BAB II

LANDASAN TEORI

2.1 Educational Data Mining (EDM)

Educational Data Mining (EDM) merupakan bidang yang memanfaatkan teknik statistik dan *machine learning* untuk mengekstraksi pola dari dataset pendidikan guna mendukung prediksi *outcome* akademik dan pengambilan keputusan berbasis data pada institusi pendidikan (Kharis & Zili, 2022). EDM telah digunakan untuk memodelkan performa akademik mahasiswa, termasuk dalam tugas klasifikasi multikelas untuk mengelompokkan tingkat capaian akademik berdasarkan fitur demografis dan historis siswa. (Shoukath & Chakkaravarthy, 2025). Prediksi performa akademik merujuk pada *classification task* di mana setiap mahasiswa dipetakan ke dalam kelas performa tertentu berdasarkan data latih historis. Pendekatan ini lebih informatif dibandingkan regresi sederhana karena memberikan segmentasi kelas yang jelas seperti *Excellent*, *Good*, *Average* dan *Below Average* untuk kebutuhan intervensi pendidikan (Johora dkk., 2025).

2.2 Supervised learning

Supervised learning adalah cabang *machine learning* di mana model dilatih menggunakan data berlabel $(\mathbf{x}, y)$, dengan tujuan memperkirakan fungsi prediksi f yang memetakan input $\mathbf{x}$ ke label target y (Bishop, 2006).

2.3 Multi-Class Classification

Multiclass classification merupakan salah satu bentuk *supervised learning* di mana setiap data hanya boleh masuk ke dalam satu pilihan kategori. Jika pada *binary classification* hanya terdapat dua kelas, maka pada *multiclass classification*

jumlah kelas $K > 2$ dan setiap sampel hanya memiliki satu label target (Hastie dkk., 2017).

2.4 SMOTEN

Data balancing dilakukan menggunakan SMOTEN (*Synthetic Minority Over-sampling Technique for Nominal Data*). Metode ini merupakan pengembangan SMOTE yang dirancang khusus untuk data dengan fitur nominal atau kategorikal. SMOTEN menentukan kedekatan antarsampel menggunakan modifikasi *Value Difference Metric* (VDM) dan membentuk nilai setiap fitur sintetis berdasarkan kategori yang paling banyak muncul pada sampel acuan beserta sejumlah tetangga terdekatnya (Chawla dkk., 2002). SMOTEN dapat meningkatkan representasi kelas minoritas tanpa menghasilkan nilai pecahan atau kategori baru yang tidak terdapat dalam domain fitur asli, sehingga sesuai digunakan pada dataset performa akademik mahasiswa yang didominasi oleh fitur kategorikal.

Secara matematis, jarak antarnilai kategorikal diatur formula VDM yang dirumuskan sebagai (Chawla dkk., 2002) (1):

$$\delta(V_1, V_2) = \sum_{i=1}^{n} \left| \frac{C_{1i}}{C_1} - \frac{C_{2i}}{C_2} \right|^q \quad (1)$$

Simbol $\delta(V_1, V_2)$ menyatakan nilai jarak antara kategori V_1 dan kategori V_2. Variabel n menunjukkan jumlah total pada kelas target. Variabel C_1 menghitung total kemunculan untuk nilai kategori V_1. Sementara itu, variabel C_{1i} merinci frekuensi kemunculan nilai V_1 secara khusus pada kelas target ke-i. Selanjutnya, variabel C_2 menghitung total kemunculan untuk nilai kategori V_2. Variabel C_{2i} merinci frekuensi kemunculan nilai V_2 secara khusus pada kelas target ke-i.

Terakhir, variabel q menentukan konstanta pangkat dengan nilai sebesar 1.

Secara matematis, persamaan jarak antara dua sampel kategorikal dirumuskan sebagai (Chawla dkk., 2002) (2):

$$\Delta(X,Y) = \sum_{j=1}^{p} \left[\delta\left(x_j, y_j\right)\right]^r \tag{2}$$

Simbol $\Delta(X,Y)$ menyatakan nilai jarak antara sampel X dan sampel Y. Variabel p menunjukkan jumlah untuk fitur kategorikal. Variabel x_j menentukan nilai sampel X pada fitur ke-j. Sementara itu, variabel y_j menentukan nilai sampel Y pada fitur ke-j. Selanjutnya, simbol $\delta\left(x_j, y_j\right)$ menghitung jarak antara dua nilai kategori pada fitur ke-j. Variabel r menetapkan parameter agregasi untuk jarak tersebut. Terakhir, variabel x_{jk} menjelaskan nilai fitur ke-k pada sampel mahasiswa ke-j.

Secara matematis, pembentukan nilai fitur sintetis dirumuskan sebagai (Chawla dkk., 2002) (3):

$$x_j^{\text{syn}} = \text{mode}\left(\left\{x_j, x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(K)}\right\}\right) \tag{3}$$

Variabel x_j^{syn} menunjukkan nilai fitur ke-j pada sampel sintetis. Sementara itu, variabel x_j menyatakan nilai fitur ke-j pada sampel acuan. Selanjutnya, sekumpulan variabel $x_j, x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(K)}$ merepresentasikan nilai fitur ke-j dari K tetangga terdekat. Terakhir, istilah $mode$ menentukan kategori yang paling sering muncul.

2.5 Stacking Ensemble

Stacking ensemble adalah pendekatan yang menggabungkan beberapa model dasar *base learners* untuk menghasilkan model prediksi yang lebih akurat dan stabil dibanding model tunggal (Dietterich, 2000).

Secara umum, model *ensemble* dapat dituliskan sebagai (Wolpert, 1992) (4).

$$H(x) = f(meta)(h1(x), h2(x), \ldots, hM(x)) \quad (4)$$

Fungsi $H(x)$ menunjukkan hasil prediksi akhir dari metode *stacking*. Variabel x merepresentasikan data masukan pada sistem tersebut. Sementara itu, fungsi $hM(x)$ menghasilkan keluaran dari model dasar ke-m. Selanjutnya, variabel M menentukan jumlah total untuk model dasar. Terakhir, fungsi $f(meta)$ mendefinisikan model meta yang menggabungkan keluaran dari seluruh model dasar.

2.6 Random Forest

Random Forest (RF) merupakan metode *ensemble learning* yang meningkatkan kinerja pohon keputusan (*decision tree*) individu dengan menerapkan teknik *bootstrap aggregating* (*bagging*), yaitu membangun beberapa pohon keputusan berdasarkan *subset* data dan *subset* fitur yang dipilih secara acak. Prinsip dasar *Random Forest* adalah menggabungkan sekumpulan pohon keputusan individu (*ensemble of decision trees*), di mana setiap pohon dilatih menggunakan *subset* data dan *subset* fitur yang berbeda secara acak sehingga menghasilkan model yang lebih robust dan mampu mengurangi risiko *overfitting* (Genuer dkk., 2010).

Random Forest umumnya menggunakan Gini Index atau *Entropy* sebagai kriteria untuk menentukan pemisahan (*split*) terbaik pada setiap simpul (*node*).

Pproses *hyperparameter tuning* model *Random Forest* dilakukan dengan mengevaluasi kedua kriteria tersebut.

Rumus Gini Index dinyatakan sebagai berikut (Banerjee, 2010) (5):

$$Gini = 1 - \Sigma^{c}_{i=1} (P_i)^2 \quad (5)$$

Variabel P_i menyatakan proporsi atau frekuensi relatif data yang termasuk ke dalam kelas ke-i. Sementara itu, variabel c menunjukkan jumlah total untuk seluruh kelas yang ada.

Entropy mengukur tingkat ketidakpastian atau ketidakteraturan data pada suatu node. Nilai *entropy* yang semakin kecil menunjukkan bahwa *node* semakin homogen, sedangkan nilai yang lebih besar menunjukkan bahwa data pada *node* masih bercampur.

Persamaan *Entropy* dinyatakan sebagai berikut (Haymet, 1994) (6):

$$\text{Entropy} = -\Sigma^{c}_{i=1} P_i \times \log_2(P_i) \quad (6)$$

Variabel P_i menyatakan proporsi atau frekuensi relatif data yang termasuk ke dalam kelas ke-i. Sementara itu, variabel c menunjukkan jumlah total untuk seluruh kelas yang ada. Terakhir, operator $\log_2$ mendefinisikan perhitungan logaritma dengan menggunakan basis 2.

2.7 Gradient Boosting

Gradient Boosting (GB) merupakan salah satu metode *ensemble learning* yang membangun pohon keputusan (*decision tree*) secara bertahap (*sequential*), di mana setiap pohon baru dilatih untuk memperbaiki kesalahan prediksi yang dihasilkan oleh pohon sebelumnya. *Gradient Boosting* membangun model secara berurutan sehingga setiap iterasi berfokus pada pengurangan kesalahan (*residual*)

dari model sebelumnya, berbeda dengan metode *bagging* seperti *Random Forest* yang membangun seluruh pohon secara paralel (Bentéjac dkk., 2021).

Gradient Boosting bekerja berdasarkan tiga konsep utama, *yaitu loss function, weak learner,* dan *additive model*. Algoritma ini secara bertahap membangun sejumlah *weak learner* yang digabungkan untuk menghasilkan model prediksi yang lebih akurat. Probabilitas setiap kelas pada permasalahan klasifikasi multikelas diprediksi menggunakan fungsi *softmax* sebagai berikut (Zang & Zhang, 2011) (7).

$$P_{\mathrm{k}}(x) = \frac{e^{F_{K\,M}(x)}}{\sum_{j=1}^{k} e^{F_{K\,M}(x)}} \tag{7}$$

Fungsi $P_k(x)$ menyatakan probabilitas prediksi bahwa data x termasuk ke dalam kelas ke-k. Sementara itu, fungsi $F_{k,m}(x)$ menghasilkan nilai keluaran atau *score* model untuk kelas ke-k setelah M iterasi. Selanjutnya, variabel K menunjukkan jumlah total untuk seluruh kelas yang ada. Terakhir, variabel M menentukan jumlah total untuk iterasi *boosting*.

2.8 Support Vector Classifier

Support Vector Classifier (SVC) merupakan metode *supervised learning* yang dikembangkan berdasarkan algoritma *Support Vector Machine* (SVM). Algoritma ini bertujuan untuk menemukan *hyperplane* optimal yang mampu memisahkan data dari kelas yang berbeda dengan margin maksimum. Dengan menentukan *hyperplane* terbaik, SVC mampu menghasilkan batas keputusan (*decision boundary*) yang optimal sehingga memberikan performa klasifikasi yang baik pada berbagai permasalahan (Ovirianti dkk., 2022).

SVM memanfaatkan sekumpulan fungsi matematis yang disebut kernel dalam proses klasifikasi. Fungsi kernel digunakan untuk memetakan data ke ruang berdimensi lebih tinggi sehingga data yang semula tidak dapat dipisahkan secara linear menjadi lebih mudah dipisahkan. Penelitian ini menggunakan dua jenis kernel, yaitu Linear Kernel dan *Radial Basis Function*

Berikut formula Linear Kernel (Patle & Chouhan, 2013)(8).

$$K(x, y) = \Sigma^{\mathrm{n}}{}_{\mathrm{i}=1}\, x_{\mathrm{i}} y_{\mathrm{i}} \tag{8}$$

Fungsi $K(x, y)$ menyatakan nilai kernel antara dua vektor x dan y. Variabel X yang berisi $(x_1, x_2, \dots, x_n)$ mendefinisikan vektor masukan berdimensi n. Sementara itu, variabel Y yang berisi $(y_1, y_2, \dots, y_n)$ mendefinisikan vektor masukan lain berdimensi n. Selanjutnya, variabel x_i menentukan komponen ke-i dari vektor X. Variabel y_i menentukan komponen ke-i dari vektor Y. Terakhir, variabel n menunjukkan jumlah dimensi pada vektor masukan tersebut.

Berikut formula dari *Radial Basis Function* (RBF) (Patle & Chouhan, 2013) (9).

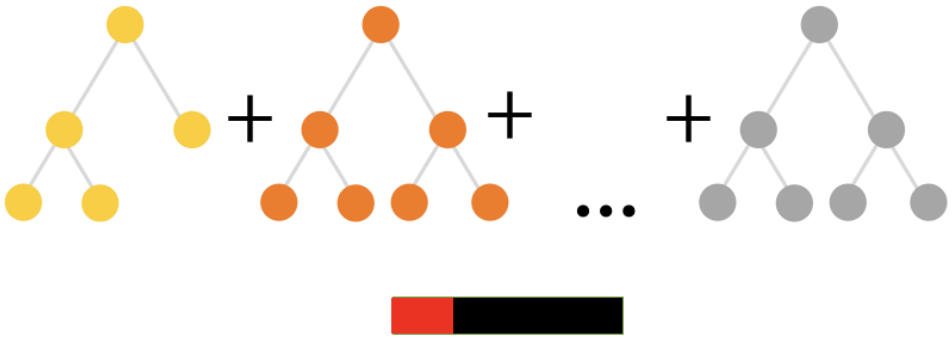
$$\mathrm{K(x, y)} = e^{-\frac{||x-y||^2}{\sigma^2}} \tag{9}$$

Fungsi $K(x, y)$ menyatakan nilai kernel antara dua vektor x dan y. Sementara itu, simbol σ menentukan parameter yang mengendalikan sebaran atau *variance* pada fungsi Gaussian. Terakhir, simbol $||x - y||$ menghitung jarak Euclidean antara vektor x dan vektor y.

2.9 CatBoost

CatBoost (*Categorical Boosting*) adalah algoritma *gradient boosting* yang dirancang untuk menangani fitur kategorikal secara *native* tanpa memerlukan *one-*

hot encoding. CatBoost menggunakan mekanisme *ordered target statistics* untuk mengubah fitur kategorikal menjadi representasi numerik. Ilustrasi pembentukan model catboost terlampir pada gambar Gambar 2. 1 (Prokhorenkova dkk., 2019).



Gambar 2. 1 Ilustrasi model CatBoost (Dokumentasi Catboost, 2017)

CatBoost membentuk model secara bertahap dengan menambahkan pohon keputusan untuk memperbaiki kesalahan model pada iterasi sebelumnya (Dorogush dkk., 2018) (10).

$$\hat{x}_i^k = (\Sigma_{j=1}^{p-1} [x\sigma_j^k = x\sigma_p^k] \cdot y\sigma_j + a \cdot P) / (\Sigma_{j=1}^{p-1} [x\sigma_j^k = x\sigma_p^k] + a) \quad (10)$$

Variabel $\hat{x}_i^k$ menunjukkan nilai hasil *encoding* untuk fitur kategorikal ke-k pada data ke-i. Sementara itu, variabel $x\sigma_p^k$ menyatakan nilai kategori fitur ke-k pada data ke-p berdasarkan urutan permutasi σ. Selanjutnya, variabel $y\sigma_j$ menentukan nilai target dari data sebelumnya pada urutan permutasi tersebut. Simbol $[x\sigma_j^k = x\sigma_p^k]$ mendefinisikan fungsi indikator dengan ketentuan bernilai 1 jika kategorinya sama dan 0 jika berbeda. Variabel P menetapkan nilai *prior* yang umumnya berupa rata-rata target. Terakhir, variabel a menjelaskan parameter *smoothing* untuk mengontrol pengaruh dari *prior* tersebut.

Ordered boosting mengurangi pergeseran distribusi prediksi dengan menghitung statistik kategori secara bertahap berdasarkan permutasi data acak,

sehingga meminimalkan target *leakage* (Prokhorenkova dkk., 2019) (11).

$$r_i^{(t)} = y_i - F_{(\sigma(i)-1)}^{(t-1)}(x_i) \tag{11}$$

Fungsi $ri(t)$ menyatakan nilai *residual* data ke-i pada iterasi ke-t. Variabel yi menentukan target aktual data ke-i. Sementara itu, variabel xi menjelaskan fitur data ke-i. Selanjutnya, variabel y_j menerangkan posisi data ke-i dalam permutasi acak. Terakhir, simbol $F_{(\sigma(i)-1)}^{(t-1)}$ mendefinisikan model yang dibentuk hanya dari data yang muncul sebelumnya dalam permutasi tersebut.

2.10 Class Weighting

Penanganan ketidakseimbangan data dilakukan menggunakan teknik *class weighting* (pembobotan kelas) di dalam algoritma CatBoost. Metode ini merupakan pendekatan berbasis algoritmik yang memberikan bobot lebih tinggi kepada kelas minoritas dan bobot lebih rendah kepada kelas mayoritas selama proses pelatihan model. CatBoost menyesuaikan fungsi kerugian (*loss function*) dengan mengalikan nilai penalti setiap sampel menggunakan bobot kelas yang sesuai, sehingga model dapat memberikan perhatian yang seimbang terhadap seluruh kelas target tanpa harus menduplikasi atau mengubah sampel asli di dalam dataset. Teknik pembobotan ini dapat meningkatkan sensitivitas model terhadap kelas yang jarang muncul, sehingga pendekatan ini sangat efektif digunakan pada dataset dengan distribusi kelas yang tidak seimbang (Nugraha & Syarif, 2023) (12).

Secara matematis, bobot untuk setiap kelas diatur melalui formula *class weighting* standar yang dirumuskan sebagai berikut (Nugraha & Syarif, 2023) (12).

$$W_c = \frac{N}{C \times N_c} \tag{12}$$

Variabel $W_c$menyatakan nilai bobot untuk kelas ke-c. Variabel Nmenunjukkan jumlah total sampel di seluruh dataset. Sementara itu, variabel Cmenghitung jumlah total kelas target yang tersedia. Terakhir, variabel $N_c$merinci jumlah total sampel khusus untuk kelas ke-c.

2.11 SHAP (Shapley Additive Explanations)

Shapley Additive Explanations (SHAP) merupakan salah satu pendekatan *explainable AI* (XAI) yang dibuat untuk menjelaskan output *machine learning* secara kuantitatif dengan cara mendistribusikan kontribusi dari setiap fitur pada hasil prediksi. SHAP merupakan penedekatan berbasis metode *attribution feature additive* yang menggabungkan kelebihan dari berbagai metode interpretasi (Lundberg & Lee, 2017).

Rumus asli nilai Shapley untuk menghitung kontribusi fitur i direpresentasikan pada Rumus (Lundberg & Lee, 2017)(13).

$$\phi_i = \sum_{S \subseteq N\{i\}} \frac{|\,S\,|!\,(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \qquad (13)$$

Variabel N mendefinisikan himpunan semua fitur. Sementara itu, variabel S menyatakan *subset* dari N yang tidak mengandung i. Terakhir, fungsi $f(S)$ menentukan nilai prediksi model ketika hanya fitur dalam *subset* S digunakan.

2.12 LIME

LIME merupakan metode interpretabilitas yang menjelaskan prediksi model kompleks dengan membangun model *surrogate* sederhana di sekitar satu *instance* tertentu. Pendekatan ini bekerja dengan melakukan perturbasi pada data input di sekitar observasi yang dianalisis, kemudian melatih model lokal yang

mudah diinterpretasikan seperti regresi linear untuk mendekati perilaku model asli di wilayah tersebut. LIME memberikan penjelasan lokal, bukan global, sehingga fokus pada alasan prediksi untuk satu individu mahasiswa (Ribeiro dkk., 2016).

Secara matematis, LIME dirumuskan sebagai (Ribeiro dkk., 2016)(14).

$$\xi(x) = arg^{min}_{g \epsilon G}\, L(f, g, \Pi_x) + \Omega(g) \quad (14)$$

Variabel f mendefinisikan model kompleks misalnya *CatBoost*. Sementara itu, variabel g menyatakan model *surrogate* sederhana. Selanjutnya, fungsi $L(f, g, \Pi_x)$ mengukur seberapa baik model *surrogate* g mendekati model asli f di sekitar titik x. Simbol Π_x mendefinisikan fungsi *proximity* atau bobot kedekatan yang memberi bobot lebih besar pada sampel yang dekat dengan *instance* x. Terakhir, fungsi $\Omega(g)$ menentukan regularisasi untuk menjaga kesederhanaan model *surrogate* tersebut.

2.13 Evaluasi Model

Evaluasi model klasifikasi dilakukan untuk mengukur seberapa baik model memprediksi kelas yang benar pada data uji. Evaluasi tidak hanya mengandalkan satu metrik, tetapi kombinasi beberapa ukuran performa agar interpretasi lebih komprehensif. *Confusion matrix* digunakan sebagai dasar perhitungan berbagai metrik dengan menunjukkan distribusi prediksi benar dan salah untuk setiap kelas (Sokolova & Lapalme, 2009).

2.13.1 *Accuracy*

Accuracy mengukur proporsi total prediksi yang benar terhadap seluruh observasi (Sokolova & Lapalme, 2009)(15):

$$Accurracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (15)$$

Variabel TP menunjukkan jumlah data yang benar diprediksi sebagai kelas tertentu. Sementara itu, variabel TN menyatakan jumlah data yang benar diprediksi bukan kelas tersebut. Selanjutnya, variabel FP menentukan data yang salah diprediksi sebagai kelas tersebut. Terakhir, variabel FN menerangkan data yang seharusnya kelas tersebut tetapi diprediksi sebagai kelas lain.

2.13.2 *Precision*

Precision menunjukkan proporsi prediksi positif yang benar, sehingga mengukur ketepatan model dalam memberikan label suatu kelas (Sokolova & Lapalme, 2009) (16).

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

Variabel TP menunjukkan jumlah data yang benar diprediksi sebagai kelas tertentu. Sementara itu, variabel FP menentukan data yang salah diprediksi sebagai kelas tersebut.

2.13.3 *Recall*

Recall menunjukkan kemampuan model dalam menangkap seluruh *instance* yang benar-benar termasuk dalam kelas tersebut (Sokolova & Lapalme, 2009) (17).

$$Recall = \frac{TP}{TP + FN} \quad (17)$$

Variabel TP menunjukkan jumlah data yang benar diprediksi sebagai kelas tertentu. Sementara itu, variabel FN menerangkan data yang seharusnya kelas tersebut tetapi diprediksi sebagai kelas lain.

2.13.4 *F1-Score*

F1-score adalah *harmonic mean* dari *precision* dan *recall*. F1-*score* digunakan ketika diperlukan keseimbangan antara *precision* dan *recall*, terutama pada kasus multikelas (Sokolova & Lapalme, 2009) (18) .

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{18}$$

Istilah *Precision* menunjukkan proporsi prediksi positif yang benar terhadap seluruh prediksi positif. Sementara itu, istilah *Recall* menerangkan proporsi data aktual suatu kelas yang berhasil diprediksi dengan benar oleh model.

2.14 State of The Art

Penelitian dalam bidang *Educational Data Mining* (EDM) menunjukkan bahwa pendekatan *machine learning* semakin luas digunakan untuk memprediksi performa akademik mahasiswa. Berbagai algoritma klasifikasi telah diterapkan, mulai dari metode konvensional hingga pendekatan berbasis *boosting* seperti XGBoost dan LightGBM yang dilaporkan memberikan performa prediktif kompetitif dalam konteks pembelajaran daring dan data akademik institusi pendidikan (Ayulani dkk., 2023; Kumar dkk., 2024).

Beberapa studi mengintegrasikan teknik SMOTEN sebelum proses pelatihan model digunakan untuk meningkatkan performa pada data yang tidak seimbang terutama dalam skema klasifikasi mahasiswa berisiko atau performa rendah (Ayulani dkk., 2023; Johora dkk., 2025; Sahlaoui dkk., 2021). Selain itu, pendekatan *ensemble* dan *stacking* juga banyak digunakan untuk meningkatkan akurasi model dengan mengombinasikan beberapa algoritma dasar (Johora dkk., 2025; Sahlaoui dkk., 2021).

Algoritma *boosting* modern seperti CatBoost mulai dievaluasi karena kemampuannya dalam menangani fitur kategorikal secara *native* dan menunjukkan performa generalisasi yang baik dibandingkan varian *boosting* lainnya (Bentéjac dkk., 2021b). Beberapa penelitian juga telah mengintegrasikan CatBoost dengan metode *Explainable Artificial Intelligence* (XAI) seperti SHAP untuk meningkatkan transparansi prediksi dalam konteks pendidikan (Mingyu dkk., 2022). Namun demikian, sebagian besar penelitian tersebut memformulasikan permasalahan dalam bentuk regresi atau klasifikasi biner, seperti deteksi mahasiswa berisiko ((Mingyu dkk., 2022; Ramaswami dkk., 2022).

Berdasarkan penelitian terdahulu, metode *boosting* dan teknik interpretabilitas telah banyak diterapkan dalam berbagai permasalahan klasifikasi. Namun, kajian yang secara khusus mengevaluasi model CatBoost tunggal pada klasifikasi multikelas, terutama dari aspek performa prediktif serta interpretabilitas global dan lokal, masih relatif terbatas. Penelitian ini berupaya mengisi kesenjangan tersebut melalui evaluasi performa dan interpretabilitas model CatBoost dalam mengklasifikasikan performa akademik mahasiswa.

Ringkasan detail penelitian terdahulu disajikan pada Lampiran 1.1 sebagai dokumentasi *state of the art* yang mendasari perumusan gap penelitian ini.

2.15 Matriks Penelitian

Matriks penelitian merupakan perbandingan antara penelitian sebelumnya dengan penelitian yang akan dilakukan. Indikator untuk melakukan sebuah matriks penelitian, yaitu dari berbagai sumber jurnal yang telah dikaitkan pada *state of the art.* Tabel 2. *1* menggambarkan perbedaan penelelitian yang diusulkan dengan

penelitian-penelitian terkait.

Berdasarkan Tabel 2. *1*, Sebagian besar penelitian terdahulu telah menggunakan berbagai algoritma *machine learning* dan *ensemble learning* untuk memprediksi performa akademik mahasiswa. Penelitian acuan menunjukkan bahwa *stacking ensemble* mampu menghasilkan kinerja yang baik pada klasifikasi multikelas performa akademik mahasiswa. Namun, kajian yang mengevaluasi sejauh mana model CatBoost tunggal, dengan dukungan teknik penyeimbangan data yang sesuai, mampu menyamai atau melampaui kinerja *stacking ensemble* tersebut masih relatif terbatas.

Penelitian yang diusulkan memiliki perbedaan utama dibandingkan penelitian sebelumnya karena mengintegrasikan CatBoost sebagai algoritma utama, SMOTEN untuk menangani ketidakseimbangan data, serta mengkaji kontribusi fitur terhadap hasil prediksi model CatBoost baik secara global maupun lokal, menggunakan SHAP dan LIME. Penelitian ini tidak hanya berfokus pada peningkatan akurasi model, tetapi juga pada transparansi dan kemudahan interpretasi hasil prediksi dalam mendukung pengambilan keputusan akademik.

Tabel 2. 1 Matriks Penelitian

Judul, Peneliti (Tahun)	Tujuan			Algoritma yang Digunakan	XAI		Output	
	Prediksi	Klasifikasi	Regresi		SHAP	LIME	Binary	Multikelas
Academic achievement prediction in higher education through interpretable modelling, Wang & Luo (2024)	✓	-	✓	XGBoost	✓	-	-	✓
Tree-Based Ensemble Methods and Their Applications for Predicting Students Academic Performance Ayulani dkk. (2023)	✓	✓	-	Random Forest; XGBoost; LightGBM; SMOTE	-	-	✓	-
An explainable AI-based approach for predicting undergraduate students academic performance, Johora dkk. (2025)	✓	✓	-	Random Forest; Gradient Boosting; SVC; Stacking; SMOTE	✓	✓	-	✓
Evaluating Boosting Algorithms for Academic Performance Prediction in E Learning Environments,Tirumanadham dkk. (2024)	✓	✓	-	AdaBoost; HistGradientBoosting; CatBoost	-	-	-	✓
Predicting and Interpreting Student Performance Using Ensemble Models and Shapley Additive Explanations, Sahlaoui dkk. (2021)	✓	✓	-	Ensemble Learning; SMOTE	✓	-	-	✓
Performance Analysis of SMOTE and SMOTEN Techniques for Daily Rainfall Classification using XGBoost, (Najwa Laila Anggraini dkk., 2025)	✓	✓	-	XGBoost, SMOTE, SMOTEN	-	-	✓	-
Utilizing Random Forest and XGBoost DataMining Algorithms for Anticipating Students'Academic Performance, Kumar dkk. (2024)	✓	✓	-	Random Forest; XGBoost	-	-	-	✓
Explainable AI Methods for Predicting Student Grades	✓	✓	-	Random Forest	✓	✓	-	✓

and Improving Academic Success, E. Ben George (2025)								
An interpretable prediction method for university student academic crisis warning, Mingyu dkk. (2022)	✓	-	✓	K-Prototypes; CatBoost	✓	-	-	-
On Developing Generic Models for Predicting Student Outcomes in Educational Data Mining, Ramaswami dkk. (2022)	✓	✓	-	Machine Learning Model; CatBoost	-	-	✓	-
A comparative analysis of gradient boosting algorithms, Bentéjac dkk. (2021)	-	✓	-	XGBoost; LightGBM; CatBoost	-	-	✓	✓
A Comparative Study of CatBoost and Double Random Forest for Multi-class Classification, Aldania dkk. (2023)	-	✓	-	CatBoost; Double Random Forest	-	-	-	✓
Ours	✓	✓	-	**CatBoost; SMOTEN**	✓	✓	-	✓