BAB III

METODOLOGI PENELITIAN

3.1 Gambaran Umum Objek Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental dengan memanfaatkan teknik *machine learning* untuk membangun model klasifikasi status kelulusan mahasiswa. Tujuan utamanya adalah menghasilkan model klasifikasi yang mampu mengelompokkan mahasiswa ke dalam empat kategori hasil studi, yaitu Lulus Lebih Awal, Lulus Tepat Waktu, Lulus Tidak Tepat Waktu, dan *Drop Out*. Model ini dikembangkan agar dapat mendukung proses pengambilan keputusan akademik yang bersifat preventif melalui sistem peringatan dini berbasis data.

Objek penelitian adalah mahasiswa Program Studi Sistem Informasi Universitas Multimedia Nusantara angkatan 2020–2024. Data yang digunakan merupakan data akademik historis yang mencakup nilai per mata kuliah, indeks prestasi semester, jumlah pengulangan mata kuliah, total SKS yang ditempuh, serta atribut administratif lainnya yang menggambarkan perkembangan studi mahasiswa. Seluruh data telah dianonimkan dengan menghilangkan informasi identitas seperti nama mahasiswa, NIM, dan informasi pribadi lain yang bersifat sensitif. Dengan demikian, data yang dianalisis hanya berisi variabel akademik, demografis dasar, serta fitur turunan hasil rekayasa data.

Dalam penelitian ini dibangun tiga model utama yang mewakili tiga titik evaluasi akademik, yaitu model untuk Semester 2, Semester 4, dan Semester 6. Setiap model dikembangkan secara terpisah karena karakteristik data yang tersedia pada masing-masing semester berbeda, baik dari sisi jumlah fitur, struktur informasi, maupun kestabilan performa akademik mahasiswa. Masing-masing model memiliki versi-versi yang digunakan untuk eksplorasi dan pengujian pendekatan terbaik. Pada Semester 2 terdapat satu versi model, pada Semester 4 terdapat lima versi model, sedangkan pada Semester 6 terdapat empat versi model. Jumlah versi ini diperlukan untuk mengevaluasi kombinasi metode penyeimbangan data, konfigurasi CatBoost, dan teknik optimasi *hyperparameter* agar dapat

ditemukan konfigurasi yang paling sesuai untuk *dataset* berukuran kecil dan memiliki distribusi kelas yang tidak seimbang.

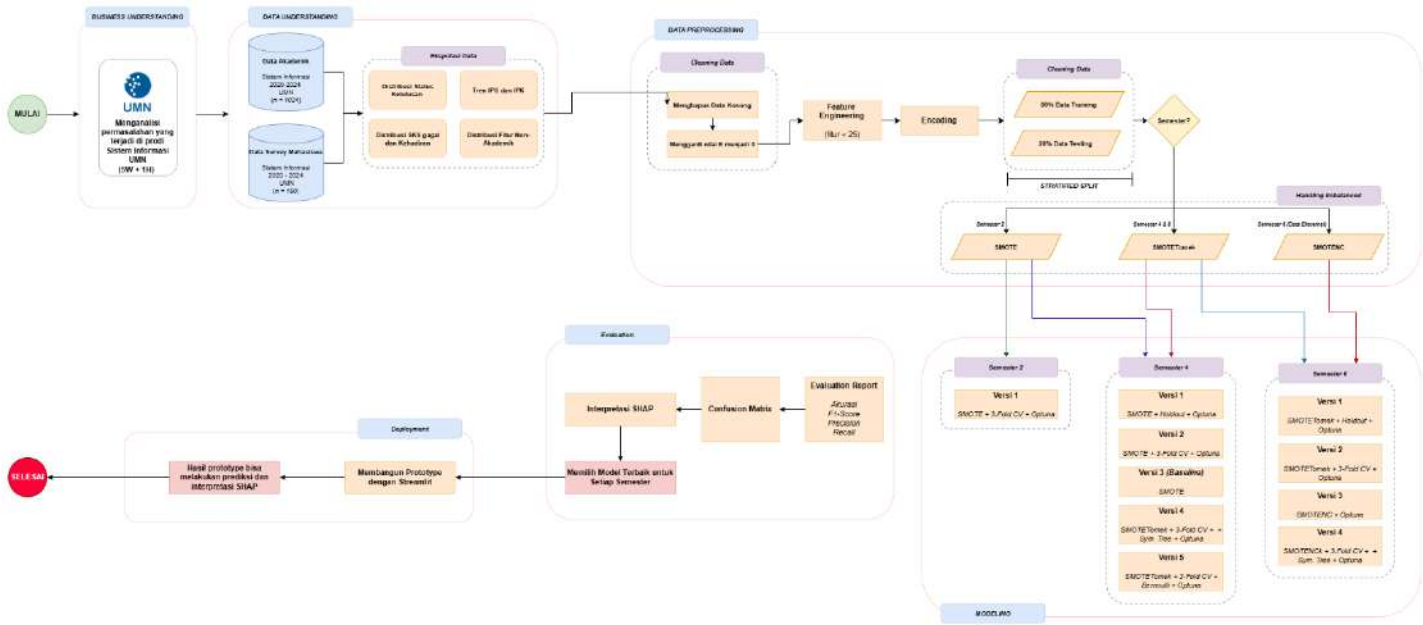
Metode utama yang digunakan dalam penelitian ini berfokus pada pembangunan dan pengujian model CatBoost sebagai model dasar (*baseline model*) yang kemudian dioptimasi agar sesuai untuk *dataset* berukuran kecil dan memiliki distribusi kelas yang tidak seimbang. Model CatBoost dipilih sebagai algoritma utama karena mampu menangani fitur kategorikal tanpa memerlukan proses *one-hot encoding* serta memiliki mekanisme *ordered boosting* yang efektif dalam mengurangi masalah *overfitting* pada *dataset* yang terbatas [74]. Setiap versi model dibangun dengan landasan parameter awal tertentu, seperti *depth*, jumlah iterasi, *learning rate*, serta pengaturan regularisasi. Keputusan membuat beberapa versi didasarkan pada kebutuhan untuk menguji pengaruh perubahan konfigurasi terhadap kinerja model, terutama mengingat *dataset* yang digunakan berskala kecil dan memiliki ketidakseimbangan kelas yang cukup tinggi. Untuk menangani ketidakseimbangan data, digunakan teknik *oversampling* sintetis seperti SMOTE serta kombinasi SMOTETomek. Proses optimasi model dilakukan melalui penyesuaian *hyperparameter* menggunakan metode *Bayesian optimization* dengan bantuan pustaka Optuna. Selain fokus pada peningkatan kinerja model, penelitian ini juga memperhatikan aspek interpretabilitas melalui penerapan metode *Explainabl*e AI (XAI) dengan SHAP (*SHapley Additive exPlanations*) sebagai alat analisis kontribusi fitur terhadap hasil prediksi.

Rangkaian penelitian dirancang berdasarkan kerangka kerja CRISP-DM (*Cross Industry Standard Process for Data Mining*) yang menyediakan struktur metodologis terstandar untuk proyek analitik berbasis data. Kerangka ini dipilih karena bersifat iteratif, fleksibel, dan komprehensif, sehingga sesuai untuk mengelola seluruh tahapan penelitian dari pemahaman permasalahan hingga evaluasi hasil model.

3.2 Tahapan Penelitian

Kerangka kerja yang menjadi dasar tahapan penelitian ini adalah CRISP-DM (*Cross Industry Standard Process for Data Mining*). CRISP-DM membagi

kegiatan proyek *data mining* menjadi enam fase yang saling berhubungan, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*. Pendekatan ini dipilih karena memberikan struktur yang sistematis sekaligus bersifat iteratif, sehingga setiap fase dapat ditinjau ulang berdasarkan temuan pada fase berikutnya. Dalam penelitian ini setiap fase diimplementasikan sebagai rangkaian kegiatan yang dapat direproduksi, terdokumentasi, dan dievaluasi secara kuantitatif agar hasilnya dapat dipertanggungjawabkan dari sisi metodologi maupun aplikasinya bagi program studi yang dapat dilihat pada Gambar 31.



Gambar 3.1 CRISP-DM EWS

3.2.1 Business Understanding

Tahap ini dimulai dengan identifikasi masalah nyata yang dialami di lingkungan akademik program studi Sistem Informasi Universitas Multimedia Nusantara. Permasalahan utama adalah praktik bimbingan akademik yang cenderung reaktif sehingga intervensi diberikan setelah masalah akademik terlihat jelas, misalnya nilai yang turun drastis atau mahasiswa melewati batas masa studi. Kondisi ini memicu tingkat keterlambatan kelulusan dan risiko *drop out* yang berdampak pada kualitas pembelajaran dan beban administrasi program studi. Selain itu, dampak transisi pembelajaran selama dan pasca

pandemi memperparah variabilitas pola akademik sehingga kebutuhan akan alat bantu yang mampu mendeteksi risiko sedini mungkin menjadi lebih mendesak.

Dalam rangka mengisi celah penelitian sebelumnya, fase *business understanding* juga memetakan *gap* yang belum terpenuhi pada studi-studi sebelumnya. Studi terdahulu banyak menggunakan fitur agregat sederhana atau model yang bersifat *black box* sehingga sulit ditindaklanjuti oleh dosen pembimbing. Beberapa penelitian yang melaporkan akurasi tinggi menunjukkan potensi *overfitting* karena metode *oversampling* yang kurang tepat atau tidak cukup menangani data kecil dan tidak seimbang [70]. Berdasarkan kajian tersebut, penelitian ini menempatkan fokus pada pengembangan rekayasa fitur dinamis dari transkrip mentah, penanganan ketidakseimbangan kelas yang lebih hati-hati, serta penerapan teknik interpretasi model yang menghasilkan penjelasan lokal dan global. Tujuan ini direpresentasikan sebagai kebutuhan fungsional dari sistem peringatan dini yang akan diusulkan.

Meski demikian, pada tahap *business understanding* juga ditetapkan batasan dan keterbatasan penelitian yang memengaruhi tujuan operasional. Penelitian dibatasi pada data internal mahasiswa Program Studi Sistem Informasi angkatan 2020 sampai 2024 yang telah dianonimkan, sehingga hasil yang diperoleh bersifat spesifik terhadap kurikulum dan aturan akademik institusi tersebut dan tidak langsung digeneralisasikan ke institusi lain tanpa validasi ulang. Fokus pemodelan dibatasi pada satu algoritma utama agar *depth* analisis interpretasi dapat lebih maksimal, sehingga studi komparatif lintas algoritma secara luas tidak dilakukan. Selain itu, tahap *deployment* atau implementasi penuh sistem peringatan dini berada di luar cakupan penelitian ini. Penelitian ini menghasilkan prototipe dan rekomendasi teknis yang dapat digunakan pada tahap pengembangan selanjutnya.

Berdasarkan identifikasi masalah, *gap* penelitian, dan keterbatasan tersebut, dirumuskan tujuan utama yang realistis dan terukur. Tujuan utama

adalah merancang dan mengoptimasi model klasifikasi yang andal untuk memprediksi status kelulusan mahasiswa pada tiga titik evaluasi yang ditetapkan, serta menyediakan penjelasan berbasis nilai kontribusi fitur agar keluaran model dapat diterjemahkan menjadi kebijakan intervensi akademik. Tujuan turunan mencakup pengembangan pipeline rekayasa fitur dinamis dari data transkrip, penerapan strategi penyeimbangan data yang sesuai, dan penerapan prosedur optimasi hyperparameter yang efisien. Semua tujuan ini diformulasikan untuk bekerja dalam batasan data dan ruang lingkup yang telah ditetapkan sehingga hasilnya praktis untuk diaplikasikan oleh program studi.

3.2.2 Data Understanding

Tahap *data understanding* berfungsi untuk memperoleh pemahaman menyeluruh mengenai struktur, karakteristik, serta kualitas data sebelum masuk ke proses pengolahan dan pemodelan. Pada penelitian ini terdapat empat sumber data utama yang digunakan, masing-masing memiliki peran berbeda dalam membangun fitur prediktif maupun label keluaran. Pemahaman awal terhadap setiap sumber data menjadi penting untuk memastikan langkah-langkah berikutnya dapat dilakukan secara tepat dan sistematis.

Sumber pertama adalah Biodata_mhs_SI_TA_2020-2024, yaitu data berisi informasi demografis dan atribut administratif mahasiswa Program Studi Sistem Informasi angkatan 2020 hingga 2024. Data ini diperoleh melalui unit Biro Administrasi Akademik (BIA) dengan proses penarikan langsung dari sistem basis data akademik kampus. Data tersebut mencakup atribut seperti tahun masuk, jenis kelamin, status mahasiswa, serta informasi identitas yang kemudian dianonimkan untuk menjaga kerahasiaan individu. Data ini tidak hanya berfungsi sebagai deskripsi awal populasi, tetapi juga digunakan untuk memetakan persebaran mahasiswa berdasarkan angkatan.

Sumber kedua adalah Transkrip_Mhs_SI_TA_2020-2024, yang juga diperoleh dari BIA melalui penarikan data langsung dari database akademik. Data transkrip ini berisi nilai per mata kuliah, jumlah SKS, informasi pengulangan mata kuliah, IPS per semester, serta IPK kumulatif

mahasiswa. Karakteristik data bersifat longitudinal, mencerminkan perkembangan performa akademik mahasiswa dari semester ke semester. Transkrip inilah yang menjadi inti dari rekayasa fitur yang digunakan dalam model prediksi kelulusan, karena setiap mata kuliah, pola IPS, serta frekuensi pengulangan memberikan indikator penting mengenai progres akademik seseorang.

Sumber ketiga adalah Data_responden, yaitu hasil survei mahasiswa yang dikumpulkan untuk memetakan faktor non-akademik seperti dukungan keluarga, kondisi finansial, kesesuaian minat dengan jurusan, dan kepuasan terhadap program studi. Data ini disusun melalui survei mandiri dan menjadi pelengkap dalam tahap eksplorasi, meskipun tidak digunakan secara langsung sebagai fitur dalam model prediksi. Namun, keberadaannya memberikan gambaran kontekstual mengenai situasi sosial-psikologis mahasiswa yang dapat memengaruhi performa akademik.

Sumber keempat adalah Transkrip_Berlabel, yaitu dataset hasil pengolahan melalui Python yang menggabungkan transkrip mentah dengan label status kelulusan akhir mahasiswa. Proses penggabungan dilakukan melalui pemetaan data transkrip terhadap catatan akademik kelulusan sehingga menghasilkan satu set data final dengan struktur siap model. Pada dataset ini informasi nilai sudah dinormalisasi, beberapa fitur sudah direkayasa, dan setiap entri telah memiliki label kategorikal yang menjadi target prediksi dalam model *machine learning*.

Setelah seluruh sumber data diidentifikasi, tahap berikutnya adalah melakukan eksplorasi awal terhadap distribusi dan pola umum dalam data. Eksplorasi ini dilakukan melalui visualisasi untuk mengidentifikasi kecenderungan, pola ketidakseimbangan, dan karakteristik penting lainnya. Visualisasi pertama yang dilakukan adalah Distribusi Jumlah Mahasiswa Berdasarkan Angkatan (2020–2024) yang berguna untuk melihat persebaran jumlah mahasiswa per tahun masuk. Hal ini membantu memahami dominasi atau kekurangan sampel di angkatan tertentu yang dapat memengaruhi

distribusi kelas kelulusan. Selanjutnya dilakukan visualisasi Rata-rata IPK Mahasiswa per Angkatan, yang menggambarkan tingkat capaian akademik rata-rata di masing-masing angkatan dan memberikan gambaran awal mengenai variasi performa antar *cohort*.

Analisis visual kemudian diperluas ke tingkat semester melalui Tren Rata-Rata IPS Seluruh Mahasiswa Berdasarkan Semester, sehingga memberikan gambaran dinamika performa akademik mahasiswa sepanjang masa studinya. Visualisasi ini membantu mendeteksi apakah terdapat semester tertentu yang memiliki tingkat kesulitan lebih tinggi atau pola penurunan performa yang konsisten.

Untuk memetakan karakteristik mata kuliah yang dominan, dilakukan visualisasi 10 Mata Kuliah dengan Jumlah Mahasiswa Terbanyak, yang menunjukkan mata kuliah dengan tingkat pengambilan paling tinggi. Selain itu terdapat grafik 10 Mata Kuliah dengan Rata-rata Nilai Terendah serta 10 Mata Kuliah Wajib dengan Rata-rata Nilai Terendah, yang bertujuan mengidentifikasi mata kuliah yang memiliki tingkat kesulitan relatif tinggi. Sebagai penyeimbang, dilakukan pula visualisasi 10 Mata Kuliah Wajib dengan Rata-rata Nilai Tertinggi, yang memberikan perspektif mengenai mata kuliah yang secara konsisten menghasilkan capaian akademik tinggi.

Pada level *outcome* studi, dilakukan visualisasi Distribusi Status Kelulusan Mahasiswa, yang menunjukkan proporsi empat kategori kelulusan dalam dataset. Untuk melihat variasi lebih spesifik per *cohort*, ditampilkan juga Proporsi Status Kelulusan Mahasiswa Angkatan 2020, 2021, 2022, dan 2023 secara terpisah, sehingga terlihat dengan jelas bagaimana pola kelulusan berbeda antar angkatan.

Selain data akademik, visualisasi juga dilakukan terhadap hasil survei mahasiswa. Grafik Distribusi Responden Berdasarkan Frekuensi Dukungan Keluarga dalam Hal Akademik digunakan untuk memahami sejauh mana dukungan keluarga berperan. Analisis diperkuat dengan visualisasi Dukungan Finansial Keluarga untuk Keperluan Kuliah, yang memetakan

stabilitas finansial mahasiswa. Selanjutnya dilakukan visualisasi Kesesuaian Jurusan dengan Keinginan Diri Sendiri, serta Distribusi Tingkat Kepuasan terhadap Program Studi, untuk melihat keterkaitan potensi motivasi internal dan kepuasan terhadap dinamika akademik.

Melalui seluruh rangkaian eksplorasi dan visualisasi tersebut, tahap data understanding memberikan gambaran komprehensif mengenai struktur dataset, distribusi variabel penting, karakteristik mata kuliah, serta pola kelulusan mahasiswa. Pemahaman ini menjadi dasar kuat untuk proses *data preparation* pada tahap berikutnya karena membantu mengidentifikasi fitur-fitur penting, potensi masalah seperti ketidakseimbangan kelas, serta strategi pembersihan dan rekayasa data yang diperlukan sebelum memasuki tahap pemodelan.

3.2.3 Data Preprocessing

Tahap *preprocessing* merupakan fondasi utama dalam penelitian ini karena seluruh proses pemodelan dan interpretasi model sangat bergantung pada bagaimana data dipersiapkan dan ditata pada tahap ini. Setiap sumber data yang berbeda memerlukan pendekatan penanganan yang berbeda pula, terutama karena karakteristik data akademik bersifat longitudinal, memiliki distribusi yang tidak merata antar angkatan, serta memuat kombinasi fitur kategorikal dan numerik yang kompleks. Oleh sebab itu, *preprocessing* dilakukan melalui serangkaian langkah sistematis yang mencakup pembersihan data, penanganan nilai hilang, pembentukan label, transformasi fitur kategorikal, normalisasi skala numerik, rekayasa fitur dinamis berdasarkan transkrip, seleksi fitur, pembagian dataset, hingga penyeimbangan kelas pada data pelatihan. Proses ini tidak hanya memastikan kualitas data yang tinggi, tetapi juga menjaga validitas analisis sehingga hasil pemodelan benar-benar menggambarkan kondisi akademik mahasiswa.

Seluruh langkah *preprocessing* dirancang untuk dapat direplikasi dan diaudit sepenuhnya. Untuk itu setiap transformasi yang diterapkan dicatat, parameter disimpan, dan dataset pada setiap tahap disimpan sebagai artefak

terpisah. Dengan cara ini, setiap percobaan yang dilakukan pada tahap modeling dapat dijalankan kembali pada kondisi yang sama, sehingga seluruh proses penelitian memenuhi standar reproducibility.

3.2.3.1 Penanganan Missing Values

Nilai hilang pada *dataset* ditangani secara hati-hati karena dapat berdampak signifikan terhadap perhitungan fitur akademik seperti IPS, jumlah SKS, dan IPK. Pada kolom numerik seperti IPS, sejumlah entri kosong ditemukan pada semester awal karena mahasiswa baru belum memiliki riwayat nilai lengkap atau sedang menyesuaikan beban studi. Ketika jumlah *missing* relatif kecil, nilai tersebut diisi menggunakan median sesuai angkatan agar menjaga distribusi asli. Misalnya, median IPS angkatan 2020 dan 2021 berada pada kisaran 3.10 hingga 3.25 sehingga nilai-nilai tersebut digunakan sebagai imputasi. Pendekatan median dipilih karena tidak dipengaruhi oleh *outlier* dan menjaga kestabilan distribusi akademik antar angkatan.

Untuk kolom kategorikal seperti nilai huruf, pemeriksaan terlebih dahulu memastikan apakah nilai hilang terjadi karena kesalahan input atau karena mahasiswa memang belum menyelesaikan mata kuliah tersebut. Jika proporsi nilai hilang di bawah 5%, imputasi dilakukan dengan modus nilai pada mata kuliah tersebut. Namun, nilai hilang terjadi pada lebih dari 30% data mata kuliah tertentu, baris tersebut cenderung dikeluarkan karena mata kuliah tersebut kemungkinan tidak ditawarkan secara konsisten atau memiliki data historis yang tidak stabil. Proses imputasi ini dilakukan hanya pada data pelatihan agar mencegah kebocoran informasi ke data validasi dan uji.

3.2.3.2 Pembentukkan Label dan Definisi Kategori Kelulusan

Label target penelitian dibentuk berdasarkan catatan kelulusan resmi dari fakultas. Pembentukan label dilakukan dengan cara menghitung durasi studi masing-masing mahasiswa dari tahun masuk hingga semester kelulusan terakhir. Mahasiswa yang lulus dalam tujuh semester atau kurang

diklasifikasikan sebagai Lulus Lebih Awal, mahasiswa yang lulus tepat pada semester kedelapan dikategorikan sebagai Lulus Tepat Waktu, mahasiswa yang menyelesaikan studi dalam sembilan hingga empat belas semester dikategorikan sebagai Lulus Tidak Tepat Waktu, dan mahasiswa yang tidak lagi tercatat aktif atau tidak memiliki catatan kelulusan digolongkan sebagai *Drop Out*. Proses *labeling* ini mengikuti aturan bisnis akademik yang berlaku di program studi dan memastikan bahwa perbedaan antar kelas bersifat terstandarisasi.

Setelah label dibentuk, distribusi kelas dianalisis dan ditemukan ketidakseimbangan yang signifikan. Sebagian besar mahasiswa berada pada kategori Lulus Tepat Waktu sedangkan kategori *Drop Out* hanya mencakup sebagian kecil mahasiswa. Ketidakseimbangan ini menjadi perhatian penting karena dapat menyebabkan model memprediksi kelas mayoritas secara berlebihan dan mengabaikan kelas minoritas yang justru paling penting untuk intervensi akademik. Oleh sebab itu, tahap penyeimbangan kelas disiapkan khusus pada data pelatihan.

3.2.3.3 Transformasi dan Pengolahan Fitur Kategorikal

Pengolahan fitur kategorikal dilakukan berdasarkan jumlah kategori dan relevansi semantiknya. Fitur seperti jenis kelamin dan angkatan memiliki sedikit kategori sehingga dikonversi menjadi nilai numerik sederhana. Transformasi ini diperlukan agar algoritma pembelajaran dapat memproses nilai tersebut tanpa mengubah maknanya. Untuk fitur *high-cardinality* seperti kode mata kuliah, jumlah kategorinya bisa mencapai lebih dari dua ratus sehingga menggunakan *one-hot encoding* akan menghasilkan dimensi fitur yang sangat besar dan tidak efisien krena CatBoost mendukung pengolahan kategori secara langsung melalui mekanisme *encoding internal*, fitur-fitur tersebut dipertahankan dalam bentuk kategorikal numerik.

Pendekatan ini tidak hanya meningkatkan efisiensi komputasi, tetapi juga menghindari masalah *sparsity* yang umum terjadi pada dataset

pendidikan dengan banyak variasi mata kuliah. Dengan cara ini, setiap mata kuliah tetap diwakili secara eksplisit tanpa membebani model dengan ratusan kolom *dummy*. Selain itu, transformasi kategori dilakukan tanpa menyentuh data validasi atau uji agar mencegah *data leakage*.

3.2.3.4 Rekayasa Fitur Dinamis dari Transkrip

Rekayasa fitur merupakan proses paling intensif yang dilakukan pada tahap *preprocessing*. Fitur-fitur dinamis dibentuk dari transkrip mentah untuk menangkap perkembangan akademik mahasiswa dari semester ke semester. Untuk keperluan penelitian, tiga snapshot akademik disiapkan, yaitu akhir semester 2, akhir semester 4, dan akhir semester 6. *Snapshot* ini dipilih karena mewakili titik-titik kritis di mana perubahan performa akademik mahasiswa sering terjadi dan dapat digunakan sebagai indikator dini untuk memprediksi risiko keterlambatan studi.

Pada setiap *snapshot*, dihitung berbagai fitur seperti *cumulative* GPA, total SKS lulus, rata-rata nilai mata kuliah wajib, jumlah mata kuliah yang digugurkan atau diulang, dan proporsi SKS lulus terhadap SKS yang diambil. Selain itu, tren IPS dihitung menggunakan regresi linier sederhana untuk mendeteksi apakah performa mahasiswa meningkat, menurun, atau stabil. Fitur lain seperti jumlah nilai rendah (nilai D dan E), mata kuliah yang diambil di luar semester rekomendasi, serta indikator jeda studi juga ditambahkan. Proses ini dilakukan dengan menjaga kausalitas temporal sehingga informasi dari semester setelah snapshot tidak memengaruhi snapshot sebelumnya. Dengan pendekatan ini, model dapat mempelajari pola perkembangan akademik mahasiswa secara natural sesuai waktu kejadian.

3.2.3.5 Pembagian Data

Setelah seluruh fitur selesai diproses dan dinyatakan layak untuk digunakan dalam pelatihan model, tahap berikutnya adalah membagi dataset historis menjadi dua subset utama, yaitu data pelatihan dan data uji. Pembagian ini dilakukan untuk memastikan bahwa proses pembelajaran

model dan pengujiannya berlangsung secara adil dan objektif, di mana model belajar sepenuhnya dari data pelatihan sementara evaluasi akhir dilakukan pada data uji yang benar-benar tidak pernah ditemukan model sebelumnya. Dalam penelitian ini, komposisi pembagian yang digunakan adalah 80 persen untuk pelatihan dan 20 persen untuk uji, karena proporsi ini memberikan keseimbangan yang baik antara kebutuhan model untuk mendapatkan variasi informasi yang cukup selama pelatihan dan kebutuhan untuk mempertahankan subset uji yang representatif.

Proses pembagian menggunakan teknik *stratified sampling* agar distribusi empat kelas kelulusan tetap proporsional pada setiap subset. Stratifikasi ini sangat penting mengingat dataset memiliki ketidakseimbangan kelas yang cukup tajam, di mana kelas "Lulus Tepat Waktu" jauh lebih dominan dibandingkan kategori lain seperti "Drop Out". Tanpa stratifikasi, ada kemungkinan data validasi atau uji tidak mengandung satu atau lebih kelas tertentu yang dapat menyebabkan model tidak pernah dievaluasi pada kelas minoritas sehingga kinerjanya tidak dapat dipastikan untuk konteks penggunaan dunia nyata. Dengan stratifikasi, setiap subset dipastikan mengandung representasi semua kelas, meskipun dalam proporsi yang sama dengan data asli.

Selain itu, pembagian dataset menggunakan *seed* acak tetap, misalnya *seed* = 42 agar seluruh proses dapat direplikasi dan diulang dengan hasil yang konsisten. Penggunaan *seed* ini penting karena tanpa parameter pengontrol tersebut, pembagian data dapat berbeda setiap kali proses dilakukan yang dapat menghasilkan perbedaan performa model [119]. Dalam penelitian ilmiah, kemampuan mereplikasi hasil merupakan aspek penting dalam memastikan validitas eksperimen. Oleh karena itu, setiap kali pembagian *dataset* dilakukan, *seed* yang sama digunakan di seluruh percobaan, baik selama eksplorasi maupun pada model final [120].

Pembagian ini juga memiliki implikasi teknis terhadap tahapan lain seperti SMOTE, *scaling*, dan *tuning hyperparameter*. Transformasi apa

pun termasuk proses penyeimbangan kelas hanya boleh dilakukan pada data pelatihan. Itulah mengapa pembagian dilakukan terlebih dahulu sebelum tahapan lain. Jika penyeimbangan dilakukan sebelum pembagian, informasi dari kelas minoritas dapat bocor ke subset lain melalui sampel sintetis yang berasal dari sampel yang seharusnya hanya ada di *training set* [83]. Praktik ini akan menyebabkan masalah *data leakage* yang dapat membuat performa model tampak jauh lebih baik daripada kenyataan. Dengan melakukan pembagian terlebih dahulu, seluruh subset tetap murni, dan model dapat dievaluasi dengan objektif di data validasi dan uji yang tidak pernah terlibat dalam proses pembelajaran.

Terakhir, pembagian dataset juga memungkinkan penelitian ini melakukan evaluasi berlapis. Subset validasi digunakan untuk memilih kombinasi *hyperparameter* terbaik selama proses optimasi, sedangkan subset uji ditahan khusus untuk evaluasi final. Hal ini meningkatkan kredibilitas hasil karena performa akhir model didasarkan pada dataset yang benar-benar tidak pernah terlihat oleh model sebelumnya, sehingga menggambarkan kemampuan generalisasi yang sesungguhnya.

3.2.3.6 Penyeimbangan Data Menggunakan SMOTE dan SMOTETomek

Setelah dataset dibagi, langkah penting berikutnya adalah menangani ketidakseimbangan kelas yang muncul secara signifikan pada data pelatihan. Ketidakseimbangan ini tidak hanya membuat model bias terhadap kelas mayoritas, tetapi juga memberi dampak negatif pada metrik-metrik evaluasi seperti *recall* dan F1-*score*, terutama pada kelas minoritas yang sebenarnya paling penting untuk tujuan prediksi dini. Untuk mengatasi hal ini, penelitian ini menerapkan dua metode *resampling* yang komplementer, yaitu SMOTE (*Synthetic Minority Oversampling Technique*) dan SMOTETomek. Keduanya hanya diterapkan pada data pelatihan agar tidak mengganggu distribusi asli pada subset validasi dan uji.

Penerapan SMOTE dimulai dengan mengidentifikasi sampel-sampel pada kelas minoritas dan kemudian memprosesnya untuk menghasilkan sampel sintetis baru melalui interpolasi antara setiap sampel minoritas dengan tetangga terdekatnya. Parameter jumlah tetangga (k-*nearest neighbors*) ditetapkan pada nilai k = 5 yang merupakan nilai optimal untuk dataset dengan jumlah fitur menengah dan distribusi yang tidak terlalu padat [121]. Proses interpolasi dilakukan di ruang fitur numerik, menggunakan nilai transformasi z-*score* yang sebelumnya telah diterapkan. Hasilnya adalah sampel sintetis yang berada pada ruang keputusan yang wajar dan representatif dari pola kelas minoritas. Dengan menambahkan sampel sintetis ini, jumlah sampel pada kelas minoritas meningkat secara bertahap sehingga proporsinya tidak lagi ekstrem terhadap kelas mayoritas.

Namun SMOTE memiliki kelemahan yaitu sering kali menciptakan sampel sintetis yang berada terlalu dekat dengan batas kelas, terutama ketika distribusi kelas tumpang tindih. Untuk mengatasi kelemahan ini, digunakan teknik lanjutan yaitu SMOTETomek yang menggabungkan SMOTE dengan proses *Tomek Links removal*. *Tomek Links* merupakan pasangan sampel dari dua kelas berbeda yang berada sangat berdekatan satu sama lain, sehingga biasanya menandakan area tumpang tindih antar kelas. Dengan menghapus sampel dari kelas mayoritas yang termasuk dalam *Tomek Links*, ruang fitur menjadi lebih bersih dan batas antar kelas lebih tegas. Proses ini penting dalam dataset akademik karena nilai-nilai transkrip mahasiswa sering kali saling berdekatan secara numerik meskipun berasal dari kategori kelulusan yang berbeda.

Selain menghasilkan dataset yang lebih seimbang, SMOTETomek juga meningkatkan kualitas pembelajaran model. Dengan menggabungkan proses penambahan dan pembersihan sampel, model tidak hanya mendapatkan representasi lebih besar dari kelas minoritas tetapi juga belajar dari ruang fitur yang lebih terstruktur. Proporsi hasil akhir pada *training set* biasanya diatur agar kelas minoritas menyumbang sekitar 25 hingga 30 persen total sampel pelatihan. Proporsi ini dipilih untuk

memberikan representasi yang lebih baik tanpa membuat distribusi menjadi terlalu artifisial atau dominan oleh sampel sintetis.

Implementasi penyeimbangan ini dilakukan secara ketat hanya di *training set* untuk menghindari *data leakage*. Selain itu, seluruh versi dataset yang telah diseimbangkan disimpan sebagai artefak terpisah agar setiap eksperimen *modeling* dapat mengaksesnya tanpa perlu melakukan ulang proses *resampling*. Dengan langkah ini, penelitian memastikan bahwa seluruh model yang dilatih mendapatkan kesempatan mempelajari kelas minoritas secara proporsional, sekaligus mempertahankan integritas validasi dan uji.

3.2.4 Modeling

Bagian pemodelan dalam penelitian ini berfokus pada pengembangan model prediksi kelulusan berdasarkan tiga titik evaluasi akademik utama, yaitu akhir Semester 2, Semester 4, dan Semester 6. Setiap model dibangun untuk mencerminkan kebutuhan analitis pada tahap perkembangan mahasiswa yang berbeda, sehingga teknik preparasi, strategi penyeimbangan kelas, serta konfigurasi model disesuaikan dengan karakteristik data di masing-masing semester. Seluruh model menggunakan CatBoost sebagai algoritma inti karena performanya yang stabil pada dataset berukuran menengah, kemampuannya menangani fitur kategorikal secara *native*, serta kecocokannya terhadap skenario data tidak seimbang. Proses pemodelan mencakup pembagian data, penyeimbangan kelas, pemilihan fitur, serta optimasi parameter menggunakan pendekatan pencarian berbasis Optuna. Dengan struktur ini, setiap model tidak hanya disiapkan untuk mencapai performa terbaik pada semesternya masing-masing, tetapi juga mengikuti prosedur yang konsisten sehingga hasilnya dapat dibandingkan dan dianalisis secara menyeluruh pada bab berikutnya.

3.2.4.1 Pemodelan Semester 2

Pemodelan pada Semester 2 bertujuan untuk menghasilkan model prediksi yang mampu mengenali mahasiswa yang berpotensi mengalami

dropout pada tahap awal studi. Seluruh proses pemodelan dirancang agar model dapat belajar dari pola akademik dua semester pertama, dengan pendekatan yang konsisten pada seluruh tahap pemodelan dari pembagian dataset, penyeimbangan kelas, pemilihan algoritma, hingga optimasi *hyperparameter* menggunakan Optuna. Pemodelan Semester 2 menggunakan algoritma CatBoost sebagai model utama karena kinerjanya yang stabil pada dataset berukuran kecil dan kemampuannya menangani fitur kategorikal maupun data hasil *encoding* secara efisien. Selain itu, Pemodelan Semester 2 ini hanya menggunakan 1 versi model karena hanya memprediksi 2 *class* yaitu Dropout atau Aman (termasuk Lulus Tepat Waktu, Tidak Lulus Tepat Waktu, atau Lulus Lebih Awal)

Tahap pertama dalam proses *modeling* adalah membagi *dataset* menjadi data pelatihan dan data pengujian dengan proporsi 80:20. Pembagian dilakukan sebelum proses penyeimbangan kelas untuk memastikan bahwa pengujian mencerminkan distribusi alami kategori *dropout* yang sangat kecil pada semester awal. Setelah pembagian dilakukan, proses penyeimbangan kelas diterapkan pada data pelatihan menggunakan SMOTE. Teknik SMOTE digunakan untuk menghasilkan sampel sintetis melalui interpolasi antar-sampel minoritas sehingga jumlah sampel *dropout* menjadi sebanding dengan kelas non-*dropout*. Transformasi ini penting karena tanpa penyeimbangan, model akan cenderung memprediksi semua mahasiswa sebagai tidak *dropout,* mengingat ketidakseimbangan kelas yang ekstrem. Hanya data pelatihan yang dikenai SMOTE, sementara subset pengujian dibiarkan murni sebagai acuan evaluasi akhir.

Setelah proses penyeimbangan selesai, langkah berikutnya adalah melakukan optimasi *hyperparameter* menggunakan Optuna. Tahap optimasi ini merupakan inti dari pemodelan Semester 2 karena bertujuan untuk menemukan konfigurasi parameter yang menghasilkan performa terbaik berdasarkan proses validasi silang. Optuna menjalankan serangkaian percobaan dengan mekanisme pencarian berbasis *Tree-structured Parzen*

Estimator (TPE) untuk mengevaluasi kombinasi parameter yang berbeda. Parameter yang dioptimalkan mencakup *learning rate* pada rentang 0.01 hingga 0.2, kedalaman pohon (*depth*) pada rentang 3 hingga 8, serta regularisasi L2 (l2_leaf_reg) antara 1 hingga 5. Setiap kombinasi parameter diuji menggunakan tiga lipatan validasi silang (3-*fold Stratified K-Fold*), yang berarti data pelatihan dibagi menjadi tiga bagian yang digunakan secara bergantian sebagai pelatihan dan validasi [19].

Pada setiap lipatan validasi, model CatBoost dilatih dengan parameter sementara yang diberikan Optuna, kemudian dinilai menggunakan *macro* F1-*score*. *Macro* F1 dipilih sebagai metrik utama karena mengukur performa rata-rata pada kedua kelas secara merata dan sangat sensitif terhadap performa kelas minoritas, sehingga cocok untuk mengukur kemampuan model mengenali mahasiswa yang berisiko *dropout*. Setiap percobaan Optuna tidak hanya mengembalikan nilai F1-*score*, tetapi juga mencatat metrik tambahan seperti nilai *loss* pada validasi, akurasi, dan waktu eksekusi untuk memberikan gambaran efisiensi parameter yang diuji. Setelah seluruh uji coba selesai, Optuna memilih parameter dengan nilai rerata F1-*score* tertinggi sebagai hasil optimasi.

Dengan parameter terbaik yang diberikan Optuna, tahap selanjutnya adalah melatih model CatBoost final menggunakan seluruh data pelatihan yang telah diseimbangkan. Model dilatih dengan mekanisme *early stopping* untuk mencegah *overfitting*, di mana proses pelatihan dihentikan jika tidak ada peningkatan performa pada subset evaluasi dalam sejumlah iterasi berturut-turut. Pada tahap ini, *subset* pengujian (20% awal yang dipisahkan) digunakan sebagai eval_set, memungkinkan model memantau performanya terhadap data yang tidak pernah digunakan selama pelatihan. Pendekatan ini memastikan bahwa model final tidak hanya belajar dari data pelatihan tetapi juga tetap selaras dengan pola pada data uji.

Setelah pelatihan selesai, evaluasi model dilakukan dengan memprediksi kelas mahasiswa pada subset pengujian. Evaluasi

menggunakan metrik seperti *precision*, *recall*, *accuracy*, *macro F1-score*, dan ROC-AUC untuk mengukur kemampuan model secara menyeluruh. Selain menghasilkan hasil klasifikasi, model juga *menampilkan confusion matrix* dan analisis *one-vs-rest* untuk melihat seberapa baik model membedakan kelas *dropout* dari kelas lainnya secara spesifik. Tambahan lagi, riwayat pelatihan ditampilkan dalam bentuk grafik *loss*, AUC, dan akurasi untuk memberikan wawasan mengenai dinamika proses optimasi model sejak iterasi awal hingga titik berhenti dini.

Sebagai langkah terakhir dalam modeling Semester 2, model dilatih ulang menggunakan 100% data historis yang telah diseimbangkan agar model final memiliki cakupan informasi selengkap mungkin. Pada tahap ini, seluruh fitur kembali melalui proses *one-hot encoding* dan SMOTE sebelum pelatihan ulang dilakukan. Model final yang dihasilkan dari proses ini kemudian siap digunakan sebagai sistem prediksi risiko *dropout* untuk mahasiswa yang berada pada semester awal.

3.2.4.2 Pemodelan Semester 4

Pemodelan pada Semester 4 mengalami beberapa iterasi pendekatan hingga membentuk lima versi model yang memiliki karakteristik berbeda. Perbedaan ini terutama terlihat dari metode penyeimbangan kelas, ruang pencarian *hyperparameter*, dan variasi konfigurasi CatBoost yang digunakan.

Pada setiap versi, fokus eksperimen diarahkan untuk memahami bagaimana perubahan teknik *balancing* dan strategi *tuning* memengaruhi pola prediksi model terhadap empat kelas *output*. Versi awal menggunakan pendekatan *balancing* agresif untuk memastikan kelas minor tidak tenggelam oleh mayoritas, sedangkan versi-versi berikutnya mulai menerapkan kombinasi SMOTE, *Tomek Links*, *reweighting*, serta variasi pengaturan kedalaman pohon, banyaknya iterasi, dan nilai regulasi. Proses ini memberi gambaran yang jelas mengenai sensitivitas kinerja CatBoost terhadap perubahan distribusi data maupun parameter pelatihan.

Tabel 3.1 *Hyperparameter Tuning* Model Semester 4

Category	Hyperparameter	Versi 1	Versi 2	Versi 3 (Baseline) [122]	Versi 4	Versi 5 (Final Optimized)
Konfigurasi Dataset	Resampling Method	SMOTE (target 400 utk kelas 1 & 3)	SMOTE (target 300 utk kelas 1 & 3)	SMOTE (dari v1/v2)	SMOTE + Tomek	SMOTE + Tomek
	Target Balancing Result	{1:400, 3:400}	{1:300, 3:300}	mengikuti data resampling	Distribusi baru: 105–317	Distribusi baru: 138–148
	Cat Features	None (tidak terdeteksi)	None	None	None	None
Search Space & Optimasi	CV Method	Holdout 80–20	3-Fold CV	Tanpa Optuna	3-Fold CV	3-Fold CV
	Objective Metric	Macro F1	Macro F1	Macro F1	Macro F1	Macro F1
	Optuna Trials	25	25	–	30	35
Arsitektur Model	Iterations	409	488	200	526	913
	Depth	7	9	4	10	9
	Learning Rate	0.1374	0.02297	0.03	0.1315	0.1087
	L2 Leaf Reg	2.104	1.097	3	6.6227	1.669
	Grow Policy	Default	Default	Default	SymmetricTree	Default

Category	Hyperparameter	Versi 1	Versi 2	Versi 3 (Baseline) [122]	Versi 4	Versi 5 (Final Optimized)
	Subsample	–	–	–	0.8375	0.8757
	RSM	–	–	–	0.7694	0.9703
	Random Strength	–	–	–	0.2503	0.422
Pengaturan Pelatihan	Loss Function	MultiClass	MultiClass	MultiClass	MultiClass	MultiClass
	Eval Metric	TotalF1	TotalF1	TotalF1	TotalF1	TotalF1
	Auto Class Weights	None	Balanced	Balanced	None (reweight manual)	Balanced
	Manual Class Reweighting	Tidak	Tidak	Tidak	Ya (best weights ditemukan: [1.0, 1.1, 1.0, 1.0])	Tidak
	Early Stopping / OD Wait	Default	Default	Tidak digunakan	OD Wait = 60	OD Wait = 60
	Bootstrap Type	Default	Default	Default	Default	Bernoulli
Waktu & Kompleksitas	Rata-rata Waktu per Trial	± 11 detik	± 49 detik	–	± 21 detik	± 95 detik
	Total Training Behaviour	Iterasi berhenti di early stop iter ke 22	Early stop di iter 54	Best iter 180	Early stop di iter 6	Early stop di iter 13

Versi pertama menggunakan SMOTE dengan target peningkatan jumlah sampel pada dua kelas minoritas sampai mencapai 400 sampel. Pendekatan ini dipilih agar model mendapat lebih banyak contoh dari kelas yang sebelumnya langka. Optimasi dilakukan menggunakan 25 percobaan Optuna dan rentang parameter yang relatif sempit, dengan kombinasi final seperti 409 iterasi, *depth* 7, *learning rate* 0.137, dan regularisasi L2 sebesar 2.10. Hasil akhir menunjukkan bahwa versi ini berfungsi sebagai pondasi proses tuning, namun ruang pencariannya masih terbatas.

Versi kedua mengadopsi pendekatan yang serupa namun dengan jumlah sampel sintetis yang lebih rendah, yaitu 300 untuk masing-masing dua kelas minoritas. Penyeimbangan yang lebih ringan ini sengaja dipilih agar distribusi data tidak menjadi terlalu artifisial. Pada versi ini, optimasi masih dilakukan dengan 25 percobaan Optuna, tetapi seluruh proses *tuning* menggunakan validasi silang tiga lipatan sehingga menghasilkan parameter yang lebih stabil. Parameter terbaik pada versi ini cenderung menunjukkan kedalaman model lebih tinggi, yaitu *depth* 9, iterasi 488, secara bersamaan dengan *learning rate* sangat kecil sekitar 0.022. Kombinasi ini menghasilkan model dengan struktur pohon yang lebih kompleks namun belajar secara lebih bertahap.

Versi ketiga berfungsi sebagai *baseline* dan tidak menggunakan Optuna. Model ini dilatih menggunakan parameter konservatif seperti 200 iterasi, depth 4, dan learning rate 0.03. Konfigurasi ini memungkinkan model mempelajari pola secara perlahan dan memberikan pembanding awal untuk mengetahui seberapa besar peningkatan yang diberikan oleh tuning pada versi lain. Model *baseline* tidak melalui proses penyesuaian yang luas, sehingga versi ini menonjolkan bagaimana CatBoost bekerja dalam kondisi default setelah penyeimbangan kelas diterapkan.

Pada versi keempat, pendekatan *balancing* diperluas dengan menggunakan SMOTE yang dipadukan dengan Tomek Links. Kombinasi ini bertujuan tidak hanya memperbesar jumlah kelas minoritas, tetapi juga

membersihkan sampel yang tumpang tindih pada batas kelas sehingga ruang fitur menjadi lebih rapi. Proses *tuning* pada versi ini diperluas menggunakan 30 percobaan Optuna dan mencakup parameter tambahan seperti *subsample*, rsm, *random strength*, dan *grow policy*. Parameter terbaik menunjukkan kedalaman pohon maksimal (*depth* 10), iterasi 526, serta *learning rate* menengah sekitar 0.131. Selain itu, versi ini menggunakan pendekatan reweighting manual untuk menyesuaikan bobot kelas setelah balancing. Perubahan ini menjadikan versi keempat salah satu konfigurasi paling agresif dalam hal eksplorasi parameter dan pengendalian struktur model.

Versi kelima menjadi versi paling komprehensif dalam tahap pemodelan Semester 4. Pendekatan *balancing* menggunakan SMOTE+Tomek seperti versi sebelumnya, namun proses *tuning* dibuat jauh lebih luas dengan 35 percobaan Optuna dan ruang pencarian hyperparameter yang lebih besar. Parameter tambahan seperti *subsample*, rsm, dan *random strength* dioptimalkan secara aktif, serta model menggunakan *bootstrap* Bernoulli untuk variasi pohon yang lebih stabil. Parameter terbaik versi ini memiliki iterasi 913, *depth* 9, *learning rate* sekitar 0.108, dan regularisasi L2 sebesar 1.67. Dari seluruh versi, versi kelima memiliki konfigurasi paling ekstensif yang menggabungkan *balancing* yang bersih, pencarian *hyperparameter* yang luas, dan kompleksitas model yang maksimal untuk meningkatkan representasi pola dari seluruh kelas.

Secara keseluruhan, pemodelan Semester 4 berkembang dari pendekatan sederhana berbasis SMOTE dan parameter konservatif menuju pendekatan yang lebih matang yang menggabungkan *balancing* berbasis SMOTE+Tomek, *tuning multi-parameter*, dan pengaturan pembobotan otomatis. Setiap versi dibangun di atas kekuatan dan keterbatasan versi sebelumnya, hingga akhirnya menghasilkan strategi pemodelan yang lebih stabil dan responsif terhadap variasi kelas pada semester menengah.

3.2.4.3 Pemodelan Semester 6

Pemodelan Semester 6 dikembangkan melalui beberapa versi (4 versi berbeda) yang menunjukkan peningkatan bertahap dalam cara model menangani data akademik dan biodata mahasiswa pada tahap akhir perkuliahan. Perbedaan utama antara versi tersebut terletak pada mekanisme penyeimbangan kelas, ruang pencarian *hyperparameter*, integrasi fitur tambahan, serta strategi *ensemble* yang diterapkan untuk mendapatkan representasi performa yang lebih stabil pada Tabel 3.2. Versi pertama berfungsi sebagai pondasi utama. Proses dimulai dengan memadukan fitur hasil ekstraksi semester enam dengan label kelulusan mahasiswa. Seluruh nilai kosong diubah menjadi nol agar tidak mengganggu pemrosesan numerik. Fitur kategorikal dideteksi, meskipun dalam versi pertama jumlah fitur bertipe kategori masih terbatas. Setelah pemisahan data latih dan data uji, penyeimbangan distribusi kelas dilakukan menggunakan SMOTE yang dipadukan dengan *Tomek Links* sehingga empat kelas akhir memiliki komposisi yang lebih seragam. Setelah data seimbang terbentuk, model CatBoost dioptimalkan menggunakan pendekatan pencarian *hyperparameter* berbasis Optuna dengan validasi silang tiga lipatan. Ruang pencarian mencakup iterasi antara 400 sampai 900, kedalaman 5 sampai 10, serta variasi parameter seperti *subsample*, rsm, dan tingkat regularisasi. Model akhir dilatih kembali menggunakan parameter terbaik yang diperoleh dari proses pencarian tersebut.

Versi kedua memperluas pendekatan dengan melakukan integrasi fitur yang jauh lebih kaya. Pada versi ini, fitur akademik tambahan untuk semester satu hingga enam dimasukkan ke dalam pemodelan. Fitur yang ditambahkan mencakup *slope* perkembangan IPS, standar deviasi IPS, total SKS gagal, frekuensi mata kuliah yang diulang, rata-rata kehadiran, serta rasio SKS gagal terhadap total SKS. Selain itu, biodata mahasiswa diproses lebih lanjut melalui kategorisasi asal daerah, tipe sekolah, dan konversi tingkat pendidikan orang tua menjadi skala numerik. Gabungan fitur tambahan ini menciptakan representasi akademik dan non-akademik yang

lebih utuh. Mengingat keberadaan fitur kategorikal yang lebih banyak, CatBoost digunakan sepenuhnya dalam mode *native categorical handling*. Proses *balancing* tetap menggunakan SMOTE+Tomek, tetapi dilakukan melalui *encoding* sementara untuk menghindari kesalahan struktur data. Setelah penyeimbangan berhasil dilakukan, versi ini menerapkan pencarian *hyperparameter* menggunakan rentang parameter yang serupa, namun dengan orientasi pada stabilitas melalui pengaturan *bootstrap Bernoulli*. Setelah parameter terbaik diperoleh, versi ini menyertakan strategi *ensemble* tiga model dengan *seed* berbeda. Probabilitas dari ketiga model dirata-ratakan untuk menghasilkan prediksi akhir. Pendekatan *ensemble* ini memberikan stabilitas yang lebih baik saat model dihadapkan pada variasi data.

Versi ketiga dirancang untuk mengurangi risiko *overfitting* yang teridentifikasi pada versi sebelumnya. Pendekatan ini menggunakan regularisasi yang lebih kuat dan validasi silang lima lipatan untuk memberikan gambaran performa yang lebih realistis. Ruang pencarian *hyperparameter* dibatasi pada kedalaman yang lebih dangkal, tingkat pembelajaran yang lebih konservatif, dan regularisasi L2 yang lebih tinggi. CatBoost tetap digunakan dalam format kategori *native*, sedangkan *balancing* data tetap dilakukan menggunakan SMOTE+Tomek dengan mekanisme *encoding* sementara yang sama. Setelah parameter optimal diperoleh, versi ini membangun *ensemble* lima model untuk memperkuat stabilitas prediksi dan menangkap variasi antar model yang dihasilkan oleh *seed* berbeda. *Ensemble* probabilitas kembali digunakan sebagai mekanisme penggabungan keluaran. Selain itu, validasi silang global diterapkan untuk mengukur konsistensi performa model pada seluruh *dataset* tanpa mengandalkan satu kali *split* saja.

Versi keempat memadukan rangkaian lengkap fitur akademik dan biodata, serta menyesuaikan *pipeline* untuk memastikan kompatibilitas penuh antara proses *balancing*, *encoding*, dan pemodelan. Pada versi ini,

seluruh fitur diubah menjadi numerik melalui teknik *one-hot encoding* agar kompatibel dengan mekanisme SMOTE+Tomek. Setelah penyeimbangan

berhasil dilakukan, paket fitur numerik diteruskan ke CatBoost meskipun tidak lagi ada fitur kategorikal yang ditangani secara *native*. Optimasi dilakukan menggunakan ruang pencarian yang luas seperti iterasi 300 hingga 900, kedalaman 5 hingga 10, variasi *subsample*, serta kontrol kekuatan *random*. Validasi silang tiga lipatan memastikan bahwa model tidak bergantung pada satu set data tertentu selama proses pencarian. Setelah parameter terbaik diperoleh, model akhir dilatih kembali dengan konfigurasi optimal tersebut sehingga seluruh fitur yang telah diolah dapat digunakan secara penuh oleh algoritma.

Tabel 3.2 *Hyperparameter Tuning* Model Semester 6

Category	Hyperparameter	Versi 1	Versi 2	Versi 3	Versi 4
Konfigurasi Dataset	Resampling Method	SMOTE (target 400 utk kelas 1 & 3)	SMOTE (target 300 utk kelas 1 & 3)	SMOTE (dari v1/v2)	SMOTE + Tomek
	Target Balancing Result	{1:400, 3:400}	{1:300, 3:300}	mengikuti data resampling	Distribusi baru: 105–317
	Cat Features	None (tidak terdeteksi)	None	None	None
Search Space & Optimasi	CV Method	Holdout 80–20	3-Fold CV	3-Fold CV	3-Fold CV
	Objective Metric	Macro F1	Macro F1	Macro F1	Macro F1
	Optuna Trials	25	25	–	30
Arsitektur Model	Iterations	409	488	200	526
	Depth	7	9	4	10
	Learning Rate	0.1374	0.02297	0.03	0.1315
	L2 Leaf Reg	2.104	1.097	3	6.6227
	Grow Policy	Default	Default	Default	SymmetricTree
	Subsample	–	–	–	0.8375

Category	Hyperparameter	Versi 1	Versi 2	Versi 3	Versi 4
	RSM	–	–	–	0.7694
	Random Strength	–	–	–	0.2503
Pengaturan Pelatihan	Loss Function	MultiClass	MultiClass	MultiClass	MultiClass
	Eval Metric	TotalF1	TotalF1	TotalF1	TotalF1
	Auto Class Weights	None	Balanced	Balanced	None (reweight manual)
	Manual Class Reweighting	Tidak	Tidak	Tidak	Ya (best weights ditemukan: [1.0, 1.1, 1.0, 1.0])
	Early Stopping / OD Wait	Default	Default	Tidak digunakan	OD Wait = 60
	Bootstrap Type	Default	Default	Default	Default
Waktu & Kompleksitas	Rata-rata Waktu per Trial	± 11 detik	± 49 detik	–	± 21 detik
	Total Training Behaviour	Iterasi berhenti di early stop iter ke 22	Early stop di iter 54	Best iter 180	Early stop di iter 6

Secara keseluruhan, pemodelan Semester 6 berkembang dari pendekatan dasar yang menggabungkan *balancing* sederhana dan *tuning* terarah menjadi *pipeline* yang lebih matang, lengkap, dan stabil. Pengayaan fitur bertahap, eksplorasi *hyperparameter* yang diperluas, penanganan kategori secara *native*, hingga *ensemble multi-seed* menunjukkan bahwa model Semester 6 dirancang untuk menangkap dinamika akademik mahasiswa tingkat lanjut dengan lebih baik dibandingkan model semester sebelumnya. Pendekatan ini menghasilkan pemodelan yang lebih komprehensif dan representatif terhadap karakteristik kemajuan studi mahasiswa mendekati masa kelulusan.

3.2.4.4 Pemodelan Semester 6 (Pemanfaatan Data Eksternal)

Pada tahap pemodelan Semester 6 dengan tambahan data eksternal, proses pengembangan model dilakukan melalui rangkaian langkah yang lebih komprehensif dibanding versi sebelumnya. *Dataset* yang digunakan menggabungkan fitur akademik hasil rekonstruksi hingga semester enam, fitur biodata mahasiswa, serta lima variabel baru dari data responden, yaitu Dukungan Keluarga Akademik, Dukungan Finansial Keluarga, Diskusi Akademik dengan Orang Tua, Dukungan Keluarga terhadap Jurusan, dan Kesesuaian Jurusan dengan Keinginan. Seluruh variabel responden diperlakukan sebagai fitur kategorikal. Proses *resampling* dilakukan menggunakan SMOTENC karena fitur kategorikal cukup dominan. Pada tahap ini, fitur kategorikal terlebih dahulu diubah menjadi representasi ordinal untuk kebutuhan SMOTENC, kemudian dikembalikan ke bentuk string agar kompatibel dengan mekanisme penanganan kategori asli CatBoost. *Resampling* menghasilkan distribusi yang benar–benar seimbang untuk keempat kelas keluaran, masing-masing berjumlah 149 sampel.

Pada proses optimasi, model menjalankan pencarian *hyperparameter* menggunakan Optuna dengan skema 3-*fold cross-validation*. Setiap trial diatur untuk memaksimalkan skor *Macro* F1 sebagai metrik objektif. Jumlah percobaan ditetapkan sebanyak 25 *trial*, dan setiap

trial menggunakan kombinasi parameter yang diambil dari ruang pencarian yang telah ditentukan. Dari eksplorasi tersebut, Optuna memilih konfigurasi terbaik yang mencakup 784 iterasi, kedalaman pohon sebesar 9, *learning rate* sebesar 0.10697, serta nilai l2_leaf_reg sekitar 2.65. Aspek regularisasi lain juga muncul sebagai temuan penting, seperti *subsample* sebesar 0.8012 dan *random subspace method* (RSM) mendekati 1.0. Nilai *random_strength* yang sangat kecil menunjukkan bahwa model cenderung lebih stabil dan tidak banyak menambahkan *noise* selama proses pembentukan pohon. Seluruh konfigurasi berjalan dengan *bootstrap_type Bernoulli*, *loss function MultiClass*, *eval metric* TotalF1, serta *auto_class_weights* dalam mode *Balanced*. Parameter *overfitting detector* ditetapkan pada skema Iter dengan od_wait selama 50 iterasi.

Tabel 3.3 *Hyperparameter Tuning* pada Pemodelan Sems. 6 dengan Data Eksternal

Category	**Hyperparameter**	
Konfigurasi Dataset	Resampling Method	SMOTENC (handling categorical via ordinal encode ke restore)
	Target Balancing Result	{Dropout:149, LLA:149, LTW:149, TLTTW:149}
	Cat Features (Native)	ASAL_DAERAH_CAT, ASAL_SEKOLAH_CAT, Dukungan_Keluarga_Akademik, Dukungan_Finansial_Keluarga_Cukup, Diskusi_Akademik_Ortu, Dukungan_Keluarga_Jurusan, Jurusan_Sesuai_Keinginan
Search Space & Optimasi	CV Method	3-Fold CV (Optuna)
	Objective Metric	Macro F1
	Optuna Trials	25
Arsitektur Model	Iterations	400–900, best = 784
	Depth	5–10, best = 9
	Learning Rate	0.03–0.15, best = 0.10697
	L2 Leaf Reg	1–6, best = 2.651
	Grow Policy	Default
	Subsample	0.8–1.0, best = 0.8012
	RSM	0.8–1.0, best = 0.9999

	Random Strength	0.0–1.0, best = 0.00198
Pengaturan Pelatihan	Loss Function	MultiClass
	Eval Metric	TotalF1
	Auto Class Weights	Balanced
	Manual Class Reweighting	Tidak
	Early Stopping / OD Wait	50
	Bootstrap Type	Bernoulli
	Ensemble	Tidak
Waktu & Kompleksitas	Rata-rata Waktu per Trial	± 156 detik
	Total Training Behaviour	Early stop pada iterasi ke 114 (model shrink ke 115)

Setelah menemukan set *hyperparameter* terbaik pada Tabel 3.3, model akhir dilatih menggunakan dataset historis hasil resampling yang sudah seimbang. CatBoost dijalankan dengan memanfaatkan fitur kategorikal secara *native* dan menyesuaikan semua parameter yang diperoleh dari proses *tuning*. Meskipun model melakukan pelatihan hingga 784 iterasi maksimum, mekanisme *early stopping* menghentikan proses lebih cepat ketika model mencapai kondisi terbaik di sekitar iterasi ke-114. Konfigurasi ini menjadi dasar model akhir Semester 6 dengan integrasi data eksternal, yang merepresentasikan versi pertama dari pendekatan yang menggabungkan data akademik, biodata, dan respon psikososial mahasiswa. Model ini menjadi *baseline* awal sebelum dilakukan eksplorasi ke versi-versi berikutnya yang dapat mencakup penyesuaian fitur, optimasi lanjutan, atau penambahan mekanisme *ensemble*.

3.2.5 Evaluation

Tahap evaluasi dilakukan untuk menilai kemampuan model dalam memprediksi status kelulusan mahasiswa berbasis fitur akademik, biodata, serta data responden. Mengingat penelitian ini berfokus pada permasalahan klasifikasi dengan dataset yang relatif kecil dan memiliki ketidakseimbangan

kelas, proses evaluasi dirancang agar mampu memberikan gambaran yang akurat, stabil, dan tidak bias terhadap kelas mayoritas. Seluruh evaluasi dilakukan menggunakan data *holdout test set* yang dipisahkan secara *stratified* pada tahap awal, sehingga hasil penilaian benar-benar merepresentasikan performa model pada data baru.

Evaluasi utama dilakukan menggunakan *classification report* yang menyajikan *precision*, *recall*, dan F1-*score* untuk setiap kelas keluaran yang dapat dilihat pada Rumus 3.1. Penggunaan *macro* F1 menjadi metrik sentral karena dapat menilai performa model dengan memberikan bobot yang sama pada setiap kelas, termasuk kelas minoritas yang jumlahnya jauh lebih kecil dibanding kelas lainnya. Metrik ini selaras dengan tujuan penelitian yang ingin memastikan kemampuan model dalam mengidentifikasi semua kategori mahasiswa, bukan hanya kategori dengan jumlah data besar.

$$ccuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

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

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

$$F1 - Score = 2\,x\,\frac{Precision\,x\,Recall}{Precision + Recall}$$

$$False\ positive\ Rate = \frac{FP}{FP + TN}$$

$$AUC = \int_0^1 TPR(FPR)\,d\,(FPR)$$

$$Confusion\ Matrix = \begin{matrix} TP & FP \\ FN & TN \end{matrix}$$

$$T_{batch\_avg} = \frac{\sum_I^N = 1^{Ti}}{N}$$

Rumus 3.1 *Evaluation Metrics* yang Digunakan

Selain itu, model juga dievaluasi melalui *confusion matrix*. Matriks ini memberikan ilustrasi proporsi prediksi yang benar dan salah per kelas sehingga memungkinkan analisis detail terhadap pola kesalahan yang terjadi. Pada kasus multi-kelas seperti dalam penelitian ini, *confusion matrix* membantu mengidentifikasi kelas mana yang paling sulit dibedakan oleh model dan apakah terdapat *misclassification* yang konsisten antar kategori tertentu.

Pada beberapa versi model, evaluasi diperluas menggunakan validasi silang *stratified* k-*fold*, terutama pada proses *tuning* dengan Optuna. Teknik ini membantu memperkirakan kestabilan performa model terhadap variasi subset data latih dan meminimalkan kemungkinan *overfitting*. Di beberapa percobaan, model juga dianalisis melalui riwayat pelatihan (*training history*), seperti *training loss*, *validation loss*, AUC, dan akurasi pada setiap iterasi. Visualisasi kurva pelatihan membantu mengevaluasi apakah model mengalami *overfitting* atau *underfitting*, sekaligus memastikan bahwa mekanisme *early stopping* bekerja sebagaimana mestinya.

Evaluasi tambahan dilakukan dengan mencatat *best iteration* hasil *early stopping* untuk melihat pada titik mana model mencapai performa optimal. Informasi ini penting terutama untuk model CatBoost yang cenderung melatih banyak iterasi jika tidak dibatasi. Pada beberapa eksperimen, terutama versi lanjutan Semester 6, penelitian juga menyertakan evaluasi per kelas pada skema validasi silang, untuk memastikan bahwa seluruh target kelas memiliki performa yang dapat diterima.

Secara keseluruhan, tahap evaluasi dalam penelitian ini menggabungkan penilaian metrik klasifikasi, visualisasi kesalahan, pemantauan dinamika pelatihan, serta validasi silang, sehingga memberikan gambaran menyeluruh mengenai kemampuan model dalam memprediksi status akademik mahasiswa.

3.2.6 Deployment

Tahap *deployment* menjadi langkah akhir dalam alur pengembangan model, di mana sistem yang telah melalui proses pelatihan, *tuning*, dan evaluasi diimplementasikan ke dalam bentuk aplikasi yang dapat digunakan secara langsung oleh pengguna akhir. Tujuan utama tahap ini adalah memastikan bahwa model *Early Warning System* (EWS) mampu beroperasi dalam lingkungan nyata, menerima input data mahasiswa baru, memprosesnya melalui *pipeline* fitur yang telah dibangun, dan menghasilkan prediksi status kelulusan secara otomatis. Dengan demikian, *deployment* tidak hanya berfungsi sebagai integrasi teknis, tetapi juga sebagai sarana untuk menguji keterpakaiannya dalam operasional program studi.

Proses *deployment* dilakukan dengan membangun aplikasi antarmuka berbasis *web* menggunakan *framework* Streamlit. *Framework* ini dipilih karena kemampuannya menyediakan tampilan interaktif yang ringan serta mudah diintegrasikan dengan model *machine learning* yang telah disimpan dalam format CatBoost Model (CBM). Model yang digunakan pada tahap ini adalah versi terbaik dari hasil eksperimen pada semester tertentu, kemudian disimpan ke dalam file menggunakan fungsi bawaan CatBoost. Pada saat aplikasi dijalankan, model tersebut dimuat kembali ke memori sehingga dapat melakukan prediksi secara cepat tanpa perlu melatih ulang.

Aplikasi *deployment* juga dilengkapi dengan *pipeline preprocessing* yang selaras dengan *pipeline* pada tahap pelatihan. Seluruh proses *encoding* fitur, normalisasi, penanganan nilai hilang, serta identifikasi fitur kategorikal diterapkan ulang agar input baru diproses secara konsisten dengan data historis. Dengan pendekatan ini, sistem mampu menjaga integritas prediksi dan mengurangi risiko perbedaan antara data *inference* dan data *training*. Selain itu, fungsi prediksi dibuat dalam format yang mudah dibaca pengguna, sehingga hasil prediksi status kelulusan dapat ditampilkan dalam label yang telah dikenali, bukan dalam bentuk kode numerik internal.

Setelah antarmuka berhasil dibangun, aplikasi diuji kembali menggunakan beberapa sampel data untuk memastikan bahwa *pipeline* berjalan tanpa kesalahan dan hasil prediksi konsisten dengan performa model pada fase evaluasi. *Deployment* ini memungkinkan model dipakai oleh pihak fakultas atau program studi secara praktis, misalnya untuk memantau mahasiswa yang berpotensi tidak lulus tepat waktu atau membutuhkan intervensi lebih awal. Dengan menyediakan platform interaktif, hasil penelitian ini tidak hanya berhenti pada ranah akademik, tetapi juga dapat memberikan manfaat langsung bagi pengelolaan proses akademik dan bimbingan mahasiswa di lingkungan kampus.

3.3 Teknik Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan melalui survei terstruktur menggunakan kuesioner yang dirancang untuk mengidentifikasi berbagai faktor eksternal yang berpengaruh terhadap kinerja akademik mahasiswa. Kuesioner tersebut mencakup sejumlah pertanyaan mengenai latar belakang sosial ekonomi, tingkat dukungan keluarga, kondisi lingkungan tempat tinggal, ketersediaan fasilitas belajar, serta motivasi belajar individu. Penyusunan instrumen dilakukan berdasarkan hasil kajian literatur dari penelitian-penelitian sebelumnya dan telah melalui proses validasi isi oleh ahli di bidang pendidikan dan teknologi pembelajaran agar setiap indikator yang digunakan sesuai dengan variabel penelitian.

Proses penyebaran kuesioner dilakukan secara daring melalui platform survei digital guna meningkatkan efisiensi pengumpulan data, mengingat sebagian besar responden aktif menggunakan perangkat berbasis internet. Metode ini dipilih karena mampu mempercepat distribusi, mengurangi potensi kehilangan data, serta mempermudah tahap awal pengolahan karena data tersimpan langsung dalam format terstruktur. Pengumpulan data dilaksanakan dalam periode waktu tertentu untuk menjaga konsistensi kondisi responden. Seluruh data yang diperoleh disimpan dengan menjaga kerahasiaan identitas responden dan hanya dimanfaatkan untuk keperluan akademik sesuai dengan prinsip etika penelitian ilmiah..

3.3.1 Periode Pengambilan Data

Data utama (*record* akademik mahasiswa) yang diambil pada *database* Biro Informasi Akademik (BIA) dalam penelitian ini dikumpulkan dalam rentang tahun 2020 hingga 2024. Rentang waktu tersebut dipilih agar seluruh perjalanan akademik mahasiswa Sistem Informasi dapat terpantau secara menyeluruh, mulai dari awal masa perkuliahan hingga status akhir kelulusan. Periode ini juga mencakup masa sebelum, selama, dan setelah pandemi Covid-19, sehingga memberikan variasi kondisi pembelajaran yang cukup luas. Variasi ini penting untuk menghasilkan model prediksi yang mampu mempelajari pola akademik dalam situasi yang beragam, termasuk perubahan perilaku belajar dan performa akademik akibat transisi pembelajaran daring.

Selain data akademik, penelitian ini juga menggunakan data eksternal yang diperoleh melalui survei kepada mahasiswa. Pengumpulan data survei dilakukan pada Agustus hingga Oktober 2025, setelah model dasar dibangun. Tujuan pengambilan data eksternal ini adalah untuk menambahkan dimensi non-akademik yang dapat mempengaruhi risiko keterlambatan studi maupun *dropout*, seperti dukungan keluarga, motivasi memilih jurusan, serta dinamika diskusi akademik di lingkungan rumah. Penempatan survei pada periode tersebut dilakukan secara sengaja setelah keseluruhan data akademik tersedia, sehingga integrasi antara data internal dan eksternal dapat dilakukan dengan lebih terstruktur. Data eksternal kemudian digunakan terutama untuk pemodelan pada semester 6.

3.3.2 Populasi

Populasi dalam penelitian ini mencakup seluruh mahasiswa Program Studi Sistem Informasi di Universitas Multimedia Nusantara (UMN) yang terdaftar dalam rentang tahun akademik 2020 hingga 2024. Berdasarkan rekapitulasi data akademik dan biodata mahasiswa, total populasi terdiri dari 1.024 mahasiswa. Jumlah tersebut mewakili seluruh angkatan yang menjadi fokus penelitian, mulai dari mahasiswa yang memulai studi sebelum pandemi,

hingga angkatan yang mengikuti perkuliahan dalam masa transisi pembelajaran daring dan kembali ke tatap muka.

Populasi ini dipilih karena memiliki karakteristik akademik yang lengkap, mencakup nilai setiap mata kuliah, riwayat pengambilan semester, kehadiran, biodata, serta status akhir kelulusan. Selain itu, populasi mencerminkan kondisi nyata di program studi, sehingga temuan penelitian dapat menggambarkan pola risiko akademik yang relevan untuk kebutuhan *Early Warning System* (EWS). Dengan cakupan populasi yang besar dan beragam, penelitian ini memperoleh gambaran yang memadai mengenai dinamika studi mahasiswa Sistem Informasi di UMN.

3.3.3 Sampel

Sampel dalam penelitian ini terdiri dari mahasiswa Sistem Informasi yang berhasil diberi label status kelulusan secara lengkap berdasarkan empat kategori utama, yaitu Lulus Lebih Awal, Lulus Tepat Waktu, Lulus Terlambat, dan *Dropout*. Dari total populasi sebanyak 1.024 mahasiswa, hanya 421 mahasiswa yang memenuhi kriteria pelabelan tersebut.

Pembentukan sampel dilakukan melalui proses penyaringan data yang memastikan bahwa setiap mahasiswa memiliki rekam jejak akademik yang lengkap, mulai dari nilai per semester, informasi biodata, hingga status akhir studi. Mahasiswa yang masih aktif dan belum mencapai masa kelulusan dikeluarkan dari sampel karena belum memiliki outcome akhir, sehingga tidak dapat digunakan dalam proses pemodelan klasifikasi.

Pengurangan jumlah data dari 1.024 mahasiswa menjadi 421 mahasiswa terjadi sebagai konsekuensi dari proses penetapan status kelulusan yang dilakukan secara selektif dan berbasis kelengkapan riwayat akademik. Dari keseluruhan populasi, hanya mahasiswa yang telah memiliki pola studi yang dapat diinterpretasikan secara jelas yang dapat dimasukkan ke dalam sampel. Mahasiswa dari angkatan 2022 hingga 2024 sebagian besar masih berada pada tahap awal hingga pertengahan studi, sehingga status akhir kelulusannya belum dapat dipastikan secara akademik. Oleh karena itu, data

dari angkatan tersebut tidak dapat digunakan sebagai dasar pelatihan model klasifikasi karena belum memiliki hasil studi yang definitif. Sebaliknya, mayoritas mahasiswa yang masuk dalam sampel berjumlah 421 orang berasal dari angkatan 2020 dan 2021, karena telah memiliki rekam jejak akademik yang cukup untuk menentukan apakah mahasiswa tersebut lulus lebih awal, lulus tepat waktu, lulus tidak tepat waktu, atau mengalami putus studi. Meskipun demikian, tidak seluruh mahasiswa angkatan tersebut telah menyelesaikan studi, sehingga hanya mahasiswa dengan status yang dapat diidentifikasi secara pasti yang dipertahankan sebagai bagian dari sampel penelitian.

Sampel berjumlah 421 mahasiswa ini kemudian digunakan sebagai data utama untuk pelatihan model prediksi pada setiap *snapshot* semester (semester 2, 4, dan 6). Dengan hanya memasukkan mahasiswa yang telah memiliki status kelulusan final, sampel yang diperoleh memberikan dasar yang lebih akurat bagi proses pembelajaran model dalam memetakan pola risiko akademik.

Pada setiap semester digunakan jenis data yang berbeda sesuai dengan informasi akademik yang tersedia pada periode tersebut. Untuk Semester 2, data yang digunakan mencakup nilai IPS dan IPK awal, jumlah SKS yang telah diambil, jumlah mata kuliah yang gagal atau diulang, serta tingkat kehadiran. Pada tahap ini juga disertakan informasi dasar seperti asal daerah dan asal sekolah karena dapat membantu menggambarkan kondisi awal mahasiswa. Memasuki Semester 4, data yang digunakan menjadi lebih lengkap karena mencakup perkembangan perkuliahan lintas semester, seperti rata-rata SKS yang diambil, variasi tingkat kehadiran, nilai terendah yang pernah diperoleh, dan pola perubahan kinerja dari Semester 1 hingga 4. Pada Semester 6, data yang digunakan meliputi seluruh riwayat akademik hingga tahap akhir studi, termasuk perhitungan total SKS yang sudah ditempuh, jumlah mata kuliah yang pernah diulang, tren nilai dari awal perkuliahan, serta informasi semester aktif terakhir. Semua data yang digunakan hanya mencakup informasi

akademik dan latar belakang pendidikan yang relevan, tanpa menyertakan identitas pribadi seperti nama, NIM, alamat, atau data sensitif lainnya.

Selain data akademik internal, penelitian ini juga memanfaatkan data eksternal yang diperoleh melalui survei mahasiswa sebagai sumber informasi pendukung. Survei tersebut diikuti oleh 160 responden yang berasal dari angkatan 2020 hingga 2024. Seluruh responden survei merupakan bagian dari populasi mahasiswa yang sama dengan data akademik utama, sehingga keberadaan data survei tidak menambah jumlah baris data baru. Informasi hasil survei hanya digunakan untuk memperkaya atribut mahasiswa yang bersangkutan dengan menambahkan fitur tambahan pada baris data yang sudah ada. Dengan pendekatan ini, model memperoleh wawasan yang lebih luas terkait kondisi mahasiswa tanpa mengubah struktur dasar dataset maupun menimbulkan bias akibat penambahan sampel baru.

3.4 Teknik Analisis Data

Teknik analisis data dalam penelitian ini difokuskan untuk mengembangkan dan membandingkan performa model *Early Warning System* yang dibangun pada tiga titik evaluasi akademik, yaitu semester 2, semester 4, dan semester 6. Setiap model dirancang untuk memprediksi status akhir mahasiswa, seperti lulus lebih awal, lulus tepat waktu, lulus terlambat, atau dropout, dengan memanfaatkan kombinasi data akademik historis, data biodata mahasiswa, dan pada semester 6 ditambah data eksternal hasil survei terkait dukungan keluarga, kondisi finansial, dan persepsi terhadap jurusan.

Analisis dilakukan secara bertahap, dimulai dari pengolahan data (*cleaning*, *encoding*, transformasi, dan penanganan ketidakseimbangan kelas), dilanjutkan dengan proses pelatihan model menggunakan algoritma CatBoost sebagai model utama. Pada semester 4 dan semester 6, proses optimasi dilakukan lebih mendalam dengan pendekatan pencarian hiperparameter berbasis Optuna melalui skema 3-Fold Stratified Cross-Validation. Evaluasi performa model dilakukan pada data uji terpisah, sehingga pengukuran kinerja tetap objektif meskipun pemodelan menggunakan data hasil resampling.

Tahap evaluasi mengukur kualitas prediksi menggunakan sejumlah metrik, yaitu *accuracy*, *precision*, *recall*, F1-*score*, serta ROC-AUC untuk analisis tambahan pada model *binary* (khusus semester 2) dan *multiclass*. Metrik-metrik ini dipilih agar kinerja model dapat dinilai secara komprehensif, mencakup kemampuan mengenali masing-masing kelas, kesalahan prediksi yang terjadi, serta stabilitas model terhadap variasi data. Selain itu, visualisasi seperti *confusion matrix* keseluruhan dan *one-vs-rest* membantu menilai performa model terhadap setiap kelas kelulusan.

Untuk memastikan model dapat diinterpretasikan, penelitian ini menyertakan analisis berbasis *Feature Importance* dan SHAP (*Shapley Additive Explanations*). Pendekatan ini memberikan gambaran yang lebih jelas mengenai kontribusi setiap variabel, baik akademik, biodata, maupun faktor eksternal, terhadap hasil prediksi. Dengan demikian, model tidak hanya berfungsi sebagai alat prediksi, tetapi juga mendukung pemahaman konseptual mengenai faktor-faktor yang paling memengaruhi risiko kegagalan studi mahasiswa.

Seluruh rangkaian analisis dilakukan menggunakan Python karena kemampuan pustaka seperti CatBoost, scikit-learn, Optuna, dan SHAP yang mendukung pemodelan, *tuning*, visualisasi, serta interpretasi secara terstruktur. Proses ini menghasilkan model yang teroptimasi dan dapat digunakan sebagai dasar pengembangan sistem EWS yang akurat dan dapat dipertanggungjawabkan seperti pada Tabel 3.3.

Tabel 3.4 Perbandingan Bahasa Pemrograman

Sumber: [123], [124], [125]

Faktor	Python	R	MATLAB
Deskripsi	Bahasa pemrograman serbaguna untuk pengembangan aplikasi,	Bahasa pemrograman khusus untuk analisis statistik dan visualisasi data.	Digunakan terutama oleh *engineer* untuk komputasi numerik dan simulasi

Faktor	Python	R	MATLAB
	otomatisasi, dan analisis data		
Harga	Gratis (*open source*)	Gratis (*open source)*	Berbayar, membutuhkan lisensi dengan biaya cukup besar
Penggunaan	Mudah dipelajari dengan struktur sederhana, komunitas besar mendukung inovasi yang cepat	Optimal untuk statistik dan visualisasi, tetapi komunitas lebih kecil dibanding Python	Memiliki banyak fitur komputasi numerik tetapi kompleksitas lebih tinggi
Library	Banyak library seperti PyTorch, TensorFlow, dan NetworkX yang mendukung pembelajaran mesin, graf, dan analisis data.	Memiliki library statistik yang baik, tetapi variasinya lebih sedikit dibandingkan Python	Library terbatas pada fungsi khusus MATLAB dan dukungannya bergantung pada lisensi
Kinerja	Cepat dalam eksekusi, terutama dengan dukungan GPU untuk *deep learning*.	Cepat untuk analisis statistik, tetapi kurang optimal untuk *deep learning*	Lambat dalam eksekusi skala besar dibanding Python

Pemilihan Python dalam penelitian ini didasarkan pada kemampuannya dalam mengolah data terstruktur maupun semi-terstruktur, serta fleksibilitasnya dalam mengintegrasikan berbagai pustaka pendukung analisis data. Beberapa pustaka yang digunakan antara lain Scikit-learn untuk penerapan *algoritma*

machine learning dan SHAP untuk melakukan interpretasi model. Kombinasi pustaka tersebut memungkinkan proses analisis dilakukan secara efisien dan komprehensif, sekaligus mendukung penerapan prinsip Explainable AI (XAI) yang menjadi salah satu fokus utama dalam penelitian ini.