JURNAL PINTER
VOL 8. NO. 1 JUNI 2024
e – ISSN: 2597 – 4475
http://doi.org/10.21009/pinter.8.1.10

PERINGKASAN TEKS BERBAHASA INDONESIA MENGGUNAKAN TEKNIK EKSTRAKSI DENGAN ALGORITMA *LATENT SEMANTIC ANALYSIS (LSA)* DENGAN VARIASI TF-IDF UNTUK PERINGKASAN *SINGLE DOCUMENT*

Deki Riana[1], Widodo[2], Murien Nugraheni[3]

[1] Mahasiswa Prodi Pendidikan Teknik Informatika dan Komputer, FT – UNJ
[2,3] Dosen Prodi Pendidikan Teknik Informatika dan Komputer, FT – UNJ
[1]DekiRiana_5235163567@mhs.unj.ac.id, [2]widodo@unj.ac.id, [3]muriennugraheni@unj.ac.id

Abstrak

Perkembangan teknologi menyebabkan masyarakat mudah untuk mencari dan mendapatkan informasi. Banyaknya informasi yang disajikan menyebabkan masyarakat memerlukan penggalian informasi yang mencakup keseluruhan dokumen secara ringkas. Peringkasan dokumen dapat menyajikan inti dari dokumen secara singkat tanpa mengurangi esensi dokumen. Peringkasan *single* dokumen adalah peringkasan yang diekstrak dari satu dokumen. Terdapat banyak algoritma yang dapat digunakan untuk membuat sistem peringkas *single* dokumen otomatis. Salah satu algoritma tersebut adalah *Latent semantic analysis* (LSA). Serta algoritma yang umum digunakan dalam *Text Processing* adalah TF-IDF. Penelitian ini bertujuan untuk mengetahui kegunaan algoritma LSA dan TF-IDF untuk peringkasan *single* dokumen ilmiah berbahasa Indonesia. Bahan yang digunakan untuk penelitian ini adalah jurnal ilmiah berbahasa Indonesia sebanyak lima puluh (50) dokumen dengan topik yang sama yaitu "*Natural language processing".* Penelitian ini menggunakan *library* NLTK dan Sastrawi untuk *library corpus, stopword,* dan *stemming* berbahasa Indonesia, kemudian hasil peringkasan menggunakan algoritma *Latent semantic analysis* (LSA) dengan variasi TF-IDF dievaluasi menggunakan *Recall-Oriented Understudy for Gisting Evaluation* (*ROUGE*-n). Hasil penelitian yang didapatkan menggunakan *ROUGE-1* dari peringkasan otomatis oleh sistem menggunakan algoritma *LSA* dengan variasi *TF-IDF* adalah nilai *presisi* 0,713, *recall 0,718,* dan *f-measure 0,715* untuk kompresi sebanyak 5 kalimat. Selanjutnya hasil optimal pada kompresi sebanyak 10 kalimat algoritma *LSA* dengan variasi *TF-IDF* didapatkan nilai optimal untuk *presisi* 0,680, *recall 0,698,* dan *f-measure* 0,689. Untuk kompresi sebanyak 20 kalimat algoritma *LSA* dengan variasi *TF-IDF* didapatkan hasil nilai optimal untuk *presisi 0,752, recall 0,739, dan f-measure 0,745.* Dapat disimpulkan *ROUGE-1* didapatkan hasil pengujian optimal tertinggi dengan kompresi sebanyak 20 kalimat dengan nilai rata-rata *presisi 0,529, recall 0,562, dan f-measure 0,538*. Kemudian hasil optimal dihasilkan dengan kompresi sebanyak 10 kalimat dengan nilai rata-rata nilai *presisi* 0,493 *recall 0*, 518 dan *f-measure* 0, 499.

Kata kunci: **Peringkasan *Single* Dokumen, *Latent semantic analysis* (LSA), TF-IDF, *ROUGE*.**

1. Pendahuluan

Penelitian mengenai peringkasan dokumen telah berlangsung sejak 1958 oleh Luhn (Luhn, 1958, 159-165) dan terus berkembang sampai sekarang. Secara umum pendekatan yang digunakan untuk peringkasan dokumen ada dua yaitu metode abstraksi dan ekstraksi. Pendekatan abstraksi melakukan peringkasan teks dengan cara menginterpretasikan teks asal dan menuliskan kembali ke dalam versi yang lebih singkat tetapi mempunyai semantik yang sama. Pendekatan ekstraksi berusaha membangkitkan ringkasan dokumen berdasarkan kata-kata atau kalimat- kalimat yang ada dalam teks asal berdasarkan tingkat kepentingannya (Lin, 1999, pages 81–94).

Secara umum, peringkasan metode otomasi generik terbagi dalam dua kategori yaitu supervisi dan unsupervisi. Metode supervisi menggunakan algoritma dengan sejumlah besar data latih dan hasilnya adalah model peringkasan. Kedua metode tersebut membuat ringkasan dokumen dengan cara membangkitkan fitur

Available at:
https://journal.unj.ac.id/unj/index.php/pinter/article/view/52881

semantik dokumen. Metode pertama yang dikembangkan yaitu mencoba membangkitkan ringkasan dokumen berdasarkan keterkaitan semantic antar kata, sedang metode kedua melakukan peringkasan berdasarkan keterkatian semantik antar kalimat.

Pembobotan kata berdasarkan frekuensi sebuah *term* dalam dokumen yang bersangkutan disebut dengan *(Term Frequency)* TF. Semakin besar jumlah kemunculan suatu *term* dalam dokumen semakin besar pula bobot kata tersebut yang akan memberikan nilai kesesuaian yang semakin besar. *(Inverse document frequency)* IDF merupakan perhitungan dari *term* yang didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Kedua metode tersebut akan divariasikan dengan LSA.

Berdasarkan uraian diatas, penulis akan melakukan eksperimen peringkasan dokumen ilmiah berbahasa indonesia dengan algoritma LSA dan TF-IDF untuk menguji coba algoritma tersebut sebagai algoritma peringkasan dokumen jurnal ilmiah berbahasa indonesia.

Berdasarkan latar belakang maka permasalahan yang menjadi objek penelitian adalah perlunya pemahaman yang cepat tanpa harus membaca keseluruhan suatu dokumen, belum adanya penggunaan algoritma LSA dan TF-IDF untuk peringkasan dokumen jurnal ilmiah berbahasa indonesia, belum adanya pengujian algoritma LSA dan TF-IDF untuk peringkasan.

Penelitian ini bertujuan untuk mengetahui kegunaan algoritma LSA dan TF- IDF untuk peringkasan *single* dokumen. Kombinasi dari kedua algoritma dan metode tersebut dan efektivitasnya untuk peringkasan dokumen dengan hasil peringkasan yang sesuai tanpa mengurangi informasi penting yang terdapat dalam dokumen.

2. Dasar Teori

2.1. Peringkasan Dokumen

Peringkasan dokumen adalah proses mempersingkat suatu dokumen berupa teks seperti jurnal, karya tulis, berita, atau tulisan dengan menyisihkan teks yang berisi informasi tidak penting. Menurut (Najibullah dan Mingyan, 2015) peringkasan dokumen adalah proses penyajian kembali dokumen dalam bentuk yang lebih singkat tanpa membuang informasi penting yang terdapat pada dokumen tersebut. Peringkasan dokumen adalah salah satu bidang *Natural language processing* (NLP) yang dapat mengekstrak informasi penting dari teks asli untuk menghasilkan ringkasan (Gamaria Mandar dan Gunawan, 2018).

Peringkasan dokumen bertujuan untuk memudahkan memperoleh informasi dan ide pokok dari teks yang disajikan. Peringkasan dokumen diperoleh dengan menghilangkan kata, kalimat yang dianggap tidak relevan tanpa menghilangkan makna dari dokumen. Menurut (Zeniarja, Salam, Luthfiarta, Handoko, & Jamhari, 2013) Peringkasan dokumen ialah untuk memperoleh informasi yang penting dari sebuah dokumen (teks) yang disajikan kepada pembaca, karena peringkasan teks otomatis nampu menghilangkan kata, kalimat yang dianggap tidak relevan atau reundan dengan tetap menjaga inti makna dari dokumen. Peringkasan dapat dikategorikan menjadi beberapa golongan berdasarkan fungsi, jenis dan lainnya.

2.2. Peringkasan *Single Document*

Peringkasan *single document* adalah peringkasan yang diekstrak dari satu dokumen. (Ozsoy, Alpaslan, dan Cicckli, 2010). Peringkasan *single dcoument* hanya menyediakan peringkasan dari satu dokumen saja (Pranowo, Fabianus, Sigit 2014). Berbeda dengan *multi-document* input dapat berupa lebih dari satu sekaligus secara bersamaan dokumen dengan topik yang sama.

2.3. Peringkasan Dokumen dengan Metode Ekstraksi

Umumnya peringkasan dokumen diklasifikasi menjadi dua yaitu peringkasan ekstraktif dan peringkasan abstraktif (Gunawan, Juandi, dan Soewito, 2015). Peringkasan ekstraktif adalah ringkasan dengan memilih kalimat penting dari dokumen asli dengan cara mengekstrak kalimat berdasarkan fitur statistik dan linguistik, sedangkan peringkasan abstraktif adalah memahami dokumen asli dan menghasilkan kalimat baru dari dokumen yang diringkas, metode ini lebih kompleks serupa dengan ringkasan yang dilakukan oleh manusia (Badry, Eldin, dan Elzanfally, 2013).

Kategori pada peringkasan dokumen dapat mempengaruhi peringkasan yang dihasilkan. Salah satu kategori dalam peringkasan dokumen adalah peringkasan dengan ekstraksi informasi. Ekstraksi informasi adalah suatu teknik yang digunakan untuk menghasilkan informasi yang relevan dari dokumen berskala besar dengan hasil berupa informasi yang terstruktur *Resolution* (Fajri Umbara, 2016).

Berdasarkan teori, hasil ringkasan ekstraktif lebih baik dibandingkan dengan ringkasan abstraktif. Hal ini dikarenakan peringkasan abstraktif, seperti representasi semantik, inferens dan pembangun natural language relatif lebih sulit, dibandingkan pendekatan data driven, seperti ekstraksi kalimat (Erkan dan Radev, 2004).

2.4. *Term Frequency Inverse document frequency* (TF-IDF)

Algoritma *Term Frequency-Inverse document frequency* adalah cara pemberian bobot hubungan suatu kata (*term*) terhadap dokumen. Untuk dokumen tunggal tiap kalimat dianggap sebagai dokumen. Algoritma ini

menggabungkan dua konsep untuk perhitungan bobot, yaitu *term frequency (tf)* merupakan frekuensi kemunculan *term* t pada dokumen i. Dan *document frequency (df)* merupakan frekuensi banyaknya dokumen i dimana *term* t muncul (Ihsan Khairul, 2019).

Dalam *tf* frekuensi *term* pilihan yang paling sederhana adalah dengan menggunakan frekuensi baku dalam dokumen, yaitu berapa kali *term(t)* terjadi dalam dokumen *(d)*. Nilai *idf* sebuah *term* dapat dihitung menggunakan persamaan berikut:

$IDF = log\ (\frac{n}{df})$.....(1)

n = jumlah dokumen/kalimat yang terdapat *term* t

df = kemunculan frekuensi *term* terhadap *n*

Adapun algoritma yang digunakan dalam menghitung bobot *(w)* untuk masing- masing dokumen terhadap kata kunci atau *query* menggunakan persamaan berikut:

$W_{d.f} = tf_{d.f} \times IDF_t$.....(2)

$W_{d.f}$ = Bobot dokumen ke-*d* terhadap *term* ke-t

d = Dokumen ke-*d*

t = *term* ke-*t*

$tf_{d.f}$ = Frekuensi *term* dokumen ke-*d* terhadap *term* ke-t

IDF_t = Nilai IDF *term* ke-*t*

2.5. *Latent semantic analysis* (LSA)

LSA *(Latent semantic analysis)* adalah algoritma statistik aljabar yang mengekstrak struktur *semantic* yang tersembunyi dari kata dan kalimat (Muhammad Jamhari, Edi Noersasongko, Hendro Subagyo, 2014). Menurut (Ihsan Khoerul , 2019) *Latent semantic analysis (LSA)* adalah metode untuk mengekstrak sebuah tulisan dalam suatu dokumen dan kemudian mengaplikasikannya dalam perhitungan matematis. Penelitian dengan algoritma *LSA* cenderung berfokus pada kata – kata yang terdapat dalam tulisan tanpa memperhatikan urutan kata dan tata Bahasa dalam tulisan tersebut, sehingga suatu kalimat yang dinilai adalah kata-kata kunci pada kalimat tersebut. LSA adalah salah satu metode yang paling banyak digunakan untuk representasi makna kata (Altszyler, 2017).

3. Metodologi

3.1. Alat dan Bahan Penelitian

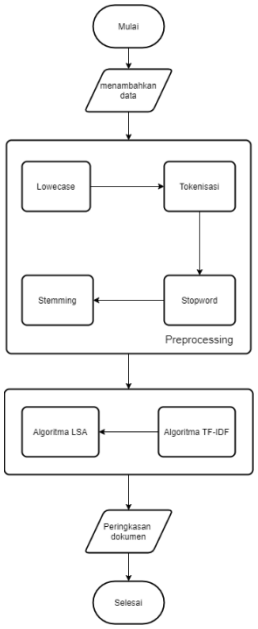
Alat yang digunakan untuk penelitian ini adalah komputer dan *library* yang akan digunakan untuk membuat *prototipe* dan menganalisis data, sedangkan data yang akan dianalisis didapatkan dari situs jurnal daring. Berikut penjelasan lengkapnya. Alat-alat yang digunakan dalam melaksanakan penelitian ini, yaitu berupa perangkat keras, antara lain :

1. Laptop dengan *processor* AMD *Ryzen 3* 2200U dan RAM 8GB.
2. Python 3.7, sebagai bahasa pemrograman
3. Pycharm, sebagai IDE Python
4. NLTK dan Sastrawi sebagai *library corpus, stopword,* dan *stemming* berbahasa Indonesia
5. Piptools sebagai LSA *library*
6. py*ROUGE* sebagai *library* untuk metode evaluasi *ROUGE*-N

Bahan yang digunakan untuk penelitian ini adalah jurnal ilmiah berbahasa Indonesia sebanyak lima puluh (50) paper dengan topik yang sama yaitu "*Natural language processing*". Hal tersebut bertujuan untuk memenuhi kriteria minimal analisis data statistik sebagaimana dikemukakan oleh Eri Yuningsih dalam Baley dalam Mahmud (2014:67) yang menyatakan bahwa untuk penelitian yang menggunakan analisis data statistik, ukuran sampel paling minimum adalah 30. Oleh karena itu, penelitian ini menggunakan sebanyak 50 dokumen ilmiah berbahasa indonesia.

3.2. Diagram Alir Penelitian

Secara garis besar, metode penelitian yang akan dilaksanakan seperti diagram alir dibawah ini:



Gambar 3.1 Diagram Alir Penelitian

3.3. Perancangan

Perancangan sistem analisis teks melibatkan empat tahap utama: penambahan data, pra-pemrosesan, pembobotan kata menggunakan algoritma TF-IDF, dan penerapan algoritma LSA dengan penjelasan berikut:

1. Menambahkan data: Pada tahap awal, data berupa penelitian ilmiah diambil dari berbagai sumber secara daring. Sebelum diolah teks dilakukan perubahan format menjadi *plain text.* Setelah itu, abstrak pada teks dihilangkan. Kemudian text akan disimpan dengan format (.txt) agar dapat diolah berikutnya. Untuk indikator – indikator data yang diambil adalah : Judul Penelitian, Bab 1, Bab 2, dan Bab 3.
2. Preprocessing: Sebelum data diolah data akan melalui beberapa tahapan guna meningkatkan akurasi data dalam *Natural language processing* (NLP).
    1) Lowercase: Tahapan awal pada proses ini adalah perubahan setiap huruf yang terdapat pada dokumen menjadi huruf kecil tanpa mengubah struktur kata pada dokumen. Perubahan ini dilakukan karena penggunaan huruf kapital dapat mempengaruhi arti pada setiap kata. Sebagai contoh kata "bangun" akan diartikan berbeda dengan "Bangun", "BANGUN' dan lainnya (*case sensitive)*.
    2) Tokenisasi: Pada tahap selanjutnya setelah semua huruf diubah menjadi huruf kecil kemudian kata-kata akan dibuat menjadi potongan kecil atau kata tunggal yang disebut token. Tahapan ini akan menghilangkan tanda baca, angka, dan karakter lainnya yang tidak terdapat dalam alfabet sehingga setiap kata akan menjadi masing – masing.
    3) *Stopwords:* Tahapan berikutnya yaitu *stopwords.* Tahapan ini bertujuan untuk menghilangkan kata yang tidak memiliki makna yang dapat mempengaruhi hasil sehingga akan merusak hasil akhir.
    4) *Stemming:* Tahapan dengan merubah kata menjadi kata dasar dengan menghilangkan imbuhan yang bertujuan mengurangi jumlah kata yang masuk ke dalam data penelitian dengan harapan dapat meningkatkan akurasi. Pada tabel 3.4 dapat dilihat contoh penerapan sebelum dan sesudah *stemming.*
3. Pembobolan Kata Menggunakan Algoritma TF-IDF*:* Pembobotan kata adalah proses untuk menghitung bobot suatu kata pada dokumen yang dilihat dari banyaknya frekuensi kemunculan kata pada sebuah teks atau dokumen kalimat. Algoritma yang digunakan pada penelitian ini adalah menggunakan *Term Frequency-Inverse Document* (TF – IDF).
4. Penerapan Algoritma LSA*:* Konsep LSA direalisasikan dengan menggunakan dua fitur utama yaitu matriks SVD, strukur bahasa dalam hal ini ialah, kalimat atau kata yang diubah menjadi sebuah matriks. secara matematis LSA dapat ditulis sebagai berikut :

$$A = USV^T \qquad (9)$$

A adalah matriks dokumen yang mewakili kalimat atau kata yang dikenal dengan matriks A_{mn}, *U* mendeskripsikan matriks *orthogonal* m x n yang dikenal dengan istilah *left singular vector*, *U* dihasilkan

dari perkalian antara $U = A.V.S^{-1}$. *Right singular vector (V)* merupakan matriks *orthogonal* n x m yang diperoleh dari *eigenvector* akar matriks A^TA. Adapun langkah-langkah LSA sebagai berikut (Geetha & Deepamala, 2015) :

1) Membentuk matriks A_{mn}.
2) Membuat matriks V dengan *eigenvalue*, dimana matriks V adalah hasil dari *eigenvector* matriks A^TA.
3) Membentuk matriks S dengan cara mengurutkan nilai tertinggi *eigenvalue* kemudian diakarkan.
4) Menghitung *length* pada setiap nilai matriks V^T dengan menggunakan rumus

   $$Sk = \sqrt{\sum_{i=1}^{n}(V^T)^2 . S^2} \quad (10)$$

5) Menentukan hasil ringkasan berdasarkan skor tertinggi dari dokumen kalimat.

S_k adalah panjang *vector k* pada kalimat yang dimodifikasi oleh laten *vector*. *n* adalah jumlah ruang dimensi baru. Hasil dari *length* terbesar pada setiap dokumen akan dijadikan ringkasan.

3.4. Pengujian

Pengujian dilakukan dengan menggunakan metode evaluasi Recall-Oriented Understudy for Gisting Evaluation (*ROUGE*). *ROUGE* menghitung jumlah n-gram kata yang overlap antara ringkasan sistem dengan ringkasan referensi Adapun teknik penghitungan *ROUGE*-N antara sebuah ringkasan sistem dan sekumpulan ringkasan manual terdapat pada persamaan :

$$ROUGE\text{-N} = \frac{\sum S \in \{ReferenceSummaries\}. \in S \sum Countmatch(gram)n}{\sum S \in \{ReferenceSummaries\}.gram \in S \sum Count(gram)n} \quad (11)$$

Dimana n adalah panjang dari n-gram, Countmatch(gramn) adalah jumlah n- gram yang sama antara sebuah ringkasan sistem dan sebuah ringkasan referensi, Count(gramn) adalah jumlah n-gram dalam ringkasan referensi.

4. Hasil dan Pembahasan

4.1. Deskripsi Hasil Penelitian

Penelitian ini dilakukan untuk mengetahui dan menganalisa kinerja peringkasan *single* dokumen ilmiah berbahasa Indonesia menggunakan metode *Latent Semantic Anaylisis (LSA)* dan *Term Frequency – Inverse document frequency (TF-IDF).* Setelah didapatkan hasil peringkasan kemudian dievaluasi dengan metode *Recall-Oriented Understady for Gisting Evaluation (ROUGE)* yang membandingkan hasil peringkasan sistem dengan peringkasan pakar; Perbandingan ringkasan 5 kalimat sistem dengan 5 kalimat pakar; Perbandingan ringkasan 10 kalimat sistem dengan 10 kalimat pakar; Perbandingan ringkasan 20 kalimat sistem dengan 20 kalimat pakar; Diperoleh nilai evaluasi *recall, precision, dan f-measure* dengan skala 0 – 1. Nilai yang digunakan sebagai parameter adalah nilai *f-measure* dimana merupakan nilai tengah antara nilai *precision* dan *recall*. Semakin tinggi nilai *f-measure* dengan mendekati 1 maka akan semakin baik.

4.2. Analisis Data Penelitian

Setelah dilakukan peringakasan menggunakan Algoritma *LSA* tahap berikutnya adalah menguji Hasil Ringkasan Menggunakan Algoritma LSA dan TF-IDF dengan *ROUGE*-1. Dari hasil pengujian didapatkan nilai *presisi, recall,* dan *F-Measure.* Untuk nilai lebih lengkap dari pengujian *ROUGE*-1 dapat dilihat pada Tabel 4.1.

Tabel 4.1. Hasil Pengujian *ROUGE*-1

No.	Evaluasi *ROUGE*-1								
	Kompresi 5 Kalimat			Kompresi 10 Kalimat			Kompresi 20 Kalimat		
	Presisi	*Recall*	*F-Measure*	*Presisi*	*Recall*	*F-Measure*	*Presisi*	*Recall*	*F-Measure*
1	0,484	0,431	0,456	0,496	0,466	0,481	0,589	0,591	0,590
2	0,466	0,460	0,463	0,539	0,459	0,496	0,543	0,399	0,460
3	0,326	0,561	0,412	0,324	0,649	0,432	0,373	0,652	0,475
4	0,444	0,459	0,451	0,440	0,484	0,461	0,537	0,603	0,568
5	0,476	0,421	0,447	0,584	0,636	0,609	0,607	0,609	0,608

6	0,604	0,477	0,533	0,595	0,405	0,482	0,640	0,484	0,551
7	0,513	0,356	0,420	0,594	0,500	0,543	0,574	0,556	0,564
8	0,398	0,526	0,453	0,414	0,495	0,451	0,322	0,652	0,431
9	0,368	0,499	0,424	0,414	0,521	0,461	0,439	0,554	0,490
10	0,349	0,555	0,429	0,422	0,530	0,470	0,529	0,498	0,513
11	0,438	0,445	0,442	0,421	0,487	0,452	0,442	0,477	0,459
12	0,477	0,417	0,445	0,434	0,467	0,450	0,460	0,537	0,495
13	0,468	0,437	0,452	0,460	0,517	0,487	0,521	0,524	0,522
14	0,517	0,498	0,507	0,528	0,507	0,517	0,576	0,542	0,559
15	0,434	0,671	0,527	0,485	0,488	0,487	0,585	0,485	0,512
16	0,312	0,626	0,416	0,400	0,650	0,495	0,548	0,703	0,616
17	0,502	0,434	0,466	0,486	0,459	0,472	0,458	0,558	0,503
18	0,231	0,627	0,337	0,333	0,566	0,419	0,416	0,662	0,511
19	0,370	0,541	0,439	0,400	0,532	0,457	0,463	0,555	0,505
20	0,305	0,525	0,386	0,408	0,531	0,462	0,515	0,597	0,553
21	0,398	0,541	0,459	0,562	0,593	0,577	0,576	0,569	0,572
22	0,503	0,391	0,440	0,555	0,481	0,515	0,556	0,472	0,510
23	0,575	0,592	0,583	0,680	0,698	0,689	0,573	0,594	0,583
24	0,569	0,594	0,577	0,608	0,592	0,600	0,591	0,591	0,591
25	0,444	0,431	0,437	0,507	0,503	0,505	0,640	0,612	0,626
26	0,632	0,601	0,616	0,611	0,449	0,518	0,622	0,577	0,599
27	0,598	0,428	0,499	0,479	0,458	0,468	0,542	0,458	0,496
28	0,642	0,437	0,520	0,471	0,575	0,518	0,567	0,625	0,594
29	0,600	0,298	0,398	0,451	0,483	0,466	0,559	0,577	0,568
30	0,540	0,401	0,460	0,586	0,501	0,542	0,653	0,657	0,656
31	0,469	0,421	0,444	0,453	0,455	0,454	0,420	0,564	0,482
32	0,502	0,366	0,423	0,595	0,497	0,542	0,562	0,489	0,523
33	0,554	0,586	0,569	0,545	0,565	0,555	0,588	0,574	0,581
34	0,432	0,461	0,446	0,418	0,527	0,466	0,487	0,553	0,518
35	0,480	0,439	0,458	0,374	0,523	0,436	0,419	0,510	0,460
36	0,629	0,375	0,470	0,623	0,500	0,555	0,586	0,618	0,601
37	0,503	0,393	0,441	0,463	0,432	0,447	0,454	0,446	0,450
38	0,517	0,467	0,491	0,588	0,453	0,511	0,583	0,478	0,525
39	0,713	0,718	0,715	0,598	0,620	0,609	0,516	0,603	0,556
40	0,583	0,599	0,591	0,647	0,554	0,597	0,752	0,739	0,745
41	0,553	0,429	0,483	0,548	0,420	0,476	0,605	0,506	0,551
42	0,442	0,497	0,468	0,489	0,581	0,531	0,488	0,595	0,536
43	0,394	0,500	0,441	0,447	0,441	0,444	0,546	0,558	0,552
44	0,501	0,666	0,572	0,450	0,587	0,510	0,423	0,571	0,486
45	0,347	0,540	0,422	0,397	0,512	0,447	0,447	0,470	0,458
46	0,402	0,636	0,493	0,395	0,608	0,479	0,361	0,698	0,476
47	0,584	0,350	0,438	0,590	0,371	0,455	0,655	0,549	0,597

48	0,527	0,497	0,512	0,442	0,543	0,487	0,512	0,380	0,436
49	0,424	0,416	0,420	0,508	0,505	0,506	0,596	0,645	0,619
50	0,375	0,518	0,435	0,395	0,538	0,455	0,426	0,571	0,488

4.3. Pembahasan

Dari hasil penelitian yang dilakukan dengan menguji peringkasan dokumen otomatis dengan menggunakan dokumen sebanyak lima puluh dokumen dan dilakukan pengujian menggunakan metode *ROUGE-N*, maka didapatkan hasil pengujian berupa nilai *presisi*, *recall*, dan *f-measure* yang dapat dilihat pada Tabel 4.23. Nilai yang dibandingkan pada hasil evaluasi peringkasan menggunakan metode *ROUGE-N* adalah nilai *f-measure.* Nilai tersebut merepresentasikan kualitas ringkasan sistem dan manual. Nilai *ROUGE-N* memiliki nilai antara 0 sampai 1. Jika hasil nilai = 0 berarti ringkasan hasil sistem dinyatakan tidak ada kemiripan sama sekali dengan hasil pakar. Sedangkan, jika hasil mendekati nilai = 1, maka ringkasan hasil sistem mirip dengan ringkasan hasil pakar. Berikut terlampir pada tabel 4.2

Tabel 4.2. Hasil Pengujian *ROUGE*-1

Nilai	**Kompresi**								
	5			**10**			**20**		
	Precision	***Recall***	***F-Measure***	***Precision***	***Recall***	***F-Measure***	***Precision***	***Recall***	***F-Measure***
Tertinggi	0,713	0,718	0,715	0,680	0,698	0,689	0,752	0,739	0,745
Terendah	0,231	0,298	0,337	0,324	0,371	0,419	0,322	0,380	0,431
Rerata	0,478	0,491	0,473	0,493	0,518	0,499	0,529	0,562	0,538

Berdasarkan pengujian dan analisis peringkas dokumen otomatis menggunakan algoritma *LSA* dengan variasi *TF-IDF* yang telah dilakukan, dapat disimpulkan bahwa algoritma ini dapat digunakan untuk algoritma peringkas dokumen otomatis dengan Hasil Ringkasan Menggunakan Algoritma LSA dan TF-IDF yang memiliki kemiripan bervariatif sesuai dengan jumlah kompresi kalimat. Untuk tingkat kemiripan dokumen Hasil Ringkasan Menggunakan Algoritma LSA dan TF-IDF sistem dengan hasil pakar ketiga kategori kompresi kalimat memiliki nilai yang tidak berbeda jauh. Dari nilai *f-measure* ketiga kategori kompresi akan lebih baik jika minimal kompresi kalimat yang diringkas adalah sebanyak 10 kalimat karena mendekati 50%.

4.4. Aplikasi Hasil Penelitian

Hasil penelitian ini bertujuan untuk mengetahui efektivitas algoritma peringkas dokumen ilmiah tunggal otomatis berbahasa Indonesia dengan algoritma *Latent semantic analysis (LSA)* dengan variasi algoritma *Term-Frequency - Inverse Document Frequency* (*TF-IDF).* Melalui hasil penelitian ini, diharapkan dapat dimanfaatkan sebagai bahan pertimbangan bagi *programmer* dalam merancang sistem peringkas dokumen ilmiah tunggal otomatis berbahasa Indonesia dan menjadi bahan referensi penelitian tentang peringkas dokumen ilmiah otomatis berbahasa indonesia dengan menggunakan algoritma (*LSA) dan (TF-IDF).*

5. Kesimpulan dan Saran

5.1. Kesimpulan

Hasil kinerja yang didapatkan pada penelitian peringkas dokumen otomatis algoritma *LSA* dengan variasi *TF-IDF* memiliki kinerja yang lebih baik jika minimal kompresi ringkasan yang dihasilkan sistem adalah sebanyak 10 kalimat. Sebaliknya, jika kompresi kalimat yang dihasilkan ringkasan sistem adalah kurang dari 10 kalimat yakni 5 kalimat maka kinerjanya relatif kurang baik. Pada pengujian *ROUGE-1* didapatkan hasil pengujian optimal tertinggi dengan kompresi sebanyak 20 kalimat dengan nilai rata-rata *presisi 0,529, recall 0,562, dan f-measure 0,538*. Kemudian hasil optimal dihasilkan dengan kompresi sebanyak 10 kalimat dengan nilai rata-rata nilai *presisi* 0,493 *recall 0,* 518 dan *f-measure* 0, 499. Nilai rata-rata *presisi, recall dan f-measure* kedua kategori kompresi kalimat tersebut mendekati 50%. Melalui hasil penelitian ini diharapkan dapat membantu memberikan informasi terkait algoritma *LSA* dan *TF-IDF*, serta menjadikan bahan referensi maupun perbandingan peringkasan dokumen *single* ilmiah berbahasa indonesia dengan menggunakan algoritma *LSA* dan *TF-IDF*.

5.2. Saran

Untuk penelitian dan analisis lebih lanjut disarankan untuk melakukan hal sebagai berikut :

1. Dalam penentuan topik untuk peringkasan *single* dokumen ilmiah sebaiknya mencari topik dengan lebih sedikit gambar, tabel dan rumus.
2. Perlu dilakukan penelitian dan kajian lebih lanjut menggunakan algoritma lain sebagai variasi sehingga mendapatkan hasil penelitian yang lebih beragam.
3. Jika menggunakan *tools* NLTK dan *library* Sastrawi, sebaiknya memastikan dahulu bahwa *tools* dan *library* tersebut dalam versi terbaru agar *corpus* yang dihasilkan lebih baik.
4. Untuk proses peringkasan manual sebaiknya mencari pakar atau ahli dibidang tersebut minimal Dosen atau Guru Bahasa Indonesia

Daftar Pustaka:

Altszyler, E., Ribeiro, S., Sigman, M., & Slezak, D. F. (2017). The interpretation of dream meaning: Resolving ambiguity using Latent Semantic Analysis in a small corpus of text. Consciousness and cognition, 56, 178-187.

Darmawan, I., Harianto, R. A., & Armanto, H. (2017). Peringkasan Teks Model Graf Pada *Single* Dokumen Dengan Metode Sparse Non Negative Matrix Factorization. *Seminar Nasional Teknologi Dan Rekayasa (SENTRA)*, issn 2527-6050, V-1-V-11. Retrieved February 26, 2020, from http://research-report.umm.ac.id/index.php/sentra/article/view/1469/1661.

DOKUN, O., & CELEBI, E. (2017). *Single*-Document summarization using *Latent semantic analysis*. *International Journal of Scientific Research in Information Systems and Engineering (IJRSRISE)*, *1*(2), december2015, 1-12. Retrieved October 16, 2020, from https://www.researchgate.net/profile/Dokun_Oluwajana3/publication/27984 9764_*Single*-Document_summarization_using_Latent_Semantic_Analysis/links/559bd4a 508aee2c16df02373/*Single*-Document-summarization-using-Latent- Semantic-Analysis.pdf.

Evan, F. H., P., & W.P, Y. P. (2014). Pembangunan Perangkat Lunak Peringkas Dokumen dari Banyak Sumber Menggunakan Sentence Scoring dengan Metode TF-IDF. *Pembangunan Perangkat Lunak Peringkas Dokumen Dari Banyak Sumber Menggunakan Sentence Scoring Dengan Metode TF-IDF*, G-17. Retrieved 2020, from https://journal.uii.ac.id/index.php/Snati/article/view/3286

Ilyas, R., & Umbara, F. (2016). Peringkasan Otomatis Dengan Ekstraksi Informasi Untuk Dokumen Berita Ter-cluster. *ANNUAL RESEARCH SEMINAR 2016, 2 No. 1*, isbn : 979-587-626-0, 405-407. Retrieved 2020, from https://media.neliti.com/media/publications/170385-ID-peringkasan- otomatis-dengan-ekstraksi-in.pdf.

Luthfiarta, A., Zeniarja, J., & Salam, A. (2013). Algoritma *Latent semantic analysis* (LSA) Pada Peringkas Dokumen Otomatis Untuk Proses Clustering Dokumen. *SEMINAR NASIONAL TEKNOLOGI INFORMASI & KOMUNIKASI TERAPAN*, isbn : 979-26-0266-6, 13-19. Retrieved June 19, 2020, from http://eprints.dinus.ac.id/5177/1/P3-TI29-SEMANTIK- 92 Ardytha_Luthfiarta_[Universitas_Dian_Nuswantoro]_(FINAL).Pdf.

Mandar, G., & Gunawan, G. (2017). Peringkasan dokumen berita Bahasa Indonesia menggunakan metode Cross *Latent semantic analysis*. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, *3*(2), 94. doi:10.26594/register.v3i2.1161.

Moiyadi, H. S., Desai, H., Pawar, D., Agrawal, G., & Patil, N. M. (2016). NLP Based Text Summarization Using Semantic Analysis. *NLP Based Text Summarization Using Semantic Analysis, 2*(10), 2454-1331, 1812-1818. Retrieved 2020, from https://www.neliti.com/publications/239678/nlp- based-text-summarization-using-semantic-analysis.

Mustaqhfiri, M., Abidin, Z., & Kusumawati, R. (2012). Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance. *Matics*. doi:10.18860/mat.v0i0.1578

O., & Harahap, A. H. (2016). Implementasi Metode *Term*s Frequency-*Inverse document frequency* (TF-IDF) dan Maximum Marginal Relevance untuk Monitoring Diskusi Onlin. *Jurnal Sains, Teknologi Dan Industri,, 13, No. 2*, issn 1693-2390, 152-159.

Ridok, Achmad. (2014). Peringkasan Dokumen Bahasa Indonesia Berbasis Non- Negative Matrix Factorization (NMF). Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK). 1. 39-44. 10.25126/jtiik.201411104.

Yudiarta, N. G., Sudarma, M., & Ariastina, W. G. (2018). Penerapan Metode Clustering Text Mining Untuk Pengelompokan Berita Pada Unstructured Textual Data. *Majalah Ilmiah Teknologi Elektro, 17, No 3*, issn 2503-2372, 339-344. doi:DOI: https://doi.org/10.24843/MITE.2018.v17i03.P06.