TUGAS AKHIR – KI141502

STRATEGI PEMILIHAN KALIMAT PADA PERINGKASAN MULTI DOKUMEN

SATRIO VERDIANTO
NRP 5111 100 183

Dosen Pembimbing
Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Diana Purwitasari, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

UNDERGRADUATE THESIS – KI141502

SENTENCE SELECTION STRATEGY FOR MULTI-DOCUMENT SUMMARIZATION

SATRIO VERDIANTO
NRP 5111 100 183

Advisor
Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Diana Purwitasari, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2016

LEMBAR PENGESAHAN

Strategi Pemilihan Kalimat pada Peringkasan Multi Dokumen

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visualisasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:
SATRIO VERDIANTO
NRP. 5111 100 183

Disetujui oleh Pembimbing Tugas Akhir:

1. Dr. Agus Zainal Arifin, S.Kom., M.Kom. ……..
   NIP: 197208091995121001 (Pembimbing 1)

2. Diana Purwitasari, S.Kom., M.Sc. ……..
   NIP: 197804102003122001 (Pembimbing 2)

SURABAYA
JULI, 2016

STRATEGI PEMILIHAN KALIMAT PADA PERINGKASAN MULTI DOKUMEN

Nama Mahasiswa : Satrio Verdianto
NRP : 5111 100 183
Jurusan : Teknik Informatika, FTIF-ITS
Dosen Pembimbing 1 : Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dosen Pembimbing 2 : Diana Purwitasari, S.Kom., M.Sc.

Abstrak

Ringkasan berita diartikan sebagai teks yang dihasilkan dari satu atau lebih kalimat yang menyampaikan informasi penting dari berita. Salah satu fase penting dalam peringkasan adalah pembobotan kalimat (*sentence scoring*). Dimana pada peringkasan berita, metode pembobotannya sebagian besar menggunakan fitur dari berita sendiri. Berdasarkan hasil dari penelitian (Ferreira, et al., 2014) bahwa untuk pembobotan kalimat pada dokumen yang memiliki karakter teks pendek dan terstruktur seperti berita maka teknik pembobotan kalimat terbaik adalah dengan menggunakan kombinasi dari keempat fitur yaitu *word frequency*, TF-IDF, posisi kalimat, dan kemiripan kalimat terhadap judul *(Resemblance to the title).*

Pada penelitian ini kombinasi keempat fitur tersebut dibandingkan dengan kombinasi tiga fitur dan dua fitur dan dievaluasi menggunakan nilai ROUGE-N dan dievaluasi berdasarkan lama waktu eksekusi. Berdasarkan hasil uji coba didapatkan hasil bahwa yang paling optimal diantara keempat kombinasi fitur tersebut adalah kombinasi antara dua buah fitur yakni fitur posisi kalimat dan *word frequency* dengan nilai ROUGE-N sebesar 0.679 dan lama waktu eksekusi 28.458 detik.

Kata kunci: kemiripan kalimat terhadap judul, pembobotan kalimat, posisi kalimat, ROUGE-N, TF-IDF, word frequency

SENTENCE SELECTION STRATEGY FOR MULTI-DOCUMENT SUMMARIZATION

Student's Name	**: Satrio Verdianto**
Student's ID	**: 5111 100 183**
Department	**: Informatics Engineering, FTIF-ITS**
First Advisor	**: Dr. Agus Zainal Arifin, S.Kom., M.Kom.**
Second Advisor	**: Diana Purwitasari, S.Kom., M.Sc.**

Abstract

Summary of news is defined as a text resulting from one or more sentences that convey important information from news. One important phase in text summarization is weighting sentence (sentence scoring). In the news summarization the weighting method mostly using the features of the news itself. Based on the results of the study (Ferreira, et al., 2014) that for weighting sentences in documents that have character short text and structured as news, the technique of weighting sentence is best to use a combination of all four features that word frequency, TF-IDF, position, and Resemblance to the title.

In this study, the combination of four features compared to the combination of three features and two features and evaluated using a value ROUGE-N and evaluated based on the length of time of execution. Based on test results showed that among the four combination of these feature, the most optimal combination is the combination of two features those are position of the sentence feature and word frequency feature with ROUGE-N 0.679 and length of time of execution 28.458 sec.

Keywords: resemblance to the title, ROUGE-N, sentence position, sentence scoring, TF-IDF, word frequency

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE SUMBER

BAB I
PENDAHULUAN

Dalam bab ini akan dijelaskan hal-hal dasar mengenai Tugas Akhir ini yang meliputi latar belakang, rumusan permasalahan, batasan permasalahan, tujuan dan manfaat, metodologi, serta sistematika penulisan. Dari adanya uraian tersebut diharapkan dapat menjelaskan dengan baik mengenai permasalahan sekaligus pemecahan dalam Tugas Akhir ini.

1.1 Latar Belakang

Kebutuhan untuk mengakses informasi khususnya berita secara praktis menjadi masalah yang harus diselesaikan seiring berkembang-pesatnya berita yang dapat diakses secara *online*. Peringkasan berita secara otomatis adalah salah satu solusi untuk menjawab permasalahan diatas. Ringkasan berita dapat diartikan sebagai sebuah teks yang dihasilkan dari satu atau lebih kalimat yang mampu menyampaikan informasi penting dari sebuah berita. Dimana panjang dari sebuah ringkasan tidak lebih dari setengah panjang dokumen asli, dan biasanya lebih pendek (Radev, Hovy, & McKeown, 2002). Peringkasan multi dokumen berita merupakan sistem peringkasan yang melibatkan lebih dari satu berita sebagai input.

Selain itu, dibutuhkan teknik pembobotan kalimat yang handal untuk dapat menghasilkan ringkasan berita yang baik. Berdasarkan hasil dari penelitian (Ferreira, et al., 2014) bahwa untuk pembobotan kalimat pada dokumen yang memiliki karakter teks pendek dan terstruktur seperti berita maka teknik pembobotan kalimat terbaik adalah dengan menggunakan kombinasi dari keempat fitur yaitu *word frequency*, TF-IDF, posisi, dan *Resemblance to the title.*

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara memilih kalimat representatif dari beberapa dokumen untuk dijadikan sebagai ringkasan dengan memanfaatkan empat fitur : *word frequency*, TF-IDF, posisi kalimat, dan kemiripan kalimat dengan judul berita?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

1. Dataset berupa dokumen berita berbahasa Indonesia dengan format .xml.
2. Dataset berjumlah 45 data yang dibagi menjadi 15 topik berita.
3. Penyusunan ringkasan tidak mempertimbangkan urutan kalimat berdasarkan sistematika penulisan.
4. Ringkasan yang dihasilkan berupa kumpulan kalimat, bukan berformat paragraf.
5. Metode evaluasi yang digunakan adalah ROUGE-N, yaitu ROUGE-1.

1.4 Tujuan dan Manfaat

Tujuan dari pengerjaan Tugas Akhir ini adalah memanfaatkan fitur *word frequency*, TF-IDF, posisi kalimat, dan kemiripan kalimat dengan judul berita sebagai strategi untuk memilih kalimat dalam peringkasan multi dokumen.

Sedangkan manfaat dari pengerjaan Tugas Akhir ini adalah dapat mempermudah *user* (dalam hal ini pembaca berita *online*) mengetahui informasi penting dari sebuah berita tanpa harus membaca keseluruhan isi dari berita.

1.5 Metodologi

Ada beberapa tahap dalam proses pengerjaan Tugas Akhir ini. Berikut ini adalah tahap-tahap dalam pembuatan Tugas Akhir.

a. Studi Literatur

Pada tahap ini dipelajari mengenai strategi pemilihan kalimat berdasarkan empat fitur yakni posisi kalimat, w*ord frequency list*, TF-IDF, dan kemiripan antara kalimat dan judul pada proses peringkasan multi dokumen berita berbahasa Indonesia..

Tahap ini meliputi analisa, perancangan dan desain sistem perangkat lunak berdasarkan hasil yang diperoleh pada tahap 1. Perancangan perangkat lunak meliputi perancangan data dan proses-proses dalam system.

b. Pembuatan Perangkat Lunak

Tahap ini merupakan tahap implementasi dari rancangan yang telah dibuat pada tahap sebelumnya menjadi suatu perangkat lunak.

c. Uji Coba dan Evaluasi

Tahapan ini merupakan tahap dimana digunakan bermacam masukan untuk mengetahui apakah aplikasi dapat berjalan sesuai dengan rancangan dan desain yang dibuat. Selain itu juga untuk mencari kesalahan-kesalahan program yang mungkin terjadi sehingga dapat dilakukan penyempurnaan.

d. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan Tugas Akhir.

1.6 Sistematika Penulisan

Sistematika penulisan buku Tugas Akhir ini adalah sebagai berikut:

1. **Bab I Pendahuluan**
   Bab ini meliputi latar belakang masalah, rumusan permasalahan, batasan permasalahan, tujuan dan manfaat pembuatan Tugas Akhir, metodologi yang digunakan, dan sistematika penulisan buku Tugas Akhir.

2. **Bab II Tinjauan Pustaka**
   Bab ini meliputi dasar teori dan penunjang yang berkaitan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

3. **Bab III Metodologi**
   Bab ini berisi perancangan sistem, perancangan data, gambaran umum sistem, algoritma dan pemodelan proses, serta perancangan antar muka yang digunakan pada Tugas Akhir.

4. **Bab IV Implementasi**
   Bab ini memaparkan tahap-tahap pengimplementasian tiap fungsi atau metode yang digunakan pada Tugas Akhir ini melalui *source-code* pada perangkat lunak yang digunakan (NetBeans)

5. **Bab V Uji Coba dan Analisa Hasil**
   Bab ini memaparkan uji coba dan hasil peringkasan menggunakan kombinasi empat buah fitur.

6. **Bab VI Kesimpulan dan Saran**

   Bab ini menguraikan kesimpulan yang diambil berdasarkan hasil uji coba dan memberikan saran-saran yang bisa digunakan pada pengembangan proses peringkasan multi dokumen berita berbahasa Indonesia di masa yang akan datang.

BAB II
TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan perancangan sistem. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan.

2.1 Peringkasan Dokumen

Sebuah ringkasan menurut Dragomir (2002) dapat diartikan sebagai sebuah teks yang dihasilkan dari satu atau lebih kalimat yang menyampaikan informasi penting dari dokumen asli. Panjang dari sebuah ringkasan tidak lebih dari setengah panjang dokumen asli, dan biasanya lebih pendek lagi. Sedangkan menurut Karel (2008), peringkasan dokumen didefinisikan sebagai sebuah penyulingan informasi yang paling penting dari dokumen sumber untuk menghasilkan sebuah versi singkat untuk tugas maupun pengguna tertentu. Ketika peringkasan dilakukan oleh komputer maka disebut dengan peringkasan dokumen secara otomatis. Jenis data yang dapat diproses untuk peringkasan dapat berupa teks, video, citra atau suara. Sedangkan berdasarkan jenis medianya, peringkasan dokumen dibedakan menjadi beberapa domain yaitu berita, *email*, *social media*, artikel ilmiah, buku dan *website*.

Selain itu berdasarkan hasilnya peringkasan dokumen juga dibedakan menjadi beberapa tipe, yaitu: ekstraktif dan abtraktif, sedangkan berdasarkan orientasi ada tipe generik dan *query-focused*. Berdasarkan bahasa dibedakan menjadi monolingual dan multilingual, dan berdasarkan banyaknya sumber berita yang digunakan dibedakan menjadi *single document* dan *multi document*. Hasil ringkasan yang disusun dari beberapa kalimat yang langsung diambil dari dokumen asli disebut ringkasan ekstraktif. Sedangkan hasil ringkasan abstraktif tersusun dari kalimat yang secara struktur berbeda dari kalimat yang menyusun dokumen asli walaupun informasi yang disampaikan sama.

Berdasarkan orientasinya, peringkasan secara generik menggunakan fitur yang ada pada dokumen. Hal ini berbeda dengan peringkasan *secara user-focused* yang berorientasi pada informasi yang diberikan *user*, dapat berupa *query* atau topik yang dijadikan pertimbangan pada saat proses peringkasan dokumen. *Monolingual* hanya fokus pada satu bahasa, sedangkan *multilingual* untuk lebih dari satu bahasa. Pada *Single Document Summarization* (SDS) dan *Multi Document Summarization* (MDS) perbedaannya terletak pada jumlah dokumen input yang akan diringkas.

2.2 Peringkasan Multi Dokumen

Peringkasan pada multi dokumen berita adalah peringkasan yang melibatkan lebih dari satu dokumen berita. Dalam hal ini, biasanya berita yang digunakan sebagai input adalah berita *online*. Perbedaan antara berita yang ditulis di media cetak dengan berita *online* adalah dari judul berita dan alinea yang mana berita *online* memiliki struktur yang relatif lebih pendek. Perbedaan yang kedua, berita *online* memiliki *link* untuk mengikat antara satu berita dengan berita lainnya. Dengan adanya *link* tersebut pembaca bisa mengikuti dan membaca berita sebelumnya yang terkait dengan berita yang ada. Merujuk pada (Lumowa, 2014), secara umum format yang digunakan pada penulisan berita adalah menggunakan teknik piramida terbalik. Teknik piramida terbalik merupakan teknik penulisan berita yang akan menuliskan bagian yang dianggap penting di awal dan semakin kebawah berisi bagian yang kurang penting atau hanya sebagai pendukung. Anatomi tubuh berita terdiri dari empat bagian pokok, yaitu Judul Berita (*Head*), Teras Berita (*Lead*), Tubuh Berita (*Body*), dan Kaki Berita (*Leg*). Namun kebanyakan penulisan berita *online* hanya menggunakan deretan judul tanpa menuliskan teras berita (*lead*).

Judul merupakan satu komponen penting dalam penulisan berita. Dalam berita *online* (Lumowa, 2014), Judul berita itu harus ringkas dan jelas (*to the point*) namun tetap

dapat menarik perhatian pembaca sekaligus bisa menggambarkan isi beritanya. Dalam judul minimal mengandung unsur S-P-O-K atau berupa kalimat lengkap, pada berita *online* judul tidak boleh menggunakan kalimat tanya. Judul dapat diambil dari beberapa kata atau kutipan yang ada dalam isi berita. Panjang judul maksimal dua baris yang terdiri atas empat hingga enam kata. Sedangkan jika panjang judul hanya satu baris maka maksimal terdiri atas lima kata. Untuk judul berita utama maksimal lima kata. Semua kata di dalam judul dimulai dengan huruf besar, kecuali kata sambung seperti dan, di, yang, bila, dalam, pada, oleh, dan kata tugas lainnya yang ditentukan redaksi. Penulisan judul tidak boleh dimulai dengan angka dan harus menghindari dari penggunaan singkatan yang tidak populer. Setelah judul berita, bagian penting yang berisi penjelasan informasi yang lebih utuh dalam sebuah berita disebut dengan tubuh berita (*body*). Tubuh berita berisi penjelasan, kronologi, perincian, dan pelengkap dari lead sekaligus penghubung dengan kaki berita. Sementara di posisi paling bawah dalam piramida terbalik biasa disebut dengan kaki berita. Kaki berita adalah bagian berita yang menjelaskan hal-hal ringan yang menunjang isi berita. Seperti asal-usul narasumber, sekilas berita yang terkait dan lain sebagainya.

MEAD (*Centroid based multi-document summarization*) adalah sebuah metode peringkasan dokumen berita yang populer hasil penelitian dari Radev. Penelitian tersebut memaksimalkan fitur yang dimiliki oleh dokumen yaitu *centroid*, posisi, dan kemiripan kalimat terhadap kalimat pertama (Radev, Jing, Stys, & Tam, 2004). Penelitian lain tentang peringkasan dokumen berita yang juga menggunakan fitur yang ada pada dokumen adalah Sarkar. Penelitian Sarkar melakukan peringkasan dokumen dengan menggunakan teknik klasterisasi menggunakan dua fitur penting dari dokumen yaitu *global importance* dan *local importance* (Sarkar, 2009). Kedua penelitian tersebut adalah contoh penelitian tentang peringkasan dokumen secara generik yang hanya menggunakan fitur yang ada pada dokumen itu

sendiri. Padahal pada dokumen berita, kemunculan lebih dari satu *issue* (*multiple issue*) pada topik yang sama dapat terjadi (Kim, Kim, & Kim , 2014). Sehingga ketika fitur yang digunakan untuk peringkasan dokumen hanya diambil dari dokumen input maka kemungkinan besar akan mengakibatkan susunan ringkasan berita yang dihasilkan kurang koheren (keterpaduan makna) dikarenakan kalimat-kalimat yang menyusun ringkasan berasal dari berbagai macam *issue*. Dari banyak *issue* yang muncul pada topik yang sama dalam sebuah berita kemungkinan hanya ada beberapa *issue* yang akan menjadi pokok pembicaraan (*Trending Issue*). *Trending Issue* inilah yang harusnya dijadikan pertimbangan untuk menyeleksi kalimat penting pada proses peringkasan dokumen.

2.3 Teknik Pembobotan Kalimat

Pembobotan kalimat (*sentence scoring*) merupakan salah satu fase penting dalam peringkasan dokumen dan telah banyak digunakan pada peringkasan dokumen secara ekstraktif, baik untuk *single document* maupun *multi document*. Secara umum, metode pembobotan kalimat pada peringkasan dokumen dikelompokan menjadi tiga kategori (Ferreira, et al., 2014) yaitu berdasarkan bobot kata (*word-based scoring*), berdasarkan fitur kalimat (*sentence-based scoring*), dan berdasarkan pada relasi antar kalimat yang direpresentasikan dengan graf (*graph-based scoring*).

2.3.1 Word Frequency

Konsep dari *Word Frequency* (*WF*) adalah semakin sering suatu kata muncul dalam sebuah teks maka kata tersebut dianggap sebagai kata penting (Ferreira, et al., 2014). Sehingga untuk mendapatkan kata-kata penting dari sebuah dokumen dilakukan pembobotan kata dengan menghitung frekuensi kemunculan kata tersebut pada dokumen. Semakin besar frekuensi kemunculan sebuah kata maka skornya akan semakin tinggi. Langkah awal

yang dilakukan adalah ekstraksi *term* dari dokumen kemudian memberikan bobot pada tiap *term* tersebut berdasarkan jumlah kemunculan *term* pada dokumen. Kemudian meranking *term* berdasarkan bobot dan menyeleksi *term* yang memiliki bobot diatas nilai ambang (*threshold*). *Term* yang terseleksi akan menjadi *Word Frequency List* (*WFList*). *WFList* inilah yang nantinya digunakan sebagai fitur pada pembobotan kalimat dengan cara mengukur kemiripan antara kalimat terhadap *WFList*. Metode untuk mengukur kemiripan dapat menggunakan *cosine similarity* atau metode pengukur kemiripan yang lain.

2.3.2 TF-IDF

Term Frequency Inverse Document Frequency (TF-IDF) adalah konsep pembobotan *term* pada sebuah dokumen. Ketika TF-IDF diterapkan pada lingkup kalimat, maka sebuah kalimat akan diberlakukan sebagai dokumen. Konsep dari TF-IDF adalah jika ada "kata-kata yang spesifik" muncul pada kalimat tertentu maka kalimat tersebut relatif dianggap sebagai kalimat penting (Ferreira, Cabral, Lins, e Silva, & Freitas, 2013). Metode ini melakukan perbandingan antara frekuensi kemunculan *term j* pada kalimat *i* (TF_{ij}) dengan frekuensi kalimat yang mengandung *term j* (DF_j). Bobot TF-IDF dari *term j* dapat dihitung dengan menggunakan persamaan 2.1, dimana $tf_{w_i doc_j}$ adalah frekuensi kata *w* ke-*i* pada dokumen ke-*j*. Konsep tersebut memberikan pengukuran terhadap pentingnya kata *w* ke-*i* pada dokumen tersebut. Sedangkan idf_{w_i} ditentukan melalui Persamaan 2.2, dimana *N* adalah jumlah dokumen, df_{w_i} adalah jumlah dari dokumen yang mengandung kata *w* ke-*i*.

$$tf\_idf_{w_i doc_j} = tf_{w_i doc_j} * idf_{w_i} \quad (2.1)$$

$$idf_{w_i} = \log\left(\frac{N}{df_{w_i}}\right) \quad (2.2)$$

2.3.3 Posisi Kalimat

Posisi kalimat merupakan salah satu fitur yang dapat digunakan untuk pembobotan kalimat. Dimana penilaiannya berdasarkan pada letak kalimat dalam sebuah dokumen. Sama seperti penelitian (Mei & Chen, 2012) yang menggunakan posisi sebagai salah fitur pembobotan kalimat. Dengan menggunakan aturan, kalimat yang posisinya berada diawal dokumen memiliki skor lebih besar dibanding kalimat yang posisinya diakhir. Penelitian tersebut mampu memberikan penjelasan ilmiah tentang alasan penggunaan aturan tersebut untuk pembobotan kalimat dengan mengutip pernyataan dari Baxendale bahwa kebanyakan kalimat yang muncul diawal paragraf merupakan *topic sentence*. Hal inilah yang menjadi dasar Jiang-ping (2012) untuk memberikan skor lebih besar pada kalimat yang muncul di awal dokumen. Namun sebenarnya, *topic sentence* kurang tepat digunakan sebagai alasan dikarenakan *topic sentence* berlaku untuk semua jenis tulisan *term*asuk berita *online*. Dan *topic sentence* bisa saja muncul disemua posisi dokumen, baik di awal, tengah, maupun akhir.

Dalam ilmu jurnalistik, ada beberapa teknik penulisan berita. Teknik yang paling banyak digunakan untuk berita *online* adalah "piramida terbalik". Pola "piramida terbalik" merupakan teknik penulisan berita yang dimulai atau diawali dari kalimat yang dianggap paling penting, setelah itu diikuti hal-hal yang kurang penting. Penelitian ini meyakini bahwa alasan ini merupakan alasan yang tepat untuk memberikan skor lebih besar pada kalimat yang ada di posisi awal dibanding dengan penggunaan alasan *topic sentence*.

2.3.4 Kemiripan Kalimat terhadap Judul Berita (*Resemblance to the Title*)

Judul berita merupakan satu komponen penting dalam penulisan berita. Dalam berita *online*, judul ditulis secara ringkas dan jelas. Sebuah judul minimal mengandung unsur S-P-O-K

(Subyek – Predikat – Obyek – Keterangan) dan dapat diambil dari beberapa kata atau kutipan yang ada dalam isi berita. Hal inilah yang menjadi dasar penggunaan judul sebagai informasi untuk mengetahui kalimat penting dalam sebuah berita. Konsep dari teknik pembobotan kalimat berdasarkan kemiripan kalimat terhadap judul adalah bahwa bobot sebuah kalimat besar ketika nilai kemiripan antara judul dengan kalimat tinggi. Semakin besar bobot kalimat maka kalimat tersebut akan dianggap semakin penting. Hal ini sama seperti yang ada pada penelitian (Ferreira, Cabral, Lins, e Silva, & Freitas, 2013), bahwa kalimat yang mirip dengan judul dan kalimat yang mencakup kata-kata dalam judul yang akan dianggap sebagai kalimat penting.

2.4 Cosine Similarity

Cosine similarity adalah salah satu metode untuk mengukur kemiripan teks dengan menggunakan nilai *cosinus* sudut antara dua vektor (Salton & Buckley, 1988). Konsepnya adalah jika terdapat dua vektor dokumen d_i dan d_j maka nilai *cosinus* antara dua pasangan teks tersebut dapat dihitung dengan menggunakan persamaan 2.5. Dimana *w* adalah *term* yang diekstrak dari koleksi dokumen.

$$similarity(\overrightarrow{d_i}, \overrightarrow{d_j}) = \frac{\overrightarrow{d_i} \cdot \overrightarrow{d_i}}{|\overrightarrow{d_i}| \cdot |\overrightarrow{d_j}|} = \frac{\sum_{k=1}^{t}(w_{ki} \cdot w_{kj})}{\sqrt{\sum_{k=1}^{t} w_{ki}^2 \cdot \sum_{k=1}^{t} w_{kj}^2}} \quad (2.3)$$

2.5 Metode Evaluasi ROUGE-N

ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) adalah metode yang digunakan untuk mengukur kualitas dari sebuah ringkasan berdasarkan penelitian (Lin, 2004). ROUGE akan membandingkan antara rangkuman yang dihasilkan oleh sistem terhadap rangkuman ideal (*Groundtruth*) yang dibuat oleh pakar. Dalam hal ini, ringkasan yang dihasilkan oleh sistem disebut dengan kandidat ringkasan sedangkan ringkasan yang digunakan sebagai *Groundtruth* atau *Human Gold Standart* (HGS). Pengukuran ROUGE didasarkan pada jumlah unit yang

overlap dari tiap kata terhadap kandidat ringkasan dengan *Groundtruth*. Jenis pengukuran dengan menggunakan ROUGE ada beberapa macam. ROUGE-1 adalah jenis pengukuran yang akan digunakan dalam penelitian ini.

Pengukuran ROUGE-N didasarkan pada kemunculan secara statistik dari *n-gram* (*N-gram Co-Occurrence Statistics*). Secara formal, ROUGE-N adalah nilai recall dari *n-gram* yang ada pada kandidat ringkasan terhadap *Groundtruth*. Pengukuran nilai ROUGE-N dapat dihitung dengan menggunakan persamaan 2.4. Dimana *n* merepresentasikan panjang dari *n-gram*. Sedangkan $count_{match}$ adalah jumlah *n-gram* yang sama antara *n-gram* dari ringkasan oleh sistem dengan *n-gram* yang ada pada *Groundtruth*. Dengan penyebut dari persamaan tersebut merupakan jumlah total *n-gram* yang ada pada ringkasan referensi.

$$ROUGE - N = \frac{\sum_{S \in Summ_{ref}} \sum_{gram_n \in S} count_{match}(gram_n)}{\sum_{S \in Summ_{ref}} \sum_{gram_n \in S} count(gram_n)} \quad (2.4)$$

Ketika *Groundtruth* ringkasan yang digunakan ada lebih dari satu, maka dihitung satu persatu nilai ROUGE-N antara ringkasan yang dihasilkan oleh sistem *s* terhadap setiap *Groundtruth* referensi r_i. Selanjutnya diambil nilai maksimal ROUGE-N dari pasangan ringkasan sistem dan *Groundtruth* ringkasan. Secara formal, perhitungan ROUGE-N multi *Groundtruth* dapat dilihat pada persamaan 2.7.

$$ROUGE - N_{multi} = \arg\max_i ROUGE - N(r_i, s) \quad (2.5)$$

Dalam penelitian ini digunakan ROUGE-1. Hal ini berarti bahwa jumlah *n-gram* yang dibandingkan antara ringkasan sistem dengan *Groundtruth* berjumlah satu. Jika yang dibandingkan adalah kata-kata, maka ROUGE-1 membandingkan per satu kata pada ringkasan sistem dengan ringkasan *Groundtruth*, bukan berupa rangkaian kata.

BAB III
METODOLOGI

Pada bab ini akan dibahas mengenai perancangan dan pembuatan sistem perangkat lunak (*software*). Sistem yang dikembangkan dalam Tugas Akhir ini mengimplementasikan salah satu metode dalam peringkasan dokumen yaitu dengan mengkombinasikan empat buah fitur yakni posisi kalimat, *word frequency list*, TF-IDF, dan kemiripan antara kalimat dengan judul.

3.1 Perancangan Sistem

Pada bagian ini dibagi menjadi 3 bagian, yaitu perancangan data yang diproses dalam sistem, gambaran umum sistem, algoritma dan *flowchart*.

3.1.1 Perancangan Data

Perancangan data ini diperlukan untuk menentukan format dari data-data yang digunakan dalam sistem sehingga dapat dioperasikan secara benar. Data-data tersebut dibagi menjadi 2 jenis, yaitu:

1. Data masukan
   Data masukan merupakan data yang akan diproses dalam proses peringkasan.
2. Data keluaran
   Data keluaran merupakan data hasil proses peringkasan.

3.1.1.1 Data Masukan

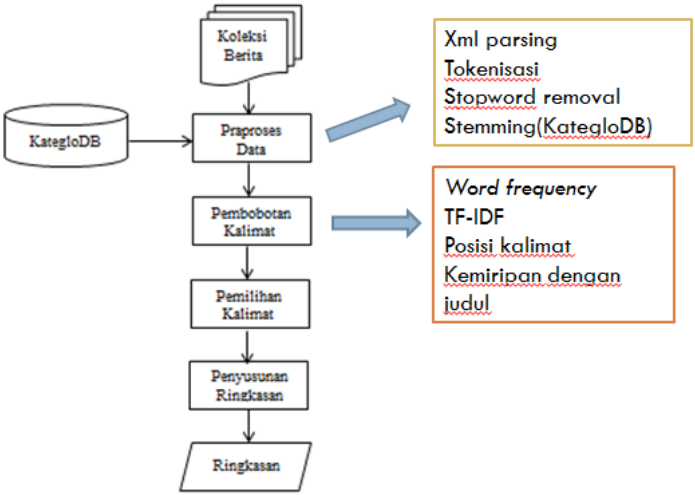
Data input yang digunakan sistem ialah teks berita berformat *.xml*.

3.1.1.2 Data Keluaran

Setelah melalui proses peringkasan, data output yang dihasilkan ialah berupa data *.txt* yang berisi kalimat-kalimat pembentuk ringkasan.

3.1.2 Gambaran Umum Sistem

Bagian ini menjelaskan secara garis besar langkah-langkah yang dilakukan baik dalam proses peringkasan dokumen.



Gambar 3.1 Diagram sistem secara umum

Fase praproses data meliputi proses xml *parsing*, *tokenizing*, *stopword removal*, dan *stemming*. XML *parsing* adalah proses pengubahan data .xml ke bentuk *string* atau teks. *Tokenizing* adalah proses pemenggalan kata-kata sehingga setiap kata dapat berdiri sendiri. *Stopword removal* adalah proses menghapus kata kunci yang tidak layak untuk digunakan, seperti kata sambung, kata depan, kata ganti dls.

Sedangkan *stemming* adalah proses untuk memperoleh kata dasar dari setiap kata. Dalam tugas akhir ini *stemming* dilakukan dengan memanfaatkan kategloDB. Proses *stemming* dilakukan dengan mengubah setiap kata ke bentuk dasarnya dengan merujuk ke kategloDB. Data hasil praproses disimpan ke dalam database.

3.1.2.2 Pembobotan Kalimat

Fase pembobotan kalimat merupakan proses perhitungan empat buah fitur untuk tiap kalimat. Keempat buah fitur tersebut ialah fitur posisi kalimat, fitur *word frequency,* fitur TF-IDF, dan fitur kemiripan kalimat dengan judul. Konsep dari pembobotan kalimat dengan menggunakan *WF* sesuai dengan penjelasan pada subbab 2.3.1. Dalam hal ini *WFList* didapatkan dari sejumlah *term* dengan nilai *WF* memenuhi nilai ambang (*threshold*), *WFList* = $\{WF_1, \ldots, WF_k\}$. Pembobotan kalimat dihitung berdasarkan nilai kemiripan antara kalimat terhadap *WFList*, persamaan 3.1. Dalam penelitian ini digunakan *cosine similarity* untuk mengukur kemiripan antara kalimat dengan *WFList*, persamaan 2.4. Bobot kalimat berdasarkan *WF* untuk selanjutnya disebut dengan w_1, dengan S adalah kalimat. Sehingga $w_1(s_i)$ adalah nilai kemiripan kalimat s_i terhadap *WFList*, dimana $S=\{s_1, \ldots, s_m\}$.

Pembobotan kalimat kedua (w_2) pada penelitian ini menggunakan pendekatan TF-IDF. Konsep dan cara penghitungan bobot *term* berdasarkan TF-IDF dijabarkan pada subbab 2.3.2. Setelah didapatkan bobot tiap *term* $TFIDF_{ij}$ dengan menggunakan persamaan 2.1, langkah selanjutnya adalah menghitung bobot kalimat berdasarkan bobot TF-IDF yang selanjutnya disebut dengan w_2 menggunakan persamaan 3.2. w_2 merupakan hasil

penjumlahan dari seluruh bobot *term j* yang muncul pada kalimat *i* (s_i).

Pembobotan kalimat ketiga (w_3) menggunakan fitur posisi. Penjelasan tentang fitur posisi dijelaskan pada subbab 2.3.3 dan 3.3.3. w_3 dihitung dengan menggunakan persamaan 3.3 yang mengadopsi dari penelitian (Mei & Chen, 2012). Dengan aturan, kalimat yang posisinya berada diawal dokumen memiliki skor lebih besar dibanding kalimat yang posisinya diakhir.

Pembobotan kalimat keempat (w_4) melibatkan judul berita (*Title*). Konsep pembobotan kalimat berdasarkan kemiripan terhadap judul dijelaskan pada subbab 2.3.4 sedangkan penjelasan tentang cara pengambilan judul dari berita dijelaskan pada subbab 3.3.3. Penghitungan w_4 menggunakan persamaan 3.4 yang mengadopsi dari (Ferreira, Cabral, Lins, e Silva, & Freitas, 2013) yaitu dengan cara membagi antara jumlah *term* judul yang muncul pada kalimat (*Ntw*) dengan jumlah seluruh *term* yang ada pada judul (*T*).

$$w_1(s_i) = Sim(s_i, WFList) \quad (3.\ 1)$$

$$w_2(s_i) = \sum_{j=1}^{n} TFIDF_{ij} \quad (3.\ 2)$$

$$w_3(s_i) = \frac{1}{\sqrt{POS(S_i)}} \quad (3.\ 3)$$

$$w_4(s_i) = \frac{NTW}{T} \quad (3.\ 4)$$

Setelah didapatkan bobot w_1 sampai w_4 langkah berikutnya adalah menghitung total bobot kalimat *i* dengan menggunakan persamaan 3.5. Bobot kalimat yang didapat berdasarkan w_1 sampai w_4 seluruhnya dijumlahkan. Seluruh kalimat akan dihitung bobotnya, hasil dari persamaan 3.5 inilah yang akan menjadi total bobot kalimat *i*.

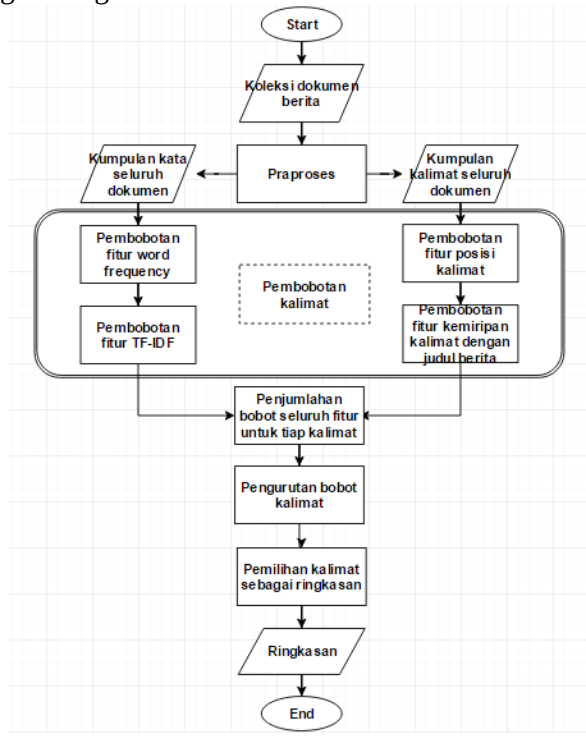
$$score(s_i) = w_1(s_i) + w_2(s_i) + w_3(s_i) + w_4(s_i) \quad (3.\ 5)$$

3.1.2.3 Pemilihan Kalimat dan Penyusunan Ringkasan

Fase pemilihan kalimat dan penyusunan ringkasan dilakukan dengan melakukan pengurutan bobot kalimat secara *descending* (terbesar ke terkecil). Kemudian beberapa kalimat dengan bobot terbesar diambil sebagai ringkasan.

3.1.3 Algoritma dan Diagram Alir

Pada subbab ini akan dijelaskan tentang alur proses sistem secara umum dan untuk alur proses untuk perhitungan masing-masing fitur.



Gambar 3.2 Diagram alir sistem secara umum

Gambar 3.2 menunjukkan alur proses sistem secara umum. Perbedaan antara gambar 3.2 dengan gambar 3.1 adalah pada gambar 3.2 dijelaskan secara lebih detil tentang bagaimana pembagian perhitungan skor fitur untuk masing-masing hasil praproses. Dalam hal ini praproses menghasilkan dua hal yakni kumpulan kata seluruh dokumen yang sudah melalui proses *stemming* dan *stopword removal*, dan kumpulan kalimat seluruh dokumen.

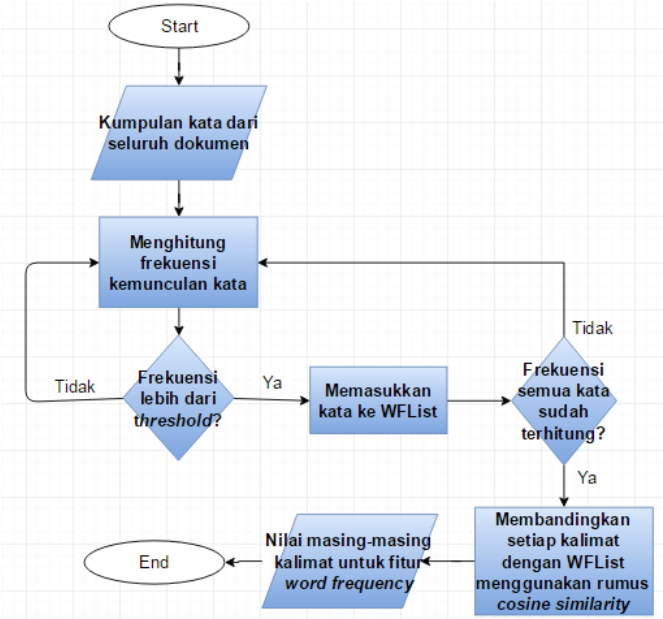
Berdasarkan gambar 3.2 dapat dilihat bahwa pembobotan kalimat melibatkan fitur kata dan fitur kalimat dari kumpulan dokumen berita. Untuk fitur kata, pembobotan dilakukan dengan menghitung fitur *word frequency* dan fitur TF-IDF. Sementara untuk fitur kalimat, pembobotan dilakukan dengan menghitung fitur posisi kalimat dan fitur kemiripan kalimat dengan judul berita.

Selanjutnya untuk masing-masing kalimat, skor keempat fitur tersebut dijumlahkan sebagai bobot total per kalimat (gambar 3.7). Kemudian kalimat diurutkan berdasarkan bobotnya dan dipilih *n* kalimat dengan bobot terbesar untuk dijadikan sebagai ringkasan (gambar 3.8).

3.1.3.1 *Word Frequency*

Berdasarkan gambar 3.3, untuk mendapatkan skor kalimat berdasarkan fitur *word frequency*, dilakukan langkah-langkah sebagai berikut.

1. Masing-masing kata yang didapat dari hasil praproses dihitung frekuensi kemunculannya di seluruh dokumen.
2. Jika frekuensi kata melebihi *threshold*, kata tersebut dimasukkan ke dalam sebuah *array WFList*.
3. Selanjutnya, setiap kalimat dibandingkan dengan *WFList* tersebut menggunakan *cosine similarity*.
4. Hasil *cosine similarity* itulah yang akan menjadi skor kalimat untuk fitur *word frequency*.



Gambar 3.3 Diagram alir perhitungan skor kalimat untuk fitur *word frequency*

3.1.3.2 TF-IDF

Berdasarkan gambar 3.4, untuk mendapatkan skor kalimat berdasarkan fitur TF-IDF, dilakukan langkah-langkah sebagai berikut.

1. Masing-masing kata yang didapat dari hasil praproses dihitung TF-IDF nya.
2. Selanjutnya, TF-IDF tiap kalimat dihitung dengan menjumlahkan nilai TF-IDF masing-masing katanya.

3. Hasil penjumlahan tersebut akan menjadi skor kalimat untuk fitur TF-IDF.

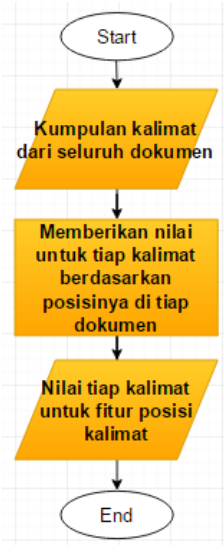
Gambar 3.4 Diagram alir perhitungan skor kalimat untuk fitur TF-IDF

3.1.3.3 Posisi Kalimat

Berdasarkan gambar 3.5, untuk mendapatkan skor kalimat berdasarkan fitur posisi kalimat, dilakukan langkah-langkah sebagai berikut.

1. Mengumpulkan seluruh kalimat dari seluruh dokumen.
2. Selanjutnya, tiap kalimat dihitung bobotnya berdasarkan posisi kalimat tersebut pada dokumen (persamaan 3.3)

3. Bobot tersebut akan menjadi skor kalimat untuk fitur posisi kalimat.

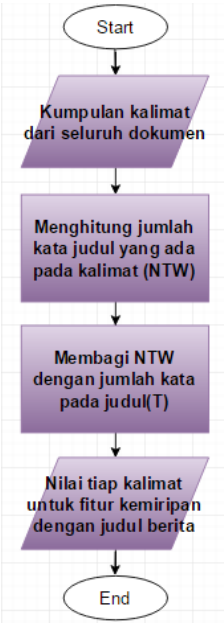


Gambar 3.5 Diagram alir perhitungan skor kalimat untuk fitur posisi kalimat

3.1.3.4 Kemiripan Kalimat dengan Judul Berita

Berdasarkan gambar 3.6, untuk mendapatkan skor kalimat berdasarkan fitur posisi kalimat, dilakukan langkah-langkah sebagai berikut.

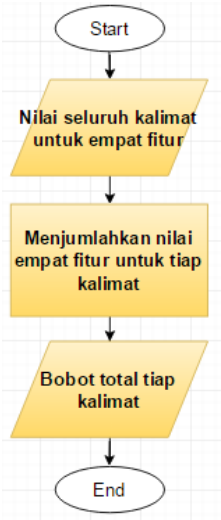
1. Mengumpulkan seluruh kalimat dari seluruh dokumen.
2. Tiap kalimat dihitung bobotnya berdasarkan posisi kalimat tersebut pada dokumen (persamaan 3.3).
3. Bobot tersebut akan menjadi skor kalimat untuk fitur posisi kalimat.



Gambar 3.6 Diagram alir perhitungan skor kalimat untuk fitur kemiripan kalimat dengan judul berita

3.1.3.5 Pembobotan Kalimat

Berdasarkan gambar 3.7, pembobotan per kalimat dilakukan dengan menjumlahkan seluruh bobot fitur untuk tiap kalimat tersebut sebagaimana ditunjukkan pada (persamaan 3.5).

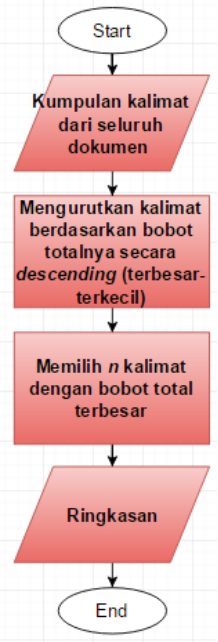


Gambar 3.7 Diagram alir perhitungan total skor kalimat

3.1.3.6 Pemilihan Kalimat sebagai Ringkasan

Berdasarkan gambar 3.8, pemilihan kalimat dilakukan dengan langkah-langkah sebagai berikut.

1. Mengurutkan seluruh kalimat berdasarkan bobotnya secara *descending* (terbesar-terkecil).
2. Sebanyak *n* buah kalimat dengan bobot terbesar dipilih untuk dijadikan sebagai ringkasan.



Gambar 3.8 Diagram alir pemilihan kalimat sebagai ringakasan

BAB IV
IMPLEMENTASI

Setelah melakukan proses perancangan, dilakukan proses implementasi sistem. Dalam tahap pengimplementasian ini terdapat beberapa bahasan yang mencakup lingkungan implementasi dan implementasi program. Kedua bahasan tersebut terangkum dalam sub-bab berikut.

4.1 Lingkungan Implementasi

Perangkat lunak diimplementasikan pada lingkungan sebagai berikut:

- Perangkat keras
  Perangkat lunak ini diimplementasikan pada sebuah *laptop* dengan spesifikasi prosesor AMD A4-3330MX APU with Radeon(tm) HD Graphics 2.30 GHz dan *memory* 3.47 GB.
- Perangkat lunak
  Perangkat lunak ini dikembangkan pada sistem operasi Microsoft Windows 7 Ultimate dengan menggunakan *tool* NetBeans 8.1.

4.2 Implementasi Program

Berikut akan dijabarkan implementasi pembobotan kalimat dalam proses pembuatan pembuatan ringkasan.

4.2.1 *Word Frequency*

Dengan menggunakan fitur *word frequency* (*WF*), setiap kata dihitung frekuensi kemunculannya pada masing-masing dokumen berita. Selanjutnya *WFList* didapatkan dari sejumlah term dengan nilai *WF* memenuhi nilai ambang

(threshold), $WFList = \{WF_1, \ldots, WF_k\}$. Pembobotan kalimat dihitung berdasarkan nilai kemiripan antara kalimat terhadap *WFList* persamaan 3.1. Dalam penelitian ini digunakan cosine similarity untuk mengukur kemiripan antara kalimat dengan *WFList*, persamaan 2.4.

4.2.1.1 Menghitung Frekuensi Masing-masing Kata (*Word Frequency*)

```
78  String q = "select distinct id_berita from kata "; //ambil smua dokumen
79  PreparedStatement ps = conn.prepareStatement(q);
80  final ResultSet rs = ps.executeQuery();
81  while(rs.next())
82  {
83      id_berita = rs.getInt("id_berita");
84      String q1 = "select distinct kata from kata where id_berita=?"; //ambil semua
85      PreparedStatement ps1 = conn.prepareStatement(q1);
86      ps1.setInt(1, id_berita);
87      final ResultSet rs1 = ps1.executeQuery();
88      while(rs1.next())
89      {
90          kata = rs1.getString("kata");
91          String q2 = "select count(kata) from kata where kata=? and id_berita=?";
92          PreparedStatement ps2 = conn.prepareStatement(q2);
93          ps2.setString(1,kata);
94          ps2.setInt(2, id_berita);
95          final ResultSet rs2 = ps2.executeQuery();
96          while(rs2.next())
97          {
98              frekuensi_kata = rs2.getInt("count(kata)");
99              //System.out.println(id_berita + " " + kata + " " + frekuensi_kata );
100             String q3="insert into frekuensi_kata (id_frekuensi_kata,kata,frekuen
101             PreparedStatement ps3 = conn.prepareStatement(q3);
102             ps3.setInt(1, count);
103             ps3.setString(2, kata);
104             ps3.setInt(3, frekuensi_kata);
105             ps3.setInt(4, id_berita);
106             ps3.execute();
107             count++;
```

Kode Sumber 4.1 Menghitung frekuensi kata

Dari Kode Sumber 4.1 dapat dilihat bahwa pada baris 91 jumlah kata *k* untuk dokumen berita ke-*id_berita* dihitung dengan menggunakan fungsi SQL *count*. Selanjutnya pada baris 98 jumlah kata disimpan pada variabel frekuensi_kata dan pada baris 100 disimpan kembali ke dalam *database* pada tabel *frekuensi_kata*.

4.2.1.2 Membangun *Word Frequency List*

Pada Kode Sumber 4.2 dapat dilihat bahwa pada baris 119 kata dengan frekuensi lebih dari *threshold* diambil untuk selanjutnya dimasukkan ke dalam *WFList* (baris 125).

```
116  int threshold=6; //frekuensi minimal kata utk masuk ke WFList
117  DBConnection dbc = new DBConnection ("summer","root","root");
118  Connection conn = dbc.connect();
119  String q = "select kata from frekuensi_kata group by kata having sum(frekuensi_kata)>=?";
120  PreparedStatement ps = conn.prepareStatement(q);
121  ps.setInt(1, threshold);
122  final ResultSet rs = ps.executeQuery();
123  while(rs.next())
124  {
125      WFList.add(rs.getString("kata"));
126  }
```

Kode Sumber 4.2 Membangun *WFList*

4.2.1.3 Membandingkan Kalimat dengan *Word Frequency List*

Pada Kode Sumber 4.3 dapat dilihat bahwa pada baris 152 *WFList* dan kalimat dihitung kemiripannya dengan menggunakan rumus *cosine similarity*. Selanjutnya pada baris 153 nilai *cosine similarity* disimpan ke dalam variabel *sentenceWFLScore[i]* untuk memberikan nilai fitur *WF* untuk kalimat ke-*i*.

```
Set<String> sentence = new HashSet<String>();
int id_kalimat;
DBConnection dbc = new DBConnection ("summer","root","root");
Connection conn = dbc.connect();
String q = "select distinct id_kalimat from kata"; //ambil smua doku
PreparedStatement ps = conn.prepareStatement(q);
final ResultSet rs = ps.executeQuery();
while(rs.next())
{
    id_kalimat = rs.getInt("id_kalimat");
    String q1 = "select distinct kata from kata where id_kalimat=?";
    PreparedStatement ps1 = conn.prepareStatement(q1);
    ps1.setInt(1, id_kalimat);
    final ResultSet rs1 = ps1.executeQuery();
    while(rs1.next())
    {
        sentence.add(rs1.getString("kata"));
    }
    CosineSimilarity cs = new CosineSimilarity();
    double cs_value = cs.calculateCosineSimilarity(WFList,sentence);
    sentenceWFLScore[id_kalimat]=cs_value;
    sentence.clear();
```

Kode Sumber 4.3 Menghitung kemiripan kalimat dengan *WFList*

4.2.2 TF-IDF

4.2.2.1 Menghitung Nilai TF-IDF Masing-masing Kata

Pada Kode Sumber 4.4 dapat dilihat proses perhitungan TF-IDF tiap kata untuk masing-masing dokumen berita. Pada baris 182, *SUM(frekuensi_kata) from frekuensi_kata where id_berita=i group by id_berita* menghitung jumlah kata pada dokumen berita ke-*id_berita*. Jumlah kata kemudian disimpan ke variabel total_kata (baris 188). Selanjutnya kata dan frekuensi kemunculannya diambil dari tabel frekuensi_kata. Kemudian frekuensi kemunculan tersebut dibagi dengan *total_kata* untuk menjadi nilai TF (baris 199). Kemudian pada baris 200, *"select count(distinct id_berita) as ndokata from frekuensi_kata where kata=k"* menghitung jumlah dokumen yang mengandung kata *k* dan disimpan ke dalam variabel *ndokumen_kata* (baris 206). Selanjutnya pada baris 208, nilai

IDF dihitung dengan rumus *IDF* = *log*$_{10}$*(ndokumen/ndokumen_kata)*, dimana *ndokumen* adalah banyaknya dokumen *D* dan *ndokumen_kata* adalah banyaknya dokumen yang mengandung kata *k*. Kemudian TF-IDF dihitung dengan mengalikan nilai TF dan nilai IDF (baris 210). Selanjutnya nilai TF-IDF untuk kata *k* pada dokumen berita *d* disimpan ke dalam *database* pada tabel tfidf.

```
191  id_berita = rs.getInt("id_berita");
192  String q1 =  "select SUM(frekuensi_kata) from frekuensi_kata where id_berita=? group by id_berita";
193  PreparedStatement ps1 = conn.prepareStatement(q1);
194  ps1.setInt(1, id_berita);
195  final ResultSet rs1 = ps1.executeQuery();
196  while(rs1.next())
197  {
198      total_kata = rs1.getInt("SUM(frekuensi_kata)");
200      String q2 = "select kata, frekuensi_kata from frekuensi_kata where id_berita = ?";
201      PreparedStatement ps2 = conn.prepareStatement(q2);
202      ps2.setInt(1, id_berita);
203      final ResultSet rs2 = ps2.executeQuery();
204      while(rs2.next())
205      {
206          kata = rs2.getString("kata");
207          nkata = rs2.getInt("frekuensi_kata");
208          //tf
209          tf = (double) nkata / total_kata;
210          String q3 ="select count(distinct id_berita) as ndokata from frekuensi_kata where kata=?";
211          PreparedStatement ps3 = conn.prepareStatement(q3);
212          ps3.setString(1, kata);
213          final ResultSet rs3 = ps3.executeQuery();
214          while(rs3.next())
215          {
216              ndokumen_kata = rs3.getInt("ndokata");
217              //idf
218              idf = (double) Math.log10(ndokumen / ndokumen_kata);
219              //tf-idf
220              tfidf = (double) tf * idf;
221              String q4 = "insert into tfidf(id_tfidf, kata, id_berita, tf, idf, tfidf) values(?,?,?,
```

Kode Sumber 4.4 Menghitung nilai TF-IDF tiap kata

4.2.2.2 Menghitung Nilai TF-IDF Kalimat

Nilai TF-IDF kalimat dihitung dengan menjumlahkan seluruh nilai TF-IDF kata pada kalimat tersebut. Pada Kode Sumber 4.5 proses ini dimulai dengan mengambil seluruh *id_kalimat* (baris 248) kemudian mengambil kata-kata yang ada pada kalimat dengan *id_kalimat* tersebut (baris 255). Selanjutnya nilai TF-IDF untuk masing-masing kata diambil dari tabel tfidf (baris 262) . Kemudian nilai TF-IDF dari kata-

kata tersebut disimpan dan diakumulasikan pada *array sentenceTF-IDFScore[id_kalimat]* (baris 269) sebagai nilai TF-IDF dari suatu kalimat.

```
247   String q = "select distinct id_berita from kata"; //mengambil id_berita dari
248   PreparedStatement ps = conn.prepareStatement(q);
249   ResultSet rs = ps.executeQuery();
250   while(rs.next())
251   {
252      id_berita = rs.getInt("id_berita");
253      String q1= "select distinct id_kalimat from kata where id_berita=?"; //men
254      PreparedStatement ps1 = conn.prepareStatement(q1);
255      ps1.setInt(1, id_berita);
256      ResultSet rs1 = ps1.executeQuery();
257      while(rs1.next())
258      {
259          id_kalimat = rs1.getInt("id_kalimat");
260          String q2= "select distinct kata from kata where id_kalimat=?";
261          PreparedStatement ps2 = conn.prepareStatement(q2);
262          ps2.setInt(1, id_kalimat);
263          ResultSet rs2 = ps2.executeQuery();
264          while(rs2.next())
265          {
266              String kata = rs2.getString("kata");
267              String q3= "select tfidf from tfidf where kata=? and id_berita=?";
268              PreparedStatement ps3 = conn.prepareStatement(q3);
269              ps3.setString(1, kata);
270              ps3.setInt(2, id_berita);
271              ResultSet rs3 = ps3.executeQuery();
272              while(rs3.next())
273              {
274                  sentenceTFIDFScore[id_kalimat] += rs3.getDouble("tfidf");
275              }
```

Kode Sumber 4.5 Menghitung nilai TF-IDF tiap kalimat

4.2.3 Posisi Kalimat

Penggunaan fitur posisi kalimat menggunakan rumus sederhana sesuai yang telah dijabarkan pada persamaan 3.3. Pada Kode Sumber 4.6 perhitungan nilai fitur posisi kalimat dapat dilihat pada baris 306. Sebelum memberikan nilai untuk tiap kalimat berdasarkan fitur posisinya, terlebih dahulu dihitung jumlah kalimat untuk masing-masing dokumen sehingga kalimat pertama untuk masing-masing dokumen selalu mendapatkan nilai tertinggi untuk fitur posisi.

```
296     String q = "select distinct(id_berita) from kalimat"; //mengambil id_berita dari tabel kata
297     PreparedStatement ps = conn.prepareStatement(q);
298     ResultSet rs = ps.executeQuery();
299     while(rs.next())
300     {
301        id_berita = rs.getInt("id_berita");
302        String q1= "select count(id_kalimat) from kalimat where id_berita=? group by id_berita";
303        PreparedStatement ps1 = conn.prepareStatement(q1);
304        ps1.setInt(1, id_berita);
305        ResultSet rs1 = ps1.executeQuery();
306        while(rs1.next())
307        {
308           int jumlah_kalimat = rs1.getInt("count(id_kalimat)");
309           for(int i=0;i<jumlah_kalimat;i++) //jumlah kalimat per id_berita
310           {
311              sentencePositionScore[indeks_kalimat] = 1 / Math.sqrt(i+1);
312              //System.out.println("sentencePosScore-" +indeks_kalimat+" = "+sentencePosition
313              indeks_kalimat++;
314           }
```

Kode Sumber 4.6 Posisi kalimat

4.2.4 Kemiripan antara Kalimat dengan Judul

Sebelum melakukan pembandingan antara kalimat dengan judul, dilakukan pengambilan kata-kata penyusun judul (baris 317) yang kemudian disimpan ke dalam *array* kata_judul (baris 325).

```
327     int id_berita;
        Set<String> kata_judul = new HashSet<String>();
        int panjang_judul=0;
330     String kalimat;
331     int index_kalimat=0;
332
333     String q = "select distinct id_berita from kalimat";
334     PreparedStatement ps = conn.prepareStatement(q);
335     ResultSet rs = ps.executeQuery();
336     while(rs.next())
337     {
338         id_berita = rs.getInt("id_berita");
339         String q1 = "select judul from judul where id_berita=?";
340         PreparedStatement ps1 = conn.prepareStatement(q1);
341         ps1.setInt(1, id_berita);
342         ResultSet rs1 = ps1.executeQuery();
343         while(rs1.next())
344         {
345             String[] judul = rs1.getString("judul").toLowerCase().split
346             Preprocessing p = new Preprocessing();
347             p.readStopWords("stopword.txt");
348             for(String j : judul)
349             {
350                 if(p.isStopWord(j))
                        continue;
352                 else
353                  kata_judul.add(j);
354             }
355         }
356         panjang_judul = kata_judul.size();
```

Kode Sumber Mengambil kata-kata judul

Selanjutnya pada Kode Sumber 4.8 setiap kalimat dicek apakah mengandung kata yang ada pada judul (baris 339). Jika mengandung maka *counter +1*. Kemudian total kata judul yang muncul kalimat dibagi dengan panjang judul dan disimpan ke dalam *sentenceTitleScore[indeks_kalimat]* sebagai nilai fitur kemiripan antara kalimat dengan judul dari suatu kalimat.

```
357     String q2 = "select kalimat from kalimat where id_berita=?";
358     PreparedStatement ps2 = conn.prepareStatement(q2);
359     ps2.setInt(1, id_berita);
360     ResultSet rs2 = ps2.executeQuery();
361     while(rs2.next())
362     {
363         kalimat = rs2.getString("kalimat");
364         int count=0;
365         Iterator it = kata_judul.iterator();
            while(it.hasNext())
367         {
368             String kataJudul = (String) it.next();
369             if(kalimat.toLowerCase().contains((kataJudul)))
370             {
371                 System.out.println(kataJudul);
372                 count++;
373             }
374         }
375         System.out.println("kalimat ke-" + index_kalimat + " = " + count);
376         sentenceTitleScore[index_kalimat++] = count / panjang_judul;
377     }
```

Kode Sumber 4.8 Menghitung nilai kemiripan antara kalimat dengan judul

4.2.5 Total Bobot Kalimat

Pada Kode Sumber 4.9 bobot dari keempat fitur untuk suatu kalimat yakni fitur *word frequency*, TF-IDF, posisi kalimat, dan kemiripan antara kalimat dengan judul dijumlahkan dan disimpan ke dalam *sentenceTotalScore[i]* sebagai bobot untuk kalimat tersebut.

```
389     for(int i=0;i<nkalimat;i++)
390     {
391         sentenceTotalScore[i] = sentenceWFLScore[i] +
392                                 sentenceTFIDFScore[i] +
393                                 sentencePositionScore[i] +
394                                 sentenceTitleScore[i];
395     }
```

Kode Sumber 4.9 Menghitung total bobot kalimat

BAB V
UJI COBA DAN ANALISA HASIL

5.1 Lingkungan Uji Coba

Lingkungan uji coba merupakan tempat atau perangkat dimana proses uji coba dilakukan sehingga nantinya mendapatkan hasil untuk analisa dan evaluasi. Tabel 5.1 berikut adalah lingkungan uji coba yang digunakan pada Tugas Akhir ini.

Tabel 5.1 Lingkungan uji coba

Perangkat Keras	Prosesor : AMD A4-3330MX APU with Radeon(tm) HD Graphics 2.30 GHz Memori : 3.47 GB
Perangkat Lunak	Sistem operasi : Microsoft Windows 7 Ultimate Perangkat Pengembang : NetBeans 8.1

5.2 Metodologi Pengujian

Pengujian pada sistem peringkasan dalam penelitian ini dilakukan dengan membandingkan hasil ringkasan sistem dengan hasil ringkasan manusia dengan menggunakan ROUGE-N. Pengujian dilakukan terhadap 15 kelompok dokumen berita berformat .xml yang dikelompokkan berdasarkan topik dimana masing-masing kelompok memiliki jumlah dokumen berita yang dijelaskan pada Tabel 5.2. Ringkasan yang dihasilkan terdiri dari 10 buah kalimat untuk masing-masing topik.

Tabel 5.2 Dataset berita

NO	TOPIK BERITA	JUMLAH BERITA
1	BLBI	4
2	LG G4	3
3	Kunjungan Mark Zuckerberg	4
4	Internet Indonesia lambat	2
5	Intel	3
6	Prosesor baru Intel	3
7	Smartphone 4G Intel	3
8	Kunjungan Jokowi	2
9	Pidato Presiden dan pemberian penghargaan	3
10	Saran SBY	3
11	Proyek LRT	3
12	Jokowi ke Arab Saudi	2
13	Jokowi ancam copot menteri	2
14	Iklan Jokowi	5
15	Unikom dan UPI di LIMA Badminton	3
TOTAL		45

5.3 Uji Coba

Uji coba dilakukan dengan mengukur performa hasil ringkasan dengan menggunakan kombinasi empat fitur berita yaitu posisi kalimat (p), *word frequency (*w*)*, TF-IDF (t), dan judul berita (j). Nantinya kombinasi 4 fitur akan dibandingkan dengan kombinasi 3 fitur dan kombinasi 2 fitur. Untuk mengukur performansi hasil ringkasan digunakan metode evaluasi ROUGE-N yaitu ROUGE-1 dan evaluasi berdasarkan waktu eksekusi.

5.3.1 Evaluasi berdasarkan Nilai ROUGE-N

Tabel 5.3 Nilai ROUGE-1 antara ringkasan sistem (kombinasi 2, 3, dan 4 fitur) dengan ringkasan *groundtruth*

RINGKASAN *GROUNDTRUTH*	NILAI ROUGE-1 TIAP KOMBINASI FITUR *p =posisi, w =word frequency, t =tfidf, j =judul*										
	pwtj	***pwt***	***pwj***	***ptj***	***wtj***	***pw***	***pt***	***pj***	***wt***	***wj***	***tj***
1	0.83221	0.83221	0.83221	0.67542	0.72477	0.83221	0.67542	0.38254	0.72477	0.75829	0.48445
2	0.38267	0.38267	0.34409	0.33566	0.27206	0.34409	0.33566	0.44898	0.27206	0.29213	0.38225
3	0.78801	0.78801	0.79911	0.80973	0.71548	0.79911	0.80973	0.50679	0.71548	0.55895	0.60938
4	0.6962	0.79654	0.62472	0.70782	0.43441	0.72517	0.75162	0.64615	0.49443	0.39912	0.51089
5	0.6501	0.6501	0.62128	0.60285	0.51402	0.62128	0.60285	0.39443	0.51402	0.47826	0.56513
6	0.61191	0.61191	0.61191	0.68526	0.569	0.61191	0.68526	0.57328	0.569	0.53719	0.58027
7	0.76082	0.76082	0.76082	0.7713	0.52008	0.76082	0.7713	0.61157	0.52008	0.52008	0.54656
8	0.504	0.504	0.504	0.52713	0.43103	0.504	0.52713	0.40304	0.43103	0.51012	0.58451
9	0.67269	0.67269	0.67269	0.79175	0.53215	0.67269	0.79175	0.40429	0.53215	0.53881	0.38968
10	0.59726	0.59726	0.59726	0.49874	0.46991	0.59726	0.49874	0.33939	0.46991	0.40491	0.36317
11	0.78286	0.78286	0.78286	0.7181	0.53786	0.78286	0.7181	0.40816	0.53786	0.53786	0.50867
12	0.94944	0.94944	0.81346	0.8	0.6513	0.81346	0.8	0.68599	0.6513	0.63128	0.67005
13	0.60047	0.60047	0.64732	0.67317	0.63514	0.64732	0.67317	0.66667	0.63514	0.63514	0.64286
14	0.79741	0.79741	0.8341	0.8341	0.62737	0.8341	0.8341	0.447	0.62737	0.61123	0.41942
15	0.58364	0.58364	0.6457	0.65376	0.60571	0.6457	0.65376	0.47692	0.60571	0.59398	0.49737
TOTAL	**10.20969**	**10.31003**	**10.09153**	**10.08479**	**8.24029**	**10.19198**	**10.12859**	**7.3952**	**8.30031**	**8.00735**	**7.75466**
RATA-RATA	**0.680646**	**0.687335**	**0.672769**	**0.672319**	**0.549353**	**0.679465**	**0.675239**	**0.493013**	**0.553354**	**0.533823**	**0.516977**
URUTAN	**2**	**1**	**5**	**6**	**8**	**3**	**4**	**11**	**7**	**9**	**10**

nilai tertinggi nilai terkecil

Dari Tabel 5.3 dapat dilihat bahwa kombinasi empat fitur yakni posisi kalimat (*p*), *word frequency (w)*, TF-IDF (*t*), dan judul berita (*j*) memiliki total nilai ROUGE-N terbesar kedua setelah kombinasi tiga fitur yakni posisi kalimat, *word frequency*, dan TF-IDF. Perbedaan ditunjukkan pada dataset ringkasan ke-4 yang disebabkan oleh struktur xml yang kurang tepat sehingga mengakibatkan sistem dengan kombinasi *pwtj* mengambil kalimat yang bukan bagian dari berita, yakni kalimat *"Baca juga:…."*

Selanjutnya posisi ketiga adalah kombinasi dua fitur yakni posisi kalimat dan *word frequency*. Perbedaan nilai ROUGE-N diantara ketiganya pun dapat dikatakan sangat kecil yakni sekitar 0.01. Hal ini menunjukkan bahwa kombinasi empat fitur bukanlah yang terbaik dalam menghasilkan ringkasan yang baik. Selain itu, hasil ini juga menunjukkan bahwa fitur posisi kalimat dan *word frequency* mengambil peranan penting dalam menghasilkan ringkasan

yang baik. Sebaliknya fitur kemiripan dengan judul berita tidak peranan penting karena dengan atau tanpa fitur tersebut nilai ROUGE-1 tidak menunjukkan perbedaan yakni antara *pwtj* dan *pwt*. Bahkan pada kombinasi tiga fitur dan dua fitur dapat dilihat bahwa penggunaan fitur tersebut menghasilkan nilai ROUGE-1 yang lebih kecil dibandingkan menggunakan fitur lain.

5.3.2 Evaluasi berdasarkan Waktu Eksekusi

Selain itu, uji coba juga dilakukan dengan mengukur waktu eksekusi masing-masing kombinasi untuk seluruh berita.

Tabel 5.4 Lama waktu eksekusi program (dalam satuan detik) tiap kombinasi fitur untuk tiap topik berita

TOPIK BERITA	LAMA WAKTU EKSEKUSI (detik) TIAP KOMBINASI FITUR *p=posisi, w=word frequency, t=tfidf, j=judul*										
	pwtj	***pwt***	***pwj***	***ptj***	***wtj***	***pw***	***pt***	***pj***	***wt***	***wj***	***tj***
1	58.611	62.921	57.114	65.169	55.764	51.504	53.929	48.504	57.308	50.092	59.809
2	35.333	27.453	25.107	27.915	28.19	27.215	28.351	24.042	27.05	24.85	28.928
3	24.832	23.303	24.433	25.115	22.905	22.072	22.541	20.436	23.568	23.302	22.804
4	30.244	28.185	27.469	30.018	30.157	26.623	25.838	25.919	28.423	24.546	27.204
5	26.384	42.805	23.453	23.793	24.976	22.045	25.556	20.907	27.579	22.081	22.85
6	29.062	26.493	22.803	23.23	23.583	22.857	23.988	22.531	25.338	28.893	23.768
7	21.614	20.307	20.012	23.18	19.509	18.195	19.299	16.858	19.22	18.693	21.051
8	16.783	16.732	16.198	18.825	17.13	17.571	16.481	14.941	17.05	16.998	16.683
9	41.057	37.637	39.223	41.214	37.898	36.606	38.398	33.63	40.22	35.419	38.617
10	37.729	36.186	32.094	34.658	41.536	30.604	34.026	31.587	36.052	30.829	34.674
11	28.073	32.132	26.505	29.66	29.832	28.818	28.748	24.206	27.796	25.93	30.958
12	20.549	19.825	18.304	18.996	18.84	17.824	18.566	17.159	19.12	16.86	19.942
13	13.461	12.948	11.94	12.484	11.809	11.427	12.323	10.421	10.707	10.853	12.143
14	83.463	78.305	75.619	88.96	83.715	78.32	86.671	72.324	80.06	77.115	78.665
15	17.998	16.9	16.025	23.267	16.465	15.184	17.27	13.946	15.428	14.728	15.895
TOTAL	**485.193**	**482.132**	**436.299**	**486.484**	**462.309**	**426.865**	**451.985**	**397.411**	**454.919**	**421.189**	**453.991**
RATA-RATA	**32.3462**	**32.14213**	**29.0866**	**32.43227**	**30.8206**	**28.45767**	**30.13233**	**26.49407**	**30.32793**	**28.07927**	**30.26607**
URUTAN	**10**	**9**	**4**	**11**	**8**	**3**	**5**	**1**	**7**	**2**	**6**

waktu tercepat waktu terlama

Dari tabel 5.4 dapat dilihat bahwa urutan rata-rata waktu eksekusi seluruh kombinasi terhadap 15 topik berita dari yang tercepat hingga yang terlambat ialah sebagai berikut.

1. *pj*
2. *wj*

3. *pw*
4. *pwj*
5. *pt*
6. *tj*
7. *wt*
8. *wtj*
9. *pwt*
10. *pwtj*
11. *ptj*

Pengukuran berdasarkan waktu eksekusi saja tentunya tidak dapat dijadikan landasan mutlak untuk mengukur performa suatu metode. Maka dari itu, untuk mengetahui kombinasi yang paling optimal untuk digunakan dalam proses pemilihan kalimat, analisis berdasarkan waktu eksekusi akan dipadukan dengan analisis berdasarkan nilai ROUGE-1.

5.4 Analisis

Tabel 5.5 Urutan kombinasi berdasarkan ROUGE-1 dan waktu eksekusi

Urutan	Nilai ROUGE-1		Waktu eksekusi (detik)	
	Kombinasi	Rata-rata	Kombinasi	Rata-rata
1	pwt	0.687	pj	26.494
2	pwtj	0.681	wj	28.079
3	pw	0.679	pw	28.458
4	pt	0.675	pwj	29.087
5	pwj	0.673	pt	30.132
6	ptj	0.672	tj	30.266
7	wt	0.553	wt	30.328
8	wtj	0.549	wtj	30.821
9	wj	0.534	pwt	32.142
10	tj	0.517	pwtj	32.346
11	pj	0.493	ptj	32.432

fitur:
p =posisi
w =word frequency
t =tfidf
j =judul

Tabel 5.5 menunjukkan urutan kombinasi berdasarkan nilai ROUGE-1 dan waktu eksekusi. Nilai ROUGE-1 diurutkan secara *descending* (terbesar - terkecil) sedangkan waktu eksekusi diurutkan secara *ascending* (tercepat – terlambat). Dari tabel tersebut dapat diambil disimpulkan

bahwa kombinasi dua fitur yakni posisi kalimat dan *word frequency* merupakan kombinasi yang optimal untuk mendapatkan ringkasan yang baik dengan waktu yang cukup cepat.

Kombinasi dua fitur yakni posisi kalimat dan *word frequency* merupakan kombinasi yang optimal disebabkan oleh hal-hal berikut.

1. Sebagian besar berita cenderung menyampaikan ide pokoknya pada awal-awal kalimat sedangkan kalimat-kalimat selanjutnya merupakan penjelas atau bahkan informasi-informasi lain di luar pokok bahasan. Sehingga dengan menggunakan fitur posisi kalimat, kita dapat mengambil intisari dari berita tersebut. Selain itu, perhitungan skor posisi kalimat juga sangat sederhana (persamaan 3.3) sehingga tidak memakan waktu eksekusi program.
2. Kalimat-kalimat berita yang dapat dijadikan sebagai ringkasan secara umum mengandung kata-kata yang sering muncul pada kumpulan dokumen.
3. Fitur TF-IDF sebenarnya merupakan fitur yang cukup penting untuk menghasilkan ringkasan yang baik. Hal ini dapat dilihat dari nilai ROUGE-1 yang tinggi ketika menggunakan fitur TF-IDF. Namun, jika dilihat berdasarkan waktu, penggunaan fitur TF-IDF cukup memakan waktu eksekusi karena dalam prosesnya fitur ini harus menghitung bobot TF-IDF tiap kata di tiap dokumen.
4. Untuk fitur kemiripan dengan judul berita, berdasarkan hasil uji coba, dapat disimpulkan bahwa fitur tersebut tidak terlalu memegang peranan penting karena dengan atau tanpa fitur tersebut nilai ROUGE-1 tidak menunjukkan perbedaan yakni antara kombinasi 4 fitur dan kombinasi fitur posisi kalimat, *word frequency*,

dan TF-IDF. Bahkan pada kombinasi tiga fitur dan dua fitur dapat dilihat bahwa penggunaan fitur tersebut menghasilkan nilai ROUGE-1 yang lebih kecil dibandingkan menggunakan fitur lain.

LAMPIRAN

Lampiran 1. Contoh satu kumpulan topik berita yang kurang baik

Topik Berita : Kunjungan Jokowi

```
<judul>Seorang Penjual Bunga di IPDN Ingatkan Jokowi Soal Hakikat Manusia</judul>
<tanggal>15/6/2015 11:07</tanggal>
<keterangan>Penjual bunga di sekitaran Kampus Institut Pemerintahan Dalam Negeri (IPDN)
Sumedang mengingatkan Jokowi tentang hakikat manusia. Ketika manusia mati, maka tidak ada
nilai lebihnya.</keterangan>
<kata_kunci>ipah,presiden,jokowi,manusia,bunga,ipdn,lokasi</kata_kunci>
<topik>Olahraga</topik>
<lokasi>Bandung</lokasi>
<penulis>Kontributor Bandung, Reni Susanti</penulis>
<editor>Glori K. Wadrianto</editor>
<tag> </tag>
<isi>
  <paragraf>Seorang penjual bunga di sekitaran Kampus Institut Pemerintahan Dalam Negeri (
  IPDN) Sumedang mengingatkan Presiden Jokowi tentang hakikat manusia. Dia menyebut,
  ketika manusia mati, maka tidak ada nilai lebihnya.</paragraf>
  <paragraf>Secara syariat, setelah jadi mayat, apakah manusia berarti? Jawabannya tidak.
  Bandingkan dengan ikan asin. Setelah dia mati, masih bisa digunakan enggak? Jawabannya
  sangat berguna, tutur Ipah, di depan kampus IPDN Sumedang, Jabar, Senin (15/6/2015).</
  paragraf>
  <paragraf>Menurut Ipah, itu artinya, selama manusia diberi nafas oleh Tuhan maka dia
  harus adil, bijaksana, dan berbuat yang terbaik dalam hidupnya. Sebab, ketika meninggal,
  manusia tidak memiliki nilai apapun.</paragraf>
  <paragraf>Apalagi pemimpin seperti Jokowi, tanggung jawabnya besar. Namun sebagai
  manusia, hakikatnya tidak akan berubah, tetap sebagai manusia. Kehidupan saat ini
  semakin sulit. Harga barang-barang semakin mahal. Kesejahteraan terus menurun. Padahal
  jika kesejahteraan masyarakat terus turun, masyarakat akan nekat, mereka akan menjadi
  beringas, tutur Ipah.</paragraf>
  <paragraf>Ipah berharap bisa sekali saja bertemu dengan Presiden. Bertahan-tahun ia
  berjualan bunga di IPDN, ia tidak pernah melihat sosok Presiden. Padahal, Presiden mana
  pun kerap bolak-balik ke IPDN.</paragraf>
  <paragraf>Minimal kalau wisuda Presiden datang ke sini. Dari jaman Bu Mega dulu, lanjut
  Pak SBY, dan sekarang Pak Jokowi, saya belum pernah melihat mereka. Padahal lokasi kami
  berdekatan, ucap dia.</paragraf>
  <paragraf>Bahkan, untuk melihat rombongan pun, sambung Ipah, tidak bisa. Namun lokasi
  berjualan digeser polisi ke arah timur, dengan alasan keamanan. Dalam wisuda IPDN,
  berbagai pernak-pernik dari batu akik hingga bunga ramai dijajakan. Untuk bunga, sengaja
  dibawa langsung dari Lembang. Harganya Rp10.000-25.000 per ikat.</paragraf>
```

```
<judul>Senangnya Luar Biasa, Saya Didadahi Pak Jokowi</judul>
<tanggal>15/6/2015 11:23</tanggal>
<keterangan>Setelah menghadiri acara wisuda mahasiswa Institut Pemerintahan Dalam Negeri
(IPDN), Presiden Joko Widodo dijadwalkan akan mengunjungi Gudang Bulog Jabar di Cimindi,
Cimahi.</keterangan>
<kata_kunci>pukul,cimindi,titin,ipdn,lambaian tangan,jokowi,presiden</kata_kunci>
<topik>Olahraga</topik>
<lokasi>Sumedang</lokasi>
<penulis>Kontributor Bandung, Reni Susanti</penulis>
<editor>Glori K. Wadrianto</editor>
<tag>Jokowi</tag>
<isi>
  <paragraf>Setelah menghadiri acara wisuda mahasiswa Institut Pemerintahan Dalam Negeri
  (IPDN), Presiden Joko Widodo dijadwalkan akan mengunjungi Gudang Bulog Jabar di
  Cimindi, Cimahi.</paragraf>
  <paragraf>Rencananya, Jokowi akan pergi ke Cimindi menggunakan jalur darat. Mendengar
  kabar tersebut, warga yang sempat pergi karena Jokowi datang ke IPDN menggunakan
  helikopter, kembali mendekati gerbang timur kampus.</paragraf>
  <paragraf>Mereka berdiri dan menanti datangnya Jokowi yang dikabarkan meninggalkan
  IPDN sekitar pukul 10.30-11.00 WIB. Mbak, saya pengin banget ketemu Presiden.
  Alhamdulillah sekarang berkesempatan ketemu, ujar Titin (35), warga Lembang, Bandung
  Barat, Senin (15/6/2015).</paragraf>
  <paragraf>Titin mengaku pergi dari rumahnya pukul 4.00 WIB. Ia memilih waktu subuh
  agar jalanan belum macet. Setibanya di Jatinangor, pukul 05.30 WIB, ia sempat membantu
  saudaranya yang kebetulan berjualan aksesoris di IPDN.</paragraf>
  <paragraf>Pas Pak Jokowi mau datang, saya lari dan nunggu di dekat gerbang. Tapi Pak
  Jokowi naik helikopter. Tapi saya yakin bisa melihat beliau, jadinya saya tunggu saja
  di tempat yang teduh, tutur dia.</paragraf>
  <paragraf>Ternyata perkiraannya benar. Jokowi melanjutkan kunjungan perjalanan
  dinasnya melalui jalur darat. Titin merasa senang karena bisa berdiri paling depan.
  Senangnya luar biasa mbak. Saya didadahi Pak Jokowi. Pak Jokowi lihat saya. Akhirnya
  dapat lambaian tangan Pak Jokowi, tutur dia.</paragraf>
  <paragraf>Di mata Titin, Jokowi adalah Presiden pilihan rakyat. Ia berharap, Jokowi
  bisa menjaga amanah dan menurunkan harga-harga bahan pokok yang terus naik.</paragraf>
  <paragraf>Saat ini, rombongan Presiden Jokowi sudah meninggalkan IPDN menuju Gudang
  Bulog di Cimindi, Cimahi.</paragraf>
```

Gambar A.1 Kumpulan berita dalam satu topik yang kurang baik

Kedua berita di atas meskipun memiliki satu topik yang sama yakni Kunjungan Jokowi, namun memiliki pokok masalah yang berbeda. Berita pertama membahas tentang kunjungan Jokowi ke IPDN sementara berita pertama membahas tentang kunjungan Jokowi ke Gudang Bulog. Hal ini membuat sistem peringkasan kurang tepat dalam menentukan kalimat-kalimat ringkasan yang mencakup kedua dokumen berita tersebut.

Lampiran 2. Contoh satu kumpulan topik berita yang baik

Topik Berita : BLBI

```
<judul>Selidiki BLBI, KPK Minta Keterangan Kwik Kian Gie</judul>
<tanggal>2/4/2013 20:07</tanggal>
<keterangan>KPK mulai menyelidiki indikasi tindak pidana korupsi dalam
penerbitan Surat Keterangan Lunas SKL beberapa obligor BLBI.</keterangan>
<kata_kunci>Kwik Kian Gie,BLBI</kata_kunci>
<topik>Nasional</topik>
<lokasi>Jakarta</lokasi>
<penulis>Icha Rastika</penulis>
<editor>Hindra</editor>
<tag> </tag>
<isi>
  <paragraf>Komisi Pemberantasan Korupsi mulai menyelidiki indikasi
  tindak pidana korupsi dalam penerbitan Surat Keterangan Lunas (SKL)
  beberapa obligor Bantuan Likuiditas Bank Indonesia (BLBI). Pada Selasa
  (2/4/2013), lembaga antikorupsi itu meminta keterangan mantan Menteri
  Koordinator Bidang Perekonomian Kwik Kian Gie.</paragraf>
  <paragraf>Kwik Kian Gie dimintai keterangan terkait KPK melakukan
  penyelidikan dalam kaitan dengan dugaan terjadinya TPK (tindak pidana
  korupsi) dalam lanjutan penyelesaian BLBI yaitu pemberian SKL, kata
  Juru Bicara KPK Johan Budi melalui pesan singkat, Selasa malam.</
  paragraf>
  <paragraf>Menurut Johan, KPK mulai melakukan penyelidikan BLBI baru-
  baru ini. Dia belum tahu apakah Kwik merupakan pihak yang pertama kali
  dimintai keterangan terkait penyelidikan ini atau bukan.</paragraf>
  <paragraf>Adapun Kwik enggan mengungkapkan apa yang disampaikannya
  kepada penyelidik KPK selama dimintai keterangan sembilan jam. Mantan
  Kepala Badan Perencanaan dan Pembangunan Nasional (Bappenas) itu hanya
  mengatakan Jadi betul-betul rahasia. Undangannya rahasia dan
  pertanyannya juga rahasia.</paragraf>
  <paragraf>Kasus BLBI ini pernah diusut KPK saat Antasari Azhar menjadi
  ketua KPK sekitar 2008. Saat itu Antasari mengatakan, KPK menaruh
  perhatian jika ada oknum atau pejabat yang melakukan penyimpangan
  dalam penerbitan SKL tersebut.</paragraf>
  <paragraf>Menurut Antasari, jika ada proses SKL ada yang tidak sesuai
  ketentuan, KPK akan merekomendasikan agar kasus BLBI itu dibuka kembali
  . Namun, KPK juga tidak bisa serta merta mencabut Surat Perintah
  Penghentian Perkara (SP3) yang dikeluarkan kejaksaan. KPK, kata
  Antasari, akan menjadikan fakta persidangan kasus dugaan suap jaksa
  Urip Tri Gunawan sebagai salah satu bahan pengusutan.</paragraf>
  <paragraf>Penyimpangan BLBI</paragraf>
  <paragraf>Kejaksaan Agung saat dipimpin MA Rachman menerbitkan SP3
```

```
<judul>Selidiki SKL BLBI, KPK Panggil Rizal Ramli</judul>
<tanggal>12/4/2013 11:55</tanggal>
<keterangan>KPK memanggil mantan Menteri Koordinator Bidang
Perekonomian, Rizal Ramli terkait penyelidikan atas penerbitan SKL BLBI.<
/keterangan>
<kata_kunci>rizal ramli,jakarta,BLBI</kata_kunci>
<topik>Nasional</topik>
<lokasi>Jakarta</lokasi>
<penulis>Icha Rastika</penulis>
<editor>Inggried Dwi Wedhaswary</editor>
<tag>Rizal Ramli'</tag>
<isi>
  <paragraf>Komisi Pemberantasan Korupsi (KPK) memanggil mantan Menteri
  Koordinator Bidang Perekonomian, Rizal Ramli terkait penyelidikan atas
  penerbitan Surat Keterangan Lunas (SKL) beberapa obligor Bantuan
  Likuiditas Bank Indonesia (BLBI), Jumat (12/4/2013). Rizal memenuhi
  panggilan KPK sekitar pukul 10.00 WIB.</paragraf>
  <paragraf>Terkait penyelidikan kasus BLBI, kata Rizal di Gedung KPK,
  Kuningan, Jakarta. Seperti diketahui, KPK memulai penyelidikan SKLI
  BLBI baru-baru ini. Pada Selasa (2/4/2013), lembaga antikorupsi itu
  meminta keterangan mantan Menteri Koordinator Bidang Perekonomian Kwik
  Kian Gie.</paragraf>
  <paragraf>Kasus BLBI ini pernah diusut KPK saat Antasari Azhar menjadi
  ketua KPK sekitar 2008. Saat itu Antasari mengatakan, KPK menaruh
  perhatian jika ada oknum atau pejabat yang melakukan penyimpangan
  dalam penerbitan SKL tersebut.</paragraf>
  <paragraf>Menurut Antasari, jika ada proses SKL ada yang tidak sesuai
  ketentuan, KPK akan merekomendasikan agar kasus BLBI itu dibuka kembali
  . Namun, KPK juga tidak bisa serta merta mencabut Surat Perintah
  Penghentian Perkara (SP3) yang dikeluarkan kejaksaan. KPK, kata
  Antasari, akan menjadikan fakta persidangan kasus dugaan suap jaksa
  Urip Tri Gunawan sebagai salah satu bahan pengusutan.</paragraf>
  <paragraf>Penyimpangan BLBI</paragraf>
  <paragraf>Kejaksaan Agung saat dipimpin MA Rachman menerbitkan SP3
  terhadap 10 tersangka kasus BLBI pada 2004. Korupsi BLBI merupakan
  salah satu perkara korupsi terbesar yang pernah ada. Hasil audit BPK
  menyebutkan, dari Rp 144,5 triliun dana BLBI yang dikucurkan kepada 48
  bank umum nasional, Rp 138,4 triliun dinyatakan merugikan negara.
  Penggunaan dana-dana tersebut kurang jelas.</paragraf>
  <paragraf>Selain itu, terdapat penyimpangan dalam penyaluran maupun
  penggunaan dana BLBI yang dilakukan pemegang saham, baik secara
```

```
<judul>Selidiki BLBI, KPK Minta Keterangan Laksamana Sukardi</judul>
<tanggal>11/6/2013 12:06</tanggal>
<keterangan>KPK meminta keterangan mantan Menteri Badan Usaha Milik
Negara, Laksamana Sukardi, Selasa 11/6/2013, terkait penyelidikan kasus
BLBI.</keterangan>
<kata_kunci> </kata_kunci>
<topik>Nasional</topik>
<lokasi>Jakarta</lokasi>
<penulis>Icha Rastika</penulis>
<editor>Inggried Dwi Wedhaswary</editor>
<tag>Rizal Ramli'</tag>
<isi>
  <paragraf>Komisi Pemberantasan Korupsi (KPK) meminta keterangan mantan
  Menteri Badan Usaha Milik Negara, Laksamana Sukardi, Selasa (11/6/2013)
  , terkait penyelidikan atas penerbitan surat keterangan lunas (SKL)
  beberapa obligor Bantuan Likuiditas Bank Indonesia (BLBI).</paragraf>
  <paragraf>Laksamana Sukardi dimintai keterangan terkait penyelidikan
  penerbitan SKL dalam menyelesaikan BLBI, kata Juru Bicara KPK Johan
  Budi.</paragraf>
  <paragraf>Adapun Laksamana diketahui sudah tiba di Gedung KPK,
  Kuningan, Jakarta Selatan, pagi tadi. Laksamana, yang mengenakan jaket
  hitam, terpantau wartawan saat tengah menunggu di lobi Gedung KPK.</
  paragraf>
  <paragraf>KPK meminta keterangan Laksamana Sukardi karena dianggap
  tahu seputar mekanisme penerbitan SKL. Mekanisme penerbitan SKL
  dikeluarkan Badan Penyehatan Perbankan Nasional (BPPN) berdasarkan
  Inpres No 8 Tahun 2002 saat kepemimpinan Presiden Megawati
  Soekarnoputri yang mendapat masukan dari mantan Menteri Keuangan
  Boediono, Menko Perekonomian Dorodjatun Kuntjoro-Jakti, dan Laksamana
  Sukardi.</paragraf>
  <paragraf>SKL tersebut berisi tentang pemberian jaminan kepastian
  hukum kepada debitor yang telah menyelesaikan kewajibannya atau
  tindakan hukum kepada debitor yang tidak menyelesaikan kewajibannya
  berdasarkan penyelesaian kewajiban pemegang saham, dikenal dengan
  inpres tentang release and discharge. Tercatat beberapa nama
  konglomerat papan atas, seperti Sjamsul Nursalim, The Nin King, dan
  Bob Hasan, yang telah mendapatkan SKL dan sekaligus release and
  discharge dari pemerintah.</paragraf>
  <paragraf>Terkait penyelidikan SKL ini, KPK sudah memeriksa
  Dorodjatun, Menteri Keuangan dan Koordinator Perekonomian periode 2000-
  2001 Rizal Ramli, Menteri Keuangan 1998-1999 Bambang Subiyanto, Menko
```

```
<judul>Selidiki BLBI, KPK Panggil Mantan Menteri Rini Soewandi</judul>
<tanggal>25/6/2013 10:41</tanggal>
<keterangan>KPK memanggil mantan Menteri Perindustrian dan Perdagangan
Rini Mariani Soewandi untuk dimintai keterangan terkait kasus BLBI.</
keterangan>
<kata_kunci> </kata_kunci>
<topik>Nasional</topik>
<lokasi>Jakarta</lokasi>
<penulis>Icha Rastika</penulis>
<editor>Caroline Damanik</editor>
<tag>Rizal Ramli'</tag>
<isi>
  <paragraf>Komisi Pemberantasan Korupsi (KPK) memanggil mantan Menteri
  Perindustrian dan Perdagangan Rini Mariani Soewandi untuk dimintai
  keterangan terkait penyelidikan atas penerbitan surat keterangan lunas
  (SKL) beberapa obligor Bantuan Likuiditas Bank Indonesia (BLBI),
  Selasa (25/6/2013).</paragraf>
  <paragraf>Dimintai keterangan terkait penyelidikan KPK soal SKL, kata
  Juru Bicara KPK Johan Budi. Adapun Rini tiba di Gedung KPK, Kuningan,
  Jakarta, sekitar pukul 09.55 WIB dengan didampingi seorang pria yang
  tampak seperti kerabatnya. Rini yang terlihat mengenakan setelan jas
  berwarna yang dipadu dengan atasan merah menyala itu tidak berkomentar
  kepada wartawan.</paragraf>
  <paragraf>KPK meminta keterangan Rini karena dia dianggap tahu seputar
  proses pemberian SKL kepada sejumlah obligor BLBI. Mekanisme
  penerbitan SKL dikeluarkan Badan Penyehatan Perbankan Nasional (BPPN)
  berdasarkan Inpres No 8 Tahun 2002 saat kepemimpinan Presiden Megawati
  Soekarnoputri yang mendapat masukan dari mantan Menteri Keuangan
  Boediono, Menko Perekonomian Dorodjatun Kuntjoro-Jakti, dan Laksamana
  Sukardi.</paragraf>
  <paragraf>SKL tersebut berisi tentang pemberian jaminan kepastian
  hukum kepada debitor yang telah menyelesaikan kewajibannya atau
  tindakan hukum kepada debitor yang tidak menyelesaikan kewajibannya
  berdasarkan penyelesaian kewajiban pemegang saham, dikenal dengan
  inpres tentang release and discharge. Tercatat beberapa nama
  konglomerat papan atas, seperti Sjamsul Nursalim, The Nin King, dan
  Bob Hasan, yang telah mendapatkan SKL dan sekaligus release and
  discharge dari pemerintah.</paragraf>
  <paragraf>Terkait penyelidikan SKL ini, KPK sudah meminta keterangan
  Laksamana Sukardi, Dorodjatun, Menteri Keuangan dan Koordinator
  Perekonomian periode 2000-2001 Rizal Ramli, Menteri Keuangan 1998-1999
```

Gambar A.2 Kumpulan berita dalam satu topik yang baik

Keempat berita di atas tergabung dalam satu topik berita yakni BLBI. Meskipun isi berita berbeda namun pokok masalahnya sama yakni membahas tentang penyelidikan BLBI dengan memanggil beberapa tokoh yang terlibat. Hal ini membuat sistem mudah dalam menentukan kalimat-

kalimat ringkasan yang merepresentasikan seluruh dokumen berita.

BAB VI
KESIMPULAN DAN SARAN

Pada bab terakhir ini dijelaskan mengenai kesimpulan yang didapat setelah melakukan serangkaian uji coba. Dalam bab ini juga dikemukakan pula saran pengembangan sistem lebih lanjut.

6.1 Kesimpulan

Berdasarkan uji coba, didapatkan kesimpulan bahwa diantara empat kombinasi fitur yakni fitur posisi kalimat, *word frequency*, TF-IDF, dan judul berita, kombinasi yang paling optimal berdasarkan nilai ROUGE-1 dan waktu eksekusi adalah kombinasi fitur posisi kalimat dan *word frequency.*

6.2 Saran

Adapun saran untuk pengembangan lebih lanjut dari proses peringkasan multi-dokumen dalam Tugas Akhir ini ialah dilakukan pengembangan lebih lanjut agar tingkat akurasi yang dihasilkan bisa lebih baik yaitu dengan cara mencari tahu nilai parameter-parameter yang optimal contohnya parameter *threshold* jumlah kata yang dimasukkan ke dalam *WFList*.

DAFTAR PUSTAKA

Fachrurrozi, M., Yusliani, N., & Yoanita, R. U. (2013). Frequent Term based Text Summarization for Bahasa Indonesia. *International Conference on Innovations in Engineering and Technology (ICIET'2013)* . Bangkok (Thailand).

Ferreira, R., Cabral, L. d., Lins, R. D., e Silva, G. P., & Freitas, F. (2013). Assessing sentence scoring techniques for extractive text summarization. *Expert Systems with Applications, 40*, 5755–5764.

Ferreira, R., Freitas, F., Cabral, L. d., Lins, R. D., Lima, R., Franc¸a, G., . . . Favaro, L. (2014). A Context Based Text Summarization System. *11th IAPR International Workshop on Document Analysis Systems.* IEEE.

Holi, M. H. (2006). Integrating tf-idf Weighting With Fuzzy View based Search. *Proceedings of the ECAI Workshop on Text-Based Information Retrieval (TIR-06).* Riva del Garda, Italy.

Karel J., J. S. (2008). Automatic Text Summarization (The State of The Art 2007 and New Challenges). *Znalosti* (hal. 1-12). Ústav informatiky a softvérového inziinierstva: FIIT STU Bratislava.

Lin, C. Y. (2004). ROUGE: a Package for Automatic Evaluation of Summaries. *In Proceedings of Workshop on Text Summarization Brances Out* (hal. 74-81). Barcelona: Association for Computational Linguistics.

Radev, D. R., Hovy, E. H., & McKeown, K. (2002). Introduction to the Special Issue on Summarization. *Computational Linguistics, 28*(4), 399-408.

Salton, G., & Buckley, C. (1988). TERM-WEIGHTING APPROACHES IN AUTOMATIC TEXT RETRIEVAL. *Information Processing & Management, 24*, 513-523.

Kavita-Ganesan (2016). *ROUGE 2.0 Documentation - Java Package for Evaluation of Summarization Tasks* [Online].Tersedia:http://kavita-ganesan.com/content/rouge-2.0-documentation [18 Juli 2016]

Mei, J.-P., & Chen, L. (2012). SumCR: A new subtopic-based extractive approach for text summarization. *Knowl Inf Syst (2012), 31*, 527–545

BIODATA PENULIS

Satrio Verdianto, dilahirkan di kota Jakarta pada tanggal 8 Juni 1993. Penulis adalah anak pertama dari tiga bersaudara. Penulis memulai pendidikan formal di SDN 37 Ampenan Mataram (2001-2003), SDN Kranji 1 Purwokerto (2003-2005), SMPN 1 Palembang (2005-2008), SMAN 1 Palembang (2008-2009) , SMAN 3 Bandung (2009-2011) dan terakhir sebagai mahasiswa Teknik Informatika ITS Surabaya (2011-2016).

Selama menjadi mahasiswa, penulis aktif berorganisasi di Himpunan Mahasiswa Teknik Computer – Informatika (HMTC) ITS. Penulis sempat menjabat sebagai staff Departemen Dalam Negeri HMTC periode 2012-2013.

Penulis yang hobi menyanyi ini mengambil bidang minat Komputasi Cerdas dan Visualisasi (KCV) sebagai bidang keahliannya, karena penulis memiliki ketertarikan pada dunia sistem cerdas. Penulis dapat dihubungi melalui email: satriov@gmail.com