**OTOMATISASI PERINGKASAN TEKS PADA DOKUMEN HUKUM MENGGUNAKAN METODE *LATENT SEMANTIC ANALYSIS***

PROPOSAL SKRIPSI

Oleh:

MILLENIA RUSBANDI NIM. 1641720029



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

**OTOMATISASI PERINGKASAN TEKS PADA DOKUMEN HUKUM MENGGUNAKAN METODE *LATENT SEMANTIC ANALYSIS***

**PROPOSAL SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

Oleh:

MILLENIA RUSBANDI NIM. 1641720029



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

HALAMAN PENGESAHAN

**OTOMATISASI PERINGKASAN TEKS PADA DOKUMEN HUKUM MENGGUNAKAN METODE  *SEMANTIC ANALYSIS***

Disusun oleh:

MILLENIA RUSBANDI NIM. 1641720029

**Proposal Skripsi ini telah diuji pada 16 Januari 2020**

**Disetujui oleh:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Penguji I | : | Pramana Yoga S, S.Kom., M.MT.  NIP. 19880504 201504 1 004 | ........................ |
|  |  |  |  |  |
| 2. | Penguji II | : | Dika Rizky Yunianto, S.Kom., M.Kom.  NIP. 19920606 201903 1 017 | ........................ |
|  |  |  |  |  |
| 3. | Pembimbing | : | Imam Fahrur Rozi, S.T., M.T.  NIP. 19840610 200812 1 004 | ........................ |

Mengetahui,

|  |  |
| --- | --- |
| Ketua Jurusan  Teknologi Informasi | Ketua Program Studi  Teknik Informatika |
| Rudy Ariyanto, S.T., M.CS  NIP. 19711110 199903 1 002 | Imam Fahrur Rozi, ST., MT.  NIP. 198406102008121004 |

DAFTAR ISI

HALAMAN JUDUL ii

HALAMAN PENGESAHAN iii

DAFTAR ISI iv

[DAFTAR GAMBAR v](#_Toc30576202)

[DAFTAR TABEL vi](#_Toc30576203)

[1. Judul Skripsi 1](#_Toc30576204)

[2. Latar Belakang 1](#_Toc30576205)

[3. Rumusan Masalah 2](#_Toc30576206)

[4. Batasan Masalah 2](#_Toc30576207)

[5. Tujuan 2](#_Toc30576208)

[6. Landasan Teori 3](#_Toc30576209)

[6.1 Penelitian Terdahulu 3](#_Toc30576210)

[6.2 *Text Mining* 4](#_Toc30576211)

[6.3 Peringkasan Teks Otomatis 4](#_Toc30576212)

[6.4 Dokumen Hukum 5](#_Toc30576213)

[6.5 *Latent Semantic Analysis* 5](#_Toc30576214)

[7. Metodologi Penelitian 6](#_Toc30576215)

[7.1 Data 6](#_Toc30576216)

[7.2 Metode Pengambilan Data 7](#_Toc30576217)

[7.3 Metode Pengolahan Data 7](#_Toc30576218)

[***7.3.1*** ***Pre-Processing*** 7](#_Toc30576219)

[***7.3.2*** ***Processing*** 10](#_Toc30576220)

[7.4 Metode Pengembangan Perangkat Lunak 15](#_Toc30576221)

[**7.4.1** **Analisis Kebutuhan** 15](#_Toc30576222)

[**7.4.2** **Perancangan** 16](#_Toc30576223)

[**7.4.3** **Implementasi** 21](#_Toc30576224)

[**7.4.4** **Pengujian Unit** 21](#_Toc30576225)

[**7.4.5** **Pemeliharaan** 21](#_Toc30576226)

[7.5 Metode Pengujian 21](#_Toc30576227)

[8. Jadwal Kegiatan 22](#_Toc30576228)

[DAFTAR PUSTAKA 24](#_Toc30576229)

DAFTAR GAMBAR

[Gambar 7.1 Contoh Data yang Diolah : Nota Pembelaan 6](#_Toc30575340)

[*Gambar 7.2 Tahapan Metode Waterfall* 15](#_Toc30575341)

[Gambar 7.3 Use Case Diagram 16](#_Toc30575342)

[Gambar 7.4 Desain Sistem 16](#_Toc30575343)

[Gambar 7.5 Alur Proses Sistem 17](#_Toc30575344)

[Gambar 7.6 Flowchart Preprocessing 17](#_Toc30575345)

[Gambar 7.7 Flowchart Perhitungan Pembobotan kata menggunakan TF-IDF 18](#_Toc30575346)

[Gambar 7.8 Flowchart Metode LSA 19](#_Toc30575347)

[Gambar 7.9 Activity Diagram Peringkasan Dokumen 20](#_Toc30575348)

[Gambar 7.10 Activity Diagram Pengujian Hasil Ringkasan 20](#_Toc30575349)

DAFTAR TABEL

[Table 7.1 Pembentukan Kalimat 7](#_Toc30575533)

[Table 7.2 Case Folding 8](#_Toc30575534)

[Table 7.3 Tokenizing 9](#_Toc30575535)

[Table 7.4 Stopword Removal 9](#_Toc30575536)

[Table 7.5 Stemming 9](#_Toc30575537)

[Table 7.6 Contoh Perhitungan DF 11](#_Toc30575538)

[Table 7.7 Contoh Perhitungan TF-IDF 11](#_Toc30575539)

[Table 7.8 Hasil Perhitungan Matriks ATA (X) 12](#_Toc30575540)

[Table 7.9 Hasil Perhitungan X-λI 13](#_Toc30575541)

[Table 7.10 Hasil Perhitungan det |X-λI| (Eigen Value) 13](#_Toc30575542)

[Table 7.11 Matriks VT 14](#_Toc30575543)

[Table 7.12 Matriks S 14](#_Toc30575544)

[Table 7.13 Hasil skor perhitungan Sk 14](#_Toc30575545)

# Judul Skripsi

Otomatisasi Peringkasan Teks pada Dokumen Hukum Menggunakan Metode Latent Semantic Analysis

# Latar Belakang

Selama periode tahun 2015–2017, jumlah kasus kriminalitas di Indonesia cenderung fluktuatif (tidak stabil). Berdasarkan data Kepolisian Republik Indonesia (Polri), total jumlah kasus kriminalitas pada tahun 2015 sebanyak 352.936 kasus. Pada tahun 2016, kasus semakin meningkat menjadi sebanyak 357.197 kasus. Sedangkan pada tahun 2017, mengalami penurunan menjadi 336.652 kasus (Badan Pusat Statistik, 2018). Dari data tersebut, rata-rata jumlah kriminalitas periode tahun 2015-2017 sebesar 348.928 kasus. Banyaknya jumlah kriminalitas di Indonesia akan berdampak pada jumlah dokumen hukum yang akan ditangani oleh aparat penegak hukum.

Dalam pemahaman dokumen hukum, aparat penegak hukum seperti advokat, hakim, maupun jaksa harus membaca keseluruhan dokumen dimana akan memerlukan waktu yang cukup lama. Kapasitas yang dimiliki manusia untuk mengonsumsi informasi sangat terbatas. Oleh karena itu, sangatlah penting untuk menyaring informasi yang hanya berguna dari jumlah data tidak terstuktur. Sehingga salah satu solusi yang dibutuhkan yaitu dengan membuat sebuah ringkasan pada dokumen hukum dimana dokumen tersebut berbentuk pdf. Dokumen hukum yang akan diringkas berupa nota pembelaan dan bagian yang diolah yaitu pendahuluan. Dilakukannya peringkasan pada bagian tersebut dikarenakan berbentuk paragraf dan dapat diringkas. Sehingga kalimat-kalimat yang tidak diperlukan akan dihilangkan.

Ringkasan adalah teks yang dihasilkan dari sebuah kumpulan teks yang mengandung informasi penting dengan panjang teks tidak lebih dari setengah teks aslinya (Radev, 2002). Dalam dunia komputer, peringkasan juga dikenal dengan Peringkasan teks Otomatis atau ATS (*automatic text summarization*). Peringkasan teks adalah pembuatan bentuk yang lebih singkat dari suatu teks dengan memanfaatkan aplikasi yang dijalankan dan dioperasikan pada komputer (Nugraha, 2008). Dengan adanya ringkasan tersebut, dapat mempermudah aparat penegak hukum dalam memahami dokumen tanpa harus membaca keseluruhan isi dokumen.

Pada penelitian yang telah dilakukan, umumnya menggunakan objek berupa artikel dan berita. Sedangkan untuk dokumen hukum, masih belum banyak dilakukan. Oleh karena itu, dalam penelitian ini mengusulkan peringkasan teks pada dokumen hukum dengan menggunakan metode latent semantic analysis (LSA) diharapkan dapat menghasilkan ringkasan kalimat dengan baik dan sesuai dengan isi dari dokumen hukum tersebut.

# Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang dapat diambil adalah sebagai berikut :

1. Bagaimanakah membaca dan mengubah dokumen hukum dalam format pdf ke dalam teks yang dapat diolah oleh sistem peringkasan otomatis.
2. Bagaimanakah pengimplementasian metode LSA (*Latent Semantic Analysis*) padaperingkasan dokumen hukum.
3. Bagaimanakah pengujian yang dilakukan untuk mengukur akurasi peringkasan dokumen otomatis.

# Batasan Masalah

Agar skripsi ini dapat berjalan sesuai dengan rencana dan tujuan awal, maka penulis memberikan batasan-batasan masalah yaitu :

1. Data yang digunakan yaitu data berupa dokumen hukum pembelaan kasus pada Kantor Advokat Ahmad Riyadh Umar Balhmar, Surabaya, Jawa Timur, Indonesia.
2. Dokumen yang diringkas berformat pdf.
3. Bagian yang digunakan untuk peringkasan yaitu pada pendahuluan.
4. Peringkasan yang digunakan yaitu secara ekstraktif.
5. Sistem ini berbasis website.

# Tujuan

Berdasarkan latar belakang dan rumusan masalah di atas, maka tujuan dari skripsi ini yaitu:

Membantu pihak aparat hukum dalam melakukan peringkasan dokumen.

Menerapkan metode *latent semantic analysis* pada ringkasan dokumen hukum.

Membaca dokumen hukum dalam format pdf dan mengubahnya ke dalam teks yang dapat diolah oleh sistem peringkasan otomatis.

Menganalisis performasi hasil ringkasan dari metode tersebut berupa akurasi berdasarkan *precision*, *recall* dan *f-measure*.

# Landasan Teori

Landasan teori merupakan bagian yang akan membahas tentang penyelesaian masalah yang akan memberikan jalan keluarnya. Dalam hal ini akan dikemukakan beberapa teori-teori yang berkaitan dengan masalah yang diangkat.

## Penelitian Terdahulu

Berdasarkan penelitian pada tahun 2014, Agustinus Widiantoro telah membangun sebuah aplikasi peringkasan dokumen berbahasa jawa secara otomatis menggunakan metode TF-IDF (*Term Frequency-Inverse Document Frequency*). Dari penelitian tersebut, hasil yang didapatkan menunjukan bahwa tingkat keakurasian sistem mencapai 64%. Tetapi, kelemahan dari sistem ini yaitu kurangnya daftar kata umum (*stopwords*) yang digunakan sehingga penulis menyarankan untuk melakukan penambahan daftar kata umum untuk meningkatkan akurasi (Widiantoro, 2014).

Pada tahun 2015, penelitian dilakukan oleh [Geetha J.K.](https://ieeexplore.ieee.org/author/37085643357)dan [Deepamala N](https://ieeexplore.ieee.org/author/37085510587). dengan judul “*Kannada text summarization using Latent Semantic Analysis*”. Penelitian tersebut menggunakan Bahasa Kanada. Tingkat akurasi yang didapatkan yaitu 94% dan *precision* sebesar 80% (N & K, 2015).

Penelitian berikutnya dilakukan oleh [Tinaliah](https://www.researchgate.net/scientific-contributions/2138998456_Tinaliah_Tinaliah) dan [Triana Elizabeth](https://www.researchgate.net/scientific-contributions/2138988319_Triana_Elizabeth) pada tahun 2018. Peneliti menggunakan metode *Latent Semantic Analysis* dan *Jaro-Winkler Distance* dalam mendeteksi plagiarisme dokumen. Tingkat akurasi pada penelitian tersebut menghasilkan nilai plagiat mencapai 97,14% (Tinaliah & Elizabeth, 2018).

Dari penelitian yang telah dipaparkan diatas, Metode *Latent Semantic Analysis* dan Metode TF-IDF dapat diterapkan pada proses peringkasan dokumen dengan tingkat akurasi yang cukup baik. Maka dari itu, pada penelitian ini penulis menggunakan dua metode tersebut dalam melakukan peringkasan teks. Untuk objek yang akan digunakan yaitu berupa dokumen hukum.

## *Text Mining*

*Text mining* memiliki definisi menambang data yang berupa teks dimana sumber data di dapatkan dari dokumen. Tujuan *text mining* adalah mencari kata-kata yang dapat mewakili isi dari dokumen dan dilakukannya analisa keterhubungan antar dokumen.

Jenis masukan (*input*) dari *text mining* berupa data tak terstruktur yang merupakan pembeda utama dari *data mining* dimana menggunakan data terstruktur atau basis data sebagai masukan. *Text mining* dapat dianggap sebagai proses dua tahap yang diawali dengan penerapan struktur terhadap sumber data teks dan dilanjutkan dengan ekstraksi informasi dan pengetahuan yang relevan dari data teks terstruktur ini dengan menggunakan teknik dan alat yang sama dengan *data mining*. Proses yang umum dilakukan oleh text mining, yaitu perangkuman otomatis, kategorisasi dokumen, penggugusan teks, dll (Nindito, 2016).

## Peringkasan Teks Otomatis

Peringkasan teks otomatis merupakan pembuatan rangkuman dari suatu teks secara otomatis dengan menggunakan serta memanfaatkan sistem peringkasan teks yang dijalankan pada komputer. Sebuah sistem peringkasan teks diberi *input* (masukan) berupa teks kemudian sistem akan memproses dengan melakukan peringkasan yang akan menghasilkan *output* (keluaran) berupa teks yang lebih singkat dari sumber teks aslinya (Hovy, 2005).Terdapat dua pendekatan peringkas teks yaitu:

1. Ekstraksi (*extractive summary*)

Pada teknik ekstraksi, sistem menyalin unit-unit yang dianggap paling penting dari sebuah teks dan diubah menjadi ringkasan. Unit-unit teks yang disalin dapat berupa klausa utama, kalimat utama, atau paragraf utama tanpa ada penambahan kalimat-kalimat baru yang terdapat pada dokumen aslinya.

1. Abstraksi (*abstractive summary*)

Teknik abstraksi menggunakan metode *linguistic* untuk memeriksa dan menafsirkan teks menjadi ringkasan. Ringkasan teks tersebut dihasilkan dengan cara menambahkan kalimat-kalimat baru yang merepresentasikan intisari teks sumber ke dalam bentuk yang berbeda dengan kalimat-kalimat yang ada pada teks sumber (Gupta & Lehal, 2010).

Pada penelitian ini, metode yang digunakan untuk melakukan peringkasan teks otomatis adalah dengan menggunakan teknik ekstraksi. Hasil dari ringkasan merupakan kalimat asli yang terdapat pada dokumen dan tidak mengalami penambahan kalimat.

## Dokumen Hukum

Pengertian dokumen hukum secara singkat adalah bentuk rekaman yang dapat dijadikan alat bukti. Rekaman tersebut beraneka ragam bentuknya, namun umumnya berupa surat. Bentuk lain dari dokumen misalnya rekaman suara, video, atau notulensi. Dokumen hukum digunakan untuk mendukung keterangan akan suatu keadaan sehingga posisi keadaan lebih meyakinkan. Keberadaan dokumen sangat penting karena terbatasnya kemampuan manusia. Nilai dokumen dapat berarti sangat tinggi sesuai dengan kepentingan informasi yang dibawanya. Salah satu bentuk dokumen yang sangat penting adalah barang bukti kejahatan di pengadilan. Dokumen ini dapat membuat seseorang bebas atau mendekam di penjara selama sisa hidupnya. Dokumen ini juga secara tidak langsung turut menentukan lama hukuman yang bisa dijatuhkan pada seseorang (Pengertian Dokumen : Fungsi Dan Kegunaan Dokumen, 2017).

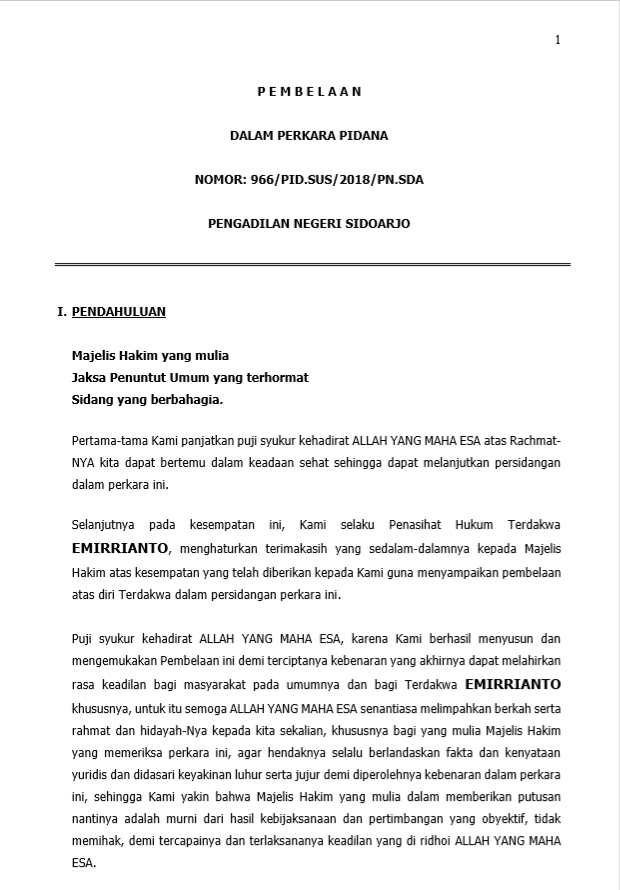
## *Latent Semantic Analysis*

*Latent Semantic Analysis* (LSA) menurut bahasa terbagi atas beberapa kata yang penting yaitu *latent* dan *semantic*, *latent* yang memiliki arti tersembunyi atau sesuatu yang masih belum terlihat, sedangkan semantic berasal dari bahasa yunani “*semanticos*” yang berarti memberi tanda, penting atau cabang linguistik yang mempelajari arti dan makna dari suatu bahasa, kode atau jenis representasi lainnya. Dari pengertian dapat ditarik kesimpulan bahwa, LSA adalah menguraikan atau menganalisa makna yang masih tersembunyi dari suatu bahasa, kode atau jenis representasi lainya, guna memperoleh informasi yang penting. Kesamaan kata dan kalimat diperoleh dengan cara menggunakan *Singular Value Decomposition* (SVD), di mana SVD mempunyai kapasitas untuk mereduksi *noise*, sehingga dapat meningkatkan hasil akurasi pada ringkasan (Peter & Kp, 2009).

# Metodologi Penelitian

## Data

Data yang diolah berupa dokumen hukum yang didapatkan dari Kantor Advokat Ahmad Riyadh Umar Balhmar, Surabaya. Data yang akan diolah tersebut memiliki format pdf. Dibawah ini merupakan contoh nota pembelaan yang akan digunakan dalam peringkasan teks otomatis.



Gambar 7.1 Contoh Data yang Diolah : Nota Pembelaan

## Metode Pengambilan Data

Metode pengumpulan data yang digunakan untuk menyusun penelitian ini adalah sebagai berikut:

1. Observasi.

Dalam pengambilan data, Penulis langsung terjun ke lapangan Kantor Advokat Ahmad Riyadh Umar Balhmar Kota Surabaya untuk melakukan observasi.

1. Wawancara

Penulis melakukan kegiatan tanya jawab dengan pihak Advokat Kota Surabaya.

## Metode Pengolahan Data

Data atau kalimat yang akan digunakan untuk pengujian metode preprocessing dan metode pengambilan ringkasan ditunjukan pada paragraf dibawah ini.

“Emmirianto memosting status tersebut hanya dalam waktu kurang dari 1 hari dan belum berdampak bagi masyarakat. Lalu ia melakukan perbuatan tersebut atas dasar keinginan sendiri dan hal tersebut merupakan bentuk emosi dari terdakwa yang akhirnya mengalahkan akal sehatnya. Sehingga terdakwa tidak kontrol dan merasa jengkel.”

### ***Pre-Processing***

Pada tahapan ini, data tekstual akan diubah menjadi teks agar dapat diolah oleh sistem. Penelitian ini menggunakan dokumen sebagai inputan awal. Proses yang digunakan antara lain :

1. Pembentukan Kalimat

Pembentukan kalimat yaitu pemecahan teks dokumen menjadi kumpulan kalimat berdasarkan delimiter.

Table 7.1 Pembentukan Kalimat

|  |  |
| --- | --- |
| No | Pembentukan Kalimat |
| 1 | Emmirianto memosting status tersebut hanya dalam waktu kurang dari 1 hari dan belum berdampak bagi masyarakat. |
| 2 | Lalu ia melakukan perbuatan tersebut atas dasar keinginan sendiri dan hal tersebut merupakan bentuk emosi dari terdakwa yang akhirnya mengalahkan akal sehatnya. |
| 3 | Sehingga Terdakwa tidak kontrol dan merasa jengkel. |

1. *Case Folding*

*Case folding* merupakan pengubahan huruf pada kalimat menjadi huruf kecil (*lowercase*) dan penghilangan karakter yang tidak valid seperti tanda baca.

Table 7.2 Case Folding

|  |  |
| --- | --- |
| No | *Case Folding* |
| 1 | emmirianto memosting status tersebut hanya dalam waktu kurang dari hari dan belum berdampak bagi masyarakat |
| 2 | lalu ia melakukan perbuatan tersebut atas dasar keinginan sendiri dan hal tersebut merupakan bentuk emosi dari terdakwa yang akhirnya mengalahkan akal sehatnya |
| 3 | sehingga terdakwa tidak kontrol dan merasa jengkel |

1. *Tokenizing*

Pada proses ini, kalimat tersebut dipecah kembali menjadi beberapa kata tunggal penyusunnya.

Table 7.3 Tokenizing

|  |  |  |  |
| --- | --- | --- | --- |
| *Tokenizing* | | | |
| emmirianto | belum | keinginan | mengalahkan |
| memosting | berdampak | sendiri | akal |
| status | bagi | dan | sehatnya |
| tersebut | masyarakat | hal | sehingga |
| hanya | lalu | tersebut | terdakwa |
| dalam | Ia | bentuk | tidak |
| waktu | melakukan | emosi | kontrol |
| kurang | perbuatan | dari | dan |
| dari | tersebut | terdakwa | merasa |
| hari | atas | yang | jengkel |
| dan | dasar | akhirnya | Dll. |

1. *Stopword Removal*

*Stopword removal* adalah proses penghilangan kata-kata yang tidak merepresentasikan isi dokumen.

Table 7.4 Stopword Removal

|  |  |  |  |
| --- | --- | --- | --- |
| *Stopword Removal* | | | |
| emmirianto | berdampak | dasar | emosi |
| memosting | masyarakat | bentuk | terdakwa |
| status | perbuatan | mengalahkan | Dll. |

1. *Stemming*

*Stemming* adalah proses pengembalian kata tunggal yang memiliki imbuhan menjadi kata dasar.

Table 7.5 Stemming

|  |  |  |  |
| --- | --- | --- | --- |
| *Stemming* | | | |
| emmirianto | dampak | dasar | emosi |
| posting | masyarakat | bentuk | dakwa |
| status | buat | kalah | Dll. |

### ***Processing***

#### **TF-IDF**

Setelah dokumen diproses dengan cara *preprocessing, tokenizing, filtering* dan *stemming*, selanjutnya dilakukan proses pembobotan kata. Pada Metode ini pembobotan kata dalam sebuah dokumen dilakukan dengan mengalikan nilai TF dan IDF.

*Term frequency* (TF) adalah pengukuran yang paling sederhana dalam metode pembobotan. Pada metode ini, masing-masing *term* diasumsikan mempunyai proporsi kepentingan sesuai jumlah kemunculan dalam teks dokumen. *Term frequency* dapat memperbaiki nilai *recall* pada information retrieval, tetapi tidak selalu memperbaiki nilai *precision* (Tokunaga & Iwayama, 1994). Hal ini disebabkan *term* yang *frequent* cenderung muncul di banyak teks, sehingga *term* tersebut memiliki kekuatan

*Inverse document frequency* (IDF) adalah metode pembobotan term yang lebih condong (fokus) untuk memperhatikan kemunculan *term* pada keseluruhan kumpulan teks. Pada IDF, *term* yang jarang muncul pada keseluruhan koleksi teks dinilai lebih berharga. Nilai kepentingan tiap *term* diasumsikan berbanding terbalik dengan jumlah teks yang mengandung *term* tersebut (Tokunaga & Iwayama, 1994).

*Term frequncy inverse document frequency* (TF•IDF) adalah metode pembobotan yang menggabungkan metode TF dan IDF. Metode ini diusulkan oleh Salton sebagai sebuah kombinasi metode yang dapat memberikan performansi yang lebih baik, khususnya dalam memperbaiki nilai *recall* dan *precision* (Tokunaga & Iwayama, 1994). Berikut ini merupakan perhitungannya :

*tf.idf* = *tf* \* log(*N*/*df*) (7.1)

Keterangan :

*Tf* : Jumlah *term* tersebut  
*N* : Total dokumen  
*df* : Jumlah dokumen yang mengandung suatu *term*

Table 7.6 Contoh Perhitungan DF

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kata penting | Frekuensi Kata pada Kalimat | | | | df |
| **Kalimat 1** | **Kalimat 2** | **Kalimat 3** |  | |
| emmirianto | 1 | 1 | 0 | 2 | |
| posting | 1 | 0 | 0 | 1 | |
| status | 1 | 0 | 0 | 1 | |
| 1 | 1 | 0 | 0 | 1 | |
| dampak | 1 | 0 | 0 | 1 | |
| masyarakat | 1 | 0 | 0 | 1 | |
| buat | 0 | 1 | 0 | 1 | |
| dasar | 0 | 1 | 0 | 1 | |
| bentuk | 0 | 1 | 0 | 1 | |
| emosi | 0 | 1 | 1 | 2 | |
| dakwa | 0 | 1 | 0 | 1 | |
| kalah | 0 | 1 | 0 | 1 | |
| sehat | 0 | 1 | 0 | 1 | |
| kontrol | 0 | 0 | 1 | 1 | |
| jengkel | 0 | 0 | 1 | 1 | |

Table 7.7 Contoh Perhitungan TF-IDF

|  |  |  |
| --- | --- | --- |
| W | | |
| tf\*idf | | |
| 1.176091 | 1.176091 | 0 |
| 1.477121 | 0 | 0 |
| 1.477121 | 0 | 0 |
| 1.477121 | 0 | 0 |
| 1.477121 | 0 | 0 |
| 1.477121 | 0 | 0 |
| 0 | 1.477121 | 0 |
| 0 | 1.477121 | 0 |
| 0 | 1.477121 | 0 |
| 0 | 1.477121 | 0 |
| 0 | 1.176091 | 1.176091 |
| 0 | 1.477121 | 0 |
| 0 | 1.477121 | 0 |
| 0 | 0 | 1.477121 |
| 0 | 0 | 1.477121 |

#### ***Latent Semantic Analysis***

Adapun langkah-langkah LSA sebagai berikut (Mandar & Gunawan, 2017):

1. Membentuk matriks Amn.

𝐴 = 𝑈𝑆𝑉𝑇 (7.2)

A adalah matriks dokumen yang mewakili kalimat atau kata yang dikenal dengan matriks Amn. Dari perhitungan TF-IDF, dilakukan pembentukan matriks A dengan cara memasukkan nilai TF-IDF. Selanjutnya dilakukan perhitungan menggunakan metode *Singular Value Decomposition* (SVD). Seperti pada rumus 7.2. Akan didapat maktriks U mendiskripsikan matriks orthogonal 𝑚 × 𝑚 yang dikenal dengan istilah *Left Singular Vector* *. Right Singular Vektor* (V) merupakan matriks orthogonal 𝑛 × 𝑛 yang diperoleh dari *eigen vector* matriks ATA, sedangkan matriks diagonal S dihasilkan dari *eigen value* matriks ATA yang diakarkan.

Table 7.8 Hasil Perhitungan Matriks ATA (X)

|  |  |  |
| --- | --- | --- |
| Kalimat 1 | Kalimat 2 | Kalimat 3 |
| 12,29262666 | 1,38319065 | 0 |
| 1,38319065 | 15,85770451 | 1,38319065 |
| 0 | 1,38319065 | 5,746965052 |

1. Perhitungan *Eigen Value*

Setelah dilakukan perhitungan ATA, dilanjutkan dengan melakukan perhitungan *eigen value*. Dengan Rumus :

det |X-λI| (7.3)

Table 7.9 Hasil Perhitungan X-λI

|  |  |  |
| --- | --- | --- |
| Kalimat 1 | Kalimat 2 | Kalimat 3 |

|  |  |  |
| --- | --- | --- |
| 12,2926266553501 - λ | 1,38319065 | 0 |
| 1,38319065 | 15,8577045061218 - λ | 1,38319065 |
| 0 | 1,38319065 | 5,74696505191634 - λ |

Polinomial yang didapat dari det |X-λI| yaitu -λ3 + 33,8973 λ2 – 352,885 λ + 1085,76. Dengan akar akar persamaan sebagai berikut :

Table 7 .10 Hasil Perhitungan det |X-λI| (Eigen Value)

|  |  |
| --- | --- |
| λ | Hasil |
| λ1 | 16,491 |
| λ2 | 11,85 |
| λ3 | 5,556 |

Dari masing-masing *eigen value* dilakukan perhitungan untuk mencari *eigen vector*-nya dan dilanjutkan dengan normalisasi. Rumus normalisasi *eigen vector* :

===  (7.4)

Dari hasil normalisasi tersebut disatukan membentuk matriks V dan di *transpose*-kan. Seperti contoh dibawah ini.

Table 7.11 Matriks VT

|  |  |  |
| --- | --- | --- |
| Kalimat 1 | Kalimat 2 | Kalimat 3 |
| 0,310572622 | 0,942769607 | 0,121367684 |
| -0,950134866 | 0,304126009 | 0,068928272 |
| 0,028072398 | -0,136722469 | 0,990211547 |

1. Membentuk matriks S

Pembentukan mastriks S yaitu dengan cara mengurutkan nilai tertinggi *eigen value* kemudian diakarkan. Hasil matriks dapat dilihat pada tabel dibawah ini.

Table 7.12 Matriks S

|  |  |  |
| --- | --- | --- |
| Matriks S | | |
| 4,060911228 | 0 | 0 |
| 0 | 3,442382896 | 0 |
| 0 | 0 | 2,357116883 |

1. Hasil ringkasan ditentukan berdasarkan skor tertinggi dari perhitungan *length* pada setiap nilai matriks VT dengan menggunakan rumus :

Sk = (7.5)

Di mana 𝑆𝑘 adalah panjang vektor k pada kalimat yang dimodifikasi oleh laten vektor. n adalah jumlah ruang demensi baru. Hasil dari length terbesar pada setiap dokumen kalimat akan dijadikan ringkasan. Hasil dari perhitungan dapat dilihat pada tabel dibawah ini.

Table 7.13 Hasil skor perhitungan Sk

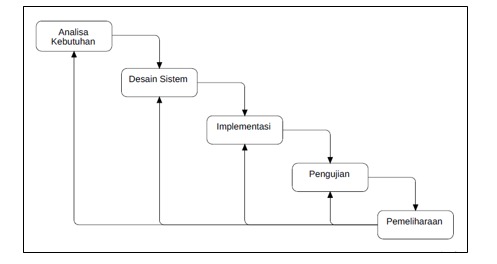
|  |  |
| --- | --- |
|  | Hasil Skor |
| Kalimat 1 | 1,575819146 |
| Kalimat 2 | **1,954901801** |
| Kalimat 3 | 1,668110957 |

Didapatkan hasil bahwa kalimat 2 akan dijadikan sebagai ringkasan dikarenakan memiliki hasil skor tertinggi.

## Metode Pengembangan Perangkat Lunak

Pengembangan perangkat lunak dapat diartikan sebagai proses membuat suatu perangkat lunak baru untuk menggantikan perangkat lunak lama secara keseluruhan atau memperbaiki perangkat lunak yang telah ada. Agar lebih cepat dan tepat dalam mendeskripsikan solusi dan mengembangkan perangkat lunak   juga hasilnya mudah dikembangkan dan dipelihara, maka pengembangan perangkat lunak memerlukan suatu metodologi khusus. Metodologi pengembangan perangkat lunak adalah suatu proses pengorganisasian kumpulan metode dan konvensi notasi yang telah didefinisikan untuk mengembangkan perangkat lunak.

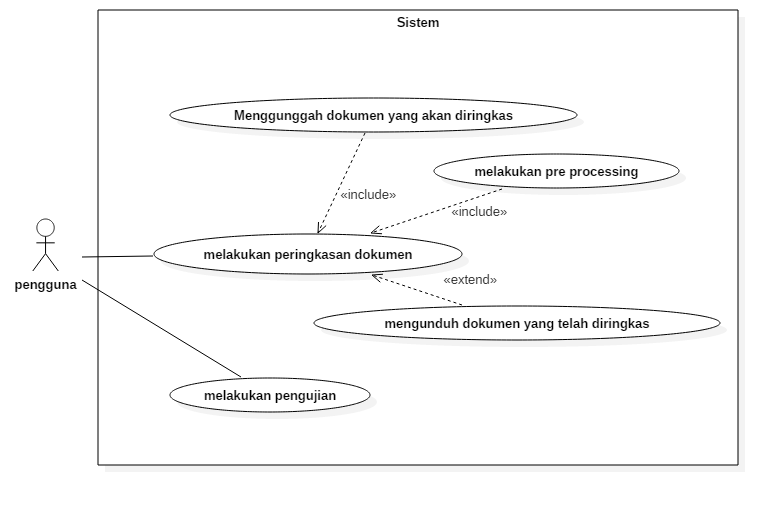
Pada penelitian ini, penulis menggunakan metode waterfall dalam pengembangan perangkat lunak. Berikut adalah tahapan – tahapan dalam metode waterfall :



*Gambar 7.2 Tahapan Metode Waterfall*

### **Analisa Kebutuhan**

Berikut kebutuhan yang telah diidentifikasi digambarkan dalam bentuk Diagram *Use Case.*



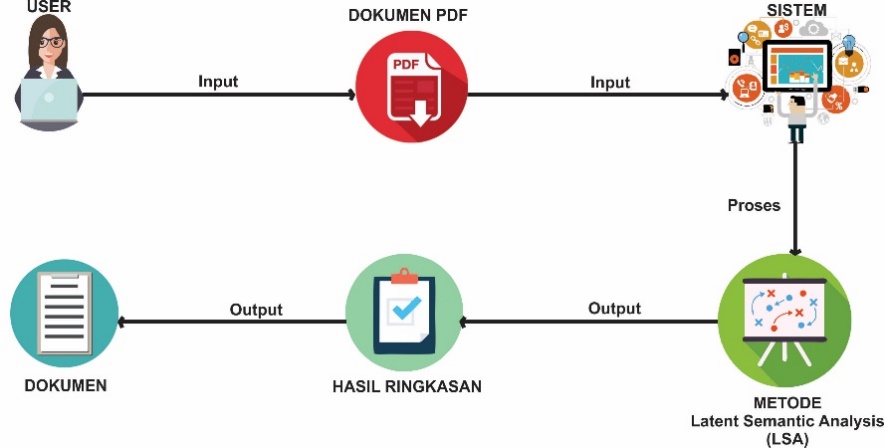
Gambar 7.3 Use Case Diagram

Dapat dilihat pada gambar 7.2, pengguna dapat melakukan peringkasan dokumen dimana harus meginputkan dokumen yang akan diringkas dan melakukan *pre-processing*. Sedangkan untuk mengunduh dokumen yang telah diringkas tidak diharuskan, pengguna dapat melihat langsung pada aplikasi hasil ringkasan tersebut. Pengguna juga dapat melakukan pengujian ringkasan yang telah dilakukan oleh sistem.

### **Perancangan**

#### **Desain Sistem**

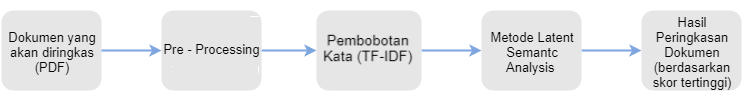
#### Berikut merupakan desain sistem peringkasan teks pada dokumen :



Gambar 7.4 Desain Sistem

#### ***Flowchart***

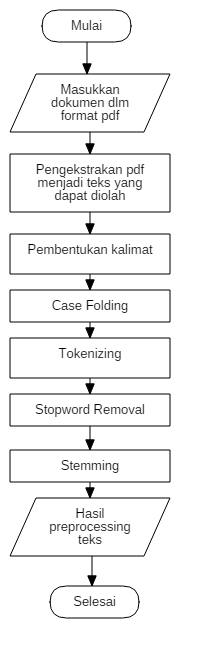
Pada gambar 7.5, dijelaskan alur proses sistem secara keseluruhan. Tahapan pertama yaitu mengunggah dokumen yang akan diringkas dalam format PDF. Selanjutnya sistem akan melakukan *pre-processing*. Kemudian dilakukan pembobotan kata menggunakan TF-IDF dan perhitungan nilai semantik menggunakan *Latent Semantic Analysis*. Hasil peringkasan akan didapatkan berdasarkan skor tertinggi dari hasil perhitungan tersebut.



Gambar 7.5 Alur Proses Sistem

Dari alur proses sistem secara keseluruhan yang telah dijelaskan diatas, dirancang beberapa *Flowchart* untuk menjelaskan masing-masing tahapan yang ada pada sistem peringkasan dokumen. Tahapannya yaitu :

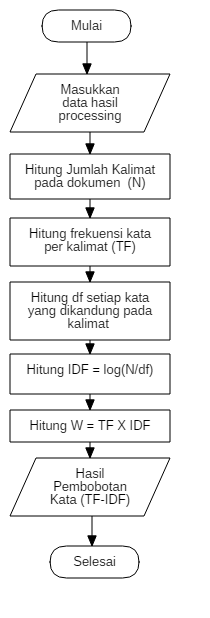
Tahap *Pre Processing*



Gambar 7.6 Flowchart Preprocessing

Pada alur proses diatas, dijelaskan langkah-langkah dalam pre-processing data yang akan diringkas.

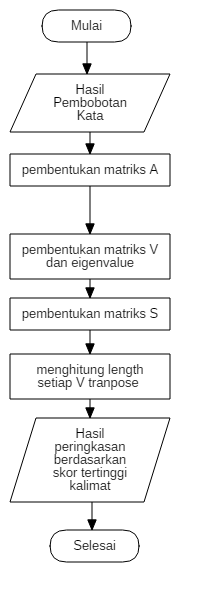
Tahap Perhitungan Pembobotan Kata



Gambar 7.7 Flowchart Perhitungan Pembobotan kata menggunakan TF-IDF

Pada alur proses diatas, dijelaskan langkah-langkah dalam perhitungan pembobotan kata menggunakan TF-IDF.

Tahap Implementasi Metode *Latent Semantic Analysis*



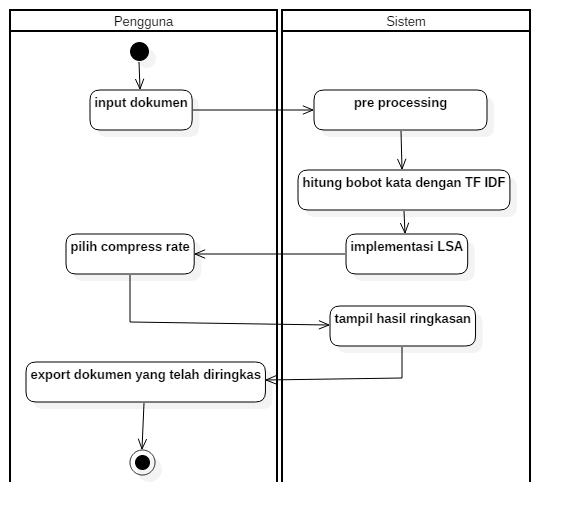
Gambar 7.8 Flowchart Metode LSA

Pada alur proses diatas, dijelaskan langkah-langkah dalam pengimplementasian metode LSA untuk peringkasan teks pada dokumen.

#### ***Activity Diagram***

Berikut adalah activity diagram yang dirancang untuk untuk menjelaskan alur proses antara pengguna dan sistem :

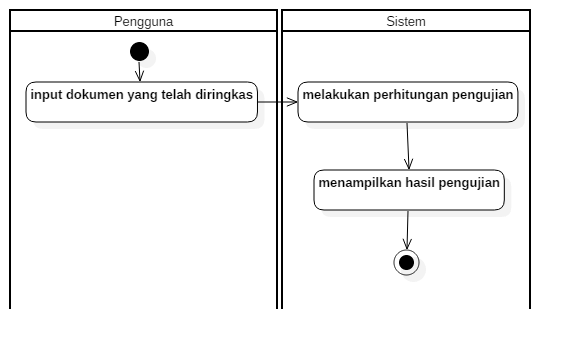
* + - 1. Alur Peringkasan Dokumen



Gambar 7.9 Activity Diagram Peringkasan Dokumen

Pada alur peringkasan teks pada dokumen seperti gambar diatas, proses dimulai dengan pengguna (*user*) memasukkan atau mengunggah file dengan format pdf. Kemudian sistem melakukan *pre-processing*. Setelah teks diolah maka akan melakukan perhitungan pembobotan kata lalu dilanjutkan dengan pengimplementasian metode LSA oleh sistem. *User* dapat memilih *compress rate* hasil ringkasan. Berikutnya, sistem akan menghasilkan hasil ringkasan sesuai dengan *compress rate* yang telah dipilih. *User* dapat mengunduh hasil ringkasan dari sistem.

* + - 1. Alur Pengujian Ringkasan



Gambar 7.10 Activity Diagram Pengujian Hasil Ringkasan

Pada alur pengujian hasil ringkasan seperti gambar diatas, proses dimulai dengan pengguna (*user*) memasukkan atau mengunggah file hasil ringkasan. Kemudian sistem akan melakukan perhitungan pengujian dan akan menampilkannya saat proses telah selesai.

### **Implementasi**

Tahap ini merupakan tahap setelah dilakukannya perancangan sistem. Perancangan yang telah dibuat akan diterjemahkan menjadi kode pemrograman dengan menggunakan bahasa php berbasis website.

### **Pengujian Unit**

Untuk pengujian unit digunakan *black box testing. Black Box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. Pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek  setiap kegagalan maupun kesalahan.

### **Pemeliharaan**

Pada tahap pemeliharaan sistem, aplikasi yang telah dijalankan akan di lakukan perbaikan jika terdapat kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

## Metode Pengujian

Untuk pengujian pada ringkasan dapat dilakukan dengan mencari nilai *recall*, *precision* dan *f-measure*. *Recall* ialah kemampuan untuk mengambil peringkat teratas yang sebagian besar relevan (benar). *Precision* adalah berapa banyak dokumen yang berhasil diambil oleh sistem, sedangkan untuk mengukur kualitas *recall* dan *precision* menggunakan *f-measure*.

# Jadwal Kegiatan

**Tabel 1. Jadwal Kegiatan**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Kegiatan | Januari | | | | Februari | | | | | Maret | | | | April | | | | Mei | | | | Juni | | | | Juli | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| **1** | **Studi Literatur** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Mempelajari pustaka sesuai topik |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Mempelajari konsep sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Mempelajari metode |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** | **Analisis** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Pengambilan Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Analisis Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Analisis Kebutuhan Fungsional |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Analisis Kebutuhan Non Fungsional |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** | **Perancangan Sistem** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Desain Flowchart |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Desain DFD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Desain ERD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Pembuatan Mockup Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** | **Implementasi Sistem** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Pembuatan UI |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Pembuatan Perhitungan Metode |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **5** | **Pengujian Sistem** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Pengujian Fungsional Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Pengujian Performa Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **6** | **Analisa Hasil Laporan dan Kesimpulan** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Pembuatan Laporan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

Badan Pusat Statistik. (2018). *Statistik Kriminal 2018.* Badan Pusat Statistik.

Gupta, V., & Lehal, G. S. (2010). A Survey of Text Summarization Extractive. *JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 2, NO. 3*.

Hovy, E. (2005). Automated Text Summarization. *Handbook of Computation Linguistics OxfordUniversityPress*.

Mandar, G., & Gunawan. (2017). Peringkasan dokumen berita Bahasa Indonesia menggunakan Cross Latent Semantic Analysis. *Jurnal Ilmiah Teknologi Sistem Informasi 3 (2) 94-104*.

N, D., & K, G. J. (2015). Kannada text summarization using Latent Semantic Analysis. *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.

Nindito, H. (2016, Desember 15). *Teori Text Mining dan Web Mining*. Retrieved from sis.binus.ac.id: https://sis.binus.ac.id/2016/12/15/teori-text-mining-dan-web-mining/

*Pengertian Dokumen : Fungsi Dan Kegunaan Dokumen*. (2017, Agustus 9). Retrieved from Ngelmu.co: https://www.ngelmu.co/pengertian-dokumen/

Peter, R., & Kp, S. (2009). Evaluation of SVD and NMF Methods for Latent Semantic Analysis. *International Journal of Recent Trends in Engineering ,Vol 1, No. 3*.

Radev, D. R. (2002). Introduction to the special issue on summarization. *Journal*, 399-408.

Sigit. (2018, September 28). *Manajemen Proyek: Penerapan Metode Waterfall*. Retrieved from medium: https://medium.com/skyshidigital/manajemen-proyek-penerapan-metode-waterfall-7c047cd2fd1f

Tinaliah, & Elizabeth, T. (2018). Perbandingan Hasil Deteksi Plagiarisme Dokumen dengan Metode Jaro-Winkler Distance dan Metode Latent Semantic Analysis. *Jurnal Teknologi dan Sistem Komputer, 6(1)*.

Tokunaga, T., & Iwayama, M. (1994). *Text Categorization based on Weighted Inverse Document Frequency.* Tokyo, Japan: Tokyo Institute of Technology.

Widiantoro, A. (2014). Peringkasan Teks Otomatis Pada Dokumen Berbahasa Jawa Menggunakan Metode TF-IDF.