DOI: https://doi.org/10.52436/1.jutif.2025.6.1.4061
p-ISSN: 2723-3863
e-ISSN: 2723-3871

A TOPIC-BASED APPROACH FOR RECOMMENDING UNDERGRADUATE THESIS SUPERVISOR USING LDA WITH COSINE SIMILARITY

Laila Rahmatin Nisa*[1], Ardytha Luthfiarta[2], Adhitya Nugraha[3], Md. Mahadi Hasan[4], Kang, Andini Wulandari[5] , Alam Muhammad Huda[6]

[1,2,3]Informatics Engineering Department, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia
[4]School of Engineering & Physical Science, North South University, Bangladesh
Email: [1]111202113475@mhs.dinus.ac.id, [2]ardytha.luthfiarta@dsn.dinus.ac.id, [3]adhitya@dsn.dinus.ac.id, [4]mahadi.hasan.232@northsouth.edu, [5]111202113273@mhs.dinus.ac.id[6], 111202113488@mhs.dinus.ac.id

(Article received: October 30, 2024; Revision: December 26, 2024; published: February 20, 2025)

Abstract

The thesis is one of the critical factors in determining student graduation. While working on the thesis, students will be guided by a lecturer who has the role and responsibility to ensure that students can prepare the thesis well so that the thesis is ready to be tested and is of good quality. Therefore, selecting a supervisor with the same expertise as the thesis topic is essential in determining students' success in completing their thesis. So far, the selection of thesis supervisors at Dian Nuswantoro University still needs to be done manually by students, so the lack of information about the supervisor can hinder students in determining the supervisor. This study aims to model the topic of lecturer research publications taken from the ResearchGate and Google Scholar platforms so that it is easier for students to choose a thesis supervisor whose research topic is relevant to the student's thesis using the Latent Dirichlet Allocation method. The LDA method will mark each word in the topic in a semi-random distribution. It will calculate the probability of the topic in the dataset and the likelihood of the word against the topic for each iteration. The results of LDA modeling present six main topics of lecturer research with the highest coherence score of 0.764, and then the resulting topics and thesis titles will be compared using cosine similarity. Students can use The highest cosine value as a reference when determining the right thesis topic. Thus, the supervisor selection process will be more focused and in accordance with the student's research interests.

Keywords: *Google Scholar, Latent Dirichlet Allocation, ResearchGate , Thesis advisor, Topic modeling.*

1. INTRODUCTION

Undergraduate students must meet several requirements to achieve a bachelor's degree, including completing a final assignment (thesis). A thesis is one of the graduation requirements for students[1], [2] of the Informatics Engineering study program, Faculty of Computer Science, Dian Nuswantoro University. Provisions regarding the thesis are regulated by each faculty, following university standards[3].

In compiling a thesis, students urgently require a supervisor[2]. Selecting a supervisor with the same research interests as the student's thesis topic can contribute significantly to the success of the thesis writing process. Lecturer research is one of the crucial components that students must consider when determining their thesis supervisor.

The previous research conducted by Ridwan Rismanto et al. (2020) on the “Research Supervisor Recommendation System Based on Topic Conformity” stated that the determination of the final project supervisor is an important factor in the work of the student's final project[4]. Another research also stated that assigning an academic supervisor whose expertise is strongly relevant to the thesis topic has become a crucial task[1].

However, many students need help finding lecturers whose research interests align with the topics they will study[5]. This problem is exacerbated by the large number of research publications spread across various digital platforms, such as ResearchGate and Google Scholar, which makes it difficult for students to select lecturers whose research topics are relevant to their interests.

In the academic field, evaluating the similarity of undergraduate students' thesis topics is essential not only to identify previously similar titles but also to recommend appropriate supervisors[6]. The urgency of this research lies in the need to provide an effective solution for students to choose a supervisor based on similarity in research topics. Although public access to lecturer publications is already available through these platforms, not all students have the time or ability to manually review the many publications available. Thus, a tool is needed to process these publications to produce relevant recommendations automatically.

One approach that can be used to solve this problem is Topic Modelling. Topic modeling stands as one of the most powerful techniques in text mining, enabling data mining, uncovering hidden patterns, and identifying relationships within data and text documents[7]. Topic modeling constitutes a computational learning methodology that is extensively employed in Natural Language Processing (NLP) applications to deduce themes within unstructured textual data [8]. Topic modeling serves as a prevalent statistical instrument for extracting latent variables from extensive datasets [9]. Among the most commonly utilized topic modeling algorithms is the Latent Dirichlet Allocation (LDA) method, which is regarded as both flexible and adaptive[10]. The primary aim of an LDA model is to identify multiple topics that, when combined, could precisely reconstruct the original corpus. As LDA represents an unsupervised approach, topics remain undefined a priori but are acquired by the model through the association of words with topics based on their distribution.

The Latent Dirichlet Allocation (LDA) technique as a topic model has been extensively employed in antecedent investigations[11]. It is particularly well adapted for application with textual information; however, it has also been utilized for the examination of bioinformatics data[4], social data[1], and environmental data. LDA is considered capable of summarizing, grouping, connecting, and processing large amounts of text data[12].

Previous research that focused on the problem of supervisor recommendations using the Cosine Similarity method has been conducted by [4],[13], with accuracy levels of 75%,and 91.3%, respectively.The topic modelling approach and similarity measure have been applied in recommendation systems[14]. LDA extracts users' latent interests and provides relevant item recommendations in e-commerce. LDA is an effective method for topic extraction; however, its limitation lies in occasionally producing topics that are difficult to interpret. To address this issue, the TF-IDF feature selection method is applied to filter out less important words, enabling LDA to generate more coherent and meaningful topics[15].

However, applying this approach in an academic context, especially for final project supervisor recommendations, has yet to be widely explored. This study aims to fill this gap by adapting the LDA approach to model lecturers' research topics based on supervisor publications and using cosine similarity to match them with students' research titles. This topic modeling technique can group various publications into several main themes. LDA allows the identification of themes or topics from a collection of publications that can be used as a reference for students in determining the appropriate supervisor. The problem-solving plan includes collecting lecturer publication data from digital platforms, topic modeling using LDA, and presenting the results as a recommendation system that is easily accessible to students. The results of this study make it easier for students to find supervisors appropriate for their research topics.

2. RESEARCH METHOD

This research was conducted through several stages using the research method. The research stages can be seen in Figure 1.

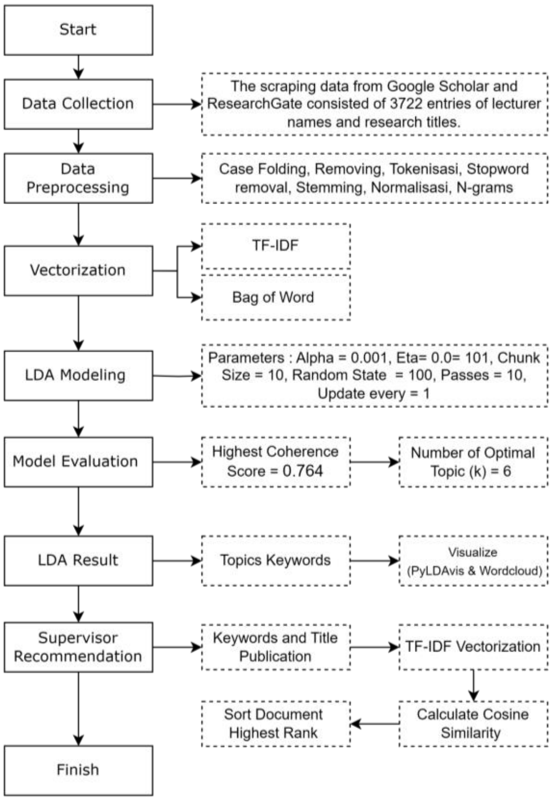

Start
Data Collection
The scraping data from Google Scholar and ResearchGate consisted of 3722 entries of lecturer names and research titles.
Data Preprocessing
Case Folding, Removing, Tokenisasi, Stopword removal, Stemming, Normalisasi, N-grams
Vectorization
TF-IDF
Bag of Word
LDA Modeling
Parameters : Alpha = 0.001, Eta= 0.0= 101, Chunk Size = 10, Random State = 100, Passes = 10, Update every = 1
Model Evaluation
Highest Coherence Score = 0.764
Number of Optimal Topic (k) = 6
LDA Result
Topics Keywords
Visualize (PyLDAvis & Wordcloud)
Supervisor Recommendation
Keywords and Title Publication
TF-IDF Vectorization
Calculate Cosine Similarity
Sort Document Highest Rank
Finish

Figure 1. Research Flow

2.1. Data Collection Method

Research data was collected using web scraping from the Google Scholar and ResearchGate platforms. The scraping process was implemented using Python and the Scholarly library, which facilitates the automated extraction of research publication data based on the unique user IDs of lecturers' research accounts. The extracted data included authors and publication titles. The obtained data was stored in .csv format for further analysis and processing.

2.2. Preprocessing Data

Before performing topic modeling, the data must undergo several preprocessing stages. This preprocessing aims to tidy up the data and ensure that only relevant words are analyzed.

Pre-processing significantly transfers text from human language to machine-readable format in text mining techniques. The pre-processing stage is essential for structuring unstructured text and keeping the keywords helpful in representing the category of text topics. Natural language text can contain many words without specific meaning, such as prepositions, pronouns, etc. [16] To simplify the text data, clean data, and reduce noise.

The following are the pre-processing stages of data in this study.

a. Case Folding

Case folding is a stage to standardize all letter characters in a document by changing capital letters to lowercase letters[17].

b. Filtering

Removing all numbers, signs, symbols, particular words, letters, and punctuation that did not add to the meaning of the text[10].

c. Tokenization

Tokenization is splitting the text into smaller pieces called (tokens). This step is crucial because mistakes made in this step can spread to subsequent parts and affect the accuracy of the final result[18].

d. Stopword

Stopword removes common words that have no meaning for the text analysis process, including conjunctions, pronouns, and prepositions. This stage utilizes the NLTK library with the Indonesian version[19].

e. Stemming

Stemming minimizes the number of words by retrieving their root and deleting inflection by dropping unnecessary characters, usually a suffix[19].

f. Normalization

Text normalization is changing non-standard words into standard words and correcting spelling errors using the Indonesian language normalization dictionary dataset[20].

g. N-Gram

After the data is preprocessed, the next step is N-gram implementation. N-grams are continuous sequences of n items in a sentence. For example, "machine learning" as a bigram conveys a distinct concept that would be lost if the words were treated separately[21].

2.3. Vectorization

Before building the LDA model, a vectorization stage is required [22]. The stage after data preprocessing and implementation of N-grams is the formation of a corpus with BoW and TF-IDF Vectorization.

a. BoW

The Bag of Words text vectorization model conceptualizes documents as collections of words, ignoring grammar and the sequence in which the words appear. Each document is represented as a set of numerical vectors with a fixed length, typically corresponding to the number of unique words in the corpus. Each feature within the vector signifies the frequency of occurrence of each word.

This method is one of the reasonably simple methods of processing text data converted into vector-shaped numbers so that a computer can process it. This method only calculates the number of frequencies of word occurrences in all documents processed[23].

b. TF-IDF Vectorizer

TF-IDF is a technique used to assess the importance of words in a document relative to a collection of other documents. This technique gives higher weight to words that appear frequently in a document but rarely in others. TF-IDF is useful for balancing common words and unique words in a document.

The formula to find the weight of words with TF-IDF is:

$$Wij = tfij\, x\, ((log\, \frac{N}{n}) + 1) \qquad (1)$$

Where:

- Wij: weight of term tj against document di
- tfij: number of occurrences of word/term tj in di
- N: number of all existing documents
- n: number of documents containing word/term tj.

2.4. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a widely recognized method for topic modelling[24]. It organizes documents into multiple topics by analyzing word distributions. The algorithm assumes that documents contain various topics, each defined by its unique word distribution[10].

As a generative probabilistic model, LDA examines a collection of texts called a corpus. The core concept is that each document is viewed as a probabilistic combination of hidden topics, with each topic characterized by a specific distribution of words present within it[24]. The LDA method as a probabilistic model, as shown in Figure 2.

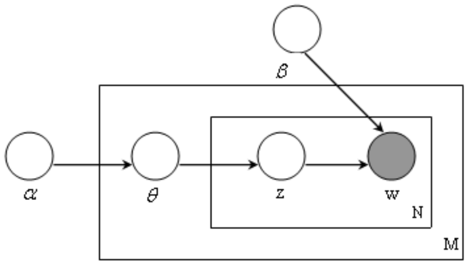


Figure 2. LDA Representation Model

The Latent Dirichlet Allocation (LDA) process can be described using the following notation, as depicted in Figure 2. Parameters α and β serve as corpus label parameters. Parameter α determines the Dirichlet distribution on the subject, indicating the pattern of the subject, higher α values suggest an increased mixture of topics in a document. Parameter β regulates the Dirichlet distribution over words, indicating the pattern of words. The higher the β value, the more words are in the topic, while the smaller the beta value, the fewer words are in the topic so that the topic contains more specific words. Variable θ represents the topic distribution at the document level (M) and indicates the proportion of each topic within the document. The higher the θ value, the more topics are in the document, while the smaller the θ value, it can be said that the document is more specific to a particular topic. Variables W and Z are word-level variables, and variable z represents the topic set in that word, while variable W represents related words for a particular topic.

This study uses LDA to identify lecturers' research topics from a collection of publications obtained through web scraping. Each lecturer's publication will be given a certain probability for each topic found, and the dominant topic of each publication can be identified.

2.5. LDA Evaluation

The evaluation metric that can be used to assess the performance of a topic model is the coherence value. In topic modelling, topic coherence measures the data quality by comparing the semantic similarity between highly repetitive words in a topic. The coherence score is a measure used to evaluate the topic modelling in which a good model will produce topics with high topic coherence scores[25].

The coherence value is obtained by calculating the similarity between topics. The coherence score is a scale from 0 to 1. The higher the score, the better the topic is considered.

2.6. Topic Visualization

LDA topic modelling results can be visualized using libraries such as pyLDAvis and Wordcloud, which provide a more intuitive view of the relationships between topics and the distribution of words within topics. This visualization provides information about existing topics and the most dominant words in each topic [26].

2.7. Cosine Similarity

The cosine similarity method measures the similarity between two objects. A formula in the vector space model algorithm determines the document's and keywords' weights. The cosine similarity method calculates the cosine value between the two vectors. The formula used for cosine similarity is as follows[4].

$$im\ (q, dj) = \frac{q,dj}{|q|x|dj|} = \frac{\sum_{i=1}^{t} \quad wiqxwij}{\sqrt{\sum_{i=1}^{t} \quad (wiq)2\ x \sqrt{\sum_{i=1}^{t} \quad (wij)2}}} \qquad (1)$$

Where:

- q = Vector query, which will be compared similarity
- d = Vector document j, which will be compared similarity
- | q | = Length of the query vector
- | d | = Document vector length j
- Wiq = Weight of the word i in the query
- Wij = Weight of the word i in document j.

3. RESULT

3.1. Data Collection

Data collection is the initial stage in research. Research data is collected through web scraping from Google Scholar and ResearchGate. The scraping process is conducted using Python programming language and the Scholarly library.

Table 1. Lecturer Research Dataset

Nama Dosen	Judul Penelitian
Abdus Salam	Deteksi Masker Menggunakan Jaringan Neural Konvolusional
Abu Salan	Sistem Monitoring Penyebaran Covid-19 Di Indonesia
Achmad Wahid	Klasifikasi Kelayakan Kredit Dengan Menggunakan Metode Naive Bayes
Aditya Nugraha	Optimasi Logistic Regression untuk Deteksi Serangan DoS pada Keamanan IoT
Adi Prihandono	Aplikasi Gamifikasi Pembelajaran Bahasa Inggris Berbasis Augmented Reality

Table 1 shows some of the lecturer research data collected. The scraping data results are 3722 entries of lecturer names and research titles stored in CSV via Pandas. After the dataset is converted, the dataset is then translated so that the research title is uniform in Indonesian. There is no specific period regarding the data taken. The initial dataset was still dirty and needed to be cleaned through several stages of data preprocessing.

3.2. Data Preprocessing Results

The raw data is then processed through the preprocessing stage. The results of the data preprocessing are data that has character uniformity (lowercase), writing that follows Indonesian language rules and no longer has special characters, sentences that have been broken down into words, do not contain general words, and have changed into basic words. This data then be changed with tf-idf vectorization into a corpus and dictionary in LDA topic modelling.

Examples of the results of pre-processing data can be seen in Table 2.

Table 2. Data Preprocessing Result

Process	Result
Raw Data	Deteksi Masker Menggunakan Jaringan Neural Konvolusional
Case Folding	deteksi masker menggunakan jaringan neural konvolusional
Filtering	deteksi masker menggunakan jaringan neural konvolusional
Tokenization	['deteksi', 'masker', 'menggunakan', 'jaringan', 'neural', 'konvolusional']
Stopword	['deteksi', 'masker', 'jaringan', 'neural', 'konvolusional']
Lemmatization	['deteksi', 'masker', 'jaringan', 'neural', 'konvolusional']
Normalization	['deteksi', 'masker', 'jaringan', 'neural', 'konvolusional']

The data preprocessing pipeline in Table 2 exemplifies a methodical strategy for the preparation of raw textual data for machine learning applications. The process initiates with Raw Data, where the unrefined input text, "Deteksi Masker Menggunakan Jaringan Neural Konvolusional," is identified as the source material. This phase is essential, as raw data frequently contains discrepancies or extraneous components that may impair analytical efficacy. The initial transformation involves Case Folding, transforming all characters to lowercase, resulting in "deteksi masker menggunakan jaringan neural konvolusional." This phase promotes uniformity and mitigates variations attributed to capitalization discrepancies.

Subsequently, Filtering is conducted to eliminate superfluous characters, such as punctuation and special symbols, thereby retaining only significant words. The filtered text remains unchanged, as it does not include any extraneous characters. Following this, Tokenization occurs, wherein the sentence is divided into distinct tokens or words, represented as ['deteksi', 'masker', 'menggunakan', 'jaringan', 'neural', 'konvolusional']. This segmentation facilitates a more granular analysis and processing of the text.

The Stopword Removal phase further optimizes the data by discarding commonly used yet semantically trivial words, such as "menggunakan." The resultant tokens are ['deteksi', 'masker', 'jaringan', 'neural', 'konvolusional']. This stage diminishes noise and concentrates the analysis on significant content. Subsequently, Lemmatization is employed to ensure that all words are condensed to their base or root form. In this instance, the tokens remain the same, as they are already in their fundamental forms. Lastly, Normalization rectifies inconsistencies by standardizing spellings or synonyms, assuring uniform representation. Again, in this case, no alterations occur, as the text is already consistent.

Each phase within this pipeline progressively diminishes noise and augments the semantic quality of the data, ensuring it is refined, succinct, and suitable for machine learning models. By rectifying inconsistencies, excluding irrelevant components, and emphasizing meaningful tokens, this preprocessing pipeline enhances the text's efficiency in analysis and modeling, ultimately leading to improved accuracy and performance.

3.3. N-gram Implementation

Dictionaries and corpus are the primary inputs for building the LDA topic model. N-grams capture the relationship between consecutive words, which can be important for understanding the meaning of text.

Topic formation is better than using unigrams when forming topics using bigrams. This is evidenced by the higher coherence score value in bigrams compared to unigrams presented in Table 3, where the highest coherence score in unigrams is only 0.72969, with the optimal number of topics being eight topics and the highest coherence score in bigrams is 0.764129 with the best number of topics of six topics. which also aligns with previous research[27].

Table 3. Comparison of Unigram and Bigram Coherence Scores

N-Gram	Num Topic	Coherence Score
Unigram	2	0.63052
	3	0.66684
	4	0.68815
	5	0.71185
	6	0.71680
	7	0.72540
	8	0.72969
Bigram	2	0.735598
	3	0.735598
	4	0.754540
	5	0.762421
	6	0.764129
	7	0.763650
	8	0.762719

Table 3 illustrates a comparative analysis of coherence scores between unigram and bigram models across varying numbers of topics. Coherence score is a critical metric for evaluating the quality of topic models, as it measures the semantic consistency of the topics generated. Higher coherence scores indicate better topic interpretability and relevance. The table evaluates both unigram (single-word tokens) and bigram (two-word combinations) approaches, showing how the number of topics impacts the coherence score.

In the unigram model, as the number of topics increases from 2 to 8, the coherence score gradually improves. Starting with a score of 0.63052 for 2 topics, the score steadily rises to 0.72969 at 8 topics. This trend suggests that adding more topics allows the unigram model to capture finer-grained distinctions between concepts, thereby enhancing semantic coherence. However, the improvement diminishes as the number of topics increases, indicating that adding too many topics might lead to redundancy or overlap.

In contrast, the bigram model consistently outperforms the unigram model across all topic numbers. For example, at 2 topics, the bigram model achieves a coherence score of 0.735598, significantly higher than the unigram's score of 0.63052. Similarly, at 8 topics, the bigram model scores 0.762719, again exceeding the unigram model. This improvement highlights the advantage of bigrams in capturing contextual relationships and compound phrases that are often critical for understanding the underlying themes in text data.

Additionally, the bigram model's coherence scores appear more stable, with minimal variation across different numbers of topics. This stability suggests that bigrams provide a more robust representation of semantic patterns, making them better suited for applications requiring high interpretability and reliability in topic modeling.

In summary, the results presented in Table 3 underscore the effectiveness of bigram models over unigram models in generating semantically coherent topics. While unigram models capture individual word distributions, bigram models leverage contextual relationships, resulting in more meaningful and interpretable topics. These findings emphasize the importance of selecting appropriate n-gram models based on the complexity and contextual richness of the textual data.

For this reason, this study built a dictionary containing a list of the most dominant combinations of 2 words (bi-gram) and their frequency of occurrence.

3.4. Topic Modeling Results

After the pre-processing stage is complete, the next stage uses LDA modelling.

Table 4. LDA Parameters

Parameters	Values
Num of Topic (K)	2 - 11
Alpha	0.001
Eta	0.01
Chunk Size	10
Random State	100
Passes	10
Update Every	1

The parameters used in this study are shown in Table 4. The number of Topics (K) determines the number of topics extracted from the document. This value determines the number of topic clusters in the model. The model will find between 2 and 11 topics, depending on the data structure and complexity of the document content.

The alpha (α) parameter controls the distribution of topics in the document. A small value such as 0.001 indicates that most documents tend to have a few dominant topics (high sparsity).

The Eta (β) parameter controls the distribution of words in topics. A small value indicates that each topic consists of only a few specific words, making the topic more focused.

Chunk Size is a parameter that determines the number of documents processed in one iteration. Per iteration, ten documents will be processed. This parameter affects the efficiency and memory usage during training.

The random state is a parameter to set the seed value so that the model results are reproducible. With a random state of 100, the model will produce the same results if rerun with the same data and parameters.

Passes indicate the number of iterations the algorithm will pass through the entire corpus of documents. The model will learn the entire corpus 10 times, helping model convergence but increasing computation time.

The last parameter determines the frequency of updating topic parameters in an iteration. With an update every iteration, the model parameters will be updated more frequently, which can improve accuracy but may be slower. This configuration allows the model to extract topics at a high level of granularity, focusing on specific words and well-defined topics.

The optimal number of topics is determined by utilizing the coherence score test. This study conducted the coherence value test by comparing the corpus processed with Bag of Words and TF-IDF.

The coherence score results for each data can be seen in the following table:

Table 5. Comparison of TF-IDF and BoW Coherence Scores

Vectorization	Num Topic	Coherence Score
TF-IDF	2	0.735598
	3	0.735598
	4	0.754540
	5	0.762421
	6	0.764129
	7	0.763650
	8	0.762719
Bag of Word	2	0.563741
	3	0.495917
	4	0.447588
	5	0.434323
	6	0.407608
	7	0.417091
	8	0.399785

Table 5 compares the coherence scores of topic models generated using two popular vectorization techniques: Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW). Coherence scores serve as a metric for evaluating the quality and interpretability of topics produced by these vectorization methods, with higher scores indicating better semantic consistency.

The results demonstrate that the TF-IDF vectorization consistently outperforms the BoW approach across all topic numbers. For TF-IDF, coherence scores begin at 0.735598 for 2 topics and gradually increase, peaking at 0.764129 for 6 topics. This improvement highlights TF-IDF's ability to assign appropriate weights to terms based on their importance, effectively capturing meaningful patterns and relationships within the data. The slight decline in coherence scores beyond 6 topics suggests that adding more topics may lead to redundancy or semantic overlap.

In contrast, the BoW approach produces significantly lower coherence scores, starting at 0.563741 for 2 topics and progressively declining to 0.399785 at 8 topics. Unlike TF-IDF, which accounts for term importance across documents, BoW merely represents the frequency of terms, disregarding contextual relevance. This limitation likely contributes to its inability to generate coherent topics, especially as the number of topics increases. The downward trend in scores suggests that BoW struggles to maintain semantic consistency when dividing the text data into a larger number of topics.

The comparison underscores the superior performance of TF-IDF over BoW in topic modeling. While BoW is simpler and computationally less expensive, its lack of weighting mechanisms makes it less effective in capturing nuanced patterns in text.

TF-IDF, on the other hand, leverages term importance across documents, resulting in more coherent and interpretable topics.

In conclusion, the findings in Table 5 highlight the critical role of vectorization techniques in determining topic model quality. For datasets requiring a higher level of semantic understanding and interpretability, TF-IDF is the preferred choice over BoW. These insights emphasize the importance of selecting an appropriate preprocessing and vectorization strategy to achieve optimal results in topic modeling tasks.

The number of topics with the highest coherence value is the number of topics (K) of six, with a coherence value of 0.764. The high coherence value indicates that the LDA model is optimal, so this study sets $K = 6$ as the number of topics resulting from LDA modeling.

Based on the experiment results, TF-IDF data obtained a coherence value with an average value of 0.73 and above. Hence, the words produced with the TF-IDF extraction feature are more accessible to interpret by human language. From the results of this experiment, TF-IDF became the best extraction feature for topic modelling using the Latent Dirichlet Allocation method.

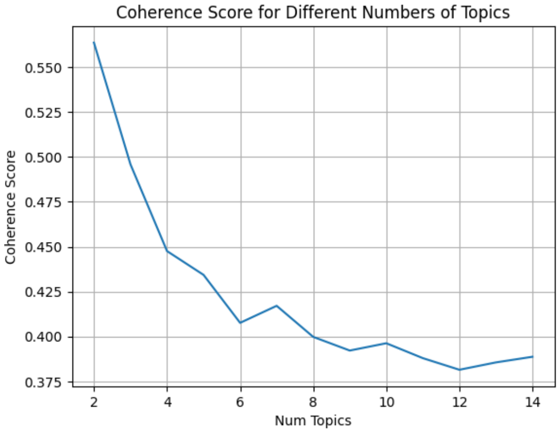


Figure 3. Coherence Score BoW Vectorizer

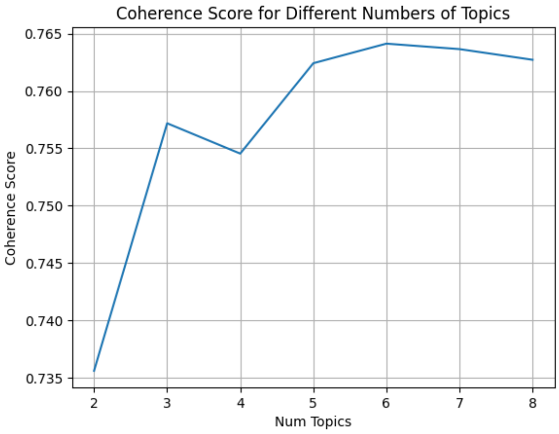


Figure 4. Coherence Score TF-IDF Vectorizer

Table 6. Topic terms of six topic

Topic	Topic terms
Topic 1	0.006*"gambar" + 0.006*"citra" + 0.005*"klasifikasi" + 0.005*"bit" + 0.004*"algoritma" + 0.004*"lsb" + 0.004*"model" + 0.004*"steganografi" + 0.003*"hybrid" + 0.003*"database"'
Topic 2	0.005*"pelatihan" + 0.005*"sistem" + 0.005*"aplikasi" + 0.004*"model" + 0.004*"implementasi" + 0.004*"klasifikasi" + 0.004*"enkripsi" + 0.004*"bahasa" + 0.003*"file" + 0.003*"teknik"
Topic 3	0.004*"sistem" + 0.004*"aplikasi" + 0.003*"naive_bayes" + 0.003*"analisis" + 0.003*"oky_dwi" + 0.003*"sistem_pakar" + 0.003*"perangkat_lunak" + 0.003*"perancangan" + 0.003*"teknologi_informasi" + 0.003*"laporan"
Topic 4	0.008*"klasifikasi" + 0.005*"fitur" + 0.005*"kecerdasan_buatan" + 0.004*"bayes" + 0.004*"pendekatan" + 0.004*"web" + 0.004*"algoritma" + 0.003*"glcm" + 0.003*"analisis" + 0.003*"jaringan_syaraf"
Topic 5	0.005*"klasifikasi" + 0.005*"game" + 0.004*"algoritma" + 0.004*"perilaku" + 0.003*"nearest_neighbor" + 0.003*"penerapan" + 0.003*"model" + 0.003*"convolutional_neural" + 0.003*"digital" + 0.003*"pembelajaran"
Topic 6	0.004*"peningkatan" + 0.003*"learning" + 0.003*"investigasi" + 0.003*"algoritma" + 0.003*"pengenalan" + 0.003*"aplikasi" + 0.003*"sistem" + 0.003*"deteksi" + 0.003*"algoritma_nearest" + 0.003*"medium"

Topic modeling using LDA produces 6 topic groups. Each topic group contains terms and their weights. Each term in a topic group has a weight that represents the probability of the term appearing in the entire dataset document[13].

Table 6 presents the terms associated with six identified topics, along with their corresponding weights, generated using topic modeling techniques such as Latent Dirichlet Allocation (LDA). Each topic is defined by a set of terms and their weights, which represent the importance or contribution of the term within the topic. This table serves to interpret and label the thematic structure of the dataset based on the most prominent terms in each topic.

Topic 1 is characterized by terms such as "gambar" (image), "citra" (image data), "klasifikasi" (classification), and "steganografi" (steganography). These terms indicate that the topic revolves around image processing, classification, and data hiding techniques, commonly applied in fields like digital forensics and computer vision.

Topic 2 features terms like "pelatihan" (training), "aplikasi" (application), and "enkripsi" (encryption), suggesting a focus on software systems, encryption methods, and training or implementation processes. This topic likely represents discussions on system development and security.

Topic 3 includes terms such as "naive_bayes", "sistem_pakar" (expert systems), and "analisis" (analysis). These terms point to themes related to artificial intelligence (AI) methods, particularly statistical classification models and expert systems, emphasizing analytical processes in AI applications.

Topic 4 is dominated by terms such as "klasifikasi" (classification), "kecerdasan_buatan" (artificial intelligence), and "jaringan_syaraf" (neural networks). This topic centers on machine learning and AI techniques, particularly classification and neural network algorithms, reflecting core areas in data science and intelligent systems.

Topic 5 contains terms like "nearest_neighbor", "convolutional_neural", and "pembelajaran" (learning). These suggest a focus on advanced machine learning methods, particularly neural network architectures and algorithms like k-Nearest Neighbor and Convolutional Neural Networks, which are commonly used in pattern recognition and deep learning.

Topic 6 features terms such as "peningkatan" (improvement), "investigasi" (investigation), and "deteksi" (detection). This indicates a focus on applications aimed at improving systems, detecting anomalies, or performing investigative analyses, which could apply to diverse fields such as cybersecurity or fraud detection.

In conclusion, the six topics provide a thematic overview of the dataset, highlighting prominent areas such as image processing, encryption, AI methodologies, and machine learning applications. The weights associated with each term emphasize their relevance, aiding in the interpretation of each topic's focus. This table underscores the effectiveness of topic modeling in extracting meaningful and interpretable themes from text data, providing valuable insights into the dataset's underlying structure.

3.5. Wordcloud Topic Visualization

Wordcloud is an image containing a collection of words that often appear in a text. The more words used, the larger the size of the words in the image[28].

Figure 5 shows that the most frequently occurring words for topic 1 are gambar, citra, klasifikasi, and bit.

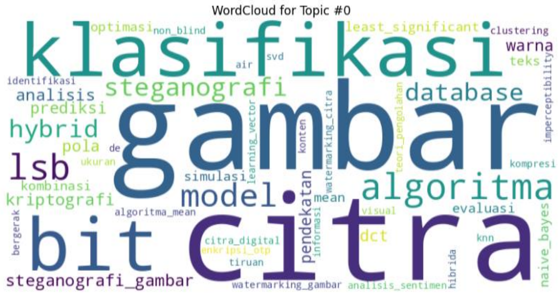


Figure 5. Wordcloud for Topic 1

Figure 6 shows that the most frequently occurring word for topic 2 are pelatihan, sistem, aplikasi and model.

Figure 7 shows that the most frequently occurring words for topic 3 are sistem, aplikasi, naive_bayes, and analisis.

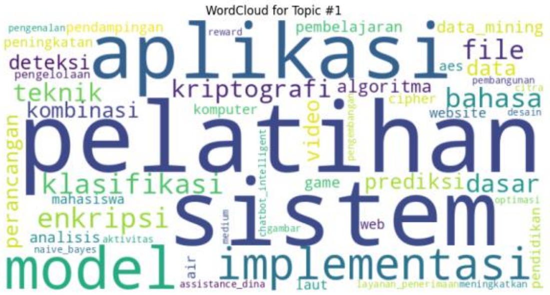


Figure 6. Wordcloud for Topic 2

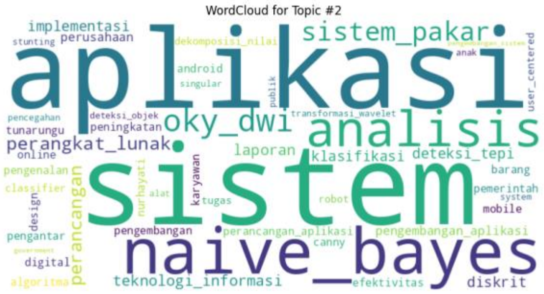


Figure 7. Wordcloud for Topic 3

Figure 8 shows that the most frequently occurring word for topic 4 are klasifikasi, fitur, and kecerdasan_buatan.

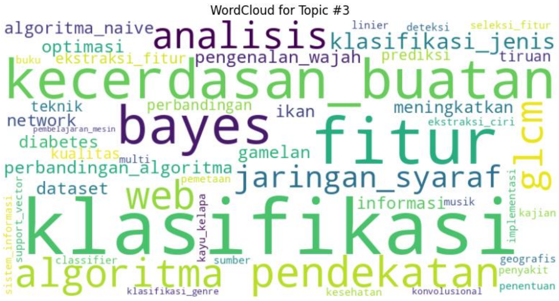


Figure 8. Wordcloud for Topic 4

Figure 9 shows that the most frequently occurring word for topic 5 are klasifikasi, game, and algoritma.

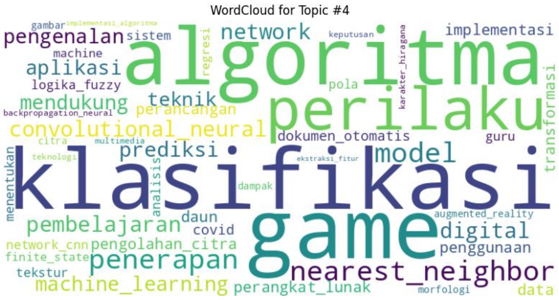


Figure 9. Wordcloud for Topic 5

Figure 10 shows that the most frequently occurring word for topic 6 are peningkatan, learning and investigasi.

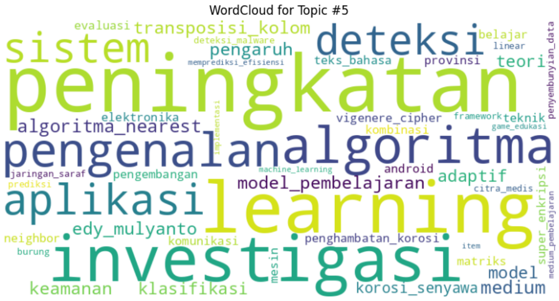


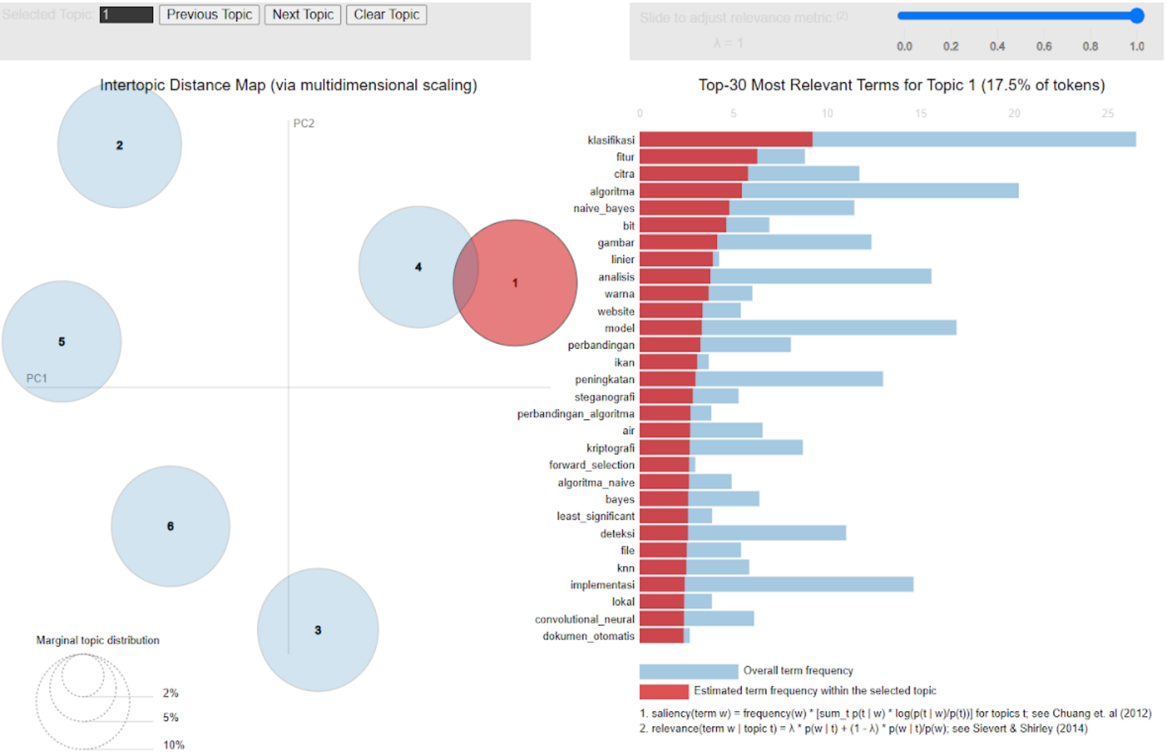
Figure 10. Wordcloud for Topic 6

The dominant word collection requires a process of human interpretation of meaning to provide information by comparing the results of topic modelling with the original data.

3.6. PyLDAvis Topic Visualization

Visualization of LDA implementation using graphs and bar charts utilizing the PyLDAvis tool in Python[29]. PyLDAvis is used to visualize the clusters formed based on the data[30]. This intertopic distance maps are used to determine the distance between topics and the relationship between one topic and another[31].

The pyLDAVis approach allows for a deeper inspection of the terms most highly associated under each individual topic yielding "information" which is more optimal for interpretation[32].



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 5. Topic Visualization using PyLDAvis

The topic model is illustrated in the left panel, where circles represent topics in a two-dimensional space. The positions of these circles are determined by measuring the separation between the topics and mapping that separation onto a two-dimensional plane using a multidimensional scaling approach.

The visualization presents an analysis of topic modeling results through two key components: the Intertopic Distance Map and the Top-30 Most Relevant Terms for Topic 1. The Intertopic Distance Map, shown on the left, uses multidimensional scaling to illustrate the semantic relationships between six identified topics. Each topic is represented as a circle, where the size corresponds to the proportion of tokens or documents associated with the topic. The positioning of the circles indicates the semantic similarity or distinction between topics. For instance, Topic 1, which accounts for 17.5% of the tokens (as shown in the marginal topic distribution), slightly overlaps with Topic 4, suggesting some shared thematic content. In contrast, topics such as 2, 3, 5, and 6 are positioned farther apart, reflecting their distinct thematic boundaries.

On the right, the Top-30 Most Relevant Terms for Topic 1 are displayed. These terms are ranked based on their relevance, calculated using a metric (λ) that balances their frequency within the topic and across the entire dataset. With λ set to 1, the chart emphasizes terms that are highly frequent within Topic 1.

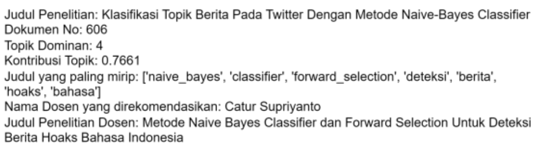
Figure 5 shows the frequency of words related to a specific topic can be compared to their overall frequency in the entire corpus. The most prominent terms include "klasifikasi" (classification), "fitur" (feature), and "citra" (image), indicating that Topic 1 focuses on themes related to classification, image processing, and feature extraction. Other significant terms, such as "algoritma" (algorithm), "naive_bayes", and "bit", further suggest a strong emphasis on machine learning and technical methodologies in the context of image classification. The bar graph shows how much of a specific key is located in a topic, indicating the key “forward selection” as it is found in almost no other topic , as represented by the red almost covering the entire bar.

Additionally, LDAvis provides a unique capability to examine how frequently a word appears across different topics, offering insights into its distribution and significance. The right panel displays a horizontal bar chart, where the bars represent the most relevant terms for a selected topic. These bars are divided into two parts, the red portion indicates the estimated frequency of the term within the chosen topic, while the blue portion shows its overall frequency across the entire corpus. This visual differentiation allows users to easily identify terms that are both specific to the topic and widely used across multiple topics, aiding in the interpretation of the topic's core themes and distinguishing features.

The combination of these two components provides a comprehensive understanding of the dataset's thematic structure. The Intertopic Distance Map highlights the global relationships and separations between topics, with Topic 1 emerging as dominant and slightly overlapping with related topics. Meanwhile, the detailed analysis of relevant terms reveals the specific focus of Topic 1 on technical applications in artificial intelligence and computer vision. This visualization effectively supports the interpretation and labeling of topics, providing valuable insights into the thematic distribution and semantic coherence within the dataset.

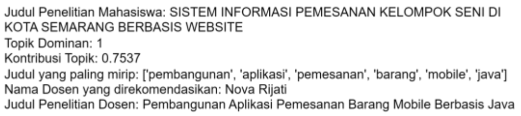
3.7. Cosine Similarity

The cosine similarity calculations were performed on the dataset to determine the most relevant thesis titles for each student input. The test resulted in the following output for sample input



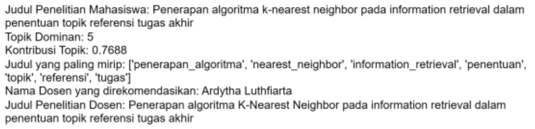
```
Judul Penelitian: Klasifikasi Topik Berita Pada Twitter Dengan Metode Naive-Bayes Classifier
Dokumen No: 606
Topik Dominan: 4
Kontribusi Topik: 0.7661
Judul yang paling mirip: ['naive_bayes', 'classifier', 'forward_selection', 'deteksi', 'berita',
'hoaks', 'bahasa']
Nama Dosen yang direkomendasikan: Catur Supriyanto
Judul Penelitian Dosen: Metode Naive Bayes Classifier dan Forward Selection Untuk Deteksi
Berita Hoaks Bahasa Indonesia
```

Figure 12. Cosine Similarity for Sample Input 1



```
Judul Penelitian Mahasiswa: SISTEM INFORMASI PEMESANAN KELOMPOK SENI DI
KOTA SEMARANG BERBASIS WEBSITE
Topik Dominan: 1
Kontribusi Topik: 0.7537
Judul yang paling mirip: ['pembangunan', 'aplikasi', 'pemesanan', 'barang', 'mobile', 'java']
Nama Dosen yang direkomendasikan: Nova Rijati
Judul Penelitian Dosen: Pembangunan Aplikasi Pemesanan Barang Mobile Berbasis Java
```

Figure 13. Cosine Similarity for Sample Input 2



```
Judul Penelitian Mahasiswa: Penerapan algoritma k-nearest neighbor pada information retrieval dalam
penentuan topik referensi tugas akhir
Topik Dominan: 5
Kontribusi Topik: 0.7688
Judul yang paling mirip: ['penerapan_algoritma', 'nearest_neighbor', 'information_retrieval', 'penentuan',
'topik', 'referensi', 'tugas']
Nama Dosen yang direkomendasikan: Ardytha Luthfiarta
Judul Penelitian Dosen: Penerapan algoritma K-Nearest Neighbor pada information retrieval dalam
penentuan topik referensi tugas akhir
```

Figure 14. Cosine Similarity for Sample Input 3

The cosine similarity is calculated between a student's thesis title and a dataset of research publications to recommend a suitable supervisor. It begins by loading and merging two datasets, creating a combined column with keywords and the title publication for each document. TF-IDF vectorization converts the documents and thesis title into numerical representations, emphasizing the importance of term. Cosine similarity is then calculated between the input title and each document. Finally, the documents are sorted by similarity score and topic contribution, with the highest-ranking document displayed as the recommended supervisor match, including details like the topic and advisor's name.

The highest cosine similarity score for the sample, 0.7688, indicates a strong relevance between the student's input title and the recommended supervisor's research. This validates the effectiveness of using cosine similarity for thesis supervisor recommendations, helping streamline the process for students in selecting supervisors whose research aligns closely with their thesis topics.

4. DISCUSSION

Previous research conducted by Roma Gabe Dalimunthe and Raissa Amanda Putri (2024) on these topic modeling of computer science study programs using the Latent Dirichlet Allocation Algorithm produced the highest coherence score of 0.4011 with a total of seven topics.

Another previous study, namely Analysis and Application of Topic Modeling in Final Assignments of Computer Science Students Using the Latent Dirichlet Allocation (LDA) Method, produced a coherence score of 0.53448 after hyperparameter tuning the coherence score to 0.617789 [33].

Several methods have been employed in previous research for thesis supervisor recommendation systems, including BM25 (Best Matching 25)[2], TF-IDF with Cosine Similarity[4][13], Content-Based Filtering[3][34] . BM25 focuses on keyword-based similarity and ranking between the student’s thesis title and supervisor publications. Although effective in certain cases, it lacks semantic understanding as it treats words independently, which can reduce accuracy for context-heavy topics. TF-IDF with Cosine Similarity method ranks the importance of terms in a document relative to the corpus and calculates similarity using cosine metrics. While it provides a solid baseline, it does not consider word relationships (bigrams, trigrams) or the latent topics that LDA identifies. Content-based filtering method matches based on explicit content features but often relies heavily on manual input, such as tagging topics, which is time-consuming and less automated.

In comparison, the proposed approach using LDA with cosine similarity has several advantages. LDA identifies hidden or latent topics within the research publications, providing a more nuanced understanding of the content. The system extracts and processes publications without requiring manual tagging or input, saving significant time and effort.

This study successfully models lecturers' research topics using Latent Dirichlet Allocation (LDA) with ResearchGate and Google Scholar publication data. By combining bigram-based TF-IDF with cosine similarity, the method ensures better contextual matching. The modeling results show six main topics with the highest coherence score of 0.764, indicating the quality and relevance of the topics identified from the lecturers' research. This coherence value is in an acceptable range, although there is still room for improvement, such as further exploring LDA parameters.

The comparison process between student thesis titles and lecturers' research topics using the cosine similarity method also shows a strong relationship between the thesis themes submitted by students and existing research topics. This method has proven effective in measuring semantic similarity between words that appear in the topic and thesis title. Students with relevant topics can more easily find suitable supervisors based on the highest similarity value.

One area for improvement in this study is the limited amount of publication data obtained from the ResearchGate and Google Scholar platforms. In some cases, the model may not optimally identify lecturers with limited publications. Therefore, further research can explore using additional data sources or increasing data coverage from other academic platforms.

Overall, this study's results indicate that applying the LDA and cosine similarity methods can provide an efficient solution for selecting a thesis supervisor appropriate to the student's research topic. with this method, students can save time and energy in finding relevant supervisors, while universities can also increase the efficiency of managing the thesis supervision process.

5. CONCLUSION

This study shows that modeling lecturer research topics using the LDA method and cosine similarity calculations can be implemented effectively to assist students in choosing a suitable thesis supervisor. By collecting publication data from academic platforms such as ResearchGate and Google Scholar, this study successfully identified six main topics with the highest coherence score of 0.764, which indicates the quality of the resulting topics. Comparing student thesis titles with lecturer research topics using the cosine similarity method provides more explicit guidance for students in finding supervisors with relevant expertise. This provides a practical solution to the manual thesis supervisor selection process by offering a more efficient data-based and technology-based approach. The success of this study also opens up opportunities for further development in terms of increasing data coverage and refining the methods used, which can make it easier for students and improve the quality of thesis supervision management in the academic environment.

REFERENCES

[1] M. C. Wijanto, R. Rachmadiany, and O. Karnalim, “Thesis Supervisor Recommendation with Representative Content and Information Retrieval,” *Journal of Information Systems Engineering and Business Intelligence*, vol. 6, no. 2, 2020, doi: 10.20473/jisebi.6.2.143-150.

[2] A. A. B. Arisetiawan, Indriati, and D. E. Ratnawati, “Sistem Rekomendasi Dosen Pembimbing Berdasarkan Dokumen Judul Skripsi di Bidang Komputasi Cerdas Menggunakan Metode BM25,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 6, 2019.

[3] Anis Budiono and Sri Eniyati, “Sistem Rekomendasi Dosen Pembimbing Tugas Akhir Menggunakan Content Based Filtering,” *Jurnal Elektronika Dan Komputer*, vol. 16, no. 1, 2023.

[4] R. Rismanto, A. R. Syulistyo, and B. P. C. Agusta, “Research supervisor recommendation system based on topic conformity,” *International Journal of Modern Education and Computer Science*, vol. 12, no. 1, 2020, doi: 10.5815/ijmecs.2020.01.04.

[5] A. Nurlayli and Moch. A. Nasichuddin, “TOPIK MODELING PENELITIAN DOSEN JPTEI UNY PADA GOOGLE SCHOLAR MENGGUNAKAN LATENT DIRICHLET ALLOCATION,” *Elinvo (Electronics, Informatics, and Vocational Education)*, vol. 4, no. 2, 2019, doi: 10.21831/elinvo.v4i2.28254.

[6] F. T. Anggraeny, I. Y. Purbasari, and E. F. Wulandari, “Undergraduate Thesis Supervisor Recommendation Based On Text Similarity,” *Darmajaya*, 2019.

[7] H. Jelodar *et al.*, “Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey,” *Multimed Tools Appl*, vol. 78, no. 11, 2019, doi: 10.1007/s11042-018-6894-4.

[8] L. George and P. Sumathy, “An integrated

clustering and BERT framework for improved topic modeling," *International Journal of Information Technology (Singapore)*, vol. 15, no. 4, 2023, doi: 10.1007/s41870-023-01268-w.

[9] I. Vayansky and S. A. P. Kumar, "A review of topic modeling methods," *Inf Syst*, vol. 94, 2020, doi: 10.1016/j.is.2020.101582.

[10] D. Maulidiya, "Topic Modelling using Latent Dirichlet Allocation (LDA) to Investigate the Latent Topics of Mathematical Creative Thinking Research in Indonesia," *Journal of Intelligent Computing & Health Informatics*, vol. 3, no. 2, 2023, doi: 10.26714/jichi.v3i2.11428.

[11] E. S. Negara and D. Triadi, "Topic modeling using latent dirichlet allocation (LDA) on twitter data with Indonesia keyword," *Bulletin of Social Informatics Theory and Application*, vol. 5, no. 2, 2022, doi: 10.31763/businta.v5i2.455.

[12] S. Karmila and V. I. Ardianti, "METODE LATENT DIRICHLET ALLOCATION UNTUK MENENTUKAN TOPIK TEKS SUATU BERITA," *Jurnal Informatika dan Komputasi: Media Bahasan, Analisa dan Aplikasi*, vol. 16, no. 01, 2022, doi: 10.56956/jiki.v16i01.100.

[13] H. Hairani and M. Mujahid, "Recommendations of Thesis Supervisor using the Cosine Similarity Method," *SISTEMASI*, vol. 11, no. 3, 2022, doi: 10.32520/stmsi.v11i3.2003.

[14] S. Dami and F. Madadi, "Recommender System Using LDA Topic Modeling Approach," *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4173345.

[15] S. A. Nugroho, F. A. Bachtiar, and R. C. Wihandika, "ASPECT EXTRACTION IN E-COMMERCE USING LATENT DIRICHLET ALLOCATION (LDA) WITH TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)," *Jurnal Ilmiah Kursor*, vol. 11, no. 2, 2022, doi: 10.21107/kursor.v11i2.247.

[16] S. H. Mohammed and S. Al-Augby, "LSA & LDA topic modeling classification: Comparison study on E-books," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, 2020, doi: 10.11591/ijeecs.v19.i1.pp353-362.

[17] Y. Matira and I. Setiawan, "Pemodelan Topik pada Judul Berita Online Detikcom Menggunakan Latent Dirichlet Allocation," *Estimasi: Journal of Statistics and Its Application*, vol. 4, no. 1, 2023.

[18] O. Kononova, T. He, H. Huo, A. Trewartha, E. A. Olivetti, and G. Ceder, "Opportunities and challenges of text mining in aterials research," 2021. doi: 10.1016/j.isci.2021.102155.

[19] D. L. C. Pardede and M. A. I. Waskita, "ANALISIS PEMODELAN TOPIK UNTUK ULASAN TENTANG PEDULI LINDUNGI," *Jurnal Ilmiah Informatika Komputer*, vol. 28, no. 1, 2023, doi: 10.35760/ik.2023.v28i1.7925.

[20] D. R. Sari, Y. Yusra, M. Fikry, F. Yanto, and F. Insani, "Klasifikasi Sentimen Masyarakat di Twitter Terhadap Ancaman Resesi Ekonomi 2023 dengan Metode Naïve Bayes Classifier," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 4, 2023, doi: 10.30865/json.v4i4.6276.

[21] M. R. Fahlevvi and A. SN, "Topic Modeling on Online News.Portal Using Latent Dirichlet Allocation (LDA)," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 16, no. 4, 2022, doi: 10.22146/ijccs.74383.

[22] N. Novarian, S. Khomsah, and A. B. Arifa, "Topic Modeling Tugas Akhir Mahasiswa Fakultas Informatika Institut Teknologi Telkom Purwokerto Menggunakan Metode Latent Dirichlet Allocation," *LEDGER: Journal Informatic and Information Technology* , vol. 2, no. 1, 2023.

[23] K. Tri Putra, M. Amin Hariyadi, and C. Crysdian, "PERBANDINGAN FEATURE EXTRACTION TF-IDF DAN BOW UNTUK ANALISIS SENTIMEN BERBASIS SVM," *Jurnal Cahaya MAndalika*, 2023.

[24] S. Zhou, P. Kan, Q. Huang, and J. Silbernagel, "A guided latent Dirichlet allocation approach to investigate real-time latent topics of Twitter data during Hurricane Laura," *J Inf Sci*, vol. 49, no. 2, 2023, doi: 10.1177/01655515211007724.

[25] R. P. Fauzie Afidh and Z. A. Hasibuan, "Indonesia's News Topic Discussion about Covid-19 Outbreak using Latent Dirichlet Allocation," in *2020 5th International Conference on Informatics and Computing, ICIC 2020*, 2020. doi: 10.1109/ICIC50835.2020.9288596.

[26] V. Alpiana, A. Salam, F. Alzami, I. Rizqa, and D. Aqmala, "Analisis Topic-Modelling Menggunakan Latent Dirichlet Allocation (LDA) Pada Ulasan Sosial Media Youtube," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 1, 2024, doi: 10.30865/mib.v8i1.7127.

[27] R. P. F. Afidh and Syahrial, "Pemodelan Topik Menggunakan n-Gram dan Non-negative Matrix Factorization," *Jurnal*

Informasi dan Teknologi, 2023, doi: 10.60083/jidt.v5i1.385.

[28] A. I. Alfanzar, K. Khalid, and I. S. Rozas, “TOPIC MODELLING SKRIPSI MENGGUNAKAN METODE LATENT DIRICLHET ALLOCATION,” *JSiI (Jurnal Sistem Informasi)*, vol. 7, no. 1, 2020, doi: 10.30656/jsii.v7i1.2036.

[29] O. B. Sezer and A. M. Ozbayoglu, “Financial trading model with stock bar chart image time series with deep convolutional neural networks,” *Intelligent Automation and Soft Computing*, vol. 26, no. 2, 2020, doi: 10.31209/2018.100000065.

[30] H. Oktafiandi, “Implementasi LDA untuk Pengelompokan Topik Twitter Bertagar# Mypertamina,” *Jurnal Ekonomi dan Teknik Informatika*, vol. 11, no. 1, 2023.

[31] C. Naury, D. H. Fudholi, and A. F. Hidayatullah, “Topic Modelling pada Sentimen Terhadap Headline Berita Online Berbahasa Indonesia Menggunakan LDA dan LSTM,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 1, 2021, doi: 10.30865/mib.v5i1.2556.

[32] A. Gottfried, C. Hartmann, and D. Yates, “Mining open government data for business intelligence using data visualization: A two-industry case study,” *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 16, no. 4, 2021, doi: 10.3390/JTAER16040059.

[33] T. Titiana and D. H. Bangkalang, “ANALISIS DAN PENERAPAN TOPIC MODELING PADA JUDUL TUGAS AKHIR MAHASISWA MENGGUNAKAN METODE LATENT DIRICHLET ALLOCATION (LDA),” *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 4, 2023, doi: 10.29100/jipi.v8i4.4254.

[34] R. Widayanti, M. H. R. Chakim, C. Lukita, U. Rahardja, and N. Lutfiani, “Improving Recommender Systems using Hybrid Techniques of Collaborative Filtering and Content-Based Filtering,” *Journal of Applied Data Sciences*, vol. 4, no. 3, 2023, doi: 10.47738/jads.v4i3.115.