BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

2.1.1. Prediksi Kemampuan Mahasiswa Dengan Educational Data Mining (EDM).

Setiap perguruan tinggi mempunyai caranya tersendiri untuk dapat bersaing dengan perguruan tinggi lainnya untuk menghasilkan lulusan terbaik. Kualitas mahasiswa yang dihasilkan oleh perguruan tinggi dipengaruhi oleh banyak faktor. Dalam memenuhi parameter kualitas luaran yang baik, diperlukan data historis terkait perkembangan kualitas pendidikan setiap mahasiswa yang ada (Yang & Li, 2018). Prediksi kemampuan mahasiswa dapat dilakukan dengan berbagai cara, salah satunya dengan pendekatan Educational Data Mining (EDM). EDM merupakan teknik yang sering digunakan untuk prediksi kemampuan mahasiswa (Aldowah et al., 2019). EDM juga mampu mengeksplorasi data-data spesifik yang dihasilkan dari proses yang terjadi di lingkungan pendidikan (Bakhshinategh et al., 2017). EDM dapat diartikan sebagai proses data mining yang mengolah sejumlah dataset yang dihasilkan dari proses di lingkungan pendidikan untuk menyelesaikan permasalahan utama pada bidang Pendidikan (Romero & Ventura, 2020).

Menurut (Shahiri et al., 2015) prediksi kemampuan mahasiswa memiliki 2 faktor utama yakni: atribut dan metode prediksi, dengan atribut yang sering digunakan adalah Cummulative Grade Point Average (CGPA). CGPA kembali digunakan sebagai atribut oleh (Natek & Zwilling, 2014), (Al Fanah & Ansari, 2019), dan (Miguéis et al., 2018). Atribut lainnya yang digunakan adalah atribut nilai ujian, nilai praktikum, dan nilai ujian akhir (Mayilvaganan & Kalpanadevi, 2015), (Christian & Ayub, 2014).

Dalam proses prediksi kemampuan mahasiswa beberapa teknik klasifikasi data mining telah digunakan, (Natek & Zwilling, 2014) menggunakan metode Decision Tree (DT) untuk memprediksi kemampuan mahasiswa pada beberapa

kelas kecil dengan jumlah 32 dan 42 mahasiswa. Metode Support Vector Machine (SVM) digunakan oleh (Gray et al., 2014) untuk memprediksi kemampuan 1.074 mahasiswa di tahun pertama perkuliahan. Algoritma Random Forest (RF) digunakan oleh (Nachouki & Naaj, 2022) untuk memprediksi kemampuan akademik dengan mengidentifikasi mahasiswa yang memiliki kemampuan rendah sehingga dapat terhindar dari drop-out. XGB juga muncul sebagai algoritma yang sangat efisien, dengan akurasi tertinggi mencapai 97,12% pada proses klasifikasi Prediksi hasil belajar sains siswa di sekolah (Harefa, 2024).

Prediksi terhadap kemampuan akademik mahasiswa berdasarkan data dari proses Penerimaan Mahasiswa Baru (PMB) telah dilakukan oleh beberapa peneliti. (Hassan & Al-Razgan, 2016) meneliti hubungan antara hasil ujian seleksi masuk dengan kemampuan akademik mahasiswa di perguruan tinggi menggunakan pendekatan regresi, dengan data sebanyak 657 mahasiswa dari masing-masing departemen di setiap fakultas. Sementara itu (Aluko et al., 2018) mengembangkan pendekatan prediksi performa akademik mahasiswa dengan memanfaatkan data capaian akademik sebelumnya. Indeks Prestasi Kumulatif (IPK) dijadikan indikator kinerja akademik, yang diklasifikasikan menjadi dua kategori: lulus (IPK $\geq$ 2,4) dan gagal (IPK < 2,4). Dalam studi tersebut, digunakan algoritma *Logistic Regression* (LR) dan *Support Vector Machine* (SVM), dengan hasil menunjukkan bahwa model SVM memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan model LR. Metode-metode yang digunakan dalam EDM untuk memprediksi prestasi akademik diantaranya adalah Decision Tree (DT), Random Forest (RF), dan Support Vector Machine (SVM) (Ul Hassan et al., 2018). Kemudian terdapat penelitian yang menunjukkan bahwa algoritma Extreme Gradient Boosting (XGB) dapat memberikan nilai evaluasi yang cukup baik pada kasus klasifikasi Prediksi hasil belajar sains siswa di sekolah (Harefa, 2024).

2.1.2. Keaslian dan Keterbaruan Penelitian

Berdasarkan hasil tinjauan pustaka yang telah disampaikan, dapat diidentifikasi sejumlah penelitian utama yang menjadi landasan dalam studi ini. Rangkuman penelitian-penelitian tersebut disajikan dalam Tabel 2.1 berikut.

Tabel 2.1. Penelitian Lain Yang Melandasi Keaslian Penelitian.

No.	Judul, Penulis, Tahun	Metodologi	Hasil	Keterangan	Perbandingan Penelitian
1.	Analyzing students' academic performance using educational data mining (S. Sarker et al., 2024)	Menganalisa kemampuan akademik Higher Secondary Certificate (HSC) di Bangladesh menggunakan nilai ujian internal, untuk digunakan dalam prediksi nilai IPK/ GPA akhir, Menggunakan DT, KNN, Naive Bayes, Neural Network, Random Forest	Menghasilkan model yang memprediksi GPA, dengan level akurasi, f1-score, dan kappa yang sangat baik. RF dan NN mendapatkan hasil evaluasi yang paling tinggi Dapat memprediksi booster/degrader subjects, early risk detection, dan intervention strategies.	Tools: Google Colab (Python), RapidMiner for modeling and evaluation dataset: 309 data mahasiswa dari universitas di Bangladesh (2018–2019); 4 internal exams (Half Yearly, Yearly, PreTest, Test) across 7 subjects; also used a synthetic dataset of 1000 students for model validation.	Terdahulu: melakukan prediksi menggunakan data akademik Algoritma: DT, KNN, Naive Bayes, Neural Network, Random Forest Baru: Prediksi menggunakan data pre-admisi SNBP, dan nilai akademik pada tahun ke-1 Algoritma: DT, RF, SVM, XGB
2	Predicting Science Learning Outcomes Of Elementary Students In Rural Area Using Machine Learning Algorithms (Harefa, 2024)	Prediksi hasil belajar sains siswa di sekolah dasar menggunakan berbagai algoritma Machine Learning; Random Forest, Support Vector Machine, Gradient Boosting, Decision Tree,	Akurasi Tertinggi: XGBoost (97,12%, cross-validation 35,67%) → mengungguli model lainnya.	XGBoost paling efektif untuk memprediksi hasil belajar sains siswa Faktor non-akademik (sosial & demografis) memainkan peran	Terdahulu: melakukan prediksi menggunakan data akademik Algoritma: Random Forest, Support Vector Machine, Gradient Boosting, Decision

No.	Judul, Penulis, Tahun	Metodologi	Hasil	Keterangan	Perbandingan Penelitian
		Extreme Gradient Boosting	Faktor paling berpengaruh terhadap kinerja akademik: jumlah absensi, status kesehatan, interaksi sosial, dan hubungan keluarga.	signifikan dalam prestasi akademik siswa.	Tree, Extreme Gradient Boosting Baru: Prediksi menggunakan data pre-admisi SNBP, dan nilai akademik pada tahun ke-1 Algoritma: DT, RF, SVM, XGB
3.	Case Study: Using Data Mining to Predict Student Performance Based on Demographic Attributes (Alghazali et al., 2023)	Dengan pendekatan Knowledge Discovery in Databases (KDD) dengan tahapan , yaitu: Pemilihan dan persiapan data, Pembersihan dan transformasi data, Evaluasi pola, Penyajian pengetahuan Algoritma yang digunakan: Neural Networks (NN), Decision Tree (DT), K-Nearest Neighbor (k-NN)	Attribut yang paling berpengaruh terhadap aspek demografi; Gender, Age, Student Status Performance of models (based on RMSE - Root Mean Squared Error): Neural Networks: Best (RMSE: 0.0006) k-NN: Second best (RMSE: 0.0622) Decision Tree: Least accurate (RMSE: 5.1604)	Menggunakan Manual data cleaning dengan Microsoft Excel, Analisis dan pemodelan menggunakan WEKA	Terdahulu: Memprediksi menggunakan data demografik; gender, residential area, race, state location. Algoritma: Neural Networks (NN), Decision Tree (DT) , K-Nearest Neighbor (k-NN) Baru: Memprediksi kemampuan akademik menggunakan data pre-admisi, dan data akademik Algoritma: DT, RF, SVM, XGB

No.	Judul, Penulis, Tahun	Metodologi	Hasil	Keterangan	Perbandingan Penelitian
4.	Student course grade prediction using the random forest algorithm: Analysis of predictors' importance (Nachouki et al., 2023)	Menggunakan model Random Forest regression, Feature importance diukur dengan RMSE variation untuk memprediksi nilai matakuliah yang diambil oleh mahasiswa,	Dihasilkan prediksi dengan Tingkat akurasi 90.33% prediction accuracy (MAE: 7.11, RMSE: 9.25). GPA/ IPK merupakan atribut prediksi paling penting, setelah nilai Pelajaran sekolah, kehadiran, kategori kelas.	Tools: Python (Anaconda 3), Scikit-learn, Pandas Dataset: 650 transcript nilai dengan sejumlah 15,596 record data)	Terdahulu: Memprediksi nilai akhir matakuliah Algoritma: RF Baru: kemampuan akademik menggunakan data pre-admisi, dan data akademik. Algoritma: DT, RF, SVM, XGB
5.	Predicting Student Performance to Improve Academic Advising Using the Random Forest Algorithm (Nachouki & Naaj, 2022)	Mengembangkan model prediksi GPA/ IPK menggunakan Random Forest untuk mendukung bimbingan akademik, kemungkinan mahasiswa beresiko, Dengan 2 tahap prediksi, yaitu; Random Forest classifier; dan pengukuran statistik (RMSE, MAE, MSE).	Model menghasilkan Tingkat akurasi ssebesar 91.32% dengan menggunakan data tahun ke-2, dan akurasi sebesar 92.87% dengan menggunakan data tahun ke-2 dan ke-3.	Tools: Python (Anaconda 3), scikit-learn, pandas; Kaggle dataset for model validation Dataset: 105 data mahasiswa IT program pada UAE university (2020–2022),	Terdahulu: Prediksi IPK menggunakan data IP per semester untuk menghindari kemungkinan DO. Algoritma: RF Baru: Memprediksi kemampuan akademik menggunakan data pre-admisi, dan data akademik. Algoritma: DT, RF, SVM, XGB
6.	Multiclass Prediction Model for Student Grade Prediction	Mengembangkan model prediksi multiclass untuk penentuan nilai akhir pada	Model dengan RF menghasilkan nilai f-measure tertingggi	Tools: WEKA 3.8.3, Python for data visualization	Terdahulu:

No.	Judul, Penulis, Tahun	Metodologi	Hasil	Keterangan	Perbandingan Penelitian
	Using Machine Learning (Bujang et al., 2021)	mahasiswa, menggunakan 6 algoritma ML (J48, kNN, NB, SVM, LR, RF);	sebesar of 99.5%. RF was most effective model.	Dataset: 1282 data mahasiswa dari Malaysia Polytechnic (CSA and ICS courses)	Prediksi menggunakan data akademik untuk penentuan nilai akhir mahasiswa. Algoritma: J48, kNN, NB, SVM, LR, RF Baru: Memprediksi kemampuan akademik menggunakan data pre-admisi, dan data akademik. Algoritma: DT, RF, SVM, XGB
7.	Using data mining techniques to predict student performance to support decision making in university admission systems. (Mengash, 2020)	Studi ini memprediksi prestasi mahasiswa pada saat ujian seleksi masuk pada perguruan tinggi dengan kriteria: high school grade average, Scholastic Achievement Admission Test score, and General Aptitude Test score	Penelitian ini menghitung sejumlah 2039 data set dari tahun 2016-2019. Hasil penelitian menunjukan bahwa algoritma ANN memiliki tingkat akurasi 79% mengalahkan algoritma lainny	Algoritma yang dibandingkan adalah Artificial Neural Network (ANN), Decision Trees (DT), Support Vector Machines (SVM), and Naïve Bayes	Terdahulu: Prediksi kemampuan akademik menggunakan data pre-admisi Algoritma: ANN, SVM, DT, dan NB Baru: kemampuan akademik menggunakan data pre-admisi, dan data akademik. Algoritma: DT, RF, SVM, XGB
8.	Supervised learning in the context of educational data	Menggunakan algoritma supervised learning pada EDM untuk mengetahui	RF dan DT memiliki akurasi tertinggi (~70%),	Tool:	Terdahulu: Prediksi kemampuan akademik menggunakan data akademik

No.	Judul, Penulis, Tahun	Metodologi	Hasil	Keterangan	Perbandingan Penelitian
	mining to avoid university students dropout (De Santos et al., 2019)	mahasiswa beresiko drop out Menggunakan 6 algoritma (DT, KNN, NN, SVM, NB, RF); data preprocessing (normalization, missing value handling); feature selection by accuracy contribution; model evaluation with K-Fold cross-validation (k=5)	Potensi drop out dengan mengidentifikasi kesalahan pada perkuliahan awal, dan beban kuliah	Scikit-learn (Python) for model training and evaluation dataset: 23,690 data mahasiswa pada Federal University of Sergipe (UFS), covering Computer Science, Information Systems, and Computer Engineering students from 2010–2018.	Menggunakan KNN, NN, SVM, NB, RF Baru: kemampuan akademik menggunakan data pre-admisi, dan data akademik Menggunakan DT, RF, SVM, XGB

Berdasarkan hasil telaah mendalam terhadap referensi yang telah dianalisis pada tabel 2.1. Dapat disimpulkan bahwa penelitian dalam ranah *Educational Data Mining* (EDM) untuk memprediksi kemampuan akademik mahasiswa telah banyak dilakukan, namun sebagian besar studi terdahulu cenderung menggunakan data tunggal**,** baik berupa data pre-admisi atau data akademik awal saja. Studi seperti (Aluko et al., 2018), (Miguéis et al., 2018), dan (Mengash, 2020) fokus pada pemanfaatan data pre-admisi, seperti nilai ujian masuk, nilai rapor, dan atribut demografis, yang digunakan untuk memprediksi kinerja akademik, namun tidak menggabungkannya dengan capaian akademik di awal perkuliahan. Pendekatan ini memberi gambaran awal potensi mahasiswa, tetapi sering kali kurang akurat dalam memprediksi keberhasilan akademik jangka panjang, karena mengabaikan indikator faktual dari kinerja aktual mahasiswa di lingkungan perguruan tinggi. Sebaliknya, beberapa studi seperti (De Santos et al., 2019), (Kanetaki et al., 2022), dan (Nachouki et al., 2023) hanya memanfaatkan data akademik tahun pertama untuk membuat prediksi, misalnya Indeks Prestasi (IP) semester awal dan hasil ujian mata kuliah. Pendekatan ini memang dapat menangkap performa riil mahasiswa, tetapi tidak memberi peluang intervensi dini karena data baru tersedia setelah mahasiswa menjalani perkuliahan. Akibatnya, institusi kehilangan kesempatan untuk mengidentifikasi mahasiswa berisiko sejak proses penerimaan.

Dari sisi metodologi, banyak ditemukan bahwa studi terdahulu menggunakan algoritma populer seperti Decision Tree (DT)**,** Random Forest (RF)**,** dan Support Vector Machine (SVM). Meskipun algoritma-algoritma ini efektif dalam klasifikasi, penggunaannya sering terbatas pada satu atau dua model tanpa perbandingan komprehensif. Penggunaan Extreme Gradient Boosting (XGB)**,** yang dalam berbagai literatur terbukti unggul dalam menangani data tabular, mengelola missing values, masih relatif jarang dilakukan. Studi seperti (Alyahyan & Düştegör, 2020), (Romero & Ventura, 2020), dan (S. Sarker et al., 2024) memang melibatkan XGB, tetapi umumnya dilakukan pada konteks internasional atau dataset yang berbeda karakteristiknya dari data penerimaan perguruan tinggi di Indonesia.

Berdasarkan tinjauan tersebut, terdapat tiga gap utama yang dapat diidentifikasi. Pertama, kurangnya integrasi antara data pre-admisi dan data akademik awal dalam satu kerangka prediksi yang terpadu. Integrasi ini berpotensi meningkatkan akurasi prediksi dan memungkinkan intervensi dini berbasis data, karena memadukan proyeksi kemampuan calon mahasiswa dari data ujian masuk dengan konfirmasi capaian faktual pada semester 1 dan semester 2. Kedua, belum adanya perbandingan algoritma supervised learning dengan cakupan yang lengkap, yaitu: DT, RF, SVM, dan XGB, yang dilakukan pada konteks data penerimaan perguruan tinggi di Indonesia. Padahal, perbandingan ini penting untuk memastikan model yang dipilih benar-benar optimal untuk kondisi data lokal. Ketiga, masih sedikitnya penelitian yang mengeksplorasi data mahasiswa dalam konteks SNBP (Seleksi Nasional Berdasarkan Prestasi), dimana karakteristik seleksi, bobot kriteria, dan distribusi kualitas calon mahasiswa berbeda dari mekanisme seleksi internasional yang sering diteliti dalam penelitian sebelumnya.

Secara umum, tinjauan dari penelitian sebelumnya mengenai algoritma supervised learning yang dipilih adalah sebagai berikut:

1. Decision Tree (DT) banyak diapresiasi karena interpretabilitasnya yang tinggi, sehingga mudah dipahami oleh pengambil kebijakan non-teknis (Aluko et al., 2018). Namun, kelemahannya terletak pada kecenderungan overfitting jika tidak dioptimasi dengan tepat (Sravani & Bala, 2020).
2. Random Forest (RF) merupakan pengembangan DT yang mengatasi kelemahan overfitting melalui *ensemble learning*. RF terbukti memberikan akurasi yang stabil di berbagai konteks, baik menggunakan data pre-admisi maupun data akademik awal (Gufroni et al., 2021), (Nachouki et al., 2023).
3. Support Vector Machine (SVM) unggul pada klasifikasi dengan *margin* yang jelas, terutama ketika data memiliki dimensi tinggi atau hubungan non-SVM linear antar variable (Aluko et al., 2018). Meski demikian, interpretabilitasnya rendah bagi pengguna non-teknis.

4. Extreme Gradient Boosting (XGB) secara konsisten dilaporkan sebagai algoritma dengan akurasi tertinggi pada banyak studi EDM terkini (Alyahyan, 2020), (Romero & Ventura, 2020), (Harefa, 2024), (S. Sarker et al., 2024), berkat kemampuannya menangani data dengan kompleksitas tinggi, melakukan *regularization*, dan mengelola *missing values* dengan baik.

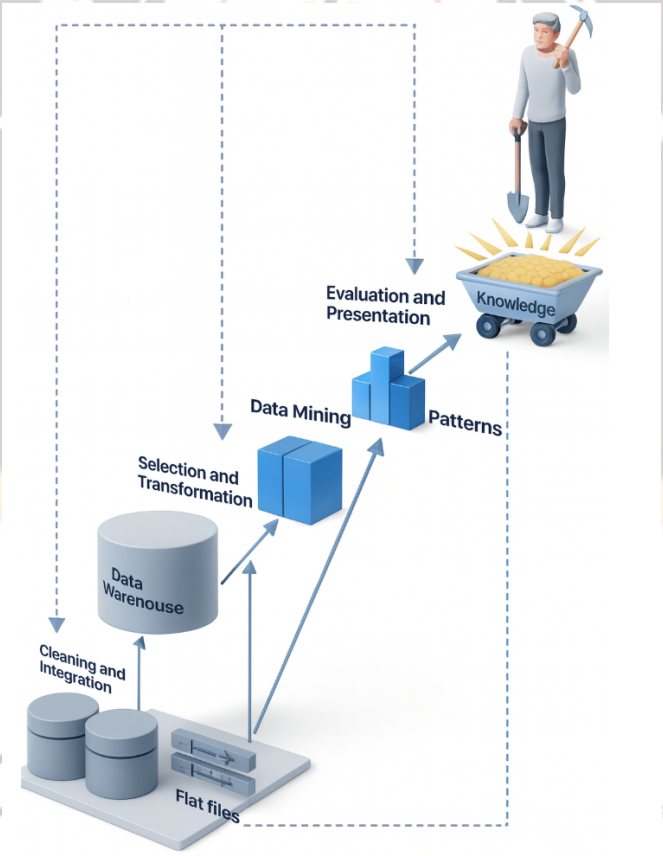
Secara metodologis, keterbaruan pada penelitian ini adalah menggabungkan data pre-admisi SNBP (nilai rapor, prestasi, atribut sekolah) dengan data akademik tahun pertama (IP semester 1 dan 2) sebagai prediktor. Dengan demikian, model yang dihasilkan mampu memberikan prediksi yang lebih akurat sekaligus memungkinkan *early warning* di awal masa studi. Selanjutnya penelitian ini melakukan perbandingan komprehensif empat algoritma supervised learning; DT, RF, SVM, dan XGB, dengan melakukan evaluasi menggunakan Confusion Matrix dan ROC-AUC. Pendekatan ini memastikan hasil yang tidak hanya akurat tetapi juga reliabel di berbagai skenario distribusi kelas. Kemudian penelitian ini dilakukan secara khusus dalam konteks seleksi mahasiswa di Indonesia, menggunakan data SNMPTN dan SNBP yang telah dilakukan sebelumnya, yang belum banyak mendapat perhatian dalam literatur internasional, sehingga hasilnya relevan secara langsung untuk perbaikan kebijakan penerimaan dan pembinaan mahasiswa di perguruan tinggi nasional.

Dengan demikian, secara umum penelitian ini memiliki kontribusi keterbaruan pada dua aspek: (1) aspek metodologis, melalui integrasi data multi-tahap dan perbandingan lintas algoritma dengan evaluasi multi-metrik; dan (2) aspek kontekstual, melalui penerapan khusus pada data SNBP dan capaian akademik awal di Indonesia. Hasil dari penelitian ini diharapkan dapat menjadi dasar pengembangan sistem prediksi kinerja akademik yang dapat diimplementasikan oleh perguruan tinggi untuk meminimalisasi risiko kegagalan studi, meningkatkan retensi mahasiswa, dan memperbaiki kebijakan penerimaan berdasarkan bukti empiris yang kuat.

2.2. Landasan Teori

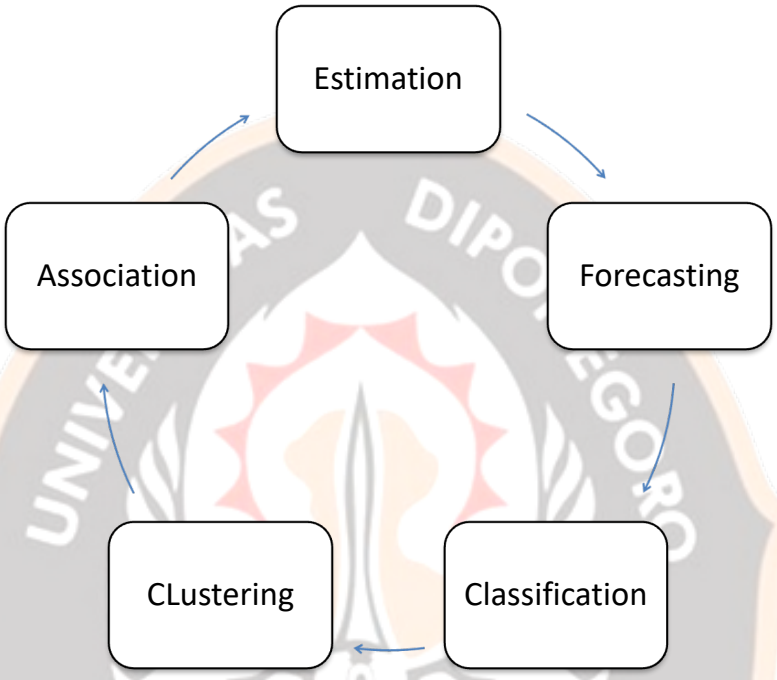
2.2.1. Data Mining

Data mining merupakan suatu proses untuk menemukan pola-pola dan hubungan tersembunyi dalam kumpulan data yang besar, dengan tujuan mendukung aktivitas seperti klasifikasi, estimasi, peramalan (*forecasting*), aturan asosiatif (*association rule*), pola berurutan (*sequential pattern*), pengelompokan (*clustering*), regresi, serta deskripsi dan visualisasi data. Proses ini juga dikenal sebagai *Knowledge Discovery from Data (KDD)*, yang terdiri dari beberapa tahapan: *data cleaning*, *data integration*, *data selection*, *data transformation*, *pattern evaluation*, dan *knowledge presentation* (Han et al., 2012).



Gambar 2.1. Ilustrasi Proses Knowledge Discovery from Data.

Data mining dapat dikategorikan ke dalam beberapa kelompok berdasarkan fungsinya, yaitu: fungsi estimasi, fungsi prediksi, fungsi klasifikasi, fungsi pengelompokan (*clustering*), dan fungsi asosiasi (Larose, 2006).

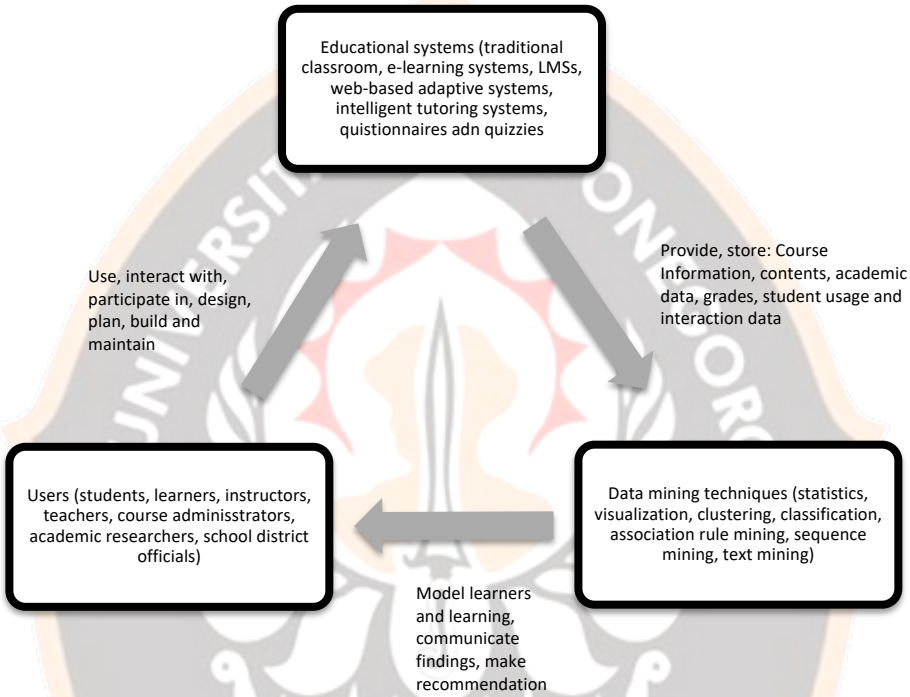


Gambar 2.2. Fungsi Data Mining.

2.2.2. Educational Data Mining (EDM)

EDM menggunakan proses penemuan pengetahuan yang iteratif yang dapat dilihat pada gambar 2.2. Proses ini berlangsung dalam berbagai lingkungan pendidikan seperti kelas konvensional, platform e-learning, serta sistem pembelajaran berbasis web yang adaptif dan cerdas. Lingkungan tersebut menghasilkan data mentah berupa interaksi mahasiswa, informasi mata kuliah, dan data akademik lainnya. Educational Data Mining (EDM) mencakup tahapan perumusan, pengujian, dan penyempurnaan hipotesis. Hipotesis ini dikembangkan dari data berjumlah besar yang dihasilkan oleh berbagai jenis lingkungan pendidikan. Tahapan awal dalam EDM adalah validasi data—yakni mengidentifikasi hubungan antar variabel, parameter, dan item data—yang juga

disebut pra-pemrosesan. Setelah tahap ini, data diolah melalui teknik data mining, lalu hasilnya diinterpretasikan dan disampaikan kepada pemangku kepentingan di bidang pendidikan. Rekomendasi kemudian diberikan untuk menyelesaikan permasalahan atau meningkatkan kualitas proses pembelajaran (Romero & Ventura, 2020)



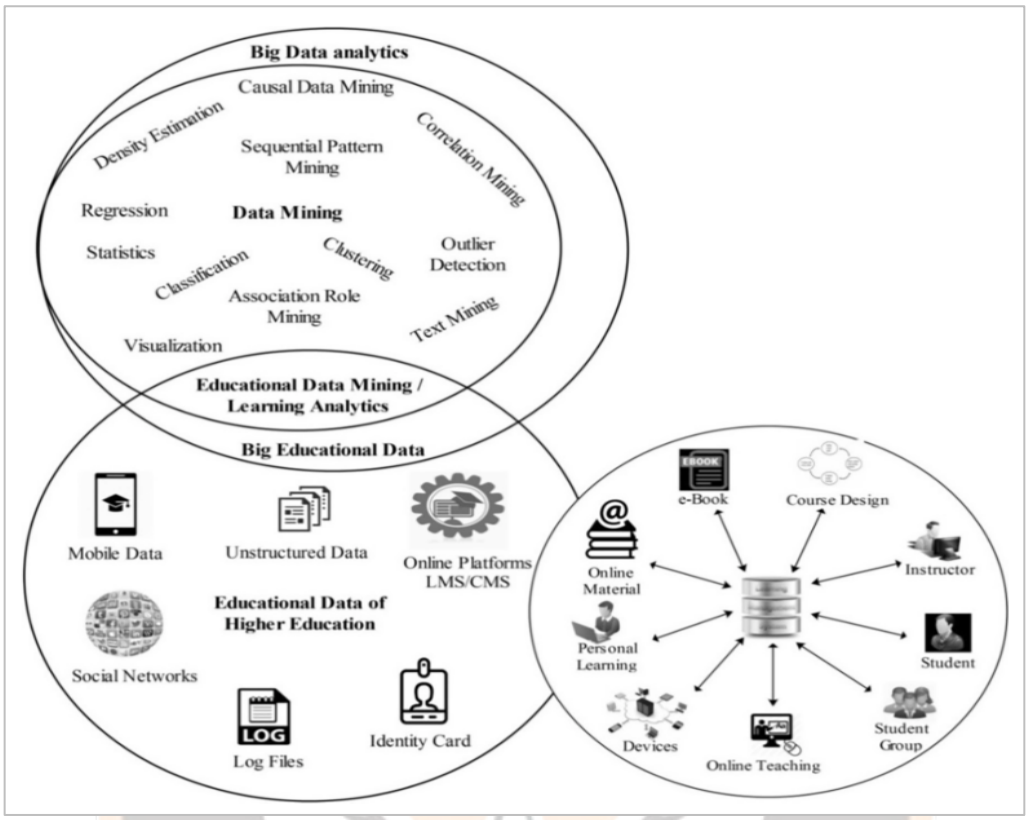
Gambar 2.3. Desain Educational Data Mining (EDM).

Tujuan utama Educational Data Mining (EDM) di lingkungan perguruan tinggi meliputi:

1. Memprediksi perilaku belajar mahasiswa di masa mendatang dengan membangun model siswa. Model ini dirancang untuk mencakup berbagai karakteristik individu mahasiswa, seperti tingkat pengetahuan, pola perilaku, serta motivasi mereka dalam proses pembelajaran.
2. Mengidentifikasi dan menyempurnakan model domain**,** yaitu melalui penerapan beragam pendekatan dan teknik EDM untuk menemukan model pembelajaran baru atau meningkatkan model yang telah ada sebelumnya.

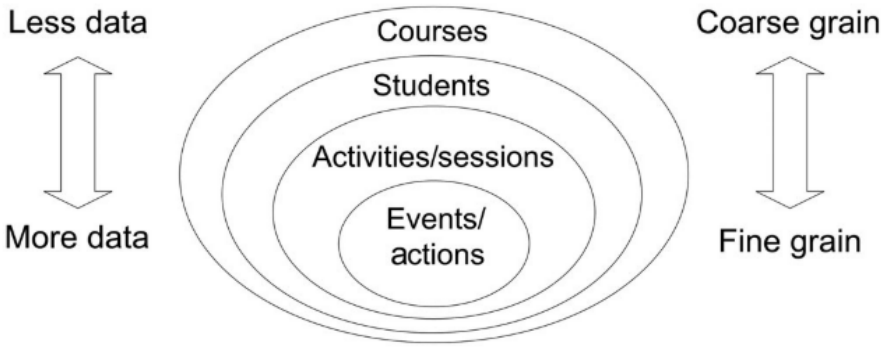
3. Menganalisis pengaruh intervensi pendidikan terhadap hasil dan efektivitas sistem pembelajaran yang diterapkan.
4. Menghasilkan pengetahuan ilmiah tentang proses belajar, melalui integrasi antara model siswa, teknologi pendukung, dan perangkat lunak pembelajaran yang digunakan.

Penerapan *Educational Data Mining* (EDM) berlangsung dalam sebuah siklus yang berulang, mencakup tahapan perumusan hipotesis, pengujian, hingga perbaikan dalam konteks pendidikan. EDM memiliki tiga komponen utama, yaitu: sistem pendidikan, metode data mining, dan pengguna. Sistem pendidikan meliputi berbagai bentuk pembelajaran seperti kelas konvensional, e-learning, Learning Management System (LMS), sistem adaptif berbasis web, *intelligent tutoring systems*, serta instrumen evaluasi seperti kuesioner dan kuis. Teknik-teknik data mining yang diterapkan mencakup statistik, visualisasi data, *clustering*, *classification*, *association rule mining*, *sequence mining*, dan *text mining*. Pengguna EDM mencakup berbagai pihak, seperti mahasiswa, pelajar, dosen, guru, staf administrasi, peneliti, serta pemangku kebijakan pendidikan (Romero & Ventura, 2020).



Gambar 2.4. Penggunaan Educational Data Mining pada Pendidikan Tinggi (Aldowah et al., 2019).

Data pendidikan diperoleh dari berbagai sumber, termasuk interaksi antara siswa, pengajar, dan sistem pembelajaran; misalnya melalui perilaku navigasi, jawaban kuis, tingkat keterlibatan, hingga percakapan dalam forum diskusi. Selain itu, data juga mencakup informasi administratif (seperti data sekolah dan guru), data demografis (seperti jenis kelamin dan usia), serta faktor-faktor yang berkaitan dengan efektivitas belajar siswa, seperti motivasi dan kondisi emosional. Lingkungan pendidikan memiliki kemampuan untuk mengakumulasi data dalam jumlah besar dari berbagai sumber, dengan format serta tingkat detail (granularitas) yang beragam (Romero & Ventura, 2020).



Gambar 2.5. Level granulitas sesuai dengan besaran jumlah data (Romero & Ventura, 2020).

Educational Data Mining (EDM) merupakan suatu pendekatan yang digunakan untuk menggali dan memanfaatkan data yang dihasilkan dari aktivitas pembelajaran guna memenuhi berbagai kebutuhan dalam bidang pendidikan (Aldowah et al., 2019).

2.2.3. CRoss-Industry Standard Process for Data Mining (CRISP-DM)

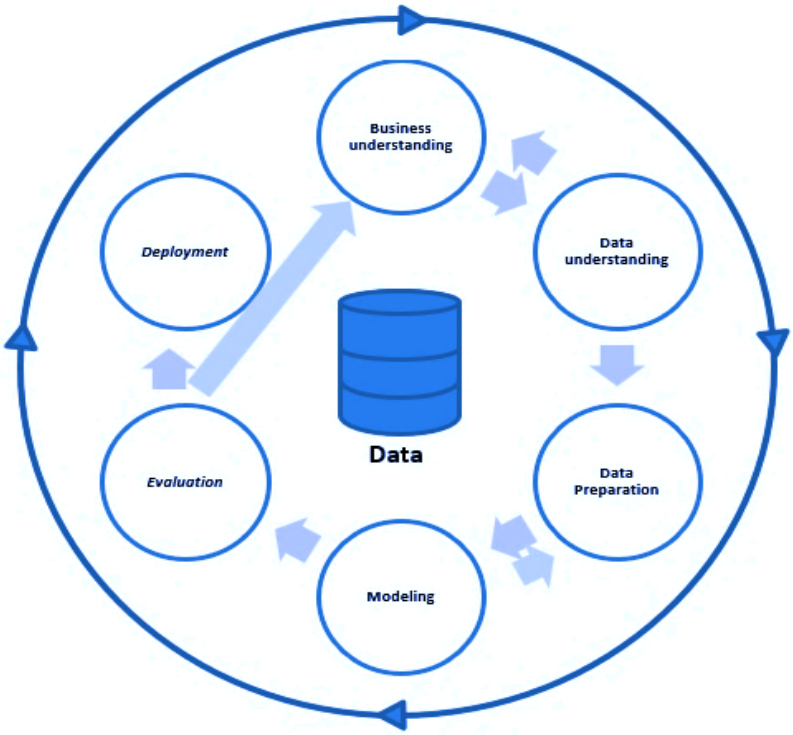
CRISP-DM (Cross-Industry Standard Process for Data Mining) merupakan metodologi standar yang digunakan untuk menerapkan teknik data mining dalam menyelesaikan permasalahan organisasi atau bisnis. Pendekatan ini dirancang berdasarkan praktik terbaik yang direkomendasikan oleh para ahli dan praktisi, dengan tujuan meningkatkan keberhasilan dalam pelaksanaan proyek data mining melalui langkah-langkah yang sistematis dan mudah diikuti. Metodologi ini akhirnya dirumuskan dalam bentuk standar yang dikenal sebagai CRISP-DM. Dalam penelitian ini, CRISP-DM digunakan sebagai kerangka kerja pemecahan masalah melalui enam tahapan utama menurut (Larose, 2006) yaitu:

1. Fase Pemahaman Bisnis (Business Understanding):
   a. Menentukan secara rinci tujuan dan kebutuhan proyek berdasarkan konteks organisasi atau penelitian.
   b. Mengonversi tujuan tersebut menjadi permasalahan data mining yang terdefinisi dengan jelas.
   c. Menyusun strategi awal untuk mencapai tujuan tersebut.

2. Fase Pemahaman Data (Data Understanding):
    a. Mengumpulkan data yang relevan.
    b. Melakukan eksplorasi data awal guna mendapatkan pemahaman mendasar.
    c. Mengevaluasi kualitas data.
    d. Jika diperlukan, memilih subset data yang berpotensi menunjukkan pola penting.
3. Fase Persiapan Data (Data Preparation):
    a. Menyiapkan data awal sebagai dasar untuk tahap-tahap berikutnya, yang membutuhkan usaha signifikan.
    b. Menentukan kasus dan variabel yang relevan dengan jenis analisis yang akan dilakukan.
    c. Melakukan modifikasi pada variabel sesuai kebutuhan.
    d. Menyusun data agar siap digunakan dalam pemodelan.
4. Fase Pemodelan (Modelling):
    a. Memilih dan mengimplementasikan metode pemodelan yang sesuai.
    b. Menyesuaikan parameter model untuk hasil yang optimal.
    c. Menyadari bahwa berbagai teknik bisa diterapkan pada permasalahan yang serupa.
    d. Jika perlu, kembali ke tahap persiapan data untuk menyesuaikan dengan kebutuhan teknik tertentu.
5. Fase Evaluasi (Evaluation):
    a. Menilai performa model yang telah dibangun untuk memastikan efektivitas dan kualitasnya.
    b. Memastikan bahwa model sesuai dengan tujuan awal proyek.
    c. Mengidentifikasi kemungkinan masalah bisnis atau penelitian yang belum terjawab.
    d. Mengambil keputusan tentang kelanjutan penggunaan hasil data mining.
6. Fase Penyebaran (Deployment):
    a. Mengimplementasikan model yang telah dikembangkan, karena pembuatan model bukanlah tahap akhir proyek.

b. Salah satu bentuk implementasi bisa berupa pembuatan laporan atau sistem prediktif.

Siklus proses CRISP-DM dapat dilihat pada gambar 2.6. berikut:



Gambar 2.6. Proses CRISP-DM (Cross-Industry Standard Process for Data Mining).

2.2.4. Prediksi Kemampuan Akademik Mahasiswa

Prediksi kemampuan akademik mahasiswa merupakan bagian dari prediksi kemampuan mahasiswa. Kemampuan akademik juga disebut hasil akademik, prestasi akademik. Sebagian besar institusi dan universitas menggunakan nilai akhir siswa sebagai kriteria kemampuan akademik siswa. Kemampuan siswa tergantung pada berapa banyak nilai yang diperoleh siswa dalam ujian akhir (Romero & Ventura, 2020), (S. Sarker et al., 2024).

Kemampuan akademik siswa perlu dipantau dengan cermat untuk membantu lembaga mengidentifikasi siswa yang berisiko gagal, mencegah mereka drop-out atau lulus terlambat (Nachouki & Naaj, 2022). Kemampuan akademik yang buruk menjadi indikator mahasiswa kesulitan dalam menyesuaikan diri dengan perguruan

tinggi dan berpeluang besar untuk putus sekolah/ drop out. Beberapa dampak negatif dari buruknya kemampuan akademik dan tingginya drop out diantaranya adalah; menambah biaya pendidikan yang sia-sia (Costa et al., 2017), pengurangan kesempatan hidup secara profesional dan sosial yang berdampak negatif kepada keluarga dan masyarakat sulitnya perguruan tinggi dalam mendapatkan akreditasi yang baik, hilangnya kepercayaan dan motivasi untuk melanjutkan Pendidikan, dan lain sebagainya. para peneliti pada umunya menggunakan IPK untuk mengukur kemampuan akademik mahasiswa, mahasiswa dengan IPK baik maka kemampuan mereka baik dan mahasiswa dengan IPK buruk maka kemampuan mereka adalah buruk (Aluko et al., 2018).

2.2.5. Pembelajaran Mesin

Pembelajaran Mesin (*Machine learning*) merupakan salah satu cabang dari kecerdasan buatan (*Artificial Intelligence*) yang menitikberatkan pada pengembangan sistem dan algoritma berbasis teknik statistik. Tujuannya adalah memungkinkan komputer belajar dari data serta pengalaman sebelumnya. Fokus utama dari pembelajaran mesin adalah menciptakan model yang mampu secara otomatis meningkatkan kinerjanya saat menerima data baru, tanpa perlu melakukan pemrograman secara manual dalam skala besar (I. H. Sarker, 2021).

Dalam prosesnya, algoritma *machine learning* dilatih menggunakan data yang mencakup contoh masukan dan hasil yang diharapkan. Melalui data tersebut, algoritma mampu mengenali pola, hubungan, serta fitur-fitur penting di dalamnya. Setelah pelatihan selesai, model yang terbentuk dapat digunakan untuk membuat prediksi atau mengambil keputusan terhadap data baru yang sebelumnya belum pernah diolah. Secara umum, terdapat beberapa jenis pendekatan dalam model pembelajaran mesin yang masing-masing memiliki karakteristik tersendiri, yaitu: (Cunningham et al., 2008).

a. Supervised Learning

   Supervised learning adalah pendekatan yang bertujuan untuk membangun pemetaan antara data masukan dan keluaran berdasarkan contoh-contoh yang

tersedia dalam dataset. Dataset ini dibagi menjadi data pelatihan dan data pengujian. Data pelatihan mencakup label atau variabel target yang ingin diprediksi, biasanya dalam bentuk klasifikasi. Algoritma belajar dari pola yang ada dalam data pelatihan, kemudian menerapkan pemahaman tersebut untuk memprediksi data pengujian.

b. Unsupervised Learning

Unsupervised learning merupakan metode yang digunakan untuk mengelompokkan data tanpa menggunakan label atau keluaran yang sudah diketahui sebelumnya. Pendekatan ini berfokus pada penemuan pola, struktur, atau kelompok tersembunyi dalam data untuk mengenali klasifikasi secara otomatis.

c. Semi-supervised Learning

Pendekatan ini menggabungkan elemen dari *supervised* dan *unsupervised learning*. Metode ini menggunakan kombinasi data yang sudah diberi label dan data yang belum berlabel dalam proses pelatihan. Tujuannya adalah agar algoritma dapat mempelajari pola dari data yang terbatas jumlah labelnya dan kemudian memperluas pembelajaran secara mandiri terhadap data lainnya.

d. Reinforcement Learning

Reinforcement learning adalah metode yang berfokus pada pengambilan keputusan secara berurutan dengan tujuan untuk memaksimalkan hasil atau *reward*. Model belajar melalui proses trial-and-error, di mana tindakan yang diambil akan mendapatkan umpan balik, sehingga perilaku algoritma disesuaikan secara bertahap untuk mencapai hasil yang optimal dalam situasi tertentu.

2.2.6. Decision Tree (DT)

Metode *Decision Tree* (DT) atau pohon keputusan merupakan representasi berbentuk diagram alur yang menyerupai struktur pohon, di mana setiap simpul (*node*) mewakili atribut yang diuji, cabang menggambarkan hasil dari pengujian tersebut, dan simpul akhir menunjukkan kelas yang diprediksi (Han et al., 2012).

Pohon keputusan dikenal sebagai teknik yang sangat populer dan andal dalam tugas klasifikasi dan prediksi. Metode ini mampu mengubah kumpulan data besar menjadi struktur pohon yang merepresentasikan seperangkat aturan keputusan. Aturan-aturan tersebut mudah dipahami karena dapat disampaikan dalam bahasa alami, dan juga dapat diterjemahkan ke dalam bahasa kueri seperti SQL (*Structured Query Language*) untuk menelusuri data dalam kategori tertentu.

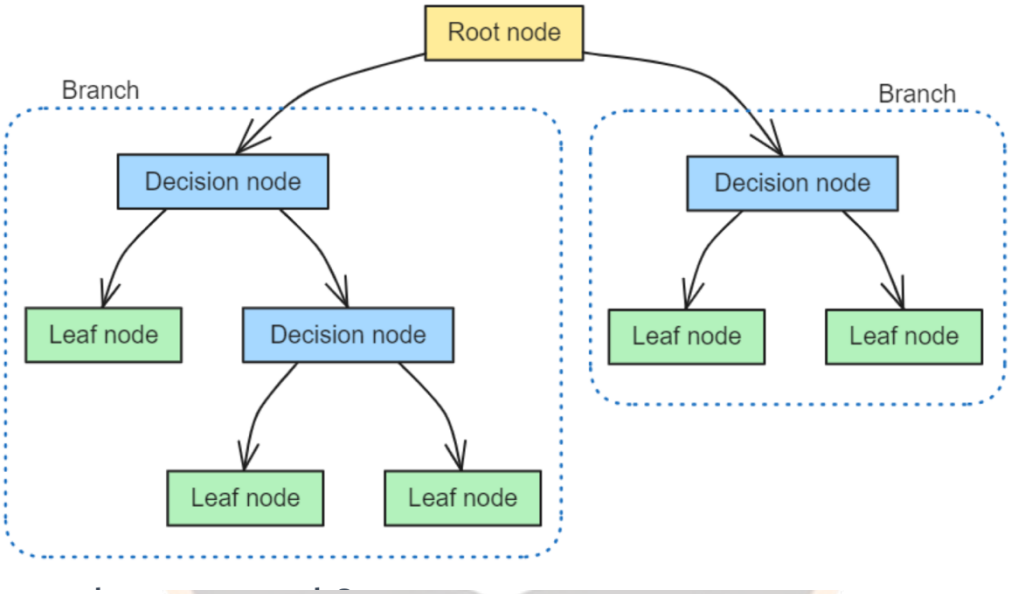
Dalam struktur decision tree, terdapat tiga jenis simpul utama:

1. Root Node (simpul akar): Node utama di bagian atas pohon. Node ini tidak menerima input, dan bisa saja tidak memiliki output atau memiliki lebih dari satu output.
2. Internal Node (simpul internal): Node percabangan yang menerima satu input dan menghasilkan dua atau lebih output.
3. Leaf Node atau Terminal Node (simpul daun): Node akhir dari sebuah cabang, yang hanya memiliki satu input dan tidak memiliki output selanjutnya.

Menurut (Frank, 2017), *decision tree* bekerja berdasarkan aturan *if-then*, namun tidak memerlukan penggunaan parameter maupun metrik tertentu. Sifatnya yang sederhana dan mudah dipahami membuat metode ini efektif dalam menangani permasalahan dengan atribut yang beragam. Selain itu, *decision tree* juga memiliki kemampuan untuk mengatasi data yang tidak lengkap (*missing values*) maupun data yang mengandung gangguan (*noise*). Pembuatan pohon keputusan melibatkan beberapa langkah penting, yaitu:

1. Menyiapkan data latih yang telah diklasifikasikan ke dalam kelompok atau kelas tertentu sebagai dasar pelatihan model.
2. Menentukan node akar dari pohon, yang dipilih berdasarkan atribut dengan nilai *gain* tertinggi. Untuk menemukan atribut ini, terlebih dahulu dilakukan perhitungan nilai *entropy*, kemudian dilanjutkan dengan menghitung nilai *gain* dari setiap atribut. Atribut dengan *gain* paling besar akan dijadikan sebagai akar pohon keputusan.

Decision tree (DT) merupakan algoritma berupa alat pendukung keputusan yang diwakili oleh grafik atau model keputusan seperti pohon yang mewakili hasil yang mungkin terjadi (Nowozin et al., 2011). Algoritma ini merupakan suatu pendekatan, yang berisi pernyataan kondisional dalam suatu algoritma. DT membuat sebuah pohon keputusan menggunakan kriteria entropi untuk menentukan pemisahan data yang akan dibentuk pada proses pelatihan. Setelah pohon keputusan terbentuk, DT melakukan prediksi pada data baru dengan melewati pohon dari akar ke daun berdasarkan nilai setiap fitur pada data yang digunakan. Ilustrasi dari proses pada algoritma DT dapat dilihat pada Gambar 2.7 sebagai berikut.



Gambar 2.7. Ilustrasi Proses Decision Tree (David Andrés, 2023).

Dalam bentuk matematis, prediksi hasil (Y) dapat direpresentasikan sebagai fungsi berulang yang melibatkan kondisi (Q) pada setiap simpul n pada pohon keputusan. Berikut perhitungan matematis dari prediksi yang dilakukan DT pada Formula (2.1).

$$Y(X) = \sum_{n=1}^{N} Q_n(X).c_n \qquad (2.1)$$

N : jumlah simpul dalam pohon

$Q_n(X)$: fungsi yang menghasilkan 1 jika sampel X memenuhi kondisi pada simpul n, dan 0 jika sebaliknya

c_n : nilai prediksi pada simpul n.

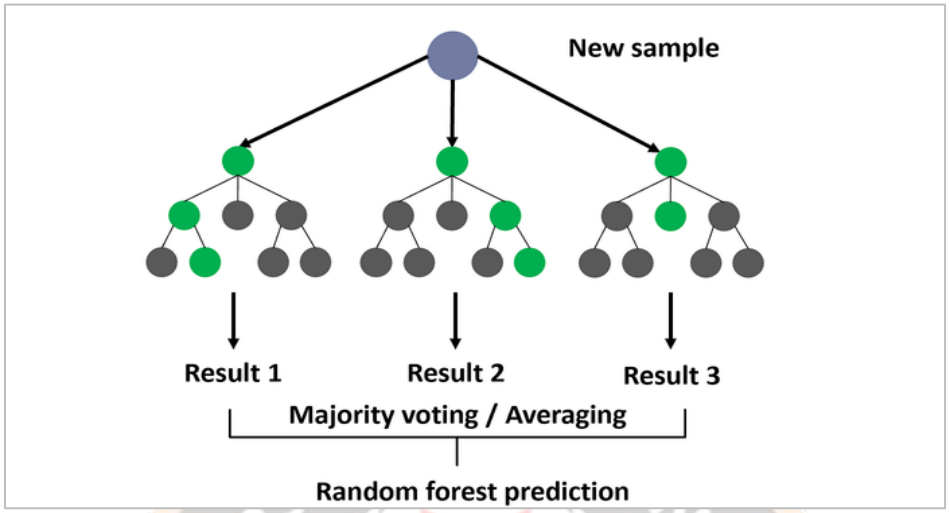
2.2.7. Random Forest (RF)

Salah satu algoritma *supervised learning* yang terkenal adalah Random Forest (RF), yang diperkenalkan oleh Breiman pada tahun 2001. Algoritma ini banyak digunakan untuk menyelesaikan tugas-tugas seperti klasifikasi dan regresi. Keunikan Random Forest terletak pada dua aspek yang membuatnya bersifat acak, yaitu:

1. Setiap pohon dalam model dibangun menggunakan sampel data latih yang diambil secara acak dengan metode *bootstrap*.
2. Pada setiap percabangan (*split*) dalam proses pembentukan *decision tree*, hanya sebagian fitur dari keseluruhan variabel yang dipilih secara acak, dan fitur terbaik dari subset tersebut digunakan untuk pembagian data pada node tersebut.

Algoritma ini merupakan gabungan dari sejumlah *decision tree* atau pohon keputusan, di mana masing-masing pohon dibangun berdasarkan *random vector* yang dipilih secara acak dan merata di seluruh pohon dalam satu *forest*. Prediksi akhir dari *Random Forest* ditentukan melalui mekanisme suara terbanyak (voting) untuk kasus klasifikasi, atau melalui perhitungan rata-rata pada kasus regresi. *Random Forest* termasuk dalam kategori algoritma *ensemble learning*, yaitu pendekatan yang menggabungkan beberapa model untuk menyelesaikan berbagai jenis tugas seperti klasifikasi, regresi, dan lainnya. Dalam proses pelatihannya, algoritma ini membentuk banyak pohon keputusan, dan hasil akhirnya merupakan gabungan dari output setiap pohon, baik itu rata-rata prediksi (untuk regresi) maupun modus dari hasil klasifikasi (untuk klasifikasi) (Frank, 2017).

Setiap pohon dalam *forest* memberikan kontribusi terhadap keputusan klasifikasi akhir berdasarkan atribut yang dimiliki oleh kasus baru, meskipun terdapat risiko *overfitting* pada data pelatihan. Ilustrasi dari proses yang terjadi pada RF ditunjukkan pada gambar 2.8.



Gambar 2.8. Ilustrasi Random Forest.

Model matematis dari RF dapat dilihat pada formula (2.2) sebagai berikut.

$$Y_{RF}(X) = argmax_c \left(\sum_{n=1}^{N} 1(T_i(X) = c) \right) \quad (2.2)$$

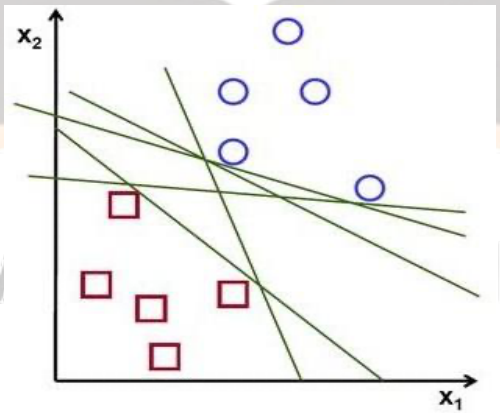
N : jumlah pohon dalam RF
X : data input yang akan diprediksi
$T_i(X)$: prediksi dari pohon keputusan ke-i
c : kelas prediksi
1(.) : fungsi indikator yang menghasilkan 1 jika pernyataan benar, dan 0 sebaliknya.

Random Forest memiliki mekanisme internal untuk memperkirakan tingkat kesalahan generalisasi secara otomatis, yang dikenal dengan istilah *out-of-bag (OOB) error estimate*. Dalam proses pembangunan pohon, hanya sekitar dua pertiga dari data pelatihan yang digunakan melalui teknik *bootstrap sampling*, sementara sepertiga sisanya digunakan untuk menguji akurasi model. Estimasi OOB error diperoleh dari rata-rata kesalahan prediksi terhadap setiap instance pelatihan yang tidak disertakan dalam pembentukan pohon yang bersangkutan. Selama pelatihan Random Forest, seluruh data pelatihan akan melewati setiap pohon yang terbentuk, dan sistem akan menghitung matriks kedekatan antar instance berdasarkan frekuensi mereka berada pada *terminal node* yang sama.

Berbagai studi menunjukkan bahwa Random Forest memberikan performa prediktif yang tinggi, baik untuk tugas klasifikasi maupun regresi, dalam berbagai bidang seperti keuangan, penginderaan jauh (*remote sensing*), serta analisis genetik dan biomedis. Selain itu, Random Forest juga sering menunjukkan keunggulan dibandingkan metode lain seperti *partial least squares regression*, *support vector machine*, maupun *neural network* (Genuer et al., 2017).

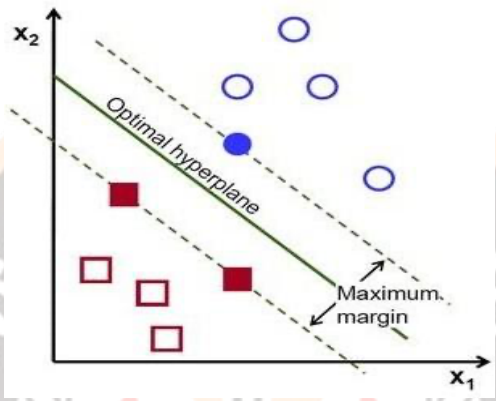
2.2.8. Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan salah satu metode dalam *machine learning* yang digunakan untuk tugas klasifikasi, baik pada data yang bersifat linear maupun non-linear. Untuk menangani data non-linear, SVM menggunakan teknik yang disebut *kernel trick*, yaitu suatu fungsi yang mentransformasikan data dari ruang fitur berdimensi rendah ke ruang fitur berdimensi lebih tinggi, sehingga memungkinkan pemisahan antar kelas secara linear melalui suatu *hyperplane* (Han et al., 2012). SVM memiliki kesamaan dengan *neural network* dalam hal fungsi serta jenis permasalahan yang dapat diselesaikan. Keduanya termasuk ke dalam kategori *supervised learning*. Prinsip utama dari SVM adalah mencari dan memaksimalkan margin pemisah (*hyperplane*) antara kelas-kelas data yang berbeda, seperti dapat dilihat pada gambar 2.9. di berikuti:



Gambar 2.9. Kemungkinan *Hyperplane* dalam model klasifikasi.

Gambar 2.9. menggambarkan adanya berbagai kemungkinan letak *hyperplane* yang dapat digunakan untuk memisahkan data. Sementara itu, Gambar 2.10. menunjukkan *hyperplane* yang dipilih berdasarkan margin pemisah maksimum, yang memberikan jarak terjauh antara kelas data yang berbeda.



Gambar 2.10. Optimal Hyperplane dengan Maximum Margin.

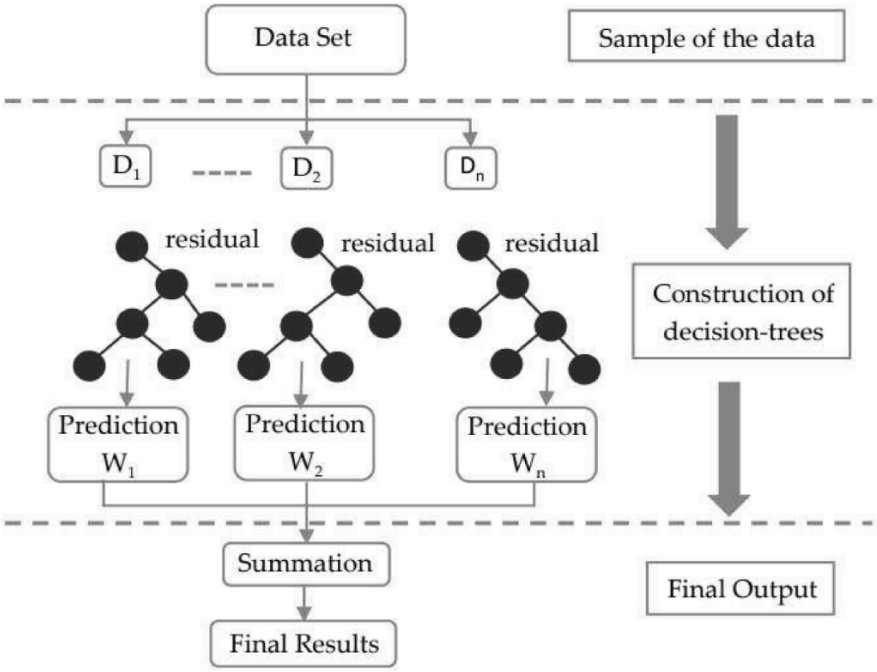
Hyperplane atau batas keputusan terbaik antara dua kelas data ditentukan dengan cara mengukur margin-nya, yaitu jarak antara *hyperplane* dan titik data terdekat dari masing-masing kelas. Titik-titik data yang paling dekat inilah yang disebut sebagai *support vector*. Awalnya, pendekatan *machine learning* dikembangkan berdasarkan asumsi bahwa data bersifat linear, sehingga algoritma yang dihasilkan hanya efektif untuk kasus-kasus linear. Namun, karena data dalam dunia nyata umumnya bersifat non-linear, maka SVM dikembangkan lebih lanjut dengan menggunakan fungsi *kernel*. Fungsi ini memungkinkan pemetaan data dari ruang berdimensi rendah ke ruang berdimensi lebih tinggi, sehingga data yang semula tidak dapat dipisahkan secara linear menjadi dapat dipisahkan. Kelebihan dari metode ini adalah bahwa dalam proses pembelajaran, SVM hanya memerlukan informasi mengenai jenis fungsi *kernel* yang digunakan tanpa harus mengetahui bentuk eksplisit dari fungsi non-linear tersebut. Fungsi matematis SVM ditunjukan pada formula (2.3) berikut: (Rezvani et al., 2024)

$$f(x) = w.x + b \tag{2.1}$$

w : vektor bobot
x : vektor input
b : bias
$w.x + b$: dot produk antara w dan x.
Jika $f(x) > 0$ maka model memprediksi kelas 1
Jika $f(x) > 0$ maka model memprediksi kelas -1

2.2.9. Extreme Gradient Boosting (XGB)

Extreme Gradient Boosting (XGB) adalah algoritma ensemble machine learning yang sangat populer dan kuat. Algoritma XGB ini adalah pengembangan dari algoritma Gradient Boosting yang telah ada sebelumnya, dengan tingkat implementasi yang sangat efisien dan efektif dari teknik ensemble boosting yang bertujuan untuk meningkatkan kinerja model prediksi, bekerja dengan cara berulang kali menambahkan pohon baru ke model sejauh pohon tersebut memperbaiki hasil prediksi secara berurutan (Velthoen et al., 2023). Setiap pohon yang ditambahkan mencoba untuk memperbaiki kesalahan prediksi model sebelumnya. Hal ini dilakukan dengan menghitung gradient dari loss function terhadap prediksi model sebelumnya. Ilustrasi dari proses XGB dapat ditinjau pada gambar 2.11. sebagai berikut.



Gambar 2.11. Struktur algoritma Extreme Gradient Boosting (XGB) (K. Khan et al., 2022).

Fungsi matematis dari XGB dapat dilihat pada formula (2.4) berikut.

$$Objective = \sum_{i=1}^{n} Binary_{Cross_{Entropy(y_i,\hat{y}_i)}} + \sum_{j=1}^{T} \Omega(f_j) \quad (2.4)$$

$\hat{y}_i$: prediksi model untuk sampel ke-i
$Binary_{Cross_{Entropy(y_i,\hat{y}_i)}}$: fungsi untuk klasifikasi biner
$\Omega(f_j)$: fungsi regularisasi untuk pohon keputusan ke-i
T : jumlah pohon.

2.2.10. Confusion Matrix

Confusion matrix adalah teknik yang digunakan untuk mengevaluasi kinerja suatu algoritma klasifikasi. Matriks ini memberikan informasi perbandingan antara hasil prediksi sistem dengan nilai sebenarnya, sehingga tingkat akurasi model dapat diukur. Selain itu, *confusion matrix* juga berfungsi sebagai alat visualisasi yang

menggambarkan hasil pembelajaran model, yang biasanya mencakup dua atau lebih kategori (Luque et al., 2019). Contoh visualisasi *confusion matrix* dapat dilihat pada Gambar 2.12.

		Prediksi	
		Positive	Negative
Aktual	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Gambar 2.12. Tabel *Confusion Matrix*

Keterangan:

a. True Positive (TP) : Jumlah data yang memang termasuk dalam kelas positif dan berhasil diklasifikasikan dengan benar sebagai positif oleh model.

b. True Negative (TN) : Jumlah data yang memang berasal dari kelas negatif dan diklasifikasikan secara tepat sebagai negatif.

c. False Positive (FP) : Jumlah data yang seharusnya termasuk dalam kelas negatif, namun secara keliru diklasifikasikan sebagai positif oleh model.

d. False Negative (FN) : Jumlah data yang sebenarnya merupakan kelas positif, tetapi salah diklasifikasikan sebagai negatif.

a. Accuracy

Accuracy merupakan salah satu metrik evaluasi yang digunakan untuk mengukur kinerja model dengan melihat sejauh mana hasil prediksi mendekati nilai aktual. Secara umum, akurasi menggambarkan tingkat ketepatan model dalam mengklasifikasikan data dengan benar. Nilai akurasi dapat dihitung menggunakan formula (2.5), sebagai berikut:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2.5)$$

b. Precision

Precision atau presisi adalah rasio antara jumlah data relevan yang berhasil diprediksi dengan benar oleh sistem terhadap total keseluruhan data yang diprediksi sebagai relevan. Dengan kata lain, presisi menunjukkan tingkat ketepatan hasil

prediksi model dalam mengidentifikasi data yang benar-benar sesuai dengan kategori yang diminta. Nilai presisi dapat dihitung menggunakan menggunakan rumus yang ditunjukan pada formula (2.6), sebagai berikut:

$$Precision = \frac{TP}{TP + FP} \tag{2.6}$$

c. Recall

Recall adalah perbandingan antara jumlah data relevan yang berhasil diidentifikasi oleh sistem dengan total keseluruhan data relevan yang ada dalam dataset. Recall mencerminkan kemampuan model dalam mendeteksi atau menemukan semua informasi yang seharusnya dikenali. Recall dapat dihitung menggunakan formula (2.7), sebagai berikut:

$$Recall = \frac{TP}{TP + FN} \tag{2.7}$$

d. F1-Score

F1-Score adalah metrik evaluasi yang digunakan untuk menilai kinerja model dengan menghitung rata-rata harmonis antara nilai presisi dan recall, sehingga mencerminkan keseimbangan antara kedua ukuran tersebut. Perhitungan *F1-Score* dapat dilakukan menggunakan formula (2.8) sebagai berikut:

$$F1 - Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \tag{2.8}$$

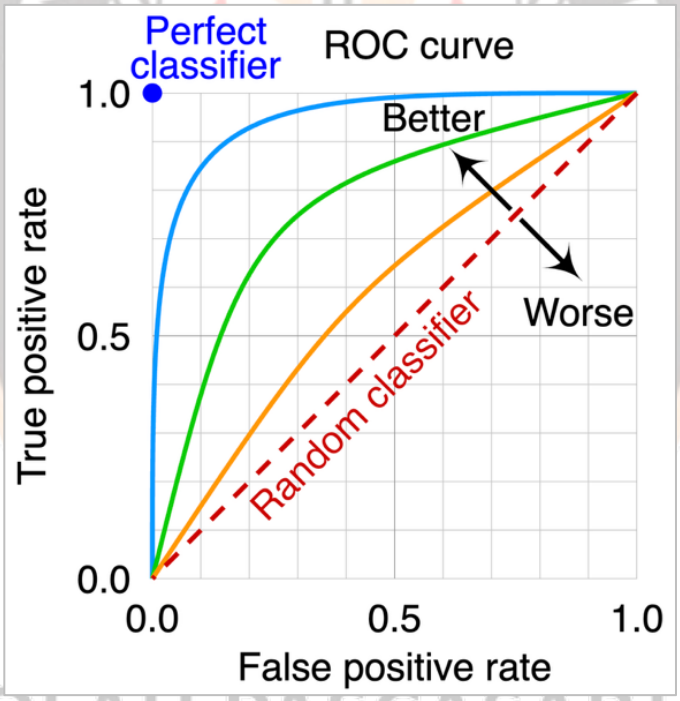
2.2.11. ROC – AUC Curve

ROC atau Receiver Operating Characteristic adalah kurva yang digunakan sebagai alat evaluasi performa model klasifikasi, khususnya dalam membedakan antara dua kelas (biner). Kurva ROC menggambarkan hubungan antara True Positive Rate (TPR) dengan False Positive Rate (FPR) pada berbagai nilai threshold (ambang batas) yang digunakan dalam proses klasifikasi. Kurva ROC diperoleh dengan memplot TPR terhadap FPR untuk setiap threshold. TPR dan FPR dapat dihitung dengan rumus (2.9) dan (2.10) berikut:

$$TPR = TP / (TP + FN) \quad (2.9)$$

$$FPR = FP / (FP + TN) \quad (2.10)$$

AUC atau Area Under the Curve adalah nilai yang mewakili luas area di bawah kurva ROC. AUC digunakan sebagai ringkasan numerik untuk mengevaluasi seberapa baik model mampu membedakan antara kelas positif dan negatif. Nilai AUC berada dalam rentang 0 hingga 1, dengan interpretasi semakin besar nilai AUC, semakin baik performa model dalam melakukan klasifikasi yang benar terhadap data dari dua kelas yang berbeda. Kurva ROC-AOC dapat dilihat pada gambar 2.13.



Gambar 2.13. ROC -AUC Curve

Pada Gambar 2.13. terlihat bahwa garis diagonal pada kurva ROC mewakili tebakan secara acak (*random guessing*); setiap kurva yang berada di atas garis tersebut menunjukkan performa yang lebih baik dari tebakan acak. Semakin dekat

kurva ke sudut kiri atas, semakin tinggi performa model. Sensitivitas dan spesifisitas biasanya memiliki hubungan yang berlawanan. Mengubah nilai ambang klasifikasi yaitu pada nilai probabilitas berapa sebuah prediksi dianggap positif) akan memengaruhi kedua metrik tersebut. Titik operasi (*operating point*) pada kurva ROC dapat dipilih dengan mempertimbangkan keseimbangan antara sensitivitas dan spesifisitas. Ketika nilai ambang klasifikasi diturunkan, sensitivitas cenderung meningkat (menangkap lebih banyak kasus positif), tetapi spesifisitas bisa menurun (lebih banyak false positive). Sebaliknya, ketika ambang dinaikkan, sensitivitas bisa menurun (lebih banyak kasus positif yang terlewat), tetapi spesifisitas meningkat (lebih sedikit false positive). Untuk dataset yang tidak seimbang, keseimbangan antara sensitivitas dan spesifisitas yang ditampilkan dalam kurva ROC sangat berguna untuk menilai performa model secara menyeluruh, melampaui sekadar melihat akurasi (Bothma, 2023).

2.2.12. Python Programming

Python merupakan bahasa pemrograman tingkat tinggi yang dikenal luas karena kemampuannya yang serbaguna dan sintaksisnya yang sederhana serta mudah dipahami. Sebagai bahasa yang bersifat interpretatif, Python mendukung proses eksplorasi dan pengujian yang cepat. Didukung oleh komunitas pengembang yang besar dan aktif, Python dapat digunakan dalam berbagai bidang, seperti pengembangan web, analisis data, kecerdasan buatan, dan lainnya. Selain itu, sifatnya yang open source memungkinkan siapa pun untuk mengakses, memodifikasi, dan menyesuaikannya sesuai kebutuhan, menjadikannya pilihan ideal baik bagi pemula maupun profesional.

2.2.13. Scikit-Learn (SkLearn)

Scikit-learn (sklearn) merupakan salah satu pustaka Python paling populer yang digunakan dalam pengembangan model *machine learning*. Sebagai pustaka open-source, Scikit-learn menyediakan berbagai alat dan algoritma canggih untuk berbagai tugas seperti klasifikasi, regresi, klastering, pra-pemrosesan data, hingga evaluasi model (Pedregosa et al., 2011). Pustaka ini mencakup banyak algoritma

machine learning siap pakai, termasuk Support Vector Machines (SVM), Decision Tree, Random Forest, Naive Bayes, dan lainnya. Selain itu, Scikit-learn juga dilengkapi dengan fungsi-fungsi utilitas untuk pra-pemrosesan, seleksi fitur, dan validasi performa model, sehingga pengguna dapat membangun dan menguji model secara lebih cepat dan efisien. Karena memiliki dokumentasi lengkap, komunitas pengguna yang aktif, serta terintegrasi baik dengan pustaka Python lain seperti NumPy, Pandas, dan Matplotlib, Scikit-learn sangat disukai oleh para ilmuwan data dan praktisi *machine learning*.

2.2.14. Google Colaboratory

Google Colaboratory, atau yang sering disebut sebagai Google Colab, merupakan platform yang digunakan untuk mengembangkan proyek-proyek *machine learning* (Bisong, 2019). Platform ini menawarkan lingkungan *notebook* Jupyter berbasis cloud yang tidak memerlukan pengelolaan server, sehingga mendukung interaksi dan eksperimen secara langsung. Untuk mendukung kinerjanya, Colab dilengkapi dengan perangkat keras berkapasitas tinggi, seperti GPU (Graphics Processing Unit) dan TPU (Tensor Processing Unit). Pengguna dapat memilih berbagai jenis perangkat keras sesuai paket yang tersedia, seperti CPU, GPU T4, GPU A100, GPU V100, hingga TPU. Dalam penelitian ini, jenis perangkat keras yang digunakan adalah GPU T4.