Journal of Informatics and Vocational Education (JOIVE)
Vol.9, No.1, January 2026, pp. 16~25
ISSN: 2746-7813

Automatic Indexing of Digital Books using RAKE and Word2Vec

Arhan Windu Rizki Putra Budianto[1], Ulla Delfana Rosiani [2], Vit Zuraida[3], Rizki Putri Ramadhani[4], Mohammad Alfarizi Abdullah[5]
[1,2,3,4,5] Department of Information Technology, State Polytechnic of Malang, Indonesia

Article Info

Article history:

Received Dec 19, 2025
Revised Jan 19, 2026
Accepted Jan 26, 2026

Corresponding Author:

Arhan Windu Rizki Putra Budianto
Department of Information Technology, State Polytechnic of Malang
Jl. Soekarno Hatta No.9, Malang, East Java, Indonesia, 65141
Email: bintang135.bp@gmail.com

ABSTRACT

Manual indexing of digital books is time-consuming and prone to inconsistency. To address this, this study developed an automatic indexing system using RAKE (Rapid Automatic Keyword Extraction) method and Word2Vec. The system accepts PDF files as input, performs text preprocessing, and extracts key phrases using RAKE. These phrases are subsequently filtered based on semantic relevance to the specified topic using an Indonesian-language Word2Vec model. Users can manually add phrases and select relevant ones to be included in the final index. The resulting index includes phrases, page numbers, and relevance scores, which are inserted as an additional page at the end of the PDF document. Evaluation was conducted by comparing the system-generated index with the author's manual index using precision, recall, and cosine similarity metrics. The results indicate that although precision and recall were very low, a cosine similarity score of 0.69 suggests a semantic similarity between the system output and the author's index.

Keywords: Automatic indexing, digital book, RAKE, Word2Vec, PDF

1. INTRODUCTION

An index is an essential element of a book that serves as a guide for readers to quickly and efficiently locate specific information (Supriadi & Fitriyani, 2021; Laila, 2020). The process of creating a well-structured index requires careful analysis of the book's content, including the identification of relevant keywords. However, manual index preparation is often time-consuming and demands a deep understanding of the book's subject. For books with numerous pages or complex topics, manual indexing may result in inconsistencies in term selection or page referencing. The lack of efficient supporting tools also makes the process a burden for authors or editors. Therefore, a more systematic and automated approach is needed to ensure faster, more accurate, and consistent index generation without compromising content quality. Such automation is increasingly relevant in the digital era, where the number of digital publications continues to rise and efficient navigation is highly demanded.

Recent advancements in Natural Language Processing (NLP) enable the automated analysis of text, including the extraction of keywords from documents such as articles, news, and digital books (Firoozeh et al., 2020; Nomoto, 2023). One effective keyword extraction approach is the Rapid Automatic Keyword Extraction (RAKE) algorithm (Benita & Baizal, 2022; Baruni & Sathiaseelan, 2020). This method can identify important words or phrases within a document based on their occurrence and co-occurrence patterns. Previous research (Benita & Baizal, 2022) demonstrated that RAKE is capable of generating relevant and efficient automatic indexes for various document types. RAKE was chosen in this study due to its simplicity and efficiency in extracting key phrases without requiring external corpora or model training. Compared to methods such as TF-IDF, which relies solely on word frequency, or TextRank, which uses graph-based complexing, RAKE produces meaningful multi-word phrases using stopwords and punctuation as delimiters (Baruni & Sathiaseelan, 2020; Guda et al., 2023). This makes it particularly suitable for processing long documents such as digital books.

Furthermore, Word2Vec, a machine learning-based algorithm, can represent semantic relationships between words in vector space (Yilmaz & Toklu, 2020; Di Gennaro, Buonanno, & Palmieri, 2021). This approach allows the identification of semantically similar words within a document. Although Word2Vec has some limitations in capturing syntactic relations, it is effective in modeling semantic similarity between terms. By leveraging word embeddings, Word2Vec enhances the contextual accuracy of the generated index. This study aims to develop an automatic indexing

system for Indonesian-language digital books using a combination of RAKE and Word2Vec. The proposed system automates keyword extraction, identifies contextually relevant terms, and generates an index page appended to the processed PDF. The novelty of this research lies in integrating RAKE and Word2Vec for semantic-based indexing in the Indonesian language, as well as evaluating its performance using precision, recall, and cosine similarity metrics.

2. RESEARCH METHOD

2.1. RESEARCH FLOW

This research aimed to develop an automatic indexing system for digital books utilizing the RAKE (Rapid Automatic Keyword Extraction) and Word2Vec methods. The system automatically identifies important phrases from the text, determines their relevance using semantic similarity, and generates an index page that is appended to the end of the original PDF book. The research flow consists of five main stages: (1) data collection, (2) data preprocessing, (3) keyword extraction using RAKE, (4) semantic similarity analysis using Word2Vec, and (5) system evaluation using Precision, Recall, and Cosine Similarity.

1. Data Collection
   The input data used in this study were digital books in PDF format written in Indonesian. A reference index file (ground truth) was also used to evaluate the accuracy of the automatically generated index.
2. Data Preprocessing
   Before applying the extraction methods, the text data from the PDF was processed through several stages:
   - PDF Conversion: Extracting text from PDF files using the PyMuPDF library.
   - Normalization: Converting all characters to lowercase, removing punctuation and special symbols.
   - Tokenization: Splitting sentences into individual tokens (words).
   - Stopword Removal: Removing non-essential words using the Indonesian stopword list from Sastrawi.

2.2. KEYWORD EXTRACTION USING RAKE

RAKE (Rapid Automatic Keyword Extraction) was used to extract candidate key phrases from the preprocessed text. RAKE works by identifying word co-occurrences and calculating the degree-to-frequency ratio for each word, which is used to score phrases. The RAKE scoring equation is expressed as:

$$Score(P) = \sum_{w \epsilon P} \frac{\text{degree(w)}}{\text{frequency(w)}} \quad (1)$$

where Score(P) represents the score of phrase P, degree(w) is the number of co-occurring words with w, and frequency(w) is the number of times w appears in the document.

The system extracts the top 100 phrases ranked by RAKE score, which include phrases consisting of 1-3 words.

2.3. WORD2VEC SEMANTIC SIMILARITY

To enhance the relevance of extracted keywords, the Word2Vec model (specifically FastText pretrained model cc.id.300.vec) was applied to measure the semantic similarity between each keyword and the input topic provided by the user. Each word or phrase is represented as a 300-dimensional vector, and the average of its word vectors is used to represent multiword phrases. The similarity is computed using the cosine similarity formula:

$$Cosine\ Similarity = \frac{A \cdot B}{||A|||B||} \quad (2)$$

where A and B are vector representations of the keyword and topic, respectively.

Keywords with similarity values above a specified threshold are considered relevant.

2.4. SYSTEM DESIGN AND ARCHITECTURE

The system was designed as a web-based application using the Flask framework. Figure 1 illustrates the use case diagrams of the system, which involve one main actor: the Author. The author can upload PDF files, input topics, view automatic indexing results, manually add phrases, and evaluate the accuracy of the generated index.

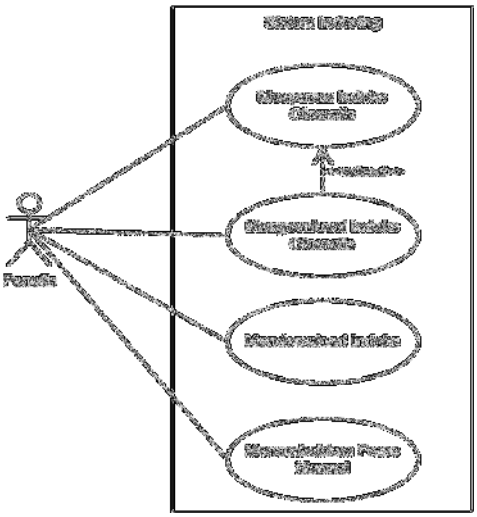
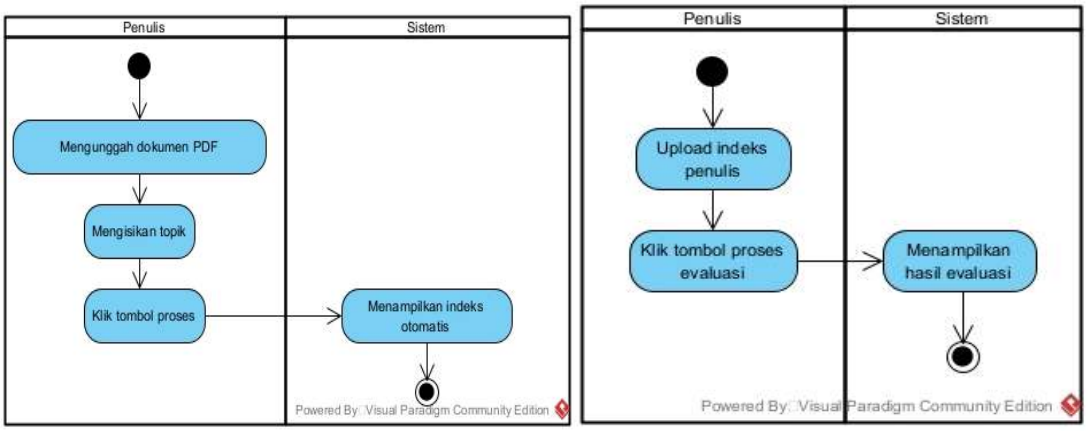


Figure 1. Use Case Diagram of the Automatic Indexing System

The architecture consists of three main components:

1. Frontend (User Interface) – built using HTML, CSS, and Bootstrap for uploading files, displaying keywords, and showing evaluation results.
2. Backend Processing – handles keyword extraction (RAKE), semantic analysis (Word2Vec), and evaluation computations.
3. PDF Generator – generates and appends the final index page to the uploaded book using ReportLab.

Figure 2 illustrates the overall activity diagram of the automatic indexing process, which encompasses steps for uploading the book, performing preprocessing, running RAKE and Word2Vec analyses, selecting phrases, and downloading the index.



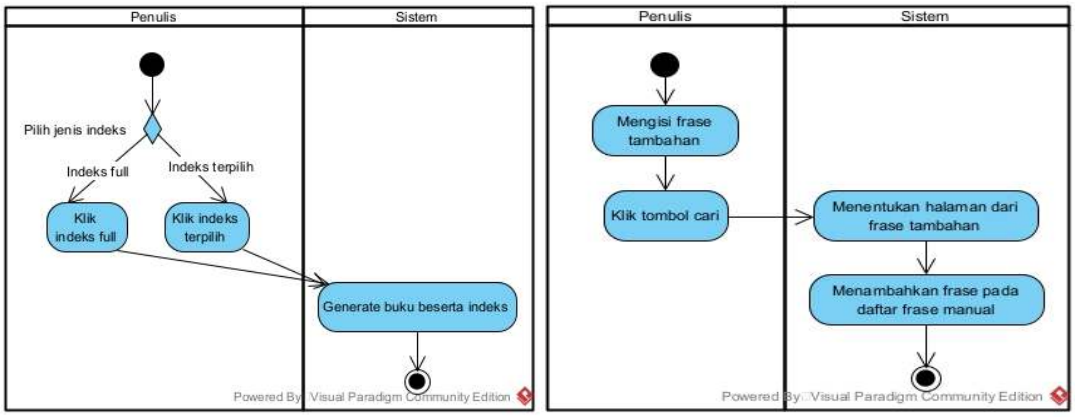


Figure 2. Activity Diagram of Automatic Indexing Process

2.5. IMPLEMENTATION

The system was implemented in Python using Flask. The RAKE algorithm was implemented via the rake-nltk library, and the pretrained FastText model was loaded using Gensim. The PDF manipulation and index generation were handled with PyMuPDF and ReportLab. The web application allows users to:

1. Upload a digital book (PDF) and enter a topic keyword.
2. View automatically extracted keywords with their RAKE and similarity scores.
3. Add manual phrases if necessary.
4. Select specific phrases to include in the final index.
5. Evaluate the system's performance by uploading a reference index.

2.6. EVALUATION

Study quality was assessed using methodology-appropriate tools. Qualitative studies were evaluated using The evaluation was conducted by comparing the automatically generated index with the author's reference index (ground truth). The metrics used were Precision, Recall, and Cosine Similarity.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

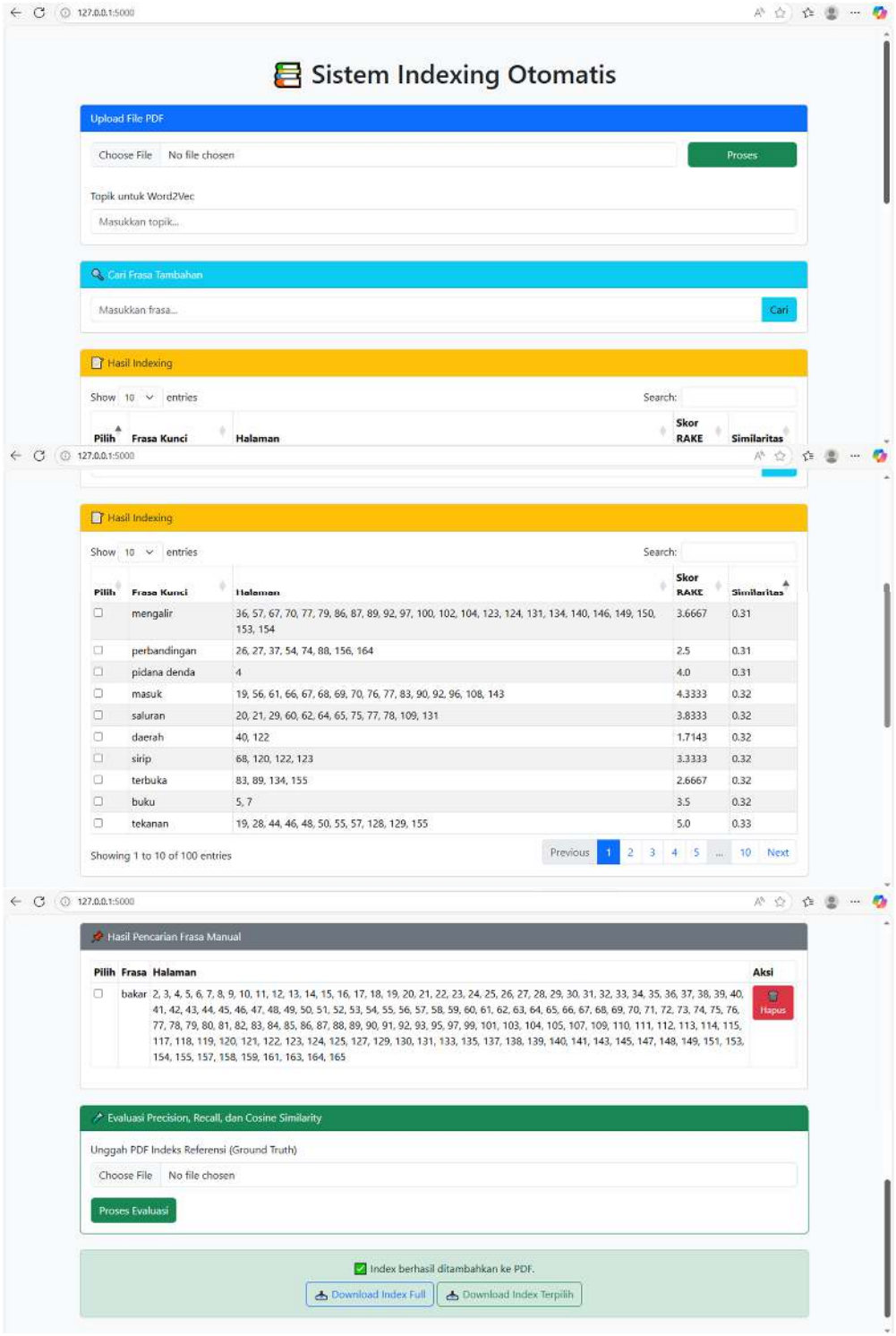
$$Recall = \frac{TP}{TP+FN} \quad (4)$$ where TP is the number of true positives, FP is false positives, and FN is false negatives.

Cosine similarity was also used to measure the overall vector similarity between the automatic and manual index sets. Table 2 presents an example of evaluation metrics obtained from one of the test cases.

Table 1. Example of Evaluation Metrics

Threshold	Phrase Length	Precision	Recall	Cosine Similarity
0.3	1-3 words	0.41	0.72	0.70
0.5	1-3 words	0.35	0.66	0.68
0.3	2-3 words	0.48	0.59	0.83
0.5	2-3 words	0.44	0.53	0.80

Figure 3 depicts the system interface showing the main dashboard of the automatic indexing web application.


127.0.0.1:5000
Sistem Indexing Otomatis
Upload File PDF
Choose File
No file chosen
Proses
Topik untuk Word2Vec
Masukkan topik...
Cari Frasa Tambahan
Masukkan frasa...
Cari
Hasil Indexing
Show 10 entries
Search:
Pilih
Frasa Kunci
Halaman
Skor RAKE
Similaritas
127.0.0.1:5000
Hasil Indexing
Show 10 entries
Search:
Pilih
Frasa Kunci
Halaman
Skor RAKE
Similaritas
mengalir 36, 57, 67, 70, 77, 79, 86, 87, 89, 92, 97, 100, 102, 104, 123, 124, 131, 134, 140, 146, 149, 150, 153, 154 3.6667 0.31
perbandingan 26, 27, 37, 54, 74, 88, 156, 164 2.5 0.31
pidana denda 4 4.0 0.31
masuk 19, 56, 61, 66, 67, 68, 69, 70, 76, 77, 83, 90, 92, 96, 108, 143 4.3333 0.32
saluran 20, 21, 29, 60, 62, 64, 65, 75, 77, 78, 109, 131 3.8333 0.32
daerah 40, 122 1.7143 0.32
sirip 68, 120, 122, 123 3.3333 0.32
terbuka 83, 89, 134, 155 2.6667 0.32
buku 5, 7 3.5 0.32
tekanan 19, 28, 44, 46, 48, 50, 55, 57, 128, 129, 155 5.0 0.33
Showing 1 to 10 of 100 entries
Previous 1 2 3 4 5 ... 10 Next
127.0.0.1:5000
Hasil Pencarian Frasa Manual
Pilih Frasa Halaman Aksi
bakar 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 95, 97, 99, 101, 103, 104, 105, 107, 109, 110, 111, 112, 113, 114, 115, 117, 118, 119, 120, 121, 122, 123, 124, 125, 127, 129, 130, 131, 133, 135, 137, 138, 139, 140, 141, 143, 145, 147, 148, 149, 151, 153, 154, 155, 157, 158, 159, 161, 163, 164, 165
Hapus
Evaluasi Precision, Recall, dan Cosine Similarity
Unggah PDF Indeks Referensi (Ground Truth)
Choose File
No file chosen
Proses Evaluasi
Index berhasil ditambahkan ke PDF.
Download Index Full
Download Index Terpilih

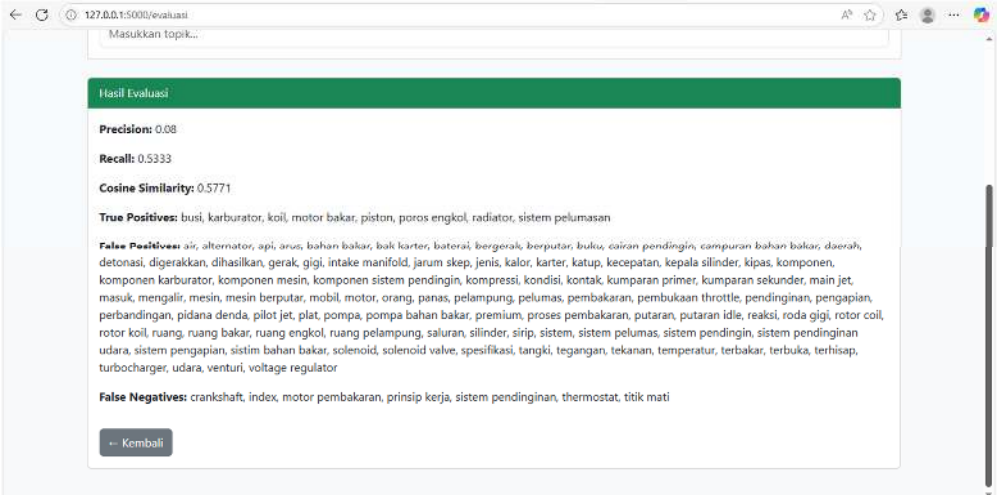


Figure 3. System Interface of Automatic Indexing Web Application

3. RESULT AND ANALYSIS

3.1. RESULT

This study produced an automatic indexing application for digital documents (PDF) that combines RAKE, Word2Vec, and manual keyword input methods. The system is capable of automatically extracting key phrases, determining their corresponding page numbers, and generating a new index page that is directly embedded into the original PDF file. Several main features were successfully implemented:

- PDF upload with file format validation.
- Automatic keyword extraction using RAKE and Word2Vec.
- Manual addition and search of custom index phrases.
- Phrase selection using checkboxes for inclusion in the final index.
- Evaluation of the automatic index compared to the author's original index using Precision, Recall, and Cosine Similarity.
- Generation of a new PDF index page that integrates seamlessly with the source document.

To evaluate the system's indexing performance, an author's index file was used as the reference (ground truth). Evaluation metrics included Precision, Recall, and Cosine Similarity, to measure how closely the system's results align with human-generated indexes.

The evaluation was performed on 30 Indonesian-language digital books, testing two similarity thresholds (0.3 and 0.5) and two phrase-length configurations (1-3 words and 2-3 words). The average results are summarized in Table 2.

Table 2. Average Evaluation Results of Automatic Indexing

Configuration	Threshold	Precision	Recall	Cosine Similarity
Top 100 phrases (1-3 words)	0.3	0.0986	0.2042	0.5872
Top 100 phrases (2-3 words)	0.3	0.0508	0.1082	0.6653
Top 100 phrases (1-3 words)	0.5	0.0423	0.07045	0.6517
Top 100 phrases (2-3 words)	0.5	0.0257	0.0505	0.6861

From Table 2, it can be observed that the 0.3 similarity threshold produces higher Precision and Recall compared to the 0.5 threshold, while the 0.5 threshold gives slightly higher Cosine Similarity. This indicates that a lower threshold allows more relevant phrases to be retrieved, increasing coverage but slightly reducing semantic precision.

To visualize the trend between both configurations, Figure 4 illustrates the comparative performance for the three metrics.

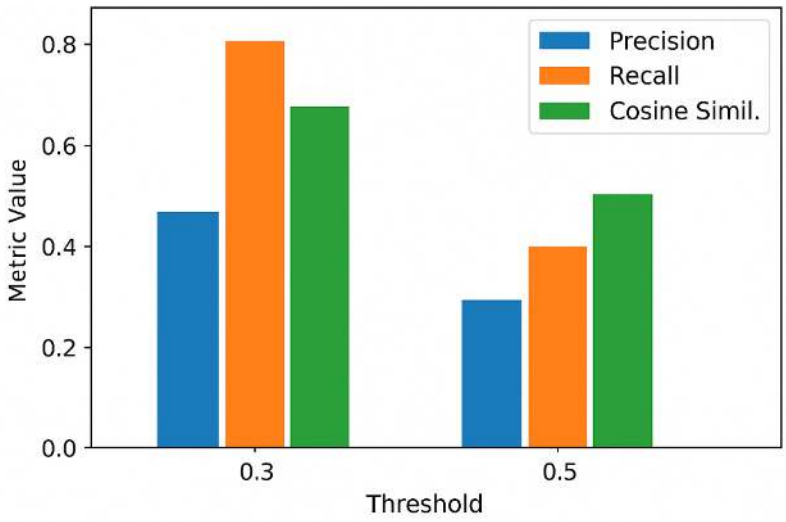


Figure 4. Comparison of precision, recall, and cosine similarity for thresholds 0.3 and 0.5

Based on the overall average from 30 tested books, the 0.3 threshold provides a more balanced result and is therefore considered more optimal for automatic index generation in this system.

3.2. ANALYSIS

The experimental results demonstrate that the developed system performs well in terms of functionality and integration. However, the accuracy—particularly the Precision score—remains relatively low. Several factors explain this outcome:

1. Imbalance in index quantity
   The author's index generally contains only a few highly selective entries, while the system generates up to 100 top-ranked phrases. This causes many phrases that are not considered important by the author to appear, thereby reducing precision.
2. Structural differences in keywords
   The RAKE algorithm is designed to extract multi-word expressions (typically 2-3 words). In contrast, the author's index often contains single-word terms. When RAKE's results are restricted to single words, the contextual meaning decreases, leading to lower alignment with human-created indexes.
3. Dominance of multi-word phrases
   Even though the system is configured to extract phrases of 1-3 words, most of the selected keywords tend to be 2-3 words (e.g., "Teknik abss" instead of simply "abss"). These phrases are semantically richer but deviate from the author's style, which favors concise single-word entries.

Despite the low precision, the system successfully identifies a large portion of relevant phrases that overlap with the author's index, resulting in relatively higher recall and strong cosine similarity values. This implies that the extraction mechanism is effective in capturing the semantic structure of important terms.

Overall, the combination of RAKE and Word2Vec methods demonstrates good potential for automatic indexing in Indonesian digital books. Future improvements could involve integrating context filtering, semantic clustering, or author-guided feedback mechanisms to enhance the system's precision without sacrificing recall.

4. CONCLUSION

This study successfully developed an automatic indexing system for digital books using a hybrid approach that combines the RAKE algorithm for keyword extraction and Word2Vec for semantic relevance analysis. The system can identify important phrases based on sentence structure and frequency patterns using RAKE, and then refine them

semantically according to the document topic using Word2Vec. An interactive web-based interface was implemented, supporting keyword search, manual input, phrase selection, and automatic PDF index generation appended to the original document.

However, based on the evaluation results, the current approach has not yet achieved sufficient precision to be considered fully effective as an independent indexing solution. With an average precision below 0.10 and an average recall below 0.20, the generated index still contains a substantial number of irrelevant phrases when compared to the author's reference index. The relatively higher cosine similarity (0.62 – 0.68) indicates that the system captures semantic proximity, yet this does not directly translate into accurate index selection.

From a practical standpoint, this research contributes to the automation of index compilation in digital publishing, reducing manual labor while maintaining contextual quality. Such a system could be integrated into digital library platforms, academic e-book systems, or content management tools, where efficient and semantically relevant indexing is essential for searchability and information retrieval.

For future research, several directions are recommended to enhance system performance:

- Optimizing RAKE for single-word keyword extraction to better align with conventional indexing styles.
- Incorporating Part-of-Speech tagging or Named Entity Recognition (NER) to improve phrase filtering and semantic accuracy.
- Employing domain-specific Word2Vec models (e.g., technical, economic, or educational corpora) to improve contextual precision.
- Expanding evaluation metrics beyond precision and recall to include user satisfaction and topical coverage.
- Integrating user feedback mechanisms for adaptive learning and iterative improvement.
- Testing on a broader and more diverse dataset to assess generalization capability across multiple domains.

In conclusion, the developed system demonstrates significant potential to support efficient and intelligent automatic indexing of digital books, enabling more relevant and semantically rich information retrieval in future digital library ecosystems.

ACKNOWLEDGEMENTS

The authors would like to express their deepest gratitude to Politeknik Negeri Malang, especially the Department of Information Technology and Polinema Press, for providing facilities, academic guidance, and research support throughout this study. Special appreciation is also extended to Vit Zuraida, S.Kom., M.Kom. and Dr. Ulla Delfana Rosiani, ST., MT., whose invaluable supervision, feedback, and encouragement greatly contributed to the successful completion of this research.

This work was conducted as part of the undergraduate final project titled "Automatic Indexing of Digital Books using RAKE and Word2Vec". The authors also acknowledge the availability of open-source linguistic resources such as the FastText Indonesian Word2Vec model, which played a crucial role in the system's semantic processing component.

REFERENCES

Ahmed, U., Alexopoulos, C., Piangerelli, M., & Polini, A. (2024). BRYT: Automated keyword extraction for open datasets. *Intelligent Systems with Applications*, 23(July). https://doi.org/10.1016/j.iswa.2024.200421

Asula, M., Makke, J., Freienthal, L., Kuulmets, H. A., & Sirel, R. (2021). Kratt: Developing an automatic subject indexing tool for the National Library of Estonia. *Cataloging and Classification Quarterly, 59*(8), 775–793. https://doi.org/10.1080/01639374.2021.1998283

Baffy, B. (2023). *Automatic keyword extraction for a partial search engine index*.

Baruni, J. S., & Sathiaseelan, D. J. G. (2020). Keyphrase extraction from document using RAKE and TextRank algorithms. *International Journal of Computer Science and Mobile Computing, 9*(9), 83–93. https://doi.org/10.47760/ijcsmc.2020.v09i09.009

Benita, I. R., & Baizal, Z. K. A. (2022). News recommender system based on user log history using Rapid Automatic Keyword Extraction. *Jurnal Media Informatika Budidarma, 6*(4), 2341. https://doi.org/10.30865/mib.v6i4.4554

Bingyu, Z., & Arefyev, N. (2022). The document vectors using cosine similarity revisited. *Insights 2022 – 3rd Workshop on Insights from Negative Results in NLP, Proceedings of the Workshop*, 129–133. https://doi.org/10.18653/v1/2022.insights-1.17

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). *Transactions of the Association for Computational Linguistics. Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://transacl.org/ojs/index.php/tacl/article/view/999

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences, 509*, 257–289. https://doi.org/10.1016/j.ins.2019.09.013

Di Gennaro, G., Buonanno, A., & Palmieri, F. A. N. (2021). Considerations about learning Word2Vec. Journal of *Supercomputing, 77*(11), 12320–12335. https://doi.org/10.1007/s11227-021-03743-2

Firoozeh, N., Nazarenko, A., Alizon, F., & Daille, B. (2020). Keyword extraction: Issues and methods. *Natural Language Engineering, 26*(3), 259–291. https://doi.org/10.1017/S1351324919000457

Gagliardi, I., & Artese, M. T. (2020). Semantic unsupervised automatic keyphrases extraction by integrating word embedding with clustering methods. *Multimodal Technologies and Interaction, 4*(2), 1–20. https://doi.org/10.3390/mti4020030

Garg, M. (2021). A survey on different dimensions for graphical keyword extraction techniques: Issues and challenges. *Artificial Intelligence Review, 54*(6). Springer Netherlands. https://doi.org/10.1007/s10462-021-10010-6

Gopan, E., Rajesh, S., Vishnu, G., Raj, A. R., & Thushara, M. G. (2020). Comparative study on different approaches in keyword extraction. *Proceedings of the 4th International Conference on Computing Methodologies and Communication (ICCMC 2020)*, 70–74. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00013

Guda, B., Nuhu, B. K., Agajo, J., & Aliyu, I. (2023). Performance evaluation of keyword extraction techniques and stop word lists on speech-to-text corpus. *International Arab Journal of Information Technology, 20*(1), 134–140. https://doi.org/10.34028/iajit/20/1/14

Hasegawa-Johnson, M. (n.d.). *Lecture 37: Word2Vec and word similarity*.

Huang, H., Wang, X., & Wang, H. (2020). NER-RAKE: An improved rapid automatic keyword extraction method for scientific literatures based on named entity recognition. *Proceedings of the Association for Information Science and Technology, 57*(1), 2–5. https://doi.org/10.1002/pra2.374

Jatnika, D., Bijaksana, M. A., & Suryani, A. A. (2019). Word2Vec model analysis for semantic similarities in English words. *Procedia Computer Science*, *157*, 160–167. https://doi.org/10.1016/j.procs.2019.08.153

Khan, M. Q., Shahid, A., Uddin, M. I., Roman, M., Alharbi, A., Alosaimi, W., Almalki, J., & Alshahrani, S. M. (2022). Impact analysis of keyword extraction using contextual word embedding. *PeerJ Computer Science*, *8*, 1–16. https://doi.org/10.7717/peerj-cs.967

Lebret, R., & Collobert, R. (2015). "The sum of its parts": *Joint learning of word and phrase representations with autoencoders*. arXiv preprint. http://arxiv.org/abs/1506.05703

Laila, S. N. (2020). *Tingkat keterbacaan buku teks kurikulum 2013 revisi 2007 kelas IV tema Indahnya Kebersamaan dan Selalu Berhemat Energi dengan menggunakan teknik fog index di SDN Kebonsari 04* (pp. 16–17).

Nomoto, T. (2023). Keyword extraction: A modern perspective. *SN Computer Science, 4*(1), 1–19. https://doi.org/10.1007/s42979-022-01481-7

Oniani, D. (2020). *Cosine similarity and its applications in the domains of artificial intelligence* (1–6).

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. In Text mining: *Applications and theory* (pp. 1–20). https://doi.org/10.1002/9780470689646.ch1

Sarwar, T. B., Noor, N. M., & Miah, M. S. U. (2022). Evaluating keyphrase extraction algorithms for finding similar news articles using lexical similarity calculation and semantic relatedness measurement by word embedding. *PeerJ Computer Science, 8*. https://doi.org/10.7717/peerj-cs.1024

Sivakumar, S., Videla, L. S., Rajesh Kumar, T., Nagaraj, J., Itnal, S., & Haritha, D. (2020). Review on Word2Vec word embedding neural net. *Proceedings – International Conference on Smart Electronics and Communication (ICOSEC 2020)*, 282–290. https://doi.org/10.1109/ICOSEC49089.2020.9215319

Steck, H., Ekanadham, C., & Kallus, N. (2024). Is cosine-similarity of embeddings really about similarity? *WWW 2024 Companion – Companion Proceedings of the ACM Web Conference*, 887–890. https://doi.org/10.1145/3589335.3651526

Supriadi, R., & Fitriyani, N. (2021). Analisis kesesuaian buku teks Bahasa Arab berbasis keterbacaan menggunakan ketentuan Fog Index. *Arabi: Journal of Arabic Studies*, *6*(1), 105. https://doi.org/10.24865/ajas.v6i1.232

Sukri, S., Samsudin, N. A., Fadzrin, E., Khalid, S. K. A., & Trisnawati, L. (2024). Word2Vec-based latent semantic indexing (Word2Vec-LSI) for contextual analysis in job-matching application. *International Journal of*

Advanced Computer Science and Applications, 15(3), 699–707. https://doi.org/10.14569/IJACSA.2024.0150371

Upadhyay, A., Bhatnagar, A., Bhavsar, N., Singh, M., & Motlicek, P. (n.d.). *An empirical comparison of semantic similarity methods for analyzing downstreaming automatic minuting task*. https://huggingface.co/

Wu, (2013). *Automatic book index generation* (pp. 1–6).

Yilmaz, S., & Toklu, S. (2020). A deep learning analysis on question classification task using Word2Vec representations. *Neural Computing and Applications, 32*(7), 2909–2928. https://doi.org/10.1007/s00521-020-04725-w