

Penerapan Teknik *Ensemble Learning* untuk Deteksi Dini Penyakit Jantung Menggunakan Metode *Voting Classifier* dan *Stacking Classifier*

^{1*}Mutiara Romana Kusuma, ²Antonius Angga Kurniawan

¹Fakultas Teknik dan Informatika, Universitas Nusa Megarkencana, ²Fakultas Teknologi Industri, Universitas Gunadarma

¹Jl. A.M. Sangaji No.49, Cokrodiningratan Kecamatan. Jetis, Kota Yogyakarta Daerah Istimewa Yogyakarta, ²Jl. Margonda Raya No, 100 Depok, Jawa Barat

¹mutiararomana@unmeke.ac.id, ²anggaku@staff.gunadarma.ac.id

Abstract

Heart disease can be detected early by identifying risk factors examined in the Framingham Heart Study. This research aims to apply machine learning models for early detection of heart disease using the dataset. The stages include data preprocessing, training and testing of individual models, as well as the implementation of ensemble models. Three algorithms were employed: Random Forest (RF), Logistic Regression (LR), and K-Nearest Neighbor (KNN). The evaluation results show that RF achieved the best performance with an accuracy of 0.91, along with consistently high precision, recall, and F1-score across both classes (≥ 0.89). LR obtained the lowest accuracy of 0.67, with balanced but relatively low precision, recall, and F1-score (around 0.65–0.69). KNN reached an accuracy of 0.81, with very high precision in class 0 (0.94) and higher recall in class 1 (0.95). To enhance performance, Voting Classifier and Stacking Classifier were applied. The results indicate that Stacking provided the best outcome with an accuracy of 0.92, precision 0.91–0.94, recall 0.90–0.95, and F1-score 0.92–0.93, while Voting only achieved an accuracy of 0.86, with precision, recall, and F1-score in the range of 0.82–0.87. This difference occurred because Voting is more suitable for combining models with balanced performance, while in this study there was a significant performance gap between RF and LR.

Keywords: ensemble learning, framingham heart study, heart disease, stacking classifier, voting classifier

Abstrak

Penyakit jantung dapat dideteksi secara dini melalui identifikasi faktor risiko yang diteliti dalam *Framingham Heart Study*. Penelitian ini bertujuan menerapkan model *machine learning* untuk deteksi dini penyakit jantung menggunakan data tersebut. Tahapan penelitian meliputi pra-pemrosesan data, pelatihan dan pengujian model individu, serta penerapan model *ensemble*. Tiga algoritma digunakan, yaitu *Random Forest* (RF), *Logistic Regression* (LR), dan *K-Nearest Neighbor* (KNN). Hasil evaluasi menunjukkan bahwa RF memberikan performa terbaik dengan akurasi 0,91, serta *precision*, *recall*, dan *F1-score* konsisten tinggi pada kedua kelas ($\geq 0,89$). LR memiliki akurasi terendah sebesar 0,67, dengan *precision*, *recall*, dan *F1-score* seimbang namun rendah (sekitar 0,65–0,69). KNN mencapai akurasi 0,81, dengan *precision* tinggi pada kelas 0 (0,94) dan *recall* lebih baik pada kelas 1 (0,95). Untuk meningkatkan performa, digunakan metode *ensemble voting classifier* dan *stacking classifier*. Hasil menunjukkan bahwa *stacking* memberikan performa tertinggi dengan akurasi 0,92, *precision* 0,91–0,94, *recall* 0,90–0,95, dan *F1-score* 0,92–0,93, sedangkan *voting* hanya mencapai akurasi 0,86 dengan *precision*, *recall*, dan *F1-score* pada kisaran 0,82–0,87. Perbedaan ini terjadi karena *voting* lebih sesuai untuk menggabungkan model dengan performa seimbang, sementara dalam penelitian ini terdapat selisih performa signifikan antara RF dan LR.

Kata Kunci: *ensemble learning*, *framingham heart study*, penyakit jantung, *stacking classifier*, *voting classifier*

1. Pendahuluan

Salah satu penyakit yang paling mematikan di dunia adalah penyakit jantung. Menurut data yang diterbitkan oleh Organisasi Kesehatan Dunia (WHO) pada tahun 2021, sekitar 17,9 juta kematian global disebabkan oleh penyakit tersebut. Jumlah ini mencakup sekitar sepertiga dari seluruh kematian yang terjadi di dunia pada saat itu. Akan tetapi, bukan tidak mungkin untuk mencegah kematian akibat penyakit jantung. Hal tersebut dapat dilakukan dengan memperbaiki pola hidup dan melakukan pengobatan berdasarkan konseling. Oleh karena itu, penting untuk melakukan deteksi dini terhadap penyakit jantung, termasuk penyakit jantung koroner, agar dapat segera dilakukan penanganan yang tepat [1, 2].

Salah satu upaya yang dapat dilakukan untuk melakukan deteksi dini terhadap penyakit jantung koroner adalah dengan mengidentifikasi faktor risiko yang dapat menjadi penyebabnya [3]. Beberapa penelitian telah dilakukan mengenai penyebab penyakit jantung koroner, salah satu yang memiliki dampak besar adalah *Framingham Heart Study*. Penelitian ini telah dilakukan sejak tahun 1948. Salah satu hasil penelitian tersebut adalah laporan mengenai faktor risiko penyakit jantung koroner yang terdiri dari tingkat kebiasaan olahraga fisik, merokok, diabetes, kadar kolesterol darah, dan obesitas. Faktor risiko ini kemudian digunakan untuk menghitung skor risiko dalam jangka 10 hingga 30 tahun terhadap perkembangan penyakit jantung koroner yang disebut *Framingham Risk Score* (FRS) [4]. Hasil penelitian ini dibuka secara umum agar dapat digunakan dalam penelitian-penelitian berikutnya.

Beberapa penelitian telah dilakukan untuk menghitung risiko penyakit jantung koroner pada kelompok tertentu dengan menggunakan FRS, di antaranya adalah penelitian yang dilakukan pada staf di salah satu universitas di Mesir. Hasil penelitian tersebut menyatakan bahwa 28% sampel penelitian memiliki risiko tinggi terhadap penyakit jantung koroner untuk 10 tahun ke depan [5]. Penelitian serupa juga telah dilakukan di Indonesia, di antaranya adalah penelitian yang dilakukan di salah satu rumah sakit di Cirebon yang menyatakan bahwa terdapat sekitar 7% sampel yang memiliki risiko penyakit jantung koroner tinggi berdasarkan FRS [6]. Penelitian lain dilakukan oleh Anharudin dan Tejamaya pada para pekerja di suatu perusahaan. Hasilnya menunjukkan bahwa sekitar 11% sampel memiliki risiko tinggi berdasarkan perhitungan dengan FRS [7]. Pada penelitian-penelitian tersebut, perhitungan dilakukan secara manual, sehingga masih memiliki rentang kesalahan yang cukup tinggi dan akan lebih sulit jika menggunakan data dalam jumlah besar.

Penelitian-penelitian lain telah dikembangkan terhadap data penyakit jantung koroner dengan menggunakan *machine learning* untuk mendapatkan akurasi yang lebih tepat. Salah satunya adalah penelitian yang dilakukan oleh Wibisono dan Fahrurrozi. Sejumlah 300 data pasien jantung koroner dengan 14 variabel digunakan untuk proses klasifikasi dengan beberapa algoritma *machine learning*. Hasil penelitian tersebut menyatakan bahwa algoritma *Random Forest* memiliki performa lebih baik dibanding algoritma lain dengan nilai akurasi sebesar 86,7% [8]. Hassan dkk. juga melakukan penelitian serupa menggunakan 303 data penyakit jantung dengan 12 atribut. Kemudian dilakukan pengklasifikasian dengan sebelas algoritma *machine learning* untuk memprediksi adanya penyakit jantung. Algoritma yang memiliki performa terbaik adalah algoritma *Random Forest* dengan nilai akurasi sebesar 96,3% diikuti dengan *Gradient Boosted Decision Tree* yang nilai akurasinya adalah 95,8% [9].

Data dari hasil penelitian *Framingham Heart Study* juga telah digunakan untuk mengembangkan sistem atau model yang dapat memprediksi penyakit jantung koroner dengan menggunakan berbagai algoritma *machine learning*. Nurrohman dan Kurniawan menggunakan data tersebut dan kemudian melakukan prediksi dengan algoritma *Random Forest* dan *Decision Tree*. Hasilnya menyatakan bahwa kedua algoritma tersebut memiliki performa yang baik, dengan akurasi *Random Forest* adalah 97,6% dan *Decision Tree* adalah

91% [10]. Krishnani dkk. juga melakukan penelitian dengan data yang sama menggunakan algoritma Random Forest, Decision Trees, and K-Nearest Neighbors untuk melakukan prediksi, kemudian dilakukan perbandingan hasil prediksi. Hasil penelitian menunjukkan bahwa ketiga algoritma tersebut memiliki performa baik, dengan nilai akurasi masing-masing sebesar 96.8%, 92.7%, and 92.89% [11]. Suhatrik dkk. melakukan penelitian dengan data dari Framingham Heart Study untuk melakukan prediksi risiko penyakit jantung koroner dan membandingkan tujuh algoritma *machine learning* berdasarkan performanya. Hasilnya menunjukkan bahwa algoritma *Random Forest* memiliki performa lebih baik dengan nilai akurasi sebesar 85% diikuti oleh *Logistic Regression* dengan nilai akurasi sebesar 84% [12].

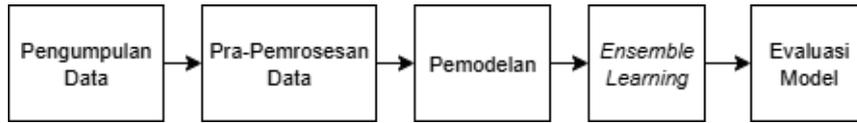
Salah satu teknik yang dapat dilakukan untuk meningkatkan akurasi adalah teknik *ensemble*. Teknik ini menggabungkan beberapa model dengan beberapa algoritma sehingga perubahan prediksi model saat dilatih dengan data yang berbeda (varians) dapat berkurang. Yawale dkk. melakukan penelitian dengan 303 data penyakit jantung koroner untuk kemudian diprediksi dengan menggunakan beberapa teknik *ensemble*. Hasil penelitian menunjukkan bahwa teknik *bagging* dengan *Random Forest* menghasilkan performa yang paling baik, yakni dengan akurasi sebesar 99,3%. Teknik *soft voting*, *stacking*, dan XGBoost juga menunjukkan performa yang cukup baik dengan nilai akurasi yang sama, meskipun nilai performa lainnya lebih rendah [13]. Shorewala juga melakukan penelitian serupa dengan menggunakan *dataset* yang berbeda, yakni sekitar 70.000 data pasien penyakit jantung koroner. Hasil penelitiannya menunjukkan bahwa teknik *stacking* adalah teknik *ensemble* yang paling efektif dan dapat menghasilkan nilai akurasi sebesar 75,1%. Teknik *bagging* juga menunjukkan performa yang cukup baik dengan peningkatan akurasi sekitar 1,9% lebih tinggi dibandingkan dengan model tanpa teknik *ensemble* [14].

Beberapa penelitian sebelumnya telah dilakukan untuk memprediksi risiko penyakit jantung koroner. Namun perhitungan pada penelitian-penelitian tersebut masih dilakukan secara manual [5 - 7], sehingga jika data yang digunakan dalam jumlah besar, akan lebih sulit dan memiliki rentan kesalahan yang tinggi. Penelitian lain dengan pendekatan *machine learning* telah dilakukan, namun data yang digunakan masih relatif sedikit, yakni hanya sekitar 300 data [8, 9, 13]. *Dataset* dengan jumlah yang lebih banyak telah digunakan pada beberapa penelitian, namun model yang dibangun belum menerapkan teknik *ensemble* [10, 11, 12]. Penelitian yang dilakukan Shorewala telah menggunakan teknik *ensemble* untuk *dataset* yang relatif cukup besar, akan tetapi nilai akurasi terbaiknya masih di bawah 80% [14].

Penelitian ini menggunakan data dari Framingham Heart Study untuk melakukan prediksi risiko penyakit jantung koroner dengan menggunakan beberapa algoritma klasifikasi. Berdasarkan penelitian-penelitian sebelumnya, algoritma *Random Forest*, *K-Nearest Neighbor*, dan *Logistic Regression* memiliki performa yang cukup baik, yakni dengan nilai akurasi 84%-99%, sehingga penelitian ini akan menggunakan ketiga algoritma tersebut untuk melakukan prediksi. Teknik *ensemble* seperti *voting* dan *stacking* digunakan terhadap model yang telah dikembangkan untuk mendapatkan hasil performa yang lebih baik.

2. Metode Penelitian

Penelitian ini terdiri dari beberapa tahapan. Tahap pertama adalah pengumpulan data yang akan digunakan dalam implementasi. Selanjutnya, dilakukan pra-pemrosesan terhadap data tersebut guna memastikan bahwa data mentah tersebut sesuai dengan kebutuhan model *machine learning* yang akan diterapkan. Data yang telah diolah kemudian diproses dengan menggunakan beberapa algoritma untuk menghasilkan prediksi. Teknik *ensemble learning* kemudian dilakukan untuk meningkatkan hasil pemodelan. Setiap model yang telah dibangun kemudian dievaluasi untuk mendapatkan model dengan performa terbaik. Tahapan-tahapan tersebut tertuang pada Gambar 1.



Gambar 1. Tahapan Penelitian

Tabel 1. Penjelasan Atribut *Dataset*

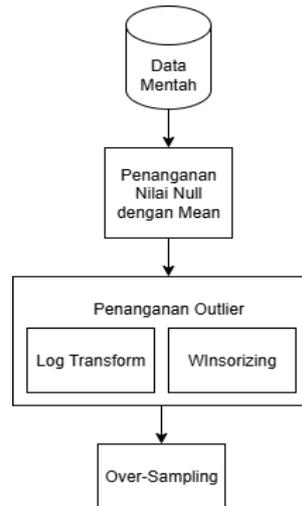
| No | Nama Atribut | Keterangan |
|----|------------------------|--|
| 1 | <i>Male</i> | Jenis kelamin (1 laki-laki, 0 perempuan) |
| 2 | <i>Age</i> | Usia (dalam tahun) |
| 3 | <i>Education</i> | Pendidikan terakhir |
| 4 | <i>currentSmoker</i> | Perokok aktif atau bukan (1 aktif, 0 bukan) |
| 5 | <i>cigsPerDay</i> | Jumlah rokok yang dikonsumsi per hari |
| 6 | <i>BPMeds</i> | Status seseorang mengonsumsi obat tekanan darah (1 mengonsumsi, 0 tidak) |
| 7 | <i>prevalentStroke</i> | Pernah menderita stroke (1 pernah, 0 tidak) |
| 8 | <i>prevalentHyp</i> | Pernah menderita hipertensi (1 pernah, 0 tidak) |
| 9 | <i>Diabetes</i> | Penderita diabetes (2 ya, 0 tidak) |
| 10 | <i>totChol</i> | Total kolesterol (mm/dl) |
| 11 | <i>sysBP</i> | Tekanan darah sistolik (tidak lebih dari 140 mmHg) |
| 12 | <i>diaBP</i> | Tekanan darah diastolik (tidak lebih dari 90 mmHg) |
| 13 | <i>BMI</i> | Indeks masa tubuh (kg/m^2) |
| 14 | <i>Heartrate</i> | Denyut jantung (normalnya 60 – 100 denyut per menit) |
| 15 | <i>Glucose</i> | Kadar glukosa dalam darah (mg/dl) |
| 16 | <i>tenYearCHD</i> | Risiko seseorang terkena penyakit jantung (1 beresiko, 0 tidak) |

2.1 Pengumpulan Data

Penelitian ini menggunakan data dari Framingham Heart Study yang telah dipublikasikan dan dapat diakses pada platform Kaggle [15]. Data tersebut dikumpulkan dari 4.240 orang yang berbeda dengan 15 kolom atribut dan satu kolom yang merupakan prediksi risiko penyakit jantung dalam 10 tahun ke depan. Penjelasan mengenai atribut-atribut tersebut terlihat seperti pada Tabel 1. *Dataset* ini dapat diunduh dengan format .csv sehingga dapat digunakan dalam tahapan berikutnya.

2.2 Pra-pemrosesan Data

Proses ini dilakukan dengan tujuan untuk membuat data yang telah didapatkan sebelumnya menjadi lebih berkualitas dan lebih siap digunakan dalam pemodelan *machine learning* [16]. Salah satu tahapan pra-pemrosesan yang dapat dilakukan untuk meningkatkan kualitas data adalah pembersihan data. Tujuan dari proses ini adalah untuk membersihkan *noise* pada data, seperti mengurangi ketidakkonsistenan, duplikasi maupun *outlier* yang masih terdapat pada data mentah [17]. Pada penelitian ini dilakukan pembersihan data dengan menangani nilai kosong (*missing values*) serta memperbaiki *outlier* (nilai ekstrim). Nilai kosong yang ada pada setiap atribut diisi dengan nilai *mean* (rata-rata) dari keseluruhan nilai atribut tersebut [18]. *Outlier* yang ada pada data kemudian ditangani dengan menggunakan teknik *log transform* dan *winsorizing*. *Log transform* dilakukan untuk mengurangi *skewness* sehingga dapat membuat distribusi data menjadi lebih simetris [19]. *Winsorizing* dapat membatasi nilai ekstrim yang terdapat pada data sehingga kemungkinan *outlier* dapat dikurangi [20].



Gambar 2. Tahapan Pra-Pemrosesan

Dataset yang didapatkan sebelumnya memiliki perbandingan jumlah data yang berbeda secara signifikan antara kelas prediksi positif dan negatif, sehingga pada penelitian ini dilakukan teknik untuk menyeimbangkan jumlah data antara kedua kelas tersebut, yakni *Synthetic Minority Oversampling Technique* (SMOTE). SMOTE mampu menghasilkan data sintetis baru dari kelas minoritas tanpa sekadar melakukan replikasi data yang sudah ada. Prosesnya dimulai dengan memilih setiap sampel pada kelas minoritas sebagai titik acuan, kemudian secara berurutan dibuat sejumlah sampel tambahan sebanyak n kali. Setelah itu, dilakukan interpolasi linier antara sampel acuan dengan sampel tambahan tersebut untuk membentuk n data sintetis baru. Dengan pendekatan ini, algoritma SMOTE dapat mengatasi masalah *overfitting* yang biasanya muncul pada metode *oversampling* acak yang hanya menggandakan data secara langsung [21]. Tahapan-tahapan pra-pemrosesan ini digambarkan pada Gambar 2.

2.3 Pemodelan

Tahap pemodelan merupakan tahap di mana data yang sebelumnya telah diolah kemudian diproses dengan menggunakan algoritma untuk menghasilkan prediksi berdasarkan data tersebut. Sebelum data digunakan ke dalam pemodelan, data yang digunakan dilakukan pemisahan atau *splitting data* terlebih dahulu menjadi dua bagian, yaitu data latih dan data uji dengan porsi 80% untuk data latih dan 20% untuk data uji. Porsi pemisahan tersebut dipilih karena berdasarkan penelitian terdahulu porsi 80% dan 20% adalah pembagian yang paling sering dilakukan dan memperoleh hasil yang lebih optimal [22].

Penelitian ini menerapkan beberapa algoritma untuk membangun model menggunakan data yang sebelumnya telah diproses. Algoritma-algoritma yang diterapkan pada penelitian ini adalah sebagai berikut.

a. *Random Forest*

Algoritma ini merupakan penggabungan dari beberapa algoritma *Decision Tree* yang telah dilatih sebelumnya. Kelebihan *Random Forest* di antaranya adalah dapat menghasilkan akurasi yang lebih baik meskipun terdapat data kosong serta memiliki seleksi fitur yang mampu memilih fitur terbaik sehingga performa model dapat lebih meningkat [23]. Hal ini sesuai dengan karakteristik *dataset* Framingham Heart Study yang memiliki banyak variabel risiko (seperti tekanan darah, kolesterol, gula darah, kebiasaan merokok) dengan tingkat

kepentingan yang berbeda-beda, sehingga *Random Forest* dapat membantu mengidentifikasi variabel yang paling relevan untuk prediksi penyakit jantung.

b. *Logistic Regression*

Logistic Regression menggunakan analisis matematika untuk mengetahui korelasi antara beberapa variabel bebas dengan satu variabel terikat yang bersifat biner. Algoritma ini termasuk algoritma yang efisien serta dapat menjelaskan hubungan antara variabel terikat yang menjadi target dengan variabel lainnya [24]. *Dataset* Framingham Heart Study menggunakan label biner (berisiko penyakit jantung = 1, tidak berisiko = 0). *Logistic Regression* sangat cocok untuk menganalisis dan memprediksi faktor risiko penyakit jantung serta memberikan interpretasi yang jelas mengenai pengaruh tiap variabel.

c. *K-Nearest Neighbor (KNN)*

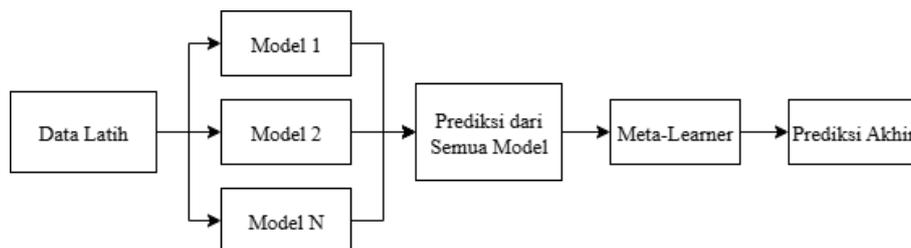
Algoritma KNN dapat mengklasifikasikan data baru berdasarkan data dari tetangga terdekat. Algoritma ini memiliki performa yang baik dalam mengklasifikasikan data dalam jumlah besar meskipun data yang dilatih memiliki banyak *noise* [25]. Pemilihan algoritma KNN didasarkan pada keunggulannya dalam mengklasifikasikan data melalui perhitungan jarak antar titik secara efisien, disertai dengan proses pelatihan yang tidak kompleks dan performa akurasi yang baik pada *dataset* medis dengan variasi fitur yang beragam [26].

2.4 *Ensemble Learning*

Teknik *ensemble learning* adalah teknik yang mengintegrasikan beberapa algoritma *machine learning* dengan tujuan untuk meningkatkan akurasi serta konsistensi dari prediksi yang dihasilkan dan mengurangi kesalahan generalisasi [27]. Terdapat beberapa teknik *ensemble learning*, yaitu *bagging*, *boosting*, *stacking*, *voting*, dan *blending*. Pada penelitian ini dilakukan menggunakan dua teknik, yaitu *stacking* dan *voting* untuk melakukan klasifikasi. *Voting classifier* dipilih karena sangat mudah diimplementasikan untuk menggabungkan beberapa model, sedangkan *stacking classifier* lebih fleksibel, namun sedikit lebih kompleks daripada *voting*. Oleh karena itu, pada penelitian ini juga bertujuan untuk melihat perbandingan hasil kedua model tersebut. Selain itu, teknik *ensemble learning* seperti *voting* dan *stacking classifier* merupakan teknik yang efektif untuk meningkatkan kekokohan model dan akurasi model terhadap tantangan terkait kemampuan generalisasi karena *overfitting* pada model *machine learning* individu [28].

a. *Stacking Classifier*

Stacking adalah metode *ensemble* yang menggabungkan beberapa model dasar (*base-level classifiers*) dengan sebuah *meta-learner* (disebut juga *meta-classifier*). Model-model dasar dilatih menggunakan *dataset* asli, kemudian hasil prediksinya digunakan sebagai *input* bagi *meta-learner* yang bertugas menghasilkan prediksi akhir. Pendekatan ini memungkinkan penggunaan model *machine learning* yang berbeda dalam satu kerangka kerja sehingga meningkatkan performa prediksi [29]. Gambar 3 menunjukkan kerangka kerja dari *stacking*.



Gambar 3. Kerangka Kerja Pendekatan *Stacking*

Model-model dilatih secara paralel pada *dataset* yang sama, kemudian hasil prediksi dari setiap model dasar tersebut dijadikan sebagai *input* untuk model tingkat kedua yang disebut *meta-learner*. *Meta-learner* ini bertugas mempelajari pola dari prediksi model-model dasar untuk menghasilkan prediksi akhir yang lebih akurat. Pendekatan ini memanfaatkan keanekaragaman model dasar guna menangkap pola data yang kompleks, serta mengurangi kesalahan generalisasi yang mungkin terjadi pada model tunggal. Kombinasi model yang berbeda struktur seperti pada penelitian sangat ideal untuk *stacking*, karena menghasilkan prediksi yang beragam untuk dipelajari *meta-learner*.

Stacking classifier menggunakan beberapa model dasar untuk kemudian dipelajari oleh *meta-learner*. Misalkan terdapat sejumlah M model dasar, yakni $h_1 h_2 \dots h_M$ serta data latih (x_i, y_i) dengan $i = 1, 2, \dots, N$, setiap model h_j menghasilkan prediksi probabilitas atau label yang diuraikan pada Persamaan (1).

$$Z_{ij} = h_j(x_i) \tag{1}$$

Vektor prediksi untuk setiap x_i kemudian bisa didapatkan berdasarkan Persamaan (2).

$$Z_i = (z_{i1}, z_{i2}, \dots, z_{iM}) \tag{2}$$

Meta-learner H kemudian dilatih menggunakan *dataset* baru, sehingga fungsi prediksi *stacking* diuraikan pada Persamaan (3) [30, 31].

$$\hat{y} = H(Z) = H(h_1(x), h_2(x), \dots, h_M(x)) \tag{3}$$

b. *Voting Classifier*

Voting classifier adalah metode *ensemble* dalam *machine learning* yang menggabungkan prediksi dari beberapa model independen untuk menghasilkan keputusan akhir yang lebih akurat. Kerangka kerja *voting classifier* terlihat pada Gambar 4. Pendekatan ini menerapkan prinsip “*wisdom of the crowd*”, yaitu memanfaatkan kekuatan kolektif dari berbagai model dibandingkan hanya mengandalkan satu model. Jika satu algoritma salah mengklasifikasikan data, keputusan akhir masih bisa benar selama mayoritas model lain memberikan prediksi yang tepat. Terdapat tiga jenis *voting*: *hard voting*, di mana prediksi akhir ditentukan berdasarkan suara terbanyak; *soft voting*, yang menggunakan rata-rata probabilitas dari tiap model; dan *weighted voting*, yang memberikan bobot berbeda pada setiap model berdasarkan performanya [29]. Persamaan matematis untuk proses *hard voting* diuraikan pada Persamaan (4).

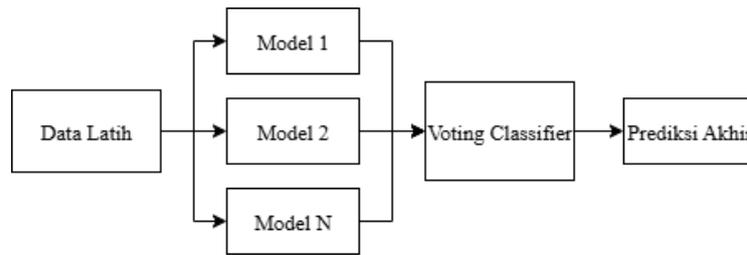
$$\hat{y} = \operatorname{argmax}_{c \in C} \sum_{j=1}^M 1(\hat{y}_j = c) \tag{4}$$

dengan \hat{y} menyatakan kelas hasil prediksi akhir, C adalah himpunan semua kelas (misalnya $\{0,1\}$), M menunjukkan jumlah model dasar, \hat{y}_j merupakan hasil prediksi model ke- j , dan $1(\hat{y}_j = c)$ adalah fungsi indikator, bernilai 1 jika model ke- j memprediksi kelas c , serta 0 jika tidak. Persamaan (4) menyatakan bahwa *hard voting* akan memilih kelas yang paling banyak dipilih oleh model.

Persamaan matematis untuk proses *soft voting* diuraikan pada Persamaan (5). *Soft voting* biasanya lebih baik jika semua model bisa memberikan probabilitas prediksi [32].

$$\hat{y} = \operatorname{argmax}_{c \in C} \sum_{j=1}^M w_j \cdot P_j(c|x) \tag{5}$$

dengan $P_j(c|x)$ menyatakan probabilitas bahwa model ke- j memprediksi kelas c untuk *input* x , w_j menyatakan bobot untuk model ke- j (*default*: semua sama $w_j = \frac{1}{M}$), dan \hat{y} adalah kelas dengan probabilitas rata-rata terbesar.



Gambar 4. Kerangka Kerja *Voting Classifier*

Penggabungan keunggulan dari berbagai model membuat *voting classifier* mampu meningkatkan akurasi, stabilitas, dan ketahanan model terhadap *overfitting*. Pendekatan ini sangat bermanfaat saat model-model dasar memiliki karakteristik dan kelemahan yang berbeda, seperti ketiga algoritma yang digunakan pada penelitian ini, sehingga dapat saling melengkapi. Selain itu, *voting classifier* juga mendukung interpretabilitas dengan memungkinkan analisis kontribusi masing-masing model terhadap keputusan akhir, menjadikannya alat yang efektif dalam berbagai tugas prediksi *machine learning*.

3. Hasil dan Pembahasan

Data dari Framingham Heart Study yang telah dikumpulkan sebelumnya kemudian diolah menggunakan perangkat lunak Jupyter Notebook dengan bahasa pemrograman Python. Hasil dari tahapan-tahapan yang dilakukan pada penelitian ini diuraikan sebagai berikut.

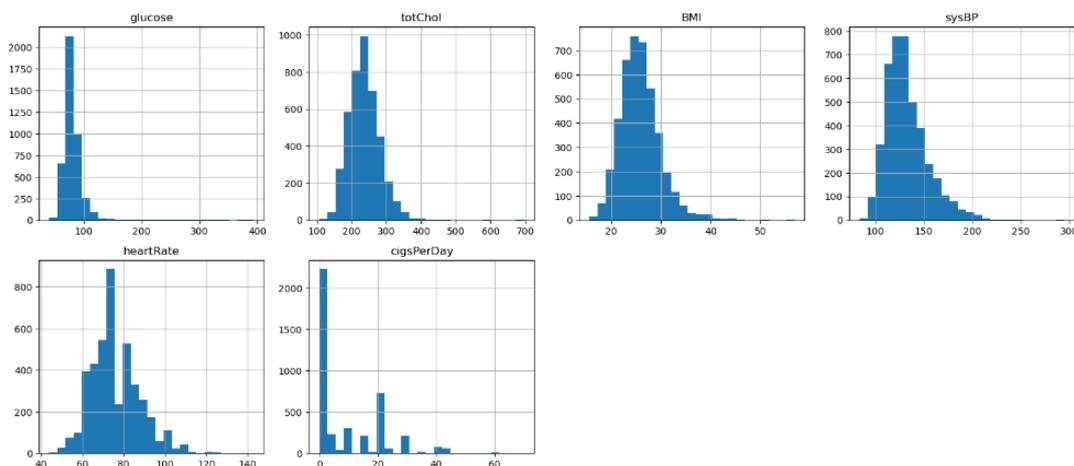
3.1 Hasil Pra-Pemrosesan

Beberapa kolom pada data mentah yang telah dikumpulkan masih memiliki nilai kosong (*null*). Kolom yang memiliki nilai *null* yakni kolom *education*, *cigsPerDay*, *BPMeds*, *totChol*, *BMI*, *heartrate*, dan *glucose* dengan jumlah baris kosong masing-masing adalah 105, 29, 53, 50, 19, 1, dan 388. Total baris dengan nilai *null* pada *dataset* yang digunakan adalah 582 baris, yakni sekitar 13,7% dari keseluruhan data. Penghapusan baris data yang memiliki nilai *null* dapat mengurangi representativitas data, terutama pada data medis seperti pada penelitian ini, di mana setiap sampel pasien sangat berharga. Pada penelitian ini untuk menghindari hal tersebut, digunakan nilai rata-rata (*mean*) dari keseluruhan data pada masing-masing kolom untuk mengisi nilai kosong tersebut sehingga baris yang memiliki kolom dengan nilai *null* tidak perlu dihapus. Data pada penelitian ini merupakan data numerik berskala kontinu, sehingga mengganti nilai yang hilang dengan rata-rata populasi masih dapat mewakili kecenderungan umum. Setelah diisi dengan nilai *mean*, tidak ada lagi kolom yang memiliki baris dengan nilai *null*. Perbandingan jumlah baris dari kolom-kolom dengan nilai *null* sebelum dan sesudah diisi dengan nilai *mean* terlihat pada Tabel 3.

Histogram kemudian ditampilkan untuk melihat distribusi data pada penelitian ini. Data dengan nilai nonbiner dikatakan memiliki distribusi yang baik jika bentuk histogramnya menyerupai lonceng, yakni jika data terkonsentrasi pada nilai rata-rata. Jika pada histogram terdapat nilai yang jauh dari distribusi data, maka dapat dikatakan bahwa data tersebut memiliki *outlier*. Gambar 5 menunjukkan bahwa distribusi data yang dikumpulkan sebelumnya masih kurang baik, di mana terdapat beberapa kolom nonbiner seperti *glucose*, *totChol*, *BMI*, *sysBP*, *heartRate*, dan *cigsPerDay* terdistribusi ke kiri dan terdapat nilai-nilai yang jauh dari pusat distribusinya atau dapat dikatakan bahwa terdapat *outlier* pada kolom-kolom tersebut.

Tabel 3. Perbandingan Jumlah Baris pada Kolom dengan Nilai Null
 a) Sebelum Diisi Nilai Mean
 b) Setelah Diisi Nilai Mean

| Nama Kolom | Jumlah Baris dengan Nilai Null | Nama Kolom | Jumlah Baris dengan Nilai Null |
|-------------------|--------------------------------|-------------------|--------------------------------|
| <i>education</i> | 105 | <i>education</i> | 0 |
| <i>cigsPerDay</i> | 29 | <i>cigsPerDay</i> | 0 |
| <i>BPMeds</i> | 53 | <i>BPMeds</i> | 0 |
| <i>totChol</i> | 50 | <i>totChol</i> | 0 |
| <i>BMI</i> | 19 | <i>BMI</i> | 0 |
| <i>heartrate</i> | 1 | <i>heartrate</i> | 0 |
| <i>glucose</i> | 388 | <i>glucose</i> | 0 |



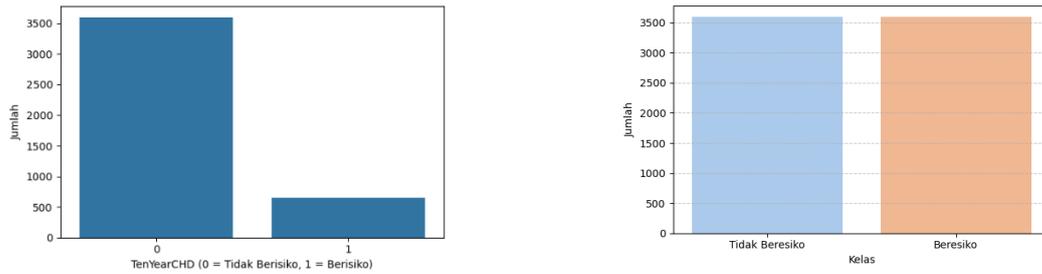
Gambar 5. Histogram Distribusi Data

Tabel 4. Perbandingan Jumlah Baris pada Kolom dengan *Outlier*
 a) Sebelum Dilakukan *Log Transform* dan *Winsorizing*
 b) Setelah Dilakukan *Log Transform* dan *Winsorizing*

| Nama Kolom | Jumlah Baris dengan <i>Outlier</i> | Persentase <i>Outlier</i> (%) | Nama Kolom | Jumlah Baris dengan <i>Outlier</i> | Persentase <i>Outlier</i> (%) |
|-------------------|------------------------------------|-------------------------------|-------------------|------------------------------------|-------------------------------|
| <i>glucose</i> | 262 | 6,18 | <i>glucose</i> | 239 | 5,64 |
| <i>totChol</i> | 126 | 2,97 | <i>totChol</i> | 126 | 2,97 |
| <i>BMI</i> | 97 | 2,29 | <i>BMI</i> | 78 | 1,84 |
| <i>sysBP</i> | 76 | 1,79 | <i>sysBP</i> | 73 | 1,72 |
| <i>heartrate</i> | 57 | 1,34 | <i>heartrate</i> | 35 | 0,82 |
| <i>cigsPerDay</i> | 12 | 0,28 | <i>cigsPerDay</i> | 0 | 0 |

Penanganan *outlier* kemudian dilakukan dengan teknik *log transform* dan *winsorizing*. Kedua teknik ini akan mengganti nilai *outlier* dengan nilai yang berbeda, sehingga dapat mengurangi jumlah nilai *outlier* pada *dataset*. Setelah dilakukan kedua teknik tersebut, jumlah *outlier* yang terdapat pada kolom *glucose*, *BMI*, *sysBP*, dan *heartRate* berkurang, serta sudah tidak ada lagi nilai *outlier* pada kolom *cigsPerDay*. Hal ini ditunjukkan pada Tabel 4.

Distribusi data kolom target, yakni prediksi resiko penyakit jantung untuk 10 tahun ke depan (kolom *tenYearCHD*) juga perlu diperhatikan. Gambar 6.a. menunjukkan bahwa data pada kolom target tidak seimbang, di mana terdapat 3.596 data yang diprediksi tidak beresiko, sedangkan yang diprediksi beresiko hanya terdapat 644 data.



a. Distribusi Kolom Target (*TenYearCHD*)

b. Distribusi Kolom Target Setelah *Oversampling*

Gambar 6. Distribusi Kolom Target

Data yang tidak seimbang dapat menyebabkan terjadinya *biased-learning*, atau kondisi di mana model hanya dapat mempelajari data kelas mayoritas dan tidak mempelajari data kelas minoritas. Hal tersebut dapat dibatasi dengan menggunakan teknik *oversampling* yakni dengan teknik *Synthetic Minority Oversampling Technique* (SMOTE). Teknik SMOTE tidak sekadar menduplikasi data minoritas yang ada, melainkan secara sintesis menghasilkan data baru dengan cara mengambil titik acuan dari kelas minoritas, kemudian membentuk sampel baru di antara titik tersebut dan tetangganya. Dengan demikian, data yang dihasilkan lebih bervariasi dibandingkan sekadar duplikasi sederhana. Setelah diterapkan SMOTE, distribusi data pada kolom target menjadi seimbang seperti terlihat pada Gambar 6.b.

3.2 Hasil Pemodelan

Pada bagian ini menampilkan model yang telah dibangun, di mana model yang dibangun terdiri dari *Random Forest*, *Logistic Regression*, *K-Nearest Neighbor*, serta model *ensemble learning* dengan teknik *voting classifier* dan *stacking classifier*.

a. *Random Forest*

Model *Random Forest* pada penelitian ini dibangun dengan jumlah pohon sebanyak 100, menggunakan indeks *Gini* sebagai ukuran kualitas pemisahan, dan pemilihan fitur maksimum berdasarkan akar jumlah total fitur pada setiap split. Kedalaman pohon tidak dibatasi sehingga dapat menyesuaikan kompleksitas data, dengan jumlah minimum sampel untuk pemisahan dan pembentukan daun masing-masing dua dan satu. Proses pelatihan dilakukan dengan teknik *bootstrap sampling*, sementara validasi *out-of-bag* serta pengaturan bobot kelas tidak digunakan. Dengan konfigurasi ini, *Random Forest* mampu memanfaatkan variasi antar pohon dan mekanisme agregasi ensemble untuk meningkatkan akurasi serta kemampuan generalisasi model.

b. *Logistic Regression*

Model *Logistic Regression* pada penelitian ini dibangun dengan batas iterasi maksimum 1000 untuk memastikan konvergensi, serta pengaturan *random state* agar hasil pelatihan dapat direplikasi secara konsisten. Algoritma ini menggunakan regularisasi L2 sebagai mekanisme pengendali kompleksitas model, dengan parameter regularisasi default ($C=1.0$) yang menjaga keseimbangan antara akurasi dan generalisasi. Proses optimasi dilakukan menggunakan *solver lbfgs* yang sesuai untuk dataset berukuran besar dan mendukung regularisasi L2. Selain itu, model menambahkan *intercept* untuk memperbaiki pemisahan kelas, serta secara otomatis memilih pendekatan *one-vs-rest* atau *multinomial* sesuai karakteristik data. Dengan konfigurasi ini, *Logistic Regression* membangun fungsi diskriminatif berbasis probabilitas

yang mampu memodelkan hubungan linier antara fitur prediktor dengan peluang terjadinya penyakit jantung, sekaligus mengendalikan risiko *overfitting* melalui regularisasi.

c. *K-Nearest Neighbor* (KNN)

Model KNN pada penelitian ini dibangun dengan jumlah tetangga sebanyak lima, sehingga prediksi kelas suatu sampel baru ditentukan berdasarkan mayoritas kelas dari lima sampel terdekat. Setiap tetangga memberikan kontribusi yang sama terhadap hasil prediksi (*uniform weights*), dengan jarak dihitung menggunakan metrik *Euclidean*. Pencarian tetangga dilakukan secara otomatis oleh pustaka untuk menyesuaikan dengan ukuran dan dimensi data, sementara parameter lainnya menggunakan nilai *default* yang tidak memengaruhi hasil prediksi secara langsung. Dengan konfigurasi ini, KNN bekerja sebagai algoritma non-parametrik yang sepenuhnya bergantung pada kedekatan data pelatihan di ruang fitur. Pendekatan ini efektif untuk dataset medis seperti *Framingham Heart Study* yang memiliki fitur heterogen, meskipun sensitif terhadap skala variabel dan keberadaan *outlier*.

d. *Voting Classifier*

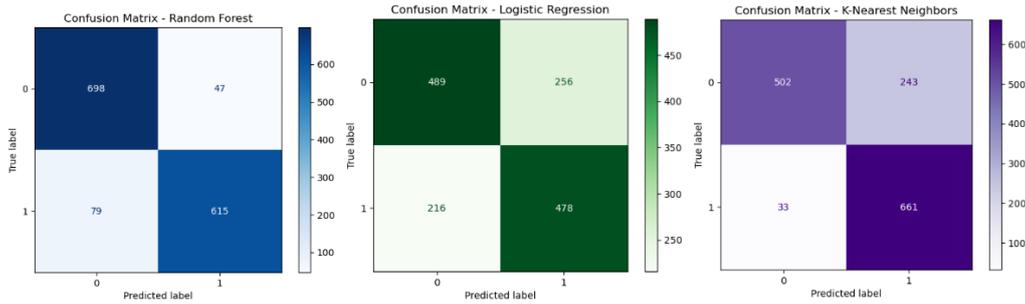
Voting Classifier pada penelitian ini dibangun dengan menggabungkan tiga model dasar yang telah dibangun sebelumnya, yaitu *Random Forest*, *Logistic Regression*, dan KNN. Metode *soft voting* digunakan, di mana keputusan akhir diperoleh dari rata-rata probabilitas prediksi masing-masing model, bukan hanya berdasarkan mayoritas suara. *Random Forest* berkontribusi dengan kekuatannya dalam menangani data kompleks dan variabel penting, *Logistic Regression* memberikan interpretasi probabilistik yang jelas untuk klasifikasi biner, sementara KNN menambahkan kemampuan berbasis kedekatan data lokal. Dengan memadukan ketiga algoritma ini, *Voting Classifier* diharapkan mampu menyeimbangkan bias dan varians, serta meningkatkan kemampuan generalisasi model dibandingkan penggunaan model tunggal.

e. *Stacking Classifier*

Stacking Classifier pada penelitian ini dibangun dengan tiga model dasar yang telah dibangun sebelumnya, yaitu *Random Forest*, *Logistic Regression*, dan KNN. Berbeda dengan *Voting*, hasil prediksi dari model dasar tidak langsung digunakan sebagai keputusan akhir, melainkan digabungkan dan dijadikan masukan bagi sebuah *meta-learner*. Pada penelitian ini, *Logistic Regression* dipilih sebagai *meta-learner* dengan batas iterasi maksimum 1000, karena kemampuannya dalam menghasilkan prediksi yang linier, stabil, dan interpretatif. Proses pelatihan dilakukan menggunakan validasi silang 5 lipatan, sehingga *meta-learner* dapat dilatih secara lebih stabil dan mengurangi risiko *overfitting*. Dengan memanfaatkan kombinasi kekuatan model linier dan nonlinier pada *base learners* serta integrasi prediksi melalui *meta-learner*, *Stacking Classifier* mampu mempelajari pola kesalahan dari masing-masing model dasar dan menghasilkan sistem klasifikasi yang lebih *robust* serta akurat.

3.3 Hasil Evaluasi Model

Evaluasi terhadap model yang telah dibangun kemudian dilakukan untuk mengetahui model mana yang memiliki performa terbaik. *Confusion Matrix* digunakan untuk memvisualisasikan kinerja model. Kinerja model dianggap baik jika nilai pada kolom diagonal, yakni kolom *true positive* dan *true negative* lebih tinggi dari nilai *false positive* dan *false negative*. Pada Gambar 7 terlihat bahwa setiap model individu yang telah diujicobakan memiliki kinerja yang cukup baik. Nilai *false positive* tertinggi adalah 256 dan nilai *false negative* tertinggi adalah 216, yakni pada model dengan algoritma *Logistic Regression*



Gambar 7. Confusion Matrix Model Individu

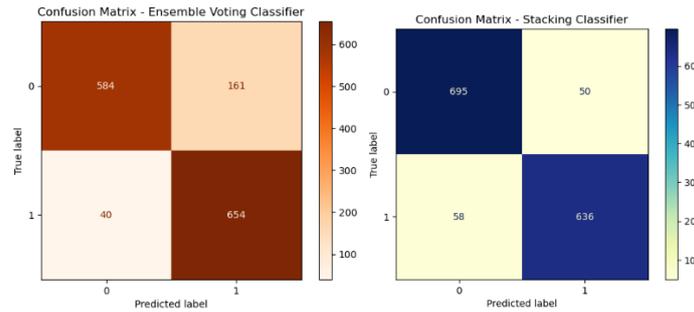
Tabel 5. Performa Model Individu

| Model | Akurasi | Precision | | Recall | | F1-Score | |
|--------------------------|---------|-----------|------|--------|------|----------|------|
| | | 0 | 1 | 0 | 1 | 0 | 1 |
| Random Forest (RF) | 0,91 | 0,90 | 0,93 | 0,94 | 0,89 | 0,92 | 0,91 |
| Logistic Regression (LR) | 0,67 | 0,69 | 0,65 | 0,66 | 0,69 | 0,67 | 0,67 |
| KNN | 0,81 | 0,94 | 0,73 | 0,67 | 0,95 | 0,78 | 0,83 |

Tabel 5 menampilkan hasil evaluasi performa tiga algoritma klasifikasi yang digunakan secara individual. Evaluasi dilakukan menggunakan metrik akurasi, precision, recall, dan F1-score pada masing-masing kelas (0 dan 1). Random Forest mencapai akurasi tertinggi sebesar 0,91, dengan keseimbangan performa pada kedua kelas. Nilai precision dan recall untuk Random Forest juga relatif tinggi pada kelas 0 maupun 1 (di atas 0,89), sehingga menghasilkan nilai F1-score yang konsisten. Hal ini menunjukkan bahwa RF mampu mengklasifikasikan data dengan baik pada kedua kelas, tanpa kecenderungan bias terhadap salah satu kelas. Model Logistic Regression memiliki akurasi terendah, yaitu 0,67. Precision, recall, dan F1-score untuk model ini pada kedua kelas hampir seimbang (sekitar 0,65–0,69), namun relatif rendah dibandingkan dua model lainnya. Hal ini menandakan bahwa LR kurang mampu menangkap pola data dengan baik, kemungkinan karena keterbatasannya sebagai model linier dalam menangani kompleksitas dataset. Model KNN mencapai akurasi 0,81, berada di antara RF dan LR. Model ini menunjukkan precision sangat tinggi pada kelas 0 (0,94), namun recall lebih tinggi pada kelas 1 (0,95). Performa F1-score KNN lebih baik pada kelas 1 (0,83) dibandingkan kelas 0 (0,78), yang menunjukkan KNN lebih efektif dalam mengidentifikasi kelas 1 dibandingkan kelas 0.

Ensemble learning kemudian diterapkan untuk meningkatkan akurasi. Gambar 8 menunjukkan *confusion matrix* untuk model *ensemble learning* dengan teknik *voting* dan *stacking*. Kedua model tersebut menunjukkan kinerja yang baik, di mana nilai *false positive* dan *false negative* tertinggi masing-masing hanya 161 dan 58.

Tabel 6 menunjukkan performa model *ensemble learning* dengan teknik *voting* dan *stacking*. Terlihat bahwa metode *ensemble learning* dengan pendekatan *Stacking* mampu memberikan performa terbaik dibandingkan dengan *Voting*. Model *Stacking* yang menggabungkan Random Forest, Logistic Regression, dan KNN mencapai akurasi sebesar 0,92, lebih tinggi dibandingkan *Voting* yang hanya sebesar 0,86. Selain itu, nilai precision, recall, dan F1-score pada *Stacking* juga lebih unggul dan seimbang di kedua kelas. Precision *Stacking* mencapai 0,91 pada kelas 0 dan 0,94 pada kelas 1, sementara recall masing-masing sebesar 0,95 dan 0,90, menghasilkan F1-score tinggi yaitu 0,93 dan 0,92. Hal ini menunjukkan bahwa *Stacking* tidak hanya akurat dalam klasifikasi, tetapi juga konsisten dalam menyeimbangkan kesalahan positif palsu dan negatif palsu. Sebaliknya, *Voting* menghasilkan performa yang lebih rendah, dengan precision, recall, dan F1-score yang rata-rata berada di kisaran 0,82–0,87.



Gambar 8. Confusion Matrix Model dengan Ensemble Learning

Tabel 6. Performa Model Ensemble Learning

| Model | Akurasi | Precision | | Recall | | F1-Score | |
|--------------------------|---------|-----------|------|--------|------|----------|------|
| | | 0 | 1 | 0 | 1 | 0 | 1 |
| Stacking (RF + LR + KNN) | 0,92 | 0,91 | 0,94 | 0,95 | 0,90 | 0,93 | 0,92 |
| Voting (RF + LR + KNN) | 0,86 | 0,87 | 0,82 | 0,82 | 0,87 | 0,84 | 0,84 |

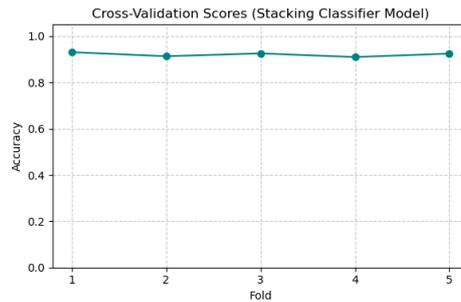
Performa model dengan teknik *voting* memiliki nilai yang lebih rendah karena teknik ini hanya mengambil mayoritas suara dari ketiga algoritma yang digunakan serta mengasumsikan bahwa semua model memiliki kontribusi yang sama sehingga teknik ini tidak belajar dari kesalahan model yang digabungkan. Model ini cocok untuk menggabungkan algoritma-algoritma dengan performa yang relatif seimbang, sedangkan pada penelitian ini algoritma *Random Forest* dan *Logistic Regression* memiliki perbedaan performa yang cukup signifikan.

Sebaliknya, teknik *stacking* menunjukkan performa yang lebih baik karena mampu memanfaatkan *meta-learner* yang berfungsi untuk mengenali pola kesalahan maupun kekuatan dari masing-masing model dasar. *Meta-learner* tidak hanya menggabungkan hasil prediksi, tetapi juga belajar bagaimana memanfaatkan *output* dari model-model sebelumnya secara optimal. Misalnya, ketika *Logistic Regression* sering salah dalam memprediksi kasus tertentu, *meta-learner* dapat mengoreksi kelemahan tersebut dengan lebih mempercayai prediksi dari *Random Forest* atau KNN untuk kasus serupa. Dengan cara ini, *stacking* dapat memberikan bobot kontribusi yang berbeda pada tiap model sesuai dengan konteks prediksi, sehingga lebih fleksibel dan adaptif dibandingkan *voting*.

Oleh karena itu, *stacking* lebih cocok digunakan untuk menggabungkan algoritma dengan performa yang tidak seimbang seperti pada penelitian ini, karena mekanisme *meta-learner* mampu mengoptimalkan kombinasi prediksi dengan mempertimbangkan kelebihan dan kelemahan tiap model dasar.

Model dengan *stacking classifier* yang memiliki performa terbaik pada penelitian ini kemudian dievaluasi lebih lanjut menggunakan teknik *k-fold cross validation*. Jumlah *k* yang digunakan pada proses ini adalah sebanyak 5 lipatan. Hasil dari proses ini terlihat seperti pada Gambar 9.

Hasil *k-fold cross validation* pada model *stacking classifier* menunjukkan bahwa nilai akurasi di setiap *fold* berada pada kisaran 0,90 – 0,95 dengan variasi yang relatif kecil. Konsistensi performa ini mengindikasikan bahwa model mampu bekerja secara stabil tanpa dipengaruhi secara signifikan oleh perbedaan pembagian data pada tiap *fold*. Akurasi yang tinggi dan stabil ini juga menandakan bahwa model memiliki kemampuan generalisasi yang baik terhadap data baru serta tidak menunjukkan indikasi *overfitting* yang kuat. *Stacking classifier* dapat dianggap sebagai model yang andal dan optimal dalam menyelesaikan melakukan deteksi penyakit jantung pada penelitian ini.



Gambar 9. Hasil *K-fold Cross Validation* untuk Model *Stacking*

4. Kesimpulan

Model untuk deteksi dini terhadap penyakit jantung dengan menggunakan data dari Framingham Heart Study berhasil dibangun. Penelitian ini membangun beberapa model untuk kemudian ditentukan model mana yang memiliki performa terbaik. Model dengan algoritma individu memiliki performa yang cukup baik, dengan *Random Forest* yang memiliki nilai akurasi tertinggi, yakni sebesar 0,91. Akan tetapi, model dengan algoritma *Logistic Regression* hanya memiliki nilai akurasi sebesar 0,67.

Algoritma *Random Forest*, *Logistic Regression*, dan KNN kemudian digabungkan dengan teknik *voting* dan *stacking*. Performa model dengan teknik *stacking* menunjukkan nilai yang lebih baik dari ketiga model individu, dengan akurasi mencapai 0,92. Akan tetapi, model dengan teknik *voting* tidak menunjukkan performa yang lebih baik dari model individu dengan algoritma *Random Forest*. Hal ini terjadi karena teknik *voting* lebih cocok untuk menggabungkan model dengan performa seimbang, sedangkan pada penelitian ini terdapat perbedaan performa yang cukup signifikan antara model dengan algoritma *Random Forest* dan *Logistic Regression*. Hasil *k-fold cross validation* untuk model *stacking* menunjukkan bahwa model ini memiliki performa yang konsisten, di mana nilai akurasi di setiap *fold* ada pada kisaran 0,90 – 0,95. Hal ini menunjukkan bahwa model ini cukup andal dan optimal dalam melakukan klasifikasi.

Penelitian selanjutnya dapat menerapkan algoritma serta teknik *ensemble learning* lainnya seperti *Gradient Boosting Machines* (GBM), *Extreme Gradient Boosting* (XGBoost), atau LightGBM, yang telah terbukti memiliki performa tinggi dalam berbagai kasus klasifikasi medis. Selain itu, teknik-teknik pra-pemrosesan lain juga dapat diterapkan agar dapat membuat data yang digunakan lebih terdistribusi dengan baik sehingga dapat menghasilkan model dengan performa yang lebih baik pula. Contohnya penggunaan teknik *feature selection* seperti *Recursive Feature Elimination* (RFE) atau *mutual information* untuk memilih fitur-fitur yang paling relevan. Validasi eksternal menggunakan *dataset* independen dari penelitian lain juga dapat dilakukan, agar dapat menguji kemampuan generalisasi model dan memastikan performanya tidak hanya baik pada *dataset* Framingham Heart Study, tetapi juga dapat diterapkan pada populasi berbeda.

Daftar Referensi

- [1] “Cardiovascular diseases (CVDs),” World Health Organization. [Online]. Available: [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)). [Accessed: May 27, 2025].
- [2] R. Fadil, “Gambaran profil LIPID pada pasien penderita jantung koroner di RSPAD Gatot Soebroto,” M.S. thesis, Universitas MH Thamrin, 2024.
- [3] A. Ridwanmo, M. Fadillah, and T. H. Irfani, “Deteksi dini faktor risiko penyakit jantung dan pembuluh darah, hubungan antara obesitas, aktivitas fisik dan kolesterol

- total di Kecamatan Kertapati, Kota Palembang,” *Jurnal Epidemiologi Kesehatan Komunitas*, vol. 5, no.2, pp. 96–103, Aug. 2020, doi: 10.14710/jekk.v5i2.6729.
- [4] C. Andersson, M. Naylor, C. W. Tsao, D. Levy, and R. S. Vasan, “Framingham Heart Study: JACC focus seminar, 1/8,” *Journal of the American College of Cardiology*, vol. 77, no. 3, pp. 293–306, Jun. 2021, doi: 10.1016/j.jacc.2021.01.059.
- [5] A. A. MB and K. WA, “Cardiovascular diseases risk prediction using the Framingham Risk Score,” *Egypt J Occup Med*, vol. 45, pp. 249–264, Sep. 2021, doi: 10.21608/ejom.2021.193283.
- [6] A. Kusnandang, “Framingham score dan Jakarta cardiovascular score untuk menentukan kejadian kardiovaskuler event pekerja Rumah Sakit Pertamina Cirebon,” *Tunas Medika Jurnal Kedokteran dan Kesehatan*, vol. 5, no. 2, pp. 1–5, Oct. 2019. [Online]. Available: <https://jurnal.ugj.ac.id/index.php/tumed/article/view/2714>. [Accessed: May 28, 2025].
- [7] Anharudin and M. Tejamaya, “Perbandingan risiko kardiovaskuler menggunakan metode Framingham, WHO chart dan ASCVD pada pekerja PT. X tahun 2021,” *Promotif: Jurnal Kesehatan Masyarakat*, vol. 12, pp. 77–84, Jun. 2022, doi: 10.56338/pjkm.v12i1.2465.
- [8] A. B. Wibisono and A. Fahrurrozi, “Perbandingan algoritma klasifikasi dalam pengklasifikasian data penyakit jantung koroner,” *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 24, pp. 161–170, 2019, doi: 10.35760/tr.2019.v24i3.2393.
- [9] C. A. ul Hassan *et al.*, “Effectively predicting the presence of coronary heart disease using machine learning classifiers,” *Sensors*, vol. 22, no. 19, p. 7227, Oct. 2022, doi: 10.3390/s22197227.
- [10] S. H. Nurrohman and D. Kurniawan, “Deteksi dini risiko penyakit jantung koroner menggunakan algoritma decision tree dan random forest,” *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 4, pp. 2582–2593, Mar. 2025, doi: 10.47065/bits.v6i4.7029.
- [11] D. Krishnani, A. Kumari, A. Dewangan, A. Singh, and N. S. Naik, “Prediction of coronary heart disease using supervised machine learning algorithms,” in *Proc. IEEE Region 10 Annual Int. Conf. (TENCON)*, Oct. 2019, pp. 367–372. doi: 10.1109/TENCON.2019.8929434.
- [12] R. J. Suhatri, R. D. Syah, M. Hermita, B. Gunawan, and W. Silfianti, “Evaluation of machine learning models for predicting cardiovascular disease based on Framingham Heart Study data,” *ILKOM Jurnal Ilmiah*, vol. 16, no. 1, pp. 68–75, Apr. 2024, doi: 10.33096/ilkom.v16i1.1952.68-75.
- [13] D. Yewale, S. Patil, A. R. Date, and A. Nanthaamornphong, “Heart disease prediction using ensemble methods, genetic algorithms, and data augmentation: A preliminary study,” *Journal of Robotics and Control (JRC)*, vol. 6, no. 3, pp. 1092–1105, 2025, doi: 10.18196/jrc.v6i3.25144.
- [14] V. Shorewala, “Early detection of coronary heart disease using ensemble techniques,” *Inform Med Unlocked*, vol. 26, p. 100655, Jan. 2021, doi: 10.1016/j.imu.2021.100655.
- [15] B. Ashish, ‘Framingham heart study dataset’. [Online]. Available: <https://www.kaggle.com/datasets/aasheesh200/framingham-heart-study-dataset>. [Accessed: July 14, 2025].
- [16] F. Alghifari and D. Juardi, “Penerapan data mining pada penjualan makanan dan minuman menggunakan metode algoritma naïve Bayes,” *Jurnal Ilmiah Informatika (JIF)*, vol. 9, no. 2, pp. 75–81, Sep. 2021, doi: 10.33884/jif.v9i02.3755.

- [17] N. P. A. Widiari, I. M. A. D. Suarjaya, and D. P. Githa, "Teknik data cleaning menggunakan snowflake untuk studi kasus objek pariwisata di Bali," *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, vol. 8, no. 2, pp. 137–145, Aug. 2020, doi: 10.24843/jim.2020.v08.i02.p07.
- [18] M. R. A. Prasetya, A. M. Priyatno, and Nurhaeni, "Penanganan imputasi missing values pada data time series dengan menggunakan metode data mining," *Jurnal Informasi dan Teknologi*, vol. 5, no. 2, pp. 52–62, Jun. 2023, doi: 10.37034/jidt.v5i2.324.
- [19] R. M. West, "Best practice in statistics: The use of log transformation," *Ann Clin Biochem*, vol. 59, no. 3, pp. 162–165, May 2022, doi: 10.1177/00045632211050531.
- [20] P. R. Sihombing, S. Suryadiningrat, D. A. Sunarjo, and Y. P. A. C. Yuda, "Identifikasi data outlier (pencilan) dan kenormalan data pada data univariat serta alternatif penyelesaiannya," *Jurnal Ekonomi dan Statistik Indonesia*, vol. 2, no. 3, pp. 307–316, Jan. 2023, doi: 10.11594/jesi.02.03.07.
- [21] R. Ridwan, E. H. Hermaliani, and M. Ernawati, "Penerapan metode SMOTE untuk mengatasi imbalanced data pada klasifikasi ujaran kebencian," *Jurnal Co-Science*, vol. 4, no. 1, Jan. 2024, doi: <https://doi.org/10.31294/coscience.v4i1.2990>.
- [22] V. R. Joseph, "Optimal ratio for data splitting," *Stat Anal Data Min*, vol. 15, no. 4, pp. 531–538, Aug. 2022, doi: 10.1002/sam.11583.
- [23] H. Marlina, Elmayati, A. Zulus, and H. O. L. Wijaya, "Penerapan algoritma random forest dalam klasifikasi penjurusan di SMA Negeri Tugumulyo," *Brahmana: Jurnal Penerapan Kecerdasan Buatan*, vol. 4, no. 2, pp. 138–143, Jun. 2023, doi: 10.30645/brahmana.v4i2.188.
- [24] R. H. Situngkir and P. Sembiring, "Analisis regresi logistik untuk menentukan faktor-faktor yang mempengaruhi kesejahteraan masyarakat kabupaten/kota di Pulau Nias," *FARABI: Jurnal Matematika dan Pendidikan Matematika*, vol. 6, no. 1, pp. 25–31, May 2023, doi: 10.47662/farabi.v6i1.432.
- [25] S. R. Cholil, T. Handayani, R. Prathivi, and T. Ardianita, "Implementasi algoritma klasifikasi K-Nearest Neighbor (KNN) untuk klasifikasi seleksi penerima beasiswa," *IJCIT (Indonesian Journal on Computer and Information Technology)*, vol. 6, no. 2, pp. 118–127, Dec. 2021, doi: 10.31294/ijcit.v6i2.10438.
- [26] J. K. Nainggolan, F. Sinaga, A. M. Sitorus, A. Khairia, and B. A. Wijaya, "Analisa komparasi dengan algoritma K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM) untuk prediksi penyakit jantung," *Dinamik*, vol. 30, no. 2, pp. 297–306, Jul. 2025, doi: 10.35315/dinamik.v30i2.10254.
- [27] B. Sunarko *et al.*, "Penerapan stacking ensemble learning untuk klasifikasi efek kesehatan akibat pencemaran udara," *Edu Komputika Journal*, vol. 10, no. 1, pp. 55–63, Sep. 2023, doi: 10.15294/edukomputika.v10i1.72080.
- [28] N. Rane, S. P. Choudhary, and J. Rane, "Ensemble deep learning and machine learning: applications, opportunities, challenges, and future directions," *Studies in Medical and Health Sciences*, vol. 1, no. 2, pp. 18–41, Jul. 2024, doi: 10.48185/smhs.v1i2.1225.
- [29] P. Mahajan, S. Uddin, F. Hajati, and M. A. Moni, "Ensemble learning for disease prediction: A review", *Healthcare*, vol. 11, no. 12, p. 1808, Jun. 2023, doi: 10.3390/healthcare11121808.

- [30] S. Raschka, “StackingClassifier: Simple stacking - mlxtend.” [Online]. Available: https://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/. [Accessed: July 31, 2025].
- [31] D. H. Wolpert, “Stacked generalization,” *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992, doi: 10.1016/S0893-6080(05)80023-1.
- [32] S. Raschka, “EnsembleVoteClassifier: A majority voting classifier – mlxtend.”, 2025. [Online]. Available: https://rasbt.github.io/mlxtend/user_guide/classifier/EnsembleVoteClassifier/. [Accessed: July 31, 2025].