

IMPLEMENTASI SKEMA PEMBOBOTAN SICBI
PADA APLIKASI *ESSAY GRADING* METODE
LATENT SEMANTIC ANALYSIS

SKRIPSI

OLEH

DUDI HERMAWANDI

04 04 03 031 8



DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008

IMPLEMENTASI SKEMA PEMBOBOTAN SICBI
PADA APLIKASI *ESSAY GRADING* METODE
LATENT SEMANTIC ANALYSIS

SKRIPSI

OLEH

DUDI HERMAWANDI

04 04 03 031 8



**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN
PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008**

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

IMPLEMENTASI SKEMA PEMBOBOTAN SICBI PADA APLIKASI *ESSAY GRADING* METODE *LATENT SEMANTIC ANALYSIS*

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro, Departemen Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 17 Juli 2008

Dudi Hermawandi

NPM 04 04 03 031 8

PERSETUJUAN

Skripsi dengan judul:

**IMPLEMENTASI SKEMA PEMBOBOTAN SICBI PADA
APLIKASI *ESSAY GRADING* METODE *LATENT SEMANTIC ANALYSIS***

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Skripsi ini telah diujikan pada sidang ujian skripsi pada tanggal 7 Juli 2008 dan dinyatakan memenuhi syarat/sah sebagai skripsi pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, 17 Juli 2008

Dosen Pembimbing

Dr. Ir. Anak Agung Putri Ratna, M.Eng

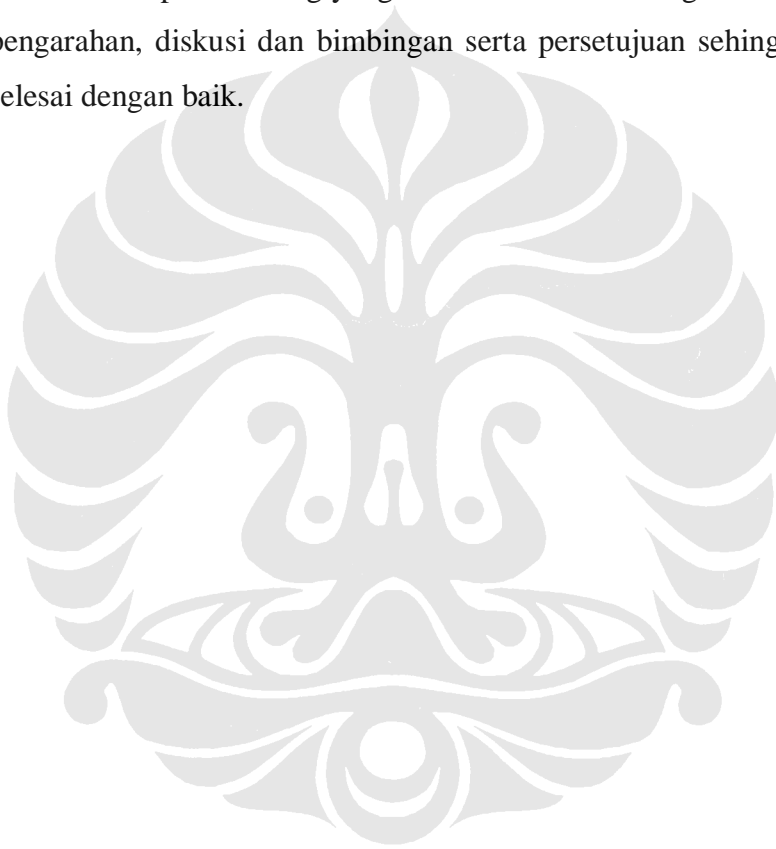
NIP. 131 865 234

UCAPAN TERIMA KASIH

Puji syukur hanya kepada ALLAH SWT, Yang Maha Berkuasa, shalawat dan salam terlimpah kepada Muhammad SAW. Alhamdulillah, tugas skripsi ini dapat diselesaikan dengan baik. Penulis mengucapkan terima kasih kepada :

Dr. Ir. Anak Agung Putri Ratna, M.Eng

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberi pengarahan, diskusi dan bimbingan serta persetujuan sehingga skripsi ini dapat selesai dengan baik.



Dudi Hermawandi
NPM 04 04 03 031 8
Departemen Teknik Elektro

Dosen Pembimbing
Dr. Ir. Anak Agung Putri Ratna, M.Eng

**IMPLEMENTASI SKEMA PEMBOBOTAN SICBI PADA APLIKASI
ESSAY GRADING METODE LATENT SEMANTIC ANALYSIS**

ABSTRAK

Salah satu metode otomatisasi *essay grading* adalah *essay grading* metode LSA. LSA merepresentasikan isi kata dalam matriks dua dimensi yang besar. Bagian pemrosesan penting dari LSA adalah komponen penganalisisan bernama SVD (*Singular Value Decomposition*) yang mengkompresi informasi yang berkaitan dalam jumlah besar ke dalam ruang yang lebih kecil. Menggunakan teknik aljabar matriks (SVD), hubungan baru antara esai mahasiswa dan esai referensi ditentukan dan dimodifikasi untuk mewakili arti sebenarnya.

SIMPLE-O adalah aplikasi penilaian esai otomatis metode LSA yang berbasis web yang dikembangkan di Indonesia. Untuk meningkatkan kualitas penilaian esai maka perlu diterapkan teknik pembobotan. Sebuah metode pembobotan merupakan susunan dari tiga buah pembobotan: pembobotan lokal (*local weighting*), pembobotan global (*global weighting*) dan normalisasi (*normalization*) [1]. Untuk mengimplementasikan pembobotan maka pada SIMPLE-O dilakukan perubahan pada bagian proses memasukan jawaban esai mahasiswa dan proses penilaiannya. SIMPLE-OM adalah SIMPLE-O yang telah mengalami perubahan. Pada SIMPLE-OM skema pembobotan yang diterapkan adalah skema pembobotan SICBI (SQRT-IGFF-COSN-BNRY-IDFB).

Berdasarkan hasil pengamatan dan perhitungan dari beberapa skenario pengujian, sistem aplikasi dengan pembobotan SICBI memberikan hasil yang lebih baik daripada sistem aplikasi tanpa pembobotan. Skenario pengujian yang memberikan hasil paling baik (mendekati *human rater*) adalah skenario yang memiliki jumlah mahasiswa terbanyak yaitu skenario 3 (20 mahasiswa). Pada skenario 3, rata-rata selisih antara penilaian sistem aplikasi dengan *human rater* adalah 10,9. Penerapan pembobotan akan membuat sistem aplikasi bekerja lebih lama dalam hal penilaian esai. Selain itu, beberapa hal lain yang berpengaruh pada kecepatan proses penilaian esai antara lain banyaknya kata kunci mahasiswa dan jumlah mahasiswa yang mengikuti ujian.

Keywords : *Essay Grading*, LSA, SVD, Pembobotan

Dudi Hermawandi
NPM 04 04 03 031 8
Electrical Engineering Departement

Counsellor
Dr. Ir. Anak Agung Putri Ratna, M.Eng

SICBI WEIGHTING SCHEME IMPLEMENTATION IN ESSAY GRADING WITH LATENT SEMANTIC ANALYSIS METHODE

ABSTRACT

One method of automatic essay grading is “LSA Essay Grading Method”. LSA represents words contained in a huge bi-dimensional matrix. Main processing part of LSA is analyzing component that called SVD (Singular Value Decomposition) which compress the large-scaled related information into smaller scale. Using matrix algebraic method (SVD), the new relations between student’s essay and the reference essay can be determined and modified in the real meaning.

SIMPLE-O is an automatic essay grading application using web-based LSA method which has been developed in Indonesia. To increase essay grading quality, it needed to apply weighting technique. Weighting methods consist of three weighting: *local weighting*, *global weighting*, and *normalization* [1]. To implement the weighting in SIMPLE-O, it needs to make changes in student’s answers and grading process. SIMPLE-OM is a modified SIMPLE-O. In SIMPLE-OM, the weighting scheme which is being implemented is SICBI (SQRT-IGFF-COSN-BNRY-IDFD) weighting scheme.

According to observation results and calculation from several testing scenario, SICBI weighting application system gives better results than application system without weighting method. The best result (approaching the human rater) is given by the testing method which has the most student participants, that is in third scenario (20 students). In this scenario, the average differences between application system grading and human rater is 10.9. Weighting implementation will make the application system work longer in essay grading. The number of word and the students also affect to the essay grading speed.

Keywords : *Essay Grading*, LSA, SVD, weighting

DAFTAR ISI

	Halaman
JUDUL	ii
PERNYATAAN KEASLIAN SKRIPSI	iii
PERSETUJUAN	iv
UCAPAN TERIMA KASIH	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
DAFTAR SINGKATAN	xiii
BAB I PENDAHULUAN	1
1.1. LATAR BELAKANG	1
1.2. TUJUAN PENULISAN	2
1.3. BATASAN MASALAH	2
1.4. SISTEMATIKA PENULISAN	2
BAB II ESSAY GRADING METODE LSA DAN LATENT SEMANTIC ANALYSIS (LSA)	3
2.1. ESSAY GRADING METODE LSA	3
2.2. SVD (SINGULAR VALUE DECOMPOSITION)	4
2.3. LSA VIA SVD	5
2.4. RELEVANSI DOKUMEN DAN QUERY	7
2.5. PROGRAM PENDUKUNG	8
2.5.1 <i>PHP</i>	9
2.5.2 <i>MySQL</i>	9
2.5.3 <i>Apache</i>	10

2.5.4	JAMA	11
2.6.	PEMBOBOTAN (<i>WEIGHTING</i>)	11
2.6.1	<i>Pembobotan Lokal</i>	12
2.6.2	<i>Pembobotan Global</i>	14
2.6.3	<i>Normalisasi</i>	15
BAB III PERANCANGAN APLIKASI & MEKANISME		
PEMBOBOTAN SICBI		17
3.1.	KONSEP APLIKASI	17
3.2.	PROSES PEMBOBOTAN	21
3.2.1	<i>Pembobotan SICBI</i>	21
3.2.2	<i>Mekanisme Pembobotan</i>	23
3.2.3	<i>Operasi Svd, Truncated Svd, Pembentukan q_k dan d_k, serta Relevansi q_k dan d_k</i>	26
BAB IV UJI COBA DAN ANALISA PENGUJIAN APLIKASI		29
4.1.	UJI COBA APLIKASI	29
4.1.1	<i>Proses Input Esai Mahasiswa</i>	31
4.1.2	<i>Proses Penilaian Ujian</i>	31
4.2.	ANALISIS PENGUJIAN APLIKASI	32
4.2.1	<i>Skenario Pengujian 1</i>	33
4.2.2	<i>Skenario Pengujian 2</i>	35
4.2.3	<i>Skenario Pengujian 3</i>	37
4.2.4	<i>Kecepatan Proses Penilaian Ujian</i>	39
4.2.5	<i>Pengaruh Jumlah Mahasiswa Terhadap Unjuk Kerja Sistem</i>	41
BAB V KESIMPULAN		41
DAFTAR ACUAN		43
DAFTAR PUSTAKA		44
LAMPIRAN		

DAFTAR GAMBAR

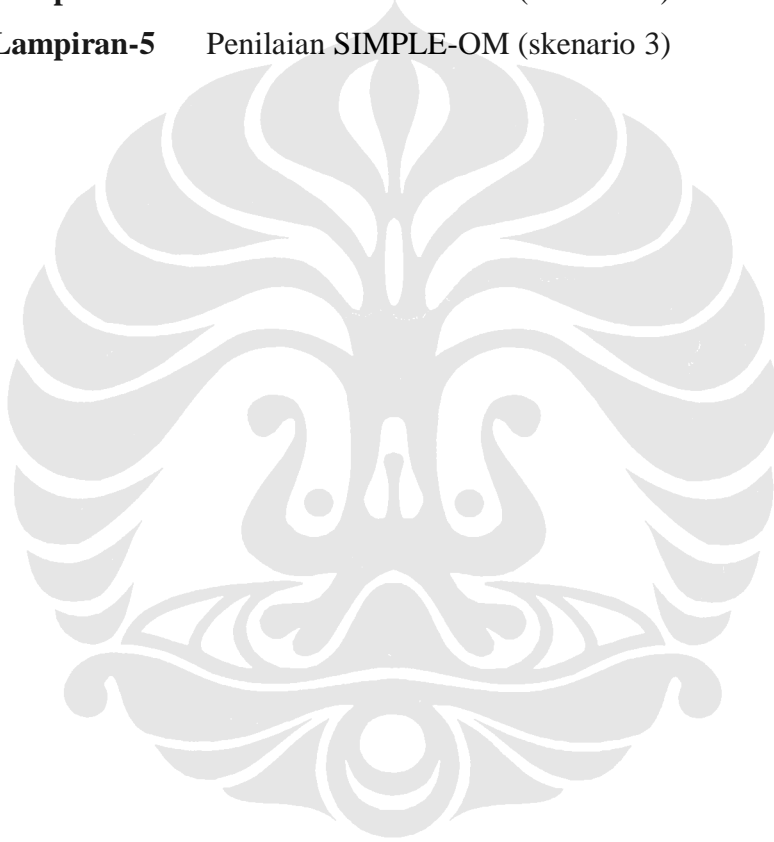
	Halaman
Gambar 2.1. Dekomposisi matriks A dengan SVD	6
Gambar 2.2. Dekomposisi SVD dengan faktor k [4]	6
Gambar 2.3. Contoh tabel dalam MySQL	10
Gambar 3.1. Use Case Diagram SIMPLE-O yang dimodifikasi (SIMPLE-OM)	17
Gambar 3.2. Activity Diagram Input Jawaban Mahasiswa	18
Gambar 3.3. Activity Diagram Input Esai Referensi (untuk dosen)	19
Gambar 3.4. Activity Diagram Proses Penilaian Esai Mahasiswa (proses LSA, ada di sisi dosen)	19
Gambar 3.5. Sequence Diagram Input Soal Jawaban Sisi Dosen	21
Gambar 3.6. Sequence Diagram Input Jawaban Sisi Mahasiswa	21
Gambar 3.7. Algoritma Proses Pemilihan Pembobotan	24
Gambar 3.8. Algoritma Fungsi Pembobotan SICBI	26
Gambar 3.9. Algoritma Pembobotan	26
Gambar 3.10. Algoritma Truncated SVD	27
Gambar 3.11. Algoritma/Rumus Perhitungan Frobenius	28
Gambar 3.12. Algoritma Proses Kalkulasi Kosinus	28
Gambar 4.1. Halaman Muka SIMPLE-O Modifikasi	30
Gambar 4.2. Halaman Muka Pengguna Admin	30
Gambar 4.3. Halaman Muka Pengguna Dosen	30
Gambar 4.4. Halaman Muka Pengguna Mahasiswa	31
Gambar 4.5. Periksa Esai Mahasiswa	32
Gambar 4.6. Grafik Skenario Pengujian 1	35
Gambar 4.7. Grafik Skenario Pengujian 2	37
Gambar 4.8. Grafik Skenario Pengujian 3	39
Gambar 4.9. Diagram Hubungan Banyak Kata Kunci Dengan Jumlah Mahasiswa	40

DAFTAR TABEL

	Halaman
Tabel 2-1. Interpretasi komponen SVD dalam LSA	6
Tabel 2-2. Macam-macam pembobotan lokal	13
Tabel 2-3. Macam-macam pembobotan global	14
Tabel 2-4. Macam-macam Normalisasi	16
Tabel 3-1. Dimensi matriks SVD	27
Tabel 4-1. Data Pengujian 1	33
Tabel 4-2. Data Pengujian 2	36
Tabel 4-3. Data Pengujian 3	38
Tabel 4-4. Waktu Untuk Proses Penilaian Esai Skenario 1, 2, dan 3	40
Tabel 4-5. Pengaruh Jumlah Mahasiswa Terhadap Unjuk Kerja Sistem Pembobotan SICBI	41

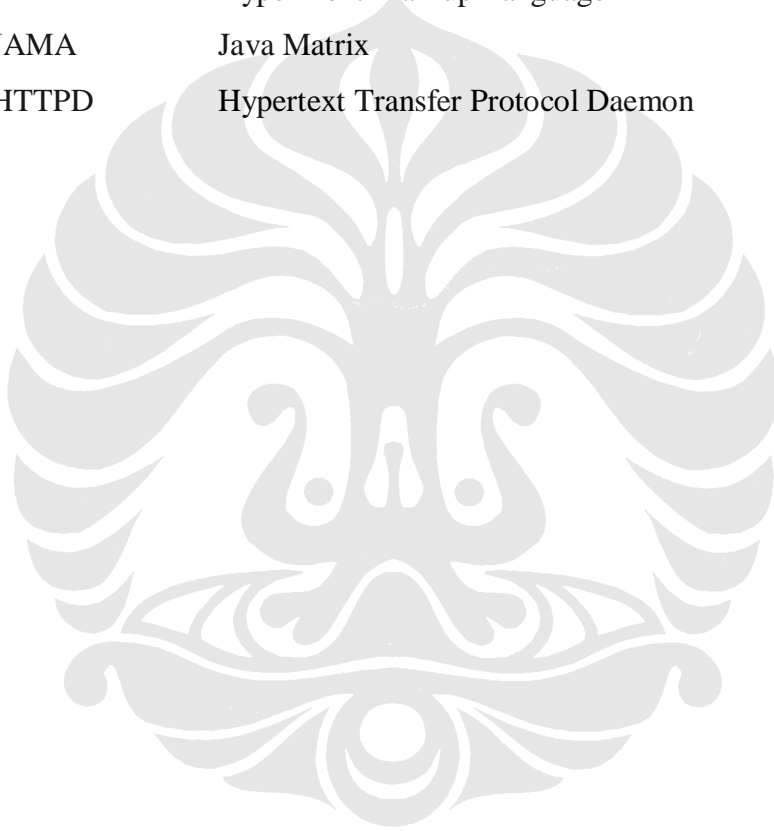
DAFTAR LAMPIRAN

	Halaman
Lampiran-1 Soal Ujian	45
Lampiran-2 Esai Referensi dan Kata Kunci Esai Referensi (query)	46
Lampiran-3 Penilaian Human Rater	50
Lampiran-4 Penilaian SIMPLE-OM (skenario 1)	52
Lampiran-5 Penilaian SIMPLE-OM (skenario 2)	53
Lampiran-5 Penilaian SIMPLE-OM (skenario 3)	55



DAFTAR SINGKATAN

LSA	Latent Semantic Analysis
SIMPLE-O	Sistem Penilaian Esai Otomatis
SIMPLE-OM	Sistem Penilaian Esai Otomatis Modifikasi
SVD	Singular Value Decomposition
SICBI	SQRT-IGFF-COSN-BNRY-IDFB
HTML	Hyper Text Markup Language
JAMA	Java Matrix
HTTPD	Hypertext Transfer Protocol Daemon



BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Web course adalah penggunaan internet untuk keperluan pendidikan, yang mana peserta didik dan pengajar sepenuhnya terpisah dan tidak diperlukan adanya tatap muka. Seluruh bahan ajar, diskusi, konsultasi, penugasan, latihan, ujian, dan kegiatan pembelajaran lainnya sepenuhnya disampaikan melalui internet (Haughey, 1998).

Namun demikian masih ada beberapa sektor dalam *web course* yang masih dalam perkembangan. Terutama pada pemberian ujian, bentuk ujian yang selama ini bisa diakomodir oleh internet masih terbatas pada bentuk soal pilihan ganda atau jawaban singkat. Bentuk ujian lain yang tidak kalah penting, yaitu bentuk ujian esai. Pada dasarnya ujian esai ini sebenarnya terkendala pada penilaiannya (*essay grading*). Pada kasus ini otomasi *essay grading* dengan bantuan komputer melalui internet sangatlah penting. Karena ketika peserta didik sangat banyak, maka penilaian secara manual oleh pengajar akan sangat sulit dilakukan dan memakan waktu yang sangat lama. Salah satu metode otomasi *essay grading* adalah *essay grading* metode LSA. *Latent Semantic Analysis* (LSA) merupakan representasi terhadap jumlah dan kebolehjadian kata untuk dibandingkan secara geometris (matrik). LSA merepresentasikan isi kata dalam matrik dua dimensi yang besar. Bagian pemrosesan penting dari LSA adalah komponen penganalisisan bernama SVD (*Singular Value Decomposition*) yang mengkompresi informasi yang berkaitan dalam jumlah besar ke dalam ruang yang lebih kecil tetapi mewakili arti sebenarnya.

Ada beberapa parameter yang mempengaruhi keefektifan suatu otomasi penilaian esai. Diantaranya adalah *text preprocessing*, *weighting* (pembobotan), *singular value dimensionality* dan *type of similarity measure* [2]. Parameter-parameter itu tentu saja bertujuan untuk meningkatkan kemampuan sistem dalam penilaian esai. Pada buku skripsi ini akan dibahas penerapan teknik pembobotan pada sistem *essay grading* metode *Latent Semantic Analysis* dengan teknik *Singular Value Decomposition*.

1.2. TUJUAN PENULISAN

Tujuan dari penulisan skripsi ini adalah pengimplementasian teknik pembobotan pada suatu sistem *essay grading* berbahasa Indonesia.

1.3. BATASAN MASALAH

Permasalahan difokuskan dan dibatasi pada penerapan skema pembobotan SICBI (SQRT-IGFF-COSN-BNRY-IDFB) pada sistem *essay grading* metode *Latent Semantic Analysis* dengan teknik *Singular Value Decomposition*.

1.4. SISTEMATIKA PENULISAN

Bab I membahas mengenai latar belakang, tujuan penulisan, batasan masalah dan sistematika penulisan untuk memberikan gambaran umum mengenai penulisan skripsi ini. Bab II menjelaskan landasan teori. Diawali penjelasan mengenai *essay grading* metode LSA, LSA, dan mekanisme perhitungan SVD. Selain itu dijelaskan pula program pendukung serta teori pembobotan. Bab III berisikan langkah-langkah konsep aplikasi yang akan dibuat. Pada bab ini juga akan diperlihatkan kapan pembobotan digunakan serta uraian mengenai skema pembobotan SICBI. Bab IV berisikan ujicoba aplikasi dan analisa analisa pengujian aplikasi. Pada bab ini akan diuraikan skenario pengujian sistem. Bab V merupakan kesimpulan dari penulisan skripsi ini.

BAB II

ESSAY GRADING METODE LSA DAN LATENT SEMANTIC ANALYSIS (LSA)

2.1. ESSAY GRADING METODE LSA

Ada beberapa metode *essay grading* yang saat ini tengah dikembangkan baik untuk kebutuhan riset ataupun komersial. Metode-metode tersebut menunjukkan tingkat korelasi yang cukup tinggi bila dibandingkan dengan pemeriksaan secara manual. Setiap metode memiliki teknik penilaian yang berbeda. Namun walaupun teknik penilaiannya berbeda tapi tujuan yang dicapai sama, yaitu membangun suatu sistem yang mampu memberikan penilaian terhadap jawaban esai secara otomatis seobjektif mungkin.

Akhir tahun 1980-an, sekelompok ilmuwan Bellcore mengembangkan metode statistikal dan berbasis jumlah untuk mengambil teks [3]. Tidak seperti teknik sederhana yang bergantung pada bobot kesamaan kata pada kalimat, metode mereka yang bernama *Latent Semantic Analysis (LSA)*, menciptakan representasi terhadap jumlah dan kebolehjadian kata untuk dibandingkan secara geometris (matrik).

Bagian pemrosesan penting dari LSA adalah komponen penganalisisan bernama SVD (*Singular Value Decomposition*) yang mengkompresi informasi berkaitan dalam jumlah besar ke dalam ruang yang lebih kecil. LSA merepresentasikan isi kata dalam matriks dua dimensi yang besar. Menggunakan teknik aljabar matriks, yaitu SVD tadi, hubungan baru antara kata dan dokumen ditentukan dan dimodifikasi untuk mewakili arti sebenarnya. Tiap analisis kata direpresentasikan dalam kolom sedangkan tiap baris mewakili kalimat, paragraf, dan sub divisi lainnya yang berkaitan.

Landauer melaporkan, LSA telah diujikan dengan lima skema penilaian, masing-masing dengan perlakuan berbeda dimana esai mahasiswa dibandingkan dengan esai referensi. Ini terutama berkaitan dengan vektor yang harus dikomputasi. Unjuk kerja penilaian LSA hampir sama handalnya

dengan penilaian manusia. Dari tes esai pada GMAT, kesepakatan antara penilaian manusia dan sistem LSA berkisar antara 85% sampai 91% [4].

Di Indonesia sendiri penilaian esai otomatis ini tengah dikembangkan terutama untuk esai berbahasa Indonesia.

2.2. SVD (SINGULAR VALUE DECOMPOSITION)

Telah diuraikan sebelumnya bahwa bagian pemrosesan dari LSA adalah SVD. Teknik *Singular Value Decomposition* (SVD) digunakan untuk melakukan estimasi struktur dalam penggunaan kata dalam dokumen-dokumen. SVD pada dasarnya merupakan teknik untuk melakukan estimasi *rank* dari matriks [3]. Jika diketahui matriks A dengan dimensi $m \times n$, dimana nilai $m \geq n$ dan $\text{rank}(A) = r$ maka *singular value decomposition* dari A , dinotasikan sebagai $\text{SVD}(A)$, didefinisikan melalui persamaan

$$A = U\Sigma V^T \quad \dots (2-1)$$

dimana

$$U^T U = V^T V = I_n \quad \dots (2-2)$$

dan memenuhi kondisi

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n) \quad \dots (2-3)$$

dimana

$$\sigma_i > 0 \text{ untuk } 1 \leq i \leq r$$

$$\sigma_j = 0 \text{ untuk } j \geq r+1$$

Kolom r pertama dari matriks U dan V mendefinisikan vektor eigen orthonormal yang bersesuaian dengan r nilai vektor eigen tidak-nol dari

matriks AA^T dan $A^T A$ berturut-turut. Kolom dari matriks U dan V berisi vektor, masing-masing disebut vektor singular kiri dan kanan.

Nilai singular dari A merupakan elemen diagonal dari matriks Σ , dimana nilai singular didapat dari akar pangkat dua dari nilai absolut dari sejumlah n nilai eigen dari AA^T [3].

2.3. LSA VIA SVD

Untuk melakukan LSA, harus dibuat sebuah matriks yang dibangun dari susunan kata kunci (*terms*) dan dokumen. Matriks ini didefinisikan sebagai matriks A . Elemen dari matriks kata kunci-dokumen ini didapat dari banyaknya kehadiran setiap kata kunci pada dokumen tertentu,

$$A = [a_{ij}] = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} \end{pmatrix} = A_{\bullet 1} \quad A_{\bullet 2} \quad \cdots \quad A_{\bullet j} = \begin{pmatrix} A_{1\bullet} \\ A_{2\bullet} \\ \vdots \\ A_{i\bullet} \end{pmatrix} \quad \dots (2-4)$$

Dimana a_{ij} bernilai frekuensi kehadiran kata kunci i pada dokumen j . Nilai frekuensi ini umumnya disebut *tf* (*term frequency*). $A_{\bullet j}$ merupakan matriks kolom yang elemennya menggambarkan kemunculan tiap kata kunci pada dokumen ke- j . $A_{i\bullet}$ menggambarkan frekuensi kemunculan kata kunci i pada setiap dokumen. Karena tidak setiap kata kunci akan muncul pada setiap dokumen, maka matriks A pada umumnya lowong (*sparse*), yakni kondisi dimana elemen bernilai 0 jauh lebih banyak.

Matriks A difaktorkan menjadi hasil perkalian dari 3 matriks triplet singular U , Σ dan V seperti diperlihatkan pada Gambar 2.1. Inilah yang disebut SVD. SVD menurunkan struktur laten semantik dari A melalui perkalian matriks ortogonal U , Σ dan V . Matriks-matriks ini merefleksikan hubungan asli antara dokumen dengan kata-kata kunci menjadi vektor-vektor yang bebas linear.

$$A = U \Sigma V^T$$

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} = \begin{bmatrix} \left[\begin{array}{c} u_1 \\ \vdots \\ u_r \end{array} \right] \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r \end{bmatrix} \cdot \begin{bmatrix} \left[\begin{array}{c} v_1 \\ \vdots \\ v_r \end{array} \right] \end{bmatrix}$$

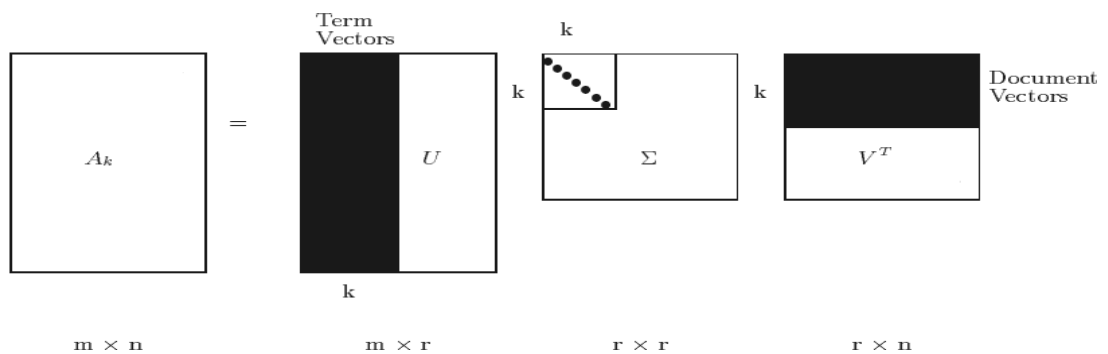
Gambar 2.1. Dekomposisi matriks A dengan SVD

Penggunaan faktor k dalam singular triplet merupakan langkah untuk membentuk A_k , yang merupakan aproksimasi matriks asli A dalam ruang k . Pada kasus ini, SVD dipandang sebagai sebuah teknik yang dipakai untuk menurunkan kumpulan variabel indeks, dimana setiap kata kunci dan dokumen dapat direpresentasikan sebagai sebuah vektor dalam ruang k .

Tabel 2-1. Interpretasi komponen SVD dalam LSA

A	= Matriks kata kunci – dokumen	m	= Banyaknya kata kunci
A_k	= Pendekatan <i>rank-k</i> terhadap A	n	= Banyaknya dokumen
U	= Vektor-vektor kata kunci	k	= Nilai faktor
Σ	= Nilai Singular	r	= <i>Rank</i> dari A
V	= Vektor-vektor dokumen		

Gambar 2.2 adalah representasi dari SVD dengan penggunaan faktor k . Bagian yang diarsir pada matriks U dan V dan diagonal Σ membentuk A_k . Untuk keterangan tiap simbolnya, dapat dilihat pada Tabel 2-1



Gambar 2.2. Dekomposisi SVD dengan faktor k [4]

Didalam metode LSA, operasi *truncated* SVD menghasilkan matriks A_k yang tidak sama dengan matriks A yang asli. Matriks A_k hanyalah sebuah pendekatan atau aproksimasi A pada faktor k . *Truncated* SVD mengambil sebagian besar struktur penting yang terdapat pada hubungan kata kunci dan dokumen. Dan, pada saat yang sama juga menghilangkan *noise* atau variabilitas penggunaan kata yang menjadi gangguan utama dalam proses *information retrieval* [3]. Selama nilai dari k jauh lebih kecil dari banyaknya kata kunci (m) maka perbedaan minor dalam terminologi dapat diabaikan. Kata-kata kunci yang terdapat dalam dokumen yang sama, akan berdekatan satu sama lain dalam ruang k walaupun kata-kata kunci itu tidak pernah hadir bersamaan lagi pada dokumen yang sama. Hal ini berarti beberapa dokumen yang tidak memiliki satupun kata kunci yang sama yang terdapat dalam *query*, maka dia tidak akan mendekati *query* tersebut dalam ruang- k .

2.4. RELEVANSI DOKUMEN DAN QUERY

Query dapat direpresentasikan sebagai vektor dalam ruang- k . Vektor inilah yang kemudian di bandingkan dengan vektor-vektor dokumen untuk selanjutnya dinilai mana yang paling mendekati. Sebuah *query* seperti halnya dokumen, merupakan kumpulan dari kata-kata. *Query* pengguna dapat representasikan sebagai

$$\bar{q} = q^T U_k \Sigma_k^{-1} \dots (2-5)$$

Mirip dengan vektor *query*, vektor dokumen direpresentasikan sebagai

$$\bar{d} = d^T U_k \Sigma_k^{-1} \dots (2-6)$$

Matriks q adalah matriks satu kolom yang elemennya berisi nilai kehadiran kata kunci dalam *query*. Sementara matriks d adalah matriks satu

kolom. Elemennya berisi nilai kehadiran kata kunci dalam dokumen. Matriks d sama dengan kolom matriks A .

\bar{q} adalah vektor *query* dan \bar{d} adalah vektor dokumen. Vektor *query* dapat dibandingkan atau dikorelasikan dengan semua vektor dokumen yang ada. Teknik korelasi yang umum digunakan adalah dengan mencari nilai kosinus sudut yang dibentuk antara vektor *query* dan vektor dokumen. Korelasi kosinus antara vektor *query* dan vektor dokumen diberikan oleh persamaan

$$\cos \alpha = \frac{\bar{q} \cdot \bar{d}}{\|\bar{q}\| \|\bar{d}\|} \quad \dots (2-7)$$

α adalah sudut diantara kedua vektor tersebut. Jika \bar{q} dan \bar{d} dinormalisasi, maka *magnitude* dari vektor tersebut adalah 1 dan persamaan diatas dapat disederhanakan menjadi $\cos \alpha = \bar{q} \cdot \bar{d}$. Jadi, nilai korelasi adalah perhitungan sudut berdasarkan kosinus antara \bar{q} dan \bar{d} . Jika dilakukan pengurutan dari dokumen yang paling dekat ke paling jauh relevansinya, maka dokumen yang paling dekat adalah dokumen yang memiliki sudut α dengan yang paling kecil.

2.5. PROGRAM PENDUKUNG

Otomasi *essay grading* dengan metode LSA yang telah dikembangkan adalah suatu sistem terpadu yang dibangun dengan bahasa *scripting* PHP dan HTML serta dukungan *database* MySQL. Selain tiga program pendukung di atas, LSA sendiri menggunakan modul JAMA yang diprogram ulang menjadi program PHP untuk menghitung matriks yang hasilnya dikirimkan ke *database* dan PHP kembali untuk ditampilkan. *Web server* yang digunakan adalah Apache. Selain itu untuk menjalankan program bisa dilakukan dengan dukungan web browser seperti Internet Explorer, Opera atau Mozilla Firefox. Pembangunan fitur pembobotan juga harus

dilakukan menggunakan bahasa *scripting* dan sistem *database* yang sama untuk menjalin integrasi yang sesuai.

2.5.1 PHP

PHP – yang merupakan singkatan rekursif “PHP: Hypertext Preprocessor” – bukan bahasa pemrograman. PHP adalah bahasa *scripting* open source yang ditulis menggunakan sintaks bahasa C, Java, dan Perl [6]. PHP membuat sebuah halaman web menjadi dinamis. Artinya halaman web menjadi lebih interaktif dan halaman yang ditampilkan dibuat saat *client* melakukan *request* halaman tersebut sehingga informasi yang diterima oleh *client* adalah selalu informasi yang terbaru.

Script PHP menyatu dengan file HTML (*HyperText Markup Language*), dieksekusi di komputer *server* dimana *script* tersebut dijalankan (*server side*), jadi semua informasi yang ingin ditampilkan di halaman *web* bisa dilihat dengan baik oleh semua jenis *browser client*.

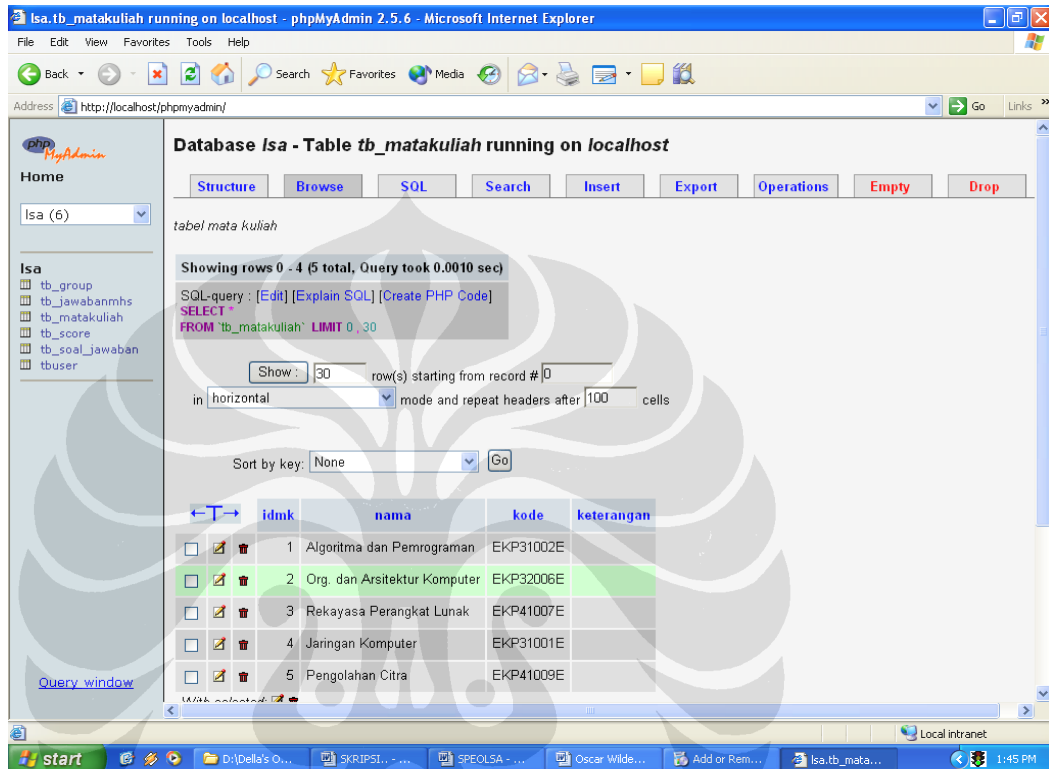
2.5.2 MySQL

MySQL merupakan salah satu program *database server* yang dikeluarkan oleh T.c.X. DataKonsultanAB, sebuah perusahaan IT Swedia dan banyak digunakan di internet saat ini. MySQL bersama PHP adalah pasangan bahasa *scripting* dan *database server* yang terbukti tangguh, memiliki jaminan keamanan yang tinggi dan cukup mudah dipelajari. Walau pada awalnya dibangun di atas *platform* Unix/Linux namun kini sudah dapat berjalan dengan baik pada sistem operasi Microsoft Windows.

Database sendiri merupakan bagian integral dalam pendataan di berbagai bidang. Pada sistem, semua data pengajar, mahasiswa, soal dan jawaban tersimpan dalam *database* sesuai dengan kategori-kategorinya. Tiap informasi dideskripsikan dalam tabel dengan *field-field* yang spesifik seperti Gambar 2.3.

Selain MySQL banyak *database server* lain yang beredar di pasaran, namun selain menyediakan dukungan *open source*, MySQL memiliki beberapa keunggulan lain. Pertama adalah kemampuannya menangani jutaan user dalam waktu yang bersamaan. Kelebihan ini tentu cocok untuk dimanfaatkan pada sebuah program penilaian esai yang ujiannya mungkin

diikuti ratusan bahkan ribuan mahasiswa. Kedua adalah kemampuannya menampung lebih dari 50.000.000 *record*. Selain itu MySQL sangat cepat mengeksekusi perintah dan memiliki sistem *user privilege* yang mudah dan efisien.



Gambar 2.3. Contoh tabel dalam MySQL

2.5.3 Apache

Seperti halnya PHP, Apache juga pertama kali didesain untuk sistem operasi Unix. Namun kini varian Apache telah dapat dijalankan di lingkungan Windows. Sebenarnya web server dibutuhkan untuk menjalankan PHP dan MySQL. Web server yang juga dikenal dengan istilah HTTPD (*Hypertext Transfer Protocol Daemon*) atau HTTP server adalah service yang bekerja untuk melayani request dari HTTP client (*web browser*) ke komputer server.

2.5.4 JAMA

Selain MATLAB, aplikasi matematis *web based* yang bisa digunakan untuk penghitungan SVD adalah JAMA. JAMA adalah singkatan dari *Java Matrix*. JAMA yang digunakan merupakan skrip php untuk perhitungan matriks kompleks. Class-class dari package JAMA akan sering digunakan dalam operasi matriks seperti perkalian matriks, transpose, dan inverse. Karena JAMA hanya merupakan script PHP –bukan merupakan aplikasi– maka kinerja server-pun tidak akan terlalu terbebani.

2.6. PEMBOBOTAN (*WEIGHTING*)

Sebuah metode pembobotan merupakan susunan dari tiga buah pembobotan: pembobotan lokal (*local weighting*), pembobotan global (*global weighting*) dan normalisasi (*normalization*) [1]. Teknik pembobotan yang tepat dapat meningkatkan performansi LSA. Pembobotan dikenakan pada tiap elemen matriks A . Pembobotan dirumuskan melalui persamaan :

$$a_{ij} = L(i, j) \times G(i) \times N(j) \quad \dots (2-8)$$

$L(i,j)$ merupakan bobot lokal untuk kata kunci i dalam dokumen j . $G(i)$ adalah bobot global untuk kata kunci i , dan $N(j)$ adalah faktor normalisasi dokumen j . Bobot lokal adalah fungsi dari berapa banyak setiap kata kunci muncul dalam suatu dokumen. Bobot global adalah fungsi dari berapa banyak setiap kunci muncul dalam semua dokumen. Faktor normalisasi digunakan untuk mengkompensasi perbedaan panjang dokumen-dokumen.

Vektor dokumen (matriks kolom A) dan vektor *query* dikenakan pembobotan dengan metode yang berbeda. Bobot lokal dihitung berhubungan dengan kata kunci pada dokumen atau *query*. Bobot global lebih didasarkan pada sejumlah dokumen yang ada tanpa memperhatikan apakah itu pembobotan pada dokumen atau *query*. Normalisasi vektor *query* sebenarnya tidak perlu karena tidak mempengaruhi urutan relevansi akhir terhadap dokumen.

2.6.1 Pembobotan Lokal

Pembobotan lokal akan bekerja dengan baik jika berdasarkan prinsip bahwa kata-kata kunci dengan frekuensi kemunculan yang banyak yang lebih berhubungan dengan dokumen [1]. Sejumlah pembobotan lokal yang umum digunakan diberikan dalam Tabel 2-2.

Pembobotan lokal yang paling sederhana adalah pembobotan biner (BNRY) dan frekuensi intra-dokumen (FREQ). Dari Tabel 2-2, f_{ij} adalah frekuensi kemunculan kata kunci i dalam dokumen j . Pembobotan ini biasanya digunakan untuk pembobotan pada *query*, dimana kata kunci hanya muncul satu-dua kali saja. Untuk pembobotan dokumen, metode ini umumnya bukan yang terbaik. Hal ini karena BNRY tidak membedakan antara kata kunci yang muncul beberapa kali dengan kata kunci yang muncul hanya sekali. Selain itu metode FREQ dinilai memberikan bobot terlalu besar untuk kata kunci yang muncul beberapa kali.

Metode logaritma digunakan untuk menyesuaikan frekuensi intra dokumen. Karena sebuah kata kunci yang muncul sepuluh kali dalam sebuah dokumen tidak berarti sepuluh kali lebih penting dibandingkan kata kunci yang muncul sekali dalam dokumen tersebut. Dua dari sejumlah metode pembobotan lokal dalam Tabel 2-2. bisa dikatakan mirip karena metode tersebut menggunakan logaritma. Dua metode itu adalah LOGA dan LOGN. Semua logaritma pada metode pembobotan berbasis 2. a_j adalah frekuensi rata-rata dari kemunculan kata kunci dalam dokumen j . Karena dalam LOGN terdapat normalisasi yakni $(1 + \log a_j)$, maka hasil pembobotan yang diberikan oleh LOGN akan selalu lebih kecil nilainya dibandingkan LOGA untuk kata kunci dan dokumen yang sama.

Pembobotan lokal lainnya, yang menjadi penengah antara metode biner dan frekuensi intra dokumen adalah metode normalisasi frekuensi diperlebar (*augmented normalized term frequency*) atau ATF1. Pada Tabel 2-2. x_j merupakan frekuensi maksimum dari kata kunci dalam dokumen j . ATF1 memberikan bobot pada sebuah kata yang muncul pada dokumen dan memberikan tambahan bobot bila kata tersebut muncul beberapa kali. Dengan

formula ini, $L(i,j)$ bervariasi hanya antara 0,5 sampai 1 untuk kata yang muncul dalam dokumen.

Tabel 2-2. Macam-macam pembobotan lokal

Formula	Nama Metode	Kependekan
$\begin{cases} 1 & \text{jika } f_{ij} > 0 \\ 0 & \text{jika } f_{ij} = 0 \end{cases}$	Biner	BNRY
f_{ij}	Frekuensi intra-dokumen	FREQ
$\begin{cases} 1 + \log f_{ij} & \text{jika } f_{ij} > 0 \\ 0 & \text{jika } f_{ij} = 0 \end{cases}$	Log	LOGA
$\begin{cases} \frac{1 + \log f_{ij}}{1 + \log a_j} & \text{jika } f_{ij} > 0 \\ 0 & \text{jika } f_{ij} = 0 \end{cases}$	Normalisasi log	LOGN
$\begin{cases} \sqrt{f_{ij} - 0,5} + 1 & \text{jika } f_{ij} > 0 \\ 0 & \text{jika } f_{ij} = 0 \end{cases}$	Akar pangkat dua	SQRT
$\begin{cases} 0,5 + 0,5 \left(\frac{f_{ij}}{a_j} \right) & \text{if } f_{ij} > 0 \\ 0 & \text{if } f_{ij} = 0 \end{cases}$	Normalisasi frekuensi diperlebar	ATF1
$\begin{cases} 0,9 + 0,1 \left(\frac{f_{ij}}{a_j} \right) & \text{jika } f_{ij} > 0 \\ 0 & \text{jika } f_{ij} = 0 \end{cases}$	Normalisasi frekuensi rata-rata diperlebar	ATFA
$\begin{cases} 0,2 + 0,8 \left(\frac{f_{ij}}{x_j} \right) & \text{jika } f_{ij} > 0 \\ 0 & \text{jika } f_{ij} = 0 \end{cases}$	ATF1 dengan perubahan koefisien	ATFC

[1]

2.6.2 Pembobotan Global

Pembobotan global ditujukan untuk memberikan sebuah "nilai beda" kepada setiap kata kunci. Pembobotan global yang berdasarkan ide bahwa semakin kecil nilai frekuensi kemunculan kata dalam seluruh koleksi dokumen, maka makin berbedalah kata tersebut [1].

Tabel 2-3. Macam-macam pembobotan global

Formula	Nama Metode	Kependekan
$\log\left(\frac{N}{n_i}\right)$	Invers frekuensi dokumen	IDFB
$\log\left(\frac{N-n_i}{n_i}\right)$	Invers probabilistik	IDFP
$1 + \frac{\sum_{j=1}^N \frac{f_{ij}}{F_i} \log\left(\frac{f_{ij}}{F_i}\right)}{\log N}$	Entropi	ENPY
$\frac{F_i}{n_i}$	Frekuensi global IDF	IGFF
$\sqrt{\frac{F_i}{n_i} - 0,9}$	Akar pangkat dua global IDF	IGFS
1	Tidak ada bobot global	NONE

[1]

Sebuah pembobotan global yang umum digunakan adalah *inverted document frequency* atau IDF. Dalam Tabel 2-3 diberikan dua variasi yakni IDFB dan IDFP. N adalah jumlah dokumen dalam koleksi dan n_i merupakan jumlah dokumen dimana kata kunci i muncul didalamnya. IDFB adalah logaritma dari invers dari probabilitas kata kunci i muncul dalam dokumen acak. IDFP adalah logaritma dari invers dari probabilitas ketidak-hadiran kata kunci i dalam dokumen acak. IDFB dan IDFP adalah sama dalam artian keduanya memberikan bobot yang lebih besar untuk kata yang tampil pada

beberapa dokumen saja dan memberikan bobot yang lebih kecil untuk kata yang muncul pada banyak dokumen dalam koleksi. IDFP memberikan bobot negatif untuk kata yang muncul pada lebih dari separuh jumlah seluruh dokumen. Sementara pada IDFB, nilai terendah pembobotan adalah 1.

Pada metode entropi (ENPY), F_i merupakan frekuensi kemunculan kata kunci i di seluruh koleksi dokumen. Jika sebuah kata kunci muncul sekali pada setiap dokumen, maka kata tersebut diberikan bobot bernilai nol. Jika sebuah kata kunci muncul sekali pada satu dokumen, maka kata tersebut diberi bobot satu. Kombinasi dan variasi lain dari frekuensi kemunculan akan menghasilkan bobot yang nilainya antara nol dan satu. Entropi adalah teknik pembobotan yang sangat berguna karena ia memberikan bobot yang lebih besar untuk kata yang frekuensi kemunculannya kecil pada sejumlah kecil dokumen.

Dalam Tabel 2-3. juga disebutkan pembobotan frekuensi global IDF (IGFF). Jika sebuah kata kunci muncul sekali pada setiap dokumen atau sekali pada satu dokumen, maka kata tersebut diberikan bobot sebesar satu, yang merupakan bobot terkecil. Sebuah kata yang muncul beberapa kali pada sejumlah dokumen akan mendapat bobot yang besar. Pembobotan ini bekerja dengan baik jika dikombinasikan dengan pembobotan global yang berbeda pada vektor *query*.

2.6.3 Normalisasi

Bagian ketiga dari sebuah pembobotan adalah faktor normalisasi atau $N(j)$, yang mana digunakan untuk mengkompensasi perbedaan panjang dokumen-dokumen dalam koleksi. Bagian ini berguna untuk menormalkan vektor dokumen sehingga dokumen-dokumen tersebut independen terhadap panjangnya. Pada Tabel 2-4. diperlihatkan dua buah metode normalisasi.

Normalisasi yang paling umum digunakan dalam model ruang vektor adalah normalisasi kosinus (COSN). Normalisasi ini memiliki faktor pembagi magnitude dari dokumen yang dibobotkan, sehingga hal ini menyebabkan magnitude dari vektor dokumen selalu bernilai satu. Dalam metode COSN, dokumen yang lebih panjang diberikan bobot lebih kecil untuk kata kunci,

sehingga dokumen yang lebih pendek akan lebih panjang dalam proses perolehan informasi.

Tabel 2-4. Macam-macam Normalisasi

Formula	Nama Metode	Kependekan
$\frac{1}{\sqrt{\sum_{i=0}^m G_i L_{ij}^2}}$	Normalisasi kosinus	COSN
$\frac{1}{(1 - slope) pivot + slope \times l_j}$	Normalisasi pivot	PUQN
1	Tidak ada normalisasi	NONE

[1]

Metode *pivoted unique normalization* (PUQN) mencoba untuk mengatasi masalah dalam penanganan dokumen-dokumen yang pendek. Dalam Tabel 2.4, l_j adalah banyaknya kata kunci yang berbeda dalam dokumen j . Nilai *slope* dapat diset sebesar 0,2 dan *pivot* adalah rata-rata banyaknya kata kunci yang berbeda per dokumen dalam seluruh koleksi. Prinsip dasar dari normalisasi pivot adalah untuk mengatasi perbedaan panjang dokumen diantara dokumen yang memiliki probabilitas relevan dan dokumen yang memiliki probabilitas akan diperoleh atau di-*retrieve*. Dengan faktor normalisasi ini, kurva relevansi dan kurva *retrieval* digambarkan berdasarkan panjang dokumen. Titik dimana kedua kurva ini bersinggungan atau memotong disebut pivot. Dokumen pada sebelah kiri dari pivot umumnya memiliki probabilitas yang lebih besar untuk di-*retrieve* daripada tingkat relevansinya. Dan, dokumen yang ada di sebelah kanan pivot memiliki probabilitas lebih relevan daripada untuk di-*retrieve*. Melalui penggeseran pivot ini, faktor normalisasi dapat diubah-ubah sedemikian rupa untuk mendapatkan hasil kombinasi yang lebih baik antara probabilitas relevansi dan *retrieval*.

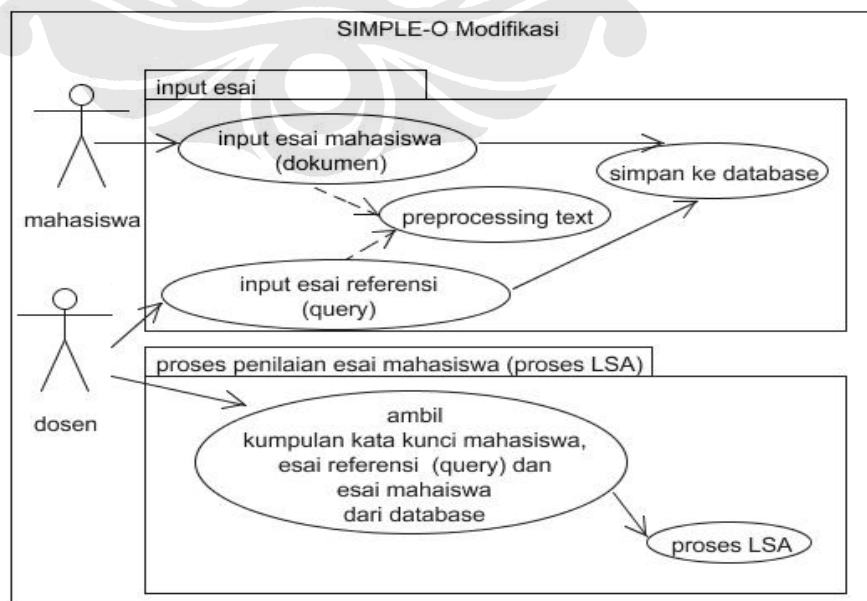
BAB III

PERANCANGAN APLIKASI & MEKANISME PEMBOBOTAN SICBI

3.1. KONSEP APLIKASI

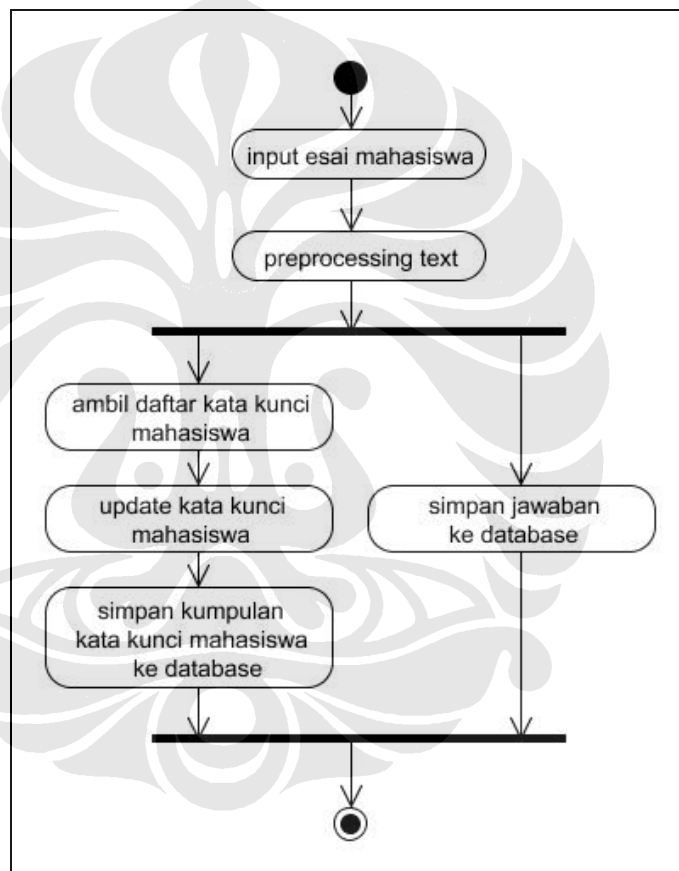
SIMPLE-O adalah aplikasi penilaian esai otomatis berbasis web yang dikembangkan di Indonesia, tepatnya di Departemen Teknik Elektro Universitas Indonesia. Dengan berbasiskan web maka mahasiswa dan dosen atau siapapun yang berkaitan dengan aplikasi ini bisa mengakses aplikasi ini dimanapun dan kapanpun. Aplikasi berjalan melalui beberapa tahapan yaitu proses pemasukan data, pemrosesan data, penyimpanan data dan penampilan hasil bagi pengguna. Pemasukan dan pemrosesan data menggunakan skrip PHP yang didalamnya telah dimasukan *script* JAMA untuk digunakan pada perhitungan matrik. Dan untuk penyimpanannya menggunakan database MySQL.

Untuk mengimplementasikan pembobotan seperti yang dijelaskan pada subbab 2.7 maka pada SIMPLE-O dilakukan perubahan pada bagian proses memasukan jawaban mahasiswa dan proses penilaiannya. Use Case Diagram dari SIMPLE-O yang dimodifikasi (SIMPLE-OM) ditunjukkan pada Gambar 3.1.

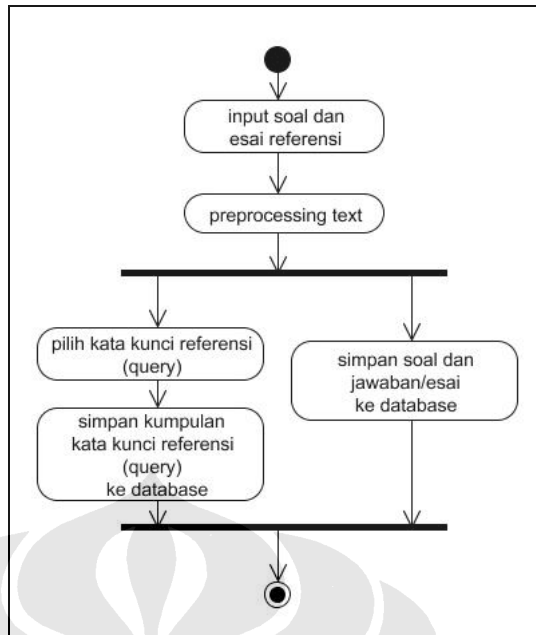


Gambar 3.1. Use Case Diagram SIMPLE-O yang dimodifikasi (SIMPLE-OM)

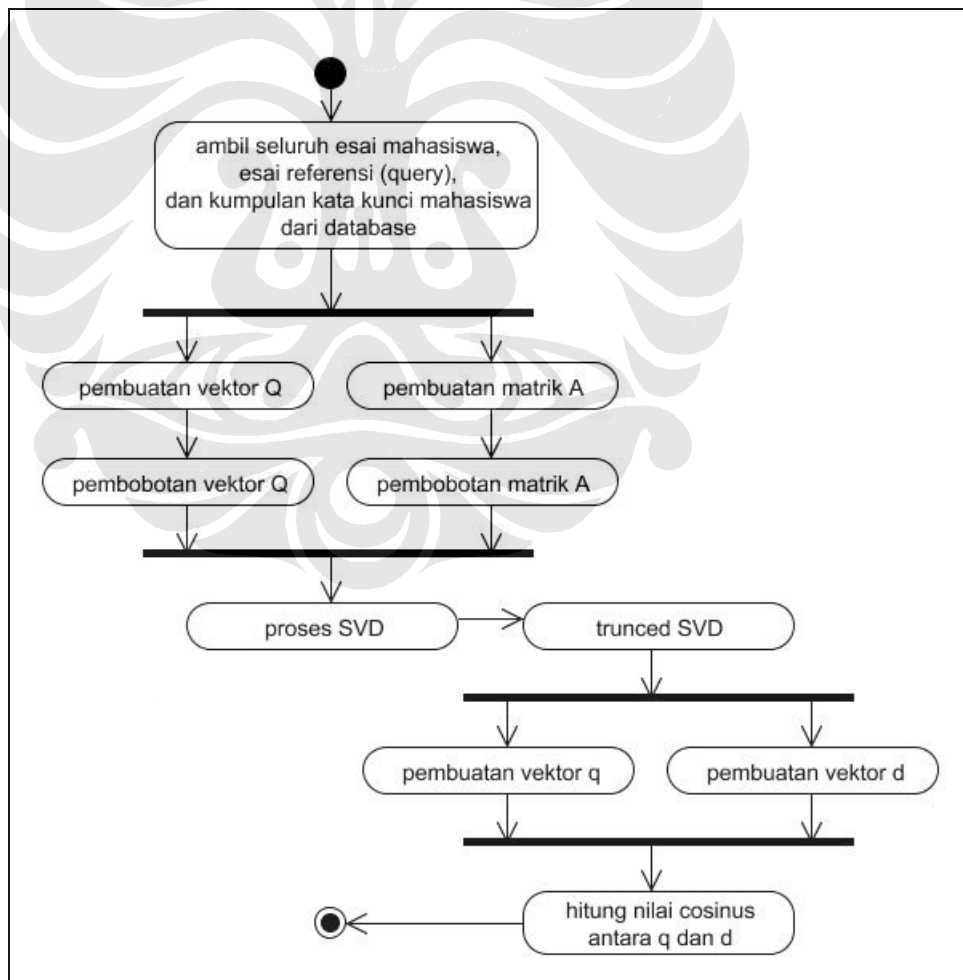
Seperti telah diuraikan pada subbab 2.1, setidaknya ada lima skema penilaian yang bisa dilakukan. Pada dasarnya skema yang digunakan yaitu menjadikan kumpulan esai mahasiswa sebagai kumpulan dokumen dan esai referensi sebagai query, seperti yang ditunjukkan pada Gambar 3.1. Kumpulan esai mahasiswa tersebut nantinya akan menjadi *semantic space* atau disebut juga Matrik A sedangkan esai referensi akan menjadi vector Q. Untuk lebih jelasnya, Gambar 3.2, Gambar 3.3, dan Gambar 3.4 menunjukkan activity diagram SIMPLE-OM dilihat dari sisi mahasiswa dan dosen.



Gambar 3.2. Activity Diagram Input Jawaban Mahasiswa



Gambar 3.3. Activity Diagram Input Esai Referensi (untuk dosen)



Gambar 3.4. Activity Diagram Proses Penilaian Esai Mahasiswa (proses LSA, ada di sisi dosen)

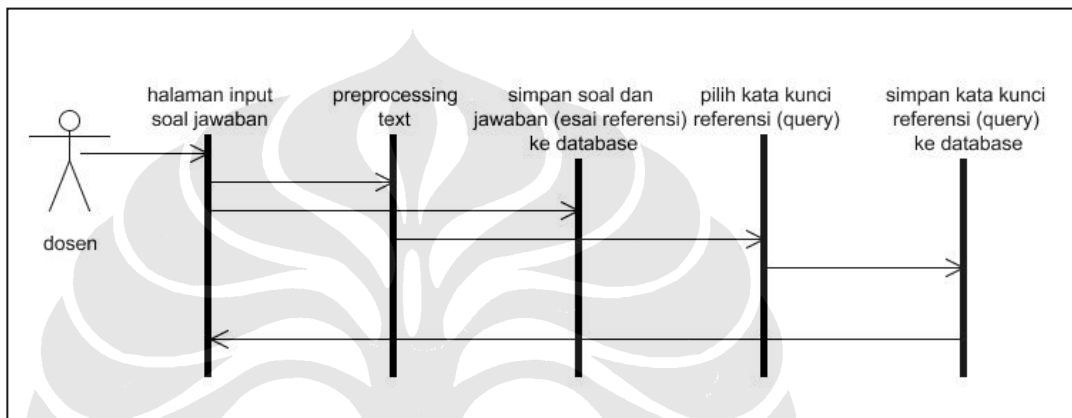
Pada skripsi ini, matrik A tersebut terdiri dari kumpulan esai mahasiswa. Selain itu, semua data yang diolah sebelumnya mengalami *preprocessing text*. *Preprocessing text* dilakukan untuk tujuan penyeragaman dan kemudahan pembacaan serta proses LSA selanjutnya. *Preprocessing* esai mahasiswa ataupun esai referensi meliputi:

- Penghilangan *white space* dan karakter null seperti spasi kosong panjang, dan tab. Dari sini diharapkan antar kata dalam hanya dipisahkan oleh 1 buah karakter spasi saja
- Penghilangan sejumlah karakter angka yang membentuk numerik sebuah bilangan (karakter numerik yang dicampur dengan alfabet tidak dihilangkan. Contohnya : "mobile8", "GSM900")
- Penghilangan karakter-karakter diluar alfabet terbaca seperti titik, koma, tanda kurung, #, \$, %, &, *, !, ? dan sejenisnya
- Penghilangan kata-kata yang sangat umum digunakan seperti "dan", "kemudian", "atau" dan sejenisnya

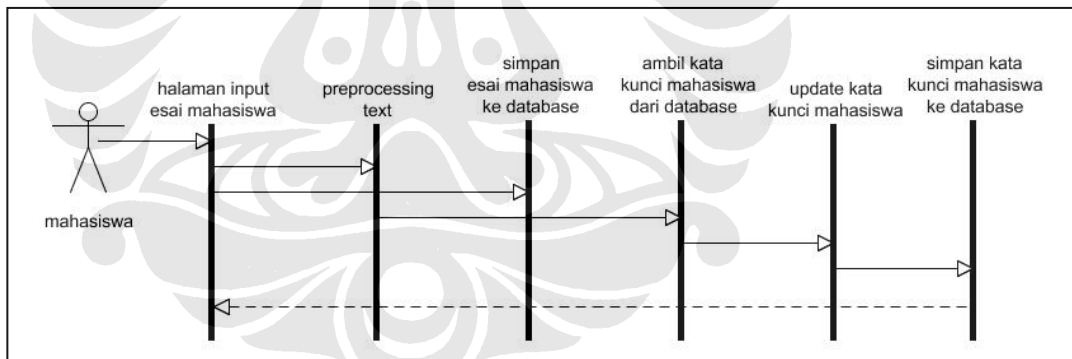
Setelah melalui tahap *preprocessing*, dokumen akan disimpan kedalam database sesuai *field*-nya. Langkah berikutnya adalah proses *update kata_kunci_mahasiswa*. Dalam LSA dikenal istilah *term* yakni sekumpulan kata kunci yang terdapat dalam dokumen-dokumen. Dalam hal ini, dokumen berarti esai mahasiswa dan *terms* adalah *kata_kunci_mahasiswa*. Kombinasi kemunculan sejumlah *kata_kunci_mahasiswa* terhadap esai-esai mahasiswa akan membangun matriks A yang diperlukan dalam proses LSA nantinya. Kumpulan *kata_kunci_mahasiswa* dipilih secara otomatis oleh program. Berikut ketentuan pemilihan dan pemrosesan kata kunci:

- Kata kunci bukan merupakan kata-kata yang sangat umum digunakan seperti "dan", "kemudian", "atau" dan sejenisnya. Untuk itu diperlukan juga daftar kata umum yang akan dihilangkan dari esai mahasiswa.
- Didalam daftar kata kunci tidak boleh ada dua kata kunci yang sama
- Daftar kata kunciurut sesuai abjad

Proses *updating* kata kunci seperti yang ditunjukkan pada Gambar 3.2 dilakukan dengan cara mengambil daftar kata kunci yang ada di database, kemudian dibandingkan dengan kata kunci dari esai mahasiswa yang baru masuk. Setelah melalui ketiga aturan diatas, daftar kata kunci baru diperoleh dan disimpan (*update*) ke database. Proses input esai mahasiswa berakhir disini dengan menampilkan pesan sukses. *Sequence diagram* yang menunjukkan peralihan dari satu proses ke proses lain dalam sisi mahasiswa dan dosen diperlihatkan pada Gambar 3.5 dan Gambar 3.6 .



Gambar 3.5. Sequence Diagram Input Soal Jawaban Sisi Dosen



Gambar 3.6. Sequence Diagram Input Jawaban Sisi Mahasiswa

3.2. PROSES PEMBOBOTAN

3.2.1 Pembobotan SICBI

Skema pembobotan yang akan diterapkan pada aplikasi SIMPLE-OM adalah skema pembobotan SICBI. SICBI adalah akronim dari SQRT-IGFF-COSN-BNRY-IDFB. Pada skema ini, matrik A (matrik yang dibangun dari kumpulan esai mahasiswa) akan mendapatkan pembobotan Lokal SQRT, pembobotan Global IGFF, serta Normalisasi menggunakan COSN.

Sedangkan untuk vektor Q, akan dilakukan pembobotan Lokal BNRV dan pembobotan Global IDFB. Skema pembobotan SICBI dianggap sangat cocok digunakan pada sistem penilaian esai otomatis metode LSA.

Jawaban/esai mahasiswa yang biasanya panjang dan tidak padat akan menyebabkan frekuensi kemunculan suatu kata kunci besar. Sedangkan frekuensi kemunculan yang besar belum tentu memiliki bobot nilai yang besar (nilai kepentingan yang tinggi), begitupun sebaliknya. Esai yang memiliki frekuensi kemunculan kata kunci yang besar juga belum tentu sama nilai bobotnya dengan esai yang memiliki frekuensi kemunculan kata kunci yang kecil. Karena itu untuk pembobotan lokal esai mahasiswa tidak sesuai jika menggunakan FREQ ataupun BNRV. Pembobotan lokal SQRT diterapkan pada skema pembobotan SICBI karena dianggap sesuai dengan dengan karakteristik esai mahasiswa. Berbeda dengan esai referensi (*query*), yang biasanya singkat dan padat yang terdiri dari kata kunci referensi saja. Sehingga pembobotan BNRV cocok digunakan untuk esai referensi.

Untuk pembobotan global, pada esai mahasiswa diterapkan Pembobotan Global IGFF. Pembobotan ini merupakan perbandingan antara banyaknya kemunculan suatu kata kunci dalam semua esai mahasiswa dengan banyaknya esai mahasiswa yang memiliki kata kunci itu. Semakin besar nilai perbandingannya maka bisa dikatakan bahwa tingkat kepentingan kata kunci tersebut tinggi dan sebaliknya. Jika sebuah kata kunci muncul sekali pada setiap esai mahasiswa, maka kata tersebut diberikan bobot sebesar satu, yang merupakan bobot terkecil. Sebuah kata yang muncul beberapa kali pada sejumlah dokumen akan mendapat bobot yang besar. Sedangkan pada esai referensi diterapkan pembobotan global yang umum digunakan yaitu Pembobotan Global IDFB. Karena esai referensi hanya terdiri dari satu esai saja, maka IDFB akan memberikan bobot yang lebih besar untuk kata yang ada. Hal ini penting karena pada esai referensi setiap kata kunci yang ada memiliki tingkat kepentingan yang tinggi. Pada IDFB, nilai terendah pembobotan adalah 1.

Normalisasi yang hanya diterapkan pada esai mahasiswa saja. Normalisasi sebenarnya berguna untuk mengkompensasi perbedaan panjang

esai-esai mahasiswa yang ada. Dalam metode COSN, dokumen yang lebih panjang diberikan bobot lebih kecil untuk kata kunci, sehingga dokumen yang lebih pendek akan lebih panjang dalam proses perolehan informasi. Hal ini sesuai dengan karakteristik esai-esai mahasiswa yang tentunya memiliki panjang esai yang berbeda tiap mahasiswanya.

3.2.2 Mekanisme Pembobotan

Pembobotan dilakukan setelah matriks A dan vektor Q selesai dibangun seperti ditunjukkan pada Gambar 3.4. Pembobotan bertujuan untuk meningkatkan performa LSA seperti yang telah dijelaskan dalam subbab 2.6. Pada dasarnya pembobotan dibagi menjadi tiga yakni pembobotan lokal, pembobotan global dan normalisasi. Dengan mengacu pada Tabel 2-2., Tabel 2-3., dan Tabel 2-4. maka terdapat banyak skema pembobotan yang mungkin dilakukan. Karena pembobotan dibagi menjadi tiga, maka setiap skema akan melakukan tiga kali perhitungan. Untuk vektor *query*, tidak perlu dilakukan normalisasi. Hal ini karena vektor *query* hanya terdiri dari satu buah esai. Hasil yang diharapkan adalah matriks A dan vektor Q yang sudah dibobotkan. Masing-masing disimpan dalam bentuk variabel *array*, *Array_A* dan *Array_Q*.

Untuk mengetahui pengaruh pembobotan pada sistem, maka sistem pun dicoba dengan tidak menyertakan tahapan pembobotan atau dengan kata lain menerapkan skema pembobotan NONE. Skema pembobotan NONE berarti matrik A dan vektor Q tidak mengalami perubahan. Hal ini bisa dilakukan dengan hanya menerapkan pembobotan lokal FREQ pada matrik A dan vektor Q.

Prosedur pembobotan yang ditunjukkan Gambar 3.7 terdiri dari dua skema pembobotan, yaitu pembobotan SICBI dan NONE. Pemilihan pembobotan dibuat statis, sehingga untuk merubah skema pembobotan dapat dilakukan dengan cara menset variabel Pembobotan = 1.

```

Proc Pilih_Pembobotan()

switch (Pembobotan) {

case Pembobotan=1 : //PEMBOBOTAN SICBI
    Bobot_lokal_A = bobot_lokal_SQRT();
    Bobot_global_A = bobot_global_IGFF();
    Bobot_normalisasi_A = bobot_normalisasi_COSN();

    Bobot_lokal_Q = bobot_lokal_BNRY();
    Bobot_global_Q = bobot_global_IDFB();

case Pembobotan=2: //PEMBOBOTAN NONE
    Bobot_lokal_A = bobot_lokal_FREQ();
    Bobot_global_A = bobot_global_NONE();
    Bobot_normalisasi_A = bobot_normalisasi_NONE();

    Bobot_lokal_Q = bobot_lokal_FREQ();
    Bobot_global_Q = bobot_global_NONE();
}

Eproc

```

Gambar 3.7. Algoritma Proses Pemilihan Pembobotan

Karena pada dasarnya pembobotan dibagi menjadi tiga yakni pembobotan lokal, pembobotan global dan normalisasi, maka setiap skema akan melakukan tiga kali perhitungan. Untuk vektor *query*, tidak perlu dilakukan normalisasi. Seperti yang terlihat dalam cuplikan kode diatas, perhitungan pembobotan dilakukan oleh fungsi tersendiri. Misalnya perhitungan pembobotan lokal SQRT yang rumusnya ada pada Tabel 2-2, dikerjakan oleh fungsi `bobot_lokal_SQRT()` seperti ditunjukkan pada Gambar 3.8. Fungsi ini memiliki nilai *return* atau output berupa matriks (dalam bentuk *array*) dengan dimensi $i \times j$.

Keluaran dari setiap fungsi merupakan matriks, dimana dimensinya sesuai dengan definisi setiap pembobotan dalam Tabel 2-2, Tabel 2-3, dan Tabel 2-4. Jadi semua fungsi pembobotan lokal akan memiliki output berupa matriks berdimensi $i \times j$, fungsi pembobotan global akan mengeluarkan output berupa matriks berdimensi $i \times 1$ dan fungsi normalisasi memberikan keluaran matriks $1 \times j$.

```

function SQRT() {
for($i=0;$i<$nKata_kunci_mahasiswa;$i++) {
    for($j=0;$j<$nMahasiswa,$Array_Bobot;$j++) {
        if($Array_Bobot[$i][$j]>0) {
            $DummyAkar = sqrt($Array_Bobot[$i][$j]-0.5);
            $SQRT[$i][$j] = $DummyAkar-1;
        }
        else {
            $SQRT[$i][$j] = 0;
        }
    }
}
return $SQRT;
} Efunc

//-----

function IGGF() {
$Dummy_Dok = ' '.implode(" ", $Dokumen). ' ';
$ni=0;
for($i=0;$i<$nKata_kunci_mahasiswa;$i++) {
    $Fi[$i]=substr_count($Dummy_Dok, ' '.$kata_kunci_mahasiswa[$i]. ' ');
    for($j=0;$j<$nMahasiswa;$j++) {
        if($Array_Bobot[$i][$j]>0) {
            $ni++;
        }
    }
    $IGGF[$i]=$Fi[$i]/$ni;
    $ni=0;
}
return $IGGF;
} Efunc

//-----

function COSN() {
for($j=0;$j<$nMahasiswa;$j++) {
    for($i=0;$i<$nKata_kunci_mahasiswa;$i++) {
        $DummyS[$i]=pow($Array_Bobot[$i][$j],2);
    }
    $DummyAkar = sqrt(array_sum($DummyS));
    $COSN[$j] = 1/$DummyAkar;
}
return $COSN;
} Efunc

//-----

function IDFB() {
$ni=0;
for($i=0;$i<$nKata_kunci_mahasiswa;$i++) {
    for($j=0;$j<$nMahasiswa;$j++) {
        if($Array_A[$i][$j]>0) {
            $ni++; }
    }
    $IDFB[$i]=log($nMahasiswa /$ni,2);
    $ni=0;
}
return $IDFB;
} Efunc

//-----

function BNRV() {
for($j=0;$j<$nKata_kunci_mahasiswa;$j++) {
    for($i=0;$i<$nMahasiswa;$i++) {
        if($Array_A[$j][$i]>0) {
            $BNRV[$j][$i] = 1;
        }
        else {
            $BNRV[$j][$i]=0;
        }
    }
}
return $BNRV;
} Efunc

```

Gambar 3.8. Algoritma Fungsi Pembobotan SICBI

Proses berlanjut dengan mengalikan semua elemen pembobotan untuk membentuk skema pembobotan. Proses ini dilakukan dengan menggunakan prosedur pembobotan yang ditunjukkan pada Gambar 3.9.

```
Proc Pembobotan()
for(i=0; i<jumlah_kata_kunci_mahasiswa ; i++) {
  for(j=0; j<jumlah_mahasiswa; j++) {
    Array_A[i][j] =Bobot_Lokal_A[i][j]*Bobot_Global_A[i]*Bobot_Normalisasi_A[j];
    Array_Q[i]   =Bobot_Lokal_Q[i]*Bobot_Global_Q[i];
  }
}
Eproc
```

Gambar 3.9. Algoritma Pembobotan

Proses *looping* dilakukan untuk melakukan perkalian tiap elemen matriks sesuai persamaan (2-8). Disini, *looping* dalam bertugas menelusuri seluruh esai mahasiswa sementara *looping* luar menelusuri kata kunci mahasiswa. Dengan kata lain, proses perkalian matriks dilakukan dengan pola menyelesaikan sebuah baris dulu baru kemudian menelusuri semua kolom. Sampai kolom terakhir, barulah pindah baris. Hasil yang diharapkan adalah matriks A dan vektor Q yang sudah dibobotkan. Masing-masing disimpan dalam bentuk variabel *array* Array_A dan Array_Q.

3.2.3 Operasi Svd, Truncated Svd , Pembentukan q_k dan d_k , serta Relevansi q_k dan d_k

Operasi SVD dijalankan setelah matriks A dan vektor Q selesai dibobotkan. Kalkulasi dilakukan secara *object oriented programming*. Proses ini akan menghasilkan U, Σ dan V.

Setelah U, Σ dan V diperoleh, dilakukan truncated SVD. Kali ini triplet SVD yakni U, Σ dan V akan mengalami pengecilan dimensi baris dan kolom, jika ditentukan faktor $k=2$ maka dimensi matriknya seperti ditunjukkan pada Tabel 3-1.

Tabel 3-1. Dimensi matriks SVD

Hasil SVD Awal	Dimensi	Truncated SVD	Dimensi
U	$i \times j$	U _k	$i \times 2$
Σ	$j \times j$	Σ_k	2×2
V	$j \times j$	V _k	$2 \times j$

Matriks-matriks hasil truncated yakni U_k, Σ_k dan V_k bertujuan untuk membentuk ruang vektor sesuai faktor-*k* disini, faktor *k* adalah 2 sehingga vektor dokumen dan vektor *query* berdimensi 2. Hasil ini bisa dilihat melalui persamaan (2-4) dan (2-5). Truncated SVD dilakukan dengan bantuan *library* JAMA, dari *class* Matrix. Gambar 3.10. merupakan cuplikan dari program yang digunakan.

```

Proc TruncatedSVD()
k = 2;
Matriks_Sk = Matriks_S.getMatrix(1, k, 1, k);
Matriks_Uk = Matriks_U.getMatrix(1, Baris_U, 1, k);
Matriks_Vk = Matriks_V.getMatrix(1, k, 1, Kolom_V);
Eproc

```

Gambar 3.10. Algoritma Truncated SVD

Variabel Baris_U dan Kolom_V pada Gambar 3.10. berturut-turut adalah baris matriks U dan V hasil SVD awal. Yang perlu diperhatikan adalah dalam *object* matrix, indeks awal adalah 1. Proses *truncated* intinya dilakukan oleh tiga baris terakhir dalam prosedur ini. Ketiga baris itu sama-sama menjalankan method *getMatrix* dan hasilnya berupa *object*, disimpan ke Matriks_Sk, Matriks_Uk dan Matriks_Vk. *getMatrix* sendiri merupakan *method* yang bertujuan untuk meng-*capture* atau memecah matriks dari *object* sesuai indeks baris dan kolom (awal dan akhir) dengan pola sintaks: *getMatrix(baris_awal, baris_akhir, kolom_awal, kolom_akhir)*

Untuk membentuk vektor *query* urutan langkahnya adalah sebagai berikut. Pertama, Matriks Σ_k di-*inverse* setelah itu matriks q di-*transpose*, kemudian lakukan perkalian matriks sesuai persamaan (2-4). Dalam blok kode diatas, perkalian matriks dilakukan dengan method *times*.

Untuk membentuk vektor d , urutannya mirip seperti pembentukan vektor *query* hanya saja, yang di-*transpose* adalah matriks d . Matriks d sama dengan kolom matriks A . Oleh karena itu persamaan (2-5) dapat dikembangkan menjadi $D = A^T U_k \Sigma_k^{-1}$. Dengan D merupakan matriks yang kolomnya merupakan vektor 2 dimensi setiap dokumen dalam koleksi.

Relevansi antara esai mahasiswa dan esai referensi dilihat dari hasil perhitungan kosinus sesuai persamaan (2-6). Namun sebelumnya perlu dilakukan penghitungan Frobenius Norm untuk vektor *query* dan vektor dokumen agar didapatkan panjang vektor *query* dan vektor dokumen. Gambar 3.11 merupakan algoritma perhitungan Frobenius Norm untuk vektor *query* (kata_kunci esai referensi) dan vektor dokumen (esai mahasiswa).

```

Proc Hitung_Frobenius_vektor_Q()
Frobenius_Q = Vektor_Q.normF(); Eproc

Proc Hitung_Frobenius_vektor_D()
for(i=0; i<Baris_D; i++) {
    Frobenius_D[i] = Vektor_D[i].normF(); } Eproc

```

Gambar 3.11. Algoritma/Rumus Perhitungan Frobenius

Dari blok kode diatas didapatkan besarnya $|d|$ dan $|q|$. Norm dari d disimpan dalam bentuk array di variabel Frobenius_D dengan indeks mengikuti indeks dokumen. Kemudian perhitungan nilai kosinus antara vektor *query* dan dokumen dapat dilakukan dengan algoritma yang ditunjukkan pada Gambar 3.12.

```

Proc Hitung_Kosinus_vektor_D()
Baris=Jumlah_Dokumen
for(i=0; i<Baris; i++) {

    // Menghitung Q . D

    Vektor_Dt = $Vektor_D.transpose();
    QD[i] = Vektor_Q.times(Vektor_Dt);

    // Menghitung Q . D / |Q| |D|

    Cosine[i] = QD[i]/(Frobenius_Q*Frobenius_D[i]);
} Eproc

```

Gambar 3.12. Algoritma Proses Kalkulasi Kosinus

BAB IV

UJI COBA DAN ANALISA PENGUJIAN APLIKASI

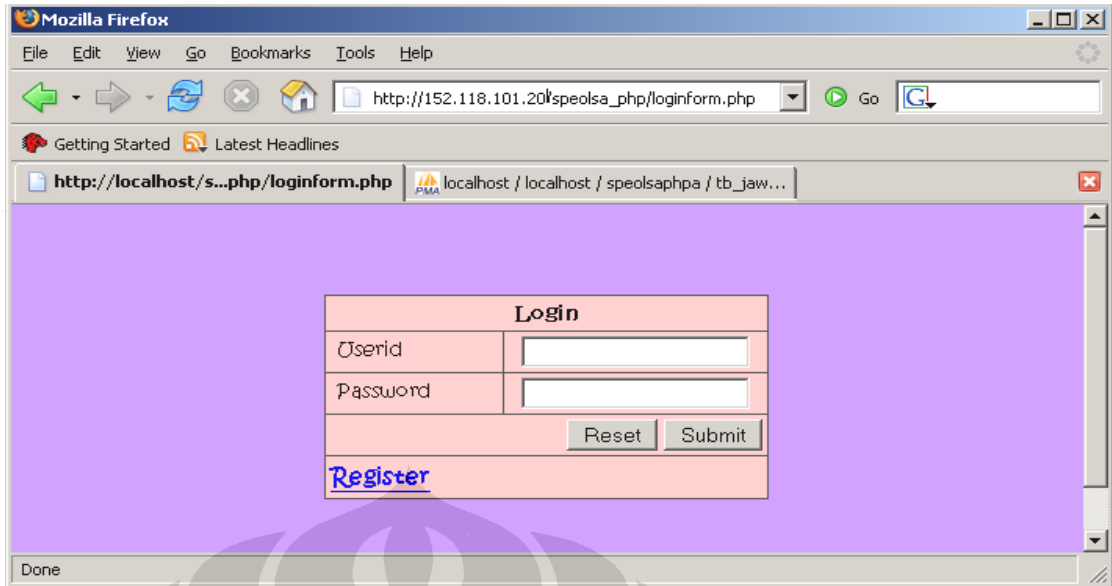
4.1. UJI COBA APLIKASI

Aplikasi ini merupakan aplikasi yang berbasis web, sehingga dalam pelaksanaannya memerlukan *web server*. Dalam *web server* itu tentunya diharuskan memiliki *PHP engine* dan *MySQL* yang nantinya akan digunakan sebagai *database system*. Ujicoba dilakukan pada sebuah *web server* yang telah ter-*install* *APACHE 2.0.49*, *PHP 4.3.9*, dan *MySQL 4.1.20* dengan spesifikasi *hardware* dan *software* sebagai berikut:

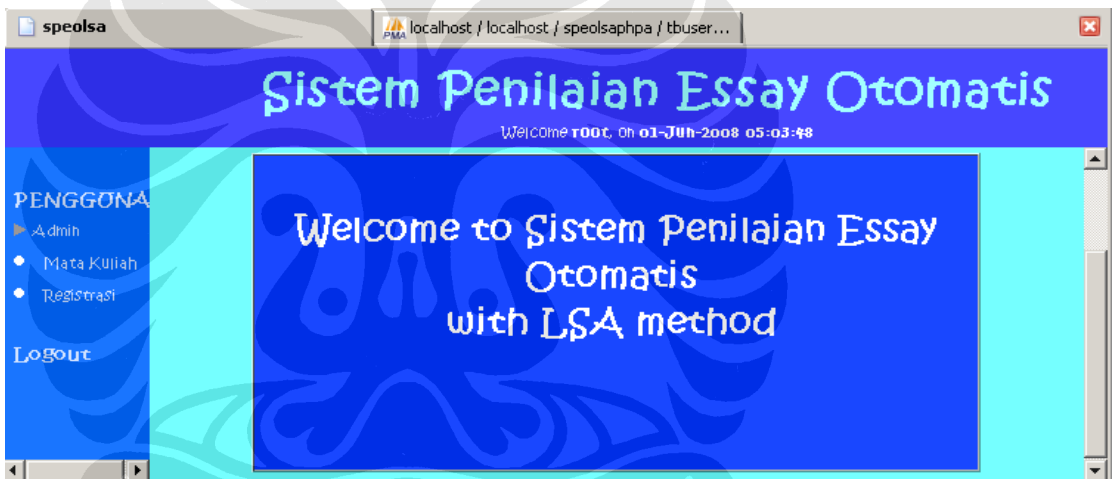
Prosesor	: Intel Pentium D 2.0 GHz
RAM	: 1 Gb
Mainboard	: ECS 945GZ/CT-M

Untuk ujicoba keadaan *server* dan *client* berada dalam jaringan yang berbeda. Dalam hal ini *client* adalah komputer yang digunakan oleh mahasiswa. Setelah *server* dijalankan, *client* dapat langsung mengakses aplikasi dengan bantuan *browser* apapun sistem operasinya.

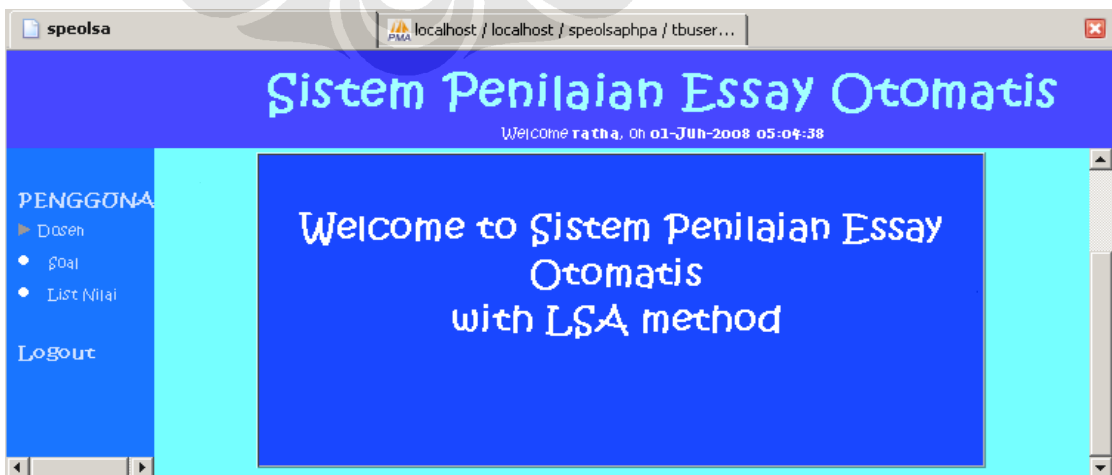
Pengguna dapat memakai layanan *SIMPLE-OM* dengan mengetikkan http://152.118.101.20/speolsa_php.php. Halaman muka langsung tampil seperti yang terlihat dalam Gambar 4.1. Pada halaman ini pengguna diharuskan memasukan *username* dan *password*. Sama halnya dengan *SIMPLE-O*, *SIMPLE-OM* ini juga memiliki tiga jenis pengguna yaitu admin, dosen, dan mahasiswa. Perbedaan antara keduanya hanyalah pada proses penginputan esai mahasiswa dan pada proses penilaian esai-esai mahasiswa saja. Pada Gambar 4.2, Gambar 4.3, dan Gambar 4.4 secara berturut-turut adalah halaman muka bagi pengguna admin, dosen, dan mahasiswa.



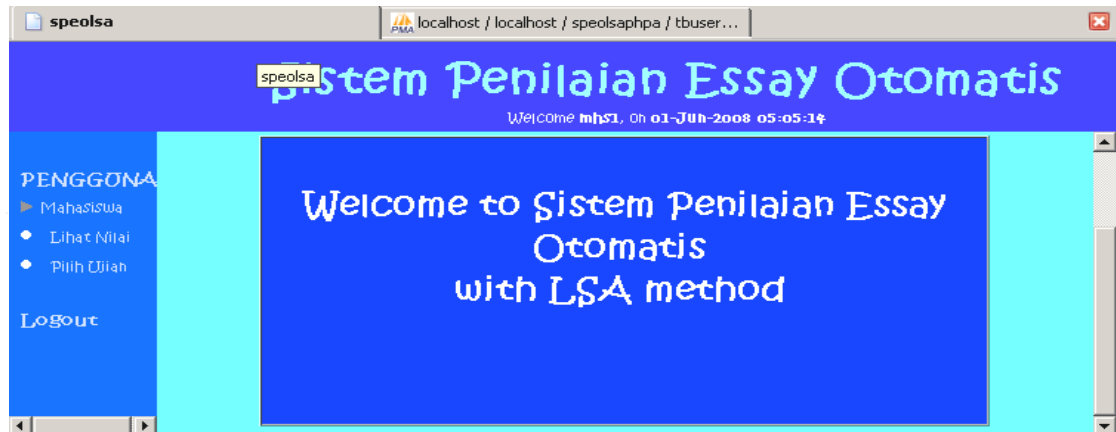
Gambar 4.1. Halaman Muka SIMPLE-O Modifikasi



Gambar 4.2. Halaman Muka Pengguna Admin



Gambar 4.3. Halaman Muka Pengguna Dosen



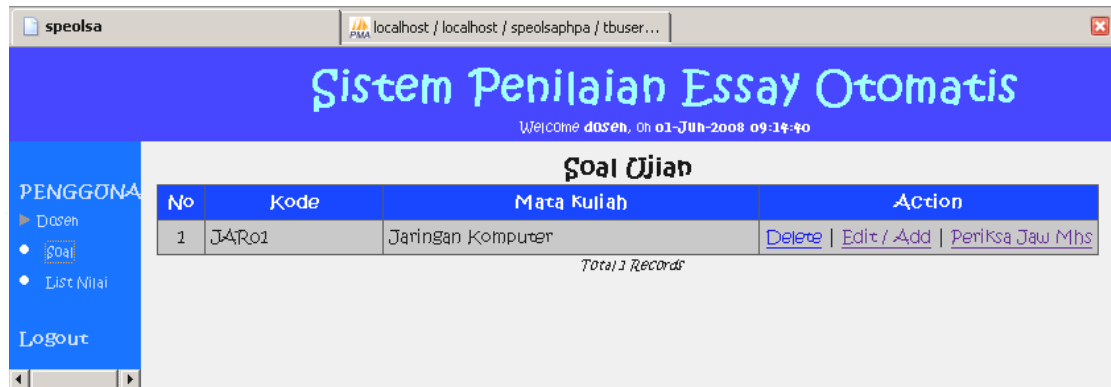
Gambar 4.4. Halaman Muka Pengguna Mahasiswa

4.1.1 Proses Input Esai Mahasiswa

Pada SIMPLE-O esai mahasiswa yang dimasukan (*submit*) pada saat ujian akan langsung tahapan SVD. Sedangkan pada SIMPLE-OM esai mahasiswa yang dimasukan akan langsung disimpan dalam database. Karena itu pada proses ini waktu yang dibutuhkan seorang mahasiswa untuk memasukan (*submit*) esainya tidaklah lama. Pada ujicoba yang dilakukan dengan menggunakan sebanyak 10 soal esai, waktu yang dibutuhkan bagi seorang mahasiswa untuk memasukan esainya adalah kurang dari satu detik.

4.1.2 Proses Penilaian Ujian

Proses penilaian esai mahasiswa dapat dilakukan ketika semua mahasiswa telah mengikuti ujian, dengan kata lain semua esai mahasiswa telah tersimpan didalam database. Penilaian esai dapat dilakukan oleh dosen yang bersangkutan dengan meng-klik link "Periksa Jaw Mhs" seperti ditunjukan pada Gambar 4.5.



Gambar 4.5. Periksa Esai Mahasiswa

Segara setelah dosen meng-klik link "Periksa Jaw Mhs" maka sistem akan mulai melakukan proses penilaian esai mahasiswa. Seperti telah diuraikan pada BAB III. Waktu rata-rata yang dibutuhkan oleh sistem untuk proses ini tergantung pada banyaknya mahasiswa yang mengikuti ujian dan juga pada banyaknya soal yang diujikan serta banyaknya kata_kunci_mahasiswa yang digunakan sistem. Untuk lebih jelasnya, bahasan mengenai waktu pemrosesan ini akan diuraikan pada sub subbab 4.2.4

4.2. ANALISIS PENGUJIAN APLIKASI

Untuk mengetahui pengaruh implementasi pembobotan SICBI pada aplikasi SIMPLE-OM maka dilakukanlah beberapa skenario pengujian. Pengujian dilakukan dengan membandingkan nilai mahasiswa yang diperoleh dengan menggunakan sistem tanpa pembobotan ($\cos \alpha$), sistem dengan pembobotan SICBI ($\cos \alpha$), dan *human rater*. Pengujian ini bertujuan untuk mengetahui unjuk kerja pembobotan SICBI terhadap kemampuan sistem (SIMPLE-OM) untuk menilai jawaban esai jika dibandingkan dengan *human rater*. Penilaian unjuk kerja akan didasarkan pada besarnya rata-rata selisih antara penilaian esai SIMPLE-OM dengan penilaian esai *human rater*. Selain itu, kecepatan proses sistem menunjukkan keefektifan sistem. Skenario pengujian yang dilakukan diantaranya:

1. Pengujian dilakukan pada kelas kecil (10 mahasiswa)
2. Pengujian dilakukan pada kelas sedang (15 mahasiswa)
3. Pengujian dilakukan pada kelas besar (20 mahasiswa)

Dari skenario-skenario pengujian diatas dapat diketahui pula pengaruh jumlah mahasiswa terhadap unjuk kerja sistem. Hal ini bisa diketahui dari perbandingan unjuk kerja sistem dari skenario 1, 2, dan 3.

4.2.1 Skenario Pengujian 1

Pada skenario pengujian 1 ini, jumlah soal yang diujikan sebanyak 10 soal dengan materi yang diujikan adalah dasar jaringan komputer. Jumlah mahasiswa yang diikutsertakan dalam skenario 1 sebanyak 10 mahasiswa. Pada Tabel 4-1. dapat dilihat rata-rata hasil penilaian SIMPLE-OM terhadap esai mahasiswa. Nilai yang tercantum pada Tabel 4-1 kolom (a), kolom (b), dan kolom (c) adalah rata-rata nilai 10 mahasiswa yang mengikuti ujian. Sedangkan untuk nilai detailnya ada pada Lampiran-3, Lampiran-4, dan Lampiran-5.

Tabel 4-1. Data Pengujian 1

Soal No	Tanpa Pembobotan (a)	Pembobotan SICBI (b)	<i>Human raters</i> (c)	Selisih (a) dan (c)	Selisih (b) dan (c)
1	76.34	51.53	49.00	27.34	2.53
2	62.25	61.16	79.00	16.75	17.84
3	65.28	59.96	62.00	3.28	2.04
4	84.92	53.40	60.00	24.92	6.60
5	73.78	68.80	16.00	57.78	52.80
6	81.92	50.07	47.00	34.92	3.07
7	64.92	51.92	23.00	41.92	28.92
8	37.56	50.62	64.00	26.44	13.38
9	89.07	65.71	75.00	14.07	9.29
10	65.80	57.63	61.00	4.80	3.37
Rata-rata Selisih				25,22	13,98

Pada Tabel 4-1. Selisih penilaian sistem tanpa pembobotan dengan *human rater* berkisar antara 3,78 – 57,78. Sedangkan untuk sistem Pembobotan SICBI berkisar antara 2,04 – 52,80. Untuk kedua sistem (sistem tanpa pembobotan dan sistem pembobotan SICBI), selisih terkecil didapat dari penilaian esai untuk soal nomor 3 dan selisih terbesar untuk soal nomor 5.

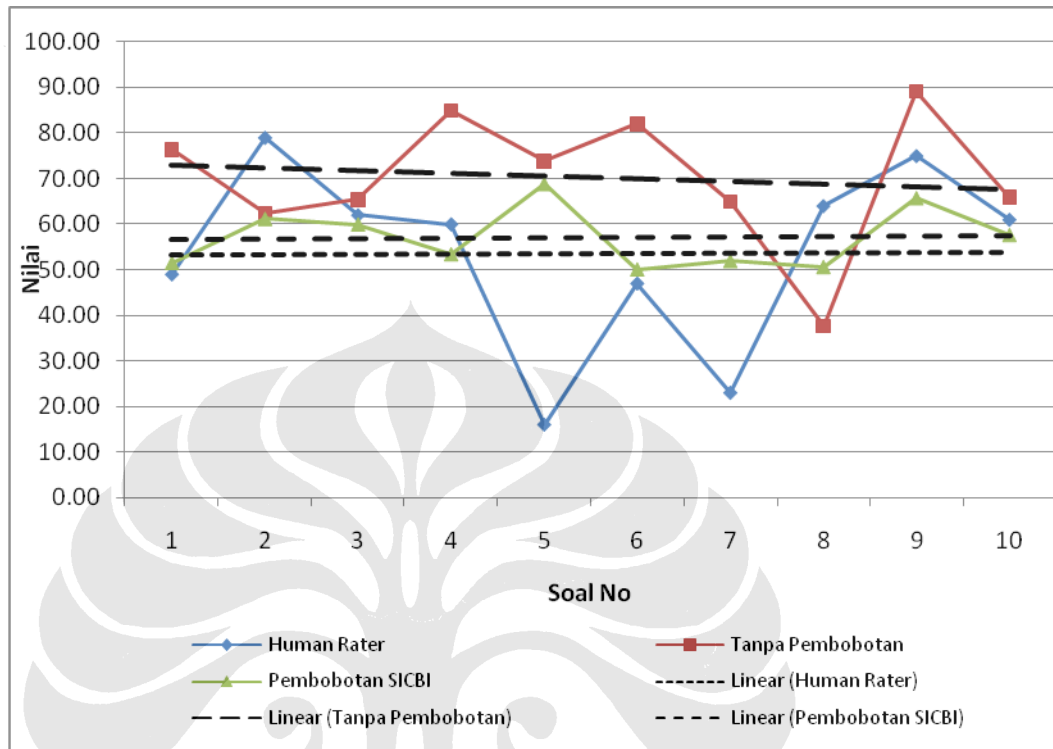
Pada soal nomor 5, rata-rata penilaian esai mahasiswa oleh *human rater* relatif kecil, yaitu 16. Hal ini dikarenakan pada terdapat beberapa mahasiswa yang mendapatkan nilai 0. Sedangkan penilaian yang dilakukan oleh sistem mahasiswa tersebut tidak mendapatkan nilai nol. Hal ini dikarenakan penilaian sistem didasarkan pada kata kunci. Selama dalam esai tersebut terdapat kata kunci, walaupun jawabannya salah maka esai tersebut akan mendapatkan nilai. Karena itulah pada soal nomor 5 selisih penilaian sistem dengan *human rater* menjadi relatif besar.

Selain itu, pada Tabel 4-1 juga ditemukan data dimana pembobotan SICBI sangat berperan dalam peningkatan kualitas penilaian esai, yaitu pada soal nomor 1 dan soal nomor 6. Pada kedua soal ini sistem pembobotan SICBI memberikan selisih yang jauh lebih kecil dibanding dengan sistem tanpa pembobotan. Hal ini berarti pembobotan SICBI sangat efektif terhadap esai nomor 1 dan 6. Jika memperhatikan esai mahasiswa pada soal nomor 1 dan 6, maka terdapat kata kunci yang frekuensi kemunculannya banyak namun sebenarnya kata tersebut memiliki tingkat kepentingan yang rendah. Sistem tanpa pembobotan tidak akan memperhatikan hal itu, dengan kata lain untuk suatu esai jika frekuensi besar maka tingkat kepentingan tinggi. Berbeda dengan sistem pembobotan SICBI yang memberikan tingkat kepentingan suatu kata tidak hanya berdasarkan frekuensi kemunculannya dalam suatu saja.

Berdasarkan data pada Tabel 4-1, maka dapat dibandingkan selisih antara penilaian yang dilakukan sistem dan penilaian yang dilakukan oleh manusia (*human rater*). Jika diperhatikan, selisih antara sistem pembobotan SICBI dengan *human rater* lebih kecil dibandingkan selisih antara sistem tanpa pembobotan dengan *human rater*. Dengan demikian, jika dilihat dari nilai selisihnya bisa dikatakan bahwa pembobotan SICBI memberikan hasil yang lebih baik, dalam artian penilaian yang dilakukan mendekati *human rater*.

Pada Gambar 4.6. dapat dilihat dengan lebih jelas bahwa *trendline* pembobotan SICBI lebih mendekati *trendline human rater* jika dibandingkan dengan *trendline* sistem tanpa pembobotan. Pada skenario pengujian 1 ini

didapatkan bahwa rata-rata selisih dan *trendline* sistem pembobotan SICBI lebih mendekati *human rater* dibandingkan dengan sistem tanpa pembobotan.



Gambar 4.6. Grafik Skenario Pengujian 1

4.2.2 Skenario Pengujian 2

Jumlah mahasiswa yang diikutsertakan dalam skenario 2 sebanyak 15 mahasiswa. Soal dan jumlah soal yang diujikan pada skenario ini sama seperti pada skenario 1. Pada Tabel 4-2, dapat dilihat rata-rata hasil penilaian SIMPLE-OM terhadap esai mahasiswa. Nilai yang tercantum pada Tabel 4-2, kolom (a), kolom (b), dan kolom (c) adalah rata-rata nilai 10 mahasiswa yang mengikuti ujian. Sedangkan untuk nilai detailnya ada pada Lampiran-3, Lampiran-4, dan Lampiran-5.

Pada skenario 2, selisih penilaian sistem tanpa pembobotan dengan *human rater* berkisar antara 2,64 – 52,54. Sedangkan untuk sistem Pembobotan SICBI berkisar antara 5,67 – 28,62. Pada skenario ini, kisaran selisih penilaian sistem dengan *human rater* mengalami penurunan. Bahkan untuk sistem pembobotan SICBI, penurunannya relatif signifikan.

Sama halnya dengan skenario 1, selisih yang paling besar terjadi pada soal nomor 5. Perlu diketahui bahwa mahasiswa yang ikut serta dalam pengujian skenario 2 ini adalah mahasiswa yang mengikuti skenario 1 ditambah 5 mahasiswa lagi. Karena itu, sama halnya dengan skenario 1, pada skenario 2 juga terdapat beberapa mahasiswa yang mendapatkan nilai nol dari *human rater*. Sementara itu sistem tidak memberikan nilai nol. Hal ini menyebabkan rata-rata nilai mahasiswa untuk soal nomor 5 relatif kecil. Akibatnya selisih penilaian esai antara sistem dan *human rater* menjadi besar.

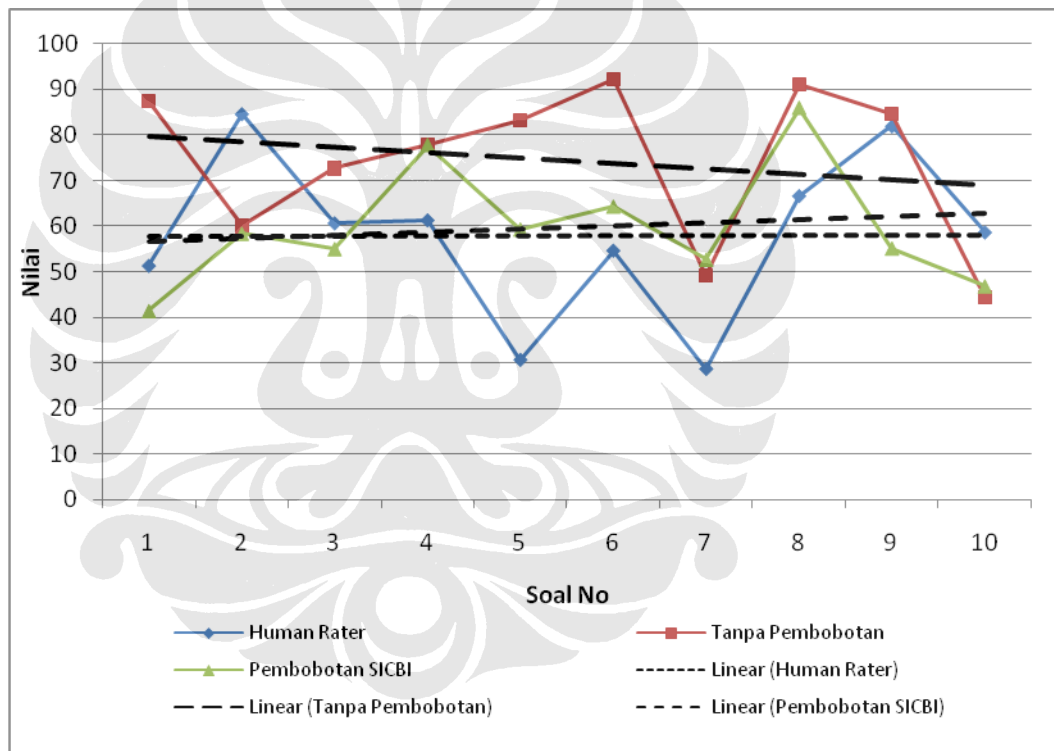
Walaupun begitu, sistem pembobotan SICBI pada skenario ini bisa dikatakan kurang efektif jika dilihat dari besarnya selisih yang terjadi pada esai soal nomor 2, 7, dan 9. Jika dibandingkan dengan sistem tanpa pembobotan, sistem pembobotan SICBI untuk ketiga nomor itu tidaklah efektif. Dengan kata lain pembobotan SICBI untuk ketiga nomor tersebut tidak memberikan hasil penilaian yang lebih baik dibanding dengan sistem tanpa pembobotan. Berbeda dengan soal nomor 1 dan nomor 6. Sistem pembobotan SICBI untuk kasus ini sangatlah efektif, seperti halnya terjadi pada skenario pengujian 1.

Tabel 4-2. Data Pengujian 2

Soal No	Tanpa Pembobotan (a)	Pembobotan SICBI (b)	<i>Human raters</i> (c)	Selisih (a) dan (c)	Selisih (b) dan (c)
1	87.46	41.46	51.33	36.12	9.87
2	60.08	58.38	84.67	24.59	26.29
3	72.74	55.00	60.67	12.07	5.67
4	77.88	77.77	61.33	16.55	16.44
5	83.21	59.29	30.67	52.54	28.62
6	92.23	64.28	54.67	37.56	9.62
7	49.16	52.69	28.67	20.49	24.02
8	91.24	85.84	66.67	24.57	19.17
9	84.64	55.12	82.00	2.64	26.88
10	44.48	46.86	58.67	14.19	11.80
Rata-rata Selisih				24.13	17.84

Berdasarkan pengamatan data pada Tabel 4-2. Secara keseluruhan sistem pembobotan SICBI memberikan penilaian yang lebih mendekati *human rater* dibandingkan dengan sistem tanpa pembobotan. Hal ini dilihat dari rata-rata selisih sistem pembobotan SICBI yang lebih kecil daripada sistem tanpa pembobotan.

Pada Gambar 4.7. dapat dilihat dengan lebih jelas bahwa *trendline* pembobotan SICBI lebih mendekati *trendline human rater* jika dibandingkan dengan *trendline* sistem tanpa pembobotan. Dengan demikian bisa dikatakan bahwa sistem pembobotan SICBI memberikan penilaian yang lebih mendekati penilaian *human rater* daripada sistem tanpa pembobotan.



Gambar 4.7. Grafik Skenario Pengujian 2

4.2.3 Skenario Pengujian 3

Jumlah mahasiswa yang diikutsertakan dalam skenario 3 sebanyak 20 mahasiswa. Soal dan jumlah soal yang diujikan pada skenario ini sama seperti pada skenario 1 dan skenario 2. Pada

Tabel 4-3 dapat dilihat rata-rata hasil penilaian SIMPLE-OM terhadap esai mahasiswa. Nilai yang tercantum pada Tabel 4-3. kolom (a),

kolom (b), dan kolom (c) adalah rata-rata nilai 10 mahasiswa yang mengikuti ujian. Sedangkan untuk nilai detailnya ada pada Lampiran-3, Lampiran-4, dan Lampiran-5.

Pada skenario 3, selisih penilaian sistem tanpa pembobotan dengan *human rater* berkisar antara 1,94 – 62,70. Sedangkan untuk sistem Pembobotan SICBI berkisar antara 0,18 – 21,61. Dari 10 soal yang diujikan hanya terdapat dua soal dimana sistem pembobotan SICBI memberikan hasil yang kurang memuaskan, yaitu pada nomor 8 dan 9. Pada kedua nomor tersebut selisih penilaian esai sistem pembobotan SICBI lebih besar daripada sistem tanpa pembobotan.

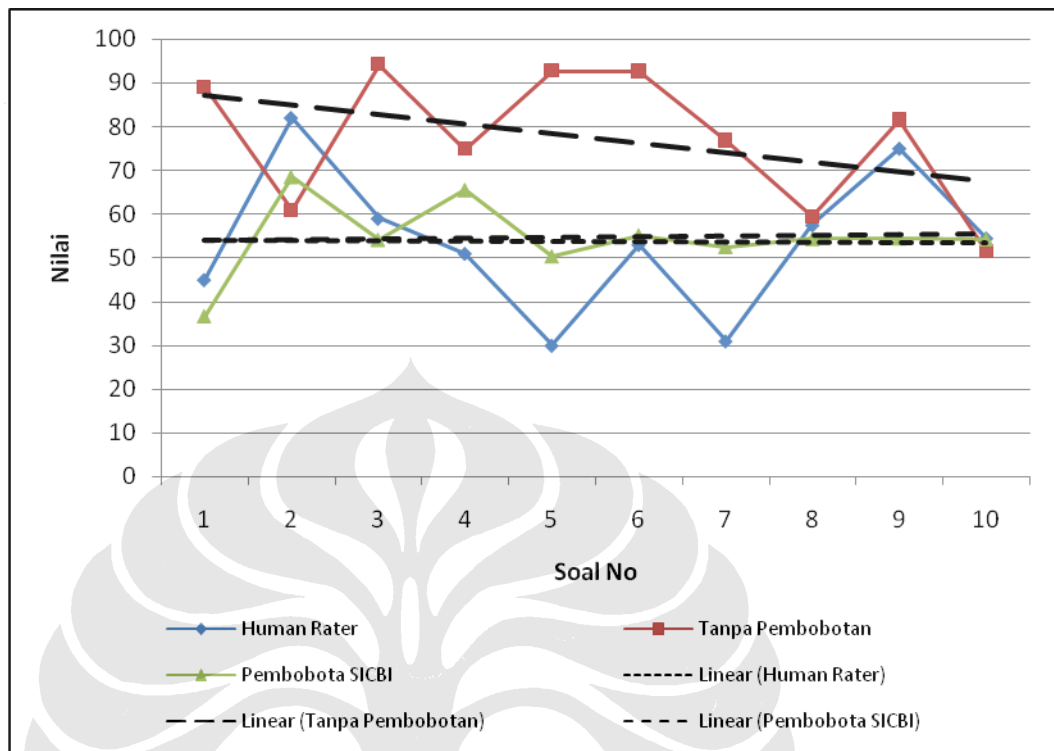
Tabel 4-3. Data Pengujian 3

Soal No	Tanpa Pembobotan (a)	Pembobotan SICBI (b)	<i>Human raters</i> (c)	Selisih (a) dan (c)	Selisih (b) dan (c)
1	89.01	36.68	45.00	44.01	8.32
2	60.97	68.60	82.00	21.03	13.40
3	94.12	54.23	59.00	35.12	4.77
4	74.87	65.61	51.00	23.87	14.61
5	92.70	50.42	30.00	62.70	20.42
6	92.61	55.25	53.00	39.61	2.25
7	76.75	52.61	31.00	45.75	21.61
8	59.44	54.45	57.50	1.94	3.05
9	81.53	54.57	75.00	6.53	20.43
10	51.63	54.32	54.50	2.87	0.18
Rata-rata Selisih				28.34	10.90

Walaupun demikian pada skenario 3 ini, hasil yang diperoleh sistem pembobotan SICBI jauh lebih baik jika dibandingkan dengan dua skenario sebelumnya. Hal ini dilihat dari rata-rata selisih penilaian yang kecil yaitu 10,90. Selain itu selisih tertinggi yang dicapai-pun relatif kecil yaitu 21,61. Jauh lebih kecil dibandingkan dengan skenario 1 (52,80) dan skenario 2 (28,62).

Pada Gambar 4.8. dapat dilihat bahwa *trendline* pembobotan SICBI lebih mendekati *trendline human rater* jika dibandingkan dengan *trendline* sistem tanpa pembobotan. Dengan demikian bisa dikatakan bahwa sistem

pembobotan SICBI memberikan penilaian yang lebih mendekati penilaian *human rater* daripada sistem tanpa pembobotan.



Gambar 4.8. Grafik Skenario Pengujian 3

4.2.4 Kecepatan Proses Penilaian Ujian

Banyaknya kata kunci mahasiswa dan jumlah mahasiswa merupakan dua variabel yang menentukan besarnya ukuran matriks A. Semakin banyak kata kunci, semakin banyak baris matriks A. Semakin banyak jumlah mahasiswa yang mengikuti ujian, semakin banyak kolom matriks A. Dan, keduanya saling berhubungan. Jumlah baris dan kolom ini akhirnya akan sangat mempengaruhi kecepatan proses karena elemen matriks yang akan melalui operasi matematis akan semakin banyak.

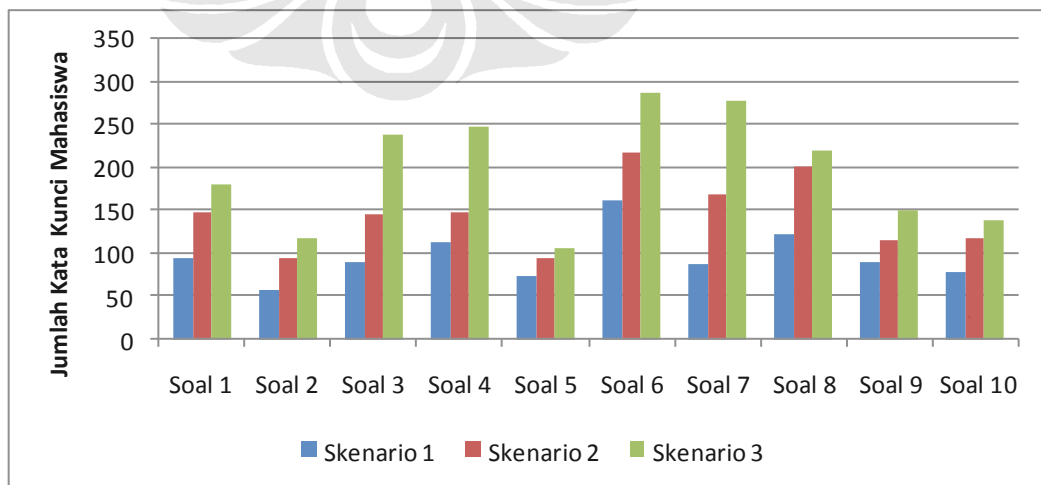
Pada Tabel 4-4 terlihat bahwa semakin banyak jumlah mahasiswa maka semakin banyak waktu yang dibutuhkan sistem untuk menilai esai. Tabel 4-4 didapat dari pengujian skenario 1, 2, dan 3 dimana setiap skenario memiliki jumlah mahasiswa dan jumlah kata kunci yang berbeda. Selain itu pengimplementasian pembobotan juga akan membuat sistem bekerja lebih lama. Hal ini dikarenakan pengimplementasian pembobotan berarti

menambah tahapan penilaian esai. Sehingga waktu yang diperlukanpun akan bertambah. Walaupun begitu, banyaknya waktu yang dibutuhkan karena adanya tahapan pembobotan tidaklah signifikan.

Tabel 4-4. Waktu Untuk Proses Penilaian Esai Skenario 1, 2, dan 3

Skenario Pengujian	Kondisi	Tanpa Pembobotan (detik)	Pembobotan SICBI (detik)
Skenario 1	10 Soal, 10 Mahasiswa,	3.86842107	3.99811983
Skenario 2	10 Soal, 15 Mahasiswa,	10.53841591	11.67230105
Skenario 3	10 Soal, 20 Mahasiswa,	23.94331002	24.99633813

Dari hasil pengamatan dapat disimpulkan bahwa hubungan antara banyaknya kata kunci dan jumlah mahasiswa adalah linier seperti yang ditunjukkan pada Gambar 4.9. Hal ini menunjukkan keragaman kata pada esai mahasiswa. Selain itu karena pada sistem tidak dilengkapai dengan persamaan kata maka keragaman kata tidak dapat dihindari. Dengan persamaan kata maka kata-kata yang memiliki persamaan, cukup disertakan salah satunya saja. Misalnya kata "kirim" dengan "mengirim". Kedua kata tersebut memiliki arti dan juga fungsi yang sama. Dengan adanya persamaan kata maka kata kunci mahasiswa yang digunakan sistem untuk pembuatan semantic space (matrik A) bisa direduksi.



Gambar 4.9. Diagram Hubungan Banyak Kata Kunci Dengan Jumlah Mahasiswa

4.2.5 Pengaruh Jumlah Mahasiswa Terhadap Unjuk Kerja Sistem

Dari hasil pengujian skenario 1, skenario 2 dan skenario 3 maka diperoleh data seperti yang terdapat pada Tabel 4-5. Pada tabel Tabel 4-5 terlihat bahwa semakin banyak jumlah mahasiswa yang mengikuti ujian maka semakin kecil range selisih penilaian esai sistem pembobotan SICBI dengan human rater. Selain itu, selisih terbesar pun akan semakin berkurang seiring dengan meningkatnya jumlah mahasiswa. Namun untuk selisih terkecil dan juga rata-rata selisih hal itu tidak terjadi. Sehingga tidak bisa ditarik kesimpulan bahwa semakin banyak mahasiswa (peserta ujian) maka akan semakin baik unjuk kerja sistem.

Untuk mengetahui lebih jauh pengaruh jumlah mahasiswa terhadap unjuk kerja sistem pembobotan SICBI, maka diperlukan penelitian lebih lanjut dengan skenario pengujian yang lebih banyak serta jumlah mahasiswa yang mengikuti ujian pun harus lebih banyak.

Tabel 4-5. Pengaruh Jumlah Mahasiswa Terhadap Unjuk Kerja Sistem Pembobotan SICBI

	Skenario 1	Skenario 2	Skenario 3
Range Selisih	2,04 – 52,80	5,67 – 28,62	0,18 – 21, 61
Selisih Terbesar	52,80	26,88	21,61
Selisih Terkecil	2,04	5,67	0,18
Rata-rata Selisih	13,98	17,84	10,90

BAB V

KESIMPULAN

Dari hasil ujicoba dan analisis yang telah dilakukan dapat diambil beberapa kesimpulan :

1. Modifikasi SIMPLE-O perlu dilakukan untuk menerapkan/mengimplementasikan pembobotan pada aplikasi *essay grading* metode LSA.
2. Kalkulasi SVD yang dilakukan dengan menggunakan *script* JAMA yang diprogram ulang menjadi program PHP bisa diterapkan sebagai pengganti MATLAB.
3. Penerapan Pembobotan SICBI meningkatkan kualitas penilaian esai jika dilihat dari selisih penilaian sistem dengan *human rater*.
4. Penerapan Pembobotan SICBI menyebabkan sistem bekerja lebih lama dalam proses penilaian esai. Walaupun demikian, tambahan waktu yang diperlukan untuk proses penilaian tidaklah terlalu signifikan.
5. Selain penerapan pembobotan, beberapa hal lain yang berpengaruh pada kecepatan proses penilaian esai antara lain banyaknya kata kunci mahasiswa dan jumlah mahasiswa yang mengikuti ujian.
6. Berdasarkan hasil pengamatan dan perhitungan dari 3 skenario pengujian yang dilakukan, skenario pengujian yang memberikan hasil paling baik (mendekati *human rater*) adalah skenario yang

memiliki jumlah mahasiswa terbanyak yaitu skenario 3 (20 mahasiswa).

DAFTAR ACUAN

- [1] Erica Chisholm, Tamara G. Kolda, *New Term Weighting Formulas for the Vector Space Method in Information Retrieval*, Oak Ridge National Laboratory, US 1999
- [2] Wild, F., et al, *Parameters Driving Effectiveness of Automated Essay Scoring with LSA*, Department of Information Systems and New Media, Vienna University of Economics and Business Administration, Austria
- [3] Haley, D. T., et al, “The Learning Grid and E-Assessment using Latent Semantic Analysis”, Computing Research Centre, UK, 2004
- [4] Palmer, J., Williams, R., Dreher, H., “Automated Essay Grading System Applied to a First Year University Subject – How Can We do it Better?”, InSITE – “Where Parallels Intersect”. Pp 1221-1229, Informing Science, Perth, Australia, June 2002.
- [5] M.W. Berry, S.T. Dumais, G.W. O’Brien, *Using Linear Algebra for Intelligent Information Retrieval*, SIAM Review 2004
- [6] Imansyah, M., *PHP & MySQL untuk Orang Awam*, CV. Maxikom, Mei 2003

DAFTAR PUSTAKA

Imansyah, M., PHP & MySQL untuk Orang Awam, CV. Maxikom, Mei 2003

Erica Chisholm, Tamara G. Kolda, *New Term Weighting Formulas for the Vector Space Method in Information Retrieval*, Oak Ridge National Laboratory, US 1999

Haley, D. T., et al, "The Learning Grid and E-Assessment using Latent Semantic Analysis", Computing Research Centre, UK, 2004

Wild, F., et al, Parameters Driving Effectiveness of Automated Essay Scoring with LSA, Department of Information Systems and New Media, Vienna University of Economics and Business Administration, Austria

M.W. Berry, S.T. Dumais, G.W. O'Brien, Using Linear Algebra for Intelligent Information Retrieval, SIAM Review 2004

Palmer, J., Williams, R., Dreher, H., "Automated Essay Grading System Applied to a First Year University Subject – How Can We do it Better?", InSITE – "Where Parallels Intersect". Pp 1221-1229, Informing Science, Perth, Australia, June 2002.

Lampiran-1: Soal Ujian

No.	Soal
Soal 1	Jelaskan apa yang dimaksud dengan point to point networks? Jelaskan juga bagaimana cara data bergerak dari satu titik ke titik yang lainnya!
Soal 2	Jelaskan apa yang dimaksudkan dengan komunikasi full duplex dan half duplex?
Soal 3	Jelaskan apa yang dimaksudkan dengan Teknologi LAN dan WAN?
Soal 4	Jelaskan apa yang dimaksud dengan CSMA/CD?
Soal 5	Jelaskan apa yang dimaksud dengan throughput?
Soal 6	Jelaskan apa yang anda ketahui mengenai kabel UTP? Jelaskan juga masalah yang diatasi oleh kabel UTP, jelaskan apa yang diatasi dengan bentuk kabel UTP serta bagaimana mengatasinya.
Soal 7	Jelaskan komponen-komponen yang diperlukan pada jaringan yang menggunakan fiber optic. Juga jelaskan apa perbedaan antara multimode dan singlemode pada kabel fiber optic!
Soal 8	Jelaskan bagaimana jaringan komputer dapat mensupport pembelajaran?
Soal 9	Jelaskan secara singkat, apa yang dimaksud dengan connection oriented service dan connectionless?
Soal 10	Jelaskan apa yang dimaksud dengan model client-server? Jelaskan juga apa yang dimaksud dengan peer to peer network?

Lampiran-2: Esai Referensi dan Kata Kunci Esai Referensi (query)

No.	Esai Referensi	Kata Kunci Referensi (query)
Soal 1	point to point networks adalah jaringan yang menghubungkan dua node secara langsung jika data hanya dapat bergerak satu arah pada satu waktu berarti jaringan beroperasi sebagai half duplex link jika data dapat dengan sukses bergerak melalui link dari masing masing node secara simultan ini merupakan full duplex link	satu,arah,satu, waktu,half, duplex, link,link,full, duplex
Soal 2	komunikasi half duplex adalah komunikasi yang hanya mengijinkan sebuah node menerima selagi node yang lain mengirim data pada media komunikasi full duplex adalah komunikasi yang mengijinkan pengiriman dan penerimaan data pada saat yang bersamaan	hanya, menerima, mengirim,saat, bersamaan
Soal 3	local area network umumnya menggunakan teknologi bandwidth tinggi yang mampu mendukung sejumlah besar host sebuah lan secara relative meliputi area geografis yang kecil sebuah kampus dengan satu gedung atau banyak gedung dan dengan densitas pengguna yang tinggi membuat teknologi ini cost effective hemat biaya wide area networks wan meliputi area geografis yang luas satu kota atau beberapa kota biaya dari physical links jarak jauh dan teknologi yang digunakan untuk membawa sinyal melalui jarak yang jauh umumnya menghasilkan kapasitas bandwidth yang rendah	area,bandwidth , tinggi,geografi s,kecil,cost, effective,hemat , biaya,luas, bandwidth
Soal 4	Csma/cd adalah mekanisme media akses dimana peralatan yang siap mengirim data terlebih dahulu memonitor media untuk mendeteksi adanya sinyal data jika selama periode waktu yang spesifik media bebas dari sinyal data peralatan	mengirim, memonitor, sinyal,serempak,collision,

	dapat mengirim data jika ada dua peralatan yang mengirim secara serempak maka terjadi sebuah collision tabrakan dan dideteksi oleh kedua peralatan akibat dari tabrakan tersebut adalah delay pengiriman kembali selama beberapa waktu acak	tabrakan,delay, waktu,acak
Soal 5	throughput adalah pengukuran pengiriman bit melalui media selama periode waktu tertentu berdasarkan sejumlah faktor throughput biasanya tidak sesuai dengan implementasi bandwidth pada physical layer seperti ethernet	pengukuran, pengiriman,bit, periode,
Soal 6	kabel utp adalah kabel yang digunakan untuk ethernet lan terdiri dari empat pasang kawat yang dililit twisted pair yang memiliki kode warna kode warna tersebut akan menentukan terminasi kabel lilitan berguna untuk menghilangkan sinyal yang mengganggu ketika ada dua kabel yang berdekatan maka medan magnet eksternal menimbulkan interferensi yang sama pada setiap kabel sepasang kabel yang dililit akan meminimalkan interferensi yang timbul ketika penerima memprosesnya dengan sama dan berlawanan akibatnya sinyal yang berasal dari interferensi elektromagnetik akan hilang cancellation effect menghilangkan juga membantu menghindari interferensi yang berasal dari sumber internal atau crosstalk crosstalk adalah interferensi medan magnet disekitar kawat didalam kabel ketika arus mengalir pada kawat akan timbul medan magnet pada kawat ketika ada arus yang mengalir berlawanan pada kawat maka akan timbul medan magnet yang berlawanan arah sehingga terjadi cancellation effect	kabel,dililit, twisted,pair, menghilangkan , sinyal, pengganggu, medan,magnet, interferensi, hilang, cancellation, effect, menghindari, crosstalk, didalam,kabel,
Soal 7	baik laser ataupun light emitting diode led menghasilkan pulsa cahaya yang digunakan untuk merepresentasikan data yang dikirim sebagai bit pada media alat elektronik semikonduktor yang disebut photo diode mendeteksi pulsa cahaya dan mengubahnya menjadi tegangan yang bisa	laser,light, emitting,diode, led,photo,diode , singlemode,las

	<p>dibentuk kembali menjadi frame data singlemode fiber optic membawa sebuah sinar dari cahaya biasanya dihasilkan dari laser karena sinar laser satu arah dan dapat berjalan ke pusat fiber tipe fiber ini dapat mengirimkan pulsa optic untuk jarak yang jauh multimode fiber biasanya menggunakan led emitter yang tidak menciptakan gelombang cahaya koheren searah tetapi fiber dengan sudut yang berbeda beda karena cahaya memasuki fiber dengan sudut yang berbeda beda maka dibutuhkan waktu untuk jalan ke fiber fiber yang panjang dapat membuat hasil pada pulsanya menjadi blur pada penerimanya efek ini dikenal juga sebagai modal dispersion batas panjang dari multimode segmen fiber multimode fiber dan sumber cahaya led lebih murah daripada singlemode fiber dan ini berdasarkan teknologi laser emitter</p>	<p>er, jarak, jauh, multimode, led, modal, batas, panjang, lebih, murah</p>
Soal 8	<p>jaringan komputer yang kuat dan handal akan memperkaya pengalaman belajar dari pelajar jaringan komputer ini mengantarkan materi belajar dalam format yang luas materinya termasuk aktifitas interaktif dengan assessment ujian dan feedback umpan balik pembelajaran yang diberikan menggunakan jaringan komputer atau internet biasanya disebut online learning atau learning tersedianya perangkat learning memperbanyak sumber pembelajaran bagi siswa metode pembelajaran tradisional setidaknya menyediakan dua sumber yaitu instruktur dan buku teks sumber ini terbatas pada tempat dan waktu pada kenyataannya kuliah online dapat menyediakan suara data dan video bagi siswa kapanpun dan dimanapun siswa dapat mencari referensi yang berbeda sehingga dapat meningkatkan pengalaman belajar online discussion groups dan message boards menjadikan siswa dapat berkolaborasi dengan instruktur dengan siswa lainnya akses pada instruksi</p>	<p>mengantarkan, materi, luas, interaktif, memperbanyak, sumber, suara, data, video, kapanpun, dimanapun, akses, menghilangkan, batasan, meningkatkan, kesempatan</p>

	yang berkualitas tinggi tidak lagi terbatas online distance learning telah menghilangkan batasan geografi dan meningkatkan kesempatan pelajar	
Soal 9	connection oriented service adalah jenis layanan yang memerlukan pertukaran data control untuk membentuk koneksi sebelum data dikirim data berikutnya akan dikirim setelah pengirim mendapat acknowledgement sudah diterimanya data sebelumnya dari penerima connectionless service adalah jenis layanan dimana pengiriman data dilakukan tanpa memberitahu penerima sehingga data diterima tidak berurutan	memerlukan, membentuk, koneksi, acknowledgement, tanpa, memberitahu, penerima, berurutan,
Soal 10	pada model client server peralatan yang meminta informasi disebut client dan peralatan yang merespon permintaan disebut server proses proses pada client dan server dianggap sebagai bagian dari application layer client memulai pertukaran dengan permintaan data dari server direspon dengan mengirim satu atau lebih aliran data ke client protokol application layer mendeskripsikan format dari permintaan dan respon antara client dan server sebagai tambahan pada data transfer yang aktual pertukaran ini memerlukan kontrol informasi seperti otentifikasi dan identifikasi dari file data file yang akan dikirim pada peer to peer network dua atau lebih komputer terhubung melalui jaringan dan dapat berbagi resource seperti printer atau file tanpa mempunyai server yang dedicated resource terdesentralisasi pada jaringan setiap peralatan yang terhubung dikenal sebagai peer dapat berfungsi sebagai server atau client satu computer dapat diasumsikan bertugas sebagai server untuk satu transaksi sedangkan secara bersamaan berfungsi sebagai client untuk computer lain tugas sebagai client dan server ditentukan berdasarkan request user accounts dan hak akses harus ditentukan secara	meminta, informasi, client, merespon, permintaan, server, kontrol, informasi, otentifikasi, identifikasi, berbagi, resource, terdesentralisasi, jaringan, server, client, bersamaan, hak, akses, individual,

individual pada setiap peralatan peer

Lampiran-3: Penilaian *Human Rater*

MHS	No.	Nilai	MHS	No.	Nilai	MHS	No.	Nilai
mhs2	1	50	mhs10	1	50	mhs19	1	100
mhs2	2	100	mhs10	2	100	mhs19	2	100
mhs2	3	60	mhs10	3	70	mhs19	3	30
mhs2	4	30	mhs10	4	50	mhs19	4	60
mhs2	5	0	mhs10	5	30	mhs19	5	80
mhs2	6	20	mhs10	6	40	mhs19	6	100
mhs2	7	20	mhs10	7	10	mhs19	7	50
mhs2	8	60	mhs10	8	80	mhs19	8	60
mhs2	9	80	mhs10	9	40	mhs19	9	100
mhs2	10	40	mhs10	10	40	mhs19	10	70
mhs4	1	70	mhs11	1	30	mhs26	1	30
mhs4	2	60	mhs11	2	80	mhs26	2	70
mhs4	3	60	mhs11	3	60	mhs26	3	60
mhs4	4	50	mhs11	4	100	mhs26	4	10
mhs4	5	10	mhs11	5	10	mhs26	5	10
mhs4	6	10	mhs11	6	70	mhs26	6	10
mhs4	7	30	mhs11	7	20	mhs26	7	20
mhs4	8	100	mhs11	8	60	mhs26	8	10
mhs4	9	80	mhs11	9	80	mhs26	9	40
mhs4	10	70	mhs11	10	80	mhs26	10	20
mhs5	1	50	mhs12	1	60	mhs35	1	0
mhs5	2	100	mhs12	2	80	mhs35	2	100
mhs5	3	60	mhs12	3	80	mhs35	3	40
mhs5	4	10	mhs12	4	60	mhs35	4	10
mhs5	5	0	mhs12	5	70	mhs35	5	80
mhs5	6	20	mhs12	6	100	mhs35	6	100
mhs5	7	0	mhs12	7	20	mhs35	7	60
mhs5	8	70	mhs12	8	70	mhs35	8	40
mhs5	9	80	mhs12	9	100	mhs35	9	20
mhs5	10	50	mhs12	10	80	mhs35	10	10
mhs6	1	40	mhs13	1	50	mhs36	1	40
mhs6	2	20	mhs13	2	100	mhs36	2	20
mhs6	3	40	mhs13	3	70	mhs36	3	30
mhs6	4	100	mhs13	4	60	mhs36	4	20
mhs6	5	10	mhs13	5	80	mhs36	5	10
mhs6	6	50	mhs13	6	70	mhs36	6	30
mhs6	7	0	mhs13	7	90	mhs36	7	10
mhs6	8	30	mhs13	8	100	mhs36	8	30
mhs6	9	70	mhs13	9	90	mhs36	9	10

mhs6	10	60		mhs13	10	80		mhs36	10	30
mhs7	1	50		mhs14	1	40		mhs37	1	30
mhs7	2	100		mhs14	2	100		mhs37	2	100
mhs7	3	60		mhs14	3	60		mhs37	3	80
mhs7	4	70		mhs14	4	100		mhs37	4	20
mhs7	5	10		mhs14	5	30		mhs37	5	10
mhs7	6	30		mhs14	6	100		mhs37	6	80
mhs7	7	20		mhs14	7	10		mhs37	7	10
mhs7	8	70		mhs14	8	70		mhs37	8	10
mhs7	9	80		mhs14	9	90		mhs37	9	100
mhs7	10	40		mhs14	10	80		mhs37	10	50
mhs8	1	40		mhs17	1	40		mhs41	1	30
mhs8	2	100		mhs17	2	80		mhs41	2	80
mhs8	3	60		mhs17	3	60		mhs41	3	60
mhs8	4	60		mhs17	4	80		mhs41	4	40
mhs8	5	0		mhs17	5	10		mhs41	5	30
mhs8	6	50		mhs17	6	60		mhs41	6	20
mhs8	7	40		mhs17	7	30		mhs41	7	90
mhs8	8	40		mhs17	8	70		mhs41	8	60
mhs8	9	60		mhs17	9	100		mhs41	9	100
mhs8	10	70		mhs17	10	10		mhs41	10	100
mhs9	1	50		mhs18	1	50				
mhs9	2	50		mhs18	2	100				
mhs9	3	70		mhs18	3	70				
mhs9	4	70		mhs18	4	20				
mhs9	5	20		mhs18	5	100				
mhs9	6	80		mhs18	6	20				
mhs9	7	70		mhs18	7	20				
mhs9	8	60		mhs18	8	60				
mhs9	9	80		mhs18	9	100				
mhs9	10	80		mhs18	10	30				

Lampiran-4: Penilaian SIMPLE-OM (skenario 1)

Mhs	No.	Nilai 1	Nilai 2	Mhs	No.	Nilai 1	Nilai 2	Mhs	No.	Nilai 1	Nilai 2
mhs2	1	98.37	60.12	mhs7	1	95.07	40.47	mhs11	1	99.01	55.34
mhs2	2	35.10	81.08	mhs7	2	82.12	40.08	mhs11	2	59.56	74.76
mhs2	3	53.17	70.39	mhs7	3	27.76	2.45	mhs11	3	9.62	63.01
mhs2	4	95.67	1.48	mhs7	4	84.12	99.89	mhs11	4	53.47	85.41
mhs2	5	60.23	79.26	mhs7	5	80.18	82.16	mhs11	5	89.14	91.56
mhs2	6	65.71	92.11	mhs7	6	53.27	79.36	mhs11	6	81.39	51.23
mhs2	7	68.70	56.43	mhs7	7	54.18	32.59	mhs11	7	86.47	10.37
mhs2	8	90.15	59.64	mhs7	8	5.10	44.61	mhs11	8	25.48	50.54
mhs2	9	99.86	90.42	mhs7	9	98.69	59.61	mhs11	9	95.58	35.99
mhs2	10	22.06	26.99	mhs7	10	99.15	71.53	mhs11	10	47.33	57.80
mhs4	1	41.79	89.14	mhs8	1	93.49	49.64	mhs12	1	33.28	87.67
mhs4	2	22.18	80.22	mhs8	2	66.49	55.14	mhs12	2	73.66	7.31
mhs4	3	84.56	54.95	mhs8	3	59.83	43.47	mhs12	3	40.07	32.54
mhs4	4	85.74	19.97	mhs8	4	92.27	99.99	mhs12	4	98.21	6.59
mhs4	5	34.02	14.00	mhs8	5	84.17	79.83	mhs12	5	75.38	100.00
mhs4	6	79.38	69.69	mhs8	6	99.77	39.86	mhs12	6	89.54	24.20
mhs4	7	52.34	27.89	mhs8	7	15.14	41.70	mhs12	7	99.86	90.63
mhs4	8	99.76	81.90	mhs8	8	38.22	61.19	mhs12	8	22.94	4.14
mhs4	9	17.92	83.85	mhs8	9	93.98	31.80	mhs12	9	93.72	32.41
mhs4	10	88.77	83.78	mhs8	10	89.77	7.58	mhs12	10	97.67	16.11
mhs5	1	81.03	0.68	mhs9	1	91.70	27.48	Ket: • Nilai 1 = Penilaian Sistem Tanpa Pembobotan • Nilai 2 = Penilaian Sistem Pembobotan SICBI			
mhs5	2	99.56	32.77	mhs9	2	51.55	70.16				
mhs5	3	84.52	63.70	mhs9	3	99.97	82.84				
mhs5	4	80.66	94.43	mhs9	4	91.70	7.01				
mhs5	5	77.47	25.37	mhs9	5	43.78	21.98				
mhs5	6	91.55	25.73	mhs9	6	82.65	27.97				
mhs5	7	39.98	36.61	mhs9	7	35.34	35.47				
mhs5	8	0.95	13.93	mhs9	8	12.76	13.64				
mhs5	9	99.94	99.97	mhs9	9	97.28	47.67				
mhs5	10	3.54	75.41	mhs9	10	12.58	77.11				
mhs6	1	82.93	6.50	mhs10	1	46.72	98.21				
mhs6	2	99.98	73.99	mhs10	2	32.26	96.10				
mhs6	3	98.14	94.66	mhs10	3	95.19	91.58				
mhs6	4	70.62	22.00	mhs10	4	96.71	97.25				
mhs6	5	93.60	94.85	mhs10	5	99.85	98.97				
mhs6	6	95.35	37.13	mhs10	6	80.63	53.41				
mhs6	7	98.72	97.49	mhs10	7	98.43	90.01				
mhs6	8	66.11	98.48	mhs10	8	14.09	78.17				
mhs6	9	96.11	81.27	mhs10	9	97.59	94.15				
mhs6	10	98.78	80.48	mhs10	10	98.37	79.51				

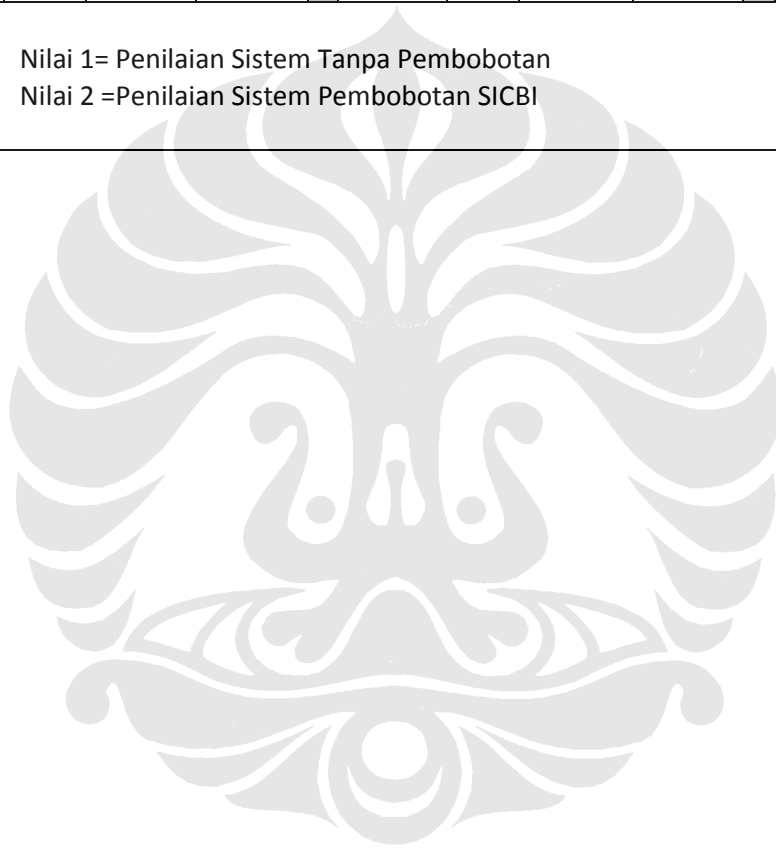
Lampiran-5: Penilaian SIMPLE-OM (skenario 2)

Mhs	No.	Nilai 1	Nilai 2	Mhs	No.	Nilai 1	Nilai 2	Mhs	No.	Nilai 1	Nilai 2
mhs2	1	85.93	65.51	mhs8	1	99.11	57.74	mhs13	1	99.80	42.84
mhs2	2	13.98	82.12	mhs8	2	59.72	69.71	mhs13	2	73.69	71.47
mhs2	3	97.26	79.39	mhs8	3	72.37	28.94	mhs13	3	79.87	34.08
mhs2	4	90.01	67.63	mhs8	4	54.85	94.17	mhs13	4	82.43	99.81
mhs2	5	98.88	84.78	mhs8	5	95.65	98.99	mhs13	5	97.45	31.90
mhs2	6	96.10	73.52	mhs8	6	99.23	67.67	mhs13	6	96.41	68.15
mhs2	7	37.84	46.99	mhs8	7	8.14	40.01	mhs13	7	99.86	94.98
mhs2	8	99.95	99.90	mhs8	8	99.27	99.40	mhs13	8	99.75	95.35
mhs2	9	84.27	24.50	mhs8	9	97.58	14.32	mhs13	9	97.43	42.22
mhs2	10	36.48	7.49	mhs8	10	32.26	20.36	mhs13	10	86.62	96.15
mhs4	1	86.62	91.16	mhs9	1	99.58	42.54	mhs14	1	85.35	22.50
mhs4	2	11.24	1.24	mhs9	2	36.88	77.29	mhs14	2	55.70	62.32
mhs4	3	62.33	62.76	mhs9	3	39.36	87.43	mhs14	3	69.50	57.93
mhs4	4	99.77	45.88	mhs9	4	89.34	72.22	mhs14	4	88.04	9.06
mhs4	5	59.59	95.55	mhs9	5	71.57	95.86	mhs14	5	67.07	7.48
mhs4	6	94.79	73.02	mhs9	6	94.45	69.42	mhs14	6	97.20	70.91
mhs4	7	16.70	32.58	mhs9	7	8.00	30.29	mhs14	7	60.03	43.40
mhs4	8	88.57	98.46	mhs9	8	96.88	91.17	mhs14	8	96.17	97.19
mhs4	9	42.60	97.06	mhs9	9	87.15	39.31	mhs14	9	78.78	89.79
mhs4	10	58.58	88.28	mhs9	10	10.98	24.98	mhs14	10	53.79	26.48
mhs5	1	98.53	10.94	mhs10	1	13.48	98.69	mhs17	1	99.71	0.11
mhs5	2	98.29	32.10	mhs10	2	27.77	94.51	mhs17	2	70.78	34.12
mhs5	3	74.75	65.21	mhs10	3	63.15	91.58	mhs17	3	66.47	61.24
mhs5	4	99.72	94.10	mhs10	4	87.08	99.82	mhs17	4	31.31	98.34
mhs5	5	86.91	97.13	mhs10	5	66.24	99.92	mhs17	5	92.98	16.96
mhs5	6	97.25	67.46	mhs10	6	95.04	68.99	mhs17	6	24.09	77.63
mhs5	7	17.55	37.31	mhs10	7	89.90	89.91	mhs17	7	68.83	81.90
mhs5	8	99.74	99.99	mhs10	8	99.66	97.79	mhs17	8	8.97	14.64
mhs5	9	76.38	93.58	mhs10	9	89.72	95.79	mhs17	9	86.83	15.89
mhs5	10	49.12	11.00	mhs10	10	70.94	71.01	mhs17	10	0.99	2.47
mhs6	1	99.99	1.55	mhs11	1	87.09	50.44	mhs18	1	98.35	14.79
mhs6	2	99.50	78.79	mhs11	2	44.81	84.88	mhs18	2	72.19	85.58
mhs6	3	48.82	95.73	mhs11	3	73.72	71.11	mhs18	3	81.09	14.78
mhs6	4	79.74	20.40	mhs11	4	58.85	99.51	mhs18	4	71.96	99.12
mhs6	5	64.03	8.47	mhs11	5	95.11	45.50	mhs18	5	78.20	39.22
mhs6	6	100.00	13.10	mhs11	6	95.72	67.66	mhs18	6	97.64	67.56
mhs6	7	94.57	98.15	mhs11	7	60.39	1.97	mhs18	7	13.82	30.86
mhs6	8	86.22	8.78	mhs11	8	99.77	91.28	mhs18	8	95.52	93.79
mhs6	9	56.92	99.64	mhs11	9	93.53	24.97	mhs18	9	95.83	81.28

mhs6	10	85.47	84.96		mhs11	10	5.65	44.26		mhs18	10	49.47	85.70
mhs7	1	92.80	30.15		mhs12	1	67.53	89.90		mhs19	1	98.00	3.11
mhs7	2	81.33	12.43		mhs12	2	73.65	23.82		mhs19	2	81.60	65.30
mhs7	3	80.89	16.24		mhs12	3	98.49	41.35		mhs19	3	83.06	17.21
mhs7	4	64.19	93.60		mhs12	4	87.33	77.58		mhs19	4	83.59	95.33
mhs7	5	84.92	99.91		mhs12	5	90.20	44.10		mhs19	5	99.35	23.54
mhs7	6	98.90	46.91		mhs12	6	98.05	67.25		mhs19	6	98.57	65.00
mhs7	7	14.44	34.27		mhs12	7	99.83	92.88		mhs19	7	47.50	34.79
mhs7	8	99.70	99.94		mhs12	8	99.87	100.00		mhs19	8	98.51	99.90
mhs7	9	84.83	0.28		mhs12	9	98.21	49.95		mhs19	9	99.46	58.19
mhs7	10	5.91	44.74		mhs12	10	20.87	0.54		mhs19	10	100.00	94.53

Ket:

- Nilai 1= Penilaian Sistem Tanpa Pembobotan
- Nilai 2 =Penilaian Sistem Pembobotan SICBI



Lampiran-6: Penilaian SIMPLE-OM (skenario 3)

Mhs	No.	Nilai 1	Nilai 2	Mhs	No.	Nilai 1	Nilai 2	Mhs	No.	Nilai 1	Nilai 2
mhs2	1	83.59	68.12	mhs10	1	14.67	99.05	mhs19	1	98.58	0.84
mhs2	2	9.95	70.72	mhs10	2	32.04	99.85	mhs19	2	81.71	99.02
mhs2	3	99.25	64.12	mhs10	3	97.63	90.43	mhs19	3	99.92	46.75
mhs2	4	64.98	31.04	mhs10	4	93.97	95.88	mhs19	4	87.18	83.78
mhs2	5	88.67	90.15	mhs10	5	94.35	100.00	mhs19	5	97.62	22.84
mhs2	6	96.09	68.15	mhs10	6	92.84	60.13	mhs19	6	97.07	56.31
mhs2	7	95.31	35.52	mhs10	7	95.97	92.62	mhs19	7	87.18	35.59
mhs2	8	51.01	35.64	mhs10	8	40.93	93.31	mhs19	8	73.52	32.15
mhs2	9	71.58	12.35	mhs10	9	83.22	95.66	mhs19	9	99.03	65.19
mhs2	10	32.11	47.33	mhs10	10	51.05	65.98	mhs19	10	97.29	93.94
mhs4	1	79.42	93.71	mhs11	1	84.72	51.10	mhs26	1	99.93	4.11
mhs4	2	10.01	4.11	mhs11	2	26.44	99.86	mhs26	2	51.58	99.77
mhs4	3	99.22	68.08	mhs11	3	96.82	72.96	mhs26	3	100.00	95.32
mhs4	4	55.65	44.52	mhs11	4	72.86	66.21	mhs26	4	63.54	32.15
mhs4	5	99.52	95.89	mhs11	5	97.87	66.24	mhs26	5	99.87	28.11
mhs4	6	93.25	67.86	mhs11	6	93.36	58.81	mhs26	6	94.42	64.40
mhs4	7	95.91	37.43	mhs11	7	97.58	4.44	mhs26	7	84.70	20.72
mhs4	8	42.19	79.29	mhs11	8	47.48	64.67	mhs26	8	49.97	29.54
mhs4	9	47.56	95.79	mhs11	9	88.54	16.45	mhs26	9	92.44	38.95
mhs4	10	46.72	85.59	mhs11	10	25.82	51.11	mhs26	10	21.64	56.48
mhs5	1	99.46	5.58	mhs12	1	67.01	93.08	mhs35	1	99.97	24.82
mhs5	2	99.33	65.66	mhs12	2	79.27	3.01	mhs35	2	71.86	87.16
mhs5	3	99.87	50.56	mhs12	3	93.36	34.62	mhs35	3	94.36	66.90
mhs5	4	72.98	63.52	mhs12	4	76.17	32.22	mhs35	4	74.09	55.69
mhs5	5	98.97	97.83	mhs12	5	97.70	58.28	mhs35	5	98.55	9.25
mhs5	6	96.97	57.94	mhs12	6	97.51	59.04	mhs35	6	92.54	58.32
mhs5	7	91.95	41.79	mhs12	7	69.38	94.11	mhs35	7	98.19	18.66
mhs5	8	46.04	51.80	mhs12	8	51.16	32.88	mhs35	8	83.62	49.00
mhs5	9	59.06	92.37	mhs12	9	96.82	52.42	mhs35	9	94.87	46.48
mhs5	10	8.21	15.42	mhs12	10	47.84	0.19	mhs35	10	86.40	85.82
mhs6	1	99.99	6.06	mhs13	1	97.40	44.82	mhs36	1	94.39	7.83
mhs6	2	98.02	16.20	mhs13	2	60.60	99.98	mhs36	2	52.22	98.95
mhs6	3	98.47	94.48	mhs13	3	99.44	32.47	mhs36	3	98.98	77.08
mhs6	4	60.24	56.04	mhs13	4	89.51	95.39	mhs36	4	70.21	48.80
mhs6	5	99.46	7.61	mhs13	5	97.93	44.25	mhs36	5	99.29	3.60
mhs6	6	99.99	52.92	mhs13	6	95.16	59.22	mhs36	6	98.02	55.12
mhs6	7	85.27	97.58	mhs13	7	60.92	94.85	mhs36	7	6.43	96.90
mhs6	8	47.27	96.56	mhs13	8	94.01	55.68	mhs36	8	82.62	72.59
mhs6	9	38.17	99.08	mhs13	9	97.04	57.07	mhs36	9	94.82	59.00
mhs6	10	60.43	90.16	mhs13	10	97.59	94.75	mhs36	10	68.34	26.11
mhs7	1	92.04	28.75	mhs14	1	84.39	25.31	mhs37	1	96.01	23.87

mhs7	2	78.98	58.06	mhs14	2	47.75	76.58	mhs37	2	92.50	99.97
mhs7	3	98.20	48.89	mhs14	3	98.55	28.79	mhs37	3	18.12	8.71
mhs7	4	85.02	98.02	mhs14	4	67.76	38.72	mhs37	4	69.26	99.99
mhs7	5	99.69	99.99	mhs14	5	96.90	6.83	mhs37	5	5.19	10.05
mhs7	6	98.73	26.94	mhs14	6	97.14	64.35	mhs37	6	99.87	24.11
mhs7	7	93.52	38.05	mhs14	7	94.66	51.52	mhs37	7	35.02	58.56
mhs7	8	51.28	18.38	mhs14	8	66.09	67.65	mhs37	8	81.79	88.09
mhs7	9	73.72	12.92	mhs14	9	79.68	90.83	mhs37	9	96.65	78.63
mhs7	10	43.43	47.86	mhs14	10	40.10	46.85	mhs37	10	87.86	88.50
mhs8	1	98.05	59.60	mhs17	1	99.18	2.89	mhs41	1	95.90	43.09
mhs8	2	66.31	8.58	mhs17	2	66.74	63.95	mhs41	2	97.39	23.00
mhs8	3	99.33	5.66	mhs17	3	95.44	75.14	mhs41	3	96.91	35.49
mhs8	4	71.76	92.21	mhs17	4	76.67	94.67	mhs41	4	94.33	99.91
mhs8	5	99.62	99.83	mhs17	5	91.13	19.54	mhs41	5	93.62	4.15
mhs8	6	99.57	56.65	mhs17	6	20.34	86.63	mhs41	6	99.10	7.74
mhs8	7	61.22	42.38	mhs17	7	22.56	55.72	mhs41	7	99.32	48.81
mhs8	8	83.28	34.94	mhs17	8	48.81	43.36	mhs41	8	51.19	74.38
mhs8	9	96.92	14.90	mhs17	9	62.80	0.52	mhs41	9	92.83	43.53
mhs8	10	73.42	10.39	mhs17	10	47.75	4.79	mhs41	10	40.93	49.28
mhs9	1	98.10	46.62	mhs18	1	97.44	4.27	Ket: • Nilai 1 = Penilaian Sistem Tanpa Pembobotan • Nilai 2 = Penilaian Sistem Pembobotan SICBI			
mhs9	2	33.74	97.92	mhs18	2	62.93	99.71				
mhs9	3	99.56	73.97	mhs18	3	98.92	14.24				
mhs9	4	66.01	16.50	mhs18	4	85.14	66.86				
mhs9	5	99.29	96.55	mhs18	5	98.85	47.49				
mhs9	6	92.94	61.15	mhs18	6	97.30	59.23				
mhs9	7	89.19	45.43	mhs18	7	70.73	41.59				
mhs9	8	45.03	25.32	mhs18	8	51.50	43.80				
mhs9	9	74.21	32.86	mhs18	9	90.68	86.47				
mhs9	10	15.45	47.62	mhs18	10	40.29	78.30				