**Peringkasan Artikel Berbahasa Indonesia Menggunakan *TextRank* dengan Pembobotan BM25**

**Yurdha Fadhila Hernawan 1, Putra Pandu Adikara 2, Randy Cahya Wihandika 3**

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: 1penulis.satu@xmail.ac.id, 2penulis.dua@xmail.ac.id

**Abstrak**

Penggunaan internet sebagai sumber informasi telah membawa manusia pada era *one click away*. Apa pun bisa diakses di mana pun kapan pun, baik secara visual maupun tidak. Namun, apakah setiap informasi yang diakses selalu sesuai dengan konteks yang diinginkan? Untuk memudahkan pengguna internet dalam mendapatkan informasi yang ringkas dengan tidak merusak atau menghilangkan informasi penting, maka dibutuhkan suatu peringkasan otomatis. Salah satu cara untuk mendapatkan ringkasan pada sebuah dokumen adalah dengan mencari kumpulan kalimat penting pada dokumen yang dapat merepresentasikan dokumen asli secara keseluruhan. Metode peringkasan tersebut disebut juga dengan peringkasan ekstraktif. Pada penelitian ini, peringkasan ekstraktif dilakukan dengan memeringkatkan setiap kalimat pada sebuah dokumen dan mengambil kalimat dengan peringkat teratas sebagai ringkasan. Metode *TextRank* yang digunakan pada penelitian ini akan merepresentasikan dokumen sebagai graf, setiap kalimat akan dianggap sebagai *node* dan hubungan antara kalimat (*node*) merupakan nilai *similarity* antar kalimat. Fungsi *similarity* yang digunakan adalah BM25 dengan metode pemeringkatan *PageRank*. Panjang ringkasan yang dihasilkan sistem akan disesuaikan dengan besar nilai *compression rate* yang digunakan. Setelah membandingkan hasil ringkasan yang didapatkan sistem peringkasan otomatis dengan hasil ringkasan yang didapatkan dari *expert* (pakar) sebanyak 10 dokumen, kualitas ringkasan terbaik didapatkan pada saat penggunaan *compression rate* sebesar 30% dengan nilai rata-rata *precision, recall,* dan *f-measure* secara berturut-turut adalah 0,551692; 0,551692; dan 0,551692.

**Kata kunci**: peringkasan otomatis, *TextRank*, BM25, *PageRank, compression rate*

**Abstract**

*The use of the internet as a source of information has brought humans to a one-click era. Anything can be accessed anywhere, visually or not. However, does every information accessed always match with the context itself? An automatic summarization is needed to help people to get the concise informations without ruin the context and missing the point. One way to get a summarize of the document is to find a collection of important sentences in the document that can represent the original document as a whole. That automatic text summarization method is also called extractive summarize. In this study, extractive summarization is done by checking each sentence in a document and ranking the important sentences. The TextRank method used in this study will represent the document as a graph, each sentence will be considered as a node and the relationship between sentences (nodes) is the value of similarity between sentences. The similarity function used is BM25 with the PageRank as ranking method. The resulting length of the system will be adjusted to the value of the level of compression used. After comparing the summarize result between the automatic system and an expert of 10 documents, the best quality is obtained when using a compression rate of 30% with an average value of precision, recall, and f-measure is 0.551692; 0.551692; and 0.551692.*

**Keywords**: *automatic summarization, TextRank, BM25, PageRank, compression rate*

# PENDAHULUAN

Penggunaan internet sebagai sumber informasi telah membawa manusia pada era *one click away*. Apa pun bisa diakses di mana pun kapan pun, baik secara visual maupun tidak. Namun, apakah setiap informasi yang diakses selalu sesuai dengan konteks yang diinginkan? Bisa dikatakan hanya sedikit pengguna yang dapat memahami semua informasi ketika membaca sebuah tulisan panjang (Niu et al., 2016). Kesulitan tersebut akan membuat pengguna untuk membaca ulang, sehingga akan menghabiskan banyak waktu.

Untuk memudahkan pengguna internet dalam mendapatkan informasi yang ringkas dengan tidak merusak atau menghilangkan informasi penting, maka dibutuhkan suatu peringkasan otomatis (Abbasi-ghalehtaki et al., 2016). Berdasarkan metode yang digunakan, peringkasan teks dapat dikategorikan dalam dua bentuk, yaitu ekstraktif dan abstraktif. Ringkasan ekstraktif merupakan ringkasan yang terdiri atas kumpulan dari bagian-bagian penting suatu tulisan yang dapat mewakili keseluruhan teks, sedangkan ringkasan abstraktif merupakan ringkasan yang terdiri dari kalimat baru yang dapat merepresentasikan konteks tulisan dalam bentuk lain. Selain itu, peringkasan teks juga dapat dikelompokkan berdasarkan jumlah dokumen yang digunakan menjadi *single document* dan *multi-document* (Fang et al., 2017).

Masalah utama yang muncul setelah melakukan peringkasan adalah kualitas hasil peringkasan. Apakah konteks yang dibicarakan pada hasil ringkasan sudah dapat merepresentasikan tulisan secara utuh. Penelitian oleh Pinandhita (2013) telah melakukan peringkasan ekstraktif teks berbahasa Indonesia dengan menggunakan kata benda yang terdapat dalam sebuah dokumen. Penelitian tersebut menghitung nilai kemiripan (*similarity*) antara kalimat berdasarkan kata benda yang terdapat pada setiap kalimat, lalu memeringkatkan kalimat yang paling penting dengan mengurutkan total nilai kemiripan tersebut. Evaluasi dilakukan dengan membandingkan hasil kinerja beberapa metode *similarity* agar dapat menentukan metode *similarity* terbaik dalam melakukan peringkasan.

Selain itu, peneliti lain telah melakukan penelitian untuk mendapatka n ringkasan secara abstraktif (Niu et al., 2016). Penelitian tersebut menggunakan dokumen teks opini pendek berbahasa China dengan mengelompokkan teks yang mirip lalu meringkasnya. Proses pengelompokan teks yang mirip dilakukan dengan menggunakan metode *K-Means* dengan fitur yang didapatkan dari nilai *word2vec* (nilai hubungan kedekatan antara kata). Hasil pengelompokan teks akan diperingkatkan menggunakan metode *TextRank*. Kalimat yang dianggap penting dan dapat dijadikan ringkasan memiliki peringkat tertinggi. Hasil pemeringkatan akan dijadikan ringkasan dengan menggunakan *encoder-decoder Reccurent Neural Network* (RNN) untuk membentuk kalimat baru.

Penelitian sebelumnya yang menjadi acuan dalam penelitian ini adalah peringkasan esktraktif menggunakan data *statistical* (Mussina, Aubakirov and Trigo, 2018). Data statistical merupakan nilai kesamaan *text units* dalam sebuah dokumen. *Text units* dalam dokumen dapat berupa kata, kalimat atau paragraf. Penelitian tersebut menggunakan metode *TextRank* untuk merepresentasikan kalimat dalam sebuah dokumen dan menggunakan beberapa metode *similarity*.

Berdasarkan pemaparan di atas, penulis mengajukan penelitian untuk melakukan peringkasan otomatis dengan objek artikel berita online berbahasa Indonesia menggunakan metode pemeringkatan *TextRank* dan BM25 sebagai fungsi *similarity*. BM25 dipilih berdasarkan penelitian sebelumnya yang telah melakukan pengujian menggunakan beberapa fungsi *similarity* dan mendapatkan bahwa BM25 menghasilkan ringkasan yang lebih baik dari pada fungsi *similarity* lainnya (Barrios et al., 2016).

# METODOLOGI DAN PERANCANGAN

Salah satu penelitian yang menjadi acuan penelitian yang akan dilakukan penulis adalah pencarian ringkasan secara ekstraktif oleh Mussina, Aubakirov & Trigo (2018). Ringkasan yang dihasilkan tidak melakukan perubahan terhadap struktur kalimat. Cara yang digunakan untuk mendapatkan ringkasan ekstraktif adalah menggunakan data *statistical,* yaitu dengan menghitung kesamaan *text units*. *Text units* tersebut dapat berupa kata, kalimat atau paragraf. Penelitian tersebut menggunakan dokumen mengenai bencana alam berbahasa Rusia dan Kazakhstan, yang mana memiliki struktur kalimat yang jelas.

Penelitian Mussina, Aubakirov & Trigo (2018) merepresentasikan dokumen menggunakan metode *TextRank,* dengan kalimat sebagai *node* dan nilai *similarity* antara kalimat sebagai *edges.* Panjang ringkasan yang diambil adalah sebanyak 30% dari total panjang dokumen. Nilai *similarity* yang digunakan adalah *content overlap,* BM25*,* dan *substring* terpanjang yang muncul antara dua kallimat. Namun, penelitian tersebut tidak menggunakan metode *PageRank* dalam proses pemeringkatan kalimat, melainkan dengan menjumlahkan seluruh nilai *similarity* yang dimiliki kalimat dan mengurutkannya berdasarkan jumlah nilai terbesar. Secara umum, setiap kalimat akan saling berkaitan antara satu dengan yang lainnya, namun terkadang ada kalimat yang tidak memiliki kesamaan dengan kalimat lain. Kalimat tersebut tidak memiliki nilai *similarity* sehingga tidak akan dijadikan ringkasan.

Sebelum melakukan pengambilan ringkasan berdasarkan nilai fungsi *silimarity*, penelitian Mussina, Aubakirov & Trigo (2018) mereduksi kalimat berdasarkan nilai *treshold*. Kalimat yang memiliki jumlah nilai *similarity* kurang dari nilai treshold tidak dijadikan ringkasan. Pengambilan ringkasan diambil sebanyak 30% dari panjang dokumen berdasarkan kalimat yang memiliki nilai *similarity* terbesar. Kalimat terpilih dijadikan ringkasan berdasarkan urutan sesuai dokumen awal. Evaluasi pada penelitian tersebut dilakukan dengan mencari nilai distribusi *key-words,* dengan nilai rata-rata untuk nilai *similarity* menggunakan *content overlap, LongestCommonSubstring*, dan BM25secara berurutan adalah 0,180; 0,175; dan 0,169.

Penelitian lain yang menjadi acuan penelitian ini adalah percobaan untuk membandingkan hasil ringkasan ekstraktif menggunakan beberapa nilai *similarity* dalam *TextRank* (Barrios et al., 2016)*.* Cara mengidentifikasi hubungan antara kalimat satu dengan kalimat lainnya adalah dengan menghitung *Longest Common Substring*, *cosine distance*, dan kesamaan *query* yang dianggap penting. Pada penelitian tersebut dokumen akan direpresentasikan sebagai graf dengan kalimat sebagai *node* dan nilai *similarity* sebagai *edges* (hubungan antara *nodes*). Fungsi similarity yang digunakan adalah *Longest Common Substring, cosine distance,* BM25*,* dan BM25*+* dengan proses pemeringkatan *PageRank*.

Dokumen yang digunakan pada penelitian Barrios et al., (2016)adalah *Document Understanding Conference (DUC)* yang berjumlah sebanyak 567 dokumen dengan peringkasan sebanyak 20% dari tiap panjang dokumen. Hasil ringkasan dari percobaan tersebut dievaluasi menggunakan metode *ROUGE-N* dengan nilai terbaik didapatkan pada peringkasan menggunakan BM25 dan BM25+.

Berdasarkan pemaparan tersebut, tahapan penelitian akan dimulai dengan pengumpulan data artikel berita berbahasa Indonesia yang akan digunakan untuk menguji sistem peringkasan yang dibuat. Selanjutnya perancangan dan implementasi sistem peringkasan otomatis menggunakan metode *TextRank* dengan pembobotan BM25, lalu menguji sistem tersebut dengan membandingkan hasil ringkasan yang didapatkan oleh sistem otomatis dengan hasil ringkasan yang didapatkan oleh *expert* (pakar) yang telah ditentukan.

Artikel berita yang digunakan pada penelitian ini didapatkan menggunakan metode *web* *scrapping,* sehingga data masukan yang dibutuhkan merupakan URL berita portal *online* berbahasa Indonesia. Pada metode *TextRank,* artikel berita akan direpresentasikan sebagai graf, yang mana setiap kalimat dianggap sebagai *node* dan nilai pembobotan BM25 sebagai *edge* antar *node*. Selanjutnya setiap kalimat (*node*) akan diperingkatkan berdasarkan nilai *PageRank,* kalimat dengan nilai *PageRank* yang tinggi dianggap sebagai kalimat penting dan dapat merepresentasikan dokumen sebagai ringkasan. Gambaran umum sistem peringkasan yang dibuat dapat dilihat pada Gambar 2.1.

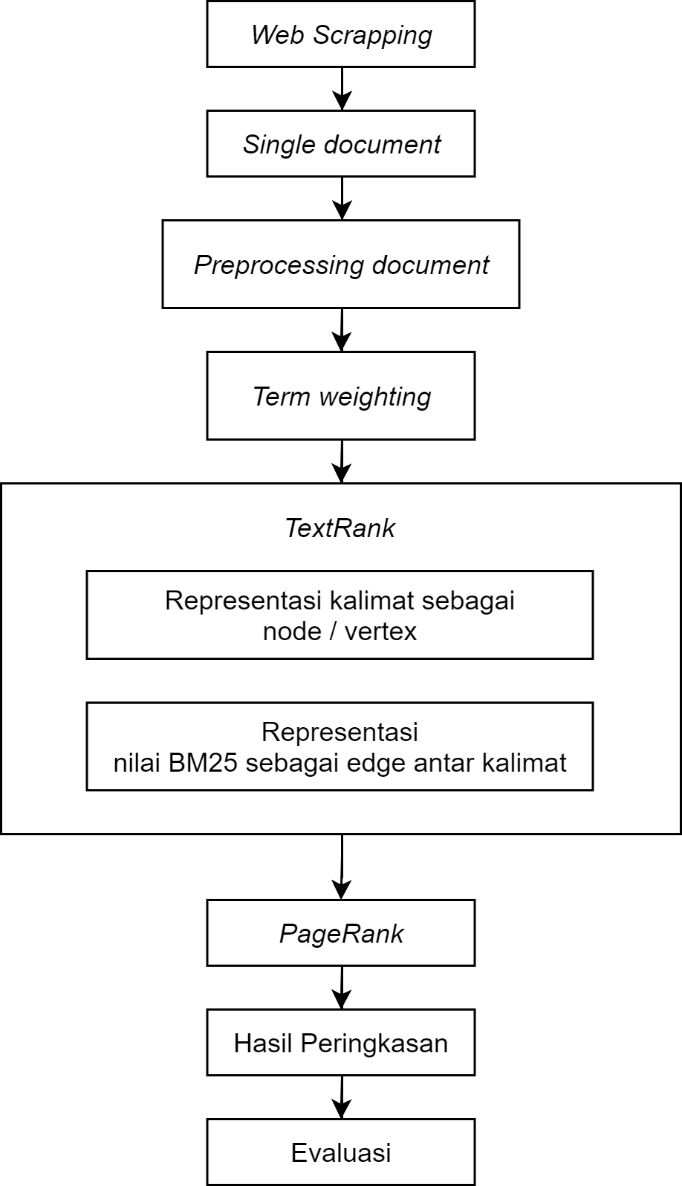
## 2.1 Pengumpulan Data

Artikel berita berbahasa Indonesia yang digunakan pada penelitian ini diambil melalui situs berita BBC Indonesia sebanyak 10 berita mengenai bencana alam dan lingkungan. Pertimbangan jumlah berita yang sedikit diambil karna penelitian ini berfokuskan pada penelitian yang mendalam berdasarkan jenis *compression rate* yang digunakan untuk mendapatkan ringkasan.

## 2.2 Perancangan Sistem

Langkah kerja peringkasan artikel berita berbahasa Indonesia diuraikan sebagai berikut:

1. Mempersiapkan URL berita *online* yang akan diringkas.
2. Melakukan ekstraksi judul dan konten berita dengan menggunakan *web scrapping.*
3. Melakukan *preprocessing* konten berita hasil *web scrapping.* Proses *preprocessing* terdiri atas segmentasi, *cleaning, stemming,* dan tokenisasi.
4. Menghitung *term weighting.*
5. Menerapkan metode *TextRank* dengan merepresentasikan kalimat sebagai *node* graf dan nilai bobot BM25 sebagai edge antara *node.*
6. Melakukan pemeringkatan setiap kalimat menggunakan metode *PageRank.*
7. Membentuk ringkasan berdasarkan nilai *PageRank* dan *compression rate* yang digunakan.



Gambar 2.1 Gambaran Umum Proses Peringkasan Otomatis

### 2.2.1 *Web Scrapping*

*Web scrapping* digunakan untukmengakses halaman *web* yang diinginkan dan mengambil data tertentu pada halaman tersebut, lalu mengubahnya sehingga menjadi kumpulan data terstruktur yang dapat digunakan untuk proses komputasi selanjutnya (Boeing and Waddell, 2017). Data yang digunakan dalam melakukan *web scrapping* adalah URL dari halaman portal berita online. *Web scrapping* akan menerima HTML data dari halaman *web* tersebut, lalu data tersebut akan direduksi sehingga hanya akan menghasilkan judul berita dan konten berita yang akan digunakan dalam proses peringkasan.

### 2.2.2 Preprocessing

*Preprocessing* dilakukan sebelum proses pembentukan ringkasan dan mempermudah proses peringkasan. Dalam preprocessing terdapat tiga tahapan yaitu segmentasi, *cleaning,* *stemming,* dan tokenisasi.

#### 2.2.2.1 Segmentasi

Segmentasi merupakan proses pemecahan teks dokumen menjadi kalimat-kalimat untuk mempermudah pemrosesan dokumen menjadi potongan-potongan yang lebih kecil.

#### 2.2.2.2 *Cleaning*

Pada proses cleaning dilakukan penghilangan tanda baca dan URLkarna dianggap tidak mempengaruhi isi dokumen.

#### 2.2.2.3 *Stemming*

*Stemming* merupakan metode pembentukan kata atau term menjadi kata dasar. Proses *stemming* dilakukan dengan membuang imbuhan yang terdapat pada term.

#### 2.2.2.4 Tokenisasi

Tokenisasi adalah metode pemecahan teks kalimat menjadi token-token (*term*) yang berurutan. Pada proses ini juga dilakukan penghilangan angka dan kata hubung atau kata yang dianggap tidak mempenggaruhi isi dari konten dokumen menggunakan metode *stopwords removal*. Hal ini juga dilakukan guna meningkatkan performa sistem agar sistem bisa secara efektif dimanfaatkan untuk pengolahan konten yang benar-benar dianggap penting saja.

### 2.2.3 *Term Weighting*

#### 2.2.3.1 *Term Frequency (tf)*

Perhitungan *term frequency* merupakan langkah awal dari *term weighting*. *Term frequency* adalah jumlah kemunculan setiap *term* dalam satu kalimat.

#### 2.2.3.2 *Document Frequency (df)*

*Document frequency* merupakan jumlah dokumen yang memiliki kemunculan sebuah *term*. Nilai *document frequency* akan digunakan untuk mendapatkan nilai *Inverse document frequency*.

#### 2.2.3.3 *Inverse Document Frequency (Idf)*

Nilai *inverse document frequency* akan digunakan untuk perhitungan BM25. Mengacu pada penelitian yang dilakukan (Lv and Zhai, 2011) perhitungan *Idf* ditunjukan pada Persamaan 2.1 :

Keterangan:

*N* = Total jumlah kalimat

= *Document frequency* tiap *term* dalam satu dokumen

### 2.2.4 *TextRank*

*TextRank* merupakan sebuah algoritme berbasis graf yang digunakan untuk menentukan *node* mana yang paling penting dalam suatu graf (Tarau, 1973). Struktur *TextRank* ditunjukan pada *Gambar 2.2*. Penggunaan *TextRank* dapat dilakukan dalam melakukan penarikan keputusan. Proses melakukan *TextRank* dalam peringkasan teks ekstraktif dimulai dengan mengidentifikasi *single document* yang digunakan. Lalu setiap kalimat direpresentasikan sebagai *node* dan hubungan antara kalimat merupakan fungsi *similarity* yang direpresentasikan sebagai *edges*. *Edges* pada graf dapat memiliki arah maupun tidak. Setelah graf terbentuk, maka selanjutnya akan dilakukan pemeringkatan graf dan mengurutkan *node* yang memiliki nilai pemeringkatan paling tinggi (paling penting).Setelah diurutkan, maka peringkasan dapat diambil berdasarkan peringkat dari kalimat-kalimat tersebut.



Gambar 2.2 Struktur TextRank Sebagai Graf

### 2.2.5 Fungsi *Similarity* BM25

Dalam merepresentasikan dokumen sebagai graf, *edges* didapatkan dari hasil fungsi *similarity* antar kalimat. Fungsi *similarity* didapatkan dari kemiripan isi kalimat satu dengan kalimat lainnya. Kalimat yang merepresentasikan suatu konteks dalam dokumen akan merekomendasikan kalimat lain yang memiliki konteks yang sama (Tarau, 1973). Fungsi *similarity* yang digunakan dalam penelitian ini adalah BM25. Nilai BM25 didapatkan dari perhitungan bobot *tf* dan *idf* pada setiap kata (*term*). Selain itu juga ditambahkan parameter bebas *k1*dan *b* dengan nilai *k1* sebesar 1.2 dan *b* sebesar 0.75 (Manning, C.C.; Ragghavan, P.; Schütze, 2009). Persamaan BM25 dijabarkan pada Persamaan 2.2:

Keterangan:

= Nilai *idf term t*

*k1* dan *b* = Parameter penskalaan terhadap *tf* dan panjang dokumen

*tftd* = Frekuensi term t pada kalimat d

*Ld* dan *Lave*  = Panjang kalimat *d* dan rata-rata dari panjang seluruh koleksi kalimat

### 2.2.6 *PageRank*

*PageRank* adalah metode yang digunakan dalam pemeringkatan graf. *PageRank* digunakan oleh Google untuk menentukan tingkat kepentingan halaman *web*. *PageRank* merupakan nilai mumerical yang menyatakan seberapa penting sebuah halaman *web* di internet. Singkatnya, perhitungan nilai tersebut bertambah bila halaman tersebut muncul sebagai sebuah *hyperlink* di sebuah halaman web lainnya. Semakin besar nilai yang dimiliki, maka semakin penting *web* tersebut. Begitu juga dengan kalimat yang saling berhubungan satu sama lain dalam sebuah graf. Kalimat yang penting akan memiliki nilai *PageRank* yang besar.

Inisialisasi awal nilai *PageRank* tiap kalimat ditentukan secara random mulai dari 0 hingga 1. Lalu sejumlah iterasi dilakukan untuk melakukan update bobot *PageRank* ditiap kalimat. Persamaan PageRank dijabarkan pada Persamaan 2.3:

Keterangan:

= Bobot nilai *PageRank* kalimat i

*d* = *Dampening factor*

= Kalimat *j* yang berhubungan dengan kalimat *i*

= Nilai fungsi *similarity* antara kalimat *j* dan *i*

= Kalimat *k* yang berhubungan dengan kalimat *j*

= Nilai fungsi *similarity* antara kalimat *j* dan *k*

= Bobot nilai *PageRank* kalimat j

*Dampening factor* (d) adalah nilai yang telah dihitung oleh Google Engineers dalam sistem *PageRank* untuk memastikan bahwa bobot *node* akan konvergen pada satu nilai. Nilai *dampening factor* bisa didapatkan dari angka *random* mulai dari nol hingga satu, namun 0.85 telah menjadi nilai yang umum saat menetapkan nilai *dampening factor*. Pada akhir perhitungan graf kalimat dapat diilustrasikan seperti pada Gambar 2.3.



Gambar 2.3 Struktur Textrank Setelah Perhitungan Pagerank

### 2.2.7 *Compression Rate*

*Compression rate* digunakan sebagai acuan panjang ringkasan yang akan dihasilkan sistem peringkasan otomatis. Semakin kecil kompresi rasio (*compression rate*) yang digunakan maka semakin banyak informasi yang dibuang, namun semakin besar kompresi rasio yang digunakan maka semakin tidak penting informasi yang dihasilkan (Alguliev & Aliguliyev, 2009). Besar kompresi rasio yang diterima untuk peringkasan adalah sebesar 5-30% (Hahn & Mani, 2000).

### 2.2.8 Evaluasi Hasil Ringkasan

Pengujian sistem peringkasan ini akan dilakukan dengan membandingkan nilai *precision, recall,* dan *f-measure* untuk setiap hasil ringkasan. Proses perhitungan tersebut dibantu dengan menggunakan *confussion matrix*.

***Confussion Matrix***

*Confusion Matrix* merupakan informasi mengenai tingkat relevansi hasil peringkasan yang didapatkan oleh sistem. Contoh *confusion matrix* ditunjukkan pada Tabel 2.5.

Tabel 2.1 Contoh Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | ***Relevant*** | ***Not relevant*** |
| ***Retrieved*** | A | B |
| ***Not retrieved*** | C | D |

Keterangan :

A = Semua data yang dihasilkan sistem dan relevan bagi *expert*

B = Semua data yang dihasilkan sistem dan tidak relevan bagi *expert*

C = Semua data yang relevan bagi *expert* dan tidak dihasilkan sistem

D = Semua data yang tidak relevan bagi *expert* dan tidak dihasilkan sistem

***Precision***

Merupakan nilai ketepatan antara informasi yang dihasilkan sistem dengan hasil informasi yang seharusnya (dianggap benar). Persamaan *precision* ditunjukan pada Persamaan 2.5:

Keterangan:  
A = Semua data yang dihasilkan sistem dan relevan bagi *expert*

B = Semua data yang dihasilkan sistem dan tidak relevan bagi *expert*

***Recall***

Merupakan tingkat keberhasilan sistem dalam peringkasan otomatis yang menentukan berapa proporsi kalimat yang dipilih oleh pakar yang juga dipilih oleh sistem. Persamaan *recall* ditunjukan pada Persamaan 2.6:

Keterangan:  
A = Semua data yang dihasilkan sistem dan relevan bagi *expert*

C = Semua data yang relevan bagi *expert* dan tidak dihasilkan sistem

***F-Measure***

Merupakan pengukuran yang menilai timbal balik antara *precision* dan *recall.* Persamaan *F-Measure* ditunjukan pada Persamaan 2.7:

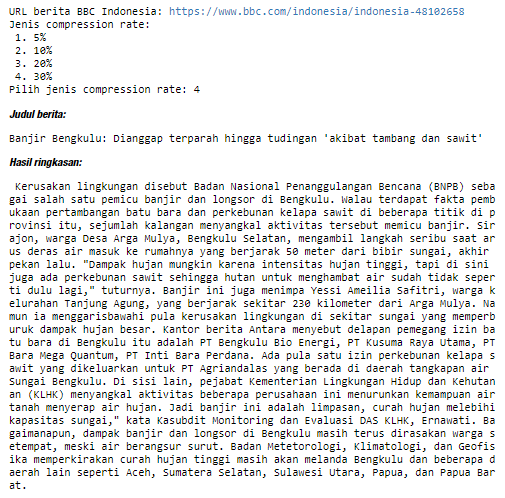
(2.7)

Keterangan:  
P = Precision  
R = Recall

# 3. IMPLEMENTASI SISTEM

Implementasi penelitian ini dilakukan dengan menggunakan bahasa pemrograman *python* dan menggunakan *software* Jupyter Notebook*.* Untuk mempermudah proses scrapping dalam mendapatkan dokumen berita, implementasi sistem dibantu dengan menggunakan library BeautifulSoup. Selanjutnya sistem akan mengambil judul berita dan konten berita berdasarkan URL yang diberikan oleh *user.*

Panjang ringkasan yang dihasilkan sistem akan disesuaikan dengan besar *compression rate* yang dipilih. Jenis *compression rate* yang digunakan pada sistem ini adalah sebesar 5%, 10%, 20%, dan 30%. Setelah memilih besar *compression rate* yang diinginkan, sistem akan mengeluarkan judul dokumen berita dan hasil ringkasan secara otomatis. Hasil implementasi peringkasan artikel berita berbahasa Indonesia menggunakan *TextRank* dengan pembobotan BM25 ditunjukkan pada Gambar 3.1.



Gambar 3.0.1 Hasil Implementasi

# 4. PENGUJIAN DAN ANALISIS

Pengujian dilakukan dengan membandingkan hasil ringkasan 10 artikel berita yang didapatkan dari pakar dan hasil ringkasan yang didapatkan dari sistem. Pakar yang terlibat dalam penelitian ini adalah dosen Bahasa Indonesia dari Universitas Brawijaya. Ringkasan akan dihasilkan berdasarkan bobot *compression rate* yang digunakan. Pada evaluasi ini bobot yang digunakan adalah 5%, 10%, 20%, dan 30%. Tingkat kualitas yang didapatkan akan dihitung sesuai perhitungan *precision, recall,* dan *f-measure*.

# 4.1 Hasil Pengujian dan Analisis *Compression Rate* 5%

Tabel 4.1 menunjukan nilai dari *precision, recall,* dan *f-measure* dari sepuluh dokumen yang diuji menggunakan nilai *compression rate* sebesar 5%. Beberapa dokumen memiliki nilai *precision, recall,* dan *f-measure* sebesar nol, yang mana menunjukan bahwa tidak ada kalimat ringkasan yang sama antara hasil ringkasan sistem otomatis dengan hasil ringkasan yang didapatkan oleh pakar. Untuk 10 dokumen yang diuji nilai rata-rata *precision, recall,* dan *f-measure* secara berturut-turut adalah 0,233333333; 0,233333333; dan 0,233333333. Perbandingan nilai *precision, recall,* dan *f-measure* untuk setiap dokumen dapat dilihat pada Gambar 4.1.

**Tabel 4.1 Hasil Pengujian Compression Rate 5%**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokumen** | ***Precision*** | ***Recall*** | ***F-Measure*** |
| Dokumen 1 | 0,5 | 0,5 | 0,5 |
| Dokumen 2 | 0,5 | 0,5 | 0,5 |
| Dokumen 3 | 0,5 | 0,5 | 0,5 |
| Dokumen 4 | 0 | 0 | 0 |
| Dokumen 5 | 0 | 0 | 0 |
| Dokumen 6 | 0 | 0 | 0 |
| Dokumen 7 | 0 | 0 | 0 |
| Dokumen 8 | 0,5 | 0,5 | 0,5 |
| Dokumen 9 | 0 | 0 | 0 |
| Dokumen 10 | 0,333333 | 0,333333 | 0,333333 |
| **Rata - rata** | 0,233333 | 0,233333 | 0,233333 |

**Gambar 4.1 Evaluasi Pengujian Compression Rate 5%**

# 4.2 Hasil Pengujian dan Analisis C*ompression Rate* 10%

Tabel 4.2 menunjukan nilai dari *precision, recall,* dan *f-measure* dari sepuluh dokumen yang diuji menggunakan nilai *compression rate* sebesar 10%. Berbeda dengan Tabel 4.1, pengujian dengan menggunakan *compression rate* 10% memiliki lebih sedikit nilai *precision, recall,* dan *f-measure* sebesar nol. Hal ini menunjukan bahwa hasil ringkasan yang didapatkan sistem otomatis memiliki lebih banyak kesamaan dengan hasil ringkasan yang didapatkan pakar dibandingkan dengan pengujian dengan menggunakan *compression rate* 5%. Untuk 10 dokumen yang diuji nilai rata-rata *precision, recall,* dan *f-measure* secara berturut-turut adalah 0,266667; 0,266667; dan 0,266667. Perbandingan nilai *precision, recall,* dan *f-measure* untuk setiap dokumen dapat dilihat pada Gambar 4.2.

**Tabel 4.2 Hasil Pengujian Compression Rate 10%**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokumen** | ***Precision*** | ***Recall*** | ***F-Measure*** |
| Dokumen 1 | 0,25 | 0,25 | 0,25 |
| Dokumen 2 | 0,333333 | 0,333333 | 0,333333 |
| Dokumen 3 | 0,333333 | 0,333333 | 0,333333 |
| Dokumen 4 | 0 | 0 | 0 |
| Dokumen 5 | 0,5 | 0,5 | 0,5 |
| Dokumen 6 | 0 | 0 | 0 |
| Dokumen 7 | 0,25 | 0,25 | 0,25 |
| Dokumen 8 | 0,5 | 0,5 | 0,5 |
| Dokumen 9 | 0 | 0 | 0 |
| Dokumen 10 | 0,5 | 0,5 | 0,5 |
| **Rata - rata** | 0,266667 | 0,266667 | 0,266667 |

**Gambar 4.2 Evaluasi Pengujian Compression Rate 5%**

# 4.3 Hasil Pengujian dan Analisis C*ompression Rate* 20%

Tabel 4.3 menunjukan nilai dari *precision, recall,* dan *f-measure* dari sepuluh dokumen yang diuji menggunakan nilai *compression rate* sebesar 20%. Untuk pengujian ini tidak lagi mendapatkan hasil nilai *precision, recall,* dan *f-measure* sebesar nol. Hasil *precision, recall,* dan *f-measure* terendah didapatkan pada dokumen 3, dokumen 4, dan dokumen 6. Sedangkan untuk hasil *precision, recall,* dan *f-measure* tertinggi didapatkan pada dokumen 7 yaitu sebesar 0,571429; 0,571429; dan 0,571429. Untuk 10 dokumen yang diuji nilai rata-rata *precision, recall,* dan *f-measure* secara berturut-turut adalah 0,368831; 0,368831; dan 0,368831. Perbandingan nilai *precision, recall,* dan *f-measure* untuk setiap dokumen dapat dilihat pada Gambar 4.3.

**Tabel 4.3 Hasil Pengujian Compression Rate 20%**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokumen** | ***Precision*** | ***Recall*** | ***F-Measure*** |
| Dokumen 1 | 0,375 | 0,375 | 0,375 |
| Dokumen 2 | 0,5 | 0,5 | 0,5 |
| Dokumen 3 | 0,166667 | 0,166667 | 0,166667 |
| Dokumen 4 | 0,142857 | 0,142857 | 0,142857 |
| Dokumen 5 | 0,545455 | 0,545455 | 0,545455 |
| Dokumen 6 | 0,166667 | 0,166667 | 0,166667 |
| Dokumen 7 | 0,571429 | 0,571429 | 0,571429 |
| Dokumen 8 | 0,428571 | 0,428571 | 0,428571 |
| Dokumen 9 | 0,375 | 0,375 | 0,375 |
| Dokumen 10 | 0,416667 | 0,416667 | 0,416667 |
| **Rata - rata** | 0,368831 | 0,368831 | 0,368831 |

**Gambar 4.3 Evaluasi Pengujian Compression Rate 20%**

# 4.4 Hasil Pengujian dan Analisis C*ompression Rate* 30%

Tabel 4.4 menunjukan nilai dari *precision, recall,* dan *f-measure* dari sepuluh dokumen yang diuji menggunakan nilai *compression rate* sebesar 30%. Hasil *precision, recall,* dan *f-measure* terendah didapatkan pada dokumen 1, yaitu sebesar 0,416667; 0,416667; dan 0,416667. Sedangkan untuk hasil *precision, recall,* dan *f-measure* tertinggi didapatkan pada dokumen 7 yaitu sebesar 0,7; 0,7; dan 0,7. Besar nilai *precision, recall,* dan *f-measure* untuk penggunaan *compression rate* sebesar 30% yang tidak lagi mendapatkan nilai nol dan lebih baik dibandingkan dengan penggunaan *compression rate* sebesar 20% menunjukan bahwa tingkat keberhasilan sistem cukup tinggi pada pengujian ini. Untuk 10 dokumen yang diuji nilai rata-rata *precision, recall,* dan *f-measure* secara berturut-turut adalah 0,551692; 0,551692; dan 0,551692. Perbandingan nilai *precision, recall,* dan *f-measure* untuk setiap dokumen dapat dilihat pada Gambar 4.4.

**Tabel 4.4 Hasil Pengujian Compression Rate 30%**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokumen** | ***Precision*** | ***Recall*** | ***F-Measure*** |
| Dokumen 1 | 0,416667 | 0,416667 | 0,416667 |
| Dokumen 2 | 0,5 | 0,5 | 0,5 |
| Dokumen 3 | 0,444444 | 0,444444 | 0,444444 |
| Dokumen 4 | 0,5 | 0,5 | 0,5 |
| Dokumen 5 | 0,625 | 0,625 | 0,625 |
| Dokumen 6 | 0,444444 | 0,444444 | 0,444444 |
| Dokumen 7 | 0,7 | 0,7 | 0,7 |
| Dokumen 8 | 0,636364 | 0,636364 | 0,636364 |
| Dokumen 9 | 0,583333 | 0,583333 | 0,583333 |
| Dokumen 10 | 0,666667 | 0,666667 | 0,666667 |
| **Rata - rata** | 0,551692 | 0,551692 | 0,551692 |

**Gambar 4.3 Evaluasi Pengujian Compression Rate 30%**

# 5. PENUTUP

# 5.1 Kesimpulan

Berdasarkan implementasi dan evaluasi sistem peringksasan otomatis yang telah berhasil dilakukan dapat disimpulkan beberapa hal sebagai berikut:

1. Sistem peringkasan artikel berita berbahasa Indonesia menggunakan TextRank dengan pembobotan BM25 dapat menghasilkan ringkasan secara ekstraktif dengan cara memeringkatkan setiap kalimat pada satu artikel berita. Jumlah kalimat yang dijadikan ringkasan dipengaruhi oleh besar compression rate yang digunakan.
2. Setelah membandingkan hasil ringkasan yang didapatkan sistem peringkasan otomatis dengan hasil ringkasan yang didapatkan dari expert (pakar) sebanyak 10 dokumen, kualitas ringkasan terbaik didapatkan pada saat penggunaan compression rate sebesar 30% dengan nilai rata-rata precision, recall, dan f-measure secara berturut-turut adalah 0,551692; 0,551692; dan 0,551692.

# 5.2 Saran

Beberapa hal yang dapat diperbaiki dan menjadi masukan untuk penelitian selanjutnya ialah agar dapat memperhatikan posisi kalimat pada saat melakukan peringkasan ekstraktif, memperhatikan kutipan kalimat yang didapatkan dari narasumber pada artikel berita, dan memperhatikan kalimat yang dicetak miring atau dicetak tebal.

# 6. DAFTAR RUJUKAN

Abbasi-ghalehtaki, R., Khotanlou, H. and Esmaeilpour, M., 2016. Fuzzy evolutionary cellular learning automata model for text summarization. *Swarm and Evolutionary Computation*, [online] 30, pp.11–26. Available at: <http://dx.doi.org/10.1016/j.swevo.2016.03.004>.

Alguliev, R. and Aliguliyev, R., 2009. Evolutionary Algorithm for Extractive Text Summarization. 2009(November), pp.128–138.

Barrios, F., López, F., Argerich, L. and Wachenchauzer, R., 2016. Variations of the Similarity Function of TextRank for Automated Summarization. [online] Available at: <http://arxiv.org/abs/1602.03606>.

Boeing, G. and Waddell, P., 2017. New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings. *Journal of Planning Education and Research*, 37(4), pp.457–476.

Brin, S. and Page, L., 1998. The anatomy of a large-scale hypertextual Web search engine BT - Computer Networks and ISDN Systems. *Computer Networks and ISDN Systems*, [online] 30(1–7), pp.107–117. Available at: <http://dx.doi.org/10.1016/S0169-7552(98)00110-X%5Cnhttp://apps.webofknowledge.com/full\_record.do?product=UA&search\_mode=GeneralSearch&qid=6&SID=X1pOWPMuSmOv1SlwJ6f&page=1&doc=2>.

Fang, C., Mu, D., Deng, Z. and Wu, Z., 2017. Word-sentence co-ranking for automatic extractive text summarization. *Expert Systems with Applications*, [online] 72, pp.189–195. Available at: <http://dx.doi.org/10.1016/j.eswa.2016.12.021>.

Fhadli, M., Fauzi, M.A. and Afirianto, T., 2017. Peringkasan Literatur Ilmu Komputer Bahasa Indonesia Berbasis Fitur Statistik dan Linguistik menggunakan Metode Gaussian Naïve Bayes. 1(4), pp.307–319.

Hahn, U. and Mani, I., 2000. of Automatic Researchers are investigating summarization tools and methods that. *Computer 33.11*, (November), pp.29–36.

Lv, Y. and Zhai, C., 2011. Lower-bounding term frequency normalization. *International Conference on Information and Knowledge Management, Proceedings*, pp.7–16.

Manning, C.C.; Ragghavan, P.; Schütze, H., 2009. *An Introduction to Information Retrival*. Cambridge, England: Cambridge University Press.

Munot, N. and S. Govilkar, S., 2013. Comparative Study of Text Summarization in Indian Languages. *International Journal of Computer Applications*, 75(6), pp.17–21.

Mussina, A., Aubakirov, S. and Trigo, P., 2018. Automatic Document Summarization based on Statistical Information. (Data), pp.71–76.

Niu, J., Zhao, Q., Wang, L., Chen, H., Atiquzzaman, M. and Peng, F., 2016. OnSeS: A novel online short text summarization based on BM25 and neural network. *2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings*, pp.1–6.

Pinandhita, R.R., 2013. Peringkas Dokumen Berbahasa Indonesia Berbasis Kata Benda Dengan BM25.

Radev, D. R., Hovy, E., & McKeown, K., 2002. Introduction to the special issue on summarization. *Computational Linguistics*, 28(4), pp.399–408.

Sankarasubramaniam, Y., Ramanathan, K. and Ghosh, S., 2014. Text summarization using Wikipedia. *Information Processing and Management*, [online] 50(3), pp.443–461. Available at: <http://dx.doi.org/10.1016/j.ipm.2014.02.001>.

Tarau, R.M. and P., 1973. TextRank: Bringing Order into Texts. *Comparative Biochemistry and Physiology -- Part B: Biochemistry and*, [online] 45(4). Available at: <http://www.aclweb.org/anthology/W04-3252>.

Yeh, J.Y., Ke, H.R., Yang, W.P. and Meng, I.H., 2005. Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing and Management*, 41(1), pp.75–95.