BAB II
LANDASAN TEORI

2.1 Stroke

Stroke adalah serangan mendadak pada otak yang terjadi karena gangguan aliran darah, yang dapat disebabkan oleh penyumbatan atau pecahnya satu atau lebih pembuluh darah otak. Gangguan ini dapat mempengaruhi fungsi otak baik secara parsial maupun total.[12] Stroke terdiri dari dua jenis. Yang pertama adalah stroke iskemik (bagian dari otak kehilangan aliran darah) dan yang kedua adalah stroke hemoragik (terjadi pendarahan di dalam otak). Secara global, setiap orang dewasa di atas usia 25 tahun berisiko mengalami stroke dalam hidup mereka. Stroke merupakan salah satu penyebab utama cacat jangka panjang yang serius. Stroke merupakan penyebab berkurangnya mobilitas pada lebih dari setengah dari penderita stroke yang berusia 65 tahun ke atas. Dampak dari stroke seringkali bersifat jangka pendek dan jangka panjang, tergantung pada bagian otak yang terpengaruh dan seberapa cepat pengobatannya.[13] Tanda dan gejala yang paling umum dari penyakit stroke meliputi kelemahan atau mati rasa pada bagian wajah, lengan, atau kaki. Gejala lainnya mencakup kesulitan berbicara atau memahami ucapan orang lain, sensasi pusing, kebingungan, gangguan penglihatan pada satu atau kedua mata, kesulitan dalam berjalan, kehilangan keseimbangan, pingsan atau kehilangan kesadaran, serta sakit kepala tanpa penyebab yang jelas. Stroke dapat dipicu oleh berbagai faktor termasuk tekanan darah tinggi, riwayat fibrilasi atrium, kolesterol tinggi, diabetes, dan lain-lain.[14]

2.2 Penelitian Terkait

Beberapa penelitian terkait yang telah dilakukan sebelumnya menggunakan teknik data mining metode klasifikasi menjadi acuan dalam penelitian ini. Berikut tabel 2.1 berisi beberapa penelitian terkait.

Tabel 2.1 Penelitian Terkait

No.	Judul, Author, Tahun	Metode	Dataset	Hasil
1	Komparasi Penerapan Metode Bagging dan Adaboost pada Algoritma C4.5 untuk Prediksi Penyakit Stroke. Nur Diana Saputri, Khalid Khalid, Dwi Rolliawati (2022)	Bagging, Adaboost, Algoritma c4.5	5110 data	Berdasarkan hasil evaluasi, algoritma C4.5 menghasilkan nilai akurasi sebesar 92,87%. Sedangkan akurasi dari algoritma C4.5 setelah menerapkan metode bagging adalah 95,02%, dan setelah menerapkan metode Adaboost adalah sebesar 94,63%. Komparasi dari metode Bagging dan Adaboost pada algoritma C4.5 terbukti dapat meningkatkan dan memperbaiki kinerja klasifikasi. Nilai akurasi algoritma C4.5 meningkat sebanyak 3% dan 2% setelah dikombinasikan dengan metode bagging dan Adaboost.
2	Comparative Analysis Of Accuracy Of Random Forest And Gradient Boosting Classifier Algorithm For Diabetes Classification Sahat Pandapotan Nainggolan, Ardiles Sinaga (2023)	Random Forest, Gradient Boosting Classifier	768 data	Berdasarkan hasil penelitian dengan menggunakan rasio 80:20, Algoritma Random Forest memiliki hasil akurasi sebesar 79%. Sementara dengan menggunakan Algoritma Gradient Boosting Classifier, hasil akurasi yang didapat sebesar 81%. Dari hasil tersebut menunjukkan bahwa Algoritma Gradient Boosting Classifier memiliki hasil evaluasi akurasi yang lebih besar dibandingkan dengan Algoritma Random Forest.
3	Penerapan Data Mining untuk Klasifikasi Penyakit Stroke Menggunakan Algoritma Naïve Bayes Agus Fajar Riany, Gusmelia Testiana (2023)	Naïve Bayes	4981 data	Dari pengujian ini menghasilkan tingkat akurasi sebesar 92,48% yang berada dalam kategori Good Classification.
4	Komparasi Metode Klasifikasi Data Mining Decision Tree dan	Decision Tree dan Naïve Bayes	520 data	Algoritma klasifikasi decision tree lebih baik dalam prediksi penyakit diabetes dengan nilai akurasi 95,58% dab nilai AUC 0,981 lebih tinggi

	Naïve Bayes Untuk Prediksi Penyakit Diabetes Baiq Andriska Candra Permana, dkk (2021)			dibandingkan naïve bayes dengan akurasi 87,69% dan nilai AUC 0,947.
5	Komparasi Algoritma C4.5 Berbasis PSO Dan GA Untuk Diagnosa Penyakit Stroke Ramdhan Saepul Rohman, Rizal Amegia Saputra, dkk (2020)	Algoritma C4.5, PSO, GA	43401 record	Berdasarkan hasil pengujian didapatkan akurasi algoritma C4.5 sebesar 99.07%. Selanjutnya Algoritma C4.5 dioptimasi dengan menggunakan Particle Swarm Optimization sehingga memperoleh akurasi sebesar 99.28% dan Algoritma C4.5 juga dioptimasi dengan menggunakan Genetic Algorithm sehingga memperoleh akurasi sebesar 99.38%.
6	Optimasi Akurasi Algoritma C4.5 Berbasis Particle Swarm Optimization dengan Teknik Bagging pada Prediksi Penyakit Ginjal Kronis Ita Yulianti, dkk (2020)	Algoritma C4.5, PSO, teknik bagging	400 data	Dari hasil penelitian yang diperoleh, akurasi yang dihasilkan algoritma C4.5 dalam prediksi penyakit ginjal kronis sebesar 91,72%, dengan AUC bernilai sebesar 0,931 sedangkan untuk akurasi optimasi algoritma C4.5 berbasis PSO dan bagging sebesar 99,70% dengan AUC bernilai 1,000, sehingga didapat selisih peningkatan akurasi sebesar 7,98% dan nilai AUC sebesar 0,069. Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa penerapan teknik optimasi PSO dan Bagging mampu menyeleksi atribut dan dapat mengatasi kelas yang tidak seimbang pada algoritma C4.5, sehingga menghasilkan tingkat akurasi diagnosis yang lebih baik dibandingkan dengan metode individual algoritma C4.5 saja.
7	Stroke Prediction using Machine Learning Method with Extreme Gradient Boosting Algorithm Abd Mizwar A. Rahim, dkk (2022)	Algoritma Xtreme Gradient Boosting	5110 data	Hasil penelitian menunjukkan bahwa klasifikasi menggunakan XGBoost (Xtreme Gradient Boosting) dengan perbandingan data training 70% dan data testing 30% mencapai akurasi terbaik sebesar 96%, yang merupakan hasil yang lebih baik dari penelitian sebelumnya.

8	Analisis Perbandingan Algoritma C4.5 Dan Naïve Bayes Dalam Memprediksi Penyakit Cerebrovascular Kelvin Leonardi Kohsasih, Zakarias Situmorang (2022)	Algoritma Decision Tree C4.5 dan Algoritma Naive Bayes	5110 data	Berdasarkan hasil penelitian yang telah dilakukan yaitu dengan membagi dataset menjadi 60% data training dan 40% data testing maka dapat disimpulkan bahwa algoritma C4.5 memiliki performa yang lebih baik yaitu dengan tingkat akurasi sebesar 95% serta nilai presisi, serta nilai presisi, recall dan f1-score masing masing yaitu 90%, 95% dan 93%. sedangkan algoritma naïvebayes mendapatkan tingkat akurasi sebesar 91%, presisi 92%, recall 91% dan f1-score sebesar 92%.
9	Penerapan Teknik Bagging Pada Algoritma Naive Bayes Dan Algoritma C4.5 Untuk Mengatasi Ketidakseimbangan Kelas Achmad Ridwan (2020)	Teknik Bagging, Algoritma Naive Bayes, Algoritma C4.5	100 data	Dari hasil penelitian, dengan menerapkan teknik bagging untuk klasifikasi berbasis ensemble pada algoritma C4.5 dapat meningkatkan akurasi sebesar 9 %. Dengan akurasi awal 68 %, setelah diterapkan teknik bagging menjadi 77 %. pada algoritma naïve bayes dapat meningkatkan akurasi sebesar 3,00 %. Dengan akurasi awal 77,00%, setelah diterapkan teknik bagging menjadi 80,00%.
10	Analisis Perbandingan Algoritma Decision Tree, kNN, dan Naïve Bayes untuk Prediksi Kesuksesan Start-up Adhitya Prayoga Permana, Kurniyatul Ainiyah, dkk (2021)	Algoritma Decision Tree, K-NN, dan Naïve Bayes	923 data	Hasil pengujian menggunakan cross validation dan T-test menunjukkan algoritma Decision Tree merupakan algoritma paling tepat untuk melakukan klasifikasi dalam studi kasus ini. Hal ini dibuktikan dengan nilai akurasi yang diperoleh oleh algoritma Decision Tree lebih besar diantara algoritma lainnya, yaitu sebesar 79,29%, sedangkan algoritma kNN memiliki nilai akurasi 66,69%, dan Naive Bayes sebesar 64,21%.
11	Analyzing the Performance of Stroke Prediction using ML Classification Algorithms Gangavarapu Sailasya, Gorli L Aruna Kumari (2021)	Logistic Regression, Decision Tree Classification, Random Forest Classification, K-Nearest Neighbors	5110 data	Berdasarkan hasil uji coba 6 algoritma klasifikasi yang dipilih, algoritma Naïve Bayes Classification memiliki kinerja terbaik dengan akurasi 82%. Sedangkan algoritma Logistic Regression memiliki akurasi 78%, Decision Tree Classification 66 %, Random Forest Classification 73 %, K-Nearest Neighbors Classification 80 %, dan Support Vector Machine sebesar 80 %.

		Classification, Support Vector Machine and Naïve Bayes Classification.		
12	Stroke Prediction Using Machine Learning based on Artificial Intelligence Youngkeun Choi, Jae Won Choi (2020)	Decision tree	43.400 data	Akurasi menggunakan algoritma decision tree adalah 0,981 dan tingkat kesalahan adalah 0,019. Tingkat akurasi non-stroke pada pasien yang diharapkan tidak mengalami stroke adalah 98,17% dan akurasi stroke yang diharapkan adalah 16,67%.
13	Classification of stroke patients using data mining with AdaBoost, Decision Tree and Random Forest models Bahtiar Imran, Erfan Wahyudi, Ahmad Subki, Salman, Ahmad Yani (2022)	AdaBoost, Decision Tree, Random Forest	5110 data	Berdasarkan hasil uji yang telah dilakukan dengan 10 fold cross validation, algoritma decision tree memperoleh hasil akurasi tertinggi, yaitu 0.953. sedangkan Adaboost 0,915 % dan Random Forest sebesar 0,950 %. Dari penelitian ini, dapat disimpulkan bahwa pohon keputusan dapat memberikan klasifikasi yang baik untuk klasifikasi stroke, hasil penelitian ini bergantung pada dataset yang digunakan.
14	Brain Stroke Prediction Using Random Forest And Adaboost Algorithm Mylapalli Kanthi Rekha, I. Phani Kumar (2023)	Random Forest, Adaboost	4981 data	Hasil penelitian ini menunjukkan bahwa baik algoritma Random Forest maupun AdaBoost berhasil mencapai hasil terbaik dalam memprediksi risiko stroke otak. Algoritma Random Forest mencapai akurasi di atas 90%, sementara algoritma AdaBoost mencapai akurasi di atas 90%.

Berdasarkan tabel penelitian diatas, maka dapat diasumsikan bahwa penggunaan algoritma Decision Tree dan Naïve Bayes sudah memiliki nilai akurasi yang tinggi diatas 75%. Namun, jika algoritma Decision Tree dan Naïve Bayes dikombinasikan dengan teknik ensemble akan menghasilkan nilai akurasi yang jauh lebih tinggi. Hal ini terlihat dari salah satu penelitian pada tabel diatas dengan judul "Penerapan Teknik Bagging Pada Algoritma Naive Bayes Dan Algoritma C4.5 Untuk Mengatasi Ketidakseimbangan Kelas" oleh Achmad Ridwan pada tahun 2020 bahwa penelitian menggunakan algoritma Decision Tree dan Naïve Bayes yang dikombinasikan dengan teknik Bagging pada algoritma C4.5 dapat meningkatkan akurasi sebesar 9 %. Dengan akurasi awal 68 %, setelah diterapkan teknik bagging menjadi 77 %. Pada algoritma naïve bayes dapat meningkatkan akurasi sebesar 3,00 %. Dengan akurasi awal 77,00%, setelah diterapkan teknik bagging menjadi 80,00%. Maka dari itu, dalam penelitian ini peneliti menggunakan Algoritma Decision Tree dan Naïve Bayes yang dikombinasikan dengan teknik ensemble yaitu Bagging dan Adaboost untuk meningkatkan nilai akurasi pada prediksi penyakit stroke.

2.3 Data Mining

Data Mining merupakan kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar. Keluaran dari *data mining* ini bisa dipakai untuk memperbaiki pengambilan keputusan di masa depan.[15]

Inti dari proses Knowledge Discovery in Database (KDD) adalah Data Mining (DM), yang melibatkan algoritma yang menyelidiki data, membangun model, dan mengungkap pola yang sebelumnya tidak diketahui. Model ini digunakan untuk mengetahui kekhasan dari informasi, pemeriksaan dan harapan. Keterbukaan dan kekayaan informasi menjadikan *Knowledge Discovery* dan *Data Mining* menjadi isu yang sangat signifikan dan penting. [16] Tujuannya adalah untuk mengubah data yang tidak terstruktur menjadi informasi berarti yang dapat digunakan untuk pengambilan keputusan yang lebih baik dan pemahaman yang lebih dalam. Ada beberapa teknik data mining yang umum digunakan untuk menggali informasi berharga dari data. Berikut adalah beberapa teknik data mining yang populer:

- Regresi: Teknik ini digunakan untuk memahami hubungan antara variabel dependen dan variabel independen. Regresi linier dan regresi logistik adalah contoh umum dari teknik regresi yang digunakan dalam data mining.
- Klasifikasi: Teknik klasifikasi digunakan untuk mengklasifikasikan entitas ke dalam kelompok atau kategori yang telah ditentukan berdasarkan atribut dan karakteristik tertentu. Contoh umum dari teknik klasifikasi adalah pohon keputusan, naive bayes, dan algoritma k-nearest neighbors (k-nn).
- Klasterisasi: Klasterisasi adalah teknik yang digunakan untuk mengelompokkan entitas yang serupa ke dalam kelompok atau klaster berdasarkan kesamaan atribut atau pola. Contoh teknik klasterisasi meliputi algoritma k-means, DBSCAN, dan algoritma hierarkis.
- Asosiasi: Teknik asosiasi digunakan untuk menemukan hubungan dan pola asosiasi antara item atau variabel dalam kumpulan data. Algoritma yang populer untuk asosiasi adalah algoritma Apriori dan FP-Growth.

- Analisis Anomali: Teknik ini fokus pada pendeteksian dan identifikasi anomali atau pencilan dalam data. Anomali adalah nilai yang berbeda secara signifikan dari pola umum atau normal dalam kumpulan data. Metode yang umum digunakan untuk analisis anomali termasuk deteksi pencilan statistik, metode berbasis jarak, dan metode berbasis model.
- Analisis Deret Waktu: Teknik ini digunakan untuk menganalisis data yang dikumpulkan secara berurutan dalam interval waktu tertentu. Analisis deret waktu melibatkan pemodelan, prediksi, dan identifikasi pola dalam deret waktu. Contoh teknik dalam analisis deret waktu termasuk moving average, ARIMA (Autoregressive Integrated Moving Average), dan analisis spektral.
- Penambangan teks: Teknik ini digunakan untuk mengekstraksi informasi yang bermanfaat dari dokumen teks atau sumber daya teks lainnya. Ini melibatkan pemrosesan teks, pengindeksan, klasifikasi, dan ekstraksi informasi terkait. Contoh teknik dalam penambangan teks termasuk pemrosesan bahasa alami (natural language processing), analisis sentimen, dan ekstraksi entitas.

2.4 Decision Tree

Decision Tree merupakan algoritma klasifikasi yang dinyatakan sebagai partisi rekursif dari ruang contoh. Algoritma ini terbentuk oleh serangkaian simpul yang membentuk struktur pohon, yang artinya pohon tersebut memiliki simpul akar sebagai titik awal. Setiap simpul dalam pohon memiliki tepi keluar dan disebut sebagai simpul internal atau simpul uji. Sementara simpul lainnya dikenal sebagai daun. Dalam konteks pohon keputusan, setiap simpul internal melakukan pemisahan ruang contoh menjadi dua atau lebih sub-ruang berdasarkan nilai atribut tertentu yang bersifat diskrit. Sebuah pohon keputusan terdiri dari internal node yang menentukan tes pada variabel masukan individu atau atribut yang membagi data menjadi himpunan bagian yang lebih kecil, dan serangkaian node daun menetapkan kelas untuk masing-masing pengamatan di segmen yang dihasilkan.[17]

2.5 Naïve Bayes

Naïve Bayes adalah metode klasifikasi probabilistik yang menggunakan Teorema Bayes, di mana probabilitas kelas untuk setiap sampel dihitung dan diasumsikan bahwa semua atribut independen satu sama lain. Dalam klasifikasi menggunakan naïve Bayes, kelas yang paling mirip dengan hasil perhitungan dipilih sebagai klasifikasi akhir. Metode ini menggunakan prinsip dasar teori probabilitas untuk menemukan probabilitas terbesar dalam klasifikasi, dengan mempertimbangkan frekuensi masing-masing klasifikasi dalam data latih. Salah satu keunggulan metode ini adalah bahwa ia membutuhkan sedikit data latih untuk menentukan parameter yang diperlukan dalam proses klasifikasi.[18]

Algoritma Naive Bayes merupakan sebuah teknik populer dalam bidang data mining dan pembelajaran mesin yang digunakan untuk melakukan klasifikasi. Algoritma ini didasarkan pada teorema Bayes dengan asumsi "naive" bahwa semua fitur dalam data adalah independen satu sama lain. Meskipun asumsi ini jarang terpenuhi dalam situasi dunia nyata, algoritma Naive Bayes tetap menjadi pilihan yang baik karena memberikan hasil yang baik dan memiliki kecepatan komputasi yang tinggi. Algoritma Naive Bayes umumnya digunakan dalam berbagai aplikasi seperti klasifikasi teks, analisis sentimen, penyaringan spam, deteksi penipuan, dan lainnya. Meskipun memiliki asumsi sederhana, algoritma ini sering menghasilkan hasil yang memuaskan dengan biaya komputasi yang relatif rendah. Namun, perlu diingat bahwa performa algoritma Naive Bayes dapat dipengaruhi oleh asumsi independensi fitur yang kuat, sehingga ada situasi di mana algoritma ini mungkin tidak memberikan hasil yang akurat. Rumus teorema bayes dapat dilihat pada persamaan sebagai berikut:

$$P(H|X) = \frac{P(X|H)\ P(H)}{P(X)}$$

Dimana:

X = Data dengan kelas yang tidak diketahui

H = Hipotesis data X yang merupakan kelas tertentu

$P(H|X)$ = Peluang Hipotesa H berdasarkan kondisi X

$P\ (H)$ = Peluang dari Hipotesa H

$P(X|H)$ = Peluang X berdasarkan kondisi H

$P\ (X)$ = Peluang dari X

2.6 Bagging

Teknik Bagging merupakan teknik yang sukses untuk menangani dataset yang tidak seimbang.[19] Teknik bagging adalah salah satu teknik ensemble yang diterapkan dalam klasifikasi untuk membagi data pelatihan menjadi beberapa set pelatihan baru melalui pengambilan sampel acak, dan selanjutnya membangun model berdasarkan set pelatihan yang baru dibentuk. Teknik bagging dapat diterapkan pada berbagai jenis model pembelajaran mesin, seperti pohon keputusan, regresi logistik, dan lainnya. Kelebihan utama teknik ini terletak pada kemampuannya untuk meningkatkan kestabilan dan kinerja model secara umum. Berikut rumus perhitungan teknik bagging:

Input: Data set D = {(Xi,Yi)} =1in (6)

Algoritma Pembelajaran $\mathfrak{L}$

Jumlah Iterasi T

1) for t=1 to T do
2) ht = $\mathfrak{L}(D,\mathcal{D}bs)$
3) end for
4) Output: H(x)=max$y$ Σt=1 n $I(ht(x) = y)$

2.7 Adaboost

Adaptive Boosting merupakan algoritma machine learning yang digunakan untuk meningkatkan performa model yang lemah atau kurang akurat dengan menggabungkan beberapa model yang sederhana. Adaboost adalah teknik ensemble learning yang terkenal dan efektif dalam mengatasi permasalahan klasifikasi. Adaboost dan variasinya telah berhasil diterapkan di berbagai bidang

karena dasar teorinya yang kuat, kemampuan prediksi yang akurat, dan kesederhanaannya.[20] AdaBoost menunjukkan sifat adaptif dengan membangun classifiers berikutnya untuk memberikan dukungan pada data yang telah keliru diklasifikasikan oleh classifier sebelumnya. Algoritma ini rentan terhadap data yang noisy dan outliers. Namun, dalam beberapa situasi, AdaBoost cenderung lebih tahan terhadap masalah overfitting jika dibandingkan dengan algoritma pembelajaran lainnya.[21]

Berikut langkah-langkah teknik pembobotan pada algoritma Adaboost:

1. Inisialisasi bobot data {Wn} dengan $W^{n(m)}$ untuk n=1,2,...,N
2. For m=1...,M

a. Training $Y^{m(x)}$ dengan meminimalkan fungsi kesalahan (*error function*) sebagai berikut:

$Jm = \sum_n^N = 1\ W_n^{(m)} \mathrm{I}(y_m(X_n) \neq t_n$

b. Evaluasi Kesalahan

$$\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n}{\sum_{n=1}^{N} w_n^{(m)}}$$

c. Kemudian digunakan evaluasi

$$\alpha^{m} = \ln\left\{\frac{\varepsilon^{m}}{1-\varepsilon^{m}}\right\}$$

d. Memperbaiki (Update) bobot data

$$w_n^{(m+1)} = w_n^{(m)} exp\{\alpha_m I(y_m(x_n) \neq t_n)\}$$

e. Membuat prediksi menggunakan model terakhir sebagai berikut:

$$Y_M(X) = sign\left(\sum^{M}\right.$$

2.8 Confusion Matrix

Confusion matrix adalah teknik yang digunakan untuk mengukur akurasi melalui perhitungan yang diterapkan dalam teknik data mining.[18] Confusion matrix adalah tabel yang digunakan untuk mengevaluasi performa model klasifikasi dengan membandingkan prediksi model dengan nilai aktual dari data. Evaluasi kinerja menggunakan *confusion matrix* melibatkan empat nilai utama. Keempat komponen dalam *confusion matrix* tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Perhitungan *Confusion matrix* ditunjukkan pada Tabel 2.2 berikut.

Tabel 2.2 Confusion Matrix

		True Value	
		True	False
Forecast Value	True	TP (*True Positive*)	FP (*False Positive*)
	False	FN (*False Negative*)	TN (*True Negative*)

Penjelasan tabel diatas adalah sebagai berikut:

- TP (*True Positive*): Jumlah data dengan nilai aktual positif dan nilai prediksi positif.
- FP (*False Positive*): Jumlah data dengan nilai aktual negative dan nilai prediksi positif.
- FN (*False Negative*): Jumlah data dengan nilai aktual positif dan nilai prediksi negative.
- TN (*True Negative*): Jumlah data dengan nilai aktual negative dan nilai prediksi negative.

2.8.1 Akurasi/*Accuracy*

Akurasi adalah metrik kinerja yang paling intuitif untuk mengukur proporsi prediksi yang tepat pada keseluruhan dataset.[18] Akurasi memberikan gambaran umum tentang seberapa baik suatu model atau tes bekerja dalam mengklasifikasikan data. Namun, akurasi mungkin tidak cukup untuk memberikan pemahaman yang lengkap tentang kinerja model atau tes dalam beberapa kasus, terutama jika kelas-kelas yang diuji tidak seimbang dalam distribusi atau jika kesalahan jenis tertentu lebih penting daripada yang lain. Dalam beberapa situasi, penting untuk mempertimbangkan metrik lain seperti sensitivitas, spesifisitas, atau skor F1 untuk mendapatkan pemahaman yang lebih holistik tentang kinerja suatu model atau tes. Berikut rumus untuk menghitung nilai akurasi [22]:

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

Dimana:

TP = Persentase kasus positif yang berhasil diidentifikasi dengan benar.

TN = Persentase kasus negative yang berhasil diklasifikasikan dengan benar.

FP = Persentase kasus negative yang salah diklasifikasikan sebagai positif

FN = Persentase kasus positif yang salah diklasifikasikan sebagai negatif.

2.8.2 Presisi/*Precision*

Presisi adalah ukuran dalam analisis statistik yang mengukur proporsi dari prediksi positif yang tepat di antara semua kasus yang telah diprediksi sebagai positif. Dalam konteks klasifikasi, presisi mengacu pada jumlah positif yang diprediksi dengan benar dibagi oleh total jumlah prediksi positif, baik yang benar maupun yang salah. Presisi memberikan informasi tentang seberapa akurat model dalam mengidentifikasi kelas yang diminati.[18] Presisi merupakan rasio antara jumlah data yang terklasifikasi secara tepat sebagai true positive (termasuk data yang diprediksi sebagai

"Average", "Block-Buster", "Flop", dan "Success") dibagi dengan total jumlah data yang terklasifikasi sebagai tinggi.[23] Presisi memberikan informasi tentang seberapa andalnya suatu model dalam mengklasifikasikan data sebagai positif. Semakin tinggi nilai presisi, semakin sedikit kasus negatif yang salah diidentifikasi sebagai positif, yang menunjukkan bahwa model tersebut cenderung memberikan hasil positif yang benar. Rumus yang digunakan yaitu:

$$Precision = \frac{TP}{(TP + FP)}$$

2.8.3 Sensitivitas/*Recall*

Recall yang juga dikenal sebagai sensitivitas, adalah ukuran yang mengindikasikan seberapa baik model klasifikasi mampu mengidentifikasi semua kasus positif yang sebenarnya dalam dataset. Dalam konteks klasifikasi, recall dihitung sebagai rasio antara jumlah positif yang berhasil diprediksi dengan benar (*true positive*) dibagi dengan jumlah seluruh kasus positif yang sebenarnya (*true positive* + *false negative*). Dengan kata lain, *recall* mengukur kemampuan model untuk "mengingat" atau "mendeteksi" semua kasus positif yang ada, tanpa melewatkan satu pun.[23] Sebagai contoh, dalam kasus deteksi kanker, *recall* akan mengukur seberapa baik model dapat mengidentifikasi semua kasus kanker yang sebenarnya dalam populasi, sehingga dapat mengurangi kemungkinan kasus kanker yang terlewatkan. Dalam konteks medis dan keamanan, *recall* sering diutamakan untuk memastikan bahwa tidak ada kasus positif yang terlewatkan, meskipun bisa saja ada peningkatan dalam jumlah kasus negatif yang salah terklasifikasi sebagai positif (*false positive*). Semakin tinggi nilai sensitivitas, semakin baik tes atau model tersebut dalam mengidentifikasi kasus positif yang sebenarnya ada. Dengan kata lain, sensitivitas mengukur kemampuan suatu tes atau model untuk mengurangi jumlah kasus positif palsu negatif, yaitu kasus positif yang tidak terdeteksi. Berikut rumus recall:

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

2.8.4 *Specificity*/Spesifisitas

Spesifisitas adalah ukuran dari seberapa baik suatu tes atau model mampu mengidentifikasi kasus negatif yang sebenarnya ada. Dalam konteks diagnostik medis atau evaluasi model klasifikasi, spesifisitas mengukur persentase kasus negatif yang benar-benar diidentifikasi sebagai negatif dari keseluruhan kasus negatif yang ada. Spesifisitas mengindikasikan seberapa baik suatu tes dapat mengidentifikasi individu yang sebenarnya tidak mengalami penyakit. Semakin tinggi spesifisitas tes, semakin sedikit hasil tes yang salah mengidentifikasi orang sehat sebagai positif atau semakin sedikit jumlah kesalahan positif palsu yang terjadi.[24] berikut rumus spesifisitas:

$$\text{Spesifisitas} = \frac{TN}{TN+TP}$$

Dimana:

- *True Negative* adalah jumlah kasus negatif yang teridentifikasi sebagai negatif
- *False Positive* adalah jumlah kasus positif yang salah diidentifikasi sebagai negatif.

Spesifisitas adalah parameter yang penting dalam evaluasi keandalan suatu tes diagnostik atau model klasifikasi. Semakin tinggi nilai spesifisitas, semakin baik tes atau model tersebut dalam mengidentifikasi kasus negatif yang sebenarnya ada. Dalam konteks klasifikasi, spesifisitas mengukur kemampuan suatu model untuk mengurangi jumlah kasus negatif palsu positif, yaitu kasus negatif yang salah diidentifikasi sebagai positif.

2.9 Kurva ROC

Kurva ROC adalah metode evaluasi yang sering dipakai untuk mengukur kehandalan sistem klasifikasi. Kurva ini sering digunakan karena mampu mengevaluasi algoritma dengan akurat. Kurva ROC menampilkan perbandingan antara *sensitivity* (tingkat positif benar (TPR)) dan *specificity* (tingkat positif palsu (FPR)), yang direpresentasikan sebagai kurva pada grafik.[20] Kurva ROC menggambarkan akurasi dan membandingkan klasifikasi secara visual. ROC mengekspresikan *confusion matrix*.[25]

2.10 Preprocessing Data

Preprocessing data adalah rangkaian langkah yang dilakukan untuk membersihkan, mengubah, dan menyiapkan data mentah sebelum digunakan untuk analisis lanjutan atau pemodelan. Tahapan *preprocessing data* merupakan aspek krusial dalam menghasilkan data berkualitas tinggi. Proses ini melibatkan beberapa langkah seperti validasi, integrasi, dan transformasi data.[26]

Tujuan utama dari *preprocessing data* adalah untuk meningkatkan kualitas data, mengurangi gangguan atau noise, dan menyiapkan data agar sesuai dengan kebutuhan analisis atau pemodelan yang diinginkan. Tahapan *preprocessing data* adalah sebagai berikut:

1. Pengumpulan Data

   Tahap pertama dalam *preprocessing data* adalah mengumpulkan data yang relevan dan sesuai dengan tujuan analisis atau pemodelan data mining yang akan dilakukan. Data dapat diperoleh dari berbagai sumber, termasuk basis data, file teks, data streaming, atau sumber data lainnya.

2. Integrasi Data

   Proses penggabungan berbagai sumber informasi disebut proses integrasi data.[27] Setelah mendapatkan informasi dari berbagai sumber, langkah berikutnya adalah menggabungkan semua data tersebut ke dalam satu dataset yang seragam. Proses ini melibatkan penyelarasan struktur data, penyesuaian format yang beragam, serta penanganan nilai yang hilang atau duplikat.

3. *Data Cleaning*

   Proses persiapan data untuk analisis mencakup penghapusan atau modifikasi data yang salah, tidak relevan, duplikat, dan tidak terstruktur.[28] Tahap ini melibatkan identifikasi dan penanganan nilai yang hilang, noise, atau outlier dalam data. Data yang tidak valid atau tidak lengkap dapat dihapus atau diisi dengan nilai yang sesuai. Noise atau outlier dapat diidentifikasi dan ditangani dengan teknik seperti deteksi outlier atau smoothing data.

4. Transformasi Data:

   Transformasi data dilakukan untuk mengubah format atau skala data agar sesuai dengan kebutuhan analisis atau pemodelan. Ini bisa termasuk transformasi logaritmik, normalisasi data, pengubahan skala, atau penggabungan atribut.

2.11 RapidMiner

RapidMiner adalah sebuah platform perangkat lunak analitik yang kuat dan user-friendly yang digunakan untuk melakukan pemrosesan data, analisis prediktif, dan pemodelan data. Rapidminer digunakan untuk mengekstraksi informasi dari dataset dengan mengintegrasikannya ke dalam basis data. Perangkat lunak ini digunakan dalam pengelolaan data mining untuk berbagai keperluan penelitian, termasuk ekstraksi, transformasi, dan penyimpanan data.[29]

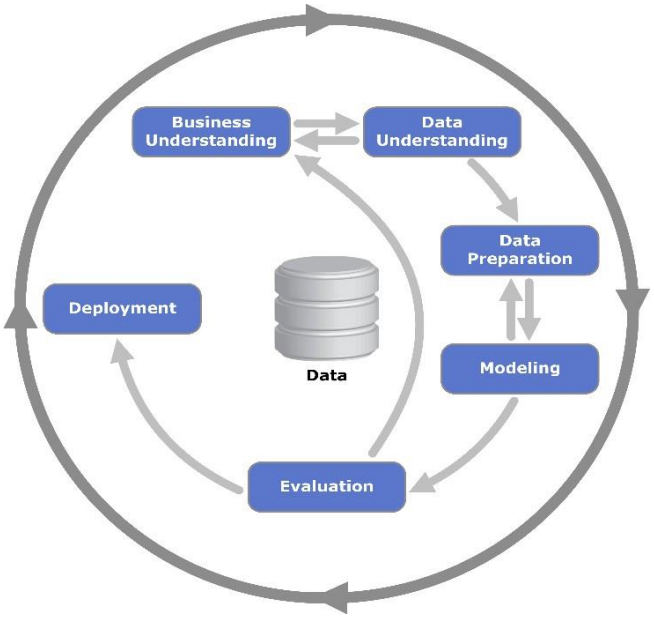
Berikut adalah beberapa fitur utama dan fungsi yang disediakan oleh RapidMiner:

1. *Preprocessing Data*: RapidMiner memungkinkan pengguna untuk melakukan berbagai operasi pra-pemrosesan data, seperti pembersihan data, pengisian nilai yang hilang, transformasi variabel, pemilihan fitur, dan normalisasi data.
2. Eksplorasi Data dan Visualisasi: Platform ini menyediakan berbagai alat untuk menganalisis dan memvisualisasikan data. Pengguna dapat mengeksplorasi distribusi data, membuat grafik dan visualisasi, serta mendapatkan wawasan yang berguna tentang pola dan hubungan dalam dataset.

3. Model Prediktif: RapidMiner memiliki kemampuan untuk membangun dan mengevaluasi model prediktif. Pengguna dapat menggunakan berbagai algoritma pembelajaran mesin seperti regresi linier, pengklasifikasi, pengelompokan, dan jaringan saraf tiruan. Selain itu, platform ini menyediakan alat validasi silang, penyetelan parameter, dan evaluasi kinerja model.
4. Integrasi dan Ekstensibilitas: RapidMiner dapat diintegrasikan dengan berbagai sumber data, termasuk database, file Excel, file CSV, dan sumber data online. Selain itu, platform ini dapat diperluas melalui pengembangan tambahan dengan menggunakan bahasa pemrograman seperti R dan Python.

2.12 *Cross Industry Standard Process for Data Mining* (CRISP-DM)

CRISP-DM, yang merupakan singkatan dari *Cross Industry Standard Process for Data Mining*, adalah metodologi yang banyak digunakan untuk memandu proyek data mining dan analitika. Metodologi ini memberikan pendekatan terstruktur yang membantu organisasi dalam efisiensi dan efektivitas dalam memecahkan masalah bisnis menggunakan wawasan berbasis data. CRISP-DM (*Cross-Industry Standard Process for Data Mining*) melibatkan serangkaian langkah yang mencakup proses keseluruhan, preprocessing data, pembentukan model, evaluasi model, dan tahap penyebaran model.[25] Gambar 2.1 merupakan proses *Data Mining* CRISP-DM sebagai berikut :



Gambar 2.1 Proses CRISP-DM

Berikut penjelasan dari proses CRISP-DM:

1. *Business Understanding* (Pemahaman Bisnis)

   Tahap ini dimulai dengan memahami tujuan bisnis proyek. Hal ini melibatkan identifikasi tujuan, kebutuhan, dan kepentingan pemangku kepentingan proyek.

2. *Data Understanding* (Pemahaman Data)

   Pada tahap ini, data yang relevan untuk proyek dikumpulkan, dieksplorasi, dan dipahami lebih dalam. Ini juga melibatkan identifikasi masalah data dan memahami karakteristik data.

3. *Data Preparation* (Pengolahan Data)

   Dalam *data preparation*, data dipersiapkan untuk analisis dengan membersihkan, mengintegrasikan, dan memformat data. Langkah ini termasuk pemilihan atribut, transformasi data, dan pengisian data yang hilang.[30]

4. *Modeling* (Pemodelan)

Tahap ini melibatkan pemilihan model yang sesuai, pembangunan model, dan validasi model. Beberapa teknik pemodelan yang umum digunakan termasuk regresi, klasifikasi, dan klastering.

5. *Evaluation* (Evaluasi)

   Model yang dibangun dievaluasi untuk memastikan bahwa mereka memenuhi tujuan bisnis yang telah ditetapkan. Evaluasi ini dapat dilakukan dengan menggunakan metrik evaluasi yang sesuai dengan jenis masalah yang dihadapi.

6. *Deployment* (Penyebaran)

   Model yang dievaluasi dan disetujui siap untuk diimplementasikan dalam lingkungan produksi. Tahap ini melibatkan penyajian model kepada pemangku kepentingan dan integritasnya dengan sistem yang ada.