ISSN: 2623-1662 (online), 2460-092X (printed)
DOI: https://doi.org/10.19109/jusifo.v10i2.21951
Vol. 10, No. 2 (2024), p.99-106

Research Article [OPEN ACCESS]

Application of the Levenshtein Algorithm for Optimizing Search Accuracy in a Web-Based Knowledge Management System

Moch. Firmansyah*, Sartiawan Deswana

ABSTRACT

This study explores the integration of the Levenshtein algorithm into a web-based Knowledge Management System (KMS) to optimize information retrieval for Academic Service Centers (ASC) in universities. By addressing common challenges such as typographical errors during searches, the Levenshtein algorithm enhances the system's ability to deliver accurate results, improving user experience and service efficiency. The KMS is designed to manage diverse knowledge resources, including procedural guides, tutorials, and documentation, while ensuring accessibility and relevance for students and staff. Testing revealed an 88.2% similarity score in handling string mismatches, demonstrating the system's effectiveness in managing unstructured academic data. The findings emphasize the value of incorporating fuzzy matching techniques for robust knowledge management, particularly in higher education contexts where accurate information retrieval is critical. This research contributes a scalable framework for implementing KMS using string similarity algorithms, with potential applications extending to broader organizational settings.

Keyword: Levenshtein algorithm, knowledge management system, web-based

Received: February 22, 2024; Revised: November 21, 2024; Accepted: December 30, 2024

Corresponding Author: Moch. Firmansyah, Department of Informatics, Universitas Nasional, Indonesia; mochammad.firmansyah@gmail.com
Authors: Satriawan Deswana, Department of Informatics, Universitas Nasional, Indonesia, satriawan@civitas.unas.ac.id

1. INTRODUCTION

The Academic Service Center (ASC) plays a critical role in delivering services and managing administrative processes at universities. Typically, this center functions as a part of the Academic Administration Bureau, ensuring smooth administrative operations and providing optimal service to students. Its responsibilities include overseeing lecture activities, managing attendance records for students and lecturers, preparing lecture schedules, and organizing mid-term and final examinations. The operational activities of the ASC not only facilitate efficient university administration but also generate valuable knowledge. This knowledge is essential for maintaining operational continuity and enabling seamless integration for new employees within the center.

The concept of knowledge management refers to the systematic process of creating, storing, sharing, and utilizing knowledge within an organization (Alavi & Leidner, 2001; Demarest, 1997; Laal, 2011; Ragab & Arisha, 2013; Wiig, 1997). Organizations increasingly recognize that knowledge, whether explicit (documented) or tacit (residing in employees' minds), is a critical asset that requires effective management to sustain its value. Implementing Knowledge Management (KM) ensures smooth operations and high-quality services for students. Key challenges highlighting the importance of KM include managing complex administrative processes, ensuring service consistency, reducing human error and task duplication, and facilitating knowledge transfer to new staff members.

To address these challenges, the ASC should develop a Knowledge Management System (KMS) aimed at enhancing student services. The KMS can draw upon the SECI (Socialization, Externalization, Combination, and Internalization) model proposed by Nonaka & Takeuchi (1995). Such a system enables the center to manage crucial procedural, regulatory, and documentation processes, which often evolve with university or government policies. Service procedures must remain consistent, regardless of the staff providing them, and documentation, including Standard Operating Procedures (SOPs), guidelines, and checklists, should be comprehensive to mitigate human error. A well-designed KMS also aids in sharing essential administrative knowledge, particularly for onboarding new employees.

In this study, a web-based KMS is combined with the Levenshtein algorithm (Damerau, 1964; Levenshtein, 1966; Navarro, 2001) to optimize information retrieval. The system organizes knowledge in various formats, including documents, procedural guides, tutorial videos, and Frequently Asked Questions (FAQs). These resources promote service transparency while enriching learning opportunities for students outside the classroom. The KMS allows users to quickly locate academic procedures, guidelines, or forms. However, challenges arise when users make typographical errors or enter incorrect keywords during searches.

This research focuses on application the Levenshtein algorithm in a web-based KMS to address search-related issues caused by typographical errors. The algorithm's fuzzy matching approach minimizes discrepancies between user input and correct keywords, ensuring accurate search results (Adawiyah & Saragih, 2022; Arnawa, 2017; Liju, 2022; Octaria et al., 2019; Vidyarsih et al., 2016). The system not only enhances user experience but also strengthens the center's ability to provide efficient and precise information.

2. MATERIALS AND METHODS

2.1 Materials

The focus of this research is the implementation of the Levenshtein algorithm, a method introduced by Levenshtein (1966) to compute the edit distance between two strings. This algorithm determines the number of operations—such as insertion, deletion, or substitution—required to transform one string into another. A smaller edit distance signifies a higher degree of similarity between the two strings. By integrating the Levenshtein algorithm into search systems, Knowledge Management Systems (KMS) can detect even minimal typographical errors and suggest more accurate terms. This integration significantly enhances the relevance of search results, reduces barriers for students accessing knowledge, and ultimately improves the overall efficiency of Academic Service Center.

The growing importance of fuzzy matching highlights the need for more tolerant search methodologies. Fuzzy matching enables systems to identify similarities between user-inputted words and database entries, even when character discrepancies exist. For instance, when a user mistakenly types "knowladge", the system can still retrieve results related to "knowledge". This functionality is especially beneficial for Academic Service Centers serving thousands of students, as it minimizes errors and accelerates information retrieval.

Previous studies have demonstrated the utility of the Levenshtein algorithm in various applications. Febiawan et al. (2019) explored its use in detecting word similarity within research proposals and papers. Similarly, Khalidah (2021) applied the algorithm in digital biology dictionaries, aiding students in locating biological terms within databases. Other researchers have addressed its use for solving diverse problems, as seen in the works of Sadiah et al. (2020), Indriyono (2020), Syarafina et al. (2021), Udayana et al. (2024), Ene & Ene (2017), and Berger et al. (2021).

The procedural steps for implementing the Levenshtein algorithm in this research are illustrated in Figure 1 and Figure 2. These steps are executed sequentially to ensure that keyword searches yield outputs tailored to user needs.

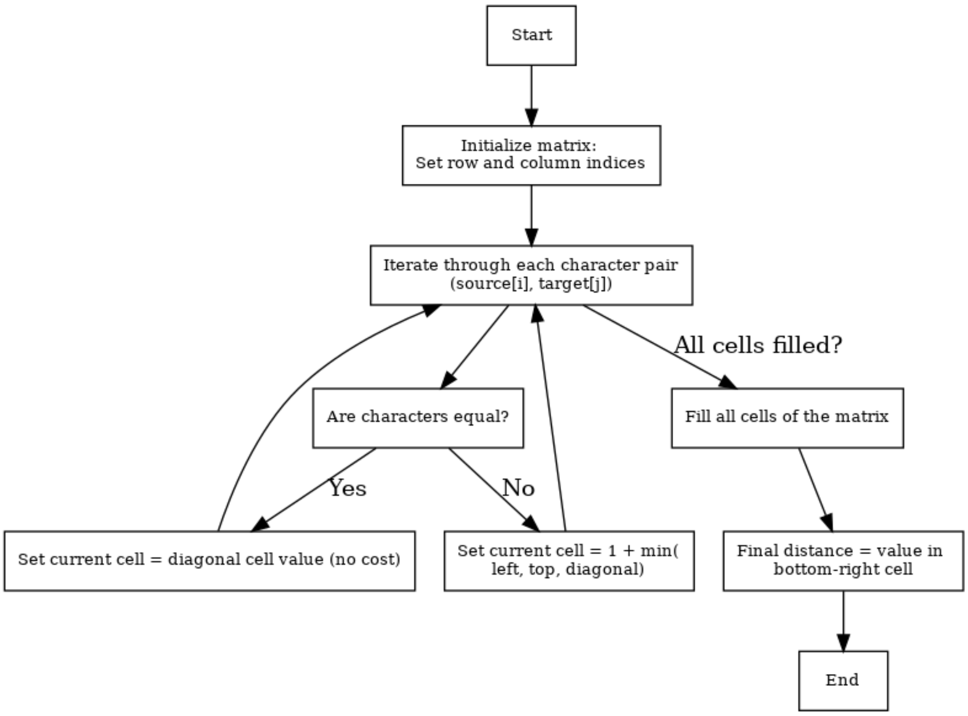

Figure 1. Implementation of the levenshtein algorithm in research

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Figure 2. Levenshtein distance equation

2.2 Methods

This study employs a structured research methodology divided into three key stages: Identification, KMS and Levenshtein Development, and Prototype Testing. These stages are systematically designed to achieve the research objectives effectively (Figure 3).

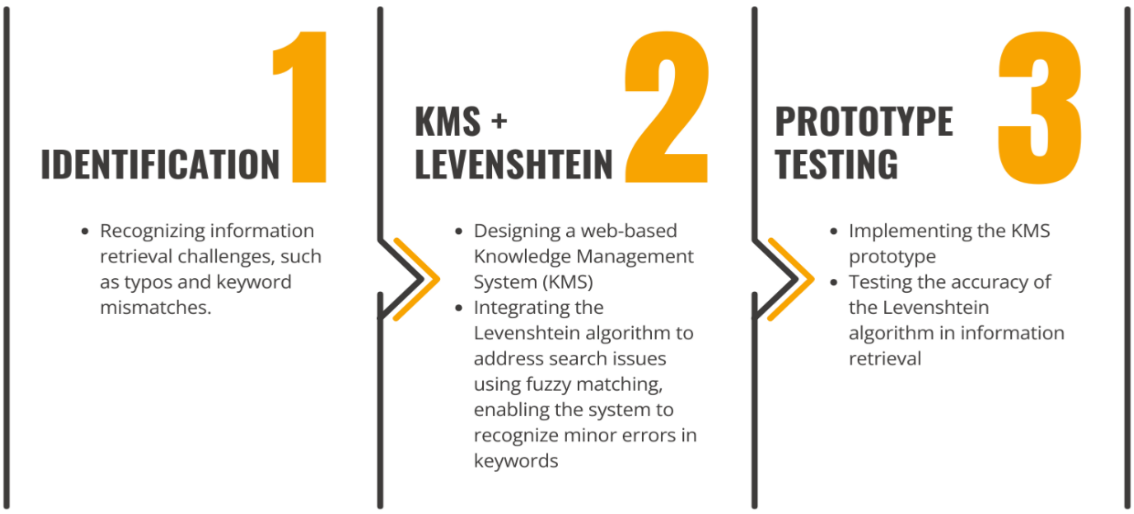

Figure 3. Research method

The first stage involves identifying the specific needs and challenges in knowledge management within the Academic Service Center. Data collection methods include interviews with staff, surveys of users, and analysis of existing documentation. Particular attention is given to identifying problems in information retrieval, such as typographical errors and keyword mismatches. The results from this stage provide the foundation for system requirements and design.

The second stage focuses on designing and developing a web-based KMS. The system architecture is built using a modular approach, ensuring scalability and flexibility. The Levenshtein algorithm is integrated into the search functionality to enhance the system's ability to handle fuzzy matching. This allows the system to tolerate minor typographical errors in user-inputted keywords.

The final stage involves the implementation and evaluation of the KMS prototype in an operational environment. The prototype is tested for its ability to retrieve accurate results despite input errors using a controlled dataset. Key metrics for evaluation include the accuracy of the Levenshtein algorithm in addressing typographical errors, user satisfaction, and efficiency improvements in service delivery. Feedback from users and system administrators is collected to refine the prototype and validate its effectiveness in meeting the identified needs.

3. RESULTS AND DISCUSSION

3.1 System Requirements

The system requirements for this study are categorized into two main types:

1. Functional Requirements
   - Login Functionality: users must be able to sign in to the application.
   - Logout Functionality: registered users must have the capability to log out of the application.
   - Data Management: the system must allow users to add, modify, and delete the following data categories:
     - Knowledge base content.
     - Question-related data.
     - User profiles and associated data.
2. Non-Functional Requirements
   - Accessibility: the application must be accessible via the web and operational 24/7.
   - User Restriction: only registered users are permitted to log out and manage application data.

3.2 System Planning

To address user needs effectively, the system planning phase involves designing and mapping the Knowledge Management (KM) process. This process is intended to facilitate the exchange of ideas, knowledge, and other relevant information. Table 1 outlines the technological design and its alignment with the KM process, detailing the required modules and features for each phase.

Table 1. Technology planning and mapping for the knowledge management process

KM Process	KM System Needs	Modules and Features
Socialization	A feature that allows SPA staff to interact with other parties so that tacit turns into tacit knowledge	1. Q&A 2. Discussion Forums 3. Search
Externalization	A feature that allows SPA staff to pour ideas, knowledge and more into other parties	1. Q&A 2. Document and article management 3. Search
Combination	A feature that allows SPA staff and other parties to exchange ideas and collaborate with each other so that new knowledge is formed	1. Q&A 2. Discussion Forums 3. Search
Internalization	Features that allow SPA staff to learn existing knowledge	1. Q&A 2. Discussion Forums 3. Search

3.3 Calculations in the Levenshtein Algorithm

The Levenshtein algorithm is applied to compute the edit distance between two strings by determining the minimum number of operations required to transform one string into another. These operations include character insertion, deletion, and substitution. The calculation process begins by constructing a comparison matrix based on the lengths of the source and target strings, where each matrix element is systematically populated. This matrix represents the edit distance values at each stage of character transformation. By employing this method, the algorithm identifies the most efficient path for executing the transformation. The computational mechanism of the Levenshtein algorithm is illustrated in Figure 4.

1

		U	A	S
	0	1	2	3
U	1			
N	2			
A	3			
S	4			

m is the length of the source string (UNAS) = 4,
n is the length of the target string (UAS) = 3.

The cost of each cell in the matrix, denoted as $\text{lev}_{i,j}$, is calculated using the recurrence relation:

$$\text{lev}_{i,j} = \begin{cases} \max(i,j), & \text{if } \min(i,j)=0, \\ \min \begin{cases} \text{lev}_{i-1,j}+1 & \text{(deletion)}, \\ \text{lev}_{i,j-1}+1 & \text{(insertion)}, \\ \text{lev}_{i-1,j-1}+\delta(a_i,b_j) & \text{(substitution)}, \end{cases} & \text{otherwise.} \end{cases}$$

Here, $\delta(a_i, b_j)$ is a binary indicator:

$$\delta(a_i,b_j) = \begin{cases} 0, & \text{if } a_i = b_j, \\ 1, & \text{if } a_i \neq b_j. \end{cases}$$

2

		U	A	S
	0	1	2	3
U	1	0	1	2
N	2			
A	3			
S	4			

Row 1 (Character: `U` from the Source String)

1. **Cell (1, 1): Comparing `U` (source) and `U` (target):**
   - Characters are equal ($a_1 = b_1$).
   - Cost: $\text{lev}(1,1) = \text{lev}(0,0) = 0$.
2. **Cell (1, 2): Comparing `U` and `A`:**
   - Characters differ ($a_1 \neq b_2$).
   - Cost:
   $$\text{lev}(1,2) = \min \begin{cases} \text{lev}(0,2)+1 = 3, \\ \text{lev}(1,1)+1 = 1, \\ \text{lev}(0,1)+1 = 2. \end{cases} = 1.$$
3. **Cell (1, 3): Comparing `U` and `S`:**
   - Characters differ ($a_1 \neq b_3$).
   - Cost:
   $$\text{lev}(1,3) = \min \begin{cases} \text{lev}(0,3)+1 = 4, \\ \text{lev}(1,2)+1 = 2, \\ \text{lev}(0,2)+1 = 3. \end{cases} = 2.$$

3

		U	A	S
	0	1	2	3
U	1	0	1	2
N	2	1	1	2
A	3	2	1	2
S	4	3	2	1

Complete the remaining sections of the matrix.

Final Result

The value in the bottom-right cell ($\text{lev}(4,3) = 1$) represents the **Levenshtein Distance** between **"UNAS"** and **"UAS"**.

This indicates that only **1 operation** (deletion of `N`) is required to transform **"UNAS"** into **"UAS"**.

Figure 4. Illustration of the levenshtein distance

3.4 Implementation of User Interface

Figure 5 illustrates the implementation of the user interface (UI) for the search feature, which represents a key focus of this study. The system incorporates the Levenshtein algorithm to calculate the edit distance between the user's inputted keywords and the data stored in the database. By employing this algorithm, the system effectively suggests the most relevant results, even when the entered keywords are not entirely accurate.

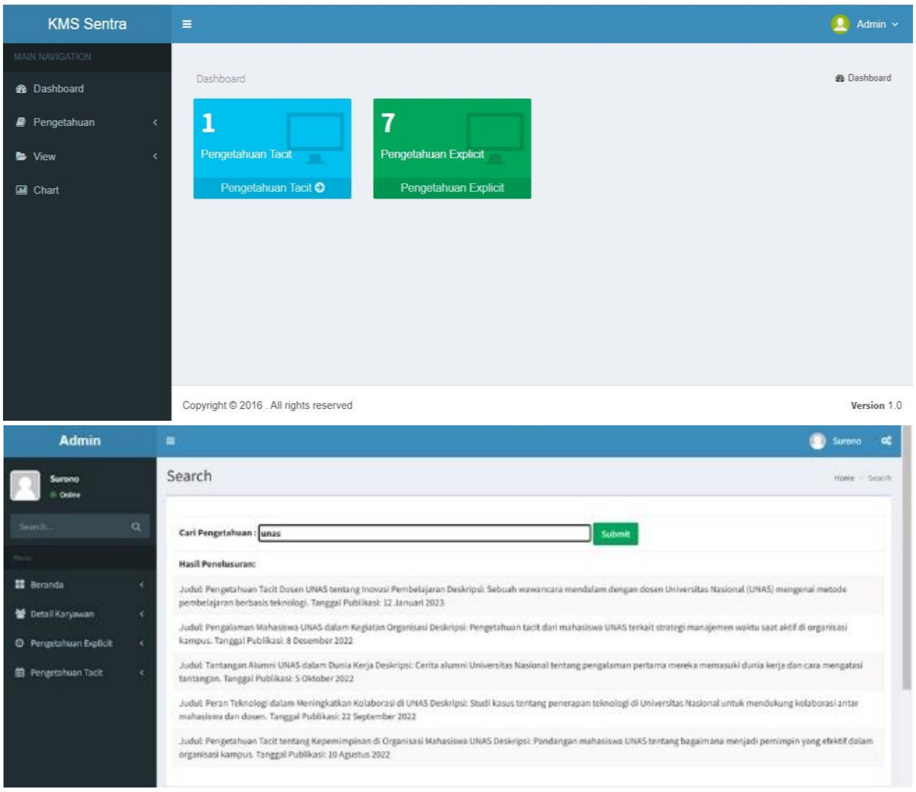

Figure 5. UI implementation

3.5 Testing and Evaluation of KMS Implementation

After completing the implementation of KMS, testing was conducted to evaluate its performance and user acceptance levels. The evaluation process included testing the Levenshtein algorithm by analyzing the similarity weights of several knowledge datasets within the KMS during search operations. The tests were performed using five separate attempts with different input words, sentences, or string lengths. The results of the similarity assessments are summarized in Table 2.

Table 2. String similarity evaluation

Search Attempt	**Number of Strings S (Source)**	**Number of Strings T (Target)**	**Similarity (%)**
1	5	5	100
2	14	10	71
3	21	19	90
4	9	9	100
5	25	20	80
Average			**88.2**

3.6 Discussion

This research developed and implemented a web-based KMS utilizing the Levenshtein algorithm, which demonstrated an 88.2% similarity performance score. This outcome signifies the system's effectiveness in managing unstructured academic data. The KMS facilitates knowledge sharing, enhances academic services, and boosts the productivity of both lecturers and administrative staff.

The study's findings are consistent with previous research that used the Levenshtein algorithm for plagiarism detection Adawiyah & Saragih (2022) and improved KMS search functionalities Octaria et al. (2019). However, this research makes a novel contribution by employing the algorithm specifically to support academic knowledge access for new or substitute lecturers and staff.

The results underscore the value of employing robust algorithms like the Levenshtein algorithm to address challenges in academic knowledge management. This finding highlights how information technology can enhance operational efficiency within higher education, particularly in onboarding and integrating new personnel into established systems.

The outcomes suggest that this system can be a model for other academic institutions aiming to improve knowledge management efficiency. Furthermore, the study opens pathways for extending the system's features, such as integrating automatic classification tools and user feedback mechanisms, to refine its functionality further.

These achievements stem from the Levenshtein algorithm's proven efficacy in string similarity measurement, which directly addresses the accuracy challenges inherent in searching unstructured academic data.

Future work will focus on testing the KMS with larger datasets, assessing its performance under high user loads, and incorporating advanced features such as automatic categorization and user feedback systems. Integrating the KMS with e-learning platforms will further enhance its applicability in academic settings.

4. CONCLUSION

This study successfully developed and implemented a web-based Knowledge Management System (KMS) at the Academic Service Center of the University, utilizing the Levenshtein algorithm to improve information retrieval. The system achieved a similarity score of 88.2%, demonstrating its effectiveness in managing unstructured academic data.

The primary contribution of this research lies in the innovative application of the Levenshtein algorithm to measure string similarity, which significantly enhanced the accuracy of search results. Beyond its technical implementation, this study offers a conceptual framework for improving academic processes, providing vital support to new or substitute lecturers in accessing key resources. Furthermore, it contributes methodologically by demonstrating the superiority of the Levenshtein algorithm over conventional keyword-based search methods.

However, this research acknowledges certain limitations, such as the small dataset used and the absence of performance evaluations under high user loads. These constraints highlight opportunities for future work. Subsequent studies could focus on testing the system with larger datasets, assessing its performance under heavy traffic, and investigating alternative algorithms or machine learning techniques to further improve search accuracy. Additionally, future enhancements could include advanced features such as automatic categorization and user feedback mechanisms, which would enhance the utility and functionality of the KMS.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Adawiyah, R., & Saragih, N. E. (2022). Implementasi algoritma levenshtein distance dalam mendeteksi plagiarisme. *Journal Computer Science and Information Technology(JCoInT)*, *3*(1), 54–63. https://jurnal.ulb.ac.id/index.php/JCoInT/article/view/3086

Alavi, M., & Leidner, D. E. (2001). Review: knowledge management and knowledge management systems: conceptual foundations and research issues. *MIS Quarterly: Management Information Systems*, *25*(1), 107–136. https://doi.org/10.2307/3250961

Arnawa, I. B. K. S. (2017). Implementasi algoritma levenshtein pada sistem pencarian judul skripsi/tugas akhir. *Jurnal Sistem Dan Informatika (JSI)*, *11*(2), 46–53.

Berger, B., Waterman, M. S., & Yu, Y. W. (2021). Levenshtein distance, sequence comparison and biological database search. *IEEE Transactions on Information Theory*, *67*(6), 3287–3294. https://doi.org/10.1109/TIT.2020.2996543

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, *7*(3), 171–176. https://doi.org/10.1145/363958.363994

Demarest, M. (1997). Understanding knowledge management. *Long Range Planning*, *30*(3), 374–384. https://doi.org/10.1016/S0024-6301(97)90250-8

Ene, A., & Ene, A. (2017). An application of levenshtein algorithm in vocabulary learning. *Proceedings of the 9th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2017*, *2017-January*, 1–4. https://doi.org/10.1109/ECAI.2017.8166449

Febiawan, M. H., Setiawan, A., & Primadewi, A. (2019). Sistem pendeteksi dini plagiarisme menggunakan algoritma levenshtein distance. *Jurnal Komtika (Komputasi Dan Informatika)*, *3*(1), 18–27.

Indriyono, B. V. (2020). Kombinasi damerau levenshtein dan jaro-winkler distance untuk koreksi kata bahasa inggris. *Jurnal Teknik Informatika Dan Sistem Informasi*, *6*(2), 2443–2229. https://doi.org/10.28932/JUTISI.V6I2.2493

Khalidah, K. (2021). Implementation of levenshtein distance algorithm in the digital biology dictionary search function. *Indonesian Journal of Artificial Intelligence and Data Mining*, *4*(2), 89–96. https://doi.org/10.24014/IJAIDM.V4I2.13989

Laal, M. (2011). Knowledge management in higher education. *Procedia Computer Science*, *3*, 544–549. https://doi.org/10.1016/J.PROCS.2010.12.090

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*. http://profs.sci.univr.it/~liptak/ALBioinfo/2012_2013/2011_2012/files/levenshtein66.pdf

Liju, P. P. M. (2022). *Algorithm to derive shortest edit script using Levenshtein distance algorithm*. https://arxiv.org/abs/2208.08823v1

Navarro, G. (2001). A guided tour to approximate string matching. *ACM Computing Surveys (CSUR)*, *33*(1), 31–88. https://doi.org/10.1145/375360.375365

Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company: how japanese companies create the dynamic of innovation*. Oxford University Press.

Octaria, O., Ermatita, E., & Sukemi, S. (2019). Penerapan knowledge management system menggunakan algoritma levenshtein. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *3*(2), 233–242. https://doi.org/10.29207/RESTI.V3I2.1045

Ragab, M. A. F., & Arisha, A. (2013). Knowledge management and measurement: a critical review. *Journal of Knowledge Management*, *17*(6), 873–901. https://doi.org/10.1108/JKM-12-2012-0381/FULL/XML

Sadiah, H. T., Ishlah, M. S. N., & Rokhmah, N. N. (2020). Autocorrect on drugs e-dictionary search module using levenshtein distance algorithm. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *4*(1), 64–69. https://doi.org/10.29207/RESTI.V4I1.1401

Syarafina, N. N., Palandi, J. F., & Radita, N. (2021). Designing a word recommendation application using the levenshtein distance algorithm. *Matrix: Jurnal Manajemen Teknologi Dan Informatika*, *11*(2), 63–70. https://doi.org/10.31940/MATRIX.V11I2.2419

Udayana, I. P. A. E. D., Putra, I. G. T. A., Udyana, I. P. S., Nugraha, I. G. S. C., Widantari, N. P., & Wijaya, B. K. (2024). Optimizing latin to balinese script transliteration: hybrid jaro winkler and damerau levenshtein methods. *2024 IEEE International Symposium on Consumer Technology (ISCT)*, 12–18. https://doi.org/10.1109/ISCT62336.2024.10791083

Vidyarsih, P., Abdillah, L. A., & Muzakir, A. (2016). *Sistem informasi pengarsipan menggunakan algoritma levensthein string pada kecamatan seberang ulu II*.

Wiig, K. M. (1997). Knowledge management: an introduction and perspective. *Journal of Knowledge Management*, *1*(1), 6–14. https://doi.org/10.1108/13673279710800682/FULL/XML