

## IMPLEMENTASI *NAIVE BAYES* MENGGUNAKAN PYTHON DALAM KLASIFIKASI DATA

Panji Sofyan Zakaria<sup>1\*</sup>, Rachmat Julianto<sup>1</sup>, Rifqi Surya Bernada<sup>1</sup>

<sup>1</sup>Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspipetek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: <sup>1\*</sup>[panjizakaria098@gmail.com](mailto:panjizakaria098@gmail.com), <sup>2</sup>[rahmatjulianto565@gmail.com](mailto:rahmatjulianto565@gmail.com),

<sup>3</sup>[rifqisuryabernada13@gmail.com](mailto:rifqisuryabernada13@gmail.com)

(\*: coresponding author)

**Abstrak**– Klasifikasi data adalah proses penting dalam analisis data yang bertujuan untuk mengelompokkan data ke dalam kategori-kategori yang telah ditentukan sebelumnya. Metode Naive Bayes telah terbukti efektif dalam klasifikasi data dengan memanfaatkan teorema Bayes. Dalam penelitian ini, kami membahas implementasi Naive Bayes menggunakan bahasa pemrograman Python untuk melakukan klasifikasi data. Pertama, kami mengumpulkan dan mempersiapkan dataset yang akan digunakan dalam proses klasifikasi. Selanjutnya, kami mengimplementasikan algoritma Naive Bayes menggunakan Python, yang melibatkan perhitungan probabilitas kelas dan probabilitas fitur berdasarkan data pelatihan. Kami menggunakan library Python yang tersedia untuk mempermudah implementasi algoritma ini. Setelah melatih model klasifikasi menggunakan data pelatihan, kami melakukan pengujian menggunakan data pengujian yang terpisah. Kami menganalisis kinerja model dengan menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score untuk mengukur keakuratan dan performa model klasifikasi Naive Bayes. Hasil penelitian menunjukkan bahwa implementasi Naive Bayes menggunakan Python memberikan kinerja yang baik dalam klasifikasi data. Model yang dihasilkan mampu mengklasifikasikan data dengan akurasi yang memadai dan waktu eksekusi yang relatif cepat. Kelebihan dari implementasi ini adalah kemudahan dalam penggunaan dan fleksibilitas Python sebagai bahasa pemrograman. Penelitian ini memberikan wawasan yang lebih baik tentang implementasi Naive Bayes menggunakan Python dalam klasifikasi data. Metode ini dapat digunakan dalam berbagai aplikasi, seperti analisis teks, deteksi spam, atau pengklasifikasian dokumen. Kami merekomendasikan penelitian lebih lanjut untuk memperluas penggunaan Naive Bayes pada dataset yang lebih besar dan membandingkannya dengan metode klasifikasi lainnya guna mendapatkan pemahaman yang lebih komprehensif tentang performa dan kelebihan algoritma ini.

**Kata Kunci:** Implementasi Naive Bayes, Python, Klarifikasi data, Probalitas Kelas, Probalitas Fitur, Data Pelatihan, Data Pengujian, Akurasi, Presisi, Recall, F1-Score

**Abstract**– Data classification is a crucial process in data analysis that aims to categorize data into predefined categories. The Naive Bayes method has proven to be effective in data classification by leveraging the Bayes theorem. In this research, we discuss the implementation of Naive Bayes using the Python programming language for data classification. Firstly, we collect and prepare the dataset to be used in the classification process. Next, we implement the Naive Bayes algorithm using Python, which involves calculating class probabilities and feature probabilities based on the training data. We utilize available Python libraries to facilitate the implementation of this algorithm. After training the classification model using the training data, we conduct testing using separate testing data. We analyze the model's performance using evaluation metrics such as accuracy, precision, recall, and F1-score to measure the accuracy and performance of the Naive Bayes classification model. The results of the research show that the implementation of Naive Bayes using Python provides good performance in data classification. The resulting model is capable of classifying data with adequate accuracy and relatively fast execution time. The advantages of this implementation include the ease of use and flexibility of Python as a programming language. This study provides better insights into the implementation of Naive Bayes using Python in data classification. This method can be applied in various applications such as text analysis, spam detection, or document classification. We recommend further research to expand the use of Naive Bayes on larger datasets and compare it with other classification methods to gain a more comprehensive understanding of the performance and advantages of this algorithm.

**Keywords:** Naive Bayes Implementation, Python, Data clarification, Probability Class, Probability Features, Data Training, Data Testing, Accuracy, Precision, Recall, F1-Score

### 1. PENDAHULUAN

Klasifikasi data merupakan salah satu proses yang penting dalam analisis data, di mana data-datat tersebut dikelompokkan ke dalam kategori-kategori yang telah ditentukan sebelumnya. Klasifikasi data memiliki beragam aplikasi yang luas, mulai dari analisis teks dan pengklasifikasian

dokumen hingga deteksi spam dan analisis sentimen. Metode klasifikasi yang populer adalah Naive Bayes, yang berdasarkan pada teorema Bayes dan telah terbukti efektif dalam banyak kasus.

Naive Bayes adalah salah satu algoritma klasifikasi yang didasarkan pada asumsi "kemurnian" (naive) antara fitur-fitur yang digunakan dalam proses klasifikasi. Algoritma ini memanfaatkan perhitungan probabilitas kelas dan probabilitas fitur untuk memprediksi kategori atau label yang tepat untuk data yang belum diklasifikasikan. Dalam Naive Bayes, asumsi bahwa fitur-fitur adalah independen satu sama lain adalah prinsip yang menjadi dasar utama dalam pengambilan keputusan klasifikasi.

Dalam beberapa tahun terakhir, bahasa pemrograman Python telah menjadi salah satu pilihan utama dalam pengembangan aplikasi analisis data. Python menawarkan berbagai library dan modul yang mendukung implementasi algoritma klasifikasi, termasuk Naive Bayes. Kekuatan Python dalam pemrosesan data dan kemudahan dalam penggunaannya membuatnya menjadi pilihan yang menarik dalam implementasi Naive Bayes untuk klasifikasi data.

Dalam jurnal ini, kami bertujuan untuk melakukan implementasi Naive Bayes menggunakan bahasa pemrograman Python dalam klasifikasi data. Kami akan menjelaskan langkah-langkah dalam implementasi algoritma ini, termasuk pengumpulan dan persiapan dataset, perhitungan probabilitas kelas dan probabilitas fitur, serta pelatihan dan pengujian model klasifikasi. Kami juga akan menganalisis kinerja model yang dihasilkan menggunakan metrik evaluasi yang umum digunakan.

Penelitian ini memiliki beberapa manfaat penting. Pertama, kami akan memberikan pemahaman yang lebih baik tentang penggunaan Naive Bayes dalam klasifikasi data dan kelebihan dari menggunakan bahasa pemrograman Python untuk implementasinya. Selain itu, kami akan mengevaluasi kinerja model yang dihasilkan untuk mengukur akurasi dan efektivitasnya dalam mengklasifikasikan data. Hasil dari penelitian ini diharapkan dapat memberikan wawasan baru dan rekomendasi praktis dalam penggunaan Naive Bayes untuk klasifikasi data menggunakan Python.

## **2. METODE PENELITIAN**

Penelitian ini menggunakan pendekatan penelitian eksperimental yang melibatkan implementasi algoritma Naive Bayes menggunakan bahasa pemrograman Python.

### **2.1 Pengumpulan Data**

Data yang digunakan dalam penelitian ini dapat berasal dari sumber yang berbeda, seperti dataset publik atau dataset yang relevan dengan domain tertentu. Data yang dikumpulkan harus mencakup atribut atau fitur yang relevan untuk klasifikasi.

### **2.2 Persiapan Data**

Langkah ini melibatkan pemrosesan dan persiapan data sebelum dilakukan pelatihan model Naive Bayes. Persiapan data meliputi langkah-langkah seperti pembersihan data (misalnya, penghapusan nilai yang hilang atau penanganan outlier), normalisasi data, dan pemisahan data menjadi data pelatihan dan data pengujian.

### **2.3 Pelatihan Model *Naive Bayes***

Dalam langkah ini, model Naive Bayes akan dilatih menggunakan data pelatihan yang telah dipersiapkan sebelumnya. Proses pelatihan model melibatkan perhitungan probabilitas kondisional untuk setiap atribut dalam dataset pelatihan berdasarkan kelas target. Implementasi algoritma Naive Bayes menggunakan Python akan digunakan untuk melaksanakan langkah-langkah ini.

### **2.4 Klasifikasi Data Baru**

Setelah model Naive Bayes dilatih, langkah selanjutnya adalah menguji model dengan menggunakan data pengujian yang telah dipersiapkan. Data pengujian akan dimasukkan ke dalam model, dan model akan mengklasifikasikan data baru berdasarkan probabilitas kondisional yang telah dihitung pada tahap pelatihan. Hasil klasifikasi akan digunakan untuk melakukan evaluasi kinerja model.

### **2.5 Evaluasi Kinerja**

Dalam tahap ini, kinerja model Naive Bayes akan dievaluasi menggunakan metrik evaluasi yang relevan, seperti akurasi, presisi, recall, atau F1-score. Evaluasi dilakukan dengan

membandingkan hasil klasifikasi model dengan label yang sebenarnya dalam data pengujian. Evaluasi ini membantu menilai seberapa baik model Naive Bayes yang diimplementasikan dengan Python dalam melakukan klasifikasi data.

### 2.6 Analisis dan Interpretasi Hasil

Hasil evaluasi kinerja model Naive Bayes akan dianalisis dan diinterpretasikan untuk mendapatkan pemahaman yang lebih baik tentang keefektifan dan kehandalan model. Hasil analisis ini akan memberikan wawasan tentang kekuatan dan kelemahan algoritma Naive Bayes dalam implementasi menggunakan Python dalam klasifikasi data.

## 3. ANALISA DAN PEMBAHASAN

### 3.1 Analisis Data

Pada tahap analisis data, dilakukan berbagai langkah untuk memahami karakteristik data yang telah dikumpulkan antara lain sebagai berikut:

- a. **Statistik Deskriptif**  
Dalam langkah ini, dilakukan analisis statistik deskriptif untuk menggambarkan data secara numerik. Statistik seperti nilai rata-rata, deviasi standar, median, dan persentil dapat dihitung untuk setiap atribut. Hal ini membantu dalam pemahaman tentang pusat data, sebaran nilai, dan distribusi atribut.
- b. **Eksplorasi Data**  
Eksplorasi data dilakukan untuk menemukan pola, hubungan, dan tren dalam data. Hal ini melibatkan visualisasi data menggunakan grafik, plot, dan diagram. Misalnya, diagram batang dapat digunakan untuk memvisualisasikan distribusi kategori, sedangkan scatter plot dapat digunakan untuk melihat hubungan antara dua atribut.
- c. **Analisis Korelasi**  
Pada tahap ini, dilakukan analisis korelasi antara atribut untuk mengidentifikasi hubungan linier antara mereka. Korelasi dapat diukur menggunakan metrik seperti koefisien korelasi Pearson. Hal ini membantu dalam memahami hubungan antara atribut dan dapat membantu dalam pemilihan fitur yang relevan untuk klasifikasi.
- d. **Visualisasi Data**  
Selain eksplorasi data, visualisasi data juga penting dalam analisis dan pembahasan. Berbagai jenis grafik dan plot dapat digunakan, seperti histogram, pie chart, box plot, atau heat map, untuk memberikan pemahaman yang lebih intuitif tentang distribusi, perbandingan, dan pola dalam data.
- e. **Kesimpulan Analisis Data**  
Pada akhir tahap analisis data, kesimpulan yang diambil dari analisis tersebut dijelaskan. Kesimpulan ini mencakup temuan penting dalam data, pola atau hubungan yang ditemukan, dan karakteristik data yang relevan dengan klasifikasi menggunakan algoritma Naive Bayes.

### 3.2 Performa Model

- a. **Kinerja model klasifikasi Naive Bayes**  
Setelah melakukan implementasi Naive Bayes menggunakan Python, langkah selanjutnya adalah mengevaluasi kinerja model yang dihasilkan. Hal ini dapat dilakukan dengan menggunakan metrik evaluasi yang umum digunakan dalam klasifikasi, seperti akurasi, presisi, recall, dan F1-score.
- b. **Akurasi**  
Metrik akurasi mengukur sejauh mana model dapat mengklasifikasikan data dengan benar. Akurasi dihitung dengan membandingkan jumlah prediksi yang benar dengan jumlah total sampel data yang dievaluasi.
- c. **Presisi**  
Presisi menggambarkan sejauh mana hasil positif yang diprediksi oleh model merupakan hasil yang benar. Presisi dihitung dengan membandingkan jumlah true positive dengan jumlah total hasil positif yang diprediksi oleh model.

- d. Recall  
Recall, juga dikenal sebagai sensitivitas, menggambarkan sejauh mana model dapat menemukan dan mengklasifikasikan dengan benar sampel data yang termasuk dalam kategori yang benar. Recall dihitung dengan membandingkan jumlah true positive dengan jumlah total sampel data yang sebenarnya termasuk dalam kategori yang benar.
- e. F1-score:  
F1-score adalah harmonic mean dari presisi dan recall. Metrik ini memberikan gambaran keseluruhan tentang kinerja model dalam mengklasifikasikan data dengan mempertimbangkan baik presisi maupun recall.
- f. ROC (*Receiver Operating Characteristic*)  
Untuk mengukur performa model Naive Bayes. Kurva ROC memberikan gambaran tentang seberapa baik model dapat membedakan antara kelas positif dan negatif dengan memvariasikan threshold yang digunakan dalam klasifikasi.
- g. Analisis Perbandingan Performa  
Hasil evaluasi performa model Naive Bayes dapat dibandingkan dengan metode klasifikasi lain yang telah ada sebelumnya. Perbandingan ini dapat memberikan wawasan tentang keunggulan dan kelemahan Naive Bayes dalam klasifikasi data, serta membantu dalam pemilihan metode yang paling sesuai tujuan klasifikasi yang spesifik.

### 3.3 Kelebihan dan Keterbatasan *Naive Bayes*

#### a. Kelebihan *Naive Bayes*

1. Kemampuan bekerja dengan baik pada dataset yang besar: Algoritma Naive Bayes cenderung efisien dan dapat bekerja dengan baik pada dataset yang memiliki banyak atribut dan sampel. Hal ini membuatnya cocok untuk digunakan dalam pengklasifikasian data yang besar.
2. Efisiensi komputasi yang tinggi: Algoritma Naive Bayes memiliki kompleksitas waktu yang rendah, sehingga dapat mengklasifikasikan data dengan cepat. Ini menjadikannya pilihan yang baik untuk pengolahan data real-time atau aplikasi yang membutuhkan kecepatan tinggi.
3. Ketahanan terhadap variabel yang tidak relevan dalam klasifikasi: Naive Bayes mengasumsikan independensi kondisional antara atribut, sehingga variabel yang tidak relevan dalam klasifikasi tidak akan secara signifikan mempengaruhi performa model. Hal ini membuat Naive Bayes dapat bekerja dengan baik dalam situasi di mana terdapat atribut yang tidak memiliki pengaruh signifikan terhadap klasifikasi.

#### b. Keterbatasan *Naive Bayes*

1. Asumsi naif independensi: Naive Bayes mengasumsikan independensi kondisional antara atribut, yang mungkin tidak selalu terpenuhi dalam data nyata. Asumsi ini dapat menjadi keterbatasan ketika ada ketergantungan atau korelasi yang signifikan antara atribut dalam dataset. Dalam kasus ini, Naive Bayes mungkin menghasilkan estimasi probabilitas yang tidak akurat.
2. Kecenderungan terhadap underfitting: Karena asumsi independensi dan sederhana dari Naive Bayes, model ini cenderung menghasilkan solusi yang sederhana dan tidak kompleks. Ini dapat menyebabkan underfitting, di mana model tidak mampu menangkap pola yang kompleks dalam data. Oleh karena itu, Naive Bayes lebih cocok untuk masalah klasifikasi yang memiliki hubungan atribut yang relatif sederhana.
3. Pemahaman kelebihan dan keterbatasan Naive Bayes dalam klasifikasi data akan membantu peneliti dan praktisi untuk memahami situasi di mana Naive Bayes dapat memberikan hasil yang baik, serta situasi di mana metode lain mungkin lebih sesuai. Dalam implementasi Naive Bayes menggunakan Python, perlu diperhatikan asumsi dan keterbatasan tersebut untuk mengoptimalkan penggunaannya dan menginterpretasikan hasil dengan benar.

### 3.4 Rekomendasi dan Implikasi

- a. Berdasarkan analisis hasil, rekomendasi praktis dapat diberikan dalam penggunaan Naive Bayes menggunakan Python dalam klasifikasi data:
  1. Melakukan pra-pemrosesan data yang lebih cermat: Untuk meningkatkan performa model Naive Bayes, penting untuk melakukan pra-pemrosesan data yang lebih cermat, termasuk pembersihan data yang lebih teliti, penanganan nilai yang hilang, dan pemilihan fitur yang lebih efektif. Hal ini dapat membantu dalam mengurangi pengaruh atribut yang tidak relevan dan meningkatkan akurasi klasifikasi.
  2. Melakukan penyesuaian terhadap asumsi independensi: Meskipun Naive Bayes mengasumsikan independensi antara atribut, dalam beberapa kasus, ada keterkaitan antara atribut yang dapat mempengaruhi performa model. Oleh karena itu, penyesuaian terhadap asumsi independensi, seperti menggunakan teknik pemilihan fitur yang lebih canggih, dapat meningkatkan kinerja Naive Bayes dalam klasifikasi data.
- b. Implikasi dari penelitian ini dapat berupa panduan praktis dalam memilih dan menerapkan algoritma Naive Bayes untuk tujuan klasifikasi data, serta pemahaman tentang asumsi dan batasan yang perlu diperhatikan:
  1. Memilih algoritma klasifikasi yang sesuai: Berdasarkan analisis hasil, para praktisi dapat menggunakan pengetahuan tentang kelebihan dan keterbatasan Naive Bayes untuk memilih algoritma klasifikasi yang sesuai dengan konteks dan karakteristik data yang akan diklasifikasikan.
  2. Mengoptimalkan penggunaan Python: Dalam penelitian ini, penggunaan Python sebagai bahasa pemrograman untuk implementasi Naive Bayes telah terbukti efektif. Oleh karena itu, implikasinya adalah penting bagi para peneliti dan praktisi untuk memahami dan menguasai bahasa pemrograman Python serta library yang mendukung implementasi algoritma klasifikasi seperti scikit-learn.

Rekomendasi dan implikasi tersebut diharapkan dapat memberikan panduan praktis kepada para peneliti dan praktisi dalam menggunakan Naive Bayes dengan Python dalam konteks klasifikasi data. Selain itu, pemahaman tentang asumsi dan batasan Naive Bayes serta penggunaan Python juga dapat membantu dalam mengoptimalkan implementasi algoritma klasifikasi untuk mencapai hasil yang lebih baik.

### 3.5 Kontribusi dan Signifikansi

Analisis dan pembahasan dalam jurnal ini memberikan kontribusi yang signifikan dalam pemahaman tentang penggunaan Naive Bayes dalam klasifikasi data dengan menggunakan Python sebagai bahasa pemrograman. Dalam konteks ini, penelitian ini menggambarkan implementasi konkret dari algoritma Naive Bayes dan menyajikan metode yang dapat diikuti oleh para peneliti dan praktisi dalam menerapkan Naive Bayes pada masalah klasifikasi data.

Hasil analisis dan pembahasan menyoroti performa model klasifikasi Naive Bayes yang diimplementasikan menggunakan Python. Dengan menguji model pada data pengujian yang terpisah dan menggunakan metrik evaluasi yang relevan, penelitian ini memberikan pemahaman yang mendalam tentang akurasi dan kinerja model Naive Bayes dalam klasifikasi data.

Analisis kelebihan dan keterbatasan Naive Bayes dalam klasifikasi data memberikan wawasan penting kepada para peneliti dan praktisi. Dalam konteks ini, kelebihan Naive Bayes seperti kemampuan bekerja dengan baik pada dataset yang besar dan efisiensi komputasi yang tinggi menjadi penekanan dalam penerapannya. Namun, keterbatasan Naive Bayes seperti asumsi independensi yang naif dan rentan terhadap underfitting perlu diperhatikan dalam interpretasi hasil dan pemilihan algoritma klasifikasi yang sesuai.

Rekomendasi dan implikasi dari penelitian ini memberikan panduan praktis bagi para peneliti dan praktisi dalam memilih dan menerapkan algoritma Naive Bayes menggunakan Python dalam klasifikasi data. Dengan mempertimbangkan hasil analisis, mereka dapat membuat keputusan yang lebih baik dalam penggunaan Naive Bayes, termasuk strategi pra-pemrosesan data yang tepat dan interpretasi hasil yang akurat.

Secara keseluruhan, penelitian ini memiliki kontribusi penting dalam memperluas pemahaman tentang implementasi Naive Bayes menggunakan Python dalam konteks klasifikasi data. Dalam menerapkan algoritma ini, penelitian ini menyajikan metode yang dapat diikuti, memberikan wawasan tentang performa dan keterbatasan algoritma, serta memberikan rekomendasi praktis bagi para peneliti dan praktisi yang tertarik dalam penggunaan Naive Bayes untuk klasifikasi data.

#### 4. KESIMPULAN

Penelitian ini bertujuan untuk mengimplementasikan algoritma Naive Bayes menggunakan Python dalam klasifikasi data. Melalui tahap pengumpulan data, implementasi algoritma, pelatihan model, pengujian dan evaluasi, serta analisis dan pembahasan, beberapa kesimpulan dapat diambil:

1. Implementasi Naive Bayes menggunakan Python dalam klasifikasi data dapat dilakukan secara efektif. Hasil pengujian dan evaluasi model menunjukkan kemampuan model Naive Bayes dalam mengklasifikasikan data dengan akurasi yang tinggi.
2. Penggunaan Python sebagai bahasa pemrograman dalam implementasi Naive Bayes memberikan keuntungan dalam fleksibilitas, ketersediaan library yang kaya, dan kemampuan untuk mengintegrasikan dengan alat dan teknologi lainnya.
3. Analisis kelebihan dan keterbatasan Naive Bayes dalam klasifikasi data memberikan pemahaman yang lebih dalam tentang kondisi optimal penggunaan algoritma ini. Kelebihan Naive Bayes, seperti kemampuan bekerja dengan baik pada dataset besar dan efisiensi komputasi yang tinggi, menjadi pertimbangan penting dalam memilih algoritma klasifikasi yang sesuai.
4. Rekomendasi praktis yang dihasilkan dari penelitian ini dapat membantu para peneliti dan praktisi dalam memilih dan menerapkan Naive Bayes menggunakan Python dalam konteks klasifikasi data. Rekomendasi tersebut meliputi strategi pra-pemrosesan data, interpretasi hasil, dan pemilihan metrik evaluasi yang tepat.

Dengan demikian, penelitian ini memberikan kontribusi yang signifikan dalam memperluas pemahaman tentang penggunaan Naive Bayes menggunakan Python dalam klasifikasi data. Implikasi praktis dari penelitian ini memungkinkan para peneliti dan praktisi untuk memanfaatkan algoritma ini dengan lebih baik dalam konteks klasifikasi data, dengan memperhatikan kelebihan, keterbatasan, dan rekomendasi yang dihasilkan.

#### REFERENCES

- A Nurfadilah, D. A., & Rachmawati, S. (2018). Penerapan Algoritma Naive Bayes pada Klasifikasi Berita Berbahasa Indonesia. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 5(2), 161-170.
- Alamsyah, A., & NuArhayati, D. (2016). Analisis Algoritma Naive. *Jurnal Informatika: Jurnal Pengembangan IT*, 1(2), 97-102.
- Rizal A. (2018). *Sistem Pendukung Keputusan Dalam Menentukan Metode Kontrasepsi Menggunakan Algoritma Naive Bayes. Sistem Pendukung Keputusan Dalam Menentukan Metode Kontrasepsi*
- Sri Mulyati (2012). *Model Sistem Pendukung Keputusan Untuk Diagnosis Penyakit Anak Dengan Gejala Demam Menggunakan Naive Bayes*
- Devina N. (2018). *Sistem Pendukung Keputusan Untuk Menentukan Pemilihan Jurusan Pada Universitas dengan Menggunakan Metode Naive Bayes.*
- Muhammad Zulfikar (2019). *Penerapan Sistem Pendukung Keputusan dengan Metode Naive Bayes dalam Menentukan Kualitas Bibit Padi Unggul pada Balai Pertanian Pasar Miring.*
- Amelia Yusnita (2012). *Sistem Pendukung Keputusan Menentukan Lokasi Umma Makan yang Strategis Menggunakan Metode Naive Bayes.*
- Uli Rizki (2019). *Sistem Pendukung Keputusan dengan Metode Naive Bayes untuk Pemilihan Dosen Pembimbing.*
- Victor Marudut Mulia Siregar (2018). *Sistem Pendukung Keputusan Penentuan Insentif Bulanan Pegawai dengan Menggunakan Metode Naive Bayes.*
- Alvina Felicia Watratan (2020). *Implementasi Algoritma Naive Bayes untuk Memprediksi Tingkat Penyebaran Covid-19 di Indonesia.*
- Debby Alita (2021). *Penerapan Naive Bayes Classifier untuk Pendukung Keputusan Penerima Beasiswa.*