BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu memiliki peran krusial dalam menyediakan kerangka teoretis, memberikan justifikasi ilmiah, dan memperdalam pemahaman terkait topik yang diteliti. Kajian terhadap studi-studi sebelumnya berfungsi sebagai peta jalan intelektual yang membantu peneliti mengidentifikasi *state-of-the-art* (kondisi terkini) dalam bidang prediksi akademik dan penggunaan faktor eksternal. Dengan meninjau hasil-hasil yang telah dicapai, peneliti dapat menentukan posisi unik dari penelitian yang diusulkan ini, terutama dalam hal kesenjangan (*research gap*) atau keterbatasan yang belum diatasi oleh studi sebelumnya.

Selain itu, penelitian terdahulu juga membantu memvalidasi variabel yang digunakan dan memperkuat landasan metodologi yang akan diterapkan. Temuan dan hasil dari penelitian terdahulu turut menjadi bahan pertimbangan utama untuk menilai kontribusi signifikan apa yang diberikan oleh penelitian ini, khususnya dalam konteks perbandingan model (*model comparison*) dan implementasi *Explainable AI* (XAI) menggunakan SHAP. Dengan demikian, sub-bab ini tidak hanya merangkum daftar referensi, tetapi juga secara eksplisit menunjukkan keterkaitan antara studi baru dengan penelitian sebelumnya, menjadikannya dasar argumentasi kuat sebelum masuk pada sintesis dan perumusan hipotesis. Tabel 2.1 merangkum daftar penelitian terdahulu yang secara langsung relevan dan mendukung penelitian ini.

Tabel 21 Penelitian Terdahulu

Judul Penelitian	Nama Peneliti	Permasalahan	Solusi yang Ditawarkan	Data yang Digunakan	Model yang Digunakan	Hasil	Kekurangan
On Developing Generic Models for Predicting Student Outcomes in Educational Data Mining [11]	Gomathy Ramaswami, Teo Susnjak, Anuradha Mathrani	Model prediksi yang bersifat spesifik per mata kuliah tidak skalabel, membutuhkan biaya tinggi, dan rentan overfitting ketika jumlah data per mata kuliah terbatas	Mengembangkan model prediksi generik (course-agnostic) yang dapat diterapkan lintas mata kuliah tanpa perlu pelatihan ulang secara spesifik	Data LMS (*Learning Management System*), SMS (*School Management System*), dan EMS (*Enterprise Messaging System*) dari berbagai mata kuliah di institusi pendidikan tinggi Australia	CatBoost, Random Forest, Naive Bayes, Logistic Regression, KNN. Interpretasi menggunakan SHAP	CatBoost memberikan performa terbaik dengan akurasi sekitar 75% dan nilai AUC 0,87	Tingkat akurasi masih relatif moderat sehingga masih memiliki ruang untuk pengembangan dan peningkatan performa
Predicting Student Academic Success Using XGBoost and Optuna Tuning Based on Student Attendance [8]	Marcello Roy, Iwan Prasetiawan, Ririn Ikana Desanti, Suryasari	Diperlukan model prediksi IPK dengan performa tinggi, sementara pengaturan hyperparameter pada model boosting bersifat	Mengoptimasi hyperparameter XGBoost menggunakan framework Optuna berbasis Bayesian Optimization	961 data mahasiswa sarjana Universitas Multimedia Nusantara	XGBoost (regresi)	Model XGBoost teroptimasi mencapai R^2 sebesar 0,8456 dan mengungguli model baseline	Fokus pada regresi IPK, bukan klasifikasi status kelulusan. Fitur yang digunakan bersifat agregat dan belum merepresentasikan

Judul Penelitian	Nama Peneliti	Permasalahan	Solusi yang Ditawarkan	Data yang Digunakan	Model yang Digunakan	Hasil	Kekurangan
		kompleks dan memakan waktu					dinamika akademik
Multiclass Prediction Model for Student Grade Prediction Using Machine Learning [12]	Siti Dianah Abdul Bujang et al.	Ketidakseimbangan data multiclass menyebabkan overfitting dan misklasifikasi pada kelas minoritas	Mengusulkan pendekatan SFS (SMOTE + Feature Selection) untuk menyeimbangkan data dan memilih fitur relevan	1.282 data nilai akhir mahasiswa dengan 5 kelas	Random Forest, J48, SVM, Naive Bayes, KNN, Logistic Regression	Kombinasi RF + SMOTE + Feature Selection mencapai F-measure tertinggi sebesar 99,5%	Nilai evaluasi sangat tinggi dan berpotensi overfitting. Tidak menggunakan model boosting modern seperti CatBoost
Prediction of Students' Academic Performance Using Long Short-Term Memory (LSTM) [10]	Luis Vives et al.	Tingginya tingkat drop out pada mata kuliah pemrograman dengan kondisi data tidak seimbang	Menggunakan model deep learning LSTM dengan perbandingan teknik oversampling SMOTE dan GAN	661 data mahasiswa dari dua universitas di Peru (Lulus/Gagal)	LSTM, DNN, Decision Tree, Random Forest, Logistic Regression, SVC, KNN	LSTM dengan oversampling GAN mencapai akurasi 98,3%	Model ML tradisional berkinerja sangat rendah saat dilatih dengan data GAN. Proses pelatihan GAN memerlukan waktu komputasi yang tinggi
Machine Learning Analysis of Factors Affecting	Jingzhao Lu et al.	Perlunya eksplorasi faktor non-akademik	Menggunakan Chi-Square Test untuk	1.101 data kuesioner mahasiswa	Logistic Regression, SVC,	XGBoost menunjukkan performa terbaik	Dataset terbatas pada satu universitas dan

Judul Penelitian	Nama Peneliti	Permasalahan	Solusi yang Ditawarkan	Data yang Digunakan	Model yang Digunakan	Hasil	Kekurangan
College Students' Academic Performance [13]		sebagai prediktor kinerja akademik	seleksi fitur dari data kuesioner		Random Forest, XGBoost	dengan akurasi 86,52%	berbasis kuesioner subjektif
Analysis and Prediction of Influencing Factors of College Student Achievement Based on Machine Learning [14]	Dongxuan Wang et al.	Hasil penelitian universitas elit tidak selalu relevan untuk universitas biasa	Seleksi fitur menggunakan Chi-Square Test dan perbandingan beberapa model ML	292 mahasiswa baru Computer Science	Logistic Regression, SVC, Random Forest, Naive Bayes	SVC menunjukkan performa paling stabil dengan akurasi 86,18%	Dataset sangat kecil dan hasil sulit digeneralisasi
Prediksi Kelulusan Mahasiswa Menggunakan Data Mining Algoritma K-means [5]	Ray Mondow Sagala	Status kelulusan mata kuliah prasyarat baru diketahui di akhir semester	Mengelompokkan mahasiswa menggunakan clustering K-Means	118 data mahasiswa	K-Means dan Chi-Square Attribute Selection	Akurasi 93%, presisi 96%, recall 92%	Dataset sangat kecil dan hasil clustering bergantung pada centroid awal
Predictive Analysis of Students' Learning Performance Using Data	S. M. F. D. Syed Mustapha	Diperlukan pemilihan fitur yang tepat agar model prediksi kinerja akademik	Membandingkan berbagai metode feature selection (Boruta, Lasso, RFE, Random	Dataset OULAD (Open University Learning Analytics Dataset)	Gradient Boosting (regresi), Random Forest (klasifikasi), Lasso, RFE	Gradient Boosting dengan Boruta menghasilkan MAE 12,93 dan RMSE 18,28;	Fokus pada prediksi performa umum, bukan status kelulusan; akurasi klasifikasi

Judul Penelitian	Nama Peneliti	Permasalahan	Solusi yang Ditawarkan	Data yang Digunakan	Model yang Digunakan	Hasil	Kekurangan
Mining Techniques: A Comparative Study of Feature Selection Methods [15]		lebih efektif, baik untuk regresi maupun klasifikasi	Forest Importance) untuk meningkatkan performa model			klasifikasi terbaik mencapai akurasi 78%	masih relatif moderat
Predicting Student Performance and Its Influential Factors Using Hybrid Regression and Multi-Label Classification [16]	Abdullah Alshanqiti, Abdallah Namoun	Banyak model hanya berfokus pada akurasi prediksi tanpa menjelaskan faktor-faktor yang memengaruhi kinerja akademik	Mengusulkan model hybrid (Collaborative Filtering, Fuzzy Rules, Lasso Regression) serta multi-label classifier untuk mengidentifikasi faktor pengaruh	Tujuh dataset publik pendidikan tinggi	Hybrid Regression, Lasso Regression, Self-Organizing Map (multi-label)	Model hybrid meningkatkan akurasi prediksi dan mampu mengidentifikasi beberapa faktor yang memengaruhi kinerja akademik	Arsitektur model kompleks dan sulit diimplementasikan ; fokus pada prediksi nilai, bukan klasifikasi status kelulusan
Advanced Machine Learning Techniques to Improve Hydrological Prediction: A Comparative	Vijendra Kumar et al.	Perlu model machine learning yang stabil dan akurat untuk prediksi data kompleks dengan kombinasi fitur	Membandingkan berbagai algoritma ML secara komprehensif untuk menentukan model paling stabil	Data time-series aliran sungai (hidrologi)	CatBoost, KNN, Lasso, Ridge, RF, XGBoost, LGBM, MLP	CatBoost menunjukkan performa terbaik dan paling stabil di berbagai metrik evaluasi	Studi berada di domain non-pendidikan; tidak membahas interpretabilitas

Judul Penelitian	Nama Peneliti	Permasalahan	Solusi yang Ditawarkan	Data yang Digunakan	Model yang Digunakan	Hasil	Kekurangan
Analysis of Streamflow Prediction Model [17]		numerik dan kategorikal					atau data akademik

Model prediksi kinerja mahasiswa yang bersifat spesifik per mata kuliah diketahui tidak berskala dengan baik dan berisiko tinggi mengalami *overfitting* ketika jumlah data historis mata kuliah terbatas. Untuk mengatasi permasalahan tersebut, dikembangkan model prediksi generik yang dapat diterapkan lintas mata kuliah dengan memanfaatkan data dari LMS, SMS, dan EMS. Hasil perbandingan beberapa algoritma menunjukkan bahwa CatBoost memberikan performa terbaik dengan nilai AUC sebesar 0,87, meskipun tingkat akurasi yang dicapai masih berada pada kisaran 75%, sehingga masih terdapat ruang untuk peningkatan performa [11].

Pendekatan lain berfokus pada prediksi Indeks Prestasi Kumulatif dengan permasalahan utama berupa kompleksitas pengaturan *hyperparameter* pada model *boosting*. Solusi yang digunakan adalah optimasi *hyperparameter* XGBoost menggunakan *Bayesian Optimization* melalui *framework* Optuna. Evaluasi pada data mahasiswa menunjukkan peningkatan kinerja yang signifikan dibandingkan model *baseline*, dengan nilai R^2 mencapai 0,8456. Namun, pendekatan ini masih terbatas pada prediksi regresi IPK dan belum mengakomodasi klasifikasi status kelulusan mahasiswa [8].

Ketidakseimbangan data pada kasus prediksi multiclass menjadi fokus penelitian berikutnya. Untuk mengurangi misklasifikasi pada kelas minoritas, diterapkan pendekatan gabungan antara SMOTE dan seleksi fitur. Hasil evaluasi menunjukkan bahwa kombinasi *Random Forest* dengan pendekatan tersebut menghasilkan nilai F-*measure* yang sangat tinggi, mencapai 99,5%. Nilai evaluasi yang ekstrem ini mengindikasikan potensi *overfitting*, sehingga menjadi keterbatasan utama dalam generalisasi model [12].

Permasalahan tingginya tingkat drop out pada mata kuliah pemrograman dasar ditangani dengan menerapkan model *deep learning* berbasis LSTM yang dikombinasikan dengan teknik *oversampling* SMOTE dan GAN. Kombinasi LSTM dan GAN menghasilkan akurasi tertinggi sebesar 98,3%. Meskipun demikian, data sintetis hasil GAN menyebabkan penurunan performa yang signifikan pada model

machine learning tradisional serta membutuhkan biaya komputasi yang relatif tinggi [10].

Pendekatan lain mengeksplorasi penggunaan faktor non-akademik, seperti motivasi belajar dan kondisi psikologis, sebagai prediktor kinerja akademik mahasiswa. Seleksi fitur dilakukan menggunakan *Chi-Square Test* terhadap data kuesioner, kemudian diuji menggunakan beberapa algoritma klasifikasi. Hasil evaluasi menunjukkan bahwa XGBoost memberikan performa terbaik dengan akurasi sebesar 86,52%. Keterbatasan pendekatan ini terletak pada penggunaan data subjektif dan cakupan dataset yang hanya berasal dari satu institusi [13].

Penelitian selanjutnya menunjukkan bahwa hasil prediksi kinerja akademik dapat berbeda secara signifikan tergantung pada karakteristik institusi. Dengan menggunakan dataset berukuran kecil dan menerapkan seleksi fitur berbasis statistik, beberapa algoritma klasifikasi dibandingkan. Model *Support Vector Classifier* menunjukkan performa paling stabil dengan akurasi sebesar 86,18%. Namun, ukuran dataset yang terbatas menyebabkan hasil penelitian sulit digeneralisasi ke konteks yang lebih luas [14].

Pendekatan berbasis *clustering* digunakan untuk mengidentifikasi potensi kegagalan mahasiswa pada mata kuliah prasyarat sebelum akhir semester. Dengan menerapkan algoritma K-*Means*, mahasiswa dikelompokkan ke dalam tiga kategori berdasarkan karakteristik akademik. Hasil evaluasi menunjukkan tingkat akurasi sebesar 93%. Keterbatasan pendekatan ini terletak pada ukuran dataset yang sangat kecil serta sensitivitas algoritma terhadap penentuan *centroid* awal [5].

Penelitian lain menekankan pentingnya pemilihan fitur dalam meningkatkan performa model prediksi kinerja akademik. Berbagai metode seleksi fitur dibandingkan pada dataset pendidikan tinggi yang berskala besar. Hasil evaluasi menunjukkan bahwa kombinasi *Gradient Boosting* dengan metode seleksi fitur tertentu memberikan performa regresi yang lebih baik, sementara akurasi klasifikasi masih berada pada tingkat moderat. Pendekatan ini masih berfokus pada prediksi performa akademik umum, bukan status kelulusan mahasiswa [15].

Pendekatan *hybrid* yang mengombinasikan regresi dan klasifikasi multi-label digunakan untuk mengidentifikasi faktor-faktor yang memengaruhi kinerja akademik mahasiswa. Model yang diusulkan mampu meningkatkan akurasi prediksi sekaligus memberikan informasi mengenai faktor yang berkontribusi. Namun, kompleksitas arsitektur model menjadi kendala utama dalam implementasi praktis, dan fokus penelitian masih terbatas pada prediksi nilai akademik [16].

Sebuah studi di luar domain pendidikan menunjukkan bahwa CatBoost memiliki kestabilan dan performa yang unggul pada data kompleks dengan kombinasi fitur numerik dan kategorikal. Meskipun diterapkan pada domain yang berbeda, temuan ini relevan secara metodologis karena menunjukkan ketangguhan CatBoost dalam menangani data heterogen. Namun, studi tersebut tidak membahas aspek interpretabilitas model maupun penerapannya pada data akademik [17].

Berdasarkan kajian terhadap penelitian terdahulu, terlihat bahwa pemilihan algoritma *machine learning* sangat bergantung pada tujuan prediksi, karakteristik data, serta konteks implementasi. Beberapa penelitian menunjukkan performa tinggi dengan menggunakan model *deep learning* seperti *Long Short-Term Memory* (LSTM) atau pendekatan berbasis *Generative Adversarial Network* (GAN). Namun, pendekatan tersebut umumnya memerlukan ukuran *dataset* yang besar, biaya komputasi yang tinggi, serta menghasilkan model dengan tingkat kompleksitas yang relatif sulit untuk dijelaskan dan diimplementasikan dalam pemantauan akademik program studi. Pendekatan ini kurang sesuai untuk data akademik yang berskala terbatas dan membutuhkan transparansi dalam pengambilan keputusan. Model berbasis *boosting* seperti XGBoost dan *Random Forest* juga banyak digunakan, tetapi sering kali memerlukan proses optimasi *hyperparameter* yang kompleks dan cenderung berfokus pada prediksi nilai atau performa tunggal, bukan klasifikasi status kelulusan multikelas. Selain itu, beberapa penelitian menunjukkan nilai evaluasi yang sangat tinggi pada *dataset* tertentu, yang mengindikasikan potensi *overfitting* dan keterbatasan generalisasi ketika diterapkan pada konteks institusi lain. Pendekatan *clustering* seperti K-Means tidak secara langsung dirancang untuk tugas klasifikasi terawasi dan sangat sensitif terhadap ukuran dataset serta inisialisasi parameter, sehingga kurang tepat

digunakan untuk prediksi status kelulusan yang membutuhkan keputusan kelas yang jelas. Model *Support Vector Classifier* dan metode *hybrid* menawarkan stabilitas pada kondisi tertentu, tetapi memiliki keterbatasan dalam hal interpretabilitas dan fleksibilitas ketika dihadapkan pada data akademik yang heterogen dan tidak seimbang. Dengan mempertimbangkan keterbatasan tersebut, penelitian ini memilih CatBoost, *Logistic Regression* dengan regularisasi L1 (Lasso), dan K-*Nearest Neighbor* (KNN) karena ketiganya menawarkan keseimbangan antara kompleksitas model, stabilitas performa, dan kesesuaian dengan karakteristik data akademik. CatBoost dipilih karena kemampuannya menangani hubungan non-linear, fitur heterogen, serta distribusi kelas tidak seimbang pada dataset berukuran terbatas. *Logistic Regression* L1 digunakan untuk merepresentasikan pendekatan linear yang sederhana dan interpretatif, sehingga memungkinkan evaluasi sejauh mana pola linear dapat menjelaskan status kelulusan mahasiswa. Sementara itu, K-*Nearest Neighbor* digunakan sebagai pendekatan non-parametrik untuk mengevaluasi kemampuan prediksi berbasis kedekatan data tanpa asumsi bentuk fungsi tertentu. Kombinasi ketiga model ini memungkinkan perbandingan yang terstruktur dan komprehensif antar pendekatan pembelajaran yang berbeda, sekaligus menjaga relevansi metodologis dan keterterapan hasil penelitian pada Program Studi Sistem Informasi Universitas Multimedia Nusantara.

2.2 Tinjauan Teori

Berikut adalah teori yang dikemukakan dalam penelitian ini. Teori tersebut yang menjadi acuan dalam menggali struktur yang lebih mendalam dalam hal yang akan dibahas.

2.2.1 Kinerja Akademik Mahasiswa

Kinerja akademik mahasiswa merupakan ukuran yang digunakan untuk menilai sejauh mana mahasiswa mampu mencapai tujuan pembelajaran yang telah ditetapkan oleh perguruan tinggi [18]. Pengukuran ini umumnya dilakukan menggunakan indikator kuantitatif seperti Indeks Prestasi Semester (IPS) dan Indeks Prestasi Kumulatif (IPK), yang dihitung berdasarkan rata-rata nilai seluruh mata kuliah yang telah diambil. Nilai-nilai tersebut mencerminkan

pencapaian mahasiswa dalam aspek kognitif, yakni penguasaan pengetahuan dan pemahaman materi perkuliahan [19]. Namun, kinerja akademik tidak hanya terbatas pada aspek kuantitatif, melainkan juga mencakup indikator kualitatif seperti kemampuan berpikir kritis, keterampilan komunikasi, kreativitas, serta partisipasi aktif dalam kegiatan akademik maupun non-akademik yang mendukung proses pembelajaran [20].

Faktor-faktor yang memengaruhi kinerja akademik mahasiswa sangat beragam, mulai dari faktor internal seperti motivasi belajar, disiplin, strategi belajar, dan kemampuan manajemen waktu, hingga faktor eksternal seperti dukungan keluarga, kondisi lingkungan, fasilitas kampus, dan metode pengajaran dosen [21]. Penilaian terhadap kinerja akademik menjadi krusial dalam dunia pendidikan tinggi karena tidak hanya menentukan kelulusan dan capaian akademik formal, tetapi juga memengaruhi peluang mahasiswa dalam memperoleh beasiswa, magang, dan kesempatan kerja setelah lulus [22].

Bagi perguruan tinggi, pemantauan kinerja akademik mahasiswa berperan penting dalam menjaga kualitas pendidikan dan meningkatkan daya saing institusi [23]. Data kinerja akademik dapat digunakan untuk mengidentifikasi mahasiswa yang berisiko rendah performa sehingga dapat diberikan intervensi akademik secara tepat waktu. Selain itu, evaluasi kinerja akademik juga membantu perguruan tinggi dalam melakukan perbaikan kurikulum, metode pembelajaran, dan kebijakan akademik agar lebih relevan dengan kebutuhan industri dan masyarakat. Dengan demikian, pemahaman mendalam mengenai kinerja akademik mahasiswa tidak hanya penting bagi mahasiswa itu sendiri, tetapi juga bagi pengelola pendidikan tinggi dalam upaya mencetak lulusan yang kompeten, berintegritas, dan siap menghadapi tantangan di dunia profesional maupun akademik lanjutan [24].

2.2.2 Fenomena Inflasi IPK

Fenomena inflasi IPK merujuk pada kecenderungan meningkatnya nilai rata-rata Indeks Prestasi Kumulatif (IPK) mahasiswa secara signifikan dalam kurun waktu tertentu tanpa diikuti oleh peningkatan nyata pada

kompetensi akademik maupun keterampilan praktis yang dimiliki lulusan [25]. Dalam kasus pendidikan tinggi di Indonesia, fenomena ini menjadi perhatian serius karena berpotensi menurunkan kredibilitas penilaian akademik dan kualitas lulusan di mata dunia kerja maupun pendidikan lanjutan. Peningkatan IPK seringkali didorong oleh tekanan institusional untuk mempertahankan citra baik kampus, memenuhi standar akreditasi, atau menjaga tingkat kepuasan mahasiswa, yang pada akhirnya dapat mendorong praktik penilaian yang lebih longgar [26].

Selain itu, inflasi IPK dapat mengaburkan perbedaan kemampuan antar mahasiswa, sehingga proses seleksi berbasis prestasi akademik menjadi kurang efektif. Kondisi ini menimbulkan tantangan bagi perekrut kerja maupun lembaga pendidikan lanjutan yang mengandalkan IPK sebagai indikator utama kompetensi [27]. Dalam jangka panjang, tren ini juga dapat memengaruhi motivasi belajar, di mana sebagian mahasiswa mungkin lebih fokus mengejar nilai tinggi ketimbang memperdalam pemahaman dan keterampilan yang relevan. Oleh karena itu, fenomena inflasi IPK perlu ditelaah secara mendalam untuk menemukan pendekatan evaluasi akademik yang lebih objektif, akuntabel, dan mencerminkan kemampuan sesungguhnya dari mahasiswa.

2.2.3 Faktor Internal dan Eksternal yang Mempengaruhi Prestasi Akademik

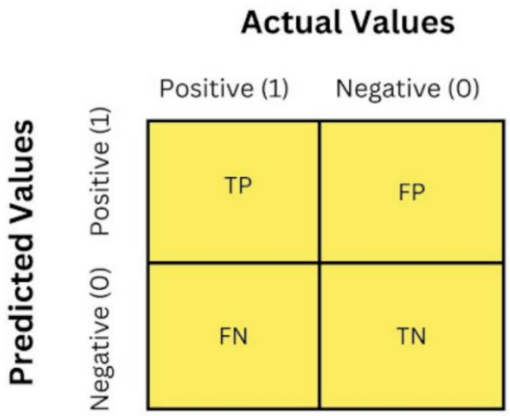
Faktor internal dan eksternal memiliki peran penting dalam membentuk prestasi akademik mahasiswa. Faktor internal mencakup aspek-aspek yang berasal dari dalam diri mahasiswa, seperti motivasi belajar, minat terhadap bidang studi, kedisiplinan, kemampuan mengelola waktu, strategi belajar yang efektif, serta kondisi kesehatan fisik dan mental. Motivasi yang tinggi, misalnya, dapat mendorong mahasiswa untuk menginvestasikan lebih banyak waktu dan usaha dalam memahami materi perkuliahan. Sebaliknya, rendahnya disiplin atau kurangnya keterampilan manajemen waktu dapat menghambat pencapaian akademik meskipun mahasiswa memiliki kemampuan intelektual yang baik [28].

Di sisi lain, faktor eksternal meliputi pengaruh dari luar diri mahasiswa, seperti dukungan keluarga, kondisi sosial ekonomi, lingkungan tempat tinggal, fasilitas kampus, kualitas pengajaran dosen, dan kebijakan akademik yang berlaku. Dukungan emosional dan finansial dari keluarga dapat menjadi pendorong utama keberhasilan akademik, sementara lingkungan belajar yang kondusif dan akses terhadap fasilitas pendukung seperti perpustakaan atau laboratorium dapat meningkatkan kualitas proses belajar [29]. Sebaliknya, faktor eksternal yang kurang mendukung, seperti keterbatasan fasilitas atau metode pengajaran yang kurang efektif, dapat menghambat perkembangan akademik mahasiswa [30].

Interaksi antara faktor internal dan eksternal ini bersifat dinamis dan saling memengaruhi. Mahasiswa dengan motivasi tinggi dapat mengatasi keterbatasan lingkungan, sementara lingkungan yang mendukung dapat membantu mahasiswa dengan motivasi rendah untuk tetap berprestasi. Pemahaman yang komprehensif terhadap kedua faktor ini penting dalam merancang strategi intervensi pendidikan yang tepat, sehingga dapat meningkatkan prestasi akademik secara menyeluruh dan berkelanjutan [31].

2.2.4 Confusion Matrix

Confusion matrix adalah alat evaluasi dalam *machine learning* yang dirancang untuk memvisualisasikan kinerja model klasifikasi melalui hubungan antara prediksi model dan data aktual dalam bentuk tabel matriks [32]. Matriks ini terdiri dari empat elemen utama, diantaranya *True Positive* (TP) yang mencatat jumlah prediksi yang benar untuk kelas positif; *True Negative* (TN) yang menghitung prediksi benar untuk kelas negatif; *False Positive* (FP) yang menggambarkan kesalahan prediksi di mana model memprediksi positif meskipun sebenarnya negatif; dan *False Negative* (FN) yang mencatat kesalahan ketika model memprediksi negatif untuk data yang sebenarnya positif. Elemen-elemen ini membentuk dasar untuk perhitungan berbagai metrik penting yang membantu dalam mengevaluasi performa model klasifikasi secara mendalam yang dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Confusion Matric*

Sumber: [32]

Akurasi adalah metrik yang menghitung proporsi prediksi yang benar terhadap seluruh jumlah data.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Rumus 2.1 Akurasi

Sumber: [33]

Metrik ini memberikan gambaran umum tentang seberapa sering model membuat prediksi yang benar, tetapi dapat menjadi kurang informatif jika *dataset* memiliki ketidakseimbangan kelas, misalnya ketika satu kelas jauh lebih dominan dibandingkan kelas lainnya.

Presisi, di sisi lain, berfokus pada keandalan prediksi positif. Presisi menunjukkan berapa proporsi prediksi positif yang relevan.

$$Presisi = \frac{TP}{TP + FP} \tag{2.2}$$

Rumus 2.2 Presisi

Sumber [33]

Presisi sangat penting di mana *False Positive* memiliki konsekuensi serius, seperti dalam diagnosis medis atau sistem keamanan.

Recall, atau sensitivitas, mengukur seberapa baik model mendeteksi semua kasus positif dalam *dataset*.

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

Rumus 2.3 Recall

Sumber: [33]

Metrik ini relevan dalam situasi di mana penting untuk meminimalkan *False Negative*, seperti mendeteksi penyakit yang memerlukan intervensi segera.

F1-*score* adalah rata-rata harmonis antara *Presisi* dan *Recall*, memberikan metrik yang lebih seimbang terutama dalam kasus ketidakseimbangan kelas.

$$F1\ Score = 2\frac{Presisi\ .\ Recall}{Presisi + Recall} \tag{2.4}$$

Rumus 2.4 F1 Score

Sumber: [33]

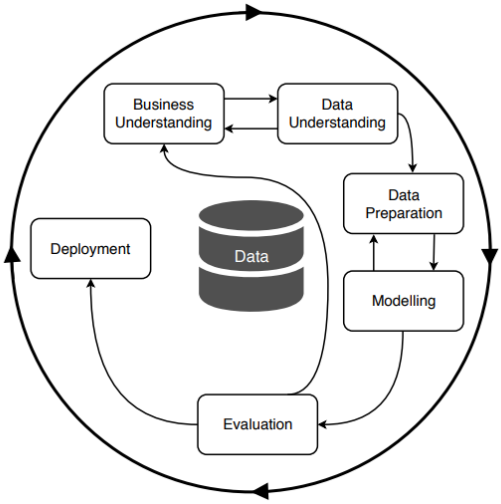
F1-*score* sangat berguna untuk mengukur keseimbangan antara presisi dan *recall* dalam satu nilai yang komprehensif.

Confusion matrix memungkinkan identifikasi pola kesalahan spesifik yang dibuat oleh model, seperti kecenderungan salah memprediksi kelas tertentu. Objek yang dievaluasi meliputi hasil prediksi model dan label sebenarnya, memberikan informasi mendalam tentang kinerja model dalam hasil yang lebih kompleks. Dengan demikian, *confusion matrix* bukan hanya alat evaluasi, tetapi juga panduan strategis untuk mengarahkan perbaikan model dengan fokus yang lebih tajam pada kelemahan yang teridentifikasi [32]. Hal ini menjadikannya alat yang esensial dalam pengembangan model *machine learning*, khususnya untuk meningkatkan kualitas dan reliabilitas sistem prediksi.

2.3 *Framework* dan Algoritma

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

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) merupakan *framework* paling popular per 2024 yang digunakan untuk mengarahkan proses analisis data secara sistematis [34]. *Framework* ini membagi proses menjadi enam tahap utama, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*. Keenam tahap tersebut bersifat iteratif, artinya peneliti dapat kembali ke tahap sebelumnya untuk melakukan perbaikan bila ditemukan kekurangan pada tahap selanjutnya [35]. Pendekatan ini memastikan setiap langkah memiliki keterkaitan yang kuat, sehingga hasil akhir tidak hanya fokus pada aspek teknis, tetapi juga selaras dengan tujuan penelitian. Dalam pendidikan, CRISP-DM memberikan kerangka kerja yang jelas untuk mengelola proses prediksi kinerja akademik mahasiswa, mulai dari memahami masalah pendidikan yang dihadapi hingga penerapan hasil prediksi dalam pengambilan Keputusan [36].



Gambar 2.2 Alur CRISP-DM

Sumber: [37]

Kelebihan utama CRISP-DM adalah fleksibilitasnya untuk diterapkan di berbagai bidang, termasuk pendidikan, bisnis, dan industri. Struktur tahapan yang jelas memudahkan peneliti atau praktisi dalam mengatur alur kerja, meminimalkan kesalahan, serta meningkatkan replikasi penelitian [38]. Selain itu, sifat iteratifnya memungkinkan evaluasi dan penyempurnaan model secara berulang sehingga kualitas prediksi dapat meningkat. Namun,

CRISP-DM juga memiliki keterbatasan. *Framework* ini bersifat umum sehingga memerlukan penyesuaian khusus agar benar-benar sesuai dengan karakteristik data dan domain penelitian [39]. Tahap *data preparation* yang menjadi salah satu bagian terpenting sering kali memakan waktu lebih lama dibanding tahap lainnya, terutama ketika data yang digunakan bersifat heterogen, tidak terstruktur, atau memiliki kualitas rendah. Hal ini menuntut peneliti untuk memiliki keterampilan teknis yang baik serta pemahaman mendalam terhadap data yang dianalisis [40].

2.3.2 CatBoost

CatBoost (*Category Boosting*) adalah *algoritma ensemble learning* berbasis pohon keputusan yang merupakan salah satu implementasi *Gradient Boosting Decision Tree* (GBDT) yang modern dan berkinerja tinggi, dikembangkan oleh Yandex. Algoritma ini dirancang khusus untuk mengatasi kekurangan model *boosting* tradisional, seperti *overfitting* dan penanganan fitur kategorikal yang kurang optimal, yang mana sangat relevan dalam kasus data survei yang umumnya mengandung banyak variabel kategorikal [41]. CatBoost memperkenalkan dua inovasi utama yaitu *Ordered Boosting* dan penanganan fitur kategorikal yang inovatif.

Ordered Boosting memastikan bahwa estimasi *gradient* untuk melatih pohon saat ini dihitung berdasarkan model yang dilatih pada subset data sebelumnya, bukan keseluruhan data, sehingga mengurangi bias prediksi. Inovasi ini menghasilkan generalisasi model yang lebih baik, terutama pada dataset kecil atau yang memiliki *noise* [42]. Dalam penelitian prediksi akademik, *CatBoost* unggul karena mampu menghasilkan akurasi tinggi sambil menyediakan estimasi pentingnya fitur (*feature importance*), yang dapat menjadi dasar yang baik untuk interpretasi menggunakan SHAP.

Prinsip dasar *CatBoost*, sebagai bagian dari keluarga *Gradient Boosting*, adalah membangun model secara aditif, di mana prediksi akhir merupakan penjumlahan dari kontribusi setiap pohon. Model berupaya

meminimalkan loss function (L) dengan menambahkan fungsi basis (f_k) (pohon keputusan) secara bertahap [43].

Secara matematis, prediksi model pada iterasi ke M dapat direpresentasikan sebagai Berikut pada Persamaan 2.5:

$$F_M(x) = \sum_{k=1}^{M} f_k(x) \tag{2.5}$$

Persamaan 2.5 CatBoost dengan iterasi ke-M

Sumber: [43]

di mana $f_k(x)$ adalah prediksi dari pohon keputusan yang baru dilatih, yang bertujuan meminimalkan *loss function L* dari prediksi sebelumnya $f_{k-1}(x)$.

CatBoost menggunakan persamaan *loss function* yang dioptimalkan untuk regresi (memprediksi IPK kontinu) atau klasifikasi (memprediksi kategori IPK). Misalnya, untuk tugas prediksi nilai seperti IPK, *loss function* yang umum digunakan adalah *Mean Squared Error* (MSE), di mana model dilatih untuk meminimalkan L pada Persamaan 2.6.

$$L(\hat{y}, y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.6}$$

Persamaan 2.6 MSE untuk meminimalkan L

Sumber: [43]

Namun, inovasi utama terletak pada proses pelatihan pohon f_k. Berbeda dengan *XGBoost* yang menggunakan *gradient* dari prediksi sebelumnya untuk melatih pohon berikutnya (yang dapat menyebabkan kebocoran informasi), *CatBoost* menggunakan estimasi *gradient* yang dihitung dari subset data yang berbeda (*Ordered Boosting*), yang bertujuan untuk mengurangi bias prediksi dalam proses pembentukan pohon. Fitur ini menjadikan *CatBoost* pilihan kuat sebagai model *boosting* modern yang unggul dalam akurasi dan ketahanan terhadap *overfitting* [44].

2.3.3 Logistic Regression L1 (Lasso)

Logistic Regression merupakan metode klasifikasi yang digunakan untuk memodelkan hubungan antara sekumpulan variabel independen dengan probabilitas suatu kelas menggunakan fungsi logistik. Model ini banyak diterapkan karena strukturnya yang sederhana, stabil, dan mudah diinterpretasikan. Pada kasus klasifikasi multikelas, *Logistic Regression* dapat diperluas melalui pendekatan *one-vs-rest* atau multinomial untuk menangani lebih dari dua kategori keluaran.

Regularisasi L1, yang dikenal sebagai Lasso (*Least Absolute Shrinkage and Selection Operator*), merupakan teknik regularisasi yang digunakan untuk mengendalikan kompleksitas model *Logistic Regression*. Teknik ini bekerja dengan menambahkan penalti berupa jumlah nilai absolut dari koefisien ke dalam fungsi *loss*. Penalti tersebut mendorong sebagian koefisien menjadi tepat nol, sehingga fitur-fitur yang kurang relevan secara otomatis dieliminasi dari model.

Karakteristik utama Lasso terletak pada kemampuannya melakukan seleksi fitur secara simultan dengan proses pelatihan model. Berbeda dengan regularisasi L2 yang hanya mengecilkan nilai koefisien tanpa menghilangkannya, L1 menghasilkan model yang lebih jarang (*sparse*) karena hanya mempertahankan fitur-fitur yang memiliki kontribusi signifikan terhadap prediksi. Hal ini membuat model lebih sederhana dan lebih mudah untuk dianalisis.

Penggunaan *Logistic Regression* dengan regularisasi L1 juga membantu mengatasi permasalahan multikolinearitas antar variabel independen. Ketika terdapat fitur-fitur yang saling berkorelasi, Lasso cenderung memilih salah satu fitur yang paling representatif dan menekan koefisien fitur lainnya menjadi nol. Mekanisme ini meningkatkan stabilitas model dan mengurangi risiko overfitting, terutama pada kondisi data dengan jumlah fitur yang relatif besar.

Selain aspek seleksi fitur, interpretabilitas menjadi keunggulan lain dari Logistic Regression L1. Koefisien yang tidak bernilai nol dapat langsung diinterpretasikan sebagai arah dan besaran pengaruh suatu fitur terhadap probabilitas kelas tertentu. Dengan demikian, model ini tidak hanya berfungsi sebagai alat klasifikasi, tetapi juga sebagai sarana untuk memahami struktur hubungan antara variabel independen dan hasil prediksi

2.3.4 K-Nearest Neighbor

K-*Nearest Neighbor* (KNN) merupakan algoritma klasifikasi berbasis *instance* yang bekerja dengan mengelompokkan suatu data baru berdasarkan kedekatannya dengan data pelatihan yang telah ada. Berbeda dengan algoritma berbasis model, KNN tidak membangun model secara eksplisit pada tahap pelatihan. Seluruh data pelatihan disimpan dan digunakan secara langsung pada saat proses prediksi.

Prinsip dasar KNN adalah menentukan sejumlah tetangga terdekat sebanyak $k$dari suatu data uji berdasarkan ukuran jarak tertentu. Jarak yang paling umum digunakan adalah *Euclidean distance*, meskipun metrik jarak lain seperti Manhattan atau Minkowski juga dapat diterapkan tergantung pada karakteristik data. Kelas dari data uji kemudian ditentukan berdasarkan mayoritas kelas dari $k$tetangga terdekat tersebut.

Pemilihan nilai $k$memiliki peran penting dalam kinerja algoritma KNN. Nilai k yang terlalu kecil dapat menyebabkan model menjadi sensitif terhadap *noise* pada data, sedangkan nilai k yang terlalu besar dapat mengaburkan batas antar kelas. Oleh karena itu, pemilihan $k$umumnya dilakukan melalui proses evaluasi atau validasi untuk memperoleh keseimbangan antara bias dan varians.

KNN sangat bergantung pada representasi fitur dan skala data. Perbedaan skala antar variabel dapat memengaruhi perhitungan jarak, sehingga proses normalisasi atau standarisasi data sering diperlukan sebelum algoritma KNN diterapkan. Tanpa penyesuaian skala, fitur dengan rentang nilai yang

lebih besar dapat mendominasi perhitungan jarak dan menghasilkan prediksi yang kurang akurat.

Keunggulan utama KNN terletak pada kesederhanaan konsep dan kemampuannya menangkap pola lokal dalam data. Algoritma ini tidak memerlukan asumsi distribusi tertentu dan dapat digunakan pada berbagai jenis data. Namun, KNN memiliki keterbatasan dari sisi efisiensi komputasi, terutama pada dataset berukuran besar, karena proses prediksi membutuhkan perhitungan jarak terhadap seluruh data pelatihan.

Selain itu, performa KNN dapat menurun pada data dengan dimensi tinggi, yang dikenal sebagai permasalahan *curse of dimensionality*. Ketika jumlah fitur meningkat, perbedaan jarak antar data menjadi kurang signifikan, sehingga kemampuan KNN dalam membedakan kelas menjadi berkurang. Oleh karena itu, pemilihan fitur yang relevan dan pengurangan dimensi data sering menjadi langkah pendukung dalam penerapan algoritma KNN.

2.3.5 Hyperopt

Hyperopt merupakan pustaka optimasi yang digunakan untuk melakukan pencarian *hyperparameter* pada model *machine learning*. *Hyperparameter* adalah parameter yang ditetapkan sebelum proses pelatihan dan tidak dipelajari langsung dari data, seperti learning rate, jumlah iterasi, kedalaman model, dan parameter regularisasi. Nilai *hyperparameter* memengaruhi proses pembelajaran model serta hasil prediksi yang dihasilkan.

Proses optimasi pada Hyperopt dilakukan dengan mendefinisikan sebuah fungsi objektif yang akan diminimalkan atau dimaksimalkan. Fungsi objektif ini umumnya merepresentasikan kinerja model berdasarkan metrik evaluasi tertentu. Setiap kombinasi *hyperparameter* yang diuji akan digunakan untuk melatih model, kemudian dievaluasi, dan hasil evaluasi tersebut dicatat sebagai dasar pengambilan keputusan pada iterasi berikutnya.

Hyperopt menyediakan beberapa algoritma pencarian, dengan *Tree-structured Parzen Estimator* (TPE) sebagai pendekatan yang paling umum digunakan. Algoritma TPE memodelkan hubungan antara hyperparameter dan nilai fungsi objektif menggunakan pendekatan probabilistik. Ruang pencarian *hyperparameter* dibagi menjadi dua kelompok, yaitu parameter yang menghasilkan nilai fungsi objektif lebih baik dan parameter yang menghasilkan nilai lebih rendah. Berdasarkan pemodelan ini, TPE memilih kombinasi hyperparameter yang memiliki peluang lebih tinggi untuk memberikan hasil yang lebih baik pada iterasi selanjutnya.

Ruang pencarian *hyperparameter* dalam Hyperopt didefinisikan menggunakan berbagai jenis distribusi, seperti uniform, log-uniform, dan pilihan diskrit. Pendekatan ini memungkinkan eksplorasi parameter numerik maupun kategorikal secara fleksibel. Selain itu, jumlah percobaan optimasi dapat dibatasi melalui parameter tertentu, sehingga proses pencarian dapat dikendalikan sesuai dengan ketersediaan sumber daya komputasi.

Hasil dari proses optimasi Hyperopt berupa satu set nilai hyperparameter yang memberikan nilai fungsi objektif terbaik berdasarkan kriteria evaluasi yang digunakan. Hyperopt tidak menjamin bahwa solusi yang diperoleh merupakan optimum global, namun pendekatan ini dirancang untuk menemukan konfigurasi parameter yang efisien dalam ruang pencarian yang luas dengan jumlah evaluasi yang terbatas.

2.4 Tools

Alat yang digunakan untuk merancang model sekaligus membangun *prototype* sederhana untuk penelitian ini.

2.4.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang serbaguna, dikenal karena sintaksisnya yang sederhana dan mudah dipahami, menjadikannya pilihan utama untuk pengembang dari berbagai tingkat keahlian. Bahasa ini dirancang untuk meningkatkan produktivitas dengan pendekatan pemrograman yang intuitif, mendukung paradigma pemrograman

berorientasi objek, fungsional, dan prosedural [45]. Python memiliki ekosistem yang kaya dengan berbagai pustaka dan *framework* seperti NumPy, Pandas, TensorFlow, dan PyTorch yang memungkinkan pengolahan data, analisis statistik, pengembangan aplikasi berbasis *web*, hingga penerapan kecerdasan buatan dan *machine learning*. Keunggulan Python terletak pada fleksibilitasnya yang memungkinkan pengguna untuk menangani data dalam jumlah besar dengan efisiensi tinggi, seperti pada analisis data berbasis grafik atau jaringan yang kompleks [46].

Bahasa ini juga didukung oleh komunitas global yang aktif, sehingga pengembang dapat dengan mudah mengakses dokumentasi, tutorial, dan solusi untuk masalah yang dihadapi. Salah satu objek utama yang sering dibahas dalam penggunaan Python adalah kemampuannya dalam memproses dan memvisualisasikan data dengan pustaka seperti Matplotlib dan Seaborn yang mempermudah pemahaman pola dalam data. Selain itu, Python dirancang untuk kompatibilitas dengan sistem operasi populer seperti Windows, macOS, dan Linux, sehingga memudahkan integrasi ke dalam berbagai lingkungan pengembangan. Python juga unggul dalam membangun aplikasi lintas platform dengan efisiensi tinggi, sekaligus mampu mendukung pengujian dan *deployment* secara cepat melalui integrasi dengan alat seperti Docker atau Jenkins. Dengan kemampuannya yang terus berkembang, Python menjadi alat yang tak tergantikan bagi berbagai kebutuhan teknologi modern, baik dalam skala kecil maupun proyek-proyek besar yang kompleks [46].

2.4.2 Google Collaboratory

Google Collaboratory adalah platform berbasis *cloud* yang memungkinkan pengguna menjalankan kode Python langsung melalui *browser* tanpa memerlukan konfigurasi perangkat keras atau perangkat lunak tambahan [47]. Platform ini didesain untuk mendukung kebutuhan komputasi data yang intensif, seperti pembelajaran mesin, analisis data, dan pemrosesan teks, dengan akses gratis ke GPU dan TPU yang mempercepat proses perhitungan. Google Colab memanfaatkan Notebook Jupyter, sebuah format interaktif yang

memungkinkan integrasi antara kode, visualisasi, dan dokumentasi dalam satu dokumen yang mudah digunakan dan dibagikan.

Google Colab mencakup data input, model pembelajaran mesin, dan hasil analisis atau prediksi yang semuanya dapat diakses dan diproses dalam lingkungan yang terintegrasi. Pengguna dapat dengan mudah mengunggah *dataset* dari berbagai sumber, seperti Google Drive, GitHub, atau sumber *online* lainnya, untuk diolah dalam proyek berbasis data. Dengan fitur kolaborasi *real-time*, Google Colab memungkinkan banyak pengguna untuk bekerja bersama secara simultan, sehingga mendukung efisiensi kerja tim. Selain itu, *platform* ini kompatibel dengan berbagai pustaka Python populer seperti TensorFlow, PyTorch, dan Scikit-learn yang sering digunakan untuk pengembangan model kecerdasan buatan dan analisis data lanjutan.

Keunggulan lainnya adalah kemampuannya untuk menyimpan dan berbagi *notebook* melalui Google Drive, menjadikan dokumen proyek dapat diakses kapan saja dan dari mana saja tanpa batasan perangkat. Dengan lingkungan yang mendukung integrasi dengan berbagai sumber daya *cloud* dan fleksibilitas penggunaan, Google Colab menjadi alat yang esensial bagi peneliti, pengembang, dan pelajar untuk mengeksplorasi potensi besar data dan teknologi secara efisien. Hal ini menjadikannya salah satu platform utama dalam dunia pemrograman modern, khususnya dalam bidang ilmu data dan kecerdasan buatan.