

Penggunaan Algoritma Stacking Classifier Pada Sistem Deteksi Risiko Kardiovaskular

Imam Bari Setiawan, Maulida Ayu Fitriani*, Elindra Ambar Pambudi, Muhammad Hamka

Fakultas Teknik dan Sains, Program Studi Teknik Informatika, Universitas Muhammadiyah Purwokerto, Banyumas, Indonesia
Email: ¹barisetiawan51@gmail.com, ^{2*}maulidaayuf@ump.ac.id, ³elindraambarpambudi@ump.ac.id, ⁴muhhammadhamka@ump.ac.id

Email Penulis Korespondensi: maulidaayuf@ump.ac.id

Submitted 04-12-2025; Accepted 31-12-2025; Published 31-12-2025

Abstrak

Penyakit kardiovaskular merupakan penyebab utama kematian global. Namun, kompleksitas data medis sering membuat model konvensional gagal menangkap pola tersembunyi sehingga performa prediksi menjadi tidak optimal. Penelitian ini mengevaluasi efektivitas model hibrida yang mengintegrasikan K-Modes Clustering dengan algoritma Stacking Classifier dan menguji apakah kompleksitas model tersebut mampu memberikan peningkatan kinerja yang baik dibandingkan model tunggal. Metodologi melibatkan pra-pemrosesan data termasuk penanganan outlier, rekayasa fitur klinis, dan ekstraksi fitur klaster menggunakan K-Modes (K=2). Arsitektur Stacking Classifier dibangun menggunakan lima base-learner heterogen yang telah dioptimalkan (CatBoost, Decision Tree, MLP, SVC, Logistic Regression) dan XGBoost sebagai meta-learner, diperlakukan melalui Stratified 5-Fold Cross-Validation. Hasil penelitian menunjukkan bahwa meskipun K-Modes efektif memetakan kategori risiko valid secara klinis, model Stacking Classifier (akurasi 87,99% dan ROC-AUC 95,89%), terbukti belum mampu melampaui performa model tunggal terbaik, yaitu CatBoost (akurasi 88,03% dan ROC-AUC 95,90%). Temuan paling signifikan terletak pada efisiensi waktu komputasi, di mana algoritma Stacking Classifier memerlukan waktu komputasi 560 kali lipat lebih lama (7587,7686 detik) dibandingkan CatBoost (13,4635 detik) tanpa memberikan peningkatan performa yang sepadan. Hal ini menunjukkan bahwa algoritma berbasis Boosting mampu menangkap pola kompleks tanpa memerlukan lapisan ensemble tambahan, sehingga model tunggal yang dioptimalkan lebih direkomendasikan untuk implementasi nyata dengan memberikan keseimbangan terbaik antara keakuratan prediksi dan efisiensi waktu komputasi.

Kata Kunci: Penyakit Kardiovaskular; Stacking Classifier; K-Modes Clustering; Stratified Cross-Validation; Efisiensi Komputasi

Abstract

Cardiovascular disease is a leading cause of global death. However, the complexity of medical data often makes conventional models fail to capture hidden patterns, resulting in suboptimal predictive performance. This study evaluates the effectiveness of a hybrid model that integrates K-Modes Clustering with the Stacking Classifier algorithm and tests whether the model's complexity can provide significant performance improvements compared to a single model. The methodology involves data preprocessing including outlier handling, clinical feature engineering, and cluster feature extraction using K-Modes (K=2). The Stacking Classifier architecture is built using five optimized heterogeneous base-learners (CatBoost, Decision Tree, MLP, SVC, Logistic Regression) and XGBoost as a meta-learner, validated through Stratified 5-Fold Cross-Validation. The results showed that although K-Modes effectively mapped clinically valid risk categories, the Stacking Classifier model (87.99% accuracy and 95.89% ROC-AUC) was not able to surpass the performance of the best single model, namely CatBoost (88.03% accuracy and 95.90% ROC-AUC). The most significant finding lies in the computational time efficiency, where the Stacking Classifier algorithm required 560 times longer computational time (7587.7686 seconds) than CatBoost (13.4635 seconds) without providing a commensurate performance improvement. This indicates that Boosting-based algorithms are able to capture complex patterns without requiring additional ensemble layers, so that an optimized single model is more recommended for real-world implementations by providing the best balance between prediction accuracy and computational time efficiency.

Keywords: Cardiovascular Disease; Stacking Classifier; K-Modes Clustering; Stratified Cross-Validation; Computational Efficiency

1. PENDAHULUAN

Penyakit kardiovaskular merupakan masalah kesehatan paling serius di dunia. Jumlah penderita diperkirakan akan terus meningkat pada tahun 2025. Sebagai contoh, pada tahun 2022 terdapat 19,8 juta kematian di seluruh dunia. Ini mencakup sekitar 32% dari seluruh kematian, dengan 85% di antaranya disebabkan oleh serangan jantung dan stroke [1]. Data dari Institute for Health Metrics and Evaluation (IHME) menunjukkan peningkatan kematian akibat penyakit kardiovaskular di Indonesia antara tahun 2020 hingga 2023. Kematian akibat penyakit kardiovaskular pada tahun 2020 memiliki persentase sekitar 5,51% dan pada tahun 2023 sekitar 11,47%, terjadi peningkatan dua kali lipat sekitar 5,96% yang mengindikasikan adanya peningkatan angka mortalitas [2]. World Health Organization (WHO) memperingatkan bahwa tanpa adanya upaya serius untuk mencegah, jumlah kematian dan kasus penyakit kardiovaskular akan terus meningkat.

Faktor risiko utama kardiovaskular seperti usia, tekanan darah, berat badan, kadar kolesterol, kadar glukosa, dan gaya hidup (merokok, olahraga, atau alkohol) merupakan fitur penting dalam pelatihan model *machine learning*. Permasalahan utama bagi model *single classifier* adalah karakteristik data yang sangat kompleks. Fitur alami penyakit kardiovaskular pada dasarnya adalah multifaktorial dan melibatkan hubungan yang sangat kompleks, seringkali bersifat non-linear antara berbagai faktor risiko tersebut [3]. Sebagai contoh, dampak dari kolesterol tinggi terhadap risiko kardiovaskular dapat diperburuk secara signifikan oleh kondisi status merokok. Hal ini merupakan suatu efek interaktif yang penting dan faktor risiko tersebut diketahui sangat terkait satu sama lain [4]. Dengan hal tersebut, model-model *single classifier* gagal menangani hubungan non-linear secara efektif [5] dan cenderung mengabaikan istilah interaksi (*interaction terms*) yang penting antar faktor risiko [4]. Kegagalan dalam menangkap pola-pola kompleks berujung pada performa prediktif yang kurang optimal dan keandalan yang terbatas pada model *single classifier*.

Menghadapi keterbatasan model *single classifier*, penerapan *Stacking Classifier* menjadi solusi efektif untuk menangani kompleksitas fitur dalam deteksi risiko kardiovaskular. Pendekatan ini terbukti efektif dalam meningkatkan akurasi klasifikasi risiko kardiovaskular dengan melatih *meta-learner* secara khusus untuk mempelajari pola terbaik ketika menggabungkan berbagai hasil prediksi dari *base learner* [6]. Strategi pemilihan algoritma *single classifier* sebagai komponen *Stacking Classifier* didasarkan dengan memaksimalkan keragaman model sekaligus mempertahankan performa model yang kuat [7]. CatBoost memanfaatkan teknik *Ordered Boosting* untuk menangani fitur kategorikal dan mencegah target *leakage* [8]. *Decision Tree* menerapkan interaksi fitur secara eksplisit melalui pemisahan rekursif yang memberikan perspektif berbeda dari model *boosting* [9]. *Logistic Regression* sebagai penyeimbang linear dengan memanfaatkan fungsi sigmoid untuk mengestimasi probabilitas kejadian dengan *varians* yang rendah [10]. XGBoost secara internal menggunakan integrasi regularisasi L1 (Lasso) dan L2 (Ridge) untuk mengontrol kompleksitas model dan mencegah overfitting secara efektif [11]. MLP memanfaatkan *hidden layers* dan aktivasi non-linear untuk memodelkan hubungan data yang sangat kompleks [12]. *Support Vector Classifier with Radial Basis Function Kernel* memanfaatkan kernel trick RBF untuk memetakan data ke ruang berdimensi tinggi dan menemukan *hyperplane* pemisah non-linear yang optimal [13]. Kombinasi algoritma yang beragam menjadikan *meta-learner* mampu mengekstraksi wawasan dari berbagai perspektif komputasi untuk menghasilkan keputusan akhir yang lebih akurat dan *robust*.

Berbagai studi sebelumnya telah membahas potensi algoritma *Stacking Classifier* dalam klasifikasi risiko kardiovaskular. Penelitian Bhatt et al. berfokus pada pra-pemrosesan data kardiovaskular dengan menerapkan *K-modes Clustering* untuk mengekstraksi fitur klaster dari data kategorikal menggunakan metode inisialisasi Huang. Fitur klaster tersebut ditambahkan sebagai fitur baru dan diuji pada beberapa algoritma *single classifier*. Terdapat dua skenario, yaitu tanpa *tuning* dan optimalisasi *hyperparameter GridSearchCV*. Hasilnya bahwa MLP yang dioptimalisasi dan menunjukkan akurasi tertinggi mencapai 87,28%, membuktikan efektivitas *K-Modes* sebagai teknik *feature engineering* [14]. Penelitian Khan et al. memanfaatkan algoritma *Stacking Classifier* untuk klasifikasi risiko kardiovaskular dan diabetes dengan *base-learner* terdiri dari *Naive Bayes*, *KNN*, *Linear Discriminant Analysis*, dan *Decision Tree*, dengan *SVM* sebagai *meta-learner*. Model *Stacking* mencapai akurasi 97,35% untuk data diabetes dan 88,71% untuk data kardiovaskular, membuktikan *SVM* sebagai *meta-learner* yang valid [15].

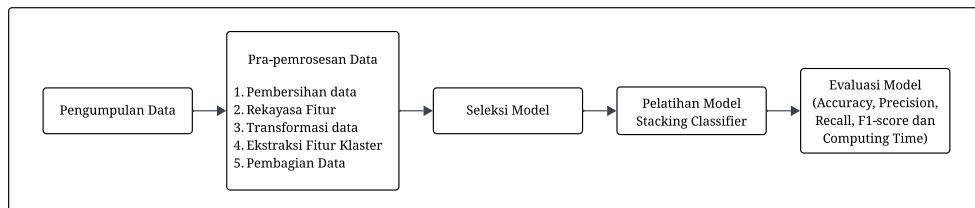
Penelitian Mohapatra et al. mengoptimalkan pra-pemrosesan data melalui *Z-score outlier detection* dan *min-max normalization* pada data kardiovaskular. Algoritma *stacking* menggabungkan 10 *base-learner* beragam, dengan MLP sebagai *meta-learner*. Model *stacking* lebih unggul dibandingkan *single classifier* dengan perbandingan akurasi mencapai 91,8%, *precision* 92,6%, *sensitivity* 92,6%, dan *specificity* 91% [16]. Penelitian Sultan et al. mengembangkan model NCDG yang berfokus pada penanganan *class imbalance* menggunakan teknik SMOTE dan BorderLineSMOTE. Arsitektur *Stacking Classifier* terdiri dari *Naive Bayes*, CatBoost, dan *Decision Tree* sebagai *base-learner* dan *Gradient Boosting* sebagai *meta-learner*. Model divalidasi menggunakan *K-Fold Cross Validation* dan menghasilkan akurasi 91%, *recall* 91%, *precision* 91%, *F1-Score* 91%, dan AUC-ROC 97%, mengungguli model *single classifier* [17].

Berdasarkan Tinjauan penelitian tersebut, ditemukan celah dalam optimalisasi deteksi risiko kardiovaskular. Penelitian Bhatt et al. telah membuktikan efektivitas *K-Modes Clustering* sebagai teknik *feature engineering*, namun penggunaannya masih terbatas pada model *single classifier*. Di sisi lain, studi mengenai *Stacking Classifier* oleh Khan et al. dan Mohapatra et al. telah menunjukkan performa unggul, tetapi belum mengintegrasikan fitur berbasis *clustering* untuk menangani kompleksitas data secara mendalam. Selain itu, penggunaan algoritma XGBoost dalam literatur *ensemble* sebelumnya yaitu Mohapatra et al. umumnya hanya diposisikan sebagai *base-learner*, sehingga potensi sebagai *meta-learner* dalam mengoreksi bias dari berbagai model heterogen belum dieksplorasi secara optimal.

Penelitian ini mengusulkan model hibrida yang mengintegrasikan fitur klaster dari *K-Modes Clustering* ke dalam algoritma *Stacking Classifier*. Pemilihan *K-Modes Clustering* didasarkan pada kemampuan menangani data kategorikal melalui *dissimilarity measure*, yang memberikan informasi baru dengan tetap mempertahankan distribusi data asli. Algoritma ini melibatkan lima *base-learner* (CAT, DT, MLP, SVC_RBF, dan LR) untuk menciptakan diversitas pola pembelajaran untuk menangkap hubungan non-linear yang kompleks. Kebaruan utama dalam penelitian ini terletak pada pemanfaatan XGBoost sebagai *meta-learner*. Dengan mekanisme *gradient boosting* dan fitur regulasi L1/L2 yang kuat, XGBoost diharapkan mampu mengoptimalkan prediksi akhir dengan mempelajari bobot kesalahan dari setiap *base-learner* secara lebih akurat dan *robust*. Secara teknis, penelitian ini menyajikan kerangka baru dalam *feature engineering* hibrida yang menggabungkan kekuatan pengelompokan data kategorikal dengan pembelajaran *ensemble* dan memberikan bukti baru mengenai efektivitas XGBoost sebagai pengambil keputusan tingkat akhir. Dengan penelitian ini, diharapkan menghasilkan sistem prediksi medis yang lebih akurat dan dapat menjadi landasan bagi praktisi kesehatan dalam pengambilan keputusan klinis berbasis data serta mendukung upaya deteksi dini risiko kardiovaskular secara lebih presisi.

2. METODOLOGI PENELITIAN

Proses penelitian ini menggunakan metode *Data Science Life Cycle*. Kerangka tersebut merupakan sebuah konseptualisasi penting dalam pengembangan ilmu data dengan menerapkan tahap-tahapan eksperimen [18]. Metodologi *Data Science Life Cycle* mencakup proses yang diperlukan untuk menganalisis, merancang, mengembangkan, dan mengevaluasi efektivitas model [19]. Langkah-langkah yang dilakukan dalam penelitian ini ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

Berdasarkan alur pada Gambar 1, proses penelitian dilakukan secara sistematis melalui tahapan pengumpulan data, pra-pemrosesan data, seleksi model, pelatihan model *Stacking Classifier*, hingga evaluasi model.

2.1 Pengumpulan Data

Penelitian ini memanfaatkan kumpulan data yang bersumber dari *platform Kaggle*. Dataset yang digunakan berjudul *Cardiovascular Disease Dataset*, dikumpulkan oleh Svetlana Ulianova. Dataset ini terdiri dari 70.000 data pasien dengan 12 atribut. Atribut yang digunakan untuk memprediksi risiko kardiovaskular meliputi fitur demografis (*age* dan *gender*), fitur antropometri (*weight* dan *height*), fitur pemeriksaan klinis (tekanan darah sistolik, tekanan darah diastolik, kolesterol, dan glukosa), dan fitur gaya hidup (merokok, alkohol, dan olahraga). Variabel target (label) adalah ada atau tidak adanya risiko terkena kardiovaskular yang direpresentasikan secara biner (0 untuk sehat dan 1 untuk terkena risiko).

2.2 Pra-pemrosesan Data

Pra-pemrosesan data merupakan tahap krusial dalam menyiapkan data mentah menjadi *input* berkualitas bagi algoritma. Kualitas data *input* secara langsung memengaruhi kinerja dan keandalan algoritma yang akan dibangun. Serangkaian proses akan dilakukan melalui beberapa tahapan, meliputi membersihkan, mentransformasi, dan memperkaya *dataset*.

2.2.1 Pembersihan Data

Proses pembersihan data dilakukan melalui tiga tahapan utama yaitu penanganan nilai yang hilang, penghapusan data duplikat, dan penanganan *outlier*. Tahap pertama adalah pengecekan nilai yang hilang pada seluruh atribut dalam dataset. Jika terdapat nilai yang hilang, maka atribut numerik diimputasi menggunakan nilai *mean*. Sementara atribut kategorikal diimputasi dengan nilai *modus*. Selanjutnya, pengecekan data duplikat untuk mengidentifikasi baris data yang identik. Data yang identik dihapus untuk memastikan setiap data mewakili individu unik dan menghindari bias dalam hasil pelatihan model.

Tahap selanjutnya proses penanganan *outlier*. Tahap ini penting dilakukan karena nilai ekstrem dapat menyebabkan distorsi pada distribusi data (*skewness*), menurunkan performa, dan keandalan generalisasi model. Penanganan ini difokuskan pada atribut yang bersifat numerik kontinu, seperti usia, berat badan, tinggi badan, tekanan darah sistolik dan tekanan darah diastolik. Sebagai langkah diagnostik, analisis *skewness* dilakukan menggunakan persamaan (1) untuk mengidentifikasi atribut dengan tingkat kemiringan tinggi yang mengindikasikan adanya *outlier*.

$$G_1 = \frac{n}{(n-1)(n-2)} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s} \right)^3 \quad (1)$$

Penjelasan persamaan:

- G_1 = koefisien kemiringan (*skewness*).
- n = jumlah total sampel dalam *dataset*.
- x_i = nilai data ke-i.
- \bar{x} = nilai rata-rata (*mean*) dari data.
- s = standar deviasi sampel.

Metode yang digunakan untuk mendeteksi *outlier* adalah *Interquartile Range* (IQR). Nilai dianggap sebagai *outlier* jika berada di luar rentang yang didefinisikan oleh kuartil pertama (Q1) dan kuartil ketiga (Q3). IQR diukur rentang antara Q1 dengan nilai persentil 25% data terendah dan Q3 dengan nilai persentil 75% data terendah, maka diperoleh IQR dengan mengurangi nilai Q3 dan Q1 seperti persamaan (2).

$$IQR = Q_3 - Q_1 \quad (2)$$

Dalam proses penentuan batas atas (*upper bound*) dan batas bawah (*lower bound*) dihitung menggunakan persamaan (3), dimana perhitungannya menjumlahkan nilai kuartil atas (Q3) dengan 1.5 kali nilai IQR serta mengurangkan nilai kuartil bawah (Q1) dengan 1.5 kali nilai IQR [20].

$$Upper Bound = Q_3 + (1.5 \times IQR), \quad Lower Bound = Q_1 - (1.5 \times IQR) \quad (3)$$

Nilai x_i diklasifikasikan sebagai *outlier* jika memenuhi kondisi $x_i < lower bound$ atau $x_i > upper bound$. Nilai *outlier* yang ditemukan kemungkinan merepresentasikan kesalahan *input* atau anomali pengukuran yang tidak mewakili populasi secara umum, sehingga data tersebut dihapus. Rangkaian proses pembersihan dilakukan untuk memastikan

model dilatih pada data yang berkualitas sehingga dapat meningkatkan performa dan kemampuan generalisasi algoritma pada data baru.

2.2.2 Rekayasa Fitur

Tahap rekayasa fitur dilakukan untuk menciptakan variabel prediktor baru yang lebih informatif dari atribut yang sudah ada, dengan tujuan meningkatkan performa prediktif model. Berdasarkan atribut *weight* dan *height* yang terdapat dalam dataset, fitur *Body Mass Index* (BMI) dihitung menggunakan persamaan (4). BMI merupakan indikator standar yang digunakan secara klinis untuk mengkategorikan status berat badan (kurang berat badan, normal, kelebihan berat badan, atau obesitas) dan merupakan faktor risiko utama penyakit kardiovaskular [21].

$$BMI = \frac{weight \text{ (kg)}}{height \text{ (m)}^2} \quad (4)$$

Terdapat perhitungan untuk fitur MAP (*Mean Arterial Pressure*) menggunakan persamaan (5). Fitur ini diturunkan dari *ap_hi* (tekanan darah sistolik) dan *ap_lo* (tekanan darah diastolik). MAP merupakan rata-rata tekanan arteri selama satu siklus jantung dan dianggap sebagai indikator perfusi organ yang lebih baik daripada tekanan darah sistolik atau diastolik [22]. Kedua fitur baru ini ditambahkan ke dalam dataset yang akan membantu model dalam *training* lebih lanjut.

$$MAP = \frac{ap_{hi} + (2 \times ap_{lo})}{3} \quad (5)$$

2.2.3 Transformasi Data

Tahap transformasi data mencakup proses *feature binning* dan *encoding data*. *Feature binning* dilakukan dengan mengubah fitur numerik kontinu (*age*, BMI dan MAP) menjadi kategorikal untuk menangkap hubungan linear dan menyederhanakan pola bagi model. Setiap kategori dibagi menjadi beberapa kategori berdasarkan metode *custom width binning* berbasis *domain-knowledge binning*. Proses pemetaan ini mengikuti persamaan (6), di mana nilai fitur numerik kontinu (x) ditransformasikan menjadi label kategori tertentu (k_i) apabila nilai tersebut berada dalam rentang ambang batas atau *bin boundaries* (b_i) yang telah ditentukan. Metode ini dipilih karena data risiko kardiovaskular bersifat non-linear dan sangat bergantung pada ambang batas (*threshold*) klinis yang sudah divalidasi.

$$y = f(x) = k_i, \text{ jika } b_{i-1} \leq x < b_i \quad (6)$$

Penentuan ambang batas (b_i) didasarkan pada *domain-knowledge binning* untuk menjamin sensitivitas klinis model terhadap risiko kardiovaskular. Fitur *age* dikategorikan dalam rentang lima tahun (*quinquennium*) melalui pendekatan $k=\lfloor age/5 \rfloor$. Pendekatan ini umum dalam studi epidemiologi untuk menangkap pola bagaimana risiko meningkat secara signifikan pada kelompok usia tertentu, yang mungkin tidak tertangkap oleh model linear [23]. Fitur BMI dikategorikan mengikuti standar global World Health Organization (WHO). Hal ini menyelaraskan data dengan definisi klinis obesitas yang merupakan faktor risiko utama kardiovaskular [24]. Fitur MAP dibagi ke dalam rentang spesifik untuk merepresentasikan status klinis perfusi organ, terutama dalam mengidentifikasi nilai kritis (misalnya, di bawah 70 mmHg yang memiliki arti klinis yang berbeda) [22].

Selanjutnya, *encoding data* diterapkan pada semua fitur kategorikal ke dalam format numerik menggunakan metode *label encoding*. Pendekatan ini merupakan fungsi pemetaan *one-to-one mapping* dari himpunan kategori ke C himpunan bilangan bulat menggunakan persamaan (7).

$$f: C \rightarrow \{0, 1, 2, \dots, n - 1\} \quad (7)$$

Di mana C yaitu himpunan kategori unik $C = \{c_1, c_2, \dots, c_n\}$. Setiap elemen c_i dipetakan ke nilai unik $f(c_i) = i$. Pendekatan ini mengubah data non-numerik menjadi representasi integer unik sebagai langkah awal untuk ekstraksi fitur klaster.

2.2.4 Ekstraksi Fitur Klaster

Ekstraksi fitur klaster dilakukan menggunakan algoritma *K-Modes Clustering* untuk mengidentifikasi pola tersembunyi pada data kategorikal dengan meminimalkan total *dissimilarity* (ketidakmiripan) melalui pendekatan *simple matching dissimilarity*. Proses pengelompokan berfokus pada minimalisasi fungsi biaya di dalam klaster (P) bagi seluruh sampel data pasien (n) berdasarkan jumlah klaster (k) yang ditetapkan, sebagaimana didefinisikan dalam persamaan (8).

$$P(W, Z) = \sum_{i=1}^n \sum_{l=1}^k w_{il} \cdot d(X_i, Z_l) \quad (8)$$

Di mana w_{il} merupakan matriks partisi biner yang bernilai jika objek i berada pada klaster l , sementara $d(X_i, Z_l)$ adalah ukuran ketidakmiripan antara vektor data pasien ke- i (X_i) dengan vektor pusat klaster ke- l (Z_l). Nilai ketidakmiripan ini dihitung berdasarkan jumlah kategori yang tidak cocok diseluruh fitur kategorikal (m) melalui persamaan (9).

$$(9)$$

$$d(X_i, Z_l) = \sum_{j=1}^m \delta(x_{ij}, z_{lj})$$

Fungsi pembanding δ memberikan nilai 0 jika terdapat kecocokan kategori ($x_{i,j} = z_{l,j}$) dan nilai 1 jika tidak cocok ($x_{i,j} \neq z_{l,j}$). Pusat klaster diperbarui secara berulang menggunakan nilai *modus* dari setiap fitur j untuk meminimalkan total biaya dalam klaster [25]. Penentuan jumlah klaster optimal (k) dilakukan melalui metode *elbow curve* dengan mengidentifikasi titik penurunan *cost* yang paling signifikan [26]. Inisialisasi pusat klaster menggunakan metode Huang yang mempertimbangkan frekuensi kategori untuk memastikan hasil klaster yang lebih stabil dan *robust* [25]. Label klaster kemudian ditambahkan ke *dataset* untuk memperkaya informasi bagi algoritma *stacking classifier* [27].

2.2.5 Pembagian Data

Proses pembagian data dipisahkan menjadi dua komponen berbeda yaitu *training set* (80%) dan *testing set* (20%). Pembagian data menjadi dua komponen dilakukan untuk mengevaluasi kinerja terhadap kemampuan generalisasi model secara objektif pada data yang belum pernah dilihat sebelumnya.

2.3 Seleksi Model

Seleksi model *base-learner* didasarkan pada pengujian dan evaluasi mendalam terhadap 11 kandidat *single classifier* (mencakup CAT, XGB, LGBM, GB, DT, RF, MLP, SVC_RBF, KNN, LR, dan Naïve Bayes) yang dioptimalkan melalui *hyperparameter tuning* menggunakan *GridSearchCV*. Metode ini dilakukan dengan menguji semua kombinasi *hyperparameter* yang telah didefinisikan dalam *grid parameter* menggunakan persamaan (10).

$$P^* = \arg \max_{P \in \theta} f(P) \quad (10)$$

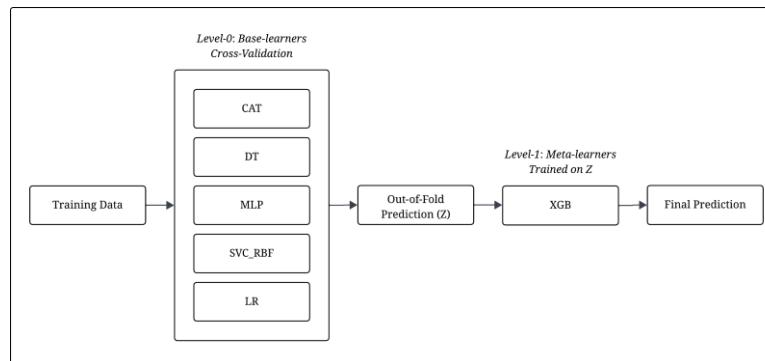
Proses optimasi bertujuan menemukan setelan *hyperparameter* optimal (P^*) di dalam ruang pencarian (θ) dengan memaksimalkan fungsi skor performa ($f(P)$), ditunjukkan pada persamaan (11). Model divalidasi menggunakan *StratifiedKFold 5-fold Cross Validation* untuk memastikan estimasi kinerja yang *robust* dan menjaga proporsi kelas target. Kinerja setiap model diukur berdasarkan nilai rata-rata performa model di seluruh lipatan menggunakan persamaan (11).

$$\bar{M} = \frac{1}{K} \sum_{i=1}^K M_i \quad (11)$$

Di mana rata-rata kinerja (\bar{M}) dihitung skor metrik setiap lipatan (M_i) untuk menentukan peringkat model. Berdasarkan evaluasi tersebut, lima algoritma terbaik dipilih sebagai *base learner* (*level-0*). Pendekatan ini bertujuan untuk menyediakan *input* yang beragam dan saling melengkapi bagi *meta-learner* (*level-1*), sehingga dapat meningkatkan kemampuan generalisasi model *stacking* secara keseluruhan.

2.4 Pelatihan Model Stacking Classifier

Proses pelatihan model menerapkan algoritma *Stacking Classifier* dengan dua tingkatan pembelajaran, disajikan pada Gambar 2. Arsitektur ini mengintegrasikan berbagai model untuk meningkatkan performa dan stabilitas prediksi.



Gambar 2. Arsitektur Model

Berdasarkan Gambar 2, proses pelatihan model dilakukan pada dua tingkatan yaitu *base-learner* (*level-0*) dan *meta-learner* (*level 1*). Pada *Level-0* (*Base-learner*), melibatkan lima model yang telah dioptimalkan melalui *GridSearchCV* (CAT, DT, MLP, SVC_RBF dan LR). Model ini divalidasi menggunakan teknik *StratifiedKFold 5-fold Cross Validation*. Setiap model dilatih pada 4 *fold* data latih dan menghasilkan *Out-of-Fold* (OOF) *Prediction* (Z) pada 1 *fold* sisanya. Secara matematis, kumpulan prediksi OOF untuk setiap model j didefinisikan seperti pada persamaan (12).

$$Z_{i,j} = h_j^{(i)}(x_k), \quad x_k \in f_i \quad (12)$$

Di mana setiap *base-learner* ($h_j^{(i)}$) memprediksi probabilitas (Z) pada sampel lipatan uji (x_k) dalam lima kali iterasi. Hasil prediksi OOF dari *base-learner* membentuk *Meta-Features* (*dataset Z*) yang diperkaya dengan fitur klinis asli dan label cluster melalui mekanisme *passthrough* sebelum diproses oleh *meta-learner*.

Pada tingkatan selanjutnya yaitu *Level-1 (Meta-Learner)*, XGBoost digunakan untuk memproses *dataset Z* untuk menghasilkan keputusan akhir (y) berdasarkan persamaan (13).

$$y = H(Z_1, Z_2, Z_3, Z_4, Z_5) \quad (13)$$

Di mana H mewakili algoritma XGBoost dalam memproses masukan dari kelima model sebelumnya untuk menghasilkan *Final Prediction* (y). Dengan mekanisme ini, XGBoost mampu mengoreksi kesalahan prediksi yang dihasilkan oleh *base-learner* secara kolektif untuk menghasilkan performa klasifikasi risiko kardiovaskular yang lebih *robust* dan akurat.

2.5 Evaluasi Model

Evaluasi model divalidasi menggunakan *Stratified 5-Fold Cross-Validation* dengan *dataset* dibagi lima *fold*; empat *fold* digunakan sebagai data latih dan satu *fold* sisanya sebagai data uji secara bergantian untuk menjamin generalisasi optimal dan menjaga keseimbangan distribusi kelas di setiap lipatan. Pengukuran performa didasarkan pada metrik *accuracy*, *precision*, *recall* dan *F1-score* menggunakan persamaan (14), (15), (16), dan (17) turunan dari komponen *confusion matrix*. Komponen tersebut meliputi keberhasilan identifikasi pada kelas positif (*True Positive (TP)*) dan negatif (*True Negative (TN)*), serta kesalahan klasifikasi pada data negatif (*False Positive (FP)*) dan positif (*False Negative (FN)*) [28].

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN} * 100\% \quad (14)$$

$$\text{precision} = \frac{TP}{TP + FP} * 100\% \quad (15)$$

$$\text{recall} = \frac{TP}{TP + FN} * 100\% \quad (16)$$

$$F1 - score = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (17)$$

Evaluasi diperkuat dengan metrik ROC-AUC, yang mengukur hubungan antara *True Positive Rate (TPR)* dan *False Positive Rate (FPR)* pada nilai *threshold* secara numerik [29]. Kualitas model dinilai melalui skor AUC (0 hingga 1), di mana nilai yang mendekati nilai 1 menunjukkan kemampuan pemisahan kelas yang sangat baik, sementara nilai 0.5 setara dengan prediksi acak [29]. Secara matematis, metrik ini dihitung menggunakan persamaan (18).

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \quad (18)$$

Evaluasi lain juga mencakup estimasi biaya komputasi yang diukur melalui waktu eksekusi (detik) untuk menilai efisiensi operasional algoritma selama proses pelatihan. Metrik ini dihitung berdasarkan selisih waktu antara dimulainya proses pelatihan hingga berakhirnya seluruh tahapan algoritma dalam satuan detik sesuai pada persamaan (19).

$$\text{Execution Time} = T_{selesai} - T_{mulai} \quad (19)$$

Evaluasi waktu ini penting untuk menilai efisiensi operasional algoritma *Stacking Classifier* yang kompleks agar dapat dipastikan bahwa model efektif diimplementasikan pada skenario klinis nyata.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Dataset ini mencakup 70.000 sampel dengan 12 fitur yang diklasifikasikan ke dalam dua label, yaitu tidak berisiko (0) dan berisiko (1). Rincian sampel *dataset* disajikan pada Tabel 1.

Tabel 1. Sampel Dataset

age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
50	2	168	62.0	110	80	1	1	0	0	1	0
55	1	156	85.0	140	90	3	1	0	0	1	1

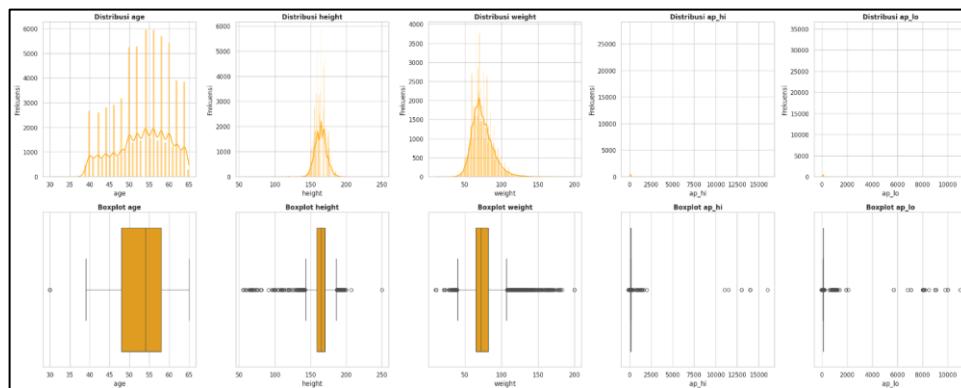
Berdasarkan Tabel 1, *dataset* ini mengintegrasikan fitur demografis, klinis, dan gaya hidup sebagai variabel independen untuk memprediksi variabel target *cardio*. Distribusi variabel target menunjukkan keseimbangan, yaitu 35.021 sampel (50,03%) untuk kelas 0 dan 34.979 sampel (49,97%) untuk kelas 1. Proporsi yang seimbang ini penting untuk meminimalkan risiko bias model terhadap kelas mayoritas sehingga menjamin validitas dan keandalan performa.

3.2 Pra-pemrosesan Data

3.2.1 Pembersihan Data

Tahap pembersihan data menunjukkan bahwa dari 70.000 sampel tidak ditemukan adanya data yang hilang pada 12 fitur. Kondisi awal mengonfirmasi data yang baik dan tidak perlu dilakukan imputasi data. Proses pembersihan data kedua adalah mengidentifikasi data duplikat, di mana terdapat 24 baris data duplikat. Seluruh sampel duplikat dihapus dari *dataset* untuk mencegah bias dalam pelatihan model, sehingga jumlah data yang terverifikasi terdapat 69.976 sampel.

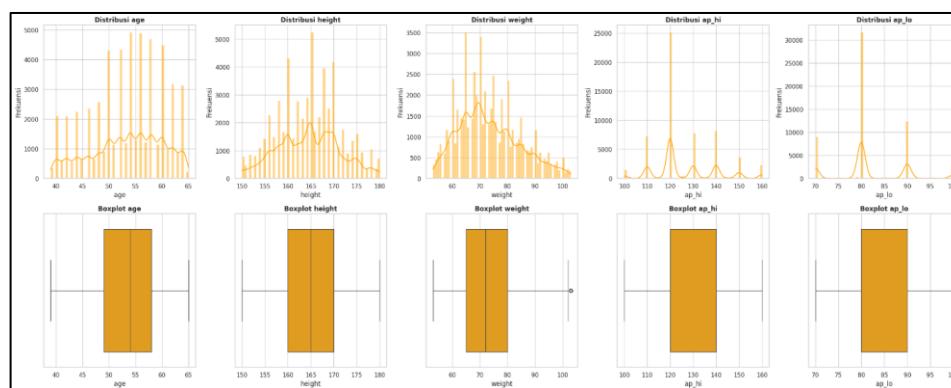
Selanjutnya dilakukan analisis distribusi fitur untuk mendeteksi keberadaan *outlier* yang dapat mengganggu performa algoritma menggunakan persamaan (1). Visualisasi distribusi data disajikan pada Gambar 3.



Gambar 3. Distribusi data dan deteksi *Outlier* (Sebelum)

Berdasarkan Gambar 3, analisis distribusi awal menunjukkan adanya kemiringan ekstrem (*skewness*) pada beberapa fitur akibat nilai yang tidak valid secara klinis, seperti tekanan darah dengan nilai dari 10.000 mengindikasikan adanya kesalahan *input* data. Nilai *skewness* tercatat sebesar 85,280 untuk *ap_hi*, 33,534 untuk *ap_lo*, 1,008 untuk *weight*, -0,532 untuk *height*, dan -0,307 untuk *age*. Melatih model *machine learning* pada data yang mustahil secara fisik akan menghasilkan model yang tidak dapat digunakan di dunia nyata sehingga perlu dilakukan penanganan *outlier*.

Langkah mengembalikan data ke rentang klinis yang valid, melalui eliminasi *outlier* menggunakan metode IQR. menggunakan persamaan (2) dan (3). Metode ini diterapkan secara spesifik pada lima fitur numerik, yaitu *age*, *weight*, *height*, *ap_hi*, dan *ap_lo*. Proses ini berhasil mengidentifikasi dan menghapus 12.742 sampel yang berada di luar batas kewajaran. Perubahan ini efektif mengurangi kemiringan (*skewness*) dan menstabilkan sebaran data. Peningkatan kualitas data pada pembersihan data dapat dilihat pada Gambar 4.



Gambar 4. Distribusi data dan deteksi *Outlier* (Sesudah)

Implementasi metode IQR berhasil mengembalikan distribusi ke dalam rentang klinis yang valid, sebagaimana divisualisasikan pada Gambar 4. Secara kuantitatif, efektivitas proses ini ditunjukkan oleh penurunan nilai *skewness* yang sangat signifikan terutama pada fitur tekanan darah di mana *ap_hi* turun dari 85,280 menjadi 0,700 dan *ap_lo* turun dari 33,534 menjadi 0,456. Hal tersebut juga terjadi pada fitur *weight* yang membaik dari 1,008 menjadi 0,485, *height* dari -0,532 menjadi 0,050 dan *age* dari -0,307 menjadi -0,313. Hasil Pembersihan ini terlihat jelas pada Gambar 4, dimana sebaran data kini lebih terkonsentrasi pada nilai yang masuk akal secara medis. Melalui tahapan eliminasi *outlier*, dihasilkan *dataset* akhir sebanyak 57.237 sampel.

3.2.2 Rekayasa Fitur

Tahap rekayasa fitur dilakukan pada 57.237 sampel yang telah melewati proses pembersihan data. Proses rekayasa fitur BMI ini dihasilkan melalui persamaan (4), di mana data *height* pada *dataset* menggunakan satuan sentimeter (cm), maka dilakukan konversi ke meter (m) terlebih dahulu sebelum diterapkan pada persamaan (4). Selain itu, proses rekayasa fitur

MAP juga dihasilkan melalui persamaan (5), di mana nilai MAP dihitung dengan menggabungkan dua variabel tekanan darah, yaitu ap_hi dan ap_lo melalui persamaan (5). Sebagai representasi data yang telah diproses, struktur *dataset* yang telah diperbarui dengan fitur baru disajikan pada Tabel 2.

Tabel 2. Sampel Dataset Rekayasa Fitur

age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio	BMI	MAP
50	2	168	62.0	110	80	1	1	0	0	1	0	21.97	90.00
55	1	156	85.0	140	90	3	1	0	0	1	1	34.93	106.67

Berdasarkan data yang disajikan pada Tabel 2, penelitian ini menghasilkan dua fitur baru yang bersifat informatif secara klinis untuk membantu proses klasifikasi. Langkah rekayasa fitur ini dilakukan untuk meningkatkan kualitas dan relevansi klinis dari prediktor. Fitur BMI dihasilkan dengan menggabungkan dua variabel (*height* dan *weight*) menjadi satu metrik standar yang diakui secara global untuk mengukur obesitas (faktor risiko utama kardiovaskular). Fitur MAP juga memberikan representasi yang lebih stabil dan bermakna terhadap perfusi organ dibandingkan nilai *ap_hi* atau *ap_lo* secara terpisah. Hal ini penting karena nilai *ap_hi* dan *ap_lo* sering kali dapat berfluktuasi secara independen.

3.2.3 Transformasi Data

Tahap transformasi data dilakukan dengan *feature binning* dan proses *encoding*. Proses *feature binning* diterapkan pada tiga fitur numerik kontinu, yaitu *age*, BMI, dan MAP menggunakan metode *custom width binning* berbasis *domain-knowledge binning*. Pemetaan dilakukan sesuai dengan persamaan (6) yang mentransformasikan nilai kontinu x menjadi label diskret y. Sebagai contoh, fitur MAP dikategorikan berdasarkan status perfusi organ (seperti $70 < \text{MAP} < 80$), yang dikelompokkan dalam representasi status klinis tertentu yang lebih akurat dibandingkan nilai mentah.

Selanjutnya, proses *encoding* diterapkan pada 9 variabel prediktor menggunakan metode *Label Encoding*. Metode ini mentransformasikan setiap label kategori menjadi representasi bilangan bulat (integer) menggunakan persamaan (7). Proses *label encoding* bekerja dengan mengidentifikasi kategori, pemetaan indeks dan mengganti nilai tekstual dengan angka indeks pasangan untuk menghasilkan format numerik yang seragam. Sebagai hasil dari seluruh rangkaian transformasi tersebut, struktur *dataset* yang telah dikonversi ke dalam format integer dapat dilihat pada Tabel 3.

Tabel 3. Sampel Dataset Sesudah Transformasi Data

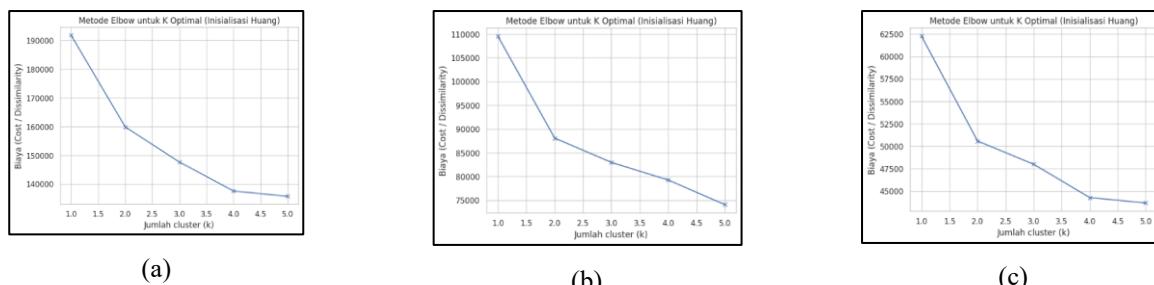
gender	age_bin	BMI_bin	MAP_bin	cholesterol	gluc	smoke	alco	active	cardio
1	2	1	1	0	0	0	0	1	0
0	3	4	2	2	0	0	0	1	1

Berdasarkan data pada Tabel 3, terlihat bahwa seluruh fitur prediktor maupun variabel target telah memiliki format numerik yang seragam. Proses ini bukan hanya mengubah format, melainkan metode yang digunakan untuk memastikan data kompatibel dengan algoritma *K-Modes Clustering* untuk tahap ekstraksi fitur *cluster* selanjutnya. Algoritma tersebut berfokus pada analisis kesamaan kategori antar data, bukan jarak matematis (*Euclidean Distance*). Transformasi ke dalam kategori numerik seperti yang ditunjukkan pada Tabel 4 menjadi prasyarat agar ekstraksi fitur kluster dapat berjalan akurat.

3.2.4 Ekstraksi Fitur Klaster

Tahap ini bertujuan mengidentifikasi struktur pengelompokan pola tersembunyi pada data kategorikal menggunakan algoritma *K-Modes Clustering* dengan metode inisialisasi Huang. Algoritma bekerja dengan meminimalkan total *cost* menggunakan persamaan (8) melalui akumulasi ketidakmiripan yang didefinisikan pada persamaan (9). Perhitungan ketidakmiripan dilakukan dengan membandingkan setiap fitur (j) antara data pasien dan pusat klaster. Sebagai contoh, perbandingan antara data pasien [*age*:4, BMI:1, MAP:1] dengan pusat klaster [*age*:4, BMI:2, MAP:2] menghasilkan nilai ketidakmiripan $0+1+1=2$. Berdasarkan nilai terkecil, setiap data dialokasikan ke klaster tertentu dan diikuti pembaruan pusat klaster secara berulang hingga mencapai kondisi konvergen di mana tidak ada lagi perubahan keanggotaan.

Penentuan jumlah klaster optimal (K) dievaluasi menggunakan metode *Elbow* pada tiga skema, yaitu seluruh data, subset pria, dan subset perempuan. Analisis pada Gambar (5) menyajikan grafik evaluasi untuk mengidentifikasi titik keseimbangan antara kompleksitas model dan minimalisasi *cost* guna menjamin kualitas pengelompokan yang dihasilkan.



Gambar 5. Grafik Analisis Metode Elbow: (a) Seluruh Data, (b) Subset Pria, (c) Subset Perempuan

Berdasarkan grafik yang disajikan pada Gambar 5, ketiga grafik menunjukkan pola penurunan yang konsisten, di mana penurunan *cost* yang paling tajam terjadi pada rentang $K=1$ ke $K=2$. Meskipun *cost* menurun pada $K > 2$, besaran penurunannya menjadi semakin kecil dan landai dibandingkan dengan penurunan awal yang tajam. Berdasarkan karakteristik kurva tersebut, ditetapkan $K = 2$ sebagai jumlah klaster optimal. Hal ini didasarkan pada prinsip parsimoni (kesederhanaan), di mana titik $K = 2$ merepresentasikan pemisahan penting dalam populasi data dan mengindikasikan bahwa pasien secara alami terbagi menjadi ke dalam dua kategori dominan yang saling bertolak belakang.

Karakteristik spesifik tiap klaster diidentifikasi melalui analisis pusat klaster yang dilakukan terpisah untuk subset pria dan perempuan. Hasil ekstraksi *centroid* tersebut disajikan pada Tabel 4 (pria) dan Tabel 5 (perempuan).

Tabel 4. Pusat Klaster (*Centroid*) pada Subset Pria

	gender	age bin	BMI bin	MAP bin	cholesterol	gluc	alco	smoke	active	cardio
Klaster 0	0	4	1	1	0	0	0	0	1	0
Klaster 1	0	5	2	2	0	0	0	0	1	1

Tabel 5. Pusat Klaster (*Centroid*) pada Subset Perempuan

	gender	age bin	BMI bin	MAP bin	cholesterol	gluc	alco	smoke	active	cardio
Klaster 0	1	4	1	1	0	0	0	0	1	0
Klaster 1	1	5	2	2	0	0	0	0	1	1

Hasil pada Tabel 4 dan Tabel 5 menunjukkan algoritma *K-Modes Clustering* berhasil mengidentifikasi dua pola risiko yang konsisten pada kedua subset gender. Klaster 0 merepresentasikan segmen pasien risiko rendah, yang ditandai oleh kelompok usia lebih muda (age_bin 4) serta parameter klinis optimal pada kategori BMI_1 dan MAP_1. Mayoritas klaster 0 tidak terdeteksi risiko kardiovaskular (cardio 0). Sebaliknya, klaster 1 menangkap pola komorbiditas yang jelas. Pola ini ditandai oleh interaksi antara peningkatan usia (age_bin 5), berat badan berlebih (BMI_bin 2) dan hipertensi (MAP_bin 2). Mayoritas klaster 1 didominasi kuat oleh pasien yang terdiagnosis penyakit kardiovaskular (cardio 1).

Keberhasilan *K-Modes Clustering* dalam mengidentifikasi pola risiko membuktikan adanya struktur internal yang kuat pada data kategorikal. Hasil klaster dimasukkan ke dalam *dataset* sebagai fitur prediktor baru. Penambahan fitur berfungsi memberikan sinyal *meta-feature* dalam memetakan profil risiko kompleks menjadi satu variabel diskrit yang informatif. Dalam algoritma *Stacking Classifier*, fitur klaster tersebut secara sistematis memperjelas batas keputusan (*decision boundary*) untuk memisahkan kelas positif dan negatif, terutama pada kasus gejala klinis yang ambigu.

3.2.5 Pembagian Data

Dataset dengan total 57.237 sampel dibagi menggunakan rasio 80:20 untuk menjamin konsistensi dan reproducibilitas hasil pelatihan. Berdasarkan distribusi pada Tabel 6, 45.789 sampel (80%) ditetapkan sebagai data latih dan 11.448 sampel (20%) sebagai data uji. data uji ini berfungsi sebagai *hold-out set* yang dikunci agar tidak terlihat oleh model selama proses pelatihan, sehingga validitas pengujian performa tetap objektif dan andal.

Tabel 6. Distribusi Data Latih dan Data Uji

Jenis Data	Percentase	Jumlah Sampel
Data Latih (<i>Training Set</i>)	45.789	80%
Data Uji (<i>Testing Set</i>)	11.448	20%
Total	52.237	100%

Berdasarkan Tabel 6, Pembagian data dilakukan dengan rasio 80:20 untuk memenuhi kebutuhan kompleksitas model *Stacking Classifier*. Alokasi 45.789 sampel (80%) pada data latih menyediakan volume yang cukup bagi model untuk mempelajari interaksi fitur asli dan fitur klaster secara mendalam untuk mencegah *overfitting*. Di sisi lain, penggunaan 11.448 sampel (20%) sebagai data uji menjamin hasil evaluasi yang meyakinkan secara statistik dan objektif untuk merepresentasikan kinerja model di dunia nyata. Dengan kondisi kelas target yang seimbang, pembagian acak ini secara konsisten menjaga proporsi kelas pada kedua himpunan data, sehingga validitas proses evaluasi tetap terjaga.

3.3 Seleksi Model

Seleksi model dilakukan dengan mengevaluasi 11 kandidat *single classifier* melalui optimasi *tuning hyperparameter* (*GridSearchCV*) untuk menemukan kombinasi parameter optimal (P^*) dalam ruang pencarian, sesuai persamaan (10). Proses ini menghasilkan setelan parameter terbaik seperti *learning rate*, *max depth*, atau *n_estimators*. Parameter ini yang memastikan setiap model bekerja pada kapasitas maksimal sebelum dibandingkan satu sama lain.

Setelah optimasi parameter, setiap model divalidasi menggunakan *Stratified 5-Fold Cross-Validation* untuk memperoleh estimasi kinerja yang objektif dan *robust*. Kinerja akhir setiap model dihitung berdasarkan nilai rata-rata menggunakan persamaan (11). Hasil pada Tabel 7 menunjukkan dominasi algoritma *boosting* dengan CAT mencatat kinerja tertinggi dengan *Overall Average* 89,63%, disusul tipis oleh XGB sebesar 89,62%. Model berbasis pohon seperti DT dan RF menunjukkan performa kompetitif dengan rata-rata 89,44% dan 89,40%. Sementara model *non-tree* seperti MLP (89,33%) dan SVC_RBF (89,13%) menunjukkan kapabilitas kuat dalam menangkap pola data yang kompleks.

Meskipun model linear seperti LR tertinggal dengan *Overall Average* 83,96%, model ini tetap diintegrasikan ke dalam *stacking* karena karakteristik varians yang berbeda dari model kompleks lainnya.

Tabel 7. Hasil Evaluasi Seleksi Model

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Overall Average	Computing Time
CAT	88,03%	88,20%	88,03%	88,02%	95,90%	89,63%	13,4635s
XGB	88,01%	88,19%	88,01%	87,99%	95,89%	89,62%	5,1961s
LGBM	87,99%	88,17%	87,99%	87,97%	95,86%	89,60%	7,3890s
GB	87,97%	88,14%	87,97%	87,96%	95,83%	89,57%	15,1887s
DT	87,83%	87,99%	87,83%	87,81%	95,74%	89,44%	0,0622s
RF	87,79%	87,90%	87,79%	87,78%	95,73%	89,40%	8,5578s
MLP	87,68%	87,90%	87,68%	87,66%	95,73%	89,33%	817,7318s
SVC_RBF	87,81%	87,95%	87,81%	87,79%	94,31%	89,13%	529,2151s
KNN	86,84%	86,86%	86,85%	86,83%	94,59%	88,39%	0,1487s
LR	82,66%	82,86%	82,66%	82,62%	89,03%	83,96%	0,2992s
Naïve Bayes	70,14%	70,84%	70,14%	69,82%	78,12%	71,81%	0,0537s

Berdasarkan Tabel 7, lima model heterogen dipilih sebagai *base-learner* (*level-0*) untuk memaksimalkan efektivitas pelatihan model dan meminimalkan korelasi antar-error. Kelima algoritma ini memiliki peran komplementer: (1) CAT unggul dalam menangani fitur kategorikal; (2) DT memiliki interpretabilitas dan kemampuan menangkap pola interaksi fitur secara eksplisit; (3) MLP memodelkan hubungan non-linear kompleks melalui *hidden layers* dengan memberikan perspektif global yang berbeda dari pemisahan hierarki model pohon; (4) SVC_RBF memetakan data ke dimensi tinggi untuk menemukan *hyperplane* pemisah; dan (5) LR bertindak sebagai penyeimbang linear. Sebagai integrator di *level-1*, XGBoost ditetapkan sebagai *meta-learner* karena kemampuannya mempelajari pola koreksi yang rumit dari kombinasi prediksi heterogen. Mekanisme regularisasi yang kuat pada XGBoost efektif mencegah *overfitting* selama tahap agregasi, sehingga menghasilkan generalisasi model yang lebih akurat dan stabil.

3.4 Pelatihan Model *Stacking Classifier*

Tahap pelatihan dilakukan dalam dua tingkatan dengan mengintegrasikan lima *base-learner* heterogen (CAT, DT, MLP, SVC_RBF, dan LR) untuk meningkatkan performa dan stabilitas prediksi risiko kardiovaskular. Pada *Level-0*, setiap model dilatih menggunakan teknik *Stratified 5-Fold Cross-Validation* untuk menghasilkan prediksi *Out-of-Fold* (OOF), yang berfungsi mencegah kebocoran data selama proses pelatihan. Sesuai dengan persamaan (12), setiap *base-learner* (j) pada iterasi (i) menghasilkan nilai prediksi probabilitas (Z) yang kemudian digabungkan dengan fitur *passthrough* (data klaster dan variabel klinis) untuk membentuk *meta-features*. Pengetahuan kolektif ini dirinci pada Tabel 8 dan Tabel 9, memberikan landasan informasi yang kaya bagi *meta-learner* untuk melakukan klasifikasi akhir secara lebih akurat.

Tabel 8. Meta-Features Probabilitas (*Level-0*)

Indeks	CAT_Proba	DT_Proba	MLP_Proba	SVC_RBF_Proba	LR_Proba
0	0.004050	0.0	2.805315e-11	0.62611	0.434417
1	0.477752	0.642857	8.958141e-02	0.289368	0.480391
2	0.001360	0.0	1.412361e-06	0.060130	0.277788
3	0.999111	1.0	6.180076e-01	0.994257	0.880142
4	0.000295	0.0	1.373506e-06	0.088481	0.072003

Tabel 9. Meta-Features *Passthrough*

Indeks	Cluster	Gender	age bin	BMI bin	MAP bin	cholesterol	gluc	alco	smoke	active
0	0.0	0.0	0.0	4.0	2.0	2.0	0.0	0.0	0.0	1.0
1	0.0	0.0	0.0	1.0	1.0	2.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	2.0	1.0	1.0	1.0	0.0	0.0	0.0
3	1.0	0.0	0.0	4.0	1.0	2.0	0.0	0.0	0.0	1.0
4	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	1.0

Analisis pada Tabel 8 menunjukkan variasi skor keyakinan antar model, di mana pada indeks 3, model CAT (0.999111) dan DT (1.0) menunjukkan kesepakatan prediksi yang sangat tinggi. Selain nilai probabilitas, algoritma ini menerapkan *passthrough* yang ditunjukkan pada Tabel 9, di mana meneruskan fitur secara langsung sebagai *input* tambahan bagi tingkatan selanjutnya. Integrasi fitur klaster hasil *K-Modes* terbukti memberikan sinyal profil risiko tambahan yang memperkaya informasi bagi *meta-learner* dalam mengoptimalkan hasil klasifikasi akhir.

Stabilitas performa model selama tahap pelatihan model sangat bergantung pada konfigurasi *hyperparameter* yang diterapkan. Berdasarkan hasil optimasi menggunakan *GridSearchCV*, setiap komponen model *base-learner* dan *meta-learner* dikonfigurasi dengan parameter spesifik seperti yang dirincikan pada Tabel 10.

Tabel 10. Parameter Optimal Hasil *GridSearchCV*

Peran Model	Algoritma	Parameter Optimal
<i>Base-Learner</i>	CAT	depth: 6, iterations: 500, l2_leaf_reg: 5, learning_rate: 0.03, subsample: 1.0
	DT	criterion: entropy, max_depth: 10, min_samples_leaf: 4, min_samples_split: 10, max_features: None
	MLP	alpha: 0.001, hidden_layer_sizes: (256, 128), learning_rate_init: 0.0001, max_iter: 500
	SVC_RBF	C: 2.0, class_weight: None, gamma: auto, kernel: rbf, probability: True
	LR	C: 0.1, penalty: l1, solver: liblinear, max_iter: 1000
<i>Meta-Learner</i>	XGB	Colsample_bytree: 0.8, learning_rate: 0.01, max_depth: 6, n_estimators: 800, subsample: 0.8

Analisis pada Tabel 10 menunjukkan strategi penyesuaian parameter yang berfokus pada penguatan generalisasi dan pencegahan *overfitting*. CAT dikonfigurasi dengan depth: 6 dan iterations: 500, serta penggunaan l2_leaf_reg: 5 sebagai regulasi L2 untuk mereduksi sensitivitas terhadap *noise* pada fitur kategorikal hasil *K-Modes*, sehingga menghasilkan prediksi yang lebih stabil. Sejalan dengan itu, DT menerapkan mekanisme *pre-pruning* melalui batasan max_depth: 10 dan min_samples_leaf: 4. Pembatasan ini dilakukan untuk mencegah pohon tumbuh terlalu kompleks berlebih, sehingga model lebih efektif dalam menangkap struktur keputusan global dari data klinis secara lebih efektif.

MLP menggunakan struktur *hidden layers* (256, 128) dan regulasi alpha: 0.0001 untuk memodelkan hubungan non-linear yang kompleks sekaligus menjaga stabilitas gradien. Pendekatan ini diperkuat oleh SVC_RBF yang memproyeksikan data ke dimensi tinggi untuk pemisahan linear serta menyediakan skor probabilitas sebagai *input* kaya informasi bagi *meta-learner*. Sebagai penyeimbang, LR menerapkan regularisasi L1 (penalty: 11) dengan nilai C: 0.1 untuk melakukan seleksi fitur implisit dengan menyusutkan koefisien fitur yang kurang relevan menjadi nol. Model tetap sederhana dan efektif dalam mereduksi dampak fitur variabel yang tidak signifikan terhadap hasil akhir klasifikasi.

Setelah data dari *base-learner* terkumpul dalam *dataset Z*, tahap selanjutnya adalah pemrosesan pada *Level-1* (*meta-learner*) menggunakan XGBoost. Algoritma ini mengintegrasikan skor probabilitas dan fitur *passthrough* menjadi satu keputusan akhir (y) berdasarkan persamaan (14). XGBoost dikonfigurasi dengan strategi *slow learning* melalui learning_rate: 0.01 dan n_estimator: 800 agar model dapat mempelajari pola koreksi dari *base-learner* secara bertahap dan mendetail. Selain itu, menerapkan subsample: 0.8 (penggunaan 80% data secara acak) untuk mencegah ketergantungan pada sampel tertentu, sehingga meningkatkan kemampuan generalisasi terhadap data baru.

Hasil dari proses perhitungan pada persamaan (13) tersebut disajikan pada Tabel 11, yang memuat skor probabilitas dan label prediksi akhir untuk sampel data pasien.

Tabel 11. Final Prediction

Indeks	MetaLearner_Proba	MetaLearner_Pred
0	0.000465	0
1	0.999754	1
2	0.000229	0
3	0.999757	1
4	0.000251	0

Analisis pada Tabel 11 menunjukkan bahwa proses *level-1* menghasilkan keputusan klasifikasi yang sangat tegas dengan nilai probabilitas yang mendekati 0 atau 1. Sebagai contoh, skor pada Indeks 1 (0.999754) dan Indeks 0 (0.000465) mencerminkan tingkat keyakinan tinggi XGBoost dalam mengintegrasikan prediksi *base-learner* dan data klinis *passthrough*. Secara keseluruhan, penerapan parameter optimal dan mekanisme penggabungan ini menciptakan sinergi prediksi yang akurat, stabil, dan andal untuk data medis pasien kardiovaskular. Dengan mengoreksi bias kolektif dari model-model dasar, *Stacking Classifier* mampu menghasilkan batas keputusan (*decision boundary*) yang sangat presisi.

3.5 Evaluasi Model

Model dievaluasi menggunakan *Stratified 5-Fold Cross-Validation* untuk menjamin generalisasi yang optimal. Performa diukur melalui metrik *accuracy*, *precision*, *recall*, *F1-score* dan ROC-AUC sesuai persamaan (14) hingga (18). Analisis diperdalam dengan nilai rata-rata performa sesuai persamaan (11) serta efisiensi komputasi melalui waktu eksekusi (detik) berdasarkan persamaan (19). Hasil evaluasi model *Stacking Classifier* disajikan pada Tabel 12.

Tabel 12. Hasil Evaluasi Algoritma *Stacking Classifier*

Model	accuracy	precision	recall	F1-Score	ROC-AUC	Overall Average	Computing Time
Stacking	87,99%	88,13%	87,99%	87,97%	95,89%	89,59%	7587,7686s

Berdasarkan hasil evaluasi pada Tabel 12, model *Stacking* menunjukkan kinerja solid dengan akurasi mencapai 87,99%. Dalam konteks diagnosis medis, akurasi bukan satu-satunya tolak ukur. Metrik *precision* (88,13%), *recall* (87,99%), dan *F1-score* sebesar 87,97%, menunjukkan selisih yang sangat tipis antara metrik tersebut mengindikasikan model memiliki stabilitas yang baik dan minimalisasi alarm palsu. Kemampuan model dalam membedakan kelas positif

dan negatif juga terbukti sangat kuat dengan skor ROC-AUC 95,89%, menghasilkan *Overall Average* sebesar 89,59%. Namun, algoritma ini memakan waktu komputasi yang sangat tinggi dengan estimasi waktu mencapai 7587,7686 detik.

Efektivitas algoritma *Stacking Classifier* divalidasi melalui perbandingan dengan seluruh kandidat model, sebagaimana dirinci pada Tabel 13. Secara teoritis, algoritma ini dirancang untuk meningkatkan kinerja dengan menggabungkan kekuatan berbagai *base-learners*. Namun, analisis pada Tabel 13 menunjukkan temuan penting bahwa kompleksitas algoritma *Stacking* tidak selalu berbanding lurus dengan peningkatan performa yang signifikan.

Tabel 13. Hasil Perbandingan Evaluasi Semua Model

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Overall Average	Computing Time
CAT	88,03%	88,20%	88,03%	88,02%	95,90%	89,63%	13,4635s
XGB	88,01%	88,19%	88,01%	87,99%	95,89%	89,62%	5,1961s
LGBM	87,99%	88,17%	87,99%	87,97%	95,86%	89,60%	7,3890s
Stacking	87,99%	88,13%	87,99%	87,97%	95,89%	89,59%	7587,7686s
GB	87,97%	88,14%	87,97%	87,96%	95,83%	89,57%	15,1887s
DT	87,83%	87,99%	87,83%	87,81%	95,74%	89,44%	0,0622s
RF	87,79%	87,90%	87,79%	87,78%	95,73%	89,40%	8,5578s
MLP	87,68%	87,90%	87,68%	87,66%	95,73%	89,33%	817,7318s
SVC_RBF	87,81%	87,95%	87,81%	87,79%	94,31%	89,13%	529,2151s
KNN	86,84%	86,86%	86,85%	86,83%	94,59%	88,39%	0,1487s
LR	82,66%	82,86%	82,66%	82,62%	89,03%	83,96%	0,2992s
Naïve Bayes	70,14%	70,84%	70,14%	69,82%	78,12%	71,81%	0,0537s

Hasil perbandingan evaluasi semua model pada Tabel 13 menunjukkan temuan menarik bahwa model *single classifier* berbasis *Gradient Boosting* mendominasi peringkat teratas. CAT mencatat kinerja tertinggi dengan akurasi 88,03%, *precision* 88,20%, *recall* 88,03%, *F1-Score* 88,02%, dan ROC-AUC 95,90%, sedikit mengungguli model *Stacking* yang berada di posisi kelima dengan akurasi 87,99%. Terdapat hal mencolok terletak pada efisiensi waktu komputasi, model *Stacking* memerlukan durasi pelatihan yang panjang mencapai 7587,7686 detik. Terlihat sangat kontras dengan CAT yang hanya membutuhkan 13,4635 detik untuk menghasilkan performa yang lebih unggul.

Perbedaan mencolok ini mengindikasikan adanya *trade-off* yang ekstrem antara kompleksitas model dan efisiensi komputasi. Algoritma *Stacking Classifier* meningkatkan waktu komputasi hingga 560 kali lipat tanpa memberikan peningkatan performa dibandingkan model *single classifier* yang telah dioptimasi. Temuan ini mengindikasikan beberapa keterkaitan penting mengenai efektivitas arsitektur model. Dominasi algoritma berbasis *Boosting* sangat terlihat jelas, di mana algoritma tersebut terbukti sangat *robust* dalam menangkap pola non-linear hingga mencapai batas optimal. Sebaliknya, kinerja *base-learner* cenderung bersifat *redundant* sehingga *meta-learner* (XGBoost) tidak memperoleh *input* tambahan yang cukup signifikan untuk meningkatkan performa gabungan secara signifikan.

Hasil penelitian ini memvalidasi efektivitas *K-Modes Clustering* dan membuktikan bahwa *Stacking Classifier* (87,99) mampu mengungguli model MLP (87,28%) milik Bhatt et al. melalui penerapan *Stratified 5-Fold Cross-Validation* yang memberikan estimasi performa lebih andal. Meskipun akurasinya sedikit di bawah Sultan et al. (91%), penelitian ini menjaga integritas data klinis asli tanpa risiko bias dari data sintetik (SMOTE). Temuan ini memperkuat kesimpulan Liu et al. bahwa kompleksitas *ensemble* tidak selalu menghasilkan performa yang sepadan. Hasil penelitian ini menegaskan penggunaan model *single classifier* yang canggih dan teroptimasi jauh lebih baik secara komputasi dan efektif dibandingkan menerapkan arsitektur *Stacking Classifier* yang kompleks tanpa memberikan keuntungan performa. Dari perspektif *deployment*, CatBoost layak untuk diterapkan karena memberikan keseimbangan optimal antara akurasi tinggi dan kecepatan eksekusi yang sangat penting bagi efisiensi komputasi pada aplikasi medis dengan respons cepat.

4. KESIMPULAN

Penerapan *K-Modes Clustering* dengan parameter optimal K=2 terbukti efektif dalam memetakan profil risiko pasien yang tersembunyi ke dalam kategori risiko rendah dan tinggi. Temuan ini memvalidasi hasil penelitian Bhatt et al. mengenai manfaat *K-Modes Clustering* ke dalam algoritma *Stacking Classifier* pada data kategorikal. Namun, integrasi fitur klaster ke dalam algoritma *Stacking Classifier* (akurasi 87,99% dan ROC-AUC 95,89%) tidak memberikan keuntungan kinerja dibandingkan model *single classifier*. Hasil pengujian menunjukkan dominasi model berbasis *Boosting*, di mana CatBoost mencatat performa tertinggi dengan *accuracy* 88,03% dan ROC-AUC 95,90%, mengungguli model *Stacking Classifier* secara keseluruhan. Meskipun akurasinya di bawah Sultan et al. (91%), penelitian ini memiliki keunggulan validitas dengan menjaga integritas data asli tanpa data sintetik (SMOTE) dan menerapkan *Stratified 5-Fold Cross-Validation* yang lebih ketat untuk menghasilkan estimasi performa yang *robust* dan tidak bias. Temuan ini membuktikan bahwa kapasitas algoritma berbasis *Boosting* sudah sangat memadai untuk menangkap pola data kompleks. Dengan adanya penambahan lapisan *ensemble* cenderung bersifat *redundant* dan meningkatkan beban komputasi tanpa memberikan peningkatan performa yang sepadan. Hal ini diperkuat oleh *trade-off* efisiensi waktu, di mana *Stacking Classifier* memerlukan waktu komputasi 560 kali lipat lebih lama (7587,7686 detik) dibandingkan CatBoost (13,4635 detik). Dengan demikian, penelitian ini menyimpulkan bahwa penggunaan model *single classifier* yang teroptimasi merupakan pendekatan yang jauh lebih baik dan efektif dibandingkan dengan algoritma *Stacking Classifier* yang

kompleks untuk kasus prediksi risiko kardiovaskular. CatBoost menjadi pilihan yang lebih rasional untuk *deployment* dengan memberikan *trade-off* terbaik antara performa model dengan efisiensi komputasi yang unggul, menjadi solusi ideal untuk sistem deteksi yang memprioritaskan keseimbangan optimal antara kecepatan dan ketepatan prediksi.

REFERENCES

- [1] WHO, "Cardiovascular diseases (CVDs)," www.who.int. [Daring]. Tersedia pada: [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
- [2] Institute for Health Metrics and Evaluation, "Global Burden of Disease Study 2023 (GBD 2023) Cause-Specific Mortality 1990-2023," Global Health Data Exchange.
- [3] D. Kasartzian, "Transforming Cardiovascular Risk Prediction: A Review of Machine Learning and Artificial Intelligence Innovations," 2025.
- [4] M. R. Sajid *et al.*, "Development of Nonlaboratory-Based Risk Prediction Models for Cardiovascular Diseases Using Conventional and Machine Learning Approaches," 2021.
- [5] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, dan N. Qureshi, "Can machine-learning improve cardiovascular risk prediction using routine clinical data?," vol. 355, hal. 1–14, 2017.
- [6] M. Imran, S. A. Ramay, dan T. Abbas, "Ensemble-Based Machine Learning for Early Detection and Risk Prediction of Cardiovascular Diseases," vol. 31, no. 9, hal. 3469–3486, 2024, doi: 10.53555/rb215y02.
- [7] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, hal. 241–259, 1992, doi: 10.1016/S0893-6080(05)80023-1.
- [8] J. T. Hancock dan T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *J. Big Data*, 2020, doi: 10.1186/s40537-020-00369-8.
- [9] J. M. Luna *et al.*, "Building more accurate decision trees with the additive tree," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 116, no. 40, hal. 19887–19893, 2019, doi: 10.1073/pnas.1816748116.
- [10] J. Liu, X. Dong, H. Zhao, dan Y. Tian, "Predictive Classifier for Cardiovascular Disease Based on Stacking Model Fusion," 2022.
- [11] T. Chen dan C. Guestrin, "XGBoost: A scalable tree boosting system," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-August, hal. 785–794, 2016, doi: 10.1145/2939672.2939785.
- [12] T. Bikku, "Multi-layered deep learning perceptron approach for health risk prediction," *J. Big Data*, 2020, doi: 10.1186/s40537-020-00316-7.
- [13] M. S. Patient, N. N. Albert, K. K. Pierre, M. M. M. Rostin, dan M. M. Eugene, "Artificial Learning Based On Kernel SVM For The Prediction of Cardiovascular Disease Hypertension," vol. 22, no. 1, hal. 28–34, 2025.
- [14] C. M. Bhatt, P. Patel, T. Ghetia, dan P. L. Mazzeo, "Effective Heart Disease Prediction Using Machine Learning Techniques," 2023.
- [15] A. Khan, A. Khan, M. M. Khan, dan K. Farid, "Cardiovascular and Diabetes Diseases Classification Using Ensemble Stacking Classifiers with SVM as a Meta Classifier," hal. 1–15, 2022.
- [16] S. Mohapatra, S. Maneesha, S. Mohanty, dan P. Kumar, "Healthcare Analytics A stacking classifiers model for detecting heart irregularities and predicting Cardiovascular Disease," *Healthc. Anal.*, vol. 3, no. June 2022, hal. 100133, 2023, doi: 10.1016/j.health.2022.100133.
- [17] S. Q. Sultan, N. Javaid, dan N. Alrajeh, "Machine Learning-Based Stacking Ensemble Model for Prediction of Heart Disease with Explainable AI and K-Fold Cross-Validation: A Symmetric Approach," no. Cvd, hal. 1–26, 2025.
- [18] V. Stodden, "The data science life cycle," *Commun. ACM*, vol. 63, no. 7, hal. 58–66, 2020, doi: 10.1145/3360646.
- [19] Irpanudin, Reka, R. N. Anggraeni, P. Pratama, A. Sujjada, dan A. Fergina, "Prediksi Penyakit Jantung Menggunakan Metode Deep Neural Network dengan Memanfaatkan Internet of Things," *J. Inf. dan Teknol.*, vol. 5, hal. 45–55, 2023, doi: 10.37034/jidt.v5i2.330.
- [20] E. F. L. Awalina dan W. I. Rahayu, "Optimalisasi Strategi Pemasaran dengan Segmentasi Pelanggan Menggunakan Penerapan K-Means Clustering pada Transaksi Online Retail," *J. Teknol. dan Inf.*, vol. 13, no. 2, hal. 122–137, 2023, doi: 10.34010/jati.v13i2.10090.
- [21] F. Lopez-jimenez, M. DI Cesare, J. Powis, N. Evans, K. Lara-breitinger, dan M. A. Rodriguez, "The Weight of Cardiovascular Diseases: Addressing the Global Cardiovascular Crisis Associated with Obesity," vol. 20, no. 1, 2025, doi: 10.5334/gh.1451.
- [22] D. DeMers dan D. Wachs, *Physiology, Mean Arterial Pressure*. StatPearls Publishing LLC, 2023.
- [23] P. W. F. Wilson, R. B. D. Agostino, D. Levy, A. M. Belanger, H. Silbershatz, dan W. B. Kannel, "Prediction of Coronary Heart Disease Using Risk Factor Categories," 1998, doi: 10.1161/01.CIR.97.18.1837.
- [24] World Health Organization, "Obesity: Preventing and Managing The Global Epidemic: Report of a WHO Consultation," 2000.
- [25] Z. Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values," vol. 304, hal. 283–304, 1998.
- [26] J. Han, M. Kamber, dan J. Pei, *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2012.
- [27] M. Kuhn dan K. Johnson, *Applied Predictive Modeling*. 2013.
- [28] A. Nurmasani dan Y. Pristyanto, "Algoritme Stacking Untuk Klasifikasi Penyakit Jantung Pada Dataset Imbalanced Class," *Pseudocode*, vol. 8, no. 1, hal. 21–26, 2021, doi: 10.33369/pseudocode.8.1.21-26.
- [29] F. S. Nahm, "Receiver operating characteristic curve: overview and practical use for clinicians," *Korean J. Anesthesiol.*, vol. 75, no. 1, hal. 25–36, 2022, doi: 10.4097/kja.21209.