UNIVERSITAS INDONESIA

# ANALISA UNJUK KERJA SISTEM PENILAI ESAI OTOMATIS BERBASIS ALGORITMA GLSA *(GENERALIZED LATENT SEMANTIC ANALYSIS)* DAN PERBANDINGANNYA DENGAN ALGORITMA LSA

**SKRIPSI**

**HENRY ARTAJAYA**
**0806339130**

**FAKULTAS TEKNIK**
**PROGRAM STUDI TEKNIK KOMPUTER**
**DEPOK**
**JUNI 2012**

**UNIVERSITAS INDONESIA**

# ANALISA UNJUK KERJA SISTEM PENILAI ESAI OTOMATIS BERBASIS ALGORITMA GLSA *(GENERALIZED LATENT SEMANTIC ANALYSIS)* DAN PERBANDINGANNYA DENGAN ALGORITMA LSA

## SKRIPSI

Skripsi ini diajukan untuk melengkapi sebagian persyaratan menjadi
Sarjana Teknik

**HENRY ARTAJAYA**
**0806339130**

**FAKULTAS TEKNIK**
**PROGRAM STUDI TEKNIK KOMPUTER**
**DEPOK**
**JUNI 2012**

## HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,**

**dan semua sumber baik yang dikutip maupun dirujuk**

**telah saya nyatakan dengan benar.**

**Nama : Henry Artajaya**

**NPM : 0806339130**

**Tanda Tangan : .....**

**Tanggal : 12 Juni 2012**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Henry Artajaya

NPM : 0806339130

Program Studi : TeknikKomputer

JudulSkripsi : Analisa Unjuk Kerja Sistem Penilai Esai Otomatis Berbasis Algoritma GLSA (Generalized Latent Semantic Analysis) dan Perbandingannya dengan Algoritma LSA

**Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Komputer, Fakultas Teknik, Universitas Indonesia.**

Pembimbing : Dr. Ir. Anak Agung Putri Ratna,M.Eng. (.....)

Penguji 1 : Ir. A. Endang Sriningsih, M.T. Si. (.....)

Penguji 2 : Prima Dewi Purnamasari, S.T., M.T., M.Sc. (.....)

Ditetapkandi : Depok

Tanggal : 29Juni 2012

## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, proses penulisan skripsi ini dapat terselesaikan. Penulisan skripsi ini dilakukan dalam rangka memenuhi persyaratan dari mata kuliah Skripsi yang terdapat dalam kurikulum program studi Teknik Komputer Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan skripsi, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

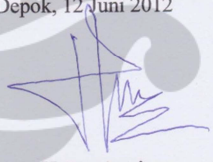
(1). Dr. Ir. Anak Agung Putri Ratna M.Eng, selaku dosen pembimbing, yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;

(2). Orang tua dan keluarga saya yang selalu memberikan dukungan moral dan semangat yang tanpanya mungkin skripsi ini tidak akan selesai;

(3). Para sahabat dan rekan-rekan yang telah banyak membantu saya dalam menyelesaikan penyusunan skripsi ini.

Akhir kata, saya berharap agar Tuhan Yang Maha Esa berkenan membalas segala kebaikan dari semua pihak yang telah membantu selesainya pengerjaan skripsi ini. Semoga skripsi ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan di masa depan.

Depok, 12 Juni 2012



Henry Artajaya

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

| | |
|---|---|
| Nama | : Henry Artajaya |
| NPM | : 0806339130 |
| Program Studi | : Teknik Komputer |
| Departemen | : Teknik Elektro |
| Fakultas | : Teknik |
| Jenis karya | : Skripsi |

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (***Non-exclusive Royalty-Free Right***) atas karya ilmiah saya yang berjudul :

**ANALISA UNJUK KERJA SISTEM PENILAI ESAI OTOMATIS BERBASIS ALGORITMA GLSA *(GENERALIZED LATENT SEMANTIC ANALYSIS)* DAN PERBANDINGANNYA DENGAN ALGORITMA LSA**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 12 Juni 2012

Yang menyatakan,

(Henry Artajaya)



vi

# ABSTRAK

Nama : Henry Artajaya
Program Studi : Teknik Komputer
Judul : ANALISA UNJUK KERJA SISTEM PENILAI ESAI OTOMATIS BERBASIS ALGORITMA GLSA (GENERALIZED LATENT SEMANTIC ANALYSIS) DAN PERBANDINGANNYA DENGAN ALGORITMA LSA

Representasi dokumen sebagai vektor GLSA pada beberapa percobaan seperti uji sinonim, klasifikasi dokumen, dan *clustering* terbukti mampu menghasilkan tingkat akurasi yang lebih baik daripada sistem sejenis yang berbasis algoritma LSA akan tetapi GLSA belum pernah diujikan pada sistem penilai esay otomatis. Percobaan ini meneliti pengaruh implementasi GLSA pada sistem penilai esay otomatis dan perbandingan unjuk kerjanya dengan sistem penilai esay otomatis berbasis LSA. Unjuk kerja sistem penilai esai otomatis berbasis GLSA lebih unggul daripada sistem berbasis LSA. Dari 60 kali pengujian, GLSA menghasilkan nilai yang lebih akurat pada 47 kali pengujian atau 78,3% total pengujian sedangkan LSA hanya unggul pada 9 kali pengujian atau 15% total pengujian dan sisanya 4 kali pengujian atau 6,7% total pengujian menghasilkan nilai dengan tingkat akurasi yang sama. Nilai Pearson Product Moment Correlation pada percobaan menggunakan sistem LSA 0.57775-0.85868 sedangkan pada GLSA sebesar 0.73335-0.76971. Hal ini mengindikasikan bahwa sistem berbasis LSA dan GLSA yang diujikan layak pakai karena memiliki performa yang sama baiknya dengan performa yang dilakukan oleh manusia. Ditinjau dari waktu proses yang dibutuhkan, LSA unggul pada soal 1 dan 2 dengan rataan 0,07466 detik dan 0,2935 detik sedangkan pada GLSA rataan waktu proses soal 1 dan 2 sebesar 1,32329 detik dan 17,3641 detik. Waktu proses yang dibutuhkan sistem penilai esai otomatis berbasis GLSA lebih lama dibandingkan dengan LSA. Akan tetapi karena GLSA menunjukkan kinerja yang amat baik, amat dipercaya bahwa manfaatnya lebih besar daripada biaya komputasi.

Kata kunci:
Generalized Latent Semantic Analysis, Sistem Penilai Esai Otomatis

# ABSTRACT

Name : Henry Artajaya
Program Study : Computer Engineering
Title : PERFORMANCE ANALYSIS OF GLSA (GENERALIZED LATENT SEMANTIC ANALYSIS) BASED AUTOMATED ESSAY GRADING SYSTEM COMPARED TO LSA BASED AUTOMATED ESSAY GRADING SYSTEM

Document representation as GLSA vectors were shown to improve performance on different tasks such as synonymy test, document classification, and clustering compared to LSA based systems, however GLSA performance has never been tested on automated essay grading system. This experiment examines the effect of GLSA implementation on automated essay grading system and evaluates its performance compared to LSA based system. GLSA performance was shown to outperform LSA based automated essay grading system. From 60 samples, GLSA outperform LSA 47 times (78,3%), LSA outperform GLSA 9 times (15%), and 4 times (6,7%) resulted the same score accuracy. Pearson Product Moment Correlation Value resulted from the experiment using LSA based system is 0.57775-0.85868 and 0.73335-0.76971 for GLSA based system. This result incidates LSA and GLSA based system used on this experiment are ready to be used as human rater replacement because both of the system deliver similar performance with human rater. Processing time of LSA based system is faster with average processing time consecutively 0,07466 second and 0,2935 second compared to GLSA consecutively 1,32329 second and 17,3641 second. GLSA requires more processing time than LSA based system because GLSA based system has more calculation steps than LSA. However GLSA showed better performance, therefore it's believed that its benefits outweigh the computational cost.

Keywords:
Generalized Latent Semantic Analysis, Automated Essay Grading System

**DAFTAR ISI**

HALAMAN COVER ..... i
HALAMAN JUDUL ..... ii
HALAMAN PERNYATAAN ORISINALITAS ..... iii
LEMBAR PENGESAHAN ..... iv
KATA PENGANTAR ..... v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS ..... vi
ABSTRAK ..... vii
DAFTAR ISI ..... ix
DAFTAR GAMBAR ..... xi
DAFTAR TABEL ..... xii
BAB 1 ..... 1
1.1 Latar Belakang ..... 1
1.2 Tujuan Skripsi ..... 2
1.3 Pembatasan Masalah ..... 2
1.4 Metodologi Penelitian ..... 2
1.5 Sistematika Penulisan ..... 2
BAB 2 ..... 4
2.1 Sistem Penilai Esai Otomatis ..... 4
2.2 Simple-O ..... 6
2.3 *Latent Semantic Analysis (LSA)* ..... 7
2.4 *Probabilistic Latent Semantic Analysis (PLSA)* ..... 13
2.5 *Generalized Latent Semantic Analysis (GLSA)* ..... 15
2.6 Perbandingan Performa Sistem Esai Grading yang Pernah Diciptakan ..... 27
BAB 3 ..... 33
3.1 Spesifikasi Sistem ..... 33
3.1.1 Spesifikasi Perangkat Keras ..... 33
3.1.2 Spesifikasi Perangkat Lunak ..... 33
3.2 Arsitektur Sistem ..... 35
3.2.1 Pembentukan Matriks Kemiripan S ..... 35

3.2.2 Pembentukan Matriks Dokumen GLSA ..... 37
3.2.3 Penilaian Dokumen ..... 38
BAB 4 ..... 40
4.1 Sistem Pengujian ..... 40
4.2 Hasil Pengujian ..... 40
4.2.1 Sistem Penilai Esai Otomatis dengan Algoritma *Latent Semantic Analysis (LSA)* ..... 43
4.2.2 Sistem Penilai Esai Otomatis dengan Algoritma *Generalized Latent Semantic Analysis (GLSA)* ..... 45
4.3 Analisa Hasil ..... 47
4.3.1 Analisis Pengujian 1 ..... 48
4.3.2 Analisis Pengujian 2 ..... 55
BAB 5 ..... 58
DAFTAR REFERENSI ..... 59

## DAFTAR GAMBAR

Gambar 2.1 Proses Reduksi Dimensi pada SVD ..... 11
Gambar 3.1 Rancangan Sistem Penilai Esai Berbasis GLSA ..... 35
Gambar 4.1 Perbandingan Nilai LSA pada Soal 1..... 48
Gambar 4.2 Perbandingan Nilai GLSA pada Soal 1..... 48
Gambar 4.3 Perbandingan Nilai LSA pada Soal 2..... 49
Gambar 4.4 Perbandingan Nilai GLSA pada Soal 2..... 49
Gambar 4.5 Perbandingan Waktu LSA dan GLSA pada Soal 1..... 55
Gambar 4.6 Perbandingan Waktu LSA dan GLSA pada Soal 2..... 56

## DAFTAR TABEL

Tabel 2.1 Perbandingan Performa yang Dihasilkan oleh Beberapa Sistem Penilai Esai Otomatis Sebelumnya ..... 28
Tabel 2.2 Perbandingan Lima Sistem Penilai Esai Otomatis..... 31
Tabel 2.2 Perbandingan Lima Sistem Penilai Esai Otomatis (lanjutan)..... 32
Tabel 4.1 Soal dan Jawaban Dosen..... 41
Tabel 4.1 Soal dan Jawaban Dosen (Lanjutan)..... 42
Tabel 4.2 Nilai Dosen *(Human Raters)* ..... 43
Tabel 4.3 Nilai yang Dihasilkan Sistem Penilai Esai Otomatis Berbasis Algoritma LSA *(Latent Semantic Analysis)* ..... 44
Tabel 4.4 Waktu Proses pada Sistem Penilai Esai Otomatis Berbasis Algoritma LSA *(Latent Semantic Analysis)* ..... 45
Tabel 4.5 Nilai yang Dihasilkan Sistem Penilai Esai Otomatis Berbasis Algoritma *GLSA (Generalized Latent Semantic Analysis)* ..... 46
Tabel 4.6 Waktu Proses pada Sistem Penilai Esai Otomatis Berbasis Algoritma *GLSA (Generalized Latent Semantic Analysis)* ..... 47
Tabel 4.7 Nilai *Pearson Product Moment Correlation* LSA dan GLSA..... 50
Tabel 4.8 Selisih Nilai dengan *Human Rater* untuk Soal 1 ..... 51
Tabel 4.9 Selisih Nilai dengan *Human Rater* untuk Soal 2 ..... 52
Tabel 4.10 Rekapitulasi Keunggulan Sistem Berbasis LSA dan GLSA..... 54
Tabel 4.11 Rataan Waktu Proses pada Sistem LSA dan GLSA ..... 56

# BAB 1
# PENDAHULUAN

## 1.1 Latar Belakang

Penelitian untuk mengembangkan sistem komputer untuk *essay grading* sudah mulai dikembangkan sejak era tahun 1960-an dan beberapa pendekatan sudah pernah diajukan oleh beberapa orang ilmuwan [1]. Dimulai oleh *Project Essay Grade* (PEG) [2] kemudian Larkey, lalu E-RATER [3] yang merupakan sistem yang mengkombinasikan *Natural Language Processing (NLP)* dan teknik statistika untuk menganalisa konten tulisan. Lalu *Bayesian Essay Testing System (BETSY)* [4]. Kemudian Foltz et al. (1999) [5] mulai mengembangkan *essay grading* yang berbasis *LSA (Latent Semantic Analysis)* untuk menganalisa konten tulisan.

Dari beberapa metode yang ada untuk melakukan penilaian berdasar isi konten, hanya LSA yang sebelumnya sudah pernah diaplikasikan untuk proses *automatic essay grading* dan aplikasi edukasi lainnya seperti *intelligent tutoring system* [6] dan untuk memeriksa rangkuman yang dibuat oleh siswa [7]. LSA sudah dibuktikan merupakan salah satu metode yang sukses dapat diterapkan pada sistem *essay grading*.

Pada beberapa skenario tes, Landauer et al. (1997) [8] dan Foltz et al. (2000) [9], dilaporkan bahwa tingkat korelasi penilaian yang dilakukan oleh dua orang pemeriksa manusia adalah sebesar 0,64 – 0,84 dan sebesar 0,59 – 0,89 untuk penilaian yang dihasilkan dari sistem *essay grading* berbasis LSA. Ini mengindikasikan bahwa sistem berbasis LSA menghasilkan performa yang sama baiknya dengan performa yang dilakukan oleh manusia. Tujuan dari skripsi ini adalah untuk mengetahui jika implementasi algoritma *Generalized Latent Semantic Analysis (GLSA)* pada sistem SIMPLE-O dapat menjadikan akurasi sistem yang sekarang menjadi semakin baik dengan ditandai oleh meningkatnya nilai korelasi antara pemeriksa manusia *(human graders)* dan nilai yang dihasilkan oleh sistem SIMPLE-O.

## 1.2 Tujuan Skripsi

Adapun tujuan dari skripsi ini:

1. Menjabarkan beberapa algoritma yang sudah pernah diaplikasikan untuk sistem *essay grading*
2. Menjabarkan kelebihan dan kekurangan masing-masing algoritma
3. Menentukan algoritma yang memiliki performa yang paling baik berdasarkan studi literatur
4. Mengimplementasikan algoritma *Generalized Latent Semantic Analysis (GLSA)* pada sistem Simple-O
5. Menganalisa pengaruh atas implementasi algoritma *Generalized Latent Semantic Analysis (GLSA)* pada sistem Simple-O

## 1.3 Pembatasan Masalah

Skripsi ini bertujuan untuk menjabarkan beberapa algoritma yang sudah pernah diaplikasikan untuk sistem *essay grading* dan menentukan algoritma yang memiliki performa yang paling baik kemudian mengimplementasikannya pada sistem Simple-O lalu menganalisa pengaruh atas implementasi tersebut.

## 1.4 Metodologi Penelitian

Metode yang digunakan dalam melakukan penelitian hingga selesainya skripsi ini ditulis yakni:

1. Studi literatur, yaitu melakukan pencarian dan penelaahan secara luas dan mendalam mengenai referensi dan tulisan dengan subjek sejenis atau berkaitan yang dapat membantu dan mendukung penulisan skripsi ini
2. Pengamatan dan pengukuran langsung dari implementasi dan eksperimen yang dilakukan

## 1.5 Sistematika Penulisan

Tulisan skripsi ini terdiri dari lima bab, yang mana tiap-tiap bab akan menjelaskan hal-hal sebagai berikut:

a. Bab 1: Pendahuluan

   Menjelaskan latar belakang skripsi, tujuan skripsi, pembatasan masalah, metodologi penelitian, serta sistematika penulisan.

b. Bab 2: Sistem Penilai Esai Otomatis: Konsep, Algoritma, Unjuk Kerja, dan Perbandingannya

   Menjabarkan dasar-dasar pengetahuan tentang sistem sistem *essay grading, Simple-O, algoritma Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Generalized Latent Semantic Analysis (GLSA),* perbandingan unjuk kerja sistem esai grading yang pernah diciptakan, komputasi hijau *(green computing)*.

c. Bab 3: Perancangan Sistem Penilai Esai Otomatis Berbasis Algoritma *GLSA* (*Generalized Latent Semantic Analysis)*

   Menerangkan perangkat lunak dan perangkat keras yang digunakan serta rancangan sistem penilai esai otomatis berbasis algoritma *GLSA* (*Generalized Latent Semantic Analysis)*

d. Bab 4: Analisa Unjuk Kerja Sistem Penilai Esai Otomatis Berbasis Algoritma *GLSA* (*Generalized Latent Semantic Analysis)* dan Perbandingannya dengan Unjuk Kerja Sistem Penilai Esai Otomatis Berbasis Algoritma *LSA (Latent Semantic Analysis)*

   Berisi pengolahan data uji coba unjuk kerja pada sistem penilai esai otomatis berbasis algoritma *LSA (Latent Semantic Analysis)* dan *GLSA* (*Generalized Latent Semantic Analysis),* analisa atas data tersebut berupa perbandingan unjuk kerja antara kedua sistem serta penjelasan mengapa dan bagaimana hasil tersebut yang diperoleh.

e. Bab 5: Kesimpulan dan Saran

   Berisi kesimpulan penelitian dan saran pengembangan serta perbaikan untuk sistem di masa mendatang.

# BAB 2
# SISTEM PENILAI ESAI OTOMATIS: KONSEP, ALGORITMA, UNJUK KERJA, DAN PERBANDINGANNYA

## 2.1 Sistem Penilai Esai Otomatis

Sebagai bagian dari pendidikan, menulis telah menjadi bagian penting dari pendidikan. Ketika otomatisasi telah menjadi kebutuhan mendesak, dengan sebagian besar pekerjaan telah diotomatisasi, mengevaluasi banyak esai secara manual telah menjadi persoalan sejak lama. Sejak tahun 1960-an, banyak sistem telah dikembangkan untuk mengevaluasi esai secara otomatis. Salah satu sistem awal yang dikembangkan untuk mengatasi persoalan ini yakni *Project Essay Grader (PEG)* [2]. Sistem ini tidak diterima secara luas, karena fakta bahwa sebagian besar faktor-faktor yang dipertimbangkan oleh sistem ini sebagai dasar perhitungan hanyalah bagian-bagian yang tidak krusial contohnya jumlah tanda koma yang ada di tulisan, jumlah preposisi dan jumlah kata-kata yang tidak biasa. Ide di balik konsep tersebut belum dapat menjawab kebutuhan yang ada dan karenanya tidak diterima secara luas.

Setelah PEG, pada tahun 1980-an dikembangkan sebuah sistem penilai esai yang disebut *Writer's Work Bench Tool (WWB)* yang membantu memberikan umpan balik kepada editor tentang berbagai aspek seperti ejaan, diksi dan keterbacaan [2]. Sistem ini memiliki fungsi untuk membantu memeriksa ejaan dan menunjukkan penggunaan kata yang kurang tepat. Tetapi karena WWB tidak memperhitungkan analisis semantik, sistem itu terbatas dalam ruang lingkup - dalam hal ini hanya untuk evaluasi. Selama periode yang sama Tom Landauer dan rekan-rekannya muncul dengan pendekatan yang disebut *Latent Semantic Analysis* [10]. Teknik berbasis *Latent Semantic Analysis* ini menggunakan pendekatan "*bag of words*", dimana urutan kata tidak dianggap penting, tetapi kesamaan antara kata-kata yang dipergunakanlah yang diperhitungkan dan diukur menggunakan sebuah matriks *co-occurence* dan *singular value decomposition (SVD).* Tom Landauer dan rekan-rekannya melakukan penelitian lebih jauh

mengenai pendekatan ini dan pada akhirnya mengembangkan sistem *Intelligent Essay Assessor (IEA)* [5].

Sekarang ada sistem lain yang mengikuti PEG, yaitu E-RATER [11] [12] sebuah sistem yang dikembangkan di ETS. Sistem ini menghasilkan pengukuran yang lebih akurat dan telah digunakan untuk menilai esai dalam ujian *GMAT (Graduate Management Admission Test).* Metode pengukuran yang digunakan oleh E-RATER ini lebih komprehensif dan mencakup struktur dokumen. Dengan demikian kesamaan jawaban mahasiswa dapat diukur dengan membandingkannya dengan sebuah training set berisi beberapa esai dan berdasarkan nilai kesamaan ini, mahasiswa akan diberi skor dengan skala dari 1-6.

Kemudian pada tahun 1999, Christie merancang sebuah sistem penilaian esai baru *SEAR (Schema Extract Analyse and Report)* [13] yang didasarkan atas parameter baru, yakni bagaimana gaya esai ditulis daripada membandingkan konten dalam esai. Pendekatan ini membutuhkan suatu pelatihan awal yang diajarkan pada sistem dan kalibrasi menggunakan seperangkat esai. Pendekatan ini juga menggunakan rujukan konten yang disebut *content schema* yang diwakili sebagai sebuah struktur data dan dapat secara fleksibel direvisi bilamana diperlukan. SEAR telah menginspirasi peneliti lain untuk ikut mempertimbangkan bagaimana gaya esai ditulis, meskipun pendekatan ini memiliki pro dan kontra. Pada tahun berikutnya, tiga mahasiswa Ming, Mikhailov dan Kuan dari Ngee Ann Polytechnic muncul dengan sistem penilaian esai berjudul *Intelligent Essay Marking Systems (IEMS) [*14], yang dibuat berdasarkan pada pengindeksan pola jaringan syaraf (*Pattern Indexing Neural Network*) atau *Indextron,* sebuah algoritma *clustering* yang dapat diimplementasikan melalui jaringan neural buatan. IEMS mendapat pujian atas fiturnya yang dapat memberikan umpan balik secara instan kepada mahasiswa pada bagian mana mereka telah mengerjakan soal ujian dengan baik dan juga secara spesifik menerangkan dimana kesalahan mereka. Setelah tiga tahun melakukan riset dan pengembangan, Mitchell, Russel, Broomhead, dan Aldridge pada tahun 2002 hadir dengan sebuah perangkat lunak yang disebut *Automark* [15], sebuah perangkat lunak yang dirancang untuk menilai jawaban atas pertanyaan-pertanyaan terbuka. Sistem ini didasarkan pada

teknik-teknik NLP (*Natural Language Processing*) dan menggabungkan metode-metode untuk mengecek ejaan, ketepatan semantik serta pengecekan tanda baca. Sistem ini mencari konten spesifik dalam jawaban terbuka. Konten akan berupa *template* yang menunjukan jawaban yang benar atau salah. Sebuah langkah baru dalam pengembangan sistem penilai esai otomatis.

## 2.2 Simple-O

Simple-O adalah suatu aplikasi berbasis web yang dikembangkan di Departemen Teknik Elektro Universitas Indonesia, dan digunakan untuk menilai jawaban dari ujian essai secara otomatis dalam bahasa Indonesia. Sistem penilaian didesain menggunakan algoritma *Latent Semantic Analysis (LSA)* yang diatur sedemikian rupa dengan menggunakan teknik aljabar linear *Singular Value Decomposition (SVD).* Pertama konten jawaban ditransformasikan menjadi sebuah matriks lalu digunakan teknik aljabar linear SVD untuk mendekomposisi matriks menjadi tiga buah komponen matriks yakni komponen matriks pertama yang mendeskripsikan entitas baris sebagai vektor orthogonal matriks, komponen matriks kedua mendeskripsikan matriks diagonal yang berisi nilai skalar dan yang ketiga adalah matriks entitas kolom sebagai vektor orthogonal matriks [16]. Setelah itu penilaian dilakukan dengan mengambil kosinus sudut antara dua vektor yang dibentuk oleh dua baris lalu membandingkannya. Nilai dekat dengan satu mewakili kata-kata yang sangat mirip sementara nilai-nilai mendekati nol mewakili kata-kata yang sangat berbeda.

Pada 2007 dari hasil uji coba sistem Simple-O yang telah dilakukan, pada kelas kecil diperoleh nilai kesesuaian dengan *human raters* berkisar 69.80 % – 94.64 %, sedangkan pada kelas menengah diperoleh nilai berkisar 77.18 % – 98.42 %. Hasil-hasil ini setara dengan hasil metoda *grading* otomatis LSA terdahulu, yang melakukan penilaian terhadap jawaban ujian dalam bahasa Inggris [16].

Aplikasi ini bekerja dengan prinsip *client-server*, dimana mahasiswa yang akan melakukan ujian diminta untuk mengakses website Simple-O untuk mengisikan jawaban dari soal esai yang diberikan pada kolom yang disediakan.

Kalkulasi penilaian akan dibebankan pada server begitu juga semua database soal, jawaban, dan nilai akan disimpan pada server.

Algoritma pada Simple-O dibagi ke dalam beberapa modul yakni modul login, modul dosen, dan modul mahasiswa. Modul login berfungsi untuk membatasi akses ke sistem sehingga tidak sembarangan individu dapat mengakses aplikasi. Modul dosen berfungsi untuk menginput kata kunci dari jawaban esai yang benar lalu memberikan bobot terhadap kata kunci yang dianggap penting diantara kata kunci yang telah dipilih sebelumnya. Kata kunci yang dipilih akan diberi nilai 1 dan kata kunci yang dianggap penting akan diberi bobot 2. Bobot ini digunakan untuk membentuk matriks kata kunci sebagai baris dan urutan kalimat sebagai kolom. Modul mahasiswa digunakan oleh peserta ujian untuk menginput jawaban atas pertanyaan yang diberikan [16].

## 2.3 *Latent Semantic Analysis (LSA)*

*Latent Semantic Analysis (LSA)* merupakan sebuah teknik dalam bidang pemrosesan bahasa alami atau *natural language processing (NLP)*, sebuah cabang ilmu komputer dan ilmu bahasa yang mengkaji interaksi antara komputer dengan bahasa alami manusia [17]. NLP sering dianggap sebagai cabang dari kecerdasan buatan (*Artificial Intelligence*) yang bidang kajiannya bersinggungan dengan linguistik komputasional, sebuah bidang disiplin antar ilmu yang mengkaji permodelan bahasa alami manusia menggunakan kombinasi statistika dan aturan dari sudut pandang komputasi.

Terdapat empat kajian spesifik dalam bidang NLP antara lain segmentasi wicara atau ucapan untuk mengidentifikasi batas antara kata, suku kata, atau fonem pada bahasa lisan dengan memperhitungkan konteks, tata bahasa, dan semantik (*speech segmentation),* segmentasi tulisan tertulis menjadi unit kata, kalimat, paragraf atau topik *(text segmentation),* penandaan kelas kata pada suatu tulisan berdasarkan definisi dan maknanya dengan kata yang mendampingi *(part-of-speech tagging),* serta pengawataksaan makna satu kata dengan banyak arti (polisemi) yang digunakan pada suatu kalimat *(word sense disambiguation).*

Teknik LSA khususnya digunakan pada semantik vektorial, sebuah model aljabar untuk merepresentasikan tulisan dokumen ataupun objek lain pada umumnya sebagai vektor pengenal. Teknik ini digunakan untuk menganalisa keterhubungan dan relevansi antara beberapa dokumen yang ada. Caranya dengan menciptakan set berisi beberapa konsep terkait menggunakan kata-kata yang sama dengan dokumen yang ingin diujikan tersebut. Teknik LSA mengasumsikan bahwa penggunaan kata-kata yang artinya mendekati akan menghasilkan keterhubungan pada tulisan. Sebuah matrik yang mengandung jumlah kata per paragraf (baris pada vektor mewakili kata-kata unik dan kolom mewakili setiap paragraf) disusun dari sepotong besar tulisan dan sebuah teknik matematika yang disebut dekomposisi nilai singular / *singular value decomposition (SVD)* digunakan untuk mengurangi jumlah kolom sambil menjaga kesamaan struktur antara baris. Keterhubungan kata-kata ini kemudian dibandingkan dengan mengambil kosinus sudut antara dua vektor yang dibentuk oleh dua baris. Nilai dekat dengan satu mewakili kata-kata yang sangat mirip sementara nilai-nilai mendekati nol mewakili kata-kata yang sangat berbeda.

LSA telah dipatenkan pada tahun 1988 (US Patent 4,839,853) oleh Scott Deerwester, Susan Dumais, George Furnas, Richard Harshman, Thomas Landauer, Karen Lochbaum dan Lynn Streeter. Dalam konteks aplikasi untuk pencarian informasi *(Information Retrieval / IR)*, kadang-kadang LSA juga disebut sebagai *Latent Semantic Indexing (LSI)* [18].

*Latent Semantic Analysis (LSA)* merupakan sebuah teknik dalam ilmu statistik yang digunakan untuk menentukan kesamaan makna dari kata-kata dan tulisan dengan cara melakukan analisis terhadap teks dalam jumlah yang besar [10]. LSA pada awalnya digunakan untuk melakukan *indexing* pada bidang *information retrieval (IR).* LSA merupakan salah satu metode reduksi dimensi dan algoritma *indexing* dokumen yang terbaik di bidang *information retrieval* dan NLP. LSA menggunakan informasi statistik kemunculan bersama untuk mengkomputasi semua ruang semantik laten yang berisi semua konsep untuk sebuah koleksi dokumen. LSA menghasilkan sebuah representasi rangkap term-dokumen. Setiap kata pada kosa kata dan setiap dokumen disajikan sebagai

sejumlah vektor pada ruang berdimensi *k* yang direntangkan oleh vektor-vektor semantik laten.

LSA menggunakan teknik perhitungan aljabar linear yaitu *singular value decomposition (SVD).* Dengan metode SVD ini semua kata-kata yang akan dianalisa akan ditransformasikan dalam sebuah matriks yang merepresentasikan asosiasi kata pada dokumen yang merupakan *semantic space*, yakni kata–kata dan dokumen-dokumen yang berasosiasi dekat akan ditempatkan dekat satu sama lain [10]. Hasil dari proses ini, kata-kata yang tidak terdapat pada suatu dokumen mungkin saja letaknya pada *semantic space* masih dekat dengan dokumen tersebut apabila hal ini konsisten dengan *major associative pattern* yang terdapat pada data [19].

Pada proses pengolahan data menggunakan *singular value decomposition* yang digunakan pada LSA, akan dilakukan proses pengurangan dimensi pada matriks hasil dekomposisi SVD. SVD pada LSA dapat dijelaskan dengan lebih detail sebagai berikut [19] [20]:

Diberikan sebuah koleksi dokumen $C = \{d_1, .., d_n\}$ dengan kosa kata $V = \{t_1, .., t_v\}$, LSA mengasumsikan bahwa ada sejumlah k konsep semantik laten dalam sebuah koleksi dokumen. Dengan demikian, semua dokumen akan direpresentasikan sebagai sejumlah vektor dalam ruang berisi sejumlah konsep semantik laten: $d = (c_1, .., c_k)$.

Setiap konsep semantik adalah sebuah vektor yang diindeks berdasarkan kata-kata pada kosa kata:

$$c_i = (t_1, t_2, .., t_v) \qquad (2.1.)$$

Representasi ini mengungkapkan bagaimana kata-kata yang berhubungan secara semantik dikelompokkan bersama ke dalam sejumlah kelas semantik. Elemen ke-*i* dari vektor dokumen, *d*[*i*] merupakan sebuah ukuran akan pentingnya konsep laten ke-*i* pada dokumen ini. Nilai dari *d*[*i*] adalah bukan nol jika *d* mengandung salah satu dari kata-kata yang digunakan pada koleksi dokumen untuk menyajikan konsep ke-*i*, yang akan membantu untuk menghitung sinonim.

Jika *d* mengandung sebuah kata polisemi, suatu kata yang memiliki makna lebih dari satu, yang dapat digunakan untuk merepresentasikan konsep ke-*i* dan sejumlah konsep lain, nilai dari *d*[*i*] akan besar hanya jika ada sejumlah kata-kata lain dari wilayah semantik yang sama hadir juga pada dokumen tersebut, yang membantu untuk menangani kata polisemi.

Ruang semantik laten direntangkan oleh sejumlah vektor semantik laten. Vektor-vektor semantik laten LSA dikomputasi dengan melakukan SVD dari matriks term-dokumen yang merepresentasikan koleksi dokumen. Salah satu sifat dari ruang LSA adalah bahwa vektor-vektor ini adalah ortogonal. *Dot product* atau kemiripan *cosine* antara sejumlah *data points* diproyeksikan ke dalam ruang yang dapat digunakan sebagai ukuran kemiripan diantara keduanya.

Masukan untuk SVD adalah matriks term-dokumen dan tidak ada informasi lain yang dibutuhkan. Diberikan sebuah koleksi dokumen $C = \{d_1, .., d_n\}$ dengan kosa kata $V = \{t_1, .., t_v\}$, LSA diawali dengan *D*, sebuah matriks term-dokumen $v \times n$. LSA mengkomputasi sebuah SVD dari *D*.

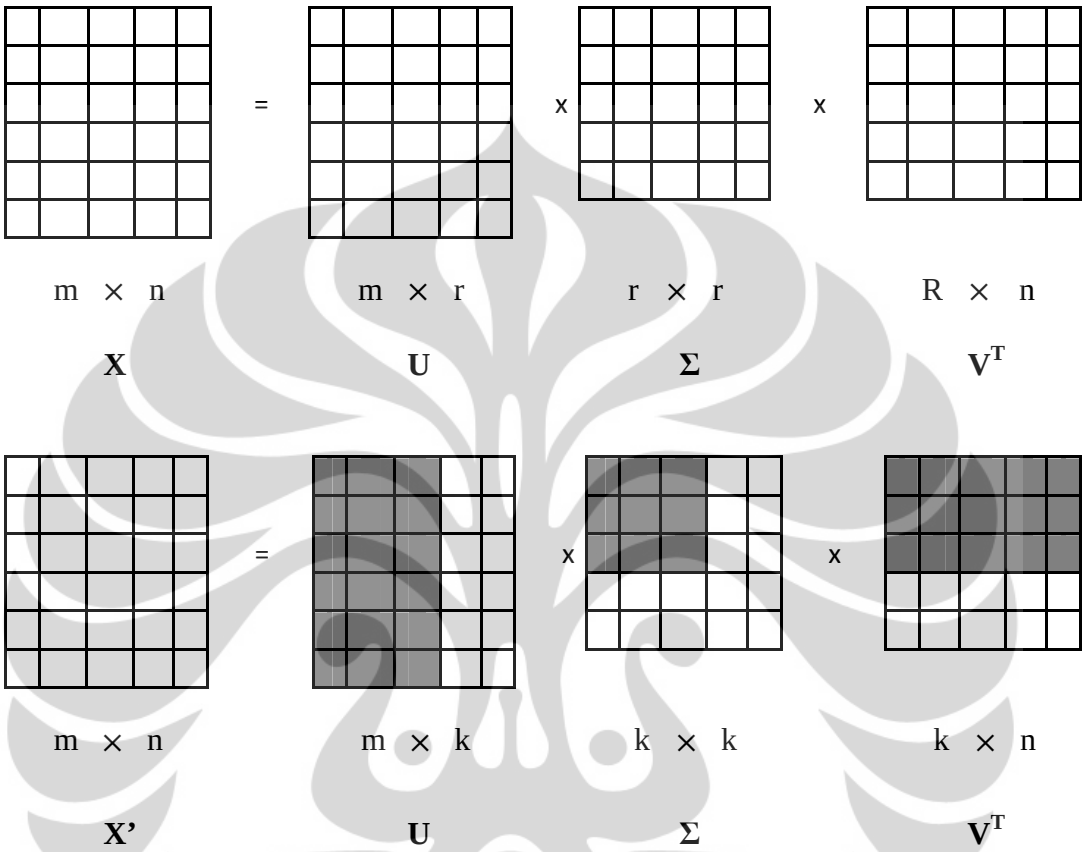
*D*, sebuah matriks persegi panjang $n \times m$ dengan *rank* $r$ dapat didekomposisi sebagai

$$D = U\Sigma V^T \qquad (2.2.)$$

dimana $U = (u_1, .., u_r)$ adalah sebuah matriks $n \times r$, $V = (v_1, .., v_r)$ adalah sebuah matriks $m \times r$. $U$ dan $V$ adalah matriks-matriks ortogonal kolom, kolom-kolom matriks ini disebut vektor-vektor singular kiri dan kanan. Σ adalah sebuah matriks diagonal $r \times r$ yang semua elemen diagonalnya adalah nilai-nilai singular. Semua nilai singular biasanya diatur sedemikian rupa sehingga $\sigma_i \geq \sigma_{i+1}$.

Langkah selanjutnya dari SVD ialah membentuk aproksimasi dari matriks *D* yakni $D_k$ dengan melakukan pengurangan dimensi. Jika nilai-nilai singular dari matriks Σ diurutkan berdasarkan nilainya, maka sejumlah *k* nilai terbesar dapat tetap disimpan dalam matriks tersebut, dan nilai–nilai lain yang lebih kecil dapat diatur menjadi nol. Jika baris dan kolom yang berkaitan pada matriks $U$ dan $V^T$ juga diatur menjadi nol, maka hasil kali dari ketiga matriks ini akan membentuk

matriks $D_k$ yang merupakan matriks *least square approximation* dari matriks *D*. Gambar 2.1 merupakan ilustrasi pembentukan matriks $D_k$ dengan menggunakan dimensi sebesar *k*.




Gambar 2.1 Proses Reduksi Dimensi pada SVD

Representasi LSA menggunakan sejumlah *k* nilai-nilai singular terbesar dan vektor-vektor singular kanan yang bersesuaian untuk membangun sebuah aproksimasi dari matriks *D* dengan *rank k*:

$$D_k = U_k \Sigma_k V_k^T \qquad (2.3.)$$

Eckart dan Young [23] telah menunjukkan bahwa $D_k$ adalah aproksimasi terbaik dengan *rank k* untuk *D* dalam hal dengan norma *Frobenius*, yaitu $D_k$ meminimalisasi

$$\|D - X\|_F^2 = \sum_{ij} (D[i][j] - X[i][j])^2 \qquad (2.4.)$$

diantara semua matriks *X* dengan *rank r*.

Pada model ini, k adalah jumlah dari konsep semantik laten. $U_k$ dapat dilihat sebagai sebuah representasi dari kata-kata pada kosa kata sebagai sejumlah vektor yang diindeks oleh sejumlah konsep semantik laten dimana $U_k[i][j]$ adalah asosiasi antara kata ke-*i* dan konsep ke-*j*. Dengan cara yang sama, $V_k$ adalah sebuah representasi dari sejumlah dokumen dalam hal sejumlah konsep semantik laten. Dengan demikian, $D_k$ memiliki keterkaitan antara kata-kata dan sejumlah dokumen yang pertama diproyeksikan dalam ruang semantik laten.

Kolom-kolom dari $U_k$ merepresentasikan sejumlah konsep semantik laten, sehingga

$$U_k^T D = \Sigma_k V_k^T \tag{2.5.}$$

adalah proyeksi dari sejumlah dokumen dalam ruang konsep-konsep laten tersebut. Bersamaan,

$$D^T = V\Sigma U^T \tag{2.6.}$$

sehingga

$$V_k^T D^T = \Sigma_k U_k \tag{2.7.}$$

adalah proyeksi dari kata-kata pada kosa kata ke dalam ruang semantik laten.

Dari matriks-matriks hasil dekomposisi SVD terdapat tiga operasi perbandingan yang dapat dilakukan yaitu: [10] [19]

1. Perbandingan seberapa besar kemiripan antara kata-kata dan sejumlah dokumen dilakukan dengan menggunakan *cosine* atau *dot product* antara kolom-kolom dari $U_k^T D$ atau $V_k^T D^T$, berturutan. Koordinat dari suatu kata pada ruang semantik direpresentasikan oleh vektor baris dari matriks Σ x *U* yang bersesuaian dengan kata tersebut. Oleh karena itu kemiripan antara dua kata yang berbeda dapat diperoleh dari cosine similarity antara koordinat-koordinat dari kedua kata tersebut.
2. Membandingkan seberapa besar kesamaan diantara dua dokumen yang berbeda. Koordinat suatu dokumen pada ruang semantik direpresentasikan

oleh vektor baris dari matriks Σ x *V* yang bersesuaian dengan dokumen tersebut. Oleh karena itu kemiripan antara dua dokumen yang berbeda dapat diperoleh dari *cosine similarity* antara koordinat-koordinat dari kedua dokumen tersebut.

3. Mengetahui seberapa besar suatu kata tertentu berasosiasi dengan suatu dokumen. Berbeda dari dua operasi sebelumnya yang memerlukan penghitungan *cosine similarity*, seberapa besar asosiasi antara suatu kata *i* dengan suatu dokumen *j*, dapat diketahui dari nilai sel *i*,*j* dari matriks aproksimasi *term*-dokumen yang dihasilkan oleh SVD.

Dokumen yang tidak muncul pada matriks *term*-dokumen, juga dapat direpresentasikan sebagai sebuah *pseudodocument* dalam ruang semantik. Untuk dapat merepresentasikan dokumen eksternal ini sebagai *pseudodocument* pada ruang semantik, dapat dilakukan dengan menggunakan formula 2.8.

$$\widehat{D} = DUS^{-1} \qquad (2.8.)$$

dimana $\widehat{D}$ adalah representasi *pseudodocument* pada ruang semantik, dan *D* adalah matriks vektor bobot kata pada dokumen.

## 2.4 *Probabilistic Latent Semantic Analysis (PLSA)*

*Probabilistic Latent Semantic Analysis (PLSA)*, juga dikenal sebagai *Probabilistic Latent Semantic Indexing (PLSI)*, khususnya dalam bidang *Information Retrieval* merupakan sebuah teknik pengambilan informasi yang dirancang untuk memperbaiki masalah yang muncul pada *Latent Semantic Analysis (LSA).* Tuomo Kakkonen, Niko Myller, Jari Timonen, dan Erkki Sutinen melaporkan hasil percobaan implementasi algoritma LSA dan PLSA dalam sistem penilaian esai otomatis yang berbahasa Finlandia yakni *Automatic Essay Assessor (AEA)* menggunakan tiga set dokumen esai dari berbagai mata pelajaran [21]. Dari percobaan ditemukan bahwa kedua metode tersebut memiliki akurasi yang hampir sama menggunakan pengukuran *Spearman Correlation* antara nilai yang didapat dari pemeriksa manusia dan pemeriksa otomatis. Lebih lanjut, mereka

mengusulkan metode untuk meningkatkan penggunaan PLSA pada sistem penilaian esai otomatis. Hasil percobaan mereka menunjukkan bahwa akurasi LSA dan PLSA hampir sama akan tetapi LSA masih lebih unggul daripada PLSA.

Dibandingkan dengan LSA biasa yang merupakan teknik aljabar linear dan reduksi matriks yang biasanya menggunakan SVD, PLSA didasarkan pada gabungan dari dekomposisi yang diturunkan dari model kelas laten. Hal ini menghasilkan pendekatan yang lebih mendasar yang didukung oleh teknik statistika. Menimbang hasil pengamatan dalam bentuk *co-occurences (w,d)* dari kata-kata dan dokumen, PLSA memodelkan kemungkinan setiap *co-occurence* sebagai kombinasi dari distribusi multinomial yang tidak bergantung satu dengan yang lain, secara matematis:

$$P(w,d) = \sum_c P(c)P(d \mid c)P(w \mid c) = P(d)\sum_c P(c \mid d)\,P(w \mid c) \qquad (2.9.)$$

Penurunan pertama adalah formulasi simetris dimana w dan d keduanya dihasilkan dari kelas laten c dengan cara yang sama menggunakan peluang kondisional P (d|c) dan P (w|c) dan formulasi kedua dengan formulasi asimetrik dimana setiap dokumen d, sebuah kelas laten dipilih sesuai dengan dokumen mengacu pada nilai P (c|d) dan sebuah kata kemudian dibuat dari kelas tersebut mengacu pada nilai P (w|c). Meskipun pada penjelasan ini hanya digunakan contoh kata dan dokumen, *co-occurence* dari variabel diskrit lain juga bisa dimodelkan dengan cara yang sama.

Telah dilaporkan bahwa model PLSA memiliki sejumlah persoalan dalam hal perhitungan [22]. Jumlah parameter yang digunakan bertambah secara linear sesuai dengan jumlah dokumen yang digunakan. Lebih jauh, walaupun PLSA merupakan model generatif dari dokumen yang digunakan sebagai data pelatihan, diperkirakan bahwa PLSA bukan sebuah model generatif yang baik untuk dokumen baru.

## 2.5 *Generalized Latent Semantic Analysis (GLSA)*

Pemikiran atas GLSA awalnya dicetuskan oleh Irina Matveeva dari University of Chicago pada disertasinya di tahun 2008 [23]. Atas hasil penemuan ini proses LSA yang sudah ada mulai ditinjau kembali dengan mempertanyakan setiap aspek yang ada di dalamnya termasuk preservasi makna esai saat proses pembentukan matriks dokumen seperti yang dicetuskan oleh Andrew M. Olney dari University of Memphis dalam tulisannya [24]. Andrew berpendapat bahwa pembentukan matriks berdasarkan kata tersebut sebenarnya dapat dikatakan berupa pendekatan yang tergantung pada penilaian tanpa pembatasan atau aturan yang pasti.

Biasanya proses LSA dilakukan dalam dua langkah, pertama pembentukan matriks dokumen berdasarkan kata dilanjutkan dengan proses singular value decomposition (SVD) matriks tersebut. Akan tetapi pembentukan matriks berdasarkan kata tersebut sebenarnya dapat dikatakan berupa pendekatan yang tergantung pada penilaian tanpa pembatasan atau aturan yang pasti. Berdasarkan kondisi tersebut, muncullah GLSA sebagai jawaban untuk merepresentasikan makna dan konteks dokumen agar lebih sesuai dengan melakukan konseptualisasi ulang LSA secara lebih umum [24]. Dengan demikian GLSA membuka kemungkinan penciptaan algoritma-algoritma baru yang bisa saja memiliki performa yang lebih unggul daripada LSA pada umumnya.

Konsep representasi dokumen sebagai vektor dokumen GLSA yang digambarkan pada eksperimen [23] dapat memperluas pendekatan LSA ke area permasalahan yang baru dan memperluas wawasan penelitian dan pengetahuan pendekatan berbasis LSA di masa yang akan datang bahwa GLSA adalah strategi yang produktif dan amat berguna sehingga patut dipelajari lebih lanjut. Penelitian Irina Matveeva [23] menunjukkan implementasi GLSA pada sejumlah sistem berbeda mulai dari uji sinonim, klasifikasi dokumen, dan *clustering* menghasilkan tingkat efektivitas yang lebih unggul daripada sistem-sistem yang ada sebelumnya. Hasil percobaan [23] membuktikan GLSA lebih unggul pada hampir semua algoritma yang sudah ada dengan kegunaan yang beraneka ragam.

Dalam bidang linguistik komputasional dan probabilitas, sebuah n-gram adalah rangkaian bersambung dari n objek atas rangkaian kata atau tulisan yang sudah ada. Objek tersebut dapat berupa fonem, suku kata, huruf, kata, atau pasangan kata berdasarkan aplikasi. Sejumlah n-gram diperoleh dari sebuah tulisan ataupun korpus bahasa. Korpus adalah himpunan karangan dengan tema, masalah, pengarang, atau bentuk yang sama. Sebuah n-gram berukuran satu disebut sebagai "*unigram*"; ukuran dua adalah "*digram*" atau lebih umum disebut "*bigram*"; ukuran tiga adalah "*trigram*". Untuk ukuran yang lebih besar kadang disebut sebagai nilai dari n, contohnya "*four-gram*", "*five-gram*", dan seterusnya.

Pada algoritma GLSA yang diciptakan oleh Irina Matveeva dari University of Chicago [23], proses dalam merepresentasikan tulisan dokumen ataupun objek lain sebagai vektor pengenal dilakukan dengan cara yang amat berbeda dengan pendekatan yang umumnya ada karena pendekatan yang ada difokuskan pada vektor-vektor kata daripada representasi ganda *term*-dokumen. Pergeseran ini memungkinkan GLSA untuk mengambil keuntungan dari hasil-hasil penelitian pada bidang kesamaan kata dan ketersediaan koleksi dokumen-dokumen yang ada contohnya data Google n-gram. Dengan metode ini juga dimungkinkan untuk menggunakan beragam metode *spectral embedding* dan oleh karena itu, memungkinkan kita untuk mengendalikan secara langsung ruang metrik yang dihasilkan.

Model kerja GLSA fokus pada proses bagaimana menghasilkan representasi terbaik yang menjaga kesamaan semantik antar kata dan oleh karena itu kesamaan topik antar dokumen. Secara umum GLSA menggunakan pendekatan berbasis *spectral embedding* untuk menjaga kesamaan semantik dokumen asli.

Pada disertasinya [23], Irina Matveeva memperkenalkan GLSA sebagai *framework* komputasi vektor dokumen. Berlawanan dengan LSA dan algoritma berbasis reduksi dimensi pada umumnya yang diaplikasikan kepada dokumen, proses difokuskan pada penghitungan vektor kata; vektor dokumen diproses sebagai kombinasi linear dari vektor-vektor kata. GLSA ini tidak berbasis pada vektor dokumen dengan pendekatan *bag of words*. Melainkan proses dimulai

dengan kesamaan-kesamaan antar kata yang berpasangan secara semantik untuk melakukan komputasi representasi untuk kata-kata. Ini bergeser dari pendekatan yang umumnya representasi dokumen-kata menjadi representasi kata yang mempunyai maksud sebagai berikut:

- Kata-kata mempunyai tingkat fleksibilitas yang lebih tinggi dalam hal melakukan pencarian hubungan kesamaan yang ada daripada dokumen-dokumen.
- Ketersediaan koleksi yang banyak seperti *web* menawarkan sumber daya yang luar biasa untuk pendekatan statistik.
- Baru-baru ini, penghitungan kemiripan semantik yang berbasis pada *co-occurence* telah menunjukkan peningkatan performa pada beberapa tugas antara lain test sinonim, induksi taksonomi, dan *clustering* dokumen [26][27][28][29]. Di sisi lain, banyak metode *semi-supervised* dan metode transduktif yang berbasis pada vektor dokumen belum bisa menangani koleksi dokumen yang besar dan mengambil keuntungan dari informasi ini.
- Kata-kata yang mengandung makna, contohnya kata-kata yang menyampaikan informasi semantik paling banyak, sering kali dikombinasikan menjadi kelas-kelas semantik yang berhubungan kepada aktifitas atau relasi tertentu dan mengandung sinonim dan kata yang berhubungan secara semantik. Oleh karena itu, sudah menjadi hal yang alami untuk menyajikan kata-kata sebagai vektor-vektor yang berdimensi rendah pada konsep-konsep semantik yang ada. Irina Matveeva [23] berhasil mengaplikasikan vektor-vektor kata pada GLSA pada test sinonim dan menggunakan vektor dokumen GLSA untuk klasifikasi dokumen, *clustering*, dan segmentasi tulisan.

Algoritma GLSA memiliki pengaturan sebagai berikut [23]. Diasumsikan bahwa sistem memiliki koleksi dokumen $C = \{d_1, .., d_n\}$ dengan kosa kata $V = t_1, .., t_v\}$ juga korpus berukuran besar yang berbasis *web* *W*.

- Bangun matriks *term*-dokumen *D* yang sudah diberi bobot, berdasarkan *C*

- Untuk daftar kosa kata dalam *V*, dapatkan sebuah matrik kesamaan atas pasangan kata, *S*, menggunakan korpus besar *W*
- Dapatkan matriks $U^T$ yaitu representasi kata-kata sebagai ruang vektor dengan dimensi rendah yang menjaga kesamaan pada *S*, $U^T \epsilon R^{kx|V|}$
- Hasilkan vektor-vektor dokumen sebagai kombinasi linear dari matriks $U^T$ dengan vektor-vektor kata

Kolom-kolom dari $\widehat{D}$ adalah dokumen-dokumen pada ruang dimensional k. Untuk membuat hal diatas lebih jelas, algoritma dasar GLSA yang digunakan dalam eksperimen akan dijelaskan lebih lanjut pada bagian berikut [23]

- Dapatkan koleksi dokumen berukuran besar *W*. Lakukan komputasi kosa kata $V = \{w_1, w_2, .., w_{|v|}\}$ untuk koleksi *C* untuk diindeks, $C = d_1, .., d_n$.
- Hitung statistik kemunculan bersama untuk pasangan kata-kata ($w_1$, $w_j$) $\epsilon$ *V* di dalam *W*. Hitung *PMI (Point-wise Mutual Information)* sebagai ukuran dari kemiripan distribusional antar pasangan kata-kata.

$$\mathrm{PMI}(w_i, w_j) = \mathrm{PMI}(w_j, w_i) = \log \frac{P(W_i=1, W_j=1)}{P(W_i=1)P(W_j=1)} \quad (2.10.)$$

$$= \log \frac{P(W_i = 1 | W_j = 1)}{P(W_i = 1)} = \log \frac{P(W_j = 1 | W_i = 1)}{P(W_j = 1)}$$

$$\text{dimana } \frac{P(W_i = 1, W_j = 1)}{P(W_j = 1)} = P(W_i = 1 | W_j = 1)$$

Kemudian bentuk matriks kemiripan *S* dari kemiripan kata yang berpasang-pasangan:

$$S[i][j] = \mathrm{PMI}(w_i, w_j) \quad (2.11.)$$

Lakukan proses komputasi untuk *eigenvalue decomposition* dari *S*

$$S = U\Sigma U^T = \left(U\Sigma^{\frac{1}{2}}\right)\left(\Sigma^{\frac{1}{2}}U^T\right) = \widehat{U}\widehat{U^T} \quad (2.12.)$$

Dimana *U* adalah sebuah matriks ortogonal kolom yang berisi sejumlah *eigenvector* dari *S* dan Σ adalah sebuah matriks diagonal dengan nilai *eigenvalue* dari *S* yang disusun secara menurun. Simpan sejumlah k kolom pertama dari $\hat{U}$

yang berhubungan kepada sejumlah k *eigenvalue* terbesar dalam bentuk $\hat{U}_k$. Baris dari $\hat{U}_k$ adalah vektor-vektor kata pada kosa kata berdimensi k, $\vec{w}_i = (c_1, .., c_k)$.

- Lakukan proses komputasi vektor-vektor dokumen GLSA sebagai sejumlah kombinasi linear dari vektor-vektor kata GLSA [23].

$$\hat{D} = U^T D \tag{2.13.}$$

$$\vec{d}_i = \sum_{w \epsilon d_i} \alpha_w^{d_i} \vec{w} \tag{2.14.}$$

Dimana $\alpha_w^{di}$ mewakili nilai *term*/kata pada dokumen.

GLSA membentuk sebuah ruang vektor untuk kata-kata pada kosa kata yang menjaga kemiripan semantik antar pasangan kata dalam matriks S. Pertama-tama, pilihan digerakkan dengan pendekatan ruang vektor menjadi representasi *term*/kata.

Representasi ruang vektor untuk kata-kata pada kosa kata dan dokumen-dokumen merupakan hal yang penting karena representasi tersebut dapat diterapkan pada banyak aplikasi seperti klasifikasi dokumen, perangkuman multi dokumen, segmentasi tulisan, dan seterusnya. Tujuan dari algoritma GLSA adalah untuk membentuk sebuah ruang yang menyediakan ukuran alami atas kemiripan semantik antar kata-kata pada kosa kata dan dokumen-dokumen. Khususnya, GLSA membentuk sebuah ruang semantik yang tersembunyi atas vektor-vektor dimana nilai *cosine similarity* yang ada tetap menjaga kemiripan semantik antar kata-kata pada kosa kata dengan cara yang paling optimal, pada beberapa kondisi optimal tertentu.

Sejauh mana kemiripan-kemiripan tersebut bisa dijaga tergantung pada metode reduksi dimensi yang digunakan contohnya *Singular Value Decomposition* (SVD). Pendekatan berbasis GLSA bisa menggunakan semua jenis pengukuran kemiripan dengan menggunakan metode reduksi dimensi apapun yang sesuai. Pada langkah akhir digunakan matriks term-dokumen untuk menyajikan bobot-bobot dalam bentuk kombinasi linear dari vektor-vektor kata.

Pada bagian berikut akan dijelaskan *PMI (Point-wise Mutual Information)* sebagai pengukuran kemiripan kata distributional yang dapat digunakan pada GLSA. Kemudian akan dijelaskan mengenai dua metode reduksi dimensi yang digunakan oleh Irina Matveeva [23] pada eksperimennya.

Seperti yang kita ketahui, ada dua langkah penting pada *framework* GLSA. Pertama, peroleh sebuah matrik kemiripan antar pasangan kata. Lalu pembentukan ruang berdimensi rendah yang menjaga kemiripan tersebut. Penggunaan sumber-sumber bahasa yang dapat ditemukan seperti pada kamus, taksonomi, dan sumber-sumber lain untuk perhitungan kemiripan semantik yang alami membutuhkan biaya yang mahal dan waktu yang lama. Oleh karena itu [23] menggunakan kemiripan kata distribusional untuk *framework*-nya untuk membuatnya menjadi lebih *scalable* dan dapat diterapkan pada data yang nyata.

Tujuan dari GLSA adalah untuk menghasilkan sebuah ruang atas representasi-representasi kata dimana kemiripan-kemiripan antar pasangan kata tetap terjaga. Oleh karena itu, penting sekali untuk menggunakan nilai kemiripan pasangan kata yang mendekati kemiripan semantik yang sebenarnya pada GLSA [23].

Kemiripan kata distribusional telah digunakan secara luas pada banyak aplikasi NLP sebagai sebuah pengukuran yang baik dan handal untuk hubungan-hubungan antar kata.

Ada beberapa pendekatan distribusional yang digunakan pada bidang ini, beberapa diantaranya:

- *Bag of words*: kejadian umum pada daerah korpus, seperti *window* kata-kata pada ukuran yang tetap, kalimat, paragraf, dokumen, dan seterusnya
- Hubungan relasional: kemunculan bersama pada struktur sintaksis, hubungan kata dengan kata atau dengan satuan lain yang lebih besar, seperti *verb-direct object*
- Pola linear atas kemunculan bersama, seperti “x merupakan jenis y”

Beraneka jenis model kemiripan distribusional bekerja paling baik untuk kelas-kelas kata yang berbeda. Setelah statistik kemunculan bersama berbasis korpus didapatkan, ini digunakan untuk melakukan perhitungan atas asosiasi semantik. Hasil eksperimen [23] menunjukkan bahwa PMI memberikan hasil paling optimal untuk semua eksperimen yang ada jika dibandingkan dengan metode lain seperti *log likelihood ratio*, tes $\chi^2$, dan *odds ratio*.

Nilai PMI antar dua variabel acak $W_1$ dan $W_2$ yang mewakili dua buah kata, $w_1$ dan $w_2$ dihitung sesuai persamaan 2.10. Nilai PMI akan besar untuk kata-kata yang cenderung untuk muncul bersama. Seperti yang dijelaskan diatas, kemunculan bersama pada konteks yang sama adalah sebuah indikator atas kedekatan semantik. Nilai PMI yang mendekati nol mengindikasikan independensi atas kemunculan kata-kata. Akan tetapi nilai PMI yang bukan nol, bergantung pada hasil perhitungan kata-kata $w_i$ dan $w_j$. Kata-kata yang jarang digunakan akan mendapatkan nilai PMI yang lebih tinggi dibanding kata-kata yang lebih umum digunakan.

Salah satu keuntungan menggunakan PMI adalah gagasan atas kedekatannya [23]. Statistik kemunculan bersama untuk PMI umumnya dikomputasi menggunakan sebuah *sliding window* [27].

Pada GLSA, matriks kemiripan *S* dengan nilai PMI atas pasangan kata-kata mungkin tidak berupa definit positif [23]. Akan tetapi karena penggunaan matriks ini berfungsi dengan baik pada prakteknya [30], salah satu pendekatan yang umum adalah hanya untuk menggunakan sejumlah *eigenvector* yang bersesuaian dengan sejumlah nilai *eigenvalue* yang positif [30].

Seperti yang dijelaskan diatas bahwa pendekatan berbasis GLSA bisa menggunakan semua jenis pengukuran kemiripan dengan menggunakan metode reduksi dimensi apapun yang sesuai. Pada percobaan yang dilakukan oleh Irina Matveeva [23], *spectral embedding* digunakan sebagai metode untuk mereduksi dimensi dan menghasilkan representasi yang berdimensi rendah. Metode ini digunakan Irina Marveeva [23] untuk menghasilkan vektor kata/term dan

dokumen yang berdimensi rendah. Metode ini bertujuan untuk menjaga hubungan antar data masukan dan oleh karena itu sesuai untuk *framework* GLSA.

Metode-metode spektral terdiri dari jenis algoritma-algoritma yang menggunakan matriks dari kemiripan pasangan kata-kata *S* dan melakukan analisa spektral atas matriks tersebut, contohnya *eigenvalue decomposition*, untuk menempatkan kata-kata dan dokumen-dokumen pada ruang vektor berdimensi rendah. Setiap metode memiliki tujuan masing-masing dan oleh karena itu menjaga kondisi awal dengan cara tertentu. Irina Marveeva [23] menggunakan dua pendekatan untuk melakukan proses komputasi *spectral embedding* untuk kata-kata pada kosa kata, *Singular Value Decomposition* (SVD) dan *Laplacian Eigenmaps Embedding*.

Pada GLSA, SVD digunakan sebagai metode untuk mereduksi dimensi matriks [23]. SVD diterapkan pada matriks *S* yang mengandung kemiripan atas pasangan kata-kata pada kosa kata. SVD atas matriks S didefinisikan sebagai:

$$S = U\Sigma V^T \tag{2.15.}$$

dimana *U* dan *V* adalah matriks-matriks ortogonal kolom yang berisi vektor singular kiri dan kanan atas *S*, berturut-turut. Σ adalah sebuah matriks diagonal dengan nilai singular yang diurutkan secara menurun. Sifat paling penting dari SVD yang digunakan pada *framework* GLSA adalah sebagai berikut [23]. Eckart & Young menunjukkan bahwa untuk setiap matriks S dan SVD-nya $S = U\Sigma V^T$, di antara semua matriks-matriks dengan *rank* K, matriks

$$S_k = U_k \Sigma_k V_k{}^T \tag{2.16.}$$

diperoleh dengan mengatur semua kecuali sejumlah k elemen diagonal pertama pada Σ menjadi nol untuk memelihara masukan dari *S* dengan hasil paling baik.

$$S_k = \underset{X,\, \mathrm{rank(X)} = \mathrm{k}}{\arg min} ||S - X||_F^2 \tag{2.17.}$$

Ini berarti bahwa $S_k$ memelihara masukan asli pada *S* dengan cara yang paling optimal sehubungan dengan norma *Frobenius*. Norma *Frobenius* untuk sebuah matriks *A* dihitung sebagai:

$$||A||_F^2 = \sum_{ij} A_{ij}^2 \quad (2.18.)$$

Penggunaan norma *Frobenius* memastikan bahwa SVD mencoba untuk memelihara setiap masukan pada *S*. Berikutnya akan ditunjukkan bagaimana SVD berkaitan dengan *Eigenvalue Decomposition* untuk matriks simetrik simetris. Hubungan ini menjelaskan bagaimana konsep dengan skala multi dimensional dibentuk pada *framework* GLSA.

Untuk setiap matriks simetris real, *S* dapat didiagonalisasi, dengan kata lain adalah mungkin untuk menyajikannya sebagai hasil perkalian dari tiga buah matriks sebagai

$$S = U\Sigma U^T \quad (2.19.)$$

dimana *U* adalah sebuah matriks ortogonal kolom yang berisi sejumlah *eigenvector* dari *S* dan Σ adalah sebuah matriks diagonal dengan nilai *eigenvalue* dari *S* yang disusun secara menurun. Ini berarti bahwa untuk sebuah matriks yang simetris, nilai *eigenvalue decomposition* $S = U\Sigma U^T$ adalah sama dengan nilai SVD yang dihasilkannya sesuai persamaan 2.19 dengan kata lain bahwa *U = V*.

Oleh karena itu, kita dapat menggunakan persamaan milik Eckart dan Young dan menuliskannya sebagai berikut. Jika sejumlah *k eigenvector* yang memiliki sejumlah k *eigenvalue* terbesar disimpan dengan mengatur semua kecuali sejumlah k masukan diagonal pada Σ menjadi nol, didapat matriks persamaan 2.16 sesuai dengan persamaan Eckart dan Young dimana $S_k$ memelihara masukan asli pada *S* dengan cara yang paling optimal sehubungan dengan norma *Frobenius*:

$$S_k = \underset{X,\, \mathrm{rank(X)} = \mathrm{k}}{\arg min} ||S - X||_F^2 \quad (2.20.)$$

Jika selain itu, jika *S* adalah semi-definit positif, semua *eigenvalue* adalah positif. Oleh karena itu, *S* dapat disajikan sebagai sebuah hasil kali dari dua matriks

$$S = \hat{U}\widehat{U^T} \quad (2.21.)$$

Dimana $\hat{U} = U\Sigma^{\frac{1}{2}}$. Fakta ini penting untuk dimengerti bagaimana GLSA melakukan proses komputasi sebuah ruang metrik yang menjaga kemiripan-kemiripan pasangan kata awal.

GLSA menggunakan sebuah matriks simetris real *S* atas kemiripan-kemiripan pasangan kata untuk melakukan komputasi sebuah ruang berdimensi rendah untuk kata-kata pada kosa kata [23]. Irina Marveeva melaporkan bahwa SVD dari sebuah matriks simetris atas kemiripan-kemiripan pasangan kata *S* adalah sama dengan nilai *eigenvalue decomposition* yang dihasilkannya [23]. Oleh karena itu, metode pertama untuk melakukan komputasi sebuah representasi berdimensi rendah yang digunakan pada disertasinya [23] adalah untuk melakukan proses komputasi *eigenvalue decomposition* dari *S* dan menggunakan sejumlah *k eigenvector* yang memiliki sejumlah k *eigenvalue* terbesar sebagai representasi atas vektor-vektor kata. Karena *S* adalah sebuah matriks kemiripan-kemiripan pasangan kata, ruang hasil GLSA akan memiliki spesifikasi semantik berikut [23].

Seperti yang ditunjukkan diatas, sejumlah *k eigenvector* yang memiliki sejumlah k *eigenvalue* terbesar pada *U* digunakan. Dengan demikian, digunakan sebuah matriks yang sudah direduksi $U_k$ yang mengandung sejumlah k kolom pertama dari *U*. Baris-baris pada $U_k$ adalah vektor-vektor dimensional *k* untuk kata-kata pada kosa kata.

Seperti yang bisa dilihat pada persamaan $S = \hat{U}\widehat{U^T}$ bahwa S adalah sebuah hasil kali dari dua buah matriks, dengan cara yang sama $S_k$ juga bisa didefinisikan sebagai sebuah hasil kali dari dua buah matriks

$$S_k = \hat{U}_k\widehat{U_k^T} \tag{2.22.}$$

dan karena $S_k$ memelihara *S* dengan cara yang paling optimal, maka $\hat{U}_k\widehat{U_k^T}$ akan memelihara semua masukan dari *S* dengan cara yang paling optimal sehingga persamaan

$$S_k = \underset{X,\,\mathrm{rank(X)}\,=\,\mathrm{k}}{\arg min} ||S - X||_F^2 \tag{2.23.}$$

dapat dituliskan sebagai

$$\hat{U}_{\mathrm{k}}\widehat{U_k^T} = \underset{X,\,\mathrm{rank(X)}=\mathrm{k}}{\arg min} ||S - X||_F^2 \qquad (2.24.)$$

Dengan kata lain bahwa, hasil-hasil antara baris-baris dari $\hat{U}_{\mathrm{k}}$ memelihara semua masukan pada *S* dengan cara yang paling optimal. Karena baris dari $\hat{U}_{\mathrm{k}}$ yang bersesuaian dengan vektor-vektor kata GLSA dan masukan-masukan pada *S* adalah kemiripan-kemiripan pasangan semantik asli antara kata-kata pada kosa kata, dapat disimpulkan bahwa pada ruang GLSA dengan SVD, hasil-hasil di dalamnya memelihara hubungan-hubungan semantik yang ada dan oleh karena itu pengukuran atas kemiripan yang masuk akal secara linguistik/kebahasaan [23].

Hal ini membenarkan pernyataan bahwa GLSA menghasilkan sebuah ruang vektor dimana kemiripan hasil didalamnya lebih masuk akal secara linguistik/kebahasaan [23]. Terutama, GLSA mencoba untuk menghasilkan sebuah ruang dimana kemiripan-kemiripan atas hasil-hasil dalam yang berpasangan antara vektor-vektor kata dan juga antara vektor-vektor dokumen memelihara hubungan-hubungan semantik antar kata-kata pada kosa kata dan dokumen-dokumen yang bersesuaian dengan hasil yang paling baik.

GLSA memfokuskan pada kesamaan hasil-hasil dalamnya [23] karena kemiripan *cosine* secara berpasangan adalah dasar untuk banyak algoritma yang digunakan misalnya untuk klasifikasi dokumen, peringkasan multi dokumen, dan segmentasi tulisan.

GLSA mempunyai kontribusi sebagai berikut. GLSA memiliki tujuan untuk menghasilkan sebuah representasi ruang vektor yang dilengkapi dengan pengukuran kemiripan yang terarah secara linguistik/kebahasaan. Dengan menggunakan konsep dasar mengenai penskalaan multidimensional, GLSA menyediakan kontrol langsung atas ruang matrik yang dihasilkan [23]. Terutama, GLSA dapat menggunakan berbagai algoritma *spectral embedding* yang memfokuskan pada sejumlah aspek yang berbeda atas kemiripan yang ada.

GLSA memfokuskan pada semua vektor kata berdimensi rendah dan dengan demikian membuatnya mungkin untuk menggunakan sejumlah

pengukuran yang efektif atas kemiripan semantik berpasangan antar kata-kata sebagai masukan untuk perhitungan atas ruang vektor untuk kata-kata [23]. Ini menghilangkan perlunya untuk menggunakan representasi dokumen-*term* sebagai masukan dan dalam ruang vektor yang dihasilkan.

Dengan memberikan representasi untuk kata-kata daripada untuk dokumen-dokumen, GLSA membuka kemungkinan untuk memperlakukan himpunan bagian dari kosa kata secara berbeda [23]. Seseorang mungkin ingin menerapkan *spectral embedding* hanya untuk kata-kata konten dan diantara kata-kata konten, sejumlah langkah-langkah yang berbeda dapat digunakan untuk kata benda dan kata kerja.

Irina Matveeva [23] menggarisbawahi dua persoalan yang sangatlah penting saat menggunakan GLSA yakni nilai *embedding dimension* dan biaya komputasi. Pertama, nilai *embedding dimension*, seperti dibahas diatas, GLSA menghitung semua vektor kata berdimensi rendah dengan menggunakan sejumlah k *eigenvector* dari matriks kemiripan. Nilai dimensi sejumlah k adalah salah satu parameter untuk GLSA. Menemukan nilai dimensi untuk *embedding dimensions* yang optimal adalah masalah yang tidak mudah dan sejumlah pendekatan yang paling berhubungan menggunakan nilai k yang menunjukkan kinerja terbaik dalam evaluasi eksperimental, umumnya nilai diantara 100 dan 500 dimensi menunjukkan hasil terbaik. Irina Matveeva [23] menggunakan k = 500 untuk kebanyakan eksperimen yang dilakukannya. Nilai ini kelihatannya adalah sebuah pilihan yang baik untuk seperangkat eksperimen pertama yang dilakukannya. Irina Matveeva [23] tidak mencoba untuk mencari nilai dimensi terbaik untuk tiap aplikasi secara berlebihan karena [23] mencoba untuk menunjukkan bahwa GLSA akan meningkatkan kinerja bahkan ketika yang digunakan adalah parameter-parameter standar.

Biaya komputasional merupakan isu penting untuk GLSA. Pada beberapa varian GLSA yang dipresentasikan oleh Irina Marveeva [23], digunakan *eigenvalue decomposition* yang berjalan dalam waktu $O(n^3)$ dimana *n* adalah jumlah dari kata-kata yang digunakan saat komputasi ruang GLSA. Ukuran kosa kata untuk banyak koleksi data baru-baru ini amat besar. Contohnya korpus TDT2

yang digunakan Irina Marveeva [23] pada eksperimennya mempunyai lebih dari 100 ribu kata-kata unik. Ini menjadikan algoritma yang melakukan komputasi secara intensif kurang skalabel.

Ada sejumlah alasan mengapa biaya komputasional bukanlah hambatan berarti dalam menggunakan GLSA [23]. Pertama, GLSA diterapkan kepada kata-kata, dan ukuran kosa kata masih lebih kecil dari banyak koleksi dokumen pada kondisi yang sebenarnya dan tidak tumbuh sama cepatnya. Lebih jauh, komputasi GLSA dapat dilakukan secara *offline* dan semua vektor kata dapat digunakan untuk mengindeks koleksi dokumen baru lainnya. Karena GLSA memfokuskan pada vektor-vektor kata, ada juga sejumlah metode lain untuk mereduksi ukuran dari matriks kemiripan. Irina Marveeva pada [23] juga memperkenalkan skema *hybrid indexing* yang menggunakan GLSA hanya untuk konten kata benda, yang mana mereduksi jumlah kata-kata dimana vektor-vektor GLSA dikomputasi dengan sejumlah besaran/*magnitude.*

Pada akhirnya, peningkatan pesat dalam sumber daya komputasi yang tersedia seperti *grid computing* dan pengembangan sejumlah *eigensolver* paralel membuat biaya komputasi menjadi pertimbangan sekunder [23]. Karena GLSA menunjukkan kinerja yang amat baik di sejumlah aplikasi, amat dipercaya bahwa manfaatnya lebih besar daripada biaya komputasi.

$$Sim(q', d'_j) = \frac{\sum_{j=1}^{t} w_{qj} * d_{ij}}{\sqrt{\sum_{j=1}^{t} (d_{ij})^2 * \sum_{j=1} (w_{qj})^2}} \qquad (2.25.)$$

Cosine Similarity dihitung dengan persamaan 2.25 dimana $w_{qj}$ adalah bobot ke-*j* dari *query vector q'* dan $d_{ij}$ adalah bobot ke-*i* dari vektor-vektor *training essay set* $d'_j$.

## 2.6 Perbandingan Performa Sistem Esai Grading yang Pernah Diciptakan

Valenti et al. melaporkan tingkat akurasi Intelligent Essay Assessor (IEA) yang berbasis LSA 0,85 – 0,91. Kakkonen et al. mengumumkan tingkat korelasi

*Automatic Essay Assessor* (AEA) dengan penilai manusia sebesar 0,75. Lemaire B. et al mengindikasikan APEX (*Assistant for Preparing Exams*), sebuah perangkat lunak untuk mengevaluasi esai mahasiswa berdasarkan konten menggunakan LSA memberikan tingkat korelasi sebesar 0,59 dengan nilai dari pemeriksa manusia [25].

Tabel 2.1 Perbandingan Performa yang Dihasilkan oleh Beberapa Sistem Penilai Esai Otomatis Sebelumnya

| Sistem | Akurasi |
|---|---|
| IEA menggunakan LSA | 0,85 – 0,91 |
| AEA menggunakan LSA | 0,75 |
| APEX menggunakan LSA | 0,59 |

*Automatic Essay Assessor (AEA)* adalah sistem yang memanfaatkan teknik-teknik *information retrieval (IR)* seperti *Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA),* dan *Latent Dirichlet Allocation (LDA)* untuk menilai esai secara otomatis. Sistem ini menggunakan bahan pembelajaran dan relatif lebih sedikit esai yang sudah diberi nilai sebelumnya oleh penilai manusia (*human graders*) untuk mengkalibrasi sistem penilaian sebelum digunakan. Tuomo Kakkonen, Niko Myller, Erkki Sutinen dan Jari Timonen melakukan serangkaian percobaan pada sistem penilai esai otomatis menggunakan algoritma LSA, PLSA, dan LDA untuk mengetahui tingkat performa masing-masing algoritma dengan menggunakan data empiris. Dari hasil percobaan ditemukan bahwa penggunaan bahan pembelajaran sebagai data pelatihan (*data training*) untuk model penilaian esai lebih unggul daripada model penilaian k-NN. Lebih jauh, mereka menemukan bahwa penilaian menggunakan algoritma LSA menghasilkan penilaian dengan tingkat akurasi sedikit lebih tinggi dibandingkan dengan PLSA dan LDA. Mereka juga menemukan bahwa kriteria pembagian bahan pembelajaran memiliki pengaruh penting terhadap performa sistem. Dari hasil percobaan diperoleh kesimpulan bahwa lebih baik membagi bahan pembelajaran ke dalam kalimat dibandingkan paragraf [1].

Mengacu kepada Kaplan, Wolff, Burstein, Li, dan Rock, ada empat kriteria kualitas untuk sebuah sistem penilaian esai otomatis yang baik yakni *accuracy, defensibility, coachability* dan *cost-efficiency.* Untuk sebuah sistem agar dapat memenuhi standar, setidaknya sistem tersebut harus bisa memenuhi keempat kriteria diatas. Sebuah sistem yang akurat mampu menghasilkan nilai yang dapat dipertanggungjawabkan serta dapat diukur korelasinya dengan nilai yang dihasilkan oleh pemeriksa manusia. Agar dapat menjadi sistem yang dapat dipertanggungjawabkan *(defensible)* maka prosedur penilaian yang digunakan oleh sistem harus bisa ditelusuri dan dapat dipertanggungjawabkan secara akademik. Dalam kata lain, sistem yang ada harus dapat disesuaikan secara rasional dan metode serta kriteria penilaian dari sistem yang ada dapat dijelaskan dengan baik. *Coachability* merujuk kepada keterbukaan dari metode penilaian yang ada. Jika sistem didasarkan pada metode yang sederhana, hanya memperhitungkan jawaban dengan menggunakan kata kunci tertentu yang diberi bobot tetapi tidak mempertimbangkan makna dan isi dari konten secara keseluruhan, pengguna dalam hal ini mahasiswa secara teoritis pada akhirnya seiring waktu dimungkinkan sekali dapat mengetahui cara kerja sistem penilai esai jenis ini, sehingga mereka sendiri dapat mencoba untuk mengagalkan kerja sistem dengan memanfaatkan celah yang ada sehingga mereka mendapatkan nilai yang lebih tinggi dari yang selayaknya. Jelas sekali bahwa sebuah sistem penilaian esai otomatis juga harus efisien dalam hal biaya karena tujuan utama dari sistem ini sebenarnya adalah untuk mengurangi biaya total penilaian.

Dari keempat persyaratan diatas, akurasi merupakan parameter yang paling mudah diperhitungkan, hasil yang dilaporkan pada tahun 1960-an menunjukkan bahwa sistem penilai esai otomatis dapat melakukan penilaian esai sama akuratnya dengan penilai manusia [2]. Karena efektivitas biaya dari sistem bergantung pada jumlah esai yang sudah dinilai oleh penilai manusia sebagai referensi (besarnya *data training*), oleh karena itu amat tidak masuk akal untuk mengumpulkan beberapa ratus esai yang sudah dinilai oleh pemeriksa manusia hanya untuk memeriksa beberapa esai saja. Dari sudut pandang ini bisa diambil suatu kesimpulan bahwa metode yang berbasis LSA adalah metode yang paling efektif karena LSA dapat menggunakan bahan pembelajaran tidak harus esai yang

sudah dinilai oleh pemeriksa manusia sebagai dasar untuk melakukan penilaian bagi sistem. Persyaratan yang paling problematik bagi sistem penilai esai otomatis mungkin adalah *coachability*. Jika seorang pengguna sistem dalam hal ini mahasiswa mengetahui dasar dari proses penilaian, mahasiswa ini dapat mengarahkan sistem untuk menghasilkan nilai yang lebih baik dari yang seharusnya. Salah satu kemungkinan solusi untuk persoalan ini yakni secara otomatis mengidentifikasi esai yang mencurigakan dan melaporkannya kepada pemeriksa esai manusia untuk dilakukan pengecekan secara manual. Tabel 2.2 menampilkan perbandingan dari beberapa sistem penilai esai otomatis yang sudah pernah dibuat.

Tabel 2.2 Perbandingan Lima Sistem Penilai Esai Otomatis

| Metode | Akurasi | *Defensibility* |
|---|---|---|
| PEG | Menilai sama akuratnya dengan penilai manusia diukur dari tingkat korelasinya | Bergantung sekali terhadap pengukuran kata-kata yang digunakan sehingga algoritma penilaian mudah untuk diketahui dan oleh karenanya bisa mengakibatkan pemanfaatan sistem untuk menghasilkan nilai yang lebih dari seharusnya |
| TCT | Sama seperti PEG | Memperhitungkan penggunaan kata dan makna keseluruhan dari esai yang ditulis, dan oleh karena itu lebih dapat dipertanggungjawabkan dibanding sistem PEG |
| BETSY | Sama seperti PEG | Hanya memperhitungkan konten esai |
| LSA | Sama seperti PEG | Seperti BETSY |
| E-RATER | Sama seperti PEG | Salah satu contoh sistem terbaik dalam hal kehandalannya |

Tabel 2.2 Perbandingan Lima Sistem Penilai Esai Otomatis (lanjutan)

| Metode | *Coachability* | Biaya |
|---|---|---|
| PEG | Karena pengukuran yang diterapkan sederhana maka *coachability* bisa mengakibatkan persoalan baru | Memerlukan banyak esai pembanding yang sudah diberi nilai terlebih dahulu, akan tetapi dalam proses komputasi membutuhkan biaya yang lebih sedikit dibanding metode lain, contohnya LSA |
| TCT | Memperhitungkan kata-kata yang digunakan dan kesatuan makna dari esai yang ditulis, oleh karena itu nilai yang dihasilkan lebih dapat dipertanggungjawabkan dibandingkan dengan PEG | Seperti PEG |
| BETSY | Hanya memperhitungkan konten dari esai | Seperti PEG |
| LSA | Seperti BETSY | Sebagai tambahan dari esai pembanding yang sudah diberi nilai terlebih dahulu sebelumnya, bahan pengajaran bisa digunakan untuk sebagai materi pelatihan *(training)* |
| E-RATER | Juga menganalisa struktur dan organisasi | Setidaknya 270 esai diperlukan untuk pelatihan *(training)* |

# BAB 3
# PERANCANGAN SISTEM PENILAI ESAI OTOMATIS BERBASIS ALGORITMA GLSA *(GENERALIZED LATENT SEMANTIC ANALYSIS)*

Dari studi literatur yang sudah dilakukan dapat dibuat sebuah hipotesa bahwa performa sistem penilai esai otomatis yang merepresentasikan dokumen sebagai vektor dokumen *GLSA (Generalized Latent Semantic Analysis*) mungkin dapat lebih unggul daripada sistem penilai esai otomatis yang menggunakan algoritma *Latent Semantic Analysis (LSA).*

## 3.1 Spesifikasi Sistem

### 3.1.1 Spesifikasi Perangkat Keras

Perangkat keras yang digunakan dalam perancangan sistem penilai esai otomatis ini adalah:

- Laptop

  Prosessor : Pentium(R) Dual-Core CPU T4500 @ 2.30 GHz
  RAM : 4,00 GB (3,87 GB usable)
  Harddisk : 160 GB

### 3.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan dalam perancangan sistem penilai esai otomatis berbasis GLSA memiliki spesifikasi sebagai berikut:

- Apache versi 2.4.2

Untuk menjalankan PHP dan phpMyAdmin dibutuhkan *web server*. Apache berfungsi sebagai perangkat lunak yang berfungsi untuk membuat *web server*.

- phpMyAdmin versi 3.5.1

Perangkat lunak yang berfungsi untuk menangani tempat penyimpanan data *(database).*

- Sistem Operasi Linux Ubuntu versi 10.10

Perangkat lunak sebagai sebuah antarmuka antara pengguna dengan perangkat keras sistem. Pada percobaan ini digunakan Ubuntu versi 10.10.

- PHP versi 5.3.3

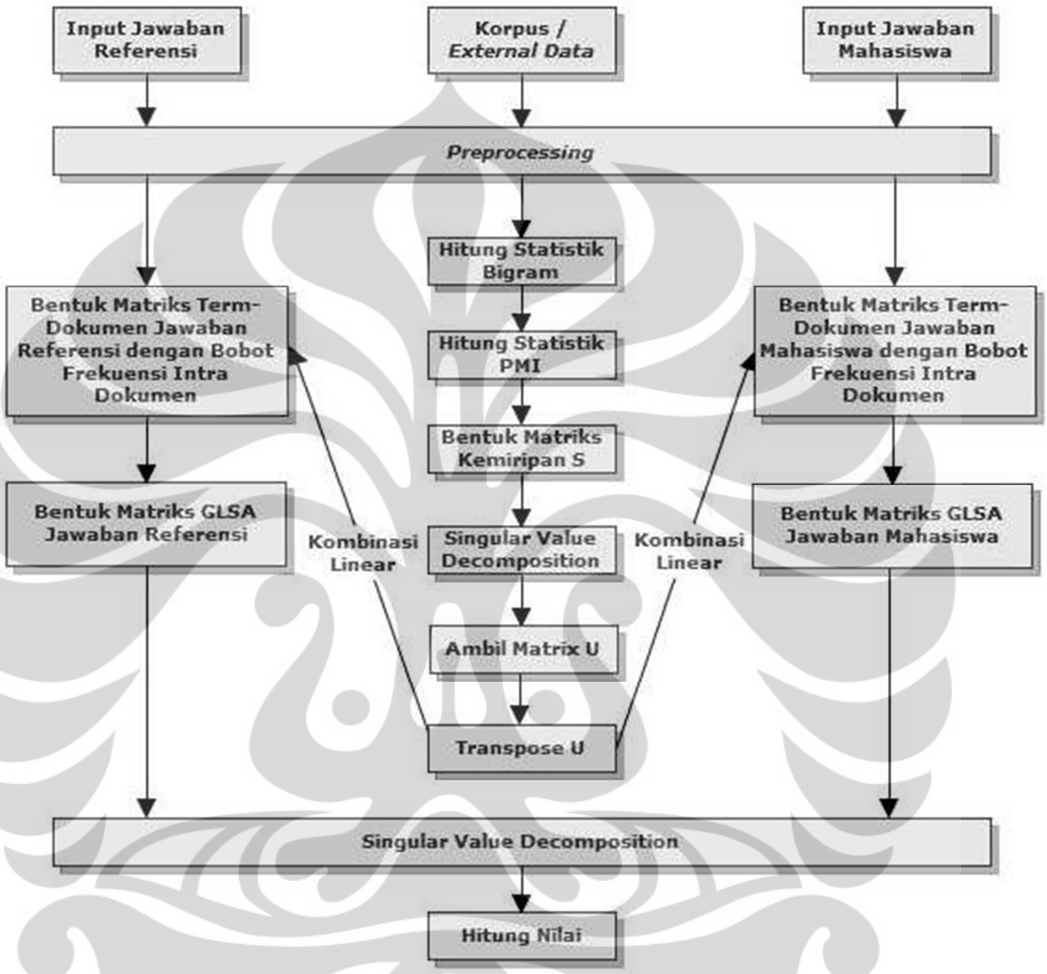
Bahasa pemrograman yang digunakan adalah PHP oleh karena itu dibutuhkan *library* fungsi-fungsi PHP. Versi library PHP yang digunakan adalah versi 5.3.3.

- Text-NSP versi 1.2.5

Perangkat lunak terbuka berbasis Perl yang digunakan untuk menghitung statistik kebahasaan dari dokumen berupa tulisan. Ada dua fungsi yang digunakan pada percobaan ini yaitu fungsi untuk menghitung statistik kemunculan bigram dan fungsi untuk menghitung nilai statistik PMI.

## 3.2 Arsitektur Sistem

Arsitektur sistem dibagi menjadi tiga bagian utama yakni pertama pembentukan matriks kemiripan S, pembentukan matriks dokumen GLSA dan ketiga penilaian dokumen.




Gambar 3.1 Rancangan Sistem Penilai Esai Berbasis GLSA

### 3.2.1 Pembentukan Matriks Kemiripan S

Pertama-tama dipilih data korpus yang memiliki kesamaan topik dengan soal yang akan diujikan ataupun korpus umum yang dapat mencakup sebagian besar / semua kata yang digunakan pada jawaban. Pada percobaan ini korpus yang digunakan merupakan kumpulan artikel, tutorial, dan bahan kuliah yang berhubungan dengan topik soal dalam hal ini adalah soal ujian dengan tema jaringan komputer yang dipilih secara acak. Setelah semuanya digabung menjadi satu, ukuran korpus yang didapat berisi 6.615 kata.

Setelah itu korpus dibersihkan dengan menghilangkan karakter non alfabet. Kemudian menggunakan korpus ini, statistik kemunculan *bigram* dihitung. Setelah itu statistik *bigram* ini diproses lebih lanjut untuk menghasilkan nilai statistik PMI antar kata pada korpus dengan *sliding window* sama dengan enam belas. Ukuran *sliding window* yang digunakan sebesar enam belas sesuai dengan hasil penelitian [23] yang menunjukkan bahwa hasil optimal dicapai pada ukuran window ini. Nilai statistik PMI ini kemudian disimpan pada *database*.

Matriks kemiripan *S* dibangun dengan mengisikan nilai PMI pasangan tiap kata yang muncul pada jawaban yang diperoleh dari *database*. Matriks ini dapat digambarkan sebagai matriks yang menjelaskan jarak / kedekatan antar kata. Matriks ini diproses lebih lanjut dengan proses SVD untuk memperoleh matriks ortogonal kiri U yang nantinya akan digunakan dalam proses kombinasi linear dengan matriks dokumen *D*.

Algoritma pembentukan matriks kemiripan *S* dapat diringkas menjadi poin-poin sebagai berikut:

- Langkah 1 : Input korpus
- Langkah 2 : Bersihkan korpus
- Langkah 3 : Hitung statistik *bigram*
- Langkah 4 : Hitung statistik PMI dengan *sliding window* 16
- Langkah 5 : Petakan nilai PMI pasangan tiap kata pada jawaban pada matriks kemiripan *S*
- Langkah 6 : Dekomposisi matriks kemiripan *S* untuk mendapatkan matriks ortogonal kiri *U*

Berikut *pseudocode* untuk menghitung statistik *bigram* dan PMI:

- Pecah korpus menjadi *bigram*
- Hitung frekuensi kemunculan untuk tiap *bigram* pada korpus
- Untuk tiap bigram:
- Hitung peluang kemunculan kata pertama pada korpus
- Hitung peluang kemunculan kata kedua pada korpus
- Hitung peluang kemunculan *bigram* pada korpus

- Kalikan peluang kemunculan kata pertama dengan kata kedua
- Bagi nilai kemunculan bigram dengan hasil kali tersebut
- Hitung nilai logaritma dari hasil tersebut sebagai nilai PMI

Proses dekomposisi menggunakan fungsi SVD dari PHP Java Matrix. PHP Java Matrix dipilih karena memudahkan sistem melakukan kalkulasi yang lebih efisien dan cepat ketimbang membuat sendiri fungsi tersebut. Setelah dekomposisi didapatkan matriks ortogonal kiri U yang akan digunakan untuk proses perhitungan berikutnya.

### 3.2.2 Pembentukan Matriks Dokumen GLSA

Ini merupakan tahap *preprocessing* untuk setiap jawaban baik jawaban referensi ataupun jawaban mahasiswa sebelum proses kalkulasi. Pada tahap ini dilakukan pembersihan imbuhan pada kata *(stemming)* dan pembersihan *stopwords* termasuk kata-kata umum *(common words)* yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna signifikan pada jawaban dan data korpus. Contoh *stopwords* untuk bahasa Inggris diantaranya *of* dan *the*. Sedangkan untuk bahasa Indonesia diantaranya "yang", "dan", "atau". Setelah melewati proses ini, hanya tersisa kata pokok tanpa imbuhan dan *stopwords*.

Matriks jawaban *D* dibentuk menggunakan nilai frekuensi kemunculan setiap kata dengan pembobotan frekuensi intra dokumen pada dokumen dengan *window* sebesar sepuluh. Matriks $U^T$ yang sudah diperoleh pada proses pembentukan matriks kemiripan *S* kemudian dikombinasi linear dengan matriks jawaban *D* untuk menghasilkan matriks dokumen GLSA $\widehat{D}$. Proses operasi matriks menggunakan *library* PHP Java Matrix (JAMA).

Algoritma pembentukan matriks dokumen GLSA dapat diringkas menjadi poin-poin sebagai berikut:

- Langkah 1 : Bersihkan jawaban dari *stopwords*
- Langkah 2 : Bersihkan jawaban dari karakter non alfabetik
- Langkah 3 : *Stemming* jawaban
- Langkah 4 : Jawaban dibagi menjadi sejumlah dokumen dengan ukuran *window* tertentu

- Langkah 5 : Hitung frekuensi intra dokumen tiap kata pada tiap dokumen
- Langkah 6 : Ambil matriks ortogonal kiri U
- Langkah 7 : Transpose matriks ortogonal kiri U menjadi $U^T$
- Langkah 8 : Kombinasi linear matriks ortogonal kiri yang sudah ditranspose $U^T$ dengan matriks dokumen *D* untuk menghasilkan matriks dokumen GLSA jawaban

*Pseudocode* untuk membersihkan jawaban dari *stopwords*:

- Cari kata pada input yang percis sama dengan kata pada array yang berisi daftar *stopwords*
- Jika ditemukan hilangkan kata tersebut
- Kembalikan input yang sudah dibersihkan

*Pseudocode* untuk membersihkan jawaban dari karakter non-alfabetik:

- Jika karakter pada input bukan karakter alfabetik maka hapus karakter tersebut
- Kembalikan input yang sudah dibersihkan

*Pseudocode* untuk *stemming* jawaban:

- Cari jika ada kata pada input yang sama dengan kata pada *database stemming* yang berisi persamaan kata berimbuhan dan kata dasarnya
- Jika ditemukan maka ubah kata tersebut menjadi kata dasarnya

Pada proses kombinasi linear antara matriks ortogonal kiri $U^T$ dengan matriks dokumen *D* digunakan fungsi perkalian matriks dari *library* PHP Java Matrix untuk mempermudah perhitungan.

### 3.2.3 Penilaian Dokumen

Perhitungan nilai jawaban mahasiswa dilakukan dengan membandingkan nilai kosinus sudut antara vektor jawaban referensi dengan vektor jawaban mahasiswa. Nilai perbandingan ini kemudian digunakan sebagai nilai mahasiswa. Algoritma penilaian jawaban mahasiswa:

- Langkah 1 : Hitung nilai *cosine similarity* antara matriks dokumen GLSA jawaban mahasiswa dengan matriks dokumen GLSA jawaban referensi
- Langkah 2 : Tetapkan nilai akhir

Proses kalkulasi nilai *cosine similarity* menggunakan fungsi operasi matriks dari *library* PHP Java Matrix untuk mempermudah perhitungan. Pada proses perhitungan, input yang digunakan adalah matriks jawaban referensi dan matriks jawaban mahasiswa.

# BAB 4
# UJI COBA DAN ANALISA

## 4.1 Sistem Pengujian

Uji coba dilakukan pada sebuah kelas kecil dengan tiga puluh mahasiswa. Tiap mahasiswa diberikan tugas menjawab dua buah soal ujian esai yang identik dan masing-masing akan dinilai secara terpisah oleh sistem.

## 4.2 Hasil Pengujian

Berikut akan dijelaskan mengenai hasil yang didapat dari pengujian pengaplikasian algoritma *GLSA (Generalized Latent Semantic Analysis)* pada sistem penilai esai otomatis dan perbandingannya dengan hasil yang didapat dari pengujian algoritma LSA untuk pasangan soal dan jawaban yang sama serta kedekatannya dengan nilai dari pemeriksa manusia *(human raters)*. Tabel 4.1 berisi soal yang digunakan beserta jawaban referensi dari dosen. Tabel 4.2 berisi nilai *human rater*.

Tabel 4.1 Soal dan Jawaban Dosen

| Soal | Soal | Jawaban Dosen |
|---|---|---|
| 1 | Jelaskan perbedaan antara peer to peer dengan client server! | Jaringan peer to peer adalah model jaringan yang mana dua atau lebih komputer yang berhubungan melalui jaringan dimana dapat berbagi pakai (share atau sharing) sumber daya tanpa memiliki server tertentu. Suatu device yang berada pada jaringan tersebut, dapat bertindak sebagai baik client maupun server pada komunikasi yang sama, atau bisa berfungsi sebagai server atau client sesuai permintaan.<br>Sedangkan jaringan client server adalah jaringan yang memiliki komputer client dan komputer server. Client meminta informasi atau layanan dari server dan server menyediakan informasi atau layanan yang diminta. Model jaringan client server ini menyediakan sekuriti dan kontrol untuk jaringan, dan server diurus oleh administrasi jaringan. |
| 2 | Jelaskan macam-macam media trasmisi yang anda ketahui! | Media transmisi yang sering dipakai dalam suatu jaringan adalah :<br>Twisted pair :<br>Sepasang kabel twist pair membentuk untaian yang mentransmisikan data. Kabel twisted pair memberikan proteksi terhadap crosstalk (electrical noise) karena cancellation effect<br>Coaxial :<br>Merupakan kabel yang mempunyai inti tembaga dan dikelilingi dengan lapisan tebal. Mempunyai banyak tipe seperti Thicknet/10Base5 yang mampu beroperasi pada 10 megabits per detik dengan panjang maksimum 500 m, Thinnet/10Base2 yang mampu beroperasi pada 10 megabits per detik dengan panjang maks 185 m, RG-59 |

Tabel 4.1 Soal dan Jawaban Dosen (Lanjutan)

| | | |
|---|---|---|
| | | yang biasa digunakan untuk TV kabel, dan RG-6 yang mempunyai kualitas lebih tinggi dari RG-59.<br>Fiberoptik : Merupakan kabel yang mempunyai gelas atau plastic strand yang mampu mentransmisikan informasi menggunakan sinar dan terbuat dari satu atau lebih fiber optik yang dijadikan satu dalam "jaket", dan mempunyai kelebihan tidak terpengaruh oleh interferensi radio atau gelombang lainnya dan mempunyai sinyal lebih jelas, jauh, dan mempunyai bandwidth lebih besar daripada kabel tembaga. Mempunyai 2 tipe yaitu Multimode dan Singlemode.<br>Wireless : Merupakan media transmisi yang tidak memakai kabel. Dianggap lebih efisien karena mampu mencakup area yang dianggap lebih luas dibandingkan memakai kabel, namun kadang bandwidth yang dipakai lebih kecil dibandingkan media transmisi kabel |

Tabel 4.2 Nilai Dosen *(Human Raters)*

| No | User Name | Nilai Soal 1 | Nilai Soal 2 |
|---|---|---|---|
| 1 | user1 | 60 | 100 |
| 2 | user2 | 75 | 100 |
| 3 | user3 | 75 | 100 |
| 4 | user4 | 75 | 100 |
| 5 | user5 | 75 | 100 |
| 6 | user6 | 60 | 100 |
| 7 | user7 | 75 | 100 |
| 8 | user8 | 70 | 90 |
| 9 | user9 | 75 | 100 |
| 10 | user10 | 40 | 35 |
| 11 | user11 | 75 | 100 |
| 12 | user12 | 90 | 100 |
| 13 | user13 | 65 | 90 |
| 14 | user14 | 40 | 100 |
| 15 | user15 | 70 | 80 |
| 16 | user16 | 70 | 100 |
| 17 | user17 | 70 | 100 |
| 18 | user18 | 90 | 90 |
| 19 | user19 | 70 | 100 |
| 20 | user20 | 90 | 90 |
| 21 | user21 | 75 | 100 |
| 22 | user22 | 60 | 85 |
| 23 | user23 | 70 | 100 |
| 24 | user24 | 65 | 100 |
| 25 | user25 | 65 | 80 |
| 26 | user26 | 65 | 80 |
| 27 | user27 | 80 | 100 |
| 28 | user28 | 75 | 100 |
| 29 | user29 | 60 | 100 |
| 30 | user30 | 75 | 100 |

### 4.2.1 Sistem Penilai Esai Otomatis dengan Algoritma *Latent Semantic Analysis (LSA)*

Pengujian pertama dilakukan pada sistem penilai esai otomatis yang menggunakan algoritma *Latent Semantic Analysis (LSA)* sebagai *engine* untuk

melakukan perhitungan. Adapun parameter-parameter yang diamati yakni nilai keluaran sistem dan waktu prosesnya (dalam detik).

Tabel 4.3 Nilai yang Dihasilkan Sistem Penilai Esai Otomatis Berbasis Algoritma LSA *(Latent Semantic Analysis)*

| No | User Name | Nilai Soal 1 | Nilai Soal 2 |
|---|---|---|---|
| 1 | user1 | 83,4058 | 98,0581 |
| 2 | user2 | 60,7919 | 100 |
| 3 | user3 | 60,7919 | 98,0581 |
| 4 | user4 | 60,7919 | 98,0581 |
| 5 | user5 | 60,7919 | 99,0338 |
| 6 | user6 | 46,6252 | 100 |
| 7 | user7 | 60,7919 | 98,0581 |
| 8 | user8 | 51,0754 | 100 |
| 9 | user9 | 62,5543 | 98,0581 |
| 10 | user10 | 29,4884 | 0 |
| 11 | user11 | 60,7919 | 100 |
| 12 | user12 | 67,5664 | 100 |
| 13 | user13 | 48,901 | 100 |
| 14 | user14 | 46,6252 | 98,0581 |
| 15 | user15 | 82,0922 | 77,211 |
| 16 | user16 | 58,9768 | 99,0338 |
| 17 | user17 | 46,6252 | 99,0338 |
| 18 | user18 | 76,6131 | 100 |
| 19 | user19 | 53,161 | 96,0769 |
| 20 | user20 | 76,6131 | 100 |
| 21 | user21 | 60,7919 | 99,0338 |
| 22 | user22 | 48,901 | 100 |
| 23 | user23 | 60,7919 | 98,0581 |
| 24 | user24 | 72,2315 | 96,0769 |
| 25 | user25 | 48,901 | 100 |
| 26 | user26 | 48,901 | 100 |
| 27 | user27 | 70,7107 | 100 |
| 28 | user28 | 60,7919 | 98,0581 |
| 29 | user29 | 39,0095 | 98,0581 |
| 30 | user30 | 60,7919 | 98,0581 |

Tabel 4.4 Waktu Proses pada Sistem Penilai Esai Otomatis Berbasis Algoritma LSA *(Latent Semantic Analysis)*

| No | User Name | Soal 1 (detik) | Soal 2 (detik) |
|---|---|---|---|
| 1 | user1 | 0,130998135 | 0,278285027 |
| 2 | user2 | 0,089365005 | 0,292762041 |
| 3 | user3 | 0,088707209 | 0,271991968 |
| 4 | user4 | 0,089690924 | 0,283885956 |
| 5 | user5 | 0,089773178 | 0,271498203 |
| 6 | user6 | 0,051747084 | 0,265107155 |
| 7 | user7 | 0,089668036 | 0,273854971 |
| 8 | user8 | 0,054693937 | 0,510625124 |
| 9 | user9 | 0,088588953 | 0,27244091 |
| 10 | user10 | 0,029855013 | 0,029687166 |
| 11 | user11 | 0,088072777 | 0,250650883 |
| 12 | user12 | 0,068680048 | 1,025436163 |
| 13 | user13 | 0,043226957 | 0,254137993 |
| 14 | user14 | 0,039054155 | 0,263470173 |
| 15 | user15 | 0,064611912 | 0,097408056 |
| 16 | user16 | 0,073848963 | 0,260695934 |
| 17 | user17 | 0,059654951 | 0,240684032 |
| 18 | user18 | 0,12277317 | 0,366548061 |
| 19 | user19 | 0,076918125 | 0,239915133 |
| 20 | user20 | 0,124614954 | 0,366091967 |
| 21 | user21 | 0,088058949 | 0,254081964 |
| 22 | user22 | 0,043833971 | 0,262613058 |
| 23 | user23 | 0,090465069 | 0,26977706 |
| 24 | user24 | 0,072834015 | 0,233178139 |
| 25 | user25 | 0,043138027 | 0,254379034 |
| 26 | user26 | 0,043736935 | 0,251111031 |
| 27 | user27 | 0,081948042 | 0,332823038 |
| 28 | user28 | 0,089406013 | 0,278055906 |
| 29 | user29 | 0,032862902 | 0,278111935 |
| 30 | user30 | 0,089071989 | 0,27561307 |

### 4.2.2 Sistem Penilai Esai Otomatis dengan Algoritma *Generalized Latent Semantic Analysis (GLSA)*

Pada pengujian ini, akan dilakukan penilaian pada sistem penilai esai otomatis yang menggunakan algoritma *Generalized Latent Semantic Analysis*

*(GLSA).* Adapun parameter-parameter yang diamati yakni nilai keluaran sistem dan waktu prosesnya (dalam detik).

Tabel 4.5 Nilai yang Dihasilkan Sistem Penilai Esai Otomatis Berbasis Algoritma *GLSA (Generalized Latent Semantic Analysis)*

| No | User Name | Nilai Soal 1 | Nilai Soal 2 |
|---|---|---|---|
| 1 | user1 | 82,2179 | 75,6171 |
| 2 | user2 | 74,3689 | 69,8227 |
| 3 | user3 | 74,3689 | 100 |
| 4 | user4 | 74,3689 | 100 |
| 5 | user5 | 74,3689 | 100 |
| 6 | user6 | 55,9329 | 100 |
| 7 | user7 | 74,3689 | 100 |
| 8 | user8 | 60,7219 | 100 |
| 9 | user9 | 75,8564 | 100 |
| 10 | user10 | 49,0126 | 13,1033 |
| 11 | user11 | 74,3689 | 66,0888 |
| 12 | user12 | 78,3917 | 100 |
| 13 | user13 | 63,422 | 83,7885 |
| 14 | user14 | 51,2416 | 60,6933 |
| 15 | user15 | 66,0117 | 37,0618 |
| 16 | user16 | 70,5129 | 100 |
| 17 | user17 | 63,422 | 69,3106 |
| 18 | user18 | 93,9511 | 87,276 |
| 19 | user19 | 79,8042 | 100 |
| 20 | user20 | 93,9511 | 87,276 |
| 21 | user21 | 74,3689 | 100 |
| 22 | user22 | 63,422 | 83,7885 |
| 23 | user23 | 74,3689 | 100 |
| 24 | user24 | 87,1652 | 100 |
| 25 | user25 | 63,422 | 83,7885 |
| 26 | user26 | 63,422 | 82,9668 |
| 27 | user27 | 100 | 100 |
| 28 | user28 | 74,3689 | 100 |
| 29 | user29 | 51,7838 | 100 |
| 30 | user30 | 74,3689 | 100 |

Tabel 4.6 Waktu Proses pada Sistem Penilai Esai Otomatis Berbasis Algoritma *GLSA (Generalized Latent Semantic Analysis)*

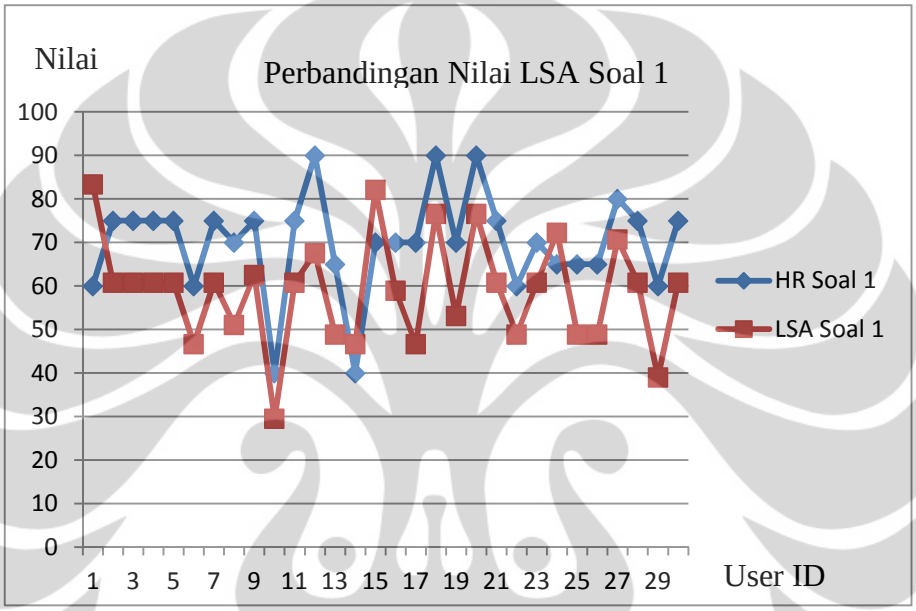
| No | User Name | Soal 1 (detik) | Soal 2 (detik) |
|---|---|---|---|
| 1 | user1 | 2,42722106 | 10,11094213 |
| 2 | user2 | 1,773919821 | 14,08115292 |
| 3 | user3 | 1,76157093 | 10,40615797 |
| 4 | user4 | 1,777443886 | 10,35972905 |
| 5 | user5 | 1,783763885 | 10,35960603 |
| 6 | user6 | 0,361378908 | 9,166823864 |
| 7 | user7 | 1,839365959 | 10,26101208 |
| 8 | user8 | 0,584374905 | 36,838938 |
| 9 | user9 | 2,213490963 | 12,50803304 |
| 10 | user10 | 0,070276976 | 0,030076981 |
| 11 | user11 | 1,842138052 | 10,83866692 |
| 12 | user12 | 1,317557096 | 153,8946989 |
| 13 | user13 | 0,22737813 | 11,97546911 |
| 14 | user14 | 0,129963875 | 12,435673 |
| 15 | user15 | 0,722945929 | 2,643706799 |
| 16 | user16 | 1,258527994 | 11,60275102 |
| 17 | user17 | 0,947381973 | 10,13577199 |
| 18 | user18 | 3,063396215 | 27,11857796 |
| 19 | user19 | 1,660125971 | 9,623934984 |
| 20 | user20 | 3,374302149 | 26,0418489 |
| 21 | user21 | 1,852134943 | 12,20184612 |
| 22 | user22 | 0,225032091 | 12,07593703 |
| 23 | user23 | 1,84848094 | 12,63638902 |
| 24 | user24 | 0,905856848 | 10,41942 |
| 25 | user25 | 0,224905014 | 12,25161004 |
| 26 | user26 | 0,232643127 | 10,97920609 |
| 27 | user27 | 1,097980976 | 17,14027214 |
| 28 | user28 | 2,063871861 | 12,13509202 |
| 29 | user29 | 0,251647949 | 10,17025185 |
| 30 | user30 | 1,859476089 | 10,47834492 |

## 4.3 Analisa Hasil

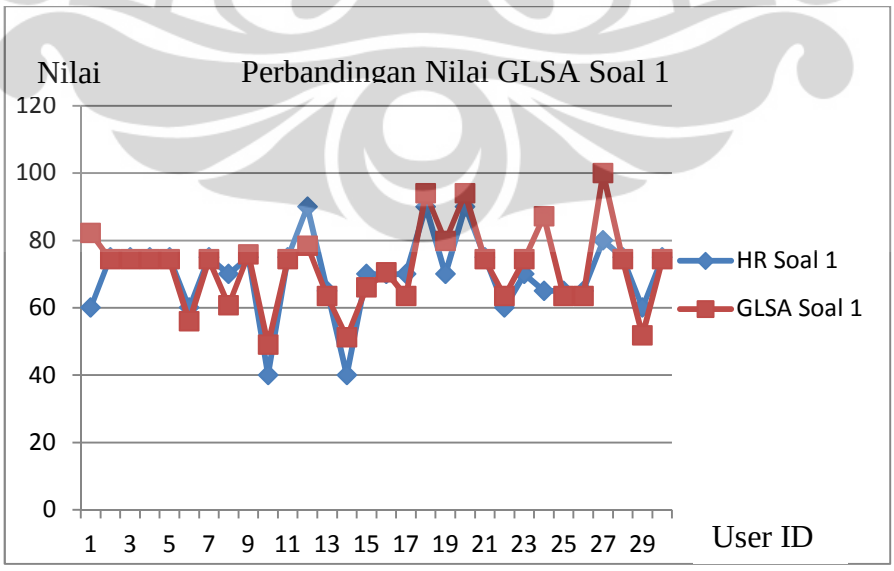
Setelah pengujian dilakukan dan hasilnya sudah didapatkan, maka sekarang akan dilakukan tahap analisa. Pada skripsi ini, hal-hal yang akan dianalisa adalah:

### 4.3.1 Analisis Pengujian 1

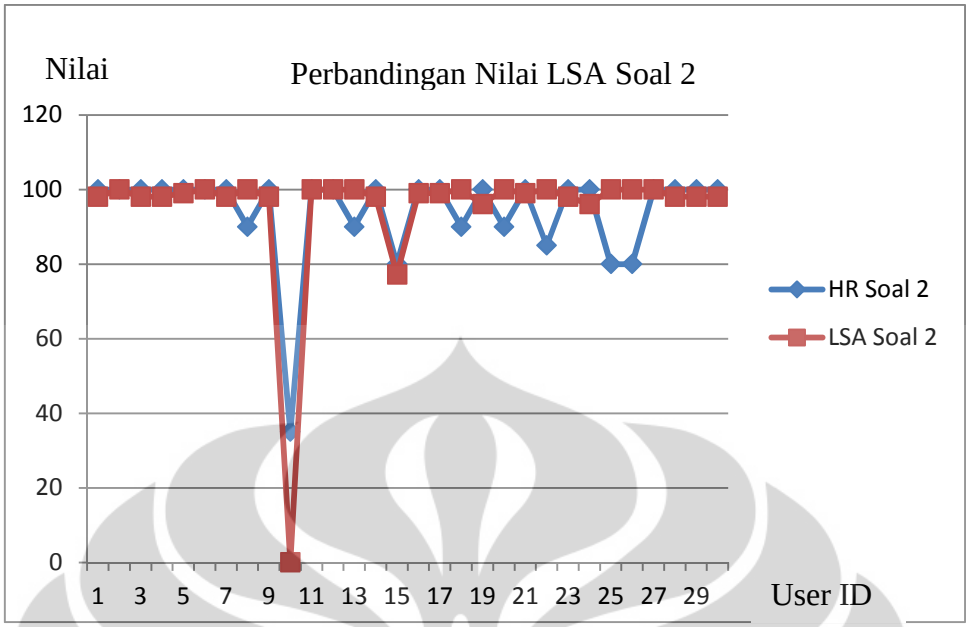
Pada analisis pengujian 1, akan digunakan dua parameter untuk membandingkan tingkat akurasi sistem, pertama dengan perbandingan nilai *Pearson Product Moment Correlation* antara nilai yang dihasilkan oleh sistem LSA dan GLSA. Nilai yang lebih tinggi mencerminkan akurasi/kedekatan yang lebih baik dengan nilai *human rater* (HR). Parameter kedua, selisih nilai sistem dengan nilai *human rater*, selisih lebih kecil menandakan sistem lebih akurat.



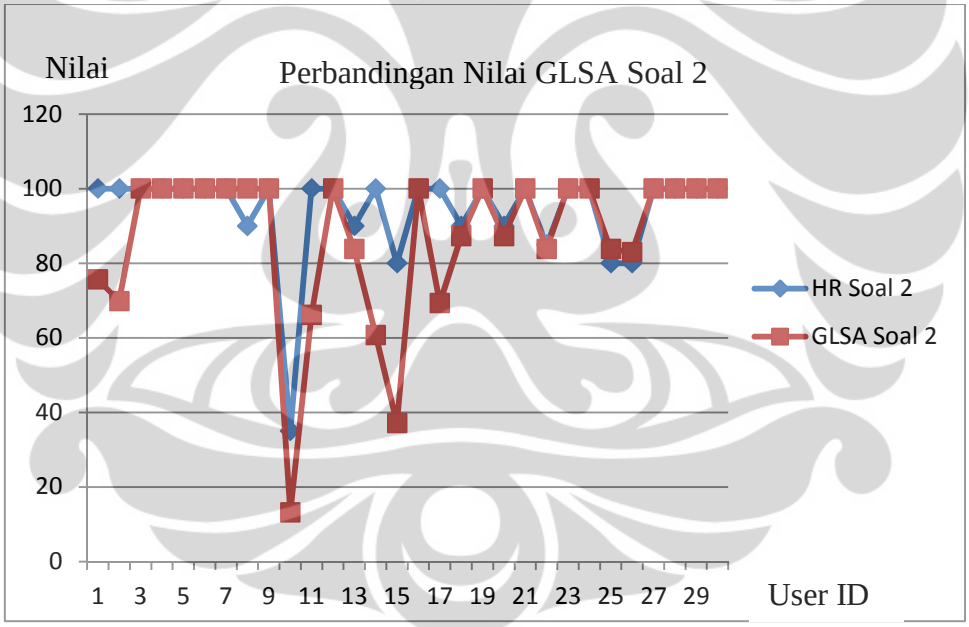

Gambar 4.1 Perbandingan Nilai LSA pada Soal 1




Gambar 4.2 Perbandingan Nilai GLSA pada Soal 1




Gambar 4.3 Perbandingan Nilai LSA pada Soal 2




Gambar 4.4 Perbandingan Nilai GLSA pada Soal 2

$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{(n-1)s_x s_y} \qquad (3.1.)$$

$$= \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\,\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

Tabel 4.7 Nilai *Pearson Product Moment Correlation* LSA dan GLSA

| No. Soal | LSA | GLSA |
|---|---|---|
| 1 | 0.577757604139928 | 0.769714067808175 |
| 2 | 0.858686001616601 | 0.733353230428142 |

Tabel 4.8 Selisih Nilai dengan *Human Rater* untuk Soal 1

| No | User Name | Δ LSA Soal 1 | Δ GLSA Soal 1 | Sistem yang Unggul |
|---|---|---|---|---|
| 1 | user1 | 23,4058 | 22,2179 | GLSA |
| 2 | user2 | 14,2081 | 0,6311 | GLSA |
| 3 | user3 | 14,2081 | 0,6311 | GLSA |
| 4 | user4 | 14,2081 | 0,6311 | GLSA |
| 5 | user5 | 14,2081 | 0,6311 | GLSA |
| 6 | user6 | 13,3748 | 4,0671 | GLSA |
| 7 | user7 | 14,2081 | 0,6311 | GLSA |
| 8 | user8 | 18,9246 | 9,2781 | GLSA |
| 9 | user9 | 12,4457 | 0,8564 | GLSA |
| 10 | user10 | 10,5116 | 9,0126 | GLSA |
| 11 | user11 | 14,2081 | 0,6311 | GLSA |
| 12 | user12 | 22,4336 | 11,6083 | GLSA |
| 13 | user13 | 16,099 | 1,578 | GLSA |
| 14 | user14 | 6,6252 | 11,2416 | LSA |
| 15 | user15 | 12,0922 | 3,9883 | GLSA |
| 16 | user16 | 11,0232 | 0,5129 | GLSA |
| 17 | user17 | 23,3748 | 6,578 | GLSA |
| 18 | user18 | 13,3869 | 3,9511 | GLSA |
| 19 | user19 | 16,839 | 9,8042 | GLSA |
| 20 | user20 | 13,3869 | 3,9511 | GLSA |
| 21 | user21 | 14,2081 | 0,6311 | GLSA |
| 22 | user22 | 11,099 | 3,422 | GLSA |
| 23 | user23 | 9,2081 | 4,3689 | GLSA |
| 24 | user24 | 7,2315 | 22,1652 | LSA |
| 25 | user25 | 16,099 | 1,578 | GLSA |
| 26 | user26 | 16,099 | 1,578 | GLSA |
| 27 | user27 | 9,2893 | 20 | LSA |
| 28 | user28 | 14,2081 | 0,6311 | GLSA |
| 29 | user29 | 20,9905 | 8,2162 | GLSA |
| 30 | user30 | 14,2081 | 0,6311 | GLSA |

Tabel 4.9 Selisih Nilai dengan *Human Rater* untuk Soal 2

| No | User Name | Δ LSA Soal 2 | Δ GLSA Soal 2 | Sistem yang Unggul |
|---|---|---|---|---|
| 1 | user1 | 1,9419 | 24,3829 | LSA |
| 2 | user2 | 0 | 30,1773 | LSA |
| 3 | user3 | 1,9419 | 0 | GLSA |
| 4 | user4 | 1,9419 | 0 | GLSA |
| 5 | user5 | 0,9662 | 0 | GLSA |
| 6 | user6 | 0 | 0 | - |
| 7 | user7 | 1,9419 | 0 | GLSA |
| 8 | user8 | 10 | 10 | - |
| 9 | user9 | 1,9419 | 0 | GLSA |
| 10 | user10 | 35 | 21,8967 | GLSA |
| 11 | user11 | 0 | 33,9112 | LSA |
| 12 | user12 | 0 | 0 | - |
| 13 | user13 | 10 | 6,2115 | GLSA |
| 14 | user14 | 1,9419 | 39,3067 | LSA |
| 15 | user15 | 2,789 | 42,9382 | LSA |
| 16 | user16 | 0,9662 | 0 | GLSA |
| 17 | user17 | 0,9662 | 30,6894 | LSA |
| 18 | user18 | 10 | 2,724 | GLSA |
| 19 | user19 | 3,9231 | 0 | GLSA |
| 20 | user20 | 10 | 2,724 | GLSA |
| 21 | user21 | 0,9662 | 0 | GLSA |
| 22 | user22 | 15 | 1,2115 | GLSA |
| 23 | user23 | 1,9419 | 0 | GLSA |
| 24 | user24 | 3,9231 | 0 | GLSA |
| 25 | user25 | 20 | 3,7885 | GLSA |
| 26 | user26 | 20 | 2,9668 | GLSA |
| 27 | user27 | 0 | 0 | - |
| 28 | user28 | 1,9419 | 0 | GLSA |
| 29 | user29 | 1,9419 | 0 | GLSA |
| 30 | user30 | 1,9419 | 0 | GLSA |

Dari hasil pengujian yang sudah dilakukan di atas, dapat dilihat bahwa pada soal nomor 1 sistem yang menggunakan algoritma GLSA memiliki nilai *Pearson Product Moment Correlation* yang secara signifikan lebih tinggi daripada

sistem yang menggunakan algoritma LSA. Hal ini mengindikasikan bahwa secara umum nilai yang dihasilkan sistem berbasis GLSA lebih dekat / akurat dengan nilai *human rater* daripada LSA. Hasil lain didapat soal nomor 2 dimana nilai *Pearson Product Moment Correlation* sistem berbasis LSA lebih tinggi. Hal dimungkinkan terjadi karena pada perhitungan menggunakan *Pearson Product Moment Correlation*, salah satu parameter yang digunakan adalah nilai standar deviasi (s). Hal ini mungkin dapat diibaratkan dengan peribahasa karena nila setitik rusak susu sebelanga. Pada enam *user* yakni *user* 1, 2, 11, 14, 15, dan 17 didapat selisih nilai yang cukup tinggi dengan *human rater* pada sistem berbasis GLSA. Hal ini mengakibatkan nilai standar deviasi (s) pada GLSA yang berfungsi sebagai pembagi memiliki nilai yang lebih besar daripada nilai standar deviasi pada LSA yang pada akhirnya mengakibatkan nilai yang dihasilkan lebih kecil karena pembagi yang lebih besar yang mengindikasikan bahwa pada soal nomor 2 sistem LSA lebih unggul jika diukur menggunakan parameter *Pearson Product Moment Correlation*. Akan tetapi dilihat dari Tabel 4.9 yang menggambarkan selisih nilai, sebenarnya GLSA yang lebih unggul secara keseluruhan dari LSA dengan perbandingan 20:6 karena dari 30 user, GLSA unggul pada 20 soal sedangkan LSA hanya unggul pada 6 soal dan sisanya nilai yang dihasilkan sama akurat. Ditinjau dari hal ini kemudian dipertimbangkan untuk menggunakan parameter kedua untuk membandingkan akurasi sistem LSA dan GLSA menggunakan selisih nilai tiap *user* dengan *human rater* lalu menentukan mana yang lebih unggul. Dengan hal ini dapat dilihat lebih jelas perbandingan unjuk kerja sistem GLSA dan LSA secara lebih proporsional untuk tiap nilai yang dihasilkan.

Tabel 4.10 Rekapitulasi Keunggulan Sistem Berbasis LSA dan GLSA

| No. Soal | GLSA | LSA | Seri |
|---|---|---|---|
| 1 | 27 | 3 | 0 |
| 2 | 20 | 6 | 4 |
| Jumlah | 47 | 9 | 4 |

Dari Tabel 4.10 dapat dilihat dengan jelas bahwa GLSA unggul pada semua pengujian. Pada soal 1, GLSA menghasilkan nilai yang lebih akurat sebanyak 27 kali berbanding 3 kali menggunakan LSA. Pada soal 2, GLSA unggul 20 kali berbanding 6 kali pada LSA dengan 4 kali diantaranya nilai yang dihasilkan sama. Dengan demikian dapat diambil kesimpulan bahwa dari 60 kali pengujian, GLSA unggul pada 47 kali pengujian atau sebesar 78,3% total pengujian sedangkan LSA hanya unggul pada 9 kali pengujian atau 15% total pengujian dengan 4 kali atau 6,7% total pengujian diantaranya akurasi yang dihasilkan sama oleh kedua sistem.

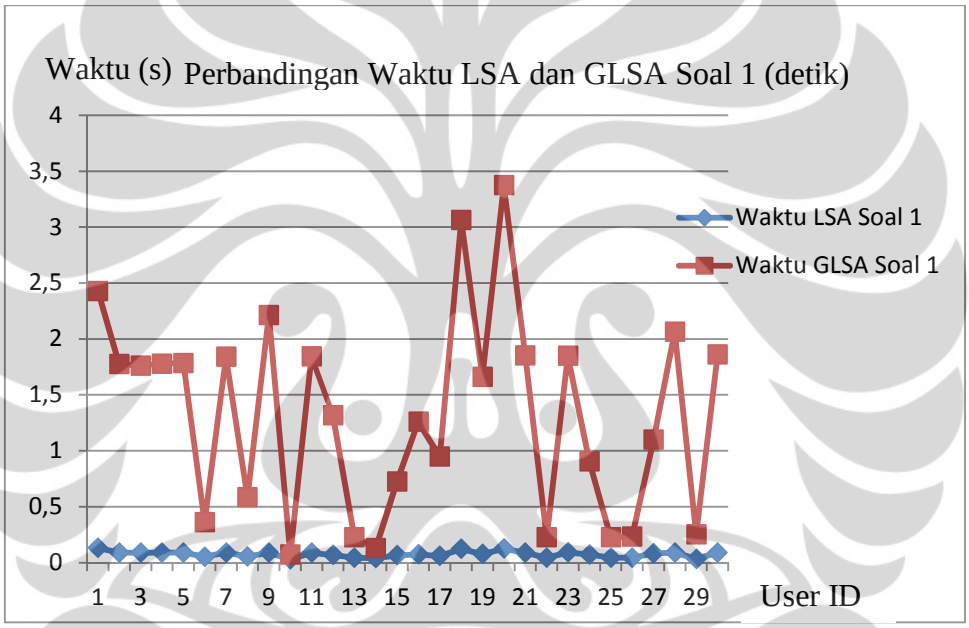
Hasil penelitian ini juga sesuai dengan hasil penelitian Irina Matveeva [23] yang menunjukkan implementasi GLSA pada sejumlah sistem berbeda mulai dari uji sinonim, klasifikasi dokumen, dan *clustering* dapat menghasilkan tingkat efektivitas yang lebih unggul daripada sistem sebelumnya.

Kesimpulan lain yang dapat diambil adalah bahwa sistem yang berbasis LSA dan GLSA yang digunakan pada percobaan ini sudah layak untuk dikategorikan sebagai sistem yang layak pakai. Hal ini merujuk pada hasil penelitian Landauer et al. (1997) [8] dan Foltz et al. (2000) [9] yang melaporkan bahwa tingkat korelasi penilaian yang dilakukan oleh dua orang pemeriksa manusia adalah sebesar 0,64 – 0,84 dan sebesar 0,59 – 0,89 untuk penilaian berbasis LSA. Hasil pengukuran korelasi yang didapat pada sistem LSA sebesar 0.57775-0.85868 dan GLSA sebesar 0.73335-0.76971. Dengan nilai korelasi yang masih berada dalam rentang tingkat korelasi *human rater*, dapat diambil kesimpulan bahwa kedua sistem ini dapat digunakan sebagai pengganti *human rater* karena nilai yang dihasilkan memiliki tingkat korelasi yang masih berada dalam rentang korelasi nilai yang dihasilkan oleh *human rater*.

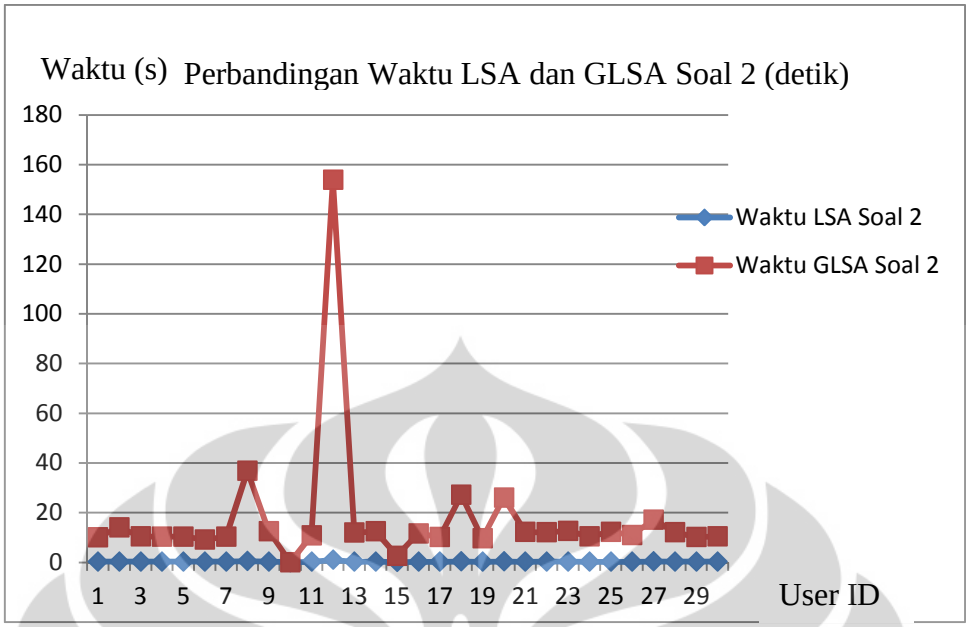
Hal lain yang didapat dari penelitian ini bahwa korelasi nilai yang dihasilkan sistem berbasis GLSA lebih stabil daripada LSA. Hal ini ditunjukkan dengan rentang nilai korelasi yang lebih sempit dibandingkan dengan rentang nilai korelasi LSA. Penelitian ini dapat dikembangkan lebih lanjut untuk menghasilkan sistem berbasis GLSA dengan tingkat akurasi yang lebih baik dari sekarang.

### 4.3.2 Analisis Pengujian 2

Pada analisis pengujian 2 ini, akan dilihat perbandingan waktu proses yang diperlukan untuk menghitung nilai mahasiswa pada sistem berbasis LSA dan GLSA.




Gambar 4.5 Perbandingan Waktu LSA dan GLSA pada Soal 1




Gambar 4.6 Perbandingan Waktu LSA dan GLSA pada Soal 2

Tabel 4.11 Rataan Waktu Proses pada Sistem LSA dan GLSA

| | **Soal 1 (detik)** | **Soal 2 (detik)** |
|---|---|---|
| Rataan Waktu Proses LSA | 0,074663313 | 0,293497372 |
| Rataan Waktu Proses GLSA | 1,323285151 | 17,36406469 |

Pada Gambar 4.5, Gambar 4.6 dan Tabel 4.11 dapat dilihat bahwa sistem penilai esai otomatis yang berbasis LSA memiliki rataan waktu proses yang lebih cepat dibanding dengan sistem penilai esai otomatis yang berbasis GLSA. Hal ini dapat dijelaskan dengan mudah, karena pada sistem GLSA terdapat proses yang lebih lama daripada LSA. Pada GLSA dibentuk matriks kesamaan dimana untuk mengisi nilai tiap sel pada matriks ini, sistem melakukan proses pencarian di *database* untuk nilai *PMI (Point-wise Mutual Information)* sedangkan pada sistem berbasis LSA hal ini tidak dilakukan. Kemudian pada proses pembentukan matriks dokumen GLSA diperlukan proses kombinasi linear yang merupakan perkalian matriks. Operasi ini tidak dilakukan pada sistem berbasis LSA. Pada proses GLSA juga dilakukan proses *SVD (Singular Value Decomposition)* sebelum pembentukan matriks dokumen GLSA dan setelah matriks dokumen terbentuk untuk menghasilkan nilai pada sistem untuk jawaban referensi dan

mahasiswa, sehingga untuk melakukan proses perhitungan pada sistem LSA hanya dibutuhkan dua kali proses SVD lain halnya dengan pada sistem GLSA yang membutuhkan empat kali proses SVD tiap kali melakukan perhitungan pada sistem.

Pada akhirnya, peningkatan pesat dalam sumber daya komputasi yang tersedia seperti *grid computing* dan pengembangan sejumlah *eigensolver* paralel membuat biaya komputasi menjadi pertimbangan sekunder [23]. Karena GLSA menunjukkan kinerja yang amat baik, amat dipercaya bahwa manfaatnya lebih besar daripada biaya komputasi.

# BAB 5
# KESIMPULAN

- Kinerja sistem penilai esai otomatis berbasis GLSA lebih unggul daripada sistem berbasis LSA. Dari 60 kali pengujian, GLSA menghasilkan nilai yang lebih akurat pada 47 kali pengujian atau 78,3% total pengujian sedangkan LSA hanya unggul pada 9 kali pengujian atau 15% total pengujian dan sisanya 4 kali pengujian atau 6,7% total pengujian menghasilkan nilai dengan tingkat akurasi yang sama.
- Nilai *Pearson Product Moment Correlation* pada percobaan menggunakan sistem LSA sebesar 0.57775-0.85868 sedangkan pada GLSA sebesar 0.73335-0.76971. Hal ini mengindikasikan bahwa sistem berbasis LSA dan GLSA yang diujikan layak pakai karena memiliki performa yang sama baiknya dengan performa *human rater* ditandai dengan nilai korelasi yang masih berada dalam rentang nilai korelasi yang dihasilkan oleh *human rater*.
- Korelasi nilai yang dihasilkan sistem berbasis GLSA lebih stabil daripada LSA. Hal ini ditunjukkan dengan rentang nilai korelasi yang lebih sempit dibandingkan dengan rentang nilai korelasi LSA, yakni sebesar 0.73335-0.76971 berbanding 0.57775-0.85868 untuk LSA.
- Waktu proses yang dibutuhkan sistem penilai esai otomatis berbasis GLSA lebih lama dibandingkan dengan LSA. Akan tetapi karena GLSA menunjukkan kinerja yang amat baik, amat dipercaya bahwa manfaatnya lebih besar daripada biaya komputasi.

## DAFTAR REFERENSI


[1] T. Kakkonen, N. Myller, E. Sutinen, and J. Timonen. 2005. *Comparison of Dimension Reduction Methods for Automated Essay Grading. Submitted.*

[2] E. B. Page. 1966. *The Imminence of Grading Essays by Computer*. Phi Delta Kappan, 47:238–243.

[3] J. Burstein. 2003. *The e-rater scoring engine: Automated essay scoring with natural language processing.* In M. D. Shermis and J. Burstein, editors, *Automated essay scoring: A cross-disciplinary perspective.* Lawrence Erlbaum Associates, Hillsdale, NJ.

[4] Rudner, L.M. and Liang, L. 2002. *National Council on Measurement in Education*, New Orleans, LA. Online: http://ericae.net/betsy/papers/n2002e.pdf

[5] P. W. Foltz, D. Laham, and T. K. Landauer. 1999. *Automated Essay Scoring: Applications to Educational Technology.* In Proc. of World Conf. Educational Multimedia, Hypermedia & Telecommunications, Seattle, USA.

[6] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser. 1999. *Approximate natural language understanding for an intelligent tutor*. In Proc. of the 12th Int'l Artificial Intelligence Research Symposium, pages 172–176, Menlo Park, CA, USA.

[7] D. Wade-Stein and E. Kintsch. 2003. *Summary street: Interactive computer support for writing.* Technical report. University of Colorado.

[8] T. K. Landauer, D. Laham, B. Rehder, and M. E. Schreiner. 1997. *How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans.* In Proc. of the 19th Annual Meeting of the Cognitive Science Society, Mawhwah, NJ. Erlbaum.

[9] Foltz, P. W., Gilliam, S., & Kendall, S. (2000). *Supporting content-based feedback in online writing evaluation with LSA. Interactive Learning Environments,* 8 (2), 111–129.

[10] Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., and Harshman R.A. , 1990, *Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science & Landauer*, T.K., Foltz, P.W., and Laham D. , 1998, *An introduction to Latent Semantic Analysis. Discourse Processes*, http://lsa.colorado.edu/pepers/dp1.LSAintro.pdf

[11] Burstein, J., et al., 1998, *Enriching automated essay scoring using discourse marking. Proceedings of the Workshop on Discourse Relations and Discourse Marking, Annual Meeting of the Association of Computational Linguistics*, Montreal, Canada

[12] Burstein, J., Leacock, C., and Swartz, R. , 2001, *Automated evaluation of essay and short answers. Proceedings of the Sixth International Computer Assisted Assessment Conference,* Loughborough University, Loughborough, UK

[13] Christie, J. R., 1999, *Automated essay marking-for both style and content. Proceedings of the Third Annual Computer Assisted Assessment Conference*, Loughborough University, Loughborough, UK

[14] Ming, P.Y., Mikhailov, A.A., and Kuan, T.L., 2000, *Intelligent essay marking system. Learners Together*, Feb 2000, NgccANN Polytechnic, Singapore http://ipdweb.np.edu.sg/lt/feb00/intelligent_essay_marking.pdf

[15] Mitchell, T., Russel, T., Broomhead, P., and Aldridge N. 2002. *Towards robust computerized marking of free-text responses. Proceedings of the Sixth International Computer Assisted Assessment Conference,* Loughborough University, Loughborough, UK

[16] Ratna, A. A. P., Budiardjo, B., Hartanto. D. 2007. SIMPLE: Sistem Penilai Esei Otomatis untuk Menilai Ujian dalam Bahasa Indonesia. Makara, Teknologi, Vol. 11, No. 1

[17] Charniak, Eugene: *Introduction to artificial intelligence*, page 2. Addison-Wesley, 1984.

[18] *The Latent Semantic Indexing home page*, http://lsa.colorado.edu, 11 Des 2011 18:09 WIB

[19] Adhitia, Rama; Purwarianti, Ayu; Penilaian Esai Jawaban Bahasa Indonesia Menggunakan Metode SVM – LSA dengan fitur Generik; Jurnal Sistem Informasi MTI UI, Volume 5, Nomor 1, ISBN 1412 – 8896, Institut Teknologi Bandung

[20] Berry, M., Dumais, S.T., O'Brien, G.W. (1994) .*Using Linear Algebra for Intelligent Information Retrieval.* SIAM Review

[21] Kakkonen, T., Myller, N., Sutinen, E., & Timonen, J. *Automatic Essay Grading with Probabilistic Latent Semantic Analysis. Proceedings of the 2nd Workshop on Building Educational Applications Using NLP*, hlm 29–36, Ann Arbor, June 2005

[22] Blei, David M.; Andrew Y. Ng, Michael I. Jordan (2003). *"Latent Dirichlet Allocation". Journal of Machine Learning Research 3*: 993–1022. doi:10.1162/jmlr.2003.3.4-5.993

[23] Matveeva, Irina. "*Generalized Latent Semantic Analysis* for Document Representation". Disertasi. University of Chicago. Chicago, USA.

[24] Olney, Andrew M. "Generalizing *Latent Semantic Analysis*". 2009 IEEE International Conference on Semantic Computing. University of Memphis. Memphis, USA.

[25] S. Valenti, F. Neri, and A. Cucchiarelli, *"An overview of current research on automated essay grading," Journal of Information Technology Education*, vol. 2, 2003, pp. 319-330.

[26] Turney, Peter D. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. Lecture Notes in Computer Science, 2167:491-502, 2001.

[27] Terra, Egidio L dan Clarke, Charles L. A. Frequency Estimates for Statistical Word Similarity Measures. HLT-NAACL, 2003.

[28] Chklovski, Timothy dan Pantel, Patrick. Verbocean: Mining The Web for Fine-Grained Semantic Verb Relations. EMNLP, 2004.

[29] Widdows, Dominic. Unsupervised Methods for Developing Taxonomies by Combining Syntactic and Statistical Information. HLT-NAACL, 2003.

[30] Cox, Trevor F dan Coxx, Micheal A. Multidimensional Scaling. CRC/Chapman and Hall, 2001.i