

PERANCANGAN APLIKASI PENDETEKSI KEMIRIPAN TEKS DENGAN MENGGUNAKAN METODE LATENT SEMANTIC ANALYSIS

Berlin Ong¹, Dali S. Naga², Viny Christanti M.³

^{1,2,3}Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, Jl. Letjen S.Parman No.1, Jakarta, 11440, Indonesia
E-mail :¹barline1@gmail.com, ²dalinaga@gmail.com, ³viny@untar.ac.id

ABSTRAK

Sistem aplikasi Pendeteksi Kemiripan Teks adalah aplikasi yang menggunakan Metode Latent Semantic Analysis (LSA). Aplikasi ini dapat memberikan informasi mengenai hubungan antar dokumen teks yang direpresentasikan menggunakan term dokumen matrix. Term dokumen matrix berguna untuk memberikan nilai kemiripan pada setiap kata dalam dokumen. Metode LSA tidak memperhatikan tata letak kata pada dokumen dengan kata lain makna yang terkandung di dalam teks tidak berpengaruh terhadap perhitungan kemiripan. Bila terdapat dua buah dokumen yang saling salin namun struktur kalimatnya telah diubah dan ketika dibandingkan menggunakan metode LSA dengan perhitungan kemiripan menggunakan metode cosine similarity maka akan didapat hasil presentase kemiripan yang sama. Aplikasi ini dibangun menggunakan bahasa pemrograman PHP dan juga bahasa pemrograman python. Aplikasi ini dilakukan pengujian menggunakan metode blackbox testing.

Kata Kunci: *Consine Similarity, Latent Semantic Analysis, Pendeteksi Kemiripan, Term Document Matrix*

Abstract

Text Similarity Detection is an application that uses the Latent Semantic Analysis (LSA) Method. This application can provide information about the relation between text documents that are represented using the term matrix document. The term matrix document is useful for giving a similarity value to each word in a document. The LSA method ignore structure of words in documents in other words the meaning contained in the text does not apply to the calculation of similarity. If there are two documents that copy each other, but the sentence structure has been changed and compiled using the LSA method with the similarity calculation using the cosine similarity method, then the result will be determined beforehand the same similarity. This application was built using the PHP programming language and also the python programming language. This application is tested using the blackbox testing method.

Keywords— *Consine Similarity, Latent Semantic Analysis, Similarity Checker, Term Document Matrix*

1. PENDAHULUAN

Informasi dapat beredar dengan cepat dan luas dari berbagai belahan dunia melalui internet. Kemudahan akses internet yang dapat di akses di mana saja merupakan salah satu faktor utama yang menyebabkan informasi dapat dengan mudah diperoleh.

Kemudahan ini dapat dirasakan dengan adanya fitur salin dan tempel (*copy-paste*). Fitur salin dan tempel bukan merupakan fitur baru di era teknologi ini. Fitur salin dan tempel mempermudah dalam pengolahan data sesuai dengan kebutuhan misalnya untuk memenuhi tugas kuliah, membuat *paper*, dan sebagainya.

Namun karena kemudahan dalam mengakses informasi, tidak sedikit tindak kejahatan yang terjadi, contohnya adalah plagiarisme. Plagiarisme adalah penggunaan ide pemikiran milik orang lain tanpa menyebutkan sumbernya.

Menurut Peraturan Menteri Pendidikan RI Nomor 17 Tahun 2010 dikatakan bahwa plagiat adalah perbuatan sengaja atau tidak sengaja dalam memperoleh atau mencoba memperoleh kredit atau nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan atau karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai [1].

Aplikasi-aplikasi pendeteksi kemiripan dokumen telah dibuat oleh perusahaan besar seperti Turnitin, Unicheck, Copyscape dan sebagainya, namun untuk Penelitian ini dilakukan dengan menerapkan metode *Latent Semantic Analysis* dalam melakukan pendeteksian Kemiripan Teks.

Metode LSA merupakan salah satu metode untuk mendeteksi kemiripan antar abstrak [2]. Ada pula beberapa paper seperti Plagiarisme Abstrak Menggunakan Algoritma Winnowing dan Synsets[3] dan Automatic Essay Scoring dan Question Answering pada Sistem E-learning menggunakan Metode LSA dan N-Gram[4] yang dijadikan referensi dalam penelitian ini. Keunggulan LSA yaitu LSA dapat melakukan reduksi dimensi dari abstrak teks asli sehingga pada perhitungannya. Sedangkan metode seperti Rabin-karp yang juga merupakan algoritma untuk pencocokan kata dengan pola yang banyak, tidak melakukan reduksi matriks namun *Rabin-karp* menggunakan metode *Rolling Hash* yang memberikan penomoran pada setiap kata dengan panjang kata tertentu. Penelitian menggunakan LSA karena metode ini diharapkan dapat mendeteksi kemiripan antar abstrak dengan tingkat akurasi yang tinggi.

2. METODE PENELITIAN

Dalam melakukan pendeteksian pada teks, mula mula perlu melakukan pra-pemrosesan terlebih dahulu. Pra-pemrosesan berguna untuk mengidentifikasi kata-kata apa saja yang dapat diproses ke dalam perhitungan. Dalam penelitian ini digunakan menggunakan dua buah metode yaitu Latent Semantic Analysis dan Cosine Similarity.

2.1. Latent Semantic Analysis

Latent Semantic Analysis (LSA) merupakan sebuah metode yang memanfaatkan model statistik matematis untuk menganalisa struktur semantik suatu teks [5]. LSA dapat digunakan untuk memberikan nilai pada teks dengan mengkonversikan teks menjadi matriks-matriks yang diberi nilai pada masing-masing term untuk dicari kesamaan dengan term referensi. Terdapat *corpus* dalam LSA, *corpus* merupakan kumpulan teks yang memiliki kesamaan subjek. Metode LSA menerima masukan berupa file teks yang selanjutnya direpresentasikan sebagai matriks.

Latent semantic Analysis (LSA) diusulkan dan dipatenkan pada tahun 1998 oleh Scott Deerwester, Thomas Launder, dan Susan Dumais. Metode ini digunakan dalam bidang information retrieval dan Natural Language Processing [6]. Contoh penggunaan metode LSA

yaitu mencari nilai kemiripan pada karya ilmiah. Metode LSA menggunakan Teknik matematika aljabar linear yaitu Singular Value Decomposition (SVD).

Pada perhitungan Singular Value Decomposition (SVD) diperlukan *Term* matriks yang telah dibentuk sebelumnya. Tujuan dilakukan SVD yaitu untuk menemukan pola atau hubungan baru antar kata. Hasil dari perhitungan SVD menghasilkan 3 komponen matriks yaitu dua matriks orthogonal dan satu matrik diagonal. SVD dari pemfaktoran sebuah matriks A yang berdimensi t x d adalah sebagai berikut [7] :

$$A_{t \times d} = U_{t \times n} \times S_{n \times n} \times V_{d \times n}^T \quad (1)$$

Berikut langkah-langkah untuk mencari nilai U, S dan V^t dalam proses perhitungan SVD dari matriks[8]

1. Hitungan $A^T \times A$
Pada tahap awal dilakukan perkalian antara matriks A yang ditranspose dengan matriks
2. Cara determinan hasil perkalian matriks $A^T \times A$. Pada langkah kedua dilakukan determinan $A^T \times A$ untuk mencari nilai eigen berikut merupakan persamaan rumus determinan:

$$\det|A^T \times A| = 0 \quad (2)$$

$$|A^T \times A - \lambda I| = 0 \quad (3)$$

Setelah dilakukan proses determinan pada matriks $A^T \times A$ maka akan dihasilkan nilai eigen sesuai dengan banyaknya kolom. Jika matriks mempunyai tiga kolom maka nilai eigen terdapat $\lambda_1, \lambda_2, \lambda_3$. Setelah ditemukan nilai eigen atau eigen value, nilai ni dapat digunakan untuk mencari nilai singular value. Nilai singular value diperoleh dari akar dari tiap nilai eigen. Hasil nilai singular digunakan untuk membentuk matriks S dengan ketentuan $S_1 > S_2 > S_3 \dots > S_n$.

3. Hitung setiap eigen vektor V dengan mengevaluasi persamaan $|A^T \times A - \lambda I| \times X_i$ dimana nilai eigen value dihitung pada tahap sebelumnya. Hasil perhitungan menghasilkan vektor $V_1, V_2, V_3, \dots, V_n$.
4. Bentuk matriks v dengan tiap eigen vektor $V_1, V_2, V_3, \dots, V_n$. Menjadi kolom pada matriks V.
5. Bentuk matrix U dengan persamaan rumus $U = AVS^{-1}$ setelah melakukan proses SVD maka langkah selanjutnya yaitu dilakukan reduksi matriks hasil perhitungan SVD. Tujuannya untuk menghilangkan noise atau kata-kata yang dianggap tidak penting tanpa menghilangkan korelasi antar kata dengan kalimat pada dimensi pada matriks [9]. Pereduksian dilakukan dengan cara mengurangi dimensi dari matriks.

2.2. Consine Similarity

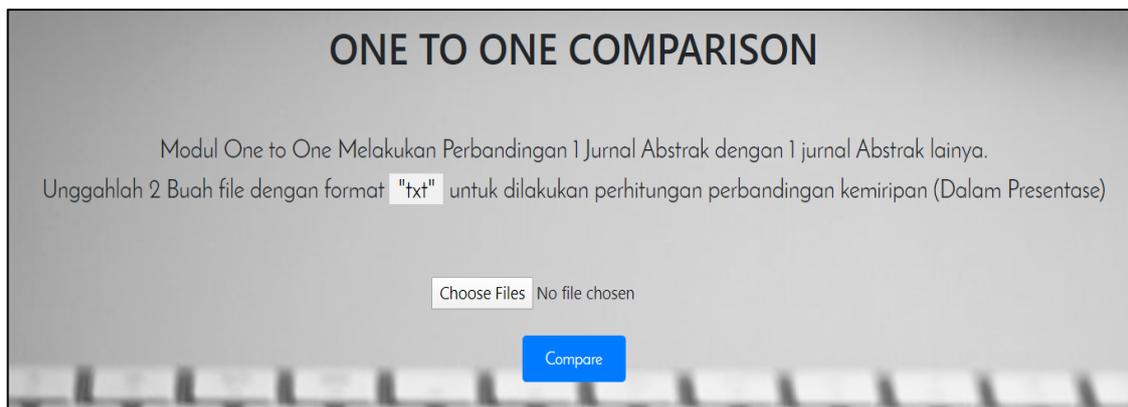
Cosine similarity digunakan untuk menghitung nilai kosinus sudut antara vektor dokumen dengan vektor kueri. Semakin kecil sudut yang dihasilkan, maka tingkat kemiripan esai semakin tinggi. Dalam hal ini vektor dokumen menunjukkan jurnal yang diunggah sedangkan vector kueri merepresentasikan jurnal yang dibandingkan. Formula dari cosine similarity adalah sebagai berikut [10]:

$$\cos(d_j, q_k) = \frac{\sum_{j=1}^t t d_{ij} \times t q_{ij}}{\sqrt{\sum_{j=1}^t (t d_{ij})^2 \times \sum_{j=1}^t (t q_{ik})^2}} \quad (4)$$

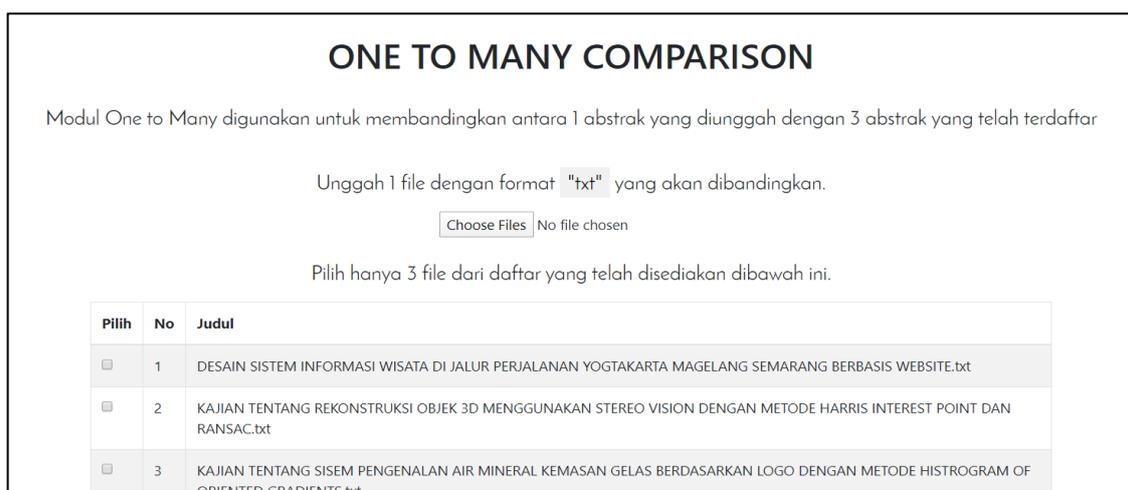
Semakin besar nilai yang dihasilkan oleh perhitungan Consine Similarity maka semakin mirip kesamaan antar abstrak itu. Nilai yang dihasilkan oleh rumus Consine Similarity yaitu berada pada rentan 0–1.

3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan dua buah modul utama yaitu modul one to one dan modul one to many. Pada modul one to one berfungsi untuk membandingkan dua buah dokumen dalam bentuk teks. Sedangkan pada modul one to many pengguna dapat melakukan membandingkan lebih dari 2 dokumen. Modul one to one dapat dilihat pada **Gambar 1** dan Modul one to many dapat dilihat pada **Gambar 2**.



Gambar 1 Modul one to one



Gambar 2 Modul one to many

Mula-mula dilakukan percobaan dengan menggunakan kata-kata yang digunakan sehari-hari, percobaan ini dilakukan untuk mengetahui tingkat keberhasilan aplikasi dalam memberikan nilai kemiripan antar kalimat. Hasil percobaan ini berada pada **Tabel 1**.

Tabel 1 Hasil percobaan menggunakan kata-kata sehari hari

Kata 1	Kata 2	Nilai Kemiripan
aku bosan makan nasi padang sepertinya hari aku mau makan bihun ayam khas pak mamat saja dibelakang rumah.	kau mau makan apa hari ini, kalo aku sepertinya hari aku mau makan bihun ayam saja.	52.98%
Budi belum makan pagi ini sehingga iya pergi mencari makan di warteg.	Budi makan nasi padang tadi pagi dan dia kini masi kenyang.	32.34%

Hasil percobaan pada **Tabel 1** dapat dikatakan bahwa aplikasi telah dapat melakukan perhitungan kemiripan antar kalimat.

Dilakukan percobaan pada aplikasi dengan membandingkan dua buah data abstrak namun data abstrak pembandingnya dikurangi datanya sebanyak 25%, 50%, dan 75%. Data abstrak yang digunakan pada penelitian ini dapat dilihat pada **Gambar 3**.

Abstrak

Tujuan dari Program Aplikasi Administrasi Claim adalah untuk membuat program yang dapat digunakan oleh karyawan dan pemilik untuk keperluan toko agar mempermudah pencatatan yang terkomputerisasi dan dapat menghasilkan laporan yang lebih akurat untuk diberikan ke pemilik toko. Metode yang digunakan dalam pembuatan perancangan ini adalah metode SDLC (System Development Life Cycle). Aplikasi ini berbasis desktop yang menghasilkan informasi mengenai sistem Claim Toko Kacamata Garuda Mas. Program aplikasi ini dibuat dengan Visual Studio 2015, dan menggunakan PhpMyadmin untuk mengolah database, dan Crystal Reports untuk membuat laporan. Aplikasi ini dapat digunakan untuk mengelola customer, barang, claim yang terjadi pada Toko Kacamata Garuda Mas. Selain mengelola data customer, barang, dan claim, aplikasi tersebut dapat menghasilkan laporan-laporan secara terkomputerisasi. Adapun laporan yang dapat dibuat oleh program ini adalah laporan penjualan, dan laporan claim. Berdasarkan uji coba yang dilakukan dapat disimpulkan bahwa sistem yang dibuat dapat dijalankan dan dapat membantu Toko Kacamata Garuda Mas dalam hal claim.

Gambar 3 Data Abtrak yang digunakan dalam percobaan

Gambar 3 merupakan data abstrak utuh dengan memuat sebanyak 153 kata didalamnya. Dari abstrak itu akan dilakukan pengujian dengan cara mengurai kata sebanyak 25%,50% dan 75%. Berikut adalah abstrak yang telah dikurangnya hingga banyak kata yang terdapat pada abstrak yang telah dikurangi pada abstrak utuh sebesar 25% adalah sebanyak 38 kata , abstrak 50% memiliki sebanyak 76 kata dan data abstrak sebanyak 75% memiliki 114 kata. Abstrak sebanyak 25%, 50% dan 75% dapat dilihat pada **Gambar 4**, **Gambar 5**, dan **Gambar 6**.

Tujuan dari Program Aplikasi Administrasi Claim adalah untuk membuat program yang dapat digunakan oleh karyawan dan pemilik untuk keperluan toko agar mempermudah pencatatan yang terkomputerisasi dan dapat menghasilkan laporan yang lebih akurat untuk diberikan ke pemilik toko. Metode

Gambar 4 Data abstrak sebanyak 25%

mengolah database, dan Crystal Reports untuk membuat laporan. Aplikasi ini dapat digunakan untuk mengelola customer, barang, claim yang terjadi pada Toko Kacamata Garuda Mas. Selain mengelola data customer, barang, dan claim, aplikasi tersebut dapat menghasilkan laporan-laporan secara terkomputerisasi. Adapun laporan yang dapat dibuat oleh program ini adalah laporan penjualan, dan laporan claim. Berdasarkan uji coba yang dilakukan dapat disimpulkan bahwa sistem yang dibuat dapat dijalankan dan dapat membantu Toko Kacamata Garuda Mas dalam hal claim.

Gambar 5 Data abstrak sebanyak 50%

Tujuan dari Program Aplikasi Administrasi Claim adalah untuk membuat program yang dapat digunakan oleh karyawan dan pemilik untuk keperluan toko agar mempermudah pencatatan yang terkomputerisasi dan dapat menghasilkan laporan yang lebih akurat untuk diberikan ke pemilik toko. Metode yang digunakan dalam pembuatan perancangan ini adalah metode SDLC (System Development Life Cycle). Aplikasi ini berbasis desktop yang menghasilkan informasi mengenai sistem Claim Toko kacamata garuda mas. Program aplikasi ini dibuat dengan Visual Studio 2015, dan menggunakan PhpMyadmin untuk mengolah database, dan Crystal Reports untuk membuat laporan. Aplikasi ini dapat digunakan untuk mengelola customer, barang, claim yang terjadi pada Toko Kacamata Garuda Mas. Selain mengelola data customer, barang, dan claim, aplikasi tersebut dapat menghasilkan laporan-laporan secara

Gambar 6 Data abstrak sebanyak 75%

Setelah dilakukan pengurangan data maka langkah selanjutnya yaitu dilakukan percobaan dengan membandingkan dua data abstrak yang utuh dan yang telah dikurangi. Hasil percobaan menggunakan data abstrak dari pendeteksian kemiripan teks dapat dilihat pada **Tabel 2**

Tabel 2 Hasil percobaan menggunakan abstrak

Banyak Data	Kemiripan
Abstrak Sebanyak 25%	32.20%
Abstrak Sebanyak 50%	67.35%
Abstrak Sebanyak 75%	95.59%

Berdasarkan hasil percobaan dengan menggunakan data abstrak sebanyak 25%, 50%, dan 75 % yang dibandingkan dengan data abstrak utuh menghasilkan akurasi yang kurang baik dengan menggunakan metode Latent Semantic Analysis.

Setelah itu dilakukan percobaan menggunakan data abstrak dari judul yang berbeda untuk mengetahui akurasi pada aplikasi. Hasil percobaan menggunakan data abstrak yang berbeda dapat dilihat pada **Tabel 3**.

Tabel 3 Hasil pengujian menggunakan data abstrak yang berbeda

Dokumen Uji 1	Dokumen Uji 2	Hasil Sistem	Hasil Sebenarnya	Kesimpulan
DESAIN SISTEM INFORMASI WISATA DI JALUR PERJALANAN YOGYAKARTA MAGELANG SEMARANG BERBASIS WEBSITE	KAJIAN TENTANG REKONSTRUKSI OBJEK 3D MENGGUNAKAN STEREO VISION DENGAN METODE HARRIS INTEREST POINT DAN RANSAC	5.23	8.0	✓
KAJIAN TENTANG SISTEM PENGENALAN AIR MINERAL KEMASAN GELAS BERDASARKAN LOGO DENGAN METODE HISTROGRAM OF ORIENTED GRADIENTS	KAJIAN TENTANG SISTEM PENGELOMPOKAN AIR MINERAL GELAS DENGAN METODE SIFT DAN CNN	7.83	13.2	X
PEMBUATAN GAME 2D ACTION ADVENTURE MAGIC MAZE ESCAPE	PEMBUATAN GAME 3D PLATFORMER SIDESCROLLING LEGEND	6.24	8.3	✓
PEMBUATAN GAME MATCH 3 BE A HERO MENGGUNAKAN UNITY	PEMBUATAN GAME PLATFORMER MULTIPLYER THE CONSTINS PROPHERCY	8.35	6.3	✓

PEMBUATAN GAME PUZZLE PICROSS THE OPERA PICROSS PADA PLATFORM ANDROID	PEMBUATAN GAME PUZZLE PLATFORMER LEGATO LETS GO TO MUSIC PADA PLATFORM PC	4.23	5.3	✓
PEMBUATAN GAME REAL TIME TACTICS BOX ARENA	PEMBUATAN GAME ROLE PLAYING TURN BASED DENGAN SISTE ROCK PAPER SCISSORS GEVAGEN	5.31	9.6	✓
PEMBUATAN GAME SIMULASI SAFETY WAY OUT	PEMBUATAN GAME STRATEGY MATCHING THREE CRAZY DONUTS	4.43	8.7	✓
PEMBUATAN GAME SURVIVAL HORRO UNLIT WORLD PADA PLATFROM WINDOWS	PEMBUATAN GAME TOP DOWN SHOOTER SURROUNDER UNTUK PLATFORM ANDROID	9.32	5.5	✓
PEMBUATAN GAME VISUAL NOVEL NOAH THE RESCUER	PEMBUATAN WEB PARTY GAME WIZARD CLASH MENGGUNKAN AIR CONSOLE	8.71	8.3	✓
PENGEMBANGAN PROGRAM APLIKASI SIMULASI UJIAN BERBASIS WEBSITE PADA SAMA CANDRA NAYA	PERANCANGAN APLIKASI E-COMMERCE JAKARTA FRUIT MARKET	7.73	8.8	✓

Pada **Tabel 3** pada kolom pertama menghasilkan kemiripan yang dihitung menggunakan sistem sebesar 5.23% sedangkan hasil sebenarnya yang merupakan hasil pemeriksaan yang dilakukan oleh pakar memberi nilai kemiripan sebesar 8% maka dapat dikatakan hasil percobaan itu dikatakan berhasil dengan diberi tanda centang dibagian kanan.

Dari hasil percobaan dari **Tabel 3**, dapat disimpulkan bahwa terdapat 9 dari 10 yang benar dalam memprediksi seberapa mirip data abstrak yang dibandingkan tersebut. Maka akurasi yang didapatkan dalam percobaan itu menggunakan Metode Latent Semantic Analysis sebesar 90%.

Berdasarkan hasil percobaan terhadap aplikasi di atas, metode Latent semantic Analysis belum dapat menghasilkan akurasi yang terbaik daripada metode yang digunakan dalam referensi sistem ini. Sistem mampu dalam mengenali kata dan kalimat, Namun Aplikasi Mendeteksi kemiripan teks menggunakan Metode Latent semantic yang dibuat ini tidak dapat melebihi akurasi dari metode yang digunakan untuk referensi.

Hal ini disebabkan sistem hanya dapat melakukan perhitungan kemiripan berdasarkan banyaknya kata dalam kalimat dan tidak mengenali urutan struktur kata pada kalimat. Alangkah baiknya kedepannya menggunakan metode gabungan yang dapat mengenali urutan struktur kata n akurasi padesehingga dapat meningkatkaa sistem.

4. KESIMPULAN

Kesimpulan yang didapat setelah dilakukan proses pengujian sistem aplikasi pendeteksi kemiripan antar teks menggunakan metode *Latent Semantic Analysis* adalah sebagai berikut:

1. Penggunaan stopword sangat mempengaruhi hasil. Sistem pendeteksi kemiripan teks ini mencari banyak kata yang sama pada teks tanpa mengenali urutan pada teks.
2. Presentase kemiripan yang paling efektif ketika dilakukan pengujian menggunakan nilai reduksi pada metode singular value decomposition sebesar dua, menggunakan Stopword, dan Stemming.
3. Tingkat akurasi pada sistem aplikasi didapatkan sebanyak 90% dengan percobaan pengujian menggunakan 10 data abstrak.

DAFTAR PUSTAKA

- [1]Wijaya, Hengki. "Plagiarisme dalam Penelitian" Sekolah Tinggi Theologia Jaffray Makassar, 2016, pp. 84-92.
- [2]G. Cosma and M. Joy, "An Approach to Source-Code Plagiarism Detection and Investigation Using Latent Semantic Analysis," in *IEEE Transactions on Computers*, vol. 61, no. 3, pp. 379-394, March 2012.doi: 10.1109/TC.2011.223
- [3]Evan Harya Chandra, Julius. Plagiarisme Abstrak menggunakan Algoritma Winnowing dan Synsets, Jakarta: Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara (Skripsi tidak dipublikasikan),2017, h. iv.
- [4]Citawan ,Rico Setiadi.Automatic Essay Scoring dan Question Answering pada Sistem E-learning menggunakan Metode LSA dan N-Gram, Jakarta: Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara (Skripsi tidak dipublikasikan), 2013, h. iv.
- [5]Suhartono, Derwin.Penggunaan Latent Semantic Analysis (LSA) dalam Pemrosesan Teks, <https://socs.binus.ac.id/2015/08/03/penggunaan-latent-semantic-analysis-lsa-dalam-pemrosesan-teks>, 22 Agustus 2019.
- [6]Ngafifudin, Dwi. <http://hirupmotekar.com/2017/05/30/dwi-ngafifudin-konsep-dan-prinsip-serta-algoritma-dalam-latent-sementic-indexing>, 21 Januari 2020
- [7]Gusmita.Penerapan Latent Semantic Analysis untuk menentukan kesamaan makna antara kata dalam bahasa inggis dan bahasa Indonesia, <http://hirupmotekar.com/2017/05/30/dwi-ngafifudin-konsep-dan-prinsip-serta-algoritma-dalam-latent-sementic-indexing/>, 8 Oktober 2019
- [8]Wall M.E., Rechtsteiner A., Rocha L.M. (2003) Singular Value Decomposition and Principal Component Analysis. In: Berrar D.P., Dubitzky W., Granzow M. (eds) *A Practical Approach to Microarray Data Analysis*. Springer, Boston, MA
- [9] Suhartono, Derwin. Penggunaan Latent Semantic Analysis (LSA) dalam Pemrosesan Teks, <https://socs.binus.ac.id/2015/08/03/penggunaan-latent-semantic-analysis-lsa-dalam-pemrosesan-teks>, 22 Agustus 2019.
- [10] Pratap Dangeti. Cosine Similarity, <https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/eb9cd609-e44a-40a2-9c3a-f16fc4f5289a.xhtml>, 18 Agustus 2019.