

Development of Administrative Document Search System Applying *Jaro-Winkler* Method (Case Study: Cipta Karya Village)

Essya¹, Yuliana², and Maria Magdala Albrisia Putri³

^{1,2,3}Information Technology Study Program, Institute Shanti Bhuana, Bengkayang, Indonesia

*Corresponding author: essyaesa0709@gmail.com

ABSTRACT

Document search is the process of finding and retrieving the required documents from the data storage system in digital form. An effective document search system is needed in village administration services, especially to overcome problems such as data irregularity, accumulation of documents and typing errors in document names so that it takes a long time to find documents. The jaro-winkler method is used as a string matching method to measure the level of similarity between user input and document data in the database, so that it is able to display search results even though there are slight differences in writing. The document data used to search are 100 documents accommodated in the database. From this data, the calculation is carried out by going through several stages, namely determining the matching characters as many as 25 with the number of transpositions worth 0.5. In the test results, the system is able to identify documents even though there are typing errors with a similarity score above the 0.90 threshold with a similarity score of 0.9961 so that the system can consider the string a match and display the appropriate document. Thus, the application of the jaro-winkler method effectively increases the accuracy and speed in finding administrative documents to facilitate village officials in providing optimal services. Document search is the process of finding and retrieving the required documents from the data.

Keywords: Document Search, Jaro-Winkler, Village Administration, String Matching Integration.

1. INTRODUCTION

Administrative tasks in rural settings often involve handling a large number of documents with varying formats, names, and structures. In Cipta Karya Village, the existing digital administrative system has failed to meet user expectations due to its dependence on exact keyword matches during searches. This limitation has resulted in long retrieval times, failure to find relevant documents when input errors occur, and increased workload for staff tasked with managing public services (Glenn Tambahani et al., 2021).

Document search is the process of finding and retrieving the required documents from the data storage system, either in digital or physical form. The document search system must be able to handle typos in search input, handle non-uniform document names, and display relevant search results in a short time. The documents included in this village administration system include cover letters and certificate letters (Syahputra & Syakti, 2022).

An effective document search method is essential for addressing this issue. One approach that can be utilized is the Jaro-Winkler method, a string comparison algorithm designed to enhance the relationship between two texts through improved character analysis and understanding. This method is particularly beneficial when drafting administrative

documents, as it can identify discrepancies between the data or names entered by users and the data stored in the system, even when there are only slight variations in writing style or format. The application of this method is expected to enhance document searches, reduce data errors, and increase the work efficiency of village officials (Salim et al., 2021).

The implementation of a document search system utilizing the Jaro-Winkler method not only assists the community in locating documents more quickly but also boosts community satisfaction with the administrative process. Consequently, individuals will not have to wait long to obtain the documents they need, allowing village officials to concentrate more on other tasks. Thus, this system will positively impact the effectiveness and transparency of village administration services (Rachman et al., 2024).

The purpose of this writing, as outlined in the problem formulation above, is to analyze the effectiveness of the Jaro-Winkler method in increasing the speed of searching for population administration documents in Cipta Karya Village. Additionally, it aims to develop and implement an administrative document search system using the Jaro-Winkler method to expedite and facilitate document searches.

2. METHODS

The Jaro-Winkler method is one of the techniques used to evaluate the similarity between two strings. It is an extension of the Jaro Similarity algorithm, which analyzes character-level matches and transpositions between the strings. The higher the resulting similarity score, the greater the degree of resemblance between the two strings (Sanjaya, 2020). There are three steps that form the basis of the Jaro-Winkler algorithm in determining the similarity or similarity of strings, namely

1. Comparing strings that are being calculated.
2. Counting the number of characters in common between the strings being compared.
3. Counting the number of transpositions.

The formula for calculating the distance (d_j) between two strings S_1 and S_2 in Jaro-Winkler is:

$$J = \frac{1}{3} \times \left(\frac{m}{S_1} + \frac{m}{S_2} + \frac{m-t}{m} \right)$$

Description:

S_2	=	String	Length	1
S_2	=	String	Length	2
m	=	Number of characters in common between S_1 and S_2		
t	=	Number of transpositions		

The flowchart of document search using the Jaro-Winkler method in this study is described in the following figure:

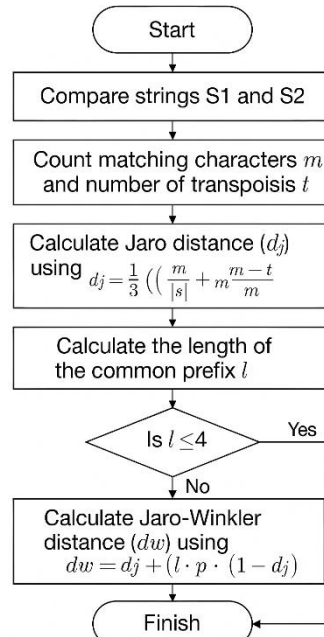


Figure 1. Flowchart Method Jaro-Winkler

2.1 Research Approach

The Jaro-Winkler method is a string-matching technique used to evaluate the similarity between two strings. It extends the Jaro Similarity algorithm by incorporating a prefix scale that gives higher scores to strings that match from the beginning. The algorithm analyzes character-level matches and transpositions, making it effective for detecting typographical errors. A higher similarity score indicates a greater degree of resemblance between the two strings (Pinajeng et al., 2020).

2.2 Data Collection

Two primary methods were used to gather data:

1. Observation

The researchers observed current workflows, including how staff search for documents, the average time spent, and difficulties encountered.

2. Documentation

Samples of administrative documents, database schemas, and search logs were collected to inform system design and validate functionality.

2.3 System Requirement Identification

The identification of system requirements is categorized into several types, namely functional, non-functional, technical, and user requirements. The purpose of this identification process is to ensure that the developed system aligns with user needs and enhances the effectiveness of administrative services in Cipta Karya Village. The table below presents the results of the system requirements identification in a structured format:

Table 1. System Requirement Identification

Category of Need	Description
Function	- Document search based on user input - Able to correct name typing errors
Non-Function	- Fast search process - Easy to use interface for village officials
Technical	- Integrated with MySQL database - Use the Jaro-Winkler algorithm as the core logic
User	- Village officials who are not familiar with technology can operate the system - Search logs are available

3. RESULTS AND DISCUSSIONS

3.1. Jaro-Wikler Method Calculation

The following is an example of the application of the Jaro-Winkler calculation by taking the problem identification of the system in Chapter 2. If we want to find the document “Cover Letter for Bella's KK”, but the village officials type:

“Cover Letter for KK Bela” (there is a typo in the name ‘Bella’ → “Bela”).

We will calculate the degree of similarity between the following two strings:

String 1 (s1) = “Cover Letter for Bella” (length 26 characters including spaces)

String 2 (s2) = “Cover Letter for Bela” (length 25 characters including spaces)

1. Jaro Formula

$$J = \frac{1}{3} \left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m} \right)$$

Description:

m = number of matching letters (matching characters)

t = number of transpositions (number of matching letters but different positions, divided by

2)

|s1|, |s2| = Length of each string

2. Determining Matching Characters

Length of string s1 = 26
 Length of string s2 = 25
 Matching rules: A character of string 1 will be considered to match a character of string 2 if that character:
 - Are the same
 - And are within the distance

$$\text{Window} = \left(\frac{\max(|s1|, |s2|)}{2} \right) - 1$$

$$= \left(\frac{\max(26, 25)}{2} \right) - 1 = 13 - 1 = 12$$

String s1 = 26, s2 = 25
 Maximum matching distance:
 $\left\lfloor \frac{26}{2} \right\rfloor - 1 = 12$

That is, each character s1 will be compared to a character s2 that is within ± 12 positions of the character index. So, this is used to keep the matching relevant by only matching letters that are relatively close.

Compare the characters of both strings one by one in a 12-character window. This window determines how far the characters in one string can look left and right in the other string to find a matching pair. If a character is outside the range of the ± 12 window, then:

s1: LETTER TO BELLA
 s2: LETTER TO BELA
 Note that all characters from the beginning to "BELA" match, except:
 s1 = 'L', 'L', 'A'
 s2 = 'L', 'A'
 So, matching characters = 25
 (The second 'L' in s1 does not have an immediate matching pair in s2)

3. Count the Number of Transpositions

Transposition is calculated as half the number of matching characters that are not in the same position, divided by 2.
 From the last position check:
 - S1[23] = L, s2[23] = A (no match)
 - S1[24] = A, s2[24] = " (no match)
 - But the character A from s1 at position 24 is matched by s2[22] which has 'A', out of position.

So it counts as 1 transposition, because the matching position is different. Since one transposition is counted twice (one for s1, one for s2), it must be divided by two:
 Thus:

$$t = \frac{1}{2} = 0.5$$

4. Calculate Jaro Value (J)

Given:

Matching	characters	(m)	=	25
Transposition	(t)	=		0.5
s1 =	26,	s2 =		25
Prefix	(p)	=		4

If Scale Factor (l) = 0.2

Enter a number:

$$J = \frac{1}{3} \left(\frac{25}{26} + \frac{25}{25} + \frac{25-0.5}{25} \right) = \frac{1}{3} (0.96154 + 0.98) = \frac{2.94154}{3} = 0.98051$$

5. Calculate Prefix Length (p)

Jaro-Winkler adds weight based on matching prefixes up to a maximum of 4 letters.

Matching prefixes from both strings:

“S”, “U”, “R”, ‘A’, “T” → 5 letters are the same, but the maximum $p = 4$ is part of the algorithm design to avoid excessive bias towards long prefixes.

So use prefix length (P) = 4

6. Jaro-Winkler Count

Formula:

$$JW = J + (l \cdot p \cdot (1 - J))$$

Description:

J = jaro result

l = scale factor 0.2

p = Prefix length (max 4)

$$JW = 0.98051 + (4 \times 0.2 \times (1 - 0.98051)) = 0.98051 + 0.8 \times 0.01949 = 0.98051 + 0.01559 = 0.9961$$

Final

result:

The Jaro-Winkler similarity score between “Bella's KK Cover Letter” and “Bela's KK Cover Letter” is 0.9961 or 100%. With a score above the 0.90 threshold, the system can consider the strings a match, and display the corresponding documents despite typing errors.

3.2. Implementation of Search with Jaro-Winkler Method

The document search system using the Jaro-Winkler method is implemented to improve document search accuracy despite typos from users. The following is an explanation of the flow of program logic in the system.

3.3. Document Search Program Logic with Jaro-Winkler

1. User Input

2. Retrieval of Document Data from Database
3. Calculation of Similarity Score
4. Filtering Based on Threshold
5. Sorting Search Results
6. Displaying Search Results

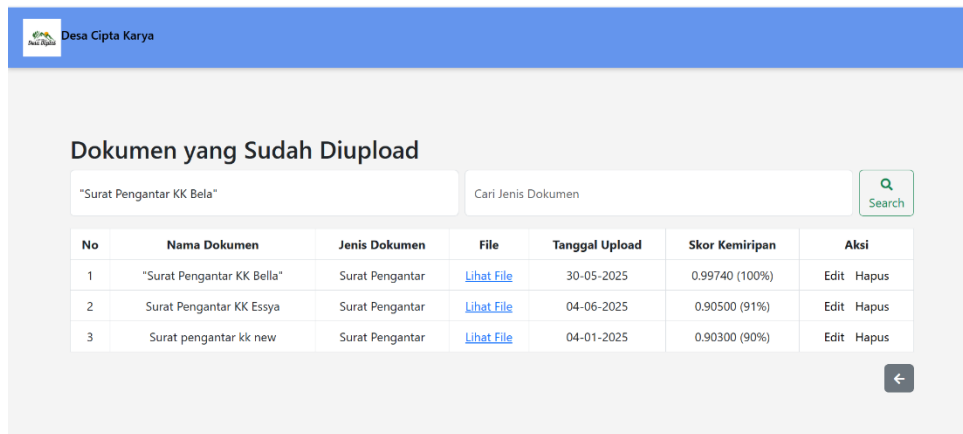
3.4. Search System Flow

- 1) The user inputs keywords to the system.
- 2) The system retrieves all document names from the database.
- 3) For each document name, the system calculates the Jaro-Winkler score against the user input.
- 4) **The system selects documents with a score ≥ 0.90 .**
- 5) The results are sorted and displayed to the user.

3.5. System Interface View

This administrative document search system was developed using a web-based approach, utilizing the PHP programming language, a MySQL database, and the Jaro-Winkler method to calculate word or string similarity. The system interface is designed to be user-friendly for village officials and to streamline the administrative service process.

The following is a view of the document search interface in the system:



No	Nama Dokumen	Jenis Dokumen	File	Tanggal Upload	Skor Kemiripan	Aksi
1	"Surat Pengantar KK Bella"	Surat Pengantar	Lihat File	30-05-2025	0.99740 (100%)	Edit Hapus
2	Surat Pengantar KK Essya	Surat Pengantar	Lihat File	04-06-2025	0.90500 (91%)	Edit Hapus
3	Surat pengantar kk new	Surat Pengantar	Lihat File	04-01-2025	0.90300 (90%)	Edit Hapus

Figure 2. Document Search Page

On this page, users are provided with the convenience of searching for documents by simply entering the desired document name into the search field. Once the user inputs a document name, the system automatically compares it with the names of documents stored in the database using the Jaro-Winkler method—an algorithm recognized for its effectiveness in measuring string similarity, even in the presence of typographical errors. The search process involves calculating a similarity score between the user's input and each document name in the database. If the resulting score exceeds a predefined threshold (e.g., 0.90), the system displays the most relevant results. This enables users to locate the intended documents even when minor input errors occur, thereby enhancing both the user experience and the efficiency of information retrieval. In this way, the system not only simplifies the document search

process but also offers a degree of error tolerance, ensuring greater accessibility and accuracy in administrative services.

3.6. System Testing

Black Box Testing is a software testing technique that focuses on testing system functionality without understanding the core code structure or program logic. In this method, the tester only examines the system's inputs and outputs and determines whether the system can perform the expected functions in a way that meets user needs. This testing does not require an understanding of how the system is programmed internally, but rather tests the system from the end user side. Each test case consists of:

- Input: Data or commands that the user provides to the system.
- Expectation: The expected result of the system based on the input.
- Actual Output: The actual result that the system delivers after receiving the input.
- Status: Whether the actual output matches the expectation or not.

The following are the results of black box testing applied to the document search system using the Jaro-Winkler method:

Table 2. Black Box Testing

No	Test Case	Input	Expectation	Actual Output
1	Input document name search according to the database	"KK Bella Cover Letter"	Documents appear with high accuracy (≥ 0.90)	Document appears with a score of 1.0
2	Search Input with typos	"KK Bela Cover Letter"	The system still finds documents with a score ≥ 0.90	Document appears with a score of 0.997 or 1.0
3	Input document name is empty	-	The system displays the entire document	All documents appear
4	Input using all lowercase letters	"letter of introduction from kk bella"	The system still recognizes even if the capitalization is not correct because it	Document appears with a score of 0.997 or 1.0

No	Test Case	Input	Expectation	Actual Output
			counts the string.	
5	Input a combination of name and document type	Cover Letter KK Bella” + “Cover Letter”	The system displays documents according to the input combination.	Document found
6	Input documents that have similarities to more than one document in the database	Transfer Certificate	The system displays several documents that have a high level of similarity, sorted by the highest similarity score.	The system displays a list of relevant documents sorted by score

4. CONCLUSION

This study developed and tested a web-based administrative document search system using the Jaro-Winkler method to address inefficiencies in the existing system at Cipta Karya Village. The system significantly improved both speed and accuracy of searches, achieving similarity scores up to 0.9961, even when users input incorrect or incomplete document names. By accommodating errors and presenting relevant results, the system enhanced both staff efficiency and citizen satisfaction. Future research may explore integration with mobile platforms and broader administrative functions to further improve service delivery.

REFERENCES

- Glenn Tambahani, A., Sanjaya, A., & Widodo, D. W. (2021). Application of Jaro-Winkler Method to Find Word Similarity in English Pronunciation Test Application. *National Seminar on Technology Innovation UN PGRI Kediri*, 98–103.
- Syahputra, I., & Syakti, F. (2022). Comparison of Levenshtein and Jaro Winkler Algorithms on Information Systems for Searching Legislative Documents (Case Study: Diskominfo Lahat). *Smatika Jurnal*, 12(02), 176-186. <https://doi.org/10.32664/smatika.v12i02>.

- Salim, R. A., Septian, M. R. D., Suhartini, S., Anggraini, D., & Qomariyah, Q. (2021). Document Similarity Detection Application Using Jaro Winkler and Levenshtein Distance Algorithms. *Sebatik*, 25(1), 35-41. <https://doi.org/10.46984/sebatik.v25i1>.
- Rachman, F. I., Alam, N., Wahyuni, T., & Anas, L. (2024). Implementation of Spelling Correction on the Faculty of Engineering Reading Room Semantic Web Using the Jaro-Winkler Distance Algorithm. 12(1).
- Sajaya, A. (2020). Data Search Optimization Using Text. *NERO Scientific Journal*, 5(1), 24-29.
- Pinajeng, I. K. T. P., Sukarsa, I. M., & Putra, I. M. S. (2020). Word Correction in Chatbot Systems Using the Jaro Winkler Method. *Scientific Journal of Technology and Computers*, 1(2), 86-95.