TEXT SUMMARIZATION PADA ARTIKEL BERITA MENGGUNAKAN *VECTOR SPACE MODEL* DAN *COSINE SIMILARITY*

Mardi Siswo Utomo, Jati Sasongko Wibowo*, Eko Nur Wahyudi
Fakultas Teknologi Informasi dan Industri, Universitas Stikubank
Email *jatisw@edu.unisbank.ac.id

ABSTRAK

Sebuah artikel yang panjang akan membuat pembaca membutuhkan waktu yang lebih lama untuk dapat menyelesaikan bacaan dan pemahamannya. Sehingga dibutuhkan sebuah bentuk ringkasan untuk mempercepat pembaca dalam memahami secara singkat isi dari artikel secara keseluruhan. Umumnya ringkasan dibuat oleh penulis dalam bentuk tulisan manual untuk menggambarkan isi keseluruhan artikel. Sehingga dibutuhkan untuk membuat ringkasan secara otomatis dengan tidak merubah isi substansi dari artikel. Meringkas secara otomatis menggunakan metode *vector space model* dan *cosine similarity.* VSM digunakan untuk memberikan bobot nilai pada semua kata yang ada di artikel. CS digunakan untuk menghitung kemiripan antara judul artikel dengan isi artikel.

Selain kedua algoritma yang telah disebutkan, masih ada beberapa proses atau metode yang dilakukan khususnya pada tahap pre-processing. Diantaranya crawling, tokenization, punctuation removal, stopword, dan stemming. Hasil dari pre-processing ini baru kemudian dilakukan proses menggunakan algoritma vector space model dan cosine similarity, dan terakhir diurutkan berdasarkan nilai cosine similarity tertinggi.

Hasil dari proses peringkasan berupa sebuah paragraf yang diambil dari beberapa kalimat yang mempunyai nilai kemiripan dengan judul paling tinggi. Pada penelitian ini dari 104 kalimat yang ada pada artikel di dapat 5 kalimat yang mempunyai nilai kemiripan paling tinggi. Lima kalimat ini dijadikan satu paragraf sebagai hasil dari proses peringkasan artikel.

Kata kunci: summarization, tf-idf, cosine similarity

1. PENDAHALUAN

Artikel yang mempunyai jumlah halaman yang banyak akan membuat pembaca memiliki rasa yang berat apabila harus membaca keseluruhan dokumen. Melalui proses peringkasan akan menghasilkan bentuk penyajian baru informasi dari sebuah dokumen yang lebih ringkas. Ringkasan tetap merepresentasikan dari keseluruhan dokumen dan tetap menjaga nilai arti substansi dari dokumen. Pengujian terhadap hasil peringkasan bergantung pada struktur dan jenis dokumen. Jenis dokumen ilmiah argmentasi akan menghasilkan peringkasan yang lebih baik dibandingkan dengan dokumen yang bukan ilmiah. Pada struktur dokumen yang mempunyai jumlah paragraf yang banyak dan pada setiap paragraf mempunyai kalimat yang lebih dari dua akan menghasilkan peringkasan yang lebih baik.

Peringkasan yang dihasilkan dari dokumen teks, panjang hasil ringkasan tidak lebih dari setengah dari dokumen asli, dan masih mempunyai arti yang signifikan terhadap isi keseluruhan dokumen (Hovy, 2001). Automatic text summarization atau peringkasan teks otomatis di awali oleh Luhn sejak tahun 1958 dengan menggunakan bermacam metode. Teknik yang diterapkan dalam peringkasan cukup banyak, diantaranya position in text (Baxendale, 1958), teknik pendekatan statistika yaitu teknik word frequency (Luhn, 1958), cue words and heading (Edmudson, 1969), sentence position (Lin dan Hoovy, 1997). Teknik dengan

pendekatan natural language analysis yaitu lexical chain (Mc Keown, 1997), maximal maginal relevance (Cabonell dan Goldstein, 1998). inverse term frequency and NLP technique (Aone, 1990).

2. TINJAUAN PUSTAKA

2.1. Peringkasan Teks Otomatis Pada Dokumen Berbahasa Jawa Menggunakan Metode Tf-Idf

Ringkasan yang dibuat secara manual memerlukan waktu yang lebih lama. Sehingga untuk menyelesaikan masalah dalam memerlukan waktu yang lama dalam membaca dibutuhkan system peringkasan dokumen teks secara otomatis. Sebuah dokumen dari Bahasa Jawa diringkas dari seluruh isi dokumen untuk mendapatkan ringkasan secara otomatis, sehingga tanpa membaca keseluruhan isi dokumen pembaca akan dapat membaca dan memahami artikel lebih mudah dan cepat.

Penelitian untuk peringkasan teks secara otomatis dengan menerapkan metode tf-idf dengan memberikan nilai pembobotan pada setiap kata dari dokumen teks. Sebuah kata yang mempunyai bobot nilai tf-idf yang besar berarti memiliki jumlah kata terbanyak dari dokumen teks. Pemberian nilai bobot pada setiap kata yang diambil dari keseluruhan isi dokumen teks akan menghasilkan kata yang mempunyai nilai bobot paling besar hingga terkecil. Kata yang mempunyai nilai bobot paling besar merupakan bagian penting dari dokumen. Pengujian dilakukan dengan melakukan koreksi secara manual dan dilakukan oleh system secara otomatis sehingga menghasilkan ringkasan yang relevan maupun tidak relevan.

Pengujian yang dilakukan secara manual menghasilkan akurasi dengan nilai rata-rata sebesar 64%. Sistem peringkasan secara otomatis akan membantu pembaca dalam mendapatkan intisari dari dokumen teks dengan akurasi yang cukup baik, sehingga pembaca dapat mempunyai pertimbangan untuk melanjutkan membaca dari keseluruhan dokumen atau tidak [1].

2.2. Peringkasan Proposal Skripsi Menggunakan Algoritma Vector Space Model

Jumlah peningkatan dokumen teks digital cukup besar seiring dengan peningkatan teknologi informasi dalam bidang publikasi. Sebuah artikel yang mempunyai jumlah halaman yang banyak akan membuat pembaca akan merasa kesulitan apabila harus membaca keseluruhan isi artikel dan memahami informasi yang terdapat dalam artikel tersebut. Peringkasan dokumen merupakan salah satu sub bagian dari Data Mining yang digunakan untuk menyederhanakan uraian artikel yang panjang menjadi lebih pendek dengan tidak menghilangkan substansi yang ada di dalam artikel. Membaca artikel secara keseluruhan akan membutuhkan waktu yang lebih lama, apalagi yang mempunyai kesibukan lain, sehingga dibutuhkan cara yang cepat untuk memahami isi artikel. Peringkasan otomatis akan melakuan ekstraksi dari dokumen teks yang disusun kembali dari beberapa kalimat menjadi intisari artikel dalam bentuk ringkasan. Proses peringkasan dengan menerapkan algoritma vector space model untuk memberikan nilai pembobotan pada tiap kalimat. Peringkasan dokumen dengan menghapus kalimat yang memiliki bobot nilai yang lebih kecil dibandingkan dengan kalimat lainnnya. Hasil ringkasan dokumen dalam bentuk kalimat berdasarkan nilai vector space model sejumlah 31% terhadap seluruh kalimat yang berjumlah 69% [2].

2.3. Otomatisasi Peringkasan Dokumen Sebagai Pendukung Sistem Manajemen Surat

Peringkasan merupakan proses untuk menyajian informasi dari dokumen teks yang lebih ringkas tanpa mengurangi arti dari substansi isi dokumen teks secara keseluruhan. Penelitian ini menerapkan metode Naïve Bayes yang menghasilkan penyajian kembali dalam bentuk lebih ringkas dari sebuah dokumen teks. Dokumen yang digunakan dalam penelitian ini berupa surat

dalam bentuk dokumen teks. Perhitungan probabilitas dalam proses peringkasan dokumen berdasarkan pada fitur teks yang terdapat dalam isi surat. Fitur isi surat diantaranya kata kunci, frase kunci, frekuensi kata, dan kata yang terkait dengan kelas entitas atau numerik. Tingkat kompresi pada hasil uji coba mempunyai nilai 53.67% dari dokumen teks mencapai ringkasan dengan nilai 96.67% yang mencerminkan informasi utama yang tersedia [3].

2.4. Text Summarization untuk Dokumen Berita Berbahasa Indonesia

Peringkasan sebuah dokumen untuk memproses informasi yang mempunyai substansi penting dalam menghasilkan bentuk dokumen yang lebih singkat. Pada peringkasan mempunyai dua metode, yaitu ekstraksi dan abstraksi. Peringkasan dengan ekstraksi pada bagian dari dokumen teks sumber di pilih beberapa kata, dan selanjutnya kata yang telah dipilih digabungkan menjadi kalimat yang lebih pendek. Peringkasan menggunakan metode abstraksi membuat hasil dari peringkasan menjadi lebih mudah dibaca dengan menggabungkan bagian pada dokumen yang dipilih. Penelitian peringkasan ini dilakukan dengan menerapkan metode ekstraksi. Proses stemming untuk membuat kata berimbuhan menjadi kata dasar diterapkan dengan menggunakan algoritma stemming telah dimodifikasi dibuat oleh Nazief dan Adriani. Beberapa hasil penelitian faktor-faktor yang mempengaruhi kinerja dari sistem peringaksan otomatis, seperti: struktur data, masukan teks, kamus, stemming, dan keyword [4].

3. LANDASAN TEORI

3.1. Text Summarization

Peringkasan teks merupakan sebuah proses untuk meringkas dokumen teks dengan tidak mengurangi arti penting dari isi dokumen teks secara keseluruhan. Meringkas dokumen teks secara umum dibatasi dengan panjang kalimat kurang lebih dua ratus kata dari keseluruhan dokumen teks yang tersedia.

Peringkasan data otomatis adalah bagian dari pembelajaran mesin dan penggalian data. Gagasan utama peringkasan adalah untuk menemukan subset data yang berisi "informasi" dari seluruh set. Teknik seperti ini banyak digunakan dalam industri saat ini. Mesin pencari adalah contoh; yang lain termasuk peringkasan dokumen, koleksi gambar, dan video. Peringkasan dokumen mencoba membuat ringkasan representatif atau abstrak dari seluruh dokumen, dengan menemukan kalimat yang paling informatif, sedangkan dalam peringkasan gambar sistem menemukan gambar yang paling representatif dan penting (yaitu yang menonjol). Untuk video pengawasan, satu mungkin ingin mengekstrak peristiwa penting dari konteks yang lancar. [5]

Proses peringkasan otomatis mempunyai dua pendekatan umum: ekstraksi dan abstraksi. Metode ekstraktif bekerja dengan memilih subset dari kata, frasa, atau kalimat yang ada dalam dokumen teks hingga menjadi sebuah ringkasan. Sedangkan metode abstraktif membuat sebuah representasi semantik internal dan selanjutnya dengan menggunakan teknik generasi bahasa alami membuat ringkasan yang dapat lebih mudah dibaca dan dipahami dengan apa yang mungkin diungkapkan manusia. Ringkasan seperti itu mungkin termasuk inovasi verbal. Penelitian sampai saat ini telah berfokus terutama pada metode ekstraktif, yang sesuai untuk peringkasan koleksi gambar dan peringkasan video.

3.2. Term Frequency – Invers Document Frequency

Dalam pencarian informasi, tf-idf atau TFIDF, kependekan dari frekuensi dokumen frekuensi-terbalik, merupakan sebuah statistik numerik yang digunakan untuk merepresentasikan pentingnya sebuah kata dalam sebuah dokumen teks. Pada pencarian data atau informasi, penambangan teks, dan pemodelan data sering kali digunakan sebagai faktor

pembobotan. Nilai tf-idf pada sebuah kata dalam dokumen teks akan meningkat secara proporsional seiring dengan bertambahnya jumlah kata tersebut dalam sebuah dokumen teks. Saat ini tf-idf merupakan salah satu model pembobotan kata yang paling popular, salah satunya sebagai sistem rekomendasi di perpustakaan digital berbasis teks menggunakan tf-idf.

Mesin pencari informasi menggunakan variasi skema pembobotan tf-idf dalam memberian penilaian dan peringkat keterkaitan kata kunci yang dimasukkan pengguna dengan data dokumen yang tersedia. Proses perhitungan tf-idf dapat digunakan untuk menfilter dalam pencarian informasi, klasifikasi dokumen, dan peringkasan dokumen. Setiap term dihitung dengan menjumlahkan tf-idf merupakan salah fungsi peringkat paling sederhana [6].

3.3. Dokumen

Pada kamus besar Bahasa Indonesia definisi dari dokumen/do·ku·men/ /dokumén/ n "1 surat yang tertulis atau tercetak yang dapat dipakai sebagai bukti keterangan (seperti akta kelahiran, surat nikah, surat perjanjian); 2 barang cetakan atau naskah karangan yang dikirim melalui pos; 3 rekaman suara, gambar dalam film, dan sebagainya yang dapat dijadikan bukti keterangan" [7].

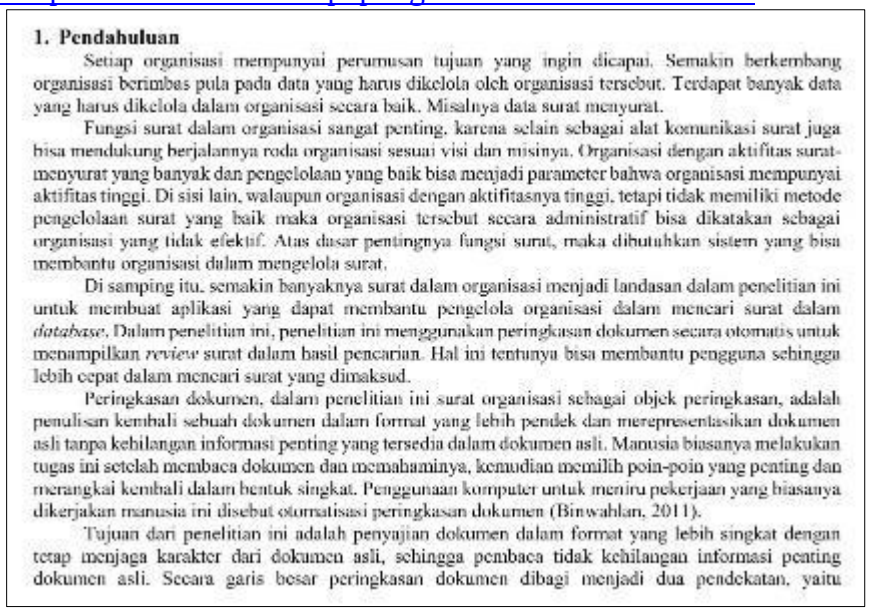
3.3. Teks

Pada kamus besar Bahasa Indonesia definisi teks /téks/ n "1 naskah yang berupa a kata-kata asli dari pengarang; b kutipan dari kitab suci untuk pangkal ajaran atau alasan; c bahan tertulis untuk dasar memberikan pelajaran, berpidato, dan sebagainya: peringatan itu didahului dengan pembacaan -- Pancasila; berulang kali ia melirik ke dalam -- terjemahan sajak-sajak yang sedang dibacanya; 2 wacana tertulis" [8]

4. METODE PENELITIAN

4.1.Memilih Dokumen

Dokumen yang digunakan dalam peringkasan ini diambil dari salah satu artikel publikasi dengan judul "otomatisasi peringkasan dokumen sebagai pendukung sistem manajemen surat" dengan penulis Ahmad Najibullah, Wang Mingyan dari Universitas Nanchang, Republik Rakyat Tiongkok. Artikel diambil dari alamat jurnal http://journal.unipdu.ac.id:8080/index.php/register/article/view/400/353



1. Pendahuluan

Setiap organisasi mempunyai perumusan tujuan yang ingin dicapai. Semakin berkembang organisasi berimbas pula pada data yang harus dikelola oleh organisasi tersebut. Terdapat banyak data yang harus dikelola dalam organisasi secara baik. Misalnya data surat menyurat.

Fungsi surat dalam organisasi sangat penting, karena selain sebagai alat komunikasi surat juga bisa mendukung berjalannya roda organisasi sesuai visi dan misinya. Organisasi dengan aktifitas surat-menyurat yang banyak dan pengelolaan yang baik bisa menjadi parameter bahwa organisasi mempunyai aktifitas tinggi. Di sisi lain, walaupun organisasi dengan aktifitasnya tinggi, tetapi tidak memiliki metode pengelolaan surat yang baik maka organisasi tersebut secara administratif bisa dikatakan sebagai organisasi yang tidak efektif. Atas dasar pentingnya fungsi surat, maka dibutuhkan sistem yang bisa membantu organisasi dalam mengelola surat.

Di samping itu, semakin banyaknya surat dalam organisasi menjadi landasan dalam penelitian ini untuk membuat aplikasi yang dapat membantu pengelola organisasi dalam mencari surat dalam *database*. Dalam penelitian ini, penelitian ini menggunakan peringkasan dokumen secara otomatis untuk menampilkan *review* surat dalam hasil pencarian. Hal ini tentunya bisa membantu pengguna sehingga lebih cepat dalam mencari surat yang dimaksud.

Peringkasan dokumen, dalam penelitian ini surat organisasi sebagai objek peringkasan, adalah penulisan kembali sebuah dokumen dalam format yang lebih pendek dan merepresentasikan dokumen asli tanpa kehilangan informasi penting yang tersedia dalam dokumen asli. Manusia biasanya melakukan tugas ini setelah membaca dokumen dan memahaminya, kemudian memilih poin-poin yang penting dan merangkai kembali dalam bentuk singkat. Penggunaan komputer untuk meniru pekerjaan yang biasanya dikerjakan manusia ini disebut otomatisasi peringkasan dokumen (Binwahlan, 2011).

Tujuan dari penelitian ini adalah penyajian dokumen dalam format yang lebih singkat dengan tetap menjaga karakter dari dokumen asli, sehingga pembaca tidak kehilangan informasi penting dokumen asli. Secara garis besar peringkasan dokumen dibagi menjadi dua pendekatan, yaitu

Gambar 1. Dokumen Artikel

4.2.Memasukkan Dokumen dalam Database

Dokumen yang sudah dalam bentuk digital tersebut kemudian dimasukkan ke dalam database. Dokumen yang dimasukkan berupa teks saja, dan yang dimasukkan pun tidak semuanya, tetapi hanya judul artikel, alamat url (uniform resource locator) dan isi artikel di mulai dari pendahuluan hingga kesimpulan. Seperti pada gambar di bawah ini yang merupakan hasil dari dokumen yang telah dimasukkan ke dalam database.

id	judul	isi	url
1	Otomatisasi Peringkasan Dokumen Seb...	1. Pendahuluan¶ Setiap organisasi mem...	http://journal.unipdu.ac.id:8080/index.php/...

Gambar 2. Dokumen dalam Database

4.3.Dokumen Tampilan Web

Dokumen yang telah masuk ke dalam database dapat ditampilkan dalam bentuk versi web seperti pada gambar di bawah ini, format tampilan web tidak sama dengan format asli dari dokumen asli, tetapi secara isi masih sama. Perbedaan lainnya yaitu yang masuk ke dalam database hanya teks saja, sedangkan semua gambar dan formula tidak dimasukkan ke dalam database, sehingga di tampilan web, tidak akan tampak adanya gambar.

1. Pendahuluan Setiap organisasi mempunyai perumusan tujuan yang ingin dicapai. Semakin berkembang organisasi berimbas pula pada data yang harus dikelola oleh organisasi tersebut. Terdapat banyak data yang harus dikelola dalam organisasi secara baik. Misalnya data surat menyurat. Fungsi surat dalam organisasi sangat penting, karena selain sebagai alat komunikasi surat juga bisa mendukung berjalannya roda organisasi sesuai visi dan misinya. Organisasi dengan aktifitas suratmenyurat yang banyak dan pengelolaan yang baik bisa menjadi parameter bahwa organisasi mempunyai aktifitas tinggi. Di sisi lain, walaupun organisasi dengan aktifitasnya tinggi, tetapi tidak memiliki metode pengelolaan surat yang baik maka organisasi tersebut secara administratif bisa dikatakan sebagai organisasi yang tidak efektif. Atas dasar pentingnya fungsi surat, maka dibutuhkan sistem yang bisa membantu organisasi dalam mengelola surat. Di samping itu, semakin banyaknya surat dalam organisasi menjadi landasan dalam penelitian ini untuk membuat aplikasi yang dapat membantu pengelola organisasi dalam mencari surat dalam database. Dalam penelitian ini, penelitian ini menggunakan peringkasan dokumen secara otomatis untuk menampilkan review surat dalam hasil pencarian. Hal ini tentunya bisa membantu pengguna sehingga lebih cepat dalam mencari surat yang dimaksud. Peringkasan dokumen, dalam penelitian ini surat organisasi sebagai objek peringkasan, adalah penulisan kembali sebuah dokumen dalam format yang lebih pendek dan merepresentasikan dokumen asli tanpa kehilangan informasi penting yang tersedia dalam dokumen asli. Manusia biasanya melakukan tugas ini setelah membaca dokumen dan memahaminya, kemudian memilih poin-poin yang penting dan merangkai kembali dalam bentuk singkat. Penggunaan komputer untuk meniru pekerjaan yang biasanya dikerjakan manusia ini disebut otomatisasi peringkasan dokumen (Binwahlan, 2011). Tujuan dari penelitian ini adalah penyajian dokumen dalam format yang lebih singkat dengan tetap menjaga karakter dari dokumen asli, sehingga pembaca tidak kehilangan informasi penting dokumen asli. Secara garis besar peringkasan dokumen dibagi menjadi dua pendekatan, yaituOtomatisasi Peringkasan

Gambar 3. Tampilan Data Teks Versi Web

4.4.Tokenisasi Kalimat Dokumen Teks

Dokumen sebelum dilakukan tokenisasi diberikan sebuah id penanda dengan nomor id 1. Apabila jumlah dokumen banyak berarti nomor id juga akan banyak. Data dokumen yang telah ada di dalam database dilakukan tokenisasi ke dalam bentuk kalimat. Dalam arti bahwa dokumen dipisahkan dalam bentuk kalimat-kalimat berdasarkan tanda titik. Sehingga dari proses tokenisasi dokumen tersebut diperoleh 104 kalimat. Dari setiap tokenisasi kalimat tersebut diberikan sebuah nomor penanda dari nomor 1 hingga nomor 104. Artinya setiap dokumen mempunyai nomor id, dan 15ystem dokumen tersebut di tokenisasi maka hasil token diberikan nomor sesuai hasil token dan nomor id dokumen. Sehingga id dokumen dan nomor hasil token akan selalu digunakan untuk menghubungkan antara kalimat hasil token dengan dokumen aslinya. Seperti tampak pada gambar di bawah ini.

id	no	judul	kalimat	url
1	1	otomatisasi peringkasan dokumen seba...	fitur rata rata frekuensi kata tf ini dihitung ...	http://journal.unipdu.ac.id:8080/index.php/...
1	2	otomatisasi peringkasan dokumen seba...	sesuai dengan teorema na ve bayes ma...	http://journal.unipdu.ac.id:8080/index.php/...
1	3	otomatisasi peringkasan dokumen seba...	penghitungan probabilitas untuk menent...	http://journal.unipdu.ac.id:8080/index.php/...
1	4	otomatisasi peringkasan dokumen seba...	jika dalam kalimat tersebut terdapat kata...	http://journal.unipdu.ac.id:8080/index.php/...
1	5	otomatisasi peringkasan dokumen seba...	fitur yang terakhir dalam penelitian ini dia...	http://journal.unipdu.ac.id:8080/index.php/...
1	6	otomatisasi peringkasan dokumen seba...	nama entitas dan numerik ini bisa didap...	http://journal.unipdu.ac.id:8080/index.php/...
1	7	otomatisasi peringkasan dokumen seba...	biasanya kalimat yang mengandung na...	http://journal.unipdu.ac.id:8080/index.php/...
1	8	otomatisasi peringkasan dokumen seba...	setiap kata dalam kalimat akan dicari ke...	http://journal.unipdu.ac.id:8080/index.php/...
1	9	otomatisasi peringkasan dokumen seba...	frase kunci ini ditentukan oleh pola yang ...	http://journal.unipdu.ac.id:8080/index.php/...
1	10	otomatisasi peringkasan dokumen seba...	ekstraksi frase kunci juga menentukan ri...	http://journal.unipdu.ac.id:8080/index.php/...
1	11	otomatisasi peringkasan dokumen seba...	semakin banyak kemunculan kata terse...	http://journal.unipdu.ac.id:8080/index.php/...
1	12	otomatisasi peringkasan dokumen seba...	selain fitur yang telah disebutkan di atas ...	http://journal.unipdu.ac.id:8080/index.php/...
1	13	otomatisasi peringkasan dokumen seba...	di dalam dokumen d terdapat kumpulan ...	http://journal.unipdu.ac.id:8080/index.php/...
1	14	otomatisasi peringkasan dokumen seba...	jika kalimat terlalu panjang maka pemba...	http://journal.unipdu.ac.id:8080/index.php/...
1	15	otomatisasi peringkasan dokumen seba...	diasumsikan bahwa kalimat yang terlalu ...	http://journal.unipdu.ac.id:8080/index.php/...
1	16	otomatisasi peringkasan dokumen seba...	panjang suatu kalimat bisa mempengar...	http://journal.unipdu.ac.id:8080/index.php/...
1	17	otomatisasi peringkasan dokumen seba...	fitur teks yang pertama adalah panjang s...	http://journal.unipdu.ac.id:8080/index.php/...
1	18	otomatisasi peringkasan dokumen seba...	terdapat lima fitur yang digunakan dalam...	http://journal.unipdu.ac.id:8080/index.php/...
1	19	otomatisasi peringkasan dokumen seba...	maka dari itu teks fitur yang akan diekstr...	http://journal.unipdu.ac.id:8080/index.php/...
1	20	otomatisasi peringkasan dokumen seba...	dalam penelitian ini penelitia ini menggu...	http://journal.unipdu.ac.id:8080/index.php/...

Gambar 4. Hasil Tokenisasi Kalimat

4.5.Tokenisasi Kata Dokumen Teks

Tokenisasi kalimat yang telah dilakukan sebelumnya, kemudian berikutnya kalimat tersebut dilakukan tokenisasi lagi berdasarkan kata. Artinya kalimat dipisahkan berdasarkan spasi sehingga akan diperoleh kata-kata dari setiap kalimat yang ditokenisasi. Setiap kata hasil tokenisasi diberikan nomor, sehingga hubungan antara dokumen, kalimat dan kata selalu terkait. Dokumen dengan nomor id, kalimat dengan nomor no, dan kata dengan nomor n. Dari tokenisasi 104 kalimat kemudian ditokenisasi kata menghasilkan 4914 record data.

```
$query = "SELECT * FROM kalimat";
$result = mysqli_query($koneksi,$query);
$numrows = mysqli_num_rows($result);
while($row = mysqli_fetch_array($result)){
$no1 = $row['no'];
$judul1 = $row['judul'];
$isi1 = $row['kalimat'];
$url1 = $row['url'];
$kata = preg_split("/[\s]+/",$isi1);
$n=1;
foreach ($kata as $key => $val) {
$insert = "insert into katam values ('$no1','$n','$val','1');";
$insert_query = mysqli_query($koneksi,$insert);
$n++;}
```

Gambar 5. Script Tokenisasi Kata

4.6.Tokenisasi Judul

Judul juga dilakukan proses tokenisasi seperti isi dari dokumen artikel yang ada. Judul "Otomatisasi Peringkasan Dokumen Sebagai Pendukung Sistem Manajemen Surat" setelah di tokenisasi kata menghasilkan 8 kata. Pada saat tokenisasi judul hasil token diberikan nomor id 0, untuk membedakan dengan hasil tokenisasi isi dokumen.

id	kata	freq
0	otomatisasi	1
0	peringkasan	1
0	dokumen	1
0	sebagai	1
0	pendukung	1
0	sistem	1
0	manajemen	1
0	surat	1

Gambar 6. Tokenisasi Judul

4.7.Proses Stopword Judul

Proses Stopword dilakukan untuk menghilangkan kata yang dianggap tidak terlalu mempunyai arti dalam proses pencarian data atau informasi. Daftar kata stopword sudah ada sejumlah 745 kata. Sehingga daftar kata dari judul akan di bandingkan dengan daftar kata stopword. Apabila terdapat kesamaan maka kata yang terdapat di judul tersebut akan dihilangkan atau dihapus. Judul artikel terdiri dari 8 kata, setelah dilakukan proses stopword menjadi 7 kata saja. Kata yang terhapus dari proses stopword ini kata ‘sebagai’. Berarti bahwa kata ‘sebagai’ pada judul artikel termasuk dalam daftar kata stopword.

id	kata	freq
0	otomatisasi	1
0	peringkasan	1
0	dokumen	1
0	pendukung	1
0	sistem	1
0	manajemen	1
0	surat	1

Gambar 7. Hasil Stopword Judul

4.8.Proses Stemming Judul (Kata Dasar)

Judul yang telah di tokenisasi selanjutnya dilakukan proses stemming yaitu kata hasil token di update datanya menjadi kata dasar. Kata peringkasan pada judul artikel berubah menjadi ringkas, kata pendukung menjadi dukung.

id	kata	freq
0	otomatisasi	1
0	ringkas	1
0	dokumen	1
0	dukung	1
0	sistem	1
0	manajemen	1
0	surat	1

Gambar 8. Hasil Proses Stemming Judul

4.9.Menghapus Record Kosong

Langkah selanjutnya menghilangkan 396 record data yang kosong, adanya record yang kosong dikarenakan saat tokenisasi dalam sebuah kalimat terdapat spasi yang ganda. Dari penghapusan record yang kosong menghasilkan 4518 record data kata. Script yang digunakan untuk menghapus record yang kosong,

```
delete from kata where kata = "".
```

Gambar 9. Script Hapus Record Kosong

5.10.Proses Stemming Dokumen (Kata Dasar)

Stemming dilakukan pada data hasil tokenisasi kata untuk mendapatkan kata dasar dari setiap kata hasil tokenisasi. Seperti tampak pada gambar di bawah ini, terlihat bahwa kata dihitung berubah menjadi kata dasar hitung, kata berdasarkan berubah menjadi kata dasar, kata terdapat berubah menjadi kata dasar dapat, kata dibandingkan berubah menjadi kata dasar banding. Dalam hal ini berlaku untuk seluruh record data yang ada sejumlah 4518 kata.

no	n	kata	freq
1	2	fitur	1
1	3	rata	1
1	4	rata	1
1	5	frekuensi	1
1	6	kata	1
1	7	tf	1
1	8	ini	1
1	9	dihitung	1
1	10	berdasarkan	1
1	11	frekuensi	1
1	12	kata	1
1	13	yang	1
1	14	terdapat	1
1	15	dalam	1
1	16	satu	1
1	17	kalimat	1
1	18	dibandingkan	1
1	19	dengan	1
1	20	frekuensi	1
1	21	kata	1

no	n	kata	freq
1	2	fitur	1
1	3	rata	1
1	4	rata	1
1	5	frekuensi	1
1	6	kata	1
1	7	tf	1
1	8	ini	1
1	9	hitung	1
1	10	dasar	1
1	11	frekuensi	1
1	12	kata	1
1	13	yang	1
1	14	dapat	1
1	15	dalam	1
1	16	satu	1
1	17	kalimat	1
1	18	banding	1
1	19	dengan	1
1	20	frekuensi	1
1	21	kata	1

Gambar 10. Hasil Proses Stemming

4.11.Proses Stopword (Menghapus Kata Tidak Berarti)

Proses stopword merupakan sebuah proses untuk menghapus kata yang tidak berarti. Dalam arti, kata tersebut dianggap tidak terlalu penting dalam proses pencarian kata. Contoh kata tersebut yaitu di, ke, dari, yang, dan, atau dan sebagainya. Daftar stopword sudah ada dengan sejumlah 745 kata. Sehingga dari jumlah tokenisasi kata 4518 dikurangi dengan 745 daftar kata stopword yang sama menjadi 2655 kata. Berarti bahwa kata yang sama antara kata hasil tokenisasi dengan daftar kata stopword berjumlah 1863 kata. Script untuk proses stopword,

```
delete from kata where kata in (select * from stopword)
```

Gambar 11. Script Kata Stopword

4.12.Term Frequency

Menghitung term frequency seharusnya bisa dilakukan saat proses tokenisasi berlangsung, tetapi bisa juga dilakukan setelah proses stopword dan proses stemming, sehingga kata yang masih ada merupakan kata yang benar-benar sudah dapat digunakan untuk proses similarity. Script yang digunakan untuk menghitung jumlah term frequency,

```
select no,kata,count(freq) from katas group by no,kata order by no+0;
```

Gambar 12. Script Hitung Term Frequency

Dari hasil proses menghitung frequency dapat dilihat pada gambar di bawah ini, salah satunya kata ‘frekuensi’ pada kalimat no 1 mempunyai jumlah 3 kata, kata ‘kelas’ pada kalimat no 2 mempunyai jumlah 2 kata, dan seterusnya. Data setelah proses menghitung frequency sejumlah 16665 kata

no	kata	term frequency
1	frekuensi	3
1	banding	1
1	kalimat	1
1	fitur	1
1	dasar	1
1	hitung	1
1	tf	1
1	dokumen	1
2	frekuensi	3
2	kalimat	2
2	kelas	2
2	masuk	2
2	ringkas	2
2	ve	1
2	bayes	1
2	tf	1
2	na	1
2	teorema	1

Gambar 13. Hasil Hitung Term Frequency

4.13.Document Freqency

Dokumen frekuensi secara umum merupakan jumlah seluruh dokumen. Tetapi dalam hal ini dokumen yang dimaksud merupakan banyaknya kalimat dalam satu dokumen. Sehingga sebuah kalimat dalam konteks ini dianggap sebuah dokumen. Jumlah kalimat dalam sebuah dokumen ini 104 data. Script yang digunakan untuk menghitung document frequency,

```
select * from (select no,kata,freq from stemming) as a join (select count(distinct
no) doc_freq from stemming) as b
```

Gambar 14. Script Document Frequency

no	kata	term frequency	document frequency
1	frekuensi	3	104
1	banding	1	104
1	kalimat	1	104
1	fitur	1	104
1	dasar	1	104
1	hitung	1	104
1	tf	1	104
1	dokumen	1	104
2	frekuensi	3	104
2	kalimat	2	104
2	kelas	2	104
2	masuk	2	104
2	ringkas	2	104
2	ve	1	104
2	bayes	1	104
2	tf	1	104
2	na	1	104
2	teorema	1	104

Gambar 15. Hasi Perhitungan Dokumen Frekuensi

4.14.Invers Document Frequency

Dokumen invers merupakan kebalikan dari Document Frequency merupakan salah satu bagian formula untuk menghitung Weighting. Document Frequency sendiri merupakan jumlah kata dalam sebuah kalimat dikalikan dengan jumlah seluruh kalimat dalam dokumen. Sedangkan Invers Document Frequency merupakan kebalikan dari nilai dari jumlah kata dalam sebuah kalimat dikalikan dengan jumlah seluruh kalimat dalam dokumen. Script untuk menghitung Invers Document Frequency,

no	kata	term_frequency	document_frequency	invers document frequency
1	frekuensi	3	104	1.5399120845791179
1	banding	1	104	2.0170333392987803
1	kalimat	1	104	2.0170333392987803
1	fitur	1	104	2.0170333392987803
1	dasar	1	104	2.0170333392987803
1	hitung	1	104	2.0170333392987803
1	tf	1	104	2.0170333392987803
1	dokumen	1	104	2.0170333392987803
2	frekuensi	3	104	1.5399120845791179
2	kalimat	2	104	1.7160033436347992
2	kelas	2	104	1.7160033436347992
2	masuk	2	104	1.7160033436347992
2	ringkas	2	104	1.7160033436347992
2	ve	1	104	2.0170333392987803
2	bayes	1	104	2.0170333392987803
2	tf	1	104	2.0170333392987803
2	na	1	104	2.0170333392987803
2	teorema	1	104	2.0170333392987803

Gambar 16. Hasil Proses Hitung Invers Document Frequency

4.15.Weighting

Weighting merupakan suatu satu proses untuk memberikan nilai bobot pada tiap kata pada hasil tokenisasi kata. Pembobotan diperlukan untuk menilai sebuah kata pada setiap kalimat dalam isi dokumen. Nilai bobot dihitung berdasarkan jumlah kata pada setiap kalimat dalam isi dokumen. Semakin tinggi nilai bobot semakin banyak jumlah kata pada kalimat. Semakin tinggi pula kata atau kalimat tersebut sebagai kandidat yang kemungkinan akan masuk ke dalam peringkasan. Script yang digunakan untuk memperoleh bobot pada tiap kata, yaitu:

```
select *, log10(doc_freq/freq) invers, freq*(log10(doc_freq/freq)) tfidf from (select no,kata,freq from stemming) as a join (select count(distinct no) doc_freq from stemming) as b
```

Gambar 17. Script Proses Weighting

4.16.Similarity

Similarity merupakan salah satu cara untuk mencari kemiripan antara judul artikel dengan setiap kalimat yang ada dalam dokumen. Mencari kemiripan judul dengan setiap kalimat dilakukan dengan menggunakan algoritma Jaccard Similarity. Mencari kemiripan dilakukan dengan tahapan memasukkan dokumen dalam database. Dokumen dipisahkan antara judul dengan isi dokumen. Selanjutnya melakukan proses removal punctuation yaitu menghilangkan semua simbol pada judul dan juga isi dokumen. Setelah semua simbol hilang, dilanjutkan proses lowercase, yaitu mengubah semua huruf menjadi huruf kecil. Berikutnya melakukan tokenisasi kata untuk judul dan melakukan tokenisasi kalimat untuk isi dokumen. Hasil tokenisasi kalimat dari isi dokumen dilanjutkan dengan melakukan tokenisasi kata. Sehingga antara judul dan isi dokumen telah tertokenisasi dengan memisahkan semua kata terhadap judul dan juga isi dokumen.

```
Select r.id1, r.id2, r.seta, r.setb, r.a, r.b, s.c, (r.a/(r.a+r.b+s.c)) jaccard from select f.id1, f.id2, seta, setb, a, yunion, (setb-a) b from (select id1, z.kata1, id2, z.kata2, d.seta seta, e.setb setb, d.seta+e.setb yunion from (select x.id id1, x.kata kata1, y.id id2, y.kata kata2, count(x.freq) a from (SELECT * FROM tokens where id!='0' group by kata,id order by id) as x join (SELECT * FROM tokens where id='0' group by kata,id order by id) as y on x.kata=y.kata group by x.id) as z left join (select id, sum(jml) seta from (SELECT id, count(id) jml FROM tokens where id!='0' group by kata,id order by id) as x group by id) as d on z.id1=d.id left join (select id, sum(jml) setb from (SELECT id, count(id) jml FROM tokens where id='0' group by kata,id order by id) as x) as e on z.id2=e.id) as f join (select x.id id1, x.kata kata1, y.id id2, y.kata kata2, count(x.freq) a from (SELECT * FROM tokens where id!='0' group by kata,id order by id) as x join (SELECT * FROM tokens where id='0' group by kata,id order by id) as y on x.kata=y.kata group by x.id) as g on f.id1=g.id1) as r left join (select z.d1, z.d2, count(z.f1) c from (select x.id d1, x.kata t1, x.freq f1, y.id d2, y.kata t2, y.freq f2 from (SELECT * FROM tokens where id!='0' group by kata,id order by id) as x left join (SELECT * FROM tokens where id='0' group by kata,id order by id) as y on x.kata=y.kata where y.kata is null) as z group by z.d1) as s on r.id1=s.d1 order by jaccard desc
```

Gambar 18. Script Proses Similarity

Tahap berikutnya melakukan proses stemming yaitu membuat semua kata yang bukan kata dasar menjadi kata dasar. Selanjutnya dilakukan proses stopword yaitu menghapus semua kata yang dianggap tidak terlalu penting berdasarkan daftar kata stopword yang sudah ada. Berikutnya menghitung term frequency yaitu menghitung jumlah kata pada setiap kalimat, begitu juga dengan judul. Menghitung juga document frequency yaitu jumlah seluruh dokumen yang ada, dalam hal ini menghitung jumlah kalimat dalam satu dokumen. Kebalikan dari document frequency yaitu invers document frequency. Terakhir menghitung kemiripan antara judul dengan tiap kalimat dalam isi dokumen dengan algoritma Jaccard Similarity. Sehingga

didapatkan kemiripan antara judul dengan tiap kalimat dalam isi dokumen, nilai kemiripan yang paling tinggi, dalam hal ini diambil 5 dari yang paling besar nilai yang dimasukkan sebagai hasil peringkasan dokumen.

id1	id2	seta	setb	a	b	c	jaccard
30	0	11	7	6	1	4	0.5455
39	0	17	7	7	0	10	0.4118
23	0	20	7	7	0	13	0.3500
38	0	6	7	3	4	3	0.3000
63	0	20	7	6	1	14	0.2857
54	0	9	7	3	4	5	0.2500
69	0	9	7	3	4	6	0.2308
19	0	10	7	3	4	7	0.2143
62	0	5	7	2	5	3	0.2000
28	0	11	7	3	4	8	0.2000
90	0	11	7	3	4	8	0.2000
50	0	12	7	3	4	9	0.1875
56	0	6	7	2	5	4	0.1818
31	0	6	7	2	5	4	0.1818
35	0	6	7	2	5	4	0.1818
92	0	7	7	2	5	5	0.1667
93	0	7	7	2	5	5	0.1667
86	0	8	7	2	5	5	0.1667

Gambar 19. Hasil Proses Similarity

4.17.Summarization

Summarization atau peringkasan merupakan salah satu cara untuk meringkas sebuah dokumen yang terdiri dari banyak kalimat menjadi beberapa kalimat saja dengan mengambil inti dari isi dokumen sesuai judul dari dokumen tersebut. Dalam hal ini sebuah dokumen yang terdiri dari 104 kalimat kemudian dari tiap kalimat tersebut dibandingkan kemiripannya dengan judul dokumen artikel itu sendiri. Hasil perbandingan kemiripan tersebut diambil 5 kalimat yang nilai kemiripannya paling tinggi. Sehingga didapatkan kalimat no 30, 39, 23, 38 dan 63 yang mempunyai nilai kemiripan paling tinggi dibandingkan kalimat yang lain.

no	kalimat
30	kesimpulan otomatisasi peringkasan dokumen bisa diterapkan sebagai penun...
39	otomatisasi peringkasan dokumen sebagai pendukung sistem manajemen sur...
23	dengan asumsi bahwa setiap fitur teks memiliki probabilitas sendiri maka peng...
38	selain itu digunakan data uji untuk mengetahui kinerja sistem peringkasan surat
63	secara garis besar peringkasan dokumen dibagi menjadi dua pendekatan yait...

Gambar 20. Hasil Proses Summarization

```
select distinct(aa.kalimat) from (select no,kalimat from kalimat) as aa right
join (select r.id1, r.id2, r.seta, r.setb, r.a, r.b, s.c, r.a/(r.a+r.b+s.c)
jaccard from (select f.id1, f.id2, seta, setb, a, yunion, (setb-a) b from
(select id1, z.kata1, id2, z.kata2, d.seta seta, e.setb setb, d.seta+e.setb
yunion from (select x.id id1, x.kata kata1, y.id id2, y.kata kata2,
count(x.freq) a from (SELECT * FROM tokens where id!='0' group by kata,id
```

```
order by id) as x join (SELECT * FROM tokens where id='0' group by kata,id
order by id) as y on x.kata=y.kata group by x.id) as z left join (select id,
sum(jml) seta from (SELECT id, count(id) jml FROM tokens where id!='0' group
by kata,id order by id) as x group by id) as d on z.id1=d.id left join (select
id, sum(jml) setb from (SELECT id, count(id) jml FROM tokens where id='0'
group by kata,id order by id) as x) as e on z.id2=e.id) as f join (select
x.id id1, x.kata kata1, y.id id2, y.kata kata2, count(x.freq) a from (SELECT
* FROM tokens where id!='0' group by kata,id order by id) as x join (SELECT
* FROM tokens where id='0' group by kata,id order by id) as y on x.kata=y.kata
group by x.id) as g on f.id1=g.id1) as r left join (select z.d1, z.d2,
count(z.f1) c from (select x.id d1, x.kata t1, x.freq f1, y.id d2, y.kata
t2, y.freq f2 from (SELECT * FROM tokens where id!='0' group by kata,id order
by id) as x left join (SELECT * FROM tokens where id='0' group by kata,id
order by id) as y on x.kata=y.kata where y.kata is null) as z group by z.d1)
as s on r.id1=s.d1 order by jaccard desc limit 5) as bb on aa.no=bb.id1
```

Gambar 21. Script Proses Summarization

5. KESIMPULAN

Peringkasan dengan membandingkan judul artikel dengan tiap kalimat yang terdapat dalam isi dokumen menghasilkan peringkasan yang cukup baik. Metode yang digunakan untuk membandingkan judul artikel system isi dokumen menggunakan Similarity dengan algoritma Jaccard Similarity. Sehingga dihasilkan peringkasan dengan urutan kalimat dengan nilai kemiripan paling besar terhadap judul, yaitu kalimat dengan nomo 30, 39, 23, 38 dan 63.
Judul artikel "otomatisasi peringkasan dokumen sebagai pendukung system manajemen surat" dengan hasil peringkasan "otomatisasi peringkasan dokumen bisa diterapkan sebagai penunjang system manajemen surat menyurat dalam suatu organisasi. Otomatisasi peringkasan dokumen sebagai pendukung system manajemen surat. Dengan asumsi bahwa setiap fitur teks memiliki probabilitas sendiri maka penghitungan untuk menentukan kelas dari suatu kalimat adalah sebagai berikut otomatisasi peringkasan dokumen sebagai pendukung system manajemen surat e issn register jurnal ilmiah. Selain itu digunakan data uji untuk mengetahui kinerja system peringkasan surat. Secara garis besar peringkasan dokumen dibagi menjadi dua pendekatan yaitu otomatisasi peringkasan dokumen sebagai pendukung system manajemen surat pendekatan abstraktif dan ekstaktif".

DAFTAR PUSTAKA

[1] Widiantoro, Agustinus (2014) Peringkasan teks otomatis pada dokumen berbahasa Jawa menggunakan metode TF-IDF. Skripsi thesis, Sanata Dharma University. http://repository.usd.ac.id/id/eprint/1384

[2] Hermawan, Latius (2018) Peringkasan Proposal Skripsi Menggunakan Algoritma Vector Space Model. In: Seminar Nasional Sains dan Teknologi 2018, 18 Juli 2018, Universitas Wahid Hasyim, Semarang, Jawa Tengah. http://eprints.ukmc.ac.id/id/eprint/1568

[3] Najibullah, A., & Mingyan, W. (2015). Otomatisasi Peringkasan Dokumen Sebagai Pendukung Sistem Manajemen Surat. In Register: Jurnal Ilmiah Teknologi Sistem Informasi (Vol. 1, Issue 1, p. 1). Universitas Pesantren Tinggi Darul Ulum (Unipdu). https://doi.org/10.26594/register.v1i1.400

[4] A. Romadhony, F. Z.R, N. Yusliani, and L. Abednego, "Text Summarization untuk Dokumen Berita Berbahasa Indonesia," in Konferensi Nasional ICT-M Politeknik Telkom, 2017.

[5] Zamzam, M. A. (2020). Sistem Automatic Text Summarization Menggunakan Algoritma Textrank. In MATICS (Vol. 12, Issue 2, pp. 111–116). Maulana Malik Ibrahim State Islamic University. https://doi.org/10.18860/mat.v12i2.8372

[6] TF–IDF (2022) Wikipedia. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Tf%E2%80%93idf (Accessed: December 10, 2022).

[7] Setiawan, E. (no date) Kamus Besar Bahasa Indonesia (KBBI), Arti kata dokumen - Kamus Besar Bahasa Indonesia (KBBI) Online. Available at: https://kbbi.web.id/dokumen (Accessed: December 10, 2022).

[8] Setiawan, E. (no date) Kamus Besar Bahasa Indonesia (KBBI), Arti kata teks - Kamus Besar Bahasa Indonesia (KBBI) Online. Available at: https://kbbi.web.id/teks (Accessed: December 10, 2022).