

Discourse Connectors Pada Peringkasan Dokumen Berbahasa Indonesia

Irwan Darmawan^{a,*}, Nirwana Haidar Hari^b

^{a,b}Universitas Madura, Jl. Raya Panglegur Km 35, Pamekasan, Indonesia
email : ^adarmawan@unira.ac.id, ^bhaidar@unira.ac.id

Abstract— Documents that have a lot of content will make it difficult for readers to find the essence or topics in a document. Therefore, a system that can summarize documents is needed to find the topics discussed by the document. In the research, the documents used as test materials were the final project papers of informatics engineering students, Madura University. Various methods have been applied in summarizing a document, including using the cosine of similarity to determine the weight and the relationship between one sentence and another and then dumping the sentences which have small weight based on the threshold value of the user. If the length of a sentence has little weight, it does not mean that the sentence is not important when it comes to other sentences. Therefore a discourse connector is needed to connect one sentence to another sentence and then it is given weight according to the discourse connector formula. In this study, it is expected that the document has a better value than before as evidenced by a size that is higher than 70%.

IndexTerms— discourse connector; document; sentence; summary.

Abstrak— Dokumen yang memiliki isi sangat banyak akan menyulitkan bagi pembaca untuk menemukan intisari atau topik-topik dalam sebuah dokumen. Oleh sebab itu dibutuhkan sistem yang dapat meringkas dokumen untuk menemukan topik yang dibicarakan oleh dokumen tersebut. Dalam penelitian dokumen yang digunakan sebagai bahan uji coba adalah paper skripsi mahasiswa S-1 informatika di Universitas Madura Pamekasan. Berbagai macam metode telah diterapkan dalam meringkas sebuah dokumen diantaranya menggunakan cosinus similarity untuk menentukan bobot serta keterhubungan antar kalimat yang satu dengan kalimat yang lain kemudian membuang kalimat-kalimat yang memiliki bobot yang kecil berdasarkan nilai threshold dari user. Bila panjang sebuah kalimat memiliki bobot kecil belum tentu kalimat tersebut tidak penting ketika berhubungan dengan kalimat yang lain. Oleh sebab itu dibutuhkan discourse connector sebagai penghubung dari kalimat yang satu dengan kalimat yang lain kemudian diberikan bobot tersendiri sesuai rumus discourse connector tersebut. Pada penelitian ini diharapkan dalam meringkas dokumen memiliki nilai akurasi yang lebih baik dari pada sebelumnya yaitu dibuktikan dengan f-measure yang tinggi lebih dari 70%.

Kata Kunci— discourse connector; dokumen; kalimat; ringkasan.

I.

II. PENDAHULUAN

Dokumen yang sangat panjang atau memiliki isi yang sangat banyak memungkinkan membuat pembaca akan kesulitan mengetahui topik yang dibicarakan. Padahal yang dibutuhkan adalah inti sari informasi yang terkandung di dalam dokumen. Hal tersebut membuat tidak efektif bagi pembaca bila harus membaca seluruh isi dokumen kemudian menyimpulkannya. Oleh sebab itu dibutuhkan sebuah sistem yang dapat mendeteksi topik-topik inti yang dibicarakan dalam sebuah dokumen. Peringkasan dokumen dibedakan menjadi 2 macam, pertama *single document* dimana dokumen ini tersusun menjadi hanya satu bagian dalam dokumen tunggal. Sedangkan yang ke-dua adalah peringkasan menggunakan *multi document* atau terdiri dari banyak dokumen yang berbeda dalam meringkas dokumen[1].

Beberapa pendekatan yang digunakan dalam meringkas sebuah dokumen yang dikenal luas diantaranya adalah *metode compressive summarization* kemudian metode *extractive* serta *abstractive*. Metode *compressive summarization* mengambil inti dari sebuah input dengan memotong teks asli, tetapi tetap memperhatikan susunan kalimatnya. Kemudian metode *extractive* menghasilkan ringkasan dengan cara memotong bagian penting dari teks dalam sebuah dokumen kemudian menyusunnya kembali secara komprehensif. Sedangkan metode *abstractive* menghasilkan ringkasan dari tulisan yang sudah ada tanpa dibatasi untuk menggunakan kembali ungkapan ungkapan dari teks aslinya[2].

Terdapat berbagai macam metode dalam melakukan peringkasan dokumen salah satunya adalah cosinus similarity utk mengetahui bobot masing-masing kalimat didalam sebuah dokumen kemudian membuang kalimat yang memiliki bobot kecil sesuai dengan nilai threshold yang di inputkan oleh user.

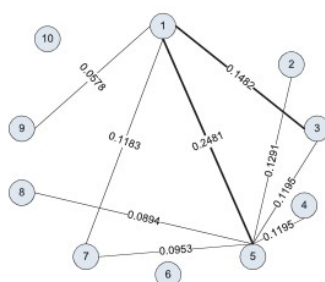
Pada tahapan selanjutnya dilanjutkan dengan metode graf tak berarah utk mengetahui seberapa dekat relasi antara kalimat yang satu dengan kalimat lainnya. Dalam membobotkan sebuah kalimat berdasarkan *cosinus similarity*-nya pasti terdapat perbedaan hasil bila menggunakan *discourse connector* dan tidak menggunakan *discourse connector*.

Pada penelitian sebelumnya telah mengujikan pada 3 orang pakar dan mengujicobakan pada 86 dokumen training dalam membuat peringkasan dokumen untuk mengetahui rata-rata nilai ambang batas pada peringkasan dokumen yaitu dengan nilai lambda yaitu (0,1, 0,2 dan 0,3) sedangkan parameter pada nilai dumping factor yaitu (0,2, 0,5 dan 0.8). Dalam penentuan akurasi, Setelah dirata-rata menghasilkan parameter tertinggi berada pada kombinasi lambda sama dengan 0,3 dan dumping factor sama dengan 0,8 dengan nilai recall 0.54627. Pada penelitian ini dengan melanjutkan penelitian sebelumnya yang telah diteliti peneliti mencoba melihat seberapa efektif dalam menghitung akurasi pada peringkasan dokumen apabila menggunakan *discourse connector* dalam meringkas sebuah dokumen dengan tetap menggunakan parameter yang sama.

II. TINJAUAN PUSTAKA

1. Pembobotan Kalimat Menggunakan Discourse Connectors Model Graf Pada Cluster Dan Rangkaian Kalimat

Peringkasan pada doukem baik berupa dokumen single ataupun multi dokumen biasanya menggunakan algoritma algoritma tertentu serta menghasilkan sebuah ringkasan dokumen tanpa menambahkan informasi tambahan apapun. Selanjutnya terdapat berbagai macam bentuk algoritma yang tertentu dalam melakukan meringkas dari dokumen berdasarkan query khusus ataaupun topik khusus yang digunakan yaitu dengan cara memproses ekstraksi pada kalimat yang dianggap penting dari isi dokumen (Shuzhi Sam dkk: 2013). kemudian didapatkan kemudian didapatkan hasil dari perhitungan semua kalimat yang sudah mengalami pengekstraksian tersebut, tahap selanjutnya adalah memilih kalimat dengan skor yang tinggi. Kemudian membuat sebuah model graf untuk dokumen, setelah pembobotan dengan model graf didapatkan dengan bobot tertinggi maka tahap berikutnya adalah mengimplementasikan algoritma ranking kalimat seperti algoritma PageRank (Ailin li dkk :2016) untuk menghitung skor dari sebuah kalimat dengan cara di iterasi sampai mencapai kestabilan.



Gambar 1 Ilustrasi Graph Model

Pada model graf diatas terdapat beberapa model hubungan kalimat yang satu dengan yang lain yang memiliki masing-masing bobot yang berbeda dimana pada model graf gambar diatas masing-masing bobot antar kalimat diambil beberapa bobot antar kalimat yang memiliki bobot terbesar kemudian membuang atau mengabaikan bobot antar kalimat yang memiliki bobot kecil, hubungan bobot antar kalimat tersebut didapat dari menghitung similaritas antar kalimat dimana similaritas tersebut dihitung dari metode term frequency-invers document frequency (TF_IDF). Jika sebuah kalimat memiliki nilai lebih dari nol maka kalimat tersebut memiliki hubungan, hubungan antar tersebut dinotasikan dengan kalimat S_i dan kalimat S_j Rumus persamaannya adalah :

$$w_{ij} = \lambda w_{sim}(s_i, s_j) + (1 - \lambda) w_{dis}(s_i, s_j)$$

Dimana $w_{sim}(s_i, s_j)$ adalah kesamaan cosinus antara vektor dari dua kalimat dan $w_{dis}(s_i, s_j)$ adalah bobot kalimat itu dan $\lambda \in [0,1]$ adalah parameter yang menyeimbangkan bobot kesamaan, dan jika sebuah kalimat menghubungkan ke dirinya sendiri w_{ii} maka nilainya 0 (nol) Sedangkan rumus w_{dis} adalah :

$$w_{dis}(s_i, s_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Dimana nilai -1 adalah diawali dengan discourse connectors (DC) seperti kata “because”, “after”, “before” dan 0 (nol) jika dipisahkan dengan tanda titik atau koma (Shuzhi Sam Ge dkk 2013). Contoh dari discourse connectors dapat dilihat pada tabel.1.

Tabel 1. Contoh kata-kata discourse connector

Jenis Kata	Contoh
Expansion	Selain itu, misalnya, seperti
Cause-effect	Karena, alasan itulah, karenanya, jadi
temporal	Sebelum, setelah, baru-baru ini, sekarang, nanti, masa lalu
Comparison	Walaupun, tapi, sementara, sebaliknya, bagaimanapun
Hubungan	Penanda wacana

Metode *Cosine Similarity* merupakan metode yang berfungsi untuk membandingkan kemiripan antar dokumen, dalam hal ini yang dibandingkan adalah query dengan dokumen latih. Dalam menghitung cosine similarity, hal pertama yang dilakukan yaitu melakukan perkalian skalar antara query dengan dokumen kemudian dijumlahkan, setelah itu melakukan perkalian antara panjang dokumen dengan panjang query yang telah dikuadratkan, setelah itu di hitung akar pangkat dua. Selanjutnya hasil perkalian skalar tersebut di bagi dengan hasil perkalian panjang dokumen dan query. Berikut adalah rumus dari cosinus similarity :

$$CosSim(d_i, q_i) = \frac{\sum_{j=1}^t q_j d_j}{\|q_i\| \|d_i\|} = \frac{\sum_{j=1}^t (q_j \cdot d_j)}{\sqrt{\sum_{j=1}^t (q_j)^2} \cdot \sqrt{\sum_{j=1}^t (d_j)^2}}$$

Keterangan :

q = bobot istilah j pada dokumen $i = df$

d = bobot istilah j pada dokumen $i = df$

Pada proses berikutnya adalah menghitung peringkat pada setiap kalimat yang memiliki hubungan antar kalimat dengan menggunakan model graf tersebut. Dalam menghitung peringkat antar kalimat digunakan algoritma pemeringkatan atau yang lebih dikenal dengan algoritma pagerank, algoritma pagerank akan menghitung peringkat kalimat dengan cara melakukan iterasi sampai dicapai posisi kalimat atau susunan kalimat yang paling tepat berdasarkan iterasi yang dilakukan oleh algoritma tersebut, berikut algoritma dalam menghitung peringkat setiap kalimat :

$$r(u_i) = d \sum_{j \in N(u_i)} \left(\frac{r(u_j)}{\text{deg}(u_j)} \right)^d$$

Dimana $r(u_i)$ dan u_j dua vertek pada graf serta d adalah parameter antara 0 dan 1. Pada permasalahan ini tidak perlu dilakukan perangkuman isi dari dokumen lain (multi dokumen) karena yang akan dibandingkan hanya antar kalimat pada satu dokumen .

2.Algoritma SNMF (Sparse Non Negative Matrix Factorization)

Algoritma SNMF merupakan algoritma unsupervised learning. Yaitu algoritma yang bertujuan untuk mengekstraksi variabel tersembunyi pada data pembelajaran.Selanjutnya data tersebut digunakan untuk clustering. Algoritma SNMF merupakan modifikasi atau penyempurnaan dari algoritma NMF(non negative matrix factorization) (Patrik O. Hoyer:2004). Oleh karena untuk mencari hasil yang lebih baik dari cluster menggunakan NMF maka ditambahkan nilai sparseness pada NMF. Sparseness disini berfungsi untuk mencari nilai kesesuaian pada masing-masing elemen matrik pada matrik bentukan sehingga didapatkan bentuk matrik yang sudah singular. Matrik yang singular adalah matrik yang dapat difaktorisasikan karena memiliki determinan sama dengan nol.

Untuk menentukan nilai pada masing-masing elemen matrik yang akan diolah oleh SNMF, maka diambil dari bobot tf-idf masing-masing kata terhadap masing-masing kalimat atau dikenal dengan istilah term-sentence yang sudah dipotong bobot grafnya dan yang akan dihitung atau diproses oleh SNMF adalah bobot- bobot kalimat yang memiliki ranking tinggi, Dari perhitungan sentence ranking atau perangkungan kalimat yang sudah tersusun dengan baik berdasarkan hasil dari iterasi antar kalimat, maka didapatkan matrik hubungan antar kalimat yang non singular yaitu matrik A. Matrik A adalah matrik yang tidak dapat di faktorisasikan karena determinannya tidak dapat dihitung, oleh karena itu dibuatkan matrik yang nilainya mendekati matrik yang non singular tersebut (matrik A) dengan menggunakan metode SNMF (sparse non negative faktorization). Setelah didapatkan matrik yang singular yaitu matrik V dimana nilai matrik V mendekati matrik yang non singular (matrik A),maka hasil faktorisasi dari matrik V adalah matrik W dan matrik H ($A \approx WH$), dimana matrik W adalah topik dari beberapa kalimat dalam satu dokumen dan matrik H adalah kedekatan antar topik dalam satu dokumen. Jadi masing-masing kalimat di clusterkan berdasarkan topiknya. Pembagian topik berdasarkan dimensi kolom dari matrik W,

sedangkan kedekatan antar kalimat pada topik tersebut berdasarkan nilai dari kolom masing-masing baris kalimat pada matrik H. Sehingga dapat dikatakan kalimat yang terdapat dalam satu topik yang sama adalah satu cluster. Topik tersebut ditunjukkan dengan hasil matrik W. Berikut adalah algoritma SNMF.

3. Rouge 2.0

ROUGE 2.0 adalah toolkit evaluasi yang mudah digunakan untuk tugas-tugas peringkasan otomatis. ROUGE 2.0 menggunakan sistem yang bekerja dengan membandingkan ringkasan atau terjemahan yang dihasilkan secara otomatis dengan serangkaian rujukan referensi (biasanya yang dibuat oleh seseorang atau pakar). ROUGE adalah salah satu cara untuk menghitung efektivitas ringkasan yang dihasilkan secara otomatis. Pada paper yang ditulis oleh kavita ganesan dijelaskan bahwa terdapat beberapa perbaikan ukuran untuk perbandingan ringkasan yang dibuat oleh manusia dan yang dihasilkan oleh sistem dalam menggunakan ROUGE, beberapa jenis ROUGE yang mengalami perbaikan adalah ROUGE-N + Sinonim, ROUGE-Topic, ROUGE-Topic + Sinonim, ROUGE-TopicUniq dan ROUGE-TopicUniq + Sinonim.

ROUGE secara default adalah lebih menekankan kepada ukuran berorientasi kepada recall. Cara menghitung ukuran antara ringkasan sistem dan ringkasan ideal adalah menghitung jumlah kata yang terdapat antara ringkasan sistem dan ringkasan ideal tersebut atau disebut n-gram. ROUGE terbukti secara efektif dapat mengevaluasi antara ringkasan sistem dan ringkasan ideal. Permasalahan yang terjadi adalah ROUGE sebelumnya tidak memberikan ukuran yang pasti tentang kinerja ringkasan sistem dibandingkan dengan ringkasan manusia. Sebagai contoh, skor recall ROUGE-1 sebesar 0,30 hanya mengatakan bahwa 30% dari konten dalam ringkasan referensi telah diambil oleh ringkasan sistem. Maka tampak seperti jumlah yang sangat rendah, skor ini tidak mempertimbangkan konsep-konsep sinonim atau persamaan kata. Contoh kata "layar" dan kata "tampilan" adalah dua kata berbeda tetapi memiliki makna yang sama. ROUGE sebelumnya tidak mempertimbangkan sinonim kata maka kata tersebut dianggap berbeda sehingga mengurangi nilai recall, precision dan f-score.

4. Hipotesis

Hipotesis dalam penelitian ini adalah peringkasan single-dokumen yang dilakukan diharapkan dapat mendeteksi ringkasan dari dokumen abstrak dari paper tugas akhir mahasiswa Universitas Madura yang tidak sesuai dengan isi paper tugas akhir. Metode yang diusulkan diharapkan mampu mendeteksi dengan nilai akurasi minimal 75 % sesuai dengan pakar setelah ditambahkan perhitungan discourse connectors. Nilai akurasi diukur dari nilai f-measure dari tebakan sistem dibandingkan dengan pakar.

5. Penelitian Sebelumnya

Beberapa penelitian yang sudah dilakukan sebelumnya sudah menghasilkan nilai kurang baik. Beberapa penelitian tersebut antara lain:

Irwan darmawan (2018), dalam penelitian tersebut dihasilkan 70 % kesesuaian abstrak terhadap isi dokumen tugas akhir mahasiswa prodi teknik informatika Universitas Madura.

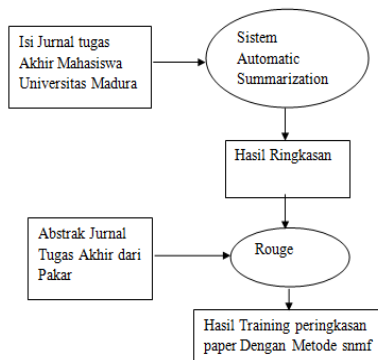
III. METODE PENELITIAN

Penelitian ini merupakan penelitian yang dilakukan di laboratorium dalam tahapan memasukkan kedalam sistem peringkasan dokumen otomatis baik dokumen yang diujikan maupun dokumen yang akan dilatih. Tahapan penelitian yang akan dilaksanakan dijelaskan dalam Gambar 2.

Detail kegiatan penelitian yang dilakukan sebagai berikut:

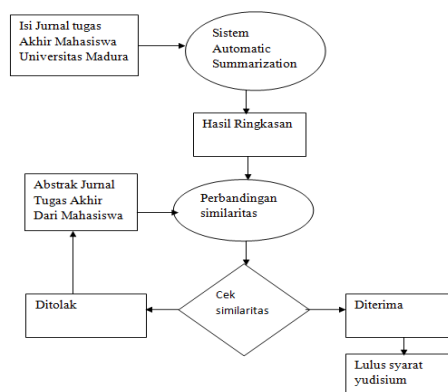
1. Pengumpulan dokumen paper tugas akhir mahasiswa universitas madura di program studi teknik informatika.
2. Mengimplementasikan sistem peringkasan otomatis dokumen dengan menggunakan discourse connectors.
3. Mengujikan dokumen pada 3 orang pakar bahasa indonesia dalam membuat abstrak kemudian dimasukkan ke dalam sistem peringkasan otomatis untuk menentukan nilai rata-rata terbaik dalam menentukan nilai threshold terhadap data yang diujikan.
4. Mengujikan dan menentukan diterima atau ditolaknya oleh sistem peringkasan otomatis tugas akhir mahasiswa dengan menggunakan nilai threshold yang diperoleh dari pakar.
5. Membandingkan dengan penelitian peringkasan dokumen otomatis yang sebelumnya yang telah dilakukan dengan penelitian yang sudah menggunakan discourse connectors dalam meringkas sebuah dokumen.

Blok diagram pada Gambar 1 menunjukkan posisi pemanfaatan dari program. Di bawah ini adalah blok diagram dalam menentukan nilai trining peringkasan paper dengan metode SNMF yang dilakukan oleh pakar.



Gambar 2. Proses dokumen training

Blok diagram pada gambar satu menjeaskan langkah awal dalam menyusun atau menghasilkan ringkasan secara otomatis dalam mentraining data ringkasan dokumen untuk digunakan sebagai pembandingan pada data testing berikutnya. Pada awalnya jurnal atau paper mahasiswa dimasukkan ke sistem peringkasan otomatis kemudian hasil ringkasan otomatis tersebut dihitung didalamrouge dan di bandingkan dengan abstrak yang dibuat oleh pakar maka hasil akhir dari perbandingan tersebut menghasilkan ringkasan data training dalam hal ini menggunakan SNMF yang akan dijadikan acuan dalam penentuan kesesuaian abstrak terhadap isi dari paper tugas akhir mahasiswa informatika.



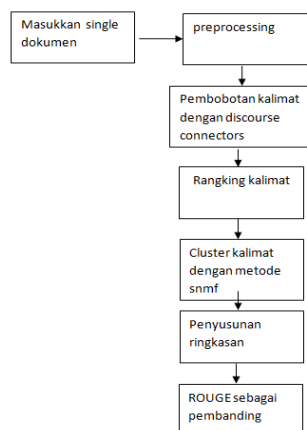
Gambar 3 Testing dokumen

Setelah melakukan data training pada tahap selanjutnya adalah melakukan tahapan testing terhadap isi dari paper tugas akhir mahasiswa, hal pertama yang dilakukan adalah memasukkan isi dari paper tugas akhir mahasiswa kedalam sistem peringkasan otomatis kemudian setelah dimasukkan akan dihasilkan sebuah ringkasan berupa abstrak dari sistem. Pada tahap berikut abstrak dari paper tugas akhir mahasiswa tersebut akan dibandingkan dengan hasil ringkasan dari sistem, berdasarkan nilai threshold yang sudah ditentukan pada data latih bila abstrak sesuai dengan isinya maka mahasiswa diloloskan dalam yudisium bila tidak sistem akan menolak serta mahasiswa disarankan untuk membuat abstrak ulang yang lebih sesuai.

Berikut adalah alur proses dari awal pada sistem untuk pemeriksaan kesesuaian abstrak terhadap isi paper tugas akhir memiliki langkah-langkah sebagai berikut:

- ➊ Awal preprocessing.
- ➋ Bobot masing-masing kalimat menggunakan discourse connectors.
- ➌ Ranking kalimat.
- ➍ Cluster Kalimat.
- ➎ Pemilihan Kalimat Representatif.
- ➏ Penyusunan Ringkasan.
- ➐ Perbandingan hasil ringkasan kalimat dengan abstrak.

Berikut adalah blok diagram dalam penyusunan peringkasan dokumen otomatis dengan menyertakan discourse connectors



Gambar 4 Proses sistem peringkasan otomatis dengan menyertakan *discourse connectors*

1. *Preprocessing*.

Pada tahapan preprocessing meliputi Tahapan *tokenizing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya, pada permasalahan ini satu dokumen dibaca secara keseluruhan isinya yang berupa teks serta hanya mengambil isi dari teksnya saja dan file tersebut bertipe PDF, penanganan seluruh subbab akan dianggap satu dokumen. Selanjutnya tahapan case folding adalah tahapan merubah huruf kapital menjadi lowercase, kemudian tahapan stop word dan menghilangkan tanda baca kecuali tanda titik (sebagai pembatas akhir kalimat) dan tidak dihilangkan. Stop words merupakan kata umum yang dimanfaatkan dalam information retrieval yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna, contoh stop words dalam bahasa Indonesia: “adalah”, “dia”, “dan”, dan lain-lain. Stemming adalah pemotongan kata dengan tujuan mencari kata dasar dari kata dalam dokumen.

2. Bobot kalimat menggunakan *discourse connectors*.

Setelah tahapan preprocessing dilalui maka tahap berikutnya adalah menentukan bobot perkalamat. Untuk memperoleh bobot perkalamat pertama kali yang harus dilakukan adalah menghitung bobot perkata dengan menggunakan metode tf-idf. Setelah ditemukan masing-masing bobot perkata didalam satu dokumen langkah selanjutnya adalah menentukan panjang vektor masing-masing kalimat kemudian menghitung cosinus similarity atau tingkat kemiripan antar kalimat yang satu dengan kalimat yang lain. Apabila nilai cosinus similarity memiliki nilai 0 (nol) berarti kalimat tersebut tidak memiliki hubungan. Dari hasil *cosinus similarity* kemudian dicek apakah kalimat merupakan bagian dari discourse connectors apa bukan tahapan berikutnya kemudian dimasukkan ke pembobotan model graph. Pada model graph ini berdasarkan jurnal rujukan utama dikatakan satu kalimat apabila dipisahkan oleh titik (.) atau koma (,)

3. Rangkaing kalimat.

Pada tahap ini kalimat yang sudah memiliki bobot berdasarkan model graph di ranking menggunakan algoritma pagerank dengan tujuan untuk menemukan kalimat mana yang paling banyak memiliki hubungan atau keterkaitan dengan kalimat yang lain.

4. Cluster Kalimat dengan menggunakan SNMF.

Dalam melakukan cluster atau pengelompokan terhadap masing masing kalimat adalah menggunakan metode *sparse non negative matrix factorization* (SNMF).

IV.HASIL PENELITIAN

Setelah di lakukan beberapa pengamatan dan ujicoba dengan 86 dokumen paper dengan menggunakan parameter uji tetap seperti penelitian sebelumnya yaitu menggunakan lamda (0,1 0,3 dan 0,5) serta *dumping factor* (0,2 0,5 dan 0,8) yang sudah disertakan dengan kumpulan kata yang merupakan bagian dari discourse connectors, dimana pada tabel berisi tentang hasil dari ujicoba parameter dokumen training *lambda* dan *dumping factor* yang dilakukan oleh tiga orang pakar bahasa Indonesia Universitas Madura maka hasilnya dapat dilihat pada tabel 2:

Tabel 2. Hasil ujicoba parameter dokumen training lambda dan dumping factor

Dok ke	Parameter								
	(0.1), (0.2)	(0.1), (0.5)	(0.1), (0.8)	(0.3), (0.2)	(0.3), (0.5)	(0.3), (0.8)	(0.5), (0.2)	(0.5), (0.5)	(0.5), (0.8)
Dokumen-1	0.62	0.58	0.67	0.5	0.64	0.57	0.52	0.63	0.63
	0.60	0.49	0.62	0.69	0.59	0.58	0.56	0.66	0.67
	0.55	0.48	0.59	0.46	0.58	0.49	0.48	0.58	0.52

Dok ke	Parameter								
	(0.1),	(0.1),	(0.1),	(0.3),	(0.3),	(0.3),	(0.5),	(0.5),	(0.5),
	(0.2)	(0.5)	(0.8)	(0.2)	(0.5)	(0.8)	(0.2)	(0.5)	(0.8)
Dokumen-2	0.69	0.68	0.78	0.53	0.65	0.73	0.66	0.78	0.66
	0.63	0.58	0.65	0.64	0.65	0.73	0.66	0.66	0.53
	0.69	0.56	0.77	0.56	0.74	0.74	0.69	0.77	0.63
Dokumen-3	0.58	0.54	0.53	0.57	0.61	0.65	0.67	0.65	0.66
	0.57	0.53	0.54	0.65	0.63	0.65	0.65	0.63	0.63
	0.46	0.43	0.45	0.56	0.51	0.56	0.56	0.52	0.51
Dokumen-4	0.34	0.47	0.38	0.53	0.47	0.48	0.58	0.54	0.54
	0.39	0.58	0.36	0.67	0.45	0.47	0.68	0.67	0.59
	0.38	0.49	0.37	0.65	0.42	0.48	0.53	0.67	0.58
Dokumen-5	0.47	0.46	0.49	0.54	0.53	0.59	0.55	0.58	0.47
	0.55	0.59	0.48	0.74	0.63	0.77	0.77	0.65	0.67
	0.53	0.55	0.54	0.87	0.79	0.79	0.70	0.73	0.68
Dokumen-6	0.57	0.54	0.56	0.75	0.79	0.77	0.64	0.72	0.66
	0.55	0.58	0.56	0.77	0.77	0.79	0.60	0.77	0.65
	0.55	0.64	0.56	0.78	0.76	0.79	0.76	0.78	0.62
Dokumen-7	0.44	0.59	0.47	0.56	0.55	0.56	0.56	0.57	0.55
	0.49	0.53	0.52	0.63	0.66	0.78	0.65	0.65	0.57
	0.44	0.59	0.57	0.67	0.67	0.67	0.64	0.63	0.59
Dokumen-8	0.48	0.34	0.42	0.45	0.40	0.43	0.43	0.47	0.49
	0.44	0.49	0.32	0.44	0.57	0.44	0.53	0.57	0.56
	0.48	0.44	0.42	0.57	0.58	0.46	0.57	0.58	0.55
Dokumen-9	0.47	0.48	0.33	0.48	0.40	0.48	0.47	0.44	0.43
	0.44	0.44	0.44	0.46	0.57	0.59	0.56	0.57	0.52
	0.46	0.46	0.35	0.59	0.63	0.58	0.66	0.57	0.51
Dokumen-10	0.56	0.56	0.57	0.56	0.54	0.55	0.57	0.68	0.59
	0.67	0.65	0.55	0.59	0.75	0.65	0.67	0.70	0.67
	0.67	0.65	0.54	0.56	0.76	0.67	0.60	0.78	0.68

Pada tabel 2 dihasilkan pengujian parameter yang lebih baik daripada sebelumnya yang tidak menggunakan *discourse connectors* dalam penentuan parameter yang akan digunakan dalam meringkas dokumen dengan kombinasi yang sama yaitu lambda dan kombinasi parameter *dumping factor*, parameter untuk lambda adalah (0,1 0,3 dan 0,5) dan untuk *dumping factor* yaitu (0,2 0,5 0,8) meskipun parameter terbaik tetap berada pada lambda 0,3 dan *dumping factor* tetap berada pada 0,8 sebagai hasil pengujian terbaik. Hal itu dibuktikan dengan angka yang meningkat pada setiap dokumen yang dijadikan data latih untuk penentuan parameter terbaik untuk digunakan dalam sistem peringkasan dokumen.

Tabel 3. Perbandingan peringkasan dokumen sebelum menggunakan *discourse connector*

Dok ke	Parameter								
	(0.1),	(0.1),	(0.1),	(0.3),	(0.3),	(0.3),	(0.5),	(0.5),	(0.5),
	(0.2)	(0.5)	(0.8)	(0.2)	(0.5)	(0.8)	(0.2)	(0.5)	(0.8)
Dokumen-1	0.52	0.44	0.58	0.33	0.55	0.45	0.46	0.57	0.57
	0.43	0.36	0.57	0.56	0.43	0.47	0.47	0.55	0.54
	0.46	0.33	0.47	0.37	0.47	0.38	0.39	0.44	0.47
Dokumen-2	0.56	0.56	0.66	0.49	0.58	0.64	0.54	0.58	0.54
	0.57	0.38	0.57	0.59	0.59	0.67	0.54	0.54	0.47
	0.58	0.47	0.65	0.46	0.54	0.64	0.55	0.53	0.56
Dokumen-3	0.43	0.43	0.46	0.44	0.54	0.59	0.56	0.57	0.58
	0.47	0.44	0.47	0.56	0.53	0.55	0.59	0.53	0.55
	0.36	0.37	0.38	0.38	0.49	0.43	0.47	0.43	0.49
Dokumen-4	0.27	0.36	0.28	0.46	0.37	0.39	0.47	0.45	0.45
	0.27	0.44	0.29	0.43	0.37	0.33	0.54	0.58	0.42
	0.24	0.33	0.25	0.58	0.35	0.36	0.43	0.44	0.44
Dokumen-5	0.33	0.33	0.34	0.43	0.48	0.40	0.45	0.48	0.38
	0.41	0.34	0.38	0.67	0.55	0.63	0.53	0.55	0.54
	0.45	0.46	0.45	0.74	0.68	0.66	0.64	0.65	0.57
Dokumen-6	0.43	0.46	0.44	0.56	0.65	0.63	0.57	0.63	0.55
	0.43	0.45	0.46	0.56	0.68	0.69	0.57	0.66	0.56
	0.45	0.53	0.48	0.67	0.64	0.63	0.68	0.68	0.57
Dokumen-7	0.37	0.44	0.39	0.48	0.46	0.44	0.46	0.43	0.45
	0.38	0.44	0.45	0.55	0.53	0.65	0.55	0.55	0.46
	0.34	0.48	0.45	0.54	0.53	0.57	0.54	0.47	0.45
Dokumen-8	0.32	0.27	0.38	0.38	0.38	0.29	0.38	0.33	0.38
	0.38	0.36	0.29	0.34	0.44	0.34	0.46	0.44	0.49
	0.36	0.34	0.34	0.47	0.49	0.33	0.48	0.47	0.36
Dokumen-9	0.36	0.35	0.22	0.33	0.36	0.25	0.38	0.35	0.36
	0.35	0.36	0.34	0.32	0.45	0.48	0.43	0.35	0.45
	0.36	0.38	0.25	0.47	0.45	0.45	0.55	0.48	0.43
Dokumen-10	0.48	0.46	0.45	0.44	0.49	0.43	0.46	0.55	0.42
	0.56	0.52	0.47	0.46	0.67	0.56	0.55	0.69	0.51
	0.55	0.53	0.45	0.48	0.64	0.55	0.55	0.65	0.50

Dengan menggunakan pendekatan data recall sebagai nilai maksimum atau nilai terbaik maka didapat nilai rata-rata recall terbaik yaitu 0,54627. Nilai recall tersebut dihitung dengan membandingkan abstrak yang dibuat oleh pakar dengan ringkasan dokumen yang dihasilkan oleh sistem dengan bantuan rouge sebagai pengukur akurasi dari recall tersebut. Recall terbaik didapat pada kombinasi para meter uji yaitu lamda 0,3 serta *dumping factor* 0,8. Pada data ujicoba bila mahasiswa memasukkan isi dari dokumen paper kemudian dibandingkan dengan abstraknya jika tidak memenuhi nilai ambang recall yang telah ditentukan maka sistem otomatis akan menolak serta mahasiswa harus membuat ulang kesesuaian abstrak terhadap isi dari dokumen paper tugas akhirnya. Untuk perhitungan recallnya tersebut dapat dilihat pada perhitungan tabel 4.

Tabel 4. Nilai recall dari proses uji coba dengan ringkasan dokumen

No	Parameter								
	(0.1),(0.2)	(0.1),(0.5)	(0.1),(0.8)	(0.3),(0.2)	(0.3),(0.5)	(0.3),(0.8)	(0.5),(0.2)	(0.5),(0.5)	(0.5),(0.8)
1	0.84	0.77	0.86	0.88	0.83	0.77	0.75	0.85	0.86
2	0.89	0.82	0.91	0.8	0.89	0.97	0.83	0.89	0.83
3	0.73	0.77	0.75	0.85	0.86	0.84	0.88	0.83	0.83
4	0.56	0.7	0.57	0.81	0.67	0.68	0.8	0.81	0.78
5	0.77	0.71	0.76	1.02	0.9	0.97	0.92	0.96	0.88
6	0.78	0.81	0.78	0.9	0.97	0.97	0.92	0.92	0.88
7	0.66	0.72	0.7	0.85	0.87	0.9	0.86	0.85	0.77
8	0.67	0.62	0.61	0.71	0.73	0.62	0.73	0.77	0.71
9	0.67	0.62	0.6	0.71	0.79	0.78	0.8	0.75	0.74
10	0.86	0.84	0.78	0.75	0.92	0.8	0.8	0.91	0.87
11	0.79	0.8	0.77	0.69	0.9	0.81	0.86	0.84	0.75
12	0.95	0.91	0.89	0.84	0.83	0.79	0.81	0.95	0.9
13	0.74	0.84	0.79	0.81	0.74	0.74	0.82	0.79	0.7
14	0.59	0.71	0.65	0.72	0.73	0.72	0.73	0.86	0.79
Rata-rata	0.77	0.77	0.76	0.83	0.81	0.84	0.83	0.83	0.81

Dapat disimpulkan bahawa dalam penggunaan *discourse connector* nilai atau hasil parameter yang digunakan meningkat daripada hasil pengujian sebelumnya yaitu menghasilkan 0,85 yang dijadikan acuan nilai dalam penentuan recall dan precision dan f-score nya seperti yang ditunjukkan oleh tabel hasil dari recall, precision dan f-score dibawah ini

Tabel 5. Nilai Recall, Precision dan F-score berdasarkan uji coba ringkasan dokumen

Dok ke	Recall	Precision	F-Score	Recall	Precision	F-Score	Recall	Precision	F-Score
	(0.1), 0.85	(0.1), 0.9	(0.1), 0.85	(0.3), 0.9	(0.3), 0.85	(0.3), 0.85	(0.5), 0.9	(0.5), 0.9	(0.5), 0.9
1	0.62	0.15	0.21	0.56	0.14	0.19	0.57	0.14	0.19
	0.61	0.13	0.18	0.54	0.12	0.17	0.57	0.13	0.17
	0.51	0.19	0.26	0.46	0.17	0.23	0.49	0.18	0.25
2	0.62	0.12	0.16	0.57	0.11	0.15	0.59	0.13	0.18
	0.55	0.11	0.15	0.61	0.12	0.16	0.59	0.13	0.19
	0.61	0.12	0.17	0.58	0.12	0.17	0.63	0.14	0.2
3	0.52	0.14	0.2	0.53	0.15	0.21	0.66	0.16	0.23
	0.55	0.16	0.22	0.57	0.16	0.23	0.64	0.17	0.24
	0.46	0.15	0.21	0.46	0.16	0.22	0.56	0.17	0.24
4	0.46	0.12	0.16	0.48	0.13	0.18	0.59	0.13	0.19
	0.5	0.12	0.16	0.46	0.12	0.16	0.47	0.11	0.15
	0.46	0.12	0.16	0.47	0.12	0.17	0.51	0.13	0.18
5	0.45	0.13	0.18	0.5	0.14	0.2	0.56	0.17	0.24
	0.52	0.13	0.17	0.59	0.14	0.19	0.71	0.18	0.26
	0.52	0.11	0.15	0.66	0.13	0.18	0.66	0.15	0.21
6	0.37	0.11	0.15	0.37	0.13	0.17	0.38	0.11	0.14
	0.44	0.11	0.15	0.46	0.13	0.17	0.45	0.11	0.14
	0.45	0.1	0.13	0.48	0.11	0.15	0.49	0.1	0.13
7	0.47	0.12	0.17	0.57	0.16	0.22	0.52	0.17	0.23
	0.48	0.11	0.15	0.61	0.14	0.2	0.58	0.15	0.21
	0.49	0.11	0.14	0.66	0.13	0.19	0.63	0.14	0.2
8	0.4	0.12	0.17	0.44	0.1	0.14	0.43	0.12	0.16
	0.37	0.1	0.13	0.46	0.09	0.12	0.48	0.1	0.13
	0.38	0.1	0.13	0.41	0.09	0.11	0.49	0.1	0.13
9	0.33	0.1	0.13	0.4	0.13	0.18	0.36	0.12	0.15
	0.35	0.1	0.13	0.48	0.16	0.22	0.42	0.13	0.18
	0.31	0.11	0.14	0.55	0.18	0.26	0.42	0.14	0.19
10	0.51	0.12	0.16	0.59	0.14	0.19	0.62	0.12	0.16
	0.51	0.11	0.15	0.68	0.14	0.19	0.68	0.12	0.16
	0.51	0.11	0.14	0.66	0.13	0.18	0.69	0.11	0.15

REFERENCES

- [1] K. Sarkar, "Automatic Single Document Text Summarization Using Key Concepts in Documents", *J Inf Process Syst*, vol. 9 no.4, pp. 602-620, 2013.
- [2] Alex Alifimoff, "Abstractive sentence Summarization with Attentive Deep RecurrentNeural Networks," Stanford University, 2015.
- [3] M. Adriani, and B. Nazief, 1996. Confix-Stripping: Approach to Stemming Algorithm for Bahasa Indonesia. Faculty of Computer Science University of Indonesia.
- [4] Ailin Li, Tao Jiang, Qingshuai Wang, Hongzhi Yu (2016), "The Mixture of TextRank and LexRank Techniques of Single Document Automatic Summarization Research in Tiben", *Journal of International Conference on Intelligent Human-Machine Systems and Cybernetics*, IEEE (2016) hal. 514-519.
- [5] Anonymous, 2015. The Rouge Resource Network. [Online] Available at:<http://rxnlp.com/rouge-2-0> [Accessed 21 Mei 2019].
- [6] Anyman El-Kilany, Iman Saleh (2012), "Unsupervised Document Summarization Using Clusters of Dependency Graph Nodes", *Journal of International Conference on Intelligent Systems Design and Applications (ISDA)*, IEEE (2012), hal557-561.
- [7] Anonymous, 2015. The Rouge Resource Network. [Online] Available at:<https://github.com/nmfpack/code/nmfsc.m> [Accessed 22 Januari 2018].
- [8] Asian J. (2007) "*Effective Techniques for Indonesian Text Retrieval*". PhD thesis School of Computer Science and Information Technology RMIT University Australia.
- [9] Asian, J., Williams, H. E., & Tahaghoghi, S. M. M. (2005). Stemming Indonesian. In *Conferences in Research and Practice in Information Technology Series*, Vol. 38, Hal. 307-314.
- [10] Babuska, R. (2009). *Fuzzy and Neural Control*. Netherlands: Delft University of Technology. Daniel D. Lee dan H. Sebastian Seung, Algorithms for Non-negative Matrix Factorization
- [11] Jyoti Bora, D. dan Anil Kumar Gupta. (2014) . A Comparative Study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm dalam

Irwan Darmawan, Meraih gelar sarjana informatika (S.Kom) dari Universitas Trunojoyo Bangkalan pada tahun 2010. Kemudian meraih gelar Master (M.Kom) dari Sekolah Tinggi Teknik Surabaya pada tahun 2018. Saat ini Penulis menjadi dosen program studi Informatika di Universitas Madura Pamekasan.

Nirwana Haidar Hari, Meraih gelar sarjana (S.Pd) dari Universitas Negeri Malang pada tahun 2011. Kemudian meraih gelar Master (M.Kom) dari STMIK Amikom pada tahun 2016. Saat ini Penulis menjadi dosen program studi Informatika di Universitas Madura Pamekasan.