Jurnal Teknik Informatika Vol. 14 No. 1, April 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: http://dx.doi.org/10.15408/jti.v14i1.15010

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

STEMMING BAHASA JAWA MENGGUNAKAN *DAMERAU LEVENSHTEIN DISTANCE* (DLD)

Aji Prasetya Wibawa[1], Muhammad Nu'man Hakim[2]

[1,2]Teknik Informatika, Fakultas Teknik
Universitas Negeri Malang
Jalan Semarang No.5, Sumbersari, Kecamatan Lowokwaru, Kota Malang, Jawa Timur 65145
E-mail : [1]aji.prasetya.ft@um.ac.id, [2]*mnumanhakim@gmail.com*

Artikel:
Diterima:11 Maret 2020
Direvisi: 13 September 2020
Diterbitkan: 06 September 2021

***Alamat Korespondensi:**
aji.prasetya.ft@um.ac.id

ABSTRACT

Stemming is one of the essential stages of text mining. This process removes prefixes and suffixes to produce root words in a text. This study uses a string matching algorithm, namely Damerau Levenshtein Distance (DLD), to find the basic word forms of Javanese. Test data of 300 words that have a prefix, insertion, suffix, a combination of prefix and suffix, and word repetition. The results of this study indicate that the Damerau Levenshtein Distance (DLD) algorithm can be used for Stemming Javanese text with an accuracy value of 49.6%.

Keywords: *Basic words, Javanese, Damerau Levenshtein Distance*

ABSTRAK

Stemming adalah salah satu tahap penting dalam *textmining*. Proses ini menghilangkan awalan dan akhiran untuk menghasilkan kata dasar pada sebuah teks. Penelitian ini menggunakan algoritma string matching yaitu *Damerau Levenshtein Distance* (DLD) untuk mencari bentuk kata dasar dari Bahasa Jawa. Data uji sebanyak 300 kata yang memiliki awalan, sisipan, akhiran, gabungan dari awalan dan akhiran serta pengulangan kata. Hasil penelitian ini menunjukkan bahwa algoritma *Damerau Levenshtein Distance* (DLD) dapat digunakan untuk *Stemming* teks bahasa jawa dengan nilai akurasi 49,6%.

Kata Kunci: *Kata Dasar, Bahasa Jawa, Stemming, Damerau Levenshtain Distance*

Jurnal Teknik Informatika Vol. 14 No. 1, April 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: http://dx.doi.org/10.15408/jti.V14I1.(id article)

I. PENDAHULUAN

Bahasa Jawa merupakan salah satu bahasa daerah di Indonesia yang memiliki ragam keunikan. Salah satu keunikan tersebut berupa ragam imbuhan yang terdiri dari awalan, sisipan, akhiran serta gabungan dari awalan dan akhiran [1]. Istilah imbuhan awalan, sisipan, akhiran, serta gabungan dari awalan dan akhiran dalam Bahasa Jawa berbeda dengan Bahasa Indonesia. Dalam Bahasa Jawa awalan disebut *ater-ater*, sisipan disebut *seselan* dan akhiran disebut *penambang* [2]. *Ater-ater* terbagi dalam tiga bagian: *Hanuswara* (*n-, m-, ny-* dan *ng-), ater-ater tripurusa* (*dak-, ko-* dan *di-0,* dan *ater-ater liyane* (*a-, pa-, pan-, pang-, pi-, pra-, tak-, tar-, ka-, ke-, sa-, kuma-,* dan *kapi-*). Se*selan* terdiri dari beberpa bentuk yaitu : *-um-, -in-, -el-*, dan *-er-*. P*enambang* terdiri dari *"-a, -e, -ii -an, -ake, -en, -na, -ne, -ku,* dan *-mu"* [2]. Imbuhan yang ditulis serangkai dengan kata dasar akan membentuk kata berimbuhan yang mungkin akan berbeda fungsi dan makna dari kata dasarnya.

Untuk mengembalikan kata berimbuhan ke bentuk kata dasarnya digunakan algoritma *Stemming*. Algoritma *Stemming* digunakan dalam sistem *Information Retrieval* [3] untuk menghilangkan awalan, sisipan, akhiran, gabungan dari awalan dan akhiran, serta pengulangan kata [4]. Salah satu algoritma *Stemming* yang sering digunakan adalah *porter*[5]–[7], *Lovins* [8], *Dawszon* [9], dan *Husk* [10]. Algoritma-algoritma tersebut umum digunakan untuk menghapus awalan dan akhiran dalam bahasa Inggris. Konsep penghapusan menggunakan aturan *lexicon* dan *rule-based* [11]. Selain itu algoritma khusus berdasar aturan bahasa tertentu juga pernah dikembangkan antara lain untuk bahasa Arab [12], Hindi [13], Uzbek [11], dan Indonesia [6] [14]. *Stemming* bahasa Jawa dapat dilakukan dengan memodifikasi aturan *Stemming* bahasa Indonesia.

Beberapa penelitian telah membahas algoritma *Stemming* untuk bahasa jawa. Metode Nazief dan Adriani dimodifikasi sesuai aturan morfologi bahasa agar dapat digunakan untuk *Stemming* teks dalam bahasa Jawa [15]. ECS *stemmer* juga pernah digunakan untuk menghasilkan kata dasar Jawa [2], [16]. Metode-metode ini berbasis bentuk kebahasaan yang sudah umum digunakan. Hasilnya cukup akurat namun terkesan kurang adaptif karena akan sulit mengadaptasi perubahan morfologis, fonologis, dan sintaksis akibat dari perkembangan zaman [17].

Penelitian ini mengusung ide *Stemming* berdasarkan asumsi bahwa kata berimbuhan memiliki kemiripan ejaan dengan kata dasarnya. Oleh karena itu, artikel ini akan membahas penggunaan algoritma string matching dalam proses *Stemming* bahasa jawa. Algoritma *Stemming* yang digunakan pada penelitian ini menggunakan algoritma *Damerau Levenshtein Distance* (DLD) [18]–[21]. Algoritma ini diharapkan dapat menyelesaikan masalah berupa *over Stemming, under Stemming, unchanged,* dan *spelling exception*. Penelitian ini dilakukan guna menguji akurasi DLD untuk *Stemming* kata Bahasa Jawa.

II. METODOLOGI

Dataset yang digunakan berasal dari penelitian sebelumnya [16] [22] yang terdiri dari 300 kata dengan detail pada Tabel 1. Pada Tabel 1, jumlah kata untuk masing-masing imbuhan adalah 60. Sehingga untuk kelima jenis imbuhan terdapat sebanyak 300 kata.

Tabel 1. Jenis dan *jumlah kata berimbuhan*

No.	Jenis Imbuhan	Jumlah Kata
1	Awalan	60
2	Sisipan	60
3	Akhiran	60
4	Gabungan	60
5	Pengulangan	60

Stemming data pada Tabel 1 dilakukan menggunakan *Damerau Levenshtein Distance* (DLD). DLD digunakan untuk *Error Spelling Correction* yaitu proses menentukan penulisan suatu kata yang salah ejaan [23]. Algoritma ini merupakan pengembangan algoritma *Levenshtein Distance* yang awalnya memiliki operasi *insertion, deletion, substitution* [24] dan disempurnakan dengan menambah *transposition* untuk dua karakter yang bersebelahan [25]. Gambar 1 menunjukkan algoritma DLD dalam bentuk *pseudocode*

Jurnal Teknik Informatika Vol. 14 No. 1, April 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: http://dx.doi.org/10.15408/jti.V14I1.(id article)

```
algorithm DLD
   input: strings x[1..length(x)], y[1..length(y)]
   output: distance, integer
d[0..length(x), 0..length(y)] //array 2 dimensi dengan
ukuran (x)+1, (y)+1
        for i := 0 to length(x) do
      d[i, 0] := i
   for j := 0 to length(b) do
      d[0, j] := j
      for i := 1 to length(x) do
      for j := 1 to length(y) do
         if x[i] = y[j] then
            cost := 0
         else
            cost := 1
         d[i, j] := minimum(d[i-1, j] + 1,    // deletion
                           d[i, j-1] + 1,    // insertion
                           d[i-1, j-1] + cost)  // substitution
         if i > 1 and j > 1 and a[i] = y[j-1] and x[i-1] =
y[j] then
            d[i, j] := minimum(d[i, j],
                           d[i-2, j-2] + 1)  // transposition
   return d[length(x), length(y)]
```

Gambar 1. *Pseudocode damerau levenshtein distance* (DLD)

DLD menggunakan perhitungan jarak terdekat berdasarkan saran kata yang terdapat di kamus kata dasar. Pada Tabel 2 kata berimbuhan MAPAG dan hasil kata *Stemming* PAPAG. Mengikuti *pseudocode* pada Gambar 1, perhitungan nilai jarak terdekat dihitung mulai indeks pertama setiap kolom dan baris sampai dengan kolom dan baris terakhir. Sebagai contoh, perbandingan M (huruf pertama MAPAG) dengan P (huruf pertama PAPAG) bernilai 1 karena berbeda huruf. Selanjutnya, A (huruf kedua MAPAG) dengan A (huruf kedua PAPAG) tetap bernilai 1 karena berbeda hurufnya sama. Nilai perubahan kata imbuhan dapat diketahui pada kolom dan baris terakhir. Pada contoh kasus ini, jarak kedua kata MAPAG dan PAPAG adalah 1. Hal ini disebabkan karena perbedaan kedua kata hanya terjadi di huruf pertama.

DLD dibuat dengan bahasa pemrograman *python*. Akurasi DLD didapatkan dengan menghitung jumlah kata yang benar, kemudian dibagi dengan jumlah kata keseluruhan, seperti pada persamaan (1).

$$Akurasi = \frac{Jumlah\ Kata\ benar}{Jumlah\ Kata\ seluruhnya} x100\% \qquad (1)$$

		P	A	P	A	G
	0	1	2	3	4	5
M	1	1	2	3	4	5
A	2	2	1	2	2	3
P	3	2	2	1	2	3
A	4	3	2	2	1	2
G	5	4	3	3	2	1

Gambar 2. Matriks jarak kata MAPAG dan PAPAG

III. HASIL DAN PEMBAHASAN

Tabel 2, menunjukkan hasil akurasi *stemmer* algoritma *Damerau Levenshtein Distance* (DLD) memiliki nilai akurasi 49.6%. DLD dapat melakukan proses *Stemming* Bahasa Jawa, namun pada beberapa kasus penggunaannya terlihat kurang efektif.

Tabel 2. Hasil akurasi stemmer damerau levenshtein distance

Imbuhan	Jumlah Kata Benar	Jumlah Kata Salah	Akurasi(%)
Awalan	34	26	56,6
Sisipan	39	21	65
Akhiran	47	13	78,3
Gabungan	27	33	45
Perulangan	2	58	3,33
Jumlah	149	151	49,6

Kesalahan pertama, pada kata berawalan. Misalkan *Stemming* kata *"NGADHO"* yang seharusnya menghasilkan kata dasar "*KADHO"* ternyata menghasilkan kata dasar yang tidak sesuai yaitu *"GADHO"*. Gambar 3 dan Gambar 4 berturut-turut mengilustrasikan matriks jarak aantara *"NGADHO"* dengan "*KADHO"* dan *"GADHO"*.

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: http://dx.doi.org/10.15408/jti.V14I1.(id article)

		K	A	D	H	O
	0	1	2	3	4	5
N	1	1	2	3	4	5
G	2	2	2	3	4	5
A	3	3	2	3	4	5
D	4	4	3	2	3	4
H	5	5	4	3	2	3
O	6	6	5	4	3	2

Gambar 3. Matriks jarak kata *NGADHO* dan *KADHO*

Pada Gambar 3 terlihat bahwa terdapat dua perbedaan huruf di awal kata "NGADHO" dan "*KADHO*". Sedangkan pada Gambar 4 terlihat bahwa kata "*NGADHO*" dan "*GADHO*" hanya berbeda pada huruf awal yaitu antara "N" dan "G". Oleh karena itu, algoritma DLD tidak menampilkan kata "*KADHO*" karena perubahan kata yang lebih banyak dibandingkan dengan "*GADHO*".

		G	A	D	H	O
	0	1	2	3	4	5
N	1	1	2	3	4	5
G	2	1	2	3	4	5
A	3	2	1	2	3	4
D	4	3	2	1	2	3
H	5	4	3	2	1	2
O	6	5	4	3	2	1

Gambar 4. Matriks jarak kata *NGADHO* dan G*ADHO*

Kesalahan kedua, pada imbuhan sisipan, misalkan pada kata *"KROGEL"*. Kata *"KROGEL"* menghasilkan kata *"KOGEL"* dengan nilai perubahan sebanyak 1 kali. Hal tersebut terjadi disebabkan terhapusnya huruf *"R"* dan merupakan kata terdekat yang mengalami perubahan berdasarkan kamus. Kata dasar seharusnya dari *"KROGEL"*. yaitu, *"OGEL"* yang mengalami perubahan sebanyak dua kali pada huruf "K*"* dan *"R"* pada *"KROGEL"*. Kata *"KOGEL"* memiliki perubahan paling sedikit dibandingkan dengan kata dasar yang lain dalam dataset, sehingga dianggap sebagai kata dasar yang benar dari kata *"KROGEL"*. Walaupun pada kenyataannya *"KROGEL"* dan *"KOGEL"* memiliki arti dan pengucapan yang berbeda. *"KROGEL"* berarti kejang sedangkan *"KOGEL"* berarti kecewa. Pengucapan huruf "E" pada *"KROGEL"* seperti pada kata "ES" sedangkan "E" pada*"KOGEL"* diucapkan seperti "KELAPA". Untuk mengatasi hal ini perlu ditambahkan aturan kebahasaan yang berkaitan dengan kajian fonetik [26].

Masalah *stemming* ketiga, terjadi pada akhiran yakni *"GALANGAN"* yang seharusnya dibentuk dari kata dasar "*GALANG"* dengan akhiran "AN". Kata *"GALANGAN"* menghasilkan kata *"ABANGAN"* yang memiliki jarak perubahan kata sebanyak dua kali. Jarak ini nilainya sama dengan jarak perubahan kata *"galangan"* dengan kata *"GALANG"*. DLD menampilkan kata *"ABANGAN"* karena prioritas pengambilan kata dasar dilakukan berdasar urutan abjad kata yang terdapat dalam kamus.

Kesalahan keempat, pada imbuhan gabungan awalan dan akhiran. Misalkan pada kata *"SAWISE"* yang menghasilkan kata *"AWIS"* dengan nilai perubahan sebanyak dua kali. Perubahan tersebut dikarenakan penghapusan huruf *"S"* diawal kata dan huruf *"E"* diakhir kata. Kata dasar sesungguhnya "WIS" tidak terambil karena setidaknya terdapat tiga perbedaan dengan kata berimbuhan *"SAWISE"*.

Kesalahan terakhir, pada imbuhan perulangan kata. Misalkan pada kata *"SEYAT-SEYOT"* dan *"ILANG-ILANGAN"*. Proses *Stemming* tidak menghasilkan kata dasar. Melainkan mencari kata perulangan yang memiliki nilai perubahan lebih sedikit dari kata imbuhan. Kata *"SEYAT-SEYOT"* menghasilkan kata *"NYAT-NYUT"* yang mengalami nilai perubahan sebanyak lima kali. Sedangkan kata dasar yang sebenarnya yaitu *"SEYOT"* yang memiliki nilai perubahan sebanyak enam kali.

Proses kesalahan *Stemming* disebabkan oleh algoritma DLD mencari kata yang memiliki perubahan paling sedikit. Kenyataannya, kata yang memiliki perubahan paling sedikit belum tentu kata dasar sebenarnya. Kedepan seleksi algoritma DLD perlu dilakukan misalnya dengan memprioritaskan kata dengan jumlah huruf paling sedikit dibandingkan dengan urutan kemunculannya dalam kamus. Hal ini berdasar pada kenyataan bahwa kata berimbuhan memiliki jumlh karakter lebih banyak jika dibandingkan dengan kata dasarnya [27], [28]. Untuk kata ulang, DLD hanya perlu diterapkan sampai pada huruf terakhir sebelum tanda hubung. Hal ini diharapkan dapat meningkatkan akurasi *Stemming* berbasis DLD.

Mengacu pada akurasi algoritma DLD untuk *Stemming* pada Tabel 2, hasil terbaik ditunjukkan pada *Stemming* kata yang mengandung akhiran. Struktur kata ini sangat mirip dengan struktur bahasa asing yang memiliki struktur lebih sederhana seperti Inggris [29] dan Uzbek [11] yang tidak memiliki sisipan.

IV. PENUTUP

Ide penggunaan algoritma *Damerau Levenshtein Distance* (DLD) sebagai *stemmer* dapat dikatakan sebagai gagasan baru. Hasil penelitian fasa awal ini mendapatkan nilai akurasi 49,6% dengan jumlah kata benar sebanyak 149 kata dan jumlah kata salah sebanyak 151 kata. Hal ini menunjukkan bahwa dapat digunakan untuk *Stemming* Bahasa Jawa.

Sampai saat ini *stemmer* yang dikembangkan belum ada yang memeuhi kaidah kebahasaan. Peluang peningkatan akurasi *stemmer* berbasis DLD masih terbuka lebar untuk menambah khasanah *stemmer* berbasis *rule* dan leksikon. Perkembangan yang mungkin muncul di masa mendatang antara lain dengan cara merubah prioritas pengambilan data serta menambahkan aturan kebahasaan yang terkait tentang kajian fonetik bahasa Jawa.

DAFTAR PUSTAKA

[1] F. Amin, W. Hadikurniawati, S. Wibisono, H. Februariyanti, and J. S. Wibowo, "A hybrid method of rule-based and string matching stemmer for Javanese language," J. Theor. Appl. Inf. Technol., vol. 95, no. 19, pp. 4973–4982, 2017.

[2] F. Amin, Purwatiningtyas, P. Utomo, S. Ramadhanu, and S. E. Cahya, "Stemmer Bahasa Jawa Ngoko dengan Metode Affix Removal Stemmers (Rule Based Approach)," J. Din., vol. 21, no. 1, pp. 16–24, 2016.

[3] C. D. Manning, P. Raghavan, and H. Schütze, An Introduction to Information Retrieval. Cambridge university press, 2009.

[4] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," Universiteit van Amsterdam The Netherlands, 2003.

[5] A. R. Kulkarni and S. D. Mundhe, "An Application of Porters Stemming Algorithm for Text Mining in Healthcare," Int. J. Manag. IT Eng., vol. 7, no. 11, pp. 223–228, 2017.

[6] L. Agusta, "Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia," in Konferensi Nasional Sistem dan Informatika, 2009, pp. 196–201.

[7] B. V. Indriyono, E. Utami, and A. Sunyoto, "Pemanfaatan Algoritma Porter Stemmer Untuk Bahasa Indonesia Dalam Proses Klasifikasi Jenis Buku," J. Buana Inform., vol. 6, no. 4, pp. 301–309, Oct. 2015.

[8] M. Panda, "Developing an Efficient Text Pre-Processing Method with Sparse Generative Naive Bayes for Text Mining," Int. J. Mod. Educ. Comput. Sci., vol. 10, no. 9, pp. 11–19, 2018.

[9] A. Hegde and S. K. Shetty, "A Study on Stemming Algorithms," Int. J. Emerg. Trends Sci. Technol., vol. 2, no. 5, pp. 2301–2364, 2015.

[10] A. Schofield and D. Mimno, "Comparing Apples to Apple: The Effects of Stemmers on Topic Models," Trans. Assoc. Comput. Linguist., vol. 4, pp. 287–300, Dec. 2016.

[11] A. Ismailov, M. M. A. Jalil, Z. Abdullah, and N. H. A. Rahim, "A comparative study of Stemming algorithms for use with the Uzbek language," in 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), 2016, pp. 7–12.

[12] R. Elhassan and M. Ahmed, "Arabic text Stemming Effectiveness," in 2016 Conference of Basic Sciences and Engineering Studies (SGCAC) Arabic, 2016, pp. 88–93.

[13] A. Sharma, R. Kumar, and V. Mansotra, "Proposed Stemming Algorithm for Hindi Information Retrieval," Int. J. Innov. Res. Comput. Commun. Eng. (An ISO Certif. Organ., vol. 3297, no. 6, pp. 11449–11455, 2016.

[14] M. S. H. Simarangkir, "Studi Perbandingan Algoritma - Algoritma Stemming Untuk Dokumen Teks Bahasa

Indonesia," J. Inkofarall, vol. 1, no. 1, pp. 40–46, 2017.

[15] A. P. Wibawa, F. A. Dwiyanto, I. A. E. Zaeni, R. K. Nurrohman, and A. Afandi, "Stemming javanese affix words using Nazief and Adriani modifications," J. Inform., vol. 14, no. 1, p. 36, 2020.

[16] N. Hidayatullah, A. P. Wibawa, and H. A. Rosyid, "Penerapan ECS Stemmer untuk Modifikasi Nazief & Adriani Berbahasa Jawa," J. RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 3, no. 3, pp. 343–348, Dec. 2019.

[17] R. Romadhianti, "Fenomena Bahasa Gaul dalam Kacamata Morfologis, Fonologis, dan Sintaksis," PESONAJurnal Kaji. Bhs. dan Sastra Indones., vol. 5, no. 11–18, 2019.

[18] F. J. Damerau, "A technique for computer detection and correction of spelling errors," Commun. ACM, vol. 7, no. 3, pp. 171–176, 1964.

[19] P. Santoso, P. Yuliawati, R. Shalahuddin, and A. P. Wibawa, "Damerau Levenshtein Distance for Indonesian Spelling Correction," J. Inform., vol. 13, no. 2, p. 11, 2019.

[20] A. Kutuzov, "Improving English-Russian sentence alignment through POS tagging and Damerau-Levenshtein distance," in Proceedings of the 4th Biennial International Work, 2013, pp. 63–68.

[21] P. Santoso, P. Yuliawati, R. Shalahuddin, and I. A. E. Zaeni, "Penghapusan kolom dan baris pertama pada matriks distance untuk optimasi spell checker damerau-levenshtein distance," Sains, Apl. Komputasi dan Teknol. Inf., vol. 2, no. 2, pp. 57–63, 2020.

[22] A. P. Wibawa, F. A. Dwiyanto, I. A. E. Zaeni, R. K. Nurrohman, and A. Afandi, "Stemming javanese affix words using Nazief and Adriani modifications," J. Inform., vol. 14, no. 1, p. 36, Jan. 2020.

[23] T. N. Maghfira, I. Cholissodin, and A. W. Widodo, "Deteksi Kesalahan Ejaan dan Penentuan Rekomendasi Koreksi Kata yang Tepat Pada Dokumen Jurnal JTIIK Menggunakan Dictionary Lookup dan Damerau-Levenshtein Distance," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 1, no. 6, pp. 498–506, 2017.

[24] J. Jupin, J. Y. Shi, and Z. Obradovic, "Understanding Cloud Data Using Approximate String Matching and Edit Distance," in 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, 2012, pp. 1234–1243.

[25] A. Pahdi, "Koreksi Ejaan Istilah Komputer Berbasis Kombinasi Algoritma Damerau- Levenshtein dan Algoritma Soundex," Sentra Penelit. Eng. dan Edukasi, vol. 8, no. 2, pp. 1–8, 2016.

[26] S. Nafisah, "Proses Fonologis dan Pengkaidahannya dalam Kajian Fonologi Generatif," DEIKSIS, vol. 9, no. 01, p. 70, Jan. 2017.

[27] M. S. Utomo, "Implementasi Stemmer Tala pada Aplikasi Berbasis Web," J. Teknol. Inf. Din., vol. 18, no. 1, pp. 41–45, 2013.

[28] R. Mandala, E. Koryanti, R. Munir, and H. Harlili, "Sistem Stemming Otomatis untuk Kata dalam Bahasa Indonesia," in Seminar Nasional Aplikasi Teknologi Informasi 2004, 2004, pp. 29–36.

[29] K. Mena, Vera Veti & Saputri, "Prefixes and Suffixes in the Descriptive Texts of Student ' S," English COmmunity J., vol. 2, pp. 175–182, 2018.