

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu menjadi dasar penting dalam memahami perkembangan dan arah penelitian yang berkaitan dengan prediksi kelulusan mahasiswa berbasis *machine learning*. Melalui kajian terhadap penelitian-penelitian sebelumnya, dapat diketahui berbagai metode, algoritma, serta pendekatan yang telah diterapkan dalam membangun sistem prediksi akademik. Analisis terhadap penelitian terdahulu juga memberikan gambaran mengenai kekuatan, kelemahan, serta peluang pengembangan model yang lebih optimal dan interpretatif.

Sejumlah penelitian telah dilakukan dengan menerapkan berbagai algoritma *machine learning* seperti XGBoost, LightGBM, CatBoost, Random Forest, dan Multilayer Perceptron (MLP) untuk memprediksi kelulusan mahasiswa berdasarkan data akademik internal. Beberapa penelitian lainnya mengombinasikan algoritma *ensemble learning* dengan pendekatan *neural network* guna meningkatkan akurasi prediksi dan mengatasi keterbatasan model tunggal. Meskipun hasil yang dicapai menunjukkan peningkatan performa, sebagian besar model masih bersifat *black box*, sehingga sulit dijelaskan dan diinterpretasikan oleh pihak akademik.

Perkembangan penelitian terbaru mulai menekankan pentingnya transparansi dan interpretabilitas model melalui penerapan *Explainable Artificial Intelligence (XAI)*, salah satunya dengan metode SHapley Additive Explanations (SHAP). Metode ini digunakan untuk menjelaskan kontribusi setiap variabel terhadap hasil prediksi, sehingga hasil yang diperoleh tidak hanya akurat tetapi juga dapat dipahami secara logis. Berdasarkan kajian terhadap berbagai penelitian terdahulu, dapat disimpulkan bahwa penggunaan pendekatan *hybrid ensemble* dengan integrasi SHAP berpotensi memberikan hasil yang lebih akurat, stabil, dan mudah diinterpretasikan. Tinjauan terhadap beberapa penelitian terdahulu yang relevan

dengan topik ini disajikan pada Tabel 2.1, yang memuat ringkasan metode, algoritma, variabel, dan hasil utama dari masing-masing penelitian

Tabel 2. 1 Penelitian Terdahulu

No	Tahun	Judul	Masalah	Metode	Hasil	Main Findings
1	2023	<i>A Study on Dropout Prediction for University Students Using Machine Learning</i> [9]	Tingginya tingkat <i>dropout</i> mahasiswa yang berdampak pada universitas dan masyarakat.	Berbagai algoritma ML seperti Logistic Regression, Decision Tree, Random Forest, SVM, Deep Neural Network, dan LightGBM; menangani ketidakseimbangan data menggunakan SMOTE, ADASYN, dan Borderline-SMOTE.	Model LightGBM menghasilkan skor F1 tertinggi (0.840), mengungguli metode lain.	LightGBM paling efektif dalam memprediksi mahasiswa yang berisiko <i>dropout</i> , terutama dengan penanganan data tidak seimbang
2	2024	<i>Supervised machine learning algorithms for predicting student dropout and academic success: a comparative study</i> [10]	Membandingkan efektivitas algoritma ML dalam memprediksi <i>dropout</i> dan kesuksesan akademik mahasiswa dengan data tidak seimbang.	Algoritma: Decision Tree, SVM, Random Forest, Gradient Boosting, XGBoost, CatBoost, LightGBM; teknik SMOTE digunakan untuk <i>resampling</i> ; <i>hyperparameter tuning</i> dengan Optuna.	LightGBM dan CatBoost menunjukkan performa terbaik dibandingkan algoritma lain.	LightGBM paling efektif dalam memprediksi mahasiswa yang berisiko <i>dropout</i> , terutama dengan penanganan data tidak seimbang
3	2022	<i>Supervised Learning Applied to Graduation Forecast of Industrial</i>	Keterlambatan kelulusan dan kegagalan akademik di jurusan teknik industri.	Menggunakan ensemble learning berbasis XGBoost, Random	Model stacking ensemble dengan XGBoost sebagai base	Kombinasi SMOTE + XGBoost + ensemble learning terbukti

		<i>Engineering Students</i> [11]		Forest, dan SVM; data diseimbangkan dengan SMOTE; tuning hiperparameter dengan GridSearchCV untuk akurasi optimal.	learner mencapai akurasi 99.37% dan recall > 0.99, mengungguli model individual.	meningkatkan akurasi dan kemampuan deteksi dini mahasiswa berisiko gagal lulus tepat waktu
4	2023	<i>Multi-Output Based Hybrid Integrated Models for Student Performance Prediction</i> [12]	Kinerja mahasiswa sulit diprediksi dengan satu model karena banyak faktor multidimensional.	Menggunakan pendekatan hybrid ensemble yang menggabungkan XGBoost, Random Forest, dan Gradient Boosting; melakukan feature engineering untuk mengoptimalkan input variabel.	Model hybrid XGBoost-based ensemble mencapai akurasi 85% pada data universitas dan menunjukkan generalisasi yang baik.	XGBoost dalam model hibrida terbukti paling berkontribusi terhadap peningkatan akurasi dan kestabilan prediksi performa akademik
5	2022	<i>An Interpretable Prediction Method for University Student Academic Crisis Warning</i> [13]	Sulit mengidentifikasi dan menjelaskan penyebab penurunan prestasi akademik mahasiswa secara transparan.	Menggunakan CatBoost-SHAP (gabungan CatBoost dan SHapley Additive exPlanations) untuk memprediksi academic crisis dan menjelaskan faktor penyebabnya.	Model menghasilkan R^2 sebesar 80.3% dengan MAE 3.55, menunjukkan akurasi tinggi serta interpretasi yang jelas untuk setiap fitur penyumbang.	Kombinasi CatBoost + SHAP mampu memberikan prediksi akurat dan interpretasi visual bagi pihak universitas untuk melakukan intervensi akademik personal secara tepat waktu.
6	2022	<i>On Developing Generic Models for Predicting Student Outcomes in Educational</i>	Kurangnya model prediksi hasil belajar yang bisa digeneralisasi	Pengembangan model prediksi generik menggunakan berbagai	Model CatBoost memiliki kinerja terbaik dengan nilai	CatBoost unggul sebagai model generik lintas mata

		<i>Data Mining</i> [14]	untuk berbagai mata kuliah dan institusi.	algoritma ML (Random Forest, XGBoost, LightGBM, dan CatBoost) untuk mendeteksi mahasiswa berisiko rendah/tinggi.	F-measure dan AUC tertinggi, karena kemampuannya menangani data kategorikal dan missing values dengan efisien.	kuliah untuk prediksi hasil belajar, karena stabil di berbagai dataset pendidikan dan lebih efisien dalam penanganan fitur kategorikal.
7	2023	<i>Prediction of Students Performance Level Using Integrated Approach of ML Algorithms</i> [15]	Tingginya variasi hasil akademik antar mahasiswa.	Menggunakan pendekatan integratif berbagai algoritma ML: Naïve Bayes, Logistic Regression, SVM, Decision Tree, KNN, ANN, dan MLP (Multilayer Perceptron) untuk membandingkan performa prediksi.	Model Logistic Regression dan MLP menunjukkan hasil terbaik dengan akurasi mencapai 98.9% dan F1 tertinggi.	MLP berperan penting dalam menangkap hubungan non-linear antar variabel akademik, meningkatkan akurasi prediksi performa mahasiswa dibandingkan algoritma klasik
8	2021	<i>Predicting Academic Performance Using an Efficient Model Based on Fusion of Classifiers</i> [16]	Model tunggal memiliki keterbatasan dalam menangani data pendidikan kompleks dan dinamis.	Menggunakan fusion of classifiers berbasis ensemble learning dengan kombinasi Decision Tree, Random Forest, SVM, dan MLP (Multilayer Perceptron) untuk meningkatkan akurasi.	Model fusi mencapai akurasi 77–93%, tergantung dataset dan parameter pelatihan.	Integrasi MLP dalam sistem fusi meningkatkan kemampuan model menangkap pola non-linear dan memperbaiki stabilitas hasil prediksi akademik

9	2025	<i>Predicting learning achievement using ensemble learning with result explanation</i> [17]	Model prediksi hasil belajar sering bias dan tidak transparan.	Ensemble learning (6 base learners, Logistic Regression meta-learner, SHAP interpretasi).	Model ensemble mengungguli model ML dan DL tradisional.	SHAP meningkatkan interpretabilitas hasil dan mendukung intervensi persona
10	2024	<i>An empirical assessment of smote variants techniques and interpretation methods in improving the accuracy and the interpretability of student performance models</i> [18]	Ketidakseimbangan data dan rendahnya interpretabilitas model prediksi kinerja akademik mahasiswa.	Mengevaluasi berbagai varian SMOTE (Borderline-SMOTE, ADASYN, dan SMOTE-ENN) dikombinasikan dengan algoritma XGBoost, Random Forest, dan MLP. Untuk meningkatkan transparansi, digunakan metode SHAP (SHapley Additive Explanations) guna menjelaskan pengaruh tiap fitur terhadap hasil prediksi.	Model XGBoost dengan SMOTE-ENN dan analisis SHAP menghasilkan AUC tertinggi (0.88) dan F1-score yang stabil.	Kombinasi SMOTE varian + XGBoost + SHAP meningkatkan akurasi sekaligus memberikan interpretasi mendalam terhadap faktor-faktor akademik yang paling berpengaruh dalam kinerja mahasiswa

Berdasarkan hasil tinjauan terhadap sepuluh penelitian terdahulu yang tercantum pada Tabel 2.1, terlihat bahwa berbagai pendekatan *machine learning* telah banyak diterapkan untuk memprediksi kelulusan, performa akademik, maupun risiko dropout mahasiswa. Secara umum, penelitian-penelitian tersebut menunjukkan bahwa pemanfaatan algoritma *ensemble learning* seperti XGBoost, LightGBM, CatBoost, Random Forest serta model berbasis *neural network* seperti

Multilayer Perceptron (MLP), memberikan hasil prediksi yang akurat dan efisien dalam menganalisis data akademik mahasiswa.

Penelitian yang dilakukan pada tahun 2023 dan 2024 menekankan perbandingan berbagai algoritma *machine learning* dalam mendeteksi mahasiswa berisiko dropout [9] [10]. Beberapa di antaranya menunjukkan bahwa model *ensemble learning* berbasis *gradient boosting*, khususnya LightGBM dan CatBoost, mampu menghasilkan performa terbaik dengan nilai akurasi dan F1-score tertinggi setelah dilakukan penyeimbangan data menggunakan metode seperti SMOTE dan ADASYN. Hal ini menunjukkan bahwa model *gradient boosting* memiliki kemampuan lebih baik dalam mengelola data tidak seimbang serta mengenali pola akademik yang kompleks dibandingkan algoritma klasik seperti Decision Tree atau SVM.

Selain itu, penelitian lain menunjukkan bahwa penggunaan pendekatan *ensemble* dengan teknik *stacking* mampu meningkatkan kinerja model dibandingkan penggunaan model tunggal [11]. Salah satu studi melaporkan bahwa kombinasi model XGBoost dengan *stacking ensemble* mencapai akurasi hingga 99,37%, jauh lebih tinggi dibandingkan model individual. Temuan tersebut memperkuat bukti bahwa penggabungan beberapa model dalam arsitektur *ensemble* dapat meningkatkan generalisasi dan stabilitas sistem prediksi kelulusan mahasiswa. Penelitian berbasis *hybrid ensemble* juga membuktikan bahwa XGBoost sering kali menjadi komponen utama yang memberikan kontribusi terbesar terhadap peningkatan akurasi keseluruhan model [12].

Di sisi lain, aspek interpretabilitas menjadi perhatian penting dalam penelitian-penelitian terbaru. Sebagian besar studi terdahulu mengakui bahwa meskipun akurasi prediksi tinggi dapat dicapai, model yang digunakan sering kali berfungsi sebagai *black box* dan sulit dijelaskan. Untuk mengatasi hal tersebut, beberapa penelitian mengintegrasikan metode *Explainable Artificial Intelligence (XAI)*, khususnya SHapley Additive Explanations (SHAP), guna menjelaskan pengaruh setiap fitur terhadap hasil prediksi. Salah satu penelitian dengan pendekatan CatBoost-SHAP menunjukkan hasil yang sangat baik dengan nilai R^2 sebesar 80,3% dan kesalahan rata-rata absolut (MAE) sebesar 3,55, sekaligus memberikan

interpretasi yang jelas terhadap kontribusi setiap variabel akademik terhadap risiko penurunan performa mahasiswa [13].

Penelitian lain memperkuat efektivitas SHAP dengan menggabungkannya bersama algoritma XGBoost, dan Random Forest serta variasi teknik *resampling* seperti SMOTE-ENN untuk mengatasi ketidakseimbangan data [18]. Hasilnya menunjukkan peningkatan signifikan pada nilai AUC (0,88) dan F1-score yang stabil, sekaligus memberikan transparansi terhadap fitur-fitur akademik yang paling berpengaruh terhadap kinerja mahasiswa. Pendekatan interpretatif seperti ini menjadi arah baru dalam bidang *educational data mining*, karena memungkinkan pihak kampus tidak hanya mengetahui hasil prediksi, tetapi juga memahami alasan di balik keputusan model [14].

Beberapa penelitian lainnya mengembangkan model integratif yang melibatkan algoritma MLP dan *artificial neural network* untuk menangkap hubungan non-linear antar variabel akademik. Model ini terbukti unggul dalam memetakan kompleksitas data mahasiswa yang tidak dapat diselesaikan secara efektif oleh algoritma berbasis pohon keputusan. Penggunaan MLP menghasilkan peningkatan akurasi hingga 98,9%, menunjukkan bahwa integrasi antara model berbasis *ensemble learning* dan *neural network* mampu memberikan hasil prediksi yang lebih kuat dan adaptif [15] [16].

Berdasarkan kajian kuantitatif terhadap penelitian terdahulu yang dirangkum pada Tabel 2.1, pemilihan algoritma dalam penelitian ini didasarkan pada kinerja numerik yang konsisten unggul dari algoritma berbasis *tree ensemble* pada berbagai metrik evaluasi utama. Studi-studi terdahulu menunjukkan bahwa LightGBM dan CatBoost secara konsisten mencapai nilai F1-score di atas 0,80 dan AUC di atas 0,85 pada tugas prediksi dropout dan kelulusan mahasiswa dengan data tidak seimbang [9], [10], [14], mengungguli algoritma linier dan model tunggal lainnya. Secara khusus, LightGBM dilaporkan mencapai F1-score hingga 0,84 pada data dropout mahasiswa [9], sementara CatBoost menunjukkan F-measure dan AUC tertinggi pada skenario lintas mata kuliah dan institusi [14], yang menandakan stabilitas dan kemampuan generalisasi yang baik. Selain itu, integrasi Multilayer Perceptron (MLP) dalam beberapa penelitian meningkatkan performa secara signifikan, dengan akurasi dilaporkan mencapai 98 99% dan peningkatan F1-score

yang nyata pada data dengan hubungan non-linear kompleks [7], [8], [15]. Dari sisi interpretabilitas, penelitian yang mengombinasikan ensemble learning dengan SHAP menunjukkan bahwa model tidak hanya mempertahankan performa tinggi secara numerik misalnya AUC hingga 0,88 dan R^2 di atas 80% tetapi juga mampu menjelaskan kontribusi fitur secara kuantitatif dan visual [13], [17], [18]. Oleh karena itu, secara numerik dan metodologis, pendekatan Hybrid Ensemble yang mengombinasikan algoritma tree-based berperforma tinggi dengan model non-linear serta dilengkapi analisis SHAP dipilih karena memberikan keseimbangan optimal antara akurasi, stabilitas, dan interpretabilitas, sehingga paling sesuai untuk implementasi *early warning system* akademik berbasis data internal mahasiswa.

2.2 Teori yang berkaitan

2.2.1 Kinerja Akademik Mahasiswa

Kinerja akademik mahasiswa merupakan indikator yang digunakan untuk menilai tingkat keberhasilan mahasiswa dalam mencapai tujuan pembelajaran selama menempuh pendidikan di perguruan tinggi. Penilaian ini umumnya diukur melalui berbagai indikator, seperti Indeks Prestasi Semester (IPS), Indeks Prestasi Kumulatif (IPK), nilai mata kuliah, tingkat kelulusan tepat waktu, dan pencapaian kompetensi sesuai kurikulum [19]. Kinerja akademik mencerminkan kemampuan mahasiswa dalam menguasai materi pembelajaran, menerapkan pengetahuan, serta mengembangkan keterampilan sesuai bidang studinya. Pemantauan kinerja akademik yang efektif memungkinkan institusi pendidikan mengidentifikasi mahasiswa yang berprestasi maupun mereka yang berpotensi menghadapi kendala akademik. Kinerja akademik tidak hanya dipengaruhi oleh faktor internal seperti kemampuan kognitif, motivasi, dan manajemen waktu, tetapi juga oleh faktor eksternal seperti latar belakang sosial ekonomi, dukungan keluarga, dan lingkungan belajar [20].

2.2.2 Motivasi Belajar

Motivasi belajar merupakan salah satu faktor internal yang memiliki pengaruh signifikan terhadap keberhasilan akademik mahasiswa. Secara umum,

motivasi belajar dapat diartikan sebagai dorongan internal dan eksternal yang mengarahkan, mempertahankan, serta mengendalikan perilaku individu dalam mencapai tujuan pembelajaran. Mahasiswa dengan motivasi belajar yang tinggi cenderung menunjukkan semangat, ketekunan, dan konsistensi dalam menjalankan aktivitas akademik, sehingga lebih mampu mencapai prestasi yang optimal. Sebaliknya, rendahnya motivasi belajar sering kali berhubungan dengan penurunan kinerja akademik, ketidakteraturan dalam mengikuti perkuliahan, hingga risiko keterlambatan kelulusan atau bahkan dropout [21].

Dalam lingkungan pendidikan tinggi, motivasi belajar tidak hanya berkaitan dengan minat terhadap bidang studi, tetapi juga mencerminkan sejauh mana mahasiswa memiliki komitmen terhadap tujuan akademiknya. Faktor-faktor seperti kepuasan terhadap jurusan, dukungan dosen, sistem pembelajaran, dan persepsi terhadap prospek karier dapat memengaruhi tingkat motivasi belajar. Mahasiswa yang memiliki orientasi tujuan yang jelas, baik secara intrinsik (keinginan untuk menguasai pengetahuan) maupun ekstrinsik (dorongan untuk memperoleh hasil tertentu seperti nilai atau pekerjaan), cenderung memiliki kinerja akademik yang lebih stabil dan berpotensi lulus tepat waktu [22].

Motivasi belajar juga berperan sebagai indikator penting dalam sistem prediksi kelulusan mahasiswa. Berdasarkan analisis data akademik, mahasiswa dengan tingkat motivasi yang konsisten umumnya menunjukkan perkembangan IPK yang positif dari semester ke semester, serta memiliki kecenderungan lebih rendah terhadap penurunan performa akademik [23]. Dalam penelitian ini, variabel-variabel yang dapat merepresentasikan motivasi belajar antara lain frekuensi kehadiran, jumlah SKS yang ditempuh, pola pengambilan mata kuliah, serta konsistensi nilai akademik. Pola-pola tersebut dapat menjadi sinyal penting bagi model *machine learning* untuk mengidentifikasi mahasiswa dengan potensi risiko akademik tertentu.

2.2.3 Peringatan Sistem Dini (Early System Warning)

Peringatan sistem dini (*Early Warning System* atau EWS) merupakan suatu mekanisme yang dirancang untuk mendeteksi secara dini potensi terjadinya permasalahan akademik pada mahasiswa, seperti penurunan prestasi belajar, keterlambatan kelulusan, atau kemungkinan putus studi. Sistem ini berperan sebagai alat bantu dalam pemantauan kinerja mahasiswa agar pihak perguruan tinggi dapat mengambil tindakan preventif sebelum masalah tersebut berkembang menjadi lebih serius. Secara konseptual, EWS bertujuan untuk memberikan peringatan awal bagi dosen pembimbing, pihak program studi, maupun lembaga akademik ketika indikator-indikator tertentu menunjukkan adanya tanda-tanda penurunan performa mahasiswa.

Pada pendidikan tinggi, penerapan *Early Warning System* menjadi penting karena proses pembelajaran di perguruan tinggi bersifat dinamis dan kompleks. Mahasiswa tidak hanya dituntut untuk menguasai kompetensi akademik, tetapi juga harus mampu menyesuaikan diri dengan lingkungan sosial dan tekanan beban studi. Faktor-faktor seperti absensi perkuliahan, keterlambatan dalam mengumpulkan tugas, penurunan nilai, serta menurunnya partisipasi dalam kegiatan akademik sering kali menjadi sinyal awal dari penurunan kinerja [24]. Melalui sistem peringatan dini, lembaga pendidikan dapat mengidentifikasi pola tersebut lebih awal dan memberikan dukungan yang sesuai, seperti bimbingan akademik tambahan, konseling, atau pendampingan belajar.

Selain berfungsi sebagai mekanisme deteksi, EWS juga berperan dalam mendukung strategi peningkatan kualitas pendidikan dan keberhasilan studi mahasiswa. Sistem ini memberikan manfaat bagi institusi dalam melakukan evaluasi terhadap efektivitas proses pembelajaran, serta membantu dosen dalam memantau perkembangan mahasiswa secara individual [25]. Dengan adanya peringatan dini, tindakan korektif dapat dilakukan tepat waktu, sehingga mencegah potensi kegagalan studi atau keterlambatan kelulusan. Penerapan EWS juga berkontribusi pada peningkatan komunikasi antara mahasiswa dan pihak akademik, karena informasi mengenai risiko akademik dapat dijadikan dasar untuk membangun dialog konstruktif mengenai kendala dan solusi yang

dihadapi mahasiswa. Secara keseluruhan, *Early Warning System* menjadi salah satu strategi penting dalam mendukung keberhasilan pendidikan tinggi. Keberadaan sistem ini memungkinkan perguruan tinggi beralih dari pendekatan reaktif menuju pendekatan proaktif dalam mengelola kinerja mahasiswa. Melalui deteksi dini, intervensi yang cepat, dan pendampingan yang berkelanjutan, EWS berpotensi menurunkan tingkat kegagalan akademik, meningkatkan retensi mahasiswa, serta memastikan bahwa proses pembelajaran berjalan secara efektif dan berkelanjutan. Dengan demikian, sistem peringatan dini dapat menjadi instrumen strategis dalam menciptakan lingkungan akademik yang adaptif, suportif, dan berorientasi pada keberhasilan mahasiswa.

2.3 Teori tentang Framework/Algoritma yang digunakan

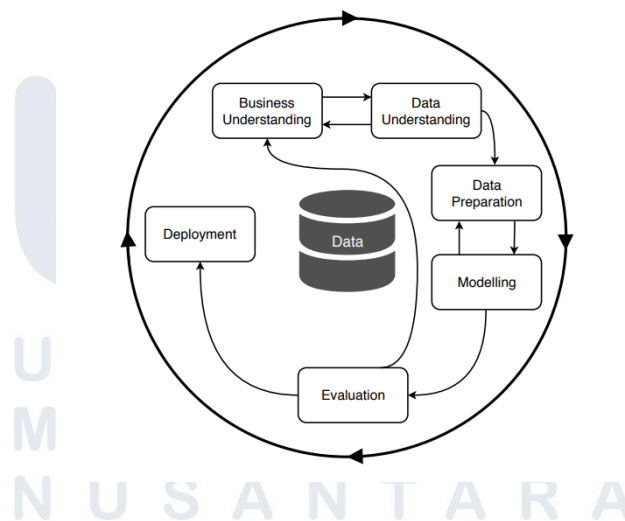
2.3.1 Data Mining

Data Mining adalah proses analisis data yang bertujuan untuk menemukan pola, tren, atau informasi berharga dari kumpulan data yang besar dan kompleks [26]. Proses ini melibatkan penggunaan teknik statistik, pembelajaran mesin, dan algoritma lainnya untuk menggali pengetahuan tersembunyi yang relevan dari data yang sering kali tidak terstruktur [27]. Data mining digunakan dalam berbagai domain, seperti keuangan, pemasaran, kesehatan, dan pendidikan, untuk membantu pengambilan keputusan berbasis data. Tahapan utama dalam data mining meliputi pemilihan data, praproses data untuk membersihkan dan menyiapkannya, transformasi data, penggalian pola atau model, serta evaluasi hasil [28]. Data mining sering mencakup penerapan algoritma seperti klasifikasi, klusterisasi, prediksi, dan asosiasi. Dengan kemampuan untuk mengungkap wawasan mendalam, data mining menjadi alat yang esensial bagi perusahaan atau organisasi yang ingin memanfaatkan data untuk mendukung strategi bisnis dan inovasi.

2.3.2 Cross Industry Standard Process for Data Mining

Cross-Industry Standard Process for Data Mining (CRISP-DM) merupakan salah satu kerangka kerja (framework) yang paling banyak digunakan dalam

proyek analisis data dan data mining karena mampu memberikan panduan proses yang sistematis, terstruktur, serta mudah disesuaikan dengan berbagai bidang penerapan. Framework ini dikembangkan agar bersifat generik namun fleksibel, sehingga dapat diterapkan di beragam domain, termasuk pendidikan, bisnis, maupun kesehatan. Pada penelitian berbasis analisis data, CRISP-DM berfungsi untuk memastikan bahwa setiap tahapan penggalian informasi dilakukan secara konsisten, logis, dan selaras dengan tujuan akhir penelitian [29]. CRISP-DM terdiri atas enam tahapan utama yang saling berkaitan dan bersifat iteratif, yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*. Keenam tahapan tersebut tidak berdiri secara terpisah, melainkan membentuk suatu siklus yang memungkinkan peneliti untuk melakukan penyesuaian dan perbaikan secara berulang hingga diperoleh hasil yang optimal. Setiap tahap memiliki peran penting dalam menjaga keakuratan, keandalan, dan relevansi hasil analisis terhadap kebutuhan penelitian. Tahapan-tahapan tersebut digambarkan dalam Gambar 2.1 Framework CRISP-DM.



Gambar 2. 1 Framework CRISP-DM [30]

Gambar 2.1 merupakan tahapan alur pada *framework* CRISP-DM, yang terdiri dari beberapa tahapan utama dalam proses *business understanding* dan analisis data. Berikut adalah penjelasan setiap tahapan pada CRISP-DM [30]:

1. *Business Understanding*

Tahap awal ini berfokus pada pendefinisian masalah utama yang ingin diselesaikan serta penentuan tujuan penelitian secara jelas. Dalam penelitian ini, tahap *business understanding* mencakup identifikasi permasalahan seputar prediksi kelulusan mahasiswa serta penentuan tujuan analisis, seperti mengukur tingkat ketepatan model dalam memprediksi kategori kelulusan (dropout, lulus tepat waktu, tidak lulus tepat waktu, dan lulus lebih awal). Pada tahap ini juga ditetapkan indikator keberhasilan, misalnya nilai akurasi dan F1-score sebagai ukuran performa model.

2. *Data Understanding*

Setelah tujuan ditentukan, langkah berikutnya adalah memahami karakteristik data yang akan digunakan. Pada tahap ini dilakukan eksplorasi awal terhadap data akademik mahasiswa untuk mengidentifikasi pola distribusi, mendeteksi nilai yang hilang (*missing values*), dan mengenali anomali. Tujuannya adalah memastikan bahwa data yang digunakan benar-benar mewakili kondisi mahasiswa dan layak untuk diolah pada tahap selanjutnya.

3. *Data Preparation*

Tahap ini merupakan proses penyiapan data agar siap digunakan dalam pemodelan. Aktivitas yang dilakukan meliputi pembersihan data (*data cleaning*), transformasi data untuk menyeragamkan format, serta pemilihan fitur (*feature selection*) yang paling relevan terhadap prediksi kelulusan. Langkah-langkah ini sangat penting untuk memastikan kualitas data yang baik, sehingga model yang dibangun dapat menghasilkan hasil yang akurat dan tidak bias.

4. *Modeling*

Tahap *modeling* berfokus pada pembangunan model prediksi menggunakan algoritma yang telah dipilih. Pada tahap ini juga dilakukan penyetelan parameter (*hyperparameter tuning*) guna

memperoleh konfigurasi model terbaik yang memberikan performa optimal.

5. *Evaluation* (Evaluasi)

Setelah model dibangun, tahap berikutnya adalah mengevaluasi kinerjanya. Evaluasi dilakukan untuk menilai sejauh mana model memenuhi tujuan yang telah ditetapkan pada tahap *business understanding*. Pengujian dilakukan menggunakan metrik seperti akurasi dan F1-score untuk mengukur keseimbangan antara presisi dan sensitivitas model.

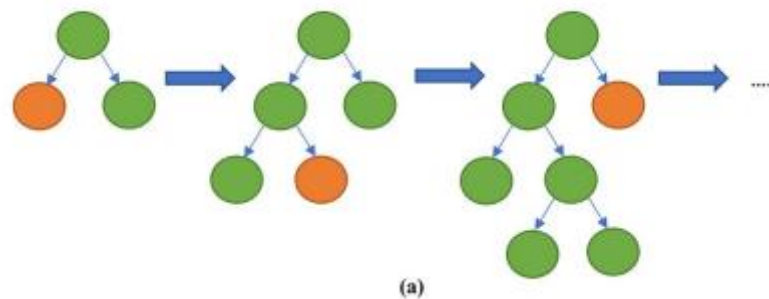
6. *Deployment*

Tahap terakhir adalah penerapan model ke dalam sistem operasional atau media visualisasi hasil analisis. Pada penelitian ini, hasil model dapat diimplementasikan dalam bentuk dashboard analitik atau sistem pendukung keputusan akademik yang membantu pihak universitas dalam memantau dan mengidentifikasi mahasiswa berisiko tinggi. Dengan demikian, hasil analisis tidak hanya berhenti pada tahap eksperimental, tetapi juga dapat memberikan dampak praktis bagi institusi pendidikan.

2.3.3 LIGHTGBM

Light Gradient Boosting Machine (LightGBM) merupakan algoritma *machine learning* berbasis *ensemble* yang dikembangkan oleh Microsoft, dan termasuk dalam keluarga metode *gradient boosting decision tree (GBDT)*. LightGBM dirancang untuk melakukan proses pelatihan yang cepat dan efisien tanpa mengorbankan akurasi, terutama ketika berhadapan dengan dataset berukuran besar dan berdimensi tinggi. Algoritma ini menggunakan pendekatan *leaf-wise growth strategy* dalam membangun pohon keputusan, berbeda dengan metode *level-wise* yang digunakan pada algoritma lain seperti XGBoost [31]. Strategi ini memungkinkan LightGBM memilih cabang dengan nilai *loss reduction* terbesar, sehingga mempercepat konvergensi dan meningkatkan performa model. Selain itu, LightGBM mendukung teknik optimisasi seperti

Gradient-based One Side Sampling (GOSS) dan *Exclusive Feature Bundling (EFB)* untuk mengurangi kompleksitas perhitungan dan penggunaan memori [32]. Keunggulan utama LightGBM terletak pada kemampuannya menangani data numerik maupun kategorikal dengan skala besar, efisiensi waktu pelatihan, serta kestabilan model yang tinggi terhadap variasi data. Dalam penelitian ini, LightGBM digunakan sebagai salah satu algoritma utama dalam pendekatan *hybrid ensemble* karena kemampuannya menghasilkan prediksi yang akurat, cepat, dan efisien dalam memproses data akademik mahasiswa.

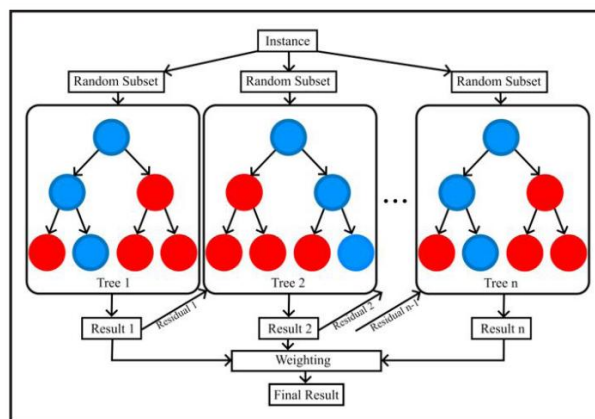


Gambar 2. 2 Arsitektur LIGHTGBM [33]

Arsitektur yang diilustrasikan pada gambar 2.2 merupakan pertumbuhan *leaf-wise* (pertumbuhan berbasis daun), yang merupakan strategi inti dari LightGBM. Berbeda dengan algoritma *boosting* lain yang membangun pohon secara *level-wise* (simetris lapis demi lapis), LightGBM bekerja dengan cara yang lebih efisien. Pada setiap langkah, LightGBM akan memindai *semua* daun yang ada di pohon dan secara selektif memilih satu daun yang akan memberikan pengurangan kerugian (*loss reduction*) terbesar jika dibelah. Seperti yang ditunjukkan pada gambar, hal ini menghasilkan pohon yang tumbuh secara tidak seimbang (*asymmetric*), di mana algoritma segera memfokuskan "upaya" pada bagian-bagian data yang paling sulit atau paling informatif [33]. Pendekatan ini tidak hanya membuat proses pelatihan menjadi jauh lebih cepat tetapi juga seringkali mengarah pada akurasi yang lebih tinggi karena kompleksitas model dialokasikan ke tempat yang paling membutuhkannya.

2.3.4 XGBoost

XGBoost (*Extreme Gradient Boosting*) adalah salah satu algoritma *ensemble learning* berbasis *gradient boosting* yang dirancang untuk memberikan kinerja prediksi yang tinggi dengan efisiensi komputasi yang optimal. Algoritma ini bekerja dengan membangun model prediktif secara bertahap, di mana setiap pohon keputusan (*decision tree*) baru dilatih untuk memperbaiki kesalahan prediksi yang dihasilkan oleh pohon-pohon sebelumnya. Proses ini dilakukan dengan meminimalkan fungsi kehilangan (*loss function*) menggunakan metode *gradient descent*, sehingga model secara iteratif menjadi semakin akurat. Keunggulan XGBoost dibandingkan metode *gradient boosting* konvensional terletak pada kemampuannya mengatasi *overfitting* melalui regularisasi (*L1* dan *L2 regularization*), optimasi kecepatan dengan teknik *parallel processing*, serta efisiensi penggunaan memori [34].



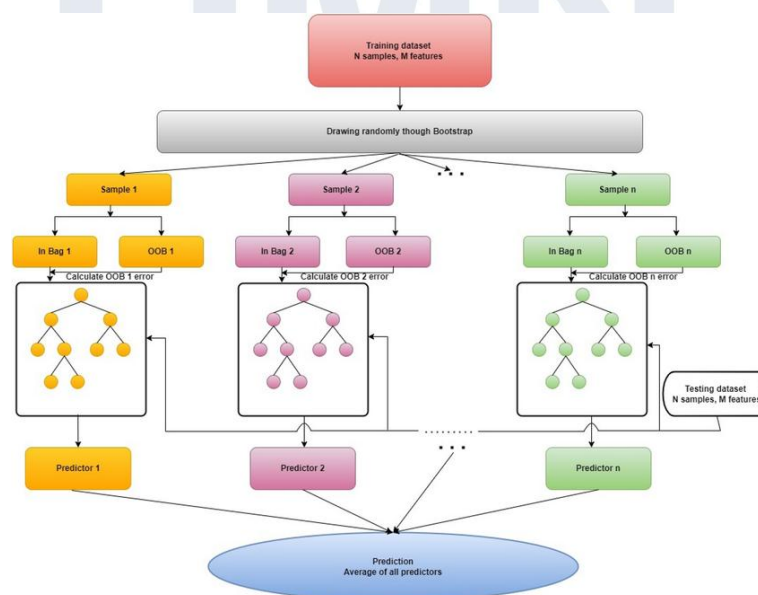
Gambar 2. 3 Arsitektur XGBoost [35]

Berdasarkan gambar arsitektur XGBoost terdiri dari beberapa komponen penting. Pertama, algoritma memulai proses pembelajaran dengan membangun model awal menggunakan data latih dan menghitung nilai kesalahan prediksi. Kedua, pohon keputusan berikutnya dibentuk dengan fokus pada perbaikan kesalahan tersebut, di mana bobot gradien dari data latih digunakan untuk menentukan arah perbaikan. Ketiga, XGBoost menerapkan teknik *shrinkage* atau *learning rate* untuk mengontrol kontribusi setiap pohon baru, sehingga pembelajaran menjadi lebih stabil dan terhindar dari *overfitting*. Selain itu, algoritma ini menggunakan *column subsampling* dan *tree pruning* berbasis

maximum depth untuk mengoptimalkan kinerja model. Hasil akhir prediksi diperoleh melalui penjumlahan kontribusi dari seluruh pohon yang telah dibangun [36].

2.3.5 Catboost

CatBoost (*Categorical Boosting*) adalah algoritma *gradient boosting* berbasis pohon keputusan yang dirancang khusus untuk menangani variabel kategori secara efisien tanpa memerlukan proses pra-pemrosesan yang kompleks seperti *one-hot encoding*. CatBoost bekerja dengan membangun serangkaian pohon keputusan secara bertahap, di mana setiap pohon baru dilatih untuk memperbaiki kesalahan prediksi dari model sebelumnya dengan meminimalkan fungsi kehilangan (*loss function*) menggunakan metode *gradient descent*. Salah satu keunggulan utama CatBoost adalah penggunaan teknik *ordered boosting*, yang mengatasi masalah *prediction shift* atau kebocoran target (*target leakage*) yang sering muncul pada metode *boosting* konvensional [37]. Selain itu, CatBoost menerapkan *efficient oblivious decision trees*, yaitu pohon keputusan simetris di mana kondisi pemisahan (*split conditions*) diterapkan secara seragam pada semua node di tingkat yang sama, sehingga meningkatkan kecepatan pelatihan dan prediksi [38].

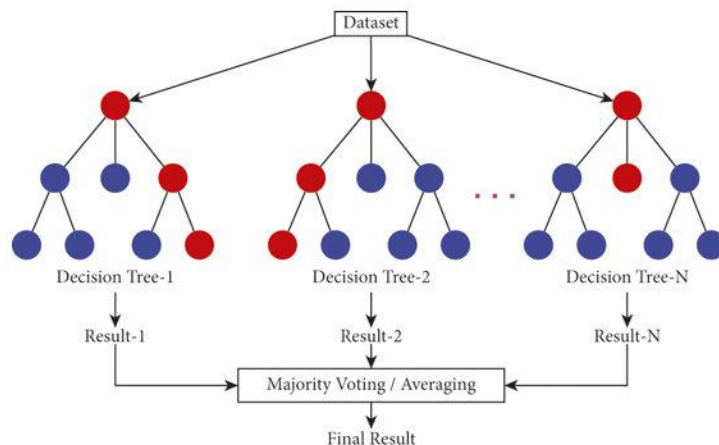


Gambar 2. 4 Arsitektur CatBoost [39]

Arsitektur CatBoost dimulai dengan konversi variabel kategori menjadi representasi numerik menggunakan teknik *target statistics* dengan skema *ordered target encoding* untuk menghindari kebocoran informasi [37]. Selanjutnya, pohon keputusan dibentuk secara berurutan, di mana setiap pohon memperbaiki residu atau kesalahan dari pohon sebelumnya. *Ordered boosting* memastikan bahwa data yang digunakan untuk membangun model pada suatu iterasi tidak mengandung informasi dari data masa depan dalam urutan pelatihan, sehingga meminimalkan bias. Pohon yang digunakan bersifat *oblivious*, sehingga semua jalur pada tingkat yang sama menggunakan aturan pemisahan yang identik, membuat model lebih sederhana dan efisien secara komputasi [38].

2.3.6 Random Forest

Random Forest merupakan algoritma *ensemble learning* berbasis *decision tree* yang dikembangkan oleh Leo Breiman pada tahun 2001. Algoritma ini bekerja dengan membangun sejumlah pohon keputusan secara acak pada data pelatihan, kemudian menggabungkan hasil prediksi dari setiap pohon untuk menghasilkan keputusan akhir yang lebih stabil dan akurat [40]. Setiap pohon dibangun dengan memilih subset acak dari data (*bootstrap sampling*) serta subset acak dari fitur (*feature bagging*), sehingga mengurangi risiko *overfitting* yang umum terjadi pada model *decision tree* tunggal [41]. Dalam proses klasifikasi, Random Forest menentukan hasil prediksi berdasarkan prinsip *majority voting*, sedangkan pada regresi digunakan nilai rata-rata dari seluruh prediksi pohon. Keunggulan utama algoritma ini terletak pada kemampuannya menangani data dengan jumlah fitur besar, toleransi tinggi terhadap *outlier*, serta kestabilan hasil meskipun terdapat variasi pada data pelatihan. Selain itu, Random Forest juga menyediakan ukuran penting fitur (*feature importance*), yang membantu dalam memahami kontribusi relatif dari setiap variabel terhadap hasil prediksi [42].



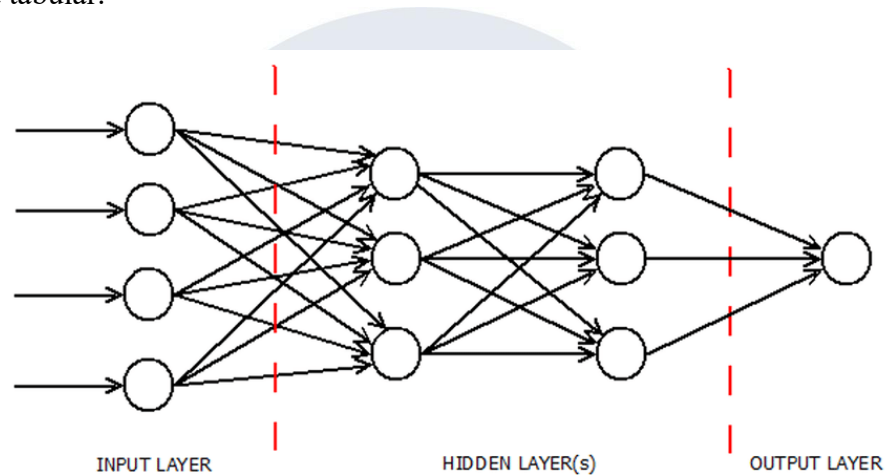
Gambar 2. 5 Arsitektur Random Forest [43]

Berdasarkan arsitektur pada gambar 2.5 , proses diawali dari satu Dataset utama yang menjadi masukan bagi sekumpulan model Pohon Keputusan (Decision Tree). Dataset ini diproses secara paralel oleh beberapa pohon, yang dilabeli sebagai Decision Tree-1, Decision Tree-2, hingga Decision Tree-N. Setiap pohon individu tersebut kemudian menghasilkan prediksinya sendiri, yang disebut Result-1, Result-2, dan seterusnya. Semua hasil individual ini lantas dikumpulkan dan digabungkan dalam satu langkah Majority Voting / Averaging (pemungutan suara mayoritas atau rata-rata) untuk menghasilkan satu Final Result atau keluaran akhir yang tunggal [44].

2.3.7 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) merupakan salah satu jenis jaringan saraf tiruan (artificial neural network) yang termasuk dalam kategori feedforward neural network, di mana aliran data bergerak searah dari lapisan input menuju lapisan output tanpa adanya umpan balik. Arsitektur MLP terdiri atas tiga komponen utama, yaitu input layer, satu atau lebih hidden layer, dan output layer, di mana setiap lapisan terdiri atas sejumlah neuron yang saling terhubung melalui bobot (*weights*) [45]v. Proses pembelajaran pada MLP dilakukan menggunakan algoritma backpropagation, yang secara iteratif memperbarui bobot berdasarkan nilai kesalahan antara output prediksi dan target sebenarnya.

Fungsi aktivasi non-linear seperti ReLU (Rectified Linear Unit) atau sigmoid digunakan untuk memungkinkan jaringan mempelajari hubungan kompleks antar variabel input [46]. Keunggulan utama MLP terletak pada kemampuannya mengenali pola non-linear yang tidak dapat diatasi oleh model linier tradisional, sehingga sangat efektif untuk tugas klasifikasi, regresi, dan prediksi berbasis data tabular.



Gambar 2. 6 Arsitektur Multilayer Perceptron [47]

Gambar tersebut mengilustrasikan arsitektur Multilayer Perceptron (MLP), yang merupakan salah satu bentuk Jaringan Saraf Tiruan (*Artificial Neural Network*) kategori *feedforward*. Arsitektur ini tersusun atas tiga komponen utama yang terlihat jelas: Input Layer (Lapisan Masukan), Hidden Layer(s) (Lapisan Tersembunyi), dan Output Layer (Lapisan Keluaran). Dalam contoh ini, *Input Layer* memiliki empat neuron yang berfungsi menerima fitur-fitur data. Data ini kemudian diteruskan ke *Hidden Layer*, yang dalam diagram ini terdiri dari dua lapisan (masing-masing dengan tiga neuron), tempat terjadinya sebagian besar komputasi untuk mempelajari pola-pola kompleks. Terakhir, *Output Layer* (dengan satu neuron) menghasilkan nilai prediksi akhir. Ciri khas arsitektur ini adalah alirannya yang *feedforward* (maju) di mana informasi bergerak satu arah dari input ke output dan konektivitas padat (*fully connected*), di mana setiap neuron di satu lapisan terhubung ke setiap neuron di lapisan berikutnya [48].

2.3.8 Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence (XAI) adalah cabang dari kecerdasan buatan (artificial intelligence) yang berfokus pada pengembangan metode dan teknik untuk membuat proses pengambilan keputusan model machine learning dapat dipahami oleh manusia [49]. XAI bertujuan memberikan transparansi dengan menunjukkan alasan di balik suatu prediksi, termasuk kontribusi masing-masing variabel terhadap hasil tersebut [50]. Selain itu, XAI tidak hanya berperan dalam meningkatkan kepercayaan pengguna terhadap model, tetapi juga penting untuk memastikan bahwa keputusan yang dihasilkan sesuai dengan prinsip etika, keadilan, dan akuntabilitas [51].

Secara umum, XAI dapat dibedakan menjadi dua pendekatan, yaitu *intrinsic interpretability* dan *post-hoc explanation*. *Intrinsic interpretability* mengacu pada penggunaan model yang secara alami mudah dipahami, seperti decision tree atau linear regression, sedangkan *post-hoc explanation* digunakan untuk menjelaskan model kompleks seperti deep learning atau ensemble methods melalui teknik interpretasi tambahan, misalnya metode SHAP (SHapley Additive exPlanations) [52].

2.3.9 SHapley Additive exPlanations (SHAP)

SHapley Additive exPlanations (SHAP) adalah suatu metode interpretabilitas model dalam *machine learning* yang dikembangkan berdasarkan konsep nilai Shapley (*Shapley value*) dari teori permainan kooperatif (*cooperative game theory*). Nilai Shapley awalnya diperkenalkan oleh Lloyd Shapley pada tahun 1953 untuk menentukan kontribusi setiap pemain terhadap hasil keseluruhan dalam sebuah permainan kooperatif. Dalam machine learning, “pemain” diartikan sebagai fitur atau variabel input, sedangkan “hasil permainan” diartikan sebagai prediksi model. Metode SHAP menghitung kontribusi marjinal setiap fitur terhadap hasil prediksi dengan mempertimbangkan semua kemungkinan kombinasi fitur, sehingga memberikan penjelasan yang adil dan konsisten mengenai pengaruh setiap variabel terhadap keluaran model [53].

Keunggulan utama SHAP adalah sifatnya yang bersifat model-agnostic dan model-specific, yang berarti dapat diterapkan pada hampir semua jenis algoritma machine learning, baik yang bersifat kotak hitam (black box) seperti neural network maupun yang lebih transparan seperti decision tree [54]. Nilai SHAP positif menunjukkan bahwa suatu fitur mendorong hasil prediksi ke arah kelas atau nilai yang lebih tinggi, sedangkan nilai SHAP negatif menunjukkan kontribusi ke arah sebaliknya [55]. Melalui visualisasi seperti summary plot, dependence plot, dan force plot, SHAP memudahkan interpretasi pengaruh fitur baik secara global (keseluruhan model) maupun lokal (prediksi individual). Dengan demikian, SHAP menjadi alat yang sangat penting dalam prediksi, sehingga keputusan yang diambil berdasarkan model dapat dipertanggungjawabkan secara transparan dan etis.

2.3.10 Evaluation Metrics

Evaluation Metrics adalah serangkaian ukuran atau indikator yang digunakan untuk mengevaluasi kinerja model analisis data atau algoritma machine learning. Tujuan utama dari evaluation metrics adalah untuk memberikan gambaran tentang seberapa baik model mampu menangkap pola, membuat prediksi, atau memenuhi tujuan analisis. Dalam *data mining* atau *machine learning*, *evaluation metrics* membantu menentukan keakuratan, relevansi, dan kegunaan model dalam menangani dataset tertentu. Evaluation metrics memiliki peran penting dalam mencapai hasil yang optimal terutama training model klasifikasi. Selain itu, pemilihan metrik evaluasi yang tepat harus disesuaikan dengan jenis data, tujuan analisis, dan jenis masalah, seperti klasifikasi. Berikut adalah penjelasan evaluation metrics yang digunakan:

1) Accuracy

Accuracy adalah alat ukur proporsi prediksi yang benar terhadap total jumlah data. *Accuracy* bekerja dengan baik jika data seimbang, tetapi kurang efektif pada dataset yang memiliki ketidakseimbangan kelas (*imbalance*). Rumus *accuracy* dapat dilihat pada rumus (2.1) [56].

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \quad (2.1)$$

Rumus 2. 1 Rumus Accuracy

Berdasarkan rumus (2.1), *TP* (True Positive) mengacu pada jumlah sampel positif yang berhasil diprediksi dengan benar oleh model, sedangkan *TN* (True Negative) adalah jumlah sampel negatif yang juga diprediksi dengan benar. Sebaliknya, *FP* (False Positive) menunjukkan jumlah sampel negatif yang keliru diklasifikasikan sebagai positif, dan *FN* (False Negative) mencerminkan jumlah sampel positif yang salah diprediksi sebagai negatif oleh model.

2) Precision

Precision digunakan untuk mengukur akurasi prediksi model terhadap kelas positif, dengan menghitung proporsi data positif yang benar dari semua data yang diprediksi sebagai positif. *Precision* penting untuk digunakan dalam kasus di mana kesalahan *false positive* harus diminimalkan. Rumus *Precision* dapat dilihat pada rumus (2.2) [57]

$$Precision = \frac{tp}{tp + fp} \quad (2.2)$$

Rumus 2. 2 Rumus Precision

Berdasarkan rumus (2.2), *TP* (*True Positive*) mengacu pada jumlah prediksi positif yang benar, dan *FP* (*False Positive*) menunjukkan jumlah prediksi positif yang salah. Nilai *precision* menunjukkan proporsi prediksi positif yang benar-benar relevan, sehingga semakin tinggi *precision*, semakin baik model dalam menghindari prediksi positif yang salah.

3) Recall (Sensitivity)

Recall mengukur kemampuan model untuk mendeteksi semua data positif yang sebenarnya. *Recall* penting dalam situasi di mana

kesalahan *false negative* memiliki konsekuensi yang besar. Berikut merupakan rumus dari recall pada (2.3) [27].

$$Recall = \frac{tp}{tp + fn} \quad (2.3)$$

Rumus 2. 3 Rumus Recall

Berdasarkan rumus (2.3), *TP (True Positive)* mengacu pada jumlah prediksi positif yang benar, dan *FN (False Negative)* menunjukkan jumlah sampel positif yang salah diprediksi sebagai negatif.

4) F1-Score

F1-Score adalah rata-rata harmonis dari *precision* dan recall. Metrik ini berguna untuk distribusi data yang tidak seimbang. F1-Score memberikan keseimbangan antara *precision* dan *recall*, sehingga menjadi metrik yang ideal untuk evaluasi model secara keseluruhan. Berikut merupakan rumus dari F1-Score (2.4) [58].

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.4)$$

Rumus 2. 4 Rumus F1-Score

Berdasarkan rumus (2.4), nilai ini merupakan rata-rata harmonis dari hasil *Precision* dan *Recall*, sehingga memberikan bobot yang seimbang.

5) Confusion Matrix

Confusion Matrix merupakan metode evaluasi yang digunakan untuk menilai kinerja model klasifikasi dengan membandingkan hasil prediksi model terhadap data aktual. Matriks ini disajikan dalam bentuk tabel yang memperlihatkan distribusi antara prediksi benar dan salah dari setiap kelas, sehingga memudahkan analisis terhadap pola kesalahan klasifikasi yang dilakukan oleh model. Melalui Confusion Matrix, peneliti dapat melihat sejauh mana model mampu mengenali kategori tertentu secara tepat, serta mengidentifikasi area di mana

model masih sering melakukan kesalahan. Informasi dari Confusion Matrix menjadi dasar untuk menghitung berbagai metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*, yang secara keseluruhan memberikan gambaran menyeluruh mengenai performa model klasifikasi yang digunakan

$$\text{Confusion Matrix} = \begin{matrix} & TP & FP \\ FN & & \\ TN & & \end{matrix} \quad (2.5)$$

Rumus 2. 5 Rumus Confusion Matrix

Rumus 2.5 menggambarkan hubungan antara hasil prediksi model dan kondisi aktual dalam bentuk matriks dua dimensi. Baris pada matriks menunjukkan kelas aktual, sedangkan kolom merepresentasikan kelas hasil prediksi. Nilai pada diagonal utama menunjukkan prediksi yang benar, sedangkan nilai di luar diagonal menunjukkan kesalahan klasifikasi. Matriks ini menjadi dasar untuk menghitung metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*.

- 6) ROC-AUC (*Receiver Operating Characteristic - Area Under Curve*)
ROC-AUC mengukur kemampuan model dalam membedakan antara kelas positif dan negatif di berbagai threshold [28]. Pada *ROC Curve* menggambarkan hubungan antara *true positive rate* (TPR) dan *false positive rate* (FPR). Sedangkan, AUC (*Area Under Curve*) memberikan nilai antara 0 dan 1. Nilai yang mendekati 1 menunjukkan model yang sangat baik dalam membedakan kelas

2.4 Tools/software yang digunakan

2.4.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang serbaguna dan populer, dirancang untuk mendukung berbagai aplikasi, termasuk analisis data, machine learning, dan banyak bidang lainnya [59]. Dengan sintaksis sederhana,

fleksibilitas tinggi, serta ekosistem pustaka yang luas seperti NumPy, pandas, Matplotlib, dan scikit-learn, Python menjadi alat utama dalam proyek-proyek analisis data, seperti prediksi customer churn. Python memungkinkan pengguna untuk melakukan preprocessing data, membangun model prediksi, dan mengevaluasi performanya. Keunggulannya terletak pada sifat open-source, kemudahan integrasi dengan berbagai teknologi, kemampuan mengolah data dalam skala besar, serta dengan komunitas yang aktif dan dokumentasi yang luas, Python terus menjadi pilihan utama untuk pengolahan data modern [60].

2.4.2 Google Colab

Google Colab (Google Collaboratory) adalah layanan berbasis cloud yang memungkinkan pengguna menjalankan kode python langsung dari browser tanpa perlu instalasi perangkat lunak tambahan. Platform ini dirancang untuk mendukung pengembangan machine learning, analisis data, dan pemrograman berbasis Python, dengan menyediakan integrasi langsung dengan Google Drive untuk menyimpan dan berbagi file [61]. Google Colab mendukung penggunaan GPU dan TPU gratis untuk mempercepat komputasi, menjadikannya alat yang efisien bagi peneliti, mahasiswa, dan praktisi data. Selain itu, Colab memiliki antarmuka mirip dengan Jupyter Notebook, sehingga mudah digunakan oleh pemula maupun pengguna berpengalaman. Dengan berbagai pustaka bawaan seperti TensorFlow, NumPy, dan Pandas, Google Colab menjadi pilihan populer untuk proyek berbasis data science dan machine learning.