Penerapan Algoritma Paice atau Husk untuk Stemming pada Kamus Bahasa Inggris ke Bahasa Indonesia

Damar Aji Asmara, Dewi Khairani, Siti Ummi Masruroh
Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Syarif Hidayatullah
Jakarta, Indonesia
damarajiasmar@gmail.com, dewi.khairani@uinjkt.ac.id, ummi.masruroh@uinjkt.ac.id

***Abstrak*—Untuk menerapkan algoritma *stemming*, penulis menggunakan dua algoritma sebagai perbandingan dari performasi masing-masing algoritma. Dari algoritma *stemming* yang terkenal, yaitu algoritma Porter *stemmer*, Paice/Husk *stemmer*, Lovins *stemmer*, Dawson *stemmer*, dan Krovetz *stemmer*, penulis tertarik pada algoritma Paice/Husk dan algoritma Porter. Penulis tertarik pada algoritma Paice/Husk karena memiliki pola *stemming* yang unik yaitu bersifat iteratif, sehingga algoritma ini dikenal dengan algoritma yang sangat kuat dan agresif. Sebagai perbandingan, penulis juga melakukan penelitian pada algoritma Porter yang memiliki proses *stemming* yang bersifat linear. Penelitian yang penulis lakukan adalah dengan cara menerapkan algoritma Paice/Husk dan algoritma Porter pada kamus bahasa Inggris-Indonesia berbasis *web*. Hasil dari penerapan algoritma Paice/Husk dan Porter tersebut diuji untuk dianalisa hasil penelusuran kata dasarnya. Hasil pengujian algoritma Paice/Husk ini telah dianalisa bahwa 98.3% masukkan menghasilkan keluaran yang sesuai dengan harapan, sedangkan algoritma Porter hanya sebesar 55.6%. Dengan demikian, banyak manfaat yang bisa diambil dari algoritma *stemming* khususnya Paice/Husk untuk *stemming* dan penelusuran kata dasar pada kamus.**

Kata Kunci*—*Algorithm; Paice/Husk; Stemming; Language

I. PENDAHULUAN

"Penerapan Algoritma Paice/Husk Untuk Stemming Pada Kamus Bahasa Inggris ke Bahasa Indonesia" diharapkan bermanfaat bagi pengguna kamus bahasa Inggris-Indonesia untuk memperoleh informasi kata dasar pada bahasa Inggris dengan akurat. Sehingga, kita dapat mengenali perbedaan terjemahan antara bentuk yang satu dengan yang lainnya khususnya antara kata berimbuhan dengan kata dasarnya.

Terdapat beberapa macam algoritma stemming yang terkenal, diantaranya adalah algoritma Porter stemmer, Paice/Husk stemmer, Lovins stemmer, Dawson stemmer, dan Krovetz stemmer (Hooper & Paice, 2005). Dari kelima Algoritma-algoritma stemming tersebut, penulis tertarik pada algoritma Paice/Husk. Algoritma Paice/Husk memiliki keunikan tersendiri yaitu pola stemming yang bersifat iteratif, sehingga algoritma ini dikenal dengan algoritma yang sangat kuat dan agresif. Namun, penulis juga akan menerapkan algoritma Porter yang bersifat linear sebagai bahan perbandingan dari tingkat akurasi.

Kamus bahasa Inggris-Indonesia banyak sekali dimanfaatkan untuk pencarian informasi berupa terjemahan. Beberapa orang kesulitan dalam menelusuri kata dasar dan kata berimbuhan lainnya pada kata bahasa Inggris yang berimbuhan. Sementara itu, belum banyak kamus yang dapat memberikan informasi berupa kata dasar, padahal kata dasar pada bahasa itu penting. Seperti pernyataan Plag (Plag, Word-Formation in English, 2003, p. 4), bahwa seorang pembicara mengerti 45.000 sampai 60.000 kata. Kata-kata tersebut harus disimpan dalam ingatan kita. Pada dasarnya, kata-kata yang kita ingat adalah kata dasar kemudian dihubungkan dengan imbuhan sehingga menimbulkan arti baru yang sedikit berbeda dari kata asal atau kata dasarnya. Dengan adanya permasalahan yang ada tersebut, diharapkan ide penulis dapat memudahkan pengguna kamus bahasa Inggris-Indonesia untuk menencari informasi kata dasar pada bahasa Inggris.

Ada beberapa manfaat yang diharapkan bisa diambil dari hasil penelitian ini.

Manfaat yang penulis dapatkan adalah untuk meneliti dan mengetahui performance atau kemampuan algoritma Paice/Husk dan algoritma Porter yang diterapkan untuk stemming bahasa Inggris pada aplikasi kamus bahasa Inggris-bahasa Indonesia. Selain itu, penulis dan pembaca juga dapat mengetahui algoritma terbaik untuk diterapkan pada kamus bahasa Inggris ke bahasa Indonesia. Manfaat lainnya yaitu pengguna mendapatkan kemudahan dalam mencari arti kata Inggris-Indonesia dan memperoleh informasi kata dasar dari bahasa Inggris berimbuhan.

Penyusunan Tulisan ini diorganisasikan sebagai berikut. Pada bagian kedua, akan dibahas mengenai Algoritma Paice/Husk dan penelitian terkait. Bagian tiga membahas tentang penerapan algoritma Paice/Husk pada Stemming Bahasa, Bagian Selanjutnya adalah Implementasi dilanjutkan oleh Bahasan mengenai Kesimpulan dan daftar referensi.

II. LANDASAN TEORI

A. Temu Kembali Informasi

Pada tahun 1961, sistem temu kembali informasi mulai diteliti oleh banyak orang karena pada tahun tersebut memiliki peran khusus dalam kegiatan mencari infomasi yang diperlukan oleh pengguna di perpustakaan. Aktivitas temu kembali informasi tidak hanya terbatas bagaimana menyimpan buku, tetapi juga meliputi pemahaman tentang penempatan informasi yang ada pada catalog dan indeks agar mudah dalam proses pencarian (Zaenab, 2002, p. 41).Temu balik informasi melakukan pengindeksan teks pada setiap dokumennya. Pengindeksan teks adalah proses untuk memutuskan apa yang akan digunakan untuk mempresentasikan dokumen tertentu.

Budi dan Aji (Budi & Aji, 2006, pp. G-27), Hal-hal yang dilakukan oleh sistem temu kembali informasi diantaranya adalah:

a. Mengolah catatan-catatan berupa teks dokumen, yaitu mengidentifikasi sejumlah istilah yang dianggap mewakili isi dokumen.
b. Mengidentifikasi permintaan informasi.
c. Menentukan dan mengambil informasi atau dokumen yang dibutuhkan sesuai dengan permintaan.

Ada beberapa masalah yang ditemui dalam penggunaan sistem temu kembali informasi adalah sebagai berikut:

a. Jumlah dokumen yang terambil bisa terlalu sedikit atau terlalu banyak jika dibandingkan dengan jumlah dokumen yang relevan (sesuai dengan keinginan pemakai) dalam sebuah kumpulan dokumen.
b. Isi dokumen yang terambil tidak sesuai dengan keinginan pemakai (user).

Permasalahan ini terjadi karena suatu dokumen diidentifikasi oleh sejumlah istilah yang belum tentu sepenuhnya mewakili isi dokumen. Suatu istilah yang dipakai bisa saja memiliki makna ganda. Hal ini menyebabkan dokumen yang terambil bisa tidak sesuai dengan keinginan pemakai.

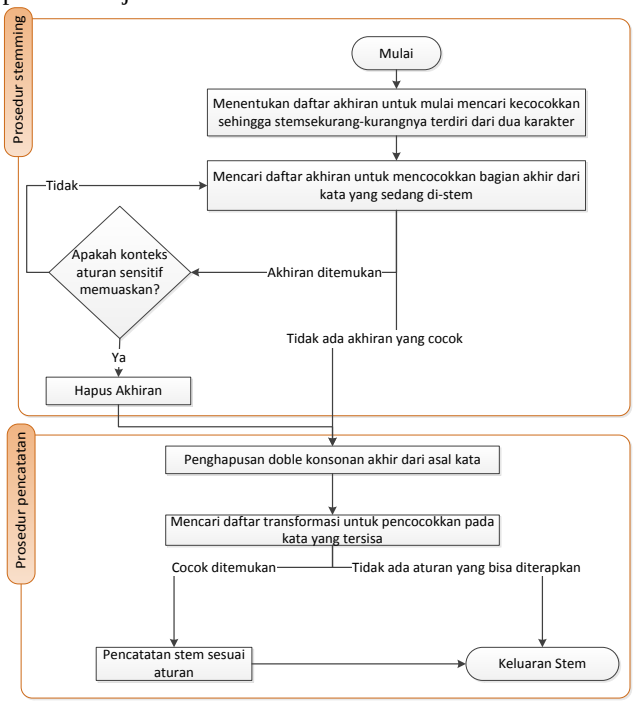
B. *Stemming*

Menurut Pribadi, Adi. W (Pribadi & Hasibuan, 2003, p. 7), proses stemming adalah proses pemotongan atau penghilangan imbuhan dari suatu kata. Menurut Lovins (Lovins, 1968, p. 22), sebuah algoritma stemming adalah prosedur komputasi yang mengurangi kata dengan akar yang sama ke bentuk umum, biasanya dengan pengupasan setiap kata akhiran yang derivasional dan infleksional.

Menurut Lennon, M. Pierce, D.S (Lennon, Peirce, Tarry, & Willett, 1981, pp. 177-183), ada 5 algoritma stemming yang dikenal, yaitu sebagai berikut ini:

1. Lovins Stemmer

   Menurut Hooper & Paice (Hooper & Paice, 2005, p. 2), lovins stemmer menggunakan context-sensitive. Stemmer ini dikembangkan oleh Julie Beth Lovins dari Institut Teknologi Massachusetts pada tahun 1968. Ini adalah awal dari penggunaan stemming yang ditargetkan untuk temu kembali informasi dan Computational Linguistics (Interaksi Komputer dan Manusia mengunakan Bahasa Alami). Stemmer ini sangat inovatif pada zamannya walaupun mempunyai masalah saat diujikan pada Information Rerieval dan Computational Linguistics. Pendekatannya tidak dapat menyamai dengan kata aslinya juga tidak cukup kompleks untuk kata dasar dengan menggunakan banyak imbuhan, tidak dapat sesuai dengan daftar aturan. Hal yang menarik dalam Lovins stemmer adalah daftar aturan yang diperoleh dari proses pembelajaran dari contoh kata-kata. Mungkin jika proses ini menggunakan contoh kata-kata yang lebih besar lagi akan dapat menghasilkan hasil yang lebih memuaskan.
   Permasalahan dalam algoritma Lovins adalah mengenai pemotongan imbuhan. Proses ini menggunakan aturan rekaman untuk mengembalikan kata imbuhan. Untuk memastikan kata dasar sesuai dengan maksud arti dari kata tersebut. Masalah utama dari algoritma Lovins adalah banyaknya hasil yang tidak sesuai dengan kata aslinya atau kata dasarnya. Stemmer ini tidak dapat mempresentasikan dokumen Temu Balik Informasi (TBI) karena pada saat menggunakan dalam skala besar, perekam akan membuat proses menjadi lambat.



Gambar 1 Flowchart Lovins Stemmer

2. Paice/Husk Stemmer

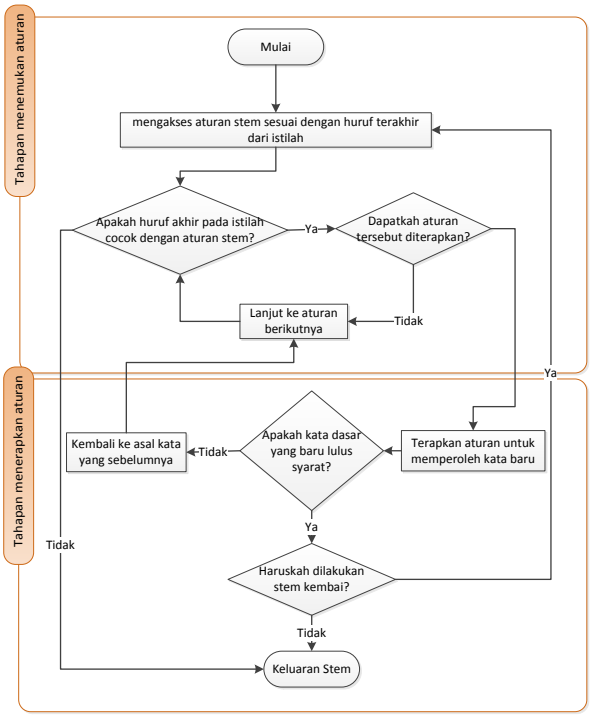
   Menurut Hooper & Paice (Hooper & Paice, 2005, p. 3), Paice/Husk stemmer dikembangkan oleh Chris Paice di Universitas Lancaster. Pada tahun 1980, Chris Paice meninggal dunia. Paice/Husk stemmer pertama kali dipublikasikan pada tahun 1990. Penelitiannya tentang stemmer dikembangkan oleh assistennya yang bernama Gareth Husk. Paice/Husk stemmer menyatakan bahwa setiap kata merupakan gabungan dari beberapa kata dasar. Walaupun Paice/Husk stemmer dikenal sangat efisien dan

dikenal sangat kuat dan agresif. Paice/Husk stemmer menggunakan satu table atau daftar aturan yang masing-masing bagian mempunyai tugas untuk memotong imbuhan. Teknik memotong yang digunakan adalah menghindari masalah pengejaan kecuali diuraikan lebih awal. Dengan mengganti kata akhiran bukan dengan memotong akhiran, stemmer melakukannya tanpa memisahkan langkah-langkah dalam proses stemming. Ini membantu memelihara efisiensi dari algoritma akan membuat menjadi lebih efektif. Aturan dari indeks kata dasar dan kata berimbuhan untuk memudahkan pencarian yang efisien adalah:

a. Sebuah akhir dari satu atau lebih karakter, yang diselenggarakan di urutan terbalik
b. Sebuah bendera utuh opsional '*'
c. Sebuah digit menentukan total penghapusan (nol atau lebih)
d. Sebuah string opsional tambahkan satu atau lebih karakter
e. Sebuah simbol kelanjutan, '>' atau '.'

Algoritma Paice/Husk stemmer mempunyai empat langkah utama. Penjelasannya berikut ini :

1. Pilih bagian yang relevan. Dengan memeriksa masukan kata dan mempertimbangkan aturan yang pertama yaitu bagian relevan dari table aturan.
2. Periksa kecocokan aturan. Jika kata inputan tidak cocok dengan aturan maka akan dilanjutkan ke dalam langkah 4.
3. Berlakunya aturan adalah dengan menghapus bagian dari imbuhan dan kemudian memeriksa simbol terminal dan terminal lainnya atau kembali ke langkah 1.
4. Melihat aturan lainnya lalu pindah ke aturan selanjutnya di dalam table, jika bagian kata telah berubah maka selesai dan pindah ke langkah 2. Seperti digambarkan Gambar 2.



Gambar 2 Flowchart Paice Husk Stemmer

3. Porter Stemmer

Porter, M.F. (1980: 130-137), Porter stemmer adalah pengembangan stemmer yang dibuat oleh Martin Porter di Universitas Cambridge pada tahun 1980. Stemmer berdasarkan pada akhiran di dalam bahasa Inggris (kurang lebih 1200 kata). Kebanyakan dengan menggunakan kombinasi akhiran yang sederhana. Stemmer ini mempunyai beberapa langkah. Secara rinci mempunyai lima langkah aturan. Jika suatu aturan akhiran disamakan dengan kata aslinya di mana kondisi-kondisi menyertakan aturan untuk diuji, apakah akan menghasilkan kata dasar. Jika akhiran sudah dapat dihikangkan, harus sesuai dengan yang diterapkan aturan tersebut (Porter, 1980). Setiap aturan yang melewati kondisi-kondisi dan diterima oleh aturan dan akhiran dihilangkan dan kemudian pindah ke langkah selanjutnya. Jika aturan tidak diterima maka aturan selanjutnya akan memeriksanya sampai semua aturan dari beberapa aturan. Hasil pemotongan akan dikembalikan kepada stemmer setelah lima langkah itu selesai diperiksa. Porter stemmer digunakan sangat luas dalam banyak aplikasi. Porter stemmer adalah algoritma yang paling banyak digunakan dalam riset Temu Balik Informasi.

Menurut Munir, et al (Koryati, Mandala, Munir, & Harlili, 2004, pp. L-30) mengatakan bahwa Efektifitas algoritma stemming dipengaruhi oleh beberapa faktor. Diantaranya adalah sebagai berikut:

a. Kesalahan dalam proses pemenggalan imbuhan dari kata dasarnya. Kesalahan ini dapat berupa:
   - Overstemming: yaitu pemenggalan imbuahan yang melebihi dari yang seharusnya.

- Understemming: yaitu pemenggalan imbuhan yang terlalu sedikit dari yang seharusnya.
- Unchange: yaitu kasus khusus dari understemming, dimana tidak terjadi pemenggalan imbuhan sama sekali.
- Spelling exception: yaitu huruf pertama kata dasar yang didapat tidak benar yang diakibatkan dari pemenggalan awalan.

b. Kekurangan dalam perumusan aturan penambahan imbuhan pada kata dasar. Hal ini dapat terjadi karena morfologi bahasa Inggris 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.

Pada penelitian "Penerapan Algoritma Paice/Husk Untuk Stemming Pada Kamus Bahasa Inggris ke Bahasa Indonesia", penulis mengunakan algorithm and experiments methodology (Moret & Shapiro, 2001). Penulis menganggap bahwa metodologi ini sangat cocok untuk diterapkan pada penelitian ini. Faktor-faktor yang mendukung penggunaan algorithm and experiments methodology pada penelitian ini adalah penulis melakukan eksperimen dengan menerapkan dan menguji algoritma pada suatu permasalahan dan setiap tahapan algorithm and experiments methodology dapat membantu penulis dalam menyelesaikan penelitian ini dengan sistematis.

Dalam Algorithm and Experiments Methodology terdapat tujuh tahapan yang harus dilakukan. tahapan-tahapan tersebut antara lain empiricism in algorithm design, implementation, modes of empirical assessment, experimental setup, measure, present and analyze the data, dan conclusion (Moret & Shapiro, 2001).

III. Analysis

Dalam penelitian yang berjudul "Penerapan Algoritma Paice/Husk Untuk Stemming Pada Bahasa Inggris ke Bahasa Indonesia", penulis melakukan metodologi algoritma dan eksperimen (Moret & Shapiro, 2001).

Berdasarkan gambar flowchart algoritma Paice/Husk pada gambar 12, dapat dijelaskan bahwa proses algoritma Paice/Husk terdiri dari empat langkah utama:

1) Pilih kelompok yang relevan
   a. Jika tidak ada kelompok yang sesuai dengan kata yang sedang diproses, maka lakukan terminasi.
   b. Jika ada kelompok yang sesuai, pertimbangkan rule pertama dari kelompok tersebut.
2) Periksa apakah rule tersebut dapat diterapkan
   a. Jika akhiran dari kata tidak beresuaian dengan akhiran dari rule (yang ditulis terbalik), lanjut ke langkah 4.
   b. Jika akhiran bersesuaian, jika terdapat intact flag dan katanya tidak intact, lanjut ke langkah 4.
   c. Jika acceptability conditions* tidak dipenuhi, lanjutkan ke langkah 4.
3) Terapkan rule
   a. Hapus akhiran sebanyak jumlah karakter yang dispesifikasikan, jika terdapat append string maka tambahkan ke bentuk kata yang baru.
   b. Jika simbol kontinuasinya adalah ".", maka lakukan terminasi.
   c. Jika simbol kontinuasinya adalah ">", kembali ke langkah 1.
4) Lihat rule lain
   a. Lanjutkan dengan rule berikutnya di tabel
   b. Jika kelompok rule telah berubah, maka lakukan terminasi.
   c. Jika belum, kembali ke langkah 2.

Acceptability conditions terdiri dari dua kondisi :

- Jika kata dimulai dengan vokal, maka setidaknya terdapat dua huruf setelah stemming.
- Jika kata dimulai dengan konsonan, maka setidaknya terdapat tiga huruf setelah stemming dan paling sedikit salah satu dari ketiga huruf tersebut adalah vokal.

IV. Implementasi

Untuk mempermudah pengkodean algoritma Paice/Husk, maka penulis merancang pseudocode algoritma Paice/Husk terlebih dahulu berdasarkan pengumpulan informasi mengenai algoritma Paice/Husk. Berikut ini adalah pseudocode algoritma Paice/Husk yang penulis terapkan pada kamus bahasa Inggris ke bahasa Indonesia.

Sebelum penjelasan mengenai algoritma pseudocode, penulis akan mendefinisikan beberapa variabel yang digunakan pada algoritma Paice/Husk. Pada bagian stemming, terdapat tiga fungsi yang dijalankan, yaitu PaiceHuskStemmerRules, checkAcceptability, dan getFirstRule. Pertama, pada fungsi checkAcceptability adalah fungsi yang bertugas untuk memeriksa kondisi jika kata dimulai dengan vokal, maka setidaknya terdapat dua huruf setelah stemming atau jika kata dimulai dengan konsonan, maka setidaknya terdapat tiga huruf setelah stemming dan paling sedikit salah satu dari ketiga huruf tersebut adalah vocal. Berikut ini adalah pseudocode checkAcceptability.

Untuk memaksimalkan hasil dari penelusuran kata dasar, maka dibutuhkan suatu tabel kata dasar sebagai tabel lookup. Tabel ini berfungsi untuk menghentikan pencarian kata dasar jika kata dasar ditemukan. Selain itu, tabel ini juga berfungsi untuk menghindari terjadinya overstemming.

Pada penerapan algoritma Paice/Husk di kamus bahasa Inggris-Indonesia, penulis menggunakan table lookup untuk membantu pencarian kata dasar.

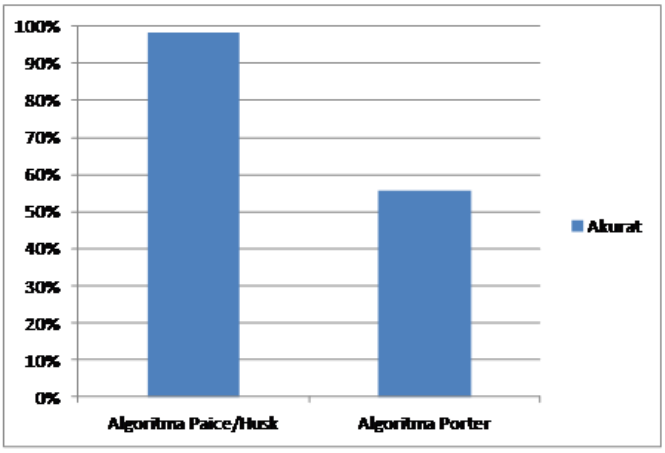
```
checkAcceptability(reversed_stem) begin
    if (preg_match("/[aeiouy]$/", reversed_stem)) begin
        // jika masukkan diawali dengan huruf vokal maka
        // setidaknya dua huruf lainnya adalah konsonan
        // (e.g., "owed"/"owing" --> "ow", but not "ear" --> "e")
        if panjang reversed_stem > 2 begin
            return 1;
        end
        return -1;
    else begin
        // jika masukkan diawali dengan huruf konsonan
        // maka sekurang-kurangnya ada tiga huruf setelah proses stemming
        if panjang reversed_stem <= 2
            return -1;
        // dan setidaknya harus ada satu huruf vokal atau "y"
        // (e.g., "saying" --> "say" and "crying" --> "cry",
        // tapi tidak "string" --> "str",
        // "meant" --> "me" or "cement" --> "ce")
        if not preg_match("/[aeiouy]/", reversed_stem) begin
            return -1;
        end
        return preg_match("/[aeiouy]/", reversed_stem);
    end
end
```

Gambar 3 *Pseudocode* Paice/Husk *Function* checkAcceptability

```
getFirstRule(reversed_form, rule_number, next_rule)begin
    PaiceHuskStemmerRules ← array_rules;
    nb_rules ← sizeOf(PaiceHuskStemmerRules);
    j ← 0;
    if next_rule is true begin
        j ←rule_number;
    end else
        j ← 0;
    for i = j; i < nb_rules; i++ begin
        rule ←PaiceHuskStemmerRules[i];
        rule ← preg_replace(rule_pattern_en,"\\1",rule);
        if strncasecmp(rule, reversed_form, strlen(rule)) = 0 begin
            return i;
        end
    end
    return -1;
end
```

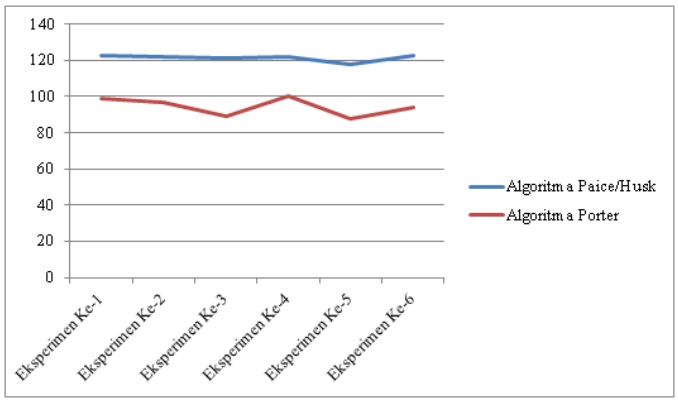
Gambar 4 *Pseudocode* Paice/Husk *Function* getFirstRule

Selanjutnya penulis menentukan penilaian apa yang akan diperoleh pada penelitian ini. Penulis menentukan bahwa pengukuran akan fokus pada tingkat akurasi (performa) dari masing-masing algoritma dan kesimpulannya akan menghasilkan best-case dan worst-case antara algoritma Paice/Husk dan Porter pada penerapannya di kamus.
Berdasarkan perhitungan tingkat akurasi algoritma Paice/Husk dan algoritma Porter pada eksperimen pertama yang melibatkan responden dan masukkan sesuai derivative pada referensi, penulis menggambarkan grafik sebagai berikut.



Gambar 5 Perbandingan Algoritma

Dari 124 percobaan pada masing-masing algoritma, algoritma Paice/Husk dapat melakukan proses stemming dengan akurat sebanyak 122 kata (98.3% akurat). Sedangkan algoritma Porter dapat melakukan proses stemming dengan akurat sebanyak 69 kata (55.6% akurat).



Gambar 6 Analisis Dan Presentasi Hasil Eksperimen

V. KESIMPULAN

Berdasarkan eksperimen data yang diambil dari Word-Formation In English dan kuesioner, tingkat akurasi algoritma Paice/Husk sebesar 98.3% sedangkan algoritma Porter sebesar 55.6%. Kemudian pada eksperimen data yang diambil secara acak, tingkat akurasi algoritma Paice/Husk juga lebih baik daripada algoritma Porter. Penyebab dari kegagalan penelusuran kata dasar pada algoritma Paice/Husk yaitu terdapat suatu kata yang hasil stem-nya sama dengan kata lainnya yang berbeda maknanya. Sedangkan penyebab dari kegagalan penelusuran kata dasar pada algoritma Porter diantaranya adalah overstemming, understemming (karena tidak ada rule untuk perubahan fonem), kurangnya rule (aturan), dan terdapat suatu kata yang hasil stem-nya sama dengan kata lainnya yang berbeda maknanya. Dengan demikian, algoritma Paice/Husk lebih cocok untuk penelusuran kata dasar pada kamus bahasa Inggris daripada algoritm

DAFTAR PUSTAKA

[1] _____. (2011). *Oxford Learner's Pocket Dictionary.* New York: Oxford University Press.

[2] Basuki, P. A. (2010). *Membangun Web Berbasis PHP dengan Framework Codeigniter*. Yogyakarta: Lokomedia.

[3] Bauer, L. (2003). *English Word-Formation.* Cambridge, United Kingdom: Press Syndicate of The University Of Cambridge.

[4] Bauer, L. (2004). *English Word-Formation.* Cambridge, United Kingdom: Press Syndicate of The University Of Cambridge.

[5] Budi, I., & Aji, R. F. (2006). Efektifitas Seleksi Fitur Dalam Temu-Kembali Informasi. *Seminar Nasional Aplikasi Teknologi Informasi 2006* , 4.

[6] Eastwood, J. (2012). *Oxford Learner's Pocket Grammar.* New York: Oxford University Press.

[7] Hooper, R., & Paice, C. (2005). *The Lancaster Algorithm*. Retrieved September 28, 2012 , from http://www.comp.lancs.ac.uk/computing/research/stemming/index.htm

[8] Jogiyanto. (2005). *Analisis dan Desain Sistem Informasi.* Yogyakarta: Penerbit Andi.

[9] Koryati, E., Mandala, R., Munir, R., & Harlili. (2004). Seminar Nasional Aplikasi Teknologi Informasi 2004. *Sistem Stemming Otomatis Untuk Kata dalam Bahasa Indonesia* .

[10] Lennon, M., Peirce, D. S., Tarry, B. D., & Willett, P. (1981). An Evaluation of some Conflation Algorithms for Information Retrieval. *Journal of Information Science* , 3: 177-183.

[11] Lovins, J. (1968). Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics, vol.11, nos.1 and 2* , 22-31.

[12] Moret, B. M., & Shapiro, H. D. (2001). Algorithms and Experiments: The New (and Old) Methodology. *Journal of Universal Computer Science, vol. 7, no. 5* , 434-446.

[13] Nazir, M. (2005). *Metode Penelitian.* Bogor: Ghalia Indonesia.

[14] Plag, I. (1983). *Word-Formation In English.* Germany: Universität-Gesamthochschule Siegen.

[15] Plag, I. (2003). *Word-Formation in English.* Cambridge: Cambridge University Press.

[16] Porter, M. (1980). An Algorithm for Suffix Stripping. *Program* , 4(3): 130-137.

[17] Pressman, R. S. (2001). *Rekayasa Perangkat Lunak.* Yogyakarta: Andi Yogyakarta.

[18] Pribadi, A. W., & Hasibuan, Z. A. (2003). Implementing inference Networks for Information Retrieval System ini Indonesia Language. *Faculty of Computer Science University of Indonesia, Indonesia.*

[19] Rizky, S. (2011). *Konsep Dasar Rekayasa Perangkat Lunak.* Jakarta: Prestasi Pustakarya.

[20] Subagja, R. (2007). *Pencarian Kata Berimbuhan Pada Kamus Bahasa Sunda dengan Menggunakan Algoritma Stemming.* (Jakarta: Perpustakaan Utama Universitas Islam Negeri Syarif Hidayatullah Jakarta) t.d.

[21] Utami, E., & Raharjo, S. (2004). *Logika, Algoritma dan Implementasinya dalam Bahasa Python di GNU/Linux.* Yogyakarta: Andi.

[22] Wardhana, R. R. (2007). *Stemming Bahasa Inggris Untuk Kamus Bahasa Inggris-Indonesia.*

[23] Yogatama, D. (2008). *Studi Penggunaan Stemming untuk Meningkatkan Performansi Sistem Temu Balik Informasi.* Bandung: Institut Teknologi Bandung.

[24] Zaenab, R. S. (2002). Efektivitas temu Kembali Informasi dengan Menggunakan Bahasa Alami pada CD-ROM AGRIS dan CAB Abstracts. *Jurnal Perpustakaan Pertanian . Vol. 11. Nomor 2* .