

**PERINGKASAN TEKS OTOMATIS
MENGUNAKAN METODE PEMBOBOTAN KALIMAT PADA
ARTIKEL ILMIAH BAHASA INDONESIA**

(Skripsi)

**Oleh :
Desti Fatmalasari**



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

ABSTRAK

PERINGKASAN TEKS OTOMATIS MENGGUNAKAN METODE PEMBOBOTAN KALIMAT PADA ARTIKEL ILMIAH BAHASA INDONESIA

Oleh

Desti Fatmalasari

Teknologi peringkasan dokumen digunakan untuk menghemat waktu dalam menggali informasi penting pada dokumen. Peringkasan teks merupakan proses penyusutan teks lebih pendek namun tetap mempertahankan informasi yang terkandung didalamnya.

Penelitian ini membahas mengenai peringkasan teks artikel ilmiah bahasa indonesia menggunakan metode pembobotan kalimat berupa *TF-IDF* dan *Similarity*. Tujuan yang ingin dicapai oleh sistem dapat meringkas teks dengan mengenali pola pada dokumen teks dalam file format *.txt*. Sistem dibangun menggunakan bahasa PHP sebagai bahasa pemrograman. Uji coba yang dilakukan menggunakan UAT (*User Acceptance Testing*) untuk mengetahui tanggapan responen terhadap sistem yang diinterpretasikan, yaitu dengan angket skala likert dengan membagi 3 aspek penilaian.

Dari hasil pengolahan data angket (kuantitatif) diperoleh nilai sebesar 82,6% untuk tampilan sistem, nilai sebesar 80,2% untuk efisiensi kalimat yang

dihasilkan pada ringkasan sistem, dan nilai 83,7% untuk kepuasan dalam menggunakan sistem peringkasan teks otomatis pada artikel ilmiah Bahasa Indonesia. Hasil pengujian dan implementasi terhadap sistem peringkasan teks otomatis diterima dengan tingkat penerimaan yang tergolong kuat.

Kata Kunci: Peringkasan Teks, Artikel Ilmiah, Pembobotan Kalimat, *TF-IDF*, *Similarity*

***AUTOMATIC TEXTING USING SENTENCE WEIGHTING METHODS IN
INDONESIAN SCIENTIFIC JOURNAL***

ABSTRACT

By

Desti Fatmalasari

Technology of document monitoring is used to save time in digging up important information on documents. Summarize is a process of shrinking text shorter but still retaining the information contained therein.

This research discusses the commemoration of the text of journal scientific using sentence weighting methods in the form of TF-IDF and Similarity. The goal the system wants to achieve can be to summarize text by recognizing patterns on text documents in txt format files. The system was built using PHP as a programming language. The trial was conducted using UAT (User Acceptance Testing) to find out the response to the interpreted system, namely by the likerts scale questionnaire by dividing 3 aspects of the assessment.

From the results of data processing (quantitative) obtained a value of 82.6% for the appearance of the system, a value of 80.2% for the efficiency of sentences generated in the system summary, and a value of 83.7% for satisfaction in using an automatic texting system in scientific articles Indonesian. The results of testing and implementation of the automatic text alerting system are received with a relatively strong acceptance rate.

Key words: *Text Appropriation, Scientific Articles, Sentence Weighting, TF-IDF, Similarity*

**PERINGKASAN TEKS OTOMATIS MENGGUNAKAN METODE
PEMBOBOTAN KALIMAT PADA ARTIKEL ILMIAH ILMIAH BAHASA
INDONESIA**

Oleh

Desti Fatmalasari

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA ILMU KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

Judul Skripsi : PERINGKASAN TEKS OTOMATIS
MENGUNAKAN PEMBOBOTAN KALIMAT
PADA ARTIKEL ILMIAH BAHASA
INDONESIA

Nama Mahasiswa : Desti Fatmalasari

Nomor Pokok Mahasiswa : 1517051037

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

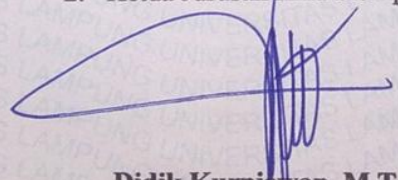
MENYETUJUI

1. Komisi Pembimbing



Favorisen R. Lumbanraja, Ph. D
NIP. 198301102008121002

2. Ketua Jurusan Ilmu Komputer

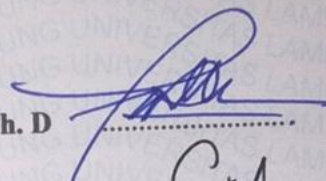


Didik Kurniawan, M.T.
NIP. 19800419200501 1 004

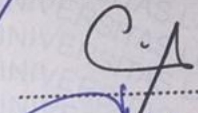
MENGESAHKAN

1. Tim Penguji

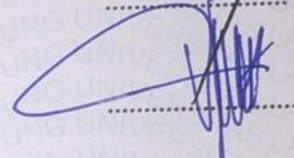
Penguji 1 : Favorisen R. Lumbanraja, Ph. D



Penguji Pembahas 1: Rico Andrian, S.Si., M.Kom



Penguji Pembahas 2: Didik Kurniwan, S.Si., M.T.



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Satripto Dwi Yuwono, S.Si., M.T.
NIP. 19740705 200003 1 001



Tanggal Lulus Ujian Skripsi: 27 Januari 2022

PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa Skripsi saya yang berjudul “Peringkasan Teks Otomatis Menggunakan Metode Pembobotan Kalimat Pada Artikel Ilmiah Bahasa Indonesia” merupakan karya saya sendiri. Seluruh tulisan yang tertuang di dalam Skripsi ini telah mengikuti Kaidah Penulisan Karya Ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 27 Januari 2022



Desti Fatmalasari
NPM. 1517051037

RIWAYAT HIDUP



Penulis bernama lengkap Desti Fatmalasari, dilahirkan di Nambahdadi pada tanggal 05 Desember 1996. Penulis merupakan anak pertama dari tiga bersaudara dari pasangan Bapak Zainudin dan Ibu Istu Ningsih.

Penulis memulai pendidikan pertama di Taman Kanak-kanak (TK) Gula Putih Mataram pada tahun 2002. Penulis melanjutkan pendidikan Sekolah Dasar di SDS 02 Gula Putih Mataram pada tahun 2003 sampai 2009. Penulis melanjutkan pendidikan Sekolah Menengah Pertama (SMP) di SMP SGula Putih Mataram pada Tahun 2009 sampai 2012. Kemudian melanjutkan Sekolah Menengah Atas (SMA) di SMAS Muhammadiyah Braja Sebah pada Tahun 2012 sampai 2015.

Pada tahun 2015 penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Lampung melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN).

Selama menjadi mahasiswa, penulis aktif di beberapa kegiatan di antaranya:

1. Peserta Karya Wisata Ilmiah (KWI) di Pekon Batutegi, Kecamatan Air Nanningan, Kabupaten Tanggamus pada Januari 2016.
2. Anggota Aktif Biro Kesekretariatan Unit Kegiatan Mahasiswa Fakultas (UMKF) Rohani Islam FMIPA Unila pada 2015-2017.

3. Aktif dalam Organisasi Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) Universitas Lampung dengan menjabat sebagai Anggota Bidang Kestari pada tahun 2015-2016.
4. Pada bulan Juli 2018 sampai dengan Agustus 2018 penulis melaksanakan Kuliah Kerja Nyata (KKN) di Desa Way Areng, Kecamatan Mataram Baru, Kabupaten Lampung Timur, Provinsi Lampung.
5. Peserta lomba kejuaraan Pencak Silat Tapak Suci di Universitas Lampung pada 19 Desember 2017.
6. Pada bulan Januari 2018 sampai dengan Maret 2018 penulis melaksanakan Kerja Praktik (KP) di SMK Muhammadiyah Braja Selehah.

MOTTO

*“Maka jangan sekali-kali membiarkan kehidupan dunia ini memperdayakan
kamu”*

(Q.S Fatir: 5)

*“Sukses bukanlah final, kegagalan tidaklah fatal. Keberanian untuk
melanjutkan yang penting.”*

(Winston S. Churchill)

I Do it

I Can Do it

I Will Do it

(Desti Fatmalasari)

PERSEMBAHAN

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa atas segala berkat dan rahmat-Nya sehingga Skripsi ini dapat terselesaikan.

Kupersembahkan karya ini kepada:

Teristimewa untuk kedua orang tuaku Bapak Zainudin dan Ibu Istu Ningsih serta keluarga besarku.

Terima kasih atas segala semangat, cinta, doa, motivasi serta dukungan yang telah diberikan yang tak ternilai sehingga tercapainya keberhasilanku.

Keluarga besar Classic A Ilmu Komputer 2015,
Serta Almamater Tercinta
Universitas Lampung.

SANWACANA

Assalamualakum wr. wb.

Puji syukur kehadirat Tuhan Yang Maha Esa atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi dengan judul “Peringkasan Teks Otomatis Menggunakan Metode Pembobotan Kalimat Pada Artikel Ilmiah Bahasa Indonesia” sebagai salah satu syarat untuk memperoleh gelar sarjana Ilmu Komputer di Universitas Lampung.

Dalam menyusun skripsi ini, penulis mendapat dukungan dan bantuan dari berbagai pihak, baik materi maupun non-materi. Ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada semua pihak yang telah membantu penulis, antara lain:

1. Kedua orang tua tercinta, Bapak Zainudin dan Ibu Istu Ningsih, kedua Adikku Bagus Adhi Kusuma dan Ananda Kurniawanserta keluarga besar yang selalu memberikan do'a, semangat dan motivasi yang sangat besar.
2. Bapak Bapak Favorisen R. Lumbanraja, Ph. D, sebagai pembimbing utama yang telah memberikan ide, saran, dan motivasi kepada penulis.
3. Bapak Rico Andrian, S.Si., M.Kom., sebagai pembahas yang telah memberikan saran kepada penulis dalam perbaikan skripsi.
4. Bapak Didik Kurniawan, S.Si., M.T., selaku pembahas yang telah memberikan saran kepada penulis dalam perbaikan skripsi.
5. Bapak Dr. Eng. Suropto Dwi Yuwono, S.Si., M.T., selaku Dekan FMIPA Universitas Lampung.
6. Bapak Didik Kurniawan S.Si., M.T., selaku Ketua Jurusan Ilmu Komputer Universitas Lampung.
7. Bapak Febi Eka Febriansyah., M.T., sebagai Pembimbing Akademik selama penulis menjadi Mahasiswa Jurusan Ilmu Komputer.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu selama penulis menjadi mahasiswa.

9. Ibu Ade Nora Maela, Kak Zainuddin, dan Mas Ardi Novalia yang telah membantu segala urusan administrasi di Jurusan Ilmu Komputer.
10. Muhammad Iqbal Abdu Salam yang telah menjadi tempat berkeluh kesah dan menemani penulis saat suka dan duka dalam proses penyusunan skripsi.
11. Teman satu kelas “Classic A” yang tercinta.
12. Keluarga besar Ilmu Komputer 2015 yang tidak bisa disebutkan satu per satu.
13. Keluarga Himakom yang telah menjadi rekan organisasi kampus.
14. Rekan-rekan Asisten Dosen yang telah memberikan pembelajaran selama menjadi Asisten Dosen.
15. Desi Yanti, Lili Adiningsih, dan Yohanes Riyanto selaku sahabat yang telah memberikan bantuan, semangat, motivasi dan dukungan selama masa kuliah hingga saat ini.
16. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak dapat disebutkan satu per satu.
17. Almamater Tercinta Universitas Lampung.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, semoga skripsi ini memberikan manfaat bagi semua keluarga Jurusan Ilmu Komputer Universitas Lampung.

Bandar Lampung, 27 Januari 2022



Desti Fatmalasari

NPM. 1517051037

DAFTAR ISI

	Halaman
DAFTAR ISI	xv
DAFTAR TABEL	xvi
DAFTAR GAMBAR	xvii
I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Ruang Lingkup.....	4
II. TINJAUAN PUSTAKA	5
2.1 Peneliti Terdahulu	5
2.2 Peringkasan Teks	7
2.3 Pembobotan Kalimat.....	7
2.4 Pembobotan Menggunakan <i>Similarity</i>	9
2.5 <i>User Acceptance Test (UAT)</i>	15
2.6 <i>Word Embedding</i>	19
III. METODE PENELITIAN	23
3.1. Tempat dan Waktu	23
3.2. Bahan dan Alat Penelitian.....	25
3.3. Metode Implementasi.....	26
V. KESIMPULAN DAN SARAN	52
5.1. Kesimpulan	52
5.2. Saran	53
DAFTAR PUSTAKA	54

DAFTAR TABEL

Tabel	Halaman
1 Penelitian Terdahulu	5
2 Gant Chart Rincian Kegiatan Penelitian	24
3 Ilustri Tahapan Parsing Kalimat.....	27
4 Ilustrasi Tahapan Case Folding.....	28
5 Ilustrasi Tahapan Tokenizing.....	29
6 Contoh Konten dari Tiga Dokumen.....	30
7 Perhitungan matriks TF-IDF.....	30
8 Pertanyaan Kuesioner Aplikasi Peringkasan Teks.....	36

DAFTAR GAMBAR

Gambar	Halaman
1 Arsitektur <i>word2vec</i> . <i>CBOW</i> (kiri) dan <i>Skip-gram</i> (kanan).....	20
2 Ilustrasi <i>window</i> pada <i>word2vec</i>	21
3 Diagram Alir Tahapan Peringkasan Dokumen	26

I PENDAHULUAN

1.1 Latar Belakang

Membaca merupakan alat untuk menjelajahi dunia sehingga membaca sudah menjadi salah satu kebutuhan bagi setiap manusia. Budaya membaca sudah seharusnya ditanamkan semenjak kecil untuk memperluas wawasan sehingga mempermudah manusia dalam mencapai kesuksesannya. Bahkan dalam al-quran diwajibkan kepada umat manusia untuk membaca yang tercantum dalam surat Al-Alaq ayat satu sampai ayat lima sebagai wahyu pertama yang diterima oleh nabi Muhammad Sallahu Alaihi Wassalam. Membaca mampu membawa perubahan positif dalam kehidupan setiap manusia. Dalam kegiatan sehari-hari pun kita sering membaca sesuatu dari media cetak maupun media digital untuk mendapatkan informasi-informasi yang dibutuhkan. Informasi yang disajikan dalam media cetak berupa buku, surat kabar, dan majalah. Sedangkan informasi yang disajikan dalam media digital berupa artikel-artikel ilmiah *online* yang dilihat melalui *web*.

Artikel-artikel ilmiah sangatlah membantu dalam pengerjaan tugas. Banyaknya artikel ilmiah yang dipublikasikan dengan berbagai topik mengharuskan pembaca untuk membaca semua artikel yang diinginkan. Artikel ilmiah yang biasa dipakai dalam penyelesaian tugas adalah jurnal. Jumlah jurnal yang tercatat dalam *Indonesian Scientific Journal Database* (ISJD) di tahun 2017 yang terupload sebanyak 23.980 jurnal dengan jumlah terbaca sebanyak 23.918 jurnal dan 23.912 jurnal yang terunduh. Pada tahun 2018 jumlah jurnal yang terupload mengalami kenaikan sebanyak 34.551 jurnal dengan jumlah jurnal yang terbaca sebanyak 34.23 jurnal dan jurnal yang terunduh sebanyak 34.215 jurnal. Adanya kenaikan yang signifikan tersebut menunjukkan bahwa minat baca dan kebutuhan pada sumber pustaka yang berasal dari jurnal tinggi, namun padatnya kegiatan atau kesibukan menghambat pembaca untuk

membaca keseluruhan isi dari artikel ilmiah yang memiliki banyak kata dan kalimat karena waktu luang yang sedikit. Selain itu membaca artikel ilmiah ataupun jurnal, kemudian melakukan kegiatan meringkas atau mengambil intisari dari suatu artikel secara manual terkadang masih belum efisien. Kegiatan meringkas artikel dilakukan apabila pembaca telah membaca keseluruhan isi dari suatu artikel sehingga dapat mengetahui hal-hal yang penting dalam artikel tersebut. Meringkas suatu artikel perlu dilakukan berulang-ulang kali dalam membaca artikel yang ingin diringkas sehingga membutuhkan waktu yang lama.

Konsep sederhana yang dilakukan dalam meringkas suatu teks dalam artikel yaitu dengan mengambil bagian-bagian yang dianggap penting dari keseluruhan isi suatu artikel.

Banyaknya teknologi tentang peringkasan dokumen digital saat ini memiliki banyak manfaat bagi para pembaca yang memiliki aktivitas maupun kesibukan yang padat. Salah satu manfaat yang dihasilkan adalah efisiensi waktu dalam membaca dan menggali informasi-informasi penting yang dibutuhkan. Peringkasan dokumen yang baik akan mempermudah seseorang dalam mengambil kebijakan yang cepat dan tepat. Setiap orang yang tidak memiliki banyak waktu namun dituntut untuk selalu memperoleh informasi penting membutuhkan solusi nyata.

Teknik yang dilakukan dalam meringkas suatu artikel ilmiah, yaitu dengan membuang paragraf suatu teks yang berisi deskripsi ataupun kutipan-kutipan dalam artikel. Teknik selanjutnya, yaitu dengan mengubah dialog menjadi kata yang mudah untuk dipahami, lalu mengurutkan susunan teks sesuai dengan artikel aslinya.

Peneliti ingin membangun sebuah aplikasi peringkasan menggunakan pendekatan metode pembobotan kalimat *Terms Frequency-Inverse Document Frequency* (TF-IDF) dan Similarity untuk memeriksa suatu kesamaan teks pada dokumen yang sudah ada. Menurut Ramos (2008) TF-IDF dapat menentukan frekuensi relatif pada kata dalam dokumen dan membandingkan dengan proporsi kata dari seluruh dokumen. Seperti pada penelitian yang dilakukan oleh Ardhy, Y Wardata (2015) yang membangun aplikasi peringkasan teks pada halaman web

berita berdasarkan kandungan intisari atau *content* berita dengan menggunakan metode pendekatan yang hampir sama dengan TF-IDF, yaitu dengan metode *Tanimoto Distance Jaccard Similarity* dan pembobotan Frekuensi kata. dari hasil uji coba yang dilakukan didapatkan bahwa peringkasan menghasilkan ringkasan dengan tingkat akurasi *f-measure* sebesar 55% untuk berita pada teks Bahasa Indonesia dan 63% untuk teks berita yang Bahasa Inggris. Proses peringkasan yang dilakukan, yaitu pada paragraf pertama pada setiap konten akan dipisahkan berdasarkan tanda baca dan koleksi *stopword* yang ada. Pada paragraf yang memiliki nilai *similarity* paling besar dalam sebagian kalimat akan dianggap kalimat penting dan masuk kedalam ringkasan akhir. Penambahan proses pembobotan kalimat sangat membantu proses peringkasan pada aplikasi karena, jika hanya menggunakan *similarity*, maka hasil ringkasan yang didapatkan masih kurang baik. Meskipun dengan pembobotan kalimat hasil ringkasan relatif sedikit tetapi dalam pengukuran nilai akurasi *f-measure* masih relatif jauh. Oleh karena, itu penulis tertarik untuk membangun sebuah aplikasi sebagai bahan pembuatan proposal penelitian dengan judul “*Peringkasan Teks Otomatis Menggunakan Metode Pembobotan Kalimat*) pada *Artikel Ilmiah Bahasa Indonesia*”. Hasil aplikasi yang akan dibuat diharapkan dapat mempermudah pembaca dalam memahami bacaan artikel ilmiah dari hasil artikel yang sudah diringkaskan oleh aplikasi.

1.2 Rumusan Masalah

Permasalahan yang dapat dirumuskan dari penelitian ini adalah bagaimana mengimplementasikan metode TF-IDF dan Similarity pada aplikasi Peringkasan teks otomatis menggunakan pembobotan kalimat pada artikel ilmiah Bahasa Indonesia.

1.3 Tujuan Penelitian

Adapun tujuan penelitian yang didapat dalam meringkas teks otomatis menggunakan metode pembobotan kalimat pada artikel ilmiah Bahasa Indonesia adalah sebagai berikut:

1. Mengetahui kinerja meringkas suatu artikel Bahasa Indonesia dalam peringkasan teks otomatis menggunakan metode pembobotan kalimat pada artikel ilmiah Bahasa Indonesia.
2. Mengimplementasikan metode pembobotan kalimat pada sistem peringkasan teks Bahasa Indonesia.

1.4 Manfaat Penelitian

Adapun manfaat penelitian dalam meringkas teks otomatis menggunakan metode pembobotan kalimat pada artikel ilmiah Bahasa Indonesia adalah sebagai berikut:

1. Menyediakan sistem berbasis *web* peringkasan teks otomatis menggunakan metode pembobotan kalimat pada artikel ilmiah Bahasa Indonesia.
2. Memperoleh hasil ringkasan dengan akurasi yang baik dalam waktu yang singkat.

1.5 Ruang Lingkup

Adapun ruang lingkup bahasan yang akan dibahas dalam peringkasan teks otomatis yaitu proses peringkasan dilakukan pada dokumen tunggal berbahasa Indonesia dengan jumlah 90 jurnal komputasi yang telah dihapus bagian abstrak, bahasa pemrograman menggunakan bahasa *php*. Sistem peringkasan teks otomatis memiliki kelemahan, yaitu sistem tidak dapat membaca teks berbahasa Inggris sehingga hasil ringkasan yang dihasilkan belum maksimal. Sistem peringkasan hanya dapat meringkas jurnal komputasi yang telah disediakan oleh sistem..

II. TINJAUAN PUSTAKA

2.1 Peneliti Terdahulu

Agar penelitian ini dipertanggungjawabkan secara akademis, maka peneliti akan menampilkan penelitian yang telah dilakukan oleh peneliti sebelumnya. Berikut merupakan tabel penelitian terdahulu yang telah diteliti.

Table 1 Penelitian Terdahulu

Tahun	Nama	Judul	Metode	Data	Hasil
2018	Annisa Nur Fadhilah	Peringkasan Teks Bahasa Indonesia Otomatis Menggunakan Metode <i>Feedforward Neural Network</i>	<i>FFNN (Feedforward Neural Network)</i> merupakan tipe jaringan syaraf tiruan paling sederhana. Koneksi dari <i>output</i> suatu layer <i>FFNN</i> akan langsung menuju input layer selanjutnya.	100 dokumen berita nasional Kompas	<ul style="list-style-type: none"> • Hasil akurasi 90.9% dengan penempatan 30% • Akurasi 76.1% dengan penempatan 20% • Akurasi 49.1% dengan penempatan 10%.
2018	Laitus Hermawan	Peringkasan Proposal Skripsi Menggunakan Algoritma <i>Vector Space Model</i>	Algoritma <i>Vector Space Model</i> untuk mewakili dokumen teks sebagai suatu vektor pengenalan.	5 Dokumen Proposal Mahasiswa Informatika	<ul style="list-style-type: none"> • Kemampuan tingkat ringkasan dari seluruh data adalah 31% • Memotong 69% kata yang tidak memiliki bobot yang besar terhadap kalimat lain.
2018	Abdurrohman	Evaluasi Algoritma <i>Textrank</i> Peringkasan Teks Berbahasa Indonesia	Algoritma <i>Textrank</i> berbasis graf bekerja dengan cara memberikan peringkat teks dengan cara merepresentasikan objek dalam teks.	100 Berita online dari liputan6.com , kompas.com	<ul style="list-style-type: none"> • Hasil <i>F-Score</i> pada <i>OUGE-1</i> dan 0.3186 • Hasil <i>F-Score</i> pada <i>ROUGE-2</i>. Hasil yang

			Setiap kalimat juga akan dihubungkan berdasarkan kesamaan (<i>similarity</i>) antar kalimat.	s.com, detik.com	diperoleh cukup baik.
2017	I Putu Gede Hendra Saputra	Peringkasan Teks Otomatis Untuk Dokumen Bahasa Bali Berbasis Metode Ekstraktif	Metode Ekstraktif dengan mengekstraksi kalimat-kalimat ringkasan berdasarkan skor fitur-fitur penting dari kalimat yang dimiliki.	Artikel Bahasa Bali	<ul style="list-style-type: none"> • Hasil evaluasi nilai rata-rata <i>ROUGE-1</i> yang terbaik adalah 0.52 • Kombinasi bobot fitur $f_1 = 0.3$, bobot $f_2 = 0.5$ dan bobot $f_3 = 0.2$. • Fitur f_2 adalah fitur yang memberikan pengaruh yang dominan dalam peringkasan dokumen Bahasa Bali.
2016	Devid Haryalesmana Wahid dan Azhari SN	Peringkasan Sentimen Ekstraktif di <i>Twitter</i> Menggunakan <i>Hybrid TF-IDF</i> dan <i>Cosine Similarity</i> .	<i>SentiStrength</i> , <i>Hybrid TF-IDF</i> , dan <i>Cosine Similarity</i> . Metode <i>Sentistrength</i> digunakan untuk mendapatkan kekuatan sentimen lalu mengklasifikasi <i>tweet</i> ke dalam kelas negatif, positif, dan netral. <i>Tweet</i> bersentimen positif dan negatif diringkas dengan cara pemeringkatan <i>tweet</i> menggunakan <i>Hybird TF-IDF</i> yang dikombinasi dengan skor kekuatan sentimen, kemudian menghilangkan <i>tweet</i>	<i>Tweet</i> Berbahasa Indonesia	<ul style="list-style-type: none"> • Hasil pengujian akurasi yang lebih baik dengan rata-rata sebesar 60% • Hasil <i>f-measure</i> yang didapat sebesar 62%.

yang mirip
menggunakan
Cosine Similarity.

2.2 Peringkasan Teks

Peringkasan teks merupakan proses penyuntingan informasi teks yang berasal dari sumber teks (dokumen/artikel) kedalam bentuk yang lebih singkat dan efektif (Trisaputra et al., 2016). Proses pengurangan teks dalam dokumen atau artikel dilakukan dengan menggunakan dua macam metode atau kriteria. Metode yang digunakan antara lain yaitu metode abstraksi dan metode ekstraksi. Metode abstraksi merupakan metode yang mengambil intisari yang berasal dari sumber teks (dokumen) sehingga didapatkan suatu ringkasan dengan kalimat-kalimat yang baru dengan interpretasi berbeda dengan sumber teks (dokumen) awal. Metode ekstraksi merupakan teknik memilih kata dalam kalimat yang paling informatif dari sumber teks (dokumen) menjadi ringkasan (Aristoteles, 2011).

2.3 Pembobotan Kalimat

Pembobotan kalimat merupakan proses memberikan nilai pada kalimat yang diukur melalui kedudukan kalimat yang mampu mencangkup topik dan terhindar dari redundansi terhadap kalimat-kalimat yang lain, baik dalam dokumen yang tunggal maupun multi-dokumen. Metode pembobotan kalimat banyak dikembangkan dan digunakan oleh para peneliti dalam bidang peringkasan dokumen, pencarian dokumen, analisis dokumen berita, dan lain sebagainya. Tujuan dalam pembobotan kalimat yaitu untuk memperoleh kalimat penting yang berbeda dalam dokumen. Kalimat penting diperoleh dari ekstraksi dokumen dengan menghitung bobot kalimat berdasarkan frekuensi kemunculan kata. Kalimat penting dari suatu dokumen harus memiliki kepadatan informasi sehingga mampu mewakili dokumen sumber. Suatu kalimat dapat menjadi penting apabila kata-kata yang menyusun kalimat merupakan kata penting. Kata penting merupakan kata-kata yang

tersebar pada setiap bagian dalam sebuah dokumen (Wahib dan Winoto, 2017).

Metode pembobotan kalimat secara umum dikelompokkan menjadi tiga kategori yaitu berdasarkan bobot kata, fitur kalimat, dan relasi antar kalimat. Pembobotan kalimat dengan menggunakan dua macam tipe yaitu pembobotan berdasarkan kata dan kalimat. Pembobotan kalimat berdasarkan kata biasanya menggunakan *TF-IDF* dan *Word Frequency (WF)*. Sedangkan pembobotan kalimat dengan kalimat dengan mengecek posisi, *centroid*, panjang kalimat, kemiripan kalimat terhadap kalimat pertama pada dokumen, dan kemiripan kalimat terhadap judul dokumen (Hayatin et al., 2015).

2.3.1 Pembobotan Kalimat Menggunakan Algoritma TF-IDF

Pembobotan kalimat menggunakan pembobotan *TF-IDF (Term Frequency and Inverse Document Frequency)*. Pembobotan menggunakan metode ini dapat diperoleh berdasarkan jumlah kemunculan suatu *term* dalam sebuah dokumen *term frequency (tf)* dan jumlah kemunculan *term* dalam koleksi dokumen *inverse document frequency (idf)*. Bobot suatu istilah semakin besar jika istilah tersebut sering muncul dalam suatu dokumen dan semakin kecil jika istilah tersebut muncul dalam banyak dokumen. Nilai *IDF* sebuah *term* (kata) dapat dihitung menggunakan Persamaan (1):

$$IDF = \log\left(\frac{X}{x_{fi}}\right) \dots \dots \dots (1)$$

X merupakan jumlah dokumen yang berisi *term (t)* dan x_{fi} adalah jumlah kemunculan (frekuensi) *term* terhadap X . Adapun algoritma yang digunakan untuk menghitung bobot (W) masing-masing dokumen terhadap kata kunci (*query*), yaitu seperti pada Persamaan (2):

$$W_{x,t} = tf_{x,t}IDF.....(2)$$

Keterangan :

x = dokumen ke- x

t = *term* ke- t dari kata kunci

tf = *term* frekuensi/frekuensi kata

W = bobot dokumen ke- x terhadap *term* ke- t

Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses pengurutan (*sorting*) dimana semakin besar nilai W , semakin besar tingkat kesamaan (*similarity*) dokumen tersebut terhadap kata yang dicari, demikian pula sebaliknya (Mustaqhfiri et al., 2012).

2.4 Pembobotan Menggunakan *Similarity*

Pembobotan kalimat menggunakan perhitungan nilai *similarity* yang sederhana dan lebih cepat. Algoritma yang digunakan sebagai berikut:

- *Input* : *list distinct term* setiap kalimat dan *list term* judul
- *Output* : nilai bobot untuk setiap kalimat
- Langkah 1 : pada langkah pertama setiap kalimat akan dihitung frekuensi dan dinormalisasi frekuensinya sesuai langkah 2-3
- Langkah 2 : menghitung frekuensi *term* pada kalimat
- Langkah 3 : menormalisasi frekuensi *term* pada kalimat
- Langkah 4 : menghitung frekuensi *term* judul
- Langkah 5 : menormalisasi frekuensi *term* judul
- Langkah 6 : pada langkah ke enam setiap kalimat akan dihitung *similarity* antara judul dengan kalimat seperti langkah 7
- Langkah 7 : menghitung *similarity* antara judul dengan kalimat

Hasil nilai *similarity* antara judul dan kalimat digunakan sebagai bobot kalimat. Perhitungan *similarity* dapat dilihat pada Persamaan (3):

$$Sim(j,k) = \sum_{vi \in IJ} \left(\frac{WJi}{NK} + \frac{WKi}{NK} \right), \text{ jika } i \in LK.....(3)$$

Dengan j adalah judul, k adalah kalimat, LJ adalah *list term* judul, LK adalah *list i term* kalimat, NK adalah total kalimat teks, WJ_i adalah nilai normalisasi frekuensi *term* ke- i pada *list term* judul, dan WK_i adalah nilai normalisasi frekuensi *term i* pada *list term* kalimat. Semakin nilai *similarity* besar semakin penting kalimat tersebut (Trisaputra et al., 2016).

2.4.1 Parsing Kalimat

Parsing merupakan cara atau metode untuk memecah suatu rangkaian masukkan. *Parsing* kalimat dilakukan untuk memisahkan kalimat berdasarkan tanda baca titik (.). Pemisahan kata dalam dokumen digunakan untuk mengambil kata atau *term* dalam kalimat yang sudah dipecah. Hasil yang didapat dari pemisahan kalimat ini berupa list kalimat yang akan digunakan untuk pengambilan *term* atau kata (Trisaputra et al., 2016).

2.4.2 Pengambilan Term

Pengambilan *term* dilakukan pada setiap kalimat hasil dari *parsing* kalimat. Dalam metode pengambilan *term* dilakukan beberapa tahap yaitu tahap *case folding*, *tokenizing*, *filtering*, *stemming* (Trisaputra et al., 2016).

1. *Case folding*

Case folding merupakan tahapan mengubah semua huruf dalam teks dokumen menjadi huruf kecil dan menghilangkan karakter selain karakter a sampai z (Mustaqhfiri et al., 2012).

2. *Tokenizing*

Tokenizing merupakan proses pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Pemecahan kalimat menjadi kata-kata tunggal dilakukan dengan

menelusuri kalimat dengan pemisah spasi, *tab*, dan *newline* (Mustaqhfiri et al.,2012).

3. *Filtering*

Filtering merupakan proses penghilangan *stopword*. *Stopword* adalah kata-kata yang sering kali muncul dalam dokumen namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Didalam Bahasa Indonesia *stopword* dapat disebut sebagai kata tidak penting, misalnya “di”, ”oleh”, “pada”, ”sebuah”, ”karena” dan lain sebagainya. Contoh satu: bukukah akan menjadi kata buku, terimalah akan menjadi kata terima, apakah akan menjadi kata apa, bukupun akan menjadi kata buku karena imbuhan dalam partikel infleksional dihilangkan. Contoh 2, yaitu: kata bukuku akan menjadi kata buku, kata bukumu akan menjadi kata buku, kata bukunya akan menjadi kata buku karena pada kata ganti milik infleksionalnya akan dihilangkan (Mustaqhfiri et al.,2012).

4. *Stemming*

Stemming merupakan proses mencari akar (*root*) kata dari tiap token kata dengan pengembalian suatu kata berimbuhan ke bentuk dasarnya (*stem*). Pada penelitian ini menggunakan porter *stemming* untuk bahasa. Terdapat lima aturan pada proses *stemming* untuk Bahasa Indonesia menggunakan porter *stemmer*, yaitu ada lima aturan tahap dalam proses *stemming* pada Bahasa Indonesia, yaitu :

I.Penanganan terhadap partikel infleksional, yaitu: lah, tah, kah. Contoh: duduklah, makanlah..

II.Penanganan terhadap kata ganti infleksional, yaitu: ku, mu, dan nya.

Contoh: sepedaku, bukunya.

III. Penanganan terhadap *prefiks* derivasional pertama, yaitu : meng dan semua variasinya, peng dan semua variasinya, di, ter, dan ke. Contoh: membakar, pengukur, kekasih.

IV. Penanganan terhadap *prefiks* derivasional kedua, yaitu: ber dan semua variasinya serta per dan semua variasinya.
Contoh: berlari, belajar, perkata.

V. Penanganan terhadap *Sufiks* derivasional, yaitu kan, am dan i. Contoh: ambilkan, janji dan dapati. Karena struktur morfologi Bahasa Indonesia yang rumit maka kelima tahap aturan tidak cukup untuk menangani proses *stemming* Bahasa Indonesia. Kesulitan membedakan kata yang mengandung imbuhan baik *prefiks* maupun *sufiks* dari kata dasar yang salah satu suku katanya merupakan bagian imbuhan, terutama dengan kata dasar yang mempunyai suku kata lebih besar dari dua.

Contoh :

sekolah → sekolah (kata dasar, tidak dilakukan stemming)

duduklah → duduk (dilakukan proses stemming).

Menurut urutan pada tahapan penanganan kata berimbuhan terdapat beberapa kemungkinan kesulitan dalam membedakan suatu suku kata yang merupakan imbuhan atau bagian kata dasar:

VI. Kata dasar mempunyai suku kata terakhir (partikel infleksional) serta kata tersebut tidak mendapat imbuhan apapun.

Contoh: istilah.

VII. Kata dasar mempunyai suku kata terakhir (partikel infleksional) dan mempunyai *prefiks*.

Contoh: bersalah.

VIII. Kata dasar mempunyai suku kata terakhir (kata ganti infleksional) serta kata dasar tersebut tidak mendapatkan imbuhan apapun. Contoh: bangku.

IX. Kata dasar mempunyai suku kata terakhir (kata ganti infleksional) dan mengandung *prefiks*.

Contoh: bertanya.

X. Kata dasar mempunyai suku kata pertama (*prefiks* derivasional pertama) serta kata dasar tersebut tidak mendapatkan imbuhan apapun.

Contoh: diagram, kenang.

XI. Kata dasar mempunyai suku kata pertama (*prefiks* derivasional pertama) dan mempunyai *sufiks* derivasional.

Contoh: disiplinkan, pentungan.

XII. Kata dasar mempunyai suku kata pertama (*prefiks* derivasional kedua) serta kata dasar tersebut tidak mendapatkan imbuhan apapun.

Contoh : pelangi, perban.

XIII. Kata dasar mempunyai suku kata pertama (*prefiks* derivasional) dan mempunyai *sufiks* derivasional.

Contoh: belakangan, pejamkan.

XIV. Kata dasar mempunyai suku kata terakhir (*sufiks* derivasional). Contoh: koleksi, dominan.

Dilihat dari permasalahan-permasalahan yang ada, maka dibuat kamus-kamus kecil untuk melengkapi proses *stemming* ini. Terdapat 9 (Sembilan) kamus kecil, yaitu :

- Kamus partikel. Seperti: masalah
- Kamus partikel ber*prefiks*. Seperti: menikah
- Kamus milik. Seperti: bangku.
- Kamus milik ber*prefiks*. Seperti: bersuku.
- Kamus *prefiks* 1. Seperti: median.
- Kamus *prefiks* 1 ber*sufiks*. Seperti: terapan.
- Kamus *prefiks* 2. Seperti : percaya.

- Kamus *prefiks 2 bersufiks*. Seperti: perasaan.
- Kamus *sufiks*. Seperti: pantai.

Kondisi ukuran adalah jumlah minimum suku kata dalam sebuah kata. Dalam Bahasa Indonesia, kata dasar setidaknya mempunyai 2 suku kata. Sehingga kondisi ukuran dalam proses *stemming* Bahasa Indonesia adalah dua. Suku kata didefinisikan memiliki satu vokal (Mustaqhfiri et al, 2012).

2.4.3 Pemilihan Kalimat

Pemilihan kalimat berdasarkan hasil bobot kalimat yang dihasilkan. Hasil pemilihan kalimat dilakukan dengan beberapa tahapan berikut:

- Input : list bobot setiap kalimat
- Output : hasil ringkasan teks
- Langkah1 : menghitung rata-rata bobot dari setiap kalimat
- Langkah2 : menetapkan nilai threshold = rata-rata
- Langkah3 : setiap kalimat dilakukan langkah 4
- Langkah4 : jika bobot kalimat yang dimiliki \geq threshold, kalimat ditampilkan.

Nilai threshold ditetapkan untuk memilih kalimat yang yang ditampilkan. Threshold digunakan sebagai compression rate dari hasil peringkasan. Hasil peringkasan bergantung pada kalimat judul. Kalimat yang relevan atau berkaitan dengan judul akan ditampilkan pada hasil peringkasan.

Adapun persamaan Compression Rate seperti Persamaan (4):

$$\text{Compression rate (cr)} = \frac{nr}{nt} * 100\% \dots \dots \dots (4)$$

Keterangan:

nt = banyak kalimat pada teks

nr = banyak kalimat pada hasil ringkasan (Trisaputra et al., 2016).

2.5 User Acceptance Test (UAT)

Menurut Perry, William E, *User Acceptance Testing* (UAT) merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah staf/karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya. Setelah dilakukan sistem *testing*, *acceptance testing* menyatakan bahwa sistem perangkat lunak memenuhi persyaratan.

Acceptance testing merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *blackbox* untuk menguji sistem terhadap spesifikasinya. Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji.

1. Jenis-jenis *User Acceptance Test* (UAT)

a. *Alpha Testing*

Pengujian *Alpha* adalah pengujian akhir sebelum perangkat lunak diluncurkan untuk pengguna secara umum. *Alpha test* memiliki dua fase:

- 1) Pada tahap pertama dari pengujian *alpha*, perangkat lunak diuji oleh pengembang di lingkungan internal *developer*. Mereka menggunakan perangkat lunak *debugger*, atau *debugger hardware-assisted*. Tujuannya adalah untuk menangkap bug dengan cepat.
- 2) Pada tahap kedua pengujian *alpha*, perangkat lunak ini diserahkan kepada staf QA (*Quality Assurance*) perangkat lunak, untuk pengujian tambahan dalam lingkungan yang mirip dengan pengguna yang dimaksudkan. Hal ini untuk mensimulasikan suasana atau lingkungan pengujian yang sebenarnya sehingga ketika sistem tersebut dipasang, sudah tidak terjadi kegagalan maupun cacat sistem secara *real*.

b. *Beta Test*

Pengujian *beta* juga dikenal sebagai pengujian pengguna berlangsung di lokasi pengguna akhir (*end-user*) untuk memvalidasi kegunaan, fungsi, kompatibilitas, dan uji reliabilitas dari perangkat lunak yang dibuat. Tes *beta* merupakan tahap kedua dari pengujian perangkat lunak di mana pengguna mencoba produk.

Dengan demikian pengujian *Alfa* dilakukan di tempat karena pengembang dan analis bisnis terlibat dengan tim penguji. Sedangkan pengujian *Beta* dilakukan di sisi klien oleh pengguna atau pelanggan nyata, oleh karena itu pengembang dan analis bisnis sama sekali tidak terlibat.

2. Proses *User Acceptance Test* (UAT)

Proses dalam UAT adalah pemeriksaan dan pengujian terhadap hasil perangkat lunak yang di buat. Diperiksa apakah *item-item* yang ada dalam dokumen *requirement* sudah ada dalam perangkat lunak yang diuji atau tidak.

Diuji apakah semua *item* yang telah ada telah dapat memenuhi kebutuhan penggunanya. Pengujian ini biasanya terjadi di lokasi klien yang dikenal sebagai pengujian *beta*. Setelah memenuhi kriteria masuk untuk UAT dipenuhi, berikut adalah tugas yang perlu dilakukan oleh penguji:

a. Menyusun rencana UAT

untuk memverifikasi dan memastikan aplikasi memenuhi persyaratan bisnisnya. Menyusun dokumen masuk dan keluar kriteria untuk UAT, test *scenarios* dan uji kasus pendekatan dan jadwal pengujian, tanggal, lingkungan, aktor, peran dan tanggung jawab akan ditemukan di UAT rencana uji.

b. Desain UAT

Kriteria penerimaan yang dikumpulkan dari pengguna digunakan dalam langkah ini. Berdasarkan kriteria, tim QA (*Quality assurance*) memberi pengguna daftar kasus uji UAT.

1) Identifikasi Skenario Pengujian dan Uji Kasus: Identifikasi skenario pengujian sehubungan dengan proses yang dibuat dan diuji dengan langkah yang jelas.

2) Persiapan Data Uji: Sangat disarankan untuk menggunakan data langsung untuk UAT. Data harus diacak untuk alasan privasi dan keamanan.

c. Eksekusi Uji UAT

Jalankan kasus uji dan laporkan *bug* jika ada. Uji kembali *bug* setelah diperbaiki. Kasus Uji membantu tim untuk menguji aplikasi secara efektif di lingkungan UAT. Setelah semua tes dijalankan dan hasilnya ada di tangan, Keputusan Penerimaan dibuat. Ini juga disebut keputusan *Go / No-Go*. Jika pengguna merasa puas, itu adalah *Go*, atau ini Tidak Bergerak.

d. Konfirmasi tujuan bisnis yang dipenuhi

Analisis Bisnis atau Pengujian UAT harus mengirim tanda surat setelah pengujian UAT. Setelah *sign-off*, produk itu bagus untuk diproduksi. Hasil kerja untuk pengujian UAT adalah *test plan*, skenario UAT dan uji kasus, dan hasil uji Kriteria keluaran untuk UAT sebelum pindah ke produksi, berikut ini perlu dipertimbangkan:

- 1) Tidak ada cacat kritis yang terbuka
- 2) Proses bisnis bekerja dengan memuaskan

e. *Sign Off*

Setelah berhasil menyelesaikan pengujian dan penyelesaian masalah tim secara umum menunjukkan penerimaan aplikasi. Begitu pengguna menerima perangkat lunak, ini berarti menunjukkan bahwa perangkat lunak tersebut memenuhi persyaratan mereka.

3. Tahap Pengujian Penerimaan Pengguna (UAT)

Proses pengujian dilakukan untuk memastikan bahwa sistem yang akan dilaksanakan bekerja benar, maka berikut ini adalah tahap-tahap pengujian UAT:

- a. Merancang kasus uji: Kasus uji dirancang untuk mencakup semua skenario fungsional dari perangkat lunak dalam penggunaan di dunia nyata. Kasus dirancang dengan bahasa dan cara yang sederhana agar proses pengujian lebih mudah bagi penguji.
- b. Pemilihan tim pengujian: Tim penguji terdiri dari pengguna akhir.
- c. Melaksanakan kasus pengujian dan mendokumentasikan: Tim penguji mengeksekusi kasus uji yang ditentukan. Kadang-kadang juga menjalankan beberapa tes acak yang relevan. Semua bug dicatat dalam dokumen pengujian dengan komentar yang relevan.
- d. Perbaiki *bug*: Menanggapi bug yang ditemukan oleh tim penguji, tim pengembangan perangkat lunak membuat penyesuaian akhir pada kode untuk membuat perangkat lunak bebas *bug*.
- e. *Sign-off*: Ketika semua *bug* telah diperbaiki, tim pengujian menunjukkan penerimaan aplikasi perangkat lunak. Ini menunjukkan bahwa aplikasi tersebut memenuhi persyaratan pengguna dan siap untuk diluncurkan di pasar.

UAT penting karena membantu menunjukkan bahwa fungsi bisnis yang diperlukan beroperasi dengan cara yang sesuai dengan keadaan dan penggunaan di dunia nyata (Supriatna, 2018).

2.6 Word Embedding

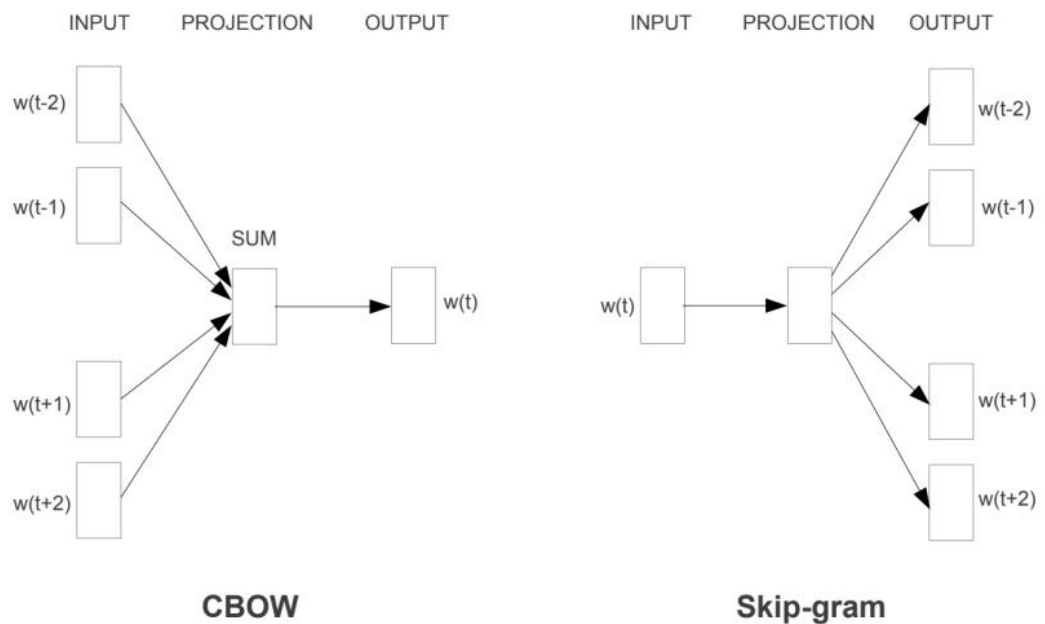
Word embeddings adalah proses konversi kata yang berupa karakter *alphanumeric* kedalam bentuk *vector*. Setiap kata adalah *vector* yang merepresentasikan sebuah titik pada *space* dengan dimensi tertentu. Dengan *word embedding*, kata-kata yang memiliki properti tertentu, misalnya berada pada konteks yang sama, atau memiliki *semantic meaning* yang sama berada tidak jauh satu sama lain pada *space* tersebut.

Word2Vec adalah model *shallow neural network* yang merubah representasi kata yang merupakan kombinasi dari karakter *alphanumeric* menjadi *vector*. Representasi *vector* tersebut memiliki properti *relationship* terhadap kata-kata yang berkaitan melalui proses *training*.

Arsitektur *neural network* yang digunakan pada *word2vec* adalah *neural network* dengan sebuah *hidden layer* yang disebut dengan *projection layer* dan di-*training* menggunakan *stochastic gradient descent* dengan algoritma *backpropagation*. *Projection layer* adalah mapping dari kata yang ada pada konteks *n-gram* ke dalam bentuk *continuous vectors*. Kata-kata yang muncul bersamaan atau berulang kali pada konteks *N-gram* memiliki tendensi untuk teraktifasi oleh *weight* yang sama, sehingga terjadi korelasi antar kata-kata.

Weights menghubungkan input layer dengan *hidden layer*, dan *hidden layer* dengan output layer. *Weights* diantara input layer dan *hidden layer* direpresentasikan dengan matriks \mathbf{W} berukuran $V \times N$, dimana V adalah dimensi dari input layer dan N adalah dimensi dari *hidden layer*. Sedangkan diantara *hidden layer* dan output layer, matriks \mathbf{W} direpresentasikan dengan matriks berukuran $N \times V$.

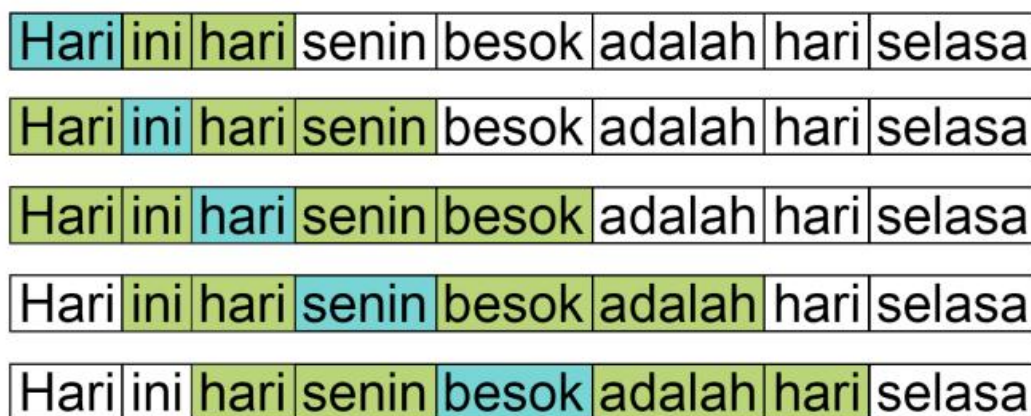
Terdapat dua model arsitektur yang dapat digunakan pada *word2vec*, yaitu *CBOW* dan *Skip-Gram*. Kedua model tersebut dapat dilihat pada gambar 1 berikut.



Gambar 1 Arsitektur *word2vec*. *CBOW* (kiri) dan *Skip-gram* (kanan)

Pada model *CBOW*, *word2vec* menggunakan kata-kata yang ada di sebelah kiri dan kanan kata target dan dibatasi dengan *window* untuk memprediksi kata target tersebut. Sedangkan *skip-gram* menggunakan sebuah kata untuk memprediksi kata-kata yang ada di sebelah kiri dan kanan kata tersebut yang dibatasi oleh *window*. Masing-masing kata yang digunakan sebagai input di-*encode* ke dalam *one-hot vector*. Perbedaan dari kedua model tersebut adalah model prediksi kata. Pada *CBOW*, terdapat intermediate layer yang akan melakukan kalkulasi average pada *vector* kata-kata input karena *CBOW* menerima sejumlah n kata sebagai input.

Window digunakan sebagai kernel untuk memperoleh input dan target *words*. *Window* digeser dari awal sampai akhir susunan kata. Ilustrasi dari *window* dapat dilihat pada gambar 2 sebagai berikut.



Gambar 2 Ilustrasi *window* pada *word2vec*

Pada gambar 2, ukuran dari *window* adalah 2. Kolom berwarna biru adalah *center word*, sedangkan kolom berwarna hijau adalah *context words*. Pada *CBOV*, *center word* digunakan sebagai target, dan *context words* digunakan sebagai input pada setiap iterasi. Sedangkan pada *skip gram*, *center word* digunakan sebagai target, dan *context words* digunakan sebagai target pada setiap iterasi.

Word Embedding digunakan untuk representasi makna yang lebih baik karena adanya keterbatasan informasi pada teks pendek (Wang & Liu, 2018), maka dari itu *Word2vec word embedding* model dengan algoritma *Skip-gram* dapat digunakan untuk mempelajari relasi kontekstual antar kata yang ada pada data training. Model *Word2vec* dengan algoritma *Skip-gram* melakukan proses training dengan cara memprediksi kata sebelumnya dan juga kata sesudahnya berdasarkan kata saat ini (Mikolov, et al., 2013).

Word embedding adalah sebuah fungsi parameter yang memetakan setiap kata ke dalam vektor berdimensi tinggi. Keunggulan *word embedding* tidak membutuhkan anotasi, dapat langsung diturunkan dari korpus tak teranotasi. *Word embedding* dapat dibuat langsung dari dataset yang dimiliki atau menggunakan *pre-trained word embedding* yang telah tersedia. *Pre-trained word embedding* ini adalah *word embedding* yang telah dilatih menggunakan dataset yang besar pada domain permasalahan tertentu yang dapat

digunakan untuk menyelesaikan permasalahan lain yang serupa (Nurdin, et al., 2020).

III. METODE PENELITIAN

3.1. Tempat dan Waktu

Tempat Penelitian ini dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

Waktu penelitian dilakukan pada semester ganjil tahun ajaran 2020/2021. Adapun rincian kegiatan yang akan dilakukan selama penelitian diuraikan pada Tabel 1.

3.2. Bahan dan Alat Penelitian

Bahan yang digunakan dalam penelitian adalah jurnal komputasi yang diambil dari web Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung dengan situs alamat <http://fmipa.unila.ac.id>. Dokumen yang digunakan adalah dokumen berbahasa Indonesia dan berekstensi *.txt*. Jumlah dokumen yang digunakan sebanyak 90 dokumen.

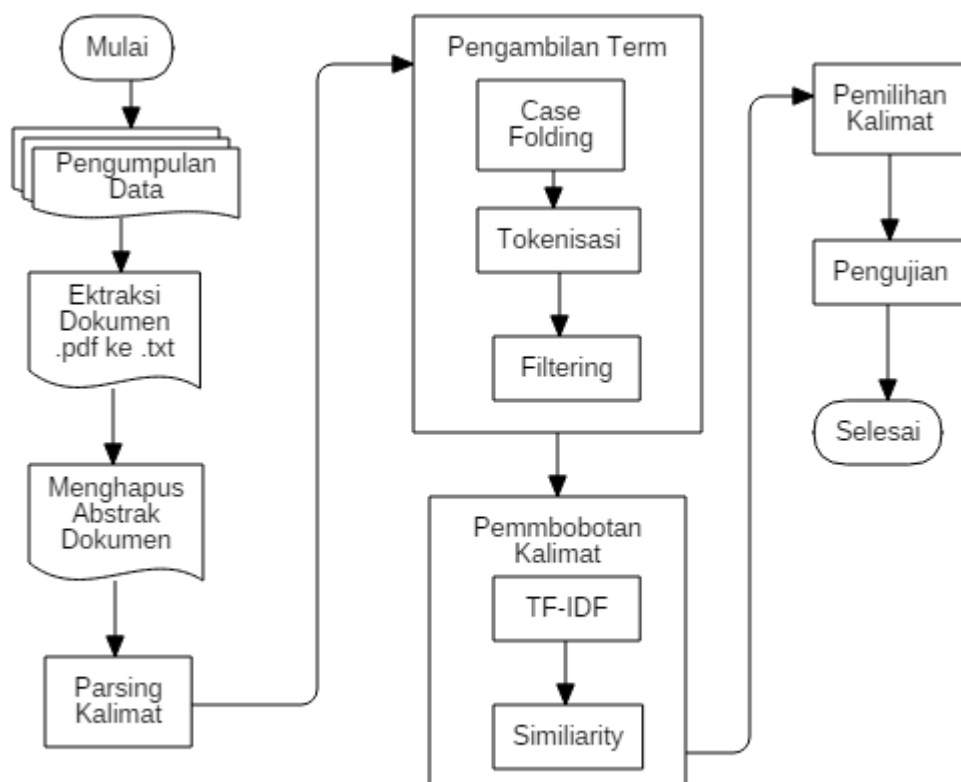
Alat Penelitian dilakukan dengan menggunakan perangkat keras dan perangkat lunak yang berguna untuk mendukung dan menunjang pelaksanaan penelitian.

- I. Perangkat keras : *Intel® Core™ i5-7200U 2.5 GHz with Turbo Boost up to 3.1 GHz, RAM 4.00 GB.*
- II. Perangkat lunak :
 1. *OS Windows 10 Professional,*
 2. *XAMPP Control Panel v3.2.2*
 XAMPP merupakan software yang terdiri dari kumpulan paket program yang berhubungan dengan database server, web server, dan lain-lain. XAMPP terdapat Apache sebagai server web, MySQL sebagai server basis data, Filezilla sebagai FTP server, dan beberapa fitur tambahan seperti Mercury dan Tomcat (Palevi dan Krisnawati, 2013).
 3. *Notepad++.*
 Notepad++ merupakan aplikasi yang menitikberatkan proses editing *text* dalam waktu yang cepat dan praktis. Notepad++ mendukung berbagai format bahasa pemrograman seperti *PHP*, *HTML*, *JavaScript*, dan *CSS* (Palevi dan Krisnawati, 2013).
 4. *PHP*
 PHP merupakan singkatan dari *Hypertext Preprocessor* yang termasuk dalam bahasa pemrograman *open source* yang sangat cocok dalam mengembangkan *web*. PHP memiliki kode-kode (*script*) yang digunakan untuk melakukan proses pengolahan suatu data dan mengirimkannya kembali pada *web browser* menjadi kode *HTML*. Data dalam bahasa *PHP* di proses menggunakan *server*

untuk menerjemahkan *script* program lalu dikirim hasilnya dikirimkan pada *client* yang melakukan permintaan (Firman, et al., 2016).

3.3. Metode Implementasi

Proses peringkasan teks otomatis menggunakan metode pembobotan kalimat pada artikel ilmiah Bahasa Indonesia digambarkan dalam diagram alir tahapan peringkasan dokumen yang dapat dilihat pada gambar 3.1.



Gambar 3 Diagram Alir Tahapan Peringkasan Dokumen

3.3.1 Pengumpulan Data

Data yang dikumpulkan berupa file dokumen dengan ekstensi *.pdf* berupa jurnal komputasi yang didapat dari *website* Fakultas Matematika

dan Ilmu Pengetahuan Alam Universitas Lampung dengan jumlah 90 dokumen yang telah dihilangkan abstrak dari setiap jurnalnya.

3.3.2 Ekstraksi File ke *.txt*

Ekstraksi file digunakan untuk mengubah ekstensi file menjadi *.txt* agar lebih mudah untuk dilakukan proses peringkasan dokumen berbahasa Indonesia.

3.3.3 Menghapus Abstrak Dokumen

Penghapusan abstrak pada jurnal komputasi yang menggunakan Bahasa Inggris agar mempermudah proses peringkasan dokumen.

3.3.4 Parsing *Kalimat*

Parsing kalimat dalam penelitian ini dilakukan untuk memisahkan kalimat berdasarkan tanda baca titik (.). Hasil yang didapat dari *parsing* kalimat berupa *list* kalimat yang akan digunakan untuk pengambilan *term* atau kata.

Table 3 Ilustri Tahapan Parsing Kalimat

	Data Uji		Tahapan <i>Parsing</i> Kalimat
Input	Untuk pengenalan citranya digunakan ciri warna pada belimbing dengan kombinasi warna Red, Green, dan Blue. Dari ekstraksi warna ini, diambil 5 nilai ciri yaitu jumlah R-jumlah Green-jumlah Blue-Mean-Standar Deviasi. Nanti	Output	- Untuk pengenalan citranya digunakan ciri warna pada belimbing dengan kombinasi warna Red, Green, dan Blue - Dari ekstraksi warna ini, diambil 5 nilai ciri yaitu jumlah R-jumlah Green-jumlah Blue-Mean-Standar Deviasi

nilai cirinya didapat dengan meratakan atau menjumlahkan semua piksel yang ada.	- Nanti nilai cirinya didapat dengan meratakan atau menjumlahkan semua piksel yang ada
---	--

Proses untuk memisahkan kalimat berdasarkan tanda baca titik dituliskan dalam *source code* berikut.

```
$load_file = file_get_contents("./corpus/".$filename);
$document = $load_file;
$sentence = preg_split("/[.]+/", $load_file);
$separate_sentence = $sentence;
```

Source code 1. Parsing Kalimat

3.3.5 Pengambilan *Term*

Proses pengambilan *term* dilakukan beberapa tahapan yaitu tahap case folding yang merupakan tahapan mengubah semua huruf dalam teks dokumen menjadi huruf kecil dan menghilangkan karakter a sampai z.

Table 4 Ilustrasi Tahapan Case Folding

Data Uji		Tahapan <i>Case Folding</i>	
Input	Membaca merupakan alat untuk menjelajahi.	Output	membaca merupakan alat untuk menjelajahi

- *Tokenizing* merupakan proses pemotongan string input berdasarkan tiap kata yang menyusunnya. Pemecahan kalimat menjadi kata-kata tunggal dilakukan dengan men-*scan* kalimat dengan pemisah *spasi*, *tab*, dan *newline*.

Table 5 Ilustrasi Tahapan Tokenizing

Data Uji	Tahapan <i>Tokenizing</i>
Reza tidur di kamar tidur	Reza Tidur Di Kamar Tidur
Input	Output

Proses untuk mengubah huruf (*Case Folding*) dan memisahkan kata berdasarkan tanda baca, spasi, *tab*, dan *newline* dituliskan dalam *source code* berikut.

```
<?php
$tokens = token_get_all('pendahuluan kemajuan
teknologi informasi ');
foreach ($tokens as $token) {
    if (is_array($token)) {
        echo "Line {$token[2]}: ",
token_name($token[0]), " ('{$token[1]}')",
PHP_EOL;
    }
}
?>
```

Source code 2. Pengambilan Term

- *Filtering* merupakan proses penghilangan *stopword*. *Stopword* adalah kata-kata yang sering kali muncul dalam dokumen namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Didalam Bahasa Indonesia *stopword* dapat disebut sebagai kata tidak penting, misalnya “di”, ”oleh”, “pada”, ”sebuah”, ”karena” dan lain sebagainya.

Contoh satu: bukukah akan menjadi kata buku, terimalah akan menjadi kata terima, apatah akan menjadi kata apa, bukupun akan menjadi kata buku karena imbuhan dalam partikel infleksional dihilangkan.

Contoh dua yaitu: kata bukuku akan menjadi kata buku, kata bukumu akan menjadi kata buku, kata bukunya akan menjadi kata buku karena pada kata ganti milik infleksionalnya akan dihilangkan.

Proses menghilangkan *stopword* dituliskan dalam *source code* berikut.

```
$stopwords =
file_get_contents("./resources/stopwords.txt");
$stopwords = preg_split("/[\s]+/", $stopwords);
```

Source code 3 Menghilangkan Stopword

3.3.6 Pembobotan Kalimat

Pembobotan kalimat diukur melalui kedudukan kalimat yang mampu mencangkup topik yang terhindar dari redundansi terhadap kalimat-kalimat yang lain, baik dalam dokumen yang tunggal. Pembobotan kalimat ini dilakukan dengan menghitung menggunakan TF-IDF (*Term Frequency dan Inverse Document Frequency*) dan perhitungan nilai *similarity* seperti yang sudah dijelaskan pada Bab II.

Berikut merupakan ilustrasi perhitungan *matriks* pada TF-IDF

Table 6 Contoh Konten dari Tiga Dokumen

ID	Konten
D1	Aplikasi <i>chat</i> sekarang sangat banyak digunakan oleh seruh kalangan.
D2	Sosial media semakin sering menimbulkan dampak negatif bagi penggunanya.
D3	Belajar membaca dapat meningkatkan wawasan kita.

Table 7 Perhitungan matriks TF-IDF

Kata	TF			DF	IDF	TF-IDF		
	D1	D2	D3			D1	D2	D3
Aplikasi	1	0	0	1	0.477	0.477	0	0
Dampak	0	1	0	1	0.477	0	0.477	0
Membaca	0	0	1	1	0.477	0	0	0.477

Proses untuk menghitung pembobotan kalimat dituliskan dalam *source code* berikut:

```
public function pembobotan_kata() {
    $l=0;
    $table1=array();
    $search=array();
    foreach ($this->stopword_remove as $key)
    {
        if
        (array_search(trim(strtolower($key)),
        $search)===false) {
            $dok=0;
            $table1[$l]['term']=trim(strtolower($key));
            $table1[$l]['dok']=array();
            foreach ($this->dokumen as
            $key1) {
                array_push($table1[$l]['dok'],
                substr_count(trim(strtolower($key1)),
                trim(strtolower($key))));
                ++$dok;}
        }
    }
}
```

Source code 4 Menghitung Pembobotan Menggunakan TF-IDF dan Similarity

3.3.7 Pemilihan Kalimat

Proses pemilihan kalimat dilakukan berdasarkan bobot kalimat yang dihasilkan dengan menghitung rata-rata bobot pada setiap kalimat, jika bobot kalimat lebih dari sama dengan *threshold* maka kalimat ditampilkan.

Proses untuk pemilihan kalimat dituliskan dalam *source code* berikut:

```
ksort($sorted);
$summary = "";
foreach ($sorted as $key => $value) {
    $summary = $summary.$separate_sentence[$key].".
";}
```

Source code 5 Memilih Kalimat

Ilustrasi peringkasan teks yang dilakukan dalam sistem:

- a) Teks yang digunakan teks berita berjudul “ Tahun Baru, Server WhatsApp Sempat Down”, seperti teks dibawah:

Aplikasi chat paling populer didunia WhatsApp sempat tak bisa diakses saat pergantian tahun, Jumat(1/1). Para Pengguna WhatsApp di Eropa dan Amerika sempat tak bisa mengirimkan atau menerima pesan melalui aplikasi tersebut.

Belum jelas apa yang menjadi penyebab gagal berfungsinya server milik WhatsApp, tapi menurut laporan dari situs DownDetector, gagal berfungsinya aplikasi tersebut terjadi pada pukul 4.30 GMT.

Kegagalan server WhatsApp melayani permintaan pengguna diduga terjadi karena kelebihan beban dimana pada saat itu banyak pengguna saling mengirimkan pesan setelah pergantian tahun baru. Terakhir kali WhatsApp mengungkapkan jumlah pengguna mereka, September lalu, perusahaan yang dimiliki Facebook ini mengklaim telah memiliki lebih dari 900 juta miliar di seluruh dunia.

- b) Teks dilakukan proses parsing kalimat dengan pemisahan tanda baca, hasil pemisahan kalimat sebagai berikut:

1. Aplikasi chat paling populer dunia WhatsApp sempat tak bisa diaksesaatpergantian tahun, Jumat (1/1)
2. Para Pengguna WhatsApp di Eropa dan Amerika sempat tak bisa mengirimkan atau menerima pesan melalui aplikasi tersebut
3. Belum jelas apa yang menjadi penyebab gagal berfungsinya server milik WhatsApp, tapi menurut laporan dari situs DownDetector, gagal berfungsinya aplikasi tersebut terjadi pada pukul 4
4. 30 GMT
5. Kegagalan server WhatsApp melayani permintaan pengguna diduga terjadi karena kelebihan beban dimana pada saat itu banyak pengguna saling mengirimkan pesan setelah pergantian tahun baru
6. Terakhir kali WhatsApp mengungkapkan jumlah pengguna mereka, September lalu, perusahaan yang dimiliki Facebook ini mengklaim telah memiliki lebih dari 900 juta pengguna
7. Diyakini saat ini WhatsApp telah memiliki jumlah pengguna lebih dari satu miliar di seluruh dunia

- c) Pengambilan term yang penting untuk setiap kalimat dan judul. Hasil pengambilan term sebagai berikut:

1. ['aplikasi', 'chat', 'populer', 'dunia', 'whatsapp', 'diakses', 'pergantian', 'jumat']
2. ['pengguna', 'whatsapp', 'eropa', 'amerika', 'mengirimkan', 'menerima', 'pesan', 'aplikasi']
3. ['penyebab', 'gagal', 'berfungsi', 'server', 'milik', 'whatsapp', 'laporan', 'situs', 'downdetector', 'gagal', 'berfungsi', 'aplikasi']
4. ['gmt']
5. ['kegagalan', 'server', 'whatsapp', 'melayani', 'permintaan', 'pengguna', 'diduga', 'kelebihan', 'beban', 'dimana', 'pengguna', 'mengirimkan', 'pesan', 'pergantian']
6. ['kali', 'whatsapp', 'pengguna', 'september', 'perusahaan', 'dimiliki', 'facebook', 'mengklaim', 'memiliki', 'juta', 'pengguna']
7. ['diyakini', 'whatsapp', 'memiliki', 'pengguna', 'miliar', 'dunia']
8. Term Judul: ['server', 'whatsapp', 'down']

- d) Proses pembobotan kalimat untuk setiap kalimat berdasarkan list term kalimat dan list term judul. Pembobotan kalimat sebagai berikut:

Normalisasi frekuensi kata

1. {'chat': 0.125, 'populer': 0.125, 'jumat': 0.125, 'whatsapp': 0.125, 'aplikasi': 0.125, 'dunia': 0.125, 'pergantian': 0.125, 'diakses': 0.125}
2. {'menerima': 0.125, 'whatsapp': 0.125, 'pesan': 0.125, 'aplikasi': 0.125, 'amerika': 0.125, 'pengguna': 0.125, 'eropa': 0.125, 'mengirimkan': 0.125}
3. {'gagal': 0.16666666666666666, 'milik': 0.08333333333333333, 'berfungsi': 0.16666666666666666, 'penyebab': 0.08333333333333333, 'aplikasi': 0.08333333333333333, 'downdetector': 0.08333333333333333, 'whatsapp': 0.08333333333333333, 'laporan': 0.08333333333333333, 'server': 0.08333333333333333, 'situs': 0.08333333333333333}
4. {'gmt': 1.0}

- ```

5. {'kegagalan': 0.07142857142857142, 'server
': 0.07142857142857142, 'kelebihan': 0.07
142857142857142, 'melayani': 0.07142857142
857142, 'permintaan': 0.07142857142857142
, 'pengguna': 0.14285714285714285, 'whatsa
pp': 0.07142857142857142, 'pesan': 0.0714
2857142857142, 'beban': 0.071428571428571
42, 'pergantian': 0.07142857142857142, 'di
mana': 0.07142857142857142, 'mengirimkan'
: 0.07142857142857142, 'diduga': 0.071428
57142857142}

6. {'september': 0.09090909090909091, 'perusa
haan': 0.09090909090909091, 'whatsapp': 0
.09090909090909091, 'memiliki': 0.09090909
090909091, 'mengklaim': 0.090909090909090
91, 'kali': 0.09090909090909091, 'juta':
0.09090909090909091, 'facebook': 0.0909090
9090909091, 'dimiliki': 0.090909090909090
91, 'pengguna': 0.18181818181818182}

7. {'whatsapp': 0.16666666666666666, 'memilik
i': 0.16666666666666666, 'dunia': 0.16666
66666666666666666, 'miliar': 0.16666666666666666
6, 'diyakini': 0.166666666666666666, 'peng
guna': 0.16666666666666666}

8. Term Judul: {'server': 0.33333333333333333,
'whatsapp': 0.33333333333333333, 'down':
0.33333333333333333}
Score similarity antara term judul dan
setiap kalima
1. 0.076388888888888888
2. . 0.076388888888888888
3. 0.27777777777777778
4. 0.0
5. 0.2698412698412698
6. 0.0707070707070707
7. 0.08333333333333333

```

- e) Pemilihan kalimat yang akan ditampilkan untuk hasil peringkasan melihat dari nilai similarity. Kalimat dipilih apabila nilai similarity lebih besar sama dengan nilai threshold. Threshold yang digunakan adalah nilai rata-rata dari nilai similarity. Pada kasus ini yang digunakan adalah 0.12. Sehingga kalimat yang akan dipilih adalah kalimat 3 dan 5, dan didapatkan hasil seperti berikut:

Belum jelas apa yang menjadi penyebab gagal berfungsinya server milik WhastApp, tapi menurut laporan dari situs DownDetector, gagal berfungsinya aplikasi tersebut terjadi pada pukul 4. Kegagalan server WhatsApp melayani permintaan pengguna diduga terjadi karena kelebihan beban dimana pada saat itu banyak pengguna saling mengirimkan pesan setelah pergantian tahun baru.

### 3.3.8 Pengujian Sistem

Pengujian menggunakan *User Acceptance Testing* (UAT) untuk mengetahui tanggapan responden (*user*) terhadap sistem yang akan diinterpretasikan, yaitu dengan angket skala likert yang umumnya digunakan untuk riset berupa survei dan memberikan pertanyaan. Proses pengujian dilakukan untuk memastikan bahwa aplikasi akan dilaksanakan bekerja dengan benar. UAT penting karena membantu menunjukkan bahwa fungsi bisnis yang diperlukan beroperasi dengan cara yang sesuai dengan keadaan dan penggunaan di dunia nyata. Pengujian UAT menggunakan tahap kuesoner/angket untuk mengumpulkan informasi dengan cara menyampaikan sejumlah pertanyaan tertulis untuk menjawab secara tertulis yang dilakukan oleh responden. Teknik pengumpulan data yang dilakukan pada penelitian ini adalah dengan memberikan angket pengujian aplikasi peringkasan teks otomatis menggunakan metode pembobotan kalimat pada artikel ilmiah Bahasa Indonesia kepada *user* yang menjadi subjek pada penelitian ini untuk mengetahui cocok atau tidaknya aplikasi tersebut.

Table 8 Pertanyaan Kuesioner Aplikasi Peringkasan Teks

| No | Variabel  | Pertanyaan                                                                                                                           | Nilai |   |    |    |    |
|----|-----------|--------------------------------------------------------------------------------------------------------------------------------------|-------|---|----|----|----|
|    |           |                                                                                                                                      | SS    | S | CS | KS | TS |
| 1  | Desain    | Tampilan media yang menarik<br>Menu atau fitur mudah untuk dipahami                                                                  |       |   |    |    |    |
| 2  | Efisiensi | Hasil kalimat ringkasan mudah dipahami<br>Kalimat yang dihasilkan tidak bertele-tele<br>Menampilkan ringkasan dalam waktu yang cepat |       |   |    |    |    |
| 3  | Kepuasan  | Kemudahan mengakses sistem<br>Kepuasan menggunakan sistem                                                                            |       |   |    |    |    |

Berdasarkan tabel Pertanyaan kuesioner aplikasi peringkasan teks akan diberikan pertanyaan sebanyak 10 butir. Pertanyaan tersebut dibagi ke dalam 3 bidang penilaian yakni desain, efisiensi, dan kepuasan *user* pada sistem.

Dalam penelitian ini data akan dianalisis menjadi data kualitatif dan kuantitatif. Kelompok data yang telah teridentifikasi kemudian dibandingkan dengan data kuantitatif yang tersedia dengan data yang dikumpulkan melalui analisis kualitatif. Metode ini digunakan untuk mengolah data dari hasil penelitian untuk memperoleh kesimpulan. Data yang telah dikumpulkan selanjutnya diolah dalam tabel kemudian di bahas secara deskriptif. Ukuran deskriptif adalah pemberian angka, baik dalam responden maupun rangka persentasen yang dituangkan dalam diagram atau tabel. Analisis dilakukan dengan teknik perhitungan deskriptif persentase ini diolah dengan cara frekuensi dibagi dengan jumlah responden di kali 100%, seperti Persamaan berikut.

$$P = \frac{f}{n} \times 100\% \dots \dots \dots (5)$$

Keterangan:

P = Persentase

f = Frekuensi Jawaban

n = Jumlah Responden (Sugiono,2012).



## **V. KESIMPULAN DAN SARAN**

### **5.1. Kesimpulan**

Hasil implementasi sistem peringkasan teks otomatis pada artikel ilmiah Bahasa Indonesia yang telah dilakukan, menunjukkan bahwa sistem yang dibangun sudah memenuhi persyaratan fungsional kemudahan penggunaan dan kebermanfaatan. Penerimaan sistem tersebut diambil dari 3 aspek penilaian, yaitu desain, efisiensi, dan kepuasan pengguna terhadap sistem. Dari ketiga aspek tersebut didapatkan hasilnya berdasarkan pengolahan data kuantitatif (angket). Dari hasil pengolahan data angket (kuantitatif) diperoleh nilainya sebesar 82,6% menyatakan setuju terhadap tampilan sistem, kemudian 80,2% menyatakan setuju terhadap efisiensi kalimat yang dihasilkan pada ringkasan sistem dengan waktu yang tidak lama untuk mendapatkan hasil ringkasan pada sistem, kemudian 83,7% menyatakan setuju pada kepuasan penggunaan sistem peringkasan teks otomatis pada artikel ilmiah Bahasa Indonesia.

Dari hasil pengolahan data tersebut dapat disimpulkan bahwa aplikasi ini sudah dapat diterima dengan kemudahan dan kebermanfaatan penggunaan secara keseluruhan pada sistem peringkasan teks otomatis artikel ilmiah Bahasa Indonesia.



## 5.2. Saran

Adapun saran yang diberikan dari penelitian ini agar aplikasi dapat dikembangkan lebih baik lagi adalah sebagai berikut.

1. Pada penelitian serupa dimasa mendatang diharapkan dapat mengembangkan sistem yang dapat meringkas jurnal yang memiliki kosakata menggunakan bahasa inggris.
2. Menambahkan atau memperbarui fitur teks pada pengembangan sistem peringkasan teks untuk meningkatkan hasil ringkasan yang tepat, akurat.
3. Menambahkan fitur pada pengembangan sistem dengan bebas inputan teks agar bisa meringkas jurnal yang diinginkan.
4. Aplikasi yang dibangun dapat dikembangkan lebih lanjut pada perangkat *mobile*.

## DAFTAR PUSTAKA

- Aminudin. 2015. *Cara Efektif Belajar Laravel*. Yogyakarta. Lokomedia.
- Ardhy, Y. W. 2015. *Peringkasan Teks Otomatis Menggunakan Tanimoto Distance Jaccard Similarity dan Pembobotan Frekuensi Kemunculan Kata untuk Dokumen Berita Berbahasa Indonesia dan Inggris*. Malang. Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Aristoteles. 2011. *Pembobotan Fitur Pada Peringkasan Teks Bahasa Indonesia Menggunakan Algoritma Genetika*. Bogor. Institut Pertanian Bogor.
- Azhar, R, et al. 2016. *Pembobotan Kata Berdasarkan Klaster Pada Optimasi Coverage Diversity Dan Coherence Untuk Peringkasan Multi Dokumen*. Jawa Timur. Institut Teknologi Sepuluh Nopember. ISSN 2407-3911 Vol II No 3.
- Firman, A, et al. 2016. *Sistem Informasi Perpustakaan Online Berbasis Web*. Jurusan Teknik Elektro Fakultas Teknik UNSRAT. E-journal Teknik Elektro dan Komputer Vol 5 No 2 ISSN 2301-8402.
- Hayatin, N, et al. 2015. *Pembobotan Kalimat Berdasarkan Fitur Berita Dan Trending Issue Untuk Peringkasan Multi Dokumen Berita*. Malang. Universitas Muhammadiyah Malang. JUTI Vol 13 No 1.
- Kurniawan, T. A. 2018. *Pemodelan Usecase (UML): Evaluasi terhadap Beberapa Kesalahan Dalam Praktik*. Universitas Brawijaya. JTIK Vol 5 No 1.
- Lumbanraja, F. R. 2013. *Sistem Pencarian Data Teks dengan Menggunakan Metode Klasifikasi Rocchio (Studi Kasus: Dokumen Teks Skripsi)*. Lampung. Universitas Lampung. Kumpulan Makalah Seminar Semirata 2013.
- Mustaqhfiri, M, et al. 2012. *Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginl Relevance*. Malang. Universitas Islam Negeri Maulana Malik Ibrahim.

[PERINGKASAN TEKS OTOMATIS BERITA BERBAHASA INONESIA MENGGUNAKAN METODE MAXIMUM MARGINAL RELEVANCE](http://www.researchgate.net/publication/300087962). Diakses pada 12 September 2018.

- Palevi, A. R dan Krisnawati. 2013. *Analisis dan Perancangan Sistem Informasi Penerimaan Peserta Didik Baru Berbasis Website Pada SMP Negeri 2 Mojosongo Boyolali*. Yogyakarta. Jurnal Ilmiah DASI Vol 14 No 4 ISSN 1411-3201.
- Sugiono. 2012. *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: Alfabeta.
- Supriatna. 2018. *Implementasi dan User Accerance Test (UAT) terhadap Aplikasi E-Learninf Pada Madrasah Aliyah Negeri (MAN) 3 Kota Banda Aceh*. Banda Aceh. Universitas Negeri Ar-Raniry.
- Trisaputra, Y, et al. 2016. *Aplikasi Peringkasan Teks Berita Otomatis Menggunakan Pembobotan Kalimat*. Bogor. Institut Pertanian Bogor.
- Wahib, A dan Sari, D. L. 2016. *Peringkasan Dokumen Berdasarkan Metode Semantik Sebaran Kalimat*. Malang. Politeknik Kota Malang. Jurnal Buana Informatika Vol 8 No 1.
- Wahib, A dan Winoto, W.A. 2017. *Menghitung Bobot Sebaran Kata*. Malang. Politeknik Kota Malang. ISSN 2089-1083.
- Wahid, D. H. 2017. *Peringkasan Sentimen Ekstraktif di Twitter Menggunakan Hybird TF-IDF dan Cosine Similarity*. Yogyakarta. Universitas Gajah Mada. IJCCS Vol 10 No 2 ISSN 1978-1520.

## **LAMPIRAN**