

BAB II

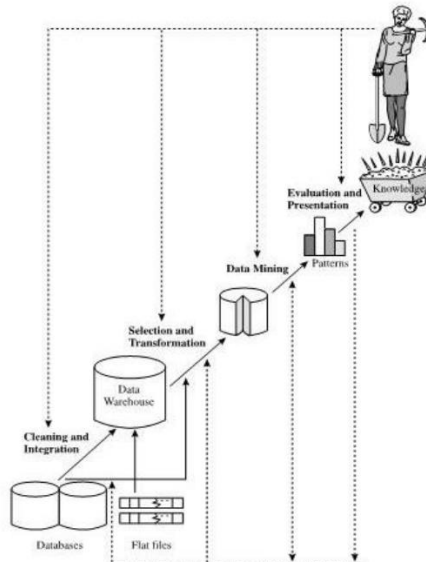
LANDASAN TEORI

2.1 Data Mining

Data mining adalah suatu algoritma didalam menggali informasi berharga yang terpendam atau tersembunyi pada suatu koleksi data/ database yang sangat besar sehingga ditemukan suatu pola yang menarik yang sebelumnya tidak diketahui. Analisa data mining berjalan pada data yang cenderung terus membesar dan teknik terbaik yang digunakan kemudian berorientasi kepada data berukuran sangat besar untuk mendapatkan kesimpulan dan keputusan paling layak [3]. Data mining memiliki beberapa sebutan atau nama lain yaitu: *Knowledge discovery (mining) in databases (KDD)*, ekstraksi pengetahuan (*knowledge extraction*), analisa data/pola, kecerdasan bisnis (*business intelligence*).

Data mining data adalah suatu proses ekstraksi informasi dari database yang besar, dan dalam pelaksanaannya, melibatkan berbagai teknik seperti statistik, matematika, kecerdasan buatan, dan pembelajaran mesin. Data mining juga dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan. Istilah data mining kadang disebut juga *knowledge discovery* [4]. Salah satu teknik yang dibuat dalam data mining adalah bagaimana menelusuri data yang ada untuk membangun sebuah model, kemudian menggunakan model tersebut agar dapat mengenali pola data yang lain yang tidak berada dalam basis data yang tersimpan. Kebutuhan untuk prediksi juga dapat memanfaatkan teknik ini. Dalam data mining, pengelompokan data juga bisa dilakukan. Tujuannya adalah agar kita dapat mengetahui pola universal data-data yang ada. Anomali data transaksi juga perlu dideteksi untuk dapat mengetahui tindak lanjut berikutnya yang dapat diambil. Semua hal tersebut bertujuan mendukung kegiatan operasional perusahaan sehingga tujuan akhir perusahaan diharapkan dapat tercapai. Data mining merupakan bagian dari proses *Knowledge*

Discovery from Data (KDD). Dibawah ini digambarkan skema dari proses KDD [5].



Gambar 2.1 Data mining sebagai dari proses knowledge discovery

2.1.1 Fungsi Data Mining

1. *Prediction*

Fungsi data mining yang pertama adalah prediksi atau *prediction*. Ini adalah proses untuk menemukan pola dari data dengan dan juga menggunakan beberapa variabel untuk memprediksikan variabel lainnya yang nilai atau jenisnya masih tidak diketahui [6].

2. *Description*

Berikutnya data fungsi deskripsi atau *description*. Ini adalah proses untuk menemukan suatu ciri krusial dari data yang terdapat di dalam suatu *database* atau basis data [5].

3. *Classification*

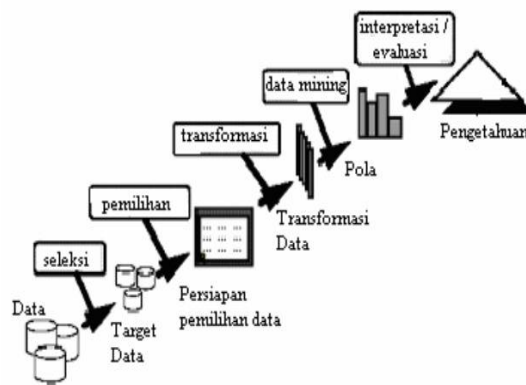
Klasifikasi atau *classification* adalah suatu proses untuk menemukan contoh atau fungsi agar dapat menggambarkan grup atau konsep dari suatu data. Proses yang digunakan untuk menggambarkan data tersebut adalah hal yang penting serta juga bisa memprediksi kecenderungan data yang terdapat pada masa depan (mendatang).

4. *Association* (Asosiasi)

Yang terakhir adalah asosiasi atau *association*. Ini adalah proses yang dipakai untuk menemukan suatu hubungan yang terdapat pada nilai atribut daripada sekumpulan data [7].

2.1.2 Tahapan Proses Data Mining

Ada beberapa tahapan dalam proses data mining. Diagram dibawah ini menggambarkan beberapa tahap/proses yang berlangsung dalam data mining. Fase awal dimulai dari data sumber dan berakhir dengan adanya informasi yang dihasilkan dari beberapa tahapan, yaitu:



Gambar 2.2 Fase – Fase Dalam Data Mining

1. Seleksi Data

Pemilihan (seleksi) data baru sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam data mining dimulai. Data hasil seleksi yang akan digunakan untuk proses data mining, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing/Cleaning* (pemilihan data)

Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data.

a. *Data Understanding*

Proses dimulai dengan memahami struktur dan tipe data yang dimiliki. Yaitu mencakup pemeriksaan bentuk dataset, jenis variabel, (numerik atau kategori) serta melihat ringkasan statistic dari dataset.

b. Missing Values

Memeriksa apakah terdapat nilai yang hilang atau null dan nilai duplikat pada data, untuk itu pada tahapan ini diperlukan identifikasi *missing values* untuk dapat meningkatkan akurasi kinerja model yang dapat guna meningkatkan keakuratan data.

c. Outlier

Pada penelitian ini pemeriksaan outlier menggunakan visualisasi boxplot dan *IQR (Inter Quartile Range)* digunakan untuk mendeteksi *outlier* dalam data dengan cara menentukan batas bawah (*lower*) dan batas (*upper*) diluar mana data yang dianggap sebagai *outlier*.

d. Univariate Analysis

Pada tahap ini analisis univariate, adalah tahapan untuk langkah memahami distribusi dan karakteristik dari masing-masing variabel dalam dataset.

e. Multivariate Analysis

Tahapan menganalisis hubungan antar variabel numerik dalam dataset.

3. Transformasi

Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses data mining.

4. Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik metode atau algoritma dalam data mining sangat bervariasi. Pemilihan metode dan algoritma sesuai dengan kebutuhan dan tujuan [8].

5. Interpretasi/Evaluasi

Pola informasi yang dihasilkan dari informasi data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan.

2.2 Machine Learning

Machine Learning (ML) merupakan bidang studi yang fokus kepada desain dan analisis algoritma sehingga memungkinkan komputer untuk dapat belajar. Menurut Samuel, ML berisi sebuah algoritma yang bersifat generik (umum)

dimana algoritma tersebut dapat menghasilkan sesuatu yang menarik atau bermanfaat dari sejumlah data tanpa harus menulis kode yang spesifik [7]. Pada intinya, algoritma yang generik tersebut ketika diberikan sejumlah data makai dapat membangun sebuah aturan atau model atau inferensi dari data tersebut. Sebagai contoh sebuah algoritma untuk mengenali tulisan tangan dapat digunakan untuk mendeteksi email yang berisi spam dan bukan spam tanpa mengganti kode. *Machine learning* telah menjadi alat penting dalam analisis data medis dan prediksi penyakit algoritma yang sama ketika diberikan data pelatihan yang berbeda menghasilkan logika klasifikasi yang berbeda [9].

Algoritma *machine learning* telah menunjukkan dengan demikian, penelitian ini tidak hanya potensi besar dalam prediksi penyakit medis, termasuk berkontribusi pada literatur yang ada dengan pendekatan penyakit jantung, dengan memanfaatkan data kesehatan komparatif yang dilakukan, tetapi juga memiliki potensi yang beragam. Algoritma *machine learning*, seperti *Random Forest* dan *Decision Tree* menawarkan solusi yang potensial. Namun, meskipun banyak penelitian telah dilakukan di bidang ini, masih ada kebutuhan untuk mengevaluasi dan membandingkan kinerja berbagai algoritma tersebut dalam konteks prediksi penyakit jantung. Penelitian ini bertujuan untuk mengembangkan model prediksi penyakit jantung yang akurat dengan menggunakan dua algoritma *machine learning* yang berbeda [8].

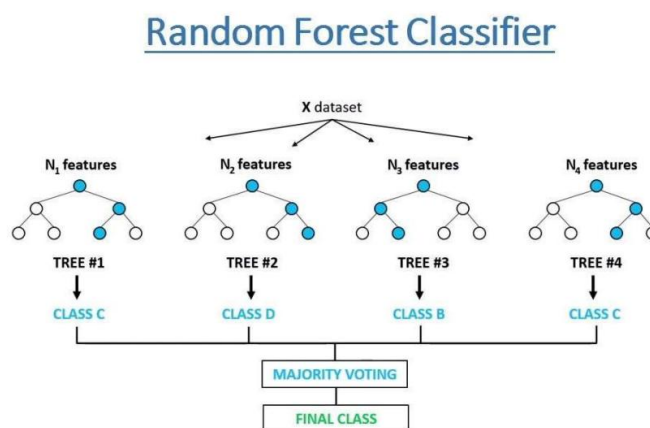
2.3 Penyakit Jantung

Penyakit jantung adalah salah satu masalah kesehatan utama yang menyebabkan kematian terbanyak di dunia berdasarkan beberapa faktor-faktor risiko seperti tekanan darah tinggi, diabetes, obesitas, gaya hidup tidak sehat, merokok, dan faktor genetic yang dapat menyebabkan resiko terkena penyakit jantung [10]. Serangan jantung terjadi akibat terhambatnya aliran darah menuju jantung sehingga suplai oksigen dan nutrisi di otot jantung dan jaringan disekitar jantung berkurang. Tidak seperti otot tubuh lainnya, otot jantung tidak memiliki kemampuan beregenerasi. Apabila terdapat saja kerusakan maka akan berakibat fatal bagi tubuh. Semakin lama serangan jantung terjadi semakin banyak pula kerusakan pada jantung. Karena itu penting bagi kita

untuk mengenali gejala-gejala dari penyakit jantung sehingga dapat memberikan pertolongan dengan segera [2]. Penyakit jantung juga merupakan penyebab kematian utama, termasuk penyakit jantung koroner, serangan jantung, dan gagal jantung. Identifikasi dan prediksi risiko penyakit jantung penting untuk pengobatan yang tepat dan pencegahan komplikasi. Algoritma pembelajaran mesin mampu menganalisis data medis untuk menemukan pola yang berhubungan dengan risiko penyakit jantung dan memberikan rekomendasi klinis yang lebih baik [11].

2.4 Random Forest

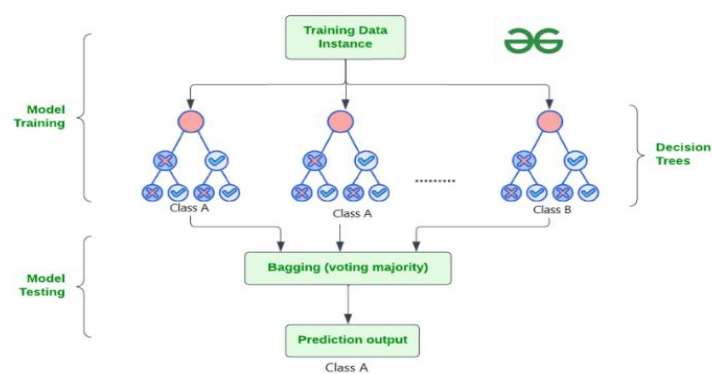
Metode *Random Forest* adalah salah satu algoritma *machine learning* yang termasuk dalam kategori *ensemble learning*. *Ensemble learning* melibatkan penggabungan hasil dari beberapa model untuk meningkatkan kinerja dan ketepatan prediksi dibandingkan dengan penggunaan satu model tunggal. Dalam konteks *Random Forest*, model yang digunakan adalah pohon keputusan (*decision trees*) [5].



Gambar 2.3 Gambar Random Forest

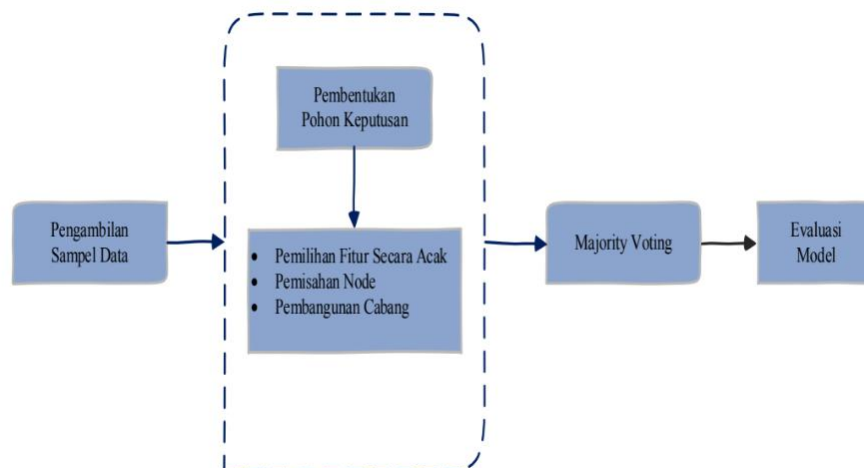
Metode *Random Forest* merupakan metode yang berbasis pada pohon keputusan, pada saat pelatihan *Random Forest* akan dibuat banyak pohon keputusan sehingga dari sampel yang ada pada set pelatihan akan menghasilkan beberapa pohon. *Random Forest* memerlukan kombinasi beberapa pohon keputusan untuk memprediksi hasil secara akurat. Saat

menggunakan *random forest* sebagai pengklasifikasi, setiap pohon keputusan dapat menghasilkan jawaban yang sama atau berbeda. Misalnya pohon keputusan A, B, E dan F memprediksi hasil 1. Sedangkan pohon keputusan C dan D memprediksi hasil 0. Karena banyak alternatif jawaban di pohon keputusan dan probabilitasnya tinggi maka random forest mengambil hasil prediksi tersebut. Hasil dari beberapa pohon keputusan berdasarkan suara mayoritas dan prediksi hasil yang lebih akurat.



Gambar 2.4 Algoritma Random Forest

Berikut tahapan cara kerja metode *random forest*:



Gambar 2.5 Cara Kerja Algoritma Random Forest

1. Pengambilan Sampel Data

Random Forest mulai dengan mengambil sampel acak dari dataset pelatihan asli untuk membentuk beberapa subset data.

2. Pembentukan Pohon Keputusan

Untuk setiap subset data, sebuah pohon keputusan dibangun. Proses pembentukan pohon melibatkan langkah-langkah berikut:

a) Pemilihan Fitur secara Acak

Di setiap *node* (simpul) dari pohon, sejumlah fitur dipilih secara acak dari semua fitur yang tersedia. Hal ini berbeda dari pohon keputusan biasa yang menggunakan semua fitur untuk memecahkan *node*.

b) Pemisahan *Node*

Dari fitur-fitur yang dipilih secara acak, algoritma mencari fitur dan titik pemisahan terbaik yang memaksimalkan pemisahan data berdasarkan kriteria tertentu menggunakan Gini Index untuk klasifikasi.

$$\begin{aligned}\text{Gini Index} &= 1 - \sum_{i=1}^n (P_i)^2 \\ &= 1 - [(P_+)^2 + (P_-)^2]\end{aligned}$$

Keterangan:

n = Jumlah dari masing-masing atribut

P_i = jumlah atribut dari masing-masing kelas atau labelnya

P_+ = Probabilitas Positif Class

P_- = Probabilitas Negatif Class

Untuk mengukur seberapa sering elemen yang dipilih secara acak dari kumpulan data, Gini Index melakukan pemisahan optimal antara simpul akar dan simpul berikutnya.

c) Pembangunan Cabang

Node tersebut dipecah menjadi dua cabang, dan proses ini berulang sampai pohon mencapai kedalaman tertentu atau *node* tidak bisa dipecah lagi (misalnya, semua data dalam *node* adalah homogen).

3. *Majority Voting*

Setelah semua pohon dalam hutan terbentuk, *Random Forest* membuat prediksi akhir dengan cara klasifikasi. Klasifikasi setiap pohon memberikan suara (prediksi) untuk kelas tertentu, dan kelas dengan suara terbanyak di antara semua pohon dipilih sebagai prediksi akhir.

4. Evaluasi Model

Random Forest dievaluasi berdasarkan kinerja prediksinya pada data uji yang tidak digunakan dalam pelatihan. Teknik ini melibatkan pengukuran *akurasi*, *precision*, *recall*, dan *F1-score*.

Tahapan proses perhitungan algoritma random forest dalam penelitian ini adalah sebagai berikut:

- a. Model akan mengambil 100 subset data secara acak dan mungkin terdapat sampel yang bisa terpilih lebih dari sekali dan bisa saja ada yang tidak terpilih.
- b. Pada setiap subset, model akan membangun 1 *decision tree* dengan mencapai kedalaman maksimal 10.
- c. Kemudian internal *node* akan ditemukan dengan mencari nilai *weighted gini index* terkecil. Rumus untuk mendapatkan *weighted gini index* terkecil, yaitu $\frac{N_1}{N} gini_1 + \frac{N_2}{N} gini_2$. Untuk mendapatkan *gini* dari rumus tersebut yaitu dengan menggunakan rumus $gini = 1 - \sum_{i=1}^k p_i^2$.
- d. Untuk mendapatkan variabel *y* prediksi, maka akan dilakukan voting dari setiap pohon. Jika mayoritas pohon memprediksi 1, maka random forest akan menghasilkan angka 1.

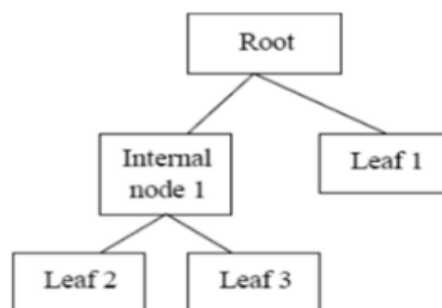
2.5 Decision Tree

Decision Tree merupakan salah satu metode dalam data mining yang menggunakan struktur pohon untuk membuat model prediksi. Metode ini bekerja dengan cara memecah dataset menjadi subset yang lebih kecil berdasarkan kriteria pemisahan tertentu [12]. Pohon keputusan adalah salah satu jenis algoritma penambangan data yang paling populer untuk klasifikasi dan prediksi. *Decision Tree* mengatur catatan dalam struktur pohon yang terdiri

dari simpul akar, cabang, dan simpul daun. *Node* akar berada di bagian atas struktur pohon. *Node* mewakili atribut, cabang mewakili hasil, lalu daun mewakili keputusan [13].

Setiap *node* dalam pohon mewakili sebuah pertanyaan tentang atribut, sedangkan cabang mewakili jawaban dari pertanyaan tersebut. *Decision Tree* juga unggul dan memiliki keunggulan dalam hal interpretabilitas karena dapat menggambarkan aturan keputusan secara visual dan dapat mudah dipahami. Konsep dasar pada metode *decision tree* adalah merubah sebuah data menjadi suatu keputusan yang berbentuk pohon dengan beberapa aturan pengambilan keputusan. Manfaat dari penerapan metode ini adalah kemampuan dalam menyederhanakan pengambilan sebuah keputusan yang bersifat kompleks, sehingga solusi yang didapat dari pengambilan keputusan tersebut nantinya akan lebih menginterpretasikan permasalahan dari studi kasus yang diambil. *Decision tree* juga dapat disebut sebagai struktur pada analisis pemecahan suatu masalah, serta dapat dijadikan sebagai sarana pemetaan alternatif dalam memecahkan suatu masalah [14].

Proses awal untuk membangun tree dimulai dengan data yang berada pada simpul akar (*root node*) yang kemudian dilanjutkan dengan langkah selanjutnya yaitu pemilihan atribut, perumusan uji logika (*logical test*) pada atribut yang sudah dipilih, serta percabangan pada setiap hasil pengujian dari tes tersebut.



Gambar 2.6 Proses Decision Tree

Proses pada *decision tree* terdiri dari akar (*root*), *internal node*, dan *leaf* sebagaimana yang ditunjukkan oleh gambar 2.6.

1. *Root* merupakan *node* teratas pada *decision tree* yang tidak memiliki *input*, dapat menghasilkan output lebih dari satu atau tidak menghasilkan *output* sama sekali
2. *Internal Node* merupakan *node* percabangan, dimana pada *node* ini hanya terdapat satu *input*, dengan *output* yang dihasilkan minimal dua.
3. *Leaf* merupakan *node* terakhir atau terminal *node* pada proses *decision tree*, dimana pada *node* ini hanya terdapat satu *input*, tanpa menghasilkan *output*.

Dilihat dari segi fungsionalnya *decision tree* merupakan salah satu metode data mining yang merepresentasikan bentuk pohon (*tree*) dengan tujuan untuk menentukan aturan pada proses klasifikasi. Salah satu kelebihan *decision tree* yaitu memiliki model yang sederhana dan mudah dipahami karena ditampilkan dalam bentuk pohon yang bercabang dan mendapatkan nilai akurasi yang tinggi [15].

Rumus dasar *Decision Tree*:

- 1) Entropy untuk mengukur impurity dataset:

$$H(S) = - \sum_{i=1}^n P_i \log_2(P_i)$$

$H(S)$: Entropy dataset

P_i : probabilitas data berada dikelas i

- 2) Information Gain (IG) untuk menentukan dan memilih atribut terbaik (berdasarkan pengurangan Entropy):

$$IG(S, A) = H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v)$$

$IG(S, A)$: Information Gain untuk atribut A .

$H(S)$: Entropy dataset awal.

$H(SV)$: Entropy subset S_v setelah dataset dibagi berdasarkan atribut A .

- 3) Gini Impurity untuk mengukur impurity dataset (alternatif dari Entropy):

$$Gini(S) = 1 - \sum_{i=1}^n P_i^2$$

$Gini(S)$: Gini Impurity dataset S .

P_i : probabilitas data berada dikelas i

2.5 Metrik Evaluasi

Pada tahap ini, *Confusion matrix* digunakan untuk menilai kinerja model pembelajaran mesin yang akan dibuat. Matriks ini memungkinkan untuk dapat mengevaluasi seberapa akurat dalam mengklasifikasikan kumpulan data dengan memberikan ringkasan hasil prediksi model pada sebuah dataset [16].

		Actual Values	
		1 (Postive)	0 (Negative)
Predicted Values	1 (Postive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 2.7 *Confusion matrix*

Keterangan:

1. TP (*True Positives*): Model memprediksi jumlah kasus yang benar- benar positif.
2. FP (*False Positives*): Jumlah kasus sebenarnya negatif, tetapi model memprediksinya positif.
3. FN (*False Negatif*): Model memprediksinya jumlah kasus sebagai negatif tetapi sebenarnya positif.
4. TN (*True Negatif*): Jumlah kasus benar-benar negatif dan diprediksi oleh model sebagai negatif.

Dengan *True Positive* (TP), ini adalah jumlah prediksi data positif tergolong positif. Positif palsu (FP) adalah banyaknya prediksi data negatif tergolong nilai positif. *False Negative* (FN) adalah banyaknya prediksi data positif yang tergolong nilai negatif. Angka negatif sebenarnya (TN) adalah banyaknya prediksi data negatif tergolong nilai negatif. Berdasarkan *confusion matrix*, metrik kinerja algoritme dapat dihitung menggunakan *accuracy*, *recall*, *precision*, dan *F1-score*.

Nilai *accuracy*, *precision*, *recall*, *F1-score*, serta akurasi disetiap fold akan diketahui melalui metode *Confusion Matrix*.

1. *Accuracy*

Akurasi adalah probabilitas bahwa nilai kelas dalam suatu klasifikasi dapat dikenali dengan benar, mencakup seluruh nilai dari kelas tersebut [17].

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

2. *Precision*

Precision adalah rasio antara jumlah sampel yang diklasifikasikan dengan benar sebagai positif dan total sampel yang telah dikategorikan sebagai positif.

$$precision = \frac{TP}{TP + FP}$$

3. Recall

Recall mengukur jumlah sampel yang benar-benar diklasifikasikan sebagai positif dan digunakan untuk menilai sejauh mana data dapat diidentifikasi dengan akurat[18].

$$Recall = \frac{TP}{TP + FN}$$

4. F1-Score

F1-score memiliki rentang nilai antara 0 hingga 1; jika mencapai nilai 1, itu berarti klasifikasi yang dihasilkan adalah sempurna[11].

$$F1\ score = 2 \times \frac{recall \times precision}{recall + precision}$$

2.6 Google Colaboratory

Google Colaboratory merupakan *platform* pengembangan berbasis *cloud* yang memungkinkan pengguna untuk melakukan pemrograman dan analisis data menggunakan *python*. Secara lebih teknis, colab merupakan layanan *notebook jupyter* yang dihosting dan dapat digunakan tanpa penyiapan, serta menyediakan akses gratis ke *resource* komputasi [19]. *Google Colab* juga dapat menawarkan kemampuan untuk menjalankan *Jupyter Notebook* (web app open- source untuk kombinasi kode, teks terformat, dan visualisasi data) langsung juga dari web browser tanpa perlu konfigurasi apa pun.

2.7 Kaggle

Kaggle adalah sebuah platform daring yang sangat dikenal di kalangan ilmuwan data dan mereka yang tertarik dengan *machine learning* [20]. Platform ini menyediakan beragam fasilitas, seperti koleksi data yang melimpah, ajang kompetisi *machine learning*, serta forum komunitas yang

ramai. Para pengguna dapat mengikuti kompetisi untuk menguji keahlian mereka, belajar dari sesama ilmuwan data, dan bahkan meraih hadiah. Tak hanya itu, *Kaggle* juga menyajikan aneka kursus dan tutorial tentang *machine learning*, menjadikannya tempat yang ideal bagi pemula maupun pakar.

2.8 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

No	Judul Penelitian	Metode dan Jumlah Dataset	Tujuan	Penulis dan Tahun	Hasil
1.	Perbandingan Algoritma K-Nearest Neighbors (K-NN) dan Random Forest Terhadap Penyakit Gagal Jantung	Algoritma K-Nearest Neighbors (K-NN) dan Random Forest. / Dataset 299	Bertujuan untuk membandingkan akurasi algoritma K-Nearest Neighbor (K-NN) dan Random Forest dalam mengklasifikasi penyebab penyakit gagal jantung.	Fredilio, Julfikar Rahmad, Stiven Hamona ngan Sinurat, Evta Indra / 2023	Berdasarkan hasil dari analisis dua algoritma K-Nearest Neighbor (K-NN) dan algoritma Randomforest, maka diperoleh hasil akurasi terbaik dari algoritma Random Forest sebesar 96,5 %
2.	Implementasi Algoritma	Random Forest. /	Dengan menggunak	Ary Prandika	Hasil model penyesuaian

	Random Forest Dalam Klasifikasi Diagnosis Penyakit Stroke	Dataset 40.910	an Metode Random Forest di harapkan bisa menjadi pilihan tepat dalam melakukan preprosessing data dalam mengidentifikasi gejala awal.	Siregar, Dwi Priyadi Purba, Jojor Putri Pasaribu, Khairul Reza Bakara/ 2023	menghasilkan 96% skor pelatihan dan dari hasil precision, recall, F1-score, dan accuracy yang mendapatkan hasil akurasi sebesar 0.95 atau 95%, serta hasil akhir dari AUC sebesar 0.80 yang menunjukkan hasil model tersebut termasuk ke dalam klasifikasi baik.
3.	Prediksi Penyakit Diabetes Melitus Menggunakan Algoritma Machine Learning	Random Forest, Regresi Logistik, dan Decision Tree. / Dataset 768	Bertujuan untuk mengembangkan model prediksi yang akurat untuk diabetes melitus	Egi Safitri, Dani Rofianto, Neni Purwati, Hendra Kurniawan, Sri	Hasil penelitian menunjukkan bahwa Regresi Logistik mencapai akurasi tertinggi (75%) dan kinerja

			menggunakan tiga algoritma machine learning: Random Forest, Regresi Logistik, dan Decision Tree	Karnila / 2024	yang seimbang dalam mengidentifikasi kasus positif dan negative. Decision Tree unggul dalam recall, sementara Random Forest menunjukkan keseimbangan yang sedikit lebih rendah antara presisi dan recall. Analisis kurva ROC mengungkapkan bahwa Random Forest memiliki AUC tertinggi (0.82), diikuti oleh Regresi Logistik (0.81) dan Decision Tree (0.73).
4.	Implementasi Data Mining Menggunakan	Random Forest dan Support	Bertujuan untuk membandin	Anggita Ghozali, Hasil	Dari hasil penelitian didapatkan

	n Metode Random Forest dan Support Vector Machine Dalam Klasifikasi Penyakit Diabetes	Vector Machine. / Dataset 520	gkan akurasi algoritma Random Forest dan Support Vector Machine dalam mengklasifikasi penyebab penyakit diabetes.	Pratiwi / 2023	bahwa algoritma Random Forest dengan split data 80%:20% mendapatkan hasil terbaik dengan akurasi yang didapatkan sebesar 0,98, presisi sebesar 0,96, recall sebesar 1, specificity sebesar 0,95, dan F1-score sebesar 0,98.
5.	Perbandingan Prediksi Penyakit Hipertensi Menggunakan Metode Random Forest dan Naïve Bayes	Random Forest dan Naïve Bayes. / Dataset 3000	Bertujuan untuk membandingkan hasil prediksi menggunakan dua metode algoritma Random Forest dan Naïve Bayes	Kharits Abdul Khalim / 2023	Penelitian tersebut telah di evaluasi menggunakan confusion matrix yang menggunakan tiga rumus yaitu Accuracy, Precision dan Recall yang mendapatkan

			mengunaka n softwere orange.		nilai masing masing, Naïve Bayes mendapatkan nilai Accuracy 85,9%, Precision 85,5% dan Recall 82,6%. Sedangkan Random Forest mendapatkan nilai 100% dari ketiga rumus yaitu Accuracy, Precision dan Recall. Dalam penelitian ini membuktikan bahwa Random Forest lebih unggul dari Naïve Bayes dalam menentukan prediksi hipertesi yang mendapatkan niaai
--	--	--	------------------------------------	--	--

					keakuratan 100%.
6.	Implementasi Prediksi Penyakit Diabetes Menggunakan Metode Decision Tree	Decision Tree. / Dataset 768	Menghasilkan prototipe tool untuk memprediksi penyakit diabetes	Lulu Safitri, Zaehol Fatah / 2023	Model berhasil mengidentifikasi 350 kasus non-diabetes dan 188 kasus diabetes dengan benar. Penelitian ini membuktikan bahwa metode Decision Tree dapat digunakan sebagai alat bantu dalam prediksi awal penyakit diabetes, meskipun masih memerlukan optimasi lebih lanjut untuk

					meningkatkan akurasi prediksi.
7.	Perbandingan Tingkat Akurasi Algoritma Decision Tree dan Random Forest Dalam Mengklasifikasi Penerima Bantuan Sosial BPNT di Desa Slangit	Algoritma Decision Tree dan Random Forest. / Dataset 936	Bertujuan untuk membandingkan dua algoritma Decision Tree dan Random Forest Mengklasifikasi Penerima Bantuan Sosial BPNT di Desa Slangit	Aldiyansyah, Ade Purnamasari, Irfan Ali / 2024	Evaluasi menunjukkan tingkat akurasi tinggi, yakni 97.86% untuk Decision Tree dan 97.86% untuk Random Forest dan mendapatkan rekomendasi penelitian ini mencakup pertimbangan terhadap metode alternatif serta perlunya penelitian lanjutan untuk memahami faktor-faktor sosial, ekonomi, dan demografis yang memengaruhi status

					kelayakan penerima BPNT.
8.	Perbandingan Model Decision Tree, Naive Bayes dan Random Forest untuk Prediksi Klasifikasi Penyakit Jantung	Decision Tree, Naïve Bayes dan Random Forest. / 401.958 Dataset	Tujuan dari penelitian ini adalah untuk bagaimana mengolah dan melakukan analisa data, penerapan metode Decision Tree, Naive Bayes dan Random Forest pada klasifikasi penyakit jantung.	Deo Haganta Depari, Yuni Widiasti wi, Mayanda Mega Santoni / 2022	Hasil dari penelitian ini adalah evaluasi performa metode klasifikasi Decision Tree, Naive Bayes dan Random Forest. Dimana nilai akurasi metode Decision Tree sebesar 0.71%, Naive Bayes sebesar 0.72% dan Random Forest sebesar 0.75%.

A. Analisis Keterkaitan dan Perbandingan dengan Penelitian Terdahulu

Beberapa penelitian terdahulu telah membahas mengenai penerapan algoritma klasifikasi *machine learning*, khususnya *Random Forest* dan *Decision Tree*, dalam bidang kesehatan untuk mendukung proses prediksi penyakit. Kajian terhadap penelitian-penelitian terdahulu dilakukan untuk memahami sejauh mana algoritma tersebut telah digunakan, bagaimana hasil yang diperoleh, serta posisi penelitian ini di antara studi-studi yang sudah ada. Secara umum, hasil-hasil penelitian tersebut menunjukkan bahwa *Random Forest* memiliki tingkat akurasi dan kestabilan yang lebih tinggi dibandingkan *Decision Tree*, terutama dalam menangani data medis yang kompleks dan bersifat non-linear. Oleh karena itu, subbab ini menyajikan analisis keterkaitan dan perbandingan antara penelitian terdahulu dengan penelitian yang dilakukan penulis, dengan tujuan untuk menegaskan keselarasan hasil serta memperlihatkan kontribusi baru yang diberikan dalam konteks prediksi risiko penyakit jantung. Berikut adalah beberapa penelitian yang relevan dengan penelitian ini:

1. Penelitian yang dilakukan oleh Sari (2021) berjudul “Komparasi Algoritma Random Forest dan Decision Tree untuk Memprediksi Keberhasilan Immunotherapy” menunjukkan bahwa Random Forest memiliki tingkat akurasi yang lebih tinggi dibandingkan Decision Tree. Hal tersebut disebabkan oleh kemampuan Random Forest dalam menggabungkan beberapa pohon keputusan (ensemble learning) sehingga dapat mengurangi risiko overfitting dan meningkatkan ketepatan hasil prediksi.
2. Penelitian oleh Dana (2024) dengan judul “Perbandingan Algoritma Decision Tree dan Random Forest dengan Hyperparameter Tuning dalam Mendeteksi Penyakit Stroke” menjelaskan bahwa Random Forest memiliki performa yang lebih akurat dan stabil dibandingkan Decision Tree, terutama setelah dilakukan hyperparameter tuning terhadap parameter model yang digunakan.

3. Penelitian yang dilakukan oleh Samosir (2024) berjudul “Komparasi Algoritma Random Forest, Naïve Bayes, dan K-Nearest Neighbor dalam Klasifikasi Data Penyakit Jantung” menunjukkan bahwa Random Forest menghasilkan akurasi tertinggi dibandingkan algoritma lainnya, termasuk Decision Tree dan Naïve Bayes, pada pengujian terhadap dataset penyakit jantung.
4. Penelitian oleh S.A. Putri (2024) berjudul “Penerapan Machine Learning Algoritma Random Forest untuk Prediksi Penyakit Jantung” dan penelitian oleh A. Aldiyansyah (2024) berjudul “Perbandingan Tingkat Akurasi Algoritma Decision Tree dan Random Forest dalam Mengklasifikasikan Penerima Bantuan Sosial BPNT di Desa Slangit” menunjukkan bahwa Random Forest memiliki performa prediksi yang lebih baik dibandingkan Decision Tree, dengan keunggulan pada kestabilan dan kemampuan generalisasi model dalam berbagai jenis data.

Berdasarkan analisis terhadap penelitian-penelitian terdahulu, dapat disimpulkan bahwa penelitian ini yang berjudul “Perbandingan Algoritma Random Forest dan Decision Tree dalam Memprediksi Resiko Terkena Penyakit Jantung” selaras dengan hasil penelitian sebelumnya, karena sama-sama menunjukkan bahwa algoritma Random Forest memiliki performa yang lebih unggul dibandingkan Decision Tree. Penelitian ini memperkuat temuan-temuan terdahulu dengan memberikan bukti empiris tambahan melalui pengujian yang lebih komprehensif menggunakan empat metrik utama, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*, serta penerapan pada dataset terbaru dari repositori *Kaggle* dalam kasus prediksi risiko penyakit jantung. Selain menunjukkan keselarasan, penelitian ini juga memberikan kontribusi ilmiah baru dalam hal pendekatan evaluasi yang lebih mendalam dan komparatif, sehingga dapat menjadi penguatan terhadap teori *Ensemble Learning*, yang menjelaskan bahwa penggabungan beberapa model prediktif mampu meningkatkan performa klasifikasi dibandingkan model tunggal.

Dengan demikian, penelitian ini dapat dipandang sebagai kelanjutan sekaligus penguatan dari studi-studi terdahulu, dengan hasil yang konsisten bahwa Random Forest lebih unggul, stabil, dan efektif untuk diterapkan dalam sistem pendukung keputusan medis berbasis data guna deteksi dini risiko penyakit jantung.