KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)
Terakreditasi Nomor 204/E/KPT/2022 | Vol. 4, No. 3, Juli (2023), pp. 689-699

Implementasi Pencarian Link graph terbaik dalam menentukan Kalimat Representatif Pada Peringkasan Dokumen Berbahasa Indonesia

Irwan Darmawan[1], Sholeh Rachmatullah[2], Nilam Ramadhani[3]
[1,2,3]Prodi Informatika, Fakultas Teknik, Universitas Madura, Pamekasan, Indonesia
E-mail: [1]darmawan@unira.ac.id, [2]sholeh@unira.ac.id, [3]nilamramadhani@unira.ac.id

Abstract

Indonesian documents have many or several sentences that are considered important in summarizing one document or many documents, sentences that appear a lot in documents are usually considered important sentences or reflect representative sentences on a document, even though these are not necessarily sentences that should be taken and put into representative sentences in determining the summarization of documents. In the link graph method used, it can indeed determine the amount of sentence weight in each sentence, it becomes a problem when sentence cutting is done if it is considered to have a very small weight. So a solution is needed to overcome this, namely by finding or determining the best parameters on the link graph of each sentence in the document summary. The value of such parameters is used to determine the truncation of each sentence. If the weight value of the sentence does not reach the limit of the parameter to be searched then the sentence is not included in the sentence processing in determining the summary of the document. The test parameters used are lambda i.e. (0.1, 0.3, 0.5) and for dumping factor i.e. (0.3, 0.5, 0.8).

Keywords: *Link, Graph, Summarization, Document*

Abstrak

Dokumen berbahasa Indonesia memiliki banyak atau beberapa kalimat yang dianggap penting dalam melakukan peringkasan pada satu dokumen maupun banyak dokumen, kalimat yang banyak muncul dalam dokumen biasanya dianggap kalimat penting atau mencerminkan kalimat representatif pada sebuah dokumen, padahal hal tersebut belum tentu merupakan kalimat yang seharusnya diambil dan dimasukkan ke kalimat representatif dalam menentukan peringkasan dokumen. Pada metode link graph yang digunakan memang dapat menentukan besaran bobot kalimat pada masing-masing kalimat, hal tersebut menjadi permasalahan ketika pemotongan kalimat dilakukan apabila dianggap memiliki bobot yang sangat kecil. Maka diperlukan solusi untuk mengatasi hal tersebut dengan cara mencari atau menentukan parameter terbaik pada link graph masing-masing kalimat dalam peringkasan dokumen. Nilai dari parameter tersebut digunakan untuk menentukan pemotongan masing-masing kalimat. Apabila nilai bobot kalimat tersebut tidak mencapai batasan parameter yang akan dicari maka kalimat tersebut tidak dimasukkan dalam pengolahan kalimat dalam menentukan ringkasan dokumen. Parameter uji yang digunakan adalah untuk dumping factor yang digunakan adalah (0,3, 0,5 dan 0,8) dan lambda adalah (0,1, 0,3 dan 0,5).

Keywords: *Link, Graf, Peringkasan, Dokumen*

1. Pendahuluan

Terdapat banyak cara dalam meringkas sebuah dokumen salah satu caranya adalah dengan meringkas dokumen dengan cara ektraktif atau abstraktif[1]. Ekstraktif bekerja

ISSN: 2720-992X

dengan memilih subset dari kata, frasa atau kalimat yang ada dari teks asli untuk membentuk ringkasan. Teknik ekstraktif menggunakan pendekatan statistik untuk memilih kalimat atau kata kunci untuk menghasilkan sebuah ringkasan[2]. Sedangkan abstraktif menghasilkan kalimat dari representasi semantik dan kemudian menggunakan teknik generasi bahasa alami untuk membuat ringkasan yang lebih dekat dengan apa yang dihasilkan manusia. Teknik abstraktif lebih menggunakan pendekatan linguistik untuk memahami teks asli untuk dapat menghasilkan ringkasan.

Pada penelitian ini terdapat beberapa permasalahan ketika dokumen akan di ringkas yaitu cara menentukan kalimat di dalam dokumen yang akan diambil sebagai kalimat yang akan diolah untuk dijadikan ringkasan dokumen atau kalimat yang menggambarkan kalimat representatif pada sebuah dokumen. Oleh karena itu metode link graf yang menggambarkan bobot pada masing-masing kalimat terhadap kalimat yang lain mempermudah kita untuk menentukan kalimat representatif tersebut untuk diolah menjadi ringkasan[3]. Permasalahan yang terjadi adalah cara menentukan besaran bobot kalimat yang akan diambil sebagai kalimat ringkasan, oleh karena itu dengan cara mencari nilai kombinasi terbaik antara dumping factor dan nilai lambda pada kalimat yang terdapat di link graf diharapkan mampu menghasilkan ringkasan yang lebih baik daripada langsung memotong kalimat yang memiliki bobot kecil padahal kalimat yang dimaksud belum tentu bukan merupakan kalimat yang harus dibuang dalam meringkas dokumen[3].
Untuk ujicoba menggunakan abstrak paper dari tugas akhir mahasiswa program studi Informatika di Universitas Madura dengan jumlah dokumen uji sebanyak 50 dokumen paper tugas akhirsebagai penentu dari kombinasi lambda dan dumping factor, sedangkan 50 dokumen paper tugas akhir sebagai data ujicoba peringkasan dokumen secara otomatis.

2. Metodologi Penelitian

2.1. Preprocessing

Preprocessing merupakan langkah awal yang harus dilalui sebelum dokumen mentah diolah menjadi sebuah ringkasan dokumen dengan menggunakan metode tertentu. Dalam hal ini ada beberapa tahapan preprocessing yang harus dilalui dengan tujuan agar mendapat data yang berkualitas[4], tahapannya sebagai berikut :

a. Tokenizing :proses ini membaca semua kata yang ada didalam dokumen.
b. Case folding adalah tahapan merubah huruf besar atau upper case menjadi lower case semua hal ini dimaksudkan supaya data kata yang terbaca memiliki nilai bobot yang sama pada tahap-tahap berikutnya.
c. Tahap stopword removal : pada tahapan ini melewati cek setiap kata yang termasuk didalam kata stoplist yang ada di basis data jika kata tersebut merupakan kata yang termasuk didalam stoplist maka kata tersebut dilewati atau tidak dimasukkan ke tahap berikutnya untuk menghitung bobot kata.
d. Tahap stemming : pada tahapan ini adalah menentukan kata dasar setiap kata yang terdapat didalam data base untuk dihitung bobot setiap kata pada tahapan berikutnya yaitu menggunakan TF-IDF

2.2. Stemmer Bahasa Indonesia

Pada buku yang ditulis oleh Jelita Asian B.Comp. Sc.(Hons.) Stemming lemmatization adalah proses merubah kata menjadi kata dasar dari kata yang memilki imbuhan. Dalam mengolah lemmatization harus menggunakan aturan ketatabahasaan[4] .

2.3. Algoritma Sastrawi Stemmer Bahasa Indonesia

Menemukan kata dasar pada sebuah kata adalah proses membuang imbuhan. Prosedur stemming terdiri dari beragam cara. Tabel pemenggalan terlampir adalah sumber dari teknik pertama. Dengan teknik ini, sebuah istilah dikelompokkan dengan memenggal

kepala imbuhan yang digunakan dengan menghilangkannya dari istilah menggunakan Tabel referensi [5]. Karena hasil stemming menggunakan metode ini dipengaruhi oleh pengumpulan dokumen yang digunakan dalam prosedur percobaan, cara kedua dikenal sebagai metode stemming berbasis korpus[2]. Metode pertama dikembangkan menjadi metode ketiga. Ini juga menggunakan istilah-istilah kunci selain Tabel referensi imbuhan pemenggalan. Ketika prosedur pemenggalan kata selesai, hasil kata dasar digunakan sebagai panduan dari data kata didalam data base pada istilah-istilah penting ini. Pendekatan ini membutuhkan frasa bertangkai untuk berada dalam kamus kata-kata yang diperlukan; jika tidak, istilah yang dimasukkan dianggap sebagai bentuk dasar [6]. Strategi ketiga algoritma sastrawi adalah algoritma stemming yang merukan sebuah library yang dapat digunakan secara langsung dimana algoritma ini didasarkan pada algoritma Nazief dan Adriani yang kemudian mengalami perbaikan algoritma serta dikenal dengan algoritma confix stripping(CS) pada perkembangannya ditingkatkan lagi menjadi algoritma Enhanced connfix Stripping) dan terakhir mengalami peningkatan oleh modified Enhanced confix stripping.

Beberapa hal yang dapat di selesaikan dengan algoritma stersebut adalah sebagai berikut:

a) Mencegah pemotongan kata yang terlalu over dengan kamus kata dasar.
b) Mencegah under stemming dengan aturan-aturan tambahan.
c) Kata dalam bentuk jamak yang distem: makan-makan menjadi kata makan.

2.4. Algoritma stemming Nazief dan Adriani

Algoritma ini pertama kali diperkenalkan oleh bobby nazief dan andriani, dimana algoritma ini di dasarkan pada aturan morfologi Bahasa Indonesia menjadi satu kelompok. Algoritma ini disusun embali dari proses hasil stemming berdasarkan kata dasar pada setiap kata[7]. Secara garis besar Langkah-langkahnya adalah sebagai berikut :

a) Menentukan kata dasar yang akan mengalami stemming kemudian melakukan pencarian kata tersebut dan berhenti ketika kata tersebut sudah ditemukan
b) Langkah berikutnya adalah menghilangkan kata yang termasuk Inflectional suffixes, yaitu dengan menghilangkan bagian tertentu contoh (simpanlah menjadi simpan dengan menghilangkan bagian kata “-lah”, siapakah menjadi kata siapa dengan menghilangkan bagian kata ”-kah”, atau kata kapanpun menjadi kapan dengan menghilangkan bagian kata “-pun”), Langkah selanjutnya menghilangkan bagian dari pronoun suffixes contohnya (“-mu” ,“-ku”, atau ”-nya”). Kemudian melakukan pengecekan kata dasar didalam kamus, jika sudah ditemukan, maka algortima dapat dihentikan, jika tidak maka dilanjutkan pada langkah ke 3 atau Langkah berikutnya.
c) Menghapus kata yang dapat merubah menjadi noun atau Derivational Suffix yaitu kata yang dapat memunculkan makna baru seperti kata “-an” atau ”-i”. algoritma berhenti bila kata ditemukan dalam kamus kata dasar. bila tidak ditemukan, maka mengikuti Langkah-langkah sebagai berikut 3a: a. Jika akhiran “- an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b. b. Akhiran yang dihapus (“- an” ,“-i”, atau “-kan”) akan dikembalikan, Langkah selanjutnya adalah Langkah 4.
d) Menghapus Derivational Prefix (”di-”,”ke-”,“be-”,”me-”,”pe-“,”se-” atau “te-“). Bila kata ditemukan dalam kamus kata dasar , maka Langkah berhenti, jika tidak, maka lakukan perekaman. Tahapan ini dapat dikerjakan jika memenuhi kondisi sebagai berikut:
    1. Terdapat kombinasi awalan dan akhiran yang tidak diijinkan .
    2. Awalan yang dideteksi sama dengan awalan yang dihilangkan sebelumnya.
    3. Tiga awalan telah dihilangkan .

e) apabila Langkah-langkah tersebut sudah dikerjakan dan kata tersebut tidak terdapat didalam database maka kata tersebut dikembalikan ke bentuk asalnya [5].

2.5. Algoritma TF-IDF (Term Frquency-Invers Document Frequency)

Metode TF-IDF digunakan untuk menghitung bobot kata yang muncul terhadap setiap corpus dokumen. Dalam penelitian ini setiap satu dokumen dianggap mewakili setiap kalimat didalam dokumen yang diujikan. Dengan perhitungan TF-IDF melalui persamaan 1 berikut ini :

$$Wij = tf * idf \quad (1)$$

Pada persamaan 2 berikut ini adalah rumus perhitungan idf dengan logaritma.

$$Wij = tfij * \log \frac{N}{n} \quad (2)$$

Keterangan :

Wij= bobot kata/term tj terhadap dokumen di.

$tfij$= jumlah kemunculan kata/term tj dalam di.

N= jumlah semua dokumen yang ada dalam basis data.

n = jumlah dokumen yang mengandung kata/term tj (minimal ada satu kata yaitu term tj).

Persamaan diatas dapat dinormalisasi dengan persamaan TF-IDF berikut :

Wij= tfij*log N/n+1 (3)

2.6. Cosinus similarity

Metode Cosine Similarity merupakan metode yang digunakan untuk menghitung similaritas (tingkat kesamaan) antar dua buah objek. Secara umum penghitungan metode ini didasarkan pada vector space similarity measure [8]. Metode cosine similarity ini menghitung similarity antara dua buah objek (misalkan D1 dan D2) yang dinyatakan dalam dua buah vector dengan menggunakan keywords (kata kunci) dari sebuah dokumen sebagai ukuran. rumus cosinus similarity adalah sebagai berikut :

$$\text{CosSim}(d_i, q_i) = \frac{q_i.d_i}{|q_i||d_i|} - \frac{\sum_{j=1}^{t}(q_{ij}.d_{ij})}{\sqrt{\sum_{j=1}^{t}(q_{ij})^2 \sum_{j=1}^{t}(d_{ij})^2}} \quad (4)$$

Keterangan :

qij = bobot istilah j pada dokumen i=idf.

dij = bobot istilah j pada dokumen i=$idfj$.

Untuk mencari peringkat dari sebuah kalimat dapat menggunakan algoritma pagerank dengan cara semua kalimat di peringkat dalam sebuah dokumen. Berdasarkan model graph diatas maka peringkat masing-masing vertex dapat dihitung. Sedangkan rumus ranking kalimat sebagai berikut :

$$r(u_i) = d \sum_{j=1}^{n} r(u_i)\, w_{ij} + (1 - d) \quad (5)$$

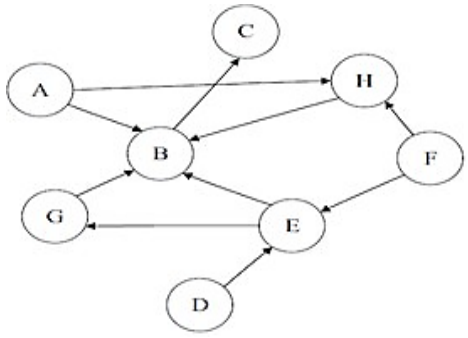
Dimana r(ui) dan u j dua vertex pada graph dan d adalah parameter antara 0 dan 1. Selain menggunakan algoritma faktorisasi matriks seperti Singular Value Decomposition (SVD), Non-Negative Matrix Factorization (NMF), dan Symmetric Matrix Factorization untuk pengumpulan dokumen, pemetaan dokumen dilakukan dalam bentuk matriks. Kalimat yang berbeda dipecah menjadi beberapa kelompok, dan kalimat pembuka akan diambil dari masing-masing kelompok. Dengan mengidentifikasi sekelompok sub-topik laten dari dokumen yang secara miring memberikan informasi tambahan untuk kluster, pendekatan kluster frasa akan mencapai potensi penuhnya. metode ini menyarankan teknik untuk meringkas teks yang tidak membawa sesuatu yang baru[9]. Pendekatan yang disarankan menggunakan data dari efek timbal balik kalimat dan efek konektivitas antar kalimat. Dengan memanfaatkan algoritma peringkat grafik untuk memodelkan dokumen yang ada ke dalam grafik, penelitian dibuat.

2.7. TextRank Dan LexRank Dokumen Tunggal

Peringkasan dokumen secara otomatis adalah proses ektraksi dari dokumen yang mengandung informasi menggunakan software yang berasal dari dokumen aslinya. Peringkasan dokumen otomastis merupakan bagian penting dari pengolahan Bahasa alami (NLP), machine learning dan kecerdasan buatan[10].

Paper ini menjelaskan peringkasan dokumen secara ektraksi dimana hasil ektraksi berupa dokumen tunggal yang berasal dari dokumen aslinya yang terdiri dari kalimat-kalimat representative. Abstrak dokumen yang diringkas harus mengandung kalimat pokok sebagai kata kunci dalam melakukan peringkasan dokumen tunggal[11]

Penggunaan algoritma sorting berdasarkan graph adalah untuk merekomendasikan seberapa penting kalimat tersebut terhadap kalimat yang lain . Bila misalkan ada kalimat yang terdapat pada titik A didalam graph menunjuk ke titik B maka bisa dikatakan kalimat pada titik B lebih penting daripada kalimat yang terdapat di titik A. pentingnya memilih kalimat pada titik A sebagai titik awal menentukan pentingnya kalimat pada titik B berikutnya. Model sorting graf dapat ditujunkkan pada gambar 1 berikut ini :



Gambar 1. Model Sorting Graf.

Penggunaan algoritma taxrank dimaksudkan untuk penggunaan dengan graph yang memilki arah dengan disertai bobot pada masing-masing simpul graph[8]. Dimana rumus dasarnya adalah G={V, E}, V adalah pengelompokan dari node-node yang ada atau vertex yang mewakili setiap kata didalam dokumen, E adalah tepi (edge) yang mewakili relevansi antar kata-kata. Bobot Eijantara simpul Vi dan Vj adalah Wij, dan Wij mewakili kesamaan antara simpul Vi dan Vj yang berasal dari jarak atau cosinus similarity sebuah dokumen. . Bobot setiap kata S(v) dapat dihitung dengan keterhubungan antar simpul S(v) yang lsatu dengan simpul yang lainnya. perhitungan S(V) perihtungan ranking kalimat didapat degan persamaan berikut :

$$S(v) = d + (1-d)\sum u \in B(v) \frac{s(u)}{\sum_{j \in F(u)} w_{uj}} \quad (6)$$

B(v) adalah simpul yang menunjuk ke simpul v, F(v) adalah himpunan simpul yang ditunjuk oleh simpul v, d menunjukkan dumping faktor yang memilki nilai antara 0 sampai 1. Pada persamaan diatas digunakan untuk menghitung nilai note yang ada didalam graph tak berarah. S(v) adalah nilai awal sesuai yang menghitung nilai iterasi didalam graph sampai mencapai nilai konvergen. Nilai dari simpul awal dari graph tidak mempengaruhi nilai akhir dari perhitungan bobot graph tak berarah dan hanya mempengaruhi konvergensi algoritma. Setelah didapatkan hitungan dari nilai konvergensi maka didapatkan stabilitas setiap node dalam graph. Dimana kestabilan tersebut merepresentasikan pusat kata yang sesuai dengan simpul yaitu dapat mengetahui ciri dari dokumen tersebut[12].

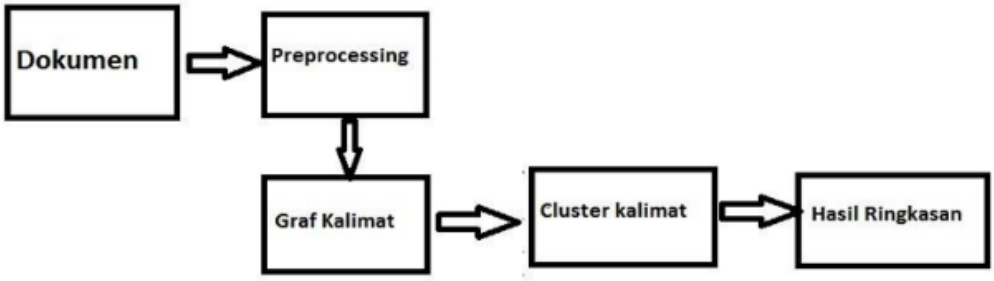
ALgoritma LaxRank membutuhkan perhitungan dari hubungan antara kalimat yang satu dengan kalimat yang lain. Untuk mengukur kesamaan antar kalimat algoritma Laxrank menggunakan frekuensi kemunculan kata di dalam kalimat[13] .Maka

menggunakan rumus graph yang tidak berarah dengan persamaan sebagai berikut : G = (S, E). Di antara keduanya, s S dinyatakan sebagai kalimat, Sisi (si, sj) sedangkan E adalah hubungan antar kalimat didalam dokumen dalam hal ini dokumen tunggal yang akan mengalami peringkasan, sedangkan d dari simpul s adalah link yang terhubung ke s, dimana s merupakan seberapa penting hubungan antar kalimat dalam kalimat yang sesuai. Semakin besar D, maka semakin banyak jumlah kalimat yang terkait dengan kalimat yang sesuai, semakin banyak pula informasi penting yang terkandung dalam kalimat ini, dan dan sebaliknya. jika tingkat simpulnya adalah relatif besar, maka kalimat yang terkait pun juga lebih penting. Dengan cara ini, pertama, paper ini disusun graph G yang tidak berarah dengan menghitung kesamaan antara kalimat. Cara kedua adalah menggunakan iterasi didalam menghitung kestabilan kalimat yang satu terhadap kalimat yang lainnya caraini dilakukan untuk mendapatkan urutan peringkat dari masing-masing kalimat, iterasi akan berhenti Ketika sudah mencapai kestabilan dalam menentukan peringkat kalimat[1].

2.8. Cluster kalimat dengan Node-Node Graph

Metode ini diperkenalkan oleh Anyman El-Kilany, Iman Saleh, Journal IEEE 2012. Dengan bantuan kalimat awal dokumen, ringkasan ekstraktif dibangun. Elemen dokumen, seperti frekuensi kata atau frasa, memengaruhi kalimat-kalimat penting ini. Kalimat yang harus diambil dan lokasinya di dalam teks keduanya dapat ditentukan oleh kata kunci ini[9]. Metode baru untuk mengekstraksi Ringkasan dokumen tunggal disarankan dalam makalah ini. Selain Algoritma Louvain cluster, pendekatan kami juga menggunakan kalimat Grafik dependensi untuk mengekstrak kata kunci dari dokumen. Untuk menguji kinerja metode, kami menggunakan tiga data uji yaitu corpus of British Colombia(BC3), Document Understanding Conference corpus(DUC)1 dan the Concisus corpus of event summaries[9].

Pada Metode ini menjelaskan tentang alur yang dilaksanakan dalam proses pengolahan dokumen sampai menjadi ringkasan pada dokumen tersebut. Berikut adalah gambar pengolahan kalimat pada masing-masing dokumen yang ditunjukkan pada gambar 2 berikut ini :

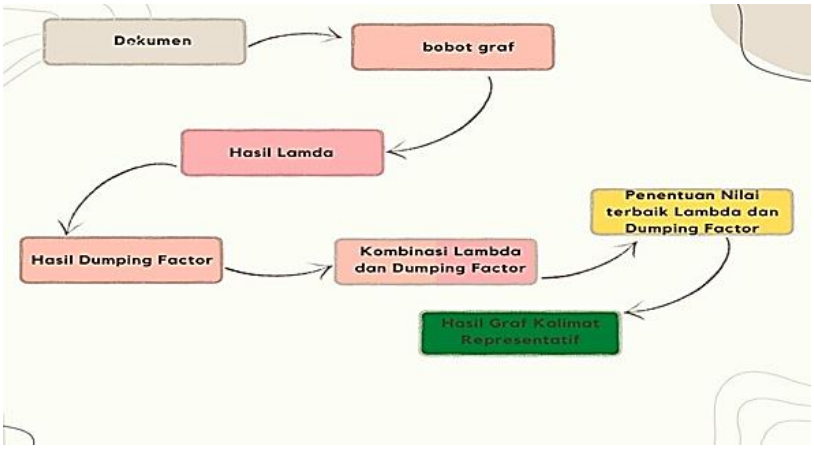


Gambar 2 Pengolahan kalimat pada dokumen

Pada tahapan ini menjelaskan tentang proses yang dilalui oleh dokumen sehingga membentuk sebuah ringkasah dokumen. Pertama dokumen dimasukkan ke sistem peringkasan kemudian diolah melalui proses preprocessing. Tahap berikutnya dari hasil pengolahan preprocessing maka kalimat yang sudah terbentuk di masukkan pada masing-masing node-node graf dimana antar node graf terdapat masing-masing link antar kalimat yang satu dengan kalimat yang lainnya. Pada masing-masing link kalimat memiliki bobot tertentu sesuai seberapa penting hubungan kalimat tersebut dengan kalimat yang lainnya. Pada tahapan berikutnya adalah melakukan cluster pada masing-masing kalimat yang merupakan kalimat representatif dari sebuah dokumen untuk menghasilkan ringkasan dokumen. Algoritma cluster yang digunakan untuk menghasikan ringkasan adalah

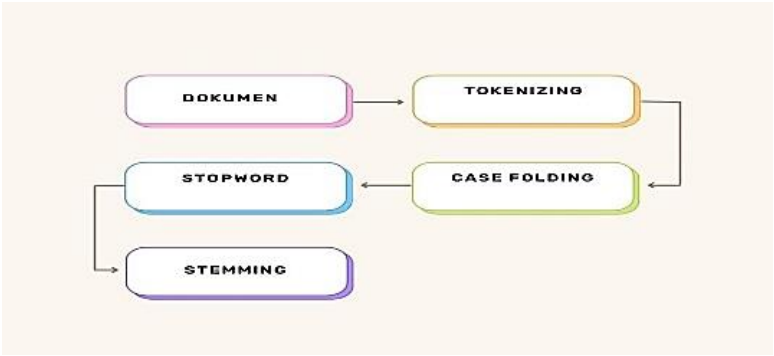
menggunakan SNMF (sparse non negative matrix factorization). Tahapan yang terakhir maka sistem peringkasan akan memunculkan hasil ringkasan dokumen tersebut[1].

Berikut adalah proses pencarian link atau bobot graf terbaik pada masing-masing kalimat untuk menentukan kombinasi terbaik antara dumping factor dengan dengan lambda sehingga didapatkan kalimat representatif dari sebuah dokumen uji yang ditunjukkan pada gambar 3 berikut ini .



Gambar 3. proses pencarian atau penentuan link graf terbaik

Pada tahapan ini dijelaskan bahwa dokumen ditentukan terlebih dahulu bobot graf masing-masing kalimat pada dokumen tersebut kemudian mencari masing-masing nilai lambda disetiap kalimat yang ada, setelah diketahui masing-masing nilai lambda tersebut maka dihitung berdasarkan nilai kombinasi lambda yang telah ditentukan yaitu 0,1 ,03 dan 0,4. Pada proses berikutnya tentukan nilai dumping factor dari masing-masing kalimat berdasarkan bobot kalimat pada graf, setelah proses tersebut selesai maka keduanya di kombinasikan antara lambda dan dumping factor kemudian tentukan nilai tertinggi yang dihasilkan oleh kombinasi tersebut. Bila nilai bobot pada pada masing-masing kalimat kurang dari kombinasi antara nilai dumping factor dan lambda maka kalimat tersebut pada link graf dipotong atau dibuang dan tidak dianggap sebagai kalimat representatif dari ringkasan dokumen sedangkan kalimat pada link graf yang memilki bobot diatas kombinasi antar dumping factor dan lambda maka kalimat tersebut merupakan kalimat represtatif dari dokumen tersebut. Tahapan preprocessing ditunjukkan pada gambar 4 berikut ini :



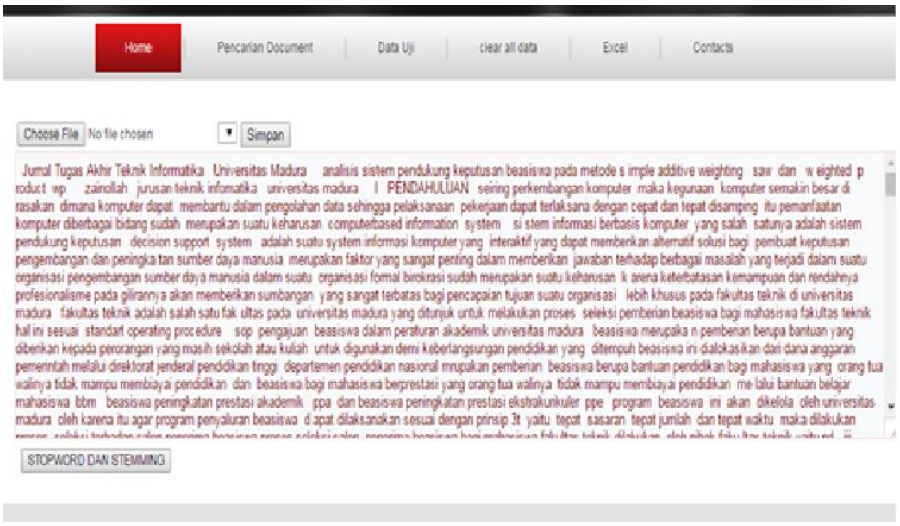
Gambar 4. Tahapan Preprocessing dokumen

Sebelum dokumen diolah maka ada tahapan preprocessing yang harus dilakukan, dokumen dimasukkan kedalam sistem untuk diolah, pertama membaca keseluruhan isi dari dokumen yang dikenal dengan tahap tokenizing, pada tahap berikutnya menjadikan semua kata didalam dokumen menjadi lower case atau mengubah ke huruf kecil semua

kemudian tahap berikutnya cek kata yang ada didalam stoplist atau yang lebih dikenal dengan stopword removal yaitu jika kata tersebut ada didalam stopword removal maka kata tersebut tidak dimasukkan ke tahapan berikutnya, pada tahapan terakhir adalah melakukan stemming yaitu mengambil kata dasar dari seluruh kata yang ada didalam dokumen.

3. Hasil dan Pembahasan

Berikut adalah gambar salah satu proses preprocessing dokumen yang diolah. Pada tahapan ini menampilkan hasil dokumen yang telah diproses sampai tahapan tokenisasi dokumen dimana isi dokumen dibaca terlebih dahulu serta dilakukan proses lowercase dengan tujuan untuk diambil masing-masing kata dasarnya dari dokumen tersebut.



Gambar 5. proses tokenisasi dokumen.

Pada hasil uji parameter dengan menentukan nilai lambda dan dumping factor yang telah dikombinasikan berdasarkan percobaan yang dilakukan oleh 3 orang pakar Bahasa Indonesia dan selanjutnya dijadikan sebagai data latih didapatkan nilai uji pada setiap kalimat didalam single dokumen dengan Tabel 1 sebagai berikut ini
:

Tabel 1. uji parameter oleh pakar

Parameter									
Dok ke	(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)
dok1	0,61	0,57	0,66	0,49	0,63	0,56	0,51	0,62	0,62
	0,59	0,48	0,61	0,68	0,58	0,57	0,55	0,65	0,66
	0,54	0,47	0,58	0,45	0,57	0,48	0,47	0,57	0,51
dok2	0,69	0,62	0,7	0,55	0,67	0,72	0,62	0,69	0,62
	0,6	0,49	0,65	0,6	0,68	0,71	0,61	0,64	0,57
	0,63	0,54	0,71	0,56	0,69	0,77	0,63	0,69	0,63
dok3	0,52	0,57	0,52	0,58	0,64	0,63	0,68	0,63	0,63
	0,53	0,56	0,55	0,65	0,66	0,64	0,68	0,63	0,62
	0,43	0,47	0,47	0,49	0,56	0,51	0,57	0,53	0,53
dok4	0,35	0,45	0,37	0,57	0,46	0,45	0,55	0,55	0,53
	0,36	0,5	0,37	0,59	0,47	0,48	0,6	0,61	0,58
	0,36	0,47	0,36	0,61	0,46	0,45	0,54	0,59	0,56
dok5	0,4	0,47	0,42	0,54	0,55	0,52	0,5	0,5	0,48
	0,51	0,49	0,48	0,72	0,68	0,7	0,69	0,64	0,62
	0,57	0,5	0,56	0,82	0,7	0,77	0,72	0,76	0,68
dok6	0,54	0,58	0,56	0,69	0,75	0,73	0,68	0,71	0,68
	0,57	0,58	0,57	0,69	0,72	0,72	0,67	0,71	0,67
	0,58	0,61	0,58	0,7	0,77	0,77	0,72	0,72	0,65
dok7	0,4	0,5	0,47	0,58	0,53	0,58	0,54	0,56	0,53
	0,45	0,5	0,5	0,65	0,61	0,7	0,66	0,65	0,57

	0,46	0,52	0,5	0,65	0,67	0,61	0,61	0,59	0,56
dok8	0,47	0,36	0,41	0,4	0,42	0,39	0,42	0,47	0,42
	0,44	0,42	0,38	0,48	0,53	0,42	0,52	0,57	0,51
	0,45	0,42	0,41	0,51	0,53	0,4	0,53	0,54	0,49
dok9	0,41	0,41	0,36	0,45	0,47	0,39	0,48	0,48	0,44
	0,47	0,41	0,4	0,48	0,52	0,53	0,51	0,49	0,51
	0,47	0,42	0,38	0,51	0,59	0,58	0,6	0,55	0,54
Dok10	0,53	0,55	0,51	0,51	0,58	0,5	0,58	0,61	0,58
	0,657	0,64	0,58	0,55	0,72	0,6	0,6	0,71	0,66
	0,66	0,64	0,57	0,54	0,72	0,6	0,6	0,7	0,67

Selanjutnya adalah menentukan hasil dari uji dokumen dengan menggunakan ketentuan parameter yang sama yang ditunjjukan di Tabel 2 sebagai data testing dari sistem didapatkan hasil uji para meter terbaik setelah di rata-rata berada pada nilai lambda 0,3 dan nilai dari dumping factor 0,8 dengan mengambil nilai terbaik dari recall adalah 0,64421 dapat dilihat dari hasil uji coba Tabel 2 berikut ini :

Tabel 2. Hasi Ujicoba 50 Dokumen maksimum nilai dari pakar tata Bahasa Indonesia.

	Parameter								
No	**(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	0,61	0,57	0,66	0,68	0,63	0,57	0,55	0,65	0,66
2	0,69	0,62	0,71	0,6	0,69	0,77	0,63	0,69	0,63
3	0,53	0,57	0,55	0,65	0,66	0,64	0,68	0,63	0,63
4	0,36	0,5	0,37	0,61	0,47	0,48	0,6	0,613	0,58
5	0,57	0,505	0,56	0,82	0,7	0,77	0,72	0,76	0,68
6	0,58	0,61	0,58	0,7	0,77	0,77	0,72	0,72	0,68
7	0,46	0,52	0,5	0,65	0,67	0,7	0,66	0,65	0,57
8	0,47	0,42	0,41	0,51	0,53	0,42	0,53	0,57	0,51
9	0,47	0,42	0,4	0,51	0,59	0,58	0,6	0,55	0,54
10	0,66	0,64	0,58	0,55	0,72	0,6	0,6	0,71	0,67
11	0,59	0,6	0,57	0,49	0,7	0,61	0,66	0,64	0,55
12	0,75	0,71	0,69	0,64	0,63	0,59	0,61	0,75	0,7
13	0,54	0,64	0,59	0,61	0,54	0,54	0,62	0,59	0,5
14	0,39	0,51	0,45	0,52	0,53	0,52	0,53	0,66	0,59
Rata-Rata	0,57	0,57	0,56	0,63	0,61	0,64	0,63	0,63	0,61

Tahapan selanjutnya adalah menentukan nilai dari recall, precision dan F-score sebagai pengukur hasil dari ringkasan dokumen tunggal dengan menggunakan Rouge 2.0 yang ditunjukkan oleh Tabel 3, hasil yang diperoleh terjadi peningkatan nilai jika dibandingkan dengan penelitian sebelumnya yang menggunakan nilai lambda dan dumping factor secara acak, dengan menggunakan nilai kombinasi dari nilai lambda (0,1, 0.3 dan 0,5) serta untuk nilai dari dumping factor adalah (0,3, 0,5 dan 0,8) maka didapatkan hasil seperti pada Tabel 3 berikut :

Tabel 3. Penentuan recall, precission dan F-score

Dok ke	**Recall (0.1), (0.85)**	**Precision (0.1), (0.85)**	**F-Score (0.1), (0.85)**	**Recall (0.3), (0.85)**	**Precision (0.3), (0.85)**	**F-Score (0.3), (0.85)**	**Recall (0.5), (0.85)**	**Precision (0.5), (0.85)**	**F-Score (0.5), (0.85)**
1	0,65	0,18	0,24	0,59	0,17	0,22	0,6	0,17	0,22
	0,64	0,16	0,21	0,57	0,15	0,2	0,6	0,16	0,2
	0,54	0,22	0,29	0,49	0,2	0,26	0,52	0,21	0,28
2	0,65	0,15	0,19	0,6	0,14	0,18	0,62	0,16	0,21
	0,58	0,14	0,18	0,64	0,15	0,19	0,62	0,16	0,22
	0,64	0,15	0,2	0,61	0,15	0,2	0,66	0,17	0,23
3	0,55	0,17	0,23	0,56	0,18	0,24	0,69	0,19	0,26
	0,58	0,19	0,25	0,6	0,19	0,26	0,67	0,2	0,27
	0,49	0,18	0,24	0,49	0,19	0,25	0,59	0,2	0,27
4	0,49	0,15	0,19	0,51	0,16	0,21	0,62	0,16	0,22
	0,53	0,15	0,19	0,49	0,15	0,19	0,5	0,14	0,18
	0,49	0,15	0,19	0,5	0,15	0,2	0,54	0,16	0,21
5	0,48	0,16	0,21	0,53	0,17	0,23	0,59	0,2	0,27
	0,55	0,16	0,2	0,62	0,17	0,22	0,74	0,21	0,29
	0,55	0,14	0,18	0,69	0,16	0,21	0,69	0,18	0,24
6	0,4	0,14	0,18	0,4	0,16	0,2	0,41	0,14	0,17

Dok ke	Recall (0.1), (0.85)	Precision (0.1), (0.85)	F-Score (0.1), (0.85)	Recall (0.3), (0.85)	Precision (0.3), (0.85)	F-Score (0.3), (0.85)	Recall (0.5), (0.85)	Precision (0.5), (0.85)	F-Score (0.5), (0.85)
	0,47	0,14	0,18	0,49	0,16	0,2	0,48	0,14	0,17
	0,48	0,13	0,16	0,51	0,14	0,18	0,52	0,13	0,16
7	0,5	0,15	0,2	0,6	0,19	0,25	0,55	0,2	0,26
	0,51	0,14	0,18	0,64	0,17	0,23	0,61	0,18	0,24
	0,52	0,14	0,17	0,69	0,16	0,22	0,66	0,17	0,23
8	0,43	0,15	0,2	0,47	0,13	0,17	0,46	0,15	0,19
	0,4	0,13	0,16	0,49	0,12	0,15	0,51	0,13	0,16
	0,41	0,13	0,16	0,44	0,12	0,14	0,52	0,13	0,16
9	0,36	0,13	0,16	0,43	0,16	0,21	0,39	0,15	0,18
	0,38	0,13	0,16	0,51	0,19	0,25	0,45	0,16	0,21
	0,34	0,14	0,17	0,58	0,21	0,29	0,45	0,17	0,22
10	0,54	0,15	0,19	0,62	0,17	0,22	0,65	0,15	0,19
	0,54	0,14	0,18	0,71	0,17	0,22	0,71	0,15	0,19
	0,54	0,14	0,17	0,69	0,16	0,21	0,72	0,14	0,18

4. Kesimpulan

Dengan hasil uji coba yang sudah dilakukan untuk penentuan parameter uji terbaik dengan menggunakan kombinasi lambda dan dumping factor seperti yang tertera pada Tabel pengujian ringkasan. Dengan parameter tersebut didapatkan hasil ringkasan sesuai perameter uji terbaik yaitu berada pada kombinasi nilai lambda 0.3 dan damping factor 0.8 maka nilai threshold yang digunakan adalah 0.6444 dalam meringkas dokumen, apabila nilai atau bobot kurang dari nilai recall dan precission serta f-score maka kalimat tersebut tidak dimasukkan kedalam ringkasan dokumen tunggal. Pada penelitian ini disarankan untuk mengujicobakan parameter-parameter baru untuk dimasukkan dalam ujicoba dengan tujuan mencari atau menentukan link graf terbaik pada kalimat-kalimat didalam dokumen yang lebih menggambarkan kalimat representatif dari sebuah dokumen sebagai nilai threshold.

Daftar Pustaka

[1] C. Kwatra and K. Gupta, “Extractive and Abstractive Summarization for Hindi Text using Hierarchical Clustering,” *Proc. 2021 IEEE Int. Conf. Innov. Comput. Intell. Commun. Smart Electr. Syst. ICSES 2021*, 2021, doi: 10.1109/ICSES52305.2021.9633789.

[2] J. Asian, “Effective Techniques for Indonesian Text Retrieval,” *Ph.D Thesis*, pp. 1–286, 2007, [Online]. Available: https://researchrepository.rmit.edu.au/esploro/outputs/doctoral/Effective-techniques-for-Indonesian-text-retrieval/9921861570701341

[3] S. Singh and A. Mahmood, “The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures,” *IEEE Access*, vol. 9, pp. 68675–68702, 2021, doi: 10.1109/ACCESS.2021.3077350.

[4] J. Asian, H. E. Williams, and S. M. M. Tahaghoghi, “Stemming Indonesian,” *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 38, no. September 2018, pp. 307–314, 2005, doi: 10.1145/1316457.1316459.

[5] I. Darmawan and N. Haidar Hari, “Discourse Connectors Pada Peringkasan Dokumen Berbahasa Indonesia,” *SMARTICS J.*, vol. 7, no. 1, pp. 33–41, 2021, [Online]. Available: https://doi.org/10.21067/smartics.v7i1.5046

[6] I. Darmawan, R. A. H, and H. Armato, “InfoTekJar : Jurnal Nasional Peringkasan paper dengan metode Sparse Nonnegative Matrix Factorization untuk Pemeriksaan Kesesuaian dengan Abstrak Tugas Akhir,” vol. 1, 2020.

[7] N. Ramadhani, “View of Penerapan Algoritma Naïve Bayes Classifier dan Fungsi Gaussian Untuk Penentuan Penjurusan Siswa Kelas X.pdf,” vol. 8, no. 1, pp. 14–21, 2022.

[8] M. Asfi and N. Fitrianingsih, “Implementasi Algoritma Naive Bayes Classifier sebagai Sistem Rekomendasi Pembimbing Skripsi,” *InfoTekJar J. Nas. Inform.*

dan Teknol. Jar., vol. 5, no. 1, p. 44, 2020.

[9] A. El-Kilany and I. Saleh, "Unsupervised document summarization using clusters of dependency graph nodes," *Int. Conf. Intell. Syst. Des. Appl. ISDA*, pp. 557–561, 2012, doi: 10.1109/ISDA.2012.6416598.

[10] H. Zheng and M. Lapata, "Sentence centrality revisited for unsupervised summarization," *ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, no. 2, pp. 6236–6247, 2020, doi: 10.18653/v1/p19-1628.

[11] A. Alifimoff, "Abstractive Sentence Summarization with," pp. 1–9.

[12] Y. C. Tseng, M. H. Yang, Y. C. Fan, W. C. Peng, and C. C. Hung, "Template-Based Headline Generator for Multiple Documents," *IEEE Access*, vol. 10, pp. 46330–46341, 2022, doi: 10.1109/ACCESS.2022.3157287.

[13] A. Li, T. Jiang, Q. Wang, and H. Yu, "The mixture of TextRank and LexRank techniques of single document automatic summarization research in Tibetan," *Proc. - 2016 8th Int. Conf. Intell. Human-Machine Syst. Cybern. IHMSC 2016*, vol. 1, pp. 514–519, 2016, doi: 10.1109/IHMSC.2016.278.