

Sistem *Stemming* Otomatis untuk Kata dalam Bahasa Indonesia

Rila Mandala, Erry Koryanti, Rinaldi Munir, Harlili

Laboratorium Ilmu dan Rekayasa Komputasi, Departemen Teknik Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung 40132

e-mail: {rila, rinaldi, harlili}@if.itb.ac.id

Abstract

Stemming is a process to restore words to its base form, by stripping each word from its derivational and affixes. A stemming process has an important role for machine-translation and other computational linguistics area. In Malaysian there is a stemming algorithm that has been developed and tested for application in information retrieval which is known as Othman algorithm. There are several differences of Bahasa Indonesia's morphology and Malay's morphology, so The Othman algorithm can not be applied directly in bahasa Indonesia. Furthermore, the accuracy of Othman algorithm also is not good. This paper proposes some modifications from Othman algorithm. The modifications includes, various stemming procedures, rule of affixes, and dictionary of root words. Experiments show that Our modification method has a better accuracy in stemming Bahasa Indonesia's words.

Keywords: *stemming, word-lemmatization, affix-stripping*

1. Pendahuluan

Algoritma *stemming* adalah prosedur komputasi yang mencari asal kata dari suatu kata dalam kalimat yang dilakukan dengan cara memisahkan masing-masing kata dari kata dasar dan imbuhan[10]. Sebagai contoh: kata *group, groups, grouped, grouping*, dan *subgroup* dihasilkan dari kata dasar *group*.

Algoritma *stemming* banyak dibutuhkan dan diaplikasikan dibidang sistem temu kembali informasi dan komputasi linguistik. Dalam sistem temu kembali informasi, algoritma *stemming* digunakan untuk mengurangi perbedaan bentuk dari suatu kata dengan mengembalikannya ke dalam bentuk kata dasar sehingga proses temu kembali menjadi efektif. Dalam bidang komputasi linguistik, algoritma *stemming* digunakan untuk mengidentifikasi kata dasar yang benar secara linguistik dan membantu analisis dari suatu kalimat.

Pada saat ini ada algoritma *stemming* untuk bahasa Melayu yang telah dikembangkan dan diuji yang dikenal dengan algoritma Othman[1]. Algoritma ini masih belum efektif jika diterapkan pada bahasa Indonesia. Oleh karena itu makalah ini mencoba untuk mengaplikasikan dan membuat beberapa perbaikan berupa penambahan penerapan aturan pemenggalan imbuhan dan perbaikan algoritma sebagai usaha untuk mengurangi kesalahan pemenggalan imbuhan dari suatu kata yang masih ditemui pada algoritma Othman, sehingga diperoleh algoritma yang dapat digunakan untuk teks bahasa Indonesia.

2. Stemming

Stemming adalah proses menghilangkan imbuhan dari suatu kata dari suatu dokumen atau *query*. Permasalahan utama dalam stemming adalah bagaimana mendapatkan kata dasar yang benar dari hasil pemenggalan imbuhan dari suatu kata jadian. Efektifitas algoritma stemming dipengaruhi oleh beberapa faktor:

- a. Kesalahan dalam proses pemenggalan imbuhan dari kata dasarnya. Kesalahan ini dapat berupa:
 - *Overstemming*: yaitu pemenggalan imbuhan yang melebihi dari yang seharusnya. Contoh: kata *masalah* menjadi *masa*. Kesalahan ini dapat timbul karena bentuk kata dasar yang menyerupai imbuhan.
 - *Understemming*: yaitu pemenggalan imbuhan yang terlalu sedikit dari yang seharusnya. Contoh: kata *belajar* menjadi *lajar*. Kesalahan ini dapat timbul karena kekurangan pada aturan pola imbuhan yang didefinisikan.
 - *Unchange*: yaitu kasus khusus dari *understemming*, dimana tidak terjadi pemenggalan imbuhan sama sekali. Contoh: kata *telapak*, setelah pemenggalan kata dasar yang didapat tetap *telapak*. Kesalahan ini dapat ditimbulkan karena kekurangan pada aturan pola imbuhan yang didefinisikan.
 - *Spelling exception*: yaitu huruf pertama kata dasar yang didapat tidak benar yang diakibatkan dari pemenggalan awalan. Contoh: kata *memukul* menjadi *ukul*. Kesalahan ini dapat ditimbulkan karena ada beberapa imbuhan yang berubah bentuk ketika ditempelkan pada suatu kata dasar. Misalnya awalan *beR-*, *meN-*, *teR-*, *peR-*, akan bergantung pada huruf pertama kata dasar dimana imbuhan tersebut ditempelkan (Contoh: *ber-* + *ajar* = *belajar*, *pen-* + *lihat* = *penglihatan*, *pen-* + *sakit* = *penyakit*). Atau sebaliknya ada imbuhan yang mengakibatkan huruf pertama kata dasar yang ditempelinya menjadi luluh. Misalnya *meng-* / *peng-* meluluhkan huruf 'k' (Contoh: mengarang dari *meng-* dan *karang*) atau *men-* / *pen-* meluluhkan huruf 'p' (Contoh: menuai dari *men-* dan *tuai*).
- b. Kekurangan dalam perumusan aturan penambahan imbuhan pada kata dasar. Hal ini dapat terjadi karena morfologi bahasa Indonesia yang kompleks, sehingga sangat sulit atau bahkan tidak mungkin untuk merumuskan aturan yang sempurna.
- c. Jumlah total aturan imbuhan yang didapat berhubungan dengan efektifitas proses temu kembali. Dimana semakin banyak pola penambahan imbuhan yang dapat dirumuskan, maka proses temu kembali akan semakin efektif.

3. Algoritma Stemming Othman

Algoritma Othman dikembangkan pada tahun 1993, menggunakan pendekatan berdasarkan aturan morfologi sederhana. Aturan yang digunakan algoritma Othman berisi 121 aturan imbuhan[1], yang mendefinisikan awalan, akhiran, sisipan dan pasangan awalan-akhiran dan ditulis sebagai berikut:

- a. Format aturan awalan: awalan +; contoh: ber+
- b. Format aturan akhiran: +akhiran; contoh: +kan
- c. Format aturan sisipan: +sisipan+; contoh: +el+
- d. Format aturan pasangan awalan-akhiran: awalan+akhiran; contoh: di + kan

Proses pemenggalan imbuhan dilakukan dengan cara mencocokkan imbuhan yang ada di dalam aturan dengan kata yang dimasukkan, dimulai dari aturan imbuhan yang terpanjang yang mungkin melekat pada kata masukan.

Setelah proses pemenggalan dilakukan, kata hasil penggal diperiksa terhadap kamus. Jika kata hasil penggal ada di dalam kamus, maka kata hasil penggal adalah kata dasar. Jika

tidak maka dicari lagi aturan yang lebih kecil yang masih sesuai sampai semua aturan imbuhan selesai dicocokkan.

Ada tiga karakteristik penting yang dicatat dari algoritma ini[1]. Pertama, algoritma dapat *overstem*, misalnya karena kamus tidak dicek sampai setelah aturan pertama diterapkan pada suatu kata dalam proses pemenggalan imbuhan dari suatu kata dasar. Sebagai contoh kata *masalah* akan *overstem* menjadi *masa*, jika *lah* dianggap sebagai akhiran. Karakteristik kedua adalah mode operasi yang sesuai dari algoritma bergantung pada penerapan dari aturan imbuhan. Morfologi bahasa yang kompleks menyebabkan kesulitan untuk menentukan penerapan aturan yang sempurna pada aplikasi agar memberikan kata dasar yang benar dari semua kemungkinan kata. Othman menerapkan penggunaan aturan untuk empat kelas imbuhan yang secara berturut-turut sebagai berikut: pertama aturan pasangan awalan-akhiran, aturan awalan, aturan akhiran, dan terakhir aturan sisipan. Karakteristik yang ketiga adalah jumlah total aturan yang digunakan. Othman menggunakan 121 aturan pembentukan imbuhan, tetapi algoritma ini masih belum bisa menangani semua kata dalam bahasa Malaysia.

4. Algoritma Stemming yang Diusulkan

Makalah ini mengajukan algoritma stemming dasar, dan selanjutnya mengusulkan empat macam algoritma yang dikembangkan dari algoritma Othman tersebut agar dapat diaplikasikan untuk bahasa Indonesia. Empat algoritma hasil modifikasi yang diusulkan penulis diberi nama *algoritma Stemming Versi1*, *algoritma Stemming Versi2*, *algoritma Stemming Versi3*, dan *algoritma Stemming Versi4*. Selain modifikasi terhadap algoritmanya juga dilakukan modifikasi terhadap koleksi aturan imbuhan dan kamus kata dasar yang dipakai di dalam algoritma stemming yang diusulkan. Sub bab berikut akan menjelaskan tentang modifikasi yang telah dilakukan, meliputi algoritma yang diusulkan, koleksi aturan imbuhan, dan koleksi data kamus kata dasar.

4.1 Algoritma Stemming Dasar

Operasi umum algoritma stemming dasar adalah sebagai berikut:

Langkah 1: Jika tidak ada lagi kata maka *stop*. Jika tidak ambil kata berikutnya.

Langkah 2: Jika tidak ada lagi aturan yang dapat diterima, maka kata dianggap sebagai kata dasar kemudian lompat ke *langkah1*. Jika tidak, ambil aturan berikutnya.

Langkah 3: Cek pola aturan yang diberikan dengan kata: jika sistem menemukan kecocokan, gunakan aturan pada kata untuk mendapatkan asal kata(*stem*)

Langkah 4: Cek asal kata terhadap kamus, buat beberapa catatan yang perlu dan cek kamus sekali lagi.

Langkah 5: Jika asal kata mendekati kamus, dan asal kata ini merupakan kata dasar, lompat ke *langkah1*, jika tidak lompat ke *langkah2*.

4.2 Algoritma Stemming_Versi 1

Algoritma Stemming_Versi1 merupakan algoritma stemming dasar dengan menambahkan pengecekan kamus pada langkah awal algoritma dan perubahan pada aturan pola imbuhan yang diterapkan, serta kamus yang dijadikan referensi adalah kamus kata dasar yang dikoleksi secara manual oleh penulis.

Operasi umum algoritma *Stemming_Versi1* adalah sebagai berikut:

Langkah 1: Selama belum *EOF* baca teks masukan;

Langkah 2: Jika tidak ada lagi kata lompat maka *Stop*; Jika masih ada kata ambil satu kata;

Langkah 3: Cocokkan kata yang dibaca dengan kata yang ada di dalam kamus; Jika kata yang dibaca ada di dalam kamus maka kata adalah kata dasar dan lompat ke *langkah 2*;

Langkah 4: Selama masih ada aturan pola imbuhan, bandingkan karakter kata yang dibaca dengan pola aturan imbuhan yang ada; Jika ada aturan imbuhan yang sesuai lakukan pemenggalan imbuhan dan lompat ke *langkah 3*; Jika tidak dan atau tidak ada lagi aturan imbuhan yang tersisa maka kirim kata asal sebagai kata hasil; Lompat ke *langkah 2*.

4.3 Algoritma Stemming_Versi2

Algoritma Stemming_Versi2 adalah algoritma yang mengikuti urutan langkah algoritma dasar dengan perubahan pada pengecekan terakhir, yaitu jika kata yang didapat dari hasil pemenggalan tidak terdapat di dalam kamus maka pemenggalan terhadap kata tersebut dibatalkan. Sehingga kata hasil penggal dikembalikan kedalam bentuk kata masukan. Pada algoritma ini aturan imbuhan dan kamus yang dipakai adalah aturan imbuhan dan kamus kata dasar yang dikoleksi oleh penulis.

Operasi umum algoritma *Stemming_Versi2* adalah sebagai berikut:

Langkah 1: Selama belum *EOF* baca teks masukan;

Langkah 2: Jika tidak ada lagi kata lompat maka *Stop*; Jika masih ada kata ambil satu kata;

Langkah 3: Selama masih ada aturan pola imbuhan, bandingkan karakter kata yang dibaca dengan pola aturan imbuhan yang ada; Jika ada aturan imbuhan yang sesuai lakukan pemenggalan imbuhan; Jika tidak dan atau tidak ada lagi aturan imbuhan yang tersisa maka kirim kata asal sebagai kata hasil; Lompat ke *langkah 2*.

Langkah 4: Cocokkan kata hasil pemenggalan dengan kata yang ada di dalam kamus; Jika kata hasil pemenggalan ada di dalam kamus maka kata adalah kata dasar dan lompat ke *langkah 2*; Jika tidak lompat ke *langkah 3*.

4.4 Algoritma Stemming_Versi3

Algoritma Stemming_Versi3, yaitu algoritma yang mengikuti urutan langkah algoritma stemming dasar dengan perubahan pada aturan imbuhan dan kamus yang digunakan, dimana aturan imbuhan dan kamus yang digunakan adalah aturan imbuhan dan kamus kata dasar yang dikoleksi oleh penulis.

Operasi umum algoritma *Stemming_Versi3* adalah sebagai berikut:

Langkah 1: Selama belum *EOF* baca teks masukan;

Langkah 2: Jika tidak ada lagi kata lompat maka *Stop*; Jika masih ada kata ambil satu kata;

Langkah 3: Selama masih ada aturan pola imbuhan, bandingkan karakter kata yang dibaca dengan pola aturan imbuhan yang ada; Jika ada aturan imbuhan yang sesuai lakukan pemenggalan imbuhan; Jika tidak dan atau tidak ada lagi aturan imbuhan yang tersisa maka kirim kata bentuk terakhir sebagai kata hasil; Lompat ke *langkah 2*.

Langkah 4: Cocokkan kata hasil pemenggalan dengan kata yang ada di dalam kamus; Jika kata hasil pemenggalan ada di dalam kamus maka kata adalah kata dasar dan lompat ke *langkah 2*; Jika tidak lompat ke *langkah 3*.

4.5 Algoritma Stemming_Versi4

Algoritma Stemming_Versi4, adalah algoritma yang dibuat dengan mengikuti urutan langkah algoritma stemming dasar yang orisinil dengan perubahan tanpa adanya pengecekan

terhadap kamus. Jadi algoritma ini tidak memakai koleksi kamus kata dasar. Aturan pola imbuhan yang dipakai adalah aturan pola imbuhan yang dikumpulkan penulis.

Operasi umum algoritma *Stemming_Versi4* adalah sebagai berikut:

Langkah 1: Selama belum *EOF* baca teks masukan;

Langkah 2: Jika tidak ada lagi kata lompat maka *Stop*; Jika masih ada kata ambil satu kata;

Langkah 3: Selama masih ada aturan pola imbuhan, bandingkan karakter kata yang dibaca dengan pola aturan imbuhan yang ada; Jika ada aturan imbuhan yang sesuai periksa apakah jumlah karakter kata yang dibaca – jumlah karakter aturan imbuhan ≥ 3 ; Jika kurang dari tiga karakter maka aturan tidak sesuai dan periksa aturan berikutnya; Jika jumlah karakter ≥ 3 maka lakukan pemenggalan imbuhan. Jika tidak ada lagi aturan imbuhan yang tersisa maka kirim kata bentuk terakhir sebagai kata hasil;

Langkah 4: Lompat ke *langkah 2*.

4.6 Kamus Kata Dasar

Kamus yang digunakan dalam algoritma yang diusulkan adalah kamus kumpulan kata dasar bahasa Indonesia yang diambil dari kamus elektronik bahasa Indonesia-Inggris dan kamus elektronik bahasa Indonesia-Belanda. Kumpulan kata dari kamus tersebut kemudian dipilih dan dimasukkan penulis secara manual untuk mengisi data kamus kata dasar.

Untuk metode yang menggunakan pengecekan terhadap kamus, jumlah kata dasar dalam kamus akan menentukan efektifitas pemenggalan imbuhan. Dimana semakin banyak kata dasar yang tersimpan didalam kamus kata dasar, maka akan semakin baik penentuan apakah suatu kata sudah merupakan kata dasar atau belum.

4.7 Aturan Imbuhan

Aturan pola imbuhan yang digunakan dalam makalah ini adalah aturan imbuhan yang dikumpulkan penulis dari beberapa buku tentang bahasa Indonesia, antara lain:

- Buku Tata Bahasa Baku Bahasa Indonesia[2]
- Pembentukan kata Bahasa Indonesia[5]
- Morfologi[9]
- Pelik-pelik Bahasa Indonesia[3]

Yang menghasilkan 547 variasi imbuhan bahasa Indonesia.

Aturan imbuhan ini digunakan untuk menentukan karakter yang akan dibuang dari kata masukan. Jumlah total aturan imbuhan yang didapat berhubungan dengan efektifitas proses pemenggalan imbuhan, semakin banyak pola penambahan imbuhan yang dapat dirumuskan maka proses pemenggalan imbuhan akan semakin baik.

5. Evaluasi

Evaluasi dilakukan dengan menjalankan empat macam algoritma *stemming* yang diusulkan penulis pada semua koleksi data uji yang terdiri dari:

- a. Data uji 01, berisi kumpulan kata unik dari lima buah abstrak penelitian ilmiah.
- b. Data uji 02, berisi kumpulan kata unik dari lima buah synopsis buku ilmiah.
- c. Data uji 03, berisi kumpulan kata unik dari lima macam artikel majalah *Tempo*, yang diambil dari kategori *berita utama*, *hukum*, *ekonomi*, *teknologi*, dan *kesehatan*.

Tujuan penggunaan beberapa koleksi data pengujian yang berbeda adalah untuk mendapatkan variasi bentuk kata yang beragam.

Kemudian hasil yang didapat dari masing-masing metoda dibandingkan dengan kata dasar yang didapat secara manual. Lalu dihitung jumlah kata hasil yang benar dan yang tidak benar dari masing-masing metoda tersebut.

Selain kata dasar yang dihasilkan sistem, juga dicatat waktu yang diperlukan sistem dalam proses penentuan kata dasar tersebut. Waktu proses yang dihitung diukur dalam satuan milidetik.

Kinerja algoritma *stemming* yang diusulkan diukur dengan cara membandingkan total hasil *stemming* yang benar terhadap jumlah total hasil *stemming* yang tidak benar. Semakin sedikit hasil *stemming* yang tidak benar maka semakin baik kinerja algoritma yang bersangkutan.

Hasil pengujian dapat dilihat pada tabel 1, 2, dan 3.

Tabel 1. Hasil *stemming* data uji 01

<i>Metoda</i>	<i>Kata Unik</i>	<i>Waktu Proses</i>	<i>BR</i>	<i>OV</i>	<i>UD</i>	<i>UC</i>	<i>SP</i>	<i>OT</i>
Versi 1	258	1317	253	2	-	-	-	3
Versi 2	258	1088	250	4	-	-	-	4
Versi 3	258	1088	200	52	-	-	1	5
Versi 4	258	93	193	49	1	-	9	2

Tabel 2. Hasil *stemming* data uji 02

<i>Metoda</i>	<i>Kata Unik</i>	<i>Waktu Proses</i>	<i>BR</i>	<i>OV</i>	<i>UD</i>	<i>UC</i>	<i>SP</i>	<i>OT</i>
Versi 1	205	1041	202	1	-	-	-	2
Versi 2	205	819	202	1	-	-	-	2
Versi 3	205	819	150	53	-	-	-	2
Versi 4	205	93	145	48	2	-	6	4

Tabel 3. Hasil *stemming* data uji 03

<i>Metoda</i>	<i>Kata Unik</i>	<i>Waktu Proses</i>	<i>BR</i>	<i>OV</i>	<i>UD</i>	<i>UC</i>	<i>SP</i>	<i>OT</i>
Versi 1	2513	2513	2503	3	-	2	3	2
Versi 2	2513	2520	2486	14	-	2	3	8
Versi 3	2513	2520	2347	153	1	-	5	7
Versi 4	2513	329	2312	168	5	-	24	4

Keterangan:

BR: Benar

OV: Overstemming

UD: Understemming

UC: Unchanged

SP: Spelling exception

OT: Others

Waktu Proses: Hitungan waktu dalam satuan milidetik.

Dari tabel hasil pengujian di atas terlihat bahwa algoritma Versi_1 menghasilkan jumlah kata dasar hasil *stemming* yang tidak benar paling sedikit dibanding dengan tiga algoritma lainnya. Tetapi algoritma Versi_1 ini membutuhkan waktu proses yang lebih lama dibanding tiga algoritma lainnya, kecuali pada kasus uji 03. Hal ini terjadi karena pada algoritma Versi_1 dilakukan pengecekan kamus terlebih dahulu sebelum melakukan pemenggalan, setelah dipastikan kata tersebut tidak terdapat di dalam kamus barulah metoda ini menyelidiki imbuhan, sehingga untuk data yang kata jadiannya lebih banyak dibanding kata dasarnya maka metoda ini memerlukan waktu proses yang lebih panjang dari pada metoda lainnya. Sebaliknya ketika jumlah kata dasar dari suatu data uji lebih banyak

dibandingkan dengan kata jadiannya maka metoda ini akan menghasilkan waktu proses yang lebih sedikit dibandingkan algoritma Versi_2 dan algoritma Versi_3, karena untuk kata masukan yang sudah dalam bentuk kata dasar metoda ini tidak akan melakukan proses pengecekan imbuhan dan pemenggalan imbuhan lagi tetapi langsung mendapatkan hasil *stemmingnya* segera setelah menemukan kata tersebut ada di dalam kamus. Pada algoritma Versi_1 ini masih terjadi kesalahan untuk kata dasar yang belum dimasukkan ke dalam kamus, dimana akan menghasilkan kata hasil yang *unchanged*. *Overstemming* juga masih terjadi pada algoritma ini, hal ini dapat terjadi jika kata jadian memiliki kata dasar yang karakter penyusunnya menyerupai kata dasar lain yang lebih kecil, contohnya: *sejatinya* menjadi *jati*, *apak* menjadi *pak*. Selain itu juga masih terdapat *Spelling exception*. Hal ini dapat terjadi karena kata jadian yang memiliki kata dasar yang ambigu, contohnya: *menyangkut* menjadi *angkut*, dan *menyeramkan* menjadi *eram*.

Metoda kedua yaitu algoritma Versi_2 memerlukan waktu yang relatif lebih sedikit dibanding algoritma Versi_1 terutama untuk data yang mengandung kata jadian yang lebih banyak dibanding dengan kata dasarnya. Algoritma ini masih memiliki kelemahan dalam penentuan kata dasar selain kesalahan seperti pada algoritma Versi_1 juga untuk kata yang sudah merupakan kata dasar yang karakter penyusunnya memiliki bentuk seperti kata dasar lain yang lebih kecil. Misalnya untuk kata *masalah* dihasilkan kata *masa*, *pemilu* menjadi *pilu*, *belanda* menjadi *anda*, *aroma* menjadi *roma*, dsb. Hal ini dikarenakan pada algoritma ini yang pertama dikerjakan adalah mencari aturan imbuhan yang sesuai, melakukan pemenggalan, baru kemudian memeriksa kamus. Sehingga untuk kata-kata *masalah*, *pemilu*, *belanda* dan *aroma*, akan tetap dilakukan pemenggalan karena kata *masa*, *pilu*, *anda* dan *roma* masing-masing merupakan kata dasar yang ada di dalam kamus walaupun bukan kata dasar dari bentuk kata sebelumnya.

Metoda ketiga yaitu algoritma Versi_3 memerlukan waktu yang relatif lebih sedikit dari algoritma Versi_1, tetapi masih menghasilkan kata dasar *overstemming* yang cukup banyak terutama untuk kata masukan yang sudah merupakan kata dasar, contohnya: *sepak* menjadi *pak*, *perilaku* menjadi *pilaku*, *akbar* menjadi *kbar*, *sukarnoputri* menjadi *sukarnoputr*, dan *kelompok* menjadi *kompok*. Hal ini terjadi karena pada metoda ini kamus diperiksa hanya setelah dilakukan proses pemenggalan, dan karena hasil yang diambil adalah hasil pemenggalan terakhir yang didapat dari aturan imbuhan yang masih sesuai dengan karakter pembentuk kata masukan, maka walaupun kata hasil penggal tidak terdapat di dalam kamus tetapi karena sudah tidak ada lagi aturan imbuhan yang masih dapat dibandingkan maka kata yang dihasilkan menjadi *overstemming*.

Metoda keempat yaitu algoritma Versi_4 membutuhkan waktu proses yang paling singkat, tetapi masih menghasilkan kata dasar hasil *stemming* yang tidak benar cukup banyak, terutama untuk kata dasar yang jumlah karakter pembentuknya melebihi tiga karakter. Hal ini terjadi karena pada algoritma tanpa kamus ini pemenggalan terhadap suatu kata dilakukan sesuai dengan pola imbuhan terpanjang yang sesuai dengan pola karakter pembentuk kata masukan yang jika dihilangkan dari kata tersebut menghasilkan kata yang jumlah karakter minimalnya adalah tiga. Dengan demikian untuk kata dasar yang lebih dari tiga karakter dan karakter penyusunnya memiliki pola seperti suatu imbuhan, maka sistem akan memenggal kata dasar tersebut, sehingga banyak terjadi *overstemming*. Disamping itu juga masih banyak terdapat *Spelling exception*, contohnya: kata *memasuki* menjadi *pasuk*, *mengamati* menjadi *kamat*, *mengubah* menjadi *kubah*, *peminat* menjadi *pinat*, *pengalaman* menjadi *kalam*. Hal ini terjadi karena imbuhan *mem-*, *pem-*, *peng-*, dan *meng-* dapat menyebabkan peluluhan huruf pertama kata yang ditempelinya. Karena algoritma Tanpa Kamus ini dibuat untuk langsung menambahkan huruf yang luluh pada kata hasil penggal, maka untuk kata yang tidak terjadi peluluhan akan mengakibatkan terjadinya *Spelling exception*.

6. Kesimpulan

Kesimpulan terhadap keseluruhan penelitian yang dikerjakan pada makalah ini adalah sebagai berikut:

- a. Algoritma yang diusulkan dapat diaplikasikan dalam bahasa Indonesia dengan memodifikasi aturan imbuhan dan kamus yang dipakai sesuai dengan morfologi bahasa Indonesia.
- b. Efektifitas penentuan kata dasar dipengaruhi oleh jumlah aturan yang dikenali sistem dan jumlah kata dasar yang ada di dalam kamus. Semakin banyak aturan imbuhan yang dikenali sistem dan semakin banyak kata dasar yang dimasukkan ke dalam kamus akan semakin baik hasil yang didapat.
- c. Dari empat algoritma yang diterapkan, algoritma Othman_Versi1 memberikan hasil yang terbaik karena menghasilkan kata yang tidak benar paling sedikit dibandingkan dengan ketiga algoritma lainnya

7. Ucapan Terima Kasih

Penelitian yang dilakukan oleh penulis di Laboratorium Ilmu dan Rekayasa Komputasi, Departemen Teknik Informatika ITB ini sebagian didanai oleh dana RUT (Riset Unggulan Terpadu) IX dari KMNRT (Kantor Menteri Negara Riset dan Teknologi), oleh karena itu penulis mengucapkan terima kasih.

Daftar Pustaka

- [1] Ahmad, Fatimah, Yusoff, M., & Sembok M.T.,Tengku, *Experiments with a Stemming Algorithm for Malay Words*, Journal of the American Society for Information Science, 47,909-918,1996.
- [2] Alwi, Hasan, *Tata Bahasa Baku Bahasa Indonesia*, Balai Pustaka, 1998.
- [3] Badudu, J.S., *Pelik-Pelik Bahasa Indonesia*, Nawaputra, 2001.
- [4] Depdikbud, *Pedoman Umum EYD & Pedoman Umum Pembentukan Istilah*, Pusat pembinaan dan pengembangan bahasa..
- [5] Kridalaksana, Harimurti, *Pembentukan Kata dalam Bahasa Indonesia*, Gramedia Pustaka Utama, 1996.
- [6] Poerwadarminta, W.Js., *Kamus Umum Bahasa Indonesia*, Balai Pustaka, 1981.
- [7] Ramlan, M., *Morfologi Suatu Tinjauan Deskriptif*, CV. Karyo, 1987.
- [8] <http://snowball.tartarus.org/lovins/stemmer.html>, *The Lovins Stemming Algorithm*, diakses tanggal 29 Desember 2002.