SEGMENTASI DOKUMEN BAHASA INDONESIA MENGGUNAKAN TEXT TILING

Yunianita Rahmawati[1)], Gunawan[2)]
[1] Informatika, Universitas Muhammadiyah Sidoarjo
email: yunianita@umsida.ac.id
[2] Teknologi Informasi,Institut Sekolah Tinggi Teknik SUrabaya
email: gunawan@stts.ac.id

Article history
Received August 29, 2021
Revised Sept 23, 2021
Accepted Sept 30, 2021
Available online Oct 11, 2021

Keywords
text tiling, segmentation, multiparagraph segmentation

Abstract

Text tiling aims to split long documents into multiple related paragraphs. In this study, the documents are used as data by omitting the reading format as inputs in the segmentation. Text tiling method has three stages, namely tokenisation, determination of similarity, and the introduction of limits. In this study, the results of the segmentation algorithm using tiling text has not yet reached the objective. This is because the segmentation of the document is strongly influenced by a common word file, the determined number of tokens in a token-sequence, and the determination of the number token-sequence within a block.Tthe writing of a word and text tiling algorithm is very sensitive to the reading format, such as titles and subtitles, so that the reading format must be removed to have the body of the text only. Segmentation results increased after the trials. From the experiment of the 15 reading segmentation results show that an accuracy of precision is 59,3% and of recall is 80%. These trials used 4140 common words. The total coefficient score for similarity is 5, the number of tokens in a token-sequence is 20, and the number of token-sequence within a block is 3.

Riwayat
Diterima 29 Agustus 2021
Revisi 23 Sept 2021
Disetujui 30 Sept 2021
Terbit 11 Okt 2021

Kata Kunci
text tiling, segmentasi, segmentasi multiparagraf

Abstrak

Text tiling bertujuan untuk membagi dokumen panjang menjadi unit multi paragraf yang berhubungan. Dalam penelitian ini, dokumen dijadikan data latih dalam uji coba dan dokumen tersebut dihilangkan format bacaan untuk dijadikan input dalam aplikasi segmentasi ini. Metode text tiling ini mempunyai tiga tahapan yaitu tokenisasi, penentuan kemiripan, dan pengenalan batas. Pada penelitian ini, hasil dari segmentasi menggunakan algoritma text tiling belum mencapai hipotesa, hal ini dikarenakan segmentasi dokumen sangat dipengaruhi oleh file common word, penentuan jumlah token dalam token-sequence, dan penentuan jumlah token-sequence dalam satu blok. Selain ketiga hal tersebut, dan benar tidaknya penulisan suatu kata dan algoritma text tiling ini sangat sensitive dengan format bacaan, seperti judul dan sub judul, sehingga format bacaan tersebut harus dihapus hingga meninggalkan badan teks saja. Dari berbagai hasil uji coba segmentasi bacaan yang dilakukan pada data sebanyak 15 bacaan mendapatkan hasil segmentasi dengan nilai precision 59,3% dan recall 80%. Hasil segmentasi dari uji coba tersebut menggunakan jumlah common word 4140, total koefisien similarity score sebesar 5, jumlah token dalam token-sequence sebesar 20, dan jumlah token-sequence dalam blok sebesar 3.

PENDAHULUAN

Perkembangan teknologi di bidang komputasi dan telekomunikasi telah memberi kemudahan kepada setiap orang untuk menyampaikan dan mendapatkan informasi dengan bermacam-macam bahasa. Kemudahan ini menyebabkan informasi menjadi semakin banyak dan beragam. Informasi dapat berupa berita, dokumen, surat, laporan penelitian, cerita, data keuangan, dan lain-lain. Kebanyakan dari informasi tersebut terdiri dari bacaan yang panjang (dokumen panjang). Dokumen panjang biasanya terdiri dari bermacam-macam sub topik / sub bab / sub judul, berbeda dengan abstraksi yang ringkas dan padat informasi. Struktur sub topik ditandai dalam teks-teks teknis dengan judul dan sub judul. Menurut Brown dan Yule (1983, 140) dalam Hearst , menyatakan bahwa pembagian semacam ini adalah salah satu hal yang paling mendasar dalam wacana. Namun, banyak dokumen panjang terdiri dari urutan panjang paragraf dengan sangat sedikit demarkasi (pemisah) struktural. Sehingga kegiatan segmentasi dapat berguna untuk pemisahan dokumen yang terdiri atas bebepara paragraf menjadi unit paragraf yang saling berkaitan berdasarkan subbabnya.

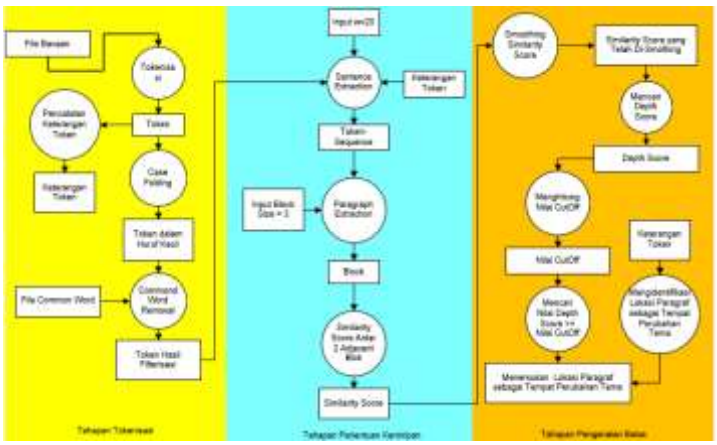
Menurut Hearst (1997), alasan menggunakan multi paragraf yaitu bahwa sebagian besar paragraf memiliki jenis tertentu dari struktur yang terbentuk, lengkap dengan kalimat topik dan ringkasan kalimat. Dalam teks dunia nyata, harapan ini sering tidak terpenuhi. Tetapi bahkan jika paragraf ditulis dalam cara dikemas mandiri, diskusi subtopik tertentu dapat menjangkau beberapa paragraf, dengan hanya nuansa yang berbeda yang dibahas pada paragraf yang terdiri diskusi. Tanda paragraf tidak selalu digunakan untuk menunjukkan perubahan dalam diskusi, tetapi kadang-kadang paragraf digunakan hanya untuk memecah penampilan fisik teks dalam rangka untuk membantu membaca (Stark 1988). Contoh tata letak kolom teks di banyak surat kabar (Longacre 1979). Brown dan Yule (1983, 95-96) mencatat bahwa bergenre teks memiliki pengaruh yang kuat pada peran tanda paragraf, dan bahwa tanda berbeda untuk bahasa yang berbeda. Hinds (1979, 137) juga menunjukkan bahwa jenis wacana yang berbeda memiliki prinsip pengorganisasian yang berbeda.

Tujuan dari metode text tiling yaitu metode yang digunakan untuk partisi dokumen teks panjang menjadi unit multi paragraf yang koheren / berhubungan, unit wacana yang mencerminkan struktur subtopik dari teks. TextTiling mendekati struktur subtopik dokumen dengan menggunakan pola konektivitas leksikal (makna dasar) untuk menemukan sub diskusi koheren. “Text Tiling” dimaksudkan untuk mencerminkan pola subtopik yang terkandung dalam dokumen teks panjang. Pendekatan ini menggunakan analisis leksikal kuantitatif dan mengklasifikasikan sehubungan dengan basis pengetahuan umum. “Tiles” mempunyai penilaian yang baik jika dibandingkan dengan penilaian manusia pada batas-batas subtopik.

Berdasarkan latar belakang di atas dan kajian pustaka yang sudah dilakukan, maka penulis tertarik untuk mempelajari lebih lanjut pada bidang TextTiling untuk melakukan penelitian pada segmentasi multi paragraf untuk dokumen bahasa Indonesia.

METODE PENELITIAN

Text tiling adalah sebuah metode untuk membagi dokumen text (seluruhnya) menjadi beberapa unit paragraf yang saling berkaitan. Text tiling memperkirakan struktur sub topik dari sebuah dokumen dengan menggunakan pola dari hubungan lexical untuk menemukan sub diskusi yang berkaitan. Rancangan dari “tiles” digunakan untuk mencerminkan pola dari sub topik yang terdapat dalam sebuah teks bacaan. Pendekatan menggunakan analisa leksikal secara kuantitatif untuk menunjukkan pembagian dari dokumen.



Gambar 1 : Diagram Arsitektur Algoritma Text Tiling

Inputan dari algoritma text tiling adalah sebuah teks bacaan yang panjang. Text tiling menggunakan pendekatan hitungan untuk

membagi teks bacaan tertulis menjadi unit tulisan yang tidak overlap dan berkesinambungan dengan pola sub topik pada sebuah teks bacaan. Sebuah sub topik utama dapat terbagi atas beberapa sub topik. Dua buah kata dikatakan dalam sub topik yang sama apabila keduanya berada dalam *passage* yang sama. Istilah *passage* tidak didefinisikan dengan baik. Asumsi sederhananya adalah setiap paragraf adalah sebuah *passage* dan setiap *passage* adalah sebuah paragraf. *Passage* secara sederhana dapat ditentukan secara otomatis dengan membagi teks ke dalam sejumlah blok yang memiliki besar yang sama.

Algoritma text tiling menggunakan pengulangan istilah sebagai indikator *lexical cohesion* (dua buah item yang sama secara leksikal atau berkaitan erat). Algoritma ini membandingkan setiap pasang blok teks yang bersebelahan tergantung pada kemiripan leksikal dari ukuran jendela yang diberikan. Algoritma ini menyimpulkan bahwa semakin mirip kedua teks blok, maka sub topik mempunyai arah pembicaraan / tema yang sama dan kebalikannya, jika kedua blok teks yang berbatasan tidak mirip, maka hal ini akan menyatakan secara tidak langsung adanya perubahan arah pembicaraan / tema.

Tahapan-tahapan yang harus dilalui algoritma text tiling untuk menemukan lokasi perubahan tema, secara garis besar dapat dilihat pada gambar 1. Pada gambar 1 di atas menjelaskan tentang diagram alir text tiling. Berikut ini adalah tiga tahapan yang harus dilalui algoritma text tiling untuk menemukan lokasi perubahan tema yaitu tahapan tokenisasi, tahapan penentuan kemiripan, dan tahapan pengenalan batas.

2.1. Tahapan Tokenisasi

Algoritma text tiling memerlukan dua buah file sebagai inputan yaitu file teks bacaan dan file *stoplist / common word*. File teks bacaan adalah teks bacaan yang akan dicari struktur sub topiknya. File *stoplist / common word* adalah suatu file yang berisi kata-kata yang sering muncul dalam teks, tetapi tidak memberikan pengaruh terlalu banyak terhadap isi teks. Pada tahapan tokenisasi (*tokenization*) terdapat beberapa proses yaitu proses tokenisasi, proses pencatatan keterangan token, proses *case folding*, dan proses *commond word removal*. Output dari tahapan tokenisasi adalah keterangan token dan token yang tidak termasuk dalam *file common word*. Kedua output ini akan digunakan pada tahapan selanjutnya, tahapan kedua yaitu tahapan penentuan kemiripan dan tahapan ketiga yaitu pengenalan batas.

Pada algoritma text tiling terdapat istilah baru, istilah "kata" diubah menjadi token / text token, istilah "kalimat" diubah menjadi token-sequence, dan istilah "paragraf" diubah menjadi blok. Pada proses tokenisasi terdapat proses membagi teks input menjadi token per token dan menghasilkan token. Pada proses pencatatan keterangan token dilakukan pencatatan keterangan dari suatu token dan menghasilkan keterangan token. Sebuah token mempunyai keterangan no baris, no kolom, no paragraf, nilai token, no token-sequence, jenis token, dan no kata dalam bacaan. Pada proses *case folding* dilakukan konversi huruf menjadi huruf kecil (*lowercase*) dan menghasilkan token dalam huruf kecil. Pada proses *commond word removal* dilakukan pencocokan seluruh token dalam bacaan dengan *file commond word* dan menghasilkan token hasil filterisasi. Jika token termasuk dalam *file commond word* maka token itu tidak diteruskan ke proses selanjutnya. Sebaliknya, jika token tidak termasuk dalam *file commond word* maka token tersebut dilanjutkan ke proses selanjutnya.

2.2. Tahapan Penentuan Kemiripan

Input dari tahapan penentuan kemiripan (*similarity determination*) ini yaitu token yang tidak termasuk dalam file *commond word* dan keterangan dari token yang didapat dari proses tokenisasi. Sedangkan output dari tahapan kedua ini yaitu similarity score untuk setiap celah token-sequence. Pada tahapan penentuan kemiripan ini terdapat beberapa proses yaitu proses *sentence extraction, paragraph extraction,* dan similarity score antar 2 *adjacent block.*

Pada proses *sentence extraction* dilakukan pembentukan *pseudosentences* (kalimat semu) dari teks bacaan dengan cara mengelompokkan sejumlah token. Istilah pengelompokan token untuk membentuk *pseudosentence* ini akan diubah menjadi token-sequence agar memudahkan pembahasan selanjutnya. Untuk membentuk sebuah token-sequence, token dikelompokkan sebesar w (sebuah parameter algoritma) yang telah ditentukan untuk menghindari masalah normalisasi. Pada prakteknya, pemberian nilai w sebesar 20 akan berdampak lebih baik untuk banyak teks. Hal ini berarti sejumlah token berjumlah 20 token

dikelompokkan menjadi satu membentuk sebuah token-sequence atau setiap token-sequence berisi 20 token. Sebuah bacaan memiliki jumlah token yang sama untuk setiap token-sequence. Sejumlah token yang tersisa dari hasil penyaringan dan informasi mengenai token tersebut yang berasal dari tahapan sebelumnya yaitu tahapan tokenisasi. Hasil dari proses *sentence extraction* adalah beberapa token-sequence dari suatu bacaan.

Pada proses *paragraph extraction* dilakukan pembentukan *pseudoparagraph* (paragraf semu) dengan cara mengelompokkan sejumlah token-sequence. Istilah pengelompokan token-sequence ini akan diubah menjadi blok agar memudahkan pembahasan selanjutnya. Untuk membentuk sebuah blok, token-sequence dikelompokkan sebesar *blocksize* atau k (sebuah parameter algoritma yang telah ditentukan) yaitu jumlah dari token-sequence yang dikelompokkan menjadi sebuah blok yang akan digunakan dalam proses perhitungan kemiripan. Nilai *blocksize* atau k merupakan nilai rata-rata dari panjang paragraf (dalam token-sequence). Dalam prakteknya, nilai k sebesar 6 bekerja sangat baik pada kebanyakan teks. Hal ini berarti sejumlah token-sequence berjumlah 6 token-sequence dikelompokkan menjadi satu membentuk sebuah blok atau setiap blok berisi 6 token-sequence. Paragraf yang sesungguhnya tidak digunakan karena panjangnya sangat tidak beraturan, mengarah pada perbandingan (*comparison*) yang tidak seimbang. Hasil dari proses *paragraph extraction* adalah dua buah blok yang tersusun dari beberapa token-sequence yang berbeda-beda.

Untuk membandingkan 2 blok yang bersebelahan dengan nilai k sebesar 6 yang berarti bahwa 6 buah token-sequence yang dikelompokkan ke dalam 2 buah blok yang bersebelahan dengan *blocksize* yang masing-masing berukuran 3 dan masing-masing token-sequence berisi 20 token. Pada proses ini menggunakan pendekatan *moving window*. *Moving window* ini dilakukan dengan cara menggeser kedua blok satu token-sequence ke kanan pada setiap perhitungan celah token-sequence. Misalnya, saat menghitung celah token-sequence 3 maka blok yang dibandingkan yaitu blok 1 yang berisi token-sequence 1,2,3 dan blok 2 yang berisi 4,5,6. Kemudian saat menghitung celah token-sequence 4 maka blok yang dibandingkan yaitu blok 1 yang berisi token-sequence 2,3,4 dan blok 2 yang berisi 5,6,7 dan seterusnya hingga token-sequence berakhir.

Pada proses similarity score antar 2 blok yang saling bersebelahan dilakukan proses menyamakan blok yang bersebelahan (*adjacent*) dari token-sequence untuk kemiripan seluruh leksikal. Proses ini menghasilkan nilai similarity yang akan digunakan pada tahapan ketiga yaitu tahapan pengenalan batas. Rumus untuk proses similarity score yaitu :

$$Sim(b_1, b_2) = \frac{\sum_t w_{t1b1} w_{t1b2}}{\sqrt{\sum_t w_{t1b1}^2 \sum_{i=1}^{n} w_{t1b2}^2}}$$

Dimana t digunakan untuk mewakili sebuah token yang ada dalam blok. Variabel $w_{t1}b_1$ dalam perhitungan celah token-sequence i adalah jumlah token t yang muncul dalam blok b1 (token-sequence i – k + 1 hingga token-sequence i), sedangkan variabel $w_{t1}b_2$ dalam perhitungan celah token-sequence i adalah jumlah token t yang muncul dalam blok b2 (token-sequence i + 1 hingga token-sequence i + k).

2.3. Tahapan Pengenalan Batas

Inputan dari tahapan pengenalan batas (*boundary identification*) yaitu nilai similarity yang didapatkan dari proses tahapan kedua yaitu tahapan penentuan kemiripan dan keterangan token yang didapatkan dari proses tahapan pertama yaitu tahapan tokenisasi. Output dari tahapan pengenalan batas yaitu menemukan lokasi paragraf sebagai tempat perubahan tema/subbab pada teks bacaan. Terdapat beberapa proses pada tahapan pengenalan batas yaitu proses *smoothing similarity score*, proses mencari depth score, proses menghitung nilai cutoff, mencari nilai depth score yang lebih besar atau sama dengan nilai cutoff, dan proses mengidentifikasi lokasi paragraf sebagai tempat terjadinya perubahan tema.

Pada proses smoothing similarity score dilakukan proses smoothing terhadap seluruh nilai similarity untuk setiap celah token-sequence. Proses smoothing dilakukan agar similarity score pada sebuah celah token-sequence tidak terpaut terlalu jauh dengan similarity score pada celah token-sequence yang lainnya. Hasil dari proses *smoothing similarity score* yaitu nilai similarity yang telah di-smoothing pada setiap celah token-sequence. Langkah-langkah proses *smoothing similarity score* yang harus dilakukan untuk

setiap celah token-sequence i dengan window size w+1 adalah:

- Mencari similarity score yang berada w/2 token-sequence di sebelah kiri celah token-sequence i.
- Mencari similarity score yang berada w/2 token-sequence di sebelah kanan celah token-sequence i.
- Mencari similarity score dari celah token-sequence i.
- Mencari nilai rata-rata dari ketiga nilai di atas dan menyimpan nilai tersebut pada celah token-sequence i.
- Mengulang prosedur di atas sebanyak n kali.

Dalam prakteknya, untuk sebagian besar teks yang telah diperiksa, satu putaran *average smoothing* dengan window size 3 adalah yang terbaik.

Pada proses mencari nilai depth score dilakukan pencarian nilai depth score, yang didapatkan dari perhitungan kelandaian nilai similarity yang telah di-smoothing pada sebuah celah token-sequence terhadap nilai similarity yang telah di-smoothing pada kedua celah token-sequence yang berada pada kanan dan kirinya. Selanjutnya depth score yang telah didapatkan itu diurutkan mulai dari depth score terkecil hingga depth score terbesar (*ascending*). Hasil dari proses pencarian depth score yaitu depth score. Langkah-langkah untuk mencari depth score yaitu :

1. Mengamati similarity score yang telah melalui tahap smoothing pada seluruh celah token-sequence. Lakukan pengamatan pada similarity score selama nilainya terus meningkat. Jika telah menemukan nilai yang paling tinggi maka nilai tersebut dinamakan nilai puncak atau puncak tertinggi dari suatu lembah. Nilai-nilai sebelum puncak dianggap sebagai bagian dari sebelah kiri puncak. Setelah menemukan nilai puncak, lakukan pengamatan lagi terhadap similarity score yang berada setelah nilai similarity score puncak selama nilainya menurun. Nilai-nilai yang berada setelah nilai puncak dinamakan sebagai bagian dari sebelah kanan lembah. Dari proses tersebut dapat menghasilkan satu lembah karena terdapat lembah awal yang berada pada bagian sebelah kiri puncak, puncak, dan akhir lembah yang berada pada bagian sebelah kanan puncak. Proses tersebut terus berulang hingga similarity score yang telah di smoothing terakhir dari seluruh token-sequence.
2. Menghitung nilai selisih dari masing-masing similarity score yang telah di smoothing yang berada pada sebelah kiri puncak yang dimulai dari awal lembah hingga pucak. Juga menghitung selisih dari masing-masing similarity score yang berada pada sebelah kanan puncak yang dimulai dari pucak hingga akhir lembah.
3. Menjumlahkan seluruh nilai selisih yang ada pada sebelah kiri dan kanan puncak. Nilai inilah yang disebut sebagai depth score.

Pada proses menghitung nilai cutoff dilakukan proses perhitungan nilai cutoff yang diperoleh dari pengurangan nilai rata-rata depth score terhadap setengah dari nilai standar deviasi depth score atau dapat ditulis menjadi $\bar{s} - \sigma/2$. Depth score yang digunakan dalam perhitungan nilai cutoff ini terbatas pada depth score yang menunjuk pada lokasi terjadinya peralihan paragraf. Hasil dari proses menghitung nilai cutoff adalah nilai cutoff. Nilai cutoff ini selanjutnya digunakan sebagai nilai tengah antara depth score yang menunjuk pada lokasi terjadinya peralihan paragraf yang pantas untuk dipilih sebagai tempat terjadinya perubahan tema, dan depth score yang lainnya.

Pada proses mencari nilai depth score yang lebih besar atau sama dengan nilai cutoff dilakukan proses pencarian depth score yang lebih besar atau sama dengan nilai cutoff. Proses ini menghasilkan depth score yang lebih besar atau sama dengan nilai cutoff.

Setelah menemukan depth score yang lebih besar atau sama dengan nilai cutoff, maka berikutnya dilakukan proses pencarian lokasi peralihan paragraf yang paling dekat dengan masing-masing nilai depth score tersebut. Pada proses ini dibutuhkan inputan depth score dan keterangan token. Paragraf-paragraf yang telah ditemukan akan ditandai, hal ini dilakukan untuk menghindari pemilihan sebuah lokasi peralihan paragraf yang sama oleh sejumlah depth score yang berbeda. Hal ini menyebabkan pemilihan sebuah lokasi peralihan paragraf hanya terbatas pada sebuah depth score saja.

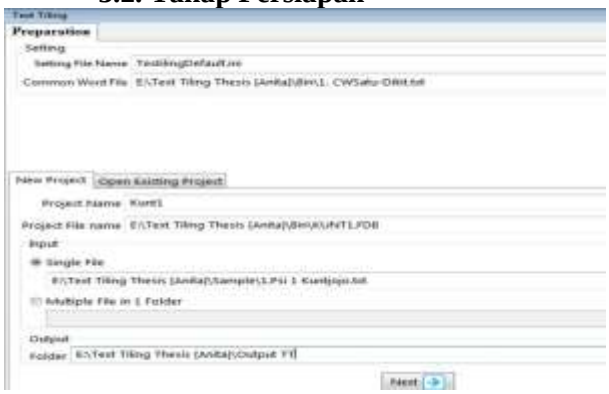
HASIL DAN PEMBAHASAN

Pada subbab ini menjelaskan jalannya aplikasi segmentasi dokumen mulai dari awal, yaitu tahap persiapan, hingga akhir, yaitu tahap menghasilkan output segmentasi dokumen.

3.1. Tahap Pengumpulan Data

Data yang dikumpulkan berupa *ebook* ilmu sosial, novel, jurnal, berita, artikel yang di-*download* pada www.google.com.

3.2. Tahap Persiapan

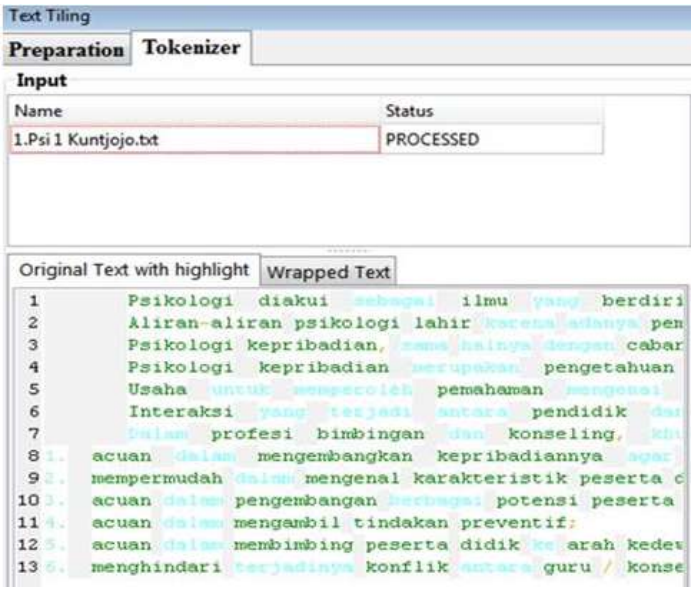


Gambar 2 : Tahap Persiapan pada Program

Tahap pertama dalam menjalankan aplikasi segmentasi dokumen menggunakan metode text tiling adalah tahap persiapan yang bertujuan memasukkan data yang akan digunakan dalam proses segmentasi dokumen. Seluruh data yang telah di dapatkan dari situs www.google.com dihilangkan format bacaan, gambar, dan tabel. Lalu data tersebut diletakkan di aplikasi notepad agar dapat di baca oleh sistem. Lalu data yang disimpan dalam bentuk notepad dimasukkan ke dalam program untuk diproses segmentasi dokumen. Program ini dinamakan Aplikasi Segmentasi Dokumen Bahasa Indonesia Menggunakan Metode Text Tiling. Pada program ini, dapat memasukkan dua macam data yaitu data single (data yang dimasukkan ke aplikasi berjumlah satu) atau data multiple data (data yang dimasukkan ke aplikasi berjumlah banyak, bias satu folder). Penjelasan tersebut dapat dilihat pada gambar 2.

3.3. Tahap Tokenizer

Tahap kedua dalam aplikasi segmentasi dokumen adalah tahap tokenizer yang bertujuan untuk menampilkan setiap token pada bacaan. Pada tahap tokenizer ini terdapat proses mengambil setiap token yang berupa karakter latin dari file bacaan (tokenisasi), proses mengubah token menjadi huruf kecil (*case folding*), dan proses filterisasi setiap token pada file bacaan dengan file *common word* (*common word removal*). Berikut tampilan aplikasi tahap tokenizer yang ditampilkan pada gambar 3.



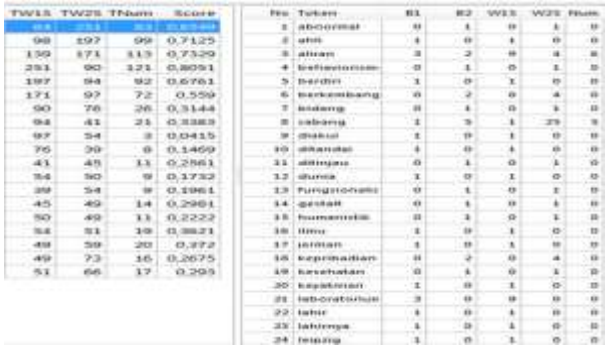
Gambar 3.Tahap Tokenizer

3.4. Tahap Similarity

Tahap ketiga dalam aplikasi segmentasi dokumen ini adalah tahap Similarity yang bertujuan untuk menghitung nilai similarity score pada setiap celah token-sequence. Hal pertama yang harus dilakukan pada aplikasi tahap ketiga ini yaitu menentukan jumlah token dalam token-sequence pada masukan Total Token In 1 Sequence dan jumlah token-sequence dalam blok pada masukan Total Sequence In 1 Block lalu menekan tombol Process untuk melakukan proses similarity. Selain tombol proses terdapat tombol Next yang digunakan untuk menuju tahap berikutnya, tombol Close & Reset yang digunakan untuk menutup project yang telah terbuka dan menuju tahap preparation, dan tombol Exit yang digunakan untuk keluar dari aplikasi segmentasi dokumen. Tombol Exit terletak di tepi kanan atas pada aplikasi ini. Selanjutnya, pada seluruh tampilan aplikasi hanya terdapat tombol-tombol tersebut yang mempunyai fungsi yang sama. Tampilan aplikasi pada tahap similarity ditampilkan pada gambar 4. Pada tahap similarity ini terdapat dua hasil yaitu hasil dari pembentukan pseudosentences (kalimat semu) dan perhitungan nilai similarity score. Pada gambar 4, jumlah token dalam token-sequence yang diinputkan sebanyak 20 dan dan jumlah token-sequence dalam blok sebanyak 3.

Gambar 4. Tahap Similairty

Pada gambar 5 ditampilkan hasil kedua dari proses similarity yaitu perhitungan similarity score dari file bacaan yang ditampilkan pada aplikasi segmentasi dokumen. Untuk proses perhitungan similarity score dilakukan dengan cara membandingkan dua buah blok yang saling bersebelahan dengan menggunakan perhitungan cosinus. Pada gambar 5 terdapat dua tabel yaitu tabel yang sebelah kiri dan tabel yang sebelah kanan. Tabel yang sebelah kanan merupakan detail perhitungan similarity score, yang mana menghitung bobot kemunculan token pada blok 1 yang dibandingkan dengan kemunculan token pada blok 2. Tabel sebelah kiri merupakan perhitungan lanjut dari tabel sebelah kanan sehingga tabel sebelah kiri merupakan hasil akhir perhitungan similarity score. Untuk perhitungan similarity score pada tiap blok berbeda karena kemunculan token pada blok yang satu dengan blok yang lain berbeda pula.



TW1S	TW2S	TNum	Score
98	197	99	0,7125
139	171	113	0,7329
231	90	121	0,8091
197	94	92	0,6761
171	97	72	0,559
90	76	26	0,3144
94	41	21	0,3383
97	54	3	0,0415
76	39	6	0,1469
41	45	11	0,2561
54	90	9	0,1732
39	54	9	0,1961
45	49	14	0,2901
90	49	11	0,2222
54	51	10	0,3821
49	59	20	0,372
49	73	16	0,2675
51	66	17	0,293

No	Token	B1	B2	W1S	W2S	Num
1	abnormal	0	1	0	1	0
2	ahli	1	0	1	0	0
3	aliran	3	2	9	4	6
4	beberapanya	0	1	0	1	0
5	berdiri	1	0	1	0	0
6	berkembang	0	2	0	4	0
7	bidang	0	1	0	1	0
8	cabang	1	3	1	29	5
9	dakui	1	0	1	0	0
10	dikenalkan	1	0	1	0	0
11	dimana	0	1	0	1	0
12	dunia	1	0	1	0	0
13	fungsionalnya	0	1	0	1	0
14	gestalt	0	1	0	1	0
15	humanistik	0	1	0	1	0
16	ilmu	1	0	1	0	0
17	jiwanya	1	0	1	0	0
18	kepribadian	0	2	0	4	0
19	kesehatan	0	1	0	1	0
20	kejahiwan	1	0	1	0	0
21	lebih	3	0	9	0	0
22	lahir	1	0	1	0	0
23	lahirnya	1	0	1	0	0
24	lelaku	1	0	1	0	0

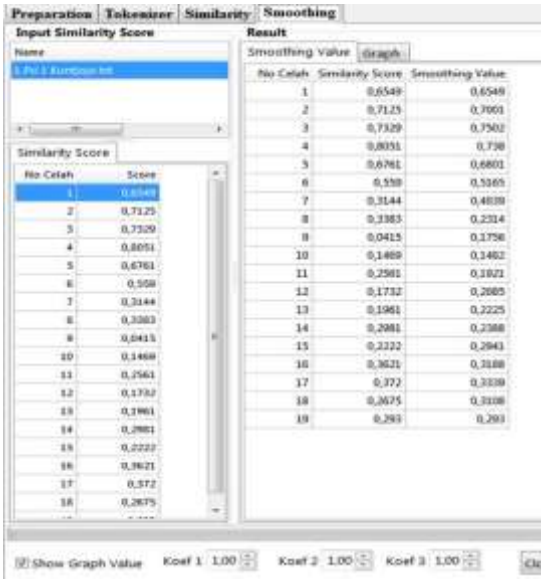
Gambar 5. Detail dan Hasil Perhitungan Similarity Score pada Aplikasi Segmentasi Dokumen

Pada tabel yang sebelah kanan yang menampilkan detail perhitungan similarity score yaitu proses pencarian kemiripan leksikal antara blok pertama dan blok kedua pada seluruh blok/celah token-sequence. Terdapat kolom No yang berisi no token, kolom Token yang berisi token yang lolos dari proses filterisasi dan setiap token dicatat sekali, kolom B1 yang berisi jumlah token yang muncul pada blok 1, kolom B2 berisi jumlah token yang muncul pada blok 2, kolom W1S yang berisi perhitungan nilai b1 kuadrat, kolom W2S yang berisi perhitungan nilai b2 kuadrat, dan kolom Num yang berisi perhitungan perkalian antara b1 dan b2. Misalkan, pada no 3 yaitu token "aliran", jumlah token "aliran" yang muncul pada blok 1 sebanyak 3 sehingga nilai b1 adalah 3. Begitu pula pada blok 2, token "aliran" muncul sebanyak 2 sehingga nilai b2 adalah 2. Untuk nilai W1S yang merupakan kuadrat dari b1 maka nilai W1S yaitu 3 kuadrat adalah 9 dan untuk nilai W2S yang merupakan kuadrat dari b2 maka nilai W2S yaitu 2 kuadrat adalah 4. Nilai Num didapatkan dari perkalian antara b1 dan b2 sehingga nilai num adalah 3 dikali 2 yaitu 6.

Pada tabel sebelah kiri yang menampilkan hasil akhir dari perhitungan similarity score terdapat kolom TW1S yang berisi total nilai dari W1S secara keseluruhan pada suatu blok, kolom TW2S yang berisi total nilai dari W2S secara keseluruhan pada suatu blok, kolom Tnum yang berisi total nilai dari num secara keseluruhan pada suatu blok, kolom Score yang berisi similarity score. Misalkan, baris pertama yang merupakan blok/no 1 pada tabel sebelah kiri yang merupakan perbandingan antara blok 1 yang terdiri dari token-sequence 1,2,3 dan blok 2 yang terdiri dari token-sequence 4,5,6. Pada blok/no 1 terdapat nilai TW1S sebesar 64 yang didapatkan dari total keseluruhan dari perhitungan W1S pada blok/no 1, nilai TW2S sebesar 251 yang didapatkan dari total keseluruhan dari perhitungan W2S pada blok/no 1, nilai Tnum sebesar 83 yang didapatkan dari total keseluruhan dari perhitungan num pada blok/no 1, dan nilai score sebesar 0,6549 yang didapatkan dari Tnum dibagi dengan akar kuadrat dari perkalian TW1S dan TW2S pada blok/no 1.

3.4. Tahap Smoothing

Tahap keempat dalam proses segmentasi dokumen ini adalah tahap smoothing yang bertujuan untuk menghitung smoothing dari similarity score pada setiap celah token-sequence. Pertama kali yang dilakukan pada tahap keempat ini yaitu menentukan jumlah koefisien similarity score pada masing-masing masukan yaitu pada masukan koef 1, koef 2, dan koef 3. Tampilan aplikasi tahap smoothing yang ditampilkan pada gambar 6.

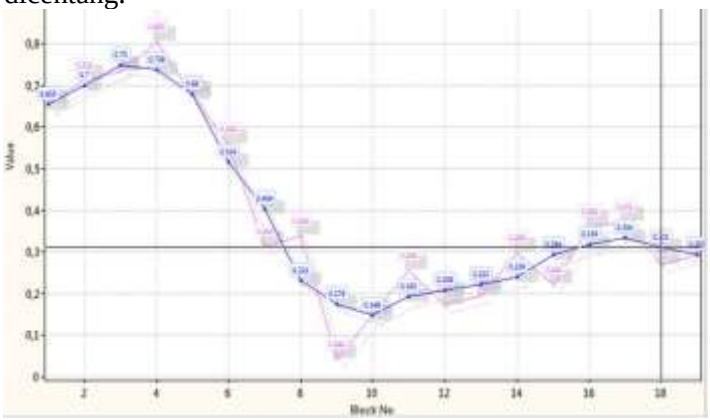


Gambar 6. Tahap Smoothing

Pada gambar 6 hasil proses smoothing ditampilkan pada tabel yang sebelah kanan pada tab Smoothing Value. Pada tab Smoothing Value terdapat kolom no celah yang berisi no urut blok, kolom similarity score yang berisi nilai similarity score yang belum dilakukan proses smoothing, dan kolom smoothing value yang berisi similarity score yang telah dilakukan proses smoothing. Proses smoothing terhadap similarity score pada tahap ini disebut *average smoothing*. Langkah-langkah proses *average smoothing* yang harus dilakukan untuk setiap blok/celah token-sequence i adalah mencari similarity score yang berada di sebelah kiri celah token-sequence i, mencari similarity score yang berada di sebelah kanan celah token-sequence i, mencari similarity score di celah token-sequence i, menghitung rata-rata dari ketiga nilai tersebut pada celah token-sequence/blok i. Untuk perhitungan rata-rata dilakukan dengan rumus (((koef1 * sim1)+ (((koef2 * sim2) + (((koef3 * sim3)) / (koef1 + koef2 + koef3)) (baris 3).

Misalnya pada hasil perhitungan smoothing similarity score yang ditampilkan pada gambar 6, untuk menghitung smoothing similarity score pada blok 2, maka harus dicari similarity score yang berada di sebelah kiri blok 2 yaitu similarity score pada blok 1 sebesar 0,6549. Lalu dicari similarity score yang berada di sebelah kanan blok 2 yaitu similarity score pada blok 3 sebesar 0,7329. Kemudian dicari similarity score pada blok 2 sebesar 0,7125. Dari ketiga angka tersebut maka dihitung nilai rata-ratanya menggunakan rumus di atas. Dengan menggunakan nilai koefisien similarity score di atas, maka untuk menghitung smoothing similarity score pada blok 2 adalah (((1 * 0,6549)+ (((1 * 0,7329) + (((1 * 0,7125)) / (1 + 1 + 1)), dari perhitungan tersebut didapatkan hasil sebesar 0,7001.

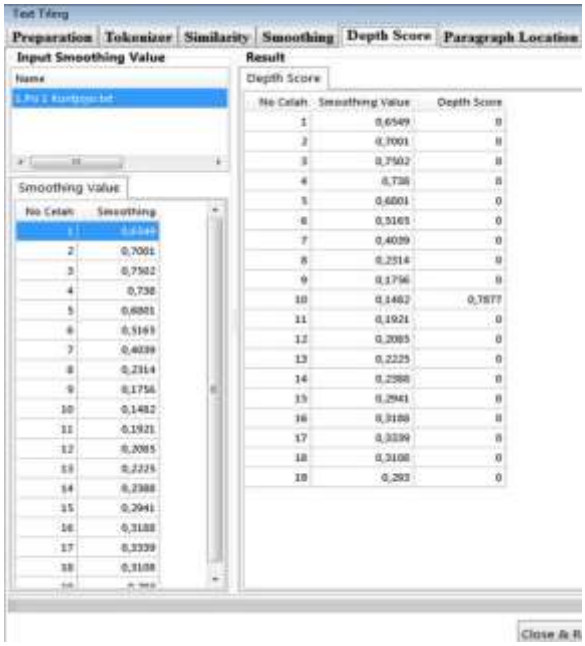
Pada aplikasi segmentasi dokumen, hasil dari perhitungan smoothing similarity score juga ditampilkan grafik dari smoothing similarity score yang ditampilkan pada group Result tab Graph. Untuk tampilan grafik pada group Result tab Graph ditampilkan pada gambar 7. Pada grafik, terdapat dua warna garis yaitu garis biru dan garis pink. Garis biru pada grafik menunjukkan garis smoothing similarity score yang sedangkan garis pink pada grafik menunjukkan garis similarity score yang belum dilakukan proses smoothing. Sumbu x menunjukkan no celah token-sequence dan sumbu y menunjukkan nilai similarity score. Jika ingin melihat titik dengan jelas maka tekan dua kali pada salah satu nilai pada grafik maka grafik akan membesar dan menampilkan detail setiap titik pada grafik tersebut. Untuk menampilkan ke bentuk semula (mengecilkan grafik) lakukan klik dua kali pada grafik maka grafik akan tampil seperti semula. Pada grafik terlihat nilai pada grafik, hal ini menunjukan bahwa pilihan Show Graph Value telah dicentang.



Gambar 7. Tampilan Grafik Smoothing Similarity Score

3.5. Tahap Depth Score

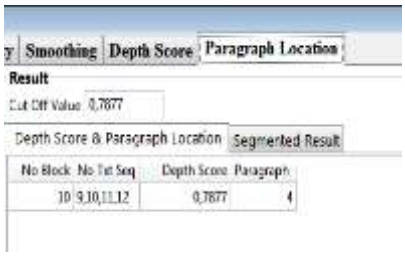
Tahap kelima dalam proses segmentasi dokumen ini adalah tahap depth score yaitu proses untuk mencari nilai depth score.Tampilan aplikasi segmentasi dokumen untuk tahap depth score akan ditampilkan pada gambar 8 yang terletak pada tab sebelah kanan. Hasil dari pencarian depth score ditemukan nilai depth score pada No Celah 10 dengan nilai 0,7877.



Gambar 8. Tampilan Aplikasi Segmentasi Dokumen Tahap Depth Score

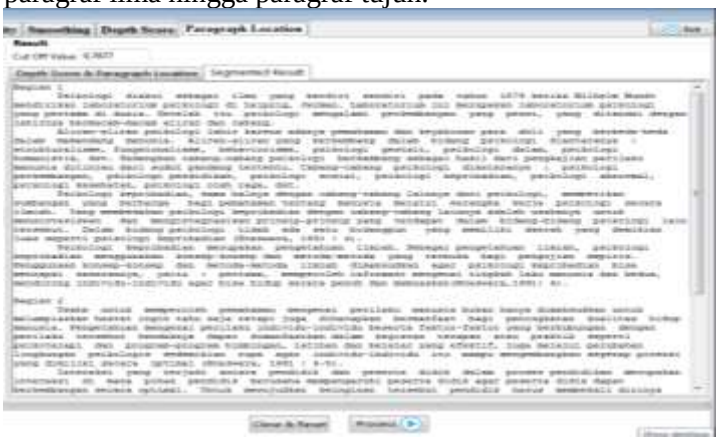
3.6. Tahap Lokasi Paragraf

Tahap keenam, sekaligus tahap terakhir, dalam proses segmentasi dokumen ini adalah tahap *paragraph location* yaitu proses untuk mencari nilai cutoff, proses mencari nilai depth score yang melebihi nilai cutoff, proses mengidentifikasi lokasi paragraf sebagai tempat perubahan tema, dan menampilkan hasil segmentasi. Tampilan aplikasi tahap paragraph location tab Depth Score & Location ditampilkan pada gambar 9. terlihat bahwa lokasi paragraf letak perubahan tema berada pada no blok 10 yang terletak pada paragraf 4. Sehingga paragraf yang terletak setelah paragraf 4 mempunyai tema yang berbeda dengan paragraf-paragraf sebelumnya, dengan kata lain paragraf 1 hingga paragraf 4 tergolong menjadi satu tema dan paragraf 5 hingga paragraf akhir tergolong menjadi tema yang lain.



Gambar 9. Tahap Paragraph Location Tab Depth Score & Paragraph Location

Pada gambar 10 menampilkan tahap *paragraph location* tab tab Segmented Result pada aplikasi segmentasi dokumen dari file bacaan yang menampilkan hasil akhir dari aplikasi segmentasi dokumen yang menampilkan bacaan asli disertai letak terjadinya perubahan tema yang ditandai dengan "bagian " dan diikuti dengan no hasil segmentasi. Untuk menentukan letak terjadinya perubahan perubahan tema dipilih nilai depth score yang melebihi nilai cutoff. Pada gambar 10 terlihat hasil segmentasi dari proses segemntasi pada file bacaan. Pada file bacaan tersebut mempunyai tujuh paragraf setelah dilakukan proses segmentasi menghasilkan hasil segmentasi yaitu terdapat dua bagian, bagian pertama meliputi paragraf satu hingga paragraf empat dan bagian kedua meliputi paragraf lima hingga paragraf tujuh.



Gambar 10. Tahap Paragraph Location Tab Segmented Result

3.7. Upaya Peningkatan Performansi Algoritma Text Tiling

Upaya peningkatan performasi algoritma text tiling dilakukan agar hasil segmentasi menjadi lebih baik. Beberapa upaya yang dilakukan untuk meningkatkan performansi algoritma text tiling antara lain :

- Merubah perhitungan nilai cutoff.
  Perhitungan nilai cutoff dengan cara yang baru ini didapatkan dengan cara mengurutkan depth score secara *ascending*, lalu memberikan nilai awal pada selisih dengan nilai 0, kemudian menghitung selisih antar depth score yaitu dengan cara mengurangkan depth score sekarang dengan depth sebelumnya, dan mencari nilai maksimal dari selisih tersebut. Nilai maksimal dari nilai selisih tersebut yang dijadikan sebagai nilai cutoff.
- Merubah nilai koefisien smoothing similarity score.

Untuk perubahan koefisien similarity score, dilakukan dengan input dari user. Untuk nilai koefisien tidak ketentuan pasti jadi nilainya tergantung oleh user hingga ditemukan hasil segmentasi yang sesuai dengan bacaan asli. Sedangkan ukuran window size yang digunakan yaitu 3. Rumus yang digunakan pada perhitungan *average smoothing* dengan merubah nilai koefisien yaitu :

$$\frac{(koefisien1 * similarity\ score1) + (koefisien2 * similarity\ score2) + (koefisien3 * (2}{koefisien1 + koefisien2 + koefisien1}$$

Rumus yang tertera pada rumus (2) digunakan untuk perhitungan smoothing dengan merubah nilai koefisien. Dimana similarity score yaitu nilai similarity yang didapatkan dari proses sebelumnya yaitu perhitungan nilai similarity score dan koefisien yaitu nilai koefisien yang diinputkan oleh user.

- Menambah jumlah token pada common word.
  Tujuan dari penambahan file common word ini yaitu agar dapat mendapatkan hasil segmentasi yang lebih baik. Jumlah common word yang digunakan pada percobaan pertama pada penelitian adalah 1159 kata sedangkan jumlah common word yang digunakan pada uji coba kedua saat dilakukan penambahan jumlah common word adalah 4140 kata.
- Mengurangi Token Berdasarkan Persebaran Token-Sequence.
  Tujuan dari pengurangan token berdasarkan persebaran token-sequence adalah agar mendapatkan hasil segmentasi yang lebih baik. Pengurangan token dilakukan pada proses filterisasi dirasa kurang cukup. Hal ini dikarenakan token yang dicatat pada common word diberlakukan untuk semua bacaan, padahal suatu token pada satu bacaan bisa jadi token tersebut menjadi common word tetapi di bacaan lain token tersebut tidak menjadi common word. Karena alasan tersebut maka perlu dilakukan pengurangan token untuk kedua kalinya agar menghasilkan hasil segmentasi yang lebih baik. Jumlah token pada suatu bacaan sangat mempengaruhi hasil perhitungan dalam tahapan text tiling, yang nantinya juga akan mempengaruhi hasil segmentasi. Pengurangan token pada metode text tiling ini dilakukan sebanyak dua kali, yang pertama dilakukan saat filterisasi token pada tahap tokenisasi, apakah suatu token termasuk common word atau bukan. Jika token tersebut termasuk file common word maka token tersebut harus dihapus dan tidak digunakan pada proses yang ada pada tahapan text tiling. Pengurangan token yang kedua ini dilakukan saat tahap similarity. Jadi, pengurangan token pada persebaran token-sequence ini dilakukan pada token yang telah lolos proses filterisasi. Proses token pada persebaran token-sequence dilakukan setelah mealui proses pembentukan token-sequence dan sebelum proses pembentukan blok. Token yang dihapus pada sesi yang kedua ini harus memiliki kriteria sebagai berikut :

1. Token tersebut hanya muncul sekali pada suatu token-sequence dan berjumlah satu

Token	Token-Sequence				
	1	2	3	4	5
Surabaya	-	-	1	-	-

2. Token tersebut hanya muncul sekali pada suatu token-sequence dan berjumlah lebih dari satu

Token	Token-Sequence				
	1	2	3	4	5
Surabaya	-	7	-	-	-

3. Token tersebut muncul pada semua token-sequence

Token	Token-Sequence				
	1	2	3	4	5
Surabaya	1	2	1	1	1

4. Token tersebut muncul pada hampir seluruh token-sequence kecuali satu token-sequence.

Token	Token-Sequence				
	1	2	3	4	5
Surabaya	1	-	1	1	2

KESIMPULAN

Penelitian ini mengembangkan salah satu algoritma segmentasi yaitu algoritma text tiling. Algoritma text tiling bertujuan untuk membagi dokumen panjang menjadi unit multi paragraf yang berhubungan. Data yang digunakan dalam uji coba yaitu dokumen lalu dokumen tersebut dihilangkan format bacaan untuk dijadikan input dalam aplikasi segmentasi ini. Metode text tiling ini

mempunyai tiga tahapan yaitu tokenisasi, penentuan kemiripan, dan pengenalan batas. Hasil dari segmentasi menggunakan algoritma text tiling sangat dipengaruhi oleh file common word, penentuan jumlah token dalam token-sequence, penentuan jumlah token-sequence dalam satu blok, benar tidaknya penulisan suatu kata, dan algoritma text tiling ini sangat sensitive dengan format bacaan, seperti judul dan sub judul, sehingga format bacaan tersebut harus dihapus hingga meninggalkan badan teks saja.

Dikarenakan dalam uji coba metode text tiling ini belum menghasilkan hasil yang optimal maka dilakukan upaya peningkatan performansi dilakukan untuk meningkatkan hasil segmentasi menjadi lebih baik. Beberapa upaya yang dilakukan untuk peningkatan performansi adalah merubah perhitungan nilai cutoff, merubah koefisien similarity score, menambah token pada file common word, dan menghapus token pada persebaran token-sequence. Hasil segmentasi setelah dilakukan upaya segmentasi menjadi meningkat. Dari berbagai hasil uji coba segmentasi bacaan yang dilakukan pada data sebanyak 15 bacaan mendapatkan hasil segmentasi dengan nilai precision 59,3% dan recall 80%. Hasil segmentasi dari uji coba tersebut menggunakan jumlah common word 4140, total koefisien similarity score sebesar 5, jumlah token dalam token-sequence sebesar 20, dan jumlah token-sequence dalam blok sebesar 3.

Terdapat beberapa hal yang dapat mempengaruhi hasil dari algoritma text tiling yang dapat dihandle oleh sistem yaitu benar tidaknya penulisan suatu kata, dan algoritma text tiling ini sangat sensitive dengan format bacaan. Sehingga saran untuk penelitian selanjutnya yaitu dengan menambahkan fitur perbaikan penulisan kata dan menghilangkan format bacaan secara otomatis.

REFERENSI

Claudia Regina Rahardjo. (2003). Studi Analisa Pengenalan Struktur Sub Topik dalam Teks dengan Menggunakan Algoritma Text Tiling. Perpustakaan Sekolah Tinggi Teknik (STTS) Surabaya, Indonesia.

Jati Sasongko Wibowo dan Sri Hartati. (Jan, 2011). *Text Document Retrievel In English Using Keywords of Indonesian Dictionary Based*. IJCCS, Vol. 5 No. 1.

Kosasih, E. (2007). *1700 Bank Soal Bimbingan Pemantapan Bahasa Indonesia Untuk SMA/MA*. Bandung : Yrama Widya.

Lamhot Robinson. *Implementasi Metode Generalized Vector Space Model Pada Aplikasi Information Retrieval untuk Pencarian Informasi Pada Kumpulan Dokumen Teknik Elektro Di UPT BPI LIPI*. Universitas Komputer Indonesia. Bandung. ISSN : 2089-903.

M.K., Sabarti Akhadiah., Maidar Arsjad., dan Sakura Ridwan. (1986). *Materi Pokok Bahasa Indonesia*. Jakarta : Karunika Jakarta.

Marti A. Hearst. (29 April 1994). *Context and Structurein Automated Full-Text Information Access*. Computer Science Division (EECS) University of California Berkeley, California 94720.

Marti A. Hearst. (June 1994). *Multi-Paragraph Segmentation of Expository Text*. Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, NM.

Marti A. Hearst. (1997). *TextTiling : Segmenting Text into Multi-Paragraph Subtopic Passages*. Comput. Linguist., vol. 23, no. 1, pp. 33–64. Retrieved from http://dl.acm.org/citation.cfm?id=972687%5Cnhttp://dl.acm.org/citation.cfm?id=972684.972687.

Marti A. Hearst and C. Plaunt. (1993). *Subtopic Structuring for Full-Length Document Access*. Proc. Annu. Int. ACM SIGIR Conf. Res. Dev. Infofmation Retr., no. June 2002, pp. 59–68. Retrieved from doi: 10.1145/160688.160695.

Rahardi, R. Kunjana. (2009). *Penyuntingan Bahasa Indonesia Untuk Karang-Mengarang*. Jakarta : Erlangga.

Rahardi, R. Kunjana. (2006). *Dimensi-Dimensi Kebahasan Aneka Masalah Bahasa Indonesia Terkini*. Jakarta : Erlangga.

Satanjeev Banerjee and Alexander I. Rudnicky. (2006). *A TextTiling Based Approach to Topic Boundary Detection in Meetings*. Language Technologies InstituteCarnegie Mellon UniversityPittsburgh, PA. United States.