkan probabilitas awal sebesar 50%, sehingga mengindikasikan risiko penyakit jantung yang lebih rendah.

Sampel 1, 9, dan 10 (*Right Leaf*):

$$(p_{1,9,10} = \frac{1}{1+e^{-0.086}} = \frac{1}{1+0.918} = \frac{1}{1.918} = 0.521)$$

Probabilitas naik dari 50% ke 52.1%, menunjukkan risiko lebih tinggi untuk

Nilai probabilitas sebesar 0,521 menunjukkan bahwa peluang sampel 1, 9, dan 10 mengalami penyakit jantung sebesar 52,1%. Nilai tersebut lebih tinggi dibandingkan probabilitas awal sebesar 50%, sehingga mengindikasikan peningkatan risiko penyakit jantung.

Berdasarkan hasil konversi probabilitas menggunakan Persamaan (3.26), sampel pada *Left* memiliki probabilitas risiko yang lebih rendah dibandingkan sampel pada *Right Leaf*. Ringkasan hasil perhitungan pada iterasi pertama XGBoost disajikan pada Tabel 3.20.

Tabel 3.20 Ringkasan Iterasi 1

No	*TenYear CHD*	*y(0)*	g_i	h_i	*Leaf*	*w*	*y(1)*	*p(1)*	Prediksi
1	0	0.5	0.5	0.25	R	0.286	0.086	0.521	Positif
2	0	0.5	0.5	0.25	L	-0.400	-0.120	0.470	Negatif
9	1	0.5	-0.5	0.25	R	0.286	0.086	0.521	Positif
10	1	0.5	-0.5	0.25	R	0.286	0.086	0.521	Positif

Catatan: Prediksi = Positif jika p ≥ 0.5 dan Prediksi = Negatif jika p < 0.5

Tabel 3.20 mengilustrasikan transisi dari prediksi awal (Iterasi 0) menuju prediksi yang lebih terperinci melalui penghitungan nilai *gradient* (gi) dan *hessian* (hi) pada setiap sampel data. Kolom *Leaf* menunjukkan pengelompokkan sampel ke dalam simpul daun kiri (L) atau kanan (R) berdasarkan aturan keputusan yang terbentuk, yang kemudian

menghasilkan bobot optimal (w) untuk mengoreksi kesalahan prediksi sebelumnya.

Perubahan signifikan terlihat pada nilai *p* (1), di mana probabilitas setiap sampel tidak lagi bernilai seragam 0.5. Sampel nomor 9 dan 10 yang memiliki label aktual positif kini memiliki probabilitas di atas ambang batas (0.521), sehingga diklasifikasikan sebagai positif. Hasil ini menunjukkan bahwa pada iterasi pertama, model mulai mampu menyesuaikan bobot prediksinya untuk membedakan karakteristik antara responden yang berisiko penyakit jantung dan yang tidak, berdasarkan pola fitur yang telah dipelajari.

Pada tahap analisis, diketahui bahwa Sampel 2 berhasil diklasifikasikan dengan benar sebagai negatif, sedangkan Sampel 9 dan 10 juga diklasifikasikan dengan tepat sebagai positif. Namun, terdapat kesalahan pada sampel 1 yang seharusnya termasuk kategori negatif tetapi diprediksi sebagai positif. Dengan demikian, akurasi sementara pada tahap ini adalah 3 dari 4 sampel, yaitu sebesar 75%. Tahapan ini terus berlanjut ke iterasi selanjutnya dengan membangun pohon keputusan kedua, ketiga, dan seterusnya hingga jumlah *n_estimators* terpenuhi. Setiap pohon baru difokuskan untuk memperbaiki error dari model sebelumnya sehingga meningkatkan akurasi prediksi secara bertahap.

3.6.3 *Hyperparameter* XGBoost

Pada iterasi pertama, model XGBoost menghasilkan akurasi sebesar 75%, di mana 3 dari 4 sampel berhasil diprediksi dengan benar. Kesalahan pada Sampel

1 memicu model untuk melakukan pembelajaran ulang pada iterasi berikutnya. Konsep ini disebut *boosting*, di mana setiap *decision tree* yang dibangun secara berurutan berupaya memperbaiki kesalahan model sebelumnya dengan memanfaatkan *residual error* atau gradien dari fungsi *loss* sebagai dasar pembaruan. Namun, agar proses perbaikan kesalahan tersebut berjalan optimal dan terhindar dari risiko *overfitting* maupun *underfitting*, diperlukan pengaturan parameter yang presisi melalui *hyperparameter tuning*. Pada penelitian ini, *GridSearchCV* diterapkan untuk mengoptimalkan kombinasi *hyperparameter* agar diperoleh akurasi yang maksimal dengan model yang tetap stabil.

Rincian konfigurasi parameter yang diuji dalam proses *tuning* tersebut disajikan pada Tabel 3.21 berikut, yang mencakup nilai *default* serta rentang nilai pencarian. Pemilihan parameter ini didasarkan pada peran krusialnya dalam mengontrol kompleksitas, kecepatan pembelajaran, serta stabilitas model dalam mengenali pola data risiko penyakit jantung secara lebih akurat.

Tabel 3.21 Parameter yang Digunakan

Parameter	Nilai *Default*	Nilai *Tuning* (*GridSearchCV*)	Keterangan
n_estimators	100	[50, 100, 200]	Jumlah pohon yang akan dibangun
max_depth	6	[3, 5, 7]	Kedalaman maksimal setiap pohon
learning_rate	0.3	[0.01, 0.1, 0.3]	Laju pembelajaran/kontribusi setiap pohon
subsample	1.0	[0.8, 0.9, 1.0]	Rasio sampel data untuk membangun pohon
colsample_bytree	1.0	[0.8, 0.9, 1.0]	Rasio fitur yang digunakan per pohon
gamma	0	[0, 0.1, 0.2]	Minimum *loss reduction* untuk *split*
objective	*binary*:*logistic*	*binary*:*logistic*	Fungsi objektif untuk klasifikasi biner
eval_metric	*logloss*	*logloss*	Metrik evaluasi selama training
random_state	42	42	*Seed* untuk reprodusibilitas

Tabel 3.21 merinci konfigurasi parameter yang digunakan dalam pengembangan model XGBoost, yang mencakup nilai bawaan (*default*) serta rentang nilai pencarian (*tuning*) melalui metode *GridSearchCV*. Pengaturan parameter ini bertujuan untuk mencari kombinasi optimal yang dapat menghasilkan akurasi prediksi tertinggi serta mencegah terjadinya *overfitting*.

Proses *tuning* difokuskan pada beberapa parameter krusial, seperti *n_estimators* untuk menentukan jumlah iterasi pohon, *max_depth* untuk mengontrol kompleksitas struktur pohon, dan *learning_rate* untuk mengatur tingkat koreksi kesalahan pada setiap iterasi. Selain itu, parameter seperti *subsample* dan *colsample_bytree* digunakan untuk memperkenalkan elemen acak dalam pemilihan data dan fitur guna meningkatkan ketahanan model terhadap variansi data. Seluruh kombinasi ini diuji secara sistematis untuk mendapatkan model final yang memiliki kemampuan generalisasi terbaik pada dataset risiko penyakit jantung.

3.6.4 *Tuning Parameter*

Proses optimasi parameter dilakukan menggunakan *GridSearchCV*, yaitu metode sistematis untuk mencari kombinasi parameter terbaik dengan menguji seluruh ruang pencarian (*parameter space*) yang telah ditentukan sebelumnya. Dengan total dataset sebanyak 4.238 baris, *GridSearchCV* memungkinkan model XGBoost untuk mengeksplorasi konfigurasi fitur secara mendalam guna meminimalkan *error* dan meningkatkan akurasi prediksi.

Agar hasil pelatihan lebih dapat dipercaya dan tidak mudah mengalami *overfitting*, *GridSearchCV* dipadukan dengan teknik 5-*Fold Cross Validation*. Pada pendekatan ini, data dibagi menjadi lima subset (*fold*). Setiap proses iterasi

menggunakan empat subset sebagai data latih dan satu subset sebagai data validasi secara bergantian hingga seluruh bagian mendapatkan giliran sebagai data validasi sebanyak lima kali.

Kinerja model pada setiap iterasi dievaluasi menggunakan nilai *F1-score*. Selanjutnya, rata-rata nilai *F1-score* dari seluruh *fold* dihitung menggunakan Persamaan (3.27).

$$F1\text{-}Score_{mean} = \frac{1}{k}\sum_{i=1}^{k} F1\text{-}Score_i \quad (3.\ 27)$$

Berdasarkan Persamaan (3.27), diperoleh nilai rata-rata *F1-score* yang merepresentasikan performa model secara keseluruhan pada seluruh proses *cross validation*. Semakin tinggi nilai rata-rata *F1-score*, semakin baik kemampuan model dalam menyeimbangkan nilai *precision* dan *recall*.

Selain nilai rata-rata *F1-score*, stabilitas model juga dievaluasi menggunakan nilai standar deviasi. Perhitungan standar deviasi dilakukan menggunakan Persamaan (3.28).

$$(\text{Std} = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(F1\text{-}Score_i - F1\text{-}Score_{mean})^2}) \quad (3.\ 28)$$

Berdasarkan Persamaan (3.28), standar deviasi menggambarkan seberapa besar variasi kinerja model pada tiap *fold*. Semakin kecil nilai standar deviasi, maka semakin stabil dan konsisten performa model di seluruh proses *cross-validation.*

Kombinasi parameter yang menghasilkan nilai *F1-score* tertinggi dipilih sebagai model terbaik. Penggunaan *F1-score* pada penelitian ini sangat krusial untuk menyeimbangkan nilai *precision* dan *recall*, terutama jika terdapat

ketidakseimbangan kelas pada data. Model optimal yang terpilih kemudian diuji pada tahap prediksi menggunakan data uji yang telah dipisahkan di awal.

3.7 Evaluasi Model dan Skenario Pengujian

Evaluasi model dilakukan untuk menilai performa klasifikasi secara menyeluruh dengan menggunakan beberapa metrik yang dihitung berdasarkan *confusion matrix*, yaitu tabel yang membandingkan hasil prediksi model dengan label aktual. *Confusion matrix* terdiri dari empat komponen utama dengan label aktual. Tabel ini terdiri dari empat komponen utama:

1. *True Positive* (TP): Model memprediksi kelas positif (berisiko) dan prediksi tersebut benar.
2. *True Negative* (TN): Model memprediksi kelas negatif (tidak berisiko) dan prediksi tersebut benar.
3. *False Positive* (FP) atau *Type I Error*: Model memprediksi kelas positif tetapi sebenarnya negatif (*false alarm*).
4. *False Negative* (FN) atau *Type II Error*: Model memprediksi kelas negatif tetapi sebenarnya positif (gagal deteksi).

Keempat komponen tersebut membentuk struktur *confusion matrix* yang digunakan sebagai dasar dalam perhitungan berbagai metrik evaluasi model. Bentuk *confusion matrix* yang digunakan pada penelitian ini disajikan pada Tabel 3.22.

Tabel 3.22 Contoh *Confusion Matrix*

	Prediksi Negatif	Prediksi Positif	Total
Aktual Negatif	580 (TN)	40 (FP)	620
Aktual Positif	15 (FN)	97 (TP)	112
Total	595	137	732

Tabel 3.22 adalah contoh ilustrasi nilai acak *confusion matrix* yang digunakan untuk menjelaskan cara model klasifikasi, seperti XGBoost, mengevaluasi hasil prediksi terhadap data aktual. Nilai-nilai yang ditampilkan bukan hasil akhir penelitian, melainkan gambaran umum mengenai posisi *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) dalam perhitungan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*.

Dalam konteks prediksi risiko penyakit jantung, *False Negative* sangat berbahaya karena menunjukkan pasien yang sebenarnya berisiko namun tidak terdeteksi oleh model. Beberapa metrik yang umum digunakan untuk evaluasi model antara lain:

1. Accuracy

Accuracy digunakan untuk menggambarkan tingkat ketepatan model dalam mengklasifikasikan data secara keseluruhan. Nilai ini dihitung dengan membandingkan jumlah prediksi yang benar, yang terdiri dari *True Positive* (TP) dan *True Negative* (TN), terhadap seluruh data pengujian yang mencakup TP, TN, *False Positive* (FP), dan *False Negative* (FN), sebagaimana ditunjukkan pada Persamaan (3.29).

$$Accuracy: \frac{TP + TN}{TP + TN + FP + FN} \quad (3.\ 29)$$

Berdasarkan Persamaan (3.29), sebagai ilustrasi digunakan nilai TP = 97, TN = 580, FP = 40, dan FN = 15 sehingga diperoleh perhitungan sebagai berikut.

$$(Accuracy = \frac{97 + 580}{97 + 580 + 40 + 15} = \frac{677}{732} = 0.92 = 92.49\%)$$

Berdasarkan ilustrasi tersebut, diperoleh nilai *Accuracy* sebesar 92,49%. Contoh perhitungan ini bertujuan untuk menunjukkan cara menghitung metrik *Accuracy* menggunakan *confusion matrix* pada Tabel 3.22. Nilai tersebut bukan merupakan hasil akhir penelitian, sedangkan nilai Accuracy hasil pengujian model akan disajikan pada Bab IV.

2. *Precision*

Precision digunakan untuk menggambarkan tingkat ketepatan model dalam memberikan hasil prediksi positif. Nilai *precision* dihitung dengan cara membandingkan *True Positive* (TP) terhadap seluruh prediksi positif, yang terdiri dari TP dan *False Positive* (FP), sebagaimana ditunjukkan pada Persamaan (3.30).

$$Precision: \frac{TP}{TP+FP} \qquad (3.\ 30)$$

Berdasarkan Persamaan (3.30), sebagai ilustrasi digunakan nilai TP = 97 dan FP = 40 sehingga diperoleh perhitungan sebagai berikut.

$$(Precision = \frac{97}{97+40} = \frac{97}{137} = 0.7080 = 70.80\%)$$

Berdasarkan ilustrasi tersebut, diperoleh nilai *Precision* sebesar 70,80%. Contoh perhitungan ini bertujuan untuk menunjukkan cara menghitung metrik *Precision* menggunakan *confusion matrix* pada Tabel 3.22. Nilai tersebut bukan merupakan hasil akhir penelitian, sedangkan nilai *Precision* hasil evaluasi model akan disajikan pada Bab IV. Precision penting ketika biaya *False Positive* tinggi, Hal ini bertujuan untuk meminimalkan terjadinya *false alarm* yang dapat menimbulkan

kekhawatiran berlebihan ataupun mendorong dilakukannya pemeriksaan medis yang sebenarnya tidak diperlukan.

3. *Recall* (*Sensitivity*)

Recall merupakan ukuran yang menunjukkan kemampuan model dalam mendeteksi seluruh kasus positif yang sebenarnya ada pada data. Pada Persamaan (3.31), nilai recall diperoleh dengan membandingkan *True Positive* (TP) terhadap total data positif aktual, yaitu gabungan antara TP dan *False Negative* (FN).

$$Recall = \frac{TP}{TP+FN} \tag{3. 31}$$

Berdasarkan Persamaan (3.31), sebagai ilustrasi digunakan nilai TP = 97 dan FN = 15 sehingga diperoleh perhitungan sebagai berikut.

$$(Recall = \frac{97}{97+15} = \frac{97}{112} = 0.8661 = 86.61\%)$$

Berdasarkan ilustrasi tersebut, diperoleh nilai *Recall* sebesar 86,61%. Perhitungan ini hanya digunakan untuk menjelaskan cara menghitung metrik *Recall* berdasarkan contoh *confusion matrix* pada Tabel 3.22. Nilai tersebut bukan merupakan hasil evaluasi akhir model, sedangkan nilai *Recall* hasil penelitian akan disajikan pada Bab IV. *Recall* memiliki peran penting dalam bidang medis karena mencerminkan kemampuan model dalam mengidentifikasi pasien yang benar-benar memiliki risiko. Nilai *Recall* yang rendah menunjukkan masih banyak kasus positif yang tidak terdeteksi (*False Negative*), sehingga pasien yang sebenarnya berisiko

dapat tidak memperoleh penanganan atau tindakan preventif yang diperlukan.

4. *F1-Score*

F1-Score merupakan ukuran yang diperoleh dari rata-rata harmonik antara *Precision* dan *Recall*, yang digunakan untuk memberikan keseimbangan antara kedua metrik tersebut sehingga hasil evaluasi model menjadi lebih menyeluruh. Pada Persamaan (3.32), nilai F1-Score dihitung berdasarkan nilai *Precision* dan *Recall* yang telah diperoleh sebelumnya.

$$F1\text{-}Score = 2 \text{ x } \frac{Precision \times Recall}{Precision + Recall} \quad (3.\ 32)$$

Berdasarkan Persamaan (3.32), nilai *F1-Score* dihitung sebagai berikut.

$$(F1\text{-}Score = \frac{2 \times 0.7080 \times 0.8661}{0.7080 + 0.8661})$$

$$(F1\text{-}Score = \frac{2 \times 0.6132}{1.5741} = 2 \times 0.3895 = 0.779 = 77.90\%)$$

Berdasarkan ilustrasi tersebut, diperoleh nilai *F1-Score* sebesar 77,90%. Perhitungan ini hanya bertujuan untuk menunjukkan cara menghitung metrik *F1-Score* menggunakan nilai *Precision* dan *Recall* yang berasal dari contoh *confusion matrix* pada Tabel 3.22. Nilai tersebut bukan merupakan hasil akhir penelitian, sedangkan nilai *F1-Score* hasil evaluasi model pada penelitian ini akan disajikan pada Bab IV. *F1-Score* sangat sesuai digunakan pada imbalanced dataset karena mempertimbangkan nilai *False Positive* dan *False Negative*, sehingga mampu memberikan gambaran performa model yang lebih komprehensif dibandingkan *Accuracy*.

5. AUC-ROC (*Area Under* the ROC *Curve*)

AUC-ROC (*Area Under the Receiver Operating Characteristic Curve*) merupakan metrik yang digunakan untuk mengevaluasi kemampuan model dalam membedakan antara kelas positif dan kelas negatif pada berbagai nilai ambang batas (*threshold*). Kurva ROC dibentuk berdasarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR). Pada Persamaan (3.33), nilai *True Positive Rate* (TPR) dihitung dengan persamaan berikut.

$$(\text{TPR} = \textit{Recall} = \frac{TP}{TP + FN}) \tag{3. 33}$$

Berdasarkan persamaan (3.33), sebagai ilustrasi digunakan nilai TP = 97 dan FN = 15 yang diperoleh dari contoh *confusion matrix* pada Tabel 3.22 sehingga diperoleh perhitungan sebagai berikut.

$$(\text{TPR} = \frac{97}{97 + 15} = \frac{97}{112} = 0.8661 = 86.61\%)$$

Selanjutnya, nilai *False Positive Rate* (FPR) dihitung menggunakan persamaan (3.34). Persamaan tersebut menunjukkan bahwa nilai FPR diperoleh dari perbandingan jumlah *False Positive* (FP) terhadap seluruh data negatif aktual yang terdiri atas FP dan *True Negative* (TN).

$$\text{FPR} = \frac{FP}{FP+TN} \tag{3. 34}$$

Berdasarkan persamaan (3.34), sebagai ilustrasi digunakan nilai FP = 40 dan TN = 580 yang diperoleh dari contoh *confusion matrix* pada Tabel 3.22 sehingga diperoleh hasil sebagai berikut.

$$(\text{FPR} = \frac{40}{40 + 580} = \frac{40}{620} = 0.0645 = 6.45\%)$$

Selain TPR dan FPR, evaluasi ROC juga dapat menggunakan nilai *Specificity* atau *True Negative Rate* yang menunjukkan kemampuan model dalam mengidentifikasi kelas negatif secara benar. Nilai *Specificity* dihitung menggunakan Persamaan (3.35).

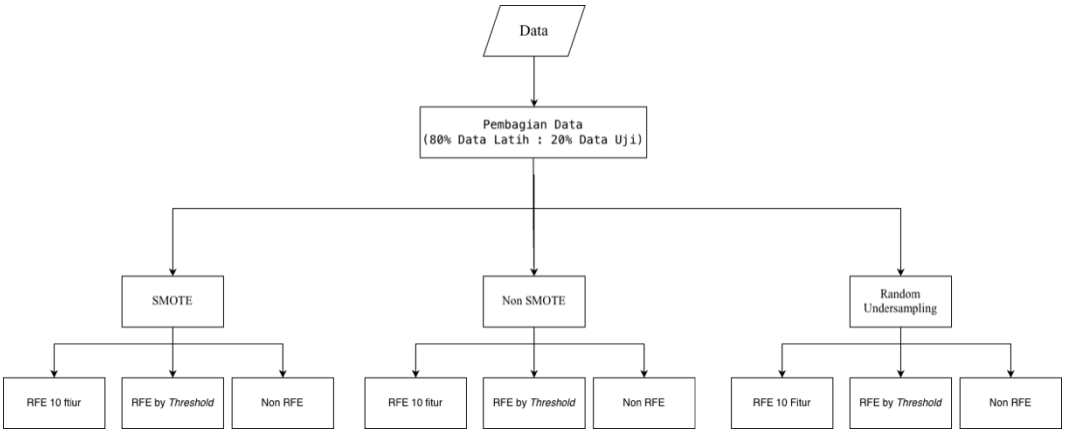
$$\textit{Specificity} = 1 - \text{FPR} \quad (3.\ 35)$$

Berdasarkan Persamaan (3.35), diperoleh nilai *Specificity* sebagai berikut.

$$(\textit{Specificity} = 1 - \text{FPR} = 1 - 0.0645 = 0.9355 = 93.55\%)$$

Nilai AUC (*Area Under the Curve*) merupakan luas area di bawah kurva ROC yang digunakan untuk mengukur kemampuan model dalam membedakan kelas positif dan kelas negatif. Perhitungan TPR, FPR, dan *Specificity* pada bagian ini hanya digunakan sebagai ilustrasi berdasarkan contoh *confusion matrix* pada Tabel 3.22 untuk menjelaskan pembentukan kurva ROC. Nilai-nilai tersebut bukan merupakan hasil akhir penelitian, sedangkan nilai AUC hasil evaluasi model akan disajikan pada Bab IV.

Kemudian pengujian ini disusun ke dalam beberapa skenario yang bertujuan untuk mengevaluasi pengaruh perbedaan komposisi data serta penerapan teknik seleksi fitur terhadap akurasi klasifikasi. Skenario yang dilakukan untuk pengujian pada penelitian ini ditunjukkan oleh Gambar 3.3



Gambar 3.3 Bagan Skenario Pengujian

Sesuai dengan ilustrasi pada Gambar 3.3, tahapan awal penelitian diawali dengan proses pemisahan dataset dengan perbandingan 80:20. Proporsi 80% dialokasikan untuk fase pelatihan model, sementara 20% sisanya berfungsi sebagai data uji. Strategi pembagian ini bertujuan untuk memastikan bahwa model mendapatkan pasokan data yang memadai selama pembelajaran, sekaligus menyediakan dataset yang objektif dan belum pernah dilihat model guna menilai performa akhirnya secara akurat. Selanjutnya, data latih diproses melalui tiga pendekatan penanganan ketidakseimbangan kelas, yaitu SMOTE, tanpa penyeimbangan kelas (*Non Balancing*), dan *Random Undersampling*. Guna mencegah terjadinya *data leakage* yang berpotensi menghasilkan evaluasi bias, penerapan teknik *balancing* secara eksklusif hanya dilakukan pada dataset pelatihan, tidak pada dataset pengujian.

Pada masing-masing pendekatan penyeimbangan kelas tersebut diterapkan empat variasi seleksi fitur. Variasi pertama menggunakan RFE dengan pemilihan 10 fitur terbaik. Variasi kedua menggunakan seleksi fitur berbasis *threshold* sebesar 0,07, sedangkan variasi ketiga menggunakan *threshold* berbasis nilai rata-rata

(*mean*) *feature importance* yang bersifat adaptif terhadap karakteristik data pada setiap skenario. Variasi keempat merupakan kondisi tanpa seleksi fitur (Non-RFE) yang menggunakan seluruh fitur yang tersedia sebagai pembanding. Kombinasi antara tiga metode penyeimbangan kelas dan empat variasi seleksi fitur menghasilkan total 12 kombinasi eksperimen. Kombinasi tersebut terdiri atas empat skenario pada kelompok SMOTE, empat skenario pada kelompok tanpa *balancing*, dan empat skenario pada kelompok *Random Undersampling*. Seluruh kombinasi pengujian digunakan untuk mengevaluasi pengaruh teknik penyeimbangan kelas dan metode seleksi fitur terhadap performa model klasifikasi XGBoost. Untuk lebih jelasnya ditunjukkan pada table 3.23 berikut.

Tabel 3.23 Skenario Pengujian dan Parameternya

Skenario	**Metode *Balancing***	**Seleksi Fitur**	*Hyper Parameter*
1	SMOTE	RFE (10 Fitur Terbaik)	*n_estimators, max_depth, learning_rate (eta), subsample, colsample_bytree, gamma, objective, eval_metric, random_state*
2	SMOTE	*Threshold* 0.07	
		Threshold Mean	
3	SMOTE	Tanpa Seleksi	
4	Tanpa *Balancing*	RFE (10 Fitur Terbaik)	
5	Tanpa *Balancing*	*Threshold* 0.07	
		Threshold Mean	
6	Tanpa *Balancing*	Tanpa Seleksi	
7	*Random Undersampling*	RFE (10 Fitur Terbaik)	
8	*Random Undersampling*	*Threshold* 0.07	
		Threshold Mean	
9	*Random Undersampling*	Tanpa Seleksi	

Berdasarkan Tabel 3.23, penelitian ini menyusun sembilan skenario pengujian utama, dengan tiga skenario yang memiliki dua variasi metode seleksi fitur berbasis *threshold*, yaitu *threshold* 0.07 dan *threshold mean*. Dengan demikian, total kombinasi eksperimen yang diuji adalah dua belas kombinasi

pengujian yang dirancang secara komprehensif untuk mengevaluasi pengaruh komposisi data serta penerapan teknik seleksi fitur terhadap performa klasifikasi algoritma XGBoost. Agar model dapat dievaluasi secara objektif terhadap *unseen data*, pengujian dilakukan dengan menetapkan perbandingan dataset sebesar 80:20. Proporsi 80% digunakan sebagai basis pembelajaran model, sementara 20% sisanya disisihkan sebagai dataset independen untuk menguji performa model terhadap data yang benar-benar baru. Seluruh tahapan *preprocessing*, baik teknik penyeimbangan kelas seperti SMOTE dan *Random Undersampling* maupun seleksi fitur menggunakan RFE, diterapkan secara eksklusif hanya pada data latih untuk menjaga integritas model dari potensi data *leakage*.

Keseluruhan skenario tersebut dikelompokkan berdasarkan tiga metode penyeimbangan kelas, yaitu *SMOTE*, tanpa penyeimbangan (*non-balanced*), serta *Random Undersampling*. Pada setiap kelompok metode penyeimbangan, diterapkan empat variasi seleksi fitur untuk membedah efektivitas masing-masing pendekatan secara mendalam. Variasi pertama menggunakan *RFE* dengan jumlah fitur tetap sebanyak 10 fitur terbaik untuk melihat efektivitas model pada dimensi yang tereduksi secara spesifik. Variasi kedua dan ketiga merupakan pengembangan dari teknik seleksi fitur berbasis ambang batas (*threshold*), di mana variasi kedua menggunakan nilai statis 0,07 untuk mengeliminasi fitur dengan kontribusi kepentingan di bawah 7%, sedangkan variasi ketiga menggunakan ambang batas berbasis rata-rata (*mean*) yang bersifat adaptif terhadap sebaran data di setiap skenario. Variasi keempat adalah kondisi *non-RFE* yang menggunakan seluruh

fitur sebagai kontrol untuk membandingkan apakah penggunaan fitur lengkap atau fitur terpilih yang lebih optimal bagi performa model.

Pada setiap skenario yang dijalankan, model XGBoost tidak langsung digunakan dengan parameter *default*, melainkan melalui proses optimasi *hyperparameter* menggunakan *GridSearchCV* yang terintegrasi dengan *K-Fold Cross-Validation*. Parameter yang dioptimasi meliputi *n_estimators, max_depth, learning_rate (eta), subsample, colsample_bytree,* serta *gamma*. Optimalisasi ini dilakukan dengan tujuan untuk menjamin bahwa model akhir yang dikembangkan tidak sekadar mencapai akurasi tinggi, melainkan juga memiliki ketahanan (*stability*) serta kemampuan generalisasi yang mumpuni saat diterapkan pada data baru. Penilaian kinerja model pada data uji mencakup beberapa parameter utama, yakni *Accuracy, Precision, Recall, F1-Score,* serta *AUC-ROC*. Mengingat penelitian ini berfokus pada ranah medis, perhatian khusus diberikan pada *Recall* dan *F1-Score*. Prioritas ini ditetapkan untuk menekan sekecil mungkin angka *False Negative*, guna menghindari kesalahan fatal di mana pasien yang mengidap penyakit justru diklasifikasikan sebagai sehat oleh model.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil *Processing* Data

Sebelum melangkah ke tahap pengembangan model prediksi, dilakukan fase *data processing* dengan tujuan untuk menyiapkan dan membersihkan dataset agar siap diolah oleh algoritma.

4.1.1 EDA

Langkah awal penelitian dimulai dengan EDA untuk memetakan karakteristik dataset secara mendalam, terutama dalam mendeteksi *missing values* yang dapat merusak akurasi model. Berdasarkan pemeriksaan awal, ditemukan beberapa variabel yang memiliki data tidak lengkap sebagaimana ditunjukkan pada Tabel 4.1.

Tabel 4.1 Tabel Data dengan *Missing Value*

Variabel	Jumlah Kosong	Persentase
Education	105	2.48 %
CigsPerDay	29	0.68 %
BPMeds	53	1.25 %
TotChol	50	1.18 %
BMI	19	0.45 %
HeartRate	1	0.02 %
Glucose	388	9.16 %
Total Sel Kosong	645	-
Total Baris ber-NA	582	13.73 %
Variabel	Jumlah Kosong	Persentase

Analisis pada Tabel 4.1 menunjukkan adanya 645 sel kosong yang tersebar di 582 baris data. Variabel *Glucose* memiliki tingkat kekosongan tertinggi sebesar

9,16% (388 data), jauh melampaui variabel lain seperti *Education* (2,48%) dan *BPMeds* (1,25%). Secara akumulatif, terdapat 13,73% dari total 4.238 baris data yang mengandung nilai kosong (NA). Masalah kualitas data ini menuntut penanganan serius pada tahap *preprocessing* sebelum masuk ke proses pemodelan.

Tahap pemeriksaan data turut mencakup evaluasi distribusi pada kelas target *TenYearCHD* untuk mengidentifikasi perbandingan proporsi antara kelas mayoritas dan minoritas. Hasil analisis distribusi tersebut disajikan pada Tabel 4.2.

Tabel 4.2 Analisis Distribusi Kelas Target (*TenYearCHD*)

TenYearCHD	**Jumlah Data**	**Persentase (%)**
0 (Tidak Risiko)	3.594	84.80 %
1 (Risiko CHD)	644	15.20 %
TOTAL	4.238	100 %

Berdasarkan data pada Tabel 4.2, terlihat adanya ketimpangan distribusi kelas (*class imbalance*) yang mencolok. Kelas 'tidak berisiko' berjumlah 3.594 sampel (84,80%), sementara kelas 'berisiko' hanya mencakup 644 sampel atau sekitar 15,20% dari total dataset. Dominasi kelas mayoritas ini berpotensi menyebabkan model menjadi bias dalam melakukan prediksi. Oleh karena itu, teknik penyeimbangan data dilakukan pada tahap selanjutnya untuk memastikan model dapat mengenali kedua kelas secara adil.

4.1.2 Hasil Data *Cleaning*

Untuk menjaga kualitas dataset, proses data *cleaning* dilakukan dengan menghapus nilai-nilai yang kosong. Pasca-pembersihan, dataset kemudian dipisahkan menjadi bagian *training* dan *testing* untuk memfasilitasi proses evaluasi performa model secara akurat. Tahap ini merupakan fondasi krusial agar dataset siap diproses lebih lanjut menggunakan teknik penyeimbangan data (*oversampling*

atau *undersampling*) demi mengoptimalkan model prediksi yang ditunjukkan dalam Tabel 4.3.

Tabel 4.3 Pembersihan dan *Splitting* Data

Indikator	Jumlah Baris	Proporsi
Data Awal	4238	100%
Data Hilang	582	13,7%
Data Bersih	3656	100% Valid
Data *Training* (80%)	2924	80% *Subset*
Data *Testing* (20%)	732	20% *Subset*

Sesuai Tabel 4.3, dari 4.238 baris data awal, dilakukan penghapusan pada 582 baris yang memiliki nilai kosong, sehingga menyisakan 3.656 baris data bersih. Pemisahan dataset dilakukan dengan proporsi 80:20, yang menghasilkan 2.924 sampel untuk tahap pelatihan model dan 732 sampel yang dialokasikan untuk fase pengujian. Strategi *splitting* ini diterapkan untuk menguji seberapa baik model dapat menggeneralisasi informasi saat menghadapi data baru di luar proses *training*.

4.1.3 Data *Balancing*

Setelah tahap pembersihan dan pembagian dataset, dilakukan prosedur penyeimbangan data pada komponen *training set*. Untuk mengatasi *class imbalance* pada variabel target *TenYearCHD*, diterapkan teknik *Synthetic Minority Over-sampling Technique* (SMOTE). Metode ini bekerja dengan cara menciptakan observasi sintetis pada kelas minoritas dengan memperhatikan kemiripan antar-sampel. Sebagaimana tercantum pada Tabel 4.4, pendekatan ini dipilih agar distribusi kelas menjadi seimbang, sehingga model mampu mengenali pola dari kedua kategori secara lebih adil dan proporsional.

Tabel 4.4 Teknik 1: *Oversampling* (SMOTE)

Kondisi	Kelas 0 (Sehat)	Kelas 1 (Sakit)	Total *Train*
Sebelum SMOTE	2.479	445	2.924
Setelah SMOTE	2.479	2.479	4.958

Pada hasil pengolahan data di Tabel 4.4, distribusi awal data pelatihan menunjukkan ketimpangan signifikan dengan 2.479 data pada kelas 0 (sehat) dan hanya 445 data pada kelas 1 (sakit). Implementasi metode SMOTE berhasil menyetarakan jumlah data kelas minoritas menjadi 2.479 data, sehingga tercipta distribusi kelas yang seimbang. Transformasi ini mengakibatkan volume data pelatihan meningkat dari 2.924 menjadi total 4.958 data. Langkah penyeimbangan ini diambil untuk menekan bias model terhadap kelas mayoritas, sekaligus meningkatkan kapabilitas algoritma dalam mengidentifikasi pola dari kedua kelas secara objektif dan seimbang.

Selain menggunakan teknik *oversampling,* Strategi lain yang diterapkan adalah *undersampling*, di mana jumlah data pada kelas mayoritas dikurangi secara selektif. Hal ini dilakukan untuk menyamakan distribusi antara kedua kelas, sehingga jumlah sampel pada kelas mayoritas dan minoritas menjadi setara. Metode yang digunakan adalah *Random Undersampling*. Sebagaimana terlihat pada Tabel 4.5, metode *Random Undersampling* diterapkan dengan mereduksi data kelas mayoritas secara acak hingga mencapai kesetaraan jumlah dengan kelas minoritas. Pendekatan yang dilakukan pada data latih ini bertujuan untuk meningkatkan keadilan model dalam mempelajari pola dari kedua kategori dan meminimalisir dominasi kelas mayoritas selama fase pelatihan.

Tabel 4.5 Teknik 2: *Undersampling* (*Random Undersampling*)

Kondisi Data	Kelas 0 (Sehat)	Kelas 1 (Sakit)	Total Baris
Data Latih Asli	2.479	445	2.924
Data Latih *Undersampled*	445	445	890
Data Uji (Original)	620	112	732

Berdasarkan Tabel 4.5 pada data latih asli terdapat 2.479 data kelas sehat (kelas 0) dan 445 data kelas sakit (kelas 1), yang menunjukkan adanya ketidakseimbangan kelas. Setelah dilakukan proses *undersampling*, jumlah data pada kelas mayoritas dikurangi secara acak hingga sama dengan kelas minoritas, sehingga masing-masing kelas memiliki 445 data dengan total 890 data latih.

Sementara itu, data uji tidak dilakukan *undersampling* dan tetap menggunakan distribusi asli sebanyak 732 data, yang terdiri dari 620 data sehat dan 112 data sakit, agar evaluasi model mencerminkan kondisi data yang sebenarnya. Teknik *undersampling* ini diterapkan untuk mengatasi *class imbalance* sehingga model dapat mempelajari pola dari kedua kelas secara lebih seimbang.

4.1.4 Seleksi Fitur (RFE)

Guna mengoptimalkan kinerja model, diterapkan metode RFE untuk menyeleksi fitur-fitur yang paling relevan. Melalui metode ini, variabel-variabel yang memberikan dampak paling kuat terhadap target *TenYearCHD* dapat diidentifikasi dengan lebih akurat. Proses RFE dijalankan dengan menggunakan algoritma XGBoost sebagai *base estimator*, di mana model mengevaluasi nilai *feature importance* secara iteratif dan mengeliminasi fitur dengan kontribusi terendah pada setiap tahapan. Penggunaan XGBoost dalam RFE memungkinkan

proses seleksi fitur mempertimbangkan pola non-linear serta interaksi yang kompleks antarvariabel medis. Hasil seleksi fitur kemudian digunakan dalam tiga pendekatan, yaitu mempertahankan 10 fitur terbaik, mempertahankan fitur dengan nilai *feature importance* di atas *threshold* 0,07, dan mempertahankan fitur dengan nilai *feature importance* di atas rata-rata *feature importance*. Untuk mengevaluasi kontribusi jumlah fitur dan komposisi variabel terhadap akurasi klasifikasi penyakit jantung, ketiga pendekatan tersebut diintegrasikan ke dalam berbagai skenario pengujian.

4.2 Hasil Pengujian Model XGBoost

Tahap ini menguji algoritma XGBoost dalam memprediksi risiko penyakit jantung melalui sembilan skenario pengujian utama yang menghasilkan dua belas kombinasi eksperimen. Jumlah dua belas kombinasi tersebut diperoleh karena pada skenario 2, 5, dan 8 terdapat dua variasi sub-skenario berbasis *threshold*, yaitu sub-skenario (a) menggunakan *threshold feature importance* sebesar 0,07 dan sub-skenario (b) menggunakan nilai rata-rata (*mean*) *feature importance*. Pengujian dilakukan untuk menganalisis pengaruh teknik penyeimbangan data, yaitu SMOTE dan *Random Undersampling*, serta berbagai pendekatan seleksi fitur yang meliputi RFE dengan 10 fitur terbaik, seleksi fitur berdasarkan *threshold feature importance* sebesar 0,07, seleksi fitur berdasarkan nilai rata-rata *feature importance*, dan penggunaan seluruh fitur, terhadap performa model. Setiap kombinasi eksperimen dievaluasi menggunakan metrik *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan AUC ROC untuk menentukan konfigurasi terbaik dalam mengklasifikasikan risiko penyakit jantung.

4.2.1 Hasil Uji Coba Skenario 1 (SMOTE dengan RFE)

Pada skenario pertama, proses pemodelan dilakukan dengan menerapkan SMOTE untuk menyeimbangkan data yang dikombinasikan dengan seleksi fitur menggunakan *Recursive Feature Elimination* (RFE) sebelum pelatihan model menggunakan algoritma XGBoost. Seleksi fitur ini bertujuan untuk mengidentifikasi fitur yang paling relevan dalam proses klasifikasi. Melalui metode RFE, setiap fitur diberikan peringkat dan nilai kepentingan, di mana fitur dengan kontribusi tinggi akan dipertahankan, sedangkan fitur dengan kontribusi rendah akan dieliminasi. Hasil seleksi fitur menggunakan metode RFE yang menghasilkan 10 fitur terbaik dapat dilihat pada Tabel 4.6.

Tabel 4.6 Hasil RFE 10 Fitur

Fitur	***Ranking***	***Importance***	***Status***
prevalentHyp	1	0.1434	*RETAINED*
age	2	0.1072	*RETAINED*
glucose	3	0.0744	*RETAINED*
male	4	0.0725	*RETAINED*
currentSmoker	5	0.0708	*RETAINED*
cigsPerDay	6	0.0688	*RETAINED*
sysBP	7	0.0666	*RETAINED*
education	8	0.0639	*RETAINED*
totChol	9	0.0625	*RETAINED*
diaBP	10	0.0622	*RETAINED*
BPMeds	11	0.0584	*ELIMINATED*
BMI	12	0.0558	*ELIMINATED*
heartRate	13	0.0545	*ELIMINATED*
diabetes	14	0.0246	*ELIMINATED*
prevalentStroke	15	0.0145	*ELIMINATED*

Berdasarkan Tabel 4.6 hasil seleksi fitur, diperoleh 10 fitur yang dipertahankan dan 5 fitur yang dieliminasi. Fitur dengan nilai *importance* tertinggi seperti *prevalentHyp* (0.1434), diikuti oleh *age* (0.1072) dan *glucose* (0.0744), yang menunjukkan bahwa fitur-fitur tersebut memiliki kontribusi lebih besar dalam proses prediksi risiko penyakit jantung. Sementara itu, fitur *BPMeds*, BMI,

heartRate, diabetes, dan *prevalentStroke* dieliminasi karena memiliki tingkat kepentingan yang relatif lebih rendah terhadap model.

Untuk mencapai hasil klasifikasi yang lebih superior, penelitian ini menerapkan seleksi fitur guna mencari kombinasi atribut yang paling optimal sebagai *input* bagi model XGBoost. Oleh karena itu, pemilihan fitur didasarkan pada kontribusinya terhadap performa model pada dataset yang digunakan. Dalam konteks kesehatan, suatu variabel tetap dapat memiliki relevansi klinis meskipun menunjukkan nilai *importance* yang relatif rendah pada model *machine learning*. Dengan demikian, fitur yang dieliminasi oleh metode *Recursive Feature Elimination*, seperti *BPMeds*, *BMI*, *heartRate*, *diabetes*, dan *prevalentStroke*, tidak dapat disimpulkan sebagai faktor yang tidak berhubungan dengan risiko penyakit jantung. Hasil eliminasi tersebut hanya menunjukkan bahwa kontribusi fitur-fitur tersebut terhadap peningkatan performa model pada penelitian ini relatif lebih rendah dibandingkan fitur lainnya. Oleh karena itu, hasil seleksi fitur dalam penelitian ini perlu dipahami sebagai bagian dari upaya optimasi model klasifikasi dan tidak dimaksudkan untuk menggantikan pertimbangan klinis maupun mengabaikan faktor-faktor yang digunakan dalam proses diagnosis medis.

Rangkaian analisis diteruskan dengan prosedur *oversampling* guna menyeimbangkan distribusi kelas, yang dilakukan segera setelah proses seleksi fitur selesai dijalankan. Untuk mencapai keseimbangan distribusi data, teknik SMOTE diterapkan dalam studi ini guna menggenerasi tambahan sampel pada kelas minoritas. Proses *oversampling* ini dilakukan pada data pelatihan (*training data*) untuk mengurangi ketidakseimbangan kelas yang sebelumnya ditemukan pada

dataset. Hasil penerapan metode SMOTE terhadap data pelatihan dapat dilihat pada Tabel 4.7.

Tabel 4.7 Hasil SMOTE

Kategori	Latih Asli	Latih SMOTE	Uji Original
Sehat (Kelas 0)	2479	2479	620
Sakit (Kelas 1)	445	2479	112
Total Keseluruhan	2924	4958	732

Sebagaimana tercantum pada Tabel 4.7, terjadi peningkatan jumlah sampel pada kelas minoritas pasca-implementasi prosedur *oversampling*. Sebelum penerapan SMOTE, jumlah data pada kelas minoritas masih jauh lebih sedikit dibandingkan kelas mayoritas. Setelah proses seleksi fitur dan penyeimbangan data selesai dilakukan, tahap selanjutnya adalah melakukan optimasi parameter model menggunakan teknik *hyperparameter tuning*. Proses ini bertujuan untuk memperoleh kombinasi parameter terbaik yang dapat meningkatkan performa model klasifikasi. Pada penelitian ini, proses optimasi parameter dilakukan menggunakan metode *GridSearchCV* dengan pendekatan 5-*fold cross validation*.

Metode *GridSearchCV* bekerja dengan cara menguji berbagai kombinasi parameter secara sistematis untuk menemukan konfigurasi parameter yang menghasilkan performa model terbaik. Dalam proses pencarian ini dilakukan pengujian terhadap 729 kombinasi parameter, sehingga dengan penerapan 5-*fold cross validation* total proses pelatihan model yang dilakukan adalah sebanyak 3645 proses *fitting*. Model yang digunakan dalam proses optimasi ini adalah algoritma XGBoost. Hasil parameter terbaik yang diperoleh dari proses *hyperparameter tuning* dapat dilihat pada Tabel 4.8.

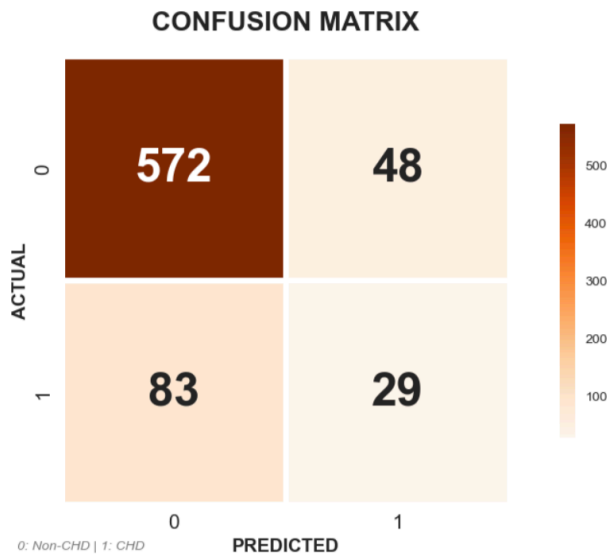
Tabel 4.8 Hasil Seleksi *Hyperparameter Tunning*

Parameter	Nilai Optimal
Learning Rate	0.3
Max Depth	7
N Estimators	200
Subsample	0.8
Colsample	0.9
Gamma	0

Berdasarkan Tabel 4.8, proses *hyperparameter tuning* menghasilkan beberapa parameter optimal yang digunakan dalam model. Parameter *learning rate* sebesar 0.3 mengatur kecepatan model dalam mempelajari pola dari data. Parameter *max depth* sebesar 7 menunjukkan kedalaman maksimum pohon keputusan yang digunakan untuk menangkap pola yang lebih kompleks. Selanjutnya, *n-estimators* sebanyak 200 menunjukkan jumlah pohon yang dibangun dalam proses *boosting*. Parameter *subsample* sebesar 0.8 dan *colsample* sebesar 0.9 digunakan untuk mengontrol proporsi data dan fitur yang digunakan pada setiap iterasi guna mengurangi risiko *overfitting*. Sementara itu, gamma sebesar 0 menunjukkan bahwa tidak ada batas minimum pengurangan *loss* yang diperlukan untuk melakukan pembagian *node* pada pohon keputusan.

Setelah proses pelatihan model selesai, tahap selanjutnya adalah melakukan evaluasi performa model untuk mengetahui kemampuan model dalam melakukan klasifikasi terhadap data uji. Evaluasi dilakukan menggunakan beberapa metrik kinerja yaitu *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan ROC-AUC, yang memberikan gambaran menyeluruh mengenai kemampuan model dalam membedakan antara kelas responden yang berisiko dan tidak berisiko penyakit jantung. Selain itu, performa model juga dianalisis menggunakan *Confusion Matrix*

untuk melihat jumlah prediksi yang benar maupun salah pada masing-masing kelas ditunjukkan pada Gambar 4.1.



Gambar 4.1 *Confusion Matrix* Skenario 1

Berdasarkan hasil *confusion matrix* pada Gambar 4.1, model menunjukkan performa yang cukup baik dalam mengklasifikasikan kelas mayoritas yaitu sehat (kelas 0) dengan 572 data *True Negative*. Model juga berhasil mendeteksi 29 data *True Positive* sebagai kasus berisiko CHD. Namun, masih terdapat 83 data *False Negative*, yaitu kasus berisiko CHD yang diprediksi sebagai sehat, sehingga menunjukkan bahwa kemampuan model dalam mendeteksi kelas positif masih perlu ditingkatkan. Selain itu, terdapat 48 data *False Positive*, yaitu data sehat yang diprediksi sebagai berisiko CHD. Secara keseluruhan, model memiliki performa yang baik pada kelas mayoritas, tetapi masih perlu dioptimalkan untuk meningkatkan deteksi pada kelas minoritas.

Hasil evaluasi performa model dilakukan untuk mengetahui kemampuan model dalam melakukan klasifikasi terhadap data yang digunakan. Evaluasi ini menggunakan beberapa metrik pengukuran yaitu *accuracy*, *precision*, *recall*, *F1-score*, dan ROC-AUC. Masing-masing metrik digunakan untuk memberikan gambaran yang lebih komprehensif mengenai kinerja model, terutama dalam mendeteksi kelas positif pada dataset. Hasil prediksi model disajikan pada Tabel 4.9.

Tabel 4.9 Hasil Prediksi Model

Metric	***Value***
Accuracy	82.10%
Precision	37.66%
Recall	25.89%
F1-Score	30.69%
ROC-AUC	63.73%

Berdasarkan Tabel 4.9, model memperoleh nilai *Accuracy* sebesar 82.10% yang menunjukkan bahwa sebagian besar data berhasil diklasifikasikan dengan benar secara keseluruhan. Nilai *Precision* sebesar 37.66% menunjukkan bahwa dari seluruh data yang diprediksi sebagai kelas positif, hanya 37.66% yang merupakan prediksi benar, sementara nilai *Recall* sebesar 25.89% mengindikasikan bahwa model baru mampu mengidentifikasi sekitar seperempat dari total data yang benar-benar termasuk dalam kelas positif. Nilai *F1-Score* sebesar 30.69% menggambarkan keseimbangan yang masih rendah antara *precision* dan *recall* dalam mengukur performa model terhadap kelas positif, sedangkan nilai ROC-AUC sebesar 63.73% menunjukkan kemampuan model yang cukup dalam membedakan antara kelas positif dan negatif. Secara keseluruhan, meskipun model

memiliki tingkat akurasi yang cukup tinggi, kemampuan dalam mendeteksi kelas positif masih perlu ditingkatkan secara signifikan, terutama pada nilai *recall* dan *F1-score*, agar performa model menjadi lebih optimal dan seimbang.

4.2.2 Hasil Uji Coba Skenario 2 (SMOTE dengan RFE *by Threshold*)

Pada Skenario 2, pengujian dilakukan dengan mengombinasikan teknik penyeimbangan data SMOTE dan metode seleksi fitur berbasis *feature importance* dengan pendekatan *threshold*. Tujuan dari skenario ini adalah untuk mengevaluasi performa model XGBoost dalam menangani data yang telah diseimbangkan serta direduksi dimensinya berdasarkan nilai kontribusi fitur yang melebihi ambang batas tertentu. Nilai *threshold* sebesar 0,07 ditetapkan berdasarkan observasi nilai *feature importance* pada 10 fitur teratas di Skenario 1 dan digunakan sebagai acuan nyata untuk mengurangi jumlah fitur pada pengujian lanjutan. Skenario ini dibagi menjadi dua sub-skenario, yaitu Skenario 2a yang menggunakan *threshold* sebesar 0,07 dan Skenario 2b yang menggunakan *threshold* berdasarkan nilai rata-rata *feature importance*.

Hasil pengujian pada Skenario 2a (SMOTE dengan *feature selection* berbasis *threshold* 0,07) disajikan pada bagian berikut.

a. Hasil Uji Coba Skenario 2a (SMOTE dengan RFE *by Threshold* 0.07)

Pada skenario ini, pemodelan difokuskan pada penanganan ketidakseimbangan dataset melalui teknik *SMOTE*, yang kemudian disandingkan dengan proses seleksi fitur berbasis ambang batas (*threshold*).

Pemilihan ambang batas sebesar 0,07 didasarkan pada analisis distribusi *gain score* seluruh variabel, di mana nilai tersebut merupakan titik potong optimal yang mampu memisahkan fitur dengan kontribusi informatif tinggi dari fitur yang cenderung bersifat *noise*. Pendekatan ini diterapkan dengan prinsip bahwa fitur yang memiliki nilai kepentingan (*gain score*) di atas atau sama dengan 0,07 akan dipertahankan (*retained*), sedangkan fitur yang berada di bawah nilai tersebut akan dieliminasi (*eliminated*).

Langkah ini bertujuan untuk menyederhanakan kompleksitas model dengan mereduksi dimensi fitur secara signifikan. Dengan hanya menyertakan variabel yang memiliki daya prediksi kuat, model XGBoost yang dibangun diharapkan tidak hanya memiliki akurasi yang lebih tajam, tetapi juga waktu pemrosesan yang lebih efisien karena berkurangnya beban komputasi. Reduksi fitur ini sangat krusial untuk mencegah terjadinya *overfitting* yang sering dipicu oleh penggunaan variabel input yang terlalu banyak namun kurang relevan dengan target klasifikasi. Tabel 4.10 berikut menyajikan hasil seleksi fitur berdasarkan ambang batas 0,07.

Tabel 4.10 Hasil RFE by *Threshold* 0.07

Ranking	***Feature Name***	**Status**	***Importance* (*Gain Score*)**
1	*prevalentHyp*	*RETAINED*	0.1434
2	*age*	*RETAINED*	0.1072
3	*glucose*	*RETAINED*	0.0744
4	*male*	*RETAINED*	0.0725
5	*currentSmoker*	*RETAINED*	0.0708
6	*cigsPerDay*	*ELIMINATED*	0.0688
7	*sysBP*	*ELIMINATED*	0.0666
8	*education*	*ELIMINATED*	0.0639
9	*totChol*	*ELIMINATED*	0.0625
10	*diaBP*	*ELIMINATED*	0.0622
11	*BPMeds*	*ELIMINATED*	0.0584
12	*BMI*	*ELIMINATED*	0.0558
13	*heartRate*	*ELIMINATED*	0.0545

Ranking	***Feature Name***	**Status**	***Importance* (*Gain Score*)**
14	*diabetes*	*ELIMINATED*	0.0246
15	*prevalentStroke*	*ELIMINATED*	0.0145

Berdasarkan data pada Tabel 4.10, terlihat bahwa model mempertahankan lima fitur utama yang memiliki kontribusi paling dominan, dengan *prevalentHyp* menempati urutan pertama (skor 0.1434). Selain itu, variabel *age* (0.1072) dan *glucose* (0.0744) juga tercatat sebagai indikator kunci yang memengaruhi hasil prediksi risiko penyakit jantung. Sebaliknya, fitur-fitur seperti *BPMeds*, BMI, *heartRate*, diabetes, hingga *prevalentStroke* tereliminasi karena dianggap kurang memberikan kontribusi informatif yang signifikan bagi performa model.

Setelah fitur-fitur yang relevan terpilih, langkah berikutnya adalah menyeimbangkan distribusi kelas melalui teknik oversampling menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*). Metode ini dijalankan khusus pada data latih (*training set*) untuk memperbanyak jumlah sampel pada kelas minoritas. Langkah ini krusial dilakukan untuk meminimalisir bias model terhadap kelas mayoritas, sehingga kemampuan prediksi pada kelas target menjadi lebih objektif. Ringkasan hasil penyeimbangan data menggunakan SMOTE tersebut tercantum pada Tabel 4.11.

Tabel 4.11 Hasil SMOTE

Kategori	**Latih Asli**	**Latih SMOTE**	**Uji Original**
Sehat (Kelas 0)	2479	2479	620
Sakit (Kelas 1)	445	2479	112
Total Keseluruhan	2924	4958	732

Berdasarkan Tabel 4.11 jumlah data pada kelas minoritas mengalami peningkatan setelah dilakukan proses *oversampling*. Sebelum penerapan SMOTE, jumlah data pada kelas minoritas masih jauh lebih sedikit dibandingkan kelas mayoritas. Setelah dilakukan *oversampling* dengan SMOTE, distribusi kelas pada data latih menjadi seimbang. Peningkatan jumlah sampel pada kelas minoritas ini sangat krusial, mengingat sebelumnya terdapat ketimpangan jumlah data yang cukup signifikan dibandingkan kelas mayoritas.

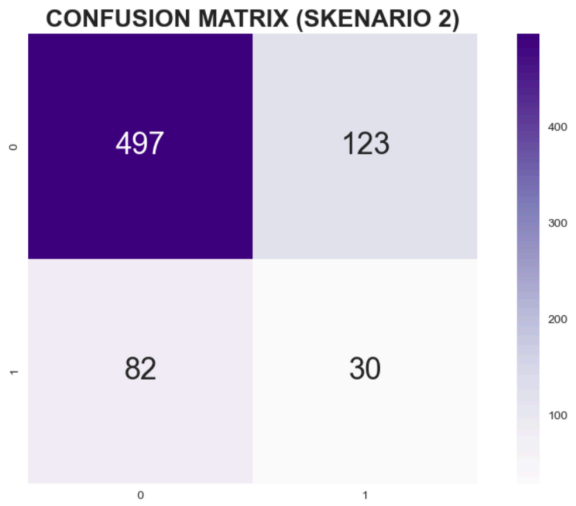
Setelah dataset siap, langkah berikutnya adalah optimasi model melalui *hyperparameter tuning* guna menemukan konfigurasi parameter yang mampu menghasilkan performa klasifikasi paling optimal. Proses ini diimplementasikan menggunakan teknik *GridSearchCV* dengan skema 5-*fold cross-validation*. Metode ini bekerja dengan memvalidasi performa model secara sistematis melalui berbagai kombinasi parameter. Dalam penelitian ini, terdapat 729 kombinasi parameter yang diuji, sehingga dengan penerapan 5 *fold cross validation*, total terdapat 3.645 proses *fitting* model XGBoost yang dilakukan. Detail parameter terbaik yang diperoleh dari proses ini disajikan pada Tabel 4.12.

Tabel 4.12 Hasil Seleksi *Hyperparameter Tunning*

Parameter	**Nilai Optimal**
Learning Rate	0.3
Max Depth	7
N Estimators	200
Subsample	0.9
Colsample	1.0
Gamma	0

Berdasarkan Tabel 4.12, diperoleh konfigurasi parameter yang menjadi acuan utama dalam pelatihan model akhir. Nilai *learning rate* sebesar 0.3 menentukan laju pembelajaran model dalam menyerap pola data. Kedalaman pohon (*max depth*) yang diset pada angka 7 memungkinkan model untuk menangkap pola kompleks namun tetap menjaga generalisasi, didukung oleh 200 *n_estimators* yang membangun pohon keputusan secara bertahap. Parameter *subsample* sebesar 0.9 dan *colsample* sebesar 1.0 berperan penting dalam memitigasi risiko *overfitting* dengan mengontrol proporsi data dan fitur pada tiap iterasi. Sementara itu, nilai *gamma* sebesar 0 mengindikasikan bahwa model tidak menetapkan batasan minimum *loss reduction* untuk setiap *node split*, memberikan keleluasaan lebih pada proses pembangunan pohon.

Tahap akhir adalah evaluasi model untuk mengukur efektivitasnya dalam mengklasifikasikan data uji. Kinerja model dianalisis menggunakan metrik evaluasi komprehensif, yaitu *Accuracy, Precision, Recall, F1-Score,* dan *ROC-AUC*. Metrik-metrik ini memberikan gambaran menyeluruh mengenai kemampuan model dalam membedakan responden yang berisiko terkena penyakit jantung dengan mereka yang tidak. Selain itu, performa model juga ditinjau secara mendalam melalui *Confusion Matrix* untuk memetakan jumlah prediksi yang benar maupun salah, yakni *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang ditunjukkan pada Gambar 4.2.



Gambar 4.2 *Confusion Matrix* Skenario 2a

Berdasarkan hasil *Confusion Matrix* pada Gambar 4.2, model menunjukkan performa yang cukup stabil dalam mengklasifikasikan kelas mayoritas (kelas 0 atau sehat). Model berhasil melakukan prediksi benar sebagai *True Negative* (TN) sebanyak 497 data. Sementara itu, untuk kelas minoritas (kelas 1 atau berisiko CHD), model berhasil mendeteksi sebanyak 30 data sebagai *True Positive* (TP).

Namun, hasil evaluasi juga menunjukkan adanya kesalahan prediksi, di mana terdapat 82 data yang termasuk *False Negative* (FN) yakni responden berisiko CHD namun diprediksi sehat serta 123 data *False Positive* (FP), yaitu responden sehat yang justru diprediksi berisiko CHD. Secara keseluruhan, angka *False Negative* yang cukup tinggi mengindikasikan bahwa kemampuan model dalam mendeteksi kelas positif masih perlu dioptimalkan agar lebih sensitif terhadap kasus berisiko penyakit jantung.

Hasil evaluasi performa model dilakukan untuk mengetahui kemampuan model dalam melakukan klasifikasi terhadap data yang digunakan. Evaluasi ini menggunakan beberapa metrik pengukuran, yaitu *accuracy*, *precision*, *recall*, *F1-score*,dan ROC-AUC. Masing-masing metrik digunakan untuk memberikan gambaran yang lebih komprehensif mengenai kinerja model, terutama dalam meminimalisir kesalahan klasifikasi pada kelas minoritas yang menjadi fokus utama dalam penelitian ini. Hasil prediksi model disajikan pada Tabel 4.13.

Tabel 4.13 Hasil Prediksi Model

Metric	***Value***
Accuracy	71.99%
Precision	19.61%
Recall	26.79%
F1-Score	22.64%
ROC-AUC	59.34%

Berdasarkan Tabel 4.13, model menghasilkan nilai *Accuracy* sebesar 71.99%, yang mengindikasikan bahwa proporsi klasifikasi benar secara keseluruhan berada pada tingkat yang cukup baik. Namun, jika ditinjau dari metrik untuk kelas minoritas, nilai *Precision* tercatat sebesar 19.61%, yang berarti dari seluruh prediksi positif yang dihasilkan model, hanya sekitar 19.61% yang benar-benar merupakan kasus positif (pasien berisiko).

Sementara itu, nilai *Recall* sebesar 26.79% menunjukkan bahwa model baru mampu mengidentifikasi sekitar 26.79% dari total kasus positif yang sebenarnya ada dalam dataset. Nilai *F1-Score* yang berada pada angka 22.64% menggambarkan bahwa keseimbangan antara *precision* dan *recall*

masih relatif rendah, hal ini menegaskan perlunya peningkatan performa pada model dalam mengenali kelas positif. Adapun nilai ROC-AUC sebesar 59.34% mencerminkan kemampuan model yang masih terbatas dalam membedakan antara responden yang berisiko dan tidak berisiko penyakit serangan jantung. Secara keseluruhan, performa model pada Skenario 2a ini menunjukkan bahwa meskipun akurasi secara umum cukup stabil, efektivitas model dalam mendeteksi kelas minoritas masih memerlukan optimasi lebih lanjut pada skenario berikutnya.

Berdasarkan hasil tersebut, dilakukan pengujian lanjutan pada Skenario 2b untuk mengevaluasi performa model dengan pendekatan *feature selection* menggunakan nilai rata-rata *feature importance*.

b. Hasil Uji Coba Skenario 2b (SMOTE dengan RFE *by Threshold Mean*)

Pada skenario ini, pemodelan difokuskan pada penanganan ketidakseimbangan dataset melalui Teknik SMOTE, yang kemudian disandingkan dengan proses seleksi fitur berbasis ambang batas rata-rata (*threshold mean*). Berbeda dengan penggunaan nilai statis, ambang batas mean bersifat adaptif, di mana titik potong ditentukan berdasarkan nilai rata-rata dari seluruh gain score fitur yang ada. Fitur yang memiliki nilai kepentingan (*importance*) di atas atau sama dengan rata-rata akan dipertahankan (*retained*), sedangkan fitur yang berada di bawah nilai rata-rata akan dieliminasi (*eliminated*). Pendekatan ini bertujuan untuk menyaring variabel yang benar-benar memberikan kontribusi di atas

performa rata-rata fitur dalam model. Tabel 4.14 menyajikan hasil seleksi fitur berdasarkan ambang batas *mean* tersebut.

Tabel 4.14 Hasil RFE *by Threshold Mean*

No	Fitur	*Importance*	Status
1	*prevalentHyp*	0.1434	*Retained*
2	*age*	0.1072	*Retained*
3	*glucose*	0.0744	*Retained*
4	*male*	0.0725	*Retained*
5	*currentSmoker*	0.0708	*Retained*
6	*cigsPerDay*	0.0688	*Retained*
7	*sysBP*	0.0666	*Eliminated*
8	*education*	0.0639	*Eliminated*
9	*totChol*	0.0625	*Eliminated*
10	*diaBP*	0.0622	*Eliminated*
11	*BPMeds*	0.0584	*Eliminated*
12	*BMI*	0.0558	*Eliminated*
13	*heartRate*	0.0545	*Eliminated*
14	diabetes	0.0246	*Eliminated*
15	*prevalentStroke*	0.0145	*Eliminated*

Berdasarkan data pada Tabel 4.14, terlihat bahwa model mempertahankan enam fitur utama yang memiliki kontribusi di atas nilai ambang batas rata-rata. Fitur *prevalentHyp* menjadi kontributor utama dengan skor 0,1434, diikuti oleh *age* (0,1072) dan *glucose* (0,0744). Penggunaan *threshold mean* secara otomatis mengeliminasi variabel-variabel yang memiliki skor kepentingan di bawah rata-rata, seperti *sysBP*, *education*, hingga *prevalentStroke*. Keunggulan dari metode ini adalah kemampuannya untuk beradaptasi dengan distribusi data yang ada, sehingga memastikan bahwa hanya variabel yang benar-benar memiliki performa di atas rata-rata yang dilibatkan dalam proses klasifikasi XGBoost, guna mengoptimalkan ketepatan prediksi pada data uji.

Setelah fitur-fitur yang relevan terpilih, langkah berikutnya adalah menyeimbangkan distribusi kelas melalui Teknik *oversampling*

menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*). Metode ini dijalankan khusus pada data latih (*training set*) untuk memperbanyak jumlah sampel pada kelas minoritas. Langkah ini krusial dilakukan untuk meminimalisir bias model terhadap kelas mayoritas, sehingga kemampuan prediksi pada kelas target menjadi lebih objektif. Ringkasan hasil penyeimbangan data menggunakan SMOTE tersebut tercantum pada Tabel 4.15.

Tabel 4.15 Hasil SMOTE

Kategori	Latih Asli	Latih SMOTE	Uji Original
Sehat (Kelas 0)	2479	2479	620
Sakit (Kelas 1)	445	2479	112
Total Keseluruhan	2924	4958	732

Berdasarkan Tabel 4.15 jumlah data pada kelas minoritas mengalami peningkatan setelah dilakukan proses *oversampling*. Sebelum penerapan SMOTE, jumlah data pada kelas minoritas masih jauh lebih sedikit dibandingkan kelas mayoritas.

Setelah dilakukan *oversampling* dengan SMOTE, distribusi kelas pada data latih menjadi seimbang. Peningkatan jumlah sampel pada kelas minoritas ini sangat krusial, mengingat sebelumnya terdapat ketimpangan jumlah data yang cukup signifikan dibandingkan kelas mayoritas.

Setelah *dataset* siap, langkah berikutnya adalah optimasi model melalui *hyperparameter tuning* guna menemukan konfigurasi parameter yang mampu menghasilkan performa klasifikasi paling optimal. Proses ini diimplementasikan menggunakan teknik *GridSearchCV* dengan skema 5-*fold cross-validation*. Metode ini bekerja dengan memvalidasi performa

model secara sistematis melalui berbagai kombinasi parameter. Dalam penelitian ini, terdapat 729 kombinasi parameter yang diuji, sehingga dengan penerapan 5-*fold cross-validation*, total terdapat 3.645 proses *fitting* model XGBoost yang dilakukan. Detail parameter terbaik yang diperoleh dari proses ini disajikan pada Tabel 4.16.

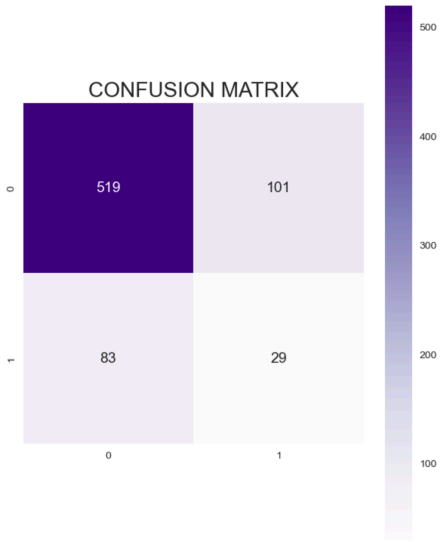
Tabel 4.16 Hasil Seleksi *Hyperparameter Tunning*

Parameter	Nilai Optimal
Learning Rate	0.3
Max Depth	7
N Estimators	200
Subsample	0.9
Colsample	1.0
Gamma	0.1

Berdasarkan Tabel 4.16, diperoleh konfigurasi parameter yang menjadi acuan utama dalam pelatihan model akhir. Nilai *learning rate* sebesar 0.3 menentukan laju pembelajaran model dalam menyerap pola data. Kedalaman pohon (*max depth*) yang diset pada angka 7 memungkinkan model untuk menangkap pola kompleks namun tetap menjaga generalisasi, didukung oleh 200 *n_estimators* yang membangun pohon keputusan secara bertahap. Parameter *subsample* sebesar 0.9 dan *colsample* sebesar 1.0 berperan penting dalam memitigasi risiko *overfitting* dengan mengontrol proporsi data dan fitur pada tiap iterasi. Sementara itu, nilai *gamma* sebesar 0.1 mengindikasikan bahwa model tidak menetapkan batasan minimum *loss reduction* untuk setiap *node split*, memberikan keleluasaan lebih pada proses pembangunan pohon.

Tahap akhir adalah evaluasi model untuk mengukur efektivitasnya dalam mengklasifikasikan data uji. Kinerja model dianalisis menggunakan

metrik evaluasi komprehensif, yaitu *Accuracy, Precision, Recall, F1-Score,* dan *ROC-AUC*. Metrik-metrik ini memberikan gambaran menyeluruh mengenai kemampuan model dalam membedakan responden yang berisiko terkena penyakit jantung dengan mereka yang tidak. Selain itu, performa model juga ditinjau secara mendalam melalui *Confusion Matrix* untuk memetakan jumlah prediksi yang benar maupun salah, yakni *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) sebagaimana ditunjukkan pada Gambar 4.3.



Gambar 4.3 *Confusion Matrix* Skenario 2b

Berdasarkan hasil *Confusion Matrix* pada Gambar 4.3, model menunjukkan performa yang cukup stabil dalam mengklasifikasikan kelas mayoritas (kelas 0 atau sehat). Model berhasil melakukan prediksi benar sebagai *True Negative* (TN) sebanyak 519 data. Sementara itu, untuk kelas minoritas (kelas 1 atau berisiko CHD), model berhasil mendeteksi sebanyak 29 data sebagai *True Positive* (TP).

Namun, hasil evaluasi juga menunjukkan adanya kesalahan prediksi, di mana terdapat 83 data yang termasuk *False Negative* (FN) yakni responden berisiko CHD namun diprediksi sehat serta 101 data *False Positive* (FP), yaitu responden sehat yang justru diprediksi berisiko CHD. Secara keseluruhan, angka *False Negative* yang cukup tinggi mengindikasikan bahwa kemampuan model dalam mendeteksi kelas positif masih perlu dioptimalkan agar lebih sensitif terhadap kasus berisiko penyakit jantung.

Hasil evaluasi performa model dilakukan untuk mengetahui kemampuan model dalam melakukan klasifikasi terhadap data yang digunakan. Evaluasi ini menggunakan beberapa metrik pengukuran, yaitu *accuracy*, *precision*, *recall*, *F1-score*,dan ROC-AUC. Masing-masing metrik digunakan untuk memberikan gambaran yang lebih komprehensif mengenai kinerja model, terutama dalam meminimalisir kesalahan klasifikasi pada kelas minoritas yang menjadi fokus utama dalam penelitian ini. Hasil prediksi model disajikan pada Tabel 4.17.

Tabel 4.17 Hasil Prediksi Model

Metric	***Value***
Accuracy	74.86%
Precision	22.31%
Recall	25.89%
F1-Score	23.97%
ROC-AUC	56.31%

Berdasarkan Tabel 4.17, model menghasilkan nilai *Accuracy* sebesar 74.86%, yang mengindikasikan bahwa proporsi klasifikasi benar secara keseluruhan berada pada tingkat yang cukup baik. Namun, jika

ditinjau dari metrik untuk kelas minoritas, nilai *Precision* tercatat sebesar 22.31%.

Sementara itu, nilai *Recall* sebesar 25.89% menunjukkan bahwa model baru mampu mengidentifikasi sekitar a% dari total kasus positif yang sebenarnya ada dalam dataset. Nilai *F1-Score* yang berada pada angka 23.97% menggambarkan bahwa keseimbangan antara precision dan recall masih relatif rendah, hal ini menegaskan perlunya peningkatan performa pada model dalam mengenali kelas positif. Adapun nilai ROC-AUC sebesar 56.31% mencerminkan kemampuan model yang masih terbatas dalam membedakan antara responden yang berisiko dan tidak berisiko penyakit jantung. Secara keseluruhan, performa model pada Skenario 2 ini menunjukkan bahwa meskipun akurasi secara umum cukup stabil, efektivitas model dalam mendeteksi kelas minoritas masih memerlukan optimasi lebih lanjut pada skenario berikutnya.

4.2.3 Hasil Uji Coba Skenario 3 (SMOTE Non RFE)

Berbeda dengan skenario sebelumnya, pada skenario ketiga proses pelatihan model dilakukan tanpa melibatkan seleksi fitur melalui *Recursive Feature Elimination* (RFE). Artinya, seluruh fitur yang tersedia dalam dataset digunakan secara utuh untuk melatih model. Pendekatan ini digunakan sebagai pembanding untuk mengevaluasi apakah pengurangan jumlah fitur melalui RFE benar-benar memberikan kontribusi terhadap peningkatan kinerja model atau justru berpotensi menghilangkan informasi penting. Dalam konteks data medis, penggunaan seluruh atribut juga sering dipertimbangkan untuk meminimalkan risiko hilangnya

informasi klinis yang relevan. Data latih yang digunakan tetap melalui proses penyeimbangan kelas menggunakan SMOTE guna mengatasi ketimpangan distribusi antar kelas.

Setelah dataset siap dengan seluruh fitur yang disertakan, langkah selanjutnya adalah melakukan *hyperparameter tuning* pada model XGBoost. Optimasi ini dijalankan menggunakan metode *GridSearchCV* dengan pendekatan 5-*fold cross-validation* untuk mencari konfigurasi parameter paling optimal dalam meningkatkan akurasi klasifikasi. Hasil kombinasi parameter terbaik yang diperoleh dari proses tersebut disajikan pada Tabel 4.18.

Tabel 4.18 Hasil Seleksi *Hyperparameter Tunning*

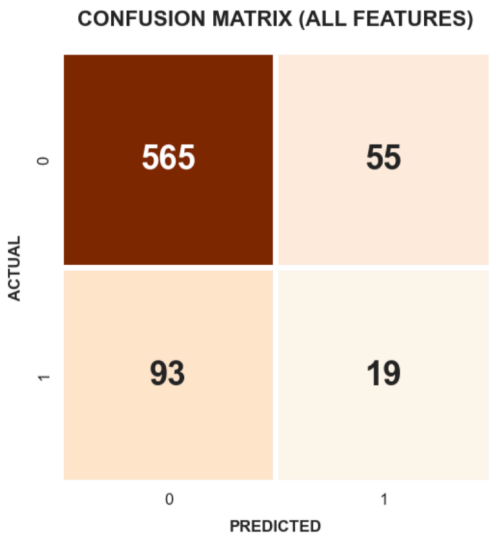
Parameter	**Nilai Optimal**
Learning Rate	0.3
Max Depth	7
N Estimators	100
Subsample	0.8
Colsample	0.8
Gamma	0.2

Berdasarkan Tabel 4.18, diperoleh konfigurasi parameter yang menjadi landasan pelatihan model akhir. Nilai *learning rate* sebesar 0.3 digunakan untuk mengatur kecepatan konvergensi model dalam mempelajari pola data. Kedalaman pohon (*max depth*) sebesar 7 dipilih untuk menangkap relasi data yang kompleks, sementara 100 *n_estimators* menentukan jumlah pohon keputusan yang dibangun dalam proses *boosting*.

Untuk menjaga generalisasi model dan mencegah *overfitting*, digunakan nilai *subsample* sebesar 0.8 dan *colsample* sebesar 0.8, yang berarti model melakukan proses *sampling* terhadap data maupun fitur pada setiap iterasinya. Terakhir, nilai *gamma* sebesar 0.2 diterapkan untuk memberikan batasan

minimum *loss reduction* pada setiap *node split*, yang berfungsi sebagai regularisasi tambahan agar model lebih stabil.

Guna mengevaluasi efektivitas model pada skenario ini, dilakukan analisis melalui *Confusion Matrix*. Evaluasi ini bertujuan untuk membandingkan label hasil prediksi model dengan nilai aktual pada data uji, sehingga dapat teridentifikasi secara presisi jumlah data *True Positive, True Negative, False Positive,* maupun *False Negative*. Hasil visualisasi *Confusion Matrix* tersebut ditampilkan pada Gambar 4.4.



Gambar 4.4 *Confusion Matrix* Skenario 3

Berdasarkan hasil *Confusion Matrix* pada tahap *All Features* pada Gambar 4.4, model berhasil mengklasifikasikan 565 data sebagai *True Negative* (TN), yaitu individu sehat yang diprediksi dengan benar. Sementara itu, untuk kelas minoritas, model mampu mendeteksi 19 data sebagai *True Positive* (TP), yaitu kasus penyakit jantung (CHD) yang teridentifikasi secara tepat. Namun, model juga mencatat 93 data sebagai *False Negative* (FN) individu yang sebenarnya berisiko CHD tetapi

diprediksi sehat serta 55 data sebagai *False Positive* (FP), yakni individu sehat yang diprediksi berisiko CHD. Hasil ini mengindikasikan bahwa model memiliki performa yang sangat dominan dalam mengenali kelas mayoritas (sehat), namun masih memiliki keterbatasan signifikan dalam mendeteksi kelas minoritas (CHD). Hal ini menjadi alasan mendasar perlunya penerapan teknik penyeimbangan data dan seleksi fitur pada skenario selanjutnya.

Setelah tahap pelatihan selesai, langkah berikutnya adalah mengevaluasi performa model menggunakan metrik klasifikasi untuk mengetahui efektivitasnya pada data uji. Metrik yang digunakan meliputi *accuracy, precision, recall, F1-score,* dan *ROC-AUC*. Evaluasi ini bertujuan memberikan gambaran komprehensif mengenai kemampuan model dalam membedakan responden yang tidak berisiko (kelas 0) dan berisiko (kelas 1). Ringkasan performa model tersebut disajikan pada Tabel 4.19.

Tabel 4.19 Hasil Metrik Evaluasi

Metric	***Value***
Accuracy	79.78%
Precision	25.68%
Recall	16.96%
F1-Score	20.43%
ROC-AUC	63.06%

Berdasarkan Tabel 4.19, model mencapai *Accuracy* sebesar 79.78%, yang menunjukkan tingkat keberhasilan klasifikasi yang tinggi secara keseluruhan. Namun, nilai *Precision* sebesar 25.68% mengindikasikan bahwa dari seluruh prediksi positif, hanya sekitar seperempat yang terbukti benar. Lebih lanjut, nilai *Recall* sebesar 16.96% menyoroti bahwa model baru mampu mengenali sebagian kecil dari total kasus CHD yang sebenarnya ada dalam data uji.

Nilai *F1-Score* sebesar 20.43% semakin menegaskan bahwa keseimbangan antara *precision* dan *recall* dalam mengenali kelas positif belum mencapai tingkat yang optimal. Begitu pula dengan nilai ROC-AUC sebesar 63.06% yang mencerminkan kemampuan diskriminasi model terhadap kedua kelas yang masih terbatas. Kesimpulan dari tahap ini adalah bahwa penggunaan seluruh fitur tanpa proses seleksi dan penyeimbangan data belum mampu menghasilkan model yang sensitif terhadap deteksi kelas minoritas, sehingga optimasi pada skenario berikutnya sangat diperlukan.

4.2.4 Hasil Uji Coba Skenario 4 (Non SMOTE dengan RFE)

Pada skenario keempat, model dikembangkan tanpa menerapkan teknik penyeimbangan data. Dengan kata lain, pelatihan model dilakukan menggunakan dataset dengan distribusi kelas asli yang bersifat *imbalanced*, di mana proporsi kelas mayoritas jauh lebih dominan dibandingkan kelas minoritas. Pendekatan ini bertujuan untuk mengevaluasi bagaimana performa model XGBoost merespons dataset yang memiliki ketimpangan distribusi kelas.

Meskipun tanpa penyeimbangan, proses seleksi fitur tetap diterapkan melalui metode *Recursive Feature Elimination* (RFE). RFE berfungsi untuk memfilter variabel yang paling relevan dengan mengeliminasi fitur-fitur yang memiliki kontribusi minimal secara bertahap. Mengingat RFE diterapkan pada dataset yang sama sebelum proses penyeimbangan dilakukan, hasil seleksi fitur pada skenario ini bersifat identik dengan hasil yang telah dipaparkan pada Sub-bab 4.2.1 (Tabel 4.6), yakni mempertahankan 10 fitur utama dan mengeliminasi 5 fitur lainnya.

Setelah tahap seleksi fitur selesai, langkah berikutnya adalah mengoptimasi parameter model melalui *hyperparameter tuning* menggunakan *GridSearchCV* dengan skema 5-*fold cross-validation*. Optimasi ini mencakup pengujian terhadap 729 kombinasi parameter, yang melibatkan total 3.645 proses *fitting*. Hasil konfigurasi parameter terbaik yang diperoleh disajikan pada Tabel 4.20.

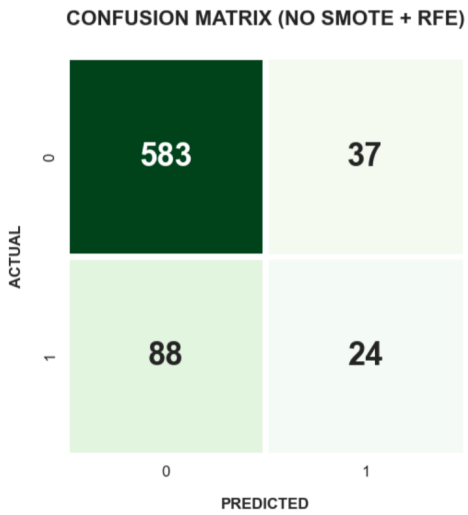
Tabel 4.20 *Hasil Seleksi Hyperparameter Tunning*

Hyperparameter	**Nilai Optimal**
Learning Rate	0.3
Max Depth	7
N Estimators	200
Subsample	1.0
Colsample Bytree	0.9
Gamma	0

Berdasarkan Tabel 4.20, diperoleh parameter optimal dengan *learning rate* 0.3, yang mengindikasikan tingkat pembelajaran model yang cukup agresif dalam pembaruan bobot. Kedalaman pohon (*max depth*) sebesar 7 dipilih untuk menangkap pola kompleks pada data, didukung oleh 200 *n_estimators* yang membangun struktur model *boosting* secara bertahap. Nilai *subsample* sebesar 1.0 menunjukkan penggunaan seluruh data pada setiap iterasi, sementara *colsample bytree* sebesar 0.9 mengisyaratkan penggunaan sebagian besar fitur dalam setiap pohon. Adapun nilai *gamma* 0 menunjukkan bahwa tidak ada Batasan minimum *loss reduction* yang diterapkan untuk melakukan *node split*.

Untuk menganalisis lebih dalam mengenai kemampuan klasifikasi model terhadap data uji, digunakan *Confusion Matrix*. Evaluasi ini memberikan gambaran akurat mengenai jumlah prediksi yang benar (*True Positive* dan *True Negative*) serta kesalahan klasifikasi (*False Positive* dan *False Negative*) yang dihasilkan oleh

model pada distribusi data yang tidak seimbang. Visualisasi *Confusion Matrix* untuk skenario ini disajikan pada gambar 4.5.



Gambar 4.5 *Confusion Matrix* Skenario 4

Berdasarkan hasil *Confusion Matrix* pada Gambar 4.5, model berhasil mengklasifikasikan 583 data sebagai *True Negative* (TN), yaitu responden sehat yang diprediksi dengan benar. Sementara itu, untuk kelas minoritas, model mendeteksi 24 data sebagai *True Positive* (TP), yakni kasus sakit yang berhasil dikenali dengan tepat. Di sisi lain, model masih menghasilkan kesalahan prediksi berupa 37 data *False Positive* (FP) responden sehat yang diprediksi sakit dan 88 data *False Negative* (FN) responden sakit yang diprediksi sehat. Hasil ini mempertegas bahwa model lebih dominan dalam mengenali kelas mayoritas (sehat), yang dipengaruhi oleh ketidakseimbangan distribusi data pada dataset asli di mana kelas sehat jauh lebih dominan dibandingkan kelas sakit.

Setelah proses klasifikasi selesai, tahap selanjutnya adalah mengevaluasi performa model secara komprehensif menggunakan metrik *accuracy, precision, recall, F1-score,* dan ROC-AUC. Metrik-metrik ini bertujuan untuk mengukur

efektivitas model secara menyeluruh, baik dalam mendeteksi kelas mayoritas maupun minoritas. Hasil evaluasi performa model untuk skenario *No* SMOTE + RFE disajikan pada Tabel 4.21.

Tabel 4.21 Hasil Prediksi Model

Metrik Evaluasi	Nilai
Accuracy	82.92%
Precision	39.34%
Recall	21.43%
F1-Score	27.75%
ROC-AUC	64.75%

Berdasarkan Tabel 4.21, model mencapai *Accuracy* sebesar 82.92%, yang menunjukkan tingkat keberhasilan klasifikasi yang tinggi secara keseluruhan. Namun, perlu dicatat bahwa nilai akurasi yang tinggi tersebut tidak sepenuhnya mencerminkan efektivitas model dalam mendeteksi kelas positif, mengingat dataset yang digunakan bersifat *imbalanced*. Nilai *Precision* sebesar 39.34% menunjukkan bahwa dari seluruh prediksi positif, hanya sekitar 39.34% yang terkonfirmasi benar sebagai kasus sakit.

Lebih lanjut, nilai *Recall* sebesar 21.43% mengindikasikan bahwa model baru mampu mengenali sekitar 21.43% dari total kasus sakit yang sebenarnya ada di data uji. Nilai *F1-Score* sebesar 27.75% memperkuat bukti bahwa keseimbangan antara *precision* dan *recall* masih belum optimal dalam mendeteksi kelas minoritas. Sementara itu, nilai ROC-AUC sebesar 64.75% mencerminkan kemampuan diskriminasi model yang berada pada tingkat moderat. Secara keseluruhan, performa model pada skenario ini menunjukkan bahwa tanpa teknik penyeimbangan data, kemampuan model untuk mendeteksi kelas minoritas (sakit) masih sangat terbatas meskipun akurasi secara umum terlihat tinggi.

4.2.5 Hasil Uji Coba Skenario 5 (Non SMOTE dengan RFE *by Threshold*)

Pada Skenario 5, pengujian dilakukan tanpa penerapan Teknik penyeimbangan data, sehingga menggunakan data asli tanpa proses *oversampling* maupun *undersampling*. Pengujian ini bertujuan untuk mengevaluasi performa model XGBoost pada data yang tidak diseimbangkan, sekaligus melihat pengaruh seleksi fitur berbasis *feature importance* dengan pendekatan *threshold*. Nilai *threshold* sebesar 0,07 digunakan sebagai acuan yang sama seperti pada skenario sebelumnya untuk mengurangi jumlah fitur pada pengujian lanjutan. Sama seperti pada Skenario 2, pengujian ini juga dibagi menjadi dua sub-skenario, yaitu Skenario 5a yang menggunakan *threshold* sebesar 0,07 dan Skenario 5b yang menggunakan *threshold* berdasarkan nilai rata-rata *feature importance*.

a. Hasil Uji Coba Skenario 5a (Non SMOTE dengan RFE *by Threshold* 0.07)

Pada skenario ini, karakteristik ruang fitur yang terbentuk identik dengan hasil seleksi fitur pada Skenario 2a sebagaimana disajikan pada Tabel 4.10 di Sub-bab 4.2.2, baik dari urutan peringkat, nilai *gain score*, hingga status akhir fiturnya. Kesamaan ini membuktikan secara metodologis bahwa proses RFE dan pemotongan ambang batas (*threshold*) memang dilakukan lebih dulu pada data latih asli (*original train set*) sebelum ada manipulasi data lainnya. Oleh karena itu, kondisi apakah data tersebut nantinya dibiarkan tidak seimbang (*non-balanced*) seperti pada Skenario 5a ini, atau akan diseimbangkan dengan SMOTE seperti pada

Skenario 2a, tidak mengubah hasil penyaringan fitur awal yang dilakukan oleh model XGBoost.

Berdasarkan data konkrit tersebut, proses pemotongan berbasis ambang batas (*threshold* 0,0700) berhasil menyaring dan mempertahankan 5 fitur utama dengan status retained, di mana variabel *prevalentHyp* menempati peringkat pertama (*Final Rank* 1) dengan nilai gain score paling dominan sebesar 0,1434. Kontribusi signifikan berikutnya diikuti secara berurutan oleh fitur *age* (0,1072), *glucose* (0,0744), *male* (0,0725), serta fitur *currentSmoker* (0,0708) sebagai variabel terakhir yang lolos batas relevansi minimal. Sebaliknya, 10 fitur lainnya secara ketat dinyatakan gugur dengan status *eliminated*, dimulai dari fitur *cigsPerDay* pada peringkat keenam yang tereliminasi tipis dengan skor 0,0688, hingga variabel dengan bobot kontribusi terendah yaitu diabetes (0,0246) pada peringkat 14 dan *prevalentStroke* (0,0145) pada peringkat 15. Pemangkasan 10 fitur yang kurang informatif ini tetap konsisten dilakukan untuk mereduksi dimensi data, sehingga model dapat lebih fokus mempelajari karakteristik kunci dari kombinasi fitur yang paling relevan.

Setelah fitur relevan terpilih, tahap selanjutnya adalah mengoptimalkan parameter model melalui *hyperparameter tuning* menggunakan *GridSearchCV* dengan skema 5-*fold cross-validation*. Optimasi ini melibatkan pengujian 729 kombinasi parameter dengan total 3.645 proses fitting model. Hasil konfigurasi parameter terbaik yang diperoleh disajikan pada Tabel 4.22.

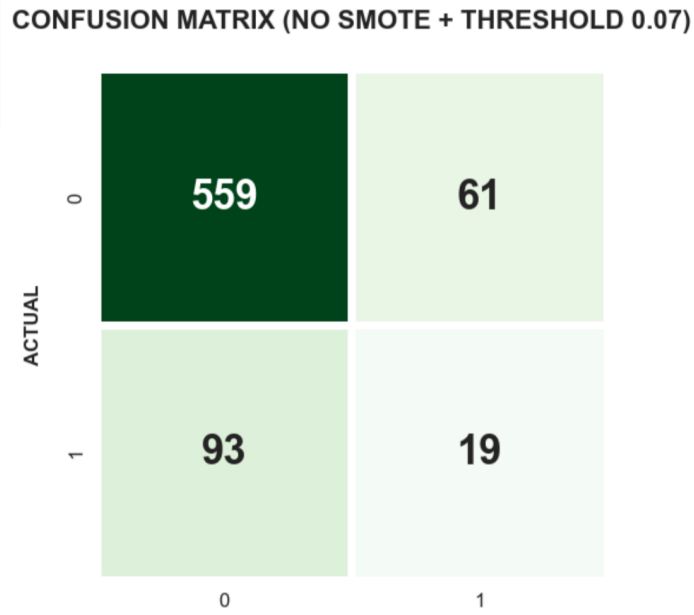
Tabel 4.22 *Hyperparameter Tunning*

Hyperparameter	**Nilai Optimal**
Learning Rate	0.3
Max Depth	7
N Estimators	200
Subsample	0.8
Colsample Bytree	0.8
Gamma	0.2

Berdasarkan Tabel 4.22, kombinasi parameter optimal yang didapatkan meliputi *learning rate* 0.3, *max depth* 7, serta 200 *n_estimators*. Nilai *learning rate* 0.3 menunjukkan model melakukan pembelajaran dengan tingkat konvergensi yang cukup agresif. Kedalaman pohon (*max depth*) sebesar 7 dipilih untuk menangkap kompleksitas pola dalam data, sementara 200 *n_estimators* menentukan jumlah pohon keputusan yang dibangun dalam proses *boosting*. Untuk menjaga stabilitas model, digunakan nilai *subsample* 0.8 dan *colsample bytree* 0.8, yang berarti model melakukan proses *random sampling* pada data dan fitur di setiap iterasinya guna mengurangi risiko *overfitting*. Selain itu, nilai *gamma* sebesar 0.2 diterapkan untuk menetapkan batas minimum *loss reduction* yang diperlukan sebelum dilakukan *node split*. Dengan konfigurasi ini, model diharapkan mampu memaksimalkan kapasitas pembelajarannya pada dataset asli yang tidak seimbang.

Evaluasi performa model dilakukan dengan membandingkan hasil prediksi pada data uji terhadap label aktual menggunakan *Confusion Matrix*. Evaluasi ini krusial untuk memetakan jumlah prediksi yang benar dan kesalahan klasifikasi (seperti *False Positive* dan *False Negative*), yang akan memberikan gambaran mendalam mengenai seberapa efektif model

dalam melakukan klasifikasi pada kondisi data yang tetap *imbalanced*. Hasil visualisasi dari *Confusion Matrix* pada skenario ini ditampilkan pada gambar 4.6.



Gambar 4.6 *Confusion Matrix* Skenario 5a

Berdasarkan hasil *Confusion Matrix* pada Gambar 4.6, model berhasil mengklasifikasikan 559 data sebagai *True Negative* (TN), yaitu responden sehat yang diprediksi dengan benar. Sementara itu, untuk kelas minoritas, model mendeteksi 19 data sebagai *True Positive* (TP), yakni kasus sakit yang berhasil dikenali dengan tepat. Di sisi lain, model masih mencatat kesalahan prediksi berupa 61 data *False Positive* (FP) responden sehat yang diprediksi sakit dan 93 data *False Negative* (FN) responden sakit yang diprediksi sehat. Hasil ini mempertegas bahwa model tetap menunjukkan bias terhadap kelas mayoritas, di mana ketidakseimbangan

jumlah data pada dataset asli membuat model lebih dominan dalam mengenali kelas sehat dibandingkan kelas sakit.

Setelah proses klasifikasi selesai, tahap selanjutnya adalah mengevaluasi performa model menggunakan metrik *accuracy, precision, recall, F1-score,* dan ROC-AUC. Metrik-metrik ini bertujuan untuk mengukur efektivitas model secara menyeluruh dalam mendeteksi kelas pada dataset yang tidak seimbang. Hasil evaluasi performa model untuk skenario *No* SMOTE + *Threshold 0.07* disajikan pada Tabel 4.23.

Tabel 4.23 Hasil Prediksi Model

Metrik Evaluasi	Nilai
Accuracy	78.96%
Precision	23.75%
Recall	16.96%
F1-Score	19.79%
ROC-AUC	64.35%

Berdasarkan Tabel 4.23, model memperoleh nilai *Accuracy* sebesar 78.96%. Meskipun angka ini menunjukkan tingkat klasifikasi benar secara keseluruhan yang cukup tinggi, akurasi tersebut tidak mencerminkan performa model dalam mendeteksi kelas minoritas secara optimal. Nilai *Precision* sebesar 23.75% mengindikasikan bahwa dari seluruh prediksi positif yang dihasilkan, hanya 23.75% yang terkonfirmasi benar sebagai kasus sakit. Sementara itu, nilai *Recall* sebesar 16.96% menunjukkan bahwa model baru mampu mengenali 16.96% dari total kasus sakit yang sebenarnya ada di data uji. Nilai *F1-Score* sebesar 19.79% menegaskan bahwa keseimbangan antara *precision* dan *recall* dalam mengenali kelas minoritas masih rendah, sehingga diperlukan optimasi

lebih lanjut. Adapun nilai ROC-AUC sebesar 64.35% mencerminkan kemampuan diskriminasi model yang berada pada tingkat moderat dalam membedakan kedua kelas.

b. Hasil Uji Coba Skenario 5b (Non SMOTE dengan RFE *by Threshold Mean*)

Pada skenario ini, karakteristik ruang fitur yang terbentuk juga mencatat hasil yang identik dengan seleksi fitur pada Skenario 2b yang ditampilkan pada Tabel 4.14 di Sub-bab 4.2.2, baik dari urutan peringkat, nilai *gain score*, hingga status akhir fiturnya. Kesamaan ini menegaskan kembali bahwa proses RFE dan penentuan ambang batas berbasis rata-rata (*threshold mean*) dilakukan langsung pada data latih asli (*original train set*) sebelum mendapat intervensi manipulasi data lainnya. Oleh karena itu, kondisi apakah data tersebut nantinya dibiarkan tidak seimbang (*non-balanced*) pada Skenario 5b ini, atau akan diseimbangkan dengan teknik SMOTE seperti pada Skenario 2b, tidak memberikan pengaruh terhadap profil penyaringan fitur awal.

Merujuk pada data tersebut, perhitungan nilai rata-rata dari seluruh *gain score* menghasilkan angka ambang batas (*threshold*) sebesar 0,0667. Berdasarkan batasan adaptif ini, model berhasil menyaring dan mempertahankan 6 fitur utama yang memiliki kontribusi di atas atau sama dengan nilai rata-rata (*retained*). Variabel *prevalentHyp* menjadi contributor paling dominan dengan skor 0,1434, diikuti oleh *age* (0,1072),

glucose (0,0744), *male* (0,0725), *currentSmoker* (0,0708), serta *cigsPerDay* (0,0688) sebagai fitur terakhir yang lolos batas minimal.

Sementara itu, 9 fitur sisanya secara otomatis gugur karena memiliki nilai kepentingan di bawah rata-rata (*eliminated*). Proses eliminasi ini dimulai dari fitur *sysBP* yang berada tipis di bawah batasan dengan skor 0,0666, hingga fitur dengan bobot terendah yaitu *prevalentStroke* dengan skor 0,0145. Pemangkasan 9 fitur yang kurang informatif ini bertujuan untuk mereduksi dimensi data, sehingga model XGBoost bisa lebih fokus dalam mempelajari pola dari kombinasi fitur yang paling optimal.

Setelah fitur relevan terpilih, tahap selanjutnya adalah mengoptimalkan parameter model melalui *hyperparameter tuning* menggunakan *GridSearchCV* dengan skema 5-*fold cross-validation*. Optimasi ini melibatkan pengujian 729 kombinasi parameter dengan total 3.645 proses fitting model. Hasil konfigurasi parameter terbaik yang diperoleh disajikan pada Tabel 4.24.

Tabel 4.24 Hasil Seleksi *Hyperparameter Tunning*

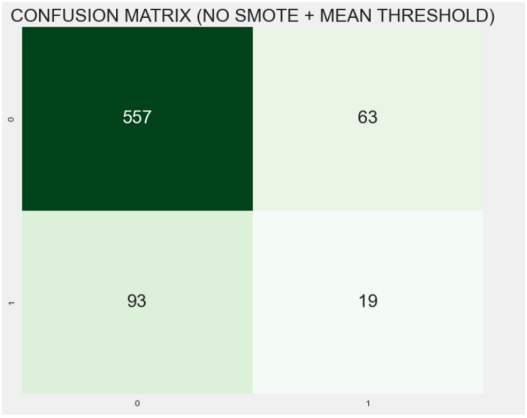
Hyperparameter	**Nilai Optimal**
Learning Rate	0.3
Max Depth	7
N Estimators	200
Subsample	0.8
Colsample Bytree	0.9
Gamma	0

Berdasarkan Tabel 4.24, diperoleh kombinasi parameter optimal untuk model XGBoost yang terdiri dari *learning rate* 0,3, *max depth* 7, dan *n_estimators* 200. Nilai *learning rate* sebesar 0,3 menunjukkan bahwa model melakukan pembelajaran dengan tingkat konvergensi yang cukup

agresif, sementara kedalaman pohon (*max depth*) sebesar 7 dipilih agar model mampu menangkap kompleksitas pola data secara lebih mendalam. Penggunaan 200 *n_estimators* menentukan jumlah pohon keputusan yang dibangun dalam proses *boosting* untuk mengoptimalkan hasil prediksi.

Untuk menjaga stabilitas serta meminimalisir risiko *overfitting*, parameter *subsample* ditetapkan sebesar 0,8 dan *colsample bytree* sebesar 0,9. Konfigurasi ini berarti bahwa pada setiap iterasi pembentukan pohon, model melakukan *random sampling* terhadap 80% data dan 90% fitur yang tersedia. Selain itu, nilai *gamma* sebesar 0 diterapkan untuk menunjukkan bahwa model tetap melakukan pemisahan cabang (*node split*) selama pemisahan tersebut memberikan kontribusi positif terhadap penurunan *loss*. Dengan konfigurasi parameter tersebut, model diharapkan mampu memaksimalkan kapasitas pembelajarannya dalam mengenali pola risiko penyakit jantung secara lebih akurat dan stabil.

Evaluasi performa model dilakukan dengan membandingkan hasil prediksi pada data uji terhadap label aktual menggunakan *Confusion Matrix*. Evaluasi ini krusial untuk memetakan jumlah prediksi yang benar dan kesalahan klasifikasi (seperti *False Positive* dan *False Negative*), yang akan memberikan gambaran mendalam mengenai seberapa efektif model dalam melakukan klasifikasi pada kondisi data yang tetap *imbalanced*. Hasil visualisasi dari *Confusion Matrix* pada skenario ini ditampilkan pada gambar 4.7.



Gambar 4.7 *Confusion Matrix* Skenario 5b

Berdasarkan hasil *Confusion Matrix* pada Gambar 4.7, model berhasil mengklasifikasikan 557 data sebagai *True Negative* (TN), yaitu responden sehat yang diprediksi dengan benar. Sementara itu, untuk kelas minoritas, model mendeteksi 19 data sebagai *True Positive* (TP), yakni kasus sakit yang berhasil dikenali dengan tepat. Di sisi lain, model mencatat kesalahan prediksi berupa 63 data *False Positive* (FP) responden sehat yang justru diprediksi sakit dan 93 data *False Negative* (FN) responden sakit yang diprediksi sehat. Hasil ini mempertegas bahwa model masih menunjukkan bias terhadap kelas mayoritas. Ketidakseimbangan jumlah data pada dataset asli membuat model jauh lebih dominan dalam mengenali kelas sehat dibandingkan kelas sakit.

Setelah proses klasifikasi selesai, tahap selanjutnya adalah mengevaluasi performa model menggunakan metrik *accuracy, precision, recall, F1-score,* dan ROC-AUC. Metrik-metrik ini bertujuan untuk mengukur efektivitas model secara menyeluruh dalam mendeteksi kelas

pada dataset yang tidak seimbang. Hasil evaluasi performa model untuk skenario *Non* SMOTE + *Threshold 0.07* disajikan pada Tabel 4.25.

Tabel 4.25 Hasil Prediksi Model

Metrik Evaluasi	Nilai
Accuracy	78.69%
Precision	23.17%
Recall	16.96%
F1-Score	19.59%
ROC-AUC	59.99%

Berdasarkan Berdasarkan Tabel 4.25, model memperoleh tingkat *Accuracy* sebesar 78,69%. Meskipun angka tersebut menunjukkan tingkat klasifikasi benar secara keseluruhan yang cukup tinggi, akurasi ini tidak mencerminkan performa optimal dalam mendeteksi kelas minoritas. Nilai *Precision* sebesar 23,17% mengindikasikan bahwa dari seluruh hasil prediksi sakit yang diberikan oleh model, hanya 23,17% yang terkonfirmasi benar sebagai kasus sakit. Sementara itu, nilai *Recall* sebesar 16,96% menunjukkan bahwa model baru mampu mengenali 16,96% dari total kasus sakit yang sebenarnya ada di data uji. Rendahnya nilai *F1-Score* sebesar 19,59% menegaskan bahwa keseimbangan antara *Precision* dan *Recall* dalam mengenali kelas minoritas masih belum tercapai, sehingga model cenderung memiliki bias terhadap kelas mayoritas. Adapun nilai ROC-AUC sebesar 59,99% mencerminkan kemampuan diskriminasi model yang berada pada tingkat moderat dalam membedakan antara responden sehat dan responden sakit.

4.2.6 Hasil Uji Coba Skenario 6 (Non SMOTE Non RFE)

Pada skenario keenam, dilakukan proses *hyperparameter tuning* pada model XGBoost menggunakan data *baseline*. Dalam tahap ini, tidak diterapkan teknik seleksi fitur (RFE) maupun penyeimbangan data (SMOTE), sehingga model mempelajari seluruh karakteristik dataset dalam distribusi aslinya. Pendekatan ini berfungsi sebagai tolok ukur (*benchmark*) untuk mengukur performa dasar model sebelum diberikan intervensi teknis.

Proses optimasi dilakukan melalui metode *GridSearchCV* dengan skema 5-*fold cross-validation*. Metode ini menguji 729 kombinasi *hyperparameter* secara sistematis, yang melibatkan total 3.645 proses *fitting* model. Konfigurasi parameter terbaik yang diperoleh dari proses tersebut disajikan pada Tabel 4.26.

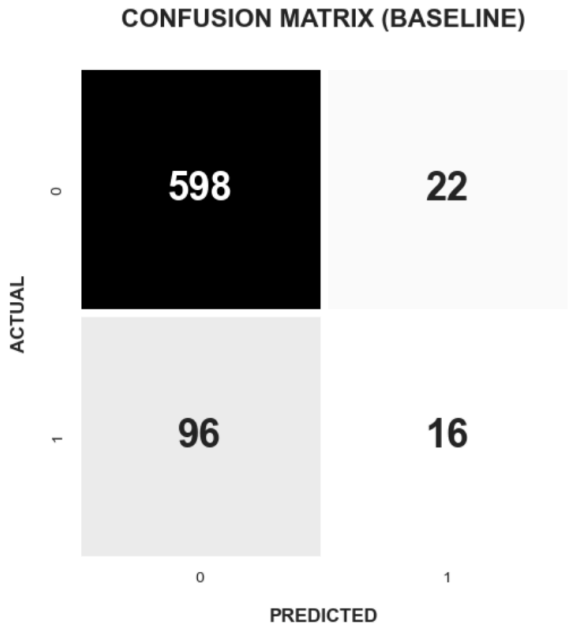
Tabel 4.26 *Hasil Seleksi Hyperparameter Tunning*

Hyperparameter	**Nilai Optimal**
Learning Rate	0.3
Max Depth	5
N Estimators	100
Subsample	1.0
Colsample Bytree	1.0
Gamma	0.2

Berdasarkan Tabel 4.26, parameter optimal yang didapatkan mencakup *learning rate* 0.3, *max depth* 5, serta 100 *n_estimators*. Nilai *learning rate* 0.3 mengindikasikan laju pembelajaran yang cukup agresif, sementara *max depth* sebesar 5 dipilih agar model tetap mampu menangkap pola data yang relevan tanpa terjebak pada kompleksitas yang berlebih (*over-complexity*). Untuk membangun struktur *boosting*, digunakan 100 pohon keputusan. Guna menjaga stabilitas model dan memitigasi risiko *overfitting*, nilai *subsample* ditetapkan sebesar 1.00, sementara *colsample bytree* sebesar 1.0 mengindikasikan bahwa

seluruh fitur disertakan dalam setiap pembentukan pohon. Terakhir, nilai *gamma* 0.2 diterapkan sebagai parameter regularisasi untuk mengontrol kedalaman *node split*. Dengan konfigurasi ini, model diharapkan mampu menghasilkan representasi pola yang paling stabil dari dataset dasar.

Untuk mengevaluasi efektivitas model, digunakan *Confusion Matrix* yang membandingkan hasil prediksi model terhadap label aktual pada data uji. Analisis ini sangat krusial untuk mengidentifikasi jumlah prediksi yang benar (*True Positive*dan *True Negative*) serta kesalahan klasifikasi (*False Positive* dan *False Negative*) yang dihasilkan oleh model pada kondisi data *baseline*. Visualisasi *Confusion Matrix* untuk skenario *Non-SMOTE* dan *Non-RFE* ini ditampilkan pada gambar 4.8.



Gambar 4.8 *Confusion Matrix* Skenario 6

Berdasarkan *Confusion Matrix* pada Gambar 4.8, model *baseline* berhasil mengklasifikasikan 598 data sebagai *True Negative* (TN), yaitu individu sehat yang

diprediksi dengan benar. Sementara itu, untuk kelas minoritas, model mendeteksi 16 data sebagai *True Positive* (TP), yakni kasus penyakit jantung (CHD) yang berhasil teridentifikasi dengan tepat. Di sisi lain, model mencatat 22 data sebagai *False Positive* (FP) individu sehat yang diprediksi berisiko CHD dan 96 data sebagai *False Negative* (FN) individu yang sebenarnya berisiko CHD namun diprediksi sebagai sehat. Angka *False Negative* yang tinggi ini menunjukkan bahwa model cenderung bias terhadap kelas mayoritas, sehingga kemampuan dalam mengidentifikasi kasus penyakit jantung masih belum optimal pada model *baseline* ini.

Sebagai langkah evaluasi akhir, dilakukan pengukuran performa model menggunakan metrik *Accuracy, Precision, Recall, F1-Score,* dan ROC-AUC. Penggunaan metrik-metrik ini sangat krusial untuk memberikan gambaran komprehensif mengenai kemampuan model dalam melakukan klasifikasi, terutama dalam mendeteksi kasus penyakit jantung sebagai kelas minoritas. Hasil evaluasi performa pada skenario *baseline* disajikan pada Tabel 4.27.

Tabel 4.27 Hasil Prediksi Model

Metric	**Nilai**
Accuracy	83.88%
Precision	42.11%
Recall	14.29%
F1-Score	21.33%
ROC-AUC	66.79%

Berdasarkan Tabel 4.27, model memperoleh *Accuracy* sebesar 83.88%, yang menunjukkan tingkat klasifikasi benar secara keseluruhan yang tinggi. Namun, perlu ditekankan kembali bahwa akurasi yang tinggi ini tidak mencerminkan efektivitas model secara menyeluruh karena pengaruh

ketidakseimbangan dataset (*imbalanced data*). Nilai *Precision* sebesar 42.11% menunjukkan bahwa dari seluruh prediksi positif, hanya sekitar 42.11% yang merupakan prediksi yang benar. Nilai *Recall* sebesar 14.29% menegaskan bahwa kemampuan model dalam mendeteksi kasus penyakit jantung masih sangat terbatas, dengan sebagian besar kasus tidak teridentifikasi. Nilai *F1-Score* sebesar 21.33% menggambarkan keseimbangan yang rendah antara *precision* dan *recall*, sementara nilai ROC-AUC sebesar 66.79% mengindikasikan kemampuan diskriminasi model yang masih berada pada tingkat moderat. Hasil ini memberikan landasan (*benchmark*) bahwa tanpa teknik penyeimbangan data dan seleksi fitur, model memiliki kendala signifikan dalam mengenali kelas minoritas.

4.2.7 Hasil Uji Coba Skenario 7 (*Random Undersampling* dengan RFE)

Pada skenario ketujuh, proses pemodelan diawali dengan tahap seleksi fitur untuk mengidentifikasi variabel yang paling relevan terhadap variabel target. Langkah ini krusial untuk meminimalisir *noise* dari fitur yang kurang berkontribusi, sehingga model dapat mempelajari pola data secara lebih fokus dan efisien.

Metode yang digunakan adalah *Recursive Feature Elimination* (RFE). RFE bekerja dengan mengeliminasi fitur secara bertahap berdasarkan kontribusi terkecil terhadap performa model hingga diperoleh subset fitur optimal. Dalam skenario ini, target seleksi ditetapkan untuk mempertahankan 10 fitur terbaik dari 15 fitur awal yang tersedia. Hasil seleksi fitur menggunakan metode RFE disajikan pada Tabel 4.28.

Tabel 4.28 Hasil RFE 10 Fitur

Fitur	*Ranking*	*Importance*	Status
prevalentHyp	1	0.1434	*RETAINED*
age	2	0.1072	*RETAINED*
glucose	3	0.0744	*RETAINED*
male	4	0.0725	*RETAINED*
currentSmoker	5	0.0708	*RETAINED*
cigsPerDay	6	0.0688	*RETAINED*
sysBP	7	0.0666	*RETAINED*
education	8	0.0639	*RETAINED*
totChol	9	0.0625	*RETAINED*
diaBP	10	0.0622	*RETAINED*
BPMeds	11	0.0584	*ELIMINATED*
BMI	12	0.0558	*ELIMINATED*
heartRate	13	0.0545	*ELIMINATED*
diabetes	14	0.0246	*ELIMINATED*
prevalentStroke	15	0.0145	*ELIMINATED*

Berdasarkan Tabel 4.28, sebanyak 10 fitur telah terpilih dengan status *retained* untuk digunakan dalam tahap pemodelan selanjutnya. Fitur *prevalentHyp* mencatatkan *importance score* tertinggi sebesar 0.1434, yang menandakan bahwa variabel tersebut memiliki kontribusi paling dominan terhadap model XGBoost dibandingkan fitur lainnya. Diikuti oleh fitur *age* (0.1072) dan *glucose* (0.0744) di posisi tiga besar. Sebaliknya, lima fitur sisanya, yaitu *BPMeds*, BMI, *heartRate*, diabetes, dan *prevalentStroke*, dieliminasi (*eliminated*) dari *feature space* karena memiliki kontribusi informatif (*gain score*) yang relatif lebih rendah atau berada di luar batas 10 fitur terbaik yang ditargetkan oleh metode RFE.

Setelah proses seleksi fitur, dilakukan penyeimbangan data menggunakan teknik *Random Undersampling* pada data latih (*training set*). Teknik ini dipilih untuk menangani ketimpangan distribusi kelas, di mana rasio awal antara kelas sehat dan kelas sakit mencapai 1:5,6. Dengan *undersampling*, jumlah sampel pada kelas mayoritas dikurangi hingga setara dengan jumlah sampel kelas minoritas,

sehingga distribusi data latih menjadi seimbang (rasio 1:1). Distribusi data sebelum dan sesudah proses *undersampling* diuraikan pada Tabel 4.29.

Tabel 4.29 Hasil *Random Undersampling*

Kategori	Latih Asli	Latih *Undersampled*	Uji (*Original*)
Sehat (Kelas 0)	2479	445	620
Sakit (Kelas 1)	445	445	112
Total	2924	890	732

Data pada Tabel 4.29 menunjukkan bahwa setelah *undersampling*, total data latih menjadi 890 sampel yang terdistribusi seimbang. Perlu dicatat bahwa data pengujian (*test set*) tetap mempertahankan distribusi aslinya (732 sampel) agar hasil evaluasi model tetap mencerminkan performa pada kondisi data riil.

Selanjutnya, dilakukan optimasi model melalui *hyperparameter tuning* menggunakan *GridSearchCV* dengan skema 5-*fold cross-validation*. Proses ini melibatkan pengujian terhadap 729 kombinasi parameter untuk menemukan konfigurasi yang paling optimal dalam memitigasi *overfitting* serta meningkatkan daya generalisasi model. Hasil kombinasi parameter terbaik disajikan pada Tabel 4.30.

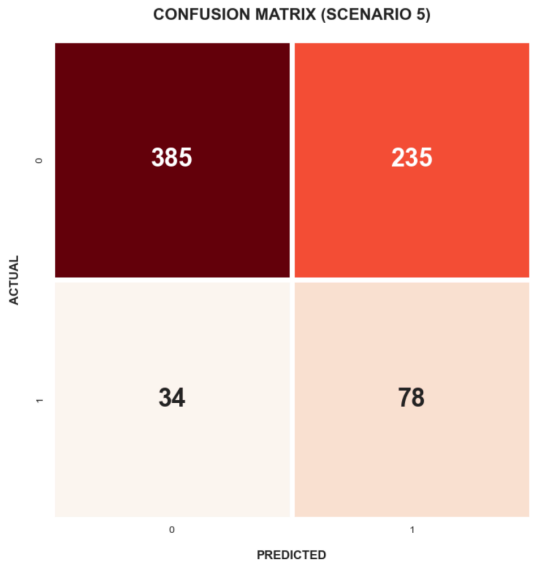
Tabel 4.30 Hasil Seleksi *Hyperparameter Tunning*

Hyperparameter	**Nilai Optimal**
Learning Rate	0.01
Max Depth	3
N Estimators	100
Subsample	0.8
Colsample Bytree	0.9
Gamma	0

Berdasarkan Tabel 4.30, diperoleh parameter optimal dengan *learning rate* 0.01, yang memberikan laju pembelajaran lebih stabil. Penggunaan *max depth* 3 menunjukkan bahwa model menggunakan pohon keputusan yang relatif dangkal untuk menjaga generalisasi, sementara 100 *n_estimators* membangun

struktur *boosting* yang komprehensif. Parameter *subsample* 0.8 dan *colsample bytree* 0.9 berfungsi untuk memastikan keberagaman data dan fitur pada setiap iterasi, serta nilai *gamma* 0 menandakan tidak adanya restriksi *loss reduction* yang ekstrem.

Untuk mengevaluasi performa akhir, digunakan *Confusion Matrix* yang memetakan hasil prediksi model terhadap label aktual pada data uji. Analisis ini memberikan gambaran konkret mengenai presisi dan sensitivitas model dalam mengklasifikasikan kelas minoritas pada kondisi dataset yang telah diseimbangkan. Visualisasi *Confusion Matrix* tersebut ditunjukkan pada gambar 4.9.



Gambar 4.9 *Confusion Matrix* Skenario 7

Berdasarkan hasil *Confusion Matrix* pada Gambar 4.9, model berhasil mengklasifikasikan 385 data sebagai *True Negative* (TN), yaitu responden sehat yang diprediksi dengan benar sebagai sehat. Sementara itu, untuk kelas minoritas, model berhasil mendeteksi 78 data sebagai *True Positive* (TP), yakni kasus penyakit jantung (CHD) yang berhasil dikenali dengan tepat. Di sisi lain, model

mencatat 235 data sebagai *False Positive* (FP) responden sehat yang diprediksi berisiko CHD dan 34 data sebagai *False Negative* (FN) responden yang sebenarnya berisiko CHD namun diprediksi sebagai sehat. Hasil ini menunjukkan peningkatan yang signifikan dalam kemampuan model untuk mengenali kelas minoritas (positif) dibandingkan skenario-skenario sebelumnya. Hal ini merupakan dampak langsung dari penggunaan teknik *undersampling* yang berhasil menyeimbangkan distribusi data latih, sehingga model dapat mempelajari pola dari kedua kelas secara lebih proporsional.

Setelah melalui evaluasi *Confusion Matrix*, langkah berikutnya adalah melakukan pengukuran performa model menggunakan metrik *Accuracy, Precision, Recall, F1-Score,* dan ROC-AUC. Penggunaan metrik-metrik ini memberikan gambaran komprehensif mengenai kinerja model, terutama dalam memvalidasi efektivitas penyeimbangan kelas pada dataset. Hasil evaluasi performa model untuk skenario *Random Undersampling* + RFE disajikan pada Tabel 4.31.

Tabel 4.31 Hasil Prediksi Model

Metric	**Nilai**
Accuracy	63.25%
Precision	24.92%
Recall	69.64%
F1-Score	36.71%
ROC-AUC	70.77%

Berdasarkan Tabel 4.31, model memperoleh nilai *Accuracy* sebesar 63.25%. Meskipun angka akurasi ini terlihat lebih rendah dibandingkan skenario sebelumnya, nilai ini justru menunjukkan distribusi prediksi yang lebih objektif karena dataset telah diseimbangkan. Nilai *Precision* sebesar 24.92% membuktikan

dari seluruh prediksi positif yang dihasilkan, 24.92% merupakan prediksi yang tepat.

Di sisi lain, model mencatatkan nilai *Recall* yang sangat baik, yaitu sebesar 69.64%. Angka ini membuktikan bahwa penerapan *Random Undersampling* telah berhasil membuat model menjadi jauh lebih sensitif dalam mengidentifikasi kasus penyakit jantung dibandingkan skenario tanpa penyeimbangan data. Nilai *F1-Score* sebesar 36.71% merepresentasikan keseimbangan yang lebih baik antara *precision* dan *recall* dalam mengenali kelas minoritas. Selain itu, nilai ROC-AUC sebesar 70.77% menunjukkan bahwa model memiliki kemampuan diskriminasi yang lebih kuat dalam membedakan antara kelas sehat dan sakit. Secara keseluruhan, kombinasi *undersampling* dan seleksi fitur terbukti mampu meningkatkan kemampuan model secara signifikan dalam mendeteksi kasus risiko penyakit jantung.

4.2.8 Hasil Uji Coba Skenario 8 (*Random Undersampling* dengan RFE *by Threshold*)

Pada Skenario 8, pengujian dilakukan dengan mengombinasikan teknik penyeimbangan data *Random Undersampling*dan metode seleksi fitur berbasis *feature importance* dengan pendekatan *threshold*. Tujuan dari skenario ini adalah untuk mengevaluasi performa model XGBoost dalam menangani data yang telah diseimbangkan serta direduksi dimensinya berdasarkan nilai kontribusi fitur yang melebihi ambang batas tertentu. Nilai *threshold* sebesar 0,07 digunakan sebagai acuan yang sama seperti pada skenario sebelumnya untuk mengurangi jumlah fitur

pada pengujian lanjutan. Sama seperti pada skenario sebelumnya, pengujian ini juga dibagi menjadi dua sub-skenario, yaitu Skenario 8a yang menggunakan *threshold* sebesar 0,07 dan Skenario 8b yang menggunakan *threshold* berdasarkan nilai rata-rata *feature importance*.

a. Hasil Uji Coba Skenario 8a (*Random Undersampling* dengan RFE *by Threshold* 0.07)

Pada skenario kedelapan, dilakukan penggabungan dua pendekatan seleksi fitur, yakni *Recursive Feature Elimination* (RFE) dan seleksi berbasis ambang batas (*threshold*), yang dikombinasikan dengan Teknik *Random Undersampling*. Pendekatan *hybrid* ini diterapkan untuk memastikan bahwa hanya fitur dengan daya prediksi tinggi yang disertakan dalam model, sekaligus mengatasi bias akibat ketidakseimbangan kelas pada dataset asli.

Seleksi fitur dilakukan dengan menyaring variabel melalui metode RFE untuk mendapatkan peringkat kepentingan, yang kemudian diperketat dengan ambang batas (*threshold*) sebesar 0.07 untuk memfilter fitur yang benar-benar memberikan kontribusi signifikan terhadap target. Proses ini bertujuan untuk menyederhanakan kompleksitas data agar model XGBoost lebih fokus mempelajari pola-pola kunci dari variabel yang paling relevan. Hasil audit fitur pada skenario ini disajikan dalam tabel 4.32.

Tabel 4.32 Hasil RFE *by Threshold* 0.07

No	*Feature Name*	*Final Rank*	*Gain Score*	*Status*
1	*prevalentHyp*	1	0.1434	*Retained*
2	*age*	2	0.1072	*Retained*
3	*glucose*	3	0.0744	*Retained*
4	*male*	4	0.0725	*Retained*
5	*currentSmoker*	5	0.0708	*Retained*

No	*Feature Name*	*Final Rank*	*Gain Score*	*Status*
6	*cigsPerDay*	6	0.0688	*Eliminated*
7	*sysBP*	7	0.0666	*Eliminated*
8	*education*	8	0.0639	*Eliminated*
9	*totChol*	9	0.0625	*Eliminated*
10	*diaBP*	10	0.0622	*Eliminated*
11	*BPMeds*	11	0.0584	*Eliminated*
12	BMI	12	0.0558	*Eliminated*
13	*heartRate*	13	0.0545	*Eliminated*
14	diabetes	14	0.0246	*Eliminated*
15	*prevalentStroke*	15	0.0145	*Eliminated*

Berdasarkan Tabel 4.32, dari 15 fitur awal, model berhasil mempertahankan 5 fitur utama yang memenuhi ambang batas *gain score* >= 0.07. Fitur *prevalentHyp* dengan skor 0.1434 menjadi variabel paling berpengaruh. Sementara itu, fitur lainnya dieliminasi karena dianggap kurang memberikan kontribusi informatif yang signifikan bagi performa prediksi model.

Langkah selanjutnya adalah penyeimbangan data menggunakan *Random Undersampling*. Teknik ini dilakukan dengan memangkas jumlah sampel pada kelas mayoritas hingga setara dengan kelas minoritas (rasio 1:1), sehingga model tidak cenderung hanya mempelajari satu kelas selama proses pelatihan. Perbandingan distribusi data pada skenario ini disajikan pada Tabel 4.33.

Tabel 4.33 Hasil *Random Undersampling*

Kategori	Latih Asli	Latih *Undersampled*	Uji (*Original*)
Sehat (Kelas 0)	2479	445	620
Sakit (Kelas 1)	445	445	112
Total	2924	890	732

Berdasarkan Tabel 4.33, terlihat bahwa setelah proses *undersampling*, total data latih menjadi 890 sampel yang terbagi secara

proporsional. Perlu ditegaskan bahwa data pengujian (*test set*) tetap mempertahankan distribusi aslinya sebanyak 732 data, guna memastikan evaluasi model mencerminkan performa pada kondisi data riil di lapangan.

Setelah data diseimbangkan, dilakukan optimasi parameter (*hyperparameter tuning*) menggunakan metode *GridSearchCV* dengan skema 5-*fold cross-validation*. Metode ini menguji 729 kombinasi parameter secara sistematis untuk mencari konfigurasi terbaik bagi model XGBoost. Total proses *fitting* yang dijalankan selama optimasi mencapai 3.645 kali, memastikan bahwa setiap kandidat parameter divalidasi dengan cermat. Hasil konfigurasi parameter optimal disajikan pada Tabel 4.34.

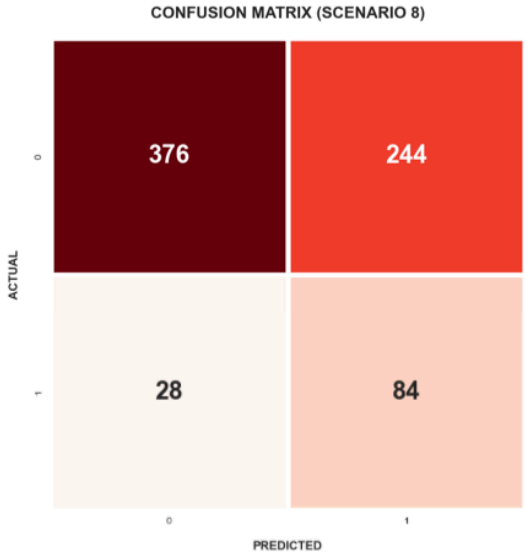
Tabel 4.34 Hasil Seleksi *Hyperparameter Tunning*

Hyperparameter	**Nilai Optimal**
Learning Rate	0.01
Max Depth	3
N Estimators	100
Subsample	0.9
Colsample Bytree	1
Gamma	0

Berdasarkan Tabel 4.34, parameter optimal yang diperoleh adalah *learning rate* 0.01, *max depth* 3, dan *n_estimators* 100. Nilai *learning rate* sebesar 0.01 menunjukkan bahwa model melakukan pembelajaran secara bertahap untuk menjamin stabilitas konvergensi. *Max depth* sebesar 3 dipilih untuk membangun pohon keputusan yang relatif dangkal guna mencegah terjadinya *overfitting*. Parameter subsample sebesar 0.9 dan *colsample bytree* sebesar 1 memastikan keragaman data dan fitur yang digunakan pada setiap iterasi *boosting*. Sementara itu, nilai *gamma* 0

menandakan tidak adanya restriksi *loss reduction* yang diterapkan dalam proses pembentukan *node split*.

Langkah terakhir adalah mengevaluasi kinerja model melalui *Confusion Matrix*. Evaluasi ini bertujuan untuk memetakan jumlah prediksi benar (*True Positive* dan *True Negative*) serta kesalahan klasifikasi (*False Positive* dan *False Negative*) pada data uji. Visualisasi hasil *Confusion Matrix* pada skenario kedelapan ditampilkan pada gambar 4.10.



Gambar 4.10 *Confusion Matrix* Skenario 8a

Berdasarkan hasil *Confusion Matrix* pada Gambar 4.10, model berhasil mengklasifikasikan 376 data sebagai *True Negative* (TN), yaitu responden sehat yang diprediksi dengan benar. Sementara itu, untuk kelas minoritas, model mendeteksi 84 data sebagai *True Positive* (TP), yakni kasus penyakit jantung (CHD) yang berhasil dikenali dengan tepat. Di sisi lain, terdapat 244 data sebagai *False Positive* (FP) responden sehat yang diprediksi berisiko CHD dan 28 data sebagai *False Negative* (FN)

responden yang sebenarnya berisiko CHD namun diprediksi sehat. Hasil ini menunjukkan peningkatan yang signifikan dalam sensitivitas model untuk mengenali kelas minoritas dibandingkan skenario tanpa penyeimbangan data. Peningkatan ini merupakan dampak langsung dari Teknik *undersampling* yang berhasil menyeimbangkan distribusi data latih, sehingga model mampu mempelajari pola dari kedua kelas secara lebih proporsional.

Setelah melalui evaluasi *Confusion Matrix*, langkah berikutnya adalah mengukur performa model menggunakan metrik *Accuracy, Precision, Recall, F1-Score,* dan ROC-AUC. Metrik-metrik ini memberikan gambaran komprehensif mengenai kinerja model, terutama dalam memvalidasi efektivitas kombinasi teknik seleksi fitur dan penyeimbangan kelas. Hasil evaluasi performa model disajikan pada Tabel 4.35.

Tabel 4.35 Hasil Prediksi Model

Metric	**Nilai**
Accuracy	62.84%
Precision	25.61%
Recall	75.00%
F1-Score	38.18%
ROC-AUC	71.58%

Berdasarkan Tabel 4.35, model memperoleh nilai *Accuracy* sebesar 62.84%. Meskipun nilai akurasi ini terlihat moderat, hal tersebut merepresentasikan distribusi prediksi yang lebih objektif pada dataset yang telah diseimbangkan. Nilai *Precision* sebesar 25.61% mengindikasikan bahwa dari seluruh prediksi positif yang dihasilkan, 25.61% merupakan prediksi yang tepat.

Di sisi lain, model mencatatkan nilai *Recall* yang sangat tinggi, yaitu sebesar 75.00%. Angka ini membuktikan bahwa kombinasi teknik *Random Undersampling* dengan seleksi fitur hibrid telah menjadikan model jauh lebih sensitif dalam mengidentifikasi kasus penyakit jantung dibandingkan skenario lainnya. Nilai *F1-Score* sebesar 38.18% merepresentasikan keseimbangan yang jauh lebih baik antara *precision* dan *recall* dalam mengenali kelas minoritas. Selain itu, nilai ROC-AUC sebesar 71.58% menunjukkan bahwa model memiliki kemampuan diskriminasi yang kuat dalam membedakan antara kelas sehat dan sakit. Secara keseluruhan, skenario kedelapan ini memberikan performa terbaik dalam hal deteksi kasus penyakit jantung, menjadikannya model yang paling efektif untuk meminimalisir risiko kesalahan deteksi pada kelas minoritas.

b. Hasil Uji Coba Skenario 8b (*Random Undersampling* dengan RFE *by Threshold Mean*)

Pada Pada skenario kedelapan bagian b (Skenario 8b), dilakukan penggabungan dua pendekatan seleksi fitur, yakni *Recursive Feature Elimination* (RFE) dan seleksi berbasis nilai ambang batas rata-rata (*threshold mean*), yang kemudian dikombinasikan dengan teknik *Random Undersampling*. Pendekatan ini diterapkan untuk memastikan bahwa hanya fitur dengan daya prediksi tinggi yang disertakan dalam model, sekaligus mengatasi bias akibat ketidakseimbangan kelas pada dataset asli.

Seleksi fitur dilakukan dengan menyaring variabel melalui metode RFE untuk mendapatkan peringkat kepentingan, yang kemudian diperketat

dengan ambang batas berdasarkan nilai rata-rata kontribusi (*mean importance*) sebesar 0.0667 untuk memfilter fitur yang benar-benar memberikan kontribusi signifikan terhadap target. Proses ini bertujuan untuk menyederhanakan kompleksitas data agar model XGBoost lebih fokus mempelajari pola-pola kunci dari variabel yang paling relevan. Hasil audit fitur pada skenario ini disajikan dalam tabel 4.36.

Tabel 4.36 Hasil Seleksi Fitur menggunakan RFE *by Threshold Mean*)

No	*Feature Name*	*Final Rank*	*Gain Score*	*Status*
1	*prevalentHyp*	1	0.1434	*Retained*
2	*age*	2	0.1072	*Retained*
3	*glucose*	3	0.0744	*Retained*
4	*male*	4	0.0725	*Retained*
5	*currentSmoker*	5	0.0708	*Retained*
6	*cigsPerDay*	6	0.0688	*Retained*
7	*sysBP*	7	0.0666	*Eliminated*
8	*education*	8	0.0639	*Eliminated*
9	*totChol*	9	0.0625	*Eliminated*
10	*diaBP*	10	0.0622	*Eliminated*
11	*BPMeds*	11	0.0584	*Eliminated*
12	BMI	12	0.0558	*Eliminated*
13	*heartRate*	13	0.0545	*Eliminated*
14	diabetes	14	0.0246	*Eliminated*
15	*prevalentStroke*	15	0.0145	*Eliminated*

Berdasarkan Tabel 4.36, dari 15 fitur awal yang diuji, proses seleksi ini berhasil mempertahankan 6 fitur utama yang memiliki kontribusi optimal berdasarkan batas ambang *mean*. Fitur *prevalentHyp* dengan *gain score* sebesar 0.1434 menempati peringkat pertama (*Final Rank* 1) sebagai variabel yang paling berpengaruh secara dominan dalam model. Kontribusi signifikan berikutnya diikuti secara berurutan oleh fitur *age* (0.1072), *glucose* (0.0744), *male* (0.0725), *currentSmoker* (0.0708), hingga fitur *cigsPerDay* (0.0688) yang menempati posisi terakhir dalam kluster fitur yang dipertahankan (*retained*). Sementara itu, 9 fitur sisanya mulai dari

peringkat ketujuh yaitu *sysBP* (0.0666) hingga fitur dengan kontribusi terendah yaitu *prevalentStroke* (0.0145) secara resmi dieliminasi (*eliminated*) dari tahapan pemodelan. Variabel-variabel tersebut dipangkas karena nilai kontribusinya berada di bawah standar ambang batas informatif yang dibutuhkan oleh model XGBoost pada skenario ini.

Langkah selanjutnya adalah penyeimbangan data menggunakan *Random Undersampling*. Teknik ini dilakukan dengan memangkas jumlah sampel pada kelas mayoritas hingga setara dengan kelas minoritas (rasio 1:1), sehingga model tidak terobsesi hanya pada satu kelas saja selama proses pelatihan. Perbandingan distribusi data pada skenario ini disajikan pada Tabel 4.37.

Tabel 4.37 Hasil *Random Undersampling*

Kategori	**Latih Asli**	**Latih *Undersampled***	**Uji (*Original*)**
Sehat (Kelas 0)	2479	445	620
Sakit (Kelas 1)	445	445	112
Total	2924	890	732

Berdasarkan Tabel 4.37, terlihat bahwa setelah proses *undersampling*, total data latih menjadi 890 sampel yang terbagi secara proporsional. Perlu ditegaskan bahwa data pengujian (*test set*) tetap mempertahankan distribusi aslinya sebanyak 732 data, guna memastikan evaluasi model mencerminkan performa pada kondisi data riil di lapangan.

Setelah data diseimbangkan, dilakukan optimasi parameter (*hyperparameter tuning*) menggunakan metode *GridSearchCV* dengan skema 5-*fold cross-validation*. Metode ini menguji 729 kombinasi parameter secara sistematis untuk mencari konfigurasi terbaik bagi model

XGBoost. Total proses *fitting* yang dijalankan selama optimasi mencapai 3.645 kali, memastikan bahwa setiap kandidat parameter divalidasi dengan cermat. Hasil konfigurasi parameter optimal disajikan pada Tabel 4.38.

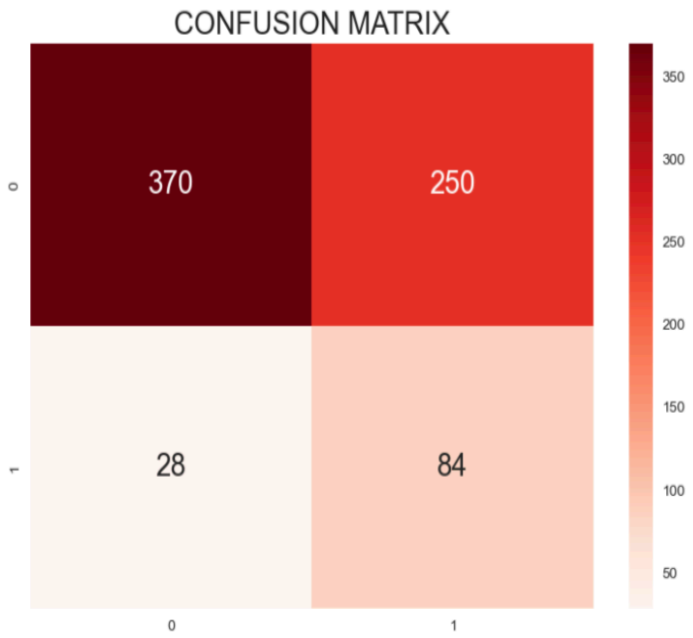
Tabel 4.38 Hasil Seleksi *Hyperparameter Tunning*

Hyperparameter	**Nilai Optimal**
Learning Rate	0.01
Max Depth	3
N Estimators	100
Subsample	1.0
Colsample Bytree	1.0
Gamma	0

Berdasarkan Tabel 4.38, parameter optimal yang diperoleh adalah *learning rate* 0,01, *max depth* 3, dan *n_estimators* 100. Nilai *learning rate* sebesar 0,01 menunjukkan bahwa model melakukan pembelajaran secara bertahap untuk menjamin stabilitas konvergensi. *Max depth* sebesar 3 dipilih untuk membangun pohon keputusan yang relatif dangkal guna mencegah terjadinya *overfitting*. Parameter *subsample* sebesar 1,0 dan *colsample bytree* sebesar 1,0 memastikan bahwa seluruh data dan fitur digunakan pada setiap iterasi *boosting* untuk memaksimalkan informasi yang diserap oleh model. Sementara itu, nilai *gamma* 0 menandakan tidak adanya restriksi *loss reduction* yang diterapkan dalam proses pembentukan *node split*. Kombinasi parameter ini diharapkan mampu mengoptimalkan kemampuan generalisasi model dalam menangani dataset yang telah diseimbangkan.

Langkah terakhir adalah mengevaluasi kinerja model melalui *Confusion Matrix*. Evaluasi ini bertujuan untuk memetakan jumlah prediksi benar (*True Positive* dan *True Negative*) serta kesalahan klasifikasi (*False*

Positive dan *False Negative*) pada data uji. Visualisasi hasil *Confusion Matrix* pada skenario kedelapan ditampilkan pada gambar 4.11.



Gambar 4.11 *Confusion Matrix* Skenario 8b

Berdasarkan hasil *Confusion Matrix* pada Gambar 4.11, model berhasil mengklasifikasikan 370 data sebagai *True Negative* (TN), yaitu responden sehat yang diprediksi dengan benar. Sementara itu, untuk kelas minoritas, model mendeteksi 84 data sebagai *True Positive* (TP), yakni kasus penyakit jantung (CHD) yang berhasil dikenali dengan tepat. Di sisi lain, terdapat 250 data sebagai *False Positive* (FP) yaitu responden sehat yang diprediksi berisiko CHD, dan 28 data sebagai *False Negative* (FN) yaitu responden yang sebenarnya berisiko CHD namun diprediksi sehat. Hasil ini menunjukkan peningkatan yang signifikan dalam sensitivitas model untuk mengenali kelas minoritas dibandingkan skenario tanpa penyeimbangan data. Peningkatan ini merupakan dampak langsung dari

teknik *undersampling* yang berhasil menyeimbangkan distribusi data latih, sehingga model mampu mempelajari pola dari kedua kelas secara lebih proporsional.

Setelah melalui evaluasi *Confusion Matrix*, langkah berikutnya adalah mengukur performa model menggunakan metrik *Accuracy, Precision, Recall, F1-Score,* dan ROC-AUC. Metrik-metrik ini memberikan gambaran komprehensif mengenai kinerja model, terutama dalam memvalidasi efektivitas kombinasi teknik seleksi fitur dan penyeimbangan kelas. Hasil evaluasi performa model disajikan pada Tabel 4.39.

Tabel 4.39 Hasil Prediksi Model

Metric	**Nilai**
Accuracy	62.02%
Precision	25.15%
Recall	75.00%
F1-Score	37.67%
ROC-AUC	70.93%

Berdasarkan Tabel 4.39, model memperoleh nilai *Accuracy* sebesar 62.02%. Meskipun nilai akurasi ini terlihat moderat, hal tersebut merepresentasikan distribusi prediksi yang lebih objektif pada *dataset* yang telah diseimbangkan. Nilai *Precision* sebesar 25.15% mengindikasikan bahwa dari seluruh prediksi positif yang dihasilkan, 25.15% merupakan prediksi yang tepat.

Di sisi lain, model mencatatkan nilai *Recall* yang sangat tinggi, yaitu sebesar 75.00%. Angka ini membuktikan bahwa kombinasi teknik *Random Undersampling* dengan seleksi fitur telah menjadikan model jauh lebih sensitif dalam mengidentifikasi kasus penyakit jantung dibandingkan

skenario lainnya. Nilai *F1-Score* sebesar 37.67% merepresentasikan keseimbangan yang jauh lebih baik antara *precision* dan *recall* dalam mengenali kelas minoritas. Selain itu, nilai ROC-AUC sebesar 70.93% menunjukkan bahwa model memiliki kemampuan diskriminasi yang kuat dalam membedakan antara kelas sehat dan sakit. Secara keseluruhan, skenario kedelapan ini memberikan performa terbaik dalam hal deteksi kasus penyakit jantung, menjadikannya model yang paling efektif untuk meminimalisir risiko kesalahan deteksi pada kelas minoritas.

4.2.9 Hasil Uji Coba Skenario 9 (*Random Undersampling* Non RFE)

Pada skenario kesembilan, model XGBoost dikembangkan menggunakan dataset *baseline* tanpa melibatkan proses seleksi fitur (RFE). Pendekatan ini bertujuan untuk menguji sejauh mana model mampu mengekstraksi pola dari seluruh informasi fitur (15 fitur asli) secara utuh, guna mengetahui apakah eliminasi fitur memang diperlukan atau justru menghilangkan informasi berharga bagi model.

Meskipun fitur digunakan secara keseluruhan, teknik penyeimbangan data *Random Undersampling* tetap diterapkan pada data latih (*training set*). Hal ini dilakukan untuk mengatasi bias kelas dominan agar model lebih sensitif terhadap karakteristik kelas minoritas (Sakit/CHD). Distribusi data sebelum dan sesudah proses *undersampling* disajikan pada Tabel 4.40.

Tabel 4.40 Hasil *Random Undersampling*

Kategori	Latih Asli	Latih *Undersampled*	Uji (*Original*)
Sehat (Kelas 0)	2479	445	620
Sakit (Kelas 1)	445	445	112
Total	2924	890	732

Berdasarkan Tabel 4.40, terlihat bahwa proses *undersampling* berhasil menyeimbangkan rasio kelas menjadi 1:1 (masing-masing 445 sampel). Sementara itu, data pengujian (*test set*) tetap mempertahankan distribusi aslinya untuk menjaga validitas evaluasi pada kondisi data yang sebenarnya.

Setelah dataset siap, langkah berikutnya adalah optimasi *hyperparameter* menggunakan *GridSearchCV* dengan skema 5-*fold cross-validation*. Proses ini melibatkan pengujian 729 kombinasi parameter dengan total 3.645 kali fitting model. Konfigurasi parameter terbaik yang dihasilkan disajikan pada Tabel 4.41.

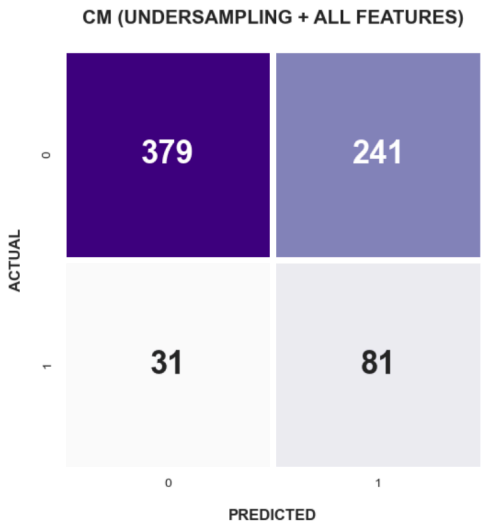
Tabel 4.41 Hasil Seleksi *Hyperparameter Tunning*

Hyperparameter	**Nilai Optimal**
Learning Rate	0.01
Max Depth	3
N Estimators	100
Subsample	0.8
Colsample Bytree	1.0
Gamma	0

Berdasarkan Tabel 4.41, memperoleh hasil *learning rate* 0.01 yang memastikan stabilitas dalam pembaruan bobot model, sedangkan *max depth* 3 dipilih untuk menjaga model tetap sederhana guna mencegah *overfitting*. Dengan nilai *colsample bytree* 1.0, model memanfaatkan seluruh fitur yang ada untuk membangun setiap pohon keputusan, sementara *subsample* 0.8 berperan dalam meningkatkan keragaman data pelatihan. Nilai *gamma* 0 menunjukkan tidak adanya restriksi *loss reduction* yang diterapkan dalam proses *node split*.

Untuk melihat performa secara mendalam, dilakukan analisis menggunakan *Confusion Matrix* pada data uji. *Confusion Matrix* memberikan rincian jumlah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN)

untuk memetakan keakuratan klasifikasi pada setiap kelas. Hasil visualisasi tersebut ditampilkan pada Gambar 4.12.



Gambar 4.12 *Confusion Matrix* Skenario 9

Berdasarkan *Confusion Matrix* pada Gambar 4.12, model berhasil mengklasifikasikan 379 data sebagai *True Negative* (TN), yaitu responden sehat yang diprediksi dengan benar. Sementara itu, untuk kelas minoritas, model mendeteksi 81 data sebagai *True Positive* (TP), yakni kasus penyakit jantung (CHD) yang berhasil dikenali dengan tepat. Di sisi lain, model mencatat 241 data sebagai *False Positive* (FP) responden sehat yang diprediksi berisiko CHD dan 31 data sebagai *False Negative* (FN) responden yang sebenarnya berisiko CHD namun diprediksi sehat. Hasil ini menunjukkan bahwa model memiliki sensitivitas yang baik dalam mendeteksi kelas minoritas. Meskipun tingkat *False Positive* cukup tinggi, penggunaan Teknik *Random Undersampling* terbukti berhasil menyeimbangkan distribusi data latih sehingga model tidak lagi bias terhadap kelas mayoritas.

Setelah proses klasifikasi selesai, tahap selanjutnya adalah melakukan evaluasi performa model secara komprehensif menggunakan metrik *Accuracy, Precision, Recall, F1-Score,* dan ROC-AUC. Evaluasi ini bertujuan untuk mengukur sejauh mana model mampu melakukan klasifikasi secara akurat pada data pengujian yang memiliki distribusi kelas asli. Hasil evaluasi performa model untuk skenario *Random Undersampling* tanpa RFE disajikan pada Tabel 4.42.

Tabel 4.42 Hasil Prediksi Model

Metric	Nilai
Accuracy	62.84%
Precision	25.16%
Recall	72.32%
F1-Score	37.33%
ROC-AUC	70.37%

Berdasarkan Tabel 4.42, model memperoleh nilai *Accuracy* sebesar 62.84%. Perlu dipahami bahwa angka akurasi yang moderat ini merepresentasikan kinerja model yang lebih objektif pada dataset yang telah diseimbangkan. Nilai *Precision* sebesar 25.16% menunjukkan bahwa dari seluruh prediksi positif, sekitar 25.16% merupakan prediksi yang akurat.

Di sisi lain, model mencatatkan nilai *Recall* yang cukup tinggi, yaitu sebesar 72.32%, yang membuktikan bahwa penerapan *Random Undersampling* secara signifikan meningkatkan kemampuan model dalam mendeteksi sebagian besar kasus penyakit jantung pada dataset. Nilai *F1-Score* sebesar 37.33% merepresentasikan keseimbangan yang moderat antara *precision* dan *recall*. Selain itu, nilai ROC-AUC sebesar 70.37% mengindikasikan kemampuan diskriminasi model yang cukup baik dalam membedakan antara kelas sehat dan sakit. Secara keseluruhan, hasil ini menunjukkan bahwa penggunaan *Random Undersampling*

tanpa proses seleksi fitur (RFE) tetap mampu menghasilkan performa deteksi kelas minoritas yang kuat, meskipun masih terdapat ruang untuk optimasi pada metrik *precision* guna mengurangi *False Positive*.

4.3 Pembahasan

Setelah seluruh skenario eksperimen dilakukan, langkah selanjutnya adalah melakukan perbandingan performa model pada setiap skenario. Perbandingan ini bertujuan untuk menganalisis pengaruh penerapan teknik *resampling* dan seleksi fitur terhadap kinerja model XGBoost dalam melakukan klasifikasi risiko penyakit serangan jantung. Evaluasi dilakukan menggunakan beberapa metrik, yaitu *Accuracy*, *Precision*, *Recall*, F1-Score, dan ROC-AUC.

Seluruh skenario yang diuji terdiri atas sembilan skenario utama yang menghasilkan dua belas kombinasi eksperimen. Jumlah dua belas kombinasi tersebut diperoleh karena pada tiga skenario utama, yaitu Skenario 2, Skenario 5, dan Skenario 8, dilakukan pengujian lebih lanjut yang dibagi menjadi dua variasi sub-skenario berbasis *threshold*. Variasi tersebut meliputi sub-skenario (a) yang menggunakan nilai *threshold feature importance* sebesar 0,07 dan sub-skenario (b) yang menggunakan nilai rata-rata (*mean*) *feature importance*. Kombinasi keseluruhan ini mencakup metode SMOTE, *Random Undersampling* (RUS), serta kondisi tanpa penyeimbangan data (*Pure*). Selain itu, digunakan beberapa pendekatan seleksi fitur, yaitu *Recursive Feature Elimination* dengan pemilihan 10 fitur terbaik, RFE berbasis *threshold feature importance* sebesar 0,07, serta RFE berbasis *threshold* berdasarkan nilai rata-rata (*mean*) *feature importance*. Ketiga

pendekatan ini digunakan untuk menganalisis pengaruh jumlah fitur yang dipertahankan terhadap performa model XGBoost.

4.3.1 Perbandingan Hasil Skenario Uji Coba

Sebagai hasil dari seluruh skenario uji coba yang telah dilakukan sebelumnya, berikut ditampilkan skenario perbandingan performa model XGBoost pada setiap skenario eksperimen berdasarkan metrik evaluasi yang digunakan. Perbandingan tersebut disajikan pada Tabel 4.43.

Tabel 4.43 Perbandingan Hasil Skenario Uji Coba

No.	*Balancing*	*Accuracy*	*Precision*	*Recall*	*F1-Score*	ROC-AUC
1	SMOTE	82.10%	37.66%	25.89%	30.69%	63.73%
2	S2a: SMOTE+*Threshold* (0.07)	71.99%	19.61%	26.79%	22.64%	59.34%
3	S2b: SMOTE+*Threshold* (Mean)	74.86%	22.31%	25.89%	23.97%	56.31%
4	S3: SMOTE+*Full*	79.78%	25.68%	16.96%	20.43%	63.06%
5	S4: *Pure*+RFE (10)	82.92%	39.34%	21.43%	27.75%	64.75%
6	S5a: *Pure+Threshold* (0.07)	78.96%	23.75%	16.96%	19.79%	64.35%
7	S5b: *Pure+Threshold* (*Mean*)	78.69%	23.17%	16.96%	19.59%	59.99%
8	S6: *Pure+Full*	83.88%	42.11%	14.29%	21.33%	66.79%
9	S7: RUS+RFE (10)	63.25%	24.92%	69.64%	36.71%	70.77%
10	S8a: RUS+*Threshold* (0.07)	62.84%	25.61%	75.00%	38.18%	71.58%
11	S8b: RUS+*Threshold* (Mean)	62.02%	25.15%	75.00%	37.67%	70.93%
12	S9: RUS+*Full* (*Default*)	62.84%	25.16%	72.32%	37.33%	70.37%

Tabel 4.43 menunjukkan adanya perbedaan performa yang cukup signifikan antar skenario pengujian. Konfigurasi model *Pure* tanpa *resampling* (S4, S5a, S5b, S6) dominan menghasilkan tingkat *accuracy* tertinggi, memuncak pada 83,88%

(S6). Namun, tingginya *accuracy* ini berbanding terbalik dengan nilai *Recall* yang sangat rendah (14,29%), mengindikasikan bahwa model gagal mendeteksi kelas minoritas secara optimal. Secara ilmiah, fenomena ini terjadi karena model mengalami bias akibat *imbalanced data*, di mana jumlah sampel kelas mayoritas jauh lebih mendominasi. Model cenderung memprediksi kelas mayoritas untuk mengejar kecocokan global, sehingga angka *accuracy* 83,88% tersebut dikategorikan sebagai akurasi semu. Nilai *Recall* yang hanya 14,29% membawa dampak fatal dalam bidang medis yaitu dari total 112 pasien yang diprediksi beresiko terkenaa penyakit jantung, model pada Skenario 6 hanya mampu menjaring 16 orang, sementara 96 orang lainnya lolos sebagai *False Negative* tanpa mendapatkan penanganan medis.

Sebaliknya, penerapan teknik *Random Under-Sampling* (RUS) pada skenario S7, S8a, S8b, dan S9 memang berdampak pada penurunan *accuracy* ke kisaran 62%–63%, tetapi berhasil meningkatkan kemampuan model secara drastis. Hal ini dibuktikan dengan lonjakan nilai *Recall* yang mencapai 75,00% serta skor ROC-AUC tertinggi sebesar 71,58% pada Skenario 8a (RUS + *Threshold* 0.07). Keberhasilan angka *Recall* mencapai 75,00% ini menunjukkan bahwa reduksi data mayoritas melalui teknik RUS efektif memaksa model untuk lebih peka dalam mengenali karakteristik unik dari kelompok kelas minoritas.

Penurunan nilai *accuracy* menjadi 62,84% pada Skenario 8a merupakan hasil yang normal setelah diterapkannya teknik *Random Under-Sampling* (RUS). Ketika jumlah data mayoritas dikurangi agar lebih seimbang dengan data minoritas, model menjadi lebih fokus untuk mengenali pasien yang berisiko mengalami

penyakit jantung. Akibatnya, kemampuan model dalam mendeteksi kasus positif meningkat secara signifikan yang ditunjukkan oleh nilai *recall* sebesar 75,00%. Meskipun demikian, peningkatan kemampuan dalam mendeteksi pasien yang berisiko juga menyebabkan model lebih sering memberikan prediksi positif. Dampaknya, sebagian pasien yang sebenarnya tidak berisiko ikut diprediksi sebagai pasien berisiko. Kondisi ini menyebabkan jumlah *false positive* meningkat sehingga nilai *accuracy* dan *precision* menjadi lebih rendah dibandingkan beberapa skenario lainnya. Dengan kata lain, model menjadi lebih berhati-hati dalam mengambil keputusan sehingga lebih banyak pasien yang dicurigai berisiko daripada yang seharusnya.

Dalam konteks deteksi dini penyakit jantung, kondisi tersebut masih dapat diterima. Pasien yang terdeteksi berisiko masih dapat menjalani pemeriksaan lanjutan untuk memastikan kondisi kesehatannya. Sebaliknya, apabila pasien yang sebenarnya berisiko tidak berhasil terdeteksi oleh model, maka pasien tersebut berpotensi tidak mendapatkan penanganan lebih awal. Oleh karena itu, kemampuan model dalam menemukan sebanyak mungkin pasien yang berisiko menjadi aspek yang sangat penting dalam penelitian ini.

Keunggulan Skenario 8a tidak hanya terlihat dari nilai *recall* yang tinggi, tetapi juga dari nilai *F1-Score* sebesar 38,18% dan *ROC-AUC* sebesar 71,58%, yang merupakan nilai tertinggi dibandingkan seluruh skenario pengujian. Nilai *F1-Score* yang tertinggi menunjukkan bahwa Skenario 8a mampu memberikan keseimbangan terbaik antara kemampuan model dalam menemukan pasien yang berisiko (*recall*) dan ketepatan model ketika memberikan prediksi positif

(*precision*). Hal ini menunjukkan bahwa peningkatan *recall* yang diperoleh tidak mengorbankan *precision* secara berlebihan.

Selain itu, nilai ROC-AUC sebesar 71,58% menunjukkan bahwa Skenario 8a memiliki kemampuan paling baik dalam membedakan pasien yang berisiko penyakit jantung dan pasien yang tidak berisiko. Hasil ini menunjukkan bahwa model mampu mengenali pola dari kedua kelompok data dengan lebih baik dibandingkan skenario lainnya. Semakin tinggi nilai *ROC-AUC*, semakin baik kemampuan model dalam memberikan prediksi yang tepat pada berbagai kondisi pengujian.

Hasil tersebut juga didukung oleh hasil seleksi fitur pada skenario 8a yaitu RUS dengan RFE by *Threshold* 0,07. Pada skenario ini, model mempertahankan lima fitur yang memiliki nilai *importance* di atas ambang batas 0,07, yaitu *prevalentHyp*, *age*, *glucose*, *male*, dan *currentSmoker*. Sementara itu, fitur *cigsPerDay*, *sysBP*, *education*, *totChol*, *diaBP*, *BPMeds*, *BMI*, *heartRate*, *diabetes*, dan *prevalentStroke* tidak terpilih karena memiliki nilai *importance* di bawah *threshold* yang ditetapkan.

Dari sudut pandang klasifikasi, hasil tersebut menunjukkan bahwa model lebih banyak memanfaatkan lima fitur yang dianggap paling berkontribusi dalam membedakan kelas risiko penyakit jantung pada *dataset* yang digunakan. Namun demikian, hasil seleksi fitur ini tidak dapat dijadikan dasar bahwa fitur yang dieliminasi tidak penting secara medis. Meskipun demikian, berdasarkan *Fact Sheet* dari (World Health Organization, 2025), sebagian besar dari fitur tersebut, yaitu fitur *cigsPerDay*, *sysBP, diaBP*, *totChol*, *BMI*, serta *diabetes* secara klinis

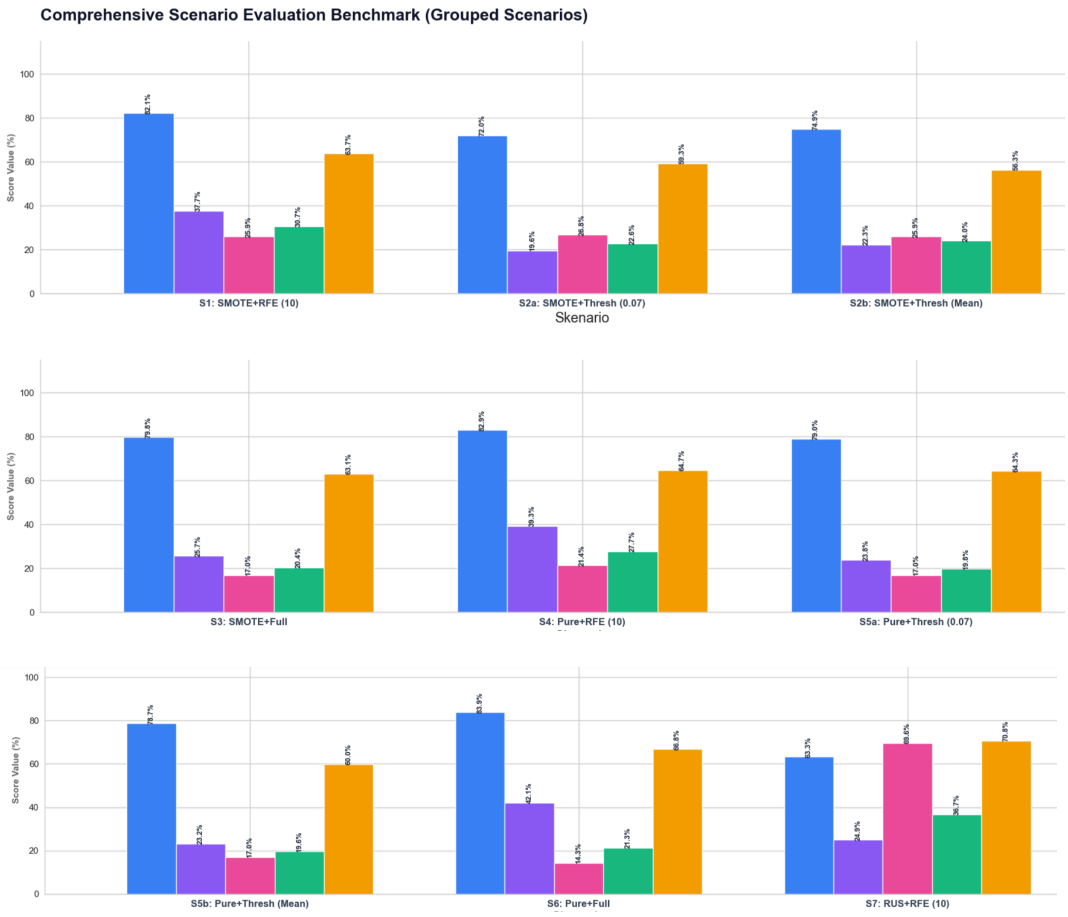
merupakan faktor risiko utama yang diakui secara internasional dalam memicu penyakit jantung. Tidak terpilihnya fitur-fitur tersebut pada penelitian ini dipengaruhi oleh karakteristik *dataset*, distribusi data, serta hubungan antar fitur yang menyebabkan kontribusinya terhadap model menjadi lebih rendah.

Oleh karena itu, hasil seleksi fitur pada penelitian ini perlu dipahami sebagai proses optimasi model klasifikasi, bukan sebagai dasar untuk mengesampingkan pentingnya suatu atribut dalam kajian medis. Dari sisi medis, seluruh atribut tetap memiliki nilai klinis yang dapat membantu proses penilaian risiko penyakit jantung. Sementara itu, dari sisi *machine learning*, hanya sebagian fitur yang memberikan kontribusi terbesar terhadap performa model pada *dataset* yang digunakan. Sehingga dalam konteks data medis, atribut tidak dapat disederhanakan sepenuhnya hanya berdasarkan hasil seleksi fitur. Penggunaan seluruh fitur tetap perlu dipertimbangkan agar hasil penelitian tetap dapat dijelaskan dengan jelas dan sesuai dengan kondisi klinis.

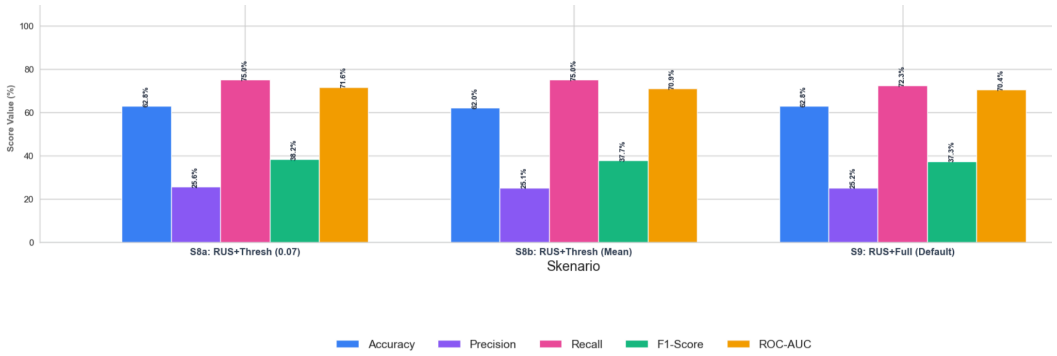
Berdasarkan seluruh metrik evaluasi yang digunakan, Skenario 8a memberikan hasil yang paling seimbang. Meskipun nilai *accuracy* yang diperoleh tidak setinggi Skenario 6 (model *Pure*), model pada Skenario 8a mampu mendeteksi jauh lebih banyak pasien yang berisiko penyakit jantung. Selain itu, Skenario 8a juga memperoleh nilai *F1-Score* dan *ROC-AUC* tertinggi, yang menunjukkan bahwa performa model tidak hanya baik pada satu metrik saja, tetapi juga konsisten pada berbagai aspek evaluasi. Oleh karena itu, Skenario 8a dipilih sebagai model terbaik dalam penelitian ini karena dinilai paling sesuai dengan

tujuan utama penelitian, yaitu mendeteksi sebanyak mungkin pasien yang berisiko mengalami penyakit jantung.

Sebagai perbandingan yang lebih jelas, visualisasi antar skenario ini diilustrasikan pada Gambar 4.13 dan Gambar 4.14. Melalui representasi grafik tersebut, tren pergerakan fluktuasi nilai dari seluruh metrik evaluasi dapat diamati secara jelas.



Gambar 4.13 Grafik Perbandingan Uji Coba Skenario 1 sampai Skenario 7



Gambar 4.14 Grafik Perbandingan Uji Coba Skenario 8a sampai Skenario 9

Berdasarkan Gambar 4.13 dan Gambar 4.14, terlihat adanya perbedaan pola performa pada masing-masing kelompok skenario pengujian. Kelompok skenario SMOTE (S1–S3) dan *Pure* (S4–S6) cenderung menghasilkan nilai *Accuracy* yang lebih tinggi dibandingkan kelompok skenario yang menggunakan *Random Under-Sampling* (RUS). Sebaliknya, kelompok skenario RUS (S7–S9) menunjukkan nilai *Recall* yang jauh lebih tinggi dibandingkan kelompok lainnya.

Secara visual, grafik memperlihatkan adanya hubungan yang berlawanan antara *Accuracy* dan *Recall*. Pada skenario dengan nilai *Accuracy* yang tinggi, nilai *Recall* cenderung lebih rendah. Sebaliknya, ketika nilai *Recall* meningkat, nilai *Accuracy* mengalami penurunan. Pola ini menunjukkan adanya *trade-off* dalam proses klasifikasi, terutama pada dataset yang memiliki distribusi kelas tidak seimbang.

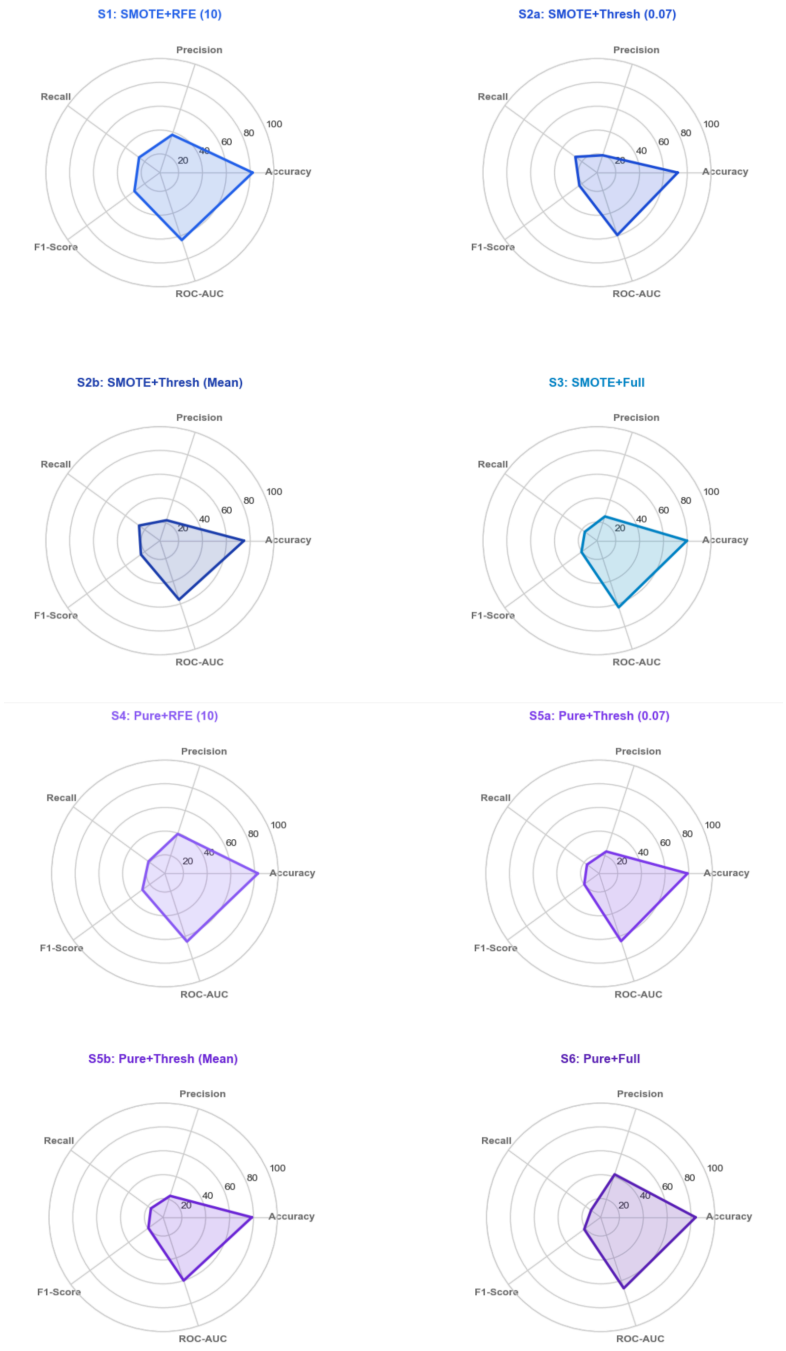
Selain itu, penerapan teknik *Random Under-Sampling* (RUS) menghasilkan peningkatan yang konsisten pada nilai *Recall*, *F1-Score*, dan ROC-AUC. Hal ini menunjukkan bahwa proses penyeimbangan data membantu model menjadi lebih sensitif dalam mendeteksi pasien yang berisiko penyakit jantung serta

meningkatkan kemampuan model dalam membedakan antara pasien yang berisiko dan tidak berisiko.

Berdasarkan pola yang ditunjukkan oleh grafik, Skenario 8a (RUS + *Threshold* 0,07) memiliki kombinasi performa yang paling seimbang dibandingkan skenario lainnya. Skenario ini tidak hanya menunjukkan kemampuan deteksi kasus positif yang tinggi, tetapi juga menghasilkan nilai *F1-Score* dan ROC-AUC yang lebih baik dibandingkan sebagian besar skenario lainnya. Oleh karena itu, Skenario 8a dipandang sebagai konfigurasi yang paling sesuai untuk mendukung proses deteksi dini risiko penyakit jantung.

Secara keseluruhan, visualisasi pada Gambar 4.13 memperjelas bahwa penggunaan metrik *Accuracy* saja tidak cukup untuk menilai kualitas model pada data yang tidak seimbang. Metrik *Recall*, *F1-Score*, dan ROC-AUC perlu dipertimbangkan secara bersama-sama agar diperoleh model yang mampu mendeteksi kasus risiko penyakit jantung secara lebih efektif.

Sebagai tahap akhir dalam evaluasi pemodelan, dilakukan analisis *Feature Importance* guna membedah variabel-variabel yang menjadi pilar utama dalam proses pengambilan keputusan algoritma XGBoost. Transparansi analitik ini krusial untuk mengetahui fitur klinis mana yang memberikan kontribusi paling dominan. Berdasarkan hasil ekstraksi pada model dengan performa optimal, atribut seperti *prevalentHyp*, *age*, dan *glucose* secara konsisten muncul sebagai prediktor penentu. Pemetaan tingkat signifikansi dari masing-masing fitur tersebut kemudian diilustrasikan secara lebih komprehensif pada Gambar 4.15 dan Gambar 4.16.



Gambar 4.15 Performa *Fingerprint* pada Skenario 1 sampai Skenario 6

Gambar 4.16 Performa *Fingerprint* pada Skenario 7 sampai Skenario 9

Berdasarkan visualisasi *Radar Chart* pada Gambar 4.15 dan Gambar 4.16, terlihat perbedaan karakteristik performa pada setiap skenario pengujian. Berbeda dengan grafik batang yang menampilkan perbandingan nilai masing-masing metrik, *Radar Chart* memberikan gambaran mengenai keseimbangan performa model secara keseluruhan.

Pada kelompok skenario *Pure* (S4–S6), bentuk radar cenderung tidak seimbang karena didominasi oleh nilai *Accuracy* yang tinggi, sementara nilai *Recall* dan *F1-Score* relatif rendah. Kondisi ini menunjukkan bahwa model mampu menghasilkan prediksi yang benar dalam jumlah besar secara keseluruhan, tetapi masih kurang efektif dalam mendeteksi pasien yang berisiko penyakit jantung.

Kelompok skenario SMOTE (S1–S3) menunjukkan bentuk radar yang sedikit lebih merata dibandingkan kelompok *Pure*. Penerapan teknik

oversampling membantu meningkatkan kemampuan model dalam mendeteksi kelas positif, meskipun peningkatan tersebut belum terlalu signifikan. Hal ini terlihat dari area radar yang mulai melebar pada sumbu *Recall* dan *F1-Score*.

Perubahan yang paling jelas terlihat pada kelompok skenario *Random Under-Sampling* (*RUS*) yaitu S7, S8a, S8b, dan S9. Bentuk radar pada kelompok ini memiliki area yang lebih luas dan lebih seimbang, terutama pada metrik *Recall*, *F1-Score*, dan ROC-AUC. Kondisi tersebut menunjukkan bahwa model memiliki kemampuan yang lebih baik dalam mendeteksi pasien yang berisiko penyakit jantung sekaligus mempertahankan kemampuan klasifikasi yang cukup baik secara keseluruhan.

Di antara seluruh skenario, Skenario 8a (*RUS* + *Threshold* 0,07) menunjukkan bentuk radar yang paling proporsional. Tidak terdapat dominasi yang berlebihan pada satu metrik tertentu, melainkan distribusi performa yang relatif seimbang pada berbagai metrik evaluasi. Hal ini menunjukkan bahwa model tidak hanya unggul pada kemampuan mendeteksi kasus positif, tetapi juga memiliki kemampuan yang baik dalam membedakan kedua kelas data.

Secara keseluruhan, visualisasi *Radar Chart* memperkuat hasil analisis sebelumnya bahwa skenario berbasis *Random Under-Sampling* menghasilkan performa yang lebih seimbang dibandingkan skenario *Pure* maupun SMOTE. Oleh karena itu, Skenario 8a dipandang sebagai konfigurasi yang paling sesuai untuk mendukung proses deteksi dini risiko penyakit jantung karena mampu memberikan keseimbangan terbaik di antara seluruh metrik evaluasi yang digunakan.

Berdasarkan hasil evaluasi yang telah disajikan dalam bentuk tabel, grafik batang, dan *radar chart*, dapat dilihat bahwa setiap visualisasi memberikan sudut pandang yang berbeda terhadap performa model. Tabel digunakan untuk menampilkan nilai masing-masing metrik secara rinci, grafik batang memudahkan perbandingan performa antar skenario, sedangkan *radar chart* memberikan gambaran mengenai keseimbangan performa model pada seluruh metrik evaluasi. Hasil dari ketiga bentuk visualisasi tersebut menunjukkan bahwa skenario berbasis *Random Under-Sampling* (RUS), khususnya Skenario 8a, menghasilkan performa yang paling seimbang dibandingkan skenario lainnya.

4.3.2 Perbandingan Hasil Skenario pada Tipe Balancing

Analisis pada subbab ini difokuskan pada evaluasi kinerja model berdasarkan berbagai metode *data balancing* yang digunakan untuk meningkatkan performa klasifikasi. Perbandingan performa antar skenario dengan pendekatan balancing tersebut dirangkum pada Tabel 4.44.

Tabel 4.44 Perbandingan Skenario pada Tipe *Balancing*

Tipe *Balancing*	**Rata-rata *Accuracy***	**Rata-rata *Precision***	**Rata-rata *Recall***	**Rata-rata *F1-Score***	**Rata-rata ROC-AUC**
SMOTE	77,19%	26,31%	23,88%	24,43%	60,61%
Non-SMOTE (Pure)	81,11%	32,09%	17,41%	22,11%	63,97%
Random Undersampling (RUS)	62,74%	25,21%	72,99%	37,47%	70,91%

Berdasarkan Tabel 4.44 memaparkan perbandingan performa model *Extreme Gradient Boosting* (XGBoost) dalam menangani isu ketidakseimbangan data kelas. Analisis ini menyoroti tiga skenario penanganan, yang akan dibahas

mulai dari penerapan *Synthetic Minority Over-sampling Technique* (SMOTE), penggunaan data asli (Non-SMOTE/*Pure*), hingga *Random Undersampling* (RUS).

Penerapan metode *oversampling* SMOTE pada model menunjukkan adanya sedikit perbaikan, di mana nilai *Recall* naik menjadi 23,88% dan *F1-Score* berada di angka 24,43%. Sayangnya, performa model secara keseluruhan tidak menunjukkan lonjakan yang berarti. Bahkan, jika dibandingkan dengan performa data aslinya, nilai rata-rata *Area Under Curve* (ROC-AUC) pada skenario SMOTE ini justru sedikit menyusut ke angka 60,61%.

Sebagai perbandingan, ketika model dijalankan murni menggunakan data asli tanpa penyeimbangan (Non-SMOTE), nilai rata-rata Akurasi memang terlihat paling tinggi, yakni mencapai 81,11% dengan Presisi 32,09%. Namun, tingginya akurasi ini sebenarnya mengecoh dan sering disebut sebagai *accuracy paradox*. Hal ini terbukti dari nilai *Recall* yang anjlok di angka 17,41%. Artinya, tanpa penanganan data yang seimbang, model hanya menebak aman dengan memprediksi mayoritas pasien berstatus sehat, tetapi gagal total saat harus mendeteksi pasien yang sebenarnya berisiko terkena penyakit.

Perubahan yang paling berdampak positif baru terlihat saat metode *Random Undersampling* (RUS) diterapkan. Walaupun secara angka rata-rata Akurasinya terpaksa turun menjadi 62,74%, pendekatan RUS sukses mendongkrak nilai rata-rata *Recall* secara drastis hingga menyentuh 72,99%. Untuk kasus krusial seperti deteksi serangan jantung, sensitivitas setinggi ini sangatlah vital demi meminimalkan *False Negative* (pasien sakit namun terdeteksi sehat). Tidak hanya itu, skenario RUS juga berhasil mencetak rata-rata *F1-Score* (37,47%) dan ROC-

AUC (70,91%) paling tinggi di antara semua metode. Temuan ini menegaskan bahwa RUS merupakan pendekatan yang paling pas dan optimal dalam menyeimbangkan kemampuan klasifikasi model XGBoost.

4.3.3 Perbandingan Hasil Skenario pada Tipe Seleksi Fitur

Analisis pada bagian ini difokuskan pada evaluasi kinerja model saat diterapkan berbagai metode seleksi fitur guna mencari tingkat optimalisasi terbaik. Rincian perbandingan performa antar skenario seleksi fitur ini dirangkum selengkapnya pada Tabel 4.45.

Tabel 4.45 Perbandingan Skenario pada Tipe Seleksi Fitur

Tipe Seleksi Fitur	**Rata-rata *Accuracy***	**Rata-rata *Precision***	**Rata-rata *Recall***	**Rata-rata *F1-Score***	**Rata-rata ROC-AUC**
RFE (10 Fitur)	76,09%	33,98%	38,99%	31,71%	66,42%
Threshold (0,07)	71,27%	22,99%	39,58%	26,87%	65,09%
Threshold (Mean)	71,86%	23,54%	39,29%	27,07%	62,41%
Full Feature (Semua Fitur)	75,50%	30,98%	34,52%	26,36%	66,74%

Berdasarkan Tabel 4.45, penerapan berbagai pendekatan seleksi fitur memberikan pengaruh terhadap performa model XGBoost dalam mengklasifikasikan risiko penyakit jantung. Hasil pengujian menunjukkan bahwa pendekatan RFE yang mempertahankan 10 fitur terbaik menghasilkan performa yang paling seimbang dibandingkan pendekatan seleksi fitur berbasis *threshold* maupun penggunaan seluruh fitur tanpa proses seleksi.

Pendekatan RFE dengan 10 fitur menghasilkan rata-rata *Accuracy* sebesar 76,09%, *Precision* sebesar 33,98%, *Recall* sebesar 38,99%, dan *F1-Score* sebesar 31,71%. Nilai *Accuracy*, *Precision*, serta *F1-Score* tersebut merupakan yang paling tinggi dibandingkan seluruh metode yang diuji. Hal ini mengindikasikan bahwa

proses eliminasi fitur secara bertahap berhasil mempertahankan fitur-fitur yang paling relevan terhadap variabel target, sehingga XGBoost dapat menangkap pola data dengan lebih optimal.

Sementara itu, pendekatan seleksi fitur berbasis *threshold* menghasilkan nilai *Recall* yang sedikit lebih tinggi. *Threshold* 0,07 memperoleh *Recall* sebesar 39,58%, sedangkan *Threshold Mean* memperoleh *Recall* sebesar 39,29%. Namun, peningkatan *Recall* tersebut diikuti oleh penurunan nilai *Accuracy*, *Precision*, dan *F1-Score*. Hal ini menunjukkan bahwa meskipun model mampu mendeteksi lebih banyak kasus positif, ketepatan prediksi yang dihasilkan menjadi lebih rendah dibandingkan pendekatan RFE.

Pada pendekatan *Full Feature*, seluruh fitur digunakan tanpa proses seleksi. Hasil pengujian menunjukkan bahwa pendekatan ini menghasilkan nilai ROC-AUC tertinggi sebesar 66,74%. Akan tetapi, nilai *Recall* sebesar 34,52% dan *F1-Score* sebesar 26,36% masih lebih rendah dibandingkan pendekatan RFE. Kondisi ini mengindikasikan bahwa penggunaan seluruh fitur tidak selalu menghasilkan performa klasifikasi yang lebih baik karena adanya kemungkinan fitur yang kurang relevan ikut memengaruhi proses pembelajaran model.

Secara keseluruhan, hasil pengujian menunjukkan bahwa kombinasi algoritma XGBoost dan pendekatan RFE dengan 10 fitur terbaik memberikan performa yang paling seimbang dalam mengklasifikasikan populasi yang berisiko serangan jantung. Proses seleksi fitur menggunakan RFE membantu model XGBoost dalam memusatkan pembelajaran pada fitur-fitur yang paling relevan, sehingga mampu meningkatkan nilai *Accuracy*, *Precision*, dan *F1-Score*

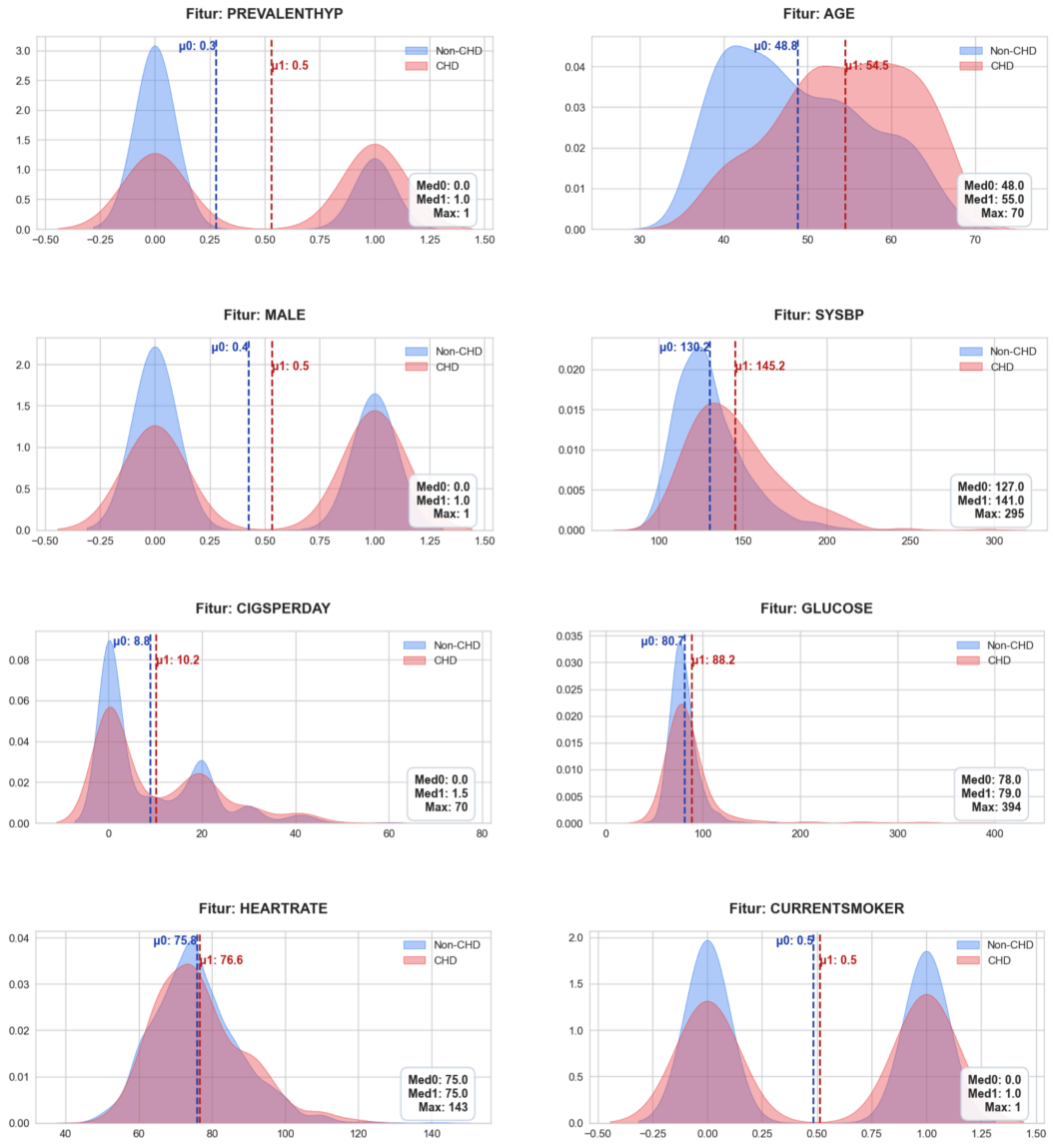
dibandingkan metode seleksi fitur lainnya. Oleh karena itu, kombinasi XGBoost dengan RFE dapat dikatakan efektif dalam mendukung klasifikasi risiko penyakit jantung pada penelitian ini.

Berdasarkan hasil pengujian yang telah dilakukan, kombinasi metode *Extreme Gradient Boosting* (XGBoost) dengan *Recursive Feature Elimination* (RFE) menunjukkan kinerja klasifikasi yang baik dalam mengidentifikasi individu yang memiliki risiko penyakit jantung. Pendekatan RFE yang mempertahankan 10 fitur terbaik menghasilkan rata-rata *Accuracy* sebesar 76,09%, Precision sebesar 33,98%, dan *F1-Score* sebesar 31,71%, yang merupakan nilai tertinggi dibandingkan pendekatan seleksi fitur lainnya. Hasil tersebut menunjukkan bahwa proses seleksi fitur menggunakan RFE mampu membantu XGBoost mempertahankan fitur-fitur yang paling relevan sehingga menghasilkan performa klasifikasi yang lebih konsisten dan seimbang. Selain meningkatkan efisiensi model melalui pengurangan jumlah fitur, penerapan RFE juga membantu mengurangi pengaruh fitur yang kurang relevan terhadap proses klasifikasi. Dengan demikian, kombinasi XGBoost dan RFE terbukti efektif dalam mendukung proses klasifikasi risiko penyakit jantung pada dataset yang digunakan dalam penelitian ini.

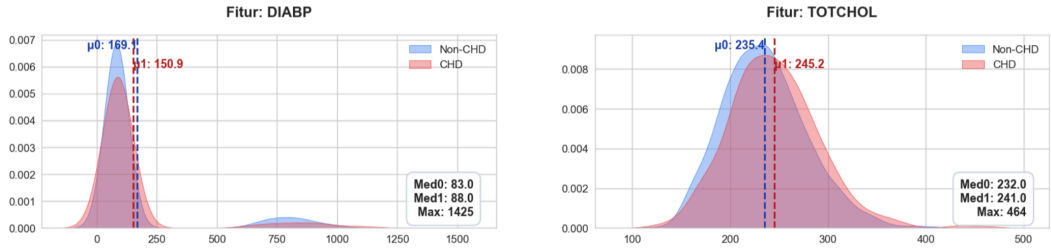
4.3.4 Analisis Overlap Data pada Kelas yang Tidak Seimbang

Berdasarkan hasil evaluasi, nilai *F1-Score* pada seluruh skenario masih berada pada kategori rendah. Hal ini menunjukkan bahwa model belum mampu secara optimal membedakan antara pasien yang memiliki risiko dan yang tidak memiliki risiko penyakit jantung. Dengan demikian, proses seleksi fitur yang diterapkan masih belum sepenuhnya berhasil menangkap kompleksitas pola yang

terdapat pada data. Kondisi tersebut menjadi dasar untuk melakukan analisis lebih lanjut terhadap distribusi fitur dalam dataset guna memahami karakteristik data yang memengaruhi performa model. Visualisasi distribusi fitur disajikan pada Gambar 4.17 dan Gambar 4.18.



Gambar 4.17 Pola Distribusi pada Fitur *Prevelenthyp, Age, Male, Sysbp, Cigsperday, Glucose, Heartrate, Currentsmoker*



Gambar 4.18 Pola Distribusi pada Fitur *Diabp, Totchol*

Gambar 4.17 dan Gambar 4.18 menunjukkan distribusi data pada 10 fitur yang digunakan dalam proses klasifikasi. Secara umum, terlihat bahwa distribusi kelas *Non-CHD* dan *CHD* masih banyak yang saling tumpang tindih (*feature overlap*). Hal ini menunjukkan bahwa perbedaan karakteristik antara pasien yang berisiko dan tidak berisiko penyakit jantung belum terlihat secara jelas pada sebagian besar fitur.

Beberapa fitur seperti *Age*, *SysBP*, *Glucose*, dan *TotChol* memiliki pola distribusi yang hampir sama antara kedua kelas. Meskipun terdapat perbedaan pada beberapa rentang nilai, sebagian besar area distribusinya masih saling beririsan. Kondisi ini menyebabkan model sulit membedakan data pasien yang berisiko dan yang tidak berisiko hanya berdasarkan satu fitur tertentu.

Tumpang tindih data tersebut menjadi salah satu penyebab nilai *Precision* dan *F1-Score* pada seluruh skenario masih relatif rendah. Ketika model berusaha meningkatkan kemampuan mendeteksi pasien yang berisiko melalui peningkatan nilai *Recall*, jumlah prediksi positif yang salah (*False Positive*) juga ikut meningkat. Akibatnya, nilai *Precision* menjadi lebih rendah dan berdampak pada nilai *F1-Score*.

Kondisi ini menunjukkan bahwa permasalahan utama pada dataset tidak hanya berasal dari ketidakseimbangan jumlah data antar kelas, tetapi juga karena karakteristik data pasien sehat dan pasien yang berisiko penyakit jantung masih memiliki banyak kemiripan. Oleh karena itu, meskipun teknik penyeimbangan data mampu meningkatkan nilai *Recall*, peningkatan performa model secara keseluruhan tetap dibatasi oleh tingginya tingkat *feature overlap* pada data.

4.4 Integrasi Islam

Penelitian mengenai deteksi dini penyakit jantung menggunakan algoritma *XGBoost* ini tidak hanya merupakan upaya pengembangan teknologi, tetapi juga manifestasi nyata dari nilai-nilai keislaman yang tertuang dalam tiga dimensi hubungan utama: *hablum minallah* (hubungan dengan Allah), *hablum minannas* (hubungan dengan sesama manusia), dan *hablum minal alam* (hubungan dengan alam)

4.4.1 Dimensi Hubungan dengan Allah (*Hablum Minallah*)

Dalam perspektif Islam, seluruh ilmu pengetahuan bersumber dari Allah SWT dan dianugerahkan kepada manusia agar dimanfaatkan untuk kemaslahatan. Hal ini ditegaskan dalam wahyu pertama yang diturunkan kepada Nabi Muhammad SAW, yakni Surah Al-'Alaq ayat 1 hingga 5:

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿١﴾ خَلَقَ الْإِنسَانَ مِنْ عَلَقٍ ﴿٢﴾ اقْرَأْ وَرَبُّكَ الْأَكْرَمُ ﴿٣﴾ الَّذِي عَلَّمَ بِالْقَلَمِ ﴿٤﴾ عَلَّمَ الْإِنسَانَ مَا لَمْ يَعْلَمْ﴾

"Bacalah dengan (menyebut) nama Tuhanmu yang menciptakan. Dia telah menciptakan manusia dari segumpal darah. Bacalah, dan Tuhanmulah yang Maha Pemurah. Yang mengajar (manusia) dengan perantaraan kalam. Dia mengajarkan kepada manusia apa yang tidak diketahuinya." (Q.S. Al-'Alaq [96]: 1-5).

Ismail bin Katsir dalam kitabnya, Tafsir Al-Qur'an Al-'Azhim (Juz 30), menjelaskan bahwa ayat ini merupakan nikmat pertama yang Allah berikan kepada hamba-Nya. Beliau memaparkan bahwa kemuliaan manusia terletak pada ilmu yang diajarkan Allah, di mana penguasaan ilmu tersebut mencakup tiga aspek perantara, yaitu: hati (pemahaman), lisan (pengucapan), dan tulisan atau pena (al-qalam). Beliau menekankan bahwa eksistensi pena merupakan instrumen penting untuk mengikat dan menyebarkan ilmu pengetahuan yang luas (Ibnu Katsir, Tafsir Al-Qur'an Al-'Azhim, Jilid 8, hal. 437).

Dengan demikian, pengembangan sistem deteksi penyakit jantung menggunakan algoritma XGBoost dalam penelitian ini merupakan manifestasi modern dari perintah membaca (Iqra) dan penggunaan pena (Al-Qalam). Algoritma XGBoost bekerja dengan cara membaca pola-pola rumit dan mengekstraksi informasi dari data kesehatan yang sebelumnya tidak terjangkau oleh logika manusia secara manual. Aktivitas ilmiah ini bukan sekadar urusan duniawi, melainkan bentuk ketundukan kepada Allah SWT dengan mendayagunakan akal untuk menjaga jiwa (*Hifdzun Nafs*), sebagaimana hakikat ilmu adalah untuk menyingkap kebenaran demi kemaslahatan umat.

Islam memandang kesehatan jiwa manusia sebagai salah satu tujuan utama syariat yang dikenal dengan istilah maqashid syariah, khususnya pada aspek *hifz al-nafs* (menjaga jiwa). Upaya mendeteksi dini penyakit jantung sejalan langsung dengan misi menjaga jiwa ini. Allah SWT berfirman dalam Surah Al-Ma'idah ayat 32:

مِنْ أَجْلِ ذَٰلِكَ كَتَبْنَا عَلَىٰ بَنِي إِسْرَائِيلَ أَنَّهُ مَن قَتَلَ نَفْسًا بِغَيْرِ نَفْسٍ أَوْ فَسَادٍ فِي الْأَرْضِ فَكَأَنَّمَا قَتَلَ النَّاسَ جَمِيعًا ۚ وَمَنْ أَحْيَاهَا فَكَأَنَّمَا أَحْيَا النَّاسَ جَمِيعًا

"Oleh karena itu, Kami menetapkan (suatu hukum) bagi Bani Israil bahwa siapa yang membunuh seseorang bukan karena (orang yang dibunuh itu) telah membunuh orang lain atau karena telah berbuat kerusakan di bumi, maka seakan-akan dia telah membunuh semua manusia. Sebaliknya, siapa yang memelihara kehidupan seorang manusia, dia seakan-akan telah memelihara kehidupan semua manusia." (Q.S. Al-Ma'idah [5]: 32).

Ismail bin Katsir dalam Tafsir Al-Qur'an Al-Azhim (Juz 6) menjelaskan bahwa barang siapa memelihara kehidupan seorang manusia, maka seolah-olah dia telah memelihara kehidupan manusia semuanya, karena menurut Allah tidak ada bedanya antara satu jiwa dengan jiwa yang lainnya dalam hal kemuliaan hak hidup. Syaikh Wahbah Az-Zuhaili dalam Tafsir Al-Munir (Jilid 3) menambahkan bahwa memelihara kehidupan seseorang mencakup segala upaya yang menciptakan keamanan dan menghilangkan ketakutan serta kekhawatiran terhadap hilangnya nyawa. Sementara itu, Buya Hamka dalam Tafsir Al-Azhar (Jilid 2) menegaskan bahwa prinsip menjaga kehidupan ini bersifat universal bagi seluruh umat manusia dan tidak terbatas hanya pada kaum tertentu.

Sistem klasifikasi penyakit jantung yang dihasilkan dalam penelitian ini berpotensi menjadi sarana deteksi risiko lebih dini sehingga dapat menjadi jalan bagi terselamatkannya nyawa seseorang melalui tindakan medis yang tepat waktu. Dengan demikian, secara nilai keislaman, penelitian ini merupakan implementasi nyata dari semangat memelihara kehidupan yang bernilai sangat tinggi di sisi Allah SWT.

4.4.2 Hubungan dengan Sesama Manusia (*Hablum Minannas*)

Islam sangat menekankan pentingnya dimensi sosial dalam setiap amal perbuatan. Penelitian ini tidak hanya menjadi karya ilmiah, tetapi juga wujud nyata dari prinsip ta'awun (tolong-menolong) yang diperintahkan Allah SWT. Allah SWT berfirman dalam Surah Al-Ma'idah ayat 2:

وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ ۖ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ ۚ وَاتَّقُوا اللَّهَ ۖ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ

"Tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan permusuhan. Bertakwalah kepada Allah, sesungguhnya Allah sangat berat siksaan-Nya." (Q.S. Al-Ma'idah [5]: 2).

Ismail bin Katsir dalam kitab tafsirnya menjelaskan bahwa Allah memerintahkan hamba-Nya yang beriman untuk senantiasa tolong-menolong dalam perbuatan baik yang disebut kebajikan (birru) serta meninggalkan perbuatan mungkar (Ibnu Katsir, Tafsir Al-Qur'an Al-Azhim, Juz 6). Ibnu Asyur dalam kitab At-Tahrir wa At-Tanwir menegaskan bahwa dengan bekerja sama dalam kebajikan, suatu pekerjaan menjadi lebih ringan, maslahatnya menjadi lebih luas, serta dapat memperkuat rasa persatuan di tengah masyarakat (Ibnu Asyur, At-Tahrir wa At-Tanwir, Juz 6). Lebih lanjut, Al-Qurtubi menukil pendapat Al-Mawardi dalam kitab Al-Jami' Li Ahkam Al-Qur'an bahwa dengan bertakwa seseorang mendapatkan ridha Allah, dan dengan tolong-menolong dalam kebaikan seseorang akan mendapatkan ridha dari sesama manusia (Al-Qurtubi, Al-Jami' Li Ahkam Al-Qur'an, Juz 6).

Pengembangan sistem deteksi penyakit jantung berbasis XGBoost dalam penelitian ini merupakan bentuk konkret dari ta'awun dalam bidang kesehatan.

Sistem ini dirancang untuk membantu tenaga medis maupun masyarakat umum dalam mengenali risiko penyakit jantung secara lebih awal, yakni sebuah pertolongan nyata kepada sesama manusia yang mungkin belum menyadari kondisi kesehatannya.

Dimensi sosial ini diperkuat oleh sabda Rasulullah SAW dalam hadits yang diriwayatkan dari Abu Hurairah radhiyallahu 'anhu:

مَنْ نَفَّسَ عَنْ مُؤْمِنٍ كُرْبَةً مِنْ كُرَبِ الدُّنْيَا، نَفَّسَ اللهُ عَنْهُ كُرْبَةً مِنْ كُرَبِ يَوْمِ الْقِيَامَةِ

"Barangsiapa yang menghilangkan satu kesusahan dunia dari seorang mukmin, maka Allah akan menghilangkan darinya satu kesusahan di hari kiamat. Barangsiapa yang menolong saudaranya, maka Allah akan menolongnya sebagaimana ia menolong saudaranya." (H.R. Muslim).

Hadits tersebut menegaskan bahwa meringankan beban orang lain bukan sekadar kebaikan sosial semata, melainkan ibadah yang bernilai pahala besar di sisi Allah SWT. Sistem deteksi dini penyakit jantung yang dihasilkan dalam penelitian ini berpotensi menjadi salah satu sarana untuk menghilangkan kesusahan bagi orang yang terancam penyakit berbahaya namun belum terdiagnosis secara medis.

4.4.3 Hubungan dengan Alam (*Hablum Minal Alam*)

Islam menegaskan bahwa manusia diamanahkan sebagai *khalifah* di muka bumi. Oleh karena itu, setiap aktivitas ilmiah dan pengembangan teknologi harus senantiasa memperhatikan prinsip kemaslahatan serta menghindari segala bentuk kerusakan terhadap lingkungan. Allah SWT berfirman dalam Surah Al-A’raf ayat 56:

وَلَا تُفْسِدُوا فِي الْأَرْضِ بَعْدَ إِصْلَاحِهَا وَادْعُوهُ خَوْفًا وَطَمَعًا ۚ إِنَّ رَحْمَتَ اللَّهِ قَرِيبٌ مِّنَ الْمُحْسِنِينَ

"Dan janganlah kamu berbuat kerusakan di bumi setelah (diciptakan) dengan baik. Berdoalah kepada-Nya dengan rasa takut dan penuh harap. Sesungguhnya rahmat Allah sangat dekat kepada orang-orang yang berbuat kebaikan."
(Q.S. Al-A'raf [7]: 56).

Penelitian mengenai deteksi penyakit jantung berbasis algoritma XGBoost ini selaras dengan prinsip tersebut. Metode yang digunakan berupa komputasi berbasis data (*machine learning*), yang tidak melibatkan penggunaan bahan kimia berbahaya, tidak menghasilkan limbah beracun, serta tidak memasukkan zat apa pun yang berpotensi membahayakan tubuh manusia maupun lingkungan. Berbeda dengan prosedur diagnostik konvensional yang dalam beberapa kasus memerlukan zat kontras, bahan radioaktif, atau tindakan invasif, sistem klasifikasi ini hanya memproses data klinis yang telah tersedia dalam bentuk digital. Dengan demikian, pendekatan ini bersifat aman, bersih, dan ramah lingkungan. Prinsip ini juga sejalan dengan kaidah fikih yang fundamental:

لَا ضَرَرَ وَلَا ضِرَارَ

"Tidak boleh ada bahaya dan tidak boleh membahayakan orang lain." *(H.R. Ibnu Majah, dari Ubadah bin Ash-Shamit).*

Imam An-Nawawi dalam *Al-Majmu'* menjelaskan bahwa kaidah ini merupakan salah satu fondasi utama syariat Islam dalam mengatur hubungan manusia dengan sesama dan lingkungan. Segala sesuatu yang berpotensi menimbulkan mudarat wajib dihindari, baik terhadap individu maupun alam. Sistem berbasis *machine learning* seperti XGBoost memenuhi kaidah ini, karena prosesnya tidak menimbulkan dampak negatif terhadap tubuh manusia maupun

kelestarian lingkungan. Lebih lanjut, Allah SWT juga berfirman dalam Surah Al-Baqarah ayat 195:

وَأَنفِقُوا فِي سَبِيلِ اللَّهِ وَلَا تُلْقُوا بِأَيْدِيكُمْ إِلَى التَّهْلُكَةِ

"Dan infakkanlah (hartamu) di jalan Allah, dan janganlah kamu jatuhkan (diri sendiri) ke dalam kebinasaan." (Q.S. Al-Baqarah [2]: 195).

Para ulama tafsir, di antaranya Imam Al-Qurtubi dalam *Al-Jami' li Ahkam Al-Qur'an*, menafsirkan ayat ini sebagai larangan melakukan segala tindakan yang dapat mendatangkan kebinasaan, baik terhadap diri sendiri maupun lingkungan. Pengembangan sistem deteksi penyakit jantung ini merupakan implementasi dari semangat ayat tersebut, yaitu memilih ikhtiar ilmiah yang aman, tidak membahayakan pasien, serta tidak menimbulkan dampak destruktif terhadap alam.

4.4.4 Sintesis Integrasi Keislaman

Berdasarkan uraian di atas, penelitian ini memiliki tiga dimensi nilai keislaman yang saling menguatkan satu sama lain. Pertama, dari sisi *hablum minallah*, penelitian ini merupakan wujud syukur dan ketundukan kepada Allah SWT melalui pemanfaatan akal yang telah dikaruniakan-Nya, serta menjadi bagian dari upaya menjaga jiwa (*hifz al-nafs*) sebagai salah satu maqashid syariah. Kedua, dari sisi *hablum minannas*, penelitian ini merupakan realisasi nyata dari perintah tolong-menolong dalam kebajikan (*ta'awun alal birr*) sebagaimana diperintahkan Allah SWT dalam Surah Al-Ma'idah ayat 2, yakni dengan membantu sesama manusia melalui sistem deteksi dini penyakit jantung. Ketiga, dari sisi *hablum minal alam*, penelitian ini merupakan bentuk eksplorasi terhadap tanda-tanda

kekuasaan Allah yang tersimpan dalam data tubuh manusia, selaras dengan semangat ikhtiar ilmiah yang diperintahkan Islam.

Dengan demikian, penelitian tentang deteksi penyakit jantung menggunakan algoritma XGBoost ini tidak hanya memberikan kontribusi pada pengembangan ilmu pengetahuan dan teknologi, tetapi juga merupakan amal kebajikan yang berakar pada nilai-nilai Islam: menjaga jiwa, menolong sesama, dan mengagungkan ilmu sebagai jalan mendekatkan diri kepada Allah SWT.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan serangkaian eksperimen yang telah dilakukan pada penelitian prediksi risiko penyakit serangan jantung menggunakan algoritma XGBoost, dapat disimpulkan bahwa kualitas data awal memegang peranan krusial terhadap integritas model. Temuan menunjukkan adanya 13,73% data yang mengandung nilai kosong serta ketimpangan kelas yang signifikan. Melalui tahap pra-pemrosesan, implementasi *Recursive Feature Elimination* (RFE) terbukti efektif dalam mereduksi dimensi data dari 15 menjadi 10 fitur inti. Fitur medis seperti riwayat hipertensi (*prevalentHyp*), faktor usia (*age*), dan kadar glukosa darah (*glucose*) teridentifikasi sebagai indikator paling dominan yang secara signifikan memengaruhi kemampuan XGBoost dalam mendeteksi risiko serangan jantung.

Secara performa, penelitian ini mengungkapkan bahwa akurasi tertinggi yang mencapai 83,88% pada Skenario S6 (*Pure+Full*) bersifat semu (*accuracy paradox*), karena model gagal mengenali kelas minoritas yang ditunjukkan dengan nilai *Recall* sangat rendah sebesar 14,29%. Penerapan teknik penyeimbangan data terbukti menjadi solusi krusial, di mana *Random Undersampling* (RUS) memberikan dampak yang jauh lebih positif dibandingkan SMOTE dalam meningkatkan sensitivitas model. Berdasarkan hasil evaluasi, Skenario S8a (RUS + *Threshold* 0.07) ditetapkan sebagai model terbaik dengan capaian *Recall* tertinggi

sebesar 75,00% dan ROC-AUC 71,58%. Pemilihan ini didasarkan pada prioritas medis untuk meminimalkan risiko *false negative* demi menjaga keselamatan pasien, meskipun harus mengorbankan tingkat akurasi dan presisi secara keseluruhan.

Terakhir, penelitian ini berhasil mengidentifikasi bahwa hambatan utama dalam mencapai nilai *F1-Score* yang tinggi (berada di bawah 40% pada seluruh skenario) bukan terletak pada algoritma XGBoost, melainkan pada karakteristik dataset itu sendiri. Berdasarkan analisis pola distribusi, ditemukan fenomena tumpang tindih fitur (*feature overlap*) yang sangat masif antara pasien sehat dan pasien berisiko pada variabel klinis utama. Ambiguitas data ini menyebabkan garis batas keputusan (*decision boundary*) sulit terbentuk secara optimal, sehingga setiap upaya untuk meningkatkan sensitivitas model akan secara otomatis menaikkan angka *false positive*. Hal ini menegaskan bahwa optimasi parameter, seleksi fitur, dan teknik *resampling* telah bekerja maksimal dalam batasan kompleksitas data klinis yang tersedia.

5.2 Saran

Berdasarkan hasil penelitian yang telah diperoleh, peneliti menyadari bahwa masih terdapat ruang untuk pengembangan demi mencapai kesempurnaan hasil prediksi. Pengembangan ini bukan sekadar upaya teknis untuk meningkatkan angka statistik, melainkan sebuah tanggung jawab moral untuk memberikan hasil yang paling akurat bagi kemaslahatan umat dalam menjaga kesehatan jantung. Oleh karena itu, peneliti memberikan beberapa saran yang disusun berdasarkan prinsip penyempurnaan berkelanjutan dalam metodologi penelitian sebagai berikut:

1. Penelitian selanjutnya disarankan untuk menerapkan teknik imputasi data yang lebih lanjut, seperti *K-Nearest Neighbors* (KNN) *Imputer* atau *Iterative Imputer*, dalam menangani *missing values*. Hal ini bertujuan agar informasi pada setiap responden dapat dipertahankan secara utuh tanpa harus mengeliminasi data yang berpotensi memiliki pola klinis penting.
2. Mengingat adanya kendala *feature overlap* yang masif pada dataset ini, penelitian berikutnya dapat mengeksplorasi penggunaan metode *hybrid resampling* seperti SMOTE-*Tomek* atau SMOTE-ENN. Teknik ini diharapkan mampu mempertegas batas keputusan (*decision boundary*) antar kelas dengan cara menghapus sampel yang dianggap sebagai *noise* setelah proses *oversampling* dilakukan.
3. Disarankan untuk memperluas cakupan variabel klinis dengan menambahkan fitur-fitur medis terbaru yang relevan dengan perkembangan ilmu kardiologi, seperti kadar LDL/HDL yang lebih spesifik, riwayat genetik, atau variabel gaya hidup yang lebih mendetail. Penambahan dimensi fitur ini diharapkan dapat meningkatkan daya pisah (*separability*) antar-kelas.
4. Disarankan Penelitian selanjutnya juga dapat membandingkan performa XGBoost dengan algoritma lain seperti LightGBM atau metode *ensemble* lainnya untuk mengetahui model yang paling efektif dalam memprediksi penyakit jantung.

DAFTAR PUSTAKA

Al Aziz, H., & Santoso, H. A. (2025). Model Prediksi Stunting Anak di Indonesia Menggunakan Extreme Gradient Boosting. *Jurnal Algoritma*, *22*(1), 1072–1085. https://doi.org/10.33364/algoritma/v.22-1.2289

Aniamarta, T., Salsabilla Huda, A., & Lizariani Aqsha, F. (2022). Penyebab dan Pengobatan Serangan Jantung. *Biologica Samudra*, *4*(1), 22–31. https://doi.org/10.33059/jbs.v4i1.3925

Anita, M., Yulianti, I. G. D., & Pasaribu, S. V. (2025). Klasifikasi Faktor Risiko Penyakit Jantung Menggunakan Machine Learning. *Jurnal Teknologi Informasi*, *16*(1), 68–78. https://doi.org/10.52972/hoaq.vol16no1

Azhari, M., Situmorang, Z., & Rosnelly, R. (2021). Perbandingan Akurasi, Recall, dan Presisi Klasifikasi pada Algoritma C4.5, Random Forest, SVM dan Naive Bayes. *Jurnal Media Informatika Budidarma*, *5*(2), 640–651. https://doi.org/10.30865/mib.v5i2.2937

Barata, M. A., Noersasongko, E., Purwanto, & Soeleman, M. A. (2023). Improving the Accuracy of C4.5 Algorithm with Chi-Square Method on Pure Tea Classification Using Electronic Nose. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *7*(2), 226–235. https://doi.org/10.29207/resti.v7i2.4687

Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. https://doi.org/10.1145/2939672.2939785

Darmawan, Z. M. E., & Dianta, A. F. (2023). Implementasi Optimasi Hyperparameter GridSearchCV Pada Sistem Prediksi Serangan Jantung Menggunakan SVM. *Teknologi: Jurnal Ilmiah Sistem Informasi*, *13*(1), 8–15. https://doi.org/10.26594/teknologi.v13i1.3098

Dhany, H. W., Permana, A. I., Izhari, F., Ginting, A. P., & Pratama, Z. G. (2024). Aplikasi Prediksi Serangan Jantung Untuk Warga Kelurahan Pelawi Utara. *Jurnal Minfo Polgan*, *13*(2), 2096–2101. https://doi.org/10.33395/jmp.v13i2.14396

Dullah, A. U., Darmawan, A. Y., Pertiwi, D. A. A., & Unjung, J. (2025). Extreme Gradient Boosting Model with SMOTE for Heart Disease Classification. *Jurnal Informatika Sunan Kalijaga)*, *10*(1), 48–62. https://doi.org/10.14421/jiska.2025.10.1.48-62

Fathima, M. D., Samuel, S. J., & Raja, S. P. (2023). HDDSS: An Enhanced Heart Disease Decision Support System Using RFE-ABGNB Algorithm. *International Journal of Interactive Multimedia and Artificial Intelligence*, *8*(2), 29–37. https://doi.org/10.9781/ijimai.2021.10.003

Hakim, L., Zyen, A. K., & Sarwido. (2025). Optimasi Model Klasifikasi Diabetes dengan Stacking pada Algoritma XGBoost dan LightGBM. *JUPITER*, *17*, 821–834.

Herza, F. M., Rahmat, B., & Haromainy, M. M. AL. (2024). Pengaruh RFE terhadap Logistic Regression dan Support Vector Machine pada Analisis

Sentimen Hotel Shangri-La Surabaya. *Jurnal Mahasiswa Teknik Informatika*, *8*(6). https://doi.org/https://doi.org/10.36040/jati.v8i6.11272

Jinbo, Z., Yufu, L., & Haitao, M. (2025). Handling Missing Data Using XGBoost-Based Multiple Imputation by Chained Equations Regression Method. *Frontiers in Artificial Intelligence*, *8*. https://doi.org/10.3389/frai.2025.1553220

Khalisatifa, A., Arum, H. Dela, & Jambak, M. I. (2024). Klasifikasi Risiko Penyakit Serangan Jantung dengan Menggunakan Algoritma C4.5. *Jurnal Device*, *14*(1), 57–64.

Laili, E. F., Alawi, Z., Rohmah, R., & Barata, M. A. (2025a). Komparasi Algoritma Decision Tree dan Support Vector Machine (SVM) dalam Klasifikasi Serangan Jantung. *Jurnal Sistem Informasi Dan Informatika (Simika)*, *8*(1), 67–76. https://doi.org/10.47080/simika.v8i1.3683

Laili, E. F., Alawi, Z., Rohmah, R., & Barata, M. A. (2025b). Komparasi Algoritma Decision Tree dan Support Vector Machine (SVM) dalam Klasifikasi Serangan Jantung. *Jurnal Sistem Informasi Dan Informatika (Simika)*, *8*(1), 67–76. https://doi.org/10.47080/simika.v8i1.3683

Liu, J., Wu, J., Liu, S., Li, M., Hu, K., & Li, K. (2021). Predicting mortality of patients with acute kidney injury in the ICU using XGBoost model. *PLOS ONE*, *16*(2), 1–1. https://doi.org/10.1371/journal.pone.0246306

Mayang Pratiwi, D., & Mufidah, L. (2024). *Perbandingan Metode Decision Tree Classifier dan XGBoost Classifier Dalam Memprediksi Penyakit Jantung* (Vol. 4, Number 1).

Nafisah, S., Inayah, N. N., & Yusuf, B. (2024). Literatur Review: Penyebab dan Perkembangan Penyakit Jantung Koroner. *Jurnal Forum Kesehatan : Media Publikasi Kesehatan Ilmiah*, *14*(1), 27–36. https://doi.org/10.52263/jfk.v14i1.254

Pagrut, S. B., Ozair, M. A., Ingle, S., Dharangaonkar, R., & Mundhada, A. (2023). Heartcare: Heart Disease Detection using Machine Learning. *International Journal of Advanced Research in Science, Communication and Technology*, *3*(5), 249–254. https://doi.org/10.48175/ijarsct-9352

Pradana, M. G., Wijaya, D. P., & Saputro, P. H. (2022). Komparasi Metode Support Vector Machine dan Naïve Bayes dalam Klasifikasi Peluang Penyakit Serangan Jantung. *Indonesian Journal of Business Intelligence (IJUBI)*, *5*(2), 87–91. https://doi.org/10.21927/ijubi.v5i2.2659

Pratama, A. R. I., Latipah, S. A., & Sari, B. N. (2022). Optimasi Klasifikasi Curah Hujan Menggunakan Support Vector Machine (SVM) dan Recursive Feature Elimination (RFE). *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *7*(2), 314–324. https://doi.org/10.29100/jipi.v7i2.2675

Priyatno, A. M., & Widiyaningtyas, T. (2024). A Systematic Literature Review: Recursive Feature Elimination Algorithms. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *9*(2), 196–207. https://doi.org/10.33480/jitk.v9i2.5015

Purnomo, A., Barata, M. A., Soeleman, M. A., & Alzami, F. (2020). Adding feature selection on Naïve Bayes to increase accuracy on classification heart attack

disease. *Journal of Physics: Conference Series*, *1511*, 1–6. https://doi.org/10.1088/1742-6596/1511/1/012001

Putra, L. G. R., Prasetya, D. D., & Mayadi. (2025). Student Dropout Prediction Using Random Forest and XGBoost Method. *INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, *9*(1), 147–157. https://doi.org/10.29407/intensif.v9i1.21191

Rahayu, D. S., Nursafika, Afifah, J., & Intan, S. (2023). Classification of Diabetes Mellitus Using C4.5 Algorithm, Support Vector Machine (SVM) and Linear Regression. *SENTIMAS: Seminar Nasional Penelitian Dan Pengabdian Masyarakat*, 56–63. https://journal.irpi.or.id/index.php/sentimas

Sah, A., Niesa, C., Rumandan, R. J., & Muharrom, M. (2025). Analisis Model Prediksi Penyakit Jantung Menggunakan Adaptive Boosting, Gradient Boosting, dan Extreme Gradient Boosting. *Jurnal Ilmiah FIFO*, *17*(1), 46–56. https://doi.org/10.22441/fifo.2025.v17i1.006

Sari, R. P., & Ikbal, R. N. (2024). Edukasi Pertolongan Pertama Serangan Jantung. *JPIK (Jurnal Pengabdian Ilmu Kesehatan)*, *3*(2), 119–112. https://doi.org/10.33757/jpik.v3i2.66

Sausan, Pratiwi, D. M., & Mufidah, L. (2024). Perbandingan Metode Decision Tree Classifier dan XGBoost Classifier Dalam Memprediksi Penyakit Jantung. *CENTIVE*, *4*(1), 991–1000. https://doi.org/10.20895/centive.v4i1.336

Schober, P., & Schwarte, L. A. (2018). Correlation coefficients: Appropriate use and interpretation. *Anesthesia and Analgesia*, *126*(5), 1763–1768. https://doi.org/10.1213/ANE.0000000000002864

Shrivastava, A., Kotiyal, A., Habelalmateen, M. I., Rana, A., Devi, V. S. A., Rao, B. D., & Bansal, S. (2024). Leveraging XGBoost for Predictive Analytics in Healthcare: Enhancing Disease Diagnosis. *2024 7th International Conference on Contemporary Computing and Informatics (IC3I)*, 1666–1672. https://doi.org/10.1109/IC3I61595.2024.10829136

Syukron, M., Santoso, R., & Widiharih, T. (2020). Perbandingan Metode SMOTE Random Forest dan SMOTE XGBoost untuk Klasifikasi Tingkat Penyakit Hepatitis C pada Imbalanced Class Data. *GAUSSIAN*, *9*, 227–236. https://doi.org/10.14710/j.gauss.9.3.227-236

World Health Organization. (2025). *Cardiovascular diseases (CVDs)*. https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

Yang, J., & Guan, J. (2022). A Heart Disease Prediction Model Based on Feature Optimization and Smote-Xgboost Algorithm. *Information (Switzerland)*, *13*. https://doi.org/10.3390/info13100475

Zaiem Mustaqiem. (2021). Heart Attack Prediction. *Kaggle*. https://www.kaggle.com/code/zaiemmustaqiem/heart-attack/input

Zhang, B., Dong, X., Hu, Y., Jiang, X., & Li, G. (2023). Classification and prediction of spinal disease based on the SMOTE-RFEXGBoost model. *PeerJ Computer Science*, *9*, 17–20. https://doi.org/10.7717/PEERJ-CS.1280