PERINGKASAN ARTIKEL BERBAHASA INDONESIA MENGGUNAKAN *TEXTRANK* DENGAN PEMBOBOTAN BM25

PROPOSAL SKRIPSI

Disusun oleh:

Yurdha Fadhila Hernawan

NIM: 165150200111094



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2019

DAFTAR ISI

[DAFTAR ISI i](#_Toc22258099)

[DAFTAR TABEL ii](#_Toc22258100)

[DAFTAR GAMBAR iii](#_Toc22258101)

[BAB 1 PENDAHULUAN 1](#_Toc22258102)

[1.1 Latar Belakang 1](#_Toc22258103)

[1.2 Rumusan Masalah 2](#_Toc22258104)

[1.3 Tujuan 2](#_Toc22258105)

[1.4 Manfaat 2](#_Toc22258106)

[1.5 Batasan Masalah 2](#_Toc22258107)

[1.6 Sistematika Pembahasan 2](#_Toc22258108)

[BAB 2 LANDASAN KEPUSTAKAAN 4](#_Toc22258109)

[2.1 Kajian Pustaka 4](#_Toc22258110)

[2.2 Dasar Teori Peringkasan Teks 5](#_Toc22258111)

[2.3 *Preprocessing* 6](#_Toc22258112)

[2.3.1 Segmentasi 6](#_Toc22258113)

[2.3.2 Tokenisasi 7](#_Toc22258114)

[2.3.3 Stemming 7](#_Toc22258115)

[2.4 *TextRank* 7](#_Toc22258116)

[2.5 Fungsi Similarity BM25 8](#_Toc22258117)

[2.6 *PageRank* 8](#_Toc22258118)

[2.7 Evaluasi 9](#_Toc22258119)

[BAB 3 METODOLOGI 12](#_Toc22258120)

[3.1 Tempat dan Waktu Penelitian 12](#_Toc22258121)

[3.2 Pengambilan Data 12](#_Toc22258122)

[3.3 Metode Penelitian 12](#_Toc22258123)

[3.4 Peralatan Pendukung 13](#_Toc22258124)

[3.5 Teknik Analisis Data 14](#_Toc22258125)

[DAFTAR RUJUKAN 15](#_Toc22258126)

DAFTAR TABEL

[Tabel 2.1 Hasil segmentasi dokumen 6](#_Toc22258190)

[Tabel 2.2 Contoh *confusion matrix* 9](#_Toc22258191)

[Tabel 3.1 Jadwal penelitian 12](#_Toc22258192)

DAFTAR GAMBAR

[Gambar 2.1 Hasil tokenisasi kalimat index 1 7](file:///D:\college\script\proposal\revisi%20p0\PERINGKASAN%20ARTIKEL%20BERBAHASA%20INDONESIA%20MENGGUNAKAN%20TEXTRANK%20DENGAN%20PEMBOBOTAN%20BM25.docx#_Toc22258205)

[Gambar 2.2 Hasil *stemming* kalimat index 1 7](file:///D:\college\script\proposal\revisi%20p0\PERINGKASAN%20ARTIKEL%20BERBAHASA%20INDONESIA%20MENGGUNAKAN%20TEXTRANK%20DENGAN%20PEMBOBOTAN%20BM25.docx#_Toc22258206)

[Gambar 2.3 Struktur *TextRank* sebagai graf 8](#_Toc22258207)

[Gambar 2.4 Struktur TextRank setelah perhitungan PageRank 11](#_Toc22258208)

[Gambar 3.1 Diagram proses peringkasan teks 13](file:///D:\college\script\proposal\revisi%20p0\PERINGKASAN%20ARTIKEL%20BERBAHASA%20INDONESIA%20MENGGUNAKAN%20TEXTRANK%20DENGAN%20PEMBOBOTAN%20BM25.docx#_Toc22258209)

# PENDAHULUAN

## Latar Belakang

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 hasil ringkasannya, 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 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 mengenai peringkasan teks bukanlah hal baru dalam bidang *Natural Language Processing* (NLP). Salah satu penelitian peringkasan teks terhadap novel *online* dilakukan dengan memanfaatkan metode BM25 dan Neural Network (Niu et al., 2016). Peringkasan ini memilih menggunakan *deep learning* daripada TF IDF untuk menghasilkan kalimat baru yang lebih singkat, hal ini dikarenakan penggunaan TF IDF akan menghasilkan ringkasan yang tidak lengkap.

Peringkasan teks juga dapat dilakukan dengan bantuan metode *supervised* dan *unsupervised learning* (Mao et al., 2019)*.* Pada saat melakukan peringkasan teks dengan bantuan *supervised learning*, pembobotan kalimat dilakukan secara pengklasifikasian biner. Apakah kalimat dapat dikategorikan sebagai ringkasan atau tidak. Penelitian lain yang menggunakan metode *TextRank* dan Gensim menghasilkan kualitas ringkasan yang lebih baik dengan menggunakan fungsi *similarity* BM25 (Barrios et al., 2016).

Tidak hanya peringkasan berbahasa Inggris, peringkasan teks berbahasa Indonesia sudah dilakukan oleh beberapa peneliti sebelumnya. Salah satu penelitian mengenai peringkasan teks, menggunakan fitur statistik dan linguistik untuk mengklasifikasi kalimat yang akan dijadikan ringkasan (Fhadli et al., 2017). Metode yang digunakan dalam peringkasan tersebut adalah *Gaussian Naïve Bayes*. Namun penelitian ini menghasilkan ringkasan dengan kualitas yang kurang baik dengan nilai *f-score* 0,206538 dan rata-rata *relative utility* 0,116657.

Pengujian beberapa fungsi *similarity* telah dilakukan untuk mendapatkan peringkasan otomatis berdasarkan kata benda (Pinandhita, 2013). Hasil yang didapatkan ternyata menunjukan bahwa memang peranan kata benda sangat berpengaruh dalam mendapatkan suatu ringkasan. Selain itu penelitian ini juga menunjukan bahwa kinerja peringkasan dengan menggunakan BM25 menghasilkan ringkasan yang lebih baik dari pada *cosine similarity* dan *content overlap*.

Berdasarkan pemaparan di atas, penulis mengajukan penelitian untuk melakukan peringkasan otomatis dengan objek artikel berita *online* berbahasa Indonesia menggunakan metode *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 (Pinandhita, 2013).

## Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan, dapat dirumuskan masalah sebagai berikut:

1. Apakah *TextRank* dapat menghasilkan ringkasan ekstraktif otomatis dengan kualitas yang baik?

## Tujuan

Menghasilkan ringkasan ekstraktif secara otomatis dengan kualitas yang baik menggunakan *TextRank*

## Manfaat

Selain menghasilkan ringkasan secara otomatis, penelitian ini dapat digunakan dalam *Question Answering System, Information retrival*, dan ekstraksi informasi.

## Batasan Masalah

Artikel online yang digunakan adalah artikel berita berbahasa Indonesia yang memiliki tulisan sesuai Ejaan Bahasa Indonesia (EBI)

## Sistematika Pembahasan

Laporan penelitian ini terdiri atas beberapa bagian yaitu sebagai berikut.

**Bab I Pendahuluan**

Berisikan latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat, batasan masalah yang diberikan, dan sistematika pembahasan.

**Bab II Landasan Kepustakaan**

Berisikan pembahasan mengenai teori yang dijadikan pedoman pengerjaan, konsep pengerjaan penelitian, metode atau sistem yang diterapkan, pustaka ilmiah yang berkitan dengan *Natural Language Processing, Text Summarize, dan TextRank*.

**Bab III Metode Penelitian**

Berisikan langkah pengerjaan dalam penelitian, teknik yang digunakan, data yang akan digunakan, dan representasi berdasarkan metode yang dipilih untu menyelesaikan masalah.

**Bab IV Perancangan**

Berisikan proses penyelesaian masalah dengan visualisasi diagram alir dan perhitungan manual dari metode yang digunakan.

**Bab V Implementasi**

Berisikan implementasi metode yang digunakan dalam menyelesaikan masalah dalam sebuah sistem.

**Bab VI Pengujian dan Analisis**

Berisikan scenario pengujian dan analisis terhadap hasil pengujian.

**Bab VII Penutup**

Berisikan kesimpulan dan saran sebagai rekomendasi untuk penelitian selanjutnya.

# LANDASAN KEPUSTAKAAN

## Kajian Pustaka

Penelitian dalam melakukan peringkasan teks telah banyak dilakukan sebelumnya. Salah satunya dengan memanfaatkan kata benda yang terdapat pada sebuah dokumen (Pinandhita, 2013). Penelitian tersebut dilakukan untuk mendapatkan ringkasan ekstraktif dengan membandingkan beberapa metode *similarity* yang didapatkan dari kata benda. Dokumen yang digunakan berupa artikel koran dengan topik di luar pertanian yang juga digunakan pada penelitian Miptahudin (2010) dan dokumen dari penelitian Aristoteles (2011). Hasil peringkasan yang didapatkan dari beberapa metode *similarity* yang digunakan Pinandhita akan dibandingkan dengan hasil peringkasan yang didapatkan Miptahudin (2010) dan Aristoteles (2011).

Total percobaan yang dilakukan Pinandhita adalah sebanyak tujuh kali (dengan mengikutsertakan judul dokumen dan tanpa judul dokumen) yaitu penerapan *PageRank* dengan bobot *cosine similarity*, *PageRank* dengan bobot BM25, *PageRank* dengan bobot *content overlap*, bobot *cosine similarity* tanpa PageRank, bobot BM25 tanpa *PageRank*, bobot *content overlap* tanpa *PageRank*, dan yang terakhir menggunakan bobot koefisien *dice*. Rata rata panjang dokumen yang digunakan pada penelitian ini adalah sebanyak 47 kalimat yang mengandung rata rata sebanyak 282 kata benda. Pengujian dilakukan dengan mencari nilai *kappa* berdasarkan kesepakatan hasil ringkasan antara tiga orang dosen Jurusan Sastra Indonesia dan tiga orang mahasiswa. Kualitas ringkasan terbaik didapatkan dari hasil pembobotan BM25 dengan *PageRank* tanpa menggunakan judul dokumen.

Selain mendapatkan ringkasan secara ekstraktif, peneliti lain juga telah melakukan penelitian untuk mendapatkan ringkasan secara abstraktif (Niu et al., 2016). Penelitian tersebut menggunakan dokumen teks opini pendek berbahasa China dengan mengelompokkan teks yang mirip lalu meringkasnya. Proses peringkasan dibagi atas tiga tahap yaitu *clustering* teks (mengelompokkan), pemeringkatan teks, dan peringkasan teks. Proses *clustering* dilakukan dengan menggunakan metode *K-Means* dengan fitur yang telah didapatkan dari nilai *word2vec*. *Word2vec* menunjukan hubungan kedekatan antara suatu kata dengan kata lainnya.

Hasil *clustering* yang telah didapatkan akan diperingkatkan menggunakan model pemeringkatan graf yaitu *textrank* dengan bantuan pembobotan BM25, setelahnya hasil pemeringkatan tertinggi akan diringkas secara abstraktif. Peringkasan abstraktif dilakukan dengan metode *encoder-decoder Reccurent Neural Network (RNN)*. Pengujian kualitas ringkasan yang dihasilkan pada penelitian ini dilakukan dengan mencari nilai *precision, recall, F-measure, ROUGE-N, dan ROUGE-L*. Secara berurut rata-rata nilai *precision, recall, dan F-measure* yang didapatkan adalah sebesar 0,94; 0,932; dan 0,933. Nilai ROUGE-N didapatkan dari hasil kesamaan *N-gram* yang serupa dari hasil peringkasan manual dengan hasil ringkasan yang didapatkan sistem. Pada evaluasi ini digunakan ROUGE-1 (*unigrams*) dan ROUGE-2 (*bigrams*).

Peringkasan dokumen bisa dilakukan dengan metode klasifikasi kalimat (Fhadli, 2017). Kalimat akan diklasifikasikan sebagai kelas yang termasuk ringkasan dan kelas yang termasuk bukan ringkasan. Fitur yang digunakan dalam pengklasifikasian ini adalah fitur statistik dan fitur linguistik, sedangkan dokumen yang digunakan dalam penelitian ini adalah literatur Ilmu Komputer Berbahasa Indonesia.

Fitur statistik didapatkan dari nilai TF-IDF sebuah kalimat, sedangkan fitur linguistik didapatkan dari banyaknya kata pada judul yang terdapat pada kalimat (title word), posisi kalimat pada dokumen (sentence location), panjang kalimat (sentence length), kalimat yang mengandung akronim atau singkatan (upper-case word), kalimat yang memiliki frasa yang dianggap penting seperti "jadi", "hasilnya", dan "kesimpulannya" (cue phrase), kalimat yang mengandung kata spesifik yang menyatakan sebuah topik dokumen (biased word), dan kalimat yang mengandung kata-kata yang dianggap tidak penting (occurrence of non-essential information). Proses pengklasifikasian kalimat dilakukan dengan menggunakan metode Naive Bayes. Naive Bayes bekerja dengan cara menghitung peluang suatu kalimat terhadap kelas dengan bantuan data latih. Hasil kualitas ringkasan diuji dengan melakukan pencarian nilai precision, recall, F-measure, dan relative utility. Hasil rata-rata F-measure dan relative utility yang didapatkan adalah 0,206538 dan 0,116657.

## Dasar Teori Peringkasan Teks

Ketika jumlah informasi *online* semakin banyak, maka kebutuhan akan sistem yang dapat merangkum satu atau lebih dokumen secara otomatis sangat diperlukan (Radev et al., 2002). Selain dapat membantu dalam mengatasi informasi yang berlebihan, peringkasan otomatis juga berguna dalam penyajian informasi singkat mengingat ukuran perangkat *handy* yang digunakan pembaca (Sankarasubramaniam et al., 2014). Ukuran ringkasan biasanya tidak lebih dari setengah dokumen aslinya. Berdasarkan kebutuhannya ringkasan dokumen terbagi atas (Munot and S. Govilkar, 2013):

Metode

Abstraktif

Ringkasan yang terdiri dari kalimat baru yang dapat merepresentasikan konteks tulisan dalam bentuk kalimat lain.

Ekstraktif

Ringkasan yang terdiri atas kumpulan dari bagian-bagian penting suatu tulisan yang dapat mewakili keseluruhan teks.

Konten

*Generic*

Ringkasan umum yang tidak bergantung pada syarat apapun.

*Query based*

Ringkasan hanya didapatkan berdasarkan *query* yang diinginkan pengguna.

Jumlah dokumen

*Single document*

Ringkasan yang didapatkan dari satu dokumen.

*Multi document*

Ringkasan yang didapatkan dari beberapa dokumen.

Bahasa

*Monolingual*

Ringkasan yang didapatkan dari dokumen dengan bahasa yang sama.

*Multilingual*

Ringkasan yang didapatkan dari beberapa dokumen dengan bahasa yang berbeda.

Dalam penelitian ini peringkasan akan dilakukan secara ekstraktif dengan memilih kalimat yang dianggap penting dan dapat dijadikan ringkasan. Dokumen yang digunakan merupakan *single document* dengan berbahasa Indonesia.

## *Preprocessing*

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

### Segmentasi

Segmentasi merupakan proses pemecahan teks dokumen menjadi kalimat-kalimat untuk mempermudah pemrosesan dokumen menjadi potongan-potongan yang lebih kecil. Pada penelitian yang akan dilakukan, judul pada artikel berita diikutsertakan dalam percobaan guna membantu performa sistem dalam mendapatkan konteks tulisan karna judul dianggap memiliki informasi penting mengenai konteks dari isi dokumen. Contoh hasil segmentasi dokumen ditunjukan pada Tabel 2.1.

Tabel 2.1 Hasil segmentasi dokumen

|  |  |
| --- | --- |
| Index | Kalimat |
| 1 | Dia mengambilnya dengan senang hati. |
| 2 | Esoknya setelah pulang sekolah, ia melarikan diri. |
| 3 | Ibu yang bekerja sebagai pembantu. |

### Tokenisasi

Tokenisasi adalah metode pemecahan teks kalimat menjadi token-token (term) yang berurutan. Pada proses ini juga dilakukan penghilangan 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. Contoh hasil tokenisasi dokumen ditunjukan pada Gambar 2.1.

mengambilnya

senang hati

Gambar 2.1 Hasil tokenisasi kalimat index 1

### Stemming

*Stemming* merupakan metode pembentukan kata atau term menjadi kata dasar. Biasanya proses *stemming* dilakukan dengan membuang imbuhan yang terdapat pada term. Contoh hasil tokenisasi dokumen ditunjukan pada Gambar 2.2.

ambil

senang hati

Gambar 2.2 Hasil *stemming* kalimat index 1

## *TextRank*

*TextRank* merupakan sebuah algoritma berbasis graf yang digunakan untuk menentukan *node* mana yang paling penting dalam suatu graf (Tarau, 1973). Struktur *TextRank* ditunjukan pada. 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 2.3 Struktur *TextRank* sebagai graf

## 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 text akan merekomendasikan kalimat lain yang memiliki konteks yang sama (Tarau, 1973). Fungsi similarity yang digunakan 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 et al., 2009). Persamaan BM25 dijabarkan pada Persamaan 2.3:

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

## *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 2.4:

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 2.2.

## Evaluasi

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

*Confusion Matrix*

*Confusion Matrix* merupakan informasi mengenai tingkat relevansi hasil peringkasan yang didapatkan oleh sistem. Contoh *confusion matrix* ditunjukkan pada Tabel 2.2.

Tabel 2.2 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

Recall

Merupakan tingkat jumlah banyak dan sedikitnya kesesuaian informasi yang didapatkan dari hasil percobaan berdasarkan sudut pandang kelas atau label yang digunakan. Persamaan *recall* ditunjukan pada Persamaan 2.6:

(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 bobot *harmonic mean* pada *recall* dan *precision.* Persamaan *F-Measure* ditunjukan pada Persamaan 2.7:

(2.7)

Keterangan:  
P = Precision  
R = Recall



Gambar 2.4 Struktur *TextRank* setelah perhitungan *PageRank*

# METODOLOGI

## Tempat dan Waktu Penelitian

Penelitian yang diajukan bersifat Non-implementatif dengan jenis penelitian eksperimen. Yang mana hasil dari penelitian ini akan dinilai oleh pakar untuk mengetahui kualitas ringkasan yang dihasilkan dari beberapa fungsi *similarity* yang digunakan. Hasil tersebut akan dibandingkan dan dianalisis menggunakan metode evaluasi.

## Pengambilan Data

Jumlah populasi pada penelitian ini termasuk ke dalam kategori tak terbatas karna dokumen yang digunakan merupakan artikel berita online berbahasa Indonesia. Populasi tak terbatas yaitu sumber datanya tidak dapat ditentukan batasan-batasannya sehingga relatif tidak dapat ditentukan dalam bentuk angka. Teknik pengambilan sampel yang digunakan adalah *simple random sampling*. Yaitu pengambilan sampel dari anggota populasi secara acak tanpa memperhatikan tingkatan, karena populasi bersifat homogen (sama).

Instrumen pengumpulan dokumen pada penelitian ini adalah peneliti sendiri, yang mana dokumen diambil dari artikel berita online BBC Indonesia yang dapat diakses melalui situs resmi BBC Indonesia. Jumlah dokumen yang akan diambil sebagai dokumen pengujian sistem adalah sebanyak 30 artikel berita yang terdiri dari enam kategori yaitu politik, komputer, lingkungan, olahraga, bencana alam, dan tren sosial.

## Metode Penelitian

Penelitian ini terbagi atas dua tahap, yaitu pembentukan ringkasan dan evaluasi ringkasan yang dihasilkan. Tahap pembentukan sendiri terdiri atas tahapan *preprocessing*, penerapan *TextRank dan PageRank,* dan pembentukan ringkasan*.* Hasil ringkasan yang didapatkan dari penelitian ini adalah kalimat yang memiliki nilai *PageRank* yang besar setelah diurutkan. Kalimat tersebut dianggap penting dan dapat mereprenstasikan keseluruhan dokumen sehingga dapat dijadikan ringkasan. Total kalimat yang akan diambil sebagai ringkasan ialah sebanyak 25% dari total keseluruhan kalimat. Karna 25% hasil ekstrak dari sebuah teks sumber memiliki tingkat informasi yang sama besar dengan teks itu sendiri (Miptahudin, 2010). Secara umum, diagram proses peringkasan teks pada penelitian ini ditunjukan pada Gambar 3.1.

Gambar 3.1 Diagram proses peringkasan teks

*TexkRank*

*Single document*

*Preprocessing*

Tokenisasi

Segmentasi

Stemming

Representasi kalimat sebagai node / vertex

Representasi nilai BM25 sebagai edge antar kalimat

*PageRank*

Evaluasi

Hasil ringkasan

## Peralatan Pendukung

Lingkungan implementasi system *hardware* meliputi:

Processor 2.00 GHz

RAM 8.00 GB

Spesifikasi *software* meliputi:

1. Operating System Windows 10 64-bit

2. Jupyter Notebook

## Teknik Analisis Data

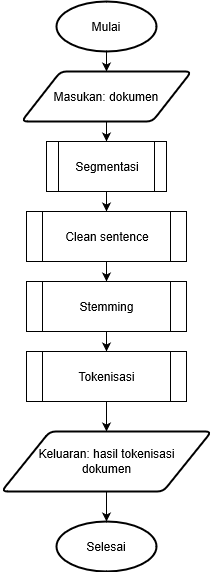
Pengujian hasil ringkasan yang dihasilkan sistem akan dilakukan oleh dosen Bahasa Indonesia dari Fakultas Ilmu Budaya Universitas Brawijaya. Dosen Bahasa Indonesia tersebut dijadikan pakar dan diminta untuk memilih kalimat yang dianggap penting dan dapat dijadikan ringkasan pada sekumpulan dokumen yang diujikan. Hasil pemilihan kalimat pada tiap dokumen akan dibandingkan dengan hasil peringkasan otomatis yang didapatkan oleh sistem yang dibuat menggunakan *textrank.* Perhitungan *precision, recall,* dan *F-Measure* digunakan untuk mengetahui kualitas ringkasan otomatis tersebut.

# perancangan dan implementasi

## Diagram Alir Sistem

### Preprocessing

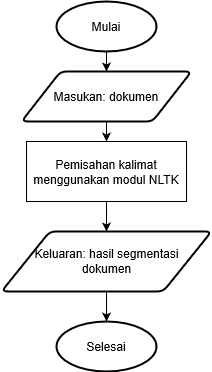
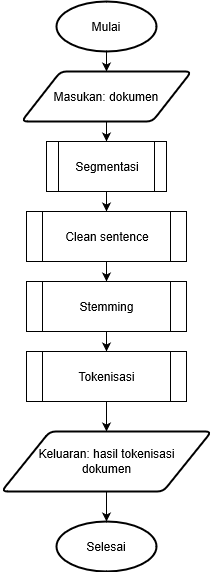
Pada preprocessing dokumen berita akan dibersihkan dengan cara menghilangkan URL, angka, symbol, dan *stopword*. *Stopword* merupakan kumpulan kata yang dianggap tidak penting dan tidak mempengaruhi isi dokumen. *Stopword* dapat dihilangkan sehingga proses peringkasan dapat berfokus pada kata yang dianggap penting dan sesuai dengan konteks dokumen. Setelah pembersihan dokumen, tiap kalimat akan dipecah menjadi token-token. Alur *preprocessing* ditunjukan pada Gambar 4.1.



Gambar 4.1 Diagram alir *preprocessing* dokumen

#### Segmentasi

Segmentasi merupakan proses pemisahan dokemen menjadi potongan-potongan kalimat. Proses segmentasi dilakukan dengan bantuan *library* NLTK dengan modul *sent\_tokenize.* Hal ini dilakukan agar pemisahan kalimat tidak hanya mengandalkan tanda titik karna pemisahan bisa terjadi ditengah nama orang atau jalan memiliki titik karna adanya singkatan atau gelar. Alur segmentasi ditunjukan pada Gambar 4.2.



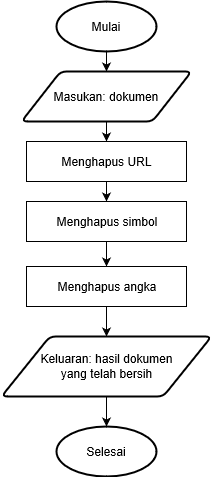
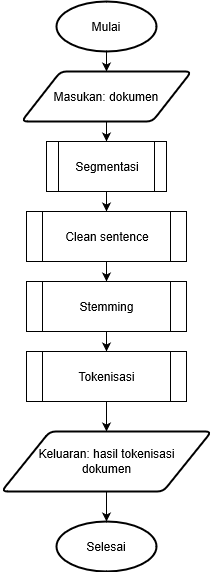
Gambar 4.2 Diagram alir segmentasi dokumen

#### Clean Sentence

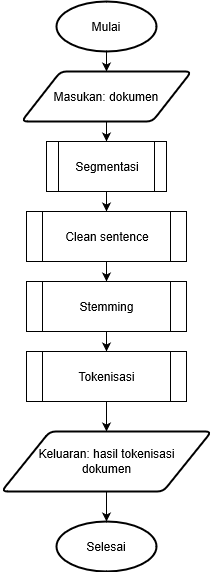
Pada proses ini dilakukan pembersihan dokumen dengan menghilangkan URL, symbol, dan angka. Hal ini diperlukan untuk mendapatkan kata penting yang dianggap dapat merepresentasikan keseluruhan dokumen. Alur *clean sentence* ditunjukan pada Gambar 4.3.

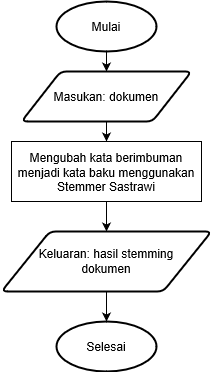
#### Stemming

Stemming dilakukan untuk menghilangkan imbuhan pada kata dalam kalimat sehingga akan menghasilkan kata dasar. Proses stemming dilakukan dengan bantuan *library* stemming bahasa IndonesiaStemmer Sastrawi. Hasil stemming akan digunakan untuk proses berikutnya yaitu tokenisasi. Alur dari stemming ditunjukan pada Gambar 4.4.



Gambar 4.3 Diagram alir *clean sentence*

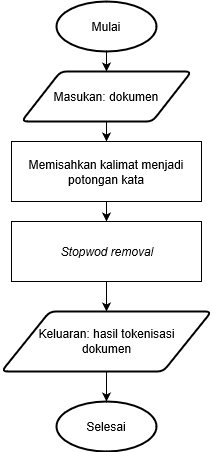
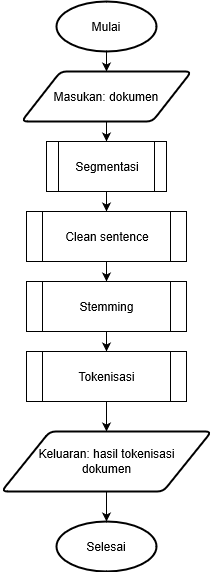




Gambar 4.4 Diagram alir stemming dokumen

#### Tokenisasi

Proses tokenisasi dilakukan untuk memisahkan kalimat menjadi potongan-potongan token atau kata. Pada proses ini juga dilakukan *stopword* *removal,* yaitu penghilangan kata yang dianggap tidak penting dan tidak mempengaruhi kata penting yang dapat berkaitan dengan konteks dokumen. Alur tokenisasi ditunjukan pada Gambar 4.5.



Gambar 4.5 Diagram alir tokenisasi

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>.

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>.

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.

Mao, X., Yang, H., Huang, S., Liu, Y. and Li, R., 2019. Extractive Summarization Using Supervised and Unsupervised Learning. *Expert Systems with Applications*. [online] Available at: <https://linkinghub.elsevier.com/retrieve/pii/S0957417419303240>.

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.

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>.