Vol. 10 No. 2 September 2022
PIKSEL
Penelitian Ilmu Komputer
Sistem Embedded and Logic

Penelitian Ilmu Komputer, Sistem Embedded and Logic
p-ISSN: 2303-3304, e-ISSN: 2620-3553
Vol. 10 (2): 143 – 156 (September 2022)
https://doi.org/10.33558/piksel.v10i2.5606

Web-Based Mail Search Using the Levenshtein Distance Algorithm

Mayora Lolly Ishimora [1], Herlawati [1,*], Sri Rejeki [1]

* Corespondence Author: e-mail: herlawati@dsn.ubharajaya.ac.id

[1] Informatics; Universitas Bhayangkara Jakarta Raya; Jl. Raya Perjuangan No. 81 Margamulya, Bekasi utara, Bekasi; Telp. (021) 88955882; e-mail: mayora.lolly@gmail.com, herlawati@dsn.ubharajaya.ac.id sri.rejeki@dsn.ubharajaya.ac.id

Submitted : **19/08/2022**
Revised : **27/08/2022**
Accepted : **12/09/2022**
Published : **27/09/2022**

Abstract

The use of information technology in filing letters has been widely applied, especially in government institutions. In the Administrative Affairs of the Police Information Technology Bureau, the archiving of incoming and outgoing letters is still done manually, by utilizing records in a large agenda book and stored in a large folder. The absence of a system capable of assisting mail processing causes work to be inefficient, especially in terms of searching for letters. The purpose of this research is to produce a website that can help search letters by applying the Levenshtein Distance Algorithm. The design begins with needs analysis, design, implementation, verification (system testing), operation and maintenance. A website that can search letters by applying the Levenshtein Distance Algorithm was generated. This website is very helpful in the process of searching for letters to the Administrative Affairs of the Police Information Technology Bureau.

Keywords*: letters, letter archives, levenshtein distance, searching method*

1. Introduction

The use of technology in one of the law enforcement agencies in Indonesia, Indonesian National Police department (Polri) contributes to solving various problems that exist within the scope of the Police itself, e.g., in the implementation of filing letters. The letter, which is said to be an official document in the National Police, also plays an important role as a form of delivering important written instructions, orders, and directions (Peraturan Kapolri No 7, 2017).

One of the divisions in the National Police Headquarters is the Information and Communication Technology Division, called Div ICT Polri. It is one of the work units that has the duties and functions to carry out management, guidance, and development, as well as supervision of the Technology, Information and Communication (ICT) within the Police. The ICT Division of the Police consists of bureaus that have a role to support the duties of the Head of the ICT Division of the Police, namely the Communication Technology Bureau (Rotekkom) and the

Information Technology Bureau (Rotekinfo). In each bureau, there are administrative affairs that have a role to regulate, run, and distribute the flow of incoming and outgoing mail (Peraturan Kapolri No 21, 2010).

Administrative affairs, hereinafter abbreviated as Urtu, is an implementing unit related to incoming and outgoing letters at a work unit at the National Police Headquarters (Peraturan Kapolri Nomor 17, 2007). Receiving and storing letters at the Urtu Rotekinfo Div ICT Polri is done manually in its implementation. Writing agendas for incoming and outgoing letters is still using the ledger. The use of ledgers as a form of recording incoming and outgoing letters causes inefficiency of members in carrying out their work, especially in terms of searching for letters. The search for incoming mail that still must open the ledger, as well as the search for outgoing mail archives which still require opening a large folder to get to the physical file, causes members to have difficulty in their work.

To help members in searching for incoming and outgoing mail archives, the application of the Levenshtein Distance Algorithm is expected to be able to provide time efficiency for members in searching for letters. The Levenshtein Distance algorithm is an algorithm that works by measuring the distance between the number of string differences between two (2) strings. The smaller the distance between two (2) strings, the higher the level of similarity that will be obtained (Gilleland & Park Software, 2006).

In previous studies, the Levenshtein Distance Algorithm which was applied to perform data searches could be carried out and succeeded in finding the desired data. Researchers built a website-based information system about managing mail archives in Seberang Ulu II District and can prove that the Levenshtein Distance Algorithm can be applied to the data search process based on the number of string differences between two (2) strings (Vidyarsih dkk., 2016).

Another study conducted by (Sadiah dkk., 2019) focused on the use of the Levenshtein Distance Algorithm in e-dictionary medicines by applying query suggestion. The application of this algorithm is intended to obtain information about drugs based on a search from a string in the database. In this study, the Levenshtein Distance Algorithm and its operation function was to add, subtract, and substitute a character string, where the results will be displayed in the form of a query suggestion, e.g., "Do you mean: paraco ?". The results of the application prove the accuracy rate up to 90% with 90% precision and 90% recall. It can be concluded

from this study that the application of the Levenshtein Distance Algorithm to search for drugs in the e-Dictionary application by applying query suggestion can be done with a high level of accuracy.

The next previous study discussed search optimization on the information system dashboard owned by STIKOM Bali by applying the Levenshtein Distance Algorithm. This research was conducted to be able to search the data in the database owned by STIKOM Bali by entering the wrong data. therefore, a suggestion string will appear on the dashboard display to be seen by the user, whether it is as desired or not. The results of this study indicate that the use of the Levenshtein Distance Algorithm is proven to increase word search with accuracy up to 66% of the 100 words tested, where when the dashboard does not apply this algorithm it only has an accuracy rate of 9% in word search (Sumiari dkk., 2019).

Based on the previous studies, to help users in searching for letters at the Urtu Rotekinfo Div ICT Polri, it is necessary to create a mail archiving system by applying the Levenshtein Distance Algorithm as a form of search implementation.

2. Data and Methods

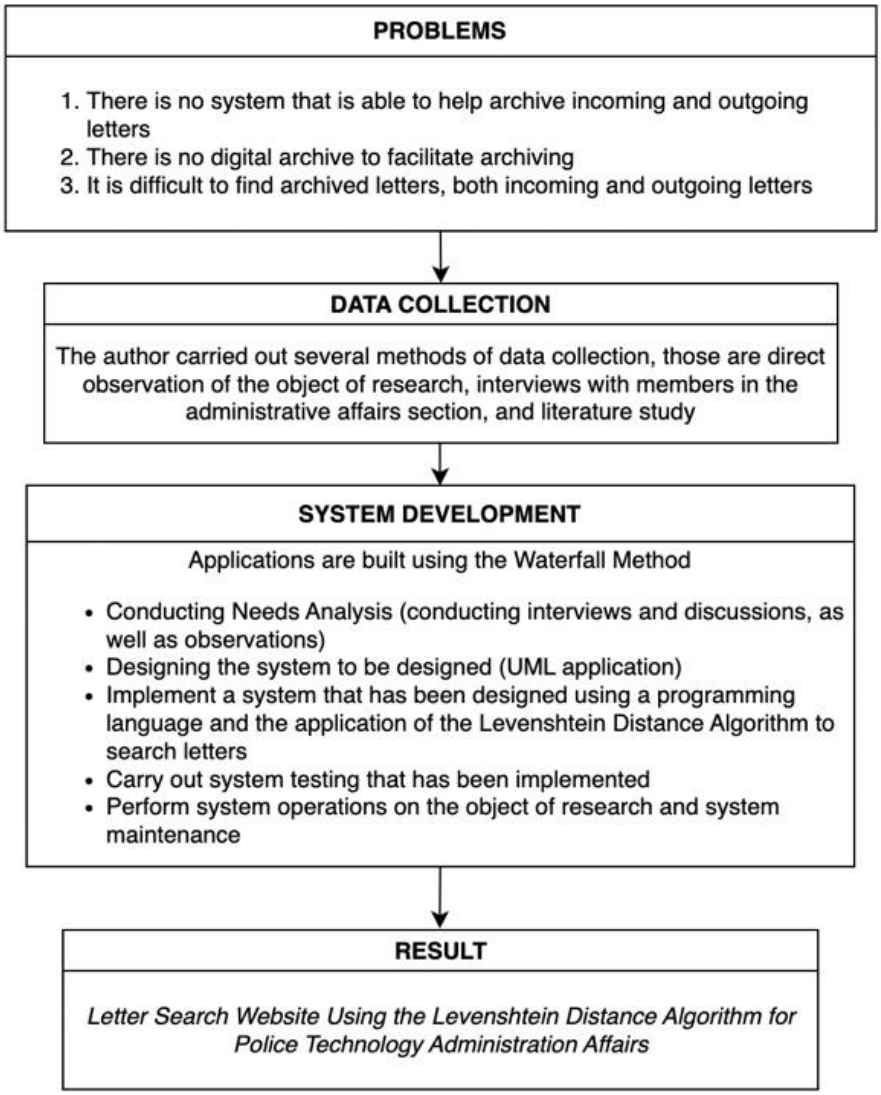
This subsection will discuss the Levenshtein Distance Algorithm and data collection.

2.1. Research Framework

In this research framework, it will describe the sequence of research so that the results of the research are in accordance with the objectives intended in the previous introduction. This research framework is expected to provide an overview of website-based system design in searching for letters using the Levenshtein Distance Algorithm. The research framework can be seen in Figure 1.

2.2. Data Collection

In this study, several data collection methods were carried out, including observations on the object of research, namely Administrative Affairs of the Information Technology Bureau at the Information Technology and Communications Division of the Police, interviews with relevant sources in order to obtain accurate data and in accordance with reality, as well as conducting a literature study with conduct a search and review of literature on books, journals, and articles that support research and are related to the theme raised.



Source: Research Result

Figure 1. Research Framework

2.3. Levenshtein Distance Algorithm

The Levenshtein Distance algorithm was first coined by a scientist from Russia named Vladimir Levenshtein in 1965. The Levenshtein Distance algorithm can also be referred to as edit distance or edit distance is a matrix used to calculate the minimum distance difference between two strings (Gilleland & Park Software, 2006).

The distance calculation applied in this algorithm determines the minimum number of change operations between two strings capable of converting the first string into the second string. Mathematically, the Levenshtein distance between these two strings a, b has the following equation (Qibti Da'iyah, 2019):

$$if\ \min(i,j)\ =\ 0$$

$$Lev_{a,b}(i,j) = \begin{cases} max(i,j) \\ min\begin{cases} lev_{a,b}(i-1,j)\ +\ 1 \\ lev_{a,b}(i,j-1)\ +\ 1 \\ lev_{a,b}(i-1,j-1)\ +\ 1_{(a_i \neq b_j)} \end{cases} \end{cases} \quad (1)$$

Where:

a = string (1)

b = string (2)

i = index of character position (1)

j = index of character position (2)

The operations in the Levenshtein Distance Algorithm include operations for adding characters (insertion), deleting characters (deletion), and swapping characters (substitution). The use of this algorithm starts from the upper left corner of the array, a two-dimensional string filled with characters from the first string and the second string which is the target string and will be assigned a cost value. The cost value in this algorithm is the number of operations performed by the algorithm that shows the Levenshtein distance and is in the lower right corner. If a character and the number of characters in the initial string are equal to the target string, then no operation will be performed on the string or it can be said to be "0" (Theodosius Nainggolan, 2020).

The results of the Levenshtein Distance Algorithm calculation will then be calculated into a similarity calculation, to get the percentage of similarity of search words to words stored in the database. Equation 2 shows the similarity formula (Fajar Dewantara, 2013):

$$similarity\ =\ \left(1 - \frac{levenshteinDistance(s1,s2)}{max(|s1|,|s2|)}\right) \times\ 100\% \quad (2)$$

Were,

s1 = word input by the user

s2 = word stored in the database

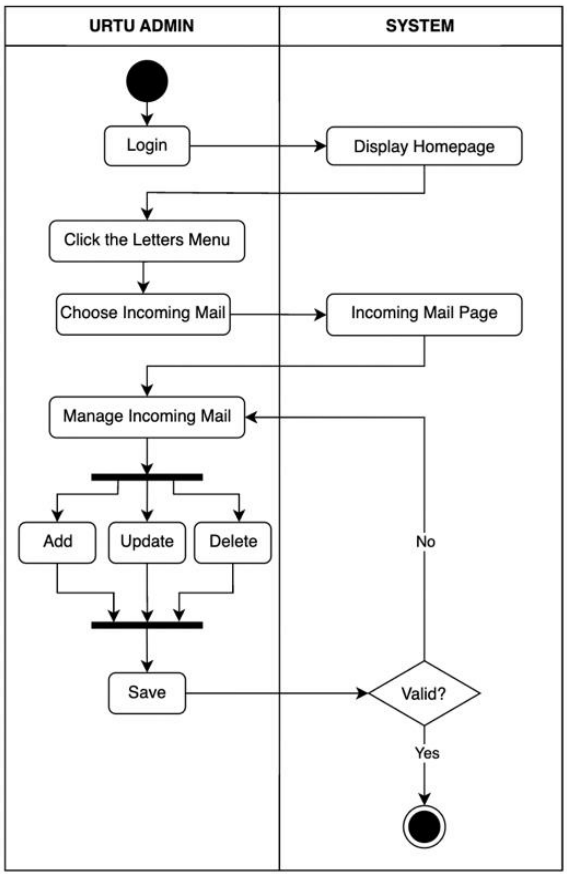
3. Results and Analysis

Based on the problems, it is necessary to build a system that is expected to be able to help staff members of the Urtu Rotekinfo ICT Polri in managing incoming

and outgoing letters, as well as being able to streamline the search time for mail archives based on search words in terms of letters using the Levenshtein Distance Algorithm.

3.1. System Design

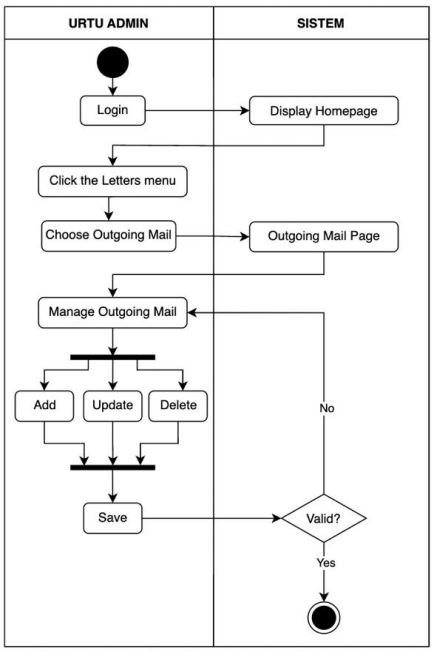
The following is the design of the Activity Diagram of the incoming mail management system in Figure 2, outgoing mail in Figure 3, and the search for letters based on existing words in the subject of the letter in Figure 4.



Source: Research Result

Figure 2. Activity Diagram of Manage Incoming Mail

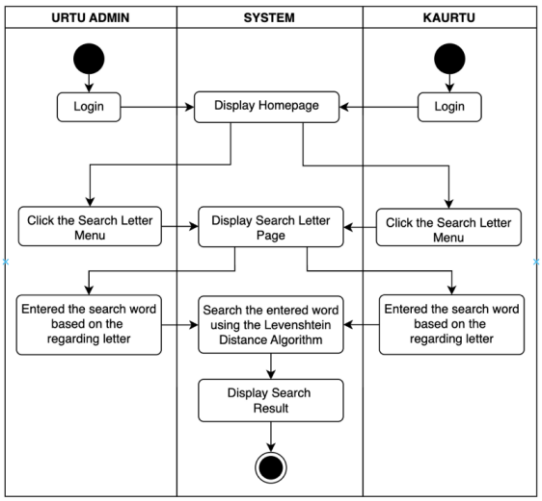
The explanation of the Activity Diagram for Incoming Mail Management in Figure 2 is as follows: (1) Admin logs into the system; (2) The system displays the homepage display; (3) Admin selects the mail menu; (4) Admin selects the incoming mail section; (5) Admin manages incoming mail data (create, read, update, delete); (6) The system validates the entry of incoming mail, if appropriate, the data will be stored in the database, otherwise it will return to display the data entry page if the data does not comply with the provisions; (7) The system has successfully saved incoming mail data.



Source: Research Result

Figure 3. Activity Diagram of Manage Outgoing Mail

The explanation of the Activity Diagram for Managing Outgoing Mail in Figure 3 is as follows: (1) Admin logs into the system; (2) The system displays the homepage display; (3) Admin selects the mail menu; (4) Admin selects the outgoing mail section; (5) Admin manages outgoing mail data (create, read, update, delete); (6) The system validates the entry of outgoing mail, if appropriate, the data will be stored in the database, otherwise it will return to display the data entry page if the data does not comply with the provisions; (7) The system has successfully saved outgoing mail data.



Source: Research Result

Figure 4. Activity Diagram of Searching for Letters

The explanation of the Letter Dancing Activity Diagram in Figure 4 is as follows: (1) Urtu and Kaurtu admins log into the system; (2) The system displays the homepage display; (3) Urtu and Kaurtu admins select the mail search menu; (4) The system displays the mail search page display; (5) Urtu and Kaurtu admins enter search terms based on the subject of the letter; (6) The system searches for letters using the Levenshtein Distance Algorithm; (7) The system displays the mail search results.

3.2. Appliaction of the Levenshtein Distance Algorithm

In this study, the Levenshtein Distance Algorithm is used to calculate the minimum edit distance between the string entered in the letter search column and the subject of the letter stored in the database.

a. Distance Calculation

In this section, the distance calculation uses the formula described earlier. An example of a case that can be applied is to calculate the distance between the strings "HOSTING" and "HOSTIN". Application of calculations using a matrix table.

Table 1. Matrix Operation Table ‘’HOSTIN

		H	**O**	**S**	**T**	**I**	**N**	**G**
	0	1	2	3	4	5	6	7
H	1	0						
O	2							
S	3							
T	4							
I	5							
N	6							

Source: Research Result

The calculation starts from the top left corner of the matrix, i.e. d(1,1), where d(0,1) has the character string 'H' which has the same value as the character string in d(1,0), so nothing changes , then the value is 0. At d(1,2), it can be calculated by determining the minimum value of the 3 operations performed, namely by the formula d(1,2) = min(d(0,1)+1, d(0, 2)+1, d(1,1)+1). The results of the calculation of at least 3 operations are 1. The calculation is carried out so on until the table in d(6.7) finds the results of the Levenshtein distance which will be

calculated into the similarity formula. The results of the calculation of the Levenshtein distance can be seen in table 2.

Table 2. Matrix Operation Table ''HOSTIN

		H	**O**	**S**	**T**	**I**	**N**	**G**
	0	1	2	3	4	5	6	7
H	1	0	1	2	3	4	5	6
O	2	1	0	1	2	3	4	5
S	3	2	1	0	1	2	3	4
T	4	3	2	1	0	1	2	3
I	5	4	3	2	1	0	1	2
N	6	5	4	3	2	1	0	1

Source: Research Result

b. Similarity Calculation

After calculating the Levenshtein distance, the distance results will be entered into the similarity formula. The use of the similarity formula in this study serves to obtain search results for words that have similarities with words in the subject of letters stored in the database.

The following is the result of calculating similarity based on the Levenshtein distance obtained based on the example case above:

$$similarity = \left(1 - \frac{1}{7}\right) \times 100\%$$

$$similarity = \left(\frac{6}{7}\right) \times 100\%$$

$$similarity = 85{,}71\%$$

The following are the results of a similar calculation on 30 words that were searched on the system that had been built:

Table 3. Similarity Calculation

No.	Source Word	Target Word	Word Length	Distance	Similarity
1.	Permohonan	Permohonan	10	0	100%
2.	Permohonan	Permohonan	10	1	90%
3.	Pemasukan	Pemasukan	9	2	88,89%
4.	Pemasukan	Pemasukan	9	2	77,78%
5.	Laporan	Laporan	7	2	71,43%

No.	Source Word	Target Word	Word Length	Distance	Similarity
6.	Laporan	Laporan	7	3	57,14%
…					
59.	Personell	Personel	9	1	88,89%
60.	Personel	Personel	8	2	75%
Average Value of the Similarity					**70%**

Source: Research Result

Based on the table of similarity calculation results above for 60 source words entered in the search column with different conditions for each word with a Levenshtein distance between 0 to 5, it can be obtained an average similarity value of 70%. The calculation of the similarity value is closely related to the maximum word length between the words entered and the words being searched for. While the resulting Levenshtein distance value is not affected by the length of the word entered but is related to how many operations in the algorithm are used, namely the addition, subtraction, and exchange or substitution operations that are applied to the entered word. The smaller the resulting levenshtein distance, the more similar the word in terms of the letter will be displayed.

The following are the results of the search for search terms from the subject of letters stored in the database.

Table 4. Search Word Rediscovery Result

No.	Search Word	Input Word	Found	Not Found
1.	Permohonan	Permohonan		
2.	Pemasukan	Pemasukan		
3.	Laporan	Laporan		
4.	Surat	Surat		
5.	Penyampaian	Penyampaian		
6.	Protokoler	Protokolerrr		
…				
60.	Surat Kehilangan	Surat Kehilangan		

Source: Research Result

From the table of recoveries of 60 search words entered into the search column, there were 3 search words that were not found according to the words you wanted to find. This is because the value of similarity is small and related to the operations applied in the Levenshtein Distance Algorithm which can change the equivalent words in the database according to those entered in the search field.

c. Interpreation and Result

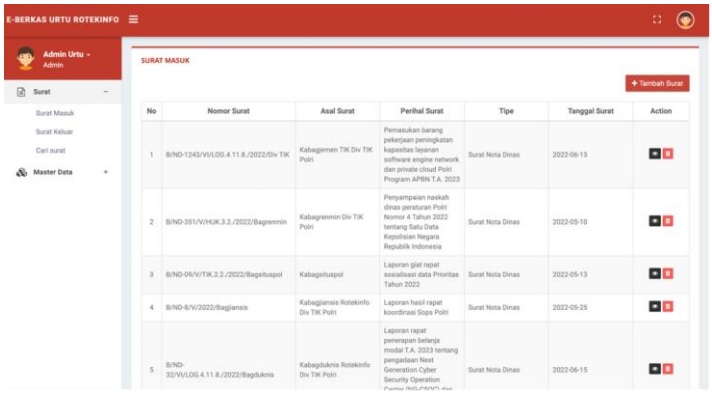
Based on the calculation of the similarity results in table 3 above, letters that have the highest level of similarity will appear from the highest to the lowest, according to the words entered in the letter search section.

3.3. System Implementation

After doing the modeling process, then the system design that has been made will be implemented by coding the program. The following is the interface of the mail search website using the Levenshtein Distance Algorithm based on the words contained in the subject of the letter.

a. Incoming Mail Page View

Figure 5 is a display of the incoming mail page. This menu shows that admin can add, edit, or delete incoming mail data.

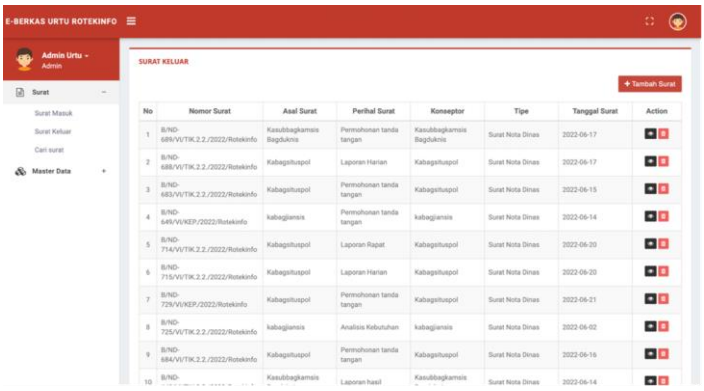


Source: Research Result

Figure 5. Incoming Mail Page View

b. Outgoing Mail Page View

Figure 6 is a view of the outgoing mail page. In this menu Admin can add, edit, or delete outgoing mail data.

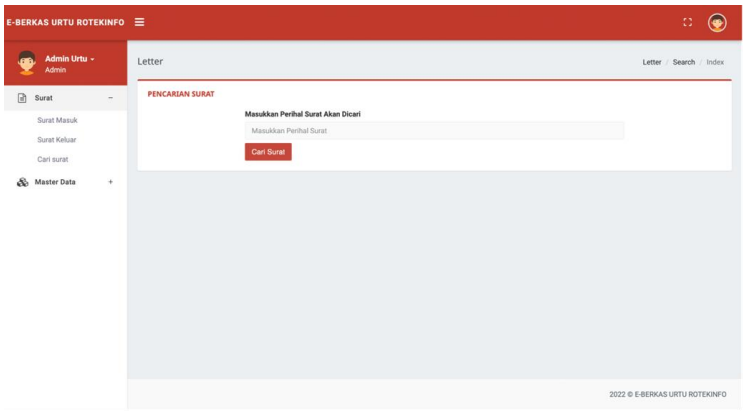


Source: Research Result

Figure 6. Outgoing Mail Page View

c. Mail Search Page View

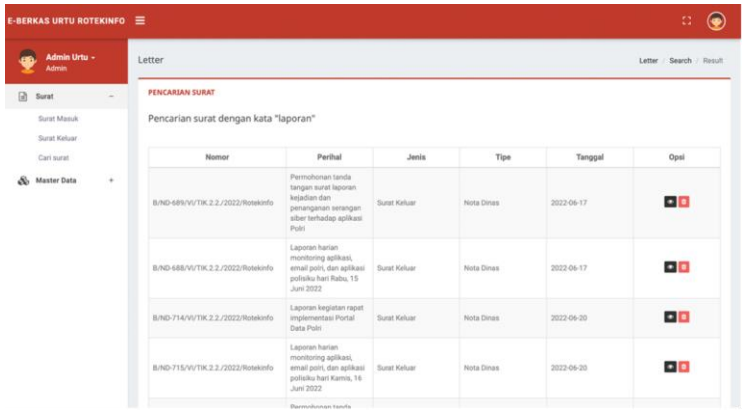
Figure 7 is a display of the mail search page, where admin and Kaurtu can access the page.



Source: Research Result

Figure 7. Mail Search Page View

The display of the letter search results by applying the Levenshtein Distance Algorithm in Figure 8. The search word is taken from the words contained in the subject of the letter.



Source: Research Result

Figure 8. Mail Search Page View

4. Conclusion

Levenshtein Distance Algorithm can be applied to search letters based on the words contained in the subject of the letter. Searching based on the words contained in the subject of the letter, is closely related to the distance and also the level of similarity between words. The smaller the distance calculation results using the Levenshtein Distance Algorithm, the more accurate the results found and displayed by the system.

Acknowledgements

The authors thank the Administrative Affairs of the Information Technology Bureau, the Information and Communication Technology Division of the Indonesian National Police for granting permission and assisting in this research process. Also, LPMPP Bhayangkara University, Greater Jakarta as a facilitator and observer who has provided interesting and constructive comments.

Author Contributions

Mayora Lolly Ishimora proposed the topic; Mayora Lolly Ishimora, Herlawati and Sri Rejeki conceived models and designed the experiments; Mayora Lolly Ishimora, Herlawati and Sri Rejeki conceived the optimisation algorithms; Herlawati and Sri Rejeki analysed the result.

Conflicts of Interest

The author declare no conflict of interest.

References

Fajar Dewantara, M. (2013). *Mesin Pencari Kata Pada Terjemahan Al-Quran Dengan Menggunakan Metode Algoritma Levenshtein* [Skripsi Tugas Akhir]. Universitas Brawijaya.

Gilleland, M., & Park Software, M. (2006). *Levenshtein Distance*. Diambil 9 April 2022, dari https://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Fall2006/Assignments/editdistance/Levenshtein%20Distance.htm

Peraturan Kapolri No 7. (2017). *Peraturan Kapolri No 7 Tahun 2017*. Kepolisian Negara Republik Indonesia.

Peraturan Kapolri No 21. (2010). *Peraturan Kapolri No 21 Tahun 2010* (hlm. 81). Kepolisian Negara Republik Indonesia.

Peraturan Kapolri Nomor 17. (2007). *Peraturan Kepala Kepolisian Negara Republik Indonesia Nomor 17 Tahun 2007*.

Qibti Da'iyah, S. (2019). *Implementasi Algoritma Levenshtein Distance Dalam Sistem Informasi Pengarsipan Surat Perkantoran Berbasis Web (Studi Kasus: Bappeda Oku Sumsel)* [Institut Teknologi Yogyakarta]. http://eprints.uty.ac.id/2724/

Sadiah, H. T., Saad Nurul Ishlah, M., & Najwa Rokhmah, N. (2019). Query Suggestion on Drugs e-Dictionary Using the Levenshtein Distance Algorithm. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, 193. https://doi.org/10.24843/lkjiti.2019.v10.i03.p07

Sumiari, N. K., Ketut, N., Ari, D., & Lis, J. (2019). Optimasi Dashboard Information System STIKOM Bali dengan Algoritma Levenshtein Distance. *Citec Journal*, *6*(1), 12–26.

Theodosius Nainggolan, A. (2020). *Mesin Pencarian Judul Tugas Akhir Berbahasa Indonesia*. Universitas Sanata Daharma.

Vidyarsih, P., Andretti Abdillah, L., Muzakir, A., & Ahmad Yani No, J. (2016). *Seminar Hasil Penelitian Sistem Informasi dan Teknik Informatika ke-2 (SHaP-SITI2016) Sistem Informasi Pengarsipan Menggunakan Algoritma Levensthein String pada Kecamatan Seberang Ulu II*.