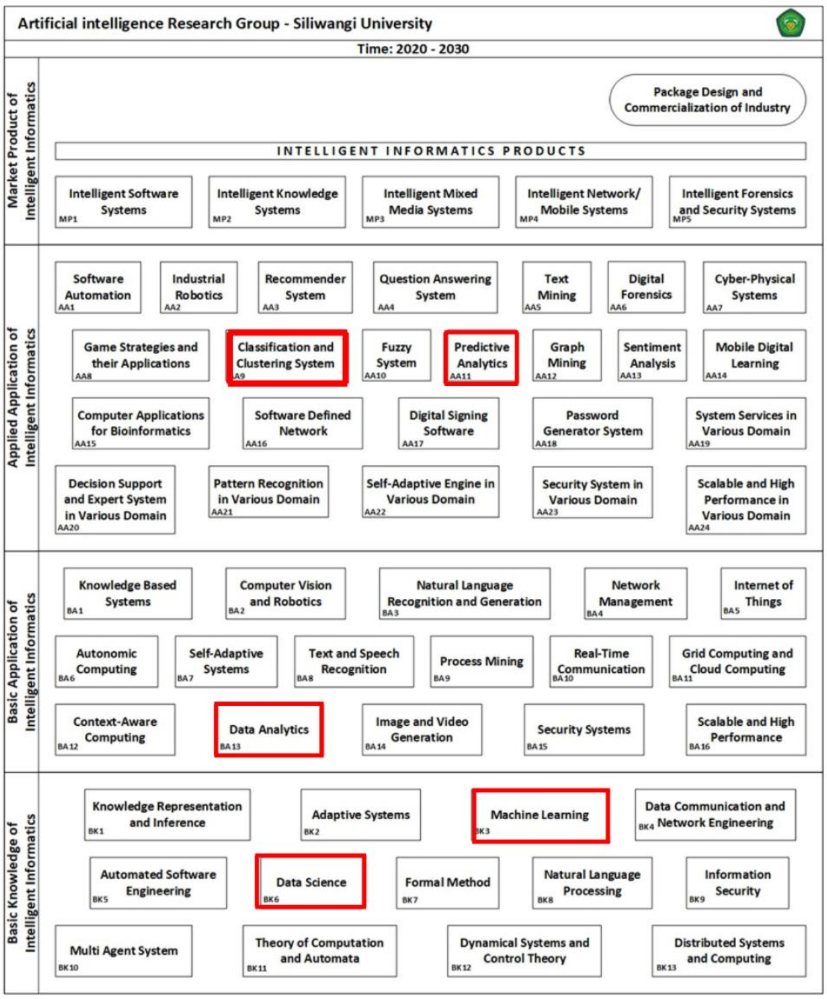
BAB III

METODOLOGI PENELITIAN

3.1 Peta Jalan Penelitian

Peta jalan penelitian pada Gambar 3. *1* merujuk pada *Roadmap Artificial Intelligence Research Group* Universitas Siliwangi Tahun 2020–2030. *Roadmap* tersebut menggambarkan arah pengembangan riset kecerdasan buatan secara sistematis.



Gambar 3. 1 *Roadmap* dan Kontribusi terhadap Penelitian AIS

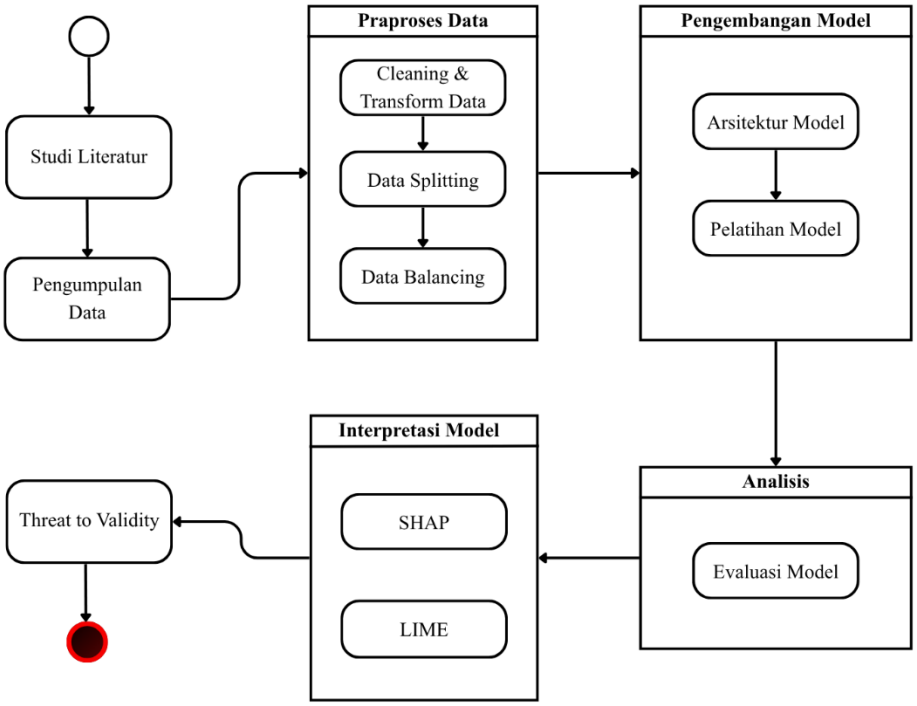
Sumber: (AIS Universitas Siliwangi, 2024)

Berdasarkan pemetaan tersebut, penelitian ini berada pada domain *Basic Knowledge* khususnya pada area *Machine Learning* dan *Data Science*, karena penelitian berfokus pada pengembangan serta evaluasi model *supervised learning* menggunakan algoritma CatBoost untuk klasifikasi multi-kelas performa akademik. Selain itu, penelitian ini juga termasuk dalam ranah *Basic Application* pada area *Data Analytics*, karena melibatkan proses eksplorasi, pemodelan, evaluasi, serta interpretasi pola data akademik mahasiswa menggunakan pendekatan analitik berbasis model.

Penelitian ini berkaitan dengan *Classification System* dan *Predictive Analytics* pada level *Applied Application*, karena model yang dikembangkan berfungsi untuk memetakan atribut mahasiswa ke dalam kategori performa akademik tertentu serta menghasilkan prediksi yang dapat digunakan sebagai dasar pengambilan keputusan akademik. Penelitian ini berkontribusi pada penguatan fondasi metodologis dan implementasi analitik prediktif dalam bidang pendidikan berbasis kecerdasan buatan.

3.2 Tahapan Penelitian

Penelitian ini bertujuan untuk membangun model yang mampu memprediksi performa akademik mahasiswa secara multikelas serta menginterpretasikan hasil prediksi tersebut. Tahapan penelitian ini menggunakan *machine learning workflow* yang dimodifikasi dari sisi pengembangan model (Johora dkk., 2025). Rangkaian tahapan penelitian yang akan dilakukan ditunjukkan pada Gambar 3. 2.

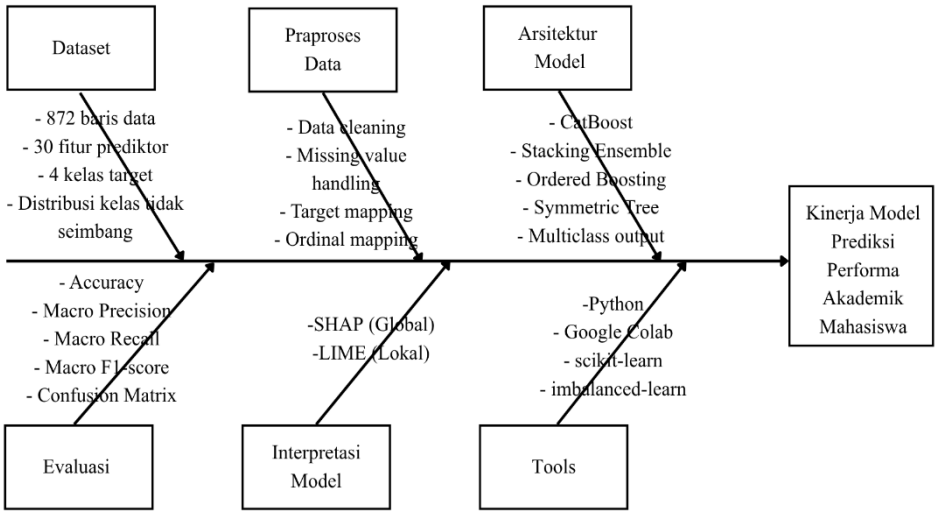


Gambar 3. 2 Tahapan Penelitian

Studi literatur mengawali tahap awal penelitian untuk membangun landasan teori yang kuat. Pengumpulan data menghasilkan dataset yang dibutuhkan dalam penelitian. Tahap praproses data memuat kegiatan *cleaning* dan *transform data* pada urutan pertama untuk meningkatkan kualitas data. *Data splitting* membagi dataset menjadi data latih dan data uji pada langkah berikutnya. Proses *data balancing* menyelesaikan fase praproses ini untuk menangani ketidakseimbangan distribusi kelas pada dataset.

Perancangan arsitektur model memulai tahap pengembangan model setelah fase praproses selesai. Pelatihan model menggunakan data yang telah dipersiapkan sebelumnya untuk membentuk kemampuan prediksi. Evaluasi model mengisi tahap analisis guna mengukur performa dari hasil latih secara komprehensif. Metode

SHAP dan LIME mendukung tahap interpretasi model untuk menjelaskan transparansi keputusan prediksi. Analisis *threat to validity* mengakhiri rangkaian penelitian ini guna mengidentifikasi batasan serta potensi ancaman terhadap validitas hasil akhir.



Gambar 3. 3 Fishbone Diagram

Komponen-komponen penting pada penelitian ini dapat dilihat pada Gambar 3. 3 sebagai diagram *fishbone*. Diagram *fishbone* menggambarkan kerangka sistematis yang menghubungkan faktor utama dan faktor turunan untuk mencapai target penelitian (Johora dkk., 2025). Komponen utama dalam penelitian ini terdiri atas dataset, praproses, arsitektur CatBoost, evaluasi, interpretasi model dan tools penelitian. Dataset yang digunakan terdiri atas 872 baris data dengan 30 fitur prediktor dan 4 kelas target, yang selanjutnya melalui tahapan praproses meliputi *Ordinal Label Encoding* serta SMOTEN untuk menangani ketidakseimbangan kelas. Model CatBoost dikonfigurasi menggunakan berbagai parameter yang dioptimalkan melalui parameter *gridsearch* guna memperoleh

kombinasi *hyperparameter* terbaik. Performa model kemudian dievaluasi menggunakan beberapa metrik, yaitu *accuracy, precision, recall, F1-Score* dan *confusion matrix,* lalu diinterpretasikan secara global menggunakan SHAP dan secara lokal menggunakan LIME untuk memastikan transparansi hasil prediksi. Seluruh proses penelitian dijalankan menggunakan Python dan Google Colab dengan dukungan berbagai library *machine learning* yang terintegrasi dalam satu lingkungan kerja yang efisien (Bentéjac dkk., 2021).

3.3 Studi Literatur

Tahapan ini merupakan proses untuk mengumpulkan, membaca, dan menganalisis literatur ilmiah yang relevan dengan topik penelitian. Tujuan utamanya adalah untuk memahami landasan konsep teoritis menelusuri penelitian terdahulu, dan mengidentifikasi *gap research* yang akan menjadi dasar perumusan penelitian ini. Gambaran penelitian dapat dipahami secara menyeluruh terhadap ruang lingkup permasalahan dan menetapkan arah metodologi yang tepat.

3.4 Pengumpulan Data

Data yang digunakan pada penelitian ini adalah dataset yang sama dengan penelitian rujukan Johora dkk. (2025), yaitu dataset survei performa akademik mahasiswa sarjana dari berbagai universitas di Bangladesh yang tersedia secara publik melalui platform GitHub. Dataset terdiri atas 872 baris data dengan 32 kolom, di mana kolom Department dihapus sehingga tersisa 30 fitur prediktor dan 1 kolom target. Kolom target berisi nilai CGPA/GPA semester pertama dalam format rentang *string* yang akan dipetakan ke dalam 4 kelas. Seluruh 30 fitur bersifat kategorikal (*string*), yang merepresentasikan faktor akademik, sosial-

ekonomi, dan perilaku mahasiswa.

3.5 Praproses Data

Praproses data adalah tahapan yang dilakukan untuk menyiapkan dataset sebelum masuk ke proses pelatihan dan evaluasi model. Tahapan ini dilakukan melalui beberapa langkah yaitu *data cleaning, data transformation, data splitting* dan *data balancing*.

3.5.1 *Data Cleaning*

Tahap *data cleaning* dilakukan untuk memastikan data yang digunakan memiliki struktur dan nilai yang konsisten sebelum memasuki proses transformasi. Dataset terlebih dahulu dimuat dari berkas CSV, kemudian atribut Department dihapus karena tidak digunakan sebagai variabel prediktor dalam penelitian. Selanjutnya, nama-nama kolom disederhanakan agar lebih mudah diidentifikasi dan digunakan pada proses pengolahan data. Pembersihan juga dilakukan pada seluruh atribut bertipe teks dengan menghapus spasi pada awal dan akhir nilai, serta menyeragamkan nilai kosong seperti *string* kosong, nan, dan *None* sebagai *missing value.* Proses pemeriksaan dilakukan pada atribut target untuk memastikan tidak terdapat nilai yang hilang maupun kategori yang tidak sesuai dengan ketentuan pemetaan kelas, sehingga seluruh data target dapat diproses secara konsisten pada tahap berikutnya (Johora dkk., 2025).

3.5.2 *Data Transformation*

Tahapan klasifikasi variabel merupakan langkah krusial yang dilakukan sebelum proses *encoding*. Seluruh 30 fitur diklasifikasikan berdasarkan teori skala pengukuran Stevens (1946) menjadi dua kelompok, yaitu fitur *ordinal* (memiliki

urutan logis yang bermakna) dan fitur nominal (hanya label tanpa hierarki). Klasifikasi ini menentukan metode *encoding* yang akan diterapkan. Keseluruhan fitur ordinal dipetakan menggunakan *Manual Ordinal Mapping* dengan angka berurutan. Fitur nominal tidak ditransformasikan menggunakan *LabelEncoder*, melainkan dipertahankan dalam bentuk kategorikal agar dapat diproses melalui mekanisme *native categorical handling* pada CatBoost.

3.5.3 *Data Splitting*

Dataset dibagi menggunakan metode *Stratified 10-Fold Cross-Validation* dengan skema *Out-of-Fold* (OOF). Dataset dipartisi menjadi sepuluh *fold* dengan tetap mempertahankan proporsi keempat kelas pada setiap *fold*. Sembilan *fold* digunakan sebagai data latih dan satu *fold* digunakan sebagai data validasi secara bergantian pada setiap iterasi, sehingga setiap observasi memperoleh satu prediksi dari model yang tidak dilatih menggunakan observasi tersebut. Proses penyeimbangan data hanya diterapkan pada data latih di setiap *fold*, sedangkan data validasi dipertahankan sesuai dengan distribusi aslinya untuk mencegah kebocoran informasi. Seluruh prediksi dari data validasi kemudian digabungkan menjadi prediksi OOF dan digunakan untuk menghitung performa model secara keseluruhan. Penggunaan stratifikasi bertujuan menjaga keterwakilan setiap kelas pada seluruh *fold*, khususnya pada dataset dengan distribusi kelas yang tidak seimbang, sedangkan *cross-validation* digunakan agar estimasi kemampuan generalisasi model tidak bergantung pada satu pembagian data saja (Szeghalmy & Fazekas, 2023).

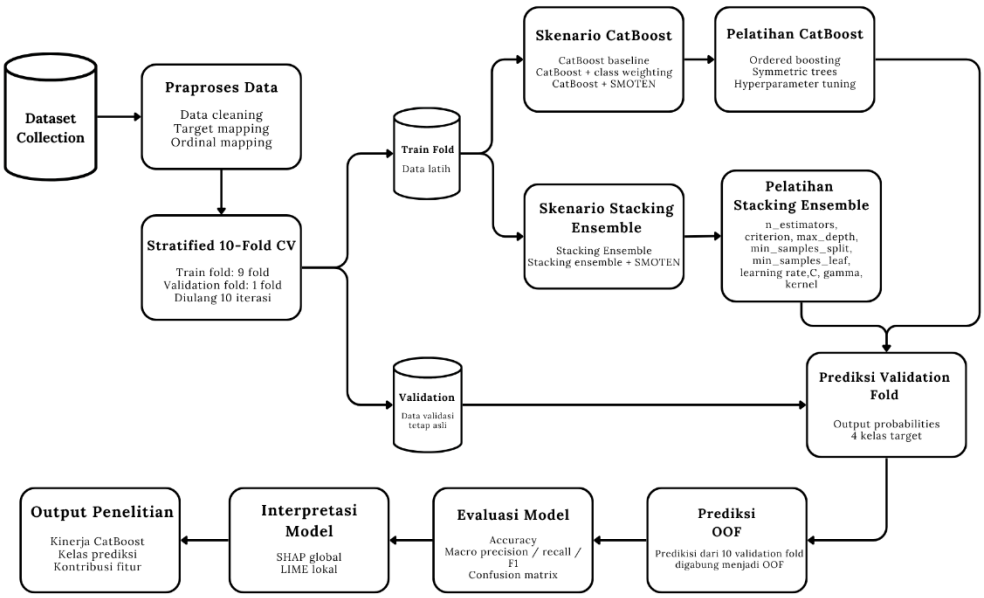
3.5.4 *Data Balancing*

Permasalahan pada dataset penelitian ini adalah distribusi kelas target yang tidak seimbang, yaitu jumlah sampel pada kelas mayoritas lebih banyak dibandingkan kelas minoritas. Kondisi tersebut berpotensi menyebabkan model lebih dominan mempelajari pola kelas mayoritas sehingga kemampuan dalam mengenali pola pada kelas minoritas menjadi kurang optimal. Penelitian ini menerapkan *Synthetic Minority Over-sampling Technique for Nominal* (SMOTEN), untuk mengatasi permasalahan tersebut. Metode ini membentuk sampel sintetis pada kelas minoritas berdasarkan kemiripan kategori dari tetangga terdekat, sehingga sesuai digunakan pada dataset yang didominasi fitur kategorikal. Proses penyeimbangan hanya diterapkan pada data latih di setiap *fold*, sedangkan data validasi tetap mempertahankan distribusi aslinya untuk mencegah kebocoran informasi. Setelah proses SMOTEN, distribusi kelas pada data latih menjadi lebih seimbang sehingga model memperoleh kesempatan yang lebih proporsional untuk mempelajari pola dari setiap kategori performa akademik (García-Vicente dkk., 2023).

3.6 Pengembangan Model

Pengembangan model pada penelitian ini dilakukan untuk mengevaluasi kinerja model prediksi performa akademik mahasiswa melalui beberapa skenario eksperimen. Model utama yang digunakan adalah CatBoost sedangkan *stacking ensemble* digunakan sebagai model pembanding berdasarkan penelitian terkait. Pengembangan model tidak hanya berfokus pada proses implementasi algoritma, tetapi juga pada pengujian pengaruh penanganan ketidakseimbangan kelas terhadap

kinerja model. Tahap pengembangan model terdiri atas dua bagian utama, yaitu arsitektur model dan pelatihan model. Arsitektur model menjelaskan struktur algoritma yang digunakan dalam penelitian sedangkan pelatihan model menjelaskan mekanisme *training*, validasi, penerapan SMOTEN, serta pembentukan prediksi *out-of-fold.*



Gambar 3. 4 *Pipeline* Penelitian

Gambar 3. 4 menunjukkan alur tahapan penelitian untuk model utama CatBoost dan model pembanding *stacking ensemble*. Tahap pertama mengumpulkan *dataset* awal untuk memulai rangkaian penelitian. Proses praproses data menjalankan kegiatan *data cleaning*, *target mapping*, dan *ordinal mapping* guna meningkatkan kualitas data. *Stratified 10-Fold CV* membagi *dataset* menjadi *train fold* sebanyak 9 *fold* dan *validation fold* sebanyak 1 *fold* dengan pengulangan sebanyak 10 iterasi. *Train fold* menyediakan data latih untuk proses pemodelan selanjutnya. Data validasi mempertahankan distribusi data asli pada *validation fold*

tanpa mengalami manipulasi.

Skenario CatBoost membagi pengujian model utama menjadi beberapa variasi yang meliputi CatBoost *baseline*, CatBoost + *class weighting*, dan CatBoost + SMOTEN. Pelatihan CatBoost memproses data latih tersebut menggunakan karakteristik *ordered boosting*, *symmetric trees*, dan *hyperparameter tuning*. Skenario *Stacking Ensemble* mengelompokkan variasi pengujian model pembanding ke dalam opsi *stacking ensemble* standar serta *stacking ensemble* + SMOTEN. Pelatihan *Stacking Ensemble* mengoptimalkan model pembanding melalui penyetelan parameter *n_estimators*, *criterion*, *max_depth*, *min_samples_split*, *min_samples_leaf*, *learning rate*, C, *gamma*, dan *kernel*.

Prediksi *validation fold* menghasilkan probabilitas keluaran untuk 4 kelas target berdasarkan model hasil latih dan data validasi asli. Prediksi OOF menggabungkan seluruh hasil prediksi dari 10 *validation fold* menjadi satu kesatuan data *Out-of-Fold*. Tahap evaluasi model mengukur kinerja prediksi menggunakan metrik *accuracy*, *macro precision*, *macro recall*, *macro F1-score*, dan *confusion matrix*. Tahap interpretasi model menerapkan metode SHAP global dan LIME lokal untuk menganalisis transparansi keputusan model. Tahap akhir menghasilkan *output* penelitian berupa visualisasi tingkat kinerja CatBoost, pemetaan kelas prediksi, beserta analisis kontribusi fitur secara komprehensif.

3.6.1 Arsitektur Model

CatBoost (Prokhorenkova dkk., 2018) memiliki dua keunggulan struktural yang menjadi dasar diimplementasikan sebagai model utama dalam penelitian ini. *Ordered Boosting* membangun setiap pohon keputusan berdasarkan permutasi acak

data secara berurutan, prediksi pada setiap sampel dihitung tanpa menggunakan informasi dari sampel tersebut dan menghasilkan model yang lebih tahan terhadap *overfitting*, terutama pada dataset berukuran kecil hingga menengah. *Symmetric Trees* membangun pohon secara simetris dengan *split* yang sama di setiap level, sehingga menghasilkan proses prediksi yang lebih cepat dan memiliki ketahanan terhadap data yang tidak relevan.

Penelitian ini juga menggunakan *stacking ensemble* sebagai model pembanding. *Stacking ensemble* merupakan metode *ensemble learning* yang menggabungkan beberapa model dasar atau *base learner*, kemudian menggunakan model *meta learner* untuk menghasilkan prediksi akhir. Arsitektur *stacking ensemble* terdiri atas beberapa *base learner*, yaitu *Random Forest, Gradient Boosting*, dan *Support Vector Classifier.* Prediksi dari masing-masing *base learner* kemudian digunakan sebagai masukan bagi meta learner untuk menentukan kelas akhir.

3.6.2 Pelatihan Model

Pelatihan model dilakukan menggunakan skema *Stratified 10-Fold Cross-Validation*. Skema ini membagi dataset menjadi sepuluh bagian atau *fold* dengan tetap mempertahankan proporsi kelas pada setiap *fold*. Sembilan *fold* digunakan sebagai data latih, sedangkan satu *fold* digunakan sebagai data validasi. Proses ini dilakukan sebanyak sepuluh kali hingga setiap *fold* pernah menjadi data validasi. Penggunaan *Stratified 10-Fold Cross-Validation* bertujuan untuk memperoleh evaluasi model yang lebih stabil, terutama karena dataset yang digunakan memiliki distribusi kelas yang tidak seimbang. Setiap *fold* tetap merepresentasikan komposisi

kelas yang relatif seimbang sesuai distribusi data awal menggunakan stratifikasi agar evaluasi model tidak hanya dipengaruhi oleh pembagian data tertentu.

Proses *resampling* hanya diterapkan pada data latih di setiap *fold*. Data validasi tidak dikenai proses SMOTEN agar tetap merepresentasikan distribusi asli dataset. Prosedur ini dilakukan untuk menghindari information *leakage*, yaitu kondisi ketika informasi dari data validasi secara tidak langsung ikut memengaruhi proses pelatihan model. Dengan demikian, model diuji pada data validasi yang belum pernah digunakan dalam proses pembentukan sampel sintetis. Prediksi yang dihasilkan pada setiap validation *fold* dikumpulkan menjadi *out-of-fold prediction*. *Out-of-fold prediction* merupakan prediksi terhadap seluruh data, di mana setiap data diprediksi oleh model yang tidak dilatih menggunakan data tersebut.

Tabel 3. 1 Skenario Replikasi Model

Skenario	Deskripsi	Tujuan
Skenario 1	*Stacking Ensemble* + *balancing* sebelum pemisahan data	Mereplikasi *pipeline* dan hasil penelitian acuan
Skenario 2	*Stacking Ensemble* + *balancing* setelah pemisahan data	Memperbaiki *pipeline* penelitian acuan dan mengurangi risiko *information leakage*

Eksperimen model pada penelitian ini dibagi menjadi dua skenario utama. Skenario pertama berfokus pada replikasi dan perbaikan *pipeline* penelitian acuan yang ditunjukkan pada Tabel 3. 1. Replikasi dilakukan dengan menerapkan *balancing data* sebelum proses pemisahan data, sesuai dengan tahapan yang digunakan pada penelitian sebelumnya. Selanjutnya, *pipeline* diperbaiki dengan menerapkan *balancing data* hanya pada data latih. Perbandingan kedua konfigurasi

tersebut dilakukan untuk mengetahui pengaruh posisi data *balancing* terhadap hasil evaluasi *stacking ensemble* serta mengidentifikasi kemungkinan estimasi performa yang terlalu optimistis akibat penerapan *balancing* sebelum pemisahan data.

Tabel 3. 2 Skenario Eksperimen Model

Skenario	Deskripsi	Tujuan
Skenario 1	*Stacking Ensemble*	Mengetahui kemampuan dasar *Stacking Ensemble* pada kondisi data tidak seimbang
Skenario 2	*Stacking Ensemble* + *SMOTEN*	Mengetahui pengaruh penyeimbangan data terhadap performa *Stacking Ensemble*
Skenario 3	CatBoost *baseline*	Mengetahui kemampuan dasar CatBoost pada kondisi data tidak seimbang
Skenario 4	CatBoost + *class weighting*	Mengetahui pengaruh pembobotan kelas terhadap kemampuan model mengenali kelas minoritas
Skenario 5	CatBoost + SMOTEN	Mengetahui pengaruh penyeimbangan data terhadap performa CatBoost

Skenario kedua berfokus pada evaluasi CatBoost melalui tiga konfigurasi, yaitu CatBoost *baseline*, CatBoost dengan *class weighting*, dan CatBoost dengan SMOTEN yang ditunjukkan pada

Eksperimen model pada penelitian ini dibagi menjadi dua skenario utama. Skenario pertama berfokus pada replikasi dan perbaikan *pipeline* penelitian acuan yang ditunjukkan pada Tabel 3. 1. Replikasi dilakukan dengan menerapkan *balancing data* sebelum proses pemisahan data, sesuai dengan tahapan yang

digunakan pada penelitian sebelumnya. Selanjutnya, *pipeline* diperbaiki dengan menerapkan *balancing data* hanya pada data latih. Perbandingan kedua konfigurasi tersebut dilakukan untuk mengetahui pengaruh posisi data *balancing* terhadap hasil evaluasi *stacking ensemble* serta mengidentifikasi kemungkinan estimasi performa yang terlalu optimistis akibat penerapan *balancing* sebelum pemisahan data.

Tabel 3. *2* Seluruh konfigurasi tersebut menggunakan *pipeline* yang telah diperbaiki melalui *Stratified 10-Fold Cross-Validation*, dengan penanganan ketidakseimbangan kelas hanya diterapkan pada proses pelatihan.

Terdapat beberapa penerapan training configuration dalam pelatihan model Catboost yang digunakan pada penelitian ini ditunjukkan pada Tabel 3. 3

Tabel 3. 3 Konfigurasi CatBoost

Parameter	Keterangan
boosting_type	jenis *boosting* yang digunakan (Ordered , Plain)
iterations	Jumlah pohon maksimum, *early stopping* akan berhenti lebih awal jika tidak ada peningkatan
learning_rate	Step *size* gradien yang konservatif untuk mengurangi *overfitting*
depth	Kedalaman pohon; memberikan kapasitas ekspresi cukup untuk 4 kelas
l2_leaf_reg	Regularisasi L2 pada *leaf node* untuk mencegah *overfitting*
auto_class_weights	Memberikan bobot lebih tinggi ke kelas minoritas *complementary* dengan SMOTEN
eval_metric	Optimasi langsung terhadap metrik target utama
random_seed	*Seed* tetap untuk *reproducibility* hasil penelitian

Penelitian ini juga menggunakan *stacking ensemble* sebagai model pembanding. *Stacking ensemble* terdiri atas beberapa *base learner,* yaitu *Random Forest, Gradient Boosting,* serta *Support Vector Classifier* sebagai *meta learner.* Setiap *base learner* dilatih untuk menghasilkan prediksi awal, kemudian hasil prediksi tersebut digunakan oleh *meta learner* untuk menghasilkan prediksi akhir. Konfigurasi *stacking ensemble* yang digunakan pada penelitian ini ditunjukkan pada Tabel 3. *4*.

Tabel 3. 4 Konfigurasi Stacking Ensemble

Model	Parameter	Keterangan
Random Forest	n_estimators	Jumlah pohon keputusan yang dibangun dalam ensemble.
	criterion	Kriteria untuk mengukur kualitas pemisahan node.
	max_depth	Kedalaman maksimum setiap pohon.
	min_samples_split	Jumlah minimum sampel yang dibutuhkan untuk membagi node.
	min_samples_leaf	Jumlah minimum sampel pada leaf node.
Gradient Boosting	n_estimators	Jumlah estimator atau pohon yang dibangun secara bertahap.
	learning_rate	Besar kontribusi setiap estimator terhadap model akhir.
	max_depth	Kedalaman maksimum pohon pada setiap estimator.
SVC	C	Parameter regularisasi untuk mengontrol margin dan kesalahan klasifikasi.
	gamma	Parameter yang mengontrol pengaruh satu sampel latih terhadap batas keputusan.
	kernel	Jenis kernel yang digunakan untuk membentuk batas keputusan.

3.7 Analisis Model

Hasil model akan dievaluasi menggunakan berbagai metrik, seperti *accuracy*, *precision, recall, F1-Score* dan *confusion matrix* setelah melakukan *training data*. *Accuracy* memiliki kelebihan dengan memberikan penjelasan yang mudah dimengerti tentang kinerja keseluruhan model. *Precision* digunakan untuk mengevaluasi ketepatan dalam memprediksi kelas positif, sehingga ideal digunakan dalam situasi dengan kesalahan positif yang harus diminimalkan. *Recall* unggul dalam menilai sejauh mana model dapat mengidentifikasi semua data positif. F1-score menggabungkan *precision* dan *recall* menjadi satu nilai yang seimbang menilai kinerja model ketika data memiliki distribusi kelas yang tidak seimbang (Sokolova & Lapalme, 2009). Evaluasi model dilakukan untuk menganalisis model

yang sudah dilatih, mengetahui pengaruh dari SMOTEN, dan membandingkan performa model dengan penelitian sebelumnya.

3.8 Interpretasi Model

Selain mengevaluasi performa prediktif, penelitian ini juga menganalisis interpretabilitas model menggunakan pendekatan *Explainable Artificial Intelligence* (XAI). Interpretabilitas penting agar hasil prediksi dapat digunakan sebagai dasar pengambilan keputusan akademik. Dua metode interpretasi digunakan dalam penelitian ini, yaitu SHAP dan LIME. SHAP digunakan untuk menghasilkan penjelasan global mengenai kontribusi setiap fitur terhadap prediksi model. LIME digunakan untuk menghasilkan penjelasan lokal yang menjelaskan alasan di balik prediksi model terhadap satu mahasiswa secara spesifik (Salih dkk., 2023).

3.9 Threat to Validity

Langkah selanjutnya adalah melakukan analisis validasi terhadap hasil eksperimen yang telah dilakukan. Analisis bertujuan untuk memahami dan mengidentifikasi hasil yang sudah dilakukan dari pengumpulan data sampai pengembangan model. Analisis validasi dilakukan dengan menggunakan *threat to validity. Threats to validity* merupakan elemen-elemen yang dapat mempengaruhi ketepatan atau keakuratan berdasarkan hasil penelitian terdiri dari *internal validity, external validity, construct validity, conclusion validity* (Matthay & Glymour, 2020).