ANALISIS PERBANDINGAN ALGORITMA *TF-IDF* DAN *FUZZY MATCHING* DALAM MENDETEKSI KEMIRIPAN TEKS

Arif Nur Listanto[1], Muhammad Fauzan Azima, S.Kom[2], M.T.I, Fitria, S.T[3]., M.Kom, Dr. CHAIRANI, S.Kom., M.Eng[4]
[1,2,3,4]Institut Informatika Dan Bisnis Darmajaya, Jl. ZA. Pagar Alam No.93 35141 Kota Bandar Lampung, Indonesia.
Email Penulis Korespondensi: arifnurlis.2011010035@mail.darmajaya.ac.id

Abstrak

Pendeteksian kemiripan teks merupakan komponen penting dalam berbagai aplikasi pemrosesan bahasa alami (NLP), seperti plagiarisme, pencarian informasi, dan analisis sentimen. Penelitian ini membandingkan kinerja dua algoritma populer, yaitu Term Frequency-Inverse Document Frequency (TF-IDF) dan Fuzzy Matching, dalam mendeteksi kemiripan teks. TF-IDF menghitung bobot kata berdasarkan frekuensi kemunculannya dalam dokumen dan seberapa umum kata tersebut di seluruh korpus, sedangkan Fuzzy Matching mengukur kemiripan teks dengan mempertimbangkan kesalahan penulisan dan variasi kata. Melalui eksperimen yang dilakukan pada berbagai set data teks, hasil menunjukkan bahwa TF-IDF lebih unggul dalam mendeteksi kemiripan semantik pada dokumen panjang, sementara Fuzzy Matching lebih efektif dalam mengidentifikasi kemiripan pada teks pendek dengan variasi kata dan kesalahan penulisan. Penelitian ini memberikan wawasan mendalam tentang kelebihan dan kekurangan masing-masing algoritma serta rekomendasi penggunaan berdasarkan karakteristik teks yang dianalisis.

Kata Kunci*– Pendeteksian Kemiripan Teks, TF-IDF, Fuzzy Matching, Pemrosesan Bahasa Alami, Algoritma.*

Abstract

Text similarity detection is an important component in various natural language processing (NLP) applications, such as plagiarism, information retrieval, and sentiment analysis. This study compares the performance of two popular algorithms, namely Term Frequency-Inverse Document Frequency (TF-IDF) and Fuzzy Matching, in detecting text similarity. TF-IDF calculates the weight of a word based on the frequency of its occurrence in the document and how common the word is across the corpus, while Fuzzy Matching measures text similarity by considering spelling

errors and word variations. Through experiments conducted on various text datasets, results show that TF-IDF is superior in detecting semantic similarity on long documents, while Fuzzy Matching is more effective in identifying similarity on short texts with word variations and writing errors. This research provides an in-depth insight into the advantages and disadvantages of each algorithm as well as usage recommendations based on the characteristics of the text being analyzed

Kewords– *Text Similarity Detection, TF-IDF, Fuzzy Matching, Natural Language Processing, Algorithm.*

I. PENDAHULUAN

Dalam era digital yang semakin berkembang, pemrosesan bahasa alami (*NLP*) telah menjadi bidang yang sangat penting, terutama dalam aplikasi seperti pendeteksian plagiarisme, pencarian informasi, dan analisis sentimen. Salah satu tugas utama dalam *NLP* adalah mendeteksi kemiripan teks, yang memerlukan algoritma yang akurat dan efisien. *Term Frequency-Inverse Document Frequency (TF-IDF)* dan *Fuzzy Matching* adalah dua algoritma yang digunakan untuk mendeteksi kemiripan teks[1]. *TF-IDF* berfokus pada menghitung bobot kata berdasarkan frekuensi kemunculan dan keumuman kata, sedangkan *Fuzzy Matchin*g mempertimbangkan variasi kata dan kesalahan penulisan[2]. Penelitian ini berusaha untuk mengevaluasi dan membandingkan kinerja kedua algoritma tersebut dalam berbagai konteks teks, dengan tujuan memberikan gambaran yang lebih jelas bagaimana menggunakan kedua algoritma ini, dan bagaimana mengkombinasikan keduanya menghasilkan hasil yang akurat dalam proses pendeteksian kemiripan teks.

II. METODE PENELITIAN

Penelitian ini menggunakan metode kuantitatif untuk membandingkan kinerja algoritma *Term Frequency-Inverse Document Frequency (TF-IDF)* dan *Fuzzy Matching* dalam mendeteksi kemiripan teks. Berikut merupakan beberapa langkah yang dilakukan dalam penelitian ini:

2.1 Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini merupakan kumpulan data teks yang dikumpulkan dari berbagai sumber, termasuk dokumen berupa *spreadsheet.*

2.2 Pra-Pemrosesan Data

Pemrosesan Data (DP) mengacu pada ekstraksi informasi melalui pengorganisasian, pengindeksan, dan manipulasi data. Informasi di sini berarti hubungan dan pola yang berharga yang dapat membantu memecahkan masalah yang diminati. Dalam sejarahnya, kemampuan dan efisiensi DP telah meningkat seiring dengan kemajuan teknologi[3].

2.3 Implementasi Algoritma

Algoritma *TF-IDF* dan *Fuzzy Matching* diimplementasikan menggunakan bahasa pemrograman *Python* dan Pustaka *NLP* yang relavan diantaranya *Scikit-Learn* untuk *TF-IDF* dan *fuzzywuzzy* untuk *Fuzzy Matching*

2.3.1 Algoritma *TF-IDF*

TFIDF adalah sebuah metode yang merupakan integrasi antar *term frequency (TF),* dan *inverse document frequency (IDF). Term Frequency* dihitung menggunakan Persamaan dengan term frequency ke-i adalah frekuensi kemunculan term ke-i dalam dokumen ke-j. *Inverse Document Frequency (IDF)* adalah logaritma dari rasio jumlah seluruh dokumen dalam korpus dengan jumlah dokumen yang memiliki term yang dimaksud seperti yang dituliskan secara matematis pada Persamaan[4][5]. Berikut merupakan rumus dari algoritma *TF-IDF*

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t} inv\, f_{t} inv, d}$$

Gambar 1 Rumus *TF (Term Frequency)*

$$idf(t, D) = \log \frac{N}{|\{d : d \in D \ and \ t \in d\}|}$$

Gambar 2 Rumus *IDF (Inverse Document Frequency)*

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

Gambar 3 Rumus *TF-IDF (Term Frequency-Inverse Document Frequency)*

2.3.2 Algoritma *Fuzzy Matching*

Fuzzy Matching adalah teknik yang digunakan untuk mengidentifikasi tingkat kesamaan antara string, yang sangat berguna untuk penyelarasan ontologi di mana pencocokan yang tepat tidak selalu memungkinkan karena adanya variasi data. Makalah ini membahas beberapa algoritme untuk pencocokan string *fuzzy*, termasuk *Jaro-Winkler dan Levenshtein*[6][7], yang membantu mendeteksi perkiraan kecocokan dengan mempertimbangkan faktor-faktor seperti pengurutan karakter dan jarak pengeditan. Pendekatan ini memungkinkan penyelarasan istilah ontologis yang lebih fleksibel dan akurat dengan menangkap nuansa leksikal dan semantik[8].

2.4 Eksperimen dan Evaluasi

Eksperimen dilakukan dengan menerapkan kedua algoritma pada dataset yang telah di pra-proses. Selanjutnya hasil eksperimen pada kedua algoritma tersebut akan dibandingkan kinerjanya berdasarkan tingkat persentase deteksi kemiripan teks. Dan yang terakhir eksperimen untuk menggabungkan kedua algoritma tersebut, serta membandingkannya dengan masing-masing algoritma.

Dengan beberapa metode penelitian ini, diharapkan memberikan gambaran yang cukup jelas tentang kinerja dari algoritma *TF-IDF* dan *Fuzzy Matching* serta kinerja yang dihasilkan apabila kedua algoritma tersebut digabungkan dalam mendeteksi kemiripan teks.

III. HASIL DAN PEMBAHASAN

3.1. Hasil Analisis

Dalam hasil analisis pada penelitian ini menggunakan *tools Python* dan menggunakan *Google Colab* sebagai lingkungan pengembangan. *Google Colab* dipilih karena kemudahan aksesnya. Sumber daya komputasi yang memadai untuk menjalankan algoritma *TF-IDF* dan *Fuzzy*

Matching, serta kemampuannya untuk berkolaborasi secara real-time dalam proses analisis data teks.

3.1.1 Analisis Algoritma *TF-IDF*

Data yang akan diujikan pada analisis ini adalah sebagai berikut:

Document 1: the cat sat on the mat
Document 2: the quick brown fox
Document 3: the dog barks loudly
Document 4: the fox and the hound

Query: the cat was brown

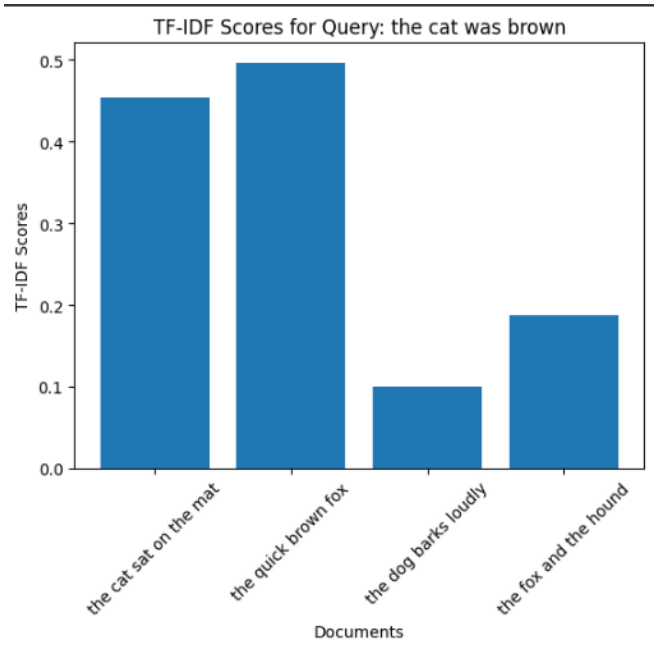
Langkah selanjutnya adalah pra-proses data menjadi vektor menggunakan *library* dari *Python* yaitu *Scikit-Learn.*

```
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(document_example)
print(tfidf_matrix.shape)
```

```
  (0, 8)        0.4432738148936904
  (0, 9)        0.4432738148936904
  (0, 11)       0.4432738148936904
  (0, 3)        0.4432738148936904
  (0, 12)       0.46263733109032296
  (1, 5)        0.4634579560648343
  (1, 2)        0.5878376510497553
  (1, 10)       0.5878376510497553
  (1, 12)       0.3067580723906352
  (2, 7)        0.5528053199908667
  (2, 1)        0.5528053199908667
  (2, 4)        0.5528053199908667
  (2, 12)       0.2884767487500274
  (3, 6)        0.5191134919154226
  (3, 0)        0.5191134919154226
  (3, 5)        0.40927503962898937
  (3, 12)       0.5417899103521974
```

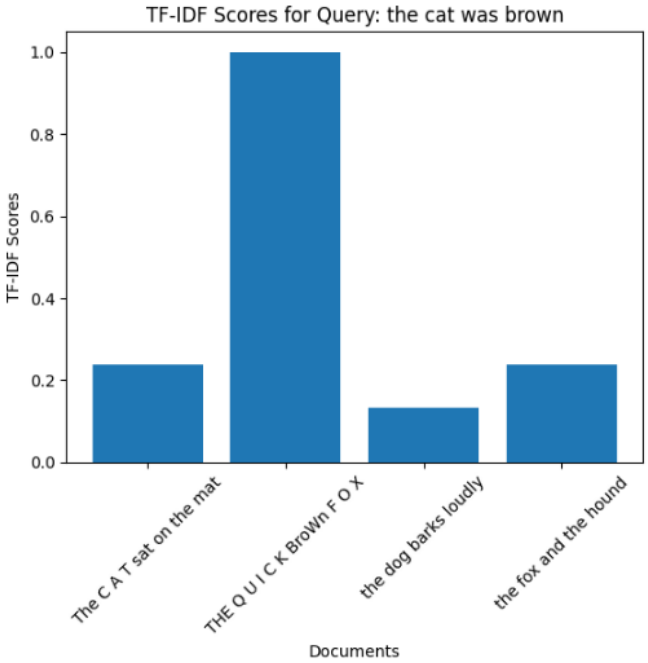
Gambar 4 Data Pra-Proses

Berdasarkan gambar diatas, hasil didapatkan dari mengubah *document* menjadi vektor. Selanjutnya pada penelitian ini dilakukan test kemiripan teks dengan menggunakan algoritma *TF-IDF*, dan didapatkan hasil sebagai berikut:



Gambar 5 Hasil Kemiripan Teks

Berdasarkan gambar diatas kemiripan untuk *query "the cat was brown"* terhadap *document* memiliki persentase maksismum sekitar 50%, dan untuk yang memiliki tingkat kemiripan tertinggi ialah pada *document "the quick brown fox"* dan *"the cat sat on the mat"*. Dari hasil analisi ini diketahui bahwa *TF-IDF* memiliki kelemahan saat beberapa huruf diubah menjadi huruf besar. Hasil dapat terlihat pada gambar dibawah ini.



Gambar 6 Kelemahan *TF-IDF*

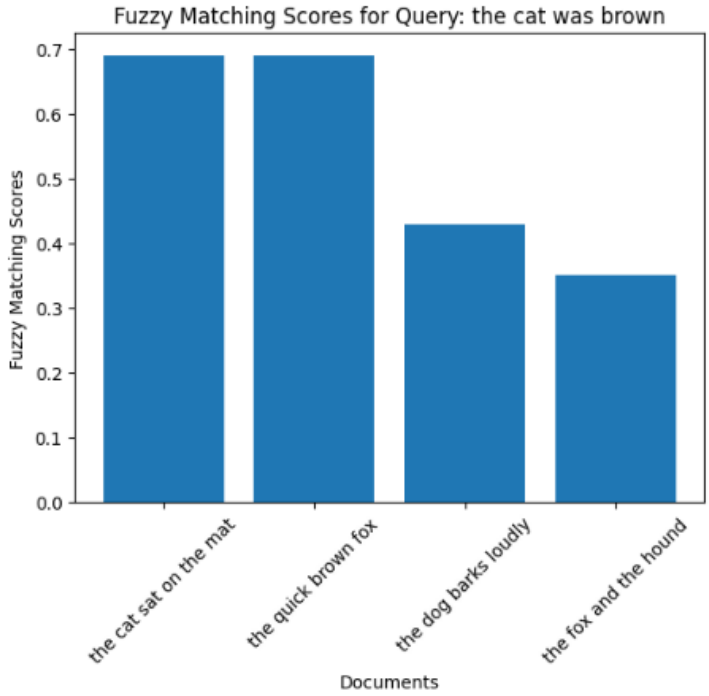
3.1.2 Analisis Algoritma *Fuzzy Matching*

Dalam analisis *Fuzzy Matching* ini data yang digunakan sama dengan data yang digunakan untuk analisis algoritma *TF-IDF*. Data yang akan diujikan pada analisis ini adalah sebagai berikut:

Document 1: the cat sat on the mat
Document 2: the quick brown fox
Document 3: the dog barks loudly
Document 4: the fox and the hound

Query: the cat was brown

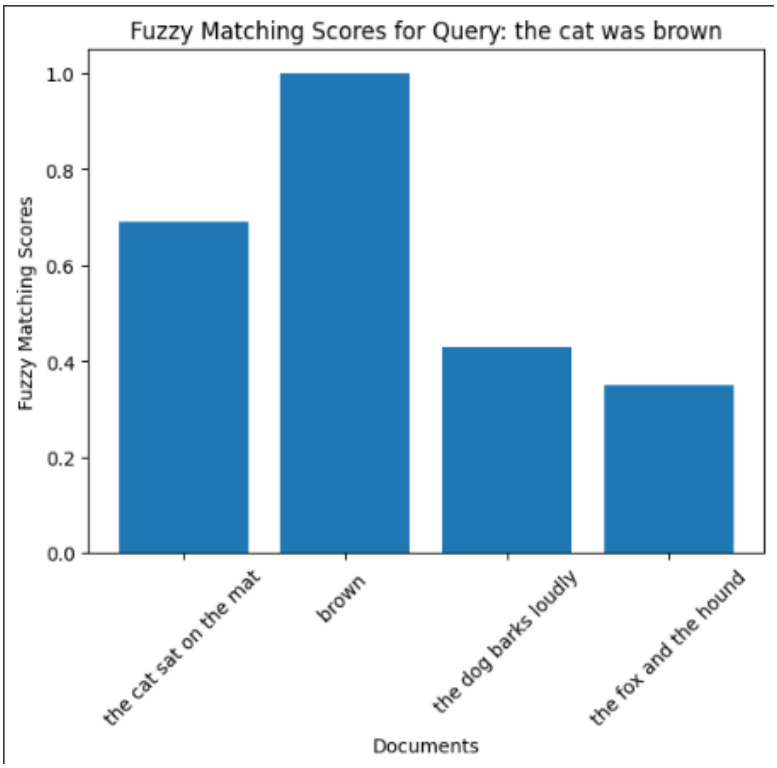
Pada analisi *Fuzzy Matching* data tidak perlu untuk di *vectorize* seperti pada *TF-IDF*. Dan untuk hasil pada analisis ini adalah sebagai berikut:



Gambar 7 Hasil *Fuzzy Matching*

Dalam hasil analisis terhadap *Fuzzy Matching* dapat terlihat bahwa hasil yang ditunjukan sama sepert *TF-IDF* yang mana pada *Document "the cat sat on the mat"* dan *"the quick brown fox"* memiliki hasil yang sama. Tetapi sama seperti *TF-IDF* algoritma ini memiliki kelemahannya

sendiri, apabila salah satu *document* di hapus dan disisakan 1 kata, maka algoritma *Fuzzy Matching* menghasilkan *False Detection* atau bias pada deteksinya. Untuk hasil analisisnya dapat terlihat pada gambar dibawah ini:



Gambar 8 Kelemanan *Fuzzy Matching*

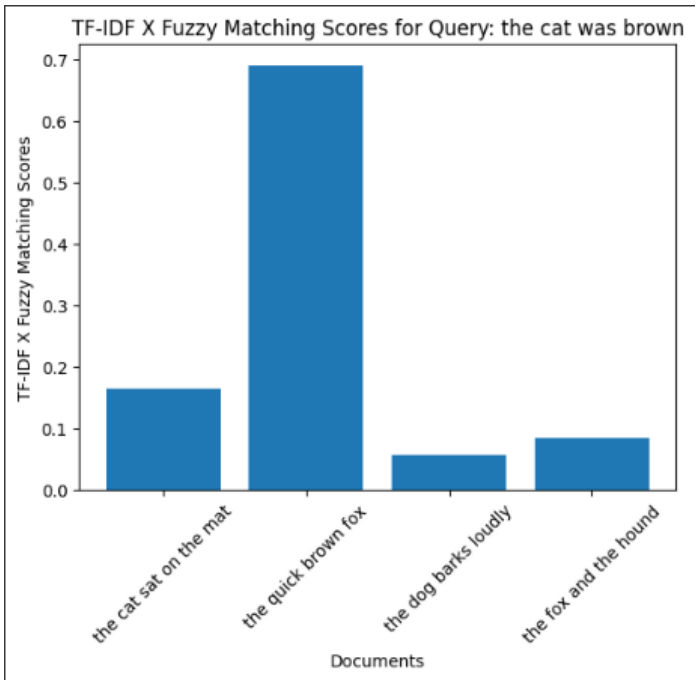
3.1.3 Analisis Kombinasi Algoritma *TF-IDF* dan Fuzzy Matching

Dalam analisis kombinasi kedua algoritma ini, data yang digunakan tetap sama, yaitu:

Document 1: the cat sat on the mat
Document 2: the quick brown fox
Document 3: the dog barks loudly
Document 4: the fox and the hound

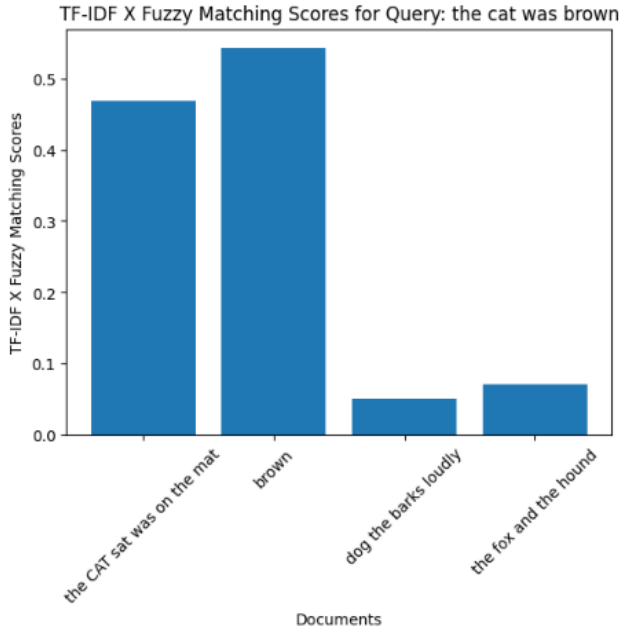
Query: the cat was brown

Pada analisis ini, dapat terlihat bahwa hasilnya memiliki tingkat akurasi yang cukup, dapat terlihat pada gambar dibawah ini.



Gambar 9 Hasil analisis kombinasi algoritma *TF-IDF* dan *Fuzzy Matching*

Berdasarkan hasil analisis pada gambar diatas, terlihat bahwa hanya *document "the quick brown fox"* yang memiliki tingkat kemiripan lebih dari 50%, dan sekitar 70%, sedangkan yang lainnya memiliki tingkat kemiripan dibawah 50%. Dan apabila beberapa *document* diubah, maka akan mereduksi bias yang dihasilkan oleh *Fuzzy Matching* maupun *TF-IDF* dengan membuat tingkat kemiripannya dibawah 60%, hasilnya dapat terlihat pada gambar dibawah.



Gambar 10 Hasil kombinasi algoritma *TF-IDF* dan *Fuzzy Matching* terhadap perubahan data

IV. KESIMPULAN

Kesimpulan dari penelitian ini adalah kedua algoritma ini memiliki kelemahannya masing-masing, tetapi apabila kedua algoritma ini digabungkan maka akan menghasilkan tingkat akurasi yang cukup untuk mendeteksi kemiripan teks. Tetapi kedua algoritma ini masih memiliki banyak kekurangan, salah satunya ialah dihadapkan pada teks yang sangat panjang seperti *document* yang memiliki lebih dari sekitar 100 kata. Untuk itu masih terdapat algoritma lain yang mampu untuk menghasilkan deteksi tingkat kemiripan teks yang akurat, bahkan untuk *document* yang memiliki kata lebih dari 100 diantarnya adalah *Large Language Model, Winnowing, LCS (Longest Common Subsequence),* dan masih banyak lagi. Alasan saya melakukan penelitian kedua algoritma ini adalah karena kedua algoritma ini cocok untuk melakukan cek kemiripan judul baik itu skripsi maupun judul-judul jurnal lainnya.

DAFTAR PUSTAKA

[1] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed. Tools Appl.*, 2023, doi: 10.1007/s11042-022-13428-4.

[2] B. Bhasuran, G. Murugesan, S. Abdulkadhar, and J. Natarajan, "Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases," *J. Biomed. Inform.*, 2016, doi: 10.1016/j.jbi.2016.09.009.

[3] B. T. Atuahene, S. Kanjanabootra, and T. Gajendran, "Mapping the Barriers of Big Data Process in Construction: The Perspective of Construction Professionals," *Buildings*, 2023, doi: 10.3390/buildings13081963.

[4] Y. Gu, Y. Wang, J. Huan, Y. Sun, and S. Xu, "An improved TFIDF algorithm based on dual parallel adaptive computing model," *Int. J. Embed. Syst.*, 2020, doi: 10.1504/IJES.2020.108278.

[5] N. K. Widyasanti, I. K. G. Darma Putra, and N. K. Dwi Rusjayanthi, "Seleksi Fitur Bobot Kata dengan Metode TFIDF untuk Ringkasan Bahasa Indonesia," *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, 2018, doi: 10.24843/jim.2018.v06.i02.p06.

[6] T. Efriyanto and M. Hayaty, "JARO WINKLER ALGORITHM FOR MEASURING SIMILARITY ONLINE NEWS," *J. Tek. Inform.*, 2022.

[7] B. Berger, M. S. Waterman, and Y. W. Yu, "Levenshtein Distance, Sequence Comparison and Biological Database Search," *IEEE Trans. Inf. Theory*, 2021, doi: 10.1109/TIT.2020.2996543.

[8] L. Mo, "Fuzzy matching algorithm of network information retrieval based on discrete mathematics," *Appl. Nanosci.*, 2023, doi: 10.1007/s13204-021-02190-y.