

**IMPLEMENTASI ALGORITMA TWO SLIDING WINDOWS UNTUK MEMPERCEPAT  
PENCARIAN DOKUMEN****TWO SLIDING WINDOWS ALGORITHM IMPLEMENTATION TO INCREASE DOCUMENTS  
SEARCHING PERFORMANCE**

Dedy Arisandi\*, Erna Budhiarti Nababan\*\*, dan Frans Octavianus\*\*\*

Program Studi S1 Teknologi Informasi

Fakultas Ilmu Komputer dan Teknologi Informasi

Universitas Sumatera Utara

\*dedyarisandi@usu.ac.id

\*\* ernabr@usu.ac.id

\*\*\*bryanlite255@gmail.com

Diterima : 20 April 2016

Direvisi : 9 Mei 2016

Disetujui : 20 Juni 2016

**ABSTRAK**

Salah satu metode yang umum digunakan dalam pencarian dokumen adalah dengan menggunakan kata dasar atau kata berimbuhan sebagai kata kunci. Kata-kata tersebut kemudian akan dicocokkan dengan seluruh dokumen yang tersimpan. Pencarian dilakukan dengan mencocokkan kata yang sama terhadap seluruh isi dokumen tersebut. Metode ini kurang efektif dan cenderung lambat. Oleh karena itu, pencarian dokumen perlu menggunakan algoritma yang dapat mempercepat proses tersebut. Pengubahan kata berimbuhan menjadi kata dasar dilakukan dengan menggunakan Metode Stemming Porter. Proses berikutnya adalah dengan menggunakan algoritma Two Sliding Windows (TSW) untuk mempercepat pencarian pola. TSW menggunakan jendela dua sisi secara paralel untuk mencari kata atau sinonim yang cocok. Berdasarkan hasil uji coba, algoritma ini dapat mempercepat proses pencarian dokumen dengan tingkat kebenaran 97.93%.

**Kata Kunci :** Two Sliding Windows, Pencarian dokumen, Sinonim kata, Stemming Porter.

**ABSTRACT**

A common method to perform document searching is to use the root or inflected words as keywords. Those keywords is then matched to all stored documents. The search is performed through all the document's content to find word with exact similarity. This is not an effective solution and tend to be slow. A new algorithm is needed to speed up the performance of document-searching. Stemming Porter Method is used to reduce the inflected words to it's root form. The next step is to use Two Sliding Windows (TSW) Algorithm to speed up pattern finding. TSW uses two-sided windows parallely to find matched word or synonym. Based on the results, this algorithm speeds up the performance of document searching with 97.93% accuracy.

**Keywords :** Two Sliding Windows, Document Searching, Synonym, Porter Stemming.

**PENDAHULUAN**

Saat ini informasi sangat mudah diperoleh terutama melalui media internet dan media penyimpanan. Dengan banyaknya informasi

yang terkumpul atau tersimpan dalam jumlah yang besar, pengguna (*user*) akan kesulitan mendapatkan informasi berbentuk dokumen yang diinginkan. Semakin bertambahnya

dokumen, penggunaan sistem pencarian pola teks ataupun informasi yang terdapat di dalamnya menjadi penting. Dengan adanya sistem pencarian teks banyak menghemat waktu pengerjaan untuk menemukan informasi yang terdapat dalam dokumen. Sistem pencarian menerima kata kunci (*keyword*) yang ditulis oleh pengguna dan melakukan pencocokan dengan basis data (*database*). Selama ada penyimpanan dan pencarian kembali dokumen, sistem dapat bekerja dengan baik.<sup>1</sup>

Dokumen yang ada bisa diolah untuk menemukan informasi yang dibutuhkan. Dalam dokumen terkadang terdapat kata penghubung yang saling berhubungan dan menimbulkan makna yang berbeda, sedangkan dalam bahasa seringkali terdapat sinonim kata yang sama maknanya dengan kata kunci pencarian. Dalam beberapa dokumen seperti artikel bahasa terkadang memiliki kata-kata yang masih jarang digunakan atau dijumpai seperti beranda, telaga, bilik, misai, dan jeram. Kata-kata tersebut mengandung persamaan kata dengan kata-kata umum yang sudah sering digunakan dalam pembicaraan sehari-hari. Akan tetapi, kata kunci pencarian yang umumnya dicari oleh pengguna, dominan berupa kata-kata umum yang sering dijumpai.<sup>2</sup>

Apabila dihadapkan dengan jumlah dokumen yang banyak dan kemunculan kata baku yang masih jarang dijumpai, diperlukan adanya suatu metode untuk menyelesaikan permasalahan tersebut. Dalam implementasinya, penulis menggunakan algoritma *pattern matching* sebagai algoritma pencarian kata dalam dokumen. Salah satu metode yang digunakan dalam pencocokan pola (*pattern matching*) yaitu algoritma *Two Sliding Windows* (TSW) yang mempunyai keunggulan dalam fase pencarian (*searching phase*).<sup>3</sup> Algoritma *Two Sliding Windows* mempunyai performansi yang lebih unggul khususnya jika pola tersebut berada diakhir teks. Hasil pengujian menunjukkan percobaan dan perbandingan dengan algoritma lainnya seperti algoritma *Knuth-Morris-Pratt* dan

*Boyer-Moore* lebih cepat dan membutuhkan usaha yang lebih sedikit.

Algoritma TSW memulai *pre-processing phase* untuk membagi *string* menjadi dua bagian (*two windows*) sebanyak  $n/2$  ukuran *string*. Kemudian, dilakukan *scanning* pada fase pencarian dari kiri dan kanan secara paralel dan bersamaan.<sup>3</sup> Dengan demikian, dalam penelitian ini penulis akan menggunakan algoritma *Two Sliding Windows* untuk mempercepat pencarian dokumen.

Terdapat beberapa penelitian yang sudah dilakukan penulis terdahulu tentang pencarian kata dalam dokumen antara lain: Algoritma Porter *Stemming* for Bahasa Indonesia untuk *pre-processing text mining* berbasis metode *Market Basket Analysis*<sup>4</sup>, Sistem Penilaian Otomatis Jawaban *Essay* menggunakan Ontologi Moodle<sup>5</sup>, Implementasi modifikasi *Enhanced Confix Stripping Stemmer*.<sup>6</sup>

## METODE PENELITIAN

Tahapan awal sistem dimulai dengan proses *input* yang berisi kata-kata yang ingin dicari oleh pengguna. Sistem melakukan pengecekan tiap kata dan melakukan penghapusan kata yang tidak penting atau tepat untuk pencarian kata selanjutnya. Kemudian sistem akan memulai tahapan *parsing* yang diawali dengan *stemming* kata, pemisahan struktur kata antara imbuhan dengan kata dasarnya.

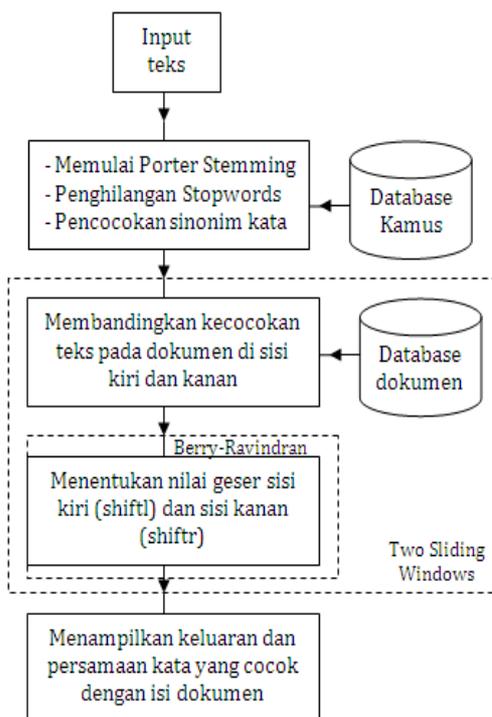
Pada saat proses *stemming* berlangsung, sistem mencocokkan inputan pengguna terhadap kata dasar yang sudah disimpan di *database*, data yang digunakan merujuk Kamus Besar Bahasa Indonesia (KBBI) yang berisi kumpulan kata-kata lengkap bahasa Indonesia dan makna persamaan kata merujuk kepada *Thesaurus* bahasa Indonesia. Penambahan persamaan kata yang relevan dengan makna kata tersebut baik kata tunggal maupun frasa untuk pencarian selanjutnya; pencarian kata atau sinonim kata yang sesuai dengan dokumen yang ada di *database*. Hasil berupa nama dan isi dari

dokumen yang sesuai dengan sinonim kata yang ingin dicari oleh pengguna. Rancangan arsitektur umum dari sistem dapat dilihat pada Gambar 1.

Perincian tiap proses pada arsitektur umum gambar 1 dijelaskan dengan tahapan sebagai berikut :

### 1. Penghilangan *Stopwords*

Kata-kata yang telah dibagi menjadi kumpulan *array*, yakni *arr\_kata()*, program akan memulai proses penghilangan kata-kata yang tidak relevan atau tidak tepat terutama tanda baca dan kata penghubung (*stopwords*). Program melakukan pendeteksian kata per kata berdasarkan *stopword* yang ada di *database*. Kemudian kata-kata tersebut dihapus berdasarkan kamus kata yang berisi kata-kata “tidak relevan” (*stopwords*).



**Gambar 1.** Arsitektur Umum

### 2. Porter Stemming

Setelah selesai proses *stopword*, maka dilanjutkan proses *stemming* menggunakan algoritma Porter. Pada penelitian ini, algoritma Porter dimodifikasi agar sesuai dengan imbuhan Bahasa Indonesia.<sup>7</sup> Aturan ditambahkan untuk

memberikan hasil yang maksimal dan mempermudah proses *stemming*. Berikut adalah aturan yang ditambahkan dalam algoritma Porter.<sup>8</sup>

- Kata dasar yang dilekati partikel infleksional yang tidak mempunyai imbuhan apapun. Contoh: masalah.
- Kata dasar yang dilekati partikel berprefiks yang tidak mempunyai imbuhan apapun. Contoh: menikah.
- Kata dasar yang dilekati kata ganti milik yang tidak mempunyai imbuhan apapun. Contoh: bangku.
- Kata dasar yang dilekati kata ganti milik berprefiks yang tidak mempunyai imbuhan apapun. Contoh: bersuku.
- Kata dasar yang dilekati prefix pertama yang tidak mempunyai imbuhan apapun. Contoh: median.
- Kata dasar yang dilekati prefix pertama bersufiks yang berarti kata dasar yang memiliki suku kata pertama awalan dan mempunyai akhiran. Contoh: terapan.
- Kata dasar yang dilekati prefix kedua yang tidak mempunyai imbuhan apapun. Contoh: percaya.
- Kata dasar yang dilekati prefix kedua bersufiks yang berarti kata dasar yang memiliki suku pertama awalan kedua dan mempunyai akhiran. Contoh: perasaan.
- Kata dasar yang dilekati sufiks yang tidak mempunyai imbuhan apapun. Contoh: pantai.

Algoritma Porter dimulai dengan langkah-

langkah antara lain :

- a. Kata yang di *stemming* mula-mula dicari di kamus data, jika ada kata dasar yang sesuai, maka algoritma berhenti, jika tidak ada, maka lanjutkan ke langkah b.
- b. Cek aturan tambahan yang bukan merupakan imbuhan Bahasa Indonesia, jika ada yang sesuai, maka algoritma berhenti. Jika tidak ada, maka lanjutkan ke langkah c.
- c. Hilangkan partikel (“-lah”, “-kah”, “-tah”, “-pun”).

- d. Hilangkan kata ganti kepemilikan (“-ku”, “-mu”, “-nya”).
- e. Hilangkan awalan pertama (“meng-”, “meny-”, “men-”, “mem-”, “me-”, “peng-”, “peny-”, “pen-”, “pem-”, “di-”, “ter-”, “ke-”). Jika tidak terdapat awalan pertama pada kata, maka lanjutkan ke langkah g. Jika ada, maka lanjutkan ke langkah f.
- f. Hilangkan awalan kedua (“ber-”, “bel-”, “be-”, “per-”, “pel-”, “pe-”). Algoritma akan berhenti jika sebelumnya sudah melalui langkah g dan masih tidak menemukan adanya awalan kedua pada kata. Algoritma akan menganggap kata yang dimasukkan sebagai kata dasar.
- g. Hilangkan akhiran (“-kan”, “-an”, “-i”). Jika tidak terdapat akhiran pada kata, maka lanjutkan ke langkah f.

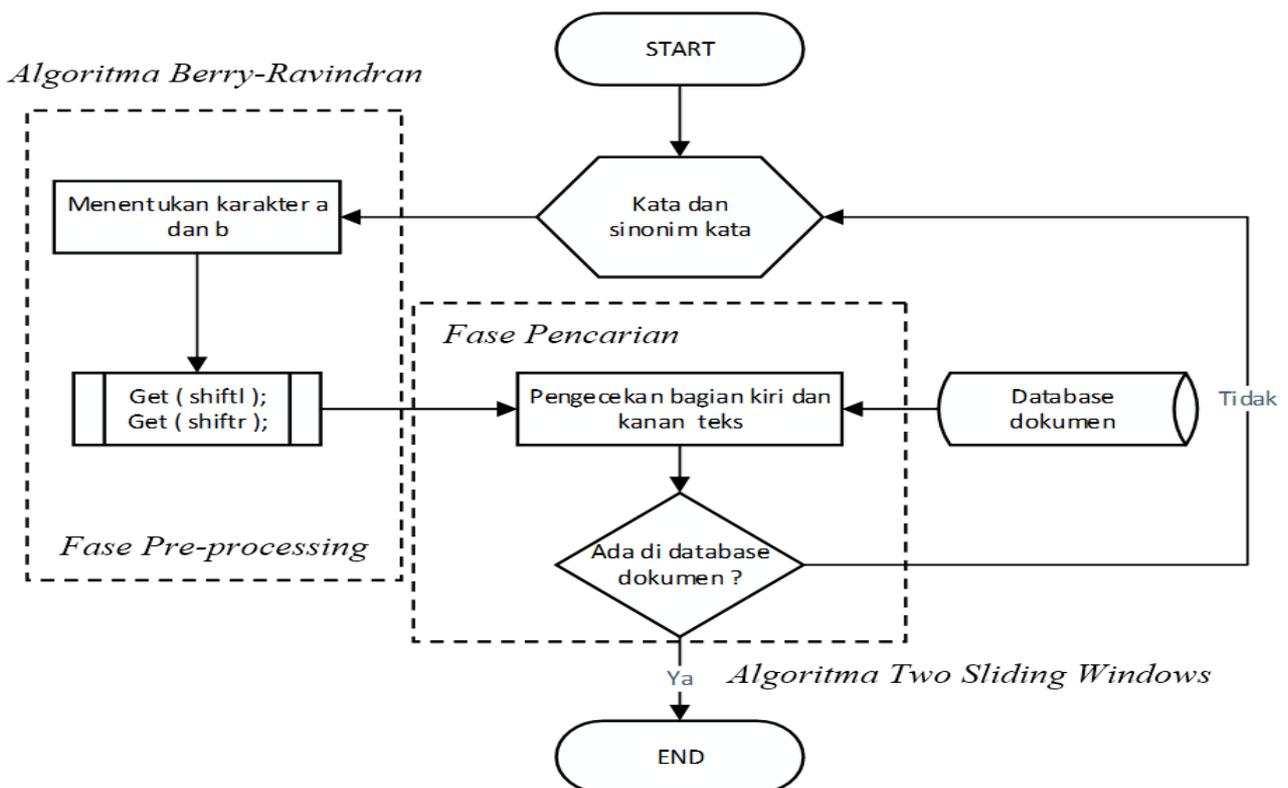
### 3. Pencocokan Kata dan Sinonim Kata

Selanjutnya mencari persamaan kata (sinonim) yang ada di dalam kamus sinonim kata. Tiap *arr\_kata[]* (kata hasil *stemming*) dilakukan *scanning* dan hasil pencocokan

persamaan kata akan dimuat ke dalam *array* baru, yaitu *arr\_Snmkata[]*. Apabila hasil pencocokan terdapat lebih dari satu makna kata, maka program tetap akan memuat sinonim kata tersebut ke dalam *arr\_Snmkata[]*.

### 4. Pencarian Teks dengan TSW

Pada tahapan pencarian teks digunakan Algoritma *Two Sliding Windows* (TSW) yaitu sebuah algoritma yang mempunyai keunggulan dalam fase pencarian yang mencakupi dua proses utama yaitu fase *pre-processing* dan fase pencarian. Pada dasarnya, algoritma TSW melakukan pencarian dari dua sisi (*windows*) yaitu sisi kiri (*left window*) dan sisi kanan (*right window*) secara paralel<sup>3</sup>. Fase *pre-processing* menggunakan algoritma *Berry-Ravindran* untuk menentukan nilai *shift* yang akan digunakan pada saat fase pencarian. Proses keseluruhan pencarian teks dapat dilihat pada gambar.



Gambar 2. Proses Pencarian Teks

### 5. Tahap Pre-Processing

Fase *pre-processing* menggunakan dua *array* yaitu *nextl* dan *nextl*. Nilai dari kedua *array* tersebut ditentukan oleh algoritma *bad character Berry Ravindran*. Nilai dari kedua *array* tersebut akan dimasukkan ke dalam masing-masing variabel *shifl* dan *shiftr*. Kata-kata dasar yang telah diproses akan melalui proses penentuan nilai kedua *shift* sebagai berikut :

Variabel *shifl*

Langkah-langkah pencarian nilai *shifl* dijelaskan sebagai berikut :

- Apabila pencarian dimulai pertama kalinya, maka nilai variabel *shifl* adalah 0. Jika tidak, maka nilai awal *shifl* =  $m+2$  dimana  $m$  merupakan jumlah pola (*pattern*) yang ingin dicari.
- Penetapan nilai  $a$  dan  $b$  yang merupakan dua karakter berurutan di hitung setelah posisi pola, antara lain :

$$\begin{aligned} a &= P[n-m-1] \\ b &= P[n-m-2] \end{aligned}$$

Dimana,  $P = Pattern$  ;  $m$  = banyak pola yang dicari ;  $n$  = nilai tengah dari banyak pola yang dicari

- Lakukan *loop* yang dimulai dari awal sampai akhir karakter yang sebanyak  $m-2$ .
- Pada saat terjadi *looping*, algoritma menghitung nilai *array nextl* yang akan dimasukkan kedalam variabel *shifl*. Apabila hasil *array nextl* lebih dari satu nilai, maka algoritma memilih nilai terkecil dari hasil tersebut. Proses perhitungan nilai *shifl* dapat dilihat pada gambar 3.

$$\text{shifl}(a,b) = \min \left\{ \begin{array}{ll} 1 & \text{if } P[m-1] = a \\ m-i & \text{if } P[i][i+1] = ab \\ m+1 & \text{if } P[0] = b \\ m+2 & \text{Otherwise} \end{array} \right\}$$

**Gambar 3.** Aturan Operasi Hitung Nilai *shift* kiri

Variabel *shiftr*

Langkah-langkah penentuan nilai *shiftr* dijelaskan sebagai berikut :

- Apabila pencarian dimulai pertama kalinya, maka nilai variabel *shifl* adalah 0. Jika tidak,

maka nilai awal *shiftr* =  $m+2$  dimana  $m$  merupakan jumlah pola (*pattern*) yang ingin dicari.

- Penetapan nilai  $a$  dan  $b$  seperti pada gambar 3.6 yang merupakan dua karakter berurutan di hitung setelah pola tersebut, antara lain :

$$\begin{aligned} a &= P[m+1] \\ b &= P[m+2] \end{aligned}$$

Dimana,  $P = Pattern$  ;  $m$  = banyak pola yang dicari ;  $n$  = nilai tengah dari banyak pola yang dicari.

- Lakukan *loop* yang dimulai dari awal sampai akhir karakter sebanyak  $m-2$ .
- Pada saat terjadi *looping*, algoritma menghitung nilai *array nextl* yang akan dimasukkan kedalam variabel *shifl*. Apabila hasil *array nextl* lebih dari satu nilai, maka algoritma memilih nilai terkecil dari hasil tersebut. Proses perhitungan nilai *shifl* pada gambar 4.

$$\text{shiftr}(a,b) = \min \left\{ \begin{array}{ll} m+1 & \text{if } P[m-1] = a \\ m - ((m-2) - i) & \text{if } P[i][i+1] = ab \\ 1 & \text{if } P[0] = b \\ m+2 & \text{Otherwise} \end{array} \right\}$$

**Gambar 4.** Aturan Operasi Hitung Nilai *shift* kanan

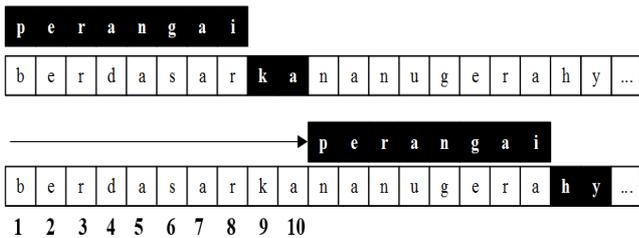
### 6. Algoritma Two Sliding Windows

Apabila nilai *shifl* dan *shiftr* sudah didapatkan dari fase *pre-processing*, maka program akan memulai fase pencarian menggunakan algoritma *Two Sliding Windows*. Proses pencarian kata yang sesuai dengan dokumen adalah sebagai berikut :

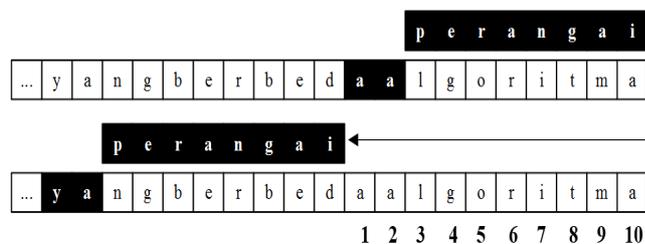
- Pencocokan *string* dari sebelah kiri teks dimulai dari posisi akhir string sampai posisi awal string. Jika terjadi kecocokan, maka algoritma akan berhenti. Jika tidak, maka lanjutkan ke langkah 3.
- Pencocokan *string* dari sebelah kanan teks dimulai dari posisi awal string sampai posisi akhir string. Jika terjadi kecocokan, maka algoritma akan berhenti. Jika tidak, maka lanjutkan ke langkah 3.

- Penentuan nilai *shift* kiri dan kanan dari algoritma *Berry Ravindran*.
- Menggeser *pattern* sesuai nilai *shift* yang didapat dari langkah 3. Pada bagian kiri akan menggeser ke sebelah kanan dan bagian kanan menggeser ke sebelah kiri. Ilustrasi sederhana proses algoritma *Two Sliding Windows* dapat dilihat pada gambar 5.

Bagian kiri teks



Bagian kanan teks



**Gambar 5.** Proses Algoritma *Two Sliding Windows*

- Kembali ke langkah 1 untuk melakukan pencocokan *string* dan posisi indeks pencarian yang baru ditentukan dari nilai *shift* pada proses *pre-processing* algoritma BR. Pencocokan terus dilakukan hingga *pattern* berada ditengah teks ( $n/2$ ). Apabila masih tidak terdapat kecocokan dengan *pattern* sampai di tengah teks, maka algoritma memberikan hasil kosong.

## HASIL DAN PEMBAHASAN

### Implementasi Sistem

Implementasi perancangan sistem menggunakan bahasa pemrograman PHP dan *database* MySQL. Dalam tahap ini juga dibahas hasil perancangan dan pengujian pada sistem.

#### 1. Halaman pencarian dokumen

Pada halaman utama terdapat *textbox* dimana pengguna dapat memasukkan input yang

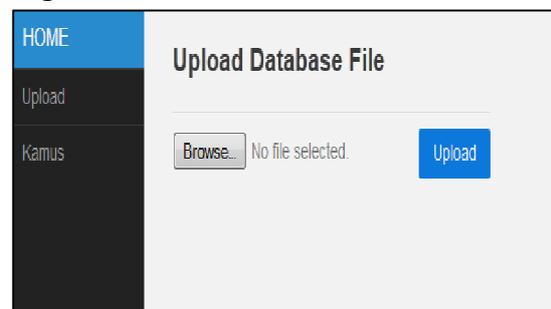
ingin dicari dan tombol “Proses” untuk melanjutkan proses pencarian. Sebelum melakukan “Proses”, ada tiga parameter yaitu “*stopword*”, “*stemming*”, dan “*sinonim*” untuk melakukan hasil pengujian dengan parameter yang berbeda. Tampilan halaman pencarian dokumen dapat dilihat pada gambar 6.



**Gambar 6.** Halaman pencarian dokumen

#### 2. Halaman *upload*

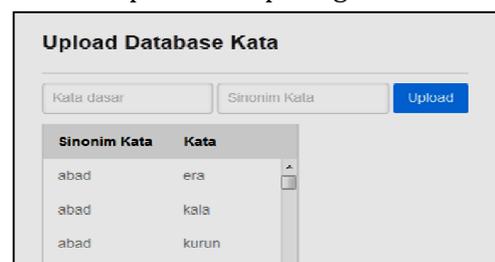
Pada menu *upload* terdapat tombol untuk menambahkan *file* yang belum ada di dalam *database*. Tampilan halaman *upload* dapat dilihat pada gambar 7.



**Gambar 7.** Halaman *upload*

#### 3. Halaman kamus kata

Halaman ini bertujuan untuk menambahkan kata dan sinonim kata yang tidak terdapat pada kamus kata. Tampilan halaman kamus kata dapat dilihat pada gambar 8.



**Gambar 8.** Halaman kamus kata

## Tahapan Operasi Aplikasi

### 1. Pencarian Dokumen

Pada halaman utama ada tiga buah *checkbox* yaitu “*stemming*”, “*stopword*”, “*sinonim*”. Masing-masing mempunyai fungsi tersendiri dalam melakukan proses pengolahan kata yang akan dijadikan sebagai kata kunci (*keyword*) pada pencarian dokumen. Sebagai contoh, apabila pengguna memilih “*stemming*” dan “*stopword*”, maka sistem hanya menjalankan dua tahapan proses pengolahan kata dan tidak melaksanakan proses “*sinonim*”. Dengan kata lain, hasil pencarian hanya memunculkan dokumen terkait tanpa menelusuri persamaan kata dari *input*-an kata.

Langkah awal dalam mencari persamaan kata dalam dokumen adalah menuliskan kata-kata yang dicari ke dalam *textbox* dan mencentang semua *checkbox* yang terdapat pada gambar 6 agar hasil pencarian lebih akurat. Tampilan pada saat menuliskan kata-kata pada *textbox* pada gambar 9.

Gambar 9. Form

pencarian dokumen

Setelah memasukan input kedalam *textbox*, program akan menampilkan hasil yang terkait dengan pilihan parameter yang disediakan. *Output* ditampilkan di bagian bawah *input* dan program menampilkan hasil setiap langkah pengolahan kata yang telah di *input* oleh pengguna serta hasil pencarian berupa judul dokumen dan beberapa baris di dalam isi dokumen yang serupa dengan kata yang di *input* oleh pengguna. Hasil pencarian sinonim kata dapat dilihat pada gambar 10.

Gambar 10. Hasil pencarian sinonim kata pada dokumen

### 4. Pengujian Sistem

Pada bagian akan ditunjukkan hasil pengujian berdasarkan tiga parameter yang disediakan oleh program. Tahapan proses pengujian dimulai dengan memasukkan berbagai jumlah kata sebagai *keyword* pada hasil pencarian. Jumlah variasi kata dimulai dari 1 kata sampai dengan 5 kata yang berupa kalimat baku yang jarang diucapkan pada bahasa Indonesia. Hasil pengujian dengan kata kunci “algoritma pencocokan pola kalimat baku” ditampilkan pada tabel 1.

Tabel 1. Pengujian terhadap variasi jumlah kata pada kalimat baku.

No	Jumlah kata	Jumlah dokumen hasil pencarian	Jumlah sinonim kata yang terdeteksi	Tingkat Keakuratan
1	1	25	1	95,83%
2	2	52	1	98,07%
3	3	69	1	98,55%
4	4	94	1	98,93%
5	5	232	4	98,27%

Hasil pengujian pada tabel 1 menunjukkan semakin banyak kata yang ingin dicari oleh pengguna semakin banyak dokumen yang cocok dengan kata kunci tersebut. Hal ini dipengaruhi oleh pilihan pencarian yang lebih banyak sehingga jumlah dokumen yang terdeteksi juga lebih banyak. Hasil pengujian juga menunjukkan isi dokumen yang sesuai dengan kata kunci tidak mencapai 100% tetapi hanya 97,93% karena adanya perbedaan kata imbuhan baik serapan maupun sisipan yang menimbulkan makna berbeda. Pengujian selanjutnya menggunakan kata kunci “Para pelaku pasar berjibaku redakan polemik krisis ekonomi” sehingga didapat hasil yang terlihat pada gambar 11.

<b>Input</b>	Para pelaku pasar berjibaku redakan polemik kris
<b>Stopword</b>	pelaku pasar berjibaku redakan polemik krisis ek
<b>Stemming</b>	pelaku pasar jibaku reda polemik krisis ekonomi
<b>Sinonim</b>	aktor nekat surut reda perdebatan berhenti daru
Jumlah hasil pencarian : 161	

**Gambar 11.** Hasil pengujian dengan kata kunci “para pelaku pasar berjibaku redakan polemik krisis ekonomi”

Hasil pengujian dengan menggunakan algoritma TSW dapat dilihat pada tabel 2. dan tabel 3.

**Tabel 2.** Hasil pencocokan dengan kata kunci “pelaku, pasar, jibaku, reda, polemik, krisis, ekonomi”.

<b>Keyword</b>	<b>Judul dokumen yang sesuai</b>	<b>Letak kata (kiri)</b>	<b>Letak Kata (kanan)</b>
<i>pelaku</i>	Landasan Teori Pengajaran Bahasa Indonesia sebagai bahasa asing	-	256757
	Perjuangan Memperbaiki Citra Guru Indonesia	5647	-
	Analisis Mendelian 1	4984	-
	Etika Penulisan Ilmiah	19248	-
	Kebijakan Pemidanaan dalam Tindak Pidana Psicotropika	2920	-
	Landasan Teori Pengajaran Bahasa Indonesia sebagai	-	277181

	bahasa asing		
	Perjuangan memperbaiki Citra Guru Indonesia	-	33700
	Metode Adopsi E-Business oleh Pengusaha Kecil	-	45600
	Etika Penulisan Ilmiah	7129	-
	Kebijakan Pemidanaan dalam Tindak Pidana Psicotropika	-	101393
<i>jibaku</i>	-	-	-
	Kebijakan Pemidanaan dalam Tindak Pidana Psicotropika		95089
	Monitoring dan Analisis Kualitas Layanan Trafik Kamera CCTV	4730	-
	Dialog dengan Tuhan	88399	-
<i>polemik</i>	Etika Penulisan Ilmiah	-	37754
	Proud to be Confucian	45822	-
	Perjuangan Memperbaiki Citra Guru Indonesia	1875	17301
	Landasan Teori Pengajaran Bahasa Indonesia sebagai bahasa asing	46110	254368
	Proud to be Confucian	85074	-
<i>Ekonomi</i>	Perjuangan Memperbaiki Citra Guru Indonesia	-	25878
	Kebijakan Pemidanaan dalam Tindak Pidana Psicotropika		
	Landasan Teori Pengajaran Bahasa Indonesia sebagai bahasa asing	-	222270
	Mengembangkan Kurikulum BIPA	-	18594
	Monitoring dan Analisis Kualitas Layanan Trafik Kamera CCTV	-	44101
	Pilihan Kata (Diksi) template	10860	-

Dari tabel 2 dapat dijelaskan bahwa kata “pelaku” ditemukan pada 5 buah dokumen yaitu dokumen dengan judul “Landasan Teori Pengajaran Bahasa Indonesia sebagai bahasa asing” ditemukan pada letak kata sisi kanan dokumen pada karakter ke 256757. Dokumen “Perjuangan Memperbaiki Citra Guru Indonesia” ditemukan pada letak kata sisi kiri dokumen pada karakter ke 5647. Dokumen “Analisis Mendelian 1” ditemukan pada letak kata sisi kiri dokumen pada karakter ke 4984. Demikian seterusnya untuk dokumen-dokumen lainnya.

**Tabel 3.** Hasil pencocokan dengan sinonim kata kunci “aktor, nekat, surut,reda, perdebatan, berhenti, darurat,krisis”

Keyword	Judul dokumen yang sesuai	Letak kata (kiri)	Letak kata (kanan)
Aktor	Perjuangan Memperbaiki Citra Guru Indonesia	-	25878
	Analisis Mendelian 1	-	45600
	Etika Penulisan Ilmiah	7129	-
	Kebijakan Pemidanaan dalam Tindak Pidana Psikotropika	-	119064
	Landasan Teori Pengajaran Bahasa Indonesia sebagai bahasa asing	-	222270
	nekat	-	-
surut	Ekosistem Sawah	1003	-
	perdebatan Proud to be Confusian	36258	-
berhenti	Pendekatan Simantik menurut Toshiko Izutsu	-	88023
	Analisis Mendelian 1	-	45523
darurat	Landasan Teori Pengajaran Bahasa Indonesia sebagai bahasa asing	46110	-
	Dialog dengan Tuhan	-	477606
	Pembentukan Daftar Kunci untuk Pengklasifikasian Opini	37758	-
	Proud to be Confusian	-	210809
	Peraturan Komisi Informasi	-	16025
	Anatomi dan Fisiologi Sistem Endokrin	1134	-

Pada hasil pengujian diatas dapat disimpulkan bahwa proses pengujian sistem berhasil dilakukan. Dokumen-dokumen yang muncul pada hasil pengujian memiliki hubungan persamaan kata pada *keyword* di dalam isi dokumen dan menunjukkan pencocokan letak kata yang bervariasi sesuai dengan *keyword* mulai dari sisi kiri sampai sisi kanan pada isi dokumen.

## SIMPULAN

Algoritma *Two Sliding Windows* dapat diimplementasikan untuk mempercepat pencarian dokumen. Kata-kata untuk pencarian dokumen yang jarang digunakan dalam bahasa Indonesia dapat dipahami dengan menemukan sinonim katanya. Pada saat proses pencarian,

kata pertama yang ditemukan sesuai dengan kata kunci akan dianggap sebagai hasil pencarian dan proses algoritma berhenti. Semakin banyak kata dan sinonim kata kunci yang ingin dicari oleh pengguna (*user*), maka semakin banyak dokumen yang terdeteksi pada hasil pencarian.

## DAFTAR PUSTAKA

- <sup>1</sup>Aminuddin. 2008. *Semantik Pengantar Studi Tentang Makna*. Bandung: SinarBaru Algesindo.
- <sup>2</sup>Chaer, Abdul. 2009. *Pengantar Semantik Bahasa Indonesia*. Jakarta: RinekaCipta.
- <sup>3</sup>Hudaib, A., Al-Khalid, R., Suleiman, D., Itriq, M., & Al-Alnani, A. 2008. A Fast Pattern Matching Algorithm with Two Sliding Windows. *Journal of Computer Science*, 4(5): 393-401.
- <sup>4</sup>Budhi, Gregorius S., Gunawan I., Yuwono F. 2006. *Algoritma Porter Stemmer For Bahasa Indonesia untuk Pre-Processing Text Mining Berbasis Metode Market Basket Analysis*. Paper, Universitas Kristen Petra.
- <sup>5</sup>Fidaursiah A.B., Siahaan D. O., Yuhana U. L., Kita T. 2008. *Sistem Penilaian Otomatis Jawaban Essay menggunakan Ontologi pada Moodle*. Skripsi, Surabaya: Institut Teknologi Sepuluh November.
- <sup>6</sup>Tahitoe, A.D., Purwitasari D., 2010. *Implementasi Modifikasi Enhanced Confix Stripping Stemmer untuk Bahasa Indonesia dengan Metode Corpus Based Stemming*. Skripsi, Surabaya: Institut Teknologi Sepuluh Nopember.
- <sup>7</sup>Tala, Fadillah Z. 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Master of Logic Project. Institute for Logic, Language and Computation. Universiteit van Amsterdam. The Netherlands.
- <sup>8</sup>ChoirohU., 2011. *Pembuatan Aplikasi Penerjemah Kalimat Tunggal Bahasa Indonesia Kedalam Kalimat BahasaJawa Berbasis Kamus*. Skripsi, Malang: Universitas Islam Negeri (UIN) Maulana Malik Ibrahim.

