Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Analysis of Student Satisfaction Sentiment Towards Lecturer Performance using the Naive Bayes Classifier Method (Case Study: STMIK KAPUTAMA)

Azizhil Hakim[1]*, Relita Buaton[2], Muammar Khadapi[3]

[1,2,3]*STMIK Kaputama*
azizhilhakim@gmail.com [1*], bbcbuaton@gmail.com [2], khdafi5@gmail.com [3]

Abstract

Student satisfaction with lecturer performance is an essential indicator in assessing the quality and competitiveness of higher education institutions. This study aims to analyze student sentiment regarding lecturer performance using the Naïve Bayes Classifier method. The research data were collected from student satisfaction surveys conducted during the 2022–2025 academic years, consisting of open-ended comments about lecturer performance. The research follows the CRISP-DM methodology, including text preprocessing (case folding, cleaning, tokenizing, stopword removal, normalization, and stemming), word weighting using the Term Frequency-Inverse Document Frequency (TF-IDF) method, and sentiment classification into positive, negative, and neutral categories using the Naïve Bayes Classifier algorithm. The implementation was carried out using the Python programming language through Google Colab. The results show that the Naïve Bayes Classifier model achieved an accuracy of 76.9% in classifying student opinions, providing a reliable representation of students' perceptions. These findings are expected to serve as a basis for evaluation and strategic decision-making to improve teaching quality and lecturer performance in higher education institutions.

Keywords: *Sentiment Analysis, Student Satisfaction, Lecturer Performance, Naïve Bayes Classifier, TF-IDF*

1. Introduction

Higher education institutions are institutions of higher learning that focus on the interests of the academic community, including students, teaching staff, and employees. In accordance with Law Number 20 of 2023 on the National Educational System, higher education institutions are obligated to conduct education, research, and community service. Therefore, higher education institutions are required to improve their quality, institutional structure, and services in order to compete amid the rapid growth of higher education institutions [1]

Student satisfaction with lecturer performance is an important factor in determining the quality of education and competitiveness of a university. Universities that are able to provide the best performance in teaching will be more attractive to students and have high competitiveness. STMIK Kaputama, as one of the higher education institutions, is committed to continuously improving the quality of learning and teaching through student satisfaction surveys. Student satisfaction surveys regarding faculty performance provide important insights into students' perceptions of the institution. This data can be used to evaluate and develop strategies for improving teaching processes [2]. One method that can be used to analyse this research is the Naive Bayes Classifier algorithm.

Given these issues, the Naive Bayes Classifier method can be used as an appropriate solution for analysing student satisfaction sentiment towards lecturer performance. This method is capable of grouping data based on various factors such as teaching, lecturer-student interaction, lecturer attitudes and behaviour towards students, and student comprehension of the material presented by lecturers, which will be summarised into one open-ended question related to these factors. With this sentiment analysis, students can easily express their opinions, suggestions, and criticisms, enabling the university to respond with more targeted actions. As a result, this analysis can generate more accurate data to assist higher education institutions in evaluating faculty performance.

This study focuses on measuring lecturer performance based on student opinions through questionnaires as the data source to be used. The results of this study will be classified using the naive Bayes classifier method. The advantage of using the Naïve Bayes classifier is that this method only requires a small amount of training data to determine the parameter estimates needed in the classification process [3]. This study utilises the Python programme, which has various supporting libraries in Text Mining, such as NLP (Natural Language Processing), scikit-learn, and pandas. By using Python, the process of data extraction, preprocessing, and sentiment classification can be carried out more efficiently. The selection of the Naïve Bayes method in text mining is also based on its reliability in handling large and complex text data. Previous studies have shown that this method is capable of providing optimal analysis results in sentiment classification, especially when compared to other methods such as Decision Tree or KNN (K-Nearest Neighbours). The application of text mining in this study aims to gain deeper insights into students' perceptions of lecturer performance at STMIK Kaputama. The results of this study are expected to serve as evaluation material for higher education institutions to improve teaching quality and as a reference for similar research in the future.

The implementation of sentiment analysis also involves preprocessing (tokenising, stemming, and stopword removal) to improve the accuracy of the model. After the data is processed, the Naive Bayes algorithm can be used to classify sentiment based on patterns in the dataset. By applying this, universities need to follow up on the results of this analysis as an evaluation to improve the quality of teaching, faculty training, and learning strategies to be more effective according to student needs. Additionally, student participation in providing useful feedback to faculty is a key factor in improving the quality of education

.

2. Method

Problem-solving methods are one of the methods that will be used in research to solve a problem. The problem-solving method that will be used in this research is the Naïve Bayes Classifier algorithm in the modelling process of sentiment analysis.

2.1. Sentiment Analysis

Sentiment analysis is a type of data mining research, specifically text mining, which serves to analyse reviews or comments made by individuals on a particular topic in order to obtain the value contained within them [4].

According to [5], sentiment analysis or opinion mining is a branch of data mining used to analyse textual data in the form of opinions that contain polarity, thereby generating information that has a positive, negative, or neutral value.

2.2. Natural Language Processing

Natural Language Processing (NLP) is a field of Artificial Intelligence (AI) that studies human communication with computers through natural language. NLP does not aim to convert spoken language into digital data or vice versa, but rather to understand the meaning of spoken or written language in its natural form and provide an appropriate response, such as displaying specific data [6].

2.3. Text Mining

Text mining is a technique used to extract valuable information from text data, such as student reviews or opinions. In the process, various text processing techniques are used to convert raw data into a text format that can be understood by Machine Learning.

According to [7], text mining, as a branch of data mining, is believed to have higher commercial value than data mining itself, as 80% of every company has information documents in text form. However, data extraction and retrieval are more complex due to unstructured text patterns that make it difficult. Text mining is a comprehensive field of study that permeates almost every aspect of our lives.

2.4. Text Preprocessing

The process of preparing raw document text or datasets is also known as text preprocessing [8]. Text data collected from questionnaires filled out by students and saved in Comma Separated Values (CSV) file format sometimes presents its own challenges due to the presence of non-standard words, regional language usage, or abbreviations not found in the Indonesian Language Dictionary (KBBI). To make the data more structured, a pre-processing stage is therefore necessary. There are several processes involved in pre-processing, as follows:

a) Casefolding: Unrelated words will be changed. For example, words containing capital letters will be changed to lower case.
b) Cleaning: In this process, punctuation marks, special characters, symbols, or characters that are not letters will be removed.
c) Filter Token (tokenizing): Words shorter than four letters or longer than 25 letters will be deleted, for example “tdk”, “yg”, and “gan”.
d) Stopword Removal: Irrelevant words or conjunctions, such as ‘but, with, for, that’ and other connecting words will be removed.
e) Normalization: Standardising non-standardised words into standardised words, such as correcting typos and changing abbreviations into standard forms.
f) Stemming: Affixed words, such as mem-, me-, meny-, meng-, per-, and ber-, will be selected and changed to their root words.

2.5. Term Frequency-Inverse Document (TF-IDF)

Term weighting is a stage that aims to assign weights to each term or word. Calculating these weights requires two things, namely Term Frequency (TF) and Inverse Document Frequency (IDF). The number of times a particular term appears in a document is called TF, while the frequency of a term appearing in all documents is called IDF [9].

According to [10], Term Frequency indicates the frequency of a term that frequently appears in a document. The higher the number of occurrences of a term in a document, the greater its weight or the higher the relevance score it will provide. As explained by [11], TF-IDF is chosen because it is easy to apply and can produce fairly accurate word evaluations. The equation used to calculate TF-IDF is as follows:

$$TF - IDF_{(d,t)} = TF_{(d,t)} \times IDF_{(t)}$$

Description : t = word
d = document

Where : $TF_{(d,t)} = \frac{Number\ of\ t\ words\ in\ document\ d}{total\ words\ in\ document\ d}$

$$IDF_{(t)} = \log\left(\frac{total\ document\ d}{number\ of\ documents\ containing\ the\ word\ d}\right)$$

2.6. Naive Bayes Classifier

Naive Bayes classification is a classification method based on probability and Bayes' theorem, assuming that each value of X is independent or stands alone and is not related to other variables. According to [12]. The classification method using probability and statistics proposed by British scientist Thomas Bayes involves predicting future probabilities based on past experience, hence it is known as Bayes' theorem. The main feature of Naive Bayes Classification is the strong (naive) assumption of independence between each condition or event.

When using the Naive Bayes Classifier method, researchers combine it with the application of the Multinomial Naive Bayes classification model, which is the most common implementation of Naive Bayes for classifying text. The calculation of the Multinomial Naive Bayes model application refers to a formula adapted from research by [13].

$$P(c, d) = \frac{Nc}{N}\ x\ P(t_1\ , c)\ x\ \dots\ x\ P(t_n, c)$$

Explanation:

$P(c, d)$: Probability of a document belonging to class c
Nc : Number of class c documents in the entire document set
N : Total number of documents
tn : Word in document d to n
$P(tn, c)$: Probability of word n given class c
The probability formula for word n used with TF-IDF word weighting can be seen as follows:

$$P(t_n, c) = \frac{Wct+1}{(\sum W'€VW'ct)+B'}$$

Explanation:

Wct : Weighted tfidf or W value of term t in category c
$\sum W'€VW'ct$: Total W of all terms in category c
B' : Total W of unique words (IDF value not multiplied by tf) in all documents

2.7. Google Colaboratory (Google Colab)

Google Colab, or Google Colaboratory, is a web-based platform that allows users to write and run Python code directly in their browser. Users do not need to install software or configure a local environment, which can often be time-consuming and labor-intensive. With Google Colab, anyone with an internet connection and a Google account can access a complete Python programming environment and start writing code immediately.

In this study, Google Colab was used as the primary platform for the entire sentiment analysis process, from text preprocessing, TF-IDF, training the Naïve Bayes Classifier model, to evaluating model performance. The use of Google Colab enhances time and resource efficiency, enabling flexible research methods that are not dependent on specific hardware requirements.

3. Results and Discussion

The research method is an outline of the steps required to conduct structured research, and therefore a framework is developed from the beginning to the final results. In this study, the method used follows the CRISP-DM approach, which consists of several interrelated stages. The stages of CRISP-DM are described as follows:

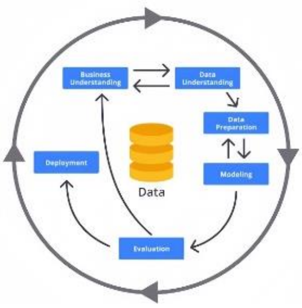

Fig. 1: Research Workflow CRISP-DM

Based on the image above, it can be explained that there are several stages, as follows:

1. Business Understanding

In this first stage, the main focus is to understand the objectives of the research and how analyzing student satisfaction sentiment towards lecturer performance can provide useful insights for universities. Business Understanding aims to set priorities and define the problems to be solved, namely how student satisfaction perceptions of lecturer performance can be analyzed and translated into strategic recommendations.

In this study, the goal is to provide improvement recommendations to universities to enhance teaching quality and academic services based on sentiment analysis results.

2. Data Understanding

Once the research objectives are understood, the next step is to understand the data that will be used in the analysis. In this second stage, data is collected through a questionnaire for STMIK Kaputama lecturers for the 2022-2025 academic year, which contains open-ended questions about student satisfaction with lecturer performance at STMIK Kaputama. The data collected consists of student comments and opinions, which will later be analyzed to classify their sentiments. The collected data was then analyzed to understand the patterns in the text. This analysis included identifying the distribution of sentiments (positive, negative, and neutral) and examining the relevance of the comments to student satisfaction with faculty performance. At this stage, preliminary exploration of the data was also conducted, such as the amount of data received and the types of sentiments present.

3. Data Preparation

This stage involves the process of cleaning and transforming data to prepare it for use in modeling. Data obtained through questionnaires will be processed through several Text Preprocessing steps as follows:

a) Case Folding: Change all text to lowercase to standardize the format and facilitate analysis. The following table shows an example of the case folding stages:

Table 1: Process Case Folding

Komentar	Case Folding
Pengajaran dosen di STMIK Kaputama sangat mudah dipahami oleh saya dan sebagai mahasiswa saya sudah sangat paham tentang materi yang dibawakan. interaksi dosen dengan mahasiswa dalam proses belajar mengajar sudah sangat interaktif antara 2 arah. sikap dan perilaku dosen terhadap mahasiswa sudah selayaknya pengajar dan mahasiswa. menurut saya cara mengajar dosen bermacam - macam ada dosen yang saat mengajar saya sudah sangat paham tentang materi yang dibawakan tapi ada dosen yang mengajar saya kurang paham harus lebih dari 1 kali menjelaskannya.	pengajaran dosen di stmik kaputama sangat mudah dipahami oleh saya dan sebagai mahasiswa saya sudah sangat paham tentang materi yang dibawakan. interaksi dosen dengan mahasiswa dalam proses belajar mengajar sudah sangat interaktif antara 2 arah. sikap dan perilaku dosen terhadap mahasiswa sudah selayaknya pengajar dan mahasiswa. menurut saya cara mengajar dosen bermacam - macam ada dosen yang saat mengajar saya sudah sangat paham tentang materi yang dibawakan tapi ada dosen yang mengajar saya kurang paham harus lebih dari 1 kali menjelaskannya.

b) Cleaning: Remove irrelevant elements such as URLs, emoticons, numbers, and special characters. Here is an example of the cleaning stages in the following table:

Table 2: Process Case Cleaning

Case Folding	Cleaning
pengajaran dosen di stmik kaputama sangat mudah dipahami oleh saya dan sebagai mahasiswa saya sudah sangat paham tentang materi yang dibawakan. interaksi dosen dengan mahasiswa dalam proses belajar mengajar sudah sangat interaktif antara 2 arah. sikap dan perilaku dosen terhadap mahasiswa sudah selayaknya pengajar dan mahasiswa. menurut saya cara mengajar dosen bermacam - macam ada dosen yang saat mengajar saya sudah sangat paham tentang materi yang dibawakan tapi ada dosen yang mengajar saya kurang paham harus lebih dari 1 kali menjelaskannya.	pengajaran dosen di stmik kaputama sangat mudah dipahami oleh saya dan sebagai mahasiswa saya sudah sangat paham tentang materi yang dibawakan interaksi dosen dengan mahasiswa dalam proses belajar mengajar sudah sangat interaktif antara arah sikap dan perilaku dosen terhadap mahasiswa sudah selayaknya pengajar dan mahasiswa menurut saya cara mengajar dosen bermacam macam ada dosen yang saat mengajar saya sudah sangat paham tentang materi yang dibawakan tapi ada dosen yang mengajar saya kurang paham harus lebih dari kali menjelaskannya

c) Tokenizing: Breaking down text into smaller words or tokens. Here is an example of the tokenizing process in the following table

Table 3: Process Tokenizing

Cleaning	Tokenizing
pengajaran dosen di stmik kaputama sangat mudah dipahami oleh saya dan sebagai mahasiswa saya sudah sangat paham tentang materi yang dibawakan interaksi dosen dengan mahasiswa dalam proses belajar	['pengajaran', 'dosen', 'di', 'stmik', 'kaputama', 'sangat', 'mudah', 'dipahami', 'oleh', 'saya', 'dan', 'sebagai', 'mahasiswa', 'saya', 'sudah', 'sangat', 'paham', 'tentang',

Cleaning	Tokenizing
mengajar sudah sangat interaktif antara arah sikap dan perilaku dosen terhadap mahasiwa sudah selayaknya pengajar dan mahasiswa menurut saya cara mengajar dosen bermacam macam ada dosen yang saat mengajar saya sudah sangat paham tentang materi yang dibawakan tapi ada dosen yang mengajar saya kurang paham harus lebih dari kali menjelaskannya	'materi', 'yang', 'dibawakan', 'interaksi', 'dosen', 'dengan', 'mahasiswa', 'dalam', 'proses', 'belajar', 'mengajar', 'sudah', 'sangat', 'interaktif', 'antara', 'arah', 'sikap', 'dan', 'perilaku', 'dosen', 'terhadap', 'mahasiwa', 'sudah', 'selayaknya', 'pengajar', 'dan', 'mahasiswa', 'menurut', 'saya', 'cara', 'mengajar', 'dosen', 'bermacam', 'macam', 'ada', 'dosen', 'yang', 'saat', 'mengajar', 'saya', 'sudah', 'sangat', 'paham', 'tentang', 'materi', 'yang', 'dibawakan', 'tapi', 'ada', 'dosen', 'yang', 'mengajar', 'saya', 'kurang', 'paham', 'harus', 'lebih', 'dari', 'kali', 'menjelaskannya']

d) Stopword Removal: Remove common words that do not provide important information, such as “yang,” “dan,” “dll.” Here is an example of the stopword removal stage in the following table:

Table 4: Process Stopword Removal

Tokenizing	Stopword Removal
['pengajaran', 'dosen', 'di', 'stmik', 'kaputama', 'sangat', 'mudah', 'dipahami', 'oleh', 'saya', 'dan', 'sebagai', 'mahasiswa', 'saya', 'sudah', 'sangat', 'paham', 'tentang', 'materi', 'yang', 'dibawakan', 'interaksi', 'dosen', 'dengan', 'mahasiswa', 'dalam', 'proses', 'belajar', 'mengajar', 'sudah', 'sangat', 'interaktif', 'antara', 'arah', 'sikap', 'dan', 'perilaku', 'dosen', 'terhadap', 'mahasiwa', 'sudah', 'selayaknya', 'pengajar', 'dan', 'mahasiswa', 'menurut', 'saya', 'cara', 'mengajar', 'dosen', 'bermacam', 'macam', 'ada', 'dosen', 'yang', 'saat', 'mengajar', 'saya', 'sudah', 'sangat', 'paham', 'tentang', 'materi', 'yang', 'dibawakan', 'tapi', 'ada', 'dosen', 'yang', 'mengajar', 'saya', 'kurang', 'paham', 'harus', 'lebih', 'dari', 'kali', 'menjelaskannya']	['pengajaran', 'dosen', 'stmik', 'kaputama', 'mudah', 'dipahami', 'mahasiswa', 'paham', 'materi', 'dibawakan', 'interaksi', 'dosen', 'mahasiswa', 'proses', 'belajar', 'mengajar', 'interaktif', 'arah', 'sikap', 'perilaku', 'dosen', 'terhadap', 'mahasiwa', 'selayaknya', 'pengajar', 'mahasiswa', 'mengajar', 'dosen', 'bermacam', 'macam', 'dosen', 'mengajar', 'paham', 'materi', 'dibawakan', 'dosen', 'mengajar', 'kurang', 'paham', 'kali', 'menjelaskannya']

e) Normalisasi: Standardizing non-standard words into standard words, such as correcting typos and changing abbreviations into standard forms. Here is an example of the normalization process in the following table:

Table 5: Process Normalisasi

Stopword Removal	Normalisasi
['pengajaran', 'dosen', 'stmik', 'kaputama', 'mudah', 'dipahami', 'mahasiswa', 'paham', 'materi', 'dibawakan', 'interaksi', 'dosen', 'mahasiswa', 'proses', 'belajar', 'mengajar', 'interaktif', 'arah', 'sikap', 'perilaku', 'dosen', 'terhadap', 'mahasiwa', 'selayaknya', 'pengajar', 'mahasiswa', 'mengajar', 'dosen', 'bermacam', 'macam', 'dosen', 'mengajar', 'paham', 'materi', 'dibawakan', 'dosen', 'mengajar', 'kurang', 'paham', 'kali', 'menjelaskannya']	['pengajaran', 'dosen', 'stmik', 'kaputama', 'mudah', 'paham', 'mahasiswa', 'paham', 'materi', 'bawa', 'interaksi', 'dosen', 'mahasiswa', 'proses', 'belajar', 'ajar', 'interaktif', 'arah', 'sikap', 'perilaku', 'dosen', 'hadap', 'mahasiswa', 'layak', 'ajar', 'mahasiswa', 'ajar', 'dosen', 'macam', 'macam', 'dosen', 'ajar', 'paham', 'materi', 'bawa', 'dosen', 'ajar', 'kurang', 'paham', 'kali', 'jelas']

f) Stemming: Changing words to their basic form (root word). Here are examples of the stemming stages in the following table:

Table 6: Proses Stemming

Normalisasi	Stemming
['pengajaran', 'dosen', 'stmik', 'kaputama', 'mudah', 'paham', 'mahasiswa', 'paham', 'materi', 'bawa', 'interaksi', 'dosen', 'mahasiswa', 'proses', 'belajar', 'ajar', 'interaktif', 'arah', 'sikap', 'perilaku', 'dosen', 'hadap', 'mahasiswa', 'layak', 'ajar', 'mahasiswa', 'ajar', 'dosen', 'macam', 'macam', 'dosen', 'ajar', 'paham', 'materi', 'bawa', 'dosen', 'ajar', 'kurang', 'paham', 'kali', 'jelas']	['ajar', 'dosen', 'stmik', 'kaputama', 'mudah', 'paham', 'mahasiswa', 'paham', 'materi', 'bawa', 'interaksi', 'dosen', 'mahasiswa', 'proses', 'belajar', 'ajar', 'interaktif', 'arah', 'sikap', 'laku', 'dosen', 'hadap', 'mahasiswa', 'layak', 'ajar', 'mahasiswa', 'ajar', 'dosen', 'macam', 'macam', 'dosen', 'ajar', 'paham', 'materi', 'bawa', 'dosen', 'ajar', 'kurang', 'paham', 'kali', 'jelas']

After Text Preprocessing, the data will be converted into numerical representations using the TF-IDF (Term Frequency – Inverse Document Frequency) technique to measure the importance of each term in the document compared to the entire dataset.

4. Modeling

Modeling At this stage, the Naïve Bayes Classifier mode is used to classify student sentiment based on the processed data. This model was chosen for its simplicity in classifying text into positive, negative, and neutral categories based on probability. The modeling process begins with training the model using the previously prepared training data. The training data serves to teach the model how to classify new data based on patterns found in the training data. After the model is trained, testing is carried out to evaluate the model's performance using test data that has not been used in training.

5. Evaluation

After the model is built, the next step is to evaluate its performance. At this stage, evaluation metrics such as accuracy, precision, recall, and F1-Score are used to assess how well the model classifies sentiment correctly. These metrics help measure the effectiveness of the model in identifying positive, negative, and neutral sentiments. If the evaluation results show that the model is not working well, for example, low accuracy or imbalance between classified sentiments, adjustments can be made at the Data Preparation or Modeling stage, such as changing model parameters or performing fine-tuning.

6. Deployment

This final stage involves applying the results of the analysis into practice. At this stage, the results of the sentiment analysis can be used by the university as a basis for decision-making in improving the performance of lecturers at STMIK Kaputama. The recommendations generated from this analysis can be used to develop better learning and teaching improvement strategies that are relevant to the needs of students.

In addition, the results of the evaluated model can be presented in the form of reports or data visualizations to facilitate the university's understanding of student satisfaction with lecturer performance.

Table 7: training data example

ID Documen	Before Preprocessing	After Preprocessing	Class Sentiment
D1	Sejauh ini saya sudah puas. Berdasarkan pengalaman yang didapat, sistem belajar dengan sudah membagikan seluruh materi perkuliahan sejak pertemuan awal cukup membuat mahasiswa terdorong untuk belajar lebih. Tidak hanya itu, berdiskusi dengan mengerjakan tugas yang berbentuk diskusi yang tenggatnya setiap pertemuan tersebut mampu membuat mahasiswa terdorong untuk harus bisa, sehingga cukup efektif.	puas dasar alam sistem ajar bagi materi kuliah temu mahasiswa dorong ajar diskusi tugas bentuk diskusi tenggat temu mahasiswa dorong efektif'	Positive
D2	pengajaran sudah bagus semoga lebih baik lagi	ajar bagus baik	Positive
D3	Kurang puas dengan kinerja dosen kaputama, mahasiswa di tuntut agar disiplin tetapi ada beberapa dosen sendiri tidak menerapkan kedisiplinan tersebut	Kurang puas kinerja dosen kaputama tuntut siplin dosen tidak terap plin	Negative
D4	Pengalaman saya selama menjadi mahasiswa di kaputama binjai saya merasa kurang puas terhadap itu	Alam mahasiswa kaputama binjai rasa kurang puas	Negative
D5	Kinerja dosen di STMIK Kaputama cukup baik Materi dijelaskan..beberapa dosen terbuka, tapi tidak semuanya aktif berinteraksi. Pemahaman saya terhadap materi tergantung cara mengajar.	kinerja dosen stmik kaputama materi jelas dosen terbuka aktif interaksi pemahaman materi tergantung ajar	Neutral

3.1. TF-IDF Weighting

Word weighting aims to determine the importance of a word in a document by utilizing the Term Frequency - Inverse Document Frequency (TF-IDF) method. Term Frequency measures how often a word appears in a single document, while Inverse Document Frequency assesses how rarely the word is found in the entire collection of documents. TF-IDF aims to measure the relevance of a word by considering how often the word appears in a particular document and how commonly the word is found in other documents.

1. Calculating TF Values

The sample to be calculated is in the sentence found in document 1. First, calculate the TF(d,t) value word by word. The following is the calculation of the TF(d,t) value:

The word "puas"
D1 = TF(d,t) = (Number of words t in document d)/(total words in document d) = 1/21 = 0.048
D2 = TF(d,t) = (Number of words t in document d)/(total words in document d) = 0/3 = 0
D3 = TF(d,t) = (Number of words t in document d)/(total words in document d) = 1/9 = 0.111
D4 = TF(d,t) = (Number of t words in document d)/(total words in document d) = 1/7 = 0.143
D5 = TF(d,t) = (Number of t words in document d)/(total words in document d) = 0/14 = 0

The calculation of TF(d,t) values for each word continues in this manner. The results of all TF(d,t) value calculations can be seen in the table below:

Table 8: TF Results

Term	Frequency					Value TF(d,f)				
	D1	D2	D3	D4	D5	D1	D2	D3	D4	D5
ajar	2	1	0	0	1	0,095	0,333	0	0	0,071
aktif	0	0	0	0	1	0	0	0	0	0,071
alam	1	0	0	1	0	0,048	0	0	0,143	0
bagi	1	0	0	0	0	0,048	0	0	0	0
bagus	0	1	0	0	0	0	0,333	0	0	0
baik	0	1	0	0	0	0	0,333	0	0	0
bentuk	1	0	0	0	0	0,048	0	0	0	0
binjai	0	0	0	1	0	0	0	0	0,143	0
dasar	1	0	0	0	0	0,048	0	0	0	0
diskusi	2	0	0	0	0	0,095	0	0	0	0

2. Calculating IDF Values

The sample to be calculated is in the sentence found in document 1. First, calculate the IDF(t) value word by word. The following is the calculation of the IDF(t) value:

The word “puas”

IDF(t) = log((total documents d)/(number of documents containing the word d)) = log(5/3)

= 0.222

The calculation of IDF(t) values for each word continues in this manner. The results of all IDF(t) value calculations can be seen in the table below:

Table 9: IDF Results

Term	Frequency					Value TF(d,f)					IDF
	D1	D2	D3	D4	D5	D1	D2	D3	D4	D5	
ajar	2	1	0	0	1	0,095	0,333	0	0	0,071	0,097
aktif	0	0	0	0	1	0	0	0	0	0,071	0,699
alam	1	0	0	1	0	0,048	0	0	0,143	0	0,398
bagi	1	0	0	0	0	0,048	0	0	0	0	0,699
bagus	0	1	0	0	0	0	0,333	0	0	0	0,699
baik	0	1	0	0	0	0	0,333	0	0	0	0,699
bentuk	1	0	0	0	0	0,048	0	0	0	0	0,699
binjai	0	0	0	1	0	0	0	0	0,143	0	0,699
dasar	1	0	0	0	0	0,048	0	0	0	0	0,699
diskusi	2	0	0	0	0	0,095	0	0	0	0	0,398

3. Calculating TF-IDF

At this stage, the TF and IDF values will be multiplied to produce the TF-IDF score for each word. The results of all TF-IDF calculations can be seen in the table below:

Table 9: TF-IDF Results

Term	Value TF-IDF				
	D1(Positive)	D2(Positive)	D3(Negative)	D4(Negative)	D5(Neutral)
ajar	0,009	0,032	0	0	0,007

Term	Value TF-IDF				
	D1(Positive)	D2(Positive)	D3(Negative)	D4(Negative)	D5(Neutral)
aktif	0	0	0	0	0,05
alam	0,019	0	0	0,057	0
bagi	0,034	0	0	0	0
bagus	0	0,233	0	0	0
baik	0	0,233	0	0	0
bentuk	0,034	0	0	0	0
binjai	0	0	0	0,1	0
dasar	0,034	0	0	0	0
diskusi	0,038	0	0	0	0

3.2. Test Data Classification Process

The first step in constructing test data from five documents as probability calculations for each word can be seen in Table 7, and one document as test data for calculating the Naive Bayes Classifier (NBC) classification.

Table 10: Test Data Example

ID Documen	Before Preprocessing	After Preprocessing	Class Sentiment
D6	Cukup puas karena ada beberapa dosen yang menjelaskan ya cukup Jelas namun ada beberapa dosen juga yang menjelaskannya tidak berkaitan dengan materi	cukup puas dosen jelas cukup jelas dosen jelas tidak kait materi	?

The second step is to calculate the probability for each class (from the class distribution in the dataset). Table 11 shows the calculation of $P(tn, c)$ for each class.

Table 11: Calculation formation $P(t_n, c)$

C	Positive Sentiment	Negative Sentiment	Neutral Sentiment
	α	α	α
	1	1	1
$\sum W'€VW'c$t	0,955	0,817	0,465
B'	34	34	34
$P(t_n, c)$	$\frac{n+1}{0{,}955+34}$	$\frac{n+1}{0{,}817+34}$	$\frac{n+1}{0{,}465+34}$

This table shows how the probability value $P(tn, c)$ is calculated in the context of combining TF-IDF values with the application of Multinomial Naive Bayes and also, using the Laplace Smoothing technique.

Explanation:
A : Laplace smoothing (usually 1, to avoid a probability of 0).
$\sum W'€VW'c$t : Total number of W from all terms in category c.
B' : Number of unique W words (IDF values are not multiplied by tf) in all documents.

The third step is to calculate $P(tn, c)$ for all words in the document and class, using Laplace smoothing. The calculation results can be seen in the table below.

Table 12 : Probability results for each word when given a class

Gn	$P(t_n, c)$		
	Positive P(d)= $\frac{2}{5}$	Negative P(d)= $\frac{2}{5}$	Neutral P(d)= $\frac{1}{5}$
ajar	0,07	0,029	0,036
aktif	0,029	0,029	0,079
alam	0,048	0,086	0,029
bagi	0,063	0,029	0,029
bagus	0,262	0,029	0,029

Gn	$P(t_n, c)$		
	Positive P(d)=$\frac{2}{5}$	Negative P(d)=$\frac{2}{5}$	Neutral P(d)=$\frac{1}{5}$
baik	0,262	0,029	0,029
bentuk	0,063	0,029	0,029
binjai	0,029	0,129	0,029
dasar	0,063	0,029	0,029
diskusi	0,067	0,029	0,029

The fourth step is to calculate the final probability $P(c, d)$ for the document in the test data. The calculation process can be seen as follows:

a. Probability for the Positive class

Pr(Positive |d) = $\frac{\text{Nc}}{\text{N}}$× Pr(cukup|Positive) × Pr(puas|Positive) × Pr(dosen|Positive) × Pr(jelas|Positive) × Pr(dosen|Positive) × Pr(jelas| Positive) .× Pr (tidak| Positive) .× Pr (kait|Positive) .× Pr (materi|Positive)
$P(\text{c, d}) = 0{,}4\ x\ 0{,}029\ x\ 0{,}04\ x\ 0{,}029\ x\ 0{,}029\ x\ 0{,}029\ x\ 0{,}029\ x\ 0{,}029\ x\ 0{,}029\ x\ 0{,}04 \quad = 0{,}00464619373{\times}10^{-14}$

b. Probability for the Negative class

Pr(Negative|d) = $\frac{\text{Nc}}{\text{N}}$ × Pr(cukup|Negative) × Pr(puas|Negative) × Pr(dosen|Negative) × Pr(jelas|Negative) × Pr(dosen|Negative) × Pr(jelas|Negative) .× Pr (tidak|Negative) .× Pr (kait|Negative) .× Pr (materi|Negative)
$P(\text{c, d}) = 0{,}4\ x\ 0{,}029x\ 0{,}086\ x\ 0{,}054\ x\ 0{,}029\ x\ 0{,}054\ x\ \ 0{,}029\ x\ 0{,}029\ x\ 0{,}029\ x\ 0{,}029 = 0{,}0251198539{\times}10^{-14}$

c. Probability for the Neutral class

Pr(Neutral|d) = $\frac{\text{Nc}}{\text{N}}$ × Pr(cukup|Neutral) × Pr(puas|Neutral) × Pr(dosen|Neutral) × Pr(jelas|Neutral) × Pr(dosen|Neutral) × Pr(jelas|Neutral) .× Pr (tidak|Neutral) .× Pr (kait|Neutral) .× Pr (materi|Neutral)
$P(\text{c, d}) \ = \ 0{,}2\ x\ 0{,}029\ x\ 0{,}029\ x\ 0{,}061\ x\ 0{,}079\ x\ 0{,}061\ x\ 0{,}079\ x\ 0{,}029\ x\ 0{,}029\ x\ \ 0{,}061 = 3{,}23743328{\times}10^{-14}$

Compare the final probability results:
- Positive Probability = 0.00464619373×10−14
- Negative Probability = 0.0251198539×10−14
- Neutral Probability = 3.23743328×10−14

Because 3.23743328×10−14 > 0.0251198539×10−14 , 0.00464619373×10−14, document D6 with the comment "quite satisfied, lecturer is quite clear, lecturer is clear, no connection to the material" will be classified as **Neutral.**

3.3. Evaluation

After the classification process is performed on the test data using the Naive Bayes Classifier algorithm, the next step is to evaluate the model results by comparing the model predictions with the predetermined original labels.

Table 13 : Original Label

ID Document	Before Preprocessing	After Preprocessing	Class Sentiment
D6	Cukup puas karena ada beberapa dosen yang menjelaskan ya cukup Jelas namun ada beberapa dosen juga yang menjelaskannya tidak berkaitan dengan materi	cukup puas dosen jelas cukup jelas dosen jelas tidak kait materi	?

For example, in a certain test data, it is known that the highest probability value is obtained in the neutral class, compared to the positive and negative classes. Therefore, the model decides that the data belongs to the neutral class.

Table 14 : Prediction Results

Before Preprocessing	After Preprocessing	Class Sentiment
Cukup puas karena ada beberapa dosen yang menjelaskan ya cukup Jelas namun ada beberapa dosen juga yang menjelaskannya tidak berkaitan dengan materi	cukup puas dosen jelas cukup jelas dosen jelas tidak kait materi	Neutral

To measure the performance of the classification model used, this study applied an evaluation method based on a confusion matrix and calculated several evaluation metrics, namely accuracy, precision, recall, and F1-score. Next, it displays the sentiment visualization results shown in the image below:

1. Sentiment Visualization with Pie Charts

A pie chart divided into several slices (like pieces of cake), where each slice represents the proportion or percentage of the overall data: 76.4% positive, 13.9% negative, and 9.7% neutral.

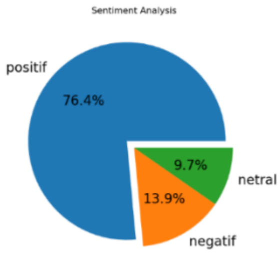

Fig. 2: Research Workflow CRISP-DM

2. Sentiment Visualization Word cloud Sentiment

visualization of words from a text, where the most frequently occurring words are displayed larger

Fig. 3: Word Cloud Positve

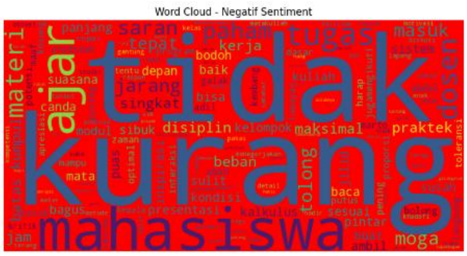

Fig. 4: Word Cloud Negative

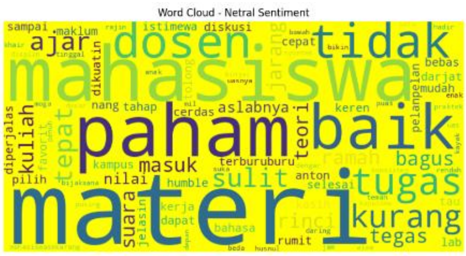

Fig. 5: Word Cloud Neutral

4. A step before the final submission

After the training process is complete, the next step is to test the performance of the Naïve Bayes Classifier model on the test data to determine the model's ability to classify comments into negative, positive, or neutral sentiment categories, as well as to measure performance using accuracy metrics. Based on the image below, the Naive Bayes Classifier model achieved an accuracy rate of 76.92%.

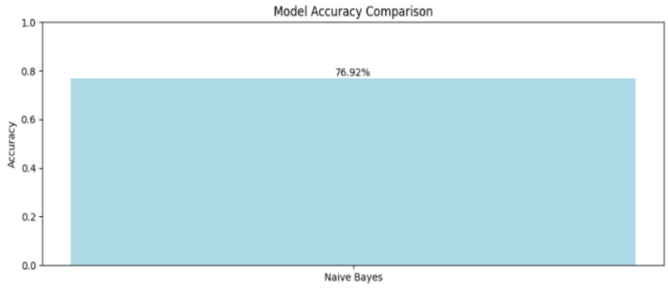

Fig. 6: Accuracy

To provide a clearer picture of the model's performance and sentiment data distribution, a confusion matrix, sentiment label distribution, and word frequency based on sentiment class were used. This visualization helps in understanding the model's prediction patterns, the proportion of data in each class, and the words that are dominant in each sentiment category.

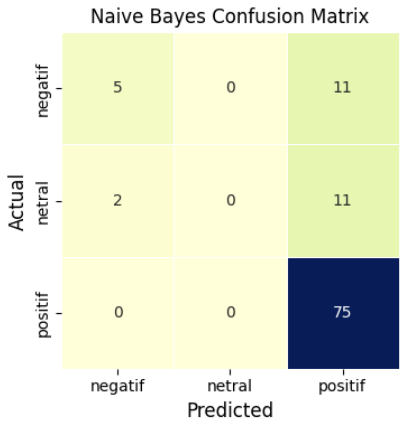

Fig. 7: Confusion Matix

Figure 7 Confusion Matrix of the Naive Bayes classification model showing the model's performance. This matrix reveals that the model is very good at predicting the “positive” class with 75 correct predictions, but fails completely in predicting the ‘neutral’ class (0 correct predictions) and performs poorly on the “negative” class (only 5 correct predictions). This indicates that the model has a strong tendency to classify data as “positive” and is ineffective for the ‘negative’ and “neutral” classes.

Based on the evaluation results presented, it can be concluded that the classification model used has varying performance depending on the sentiment class. Overall, the model achieved an accuracy of 76.9%, demonstrating its ability to predict sentiment with a fairly good level of accuracy. The following is a visualization of the naive bayes classifier classification mode report.

Classification Report for Naive Bayes:

	precision	recall	f1-score	support
negatif	0.714	0.312	0.435	16.000
netral	0.000	0.000	0.000	13.000
positif	0.773	1.000	0.872	75.000
accuracy	0.769	0.769	0.769	0.769
macro avg	0.496	0.438	0.436	104.000
weighted avg	0.667	0.769	0.696	104.000

Fig. 8: Classification Report

5. Conclusion

Based on the analysis results, this study successfully classified 518 student comments on lecturer performance at STMIK Kaputama into positive, negative, and neutral sentiments, with positive sentiments being the most dominant (396 comments). The use of the Naive Bayes Classifier method with Text Preprocessing and TF-IDF weighting proved to be effective in achieving an accuracy of 76.92%. This model excelled in predicting positive sentiment with 100% recall and 77.3% precision. However, the model showed weaknesses in classifying neutral sentiment and had suboptimal performance on negative sentiment (recall of only 31.2%), which was caused by data imbalance. As a suggestion, data expansion, especially for neutral and negative sentiments, as well as testing with other methods such as SVM or KNN, can be done to improve model performance. The results of this analysis can be used as a reference for campus management to improve the quality of education and encourage active student participation in providing feedback.

References

[1] S. Tinggi and I. E. Jambi, “ANALISIS KEPUASAN MAHASISWA TERHADAP KINERJA DOSEN SEKOLAH TINGGI ILMU EKONOMI JAMBI Henky Setiadi,” *Jurnal ManajemenTerapan dan Keuangan (Mankeu)*, vol. 10, no. 03, 2021.

[2] Muhammad Rizky R Ritonga, Marto Sihombing, and Selfira Selfira, “Pemodelan K-Nearest Neighbor Untuk Identifikasi Pola Kepuasan Mahasiswa Terhadap Pelayanan Kampus (Studi Kasus : STMIK Kaputama),” *Modem : Jurnal Informatika dan Sains Teknologi.*, vol. 2, no. 4, pp. 152–161, Sep. 2024, doi: 10.62951/modem.v2i4.238.

[3] E. Apriliyani and Y. Salim, “Analisis performa metode klasifikasi Naïve Bayes Classifier pada Unbalanced Dataset,” *Indonesian Journal of Data and Science (IJODAS)*, vol. 3, no. 2, pp. 47–54, 2022.

[4] I. Komang *et al.*, “ANALISA SENTIMEN MAHASISWA TERHADAP LAYANAN STMIK PRIMAKARA MENGGUNAKAN ALGORITMA NAIVE BAYES DAN K-NEAREST NEIGHBOR,” 2023.

[5] B. Laurensz, A. Sentimen, and E. Sediyono, “Analisis Sentimen Masyarakat terhadap Tindakan Vaksinasi dalam Upaya Mengatasi Pandemi Covid-19 (Analysis of Public Sentiment on Vaccination in Efforts to Overcome the Covid-19 Pandemic),” 2021.

[6] J. Suciadi, “STUDI ANALISIS METODE-METODE PARSING DAN INTERPRETASI SEMANTIK PADA NATURAL LANGUAGE PROCESSING.” [Online]. Available: http://puslit.petra.ac.id/journals/informatics/

[7] A. Firdaus and W. I. Firdaus, “Text Mining Dan Pola Algoritma Dalam Penyelesaian Masalah Informasi : (Sebuah Ulasan),” 2021.

[8] D. Alita and A. Rahman, “Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier,” 2020.

[9] T. Ernayanti, M. Mustafid, A. Rusgiyono, and A. R. Hakim, “PENGGUNAAN SELEKSI FITUR CHI-SQUARE DAN ALGORITMA MULTINOMIAL NAÏVE BAYES UNTUK ANALISIS SENTIMEN PELANGGGAN TOKOPEDIA,” *Jurnal Gaussian*, vol. 11, no. 4, pp. 562–571, Feb. 2023, doi: 10.14710/j.gauss.11.4.562-571.

[10] C. H. Yutika, A. Adiwijaya, and S. Al Faraby, “Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naïve Bayes,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 2, p. 422, Apr. 2021, doi: 10.30865/mib.v5i2.2845.

[11] D. Alita and R. A. Shodiqin, “Sentimen Analisis Vaksin Covid-19 Menggunakan Naive Bayes Dan Support Vector Machine,” *Journal of Artificial Intelligence and Technology Information (JAITI)*, vol. 1, no. 1, pp. 1–12, Mar. 2023, doi: 10.58602/jaiti.v1i1.20.

[12] A. Felicia Watratan, A. B. Puspita, D. Moeis, S. Informasi, and S. Profesional Makassar, “Implementasi Algoritma Naive Bayes Untuk Memprediksi Tingkat Penyebaran Covid-19 Di Indonesia,” 2020. [Online]. Available: http://journal.isas.or.id/index.php/JACOST

[13] F. A. Ramadhan, S. H. Sitorus, and T. Rismawan, “Penerapan Metode Multinomial Naïve Bayes untuk Klasifikasi Judul Berita Clickbait dengan Term Frequency - Inverse Document Frequency,” *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 11, no. 1, p. 70, Jan. 2023, doi: 10.26418/justin.v11i1.57452.