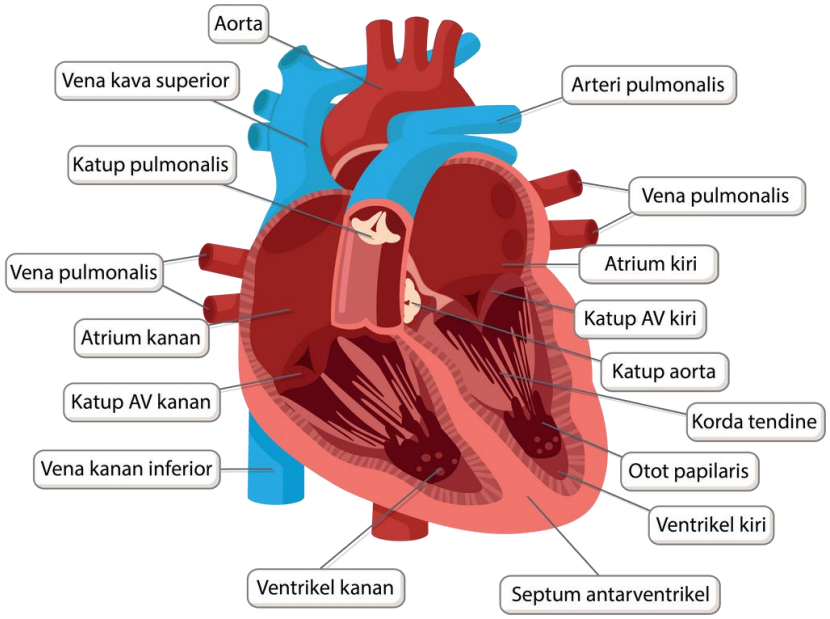
BAB II
LANDASAN TEORI

2.1 Jantung

Jantung merupakan organ vital dalam sistem peredaran darah yang berfungsi untuk memompa darah ke seluruh tubuh, sehingga oksigen dan nutrisi dapat terdistribusi dengan baik. Organ ini terletak di rongga dada, sedikit contong ke kiri dan dilindungi oleh tulang dada serta rongga rusuk. Jantung terdiri dari empat ruang utama, yaitu kiri dan kanan di bagian atas serta ventrikel kiri dan kanan di bagian bawah yang dipisahkan oleh otot yang disebut septum untuk mencegah pencampuran darah kaya oksigen dan miskin oksigen. Darah yang miskin oksigen dari tubuh masuk ke atrium kanan, kemudian dialirkan ke ventrikel kanan melalui katup trikuspid sebelum di pompa ke paru-paru melalui katup pulmonal untuk mendapatkan oksigen. Setelah darah teroksigensi di paru-paru, darah dialirkan ke atrium kiri melalui vena pulmonal lalu diteruskan ke ventrikel kiri melalui katup mitral yang berfungsi mencegah darah kembali ke atruim kiri. Dari ventrikel kiri, darah yang kaya oksigen dipompa ke seluruh tubuh melalui aorta. Proses ini terjadi secara terus-menerus untuk memastikan tubuh mendapatkan suplai oksigen yang cukup guna menunjang berbagai fungsi organ (Dewi, 2023).



Gambar 2. 1. Anatomi Jantung (Junifer Pangaribuan et al., 2021)

Namun, bebagai faktor dapat menyebabkan gangguan pada jantung yang dikenal sebagai penyakit jantung. Penyakit ini dapat menyerang berbagai bagian jantung seperti pembuluh darah, irama, katup atau bahakan terjadi akibat kelainan bawaan. Jika tidak ditangani dengan baik, penyakit jantung dapat meningkatkan risiko komplikasi serius seperti serangan jantung, stroke, hingga kematian (Junifer Pangaribuan et al., 2021). Banyak faktor yang berperan dalam meningkatkan risiko penyakit jantung. Faktor-faktor tersebut meliputi usia, tekanan darah, kadar kolestrol, kadar gula darah, obesitas, kebiasaan merokok, tingkat aktivitas fisik, pola makan, serta tingkat stres. Pemahaman mengenai kontribusi masing-masing faktor terhadap risiko penyakit jantung, tabel 2.1 menyajikan rentang nilai yang digunakan sebagai indikator umum dalam menilai tingkat risiko penyakit jantung (Tampubolon et al., 2023), (Usri et al., 2022), (Bachtiiar et al., 2023), (Naomi et al., 2021).

Tabel 2.1 Indikator Umum Risiko Penyakit Jantung

Indikator Umum Risiko	**Rentang Nilai**	**Keterangan**
Usia	≥ 45 tahun	Risiko meningkat dengan bertambahnya usia.
Tekanan Darah (Hipertensi)	≥140/90 mmHg	Tekanan darah tinggi meningkatkan risiko aterosklerosis.
Kolestrol Total	≥200 mg/dL	Peningkatan kadar kolestrol meningkatkan risiko penyakit jantung.
LDL (Kolestrol Jahat)	≥ 130 mg/dL	LDL tinggi menyebabkan penumpukan plak pada arteri.

Indikator Umum Risiko	Rentang Nilai	Keterangan
HDL (Kolestrol Baik)	≤ 40 mg/dL (laki-laki), ≤ 50 mg/dL (perempuan)	HDL rendah meningkatkan risiko penyakit jantung.
Diabetes (Gula Darah Puasa)	≥ 126 mg/dL	Diabetes meningkatkan risiko komplikasi kardiovaskular.
Obesitas (BMI)	≥ 25 kg/m^2.	Obesitas berhubungan dengan peningkatan tekanan darah dan kolestrol.
Merokok	Ya (Aktif/Pasif)	Merokok merusak pembuluh darah dan meningkatkan risiko penyakit jantung.
Aktivitas Fisik	< 150 menit/minggu (olahraga ringan-sedang).	Kurangnya aktivitas fisik meningkatkan risiko penyakit jantung.
Lingkar Pinggang	> 90 cm (pria), > 80 cm (wanita)	Lingkar pinggang yang melebihi batas normal menunjukkan obesitas sentral.
Presentase Lemak Tubuh	> 22% (pria), > 35% (wanita).	Persentase lemak tubuh yang tinggi berhubungan dengan peningkatan risiko penyakit jantung.
Trigliserida	≥ 150 mg/dL.	Kadar trigliserida yang tinggi dapat meningkatkan risiko penyakit jantung.
Tekanan Darah Diastolik.	≥ 90 mmHg	Tekanan darah diastolik yang tinggi juga berkontribusi terhadap peningkatan risiko penyakit jantung.

Dalam hal ini, penggunaan teknik *machine learning* di bidang medical memiliki potensi yang signifikan dalam meningkatkan ketepatan diagnosis serta prediksi. *Machine learning* merupakan cabang dari kecerdasan buatan yang memungkinkan komputer belajar dari data dan membuat prediksi atau keputusan

tanpa perlu diprogram secara eksplisit. Dengan menanfaatkan model tertentu, *machine learning* mampu menganalisis menganalisis data dalam jumlah besar dan kompleks secara efisien(Rahman, 2024). Penelitian oleh (Mohamed et al., 2023), (Ahmad & Polat, 2023), (Kavitha et al., 2021), (Saboor et al., 2022) menunjukan bahwa penerapan *machine learning* dalam sistem deteksi otomatis telah mengalami perkembangan yang signifikan.

2.2 *Extreme Gradient Boosting* (*XGBoost*)

Extreme Gradient Boosting (*XGBoost*) merupakan metode *supervised learning* yang dapat digunakan untuk tugas klasifikasi dan regresi (Nugraha & Irawan, 2023). Model ini termasuk dalam sistem *machine learning* berbasis tree boosting yang dioptimalkan untuk membangun pohon keputusan dalam skala besar. Sebagai penyempurnaan dari gradient boosting, *XGBoost* dikembangkan sebagai metode *ensemble* berbasis decision tree yang dirancang untuk mempercepat proses komputasi bahkan saat menangani dataset berukuran besar (Roihan et al., 2020). Model *XGBoost* beroperasi dengan mengkombinasikan beberapa pengklasifikasi lemah menjadi model yang lebih kuat melalui proses pelatihan berurutan, dimana hasil klasifikasi sebelumnya yang dikenal sebagai residuals atau error digunakan untuk meningkatkan prediksi selanjutnya. Model ini juga menerapkan regulasi dalam fungsi objektif untuk mengurangi risiko *overfitting* dengan fungsi objektif yang dijelaskan pada persamaan (1) (Aditya et al., 2024).

$$0 - \sum_{i=1}^{n} L(Y_I, F_{XI})) + \sum_{k=1}^{t} R(F_k) + C \tag{2.1}$$

Penjelasan persamaan (1) sebagai berikut:

a. $L(yi,Fxi))$: *Loss function* yang berfungsi untuk mengukur tingkat akurasi model dalam memprediksi model.

b. $R(fk)$: Regularisasi yang berfungsi mencegah overfitting, diformulasikan sebagai $aH + \frac{1}{2}n + \sum_{j=1}^{H} wj^2$ dengan :

a : menyatakan tingkat kompleksitas pada daun.

H : menunjukan jumlah daun dalam model.

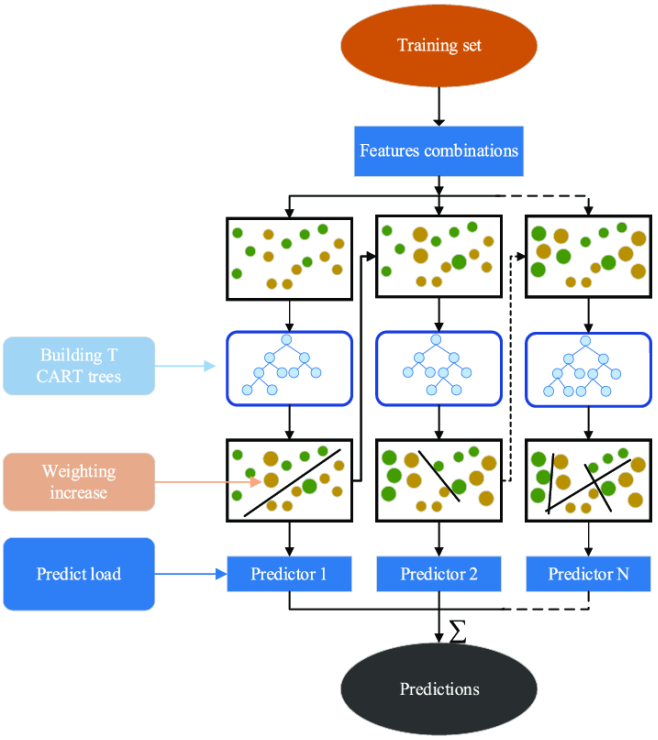
n : mengindikasikan parameter penalti.

w_j^2 : mengacu pada output yang dihasilkan oleh setiap simpul daun.

c. C : Konstanta yang dapat diabaikan secara efektif.

Sesuai dengan maknanya, Boosting berfungsi sebagai teknik peningkatan bagi model pembelajaran lemah dengan menyesuaikan bobot kesalahan klasifikasi sehingga mendekati nilai optimum pada hasil akhir. Beberapa metode boosting yang telah banyak digunakan mencakup Categorical Boosting, *Gradient Boosting Machine, Light Gradient Boosting mechine, Adaptive Bossting*, dan *XGBoost*. *XGBoost* dianggap sebagai pengembangan lebih lanjut dari teknik bagging dan model boosting lainnya karena mampu mengatasi *overfitting* melalui proses regulasi. Selain itu, model ini memiliki kinerja yang lebih cepat berkat pemrosesan paralel (multu-threading CPU) yang memungkinkan pengolahan node secara lebih efisien (Toharudin et al., 2023).

Berikut merupakan skematik dari model *XGBoost* yang tercantum pada gambar 2:



Gambar 2. 2 Skematik model *XGBoost* (Yao et al., 2022).

Pada gambar 2.2 model *XGBoost* bekerja dengan membangun serangkaian pohon keputusan secara bertahap untuk memperbaiki kesalahan prediksi yang terjadi di iterasi sebelumnya. Proses ini dimulai dengan memasukan data fitur dan label target ke dalam model. Persamaan yang digunakan untuk hasil akhirnya dari fungsi prediksi dari setiap pohon keputusan dan kombinasi dari semua pohon yang telah dibangun dalam tahap ini sebagai berikut.

$$y_i = \sum_{k=1}^{K} f_k(x_i), \qquad f_k \in F \tag{2.2}$$

Penjelasan persamaan (2) adalah sebagai berikut:

a. y_i = prediksi akhir untuk data ke-i.

b. K = jumlah total pohon dalam model.

c. $f_k(x_i)$ = fungsi pohon ke-k yang menghasilkan prediksi untuk input x_i.

d. F = ruang fungsi yang berisi semua pohon keputusan.

Model harus mengoptimalkan fungsi objektifnya agara prediksi lebih akurat dan model tidak mengalami *overfitting* yang telah dijelaskan pada persamaan 1. Pada setiap iterasi, *XGBoost* menghitung turunan pertama (gradient) dan turunan kedua (hessian) dari fungsi loss. Gradien $g_i = \frac{\partial L(y_i, y\hat{}i)}{\partial y\hat{}_i}$ digunakan untuk menentukan arah optimasi, sedangakan Hessian $h_i = \frac{\partial^2 L(y_i, y\hat{}i)}{\partial y\hat{}_i^{\,2}}$ digunakan untuk menghitung langkah optimal yang harus diambil dalam pembaruan parameter. Dengan menggunakan kedua informasi ini, model dapat menentukan seberapa besar perubahan yang perlu dilakukan untuk meningkatkan prediksi.

Setelah mendapatkan gradien dan Hessian, model memperbarui bobot pada setiap leaf node dalam pohon keputusan menggunakan persamaan sebagai berikut.

$$w^*_j = -\frac{\sum_{i \epsilon Ij} g_i}{\sum_{i \epsilon Ij} h_i + \lambda} \tag{2.3}$$

Penjelasan persamaan (3) sebagai berikut :

a. w^*_j= bobot optimal leaf node ke-j.

b. $\sum_{i\epsilon Ij} g_i$ = total gradien pada leaf node.

c. $\sum_{i\epsilon Ij} h_i$ = total Hessian pada leaf node.

d. λ = faktor regulasi.

Selain itu, *XGBoost* memilih split terbaik dalam pohon keputusan berdasarkan fungsi Gain. Fungsi ini menentukan pemisahan data terbaik dengan menghitung seberapa besar peningkatan informasi yang diperoleh setelah melakukan split. Persamaannya sebagai berikut.

$$Gain = \frac{1}{2}\left[\frac{\left(\sum_{i\epsilon I_L} g_i\right)^2}{\sum_{i\epsilon I_L} h_i + \lambda} + \frac{\left(\sum_{i\epsilon I_L} g_i\right)^2}{\sum_{i\epsilon I_L} h_i + \lambda} + \frac{\left(\sum_{i\epsilon I_L} g_i\right)^2}{\sum_{i\epsilon I_L} h_i + \lambda}\right] - \gamma \tag{2.4}$$

Penjelasan persamaan (4) sebagai berikut:

a. I_L dan I_R = data yang masuk ke cabang kiri dan kanan setelah split.

b. g_i dan h_i= gradien dan Hessian.

c. λ = regulasi bobot.

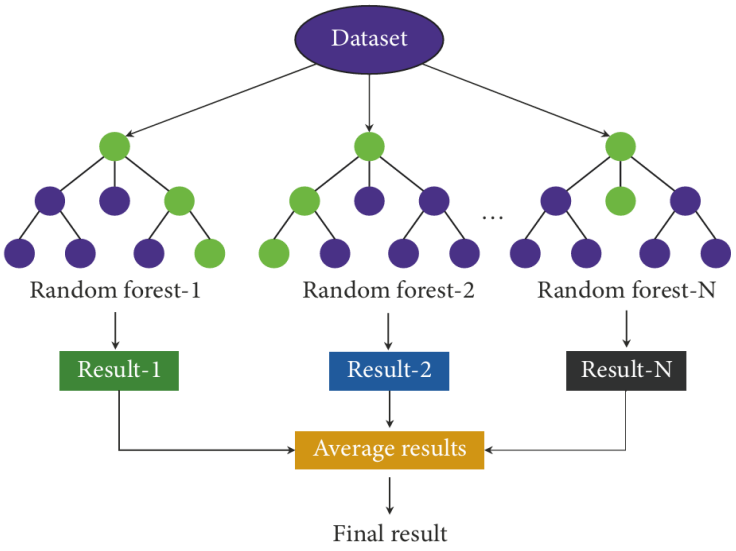
d. γ = penalti untuk jumlah leaf nodes.

Terakhir, proses ini berlangsung secara iteratif dengan setiap pohon baru menyesuaikan bobotnya berdasarkan kesalahan sebelumnya hingga model mencapai konvergensi. Setelah semua pohon dibangun, model akhirnya menghasilkan prediksi berdasarkan kombinasi dari semua pohon yang telah dibangun menghasilkan akurasi yang lebih tinggi.

2.3 *Random Forest*

Random Forest merupakan metode untuk klasifikasi dan regresi, keputusan yang dibuat oleh model *Random Forest* didasarkan pada keputusan yang dibuat oleh banyak pohon keputusan. *Random Forest* merupakan model yang di rekomendasikan untuk pemilihan komponen penting yang stabil. Salah satu keuntungan dari *Random Forest* adalah dapat membatasi overfitting tanpa mengurangi akurasi prediksi secara substansial (Zhou et al., 2020). *Flowchart* dari

Random Forest ditunjukan pada gambar 3.



Gambar 2. 3 *Flowchart Random Forest* (Fu & Qi, 2022).

Pada gambar 2.3, proses utama dalam *Random Forest* dimulai dengan mengambil dataset awal yang kemudian dibagi menjadi beberapa subset menggunakan teknik boosttrap resampling. Setiap subset digunakan untuk melatih model pohon keputusan yang berbeda, menghasilkan berbagai model yang memiliki variasi dalam cara mereka membagi data. Setelah setiap pohon keputusan selesai dilatih, lalu memberikan prediksi masing-masing terhadap data uji. Untuk klasifikasi, metode yang digunakan adalah voting mayoritas, dimana kelas yang paling sering dipilih oleh pohon-pohon keputusan dianggap sebagai hasil akhir. Selain itu dengan memilih fitur secara acak dalam setiap split, model ini dapat menangani data dengan jumlah fitur yang besar dan tetap mempertahankan keakuratan yang tinggi, Namun, kelemahannya adalah meningkatnya kompleksitas komputasi, terutama jika jumlah pohon yang digunakan terlalu besar (Fu & Qi, 2022).

Model *Random Forest* dibangun terutama berdasarkan dua metode yaitu bagging dan random subspace. *Bagging* merupakan proses mengambil sampel bootstrapping dan kemudian menggabungkan model yang dipelajari pada setiap sampel *bootstrapping*. *Bootstrapping* adalah metode resampling acak statistik dengan penggantian untuk menangani data yang tidak seimbang. *Random subspace*

merupakan memilih variabel atribut secara acak untuk membagi node setiap pohon keputusan dalam model *Random Forest*. Hal tersebut dapat mengontrol jumlah atribut yang digunakan untuk membagi node setiap pohon keputusan dalam model *Random Forest* (Zhou et al., 2020).

Setiap pohon dalam *Random Forest* memutuskan fitur mana yang akan digunakan dalam pemisahan data (splitting) menggunakan indeks Gini. Indeks Gini mengukur impurity (ketidakmurnian) suatu node yang dinyatakan dengan persamaan 5 berikut.

$$Gini(S_i) = 1 - \sum_{i-0}^{c-1} p^2{}_i \tag{2.5}$$

a. p_i = frekuensi relatif kelas C_i dalam satu set data.

b. c = jumlah total kelas dalam dataset

Semakin kecil nilai $Gini(S_i)$ semakin murni node tersebut yang berarti fitur tersebut lebih baik dalam membagi data. Setelah menghitung indeks Gini dari masing-masing node, pemisahan terbaik dipilih berdasarkan nilai Gini Split yang dirumuskan dalam persamaan 6.

$$Gini_{split} = \sum_{i=0}^{k-1} \left(\frac{n_i}{n}\right) Gini(S_i) \tag{2.6}$$

a. n_i = jumlah sampel dalam subset S_i $setelah\ spilt$.

b. n = jumlah total sampel dalam node yang sedang di evaluasi.

Pemisahan yang memiliki nilai Gini Split paling rendah akan dipilih untuk membentuk struktur pohon yang lebih baik. Dalam kasus klasifikasi seperti penyakit jantung, setiap pohon dalam *Random Forest* menghasilkan prediksi independen. Hasil akhir diperoleh dengan menggunakan voting mayoritas uang dirumuskan dalam persamaan 7 sebagai berikit.

$$\hat{y} = arg_k max \sum_{i=1}^{N} 1(h_i(x) = k) \quad (2.7)$$

a. $\hat{y}$ = prediksi akhir model

b. N = jumlah total pohon dalam *Random Forest*.

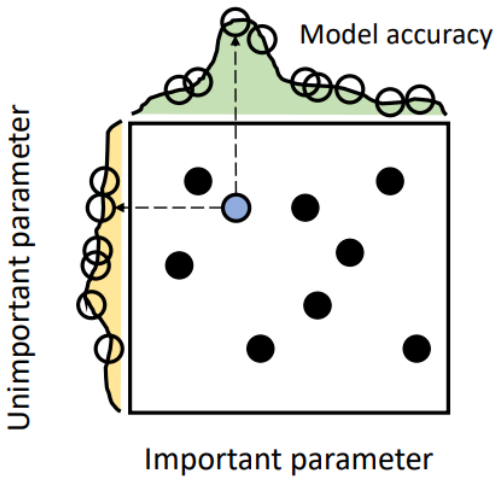
c. $h_i(x)$ = prediksi dari pohon ke-i untuk sampe X.

k = kelas yang di prediksi.

Hasil akhir model di tentukan oleh kelas yang paling banyak dipilih oleh pohon keputusan dalam *Random Forest* (Suci Amaliah et al., 2022).

2.4 *Randomized search*

Random search merupakan metode optimasi *hyperparameter* dimana nilai-nilai *hyperparameter* dipilih secara acak dari distribusi tertentu. Metode ini secara acak memilih beberapa titik dalam ruang *hyperparameter* dan mengevaluasi kinerjanya. *Random search* cocok dalam dimensi tinggi karena fungsi yang diinginkan memiliki dimensi praktis rendah dan lebih sensitif terhadap perubahan dalam beberapa dimensi daripada yang lain (Khotimah et al., 2024). Representasi *Random search* ditunjukkan pada gambar 2.4.



Gambar 2. 4 Representasi *Random search(Pilario et al., 2021).*

Pada gambar 2.4, sumbu horizontal mempresentasikan parameter yang penting, sedangkan sumbu vertikal menunjukan parameter yang kurang penting. Setiap titik dalam diagram melambangkan kombinasi parameter yang diuji dengan akurasi model sebagai hasil evaluasi. Keunggulan utama random search terlihat dari kemampuannya menemukan kombinasi parameter optimal dengan lebih efisien, seperti yang ditunjukkan oleh titik biru di zona dengan akurasi tinggi di zona hijau (Pilario et al., 2021).

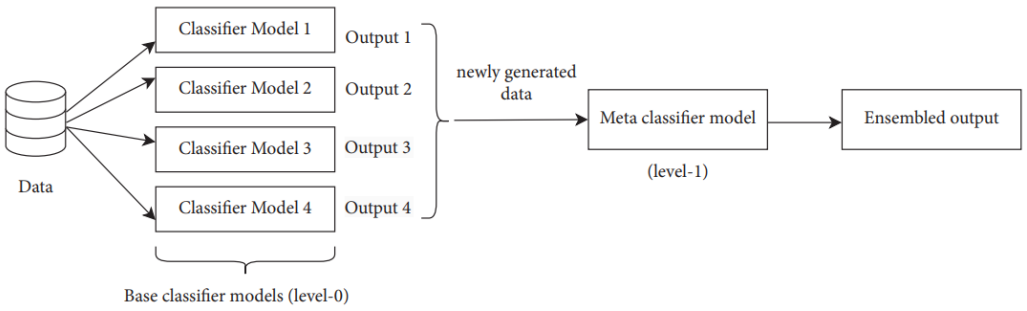
2.5 *Feature Importance*

Feature importance merupakan metode interpretasi model dalam *machine learning* yang digunakan untuk mengidentifikasi fitur mana yang paling mempengaruhi hasil prediksi. Dalam model *Random Forest*, *importance* diukur berdasarkan pengurangan impurity (Gini importance) pada setiap split. Pada *XGBoost, feature importance* biasanya dihitung berdasarkan rata-rata *gain*, yaitu peningkatan akurasi prediksi setelah fitur tersebut digunakan dalam pemisahan data. penggunaan *feature importance* tidak hanya memperjelas proses pengambilan keputusan dalam model, tetapi juga meningkatkan kepercayaan pengguna terhadap sistem berbasis *machine learning* (Cava et al., 2019).

2.6 *Stacking Ensemble*

Stacking ensemble merupakan *ensemble* yang terdiri dari base model dan meta-model dengan mempelajari dan menggabungkan prediksi. *stacking ensemble* dianggap sebagai *ensemble heterogen* yang mendorong keberagaman klasifikasi karena base leaner *stacking ensemble* biasanya menghasilkan yang berbeda dan beragam (Zian et al., 2021).

Konsep *stacking ensemble* ditunjukkan pada gambar 2.5.



Gambar 2. 5 Konsep *Stacking Ensemble* (Ganesan et al., 2022).

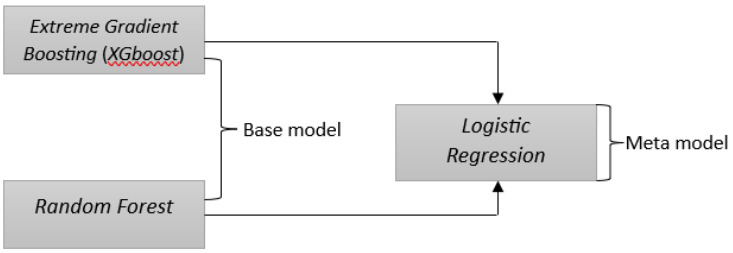
Pada gambar 2.5, teknik ini bekerja dengan membangun. dua level model klasifikasi yaitu level 0 (base model) dan level 1 (meta model). Pada tahap awal, dataset digunakan sebagai input untuk beberapa model klasifikasi dasar yang berbeda. Model bekerja secara paralel dan menghasilkan prediksi masing-masing.

Hasil prediksi dari setiap model dasar kemudian dikumpulkan dan digunakan untuk membentuk dataset baru. Dataset baru yang terdiri dari *output* base model kemudian diberikan ke meta model untuk mempelajari pola hubungan antar *output* dari model dasar akhir yang lebih akurat. Meta model menghasilkan *ensemble* output yaitu hasil akhir yang telah dioptimalkan dengan mempertimbangkan kontribusi dari base model. Kelebihannya mengurangi *overfitting* dan meningkatkan generalisasi model karena memanfaatkan dari berbagai model yang memiliki karakteristik yang berbeda (Ganesan et al., 2022).

2.7 *Logistic Regression*

Metode *ensemble* efektif dalam meningkatkan akurasi prediksi. Salah satu metode *ensemble* yang populer adalah *stacking*, dimana beberapa model dasar dilatih dan hasil prediksinya digabungkan dengan meta model. *Logistic Regression* sering digunakan sebagai meta model dalam *stacking ensemble* karena kesederhanaanya dan kemampuannya untuk memberikan interpretasi yang jelas terhadap hasil prediksi. *Logistic Regression* dikenal sebagai teknik *machine learning* untuk klasifikasi yang digunakan ketika variabel dependen bersifat biner yaitu 0 atau 1 dengan 1 menunjukan keberhasilan dan 0 menunjukan kegagalan. Model ini memprediksi probabilitas Y=1 berdasarkan variabel independen X dan

termasuk dalam analisis prediktif. *Logistic Regression* menggunakan sekelompok bobot yang dikenal sebagai koefisien (Chaurasia & Pal, 2021). Penggunaan *Logistic Regression* sebagai meta model dalam *stacking ensemble* ditunjukan pada gambar 2.6.



Gambar 2. 6 Konsep *Logistic Regression* (Chaurasia & Pal, 2021).

Pada gambar 2,7 , ditunjukkan bahwa *Logistic Regression* berperan sebagai meta-model dalam pendekatan *stacking ensemble*. *Logistic Regression* menerima input dari dua base model yaitu *Extreme Gradient Boosting (XGBoost)* dan *Random Forest* yang berfungsi sebagai model dasar dalam proses klasifikasi. Dalam prosesnya, *XGBoost* dan *Random Forest* terlebih dahulu dilatih menggunakan dataset asli, lalu hasil prediksi dari kedua model ini digunakan sebagai fitur baru untuk melatih *Logistic Regression*. Dengan kata lain, *Logistic Regression* tidak bekerja langsung dengan fitur asli dataset melainkan belajar dari pola prediksi yang dihasilkan oleh base models. Pemilihan *Logistic Regression* sebagai meta-model dalam gambar 2.7 bertujuan untuk menggabungkan informasi dari base models secara optimal. *Logistic Regression* mampu menangani output probabilistik dengan baik dan memiliki sifat yang lebih sederhana dibandingkan model non-linear lainnya, sehingga dapat mengurangi risiko *overfitting* pada meta-model.

2.8 Penelitian Terkait

Terdapat beberapa penelitian yang signifikan menggunakan berbagai pendekatan untuk memodelkan prediksi penyakit jantung. Kontribusi dan temuan penelitian memiliki potensi untuk menjadi dasar dalam pengembangan model prediksi yang tepat dan efisien dalam penyakit jantung. Berikut Tabel 2.1 penelitian

terkait menjadi landasan dalam penelotian ini.

Tabel 2.2 Penelitian Terkait.

No	Judul	Dataset	Parameter	Hasil
1.	Implementation of *Random Forest* and Extreme Gradient Boosting in the Classification of Heart Disease using Particle Swarm Optimization Feature Selection (Ansyari et al., 2023)	UCI Heart Disease Dataset.	Model *XGBoost*,*Random Forest+Particle Swarm Optimization*, Model uji 10-*Fold* Cross Validation dan evaluasi AUC – ROC	Model *XGBoost* tanpa PSO dengan nilai AUC – ROC: 0.877, *Random Forest* tanpa PSO dengan AUC – ROC : 0.874. Diterapkan PSO, nilai AUC – ROC *XGBoost* = 0.913, *Random Forest* meningkat menjadi 0.918.
2.	Performance Comparison of the SVM and SVM-PSO Algorithms for Heart Disease Prediction (Saputra et al., 2022)	UCI Heart Disease Dataset	*Support Vector Machine* (SVM), optimasi PSO, evaluasi dengan AUC	Optimasi parameter SVM dengan PSO meningkatkan akurasi dari 82,3% menjadi 84,81% dan AUC – ROC mencapai 0,898.
3.	Prediction of Heart Diseases using *Random Forest* (Pal & Parija, 2021)	Kaggle Heart Disease Dataset (303 sampel, 14 fitur)	*Random Forest*, 10-*Fold* cross-validation, evaluasi dengan akurasi, sensitivitas, spesifisitas, AUC	*Random Forest* menghasilkan akurasi 86.9%, sensitivitas 90.6%, spesifisitas 82.7%, dan AUC - ROC93.3%, menunjukkan bahwa metode ini efektif untuk klasifikasi penyakit jantung.
4.	Perbandingan Model Decision Tree, Naive Bayes dan *Random Forest* untuk Prediksi Klasifikasi Penyakit Jantung (Depari et al., 2022)	Personal Key Indicators of Heart Disease Dataset.	*Decision Tree, Naive Bayes*, *Random Forest*, evaluasi dengan precision, recall, F1-score	Model *Random Forest* mencapai akurasi 75%, lebih baik dibanding *Decision Tree* (72%) dan *Naive Bayes* (71%),
5.	Mendeteksi Penyakit Jantung Menggunakan *Machine learning* Dengan Model *Logistic Regression*	Kaggle Heart Disease UCI Dataset (303 data,	*Logistic Regression*, evaluasi dengan Confusion Matrix (Akurasi, Sensitivitas,	*Logistic Regression* memiliki sensitivitas tertinggi (88.54%) pada data training dan spesifisitas tertinggi (87.50%) pada data

	(Junifer Pangaribuan et al., 2021)	14 fitur)	Spesifisitas)	testing dibanding metode lainnya, membuktikan efektivitasnya dalam klasifikasi penyakit jantung.
6.	A Method for Improving Prediction of Human Heart Disease Using *Machine learning* Algorithms (Saboor et al., 2022)	Cleveland Heart Disease Dataset, StatLog, Z-Alizadeh Sani	*Random Forest*, *XGBoost*, *Decision Tree, SVM, Logistic Regression, Naïve Bayes, AdaBoost*, LDA, *GridSearchCV* untuk tuning	Model SVM mencapai akurasi 96.72% setelah tuning *hyperparameter*, sementara *XGBoost* dan Extra Trees memiliki akurasi tertinggi sebelum tuning. Data preprocessing dan standardisasi dataset meningkatkan akurasi model secara signifikan.
7.	Heart Disease Prediction using Hybrid *machine learning* Model (Kavitha et al., 2021)	Cleveland Heart Disease Dataset	Hybrid Model (*Random Forest* + *Decision Tree*), evaluasi dengan akurasi.	Hybrid Model (*Random Forest* + *Decision Tree*) mencapai akurasi 88.7%, lebih baik dibandingkan model individu.
8.	Prediction of Heart Disease Based on *Machine learning* Using Jellyfish Optimization Algorithm (Ahmad & Polat, 2023)	UCI Cleveland Heart Disease Dataset	SVM + *Jellyfish Optimization Algorithm* (JFO), Evaluasi dengan Sensitivity, Specificity, Accuracy, AUC.	Model dengan SVM + JFO mencapai Accuracy 98.47%, AUC – ROC 94.48%, lebih tinggi dibanding metode tanpa optimasi.
9.	An Empirical Evaluation of Stacked *Ensemble*s with Different Meta-Learners in Imbalanced Classification (Zian et al., 2021)	Dataset Imbalanced	*Stacked ensemble* dengan 19 Meta-Learner, Evaluasi dengan Precision, Recall, F1-Score	Model *Stacked ensemble* dengan meta-learner berbasis kombinasi bobot memberikan hasil lebih baik dalam klasifikasi dataset tidak seimbang. Model ini meningkatkan stabilitas prediksi dan performa dibandingkan *ensemble* konvensional seperti Bagging dan Boosting.

No	Penulis	Dataset	Parameter	Hasil
10.	Komparasi Deteksi Kecurangan pada Data Klaim Asuransi Pelayanan Kesehatan Menggunakan Metode Support Vector Machine (SVM) dan Extreme Gradient Boosting (*XGBoost*) (Nugraha & Irawan, 2023)	Data Klaim Asuransi Kesehatan	*XGBoost*, SVM, Fraud Detection	*XGBoost* menunjukkan performa klasifikasi terbaik dalam deteksi fraud layanan kesehatan, dengan Balanced Accuracy 0.9995 dan Recall 0.9994, dibandingkan dengan SVM.
11.	Peningkatan Keberagaman Data untuk Klasifikasi Penyakit Diabetes Berbasis *stacking ensemble* Learning (Majid et al., 2025)	BRFSS Diabetes Dataset.	*stacking ensemble* (SVM, *Logistic Regression*, MLP, RF	Kombinasi SMOTE-ENN dan *stacking ensemble* mencapai akurasi 97,3%, lebih baik dibanding model individu.
12.	A Comprehensive *stacking ensemble* Approach for Stress Level Classification in Higher Education (Fonda et al., 2024)	Data fisiologis (SPO2, heart rate, suhu tubuh, tekanan darah) dari mahasiswa di Universitas Hang Tuah Pekanbaru	SMOTE, *K-Fold Cross Validation* (K=10), *stacking ensemble* (SVM, *Logistic Regression*, MLP, RF) dengan *XGBoost* sebagai meta-model	Metode *stacking* mencapai akurasi 95% dan AUC - ROC0.95, menunjukkan peningkatan akurasi dalam klasifikasi tingkat stres mahasiswa.
13.	Penerapan *Machine learning* Model *Random Forest* Untuk Prediksi Penyakit Jantung (Putri et al., 2024)	Kaggle Heart Failure Prediction (918 record, 12 fitur)	*Random Forest*, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), *Logistic Regression*, SMOTE, Normalisasi	*Random Forest* menunjukkan performa terbaik dengan akurasi 87.7% pada data uji dan 92.63% pada data validasi, dibanding model lainnya seperti SVM dan KNN.

No	Penulis	Dataset	Parameter	Hasil
14.	Analisis Perbandingan Model *XGBoost* Dan Model *Random Forest* Untuk Klasifikasi Data Kesehatan Mental (Aditya et al., 2024)	Mental Health Dataset 292.364 data.	*XGBoost*, *Random Forest*, SMOTE, Normalisasi (MinMaxScaler)	*XGBoost* mencapai akurasi 99.82%, sedangkan *Random Forest* 99.04% dalam 30 kali percobaan.
15.	An improved approach to Arabic news classification based on *hyperparameter* tuning of *machine learning* algorithms (Jamaleddyn et al., 2023)	CNN Arabic Corpus (5070 dokumen teks)	NLP, *Machine learning*, *Hyperparameter* Tuning (Grid Search, *Random search*), MLR, SVM, ANN	*Random search* menunjukkan akurasi terbaik 95.16% dibanding Grid Search dengan 94.97%.
16.	Perbandingan Teknik Optimasi Grid Search dan *Randomized search* dalam Meningkatkan Akurasi Metode Klasifikasi SVM Pada Sentimen Ulasan Pengguna Aplikasi JKN Mobile (Agus Dendi Rachmatsyah, 2024)	Dataset ulasan pengguna aplikasi JKN Mobile (1.000 ulasan)	SVM, Grid Search, *Randomized search*, SMOTE, TF-IDF	*Randomized search* menghasilkan akurasi tertinggi 82% dibanding Grid Search 81.5%. SMOTE berhasil menyeimbangkan dataset.
17.	Usulan Penelitian	UCI Heart Disease Dataset Repository, 303 data 14 atribut.	Metode *stacking ensemble* (base model *XGBoost* dan *Random Forest*, meta model *Logistic Regression*), Optimasi *Randomized search*.	

Pada tabel 2.1, berbagai penelitian dalam prediksi penyakit jantung telah menguslkan berbagai model Penelitian oleh (Ansyari et al., 2023), (Saboor et al., 2022) menggunakan model *Random Forest*, *XGBoost*, SVM, *Logistic Regression*, *Naïve Bayes, AdaBoost*, LDA, kemudian penelitian dari (Ahmad & Polat, 2023) menggunakan SVM + *Jellyfish Optimization Algorithm* (JFO) dari beberapa

penelitian tersebut menunjukan performa baik dalam prediksi penyakit jantung ditunjukan dengan akurasi yang signifikan. Penelitian oleh (Agus Dendi Rachmatsyah, 2024) (Jamaleddyn et al., 2023) menambahkan optimasi *Random search* dengan menghasilkan hasil tertinggi dibandingkan dengan *Grid Search*. Mengkombinasikan beberapa model dengan esemble yang dilakukan oleh (Fonda et al., 2024) dengan SVM, *Logistic Regression*, MLP, RF dan meta model *XGBoost* dan penelitian oleh (Majid et al., 2025) SVM, *Logistic Regression*, MLP, RF dengan keduanya menggunakan *stacking ensemble* menghasilkan nilai akurasi yang signifikan. Meskipun demikian, beberapa penelitian belum terdapat yang mempertimbangkan beberapa fitur diatas di kombinasikan.

Oleh karena itu, penelitian ini merupakan salah satu upaya untuk mengisi kesenjanganyang masih ada dalam model prediksi penyakit jantung. Mengkombinasikan model *XGBoost*, *Random Forest* dan meta model *Logistic Regression* pada *stacking ensemble* dan optimasi *hyperparameter* dengan *Randomized search* dan PSO dalam upaya meningkatkan performa prediksi.

Perbandingan penelitian terkait dapat dilihat melalui matriks penelitian pada tabel 2.3.

Tabel 2. 3 Matrik Penelitian

No	Penulis	Model	Parameter			
			Seleksi Fitur	**Optimasi *Hyperparameter***	**Evaluasi AUC**	**Ensemble Learning**
1.	(Ansyari et al., 2023)	Model *XGBoost*,*Random Forest*+PSO	√	√	√	-
2.	(Depari et al., 2022)	Decision Tree, Naive Bayes, *Random Forest*	-	-	√	√
3.	(Saputra et al., 2022)	Support Vector Machine (SVM), optimasi PSO	-	√	√	-

No	Penulis	Model	Parameter			
			Seleksi Fitur	**Optimasi *Hyperparameter***	**Evaluasi AUC**	***Ensemble Learning***
4.	(Pal & Parija, 2021)	*Random Forest*	-	√	√	-
5.	(Junifer Pangaribuan et al., 2021)	*Logistic Regression*	-	-	√	-
6.	(Saboor et al., 2022)	*Random Forest*, *XGBoost*, Decision Tree, SVM, *Logistic Regression*, Naïve Bayes, AdaBoost, LDA, GridSearchCV.	-	√	√	-
7.	(Kavitha et al., 2021)	Hybrid Model (*Random Forest* + Decision Tree).	-	-	√	√
8.	(Ahmad & Polat, 2023)	SVM + Jellyfish	√	√	√	-
9.	(Zian et al., 2021)	*Stacked ensemble* dengan 19 Meta-Learner	-	√	√	√
10.	(Nugroho et al., 2023)	*XGBoost*, SVM, Fraud Detection	-	√	√	-

No	Penulis	Model	Parameter			
			Seleksi Fitur	**Optimasi *Hyperparameter***	**Evaluasi AUC**	***Ensemble Learning***
11.	(Majid et al., 2025)	SVM, *Logistic Regression*, MLP, RF.	√	√	√	-
12.	(Fonda et al., 2024)	SVM, *Logistic Regression*, MLP, RF) dengan *XGBoost*	-	√	√	√
13.	(Putri et al., 2024)	*Random Forest*, KNN, Support Vector Machine (SVM), *Logistic Regression*	√	√	√	-
14.	(Aditya et al., 2024)	*XGBoost*, *Random Forest*, SMOTE	√	√	√	-
15.	(Jamaleddyn et al., 2023)(Agus Dendi Rachmatsyah, 2024)	(Grid Search, *Random search*), MLR, SVM, ANN	√	√	√	-
16.	(Agus Dendi Rachmatsyah, 2024)	SVM, Grid Search, *Randomized search*, SMOTE, TF-IDF	√	√	√	-

No	Penulis	Model	Parameter			
			Seleksi Fitur	Optimasi *Hyperparameter*	Evaluasi AUC	*Ensemble Learning*
17.	Usulan Penelitian, 2025	*XGBoost*, *Random Forest*, *Logistic Regression*, Randomzide Search.	-	√	√	√

Pada Tabel 2.3, matrik penelitian menunjukkan perbedaan di antara penelitian terkait lainnya dalam prediksi penyakit jantung menggunakan metode *machine learning* dan optimasi model. Penelitian yang dilakukan oleh (Ansyari et al., 2023) menggunakan model *XGBoost* dan model *Random Forest* untuk klasifikasi penyakit jantung, serta menerapkan *Particle Swarm Optimization (PSO)* sebagai metode seleksi fitur. Hasilnya menunjukkan peningkatan performa model setelah fitur yang kurang relevan dieliminasi, dengan nilai AUC *XGBoost* mencapai 0,913 dan model *Random Forest* sebesar 0,918. Meskipun demikian, penelitian tersebut belum menerapkan pendekatan ensemble learning seperti stacking, sehingga keunggulan dua model tersebut belum dikombinasikan secara optimal. Selain itu, optimasi hyperparameter belum dilakukan secara eksplisit, proses tuning parameter sangat penting untuk meningkatkan generalisasi dan stabilitas prediksi.

Berdasarkan hal tersebut, penelitian ini mengatasi kekurangan tersebut dengan mengembangkan metode *stacking ensemble* yang menggabungkan model XGBoost dan Random Forest sebagai base model, dan model Logistic Regression sebagai meta-model. Metode stacking dipilih karena kemampuannya menggabungkan output prediksi dari berbagai model dasar untuk menghasilkan keputusan yang lebih kuat dan stabil. Guna menyempurnakan kinerja masing-masing model dasar, dilakukan optimasi hyperparameter menggunakan *Randomized Search*, yang secara efisien mencari kombinasi parameter terbaik dalam ruang pencarian yang luas.

Penggunaan Randomized Search ini diharapkan mampu mengatasi permasalahan *overfitting* serta meningkatkan performa model tanpa membebani proses komputasi secara berlebihan.

Kombinasi metode stacking ensemble dan optimasi hyperparameter, penelitian ini bertujuan membangun model prediksi penyakit jantung yang tidak hanya akurat, tetapi juga lebih stabil dibanding model individu.