MULTIPLE
Journal of Global and Multidisciplinary
https://journal.institercom-edu.org/index.php/multiple
Volume 3 Issue 1 (2025) E-ISSN 2988-7828
INSTITERCOM PUBLISHER

Analisis Dan Perbandingan Algoritma Decision Tree, Naïve Bayes, Dan Random Forest Dalam Evaluasi Prestasi Siswa Smk (Study Kasus: Smkn 8 Kabupaten Tangerang)

Reno Saeprani[1]*, Murni Handayani[2], Taswanda Taryo[3]
[1-3]Universitas Pamulang, Indonesia
Article Info: Accepted: 15 Januari 2025; Approve: 25 Januari 2025; Published: 31 Januari 2025

Abstrak: Evaluasi prestasi siswa merupakan langkah penting dalam mengukur efektivitas pendidikan, terutama di lingkungan Sekolah Menengah Kejuruan (SMK). Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja tiga algoritma machine learning, yaitu Decision Tree, Naïve Bayes, dan Random Forest, dalam evaluasi prestasi siswa SMKN 8 Kabupaten Tangerang menggunakan dua platform pemrosesan data, yaitu RapidMiner dan Python. Algoritma tersebut diuji menggunakan teknik cross-validation dengan skema 5-fold, 7-fold, dan 10-fold, untuk mendapatkan akurasi terbaik dari masing-masing model. Dataset penelitian ini terdiri dari 493 data siswa dari enam jurusan: Teknik Jaringan Komputer dan Telekomunikasi (TJKT), Teknik Bisnis Sepeda Motor (TBSM), Teknik Pemesinan (TPM), Teknik Instalasi Tenaga Listrik (TITL), Otomatisasi dan Tata Kelola Perkantoran (OTKP), serta Akuntansi (AK). Hasil pengujian Cross Validation menunjukkan bahwa algoritma Random Forest consistently memberikan hasil terbaik pada kedua platform dengan akurasi tertinggi sebesar 87.01% pada RapidMiner (7-fold) dan 89.66% pada Python (10-fold). Naïve Bayes menempati posisi kedua dengan akurasi tertinggi 84.99% pada RapidMiner (7-fold) dan 84.18% pada Python (10-fold). Sementara itu, Decision Tree menunjukkan performa terendah, dengan akurasi tertinggi sebesar 72.82% pada RapidMiner (10-fold) dan 76.47% pada Python (5-fold). Temuan ini menunjukkan bahwa Random Forest adalah algoritma yang paling andal untuk mengevaluasi prestasi siswa dengan akurasi yang lebih tinggi dibandingkan dua algoritma lainnya. Selain itu Python menunjukkan kinerja yang lebih baik dibandingkan RapidMiner dalam hal akurasi model. Hasil penelitian ini diharapkan dapat menjadi acuan bagi pihak sekolah dalam memilih metode evaluasi prestasi siswa yang lebih efektif dan efisien, serta memberikan kontribusi signifikan dalam penerapan Data Mining di dunia pendidikan.
Kata Kunci: Prestasi Siswa; Data Mining; Decision Tree; Naïve Bayes; Random Forest.

Correspondence Author: Reno Saeprani
Email: kumumala15@gmail.com

Pendahuluan

Pendidikan berperan penting dalam meningkatkan kualitas siswa, terutama di SMK yang mempersiapkan mereka untuk dunia kerja. Evaluasi prestasi siswa diperlukan untuk memahami potensi, memperbaiki pembelajaran, dan menilai efektivitas pengajaran. Di SMKN 8 Kabupaten Tangerang, dengan enam jurusan dan 493 siswa, evaluasi prestasi mencakup variabel seperti nilai akademik, kehadiran, kegiatan ekstrakurikuler, dan latar belakang sosial-ekonomi. Untuk menganalisis data yang kompleks, algoritma machine learning seperti Decision Tree, Naïve Bayes, dan Random Forest digunakan untuk menemukan pola yang memprediksi prestasi siswa, mendukung kebijakan berbasis data. Penelitian sebelumnya oleh Aldi Sholihin Fauzan dan tim (2024) membandingkan algoritma Decision Tree dan Naïve Bayes untuk mengevaluasi prestasi siswa di SMK Al-Musyawirin. Hasilnya menunjukkan Naïve Bayes memiliki akurasi tertinggi 85,93% dengan 5-fold cross-validation, sedangkan Decision Tree mencapai 100% dengan split

data. Penelitian tersebut menggunakan RapidMiner tetapi belum melibatkan Random Forest (Fauzan., et al 2024).

Penelitian ini memperbarui pendekatan dengan menambahkan algoritma Random Forest, menggunakan Python selain RapidMiner, dan melibatkan data lebih besar dari 493 siswa di enam jurusan. Pengujian dilakukan dengan berbagai skema cross-validation (5-fold, 7-fold, dan 10-fold). Hasilnya menunjukkan Random Forest memberikan akurasi terbaik dibandingkan Decision Tree dan Naïve Bayes. Penelitian ini memberikan panduan bagi sekolah dalam memilih metode evaluasi yang efektif, memanfaatkan machine learning, dan meningkatkan kualitas evaluasi prestasi siswa di SMK.

Kajian Teori

Penelitian ini mengkaji literatur yang relevan terkait penerapan metode Naïve Bayes, Decision Tree, dan Random Forest dalam data mining. Tinjauan ini mencakup landasan teori, penelitian terdahulu, analisis prestasi siswa, penggunaan algoritma machine learning, serta pengolahan data dengan RapidMiner dan Python.

Penelitian yang mendekati adalah karya Aldi Sholihin Fauzan (2024) berjudul *"Analisis Perbandingan Algoritma Decision Tree dan Naïve Bayes untuk Evaluasi Prestasi Belajar Siswa"*. Hasilnya menunjukkan bahwa algoritma Decision Tree lebih unggul dengan akurasi 89,22% dibandingkan Naïve Bayes yang mencapai 85,93%, menggunakan metode cross-validation untuk memastikan keakuratan model.

Metode

1. Data dan Sumber Data

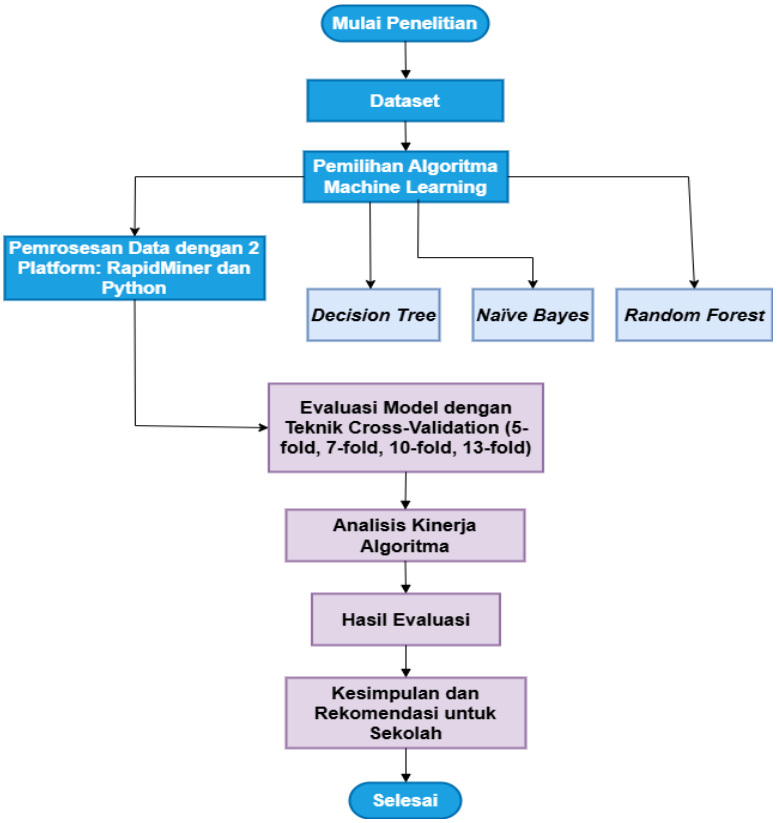
Penelitian ini menggunakan data akademik siswa kelas X SMKN 8 Kabupaten Tangerang tahun ajaran 2023/2024 dari enam program keahlian: TJKT, AKL, OTKP, TITL, TPM, dan TBSM. Dataset terdiri dari 493 entri dengan 23 atribut, termasuk identitas siswa, informasi pribadi, dan nilai mata pelajaran. Analisis bertujuan memahami performa akademik siswa, mengevaluasi kemampuan, dan mendukung pengambilan keputusan untuk pengembangan pembelajaran. Hasil analisis diharapkan memberikan wawasan tentang distribusi nilai, perbedaan antar jurusan, dan tren nilai tertentu. Informasi ini digunakan untuk evaluasi, perencanaan program remedial, pelatihan guru, dan peningkatan kualitas pembelajaran. Dengan data ini, sekolah dapat memberikan intervensi tepat

2. Pengumpulan Data

Pada tahapan ini, peneliti akan menggunakan data siswa SMKN 8 Kabupaten Tangerang yang terdiri dari 493 record dari enam jurusan, yaitu TJKT, TBSM, TPM, TITL, OTKP, dan AK, dengan data yang berasal dari siswa kelas X. Data tersebut memiliki total 15 atribut dan 2 label yang akan digunakan dalam analisis. Data asli sebenarnya memiliki 23 atribut, namun hanya atribut Nama Siswa dan Nilai Mata Pelajaran yang digunakan dalam penelitian ini, sedangkan atribut lain seperti No, Tempat Lahir, Tanggal Lahir, NIS, Nama Orang Tua, NISN, dan Program Keahlian dihapus karena tidak relevan terhadap tujuan analisis. Selanjutnya, data yang telah diseleksi akan diklasifikasikan menggunakan tiga algoritma, yaitu Decision Tree, Naive Bayes, dan Random Forest. Tujuan dari klasifikasi ini adalah untuk mengevaluasi prestasi siswa dengan membandingkan hasil dari ketiga algoritma tersebut guna menentukan metode yang paling efektif dalam menganalisis data prestasi siswa.

3. Peraancangan Penelitian

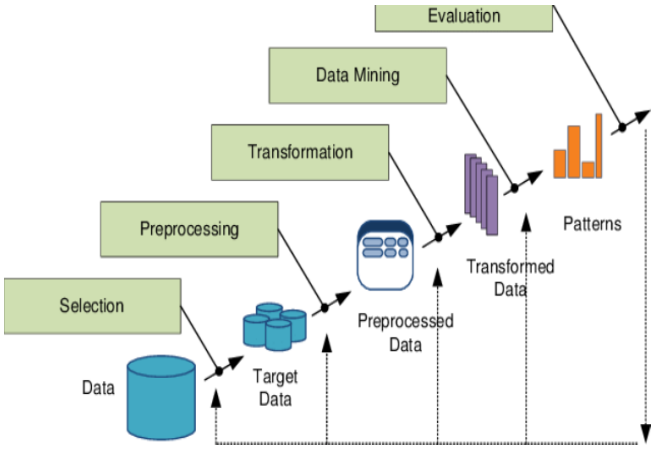
Penelitian ini menerapkan metode data mining menggunakan algoritma Naïve Bayes, Decision Tree, dan Random Forest untuk mengungkap faktor-faktor yang memengaruhi prestasi siswa SMK. Analisis hasil klasifikasi dilakukan guna menentukan algoritma yang paling efektif dalam mengevaluasi dan memprediksi prestasi akademik. Hasil penelitian ini diharapkan dapat menjadi dasar dalam merancang strategi peningkatan prestasi siswa.



Gambar 1. Tahapan Penelitian

4. Pengolahan Data

Penelitian ini menerapkan tahap pengolahan data dengan menggunakan kerangka Knowledge Discovery in Databases (KDD), yang merupakan proses untuk mengidentifikasi pola tersembunyi dalam data. Proses KDD terdiri dari beberapa tahapan yang dapat disesuaikan sesuai dengan kebutuhan penelitian atau proyek tertentu dalam konteks pengolahan data [1].

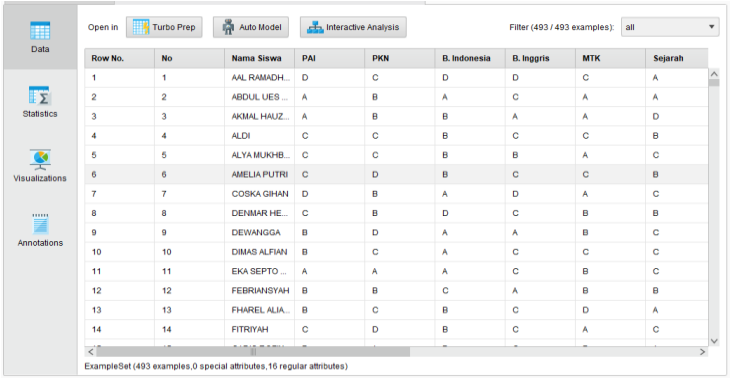


Gambar 2. Tahapan Penelitian

5. Data Selection

Tahap pertama dalam metode Knowledge Discovery in Database (KDD) adalah seleksi data, yang bertujuan memilih data yang relevan untuk proses data mining. Dalam kasus ini, seleksi data dilakukan menggunakan dua tools, yaitu RapidMiner dan Python. Dari data awal yang memiliki 21 atribut, hanya dua atribut yang dipilih, yaitu Nama Siswa dan Nilai Mata Pelajaran. Atribut lainnya, seperti No, Tempat Lahir, Tanggal Lahir, Nomor Induk Sekolah (NIS), Nama Orang Tua, Nomor Induk Siswa Nasional (NISN), dan Program Keahlian, dihapus untuk menyederhanakan data dan menjaga fokus pada analisis yang relevan. Seleksi ini penting untuk meningkatkan efisiensi proses data mining dan memastikan hanya data yang memberikan kontribusi signifikan yang digunakan. Hasil dari proses seleksi ini dapat dilihat pada gambar 3.

Gambar 3 Operator Dataset



Gambar 4. Dataset Read File CSV

6. Data Preprocessing

Tahap kedua dalam metode Knowledge Discovery in Database (KDD) adalah pembersihan data, yang bertujuan memastikan data bebas dari masalah seperti data yang hilang, tidak lengkap, atau tidak valid. Pembersihan data penting untuk menjaga kualitas dataset agar analisis yang dilakukan menghasilkan informasi yang akurat. Dalam kasus ini, dataset yang digunakan telah diperiksa secara menyeluruh dan dipastikan tidak ada data yang hilang atau tidak valid, sehingga proses pembersihan tidak diperlukan.

7. Transformasi Data Label

Tahap selanjutnya dilakukan proses tranformasi data dimana atribut nilai mata pelajaran yang awalnya berbentuk numerik, diubah menjadi kategori Huruf A, B, C, dan D dengan ketentuan nilai yang dapat dilihat pada tabel 1, Serta diberi predikat akhir sebagai hasil nilai

Tabel 1 Ketentuan Katagori Huruf dan Predikat

Nilai Angka	Nilai Huruf	Predikat
90-100	A	Sangat baik
90-80	B	Baik
70-79	C	Cukup
>70	D	Kurang

Nilai Angka	**Predikat Akhir**
83-100	Berprestasi
75-82	Kurang Berprestasi

Dataset dalam penelitian ini dibagi menjadi dua bagian, yaitu 80% untuk data latih (training set) dan 20% untuk data uji (testing set), guna memastikan bahwa evaluasi model dilakukan secara objektif. Pembagian ini menghasilkan sekitar 395 entri untuk data latih dan 98 entri untuk data uji setelah dilakukan pembulatan. Data latih digunakan untuk melatih model agar dapat mempelajari pola dari data, sementara data uji digunakan untuk mengevaluasi performa model pada data baru. Pembagian ini bertujuan untuk memastikan bahwa model yang dihasilkan memiliki kemampuan generalisasi yang baik dan tidak mengalami overfitting terhadap data latih.

8. Metode Model Pengujian

a. Confusion Matrix

Confusion matrix adalah tabel yang digunakan untuk mengukur kinerja model dengan membandingkan prediksi model dengan keadaan sebenarnya dari data. Pengukuran kinerja model klasifikasi melibatkan penggunaan matriks kebingungan (confusion matrix) sebagai dari hasil klasifikasi (Ramadani and Hayadi, 2022).

Keempat istilah yang umum digunakan dalam confusion matrix adalah True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Berikut adalah perhitungan manual pada confusion matrix untuk mengevaluasi kinerja model (Nurrizta, 2022).

Table 2. Confusion Matrix

	Prediksi Berprestasi	**Prediksi Tidak Berprestasi**
Aktual Berprestasi	TP (True Positive)	FN (False Negative)
Aktual Tidak Berprestasi	FP (False Positive)	TN (True Negative)

Pada contoh tabel diatas Confusion Matrix yang digunakan adalah data evaluasi aktual berprestasi dan data aktual tidak berprestasi dengan keterangan sebagai berikut :

- TP *(True Positive)* : Jumlah siswa yang benar-benar berprestasi dan diprediksi berprestasi.
- FN *(False Negative)* : Jumlah siswa yang sebenarnya berprestasi tetapi diprediksi tidak berprestasi.
- FP *(False Positive)* : Jumlah siswa yang tidak berprestasi tetapi diprediksi berprestasi.
- TN *(True Negative)* : Jumlah siswa yang tidak berprestasi dan diprediksi tidak berprestasi.

Rumus untuk menghitung akurasi, recall, dan precision dalam konteks algoritma *Naïve Bayes, Decision Tree*, dan *Random Forest* :

Akurasi *(accuracy)* $$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision (Positive Predictive Value) Presisi = $$\frac{TP}{TP+FP}$$

Recall (Sensitivitas) : $$\text{Recall} = \frac{TP}{TP + FN}$$

b. Cross-Validation

Dalam evaluasi model ini menggunakan metode cross-validation untuk mengukur kinerja model dengan lebih akurat. Cross-validation membagi dataset menjadi beberapa bagian atau fold, dan pada setiap iterasi, model dilatih menggunakan sebagian data dan diuji pada bagian yang tersisa. Pendekatan ini membantu mengurangi kemungkinan model mengalami overfitting serta memastikan model memiliki kemampuan generalisasi yang baik pada data baru. Dalam uji kali ini saya akan melakukan cross-validation dengan 5 fold, 7 fold, dan 10 fold, guna memeriksa performa model pada berbagai pengaturan pembagian data. Pengukuran kinerja dilakukan dengan rumus berikut:

- **Accuracy** = $\frac{Prediksi\ Benar}{Total\ Data}$ X 100%

Menghitung proporsi prediksi yang benar dibandingkan dengan total data.

- **Precision** = $\frac{Benar\ (TP)}{Benar\ (TP)+Salah\ Positif\ (FP)}$ X 100%

Mengukur seberapa akurat model dalam memprediksi kelas positif.

- **Recall** = $\frac{Benar\ (TP)}{Benar\ (TP)+Salah\ Negatif\ (FN)}$ X 100%

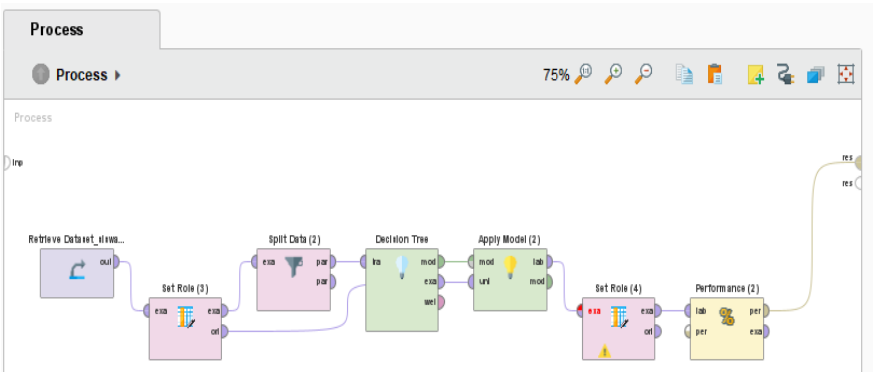
Menilai seberapa baik model dalam mendeteksi semua kelas positif.

Hasil Dan Pembahasan

1. Hasil

a. **Analisis Model Decision Tree**

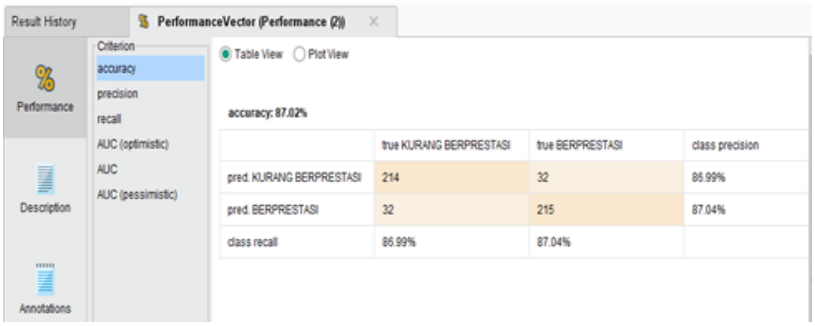
Tahapan selanjutnya adalah pengolahan data menggunakan RapidMiner untuk model Decision Tree, dimulai dari memuat dataset yang berisi 493 record hingga evaluasi model. Proses ini mencakup seleksi data, preprocessing, transformasi, pemisahan data menjadi training dan testing set, serta validasi model secara visual. Evaluasi model dilakukan dengan mengukur akurasi, presisi, recall, dan F1-score untuk menilai kinerjanya. RapidMiner digunakan sebagai alat efisien dan intuitif dalam pengolahan data berbasis Decision Tree, seperti yang terlihat pada Gambar 5.



Gambar 5. Workflow Decision Tree di RapidMiner

Pada penelitian ini data dibagi dengan rasio 80:20, di mana 80% (393 sampel) digunakan untuk pelatihan model, dan 20% (100 sampel) digunakan untuk pengujian. Model dilatih dengan 393 sampel untuk mempelajari pola antara fitur input dan hasil target. Setelah pelatihan, model diuji menggunakan 100 sampel data yang tidak digunakan selama pelatihan untuk mengevaluasi kemampuannya dalam memprediksi data baru. Hasil evaluasi menunjukkan bahwa 215 data yang berlabel "BERPRESTASI" dan 214 data yang berlabel "KURANG BERPRESTASI" berhasil diprediksi dengan benar. Namun, terdapat 32 data "KURANG BERPRESTASI" yang salah diprediksi sebagai "BERPRESTASI" dan 32 data "BERPRESTASI" yang salah diprediksi sebagai "KURANG

BERPRESTASI." Hasil ini memberikan gambaran tentang akurasi dan kesalahan model dalam mengklasifikasikan kinerja siswa, seperti yang ditunjukkan pada Gambar 6.



Gambar 6. Hasil akurasi pada Decision Tree

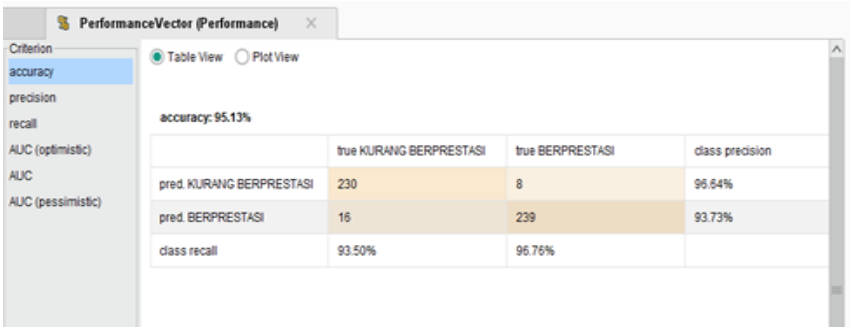
Perhitungan confusion matrix pada model Decision Tree memberikan wawasan tentang performa model dalam memprediksi data. Berikut adalah hasil perhitungan confusion matrix dan berbagai metrik evaluasi

- Akurasi = $\frac{214+215}{493}\frac{429}{493} \approx 0.8702$
  (akurasi model adalah sekitar 87.02%.)
- Presisi = $\frac{215}{215+32} = \frac{215}{247} \approx 0.8704$
  (presisi model untuk kelas positif Berprestasi adalah sekitar 87.04%.)
- Recall = $\frac{215}{215+32} = \frac{215}{247} \approx 0.8704$
  (recall model untuk kelas positif Berprestasi adalah sekitar 87.04%.)

b. Analisis Model Naïve Bayes

Selanjutnya adalah penggunaan pengolahan data Naïve Bayes, dimulai dari memuat dataset berisi 493 record hingga evaluasi model. Pada proses workflow Naïve Bayes sama seperti proses sebelumnya, hanya saja operator algoritma disesuaikan dengan metode Naïve Bayes.

Penelitian ini membagi data dengan proporsi 80:20, di mana 393 dari 493 sampel digunakan untuk pelatihan, dan 100 sisanya untuk pengujian. Model dilatih untuk mengenali pola antara fitur input dan output target menggunakan data pelatihan. Setelah itu, model diuji dengan data pengujian untuk menilai kemampuannya memprediksi data baru. Hasil evaluasi menunjukkan 239 data "BERPRESTASI" dan 230 data "KURANG BERPRESTASI" diprediksi dengan benar. Namun, terdapat 16 data "KURANG BERPRESTASI" yang salah diprediksi sebagai "BERPRESTASI" dan 8 data "BERPRESTASI" yang salah diprediksi sebagai "KURANG BERPRESTASI." Hasil ini menunjukkan akurasi dan kesalahan prediksi model, seperti pada Gambar 7.



Gambar 7. Hasil akurasi pada Naïve Bayes

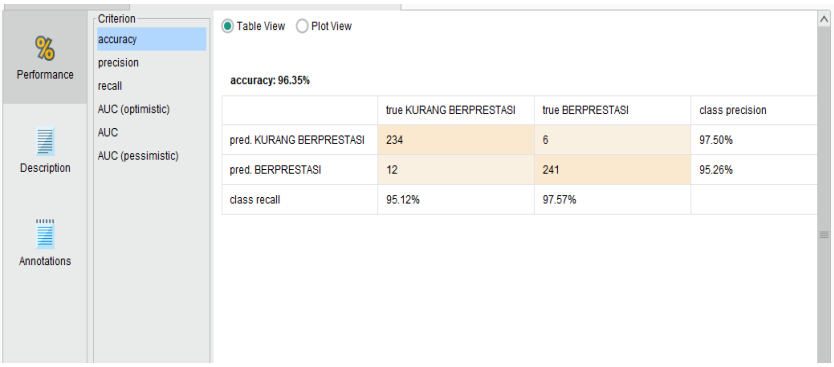
Perhitungan confusion matrix pada model Naïve Bayes pada performa model dalam memprediksi data. Berikut adalah hasil perhitungan confusion matrix dan berbagai metrik evaluasi.

- Akurasi = $\frac{239+230}{493}\frac{469}{493} \approx 0.9513$
  (akurasi model adalah sekitar 95.13%.)
- Presisi = $\frac{239}{239+16} = \frac{239}{255} \approx 0.9373$
  (presisi model untuk kelas positif Berprestasi adalah sekitar 93.73%.)
- Recall = $\frac{239}{239+8} = \frac{215}{247} \approx 0.96.76$
  (recall model untuk kelas positif Berprestasi adalah sekitar 96.76%.)

c. Analisis Model Random Forest

Selanjutnya adalah penggunaan pengolahan data Random Forest, dimulai dari memuat dataset berisi 493 record hingga evaluasi model. Pada proses workflow Random Forest sama seperti proses sebelumnya, hanya saja operator algoritma disesuaikan dengan metode Random Forest.

Penelitian ini menggunakan pembagian data dengan proporsi 80:20, di mana 393 dari 493 sampel digunakan untuk pelatihan, dan 100 sampel sisanya untuk pengujian. Model dilatih dengan data pelatihan untuk mengenali pola antara fitur input dan output target. Setelah itu, model diuji menggunakan data pengujian untuk menilai kemampuannya memprediksi data baru. Hasil evaluasi menunjukkan bahwa 241 data "BERPRESTASI" dan 234 data "KURANG BERPRESTASI" diprediksi dengan benar. Namun, terdapat 12 data "KURANG BERPRESTASI" yang salah diprediksi sebagai "BERPRESTASI" dan 6 data "BERPRESTASI" yang salah diprediksi sebagai "KURANG BERPRESTASI." Hasil ini menggambarkan akurasi dan kesalahan prediksi model, sebagaimana ditunjukkan pada Gambar 8.



Gambar 8. Hasil akurasi pada Random Forest

Perhitungan confusion matrix pada model Random Forest pada performa model dalam memprediksi data. Berikut adalah hasil perhitungan confusion matrix dan berbagai metrik evaluasi.

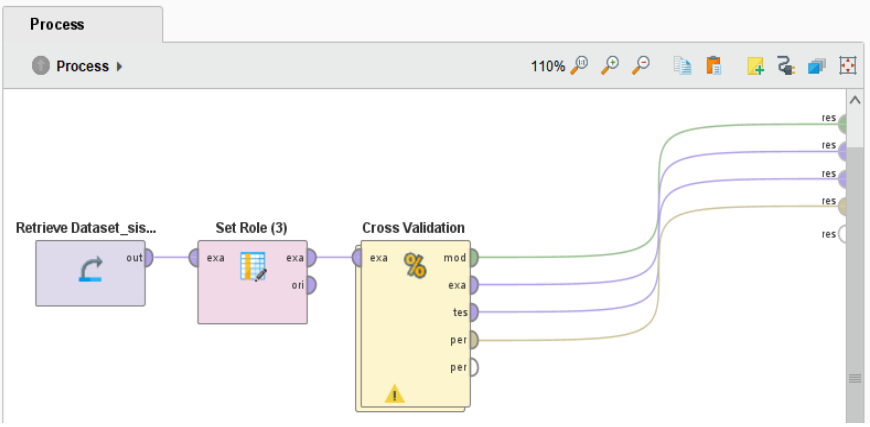
- Akurasi = $\frac{241+243}{493}\frac{475}{493} \approx 0.9635$
  (akurasi model adalah sekitar 96.35%.)
- Presisi = $\frac{241}{241+12} = \frac{241}{253} \approx 0.9526$
  (presisi model untuk kelas positif Berprestasi adalah sekitar 93.73%.)
- Recall = $\frac{241}{241+6} = \frac{241}{247} \approx 0.9757$

(recall model untuk kelas positif Berprestasi adalah sekitar 97.57%.)

d. Pengujian Cross-Validation

a) Evaluasi Model Menggunakan Cross Validation Pada RapidMinier

Cross Validation adalah teknik evaluasi model yang membagi dataset menjadi beberapa subset (folds) untuk memastikan keakuratan dan keandalan prediksi. Setiap subset secara bergantian digunakan sebagai data uji, sementara sisanya melatih model. Hasil rata-rata dari semua iterasi menjadi evaluasi akhir. Dalam analisis ini, algoritma Decision Tree, Naive Bayes, dan Random Forest diuji menggunakan 5-fold, 7-fold, dan 10-fold Cross Validation. Sebelumnya, pengujian awal dilakukan dengan pembagian data 80:20 untuk gambaran awal kinerja model. Hasil dari kedua metode dibandingkan menggunakan akurasi, presisi, recall, dan F1-Score untuk menentukan algoritma terbaik. Diagram proses Cross Validation ditampilkan pada Gambar 9.

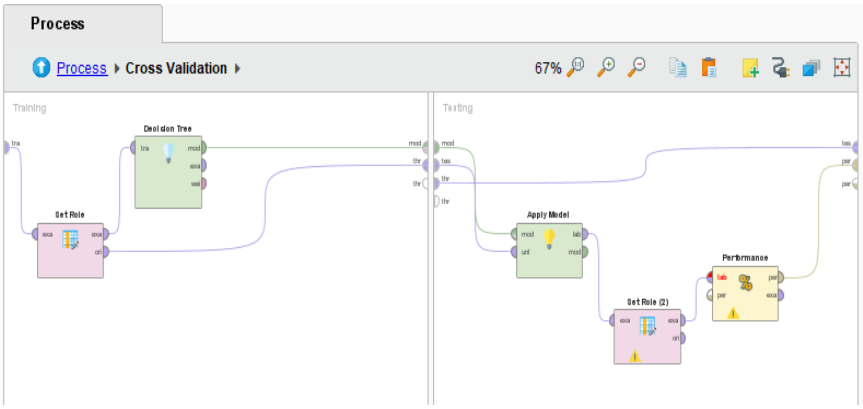


Gambar 9. Diagram proses cross validation pada RapidMiner

Membagi dataset menjadi beberapa subset (folds) untuk melatih dan menguji model secara bergantian. Pada pengujian ini digunakan konfigurasi:

- 5-fold Cross Validation: Dataset dibagi menjadi 5 bagian.
- 7-fold Cross Validation: Dataset dibagi menjadi 7 bagian.
- 10-fold Cross Validation: Dataset dibagi menjadi 10 bagian.

Setiap iterasi menghasilkan model pelatihan (mod) dan evaluasi performa (per) yang mencakup akurasi, presisi, recall, dan F1-Score. Proses ini diterapkan pada tiga algoritma Decision Tree, Naive Bayes, dan Random Forest. Hasil evaluasi dari setiap fold membantu menentukan algoritma dan konfigurasi terbaik untuk prediksi yang akurat dan andal. Berikut adalah proses Cross Validation di RapidMiner pada gambar 10.

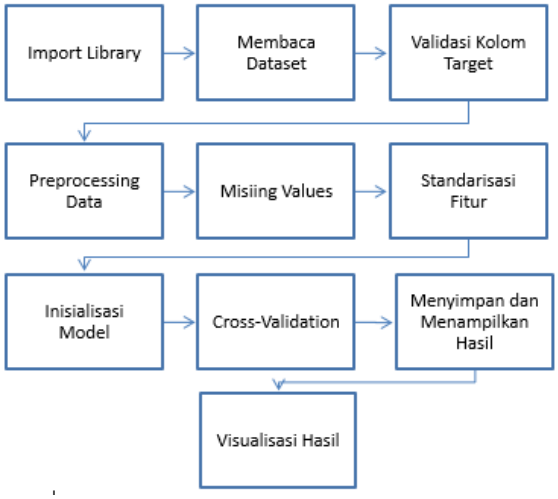


Gambar 10. Diagram Cross Validation model Decision Tree

Proses yang sama juga dilakukan dengan algoritma Naive Bayes dan Random Forest pada alur pelatihan dan pengujian model pada setiap iterasi Cross Validation.

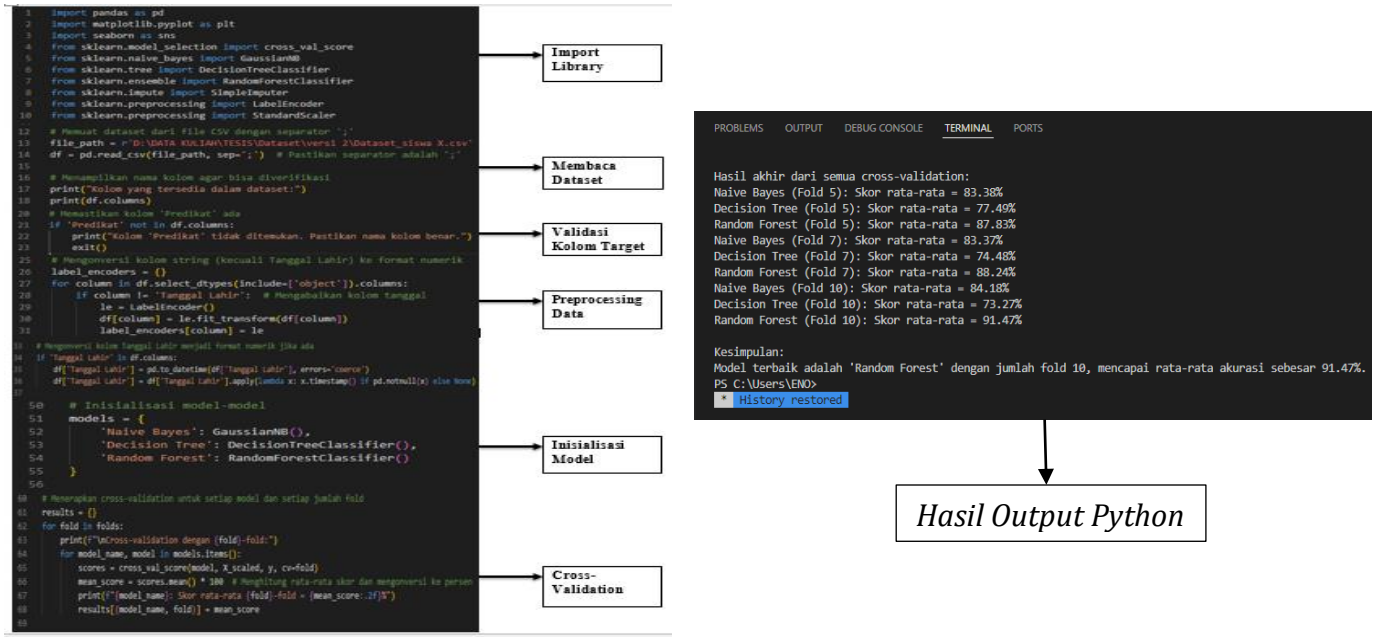
b) Evaluasi Model Menggunakan Cross Validation Pada Python

Pada gambar 14 diagram alur menggambarkan tahapan utama dalam proses pengolahan data dan pembuatan model machine learning, dimulai dari impor pustaka hingga penarikan visualisasi hasil. Setiap tahap merepresentasikan langkah krusial dalam memastikan model yang dihasilkan memiliki performa optimal dan validasi yang akurat. Proses ini mencakup penanganan data mentah, pembersihan data, standardisasi fitur, hingga evaluasi model menggunakan teknik cross-validation. Dengan visualisasi ini, setiap langkah terlihat jelas dan terstruktur, membantu memahami alur kerja secara keseluruhan



Gambar 14. Diagram Proses pengolahan data pada Python

Penelitian ini menganalisis data melalui tahapan persiapan dan pengolahan menggunakan pustaka Python. Proses dimulai dengan pembacaan dataset, validasi struktur data, penanganan nilai hilang, serta konversi data kategorikal dan berbasis waktu menjadi format numerik. Setelah transformasi selesai, dilakukan standardisasi untuk menyamakan skala fitur. Model Naive Bayes, Decision Tree, dan Random Forest kemudian diterapkan dan dievaluasi menggunakan validasi silang dengan berbagai jumlah lipatan (folds) untuk menilai performa. Hasil analisis disajikan dalam bentuk



statistik dan visualisasi untuk menentukan model terbaik berdasarkan akurasi rata-rata tertinggi. Tahapan utama pengolahan data dan pembuatan model ditampilkan pada gambar berikut.

Gambar 15. Evaluasi Model Machine Learning dengan Cross-Validation

Dari hasil pengolahan pada sebuah skrip Python file eksternal dataset bernama "Dataset_Siswa X" menggunakan Python versi 3.11. Outputnya menunjukkan daftar kolom dalam dataset, seperti No, Nama Siswa, dan berbagai mata pelajaran, termasuk PAI, PKN, B. Indonesia, hingga Produktif, serta kolom Predikat yang mungkin berisi hasil evaluasi siswa. Daftar kolom ini kemungkinan dihasilkan oleh pustaka pandas, yang digunakan untuk mengelola data dalam bentuk tabel. Skrip ini tampaknya dirancang untuk mengolah dan menganalisis data nilai siswa.

e. Uji Cross-Validation Menggunakan RapidMiner

Hasil pada pengujian cross-validation menggunakan nilai fold 5, fold 7, dan fold 10 dapat dilihat pada tabel 3 dibawah ini**.**

Tabel 3. Hasil cross-validation pada RapidMiner

Cross Validation	Keakuratan		
	Decision Tree	**Naïve Bayes**	**Random Forest**
5-fold	71.60%	84.20%	85.20%
7-fold	72.62%	84.99%	87.01%
10-fold	72.82%	84.37%	86.00%

- Pada cross-validation 5-fold
  akurasi tertinggi dicapai oleh Random Forest dengan nilai 85.20%, diikuti oleh Naïve Bayes (84.20%) dan Decision Tree (71.60%).
- Pada cross-validation 7-fold
  Random Forest tetap unggul dengan akurasi 87.01%, lebih tinggi dibandingkan Naïve Bayes (84.99%) dan Decision Tree (72.62%).
- Pada cross-validation 10-fold
  akurasi tertinggi juga dicapai oleh Random Forest dengan nilai 86.00%, diikuti oleh Naïve Bayes (84.37%) dan Decision Tree (72.82%).

Dari pengujian dengan *RapidMiner*, model terbaik adalah *Random Forest*, dengan akurasi tertinggi sebesar 87.01% pada 7-fold cross-validation.

f. Uji Cross-Validation Menggunakan Python

Hasil pada pengujian cross-validation menggunakan nilai fold 5, fold 7, dan fold 10 dapat dilihat pada tabel 4 dibawah ini**.**

Tabel 4. Hasil cross-validation pada python

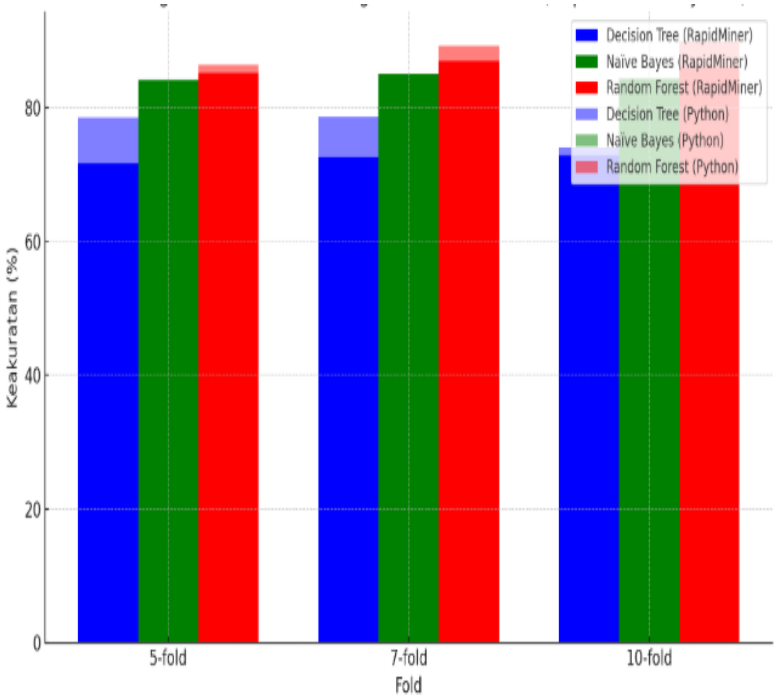
Cross Validation	Keakuratan		
	Decision Tree	**Naïve Bayes**	**Random Forest**
5-fold	78.51%	83.38%	86.40%
7-fold	78.63%	83.37%	89.26%
10-fold	74.09%	84.18%	89.86%

- Pada 5-fold cross-validation,
  Random Forest menunjukkan performa terbaik dengan akurasi 86.40%, diikuti oleh Naïve Bayes (83.38%) dan Decision Tree (78.51%).
- Pada 7-fold cross-validation,

akurasi tertinggi juga diraih oleh Random Forest, yaitu 89.26%, lebih tinggi dibandingkan Naïve Bayes (83.37%) dan Decision Tree (78.63%).

- Pada 10-fold cross-validation,
  Random Forest kembali unggul dengan akurasi 89.86%, sementara Naïve Bayes mencatat akurasi 84.18%, dan Decision Tree sebesar 74.09%.

Pengujian dengan *Python* menunjukkan bahwa model terbaik adalah Random Forest, dengan akurasi tertinggi sebesar 89.86% pada 10-fold cross-validation. Berdasarkan hasil pada kedua tabel tersebut maka hasil menunjukkan akurasi masing-masing model pada 3 level fold (5-fold, 7-fold, 10-fold) untuk kedua platform dengan Random Forest menunjukkan hasil terbaik di kedua pengujian yang ditampilkan pada (gambar 16) sebuah grafik dibawah ini.



Gambar 9. Diagram Proses pengolahan data pada Python

g. Hasil Akhir Predikasi Metode Algoritma

Tabel 5. Hasil akhir prediksi metode algoritma

Platform	Hasil Akhir	Ketrangan
RapidMiner	87.01%	*Random Forest* pada 7-fold cross-validation.
Python	89.86%	*Random Forest* pada 10-fold cross-validation.

Hasil uji cross-validation dari kedua platform menunjukkan konsistensi bahwa Random Forest adalah model dengan performa terbaik dalam prediksi prestasi siswa. Namun, akurasi tertinggi dicapai pada pengujian dengan Python yaitu 89.86% pada 10-fold cross-validation, yang mengindikasikan bahwa Python memberikan hasil yang lebih optimal dalam hal prediksi dibandingkan dengan platform lainnya.

2. Pembahasan

Penelitian ini menganalisis kinerja tiga algoritma klasifikasi, yaitu Decision Tree, Naïve Bayes, dan Random Forest, dalam memprediksi kinerja siswa berdasarkan dataset yang telah disiapkan. Proses pengolahan data menggunakan RapidMiner mencakup tahapan mulai dari pemuatan dataset, seleksi dan preprocessing data, pemisahan menjadi training dan testing set, hingga validasi model. Hasil evaluasi menunjukkan bahwa model Decision Tree memiliki akurasi sebesar 87.02%, dengan presisi dan recall masing-masing sebesar 87.04%. Model ini mampu mengklasifikasikan data dengan baik, tetapi masih terdapat kesalahan dalam prediksi dengan sejumlah data berprestasi dan kurang berprestasi yang diklasifikasikan secara keliru. Hasil ini sejalan dengan penelitian yang dilakukan oleh Quinlan (1996) yang menunjukkan bahwa Decision Tree efektif dalam menangkap pola dari data yang kompleks, tetapi rentan terhadap overfitting jika tidak dioptimalkan dengan baik.

Selanjutnya, analisis menggunakan Naïve Bayes menunjukkan peningkatan performa dengan akurasi sebesar 95.13%. Model ini mampu memprediksi kategori berprestasi dengan presisi 93.73% dan recall 96.76%, menunjukkan keunggulan dalam menangani data dengan probabilitas bersyarat. Keakuratan tinggi ini mendukung temuan dari Domingos dan Pazzani (1997) yang menegaskan bahwa meskipun asumsi independensi kondisional dalam Naïve Bayes seringkali tidak realistis, algoritma ini tetap memberikan hasil yang kompetitif dalam berbagai tugas klasifikasi. Namun, kesalahan prediksi masih terjadi, meskipun dalam jumlah yang lebih kecil dibandingkan dengan Decision Tree.

Sementara itu, analisis menggunakan Random Forest menunjukkan hasil terbaik dengan akurasi sebesar 96.35%, presisi 95.26%, dan recall 97.57%. Model ini memiliki keunggulan dalam mengatasi overfitting dengan menggabungkan prediksi dari berbagai pohon keputusan, menghasilkan prediksi yang lebih stabil dan akurat. Temuan ini sejalan dengan Breiman (2001) yang menunjukkan bahwa Random Forest secara konsisten memiliki performa superior dibandingkan model klasifikasi tunggal dalam berbagai skenario data. Selain itu, penggunaan ensemble learning dalam Random Forest memberikan keunggulan dalam menangani variasi dalam data yang beragam, sehingga menghasilkan model yang lebih andal.

Evaluasi lebih lanjut menggunakan teknik Cross Validation dengan 5-fold, 7-fold, dan 10-fold dilakukan untuk memastikan keandalan model. Hasil menunjukkan bahwa Random Forest tetap unggul dalam berbagai skenario validasi, dengan performa yang lebih konsisten dibandingkan dua model lainnya. Teknik validasi silang ini mendukung penelitian sebelumnya oleh Kohavi (1995), yang menyebutkan bahwa cross-validation membantu mengurangi bias dan varians dalam evaluasi model, sehingga memberikan gambaran yang lebih akurat tentang performa model pada data baru.

Selain menggunakan RapidMiner, penelitian ini juga mengimplementasikan proses pengolahan data menggunakan pustaka Python untuk memastikan validasi yang lebih menyeluruh. Analisis dilakukan dengan mengimpor dataset, melakukan pembersihan data, standardisasi fitur, serta mengimplementasikan algoritma klasifikasi dengan evaluasi berbasis cross-validation. Hasil dari kedua metode menunjukkan bahwa Random Forest tetap menjadi model terbaik dengan akurasi tertinggi, menguatkan temuan sebelumnya tentang efektivitasnya dalam klasifikasi data.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa Random Forest memiliki performa terbaik dalam memprediksi kinerja siswa dibandingkan dengan Decision Tree dan Naïve Bayes. Keunggulan Random Forest dalam menangani data yang kompleks dan mengurangi

overfitting menjadikannya model yang paling andal dalam penelitian ini. Namun, penelitian lebih lanjut dapat dilakukan dengan mengeksplorasi teknik optimasi parameter atau menggabungkan metode ensemble lainnya untuk meningkatkan performa model lebih lanjut. Temuan ini memperkaya literatur terkait penerapan algoritma machine learning dalam bidang pendidikan, terutama dalam upaya meningkatkan akurasi prediksi performa siswa berdasarkan data historis.

Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan penulis dalam mengevaluasi prestasi siswa di SMKN 8 Kabupaten Tangerang dengan menggunakan algoritma Decision Tree, Naïve Bayes, dan Random Forest berdasarkan dataset 493 siswa dari enam jurusan, melalui platform RapidMiner dan Python. Berdasarkan hasil analisis, algoritma Random Forest memberikan akurasi tertinggi, yaitu 89,86% pada Python dengan validasi silang 10-fold dan 87,01% pada RapidMiner dengan validasi 7-fold, menjadikannya metode terbaik untuk evaluasi prestasi siswa. Algoritma Naïve Bayes berada di posisi kedua dengan akurasi 84,99% di RapidMiner dan 84,18% di Python, sedangkan Decision Tree menunjukkan akurasi terendah, yaitu 76,47% pada Python dan 72,82% pada RapidMiner. Secara keseluruhan, Python menunjukkan hasil yang lebih baik dibandingkan RapidMiner, berkat validasi yang lebih mendalam dan pengaturan parameter yang lebih optimal. Dengan demikian, Random Forest adalah algoritma terbaik untuk mendukung pengambilan keputusan berbasis data dalam meningkatkan kualitas pendidikan di SMKN 8 Kabupaten Tangerang.

Beberapa saran dari penulis yang dapat diberikan. Pertama, dataset bisa diperluas dengan menambah jumlah data dan variabel baru, serta menggunakan data dari periode yang lebih panjang. Penelitian selanjutnya juga bisa menguji algoritma lain seperti SVM atau XGBoost untuk perbandingan. Kedua, hasil analisis dapat digunakan untuk merancang program pembinaan siswa dan mengintegrasikan data mining dalam manajemen sekolah. Ketiga, pelatihan untuk guru dan staf diperlukan agar hasil analisis bisa dimanfaatkan dengan maksimal, dan kerja sama dengan industri serta pendidikan tinggi bisa mendukung pengembangan kurikulum yang lebih relevan. Terakhir, hasil penelitian ini dapat menjadi dasar untuk kebijakan peningkatan mutu pendidikan, sehingga lulusan lebih siap menghadapi dunia kerja atau melanjutkan pendidikan ke jenjang yang lebih tinggi

Referensi

Fauzan, A S, A I P Sari, and I Ali. 2024. Jati (Jurnal Mahasiswa Teknik Informatika *Analisis Perbandingan Algoritma Decisioin Tree Dan Naïve Untuk Mengevaluasi Prestasi Belajar Siswa*. ejournal.itn.ac.id.

Ramadani, R, and B H Hayadi. 2022. Journal Computer Science and Information Technology *Perbandingan Metode Naive Bayes Dan Random Forest Untuk Menentukan Prestasi Belajar Siswa Pada Jurusan Rpl (Studi Kasus Smk Swasta Siti Banun Sigambal)*. jurnal.ulb.ac.id.

Hayati, I, J Marzal, and E Saputra. 2021. *Klasifikasi Mahasiswa Berpotensi Drop Out Menggunakan Algoritma Decision Tree C4. 5 Dan Naive Bayes Di Universitas Jambi*. repository.unja.ac.id.

Nurrizta, S N. 2022. *Penerapan Metode Algoritma C4. 5 Untuk Prediksi Hasil Prestasi Belajar Siswa Di Sekolah Menengah Pertama/Syahla Nabila Nurrizta 14180044 / Program Studi Sistem Informasi / Pembimbing I : Irmayansyah / Pembimbing II : Mochamad Sanwasih.f*inkom.repository.unbin.ac.id.

Putra, M Y, and D I Putri. 2022. Jurnal Tekno Kompak *Pemanfaatan Algoritma Naïve Bayes Dan K-Nearest Neighbor Untuk Klasifikasi Jurusan Siswa Kelas XI*. ejurnal.teknokrat.ac.id.

Riyanto, A, and E S Ompusunggu. 2024. Jurnal Teknologi Dan Ilmu Komputer Prima (Jutikomp), *Implementasi Data Mining Untuk Mengklasifikasi Hasil Belajar Siswa/i Dengan Metode Naïve Bayes*. jurnal.unprimdn.ac.id.

Triwidianti, J, F Y Alfian, and Margi Prasojo 2021. Prosiding Seminar Nasional Darmajaya. *Perbandingan Metode Data Mining Untuk Prediksi Prestasi Siswa Tingkat Pendidikan Menengah Kejuruan Pada Sekolah Menengah Kejuruan Negeri (SMKN 1) Gadingrejo Pringsewu Lampung* jurnal.darmajaya.ac.id.

Latifah, H, and S Mujiyono. 2023. Jurnal Mahasiswa Teknik Informatika *Perbandingan Algoritma Naã Ve Bayes, K-Nn, Id3, Dan Svm Dalam Menentukan Prediksi Kelulusan Siswa Di Smk Muhamadiah Majenang*. jurnal.unw.ac.id.

Hidayat, I. 2023. *Penerapan Algoritma C4. 5 Dan Naïve Bayes Untuk Mengukur Tingkat Kepuasan Mahasiswa Dalam Penggunaan Edlink*. repository.unilak.ac.id.

[Andriyani, Wini, Rudi Kurniawan, and Yudhistira Arie Wijaya. 2024. 8 Jati (Jurnal Mahasiswa Teknik Informatika) *Analisis Data Penerimaan Peserta Didik Baru Menggunakan Cross Validation Dan Algoritma Decision Tree Di Sma Negeri 1 Bandung*. ejournal.itn.ac.id.

Almufqi, F M, and A Voutama. 2023. Jurnal Teknika *Perbandingan Metode Data Mining Untuk Memprediksi Prestasi Akademik Siswa*. jurnalteknik.unisla.ac.id.

[Darmawan, A, I Yudhisari, A Anwari, and Makruf M 2023. Jurnal Minfo Polgan *Pola Prediksi Kelulusan Siswa Madrasah Aliyah Swasta Dengan Support Vector Machine Dan Random Forest*. jurnal.polgan.ac.id.

[Hidayat, I. 2023. *Penerapan Algoritma C4. 5 Dan Naïve Bayes Untuk Mengukur Tingkat Kepuasan Mahasiswa Dalam Penggunaan Edlink*. repository.unilak.ac.id.

[Indrayana, Y K, R K Ramadhan, and IB Kurniawan, 2023. Prosiding Seminar Nasional Amikom Surakarta *Implementasi Decision Tree Untuk Mengklasifikasikan Metode Pembayaran Di Supermarket*. ojs.amikomsolo.ac.id.

Lestari, S, and B Dina. 2023. Jurnal Indonesia: Manajemen Informatika dan Komunikasi *Klasifikasi Ketepatan Kelulusan Siswa Pada Smk Yadika 9 Bintara Jaya Kota Bekasi Menggunakan Algoritma C4. 5*. journal.stmiki.ac.id.