ANALISIS PREDIKSI PERFORMA AKADEMIK MENGGUNAKAN ALGORITMA DECISION TREE (STUDI KASUS: PRODI S1 SISTEM INFORMASI UNIVERSITAS TELKOM)

Wihda Sifwi Hanid
Fakultas Rekayasa Industri
Telkom University
Bandung, Indonesia
wihdahanid@student.telkomuniversity.ac.id

Oktariani Nurul Pratiwi
Fakultas Rekayasa Industri
Telkom University
Bandung, Indonesia
onurulp@telkomuniversity.ac.id

Irfan Darmawan
Fakultas Rekayasa Industri
Telkom University
Bandung, Indonesia
irfandarmawan@telkomuniversity.ac.id

Abstrak*— Peningkatan kualitas pendidikan dan hasil belajar menjadi tujuan utama sistem pendidikan, termasuk mengetahui performa akademik mahasiswa sejak dini. Jalur seleksi masuk perguruan tinggi terbukti berpengaruh terhadap perbedaan prestasi belajar dan dapat dijadikan faktor dalam memantau performa akademik mahasiswa. Penelitian ini menggunakan algoritma Decision Tree untuk memprediksi performa mahasiswa Prodi S1 Sistem Informasi Universitas Telkom berdasarkan data histori akademik angkatan 2017-2019 yang dikumpulkan melalui sistem informasi akademik resmi, yaitu iGRACIAS. Proses pengolahan data mencakup tahap data preparation, training, dan testing dengan penanganan ketidakseimbangan data menggunakan Synthetic Minority Oversampling Technique (SMOTE). Evaluasi performansi model dilakukan menggunakan confusion matrix untuk mengukur akurasi, presisi, recall, dan f1-score. Dari perhitungan confusion matrix, hasil penelitian menunjukkan bahwa baik model dengan maupun tanpa penanganan SMOTE menghasilkan akurasi 66%, namun terdapat perbedaan pada hasil metrik di kelas "Memuaskan", yaitu recall yang meningkat dari 50% menjadi 67%. Selain itu, evaluasi menggunakan k-fold cross validation menunjukkan perbedaan yang signifikan, dengan akurasi sebelum menggunakan SMOTE sebesar 66%, sementara setelah menggunakan SMOTE, akurasi meningkat menjadi 84%. Penelitian ini juga melakukan deployment dengan membangun sistem input sederhana menggunakan Streamlit untuk memudahkan pengguna dalam memprediksi performa akademik mahasiswa. Penelitian ini memberikan kontribusi dalam mendukung pengambilan keputusan akademik untuk meningkatkan pemahaman terhadap performa mahasiswa.

Kata kunci*— Performa Akademik, Decision Tree, Data Mining

I. PENDAHULUAN

Peningkatan kualitas pendidikan dan hasil belajar merupakan salah satu tujuan utama sistem pendidikan. Penting untuk mengetahui performa akademik siswa sejak dini agar institusi pendidikan dapat memberikan penanganan khusus yang tepat terkait dengan prestasi belajar siswa[1]. Hal ini sejalan dengan tujuan memberikan dukungan yang lebih baik kepada mahasiswa dalam mencapai potensi akademiknya.

Terdapat faktor penentu untuk mengetahui keberhasilan pencapaian mahasiswa, diantaranya faktor eksternal seperti jalur penerimaan mahasiswa. Pada penelitian sebelumnya membuktikan bahwa jalur seleksi masuk perguruan tinggi ternyata memiliki pengaruh terhadap perbedaan prestasi belajar mahasiswa [2]. Hal ini menunjukkan bahwa jalur seleksi dapat menjadi salah satu faktor penting dalam memprediksi performa mahasiswa selama masa studinya. Penelitian lain menekankan bahwa peningkatan kualitas pendidikan serta membantu siswa dalam mencapai tujuan akademik mereka merupakan langkah strategis untuk menciptakan pendidikan yang lebih efektif [3]. Dengan data yang tersedia, pendekatan ini memungkinkan institusi pendidikan untuk mengidentifikasi pola dan faktor yang mempengaruhi keberhasilan siswa, sehingga mendukung pengambilan keputusan yang lebih baik.

Data Mining adalah proses analisis data yang kompleks dan berukuran besar yang dilakukan secara otomatis untuk mendapatkan suatu pola atau tren yang umumnya tidak disadari [4]. Salah satu proses dalam *data mining* adalah penerapan klasifikasi dengan algoritma tertentu [5]. Berdasarkan pernyataan tersebut, maka akan dilakukan proses *data mining* dengan metode klasifikasi menggunakan algoritma *Decision Tree*. Penelitian ini akan memprediksi dengan lebih akurat performa mahasiswa berdasarkan histori nilai dan data demografi mahasiswa.

II. KAJIAN TEORI

A. Performa Akademik

Performa akademik merupakan indikator yang digunakan untuk mengukur keberhasilan belajar seorang siswa atau mahasiswa dalam menyelesaikan tugas-tugas akademiknya. Beberapa faktor seperti pemikiran dan pembelajaran metareflective, motivasi, keterampilan belajar, keterlibatan versus pelepasan, kualitas instruksi, dan status sosial ekonomi berperan penting dalam menentukan performa akademik seorang siswa [6].

Prestasi akademik dapat diukur menggunakan beberapa indikator [7], yaitu:

a. Nilai rapor
b. Indeks Prestasi Akademik
c. Angka kelulusan
d. Predikat kelulusan
e. Waktu tempuh Pendidikan

B. CRISP-DM

CRISP-DM adalah salah satu *framework data mining* yang digunakan secara luas untuk memandu proses analisis data dan pengembangan model. CRISP-DM, yang merupakan singkatan dari *Cross-Industry Standard Process for Data Mining*, menyediakan struktur yang sistematis dan terorganisir untuk mengelola proyek *data mining* [8]. Framework ini terdiri dari enam fase utama:

a. *Business Understanding*: Memahami tujuan proyek dari perspektif bisnis dan mengubahnya menjadi masalah data mining untuk mencapai tujuan yang telah direncanakan.
b. *Data Understanding*: Dimulai dengan mengumpulkan data awal, mendeskripsikan data, mengeksplorasi data, dan memverifikasi kualitas data.
c. *Data Preparation*: Menyiapkan data akhir yang akan digunakan dalam pemodelan, termasuk pembersihan data, transformasi, dan penggabungan data.

d. *Modeling*: Menerapkan teknik pemodelan yang berbeda, menyesuaikan parameter ke nilai optimal, dan memilih model terbaik berdasarkan performa.

e. *Evaluation*: Mengevaluasi model yang telah dibangun untuk memastikan bahwa model tersebut memenuhi tujuan bisnis dan memberikan *insight* yang relevan.

f. *Deployment***:** Mengimplementasikan hasil analisis ke dalam sistem operasional atau proses bisnis, sehingga dapat digunakan oleh pemangku kepentingan.

C. *Data Preprocessing*

Preprocessing data merupakan tahap penting yang sering diabaikan, namun penting dalam proses *data mining*. Proses ini mencakup persiapan data yang melibatkan integrasi, pembersihan, normalisasi, transformasi data, serta pengurangan jumlah data seperti pemilihan fitur, pemilihan instansi, pengubahan bentuk variabel, dan lainnya. Tujuan akhir dari serangkaian kegiatan preprocessing data ini adalah menghasilkan dataset yang dianggap valid dan bermanfaat untuk tahapan data mining selanjutnya [9].

a. *Data Cleansing*

Pembersihan data dilakukan dengan menghilangkan data yang terdapat duplikasi dan memiliki nilai pencilan. Kemudian, untuk mengatasi nilai kosong dalam data, dapat dilakukan beberapa pendekatan. Sebuah cara yang paling umum digunakan adalah dengan mengisi nilai kosong tersebut. Pilihan pengisian dapat bervariasi, antara lain menggunakan nilai rata-rata, median, modus, atau bahkan metode pengisian yang lebih kompleks berdasarkan karakteristik data tertentu.

b. *Data Reduction*

Proses reduksi data mengurangi jumlah data baik dari segi volume maupun jumlah atribut (juga disebut dimensi) atau keduanya, tanpa mengorbankan integritas data asli terkait hasil [10]. Reduksi data dapat dilakukan dengan berbagai metode, seperti teknik pengurangan dimensi, pemilihan fitur, atau teknik agregasi untuk mengurangi kompleksitas data tanpa mengorbankan informasi penting.

c. *Data Transformation*

Transformasi dapat dilakukan dengan menerapkan berbagai metode, seperti perubahan skala, normalisasi, atau fungsi matematis tertentu, untuk memodifikasi struktur atau distribusi data dengan tujuan meningkatkan interpretasi atau memenuhi asumsi tertentu dalam analisis statistik.

d. *Data Integration*

Integrasi atau penggabungan data merupakan tahap analisis dengan banyaknya sumber data yang disatukan bersama untuk menjelaskan keseluruhan data [11]. Integrasi data bisa dilakukan dengan menyatukan dan menggabungkan data yang berasal dari berbagai sumber.

D. Klasifikasi

Klasifikasi adalah langkah pembuatan model dengan mengidentifikasi dan melihat perbedaan objek sesuai dengan kelas data atau konsep dengan maksud untuk memungkinkan prediksi objek yang tidak memiliki label kelas [12]. Klasifikasi juga bisa didefinisikan sebagai sebuah teknik data mining yang mengkategorikan data sesuai dengan hubungan data terhadap sampel [13]. Dalam penelitian lain dijelaskan bahwa klasifikasi didefinisikan seperti suatu pekerjaan yang mengevaluasi objek pada data untuk menentukan sebuah kelas yang sesuai [14].

E. *Decision Tree*

Decision Tree adalah algoritma yang digunakan untuk klasifikasi dan regresi dengan prinsip pemecahan masalah ke dalam sejumlah langkah keputusan. Pohon keputusan dihasilkan melalui proses pemilihan atribut terbaik yang digunakan untuk memecah data menjadi kelompok yang lebih kecil dan lebih homogen [12].

Dalam penerapannya, algoritma *Decision Tree* menggunakan konsep entropi dan *information gain* untuk menentukan atribut terbaik yang akan dijadikan node pembagi pada setiap langka h.

Entropi digunakan untuk mengukur tingkat ketidakpastian atau ketidakteraturan dalam data. Nilai entropi yang rendah menunjukkan bahwa data lebih homogen, sedangkan nilai entropi yang tinggi menunjukkan bahwa data lebih beragam. Rumus entropi dapat dituliskan sebagai:

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

Dimana:

- $E(S)$: Entropi dari dataset S.
- p_i: Proporsi elemen dalam kategori ke-i dari dataset S.
- n: Jumlah kategori dalam dataset.

Information Gain (IG) mengukur seberapa besar pengurangan ketidakpastian setelah data dibagi berdasarkan atribut tertentu. Semakin besar nilai *information gain*, semakin baik atribut tersebut untuk dijadikan node pembagi. Rumus untuk *information gain* adalah:

$$IG(S, A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} E(S_v) \quad (2)$$

Dimana:

- $IG(S, A)$: *Information gain* dari atribut A terhadap dataset S.
- $E(S)$: Entropi dataset S sebelum pemisahan.
- $Values(A)$: Semua nilai unik atribut A.
- S_v: Subset dataset S dengan atribut $A = v$.
- $|S_v|/|S|$: Proporsi elemen dalam subset S_v terhadap dataset S.
- $E(S_v)$: Entropi subset S_v.

F. *Confusion Matrix*

Confusion matrix merupakan suatu tabel yang digunakan untuk menilai performa model dengan melihat perbandingan *output* prediksi model dengan nilai sebenarnya dari data yang telah diuji [1]. *Confusion matrix* memiliki empat kategori, yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Penjelasan dari tiap kategori sebagai berikut:

- *True Positive* (TP) mencerminkan jumlah data yang berhasil diprediksi dengan benar ke dalam kelas positif.
- *False Positive* (FP) berisi jumlah data yang salah diprediksi ke dalam kelas positif.
- *False Negative* (FN) mencerminkan jumlah data yang salah diprediksi ke dalam kelas negatif.
- *True Negative* (TN) berisi jumlah data yang berhasil diprediksi benar ke dalam kelas negatif.

Confusion matrix memberikan gambaran jumlah data dengan prediksi yang tepat dan tidak tepat dari pemodelan. Dengan bantuan *confusion matrix*, peneliti mampu mengukur beberapa metrik evaluasi, yaitu akurasi, presisi, *recall*, dan

F1-score. Berikut adalah rumus dari masing-masing metrik evaluasi:

- Akurasi $= \frac{TP+TN}{TP+FP+TN+FN}$
- Presisi $= \frac{TP}{TP+FP}$
- *Recall* $= \frac{TP}{TP+FN}$
- *F1-Score*$= 2 \times \frac{Presisi \times Recall}{Presisi+Recall}$

G. *K-Fold Cross Validation*

K-fold cross validation adalah teknik validasi yang digunakan untuk mengevaluasi kinerja model dengan cara membagi data menjadi beberapa subset atau "*fold*" [15]. Prosesnya dimulai dengan membagi data menjadi k bagian yang sama besar. Model kemudian dilatih sebanyak k kali, di mana setiap kali, satu bagian data digunakan sebagai data uji (*testing set*) sementara k−1 bagian lainnya digunakan sebagai data latih (*training set*) [16]. Hasil dari setiap iterasi kemudian dirata-rata untuk mendapatkan perkiraan kinerja model yang lebih tepat dan mengurangi variasi yang disebabkan oleh pembagian data yang acak. Metode ini membantu memastikan bahwa model tidak hanya cocok dengan data pelatihan tertentu, tetapi juga memberikan gambaran yang lebih akurat tentang bagaimana model akan bekerja dengan data yang baru dan belum pernah dilihat sebelumnya. *K-fold cross validation* sering dipilih karena kemampuannya untuk memberikan penilaian yang menyeluruh dan dapat diandalkan terhadap model.

H. SMOTE

*Synthetic Minority Over-sampling Technique (*SMOTE) adalah sebuah metode yang dirancang untuk menangani ketidakseimbangan data dalam klasifikasi. Metode ini menciptakan data sintetis untuk kelas minoritas dengan cara melakukan interpolasi antara data-data yang ada, sehingga meningkatkan representasi kelas minoritas tanpa hanya menggandakan data yang sudah ada [17]. SMOTE bekerja dengan menghasilkan data baru yang terletak di antara sampel kelas minoritas yang berdekatan dalam fitur yang tersedia. Teknik ini membantu model pembelajaran mesin mengenali pola dalam kelas minoritas dengan lebih baik sehingga meningkatkan akurasi prediksi. SMOTE telah digunakan secara luas dalam berbagai penelitian untuk meningkatkan performa model klasifikasi pada dataset yang tidak seimbang.

I. *Streamlit*

Streamlit adalah *framework open-source* berbasis Python yang dirancang untuk memudahkan pembuatan aplikasi web interaktif. Dengan sintaks sederhana seperti menulis *script* Python biasa, pengguna dapat membuat *dashboard* tanpa perlu keahlian pengembangan web yang kompleks [18]. Keunggulan utama *Streamlit* adalah kemampuannya dalam pengembangan prototipe yang cepat dan visualisasi interaktif. Selain itu, aplikasi yang dibuat dapat dengan mudah dibangun pada beberapa platform seperti *Streamlit Cloud* dan *Heroku*.

III. METODE

A. Pengumpulan Data

Pengumpulan data akan dilakukan untuk memperoleh data histori akademik mahasiswa prodi S1 Sistem Informasi Universitas Telkom melalui website *iGracias* Universitas Telkom. Data yang diperlukan bersifat sekunder, yang berarti akan diambil dari sumber yang telah ada dan dielaborasi sesuai kebutuhan penelitian.

Data akan diperoleh secara daring (*online*) dari *website* yang merupakan sistem informasi akademik resmi yang menyimpan catatan akademik mahasiswa. Jenis data yang akan dikumpulkan mencakup informasi tentang riwayat akademik mahasiswa, termasuk nilai mata kuliah, sks, semester, dan informasi terkait lainnya. Data tersebut akan diidentifikasi berdasarkan NIM (Nomor Induk Mahasiswa) dan periode akademik tertentu.

B. Pengolahan Data

Proses pengolahan data dibagi menjadi tiga tahap yaitu data preparation, training, dan testing. Dalam proses pengolahan data, langkah awal dilakukan melalui tahap data preparation. Pada tahap ini, data histori akademik mahasiswa dikumpulkan dengan cermat dari sumber resmi yaitu website LAAK FRI Universitas Telkom. Selanjutnya, data dikelompokkan dan atribut yang dianggap relevan ditentukan untuk memastikan data yang akan digunakan dalam analisis. Kemudian dilakukan cleansing untuk membersihkan data dari nilai yang hilang atau tidak valid. Terakhir, data dibagi menjadi dua bagian, yaitu data training dan data testing yang nantinya akan menjadi dasar dalam melatih dan menguji model prediksi.

Kemudian *data training* yang telah dipersiapkan sebelumnya digunakan untuk melatih model prediksi. Pada tahap ini, algoritma *Decision Tree* diterapkan untuk membangun model aturan klasifikasi.

Tahap pengolahan data terakhir adalah *testing*. Model aturan klasifikasi yang telah dibangun diimplementasikan pada *data testing*. Kemudian evaluasi performansi algoritma dilakukan pada data testing untuk mengukur akurasi prediksi secara menyeluruh dan memberikan pemahaman mendalam tentang sejauh mana model dapat diandalkan dalam memprediksi performa akademik mahasiswa.

C. Proses *Deployment*

Pada penelitian ini, sistem input sederhana dikembangkan menggunakan *Streamlit* sebagai antarmuka pengguna berbasis web. Sistem ini dirancang untuk memungkinkan pengguna memasukkan data secara langsung dan memperoleh hasil prediksi dari model yang telah dibangun sebelumnya. Penggunaan *Streamlit* dipilih karena kemudahannya dalam membuat antarmuka yang interaktif dengan sintaks Python yang sederhana serta dukungan terhadap berbagai *library*.

Tujuan dari penerapan sistem ini adalah untuk memudahkan pengujian model oleh pengguna akhir tanpa memerlukan keterampilan teknis yang mendalam. Selain itu, sistem ini juga bertujuan untuk mempercepat proses validasi model dengan menyediakan akses yang lebih praktis dalam menginput data dan melihat hasil prediksi secara *real-time*.

D. Metode Evaluasi

Pada penelitian ini, penulis menggunakan *confusion matrix* untuk menunjukkan keabsahan atau validitas dari proses dan hasil penelitian dalam memprediksi performa mahasiswa menggunakan metode klasifikasi algoritma *Decision Tree*. Untuk mengetahui performa pemodelan dilakukan perhitungan hasil akurasi dan metrik evalusi dengan *confusion matrix* yang terdiri dari empat elemen utama, yaitu *True Positive*, *False Positive*, *True Negative*, dan *False Negative*. Kemudian *K-fold cross validation* juga digunakan

untuk mengetahui lebih baik akurasi dari keseluruhan data dengan membagi data menjadi beberapa bagian (*fold*).

IV. HASIL DAN PEMBAHASAN

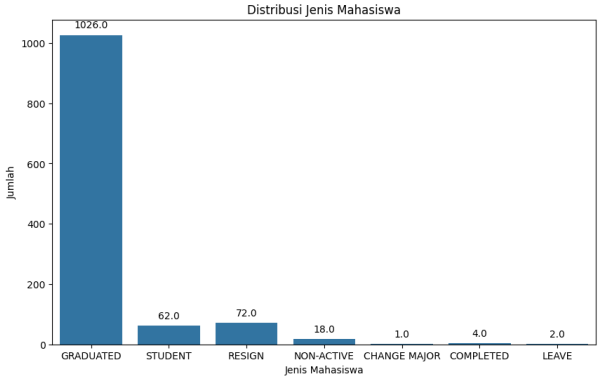
A. *Business Understanding*

Fokus utama dilakukannya penelitian ini adalah untuk memprediksi performa akademik mahasiswa program studi Sistem Informasi di Universitas Telkom dengan bantuan algoritma *Decision Tree*. Maka, tujuan bisnis yang ingin dicapai dengan dilakukannya penelitian ini yaitu:

1. Meningkatkan efisiensi proses bimbingan akademik
2. Optimalisasi penggunaan sumber daya akademik
3. Pengambilan keputusan berdasarkan data
4. Meningkatkan keberhasilan mahasiswa

B. *Data Understanding*

Jumlah keseluruhan data yang diperoleh adalah 1185 data dengan 50 atribut. Penelitian ini memutuskan hanya menggunakan data dengan jenis mahasiswa yang sudah lulus dengan asumsi bahwa atribut "yudisium" akan digunakan sebagai variabel target.



GAMBAR 1
(Distribusi Data Berdasarkan Jenis Mahasiswa)

Dapat dilihat bahwa mahasiswa yang sudah lulus dari keseluruhan data berjumlah 1026 sehingga masih memungkinkan jumlah data layak untuk diolah.

C. *Data Preparation*

Data reduction dilakukan untuk mengurangi ukuran dataset melalui teknik-teknik seperti pemilihan fitur dan penghapusan atribut yang tidak relevan, hal ini akan meningkatkan efisiensi pemrosesan data. Untuk data yang dihapus berasal dari mahasiswa dengan atribut "Jenis Mahasiswa" seperti *Change Major*, *Resign*, *Non-Active*, *Completed*, dan *Leave*. Data tersebut dihapus karena sebagian besar memiliki banyak nilai kosong pada dan jumlahnya juga relatif sedikit. Selain itu, penelitian ini menggunakan atribut "Yudisium" yang mencakup predikat kelulusan, sehingga lebih relevan jika hanya mahasiswa yang sudah lulus saja yang dimasukkan dalam analisis.

TABEL 1
(DATA SETELAH REDUCTION)

Atribut	Deskripsi
TAK	Nilai kumulatif aktivitas non-akademik mahasiswa dengan nilai >=0
Jalur Seleksi	Metode atau jalur yang ditempuh mahasiswa untuk masuk ke program studi, seperti JPU, JPA REGULER, USM, dan lain-lain.
Jenis	Kategori mahasiswa, misalnya reguler, internasional, atau pindahan.
Yudisium	Status kelulusan mahasiswa yang dinyatakan melalui proses yudisium, seperti "Sangat Memuaskan", "Memuaskan", "Dengan Pujian", dan "Tanpa Predikat".
Nilai Mata Kuliah Prasyarat (Atribut SE, Probstat, Alpro, PBO, Desjar, Basdat, RPB, Manjarkom, APSI, MRP, Akuntansi)	Nilai yang diperoleh mahasiswa pada mata kuliah prasyarat yang diperlukan untuk melanjutkan ke mata kuliah berikutnya dengan indeks nilai A, AB, B, BC, C, D, dan E.
IPS 1-4	Indeks Prestasi Semester, angka yang mencerminkan prestasi akademik mahasiswa pada satu semester tertentu dengan rentang nilai 0-4.

Tahapan *data cleansing* akan dilakukan untuk membersihkan data dari kesalahan dan nilai yang hilang, tahap ini akan menghasilkan kualitas data yang optimal. Penghapusan data dibantu menggunakan fungsi data.dropna() pada Python.

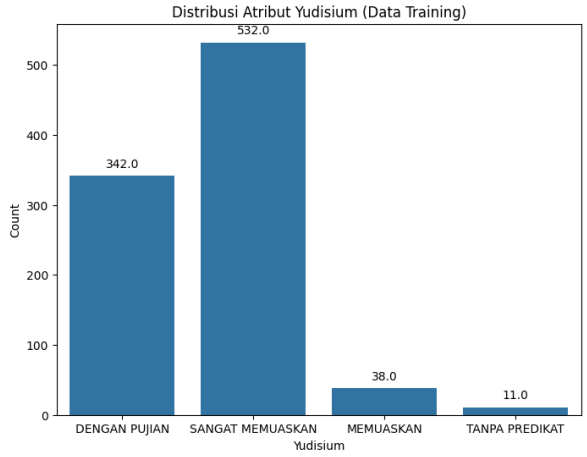
Tahapan *data transformation* dilakukan untuk mengubah data ke dalam format yang lebih sesuai untuk analisis. Pada tahap tersebut dilakukan pengkodean kategori agar siap digunakan dalam tahap pemodelan. Sebelumnya, terdapat data yang bisa dikelompokkan lagi menjadi kelas baru seperti atribut "Jalur Seleksi" karena terdapat jalur yang dibedakan dengan beberapa periode, sehingga pengelompokkan ini dilakukan untuk mempersempit jumlah kelas yang tersedia.

Metode pemilihan fitur dengan entropi dan *information gain* digunakan untuk menentukan fitur yang paling relevan dalam algoritma *Decision Tree*. Entropi mengukur tingkat ketidakpastian atau keragaman dalam dataset, sedangkan *information gain* menghitung pengurangan entropi setelah membagi data berdasarkan suatu fitur. Fitur dengan *information gain* tertinggi dianggap paling informatif dan digunakan sebagai dasar untuk membangun *Decision Tree*. Metode ini memastikan bahwa fitur yang dipilih memiliki kontribusi terbesar dalam memprediksi target. Berikut adalah hasil perhitungan melualui Python.

```
Entropy of the dataset: 1.2482856607835668
Information Gain for Jalur Seleksi: 0.025944811460105965
Information Gain for IPS1: 0.20685285155534627
Information Gain for IPS2: 0.31994789131774637
Information Gain for IPS3: 0.3168056242277183
Information Gain for IPS4: 0.290888554311562
Information Gain for Jenis: 0.015956969743065486
Information Gain for SE: 0.11204421729797587
Information Gain for Probstat: 0.22845406481451058
Information Gain for Alpro: 0.11532496539486603
Information Gain for PBO: 0.10523022727166342
Information Gain for Desjar: 0.11281343423616996
Information Gain for Basdat: 0.17082831127387887
Information Gain for RPB: 0.20992410719084842
Information Gain for Manjarkom: 0.06905209414838875
Information Gain for APSI: 0.1367095695130054
Information Gain for MRP: 0.1563242877244606
Information Gain for Akuntansi: 0.21558653735962752
Information Gain for TAK: 0.014927981273276103
```

GAMBAR 2
(Hasil Entropi Data dan Information Gain Dengan Python)

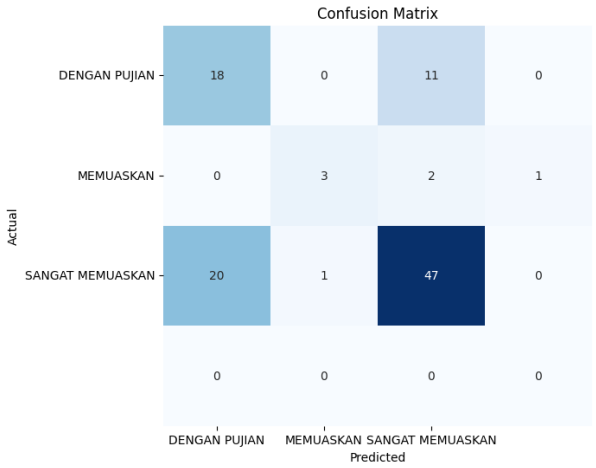
D. *Modeling*



GAMBAR 3
(Dataset Setelah Dilakukan *Oversampling*)

Grafik menunjukkan distribusi data training untuk atribut "Yudisium" sebagai atribut target setelah dilakukan splitting dengan 10% dari dataset akan digunakan untuk menguji, sedangkan 90% untuk sebagai data latih. Grafik menunjukkan bahwa distribusi data tidak merata. Pada kategori Sangat Memuaskan memiliki jumlah terbanyak dengan 532 data, kategori Dengan Pujian sebanyak 342 data, sedangkan kategori Memuaskan dan Tanpa Predikat memperoleh hanya 38 dan 11 data saja. Random state yang digunakan sebesar 42 untuk memastikan bahwa proses pembagian data tetap konsisten setiap kali dijalankan. Kemudian dilakukan *oversampling* dengan menggunakan SMOTE untuk mengatasi masalah ketidakseimbangan kelas pada dataset.

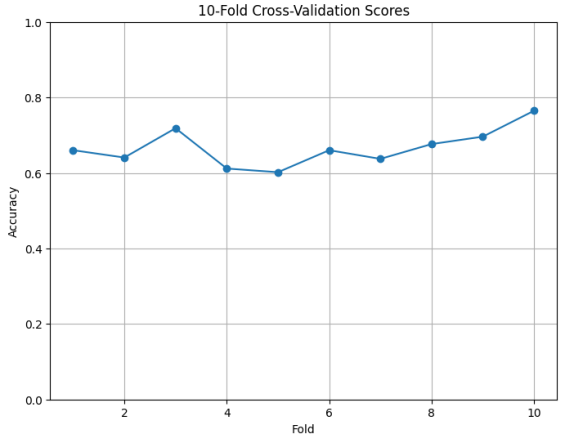
E. *Evaluation*



GAMBAR 4
(*Confusion Matrix* Tanpa *Oversampling*)

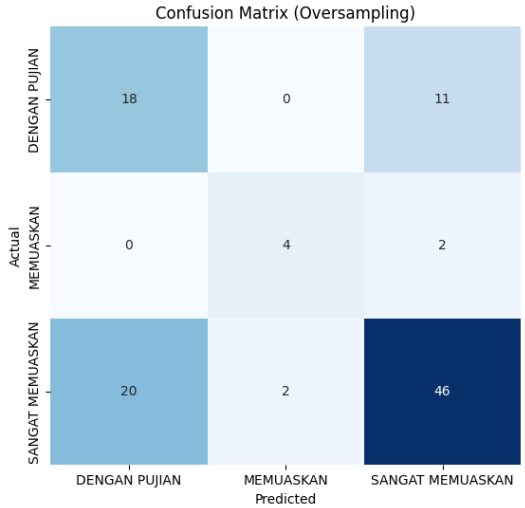
Hasil akurasi untuk pemodelan tanpa *oversampling* adalah 0.66 atau 66%. Terlihat juga hasil metrik evaluasi dari pemodelan terbagi menjadi 4 kelas yang berbeda sesuai dengan variabel target yang digunakan, yaitu "Yudisium" dan juga jumlah data *testing* yang digunakan sebanyak 103 data.

Dari hasil evaluasi, pada kelas *Sangat Memuaskan* yang terdiri dari 68 data, model berhasil memprediksi dengan benar 47 data (TP), sedangkan 21 data diprediksi salah sebagai kelas tersebut (FN). Pada kelas *Memuaskan*, dari 6 data, model hanya berhasil memprediksi 3 data dengan benar (TP), sementara 3 data salah diprediksi sebagai kelas tersebut (FN). Pada kelas *Dengan Pujian* dengan 29 data, model berhasil memprediksi dengan benar 18 data (TP), salah memprediksi 11 data sebagai kelas tersebut (FN). Pada kelas *Tanpa Predikat*, tidak ada data yang diprediksi benar (TP) ataupun salah diprediksi (FN dan FP), sehingga tidak ada perhitungan manual yang dapat dilakukan, dan semua 103 data lainnya diprediksi benar sebagai kelas selain *Tanpa Predikat* (TN).



GAMBAR 5
(*K-Fold Cross Validation* Tanpa *Oversampling*)

Hasil untuk prediksi model tanpa *oversampling* ini secara keseluruhan menunjukkan bahwa performa algoritma *Decision Tree* kurang baik digunakan pada dataset, karena rentang yang dihasilkan hanya berkisar dari 60% hingga 76% saja dengan rata-rata 66%.

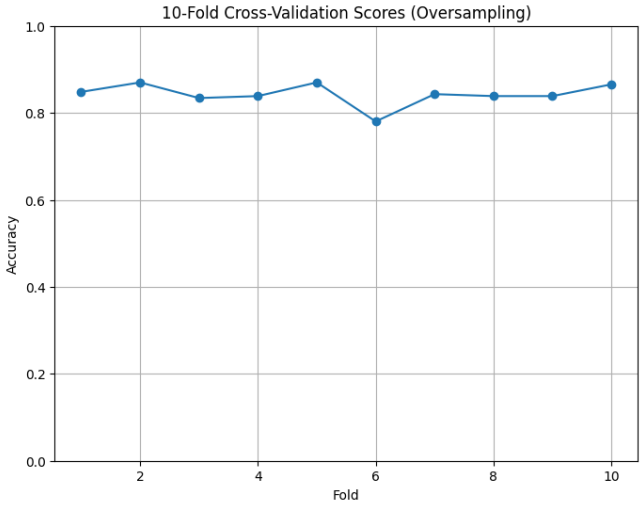


GAMBAR 6
(*Confusion Matrix* Dengan *Oversampling*)

Hasil akurasi untuk pemodelan dengan oversampling adalah 0.6601 atau 66% sama seperti pemodelan tanpa dilakukan *oversampling*. Kemudian sebagian besar hasil perhitungan metrik evaluasi dari pemodelan ini menujukkan hasil yang sama seperti tanpa *oversampling*, namun terdapat beberapa perbedaan hasil perhitungan pada kelas Sangat Memuaskan dan Memuaskan. Pada pemodelan ini juga tidak

ada data testing yang berhasil maupun salah diprediksi sebagai kelas Tanpa Predikat.

Dari hasil evaluasi, pada kelas *Sangat Memuaskan* yang terdiri dari 68 data, model berhasil memprediksi dengan benar 46 data (TP), sedangkan 22 data diprediksi salah sebagai kelas tersebut (FN). Pada kelas *Memuaskan* dengan 6 data, model berhasil memprediksi dengan benar 4 data (TP), sementara 2 data salah diprediksi sebagai kelas tersebut (FN). Pada kelas *Dengan Pujian* yang terdiri dari 29 data, model berhasil memprediksi dengan benar 18 data (TP), salah memprediksi 11 data sebagai kelas tersebut (FN).



GAMBAR 7
(*K-Fold Cross Validation* Dengan *Oversampling*)

Hasil untuk prediksi model dengan *oversampling* ini secara keseluruhan menunjukkan bahwa performa algoritma *Decision Tree* dengan data setelah dilakukan *oversampling* semakin baik. Meningkatkan jumlah sampel dari kelas minoritas membantu model untuk belajar lebih baik tentang karakteristik kelas tersebut. Dengan begitu, model menjadi lebih seimbang dan dapat memprediksi setiap kelas dengan lebih adil yang ditandai dengan hasil akurasi lebih besar, berkisar antara 78% hingga 87% dengan rata-rata keseluruhan yaitu 84%.

F. *Deployment*

Pada tahapan *deployment* dalam penelitian ini, sistem prediksi yang sederhana dirancang untuk memprediksi kategori performa akademik mahasiswa yang akan diidentifikasi dengan kategori serupa kategori dalam atribut yudisium, seperti “Sangat Memuaskan”, “Memuaskan”, “Dengan Pujian” dan “Tanpa Predikat”. Sistem ini menerapkan pemodelan menggunakan algoritma *Decision Tree* yang telah dibuat sebelumnya.

Sistem ini diimplementasikan dalam bentuk aplikasi sederhana menggunakan *Streamlit* yang memungkinkan pengguna untuk memberikan input berupa data mahasiswa, dan kemudian memberikan prediksi mengenai kategori performa akademik mereka. Berikut adalah keterangan input dan contoh hasil dari sistem input sederhana yang berhasil diprediksi.

V. KESIMPULAN

Penelitian ini berhasil menerapkan data mining dengan algoritma klasifikasi *Decision Tree* dalam memprediksi performa akademik mahasiswa Prodi S1 Sistem Informasi Universitas Telkom. Algoritma ini berhasil digunakan untuk menganalisis data histori akademik mahasiswa angkatan 2017-2019 yang memungkinkan identifikasi kategori performa akademik mahasiswa. Meskipun terdapat ketidakseimbangan data, penggunaan teknik SMOTE berhasil meningkatkan performa model secara signifikan.

Hasil akurasi dengan *k-fold cross validation* menunjukkan bahwa model tanpa penanganan ketidakseimbangan data menghasilkan akurasi sebesar 66%. Namun, setelah penerapan SMOTE, akurasi meningkat menjadi 84% yang menandakan pentingnya penanganan ketidakseimbangan data dalam meningkatkan akurasi model. Selain itu, evaluasi terhadap metrik *recall* dengan *confusion matrix* menunjukkan adanya perbaikan di kelas "Memuaskan", dengan hasil yang meningkat dari 50% menjadi 67%, yang menunjukkan perbaikan dalam kemampuan model untuk memprediksi mahasiswa dengan kategori tersebut. dan memantau performa akademik mahasiswa.

REFERENSI

[1] A. Rahman, "Klasifikasi Performa Akademik Siswa Menggunakan Metode Decision Tree dan Naive Bayes," 31 March 2023. [Online]. Available: https://doi.org/10.33020/saintekom.v13i1.349.

[2] F. Dinarti and M. , "ORIENTASI JALUR SELEKSI MASUK PERGURUAN TINGGI TERHADAP PERBEDAAN PRESTASI BELAJAR MAHASISWA ANGKATAN 2012 - 2014 JURUSAN PENDIDIKAN SENI RUPA UNIVERSITAS NEGERI SURABAYA," 2015. [Online]. Available: https://core.ac.uk/download/pdf/230660328.pdf.

[3] E. Alhazmi and A. Sheneamer, "Early Predicting of Students Performance in Higher Education," January 2023. [Online]. Available: https://www.researchgate.net/publication/368965894_Early_Predicting_of_Students_Performance_in_Higher_Education.

[4] I. Pramudiono, Apa Itu Data Mining, Yogyakarta: Penerbit Andi, 2006.

[5] Q. A'yuniyah and M. Reza, "Penerapan Algoritma K-Nearest NeighborUntukKlasifikasi Jurusan Siswa Di Sma Negeri 15 Pekanbaru," 1 March 2023. [Online]. Available: https://journal.irpi.or.id/index.php/ijirse/article/view/484/259.

[6] H. Mason, "Sense of meaning and academic performance: A brief report.," June 2017. [Online]. Available: https://www.researchgate.net/publication/318959674_Mason_HD_2017_Sense_of_meaning_and_academic_performance_A_brief_report_Journal_of_Psychology_in_Africa_273_282-285.

[7] S. Azwar, Sikap Manusia: Teori dan Pengukurannya, Yogyakarta: Pustaka Pelajar, 2013.

[8] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer and R. Wirth, "CRISP-DM 1.0: Step-by-step data mining guide," August 2000. [Online]. Available: https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf.

[9] S. Garcia, J. Luengo and F. Herrera, "Data Preprocessing in Data Mining," 30 Agustus 2014. [Online]. Available: https://books.google.co.id/books/about/Data_Preprocessing_in_Data_Mining.html?id=SbFkBAAAQBAJ&redir_esc=y.

[10] S. Roy, P. Sharma, K. Nath and D. K. Bhattacharyya, "Pre-Processing: A Data Preparation Step," January 2019. [Online]. Available: https://www.researchgate.net/publication/323808183_Pre-Processing_A_Data_Preparation_Step.

[11] M. V. Schneider and R. C. Jimenez, "Teaching the Fundamentals of Biological Data Integration Using Classroom Games," 27 Desember 2012. [Online]. Available: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002789.

[12] J. Han, M. Kamber and J. Pei, "Data Mining: Concepts and Techniques," 2012. [Online]. Available: https://archive.org/details/the-morgan-kaufmann-series-in-data-management-systems-jiawei-han-micheline-kambe/page/n29/mode/2up.

[13] I. Oktanisa and A. A. Supianto, "Perbandingan Teknik Klasifikasi Dalam Data Mining Untuk Bank Direct Marketing," October 2018. [Online]. Available: https://jtiik.ub.ac.id/index.php/jtiik/article/view/958.

[14] E. Prasetyo, Data Mining Konsep dan Aplikasi Menggunakan MATLAB, Yogyakarta: ANDI Yogyakarta, 2012.

[15] G. James, D. Witten, T. Hastie and R. Tibshirani, "An Introduction to Statistical Learning with Applications in R," 2013. [Online]. Available: https://archive.org/details/an-introduction-to-statistical-learning_202202/mode/2up.

[16] M. Kuhn and K. Johnson, "Applied predictive modeling," 2013. [Online]. Available: https://warin.ca/ressources/books/2013_Book_AppliedPredictiveModeling.pdf.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," 1 June 2002. [Online]. Available: https://www.jair.org/index.php/jair/article/view/10302.

[18] Snowflake Inc., "Streamlit documentation," 2025. [Online]. Available: https://docs.streamlit.io/.