Logo

Description automatically generated

**LAPORAN FINAL PROJECT**

**PENERAPAN MODEL MACHINE LEARNING DENGAN FEATURE ENGINEERING, SMOTE, DAN TUNING UNTUK PREDIKSI KELULUSAN SNBP**

**MUHAMMAD SURYA YASAR**

NRP 5025231114

**ARKANANTA MASARIEF**

NRP 5025231115

**RAFI FAHEEM AZIZ**

NRP 5025231116

Dosen Pengampu

**Dini Adni Navastara, S.Kom., M.Sc.**

NIDN NIDN-0017108506

**Program Studi Teknik Informatika**

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

Tahun 2025

Icon

Description automatically generated

**LAPORAN FINAL PROJECT**

**PENERAPAN MODEL MACHINE LEARNING DENGAN FEATURE ENGINEERING, SMOTE, DAN TUNING UNTUK PREDIKSI KELULUSAN SNBP**

**MUHAMMAD SURYA YASAR**

NRP 5025231114

**ARKANANTA MASARIEF**

NRP 5025231115

**RAFI FAHEEM AZIZ**

NRP 5025231116

Dosen Pengampu

**Dini Adni Navastara, S.Kom., M.Sc.**

NIDN NIDN-0017108506

Program Studi Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

Tahun 2025

ABSTRAK

**PENERAPAN MODEL MACHINE LEARNING DENGAN FEATURE ENGINEERING, SMOTE, DAN TUNING UNTUK PREDIKSI KELULUSAN SNBP**

|  |  |  |
| --- | --- | --- |
| **Nama Mahasiswa / NRP** | **:** | **Muhammad Surya Yasar / 5025231114** **Arkananta Masarief / 5025231115** **Rafi Faheem Aziz / 502523116** |
| **Departemen** | **:** | **Teknik Informatika FTEIC - ITS** |
| **Dosen Pembimbing** | **:** | **Dini Adni Navastara, S.Kom., M.Sc.** |

**Abstrak**

Proses seleksi masuk perguruan tinggi jalur SNBP (Seleksi Nasional Berbasis Prestasi) menekankan pada rekam jejak akademik dan non-akademik siswa, dengan tingkat persaingan yang tinggi dan faktor penentu yang kompleks. Memprediksi status kelulusan SNBP merupakan tantangan dalam analisis data karena sifat data yang tabular, keberadaan nilai yang hilang, serta ketidakseimbangan jumlah siswa yang lulus dan tidak lulus. Penelitian ini bertujuan untuk membangun model klasifikasi yang mampu memprediksi status kelulusan SNBP berdasarkan data historis pendaftar, dengan fokus pada penanganan data yang tidak seimbang. Data yang digunakan meliputi informasi nilai mata pelajaran, prestasi, asal sekolah, usia, dan jenis kelamin.

Penelitian ini menerapkan pendekatan empat skenario sistematis: (1) pembuatan model *baseline* untuk menetapkan tolok ukur, (2) *feature engineering* untuk menciptakan variabel informatif seperti indeks\_prestasi, (3) perbandingan teknik *resampling* untuk menangani data tidak seimbang, dan (4) *hyperparameter tuning* serta *ensemble learning* untuk optimisasi akhir. Beberapa algoritma klasifikasi seperti Random Forest, LightGBM, dan XGBoost dievaluasi. Untuk mengatasi ketidakseimbangan kelas, teknik *Synthetic Minority Over-sampling Technique* (SMOTE) dan *Random Undersampling* diterapkan dan dibandingkan secara langsung.

Hasil penelitian menunjukkan bahwa *feature engineering* saja tidak cukup untuk mengatasi bias pada data yang sangat tidak seimbang. Teknik *oversampling* dengan SMOTE terbukti jauh lebih efektif daripada *undersampling* dalam meningkatkan kemampuan model untuk mengenali kelas minoritas. Kesimpulan utama dari penelitian ini adalah bahwa model **LightGBM**, setelah melalui proses *hyperparameter tuning* menggunakan Optuna, berhasil memberikan performa terbaik dengan mencapai **F1-score sebesar 0.4512** untuk kelas 'Lulus' pada data uji, menunjukkan keseimbangan terbaik antara presisi dan recall dalam memprediksi kelulusan.

**Kata kunci: *Machine Learning, Classification Model, Supervised Learning, Feature Engineering, Oversampling, Undersampling, Hyperparameter Tuning, Ensemble Learning***

ABSTRACT

**APPLICATION OF MACHINE LEARNING MODEL WITH FEATURE ENGINEERING, SMOTE, AND HYPERPARAMETER TUNING FOR SNBP ADMISSION PREDICTION**

|  |  |  |
| --- | --- | --- |
| **Student Name / NRP** | **:** | **Muhammad Surya Yasar / 5025231114** **Arkananta Masarief / 5025231115** **Rafi Faheem Aziz / 502523116** |
| **Department** | **:** | **Teknik Informatika FTEIC - ITS** |
| **Advisor** | **:** | **Dini Adni Navastara, S.Kom., M.Sc.** |

**Abstract**

The university admission process through the SNBP (National Selection Based on Achievement) track emphasizes students' academic and non-academic track records, featuring a high level of competition and complex determining factors. Predicting SNBP admission status presents a challenge in data analysis due to the tabular nature of the data, the presence of missing values, and the class imbalance between admitted and rejected students. This research aims to build a classification model capable of predicting SNBP admission status based on historical applicant data, with a focus on handling imbalanced data. The data used includes information on subject grades, achievements, school origin, age, and gender.

This study employs a systematic four-scenario approach: (1) building a baseline model to establish a benchmark, (2) feature engineering to create informative variables such as an achievement index (indeks\_prestasi), (3) comparing resampling techniques to handle imbalanced data, and (4) hyperparameter tuning and ensemble learning for final optimization. Several classification algorithms such as Random Forest, LightGBM, and XGBoost are evaluated. To address the class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) and Random Undersampling are applied and directly compared.

The results indicate that feature engineering alone is insufficient to overcome the bias in a highly imbalanced dataset. The oversampling technique with SMOTE proved to be far more effective than undersampling in enhancing the model's ability to recognize the minority class. The main conclusion of this study is that the LightGBM model, after undergoing a hyperparameter tuning process using Optuna, delivered the best performance by achieving an F1-score of 0.4512 for the 'Admitted' class on the test data, demonstrating the best balance between precision and recall in predicting admission status.

**Keywords: *Machine Learning, Classification Model, Supervised Learning, Feature Engineering, Oversampling, Undersampling, Hyperparameter Tuning, Ensemble Learning***

DAFTAR ISI

[ABSTRAK i](#_Toc1344489746)

[ABSTRACT iii](#_Toc1867001734)

[DAFTAR ISI iv](#_Toc1739546796)

[DAFTAR GAMBAR v](#_Toc83447404)

[DAFTAR TABEL vi](#_Toc233369142)

[DAFTAR SIMBOL vii](#_Toc1636346076)

[BAB 1 PENDAHULUAN 8](#_Toc1492339100)

[1.1 Latar Belakang 8](#_Toc804381458)

[1.2 Rumusan Masalah 9](#_Toc251224141)

[1.3 Tujuan dan Manfaat 10](#_Toc1685191840)

[BAB 2 TINJAUAN PUSTAKA 11](#_Toc1349434656)

[2.1 Penelitian Terkait 11](#_Toc45125533)

[2.2 Dasar Teori 14](#_Toc102978933)

[BAB 3 METODOLOGI 20](#_Toc631413882)

[3.1 Perancangan Sistem 20](#_Toc1860306231)

[3.2 Implementasi Sistem 34](#_Toc1853475914)

[BAB 4 Hasil dan Pembahasan 38](#_Toc1744420120)

[4.1 Hasil penelitian 38](#_Toc1099428235)

[4.2 Pembahasan 41](#_Toc1867075518)

[BAB 5 KESIMPULAN 43](#_Toc521547625)

[DAFTAR PUSTAKA 44](#_Toc1385962751)

[LAMPIRAN 46](#_Toc1000540218)

DAFTAR GAMBAR

Gambar 2.1. Ilustrasi algoritma Random Forest

Gambar 2.2. Ilustrasi algoritma SVM

Gambar 2.3. Ilustrasi algoritma Logistic Regression

Gambar 2.4. Ilustrasi algoritma LightGBM

Gambar 2.5. Ilustrasi algoritma XGBoost

Gambar 2.6. Ilustrasi algoritma CatBoost

Gambar 3.1 Distribusi Status Kelulusan

Gambar 3.2 Heatmap Korelasi

Gambar 3.3 Missing Values

Gambar 3.4 Plot Jenjang 1 Prestasi dan Status Kelulusan

Gambar 3.5 Plot Jenjang 2 Prestasi dan Status Kelulusan

Gambar 3.6 Plot Jenjang 3 Prestasi dan Status Kelulusan

Gambar 3.7 Plot Proporsi Kelulusan Berdasarkan Jenis Kelamin

Gambar 3.8 Histogram Tahun

Gambar 3.9 Histogram Distribusi Usia

Gambar 3.10 Persentase Kelulusan SNBP per Provinsi

Gambar 3.11 Flowchart Skenario Baseline

Gambar 3.12 Flowchart Skenario Feature Engineering

Gambar 3.13 Flowchart Skenario Imbalanced Data

Gambar 3.14 Flowchart Skenario Hyperparameter Tuning

DAFTAR TABEL

Tabel 2.1.1 Penelitian terkait

Tabel 4.1.1 Hasil Skenario 1: Model Baseline

Tabel 4.1.2 Hasil Skenario 2: Feature Engineering dan Preprocessing

Tabel 4.1.3 Model dengan SMOTE

Tabel 4.1.4 Model dengan Undersampling

Tabel 4.1.5 Model dengan Hyperparameter Tuning

Tabel 4.1.6 Model Ensemble Learning

# PENDAHULUAN

## Latar Belakang

Pendidikan tinggi memegang peranan krusial dalam pembangunan suatu bangsa. Kualitas sumber daya manusia (SDM) yang unggul, inovatif, dan berdaya saing tinggi merupakan produk dari sistem pendidikan tinggi yang efektif dan berkualitas. Di Indonesia, upaya untuk meningkatkan kualitas SDM secara berkelanjutan menjadikan proses seleksi masuk perguruan tinggi negeri (PTN) sebagai salah satu momen paling strategis dan kompetitif bagi para siswa sekolah menengah atas (SMA) dan sederajat. Setiap tahunnya, ratusan ribu hingga jutaan siswa bersaing untuk memperebutkan kursi di berbagai program studi favorit di PTN terkemuka di seluruh penjuru negeri.

Pemerintah Indonesia, melalui Balai Pengelolaan Pengujian Pendidikan (BP3), menyelenggarakan Seleksi Nasional Penerimaan Mahasiswa Baru (SNPMB) sebagai mekanisme utama untuk menyaring calon mahasiswa baru. Sistem SNPMB ini dirancang untuk menjamin proses seleksi yang kredibel, adil, transparan, fleksibel, efisien, dan akuntabel. Terdapat dua jalur utama dalam seleksi nasional, yaitu Seleksi Nasional Berdasarkan Prestasi (SNBP) dan Seleksi Nasional Berdasarkan Tes (SNBT).

Jalur SNBP menjadi gerbang pertama dan seringkali yang paling didambakan oleh para siswa. Berbeda dengan SNBT yang berfokus pada hasil tes skolastik, SNBP merupakan jalur seleksi yang didasarkan pada prestasi akademik dan non-akademik siswa selama menempuh pendidikan di bangku SMA. Komponen penilaian utama dalam SNBP mencakup nilai rapor dari semester satu hingga lima, serta prestasi lainnya yang relevan, seperti sertifikat kejuaraan. Selain prestasi individu siswa, faktor lain seperti akreditasi sekolah dan indeks sekolah juga turut menjadi pertimbangan penting dalam proses seleksi (SNPMB, 2025).

Meskipun mekanisme ini bertujuan untuk memberikan penghargaan kepada siswa-siswa berprestasi secara konsisten, proses SNBP seringkali dirasa kurang transparan dan adil oleh siswa dan pihak sekolah. Bobot penilaian untuk setiap komponen seperti nilai rata-rata rapor, nilai mata pelajaran pendukung, jumlah dan tingkatan sertifikat, serta pengaruh akreditasi sekolah tidak pernah dipublikasi secara eksplisit oleh panitia SNPMB. Ketidakpastian ini menimbulkan tantangan signifikan. Siswa mengalami kesulitan dalam mengukur peluang kelulusan mereka secara objektif dan seringkali membuat pilihan program studi yang kurang aman. Di sisi lain, guru-guru di sekolah juga menghadapi kendala dalam memberikan arahan yang tepat sekaligus berbasis data kepada para siswanya.

Di tengah tantangan ini, kemajuan pesat di bidang teknologi informasi, khususnya dalam kecerdasan buatan (*artificial intelligence*), menawarkan solusi baru untuk memecahkan masalah-masalah yang melibatkan analisis data dalam skala besar. Salah satu cabang dari kecerdasan buatan yang relevan untuk masalah ini adalah pembelajaran mesin (*machine learning*), yaitu sebuah metode yang menginginkan sistem komputer untuk “belajar” dari data yang ada guna mengenali pola dan membuat prediksi atau keputusan tanpa diprogram secara eksplisit.

Model klasifikasi adalah salah satu teknik fundamental dalam *machine learning* yang bertujuan untuk mengkategorikan sebuah data ke dalam kelas-kelas yang telah ditentukan sebelumnya. Dalam konteks seleksi PTN, data historis kelulusan siswa pada jalur SNBP dapat dimanfaatkan untuk membangun model klasifikasi. Model ini dapat mempelajari pola hubungan antara berbagai variabel input dengan hasil akhir. Berdasarkan dataset yang ada, variabel input dapat mencakup nilai rata-rata rapor, nilai rata-rata mata pelajaran pendukung program studi, jumlah sertifikat yang dimiliki, dan variabel lainnya, sedangkan variabel targetnya adalah status kelulusan siswa.

Pemanfaatan kecerdasan buatan untuk memprediksi performa akademik siswa telah menjadi pendekatan penting yang diterapkan di berbagai institusi pendidikan modern. Hal ini sejalan dengan penelitian yang dilakukan oleh Kaensar dan Wongnin (2023), yang menyampaikan bahwa prediksi kesuksesan mahasiswa sedini mungkin (*early prediction*) dengan menggunakan *machine learning* adalah salah satu pendekatan paling krusial untuk mengatasi masalah kegagalan mahasiiswa. Dalam studi mereka di Thailand, data skor admisi dari ribuan mahasiswa berhasil digunakan untuk membangun dan membandingkan berbagai model *machine learning* guna mengidentifikasi calon mahasiswa yang berpotensi sukses.

Pendekatan yang diuraikan oleh Kaensar dan Wongnin (2023) sangat relevan untuk diterapkan dalam konteks SNBP di Indonnesia. Dengan memanfaatkan data historis kelulusan, kita dapat menerapkan teknik model klasifikasi. Model klasifikasi dalam konteks SNBP dapat digunakan untuk mempelajari data historis guna mengenali pola kompleks dan mengkategorikan data baru ke dalam kelas-kelas yang telah ditentukan, yaitu “Lulus” atau “Tidak Lulus”.

Oleh karena itu, penelitian ini diusulkan untuk merancang dan membangun sebuah model klasifikasi yang mampu memprediksi peluang kelulusan seorang siswa pada jalur SNBP. Dengan memanfaatkan data kelulusan yang mencakup variabel seperti nilai rata-rata rapor, nilai mata pelajaran pendukung, dan sertifikat kompetisi, penelitian ini akan mengeksplorasi berbagai algoritma klasifikasi untuk menemukan model dengan tingkat akurasi dan keandalan tertinggi.

Pengembangan model prediktif ini memiliki urgensi dan potensi manfaat yang besar. Sejalan dengan tujuan riset pendidikan modern untuk menekan angka kegagalan, model ini dapat berfungsi sebagai alat bantu bagi siswa untuk melakukan evaluasi diri dan memilih program studi secara lebih strategis. Bagi pihak sekolah, model ini dapat menjadi dasar untuk memberikan rekomendasi yang lebih objektif dan berbasis data. Lebih jauh lagi, hasil dari penelitian ini dapat memberikan wawasan berharga bagi para pemangku kepentingkan mengenai faktor-faktor penentu keberhasilan siswa di jalur SNBP, sebuah langkah menuju sistem seleksi yang lebih transparan dan dapat dipahami.

Dengan demikian, penelitian mengenai “info judul” tidak hanya menjadi sebuah eksplorasi teknis dalam penerapan kecerdasan buatan, tetapi juga merupakan sebuah upaya strategis yang didukung oleh tren riset global untuk menghadirkan transparansi dan membantu para calon penerus bangsa dalam meraih cita-cita pendidikan mereka.

## Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan di atas, kami memilih beberapa rumusan masalah, diantaranya yaitu:

1. Apa model dan skema klasifikasi yang paling akurat dalam memprediksi status kelulusan SNBP?
2. Apa pengaruh fitur-fitur yang ada terhadap status kelulusan siswa? Bagaimana fitur tersebut mempengaruhi status kelulusan siswa?
3. Bagaimana faktor yang ada pada skema menyebabkan perubahan tingkat keakurasian model?

## Tujuan dan Manfaat

Tujuan dari penelitian adalah:

1. Menganalisis pengaruh dari berbagai fitur (nilai siswa, usia, tahun mendaftar, prestasi, akreditasi sekolah, provinsi) terhadap status kelulusan siswa pada SNBP.
2. Membangun dan mengevaluasi model klasifikasi untuk memprediksi status kelulusan siswa (lulus atau tidak lulus) pada SNBP dengan menggunakan dataset yang ada.

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi Calon Mahasiswa**:** Memberikan gambaran mengenai faktor-faktor yang dapat mempengaruhi kelulusan SNBP, sehingga dapat digunakan sebagai referensi dalam menentukan strategi untuk pendaftaran SNBP.
2. Bagi Pihak Sekolah : Menyediakan informasi yang dapat digunakan untuk mengidentifikasi potensi siswa dan memberikan bimbingan yang lebih terarah dalam menghadapi seleksi SNBP.

# TINJAUAN PUSTAKA

## Penelitian Terkait

|  |  |  |  |
| --- | --- | --- | --- |
| Judul | Dataset | Metode | Analisis Gap |
| **Binary Classification with Supervised Machine Learning: A Comparative Analysis** | 11,969 data rumah tangga di Maroko tahun 2014 dengan 784 variabel | - **Model**: *Support Vector Machine, Decision Tree, Logistic Regression*  - **Evaluasi**: Presisi, recall, F1-score, akurasi | Berfokus pada domain sosio-ekonomi. **Gap**: Penerapan pada konteks seleksi mahasiswa (SNBP) dengan rekayasa fitur dan fokus penanganan data tidak seimbang. |
| Balancing the Scales: A Comprehensive Study on Tackling Class Imbalance in Binary Classification | 30 dataset bervariasi dengan ukuran 500-20,000 sampel dan persentase kelas minoritas 1%-15% | - **Model**: 15 model berbeda termasuk model berbasis *tree*, *boosting*, dan *neural network*  -**Teknik**: Membandingkan SMOTE, *Class Weights Tuning*, dan *Decision Threshold Calibration* | Merupakan studi umum teknik *imbalance handling*. **Gap**: Penerapan dan perbandingan spesifik (SMOTE vs. Undersampling) pada domain unik data SNBP, yang dilanjutkan dengan *hyperparameter tuning*. |
| **Predicting new student performances and identifying important attributes of admission data using machine learning techniques with hyperparameter tuning** | 5,919 data mahasiswa dari 10 program studi di universitas Thailand (2011-2021) | - **Model**: Membandingkan 6 model *machine learning* (termasuk *Random Forest* dan *XGBoost*)  - **Teknik**: Penyetelan *hyperparameter* untuk optimasi | Konteks seleksi di universitas Thailand. **Gap**: Penerapan pada konteks SNBP Indonesia, pembuatan fitur unik indeks\_prestasi, dan perbandingan eksplisit teknik *resampling*. |
| A comparative study on predicting student’s performance using machine learning algorithms | 258 data mahasiswa dengan 22 atribut demografis dan akademik. | - **Model**: Membandingkan *Decision Tree, Random Forest, SVM,* dan *Naive Bayes*.  - **Hasil**: *Random Forest* menjadi yang terbaik dengan akurasi 95.05%. | Konteks dataset yang berbeda dan lebih kecil. **Gap**: Fokus lebih dalam pada rekayasa fitur spesifik SNBP dan analisis perbandingan penanganan data tidak seimbang. |
| Predicting Student’s Academic Performance Using a Hybrid Feature Selection Method and Machine Learning Classifiers | Data mahasiswa dengan 16 fitur awal yang direduksi menjadi 6 fitur. | - **Teknik**: Seleksi fitur hibrida (*Chi-Square* & *ReliefF*).  - **Model**: *Decision Tree, Naive Bayes, Random Forest*.  - **Hasil**: *Random Forest* dengan seleksi fitur hibrida mencapai akurasi 93.3%. | Berfokus pada **seleksi fitur** (memilih fitur). **Gap**: Penelitian kami berfokus pada **rekayasa fitur** (membuat fitur baru indeks\_prestasi) dan membandingkan teknik *resampling*. |
| LightGBM Algorithm and Hyperparameter Tuning in Intrusion Detection System | Dataset CIC-IDS-2017 untuk deteksi intrusi;  >2.8 juta data, *imbalanced*. | - **Model**: LightGBM. - **Teknik**: *Hyperparameter Tuning* dengan *Grid Search CV*. | Fokus pada domain keamanan siber.  **Gap**: Domain berbeda (SNBP), metode tuning lebih modern (Optuna), & cakupan perbandingan model/resampling yang lebih luas. |

Tabel 2.1.1. Penelitian terkait

### Penelitan oleh Yassine El aachab (2023)

Penelitian oleh Yassine El aachab pada tahun 2023 dengan judul Binary Classification with Supervised Machine Learning: A Comparative Analysis. Penelitian ini membahas tentang penggunaan teknik Support Vector Machine, Decision Tree dan Logistic Regression untuk melakukan prediksi terhadap survey standar hidup rumah tangga di Morroco pada tahun 2014 untuk mengklasifikasikan rumah tangga miskin dan tidak miskin. Terdapat total 11969 data valid dan 784 variabel total. Metode yang digunakan adalah preprocessing, pembersihan data, dan menggunakan software R untuk melakukan prediksi matematika dan statistika. Setelah itu, dilakukan pembandingan antara tiga model (Support Vector Machine, Decision Tree, dan Logistic Regression) untuk melihat performa masing-masing model berdasarkan beberapa metrix penilaian, yaitu presisi, recall, accuracy, F1-score, specificity, dan Cohen’s Kappa. Hasilnya, model Decision Tree memiliki hasil terbaik diantara ketiga model, dengan akurasi 99.61%, presisi 99.86%, serta F1 score 99.79%. Persentase yang tinggi pada presisi dan F1 score mengindikasikan bahwa model yang diimplementasikan sudah bekerja sangat baik dalam mengklasifikasikan rumah tangga miskin dan tidak miskin di Morroco pada tahun 2014.

### Penelitian oleh Mohamed Abdelhamid (2024)

Penelitian oleh Mohamed Abdelhamid pada tahun 2024 dengan judul Balancing the Scales: A Comprehensive Study on Tackling Class Imbalance in Binary Classification. Dalam penelitian ini, terdapat pembandingan beberapa metode mengatasi class imbalance, yaitu Synthetic Minority Over-sampling Technique (SMOTE), Class Weights tuning, and Decision Threshold Calibration. Terdapat 30 dataset dengan domain yang bervariasi, dengan ukuran sampel 500-20,000 dan *rare classs percentage* berkisar antara 1% hingga 15%. Model yang digunakan dalam penelitian ini termasuk model berbasis tree, algoritma boosting, dan neural network, sehingga total model yang digunakan adalah 15. Dari hasil penelitian dengan berbagai metrik seperti F1 score, F2 score, MCC, recall, presisi, PR-AUC, AUC, akurasi, log loss, dan brier score, metode yang paling optimal dalam mengatasi class imabalance adalah Decision Threshold. Metode ini dibuktikan paling efektif di 40% dataset dan menunjukkan performa robust diantara model yang berbeda-beda. Walaupun pada beberapa dataset tidak menunjukkan metode ini paling optimal, tapi performnya tetap stabil dan hasilnya mendekati solusi optimal. Kinerja pada F2 score dan recall tinggi yang artinya efektif dalam mengidentifikasi kelas minoritas serta MCC score terbaik mengindikasikan keseimbangan yang baik antara true/false positive dan negative. Namun, solusi ini tidak bisa diterapkan pada semua dataset dan terkadang SMOTE dan Class Weights tuning bisa menjadi solusi terbaik pada beberapa dataset. Dalam analisis *post-hoc,* tidak ditemukan perbedaan signifikan antara Decision Threshold dan SMOTE pada tingkat alpha tertentu, sehingga SMOTE dan Decision Threshold bisa bekerja dengan performa yang setara.

### Penelitian oleh Chayaporn Kaensar & Worayoot Wongnin (2023)

Penelitian oleh Chayaporn Kaensar dan Worayoot Wongnin pada tahun 2023 dengan judul "Predicting new student performances and identifying important attributes of admission data using machine learning techniques with hyperparameter tuning". Penelitian ini dilatarbelakangi oleh tingginya angka kegagalan dan drop out mahasiswa yang menjadi tantangan bagi banyak universitas secara global. Tujuannya adalah untuk melakukan prediksi kesuksesan mahasiswa sedini mungkin (early prediction) dengan menganalisis data seleksi masuk di Fakultas Sains, Universitas Ubon Ratchathani, Thailand. Data yang digunakan mencakup skor admisi dari total 5.919 mahasiswa dari 10 program studi berbeda selama periode 2011-2021. Metode yang digunakan adalah membangun dan membandingkan enam model machine learning (Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Decision Tree, Random Forest, dan XGBoost) dengan penekanan pada penyetelan hyperparameter untuk mendapatkan performa optimal. Performa model dievaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score. Hasilnya, model XGBoost dengan penyetelan hyperparameter menunjukkan performa terbaik, dengan akurasi 96.54%, presisi 96.55%, recall 96.54%, dan F1-score 96.54%. Performa tinggi ini membuktikan bahwa data admisi sangat efektif untuk memprediksi kinerja mahasiswa baru, yang dapat membantu universitas dalam mengatasi masalah kegagalan akademik sejak dini.

### Penelitian oleh Theresa-Luba A.N. Ha, et al. (2021)

Penelitian oleh Theresa-Luba A.N. Ha, Thi-Minh-Tuyen Nguyen, dan Tuan-Anh T. Nguyen pada tahun 2021 berjudul “A comparative study on predicting student’s performance using machine learning algorithms”. Penelitian ini bertujuan untuk memprediksi kinerja akademik mahasiswa pada akhir semester berdasarkan data demografis dan catatan akademik sebelumnya. Tujuannya adalah untuk mengidentifikasi mahasiswa yang berisiko gagal lebih awal sehingga institusi dapat memberikan intervensi yang tepat. Dataset yang digunakan mencakup data dari 258 mahasiswa dengan 22 atribut, seperti nilai ujian masuk, nilai mata kuliah prasyarat, kehadiran, dan data demografis lainnya. Metode yang digunakan adalah perbandingan performa antara empat algoritma klasifikasi populer: Decision Tree (J48), Random Forest, Support Vector Machine (SVM), dan Naive Bayes. Hasil penelitian menunjukkan bahwa model Random Forest memberikan performa terbaik di antara keempat model tersebut, dengan mencapai akurasi prediksi sebesar 95.05%. Studi ini mengonfirmasi bahwa data historis akademik sangat efektif untuk prediksi dan Random Forest merupakan algoritma yang sangat andal untuk tugas klasifikasi di bidang data pendidikan.

### Penelitian oleh K. Muthumari, et al. (2022)

Penelitian oleh K. Muthumari, P. A. B. Ananth, dan T. M. Kumar pada tahun 2022 dengan judul “Predicting Student’s Academic Performance Using a Hybrid Feature Selection Method and Machine Learning Classifiers”. Latar belakang penelitian ini adalah pentingnya seleksi fitur (feature selection) untuk meningkatkan akurasi model prediksi dengan menghilangkan fitur yang tidak relevan dan redundant. Penelitian ini berfokus pada prediksi kinerja akademik mahasiswa (lulus atau gagal) dengan terlebih dahulu menerapkan metode seleksi fitur hibrida yang menggabungkan teknik Chi-Square dan ReliefF. Dataset yang digunakan terdiri dari 16 fitur awal yang mencakup informasi pribadi dan akademik mahasiswa. Setelah proses seleksi fitur, jumlah fitur direduksi menjadi 6 fitur yang paling berpengaruh. Fitur-fitur terpilih ini kemudian digunakan untuk melatih tiga model klasifikasi: Decision Tree, Naive Bayes, dan Random Forest. Hasilnya menunjukkan bahwa kombinasi metode seleksi fitur hibrida dengan model Random Forest menghasilkan akurasi tertinggi, yaitu 93.3%. Penelitian ini membuktikan bahwa rekayasa dan seleksi fitur yang tepat merupakan langkah krusial sebelum pemodelan untuk mencapai performa prediksi yang optimal.

### Penelitian oleh I Gede Widiase dan I Putu Agus Eka Pratama (2023)

Penelitian oleh I Gede Widiase dan I Putu Agus Eka Pratama dengan judul ”LightGBM Algorithm and Hyperparameter Tuning in Intrusion Detection System”. Penelitian ini berfokus pada optimisasi model LightGBM untuk tugas klasifikasi pada sistem deteksi intrusi (IDS), di mana dataset yang digunakan seringkali bersifat sangat tidak seimbang. Dataset yang digunakan adalah CIC-IDS-2017, sebuah dataset publik berskala besar yang berisi data lalu lintas jaringan dengan lebih dari 2.8 juta catatan dan 78 fitur. Metode utama yang diterapkan adalah *hyperparameter tuning* pada algoritma LightGBM menggunakan teknik *Grid Search CV* untuk menemukan kombinasi parameter terbaik. Hasil penelitian menunjukkan bahwa proses tuning berhasil meningkatkan performa model secara signifikan, di mana model LightGBM yang telah dioptimalkan mencapai akurasi 99.98% dan F1-Score 99.98%. Studi ini membuktikan bahwa penyetelan hiperparameter adalah langkah krusial untuk memaksimalkan potensi algoritma seperti LightGBM, terutama pada dataset yang besar dan tidak seimbang.

## Dasar Teori

### **Dataset**

Dataset SNBP merupakan kumpulan data historis yang digunakan dalam penelitian ini untuk melatih dan menguji model prediksi kelulusan SNBP. Dataset ini terdiri dari beberapa fitur (variabel independen) yang merepresentasikan karakteristik siswa dan satu variabel target (variabel dependen) yang menunjukkan status kelulusan.

* Nilai Siswa: Merepresentasikan capaian akademik siswa dalam bentuk rata-rata nilai rapor. Ini adalah indikator utama performa akademik.
* Usia: Usia siswa pada saat pendaftaran, yang mungkin berkorelasi dengan kematangan atau tahun kelulusan sekolah menengah.
* Tahun Mendaftar: Tahun di mana siswa mendaftar SNBP, yang dapat menunjukkan tren atau perubahan kebijakan seleksi dari waktu ke waktu.
* Prestasi: Mengacu pada prestasi non-akademik siswa, seperti kejuaraan lomba, kegiatan ekstrakurikuler, atau kepemimpinan, yang dapat memberikan poin tambahan dalam seleksi.
* Akreditasi Sekolah: Tingkat akreditasi sekolah asal siswa, yang dapat mencerminkan kualitas pendidikan dan reputasi sekolah.
* Provinsi: Provinsi asal siswa, yang mungkin memiliki pengaruh terkait dengan kuota regional atau sebaran geografis.
* Status Kelulusan: Variabel target yang menunjukkan apakah siswa dinyatakan Lulus atau Tidak Lulus dalam SNBP. Ini adalah label kelas yang akan diprediksi oleh model.

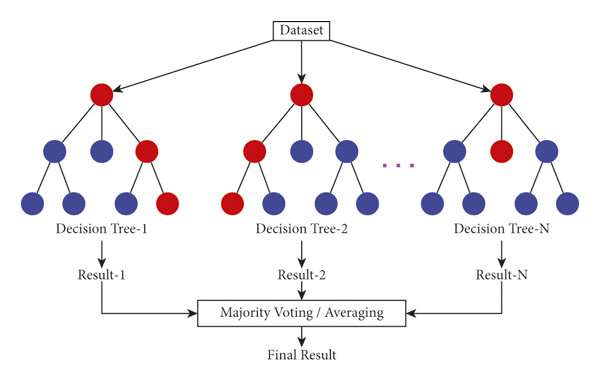
### **Metode**

Dalam penelitian ini, beberapa metode klasifikasi dari bidang *machine learning* akan diimplementasikan untuk memprediksi status kelulusan SNBP. Metode klasifikasi adalah algoritma yang digunakan untuk mengkategorikan data input ke dalam salah satu dari beberapa kelas diskrit.

* Random Forest

Random Forest memiliki algoritma seperti membangun sebuah “hutan” yang terdiri dari pohon-pohon yang tidak saling berkorelasi dan menghasilkan kelas yang menjadi modus dari output masing-masing pohon untuk klasifikasi atau prediksi. Kekuatan utama dari metode ini terletak pada dua sumber random utamanya yang mengurangi varians dan mencegah overfitting. (Breiman, 2001)

Random Forest bekerja dengan membangun sebuah forest atau kumpulan pohon keputusan independen. Lalu, setiap pohon dilatih pada subset data pelatihan yang diambil secara acak. Kemudian, pada setiap node dalam pohon, sejumlah fitur dipilih secara acak untuk dipertimbangkan sebagai pemisah terbaik. Untuk klasifikasi, hasil prediksi dari setiap pohon “divoting” untuk menentukan kelas akhir. Hal ini menjadikan Random Forest sangat akurat dan mampu menangani dataset besar dengan banyak fitur. Namun, Random Forest sulit menjelaskan keputusan model dibandingkan Single Decision Tree dan membutuhkan lebih banyak komputasi.

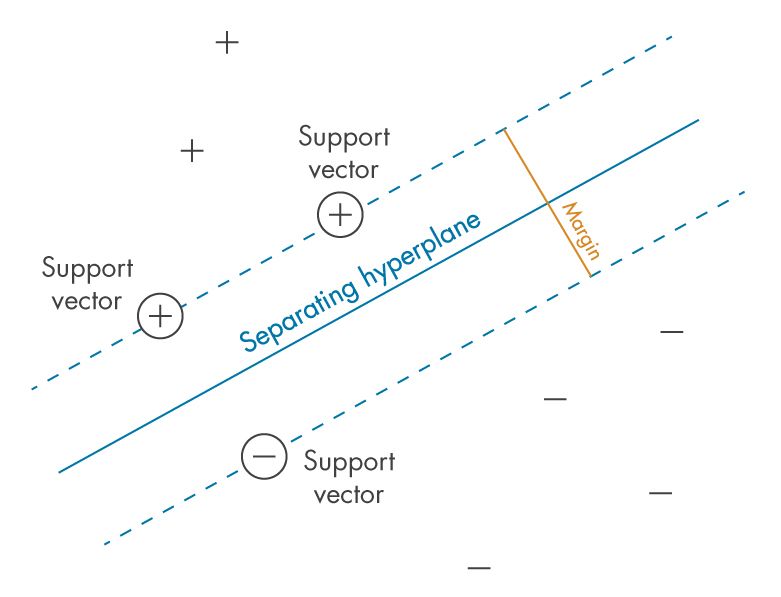


Gambar 2.1. Ilustrasi algoritma *Random Forest*

* Support Vector Machine (SVM)

Support Vector Machine merupakan algoritma supervised learning yang cukup andal. Cara kerja SVM adalah dengan menemukan hyperplane atau bidang pemisah optimal yang memaksimalkan jarak antara dua kelas yang berbeda. Titik-titik data yang berada paling dekat dengan hyperplane dan menentukan posisinya disebut support vector. Keunikan dan keunggulan SVM merupakan pada kemampuannya untuk menangani data yang tidak dapat dipisahkan secara linear melalui kernel trick. Metode ini memetakan data ke high-dimensional feature space yang mana merupakan tempat pemisah linear. (Cortes et al., 1995)

SVM sangatlah cocok untuk digunakan pada dataset yang memiliki jumlah fitur yang banyak sehingga membuatnya unggul dalam klasifikasi biner. Selain itu, SVM juga menunjukkan performa yang baik dalam membuat batas keputusan yang jelas karena cara kerjanya yang membuat hyperplane yang memisahkan dua kelas. Oleh karena itu, jika SVM bisa memisahkan kelas secara jelas, model yang dilatih akan semakin akurat. Namun, SVM memiliki waktu training yang sangat lama dan kurang efisien, oleh sebab itu, SVM tidak cocok untuk dataset yang besar.

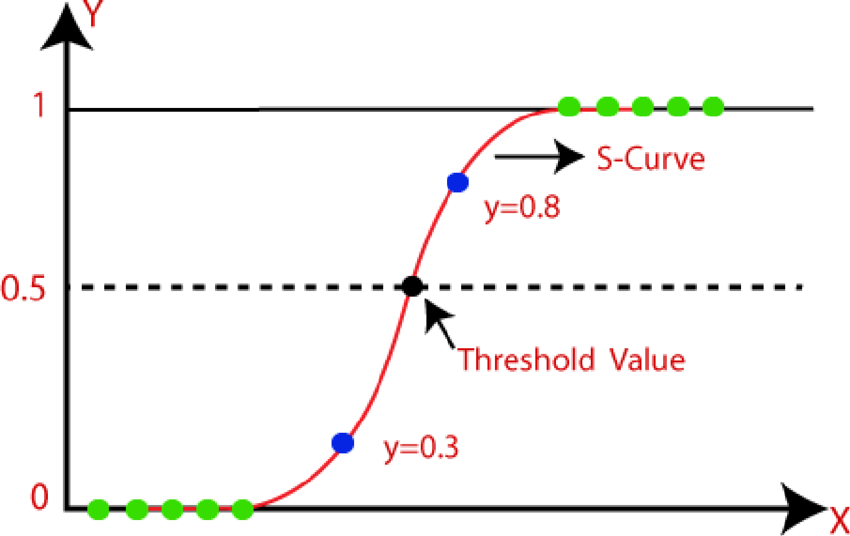


Gambar 2.2. Ilustrasi algoritma SVM

* Logistic Regression

Logistic Regression adalah teknik statistik yang serbaguna untuk memodelkan hubungan antarrangkaian variabel independen dengan sebuah hasil biner. Metode ini sangat berguna karena dapat digunakan untuk memprediksi probabilitas terjadinya suatu peristiwa. Cara kerja model ini adalah dengan transformasi logit, yaitu mengubah output dari kombinasi linear variabel prediktor menjadi probabilitas (0 hingga 1). (Stoltzfus, 2011)

Model ini sangatlah sederhana dan mudah diimplementasikan pada banyak dataset, sehingga cara setiap variabel independen mempengaruhi hasil akhir bisa sangat mudah dipahami. Selain itu, metode ini dikenal cepat dalam proses pelatihannya dan bisa memberikan performa baik pada data yang linear. Namun, model ini tidak mampu mengatasi dataset yang memiliki hubungan non-linear karena model ini mengasumsikan hubungan antara fitur dan hasil bersifat linear. Model ini juga rentan terhadap underfitting dan tidak mampu memahami pola rumit dalam data seperti model-model lainnya.



Gambar 2.3. Ilustrasi algoritma *Logistic Regression*

* LightGBM

LightGBM adalah sebuah Gradient Boosting Decision Tree yang mengimplementasikan Gradient-based One-Side Sampling (GOSS) yaitu mempertahankan data dengan gradien besar, sehingga secara acak mengabaikan data dengan gradien yang kecil. Ini memungkinkan perhitungan gain yang efisien dengan jumlah data lebih sedikit. Selain itu, LightGBM juga mengimplementasikan Exclusive Feature Bundling, yaitu data berdimensi tinggi dan sparsity tinggi, fitur-fitur yang jarang bernilai nol secara bersamaan akan dikelompokkan untuk mengurangi jumlah fitur yang diproses. LightGBM mengutamakan kecepatan dan skalabilitas tanpa mengurangi akurasi, dan terbukti bisa mempercepat proses training hingga 20 kali lebih cepat. (Ke et al., 2017)

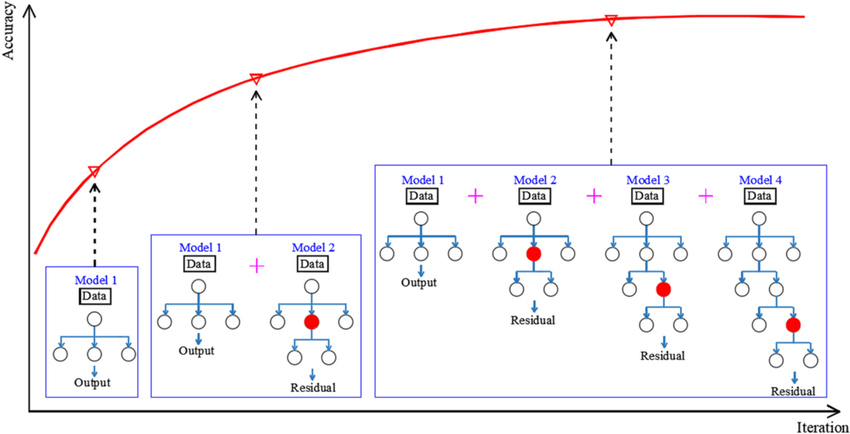
LightGBM terkenal akan kecepatannya dalam proses training, namun tetap mempertahankan ketepatan atau akurasinya. Selain itu, LightGBM juga memiliki efisiensi memori yang baik dan dapat menangani dataset yang besar. Namun, LightGBM rentan terhadap overfitting pada dataset kecil karena cara kerja leaf-wise growth yang agresif dapat membuat model menghafal dataset ketika datasetnya kecil. Selain itu, untuk hasil yang maksimal, LightGBM memerlukan hyperparameter tuning yang tepat untuk dataset kecil.

LightGBM memiliki parameter untuk mengatur beberapa hal yang akan digunakan model untuk melakukan pembelajaran. Hal inilah yang mampu mempengaruhi kemampuan model dalam memprediksi sesuatu. Berikut penjelasan terkait parameter yang ada di LightGBM:

* + num\_leaves: parameter utama untuk mengontrol kompleksitas model, idealnya kurang dari 2^(max\_depth)
  + min\_data\_in\_leaf: parameter untuk mencegah over-fitting. Nilai optimalnya tergantung jumlah sampel training dan num\_leaves. Jika terlalu besar, bisa menyebabkan under-fitting, sehingga ratusan maupun ribuan cukup untuk dataset besar.
  + max\_depth: parameter untuk mengatur kedalaman tree, jika max\_depth di set, maka num\_leaves harus <= 2^max\_depth

Maka, untuk mengatasi overfitting dan meningkatkan akurasi, bisa dengan menggunakan learning\_rate kecil dan iteration besar, menggunakan bagging, limit max\_depth untuk mengatur kedalaman agar tidak overfit, dan menggunakan num\_leaves yang kecil (LightGBM Team, n.d.)

Hyperparameter yang memiliki dampak signifikan dalam mengatasi kelas yang tidak seimbang adalah parameter is\_unbalance. Selain itu, untuk parameter lain seperti max\_depth membuat dampak yang tidak langsung, namun signifikan dalam peningkatan performa. Maka, hyperparameter sendiri memiliki dampak dalam perubahan performa dari LightGBM (Caballero, 2024)



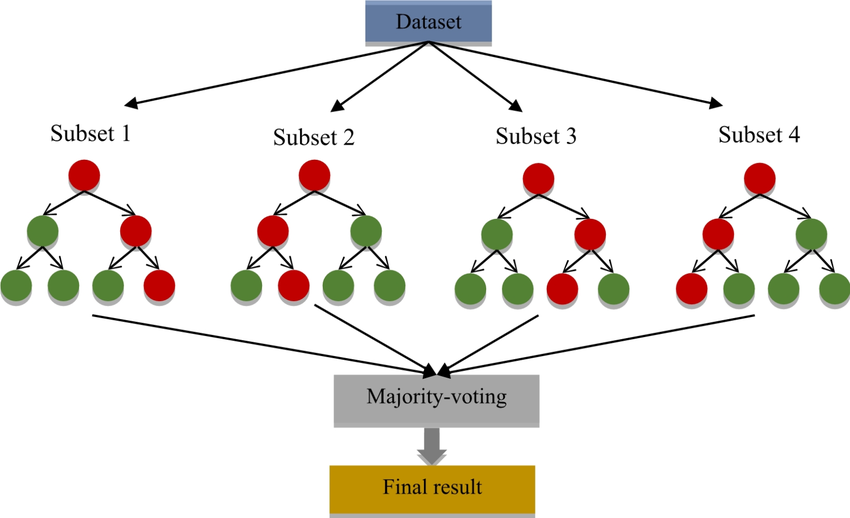
Gambar 2.4. Ilustrasi algoritma LightGBM

* XGBoost

eXtreme Gradient Boosting adalah framework gradient boosting yang dioptimalkan untuk efisiensi dan fleksibilitas. Selayaknya boosting lainnya, XGBoost juga menggunakan *ensembling weak learner*. Kualitas lain model ini adalah kemampuan untuk mengatasi overfitting melalui regularisasi internal.

Untuk meminimalisir kerugian, XGBoost bekerja dengan membangun decision tree secara bertahap. Hal yang membedakan XGBoost adalah penggunaan regularisasi dalam fungsi objektifnya, sehingga dapat mengontrol kompleksitas model dan mencegah overfitting. Selain itu, untuk mempercepat waktu pelatihan model, XGBoost memanfaatkan gradient dan hessian. XGBoost juga dioptimasi dengan pembangungan decision tree secara paralel dan data handling yang efisien sehingga mampu mengatasi dataset berukuran besar. Hal ini membuat XGBoost menjadi salah satu algoritma paling efisien. (Chen et al., 2016)

XGBoost memiliki beberapa hyperparameter yang bisa disesuaikan untuk mengatur bagaimana model memproses atau melatih data. Untuk mengatasi overfitting, terdapat max\_depth, min\_child\_weight, gamma, dan max\_cat\_threshold yang bisa dikurangi untuk mencegah model terlalu menghafal data training. Selain itu, terdapat metode untuk menambahkan randomness untuk membuat robust menjadi noise, yaitu subsample dancolsample\_bytree, dan num\_parallel\_tree. Hal lain yang bisa dilakukan juga adalah dengan mengurangi eta dan meningkatkan num\_round jika eta ditingkatkan. (XGBoost Developers, n.d.)



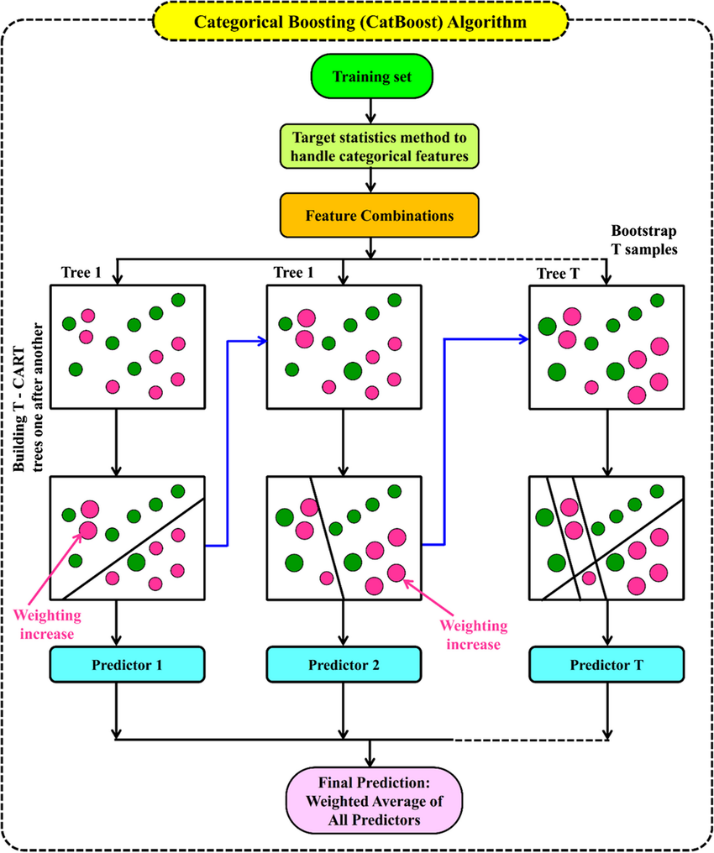
Gambar 2.5. Ilustrasi algoritma XGBoost

* CatBoost

Categorical Boosting juga merupakan framework gradient boosting yang canggih dengan memanfaatkan pendekatan baru untuk mengurangi prediction shift dan memanfaatkan fitur kategotikal secara efektif. Algoritma ini mengimplementasi ordered boosting, yaitu pemanfaatan permutasi acak dari training dataset untuk menghasilkan perkiraan gradien yang tidak bias. Catboost mengintegrasikan ordered target statistic langsung ke dalam proses training. Teknik ini menghitung representasi numerik untuk setiap kategori berdasarkan nilai target dari sampel sebelumnya dalam permutasi. (Prokhorenkova et al., 2018)

CatBoost memungkinkan model untuk belajar dari fitur kategorikal tanpa mengalami overfitting. Sehingga, CatBoost bisa menjadi model training yang sesuai untuk dataset dengan fitur kategorikal berkardinalitas tinggi. Hal itu menjadikan CatBoost salah satu training model canggih yang sering dipakai karena keakurasian yang tinggi dibandingkan model lain.

CatBoost juga memiliki hyperparameter yang bisa di*tuning* agar bisa menyesuaikan dengan dataset yang sedang dilatih. CatBoost memiliki fitur sendiri untuk mengatasi class imbalance yaitu auto\_class\_weights yang bisa mengatur agar CatBoost tahu bahwa data training imbalance dan CatBoost akan menyesuaikan dengan data tersebut. Selain itu, terdapat iteration yang mengindikasikan berapa iterasi yang akan menentukan seberapa kompleks model yang akan dipelajari. Terdapat juga learning\_rate yaitu kecepatan belajar dari CatBoost yang digunakan untuk mengontrol kontribusi pohon baru terhadap model secara menyeluruh.



Gambar 2.6. Ilustrasi algoritma CatBoost

* Ensemble Learning

Ensemble Learning adalah sebuah paradigma dalam machine learning di mana beberapa model individual digabung untuk menghasilkan satu model yang lebih kuat dan akurat. Tujuan utama dari metode ini adalah untuk meningkatkan stabilitas dan keandalan prediksi dengan mengurangi bias dan variance yang mungkin terjadi jika hanya mengandalkan satu model. Beberapa metode yang digunakan dalam penelitian ini, seperti Random Forest, XGBoost, dan LigtGBM, merupakan implementasi dari ensemble learning.

### **Metrik Evaluasi**

Metrik Evaluasi adalah alat pengukuran kuantitatif untuk mengukur performa dari suatu model dalam melakukan berbagai tugas, terutama klasifikasi biner. Metriks ini akan menunjukkan seberapa baik performa sebuah model dalam memprediksi hasil yang sesuai dengan kenyataan (Oona Rainio et al., 2024).

Terdapat empat jenis kemungkinan prediksi untuk klasifikasi biner dalam kasus ini yaitu

* True Positive (TP): Jumlah hasil positif yang diprediksi positif
* True Negative (TN): Jumlah hasil negatif yang diprediksi negatif
* False Positive (FP): Jumlah hasil negatif yang diprediksi positif
* False Negative (FN): Jumlah hasil positif yang diprediksi negatif

Metriks evaluasi untuk klasifikasi biner untuk mengekspresikan persentase adalah:

1. Accuracy

Akurasi adalah jumlah prediksi yang benar diantara total jumlah prediksi dan bisa dikalkulasi dengan:

1. Sensitivity/Recall

Recall adalah persentase jumlah prediksi positif yang benar diantara jumlah total predisksi data positif dan bisa dikalkulasikan dengan:

1. Specifity

Specifity atau kekhususan adalah persentase jumlah prediksi negatif yang benar diantara total jumlah prediksi data negatif dan bisa dikalkulasikan dengan:

1. Precision

Precision adalah persentase instansi positif yang benar-benar positif diantara prediksi positif yang telah dibuat dan bisa dikalkulasikan dengan:

1. F1 Score

F1 score merupakan rata-rata harmonis antara precision dan recall. F1 score mengukur secara seimbang antara performa model untuk memprediksi hasil positif maupun negatif (Oona Rainio et al., 2024) dan bisa dikalkulasikan dengan:

### Cross-Validation

*Cross-validation* adalah metode statistik yang digunakan untuk mengevaluasi performa model *machine learning* dan memastikan keandalannya. Tujuan utama dari *cross-validation* adalah untuk menghindari *overfitting,* yaitu masalah yang terjadi ketika model bekerja sangat baik pada data *training* namun gagal memberikan prediksi yang akurat pada data *testing* atau data baru yang belum pernah diliat oleh model sebelumnya. Metode ini bekerja dengan cara membagi dataset menjadi beberapa bagian atau subset. Model kemudian dilatih pada salah satu subset dan diuji pada subset lainnya. Proses ini diulang beberapa kali hingga setiap subset pernah digunakan sebagai data uji. Dalam penelitian ini, teknik *cross-validation* yang digunakan adalah Stratified K-Fold.

#### Stratified K-Fold

Stratified K-Fold merupakan variasi dari metode K-Fold Cross-Validation yang dirancang khusus untuk menangani dataset dengan distribusi kelas yang tidak seimbang (*imbalanced dataset*). Seperti yang telah diidentifikasi pada penelitian ini, jumlah siswa yang lulus dan tidak lulus pada seleksi SNBP sangat tidak seimbang.

Prinsip kerja Stratified K-Fold adalah sebagai berikut:

1. Dataset dibagi menjadi ‘k’ bagian (*fold*) yang memiliki ukuran yang sama.
2. Tidak seperti K-Fold biasa, proses pembagian pada Stratified K-Fold memastikan bahwa setiap fold memiliki proporsi atau persentase sampel dari setiap kelas yang sama dengan proporsi pada keseluruhan dataset. Misalnya, jika kelas “Lulus” hanya merupakan 10% dari total dataset, maka setiap fold juga akan berisi sekitar 10% data dari kelas “Lulus”.
3. Proses validasi silang kemudian dijalankan. Model akan dilatih sebanyak 'k' kali. Pada setiap iterasi, satu *fold* akan digunakan sebagai data uji (*validation set*) dan 'k-1' *folds* sisanya digunakan sebagai data latih (*training set*).
4. Metrik evaluasi (seperti akurasi, presisi, recall, F1-score) dihitung untuk setiap iterasi.
5. Hasil akhir dari metrik evaluasi adalah rata-rata dari 'k' iterasi tersebut.

Dengan menggunakan Stratified K-Fold, evaluasi model menjadi lebih robust dan dapat diandalkan karena memastikan bahwa model diuji pada semua bagian data dengan tetap menjaga representasi setiap kelas. Hal ini sangat penting dalam kasus prediksi kelulusan SNBP agar performa model tidak bias terhadap kelas mayoritas (siswa yang 'Tidak Lulus').

### Hyperparameter Tuning

Hyperparameter tuning bertujuan untuk menemukan kombinasi hyperparameter yang optimal guna memaksimalkan performa model. Berbeda dengan parameter yang nilainya dipelajari secara otomatis oleh model dari data selama proses training, hyperparameter adalah konfigurasi eksternal yang ditetapkan sebelum proses pelatihan dimulai. Pengaturan ini mengendalikan perilaku algoritma learning, seperti kompleksitas model atau kecepatan model dalam belajar. Proses pencarian konfigurasi terbaik ini seringkali menjadi tantangan karena ruang pencarian bisa sangat besar dan evaluasi setiap kombinasi membutuhkan sumber daya komputasi dan waktu yang intensif.

Mengacu pada ulasan komprehensif oleh Ilemobayo (2024), hyperparameter tuning sangat berpengaruh dalam meningkatkan kinerja dan kemampuan generalisasi model machine learning. Penelitian tersebut menjelaskan bahwa tanpa tuning yang tepat, sebuah model yang kuat sekalipun dapat menunjukkan performa yang buruk. Berbagai metode sistematis telah dikembangkan untuk menggantikan pencarian manual yang tidak efisien, di antaranya adalah *Grid Search*, *Random Search,* dan *Bayesian Optimization. Grid Search* secara menyeluruh menguji semua kemungkinan kombinasi *hyperparameter* dalam sebuah grid yang telah ditentukan. *Random Search* secara acak memilih kombinasi dari ruang pencarian dan seringkali lebih efisien. *Bayesian Optimization*, metode yang lebih canggih dari metode-metode sebelumnya, menggunakan hasil dari iterasi sebelumnya untuk secara cerdas memilih kombinasi berikutnya yang paling menjanjikan untuk dievaluasi. Pemilihan strategi *tuning* ini seringkali melibatkan pertukaran antara biaya komputasi dan kualitas akhir dari model yang dihasilkan.

# METODOLOGI

## Perancangan Sistem

Perancangan sistem penelitian ini menggunakan pendekatan komparatif berbasis skenario untuk membangun dan mengevaluasi model prediksi kelulusan SNBP. Pendekatan ini memungkinkan analisis mendalam mengenai dampak dari setiap tahapan—mulai dari pra-pemrosesan data hingga penanganan data tidak seimbang—terhadap performa model. Alur kerja dibagi menjadi beberapa skenario eksperimen yang berbeda, di mana setiap skenario dibangun di atas skenario sebelumnya untuk mengukur peningkatan kinerja secara sistematis.

### Alur Kerja

Meskipun setiap skenario memiliki fokus pemrosesan yang berbeda, terdapat alur kerja umum yang menjadi fondasi bagi semua eksperimen:

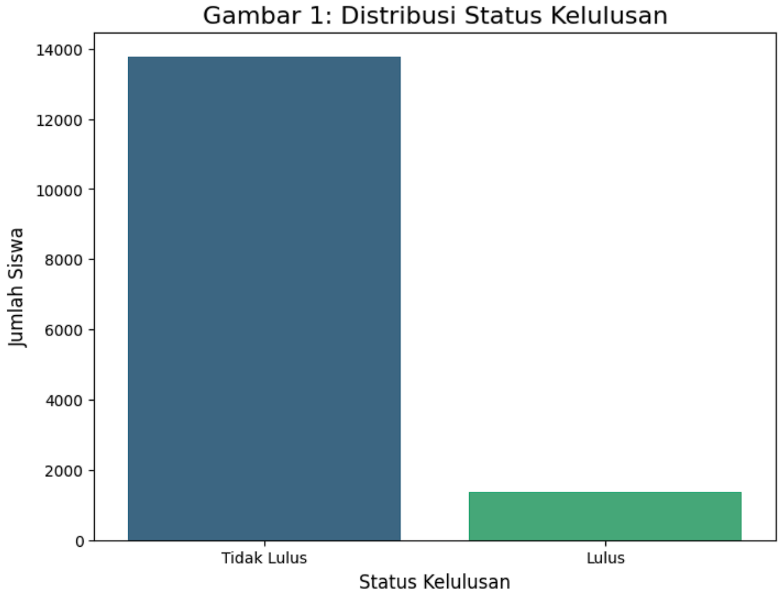
1. Pengumpulan Data: Proses diawali dengan memuat dataset menggunakan pustaka pandas dari dua file: train\_snbp.csv sebagai data latih dan test\_snbp.csv sebagai data uji.
2. Analisis Data Awal: Eksplorasi data awal dilakukan untuk memahami karakteristik dataset melalui fungsi-fungsi seperti .info(), .head(), .describe(), dan .isnull().sum() untuk memeriksa tipe data, format, statistik, dan persentase nilai yang hilang.
3. Model Klasifikasi yang Digunakan: Seluruh skenario akan menguji dan membandingkan beberapa algoritma klasifikasi yang sama, yaitu:
   1. CatBoost Classifier
   2. XGBoost Classifier
   3. LightGBM Classifier
   4. Support Vector Machine (SVM)
   5. Logistic Regression
   6. Random Forest Classifier
4. Metode Evaluasi: Kinerja model dievaluasi secara konsisten menggunakan:
   1. Cross-Validation: Menggunakan Stratified K-Fold dengan 5-lipatan (splits) pada data training untuk memastikan evaluasi yang robust dan tidak bias, terutama pada data yang imbalance.
   2. Metrik Evaluasi: classification\_report (mencakup precision, recall, f1-score), roc\_auc\_score, dan confusion\_matrix digunakan untuk menilai performa model pada data latih (hasil validasi silang) dan data uji.

### Exploratory Data Analysis (EDA)

Untuk mengenali dataset lebih dalam, maka Exploratory Data Analysis perlu dilakukan. Hal ini bertujuan agar bisa melanjutkan ke proses selanjutnya seperti preprocessing dan feature engineering, serta memilih model yang tepat untuk dataset.

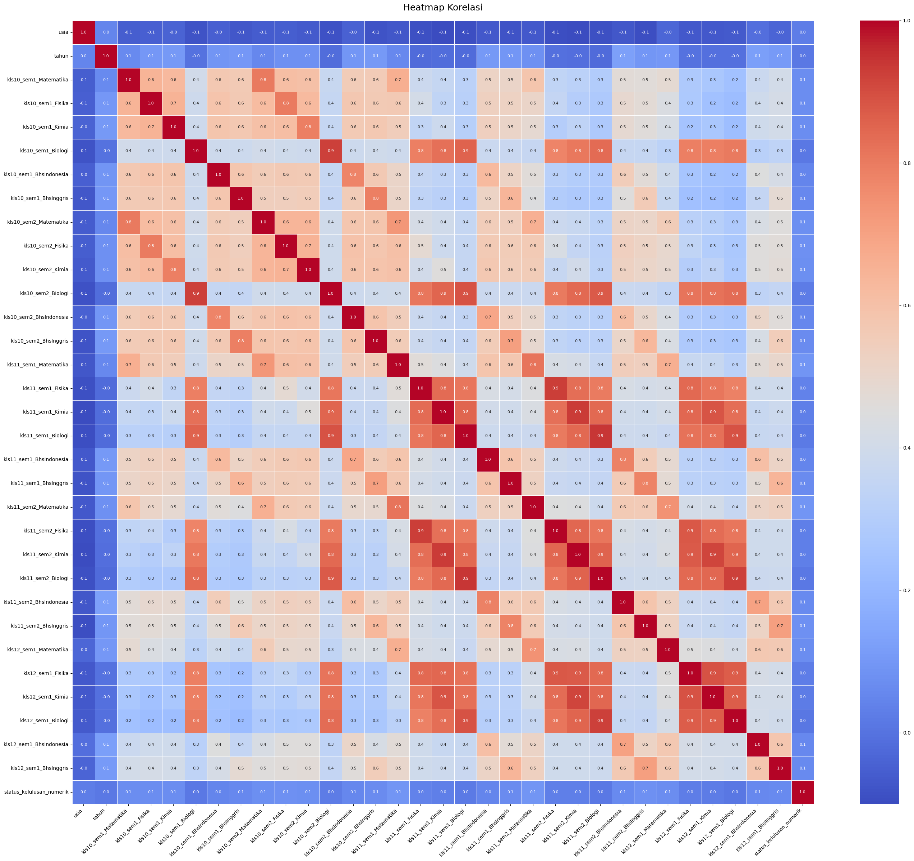
1. Status Kelulusan

Terdapat dua jenis status kelulusan, yaitu ‘Lulus’ dan ‘Tidak Lulus’. Jika dilihat dari perbandingan data, persentase ‘Tidak Lulus’ adalah 90.9% dan ‘Lulus’ hanya 9.09%, sehingga dataset ini termasuk dataset yang tidak seimbang. Bisa dilihat perbandingannya dari gambar dibawah ini.

   
Gambar 3.1. Distribusi Status Kelulusan

1. Heatmap Korelasi Variabel

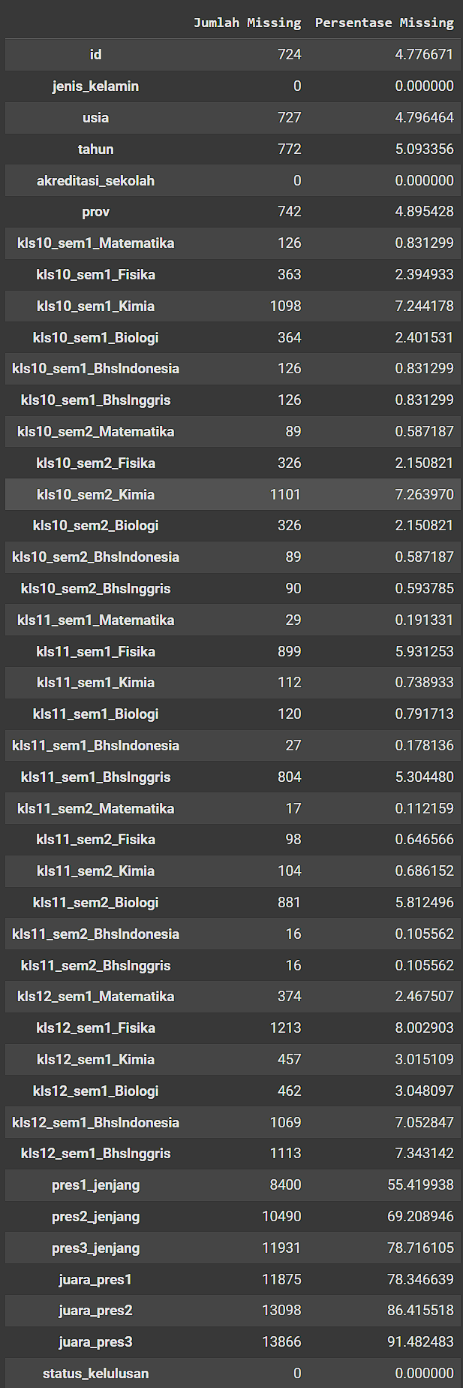
Untuk melihat korelasi antarvariabel serta hubungan suatu variabel dengan status kelulusan, maka diperlukan heatmap. Berikut adalah heatmap untuk korelasinya.

  
Gambar 3.2 Heatmap Korelasi

Dari heatmap ini, informasi yang didapatkan cukup banyak, yaitu mata pelajaran Kimia, Fisika, dan Biologi memiliki korelasi yang cukup tinggi, serta rata-rata korelasi variabel dengan status kelulusan sangat rendah, sehingga tidak ada satu variabel kuat yang menentukan status kelulusan.

1. Missing Value

Missing value atau data yang kosong di dalam dataset merupakan hal yang kerap terjadi. Namun, data-data ini bisa menyebabkan model kesulitan menemukan pola-pola tertentu pada data untuk memprediksi status kelulusan. Oleh sebab itu, terdapat berikut tabel yang menggambarkan jumlah missing values pada dataset ini.

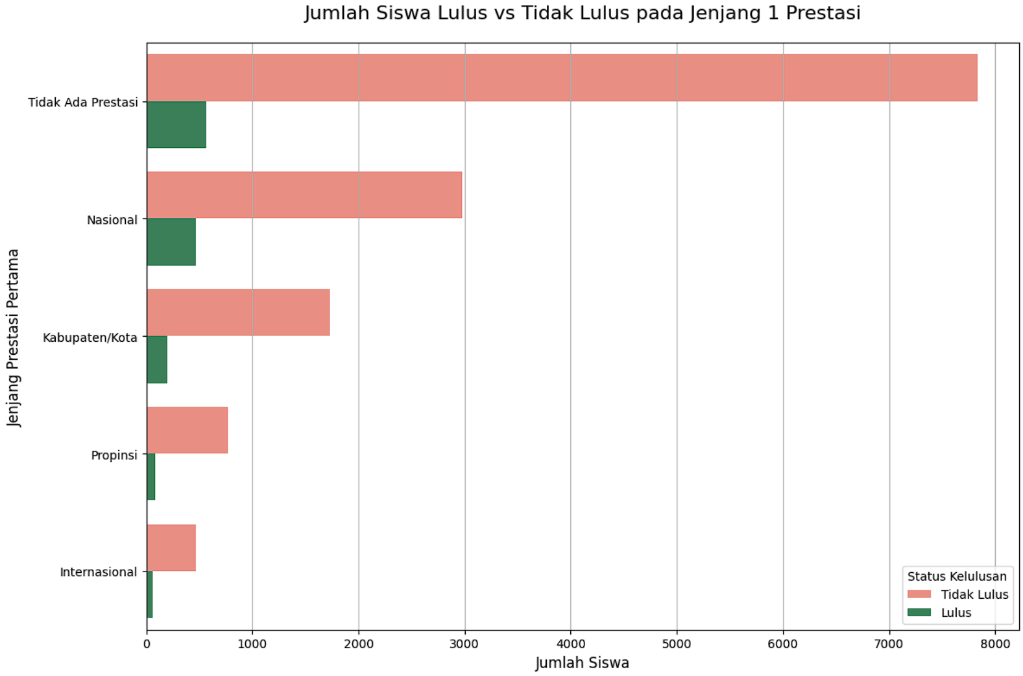


Gambar 3.3 Missing Values

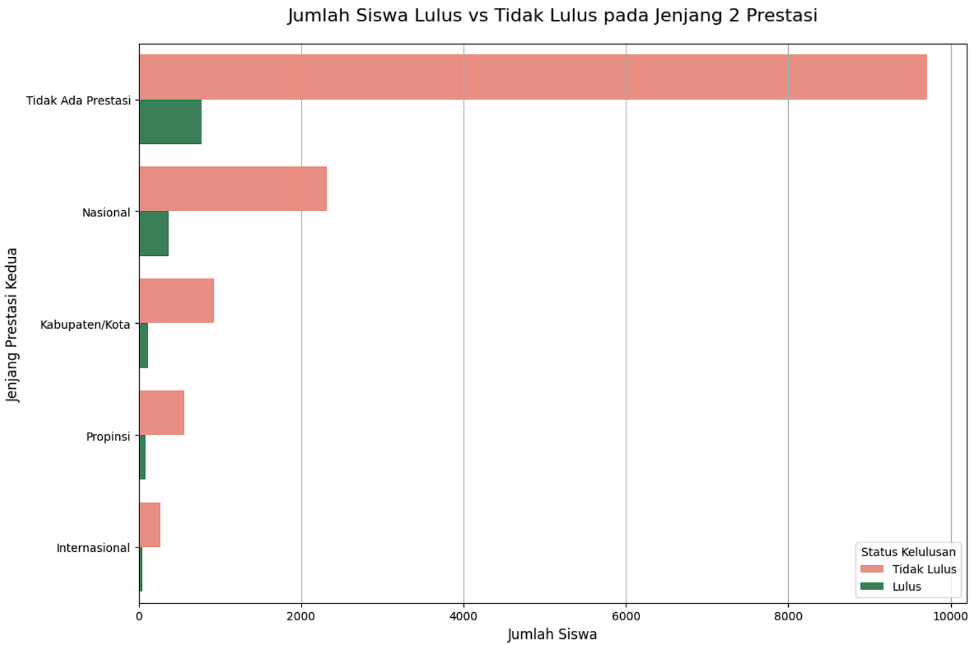
Jika dilihat dari missing values yang terdapat pada dataset, pada prestasi terdapat banyak yang tidak memiliki prestasi, oleh sebab itu wajar jika banyak null atau missing valuesnya. Selain itu, untuk nilai juga terdapat beberapa missing values, ini kemungkinan tidak wajar karena pada SMA dan SMK, mayoritas mata pelajaran dasar pasti ada dan mungkin missing values pada nilai bisa disebabkan oleh kesalahan input data. Lalu, pada kolom seperti usia, tahun, provinsi, dan id ada sekitar 700 data yang kosong, ini bisa terjadi karena kesalahan input data juga sehingga perlu dilakukan pengisian atau penanganan missing value.

1. Prestasi

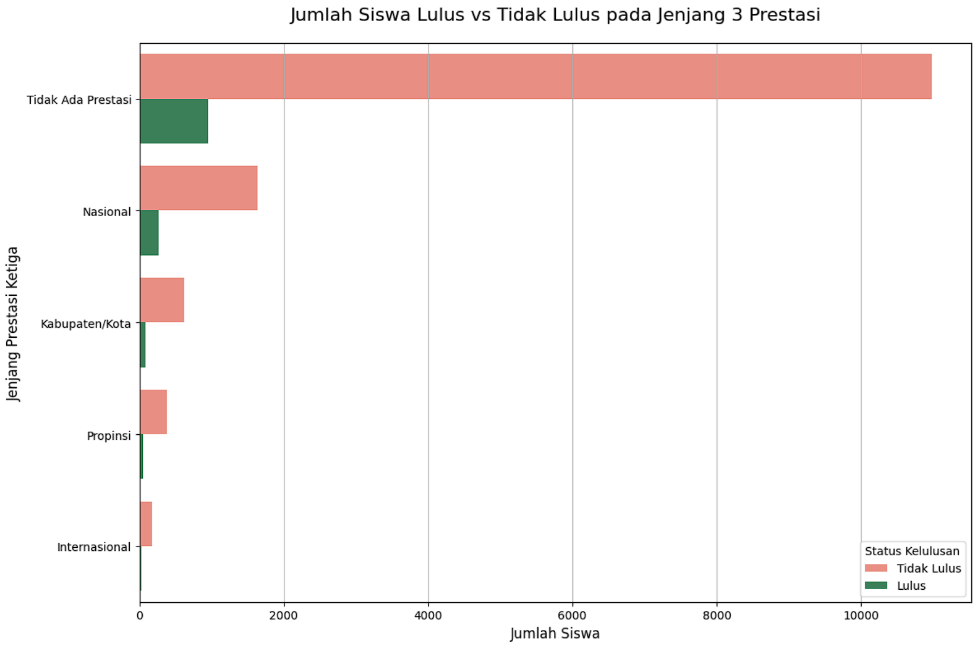
Prestasi merupakan faktor yang penting untuk dipertimbangkan dalam SNBP. Maka, analisis lebih lanjut tentang bentuk prestasi juga perlu diperhatikan. Berikut grafik yang menunjukkan status kelulusan tiap jenjang prestasi



Gambar 3.4 Plot Jenjang 1 Prestasi dan Status Kelulusan



Gambar 3.5 Plot Jenjang 2 Prestasi dan Status Kelulusan

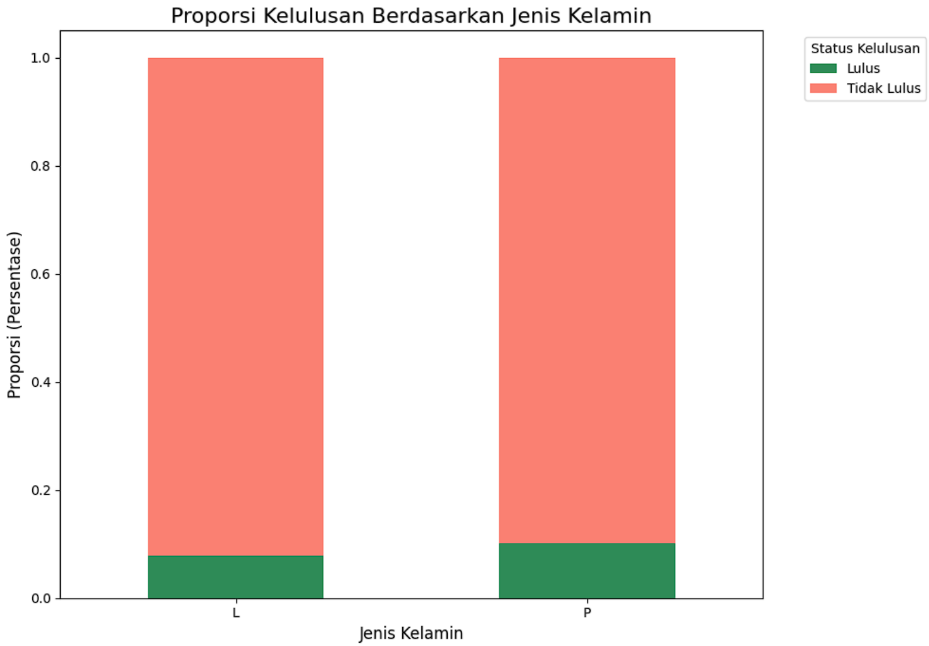


Gambar 3.6 Plot Jenjang 3 Prestasi dan Status Kelulusan

Pada ketiga plot tersebut, bisa dilihat bahwa siswa yang memiliki prestasi pada jenjang pertama lebih banyak yang ‘Lulus’ dibandingkan kedua jenjang dibawahnya. Namun, untuk jenjang-jenjang dibawahnya, siswa yang tidak memiliki prestasi lebih banyak yang lulus dibandingkan jenjang pertama. Namun, perbedaannya tidak terlalu signifikan karena perbandingan siswa yang memiliki prestasi dan ‘Lulus’ serta memiliki prestasi dan ‘Tidak Lulus’ tidak seimbang. Artinya, memiliki prestasi tidak meningkatkan kemungkinan ‘Lulus’ yang signifikan.

1. Jenis Kelamin

Dalam dataset ini, terdapat dua jenis kelamin, yaitu L untuk laki-laki dan P untuk perempuan. Secara teori, jenis kelamin bukan menjadi faktor penentu untuk kelulusan SNBP suatu orang. Bisa dilihat dari plot dibawah ini terkait distribusi jenis kelamin dan status kelulusan.

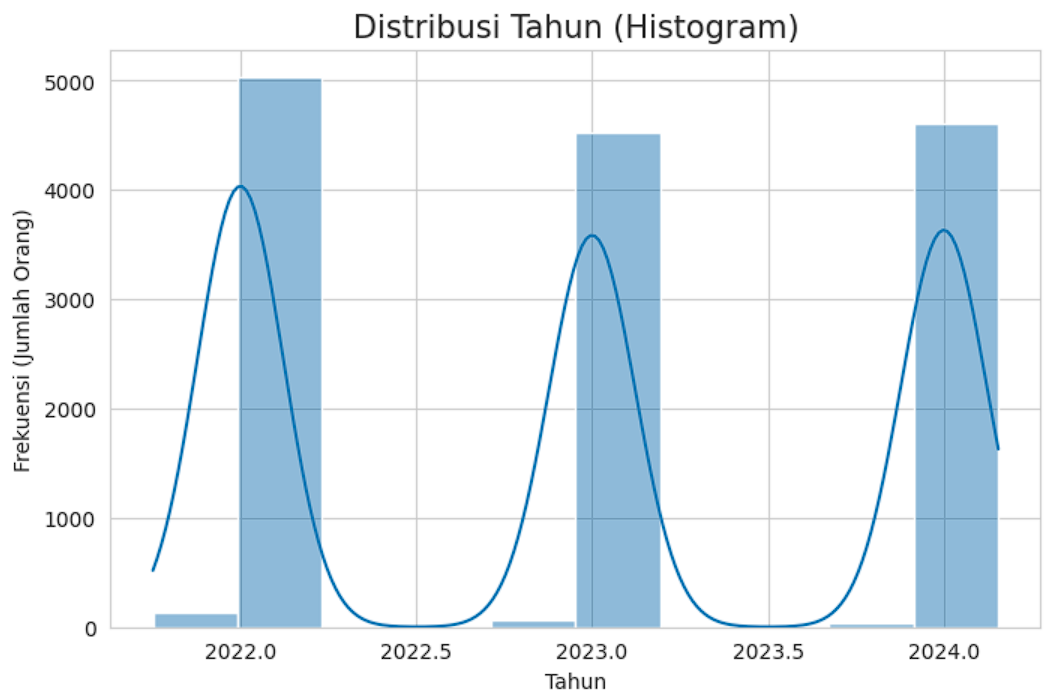


Gambar 3.7 Plot Proposi Kelulusan Berdasarkan Jenis Kelamin

Plot tersebut menggambarkan bahwa perbedaan proporsi tidak terlalu tinggi, terdapat sedikit lebih banyak perempuan dibandingkan laki-laki yang ‘Lulus’. Hal ini tidak menunjukkan bahwa jenis kelamin memiliki pengaruh yang tinggi dalam menentukan status kelulusan.

1. Tahun

Dalam dataset SNBP, terdapat tahun untuk mengindikasikan data tersebut berasal dari tahun berapa. Tahun bisa menjadi fitur kategorikal yang akan diklasifikasi menjadi beberapa tahun, namun bentuknya masih float dan banyak bentuk desimal. Seperti yang dilihat dari histogram tahun di bawah ini.

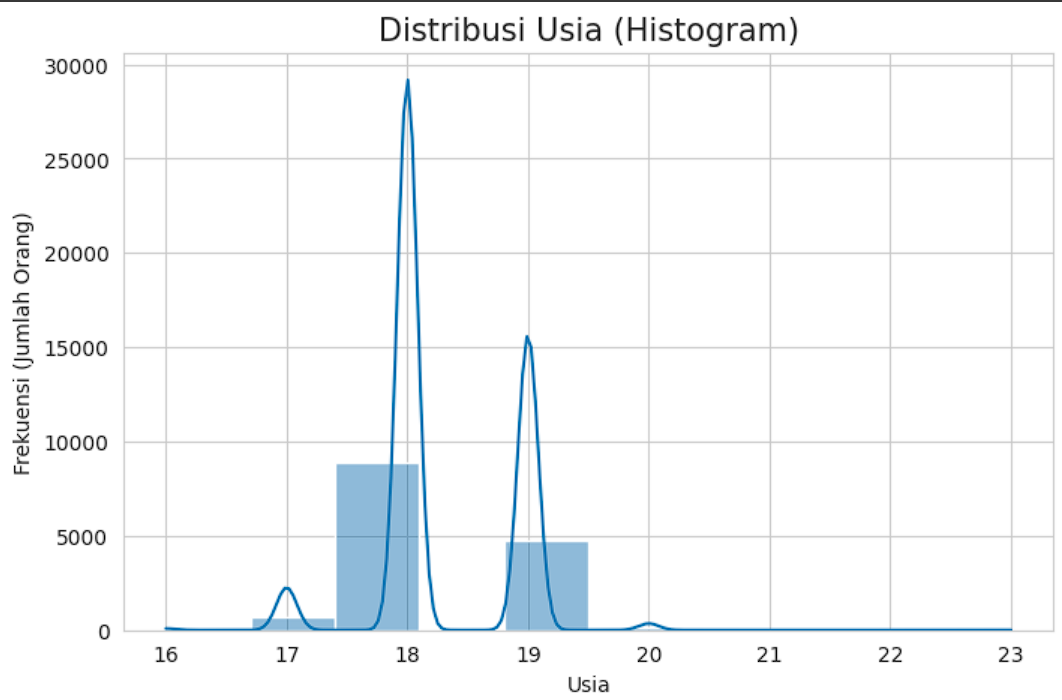


Gambar 38 Histogram Tahun

Terdapat banyak value berbeda-beda antara tahun dengan kebanyakan di sekitar 2022, 2023, dan 2024. Maka, tahun perlu disederhanakan menjadi int atau string sehingga bisa diklasifikasikan sebagai fitur kategorikal.

1. Usia

Setiap pendaftar harus memasukkan usia saat ia mendaftar SNBP sebagai salah satu syarat, dikarenakan menurut aturan SNBP, usia maksimal adalah 25. Namun, setelah dilakukann pengecekan, hasilnya menunjukkan tidak ada siswa yang berusia diatas 25 tahun. Maka, bisa dilihat dari histogram berikut terkait pembagian usia siswa.

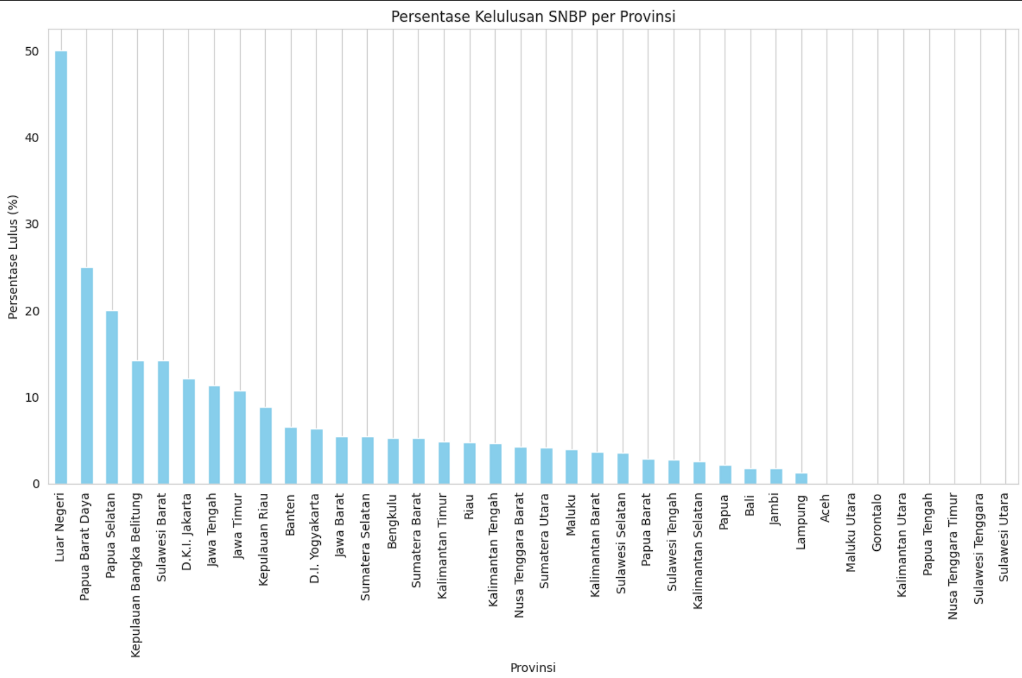


Gambar 3.9 Histogram Distribusi Usia

Mayoritas usia pendaftar SNBP adalah 18 tahun, lalu disusul dengan 19 tahun. Selanjutnya hanya ada sedikit orang yang 17 tahun. Selain ketiga angka tersebut, jumlahnya sangatlah sedikit.

1. Provinsi

Provinsi asal seorang siswa bisa menentukan kelulusan siswa tersebut, karena SNBP juga memandang alumni sekolah dan sekolah dari pendaftar. Maka, provinsi bisa menjadi pertimbangan kelulusan seorang siswa. Oleh sebab itu, berikut persentase kelulusan SNBP per provinsi



Gambar 3.10 Persentase Kelulusan SNBP per Provinsi

Bisa terlihat persentase kelulusan luar negeri sangatlah tinggi yaitu 50%, diikuti oleh Papua dan Kepulauan Bangka Belitung. Namun, persentase ini bukanlah segalanya, karena jika jumlah pendaftar kecil, maka persentase akan semakin mudah meningkat. Dari provinsi, kebanyakan pendaftar berasal dari Jawa Timur dengan total 8033 pendaftar dan persentase kelulusannya 10.73%. Hal ini cukup tinggi dibandingkan provinsi lainnya yang jumlahnya lebih sedikit namun persentasenya lebih sedikit. Namun, terdapat juga provinsi yang memiliki persentase kelulusan 0% dan mendekatinya, padahal jumlah pendaftarnya juga tidak sedikit. Hal ini bisa berarti beberapa provinsi bisa mempengaruhi status kelulusan siswa.

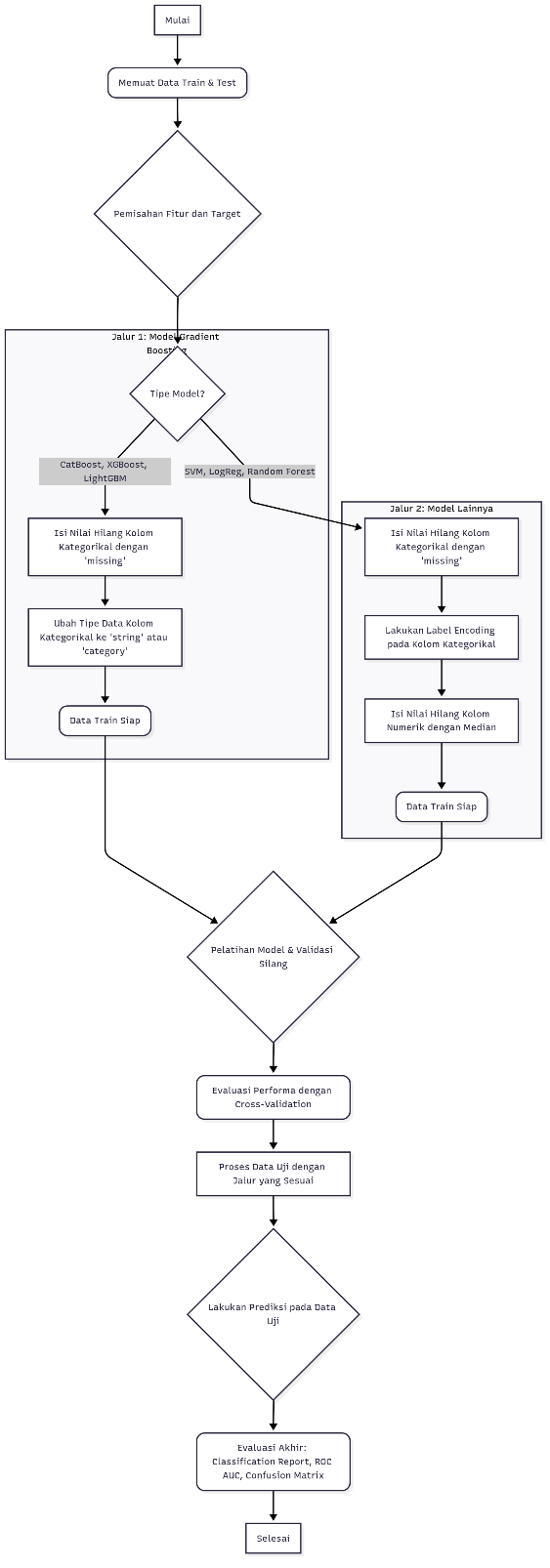
1. Akreditasi Sekolah

Setiap sekolah memiliki akreditasi dari A hingga C. Jumlah akreditasi terbanyak adalah A dengan angka hampir mencapai 13.000 lebih siswa. Sedangkan akreditasi B dan C jumlahnya sangat kecil. Namun, tingkat kelulusan SNBP berdasarkan akreditasi sekolah mengatakan bahwa 9.26% siswa dari sekolah berakreditasi A ‘Lulus’. Hal ini bisa menjadi tanda bahwa dengan adanya akreditasi A, hal tersebut memudahkan siswa atau meningkatkan peluangnya untuk ‘Lulus’, namun tidak terlalu signifikan karena juga banyak pesaing yang berasal dari sekolah berakreditasi A.

j. Nilai

Nilai merupakan faktor penentu paling penting bagi para siswa, namun perlu dilakukan beberapa pengecekan terhadap nilai. Seperti yang dilihat dari missing values, terdapat beberapa nilai yang masih kosong, sehingga hal tersebut nantinya perlu diisi. Selain itu, terdapat sejumlah nilai yang menggunakan indeks 0-10 alih-alih 0-100. Hal ini perlu diperbaiki agar model tidak salah menginterpretasikan nilainya. Dapat dilihat bahwa siswa yang terdapat value 0 atau NaN pada nilai rapot semesternya masih banyak yang diterima, oleh karena itu bisa diasumsikan bahwa nilai 0 atau NaN merupakan kesalahan input, karena semua mata pelajaran tersebut adalah mata pelajaran dasar yang harus dimiliki siswa SMA maupun SMK.

### Skenario 1: Model Baseline



Gambar 3.11. Flowchart Skenario Baseline

Skenario ini bertujuan untuk membangun sebuah tolok ukur (benchmark) performa. Model dilatih dengan pre-processing data yang minimal untuk melihat kinerja dasar dari setiap algoritma pada dataset asli.

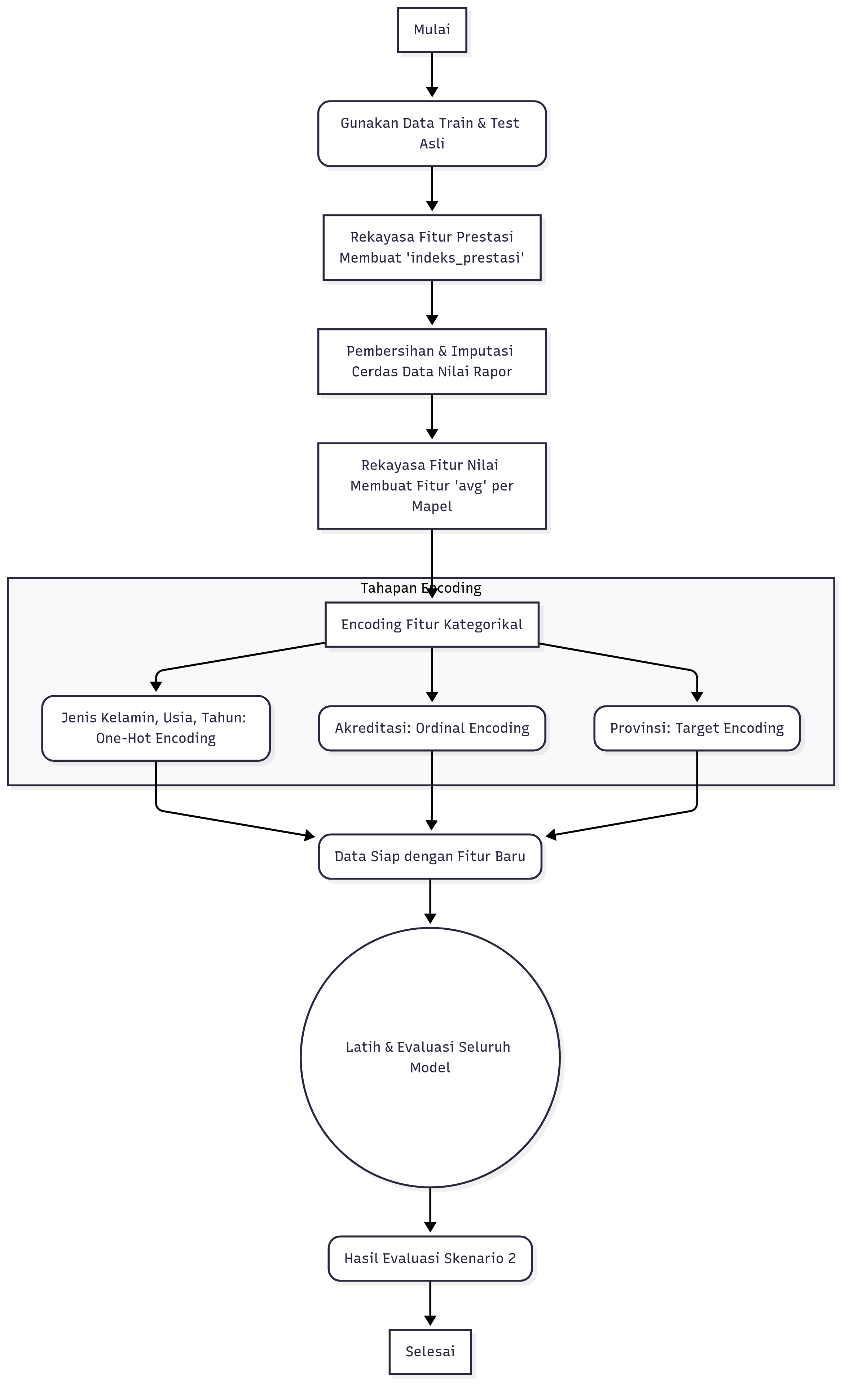
#### Preprocessing

Untuk model berbasis gradient boosting (CatBoost, XGBoost, LightGBM), nilai yang hilang pada kolom kategorikal diisi dengan string ‘missing’, karena model-model ini memiliki kemampuan internal untuk menanganinya. Sedangkan untuk model lainnya (SVM, Logistic Regression, Random Forest), nilai yang hilang pada kolom numerik diisi dengan nilai median dari masing-masing kolom, dan kolom kategorikal di-encode menggunakan Label Encoding.

#### Training dan Evaluasi

Semua model dilatih dan dievaluasi menggunakan data hasil pemrosesan minimal ini. Hasil dari skenario ini menjadi dasar perbandingan untuk skenario-skenario berikutnya.

### Skenario 2: Feature Engineering dan Preprocessing Lanjutan



Gambar 3.12. Flowchart Skenario Feature Engineering

Skenario ini bertujuan untuk mengukur dampak dari feature engineering dan strategi preprocessing yang lebih canggih terhadap peningkatan performa model.

#### Preprocessing dan Feature Engineering

* Penanganan Nilai

Nilai mata pelajaran ditangani dengan menerapkan strategi imputasi berlapis, termasuk normalisasi skala, pengisian NaN berdasarkan rata-rata nilai siswa per mata pelajaran, dan penggunaan korelasi antar-mapel. Fitur baru seperti avg\_Matematika juga dibuat.

* Feature Engineering Prestasi

Fitur jenjang dan juara distandarisasi, misalnya juara 'Medali Emas' diubah menjadi ‘Juara 1’. Setelah itu encode jenjang dan juara menjadi numerik dengan memberikan bobot, semakin tinggi jenjang dan juaranya maka akan semakin besar bobotnya. Langkah terakhir adalah membuat fitur komposit indeks\_prestasi dengan mengalikan bobot jenjang dan juara.

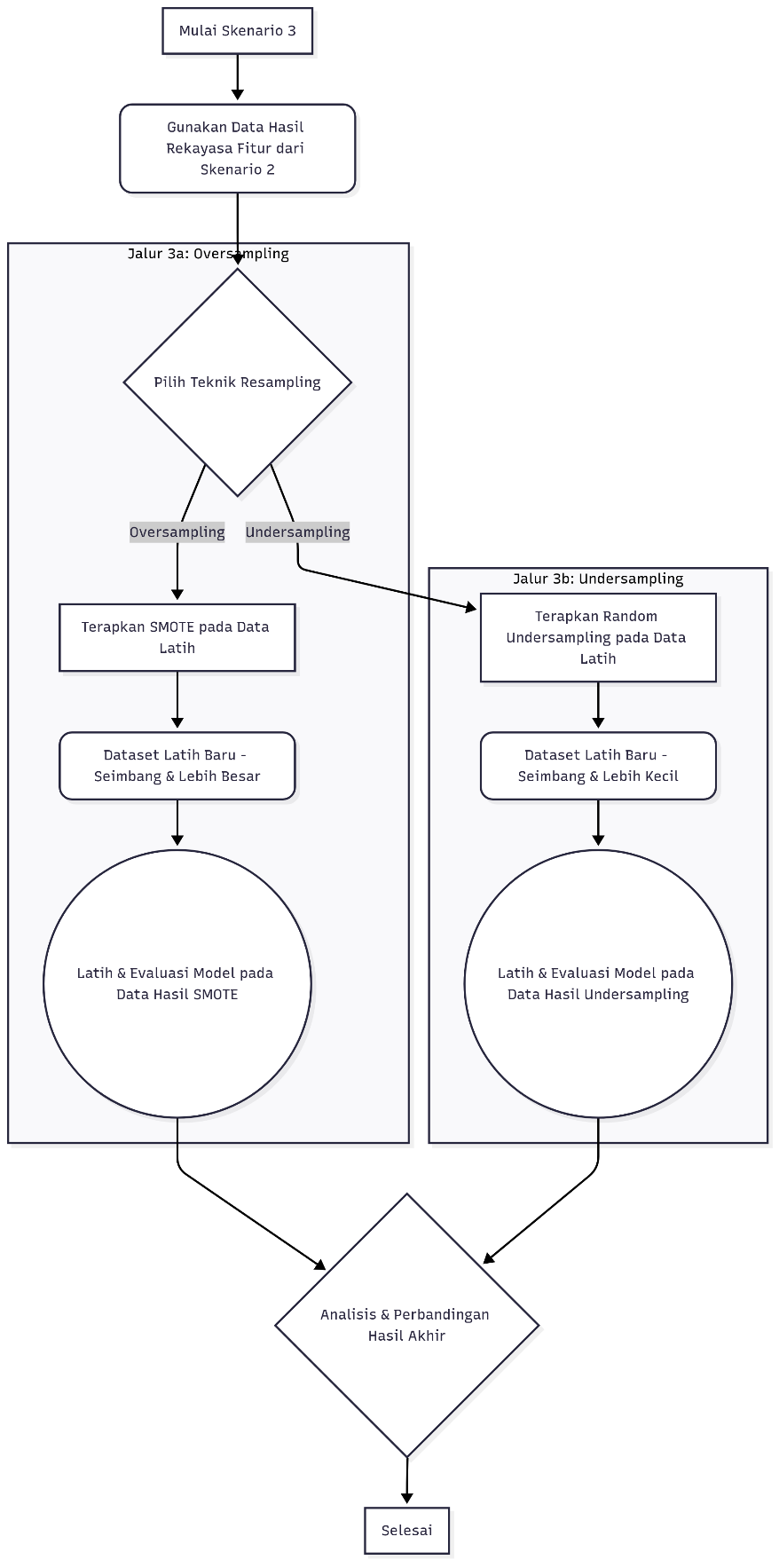
* Encoding Kategorikal

Menerapkan target encoding dengan smoothing pada fitur prov, One-Hot Encoding pada jenis\_kelamin, usia, dan tahun, serta Ordinal Encoding pada akreditasi\_sekolah.

#### Training dan Evaluasi

Semua model dilatih kembali menggunakan dataset yang telah diperkaya dengan fitur-fitur baru ini, lalu hasilnya dibandingkan dengan skenario baseline.

### Skenario 3: Handling Imbalanced Data



Gambar 3.13. Flowchart Skenario Imbalanced Data

Skenario ini berfokus untuk mengatasi masalah class imbalance yang signifikan dalam dataset. Tujuannya adalah untuk meningkatkan kemampuan model dalam memprediksi kelas minoritas. Terdapat dua macam teknik yang digunakan dalam mengatasi data imbalance pada penelitian ini, yaitu Oversampling dengan menggunakan SMOTE (Synthetic Minority Over-sampling Technique, serta Undersampling.

#### Oversampling dengan SMOTE (Synthetic Minority Over-sampling Technique)

Oversampling pada sub-skenario ini diimplementasikan menggunakan SMOTE dari library imblearn. Tujuannya adalah untuk menyeimbangkan distribusi kelas dengan cara membuat sampel sintetis dari kelas minoritas (‘Lulus’). Proses ini dijalankan pada data train yang telah melalui seluruh tahap rekayasa fitur dari Skenario 2.

Implementasi spesifiknya adalah sebagai berikut:

1. Sebelum SMOTE diterapkan, semua fitur kategorikal diidentifikasi untuk memastikan data siap diproses.
2. SMOTE kemudian diinisialisasi dengan random\_state=100 untuk memastikan proses pembuatan data sintetis dapat direproduksi secara konsisten.
3. Metode .fit\_resample() diterapkan pada data latih, yang secara efektif menghasilkan sampel-sampel baru untuk kelas ‘Lulus’ hingga jumlahnya setara dengan kelas ‘Tidak Lulus’

Keunggulan pendekatan ini adalah model dapat berlatih pada dataset yang lebih kaya dan seimbang secara proporsional tanpa membuang informasi apapun dari kelas mayoritas.

##### Training dan Evaluasi

Setelah dataset train diseimbangkan melalui SMOTE, semua model dilatih kembali menggunakan data X\_resampled dan y\_resampled. Proses evaluasi menggunakan Stratified K-Fold pada data balance ini, diikuti dengan pengujian akhir pada data test yang asli dan tidak seimbang untuk mengukur kinerja generalisasi model.

#### Undersampling dengan Random Undersampling

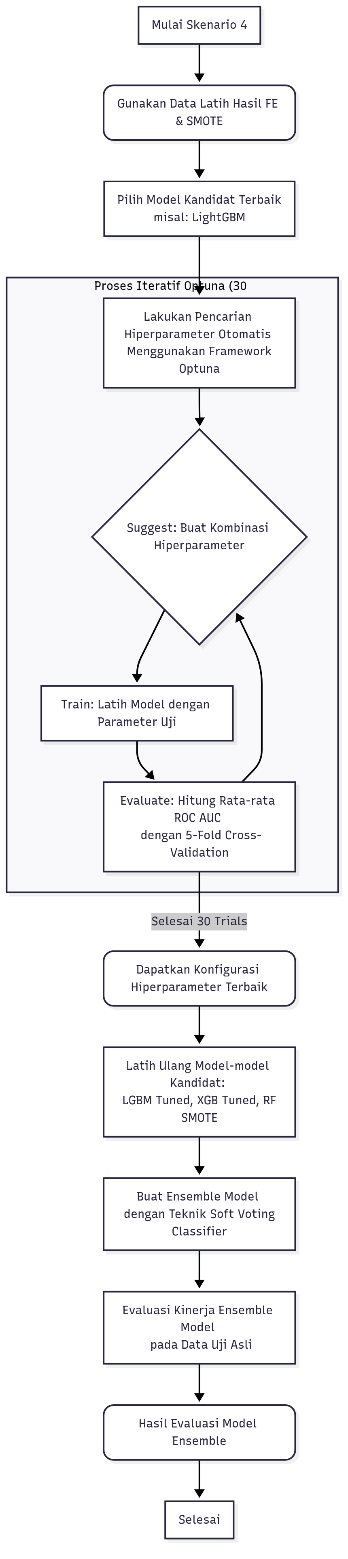
Sub-skenario ini menerapkan teknik Random Undersampling menggunakan implementasi RandomUnderSampler dari library imblearn. Berbeda dengan SMOTE, metode ini menyeimbangkan dataset dengan cara mengurangi jumlah sampel dari kelas mayoritas ('Tidak Lulus'). Proses ini dilakukan dengan menghapus sampel dari kelas mayoritas secara acak hingga jumlahnya setara dengan jumlah sampel pada kelas minoritas ('Lulus'), menghasilkan dataset latih dengan rasio kelas 1:1 yang seimbang. Untuk memastikan hasil yang dapat direproduksi, parameter random\_state=100 digunakan selama proses inisialisasi. Penting untuk dicatat bahwa teknik ini diterapkan pada data yang telah melalui seluruh proses rekayasa fitur dari Skenario 2.

Teknik ini memiliki keunggulan dalam hal efisiensi komputasi karena secara signifikan mengurangi ukuran dataset latih. Namun, pendekatan ini memiliki risiko inheren, yaitu potensi kehilangan informasi berharga yang terkandung dalam sampel-sampel kelas mayoritas yang dibuang, yang dapat memengaruhi kemampuan generalisasi model. Risiko ini tentunya akan sangat berpengaruh pada dataset imbalance seperti yang digunakan pada penelitian ini.

##### Training dan Evaluasi

Seluruh model dilatih ulang menggunakan dataset train baru yang berukuran lebih kecil dan seimbang ini (X\_resampled\_us, y\_resampled\_us). Seperti pada sub-skenario sebelumnya, evaluasi akhir dilakukan terhadap dataset uji asli untuk melihat bagaimana model yang dilatih pada data yang lebih sedikit ini berperforma pada data dunia nyata yang tidak seimbang.

### Skenario 4: Hyperparameter Tuning



Gambar 3.14. Flowchart Skenario Hyperparameter Tuning

Skenario ini bertujuan untuk melakukan optimisasi pada model-model yang menunjukkan performa paling baik setelah melalui skenario-skenario sebelumnya. Tujuannya adalah untuk menemukan kombinasi hyperparameter yang dapat memaksimalkan kemampuan generalisasi model pada data baru, sekaligus mengatasi potensi overfitting yang teridentifikasi pada skenario sebelumnya.

#### Metode

* Pemilihan Model dan Data

Berdasarkan hasil dari Skenario 3, model yang dipilih untuk proses optimisasi adalah **LightGBM** dan **XGBoost**. Proses tuning ini secara spesifik dilakukan pada data latih yang telah melalui **rekayasa fitur dan *oversampling* menggunakan SMOTE**, karena kombinasi ini menunjukkan hasil awal terbaik.

* Framework Optimisasi

Hyperparameter tuning dilakukan secara otomatis menggunakan *framework* **Optuna**. Optuna dipilih karena kemampuannya untuk secara efisien mencari ruang parameter yang kompleks untuk menemukan kombinasi nilai yang optimal.

* Proses Pencarian
  + Sebuah fungsi objective didefinisikan untuk setiap model, di mana Optuna akan mencoba berbagai kombinasi hiperparameter.
  + Ruang pencarian (search space) untuk setiap hiperparameter (seperti learning\_rate, max\_depth, num\_leaves, n\_estimators) ditentukan dengan rentang nilai yang dipilih untuk mendorong regularisasi dan mengurangi overfitting.
  + Untuk setiap iterasi (trial), performa dari kombinasi hiperparameter dievaluasi menggunakan metrik rata-rata ROC AUC dari proses 5-Fold Stratified Cross-Validation pada data latih hasil SMOTE.
  + Optuna menjalankan proses pencarian ini sebanyak 30 trials untuk setiap model untuk memastikan eksplorasi yang memadai dalam menemukan parameter terbaik.

#### Training Akhir dan Evaluasi

Setelah kombinasi hiperparameter terbaik ditemukan oleh Optuna, model final (LGBM dan XGBoost) diinisialisasi menggunakan parameter tersebut. Model yang telah dioptimalkan ini kemudian dilatih pada keseluruhan data latih hasil SMOTE dan dievaluasi performanya pada data uji yang asli dan tidak seimbang.

## Implementasi Sistem

Implementasi sistem dilakukan di dalam file python notebook dan dikerjakan secara daring di dalam Google Colab.

### Pemuatan Data dan Analisis Awal

Proses dimulai dengan memuat data dari file CSV dan melakukan EDA untuk memahami karakteristiknya.

|  |  |  |
| --- | --- | --- |
| **Load Data** | | |
| **1** | PROCEDURE Load\_Data: | |
| **2** |  | train\_data = LOAD\_CSV(‘train\_snbp.csv’) |
| **3** |  | test\_data = LOAD\_CSV(‘test\_snbp.csv’) |
| **4** |  | RETURN train\_data, test\_data |
| **5** | END PROCEDURE | |
|  |  | |
| Exploratory Data Analysis | | |
| **1** | PROCEDURE Exploratory\_Data\_Analysis(data): | |
| **2** |  | DISPLAY data.info() |
| **3** |  | DISPLAY data.head() |
| **4** |  | DISPLAY data.describe() |
| **5** |  | DISPLAY data.isnull().sum() |
| **6** | END PROCEDURE | |
|  |  | |
| EKSEKUSI | | |
| **1** | (train\_data, test\_data) = Load\_Data() | |
| **2** | Exploratory\_Data\_Analysis(train\_data) | |

### Skenario 1: Implementasi Baseline

Pseudocode ini menunjukkan logika percabangan untuk pra-pemrosesan minimal yang berbeda antara model *gradient boosting* dan model lainnya.

|  |  |  |
| --- | --- | --- |
| Procedure Preprocess\_Baseline(data, model\_type): | | |
| 1 | DEFINE categorical\_features = [‘jenis\_kelamin’, ‘akreditasi\_sekolah’, ‘prov’, …] | |
| 2 |  | |
| 3 | // Handling missing value | |
| 4 | FOR each column IN categorical\_features: | |
| 5 |  | FILL missing values in column WITH “missing” |
| 6 | ENDFOR | |
| 7 |  | |
| 8 | // Penyesuaian tipe data berdasarkan jenis model | |
| 9 | IF model\_type IS “Boosting”: | |
| 10 |  | CONVERT categorical\_features TO ‘category’ type |
| 11 | ELSE: | |
| 12 |  | APPLY LabelEncoder to categorical\_features |
| 13 |  | FILL missing numeric values WITH data.median() |
| 14 | ENDIF | |
| 15 |  | |
| 16 | RETURN preprocessed\_data | |
| END PROCEDURE | | |

### Skenario 2: Implementasi Feature Engineering

Proses ini mencakup beberapa langkah transformasi data yang kompleks untuk menciptakan fitur-fitur baru yang lebih informatif.

|  |  |  |
| --- | --- | --- |
| Procedure Feature\_Engineering(data): | | |
| 1 | // 1. Rekayasa Fitur Prestasi | |
| 2 | FOR each prestasi\_column IN [‘juara\_pres1’, ‘juara\_pres2’, ‘juara\_pres3’]: | |
| 3 |  | STANDARDIZE award names )e.g., ‘Medali Emas’ -> ‘Juara 1’) |
| 4 |  | ENCODE award ranks to numeric (Juara 1 -> 3) |
| 5 |  | ENCODE award levels to numeric (Internasional -> 4) |
| 6 | ENDFOR | |
| 7 | CALCULATE ‘indeks\_prestasi’ by multiplying rank and level scores | |
| 8 |  | |
| 9 | // 2. Pembersihan & Imputasi Nilai | |
| 10 | DEFINE nilai\_columns | |
| 11 | REPLACE 0 with NaN in nilai\_columns | |
| 12 | SCALE values <= 10 by multiplying with 10 | |
| 13 | IMPUTE NaN using student’s own average score per subject | |
| 14 | IMPUTE remaining NaN using most correlated subject in the same semester | |
| 15 | CREATE ‘avg\_’ features for each subject | |
| 16 |  | |
| 17 | // 3. Encoding Fitur Kategorikal | |
| 18 | ENCODE ‘prov’ using Target Encoding with smoothing | |
| 19 | ENCODE ‘akreditasi\_sekolah’ using Ordinal Encoding | |
| 20 | ENCODE ‘jenis\_kelamin’, ‘usia’, ‘tahun’ using One-Hot Encoding | |
| 21 |  | |
| 22 | RETURN engineered\_data | |
| END PROCEDURE | | |

### Skenario 3: Implementasi Teknik Resampling

Pseudocode ini menunjukkan dua pendekatan berbeda untuk menangani data tidak seimbang, yang diterapkan setelah Skenario 2.

|  |  |
| --- | --- |
| Procedure Apply\_SMOTE(x\_train, y\_traiin): | |
| 1 | INITIALIZE smote = SMOTE(random\_state=100) |
| 2 | (x\_resampled, y\_resampled) = smote.fit\_resample(x\_train, y\_train) |
| 3 | RETURN x\_resampled, y\_resampled |
| END PROCEDURE | |

|  |  |
| --- | --- |
| Procedure Apply\_random\_undersampling(x\_train, y\_traiin): | |
| 1 | INITIALIZE rus = RandomUnderSampler(random\_state=100) |
| 2 | (x\_resampled, y\_resampled) = rus.fit\_resample(x\_train, y\_train) |
| 3 | RETURN x\_resampled\_us, y\_resampled\_us |
| END PROCEDURE | |

### Skenario 4: Implementasi Hyperparameter Tuning

Proses ini menggunakan *framework* Optuna untuk mencari kombinasi hiperparameter terbaik secara otomatis.

|  |  |  |  |
| --- | --- | --- | --- |
| Procedure tune\_hyperparameters(x\_train, y\_train): | | | |
| 1 | DEFINE list\_of\_max\_depth AS [5, 16, 28, 39, 50] | | |
| 2 | DEFINE list\_of\_min\_child\_weight AS [1, 6, 11, 15, 20] | | |
| 3 | // … parameter lainnya | | |
| 4 |  | | |
| 5 | CREATE search\_space\_grid WITH { | | |
| 6 |  | “max\_depth”: list\_of\_max\_depth | |
| 7 |  | “min\_child\_weight”: list\_of\_min\_child\_weight | |
| 8 |  | // … parameter lainnya | |
| 9 | } | | |
| 10 |  | | |
| 11 | INITIALIZE study = Optuna.create\_study(direction=’maximize’) | | |
| 12 |  | | |
| 13 | DEFINE FUNCTION evaluate\_one\_combination(current\_trial): | | |
| 14 |  | params = { | |
| 15 |  |  | ‘max\_depth’: trial.suggest\_categorical(‘max\_depth’, search\_space [‘max\_depth], |
| 16 |  |  | 'min\_child\_weight': trial.suggest\_categorical('min\_child\_weight', search\_space['min\_child\_weight']), |
| 17 |  |  | // … parameter lainnya |
| 18 |  |  | |
| 19 |  | model = model(\*\*params) | |
| 20 |  | score = cross\_val\_score(model, x\_train, y\_train, cv=cv, scoring = ‘roc\_auc’).mean() | |
| 21 |  |  | |
| 22 |  | return score | |
| 23 |  | | |
| 24 | n\_trials = total combination of parameters | | |
| 25 | study\_model.optimize(objective, n\_trials) | | |
| END PROCEDURE | | | |

### Proses Umum Training dan Evaluasi Model

Pseudocode ini merupakan kerangka umum yang digunakan di setiap akhir skenario untuk melatih dan mengevaluasi model.

|  |  |
| --- | --- |
| Procedure train\_and\_evaluate(model, x\_train, y\_train, x\_test, y\_test): | |
| 1 | // Evaluasi pada data latih dengan cv |
| 2 | INITIALIZE cv = StratifiedKFold(n\_splits=5) |
| 3 | y\_pred\_cv = cross\_val\_predict(model, x\_train, y\_train, cv=cv) |
| 4 | DISPLAY classification\_report(y\_train, y\_pred\_cv) |
| 5 |  |
| 6 | //Latih model final pada seluruh data latih |
| 7 | FIT model on x\_train, y\_train |
| 8 |  |
| 9 | //Evaluasi pada data uji |
| 10 | y\_pred\_test = model.predict(x\_test) |
| 11 | DISPLAY classification\_report(y\_test, y\_pred\_test) |
| 12 | DISPLAY confusion\_matrix(y\_test, y\_pred\_test) |
| 13 | DISPLAY roc\_auc\_score(y\_test, model.predict\_proba(x\_test)[:, 1]) |
| END PROCEDURE | |

# Hasil dan Pembahasan

## Hasil penelitian

### **Skenario 1: Model Baseline**

Skenario ini bertujuan untuk membangun sebuah tolak ukur performa awal dari setiap model. Dalam skenario ini, model dilatih dengan *pre-processing* data yang minimal untuk melihat kinerja dasar dari setiap algoritma pada dataset asli. Tujuan utamanya adalah untuk mengukur performa model sebelum penerapan teknik-teknik yang lebih kompleks seperti *feature engineering*, penanganan ketidakseimbangan data, dan *hyperparameter tuning*.

Pendekatan *pre-processing* disesuaikan dengan jenis model yang digunakan:

* **Untuk model berbasis *gradient boosting* (CatBoost, XGBoost, LightGBM):** Nilai yang hilang pada kolom kategorikal diisi dengan *string* 'missing'. Hal ini didasarkan pada kemampuan internal model-model ini yang memang dirancang untuk menangani nilai hilang secara efektif tanpa memerlukan imputasi yang kompleks.
* **Untuk model lainnya (SVM, Logistic Regression, Random Forest):**   
  Nilai yang hilang pada kolom numerik diisi dengan nilai median dari masing-masing kolom. Penggunaan median lebih robust terhadap *outlier* dibandingkan *mean*. Sedangkan untuk kolom kategorikal di-*encode* menggunakan *Label Encoding*. Metode ini mengubah setiap kategori unik menjadi bilangan bulat

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | Accuracy | ROC AUC |
| LightGBM | 0.4146 | 0.4997 | 0.4532 | 0.8288 | 0.6263 |
| XGBoost | 0.5815 | 0.5009 | 0.4563 | 0.8288 | 0.5500 |
| CatBoost | 0.9149 | 0.5016 | 0.4567 | 0.8299 | 0.6912 |
| Logistic Regression | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.6293 |
| SVM | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.4132 |
| Random Forest | 0.4146 | 0.4997 | 0.4532 | 0.8288 | 0.5968 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | True Positive | False Negative | False Positive | True Negative |
| LightGBM | 0 | 315 | 1 | 1530 |
| XGBoost | 1 | 314 | 2 | 1529 |
| CatBoost | 1 | 314 | 0 | 1531 |
| Logistic Regression | 0 | 315 | 0 | 1531 |
| SVM | 0 | 315 | 0 | 1531 |
| Random Forest | 0 | 315 | 1 | 1530 |

### **Skenario 2: Feature Engineering dan Preprocessing**

Skenario ini menguji performa model setelah melalui proses feature engineering dan preprocessing secara lanjut. Tujuannya adalah untuk melihat apakah penambahan fitur-fitur informatif seperti indeks\_prestasi, nilai rata-rata, dan encoding yang lebih canggih dapat meningkatkan kemampuan model dalam memprediksi status kelulusan dibandingkan dengan skenario baseline. Berikut adalah tabel hasil evaluasi dari setiap model pada data uji:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | Accuracy | ROC AUC |
| LightGBM | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.6203 |
| XGBoost | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.5260 |
| CatBoost | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.6359 |
| Logistic Regression | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.5845 |
| SVM | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.4297 |
| Random Forest | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.6176 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | True Positive | False Negative | False Positive | True Negative |
| LightGBM | 0 | 315 | 0 | 1531 |
| XGBoost | 0 | 315 | 0 | 1531 |
| CatBoost | 0 | 315 | 0 | 1531 |
| Logistic Regression | 0 | 315 | 0 | 1531 |
| SVM | 0 | 315 | 0 | 1531 |
| Random Forest | 0 | 315 | 0 | 1531 |

### **Skenario 3: Handling Imbalance data**

Pada Skenario ini akan mengatasi masalah dari ketidakseimbangan kelas dalam dataset, di mana jumlah sampel dari satu kelas (‘'tidak lulus') jauh lebih besar dibandingkan dengan kelas lainnya ('lulus'). Dua metode utama diterapkan dan dievaluasi:

* **Oversampling (SMOTE - Synthetic Minority Over-sampling Technique):** Metode ini bertujuan untuk meningkatkan jumlah sampel dari kelas minoritas dengan menghasilkan data sintetis. Sampel sintetis kemudian dibuat di sepanjang garis segmen yang menghubungkan sampel minoritas asli dengan salah satu tetangga terdekatnya. Tujuannya adalah untuk memberikan representasi yang lebih seimbang bagi kelas minoritas dalam data pelatihan tanpa hanya menduplikasi sampel yang ada, sehingga model dapat mempelajari pola kelas minoritas dengan lebih baik.
* **Undersampling:** Metode ini bekerja dengan mengurangi jumlah sampel dari kelas mayoritas. Tujuannya adalah untuk menyeimbangkan distribusi kelas dengan menghapus sebagian data dari kelas yang lebih banyak. Pendekatan ini dapat bervariasi, mulai dari penghapusan acak hingga metode yang lebih canggih yang mencoba mempertahankan informasi penting dari kelas mayoritas.

Tabel Model dengan SMOTE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | Accuracy | ROC AUC |
| LightGBM | 0.5372 | 0.5642 | 0.5052 | 0.5867 | 0.5884 |
| XGBoost | 0.5351 | 0.5123 | 0.5000 | 0.8017 | 0.5500 |
| CatBoost | 0.0853 | 0.5000 | 0.1458 | 0.1706 | 0.5429 |
| Logistic Regression | 0.5419 | 0.5378 | 0.5393 | 0.7519 | 0.5473 |
| SVM | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.5964 |
| Random Forest | 0.5390 | 0.5370 | 0.2968 | 0.2969 | 0.5900 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | True Positive | False Negative | False Positive | True Negative |
| LightGBM | 167 | 148 | 615 | 916 |
| XGBoost | 73 | 292 | 74 | 1457 |
| CatBoost | 315 | 0 | 1351 | 0 |
| Logistic Regression | 67 | 248 | 210 | 1341 |
| SVM | 0 | 315 | 0 | 1531 |
| Random Forest | 284 | 31 | 1267 | 264 |

Tabel Model dengan Undersampling

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | Accuracy | ROC AUC |
| LightGBM | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.5000 |
| XGBoost | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.5000 |
| CatBoost | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.5000 |
| Logistic Regression | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.5000 |
| SVM | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.5000 |
| Random Forest | 0.4147 | 0.5000 | 0.4534 | 0.8294 | 0.5000 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | True Positive | False Negative | False Positive | True Negative |
| LightGBM | 0 | 315 | 0 | 1531 |
| XGBoost | 0 | 315 | 0 | 1531 |
| CatBoost | 0 | 315 | 0 | 1531 |
| Logistic Regression | 0 | 315 | 0 | 1531 |
| SVM | 0 | 315 | 0 | 1531 |
| Random Forest | 0 | 315 | 0 | 1531 |

### **Skenario 4: Tuning Hyperparameter dan Ensemble Model**

Pada metode ini, dilakukan penyesuaian atau tuning pada hyperparamter dari setiap model. Tentunya setiap model memiliki hyperparameter yang berbeda-beda, parameter inilah yang sebenarnya mempengaruhi hasil kinerja dari model. Metode ini ditujukan untuk menentukan jawaban atau hasil terbaik dari sebuah model dengan meng*brute force* hyperparameternya dengan Randomized Search sebanyak *n* kombinasi.

Dari hasil model terbaik, dilakukan Ensemble Learning untuk meningkatkan kemampuan prediksi dari model. Teknik ini menggabungkan beberapa model untuk membuat prediksi yang lebih baik daripada jika menggunakan satu model saja. Pada penelitian ini, logistic regression, LightGBM, dan XGBoost memiliki kemampuan prediksi yang baik, maka dilakukan ensembling beberapa campuran pada ketiga model tersebut.

Tabel Model dengan Hyperparameter Tuning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | Accuracy | ROC AUC |
| LightGBM | 0.5342 | 0.5593 | 0.4512 | 0.4886 | 0.5946 |
| XGBoost | 0.4146 | 0.4997 | 0.4532 | 0.8288 | 0.5416 |
| CatBoost | 0.0853 | 0.5000 | 0.1458 | 0.1706 | 0.4940 |
| Logistic Regression | 0.0853 | 0.5000 | 0.1458 | 0.1706 | 0.5060 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | True Positive | False Negative | False Positive | True Negative |
| LightGBM | 210 | 105 | 839 | 692 |
| XGBoost | 0 | 315 | 1 | 1530 |
| CatBoost | 315 | 0 | 1531 | 0 |
| Logistic Regression | 315 | 0 | 1531 | 0 |

Tabel Model Ensemble Learning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | Accuracy | ROC AUC |
| LightGBM + Logistic Regression Soft | 0.5337 | 0.5339 | 0.5338 | 0.7351 | 0.5763 |
| XGBoost + Logistic Regression Soft | 0.5326 | 0.5339 | 0.5332 | 0.7308 | 0.5758 |
| LightGBM + Logistic Regression Hard | 0.5258 | 0.5271 | 0.3054 | 0.3055 |  |
| XGBoost + Logistic Regression Hard | 0.5340 | 0.5232 | 0.2522 | 0.2551 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | True Positive | False Negative | False Positive | True Negative |
| LightGBM + Logistic Regression Soft | 72 | 243 | 246 | 1285 |
| XGBoost + Logistic Regression Soft | 74 | 241 | 256 | 1275 |
| LightGBM + Logistic Regression Hard | 273 | 43 | 1239 | 292 |
| XGBoost + Logistic Regression Hard | 293 | 22 | 1353 | 178 |

## Pembahasan

Berisi analisis hasil setiap scenario dan insight/temuan yang diperoleh. Dibahas dan dianalisis juga hasil yang gagal diprediksi/misklasifikasi

### **Skenario 1**

Dapat dilihat bahwa hasil *training* dan *testing* dari model-model yang dilatih tanpa *preprocessing* yang ekstensif (minimal) menunjukkan pola yang jelas: meskipun beberapa model seperti CatBoost dan XGBoost menunjukkan F1-score yang cukup baik untuk kelas 'Tidak Lulus' (mayoritas), performa prediksi untuk kelas 'Lulus' (minoritas) sangat buruk pada semua model. Pada data test, sebagian besar, atau bahkan semua, data diprediksi sebagai 'Tidak Lulus'.

Hal ini kemungkinan besar terjadi karena ketidakseimbangan kelas yang parah pada data asli, di mana jumlah data 'Lulus' jauh lebih sedikit dibandingkan data 'Tidak Lulus'. Model yang dilatih pada data yang sangat tidak seimbang cenderung bias terhadap kelas mayoritas, karena model belajar untuk meminimalkan kesalahan keseluruhan dengan memprediksi kelas mayoritas untuk sebagian besar kasus. Akibatnya, model kesulitan mengenali pola yang spesifik untuk kelas minoritas ('Lulus') dan gagal memprediksi *instance* dari kelas ini dengan benar pada data test yang tidak seimbang. Metrik seperti presisi dan *recall* untuk kelas 'Lulus' sangat rendah, dan pada beberapa model, *true positive* untuk kelas 'Lulus' bahkan nol.

### **Skenario 2**

Setelah melakukan berbagai penanganan missing values pada berbagai kolom, one-hot encoding menggunakan beberapa kolom serta feature engineering yang cukup signifikan pada prestasi dan nilai, dapat dilihat bahwa feature engineering dan preprocessing sendiri tidak cukup untuk mendorong presisi lulus, mayoritas model memprediksi semua ke ‘Tidak Lulus’ pada test dataset yang telah diberikan dan tidak ada satupun prediksi ‘Lulus’.

Hal ini mungkin disebabkan oleh ketidakseimbangan data yang mendorong model cenderung menghadapi kelas mayoritas seperti pada Skenario 1. Walaupun F1-score yang dimiliki model cukup tinggi, hal itu disebabkan karena dataset test yang diberikan mayoritas adalah ‘Tidak Lulus’, oleh karena itu model berhasil memprediksi semua yang ‘Tidak Lulus’ dan akurasi kelas ‘Tidak Lulus’ menjadi tinggi dan menyebabkan F1-score tinggi. Maka, diperlukan metode untuk mengatasi ketidakseimbangan agar model bisa mengenali pola dan memprediksi hasil lulus untuk dataset test.

### **Skenario 3**

Dari hasil pengujian yang dilakukan, dapat diamati bahwa hasil test dari model-model yang dilatih setelah menerapkan undersampling menunjukkan bahwa semua model, semua data test diprediksi tidak lulus. Kejadian ini kemungkinan besar terjadi karena ketidakseimbangan kelas yang parah pada data asli, dikombinasikan dengan kekurangan utama undersampling pada dataset yang relatif kecil. Undersampling secara drastis mengurangi jumlah data pelatihan yang tersedia, membuat model dilatih pada dataset yang sangat kecil dan berpotensi kehilangan informasi berharga dari kelas mayoritas. Akibatnya, model kesulitan mempelajari pola yang cukup untuk memprediksi kelas minoritas ('Lulus') dengan benar pada data test yang tidak seimbang. Dalam kondisi ini, model cenderung mengklasifikasikan semua *instance* ke dalam kelas mayoritas sebagai strategi "aman" karena kurangnya informasi yang cukup untuk membedakan.

Dari Skenario kedua ini dapat disimpulkan bahwa oversampling dalam hal ini menggunakan SMOTE terbukti lebih efektif dibandingkan undersampling. Undersampling, dengan mengurangi jumlah data pelatihan secara signifikan, mempersulit model untuk mempelajari pola yang memadai, terutama untuk kelas minoritas. Sebaliknya, SMOTE meningkatkan jumlah sampel kelas minoritas dengan menghasilkan data sintetis, sehingga model dilatih pada dataset yang lebih seimbang tanpa kehilangan data berharga dari kelas mayoritas. Hal ini menghasilkan performa prediksi yang lebih baik, terutama dalam mengidentifikasi kelas minoritas, karena model memiliki representasi yang lebih kaya dan seimbang dari kedua kelas selama pelatihan.

### **Skenario 4**

Menurut hasil pengujian, beberapa model menunjukkan performa yang cukup signifikan setelah dilakukan tuning pada hyperparameter. Metode yang digunakan adalah GridSearch berdasarkan hyperparameter yang mengubah hasil secara signifikan berasal dari penilitian terkait. Oleh sebab itu, beberapa model menunjukkan performa yang justru menurun dibandingkan dengan parameter default.

Setelah melakukan ensemble learning, terbukti bahwa voting metode dengan metode soft terbukti memiliki hasil yang lebih baik dibandingkan metode hard. Namun, kombinasi yang tepat antara dua model belum ditemukan, karena yang masih dicoba hanya kombinasi dari 3 model saja. Oleh sebab itu, hasil dari ensembling masih belum bisa maksimal. Walaupun begitu, ensembling dengan metode soft menunjukkan peningkatan performa model dibandingkan saat model dijalankan secara individu. Sebaliknya, dengan metode hard, model justru menunjukkan penurunan performa.

# KESIMPULAN

Berdasarkan analisis yang telah dilakukan, penelitian ini menyimpulkan bahwa inti permasalahan dari dataset ini merupakan imbalanced data atau pembagian kelas hasil yang tidak seimbang, dengan persentase positive(Lulus) 9.1% dan negatif(Tidak Lulus) 90.9%. Hal ini mengakibatkan model tidak mampu menentukan pola yang membedakan Lulus dan Tidak Lulus dikarenakan jumlah Tidak Lulus yang terlalu banyak. Walaupun hasilnya nampak cukup baik, dengan akurasi 83%, model tersebut memiliki bias terhadap Tidak Lulus dan tidak berhasil memprediksi kelas Lulus. Hal ini menyebabkan model ini masih kurang akurat dan kurang layak digunakan oleh staf akademika maupun calon pendaftar SNBP.

Penelitian ini membandingkan berbagai skenario feature engineering dan preprocessing pada dataset SNBP. Terdapat pembuatan indeks\_prestasi untuk mengukur prestasi seorang siswa alih-alih tetap menggunakan prestasi dan tingkat sebagai fitur kategorikal. Selain itu, terdapat pengisian missing values pada nilai menggunakan rata-rata semester sebelumnya atau mata pelajaran dengan korelasi tertinggi. Namun, feature engineering dan preprocessing data tersebut tidak cukup untuk meningkatkan performa model. Biarpun telah menggunakan feature engineering, performa model justru tidak kunjung naik. Hal ini membuktikan bahwa diperlukan metode lain untuk meningkatkan performa akurasi model.

Setelah dilakukan uji coba menggunakan SMOTE, performa model meningkat drastis dan lebih stabil. Dari skenario ketiga dapat disimpulkan bahwa oversampling, dalam hal ini menggunakan SMOTE terbukti lebih efektif dibandingkan dengan undersampling. Undersampling, dengan mengurangi jumlah data pelatihan secara signifikan, mempersulit model untuk mempelajari pola yang memadai, terutama untuk kelas minoritas. Sebaliknya, SMOTE meningkatkan jumlah sampel kelas minoritas dengan menghasilkan data sintetis, sehingga model dilatih pada dataset yang lebih seimbang tanpa kehilangan data berharga dari kelas mayoritas.

Penggunaan hyperparameter tuning pada bebrapa model menunjukkan hasil yang cukup baik, contohnya adalah model LightGBM dan juga Random Forest menhalami peningkatan setelah dilakukan tuning. Selain itu, setelah melakukan *ensemble learning*, terbukti bahwa *voting* metode dengan strategi *soft* menunjukkan hasil yang lebih baik dibandingkan metode *hard*. Meskipun demikian, kombinasi optimal antara model-model belum sepenuhnya ditemukan, karena yang masih dicoba hanya kombinasi dari tiga model saja. Oleh sebab itu, hasil dari *ensembling* masih belum bisa dikatakan maksimal.

Secara keseluruhan, penelitian ini menegaskan pentingnya penanganan ketidakseimbangan data sebagai langkah krusial dalam membangun model prediktif yang akurat, terutama untuk kelas minoritas. Meskipun feature engineering dan preprocessing memberikan dasar yang kuat, metode oversampling seperti SMOTE terbukti menjadi kunci peningkatan performa yang signifikan. Optimasi hyperparameter dan penggunaan teknik ensemble lebih lanjut juga menjanjikan, meskipun masih memerlukan eksplorasi lebih dalam untuk mencapai potensi maksimal. Hasil terbaik adalah LightGBM dengan hyperparameter tuning yang memberikan hasil yang lebih seimbang dengan akurasi F1 score 0.4512 dan ROC AUC 0.5946.

DAFTAR PUSTAKA

Abdelhamid, M., & Desai, A. (2024). Balancing the scales: A comprehensive study on tackling class imbalance in binary classification [Preprint]. arXiv <https://doi.org/10.48550/arXiv.2409.19751>

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. <https://doi.org/10.1023/A:1010933404324>

Caballero, J. (2024). *Impact of LightGBM hyperparameters on class imbalance* [Bachelor's thesis, KTH Royal Institute of Technology]. DIVA. <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1900560>

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. <https://doi.org/10.1145/2939672.2939785>

Cortes, C., & Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. <https://doi.org/10.1007/BF00994018>

El aachab, Y., Kaicer, M., & Jouilil, Y. (2023). Binary classification with supervised machine learning: A comparative analysis. \*Applied Mathematics & Information Sciences, 17\*(4), 589–598. <https://doi.org/10.18576/amis/170407>

Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 31* (pp. 6639-6649). Curran Associates, Inc. <https://papers.nips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf>

Ha, T. L. A. N., Nguyen, T. M. T., & Nguyen, T. A. T. (2021, October). A comparative study on predicting student’s performance using machine learning algorithms. In *2021 7th International Conference on the Development of Biomedical Engineering in Vietnam (BME7)* (pp. 305-310). IEEE.

Ilemobayo, J. A. (2024). Hyperparameter Tuning in Machine Learning: A Comprehensive Review. *Journal of Engineering Research and Reports*, *26*(6), 388-395. <https://doi.org/10.9734/jerr/2024/v26i61188> (Diakses tanggal 23 Juni 2025)

Kaensar, C., & Wongnin, W. (2023). Predicting new student performances and identifying important attributes of admission data using machine learning techniques with hyperparameter tuning. EURASIA Journal of Mathematics, Science and Technology Education, 19(12), em2369. <https://doi.org/10.29333/ejmste/13863>

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*. <https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>

LightGBM Team. (n.d.). Parameters tuning. LightGBM Documentation. Diakses pada 23 Juni 2025, dari <https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html>

Muthumari, K., Ananth, P. A. B., & Kumar, T. M. (2022). Predicting Student’s Academic Performance Using a Hybrid Feature Selection Method and Machine Learning Classifiers. In *Intelligent Systems and Computer Technology* (pp. 51-61). Singapore: Springer Nature Singapore.

Panitia Seleksi Nasional Penerimaan Mahasiswa Baru. (n.d.). Informasi umum SNBP. Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. <https://snpmb.bppp.kemdikbud.go.id/snbp/informasi-umum> (Diakses tanggal 21 Juni 2025)

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K.

Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, *14*(1), 6086. <https://doi.org/10.1038/s41598-024-56706-x>

Stoltzfus, J. C. (2011). Logistic regression: A brief primer. *Academic Emergency Medicine*, *18*(10), 1099–1104. <https://doi.org/10.1111/j.1553-2712.2011.01185.x>

The XGBoost developers. (n.d.). Notes on parameter tuning. XGBoost Documentation. Diakses pada 25 Juni 2025, dari <https://xgboost.readthedocs.io/en/stable/tutorials/param_tuning.html>

LAMPIRAN