BAB II
LANDASAN TEORI

2.1 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

No	Algoritma yang Digunakan	Hasil Utama	Keterbatasan Penelitian	Referensi
1	Logistic Regression, Random Forest, XGBoost	XGBoost: akurasi 85,19%; LR: 83,33%; RF: 16,67% (akibat parameter buruk).	Dataset hanya 303 sampel dari Cleveland, tidak menggunakan validasi silang.	[3]
2	Hybrid Stacking Ensemble (XGBoost, Random Forest, SVM sebagai base learner, Multilayer Perceptron sebagai meta-learner)	Model stacking ensemble dengan seleksi fitur Boruta mencapai performa sangat baik, akurasi tinggi dengan sangat sedikit error (presisi & recall sangat tinggi).	Hanya menggunakan dataset Cleveland (303 sampel), tidak menjelaskanmekanisme voting secara rinci	[7]
3	Random Forest, XGBoost, SVM, Naive Bayes, KNN dalam klasifikasi multi-dataset (Cleveland, Hungary, Switzerland)	RF dan XGBoost mencapai akurasi 0,90 dan 0,92;	Tidak melakukan hyperparameter tuning, hanya melaporkan akurasi tanpa F1-score atau recall	[9]
4	Social Group Optimization (SGO) + Random Forest dan XGBoost pada dataset Cleveland (303 sampel) & Statlog (270 sampel)	Sebelum optimasi: RF (84%-88%), XGB (82%-90%). Setelah optimasi: RF (92%), XGB (90%-93%)	Dataset sangat kecil (303 dan 270 sampel), tidak menggunakan validasi silang	[11]
5	XGBoost, Bagged Tree, Random Forest	XGBoost & Bagged Trees: akurasi 93%; Random Forest: 91% (lebih stabil pada K=10 dengan akurasi 94%).	Hanya dataset Cleveland, metrik hanya akurasi,	[12]
6	XGBoost, Bagged Trees, Gradient Boosting (K-fold cross-validation K=5, K=10)	RF lebih stabil dengan akurasi 94% (K=10) dan 92% (K=5); XGBoost & Bagged Trees: akurasi 93%; RF & Bagged Trees: ROC-AUC tertinggi (95%).	Tidak melakukan seleksi fitur, semua fitur digunakan sehingga kurang praktis untuk deteksi mandir	[17]
7	Logistic Regression, SVM, KNN, Random Forest	RF terbaik: F1-score 92%, recall 93%; cross-validation meningkatkan akurasi 3-11%.	Hanya dataset Cleveland (303 sampel	[18]
8	Logistic Regression, Random Forest, XGBoost	LR: 87%; RF: 89%. RF unggul di sensitivitas (81%) dan F1-score (0,87).	Akurasi dan F1-score masih di bawah 90%,	[20]
9	Logistic Regression, Decision Tree, Random Forest, SVM, XGBoost, ANN	XGBoost: akurasi 90,2%, ROC-AUC 0,94; Random Forest: 89,1%.	Dataset kecil (303 sampel) hanya menggunakan akurasi	[21]
10	Decision Tree, Random Forest, XGBoost.	XGBoost 90%, Random Forest 92%	Tidak menjelasakn alasan RF lebih tinggi	[22]

Berdasarkan sepuluh penelitian terdahulu yang dianalisis, umumnya algoritma seperti

Random Forest, XGBoost, Logistic Regression, SVM, dan Decision Tree digunakan untuk memprediksi penyakit jantung. Hasil utama menunjukkan akurasi model tunggal berkisar antara 85–94%, sementara pendekatan ensemble seperti stacking atau optimasi dengan Social Group Optimization (SGO) mampu meningkatkan performa hingga 93% dengan presisi dan recall yang sangat baik. Namun, sebagian besar penelitian ini memiliki keterbatasan signifikan. Pertama, hampir semuanya menggunakan dataset berukuran kecil (303 sampel Cleveland, 270 sampel Statlog, atau gabungan terbatas dari Hungary dan Switzerland), sehingga generalisasi model masih dipertanyakan. Kedua, beberapa penelitian tidak menerapkan validasi silang yang konsisten atau hanya mengandalkan akurasi sebagai metrik utama tanpa memperhatikan ketidakseimbangan kelas dan dampak fatal dari *false negative* dalam skrining medis. Ketiga, belum ada penelitian yang secara sistematis melakukan seleksi fitur untuk memilih variabel yang dapat diakses masyarakat awam tanpa pemeriksaan invasif seperti EKG atau angiografi. Keempat, tidak satu pun dari penelitian terdahulu yang mengimplementasikan model ke dalam aplikasi nyata yang dapat digunakan secara mandiri oleh pasien atau masyarakat umum.

Penelitian ini hadir untuk mengisi celah (*research gap*) tersebut**.** Pertama, penelitian menggunakan dataset gabungan dari lima sumber (Cleveland, Hungary, Switzerland, Long Beach VA, Statlog) yang menghasilkan 11.543 rekam medis, jauh lebih besar dan representatif. Kedua, model hibrida Random Forest–XGBoost dengan *soft voting* berbobot (2:1) Alasan utama penggunaan model *hybrid* adalah untuk menggabungkan keunggulan komplementer kedua algoritma. Random Forest berbasis *bagging* efektif mengurangi varians dan tahan terhadap *overfitting*, sedangkan XGBoost berbasis *boosting* unggul dalam menekan bias sehingga mampu menangkap pola kompleks dengan akurasi tinggi, meskipun rentan *overfitting* jika tidak dituning. Dengan menggabungkan keduanya melalui *soft voting*, model hibrida diharapkan mencapai keseimbangan *bias-variance trade-off* yang optimal, sehingga performa prediksi lebih unggul dibandingkan model tunggal. Bobot 2:1 (XGBoost : Random Forest) diberikan karena XGBoost memiliki performa individu lebih tinggi (F1-score 0,9652 vs 0,9489). Bobot lebih besar pada XGBoost bertujuan memaksimalkan akurasi, sementara bobot lebih kecil pada Random Forest berfungsi sebagai regularisasi tambahan untuk menjaga stabilitas prediksi dan mencegah *overfitting*. Kombinasi bobot ini terbukti menghasilkan F1-score tertinggi pada

data validasi serta selisih akurasi latih-uji terkecil (0,62%) dibandingkan skenario bobot lainnyaddikembangkan untuk menggabungkan stabilitas bagging dari Random Forest dan kemampuan diskriminasi tinggi dari XGBoost, serta dievaluasi dengan metrik yang sesuai konteks medis (F1-score sebagai metrik utama, recall diprioritaskan untuk meminimalkan *false negative*, dan AUC-ROC). Ketiga, *backward feature selection* diterapkan untuk menyederhanakan dari 11 fitur menjadi 7 fitur yang mudah diakses masyarakat (usia, jenis kelamin, nyeri dada, tekanan darah istirahat, kolesterol, gula darah puasa, denyut jantung maksimal) tanpa menurunkan performa secara signifikan (akurasi hibrida tetap >98%). Keempat, model terbaik yang telah melalui proses validasi dan pengujian diimplementasikan ke dalam sebuah prototipe aplikasi web bernama CardioPredict menggunakan framework Streamlit. Melalui antarmuka yang sederhana dan ramah pengguna, masyarakat umum dapat melakukan skrining dini risiko penyakit kardiovaskular secara mandiri, cukup dengan memasukkan beberapa parameter kesehatan dasar seperti usia, tekanan darah, kadar kolesterol, kebiasaan merokok, dan riwayat diabetes. Proses ini hanya memerlukan biaya sangat rendah, berkisar antara Rp15.000 hingga Rp120.000, tergantung pada jumlah parameter yang digunakan. Angka ini sangat kontras dibandingkan dengan metode konvensional seperti pemeriksaan EKG, ekokardiografi, atau tes darah lanjutan yang dapat menghabiskan biaya Rp700.000 hingga Rp10.000.000, belum termasuk biaya konsultasi dokter dan transportasi ke fasilitas kesehatan. Dengan hadirnya CardioPredict, kendala biaya tinggi dan keterbatasan akses terhadap layanan kesehatan profesional dapat ditekan secara signifikan. Dengan demikian, penelitian ini tidak hanya berhasil menghasilkan model prediksi yang lebih akurat dan stabil dibandingkan pendekatan sebelumnya, tetapi juga memberikan solusi praktis, terjangkau, dan siap pakai yang dapat langsung diakses oleh masyarakat luas—termasuk mereka yang tinggal di daerah terpencil dengan fasilitas kesehatan terbatas.

2.2 Teori yang berkaitan

2.2.1 Penyakit Jantung

Di tingkat global, penyakit jantung tercatat sebagai salah satu jenis penyakit dengan tingkat fatalitas tertinggi. Gangguan pada organ vital ini dapat menghambat aliran darah ke seluruh tubuh, memicu kerusakan pada berbagai organ, dan pada akhirnya berujung pada kematian [2]. Beragam faktor berkontribusi terhadap munculnya penyakit jantung, di antaranya hipertensi, diabetes, obesitas, serta kebiasaan hidup yang kurang sehat. Mengingat dampaknya yang serius, upaya deteksi dini dan penanganan yang tepat menjadi langkah krusial untuk meminimalisir risiko kesehatan yang lebih berat.

Dalam praktik medis, diagnosis penyakit jantung dapat ditempuh melalui berbagai prosedur seperti pemeriksaan fisik langsung, analisis sampel darah, pemeriksaan menggunakan elektrokardiogram (EKG), uji latih jantung (stress test), hingga teknologi pencitraan jantung [19]. Masing-masing pendekatan memiliki kelebihan dan keterbatasan tersendiri yang disesuaikan dengan jenis kelainan jantung serta kondisi pasien. Perkembangan teknologi digital beberapa tahun terakhir membuka peluang baru melalui pemanfaatan pemrosesan citra dan machine learning yang memungkinkan diagnosis penyakit jantung dilakukan secara non-invasif dengan tingkat akurasi yang menjanjikan [20].

Bagi pasien yang telah terdiagnosis penyakit jantung atau memiliki faktor risiko tertentu, prediksi tingkat risiko dapat dilakukan menggunakan beragam model statistik maupun machine learning [21]. Hasil prediksi ini menjadi acuan bagi tenaga medis dalam menentukan langkah pencegahan maupun terapi yang paling sesuai, sekaligus memperkirakan perkembangan kondisi pasien ke depannya. Berbagai parameter umumnya dijadikan masukan dalam model prediksi, antara lain usia, jenis kelamin, riwayat kesehatan keluarga, tekanan darah, kadar kolesterol, serta pola hidup pasien [22].

2.2.2 UCI Machine Learning Repository

UCI Machine Learning Repository merupakan salah satu sumber dataset yang paling banyak dirujuk dan dimanfaatkan dalam berbagai studi pengembangan algoritma machine learning serta teknik data mining. Repositori ini menyimpan beragam jenis data, mencakup data terstruktur, citra, teks, hingga data berukuran kecil [23]. Secara umum, UCI Machine Learning Repository adalah kumpulan basis data, teori domain, dan pembangkit data yang dirancang untuk mendukung komunitas machine learning dalam melakukan analisis empiris terhadap berbagai algoritma yang dikembangkan.

Repositori ini pertama kali dibangun sebagai arsip terbuka pada tahun 1987 oleh David Aha bersama rekan-rekan mahasiswa pascasarjana di Universitas California, Irvine. Sejak awal kemunculannya, UCI Machine Learning Repository telah menjadi rujukan utama bagi para peneliti di seluruh dunia dalam memperoleh kumpulan data untuk keperluan eksperimen machine learning. Popularitasnya tercermin dari jumlah sitasi yang mencapai lebih dari 1000 kali, menjadikannya salah satu dari 100 karya tulis yang paling banyak dikutip di seluruh bidang ilmu komputer.

2.2.3 Klasifikasi

Klasifikasi merupakan suatu proses penyusunan model atau algoritma yang bertujuan untuk menempatkan atau memprediksi label kelas dari suatu data berdasarkan karakteristik yang dimilikinya. Secara lebih luas, klasifikasi dapat dipahami sebagai bentuk analisis data yang digunakan untuk memisahkan atau mengelompokkan entitas berdasarkan atribut-atribut yang melekat padanya.

Salah satu keunggulan utama dari metode klasifikasi adalah kemampuannya dalam memberikan keputusan yang bersifat objektif dan konsisten, karena tidak dipengaruhi oleh faktor emosional maupun subjektivitas pengguna. Metode ini juga sangat berguna ketika berhadapan dengan data yang kompleks, terutama yang memiliki jumlah atribut banyak [26]. Namun demikian, klasifikasi juga tidak lepas dari kelemahan, salah satunya

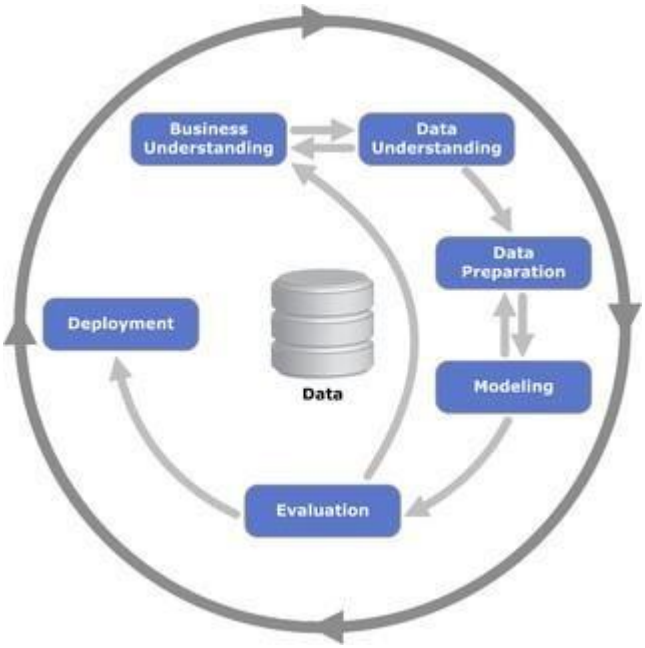
adalah masalah akurasi yang mana hasil prediksi bisa saja tidak tepat atau menyimpang dari kondisi sebenarnya [27].

Penerapan metode klasifikasi sangat beragam tergantung pada bidang yang memanfaatkannya. Dalam dunia kesehatan, misalnya, data medis dapat dikelompokkan ke dalam kategori penyakit atau kondisi tertentu, sehingga membantu para dokter dalam menegakkan diagnosis dan memberikan terapi yang lebih tepat sasaran [28]. Di sektor bisnis, klasifikasi digunakan untuk meramalkan perilaku konsumen atau preferensi terhadap suatu produk, yang pada gilirannya mendukung perusahaan dalam mengambil keputusan strategis yang lebih efektif [29].

Pemilihan metode klasifikasi yang sesuai menjadi faktor kunci untuk memperoleh hasil yang akurat dan bermanfaat. Keputusan ini sangat ditentukan oleh jenis data yang tersedia serta tujuan analisis yang ingin dicapai [30]. Selain itu, dalam menerapkan metode klasifikasi, penting pula mempertimbangkan aspek-aspek seperti validasi model, dan evaluasi kinerja model. Teknik validasi silang, misalnya, dapat diterapkan untuk menguji keandalan model, sementara teknik interpretasi model membantu memahami faktor-faktor apa saja yang paling berpengaruh terhadap hasil klasifikasi [31].

2.3 Framework/Algoritma yang digunakan

2.3.1 CRISP-DM



Gambar 2. 1 CRISP-DM Methodology

Sumber: Medium (2023)

Gambar 2.1 menunjukkan alur proses Cross-Industry Standard Process for Data Mining (CRISP-DM). CRISP-DM merupakan salah satu metodologi yang paling banyak digunakan sebagai standar dalam pelaksanaan proses *data mining*. Metode ini terdiri dari enam tahapan utama, yaitu business understanding, data understanding, data preparation, modeling,evaluation, dan deployment [19].

2.3.1.1Business Understanding

Sebagai tahap awal dalam kerangka kerja CRISP-DM, tahap business understanding berfokus pada pemahaman terhadap permasalahan dan tujuan penelitian. Pada tahap ini, dilakukan identifikasi kebutuhan serta penentuan tujuan analisis agar sesuai dengan permasalahan yang akan diselesaikan. Selain itu, perlu ditentukan juga sumber data yang akan digunakan serta bagaimana data tersebut diperoleh untuk mendukung proses analisis selanjutnya.

2.3.1.2 Data Understanding

Tahap data understanding berfokus pada proses memahami data yang akan digunakan dalam penelitian. Pada tahap ini dilakukan eksplorasi data untuk mengetahui karakteristik, struktur, dan isi data secara menyeluruh. Selain itu, data juga dideskripsikan agar setiap informasi yang terdapat di dalam dataset dapat dipahami dengan jelas.

Pemeriksaan kualitas data juga menjadi bagian penting dalam tahap ini, seperti mengidentifikasi adanya data yang hilang, data tidak konsisten, maupun kesalahan pencatatan. Setiap atribut atau variabel dalam dataset juga akan ditentukan serta dijelaskan fungsinya sehingga dapat diketahui peran masing-masing atribut dalam proses analisis selanjutnya.

2.3.1.3 Data Preparation

Tahap data preparation atau data preprocessing berfokus pada proses penyiapan data agar siap digunakan dalam tahap pemodelan. Data yang telah dianalisis pada tahap sebelumnya dan ditemukan memiliki kualitas yang kurang baik akan diperbaiki pada tahap ini.

Perbaikan kualitas data dilakukan melalui proses pembersihan data (data cleaning), seperti mengatasi nilai kosong (null value), nilai yang hilang (missing value), data yang mengandung noise, serta data yang menyimpang (outlier). Tahap ini bertujuan untuk menghasilkan dataset yang lebih bersih dan terstruktur sehingga dapat meningkatkan kinerja model machine learning yang akan dibangun.

2.3.1.4 Modeling

Tahap modeling dalam metode CRISP-DM berfokus pada proses pembangunan model dengan menggunakan algoritma yang telah dipilih sebelumnya. Pada tahap ini, data yang telah melalui proses persiapan akan digunakan untuk melatih model sehingga dapat menghasilkan pola atau prediksi sesuai dengan tujuan penelitian.

Pemilihan teknik data mining yang digunakan harus disesuaikan dengan jenis permasalahan yang dihadapi serta karakteristik data yang tersedia, sehingga model yang dihasilkan dapat bekerja secara optimal dan memberikan hasil yang akurat.

2.3.1.5 Evaluation

Tahap evaluation berfokus pada proses penilaian terhadap model yang telah dibangun pada tahap modeling. Evaluasi dilakukan untuk mengetahui sejauh mana kinerja model dalam menghasilkan prediksi yang akurat serta sesuai dengan tujuan penelitian.

Penilaian model dapat dilakukan menggunakan beberapa metrik evaluasi, seperti akurasi, precision, recall, dan F1-score. Hasil evaluasi ini digunakan untuk menentukan apakah model yang dibangun sudah cukup baik atau masih perlu dilakukan perbaikan sebelum digunakan pada tahap selanjutnya.

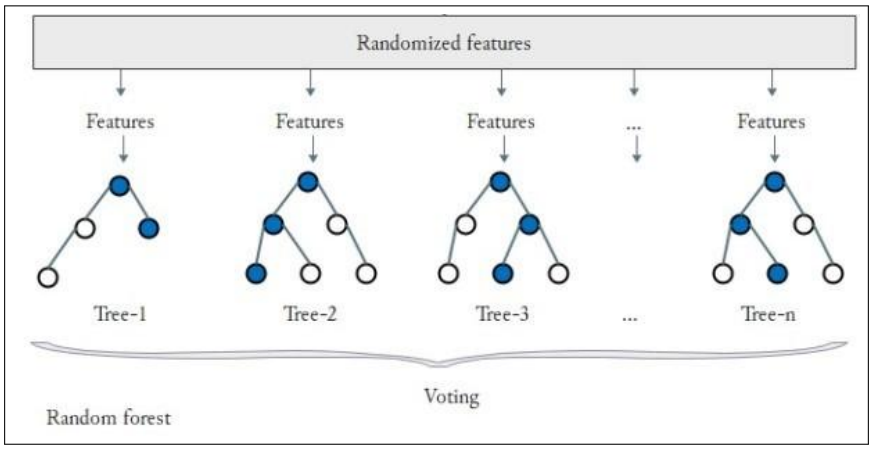
2.3.1.6 Deployment

Tahap deployment merupakan tahap terakhir dalam metode CRISP-DM yang berfokus pada penerapan model yang telah dibangun ke dalam penggunaan nyata. Pada tahap ini, model yang sudah melalui proses evaluasi akan digunakan untuk mengolah data baru atau membantu proses pengambilan keputusan.

Hasil dari tahap deployment dapat berupa sistem prediksi, aplikasi, laporan analisis, atau visualisasi yang dapat dimanfaatkan oleh pengguna sesuai dengan tujuan penelitian yang telah ditentukan

2.3.2 Algoritma Random Forest

Algoritma Random Forest (RF) merupakan salah satu metode *machine learning* yang banyak digunakan untuk tugas klasifikasi maupun prediksi [31]. Algoritma ini bekerja dengan membangun sejumlah *Decision Tree* menggunakan sampel data yang dipilih secara acak, kemudian hasil prediksi dari setiap pohon digabungkan untuk menghasilkan keputusan akhir [32]. Dengan pendekatan tersebut, Random Forest mampu menangani data yang tidak seimbang serta tetap dapat digunakan meskipun terdapat nilai yang hilang (*missing value*) [33]. Selain itu, Random Forest memiliki kemampuan yang baik dalam mengurangi risiko *overfitting*, karena setiap pohon keputusan dilatih menggunakan subset data yang berbeda [34]. Metode ini juga dapat digunakan untuk mengenali pola dalam data, membantu proses pengambilan keputusan, membangun model prediktif, serta menyelesaikan berbagai permasalahan analisis data [35]. Keunggulan lainnya adalah Random Forest mampu memproses jumlah variabel input yang sangat banyak tanpa harus menghilangkan variabel tertentu [36]. jumlah variabel input yang sangat banyak tanpa harus menghilangkan variabel tertentu . Diagram alur proses kerja algoritma Random Forest dapat dilihat pada Gambar 2.1.



Gambar 2.2. Diagram Alur Cara Kerja Algoritma Random Forest

Sumber: [20]

Pada Gambar 2.2 menunjukkan proses dari cara kerja algoritma *Random Forest* sebagai berikut [20].

1. Randomized Features, yaitu proses pemilihan sejumlah fitur dari dataset secara acak untuk setiap pohon keputusan yang akan dibangun.

2. Pembentukan Decision Tree, yaitu membuat beberapa pohon keputusan menggunakan subset fitur yang telah dipilih secara acak sebelumnya.

3. Proses Voting (Ensemble Learning) dilakukan setelah seluruh pohon keputusan terbentuk, di mana setiap pohon menghasilkan prediksi terhadap data yang diuji.

4. Penentuan Hasil Akhir (Final Prediction) diperoleh berdasarkan hasil voting mayoritas dari seluruh pohon keputusan yang ada.

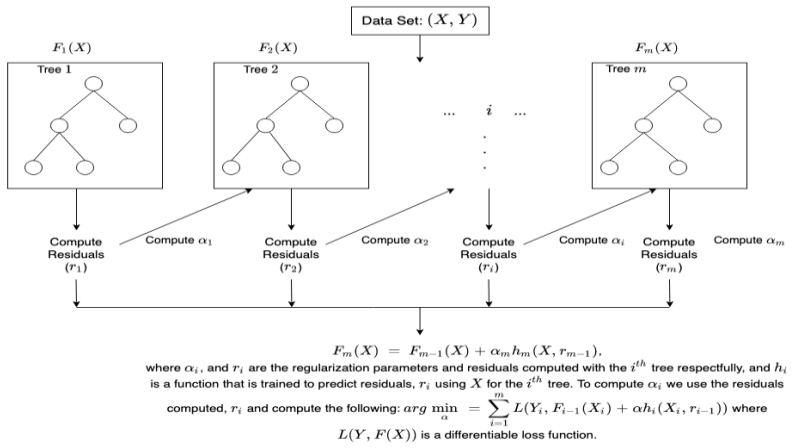
Dalam bidang data mining, algoritma Random Forest dapat digunakan baik untuk pendekatan deskriptif maupun prediktif. Pendekatan deskriptif bertujuan untuk memberikan gambaran atau ringkasan dari data yang dianalisis, sedangkan pendekatan prediktif memanfaatkan data historis untuk menemukan pola atau kecenderungan yang dapat digunakan dalam melakukan prediksi di masa mendatang.

Data mining prediktif umumnya menggunakan metode Supervised Learning, yaitu pembelajaran yang memanfaatkan data berlabel, sedangkan data mining deskriptif

biasanya menggunakan metode Unsupervised Learning, yaitu pembelajaran tanpa label. Random Forest mampu mengolah dataset berukuran besar dan menangani banyak variabel sekaligus tanpa perlu menghilangkan fitur tertentu. Oleh karena itu, algoritma ini banyak digunakan dalam klasifikasi otomatis, analisis prediktif, serta berbagai aplikasi pembelajaran terawasi [19].

2.3.3 XGBoost

XGBoost merupakan pengembangan dari algoritma Gradient Boosting yang dirancang untuk meningkatkan kinerja dan efisiensi model. Algoritma ini dapat digunakan untuk berbagai permasalahan *machine learning*, baik klasifikasi maupun regresi. XGBoost bekerja dengan cara menggabungkan beberapa model sederhana (*weak learners*) menjadi model yang lebih kuat (*strong learner*) melalui proses *boosting*. Dalam prosesnya, XGBoost umumnya menggunakan struktur Decision Tree sebagai dasar pembentukan model sehingga mampu menghasilkan prediksi yang lebih akurat [3].



Gambar 2. 3 XGBoost *Classifier*

Sumber: [25]

Cara kerja XGBoost dapat digambarkan melalui model berbasis pohon keputusan (tree-based model) yang menggunakan fitur-fitur dalam dataset sebagai node kondisi (conditional node). Setiap node akan membagi data ke dalam beberapa cabang berdasarkan nilai fitur

tertentu hingga terbentuk leaf node, yaitu node akhir yang merepresentasikan hasil prediksi atau kelas yang ditentukan [3].

Pada proses pelatihan model, XGBoost menghitung residual atau selisih antara nilai aktual dan nilai prediksi. Residual tersebut dihitung pada setiap pohon yang terbentuk dan digunakan sebagai dasar untuk membangun pohon berikutnya. Proses pembentukan pohon dilakukan secara iteratif, di mana hasil prediksi dari setiap pohon akan dijumlahkan secara bertahap hingga mencapai nilai tertentu yang berada di atas atau di bawah batas (*threshold*). Nilai akhir tersebut kemudian digunakan untuk menentukan kelas atau hasil prediksi dari suatu data [25].

Penggunaan algoritma XGBoost memiliki beberapa keunggulan, di antaranya waktu pelatihan model yang relatif cepat, kemampuan menghasilkan model dengan tingkat akurasi yang tinggi, serta kemampuan dalam mengurangi kesalahan (*error*) pada model prediksi [26]

2.3.4 Klasifikasi Hibrida

Klasifikasi Hibrida (Hybrid Classification) merupakan metode klasifikasi yang mengombinasikan dua atau lebih algoritma klasifikasi dengan tujuan meningkatkan kinerja model yang dihasilkan. Pendekatan ini memanfaatkan keunggulan dari
masing-masing algoritma sehingga hasil klasifikasi menjadi lebih akurat dan stabil.Dalam penerapannya, klasifikasi hibrida sering digunakan pada dataset yang memiliki jumlah atribut besar dan tingkat kompleksitas yang tinggi. Dengan menggabungkan beberapa algoritma klasifikasi, kelemahan yang terdapat pada satu algoritma dapat ditutupi oleh kelebihan algoritma lainnya, sehingga performa model secara keseluruhan dapat ditingkatkan [33].

2.3.4.1 Soft Voting

Metode *voting* terbagi menjadi *hard voting* dan *soft voting*. *Hard voting* menentukan kelas akhir berdasarkan suara terbanyak dari seluruh model, tanpa mempertimbangkan tingkat keyakinan masing-masing model terhadap prediksinya. Sebaliknya, *soft voting* memanfaatkan probabilitas prediksi dari setiap model, di mana probabilitas untuk setiap kelas dijumlahkan atau dirata-rata, lalu kelas dengan nilai probabilitas tertinggi dipilih sebagai prediksi akhir [4]. Pendekatan ini lebih informatif dan unggul karena mempertimbangkan seberapa yakin suatu model terhadap keputusannya, bukan hanya label kelas yang diberikan. Pada model *hybrid*, *soft voting* menggabungkan berbagai algoritma yang berbeda arsitektur, seperti Random Forest, Support Vector Machine, dan K-Nearest

Neighbors, untuk memanfaatkan kelebihan komplementer masing-masing sekaligus saling menutupi kelemahannya [8]. Penerapan *soft voting* pada model *hybrid* menawarkan sejumlah keunggulan, antara lain peningkatan akurasi prediksi, reduksi risiko *overfitting*, serta kemampuan generalisasi yang lebih baik terhadap data baru.

$$P_{\text{hybrid}}(y = c) = \frac{w_{\text{XGB}} \cdot P_{\text{XGB}}(y = c) + w_{\text{RF}} \cdot P_{\text{RF}}(y = c)}{w_{\text{XGB}} + w_{\text{RF}}}$$

Symbol	Description
c	Class index (0 = no heart disease, 1 = heart disease)
y	Actual target variable
$\hat{y}$	Final prediction output by the hybrid model
$P_{\text{hybrid}}(y = c)$	Final probability of class c from the hybrid model
$P_{\text{XGB}}(y = c)$	Probability of class c predicted by the XGBoost model
$P_{\text{RF}}(y = c)$	Probability of class c predicted by the Random Forest model
w_{XGB}	Weight assigned to the XGBoost model (value = 2)
w_{RF}	Weight assigned to the Random Forest model (value = 1)

Gambar 2.4 Formula Hybrid model (Soft Voting)

2.3.5 Confusion Matrix

Confusion Matrix merupakan salah satu metode yang digunakan untuk mengevaluasi kinerja model klasifikasi. Metode ini menunjukkan perbandingan antara hasil prediksi yang dihasilkan oleh model dengan nilai aktual dari data yang sebenarnya. Melalui Confusion Matrix, kesalahan klasifikasi yang dilakukan oleh model dapat dianalisis sehingga dapat diketahui bagian mana dari model yang masih perlu diperbaiki. Confusion Matrix disusun dalam bentuk tabel yang terdiri dari dua sumbu, yaitu sumbu vertikal (baris) yang menunjukkan kelas aktual (actual class**)** dan sumbu horizontal (kolom) yang menunjukkan kelas hasil prediksi (predicted class**)** dari model.

1. *True Positive*

   True Positive yaitu data positif yang diprediksi dengan benar sebagai positif.

2. *False Positive*

   False Positive yaitu data negatif yang diprediksi secara salah sebagai positif atau *Type I Error*.

3. *True Negative*

   True Negative yaitu data negatif yang diprediksi dengan benar sebagai negatif.

4. *False Negative*

   False Negative yaitu data positif yang diprediksi secara salah sebagai negatif atau *Type II Error*.

lain. Berikut merupakan metode evaluasi menggunakan metrik evaluasi, antara

1. Akurasi (*Accuracy)*

   Akurasi yaitu akurasi mengukur proporsi prediksi yang benar terhadap total prediksi.

   Rumus 2.1 menunjukkan cara perhitungan *Accuracy*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

2. Presisi (*Precision*)

   Presisi yaitu presisi mengukur seberapa akurat model dalam memprediksi kelas positif. Rumus 2.2 menunjukkan cara perhitungan *Precision*.

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

3. Recall

   Recall yaitu untuk mengukur seberapa baik model dalam mendeteksi semua kasus positif.

   Rumus 2.3 menunjukkan cara perhitungan *Recall*.

$$Recall = \frac{TP}{TP + FN} \quad (2.3)$$

4. F1-*Score*

   F1-*Score* yaitu rata rata harmonis dari presisi dan *recall* ketika distribusi kelas tidak seimbang.

   Rumus 2.4 menunjukkan cara perhitungan *F1 Score*.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

2.3.6 Tools yang digunakan

Gambar 2.5 Logo Google collab

Pada penelitian ini, proses pengolahan data dan pembangunan model *machine learning* dilakukan menggunakan Google Colab. Google Colab merupakan platform berbasis *cloud* yang memungkinkan pengguna untuk menulis dan menjalankan kode Python melalui browser tanpa perlu melakukan instalasi perangkat lunak tambahan pada komputer.

Google Colab dipilih karena menyediakan berbagai pustaka (*library*) yang mendukung proses *data mining* dan *machine learning*, seperti NumPy, Pandas, Matplotlib, dan Scikit-learn, sehingga memudahkan dalam melakukan pengolahan data, pembuatan model, serta evaluasi model. Selain itu, Google Colab juga menyediakan sumber daya komputasi yang cukup baik, seperti CPU dan GPU, yang dapat mempercepat proses pelatihan model.

Penggunaan Google Colab memudahkan proses penelitian karena data dapat diakses secara online, kode dapat dijalankan secara interaktif, serta hasil analisis dapat langsung ditampilkan dalam bentuk tabel maupun grafik. Hal ini membuat Google Colab menjadi salah satu tools yang efektif dan efisien dalam pengembangan model *machine learning*.

Gambar 2.6 Logo Streamlit

Pada penelitian ini, Streamlit digunakan sebagai tools untuk menampilkan hasil model *machine learning* dalam bentuk aplikasi berbasis web yang interaktif. Streamlit merupakan framework Python yang digunakan untuk membangun aplikasi web sederhana tanpa memerlukan pengetahuan mendalam tentang pengembangan web seperti HTML, CSS, atau JavaScript.

Streamlit memungkinkan model *machine learning* yang telah dibuat untuk diintegrasikan ke dalam sebuah aplikasi sehingga pengguna dapat melakukan pengujian atau prediksi secara langsung dengan memasukkan data melalui antarmuka yang tersedia. Dengan menggunakan Streamlit, hasil prediksi dapat ditampilkan secara cepat dan mudah dipahami oleh pengguna.

Selain itu, Streamlit memudahkan proses pengembangan aplikasi karena memiliki tampilan yang sederhana, mudah digunakan, serta dapat dijalankan secara lokal maupun secara online. Oleh karena itu, Streamlit sangat cocok digunakan sebagai media implementasi atau deployment model *machine learning* yang telah dibangun.