

Chatbot Menggunakan Metode Levenshtein Distance Dalam Pencarian Mobil Bekas

Muhammad Alif Rahmat Novian Arsianto^{a1}, Faisal Fajri Rahani^{a2}

^a Informatika, Universitas Ahmad Dahlan
Jl. Ringroad Selatan, Kragilan, Tamanan, Kec. Banguntapan, Kabupaten Bantul, Daerah
Istimewa Yogyakarta

¹muhammad1700018086@webmail.uad.ac.id

²faisal.fajri@tif.uad.ac.id (Corresponding author)

Abstract

Car as private transportation has an important role as a means to carry out daily activities including going to work, shopping, and other activities. The potential buyer of its vehicle often faces some difficulties related to the price and condition of the car. They were confused about finding information for the required car. Nowadays, technology is used to solve these problems. One of the methods is by using a chatbot application that applies artificial intelligence. Chatbot is a software application that replaces a live human agent to conduct a conversation via text or text to speech. Population in this research are 235 data of used car in 2020. The variables that used are brand, built year, petroleum, transmission, milestone, engine capacity, and price. The natural language programming using levenshtein distance algorithm that can be measured of difference between two strings. The stages including data preparation, cleaning, selection, transformation, grouping questions, implementation of levenshtein distance algorithm, and system testing. The aim to determine whether Chatbot using Levenshtein distance algorithm is good enough to help potential buyer of used car. The accuracy value of this method reaches 69% with a total of 40 grouping questions from 2050 data.

Keywords: Chatbot, Confusion matrix, Levenshtein Distance, Natural Language programming, Used Car.

Abstrak

Kendaraan pribadi merupakan properti yang penting sebagai sarana untuk melakukan kegiatan sehari-hari seperti bekerja, mudik, dan berbelanja. Calon pembeli kendaraan umumnya mengalami kesulitan terkait harga dan kondisi kendaraan. Calon pembeli terkadang bingung mulai mencari informasi yang mana terlebih dulu. Teknologi jaman sekarang dapat digunakan untuk membantu menyelesaikan masalah tersebut diantaranya dengan membangun aplikasi chatbot yang menerapkan kecerdasan buatan. Penelitian ini menggunakan 235 data mobil bekas pada tahun 2020 menggunakan beberapa variabel, yaitu nama kendaraan, tahun, bahan bakar, transmisi, jarak tempuh, kapasitas mesin, dan harga. Implementasi Natural language programming dengan algoritma levenshtein distance dilakukan secara bertahap dari pembersihan data, seleksi data, transformasi data, pengelompokan pertanyaan, penerapan serta hasil algoritma levenshtein distance, dan pengujian sistem. Hasil dari sistem pencarian mobil bekas ini diharapkan dapat dimanfaatkan oleh calon pembeli kendaraan. Nilai akurasi metode ini mencapai 69% dengan jumlah 40 kelas pertanyaan dari 2050 data.

Kata Kunci : Chatbot, Confusion Matrix, Levenshtein Distance, Mobil Bekas, Natural Language Programming.

1. Pendahuluan

Pertumbuhan teknologi komunikasi dan informasi kini telah memasuki segala aspek dalam kehidupan bahkan lebih kompleks, yaitu dalam hal kecerdasan buatan [1]. Kecerdasan buatan akan melakukan pekerjaan sesuai dengan program yang mendekati pekerjaan manusia seperti yang diterapkan pada robot, program bahasa alami, sistem pakar, dan sebagainya [2].

Pemrosesan bahasa alami akan melakukan komunikasi secara interaktif dan mensimulasi obrolan antara user dengan sistem melalui bentuk teks [3]. Interaksi tersebut terjadi ketika wujud teks yang dikeluarkan lewat pemindahan kata kunci pada masukkan yang diberikan user kepada sistem sehingga menciptakan reaksi keluaran yang sesuai.

Salah satu aplikasi dari pemrosesan Bahasa alami ini dapat diaplikasikan pada kasus jual beli mobil bekas. Mobil bekas umum dijual menggunakan pihak ketiga atau yang biasa disebut makelar dan untuk makelarnya sendiri tidak hanya 1 orang per produk tapi bisa 8 bahkan lebih karena sistem tersebut menggunakan sistem jual beli mulut ke mulut, sehingga persebaran promosi produknya terbilang lambat serta sering juga terjadi masalah buyer. Masalah – masalah tersebut terjadi saat pembeli tidak mengetahui kondisi harga pasar, keadaan mobil, ataupun mobil apa yang akan dibelinya secara rinci.

Chatbot dirancang untuk melakukan percakapan antara user dan sistem yang sudah terdapat sebuah big data atau database berupa keputusan dalam memberikan jawaban atas pertanyaan yang telah di masukkan oleh user [4]. Beberapa metode digunakan untuk aplikasi chatbot ini dalam mencocokkan pertanyaan dengan data base data yang ada. Misalkan String Matching, dengan beberapa metode khusus seperti Metode Boyer Moore, Metode Levenshtein Distance, dan Brute Force.

Metode String Matching merupakan susunan dari karakter-karakter seperti angka, alfabet atau karakter yang lain serta biasanya direpresentasikan sebagai struktur data array [5]. Kelebihan yang diberikan Fuzzy String Matching yaitu menetapkan bahwa sebuah rangkaian kata yang dicari memiliki kecocokan dengan rangkaian kata tertampung dikamus, meskipun tidak mirip dalam susunan karakternya [6].

Metode Boyer Moore yaitu metode yang menjalankan kesamaan rangkaian kata mulai dari sebelah kanan ke kiri [7]. Sesuai karakteristik Boyer Moore ketidaksetaraan rangkaian kata saat dibandingkan akan menghasilkan pergerakan rangkaian kata yang melompat lebih jauh untuk menghindari perbandingan karakter pada rangkaian kata yang diperkirakan tidak berhasil .

Sedangkan Algoritma Brute Force merupakan algoritma yang mencocokkan komponen pada pattern dengan semua komponen pada text [8]. Pattern akan mulai mencocokkan dari komponen paling kiri pada pattern dengan komponen paling kiri pada teks. Bila menemukan ketidakcocokan, pattern akan terus bergeser kearah kanan hingga menemukan kecocokan dengan komponen pattern paling kiri. Ketika komponen pattern paling kiri cocok dengan komponen teks, pencocokan dilanjutkan dengan komponen sebelah kanan pattern yang telah cocok [9].

Metode Levenshtein Distance merupakan metode untuk mendeteksi setiap karakter yang dilakukan oleh chatbot [10]. Dalam memperoleh hasil keluaran yang sesuai serta akurat dengan menggunakan logika Levenshtein Distance yaitu mencari tingkatan kecocokan dengan menerapkan pencarian rangkaian kata yang sama serta juga rangkaian kata yang mendekati dengan rangkaian kata lain yang terkumpul walaupun tidak sesuai seperti lapisan awal karakternya [11].

Berdasarkan poin permasalahan tersebut, sehingga pada penelitian ini menggunakan mesin pencari dalam mencari mobil bekas dengan membangun sebuah aplikasi chatbot yang memakai metode Levenshtein Distance. Keluaran dari hasil penelitian ini berupa informasi tentang mobil bekas yang dicari oleh pengguna saat memasukkan pesan ke dalam chatbot tersebut.

2. Metode

Algoritma Levenshtein Distance merupakan algoritma yang dipakai dalam pengukuran tingkat kesamaan antara dua kata [12]. Pada tahun 1965 ilmuan asal Rusia Vladimir Losifovich Levenshtein merupakan penemu algoritma ini. Contoh hasil penggunaan algoritma ini, kata target “algoritma” dan kata sumber “algorithm” mempunyai perbedaan 2 karakter huruf perbedaan tersebut terletak pada index 7 dan 8. Di kasus tersebut 2 kata di atas, kata s=sumber “algorithm” bisa jadi t=target “algoritma” kata serta mempunyai jarak sebanyak 2 ialah karakter huruf “h” dirubah karakter huruf “m” serta huruf “m” dirubah “a”. Levenshtein dipakai secara global diberbagai bidang, contohnya pengecekan ejaan (spell checking), mesin pencarian, pengucapan dialek, pengenalan pembicaraan (speech recognition), pendeteksi pemalsuan, analisis DNA, serta

masih banyak lagi. *Levensthein* dalam mentransformasikan suatu kata menjadi kata yang lain memerlukan penghitungan operasi kata paling sedikit [13].

Dalam penulisan rumus dan persamaan menggunakan *equation* disertai nomor seperti pada persamaan (1).

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

Keterangan:

a = kata sumber, b = kata target, i = merupakan posisi karakter dari kata sumber, j = merupakan posisi karakter dari kata target. Kondisi $a_i \neq b_j$ mengacu pada karakter kata yang ada pada posisi i serta b_j juga mengacu pada karakter kata posisi j, maksudnya yaitu Ketika nilai $a_i \neq b_j$ terpenuhi maka tambahkan 1 tapi jika tidak maka tidak ditambah 1

Dalam penulisan rumus dan tabel agar tidak *dicapture* dari lembaran lain. Untuk penulisan tabel dapat mengikuti contoh tabel 1. Apabila gambar atau tabel yang memiliki ukuran kolom yang besar, dapat menggunakan satu kolom yang ditempatkan di awal atau di akhir halaman seperti pada Tabel 1.

Tabel 1 Innova matriks

		<u>I</u>	<u>N</u>	<u>O</u>	<u>V</u>	<u>A</u>
	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>
<u>I</u>	<u>1</u>					
<u>N</u>	<u>2</u>					
<u>N</u>	<u>3</u>					
<u>O</u>	<u>4</u>					
<u>V</u>	<u>5</u>					
<u>A</u>	<u>6</u>					

Pada Tabel 1 diketahui bahwa kata sumber “inova” memiliki 5 karakter dan kata target “innova” mempunyai 6 karakter. Lalu karakter ke-1 pada masing-masing kata dibandingkan dan diketahui bahwa isi karakter ke-1 pada masing-masing kata sama, maka nilai matriks yang diberikan sesuai dengan persamaan (2), (3), dan (4):

$$Lev_{a,b}(1,1) = \begin{cases} \max(i,j) \\ \min, \begin{cases} Lev_{a,b}(1-1,1) + 1 \\ Lev_{a,b}(1,1-1) + 1 \\ Lev_{a,b}(1-1,1-1) + 1_{(a_i \neq b_j)} \end{cases} \end{cases} \quad (2)$$

$$Lev_{a_i,b_j}(1,1) = \begin{cases} (0,1) + 1 \\ (1,0) + 1 \\ (0,0) + 1_{(a_i \neq b_j)} \end{cases} \quad (3)$$

$$Lev_{a_i,b_j}(1,1) = \begin{cases} 2 \\ 2 \\ 0 \end{cases} = 0 \quad (4)$$

Sehingga nilai matriks $D(1,1) = 0$ seperti diisi pada Tabel 2.

Tabel 2 Innova matriks D(1,1)

		<u>I</u>	<u>N</u>	<u>O</u>	<u>V</u>	<u>A</u>
	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>
<u>I</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
<u>N</u>	<u>2</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>
<u>N</u>	<u>3</u>					
<u>O</u>	<u>4</u>					
<u>V</u>	<u>5</u>					
<u>A</u>	<u>6</u>					

Selanjutnya perhitungan jarak dilakukan pada karakter ke-1 kata sumber dengan karakter ke-2 pada kata target serta diketahui bahwa dibutuhkan operasi penyisipan karakter “e” pada kata sumber, maka nilai yang diberikan sesuai dengan rumus operasi penyisipan 2, yaitu $D(1, 2) = D(1,2-1) + 1$ jadi nilai yang diberikan pada $D(1,2) = D(1,1)+1$ yang bernilai $D(1,2) = 0+1 = 1$. Sehingga hasil akhir dari keyword “innova” = 1 yaitu ada penggantian suku huruf.

Objek yang dipakai pada penelitian ini adalah data mobil bekas hasil scrapping data marketplace online yaitu website olx. Data-data yang diharapkan pada penelitian ini yaitu data scrapping dengan variabel yang digunakan yaitu nama kendaraan, tahun, bahan bakar, transmisi, jarak tempuh, kapasitas mesin, harga.

Data yang dipakai saat penelitian ini yaitu data mobil bekas tahun 2020. Data didapat dengan metode scrapping di website olx sebagai sumber datanya. Jumlah data yang diperoleh berjumlah 115 data dengan masing-masing memiliki variabel yaitu nama kendaraan, tahun, bahan bakar, transmisi, jarak tempuh, kapasitas mesin, serta harga. Contoh data yang digunakan pada penelitian ini ditunjukkan pada Tabel 3.

Tabel 3 Contoh data yang digunakan

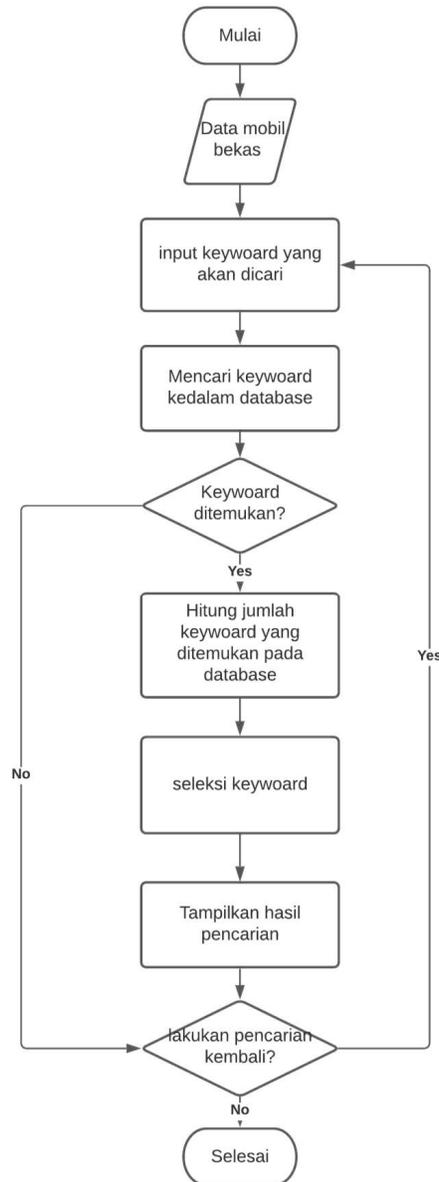
no	nama_produk	Th	merek	model	Bahan bakar	transmisi	jarak	Kapasitas mesin	harga
1	Toyota Innova	2015	Toyota	Innova	Diesel	Manual	80.000-85.000 km	>2.000 - 3.000 cc	Rp 257.000.000
2	HONDA ACCORD VTI M/T	2000	Honda	Accord	Bensin	Manual	100.000-105.000 km	>2.000 - 3.000 cc	Rp 53.000.000
3	L.300	2006	Mitsubishi	L300	Diesel	Manual	45.000-50.000 km	>1.500 - 2.000 cc	Rp 66.000.000
4	SUZUKI CARRY FUTURA 1.5 DRV	2004	Suzuki	Carry	Bensin	Manual	105.000-110.000 km	>1.000 - 1.500 cc	Rp 48.000.000
5	Toyota InnovaReborn	2017	Toyota	Kijang Innova	Diesel	Automatic Triptonic	50.000-55.000 km	>2.000 - 3.000 cc	Rp 325.000.000

Daftar pertanyaan yang digunakan untuk menjadi acuan dalam penelitian ini ditunjukkan pada Tabel 4. Dari daftar pertanyaan yang dibuat ini, akan dibandingkan dengan masukkan dari pertanyaan chatbot dengan metode yang digunakan. Saat pertanyaan sesuai dengan kemiripan tertentu yang sudah ditentukan maka sistem akan mengeluarkan data sesuai dengan database yang tersedia.

Tabel 4 Data QnA mobil bekas

No	question	Answers
1	berapa mobil innova keluaran tahun 2005?	Mobil Innova tahun 2005 ada 1 yaitu Toyota Innova E harga Rp 79.800.000
2	berapa mobil innova keluaran tahun 2011?	Mobil Innova tahun 2011 ada 1 yaitu Toyota Innova G Rp 142.500.000
3	berapa mobil innova keluaran tahun 2012?	Mobil Innova tahun 2012 ada 1 yaitu Toyota Innova G harga Rp 175.000.000
4	berapa mobil innova keluaran tahun 2013?	Mobil Innova tahun 2013 ada 1 yaitu Toyota Innova harga Rp 148.000.000
5	berapa mobil innova keluaran tahun 2015?	Mobil Innova tahun 2015 ada 1 yaitu Toyota Innova harga Rp 257.000.000
6	berapa mobil innova keluaran tahun 2016?	Mobil Innova tahun 2016 ada 2 yaitu Innova Reborn G harga Rp 214.000.000 dan Innova Reborn 2.0 harga Rp 218.000.000
7	berapa mobil innova keluaran tahun 2017?	Mobil Innova tahun 2017 ada 1 yaitu Toyota Innova Reborn dengan harga Rp 325.000.000
8	daftar mobil dengan jarak tempuh 5.000km?	berikut mobil dengan jarak tempuh 5.000 km, Mustang, Avanza G, Toyota Fortuner Vrz
9

Berdasarkan Gambar 1 yang berisi flowchart dapat dijabarkan alur dari metode Chatbot, dimulai dari sistem membaca data latih dan data uji mobil bekas. Kemudian user menginputkan keyword sesuai yang dicari lalu sistem melakukan pencarian ke database agar mendapatkan hasil pencarian. Setelah itu sistem mengecek apakah terdapat di dalam database atau tidak, jika data tersebut ada atau ada kesamaan huruf awal huruf akhir di antara kelompok data maka akan melakukan pencocokan rangkaian kata yang sesuai atau mendekati dari rangkaian kata yang diinputkan, yaitu dengan menghitung panjang rangkaian kata untuk disesuaikan dengan kelas pada masing-masing atribut jika belum ditemukan kesamaan maka ditampilkan. Jika belum ditemukan rangkaian kata tersebut maka dilanjutkan ke tahap berikutnya ialah dengan cek karakter huruf awal dan akhir tanpa menggunakan panjang len, jika ditemukan kesamaan maka ditampilkan. Bila belum berhasil atau tidak mempunyai kesamaan rangkaian kata yang terdapat dalam kamus data maka rangkaian kata tersebut dimunculkan sesuai dengan default response di dalam program.

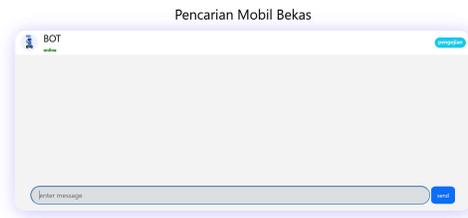


Gambar 1 *Flowchart system*

3. Hasil dan Pembahasan

3.1. Tampilan *Room Chatbot*

Tampilan ini bisa langsung digunakan untuk melakukan komunikasi secara langsung dengan sistem tanpa perlu login, yang membantu pengguna dalam melakukan pencarian mobil bekas. Seperti Gambar 2 halaman room chat. Tampilan hasil dari pencarian program terdapat pada Gambar 3. Pada gambar tersebut merupakan hasil masukkan user yang kemudian dijawab oleh sistem.



Gambar 2 Halaman chatbot
Pencarian Mobil Bekas



Gambar 3 Halaman pencarian data

3.2. Pengujian Confusion Matrix

Pengujian *Confusion Matrix* dilakukan untuk mengetahui seberapa baik keakuratan metode *Levenshtein Distance* dalam memprediksi. Diketahui data awal yang dilakukan *import* kedalam sistem berjumlah 115 data kemudian setelah itu data diolah menjadi sebuah QnA dengan cara manual sehingga mendapatkan sebuah data dengan jumlah datanya yaitu 1770 data. Dari 1770 data tersebut kemudian dibagi 2 menjadi data *training* sebanyak 80% atau sejumlah 1416 data dan data *testing* sebanyak 20% atau sejumlah 354 data. Berdasarkan 354 data testing yang telah dipilih secara acak kemudian didapatkan hasil 87 data diterima (*true*) dan hasil prediksinya juga diterima (*positive*), terdapat 10 data ditolak (*false*) tapi hasil prediksinya diterima (*positive*), terdapat 6 data diterima (*true*) tapi hasil prediksinya ditolak (*negative*), dan terdapat 61 data ditolak (*false*) dan hasil prediksinya juga ditolak (*negative*).

Berdasarkan perhitungan *confussion matrix* dengan menggunakan data *training* sejumlah 1025 data dan data *testing* sejumlah 1025 data maka didapatkan nilai *accuracy* sebesar 69%, nilai *preccission* sebesar 71% dan nilai *recall* sebesar 61%. Dari hasil tersebut dapat disimpulkan bahwa hasil pengujian pada sistem ini dapat dikatakan baik.

4. Kesimpulan

Kesimpulan yang didapatkan dari hasil penelitian mengembangkan chatbot dengan menggunakan metode *levenshtein distance* dan juga berdasarkan uraian dari bab-bab sebelumnya maka diperoleh beberapa kesimpulan sebagai berikut:

1. Telah dihasilkan sebuah sistem aplikasi yang mampu melakukan pencarian kata yang terdapat di dalam data scrapping olx.
2. Sistem yang telah dihasilkan mampu melakukan proses perhitungan *Levenshtein Distance*.
3. Berdasarkan pengujian sistem didapatkan hasil *accuracy* dengan nilai mencapai 69%, hasil *preccission* dengan nilai mencapai 71%, dan hasil *recall* dengan nilai mencapai 61%.

References

- [1] V. Lampos, J. Mintz, and X. Qu, "An artificial intelligence approach for selecting effective teacher communication strategies in autism education," *npj Sci. Learn.*, vol. 6, no. 1, Art. no. 1, Sep. 2021, doi: 10.1038/s41539-021-00102-x.
- [2] "Artificial Intelligence and the Public Sector—Applications and Challenges: International Journal of Public Administration: Vol 42, No 7." <https://www.tandfonline.com/doi/abs/10.1080/01900692.2018.1498103> (accessed Jul. 13, 2022).
- [3] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Sci. China Technol. Sci.*, vol. 63, no. 10, pp. 1872–1897, Oct. 2020, doi: 10.1007/s11431-020-1647-3.
- [4] G. Daniel, J. Cabot, L. Deruelle, and M. Derras, "Xatkit: A Multimodal Low-Code Chatbot Development Framework," *IEEE Access*, vol. 8, pp. 15332–15346, 2020, doi: 10.1109/ACCESS.2020.2966919.
- [5] Y. Yanfi, F. Gaol, B. Soewito, and H. L. H. Spits Warnars, "Spell Checker for the Indonesian Language: Extensive Review," *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, pp. 1–7, May 2022, doi: 10.46338/ijetae0522_01.
- [6] D. Sita, "PERBANDINGAN KLASIFIKASI INDEKS PEMBANGUNAN MANUSIA (IPM) DENGAN METODE K-NEAREST NEIGHBOR (K-NN) DAN SUPPORT VECTOR MACHINE (SVM) KABUPATEN/KOTA DI PULAU JAWA TAHUN 2019," *Jurnal Ilmu Komputer*, vol. 15, no. 1, Art. no. 1, Apr. 2022.
- [7] S. B. Bhonde, R. N. Baheti, A. M. Abhale, S. D. Andhale, and A. B. Arote, "CONTENT ANALYSIS AND DUPLICATION AVOIDANCE SYSTEM USING MACHINE LEARNING: A REVIEW," vol. 6, no. 4, p. 4, 2021.
- [8] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text Data Augmentation for Deep Learning," *J Big Data*, vol. 8, no. 1, p. 101, Jul. 2021, doi: 10.1186/s40537-021-00492-0.
- [9] U. Hasson, S. A. Nastase, and A. Goldstein, "Direct Fit to Nature: An Evolutionary Perspective on Biological and Artificial Neural Networks," *Neuron*, vol. 105, no. 3, pp. 416–434, Feb. 2020, doi: 10.1016/j.neuron.2019.12.002.
- [10] N. Anggraini and M. J. Tursina, "Sentiment Analysis of School Zoning System On Youtube Social Media Using The K-Nearest Neighbor With Levenshtein Distance Algorithm," in *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, Nov. 2019, vol. 7, pp. 1–4. doi: 10.1109/CITSM47753.2019.8965407.
- [11] B. Berger, M. S. Waterman, and Y. W. Yu, "Levenshtein Distance, Sequence Comparison and Biological Database Search," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3287–3294, Jun. 2021, doi: 10.1109/TIT.2020.2996543.
- [12] F. Firdausillah and A. Arieansyah, "Implementasi Algoritma Levenshtein Distance Sebagai Chatbot Agen Pariwisata Berbasis Aplikasi LINE," *SEMINAR NASIONAL APTIKOM (SEMNASITIK) 2019*, no. 0, Art. no. 0, Nov. 2019.
- [13] A. Putera Utama Siahaan *et al.*, "Combination of levenshtein distance and rabin-karp to improve the accuracy of document equivalence level," *IJET*, vol. 7, no. 2.27, p. 17, Jun. 2018, doi: 10.14419/ijet.v7i2.27.12084.