IJDASEA
International Journal of Data Science Engineering, and Analytics

e-ISSN : 2807-1689
p-ISSN : 2798-9208

Word Search on the "SITUK" Application Using the Levenshtein Distance Algorithm

Kraugusteeliana [1], Gideon Setya Budiwitjaksono[2], Alfiatun Masrifah[3], Muhammad Rif'an Dzulqornain[4], I Gede Susrama Mas Diyasa[5*]

Citation: Kraugusteeliana, Budiwijaksono, G.S., Masrifah,m A., Dzulqornai, M.R., and Diyasa., I.G.S.M. *C* **2021**, Volume *1*, Number 2, Page 55-62.

Received: November 12, 2021
Accepted: November 22, 2021
Published: November 25, 2021

[1] Department of Information System, University of Pembangunan Nasional "Veteran" Jakarta; kraugusteeliana@upnvj.ac.id
[2] Department of Accounting, University of Pembangunan Nasional "Veteran" Jawa Timur; gideon.ak@upnjatim.ac.id
[3] Department of Informatics, University of Pembangunan Nasional "Veteran" Jawa Timur; muhammadrifan153@gmail.com
[4] Department of Informatics, University of Pembangunan Nasional "Veteran" Jawa Timur; alfiatunmasrifah9@gmail.com
[5] Department of Data Science, University of Pembangunan Nasional "Veteran" Jawa Timur; igsusrama.if@upnjatim.ac.id
* Correspondence: igsusrama.if@upnjatim.ac.id;

Abstract: Integrated System for Online Competency Certification Test (SITUK) is an application used to carry out the assessment process (competency certification) at LSP (Lembaga Sertifikasi Profesional) UPN (University of Pembangunan Nasional) "Veteran" Jawa Timur, each of which is followed by approximately five hundred (500) assessments. Thus the data stored is quite a lot, so to find data using a search system. Often, errors occur in entering keywords that are not standard spelling or typos. For example, the keyword "simple," even though the default spelling is "simple." Of course, the admin will get incomplete information, and even the admin fails to get information that matches the entered keywords. To overcome the problems experienced in conducting data searches on the SITUK application, we need a string search approach method to maximize the search results. One of the algorithms used is Levenshtein which can calculate the distance of difference between two strings. Implementation of the Levenshtein algorithm on the data search system in the SITUK application has been able to overcome the problem of misspelling keywords with the mechanism of adding, inserting, and deleting characters.

Keywords: Levenshstein; word search; SITUK application

1. Introduction

Information technology in the current millennial era makes human work more accessible in many fields. This development impacts the field of computer science, but other areas also feel the benefits of the development of information technology. In daily activities, technological advances help other fields speed up and make it easier to solve a problem. One of them is the use of information technology at the Professional Certification Institute (LSP), Diyasa (2021). In LSP, information technology can help facilitate the implementation of competency assessment and professional certification by adapting manual processes into information technology. LSP UPN "Veteran" Jawa Timur is a supporting institution for the National Professional Certification Agency

(BNSP: Badan Nasional Sertifikasi Profesional), which is responsible for carrying out competency certification and competency assessment to students by their professional expertise. This institution already has a license and meets the requirements to carry out professional certification activities by BNSP, Diyasa (2020).

LSP UPN "Veteran" Jawa Timur is an institution that conducts competency assessments and competency certification for students according to professional skills that have been licensed by the BNSP. Competency certification is the process of providing competency certificates that are carried out systematically and objectively through a good work competency test that refers to work competency standards.

Sampurno(2020), In a previous study, "The Integrated Competency Testing System (SITUK) of Professional Certification Agencies Using the Nuxt.JS Framework" conducted research that resulted in a website-based certification competency test application that facilitates the implementation and processing of data from previously manual to digital and computerized in a structured manner. . Where changes are made in the registration process, the performance of independent assessments, maintenance and data processing by the admin, to related functions in the implementation of the competency test. Another research is Sugiarto (2020), which explains the application of the filing and disposition of mail information systems by implementing the Levenshtein distance algorithm, making it easier for agencies to process correspondence.

In contrast to previous research, the author will develop this research from an earlier study with a website-based system that has made registering and carrying out independent assessments easier. Data processing by admins is more accessible than using a manual system. This research will focus on the side of prospective participants and participants or called "Asesi". This research will be developed to be mobile-based, where registration can be done through the website and a mobile application. The advantage of this mobile-based system is that it makes it easier for registrants to register for exams. Besides that, in the application, there are various features. The exam schedule notification feature can make it easier for participants to remember the exam schedule to be held. There is a feature to download exam guidelines or print out the results of "Asesi" observations. In addition to the exam schedule notification feature that distinguishes it from previous research. There is also a search feature. This feature can easily and quickly search for a list of exams in the application. The search process can be assisted by using an algorithmLevenshtein Distance.

Using the algorithm Levenshtein Distance (Yulianto, 2018) This can make it easier for writers to create a search feature that can overcome writing errors in conducting searches, in contrast to not using the Levenshtein Distance algorithm method. If the user makes a typo in the search feature, the data they are looking for cannot appear. The advantage of using the Levenshtein Distance algorithm is that it simplifies the picarian process. If there is an error in typing in the search, the data will still display what the user wants to find.

2. Related Works

2.1. Levensthein Distance Algorithm

Levenshtein Distance is a string matrix used to measure the difference between two strings. The value of the distance between two strings is determined by the minimum number of change operations required to transform from one string to another. These operations are insertion, deletion,

and substitution. (Halimah, 2019) Mathematically, the value of the Levenshtein distance between two strings a, b. Equation (1) is the formula and calculation of the Levenshtein Distance algorithm (Mawardi, 2020).

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

2.2. Operations on the Levenshtein Distance Algorithm

There are three kinds of operations in the Levenshtein distance algorithm (Santoso, 2019) that can be performed including:

1. Character Insert Operation (insection)

   The operation of inserting characters into a string uses the Levenshtein distance algorithm. A simple example is the string 'laptop to string 'laptop' can insert the character 'p' at the end of the word. Insertion is done at the end of the string and can also be inserted at the beginning and between the strings—illustrations such as Table 1.

Table 1. Insertion of characters

Strings 1	l	a	p	o	P
String 2	l	a	p	o	-
insection					P

2. Character Deletion Operation (Deletion)

   The Levenshtein distance algorithm (Clarissa, 2020) can perform the operation of deleting characters in a string. For example, the string 'keyboard' can be changed to 'keyboard' by removing one character, namely 'r'—illustrations in Table 2.

Table 2. Character deletion

Strings 1	k	e	y	-	b	o	a	r	d
String 2	k	e	y	R	b	o	a	r	d
Deletion				R					

3. Character Swap Operation (Substitution)

   The Levenshtein distance algorithm can do the operation of exchanging characters in a string. For example, the string 'botor' with character swapping the string 'motor' with changing the character 'b' with the character 'm' with illustrations such as Table 3.

Table 3. Character swap

Strings 1	m	o	t	o	r
String 2	b	o	t	o	r
substitution	m				

2.3. Competency Test with SITUK Application

The LSP UPN "Veteran" Jawa Timur is an institution that conducts competency assessments and competency certification for students according to professional skills that have been licensed by the BNSP. Competency Certification is the process of providing competency certificates that are carried out systematically and objectively through competency exams that refer to the certification scheme that has been made by LSP and approved by BNSP. The competency certification process organized by LSP consists of registering prospective participants up to the issuance of competency certificates.

Professional certification activities at the LSP begin with registration through the LSP by filling out the APL 1 file. After the registration process is accepted, the registrant will be called an "asesi" and allowed to take the competency test. The accepted assessor will receive information on the place of the competency test (TUK: Tempat Uji Kompetensi), the date of implementation, and the assessor in charge of conducting the competency test. Of course, there are parties in order of testing in the competency test process, namely assessors. The assessor's task is to conduct an assessment and competency test for the assessor. Before carrying out the competency test, the assessor must conduct an independent assessment, namely filling out the APL II form, which will test according to the assessor's ability. Furthermore, the assessor performs a competency test on whether the assessment is worthy of being assessed as competent.

Based on the explanation above, it can be seen that the process that occurs in LSP is quite long and requires sound data processing. So far, LSP has carried out the procedure manually, meaning that all activities from registration to competency test results are still using the manual method, which is paper-based and carried out face-to-face. As a result, these activities run slowly, and the recorded data is not automatically connected to other sections, and data related to the assessment is not stored neatly.

Related to the above problems, the solution to solve these problems is to create an Integrated System for Online Competency Certification Tests based on Web, Android and IoS called the SITUK Application, which can carry out the participant registration process, independent assessment by "Asesi", data maintenance by admin or LSP employees, and the process of determining graduation becomes systemized and integrated.

3. Experiment and Analysis

3.1. SITUK Design with Levensthein Distance Algorithm

In the implementation phase, the search and collection of information needed during system design are carried out. The method used to collect information and data is a literature study method, namely by studying literature related to research, including the Levenshtein Distance Algorithm and its application to the Indonesian spelling checking system, programming languages PHP, HTML, and Javascript.

Checking is done word by word. The input sentence will enter the preprocessing stage first before being processed further. The preprocessing process includes the removal of punctuation marks, tokenization of each word. Then, each token will be matched to the database using the Levenshtein Distance Algorithm and Empirical Method (Abdulkhudhur, 2016). After the calculation

is done, the system will display a word suggestion close to a writing error—flowchart of the spelling check system as shown in Figure 1.

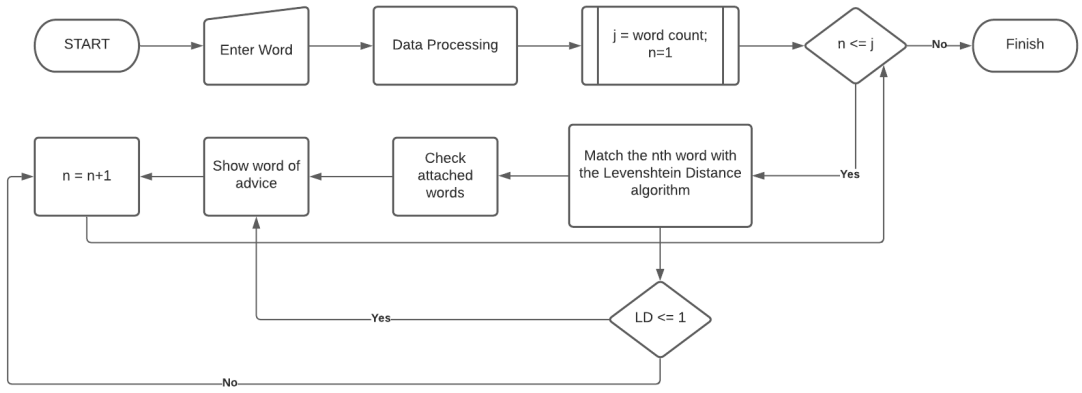

Figure 1. Spell Checking System Flowchart

3.2. Data preprocessing

Data preprocessing (Veena, 2015) includes checking whether the input characters have input in letters, numbers, or symbols. The following is a code snippet of the tokenization process, as shown in Figure 2.

```
$length=strlen($a);
 $c=0;
 $d=0;
 $word="";
 for ($i=0;$i<$ length;$i++)
 {
   if(strcmp($a[$i]," ")==0)
   {
$kata="";
    for($j=$d; $j<$i; $j++)
   {
    $kata=$word."".$a[$j];
   }
    $b[$c] = $word;
    $c=$c+1;
    $b[$c] = $a[$i];
    $c=$c+1;
$d = $i+1; }
else {
$kata="";
   {
   $kata=$word."".$a[$j];
   $b[$c] = $word;
   }
}
}
return $b;
```

```
$m[$i][$j]=0;
for($i=1;$i<=$x;$i++){
    $m[$i][0]= $i;
}
for($j=1;$j<=$y;$j++){
    $m[0][$j]=$j;
}
for($i=1;$i<=$x;$i++)
{
for($j=1;$j<=$y;$j++)
{
    if($sb1[$i]==$sb2[$j]){
    $cost=0;
} else{
$cost=1; }
    $m1=$m[$i-1][$j-1]+ $cost;
    $m2=$m[$i-1][$j]+1;
    $m3=$m[$i][$j-1]+ 1;
    $m[$i][$j]=min($m3,$m2,$m1);
     if ($i==$x and $j==$y)
      {
       $minimum = $m[$i][$j];
       if ($m[$i][$j]<=1){
        $wordof[$z]=$kata2;
        $z=$z+1;
        }
}
}
C.Empiris methods
}
return $wordof;
```

```
function break ()
{
Inisialisasi $teks,$lenght_teks;
for($i=0;$i<1;$i++)
{
    $b=($amount+1)-$i;
for($j=1;$j<$b;$j++)
    {
$text=substr($data['test'],$i,$j);
$text2=substr($data['test'], $j, $b);
if(right($text)==1)
    {
$datatext = array($text,$text2);
    }
}
echo "<br>";
}
    return $datatext;
}
function right($teks)
{
$result=0;
$query= mysql_query("SELECT *
FROM list_words where
list_words='$teks'");
$data=mysql_fetch_assoc($query);
if($data['list_words'] != NULL)
}
{
$result=1;
}
return $result;
```

a. Tokenization Process b. Levenshtein Distance Algorithm c. Empirical Method

Figure 2. Data Processing with Levenshtein Distance Algorithm

3.3. Levenshtein Distance Algorithm Function Analysis

A 2 (two) dimensional matrix is used in calculating the Levenshtein Distance value. The value in the matrix is the number of deletion, insertion, and exchange operations required to convert the source string to the target string (Alexandru, 2017). The operation formula for deleting, inserting, and exchanging characters is used to fill in the matrix values, as in equation (2)-(4).

$$D(s, t) = \min D(s - 1, t) + 1 \text{ (Deletion)} \quad (2)$$

$$D(s, t) = min\ D(s, t - 1) + 1 \text{ (Insertion)} \quad (3)$$

$$D(s, t) = min\ D(s - 1, t - 1) + 1,\ sj \neq ti \text{ (Exchange)} \quad (4)$$

$$D(s, t) = \min D(s - 1, t - 1)\ ,\ sj = ti \text{ (No changes)} \quad (5)$$

s= Source String; $s(j)$ = Source String Character to-j; t = Target String; $t(i)$ = Source String Character to-i; D = Levenshtein Distance

The Levenshtein Distance Algorithm function is tested in the case of searching for application data SITUK, which contains "Online Competency Certification" as the source string and the keywords "Competent" and "Competency" contained in the issue category "Competency", based on this example, each word in the SITUK application, namely "Competent" and "Oline" is calculated the distance from all keywords. The first step of the Levenshtein Distance Algorithm is to calculate the distance of the first "simple" source string with the first target string, "simulate". The matrix calculation begins with initiating the sequence of characters in each string as illustrated in Figure 3.

In Figure 3, it is known that the source string "simulate" has 8 (eight) characters, and the target string "simple" has 6 (six) characters. Furthermore, the 1st character in each string is compared, and it is known that the contents of the 1st character in each string are the same, then the matrix value given is by equation (4), namely $D(1,1) = D(1-1,1-1)$, $sj = ti$. So the value of the matrix given to ($D(1,1) = D(0,0)$ worth 0. Then the matrix value at $D(1,1)$ is filled as depicted in Figure 3(a).

String Target

Source String		**s**	**i**	**m**	**p**	**e**	**l**
	0	1	2	3	4	5	6
s	1	0					
i	2						
m	3						
u	4						
l	5						
a	6						
t	7						
e	8						

(a) Distance D(1,1)

Source String		**s**	**i**	**m**	**p**	**e**	**l**
	0	1	2	3	4	5	6
s	1	0	1				
i	2						
m	3						
u	4						
l	5						
a	6						
t	7						
e	8						

(b) Distance D(1,2)

Source String		**s**	**i**	**m**	**p**	**e**	**l**
	0	1	2	3	4	5	6
s	1	0	1	2	3	4	5
i	2	1	0	1	2	3	4
m	3	2	1	0	1	2	3
u	4	3	2	1	1	2	3
l	5	4	3	2	2	2	3
a	6	5	4	3	3	3	3
t	7	6	5	4	4	4	4
e	8	7	6	5	5	5	**5**

(c) Distance D(8,6)

Figure 3. Character Sequence Initiation

Furthermore, the distance calculation is carried out on the 1st character of the source string with the 2nd character in the target string. It is known that the insertion operation of the "i" character

in the source string is required, then the value given is by the insertion operation formula in Formula (2), $D(1,2) = D(1,2-1) + 1$. So the value assigned to $D(1,2) = D(1,1) + 1$ worth $D(1,2) = 0 + 1 = 1$. Then the matrix value at $D(1,2)$ is filled with 1, as illustrated in Figure 2 (b). Distance calculation is carried out on the 1st character of the source string with the target string up to the 3rd character. It is known that the insertion operation of "i" and "m" characters is needed in the source string, then the value given is by the insertion operation formula in Formula 2, $D(1,3) = D(1,3-1) + 1$. So the value assigned to $D(1,3) = D(1,2) + 1$ worth $D(1,3) = 1 + 1 = 2$. Then the matrix value at $D(1,3)$ is filled with value 2.

The distance calculation is carried out on the 1st character of the source string with the target string up to the 4th character. It is known that the insertion operation of the characters "i", "m", and "p" in the source string is required, then the value given is by the formula the insertion operation in equation (2), $D(1,4)=D(1,4-1)+1$. So the value assigned to $D(1,4) = D(1,3) + 1$ worth $D(1,3) = 2 + 1 = 3$. Then the matrix value at (1,4) is Filled with value 3. The distance calculation is carried out on the 1st character of the source string with the target string up to the 5th character. It is known that the insertion operation of the characters "i", "m", "p", and "e" in the source string is required, then the value given is by the insertion operation formula in equation (2), $D(1,5)=D(1,5-1)+1$. So the value assigned to $D(1,5) = D(1,4) + 1$ worth $D(1,5) = 3 + 1 = 4$. Then the matrix value at $D(1.5)$ is filled with value 4.

And so on until the following Levenshtein Distance calculation runs until all the values in the matrix are filled. The Levenshtein Distance is the value in the bottom-right of the matrix, and in the case of the source string “Simulate” and the target string “simple” it is at (8,6). After doing all the matrix calculations, it is known that the result of calculating the distance between the source string "Simulate" and the target string "simple" is 5 (five), as illustrated in the matrix Figure 3 (c).

4. Conclusions

From the discussion that has been done to implement the Levenshtein Distance algorithm in The Online Competency Certification Test Integrated System (SITUK) can be concluded that applying the Levenshtein Distance Algorithm can help overcome the problem of typing errors with the mechanism of adding, inserting, and deleting characters. The optimization of corrective words given by the system can be improved by implementing the Empirical Method to find out if there are words written without spaces so that the suggestions given can reach the users' expectations. This check is still limited to checking for typing errors, not matching Indonesian sentence patterns.

References

1. Abdulkhudhur, H. N. (2016), ‘Implementation Of Improved Levenshtein Algorithm For Spelling Correction Word Candidate List Generation’, *Journal Of Theoretical And Applied Information Technology*, Vol. 88 (3). Pp. 449-455.
2. Alexandru, E. (2017), ‘An application of Levenshtein algorithm in vocabulary learning’, International Conference –Electronics, Computers and Artificial Intelligence (ECAI)-IEEE Xplore, pp. 1-4.
3. Clarissa, W. (2020), ‘MeDict: Health Dictionary Application Using Damerau-Levenshtein Distance Algorithm’, IJNMT, Vol. VII (2), DOI: https://doi.org/10.31937/ijnmt.v7i2.1654
4. Diyasa, I. G. S. M. (2021) ‘Comparative Analysis of Rest and GraphQL Technology on Nodejs-Based API Development’, NST Proceeding, 5thInternational Seminar of Research Month 2020,

Volume 2021. http://dx.doi.org/10.11594/nstp.2021.0908

5. Diyasa, I. G. S. M. (2020) 'Graph-QL Responsibility Analysis at Integrated Competency Certification Test System Base on Web Service ', Lontar Komputer VOL. 11 (2), DOI : 10.24843/LKJITI.2020.v11.i02.p05
6. Halimah, T. S., 'Query Suggestion on Drugs e-Dictionary Using the Levenshtein Distance Algorithm', Lontar Komputer, Vol. 10 (3), 2088-1541 DOI: 10.24843/LKJITI.2019.v10.i03.p07
7. Mawardi, V. C., 'Spelling Correction Application with Damerau- Levenshtein Distance to Help Teachers Examine Typographical Error in Exam Test Scripts', *ICESTI,* E3S Web of Conferences 188, 00027, https://doi.org/10.1051/e3sconf/202018800027.
8. Veena, G. (2015), 'Levenshtein Distance based Information Retrieval', International Journal of Scientific & Engineering Research, Vol. 6 (5), http://www.ijser.org
9. Santoso, P. (2019), 'Damerau levenshtein distance for indonesian spelling correction', Jurnal Informatika, Vol. 13 (2), DOI: http://dx.doi.org/10.26555/jifo.v13i2.a15698
10. Sampurno (2020), "The Integrated Competency Testing System (SITUK) of Professional Certification Agencies Using the Nuxt.JS Framework, Skripsi, Informatic, UPN Veteran Jawa Timur
11. Sugiarto (2020),'Levenshtein Distance Algorithm Analysis on Enrollment and Disposition of Letters Application', IEEE *Xplore 2021,* Information Technology International Seminar (ITIS), **DOI:** 10.1109/ITIS50118.2020.9321030
12. Yulianto (2018), 'Autocomplete and Spell Checking Levenshtein Distance Algorithm to Getting Text Suggest Error Data Searching in Library', Scientific Journal of Informatics, Vol 1(5), http://journal.unnes.ac.id/nju/index.php/sji