PENGEMBANGAN ALGORITMA *SOUNDEX* PADA *SPELL CHECKER* BAHASA INDONESIA

Ika Purwanti Ningrum[1], Muh. Yamin[2], Samsul[3]

(1) Jurusan Teknik Informatika, Fakultas Teknik, UHO,
(Contact : 081328806820, ika.purwanti.n@gmail.com)
(2) Jurusan Teknik Informatika, Fakultas Teknik, UHO,
(Contact : 085292227887, putra0683@yahoo.com)
(3) Jurusan Teknik Informatika, Fakultas Teknik, UHO,
(Contact : 081943273914, e1e110091@gmail.com)

Abstrak

Spell checker merupakan salah satu fitur dalam aplikasi pengolah kata yang memberikan pemeriksaan ejaan pada sebuah dokumen serta memberikan kata saran untuk kata yang salah dalam penulisan ejaannya. *Spelling checker* umumnya diimplementasi berdasarkan bahasa Inggris, sedangkan *spelling checker* bahasa Indonesia masih belum umum digunakan. Penelitian ini dibuat sebuah aplikasi *spell checker* bahasa Indonesia, yaitu mengembangkan algoritma *Soundex* hasil dari penelitian yang pernah dilakukan oleh [6]. Pada penelitian tersebut, pengelompokan konsonan dilakukan berdasarkan aturan pemberian kode fonetis perhuruf pada algoritma *Soundex*, berjumlah 7 (0-6) kode. Namun pengelompokan konsonan yang dilakukan belum mendapatkan hasil yang maksimal untuk kata saran yang dihasilkan, terutama untuk kata dengan awalan yang sama seperti kata dengan imbuhan "mem" dan "ber". Dalam penelitian ini, pengembangan algoritma *Soundex* dilakukan dengan membagi huruf-huruf pada kode 0 algoritma awal ke dalam dua kode yaitu kode 0 dan kode 8, serta kode 6 algoritma awal dibagi menjadi kode 6 dan kode 7. Hasil yang diperoleh, kata saran yang ditampilkan untuk kata dengan imbuhan "mem" dan "ber" jumlahnya lebih sedikit daripada menggunakan algoritma awal.

Key word : *spell checker, algoritma Soundex.*

1. Pendahuluan

Bahasa menjadi faktor penting dalam penulisan sebuah dokumen. Jika dalam penulisan sebuah dokumen terdapat ejaan kata yang salah, maka kata tersebut akan memiliki arti yang berbeda atau tidak memiliki arti sama sekali. Kesalahan ejaan adalah jika kata yang dituliskan tidak ada atau tidak terdaftar pada kamus kata Bahasa Indonesia [5].

Spell checker merupakan salah satu fitur yang terdapat pada aplikasi pengolah kata. Fitur ini memberikan pemeriksaan ejaan pada sebuah dokumen yang diketik serta memberikan usulan kata-kata untuk kata yang salah dalam penulisannya. Salah satu aplikasi pengolah kata yang umum digunakan adalah *Microsoft Office Word*. Akan tetapi *spelling checker* diimplementasi berdasarkan bahasa Inggris, sedangkan *spelling checker* untuk bahasa Indonesia masih belum umum digunakan. Dalam penelitian ini dibuat sebuah aplikasi *spell checker* untuk bahasa Indonesia menggunakan algoritma *Soundex*. Penggunaan algoritma *Soundex* pada pembuatan aplikasi ini adalah pengembangan dari algoritma *Soundex* bahasa Indonesia yang merupakan penelitian yang pernah dilakukan oleh [6].

2. Metodologi

2.1 Algoritma *Soundex*

Algoritma *Soundex* merupakan algoritma yang mengevaluasi setiap huruf dari kata yang dimasukkan dan memberikan nilai numerik. Fungsi utama dari algoritma ini adalah mengubah setiap kata menjadi kode fonetik dengan panjang empat karakter [4]. Fonetik adalah ilmu yang menyelidiki bunyi bahasa tanpa melihat bunyi itu sebagai pembeda makna dalam suatu bahasa.sebuah *string* yang berbeda namun mempunyai cara pengucapan yang sama, akan memiliki kode fonetis yang sama [7]. Algoritma *Soundex* dibuat berdasarkan pengucapan dalam bahasa Inggris. Untuk mendukung pencocokan *string* berdasarkan bahasa Indonesia, pada tahun 1997 oleh Primasari, algoritma *Soundex* dikembangkan ke dalam bahasa Indonesia dengan mengganti pengelompokan konsonannya ke dalam faktor penyusun konsonan bahasa Indonesia [6].

Faktor-faktor pembentuk konsonan bahasa Indonesia adalah sebagai berikut [6]:

a. Faktor artikulator dan titik artikulasi.

b. Faktor jalan yang dilalui oleh suara.
c. Faktor jenis halangan yang dijumpai tatkala udara keluar.

Aturan pemberian kode fonetis per huruf pada algoritma *Soundex* (selanjutnya disebut : algoritma awal) untuk bahasa Indonesia dapat dilihat pada Tabel 1.

Tabel 1. *Aturan Pemberian Kode Fonetis pada Algoritma Soundex untuk Bahasa Indonesia [6]*

Huruf	Kode	Jenis Konsonan
A, I, U, E, O, H, W, Y	0	bunyi vokal
F, V	1	*labiodental*
S, X, Z	2	*frikatif*
L	3	*lateral*
R	4	*trill*
M, N	5	*nasal*
B, C, D, G, J, K, P, Q, T	6	*stop*

Namun pengelompokan konsonan yang dilakukan berdasarkan aturan pemberian kode fonetis diatas belum mendapatkan hasil yang maksimal untuk kata saran yang dihasilkan. Seperti diketahui bahwa dalam bahasa Indonesia, kata-kata yang memiliki awalan yang sama seperti kata yang memiliki imbuhan "mem" dan "ber" jumlahnya sangat banyak. Penggunaan klasifikasi konsonan pada algoritma Soundex diatas untuk kata-kata yang memiliki awalan yang sama akan memiliki kode fonetik yang sama pula. Hal ini jelas akan mempengaruhi jumlah kata saran yang akan dihasilkan.

2.2 Pembentukan Konsonan berdasarkan Cara Artikulasi dan Tempat Artikulasi

Berdasarkan cara artikulasi atau jenis halangan udara yang terjadi pada waktu udara keluar dari rongga ujaran, konsonan dapat dibedakan menjadi lima kelompok, yaitu [2] :

1. Konsonan hambat (*stop*), terdiri dari konsonan [p], [t], [c], [k], [b], [d], [j], [g], dan [q].
2. Konsonan geser (*frikatif*), terdiri dari konsonan [f], [v], [h], [s], [z], dan [x].
3. Konsonan likuida (*lateral*), terdiri dari konsonan [l].
4. Konsonan getar (*trill*), terdiri dari konsonan getar apikal [r] dan konsonan getar uvular [R].
5. Semi-vokal, terdiri dari [w] dan [y].

2.3 Pembentukan Konsonan berdasarkan Struktur

Berdasarkan strukturnya, yakni hubungan antara artikulator dan titik artikulasi, konsonan dalam bahasa Indonesia dapat dibedakan menjadi tujuh kelompok, yaitu [2] :

1. *Bilabial*, bunyi yang dihasilkan ialah [p], [b], [m], dan [w].
2. *Labiodental*, bunyi yang dihasilkan ialah [f] dan [v].
3. *Apiko-dental*, bunyi yang dihasilkan ialah [s], [z], [r], dan [l].
4. *Palatal* atau *Lamino-Palatal*, bunyi yang dihasilkan ialah [c], [j], dan [y].
5. *Velar* atau *Dorso-velar*, bunyi yang dihasilkan ialah [k], [g], [x], dan [h].
6. *Glottal* atau *Hamza*, misalnya bunyi yang memisahkan bunyi [a] yang pertama dan [a] yang kedua pada kata "saat".
7. *Laringal*, bunyi yang dihasilkan ialah [h].

2.3 Pembentukan Konsonan berdasarkan Bergetarnya Pita Suara

Berdasarkan posisi pita suara atau bergetar tidaknya pita suara, konsonan dapat dibedakan menjadi tiga kelompok, yaitu [2] :

1. Konsonan bersuara, konsonan yang dihasilkan ialah [m], [b], [v], [n], [d], [r], [j], [h], dan [g].
2. Konsonan tak bersuara, konsonan yang dihasilkan ialah [p], [t], [c], [k], [q], [b], [d], [j], [g], [f], [s], [x], [h], [r], [l], [w], dan [y].
3. Konsonan *nasal*, konsonan yang dihasilkan ialah [m], dan [n].

2.4 Metode Empiris

Metode empiris bekerja dengan cara memecah *string* menjadi beberapa kemungkinan kata kemudian dicocokkan ke dalam *database* [3]. Cara kerja metode ini adalah memecah kata mulai dari huruf awal sampai dengan huruf yang terakhir, dimana pada setiap pemecahan satu huruf, *string* yang dihasilkan akan dicocokkan ke *database*. Jika *string* tersebut terdaftar di *database* maka proses pemecahan huruf dihentikan dan kata saran ditampilan.

2.5 Tokenisasi

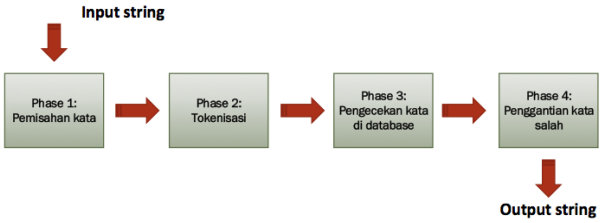
Tokenisasi atau pemisahan rangkaian kata adalah tugas memisahkan deretan kata di dalam kalimat, paragraf atau halaman menjadi token atau potongan kata tunggal atau *termmed word*. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca atau mengubah semua token ke bentuk huruf kecil (*lowercase*) [1].

2.2 Sistem *Spell Checker* Bahasa Indonesia

Penelitian tentang *spell checker* telah banyak dilakukan, diantaranya dilakukan oleh [5]. Seperti umumnya sistem pengecekan ejaan, *spell checker* mempunyai basis data yang berisi kata-kata yang mempunyai ejaan yang benar. Data tersebut yang akan dijadikan acuan untuk mengidentifikasi kata-kata yang salah. Dari kata yang salah, sistem akan menampilkan kata-kata saran berdasarkan kode fonetik yang sama dari kata yang salah. Pada penelitian ini, basis data yang digunakan diperoleh dari Kamus Besar Bahasa Indonesia (KBBI) luring/*offline* versi 1.5.

2.3 Metode yang Diusulkan

Blok diagram sistem yang dirancang untuk mengoreksi kata dengan ejaan yang salah disajikan pada Gambar 1.



Gambar 1. *Tahap Pemrosesan Sistem*

Pada tahap pemisahan kata dilakukan pemisahan kata dari kalimat yang dimasukkan. Kata-kata dipisahkan menjadi bentuk *array* agar kata bisa diproses lebih lanjut.

Pada tahap tokenisasi dilakukan penghilangan karakter-karakter tertentu seperti tanda baca pada sebuah kata [1]. Beberapa karakter yang dihilangkan dalam proses tokenisasi dapat dilihat pada tabel 3.

Tabel 3. *Karakter Tokenisasi*

No.	Simbol	Keterangan
1		Spasi
2	?	Tanda Tanya
3	!	Tanda seru
4	,	Koma
5	.	Titik
6	:	Titik dua
7	;	Titik koma
8	(	Buka kurung
9	)	Tutup kurung
10	-	Tanda kurang

Pada tahap pengecekan kata di *database* dilakukan pengecekan kata yang dimasukkan apakah ada di dalam *database* atau tidak. Jika kata ada di dalam *database* maka kata akan diabaikan dan terbaca sebagai kata yang benar. Jika kata tidak ada dalam *database*, maka kata akan diubah menjadi kode fonetik berdasarkan tabel fonetik. Selanjutnya sistem akan memberikan kata saran berdasarkan kode fonetik yang sama dengan kata yang salah.

Pada tahap penggantian kata salah dilakukan dengan mengganti kata salah dengan kata yang dipilih pada kata saran yang ditampilkan oleh sistem berdasarkan kode fonetik yang sama.

3. Pembahasan

3.1 Pengembangan Algoritma *Soundex*

Dalam pengembangan algoritma *Soundex*, dilakukan pengubahan pengelompokan konsonan untuk mendapatkan klasifikasi yang lebih baik daripada pengelompokan sebelumnya. Faktor-faktor yang digunakan dalam penentuan pengelompokan konsonan yang baru, yaitu [2]:

a. Cara artikulasi,
b. Struktur, dan
c. Bergetarnya pita suara.

Pada pengelompokan konsonan algoritma awal untuk kode 0 (A, I, U, E, O, H, W, Y), huruf-huruf yang masuk di dalamnya dikategorikan sebagai bunyi vokal. Berdasarkan hasil penelitian [2] tentang pembentukan konsonan berdasarkan cara artikulasi, huruf "W" dan "Y" merupakan konsonan semi vokal. Oleh sebab itu, kode 0 pada algoritma awal kami pecah menjadi dua kelompok yaitu kode 0 yang terdiri dari huruf A, I, U, E, O, H dengan jenis konsonan bunyi vokal; dan kode 8 yang terdiri huruf W dan Y dengan jenis konsonan semi vokal.

Untuk kode 6 pada algoritma awal, huruf-huruf yang masuk di dalamnya merupakan konsonan *stop* yaitu "B", "C", "D", "G", "J", "K", "P", "Q", "T", tetapi jumlah huruf yang ada di dalamnya cukup banyak sehingga dilakukan pembagian lagi menjadi 2 kelompok yaitu kode 6 yang terdiri dari huruf B, D, P, T dengan jenis konsonan *stop* dan kode 7 yang terdiri dari huruf C, G, J, K, Q dengan jenis konsonan *stop*.

Dari pengembangan yang dilakukan, didapatkan hasil pengelompokan konsonan yang baru sebanyak 9 klasifikasi yang dapat dilihat pada Tabel 2.

Tabel 2. *Klasifikasi Konsonan yang Dikembangkan*

Huruf	Kode	Jenis Konsonan
A, I, U, E, O, H	0	bunyi vokal
F, V	1	*labiodental*
S, X, Z	2	*frikatif*
L	3	*lateral*
R	4	*trill*
M, N	5	*nasal*
B, D, P, T	6	*stop*
C, G, J, K, Q	7	*stop*
W, Y	8	semi vokal

Adapun proses dari algoritma *Soundex* adalah sebagai berikut:

a. Memasukkan inputan *string* kata.
b. Menentukan panjang jumlah kode fonetik = 4.
c. Mempertahankan huruf pertama pada kata tersebut.
d. Mengubah huruf lainnya menjadi kode fonetis berdasarkan tabel fonetis.
e. Menulis empat posisi karakter pertama yang mengikuti pola:
<lowercase letter><digit><digit><digit>

Jika kode fonetis tidak sampai empat karakter, maka kode lainnya adalah 0.

3.2 Pengujian terhadap Kesalahan Ejaan Kata

Pengujian terhadap kesalahan ejaan kata dilakukan dengan cara memasukkan 6 kata berbeda (2 kata berimbuhan "mem", 2 kata berimbuhan "ber", dan 2 kata dasar) dan mengecek kata saran yang diberikan menggunakan algoritma *Soundex* bahasa Indonesia oleh [6] dan algoritma *Soundex* yang dikembangkan), kemudian membandingkan jumlah kata saran yang diberikan dari kedua algoritma tersebut. Parameter keberhasilan pada pengujian ini adalah berkurang atau samanya kata saran yang tampil tanpa menghilangkan kata saran yang diinginkan. Jumlah kata yang digunakan pada *database* sebanyak 2000 kata.

Pada Tabel 4 menunjukkan perbandingan kinerja algoritma *Soundex* oleh [6] dan algoritma *Soundex* hasil pengembangan terhadap 6 kata yang mempunyai kesalahan ejaan.

Tabel 4. *Perbandingan Kinerja Algoritma Soundex Oleh [6] Dan Algoritma Soundex Hasil Pengembangan*

No.	Kata Salah	Kata yang Diinginkan	Kata Saran dengan Algortima Awal	Kata Saran dengan Algoritma yang Dikembangkan	Kesimpulan
1	Memperbaiti	Memperbaiki	Membayar Membayari Membayarkan Memberi Memberikan **Memperbaiki** Memperoleh Memutar Menabur Menaburi Menaburkan Mencari Mencarikan Menggertak Menggoreng Mengirim Mengirimi Mengirimkan Mengurus Mengurusi Menyabarkan Menyebar Menyebarkan Menyederhanakan Menyegarkan Menyetrika Menyopir Menyopiri	Memberi Memberikan **Memperbaiki** Memperoleh Memutar Menabur Menaburi Menaburkan	Berhasil
	Jumlah kata		**28 kata**	**8 kata**	
2	Menceri	Mencari	Membayar Membayari Membayarkan Memberi Memberikan Memperbaiki Memperoleh Memutar Menabur Menaburi Menaburkan **Mencari** Mencarikan Menggertak Menggoreng Mengirim	**Mencari** Mencarikan Menggertak Menggoreng Mengirim Mengirimi Mengirimkan Mengurus Mengurusi	Berhasil

			Mengirimi Mengirimkan Mengurus Mengurusi Menyabarkan Menyebar Menyebarkan Menyederhanakan Menyegarkan Menyetrika Menyopir Menyopiri		
	Jumlah kata		**28 kata**	**9 kata**	
3	Beradap	Beradab	**Beradab** Beradaptasi Berakibat Berbagi Berbaik Berbaikan Berbatasan Berbataskan Berbatu Berbicara Berbuat Berbukti Bercabang Bercat Bercekcok Bercetak Berdagang Berdatangan Berdekat Berdekatan Berhadapan Beribadah Berikat Berjaga Berjatuh Berjatuhan Berjawab Berkacamata Berkeadilan Berkecepatan Berkedudukan Berkekurangan Berkepung Berkeyakinan Berkipas Berkokok Berpakaian Berpaku Berpecah Berpecahan Berpegang Berpegangan Berpidato Berpikir Berpikiran Bertabur	**Beradab** Beradaptasi Berbatasan Berbataskan Berbatu Berbuat Berdatangan Berhadapan Beribadah Berpidato Bertabur Bertaburan	Berhasil

			Bertaburan Bertahukan Bertekan Bertujuan Bertukar Bertukaran Berucap		
	Jumlah kata		**53 kata**	**12 kata**	
4	Berpakaiyan	Berpakaian	Beradab Beradaptasi Berakibat Berbagi Berbaik Berbaikan Berbatasan Berbataskan Berbatu Berbicara Berbuat Berbukti Bercabang Bercat Bercekcok Bercetak Berdagang Berdatangan Berdekat Berdekatan Berhadapan Beribadah Berikat Berjaga Berjatuh Berjatuhan Berjawab Berkacamata Berkeadilan Berkecepatan Berkedudukan Berkekurangan Berkepung Berkeyakinan Berkipas Berkokok **Berpakaian** Berpaku Berpecah Berpecahan Berpegang Berpegangan Berpidato Berpikir Berpikiran Bertabur Bertaburan Bertahukan Bertekan Bertujuan Bertukar	Berbagi Berbaik Berbaikan Berbicara Berbukti Berdagang Berdekat Berdekatan **Berpakaian** Berpaku Berpecah Berpecahan Berpegang Berpegangan Berpikir Berpikiran Bertahukan Bertekan Bertujuan Bertukar Bertukaran	

			Bertukaran Berucap		
	Jumlah kata		**53 kata**	**21 kata**	
5	Laper	Lapar	**Lapar** Lapor Lapur	**Lapar** Lapor Lapur	Berhasil
	Jumlah kata		**3 kata**	**3 kata**	
6	Rentam	Rentang	Rantang Ranting Rentan **Rentang** Runtuhan	Rantang Ranting Rentan **Rentang** Runtuhan	Berhasil
	Jumlah kata		**5 kata**	**5 kata**	

Berdasarkan Tabel 4, penggunaan algoritma *Soundex* bahasa Indonesia yang telah dikembangkan menunjukan kinerja yang lebih baik dari pada algoritma *Soundex* oleh [6] yaitu berkurang atau samanya kata saran yang tampil tanpa menghilangkan kata saran yang diinginkan.

4. Kesimpulan

1. Dengan mengimplementasikan algoritma *Soundex* pada aplikasi *spell checker* bahasa Indonesia, kesalahan ejaan kata dapat diatasi dengan hasil yang sangat baik.
2. Pengembangan algoritma yang dilakukan dengan menambah klasifikasi huruf dengan jumlah sebanyak 9 klasifikasi terbukti lebih baik daripada algoritma awal yang hanya memiliki 7 klasifikasi dengan hasil pemberian kata saran yang berkurang tanpa menghilangkan kata saran yang diinginkan.
3. Membandingkan algoritma *Soundex* dengan algoritma *phonetic string matching* yang lain seperti algoritma *Metaphone* atau algoritma *Caverphone* dalam kasus pengecekan ejaan kata.

Daftar Pustaka

[1] D.R. Pyriana, *Program Aplikasi Editor Kata Bahasa Indonesia Menggunakan Metode Approximate String Matching dengan Algoritma Levenshtein Distance berbasis Java*, Malang: Universitas Brawijaya, 2012.

[2] I.A. Rosmana, *Bahan Belajar Mandiri (BBM) 2: Cara Membentuk Fonem Bahasa Indonesia*, 2004.

[3] N.M. Andriyani, I.M. Santiyasa and A. Muliantara, *Implementasi Algoritma Levenshtein Distance dan Metode iEmpiris untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Dokumen Berbahasa Indonesia*,Bali: Universitas Udayana, 2012.

[4] P. David, V. Darnes, A. Yuridiana, G. Helena and L. Nahun, *The Soundex Phonetic Algorithm Revisited for SMS-based Information Retrieval **, Mexico: Benemerita Universidad Autonoma de Puebla, 2012.

[5] R.N., Dwitiyastuti, M. Adharul and A. Muhammad, *Pengoreksi Kesalahan Ejaan Bahasa Indonesia Menggunakan Metode Levenshtein Distance*, Malang: Universitas Brawijaya, 2013.

[6] S. Arifin, *Peranan Substitusi n-grams dan Code Shift Pada Algoritma Soundex*, Bogor: Institut Pertanian Bogor, 2006.

[7] T.A. Purnamasari, *Membangun Aplikasi Pencocokan String Berdasarkan Penulisan dan Kemiripan Pengucapan*, Yogyakarta: STMIK AMIKOM, 2012.