

## PERBANDINGAN ALGORITMA STEMMING PORTER DENGAN ARIFIN SETIONO UNTUK MENENTUKAN TINGKAT KETEPATAN KATA DASAR

Dian Novitasari

Program Studi Teknik Informatika, Universitas Indraprasta PGRI

Email : diannovita.unindra@gmail.com

### Abstrak

*Information Retrieval* (IR) adalah sebuah proses yang digunakan untuk menemukan kembali informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis yang didasarkan pada suatu *query* (inputan *user*). Meningkatnya jumlah dokumen teks yang dapat diakses di internet dan diikuti dengan meningkatnya kebutuhan pengguna akan perangkat pencarian informasi yang efektif dan efisien. Efektif berarti pengguna mendapatkan dokumen yang relevan dengan *query* yang diinputkan. Efisien berarti waktu dalam pencarian yang singkat. Salah satu aplikasi umum dari IR adalah *search-engine* atau mesin pencarian yang terdapat pada jaringan internet yang dapat digunakan oleh pengguna untuk mencari halaman-halaman Web yang dibutuhkannya. *Stemming* adalah salah satu cara yang digunakan untuk meningkatkan performa *Information Retrieval* dengan cara mentransformasi kata-kata dalam sebuah dokumen teks ke bentuk kata dasarnya. Kata dasar tersebut akan disimpan sebagai index. Dalam Bahasa Indonesia banyak metode *stemming* yang sudah ada, salah satunya yaitu Porter dan Arifin Setiono. Penelitian ini membandingkan dua *stemmers* Indonesia, yaitu Porter dan Arifin Setiono. Hasil yang dicapai dalam penelitian ini adalah untuk mengetahui *stemmer* mana yang lebih efektif dalam menentukan kata dasar dari perbandingan algoritma tersebut.

**Kata Kunci** : Stemming Bahasa Indonesia, Algoritma Porter, Algoritma Arifin Setiono, Presisi (%)

### Pendahuluan

*Information Retrieval* adalah studi tentang sistem pengindeksan, pencarian, dan mengingat data, khususnya teks atau bentuk tidak teratur lainnya. *Information Retrieval* merupakan suatu pencarian informasi (biasanya berupa dokumen) yang didasarkan pada suatu *query* (inputan *user*) yang diharapkan dapat memenuhi keinginan *user* dari kumpulan dokumen yang ada. *Stemming* adalah salah satu cara yang digunakan untuk meningkatkan performa *Information Retrieval* dengan cara mentransformasi kata-kata dalam sebuah dokumen teks ke bentuk kata dasarnya. Dalam penelitian ini akan dibandingkan dua algoritma stemming yaitu Algoritma Porter dengan Algoritma Arifin Setiono. Tujuan penelitian ini adalah untuk membandingkan tingkat presisi(%) Algoritma Porter dengan Algoritma Arifin Setiono dalam proses stemming pada teks berbahasa Indonesia.

### Tinjauan Pustaka

#### *Information Retrieval*

*Information Retrieval* merupakan suatu pencarian informasi (biasanya berupa dokumen) yang didasarkan pada suatu *query* (inputan *user*) yang diharapkan dapat memenuhi keinginan *user* dari kumpulan dokumen yang ada. Tujuan dari sistem IR (*Information Retrieval*) adalah memenuhi kebutuhan informasi pengguna dengan *me-retrieve* semua dokumen yang mungkin relevan, pada waktu yang sama *me-retrieve* sesedikit mungkin dokumen yang tak-relevan. Tahap-tahap yang terjadi pada proses indeksing dalam *Information Retrieval* adalah:

1. **Word Token**: yaitu mengubah dokumen menjadi kumpulan *trem* dengan cara menghapus semua karakter dalam tanda baca yang terdapat pada dokumen dan mengubah kumpulan *trem* menjadi *lowercase*.
2. **Stopword Removal**: Proses penghapusan kata-kata yang sering ditampilkan dalam dokumen seperti: dan, atau, tidak dan sebagainya.
3. **Stemming**: Proses mengubah suatu kata bentuk menjadi kata dasar.
4. **Trem Weighting**: Proses pembobotan setiap *Trem* didalam dokumen.

### **Stemming**

*Stemming* adalah proses menghilangkan imbuhan dari suatu kata dari suatu dokumen atau *query* menjadi *root word*.

Efektifitas algoritma *stemming* dipengaruhi oleh beberapa faktor:

1. **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.
2. **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.
3. **Unchange**: yaitu kasus khusus dari *understemming*, dimana tidak terjadi pemenggalan imbuhan samasekali. Contoh: kata “telapak”, setelah pemenggalan kata dasar yang didapat tetap “telapak”. Kesalahan ini dapat ditimbulkan karena kekurangan pada aturan pola imbuhan yang didefinisikan.
4. **Spelling exception**: yaitu huruf pertama kata dasar yang didapat tidak benar yang diakibatkan dari penggalan 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).

### **Algoritma Stemming**

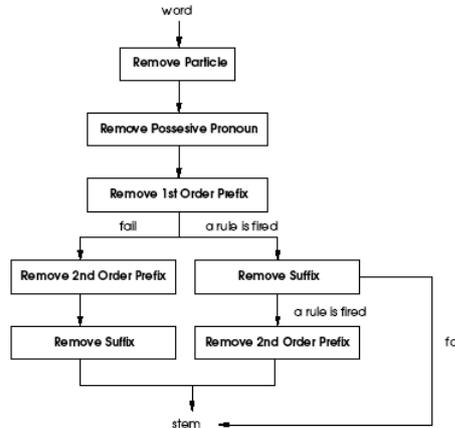
Algoritma *Stemming* adalah prosedur komputasi yang mencari asal kata dari suatu kata dalam kalimat yang dilakukan dengan cara memisahkan masing-masing kata dari kata dasar dan imbuhan. Pada saat ini ada beberapa algoritma *stemming* untuk Bahasa Indonesia yang telah dikembangkan diantaranya yaitu: Algoritma Nazief dan Andriani, Algoritma Porter, serta Algoritma Arifin dan Setiono.

Algoritma *Stemming* Nazief dan Andriani, Algoritma ini dibuat oleh Boby Nazief dan Mirna Andriani dari Fakultas Ilmu Komputer Universitas Indonesia tahun 1996, algoritma ini mengacu pada aturan morfologi Bahasa Indonesia yang mengelompokkan imbuhan, yaitu imbuhan yang diperbolehkan atau imbuhan yang tidak diperbolehkan. Pengelompokan ini termasuk imbuhan di depan (awalan), imbuhan kata belakang (akhiran), imbuhan kata di tengah (sisipan), dan kombinasi imbuhan pada awal dan akhir kata (konfiks). Algoritma ini menggunakan kamus kata keterangan yang digunakan untuk mengetahui bahwa proses *stemming* telah mendapatkan kata dasar.

### **Porter Stemmer for Bahasa Indonesia**

*Porter Stemmer for Bahasa Indonesia* dikembangkan oleh Fadillah Z. Tala pada tahun 2003. Implementasi *Porter Stemmer for Bahasa Indonesia* berdasarkan English *Porter*

Stemmer yang dikembangkan oleh W.B. Frakes pada tahun 1992. Karena bahasa Inggris datang dari kelas yang berbeda, beberapa modifikasi telah dilakukan untuk membuat Algoritma Porter dapat digunakan sesuai dengan bahasa Indonesia. Desain dari Porter Stemmer for Bahasa Indonesia dapat dilihat pada gambar dibawah ini :



Gambar 1. Desain Porter Stemmer for Bahasa Indonesia

**Arifin and Setiono stemmer for Bahasa Indonesia**

Dalam Algoritma Arifin dan Setiono ini didahului dengan pembacaan tiap kata dari data yang ada. Sehingga tahap yang dilakukan dalam Algoritma ini adalah sebagai berikut :

1. Pemeriksaan semua kemungkinan bentuk kata. Setiap kata diasumsikan memiliki 2 Awalan (*prefiks*) dan 3 Akhiran (*sufiks*).  
 Jika dalam kata yang diperiksa tidak memiliki imbuhan sebanyak imbuhan seperti formula di atas, maka imbuhan yang kosong atau tidak ada tersebut diberi tanda x untuk prefiks dan diberi tanda xx untuk sufiks.
2. Pemotongan dalam Algoritma ini dilakukan secara berurutan sebagai berikut :  
 AW : AW (Awalan)  
 AK : AK (Akhiran)  
 KD : KD (Kata Dasar)
  - a. AW I, hasilnya disimpan pada pe1 (*prefiks* 1)
  - b. AW II, hasilnya disimpan pada pe2 (*prefiks* 2)
  - c. AK I, hasilnya disimpan pada su1 (*sufiks* 1)
  - d. AK II, hasilnya disimpan pada su2 (*sufiks* 2)
  - e. AK III, hasilnya disimpan pada su3 (*sufiks* 3)
 Dalam setiap tahap pemotongan di atas selalu diikuti dengan pemeriksaan di dalam kamus. Hal ini untuk mengetahui apakah hasil pemotongan tersebut sudah ada dalam bentuk dasar. Apabila pemeriksaan ini berhasil maka proses dinyatakan selesai dan tidak perlu melanjutkan proses pemotongan imbuhan selanjutnya.
3. Akan tetapi, apabila sampai pada pemotongan AK III, belum ditemukan dalam kamus, maka akan dilakukan proses kombinasi. Kata dasar yang dihasilkan dikombinasikan dengan imbuhan-imbuhan dalam 12 konfigurasi berikut :
  - a. KD
  - b. KD + AK III
  - c. KD + AK III + AK II
  - d. KD + AK III + AK II + AK I
  - e. AW I + AW II + KD
  - f. AW I + AW II + KD + AK III
  - g. AW I + AW II + KD + AK III + AK II

- h. AW I + AW II + KD + AK III + AK II + AK I
- i. AW II + KD
- j. AW II + KD + AK III
- k. AW II + KD + AK III + AK II
- l. AW II + KD + AK III + AK II + AK I

Kombinasi a, b, c, d, h, dan l sudah diperiksa pada tahap sebelumnya, karena kombinasi ini adalah hasil pemotongan bertahap tersebut. Dengan demikian, kombinasi yang masih perlu dilakukan tinggal 6 yakni pada kombinasi-kombinasi yang belum dilakukan (e, f, g, i, j, dan k). Apabila dalam proses kombinasi yang dilakukan itu ada, maka pemeriksaan pada kombinasi lainnya sudah tidak diperlukan lagi.

Pemeriksaan dalam 12 kombinasi ini sangat diperlukan karena fenomena *overstemming* pada algoritma pemotongan imbuhan. Kelemahan ini berakibat pada pemotongan bagian kata yang sebenarnya adalah milik kata dasar itu sendiri yang kebetulan mirip dengan salah satu jenis imbuhan yang ada. Dengan 12 kombinasi itu, pemotongan yang sudah terlanjur tersebut dapat dikembalikan sesuai posisinya.

### Presisi Algoritma

Tingkat keakuratan algoritma dihitung menggunakan aturan berikut:

$$\text{Presisi} = (\text{RW} / \text{W}) * 100\% \dots\dots$$

Dimana W adalah jumlah kata yang terstem dan RW adalah jumlah kata yang distemming dengan benar. Akurasi dinyatakan dalam persen (%). Perbaikan pada algoritma akan dilakukan jika akurasi algoritma belum memenuhi syarat batas (threshold) yang telah ditentukan. Jika akurasi sudah memenuhi syarat batas maka algoritma tersebut diuji menggunakan testing set yaitu sampel kata dengan jumlah yang lebih besar yang tidak termasuk dalam training set.

### BlackBox Testing

Pengujian *BlackBox* atau sering disebut dengan pengujian *behavioral* (pengujian partisi) berfokus pada persyaratan fungsional perangkat lunak. Pengujian *BlackBox* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Pengujian *BlackBox* berusaha menemukan :

- a. Fungsi-fungsi yang tidak benar atau hilang.
- b. Kesalahan *interface*.
- c. Kesalahan dalam struktur data atau akses *database* eksternal.
- d. Kesalahan kinerja.
- e. Inisialisasi dan kesalahan terminasi.

### Penelitian yang Relevan

- a. Penelitian yang dilakukan oleh Rila Mandala, Rinaldi Munir, dkk. Laboratorium Ilmu dan Rekayasa Komputasi, Institut Teknologi Bandung dengan judul: “Sistem *Stemming* Otomatis untuk Kata dalam Bahasa Indonesia”. Algoritma Othman dikembangkan pada tahun 1993, menggunakan pendekatan berdasarkan aturan morfologi sederhana. Aturan yang digunakan algoritma Othman berisi 121 aturan imbuhan, yang mendefinisikan awalan, akhiran, sisipan dan pasangan awalan-akhiran. Proses pemenggalan imbuhan dilakukan dengan cara mencocokkan imbuhan yang ada didalam aturan dengan kata yang dimasukan, hasil penggalan diperiksa terhadap kamus. Jika kata hasil penggalan ada didalam kamus, maka kata hasil penggalan adalah kata dasar.
- b. Penelitian yang dilakukan oleh Agus Zainal Arifin dan Ari Novan Setiono, Jurusan Teknik Informatika, Institut Teknologi Sepuluh November (ITS) – Surabaya dengan judul: “Klasifikasi Dokumen Kejadian Berbahasa Indonesia dengan Algoritma

*SinglePass Clustering*” Penelitian ini berusaha untuk mengklasifikasi dokumen dengan menggunakan algoritma *single pass clustering*. Klasifikasi ini ditekankan untuk dokumen berita berbahasa Indonesia. Algoritma ini didahului dengan pembacaan tiap kata dari file sampel. Sehingga input dari algoritma ini adalah sebuah kata yang kemudian dilakukan pemeriksaan semua kemungkinan bentuk kata. Dan hasil dari uji coba tersebut adalah *Stemming* Bahasa Indonesia yang digunakan efektif untuk memilih prototype kata dasar yang digunakan sebagai pembeda antar dokumen dan *Single Pass clustering* cukup handal digunakan sebagai algoritma untuk klasifikasi *event*.

- c. Penelitian yang dilakukan oleh Ledy Agusta 2009, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana dengan Judul “Perbandingan Algoritma *Stemming* Porter dengan Algoritma *Stemming* Nazief & Andriani Untuk *Stemming* Dokumen Teks Bahasa Indonesia”.

Dalam tulisan ini, penulis akan membandingkan efektifitas algoritma *stemming* Nazief & Andriani dengan algoritma *stemming* Porter untuk proses *stemming* pada teks Bahasa Indonesia, sehingga akhirnya akan diketahui algoritma manakah yang lebih cepat, lebih akurat, atau lebih banyak melakukan kesalahan proses *stemming*. Tujuan penelitian ini adalah untuk membandingkan kemampuan dan ketepatan algoritma Nazief & Andriani dengan algoritma Porter untuk proses *stemming* pada teks Bahasa Indonesia.

Untuk membandingkan performa masing-masing algoritma maka dibuat aplikasi sederhana proses *stemming* menggunakan algoritma Porter dan Nazief & Andriani. Menghitung presisi dan waktu proses dari masing-masing algoritma. Menguji menggunakan 30 dokumen sampel.

Proses perbandingan algoritma Porter dengan algoritma Nazief & Andriani dilakukan dengan membuat program sederhana yang memproses dokumen teks inputan sehingga diketahui *stem*, waktu proses, dan presisi dari hasil *stemming* dokumen tersebut. Dan hasil dari uji coba perbandingan tersebut adalah proses *stemming* dokumen teks berbahasa Indonesia menggunakan Algoritma Porter membutuhkan waktu yang lebih singkat dibandingkan dengan *stemming* menggunakan Algoritma Nazief & Andriani. Dan Proses *stemming* dokumen teks berbahasa Indonesia menggunakan Algoritma Porter memiliki prosentase keakuratan (presisi) lebih kecil dibandingkan dengan *stemming* menggunakan Algoritma Nazief & Andriani.

## Metodologi Penelitian

### Analisa Kebutuhan

Untuk dapat mengetahui gambaran secara umum dari sistem yang akan dibangun, maka perlu dilakukan tahap analisis kebutuhan yaitu metode pemilihan sampel, metode pengumpulan data, analisa kebutuhan perangkat lunak, dan analisa kebutuhan perangkat keras.

### Metode Pemilihan Sampel

Dokumen yang digunakan sebagai sampel adalah dokumen text Bahasa Indonesia yang berisikan kata kerja, kata sifat, kata benda, dan kata keterangan yang diambil dari Kamus Besar Bahasa Indonesia (KBBI). Jumlah sampel adalah 40 kata Bahasa Indonesia yang berimbuhan yang sudah ditentukan oleh penulis. Jumlah sampel yang ditentukan mengacu kepada penelitian sebelumnya. 40 kata berimbuhan ini nantinya akan digunakan untuk mengevaluasi presisi(%) dalam penentuan kata dasar dari kata yang berimbuhan.

### Teknik Analisis

#### Teknik Validitas Perancangan

##### 1. Exact Match

Merupakan Hasil yang menyatakan bahwa hasil yang diperoleh dari proses *stemmer* dengan kata dasar sebenarnya sudah sama.

**2. Spelling Exception**

Yaitu huruf pertama kata dasar yang didapat tidak benar yang diakibatkan dari penggalan 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).

**3. Unchange**

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.

**4. 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.

**Hasil dan Pembahasan**

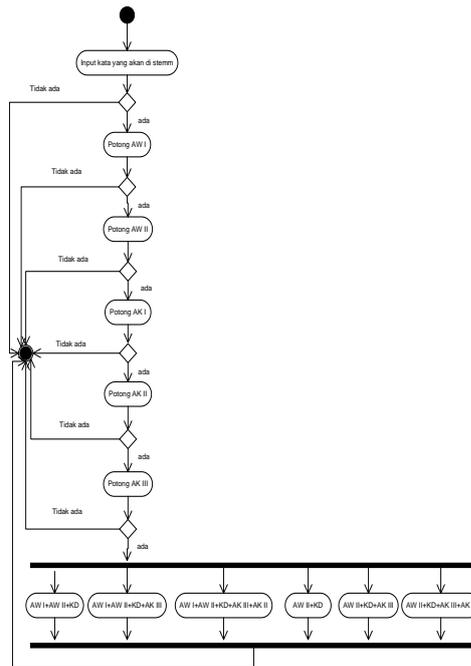
**1. Perancangan Activity Diagram**

a. *Activity diagram* proses *stemming* menggunakan aturan morfologi imbuhan yang terdapat pada algoritma Porter sebagai berikut:



**Gambar 2. Activity Diagram Alur Algoritma Stemming Porter**

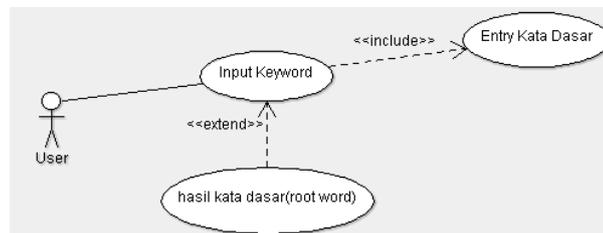
b. *Activity diagram* proses *stemming* menggunakan aturan morfologi imbuhan yang terdapat pada algoritma Arifin Setiono sebagai berikut:



Gambar 3. Activity Diagram Alur Algoritma Stemming Arifin Setiono

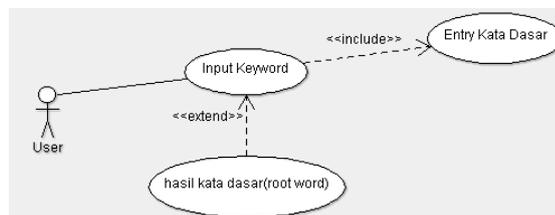
2. Perancangan Use Case Diagram

a. Use Case diagram sistem stemming Porter



Gambar 4. Use Case diagram sistem Stemming Porter

b. Use case diagram sistem Stemming Arifin Setiono



Gambar 5. Use Case diagram sistem Stemming Arifin Setiono

3. Hasil Pengujian

Pada pembahasan ini disampaikan uraian tentang spesifikasi program atau sistem yang berupa penjelasan tentang rancangan layar, perangkat keras dan perangkat lunak pendukung yang digunakan, serta hasil uji coba 40 kata berimbuhan Bahasa Indonesia.

**Hasil Tabel Uji Coba**

Berikut ini hasil table uji coba dari 40 kata yang berimbuhan yang diproses oleh algoritma *Stemming* Bahasa Indonesia sebagai berikut:

**Tabel 1. Tabel Katagori Hasil *Stemmer Porter***

<b>Katagori Hasil <i>Stemmer</i></b>	<b>Total</b>	<b>Persentase (%)</b>
<i>Exact Match</i>	36	90 %
<i>Unchange</i>	1	2,5 %
<i>Spelling Exception</i>	3	7,5 %
<i>Overstemming</i>	0	0 %
<b>Total</b>	<b>40</b>	<b>100%</b>

Perhitungan persentase *Stemmer Porter* :

- a) Exact Match :  $\frac{36}{40} \times 100\% = 90\%$
- b) Unchange :  $\frac{1}{40} \times 100\% = 2,5 \%$
- c) Spelling Exception :  $\frac{3}{40} \times 100\% = 7,5 \%$
- d) Overstemming :  $\frac{0}{40} \times 100\% = 0 \%$

**Tabel 2. Tabel Katagori Hasil *Stemmer Arifin Setiono***

<b>Katagori Hasil <i>Stemmer</i></b>	<b>Total</b>	<b>Persentase(%)</b>
<i>Exact Match</i>	38	95 %
<i>Unchange</i>	1	2,5 %
<i>Spelling Exception</i>	0	0 %
<i>Overstemming</i>	1	2,5 %
<b>Total</b>	<b>40</b>	<b>100 %</b>

Perhitungan persentase *stemmer Arifin Setiono* :

- a) Exact Match :  $\frac{38}{40} \times 100\% = 95 \%$
- b) Unchange:  $\frac{1}{40} \times 100\% = 2,5 \%$
- c) Spelling Exception :  $\frac{0}{40} \times 100\% = 0 \%$

d) Overstemming :  $\frac{1}{40} \times 100\% = 2,5 \%$

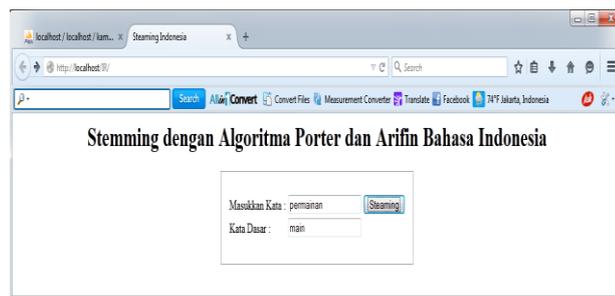
Dari hasil uji coba yang dilakukan dengan 40 sampel dokumen teks kata berimbuhan Bahasa Indonesia yang sudah dipilih oleh penulis, menghasilkan tingkat katagori hasil *stemmer Exact Match* sebesar 90% untuk stemmer Porter dan 95% untuk stemmer Arifin Setiono, *Unchange* sebesar 2,5% untuk stemmer Porter dan 2,5% untuk stemmer Arifin Setiono, *Spelling Exception* sebesar 7,5% untuk stemmer Porter dan 0% ntuk stemmer Arifin Setiono, dan *Overstemming* sebesar 0% untuk stemmer Porter dan 2,5% untuk stemmer Arifin Setiono.

**Pembahasan *Testing System***

Untuk mengetahui kesesuaian fungsionalitas sistem dengan persyaratan sistem dan kebutuhan pengguna, maka dilakukan pengujian sistem. Pengujian yang berhubungan dengan fungsionalitas perangkat lunak menggunakan metode *BlackBox*.

**Testing dengan *Black Box***

Metode pengujian *black box* adalah bentuk pengujian yang dilakukan terhadap *user interface* dari sistem yang dihasilkan agar hasil akhirnya benar dan dapat digunakan oleh *user* sebagai bahan informasi untuk mendukung pengambilan keputusan. Metode pengujiannya adalah dengan memberikan *input* ke dalam sistem dan kemudian diuji apakah dapat menghasilkan *output* sesuai yang diharapkan. Apabila sesuai, maka sistem dapat dikatakan lolos dari pengujian *black box*.



**Gambar 6. Rancangan Layar Utama Pengujian Stemming Porter dan Arifin Setiono**



**Gambar 7. Rancangan Layar Hasil Pengujian Stemming Porter dan Arifin Setiono**

**Simpulan dan Saran**

**Simpulan**

Berdasarkan hasil penelitian dan pembahasan yang disampaikan pada bab sebelumnya, maka terdapat beberapa kesimpulan sebagai berikut:

1. Dari hasil uji coba yang dilakukan dengan 40 sampel dokumen teks kata berimbuhan Bahasa Indonesia, menghasilkan tingkat katagori hasil *stemmer Exact Match* sebesar

- 90% untuk stemmer Porter dan 95% untuk stemmer Arifin Setiono, *Unchange* sebesar 2,5% untuk stemmer Porter dan 2,5% untuk stemmer Arifin Setiono, *Spelling Exception* sebesar 7,5% untuk stemmer Porter dan 0% untuk stemmer Arifin Setiono, dan *Overstemming* sebesar 0% untuk stemmer Porter dan 2,5% untuk stemmer Arifin Setiono. Hal ini menunjukkan bahwa Proses stemming dokumen teks berbahasa Indonesia menggunakan Algoritma Arifin & Setiono memiliki prosentase keakuratan (presisi) lebih tinggi dibandingkan dengan stemming menggunakan Algoritma Porter.
2. Kamus yang digunakan mempengaruhi perhitungan presisi. Semakin lengkap kamus yang digunakan maka semakin akurat pula nilai presisinya.

### Saran

Ada beberapa hal yang perlu disampaikan agar menjadi bahan pertimbangan untuk melengkapi penelitian dimasa mendatang. Hal-hal tersebut diungkapkan dalam bentuk saran sebagai berikut:

- a. Perlunya penelitian lebih lanjut untuk mengetahui tingkat efektifitas dari masing-masing algoritma tersebut.
- b. Perlunya kelengkapan kamus yang digunakan agar dapat meningkatkan tingkat presisi dari algoritma yang digunakan.

### Daftar Pustaka

- Adriani, M., Nazief, B., Asian, J., & Williams, H. E. (1996). Stemming Indonesia A Confixs Stripping Approach. *ACM Transactions on Asian Language Information Processing, Vol. 6, No. 4*, 13.
- Agusta, L.(2009). Comparasi Algoritma Stemming Porter dengan Algoritma Nazief dan Andriani untuk Stemming Dokumen Teks Bahasa Indonesia. *Konfransi Nasional Sistem dan Informatika*. Bali.
- Arifin, A. Z., & Setiono, A. N. (2002). Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering. *www.its.ac.id*.
- Asian, J., & Williams, H. E. (2005). Stemming Indonesia. *Australia Computer Science Conference*.
- Garfield, E. (1997). A Tribute To Calvin N. Mooers, A Pioneer Of Information Retrieval. *The Scientist*.
- Kurniawan, E., Fransiska, M., Tinaliah, Rachmansyah. (2014). Penerapan Algoritma K-Means untuk Clustering Dokumen E-Jurnal STMIK GI MDP
- Malem, J. M., & Abidin, T. F. (2011). Analisa dan Evaluasi Afiks Stemming untuk Bahasa Indonesia. Seminar Nasional dan Expo Teknik Elektro. Banda Aceh.
- Mandala, R., & Munir, R. (1993). Sistem Stemming Otomatis untuk kata dalam Bahasa Indonesia. *Seminar Nasional Aplikasi Teknologi Informasi*. Yogyakarta.