



## PERBANDINGAN KLASIFIKASI LABEL TUNGGAL UNTUK SOAL UJIAN FISIKA MENGGUNAKAN NAÏVE BAYES DAN *K-FOLD CROSS VALIDATION*

Christopher Kevin Herijanto<sup>1</sup>, Yulia Wahyuningsih<sup>2</sup>

<sup>1,2</sup> Ilmu Informatika, Universitas Katolik Darma Cendika  
Surabaya, Jawa Timur, Indonesia 60117  
Christopher.herijanto@student.ukdc.ac.id, yulia@ukdc.ac.id

### Abstract

*This research evaluates the use of the Naïve Bayes algorithm in classifying Physics questions with single labels. The main objective is to identify the best algorithm for classifying Physics questions to assist high school students with difficulty understanding them. The research method involves using a dataset containing Physics questions that need to be classified to facilitate learning for high school students. The Naïve Bayes algorithm is implemented using Google Colab to train the classification model using features extracted from the text of the Physics questions. Additionally, several other classification algorithms, such as Support Vector Machine (SVM), Logistic Regression, Decision Tree, and Random Forest, are tested, and their performance is compared. Experimental results show that Naïve Bayes provides competitive results in classifying single-label Physics questions. However, there are significant performance differences between Naïve Bayes and other algorithms, depending on the type and complexity of the classified Physics problems. In this study, SVM achieved higher accuracy, but Naïve Bayes excelled in training time. This research provides a deeper understanding of the strengths and weaknesses of Naïve Bayes in solving the task of classifying single-label Physics problems. These findings guide the development of more accurate classification models for application in the context of Physics learning.*

**Keywords:** Machine Learning, Naïve Bayes, Physics, Random Forest, Single Label Classification

### Abstrak

Penelitian ini mengevaluasi penggunaan algoritma Naïve Bayes dalam mengklasifikasikan soal Fisika dengan label tunggal. Tujuan utamanya adalah untuk mendapatkan algoritma terbaik untuk mengklasifikasikan soal fisika yang akan digunakan untuk siswa SMA yang kesulitan untuk memahaminya. Metode penelitian ini menggunakan *dataset* yang berisi soal-soal Fisika yang perlu diklasifikasikan untuk memudahkan pembelajaran siswa SMA. Algoritma Naïve Bayes diimplementasikan menggunakan *Google Collab* untuk melatih model klasifikasi menggunakan fitur-fitur yang diekstraksi dari teks soal Fisika. Selain itu, beberapa algoritma klasifikasi lain seperti *Support Vector Machine (SVM)*, *Logistic Regression*, *Decision Tree*, dan *Random Forest* juga diuji dan dibandingkan kinerjanya. Hasil eksperimen menunjukkan bahwa Naïve Bayes memberikan hasil yang bersaing dalam klasifikasi soal Fisika label tunggal. Namun, terdapat perbedaan performa yang signifikan antara Naïve Bayes dan algoritma lainnya, bergantung pada jenis dan kompleksitas masalah soal Fisika yang diklasifikasikan. Pada penelitian ini, SVM memiliki nilai akurasi yang lebih tinggi, tetapi Naïve Bayes unggul dalam *training time*. Penelitian ini memberikan pemahaman lebih dalam tentang kelebihan dan kelemahan Naïve Bayes dalam menyelesaikan tugas klasifikasi masalah Fisika label tunggal. Temuan ini menjadi panduan untuk mengembangkan model klasifikasi yang lebih akurat untuk aplikasi dalam konteks pembelajaran Fisika.

**Kata kunci:** Fisika, Klasifikasi Label Tunggal, Naïve Bayes, Pembelajaran Mesin, *Random Forest*

### 1. PENDAHULUAN

Klasifikasi teks adalah proses mengklasifikasikan dokumen ke dalam kategori yang berbeda[1]. Klasifikasi teks memiliki banyak fungsi, seperti klasifikasi topik, analisis sentimen, deteksi *spam*, dan pemfilteran informasi. Dalam *machine learning*, proses klasifikasi teks dapat dibagi menjadi empat bagian umum: ekstraksi fitur, reduksi

dimensi, pemilihan pengklasifikasi, dan penilaian. Proses ekstraksi fitur mengubah data teks menjadi format yang dapat digunakan di komputer. Banyak metode yang telah dikembangkan untuk klasifikasi, seperti Naïve Bayes, *Random Forest*, *SVM*, *Decision Tree*, *Logistic Regression*, dan algoritma lain yang dapat digunakan untuk klasifikasi. Mengumpulkan kumpulan data multi-label adalah hal yang

lebih sulit dan kurang terukur dibandingkan mengumpulkan satu label *dataset*, karena pengumpulannya konsisten dan menyeluruh daftar label untuk setiap gambar memerlukan usaha yang signifikan[2].

Klasifikasi terhadap soal Fisika dibutuhkan karena adanya kesulitan bagi siswa SMA untuk memahami soal-soal fisika yang dikarenakan ada banyaknya rumus dan konsep yang harus dipahami[3]. Maka dari itu, klasifikasi sangat bermanfaat untuk memberikan solusi yang lebih efisien pada tingkat pemahaman siswa terhadap soal fisika.

Naïve Bayes adalah metode klasifikasi yang mengasumsikan variabel tidak bergantung satu sama lain [4][5]. Asumsi ini menjadi dasar proses klasifikasi Naïve Bayes. Naïve Bayes dikembangkan oleh Thomas Bayes sebagai metode pembelajaran terawasi berbasis teori. Naïve Bayes juga menjadi metode pengambilan keputusan dengan perhitungan probabilitas[6]. Naïve Bayes sendiri untuk klasifikasi teks bisa dikatakan sebagai metode dengan hasil yang baik dengan kalkulasi yang tidak rumit. Model pembelajaran ini mengasumsikan bahwa variabel-variabelnya bersifat independen. Misalnya kelas  $c$  pada himpunan kelas  $C$  mempunyai fitur  $t$ , dimana  $t$  merupakan subset dari himpunan fitur  $T$ . Oleh karena itu, probabilitas kelas  $c$  dengan fitur  $t$  mengikuti teori *Bayesian* pada Rumus 1[1].

$$\Pr(c|t) = \frac{\Pr(c)\Pr(t|c)}{\Pr(t)} \quad (1)$$

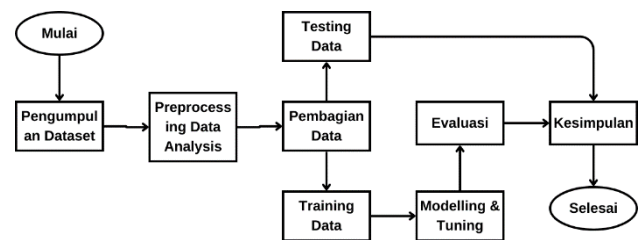
*Logistic Regression* adalah model linier yang biasa digunakan untuk klasifikasi biner yang menggunakan fungsi logistik untuk memodelkan ketergantungan biner *variable* [5][7]. *Random Forest* adalah algoritma pembelajaran ansambel yang menggunakan dan membangun struktur pohon secara bertahap. Jika digunakan, pohon keputusan dibuat dengan memilih atau mengambil sampel data secara acak. Untuk menentukan kelas data, *Random Forest* menggunakan sistem *voting* sebanyak hasil berdasarkan pohon keputusan[7]. *Support Vector Machine* (SVM) adalah kumpulan teknik pembelajaran terawasi yang membuat *hyperplanes* atau serangkaian *hyperplanes* dalam proses klasifikasi, regresi, dan deteksi *outlier*. Salah satu kegunaannya adalah untuk mengelompokkan teks dan *hypertext*. Kelebihan SVM ini adalah efektif pada ruang berdimensi tinggi, efektif bila jumlah dimensi lebih besar dibandingkan jumlah sampel, dan hemat memori karena menggunakan sub set titik pelatihan[8].

Menurut penelitian sebelumnya, dari hasil analisis seluruh pengujian yang dilakukan pada penelitian klasifikasi multi-label untuk topik berita Indonesia, dapat diambil kesimpulan model klasifikasi multi-label menggunakan metode *multinomial* Naïve Bayes yang memberikan hasil paling rendah. Pada penelitian tersebut, didapatkan hasil dari metode *hamming loss* yaitu 0,18. Selanjutnya model *multinomial* Naïve Bayes dengan menggunakan teknik *stemming* pada bagian *preprocessing* mencapai hasil yang lebih baik dibandingkan tanpa menggunakan teknik *stemming* pada bagian *preprocessing*[9]. Sebagai contoh

penerapan algoritma Naïve Bayes, hasil perbandingan penerapan algoritma klasifikasi *K-Nearest Neighbors* dan Naïve Bayes *Classifier* pada analisis sentimen dari dataset opini pengguna Twitter mengenai infeksi virus corona baru, algoritma *K-Nearest Neighbors* mencapai akurasi sebesar 72,37%, sedangkan algoritma Naïve Bayes mencapai akurasi sebesar 67,84% [10].

Selanjutnya, apakah algoritma Naïve Bayes juga merupakan algoritma yang efisien dalam membangun sistem klasifikasi untuk mengklasifikasikan data masalah fisik ke dalam label, dan apakah algoritma Naïve Bayes merupakan model yang baik untuk masalah klasifikasi multi-label. Timbul pertanyaan apakah algoritma tersebut bisa berhasil beradaptasi dengan Oleh karena itu, fokus penelitian ini adalah mengembangkan sistem klasifikasi teks multi-label pada data permasalahan fisika dengan mengimplementasikan algoritma Naïve Bayes dan menganalisis pengaruh jenis teknik transformasi permasalahan yang digunakan. Hasil klasifikasi yang dihasilkan juga dianalisis untuk memahami sumber kesalahan model saat memprediksi kelas dan label dari data. Metode evaluasi yang digunakan dalam penelitian ini adalah perbandingan hasil pelatihan dan tes serta *K-fold cross validation*.

## 2. METODE PENELITIAN



Gambar 1. Tahapan Metode Penelitian

Tahapan metode penelitian seperti yang terlihat pada gambar 1 dilakukan dari pengumpulan *dataset*, studi literatur, menentukan *basic statistic*, melakukan *preprocessing & exploratory* data, melakukan SMOTE (*Synthetic Minority Oversampling Technique*) atau *balancing dataset*, dan pengujian model. Terdapat 4 teknik pengolahan data yang digunakan yaitu *case folding*, *stopword removal*, *tokenizing* dan *stemming*. Setelah itu dilakukan pengujian dengan melihat hasil dari akurasi, *training time*, dan *prediction time*.

### 2.1 Dataset Collection

Studi kasus ini berfokus terutama pada ujian tingkat sekolah dasar di Indonesia, dan memerlukan data primer untuk dipusatkan pada soal-soal ujian fisika yang cocok untuk siswa SMA. *Input*-nya berupa data diekstrak dari sekolah-sekolah di Surabaya. *Dataset* yang digunakan adalah kumpulan soal-soal Fisika sebanyak 583 soal.

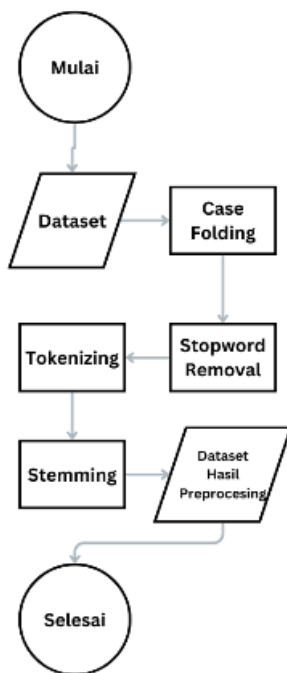
### 2.2 Klasifikasi

Suatu metode dalam *data mining* yang beroperasi dengan mengidentifikasi karakteristik dari kelompok data yang telah diketahui dan ditentukan sebelumnya disebut sebagai klasifikasi. Klasifikasi memiliki salah satu ciri khas, yaitu adanya kelas atau target yang ditetapkan. Teknik ini berfungsi dengan cara mengelompokkan data ke dalam kelas atau target tertentu untuk membentuk suatu model atau aturan[11]. Model atau aturan yang telah dibentuk ini kemudian dapat digunakan untuk mengklasifikasikan data baru ke dalam kelas atau target yang sesuai pada periode yang akan datang.

### 2.3 Basic Statistics

Pada proses ini, dilakukan perhitungan *statistic* deskriptif terhadap *dataset* untuk mengetahui nilai pemusatan data (*mean, median, modus*), ukuran letak data (kuartil dan desil), dan *standard deviation*. Statistik deskriptif pada dasarnya digunakan untuk memberikan pemahaman yang lebih mendalam tentang karakteristik dari *dataset* yang digunakan dalam klasifikasi label tunggal. Statistik deskriptif dapat membantu dalam menilai kebersihan data, mendeteksi nilai-nilai yang hilang, atau *outlier* yang mungkin mempengaruhi kualitas model klasifikasi.

### 2.4 Preprocessing



Gambar 2. Tahapan Preprocessing

*Dataset* yang digunakan pada tahapan seperti yang terlihat di gambar 2 ini berisi data yang masih belum terstruktur. Pada fase ini data akan dibersihkan terlebih dahulu dan mendapatkan hasil yaitu data yang sudah terstruktur. Penerapan metode *preprocessing* pada kumpulan data tertentu memungkinkan penyempurnaan teks secara signifikan kualitas klasifikasi dengan peningkatan yang dapat dicapai hingga 80%. Ada beberapa proses yang harus dilakukan yaitu *case folding, stopwords removal, tokenizing* dan *stemming*.

Untuk proses *case folding*, *input*-nya berupa teks lengkap artikel, dan *output*-nya berupa teks dengan rata-rata huruf kecil tanpa tanda baca. Hal ini dilakukan untuk menghindari ketidaksamaan nilai yang terjadi ketika menggunakan huruf besar dan kecil dalam kata yang sama[8][9]. Contoh hasil dari proses *case folding* dapat dilihat di tabel 1.

Tabel 1. Hasil dari Proses Case Folding

| Soal  | Contoh Hasil Proses Case Folding                  |
|---|---|
| \tSebuah bola besi digantungkan pada langit...    | sebuah bola besi digantungkan pada langit lang... |
| \tYang bukan merupakan pasangan gaya aksi...      | yang bukan merupakan pasangan gaya aksi reaksi... |
| \tPerhatikan pilihan gambar tentang seorang...    | perhatikan pilihan gambar tentang seorang anak... |
| \tPada gambar di bawah <br> <br> pasangan...      | pada gambar di bawah pasangan gaya aksi dan re... |
| \tSebuah benda digantungkan pada langit-langit... | sebuah benda digantungkan pada langit langit...   |

Setelah itu dilakukan proses *stopword removal* yang merupakan proses penghapusan kata-kata yang dianggap berdampak kecil, seperti kata yang memiliki tingkat kemunculan yang lebih tinggi dibandingkan dengan kata-kata lainnya[12]. Contoh hasil dari proses *stopword removal* dapat dilihat di tabel 2.

Tabel 2. Hasil dari Proses Stopword Removal

| Soal  | Contoh Hasil Proses Stopword Removal        |
|---|---|
| \tSebuah bola besi digantungkan pada langit...    | bola, besi, digantungkan, langit            |
| \tYang bukan merupakan pasangan gaya aksi...      | pasangan, gaya, aksi, reaksi                |
| \tPerhatikan pilihan gambar tentang seorang...    | perhatikan, pilihan, gambar, anak, ditarik  |
| \tPada gambar di bawah <br> <br> pasangan...      | gambar, pasangan, gaya, aksi, reaksi        |
| \tSebuah benda digantungkan pada langit-langit... | benda, digantungkan, langit, langit, gambar |

*Tokenizing* adalah proses pemecahan kalimat dari setiap dokumen pada *dataset* yang sudah dilakukan pembersihan ke dalam kata per kata[13]. Contoh hasil dari proses *tokenizing* dapat dilihat di tabel 3.

Tabel 3. Hasil dari Proses Tokenizing

| Soal   | Contoh Hasil Proses Tokenizing                  |
|--|---|
| \tSebuah bola besi digantungkan pada langit... | sebuah, bola, besi, digantungkan, pada, langit  |
| \tYang bukan merupakan pasangan gaya aksi...   | yang, bukan, merupakan, pasangan, gaya, aksi... |

| Soal  | Contoh Hasil Proses <i>Tokenizing</i>            |
|---|--|
| \tPerhatikan pilihan gambar tentang seorang...    | perhatikan, pilihan, gambar, tentang, seorang... |
| \tPada gambar di bawah <br> <br> pasangan...      | pada, gambar, di, bawah, pasangan, gaya, aksi... |
| \tSebuah benda digantungkan pada langit-langit... | sebuah, benda, digantungkan, pada, langit...     |

*Stemming* adalah suatu proses untuk mengubah kata menjadi bentuk dasar. Contoh hasil dari proses *stemming* dapat dilihat di tabel 4.

Tabel 4. Hasil dari Proses *Stemming*

| Soal  | Contoh Hasil Proses <i>Stemming</i>         |
|---|---|
| \tSebuah bola besi digantungkan pada langit...    | bola, besi, gantung, langit, langit         |
| \tYang bukan merupakan pasangan gaya aksi...      | pasang, gaya, aksi, reaksi                  |
| \tPerhatikan pilihan gambar tentang seorang...    | perhati, pilih, gambar, anak, tarik, mobil  |
| \tPada gambar di bawah <br> <br> pasangan...      | gambar, pasang, gaya, aksi, reaksi          |
| \tSebuah benda digantungkan pada langit-langit... | benda, gantung, langit, langit, gambar, gue |

## 2.5 Accuracy

Akurasi atau *accuracy* setiap bagian data adalah persentase label yang diprediksi dengan benar dibagi dengan jumlah total label yang diprediksi dan label sebenarnya dari bagian data tersebut. Presisi keseluruhan adalah nilai presisi rata-rata setiap titik data dibagi dengan jumlah titik data [14].

## 2.6 K-Fold Cross Validation

*K-Fold* adalah metode validasi silang umum yang melipat data sebanyak  $k$  kali dan mengulanginya sebanyak  $k$  kali. Validasi silang *K-fold* membagi data menjadi segmen yang sama. Satu segmen digunakan sebagai data pelatihan dan segmen lainnya digunakan sebagai data pengujian. Bedanya, metode ini mengharuskan seluruh data yang terdapat dalam *dataset* digunakan baik sebagai data latih maupun data uji. Validasi silang membagi semua data menjadi  $k$  kelompok. Jumlah tersebut adalah *record* dibagi dengan jumlah  $k$  yang digunakan [15].

## 3. HASIL DAN PEMBAHASAN

Dalam menentukan parameter terbaik dari metode yang digunakan, penulis melakukan tiga skenario pengujian terhadap model klasifikasi yang dibangun, yaitu sebagai berikut:

### 3.1 Skenario 1

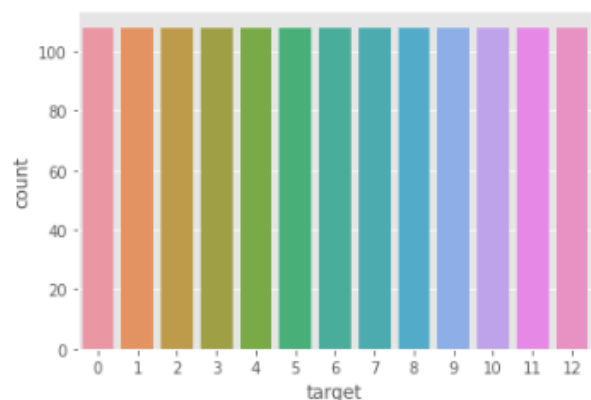
Analisis terhadap algoritma Naïve Bayes, SVM, *Random Forest*, *Decision Tree*, dan *Logistic Regression* dengan *dataset* soal-soal fisika tanpa dilakukan SMOTE atau *data balancing* terhadap *dataset*. Data yang didapatkan dari analisis ini berupa *training time*, *prediction time*, *testing accuracy*, dan *training accuracy*.

Tabel 5. Hasil Pengujian Tanpa SMOTE

|                     | <i>Training Time</i> | <i>Prediction Time</i> | <i>Testing Accuracy</i> | <i>Training Accuracy</i> |
|---------------------|----------------------|------------------------|-------------------------|--------------------------|
| Naïve Bayes         | 0.003                | 0.0006                 | 0.61                    | 0.71                     |
| SVM                 | 0.045                | 0.0005                 | 0.68                    | 0.88                     |
| Random Forest       | 0.302                | 0.0162                 | 0.74                    | 0.95                     |
| Decision Tree       | 0.030                | 0.0008                 | 0.68                    | 0.95                     |
| Logistic Regression | 0.262                | 0.0018                 | 0.74                    | 0.89                     |

Berdasarkan tabel 5, Algoritma Naïve Bayes memiliki *training time* tercepat dengan nilai 0.003 detik. Untuk *prediction time* tercepat diperoleh oleh algoritma SVM dengan nilai 0.0005 detik. Untuk *testing accuracy*, ada dua algoritma yang memiliki hasil yang sama yaitu algoritma *Random Forest* dan *Logistic Regression* yang memiliki nilai sebesar 74%. Untuk *training accuracy*, ada dua algoritma yang memiliki hasil yang sama yaitu algoritma *Random Forest* dan *Decision Tree* yang memiliki nilai sebesar 95%.

### 3.2 Skenario 2



Gambar 3. Proses SMOTE (*Data Balancing*)

Analisis terhadap algoritma Naïve Bayes, SVM, *Random Forest*, *Decision Tree*, dan *Logistic Regression* dengan *dataset* soal-soal fisika dengan dilakukan *smote* atau *data balancing* terhadap *dataset*. Data yang didapatkan dari analisis ini berupa *training time*, *prediction time*, *testing accuracy*, dan *training accuracy* yang diperoleh dari *dataset* yang sudah dilakukan *data balancing* atau SMOTE. Untuk

Hasil Proses SMOTE atau *data balancing* bisa dilihat di gambar 3.

**Tabel 6.** Hasil Pengujian dengan SMOTE (*Data Balancing*)

|                     | <i>Training Time</i> | <i>Prediction Time</i> | <i>Testing Accuracy</i> | <i>Training Accuracy</i> |
|---------------------|----------------------|------------------------|-------------------------|--------------------------|
| Naïve Bayes         | 0.009                | 0.0017                 | 0.80                    | 0.91                     |
| SVM                 | 0.029                | 0.0007                 | 0.88                    | 0.990                    |
| RandomFo rest       | 0.290                | 0.0163                 | 0.85                    | 0.996                    |
| Decision Tree       | 0.127                | 0.0008                 | 0.77                    | 0.997                    |
| Logistic Regression | 0.844                | 0.0009                 | 0.89                    | 0.95                     |

Berdasarkan tabel 6, Algoritma Naïve Bayes memiliki *training time* tercepat dengan nilai 0.009 detik. Untuk *prediction time* tercepat diperoleh oleh algoritma SVM dengan nilai 0.0005 detik. Untuk *testing accuracy*, ada dua algoritma yang memiliki hasil yang sama yaitu algoritma *Random Forest* dan *Logistic Regression* yang memiliki nilai sebesar 74%. Untuk *training accuracy*, ada dua algoritma yang memiliki hasil yang sama yaitu algoritma *Random Forest* dan *Decision Tree* yang memiliki nilai sebesar 95%.

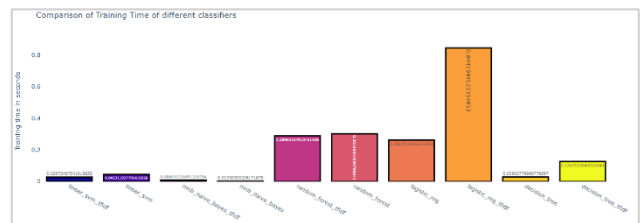
### 3.3 Skenario 3

Analisis terhadap algoritma Naïve Bayes, SVM, *Random Forest*, *Decision Tree*, dan *Logistic Regression* dengan dataset soal-soal fisika dengan dilakukan SMOTE atau data balancing terhadap dataset. Pada skenario ini, terdapat metode tambahan yaitu metode K-Folds Cross Validation dengan *k* sebanyak 10 untuk menguji *dataset* dengan tingkat akurasi yang lebih tinggi.

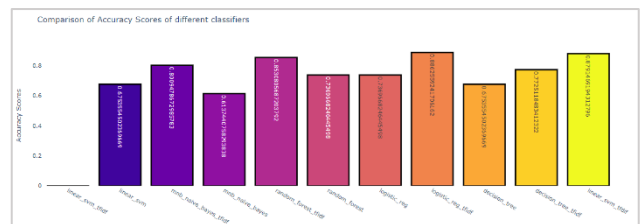
**Tabel 7.** Hasil Pengujian K-Fold Cross Validation dengan SMOTE

|                     | <i>Maximum Accuracy</i> | <i>Minimum Accuracy</i> | <i>Overall Accuracy</i> |
|---------------------|-------------------------|-------------------------|-------------------------|
| Naïve Bayes         | 86.52%                  | 79.43%                  | 83.12%                  |
| SVM                 | 94.33%                  | 88.57%                  | 91.52%                  |
| RandomForest        | 90.71%                  | 85%                     | 88.68%                  |
| Decision Tree       | 82.86%                  | 72.34%                  | 78.78%                  |
| Logistic Regression | -                       | -                       | -                       |

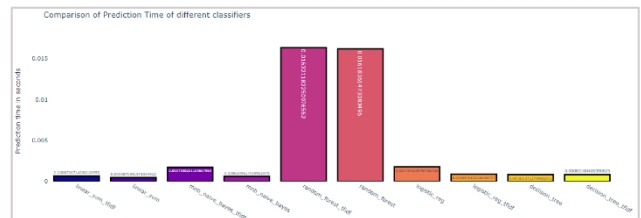
Berdasarkan tabel 7, Algoritma SVM memiliki *maximum accuracy* tertinggi dengan nilai 94.33% dibandingkan dengan Naïve Bayes yang memiliki nilai 86.52%. Untuk *minimum accuracy* tertinggi juga diperoleh oleh algoritma SVM dengan nilai 88.57% dibandingkan dengan Naïve Bayes yang memiliki nilai 79.43%. Untuk *overall accuracy*, algoritma SVM memiliki hasil tertinggi dengan nilai 91.52% dibandingkan dengan Naïve Bayes dengan nilai 83.12%. Berdasarkan hasil dari pengujian K-Fold Cross Validation, Naïve Bayes belum dapat memiliki nilai akurasi yang lebih baik dibandingkan oleh nilai akurasi yang dimiliki oleh algoritma SVM. Tetapi untuk *training time* algoritma SVM dan Naïve Bayes dengan *dataset* yang sudah dilakukan SMOTE, Naïve Bayes masih unggul dengan nilai 0.009 detik.



**Gambar 4.** Komparasi Training Time Setiap Algoritma



**Gambar 5.** Komparasi Accuracy Setiap Algoritma



**Gambar 6.** Komparasi Prediction Time Setiap Algoritma

Berdasarkan gambar 4, dapat disimpulkan bahwa algoritma *Logistic Regression* dengan data yang sudah melakukan data *balance* memiliki *training time* terlama dengan nilai 0.844 detik dibandingkan dengan algoritma lainnya. Berdasarkan gambar 5, dapat disimpulkan bahwa *Logistic Regression* dengan data yang sudah dilakukan SMOTE memiliki hasil akurasi tertinggi dengan nilai 89% dibanding dengan algoritma lainnya. Berdasarkan gambar 6, dapat disimpulkan bahwa SVM menggunakan data yang tidak dilakukan SMOTE memiliki *prediction time* tercepat dengan waktu sebesar 0.000487 detik.

Berdasarkan hasil di atas, dapat disimpulkan bahwa algoritma Naïve Bayes memiliki waktu *training data* tercepat dengan hasil akurasi yang baik dibandingkan dengan algoritma-algoritma lainnya. Maka dari itu, algoritma Naïve Bayes dapat digunakan untuk mengklasifikasikan soal fisika dan hasil dari klasifikasi tersebut dapat digunakan oleh siswa SMA untuk bisa memahami soal-soal tersebut dengan lebih mudah dan cepat.

#### 4. KESIMPULAN

Dari penelitian ini, didapatkan bahwa Naïve Bayes adalah algoritma yang memiliki tingkat kecepatan *training* yang tinggi dibandingkan dengan algoritma lainnya. Maka dari itu, algoritma Naïve Bayes dapat menjadi solusi untuk mengklasifikasikan soal fisika dengan cepat dan memberikan kemudahan untuk siswa SMA untuk dapat memahami soal fisika dengan lebih baik. Tetapi kekurangan dari algoritma ini adalah nilai akurasi yang dihasilkan bukan merupakan nilai tertinggi apabila dibandingkan dengan algoritma SVM. Untuk penelitian yang akan mendatang, dapat disarankan untuk melakukan kombinasi algoritma dan evaluasi yang lebih komprehensif untuk dapat meningkatkan nilai akurasi.

#### Ucapan Terima Kasih

Saya mengucapkan terima kasih kepada seluruh pihak yang mendukung pembuatan jurnal ini.

#### DAFTAR PUSTAKA

- [1] M. M. N. Karsana, K. M. L, and W. Astuti, "Single-Label and Multi-Label Text Classification using ANN and Comparison with Naïve Bayes and SVM," vol. 7, no. April, pp. 857–863, 2023, doi: 10.30865/mib.v7i2.6024.
- [2] T. Durand and G. Mori, "Learning a Deep ConvNet for Multi-Label Classification With Partial Labels," pp. 647–657.
- [3] P. Mata, P. Matematika, D. Z. Azhari, I. S. Damanik, and D. Suhendro, "Penerapan Algoritma C4 . 5 Untuk Klasifikasi Tingkat Pemahaman Siswa," vol. 1, no. 1, pp. 11–20, 2022.
- [4] A. Fathiarahma, A. Voutama, T. Ridwan, and N. Heryana, "Analisis Text Mining Klasifikasi Kegiatan Keluarga menggunakan Orange dengan Metode Naive Bayes," *J. Teknol. Terpadu*, vol. 9, no. 1, pp. 35–41, 2023, doi: 10.54914/jtt.v9i1.606.
- [5] H. Hendriyana, I. M. Karo Karo, and S. Dewi, "Analisis perbandingan Algoritma Support Vector Machine, Naive Bayes dan Regresi Logistik untuk Memprediksi Donor Darah," *J. Teknol. Terpadu*, vol. 8, no. 2, pp. 121–126, 2022, doi: 10.54914/jtt.v8i2.581.
- [6] I. M. Karo Karo and H. Hendriyana, "Klasifikasi Penderita Diabetes menggunakan Algoritma Machine Learning dan Z-Score," *J. Teknol. Terpadu*, vol. 8, no. 2, pp. 94–99, 2022, doi: 10.54914/jtt.v8i2.564.
- [7] A. Wiraguna, S. Al Faraby, and Adiwijaya, "Klasifikasi Topik Multi Label pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan Random Forest," *e-Proceeding Eng.*, vol. 6, no. 1, pp. 2144–2153, 2019, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/8507%0Ahttps://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/download/8507/8377>
- [8] M. R. Adrian, M. P. Putra, M. H. Rafialdy, and N. A. Rakhmawati, "Perbandingan Metode Klasifikasi Random Forest dan SVM Pada Analisis Sentimen PSBB," *J. Inform. Upgris*, vol. 7, no. 1, pp. 36–40, 2021.
- [9] I. Made Riartha Prawira and M. Syahrul Mubarak, "Klasifikasi Multi-Label Pada Topik Berita Berbahasa Indonesia Menggunakan Multinomial Naïve Bayes," *e-Proceeding Eng.*, vol. 5, no. 3, pp. 7774–7781, 2018.
- [10] H. A. N. S. Habibi, A. Nugroho, and R. Firliana, "Perbandingan Algoritma Naïve Bayes Classifier Dan K-Nearest Neighbors Untuk Analisis Sentimen Covid-19 Di Twitter," *J. Ilm. Inform.*, vol. 11, no. 01, pp. 54–62, 2023, doi: 10.33884/jif.v11i01.7069.
- [11] N. Nurainun, E. Haerani, F. Syafria, and L. Oktavia, "Penerapan Algoritma Naïve Bayes Classifier Dalam Klasifikasi Status Gizi Balita dengan Pengujian K-Fold Cross Validation," *J. Comput. Syst. Informatics*, vol. 4, no. 3, pp. 578–586, 2023, doi: 10.47065/josyc.v4i3.3414.
- [12] D. C. Hidayati, S. Al Faraby, and A. Adiwijaya, "Klasifikasi Topik Multi Label pada Hadis Shahih Bukhari Menggunakan K-Nearest Neighbor dan Latent Semantic Analysis," *JURIKOM (Jurnal Ris. Komputer)*, vol. 7, no. 1, p. 140, 2020, doi: 10.30865/jurikom.v7i1.2013.
- [13] A. Hanafi, A. Adiwijaya, and W. Astuti, "Klasifikasi Multi Label pada Hadis Bukhari Terjemahan Bahasa Indonesia Menggunakan Mutual Information dan k-Nearest Neighbor," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, pp. 357–364, 2020, doi: 10.32736/sisfokom.v9i3.980.
- [14] I. P. Y. Indrawan, I. G. Indrawan, and I. M. Candiasa, "Klasifikasi Multilabel Pada Abstrak Tugas Akhir Menggunakan Vector Space Model dan K-Nearest Neighbors Program Studi Magister Ilmu Komputer, Program Pascasarjana," vol. 2, no. 2, pp. 91–97, 2019.
- [15] N. Isnaini, Adiwijaya, M. S. Mubarak, and M. Y. A. Bakar, "A multi-label classification on topics of Indonesian news using K-Nearest Neighbor," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019, doi: 10.1088/1742-6596/1192/1/012027.