

Terbit online pada laman web jurnal: <http://jurnal.iaii.or.id>

JURNAL RESTI

(Rekayasa Sistem dan Teknologi Informasi)

Vol. 4 No. 1 (2020) 64 - 69

ISSN Media Elektronik: 2580-0760

Autocorrect pada Modul Pencarian *Drugs e-Dictionary* Menggunakan Algoritma *Levenshtein Distance*

Halimah Tus Sadiyah¹, Muhamad Saad Nurul Ishlah², Nisa Najwa Rokhmah³^{1,2}Manajemen Informatika, Program Diploma, Universitas Pakuan³Farmasi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Pakuan¹sadiaht@unpak.ac.id, ²nurul.islah@unpak.ac.id, nisanajwarokhmah@gmail.com

Abstract

The Dictionary of Medicine in the form of a physical book has many drawbacks, one of them is its thickness makes it impractical to be carried. This becomes a motivation to develop drug dictionary applications in the form of a *Drugs e-Dictionary*. One of the developed *Drugs e-Dictionary* uses A-Z index-based approach to discover any drug terms. This approach is less effective and less efficient timewise. Therefore, it is necessary to add a search function that has an autocorrect feature to aid the user. The purpose of this study is to build a search module that has an autocorrect feature on *Drugs e-Dictionary* using the *Levenshtein Distance* algorithm. The methodology or the stages of this research divided into the construction of a search module on *Drugs e-Dictionary*, implementation of the *Levenshtein Distance* algorithm, and autocorrect validation test. The results of the algorithm implementation show that the search module with the autocorrect feature can detect typing errors in the inputted terms by producing the closest drug term output in the database, then automatically provide suggestions for improvement and display the results of the improved drug terms to the user. it reaches 90% accuracy of inputted query, with 90% precision and 90% recall.

Keywords: autocorrect, drugs e-dictionary, algorithm, the *Levenshtein Distance* algorithm.

Abstrak

Kamus Obat berbentuk fisik seperti buku memiliki banyak kekurangan diantaranya adalah tebal sehingga tidak praktis. Hal ini menjadi motivasi bagi para developer aplikasi untuk mengembangkan aplikasi kamus obat salah satunya adalah *drugs e-dictionary*. *Drugs e-dictionary* dikembangkan tahun 2019. Pada aplikasi tersebut pencarian masih berupa index huruf A-Z sehingga user mencari istilah secara urutan sekuensial dengan mengklik satu persatu index abjad. Hal ini membutuhkan waktu yang lama sehingga menjadi tidak efisien dan efektif. Oleh karena itu perlu adanya penambahan fungsi pencarian yang memiliki fitur autocorrect jika user salah dalam mengetik istilah obat. Adapun tujuan ini penelitian ini adalah membangun modul pencarian yang memiliki fitur autocorrect pada *drugs e-dictionary* menggunakan algoritma *Levenshtein Distance*. Metodologi atau Tahapan penelitian, yaitu pembangunan modul pencarian pada *drugs e-dictionary*, implementasi algoritma *Levenshtein Distance*, validasi autocorrect. Berdasarkan hasil implementasi algoritma *Levenshtein Distance* dan validasi autocorrect, modul pencarian pada *drugs e-dictionary* dengan fitur autocorrect dapat mendeteksi kesalahan pengetikan istilah kemudian secara otomatis memberikan saran perbaikan dan menampilkan hasil perbaikan istilah. Adapun fungsi autocorrect bekerja dengan menghasilkan output istilah obat yang terdekat pada database berdasarkan input kesalahan ketik istilah obat.

Kata kunci: autocorrect, drugs e-dictionary, algoritma, algoritma *Levenshtein Distance*.

© 2020 Jurnal RESTI

1. Pendahuluan

Drugs atau yang dikenal dengan istilah Obat merupakan zat yang digunakan untuk pencegahan dan penyembuhan penyakit serta pemulihan dan peningkatan kesehatan bagi penggunaannya [1]. Menurut Nuryati (2017), Obat merupakan suatu zat atau bahan-

bahan yang berguna dalam menetapkan diagnosa, mencegah, mengurangi, menghilangkan, menyembuhkan penyakit atau gejala penyakit, luka atau kelainan fisik dan rohani pada manusia atau hewan, termasuk mempercantik tubuh atau bagian tubuh manusia [2]. Adapun *Drugs e-dictionary* merupakan aplikasi kamus elektronik berbasis web yang mana

Diterima Redaksi : 08-11-2020 | Selesai Revisi : 16-12-2019 | Diterbitkan Online : 02-02-2020

pengetahuan obat disimpan dalam bentuk halaman web yang dapat diakses kapan saja dan dimana saja dengan syarat terkoneksi internet. *Drugs e-dictionary* dikembangkan tahun 2019. Pada aplikasi tersebut pencarian masih berupa index huruf A-Z sehingga *user* mencari istilah secara urutan sekuensial dengan mengklik satu persatu index abjad. Hal ini membutuhkan waktu yang lama sehingga menjadi tidak efisien dan efektif. Oleh karena itu perlu adanya penambahan fungsi pencarian yang memiliki fitur *autocorrect* jika *user* salah dalam mengetik istilah obat.

Fitur *autocorrect* pada suatu aplikasi berfungsi sebagai otomatisasi pengoreksian kata yang salah dengan langsung menampilkan hasil yang benar. Fungsi *autocorrect* diimplementasikan menggunakan algoritma Levenshtein Distance guna meningkatkan *usability* [3].

Algoritma Levenshtein Distance merupakan algoritma yang dibuat pada tahun 1965 oleh Vladimir Levenshtein [4]. Algoritma ini mencari jarak antara kata yang diinputkan user dengan kata pada database dengan menghitung jumlah perbedaan antara kedua string dalam bentuk matriks [5][6][7]. Cara kerja algoritma Levenshtein Distance yaitu menghitung jarak antara kedua string kemudian jumlah minimum dicari dari operasi perubahan untuk merubah dari string A menjadi string B. Adapun perhitungan direpresentasikan dalam bentuk tabel perhitungan Levenshtein Distance, di mana nilai terakhir pada pojok kanan bawah merupakan nilai akhir dari jarak kedua string [8]. Pada Algoritma Levenshtein Distance terdapat tiga macam operasi yaitu Operasi pengubahan karakter, penambahan karakter dan penghapusan karakter [9][10][11].

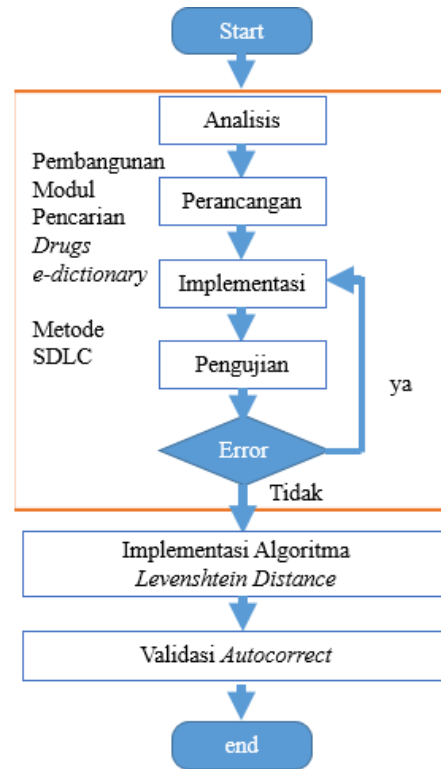
Berdasarkan permasalahan pencarian, maka Penelitian ini bertujuan untuk membangun fungsi pencarian dengan fitur *autocorrect* menggunakan algoritma Levenshtein Distance. Adapun terdapat penelitian terkait, yaitu Implementasi Fitur Autocomplete dan Algoritma Levenshtein Distance untuk Meningkatkan Efektivitas Pencarian Kata di Kamus Besar Bahasa Indonesia (KBBI) oleh Ngafidin dan Wibawanto (2015) [12].

2. Metode Penelitian

Metode penelitian terdiri atas beberapa tahapan yaitu (Gambar 1), yaitu

1. Pembangunan modul pencarian pada *Drugs e-Dictionary*, Pembangunan modul pencarian menggunakan metode SDLC (*System Development Life Cycle*), yaitu analisis kebutuhan modul pencarian, perancangan modul pencarian, implementasi dan pengujian modul pencarian.
2. Implementasi algoritma Levenshtein Distance, dilakukan setelah modul pencarian berfungsi tanpa *error*.

3. Validasi *Autocorrect*, menggunakan metode *blackbox* [13][14]. Pengujian dilakukan berdasarkan kebutuhan fungsionalitas sistem.



Gambar 1. Tahapan Penelitian

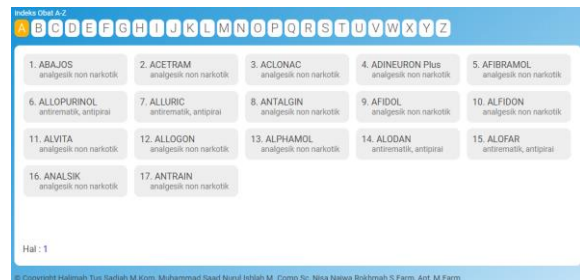
3. Hasil dan Pembahasan

3.1. Pembangunan Modul Pencarian *Drugs e-dictionary*

Analisis

Pada tahap analisis dihasilkan hasil analisis yang sedang berjalan, yaitu

1. *Drugs e-dictionary* pencariannya berdasarkan index A-Z (Gambar 2).

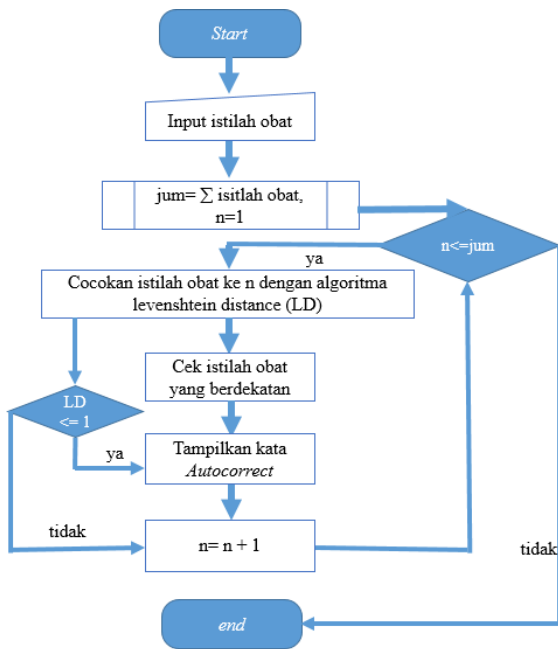


Gambar 2. Pencarian *Drugs e-dictionary* berdasarkan index A-Z

2. Belum adanya modul pencarian istilah obat
3. Data berupa istilah obat dan deskripsinya yang merujuk pada sumber buku (IAI 2019) [15].
4. Pencarian belum dilengkapi fitur *autocorrect*

Perancangan

Pada tahap perancangan, dilakukan perancangan modul pencarian berupa *flowchart* modul pencarian (Gambar 3). Pada Gambar 3, User *input* istilah obat pada *form* pencarian, lalu proses inialisasi jumlah kata = j dan inialisasi nilai $n=1$. Kemudian sistem memproses jika $n \leq j$ maka sistem akan mencocokkan istilah ke n dengan algoritma *levshtein distance*. Sistem akan mengecek istilah obat yang berdekatan sekaligus mengecek nilai algoritma. Terdapat dua kondisi, jika nilai $LD \leq 1$, maka sistem akan menampilkan kata autocorrect lalu $n++$ ($n=n+1$). Namun jika nilai $LD > 1$, maka sistem akan increment nilai n dan looping sampai nilai $n > j$ jumlah istilah obat.



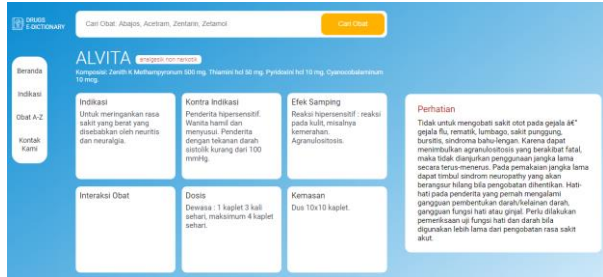
Gambar 3 Flowchart modul pencarian

Implementasi

Implementasi modul pencarian dilakukan menggunakan bahasa pemrograman PHP-MYSQLi. Implementasi dilakukan pada lingkungan perangkat lunak intel core i3. Hasil implementasi modul pencarian ditunjukkan Gambar 4. Adapun hasil dari modul pencarian akan menginformasikan mengenai informasi obat yang dicari terdiri atas indikasi, kontradiksi, efek samping, interaksi obat, dosis, kemasan dan perhatian mengenai obat (Gambar 5).



Gambar 4 Hasil implementasi modul pencarian



Gambar 5 Hasil implementasi modul pencarian

Pengujian

Hasil Pengujian pada modul pencarian ditunjukkan Tabel 1. Pengujian dilakukan dengan cara input istilah obat dengan 3 kondisi, yaitu inputan kosong, inputan salah, inputan benar.

Tabel 1. Tabel Pengujian Modul Pencarian

No	Input	Output	Keterangan
1	Istilah Obat dikosongkan	Notifikasi field masih kosong, silahkan input data istilah obat yang dicari	Berfungsi
2	Input istilah Obat Salah	Istilah obat yang dicari tidak ditemukan	Berfungsi
3	Input Istilah Obat yang benar	Tampil informasi obat yang dicari terdiri atas indikasi, kontradiksi, efek samping, interaksi obat, dosis, kemasan dan perhatian	Berfungsi

3.2 Implementasi Algoritma *Levenshtein Distance*

Langkah pertama dalam implementasi algoritma yaitu, konversi *pseudocode* (Gambar 6) ke dalam bahasa pemrograman PHP kemudian pemanggilan algoritma dalam modul pencarian.

Program Pseudocode Algoritma Levenshtein Distance

```

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

Pada implementasi algoritma Levenshtein Distance ada 3 macam operasi utama, yaitu :

1. Operasi pengubahan karakter
 Operasi pengubahan karakter pada *pseudocode* ditunjukkan pada nilai output $d[i-1, j-1] + cost$ pada

Gambar 5. Nilai ini didapatkan dari diagonal atas kiri pada komputasi matriks. Perhitungan komputasi matriks ditunjukkan Gambar 6. Adapun hasil implementasi dapat dilihat pada Gambar 7.

O R A S I C						
0	1	2	3	4	5	6
O	1	0				
R	2	1				
A	3	2				
S	4	3				
I	5	4				
K	6	5				

O R A S I C						
0	1	2	3	4	5	6
O	1	0	1			
R	2	1	0	1		
A	3	2	1	0		
S	4	3	2	1		
I	5	4	3	2	1	
K	6	5	4	3	2	1

O R A S I C						
0	1	2	3	4	5	6
O	1	0	1	2	3	4
R	2	1	0	1	2	2
A	3	2	1	0	1	2
S	4	3	2	1	0	1
I	5	4	3	2	1	0
K	6	5	4	3	2	1

O R A S I C							
0	1	2	3	4	5	6	
O	1	0	1	2	3	4	5
R	2	1	0	1	2	3	4
A	3	2	1	0	1	2	3
S	4	3	2	1	0	1	2
I	5	4	3	2	1	0	1
K	6	5	4	3	2	1	1

Gambar 6. Perhitungan komputasi matriks operasi perubahan karakter



Input : Orasik
 Output Autocorrect : Orasic



Gambar 7. Implementasi operasi perubahan algoritma Levenshtein Distance pada *drugs e-dictionary*

2. Operasi penambahan karakter
 Operasi penambahan karakter pada *pseudocode* ditunjukkan pada nilai output $d[i, j-1] + 1$ pada Gambar

5. Nilai ini didapatkan dari samping kiri pada komputasi matriks. Perhitungan komputasi matriks ditunjukkan Gambar 8. Adapun hasil implementasi dapat dilihat pada Gambar 9.

I N F E L D						
0	1	2	3	4	5	6
I	1	0				
N	2	1				
F	3	2				
E	4	3				
L	5	4				

I N F E L D						
0	1	2	3	4	5	6
I	1	0	1			
N	2	1	0	1		
F	3	2	1	0		
E	4	3	2	1		
L	5	4	3	2		

I N F E L D						
0	1	2	3	4	5	6
I	1	0	1	2	3	4
N	2	1	0	1	2	3
F	3	2	1	0	1	2
E	4	3	2	1	0	1
L	5	4	3	2	1	0

I N F E L D							
0	1	2	3	4	5	6	
I	1	0	1	2	3	4	5
N	2	1	0	1	2	3	4
F	3	2	1	0	1	2	3
E	4	3	2	1	0	1	2
L	5	4	3	2	1	0	1

Gambar 8. Perhitungan komputasi matriks operasi penambahan karakter



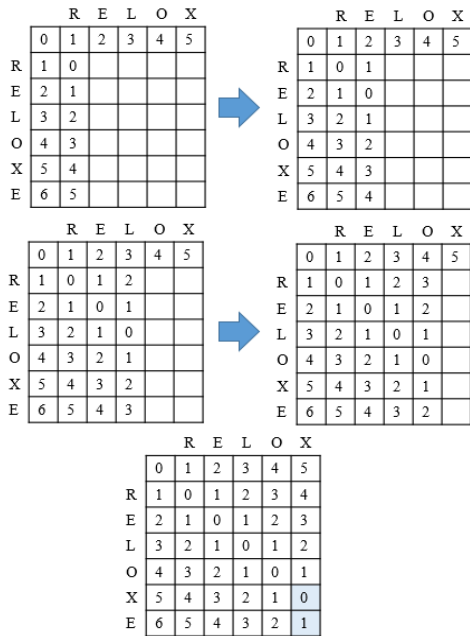
Input : Infel
 Output Autocorrect : Infeld



Gambar 9. Implementasi operasi penjumlahan algoritma Levenshtein Distance pada *drugs e-dictionary*

3. Operasi penghapusan karakter
 Operasi penambahan karakter pada *pseudocode* ditunjukkan pada nilai output $d[i-1, j] + 1$ pada Gambar

5. Nilai ini didapatkan dari atas pada komputasi matriks. Perhitungan komputasi matriks ditunjukkan Gambar 10. Adapun hasil implementasi dapat dilihat pada Gambar 11.



Gambar 10. Perhitungan komputasi matriks operasi penghapusan karakter



Gambar 11. Implementasi operasi penghapusan algoritma Levenshtein Distance pada *drugs e-dictionary*

3.3. Validasi *Autocorrect*

Modul pencarian yang telah diimplementasikan algoritma *Levenshtein Distance* divalidasi fungsi *autocorrect* berdasarkan inputan data pada modul pencarian. Berdasarkan hasil validasi, modul pencarian pada *drugs e-dictionary* dapat mengoreksi kesalahan pengetikan ejaan istilah obat. Fitur *Autocorrect* dapat menvalidasi sejumlah tiga operasi algoritma *Levenshtein Distance*. Selain divalidasi ketiga operasi, diuji pula istilah obat yang tidak terdapat dalam *database*. Berdasarkan hasil uji, modul pencarian pada *drugs e-dictionary* dapat menampilkan fitur *autocorrect* pada istilah obat yang tidak terdapat pada *database*. Fitur *autocorrect* bekerja pada *drugs e-dictionary* dengan menampilkan istilah obat yang terdekat dengan ejaan istilah obat yang ada pada *database* Adapun hasil uji disajikan Tabel 2.

Tabel 2. Tabel Pengujian Modul Pencarian

No	Input	Target	Output	Keterangan
1	Istilah Obat kosong	Notifikasi	Notifikasi field masih kosong, silahkan input data istilah obat yang dicari	Valid
2	Alogon	Allogon	Allogon	Operasi Penambahan Valid
3	Hufragin	Hufralgin	Hufralgin	Operasi Penambahan Valid
4	Raos	Raost	Raost	Operasi Penambahan Valid
5	Asetram	Acetram	Acetram	Operasi Pengubahan Valid
6	Biogesik	Biogesic	Biogesic	Operasi Pengubahan Valid
7	Ponkofen	Poncofen	Poncofen	Operasi Pengubahan Valid
8	Anallsik	Analsik	Analsik	Operasi Penghapusan Valid
9	Paracol	Paraco	Paraco	Operasi Penghapusan Valid
10	Tirafen	Tiafen	Tiafen	Operasi Penghapusan Valid

Pada Tabel 2, diuji 10 inputan berupa 1 istilah kosong, 3 istilah operasi pengubahan huruf, 3 istilah operasi penambahan huruf dan 3 istilah operasi penghapusan huruf. Pada Tabel 2 terdapat kolom target dan output. Kolom Target adalah istilah yang seharusnya dihasilkan sedangkan output adalah hasil dari luaran

modul pencarian pada *drugs e-dictionary*. Berdasarkan tabel uji (Tabel 2) didapatkan validasi istilah obat untuk implementasi tiga operasi algoritma *Levenshtein Distance* pada modul pencarian *drugs e-dictionary*.

Pada penelitian ini dilakukan pengujian algoritma *levenshtein distance* menggunakan *confusion matrix* [16][17]. Berdasarkan hasil *confusion matrix* Tabel 3 dihasilkan nilai akurasi, *recall* dan *precision* dari implementasi algoritma *levenshtein distance* yaitu sebesar 90 %. Hasil ini dihitung menggunakan persamaan (1)(2)(3).

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} * 100\% \quad 1)$$

$$precision = \frac{TP}{TP + FP} * 100\% \quad 2)$$

$$recall = \frac{TP}{TP + FN} * 100\% \quad 3)$$

Tabel 3 *Confusion Matrix*

100	Predicted: yes	Predicted: no
Actual: yes	45	5
Actual: no	5	45

4. Kesimpulan

Penambahan modul pencarian pada *drugs e-dictionary* sangat diperlukan guna memudahkan *user* fserta meningkatkan efisien dan efektifitas dalam mencari istilah obat . Berdasarkan hasil implementasi algoritma *Levenshtein Distance* dan validasi *autocorrect*, modul pencarian pada *drugs e-dictionary* dengan fitur *autocorrect* dapat mendeteksi kesalahan pengetikan istilah kemudian secara otomatis memberikan saran perbaikan dan menampilkan hasil perbaikan istilah obat. Fungsi *autocorrect* bekerja dengan menghasilkan *output* istilah obat yang terdekat pada *database* berdasarkan input kesalahan ketik istilah obat. Adapun berdasarkan hasil pengujian implementasi algoritma pada *drugs e-dictionary* dihasilkan nilai akurasi, *recall* dan *precision* sebesar 90%.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada KEMENRISTEKDIKTI, dan LLDIKTI yang telah mendanai penelitian ini. Ucapan terima kasih kepada LPPM Universitas Pakuan yang telah membantu dalam pelaksanaan penelitian.

Daftar Rujukan

- [1] GNPOPA. 2015. Materi *Edukasi Tentang Peduli Obat dan Pangan Aman*. Jakarta: GNPOPA.
- [2] Nuryati. 2017. *Farmakologi*. Jakarta: Kementerian Kesehatan republik Indonesia.
- [3] Lalwani, M., Bagmar, N., & Parikh, S. 2014. Efficient Algorithm for Auto Correction Using ngram Indexing. *International Journal of Computer & Communication Technology (IJCCT)*. 3(3), pp. 23-27.
- [4] Afriansyah, Z., Puspitaningrum, D., & Ernawati. 2015. Rancang Bangun Aplikasi Pencocokan Dna Manusia Menggunakan Algoritma Levenshtein Distance (Studi Kasus: Dna Kanker Hati Manusia). *Jurnal Rekursif*. 3(2), pp. 61-67.
- [5] Pratama, B., & Pamungkas, S. 2016. Analisis Kinerja Algoritma Levenshtein Distance Dalam Mendeteksi Kemiripan Dokumen Teks. *Jurnal Log!k@*. 6(2), pp. 131-143.
- [6] Aprilianto, T., & Badawi, A. 2017. Sistem Koreksi Kata Dan Pengenalan Struktur Kalimat Berbahasa Indonesia Dengan Pendekatan Kamus Berbasis Levenshtein Distance. *Jurnal SPIRIT*. 9 (1), pp. 48-61.
- [7] Rosmala D, Risyad ZF. 2017. Algoritma Levenshtein Distance Dalam Aplikasi Pencarian Kata Isu Di Kota Bandung Pada Twitter. *MIND Journal*. 2(2), pp.1-12.
- [8] Adriyani, N. M., Santiyasa, I. W., & Muliantara, A. (n.d.). Implementasi Algoritma Levenshtein Distance Dan Metode Empiris Untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Dokumen Berbahasa Indonesia. [Cited 2018 August 1]. Available from <https://ojs.unud.ac.id/index.php/JLK/article/view/2800>
- [9] Mishra, R., & Kaur, N. 2013. A Survey of Spelling Error Detection and Correction Techniques. *International Journal of Computer Trends and Technology*. 3 (4), pp. 372-374.
- [10] Ariyani, N., Sutardi, & Ramadhan, R. 2016. Aplikasi Pendeteksi Kemiripan Isi Teks Dokumen Menggunakan Metode Levenshtein Distance. *semantik*. 2(1), pp. 279-286.
- [11] Haldar, R., & Mukhopadhyay, D. 2011. Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach. [Cited 2018 August 1]. Available from <http://www.cornell.edu/>; <https://arxiv.org/abs/1101.1232>.
- [12] Ngafidin, K. N., & Wibawanto, H. 2015. Implementasi Fitur Autocomplete dan Algoritma Levenshtein Distance untuk Meningkatkan Efektivitas Pencarian Kata di Kamus Besar Bahasa Indonesia (KBBI). *Jurnal Teknik Elektro*. 7(1), pp.1-6.
- [13] Pressman R dan Maxim B.R. 2014. *Software Engineering a Practitioners approach*. McGraw-Hill Education : New York.
- [14] Satzinger, J., Jackson, R., & Burd, S. 2010. *System Analysis and Design in a changing World*. USA: Course Technology Cengage Learning.
- [15] IAI (Ikatan Apoteker Indonesia). 2019. *ISO Informasi Spesialite Obat Indonesia Vol 52-Tahun 2019*. Jakarta : Isfi Penerbitan.
- [16] A.Pahdi. Koreksi Ejaan Istilah Komputer Berbasis Kombinasi Algoritma Damerau-Levenshtein dan Algoritma Soundex. *Journal Speed – Sentra Penelitian Engineering dan Edukasi*.8:2, pp. 1-8. 2016.
- [17] P.A.Arsaningtyas, M.A. Bijaksana. S.A. Faraby. Sistem Pencarian Ayat Al-Quran Berdasarkan KemiripanUcapan Menggunakan Algoritma Soundex dan Damerau-Levenshtein Distance. *Jurnal Linguistik Komputasional*. 1:2, pp. 58-64. 2018