Implementation of Levenshtein Distance Algorithm in The Digital Biology Dictionary Search Function

Khalidah
Master of Computer Science Study Program, Faculty of Information Technology, Budiluhur University
Email: halidah050@gmail.com

Article Info

Article history:
Received Aug 28th, 2021
Revised Sept 1st, 2021
Accepted Sept 30th, 2021

Keyword:
Algorithm
Biology digital dictionary
Levenshtein Distance
Query suggestion
Searching Function

ABSTRACT

Digital biology dictionaries are very important to develop because they can help biology students, laboratory assistants, and general users in searching for biology terms. In the biology dictionary, there are biological terms in Latin, which sometimes users mistype in the query on the term search form. Therefore, it is important to implement the Levenshtein distance algorithm to provide query suggestion information to users. This study aims to implement the Levenshtein distance algorithm in the digital biology dictionary search function. The research stages, namely the development of a search module on the digital biology dictionary, implementation of the Levenshtein Distance algorithm, query suggestion validation. The Levenshtein distance algorithm has been successfully implemented in the digital biology dictionary by providing query suggestion output for typing errors. Meanwhile, based on the test results, the system can evaluate words with the query suggestion function with an accuracy value of 90%.

Corresponding Author:
Khalidah,
Master of Computer Science Study Program,
Budiluhur University,
Jl. Ciledug Raya No.99, Pertukangan Utara, Jakarta
Email: halidah050@gmail.com

DOI: http://dx.doi.org/10.24014/ijaidm.v4i2.13989

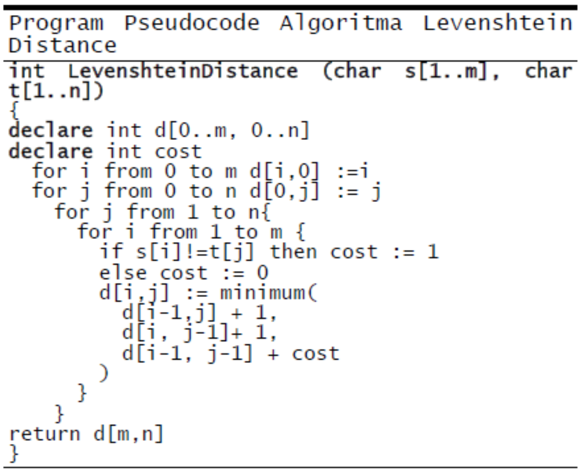
1. INTRODUCTION

Biology dictionaries are needed by students, laboratory assistants, or researchers who want to know the descriptions and terms of biology. However, the physical biology dictionary is thick enough to make it impractical to search. Therefore, it is important to build a digital biology dictionary. In the digital dictionary, there is an important function that must be developed, namely the search function. The search function is critical because this function is the main function in a digital dictionary where users will start searching for terms by utilizing the search function [1],[2],[3]. The number of biological terms and using Latin causes users to sometimes make mistakes in typing biological terms in the search function. This makes the system not find search results that match the query entered by the user. Users will be disappointed. Therefore, it is important to build a query suggestion feature to assist users in informing query suggestions. Query suggestion is a feature to provide information in the form of query suggestions to the user when the user makes a typo in the search form [4],[5],[6],[7]. This feature can improve the quality of system attributes or usability, especially in the search form [1],[8],[9],[10],[11]. This feature can be a solution for users who mistype a query on the digital biology dictionary search form. Levenshtein distance algorithm is an algorithm to find the distance between the words entered by the user and the words contained in the database. The algorithm will calculate the number of differences between the two strings in the form of a matrix [12],[13],[14]. Calculations are represented in the form of a matrix or table of Levenshtein distance calculations where the results will be seen in the lower right corner [j][k][l]. The final result of the Levenshtein distance algorithm shows the number of operations to be performed. The operations contained in the Levenshtein distance algorithm are substitution operations, deletion operations, and addition operations [15],[16],[17]. Sadiah et al. (2019) has succeeded in building a query suggestion feature in drug e-dictionary [10]. This study aims to implement the Levenshtein distance algorithm in the digital biology dictionary search function. This research is expected to make it easier for users to find descriptions of biological terms in a digital biology dictionary.

2. RESEARCH METHOD

The method or stages of this research consist of building a search module on a digital biology dictionary, implementing the Levenshtein Distance algorithm, and validating query suggestions.

1. Development of the Search Module on the Digital Biology Dictionary
   The development of a digital biology dictionary search function uses the SDLC (System Development Life Cycle) method. SDLC stages, namely analysis, design, implementation, and testing [18],[19]. At the analysis stage, the system functionality requirements are collected and the system analysis to be developed is identified [20][21]. The data entered into the system are 505 terms. The inputted term data is sourced from the physical biology dictionary [22].
   Then do the search form design. After that, the implementation phase was carried out using the PHP-MYSQLi programming language and testing [23][24][25].
2. Implementation of the Levenshtein Distance Algorithm
   At this stage, the implementation of the Levenshtein distance algorithm is carried out using the PHP programming language. The pseudocode in Figure 1 is converted to the PHP programming language. Then the algorithm is retrieved on the search module



```
Program Pseudocode Algoritma Levenshtein
Distance
int LevenshteinDistance (char s[1..m], char
t[1..n])
{
declare int d[0..m, 0..n]
declare int cost
  for i from 0 to m d[i,0] :=i
  for j from 0 to n d[0,j] := j
    for j from 1 to n{
      for i from 1 to m {
        if s[i]!=t[j] then cost := 1
        else cost := 0
        d[i,j] := minimum(
          d[i-1,j] + 1,
          d[i, j-1]+ 1,
          d[i-1, j-1] + cost
        )
      }
    }
return d[m,n]
}
```

Figure 1. Pseudocode Algorithm of Levenshtein Distance [10]

3. Validate Query Suggestion
   The search function that has been implemented and the query suggestion feature created on the search form is then tested for validation. The validation test is done by using a Test scenario. Test scenarios can be seen in Table 1.

Table 1 Scenario of Search Form Validation Test

No	Test Scenarios	*Query Suggestion*	Validation
1	User input Query on the search form by writing the correct biological terms		
2	User input Query where the input is incorrect, i.e. missing letters		
3	User input Query where the input is incorrect, i.e. excess letters		
4	User input Query where the input is wrong, i.e. wrong letter		
5	User input Query where the entered query is not in the database		

3. RESULTS AND DISCUSSION

In the first stage, an analysis of the system to be developed is carried out. At this stage, the specific system flow in the search function is identified. The analysis of the system to be developed can be seen in Figure 2. The design of the search system that has implemented the Levenshtein distance algorithm is presented in Figure 3.

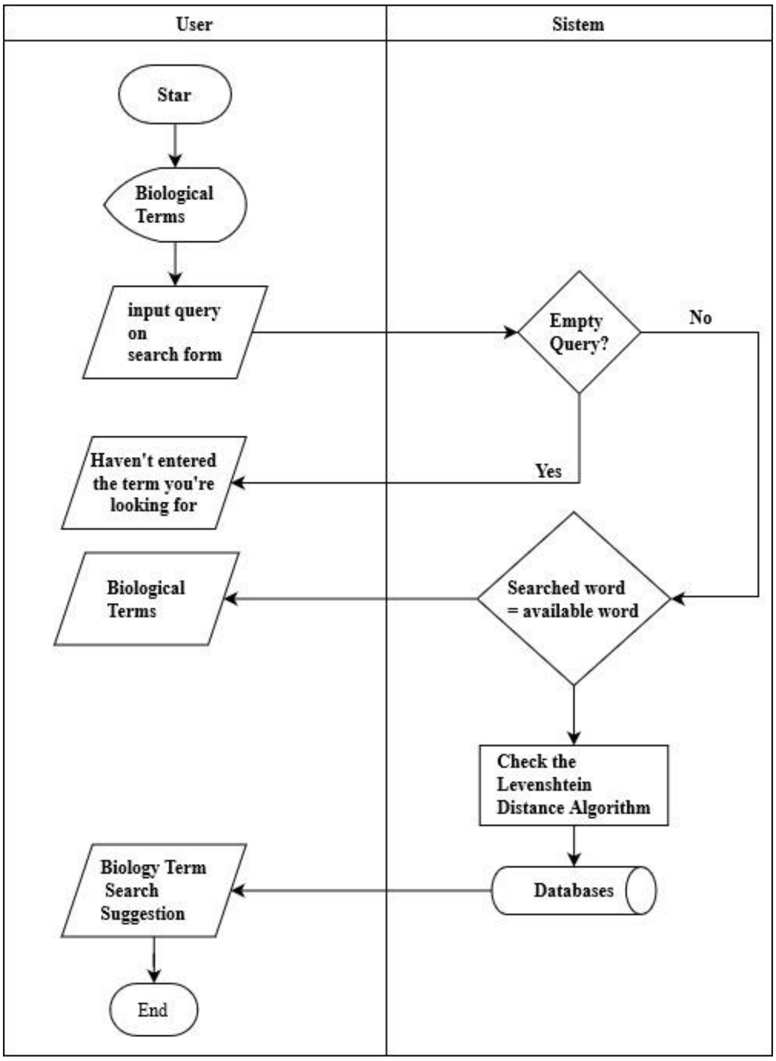

User
Sistem
Star
Biological Terms
input query on search form
Empty Query?
No
Yes
Haven't entered the term you're looking for
Biological Terms
Searched word = available word
Check the Levenshtein Distance Algorithm
Databases
Biology Term Search Suggestion
End

Figure 2. Analysis of the system to be developed

Figure 2 can be described as follows:

a. Users access the digital biology dictionary application
b. Users search for biological terms on the search form
c. If the user input query is empty, the system will display a notification "not yet entered the search term"
d. If the user inputs a query for biological terms contained in the database, the system will display the search results.
e. If the user inputs a biological term query but it is not available in the database, the system will check with the Levenshtein distance algorithm and display a query suggestion along with the results.

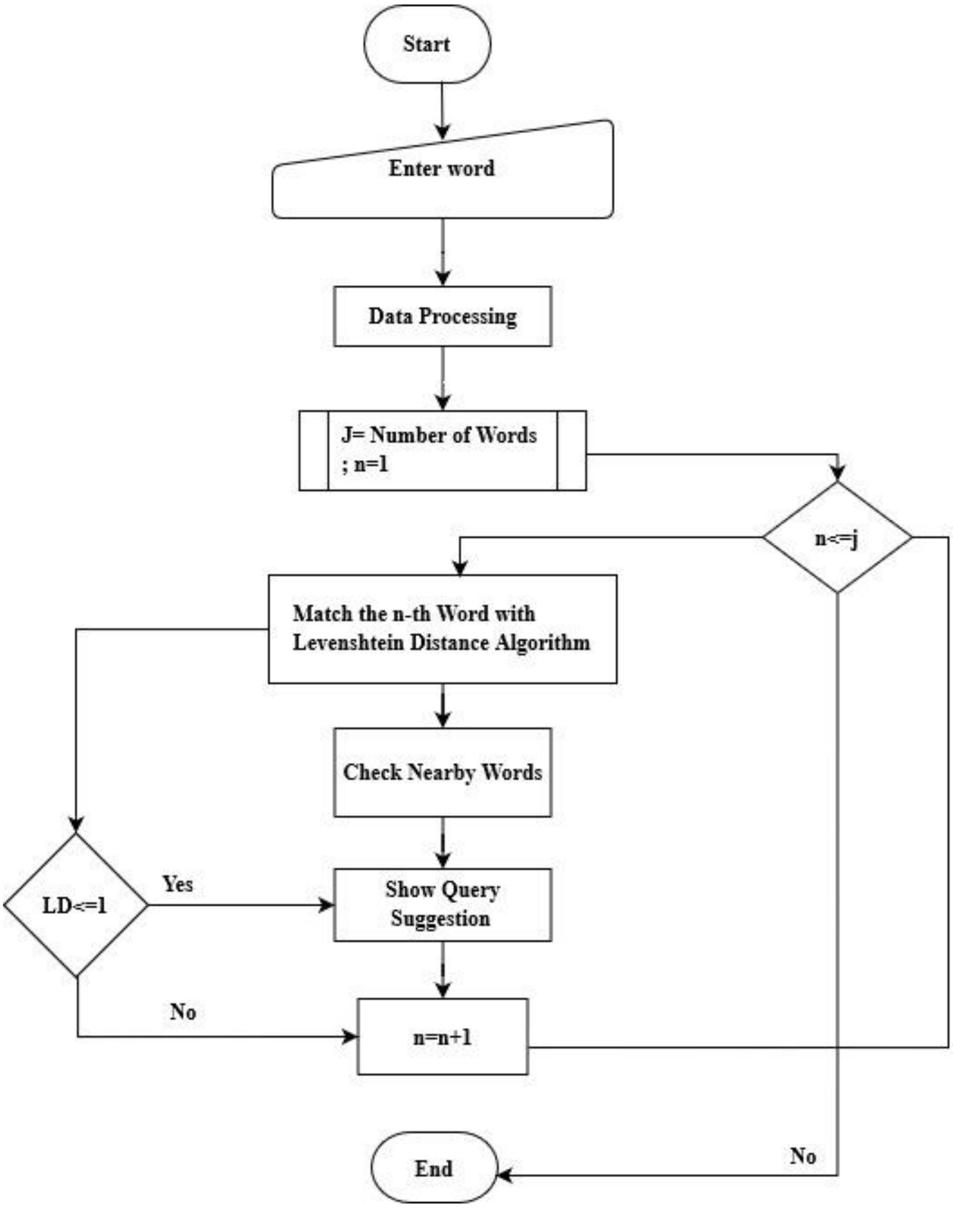
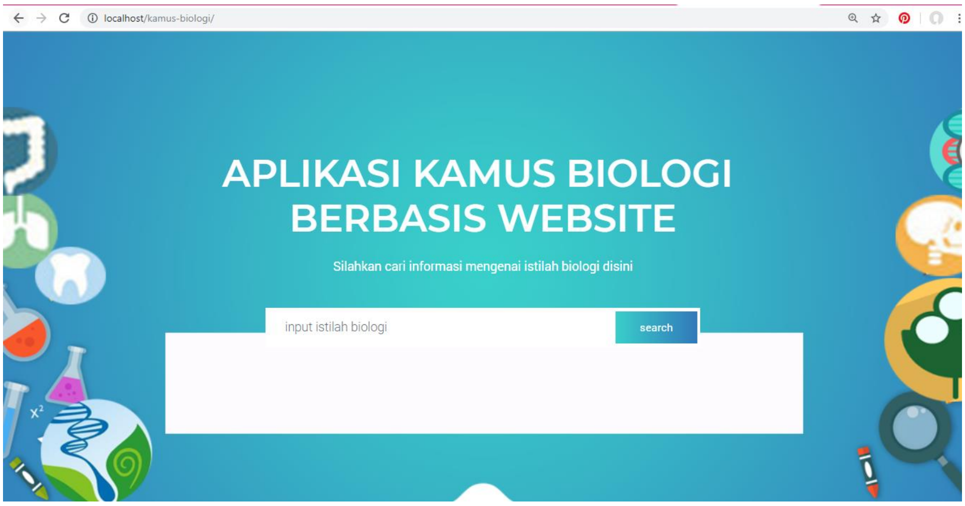


Figure 3. The design of the search function implemented by the Levenshtein distance algorithm

Figure 3 is a search function design implemented by the Levenshtein distance algorithm which can be described as follows:

a. The user enters a word in the search form, then the word entered by the user becomes data preprocessing.
b. Preprocessing data is data that is processed using the Levenshtein distance algorithm.
c. Where the number of words = j and the initialization value of n = 1
d. If n<= j then the system will match the n^{th} word with the Levenshtein distance algorithm
e. The process is checked simultaneously, namely, the words that are attached, the algorithm Levenshtein distance <= 1 if the displayed word is the same. Otherwise n=n+1 and return to the word search position.
f. As for the Levenshtein distance algorithm > 1, the system will immediately return to match words.

After designing the system flow on the search function, the next step is to build a digital biology dictionary search function. At this stage, the pseudocode is implemented into the PHP programming language to create the query suggestion feature on the search form. The user main page can be seen in Figure 4.



localhost/kamus-biologi/

APLIKASI KAMUS BIOLOGI
BERBASIS WEBSITE

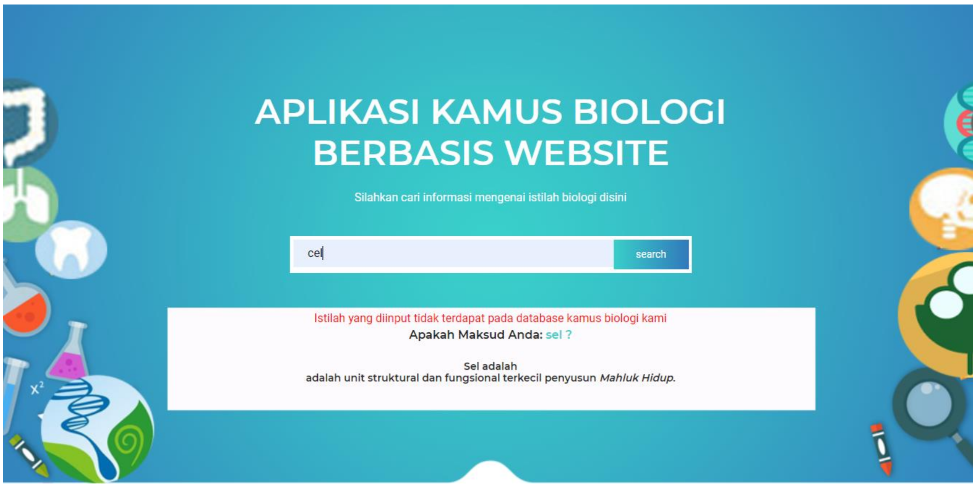
Silahkan cari informasi mengenai istilah biologi disini

input istilah biologi

search

Figure 4. User main page

On the user main page, there is a search form. The search form is used for the facility to search for biological terms. The search form that has been implemented by the Levenshtein distance algorithm can be seen in Figure 5.



APLIKASI KAMUS BIOLOGI
BERBASIS WEBSITE

Silahkan cari informasi mengenai istilah biologi disini

cel

search

Istilah yang diinput tidak terdapat pada database kamus biologi kami

Apakah Maksud Anda: sel ?

Sel adalah

adalah unit struktural dan fungsional terkecil penyusun *Mahluk Hidup.*

Figure 5. Query Suggestion Feature on the Search Form

In Figure 5 there is already a query suggestion feature on the search form. The user mistyped the query on the search form where it should be "Sel" but the user typed "cel". However, the system managed to provide a Query Suggestion with the sentence "Do you mean: sel ?". The computational process of the Levenshtein distance algorithm for the case of "cel" to "sel" can be seen in Figure 6.

Based on the Levenshtein Distance Algorithm Pseudocode in Figure 1, the computational process for the Levenshtein distance algorithm computation for the case of "cel" to "sel" is as follows:

m = User inputted word = CEL
n = Words in the database = SEL

d [0,0] = 0

Initialize the first row and first column: 0,1,2,. . . m 0,1,2,...n

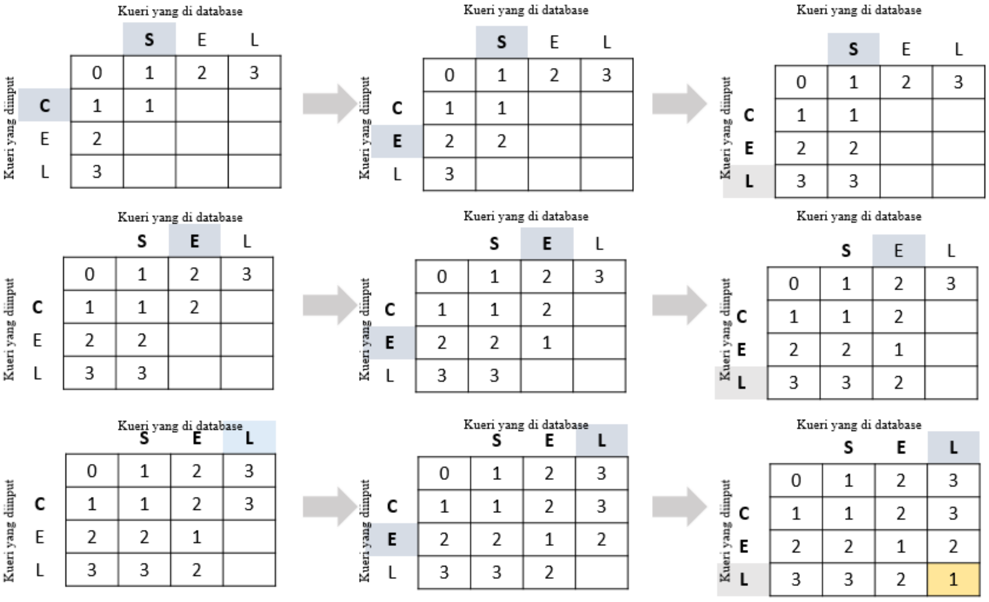
a. Check each character by character comparison
b. If the characters are the same then cost = 0
c. If the characters are different then cost = 1
d. Check the minimum value

	On	=	d[i,j]+1
d[i,j]	Side	=	d[i,j]+1
	Diagonal	=	d[i,j]+ cost

e. Compare C with S if different then cost = 1

	On = 1	
Check All Values d [i,j]	Diagonal = 0	Minimum diagonal
	Side = 1	

f. d [i,j] = d [i,j] + cos
= 0 + 1 = 1



		S	E	L
	0	1	2	3
C	1	1		
E	2			
L	3			

		S	E	L
	0	1	2	3
C	1	1		
E	2	2		
L	3			

		S	E	L
	0	1	2	3
C	1	1		
E	2	2		
L	3	3		

		S	E	L
	0	1	2	3
C	1	1	2	
E	2	2		
L	3	3		

		S	E	L
	0	1	2	3
C	1	1	2	
E	2	2	1	
L	3	3		

		S	E	L
	0	1	2	3
C	1	1	2	
E	2	2	1	
L	3	3	2	

		S	E	L
	0	1	2	3
C	1	1	2	3
E	2	2	1	
L	3	3	2	

		S	E	L
	0	1	2	3
C	1	1	2	3
E	2	2	1	2
L	3	3	2	

		S	E	L
	0	1	2	3
C	1	1	2	3
E	2	2	1	2
L	3	3	2	1

Figure 6. Computing the Levenshtein distance algorithm for the case of "cel" to "sel"

In Figure 6, it is known that the computational result of the Levenshtein distance algorithm in the lower right corner is 1. The result value of 1 is obtained from the minimum diagonal value with a cost value = 0. This shows that the operation to be performed is substitution by performing 1 letter substitution operation. Character substitution operation is an operation to swap a character with another character. In this case, the user inputs the query string "cel" in the search form, then the system will recommend the word to be "sel". In this case, the character "c" is replaced with the letter "s".

The search function that has been implemented by the Levenshtein distance algorithm is evaluated by the search form function based on the test scenario in Table 1.

Table 2. Validation Test Results of The Query Suggestion Feature

No	Test Scenario	*Query Suggestion*	Validation
1	User input Query on the search form by writing the correct biological terms Input: sel	-	Output: sel term and description of sel term Valid
2	User input Query where the input is incorrect, i.e. missing letters Input: Ace	Query Suggestion with the operation of adding letters Did you mean "Acerang"?	Output: Acerang term and description of acerang term Valid
3	User input Query where the input is incorrect, i.e. excess letters Input: abasiaa	Query Suggestion with letter deletion operation Did you mean "Abasia"?	Output: abasia term and description of abasia term

No	Test Scenario	*Query Suggestion*	Validation
4	User input Query where the input is wrong, i.e. wrong letter Input: asacia	Query Suggestion with character substitution operation Did you mean "acacia"?	Output: acacia term and description of acacia term Valid
5	User input Query where the entered query is not in the database Input: before	Query suggestions Take the closest word and the minimum number of operations Did you mean "biforate"?	Output: biforate term and description of the biforate term Valid

Based on the results of the evaluation of Tables 2 and 3, it is stated that the system can accept spelling errors of three operations at once even more on search input and generate query suggestions based on the correct proximity of letters contained in the database. In addition to the three operations tested, also tested for words that are not in the database. Based on the test results, the system has not been able to display the notification "the word is not in the dictionary", but the system displays the word closest to the spelling in the database or the most minimal operation. Evaluation of accuracy results is an evaluation of the accuracy of the Query Suggestion feature in the application of a biology dictionary. Based on the test results of 100 data from the three operations of the Levenshtein distance algorithm, an accuracy value of 90% is obtained. There are test data whose results do not match the test scenario, namely 10 data from 100 test data.

4. CONCLUSION

The system in the digital biology dictionary that has implemented the Levenshtein distance algorithm produces a query suggestion feature. The system can accept spelling errors of three operations at once or even more on search input and generate query suggestions based on the correct proximity of letters contained in the database. Meanwhile, based on the test results, the system can evaluate words with the query suggestion function with an accuracy value of 90%. Implementation of the Levenshtein distance algorithm on the search form can help users find the term and description of the biology they are looking for

REFERENCES

[1] Sadiah, H.T. Kajian Usability Website E-commerce Indonesia Berdasarkan Perspektif Tipe Pengguna Browser dan Evaluator. [Skripsi]. 2012; Bogor:IPB

[2] Sadiah, H.T, Ishlah. M.S.N. Implementation the Knuth Morris Pratt (KMP) Algorithm in Interactive Web Monitoring and Recording Rabbit Reproduction System. Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM).2019; vol.2(2) : 83-92.

[3] Sadiah, H.T. Implementasi Algoritma Knuth-Morris-Pratt Pada Fungsi Pencarian Judul Tugas Akhir Repository. Komputasi. 2017; vol.14: 115-124.

[4] Song Y, & Li-wei He. Optimal Rare Query Suggestion With Implicit User. *ACM Journals*; 2010 : 901-910

[5] Jiang S, Zilles S, Holte R. Query suggestion by query search: a new approach to user support in web search. 2008 [Online]. [Cited 2021 August 1]. Available from www.cs.uregina.ca/~zilles/jiangZH09.pdf

[6] Yangy J.-M, Cai R, Jingz F, Wangy S, Zhangy L, Ma W.Y. Search-based Query Suggestion. 2008 [Online] [Cited 2021 August 1]. Available from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.159.3499&rep=rep1&type=pdf

[7] Mei Q, Zhou D, Church K. Query Suggestion Using Hitting Time. 2008 [Online]. [Cited 2021 August 1]. Available from https://www.microsoft.com/enus/research/wpcontent/uploads/2017/01/sugg.pdf

[8] Cao H, Jiang D, Pei J, He Q, Liao Z, Chen E, Li H. Context-Aware Query Suggestion by Mining Click-Through . 2008 [Online]. [Cited 2021 August 1]. Available from https://www.cs.sfu.ca/~jpei/publications/QuerySuggestion-KDD08.pdf

[9] Zha Z.-J, Yang L, Me T, Wang M, Zengfu. Visual Query Suggestion. *ACM Journals*; 2009: 15-24.

[10] Sadiah H.T, Ishlah M.S.N, Rokhmah N.N. Query Suggestion on Drugs e-Dictionary Using the Levenshtein Distance Algorithm. *Lontar Komputer*; 2019. 10(3): 193-202.

[11] Sadiah, H.T, Gasbara M.A, Lily, N.S.A. Usability Testing on Android-based KMS for Pregnant Women using the USE Questionnaire. 2020; vol.1: 164-173.

[12] Pratama, B., & Pamungkas, S.. Analisis Kinerja Algoritma Levenshtein Distance Dalam Mendeteksi Kemiripan Dokumen Teks. *Jurnal Log!k@* . 2016; vol. 6(2) :. 131-143.

[13] Aprilianto, T., & Badawi , A. Sistem Koreksi Kata Dan Pengenalan Struktur Kalimat Berbahasa Indonesia Dengan Pendekatan Kamus Berbasis Levenshtein Distance. *Jurnal SPIRIT*. 2017.; vol. 9 (1): 48-61.

[14] Rosmala D, Risyad ZF. Algoritma Levenshtein Distance Dalam Aplikasi Pencarian Kata Isu Di Kota Bandung Pada Twitter. *MIND Journal*. 2017; vol.2(2):1-12.

[15] Adriyani, N. M., Santiyasa, I. W., & Muliantara, A. (n.d.). Implementasi Algoritma Levenshtein Distance Dan Metode Empiris Untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Dokumen Berbahasa Indonesia. [Cited 2018 August 1]. Available from https://ojs.unud.ac.id/index.php/JLK/article/view/2800

[16] Sadiah H.T, Ishlah M.S.N, Rochmah N.N. Autocorrect pada Modul Pencarian Drugs e-Dictionary Menggunakan Algoritma Levenshtein Distance. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi).* 2020; vol. 5 : 64-69.

[17] Mishra, R., & Kaur, N. A Survey of Spelling Error Detection and Correction Techniques. International Journal of Computer Trends and Technology. 2013; vol.3 (4): 372-374.

[18] Ariyani, N., Sutardi, & Ramadhan, R. 2016. Aplikasi Pendeteksi Kemiripan Isi Teks Dokumen Menggunakan Metode Levenshtein Distance. *semanTIK*. vol.2(1):279-286.

[19] Haldar, R., & Mukhopadhyay, D.2011. Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach. [Cited 2018 August 1]. Available from http://www.cornell.edu/: https://arxiv.org/abs/1101.1232.

[20] Suhendra M, Sadiah, H.T. Aplikasi Helpdesk Teknologi Informasi Berbasis Website. Jurnal Aplikasi Bisnis dan Komputer (Jubikom). 2021; vol.1(2) :44-51.

[21] Zuraiyah T.A. Sadiah, H.T. Hermawan E. Pengembangan Learning Management System (LMS) Pelatihan SDM menggunakan PHP dan MYSQLI. Jurnal Aplikasi Bisnis dan Komputer (Jubikom). 2021; vol.1(2) :77-78.

[22] Hidayati N, Retnowati D. 2019. Kamus Lengkap Biologi Sesuai Kurikulum Terbaru. Surabaya: Dwimedia Press.

[23] Hidayat F.N, Qurania A. Sadiah H.T. Aplikasi Pengelolaan Data Dokumen Mahasiswa Diploma Tiga Sistem Informasi Universitas Pakuan Jurnal Aplikasi Bisnis dan Komputer (Jubikom). 2021; vol.1(1) :13-21.

[24] Sadiah H.T , Ishlah M.S.N , Elfrieda N.S.A.L, Gasbara M.A . KMS (Knowledge Management System) Obat Ibu Hamil Berbasis Android. Jurnal Teknologi Informasi dan Ilmu Komputer.2017; 8 (2), 253-264.

[25] Budi M.A.S, Sadiah, H.T. Digitalisasi Pengarsipan Surat Pada Kantor Kecamatan Cigudeg. Jurnal Aplikasi Bisnis dan Komputer (Jubikom). 2021; vol.1(1) :38-43.

BIBLIOGRAPHY OF AUTHORS

Khalida, S.Kom, was born in Bogor on July 18, 1996. Currently, the author is a student of the Master's Program in Computer Science, Budi Luhur University. Graduated with a bachelor's degree from Pakuan University majoring in Computer Science. Her area of expertise is Algorithm, Web Technology, and Information Retrieval.