

Klasifikasi Data Mahasiswa Lampau Menggunakan Metode *Decision Tree* dan *Support Vector Machine*

Kurniawati^{1,*}, Ririen Kusumawati², Muhammad Ainul Yaqin³

Program Studi Magister Informatika, Universitas Islam Negeri Maulana Malik Ibrahim, Indonesia

¹nia08kurnia@gmail.com; ²ririen.kusumawati@ti.uin-malang.ac.id; ³yaqinov@ti.uin-malang.ac.id;

* penulis korespondensi

INFO ARTIKEL

Sejarah Artikel

Diterima: 10 November 2024
Direvisi: 30 Desember 2024
Diterbitkan: 31 Desember 2024

Kata Kunci

Decision Tree
Klasifikasi Data
Mahasiswa Lampau
Support Vector Machine

ABSTRAK

Penelitian ini bertujuan untuk mengklasifikasikan data mahasiswa guna memprediksi kelulusan di Universitas Islam Negeri Maulana Malik Ibrahim Malang, dengan fokus pada identifikasi faktor-faktor yang memengaruhi keberhasilan kelulusan dan potensi putus kuliah. Data penelitian mencakup atribut akademik seperti Indeks Prestasi Kumulatif (IPK) dan jalur masuk, yang digunakan untuk mengelompokkan mahasiswa ke dalam kategori lulus atau putus. Metode klasifikasi yang diterapkan adalah *Decision Tree* (DT) dan *Support Vector Machine* (SVM). *Decision Tree* bekerja dengan membangun model berbasis pohon keputusan berdasarkan atribut paling berpengaruh, sedangkan SVM menggunakan hyperplane optimal untuk membedakan kategori. Dataset mahasiswa angkatan 2018 dianalisis menggunakan Python dan library scikit-learn. Hasil menunjukkan bahwa *Decision Tree* mencapai akurasi sebesar 96,91%, sedikit lebih tinggi dibandingkan SVM yang mencapai 96,62%. Hasil ini mengindikasikan keunggulan *Decision Tree* dalam memprediksi kategori kelulusan. Penelitian ini berkontribusi pada pemanfaatan atribut akademik sebagai indikator utama dalam klasifikasi data kelulusan serta membandingkan efektivitas dua algoritma pada konteks pendidikan. Dengan temuan ini, universitas dapat mengembangkan strategi evaluasi dan perencanaan yang lebih baik untuk meningkatkan kelulusan mahasiswa, sekaligus memberikan dasar bagi pengembangan model prediksi yang lebih kompleks untuk institusi pendidikan lainnya.

PENDAHULUAN

Kinerja mahasiswa merupakan indikator utama keberhasilan pendidikan tinggi yang mencerminkan berbagai aspek, seperti prestasi akademik, pengembangan pribadi, dan keberhasilan pasca-kelulusan [1]. Salah satu ukuran yang sering digunakan untuk menilai kinerja mahasiswa adalah Indeks Prestasi Kumulatif (IPK), yang menjadi tolok ukur prestasi belajar selama masa studi. Meskipun demikian, IPK bukanlah satu-satunya variabel yang menentukan. Faktor lain seperti jalur masuk, latar belakang akademik sebelumnya, integrasi sosial, dan keterampilan emosional turut memainkan peran penting dalam menentukan keberhasilan mahasiswa [2]. Oleh karena itu, untuk memahami pola dan tren kinerja mahasiswa secara menyeluruh, diperlukan pendekatan berbasis data yang mempertimbangkan berbagai variabel ini. Urgensi pengelolaan data mahasiswa yang terstruktur menjadi semakin nyata ketika institusi pendidikan menghadapi tantangan administratif dan akademik [3]. Tanpa klasifikasi data yang sistematis, universitas dapat mengalami kesulitan dalam melacak perkembangan akademik mahasiswa. Hal ini dapat berdampak pada ketidakmampuan institusi untuk memahami perubahan pola studi, tingkat



kelulusan, atau faktor-faktor yang menyebabkan keterlambatan dan putus kuliah. Selain itu, laporan akademik dan akreditasi yang membutuhkan data mahasiswa yang rapi dan terstruktur menjadi sulit dipenuhi tanpa pengelolaan yang baik. Kondisi ini berisiko menyebabkan hilangnya data historis yang berharga, yang pada akhirnya dapat menghambat perencanaan strategis universitas [4]. Dengan demikian, klasifikasi data mahasiswa lampau menjadi langkah esensial untuk mendukung evaluasi dan pengambilan keputusan yang berbasis data [5].

Penelitian ini dilakukan dalam konteks Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang, yang berusaha meningkatkan tingkat kelulusan mahasiswa serta mengurangi risiko putus kuliah. Data mahasiswa yang digunakan mencakup atribut akademik seperti IPK, jalur masuk, dan informasi terkait lainnya. Tujuan utama penelitian ini adalah untuk mengklasifikasikan data mahasiswa ke dalam kategori kelulusan yang mencakup lulus tepat waktu, terlambat, dan putus kuliah. Dengan menggunakan analisis berbasis data, universitas diharapkan dapat mengidentifikasi pola-pola yang relevan dan memberikan intervensi yang lebih tepat sasaran kepada mahasiswa yang berisiko.

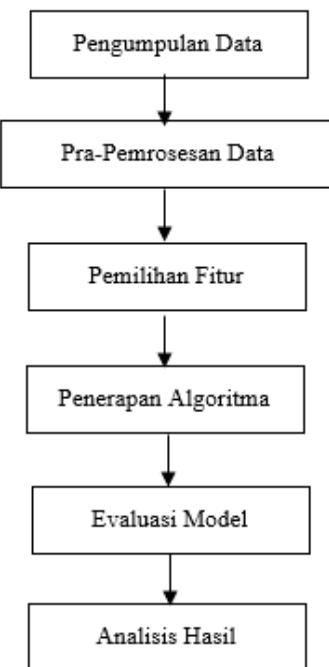
Untuk memenuhi tujuan tersebut, penelitian ini menerapkan algoritma *Decision Tree* (DT) dan *Support Vector Machine* (SVM). *Decision Tree* adalah metode klasifikasi yang membangun model berbasis pohon keputusan. Algoritma ini memiliki keunggulan dalam interpretasi hasil, karena dapat menunjukkan atribut-atribut kunci yang memengaruhi keputusan klasifikasi secara visual [6][7]. Di sisi lain, SVM merupakan algoritma yang kuat dalam menangani data kompleks, karena menggunakan hyperplane untuk memisahkan kelas secara optimal [8][9]. Kombinasi kedua algoritma ini dirancang untuk mengatasi keterbatasan masing-masing metode, dengan memanfaatkan keunggulan interpretasi DT dan keakuratan SVM. Implementasi algoritma dilakukan menggunakan Python dan library scikit-learn, yang memungkinkan pengolahan data secara efisien dan sistematis.

Kontribusi utama penelitian ini terletak pada upayanya untuk mengembangkan model klasifikasi yang tidak hanya akurat tetapi juga mudah diinterpretasikan. Penelitian ini memberikan wawasan baru tentang faktor-faktor risiko yang memengaruhi kelulusan mahasiswa, seperti atribut akademik dan jalur masuk. Selain itu, penelitian ini menghasilkan model klasifikasi yang dapat menjadi alat bagi universitas dalam merancang strategi pembinaan mahasiswa yang lebih efektif. Melalui analisis data, universitas dapat mengidentifikasi mahasiswa yang berisiko gagal lebih awal, sehingga memungkinkan intervensi yang tepat waktu, seperti bimbingan akademik, program pengembangan pribadi, atau bantuan keuangan.

Hasil penelitian ini diharapkan dapat memberikan manfaat praktis bagi UIN Maulana Malik Ibrahim Malang dalam mendukung proses akreditasi, meningkatkan tingkat kelulusan, dan mengurangi risiko putus kuliah. Dengan mengintegrasikan analitik akademik ke dalam pengelolaan data mahasiswa, universitas dapat mengambil keputusan yang lebih baik dalam mengelola sumber daya, merancang kurikulum, dan menyediakan layanan dukungan yang sesuai dengan kebutuhan mahasiswa. Selain itu, penelitian ini juga memberikan kontribusi teoretis dalam bidang data mining pendidikan, dengan menunjukkan bagaimana kombinasi metode klasifikasi dapat diterapkan secara efektif dalam analisis data mahasiswa.

METODE

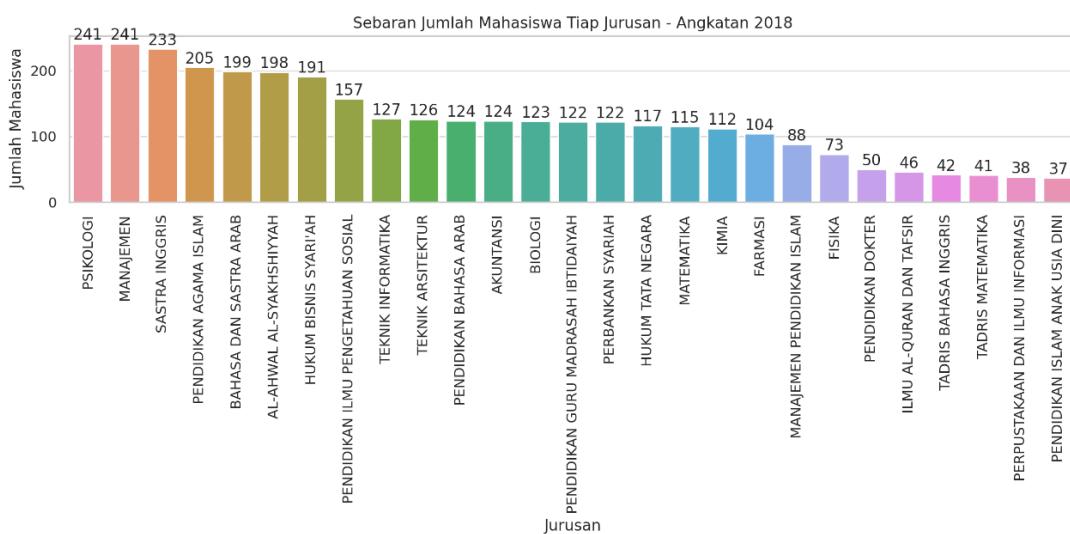
Penelitian ini dilaksanakan di UIN Maulana Malik Ibrahim Malang menggunakan metode *Decision Tree* dan *Support Vector Machine* (SVM) dalam klasifikasi data mahasiswa lampau. Bertujuan untuk mengkategorikan data mahasiswa berdasarkan kelulusan pada pola tertentu. Gambar 1 adalah gambaran umum metode yang digunakan untuk melaksanakan penelitian.



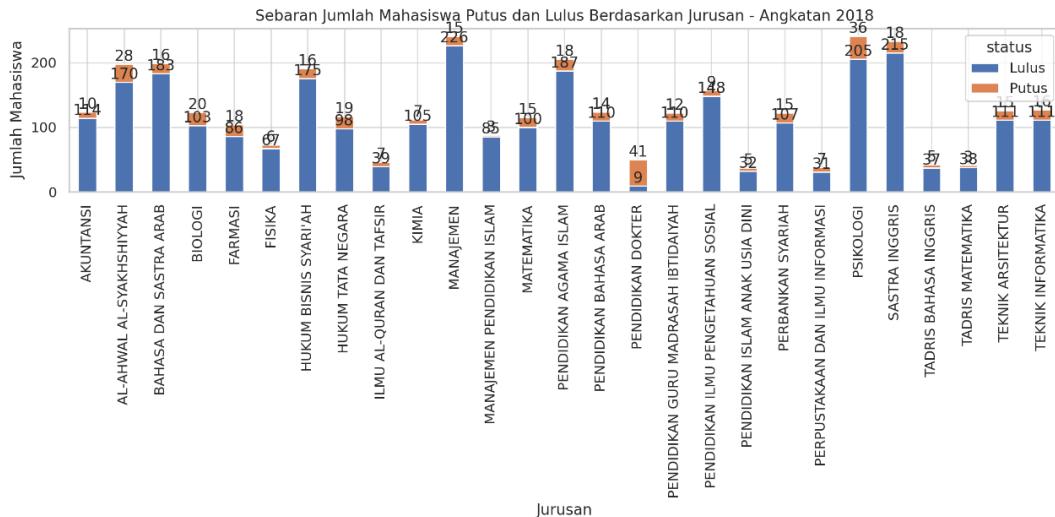
Gambar 1. Alur penelitian

Pengumpulan Dataset

Dataset berisi data 3.396 mahasiswa angkatan 2018 UIN Maliki Malang. Data ini mencakup informasi penting tentang mahasiswa, termasuk NIM, nama fakultas, jurusan, status kelulusan, IPK, dan tahun akademik serta semester. Tujuan utama dari pengumpulan data ini adalah untuk menganalisis perjalanan akademik mahasiswa, yang meliputi informasi tentang status kelulusan dan kemungkinan mahasiswa yang "putus" atau "lulus". Status kelulusan adalah variabel yang sangat penting dalam model klasifikasi yang akan digunakan. Sebaran jumlah mahasiswa berdasarkan jurusan disajikan pada Gambar 2, sedangkan Gambar 3 menunjukkan sebaran jumlah mahasiswa putus dan lulus berdasarkan jurusan.



Gambar 2. Sebaran Jumlah Mahasiswa



Gambar 3. Sebaran jumlah mahasiswa putus dan lulus berdasarkan jurusan

Preprocessing Data

Berikut ini adalah langkah-langkah preprocessing yang dilakukan pada dataset mahasiswa sebelum digunakan dalam model klasifikasi:

Menghubungkan Penyimpanan

Dataset diakses dari Google Drive dengan menghubungkan Google Drive ke lingkungan Colab. Hal ini memungkinkan pengambilan dataset yang disimpan di dalam Google Drive pengguna.

Memuat Dataset

Dataset yang disimpan dalam file Excel dimuat ke dalam sebuah DataFrame untuk diproses lebih lanjut. Pada tahap ini, struktur data ditinjau dengan menampilkan beberapa baris pertama dari dataset untuk memastikan data telah dimuat dengan benar.

Mengurutkan Data

Data diurutkan berdasarkan Nomor Induk Mahasiswa (NIM) secara menaik dan tahun akademik serta semester secara menurun. Langkah ini bertujuan untuk memastikan bahwa data terbaru (IPK dan total SKS terakhir) muncul di baris pertama untuk setiap mahasiswa.

Mengambil Data Terakhir

Setelah data diurutkan, informasi IPK dan SKS terakhir untuk setiap mahasiswa diambil dengan menghapus duplikasi berdasarkan NIM. Hanya data terbaru untuk setiap mahasiswa yang dipertahankan untuk keperluan analisis.

Menghapus Data yang Tidak Lengkap

Baris dengan nilai kosong pada kolom penting seperti status kelulusan, IPK, nama fakultas, nama jurusan, jalur penerimaan, dan SKS total dihapus. Hal ini memastikan bahwa hanya data yang lengkap digunakan dalam analisis lebih lanjut.

Mengonversi Data Kategorikal ke Numerik

Kolom dengan nilai kategorikal, seperti nama fakultas, jurusan, dan jalur penerimaan mahasiswa, dikonversi ke format numerik menggunakan teknik label encoding. Setiap

kategori teks diberikan kode numerik yang unik, sehingga data dapat diproses oleh algoritma machine learning.

Dengan langkah-langkah ini, dataset telah diproses menjadi format yang bersih, lengkap, dan siap digunakan untuk pelatihan model klasifikasi. Proses ini memastikan integritas data dan meminimalkan kemungkinan error selama pelatihan model.

Penerapan *Decision Tree*

Atribut pada dataset yang terpilih menjadi simpul akar, dan dari sana, pohon keputusan akan bercabang berdasarkan nilai-nilai atribut yang dipilih [10]. Proses ini berlanjut secara rekursif, di mana setiap cabang terbagi lagi berdasarkan atribut yang paling informatif hingga mencapai simpul daun (*leaf nodes*), di mana keputusan akhir (klasifikasi) dibuat. Model *Decision Tree* yang sudah dibuat kemudian dapat digunakan untuk melakukan klasifikasi data baru berdasarkan pola yang telah dipelajari dari data pelatihan [11]. Ada beberapa langkah untuk membangun *Decision Tree*. Proses utama adalah memilih atribut yang paling informatif. Untuk menentukan atribut terbaik, kita menggunakan metode, seperti *Information Gain* (berbasis *entropy*). Entropy mengukur ketidakteraturan atau ketidakpastian suatu dataset. Rumus Entropy (E) disajikan pada Persamaan (1)

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

dimana P_i adalah proporsi contoh yang termasuk ke dalam kelas i .

Sementara itu, *Information Gain* mengukur seberapa banyak informasi yang kita dapatkan dengan membagi dataset berdasarkan atribut tertentu. Rumus Information Gain (IG) disajikan pada Persamaan (2).

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

Di mana:

S = dataset awal.

A = atribut yang digunakan untuk membagi.

S_v = subset data setelah pembagian berdasarkan nilai v dari atribut A .

$E(S)$ = entropy dataset awal.

$E(S_v)$ = entropy subset setelah pembagian.

Penerapan *Support Vector Machine*



Gambar 4. Tahapan SVM

Tahapan pada Gambar 4 menjelaskan alur proses penerapan metode *Support Vector Machine* (SVM) dalam klasifikasi data. Tahapan dimulai dengan memvisualisasikan data,

di mana data yang akan digunakan ditampilkan untuk memahami pola distribusinya. Visualisasi ini penting untuk mengetahui apakah data dapat dipisahkan secara linear atau memerlukan pendekatan non-linear, sehingga membantu dalam menentukan kernel yang akan digunakan pada SVM. Tahap berikutnya adalah meminimalkan nilai margin, di mana SVM bekerja dengan mencari margin terbesar yang memisahkan dua kelas. Margin adalah jarak antara hyperplane (garis pemisah) dengan data terdekat dari masing-masing kelas, yang dikenal sebagai support vectors. SVM memaksimalkan margin ini untuk meningkatkan generalisasi model. Setelah margin ditemukan, SVM masuk ke tahap mencari persamaan hyperplane. Hyperplane adalah garis atau bidang yang secara optimal memisahkan dua kelas dalam dataset. Persamaan ini dihitung berdasarkan data pelatihan dengan mempertimbangkan posisi support vectors dan margin yang telah ditentukan. Langkah selanjutnya adalah memvisualisasikan hyperplane, yang bertujuan untuk menunjukkan bagaimana hyperplane membagi data dalam ruang fitur. Visualisasi ini membantu dalam memvalidasi apakah garis pemisah sudah memisahkan data dengan baik. Kemudian, model melakukan pengujian data pada data uji untuk mengevaluasi kemampuan generalisasi model. Pengujian ini memberikan hasil prediksi yang dibandingkan dengan label aktual untuk menghitung metrik kinerja seperti akurasi, precision, dan recall. Tahap terakhir adalah melakukan klasifikasi, di mana model SVM digunakan untuk mengklasifikasikan data baru. Berdasarkan hyperplane yang telah dihitung, model memprediksi kelas data baru dengan menentukan posisi data terhadap hyperplane. Proses ini mengakhiri alur kerja SVM dengan hasil klasifikasi yang siap digunakan untuk analisis lebih lanjut.

Evaluasi Kinerja Algoritma

Confusion Matrix digunakan untuk mengevaluasi kinerja model dalam mengklasifikasikan status kelulusan mahasiswa sebagaimana Tabel 1.

1. True Positive (TP): Mahasiswa yang sebenarnya "Lulus" dan diprediksi dengan benar sebagai "Lulus" oleh model.
2. False Positive (FP): Mahasiswa yang sebenarnya "Putus," tetapi diprediksi sebagai "Lulus" oleh model (kesalahan prediksi).
3. False Negative (FN): Mahasiswa yang sebenarnya "Lulus," tetapi diprediksi sebagai "Putus" oleh model (kelolosan prediksi negatif).
4. True Negative (TN): Mahasiswa yang sebenarnya "Putus" dan diprediksi dengan benar sebagai "Putus."

Tabel 1. *Confusion Matrix*

Prediksi	Lulus (Positive)	Putus (Negative)
Actual Putus	True Negative (TN)	False Positive (FP)
Actual Lulus	False Negative (FN)	True Positive (TP)

Berdasarkan *Confusion Matrix*, model dievaluasi menggunakan metrik berikut:

1. *Accuracy*
Merupakan rasio prediksi Benar (positif dan negatif) dengan keseluruhan data dengan perhitungan, $accuracy = (TP+TN) / (TP+FP+FN+TN)$.
2. *Precision*
Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Perhitungannya adalah $precision = (TP) / (TP+FP)$.
3. *Recall (Sensitivitas)*
Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Recall menjawab pertanyaan dengan hitungan $recall = (TP) / (TP + FN)$.

HASIL DAN PEMBAHASAN

Implementasi *Desicion Tree*

Hasil model klasifikasi menggunakan *Decision Tree* menunjukkan performa yang sangat baik dalam memprediksi status kelulusan mahasiswa. Akurasi model mencapai 94.85% sebagaimana disajikan pada Gambar 5, yang berarti model mampu memprediksi dengan benar hampir seluruh data uji. Tingginya akurasi ini mencerminkan kemampuan model untuk menangani data dan menghasilkan prediksi yang andal. Namun, untuk mendapatkan analisis yang lebih komprehensif, penting untuk meninjau metrik lainnya, seperti precision, recall, F1-score, dan distribusi data pada tiap kelas. Precision untuk kelas 0 (Putus) adalah 0.72, yang menunjukkan bahwa sekitar 72% dari prediksi mahasiswa "Putus" adalah benar. Meskipun nilainya tidak setinggi precision untuk kelas 1 (Lulus) yang mencapai 0.98, ini dapat dimaklumi karena jumlah data kelas 0 jauh lebih kecil dibandingkan kelas 1. Precision yang tinggi pada kelas 1 menunjukkan bahwa model sangat akurat dalam memprediksi mahasiswa yang benar-benar lulus, sehingga memberikan kepercayaan tinggi pada prediksi model untuk kategori ini.

Akurasi Model: 94.85%					
	precision	recall	f1-score	support	
0	0.72	0.83	0.77	70	
1	0.98	0.96	0.97	610	
accuracy			0.95	680	
macro avg	0.85	0.90	0.87	680	
weighted avg	0.95	0.95	0.95	680	

Gambar 5. Evaluasi Algoritma *Decision Tree*

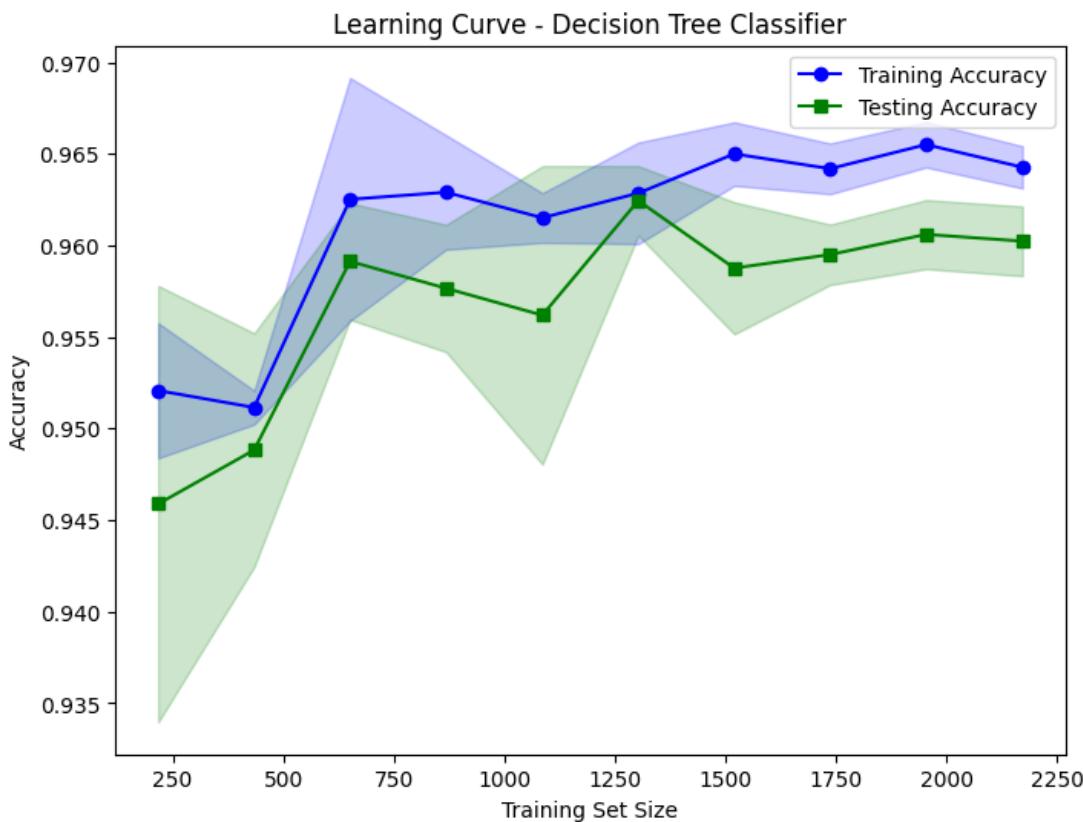
Recall pada kelas 0 (Putus) adalah 0.83, yang berarti model mampu mengidentifikasi 83% mahasiswa yang sebenarnya putus dari seluruh data yang benar-benar "Putus". Sementara itu, recall pada kelas 1 (Lulus) mencapai 0.96, menunjukkan kemampuan model yang hampir sempurna dalam mendeteksi mahasiswa yang benar-benar lulus. Perbedaan recall antara kedua kelas menunjukkan bahwa model lebih efektif dalam mengenali data pada kelas yang memiliki jumlah instance lebih besar. F1-score memberikan gambaran yang lebih seimbang antara precision dan recall. Untuk kelas 0 (Putus), F1-score adalah 0.77, yang cukup baik meskipun precision dan recall pada kelas ini lebih rendah dibandingkan kelas 1. Sebaliknya, F1-score pada kelas 1 adalah 0.97, menegaskan bahwa model sangat andal dalam memprediksi mahasiswa yang lulus. Perbedaan nilai F1-score antara kedua kelas mencerminkan ketidakseimbangan data, di mana kelas 1 mendominasi jumlah instance dalam data uji. Distribusi data yang tidak seimbang terlihat jelas pada metrik support, di mana kelas 0 hanya memiliki 70 instance dibandingkan dengan 610 instance pada kelas 1. Ketidakseimbangan ini dapat memengaruhi performa model, terutama pada kemampuan untuk memprediksi kelas 0 (Putus). Namun, meskipun terdapat ketidakseimbangan data, model tetap mampu mempertahankan performa yang baik pada kedua kelas, yang ditunjukkan oleh nilai-nilai precision, recall, dan F1-score. Macro average yang menghitung rata-rata precision, recall, dan F1-score tanpa mempertimbangkan jumlah instance pada tiap kelas menunjukkan nilai precision 0.85, recall 0.90, dan F1-score 0.87. Ini memberikan gambaran umum performa model tanpa dipengaruhi ketidakseimbangan data. Namun, weighted average yang mempertimbangkan jumlah instance dalam tiap kelas menunjukkan

precision, recall, dan F1-score masing-masing sebesar 0.95. Nilai ini lebih tinggi karena bobot lebih diberikan pada kelas yang lebih besar, yaitu kelas 1 (Lulus).

Best Parameters from Grid Search:				
{'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 10, 'min_samples_split': 2}				
Akurasi Model: 96.91%				
	precision	recall	f1-score	support
0	0.89	0.80	0.84	70
1	0.98	0.99	0.98	610
accuracy			0.97	680
macro avg	0.93	0.89	0.91	680
weighted avg	0.97	0.97	0.97	680

Gambar 6. Evaluasi Hasil Tuning Parameter *Decision Tree*

Hasil evaluasi model *Decision Tree* setelah tuning parameter menunjukkan peningkatan kinerja yang signifikan dalam mengklasifikasikan status kelulusan mahasiswa sebagaimana Gambar 6. Parameter terbaik yang dihasilkan dari GridSearchCV menggunakan kriteria 'entropy' untuk menentukan fitur terbaik, tanpa pembatasan kedalaman pohon (max_depth: None), serta menetapkan nilai minimal 10 untuk min_samples_leaf dan 2 untuk min_samples_split. Dengan konfigurasi ini, model mencapai akurasi sebesar 96.91%, yang berarti model mampu memprediksi dengan benar hampir 97% data uji. Dari sisi precision, model menunjukkan performa yang solid dengan nilai 0.89 untuk kelas 0 (mahasiswa yang putus) dan 0.98 untuk kelas 1 (mahasiswa yang lulus). Hal ini menunjukkan bahwa model sangat akurat dalam mengidentifikasi mahasiswa yang lulus, sementara tetap memberikan prediksi yang cukup baik untuk mahasiswa yang putus. Recall untuk kelas 0 adalah 0.80, yang berarti 80% mahasiswa yang benar-benar putus dapat teridentifikasi oleh model. Sebaliknya, recall untuk kelas 1 sangat tinggi, mencapai 0.99, menunjukkan bahwa hampir semua mahasiswa yang benar-benar lulus berhasil terdeteksi dengan benar. F1-score, sebagai ukuran keseimbangan antara precision dan recall, mencapai 0.84 untuk kelas 0 dan 0.98 untuk kelas 1. Nilai ini menunjukkan bahwa model bekerja sangat baik, terutama pada kelas 1 (mahasiswa lulus), yang merupakan mayoritas dalam dataset. Ketidakseimbangan data terlihat jelas dari support, di mana kelas 0 hanya memiliki 70 instance dibandingkan dengan 610 instance pada kelas 1. Namun, model tetap mampu memberikan performa yang baik meskipun menghadapi ketidakseimbangan data. Secara keseluruhan, nilai macro average, yang memberikan rata-rata sederhana untuk precision, recall, dan F1-score tanpa mempertimbangkan ukuran kelas, menunjukkan keseimbangan yang cukup baik antara performa pada kelas 0 dan kelas 1. Sementara itu, weighted average, yang memperhitungkan ukuran masing-masing kelas, menunjukkan nilai yang sangat tinggi (0.97) untuk precision, recall, dan F1-score. Ini menunjukkan bahwa model memberikan hasil yang sangat konsisten untuk seluruh dataset, dengan kinerja yang sangat baik pada kelas mayoritas (kelas 1) tanpa mengorbankan kinerja pada kelas minoritas (kelas 0). Hasil ini menunjukkan bahwa model *Decision Tree* setelah tuning dapat diandalkan untuk mendukung analisis dan pengambilan keputusan terkait status kelulusan mahasiswa.

Gambar 7. Learning Curve *Decision Tree* Setelah Tuning Parameter

Grafik learning curve pada Gambar 7 menunjukkan kinerja model *Decision Tree* setelah dilakukan tuning parameter, dengan hasil yang sangat positif. Kurva ini mencerminkan bagaimana model bereaksi terhadap peningkatan jumlah data pelatihan, serta bagaimana performa model terhadap data pengujian. Dari grafik, terlihat bahwa akurasi pelatihan meningkat secara stabil seiring bertambahnya ukuran dataset, menunjukkan bahwa model dapat mempelajari pola dari data dengan baik. Hal ini mencerminkan kemampuan model untuk menyerap informasi yang relevan dari data pelatihan. Di sisi lain, akurasi pengujian juga menunjukkan peningkatan yang konsisten, menandakan kemampuan model untuk menggeneralisasi ke data yang belum pernah dilihat sebelumnya. Performa ini penting dalam konteks klasifikasi status kelulusan mahasiswa, karena menunjukkan bahwa model tidak hanya bekerja dengan baik pada data pelatihan tetapi juga pada data uji. Kedekatan antara kurva pelatihan dan pengujian menunjukkan tidak adanya gejala overfitting yang signifikan, yang biasanya ditandai dengan jarak yang besar antara kedua kurva tersebut. Sebaliknya, kurva yang hampir sejajar ini menunjukkan bahwa model memiliki keseimbangan yang baik antara bias dan varians, sehingga kinerja model dapat dianggap stabil dan dapat diandalkan.

Implementasi SVM

Hasil evaluasi model *Support Vector Machine* (SVM) sebagaimana disajikan pada Gambar 8 menunjukkan performa yang sangat baik dalam mengklasifikasikan status kelulusan mahasiswa, dengan akurasi keseluruhan mencapai 96.62%. Akurasi yang tinggi ini menandakan bahwa model dapat mengklasifikasikan dengan benar sekitar 97% dari data uji. Meskipun begitu, akurasi saja tidak cukup untuk mengevaluasi kinerja model secara menyeluruh, sehingga diperlukan analisis lebih mendalam melalui metrik precision, recall, dan F1-score. Pada kelas 0, precision mencapai nilai 0.98, yang menunjukkan bahwa ketika model memprediksi mahasiswa sebagai "Putus", 98% dari prediksi tersebut benar. Namun,

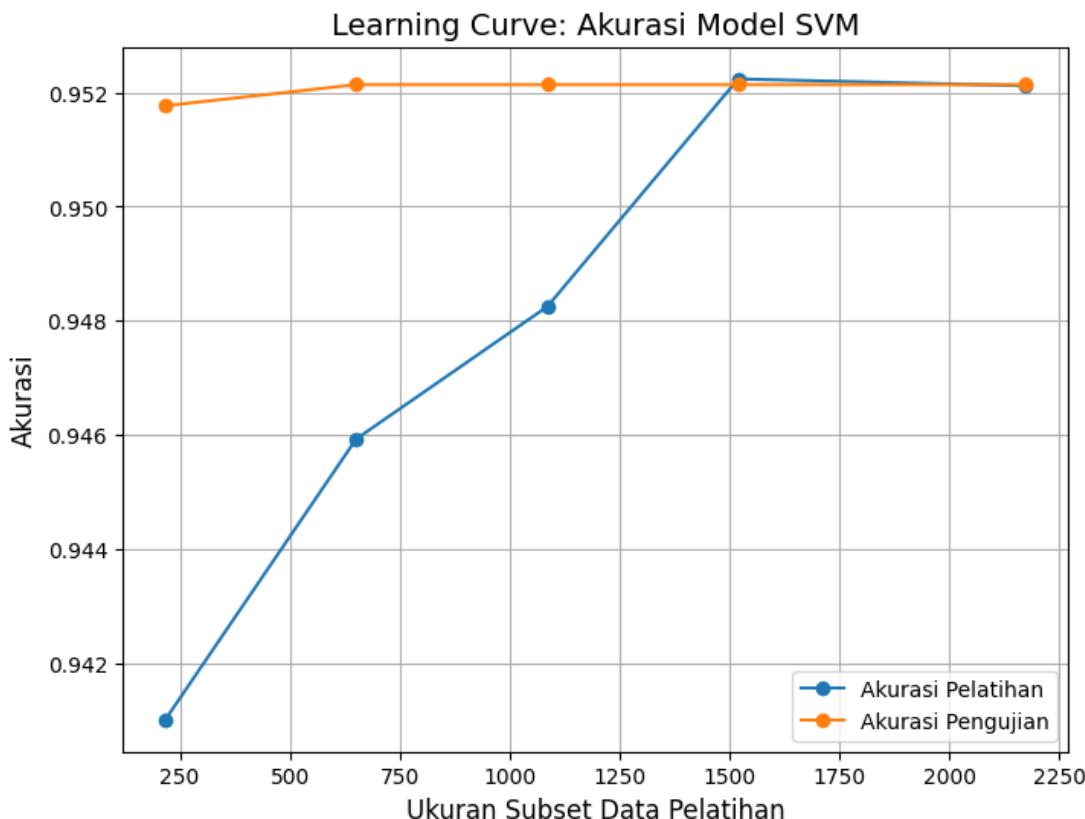
recall pada kelas ini hanya 0.69, yang berarti model hanya mampu mengidentifikasi 69% dari mahasiswa yang benar-benar putus. Hal ini menunjukkan adanya kelemahan dalam menangkap semua kasus mahasiswa yang putus. F1-score untuk kelas ini adalah 0.81, yang mencerminkan keseimbangan antara precision yang sangat tinggi dan recall yang lebih rendah. Hasil ini mengindikasikan bahwa model lebih fokus pada menghindari kesalahan dalam memprediksi mahasiswa sebagai putus, tetapi masih kurang sensitif dalam mendeteksi semua mahasiswa yang benar-benar putus.

Akurasi Model: 96.62%					
	precision	recall	f1-score	support	
0	0.98	0.69	0.81	70	
1	0.97	1.00	0.98	610	
accuracy			0.97	680	
macro avg	0.97	0.84	0.89	680	
weighted avg	0.97	0.97	0.96	680	

Gambar 8. Evaluasi Algoritma SVM

Untuk kelas 1, precision adalah 0.97, yang menunjukkan bahwa hampir semua prediksi "Lulus" oleh model benar-benar berasal dari mahasiswa yang lulus. Recall yang sempurna (1.00) menegaskan bahwa semua mahasiswa yang benar-benar lulus teridentifikasi dengan baik oleh model. F1-score sebesar 0.98 pada kelas ini menunjukkan performa yang sangat baik, mencerminkan keseimbangan antara precision dan recall yang sangat tinggi. Kinerja yang sangat baik pada kelas ini konsisten dengan proporsi data yang didominasi oleh mahasiswa yang lulus. Nilai macro average memberikan gambaran umum kinerja model pada kedua kelas tanpa memperhatikan proporsi data. Precision pada macro average mencapai 0.97, sedangkan recall sedikit lebih rendah pada 0.84. Hal ini menunjukkan bahwa meskipun precision sangat tinggi, model cenderung kurang efektif dalam mendeteksi mahasiswa yang putus dibandingkan mahasiswa yang lulus. F1-score pada macro average adalah 0.89, yang menunjukkan keseimbangan moderat antara kedua kelas. Sebaliknya, weighted average, yang mempertimbangkan proporsi data pada tiap kelas, menunjukkan hasil yang lebih konsisten dengan precision dan recall masing-masing sebesar 0.97. Hal ini wajar mengingat kelas "Lulus" (1) mendominasi data uji, sehingga kinerja yang sangat baik pada kelas ini lebih berpengaruh pada nilai rata-rata tertimbang.

Berdasarkan Gambar 9, terlihat bahwa garis biru (akurasi pelatihan) mengalami peningkatan yang signifikan seiring bertambahnya ukuran subset data pelatihan. Hal ini menunjukkan bahwa model SVM mampu belajar lebih baik saat diberikan lebih banyak data pelatihan. Akurasi pelatihan yang meningkat dengan tajam mengindikasikan bahwa model berhasil mengenali pola-pola dalam data pelatihan dengan efektif, yang berarti bahwa data pelatihan tersebut cukup representatif untuk populasi yang lebih besar. Di sisi lain, garis oranye (akurasi pengujian) menunjukkan pola yang lebih stabil dengan peningkatan bertahap. Stabilitas ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik terhadap data yang belum pernah dilihat sebelumnya. Ketidakaksenjangan yang signifikan antara akurasi pelatihan dan akurasi pengujian menunjukkan bahwa model tidak mengalami overfitting. Dalam kasus ini, model tampaknya mampu menjaga keseimbangan antara mempelajari data pelatihan secara mendalam tanpa kehilangan kemampuannya untuk menggeneralisasi ke data baru.

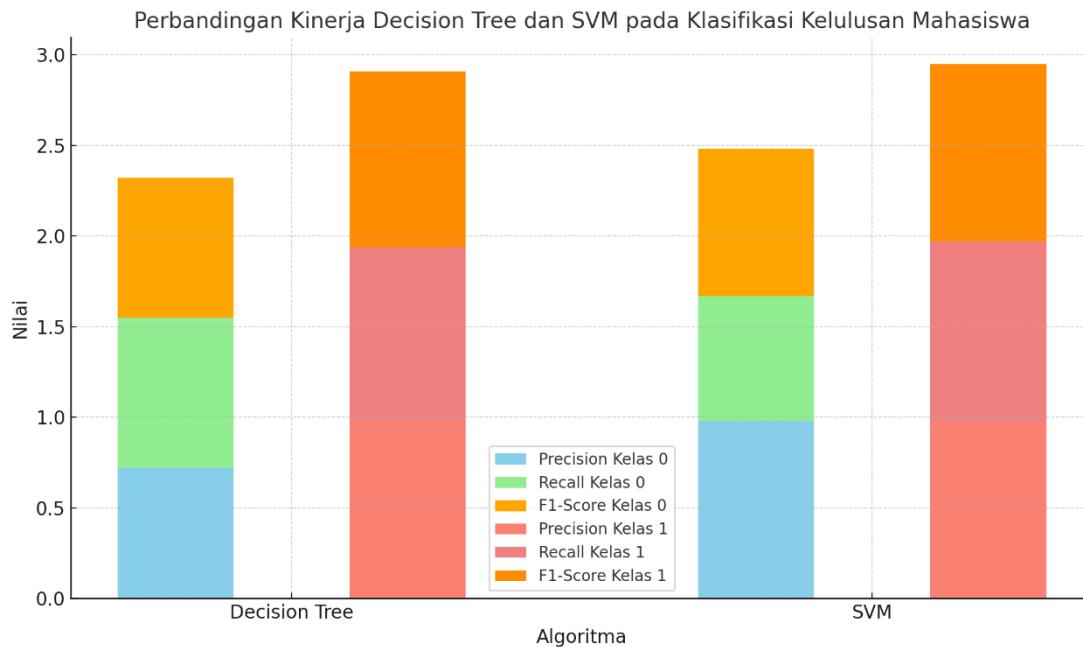


Gambar 9. Learning Curve Evaluasi SVM

Jika model menunjukkan pola di mana akurasi pelatihan meningkat tajam sementara akurasi pengujian tetap stagnan atau meningkat lebih lambat, maka itu akan menjadi tanda adanya overfitting. Namun, dalam grafik ini, akurasi pelatihan dan pengujian bergerak menuju nilai yang hampir sama, yang mengindikasikan bahwa model bekerja dengan baik pada kedua jenis data. Tren ini mencerminkan kemampuan SVM untuk menangkap pola yang relevan tanpa terjebak dalam detail atau noise pada data pelatihan.

Perbandingan kinerja *Decision Tree* dan SVM

Berdasarkan grafik perbandingan pada Gambar 10, kinerja algoritma *Decision Tree* dan SVM menunjukkan hasil yang sangat kompetitif dalam klasifikasi kelulusan mahasiswa, dengan masing-masing algoritma memiliki keunggulan spesifik pada metrik tertentu. Secara keseluruhan, SVM menunjukkan akurasi yang lebih tinggi (96.62%) dibandingkan dengan *Decision Tree* (94.85%), menandakan bahwa SVM sedikit lebih efektif dalam membedakan antara kelas "Lulus" dan "Tidak Lulus". Setelah dilakukan tuning parameter pada *Decision Tree*, kinerja algoritma ini meningkat signifikan, menjadikannya kompetitif dengan SVM dalam klasifikasi kelulusan mahasiswa. Dengan tuning parameter seperti `max_depth`, `min_samples_split`, dan `min_samples_leaf`, akurasi *Decision Tree* meningkat dari 94.85% menjadi 96.91%, bahkan sedikit lebih tinggi dibandingkan akurasi SVM (96.62%). Pada kelas 0 (Tidak Lulus), hasil tuning meningkatkan `precision` *Decision Tree* dari 0.72 menjadi 0.89, dan `recall` tetap tinggi di angka 0.80, menunjukkan keseimbangan yang lebih baik dibandingkan sebelumnya. Meskipun SVM memiliki `precision` lebih tinggi (0.98), `recall` yang rendah pada kelas ini (0.69) menunjukkan bahwa *Decision Tree* lebih andal dalam mendeteksi mahasiswa yang benar-benar tidak lulus.

Gambar 10. Perbandingan kinerja *Decision Tree* dan *SVM*

Pada kelas 1 (Lulus), performa *Decision Tree* yang sudah kuat tetap dipertahankan, dengan precision 0.98 dan recall meningkat menjadi 0.99 setelah tuning, mendekati recall sempurna SVM (1.00). Hal ini menunjukkan bahwa setelah tuning, *Decision Tree* mampu bersaing dengan SVM dalam mendeteksi mahasiswa yang lulus, dengan f1-score yang seimbang. Secara keseluruhan, macro average setelah tuning menunjukkan peningkatan signifikan pada *Decision Tree*, terutama pada precision dan recall, menjadikannya lebih kompetitif dengan SVM. Weighted average juga meningkat, mencerminkan peningkatan kinerja secara keseluruhan. Dengan hasil ini, pemilihan algoritma kini bergantung pada kebutuhan spesifik. *Decision Tree* yang sudah dituning lebih unggul dalam keseimbangan deteksi "Lulus" dan "Tidak Lulus," serta interpretabilitasnya yang tinggi. Sementara itu, SVM tetap menjadi pilihan unggul untuk stabilitas dan generalisasi, terutama pada kelas mayoritas "Lulus."

KESIMPULAN

Berdasarkan hasil penelitian, baik *Decision Tree* maupun *Support Vector Machine* (SVM) menunjukkan performa yang sangat baik dalam klasifikasi kelulusan mahasiswa, dengan peningkatan kinerja setelah tuning parameter pada *Decision Tree*. Setelah tuning, *Decision Tree* mencapai akurasi 96.91%, mendekati SVM yang memiliki akurasi 96.62%. *Decision Tree* menunjukkan peningkatan signifikan pada kelas "tidak lulus" (kelas 0), dengan precision 0.89, recall 0.80, dan f1-score 0.84, dibandingkan sebelum tuning. Pada kelas "lulus" (kelas 1), *Decision Tree* tetap mempertahankan performa tinggi dengan precision 0.98 dan recall 0.99. Sementara itu, SVM tetap unggul dalam precision (0.98) untuk kelas "tidak lulus," namun recall lebih rendah (0.69), menunjukkan keterbatasannya dalam mendeteksi semua mahasiswa yang tidak lulus. Pada kelas "lulus," SVM memiliki recall sempurna (1.00) dan f1-score 0.98, menunjukkan stabilitas dan generalisasi yang sangat baik. Kesimpulannya, setelah tuning parameter, *Decision Tree* mendekati kinerja SVM dalam akurasi keseluruhan dan memberikan keseimbangan yang lebih baik antara precision dan recall pada kedua kelas. Pemilihan model bergantung pada prioritas, dengan

Decision Tree menawarkan interpretabilitas yang lebih tinggi dan *SVM* menawarkan akurasi yang stabil serta kemampuan generalisasi yang unggul.

REFERENSI

- [1] H. Sulistiani, "Penerapan Algoritma Klasifikasi Sebagai Pendukung Keputusan Pemberian Beasiswa Mahasiswa," in *SNTI VI-2018 Universitas Trisakti*, 2018, pp. 300–305. doi: 10.31227/osf.io/yuavj.
- [2] R. M. Debora, D. Mahdiana, M. A. Rony, and H. Hasugian, "Application of C . 45 Algorithm for Potential Students Classification Drop Out at Budi Luhur University," vol. 2, no. 1, pp. 316–325, 2023.
- [3] R. H. Dewi and N. Indrawati, "Rancang Bangun Sistem Pendukung Keputusan Penerimaan Beasiswa Menggunakan Metode Naive Bayes Classifier," *Semin. Nas. ke – 9 Rekayasa Teknol. Ind. dan Inf. Ranc.*, vol. 17, no. 3, pp. 79–84, 2014.
- [4] S. Adi, "Implementasi Algoritma Naive Bayes Classifier Untuk Klasifikasi Penerima Beasiswa PPA Di Universitas Amikom Yogyakarta," *J. Mantik Penusa*, vol. 22, no. 1, pp. 11–16, 2018, [Online]. Available: <http://e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/view/342>
- [5] A. Jaelani, "Klasifikasi Mahasiswa Penerima Beasiswa Dengan Metode Naive Bayes," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.
- [6] B. Charbuty and A. Abdulazeez, "Classification Based on *Decision Tree* Algorithm for Machine Learning," *J. Appl. Sci. Technol. Trends*, vol. 2, no. 01, pp. 20–28, 2021, doi: 10.38094/jastt20165.
- [7] A. C. Fauzan, "Implementasi Algoritma *Decision Tree* Iterative Dichotomiser 3 (ID3) Untuk Prediksi Keberhasilan Pengobatan Penyakit Kutil Menggunakan Cryotherapy Implementation of *Decision Tree* Iterative Dichotomiser 3 (ID3) Algorithm for Predicting the Success of Wa," *J. Bumigora Inf. Technol.*, vol. 4, no. 1, pp. 73–82, 2022, doi: 10.30812/bite.v4i1.1949.
- [8] A. I. Putri, Y. Syarif, P. Jayadi, F. Arrazak, and F. N. Salisah, "Implementasi Algoritma *Decision Tree* dan *Support Vector Machine* (SVM) untuk Prediksi Risiko Stunting pada Keluarga," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 2, pp. 349–357, 2024, doi: 10.57152/malcom.v3i2.1228.
- [9] S. Saikin, S. Fadli, and M. Ashari, "Optimization of *Support Vector Machine* Method Using Feature Selection to Improve Classification Results," *JISA(Jurnal Inform. dan Sains)*, vol. 4, no. 1, pp. 22–27, 2021, doi: 10.31326/jisa.v4i1.881.
- [10] M. R. Anugrah, N. A. Al-Qadr, N. Nazira, and N. Ihza, "Implementation of C4.5 and *Support Vector Machine* (SVM) Algorithm for Classification of Coronary Heart Disease," *Public Res. J. Eng. Data Technol. Comput. Sci.*, vol. 1, no. 1, pp. 20–25, 2023, doi: 10.57152/predatecs.v1i1.805.
- [11] B. Ceballos, M. T. Lamata, and D. A. Pelta, "A comparative analysis of multi-criteria decision-making methods," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 315–322, 2016, doi: 10.1007/s13748-016-0093-1.