

TESIS
PENGEMBANGAN MODEL PREDIKSI PERFORMA
AKADEMIK CALON MAHASISWA MENGGUNAKAN
PENDEKATAN STACKING ENSEMBLE LEARNING
(Studi Kasus: Universitas Slamet Riyadi)



disusun oleh:

CLAUDIA SWASTIKAWATI

23.55.2506

Konsentrasi : Digital Transformation Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2026

TESIS
PENGEMBANGAN MODEL PREDIKSI PERFORMA
AKADEMIK CALON MAHASISWA MENGGUNAKAN
PENDEKATAN STACKING ENSEMBLE LEARNING
(Studi Kasus: Universitas Slamet Riyadi)

DEVELOPMENT OF A PREDICTION MODEL FOR
PROSPECTIVE STUDENT ACADEMIC PERFORMANCE
USING A STACKING ENSEMBLE LEARNING APPROACH
(Case Study: Universitas Slamet Riyadi)

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Pascasarjana
Program Studi S2 PJJ Teknik Informatika



disusun oleh:

CLAUDIA SWASTIKAWATI

23.55.2506

Konsentrasi : Digital Transformation Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2026

HALAMAN PERSETUJUAN

**PENGEMBANGAN MODEL PREDIKSI PERFORMA AKADEMIK CALON
MAHASISWA MENGGUNAKAN PENDEKATAN STACKING
ENSEMBLE LEARNING**

**DEVELOPMENT OF A PREDICTION MODEL FOR PROSPECTIVE STUDENT
ACADEMIC PERFORMANCE USING A STACKING ENSEMBLE LEARNING
APPROACH**

yang disusun dan diajukan oleh

Claudia Swastikawati

23.55.2506

Telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal, 11 Desember 2025

Dosen Pembimbing



Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

HALAMAN PENGESAHAN

PENGEMBANGAN MODEL PREDIKSI PERFORMA AKADEMIK CALON
MAHASISWA MENGGUNAKAN PENDEKATAN STACKING
ENSEMBLE LEARNING

DEVELOPMENT OF A PREDICTION MODEL FOR PROSPECTIVE STUDENT
ACADEMIC PERFORMANCE USING A STACKING ENSEMBLE LEARNING
APPROACH

yang disusun dan diajukan oleh

Claudia Swastikawati

23.55.2506

Telah dipertahankan di depan Dewan Penguji
pada tanggal 11 Desember 2025

Susunan Dewan Penguji

Nama Penguji

Dr. Kumara Ari Yuana, S.T., M.T.
NIDN. 0515067101

Dhani Ariatmanto, M.Kom., Ph.D.
NIK. 190302197

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Tanda Tangan



Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal, 11 Desember 2025

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusriul, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Claudia Swastikawati
NIM : 23.55.2506
Konsentrasi : Digital Transformation Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Pengembangan Model Prediksi Performa Akademik Calon Mahasiswa Menggunakan Pendekatan Stacking Ensemble Learning

Dosen Pembimbing Utama : Prof. Dr. Emma Utami, S.Si., M.Kom.
Dosen Pembimbing Pendamping :-

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 11 Desember 2025
Yang Menyatakan,



Claudia Swastikawati

HALAMAN PERSEMBAHAN

Segala puji dan syukur saya panjatkan kepada Tuhan Yesus Kristus, atas rahmat, penyertaan, dan kasih-Nya sehingga tesis ini dapat diselesaikan dengan baik. Tesis ini dipersembahkan kepada kedua orang tua saya, Bapak Agus Suwasji - Ibu Linda Sulistyowati, kakak dan adik kandung saya, serta tunangan saya, atas doa, pengorbanan, dan dukungan yang senantiasa diberikan. Penghargaan dan terima kasih saya sampaikan kepada pimpinan Universitas Amikom Yogyakarta, para Dosen Pembimbing dan Penguji, serta seluruh pihak yang telah memberikan bimbingan, arahan, dan dukungan selama proses penelitian dan penyusunan tesis ini. Semoga tesis ini dapat memberikan kontribusi yang bermanfaat bagi pengembangan ilmu pengetahuan dan dunia akademik.



KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan penyertaan-Nya, sehingga tesis yang berjudul “Pengembangan Model Prediksi Performa Akademik Calon Mahasiswa Menggunakan Pendekatan Stacking Ensemble Learning” dapat diselesaikan dengan baik sebagai salah satu persyaratan untuk menyelesaikan pendidikan pada Program Studi S2 PJJ Informatika. Penulis menyadari bahwa penyusunan tesis ini tidak terlepas dari bantuan, bimbingan, dan dukungan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. M. Suyanto, M.M., selaku Rektor Universitas Amikom Yogyakarta, atas dukungan dan fasilitas yang diberikan selama penulis menempuh pendidikan.
2. Ibu Prof. Dr. Kusriani, M.Kom., selaku Dekan Fakultas Ilmu Komputer Universitas Amikom Yogyakarta.
3. Bapak Dhani Ariatmanto, M.Kom., Ph.D., selaku Ketua Program Studi S2 PJJ Informatika Universitas Amikom Yogyakarta, sekaligus sebagai Dosen Penguji, atas arahan, masukan, dan dukungan yang diberikan.
4. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom., selaku Dosen Pembimbing Utama, atas bimbingan, perhatian, saran, serta ilmu yang telah diberikan dengan penuh kesabaran selama proses penyusunan tesis ini.
5. Bapak Dr. Kumara Ari Yuana, S.T., M.T., selaku Dosen Penguji, atas saran dan masukan yang konstruktif demi penyempurnaan tesis ini.

6. Kedua orang tua penulis (Bapak Agus Suwasji dan Ibu Linda Sulistyowati) serta kakak dan adik kandung penulis, atas doa, kasih, pengorbanan, dan dukungan yang senantiasa menguatkan penulis selama menempuh studi.
7. Pasangan penulis, atas kesabaran, pengertian, dan dukungan moral yang diberikan selama proses penyelesaian tesis ini.
8. Sahabat-sahabat, rekan kerja di Universitas Slamet Riyadi, serta teman-teman mahasiswa S2 PJJ Informatika (MTI PJJ Angkatan 10), atas kebersamaan, semangat, dan motivasi yang diberikan.

Penulis menyadari bahwa tesis ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan demi penyempurnaan di masa mendatang. Semoga tesis ini dapat memberikan manfaat dan kontribusi bagi pengembangan ilmu pengetahuan, khususnya di bidang informatika.

Yogyakarta,

Penulis

DAFTAR ISI

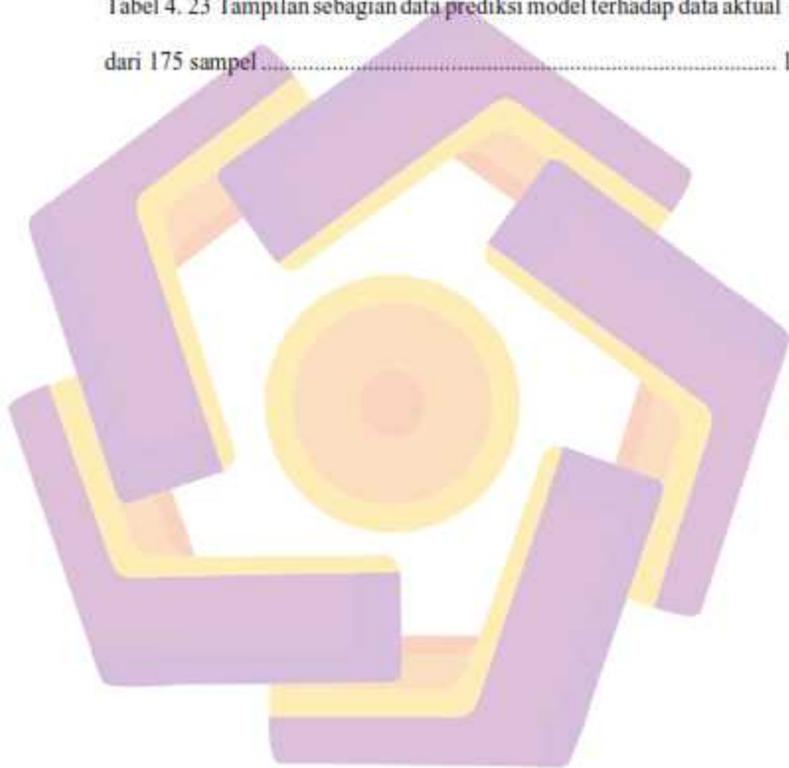
HALAMAN JUDUL	1
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSEMBAHAN	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xii
INTISARI	xiv
ABSTRACT	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	7
1.3. Batasan Masalah	7
1.4. Tujuan Penelitian.....	9
1.5. Manfaat Penelitian	9
BAB II TINJAUAN PUSTAKA.....	11
2.1. Tinjauan Pustaka.....	11
2.2. Keaslian Penelitian.....	21
2.3. Landasan Teori.....	31
BAB III METODE PENELITIAN	47
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	47
3.2. Metode Pengumpulan Data.....	47

3.3.	Metode Analisis Data.....	47
3.4.	Alur Penelitian.....	49
3.5.	Justifikasi Penelitian	60
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....		64
4.1.	Pendekatan Penelitian.....	64
4.2.	Pemahaman Data.....	65
4.3.	<i>Preprocessing Data</i>	69
4.4.	<i>Split Data</i>	77
4.5.	<i>Modeling Machine Learning</i>	82
4.5.1	<i>Pendekatan Klasifikasi Multi-Kelas</i>	82
4.5.2	<i>Hyperparameter Tuning</i>	83
4.5.3	<i>Pemodelan Level 0 (Base Model)</i>	84
4.5.4	<i>Pembentukan Meta-Feature</i>	87
4.5.5	<i>Pemodelan Level 1 (Meta Model)</i>	96
4.5.6	<i>Evaluasi Model</i>	99
4.5.7	<i>Testing</i>	105
4.5.8	<i>Perbandingan Model</i>	120
4.5.9	<i>Analisis Interpretabilitas Model</i>	123
4.6.	Evaluasi Relevansi Prediksi terhadap Data Aktual.....	130
BAB V		134
5.1	Kesimpulan.....	134
5.2	Saran.....	136
Daftar Pustaka.....		140
LAMPIRAN.....		146

DAFTAR TABEL

Tabel 2. 1 Matriks literatur review.....	21
Tabel 4. 1 Dataset awal.....	66
Tabel 4. 2 Deskripsi fitur.....	68
Tabel 4. 3 Variabel <i>independent</i>	76
Tabel 4. 4 Variabel <i>dependent</i>	76
Tabel 4. 5 Komposisi pembagian data <i>training</i> dan data <i>testing</i>	79
Tabel 4. 6 Distribusi Kelas Data Latih Sebelum Penyeimbangan.....	80
Tabel 4. 7 Distribusi Kelas Data Latih Setelah SMOTE.....	81
Tabel 4. 8 Parameter yang digunakan dalam <i>hyperparameter tuning</i>	84
Tabel 4. 9 <i>Hyperparameter</i> terbaik untuk model dasar.....	85
Tabel 4. 10 Laporan klasifikasi model dasar.....	86
Tabel 4. 11 Ringkas Hasil pembentukan <i>meta-feature</i> dari kombinasi model.....	90
Tabel 4. 12 Parameter grid untuk <i>hyperparameter tuning</i> model meta.....	97
Tabel 4. 13 <i>Output</i> 20 prediksi pertama meta-model versi Train Set.....	99
Tabel 4. 14 <i>Classification report</i> pelatihan <i>base model</i> dan meta model.....	100
Tabel 4. 15 <i>Output</i> 20 prediksi meta-model versi Test Set.....	107
Tabel 4. 16 Metrik uji <i>stacking</i> model 3 model.....	111
Tabel 4. 17 Metrik uji <i>stacking</i> model RF-KNN.....	114
Tabel 4. 18 Metrik uji <i>stacking</i> model RF-SVM.....	115
Tabel 4. 19 Metrik uji <i>stacking</i> model KNN-SVM.....	117

Tabel 4. 20 Perbandingan hasil evaluasi model (train set dan test set)...	121
Tabel 4. 21 Feature Importance Meta-Model Stacking.....	126
Tabel 4. 22 Persentase kontribusi berdasarkan feature importances dari meta-model.....	128
Tabel 4. 23 Tampilan sebagian data prediksi model terhadap data aktual 15 dari 175 sampel	132



DAFTAR GAMBAR

Gambar 3. 1 Alur Penelitian <i>Stacking Ensemble</i>	49
Gambar 4. 1 Visualisasi <i>Cramér's V Correlation Matrix</i> Terhadap Status_lulus	67
Gambar 4. 2 Tipe data	71
Gambar 4. 3 <i>Output encoding process</i>	72
Gambar 4. 4 Korelasi fitur dengan target Status_lulus	73
Gambar 4. 5 Dataset final	75
Gambar 4. 6 Split data latih dan data uji	78
Gambar 4. 8 Pengecekan kesesuaian indeks penggabungan <i>meta-features</i>	94
Gambar 4. 9 <i>Clasification report</i> RF-KNN-SVM	100
Gambar 4. 10 <i>Clasification report</i> RF-KNN	100
Gambar 4. 11 <i>Clasification report</i> RF-SVM	100
Gambar 4. 12 <i>Clasification report</i> KNN-SVM	100
Gambar 4. 13 <i>Confusion matrix</i> meta model kombinasi 3 model dasar .	103
Gambar 4. 14 <i>Confusion matrix</i> meta model kombinasi model RF-KNN	103
Gambar 4. 15 <i>Confusion matrix</i> meta model kombinasi model KNN-SVM	103
Gambar 4. 16 <i>Confusion matrix</i> meta model kombinasi model RF-SVM	103

Gambar 4. 17 Log info dataset.....	106
Gambar 4. 18 <i>Confusion matrix</i> uji model stacking kombinasi RF-KNN-SVM	109
Gambar 4. 19 <i>Confusion matrix</i> uji model stacking kombinasi RF-KNN	109
Gambar 4. 20 <i>Confusion matrix</i> uji model stacking kombinasi RF-SVM	109
Gambar 4. 21 <i>Confusion matrix</i> uji model stacking kombinasi KNN-SVM	109
Gambar 4. 22 Kurva ROC kombinasi RF-KNN-SVM	118
Gambar 4. 23 Kurva ROC kombinasi RF-KNN	118
Gambar 4. 24 Kurva ROC kombinasi RF-SVM.....	119
Gambar 4. 25 Kurva ROC kombinasi KNN-SVM	119
Gambar 4. 26 <i>Feature importance</i> pada model final <i>stacking</i> XGBoost 3 model.....	124
Gambar 4. 27 <i>Feature importance</i> pada model final <i>stacking</i> XGBoost RF-KNN	125
Gambar 4. 28 <i>Feature importance</i> pada model final <i>stacking</i> XGBoost RF-SVM	125
Gambar 4. 29 <i>Feature importance</i> pada model final <i>stacking</i> XGBoost KNN-SVM	126
Gambar 4. 30 Grafik visualisasi relevansi data.....	130

INTISARI

Penelitian ini berjudul Pengembangan Model Prediksi Performa Akademik Calon Mahasiswa Menggunakan Pendekatan *Stacking Ensemble Learning*. Penelitian ini bertujuan mengembangkan model prediksi performa akademik calon mahasiswa sejak tahap penerimaan dengan memanfaatkan data historis pendaftaran. Variabel penelitian dibatasi pada fitur akademik, *meta-features*, sosioekonomi, dan demografi yang bersumber dari data Penerimaan Mahasiswa Baru (PMB) dan PDDikti Universitas Slamet Riyadi. Metode yang digunakan adalah pembelajaran mesin dengan pendekatan *stacking ensemble learning*, menggunakan *Random Forest*, *Support Vector Machine*, dan *K-Nearest Neighbors* sebagai model dasar, serta XGBoost sebagai meta-model. Untuk mengatasi ketidakseimbangan kelas pada klasifikasi multikelas, diterapkan teknik *Synthetic Minority Over-sampling Technique* (SMOTE).

Hasil penelitian menunjukkan bahwa model *stacking ensemble* memberikan performa prediksi yang lebih stabil dan unggul dibandingkan model tunggal maupun kombinasi dua model dasar. Kombinasi tiga model dasar menghasilkan kinerja terbaik berdasarkan metrik akurasi, presisi, *recall*, *F1-score*, dan AUC-ROC. Berdasarkan analisis *feature importance* pada meta-model, fitur akademik menjadi kontributor dominan dengan persentase antara 47-50%, diikuti oleh *meta-features* sebesar 17-21%, fitur sosioekonomi sebesar 18-20%, dan fitur demografi sebesar 12-14%. Temuan ini menunjukkan bahwa indikator akademik awal dan keluaran prediktif dari model dasar memiliki peran utama dalam memetakan performa akademik mahasiswa sejak tahap pendaftaran.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa pendekatan *stacking ensemble learning* efektif dalam membangun model prediksi performa akademik calon mahasiswa secara multikelas. Model yang dikembangkan mampu melakukan generalisasi dengan baik serta merepresentasikan kondisi kelulusan aktual mahasiswa, sehingga berpotensi digunakan sebagai sistem pendukung keputusan dalam seleksi penerimaan mahasiswa baru dan perencanaan pembinaan akademik berbasis data.

Kata kunci: *stacking ensemble*, prediksi performa akademik, *feature importance*, SMOTE, XGBoost

ABSTRACT

This study is entitled Development of an Academic Performance Prediction Model for Prospective Students Using a Stacking Ensemble Learning Approach. The objective of this research is to develop a model for predicting the academic performance of prospective students from the admission stage by utilizing historical enrollment data. The research variables are limited to academic features, meta-features, socioeconomic features, and demographic features derived from the New Student Admission (PMB) data and the PDDikti database of Universitas Slamet Riyadi. The analytical method employed is machine learning using a stacking ensemble learning approach, with Random Forest, Support Vector Machine, and K-Nearest Neighbors as base learners, and XGBoost as the meta-model. To address class imbalance in multiclass classification, the Synthetic Minority Over-sampling Technique (SMOTE) is applied.

The results indicate that the stacking ensemble model achieves more stable and superior predictive performance compared to single models and two-model combinations. The combination of three base learners produces the best performance based on accuracy, precision, recall, F1-score, and AUC-ROC metrics. Based on the meta-model feature importance analysis, academic features are the dominant contributors with a contribution range of 47-50%, followed by meta-features at 17-21%, socioeconomic features at 18-20%, and demographic features at 12-14%. These findings demonstrate that early academic indicators and predictive outputs from base learners play a critical role in mapping students' academic performance from the admission stage.

Based on these findings, it can be concluded that the stacking ensemble learning approach is effective in developing a multiclass academic performance prediction model for prospective students. The proposed model demonstrates good generalization capability and accurately represents actual student graduation outcomes, indicating its potential application as a decision support system for student admission selection and data-driven academic development planning.

Keywords: *stacking ensemble, academic performance prediction, feature importance, SMOTE, XGBoost*

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Keberhasilan siswa menjadi salah satu tujuan utama lembaga pendidikan, yang diukur melalui kinerja akademik, atau seberapa baik siswa memenuhi standar yang ditetapkan oleh lembaga tersebut. Sehingga, prediksi kinerja akademik mahasiswa dianggap sebagai alat penting untuk mendukung berbagai keputusan, seperti penerimaan mahasiswa, kelulusan, dan dukungan pendidikan yang disesuaikan berdasarkan data mahasiswa. Kinerja mahasiswa juga merupakan indikator penting dalam mengukur efektivitas institusi dan merupakan faktor penting dalam keberhasilan mahasiswa di masa depan, khususnya di bidang Pendidikan (Nachouki & Naaj, 2022).

Selain itu, perguruan tinggi tidak hanya berfokus pada peningkatan prestasi mahasiswa dan kualitas program pendidikannya, tetapi juga pada upaya memastikan mahasiswa dapat menyelesaikan studi tepat waktu. Kelulusan tepat waktu membuka peluang kerja lebih besar, sehingga perhatian terhadap lama studi menjadi penting karena mencerminkan pencapaian akademik mahasiswa (Etriyanti et al., 2020). Rata-rata lama studi juga berpengaruh pada reputasi institusi, mengingat hal tersebut dijadikan salah satu kriteria penilaian akreditasi perguruan tinggi oleh Badan Akreditasi Nasional Perguruan Tinggi (Zainuddin, 2018). Dengan demikian, keberhasilan perguruan tinggi tidak hanya ditentukan oleh proses pembelajaran, tetapi juga oleh kualitas input mahasiswa. Oleh karena itu,

perguruan tinggi saat ini tidak hanya fokus pada peningkatan prestasi mahasiswa dan kualitas program pendidikannya, tetapi juga pada penjarangan awal mahasiswa.

Proses Penerimaan Mahasiswa Baru (PMB) menghasilkan data dalam jumlah besar yang terus bertambah setiap tahunnya. Tantangan yang dihadapi adalah bagaimana data tersebut dapat diolah secara optimal menjadi informasi yang berguna dan membantu pengambil keputusan dalam menganalisis serta membuat keputusan yang lebih tepat. Dalam hal ini, prediksi kinerja siswa sejak dini melalui data PMB dapat membantu universitas mengambil tindakan preventif, seperti mempertimbangkan penerimaan mahasiswa yang berpotensi tidak menyelesaikan studi tepat waktu, atau sebagai bahan pertimbangan dalam penerimaan jalur beasiswa (Sejati et al., 2022). Dalam upaya mengatasi hal ini, penelitian dilakukan untuk mengidentifikasi pola data PMB serta mengolah dan menangani permasalahan data, sehingga dapat membangun model prediksi performa calon mahasiswa yang lebih optimal dan selaras dengan karakteristik data, dengan tetap merujuk pada temuan-temuan penelitian terdahulu.

Penelitian sebelumnya menggunakan berbagai algoritma pembelajaran untuk melakukan prediksi performa mahasiswa, seperti K-Nearest Neighbors (KNN), C4.5, Support Vector Machine (SVM), J48, NNge, MLP, dan Long Short-Term Memory (LSTM). Klasifikasi Random Forest (RF) meningkatkan akurasi 4,19% dengan bertambahnya jumlah data (Nachouki & Naaj, 2022). Artificial Neural Network (ANN) menunjukkan dominasi variabel akademik, demografi, dan faktor keluarga sebagai prediktor kinerja siswa (Alsariera, Yazan A. et al., 2022), (Baashar Yahia, 2022). Teknik *ensemble* EMT menjadi metode unggul

dibandingkan banyak metode lainnya (Almasri, 2019). Pada penelitian lain teknologi big data telah diterapkan untuk mengatasi masalah akurasi, yakni algoritma SVM menunjukkan hasil akurasi tertinggi (Ouatic et al., 2022). Penelitian serupa juga dilakukan oleh (Abdullah et al., 2022) yang memanfaatkan algoritma SVM untuk memprediksi waktu kelulusan mahasiswa. Hasil penelitian tersebut menunjukkan bahwa SVM dapat diimplementasikan secara efektif untuk klasifikasi data akademik, sehingga memperkuat bukti bahwa algoritma ini memiliki potensi besar dalam konteks prediksi performa mahasiswa. Meskipun beberapa algoritma telah diusulkan dan dapat berkontribusi signifikan dalam prediksi performa akademik, masih terdapat beberapa permasalahan. Pada algoritma SVM misalnya, terdapat ketidakcocokan untuk dataset besar karena waktu pelatihan yang lama dan sensitivitas terhadap noise, serta penggunaan sampel kecil dapat mengurangi kemampuan generalisasi. Masalah serupa pada algoritma Random Forest dan *K-Nearest Neighbors*, di mana proses optimasi memakan waktu. Selain itu, terdapat risiko *overfitting*, tantangan terkait skalabilitas dan efisiensi pada model lainnya.

Untuk mengatasi keterbatasan ini, sesuai dengan arahan dalam melakukan experiment sebelumnya tentang algoritma yang digunakan dalam jurnal (Butt et al., 2023; Fernández-García et al., 2020; Fernandez-Garcia et al., 2021; Sahlaoui et al., 2021), algoritma *Random Forest*, *Support Vector Machine (SVM)* dan *K-Nearest Neighbors* dalam konteks prediksi performa akademik mahasiswa, ada beberapa langkah penting yang dapat diikuti. Pertama, fokus pada praproses data dengan menerapkan teknik menangani redundansi, korelasi, fitur, serta normalisasi dan pengkodean data, untuk memastikan kualitas input data yang optimal. Selanjutnya,

seleksi fitur dengan metode statistik dapat digunakan untuk mempertahankan fitur-fitur yang relevan, sehingga mengurangi risiko *overfitting* dan meningkatkan efisiensi algoritma. Menerapkan teknik SMOTE untuk memastikan representasi data yang lebih seimbang. Evaluasi model menggunakan metrik seperti akurasi, presisi, recall, F1-score, dan AUC-ROC sangat penting untuk mengukur efektivitas performa model dalam prediksi. Untuk meningkatkan keandalan, integrasikan pendekatan ensemble dengan menggabungkan Random Forest, SVM, dan KNN. Sehingga, model prediksi dapat dikembangkan untuk tahap sebelum masuk universitas, guna mengidentifikasi performa calon mahasiswa.

Algoritma Random Forest (RF), Support Vector Machine (SVM), dan K-Nearest Neighbors (KNN) dipilih karena ketiga model tersebut dikenal sebagai model yang kuat dalam menangani data dengan kompleksitas tinggi dan memberikan hasil prediksi yang akurat. RF, sebagai algoritma berbasis ensemble, mampu mengatasi masalah *overfitting* dan menghasilkan prediksi yang lebih stabil dengan menggunakan kombinasi dari banyak pohon keputusan, serta model ideal untuk digunakan dalam memprediksi performa akademik mahasiswa (Bujang et al., 2021). Hal ini menjadikannya cocok untuk dataset yang heterogen, seperti data akademik mahasiswa. Di sisi lain, SVM efektif dalam menangani data yang tidak dapat dipisahkan secara linear dengan memanfaatkan kernel *trick* untuk memetakan data ke dimensi yang lebih tinggi, sehingga menghasilkan *hyperplane* pemisah yang optimal (Fernandez-Garcia et al., 2021). Sementara itu, KNN (K-Nearest Neighbors) dipilih karena kesederhanaannya dan fleksibilitasnya dalam menangani dataset dengan pola yang tidak beraturan. KNN banyak digunakan untuk data

berskala kecil hingga menengah dengan distribusi yang kompleks. Selain itu, dengan menggunakan metrik jarak seperti *Euclidean distance* atau *Manhattan distance*, KNN mudah menangkap pola lokal dalam data yang seringkali sulit diidentifikasi oleh model lainnya. Kelebihan ini menjadikan KNN sebagai model yang baik dalam analisis prediktif, terutama ketika dikombinasikan dengan metode ensemble (Bujang et al., 2021; Fernández-García et al., 2020).

Banyak penelitian sebelumnya telah menunjukkan algoritma RF, SVM, dan KNN memberikan hasil akurasi terbaik dalam memprediksi, termasuk performa akademik. Dalam prediksi keberhasilan akademik, studi-studi yang menggunakan ketiga model ini menunjukkan keunggulan dalam hal keandalan dan kemampuan generalisasi. Kombinasi kemampuan RF dalam menangani data yang kompleks serta keunggulan SVM dalam membedakan kelas dengan margin yang jelas, dan KNN yang dapat dengan mudah menangkap pola lokal dalam data menjadikan ketiga algoritma sebagai pilihan tepat.

Penelitian fokus pada pengembangan model dengan mempertimbangkan penggunaan model *Stacking Ensemble Learning* dalam meningkatkan performa kombinasi model Random Forest, KNN, dan SVM dalam memprediksi. Pendekatan ini dapat memanfaatkan kekuatan setiap algoritma dasar dan mengurangi risiko *overfitting*, sehingga menghasilkan prediksi akhir yang lebih akurat dan efisien dalam menangani data yang kompleks dan bervariasi (Butt et al., 2023).

Model XGBoost dipilih sebagai meta-model dalam *Stacking Ensemble* karena kemampuannya yang unggul dalam menangani data dengan skala besar dan beragam, serta efisiensinya dalam komputasi. Selain itu, XGBoost memiliki

kemampuan menangkap interaksi antar fitur dengan lebih baik, berkat proses gradient boosting yang iterative (Sahlaoui et al., 2021). Model ini juga menawarkan fleksibilitas melalui berbagai parameter yang dapat dioptimalkan, sehingga cocok untuk menggabungkan prediksi dari berbagai model dasar.

Dalam penelitian ini, hiperparameter setiap model dioptimalkan secara terpisah untuk memastikan performa terbaik. Untuk menghindari *overfitting* dan meningkatkan presisi, dilakukan optimasi hiperparameter melalui pencarian grid sederhana, termasuk pengaturan parameter *learning rate*, *n_estimators*, dan metrik evaluasi pada XGBoost. *Learning rate* dan *epoch* berperan penting dalam proses pelatihan meta-model XGBoost untuk meningkatkan performa prediksi kelulusan calon mahasiswa.

Dengan mengadopsi pendekatan *Stacking Ensemble Learning* yang memanfaatkan XGBoost sebagai meta-model, penelitian ini diharapkan memberikan kontribusi baru dalam pengembangan ilmu pengetahuan, khususnya dalam penerapan model prediktif untuk menilai potensi kelulusan mahasiswa sejak tahap awal pendaftaran. Melalui integrasi beberapa algoritma dasar seperti *Random Forest*, *K-Nearest Neighbor*, dan *Support Vector Machine*, model yang dikembangkan tidak hanya berfokus pada peningkatan akurasi, tetapi juga pada stabilitas dan kemampuan dalam mengenali calon mahasiswa yang berisiko tidak lulus tepat waktu. Selain itu, penelitian ini turut berkontribusi dalam mengidentifikasi atribut-atribut penting yang paling relevan dalam memprediksi performa akademik calon mahasiswa. Hasil yang diperoleh diharapkan dapat dimanfaatkan oleh lembaga pendidikan untuk meningkatkan kualitas seleksi

penerimaan mahasiswa baru, menyusun strategi pembinaan akademik yang lebih terarah, serta mendukung pengambilan keputusan administratif secara lebih objektif dan berbasis data.

1.2. Rumusan Masalah

- a. Apa saja atribut demografis, akademik, dan sosioekonomi yang paling berpengaruh terhadap kemampuan model prediksi dalam menentukan status kelulusan mahasiswa lulus tepat waktu, terlambat, atau dikeluarkan?
- b. Bagaimana perbandingan performa model *Stacking Ensemble* dengan XGBoost sebagai meta-model dalam meningkatkan akurasi dan evaluasi prediksi, baik pada kombinasi dua model dasar (SVM dan RF, RF dan KNN, atau SVM dan KNN) maupun tiga model dasar (SVM, RF, dan KNN)?
- c. Sejauh mana hasil prediksi model *stacking* berbasis data historis pendaftaran dapat merepresentasikan status kelulusan mahasiswa berdasarkan data aktual perguruan tinggi?

1.3. Batasan Masalah

- a. Penelitian ini terbatas pada data yang berasal dari dua sumber data, yaitu laman PMB empat belas program studi tingkat Strata 1 (S1) Universitas Slamet Riyadi pada tahun 2019 dan dari data pelaporan PDDikti mahasiswa lulus / *dropout* yang dapat diakses melalui laman neofeeder.
- b. Atribut yang digunakan pada penelitian ini berjumlah 16 atribut dan di kelompokkan menjadi tiga:
 - Atribut Demografis diantaranya provinsi tinggal, agama, jenis kelamin,

- Atribut Akademik seperti periode dan gelombang pendaftaran, prodi pilihan, status pindahan, jurusan SLTA, nilai rata-rata SKHU, dan *score* CBT (poin terima tes)
 - Atribut Sosioekonomi diantaranya pekerjaan ayah, pekerjaan ibu, gaji ayah, gaji ibu, jumlah anak kandung, status kelulusan pada PDDikti.
- c. Penelitian ini berfokus membatasi penggunaan algoritma pembelajaran mesin yang telah terbukti mencapai akurasi tinggi dari penelitian-penelitian sebelumnya yaitu Random Forest, SVM, KNN, dan penerapan XGBoost sebagai model *stacking ensemble*.
- d. Penelitian ini akan memusatkan perhatian pada prediksi performa calon mahasiswa baru, dengan *output* klasifikasi lulus tepat waktu, tidak tepat waktu, atau dikeluarkan berdasarkan data relevan yang ada.
- e. Tidak memasukkan aspek Non-Akademik yang mendalam, hanya melibatkan variabel yang kaitanya dengan lingkungan keluarga seperti pekerjaan orangtua, penghasilan orangtua, dan jumlah anak kandung.
- f. Platform pengolahan data yakni seluruh analisis dan pemodelan akan dilakukan menggunakan Google Colab dan bahasa pemrograman Python, dengan pemanfaatan pustaka seperti Pandas, NumPy, Scikit-learn, Seaborn, Joblib, dan XGBoost.
- g. Model prediksi akan dievaluasi menggunakan metrik akurasi, presisi, *recall*, *F1-score*, dan AUC-ROC dalam penelitian ini.

1.4. Tujuan Penelitian

- a. Menganalisis dan mengidentifikasi atribut demografis, akademik, dan sosioekonomi yang paling berpengaruh terhadap prediksi status kelulusan mahasiswa (lulus tepat waktu, lulus terlambat, atau dikeluarkan).
- b. Menganalisis dan membandingkan performa model *Stacking Ensemble* dengan XGBoost sebagai meta-model pada kombinasi dua model dasar (SVM dan RF, RF dan KNN, atau SVM dan KNN) serta tiga model dasar (SVM, RF, dan KNN) dalam stabilitas prediksi kelulusan mahasiswa berdasarkan atribut pendaftaran.
- c. Mengevaluasi relevansi hasil prediksi model *stacking* terhadap data aktual kelulusan mahasiswa di perguruan tinggi, untuk melihat sejauh mana model yang dikembangkan mampu merepresentasikan kenyataan di lapangan secara kontekstual dan valid.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini dapat disampaikan dalam dua ranah utama, yaitu pengembangan keilmuan dan penerapan praktis di lingkungan pendidikan tinggi:

a. Manfaat Teoretis

Penelitian ini memperkaya literatur di bidang data *science* dan pembelajaran mesin, khususnya dalam penerapan pendekatan *stacking ensemble* untuk klasifikasi multi-kelas dalam konteks pendidikan. Hasil penelitian ini juga menjadi bukti empiris atas efektivitas pendekatan *ensemble* dalam meningkatkan performa prediksi dibandingkan model tunggal, serta

memberikan gambaran tentang faktor-faktor pendaftaran calon mahasiswa yang paling berpengaruh terhadap hasil akhir studi mereka.

b. Manfaat Praktis

Dari sisi penerapan, hasil penelitian ini dapat menjadi referensi dalam mengembangkan sistem pendukung keputusan berbasis data historis pendaftaran mahasiswa. Dengan mengidentifikasi atribut penting dan membuktikan relevansi hasil prediksi terhadap data aktual kelulusan, model yang dikembangkan berpotensi digunakan oleh perguruan tinggi untuk menyaring, memetakan, dan memantau calon mahasiswa sejak awal. Hal ini dapat membantu institusi dalam merancang kebijakan akademik preventif dan strategi pembinaan yang lebih tepat sasaran.



BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Dalam beberapa tahun terakhir, penggunaan algoritma pembelajaran mesin (machine learning) untuk prediksi kinerja mahasiswa telah banyak diteliti, baik dalam konteks penerimaan mahasiswa baru maupun prediksi kelulusan. Berbagai algoritma seperti Random Forest (RF), Support Vector Machine (SVM), dan model *ensembel* telah diimplementasikan untuk memprediksi performa akademik dengan hasil yang bervariasi. Berikut ini adalah beberapa ulasan jurnal terkait penelitian terdahulu dan relevansinya terhadap prediksi performa mahasiswa yang digunakan sebagai acuan dalam hal data dan metode.

Dalam penelitian (Fernandez-Garcia et al., 2021) yang berjudul **“A Real-Life Machine Learning Experience for Predicting University Dropout at Different Stages Using Academic Data”** menggunakan data akademik untuk memprediksi risiko *dropout* mahasiswa di berbagai tahap pendidikan tinggi. Mulai dari prediksi proses pendaftaran hingga pada tahapan setiap semester sampai mahasiswa keluar dari proses studi. Penelitian tersebut mengadopsi model seperti Random Forest, SVM, dan Gradient Boosting, yang dikombinasikan dengan teknik *ensemble* untuk meningkatkan akurasi dan mengurangi kesalahan prediksi. Pendekatan *Stacking Ensemble* memungkinkan penggabungan kekuatan dari beberapa model dasar dengan memanfaatkan meta-model. Teknik ini telah terbukti unggul dalam menangani dataset yang kompleks dan beragam. Hasil dari penelitian

Fernández-García et al. menunjukkan bahwa *Stacking Ensemble* mampu memberikan hasil prediksi yang lebih akurat dengan menggunakan data akademik. Pendekatan yang diuraikan relevan dengan penelitian yang diusulkan yaitu yang bertujuan memprediksi status kelulusan mahasiswa. Sementara itu, (Bujang et al., 2021) menyajikan analisis prediktif serupa menggunakan beberapa algoritma dasar diantaranya terdapat pula RF, KNN, dan SVM untuk memprediksi nilai akhir mahasiswa. Penelitian ini menyoroti pentingnya pengolahan dataset yang tidak seimbang dengan teknik SMOTE, yang secara signifikan meningkatkan akurasi prediksi. Namun pada penelitian saat ini akan menggunakan tiga model dasar, SVM, RF, dan KNN dengan XGBoost sebagai meta-model dalam kerangka *Stacking Ensemble*. Kombinasi ini diharapkan dapat mengoptimalkan performa prediksi, memanfaatkan kemampuan SVM dalam menangani data non-linear, KNN dalam mengenali pola lokal, serta stabilitas Random Forest dalam mengatasi ketidakseimbangan data. Dengan pendekatan ini, penelitian diharapkan dapat memperluas implementasi *Stacking Ensemble* dalam mendukung pengambilan keputusan strategis khususnya dalam proses seleksi dan evaluasi mahasiswa baru.

Dalam konteks pengembangan model prediksi kinerja akademik, penelitian (Sahlaoui et al., 2021) yang berjudul "*Predicting and Interpreting Student Performance Using Ensemble Models and Shapley Additive Explanations*" menggunakan algoritma XGBoost dan *Extra Trees* (ET) dengan pendekatan gabungan yang mengombinasikan data dan algoritma. Langkah ini dilaporkan mampu meningkatkan akurasi prediksi sebesar 20,3% dibandingkan implementasi model pada penelitian sebelumnya. Salah satu teknik yang digunakan adalah

synthetic minority oversampling technique (SMOTE), yang berfungsi menyeimbangkan data dengan meningkatkan jumlah contoh pada kelas minoritas. Penelitian menggunakan nilai SHAP (Shapley Additive Explanations) dalam memungkinkan peneliti untuk lebih memahami kontribusi setiap variabel dalam model prediksi, menjadikannya alat yang sangat efektif dalam menganalisis faktor-faktor yang mempengaruhi kinerja akademik siswa.

Berdasarkan penelitian (Sejati et al., 2022) dalam penelitiannya yang berjudul "**Studi Komparasi Naive Bayes, K-Nearest Neighbor, dan Random Forest untuk Prediksi Calon Mahasiswa yang Diterima atau Mundur**" melakukan studi komparasi antara tiga algoritma pembelajaran mesin: Naive Bayes, K-Nearest Neighbor (KNN), dan Random Forest (RF) dalam memprediksi penerimaan calon mahasiswa, dengan akurasi RF mencapai 73,61%. Hal ini menunjukkan bahwa RF memberikan kinerja yang baik dalam klasifikasi yang melibatkan data kompleks seperti profil mahasiswa baru. Penelitian ini juga menunjukkan bahwa pengoptimalan *hyperparameter* dapat meningkatkan akurasi, namun memerlukan waktu komputasi yang lebih lama pada RF. Penggunaan algoritma RF ini relevan untuk penelitian prediksi kelulusan tepat waktu, karena RF telah terbukti efisien dalam mengidentifikasi pola data yang kompleks dalam menganalisis faktor kelulusan mahasiswa. Peneliti menyarankan agar penelitian selanjutnya dapat berfokus pada pengembangan metode optimasi yang lebih efisien untuk mengurangi waktu pemrosesan RF.

Dua penelitian membahas mengenai model *ensemble* dan kombinasi algoritma. (Karalar et al., 2021) dalam penelitiannya yang berjudul "**Predicting**

students at risk of academic failure using ensemble model during pandemic in a distance learning system” menunjukkan bahwa model *ensemble* seperti *Extra Trees* (ET), RF, dan *Logistic Regression* (LR) dapat menghasilkan prediksi yang akurat mengenai mahasiswa berisiko selama pembelajaran jarak jauh, dengan spesifisitas RF mencapai 90,34%. Sementara itu, penelitian Butt et al. (2023) yang berjudul **“Performance Prediction of Students in Higher Education using Multi-model Ensemble Approach”** membuktikan bahwa model *stacking ensemble* lebih unggul dibandingkan model tunggal dalam memprediksi kinerja siswa di pendidikan tinggi. Dengan memanfaatkan data historis mahasiswa, penelitian ini menekankan pentingnya metode *ensemble* multi-model dalam analitik pembelajaran. Penelitian saat ini mengadopsi saran dari studi sebelumnya dengan mengombinasikan model SVM dan RF yang dioptimalkan menggunakan metode *stacking ensemble*. Selain itu, ruang lingkup data yang digunakan diperluas, mencakup data historis mahasiswa dari pendaftaran hingga kelulusan, serta faktor internal, ekonomi, dan keluarga. Pendekatan ini selaras dengan rekomendasi penelitian terdahulu yang menyarankan optimasi model dan penggunaan data lebih luas untuk meningkatkan akurasi prediksi.

Menurut penelitian (Yağcı, 2022) dalam penelitiannya yang berjudul **“Educational data mining: prediction of students’ academic performance using machine learning algorithms”** menggunakan RF, KNN, dan SVM untuk memprediksi nilai akhir mahasiswa berdasarkan nilai ujian tengah semester. Penelitian ini menekankan pentingnya data akademik awal dalam memprediksi hasil akhir. Hasil penelitian menunjukkan bahwa RF dan SVM memiliki akurasi

yang lebih tinggi dibandingkan KNN. Penelitian ini relevan dengan penelitian prediksi kelulusan karena faktor-faktor akademik seperti Nilai IPK awal, prestasi ujian, dan kehadiran di kelas juga dapat berfungsi sebagai prediktor utama apakah mahasiswa akan lulus tepat waktu atau terlambat. Penelitian selanjutnya disarankan untuk memasukkan parameter tambahan sebagai variabel input.

Penelitian lainnya yang menggunakan RF dan SVM yaitu (Gusnina et al., 2022) pada penelitiannya yang berjudul **“Student Performance Prediction in Sebelas Maret University Based on the Random Forest Algorithm”** menunjukkan bahwa penggunaan metode seleksi fitur seperti Information Gain dapat meningkatkan akurasi prediksi. Faktor-faktor seperti indeks prestasi, pekerjaan orang tua, jumlah tanggungan keluarga, biaya sekolah, dan nilai ujian sekolah menengah ditemukan sangat mempengaruhi kinerja akademik mahasiswa. Penggunaan metode Information Gain meningkatkan akurasi Random Forest sebesar 4,6% pada evaluasi uji coba dan 0,4% serta 1,8% pada evaluasi validasi *k-fold cross*, dengan kesimpulan akurasi RF lebih tinggi dibandingkan dengan model DT dan SVM. Penelitian menyarankan analisis lebih mendalam hubungan antar variabel dan penggunaan metode seperti *Association Rules* untuk menguji korelasi variabel, serta menambahkan variabel lain yang mungkin mempengaruhi prestasi akademik.

(Ouatik et al., 2022) dalam penelitiannya yang berjudul **“Predicting Student Success Using Big Data and Machine Learning Algorithms”** mendiskusikan penggunaan SVM, KNN, dan algoritma C4.5 untuk prediksi kinerja akademik dalam lingkungan pembelajaran virtual. Faktor yang paling

mempengaruhi dalam penelitian adalah penilaian akademik, status ekonomi, pendidikan orang tua, jarak dari rumah, minat siswa, gangguan psikologis, dan akses ke kelas virtual. Hasil perbandingan model berdasarkan ukuran kinerja dan waktu eksekusi membuktikan model algoritma Sequential Minimal Optimization (SMO) pada tahap pemilihan atribut dan algoritma SVM pada tahap klasifikasi merupakan model terbaik dibandingkan C4.5 dan KNN. Tantangan dalam pengolahan data yang beragam dan banyak dapat diatasi dengan model *ensemble* yang dioptimalkan melalui teknik *stacking* atau kombinasi algoritma.

Algoritma RF juga digunakan pada penelitian (Nachouki & Naaj, 2022) yang berjudul "***Predicting Student Performance to Improve Academic Advising Using the Random Forest Algorithm***". Dimana model *Critical Performance Measure* (CPM) dengan algoritma RF sebagai metode efektif untuk memprediksi kinerja akademik mahasiswa. Model yang digunakan pada penelitian tersebut dianggap memungkinkan identifikasi dini terhadap mahasiswa yang berisiko putus sekolah dan membantu dalam penentuan mata kuliah yang krusial dalam menentukan CGPA akhir mahasiswa. Temuan utama dari penelitian menunjukkan korelasi yang kuat antara nilai rata-rata Sekolah Menengah Atas dan CGPA mahasiswa di perguruan tinggi, serta mengidentifikasi mata kuliah spesifik seperti Matematika Diskrit, Organisasi Komputer, dan Struktur Data yang mempengaruhi hasil akhir siswa secara signifikan.

Menurut penelitian (Sahlaoui et al., 2021) variabel yang mempengaruhi kinerja siswa adalah domain keluarga, personal, akademis, ekonomi, sosial, dan kelembagaan (family factor, personal factor, institutional factor, academic factor,

economic factor). Faktor internal seperti usia dan jenis kelamin merupakan faktor yang paling sering digunakan karena mudah didefinisikan dan memiliki ukuran yang jelas untuk digunakan dalam penelitian prediksi. Faktor akademis seperti nilai akademis dan skor indeks prestasi kumulatif (IPK) menjadi atribut terpenting untuk memprediksi masa depan pendidikan mahasiswa. Faktor Ekonomi mengacu pada kemampuan finansial orang tua dalam memfasilitasi anak dalam membentuk karir di masa depan melalui Pendidikan, dalam hal ini adalah penghasilan orangtua. Faktor keluarga yang juga banyak digunakan untuk penelitian adalah menggunakan atribut pendidikan terakhir orang tua dan pekerjaan orangtua.

Secara keseluruhan, penelitian terdahulu menunjukkan bahwa model pembelajaran mesin, terutama RF, SVM, dan KNN efektif dalam memprediksi kinerja akademik mahasiswa. Algoritma ini dapat diterapkan dalam konteks penelitian prediksi kelulusan tepat waktu atau terlambat lulus pada mahasiswa, dengan mempertimbangkan data historis seperti, prestasi awal dan faktor sosial ekonomi. Penggunaan model *ensemble* juga dapat meningkatkan akurasi prediksi dengan menggabungkan kekuatan dari berbagai algoritma. Dengan menambahkan faktor-faktor yang relevan seperti latar belakang pendidikan orang tua, pekerjaan, dan keterlibatan dalam kegiatan akademik, penelitian ini dapat memberikan gambaran yang lebih komprehensif dalam memprediksi apakah mahasiswa akan lulus tepat waktu atau terlambat.

Penelitian saat ini bertujuan mengatasi keterbatasan model sebelumnya dengan mengembangkan model prediksi performa calon mahasiswa yang akan memasuki jenjang pendidikan tinggi. Penelitian ini menggunakan data historis

mahasiswa dari pendaftaran hingga kelulusan untuk mengidentifikasi atribut yang paling berpengaruh dalam prediksi performa akademis calon mahasiswa baru. Dengan menggunakan algoritma SVM, Random Forest, dan KNN, serta penerapan model XGBoost dengan konsep *stacking ensemble*, penelitian ini berusaha mengoptimalkan kinerja algoritma yang digunakan. Pendekatan ini relevan dengan penelitian terdahulu melalui penggunaan data historis, jenis algoritma yang digunakan, serta penerapan teknik optimasi *stacking ensemble* untuk mengatasi keterbatasan model prediksi dasar.

Berdasarkan hasil penelitian terdahulu, *stacking ensemble* telah terbukti lebih akurat dibandingkan model tunggal maupun pendekatan ensemble lainnya, terutama dalam prediksi kinerja akademik mahasiswa. Studi oleh Uddin et al. (2020) dan Fernández-García et al. (2021) menunjukkan bahwa penerapan *stacking* dapat meningkatkan akurasi prediksi hingga 15–20% dibandingkan model individu seperti SVM atau Random Forest. Penelitian Butt et al. (2023) dan Mahboob et al. (2023) juga menegaskan bahwa kombinasi multi-model berbasis *ensemble* memberikan hasil prediksi yang lebih stabil pada data pendidikan tinggi.

Namun demikian, mayoritas penelitian terdahulu masih terbatas pada penggunaan data akademik parsial, seperti nilai ujian tengah semester, indeks prestasi kumulatif (IPK), atau catatan kehadiran, dan belum banyak yang mengeksplorasi penggunaan data historis mahasiswa sejak tahap pendaftaran hingga kelulusan sebagai basis prediksi performa akademik sejak dini. Pendekatan berbasis *early prediction* ini sangat penting karena dapat membantu institusi

pendidikan mengidentifikasi calon mahasiswa berisiko lebih awal, sebelum mereka memulai proses perkuliahan.

Selain itu, meskipun pendekatan *stacking* sudah mulai banyak diterapkan, kombinasi spesifik antara SVM, KNN, dan *Random Forest* sebagai model dasar dengan XGBoost sebagai *meta-learner* masih jarang dieksplorasi, khususnya dalam konteks prediksi kelulusan tepat waktu menggunakan data historis pendaftaran mahasiswa yang mencakup variabel akademik, sosial, dan ekonomi. Kombinasi algoritma tersebut memiliki potensi sinergis yang tinggi: SVM unggul dalam pemisahan data non-linear, KNN efektif dalam mengenali pola lokal antar individu, sementara *Random Forest* memberikan kestabilan dan ketahanan terhadap data yang tidak seimbang.

Penelitian ini berupaya mengisi kesenjangan tersebut dengan mengembangkan model prediksi performa calon mahasiswa yang lebih komprehensif melalui pendekatan *stacking ensemble* yang dioptimalkan. Model yang dikembangkan tidak hanya menggabungkan kekuatan algoritma dasar, tetapi juga memperluas cakupan variabel untuk mencakup dimensi demografis, akademik, dan sosioekonomi. Selain itu, perhatian khusus diberikan pada proporsi distribusi kelas dalam pembagian dataset, guna memastikan representasi yang seimbang antara mahasiswa yang lulus tepat waktu, terlambat, maupun yang tidak menyelesaikan studi. Pendekatan ini dimaksudkan untuk meningkatkan kemampuan generalisasi model dan mengurangi bias terhadap kelas mayoritas.

Beberapa studi menunjukkan bahwa model pembelajaran mesin sering mengalami kesulitan dalam memprediksi kategori minoritas akibat distribusi data

yang tidak seimbang (Almutairi et al., 2022; Mohamad et al., 2021). Untuk mengatasi tantangan ini, penelitian ini menggunakan teknik stratifikasi dalam proses pemisahan data latih dan data uji, guna memastikan bahwa proporsi antara kelas mayoritas dan minoritas tetap terjaga dalam setiap subset data. Pendekatan ini penting untuk meningkatkan keakuratan dan ketahanan model terhadap bias klasifikasi.

Berbeda dari penelitian sebelumnya yang hanya berfokus pada metrik akurasi, penelitian ini juga mengevaluasi kinerja antar kelas serta kemampuan model dalam mengidentifikasi calon mahasiswa yang berisiko tidak lulus tepat waktu. Penelitian ini memiliki kebaruan dalam penerapan metode *Stacking Ensemble Learning* yang untuk pertama kalinya digunakan dalam memprediksi performa akademik calon mahasiswa baru. Berbeda dengan penelitian sebelumnya yang menggunakan pendekatan *Boosting Ensemble* atau model tunggal seperti *Random Forest* dan *Support Vector Machine* dalam konteks seleksi penerimaan mahasiswa, penelitian ini mengusulkan kombinasi model dasar (RF, KNN, dan SVM) dengan *XGBoost* sebagai *meta-learner* untuk meningkatkan akurasi serta stabilitas prediksi multikelas. Selain itu, penelitian ini memperkenalkan pendekatan *preprocessing* dan pengelolaan data yang unik, yakni integrasi antara teknik *stacking* dan pengaturan distribusi variabel melalui metode stratifikasi, yang diterapkan pada data historis pendaftaran mahasiswa. Dengan demikian, model yang dikembangkan mampu menghasilkan sistem prediksi kelulusan yang lebih adaptif, kontekstual, dan relevan terhadap karakteristik institusi pendidikan tinggi.

2.2. Keaslian Penelitian

Tabel 2. 1 Matriks literatur review

penelitian pengembangan model prediksi calon mahasiswa menggunakan pendekatan *stacking ensemble learning*

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	<i>Performance Prediction of Students in Higher Education using Multi-model Ensemble Approach</i>	Butt, Naveed Anwer Mahmood et.all, IEEE,2023	Menentukan pola data yang dapat digunakan untuk memprediksi kinerja mahasiswa berdasarkan data pendaftaran dan akademik. Selain itu, bertujuan untuk mengidentifikasi indikator kinerja utama yang efektif dan efisien untuk mendukung pengambilan keputusan yang kompleks dengan membangun model prediktif. Membandingkan algoritma <i>supervised machine learning</i> dan klasifikasi dengan	Penelitian ini menyimpulkan penggunaan metode ensemble dapat meningkatkan prediksi kinerja akademik mahasiswa. Metode <i>stacking-ensemble</i> adalah yang terbaik secara keseluruhan dalam meningkatkan model prediksi kinerja, meningkatkan akurasi, mengurangi tingkat kesalahan, dan meningkatkan efisiensi prediksi.	Peneliti menyarankan agar model prediktif menggunakan model ensemble untuk memprediksi data multi-kelas yaitu dengan mempertimbangkan aspek lintas budaya dan global dari akademik. Peneliti mengakui bahwa studi yang ada tentang penggunaan model <i>ensemble</i> dalam prediksi kinerja akademik mahasiswa masih terbatas. Selain itu, peneliti menyarankan untuk mengembangkan dan menguji model <i>stacking ensemble</i> agar dapat meningkatkan model prediksi, meningkatkan akurasi, mengurangi tingkat kesalahan, dan meningkatkan efisiensi prediksi.	Penelitian ini mengusulkan penggunaan tiga algoritma sebagai model dasar, yaitu SVM, RF, dan KNN, dimana mempertimbangkan fitur-fitur yang ada dari penelitian sebelumnya dan lebih spesifik pada penerapan satu model <i>stacking ensemble</i> . Sehingga hasil dari masing-masing prediksi dengan menggunakan meta model menggunakan XGBoost dapat dibandingkan untuk memperoleh kombinasi model tunggal yang terbaik untuk digunakan pada <i>stacking</i> .

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			<i>ensemble learning</i> untuk mengetahui performa mahasiswa.			
2	<i>A Real-Life Machine Learning Experience for Predicting University Dropout at Different Stages Using Academic Data</i>	Fernandez-Garcia et al, IEEE. 2021.	Membuat sistem prediktif untuk mengantisipasi risiko mahasiswa putus sekolah pada setiap tahap studi, dengan mengembangkan model menggunakan algoritma seperti Gradient Boosting, Random Forest, dan SVM dengan Cascade Ensemble.	Sistem prediktif mampu mendeteksi risiko putus sekolah mahasiswa dengan tingkat akurasi tinggi pada setiap tahap. Pada tahap pendaftaran, model dapat mendeteksi lebih dari 72% mahasiswa berisiko putus sekolah, dan pada akhir semester pertama, tingkat deteksi meningkat hingga 82% dengan presisi 82,91%. Pada akhir semester keempat model dapat mendeteksi hingga 91,5% mahasiswa yang menghentikan studi.	Penelitian mendatang disarankan untuk memperluas cakupan ke area studi lain, eksplorasi data pendaftaran atau fitur non-akademik untuk prediksi dini, mengeksplorasi metrik baru, dan mengadaptasi teknik optimasi <i>ensemble</i> untuk mendukung kebijakan pencegahan putus sekolah secara lebih efektif.	Penelitian yang diusulkan saat ini memperluas pendekatan tiga algoritma untuk dioptimalkan dengan pencarapan <i>stacking ensemble</i> XGBoost. Tiga model tersebut diantaranya SVM, RF, dan KNN. Fokus penelitian adalah prediksi untuk tahap penerimaan mahasiswa saja, sehingga akan lebih spesifik dalam mengoptimalkan performa prediksi. Fitur yang digunakan hanya yang berkaitan dengan penerimaan yaitu data pendaftaran sampai kelulusan dengan mengambil data mahasiswa satu angkatan.
3	<i>Creating a Recommender System to Support</i>	Fernandez-Garcia et al, IEEE. 2020	Menciptakan Sistem Pendukung Keputusan yang dapat dipergunakan	Penelitian ini mengembangkan model yang digunakan untuk membangun	Menyediakan skenario data yang lebih luas, tidak hanya difokuskan pada program studi Ilmu	Penelitian yang diusulkan saat ini tidak menghasilkan sistem pendukung

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	<i>Higher Education Students in the Subject Enrollment Decision</i>		untuk pertimbangan jalur akademis bagi mahasiswa dan fakultas	sistem rekomendasi diantaranya Decision Tree, Random Forest, Gradient Boosting Classifier, Logistic Regression, Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Multilayer Perceptron (MLP). Sistem menghasilkan rekomendasi untuk membantu mahasiswa memilih mata kuliah yang sesuai, dengan memprediksi keberhasilan akademis	Komputer, untuk validasi pendekatan yang lebih baik.	keputusan, dibatasi pada prediksi performa model yang digunakan kemudian dioptimalkan untuk memprediksi status kelulusan calon mahasiswa. Penelitian yang diusulkan menggunakan tiga algoritma dan teknik <i>stacking ensemble</i> dengan algoritma XGBoost. Tiga model tersebut diantaranya SVM, RF, dan KNN. Data yang digunakan merupakan data dari 14 program studi dalam satu Angkatan jadi tidak hanya fokus di satu program studi.
4	<i>Multiclass Prediction Model for Student Grade Prediction Using Machine Learning</i>	Siti Dianah Abdul Bujang IEEE. 2021	Membandingkan akurasi dari berbagai algoritma pembelajaran mesin, seperti Decision Tree (J48), k-Nearest Neighbor (kNN), Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression	Model prediksi yang diusulkan, terutama dengan integrasi SMOTE dan Random Forest (RF), memberikan hasil terbaik dengan f-measure tertinggi sebesar 99,5%. SMOTE secara signifikan meningkatkan akurasi	Penelitian hanya menggunakan algoritma pembelajaran mesin tertentu, tanpa mengeksplorasi metode <i>ensemble</i> atau teknik pembelajaran mesin lanjutan. Saran untuk penelitian selanjutnya adalah dengan mengeksplorasi algoritma pembelajaran mesin lanjutan atau teknik <i>ensemble</i>	Penelitian yang diusulkan saat ini mencampurkan metode <i>ensemble</i> dengan jenis <i>stacking ensemble</i> menggunakan XGBoost untuk mengoptimalkan hasil prediksi dari kombinasi tiga algoritma, diantaranya SVM, RF, dan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			(LR), dan Random Forest (RF) untuk untuk memprediksi nilai akhir mahasiswa. Mengatasi masalah ketidakseimbangan data <i>multiclass</i> dengan menggunakan teknik oversampling (SMOTE) dan algoritma seleksi fitur.	dibandingkan hanya menggunakan seleksi fitur.	untuk meningkatkan akurasi prediksi. Mengadopsi teknik evaluasi tambahan untuk menangani dataset <i>multiclass</i> yang tidak seimbang.	KNN. Penelitian yang diusulkan tidak memprediksi nilai namun mengklasifikasikan status calon mahasiswa apakah akan diprediksi lulus tepat waktu, terlambat, atau dikeluarkan.
5	<i>Predicting Student Success Using Big Data and Machine Learning Algorithms</i>	Farouk Ouatik, UJET (International Journal Emerging Technology in Learning), 2022	Memprediksi keberhasilan dan kegagalan akademik siswa menggunakan teknologi big data dan algoritma pembelajaran mesin. Mengidentifikasi faktor-faktor yang mempengaruhi kesuksesan akademik dan membangun model prediksi yang dapat membantu mengurangi tingkat kegagalan akademik.	Hasil perbandingan model berdasarkan akurasi dan waktu eksekusi membuktikan model algoritma SMO pada tahap pemilihan atribut dan algoritma SVM pada tahap klasifikasi merupakan model terbaik dibandingkan C4.5 dan KNN. SVM memiliki tingkat klasifikasi tertinggi (87,32%) dan waktu eksekusi terendah.	Kelemahan dapat dijumpai pada algoritma SVM. Pada dasarnya Algoritma SVM tidak cocok untuk dataset besar karena membutuhkan waktu training yang lama, sensitif terhadap noise dan disertai dengan kompleksitas komputasi yang tinggi. Meskipun sudah menggunakan optimasi algoritma SMO untuk melatih model SVM yang mencapai ukuran kinerja dan waktu eksekusi terbaik tetap ditemukan keraguan untuk penggunaan model SVM.	Peneliti mengusulkan prediksi multi model dengan salah satunya model SVM yang dioptimasi menggunakan <i>random search</i> . Sehingga dapat memberikan efisiensi waktu pada komputasi yang tinggi dan cocok untuk ruang parameter besar. Selain SVM algoritma lain yang digunakan adalah RF, KNN dan XGBoost yang digunakan sebagai meta-model

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	<i>Predicting Student Performance to Improve Academic Advising Using the Random Forest Algorithm</i>	Mirna Nachouki & Mahmoud Abou Naaj, International Journal of Distance Education Technologies, 2022	Mengidentifikasi atribut-atribut yang mempengaruhi prestasi akademik siswa; Mengembangkan model yang valid untuk memprediksi prestasi akademik siswa; Identifikasi tahap awal bagi siswa yang kinerjanya rendah; Mengkategorikan mata kuliah yang mempunyai dampak signifikan terhadap perjalanan akademik mahasiswa.	Penelitian ini merekomendasikan penggunaan model Critical Performance Measure (CPM), yang diimplementasikan dengan algoritma RF, sebagai metode efektif memprediksi kinerja akademik mahasiswa. Model identifikasi dini terhadap mahasiswa berisiko putus sekolah dengan tingkat akurasi tinggi. CPM membantu mengidentifikasi mata kuliah krusial dalam menentukan CGPA akhir mahasiswa. Penelitian menemukan korelasi kuat antara nilai rata-rata sekolah menengah atas dan CGPA mahasiswa di perguruan tinggi. Mata kuliah spesifik seperti Matematika Diskrit, Organisasi Komputer, dan Struktur Data ditemukan mempengaruhi hasil akhir siswa secara signifikan. Hasil penelitian menunjukkan akurasi 91,32% dan 92,87%.	Pertama, penggunaan sampel yang relatif kecil (82 dan 105 catatan) dapat mengurangi kemampuan untuk mengeneralisasi hasil secara luas ke mahasiswa program TI secara umum atau institusi lainnya. Kedua, penelitian ini terfokus pada satu bidang studi tertentu, sehingga hasilnya mungkin tidak dapat diterapkan langsung pada jurusan atau institusi pendidikan lain yang memiliki karakteristik dan faktor-faktor yang berbeda. Ketiga, tidak disebutkan secara rinci faktor-faktor yang mempengaruhi kinerja akademik yang telah dipertimbangkan dalam model CPM, sehingga membatasi pemahaman tentang kekuatan dan kelemahan prediksi. Terakhir, meskipun model CPM memberikan akurasi yang tinggi dalam pada penelitian, masih diperlukan penelitian lebih lanjut untuk menguji keandalan dan kegunaannya dalam konteks yang lebih luas.	Penelitian yang diusulkan saat ini akan menggunakan dataset dengan jumlah yang lebih banyak dan menggunakan tiga algoritma untuk prediksi. Salah satu algoritma dasar yang digunakan dalam prediksi awal adalah Random Forest dengan penerapan teknik feature importance untuk memastikan atribut yang sesuai dalam proses pelatihan. Menggunakan atribut yang berpengaruh seperti penilaian akademik berupa nilai rata-rata dari SKHU SLTA, informasi pribadi, dan kegiatan non-akademis dalam prediksi performa mahasiswa, kemudian hasil dari prediksi model RF dan kedua model lainnya akan digunakan sebagai dataset baru untuk optimasi model menggunakan XGBoost.

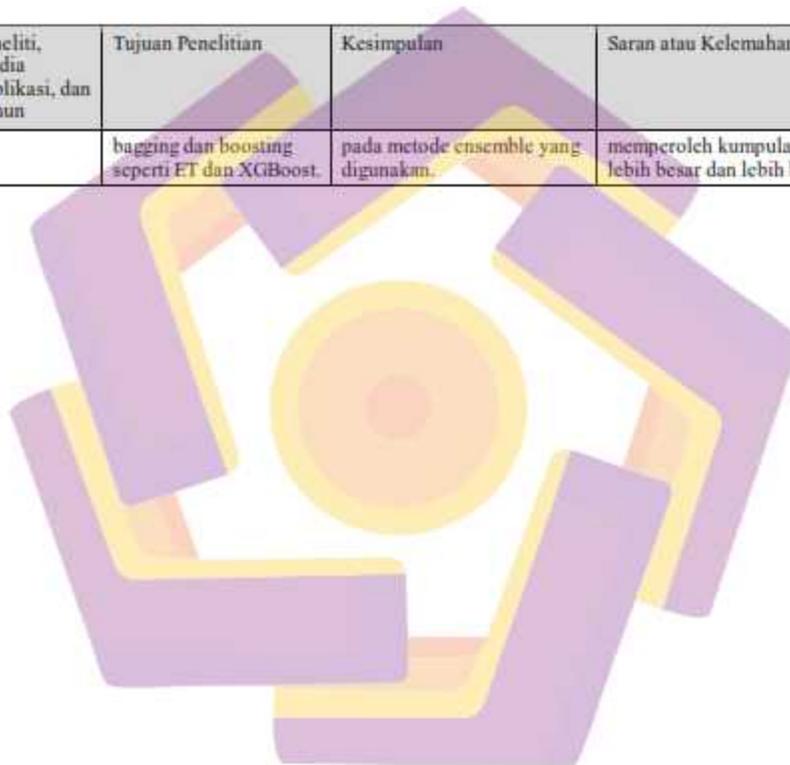
No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
7	<i>Educational data mining: prediction of students' academic performance using machine learning algorithms</i>	Mustafa Yağcı, Smart Learning Environments, 2022	Tujuan dari penelitian ini adalah untuk mengembangkan model baru berdasarkan algoritma pembelajaran mesin yang dapat memprediksi nilai ujian akhir mahasiswa sarjana dengan menggunakan data ujian tengah semester sebagai sumber informasi. Model ini diharapkan dapat membantu dalam meningkatkan kinerja akademik siswa dengan memberikan prediksi yang akurat, sehingga intervensi pedagogis dan kebijakan baru dapat dikembangkan untuk mendukung siswa yang berpotensi tidak lulus.	Studi ini mengevaluasi beberapa algoritma pembelajaran mesin untuk memprediksi nilai akademik siswa berdasarkan data nilai ujian tengah semester, fakultas, dan jurusan. Algoritma Random Forest (RF), Neural Network (NN), dan Support Vector Machine (SVM) yang menunjukkan akurasi klasifikasi tertinggi, sementara k-Nearest Neighbors (kNN) memiliki akurasi terendah. Penelitian menyimpulkan nilai ujian tengah semester adalah prediktor penting untuk memprediksi nilai ujian akhir siswa, dan bahwa model yang diusulkan mencapai akurasi klasifikasi sekitar 70-75% dengan menggunakan data yang relatif sederhana seperti nilai ujian tengah semester, fakultas, dan jurusan.	Keterbatasan dalam jumlah parameter yang digunakan untuk memprediksi nilai ujian akhir siswa, hanya menggunakan tiga jenis parameter: nilai ujian tengah semester, data jurusan, dan data fakultas. Meskipun model mencapai akurasi klasifikasi yang cukup, yaitu sekitar 70-75%, keterbatasan ini dapat memengaruhi akurasi prediksi karena tidak memperhitungkan faktor-faktor lain yang berpotensi memengaruhi prestasi akademik siswa. Diperlukan penelitian lanjutan dengan sampel data yang lebih besar dan beragam serta penggunaan parameter yang lebih komprehensif untuk memperbaiki kelemahan tersebut serta meningkatkan validitas dan keandalan model prediksi yang diusulkan.	Fokus penelitian masih pada evaluasi hasil belajar di tengah masa studi. Penelitian saat ini mengadopsi teknik stacking ensemble untuk mengintegrasikan model-model terbaik (SVM, RF, KNN) dan menambahkan variabel akademik dan non-akademik dari data awal mahasiswa guna memperkuat ketepatan prediksi kelulusan.

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
8	<i>Student Performance Prediction in Sebelas Maret University Based on the Random Forest Algorithm</i>	Gusnina, IJETA, 2022	Melakukan prediksi prestasi akademik mahasiswa Universitas Sebelas Maret berdasarkan faktor sosial, ekonomi, dan akademik dengan menggunakan algoritma pembelajaran mesin. Penelitian juga bertujuan untuk mengidentifikasi faktor-faktor yang paling berpengaruh terhadap kinerja akademik dan meningkatkan akurasi model prediksi.	Faktor yang paling berpengaruh terhadap prestasi akademik mahasiswa di Universitas Sebelas Maret meliputi indeks prestasi, pekerjaan orang tua dan tingkat pendidikan, kehadiran di kelas, ketergantungan keluarga, biaya sekolah, nilai ujian SMA, pekerjaan ibu, jam belajar, prestasi akademik, masuk perguruan tinggi, penggunaan ponsel cerdas, jenis kelamin, pendapatan keluarga, tempat tinggal, dan jarak ke perguruan tinggi. RF mencapai tingkat akurasi 90,7%, mengungguli algoritma SVM dan Decision Tree. Selain itu, penggunaan metode pemilihan fitur Information Gain meningkatkan tingkat akurasi metode klasifikasi RF	Kelemahan penelitian ini mencakup kurangnya diskusi mengenai implikasi praktis dari temuan penelitian dan kegagalan memperhitungkan potensi variabel perancu yang dapat mempengaruhi kinerja akademik. Selain itu, penelitian ini mungkin memiliki keterbatasan karena fokus spesifik pada satu universitas dan ukuran sampel yang relatif kecil yaitu 223 mahasiswa. Saran penelitian dapat mencakup menggali lebih dalam implikasi praktis dari temuan penelitian dan mempertimbangkan variabel tambahan yang dapat berdampak pada kinerja akademik.	Metode optimasi model yang digunakan pada penelitian saat ini yaitu menggunakan Stacking Ensemble Learning. Sedangkan pada penelitian Gusnina menerapkan Information Gain untuk meningkatkan akurasi metode klasifikasi Random Forest.
9	<i>Predicting students at risk of academic</i>	Karalar, International	mengembangkan model prediksi yang dapat	Pendekatan model pembelajaran ansambel, yang	Ketergantungan pada data dari satu institusi pendidikan tinggi tertentu,	Penelitian sebelumnya menggunakan ensemble

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	<i>failure using ensemble model during pandemic in a distance learning system</i>	Journal of Educational Technology in Higher Education, 2021	mengidentifikasi siswa yang berisiko mengalami kegagalan akademik selama pandemi dan non-pandemi, serta untuk memberikan dasar untuk kebijakan dan intervensi pendidikan tinggi yang lebih efektif.	terdiri dari kombinasi algoritma klasifikasi Extra Trees (ET), Random Forest (RF), dan Logistic Regression (LR), merupakan yang terbaik dalam memprediksi siswa berisiko kegagalan akademik. Hasilnya menunjukkan fitur skor kuis, gelar, jumlah pengunduhan catatan kuliah, pengunduhan materi kuliah, dan total waktu yang dihabiskan untuk menonton rekaman video kuliah, efektif dalam memprediksi siswa berisiko. Model ansambel yang diusulkan menghasilkan prediksi yang baik dengan spesifisitas 90,34% pada algoritma RF	kurangnya pengujian model pada berbagai konteks institusi, dan kendala dalam menggeneralisasi temuan ke institusi yang berbeda. Selain itu, penelitian ini mungkin memiliki keterbatasan dalam menggali faktor-faktor eksternal yang dapat memengaruhi kinerja siswa selama pandemi. Penting untuk meningkatkan kemampuan generalisasi model yang sudah ada untuk akurasi prediksi yang lebih tinggi. Disarankan untuk menambahkan fitur baru yang efektif untuk melatih model dalam penelitian serupa	umum (bagging/boosting) untuk optimasi model. Sedangkan penelitian saat ini mengembangkan stacking ensemble dengan meta-model XGBoost untuk konteks berbeda, yakni prediksi kelulusan mahasiswa baru pada pembelajaran reguler, sehingga menghasilkan model yang lebih kontekstual dan terukur.
10	Studi Komparasi Naive Bayes, K-Nearest Neighbor, dan Random Forest untuk Prediksi Calon Mahasiswa	Puteri Sejati, Munawar, Marzuki Piliang, Habibullah Akbar	Penelitian ini bertujuan untuk mengidentifikasi model prediksi terbaik dari data Penerimaan Mahasiswa Baru (PMB) antara tahun 2014 hingga 2019 dengan	Peneliti menyimpulkan bahwa pembersihan dataset dari noise dan outlier penting untuk akurasi model yang optimal. Nilai k yang tepat pada KNN dan optimasi hyperparameter pada Random	Proses optimasi pada Random Forest memerlukan waktu yang lama, sehingga perlu penelitian lebih lanjut untuk mengembangkan model optimal dengan waktu komputasi yang lebih efisien.	Pada prosesnya menggunakan Algoritma Random Forest dan menggunakan optimasi hyperparameter untuk meningkatkan performa model. Penelitian tersebut

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	yang Diterima atau Mundur	Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), 2022	membandingkan kinerja algoritma Naive Bayes, K-Nearest Neighbor, dan Random Forest. Tujuan utama penelitian ini adalah untuk membantu departemen pemasaran dalam mengurangi jumlah calon mahasiswa yang mengundurkan diri.	Forest meningkatkan akurasi. Dalam penelitian ini, Random Forest memiliki akurasi validasi terbaik (73,61%), mengungguli KNN (72,08%) dan Naive Bayes (70,47%). Hal ini menyarankan Random Forest sebagai model pilihan untuk sistem pendukung keputusan dalam seleksi calon mahasiswa baru.		menggunakan data penerimaan mahasiswa baru, namun masih terbatas pada klasifikasi biner (diterima/mundur). Penelitian saat ini mengembangkannya menjadi model multikelas dengan teknik stacking ensemble
11	<i>Predicting and Interpreting Student Performance Using Ensemble Models and Shapley Additive Explanations</i>	Hayat Sahlaoui, El Arbi Abdellaoui Alaoui, Anand Nayyar, Said Agoujil, and Mustafa Musa Jaber. IEEE, 2021	Penelitian bertujuan untuk meningkatkan akurasi dan interpretabilitas model prediksi kinerja siswa dengan menggunakan metode ensemble dan nilai SHAP. Penelitian ini bertujuan untuk meningkatkan transparansi model dengan memanfaatkan teknik seperti SMOTE untuk mengatasi ketidakseimbangan kelas dan algoritma berbasis	Kesimpulan dari penelitian ini menunjukkan akurasi dapat ditingkatkan dengan menggunakan metode ensemble, yaitu ET dan XGBoost, dengan menggunakan strategi SMOTE, optimasi hyperparameter, dan proses validasi silang sepuluh yang mempengaruhi ketergantungan model baru. Nilai SHAP yang digunakan dapat memberikan gambaran jelas tentang fitur yang mempengaruhi kinerja siswa	Saran dari penelitian ini mencakup rekomendasi untuk peneliti pembelajaran mesin agar mengungkapkan kumpulan data pendidikan lebih banyak untuk melatih model prediksi kinerja siswa dengan tetap mempertimbangkan perlindungan privasi dan dampak ekonomi. Kelemahan dari penelitian ini adalah ketergantungan pada kumpulan data yang tersedia untuk umum yang relatif kecil, yang dapat membatasi generalisasi temuan. Selain itu, penelitian ini mengakui tantangan dalam	Penelitian sebelumnya juga menggunakan nilai SHAP untuk menemukan gambaran fitur yang paling berpengaruh. Namun penelitian tersebut masih terbatas pada dua kelas (lulus/tidak lulus). Penelitian ini memperluas ke klasifikasi multikelas (tepat waktu, terlambat, dikeluarkan) dengan pendekatan stacking yang lebih kompleks.

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			bagging dan boosting seperti ET dan XGBoost.	pada metode ensemble yang digunakan.	memperoleh kumpulan data yang lebih besar dan lebih beragam.	



2.3. Landasan Teori

7.1. Algoritma SVM

SVM (Ouatik et al., 2022) merupakan serangkaian algoritma dan metode pembelajaran mesin yang digunakan untuk regresi, klasifikasi, dan deteksi anomali. Algoritma ini sangat fleksibel dan mudah diterapkan. Prinsip utama SVM adalah memisahkan data pelatihan berdasarkan kelasnya dengan memaksimalkan jarak (margin) antara batas pemisah dan kelompok data yang berbeda. Untuk data yang tidak dapat dipisahkan secara linear, SVM menggunakan kernel untuk memproyeksikan data ke ruang vektor berdimensi tinggi, sehingga data dapat dipisahkan dengan lebih baik.

Support Vector Machine (SVM) adalah algoritma yang efisien dalam klasifikasi karena didukung oleh teori matematika yang kuat. Pada tahun 1992, SVM diperkenalkan oleh Vladimir Vapnik, Bernhard Boser, dan Isabelle Guyon. Namun, dasar-dasar untuk SVM sudah ada sejak tahun 1960-an, termasuk karya awal oleh Vapnik dan Alexei Chervonenkis dalam teori pembelajaran statistik. SVM adalah jenis algoritma supervised learning karena membutuhkan target pembelajaran saat proses training. Algoritma ini bekerja dengan menyelesaikan masalah klasifikasi melalui pencarian maximum marginal hyperplane (MMH). Secara sederhana, konsep SVM adalah usaha untuk menemukan *hyperplane* atau fungsi pemisah optimal yang dapat memisahkan dua set data dari dua kelas yang berbeda. (Han dkk, 2012).

Misalkan diberikan 2 atribut input A_1 dan A_2 , himpunan $W = \{w_1, w_2, \dots, w_d\}$; W merupakan bobot/weight, d adalah jumlah atribut, dan tupel training X

$= (x_1, x_2)$ dimana x_1 dan x_2 adalah nilai-nilai atribut A_1 dan A_2 , maka fungsi hyperplane dapat dinotasikan sebagai berikut:

$$f(x) = W \cdot X + b \quad (1)$$

dimana $W, X \in \mathbb{R}^d$ (d adalah jumlah atribut) dan b adalah bias yang berupa skalar (Hamel, 2009).

Hyperplane yang terletak diantara dua set objek dari kelas positif ($y_1 = +1$) dan kelas negatif ($y_2 = -1$) dapat ditulis sebagai berikut:

$$H1: x^T w + b \geq 1 \text{ untuk } y_1 = +1 \quad (2)$$

$$H2: x^T w + b \leq -1 \text{ untuk } y_2 = -1 \quad (3)$$

Penggabungan dari persamaan (2) dan (3)

menghasilkan pertidaksamaan:

$$y_i(x^T w + b) \geq 1, \text{ untuk } \forall i = 1, 2, \dots, n \quad (4)$$

Margin antara dua kelas dapat dihitung dengan cara mencari jarak antara kedua *hyperplane* $H1$ atau $H2$. Setiap tupel pelatihan yang jatuh pada *hyperplane* $H1$ atau $H2$ yang memenuhi persamaan (2) disebut *support vector*.

7.2. Algoritma Random Forest

Metode Random Forest adalah sebuah teknik pembelajaran berbasis *ensemble learning* yang memanfaatkan algoritma *Decision Tree* (Sejati et al., 2022). Dalam metode ini, beberapa model *Decision Tree* dilatih secara independen, kemudian dilakukan prediksi menggunakan data uji yang sama untuk setiap model. Hasil prediksi dari setiap model kemudian digabungkan melalui proses majority voting menggunakan metode modus, di mana kelas yang paling sering muncul menjadi kelas prediksi akhir. Model random forest dapat diformulasikan sebagai berikut:

$$E = -\sum_{i=1}^n p_i \cdot \log_2(p_i) \quad (5)$$

p_i adalah peluang terpilihnya contoh secara acak di kelas i

$$Gain = E_{parent} - E_{children} \quad (6)$$

Persamaan (5) digunakan untuk menghitung Entropy, sedangkan persamaan (6) digunakan untuk menghitung Gain pada metode Random Forest.

7.3. Algoritma K-Nearest Neighbor

Algoritma K-Nearest Neighbor (KNN), yang sering digunakan untuk tujuan klasifikasi, juga dapat digunakan untuk estimasi dan ramalan atau prediksi. KNN merupakan salah satu contoh dari pembelajaran berbasis instan, di mana kumpulan data training disimpan untuk memungkinkan klasifikasi data baru yang belum terklasifikasi. Dalam konteks ini, algoritma mencari data sampel yang paling mirip dengan data baru dengan cara membandingkan karakteristiknya. Lebih jelasnya KNN merupakan pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama.

Dalam klasifikasi, KNN (K-Nearest Neighbors) menentukan kelas data baru berdasarkan k tetangga terdekat dari data pelatihan. Jarak antara data baru dan data pelatihan diukur menggunakan metrik seperti Euclidean distance atau Manhattan distance. Nilai k dapat disesuaikan dengan karakteristik data untuk menghasilkan klasifikasi yang optimal. Prinsip KNN didasarkan pada asumsi bahwa objek yang serupa cenderung memiliki label yang sama. Metode ini sederhana serta telah banyak digunakan untuk memprediksi kinerja siswa, seperti tingkat kelulusan dan prestasi akademik. (Wiyono & Abidin, 2018)

$$\text{Similarity: } (T, S) = \sum_{i=1}^n = \frac{f(r_1, s_1) \times w_1}{\sum_{i=1}^n w_1}$$

Dalam analisis komparatif yang melibatkan beberapa algoritma, Elouafi menemukan bahwa KNN termasuk yang berkinerja terbaik dalam memprediksi kinerja matematika siswa, yang menunjukkan efektivitasnya dalam penilaian pendidikan (Elouafi et al., 2024). Demikian pula, studi Munarsih yang membandingkan KNN dengan Naive Bayes dan C4.5 untuk memprediksi prestasi akademik di Universitas Pamulang mengungkapkan bahwa KNN secara konsisten memberikan prediksi yang andal berdasarkan berbagai faktor demografi dan akademik.

Lebih jauh, penelitian Wiyono dan Abidin juga menekankan kekuatan komparatif KNN terhadap algoritma *machine learning* lainnya, seperti *Support Vector Machines* (SVM) dan *Decision Trees*, dalam memprediksi kinerja siswa (Wiyono & Abidin, 2019). Pendekatan komparatif ini sangat penting bagi lembaga pendidikan yang ingin menerapkan strategi berbasis data untuk meningkatkan keberhasilan siswa. KNN menonjol sebagai algoritma yang kuat untuk memprediksi kinerja siswa karena akurasinya yang tinggi, kemudahan implementasi, dan fleksibilitas dalam menangani berbagai atribut data. Penerapannya dalam penggalian data pendidikan tidak hanya membantu dalam mengidentifikasi siswa yang berisiko tetapi juga memfasilitasi intervensi tepat waktu untuk meningkatkan hasil akademis.

7.4. Ensemble Learning

Ensemble adalah teknik canggih dalam pembelajaran mesin yang melibatkan penggabungan beberapa model untuk meningkatkan kinerja prediktif. Ini adalah metode yang bertujuan untuk meningkatkan kekuatan prediksi dengan memanfaatkan kekuatan berbagai model dasar. Konsep ansambel melibatkan pelatihan model meta pada prediksi yang dibuat oleh beberapa model dasar, sehingga menciptakan model tingkat lebih tinggi yang mempelajari cara terbaik menggabungkan keluaran model dasar untuk membuat prediksi yang lebih akurat (Dick, 2024).

Penggunaan pembelajaran *ensemble* sangat dianjurkan untuk meningkatkan stabilitas dan akurasi prediksi dari algoritma pembelajaran individu. Model pembelajaran ansambel menggunakan meta-algoritma yang menggabungkan berbagai model yang dilatih secara terpisah, baik yang sama maupun berbeda, untuk menghasilkan prediksi akhir (Karalar et al., 2021). Pembelajaran ansambel dapat dikategorikan berdasarkan variasi algoritma pembelajaran dasar yang digunakan dalam model dan cara pembuatannya. Pertama, pembelajaran ansambel bisa heterogen atau homogen tergantung pada variasi algoritma pembelajaran yang digunakan dalam model. Kedua, pembelajaran ansambel dapat dibedakan menjadi metode averaging dan metode boosting. Dalam metode averaging, model pembelajaran dibuat secara independen (Karalar et al., 2021). Teknik *ensemble* bertujuan untuk mengurangi bias dan varians. Beberapa pendekatan ensemble yang penting adalah stacking, bagging, boosting, dan voting. (Butt et al., 2023).

7.4.1. Perbandingan *Stacking* dengan Teknik *Ensemble* Lainnya

Teknik ensemble terdiri dari beberapa pendekatan utama, yaitu *bagging*, *boosting*, dan *stacking*, yang memiliki mekanisme dan tujuan optimasi berbeda.

Pertama, *Bagging* (Bootstrap Aggregating) berfokus pada pengurangan *variance* dengan membangun model independen pada subset data acak, lalu menggabungkan hasilnya melalui rata-rata atau voting. Contoh terkenal dari pendekatan ini adalah Random Forest (Fernandez-Garcia et al., 2021).

Selanjutnya, *Boosting*, seperti *AdaBoost* dan *Gradient Boosting*, teknik meningkatkan performa model dengan memberikan bobot lebih tinggi pada kesalahan sebelumnya. Meskipun powerful, boosting lebih rentan terhadap *overfitting* (Butt et al., 2023).

Sedangkan, *Stacking ensemble* menggabungkan *output* dari model-model dasar yang berbeda dan melatih model meta (meta-learner) untuk belajar bagaimana mengoptimalkan penggabungan prediksi. *Stacking* terbukti lebih fleksibel dan mampu menangkap pola kompleks yang tidak bisa dijangkau *bagging* dan *boosting* (Alzahrani, 2024; Sahlaoui et al., 2021).

7.5. XGBoost

Terdapat empat teknik untuk membangun suatu model *ensemble* termasuk *bootstrap aggregating* (*bagging*), *boosting*, *stacking*, dan *blending*. Contoh model yang dibangun dari teknik *bagging* adalah Random Forest, sedangkan contoh *boosting* adalah algoritma Extreme Gradient Boosting (XGBoost). (Dachi, 2023; Muhammad Ricky Perdana Putra & Ema Utami, 2024).

eXtreme Gradient Boosting (XGBoost) merupakan algoritma dalam Ensemble Learning yang menerapkan metode *boosting*. Algoritma ini

dikembangkan oleh Tianqi Chen pada tahun 2014, dan berlandaskan pada prinsip gradient boosting, di mana fokus utamanya adalah pada contoh-contoh yang salah diklasifikasikan oleh model sebelumnya (Dangeti, 2017) dalam (Jan Melvin Ayu Soraya Dachi & Pardomuan Sitompul, 2023). Dijelaskan dalam berbagai penelitian bahwa XGBoost membagi pohon yang ditingkatkan menjadi dua kategori: pohon regresi dan pohon klasifikasi. XGBoost telah banyak diakui dalam berbagai kompetisi machine learning, seperti di Kaggle, berkat keunggulannya dalam efisiensi dan fleksibilitas yang tinggi. Sebagai bukti, dalam kompetisi machine learning di Kaggle pada tahun 2015, XGBoost muncul sebagai metode paling populer, dengan lebih dari 18 pemenang menggunakan algoritma ini (D. Zhang et al., 2018) di dalam (Jan Melvin Ayu Soraya Dachi & Pardomuan Sitompul, 2023).

XGBoost dirancang untuk menghindari overfitting dan meningkatkan efisiensi komputasi. Hal ini dicapai dengan menyederhanakan fungsi objektif, yang menggabungkan istilah prediktif dan regularisasi untuk mengendalikan kompleksitas model, sekaligus mempertahankan kecepatan komputasi yang optimal (Fan et al., 2018; Jan Melvin Ayu Soraya Dachi & Pardomuan Sitompul, 2023). Dalam konteks klasifikasi biner, fungsi objektif ini digunakan untuk mengukur kesalahan prediksi dan memperbarui model demi peningkatan akurasi. Fungsi objektif terdiri dari dua komponen: fungsi kerugian (loss function) dan istilah regularisasi. Menurut H. Li et al. (2020), untuk kasus ini, fungsi kesalahan yang digunakan dapat dinyatakan sebagai berikut (Jan Melvin Ayu Soraya Dachi & Pardomuan Sitompul, 2023):

$$\text{Min } L^{(t)}(y, \hat{y}^{(t)}) = \min \left(\sum_{i=1}^n l \left(+ \left(\sum_{k=1}^t \Omega(f_k) \right) \right) \right) \quad (7)$$

Keterangan:

fungsi $L^{(t)}(y, \hat{y}^{(t)})$ merupakan fungsi kerugian dengan y_i adalah nilai riil dan $\hat{y}_i^{(t)}$

adalah nilai prediksi. $\sum_{k=1}^t \Omega(f_k)$ merupakan regulasi dari model yang digunakan untuk mengukur kompleksitas keseluruhan model

Istilah regulasi dapat dirumuskan sebagai berikut:

$$\Omega(f_k) = \gamma + \frac{1}{2} \lambda \sum_{j=1}^{T_k} \omega_{kj}^2 \quad (8)$$

Keterangan:

T_k : simpul daun pada pohon ke- k

γ : koefisien pengurangan jumlah simpul daun T

ω_{kj}^2 : nilai simpul daun ke- j pada pohon ke- k

λ : koefisien penalti dari nilai simpul daun. □

7.6. Stacking Ensemble Learning

Stacking adalah suatu teknik yang melibatkan pelatihan model meta untuk membuat prediksi berdasarkan output dari beberapa model dasar. Model dasar ini biasanya terdiri dari berbagai jenis model yang digunakan untuk melakukan klasifikasi dari data set yang sama. Selanjutnya, proses *stacking* melibatkan pembagian data pelatihan menjadi dua atau lebih subset. Beberapa model dasar dilatih pada satu dataset, dan model terlatih ini kemudian digunakan untuk membuat prediksi pada subset data lainnya. Prediksi dari model dasar ini digabungkan menjadi dataset baru, yang kemudian digunakan untuk melatih model

meta. Nilai target aktual digunakan sebagai label dalam pelatihan model meta ini, dan model meta digunakan untuk membuat prediksi pada data uji (Sunarko et al., 2023).

Pendekatan *stacking ensemble* telah terbukti efektif dalam menangani data kompleks dan bervariasi, termasuk atribut demografis, akademik, dan sosioekonomi dalam prediksi kelulusan mahasiswa. *Stacking ensemble* menggabungkan beberapa model dasar untuk membentuk model prediktif tunggal yang lebih kuat, dengan memanfaatkan kekuatan masing-masing model untuk meningkatkan akurasi prediksi secara keseluruhan (Alzahrani, 2024; Sarat Kumar Chettri, HMM. Naleer, 2023; Yan & Liu, 2020). Dalam konteks ini, *stacking* dapat menangkap pola yang lebih kompleks dibandingkan dengan metode lain seperti *bagging* dan *boosting*. Sementara *bagging* hanya mengandalkan rata-rata hasil model, *stacking* menggunakan *meta-learner* yang belajar dari data untuk mengoptimalkan penggabungan model dasar, sehingga memberikan solusi yang lebih optimal (Sarat Kumar Chettri, HMM. Naleer, 2023).

Salah satu keuntungan utama dari *stacking* adalah kemampuannya dalam mengurangi risiko *overfitting*. Metode ini lebih mampu menghasilkan model yang memiliki generalisasi yang lebih baik dibandingkan dengan *boosting*, yang sering kali rentan terhadap *overfitting* karena berfokus pada kesalahan model sebelumnya (Noviandy et al., 2024; Sun et al., 2024). Penelitian menunjukkan bahwa *stacking ensemble* dapat memberikan akurasi prediksi yang lebih tinggi dalam berbagai konteks, termasuk dalam prediksi kinerja akademik mahasiswa (Alzahrani, 2024; Mahboob et al., 2023; Yan & Liu, 2020). Misalnya, dalam studi yang dilakukan

oleh Mahboob, ditemukan bahwa pendekatan stacking dan Random Forest mencapai tingkat akurasi tertinggi dalam prediksi nilai akademik (Mahboob et al., 2023)

Dalam perbandingan dengan teknik ensemble lainnya, stacking menawarkan keunggulan dalam hal fleksibilitas dan kemampuan untuk menangkap interaksi antar fitur yang lebih kompleks. Hal ini membuatnya lebih unggul dalam situasi di mana data memiliki banyak atribut yang saling berinteraksi, seperti dalam prediksi kelulusan mahasiswa yang melibatkan data demografis dan akademik (Yan & Liu, 2020).

Gagasan utama di balik *stacking* adalah bahwa setiap model dasar memiliki kelebihan dan kelemahan tersendiri. Dengan menggabungkan prediksi dari berbagai model dasar, model meta dapat belajar untuk menimbang kelebihan dan kelemahan ini dengan tepat, sehingga menghasilkan kinerja keseluruhan yang lebih baik dibandingkan dengan model tunggal. Singkatnya, penerapan *Stacking Ensemble Learning* dalam memprediksi kinerja akademik dapat memberikan manfaat signifikan bagi institusi pendidikan dengan meningkatkan akurasi prediksi, meningkatkan efisiensi pemrosesan data, dan pada akhirnya berkontribusi terhadap kualitas pendidikan dan keberhasilan siswa.

Metode stacking ensemble terdiri dari dua level, yaitu level 0 dan level 1. (Shintabella et al., 2024).

Pada level 0, dilakukan pelatihan model dasar secara mandiri, di mana setiap model dasar menghasilkan prediksi secara terpisah. Dalam penelitian ini, model dasar yang pertama kali digunakan adalah Support Vector Machine (SVM) yang

bertindak sebagai *support vector classifier*. SVM ini menggunakan pendekatan *One vs All* untuk menangani klasifikasi pada dataset multinomial. Model dasar kedua yang diterapkan adalah *Random Forest*, yang menggunakan pendekatan hutan acak untuk menghasilkan prediksi dengan menggabungkan beberapa pohon keputusan. Selanjutnya pada level 1 dari *stacking ensemble*, dilakukan penggabungan antara model dasar dan model meta untuk meningkatkan akurasi prediksi akhir.

Pada tahap awal level 0, data dilatih secara independen dengan model dasar SVM, *Random Forest*, dan KNN. Hasil prediksi dari model dasar di level 0 kemudian digunakan sebagai input ke model meta pada level 1. Untuk kasus dengan data multinomial, teknik XGBoost diterapkan sebagai model meta. Prediksi akhir dari teknik *stacking ensemble* ini merupakan hasil kombinasi antara dua model dasar (SVM dan *Random Forest*) dan tiga kombinasi model dasar (RD, SVM, dan KNN) menjadi model meta (XGBoost), menghasilkan prediksi optimal melalui pendekatan *ensemble stacking*. Gambar 1 menggambarkan konfigurasi atau tahapan *stacking ensemble* dalam penelitian ini.

Penggunaan *stacking ensemble* dalam data pendidikan memiliki banyak kelebihan. Pertama, teknik ini dapat menggabungkan kekuatan model SVM, RF, dan KNN, yang masing-masing unggul dalam menangani pola non-linear, data tidak seimbang, serta klasifikasi berbasis jarak (Gusnina et al., 2022; Yağcı, 2022). Kedua, *stacking* cocok untuk klasifikasi multikelas, seperti kelulusan tepat waktu, terlambat, atau gagal, yang umum dalam sistem pendidikan tinggi (Ouatik et al., 2022). Namun, tantangannya mencakup kompleksitas komputasi, risiko kesalahan berantai dari *base learner* ke *meta learner*, dan kebutuhan interpretasi lanjutan

seperti SHAP untuk menjelaskan hasil prediksi (Mahboob et al., 2023; Sahlaoui et al., 2021). Meski demikian, keunggulan *stacking* dalam relevansi prediksi terhadap data aktual menjadikannya metode unggulan untuk meningkatkan sistem rekomendasi di lembaga pendidikan tinggi.

7.7. SMOTE

Ketidakseimbangan kelas (*class imbalance*) merupakan salah satu isu krusial dalam pemodelan *machine learning*, khususnya pada penelitian pendidikan tinggi di mana jumlah mahasiswa pada kategori tertentu sering jauh lebih sedikit daripada kategori lainnya. Kondisi ini dapat menyebabkan model belajar lebih dominan dari kelas mayoritas sehingga mengabaikan pola penting pada kelas minoritas. Untuk mengatasi hal tersebut, salah satu teknik yang paling banyak digunakan adalah *Synthetic Minority Oversampling Technique* (SMOTE).

SMOTE pertama kali diperkenalkan oleh Chawla et al. (2002) sebagai metode *oversampling* berbasis sintesis, yaitu menciptakan contoh baru pada kelas minoritas dengan melakukan interpolasi antar titik data yang saling berdekatan. Berbeda dari teknik *oversampling* konvensional seperti duplikasi data, SMOTE menghasilkan sampel baru yang berada di antara sampel minoritas yang sudah ada sehingga dapat membantu model mempelajari batas keputusan (*decision boundary*) dengan lebih baik. Dengan demikian, model tidak hanya terhindar dari *overfitting* yang sering muncul akibat duplikasi data, tetapi juga memiliki persebaran data yang lebih representatif.

Dalam konteks penelitian pendidikan, penggunaan SMOTE telah terbukti membantu meningkatkan kualitas prediksi ketika dijumpai kelas yang tidak seimbang. Penelitian Begum et al. (2021), misalnya, menunjukkan bahwa pemanfaatan SMOTE pada data pra-kuliah mampu memperbaiki distribusi kelas sebelum proses pembelajaran algoritma *ensemble* yang mereka kembangkan. Hasil penelitian tersebut menegaskan bahwa pembelajaran pada dataset pendidikan sering mengalami ketidakseimbangan kategori hasil belajar, sehingga SMOTE berperan penting dalam memastikan model tidak bias pada kelompok tertentu.

Secara umum, SMOTE bekerja melalui tiga langkah utama:

1. menentukan tetangga terdekat (*k*-nearest neighbors) untuk setiap sampel kelas minoritas,
2. memilih secara acak salah satu tetangga tersebut, dan
3. menghasilkan sampel baru di sepanjang garis penghubung antara dua titik.

Pendekatan ini tidak hanya membantu menyeimbangkan jumlah data antar kelas, tetapi juga memperluas ruang fitur sehingga model memiliki kesempatan memahami pola minoritas secara lebih mendalam. Meskipun penelitian ini mengadopsi *stratified* sampling pada proses pembagian data, SMOTE juga ditambahkan sebagai landasan teoritis karena teknik ini merupakan praktik umum dalam *educational data mining* untuk meningkatkan kemampuan generalisasi model, terutama pada skenario multi-kelas seperti prediksi status kelulusan.

7.8. Evaluasi Matriks dalam Pembelajaran Mesin

Matriks evaluasi adalah serangkaian metrik yang digunakan untuk menilai performa model pembelajaran mesin. Metrik ini berperan penting dalam menentukan apakah model sudah berfungsi optimal atau memerlukan peningkatan. Matriks evaluasi yang umum digunakan dalam model prediksi pembelajaran mesin mencakup akurasi, presisi, *recall*, *f1-score*. (Shintabella et al., 2024)

4. Akurasi

Akurasi menunjukkan persentase prediksi yang benar dari keseluruhan prediksi yang dilakukan model. Nilai akurasi berkisar antara 0 hingga 1, di mana 1 menunjukkan prediksi yang sempurna, sedangkan 0 menunjukkan prediksi sepenuhnya tidak akurat. Rumus akurasi menggunakan TP (true positive), TN (true negative), FP (false positive), dan FN (false negative) sebagai dasar perhitungan.

$$\text{Akurasi} = \frac{TP + TN}{TP + FN + FP + TN} \quad (9)$$

5. Presisi

Presisi adalah perbandingan antara jumlah prediksi positif yang benar dengan total prediksi positif yang dihasilkan model. Metrik ini menunjukkan seberapa akurat model dalam mengidentifikasi data positif dari keseluruhan prediksi positif yang diberikan. Nilai presisi berkisar dari 0 hingga 1, dengan 1 sebagai nilai optimal.

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (10)$$

6. Recall

Recall mengukur rasio prediksi positif yang benar dibandingkan dengan jumlah total data positif yang sesungguhnya. Metrik ini memberikan gambaran mengenai seberapa baik model dalam mengidentifikasi semua data positif di dataset. Nilai recall berkisar antara 0 dan 1, dengan 1 menunjukkan performa ideal.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (11)$$

7. F1-Score

F1-score adalah metrik yang menggambarkan keseimbangan antara presisi dan recall. F1-score yang tinggi mengindikasikan bahwa model memiliki keseimbangan yang baik antara kedua metrik tersebut. Nilai F1-score berkisar antara 0 dan 1, di mana 1 adalah hasil yang ideal, menunjukkan bahwa model memiliki presisi dan recall yang optimal.

$$F1\text{-Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (12)$$

8. Area di Bawah Kurva ROC (AUC)

AUC (Area Under the Curve) adalah metrik yang mengukur kemampuan model untuk membedakan antara kelas positif dan negatif. AUC merepresentasikan hubungan antara *True Positive Rate* (recall) dan *False Positive Rate* (1-specificity) pada berbagai ambang batas klasifikasi. Metrik ini

memberikan gambaran kinerja model di semua ambang batas, sehingga sangat berguna untuk membandingkan performa berbagai model. Semakin tinggi nilai AUC, semakin baik kemampuan model dalam membedakan kelas (mendekati nilai 1).

AUC dihitung sebagai area di bawah kurva ROC menggunakan metode numerik, seperti aturan trapesium. Rumus perhitungannya melibatkan koordinat pada sumbu *False Positive* (FP) dan *True Positive* (TP) (Alzahrani, 2024).

Rumus:
$$AUC = \sum_i^{N-1} (X_{i+1} - X_i) \left(\frac{y_{i+1} + y_i}{2} \right) \quad (13)$$

Keterangan:

- X_i dan X_{i+1} adalah titik berurutan pada sumbu FP,
- y_i dan y_{i+1} adalah titik berurutan pada sumbu TP.

Rumus ini pada menghitung luas segmen linier yang membentuk kurva ROC, dengan memberikan pandangan menyeluruh tentang kinerja model

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian kuantitatif dengan pendekatan eksperimental komputasi menggunakan analisis komparatif. Fokus penelitian adalah pada karakteristik data dengan menerapkan teknik eksperimen menggunakan komputer sebagai alat utama. Tujuannya adalah menggabungkan performa kombinasi model prediksi Random Forest, SVM, dan KNN dengan menerapkan *stacking ensemble* menggunakan model XGBoost dalam konteks prediksi kelulusan calon mahasiswa baru.

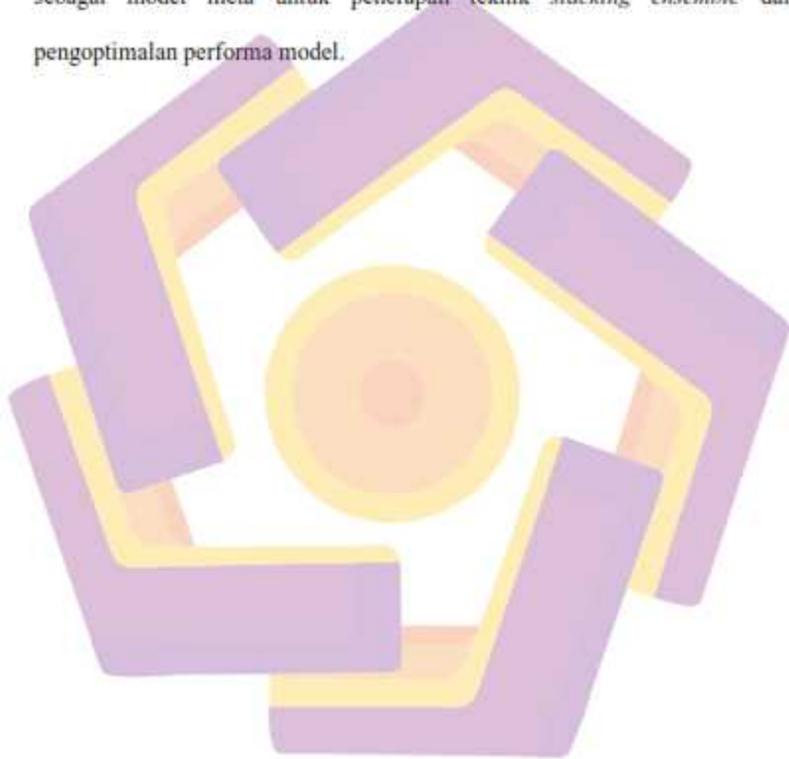
3.2. Metode Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh berbagai informasi yang diperlukan dalam penelitian. Data yang dikumpulkan mencakup landasan teori mengenai model algoritma yang akan diterapkan, serta pembelajaran literatur yang relevan dari penelitian sebelumnya. Dataset yang digunakan berasal dari data akses portal penerimaan mahasiswa baru dan portal akademik yang tersimpan di server lokal Universitas Slamet Riyadi serta data lulusan/DO yang dapat diakses melalui neofeeder. Data tersebut dikelola oleh bagian pusat data Unit Pelaksana Teknis Teknologi Informasi, Komunikasi, dan Komputer (UPT TIKK).

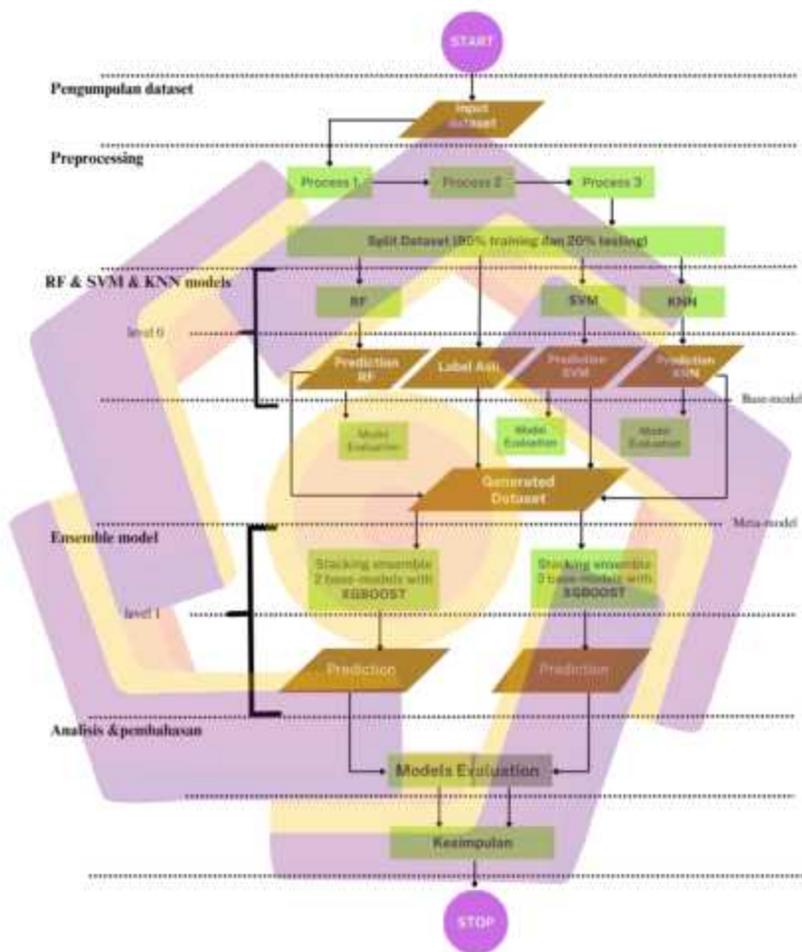
3.3. Metode Analisis Data

Setelah data terkumpul, langkah berikutnya adalah menganalisis data. Beberapa proses yang perlu dilakukan yaitu *pre-processing* data, *splitting* data, *training* data, dan menghitung korelasi antar variabel. Setelah melakukan proses-

proses tersebut maka dapat dilanjutkan untuk melakukan analisis kinerja model dari algoritma untuk mengubah data menjadi informasi yang dibutuhkan. Analisis yang dilakukan meliputi evaluasi kinerja model *Random Forest*, *SVM*, dan *KNN* dalam prediksi awal performa calon mahasiswa, selanjutnya menerapkan *XGBoost* sebagai model meta untuk penerapan teknik *stacking ensemble* dalam pengoptimalan performa model.



3.4. Alur Penelitian



Gambar 3. 1 Alur Penelitian *Stacking Ensemble*

Pada penelitian ini sesuai pada Gambar alur 1 di atas dapat dijelaskan sebagai berikut:

a. Pengumpulan dataset

Dataset yang digunakan adalah data yang didapatkan dari database portal pendaftaran mahasiswa S1 angkatan tahun 2019 dan data pelaporan PDDikti dari tahun 2020 sampai dengan tahun 2024 melalui neofeeder. Dataset yang terkumpul berupa 3 jenis file csv, csv pendaftaran dari sistem internal, csv data mahasiswa angkatan 2019, dan csv data kelulusan yang terlapor dari neofeeder PDDikti.

Secara rinci, dataset dalam penelitian ini berasal dari sistem internal universitas dan laman pelaporan PDDikti. Berkas data `backup_mahasiswa.csv` merupakan data utama mahasiswa yang diperoleh dari sistem akademik internal, dengan jumlah 41.186 baris data dan 56 kolom yang mencakup informasi identitas, akademik, demografis, serta latar belakang keluarga mahasiswa.

Selanjutnya, data status kelulusan mahasiswa diperoleh dari lima berkas `DAFTAR MAHASISWA LULUS / DO PER TAHUN KELUAR.csv` yang diunduh dari laman pelaporan PDDikti melalui NeoFeeder untuk periode tahun 2020 sampai 2024, dengan jumlah data masing-masing 1.037 (2020), 2.365 (2021), 1.102 (2022), 1.202 (2023), dan 1.551 (2024) baris data, serta 9 kolom pada setiap berkas.

Selain itu, berkas `cama.csv` digunakan sebagai data pendukung calon mahasiswa baru, yang terdiri dari 3.257 baris data dan 95 kolom, berisi informasi pendaftaran, nilai ujian masuk, latar belakang pendidikan sekolah asal, serta

kondisi sosioekonomi keluarga. Seluruh berkas tersebut digabungkan dan diselaraskan berdasarkan identitas mahasiswa (NIM) untuk membentuk dataset awal penelitian.

Dataset pertama terkumpul dengan filter data mahasiswa S1 angkatan 2019, yang berasal dari gabungan 3 file csv, berjumlah 1418 data mahasiswa. Setiap entri data mewakili satu individu yang kemudian data dilakukan *cleaning* dan pengelompokan dengan kategori lulus tepat waktu, lulus terlambat, dikeluarkan, dan mutasi. Data yang akan dipergunakan untuk prediksi tidak termasuk mahasiswa yang berstatus mutasi dan tanpa status.

Selanjutnya, fitur-fitur yang digunakan dalam model prediktif ini terdiri dari kombinasi atribut demografis, akademik, dan sosioekonomi. Atribut demografis meliputi jenis kelamin dan jumlah anak kandung. Atribut akademik terdiri dari nilai rapor, dan nilai ujian seleksi masuk. Sementara itu, aspek sosioekonomi mencakup pekerjaan orang tua dan penghasilan bulanan orang tua serta atribut administratif seperti periode pendaftaran, agama, dan asal provinsi. Seluruh variabel telah dikodekan secara numerik untuk mendukung analisis *machine learning*. Variabel target adalah Status Kelulusan, yang dikategorikan menjadi tiga kelas sebagaimana disebutkan di atas. Sehingga, data yang terkumpul di awal wajib melalui proses *cleaning* dan analisa mulai dari pendaftaran hingga memperoleh status keluar.

Proses validasi instrumen dataset ini dilakukan melalui validasi internal (academic department approval) di Biro Administrasi Akademik dan *cross-check* ke database resmi (administrative database) di Unit Pelaksana Teknis

Teknologi Informasi Komunikasi dan Komputer (UPT TIKK) sebagai bentuk validasi isi serta untuk memastikan integritas, dan akurasi data privat, sebagaimana diadopsi dalam penelitian sebelumnya (Fernandez-Garcia et al., 2021; Herianto et al., 2024). Selain itu, validasi statistik dilakukan dengan analisis korelasi antar fitur dan penghapusan atribut yang tidak relevan, selaras dengan metode yang diterapkan di penelitian sebelum ini (Fernandez-Garcia et al., 2021; Yadav* & Bhole, 2020). Tiga file yang berisi data pendaftaran, data akademik, dan data kelulusan digabungkan berdasarkan atribut identitas, hingga menghasilkan dataset awal. Data gabungan tersebut perlu dimuat ke dalam program dan dilanjutkan ke tahapan yang bernama *preprocessing*.

b. *Preprocessing*

Pada tahap ini, terdapat beberapa proses yang dapat dilakukan menggunakan pustaka Python seperti Pandas dan Numpy yaitu mulai dari *cleaning data*, konversi kategori ke numerik, dan normalisasi. Proses *preprocessing* yang dilakukan mencakup tahap *cleaning*, validasi statistik melalui analisis korelasi antar fitur, penghapusan atribut redundan, serta penerapan teknik *feature engineering* yang tepat, seperti transformasi variabel (Fernandez-Garcia et al., 2021). Proses *cleaning* dilakukan dengan membersihkan data duplikat, data dengan nilai kosong, dan status tidak diketahui (Fernandez-Garcia et al., 2021; Herianto et al., 2024). Pada tahap *feature transformation* dan normalisasi yang dilakukan berguna untuk menyesuaikan jenis data dengan kebutuhan algoritma. Pertama penggunaan *frequency encoding* untuk karakteristik data kategorikal dengan data berjenis

label, selanjutnya fitur berkarakteristik kategorikal ordinal atau fitur yang terdapat urutan penting antar kategori menggunakan *OrdinalEncoder*, dan terakhir fitur numerik seperti *rata_nil_skh*, *pointterima*, dan *anakkandung* dinormalisasi menggunakan *MinMaxScaler* agar relasi antar nilai terjaga dan nilai berada dalam skala yang seragam, sehingga mendukung stabilitas dalam proses pelatihan model klasifikasi. Kemudian validasi statistik melalui matriks korelasi dihitung menggunakan *Cramer's V* yang mengukur hubungan antar fitur diskret (nilai 0 - 1) (Fernandez-Garcia et al., 2021).

Setelah seluruh proses *preprocessing* selesai, dilakukan pembagian dataset (*data splitting*) menjadi data latih dan data uji. Langkah ini menjadi dasar untuk masuk ke tahap berikutnya, yaitu pembangunan model prediksi dengan pendekatan *Random Forest (RF)*, *Support Vector Machine (SVM)*, dan *K-Nearest Neighbor (KNN)* sebagai model dasar.

c. *Split data*

Pertama-tama dataset yang tersimpan dan sudah melalui *preprocessing* akan dipisahkan menjadi 8:2 data training mahasiswa dan data testing mahasiswa.

d. *Pemodelan Level - 0*

Proses pemodelan level dasar (*level-0*) yang didalamnya menggunakan tiga algoritma sebagai base-model, yaitu *Random Forest*, *K-Nearest Neighbors*, dan *Support Vector Machine*. Setiap model dasar dilatih menggunakan skema *five-fold stratified* untuk menghasilkan *out-of-fold meta-features*. *SMOTE* hanya diterapkan pada bagian data pelatihan (*train*) di setiap *fold* untuk mengatasi ketidakseimbangan kelas, sedangkan prediksi pada *validation fold* disimpan

sebagai meta-fitur tanpa manipulasi *oversampling*. Setelah seluruh fold selesai, masing-masing *base-model* dilatih kembali menggunakan seluruh data pelatihan yang telah di-SMOTE. Model final disimpan untuk digunakan pada tahap prediksi.

e. Random Forest Model

Random Forest digunakan sebagai model prediksi dasar yang digunakan untuk melakukan prediksi awal. Model RF diterapkan pada level 0 penelitian ini, yang dilakukan setelah proses impor pustaka, proses memuat data yang telah melalui *preprocessing*, dan proses *splitting* data. Model Random Forest kemudian dilatih menggunakan *train set*, dan performanya diuji menggunakan *test set* sesuai pada level 0 yang dijelaskan sebelumnya.

f. SVM Model

Model SVM digunakan sebagai model dasar kedua dalam prediksi kelulusan calon mahasiswa. Algoritma ini berfokus pada mencari *hyperplane* yang memisahkan data ke dalam kelas yang berbeda dengan margin maksimum. Cocok untuk dataset dengan dimensi tinggi pada dataset pendaftaran mahasiswa. Setelah mengimpor pustaka yang diperlukan dan melakukan *preprocessing*, model dilatih menggunakan *train set* dan diuji dengan *test set* yang dijelaskan sesuai dengan proses level 0.

g. KNN Model

KNN adalah algoritma ketiga yang digunakan pada *base-model* prediksi status kelulusan calon mahasiswa. Model ini bekerja dengan menghitung jarak antara data baru dan data yang sudah ada menggunakan metrik seperti *Euclidean*

distance. Data baru kemudian diklasifikasikan berdasarkan kelas mayoritas dari k tetangga terdekatnya. KNN cocok untuk dataset non-linear dan digunakan dalam analisis prediktif, seperti memproyeksikan kelulusan mahasiswa berdasarkan atribut demografis, akademik, dan sosioekonomi. Langkah-langkah penerapan KNN meliputi impor pustaka, *preprocessing* data, normalisasi, serta pembagian data menjadi *train set* dan *test set* sesuai pada proses level 0.

h. Evaluasi Base-model

Yaitu melakukan tahapan evaluasi model pertama dari hasil prediksi tiga algoritma dasar yaitu RF, SVM, dan KNN menggunakan metrik akurasi, presisi, *recall*, dan *F1-score*.

i. Pemodelan Level - 1

Pada tahap ini, pemodelan level meta (level-1) menggunakan XGBoost dibangun dengan beberapa kombinasi fitur, yang terdiri dari gabungan prediksi *base-model* dengan fitur awal. Pada *training set*, fitur asli digabungkan dengan meta fitur hasil *Out of Fold (OOF)*, kemudian dilakukan SMOTE ulang. Pada tahap ini ditambahkan sedikit *noise injection* untuk mencegah kecenderungan model meta belajar terlalu spesifik pada pola latih. Meta model, XGBoost, kemudian dilatih sebagai penggabung utama dari seluruh informasi pada level dasar.

j. Stacking *Ensemble learning*

Untuk mengoptimalkan prediksi tiga algoritma dasar, yaitu Random Forest (RF), SVM, dan KNN, akan dikombinasikan menggunakan metode *Stacking*

Ensemble. Proses ini dilakukan dalam pemodelan Level 1 dalam dua tahap terpisah, yaitu menguji performa kombinasi dua dan tiga pasangan model dasar.

Pada tahap kombinasi dua model dasar, diantaranya *Random Forest* dengan SVM, *Random Forest* dengan KNN, dan KNN dengan SVM, nantinya hasil prediksi dari masing-masing kombinasi model akan digunakan sebagai fitur tambahan dari fitur asli yang disebut *meta feature*. Kemudian *meta features* yang sudah terbentuk tersebut dapat digunakan sebagai input pada meta-model XGBoost. Tahap selanjutnya adalah mengkombinasikan ketiga *base-model* (SVM, Random Forest, dan KNN). Sama seperti tahap pertama, prediksi dari ketiga model ini akan digabungkan dan dianalisis oleh meta-model XGBoost. Setiap iterasi akan diuji secara terpisah untuk mengevaluasi performanya dalam meningkatkan performa prediksi.

Pendekatan *stacking* ini berfokus pada pengolahan data yang kompleks dengan memanfaatkan kombinasi hasil prediksi dari berbagai model dasar. Hasil *prediction* tidak hanya mengintegrasikan pola-pola unik dari setiap model, tetapi juga memanfaatkan meta-model untuk menangkap hubungan non-linear yang sulit diidentifikasi oleh model tunggal.

k. XGBoost

XGBoost dipilih sebagai model meta karena berbagai keunggulannya. Salah satunya adalah mekanisme regulasi yang kuat, yang membantunya menghindari *overfitting* dan memberikan performa yang baik pada data yang tidak terlihat. Selain itu, XGBoost dirancang untuk bekerja secara efisien dengan dataset besar, menjadikannya sangat cocok untuk aplikasi dengan banyak fitur. Model ini juga

mampu memberikan informasi tentang pentingnya fitur, yang memungkinkan analisis lebih mendalam terhadap prediktor-prediktor yang paling berkontribusi dalam pengambilan keputusan.

Dalam penelitian ini, peneliti melakukan pelatihan model meta menggunakan XGBoost dalam konteks *stacking ensemble*, dengan tiga model dasar yaitu *Support Vector Machine (SVM)*, *Random Forest (RF)*, dan *K-Nearest Neighbors (KNN)*. Tujuan dari pendekatan ini adalah untuk membandingkan performa prediksi dari dua dan tiga model dasar tersebut menggunakan teknik *stacking ensemble*, sehingga diharapkan dapat mengoptimalkan prediksi final. Proses *stacking* memanfaatkan kekuatan masing-masing model dasar, yang diharapkan dapat meningkatkan stabilitas dan akurasi prediksi.

Alur penggunaan XGBoost dalam model *stacking* dimulai dengan mengumpulkan hasil prediksi dari tiga model dasar, yaitu SVM, RF, dan KNN. Hasil dari model prediksi awal digabungkan dengan label asli, kemudian disusun menjadi dataset baru yang akan digunakan untuk model meta. Selanjutnya, model XGBoost dapat dibangun dengan mengatur parameter yang relevan, kemudian melatihnya menggunakan dataset meta yang telah disiapkan.

1. *Prediction*

Prediction dalam konteks ini adalah perubahan hasil dari keluaran prediksi model dasar menjadi pola prediksi baru yang lebih baik pada performa calon mahasiswa baik yang berasal dari *training set* maupun dari *testing set*.

Pada *train set* meta model dilatih menggunakan gabungan fitur asli dengan Meta-features OOF kemudian dilakukan SMOTE pada dataset gabungan tersebut beserta tambahan noise kecil. Berbeda dengan *test-set*, pada *testing set* model dasar yang telah dilatih dalam melakukan prediksi akan menghasilkan meta-fitur. Kemudian meta-fitur tersebut digabungkan dengan fitur asli, lalu dimasukkan ke dalam meta-model untuk menghasilkan prediksi akhir terhadap data uji. Pada tahap ini tidak diterapkan SMOTE agar model diuji secara apa adanya terhadap distribusi data sebenarnya.

m. Evaluasi Meta-model

Tahapan ini dilakukan untuk mengevaluasi performa model meta dengan menggunakan metrik evaluasi, yaitu akurasi, presisi, *recall*, *F1-score*, dan AUC-ROC untuk menilai kinerja model dalam memprediksi keputusan status kelulusan dengan tepat (Jan Melvin Ayu Soraya Dachi & Pardomuan Sitompul, 2023).

n. Analisa dan pembahasan perbandingan evaluasi hasil

Tahap ini membandingkan kinerja model dasar dengan berbagai kombinasi *stacking*, mengevaluasi dampak SMOTE, serta mengkaji stabilitas generalisasi melalui perbedaan performa latih-uji. Selain itu, pembahasan juga mencakup analisis fitur yang paling berpengaruh menggunakan SHAP serta mengetahui bobot pengaruh antar kategori terhadap performa model.

Pertama, mengenai Analisa perbandingan model. Dimulai dari model dasar SVM, RF, dan KNN dievaluasi secara terpisah atau evaluasi yang dilakukan pertama untuk mengukur akurasi, presisi, *recall*, *F1-score* masing-masing. Hasil

ini memberikan gambaran awal tentang kemampuan setiap model dalam mengatasi masalah klasifikasi. Selanjutnya, hasil prediksi dari model dasar tersebut digabungkan melalui teknik *stacking* kombinasi dua dan tiga model dasar menggunakan XGBoost, dari hasil prediksi tersebut dilakukan evaluasi kedua. Hasil evaluasi model yang kedua dianalisa secara terpisah sesuai masing-masing pasangan kombinasi. Kemudian, hasil evaluasi dari model dasar dan hasil evaluasi dari model kombinasi (dua model dan tiga model) dibandingkan untuk dilihat performa penerapan model *stacking* menggunakan XGBoost.

Selain mengevaluasi performa, penelitian ini juga mengkaji kemampuan model dalam melakukan generalisasi melalui performa latihan-uji (*generalization gap*). Du et al. (2024) menyatakan bahwa *gap* yang lebih sempit menunjukkan kemampuan generalisasi yang lebih baik. Literatur lainnya juga menerangkan bahwa *gap* performa kecil antara *training* dan *test* sering berkorelasi dengan model yang dapat melakukan generalisasi dengan baik (Ballester et al., 2022; Jiang et al., 2018; Azar et al., 2024). Studi empiris di berbagai domain (radiomics dan klasifikasi) juga memperlihatkan bahwa *gap* kecil biasanya terkait dengan model yang tidak mengalami *overfitting* yang berat. Oleh karena itu, pada konteks data kami, selisih minim digunakan sebagai indikator empiris bahwa model berpotensi memiliki generalisasi yang baik, dan hasil ini dikonfirmasi dengan analisis kurva belajar dan validasi silang.

Selanjutnya, analisis penting dilakukan pada tingkat fitur melalui SHAP. Pendekatan ini digunakan untuk menilai kontribusi fitur asli maupun meta-fitur terhadap prediksi akhir. Hasil analisis SHAP membantu mengidentifikasi faktor-

faktor yang paling berpengaruh dalam menentukan status kelulusan mahasiswa, baik yang bersifat akademik, demografis, maupun fitur turunan dari *base-model*.

Secara keseluruhan, hasil analisis menunjukkan bahwa *stacking ensemble* memberikan peningkatan yang jelas dibandingkan model tunggal, baik dari sisi akurasi, konsistensi prediksi, maupun kemampuan model untuk menangani kompleksitas klasifikasi multi-kelas. Setelah seluruh evaluasi internal dilakukan, tahap selanjutnya adalah menguji relevansi hasil prediksi terhadap data aktual kelulusan mahasiswa.

o. Evaluasi Relevansi Prediksi terhadap Data Aktual

Evaluasi relevansi dilakukan untuk menilai sejauh mana hasil prediksi model *stacking ensemble* yang dikembangkan mampu merepresentasikan kondisi kelulusan mahasiswa secara nyata di perguruan tinggi. Tahap ini berfokus pada perbandingan hasil prediksi model terhadap data aktual kelulusan mahasiswa yang telah tercatat dalam sistem akademik. Pendekatan ini penting untuk memastikan bahwa model tidak hanya unggul secara statistik, tetapi juga memiliki kemampuan praktis dalam mencerminkan realitas akademik di lapangan.

3.5. Justifikasi Penelitian

Data yang digunakan pada tahap ini berasal dari angkatan mahasiswa tahun 2019, yang dipilih karena telah menyelesaikan masa studi dan memiliki catatan kelulusan akhir yang lengkap. Pemilihan angkatan ini dilakukan secara sengaja (*purposive*) agar hasil validasi didasarkan pada data yang telah memiliki status kelulusan aktual, meliputi tiga kategori utama yaitu: lulus tepat waktu, lulus

terlambat, dan tidak lulus (dikeluarkan). Data pendaftaran tahun 2019 tersebut digunakan sebagai data historis input untuk pelatihan dan pengujian model, sedangkan status kelulusan yang diperoleh dari sistem akademik berfungsi sebagai *ground truth* atau acuan pembandingan.

Dalam pengembangan model prediksi kelulusan mahasiswa, pemilihan model dasar menjadi tahap krusial karena masing-masing algoritma memiliki keunggulan dan kelemahan tersendiri. Tiga algoritma utama yang digunakan sebagai *base learners* dalam penelitian ini adalah *Support Vector Machine* (SVM), *Random Forest* (RF), dan *K-Nearest Neighbor* (KNN). Ketiganya dipilih berdasarkan literatur yang menunjukkan performa tinggi dalam prediksi berbasis data pendidikan serta karakteristik yang saling melengkapi dalam menangani klasifikasi multikelas.

1. SVM memiliki keunggulan dalam menghasilkan margin klasifikasi yang optimal dan mampu menangani data berdimensi tinggi, menjadikannya cocok untuk prediksi dengan fitur-fitur kompleks (Li et al., 2021).
2. RF dipilih karena stabilitasnya dalam menghadapi data dengan missing values dan keunggulannya dalam menangani data yang tidak seimbang berkat mekanisme voting dari banyak decision trees (Fernández-García et al., 2021).
3. KNN memiliki kekuatan dalam mengenali pola dan memberikan prediksi berdasarkan kedekatan jarak, yang sesuai dengan konteks pendidikan di mana karakteristik individu mahasiswa cenderung bersifat heterogen (Mahboob et al., 2023).

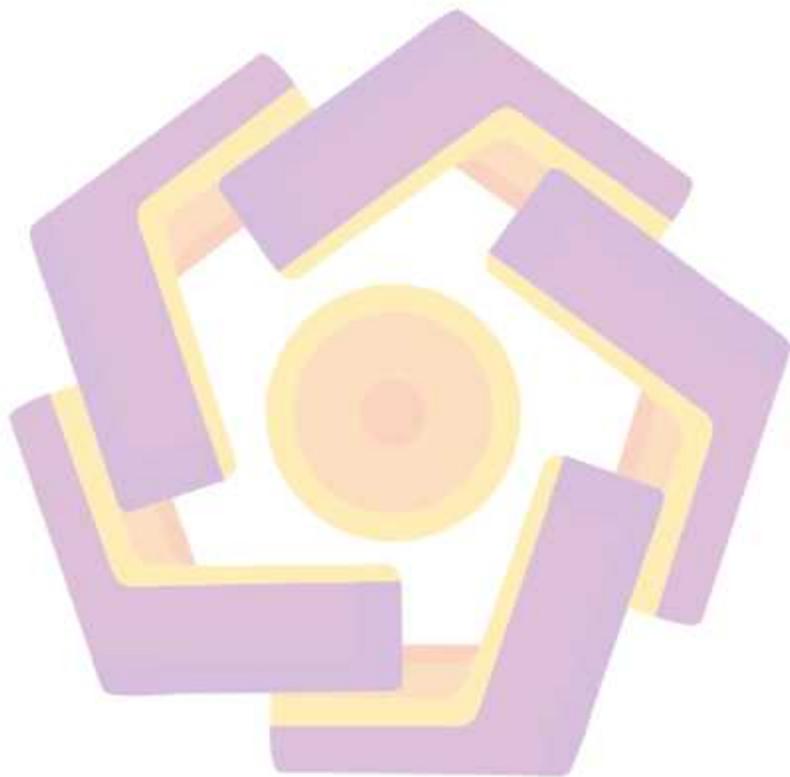
Ketiga model ini digunakan pada level-0 dalam arsitektur *stacking ensemble*. Sementara itu, XGBoost dipilih sebagai *meta-learner* pada level-1 karena kemampuannya melakukan regularisasi untuk menghindari *overfitting* dan keefektifannya dalam menangani data besar dan kompleks. Kombinasi ini memberikan sinergi antara kekuatan generalisasi, akurasi margin, serta sensitivitas terhadap pola lokal. Arsitektur *stacking* secara umum terdiri dari dua tingkat:

- a. Level-0 (base models): SVM, RF, dan KNN melakukan pelatihan secara paralel dan menghasilkan prediksi masing-masing.
- b. Level-1 (meta-model): XGBoost menerima output prediksi dari model-model dasar tersebut sebagai input baru, dan mempelajari cara terbaik untuk menggabungkannya dalam menghasilkan prediksi akhir.

Proses evaluasi dilakukan dengan cara mencocokkan hasil prediksi model terhadap data aktual kelulusan. Tingkat kesesuaian klasifikasi dievaluasi menggunakan metrik seperti akurasi aktual (*actual accuracy*), *precision*, *recall*, *F1-score*, dan *confusion matrix* untuk menggambarkan distribusi prediksi benar dan salah antar kategori kelulusan. Selain itu, hasil perbandingan juga dianalisis untuk melihat adanya pola ketidaksesuaian, misalnya kesalahan model dalam membedakan mahasiswa yang lulus tepat waktu dan yang terlambat. Analisis ini memberikan gambaran mengenai keterbatasan model dalam menangkap variabel non-akademik yang mungkin berpengaruh pada hasil studi mahasiswa.

Hasil dari tahap evaluasi relevansi ini diharapkan dapat menunjukkan sejauh mana model yang dikembangkan benar-benar mampu menggambarkan kenyataan performa akademik mahasiswa. Dengan demikian, model *stacking ensemble* yang

dihasilkan tidak hanya berfungsi sebagai alat prediksi berbasis data, tetapi juga dapat menjadi referensi awal bagi lembaga pendidikan dalam memahami profil mahasiswa dan potensi keberhasilannya sejak tahap pendaftaran.



BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Pendekatan Penelitian

Penelitian ini diawali dengan serangkaian observasi pada bagian akademik Universitas Slamet Riyadi, khususnya terhadap data kelulusan mahasiswa program sarjana angkatan 2019. Observasi dilakukan secara langsung melalui sistem pencatatan akademik universitas untuk memperoleh gambaran umum mengenai pola kelulusan mahasiswa. Hasil observasi menunjukkan adanya variasi capaian studi, mulai dari mahasiswa yang berhasil lulus tepat waktu, mahasiswa yang membutuhkan tambahan semester untuk menyelesaikan studinya, hingga mahasiswa yang tidak menyelesaikan pendidikan karena dikeluarkan dari program. Informasi tersebut menjadi dasar awal dalam mengidentifikasi faktor-faktor yang mungkin berperan dalam menentukan keberhasilan akademik mahasiswa.

Selain mengamati capaian kelulusan, penelitian ini juga menekankan pada proses pemahaman data sejak tahap penerimaan mahasiswa baru. Data yang dikumpulkan pada saat seleksi, seperti nilai akademik awal, latar belakang keluarga, serta informasi demografis, dianalisis untuk melihat keterkaitannya dengan status kelulusan mahasiswa angkatan 2019. Pendekatan ini memungkinkan pengembangan model prediktif yang dapat memproyeksikan status kelulusan secara dini, bahkan sebelum mahasiswa menjalani perkuliahan. Dengan cara tersebut, institusi pendidikan dapat merancang intervensi akademik yang lebih tepat sasaran bagi mahasiswa yang terindikasi berisiko mengalami keterlambatan atau kegagalan studi.

Selanjutnya, dilakukan telaah literatur melalui berbagai artikel ilmiah, buku, dan publikasi yang relevan dengan topik prediksi kelulusan. Kajian pustaka ini bertujuan memperkuat kerangka teoritis dan menyediakan perbandingan dengan penelitian terdahulu, khususnya yang mengaplikasikan pendekatan *machine learning* dalam konteks pendidikan tinggi. Penelitian-penelitian sebelumnya telah menunjukkan bahwa metode ensemble, seperti *stacking*, mampu meningkatkan akurasi prediksi dibandingkan model tunggal (Fernandez-Garcia et al., 2021; Herianto et al., 2024). Dengan demikian, studi ini berupaya mengintegrasikan pendekatan empiris berdasarkan data lokal dengan pijakan teoritis yang sudah teruji dalam literatur internasional.

4.2. Pemahaman Data

Pada penelitian ini, data diperoleh dari basis data sistem internal dan sistem pelaporan Universitas Slamet Riyadi, dengan tiga sumber utama dalam format CSV, yaitu data pendaftaran, data mahasiswa, dan data pelaporan kelulusan yang diunduh dari Neo Feeder. Data yang diambil adalah data mahasiswa Angkatan 2019 dengan proses awal pengolahan data yang dilakukan secara manual menggunakan Microsoft Excel, di mana data pendaftaran calon mahasiswa digabungkan dengan data mahasiswa hanya pada entri yang sudah memiliki Nomor Induk Mahasiswa (NIM). Selanjutnya, informasi kelulusan ditambahkan dari file CSV kelulusan PDDikti. Setelah dilakukan penggabungan, tahap berikutnya adalah menghapus data duplikasi pada file mahasiswa. Proses ini dilakukan dengan pengawasan serta validasi dari pihak akademik Universitas Slamet Riyadi, sehingga pada akhirnya

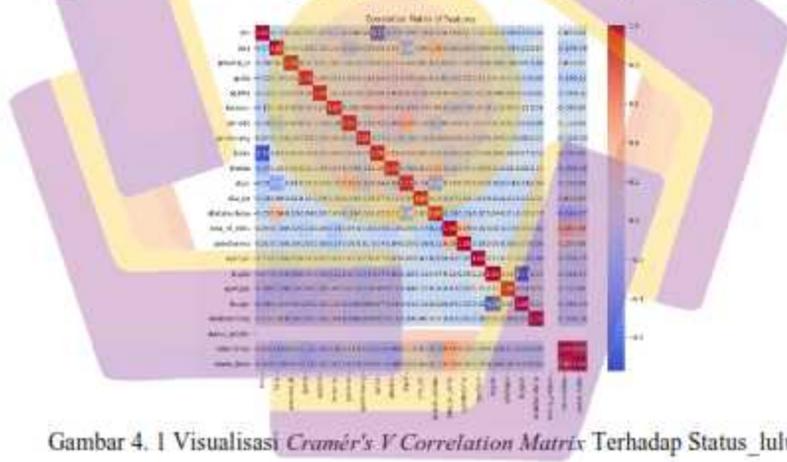
diperoleh 1418 baris data awal yang siap diproses lebih lanjut sebagai bahan penelitian prediksi dini kelulusan mahasiswa.

Dataset awal terdiri atas 1.418 entri mahasiswa Angkatan 2019 dengan 23 atribut yang merepresentasikan informasi demografis, akademik, administratif, serta status kelulusan akhir. Data tersebut disajikan pada Tabel 4.1 untuk memberikan gambaran umum komposisi data dan fitur-fitur yang direkam dari sistem pendaftaran, sistem akademik, serta sistem pelaporan kelulusan. Tabel ini berisi data awal sebelum dilakukan proses seleksi dan transformasi lanjutan. Dua atribut awal berupa NIM dan usia mahasiswa saat mendaftar, sedangkan variabel target ditetapkan pada status kelulusan yang mengklasifikasikan mahasiswa menjadi tiga kategori, yaitu lulus tepat waktu, lulus terlambat, dan dikeluarkan.

Tabel 4. 1 Dataset awal

nim	Usia	...	rata_ni skhu	pointe rma	ayah job	ibu job	ayah gaji	ibu gaji	anak kandung	status_ pddikti	lulus_ tahun	status_ lulus
1	21	...	74	41.25	6	6	1	1	2		2023	1
2	18	...	84	55.6	6	1	1	0	2	Lulus	2023	1
3	18	...	85	40.75	6	2	2	2	3	Lulus	2023	1
4	25	...	77	36.5	6	1	2	0	2	Lulus	2023	1
5	18	...	80	37.5	6	6	1	1	1	Lulus	2023	1
6	20	...	82	18.75	10	6	2	2	3	Lulus	2023	1
7	40	...	70	31.6	1	2	0	2	5			
8	18	...	89	55.75	6	1	1	0	2	Lulus	2023	1
9	18	...	85	65.55	8	1	1	0	3	Lulus	2023	1
...
1418	22	...	80	37.75	2	2	2	2	2	Lulus	2024	2

Untuk memastikan kesesuaian data dengan tujuan penelitian, dilakukan analisis awal terhadap seluruh atribut. Atribut yang bersifat identitas unik, seperti NIM, dihapus karena tidak memberikan kontribusi terhadap pola prediksi. Demikian pula, atribut seperti "statuspddikti" dan "tahun_kelulusan" dieliminasi karena secara langsung merepresentasikan label target. Analisis korelasi menggunakan matriks Cramér's V, Gambar 4.1, menunjukkan bahwa variabel "golda", "sltatahunlulus", "usia", dan "idkelas" memiliki keterkaitan yang lemah dengan status kelulusan, sehingga juga dikeluarkan dari pemodelan. Setelah tahap reduksi, diperoleh 15 atribut sebagai variabel independen dan satu variabel target dengan total 875 entri data yang layak digunakan untuk proses pembelajaran model.



Gambar 4. 1 Visualisasi *Cramér's V Correlation Matrix* Terhadap Status_lulus

Untuk memberikan gambaran yang lebih jelas mengenai karakteristik data yang digunakan, Tabel 4.2 menyajikan deskripsi dari masing-masing fitur yang telah diseleksi. Sebanyak 15 fitur hasil seleksi berdasarkan matriks korelasi terhadap variabel target kemudian dikodekan ke dalam format numerik agar dapat

diproses secara optimal oleh algoritma klasifikasi. Transformasi ini dirancang untuk memudahkan model dalam mengenali pola data, mengurangi potensi kesalahan prediksi, sekaligus meningkatkan efisiensi serta konsistensi kinerja model. Proses feature transformation dan normalisasi dilakukan sesuai karakteristik atribut. Variabel kategorikal dengan jenis label dikodekan menggunakan frequency encoding, sedangkan atribut kategorikal ordinal atau yang memiliki urutan penting antar kategori diolah dengan OrdinalEncoder. Sementara itu, fitur numerik seperti rata-rata nilai SKHU, skor seleksi masuk (pointterima), dan jumlah saudara kandung dinormalisasi menggunakan MinMaxScaler agar setiap nilai berada pada rentang yang seragam dan tetap mempertahankan proporsi antarvariabel. Pendekatan ini mendukung stabilitas model selama proses pelatihan klasifikasi (Sahlaoui et al., 2021; Bujang et al., 2021).

Tabel 4. 2 Deskripsi fitur

Atribut	Karakteristik	Type Data	Keterangan
provinsi_id	Kategorikal	Float	Provinsi domisili
agama	Kategorikal	Float	Agama pendaftar
kelamin	Kategorikal	Float	Jenis kelamin pendaftar Pria atau Wanita
periode	Kategorikal	Float	Periode pendaftaran Gasal atau Genap
gelombang	Kategorikal ordinal	Float	Gelombang pendaftaran : dini, I, II
kofak	Kategorikal	Float	Kode program studi di Unisri
stpid	Kategorikal	Float	Status pendaftaran: Mahasiswa Baru, Pindahan
slta_jur	Kategorikal	Float	Jurusan di SLTA : IPA, IPS
rata_nil_skhu	Numerik	Float	Rata-rata nilai pada SKHU saat SLTA
pointterima	Numerik	Float	Score / nilai ujian CBT
Ayahjob	Kategorikal	Float	Pekerjaan ayah
ibujob	Kategorikal	Float	Pekerjaan ibu

ayahgaji	Kategorikal ordinal	<i>Integer</i>	Penghasilan ayah: rendah, sedang, tinggi
ibugaji	Kategorikal ordinal	<i>Integer</i>	Penghasilan ibu: rendah, sedang, tinggi
anakandung	Numerik	<i>Float</i>	Jumlah saudara kandung calon mahasiswa
status_lulus	Kategorikal	<i>Float</i>	Status kelulusan: dikeluarkan, lulus tepat waktu, lulus terlambat

Pendekatan ini konsisten dengan penelitian sebelumnya yang menekankan pentingnya seleksi fitur untuk meningkatkan stabilitas dan efisiensi model prediksi mahasiswa (Bujang et al., 2021; Sahlaoui et al., 2021). Dengan dataset yang lebih ringkas dan representatif, model diharapkan mampu mengenali pola dengan lebih baik serta memiliki kemampuan generalisasi yang lebih kuat pada data baru, sebagaimana juga ditemukan dalam studi Fernandez-Garcia et al. (2021).

4.3. Preprocessing Data

Tahap preprocessing dilakukan untuk menyiapkan dataset sebelum masuk ke proses pemodelan. Setelah 1.418 baris data awal terkumpul, langkah pertama adalah persiapan dan pembersihan data melalui platform *Google Colaboratory* dengan bahasa pemrograman Python. Pendekatan ini sejalan dengan praktik penelitian sebelumnya yang menekankan pentingnya tahapan persiapan data untuk meningkatkan performa model prediksi berbasis *ensemble learning* (Herianto et al., 2024; Sahlaoui et al., 2021). Tahapan diawali dengan import library yang relevan, seperti NumPy dan pandas, untuk mendukung proses analisis data. Dataset CSV kemudian dipanggil ke dalam dataframe menggunakan fungsi `pd.read_csv()`

sehingga menghasilkan tampilan awal dataset berjumlah 1.418 entri, code dapat dilihat pada script di bawah ini.

```
import pandas as pd
import numpy as np
from google.colab import drive
drive.mount('/content/drive')
# Load dataset
file_path= "/content/drive/My
Drive/DATAmhs2019_23.06.csv"
df = pd.read_csv(file_path)
print(df.head())
```

Setelah melihat beberapa kolom pada dataset masih bertipe object. Oleh karena itu, langkah kedua dilakukan konversi ke dalam bentuk numerik (integer). Proses ini diawali dengan identifikasi kolom bertipe object menggunakan `df.select_dtypes(include='object').columns`, kemudian setiap kolom tersebut diubah dengan fungsi `pd.to_numeric()`. Parameter `errors='coerce'` digunakan agar data yang tidak dapat dikonversi otomatis diubah menjadi nilai NaN. Dengan cara ini, seluruh kolom dapat diproses lebih mudah pada tahap analisis selanjutnya. Code dan hasil output dapat dilihat pada Gambar 4.2 berikut.

```
# Identify object type columns
object_cols = df.select_dtypes(include='object').columns

# Iterate and convert to numeric, coercing errors to NaN
for col in object_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')
df.dtypes
print(df.dtypes.head(10))
```

```

Tipe data 10 kolom pertama:
nin          float64
Usia        int64
provinsi_id float64
golde       float64
agama       float64
kelamin     float64
periode     float64
gelombang   float64
kofak       float64
idkelas     float64
dtype: object

```

Gambar 4. 2 Tipe data.

Langkah ketiga adalah *feature transformation* dan normalisasi. Pada tahap ini berguna untuk menyesuaikan jenis data dengan kebutuhan algoritma. Pertama penggunaan *frequency encoding* untuk karakteristik data kategorikal dengan data berjenis label, selanjutnya fitur berkarakteristik kategorikal ordinal atau fitur yang terdapat urutan penting antar kategori menggunakan *OrdinalEncoder*, dan terakhir fitur numerik seperti *rata_nil_skh*, *pointterima*, dan *anakkandung* dinormalisasi menggunakan *MinMaxScaler* (0-10) agar relasi antar nilai terjaga dan nilai berada dalam skala yang sesuai dengan sifat data masing-masing sehingga mendukung stabilitas dalam proses pelatihan model klasifikasi. Code dan hasil output dapat dilihat pada Gambar 4.3 berikut.

```

# Frequency Encoding
categorical_cols = ['provinsi_id', 'agama', 'periode',
                  'kofak', 'stpid', 'sita_jur', 'ayahjob', 'ibujob']

for col in categorical_cols:
    if col in df.columns:
        # Calculate frequency
        fe = df[col].value_counts(normalize=True)
        # Create a mapping dictionary
        fe_map = fe.to_dict()
        # Replace column dengan frequency encoded values
        df[col] = df[col].map(fe_map)

```

```

# Kategorikal Ordinal menggunakan OrdinalEncoder
from sklearn.preprocessing import OrdinalEncoder
ordinal_cols = ['ayahgaji', 'ibugaji',
'gelombang', 'anakandung']
ordinal_encoder = OrdinalEncoder()
df[ordinal_cols] =
ordinal_encoder.fit_transform(df[ordinal_cols])

# Kategorikal NUMERIK dengan minmaxscaler
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 10))
numerical_cols = ['rata_nil_skh', 'pointterima',
'anakkandung']
for col in numerical_cols:
    if col in df.columns:
        # Pastikan kolom numerik
        if not pd.api.types.is_numeric_dtype(df[col]):
            df[col] = pd.to_numeric(df[col],
errors='coerce')
        # Terapkan normalisasi kalau benar-benar numerik
        if pd.api.types.is_numeric_dtype(df[col]):
            df[col] = scaler.fit_transform(df[[col]])

```

```

>>>

```

	nim	usia	provinsi_id	golde	agama	kelamin	periode
0	19100001.0	21	0.883395	2.0	0.852980	1.0	0.993635
1	19100002.0	18	0.872843	2.0	0.852980	0.0	0.993635
2	19100003.0	18	0.883395	3.0	0.862645	0.0	0.993635
3	19100004.0	25	0.821924	1.0	0.852980	1.0	0.993635
4	19100005.0	18	0.883395	2.0	0.893645	0.0	0.993635

	gelombang	kufek	idkelas	stpid	slra_jur	slstatabunlulus
0	0.0	0.183876	1.0	0.966761	0.632712	2.0
1	0.0	0.183876	1.0	0.966761	0.632712	1.0
2	0.0	0.183876	1.0	0.966761	0.632712	1.0
3	0.0	0.183876	2.0	0.966761	0.632712	2.0
4	0.0	0.183876	1.0	0.966761	0.632712	1.0

	rata_nil_skh	pointterima	ayah_job	ibu_job	ayahgaji	ibugaji
0	7.798233	5.892857	0.324247	0.186502	1.0	1.0
1	8.750000	7.942857	0.324247	0.486212	1.0	0.0
2	8.854167	5.821429	0.324247	0.066763	2.0	2.0
3	8.020833	5.214286	0.324247	0.486212	2.0	0.0
4	8.333333	5.357143	0.324247	0.186502	1.0	1.0

	anakkandung	status_gdtkiti	lulus_tahun	status_lulus
0	1.666667	huk	2023.0	1.0
1	1.666667	huk	2023.0	1.0
2	2.500000	huk	2023.0	1.0
3	1.666667	huk	2023.0	1.0
4	0.833333	huk	2023.0	1.0

Gambar 4. 3 Output encoding process

Setelah transformasi, perlu dilakukan analisis korelasi antarfitur untuk mengetahui seberapa kuat hubungan masing-masing variabel dengan label target status_lulus. Hasil analisis menunjukkan bahwa terdapat empat fitur dengan skor korelasi paling rendah terhadap variabel target, yaitu golda, sltatahunlulus, usia, dan idkelas seperti yang disampaikan pada tahap pengenalan data di awal. Nilai korelasi yang rendah ini mengindikasikan bahwa keempat fitur tersebut tidak memberikan kontribusi signifikan dalam menjelaskan variasi pada status kelulusan mahasiswa. Dengan kata lain, keberadaan fitur-fitur ini kurang relevan dalam mendukung proses prediksi. Code dan hasil output dapat dilihat pada Gambar 4.4 berikut.

```
correlation_matrix = df.corr()
# target 'status_lulus'
corr_with_target = correlation_matrix['status_lulus'].sort_values(ascending=False)
# Tampilkan hasil
print("Korelasi dengan status_lulus:")
print(corr_with_target)
```

```

Korelasi dengan status_lulus:
status_lulus      1.000000
lulus_tahun      0.945038
rata_rata_sks    0.358681
pointerima       0.084963
nra              0.043278
sita_jur         0.648524
periode         0.033001
kelamin         0.020115
stuid            0.029240
kofak           0.008268
provinsi_id     0.085772
ayahgaji        0.000007
gelombang      -0.000766
ibugaji         0.001070
agama          -0.018757
ibujob          -0.038153
anakandung     -0.052019
ayahjob        -0.054125
idkelas        -0.079428
usia           -0.081698
golda          -0.114074
sltatahunlulus -0.100462
status_pddikti  NaN
Name: status_lulus, dtype: float64
```

Gambar 4. 4 Korelasi fitur dengan target Status_lulus

Berdasarkan hasil tersebut, Langkah keempat dilakukan tahap penghapusan atribut yang dianggap tidak relevan. Fitur numerik seperti NIM dihapus karena hanya bersifat identitas unik. Demikian juga fitur status_pddikti dan lulus tahun dieliminasi karena secara langsung mewakili variabel target. Sementara itu, fitur golda, sltatahunlulus, usia, dan idkelas dihapus dari dataset karena memiliki korelasi yang sangat rendah terhadap variabel target status_lulus. Tahapan ini memastikan bahwa hanya fitur yang relevan dan berkontribusi signifikan yang dipertahankan untuk proses pemodelan. Code dapat dilihat sebagai berikut.

```
df = df.drop(columns=['nim', 'status_pddikti', 'lulus
tahun', 'golda', 'sltatahunlulus', 'Usia', 'idkelas'],
errors='ignore')
print(df.head(3))
print(df.shape[0])
```

Langkah akhir *preprocessing* adalah menghapus baris yang mengandung nilai kosong dengan perintah `dropna()`. Proses ini menghasilkan dataset bersih berjumlah 875 entri yang siap digunakan untuk tahap pemodelan. Jumlah ini berkurang dari 1.418 data awal karena adanya penghapusan duplikasi, pembersihan, serta seleksi fitur. Code dan hasil output dapat dilihat pada Gambar 4.5 berikut.

```
df = df.dropna()
print(df.head(3))
print(df.shape[0])
```

```

44      nim  usia  provinsi_id  golida  agama  kelamin  periode  \
0  19100001.0  21  0.803395  2.0  0.852900  1.0  0.993635
1  19100002.0  18  0.072843  2.0  0.852900  0.0  0.993635
2  19100003.0  18  0.803395  3.0  0.092645  0.0  0.993635

      gelombang  kofak  idkelas  stpid  s1ta_jur  sltatahunlulus  \
0  0.0  0.183876  1.0  0.966761  0.632712  2.0
1  0.0  0.183876  1.0  0.966761  0.632712  1.0
2  0.0  0.183876  1.0  0.966761  0.367288  1.0

      rata_nil_skh  pointerima  ayahjob  ibujob  ayahgaji  ibugaji  \
0  7.708333  5.892857  0.324247  0.186502  1.0  1.0
1  8.750000  7.942857  0.324247  0.486212  1.0  0.0
2  8.854167  5.821420  0.324247  0.066763  2.0  2.0

      anakandung  status_lulus
0  1.666667  1.0
1  1.666667  1.0
2  2.500000  1.0
875

```

Gambar 4. 5 Dataset final

Setelah proses preprocessing selesai, langkah berikutnya adalah menetapkan variabel target (dependen) serta variabel fitur (independen) yang akan digunakan dalam pemodelan. Variabel target pada penelitian ini adalah status kelulusan mahasiswa (*status_lulus*), yaitu label yang merepresentasikan apakah seorang mahasiswa lulus tepat waktu, terlambat, atau dikeluarkan.

Adapun variabel fitur yang berfungsi sebagai prediktor meliputi atribut demografis, akademik, dan administratif, yaitu: *provinsi_id*, *agama*, *jenis_kelamin*, *periode*, *gelombang*, *kofak*, *stpid*, *s1ta_jur*, *rata_nil_skh*, *pointerima*, *ayahjob*, *ibujob*, *ayahgaji*, *ibugaji*, dan *anakandung*. Keseluruhan fitur tersebut dipilih karena berpotensi memberikan informasi yang relevan dalam memprediksi status kelulusan mahasiswa. Dalam implementasi pada Python, pemisahan variabel dilakukan dengan cara mendefinisikan X sebagai kumpulan variabel *independent* dan variabel y yang didefinisikan sebagai *variable dependent*. Code dan hasil output dapat dilihat pada Tabel 4.3 dan Tabel 4.4 berikut.

```

X = df.drop(columns=[TARGET])
Y = df[TARGET]
print(X)
print(Y)

```

Tabel 4. 3 Variabel *independent*

No	provin sl_id	...	rata_nll _skhu	pointte rma	ayah Job	ibujo b	ayah gaji	ibu gaji	anakka ndung
0	0.8033 95	...	7.70833 3	5.8928 57	0.324 247	0.186 502	1.0	1.0	1.66666 7
1	0.0728 43	...	8.75000 0	7.9428 57	0.324 247	0.486 212	1.0	1.0	1.66666 7
2	0.8033 95	...	8.85416 7	5.8214 29	0.324 247	0.066 763	2.0	2.0	2.50000 0
3	0.0219 24	...	8.02083 3	5.2142 86	0.324 247	0.486 212	2.0	2.0	1.66666 7
4	0.8033 95	..	8.33333 3	5.3571 43	0.324 247	0.186 502	1.0	1.0	1.66666 7
...
8 7 5	0.0728 43	...	8.33333 3	5.3928 57	0.083 214	0.066 763	2.0	2.0	1.66666 7

Tabel 4. 4 Variabel *dependent*

No	status_lulus
0	1.0
1	1.0
2	1.0
3	1.0
4	1.0
...	...

874	0.0
875	2.0

Pada Tabel 4.3 di atas dapat dilihat berupa variabel independen (X) sedangkan pada Tabel 4.4 merupakan variabel dependen (y) yaitu, status kelulusan dengan jumlah data 875 data dan bertipe integer yang siap digunakan dalam proses pelatihan model klasifikasi.

4.4. *Split Data*

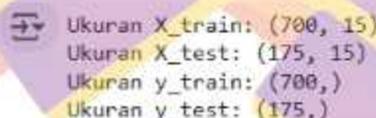
Setelah tahap *preprocessing* selesai, langkah berikutnya adalah melakukan pemisahan dataset menjadi data latih (*training set*) dan data uji (*testing set*). Pemisahan ini bertujuan untuk mengukur kinerja model secara objektif, dengan cara menguji model pada data yang sama sekali belum pernah digunakan dalam proses pelatihan. Dengan demikian, performa yang diperoleh dari data uji dapat mencerminkan kemampuan model dalam melakukan generalisasi pada data baru.

Proses pemisahan dilakukan secara acak untuk menghindari bias akibat pola tertentu pada data. Data latih digunakan untuk membangun dan menyesuaikan parameter model klasifikasi, sehingga model dapat mempelajari karakteristik mahasiswa berdasarkan kelas yang ada. Sebaliknya, data uji berfungsi untuk mengevaluasi hasil pelatihan tersebut. Model yang telah dilatih diterapkan pada data uji, kemudian hasil prediksi dibandingkan dengan label sebenarnya untuk mengetahui tingkat akurasi maupun metrik evaluasi lainnya.

Setelah pemisahan, dilakukan juga verifikasi dengan menampilkan dimensi masing-masing subset, yaitu X_{train} , X_{test} , y_{train} , dan y_{test} , untuk memastikan bahwa jumlah baris dan kolom sesuai dengan proporsi yang telah ditentukan. Proses

pembagian dataset dilakukan dengan potongan code berikut, sedangkan hasil output dari eksekusi ditampilkan pada Gambar 4.6.

```
X_train, X_test, y_train, y_test =  
train_test_split(X, y, test_size=0.2,  
random_state=42, stratify=y)  
  
print(f"Ukuran X_train: {X_train.shape}")  
print(f"Ukuran X_test: {X_test.shape}")  
print(f"Ukuran y_train: {y_train.shape}")  
print(f"Ukuran y_test: {y_test.shape}")
```



Gambar 4. 6 Split data latih dan data uji

Dalam implementasi pada Python, Gambar 4.6, pemisahan dataset dilakukan menggunakan fungsi *train_test_split* dari pustaka *scikit-learn* dengan parameter *test_size=0.2*, sehingga 20% dari total data dialokasikan sebagai data uji, sedangkan sisanya sebesar 80% digunakan untuk melatih model. Penetapan nilai *random_state=42* bertujuan menjaga reproduisibilitas hasil, artinya pembagian data akan selalu konsisten ketika kode dijalankan ulang. Selain itu, argumen *stratify=y* diterapkan untuk menjaga agar distribusi kelas target pada data latih dan data uji tetap proporsional dengan dataset utuh. Hal ini penting karena data penelitian memiliki ketidakseimbangan kelas, dengan dominasi mahasiswa yang lulus tepat waktu dibandingkan kategori lainnya.

Proporsi distribusi target variabel *status_lulus* pada dataset utuh, 875 baris data, menunjukkan bahwa mayoritas mahasiswa lulus tepat waktu berjumlah 607 mahasiswa (69,37%), diikuti oleh mahasiswa yang lulus terlambat 195 mahasiswa

(22,29%), dan sisanya dikeluarkan 73 mahasiswa (8,34%). Tabel 4 menunjukkan bahwa proporsi tetap konsisten pada data pelatihan dan pengujian, sehingga memastikan tidak terjadi bias distribusi akibat pembagian dataset. Adapun komposisi jumlah data *training* dan data *testing* disajikan pada Tabel 4.5 berikut.

Tabel 4. 5 Komposisi pembagian data *training* dan data *testing*

No	Jenis Data	Jumlah	Distribusi kelas target		
			(1)	(2)	(0)
1	Data <i>Training</i> (80%)	700 Data	69,43%	22,29%	8,34%
2	Data <i>Testing</i> (20%)	175 Data	69,14%	22,29%	8,57%

Tabel 4.5 menunjukkan bahwa proporsi kelas pada subset latih maupun uji relatif konsisten dengan proporsi pada dataset asli, sehingga validitas representasi data tetap terjaga.

4.4.1 Distribusi Kelas Sebelum Penyeimbangan Data

Setelah proses pemisahan dataset menjadi data latih dan data uji dilakukan, tahap berikutnya adalah menganalisis distribusi kelas pada data latih. Analisis ini penting untuk mengidentifikasi potensi ketidakseimbangan kelas yang dapat memengaruhi kinerja model klasifikasi, khususnya pada skema multi-kelas. Berdasarkan hasil pembagian data, data latih berjumlah 700 data, dengan distribusi kelas target status_lulus yang tidak seimbang. Kelas lulus tepat waktu (kelas 1) mendominasi data dengan 388 data, diikuti oleh kelas lulus terlambat (kelas 2) sebanyak 125 data, sedangkan kelas tidak lulus (kelas 0) hanya berjumlah 47 data. Kondisi ini menunjukkan adanya ketimpangan kelas yang cukup signifikan, di mana kelas minoritas memiliki proporsi yang jauh lebih kecil dibandingkan kelas mayoritas. Ketidakseimbangan tersebut berpotensi menyebabkan model klasifikasi

menjadi bias terhadap kelas mayoritas, sehingga meskipun akurasi tampak tinggi, kemampuan model dalam mengenali pola pada kelas minoritas dapat menurun. Oleh karena itu, diperlukan strategi penyeimbangan data sebelum proses pelatihan model dilakukan.

Tabel 4. 6 Distribusi Kelas Data Latih Sebelum Penyeimbangan

Kelas	Keterangan	Jumlah Data	Persentase
0	Tidak lulus	47	6,71%
1	Lulus tepat waktu	388	55,43%
2	Lulus terlambat	125	17,86%
	Total	700	100%

4.4.2 Penanganan Ketidakseimbangan Data dengan SMOTE

Untuk mengatasi permasalahan ketidakseimbangan kelas pada data latih, penelitian ini menerapkan Synthetic Minority Over-sampling Technique (SMOTE). SMOTE merupakan metode oversampling yang menghasilkan data sintesis pada kelas minoritas melalui proses interpolasi antar data, sehingga distribusi kelas menjadi lebih seimbang tanpa menghilangkan data asli.

Implementasi SMOTE dilakukan menggunakan pustaka *imbalanced-learn* dengan parameter `random_state` untuk menjaga konsistensi dan reproduisibilitas hasil. Parameter lain menggunakan nilai bawaan (default), yaitu jumlah tetangga terdekat (*k-neighbors*) sebesar 5 serta *sampling strategy* otomatis, di mana SMOTE akan menyesuaikan jumlah sampel sintesis hingga seluruh kelas memiliki jumlah data yang setara. Secara mekanisme, SMOTE bekerja dengan memilih satu data dari kelas minoritas, kemudian mencari sejumlah tetangga terdekat yang berasal dari kelas yang sama. Sampel sintesis baru dibentuk dengan cara melakukan

interpolasi antara data asli dan salah satu tetangganya, sehingga data yang dihasilkan tetap berada pada ruang fitur yang representatif. Pendekatan ini memungkinkan model memperoleh variasi data minoritas yang lebih kaya dibandingkan metode duplikasi sederhana. Pada penelitian ini, SMOTE diterapkan hanya pada data latih, baik pada tahap pembentukan meta-feature menggunakan skema *Stratified K-Fold Cross Validation* maupun pada tahap pelatihan penuh model dasar. Penerapan SMOTE secara terbatas pada data latih bertujuan untuk mencegah terjadinya data leakage serta menjaga validitas evaluasi model.

Hasil penerapan SMOTE menunjukkan bahwa distribusi kelas pada data latih menjadi lebih seimbang. Pada proses pembentukan meta-feature dengan validasi silang 5-fold, SMOTE diterapkan pada data fold-train di setiap iterasi. Hasilnya, pada sebagian besar fold, jumlah data pada setiap kelas menjadi 389 data, sedangkan pada satu fold lainnya masing-masing kelas berjumlah 388 data.

Selanjutnya, pada tahap pelatihan penuh model dasar, SMOTE kembali diterapkan pada seluruh data latih. Proses ini menghasilkan distribusi kelas akhir yang sepenuhnya seimbang, yaitu masing-masing kelas memiliki 486 data, dengan total keseluruhan data latih setelah SMOTE sebesar 1.458 data. Distribusi yang seimbang ini memastikan bahwa setiap kelas memiliki kontribusi yang setara dalam proses pembelajaran model.

Tabel 4. 7 Distribusi Kelas Data Latih Setelah SMOTE

Tahap Penerapan SMOTE	Kelas 0	Kelas 1	Kelas 2	Total Data
Sebelum SMOTE	47	388	125	700
SMOTE pada fold-train (rata-rata)	389	389	389	1.167
SMOTE penuh (base model)	486	486	486	1.458

Metode SMOTE dipilih dibandingkan teknik penyeimbangan data lainnya seperti random undersampling, ADASYN, maupun Borderline-SMOTE dengan beberapa pertimbangan. Random undersampling berisiko menghilangkan informasi penting dari kelas mayoritas karena mengurangi jumlah data secara signifikan. Sementara itu, ADASYN dan Borderline-SMOTE cenderung menghasilkan sampel sintetis secara lebih agresif di sekitar batas kelas, yang pada dataset multikelas dengan karakteristik fitur yang beragam berpotensi meningkatkan noise.

SMOTE dipilih karena mampu menghasilkan sampel sintetis yang stabil dan merata pada ruang fitur, serta telah banyak digunakan dalam penelitian prediksi performa akademik mahasiswa. Pendekatan ini dinilai paling sesuai untuk mendukung proses klasifikasi multikelas dalam penelitian ini, sekaligus menjaga keseimbangan antara peningkatan sensitivitas kelas minoritas dan stabilitas model secara keseluruhan.

4.5. Modeling Machine Learning

Data latih yang telah melalui proses penyeimbangan menggunakan SMOTE selanjutnya digunakan pada tahap pemodelan machine learning, baik pada pelatihan model dasar maupun pembentukan meta-feature dalam pendekatan *stacking ensemble*. Tujuan utama pada penelitian ini adalah memprediksi status kelulusan calon mahasiswa dalam skema multi-kelas (lebih dari dua kelas). Penentuan model, penyetelan parameter, serta strategi evaluasi akan dijelaskan berikut ini.

4.5.1 Pendekatan Klasifikasi Multi-Kelas

Klasifikasi multi-kelas adalah situasi di mana setiap sampel (calon mahasiswa) hanya memiliki tepat satu kelas target dari sejumlah kelas yang sudah

ditentukan. Dalam penelitian ini, untuk memprediksi status kelulusan (kelas 0 = “tidak lulus”, 1 = “lulus tepat waktu”, 2 = “lulus terlambat”) adalah target yang eksklusif seorang mahasiswa tidak bisa termasuk dua status sekaligus.

Dalam konteks ini, model yang dibangun harus mampu membedakan pola antar kelas sekaligus mengatasi tantangan ketidakseimbangan distribusi data, terutama pada kategori minoritas (dropout dan lulus terlambat). Algoritma yang dipakai harus mampu membedakan di antara beberapa kelas sekaligus dan memilih satu terbaik. Di penelitian ini, setiap model akan dilatih sebagai klasifikasi multi-kelas menggunakan fitur (X) dan label (y) yang sudah ditentukan.

4.5.2 *Hyperparameter Tuning*

Agar setiap algoritma dapat bekerja secara optimal, dilakukan penyesuaian hyperparameter (hyperparameter tuning). *Hyperparameter* berbeda dengan parameter model yang dipelajari langsung dari data, parameter harus ditentukan terlebih dahulu dan berpengaruh langsung terhadap performa algoritma. Pemilihan nilai *hyperparameter* yang tepat dapat meningkatkan akurasi, *recall*, maupun *F1-score*, terutama dalam klasifikasi dengan lebih dari dua kelas.

Pada penelitian ini, proses tuning dilakukan menggunakan GridSearchCV, sebuah metode yang secara sistematis mencoba berbagai kombinasi *hyperparameter*, kemudian mengevaluasi performa masing-masing kombinasi dengan teknik validasi silang. Pendekatan ini dipilih karena mampu menjelajahi ruang parameter dengan terstruktur, sekaligus memberikan hasil yang reproducible.

Berikut adalah parameter yang digunakan untuk klasifikasi multi-kelas berdasarkan algoritma yang diterapkan. Parameter dapat dilihat pada Tabel 4.6 berikut ini.

Tabel 4. 8 Parameter yang digunakan dalam *hyperparameter tuning*

No	Model	Nama Parameter
1	K-Nearest Neighbors (KNN)	n_neighbors : [3, 5, 7, 9] Weights : ['uniform', 'distance'] Metric : ['euclidean', 'manhattan']
2	Random Forest (RF)	n_estimators : [100, 200, 300] max_depth : [None, 10, 20] min_samples_split : [2, 5, 10] min_samples_leaf : [1, 2, 4]
3	Support Vector Machine (SVM)	C : {0.1, 1, 10, 100} Kernel : ['linear', 'rbf', 'poly'] Gamma : ['scale', 'auto', 0.1, 1]

4.5.3 Pemodelan Level 0 (Base Model)

Tahap pertama pemodelan dilakukan menggunakan tiga algoritma dasar, yaitu *Random Forest* (RF), *K-Nearest Neighbors* (KNN), dan *Support Vector Machine* (SVM). Ketiganya dilatih secara terpisah dengan data latih menggunakan skema validasi silang *5-fold* untuk menjaga konsistensi evaluasi pada dataset yang memiliki distribusi kelas tidak seimbang. Contoh implementasi *k-fold* dapat dilihat pada kode berikut:

```
from sklearn.model_selection import KFold
kf = KFold(n_splits=5, shuffle=True,
random_state=42)
```

Hasil tuning *hyperparameter* menghasilkan konfigurasi terbaik untuk masing-masing algoritma seperti ditunjukkan pada Tabel 4.7 berikut.

Tabel 4. 9 *Hyperparameter* terbaik untuk model dasar

Algoritma	Hyperparameter
Random Forest	n_estimators=100, max_depth=None, min_samples_leaf=4, min_samples_split=2, bootstrap=True, random_state=RANDOM_SEED
KNN	n_neighbors=9, weights=uniform, algorithm=auto, leaf_size=30, p=1, metric=manhattan
SVM	C=10, kernel=rbf, gamma=scale, probability=True, random_state=RANDOM_SEED

Hasil evaluasi model dasar, Tabel 4.8, pada data latih yang telah melalui proses penyeimbangan menggunakan SMOTE menunjukkan bahwa ketiga algoritma Random Forest (RF), KNN, dan SVM mampu mempelajari pola dari fitur input dengan karakteristik yang berbeda-beda. Pada kondisi data yang telah lebih seimbang, performa keseluruhan model cenderung meningkat, khususnya pada kelas-kelas yang semula memiliki representasi lebih kecil. Dari ketiga algoritma tersebut, Random Forest kembali menampilkan performa paling stabil dengan akurasi 0,74 dan weighted F1-score 0,73, menjadikannya model dasar yang paling konsisten sebelum dilakukan proses *stacking*. Pada kelas 0, yang mewakili mahasiswa dikeluarkan, RF memberikan nilai *precision* sebesar 0,55 dan *recall* 0,59, lebih tinggi dibandingkan KNN (*precision* 0,42; *recall* 0,60) maupun SVM (*precision* 0,45; *recall* 0,69). Meskipun nilai *recall* pada kelas ini belum optimal, hasil ini menunjukkan bahwa penyeimbangan data melalui SMOTE memberikan

dampak positif dengan meningkatkan kemampuan model dalam mengenali kategori minoritas, dibandingkan kondisi sebelum dilakukan *oversampling*. Sebagai contoh model SVM yang sebelumnya sama sekali tidak dapat melakukan prediksi di kelas 2 sebelum dilakukan SMOTE. Kemudian untuk kelas 1 (lulus tepat waktu), ketiga model mencatat performa yang baik, terutama RF yang memperoleh *precision* 0,83 dan *recall* 0,89. Hasil ini menunjukkan bahwa pada kelas mayoritas, RF mampu mengenali pola secara konsisten. Sementara itu KNN dan SVM memiliki *precision* yang sebanding (0,82-0,87), meskipun *recall* keduanya lebih rendah dibandingkan RF. Pada kelas 2 (lulus terlambat), tantangan model tampak lebih jelas. Meskipun data telah diseimbangkan, performa ketiga model pada kelas ini tetap relatif rendah. RF hanya mencatat *precision* 0,48 dan *recall* 0,35, sedangkan KNN dan SVM menunjukkan pola yang serupa dengan F1-score berada pada kisaran 0,35-0,40. Hal ini menunjukkan bahwa meskipun SMOTE membantu memperbaiki distribusi data, karakteristik kelas 2 masih sulit dipisahkan oleh model dasar, kemungkinan karena pola perilakunya tumpang tindih dengan kelas lainnya. Hasil klasifikasi model dasar terhadap data latih dirangkum pada Tabel 4.8 berikut.

Tabel 4. 10 Laporan klasifikasi model dasar

Kelas	Metrik	RF	KNN	SVM
Class 0	Precision	0.55	0.42	0.45
	Recall	0.59	0.60	0.69
	F1-Score	0.57	0.50	0.54
	Support	58	58	58
Class 1	Precision	0.83	0.82	0.87
	Recall	0.89	0.65	0.63
	F1-Score	0.86	0.72	0.73
	Support	486	486	486
Class 2	Precision	0.48	0.29	0.32
	Recall	0.35	0.43	0.54
	F1-Score	0.41	0.35	0.40

	Support	156	156	156
Akurasi		0.74	0.60	0.61
Avg	Weighted Precision	0.73	0.67	0.71
	Weighted Recall	0.74	0.60	0.61
	Weighted F1-Score	0.73	0.62	0.64

Secara keseluruhan, hasil evaluasi ini memberikan dua informasi penting sebelum melanjutkan ke tahap *stacking*. Pertama, SMOTE berhasil meningkatkan representasi kelas minoritas, terlihat dari naiknya *recall* pada kelas 0 dan kelas 2 dibandingkan kondisi sebelum *oversampling*. Namun peningkatan ini belum cukup untuk menjadikan model dasar benar-benar kuat pada semua kelas. Kedua, Random Forest tetap unggul sebagai model dengan performa paling stabil, sehingga prediksinya berpotensi memberikan kontribusi *meta-feature* yang lebih baik ketika dikombinasikan dalam model tingkat meta menggunakan XGBoost.

4.5.4 Pembentukan *Meta-Feature*

Keterbatasan yang ditunjukkan model tunggal pada kategori minoritas menjadi alasan perlunya pendekatan *ensemble stacking*. Pada tahap ini, hasil prediksi dari model dasar digunakan untuk membentuk *meta-feature*, yang kemudian digabungkan dengan fitur awal dataset.

Dua skema *stacking* diterapkan, yaitu kombinasi hasil prediksi dua model dasar yang menghasilkan 17 fitur pada dataset gabungan dan kombinasi tiga model dasar dengan menghasilkan total 18 fitur. Dengan strategi ini, diharapkan *meta-model* dapat memanfaatkan keunggulan masing-masing algoritma dasar dan memperbaiki kelemahan yang muncul ketika model bekerja secara individual.

Berikut ini adalah code untuk menyimpan catatan results perkombinasi dari hasil prediksi dua model dan tiga model dasar.

```
# catatan results_per_combo / KOMBINASI 3 DAN 2
MODEL UNTUK PREDIKSI (TRAIN & TEST)
# Loop kombinasi meta-model
combos = {
    "3models": [0, 1, 2], # kombinasi 3 model
    "rf_knn": [0, 1], # kombinasi 2 model
    "rf_svm": [0, 2], # kombinasi 2 model
    "knn_svm": [1, 2] # kombinasi 2 model
}

results_per_combo = []

for combo_name, cols in combos.items():
    print(f"\n=== Training meta-model for combo:
    {combo_name} ===")

    #TRAIN: ambil meta kolom sesuai kombinasi
    meta_X_train_combo = meta_X_oof[:, cols]
    X_train_combined =
    build_combined_from_meta(meta_X_train_combo,
    X_train)
    y_train_comb = y_train.reset_index(drop=True)
    # (SMOTE, add_noise, training final_meta, dll)

    # TEST: bangun meta-features test untuk kombinasi

    meta_test_final = np.zeros((X_test.shape[0],
    len(cols)), dtype=int)
    for j, col_idx in enumerate(cols):
        model_for_meta = loaded_bases[col_idx]
        meta_test_final[:, j] =
        model_for_meta.predict(X_test)

    X_test_final_comb =
    build_combined_from_meta(meta_test_final, X_test)
    y_test_pred =
    final_meta.predict(X_test_final_comb)
```

Untuk menampilkan hasil kombinasi model yang sudah disimpan pada catatan *results* perkombinasi dari hasil prediksi sebelumnya dengan tambahan fitur awal sebagai *meta-features* penelitian ini disajikan pada detail code dan Tabel 4.9 sebagai *output*.

```
# DETAIL 2 UNTUK LAPORAN / MELOAD SEMUA META
FITUR
print("  SEMUA FITUR + 5 BARIS NILAI PERTAMA")

for combo, model in loaded_meta_models.items():
    print(f"\n=== META-MODEL [{combo}] ===")

    # Cek apakah meta-model punya atribut nama fitur
    if not hasattr(model, "feature_names_in_"):
        print("Meta-model ini tidak memiliki
atribut 'feature_names_in_'")
        continue

    feature_names = list(model.feature_names_in_)
    n_features = len(feature_names)

    # Tampilkan semua nama fitur
    print(f"\nJumlah fitur: {n_features}")
    print("Daftar fitur yang digunakan meta-
model:")
    for i, fname in enumerate(feature_names,
start=1):
        print(f"{i:3d}. {fname}")

    # Menampilkan nilai fitur (5 baris pertama)
    print("\nNilai fitur (5 baris pertama dari
X_test_combined):")

    # Ambil kombinasi base model untuk meta-model ini
    if combo not in combos:
        print(f"Combo '{combo}' tidak ada di
'combos', skip nilai-fitur.")
        continue

    cols = combos[combo]
```

```

# Bangun meta-features untuk test dari base
models
    meta_test = np.zeros((X_test.shape[0],
len(cols)), dtype=int)
    for j, col_idx in enumerate(cols):
        base_model = loaded_bases[col_idx]
        meta_test[:, j] =
base_model.predict(X_test)

# Gabungkan fitur asli + meta_* (harus sama
seperti saat training meta-model)
X_test_combined =
build_combined_from_meta(meta_test, X_test)

# Ambil semua kolom sesuai urutan feature_names
try:
    preview_df =
X_test_combined[feature_names].head()
    print(preview_df.to_string(index=False))
except KeyError as e:
    print("Terjadi KeyError saat mengambil
kolom sesuai feature_names_in_")
    print("Detail error:", e)
    print("Cek apakah nama kolom
X_test_combined sama dengan feature_names_in_")

```

Tabel 4. 11 Ringkas Hasil pembentukan *meta-feature* dari kombinasi model

Model	Nama File	Jumlah Fitur	Jumlah Baris	Isi satu Baris Pertama
RF + KNN + SVM + Fitur Asli	meta_model_x gboost_3model s.pkl	18	700	[provinsi_id: 0.803395], [agama: 0.8529], [kelamin: 1], [periode: 0.993635], [gelombang: 1], [kofak: 0.183876], [stpid: 0.966761], [slta_jur: 0.632712], [rata_nil_skhu: 7.708333], [pointterima: 3.785714], [ayahjob: 0.070301], [ibujob: 0.186502], [ayahgaji: 0], [ibugaji: 1], [anakkandung: 2.25], [meta_0: 0], [meta_1: 0], [meta_2: 0]

RF + KNN + Fitur Asli	meta_model_x gboost_rf_knn. pkl	17	700	[provinsi_id: 0.803395], [agama: 0.8529], [kelamin: 1], [periode: 0.993635], [gelombang: 1], [kofak: 0.183876], [stpid: 0.966761], [slta_jur: 0.632712], [rata_nil_skh: 7.708333], [pointterima: 3.785714], [ayahjob: 0.070301], [ibujob: 0.186502], [ayahgaji: 0], [ibugaji: 1], [anakkandung: 2.25], [meta_0: 0], [meta_1: 0]
RF + SVM + Fitur Asli	meta_model_x gboost_rf_svm. pkl	17	700	[provinsi_id: 0.803395], [agama: 0.8529], [kelamin: 1], [periode: 0.993635], [gelombang: 1], [kofak: 0.183876], [stpid: 0.966761], [slta_jur: 0.632712], [rata_nil_skh: 7.708333], [pointterima: 3.785714], [ayahjob: 0.070301], [ibujob: 0.186502], [ayahgaji: 0], [ibugaji: 1], [anakkandung: 2.25], [meta_0: 0], [meta_1: 0]
KNN + SVM + Fitur Asli	meta_model_x gboost_knn_svm. pkl	17	700	[provinsi_id: 0.803395], [agama: 0.8529], [kelamin: 1], [periode: 0.993635], [gelombang: 1], [kofak: 0.183876], [stpid: 0.966761], [slta_jur: 0.632712], [rata_nil_skh: 7.708333], [pointterima: 3.785714], [ayahjob: 0.070301], [ibujob: 0.186502], [ayahgaji: 0], [ibugaji: 1], [anakkandung: 2.25], [meta_0: 0], [meta_1: 0]

Selanjutnya, untuk menjamin bahwa proses *stacking* berjalan dengan benar, dilakukan verifikasi keselarasan antara data latih asli (X_{train}) dan *meta-feature* yang dibentuk melalui prediksi base model. Pada pendekatan *stacking*, setiap baris *meta-feature* harus persis merepresentasikan baris yang sama pada data input awal. Ketidaktepatan urutan baris akan menyebabkan meta-model menerima informasi yang keliru sehingga memengaruhi proses pembelajaran maupun hasil prediksi. Oleh karena itu, dilakukan pengecekan khusus untuk memastikan bahwa *meta-feature* hasil OOF maupun *meta-feature* pada data uji tersusun dalam urutan yang sama dengan data aslinya. Pemeriksaan ini mencakup kecocokan jumlah baris, kelengkapan *meta-feature*, serta kesamaan isi fitur asli setelah digabungkan kembali dengan meta-fitur. *Script* dan Gambar 4.7 ditampilkan sebagai bukti bahwa seluruh kombinasi meta-model telah memenuhi kesesuaian tersebut.

```
def row_fingerprint(df):
    """Fingerprint baris: jika ada numerik -> sum; jika
    tidak -> join string."""
    df_num = df.select_dtypes(include=[np.number])
    return df_num.sum(axis=1) if df_num.shape[1] else
    df.astype(str).agg('|'.join, axis=1)

def _check_fp(df1, df2, combo_name, mode):
    fp1, fp2 =
    row_fingerprint(df1.reset_index(drop=True)),
    row_fingerprint(df2.reset_index(drop=True))
    if not fp1.equals(fp2):
        idx = np.where(fp1 != fp2)[0][:10]
        raise ValueError(f"Fingerprint mismatch pada
        {mode} combo [{combo_name}] - contoh index: {idx}")

def check_train_alignment(X_train, y_train, meta_X_oof,
                           combos):
    print("\n[TRAIN] Cek alignment meta vs data asli")
    n_meta = meta_X_oof.shape[0]
```

```

if not (n_meta == len(X_train) == len(y_train)):
    raise ValueError("Jumlah baris meta_X_oof,
X_train, dan y_train tidak sama!")

# cek nilai -1
unfilled = np.where((meta_X_oof == -
1).any(axis=1))[0]
if len(unfilled):
    raise ValueError(f"Ada {len(unfilled)} baris
meta_X_oof belum terisi. Contoh index: {unfilled[:10]}")
print(" Semua baris meta_X_oof sudah terisi.")

# cek tiap kombinasi
for name, cols in combos.items():
    print(f"\n >> Cek TRAIN [{name}] base index
{cols}")
    X_comb = build_combined_from_meta(meta_X_oof[:,
cols], X_train)
    if len(X_comb) != len(X_train):
        raise ValueError(f"MISMATCH jumlah baris pada
combo [{name}]")
    check_fp(X_train, X_comb[X_train.columns], name, "TRAIN")
    print(f" Alignment TRAIN OK untuk combo [{name}].")
    def check_test_alignment(X_test, y_test, combos,
loaded_bases):
        print("\n[TEST] Cek alignment meta vs data asli")
        if len(X_test) != len(y_test):
            raise ValueError("Jumlah baris X_test dan y_test
tidak sama!")
        for name, cols in combos.items():
            print(f"\n >> Cek TEST [{name}] base index
{cols}")
    # bangun meta test
    meta_test =
np.column_stack([loaded_bases[c].predict(X_test) for c in
cols])
    X_comb = build_combined_from_meta(meta_test,
X_test)
    if len(X_comb) != len(X_test):
        raise ValueError(f"MISMATCH jumlah baris pada
combo [{name}] (TEST)")
    _check_fp(X_test, X_comb[X_test.columns], name,
"TEST")print(f" Alignment TEST OK untuk combo [{name}].")

```

```

[TRAIN] Cek alignment meta-fitur vs data asli ...
meta_X_oof.shape = (700, 3)
X_train.shape[0] = 700
y_train.shape[0] = 700
Semua baris meta_X_oof sudah terisi.

>> Cek kombinasi TRAIN [3models] dengan base index [0, 1, 2]
Alignment TRAIN OK untuk combo [3models] (fitur asli sejajar baris demi baris).

>> Cek kombinasi TRAIN [rf_knn] dengan base index [0, 1]
Alignment TRAIN OK untuk combo [rf_knn] (fitur asli sejajar baris demi baris).

>> Cek kombinasi TRAIN [rf_svm] dengan base index [0, 2]
Alignment TRAIN OK untuk combo [rf_svm] (fitur asli sejajar baris demi baris).

>> Cek kombinasi TRAIN [knn_svm] dengan base index [1, 2]
Alignment TRAIN OK untuk combo [knn_svm] (fitur asli sejajar baris demi baris).

[TEST] Cek alignment meta-fitur vs data asli ...
X_test.shape[0] = 175
y_test.shape[0] = 175

>> Cek kombinasi TEST [3models] dengan base index [0, 1, 2]
Alignment TEST OK untuk combo [3models] (fitur asli sejajar baris demi baris).

>> Cek kombinasi TEST [rf_knn] dengan base index [0, 1]
Alignment TEST OK untuk combo [rf_knn] (fitur asli sejajar baris demi baris).

>> Cek kombinasi TEST [rf_svm] dengan base index [0, 2]
Alignment TEST OK untuk combo [rf_svm] (fitur asli sejajar baris demi baris).

>> Cek kombinasi TEST [knn_svm] dengan base index [1, 2]
Alignment TEST OK untuk combo [knn_svm] (fitur asli sejajar baris demi baris).

PENCEKIAN ALIGNMENT SELESAI TANPA EROR.

```

Gambar 4. 7 Pengecekan kesesuaian indeks penggabungan *meta-features*

Berdasarkan hasil pengecekan *alignment* pada *script* diatas, *seluruh meta-feature* yang dibentuk dari kombinasi model (3models, rf_knn, rf_svm, dan knn_svm) telah terbukti memiliki kesesuaian penuh dengan data asli. Validasi dilakukan dengan memeriksa jumlah baris, kelengkapan *meta-feature* hasil OOF, serta memastikan tidak ada baris yang tertinggal atau berubah urutannya. Pemeriksaan ini juga diterapkan pada data uji dengan membangun kembali *meta-feature* dari prediksi *base* model dan membandingkannya dengan panjang X_test dan y_test. Seluruh kombinasi menunjukkan bahwa *meta-feature* tersusun dalam urutan yang sama dengan data input awal, tanpa baris kosong maupun ketidaksejajaran struktur data. Dengan demikian, *meta-feature* yang dihasilkan

dipastikan aman dan konsisten untuk digunakan dalam tahap stacking, tanpa risiko misalignment antara data asli dan hasil prediksi dari *base* model.

Pada penelitian ini, pendekatan *stacking* diterapkan dalam dua tingkatan, yaitu level 0 sebagai model dasar dan level 1 sebagai meta-model, tanpa menerapkan *stacking* bertingkat lebih lanjut. Model dasar pada level 0 terdiri dari Random Forest, K-Nearest Neighbors, dan Support Vector Machine, yang masing-masing memiliki karakteristik pembelajaran yang berbeda. Random Forest memanfaatkan mekanisme bagging berbasis pohon keputusan, KNN bekerja berdasarkan kedekatan antar instance, sedangkan SVM membangun pemisahan kelas berbasis margin non-linier. Perbedaan karakteristik ini memungkinkan setiap model menangkap pola data dari sudut pandang yang berbeda sehingga prediksinya berpotensi saling melengkapi.

Tujuan utama *stacking* pada level0 adalah menghasilkan *meta-feature* berupa keluaran prediksi dari model dasar, yang kemudian digunakan sebagai masukan tambahan pada level 1. Pada level 1, XGBoost digunakan sebagai meta-model untuk mempelajari hubungan antara *meta-feature* dan fitur asli dalam menentukan prediksi akhir. Pemilihan XGBoost didasarkan pada kemampuannya dalam memodelkan interaksi non-linier, menangani data berdimensi tinggi, serta menyediakan mekanisme regularisasi yang efektif.

Berdasarkan hasil evaluasi kuantitatif, penggunaan stacking dengan XGBoost sebagai meta-model menghasilkan performa yang lebih stabil dibandingkan penggunaan XGBoost tanpa *meta-feature*, terutama pada metrik akurasi dan weighted F1-score pada data uji. Meskipun peningkatan performa yang

diperoleh bersifat moderat, konsistensi nilai metrik dan menurunnya kesenjangan performa antara data latih dan data uji menunjukkan bahwa meta-feature dari model dasar memberikan informasi tambahan yang relevan. Dengan demikian, peningkatan kinerja model tidak semata-mata disebabkan oleh kompleksitas XGBoost, tetapi juga oleh kontribusi prediksi model dasar yang bersifat komplementer.

Namun demikian, penelitian ini juga menyadari adanya potensi redundansi informasi, khususnya antara Random Forest dan XGBoost yang sama-sama berbasis pohon keputusan. Oleh karena itu, *stacking* dalam penelitian ini tidak dimaksudkan untuk selalu mengungguli model tunggal secara signifikan, melainkan untuk mengevaluasi sejauh mana penggabungan model dasar dengan karakteristik berbeda dapat memberikan nilai tambah yang terukur dan stabil dalam konteks prediksi performa akademik mahasiswa.

4.5.5 Pemodelan Level 1 (Meta Model)

Tahapan selanjutnya setelah pemodelan model dasar adalah membangun model Level 1 (meta-model) yang berfungsi untuk menggabungkan keunggulan dari beberapa algoritma dasar melalui pendekatan *stacking ensemble*. Pada tahap ini, setiap base model menghasilkan prediksi yang dikonversi menjadi *meta-feature*, lalu digabungkan kembali dengan fitur asli untuk membentuk representasi baru yang lebih kaya. Pendekatan ini digunakan untuk mengatasi keterbatasan model tunggal yang cenderung memiliki performa berbeda-beda pada tiap kelas. Dengan skema *stacking*, hasil prediksi dari model dasar dijadikan sebagai *meta-features* yang selanjutnya digunakan sebagai input bagi model tingkat meta.

Dalam penelitian ini, algoritma *XGBoost* digunakan sebagai *meta-learner* karena memiliki kemampuan yang baik dalam menangani data *non-linear*, melakukan regularization untuk mencegah *overfitting*, serta efisien dalam proses pembelajaran bertingkat (Fernandez-Garcia et al., 2021; Ghasemieh et al., 2023; Herianto et al., 2024; Sahlaoui et al., 2021). Model *XGBoost* dilatih menggunakan *meta-features* hasil kombinasi dari dua dan tiga model dasar, yang masing-masing menghasilkan empat skenario *meta-model*, yakni: Kombinasi RF-KNN, Kombinasi RF-SVM, Kombinasi KNN-SVM, dan Kombinasi RF-KNN-SVM. Proses pelatihan dilakukan secara berulang (*looping*) agar setiap kombinasi model memperoleh hasil pelatihan dan evaluasi yang terpisah namun dengan prosedur yang seragam.

Sebelum model akhir ditetapkan, dilakukan proses *hyperparameter tuning* guna mencari kombinasi parameter terbaik yang mampu menghasilkan performa optimal. Parameter grid yang digunakan di tampilkan pada Tabel 4.10.

Tabel 4. 12 Parameter grid untuk *hyperparameter tuning* model meta

No.	Parameter	Nilai yang Diujl
1	n estimators	100, 200, 300, 500
2	learning rate	0.01, 0.03, 0.05, 0.1
3	max depth	3, 5, 7
4	subsample	0.7, 0.8, 0.9
5	colsample bytree	0.6, 0.7, 0.8, 0.9
6	gamma	0, 0.1, 0.5, 1
7	reg alpha	0, 0.1, 0.5
8	reg lambda	1, 0.1, 0.5

Setelah diperoleh kombinasi optimal, meta-model XGBoost akhir dilatih menggunakan parameter hasil *tuning*. Berikut adalah *code* parameter yang digunakan untuk klasifikasi multi kelas berdasarkan algoritma yang ditetapkan.

```
base_xgb_params = {
    'objective': 'multi:softprob',
    'num_class': len(np.unique(y_train)),
    'use_label_encoder': False,
    'eval_metric': 'mlogloss',
    'random_state': RANDOM_SEED,
    'max_depth': 3,
    'learning_rate': 0.03,
    'n_estimators': 300,
    'subsample': 0.8,
    'colsample_bytree': 0.6,
    'gamma': 0.5,
    'reg_alpha': 0,
    'reg_lambda': 0.1
}
```

Parameter-parameter tersebut dipilih karena memberikan keseimbangan yang baik antara bias dan varians. Penggunaan *learning_rate* rendah dan kedalaman pohon (*max_depth*) yang moderat menjaga stabilitas model, sedangkan *subsample* dan *colsample_bytree* mendorong variasi antar pohon untuk meningkatkan generalisasi.

Selanjutnya, pada tahapan implementasi meta-model dilakukan menggunakan skrip program berbasis Python sebagaimana ditunjukkan pada *code* berikut dan *output* pada Tabel 4.11.

```

# SMOTE full + noise
sm_meta_full = SMOTE(random_state=GLOBAL_SEED)
X_train_bal, y_train_bal = sm_meta_full.fit_resample(X_train_combined, y_train_comb)
X_train_bal_noisy = add_noise(X_train_bal)

# Train meta-model XGBoost deterministik
final_meta = xgb.XGBClassifier(**base_xgb_params)
final_meta.fit(X_train_bal_noisy, y_train_bal)

meta_fname = f"saved_models/meta_model_xgboost_{combo_name}.pkl"
joblib.dump(final_meta, meta_fname)
print("Saved meta model:", meta_fname)

```

Tabel 4. 13 Output 20 prediksi pertama meta-model versi Train Set

Kombinasi Model	Output 20 Pertama
<i>Actual</i> (y_train)	[0,0, 2,0, 1,0, 1,0, 1,0, 1,0, 2,0, 1,0, 2,0, 2,0, 1,0, 2,0, 0,0, 1,0, 2,0, 0,0, 1,0, 1,0, ...]
3models (y_pred)	[0, 2, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 2, 0, 1, 2, 0, 2, 2, 1, ...]
rf_knn (y_pred)	[0, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 0, 1, 2, 0, 2, 2, 1, ...]
rf_svm (y_pred)	[0, 2, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 0, 1, 2, 0, 2, 2, 1, ...]
knn_svm (y_pred)	[0, 2, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 0, 1, 2, 0, 2, 2, 1, ...]

Pada Tabel juga menampilkan data asli dan data hasil prediksi guna memberikan gambaran konkret terhadap keluaran model. Melalui langkah ini, peneliti dapat membandingkan antara nilai prediksi dan label sebenarnya, sekaligus memastikan bahwa hasil yang diperoleh sejalan dengan pola yang diharapkan.

4.5.6 Evaluasi Model

Evaluasi hasil pelatihan meta-model dilakukan dengan menggunakan metrik akurasi, *precision*, *recall*, *F1-score*, dan *confusion matrix* untuk setiap kombinasi model. Metrik-metrik tersebut digunakan untuk menilai sejauh mana

model mampu mengenali pola data dan mengklasifikasikan status kelulusan calon mahasiswa secara tepat. Analisis ini bertujuan membandingkan efektivitas masing-masing konfigurasi *stacking ensemble* dalam menghasilkan prediksi yang akurat dan seimbang antar kelas. Code dan *output* untuk analisa performa model disajikan pada Gambar 4.9 sampai dengan Gambar 4.12 berikut.

```
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, confusion_matrix, classification_report
)
```

```
print(classification_report(y_train, y_pred,
    zero_division=0))
```

```
Train classification report:
      precision    recall  f1-score   support

 0.0         0.87     0.82     0.85        58
 1.0         0.88     0.92     0.90       485
 2.0         0.72     0.65     0.68       155

 accuracy: 0.81
 macro avg: 0.75     0.75     0.75
 weighted avg: 0.83     0.85     0.85
```

Gambar 4. 8 Classification report RF-KNN-SVM

```
Train classification report:
      precision    recall  f1-score   support

 0.0         0.70     0.84     0.82        58
 1.0         0.80     0.92     0.86       485
 2.0         0.75     0.82     0.69       155

 accuracy: 0.81
 macro avg: 0.81     0.88     0.85
 weighted avg: 0.80     0.85     0.81
```

Gambar 4. 9 Classification report RF-KNN

```
Train classification report:
      precision    recall  f1-score   support

 0.0         0.89     0.81     0.87        58
 1.0         0.87     0.93     0.90       485
 2.0         0.72     0.58     0.65       155

 accuracy: 0.83
 macro avg: 0.83     0.78     0.82
 weighted avg: 0.84     0.84     0.84
```

Gambar 4. 10 Classification report RF-SVM

```
Train classification report:
      precision    recall  f1-score   support

 0.0         0.88     0.81     0.85        58
 1.0         0.88     0.93     0.90       485
 2.0         0.72     0.68     0.69       155

 accuracy: 0.81
 macro avg: 0.81     0.78     0.81
 weighted avg: 0.81     0.85     0.81
```

Gambar 4. 11 Classification report KNN-SVM

Evaluasi menunjukkan bahwa meta-model *XGBoost* memberikan peningkatan performa secara konsisten dan signifikan dibandingkan model tunggal. Pada Tabel 4.12 berikut disajikan ringkasan hasil evaluasi keempat meta-model yang dihasilkan.

Tabel 4. 14 Classification report pelatihan base model dan meta model

Metrics	Base Model			Meta Model (Stacking Combination)			
	KNN	RF	SVM	RF-KNN	RF-SVM	KNN-SVM	3 Model
Accuracy	0.60	0.74	0.61	0.85	0.84	0.85	0.85
Weighted Avg Precision	0.67	0.73	0.71	0.85	0.84	0.85	0.85
Weighted Avg Recall	0.60	0.74	0.61	0.85	0.84	0.85	0.85
Weighted Avg F1-Score	0.62	0.73	0.64	0.85	0.84	0.85	0.85

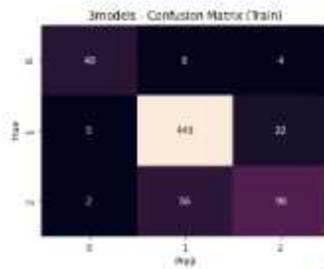
Berdasarkan hasil pada tabel di atas, terlihat bahwa seluruh kombinasi meta model menunjukkan akurasi yang konsisten tinggi, yaitu berada pada kisaran 0.84 hingga 0.85. Hal ini menunjukkan bahwa pendekatan *stacking* mampu meningkatkan performa klasifikasi dibandingkan sebagian besar model dasar, terutama dalam konteks prediksi status kelulusan mahasiswa yang melibatkan data multi-kelas. Kombinasi RF-KNN dan KNN-SVM menjadi dua meta-model dengan performa terbaik, masing-masing mencapai akurasi 0.85 serta *weighted F1-score* 0.85. Nilai *weighted F1-score* yang tinggi dan seimbang ini mengindikasikan bahwa kedua kombinasi tersebut mampu beradaptasi dengan baik terhadap distribusi kelas yang tidak seimbang, sekaligus mempertahankan konsistensi prediksi pada ketiga kelas.

Sementara itu, kombinasi tiga model (RF-KNN-SVM) juga menunjukkan performa yang sangat baik dan stabil, dengan *precision*, *recall*, dan *F1-score* yang berada pada nilai 0.85 untuk seluruh metrik. Meskipun peningkatan akurasinya tidak signifikan dibandingkan *stacking* dua model, meta-model tiga kombinasi ini memberikan keunggulan dari sisi keseimbangan performa antar kelas, menandakan

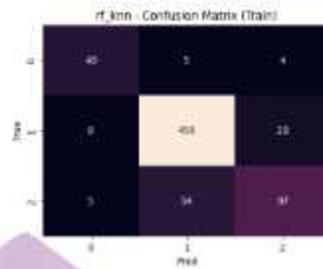
bahwa integrasi *meta-features* dari ketiga base model mampu meratakan kemampuan klasifikasi di semua kategori.

Tahap evaluasi selanjutnya adalah menggunakan *confusion matrix* untuk menilai performa model *stacking ensemble* pada setiap kombinasi model dasar. Analisis ini bertujuan untuk menilai seberapa baik model mampu mengklasifikasikan setiap kategori status kelulusan mahasiswa secara tepat, serta mengidentifikasi pola kesalahan prediksi yang masih terjadi pada kelas tertentu. Sript untuk visualisasi *confusion matrix* terhadap performa meta model terhadap train set di tampilkan pada Gambar 4.13 sampai dengan Gambar 4.16.

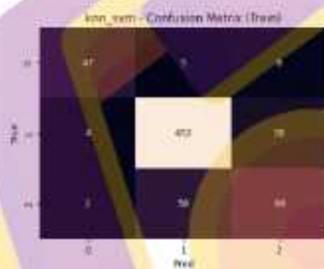
```
# Confusion matrix plotting
cm_tr = confusion_matrix(y_train, y_train_pred)
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
sns.heatmap(cm_tr, annot=True, fmt="d", cbar=False)
plt.title(f"{combo_name} - Confusion Matrix
(Train)")
plt.xlabel("Pred")
plt.ylabel("True")
```



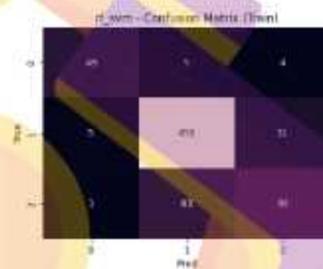
Gambar 4. 12 *Confusion matrix* meta model kombinasi 3 model dasar



Gambar 4. 13 *Confusion matrix* meta model kombinasi model RF-KNN



Gambar 4. 14 *Confusion matrix* meta model kombinasi model KNN-SVM



Gambar 4. 15 *Confusion matrix* meta model kombinasi model RF-SVM

Berdasarkan hasil evaluasi pada Gambar 4.13 hingga Gambar 4.16, seluruh meta-model menunjukkan kinerja yang lebih baik dibandingkan model dasar. Hal ini konsisten dengan rancangan *stacking*, di mana prediksi setiap model tunggal dapat saling melengkapi dan memberikan informasi tambahan pada *meta-learner*. Kondisi data latih yang telah melalui proses penyeimbangan menggunakan SMOTE juga berkontribusi terhadap meningkatnya akurasi, terutama pada kelas-kelas dengan jumlah sampel yang semula lebih sedikit.

Pada kombinasi dua model dasar, performa pada kelas 1 terlihat sangat dominan. Hampir seluruh sampel kelas ini berhasil diprediksi dengan benar, tercermin dari nilai diagonal yang mendekati 450 pada setiap kombinasi. Stabilitas yang tinggi pada kelas mayoritas mengindikasikan bahwa meta-model mampu menangkap pola inti dari mahasiswa yang lulus tepat waktu, sekalipun struktur data awal cukup beragam. Jika dicermati lebih rinci, setiap kombinasi dua model tetap menunjukkan beberapa kesalahan klasifikasi pada kelas 0 dan kelas 2, namun jumlahnya relatif kecil. Misalnya, pada kombinasi RF-KNN (Gambar 4.14), kesalahan prediksi pada kelas 0 hanya berada pada kisaran 5-7 sampel, sementara pada kelas 2 kesalahan masih muncul tetapi tetap dapat dikendalikan. Pola serupa terlihat pada kombinasi KNN-SVM dan RF-SVM, menandakan bahwa dua model dasar sudah cukup mampu mereduksi *error* meskipun belum sepenuhnya optimal pada kategori minoritas.

Sementara itu, hasil paling seimbang diperoleh dari kombinasi tiga model dasar (RF-KNN-SVM), Gambar 4.13. Meta-model ini menunjukkan penyebaran prediksi yang lebih seragam pada ketiga kelas. Pada kelas 0, jumlah prediksi benar mencapai 48 sampel, dengan kesalahan yang relatif sedikit. Pada kelas 2, prediksi benar meningkat menjadi 98 sampel, yang merupakan pencapaian tertinggi dibandingkan seluruh kombinasi dua model. Peningkatan ini menegaskan bahwa penggunaan tiga sumber prediksi sekaligus memberikan representasi fitur yang lebih kaya bagi meta-model, sehingga mampu membedakan pola perilaku mahasiswa yang lulus terlambat maupun yang berisiko keluar dari studi. Hasil ini sejalan dengan temuan (Fernandez-Garcia et al., 2021; Sahlaoui et al., 2021), yang

menyatakan bahwa pendekatan *stacking ensemble* efektif dalam memperkuat kemampuan generalisasi model sekaligus mengurangi ketimpangan antar kelas.

4.5.7 *Testing*

Tahapan *testing* model dilakukan untuk memperoleh gambaran yang objektif terhadap kemampuan model dalam menghadapi data baru, sehingga diperlukan pengujian performa menggunakan data uji yang sepenuhnya terpisah dari proses pelatihan (Buti et al., 2023; Fernandez-Garcia et al., 2021). Pengujian ini menjadi bagian penting dalam memastikan bahwa model tidak hanya bekerja baik pada data yang telah dipelajari, tetapi juga mampu mengenali pola yang belum pernah ditemui sebelumnya (Sahlaoui et al., 2021). Evaluasi terhadap data uji bertujuan untuk mengukur sejauh mana model yang dibangun mampu mempertahankan performa stabilitas model prediksi di luar data latih, serta mendeteksi potensi *overfitting* yang tersembunyi (Herianto et al., 2024). Oleh karena itu, analisis performa terhadap data uji memberikan dasar yang lebih kuat dalam menilai efektivitas model.

Sebelum proses pengujian dilakukan, sistem memverifikasi kesesuaian jumlah fitur pada data uji dengan konfigurasi yang digunakan saat pelatihan. Hal ini penting agar tidak terjadi ketidaksesuaian dimensi antara data uji dan struktur model yang telah dibentuk. Code dan hasil output disajikan pada Gambar 4.17 sebagai berikut.

```

print("INFO DATASET")
print(f"Jumlah data TRAINING : {X_train.shape[0]}
sampel, {X_train.shape[1]} fitur")
print(f"Jumlah data TESTING : {X_test.shape[0]}
sampel, {X_test.shape[1]} fitur")
print("Dataset sudah dibagi: model hanya dilatih di
TRAINING dan diuji di TESTING.")

```

```

-----
INFO DATASET
-----
Jumlah data TRAINING : 790 sampel, 15 fitur
Jumlah data TESTING : 175 sampel, 15 fitur
Dataset sudah dibagi: model hanya dilatih di TRAINING dan diuji di TESTING.
-----

```

Gambar 4. 16 Log info dataset

Pada tahapan *testing*, tiga model dasar yang sebelumnya telah dilatih yaitu *Random Forest* (RF), *K-Nearest Neighbors* (KNN), dan *Support Vector Machine* (SVM) dimuat dari berkas hasil pelatihan. Ketiga model tersebut berfungsi sebagai *base learners* dalam proses pembentukan *meta-feature*. Sama seperti tahap pelatihan model, dalam proses pengujian dilakukan empat skenario kombinasi model meta yang sudah disimpan pada tahap pelatihan sebelumnya, meliputi kombinasi RF-KNN, kombinasi RF-SVM, kombinasi KNN-SVM, dan kombinasi RF-KNN-SVM. Berikut ini adalah *code* dan Tabel 4.13 sebagai *output* dari pengujian meta-model XGBoost.

```

# EVALUASI TEST
meta_test_final=np.zeros((X_test.shape[0], len(cols)),
dtype=int)
for j, col_idx in enumerate(cols):
    model_for_meta = loaded_bases[col_idx]
    meta_test_final[:, j] =
model_for_meta.predict(X_test)
# gabungkan fitur asli + meta-features untuk TEST
X_test_final_comb=build_combined_from_meta(meta_test_fina
l, X_test)
# prediksi akhir dari META-MODEL (XGBoost) di TEST SET
y_test_pred = final_meta.predict(X_test_final_comb)

```

Tabel 4. 15 Output 20 prediksi meta-model versi Test Set

Kombinasi Model	Output 20 Pertama
Actual (y_{test})	[1.0, 1.0, 1.0, 2.0, 0.0, 1.0, 1.0, 2.0, 1.0, 2.0, 2.0, 1.0, 1.0, 2.0, 1.0, 1.0, 1.0, 2.0, 2.0, 1.0, ...]
3models (y_{pred})	[1, 1, 1, 0, 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 0, 2, ...]
rf_knn (y_{pred})	[1, 1, 1, 0, 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 0, 2, ...]
rf_svm (y_{pred})	[1, 1, 1, 0, 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 0, 2, ...]
knn_svm (y_{pred})	[1, 1, 1, 0, 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 0, 2, ...]

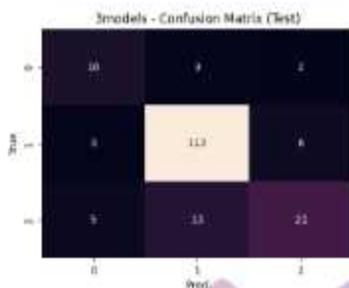
Selanjutnya, dilakukan evaluasi lagi terhadap performa setiap kombinasi meta-model terhadap data uji untuk menilai konsistensi dan kemampuan generalisasi model. Sama seperti evaluasi yang diterapkan pada pelatihan model sebelumnya, pada penilaian performa model *stacking* terhadap data uji juga dilakukan menggunakan *Confusion matrix*, *accuracy*, *precision*, *recall*, *F1-score* dan metrik AUC-ROC. Dengan menggunakan kelima metrik tersebut, setiap meta-model dapat dievaluasi baik pada data pelatihan maupun pengujian untuk mengamati adanya potensi *overfitting*, *underfitting*, serta kestabilan performa di berbagai kombinasi model dasar.

4.5.7.1 Confusion Matrix

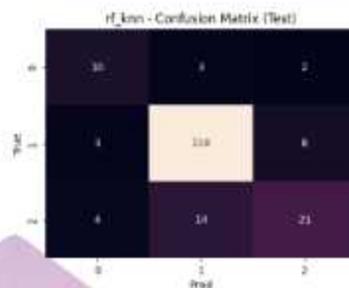
Tahapan pengujian meta-model dapat dilihat pada *listing code* di bawah ini. Hasil evaluasi dan visualisasi *confusion matrix* untuk setiap kombinasi model disajikan pada Gambar 4.18 hingga Gambar 4.21. Gambar-gambar tersebut menampilkan *confusion matrix* yang memberikan visualisasi distribusi hasil prediksi uji model terhadap label aktual. Visualisasi ini disajikan dalam bentuk *heatmap*, sehingga memudahkan dalam mengidentifikasi area di mana pengujian model *Stacking XGBoost* berhasil maupun gagal dalam melakukan klasifikasi.

```
# Confusion matrix plot
cm_te = confusion_matrix(y_test, y_test_pred)

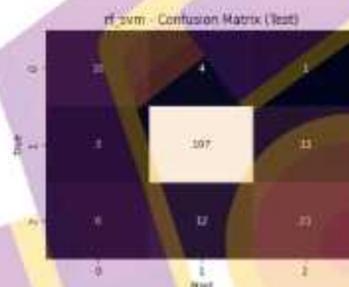
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 2)
sns.heatmap(cm_te, annot=True, fmt="d",
            cbar=False)
plt.title(f"{combo_name} - Confusion
Matrix (Test)")
plt.xlabel("Pred")
plt.ylabel("True")
plt.show()
```



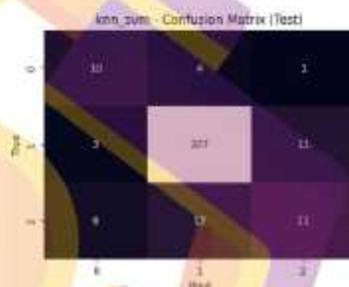
Gambar 4. 17 *Confusion matrix* uji model stacking kombinasi RF-KNN-SVM



Gambar 4. 18 *Confusion matrix* uji model stacking kombinasi RF-KNN



Gambar 4. 19 *Confusion matrix* uji model stacking kombinasi RF-SVM



Gambar 4. 20 *Confusion matrix* uji model stacking kombinasi KNN-SVM

Berdasarkan hasil pengujian pada Gambar 4.18 hingga Gambar 4.21, seluruh meta-model menunjukkan pola prediksi yang konsisten, terutama pada kelas 1 sebagai kelas dengan jumlah kasus terbesar. Pada seluruh kombinasi model, jumlah prediksi benar pada kelas ini berada pada kisaran 107 hingga 112, yang menunjukkan bahwa meta-model mampu mengenali mahasiswa yang lulus tepat waktu secara cukup stabil meskipun variasi model dasar yang digunakan berbeda.

Pada kombinasi tiga model (RF-KNN-SVM) sebagaimana ditampilkan pada Gambar 4.18, prediksi benar pada kelas 1 mencapai 112 kasus, sedikit lebih

tinggi dibandingkan kombinasi dua model. Selain itu, model ini mencatat 21 prediksi benar pada kelas 2, yang merupakan nilai tertinggi di antara seluruh kombinasi. Jumlah ini menunjukkan bahwa meta-model tiga basis mampu menangkap pola mahasiswa yang lulus terlambat dengan lebih baik, meskipun sebagian prediksi kelas 2 masih terdapat kesalahan prediksi ke kelas 1.

Selanjutnya kombinasi 2 model dasar menunjukkan performa cukup tinggi dalam memprediksi mahasiswa yang lulus tepat waktu (kelas 1). Berdasarkan *confusion matrix* kombinasi model RF-KNN, (Gambar 4.19), prediksi benar pada kelas 1 tercatat 110 kasus, sementara kelas 2 memperoleh 21 prediksi benar. Model ini juga mempertahankan ketepatan yang cukup baik pada kelas 0 dengan 10 prediksi benar, namun masih terdapat beberapa kesalahan klasifikasi yang mengarah ke kelas mayoritas. Kemudian pada kombinasi RF-SVM (Gambar 4.20), jumlah prediksi benar pada kelas 1 berada pada angka 107, lebih rendah dibanding dua kombinasi yang lain. Sementara itu, kelas 2 memperoleh 21 prediksi benar, tetapi masih terdapat perpindahan prediksi ke kelas lain, terutama menuju kelas 1. Hal ini mengindikasikan kombinasi RF-SVM cukup kuat pada kelas utama, namun relatif kurang stabil pada kelas minoritas. Sedangkan kombinasi KNN-SVM, Gambar 4.21, menunjukkan pola yang hampir serupa dengan RF-SVM. Jumlah prediksi benar pada kelas 1 mencapai 107 kasus, dan kelas 2 memperoleh 21 prediksi benar. Performa pada kelas 0 relatif konsisten dengan 10 prediksi benar, namun masih tampak adanya misklasifikasi ke kelas 1 ataupun kelas 2.

Hasil pengujian menunjukkan bahwa semua meta-model lebih mudah mengenali kelas dominan (kelas 1), sedangkan tantangan terbesar masih berada

pada kelas 0 dan kelas 2, yang secara alami memiliki karakteristik lebih bervariasi serta jumlah data yang sedikit atau tidak berimbang dengan kelas 1. Kombinasi tiga model (RF-KNN-SVM) memberikan hasil paling seimbang karena selain mencatat prediksi benar terbanyak pada kelas 1, model ini juga mencatat perbaikan yang tampak pada kelas 2. Hal tersebut menegaskan bahwa integrasi tiga model dasar memberikan informasi prediktif yang lebih kaya kepada meta-model, sehingga meningkatkan kemampuan dalam membedakan pola kelulusan mahasiswa secara lebih menyeluruh. Meskipun demikian, model dengan kombinasi 3 *base-model* masih terlihat adanya kecenderungan untuk mengarahkan prediksi ke kelas mayoritas. Hal ini ditemui karena dataset yang digunakan tidak seimbang, meskipun pada tahap pelatihan telah diterapkan SMOTE. Kondisi ini menunjukkan bahwa penyempurnaan lebih lanjut masih diperlukan untuk meningkatkan sensitivitas terhadap kelas-kelas minoritas yang jumlahnya lebih sedikit.

4.5.7.2 Matrik Akurasi, *Recall*, *Precision*, dan *F1-Score*

Selanjutnya evaluasi menggunakan matrik umum pengklasifikasian seperti akurasi, *recall*, *precision*, dan *F1-score* selengkapnya disajikan dan diinterpretasikan pada bagian berikut ini.

a. Kombinasi *Meta-Model RF-KNN-SVM* (Tiga Model Dasar)

Hasil pengujian meta-model kombinasi tiga model dasar *Random Forest*, KNN, dan SVM disajikan dalam bentuk Tabel 4.14 di bawah ini.

Tabel 4. 16 Metrik uji *stacking* model 3 model

Kelas	Metrik	<i>Train Set</i>	<i>Test Set</i>
Class 0	<i>Precision</i>	0.87	0.56
	<i>Recall</i>	0.83	0.67

	<i>F1-Score</i>	0.85	0.61
	<i>Support</i>	58	15
Class 1	<i>Precision</i>	0.88	0.88
	<i>Recall</i>	0.92	0.93
	<i>F1-Score</i>	0.90	0.90
	<i>Support</i>	486	121
Class 2	<i>Precision</i>	0.73	0.72
	<i>Recall</i>	0.63	0.54
	<i>F1-Score</i>	0.68	0.62
	<i>Support</i>	156	39
Akurasi		0.85	0.82
Rata-rata (Weighted)	<i>Precision</i>	0.85	0.81
	<i>Recall</i>	0.85	0.82
	<i>F1-Score</i>	0.85	0.81

Berdasarkan hasil evaluasi meta-model kombinasi tiga model dasar yang disajikan pada Tabel 4.14, performa yang diperoleh menunjukkan konsistensi yang cukup baik antara data latih dan data uji. Pada tahap pelatihan, model menghasilkan akurasi sebesar 0,85 dengan *weighted F1-score* 0,85, sedangkan pada data uji akurasinya berada pada kisaran 0,82 dengan *weighted F1-score* 0,81. Perbedaan nilai yang relatif kecil antara data latih dan data uji mengindikasikan bahwa model mampu melakukan generalisasi dengan baik, tanpa menunjukkan kecenderungan *overfitting* yang berarti.

Jika ditinjau pada level per kelas, performa model cukup kuat pada kategori kelas 1 (lulus tepat waktu). Pada data latih, kelas ini mencapai *recall* sebesar 0,92 dan *F1-score* 0,90. Sementara pada data uji, *recall* tetap tinggi pada 0,93 dengan *F1-score* 0,90. Stabilitas nilai ini memperlihatkan bahwa meta-model mampu mengenali pola mahasiswa yang lulus tepat waktu secara konsisten, terlepas dari variasi data.

Sebaliknya, kelas 0 (mahasiswa dikeluarkan) dan kelas 2 (lulus terlambat) menunjukkan performa yang cukup menantang. Pada data uji, kelas 0 mencatat *precision* 0,56 dan *recall* 0,67, sedangkan kelas 2 mencatat *precision* 0,72 dan *recall* 0,54. Meskipun *recall* kelas 2 lebih rendah dibandingkan kelas 1, nilai *F1-score* yang mencapai 0,62 menunjukkan bahwa model tetap memiliki kemampuan untuk membedakan kategori ini, meskipun tidak setinggi kelas mayoritas. Pola ini umum terjadi pada dataset yang cenderung tidak seimbang, meskipun proses pelatihan pada penelitian ini telah melalui penyeimbangan menggunakan SMOTE.

Dibandingkan dengan kombinasi dua model dasar, kombinasi 3 model menunjukkan distribusi prediksi yang lebih stabil pada seluruh kelas serta nilai *macro-F1* yang lebih baik, yaitu 0,81 pada data latih dan 0,71 pada data uji. Hal ini menunjukkan bahwa meta-model berhasil menggabungkan kontribusi dari ketiga model dasar dan menghasilkan representasi prediksi yang lebih kuat. Hasil ini memperlihatkan bahwa model tidak hanya bekerja baik pada kelas mayoritas, tetapi juga menunjukkan peningkatan pada prediksi kelas minoritas. Dengan demikian, meta-model dengan menggunakan kombinasi 3 *base-model* menjadi konfigurasi *stacking* yang paling efektif dalam memetakan prediksi dini status kelulusan.

b. Kombinasi Meta-Model RF-KNN

Hasil pengujian meta-model kombinasi *Random Forest* dan *K-Nearest Neighbors* (RF-KNN) disajikan dalam bentuk Tabel 4.15 di bawah ini.

Tabel 4. 17 Metrik uji *stacking* model RF-KNN

Kelas	Metrik	Train Set	Test Set
Class 0	<i>Precision</i>	0.79	0.59
	<i>Recall</i>	0.84	0.67
	<i>F1-Score</i>	0.82	0.62
	<i>Support</i>	58	15
Class 1	<i>Precision</i>	0.88	0.87
	<i>Recall</i>	0.93	0.91
	<i>F1-Score</i>	0.90	0.89
	<i>Support</i>	486	121
Class 2	<i>Precision</i>	0.75	0.68
	<i>Recall</i>	0.62	0.54
	<i>F1-Score</i>	0.68	0.60
	<i>Support</i>	156	39
Akurasi		0.85	0.81
Rata-rata (Weighted)	<i>Precision</i>	0.85	0.80
	<i>Recall</i>	0.85	0.81
	<i>F1-Score</i>	0.85	0.80

Berdasarkan hasil yang telah disajikan pada Tabel 4.15, meta-model kombinasi RF-KNN menunjukkan performa yang stabil cukup stabil pada kedua tahap pengujian. Pada data latih, model menghasilkan akurasi sebesar 0,85 dengan *weighted F1-score* 0,85, sedangkan pada data uji akurasinya 0,81 dengan *weighted F1-score* 0,80. Selisih performa yang relatif kecil antara data latih dan uji menunjukkan model minim mengalami kecenderungan *overfitting* yang berarti mampu mempertahankan pola prediksi dengan baik pada data baru.

Jika ditinjau per kelas, performa tertinggi kembali ditunjukkan pada kelas 1. Pada data latih, kelas ini mencatat *recall* sebesar 0,93 dan *F1-score* 0,90, sedangkan pada data uji *recall* 0,91 dengan *F1-score* 0,89. Nilai ini menunjukkan bahwa kombinasi RF-KNN efektif mengenali karakteristik kelas lulus tepat waktu. Pada kelas 0, model memberikan *recall* sebesar 0,84 pada data latih dan 0,67 pada data uji. Meskipun terdapat penurunan, performa ini masih menunjukkan model mampu

menangkap pola penting pada kelas minoritas tersebut. Sementara itu, kelas 2 memperlihatkan *recall* 0,62 pada data latih dan 0,54 pada data uji, dengan *F1-score* 0,60. Rendahnya performa pada kelas ini mengindikasikan bahwa pola belajar cenderung lebih kompleks dan mungkin memiliki kemiripan pola data dengan kelas lain, sehingga lebih sulit dipisahkan bahkan setelah proses penyeimbangan data.

Secara umum, meta-model RF-KNN memberikan kombinasi yang cukup baik antara akurasi, stabilitas, dan generalisasi. Selisih akurasi sekitar 4% antara data latih dan uji dapat dianggap wajar, terutama karena dataset latih telah melalui proses penyeimbangan data sehingga model memperoleh representasi yang lebih merata antar kelas. Namun, ketika diuji pada data yang tidak mengalami *oversampling*, sensitivitas terhadap kelas minoritas kembali menurun karena distribusi kelas yang tidak merata. Meski demikian sudah terdapat peningkatan pada kelas minoritas dibandingkan model tanpa penyeimbangan data. Kombinasi RF-KNN dapat dianggap sebagai alternatif meta-model, meskipun masih belum sekuat kombinasi tiga model dalam hal pemerataan performa antar kelas.

c. Kombinasi *Meta-Model RF-SVM*

Hasil pengujian meta-model kombinasi *Random Forest* dan *Support Vector Machine* (RF-SVM) disajikan dalam bentuk Tabel 4.16 di bawah ini.

Tabel 4. 18 Metrik uji *stacking* model RF-SVM

Kelas	Metrik	Train Set	Test Set
Class 0	<i>Precision</i>	0.89	0.53
	<i>Recall</i>	0.84	0.67
	<i>F1-Score</i>	0.87	0.59
	<i>Support</i>	58	15
Class 1	<i>Precision</i>	0.87	0.87
	<i>Recall</i>	0.93	0.88

	<i>F1-Score</i>	0.90	0.88
	<i>Support</i>	486	121
Class 2	<i>Precision</i>	0.72	0.64
	<i>Recall</i>	0.59	0.54
	<i>F1-Score</i>	0.65	0.58
	<i>Support</i>	156	39
Akurasi		0.84	0.79
Rata-rata (Weighted)	<i>Precision</i>	0.84	0.79
	<i>Recall</i>	0.84	0.79
	<i>F1-Score</i>	0.84	0.79

Berdasarkan hasil evaluasi meta-model kombinasi RF-SVM, yang telah disajikan pada Tabel 4.16, kinerja model terlihat konsisten pada data latih maupun data uji. Pada data pelatihan, akurasi mencapai 0,84 dengan *weighted F1-score* sebesar 0,84, sementara pada data uji akurasinya 0,79 dengan *weighted F1-score* 0,79. Selisih yang relatif kecil antara kedua tahap ini menunjukkan bahwa model mampu melakukan generalisasi secara memadai.

Performa terbaik tetap ditunjukkan pada kelas 1, dengan *recall* 0,93 pada data latih dan 0,88 pada data uji, menandakan bahwa model cukup efektif dalam mengenali mahasiswa yang lulus tepat waktu. Sementara itu, performa pada kelas 0 dan kelas 2 masih lebih rendah, dengan *recall* masing-masing 0,67 dan 0,54 pada data uji. Kondisi ini menggambarkan bahwa kategori minoritas tetap lebih sulit dipisahkan, meskipun proses pelatihan telah melalui penyeimbangan telah dilakukan.

d. Kombinasi Meta-Model KNN-SVM

Hasil pengujian meta-model kombinasi *K-Nearest Neighbors* dan *Support Vector Machine* (KNN-SVM) disajikan dalam bentuk Tabel 4.17 di bawah ini.

Tabel 4. 19 Metrik uji *stacking* model KNN-SVM

Kelas	Metrik	Train Set	Test Set
Class 0	Precision	0.89	0.53
	Recall	0.81	0.67
	F1-Score	0.85	0.59
	Support	58	15
Class 1	Precision	0.88	0.87
	Recall	0.93	0.88
	F1-Score	0.90	0.88
	Support	486	121
Class 2	Precision	0.73	0.64
	Recall	0.63	0.54
	F1-Score	0.68	0.58
	Support	156	39
Akurasi		0.85	0.79
Rata-rata (Weighted)	Precision	0.85	0.79
	Recall	0.85	0.79
	F1-Score	0.85	0.79

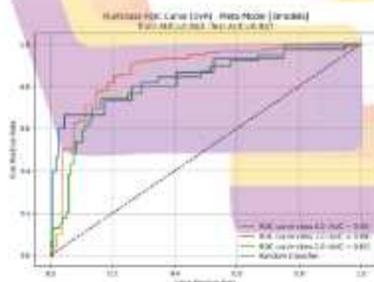
Berdasarkan hasil yang telah disajikan pada Tabel 4.18, meta-model kombinasi KNN-SVM, performa yang diperoleh berada pada tingkat yang serupa dengan kombinasi RF-SVM. Pada data latih, model mencapai akurasi 0,85 dengan *weighted F1-score* 0,85, sedangkan pada data uji akurasi berada pada kisaran 0,79 dengan *weighted F1-score* 0,79. Selisih performa antara data latih dan uji yang relatif kecil mengindikasikan bahwa model memiliki kemampuan generalisasi yang cukup baik.

Pada level per kelas, model tetap menunjukkan performa paling kuat pada kelas 1 (lulus tepat waktu), dengan *recall* 0,93 pada data latih dan 0,88 pada data uji. Hal ini menunjukkan bahwa kombinasi KNN-SVM mampu mengenali pola mahasiswa yang lulus tepat waktu secara konsisten. Sementara itu, performa pada kelas 0 dan kelas 2 masih lebih rendah, dengan *recall* masing-masing 0,67 dan 0,54

pada data uji. Meskipun demikian, nilai *precision* untuk kelas 2 mencapai 0,64, menunjukkan adanya peningkatan kemampuan model dalam membedakan kategori lulus terlambat dibandingkan beberapa kombinasi lainnya. Kombinasi ini menawarkan stabilitas yang cukup baik pada semua metrik utama dan dapat menjadi kandidat yang layak untuk konfigurasi *stacking*, terutama ketika dibutuhkan keseimbangan antara akurasi dan efisiensi pada dataset multi-kelas.

4.5.7.3 Metrik AUC-ROC

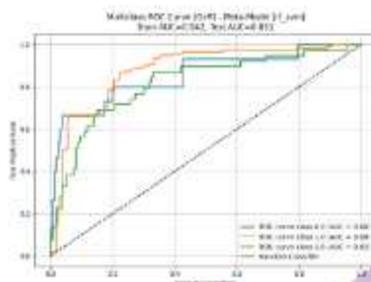
Selain metrik akurasi, *presisi*, dan *recall*, penelitian ini juga menggunakan metrik AUC-ROC untuk mengevaluasi kemampuan model dalam membedakan antar kelas, terutama pada data dengan distribusi tidak seimbang (Ghasemieh et al., 2023; Herianto et al., 2024). Visualisasi kurva ROC untuk setiap kombinasi meta-model pada penelitian ini ditampilkan pada Gambar 4.22 sampai dengan Gambar 4.25 berikut.



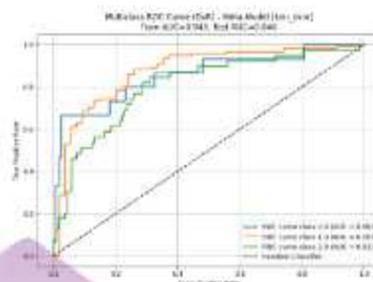
Gambar 4. 21 Kurva ROC kombinasi RF-KNN-SVM



Gambar 4. 22 Kurva ROC kombinasi RF-KNN



Gambar 4. 23 Kurva ROC kombinasi RF-SVM



Gambar 4. 24 Kurva ROC kombinasi KNN-SVM

Secara umum, seluruh meta-model menunjukkan performa yang kuat dengan nilai AUC pada data uji berada di kisaran 0,84-0,85, yang menggambarkan kemampuan model dalam membedakan ketiga kelas target secara konsisten. Kurva ROC pada Gambar 4.22 hingga Gambar 4.25 menampilkan pola pemisahan yang seragam, di mana masing-masing kelas memiliki area di bawah kurva yang berada di atas 0,80. Hal ini menunjukkan bahwa meta-model mampu menangkap perbedaan pola antar kelas meskipun karakteristiknya saling tumpang tindih.

Pada kombinasi tiga model dasar, nilai AUC mencapai 0,942 pada data latih dan 0,847 pada data uji, dengan selisih sekitar 0,095. Perbedaan yang tidak terlalu besar ini menunjukkan bahwa model tetap stabil ketika dihadapkan pada data baru. Pada tingkat kelas, nilai AUC berada di kisaran 0,85-0,88, menandakan kemampuan pemisahan yang merata di seluruh kategori.

Kombinasi RF-KNN juga memperlihatkan kinerja yang kuat, dengan AUC 0,938 pada data latih dan 0,849 pada data uji, menghasilkan selisih sekitar 0,089. Nilai ini merupakan salah satu yang paling kecil di antara seluruh kombinasi dua model, menunjukkan kemampuan generalisasi yang baik. Pada visualisasi ROC,

ketiga kelas terlihat tampil stabil dengan nilai AUC per kelas konsisten berada di antara 0.85-0.88.

Kombinasi RF-SVM memberikan tren yang serupa, dengan AUC 0.942 (train) dan 0.851 (test) serta selisih 0.090. Nilai AUC *test* yang berada di atas 0.85 mengindikasikan bahwa model tetap mampu memisahkan kelas dengan baik, walaupun sedikit di bawah performa RF-KNN pada aspek generalisasi.

Sementara itu, kombinasi KNN-SVM mencatat AUC 0.943 pada data latih dan 0.846 pada data uji, dengan selisih 0.097. Meskipun selisih ini sedikit lebih besar dibandingkan kombinasi lain, AUC *test* yang tetap berada di atas 0.84 menunjukkan bahwa pola pemisahan antar kelas masih terjaga. Ketiga kurva ROC untuk kelas 0, 1, dan 2 juga memperlihatkan konsistensi nilai AUC di kisaran 0.81-0.88.

Garis besarnya, seluruh meta-model menghasilkan performa yang solid dengan rentang AUC yang relatif seragam antar kombinasi. Dari perspektif generalisasi, RF-KNN memiliki selisih *train-test* AUC paling kecil dibandingkan kombinasi model lainnya. Sementara kombinasi tiga model menawarkan konsistensi antar kelas dan menjadi konfigurasi yang dinilai cukup seimbang untuk pemodelan prediksi kelulusan berbasis data pendaftaran awal.

4.5.8 Perbandingan Model

Hasil pengujian model *stacking ensemble* menunjukkan perbandingan performa seluruh kombinasi meta-model. Evaluasi akurasi dari setiap model memungkinkan kita untuk menentukan model mana yang memberikan kinerja

terbaik dalam mengklasifikasikan multi-kelas data kelulusan calon mahasiswa yang digunakan dalam penelitian ini. Hasil perbandingan dapat dilihat pada Tabel 4. 18.

Tabel 4. 20 Perbandingan hasil evaluasi model (train set dan test set)

Model	Akurasi (Train)	Akurasi (Test)	Weighted Precision (Train)	Weighted Precision (Test)	Weighted Recall (Train)	Weighted Recall (Test)	Weighted F1-Score (Train)	Weighted F1-Score (Test)
RF	0.7443	0.7886	0.7255	0.7870	0.7443	0.7886	0.7319	0.7838
SVM	0.6143	0.6229	0.7145	0.7230	0.6143	0.6229	0.6427	0.6519
KNN	0.5971	0.6343	0.6668	0.7149	0.5971	0.6343	0.6211	0.6592
RF-KNN	0.8514	0.8057	0.8469	0.8003	0.8514	0.8057	0.8474	0.8006
RF-SVM	0.8443	0.7886	0.8384	0.7884	0.8443	0.7886	0.8391	0.7868
KNN-SVM	0.8529	0.7886	0.8482	0.7884	0.8529	0.7886	0.8491	0.7868
RF-KNN-SVM	0.8500	0.8171	0.8453	0.8140	0.8500	0.8171	0.8464	0.8116

Berdasarkan Tabel 4.18, seluruh model baik model dasar maupun meta-model menunjukkan performa yang stabil antara data latih dan data uji. Selisih akurasi rata-rata yang berada di kisaran 3-5% mengindikasikan bahwa tidak terdapat kecenderungan *overfitting* yang berarti. Model tetap mampu mempertahankan pola prediksi ketika diuji pada data baru yang tidak mengalami proses *oversampling*. Dari seluruh model yang diuji, meta-model kombinasi tiga model dasar memberikan hasil paling kuat dengan akurasi 85% pada data latih dan 82% pada data uji, *weighted precision* 0.8140, *weighted recall* 0.8171, dan *weighted F1-score* 0.8116 pada data uji yaitu nilai tertinggi di antara seluruh konfigurasi.

Jika dibandingkan dengan penelitian sebelumnya, seperti oleh Fernandez-Garcia et al. (2021) dan Begum et al. (2021), telah menunjukkan bahwa data pendaftaran mahasiswa baru dapat digunakan sebagai dasar prediksi akademik

awal. Namun, akurasi yang dicapai pada tahap awal umumnya masih terbatas. Dalam penelitian Fernandez-Garcia et al. (2021), model hanya menghasilkan performa yang moderat ketika prediksi dilakukan murni dari data pendaftaran awal, dan peningkatan signifikan baru terjadi setelah model diperkuat dengan data nilai pada setiap semesternya. Penelitian tersebut tercatat mencapai akurasi maksimal 68,55% pada penerapan model *cascade ensemble* menggunakan dataset pendaftaran. Berbeda dengan penelitian ini secara tegas membatasi ruang lingkup fokus pada data pendaftaran saja. Meskipun inputnya jauh lebih sederhana, hasil pengujian menunjukkan bahwa model yang dibangun mampu mencapai performa prediksi yang cukup kuat. Meta-model *Xgboost* dengan kombinasi tiga model dasar (RF, KNN, dan SVM) dalam penelitian ini berhasil memperoleh konsistensi performa pada berbagai metrik evaluasi. Kondisi ini menunjukkan bahwa model mampu menjaga keseimbangan antara presisi dan sensitivitas di seluruh kelas, termasuk kelas minoritas, serta menunjukkan kemampuan generalisasi yang baik dalam skenario prediksi dini.

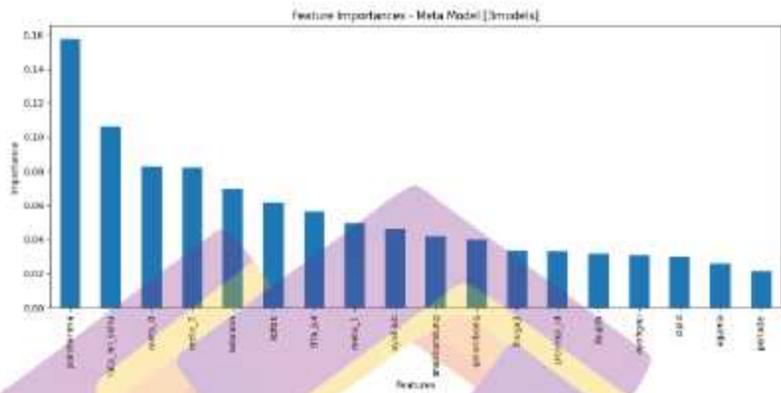
Selain itu, kontribusi penelitian ini tidak hanya terletak pada performa kuantitatif, tetapi juga pada dimensi metodologis yang tidak ditemukan pada studi sebelumnya. Pendekatan *stacking* dengan *out-of-fold meta-features*, penerapan SMOTE per *fold*, serta penggunaan SHAP untuk interpretasi memberikan nilai tambah yang signifikan dalam memahami hubungan antara faktor-faktor pendaftaran dan hasil kelulusan mahasiswa. Sehingga capaian penelitian terdahulu menunjukkan bahwa prediksi berbasis data pendaftaran memungkinkan, namun terbatas, sedangkan posisi penelitian saat ini menunjukkan bahwa prediksi dini

dapat dilakukan dengan lebih kuat, stabil, dan dapat dijelaskan, meskipun tanpa data akademik lanjutan. Hal ini mempertegas bahwa model yang dikembangkan dalam penelitian ini dapat digunakan sebagai alat pendukung keputusan yang efektif dalam memetakan risiko dan potensi kelulusan mahasiswa sejak awal proses penerimaan.

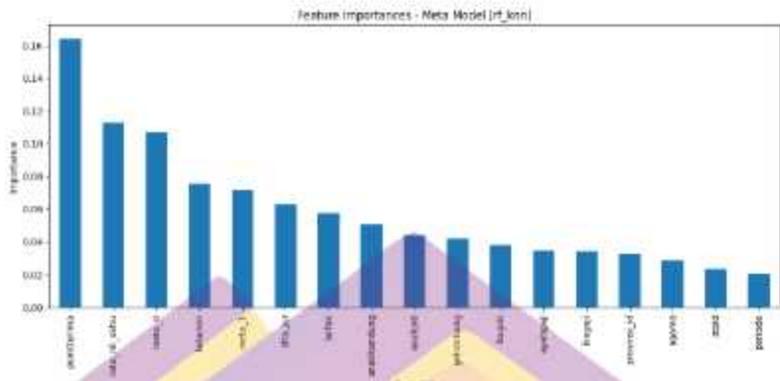
4.5.9 Analisis Interpretabilitas Model

Analisis interpretabilitas model *stacking ensemble* dilakukan dengan memanfaatkan teknik SHAP (SHapley Additive exPlanations) (Lundberg & Lee, 2017; Sahlaoui et al., 2021). SHAP digunakan untuk mengungkap dan memvisualisasikan logika prediksi dari model *ensemble* yang diterapkan. SHAP bekerja dengan menghitung kontribusi marginal rata-rata dari setiap fitur dalam semua kemungkinan kombinasi fitur yang mungkin, sehingga hasilnya bersifat adil dan dapat dijelaskan secara matematis (Lundberg & Lee, 2017). SHAP mendukung interpretasi secara global (misalnya, fitur mana yang paling mempengaruhi kelulusan secara keseluruhan) maupun secara lokal (misalnya, mengapa seorang mahasiswa tertentu diprediksi tidak lulus) (Sahlaoui et al., 2021). Pendekatan ini pertama kali diperkenalkan oleh Lloyd Shapley pada tahun 1953 dalam konteks teori permainan kooperatif, dan oleh karena itu nilai kontribusi yang dihitung dengan metode ini dikenal sebagai *Shapley values*.

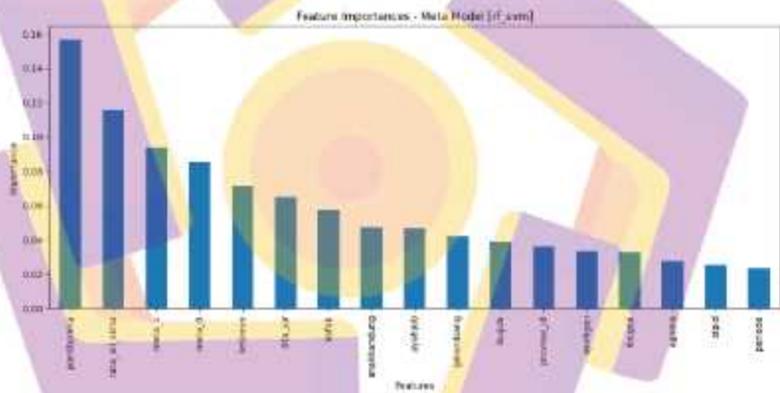
Hasil analisis teknik visualisasi menggunakan nilai SHAP untuk memberikan tampilan fitur-fitur yang memengaruhi performa calon mahasiswa, disajikan pada Gambar 4.26 sampai dengan Gambar 4.29 berikut ini.



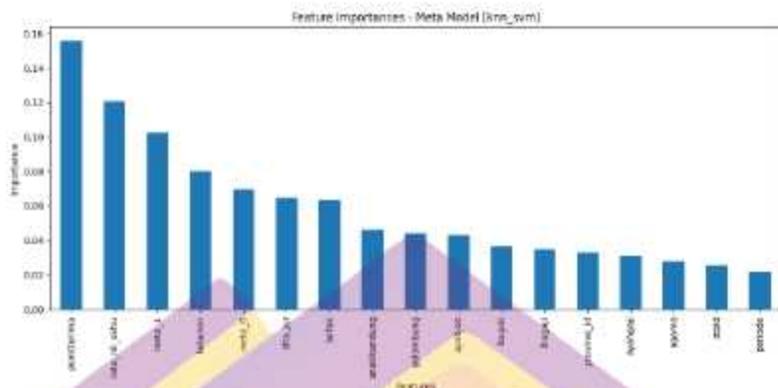
Gambar 4. 25 *Feature importance* pada model final *stacking* XGBoost 3 model



Gambar 4. 26 *Feature importance* pada model final *stacking* XGBoost RF-KNN



Gambar 4. 27 *Feature importance* pada model final *stacking* XGBoost RF-SVM



Gambar 4. 28 *Feature importance* pada model final *stacking* XGBoost KNN-SVM

Tabel 4. 21 *Feature Importance* Meta-Model *Stacking*

Fitur	3 Models	RF-KNN	RF-SVM	KNN-SVM
pointterima	0.15756	0.16449	0.15683	0.15606
rata_nil_skh	0.10637	0.11292	0.11597	0.12063
meta_0	0.08291	0.10671	0.08547	0.06977
meta_2	0.08190	—	—	—
kelamin	0.06959	0.07529	0.07165	0.07989
kofak	0.06186	0.05800	0.05760	0.06237
slta_jur	0.05640	0.06273	0.06492	0.06470
meta_1	0.04970	0.07181	0.09385	0.10252
ayahjob	0.04593	0.04404	0.04676	0.04032
anakkandung	0.04165	0.05056	0.04271	0.04608
gelombang	0.03988	0.04190	0.04207	0.04415
ibugaji	0.03363	0.03241	0.03875	0.03474
provinsi_id	0.03339	0.03259	0.03611	0.03099
ibujob	0.03158	0.03812	0.03251	0.03626
avahgaji	0.03154	0.03544	0.03353	0.03271
agama	0.02537	0.02874	0.02765	0.02717
stpid	0.02503	0.02341	0.02556	0.02556
periode	0.02133	0.02036	0.02326	0.02153

Hasil analisis *Feature Importance* (Tabel 4.19) dan SHAP pada seluruh meta-model menunjukkan pola kontribusi fitur yang konsisten. Pada kombinasi RF-KNN dan RF-SVM (Gambar 4. dan Gambar 4.), *meta-feature* yang berasal dari prediksi

model dasar memiliki pengaruh besar terhadap keputusan akhir. Fitur seperti *meta_0* atau *meta_1* menempati posisi teratas setelah variabel akademik utama. Hal ini memperlihatkan bahwa probabilitas dari model dasar berhasil menambah informasi baru yang tidak sepenuhnya ditangkap oleh fitur awal, sehingga memperkuat kemampuan meta-model dalam membedakan ketiga kelas kelulusan.

Pada meta-model KNN-SVM (Gambar 4.), kontribusi *meta-feature* tetap menonjol, namun fitur asli terutama *pointterima* (nilai masuk), *rata_nil_skhu*, dan fitur jenis kelamin tampil sebagai tiga prediktor paling penting. Peningkatan bobot fitur akademik ini menunjukkan bahwa performa data akademik awal mahasiswa masih menjadi sinyal yang paling kuat, diikuti oleh *meta feature* dari model dasar yang menambah konteks dalam pengambilan keputusan. Fitur sosial seperti status pekerjaan orang tua atau jumlah saudara kandung tetap berperan, meskipun berada pada kategori kontribusi menengah.

Pada kombinasi tiga model (Gambar 4.), pola kontribusi terlihat lebih merata. Fitur *pointterima* tetap menjadi faktor dominan, diikuti oleh *rata_nil_skhu*, kemudian *meta-feature* seperti *meta_0* dan *meta_2*. Jika dibandingkan dengan kombinasi dua model, keberadaan tiga *meta-feature* yang berbeda meningkatkan keseimbangan distribusi informasi. SHAP plot juga menunjukkan bahwa meta model tidak hanya bergantung pada satu jenis *meta-feature*, melainkan memanfaatkan seluruh prediksi dasar secara proporsional. Selain itu, fitur demografis seperti *kelamin*, *slta_jur*, serta faktor sosial seperti koefisien ekonomi keluarga seperti gaji dan pekerjaan orang tua tetap memberikan kontribusi tambahan dalam rentang yang stabil.

Secara keseluruhan, analisis Feature Importance dan SHAP memperlihatkan bahwa model *stacking* kombinasi tiga model dasar memiliki struktur kontribusi yang lebih seimbang dibandingkan kombinasi dua model. Meta-model memadukan sinyal kuat dari fitur akademik dengan *meta-feature* model dasar, tanpa mengabaikan informasi sosiodemografis. Pola ini mengonfirmasi bahwa pendekatan *stacking* tidak hanya meningkatkan akurasi, tetapi juga memperkaya struktur pengetahuan model dalam memahami faktor-faktor yang memengaruhi kelulusan mahasiswa sejak tahap pendaftaran.

Selain analisis berbasis SHAP yang digunakan untuk melihat fitur-fitur dominan, dilakukan pula pengelompokan kontribusi fitur (*feature importance*) berdasarkan kategori, diantaranya faktor demografis, faktor akademik, faktor sosioekonomi, dan *meta-features* pada setiap kombinasi model *stacking* menggunakan *XGBoost*. Bobot pengaruh antar faktor tersebut ditampilkan pada Tabel 4.20 berikut ini.

Tabel 4. 22 Persentase kontribusi berdasarkan feature importances dari meta-model

Kombinasi Model	Akademik (%)	Meta Features (%)	Sosioekonomi (%)	Demografi (%)
3models	47.29%	21.45%	18.38%	12.87%
rf knn	48.31%	17.85%	20.17%	13.66%
rf svm	48.61%	17.93%	19.92%	13.54%
knn svm	49.59%	17.23%	19.11%	14.07%

Berdasarkan Tabel 4.20, seluruh kombinasi meta-model menunjukkan pola kontribusi fitur yang relatif konsisten. Kategori akademik menjadi faktor paling dominan dengan kontribusi antara 47-50% pada semua meta-model. Hal ini

menegaskan bahwa indikator kemampuan awal mahasiswa seperti nilai tes masuk dan rata-rata nilai raport menjadi fitur prediktif terkuat dalam memetakan status kelulusan sejak tahap pendaftaran.

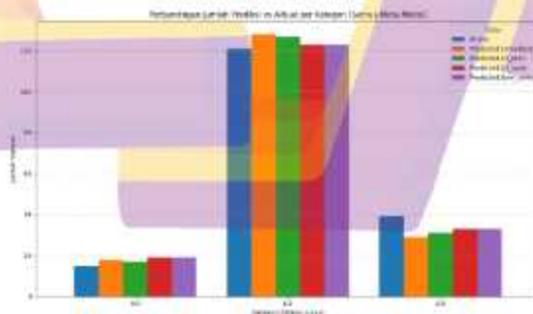
Selanjutnya kategori *meta-features*, yaitu keluaran probabilistik dari model dasar (*meta_0*, *meta_1*, *meta_2*), juga memberikan peran signifikan dengan kontribusi 17-21%. Nilai ini menunjukkan bahwa model *stacking* benar-benar memanfaatkan informasi tambahan dari *base learner* untuk memperkaya representasi data. Pada kombinasi tiga model, kontribusi *meta-feature* mencapai nilai tertinggi (21.45%), mengindikasikan bahwa meta-model menggunakan prediksi dasar secara lebih baik dalam meningkatkan akurasi.

Didamping itu faktor sosioekonomi dan faktor demografi secara berurutan berada pada rentang 18-20% dan 12-14%. Faktor sosioekonomi yang mencakup informasi terkait ekonomi keluarga, pekerjaan orang tua, dan faktor lingkungan juga berkontribusi terhadap pola kelulusan meskipun tidak sebesar fitur akademik. Terlihat dari konsistensi fitur seperti penghasilan dan pekerjaan orang tua yang selalu masuk dalam fitur berperan menengah di setiap model. Sejalan dengan itu, kategori demografi juga memainkan peran, meskipun persentasenya kecil. Fitur seperti jenis kelamin, asal provinsi, agama, dan status pindahan sebelumnya memberikan konteks tambahan bagi model meskipun tidak menjadi faktor utama. Pola kontribusi ini menunjukkan bahwa meta-model memprioritaskan sinyal akademik dan *meta-feature*, sementara faktor sosioekonomi dan demografi berfungsi sebagai pelengkap model dalam memprediksi. Kemudian untuk menunjukkan pentingnya menilai tidak hanya bagaimana model mengambil

keputusan, tetapi juga sejauh mana keputusan tersebut merepresentasikan realitas mahasiswa di lapangan. Oleh karena itu, evaluasi kesesuaian prediksi dan data aktual akan dibahas pada subbab berikutnya.

4.6. Evaluasi Relevansi Prediksi terhadap Data Aktual

Setelah dilakukan pengujian performa model melalui evaluasi metrik seperti akurasi, presisi, *recall*, dan *F1-score* pada bagian sebelumnya, langkah lanjutan yang krusial dalam penelitian ini adalah mengevaluasi relevansi hasil prediksi terhadap distribusi aktual status kelulusan mahasiswa dan dari data historis pendaftaran mahasiswa angkatan 2019. Evaluasi ini bertujuan untuk menilai sejauh mana model prediktif berbasis *Stacking Ensemble* yang dibangun mampu mencerminkan realitas di lapangan, khususnya dalam hal distribusi status kelulusan mahasiswa di perguruan tinggi. Visualisasi hasil prediksi masing-masing model meta (yaitu kombinasi dua atau tiga model dasar) dengan data aktual ditampilkan pada Gambar 4.30 berikut ini.



Gambar 4. 29 Grafik visualisasi relevansi data

Visualisasi pada Gambar 4.30 menyajikan perbandingan sampel hasil prediksi masing-masing model dengan data aktual mahasiswa pada setiap kategori

status. Dari grafik tersebut tampak bahwa seluruh model secara umum menunjukkan konsistensi dalam memetakan kelompok mahasiswa lulus tepat waktu (kategori 1), di mana jumlah prediksi berada sangat dekat dengan data aktual. Dari seluruh kombinasi, meta-model RF-KNN terlihat memberikan jumlah prediksi yang paling mendekati nilai aktual pada dua kategori penting, yaitu Dikeluarkan (0.0) dan Lulus Tepat Waktu (1.0). Kedekatan ini mengindikasikan bahwa model kombinasi RF-KNN memiliki sensitivitas yang baik baik terhadap kelas mayoritas maupun terhadap kelas *dropout* yang jumlahnya relatif kecil.

Sebaliknya, *GXBoost* dengan kombinasi tiga model menampilkan pola prediksi yang sedikit berbeda. Meskipun prediksi pada kategori 1 sangat dekat dengan jumlah aktual, model ini overestimasi pada (0.0) serta underestimasi pada kategori (2.0). Pola ini menunjukkan bahwa meta-model ini lebih sensitif terhadap sinyal *dropout* tetapi masih belum sepenuhnya mengimbangi kompleksitas kelas terlambat lulus yang jumlahnya lebih fluktuatif.

Di sisi lain, kombinasi RF-SVM dan KNN-SVM menunjukkan prediksi yang serupa serta relatif konsisten pada kategori 2, meskipun keduanya juga memprediksi jumlah yang sedikit lebih rendah dibanding data aktual. Hasil ini mengindikasikan kedua model tersebut menghadapi tantangan dalam mengenali pola mahasiswa yang lulus terlambat. Dengan demikian dapat digarisbawahi beberapa hal berikut, bahwa *stacking ensemble* model menggunakan *XGBoost* dengan *meta feature* RF-KNN unggul dalam mendekati distribusi aktual pada kelas 0 dan 1. Selanjutnya kombinasi 3 model lebih stabil secara keseluruhan, tetapi cenderung kurang presisi pada kategori 2. Terakhir kombinasi meta fitur RF-SVM

dan KNN-SVM menghasilkan prediksi menengah yang cukup konsisten antar kategori. Temuan ini dikonfirmasi lebih lanjut melalui analisis tabel perbandingan data aktual dengan prediksi di Tabel 4.21 sebagai berikut.

Tabel 4. 23 Tampilan sebagian data prediksi model terhadap data aktual 15 dari 175 sampel

Sample	<i>Actual_Status</i>	Pred (3models)	Pred (rf_knn)	Pred (rf_svm)	Pred (knn_svm)
Sample 1218	1	1	1	1	1
Sample 1414	1	1	1	1	1
Sample 761	1	1	1	1	1
Sample 802	2	0	0	0	0
Sample 884	0	2	2	2	2
Sample 1213	1	1	1	1	1
Sample 66	1	1	1	1	1
Sample 30	1	1	1	1	1
Sample 211	1	1	1	1	1
Sample 210	2	2	2	2	2
Sample 234	2	2	2	2	2
Sample 1219	1	1	1	1	1
Sample 537	1	1	1	1	1
Sample 400	2	2	2	2	2
Sample 274	1	1	1	1	1

Pada Tabel 4.21 menampilkan prediksi dengan data aktual pada sampel individual. Pada beberapa sampel, seperti Sample_884 dan Sample_802, seluruh model memberikan prediksi yang seragam meskipun berbeda dari nilai aktualnya. Temuan ini mengindikasikan bahwa sampel tersebut memiliki pola fitur yang mengarah kuat pada kelas tertentu, sehingga seluruh prediksi meta-model mengarah ke prediksi yang sama. Kondisi ini wajar terjadi pada kasus multi-kelas dengan distribusi dan batas kelas yang tidak sepenuhnya terpisah, meskipun SMOTE telah membantu menyeimbangkan data latih (Sahlaoui et al., 2021).

Sementara itu, sebagian besar sampel lainnya seperti Sample_1218, Sample_1414, dan Sample_761 menunjukkan bahwa seluruh meta-model mampu memberikan prediksi yang sesuai dengan label sebenarnya. Konsistensi prediksi ini sejalan dengan evaluasi akurasi dan *F1-score*, di mana kombinasi tiga model cenderung memberikan hasil paling stabil, diikuti oleh dua kombinasi lainnya.

Dari sisi relevansi kontekstual, pendekatan *stacking* mampu menangkap pola kelulusan mahasiswa baru dengan lebih stabil dibandingkan model tunggal. Penerapan SMOTE pada tahap pelatihan membantu mengurangi bias terhadap kelas mayoritas, sehingga meta-model dapat memanfaatkan informasi dari *base learner* secara lebih seimbang. Hal ini terlihat dari konsistensi prediksi pada sebagian besar sampel uji, terutama pada kelas mayoritas yang memiliki karakteristik lebih terdefinisi dalam data. Meskipun demikian, model masih menunjukkan deviasi pada beberapa sampel kelas minoritas. Kondisi ini masih umum terjadi mengingat struktur data pendaftaran mahasiswa baru cenderung tidak sepenuhnya terpisah antar kelas, sementara itu variabel prediktor yang bersifat administratif dan variasi fitur yang minim juga belum sepenuhnya merepresentasikan dinamika akademik mahasiswa. Meskipun demikian, performa meta-model yang stabil baik pada pelatihan-pengujian menandakan bahwa model tidak hanya akurat secara statistik, tetapi juga memiliki relevansi praktis dalam konteks pemetaan awal risiko akademik. Hasil relevansi ini dapat menjadi awal pengembangan model melalui penguatan fitur prediktor (peningkatan kedalaman informasi/ eksplorasi variabel) serta penyempurnaan teknik penyeimbangan data akan membantu meta-model menghasilkan prediksi yang lebih presisi pada seluruh kelas.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa:

1. Dengan menggunakan SHAP dan *feature importance* pada meta model, atribut yang paling berpengaruh terhadap kemampuan model dalam memprediksi kelulusan mahasiswa sejak dini adalah faktor akademik, dengan bobot pengaruh 47-50%, terutama pada fitur nilai tes CBT dan rata-rata nilai raport. Kontribusi berikutnya berasal dari *meta-features* sebesar 17-21%, yang menandakan bahwa model dasar mampu memberikan informasi tambahan untuk membantu meta-model meningkatkan ketepatan klasifikasi. Selanjutnya, faktor sosioekonomi seperti pekerjaan dan penghasilan orang tua ber kontribusi 18-20%. Sementara itu, faktor demografis berkontribusi paling rendah 12-14%, namun tetap menambah informasi dalam memodelkan perbedaan karakteristik data. Integrasi informasi dari berbagai jenis data, ditambah penerapan stratifikasi, pemilihan parameter, validasi silang dan SMOTE berhasil menekan risiko *overfitting* pada kelas mayoritas dalam klasifikasi multikelas, sehingga meningkatkan kualitas dan keandalan prediksi *stacking ensemble*.
2. Perbandingan performa beberapa model *Stacking Ensemble* dengan *XGBoost* sebagai meta-model menunjukkan model *stacking ensemble* kombinasi tiga model dasar (*meta-features*) memberikan performa paling kuat dan konsisten dalam mengenali kelas mayoritas maupun minoritas dibandingkan kombinasi

dua model. Berdasarkan hasil evaluasi kombinasi tiga model mencatat akurasi 85% pada data latih dan 0.82% pada data uji, dengan nilai *weighted precision* (85% vs 81%), *recall* (85% vs 82%), dan *F1-score* (85% vs 81%). Hal ini juga diperkuat oleh selisih AUC *train-test* yang rendah (0.942-0.847), yang menunjukkan bahwa model mampu melakukan generalisasi dengan baik pada data baru, sehingga minim potensi model *overfitting* yang signifikan. Selain itu, nilai AUC 0.85 (Dropout), 0.88 (Tepat Waktu), 0.82 (Terlambat) menunjukkan pemisahan kelas yang baik pada ketiga kategori kelulusan. Keseimbangan antar metrik juga penting untuk menjaga proporsi kesalahan antar kelas dalam klasifikasi multikelas. Hasil penelitian menunjukkan model *stacking* kombinasi tiga model mampu menggabungkan keunggulan masing-masing model dasar, di mana RF memberikan kemampuan generalisasi, SVM memberikan pemisahan batas kelas yang kuat, dan KNN memperkuat kemampuan klasifikasi berbasis kedekatan data.

3. Hasil prediksi yang diperoleh dari model *stacking ensemble*, khususnya pada kombinasi tiga model dasar (Random Forest, Support Vector Machine, dan K-Nearest Neighbor) yang dimediasi oleh XGBoost sebagai *meta-learner*, menunjukkan tingkat relevansi yang tinggi terhadap data aktual kelulusan mahasiswa. Evaluasi dilakukan dengan membandingkan hasil prediksi terhadap status kelulusan asli pada data mahasiswa tahun 2019. Seluruh model mampu menangkap pola dominan pada kategori lulus tepat waktu dengan akurasi yang baik, sedangkan variasi prediksi pada kategori minoritas (dikeluarkan dan

terlambat lulus) membuka peluang untuk dikembangkan lebih lanjut dalam konteks prediksi dini mahasiswa berisiko.

Dibandingkan penelitian terdahulu yang menggunakan data pendaftaran sebagai dasar prediksi awal, penelitian ini menunjukkan performa prediksi yang stabil dapat dicapai tanpa memerlukan data akademik tambahan. Studi seperti (Fernandez-Garcia et al., 2021) memperoleh peningkatan akurasi yang signifikan hanya setelah menambahkan data capaian per semester, sementara penelitian ini menunjukkan bahwa pendekatan *stacking* dengan *meta-features* dan penyeimbangan SMOTE mampu menghasilkan prediksi dini yang kuat pada tahap pendaftaran. Dengan demikian, penelitian ini memberikan kontribusi metodologis berupa model *stacking* dengan *XGBoost* sebagai *meta-learner* yang membuktikan bahwa prediksi kelulusan secara dini dengan data historis pendaftaran memungkinkan dilakukan, bahkan sebelum memperoleh data akademik lanjutan. Model terbukti efektif dalam menjaga stabilitas, keseimbangan prediksi multikelas. Selain itu, model prediksi yang dikembangkan dalam penelitian ini tidak hanya unggul dari sisi performa metrik, tetapi juga menunjukkan tingkat representatif yang baik terhadap realitas di lapangan, yang merupakan aspek krusial dalam integrasi *machine learning* pada sistem akademik perguruan tinggi berbasis data.

5.2 Saran

Penelitian ini difokuskan pada pengembangan model prediksi kelulusan sejak tahap awal pendaftaran mahasiswa baru dengan memanfaatkan data historis seleksi. Berdasarkan hasil yang diperoleh, terdapat beberapa hal yang dapat dijadikan acuan bagi penelitian berikutnya yang bersifat teknis dan non-teknis.

Saran Teknis:

1. Implementasi Sistem Prediksi dalam Sistem PMB

Temuan penelitian ini memiliki implikasi praktis bagi kebijakan penerimaan mahasiswa baru. Model prediksi yang dibangun dapat dijadikan alat bantu analitik untuk mendeteksi calon mahasiswa dengan risiko akademik sejak tahap pendaftaran. Penelitian mendatang dapat diarahkan pada implementasi model prediksi ke dalam sistem penerimaan mahasiswa baru (SPMB) sebagai fitur tambahan berbasis *dashboard* web. Sistem ini dapat menampilkan klasifikasi risiko kelulusan secara otomatis bagi setiap pendaftar, yang berguna bagi pimpinan sebagai bahan pertimbangan dalam evaluasi kebijakan seleksi masuk. Sehingga pertimbangan seleksi tidak hanya menitikberatkan pada nilai akademik, tetapi juga pada faktor sosial ekonomi calon mahasiswa. Perguruan tinggi dapat merancang program pembinaan akademik sejak dini, seperti pelatihan dasar mata kuliah umum atau pendampingan adaptasi belajar bagi kelompok calon mahasiswa yang diprediksi memiliki risiko keterlambatan studi. Integrasi ini juga memungkinkan institusi melakukan pemantauan berkelanjutan terhadap pola prediksi dari tahun ke tahun.

2. Perbaikan Kualitas Data

Penelitian selanjutnya disarankan untuk memperluas cakupan variabel pada tahap pendaftaran, seperti menambahkan data psikometrik, minat studi, nilai keterampilan, aktivitas ekstrakurikuler atau riwayat aktivitas akademik calon mahasiswa. Penambahan variabel tersebut diharapkan dapat memperkaya fitur

prediksi dan meningkatkan performa model dalam menilai potensi kelulusan mahasiswa sejak awal penerimaan.

3. Pengembangan Model Lebih Lanjut

Penelitian selanjutnya dapat mengeksplorasi beberapa hal diantaranya menggunakan *meta-learner* alternatif (misalnya LightGBM atau CatBoost), eksplorasi teknik penyeimbangan dataset, dan penambahan fitur baru untuk membandingkan kinerja terhadap *XGBoost* yang digunakan dalam penelitian ini. Dengan langkah-langkah pengembangan tersebut, model prediksi berbasis data pendaftaran diharapkan dapat disempurnakan menjadi alat pendukung keputusan yang lebih adaptif, akurat, dan relevan dalam proses seleksi mahasiswa baru di perguruan tinggi.

4. Pengujian Performa Model

Model yang dikembangkan masih terbatas pada data pendaftaran di Universitas Slamet Riyadi dan hanya menggunakan satu angkatan (tahun 2019) sebagai representasi data aktual kelulusan. Untuk memperoleh gambaran relevansi yang lebih komprehensif, disarankan dilakukan evaluasi lintas tahun dan lintas institusi. Hal ini penting untuk menguji konsistensi model dalam merepresentasikan pola kelulusan aktual yang mungkin berbeda tergantung waktu, kebijakan akademik, dan konteks institusional.

Saran Non-Teknis:

1. Kebijakan Akademik Proaktif

Hasil model dapat digunakan untuk menyusun intervensi ini berupa program adaptasi pembelajaran atau pelatihan soft skills bagi kelompok mahasiswa yang diprediksi berisiko tinggi mengalami keterlambatan kelulusan.

2. Penguatan Literasi Data di PMB

Penerapan model ini mendorong kebutuhan akan literasi data dan teknologi di kalangan pengelola PMB dan pimpinan institusi, sehingga pengambilan kebijakan dapat lebih berbasis data.



DAFTAR PUSTAKA

- Alzahrani, M. R. (2024). Predicting Student Performance Using Ensemble Models and Learning Analytics Techniques. *Preprints.Org*, 202406.1100.v1. <https://doi.org/10.20944/preprints202406.1100.v1>
- Azar, A. G., Saghafian, M., & Jalili, M. (2024). A study of common pitfalls behind the generalization gap in machine learning. *OpenReview*.
- Ballester, P. L., Chen, Z., & Sapiro, G. (2024). Predicting the generalization gap in neural networks using topological data analysis. *arXiv:2203.12330*
- Begum, S., Ahmed, M., & Uddin, M. (2021). Genetically optimized ensemble classifiers for multiclass student performance prediction. *International Journal of Nonlinear Analysis and Applications*, 12(2), 1–15.
- Bujang, S. D. A., Selamat, A., Ibrahim, R., Krejcar, O., Herrera-Viedma, E., Fujita, H., & Ghani, N. A. M. (2021). Multiclass Prediction Model for Student Grade Prediction Using Machine Learning. *IEEE Access*, 9, 95608–95621. <https://doi.org/10.1109/ACCESS.2021.3093563>
- Butt, N. A., Mahmood, Z., Shakeel, K., Alfarhood, S., Safran, M., & Ashraf, I. (2023). Performance Prediction of Students in Higher Education Using Multi-Model Ensemble Approach. *IEEE Access*, 11, 136091–136108. <https://doi.org/10.1109/ACCESS.2023.3336987>
- Dick, G. (2024). *An Ensemble Learning Interpretation of Geometric Semantic Genetic Programming An Ensemble Learning Interpretation of Geometric Semantic Genetic Programming*. 0–25.

- Du, X., Ma, J., Gao, H., & Liu, Y. (2024). Methodology for evaluating the generalization of ResNet. *Applied Sciences*, 14(9), 3951.
- Elouafi, A., Tammouch, I., Eddarouch, S., & Touahni, R. (2024). Evaluating various machine learning methods for predicting students' math performance in the 2019 TIMSS. *Indonesian Journal of Electrical Engineering and Computer Science*, 34(1), 565–574.
<https://doi.org/10.11591/ijeecs.v34.i1.pp565-574>
- Fernandez-Garcia, A. J., Preciado, J. C., Melchor, F., Rodriguez-Echeverria, R., Conejero, J. M., & Sanchez-Figueroa, F. (2021). A real-life machine learning experience for predicting university dropout at different stages using academic data. *IEEE Access*, 9, 133076–133090.
<https://doi.org/10.1109/ACCESS.2021.3115851>
- Fernández-García, A. J., Rodríguez-Echeverría, R., Preciado, J. C., Conejero Manzano, J. M., & Sánchez-Figueroa, F. (2020). Creating a recommender system to support higher education students in the subject enrollment decision. *IEEE Access*, 8, 189069–189088.
<https://doi.org/10.1109/ACCESS.2020.3031572>
- Ghasemieh, A., Lloyed, A., Bahrami, P., Vajar, P., & Kashef, R. (2023). A novel machine learning model with Stacking Ensemble Learner for predicting emergency readmission of heart-disease patients. *Decision Analytics Journal*, 7(February), 100242. <https://doi.org/10.1016/j.dajour.2023.100242>
- Gusnina, M., Wiharto, & Salamah, U. (2022). Student Performance Prediction in Sebelas Maret University Based on the Random Forest Algorithm. *Ingenierie*

Des Systemes d'Information, 27(3), 495–501.
<https://doi.org/10.18280/isi.270317>

Herianto, Kurniawan, B., Hartomi, Z. H., Irawan, Y., & Anam, M. K. (2024). Machine Learning Algorithm Optimization using Stacking Technique for Graduation Prediction. *Journal of Applied Data Sciences*, 5(3), 1272–1285.
<https://doi.org/10.47738/jads.v5i3.316>

Jan Melvin Ayu Soraya Dachi, & Pardomuan Sitompul. (2023). Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit. *Jurnal Riset Rumpun Matematika Dan Ilmu Pengetahuan Alam*, 2(2), 87–103.
<https://doi.org/10.55606/jurrimipa.v2i2.1470>

Karalar, H., Kapucu, C., & Gürüler, H. (2021). Predicting students at risk of academic failure using ensemble model during pandemic in a distance learning system. *International Journal of Educational Technology in Higher Education*, 18(1). <https://doi.org/10.1186/s41239-021-00300-y>

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017-*Decem*(Section 2), 4766–4775.

Mahboob, K., Sarfaraz Abdul Sattar Natha, Syed Saood Zia, Priha Bhatti, Abeer Javed Syed, & Samra Mehmood. (2023). An Ensemble Modeling Approach to Enhance Grade Prediction in Academic Engineering Programming Courses. *VFAST Transactions on Software Engineering*, 11(4), 01–14.
<https://doi.org/10.21015/vtse.v11i4.1641>

- Muhammad Ricky Perdana Putra, & Ema Utami. (2024). Comparative Analysis of Hybrid Model Performance Using Stacking and Blending Techniques for Student Drop Out Prediction In MOOC. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 8(3), 346–354. <https://doi.org/10.29207/resti.v8i3.5760>
- Nachouki, M., & Naaj, M. A. (2022). Predicting Student Performance to Improve Academic Advising Using the Random Forest Algorithm. *International Journal of Distance Education Technologies*, 20(1), 1–17. <https://doi.org/10.4018/IJDET.296702>
- Noviandy, T. R., Zahriah, Z., Yandri, E., Jalil, Z., Yusuf, M., Mohamed Yusof, N. I. S., Lala, A., & Idroes, R. (2024). Machine Learning for Early Detection of Dropout Risks and Academic Excellence: A Stacked Classifier Approach. *Journal of Educational Management and Learning*, 2(1), 28–34. <https://doi.org/10.60084/jeml.v2i1.191>
- Ouatik, F., Erritali, M., Ouatik, F., & Jourhmane, M. (2022). Predicting Student Success Using Big Data and Machine Learning Algorithms. *International Journal of Emerging Technologies in Learning*, 17(12), 236–251. <https://doi.org/10.3991/ijet.v17i12.30259>
- Sahlaoui, H., Alaoui, E. A. A., Nayyar, A., Agoujil, S., & Jaber, M. M. (2021). Predicting and Interpreting Student Performance Using Ensemble Models and Shapley Additive Explanations. *IEEE Access*, 9, 152688–152703. <https://doi.org/10.1109/ACCESS.2021.3124270>
- Sarat Kumar Chettri, HMM. Naleer, CMM. M. (2023). Hybrid Ensemble Stacking Techniques for Coronary Artery Disease Prediction using Machine Learning

Algorithms. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(8), 534–542.

<https://doi.org/10.17762/ijritcc.v11i8.9568>

Sejati, P., Munawar, Pilliang, M., & Akbar, H. (2022). Studi Komparasi Naive Bayes , K-Nearest Neighbor, dan Random Forest Untuk Prediksi Calon Mahasiswa Yang Diterima Atau Comparative Study Of Naive Bayes , K-Nearest Neighbor , And Random Forest For The Prediction Of Prospective Students. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 9(7), 1341–1348. <https://doi.org/10.25126/jtiik.202296737>

Shintabella, R., Edi Widodo, C., & Wibowo, A. (2024). Loss of Life Transformer Prediction Based on Stacking Ensemble Improved by Genetic Algorithm By IISRT. *International Journal of Innovative Science and Research Technology (IJISRT)*, 9(3), 1061–1066. <https://doi.org/10.38124/ijisrt/ijisrt24mar1125>

Sun, Z., Yuan, Y., Xiong, X., Meng, S., Shi, Y., & Chen, A. (2024). Predicting academic achievement from the collaborative influences of executive function, physical fitness, and demographic factors among primary school students in China: ensemble learning methods. *BMC Public Health*, 24(1), 1–13. <https://doi.org/10.1186/s12889-024-17769-7>

Sunarko, B., Hasanah, U., Hidayat, S., Muhammad, N., Ardiansyah, M. I., Ananda, B. P., Hakiki, M. K., & Baroroh, L. T. (2023). Penerapan Stacking Ensemble Learning untuk Klasifikasi Efek Kesehatan Akibat Pencemaran Udara. *Edu Komputika Journal*, 10(1), 55–63.

<https://doi.org/10.15294/edukomputika.v10i1.72080>

- Wiyono, S., & Abidin, T. (2018). Implementation of K-Nearest Neighbour (Knn) Algorithm To Predict Student'S Performance. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 9(2), 873–878.
<https://doi.org/10.24176/simet.v9i2.2424>
- Wiyono, S., & Abidin, T. (2019). Comparative Study of Machine Learning Knn, Svm, and Decision Tree Algorithm To Predict Student'S Performance. *International Journal of Research -GRANTHAALAYAH*, 7(1), 190–196.
<https://doi.org/10.29121/granthaalayah.v7.i1.2019.1048>
- Yadav*, S. S., & Bhole, G. P. (2020). Learning from Imbalanced Data in Classification. *International Journal of Recent Technology and Engineering (IJRTE)*, 8(5), 1907–1016. <https://doi.org/10.35940/ijrte.e6286.018520>
- Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, 9(1). <https://doi.org/10.1186/s40561-022-00192-z>
- Yan, L., & Liu, Y. (2020). An ensemble prediction model for potential student recommendation using machine learning. *Symmetry*, 12(5), 1–17.
<https://doi.org/10.3390/SYM12050728>

LAMPIRAN



FORMULIR VALIDASI DATA

A. Informasi

Informasi Penelitian

Judul Penelitian : Pengembangan Model Prediksi Performa Akademik Calon Mahasiswa Menggunakan Pendekatan Stacking Ensemble Learning
Nama Peneliti : Claudia Swastikawati
Program Studi : Teknik Informatika
Universitas : Universitas Amikom Yogyakarta

Informasi Validator Internal

Nama Penilai : Kris Agung, S.Kom., M.Kom.
Jabatan/Instansi : Kepala Biro Administrasi Akademik Universitas Slamet Riyadi
Tanggal Penilaian : 14 Januari 2025

B. Tujuan Validasi

Form ini digunakan untuk menilai kelayakan dan relevansi fitur (variabel) yang digunakan dalam penelitian "Pengembangan Model Prediksi Performa Akademik Calon Mahasiswa Menggunakan Pendekatan Stacking Ensemble Learning". Proses validasi ini bertujuan untuk memastikan bahwa instrumen penelitian dan dataset yang digunakan:

1. Telah memenuhi aspek validitas isi (content validity) serta kesesuaian konstruksi variabel yang dikelola oleh Biro Administrasi Akademik sebagai unit penyedia data resmi institusi.
2. Telah dilakukan verifikasi dan cross-check terhadap database hasil pelaporan Pangkalan Data Pendidikan Tinggi (PD-Dikti) yang dikelola oleh Biro Administrasi Akademik, guna menjamin keabsahan, konsistensi, dan keakuratan data.
3. Memenuhi prinsip integritas, akurasi, serta kerahasiaan data pribadi mahasiswa, sesuai dengan standar etika penelitian dan kebijakan perlindungan data institusi.
4. Dinyatakan layak digunakan sebagai dasar analisis ilmiah dan publikasi penelitian

C. Sumber dan Ruang Lingkup Data

Data kelulusan diperoleh secara resmi melalui Biro Administrasi Akademik Universitas Slamet Riyadi yang bersumber dari pelaporan Pangkalan Data Pendidikan Tinggi (PD-Dikti - Neo Feeder) dari tahun 2019 sampai dengan tahun 2024.

Data yang diberikan berupa 6 File CSV dengan nama file DAFTAR MAHASISWA LULUS_DO PERTAHUN yang didalamnya tersedia data NIM, Nama, Program Studi, Jenis Kelamin, Status Lulus/Mutasi/Dikeluarkan, Tanggal Keluar, dan Nomor Ijazah.

Seluruh data telah melalui proses verifikasi bagian akademik sebelum pelaporan ke PD-Dikti.

D. Pernyataan Validasi Data

Dengan ini kami menyatakan bahwa:

1. Data yang diberikan kepada peneliti merupakan data resmi akademik yang telah dilaporkan pada sistem PD-Dikti.
2. Data telah memenuhi prinsip keabsahan, integritas, dan akurasi, serta sesuai dengan catatan akademik institusi.
3. Proses pengambilan data dilakukan sesuai dengan ketentuan perlindungan data pribadi mahasiswa dan penggunaan data hanya untuk kepentingan penelitian akademik.
4. Data dinyatakan layak digunakan sebagai bahan analisis ilmiah dan publikasi penelitian.

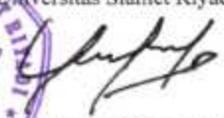
F. Saran dan Masukan Umum

E. Pernyataan Pengesahan

Dengan ini saya menyatakan bahwa data yang diberikan kepada peneliti telah tervalidasi dan dapat digunakan sebagai bahan penelitian.



Kepala Biro Administrasi Akademik
Universitas Slamet Riyadi


Kms Agung, S.Kom., M.Kom
NIPY. 0213.0294

FORMULIR UJI VALIDASI DATASET

A. Informasi

Informasi Penelitian

Judul Penelitian : Pengembangan Model Prediksi Performa Akademik Calon Mahasiswa Menggunakan Pendekatan Stacking Ensemble Learning
Nama Peneliti : Claudia Swastikawati
Program : Magister Informatika - Pendidikan Jarak Jauh
Universitas : Universitas Amikom Yogyakarta

Informasi Validator Teknis

Nama Penilai : Moenawar Kholil, S.Kom., M.Kom.
Jabatan/Instansi : Kepala UPT Teknologi Informasi, Komunikasi, dan Komputer (UPT TIKK) Universitas Slamet Riyadi
Tanggal Penilaian : 14 Januari 2025
Tempat Penilaian : Ruang Ka. UPT TIKK UNISRI

B. Tujuan Validasi

Form ini digunakan untuk menilai kelayakan dan relevansi fitur (variabel) yang digunakan dalam penelitian "Pengembangan Model Prediksi Performa Akademik Calon Mahasiswa Menggunakan Pendekatan *Stacking Ensemble Learning*". Bahwa proses validasi memastukan :

1. Telah dilakukan cross-check terhadap database sistem akademik yang dikelola oleh Unit Pelaksana Teknis Teknologi Informasi Komunikasi dan Komputer (UPT TIKK).
2. Memenuhi prinsip integritas, akurasi, dan kerahasiaan data privat mahasiswa.
3. Layak digunakan untuk keperluan analisis ilmiah dan publikasi penelitian.

C. Deskripsi Dataset yang Divalidasi

No	Nama File	Sumber Data	Keterangan
1	Data_Pendaftaran.csv	UPT TIKK / Sistem PMB (simaru.unisri.ac.id)	Data identitas dan pendaftaran calon mahasiswa di tahun 2019
2	Data_Akademik.csv	UPT TIKK / Sistem Akademik (siakat.unisri.ac.id)	Data mahasiswa angkatan 2019
3	Data_Kelulusan.csv	Biro Akademik - Sistem pelaporan PD-Dikti (NEO FEEDER)	Data pelaporan PDDikti dari tahun 2019 sampai dengan tahun 2024

D. Penilaian

Daftar Variabel/Fitur yang Dinilai

No	Nama Variabel / Fitur	Definisi (1-4)	Kelayakan (1-4)	Catatan
1	Provinsi asal pendaftar	4	3	Lebih baik ditambah kualitas sekolah asal /indeks wilayah (blm ada di DB saat ini)
2	Agama	4	3	Data demografis pendukung
3	Jenis kelamin	4	3	Capaian akademik berbasis gender
4	Periode pendaftaran	4	3	Proses seleksi dan kesiapan awal studi
5	Gelombang pendaftaran	4	3	Potensi perbedaan karakteristik akademik antar jalur
6	Jurusan asal SLTA	4	4	Relevan berkaitan kesiapan akademik sesuai bidang studi.
7	Nilai rata-rata Rapot	4	4	Relevan indikator akademik
8	Skor seleksi CBT	4	4	Ukuran kemampuan akademik terstandar
9	Pekerjaan ayah	4	3	Data pendukung ekonomi keluarga.
10	Pekerjaan ibu	4	3	Data pendukung latarbelakang ekonomi
11	Penghasilan ayah	4	3	Data pendukung finansial keluarga
12	Penghasilan ibu	4	3	Data pendukung finansial keluarga
13	Jumlah anak kandung	4	4	Relevan indikator beban ekonomi keluarga
14	Kode program studi	4	3	Administratif identitas klasifikasi data
15	Status pendaftar (baru/pindahan)	4	4	Relevan data awal yang memengaruhi adaptasi studi.

Skala Penilaian

- 1 = Tidak relevan / Tidak jelas / Tidak layak digunakan
2 = Kurang relevan / Kurang jelas / Kurang layak
3 = Cukup relevan / Cukup jelas / Layak dengan perbaikan
4 = Sangat relevan / Jelas / Sangat layak digunakan

Aspek Validasi

No	Aspek Penilaian	Kriteria Penilaian	Skor (1-4)	Catatan Validator
1	Kesesuaian sumber data dengan basis data resmi	Data berasal dari sistem akademik resmi dan sah digunakan untuk penelitian	4	Sesuai sudah dilakukan pengecekan
2	Kelengkapan atribut data	Data mencakup variabel penting yang dibutuhkan penelitian	3	Belum sepenuhnya menggunakan variable yang ada di database

3	Akurasi dan konsistensi data antarfile	Tidak ada duplikasi atau ketidaksesuaian antar tabel data	4	Sesuai sudah dilakukan pengecekan
4	Integritas dan validitas isi data	Nilai atau entri data sesuai dengan kondisi nyata di database	4	Sesuai sudah dilakukan pengecekan
5	Keamanan dan privasi data	Data bersifat anonim dan tidak menampilkan identitas pribadi secara langsung	4	Data sudah dianonimkan. Tidak ada variabel yang mengandung informasi identitas pribadi.
6	Kelayakan dataset untuk analisis statistic	Kesesuaian proses <i>encoding</i> modeling dan tahap <i>preprocessing</i> untuk Dataset siap digunakan	3	Perlu ditinjau lebih lanjut untuk proses encoding yang dilakukan

Skala Penilaian

- 1 = Tidak relevan / Tidak jelas / Tidak layak digunakan
- 2 = Kurang relevan / Kurang jelas / Kurang layak
- 3 = Cukup relevan / Cukup jelas / Layak dengan perbaikan
- 4 = Sangat relevan / Jelas / Sangat layak digunakan

E. Kesimpulan Validator

- Dataset layak digunakan untuk keperluan penelitian.
- Dataset perlu revisi sebelum dapat digunakan.
- Dataset tidak layak digunakan karena belum memenuhi standar validasi.

F. Saran dan Masukan

- Bisa ditambahkan jumlah data untuk memperkaya dataset.
- Untuk merepresentasikan kestabilan finansial dan motivasi belajar mahasiswa, dapat digunakan tambahan variabel status beasiswa dan jalur seleksi masuk yang secara representatif mencerminkan kualitas proses seleksi serta karakteristik input dan saat ini telah tersedia pada basis data.
- Pengembangan form pendaftaran di masa mendatang seperti data riwayat organisasi, dll

Kepala UPT TIKK
Universitas Slamet Riyadi



M. Mochawar Kholil, S.Kom., M.Kom
NIPY. 0122.0429