

Enhancing Typo-Tolerant Data Retrieval in the Andeng Village PKK Information System Using the Damerau-Levenshtein Distance Algorithm

Anggelia Deli¹, Christian Cahyaningtyas², Noviyanti P³, Florensus Andhika⁴

^{1,3,4}*Information Technology, Shanti Bhuana Institute, Bengkayang, Kalimantan Barat Indonesia*

²*information Systems, Shanti Bhuana Institute, Bengkayang, Kalimantan Barat Indonesia*

**Corresponding author: anggelia deli@gmail.com*

ABSTRACT

Data search is an essential feature in information systems that enables users to retrieve required information efficiently and accurately. However, the data search process in the Andeng Village Family Welfare Movement (PKK) Activity Management Information System is still limited because users must enter search queries that exactly match the stored data. Consequently, the system fails to retrieve relevant information when typographical errors (typos) occur in search queries. This study aims to improve the effectiveness of data retrieval by implementing the Damerau-Levenshtein Distance algorithm to support typo-tolerant searching, allowing the system to retrieve relevant data despite variations in user input. The algorithm was implemented by measuring the similarity between user queries and stored data using four edit operations, namely insertion, deletion, substitution, and transposition. The evaluation was conducted using 50 typo-containing search queries representing various typo patterns in PKK activity data, and the performance was measured based on retrieval accuracy and processing time. The results show that the implemented algorithm successfully retrieved relevant data despite typing errors in search queries. The testing achieved 100% accuracy on the tested PKK search scenarios with an average search processing time of 65.11 ms or 0.06511 seconds. These findings indicate that the Damerau-Levenshtein Distance algorithm provides effective and efficient performance for improving typo-tolerant data searching in the Andeng Village Family Welfare Movement (PKK) Activity Management Information System.

Keywords : *Data Search, Damerau-Levenshtein Distance, Type Errors, Search Performance, PKK System*

1. INTRODUCTION

The data search feature is an essential component of an information system that helps users find required information from data stored in a database. An effective search mechanism improves user efficiency by reducing the time required to locate information without manually browsing large amounts of data (Arifin & Sari, 2025). However, the increasing volume and complexity of stored data make conventional search methods based on exact keyword matching less effective, especially when users provide inaccurate queries or experience typing errors (Marbun, 2025).

Typographical errors (typos) are common problems that occur when users enter search queries, including missing characters, additional characters, incorrect characters, and transposition of adjacent characters (Susi Rianti, 2023). These errors may cause search systems to fail because the entered keywords do not exactly match the stored data. Therefore, modern

information retrieval systems have developed more flexible approaches beyond exact matching mechanisms. One of these approaches is fuzzy information retrieval, which enables systems to handle uncertainty and variations in user queries by considering similarity between input queries and available information. Fuzzy-based querying approaches improve search flexibility because relevant information can still be retrieved even when queries contain differences from stored data (Keskin & Yazıcı, 2022).

In addition, modern search mechanisms have incorporated recommendation concepts to improve information relevance. Search recommendation systems function as information filtering approaches that assist users in obtaining relevant information from large collections of data. By applying ranking mechanisms based on similarity values or relevance scores, recommendation approaches can reduce information overload and improve the efficiency of information access (Roy & Dutta, 2022). This concept is also related to the development of intelligent search engines, where information retrieval techniques, similarity matching, and ranking mechanisms are utilized to provide more accurate results according to user needs (Li et al., 2025).

Currently, the data search feature in the PKK Activity Management Information System of Andeng Village has not been optimized because users must enter keywords that exactly match the stored data. When typing errors occur, information related to members, activity planning, activity implementation, and attendance records cannot be retrieved. This limitation decreases the effectiveness of data management because administrators require more time to locate required information. Therefore, a search mechanism capable of recognizing word similarity and tolerating typing errors is required.

One method that can overcome this problem is the Damerau-Levenshtein Distance (DLD) algorithm. This algorithm measures the distance between two strings based on the number of operations required to transform one string into another. DLD supports four edit operations, namely insertion, deletion, substitution, and transposition of adjacent characters, making it suitable for handling common typing errors (Kurniawan et al., 2025); (Husni et al., 2025). Compared with Levenshtein Distance, which only considers three operations, and Hamming Distance, which requires strings with equal lengths, DLD provides better flexibility for typo correction (Nahampun, 2023); (Gulo, 2022).

Although previous studies have implemented edit distance algorithms for spelling correction and text matching, most applications focus on general text processing rather than village-level administrative information systems. Therefore, this study contributes by adapting fuzzy information retrieval through the Damerau-Levenshtein Distance algorithm into the PKK Activity Management Information System of Andeng Village. The proposed approach applies intelligent text matching and relevance-based ranking to improve search accessibility, usability, and efficiency while reducing failed searches caused by typographical errors.

2. METHODS

2.1 Research Stages

These research stages are systematically structured to ensure that the algorithm implementation process in the PKK activity management information system can proceed in a directed and structured manner, starting from problem identification to the evaluation stage. This ensures that each step is interconnected and that research results can be used to improve the quality of data retrieval in the PKK activity management information system.

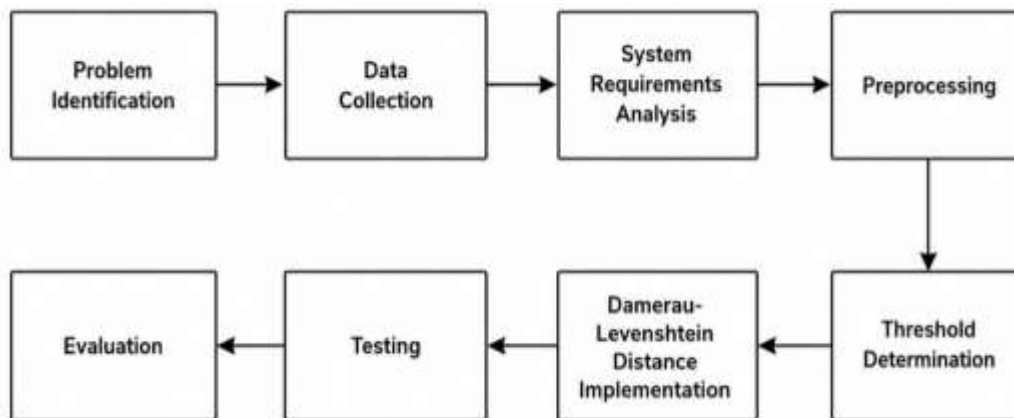


Figure 2. 1 Research Stages

The following is an explanation of Figure 2.1 regarding the stages carried out in the research, namely:

1. Problem Identification

This stage was conducted to identify problems occurring in the PKK activity management information system. The increasing amount of data has resulted in less efficient search processes. Furthermore, the search feature is still suboptimal because it relies on exact matching, requiring users to enter queries precisely. If a typing error occurs, the searched data cannot be found. This problem can hinder the data management process within the PKK organization.

2. Data Collection

Data collection was conducted through interviews, documentation, and literature studies. Interviews were performed with PKK administrators to obtain information regarding system requirements, data management processes, and existing problems in the data retrieval process. The interview results were also used to identify the main data categories used in this research. Documentation was conducted to collect existing PKK data stored in the system, including member data, activity planning data, activity implementation data, and attendance data. These data were used as the dataset for implementing and testing the Damerau-Levenshtein Distance algorithm.

The dataset used in this research consisted of four searchable data categories, as shown in Tabel 2. 1

Tabel 2. 1 Description of the Dataset Used for Search Testing

Data Type	Number Of Records	Searchable Attribute
Data perencanaan	30	Nama kegiatan/bidang,
Data pelaksanaan	20	Nama kegiatan/bidang

The dataset used in this study consists of activity planning data and activity implementation data from the Andeng Village PKK Activity Management Information System. The search process was performed flexibly based on activity information stored in the database, allowing users to search using available activity-related keywords. The searchable attributes consist of activity information contained in each activity planning and implementation record. The dataset was used to evaluate the capability of the Damerau-Levenshtein Distance algorithm in retrieving relevant information despite typographical errors in user search queries.

3. System Requirements Analysis

At this stage, an analysis of the system requirements to be developed was conducted based on the problems identified in the PKK activity management system. This analysis aimed to identify the requirements necessary for the system to support more effective and efficient data management processes. The analysis at this stage included the data retrieval process within the system. The search feature was applied to the member data, planning, implementation, and attendance menus. The search in the member data menu was used to find PKK member data, while the planning and implementation menus were used to find activity data. The attendance menu, on the other hand, was used to find member attendance data for each activity.

4. Pre-processing

Pre-processing is a crucial stage in research, aimed at cleaning, organizing, and converting raw data into a more suitable format for use in subsequent analysis. Pre-processed data results in more accurate and easier-to-interpret analysis (Fahreza et al., 2024). Before the algorithm can calculate distances, data from user input and data contained in the database must be preprocessed. This process is carried out to increase the accuracy of the results and reduce the computational burden on the algorithm. The steps taken in the pre-processing stage are as follows:

a. Space Cleaning

This process aims to remove spaces accidentally typed at the beginning or end of words. Furthermore, if there are double spaces in the middle of a sentence, the system will convert them to a single space. This is done to prevent spaces from being treated as additional characters that could affect the calculation results.

b. Case Standardization

In this stage, all characters in the text are converted to lowercase. This process aims to standardize the writing format so that the system can recognize similar words without being

affected by capitalization. This way, the algorithm will assign the same similarity score to the words "PKK" and "pkk," resulting in more consistent search results.

c. Symbol Cleaning

This stage removes punctuation and symbols such as periods (.), commas (,), exclamation points (!), or question marks (?) from the text. In this study, cleaning is crucial because the Damerau-Levenshtein Distance algorithm calculates similarity based on each character. If symbols are not removed, the system will treat them as part of a word, resulting in inaccurate calculations or even increasing the distance.

d. Word Tokenization

Tokenization is the process of breaking text into smaller parts, such as words. Word tokenization involves breaking text down into individual words, while sentence tokenization involves separating text into sentences. This process facilitates the system's analysis of text and supports subsequent language processing stages (Sulehu & Markani, 2024). Long sentences are broken down into single words through a tokenization process. For example, the sentence "PKK activities" is broken down into "activities" and "PKK." This method allows the system to search for matches within each word more flexibly.

e. Stemming

Stemming is a word normalization process that converts various affixed words into their base forms according to Indonesian morphological rules. Words containing affixes are converted to their base form by removing the affixes (Malik et al., 2026). For example, the word "berkebun" is changed to "kebun" to conform to the base form used in text processing. This process helps the system recognize words that have the same meaning despite their different forms, thus increasing the accuracy of the matching process.

2.1.6 Testing Protocol

The testing protocol was designed to evaluate the performance of the Damerau-Levenshtein Distance algorithm in handling typographical errors during the data search process. The test data were manually generated based on common typing error patterns that frequently occur during user input. The generated queries were adjusted to the PKK data context stored in the system.

A total of 50 test queries containing typing errors were prepared. These queries represented four types of edit operations supported by the Damerau-Levenshtein Distance algorithm: insertion, deletion, substitution, and transposition. Each test query was compared with the original keyword stored in the database to evaluate whether the system could retrieve the correct information.

Tabel 2. 2 Distribution of Typographical Error Types Used in the Testing Protocol

Error Type	Number of Tests
Insertion	10

Deletion	8
Substitution	5
Transposition	5
Mixed Errors	22
Total	50

The testing scenario shows that the 50 test queries consist of various typographical error patterns that commonly occur during the data search process. The highest number of test cases was found in the mixed error category with 22 cases because a single user query may contain more than one typing mistake, such as a combination of insertion, deletion, substitution, or transposition operations. Meanwhile, insertion errors consisted of 10 cases, deletion errors consisted of 8 cases, substitution errors consisted of 5 cases, and transposition errors consisted of 5 cases. This distribution reflects variations in user typing behavior and was used to evaluate the ability of the Damerau-Levenshtein Distance algorithm to handle different types of character modifications in search queries.

5. Determining Thresholds

Thresholds are a method used to determine a specific threshold value for a condition or event. This threshold value is used as a reference to distinguish whether a condition meets certain criteria or not. A threshold is used to enable the system to distinguish between relevant and irrelevant conditions (Ristanti & Pradana, 2022). The threshold value is determined to limit search results to ensure they remain relevant to the data sought by the user. In this study, the threshold value was set at 2, referring to research (Maheswari et al., 2025), which showed that the Damerau-Levenshtein Distance algorithm provided the most accurate results with a maximum editing distance of 2 errors. While using a higher editing distance tends to decrease the relevance of the results and increase processing time, this threshold value was also chosen based on the characteristics of the data in the Family Welfare Movement (PKK) activity management information system, which focuses on searching for member names, activity planning, activity implementation, and attendance data. The searched data generally consists of specific words or phrases that are not too long, so typing errors are usually small. Under these conditions, using a threshold value ≤ 2 can tolerate typing errors without reducing the relevance of the search results.

6. Implementation of the Damerau-Levenshtein Distance Algorithm

Damerau-Levenshtein Distance (DLD) is a method for measuring edit distance used to determine the level of similarity between two strings. The Damerau-Levenshtein Distance algorithm produces a value in the form of an edit distance, which is the number of editing operations required to transform one string into another. The smaller the resulting distance value, the more similar the two strings are. If the two strings are exactly the same, the distance value is zero. The operations in the Damerau-Levenshtein Distance algorithm are insertion, deletion, substitution, and transposition (James Tirta Halim et al., 2025). This step implements the Damerau-Levenshtein Distance algorithm function into a line of code. This algorithm

calculates the minimum number of steps to transform one string into another. The advantage of using the Damerau-Levenshtein Distance in this study is its ability to handle transposition errors (swapping two adjacent letters), which are the most common type of typing error made by users during searches.

The following is the Damerau-Levenshtein Distance calculation. For example, a user wants to search for "kebun" but makes a typing error and the query is "kebnu." To ensure the system still displays the intended data, "kebun," the Damerau-Levenshtein Distance algorithm is applied, which calculates the minimum number of edit operations required to transform one word into another.

The edit distance between two strings is calculated using a matrix. This matrix represents the minimum number of operations required to transform one string into another through letter deletion, insertion, substitution, and two-letter swapping. The value in the last cell of the matrix indicates the edit distance between the two strings.

1. Create a matrix as an initialization with a matrix size of $(m+1) \times (n+1)$ (Nahampun, 2023): where:

m = length of the target string

n = length of the input string

1 = Initial row and column of the matrix representing the empty string condition before the first character

In this case:

Target string = "kebun" character length = 5 $\rightarrow m = 5 + 1 = 6$

Input string = "kebnu" character length = 5 $\rightarrow n = 5 + 1 = 6$

So the matrix size is 6 x 6.

Fill in each cell in the matrix by comparing each letter with the formula (Kleshch, 2024) (Husni et al., 2025) :

$$D_{a,b}(i,j) = \min \begin{cases} 0 \\ D_{a,b}(i-1,j) + 1 & (\text{deletion}) \\ D_{a,b}(i,j-1) + 1 & (\text{insertion}) \\ D_{a,b}(i-1,j-1) + \text{cost} & (\text{substitution}) \\ D_{a,b}(i-2,j-2) + 1 & (\text{transposition}) \end{cases}$$

Description:

$D(i, j)$ = distance between strings a and b

a = input string

b = target string

i = character index in string a

j = character index in string b

Cost = 0 if $a_i = b_j$

1 if $a_i \neq b_j$

2. Comparing the letters k with k

The step of filling the matrix $D(1,1)$ is done by comparing the first character in the input string and the target string. The first character in the input string is "k", while the first character in the target string is also "k". Because both characters are the same, the cost = 0.

The value in cell $D_a, b(1,1)$ is obtained from three possible operations:

- 1) The top cell ($D_a, b(i-1, j) + 1$) represents the deletion operation, which removes one character from the input string.
- 2) The left cell ($D_a, b(i, j-1) + 1$) represents the insertion operation, which inserts one character into the input string to match the target string.
- 3) The diagonal cell ($D_a, b(i-1, j-1) + \text{cost}$) represents a character replacement or matching operation. If the characters are the same, the cost is 0; if they are different, the cost is 1.

The three possibilities are calculated as follows:

$$\text{Top cell} + 1 = 1 + 1 = 2$$

$$\text{Left cell} + 1 = 1 + 1 = 2$$

$$\text{Diagonal} + 0 = 0 + 0 = 0$$

Therefore:

$$D(1,1) = \min(2,2,0) = 0$$

Therefore, the value in cell $D(1,1)$ is 0, indicating that there is no difference in characters at that position.

		k	e	b	u	n
	0	1	2	3	4	5
k	1	0				
e	2					
b	3					
n	4					
u	5					

5. Comparison of the Last Letter u with n

At this stage, the system has detected a letter swap pattern (transposition) in adjacent characters. This condition occurs when the letter at the current position in the input string is the same as the previous letter in the target string, and the previous letter in the input string is the same as the current letter in the target string. In cell $D(5,5)$, a comparison is made between the last characters in the two strings: the character "u" in the input string "kebnu" and the character "n" in the target string "kebun". Although the two characters are different, the system checks for the possibility of a letter swap (transposition) in the adjacent characters, namely in

positions 4 and 5. Because the letters are swapped, a transposition operation is added by taking the values from the two previous diagonals.

Top cell + 1 = 1 + 1 = 2

Left cell + 1 = 1 + 1 = 2

Diagonal cell + 1 = 0 + 1 = 1

Two-step diagonal cell + 1 = 0 + 1 = 1

Therefore:

$D(5,5) = \min(2, 2, 1, 1) = 1$

	k	e	b	u	n	
k e b n	0	1	2	3	4	5
	1	0	1	2	3	4
	2	1	0	1	2	3
	3	2	1	0	1	2
	4	3	2	1	0	1
u	5	4	3	2	1	1

Based on the calculation results, the distance between the words "kebnu" and "kebun" is 1. This means that the word "kebun" appears because it falls within the threshold category.

8. Conclusion

This section explains the results of the edit distance calculation using the Damerau-Levenshtein Distance (DLD) algorithm when searching data in the PKK system. This is done by comparing the word typed by the user (input) with the word in the database (target) to obtain the distance value. This distance value is calculated based on the number of changes made, such as insertions, deletions, replacements, or letter swaps. This calculation allows us to see how accurately the system recognizes word similarities even when typing errors are present.

2.1.7 Testing

In this stage, the performance of the Damerau-Levenshtein Distance algorithm was tested in the data search process. Testing was carried out in two stages. The first stage tested the algorithm's accuracy by entering a number of words containing typos to determine whether the searched data could still be found by the system. The second stage calculated the average system processing time when searching for data.

The matching search results were then compared with the total test data. The tested data is adjusted to the context of the PKK data contained in the system. Based on the test results, the accuracy value is calculated using the following formula (Kurniawan et al., 2025):

$$\text{Accuracy} = \frac{\text{Number Of Corret Search Results}}{\text{Total Test Data}} \times 100\%$$

2.1.8 Evaluation

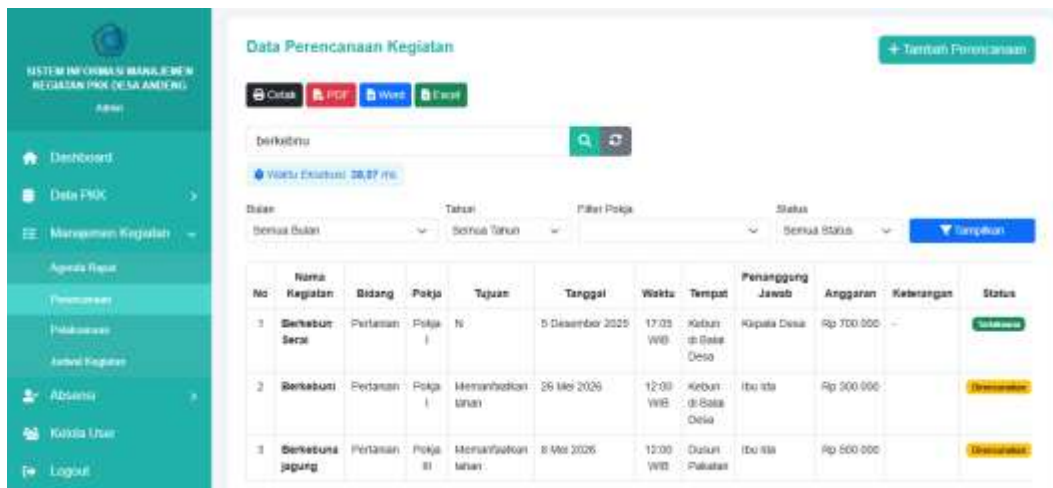
After obtaining the accuracy value, an evaluation phase was conducted to analyze the test results. The evaluation aimed to assess the performance of the Damerau-Levenshtein Distance algorithm in the data retrieval process. The results of this evaluation were used as the basis for drawing conclusions about whether the Damerau-Levenshtein Distance method was effective enough to assist the Andeng Village PKK administrators in managing data quickly and accurately.

3. RESULTS AND DISCUSSIONS

3.1 Implementation Results

2. Activity Planning Search Page

The activity planning page is used to manage data on planned PKK activities to be implemented. This menu displays information such as the activity name, activity area, implementation schedule, location, and person in charge. The data management process in this menu helps users organize and organize PKK activities in a more structured and efficient manner.



The screenshot shows a web application interface for managing PKK activities. The left sidebar contains a menu with options like Dashboard, Data PKK, Manajemen Kegiatan, and others. The main content area is titled 'Data Perencanaan Kegiatan' and features a search bar with the text 'berkebun'. Below the search bar, there are filters for 'Isian', 'Tahun', 'Pilih Pokja', and 'Status'. A table displays the search results with columns for No, Nama Kegiatan, Bidang, Pokja, Tujuan, Tanggal, Waktu, Tempat, Penanggung Jawab, Anggaran, Keterangan, and Status. The table contains three rows of data, each with a status indicator (Selesai, Berencana, or Dibatalkan).

No	Nama Kegiatan	Bidang	Pokja	Tujuan	Tanggal	Waktu	Tempat	Penanggung Jawab	Anggaran	Keterangan	Status
1	Berkebun Sera	Pertanian	Pokja I	16	5 Desember 2025	17:00 WIB	Kebun di Desa Odia	Kepala Desa	Rp 700.000		Selesai
2	Berkebun	Pertanian	Pokja I	Memanfaatkan lahan	26 Mei 2026	12:00 WIB	Kebun di Desa Odia	Ibu Ita	Rp 300.000		Berencana
3	Berkebun jagung	Pertanian	Pokja II	Memanfaatkan lahan	8 Mei 2026	12:00 WIB	Dusun Pakulan	Ibu Rina	Rp 500.000		Dibatalkan

Figure 3. 1 Activity Planning Search Page

Figure 3. 1 above shows the activity planning data menu. This menu stores information about the Family Welfare Program (PKK) activity plan, which serves as a pre-implementation planning tool. The activity planning menu, specifically the data search feature, implements the Damerau-Levenshtein Distance algorithm to allow for typographical errors. During the data search, there was a typo, where "berkebun" was instead "berkebun" (gardening). Despite the typo, the system still displayed relevant data with a search speed of 38.07 ms.

When searching, the word "berkebun" displays data similar to the searched data because the Damerau-Levenshtein Distance algorithm finds the distance between data and displays

similar data according to a predetermined threshold. Therefore, the words "Berkebuni" and "Berkebuna" in Figure 3. 1 still appear because the distance between "berkebun" and "Berkebuni" and "Berkebuna" is within the threshold. Furthermore, the "Gardening" and "Gardening" data appear due to possible typing errors during data input. The Damerau-Levenshtein Distance algorithm, which displays data based on distance between data points, is used for reference and evaluation of emerging data and potential data name errors.

3.2 System Testing

Table 3.1 shows the test results to determine how well the Damerau-Levenshtein Distance algorithm handles typos during the data search process. This is done by comparing the user's query with the data stored in the system. The test was conducted with several variations of typos, such as letter substitutions, letter additions, letter deletions, and letter repositioning. Furthermore, testing was conducted using keywords, lowercase letters, and additional symbols to ensure the pre-processing stage was effective in optimizing the Damerau-Levenshtein Distance algorithm's performance in handling differences between input words and those in the database. The test results demonstrated that the algorithm was capable of providing relevant search results despite typos. The following table shows the test results.

Table 3.1 System Testing

No	Input Query	Actual Words in the System	Processing Time	Test Results
1.	penyluhan giji	Penyuluhan gizi	80. 21 ms	successful
2.	berkebnu	Berkebun	38. 07 ms	successful
3.	Pelatihan ketrampilkan	Pelatihan keterampilan	35. 09 ms	successful
4.	mengqnyamm	Menganyam	15. 59 ms	successful
5	Pekarangan	perkarangan	55. 02 ms	successful
6.	pelatihhann	Pelatihan	67. 28 ms	successful
7.	Lobma masak, sehat	Lomba masak sehat	107. 28 ms	successful
8	menhahit	menjahit	48. 82 ms	successful
9	nanam	Menanam	48. 87 ms	successful
10	Linhungan pelestarian	Pelestarian lingkungan	33. 04 ms	successful
11	bakti sosal	Bakti sosial	75. 98 ms	successful
12	snam	Senam	14. 06 ms	successful
13	Gotongh royonh	Gotong royong	35. 96 ms	successful
14	kuarga	keluarga	42. 66 ms	successful
15	oengerlolaan	Pengelolaan	77. 10 ms	successful
16	pemanfataann	pemanfaatan	76. 37 ms	successful
17	Pertemn,yuan	Pertemuan	66. 44 ms	successful
18	Plesatrian	Pelestarian	19. 56 ms	successful

No	Input Query	Actual Words in the System	Processing Time	Test Results
19	Sosliassasi?	Sosialisasi	28. 92 ms	successful
20	kbersihann	kebersihan	69. 71 ms	successful
21	Edkuasi	Edukasi	61. 30 ms	successful
22	Klas parenitng	Kelas parenting	85. 07 ms	successful
23	lingkungan	lingkungan	25. 72 ms	successful
24	Propgram pkk	program PKK	65. 01 ms	successful
25	kertrampilan	keterampilan	67. 75 ms	successful
26	Kampadanye	Kampanye	73. 77 ms	successful
27	Magajamen,	Manajemen	65. 76 ms	successful
28	Penyuluhn kesehatan ,reproduksi	Penyuluhan kesehatan reproduksi	160. 65 ms	successful
29	pengebngan	Pengembangan	72. 26 ms	successful
30	Kerjd bakfi	Kerja bakti	72. 32 ms	successful
31	Binaan	Pembinaan	48. 79 ms	successful
32	Tatata boga,	tata boga	87. 35 ms	successful
33	Marekting?	marketing	63. 47 ms	successful
34	sabdbang	sandang	53. 59 ms	successful
35	Brekebun.,	Berkebuni	53. 06 ms	successful
36	uarausahaan	kewirausahaan	71. 01 ms	successful
37	Ucara ,.	Upacara	44. 16 ms	successful
38	Haei Kaartini	Hari Kartini	61. 31 ms	successful
39	brekebns	Berkebunas	65. 48 ms	successful
40	Prtemuan Ibu pkkk	Pertemuan Ibu PKK	91. 47 ms	successful
41	ekomim...	ekonomi	48. 36 ms	successful
42	batikkk	membatik	44. 94 ms	successful
43	Sayuan oragnik	sayuran organik	89. 60 ms	successful
44	Penfelolan sampah	Pengelolaan sampah	128. 28 ms	successful
45	Tbaman pikk	taman PKK	53. 86 ms	successful
46	Pelatigan buat kuee	Pelatihan pembuatan kue	116. 78 ms	successful
47	Pgmbuatn kompos	pembuatan kompos	108. 37 ms	successful
48	penanamn pojon	penanaman pohon	101. 27 ms	successful
49	tavaman	tanaman	61. 29 ms	successful
50	pengembagna	Pengembangan	77. 31 ms	successful

The test results are shown in Table 3.1, using 50 test queries containing typographical errors. All test queries successfully retrieved relevant data, with different processing times for each search process. The testing involved several typo patterns, including character insertion, deletion, substitution, transposition, and mixed errors. The results indicate that the Damerau-Levenshtein Distance algorithm can identify similar words despite differences between user

input and stored data. Furthermore, the preprocessing stage successfully normalized capitalization, spacing, and symbol variations before the similarity calculation process, allowing the system to focus on character differences that affect word similarity. The accuracy calculation was performed by comparing the number of successful retrieval results with the total number of test queries:

System accuracy test calculation:

$$\text{Accuracy} = \frac{50}{50} \times 100\% = 100\%$$

Calculation of the average search speed value:

$$\text{Average} = \frac{3255.39}{50} = 65.11 \text{ ms}$$

Based on the evaluation results, the system achieved 100% accuracy on the tested PKK search scenarios, where all 50 test queries successfully retrieved the expected data. This result demonstrates that the Damerau-Levenshtein Distance algorithm is capable of handling various typographical error patterns within the tested dataset. However, this accuracy value only represents the performance under the defined testing conditions and does not indicate that the same performance will always be obtained for different datasets or larger-scale systems. The average processing time of 65.11 ms shows that the system provides a fast response for the current dataset size. Nevertheless, processing time may increase as the number of records grows because the algorithm needs to calculate similarity between the search query and available data.

4. CONCLUSION

Based on the research findings, the Damerau-Levenshtein Distance algorithm was successfully implemented in the Andeng Village PKK Activity Management Information System to support typo-tolerant data retrieval. The algorithm is able to identify word similarity through edit distance calculations, allowing users to retrieve relevant activity data even when search queries contain insertion, deletion, substitution, or transposition errors. The preprocessing stages, including normalization, tokenization, and stemming, also contributed to improving the consistency of the matching process.

The evaluation results using 50 typo-containing search queries from PKK activity data showed that all tested queries successfully retrieved relevant information, achieving 100% accuracy with an average processing time of 65.11 ms. These results demonstrate that the proposed approach can improve the usability of the tested PKK information system by reducing search failures caused by typing errors. However, these results are limited to the tested dataset and predefined scenarios, so they do not represent the general performance of the algorithm for all information systems. In addition, this study has not compared the proposed method with other similarity algorithms, which can be considered for further evaluation.

The main contribution of this research is the implementation of a typo-tolerant search mechanism based on Damerau-Levenshtein Distance in a village-level administrative information system. This implementation provides practical benefits by helping PKK

administrators access activity information more easily and efficiently. Future research can improve the system by implementing hybrid search methods that combine text similarity and semantic approaches, applying machine learning-based ranking methods to improve result relevance, and developing semantic search capabilities to understand user search intent more accurately.

5. REFERENCES

- Arifin, M. K., & Sari, R. (2025). Perancangan dan Implementasi Fitur Pencairan Cepat pada Website Akademik Fakultas. *Journal of Informatic and Information Security*, 5(2), 155–166. <https://doi.org/10.31599/dvqd2c53>
- Fahreza, M. D. Al, Ardytha Luthfiarta, Muhammad Rafid, & Michael Indrawan. (2024). Analisis Sentimen: Pengaruh Jam Kerja Terhadap Kesehatan Mental Generasi Z. *Journal of Applied Computer Science and Technology*, 5(1), 16–25. <https://doi.org/10.52158/jacost.v5i1.715>
- Gulo, Y. N. (2022). Penerapan Algoritma Hamming Distance Untuk Pencarian Teks Pada Aplikasi Ensiklopedia Indonesia. *Journal Global Technology Computer*, 1(2), 50–54. <https://doi.org/10.47065/jogtc.v1i2.1378>
- Husni, H., Yoga Dwitya Pramudita, Mohammad Syarief, Army Justitia, & Suzanti, I. O. (2025). Improving Computational Efficiency and Accuracy of Damerau-Levenshtein Distance for Indonesian Spelling Correction using Cosine Similarity. *Journal of Innovation Information Technology and Application (JINITA)*, 7(2), 305–313. <https://doi.org/10.35970/jinita.v7i2.2893>
- James Tirta Halim, Lely Hiryanto, & Irvan Lewenusa. (2025). Pendeteksian Kesalahan Ketik Dengan Damerau-Levenshtein Distance Dan Trie. *Jurnal Ilmu Komputer Dan Sistem Informasi*, 13(1), 1–3. <https://doi.org/10.24912/jiksi.v13i1.32854>
- Keskin, S., & Yazıcı, A. (2022). Modeling and Querying Fuzzy SOLAP-Based Framework. *ISPRS International Journal of Geo-Information*, 11(3), 1–30. <https://doi.org/10.3390/ijgi11030191>
- Kleshch, K. (2024). Development of Fuzzy Search Method for Creating an Efficient Information Search System in Text Data. *Technology Audit and Production Reserves*, 1(2), 20–24. <https://doi.org/10.15587/2706-5448.2024.298425>
- Kurniawan, Y. D., Purnomo, A. S., Informatika, T., Informasi, F. T., Mercu, U., Yogyakarta, B., & Distance, L. (2025). Implementasi Algoritma Levenshtein Distance. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(1), 1411–1416.
- Li, X., Jin, J., Zhou, Y., Zhang, Y., Zhang, P., Zhu, Y., & Dou, Z. (2025). From Matching to Generation: A Survey on Generative Information Retrieval. *ACM Transactions on*

Information Systems, 43(3). <https://doi.org/10.1145/3722552>

- Maheswari, D. P., Putri, D. P. S., & Wirdiani, K. A. (2025). Implementation of Dictionary Lookup and Damerau-Levenshtein Distance for Correcting Non-Standard Indonesian and English Terms. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, 13(1), 63. <https://doi.org/10.24843/jim.2025.v13.i01.p07>
- Malik, A., Irawan, B., & Setiabudi, U. M. (2026). Penerapan Algoritma Bi-Lstm Dengan Optimasi Threshold Adjustment Untuk Analisis Sentimen Ulasan Aplikasi Mobile Jkn. *JITET (Jurnal Informatika Dan Teknik Elektro Terapan)*, 14(1).
- Marbun, N. (2025). Sistem Pencarian Data Karyawan Menggunakan Algoritma Boyer Moore. *Ilmu Komputer, Pemrosesan Data Dan Teknik Informatika*, 01(02), 10–14.
- Nahampun, O. H. (2023). Penerapan Algoritma Levenstein Distance Untuk Pencarian Judul Pada Aplikasi Lagu-Lagu Nasional. *RESOLUSI: Rekayasa Teknik Informatika Dan Informasi*, 3(4), 149–153. <https://djjournals.com/resolusi>
- Ristanti, N. L. P. T., & Pradana, R. (2022). Penggunaan Metode Threshold dalam Pembuatan Sistem Pendeteksi Asap dan Api dengan Berbasis Firebase Dan Android Menggunakan Nodemcu Pada BJ House 77. *Jurnal Ticom: Technology of Information and Communication*, 11(1), 44–49. <https://doi.org/10.70309/ticom.v11i1.70>
- Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9(1). <https://doi.org/10.1186/s40537-022-00592-5>
- Sulehu, M., & Markani, M. (2024). Integrasi Pemrosesan Bahasa Alami dalam Bot Telegram untuk Pengajaran Mata Kuliah Website Lebih Interaktif. *Jurnal Minfo Polgan*, 13(1), 759–771. <https://doi.org/10.33395/jmp.v13i1.13793>
- Susi Rianti, R. A. S. (2023). Perbandingan Algoritma Edit Distance, Levenshtein Distance, Hamming Distance, Jaccard Similarity Dalam Mendeteksi String Matching. *JSI (Jurnal Sistem Informasi) Universitas Suryadarma*, 10(1), 305–314. <https://doi.org/10.35968/jsi.v10i1.1010>