

IMPLEMENTASI ALGORITMA LEVENSHTTEIN DISTANCE DENGAN RESTFUL WEB SERVICE PADA KATA BAHASA INDONESIA KE BAHASA JAWA BERBASIS WEB

Welly Kuswanto

Prodi teknik informatika, STIKI Malang
Email: Wellykuswanto@gmail.com

(Naskah masuk: 14 Juni 2020, diterima untuk diterbitkan: 27 Juni 2020)

ABSTRAK

Bahasa Indonesia yang selalu kita pakai menjadi bahasa persatuan di Indonesia. Disamping bahasa persatuan, Indonesia juga memiliki banyak bahasa daerah salah satunya adalah bahasa Jawa. Semua warga negara Indonesia tentunya terbiasa dalam berbahasa Indonesia karena Bahasa Indonesia adalah Bahasa nasional. Namun dalam berkomunikasi bahasa daerah seperti bahasa Jawa, masyarakat tidak terbiasa menggunakan Bahasa Jawa dibandingkan bahasa Indonesia terutama masyarakat yang tinggal di pulau Jawa. Hal ini dikarenakan masyarakat Jawa terbiasa berbahasa Indonesia dibandingkan dengan berbahasa Jawa. Oleh karena itu, penulis mencoba membangun sistem untuk mempermudah pengembang aplikasi dengan tujuan agar efektif dan efisien dalam mengembangkan aplikasinya tanpa memandang berbagai platform dan bahasa pemrograman yang digunakan. Hal ini bertujuan untuk menyediakan komunikasi online yang menyenangkan disaat terjadi pandemic Corona Virues Disease 2019 (COVID 19) di Indonesia khususnya di pulau Jawa. Efek langsung yang dirasakan pada saat pandemi COVID 19 adalah semua pekerja mengimplementasikan komunikasi via dalam jaringan (daring/online) terkait dengan urusan kantor. Hal ini menyebabkan penulis menyediakan media komunikasi yang inovatif dalam bahasa Jawa, maka dibuatlah *api (web service)*, serta menggunakan metode levenshtein distance dalam pencocokan kata bahasa Indonesia sebelum diterjemahkan ke bahasa Jawa. *Web service* yang dirancang menggunakan susunan *restfull web service*. Hasil aplikasi menggunakan metode *approximate string matching* didapatkan hasil persentase nilai *precision* sebesar 95,32%, nilai *recall* 100%, nilai *accuracy* 99,97%, dan nilai *f-measure* 97,51%. berdasarkan input sumber dan output target sangat efektif karena mendapatkan persentase yang cukup tinggi. tidak terdapat banyak kesalahan pada saat proses, sehingga secara fungsional dapat menampilkan hasil yang yang diinginkan.

Kata kunci: Web Service, levenshtein distance, Bahasa Indonesia, Bahasa Jawa.

ABSTRACT

Indonesian which is always used as a union language. Besides of it, Indonesia has many regional languages, one of which is Javanese. All Indonesian citizens are naturally accustomed to speaking Indonesian because Indonesian is the national language. However, in communicating local languages such as Javanese, Indonesian society are not accustomed to using Javanese compared to Indonesian, especially people who live on the Java Island. This condition caused by Javanese people are accustomed to speak Indonesian compared to Javanese. Therefore, the writer tries to build a system to facilitate developers with the aim to be effective and efficient in developing applications regardless of the various platforms and programming languages used. It aims to provide pleasant online communication when the pandemic Corona Virus Disease 2019 (COVID 19) occurs in Indonesia, especially on the Java Island. The immediate effect felt during the COVID 19 pandemic is that all workers implement online communication related to their jobs. This condition causes the writer to provide innovative communication media in Javanese, create Application Programming Interface (web service), and uses levenshtein distance method in matching Indonesian words before translating to Javanese. Web service designed using a restful web service arrangement. The results of the application using the approximate string matching method obtained the percentage of precision values of 95.32%, 100% recall value, 99.97% accuracy value, and f-measure value 97.51%. Based on source input and target output is very effective because it gets a high percentage. There are not many errors during the process, so it can functionally display the desired results.

Keywords: Web Service, Levenshtein Distance, Indonesian, Javanese.

1. PENDAHULUAN

Jati diri sebuah bangsa bisa dilihat dari segi bahasa, seperti halnya bahasa Indonesia yang digunakan sebagai bahasa nasional (Rifa'i, 2015). Bahasa persatuan atau bahasa Indonesia mempunyai peran penting disemua aspek kehidupan bersosialisasi dalam hal berkomunikasi dengan masyarakat. Selain mempunyai bahasa persatuan (Arisandy et al., 2019), Indonesia juga mempunyai berbagai bahasa daerah, diantaranya bahasa Jawa.

Bahasa Jawa adalah suatu Bahasa daerah yang perannya dituturkan oleh penduduk bagian tengah dan timur pulau jawa dan merupakan kebudayaan dari Indonesia sejak pendaratan penduduk austronesia tepatnya di pantai utara jawa (Noerwidi, 2008), akan tetapi komunikasi dalam Bahasa jawa dirasa kurang sebanding dengan fakta yang ada. Karena masyarakat jawa sendiri lebih banyak berbahasa Indonesia dari pada berbahasa jawa. Sarana yang bisa menjembatani pada masyarakat agar komunikasi berjalan dengan baik adalah adanya Bahasa (Ardana, 2019), dan akan selalu digunakan oleh masyarakat luas hingga luar daerah pulau jawa dalam berkomunikasi. Bahasa Jawa merupakan suatu bahasa daerah yang sangat perlu dilestarikan supaya tidak hilang keberadaannya.

Pada tutur dalam Bahasa jawa terdapat banyak tingkat tutur dan terlihatnya tingkat tutur sudah diperkirakan pada abad ke-17, tepatnya pada zaman Raja Sultan Agung (1613-1645), yaitu pada zaman kerajaan Mataram (Budiono et al., 2018). Namun adanya tingkat tutur tersebut sudah ada pada abad ke-15 yaitu pada periode pertengahan, yang munculnya terdapat tulisan *Dewaruci Tembang Gedhe*. Walaupun sudah terlihat pada abad ke-15,

tapi khususnya tutur kata krama inggil sudah ada pada zaman kuno atau dalam bahasa Jawa kuno (Wilian, 2006).

Tingkat tutur ngoko adalah variasi bahasa dengan kosakata ngoko yang digunakan untuk berkomunikasi dengan yang sudah akrab seperti teman atau dengan pergaulan yang sama (Suryadi, 2013), bahasa ini mencerminkan rasa tidak berjarak atau bahasa pergaulan yang di pakai sehari-hari.

Ragam bahasa Jawa ngoko adalah leksikon Ngoko unggah ungguh. Imbuhan yang muncul dalam ragam ngoko menggunakan -e, di-, dan -ake. Ragam Bahasa ngoko dapat dibedakan menjadi dua yaitu ngoko alus dan ngoko lugu

Ngoko: "*dhèwèkè khanda yèn wong tuwanè ora bisa tindak mrênè*".

Madya: "*Piyambakè criyos yèn tiyang sepuhè mboten saged mriki*".

yang artinya: "dia berkata bawa orang tuanya tidak bisa datang kesini"

Penggunaan media online yang berbahasa Jawa meningkat pada saat Indonesia mengalami Corona Virus Disease 2019 yang disingkat dengan COVID 19 meningkat pesat. Akibat Covid pada segala sector menurun drastic seperti UMKM makanan dan minuman 27%, konsumsi rumah terkoreksi sampai 0.5 – 0.7%, sedangkan penggunaan teknologi meningkat sampai 85% (Amri, 2020). Berdasarkan fenomena COVID 19, peneliti bersemangat untuk mengembangkan koreksi bahasa Indonesia ke Bahasa Jawa agar memudahkan penggunaan Bahasa jawa via media online. Disamping itu, penelitian ini bertujuan untuk mengetahui seberapa cepat tingkat ke-akurasi dengan metode Levenshtein Distance. Pada penelitian ini dibangunlah sistem untuk mempermudah pengembang agar lebih fleksibel dalam mengembangkan

aplikasinya tanpa memandang berbagai platform dan Bahasa pemrograman yang digunakan yang digunakan.

2. METODELOGI

3.1. Data Penelitian

Data yang digunakan pada penelitian Implementasi Algoritma Levenshtein Distance Dengan Restful Web Service Pada Kata Bahasa Indonesia Ke Bahasa Jawa Berbasis Website yang terdiri dari 172 kata yang diambil dari website dongengceritarakyat.com.

2.1 Masukkan query client

Pada tahap ini akan dimasukkan *query client* untuk mencari hasil terjemah dari kalimat bahasa Indonesia ke bahasa Jawa. Pertama, memasukkan daftar kata dari Tabel 4.1.

2.2 Hasil proses

Proses terakhir dari sistem adalah, menampilkan hasil query dari input client. Sehingga dapat diketahui apakah input dari client terdeteksi atau tidak terdeteksi oleh metode Levenshtein Distance dalam pencocokan kata yang ditargetkan. Apabila kata tersebut terdeteksi oleh metode, maka akan ditampilkan sugesti hasil kata oleh sistem.

3. HASIL DAN PEMBAHASAN

Sebagai peneliti tentunya ingin mengetahui input, proses dan output dari algoritma levenshtein distance. Untuk mendeteksi kemiripan dan kebenaran dalam menterjemah Bahasa Indonesia ke Bahasa Jawa berbasis web peneliti mengembangkan source code, input data penelitian, menjalankan program, dan menganalisis hasil program menggunakan confusion matrix.

4.1 Source code yang digunakan

Berikut merupakan source code Levenshtein Distance dan beserta fungsinya yaitu:

```

1. <?php
2. class BackEndLevenshtein{
3.     function dapatkanNilaiLevenshtein($stringInput, $stringTujuan){
4.         // $jarakAntarPrefixes = [count($stringInput)][count($stringTujuan)];
5.         $jarakAntarPrefixes = array();
6.         for($i=0;$i<strlen($stringInput);$i++){
7.             for($j=0;$j<strlen($stringTujuan);$j++){
8.                 $jarakAntarPrefixes[$i][$j] = -1;
9.             }
10.        }
11.    }
12.    return $this->dapatkanJarakTerdekat($stringInput, (strlen($stringInput)-1), $stringTujuan, (strlen($stringTujuan)-1), $jarakAntarPrefixes);
13. }
14.
15. function dapatkanJarakTerdekat($stringInput, $stringInputIdx, $stringTujuan, $stringTujuanIdx, $jarakAntarPrefixes){
16.     //apakah string inputan kosong ? jika iya maka jarak = stringTujuanIdx + 1
17.     if($stringInputIdx < 0){
18.         return $stringTujuanIdx+1;
19.     }

```

Source code tersebut merupakan pengambilan nilai minimum levenshtein distance, yang akan di eksekusi untuk di ambil percobaan pencocokan kata sebelum di terjemahkan ke Bahasa jawa.

```

1. function pengolahanLevenshteinPrefixSuffix($kata){
2.     //STEP 1 olahPrefix terlebih dahulu
3.     $hasil = "";
4.     $hasil = $this->olahPrefix($kata);
5.     if($this->statusBerhasilOlahPrefix == 0){ // status = 0, berarti proses olahPref

```

```

ix gagal, lanjut ke proses selanjut
nya yaitu olahSuffix
6.   $hasil = $this-
>olahSuffix($kata);
7.   if($this-
>statusBerhasilOlahSuffix == 0){
8.     $hasil = $this-
>olahPreffixSuffix($kata);
9.     if($this-
>statusBerhasilOlahPrefixSuffix ==
0){
10.      $this-
>statusKata = "tidakditemukan";
11.      return $kata;
12.    }

```

Source code di atas merupakan souch code pengolahan prefix awalan kata imbuhan dan suffix atau ahiran kata levenstein distance. Dan sebelum user memasukkan kata yang akan di terjemahkan ke Bahasa jawa maka kata tersebut di sortir apakah mengandung prefix atau suffix yang salah. Jika prefix dan suffix itu salah maka akan di proses oleh metode levenstein disntace.

```

1. insertion levenshtein di
2. else if(@$this-
>cek_nonVokal($kata[1])==1 && $this
-
>levenshtein_pre($kata[0], "di")<=1)
{
3.   $this-
>status_ditemukan_pre = 1;
4.   $asumsi_pre="di";
5.   $this-
>dapatkan_kata_setelah_pre(1,$kata)
;
6. }
7.
8. //substraction levenshtein di
9. else if(@$this-
>cek_nonVokal($kata[2])==1 && $this
-
>levenshtein_pre($kata[0].$kata[1],
"di")<=1){
10.  $this-
>status_ditemukan_pre = 1;
11.  $asumsi_pre="di";
12.  $this-
>dapatkan_kata_setelah_pre(2,$kata)
;
13. }
14.
15. //deleteion levenshtein di
16. else if(@$this-
>cek_nonVokal($kata[3])==1 && $this
-
>levenshtein_pre($kata[0].$kata[1].
$kata[2], "di")<=1){

```

```

17.   $this-
>status_ditemukan_pre = 1;
18.   $asumsi_pre="di";
19.   $this-
>dapatkan_kata_setelah_pre(3,$kata)
;
20. }
21.

```

Source code di atas digunakan untuk deletion dan subtraction *prefik* atau imbuhan awal, yang yang ada di metode *levenstein distance* yaitu dari *insert*, *substraction*, dan *deletion*.

```

1.
2.   if($kata[$size-3].$kata[$size-
2].$kata[$size-1] == "kan"){
3.     $this-
>set_kata_sebelum_suf(-3,$kata);
4.     $asumsi_suf = "kan";
5.   }
6.   else if($kata[$size-
3].$kata[$size-2].$kata[$size-
1] == "lah"){
7.     $this-
>set_kata_sebelum_suf(-3,$kata);
8.     $asumsi_suf = "lah";
9.   }
10.  else if($kata[$size-
3].$kata[$size-2].$kata[$size-
1] == "nya"){
11.    $this-
>set_kata_sebelum_suf(-3,$kata);
12.    $asumsi_suf = "nya";
13.  }
14.  else if($kata[$size-
2].$kata[$size-1] == "an"){
15.    $this-
>set_kata_sebelum_suf(-2,$kata);
16.    $asumsi_suf = "an";
17.  }
18.  else if($kata[$size-
2].$kata[$size-1] == "ku"){
19.    $this-
>set_kata_sebelum_suf(-2,$kata);
20.    $asumsi_suf = "ku";
21.  }
22.  else if($kata[$size-
2].$kata[$size-1] == "mu"){
23.    $this-
>set_kata_sebelum_suf(-2,$kata);
24.    $asumsi_suf = "mu";
25.  }
26.  }
27.  }
28.  }
29.  }
30.  }
31.

```

Source code di atas digunakan untuk *suffix* contoh kata ahiran dalam pemrosesan *levenstein distance*.

```

1. function panggil_proses_translasi($
   kata){
2. header("Location:prosestranslate/g
   ethint.php?kataInput=$kata&bhsAwal=
   bsIndo&bhsTuju=bsNgoko");
3. }
    
```

Source code di atas merupakan function pemanggilan Bahasa Jawa Ngoko yang akan di translisasikan dari Bahasa Indonesia ke Bahasa Jawa Ngoko.

Adapun *pseudocode* dari algoritma *levenshtein distance* (Ilmy et al., 2006) adalah sebagai berikut:

```

function LevDis (s1 : string, s2 : string) : integer
    kamus
    i, j, cost : integer
    m : array[0..s1.length, 0..s2.length] of integer
    algoritma
    for i ← 0 to s1.length do
    for j ← 0 to s2.length do
    if i = 0 then
    m[i,j] ← j {perbandingan dengan kosong}
    else if j = 0 then
    m[i,j] ← i {perbandingan dengan kosong}
    else {implementasi pemrograman dinamis}
    if s1[i] = s2[j] then
    cost ← 0 else cost ← 1
    m[i,j] = minimum (
    m[i-1,j-1] + cost, {substitution}
    m[i-1,j] + 1, {penghapusan}
    m[i,j-1] + 1, {penambahan} )
    Return m[s1.length,s2.length]
    
```

Gambar 5. *Pseudocode* dari Algoritma *Levenshtein distance*

4.2. Hasil Output program

Untuk mengetahui aplikasi yang dibuat apakah sudah sesuai dengan membandingkan hasil dari sumber *input* dan target *output* yang diinginkan.

4.3 Confusion matrix

Untuk melihat akurasi dan presisi peneliti menggunakan confusion matrix. Matrik klasifikasi adalah alat yang memiliki fungsi untuk melakukan analisis prediksi benar dan tidak benarnya oleh model klasifikasi yang berbeda (Leidiana, 2013).

Pada tahap uji coba menggunakan *confusion matrix* yang berguna untuk mengetahui sistem sudah berfungsi sudah sesuai dengan perancangan yang sebelumnya. Pengujian sistem dilakukan dengan cara membandingkan hasil terjemah bahasa Jawa secara manual (*groundtruth*). Dengan hasil terjemah bahasa Jawa yang secara otomatis ditampilkan oleh sistem. Data untuk di terjemahkan ke bahasa Jawa yang digunakan yaitu pada sub bab sebelumnya, dengan mengambil data dari kalimat yang terdiri 172 kata, yang diambil dari cerita dongeng si kancil dan si buaya dari *website yang terkenal yaitu dongengceritarakyat.com*, dan dengan sebanyak kosakata yang di sediakan oleh sistem pada *database* dalam pernerjemahan terdapat pada Tabel 10. Berikut:

Tabel 4 Database kosakata Bahasa Indonesia dan Bahasa Jawa

Kosakata bahasa Jawa yang ada di database				
Ae	elek	icip	saben	yaiku
Abang	eling	idu	sabut	yais
Aboh	embah	ijo	sabun	yatim
Abot	embuh	saiki	yen
....	yo
Babu	gampang	udan
Babah	garing	ogah	udar	zaman
Babak	gatek	ombe	udek	zamin
Babakan	gawa	opah	udel	zamin dar
....	ora	zamrud
Total kosakata dalam database: 30382				

Keterangan: Jumlah kata yang ada di database sebanyak 30382 kata.

Rumus *confusion matrix* pada tabel 5 Berikut;

Tabel 5 Confusion Matrix

		Rumus Hasil	
		TRUE	FALSE
Nilai Prediktif	TRUE	TP (True Positif)	FP (False Positif)
	FALSE	FN (False Negatif)	TN (True Negatif)

Keterangan:

1. *True Positive* (TP): jumlah *record* yang memberikan hasil sesuai dengan *query* dan memang benar sesuai.
2. *False Positive* (FP): jumlah *record* yang memberikan hasil prediksi sesuai dengan *query*, akan tetapi tidak sesuai setelah di cek oleh (manusia).
3. *True Negative* (TN): jumlah *record* yang seharusnya memberikan prediksi tidak sesuai dengan *query* dan memang sebenarnya tidak sesuai.
4. *False Negative* (FN): jumlah *record* yang memprediksi tidak sesuai dengan *query*, tapi menurut (manusia) memang sesuai.

Berdasarkan Tabel tersebut, *precision*, *recall* dan *accuracy* dapat dirumuskan dengan persamaan berikut:

$$Precision = \frac{a}{a+b} \times 100\%$$

$$Recall = \frac{a}{a+c} \times 100\%$$

$$Accuracy = \frac{a+d}{a+b+c+d} \times 100\%$$

$$F - measure = 2 \times \frac{precision \times recall}{precision + recall}$$

F-measure merupakan bobot dari *precision* dan *recall* yang merupakan ukuran timbal balik di antara keduanya.

Uji coba sistem dilakukan dengan menggunakan data sejumlah dua belas

query yang berbeda. Dari dua belas *query*, akan didapatkan tujuh hingga tiga puluh dua kemungkinan nilai tertinggi sebagai prediksi jawaban terjemah bahasa Indonesia-Jawa. Dari masing-masing *query* yang berhasil di *input*-kan,

Dapat dilihat hasil dari pengujian *confusion matrix* pada table 6.

Tabel 6. Hasil Pengukuran *precision*, *recall*, *f-measure*, dan *accuracy*

Precision	Recall	Accuracy	F-measure
95,32%	100%	99,97%	97,52%

Berdasarkan hasil uji coba yang telah dilakukan, *query* answering system tanaman obat ini memiliki tingkat keakuratan mencapai 87.91%. Adapun beberapa masalah dan penyebab ketidakakuratan yang terjadi di dalam sistem ini adalah sebagai berikut:

Berdasarkan hasil pengujian dengan uji coba sebanyak jumlah *input* yang dilakukan, dapat disimpulkan bahwa penggunaan metode *approximate string matching* dalam pembetulan kata bahasa Indonesia sebelum di terjemahkan ke bahasa Jawa tidak terdapat banyak kesalahan proses dan secara fungsional menghasilkan dengan benar sesuai sumber *input* dan target *output* pada proses *approximate string matching* tersebut. Dan proses terjemah sesuai dengan *input* bahasa Indonesia dan *output* bahasa Jawa sesuai yang diharapkan.

5. KESIMPULAN

Berdasarkan alur uji coba yang dilakukan oleh peneliti dengan judul Implementasi Algoritma Levenshtein Distance Dengan Restful Web Service Pada Kata Bahasa Indonesia Ke Bahasa Jawa Berbasis Web, dengan uji coba data sejumlah 172 kata, dan sebanyak 30.382 kosakata yang ada di *database*. Dapat

ditarik kesimpulan bahwa:

1. Pengujian dilakukan terhadap dua belas *query*, didapatkan hasil persentase nilai *precision* sebesar 95,32%, nilai *recall* 100%, nilai *accuracy* 99,97%, dan nilai *f-measure* 97,51%
2. Penggunaan metode Levenshtein Distance pada *query* sebanyak dua belas *query* berdasarkan *input* sumber dan *output* target dikatakan efektif karena mendapatkan persentase yang cukup tinggi.
3. Tidak terdapat banyak kesalahan pada proses, sehingga secara fungsional dapat menampilkan hasil yang sesuai harapan.

Adapun saran bagi peneliti selanjutnya adalah:

1. Perlu adanya *improvement* pada metode *Approximate String Matching* dengan algoritma *Levenshtein Distance* yaitu membuat perbedaan bobot atau nilai antara penyisipan, penghapusan, dan penggantian karakter. Perbedaan bobot juga dapat dikembangkan lagi dengan melihat letak penyisipan, penghapusan, atau penggantian karakter yang terjadi. Yang dimaksud bobot perbedaan yaitu bertujuan untuk mendapatkan variasi yang mirip dari *query* yang asli.
2. Penerapan metode *Levenshtein Distance* pada penerjemah Indonesia-Jawa dengan pembetulan kata dapat juga dikombinasikan dengan algoritma lain yang berhubungan dengan pencarian *similarity* antar dua buah kata misalnya algoritma *N-Gram* atau *Knuth Morris Pratt* (KMP). Sehingga didapatkan variasi yang lebih relevan dengan *query* asli.

DAFTAR PUSTAKA

- Amri, A. (2020). DAMPAK COVID-19 TERHADAP UMKM DI INDONESIA. *BRAND Jurnal Ilmiah Manajemen*
- Pemasaran*, 2(1), 123–131.
- Ardana, D. M. J. (2019). PERANAN KOMUNIKASI STAF DALAM SOSIALISASI PROGRAM KERJA DI PUSKESMAS TEJAKULA II KECAMATAN TEJAKULA. *Locus*, 11(1).
- Arisandy, D., Rizkika, D. P., & Astika, T. D. (2019). EKSISTENSI BAHASA INDONESIA PADA GENERASI MILENIAL DI ERA INDUSTRI 4.0. *Bahastra: Jurnal Pendidikan Bahasa Dan Sastra Indonesia*, 3(2), 247–251.
- Budiono, S., Erikha, F., & Pahlevi, E. R. (2018). *Penelusuran Hubungan Bahasa Jawa Dialek Banyumas dengan Bahasa Jawa Dialek Banyuwangi Berdasarkan Variasi Bahasa Jawa Relationship Identification Banyumas Dialect in Central Java with Banyuwangi Dialect in East Java Based on Javanese Variation*.
- Dwitiyastuti, R. N., Muttaqin, A., & Aswin, M. (2013). Pengoreksi Kesalahan Ejaan Bahasa Indonesia Menggunakan Metode Levenshtein Distance. *Jurnal Mahasiswa TEUB*, 1(2).
- Ilmy, M. B., Rahmi, N., & Bu'ulölö, R. L. (2006). Penerapan Algoritma Levenshtein Distance untuk Mengoreksi Kesalahan Pengejaan pada Editor Teks. *Departemen Teknik Informatika, Institut Teknologi Bandung*.
- Indra, Z., & Trisnawati, L. (2018). Pengembangan intelligent data collector untuk analisis big data artikel berita online. *Rabit: Jurnal Teknologi Dan Sistem Informasi Univrab*, 3(1), 47–57.
- Leidiana, H. (2013). Penerapan algoritma k-nearest neighbor untuk penentuan resiko kredit kepemilikan kendaraan bermotor. *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, 1(1), 65–76.
- Noerwidi, S. (2008). Awal Pendaratan Austronesia di Pantai Utara Jawa, Sebuah Prospek Melacak Nenek Moyang Etnis Jawa. *Pertemuan Ilmiah Arkeologi XI*.
- Peggy, P. (2014). *Optimasi pencarian kata*

pada aplikasi penerjemah bahasa mandarin–indonesia berbasis android dengan algoritma levenshtein distance. Universitas Multimedia Nusantara.

- Rifa'i, A. M. (2015). Nasionalisme dalam Perspektif Bahasa Sebagai Perwujudan Jati Diri Bangsa. *Al-Mabsut: Jurnal Studi Islam Dan Sosial*, 9(2), 155–180.
- Suryadi, M. (2013). *Penggunaan Tingkat Tutur Bahasa Jawa Ngoko dan Krama pada Ranah Keluarga dan Masyarakat di Kota Semarang dan Kota Pekalongan*. UNS (Sebelas Maret University).
- Wilian, S. (2006). Tingkat tutur dalam bahasa Sasak dan bahasa Jawa. *Wacana*, 8(1), 32–53.