Perbandingan Algoritma *Machine Learning* Dalam Memprediksi Kelulusan Siswa

Nidaul Maftucha[1,*], Saffanah Salma[2], Novita Rahmayuna[3], Nur Wakhidah[4]

[1, 2, 4]Fakultas Teknologi Informasi dan Komunikasi, Sistem Informasi, Universitas Semarang, Kota Semarang, Indonesia
[3]*School of Information Systems, Information Systems*, Universitas Bina Nusantara, Kota Semarang, Indonesia
Email: [1,*]nidaul.maftucha@gmail.com, [2]saffma20@gmail.com, [3]novita.rahmayuna@binus.ac.id, [4]ida@usm.ac.id
[*)] Email Penulis Utama

Abstrak—Prediksi kelulusan siswa sangat penting karena dapat membantu sekolah, guru, dan orang tua merencanakan bagaimana membantu siswa yang berisiko tidak lulus. Prediksi ini juga dapat memberi lembaga pendidikan kesempatan untuk meningkatkan kualitas pembelajaran dan mengembangkan tindakan yang lebih efisien. Pada penelitian ini membahas tentang perbandingan algoritma *machine learning* dalam memprediksi kelulusan siswa. Masalah utama yang diidentifikasi adalah kurangnya sistem prediksi yang efektif, yang dapat memprediksi kelulusan siswa. Tujuan dari penelitian ini adalah untuk menemukan metode terbaik dengan membandingkan kinerja lima algoritma *machine learning* yaitu K-NN, *Naive Bayes*, *Decision Tree*, *Random Forest*, dan SVM dalam memprediksi kelulusan siswa berdasarkan *confusion matrix*. Kemudian, dataset yang digunakan untuk penelitian ini memiliki tiga kategori, yaitu: dataset numerikal, dataset kategorikal, dan dataset keseluruhan (gabungan dari numerikal dan kategorikal). Dataset numerikal terdiri dari berbagai fitur, antara lain: *Application mode, Course, Previous qualification, Previous qualification (grade), Nacionality, Mother's qualification, Father's qualification, Mother's occupation, Father's occupation, Admission grade, Age at enrollment, Curricular units 1st sem (enrolled), Curricular units 1st sem (evaluations), Curricular units 1st sem (approved), Curricular units 1st sem (grade), Curricular units 2nd sem (enrolled), Curricular units 2nd sem (evaluations), Curricular units 2nd sem (approved), Curricular units 2nd sem (grade), Unemployment rate, Inflation rate*, GDP, dan Target. Sementara itu fitur yang mencakup dataset kategorikal, yaitu: *Marital status, Application order, Daytime/evening attendance, Displaced, Educational special needs, Debtor, Tuition fees up to date, Gender, Scholarship holder, International, Curricular units 1st sem (credited), Curricular units 1st sem (without evaluations), Curricular units 2nd sem (credited),* dan *Curricular units 2nd sem (without evaluations).* Adapun dataset keseluruhan merupakan fitur gabungan dari dataset numerikal dan kategorikal. Hasil pengujian dari dataset numerikal algoritma *Random Forest* mendapatkan nilai akurasi terbaik sebesar 74.12%. Pada algoritma dengan fitur kategorikal K-NN dan SVM memiliki nilai akurasi tertinggi dengan mendapatkan nilai sebesar 93.11%. Namun, algoritma *Random Forest* memiliki performa yang paling konsisten dan unggul ketika seluruh fitur digabungkan. Dengan mendapatkan nilai akurasi tertinggi sebesar 76.50% dan F1-Scorenya 75.00%.

Kata Kunci: *Machine Learning*, Prediksi Kelulusan, *Random Forest*, K-NN, *Decision Tree*

Abstract— Predicting student graduation is very important as it can help schools, teachers and parents plan how to help students who are at risk of not graduating. This prediction can also give educational institutions the opportunity to improve the quality of learning and develop more efficient actions. This research discusses the comparison of machine learning algorithms in predicting student graduation. The main problem identified is the lack of an effective prediction system, which can predict student graduation. The purpose of this research is to find the best method by comparing the performance of five machine learning algorithms namely K-NN, Naive Bayes, Decision Tree, Random Forest, and SVM in predicting student graduation based on confusion matrix. Then, the dataset used for this research has three categories, namely: numerical dataset, categorical dataset, and overall dataset (combination of numerical and categorical). The numerical dataset consists of various features, including: Application mode, Course, Previous qualification, Previous qualification (grade), Nationality, Mother's qualification, Father's qualification, Mother's occupation, Father's occupation, Admission grade, Age at enrollment, Curricular units 1st sem (enrolled), Curricular units 1st sem (evaluations), Curricular units 1st sem (approved), Curricular units 1st sem (grade), Curricular units 2nd sem (enrolled), Curricular units 2nd sem (evaluations), Curricular units 2nd sem (approved), Curricular units 2nd sem (grade), Unemployment rate, Inflation rate, GDP, and Target. Meanwhile, the features that cover categorical datasets are: Marital status, Application order, Daytime/evening attendance, Displaced, Educational special needs, Debtor, Tuition fees up to date, Gender, Scholarship holder, International, Curricular units 1st sem (credited), Curricular units 1st sem (without evaluations), Curricular units 2nd sem (credited), and Curricular units 2nd sem (without evaluations). The overall dataset is a combined feature of numerical and categorical datasets. The test results from the numerical dataset of the Random Forest algorithm get the best accuracy value of 74.12%. In algorithms with categorical features K-NN and SVM have the highest accuracy value by getting a value of 93.11%. However, the Random Forest algorithm has the most consistent and superior performance when all features are combined. By getting the highest accuracy value of 76.50% and F1-Score of 75.00%.

Keywords: *Machine Learning*, *Graduation Prediction*, *Random Forest*, K-NN, *Decision Tree*

1. PENDAHULUAN

Salah satu ukuran utama keberhasilan sistem pendidikan adalah tingkat kelulusan siswa. Kelulusan siswa merupakan salah satu bidang yang termasuk ke dalam Standar Penjaminan Mutu Internal (SPMI) suatu perguruan tinggi di Indonesia[1]. Prediksi kelulusan siswa sangat penting karena dapat membantu perguruan tinggi, dosen, dan orang tua untuk merencanakan bagaimana membantu siswa yang berisiko tidak lulus. Prediksi ini juga dapat memberi lembaga pendidikan kesempatan untuk meningkatkan kualitas pembelajaran dan mengembangkan tindakan yang lebih efisien. Lembaga pendidikan dapat meningkatkan tingkat kelulusan dengan memahami faktor-faktor yang memengaruhi kelulusan siswa. Dalam situasi seperti ini, prediksi kelulusan merupakan komponen penting dari upaya untuk membuat sistem pendidikan yang lebih fleksibel dan inklusif. Namun, untuk mencegah keterlambatan kelulusan siswa, beberapa perguruan tinggi tidak memiliki sistem untuk memprediksinya, sehingga perguruan tinggi belum bisa melakukan pencegahan akan hal tersebut[2].

Sebelum penggunaan algoritma *machine learning*, metode seperti *regresi* dan *statistik* telah digunakan dalam penelitian tentang prediksi kelulusan siswa. Misalnya, penelitian yang dilakukan oleh Nahrowi Hamdani et al. 2020 yang membahas tentang penggunaan algoritma data mining untuk memprediksi kelulusan siswa asrama Universitas Muhammadiyah Yogyakarta. Tujuan dari penelitian yang dilakukan oleh Nahrowi Hamdani et al adalah untuk meneliti hubungan antara prestasi akademik siswa dan pembinaan di asrama. Penelitian ini menggunakan algoritma *Regresi Logistic* dan *Neural Network* untuk menganalisis dan memprediksi data dengan menggunakan data dari angkatan tahun 2014-2015. Hasil penelitian menunjukkan bahwa *Neural Network* memberikan akurasi sebesar 69% dan *Regresi Logistik* memberikan akurasi sebesar 65% dalam memprediksi kelulusan mahasiswa tepat waktu. Hasil menunjukkan bahwa dalam hal ini, algoritma *Neural Network* mungkin lebih efektif daripada *Regresi Logistic[3]*. Penelitian ini menunjukkan bahwa metode statistik dapat memberikan wawasan awal tentang komponen yang mempengaruhi kelulusan siswa, meskipun seringkali memiliki keterbatasan dalam hal akurasi dan kompleksitas analisis.

Teknologi *machine learning* saat ini berkembang dengan pesat dan menjadi salah satu teknologi utama dalam kecerdasan buatan. *Machine learning* adalah salah satu teknologi pembelajaran yang dapat mempermudah pekerjaan[4]. Algoritma *machine learning* dapat membantu dalam pengambilan keputusan pendidikan, seperti memprediksi kelulusan siswa. *Machine learning* dapat menemukan pola tersembunyi yang sulit ditemukan dengan metode tradisional karena kemampuannya untuk menganalisis data dalam jumlah besar dan kompleks. Pendidik dapat membuat model prediksi yang akurat dan berbasis data dengan algoritma seperti K-NN, *Naive bayes*, *Decision Tree*, *Random Forest*, dan SVM. Penggunaan *machine learning* tidak hanya meningkatkan efisiensi tetapi juga memberi peluang untuk pendidikan, sehingga setiap siswa dapat menerima tindakan yang sesuai dengan kebutuhan mereka. Oleh karena itu, penggunaan algoritma *machine learning* di bidang pendidikan tidak hanya untuk meningkatkan kualitas prediksi, tetapi juga dapat membantu mengubah sistem pendidikan menjadi sistem berbasis data yang lebih canggih.

Hasil penelitian sebelumnya oleh Satrio Junaidi et al. menggunakan metode data mining klasifikasi untuk memprediksi kelulusan mahasiswa tepat waktu dengan menggunakan empat algoritma, yaitu: *Naive Bayes*, *Random Forest*, *Support Vector Machine* (SVM), dan *Artificial Neural Network* (ANN). Jenis kelamin, usia, penghasilan orang tua, durasi bimbingan, status mahasiswa bekerja atau tidak, nilai semester 1 hingga semester 8 dan IPK adalah atribut yang digunakan. Untuk memproses dataset, penelitian ini menggunakan bahasa pemrograman python 3 pada jupyter notebook di Anaconda. Data dibagi menjadi 30% untuk data testing dan 70% untuk data training. Hasil penelitian ini ditemukan bahwa algoritma *Support Vector Machine* (SVM) mendapatkan akurasi terbaik dengan nilai 0.94[2].

Selanjutnya, penelitian yang dilakukan oleh Junta Zeniarja et al. pada tahun 2022 bertujuan untuk menghasilkan model klasifikasi terbaik dengan membandingkan tingkat akurasi tertinggi dari berbagai algoritma klasifikasi, termasuk *Naive Bayes*, *Random Forest*, *Decision Tree*, *K-Nearest Neighbor* (K-NN), dan *Support Vector Machine* (SVM) untuk memprediksi kelulusan siswa. Sebelum proses klasifikasi, proses seleksi fitur juga digunakan untuk mengoptimalkan model. Untuk penelitian ini, 2293 data digunakan yang terdiri dari mahasiswa yang lulus dari program Sarjana Teknik Informatika dengan kode A11 dari tahun 2012 hingga 2017. Hasil penelitian ini menunjukkan bahwa model klasifikasi dengan algoritma *Random Forest* dengan nilai akurasi tertinggi mencapai 77,35% lebih baik daripada algoritma lain, model ini menggunakan seleksi fitur terbaik yaitu dua belas fitur atribut regular dan satu atribut sebagai label dengan data pengujian sebesar 25%[5].

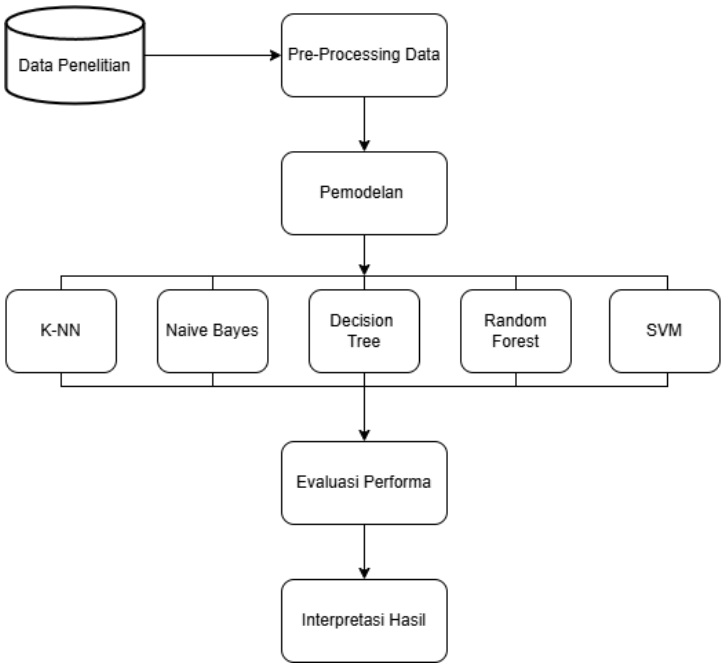
Selain itu, Fitra A. Bachtiar et al. juga meneliti perbandingan kinerja algoritma dalam klasifikasi siswa yang mengambil mata kuliah. Prediksi dibuat oleh data mahasiswa. Kita dapat memprediksi apakah siswa akan mengambil mata kuliah tertentu dengan melihat nilai, IP, IPK, SKS, SKSK, dan semester. Kategori hasil prediksi terdiri dari "Ya" (mengambil mata kuliah) dan "Tidak" (tidak mengambil mata kuliah). Teknik *Synthetic Minority Oversampling Technique* (SMOTE) digunakan untuk menangani data yang tidak seimbang. Dalam penelitian ini, klasifikasi dilakukan dengan membandingkan algoritme *K-Nearest Neighbor* (K-NN) dan *Support Vector Machine* (SVM). Hasil pengujian dengan data dari tiga mata kuliah sebagai sampel menunjukkan bahwa K-NN memiliki kinerja yang lebih baik dibandingkan SVM. Selain itu, penerapan teknik SMOTE terbukti dapat meningkatkan beberapa metrik evaluasi seperti AUC, akurasi (CA), *F1-score*, *precision*, dan *recall*[6].

Pada tahun 2024, Indra dan Dwi Agustinawati juga meneliti perbandingan metode *Naive Bayes* dan *K-Nearest Neighbor* untuk mengetahui hasil prediksi kelulusan mahasiswa. Data yang digunakan berjumlah 500 data dan diperoleh dari Direktorat Teknologi Informasi (DTI) Universitas Budi Luhur dalam program studi Teknik Informatika dan Sistem Informasi dari tahun 2015 hingga 2018. Ada 400 data training dan 100 data testing, yang diperoleh dari perbandingan 80% pelatihan dan 20% pengujian. Dalam penelitian ini, indeks prestasi semester (IPS) dari semester 2 hingga semester 7 digunakan untuk menghitung *Information Gain* dan status kelulusan (tepat waktu atau terlambat). Setelah perhitungan dan pengujian, algoritma *Naive Bayes* memiliki nilai akurasi tertinggi sebesar 74%, sedangkan algoritma *K-Nearest Neighbor* memiliki nilai akurasi sebesar 71%. Ini menunjukkan bahwa pemilihan fitur dan metode yang tepat sangat berpengaruh terhadap hasil prediksi dalam memprediksi kelulusan siswa[7].

Tujuan penelitian ini adalah untuk menemukan metode terbaik dengan membandingkan kinerja lima algoritma *machine learning* dalam memprediksi kelulusan siswa berdasarkan *confusion matrix* yang meliputi hasil akurasi, presisi, recall, dan F1-Score. Metode yang digunakan dalam penelitian ini adalah, K-NN, *Naive Bayes*, *Decision Tree*, *Random Forest*, dan SVM. K-NN dipilih karena kesederhanaan dan kemampuannya untuk menangani masalah tanpa banyak asumsi, sementara *Naive Bayes* dipilih karena efisiensinya, terutama dalam klasifikasi teks. *Decision Tree* digunakan karena kemudahannya dalam interpretasi dan kemampuannya untuk memberikan wawasan tentang fitur-fitur penting. *Random Forest*, sebagai pengembangan dari *Decision Tree*, dipilih untuk mengatasi *overfitting* dan meningkatkan akurasi melalui penggabungan banyak pohon keputusan. Terakhir, SVM dipilih karena keefektifannya dalam menangani masalah klasifikasi kompleks dengan mencari *hyperplane* optimal yang memisahkan kelas data. Hasil dari analisis dan perbandingan algoritma *machine learning* diharapkan dapat memberikan pemahaman yang lebih baik tentang efektivitas masing-masing algoritma, sehingga membantu pengambilan kebijakan dalam memilih metode terbaik. Selain itu, penelitian ini dapat membantu dalam pengembangan teknologi pendidikan yang lebih canggih dan berbasis data yang dapat memprediksi tingkat kelulusan siswa.

2. METODE PENELITIAN

Tahapan penelitian ini dimulai dengan pengumpulan data penelitian yang kemudian diproses melalui tahap *pre-processing* data untuk membersihkan dan mempersiapkan data agar sesuai untuk pemodelan. Selanjutnya, dilakukan pemodelan dengan menerapkan lima algoritma *machine learning*, yaitu: K-NN, *Naive Bayes*, *Decision Tree*, *Random Forest*, dan SVM dengan alat bantu *Google Collaboratory* atau yang biasa disingkat dengan Google Colab. Setelah pemodelan, dilakukan evaluasi performa model menggunakan *confusion matrix* seperti akurasi, presisi, *recall*, dan F1-score. Terakhir adalah interpretasi hasil untuk menentukan algoritma terbaik yang mempengaruhi kelulusan siswa. Tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Metode Penelitian

2.1 Data Penelitian

Dataset yang digunakan dalam penelitian ini adalah data public "*Predict Student Dropout and Academic Success*" yang dapat diakses di Kaggle. Dataset ini terdiri dari 4.424 baris dengan 37 atribut yang mencakup berbagai faktor yang dapat mempengaruhi kelulusan siswa. Atribut tujuan atau label dari dataset ini memiliki 3 kelas yaitu: *Graduate*, *Dropout*, dan *Enrolled*. Dari 37 atribut dalam analisis ini, langkah pertama adalah dengan membagi menjadi data kategorikal dan numerikal. 23 dari dataset numerikal adalah *Application mode, Course, Previous qualification, Previous qualification (grade), Nacionality, Mother's qualification, Father's qualification, Mother's occupation, Father's occupation, Admission grade, Age at enrollment, Curricular units 1st sem (enrolled), Curricular units 1st sem (evaluations), Curricular units 1st sem (approved), Curricular units 1st sem (grade), Curricular units 2nd sem (enrolled), Curricular units 2nd sem (evaluations), Curricular units 2nd sem (approved), Curricular units 2nd sem (grade), Unemployment rate, Inflation rate*, GDP, dan Target. Sedangkan 14 dari dataset kategorikal yaitu, *Marital status, Application order, Daytime/evening attendance, Displaced, Educational special needs, Debtor, Tuition fees up to date, Gender, Scholarship holder, International, Curricular units 1st sem (credited), Curricular units 1st sem (without evaluations), Curricular units 2nd sem (credited),* dan *Curricular units 2nd sem (without evaluations).* Pemisahan fitur ini sangat penting karena akan digunakan untuk menguji algoritma *machine learning* yang paling sesuai untuk ketiga jenis data. Oleh karena itu, diharapkan bahwa analisis akan memberikan hasil yang lebih akurat dalam memprediksi kelulusan siswa berdasarkan atribut.

2.2 Pre-processing

Pre-processing adalah serangkaian langkah awal yang dilakukan untuk mempersiapkan data mentah menjadi lebih bersih, terstruktur, dan siap digunakan dalam analisis atau pembuatan model *machine learning*[8]. Proses ini diperlukan untuk meningkatkan kualitas data dan memastikan bahwa analisis yang dilakukan menghasilkan informasi yang relevan dan akurat. Tahapan dari *pre-processing* data: yang pertama nilai non-numerik ditransformasi menjadi numerik dan diisi dengan nilai 0 menggunakan fillna(). Kemudian, variabel target non-numerik juga diubah menjadi bentuk numerik dengan menggunakan LabelEncoder. Untuk memastikan skala yang seragam, fitur numerik distandarisasi dengan "StandardScaler". Terakhir data dibagi menjadi data latih dan uji dengan menggunakan train_test_split dengan perbandingan 80:20. Pembagian ini didasarkan pada Prinsip Pareto, yang mengatakan bahwa 20% upaya menghasilkan sekitar 80% hasil. Dalam situasi ini, 20% data yang dialokasikan untuk pengujian dimaksudkan untuk menguji kinerja model pada data yang belum pernah dilihat sebelumnya, memberikan evaluasi yang lebih akurat tentang kemampuan generalisasinya. Sementara itu, 80% data yang dialokasikan untuk pelatihan bertujuan untuk memberikan model pembelajaran mesin jumlah data yang cukup untuk mengenali pola dan hubungan yang signifikan. Rasio 80:20 sering dianggap sebagai titik awal yang baik karena memberikan keseimbangan antara data pelatihan yang memadai dan data pengujian yang representatif. Namun, rasio lain seperti 70:30 atau 60:40 juga dapat digunakan. Dalam penelitian ini bahkan menunjukkan bahwa rasio 80:20 juga dapat menghasilkan tingkat kesalahan yang lebih rendah dalam model prediksi, sehingga meningkatkan akurasi secara keseluruhan.

2.3 Pemodelan

Pada tahap ini dilakukan pemodelan menggunakan algoritma *machine learning*, adapun algoritma yang digunakan diantaranya *K-Nearest Neighbors*(K-NN), *Naïve Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine*(SVM).

2.3.1 K-Nearest Neighbors (K-NN)

Dalam pembelajaran *machine learning* terdapat beberapa jenis metode untuk melakukan klasifikasi, contohnya yaitu metode *K-Nearest Neighbor* (K-NN). Metode K-NN adalah metode menemukan nilai k objek data atau pola yang paling dekat dengan pola masukan, kemudian memilih kelas dengan nilai k tertinggi di antara nilai k pada pola tersebut[9]. Algoritma ini juga dikenal sebagai *machine learning* yang malas (*lazy learning*)[10]. KNN juga merupakan salah satu algoritma klasifikasi yang sederhana namun efektif dan memiliki beberapa kelebihan, seperti kemudahan implementasi dan interpretasi, serta tidak memerlukan asumsi distribusi data. Namun, algoritma ini juga memiliki kekurangan, terutama dalam hal efisiensi pada dataset besar, karena memerlukan waktu komputasi yang lebih lama untuk menghitung jarak antara data. Persamaan (1) menunjukkan rumus perhitungan dari *K-Nearest Neighbor*.

$$Euclidien\ distance = \sqrt{\sum_{i=1}^{p}(a_k - b_k)^2} \quad (1)$$

Dimana a_k merupakan data sampel, b_k merupakan data uji testing, kemudian, p yang berarti dimensi data dan i merupakan variabel data.

2.3.2 Naive Bayes

Metode Naïve bayes merupakan metode *machine learning* yang didasarkan pada *Teorema Bayes* dengan setiap atribut independen atau tidak saling berhubungan, dapat digunakan untuk mengklasifikasikan probabilitas sederhana. Metode ini efektif untuk mengestimasi parameter dengan jumlah data pelatihan yang relatif kecil[11]. Metode klasifikasi *Naive Bayes* digunakan untuk membuat keputusan dengan melakukan prediksi berdasarkan hasil dari klasifikasi yang telah di peroleh[12]. Salah satu keunggulan utama algoritma Naïve Bayes adalah mudah digunakan dan sering memberikan hasil yang baik dalam berbagai kondisi. Selain itu, algoritma ini membutuhkan jumlah data pelatihan yang relatif kecil untuk mengestimasi parameter selama proses klasifikasi[13]. Namun, kekurangan metode ini adalah asumsi bebas antar fitur yang seringkali ada dan tidak dapat dimodelkan.

$$P(X|H) = \frac{P(X|H)P(H)}{P(H)} \quad (2)$$

Pada persamaan (2) dijelaskan: P(H|E) adalah probabilitas akhir bersyarat (*conditional probability*) suatu hipotesis H terjadi jika ada bukti E (*evidence*). P(E|H) adalah probabilitas bukti E terjadi tanpa mempertimbangkan hipotesis atau bukti lainnya.

2.3.3 Decision Tree

Decision Tree adalah salah satu cara untuk memprediksi suatu masalah dengan membuat pohon keputusan, kemudian memecahnya menjadi kumpulan yang lebih kecil, dan dapat meningkatkan proses pengambilan keputusan secara bertahap[14]. Algoritma *Decision Tree* adalah salah satu jenis algoritma *machine learning* yang digunakan untuk mengeksplorasi data untuk menemukan hubungan antara sejumlah variabel input dengan variabel target. Algoritma ini termasuk *Supervised Learning* yang terdiri dari node yang mewakili struktur cabang, kumpulan data yang dapat menunjukkan keputusan algoritma dan hasil yang diwakili oleh simpul daun. Dalam algoritma *Decision Tree*, setiap node digambarkan sebagai atribut, setiap cabang digambarkan sebagai nilai atribut, dan daun digambarkan sebagai kelas atau prediksi akhir[4]. *Decision Tree* memiliki kelebihan karena dapat menangani data yang tidak terstruktur dan mudah diinterpretasikan, sehingga mudah dipahami oleh pengguna. Namun, algoritma ini dapat overfitting, terutama pada pohon yang sangat dalam. Akibatnya, metode pruning sering digunakan untuk mengatasi masalah ini.

2.3.4 Random Forest

Dalam berbagai jenis penelitian dan model kasus, *Random forest* adalah salah satu teknik klasifikasi (*supervised classification*) yang paling banyak digunakan[15]. *Random Forest* juga disebut RF, adalah metode *machine learning* yang digunakan untuk mengoptimalkan nilai akurasi dalam kasus klasifikasi data. Teknik ini digunakan untuk melewati proses penggabungan banyak pemilah dari pendekatan yang serupa dan menghasilkan prediksi klasifikasi akhir melalui proses voting. Dalam kasus *Random Forest*, banyak pohon digunakan untuk membuat hutan, dan setiap pohon dipelajari secara bertahap[5]. Kelebihan dari algoritma *Random forest* yaitu mencakup kemampuan untuk menangani data dengan noise dan fitur yang tidak relevan, serta memberikan estimasi pentingnya fitur. Meskipun demikian, *Random Forest* dapat menjadi kurang interpretatif dibandingkan dengan *Decision Tree* dan memerlukan lebih banyak waktu komputasi.

2.3.5 Support Vector Machine (SVM)

SVM adalah mekanisme *machine learning* yang memakai ruangan tesis dan terbagi dalam beberapa fungsi linear yang dilatih algoritma evaluasi berdasarkan teori optimisasi dengan dimensi tinggi [16]. Keunggulan dari SVM terletak pada kemampuannya untuk bekerja dengan baik dalam ruang dimensi tinggi dan efisiensinya dalam menemukan *hyperplane* yang optimal untuk memisahkan data [8]. *Hyperplane* yaitu sebuah peranan yang bisa dipakai untuk pembatas antara kelas. Dalam masalah klasifikasi, SVM mencoba menemukan *hyperplane* terbaik yang dapat memisahkan dua kelas data dengan margin maksimum. Margin adalah jarak antara *hyperplane* dengan data terdekat dari masing-masing kelas. Data terdekat ini disebut sebagai *support vector*. Semakin besar margin, semakin baik kemampuan generalisasi dari model SVM. Namun, SVM juga memiliki kekurangan, yaitu waktu pelatihan yang lama pada dataset besar dan kesulitan dalam memilih parameter yang tepat, seperti kernel dan regularisasi.

2.4 Evaluasi Performa

Tahapan ini bertujuan untuk mengevaluasi hasil prediksi yang dihasilkan oleh kelima algoritma *machine learning* yang telah digunakan. Evaluasi performa ini dilakukan dengan menggunakan *confusion matrix* yang meliputi akurasi, *precission*, *recall* dan *F1-Score*. Akurasi adalah perbandingan antara informasi yang benar yang diberikan sistem kepada keseluruhan data, seperti yang digambarkan dalam persamaan (3). Sementara itu, presisi atau ketepatan, adalah ketepatan nilai antara permintaan pengguna pada respons sistem, seperti yang digambarkan dalam persamaan (4). Selanjutnya, recall adalah ketepatan antara informasi yang sama dengan informasi yang pernah dipanggil sebelumnya, seperti yang digambarkan dalam persamaan (5). Terakhir adalah F1-Score yang merupakan perbandingan rata-rata pada presisi dan recall yang dibobotkan dengan rumus dalam persamaan (6)[2]. Metrik ini memberikan wawasan mendalam tentang kekuatan dan kelemahan setiap algoritma yang digunakan untuk memprediksi kelulusan siswa.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Presisi = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1 - Score = 2.\frac{Presisi.Recall}{Presisi+Recall} \quad (6)$$

Dimana TP (*True Positive*) adalah prediksi positif yang benar, sedangkan TN *(True Negative)* adalah prediksi negatif yang benar. Selanjutnya, FP (*False Positive*) adalah prediksi positif yang salah, dan FN (*False Negative*) adalah prediksi negatif yang salah.

3. HASIL DAN PEMBAHASAN

Dataset yang digunakan untuk penelitian ini memiliki tiga kategori fitur, yaitu: numerikal, kategorikal, dan keseluruhan (gabungan numerikal dan kategorikal). Analisis ini dilakukan untuk membandingkan kinerja lima algoritma klasifikasi yaitu, *K-Nearest Neighbors* (K-NN), *Naïve Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM) berdasarkan *confusion matrix* yang meliputi akurasi, presisi, recall, dan *F1-Score*.

3.1 Hasil Pengujian pada Dataset Numerikal

Pengujian pertama dilakukan pada dataset numerikal yang bertujuan untuk menganalisis berbagai variabel yang mempengaruhi hasil pendidikan. Fitur yang digunakan pada pengujian ini meliputi: *Application mode* yaitu metode penerimaan siswa, *Course* yang menunjukkan program studi yang diambil, *Previous qualification* yaitu latar belakang pendidikan sebelum masuk perguruan tinggi, *Previous qualification (grade)* nilai akhir dari pendidikan sebelumnya, *Nacionality* kewarganegaraan siswa, *Mother's qualification and Father's qualification* yaitu tingkat pendidikan terakhir ibu dan ayah, *Mother's occupation and Father's occupation* yaitu jenis pekerjaan ibu dan ayah, *Admission grade* yaitu nilai masuk siswa ke perguruan tinggi, *Age at enrollment* yaitu usia saat pertama kali mendaftar, *Curricular units 1st sem (enrolled):* jumlah mata kuliah yang diambil di semester pertama, *Curricular units 1st sem (evaluations):* jumlah evaluasi atau ujian yang diikuti di semester pertama, *Curricular units 1st sem (approved):* jumlah mata kuliah yang lulus di semester pertama, *Curricular units 1st sem (grade)* yaitu nilai rata-rata di semester pertama, *Curricular units 2nd sem (enrolled)* yaitu jumlah mata kuliah yang diambil di semester kedua, *Curricular units 2nd sem (evaluations)* yaitu jumlah evaluasi atau ujian yang diikuti di semester kedua, *Curricular units 2nd sem (approved)* yaitu jumlah mata kuliah yang lulus di semester kedua, *Curricular units 2nd sem (grade)* yaitu nilai rata-rata di semester kedua, *Unemployment rate*: tingkat pengangguran pada tahun studi siswa, *Inflation rate* yaitu tingkat inflasi yang dapat mempengaruhi kondisi ekonomi siswa, GDP: produk domestik bruto yang menggambarkan kondisi ekonomi suatu negara, dan Target yaitu hasil akhir yang menunjukkan status akademik siswa, seperti *dropout, graduate* dan *enrolled*. Berikut adalah contoh data numerikal yang bisa dilihat pada Gambar 2,3 dan 4.

	Application mode	Course	Previous qualification	Previous qualification (grade)	Nacionality	Mother's qualification	Father's qualification	Mother's occupation	Father's occupation	Admission grade
2	17	171	1	122.0	1	19	12	5	9	127.3
3	15	9254	1	160.0	1	1	3	3	3	142.5
4	1	9070	1	122.0	1	37	37	9	9	124.8
5	17	9773	1	122.0	1	38	37	5	3	119.6
6	39	8014	1	100.0	1	37	38	9	9	141.5
7	39	9991	19	133.1	1	37	37	9	7	114.8
8	1	9500	1	142.0	1	19	38	7	10	128.4
9	18	9254	1	119.0	1	37	37	9	9	113.1
10	1	9238	1	137.0	62	1	1	9	9	129.3
11	1	9238	1	138.0	1	1	19	4	7	123.0
12	1	9670	1	139.0	1	38	19	5	7	130.6
13	1	9500	1	136.0	1	19	38	9	9	119.3
14	1	9853	1	133.0	1	19	37	4	9	130.2
15	53	9254	42	110.0	1	1	1	4	7	111.8
16	1	9085	1	149.0	1	38	37	5	5	137.1
17	1	9773	1	127.0	1	19	37	9	3	120.7
18	18	9238	1	137.0	1	19	38	5	8	137.4
19	17	9500	1	135.0	1	19	1	5	4	127.3
20	1	9130	1	137.0	1	3	19	3	5	136.3
21	1	9853	1	140.0	1	19	19	7	7	124.6
22	1	171	1	122.0	1	1	1	9	8	120.3
23	18	9556	1	127.0	1	1	38	4	7	121.8
24	1	9500	1	142.0	1	19	19	1	1	125.5
25	1	9670	1	125.0	1	1	38	4	7	114.9
26	1	9500	1	126.0	1	19	19	3	7	123.9
27	1	9238	1	151.0	1	19	38	9	9	157.0
28	17	9238	1	115.0	1	19	38	7	8	116.4

Gambar 2. Data Numerikal

	Age at enrollment	Curricular units 1st sem (enrolled)	Curricular units 1st sem (evaluations)	Curricular units 1st sem (approved)	Curricular units 1st sem (grade)	Curricular units 2nd sem (enrolled)
2	20	0	0	0	0.0	0
3	19	6	6	6	14.0	6
4	19	6	0	0	0.0	6
5	20	6	8	6	13.428.571.428.571.400	6
6	45	6	9	5	12.333.333.333.333.300	6
7	50	5	10	5	11.857.142.857.142.800	5
8	18	7	9	7	13.3	8
9	22	5	5	0	0.0	5
10	21	6	8	6	13.875	6
11	18	6	9	5	11.4	6
12	18	6	6	6	12.333.333.333.333.300	6
13	18	8	8	7	13.214.285.714.285.700	8
14	19	6	6	0	0.0	6
15	21	6	7	6	10.571.428.571.428.500	6
16	18	5	7	4	13.25	5
17	20	6	6	5	13.2	6
18	18	6	10	1	12.0	6
19	18	7	8	7	1.330.625	8
20	20	5	8	4	12.5	5
21	18	7	7	6	11.666.666.666.666.600	7
22	21	0	0	0	0.0	0
23	20	7	14	7	114.375	8
24	18	8	12	7	12.857.142.857.142.800	8
25	19	6	8	6	13.375	6
26	19	7	8	6	13.296.666.666.666.600	8
27	18	6	8	5	11.6	6
28	21	6	9	6	11.375	6

Gambar 3. Data Numerikal

	Curricular units 2nd sem (evaluations)	Curricular units 2nd sem (approved)	Curricular units 2nd sem (grade)	Unemployment rate	Inflation rate	GDP	Target
2	0	0	0.0	10.8	1.4	1.74	Dropout
3	6	6	13.666.666.666.666.600	13.9	-0.3	0.79	Graduate
4	0	0	0.0	10.8	1.4	1.74	Dropout
5	10	5	12.4	9.4	-0.8	-3.12	Graduate
6	6	6	13.0	13.9	-0.3	0.79	Graduate
7	17	5	11.5	16.2	0.3	-0.92	Graduate
8	8	8	14.345	15.5	2.8	-4.06	Graduate
9	5	0	0.0	15.5	2.8	-4.06	Dropout
10	7	6	14.142.857.142.857.100	16.2	0.3	-0.92	Graduate
11	14	2	13.5	8.9	1.4	3.51	Dropout
12	7	5	14.2	13.9	-0.3	0.79	Graduate
13	8	7	13.214.285.714.285.700	12.7	3.7	-1.7	Graduate
14	0	0	0.0	12.7	3.7	-1.7	Dropout
15	8	5	11.0	8.9	1.4	3.51	Graduate
16	5	5	12.0	10.8	1.4	1.74	Graduate
17	7	0	0.0	15.5	2.8	-4.06	Dropout
18	14	2	11.0	10.8	1.4	1.74	Enrolled
19	8	8	14.545	15.5	2.8	-4.06	Graduate
20	8	4	12.25	10.8	1.4	1.74	Graduate
21	8	6	13.5	16.2	0.3	-0.92	Enrolled
22	0	0	0.0	11.1	0.6	2.02	Graduate
23	9	8	11.425	12.7	3.7	-1.7	Enrolled
24	12	7	12.857.142.857.142.800	12.7	3.7	-1.7	Graduate
25	7	6	12.285.714.285.714.200	11.1	0.6	2.02	Graduate
26	9	7	14.114.285.714.285.700	11.1	0.6	2.02	Graduate
27	12	4	11.0	7.6	2.6	0.32	Enrolled
28	9	6	13.285.714.285.714.200	16.2	0.3	-0.92	Graduate

Gambar 4. Data Numerikal

Dari data pada Gambar 2, 3, dan 4 kemudian akan dilakukan *pre-processing* data. Tahapan dari *pre-processing* data yang pertama adalah nilai non-numerik diubah menjadi numerik dan diisi dengan nilai 0 menggunakan kode fillna() seperti pada Gambar 5. Kemudian mengubah fitur kategori menjadi bentuk numerik dengan menggunakan LabelEncoder seperti pada Gambar 6. Yang terakhir data dibagi menjadi data *training* dan data *testing* dengan menggunakan *train_test_split* yang dapat dilihat pada Gambar 7 setelah dibagi menjadi data *training* dan data *testing* kemudian diinisialisasi model menggunakan kode pada Gambar 8.

```
# Tangani data non-numerik jika ada
for col in X.columns:
    if not pd.api.types.is_numeric_dtype(X[col]):
        try:
            X[col] = pd.to_numeric(X[col], errors='coerce')
            X[col].fillna(X[col].mean(), inplace=True)
        except ValueError:
            print(f"Warning: Could not convert column '{col}' to numeric. Dropping the column.")
            X = X.drop(col, axis=1)
```

Gambar 5. Kode Fillna

```
# Enkode fitur kategori dengan menggunakan Label Encoding
label_encoder = LabelEncoder()
for column in X.columns:
    if X[column].dtype == 'object':
        X[column] = label_encoder.fit_transform(X[column])
y = label_encoder.fit_transform(y)
```

Gambar 6. Kode *LabelEncoding*

```
# pisahkan data menjadi data training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Gambar 7. Kode *Train Test Split*

```
# Inisialisasi pengklasifikasi
classifiers = {
    "KNN": KNeighborsClassifier(),
    "Naive Bayes": GaussianNB(),
    "SVM": SVC(),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier()
}
```

Gambar 8. Kode Inisialisasi

Dari hasil perhitungan data diatas diperoleh hasil bahwa algoritma *Random Forest* adalah algoritma terbaik dengan akurasi sebesar 74.12%, kemudian diikuti oleh *Decision Tree* dengan akurasi sebesar 66.89%, dan SVM adalah yang terendah dengan akurasi sebesar 47.23%. *Random Forest* (RF) umumnya menunjukkan akurasi yang lebih tinggi dibandingkan metode lain karena kemampuannya membangun banyak *Decision Tree* dari berbagai subset data dan kemudian mengambil *mean* untuk meningkatkan akurasi prediksi. Berdasarkan nilai F1-Score, *Random Forest* unggul dengan nilai 72.43% yang menunjukkan kemampuan model untuk menangkap hubungan antar fitur numerikal, sementara *Support Vector Machine* (SVM) memiliki nilai terendah sebesar 30.30% karena kompleksitas dalam optimasi model dan sensitivitas terhadap parameter serta pemilihan fitur. Meskipun demikian, terdapat penelitian yang menunjukkan bahwa SVM dapat mencapai akurasi tertinggi dalam prediksi kelulusan mahasiswa tepat waktu dengan akurasi sebesar 0.94, tergantung pada atribut yang digunakan seperti jenis kelamin, usia, penghasilan orang tua, durasi bimbingan, status mahasiswa bekerja atau tidak, nilai semester 1 hingga semester 8 dan IPK[2]. Oleh karena itu, perbedaan kinerja antara *Random Forest* dan SVM sangat bergantung pada sifat dataset dan parameter optimasi yang digunakan. Berikut merupakan hasil dari kelima algoritma *machine learning* menggunakan dataset numerikal yang dapat dilihat pada Tabel 2.

Tabel 2. Hasil data numerikal

Nomor	Algoritma	Akurasi(%)	Presisi(%)	Recall(%)	F1-Score(%)
1	K-NN	59.10	57.69	59.10	57.91
2	*Naïve Bayes*	67.01	66.60	67.01	65.08
3	*Decision Tree*	66.89	68.09	66.89	67.41
4	*Random Forest*	**74.12**	72.45	**74.12**	**72.43**
5	SVM	47.23	**75.08**	47.23	30.30

3.2 Hasil Pengujian pada Dataset Kategorikal

Dalam pengujian kedua, dataset kategorikal digunakan untuk menilai berbagai fitur yang berkaitan dengan status dan kebutuhan pendidikan. Fitur yang diuji meliputi: *Marital status*, *Application order*, *Daytime/evening attendance*, *Displaced*, *Educational special needs*, *Debtor*, *Tuition fees up to date*, *Gender*, *Scholarship holder*, *International*, *Curricular units 1st sem (credited)*, *Curricular units 1st sem (without evaluations)*, *Curricular units 2nd sem (credited)*, dan *Curricular units 2nd sem (without evaluations).* Contoh data kategorikal bisa dilihat pada Gambar 9 dan 10.

Marital status	Application order	Daytime/evening attendance	Displaced	Educational special needs	Debtor	Tuition fees up to date	Gender	Scholarship holder	International	Curricular units 1st sem (credited)
1	5	1	1	0	0	1	1	0	0	0
1	1	1	1	0	0	0	1	0	0	0
1	5	1	1	0	0	0	1	0	0	0
1	2	1	1	0	0	1	0	0	0	0
2	1	0	0	0	0	1	0	0	0	0
2	1	0	0	0	1	1	1	0	0	0
1	1	1	1	0	0	1	0	1	0	0
1	4	1	1	0	0	0	1	0	0	0
1	3	1	0	0	0	1	0	1	1	0
1	1	1	1	0	1	0	0	0	0	0
1	1	1	1	0	0	1	0	0	0	0
1	1	1	1	0	0	1	0	1	0	0
1	2	1	1	0	0	1	0	0	0	0
1	1	1	1	0	0	1	0	1	0	0
1	1	1	1	0	0	1	0	1	0	0
1	1	1	1	0	0	1	0	0	0	0
1	1	1	1	0	0	1	0	0	0	0
1	2	1	1	0	0	1	0	0	0	0
1	1	1	1	0	0	1	0	0	0	0
1	1	1	1	0	0	1	0	0	0	0
1	3	1	0	0	0	1	0	1	0	0
1	4	1	1	0	0	1	0	0	0	0
1	4	1	1	0	0	1	0	0	0	0
1	4	1	1	0	0	1	0	1	0	0
1	1	1	0	0	0	1	0	0	0	0

Gambar 9. Data Kategorikal

Curricular units 1st sem (without evaluations)	Curricular units 2nd sem (credited)	Curricular units 2nd sem (without evaluations)
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	5
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
1	0	2
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Gambar 10. Data Kategorikal

Dari data yang telah dikumpulkan kemudian akan melalui tahap pengolahan awal yang dikenal sebagai *pre-processing.* Tahapan dari *pre-processing* data yang pertama adalah variabel target non-numerik diubah menjadi bentuk numerik dengan menggunakan *LabelEncoder*. Kode *google colab* bisa dilihat pada Gambar 11.

```
# Enkode fitur kategori dengan menggunakan Label Encoding
label_encoder = LabelEncoder()
for column in X.columns:
    if X[column].dtype == 'object':
        X[column] = label_encoder.fit_transform(X[column])
y = label_encoder.fit_transform(y)
```

Gambar 11. Kode *LabelEncoding*

Kemudian data dibagi menjadi data latih dan uji dengan menggunakan *train_test_split*. Kode dapat dilihat pada Gambar 12.

```
# pisahkan data menjadi data training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Gambar 12. Kode *Train Test Split*

Dan yang terakhir di inisialisasi model seperti kode pada Gambar 13.

```
# Inisialisasi pengklasifikasi
classifiers = {
    "KNN": KNeighborsClassifier(),
    "Naive Bayes": GaussianNB(),
    "SVM": SVC(),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier()
}
```

Gambar 13. Kode Inisialisasi

Hasil dari pengujian dataset kategorikal algoritma K-NN dan SVM menunjukkan hasil yang hampir sama, dengan nilai akurasi tertinggi sebesar 93.11%. Hasil ini menunjukkan bahwa kedua algoritma ini memiliki kemampuan untuk menangani data kategorikal dengan baik. *Random Forest* dan *Decision Tree* memiliki akurasi sebesar 92.88% dan 91.98%, masing-masing di bawah K-NN dan SVM. Sebaliknya, *Naïve Bayes* memiliki akurasi terendah sebesar 00.45% pada dataset ini. Rendahnya kinerja *Naïve Bayes* disebabkan oleh asumsi independensi antar fitur yang tidak terpenuhi dalam dataset ini. Ketika fitur-fitur dalam dataset saling berkaitan, algoritma *Naïve Bayes* kesulitan dalam memodelkan hubungan antar variabel secara akurat. Oleh karena itu, *Naïve Bayes* kurang efektif untuk digunakan pada dataset kategorikal yang memiliki fitur dengan keterkaitan tinggi. Hasil dari perhitungan data kategorikal dapat dilihat pada Tabel 4.

Tabel 4. Hasil data kategorikal

Nomor	Algoritma	Akurasi(%)	Presisi(%)	Recall(%)	F1-Score(%)
1	K-NN	**93.11**	88.31	**93.11**	90.02
2	*Naïve Bayes*	00.45	**93.02**	00.45	00.28
3	*Decision Tree*	91.98	88.68	91.98	90.19
4	*Random Forest*	92.88	88.71	92.88	**90.51**
5	SVM	**93.11**	86.79	**93.11**	89.84

3.3 Hasil Pengujian pada Dataset Keseluruhan

Dari data gabungan numerikal dan kategorikal pada Gambar 2, 3, 4, 9, dan 10 yang telah dikumpulkan kemudian akan melalui tahap *pre-processing* data. Tahapan yang pertama dari *pre-processing* data adalah mengubah variabel target non-numerik menjadi bentuk numerik dengan menggunakan LabelEncoder. Kode dapat dilihat pada Gambar 14. Kemudian data dipisahkan menjadi fitur dan target seperti kode pada Gambar 15. Untuk memastikan skala yang sama, variabel numerik distandarisasi dengan menggunakan "*StandardScaler*" seperti kode pada Gambar 16. Terakhir data dibagi menjadi data latih dan uji dengan menggunakan *train_test_split* kode dapat dilihat pada Gambar 17 dan kemudian diinisialisasi model seperti kode pada Gambar 18.

```
# Mengubah target menjadi numerik
label_encoder = LabelEncoder()
data['Target'] = label_encoder.fit_transform(data['Target'])
```

Gambar 14. Kode *LabelEncoder*

```
# Memisahkan fitur dan target
X = data.drop(columns=['Target'])
y = data['Target']
```

Gambar 15. Kode Fitur dan Target

```
# Standarisasi fitur numerik
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Gambar 16. Kode *StandardScaler*

```
# Membagi data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

Gambar 17. Kode *Train Test Split*

```
# Inisialisasi model
models = {
    "Naive Bayes": GaussianNB(),
    "SVM": SVC(kernel='linear', random_state=42),
    "Random Forest": RandomForestClassifier(random_state=42),
    "Decision Tree": DecisionTreeClassifier(random_state=42)
}
```

Gambar 18. Kode Inisialisasi

Hasil pengujian pada dataset keseluruhan (gabungan numerikal dan kategorikal) menunjukkan algoritma *Random Forest* memiliki kinerja terbaik dengan nilai akurasi sebesar 76.50%, presisi 75%, recall 76% dan F1-Scorenya 75.00%, sedangkan SVM mengikuti dengan akurasi sebesar 75.71%, presisi 74%, recall 76% dan F1-Score 74.00%. Algoritma lainnya, seperti K-NN dan *Decision Tree* menunjukkan kinerja yang kompetitif tetapi masih berada di bawah *Random Forest* dan SVM. Sebaliknya, algoritma *Naive Bayes* memiliki kinerja paling rendah dengan akurasi sebesar 67.80%, presisi 67.00%, recall 68.00%, dan F1-Score 67.00%, menunjukkan bahwa algoritma ini tidak dapat menangani secara bersamaan gabungan fitur numerikal dan kategorikal secara bersamaan. Keunggulan *Random Forest* dalam analisis ini dapat dikaitkan dengan kemampuannya mengolah data campuran secara efektif, melakukan seleksi fitur secara implisit, serta mengenali hubungan non-linear antar fitur tanpa memerlukan asumsi independensi yang ketat seperti pada *Naive Bayes*. Selain itu, struktur *ensemble* yang dimiliki *Random Forest* membantu mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi, menjadikannya lebih unggul dibandingkan model tunggal seperti *Decision Tree* serta lebih mudah dioptimalkan dibandingkan algoritma lain seperti SVM atau K-NN yang sering memerlukan penyesuaian parameter yang kompleks. Berikut hasil dari data keseluruhan dapat dilihat pada Tabel 5.

Tabel 5. Hasil data keseluruhan

Nomor	Algoritma	Akurasi(%)	Presisi(%)	Recall(%)	F1-Score(%)
1	K-NN	70.96	70.00	71.00	70.00
2	*Naïve Bayes*	67.80	67.00	68.00	67.00
3	*Decision Tree*	66.67	68.00	67.00	67.00
4	*Random Forest*	**76.50**	**75.00**	**76.00**	**75.00**
5	SVM	75.71	74.00	**76.00**	74.00

Hasil dari analisis dan perbandingan algoritma *machine learning* yang dilakukan dalam penelitian ini memberikan gambaran tentang seberapa efektif masing-masing algoritma dalam menangani berbagai jenis data. K-NN dan SVM terbukti unggul dalam menangani dataset kategorikal, sementara *Random Forest* menunjukkan performa yang lebih baik pada dataset numerikal dan keseluruhan karena kemampuannya dalam menangani fitur dengan kompleksitas tinggi serta interaksi antar variabel. *Decision Tree* memiliki performa yang cukup baik namun, cenderung lebih rentan terhadap *overfitting* dibandingkan *Random Forest*, sehingga akurasinya sedikit lebih rendah. Sebaliknya, *Naïve Bayes* memiliki keterbatasan dalam menangani dataset yang tidak memenuhi asumsi independensi antar fitur, sehingga kinerjanya jauh lebih rendah dibandingkan algoritma lainnya pada dataset kategorikal. Dengan memahami keunggulan dan keterbatasan masing-masing algoritma, hasil penelitian ini dapat dijadikan acuan dalam menentukan algoritma yang lebih tepat dalam memprediksi kelulusan siswa.

4. KESIMPULAN

Hasil pengujian menunjukkan bahwa kinerja algoritma klasifikasi secara signifikan dipengaruhi oleh jenis fitur data yang dipilih. Pada pengujian dataset numerikal menunjukkan bahwa algoritma *Random Forest* adalah algoritma terbaik dengan nilai akurasi sebesar 74.12% dan SVM adalah algoritma terendah dengan akurasi sebesar 47.23%. Berdasarkan nilai F1-Score, *Random Forest* tetap unggul dengan nilai 72.43%. Algoritma dengan fitur kategorikal memiliki akurasi tertinggi pada sebagian besar algoritma, terutama untuk K-NN dan SVM mendapatkan nilai akurasi tertinggi sebesar 93.11%. Namun, algoritma *Random Forest* memiliki performa yang paling konsisten dan unggul ketika seluruh fitur digabungkan. Dengan mendapatkan nilai akurasi tertinggi sebesar 76.50% dan F1-Scorenya 75.00%. Penelitian ini menekankan betapa pentingnya memahami karakteristik data dan memilih algoritma yang tepat untuk setiap jenis fitur agar model klasifikasi dapat bekerja lebih baik. Dengan hasil penelitian ini, lembaga pendidikan dapat mempertimbangkan untuk menggunakan *machine learning* sebagai alat bantu dalam pengambilan keputusan strategis tentang intervensi bagi siswa yang berisiko tidak lulus. Selain itu, penelitian ini membuka jalan bagi penelitian lebih lanjut tentang penerapan metode *machine learning* lainnya, seperti *deep learning* atau *hybrid models*, untuk lebih akurat memprediksi kelulusan siswa. Faktor-faktor tambahan seperti kualitas data input, pemilihan fitur yang tepat, dan parameter penyesuaian juga sangat penting dalam menentukan kinerja akhir dari masing-masing algoritma dalam penelitian ini.

UCAPAN TERIMAKASIH

Dengan penuh rasa syukur dan hormat, kami mengucapkan terima kasih yang mendalam kepada semua pihak yang telah menjadi bagian penting dalam perjalanan penelitian ini. Kepada para pembimbing, kami ucapkan terima kasih yang tiada tara atas bimbingan, ilmu, dan arahan yang begitu berharga. Kami juga ingin mengungkapkan rasa terima kasih yang tulus kepada keluarga tercinta, yang tanpa lelah memberikan cinta, doa, dan dukungan moral. Ketulusan dan kasih sayang kalian adalah kekuatan yang membuat kami tetap tegar menghadapi berbagai tantangan. Tak lupa, kepada para sahabat dan rekan sejawat yang telah berbagi semangat, ide, dan motivasi, kami sangat bersyukur atas kehadiran kalian semua. Kebaikan hati dan bantuan dari setiap pihak menjadi poin penting yang membuat penelitian ini dapat terselesaikan dengan baik. Semoga segala dukungan, kasih, dan perhatian yang telah diberikan mendapatkan balasan yang berlimpah.

REFERENCES

[1] S. Widaningsih, "Perbandingan Metode Data Mining Untuk Prediksi Nilai dan Waktu Kelulusan Mahasiswa Prodi Teknik Informatika dengan Algoritma C4,5, Naive Bayes, KNN dan SVM," *Jurnal Tekno Insentif*, vol. 13, no. 1, pp. 16–25, Apr. 2019, doi: 10.36787/jti.v13i1.78.

[2] Satrio Junaidi, R. Valicia Anggela, and D. Kariman, "Klasifikasi Metode Data Mining untuk Prediksi Kelulusan Tepat Waktu Mahasiswa dengan Algoritma Naïve Bayes, Random Forest, Support Vector Machine (SVM) dan Artificial Neural Nerwork (ANN)," *Journal of Applied Computer Science and Technology*, vol. 5, no. 1, pp. 109–119, Jun. 2024, doi: 10.52158/jacost.v5i1.489.

[3] N. Hamdani and A. Setyanto, "Perbandingan Algoritma Regresi Logistic Dan Neural Network Pada Prediksi Nilai Hasil Pembinaan Dan Kelulusan Tepat Waktu," *Respati*, 2020.

[4] A. M. Majid and I. Nawangsih, "Perbandingan Metode Ensemble Untuk Meningkatkan Akurasi Algoritma Machine Learning Dalam Memprediksi Penyakit Breast Cancer (Kanker Payudara)," *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*, vol. 23, pp. 97–104, 2024, [Online]. Available: https://ojs.trigunadharma.ac.id/index.php/jis/index

[5] J. Zeniarja, A. Salam, and F. A. Ma'ruf, "Seleksi Fitur dan Perbandingan Algoritma Klasifikasi untuk Prediksi Kelulusan Mahasiswa," *Jurnal Rekayasa Elektrika*, vol. 18, no. 2, Jul. 2022, doi: 10.17529/jre.v18i2.24047.

[6] F. A. Bachtiar, I. K. Syahputra, and S. A. Wicaksono, "Perbandingan Algoritme Machine Learning Untuk Memprediksi Pengambil Mata Kuliah," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 6, no. 5, pp. 543–548, 2019, doi: 10.25126/jtiik.2019611755.

[7] D. Agustinawati, "Perbandingan Metode Data Mining pada Prediksi Kelulusan Mahasiswa Fakultas Teknologi Informasi di Perguruan Tinggi dengan Algoritma Naive Bayes dan K-Nearest Neighbor," *Jurnal SIMETRIS*, vol. 15, no. 1, 2024.

[8] M. J. Prasetyo, I. Made, and A. Agastya, "Analisis Sentimen Ulasan Aplikasi Perbankan di Google Play Store menggunakan Algoritma Support Vector Machine Sentiment Analysis of Banking Application Reviews on Google Play Store using Support Vector Machine Algorithm," *Sistemasi: Jurnal Sistem Informasi*, 2024, [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[9] N. Wakhidah, S. N. Rochmah,) Fakultas, and R. Artikel, "Klasifikasi kualitas mutu susu pasteurisasi menggunakan metode klasifikasi k-Nearest Neighbor," *AITI: Jurnal Teknologi Informasi*, vol. 21, no. Maret, pp. 58–71, 2024.

[10] A. A. Bagaskara and K. D. Hartomo, "Klasifikasi Daerah Rawan Banjir menggunakan 10-Fold Cross Validation dan K-Nearest Neighbors Classification of Flood-Prone Areas Using 10-Fold Cross Validation and K-Nearest Neighbors," *SISTEMASI: Jurnal Sistem Informasi*, 2024, [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[11] D. Abisono Punkastyo, F. Septian, and A. Syaripudin, "Implementasi Data Mining Menggunakan Algoritma Naïve Bayes Untuk Prediksi Kelulusan Siswa," 2024.

[12] W. Ningsih, B. Alfianda, R. Rahmaddeni, and D. Wulandari, "Perbandingan Algoritma SVM dan Naïve Bayes dalam Analisis Sentimen Twitter pada Penggunaan Mobil Listrik di Indonesia," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 556–562, Feb. 2024, doi: 10.57152/malcom.v4i2.1253.

[13] Chely Aulia Misrun, E. Haerani, M. Fikry, and E. Budianita, "Analisis sentimen komentar youtube terhadap Anies Baswedan sebagai bakal calon presiden 2024 menggunakan metode naive bayes classifier," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 4, no. 1, pp. 207–215, Apr. 2023, doi: 10.37859/coscitech.v4i1.4790.

[14] Nurul Chairunnisa, "Prediksi Kemampuan Pembayaran Klien Home Credit Menggunakan Model Random Forest, Decision Tree, Dan Logistic Regression," *Jurnal Elektronika dan Teknik Informatika Terapan (JENTIK)*, vol. 1, no. 3, pp. 140–147, Aug. 2023, doi: 10.59061/jentik.v1i3.383.

[15] S. Budiman, A. Sunyoto, and A. Nasiri, "Analisa Performa Penggunaan Feature Selection untuk Mendeteksi Intrusion Detection Systems dengan Algoritma Random Forest Classifier," *SISTEMASI: Jurnal Sistem Informasi*, vol. 10, 2021, [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[16] Lukman and Herlinda, "Prediksi Kelulusan Siswa dengan Metode Support Vector Machine (SVM) di SMK Adiluhur," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 9, 2024.