KOREKSI KESALAHAN EJAAN PADA HASIL KONVERSI *FILE* DALAM EKSTRAKSI INFORMASI

Abdul Rohim[1], Ken Kinanti Purnamasari[2]

[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : rohimabdul531@gmail.com[1], ken.kinanti@email.unikom.ac.id[2]

ABSTRAK

File yang dihasilkan dari proses konversi *file* dapat memiliki kesalahan ejaan yang mencakup kesalahan ejaan *non-word* dan *word-boundary*. Kesalahan ejaan *non-word* disebabkan oleh ejaan yang tidak terdapat pada kamus. Kesalahan ini dapat ditangani dengan menggunakan dua metode pengoreksi ejaan yaitu metode *Soundex* untuk memberikan rekomendasi perbaikan kata dan metode *Damerau-Levenshtein Distance* digunakan untuk memilih perbaikan kata dari yang direkomendasikan. Metode *Soundex* mengoreksi ejaan berdasarkan kemiripan ucapan (*phonetic string matching*) dan metode *Damerau-Levenshtein Distance* berdasarkan kemiripan tulisan (*approximate string matching*). *Soundex* akan mencari kata yang memiliki kemiripan berdasarkan kode fonetis yang sama sebagai kata yang direkomendasikan dan *Damerau-Levenshtein Distance* mencari nilai jarak edit terkecil dari kata yang dianggap salah eja dengan kata yang direkomendasikan sedangkan kesalahan ejaan *word-boundary* karena letak spasi yang tidak sesuai antar kata sehingga menyebabkan dua kata menjadi satu kata. Kesalahan *word-boundary* ditangani dengan mencari dua kata pada kamus Indonesia (KBBI), apabila dua kata tersebut digabungkan akan sama dengan kata yang dianggap salah eja *word-boundary*. Berdasarkan hasil pengujian terhadap 11 dokumen disertasi berupa *cover* dan abstrak, deteksi kata yang salah eja memiliki rata-rata *recall* sebesar 100% dan *precision* sebesar 59,5%, sedangkan hasil perbaikan kata memiliki rata-rata *recall* sebesar 100% dan *precision* sebesar 71,4%.

Kata kunci : koreksi kesalahan ejaan, *soundex*, *damerau-levensthein distance*, ekstraksi informasi, *non-word*, *word-boundary*.

1. PENDAHULUAN

Konversi *file* adalah suatu cara untuk mengubah dari satu format *file* menjadi format *file* lainnya. *File* yang dikonversi biasanya berupa dokumen berformat *.pdf* yang diubah menjadi format *.doc* ataupun *.docx*. Hal ini biasanya dilakukan agar informasi yang terdapat pada dokumen *.pdf* dapat diolah kembali karena dokumen *.pdf* bisa merupakan kumpulan dari hasil *scan hard file*. Oleh sebab itu pada saat proses konversi *file* mungkin terjadi kesalahan konversi yang menyebabkan kesalahan ejaan kata.

Kesalahan ejaan kata dibedakan menjadi dua jenis, yaitu kesalahan ejaan *non-word* yang terjadi saat kata yang salah tidak terdapat pada kamus [1][2][3] dan kesalahan ejaan *real-word* yang terjadi saat kata yang salah ada dalam kamus, tetapi bukan merupakan kata yang dimaksud [1]. Namun dalam penelitian koreksi kesalahan ejaan pada hasil konversi *file* terdapat kesalahan ejaan *word-boundary* yang terjadi karena letak spasi yang tidak sesuai untuk setiap kata [4]. Kesalahan ejaan *word-boundary* dapat terjadi pada setiap kata yang terdapat pada kamus ataupun kata yang tidak terdapat pada kamus.

2. ISI PENELITIAN

2.1 Penelitian Terkait

Beberapa penelitian yang telah dilakukan untuk mendeteksi dan mengoreksi kesalahan ejaan menggunakan teknik berdasarkan kemiripan penulisan (*approximate string matching*) maupun berdasarkan kemiripan pengucapan (*phonetic string matching*) seperti yang dilakukan oleh Utis Sutisna dan Julio Adisantoso [5] dengan teknik *approximate string matching* yang menggunakan metode *damerau-levenshtein distance* untuk mengoreksi ejaan query bahasa Indonesia. Penelitian tersebut menghasilkan peningkatan *precision* pada kinerja temu kembali sebesar 22% dari penelitian sebelumnya yang dilakukan oleh Arumsari[6] tentang koreksi ejaan dalam bahasa Indonesia menggunakan metode *levensthein*. Adapun teknik *phonetic string matching* seperti yang dilakukan oleh Fahmilu Kurniawan [7] yang membandingkan metode *soundex, phonix, metaphone* dan *caverphone* untuk rekomendasi perbaikan kata pada kasus koreksi ejaan bahasa Indonesia yang menghasilkan nilai akurasi paling besar 92% untuk metode *soundex*.

2.2 Kesalahan Ejaan

Kesalahan ejaan merupakan kesalahan yang terjadi pada suatu kata sehingga kata tersebut tidak sesuai dengan Ejaan Yang Disempurnakan (EYD) ataupun terjadinya perubahan makna pada suatu kalimat. Kesalahan ejaan dapat dibedakan menjadi 3 bagian, yaitu:

1. Kesalahan ejaan *non-word*

   Kesalahan ejaan *non-word* adalah kesalahan kata dimana kata tersebut tidak terdapat pada kamus [1][2][3]. Contoh kata sarjana yang menjadi saijana, sajana.
2. Kesalahan ejaan *real-word*

   Kesalahan ejaan *real-word* adalah kesalahan kata dimana kata yang salah ada dalam kamus, tetapi bukan merupakan kata yang dimaksud [8][9][1]. Kesalahan-kesalahan *real-word* biasanya mengganggu sintaks dan semantik dari suatu kalimat, sehingga dibutuhkan bantuan manusia untuk mengoreksinya. Contoh: Saya makan nasi kuningan.
3. Kesalahan ejaan *word-boundary*

   Kesalahan *word-boundary* adalah kesalahan kata dimana kata yang salah diakibatkan oleh penempatan spasi yang tidak sesuai [4]. Contoh kata kebiasaan makan menjadi kebiasaanmakan

2.3 Ekstraksi Informasi

Ekstraksi Informasi adalah pengambilan fakta dan informasi terstruktur dari isi koleksi teks yang besar. Pengertian fakta disini adalah beragam entitas yang diperhitungkan. Secara singkat ekstraksi informasi adalah sebuah proses mendapatkan fakta-fakta terstruktur dari data yang tersedia [10].

2.4 Metode *Rule-Based*

Sistem Berbasis Aturan (*rule based*) pada penelitian ini digunakan pada saat ekstraksi informasi untuk *ploting* data yang akan diproses dan pengubahan karakter berdasarkan list perubahan dan pengubahan karakter berdasarkan aturan tata bahasa. Data masukan yang di-*ploting* diantaranya adalah *cover*, abstrak. Adapun pengubahan karakter berdasarkan aturan tata bahasa yang dilakukan adalah pada huruf konsonan berdasarkan fonem-fonem konsonan.

2.5 *Soundex*

Metode *soundex* adalah salah satu algoritma *phonetic* (berdasarkan kemiripan ucapan) dan merupakan algoritma tertua. *Soundex* dipatenkan oleh Robert C. Russell pada tahun 1918. *Soundex* mengkodekan kata menjadi kode fonetis *soundex* ke dalam empat digit. Digit pertama merupakan huruf paling awal pada sebuah kata kemudian diikuti tiga digit angka yang didapat dari aturan *soundex* tersebut. *Soundex* membagi huruf-huruf ke dalam tujuh kelas yang berbeda [12] seperti pada tabel III.1.

Tabel II. 1 Kode numerik metode *soundex* [12]

Kode	Huruf
Tidak dikodekan	A, I, U, E, O, H, W, Y
1	B, F, P, V
2	C, G, J, K, Q, S, X, Z
3	D, T
4	L
5	M, N
6	R

2.6 *Damerau-Levensthein Distance*

Metode ini dibuat oleh Fred J. Damerau tahun 1964 yang menggambarkan perbandingan kata-kata dengan memperhatikan empat macam kesalahan [13] (diberikan contoh kata *ALPHABET*) seperti yang dicontohkan oleh damerau dalam Ulrich Pfeifer, Thomas Poersch, dan Norbert Fuhr [14], yaitu:

1. Penyisipan huruf (*insertion*), menjadi *ALLPHABET*.
2. Penghapusan huruf dalam kata (*deletion*), menjadi *ALHABET*.
3. Penggantian sebuah huruf dengan huruf yang lain (*substitution*), menjadi *ALPHIBET*.
4. Penukaran tempat dua huruf berurutan (*transposition*), menjadi *ALHPABET*.

Berikut ini adalah rumus untuk mencari nilai jarak edit pada setiap kata yang direkomendasikan.

$$d[i,j] = min\begin{cases} d[i-1,j]+1 \\ d[i,j-1]+1 \\ d[i-1,j-1]+cost \end{cases} \quad \dots 2.1$$

$$d[i,j] = min\{\, d[i,j], d[i-2,j-2] + cost \quad \dots 2.2 \,\}$$

Kemudian untuk mencari nilai jarak maksimum dari kata yang dianggap salah eja sebagai berikut.

$$jarak_max = round((\frac{40}{100}) \times M)$$

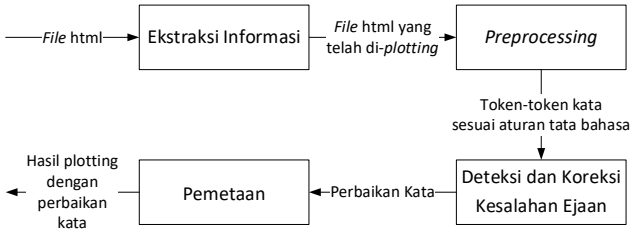
2.7 Kamus

Pada penelitian ini kamus bahasa Indonesia yang digunakan adalah hasil dari penelitian Rizki Hilmi Fauzi [15] karena cukup memuat banyak kata. Total kata yang terdapat pada kamus tersebut mencapai 52,978 kata sedangkan untuk kamus bahasa Inggris menggunakan kamus dari *Cambridge* [16].

Kamus ini akan digunakan untuk proses pengecekan kata, apabila kata masukan terdapat pada kamus maka kata tersebut dianggap benar dan apabila tidak maka kata tersebut termasuk kedalam kesalahan ejaan *non-word* atau *word-boundary*.

2.8 Skema Sistem

Penlitian-penelitian sebelumnya memanfaatkan metode pengoreksi ejaan berdasarkan kemiripan dalam pengucapan ataupun kemiripan dalam penulisan. Pada penelitian ini metode yang digunakan dalam proses deteksi dan koreksi kesalahan ejaan adalah metode *soundex* dan *damerau-levensthein distance*. Keseluruhan proses dibagi menjadi empat bagian utama, yaitu ekstraksi informasi, *preprocessing*, deteksi dan koreksi kesalahan ejaan, dan pemetaan.



Gambar II. 1 Tahapan Sistem

a. Ekstraksi Informasi

Tahapan ekstraksi informasi dibagi menjadi dua tahap, yaitu :

- Penghapusan karakter entitas html, yaitu akan dilakukan proses penghapusan karakter entitas html yang merupakan tahapan untuk menghapus karakter entitas spasi yaitu * * yang terdapat pada file html..
- *Plotting* digunakan untuk mendapatkan data terstruktur dari dokumen masukan yang tidak terstruktur sehingga data yang didapatkan sesuai dengan kebutuhan.

b. *Preprocessing*

Tahapan *preprocessing* merupakan tahapan untuk mempersiapkan data masukan hasil dari ekstraksi informasi yang akan diolah pada tahap pengecekan dan pengoreksian kesalahan ejaan. *Preprocessing* pada penelitian ini terdiri dari beberapa tahapan, yaitu:

- Penghapusan tag html, yaitu tahap penghapusan seluruh *tag-tag* html pada hasil *plotting* seperti *tag* <*p*>, <*strong*>, <*em*>, </*p*>, </*strong*>, dan tag </*em*>.
- Pengubahan karakter berdasarkan *list* perubahan, Pengubahan karakter yang dimaksud adalah pengubahan yang hanya terjadi pada suatu huruf sehingga menyebabkan suatu kata tidak sesuai dengan kaidah ejaan yang dibenarkan. Perubahan karakter huruf yang terjadi yaitu perubahan dari satu karakter menjadi karakter lain seperti menjadi angka, tanda baca, dan simbol bahkan menjadi karakter huruf lain.
- *Scanner* adalah proses utnuk mengelompokkan teks ke dalam token-token baik merupakan karakter huruf, angka, maupun simbol. Proses ini dilakukan untuk menyimpan bentuk karakter teks asli yang akan digunakan kembali pada saat proses pemetaan untuk perbaikan teks dengan kata-kata yang sudah diperbaiki berdasarkan kesalahan yang terjadi sekaligus akan digunakan untuk proses selanjutnya seperti *filtering*, *case folding*, *named entity tagger*, dan pengubahan karakter berdasarkan aturan tata bahasa serta proses deteksi dan koreksi kesalahan ejaan.
- *Filtering*, yaitu tahap pengahapusan seluruh tanda baca kecuali tanda baca titik (.) karena akan digunakan pada tahap *Named Entity Tagger* untuk mendeteksi *tag person* sehingga pada saat pengecekan kesalahan kata nama manusia tidak akan dianggap salah eja.
- *Case folding*, yaitu tahap untuk menyeragamkan seluruh karakter huruf menjadi huruf kecil. Hal ini dilakukan karena pada tahap pengubahan karakter berdasarkan aturan tata bahasa menggunakan huruf kecil sebagai aturannya.
- NE *tagger* (*Named Entity*), yaitu tahap untuk memberikan label pada suatu string apakah *string* tersebut merupakan nama manusia atau tidak. Jika *string* tersebut nama manusia maka akan diberikan label *person* sehingga *string* tersebut tidak akan dibandingkan dengan kamus untuk mengetahui kesalahan katanya. Deteksi *Named Entity* menggunakan *library* NER (*Named Entity Recognition*) hasil penelitian dari yusuf syaifudin.
- Pengubahan karakter berdasarkan aturan tata bahasa, yaitu tahap memeriksa seluruh token apakah terdapat kesalahan fonem konsonan atau tidak. Jika terdapat kesalahan fonem maka akan diperbaiki sesuai dengan aturan fonem kosonan yang ada di bahasa Indonesia. Berikut adalah aturan kesalahan fonem kosonan pada penelitian ini.

Tabel II. 2 Kesalahan Fonem

Kesalahan Fonem	Contoh	Hasil Perbaikan
/rn/	rnakan	makan
	rnasalah	masalah
	kearnanan	keamanan
	dalarn	dalam
/ch/	olch	oleh
/dm/	dmu	ilmu
/nn/	detenninasi	determinasi
/cn/	elemcn	elemen
/dc/	ordc	orde
	dcsa	desa
/lc/	lcbih	lebih
/sc/	sclesai	selesai
/cj/	cjata	data

c. Deteksi Dan Koreksi Kesalahan Ejaan

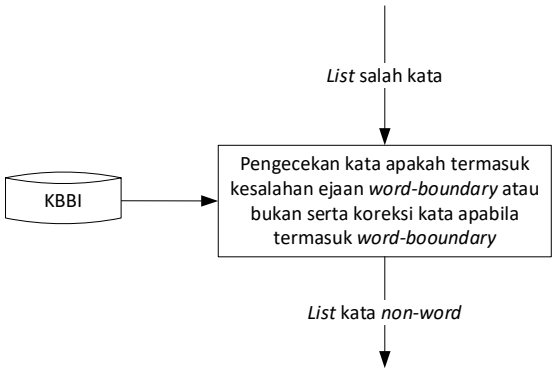
Tahapan ini terdiri dari beberapa tahap untuk proses deteksi dan koreksi kesalahan ejaan, yaitu:

- Penentuan Kesalahan

  Penentuan kesalahan kata yaitu dengan cara membandingkan setiap kata dengan *list* kata yang terdapat di dalam KBBI, kamus akronim dan kamus *cambridge*. Kata yang tidak terdeteksi di KBBI, kamus akronim dan kamus *cambridge* dianggap kata yang salah eja.

- Penentuan Dan Koreksi Kesalahan Ejaan *Word-Boundary*

Setelah proses penentuan kesalahan kata maka *list* kata yang salah kemudian akan dicek apakah dari *list* kata tersebut terdapat kesalahan ejaan *word-boundary*. Penentuan *word-boundary* adalah pengecekan kata dengan *list* kata yang terdapat di dalam kamus KBBI berdasarkan urutan huruf dalam setiap token kata, apabila salah satu kata ataupun semua kata yang dicek terdeteksi KBBI maka kata tersebut termasuk kesalahan *word-boundary*.

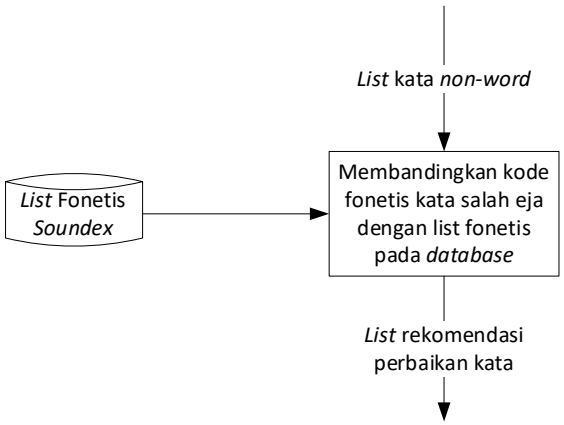


Gambar II. 2 Penentuan Kesalahan dan Koreksi *Word-Boundary*

Berikut ini adalah alur deteksi dan koreksi kesalahan ejaan *word-boundary*.

- Penentuan Rekomendasi Kesalahan Ejaan *Non-Word*

Proses penentuan rekomendasi kata menggunakan metode *soundex*. Token-token yang tidak terdeteksi pada saat pengecekan *word-boundary* akan dianggap kesalahan ejaan *non-word*. Token-token tersebut akan diubah menjadi kode fonetis dengan aturan *phonetic code soundex*. Pemberian rekomendasi berdasarkan kemiripan kode fonetis antara kata yang dianggap salah eja *non-word* dengan *list* kode fonetis *soundex* pada kamus. Berikut ini adalah proses rekomendasi kata seperti pada gambar III.5 berikut.



Gambar III. 3 Proses Penentuan Rekomendasi Perbaikan Kata Kesalahan *Non-Word*

Sebagai contoh kata yang salah eja adalah kata "disertasl". Kemudian diubah menjadi kode fonetis *soundex* yaitu menjadi d263. Berikut ini adalah rekomendasi untuk kata "disertasl".

Tabel II. 3 Rekomendasi Kata

Kata salah eja	Kode Fonetis	Rekomendasi
disertasl	d263	degradasi
		dekoratif
		dekorator
		dekret
		desertir
		diakritik
		disertasi
		diskredit
		disertai

- Penentuan Pemilihan Kata Rekomendasi Kesalahan Ejaan *Non-Word*

Tahap ini dilakukan proses pemilihan untuk koreksi kata dari kata-kata yang direkomendasikan. Pemilihan kata untuk koreksi menggunakan metode *damerau-levenshtein distance* yaitu dengan mencari nilai jarak edit paling kecil dari kata-kata yang direkomendasikan dan nilai jarak editnya kurang dari sama dengan nilai jarak maksimum dari kata yang dianggap salah eja tersebut. Berikut ini adalah algoritma *damerau-levenshtein*.

1. Hitung jarak maksimum dari kata yang dianggap salah eja *non-word*. Diasumsikan kata yang salah eja adalah "disertasl". Maka, M = 9

$$jarak_max = round((\frac{40}{100}) \times M)$$

$$jarak_max = round((\frac{40}{100}) \times 9)$$

$$jarak_max = 4$$

2. Kemudian inisialisasi s = "disertasi" sepanjang 9 karakter dan t = "disertasl" sepanjang 9 karakter. Buat matriks d dengan baris 0..9 dan kolom dari 0..9 sebagai berikut.
3. Selanjutnya bandingkan setiap karakter pada matriks d. Dimulai dari sel d[1,1], bandingkan karakter "d" pada baris dengan karakter "d" pada kolom. Pertama tentukan nilai *cost*. Karena "d" = "d", maka *cost* = 0. Lalu hitung nilai minimumnya. Maka didapat

$$d[i,j] = min\begin{cases} d[i-1,j]+1 \\ d[i,j-1]+1 \\ d[i-1,j-1]+cost \end{cases}$$

$$d[1,1] = min\begin{cases} d[1-1,1]+1 = d[0,1]+1 = 1+1 = 2 \\ d[1,1-1]+1 = d[1,0]+1 = 1+1 = 2 \\ d[1-1,1-1]+cost = d[0,0]+0 = 0+0 = \mathbf{0} \end{cases}$$

$$d[1,1] = \mathbf{0}$$

Maka, matriks untuk d[1,1] bernilai 0.

4. Selanjutnya untuk sel d[1,2], bandingkan karakter "i" dengan "i". karena "i" = "i", maka *cost* = 0. Lalu hitung nilai minimumnya, didapat

$$d[i,j] = min\begin{cases} d[i-1,j]+1 \\ d[i,j-1]+1 \\ d[i-1,j-1]+cost \end{cases}$$

$$d[1,2] = min\begin{cases} d[1-1,2]+1 = d[0,2]+1 = 2+1 = 3 \\ d[1,2-1]+1 = d[1,1]+1 = 0+1 = \mathbf{1} \\ d[1-1,2-1]+cost = d[0,1]+0 = 1+0 = \mathbf{1} \end{cases}$$

$$d[1,2] = \mathbf{1}$$

Maka, matriks untuk d[1,2] bernilai 1.

5. Lakukan untuk setiap sel matriks seperti pada langkah kedua dan ketiga sehingga didapat hasil penghitungan untuk seluruh sel matriks.

Hasil dari penghitungan dapat dilihat pada kolom terakhir baris terakhir matriks (pojok kanan baris terakhir). Dari hasil penghitungan, didapat bahwa nilai jarak edit menggunakan metode *Damerau-Levenshtein distance* dari perbandingan kata "disertasl" dan "disertasi" adalah 1. Pada tabel II.6 ditunjukkan perbandingan jarak edit yang dihasilkan dari seluruh kata yang direkomendasikan untuk kata yang dianggap salah eja, yaitu kata "disertasl".

Tabel II. 4 Nilai Jarak Edit Untuk Pemilihan Kata disertasl

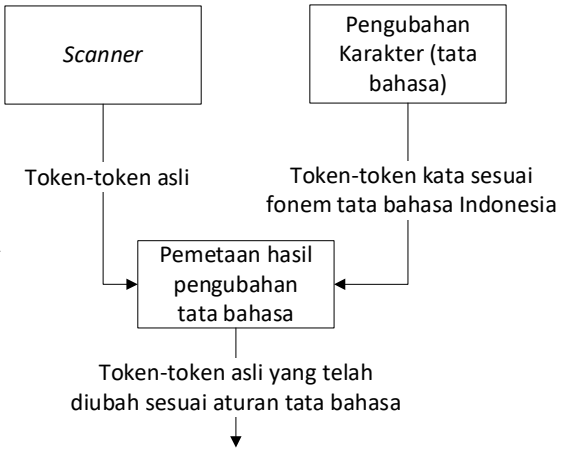
Rekomendasi Kata	Jarak Edit
degradasi	6
dekoratif	7
dekorator	7
dekret	7
desertir	**4**
diakritik	6
disertasi	**1**
diskredit	5
disertai	**2**

Berdasarkan pada tabel III.6 terdapat tiga kata yang nilai jarak editnya kurang dari atau sama dengan dari nilai jarak maksimum. Dari ketiga kata tersebut akan dipilih yang mempunyai nilai jarak edit paling kecil yaitu kata "disertasi" dengan jarak edit sama dengan satu

d. Pemetaan

Proses pemetaan adalah membandingkan antara token-token asli dengan token-token yang sudah diperbaiki dari kesalahan-kesalahan diantaranya proses pengubahan karakter berdasarkan aturan tata bahasa Indonesia, kesalahan *word-boundary*, dan kesalahan *non-word* berdasarkan *index* dari kedua token tersebut.

Pemetaan tahap pertama yang dilakukan adalah sesudah proses pengubahan karakter berdasarkan aturan tata bahasa Indonesia seperti pada gambar berikut.



Gambar II. 4 Pemetaan Hasil Pengubahan Karakter Tata Bahasa Indonesia

Pemetaan tahap kedua yaitu dilakukan sesudah proses pendeteksian dan pengoreksian kesalahan ejaan *word-boundary*.

Pemetaan tahap ketiga dilakukan setelah proses pemilihan kata untuk koreksi kesalahan ejaan *non-word*.

2.9 Pengujian Sistem

Pengujian dilakukan dengan sebelas dokumen disertasi, berikut ini adalah hasil keseluruhan pengujian untuk akurasi deteksi dan perbaikan kata setiap dokumennya.

Tabel II. 5 Pengujian Akurasi Deteksi

File	Deteksi	
	Recall	***Precision***
D 100 a.html	100%	20%
D 101 a.html	100%	50%
D 102 a.html	100%	25%
D 00901a.html	100%	54%
D 00903.html	100%	75%
D 00907.html	100%	50%
D 501.html	100%	76%
D 502.html	100%	76%
D 503.html	100%	80%
D 504.html	100%	78%
D 505.html	100%	71%
Rata-rata	100%	59,5%

Tabel II. 6 Pengujian Akurasi Perbaikan

File	Perbaikan	
	Recall	***Precision***
D 100 a.html	100%	100%
D 101 a.html	100%	50%
D 102 a.html	100%	100%
D 00901a.html	100%	100%
D 00903.html	100%	100%
D 00907.html	100%	100%
D 501.html	100%	38%
D 502.html	100%	53%
D 503.html	100%	63%
D 504.html	100%	10%
D 505.html	100%	47%
Rata-rata	100%	69,18%

3. PENUTUP

3.1 Kesimpulan

Pengujian sistem yang dilakukan menunjukkan bahwa aplikasi yang yang dibuat untuk mendeteksi dan mengkoreksi kesalahan ejaan sudah memenuhi persyaratan fungsional. Dari tabel 4.8 analisis hasil pengujian deteksi kesalahan ejaan terdapat beberapa penyebab mengapa rata-rata nilai *precision* deteksi yang dihasilkan kecil sebesar 59,5%, yaitu:

1. Karena nama orang yang tidak terdeteksi pada saat proses *Name Entity Tagger*.
2. Kata yang berakhiran "nya", "kan".
3. Kesalahan pada kata dalam bahasa inggris.

Sedangkan untuk pengujian koreksi dapat dilihat pada tabel II.9 bahwa nilai *precision* untuk koreksi sangat kecil yaitu dengan rata-rata 69,18%. Hal ini dikarenakan beberapa penyebab yaitu:

1. Kata yang direkomendasikan oleh *soundex* tidak sesuai dengan kata yang salah eja. Hal ini disebabkan jika huruf pertama yang direkomendasikan oleh *soundex* berbeda dengan kata yang salah eja.
2. Penggantian karakter huruf vokal menjadi huruf konsonan di tiga digit karakter huruf yang akan dikodekan pada saat proses konversi *file*.
3. Kurangnya sumber daya kamus bahasa Indonesia untuk kata yang memiliki imbuhan di awal, tengah dan akhir kata.

3.2 Saran

Untuk penelitian selanjutnya, disarankan untuk mencoba metode *phonetic* yang lain. Juga disarankan untuk menambahkan proses *steamming* agar kata-kata yang memiliki imbuhan baik di awal, tengah maupun akhir kata dijadikan kata dasar jika belum terdapat sumber daya kamus bahasa Indonesia yang memadai. Serta penggunaan library deteksi nama orang yang lebih baik (*Named Entity Recoginition*) untuk bahasa Indonesia agar tidak dianggap kata yang salah eja.

DAFTAR PUSTAKA

[1] Amber Wilcox-O'Hearn, Graeme Hirst, and A. Budanitsky, "Real-word spelling correction with trigrams: A reconsideration of the Mays, Damerau, and Mercer model," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4919 LNCS, no. 2000, pp. 605–616, 2008.

[2] Mohammad Ali Elmi and Martha Evens, "Spelling correction using context," *Proc. ThirtySixth Annu. Meet. Assoc. Comput. Linguist. Seventeenth Int. Conf. Comput. Linguist.*, vol. 1, pp. 360–364, 1998.

[3] Behrang Qasemizadeh, Ali Ilkhani, and Amir Ganjeii, "Adaptive language independent spell cheeking using intelligent traverse on a tree," *2006 IEEE Conf. Cybern. Intell. Syst.*, 2006.

[4] Yinghao Huang and Yi Li Murphey, "Automotive diagnosis typo correction using domain Knowledge and Machine Learning," pp. 273–280, 2013.

[5] Utis Sutisna, "Koreksi Ejaan Query Bahasa Indonesia Menggunakan Algoritme Damerau Levenshtein," vol. 15, no. 2, pp. 25–29, 2010.

[6] Arumsari. K, "Penggunaan Metode Kesamaan String pada Pemeriksaan Ejaan Bahasa Indonesia," 1998.

[7] Fahmilu Kurniawan, "Implementasi Algoritme Fonetik Untuk Koreksi Ejaan Ejaan Bahasa Indonesia," 2010.

[8] Aminul Islam and Diana Inkpen, "Real-Word Spelling Correction using Google Web 1T 3-grams," *EMNLP '09 Proc. 2009 Conf. Empir. Methods Nat. Lang. Process. Vol. 3*, no. August, pp. 1241–1249, 2009.

[9] Andrew Carlson and Ian Fette, "Memory-based context-sensitive spelling correction at web scale," *Proc. - 6th Int. Conf. Mach. Learn. Appl.*

ICMLA 2007, pp. 166–171, 2007.
[10] A. Insanudin, “Ekstraksi Informasi Kemacetan pada Media Digital,” 2013.
[11] Abdul. Chaer, *Tata Bahasa Praktis Bahasa Indonesia*, Revisi, Ce. Jakarta: Rineka Cipta, 2011.
[12] Robert C. Russell, “Russell Soundex System,” 1918.
[13] Fred J. Damerau, “A technique for computer detection and correction of spelling errors,” *Commun. ACM*, vol. 7, no. 3, pp. 171–176, 1964.
[14] Ulrich Pfeifer and L. I. VI, “Searching Proper Names in Databases,” *October*, vol. 20, pp. 1–13, 1994.
[15] Fauzi, Rizki Hilmi ‘Pembangunan Kamus Bahasa Indonesia sebagai Sumber Daya Natural Language Processing (NLP) Bahasa Indonesia,’ vol. 2017, pp. 193–200.
[16] “Cambridge Dictionary.” [Online]. Available: https://dictionary.cambridge.org/. [Accessed: 20-Jan-2018].