International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V71I1P209

Volume 71 Issue 1, 94-108, January 2023

Original Article

Improve Data Text Quality by Applying Text Pre-Processing Method (Case Study)

Rizky Dwi Novyantika[1], Sani Muhamad Isa[2]

[1,2] *Computer Science Department, Bina Nusantara University, Jakarta, Indonesia.*

[1]*Corresponding Author : rizky.novyantika@binus.ac.id*

Received: 22 August 2022 Revised: 13 December 2022 Accepted: 07 January 2023 Published: 24 January 2023

Abstract *- To develop a business, especially in a startup, they must pay attention and consider important aspects. Several articles and journals said that one of the aspects that must be considered is location. PT. MDS is one of the startups that offer a Point of Sales (POS) system to MSMEs; where to get this feature, MSMEs need to register by filling in personal data, including their business locations such as Province and City. In this step, the data entered is still typed manually, and make data entered into the database is unstructured. Therefore, this study aims to improve data quality by using the pre-processing method, Data Correction with Cosine Similarity and Jaro-Winkler Distance algorithms and Data Integration to complete the missing data. Implementing the pre-processing method itself can improve about 81.50% of Province data and 92.31% of City data. The Cosine Similarity algorithm is quite good at capturing and matching data at the word level, while Jaro-Winkler Distance is quite good at the string level. The Jaro-Winkler Distance algorithm is easier to implement than Cosine Similarity because Cosine Similarity requires converting the data into a matrix before implementing the algorithm. This study shows that combining the three methods mentioned can improve the quality of Province and City data by up to 99.36% and 97.99%. The data integration process itself successfully completes the missing data up to 97.38%.*

Keywords *- Data quality, Data pre-processing, Cosine similarity, Jaro-Winkler distance, MSMEs.*

1. Introduction

In developing a business, especially a *startup*, they must pay attention to and consider important aspects to develop the business based on certain factors. A business article stated that: to start developing a business, the *startup* should involve a marketing mix concept known as 4P (Product, Price, Place and Promotion) [1]. In other business journals, it is also stated that choosing a business place or location is one of the business decisions that must be made carefully [2] because location plays an essential role in the successful operation of its business. Location is important for sustainability and growth, especially for information technology organizations [3].

The relationship between choosing a location at *startup* (in this case: PT. MDS) is to make analysis related to MSMEs location easier to conduct and get the right insight and information, such as finding out the location distribution of apps usage and PT. MDS can increase the number of system users and transactions in certain locations, which is still low and requires accurate information regarding existing locations. PT. MDS is a *startup* with a vision and mission to digitize MSMEs (Micro, Small and Medium Enterprises) through the Point of Sales (POS) system. PT. MDS offers a variety of features that really help run business processes.

To get the features of the POS system, MSMEs need to register by filling in personal data, including business location such as Province and City. This process is still manually typed without any scroll-down option. In this way, the data that enters the database becomes unstructured, where it can be in the form of typos, data filled in the incorrect place and even data that the MSMEs do not fill in as it makes data incomplete or missing.

Therefore, in this study, data quality will be improved by using several methods, including Data Pre-processing for cleaning and removing unwanted characters, Data Correction to fix data typos by matching data from a trusted source with Cosine Similarity and Jaro-Winkler Distance algorithms and Data Integration to fill in the missing data by using the schema matching method and retrieving data based on available data. Several methods mentioned can improve quality and quantity and make data more informative and comprehensive.

2. Literature Review

In this study, the literature review will be divided into four parts: a review of research related to Data Pre-processing, Data Correction (Jaro-Winkler Distance & Cosine Similarity Algorithm) and Data Integration.

2.1. Research Related to Pre-processing Data

In their research, Hakim [4] and Jaka [5] conducted a study to determine the effect of Pre-processing data text on the accuracy of the sentiment analysis data mining model. Hakim used a dataset of 50,000 reviews on the Internet Movie Database (IMDB) with three different treatments, which are (1) *Baselines* where the dataset is left original, (2) *Stop Words* where repeated words are considered conjunctions, and (3) *Stemming* where the text dataset will be normalized and cut to get the root just the sentence. The results of this study indicate that Pre-processing data affects the accuracy of the data mining model. While Jaka used document datasets using several Pre-processing methods, including *Transform Cases*, *Tokenization* and *Stop Words*. With this pre-processing method, a lot of unused data will be eliminated before the dataset is subjected to the existing sentiment analysis method.

Khadim [27], Srividhya [7], and Nurfadila [8] conducted research to determine the effect of pre-processing data text on the accuracy of the classification analysis. Khadim classified English text with three methods of pre-processing, including (1) *Tokenization*, (2) *Stop Word* and (3) *Stemming*. Srividhya uses three methods of pre-processing data, which are (1) *Stop word*, (2) *Stemming* and (3) *TF/IDF* on the Reuters dataset, and Nurfadila classified economic journal using *Stop Word*. All research shows that pre-processing affects the result of text classification analysis, and specifically from Nurfadila shows that the performance of the Cosine Similarity method added to *Stop Word* removal increased by 2% compared to classification without implementing the pre-processing method.

2.2. Research Related to Cosine Similarity Algorithm

Sugiyamto [9] and Nurdin [10] conducted research related to plagiarism documents. To conduct this research, Sugiyamto and Nurdin used Cosine Similarity Algorithm. Sugiyamto implements Cosine Similarity using dataset academic manuscripts such as Thesis or Final Projects. In contrast, Nurdin uses dataset Indonesian text documents with several file formats, such as doc, docx, pdf, and rtf. Both research show Cosine Similarity has a higher score compared to other methods. Based on Sugiyamto research, the accuracy rate of Cosine Similarity is higher at 94,98% compared to Jaccard's of 94,90% and based on Nurdin research accuracy rate using Cosine Similarity is up to 90%. This shows that the Cosine Similarity algorithm applied to this system is proven to identify plagiarism document similarities properly.

Thada, Jaglan [11] and Kurniadi [12] conducted research on document similarity by comparing documents to find the most relevant documents. Thada and Jaglan used 10 documents that appeared and Google searches and used three coefficients approach: Jaccard, Dice and Cosine Similarity. The result of this study on those 10 documents has an average accuracy value using Jaccard (24.95%), Dice (39.17%) and Cosine Similarity (49.59%). While Kurniadi used archiving documents from Sultan Agung Islamic University. The archived documents are not yet organized and cause several problems. Kurniadi, in his research, wants to fix it using the Cosine Similarity algorithm, and this study's results indicate that the results' precision using Cosine Similarity is 88.8%. Both researches show that Cosine Similarity can be the best algorithm for checking document similarity since the result is higher than other methods and has more than 80 precision%.

2.3. Research Related to Jaro-Winkler Distance Algorithm

Novantara and Pasruli [28], in their research, created a plagiarism detection system to minimize illegal duplication by calculating the similarity of the text in the original document and the document being tested using the Jaro-Winkler Distance. The results of this study indicate that accuracy with the Jaro-Winkler Distance algorithm achieves an accuracy of 96.01%.

Friendly [14], in his research, made improvements to the Jaro-Winkler Distance method in lexicographic comparisons to find words that match or approach the words that are looking for in multi-user-based applications so the repetition data can be minimized. The results of this study show a Mean Average Precision (MAP) value of 0.87, and the subsequent search process was between 90-92% faster than searching using the Jaro-Winkler method.

Yulianingsih [15] compared Levenshtein Distance and Jaro-Winkler Distance to find out which algorithm is faster for searching data that fits the needs, which was done five times with different parameters. The results of this study indicate that the Jaro-Winkler Distance algorithm obtains a faster time than the Levenshtein algorithm, with an average speed of 50% ahead average using the Jaro-Winkler algorithm is around 93 seconds. The Levenshtein algorithm is around 168 seconds.

2.4. Research Related to Information Integration

Saadah et al. [16] conducted research on integrating information on a document using the Longest Common subsequence (LCS) in adjusting TF-IDF weights considering the appearance of the same word order between the query and the text in the document. The existence of very long but irrelevant documents causes the resulting weights to be unable to represent the value of document relevance.

Huang et al. [17] researched information searches on text content using the library book index (book name, author, publisher information and date of publication) so librarians can quickly find book locations. Retrieval of this information using a search engine with a combination of text and web search. This study shows that there is the integration of information with a text search algorithm using keywords can be used to search documents based on the keywords input.

3. Materials and Methods

This section describes the process of this research, such as problem analysis, study literature, dataset collection and algorithm implementation, which includes Data Pre-processing, Data Correction, Data Integration, and Result and Evaluation. These stages can be described in the research flow as follows:

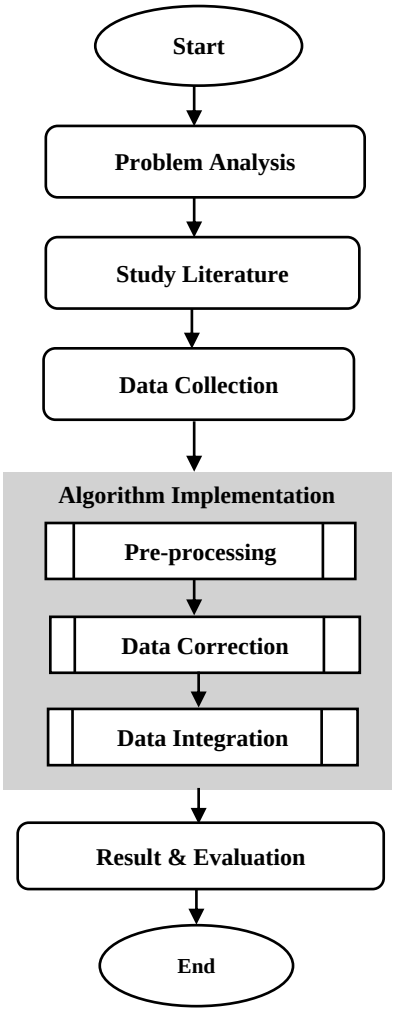

Fig. 1 Research flow

The following is an explanation of each of these stages of research flow shown in Figure 1 above:

3.1. Problem Analysis

From a total of 141,842 Province and 142,566 City data, there are three main problems with the highest frequency which are (1) "Punctuation & Typos", (2) "Aliases or Abbreviations" and (3) "Incorrect Spaces". However, besides those main problems mentioned, in City data, there is also a specific issue that has a high frequency which is in Indonesia, the City name following by the “Kabupaten” or “Kota” name, but most of the data inputs are not including those words.

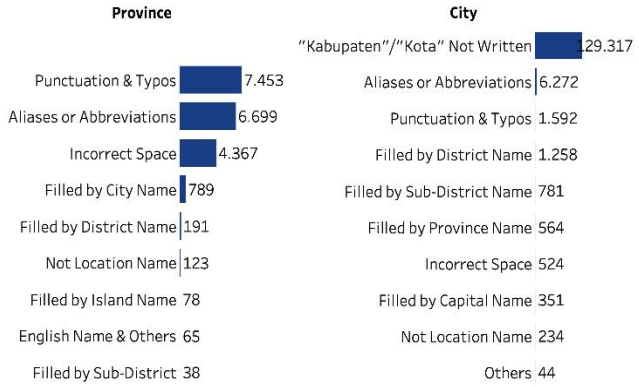

Fig. 2 Problem in Province and City Data

3.2. Study Literature

The literature study was conducted to obtain the best methods to solve the problem of MSMEs location text data at PT. MDS by collecting research journals related to data text quality improvement methods and based on a literature review that Data Pre-processing, Data Correction (Cosine Similarity and Jaro-Winkler Distance) and Integration Information or Data Integration are good for improving data quality by cleaning data from unwanted characters, correcting data that contains typos and research related to integration data from several locations (Province and City) to make better data quality.

3.3. Data Collection

At this stage, data is collected as material for solving the problems that have been formulated. As mentioned, data on MSMEs location was obtained from PT. MDS and reference data for data improvement, such as Province and City data, were collected from the Indonesia National Education Institute (Lediknas). In this study, MSMEs data locations were collected from 2020 to 2021, where the total data collected was 241,267. There were 141,842 Province data and 142,566 City data, which would be matched with data from Lediknas. Total data from Lediknas itself was obtained from 34 provinces and 514 cities [18].

3.4. Implementation Algorithm

3.4.1. Data Pre-processing

This section will implement pre-processing data to clean the data from unwanted characters [29]. This section is divided into several stages as follows:

Figure 3 shows different treatments for data locations obtained from Lediknas and MSMEs PT.MDS where Lediknas data is only given Case Folding treatment while MSMEs data is treated with Case Folding, Remove Punctuation and Replace sentence. The following describes each pre-processing stage:

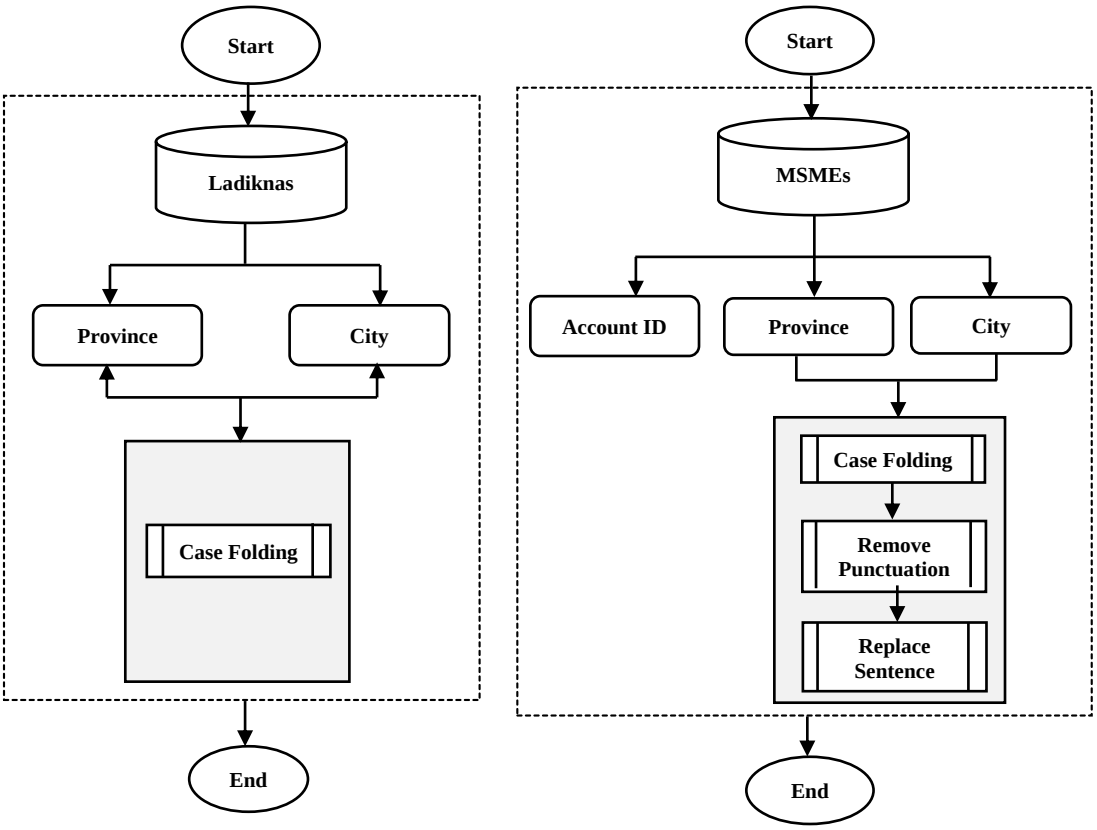

Fig. 3 Data Pre-processing Flow

Case Folding

At this step, all data for both the Province and City of MSMEs will be converted to lowercase letters. This step needed to separate the correct data (according to the location format from Lediknas); it also can make the next step easier by simplifying the data validation process [20].

Replace Sentence

This step is basically transforming data text by replacing a word with another word/character that is more in line with the reference data [21]. This step is also needed to make the Data Corrections process easier and make MSMEs data can be classified properly with reference data from Lediknas.

Remove Punctuation

This step returns keywords that are formed and processed into basic words. At this stage, all kinds of punctuation marks are removed. This step aims to get root words to prevent calculation errors when correcting data [20].

After pre-processing step is done, the second process is data matching. This matching process does not need a special algorithm because process matching directly reads text word by word. Data that does not match in this process will be continued to the following process, Data Correction.

3.4.2. Data Correction

After Data Pre-processing was completed, this research process moved to Data Correction using Cosine Similarity and Jaro-Winkler Distance algorithms. Data Correction is the activity of checking data which was declared (is possible) erroneous [22]. This step is needed to fix text with a problem such as writing with punctuation marks, typos and wrong spacing. These problems are the 1st and 3rd problems in Province data and the 3rd and 7th problems in City data. The data correction process can be seen in Figure 4 below:

Figure 4 shows that there are three steps in the data correction process:

1. Implementation of Cosine Similarity and Jaro-Winkler Distance

   This step is the implementation of Cosine Similarity and Jaro-Winkler Distance algorithms. In this step, both algorithms run respectively.

   However, both algorithms have the same goal: to get similarity scores from MSMEs and Lediknas data, where the score will be in the range from 0 to 1 [23]. Both algorithms have the same goal, but in this research, both algorithms will complete each other.

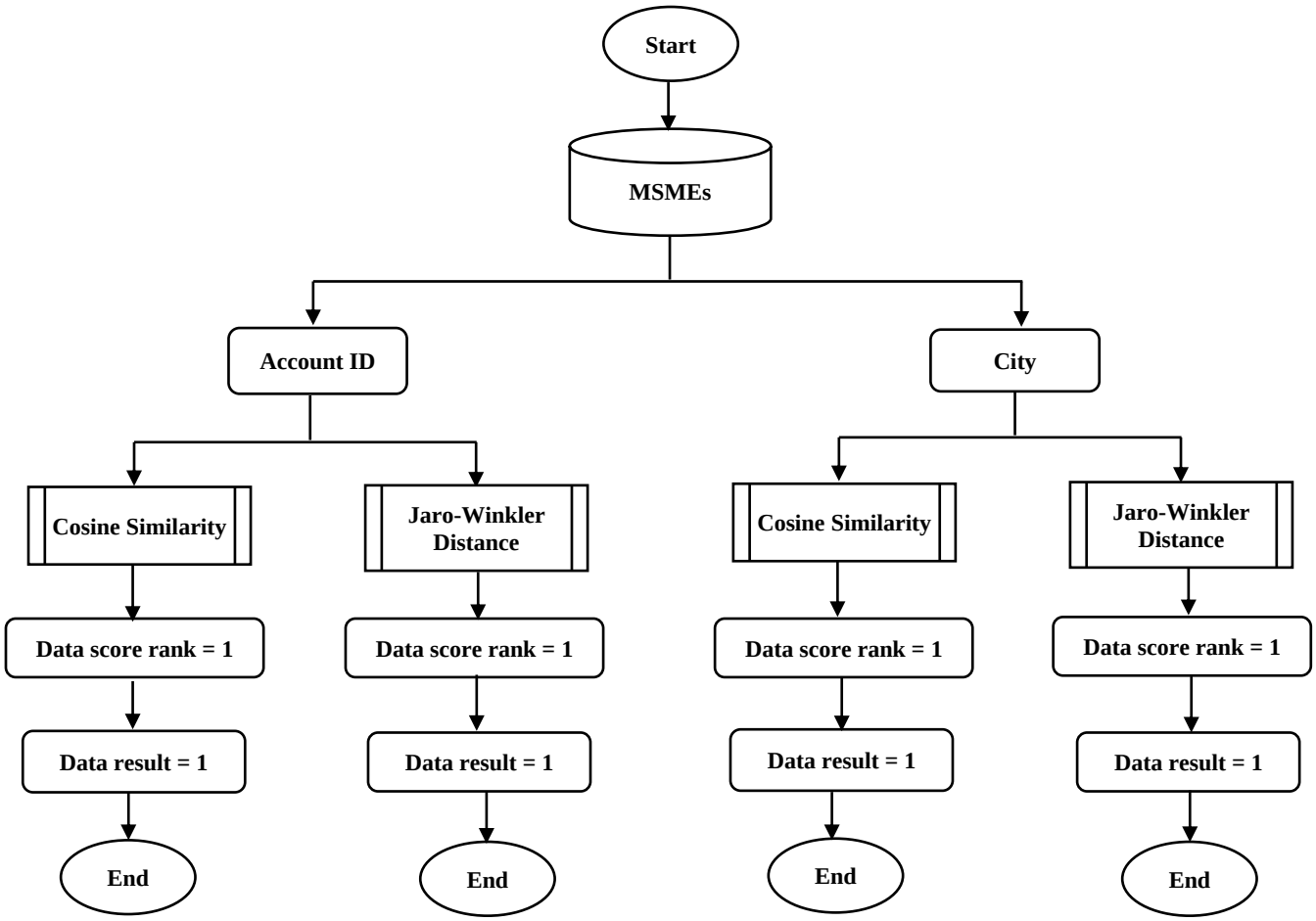

Fig. 4 Data Correction Flow

2. Data Score Rank = 1
   Both provinces implement this step, and City MSMEs, where this step will start to proceed after the implementation of Cosine Similarity and Jaro-Winkler Distance, show the score of each data. In this case, implementation in both algorithms might produce more than one score for only one data. Because of that, we need to get data with the highest similarity, which comes from data with the highest score and rank the data. Rank 1 is for the highest score of each data, assuming that the data with the highest similarity score is the data with the correct Lediknas reference.

3. Result data = 1
   After getting the most similar data based on data ranking, there is also a possibility that a data has a match on different Lediknas data but has the same similarity score; this means one data has two or more references which will be challenging to determine which is the most correct. So, this step is needed to handle this case. While data match with more than one Lediknas data, we assume we can not use it because it has high ambiguity. It also means that this research only used data with one similarity after the ranking process.

3.4.3. Information Integration

Based on background and problem analysis, it stated that MSMEs are located in PT. MDS have a data gap between total Province and City data, which shows City data is higher than Province data, which should be the same.

This problem still has the opportunity to be corrected, especially for MSMEs data that have City data. Still, data Province is missing because a City must have one province, so both data are related. We can integrate data or information based on City data in accordance with Lediknas data based on the result of Data Correction.

For example, one of the MSMEs has City "Bandung", and no Province data exists. So, based on Lediknas data, "Bandung" is included in the "West Java" Province, and then we can add "West Java" to the data that are missing.

This integration process can make the quantity of MSMEs data increase. Here is the detail of the step for integration data information in Figure 5:

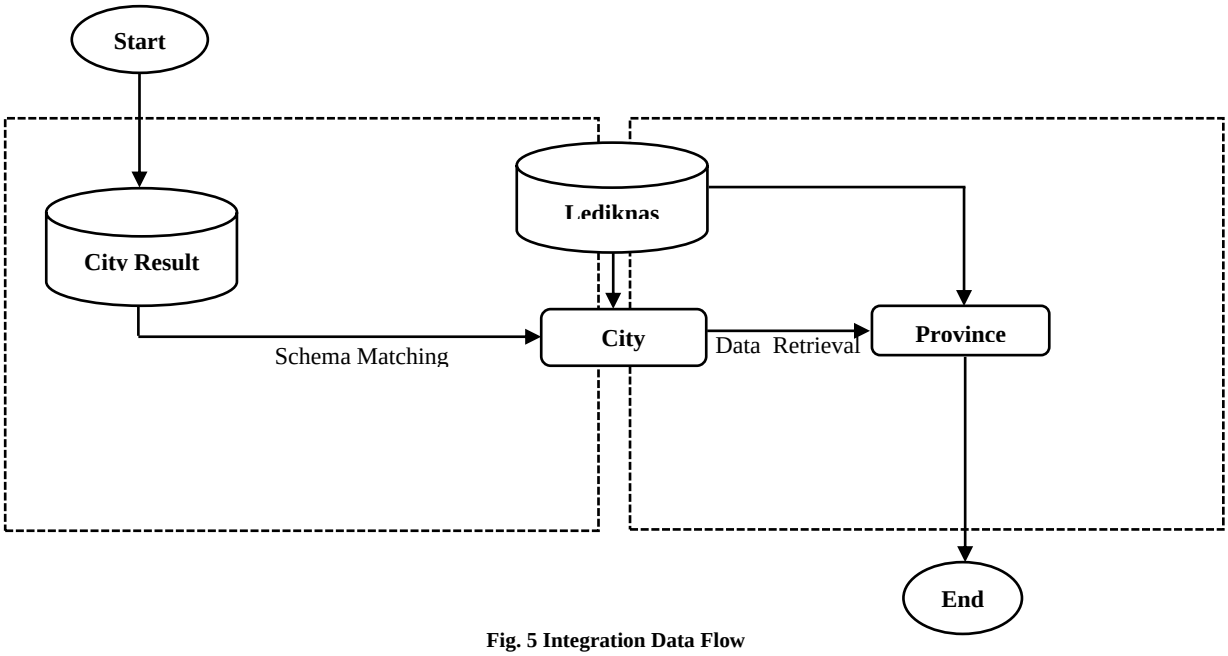

Fig. 5 Integration Data Flow

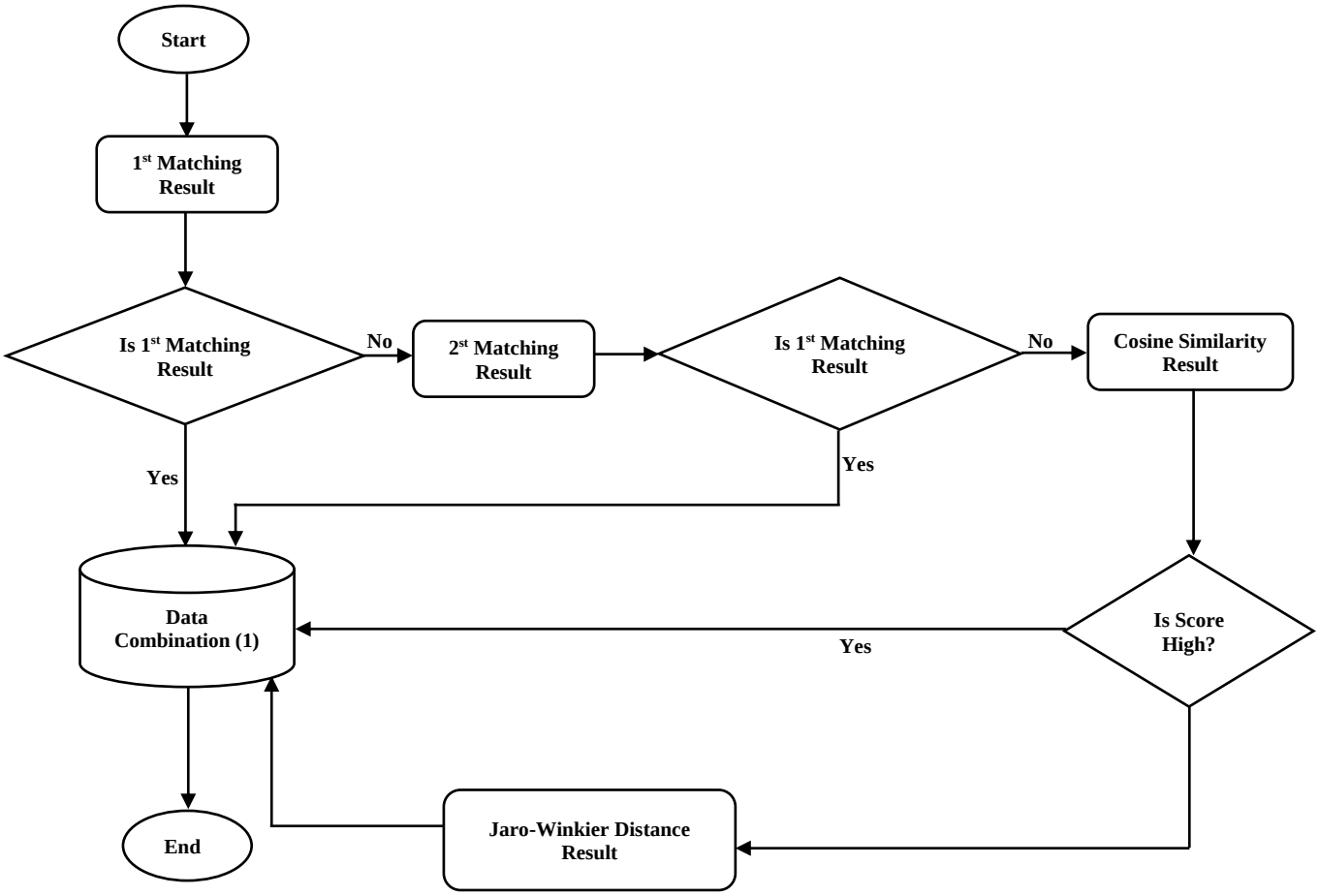

Fig. 6 1st Data Combination Flow

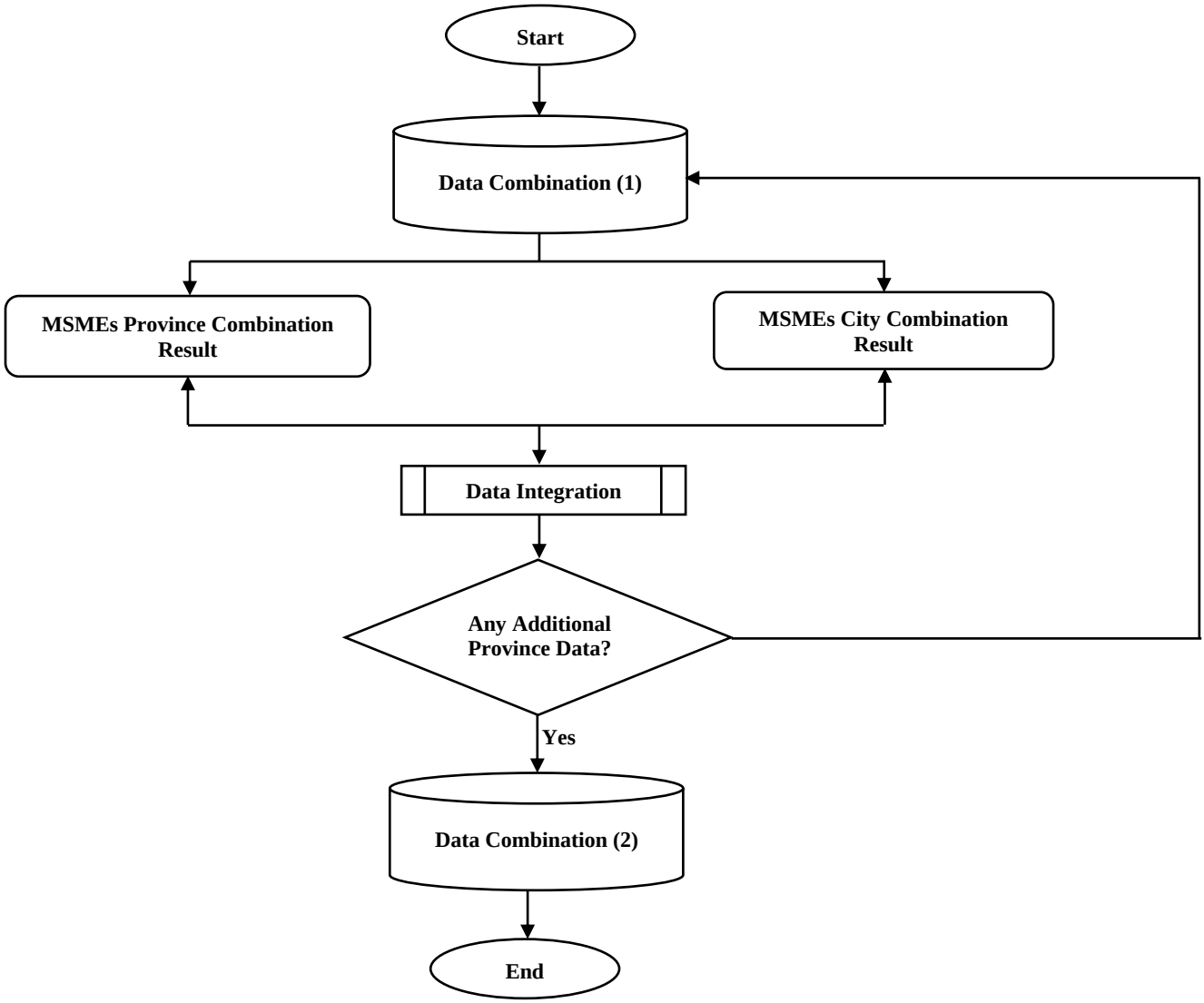

Fig. 7 2nd Data Combination Flow

Based on previous processes, data can be combined based on the best results from each process to produce one of the best data for each data as a result of data combination. Data has been combined in format according to the Lediknas data. In this process, schema matching is needed to combine City data that matched with Lediknas City and Data Retrieval is also needed to retrieve information on Province data from Lediknas.

3.4.4. Data Combination

Data Combination is basically a combination of data from previous processes. Based on Figure 1, which is the research flow, two Data Combinations are referred to as Data Combination (1) and Data Combination (2). Data Combination (1) combines data from the first match, second match and Data Correction result. The following is the flow of the Data Combination (1):

This combination process is implemented with the rule: if 1st data matching is available, then the data used is from these data; if the 2nd data matching is not available, then the Cosine Similarity result with a high score is used, but if the Cosine Similarity result is low then the Jaro-Winkler Distance result will be used.

These three results are combined to find out which reference data results have been matched by Lediknas, which will become a reference and be processed into the Information Integration step.

Data Combination (2) is the step to merge Data Combination (2) and the results of Data Integration. This process makes all data becomes more complete. The following is the flow of the Data Combination (2) process:

Figure 7 shows Data Combination (1) with the Information Integration result. In contrast, in Data Combination (1), there are still missing Province data, which goes through the Information Integration step. The data is complete and makes Data Combination (2) more comprehensive.

3.5. Result and Evaluation

This step basically evaluates the results of the implementation algorithm. The method used to evaluate and determine the performance or correctness is Confusion Matrix which can be divided into values using Accuracy, Precision and Recall. *Accuracy* is used to determine the degree of closeness between the corrected value and the actual value, *precision* is used to determine the comparison of the amount of relevant information and *Recall* is used to compare the amount of relevant information from the results of the algorithm. The following is an example of implementing the Confusion Matrix on MSME location data:

Table 1. Confussion Matrix Implementation

MSMEs Province	Lediknas Province	Correction Result	Confusion Matrix
DIY	Daerah Istimewa Yogyakarta	Daerah Istimewa Yogyakarta	TP
Kab. Bandung		Banten	FN
Aceh Jaya		Nanggroe Aceh Darussalam	FP
Bantul			TN

Table 1 shows that when Lediknas and Data Correction results give exactly the same results, the Confusion Matrix results are True Positive (TP). When Lediknas data is unavailable, and Data Correction gives certain results, the Confusion Matrix results are False Negative (FN) where these results can be dangerous, especially in large quantities – in this case, reference data itself cannot give certain results but corrected data gives wrong results, in contrast to FN, False Positive (FP) on the reference data the data is available but the correction results are correct and on True Negative (TN), which is where the data is equally unavailable. Then the results of the entire matrix will be calculated for Accuracy, Precision and Recall values.

4. Results and Discussion

4.1. Data Preparation

The data used in this research is the MSMEs obtained from PT. MDS that has at least one data, either Province or City data. Therefore, before implementing the algorithm, we need to filter data. The filtering process shows that 98,701 data did not have province and City data from all the data obtained.

Table 2. Before and After Data Cleaning

	Before		After	
	Prov	City	Prov	City
count	241.267	241.267	142.566	142.566
unique	1.632	3.520	1.632	3.520

Table 2 shows the total MSMEs location data collected from PT. MDS is 241,267 then eliminating data that does not have Province and City data with a total of 142,566 data, of which there are 141,842 Province data and 142,554 City data means there are 724 MSMEs data that have City data but do not have Province data.

In addition, as mentioned in the Method section, after the data is collected, the library expansion process is implemented on Lediknas data, both Province and City data. Library expansion is basically a word extension process in the data. A more specific extension means that one clause extends another clause by adding something new, giving an exception, or offering an alternative [25]. Province data obtained from Lediknas contained 34 data which was then expanded to 64 data. Some of the data expansion on the Lediknas data can be seen in Table 3 below:

Table 3. Province Data Expansion

Province	Province Expansion
Daerah Istimewa Yogyakarta	DIY
Daerah Istimewa Yogyakarta	Jogja
Daerah Istimewa Yogyakarta	Yogyakarta

Table 3 shows that Lediknas data for “Daerah Istimewa Yogyakarta” is expanded to “DIY”, “Jogja”, and “Yogyakarta”. Many people commonly use these three words to refer to or abbreviate the province of “Daerah Istimewa Yogyakarta”. In addition, in the Lediknas City data, there were 514 data which was later expanded to 513 data. One of the results of implementing data expansion on City Lediknas data can be seen in Table 4 below:

Table 4. Province Data Expansion

City	City Expansion
Surakarta	Solo
Kota Bandung	Kota/Kabupaten Bandung
Kabupaten Bandung	Kota/Kabupaten Bandung

One of the implementations of expansion on City data is data “Surakarta” becomes “Solo” as well as other data. Because the City in Indonesia data has many of the same data, from a total of 514 data, 52 City data are combined into one so that the data is reduced to 26 data. So that from 514 data, it was reduced to 488 data which was then expanded to 513 data.

Using the library expansion method, 66,915 Province data and 78,661 MSMEs City data successfully matched

Lediknas data, where this number shows that the library expansion can improve around 46.94% of Province data and 55.18% of City data. So the MSMEs data included in other pre-processing is 75,651 Province data and 63,905 City data.

4.2. Data Preprocessing

Data Processing is implemented by removing unwanted characters such as question marks, hash marks, brackets, quotation marks, and others [24]. In detail, this step will show the results of pre-processing data implementation**.**

4.2.1. Case Folding

Case folding (CF) was implemented after data from Lediknas had been through the Library Expansion process. Case folding is implemented on all data from Lediknas and MSMEs location data from PT. MDS.

Table 5. Case Folding Implementation

Province	Province *CF*	City	City CF
Jakarta	jakarta	Jakarta Pusat	jakarta pusat
DKI Jakarta	dki jakarta	Jakarta Barat	jakarta barat
DKI Jakarta	dki jakarta	Jakarta Timur	jakarta timur

Table 2 shows results on Province and City data from MSMEs and Case Folding results. The Province column shows that “DKI Jakarta” become “dki jakarta”, and it shows in the CF Province column. Table 2 also shows in City column data, “Jakarta Pusat” become “jakarta pusat”, etc.

4.2.2. Replace the Sentence

Replace Sentence (RS) only implemented on City data from MSMEs data because most of Province data is already in root word format. So in this process will be done by removing words such as “Kabupaten” and “Kota” and their derivatives such as “Kab.” to get the name root word from City data.

Table 6. Replace Sentence Implementation

City	City *CF*	City *RS*
Kab. Serang	kab. serang	serang
Kota Bandung	kota bandung	bandung
Kabupaten Bandung	kabupaten bandung	bandung

One implementation of Replace Sentence in City data for MSMEs data in Table 6 is “kab. serang” becomes “serang”, “kota bandung” becomes “bandung” and another implementation is “kabupatan bandung" becomes "bandung".

4.2.3. Remove Punctuation

This Remove punctuation (RP) was implemented after the sentence Replace process. This step is implemented on Province and City MSMEs data from PT. MDS.

In Table 7, the implementation of Remove Punctuation in MSMEs Province data can be seen in the columns CF Province and City CF where one of the data is “prov. jakarta” to “prov jakarta”. Where in City data one of them is “kab. serang” becomes “kab serang”.

Table 7. Remove Punctuation Implementation

Province *CF*	Province *RP*	City *CF*	City
prov. jakarta	prov jakarta	jakarta utara	jakarta utara
provinsi banten	provinsi banten	kab. serang	kab serang
jawa barat	jawa barat	kota/kab bandung	kotakab bandung

Based on several pre-processing steps completed on MSMEs data, there are 75,651 Province data and 63,905 City data, 61,657 Province data and 58,990 City data that already match with Lediknas data. It means around 81.50% of the province and 92.31% of City data have been successfully corrected using the pre-processing method.

4.3. Data Correction

Based on the results of data pre-processing, data needed to be corrected is 13,994 MSMEs Province data and 4,915 MSMEs City data. From this data, 11,740 provinces and 1,659 cities are target data. Targeted data means data is categorised as fixable and included in the focus of this research.

4.3.1. Results of Jaro-Winkler Distance Implementation on Province Data

Fig. 8 Jaro-Winkler Distance Province Data

Figure 8 shows that 12,984 Province data were successfully got a similarity score between MSMEs location and Lediknas location using Jaro-Winkler Distance where total data correct 12,146 and 838 incorrect, which means around 86.79% of the overall Province data and 93.55% which got similarity score from MSMEs Province data from PT. MDS.

Fig. 9 Jaro-Winkler Distance Target Province Data

Figure 9 shows that number of correct data for targeted Province data is 11,663 and 35 data are incorrect, which means that around 99.34% of all target data and 99.70% of all data that got a score from the Jaro-Winkler Distance algorithm this mean good. The algorithm can fix almost all of the MSMEs Province data, which has problems because almost 100% of the data matches Lediknas data.

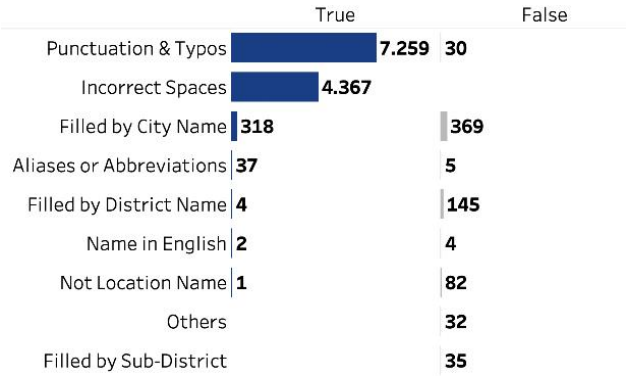

Fig. 10 Province Classification Issue with Jaro-Winkler Distance

The following are details of each main issue and correction data results using Jaro-Winkler Distance on Province data:

- There are 7,256 data has been fixed from the problem "Punctuation & typos", and 30 data that is unfixed/incorrect. This means usage of the Jaro-Winkler Distance is very significant in this problem because the value is equal to 99.58%
- There are 4,367 data out of a total of 4,367 data from the "Incorrect Spaces" problem that has been fixed. This means that Jaro-Winkler Distance can fix 100% "Incorrect Spaces" problem in Province data.
- There are "Aliases or Abbreviations" problems containing 37 correct data and 5 incorrect data. Where are these numbers show that the amount of data with correct correction is greater than 88.09%

Otherwise, data with incorrect results is not the main problem and does not enter into the scope of research, such as the problem of Province data which is "Filled by City Name" with a total of 369 data and the problem data "Filled by District Name" with a total of 145 data.

Based on the explanation above, the implementation of Data Corrections in MSMEs Province data using Jaro-Winkler Distance is good for solving the main problem in this research, especially in problems such as "Punctuation & typos" and "Incorrect Spaces" are high. However, the issue of "Aliases or Abbreviations" is low because this problem has been covered a lot in the data preparation and pre-processing step.

4.3.2. Results of Jaro-Winkler Distance Implementation on City Data

Fig. 11 Jaro-Winkler Distance City Data

Figure 11 shows that 3,734 City data were successfully got a similarity score between MSMEs location and Lediknas location using Jaro-Winkler Distance where total data correct 1,722 and 2012 were incorrect, which means only about 35.05% of the overall City data and 46.12% were successfully matched with Lediknas data.

Fig. 12 Jaro-Winkler Distance Target City Data

Figure 12 shows that the number of correct data for targeted City data is 11,460 and 159 data are incorrect, which means that around 88.0% of all target data and 90.18% of all data that got a score from the Jaro-Winkler Distance algorithm means good because the algorithm can fix almost all of the MSMEs Province data.

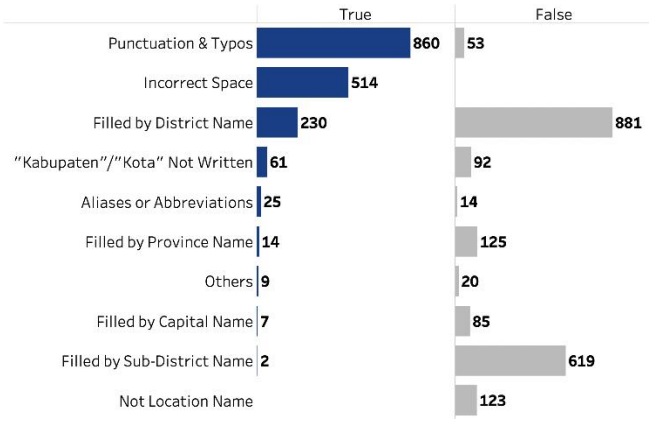

Fig. 13 City Classification Issue with Jaro-Winkler Distance

The following are details of each main issue and correction data results using Jaro-Winkler Distance on City data:

- There are 890 data has been fixed from the "Punctuation & typos" problem and 53 data that are unfixed/incorrect. This means usage of the Jaro-Winkler Distance is very significant in this problem because the value is equal to 94.19%
- There are 514 data out of a total of 514 data from the "Incorrect Spaces" problem that has been fixed. This means that Jaro-Winkler Distance can fix 100% "Incorrect Spaces" problem in City data.
- There are 61 data with "Kabupaten/Kota Not Written" have been fixed, and 92 data is unfixed/incorrect. It shows that the total data incorrect is higher than the correct data with 60,13%
- There are "Aliases or Abbreviations" problems containing 25 correct data and 14 incorrect data. It shows that the total data incorrect is higher than the correct data, with 64.10%

Based on the explanation above, implementation of Data Corrections in MSMEs City data using Jaro-Winkler Distance have quite high fixed/correct data, especially on the problems "Punctuation & typos", "Incorrect Spaces" and "Aliases or Abbreviations".

4.3.3.Results of Cosine Similarity Implementation on Province Data

The total data entered into the Data Pre-processing step is 13,994 data, and there were only 6,958 data that succeeded in getting a similarity score using the Cosine Similarity algorithm. It means that only 49.72% of all data can be identified its similarity to Lediknas data.

Fig. 14 Cosine Similarity Province Data

Figure 14 shows that number of correct data is 5,370, 1,360 has correct and incorrect results and 228 incorrect results, which means around 38.37% of the data was correct from all data entered into the pre-processing step. However, the total data correct from all data successfully got a similarity score are 77.18%. Data with more than one result (correct and incorrect) are basically data that has matched more than one data in Lediknas, leading to ambiguity. Data with high variability of data types and formats are possibly ambiguous and low quality [26].

Fig. 15 Cosine Similarity Target Province Data

Figure 15 shows that number of correct data in the Province Data Target is 4,843; 1,355 has correct and incorrect results and 38 incorrect results, which means only 41.25% of all data target was correct from all data entered into the pre-processing step. However, the total data correct from all data successfully got a similarity score are 77.66%. This percentage shows that correction results using Cosine Similarity on the target data has a higher percentage than overall data.

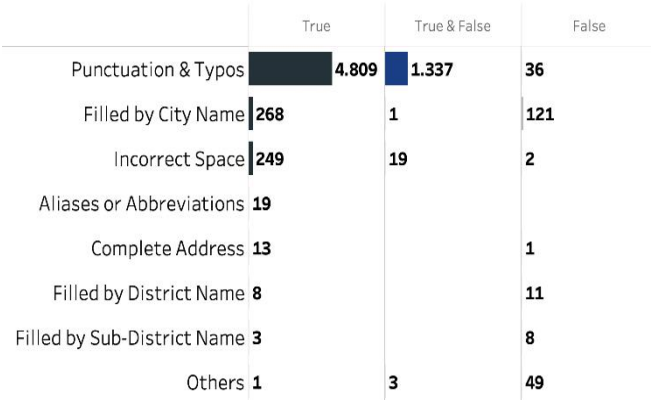

Fig. 16 Province Classification Issue with Cosine Similarity

The following are details of each main issue and correction data results using Cosine Similarity on Province data:

- There are 4,809 data has been fixed from the problem "Punctuation & typos"; 1,337 data have an incorrect and correct result, and 36 data are incorrect. It means that 77.79% of the data is validated correctly.
- These 268 data have been fixed from the problem "Filled by City Name"; 121 data have an incorrect and correct result, and 36 data are incorrect. It means that 68.71% of the data is validated correctly.
- There are 249 data that have been fixed from the problem "Incorrect Spaces", 19 data that have an incorrect and correct result and 2 data that are incorrect. It means that 92.22% of the data is validated correctly.
- There are 19 data that have been fixed from the problem "Aliases or Abbreviations" this means all data is 100% correct.

Based on the explanation above, implementation of Data Corrections in MSMEs Province data using Cosine Similarity can solve several main problems in this research, especially in problems such as "Punctuation & typos" and "Incorrect Spaces" and "Aliases or Abbreviations". However, data with the problem "Filled by City Name" was successfully fixed with this algorithm.

4.3.4. Results of Cosine Similarity Implementation on City Data

Total all data entered into the Data Pre-processing step is 4,915 data; there were only 1,752 data that succeeded in getting a similarity score using the Cosine Similarity algorithm. It means that only 35.64% of all data can be identified its similarity to Lediknas data.

Fig. 17 Cosine Similarity City Data

Figure 17 shows that number of correct data is 730, 58 have correct and incorrect results and 964 incorrect results, which means around 14.85% of the data was correct from all data entered into the pre-processing step. However, the total data correct from all data successfully got a similarity score are 41.67%.

Fig. 18 Cosine Similarity Target City Data

Figure 15 shows that number of correct data in the City data target is 443, 13 have correct and incorrect results, and 380 have incorrect results, which means only 26.70% of all data target was correct from all data entered into the pre-processing step. However, the total data correct from all data successfully got a similarity score are 52.99%. This percentage shows that correction results using Cosine Similarity on the target data has a higher percentage than overall data.

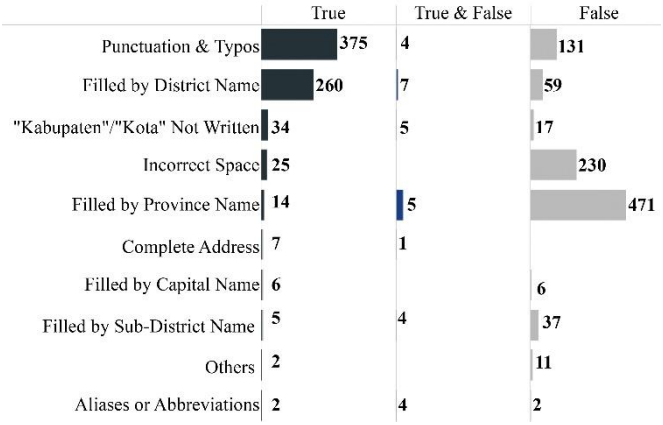

Fig. 19 City Classification Issue with Cosine Similarity

The following details each of the main problems and the results of corrections using Cosine Similarity in City/District data:

- These 375 data have been fixed from the problem "Punctuation & typos," 4 data have the incorrect and correct result, and 131 data are incorrect. It means that 73.52% of the data is validated correctly.
- These 260 data have been fixed from the problem "Filled by District Name"; 7 data have the incorrect and correct result, and 59 data are incorrect. It means that 79.75% of the data is validated correctly.
- There are 34 data with "Kabupaten/Kota Not Written" as correct, 5 data have incorrect and correct results, and 17 data are incorrect. It means that 60.71% of the data is validated correctly.
- These 25 data have been fixed from the problem "Incorrect Spaces", and 230 2 data is incorrect. It means that 9.8% of the data is validated correctly.

Based on the explanation above, implementing Data Corrections in MSMEs City data using Cosine Similarity can solve several main problems in this research. However, data with the problem "Filled by District" was in the second position and has been fixed, while other problems show a low percentage.

4.3.5. Comparison of Cosine Similarity and Jaro-Winkler Distance Results

Based on observation when implementing Cosine Similarity and Jaro-Winkler Distance algorithms, researchers found that implementation using Jaro-Winkler Distance was easier to implement on MSMEs location data compared to Cosine Similarity. This statement appears because the implementation of Cosine Similarity requires several batches of groups to get a similarity score on all research data using Google Colabs, besides that before implementing the Cosine Similarity algorithm itself, text data must be changed into a matrix using TF-IDF since Cosine Similarity reads similarities based on the matrix.

The results of both implementation algorithms show a significant difference. These differences show in data Province using the Cosine Similarity algorithm only improves target data by 41.25% while the Jaro-Winkler Distance algorithm succeeds in improving target data by up to 99.34%. It also happened for City data while using the Cosine Similarity algorithm can only improve target data by 26.70%, while the Jaro-Winkler Distance algorithm succeeds in improving target data up to 88.0%. This significant difference occurs due to the way the algorithm works.

The Cosine Similarity algorithm only reads the similarity in words level; if no word matches Lediknas data, then the score will be zero. For example, the location data for one of the MSMEs is "Kec. Losarang Kab. Indramayu", which is matched with the reference data "Kabupaten Indramayu" because both data show the exact word "Indramayu", the Cosine Similarity algorithm will give a certain score. Still, if there is a typo in the data, such as "Kec. Losarang Kab. Indramyu", where the word "Indramayu" is written "Indramyu", then this algorithm will give zero scores.

The Jaro-Winkler Distance algorithm only reads similarities in string level. So, this algorithm can read small details of a word and makes percentage data corrected using the Jaro-Winkler Distance algorithm is higher than that of Cosine Similarity. For example, MSMEs location data "Kabupaten Indramyu" is matched with reference data "Kabupaten Indramayu" because the writing error only misses one letter and all the words are in good order, the Jaro-Winkler Distance algorithm will give a certain score. In this condition, no matter how small and how big the error in writing in the text, the Jaro-Winkler Distance algorithm will give a similarity score.

4.4. Data Integration

This section explains the data integration result, which only focuses on completing missing Province data. Province data that is missing is basically data that was not completed by MSMEs when registering an application, but MSMEs completed City data where from data collected, 724 data fell into that criteria.

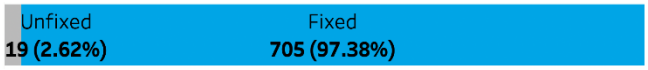

Fig. 20 Data Integration Result

Figure 20 shows that 705 data from the data Province included in this criteria were successfully completed, which means that the data/information integration process successfully completed the missing data by 97.38%, of which only about 2.62% of the data could not be completed.

4.5. Data Combination

This section explains combining data from Data Pre-processing, Data Correction and Data Integration processes

to determine whether all the combined steps can improve the quality of MSMEs location data. To be able to run the combination process properly, where the process is divided into two parts, which are Data Combination (1) and Data Combination (2). As explained in the material and methods in the Data Combination section (2), the data is taken according to the rules; if the Cosine Similarity has a high score, then use the results from the Cosine Similarity. The high definition uses a quantile distribution with an upper quantile for each data which can be seen in Figure 21.

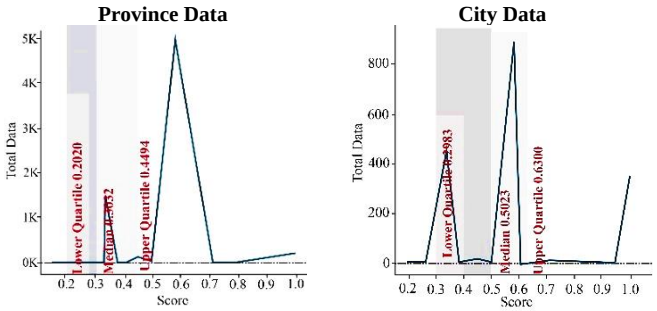

Fig. 21 Province and City Data Distribution

Figure 21 shows the distribution of score similarity in Province and City data. In Province data, the Lower Quartile value is 0.16, the Median is 0.25, and the Upper Quartile is 0.41, while in City data, the Lower Quartile value is 0.2061, the Median is 0.3555, and the Upper Quartile is 0.5101. When the Cosine Similarity algorithm score on Province and City data is greater than the upper quantile, the result of Cosine Similarity will be used. If it is not, then data result from the Jaro-Winkler Distance algorithm will be used.

As mentioned in the data collection process, there are 142,566 data, wherein the data preparation steps, the data was successfully corrected is 66,915. In the data pre-processing step, the data was successfully matched is 61,657, and in the data correction step using the Jaro-Winkler Distance algorithm, successfully bringing a similarity score for 12,948 Province data which 12,146 were correct and successfully bringing similarity score for 3,734 City data which 1,722 data were correct. However, the Cosine Similarity algorithm was successfully bringing a similarity score for 6,958 Province data which 6,731 were correct and was successfully bringing a similarity score for 1,752 City data which 788 data were correct.

The result of Data Combination in Province data from all data (141,842) there are 141,589 (99.82%) of data has been corrected, and in City data from all data (142,554) there are 141,634 (99.35%) of data has been corrected. Some data cannot be corrected because the data does not meet the target data criteria. In other words, the data has problems that are not included in the scope of research, such as “Filled by Island Name”, “Filled by City Name”, “Filled by District Name”, and "Not Location Name”.

4.6. Result and Evaluation

4.6.1. Province Data Results and Evaluation

The Confusion Matrix on Province data can be seen in Table 8:

Table 8. Province Data Confusion Matrix

	Actual: Positive (1)	**Actual: Negative (1)**
Correction: Positive (1)	TP: 140.856	FN: 109
Correction: Negative (1)	FN: 84	TN: 540
	140.940	649

Based on the Confusion Matrix values in Table 8, for Province data, the accuracy is 99.86%, which shows that Province data has a very high level of closeness of correction and actual value. The Precision value is 99.92% which indicates that the amount of relevant information obtained from the research process is very high, and the Recall value is 99.94% which indicates that the relevant information contained in the information is also very high.

4.6.2.Results and Evaluation of City Data

The Confusion Matrix on City data can be seen in Table 9:

Table 9. City-Data Confusion Matrix

	Actual: Positive (1)	**Actual: Negative (1)**
Correction: Positive (1)	TP: 139.492	FN: 72
Correction: Negative (1)	FN: 197	TN: 1973
	139.689	2045

Based on the Confusion Matrix values in Table 9, for City data, the accuracy is 99.81%, which shows that City data has a very high level of closeness of correction and actual value. The Precision value is 99.94%, indicating that the amount of relevant information obtained from the research process is very high. The Recall value is 99.85%, indicating that the information's relevant information is also very high.

5. Conclusion

Based on the results of the research and analysis that has been implemented, the following are the conclusions for this research:

1. The data Preparation process can clear data that are correct but have aliases or abbreviations by 46.94% in Province data and 55.18% in City data.

2. Data Pre-processing method (Case Folding, Replacing Sentences and Removing Punctuation) managed to improve data by 81.50% in Province data and 92.31% in City data

3. In the implementation of Data Correction, there are different results for each data and method; here are the details:
   - Data Correction using Jaro-Winkler Distance shows that 99.34% Province data target and 88.0% of the City data target are correct and can fix the data's main problem.
   - Data Correction using Cosine Similarity shows that 41.25% Province data target and 26.70% of the City data target are correct. Besides can be fixed the main problem, another problem, such as "Filled by District Name", was successfully corrected in second place.

4. Due to the process, the Jaro-Winkler Distance algorithm is easier to implement than the Cosine Similarity algorithm. Implementing the Cosine Similarity algorithm requires changing text data into a matrix using TF-IDF and several batches to get all data scores, which takes more time.

5. The Jaro-Winkler distance algorithm calculates text similarity based on a string of words. The advantage of the Jaro-Winkler Distance algorithm is that it can read the smallest errors from a text, while the disadvantage is that it cannot work well for comparisons of long data text.

6. The Cosine Similarity algorithm calculates the similarity of text based on words. The advantage of the Cosine Similarity algorithm is that it can work well on long texts if there are appropriate words; it means that Cosine Similarity can work on a larger scale, while the disadvantage is that it cannot work well on data that has small errors such as incorrect spaces, typos and can give high ambiguity of similarity result.

7. The implementation of Data integration to fill missing data can improve 97.38% of data missing data. Where the data cannot be corrected comes from data with City have another issue than the main issue in this research. So, it does not produce good matches based on Data Preprocessing and Data Correction.

8. Data Combination using the 1st Data Combination gives maximum results for both Province and City data, which can improve 99.36% of all Province data and 97.99% of all City data. The 2nd Data Combination can improve the quantity of, especially, Province data.

9. The evaluation using Confusion Matrix on Province data shows that the accuracy of the data is 99.86%, Precision 99.92% and Recall 99.94%. Whereas City data show that accuracy is 99.81%, precision is 99.94%, and Recall is 99.85%.

Acknowledgments

I would like to thank all those who have supported me in completing the implementation I presented in this paper. A very special thanks go to Management, the Principal, Head of Department & Faculty of Information Technology Bina Nusantara University, Sani Muhamad Isa, for all his support & providing me with the necessary resources to complete the development work.

References

[1] Thabit Hassan Thabit, and Manaf Raewf, "The Evaluation of Marketing Mix Elements: A Case Study," *International Journal of Social Sciences & Educational Studies,* vol. 4, no.4, pp. 100-109, 2018. *Crossref,* http://dx.doi.org/10.23918/ijsses.v4i4p100

[2] Nurul Indarti, "Business Location and Success: The Case of Internet Cafe Business in Indonesia," *Gadjah mada International Journal of Business,* vol. 6, no. 2, pp. 171-192, 2004. *Crossref,* https://doi.org/10.22146/gamaijb.5543

[3] Rajkumar, P, "A Study of the Factors Influencing the Location Selection Decision of Information Technology Firms," *Asian Academy of Management Journal*, vol. 18, no. 1, pp. 35–54, 2013.

[4] Hakim, B, "Data Text Pre-Processing Sentiment Analysis in Data Mining using Machine Learning," *Journal of Business and Audit Information Systems.*, vol. 4, no. 2, pp. 16-22, 2021.

[5] Jaka, A. T, "Preprocessing Text to Minimize Meaningless Words in the Text Mining Process," *Informatics Journal,* vol. 1, 2015.

[6] Tumula Mani Harsha et al., "Survey on Resume Screening Mechanisms," *SSRG International Journal of Computer Science and Engineering*, vol. 9, no. 4, pp. 14-22, 2022. *Crossref,* https://doi.org/10.14445/23488387/IJCSE-V9I4P103

[7] Srividhya, V, and Anitha, R, "Evaluating Preprocessing Techniques in Text Categorization," *International Journal of Computer Science and Application Issue,* pp. 49-51, 2010.

[8] Piska Dwi Nurfadila et al., "Journal Classification Using Cosine Similarity Method on Title and Abstract with Frequency-Based Stopword Removal," *International Journal of Artificial Intelligence Research, Crossref,* https://doi.org/10.29099/ijair.v3i2.99

[9] Sugiyamto et al, "Performance Analysis of the Cosine and Jaccard Methods in the Document Similarity Test," *Journal of Informatics Society,* vol. 5, no. 10, pp. 1-8, 2014.

[10] Nurdin et al, "Plagiarism Document Detection Using the Weigh Tree Method," *Telematics Journal*, vol. 1 no. 1, 2019.

[11] Vikas Thad, and Dr Vivek Jaglan, "Comparison of Jaccard, Dice, Cosine Similarity Coefficient to Find Best Fitness Value for Web Retrieved Documents Using Genetric Algorithm," *International Journal of Innovations in Engineering and Technology*, vol. 2, no. 4, pp. 202-205, 2013.

[12] Dedy Kurniadi, Sam Farisa Chaerul Haviana, and Andika Novianto et al., "Implementation of the Cosine Similarity Algorithm in Archive Document System at Sultan Agung Islamic University," *Journal of Transformation*, vol. 17, no. 2, pp. 124-132, 2020.

[13] Joyassree Sen et al., "Face Recognition Using Deep Convolutional Network and One-shot Learning," *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 4, pp. 23-29, 2020. *Crossref*, https://doi.org/10.14445/23488387/IJCSE-V7I4P107

[14] Friendly, "Improvements to the Jaro-Winkler Distance Method for Approximate String Search Using Indexed Data for Multi-User Applications," *Technology Journal*, vol. 04, no. 02 pp. 69 – 78, 2017.

[15] Yulianingsih, "Implementation of Jaro-Winkler and Levenstein Distance Algorithms in Searching Data in Databases," *Journal of Technology Research and Innovation Writing Unit*, vol. 2, no. 1, 2017.

[16] Munjiah Nur Saada et al., "Information Retrieval of Text Document with Weighting TF-IDF and LCS," *Journal of Computer Sciences and Information*, vol. 6, no. 1, 2013. *Crossref*, https://doi.org/10.21609/jiki.v6i1.216

[17] Chunhao Huang et al, "Text Retrieval Technology Based on Keyword Retrieval," *Journal of Physics: Conference Series*, 2020. *Crossref*, https://doi.org/10.1088/1742-6596/1607/1/012108

[18] Lediknas, Provinces Regencies and Cities in Indonesia, 2022. [Online]. Accessed https://www.lediknas.com/provinsi-kabupaten-dan-kota-di-indonesia

[19] Pooja Goyal, Sushil Kumar, and Komal Kumar Bhatia, "Hashing and Clustering Based Novelty Detection," *SSRG International Journal of Computer Science and Engineering*, vol. 6, no. 6, pp. 1-9, 2019. *Crossref*, https://doi.org/10.14445/23488387/IJCSE-V6I6P101

[20] Sidiq, M, "*The Effect of Pre-Process on Sentiment Analysis in Indonesian Language Texts*," *Thesis*, 2019.

[21] Luís Batista, and Luís A. Alexandre, "Text Pre-processing for Lossless Compression," *Data Compression Conference*, pp. 506-506, 2008. *Crossref*, https://doi.org/10.1109/DCC.2008.78

[22] OECD, Data Correction, 2022. [Online]. Available: https://stats.oecd.org/glossary/detail.asp?ID=3402

[23] Ariantini, D. A et al., "Measurement of Similarity to Indonesian Text Documents Using the Cosine Similarity Method," *E-Journal of Computer Science*, vol. 9, no. 1, 2016.

[24] Makmun, Agus, "Performance Study of Similarity Algorithm for Identification and Mapping of SWOT Statements," Muhammadiyah University of Surakarta," Final Project: 2018.

[25] Setyaji, Arso, "Analysis of Taxis Significant Translation in the Novel "The Old Man and the Sea" (Systemic Functional Linguistics Approach)," *Indonesian Surakarta*, 2018.

[26] Wang, Lidong, "Heterogeneous Data and Big Data Analytics," *Automatic Control and Information Sciences*, vol. 3, no. 1, pp. 8-15, 2017. *Crossref*, https://doi.org/10.12691/acis-3-1-3

[27] Khadim, A. I, "An Evaluation of Preprocessing Tehcniques for Text Classfication," *International Journal of Computer Science and Information Security*, vol. 16, no. 6, pp. 22-32, 2018.

[28] Novantara, P, and Pasruli, O, "Implementation of the Jaro-Winkler Distance Algorithm for Plagiarism Detection Systems in Thesis Documents," *Journal of Buffer Informatics*, vol. 3, no. 2, 2017.

[29] Jumeilah, F. S, "Application of Support Vector Machine (SVM) for Research Categorization," *Journal of Systems Engineering and Information Technology*, vol. 1 no. 1, pp. 19-25, 2017.