BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponen dengan maksud untuk mengidentifikasikan dan melakukan evaluasi permsalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga menghasilkan rancangan sistem yang siap untuk diimplementasikan.

3.1.1 Analisis Masalah

Berdasarkan latar belakang masalah yang telah dijelaskan, yang menjadi titik permasalahan dalam penelitian ini adalah sebagai berikut:

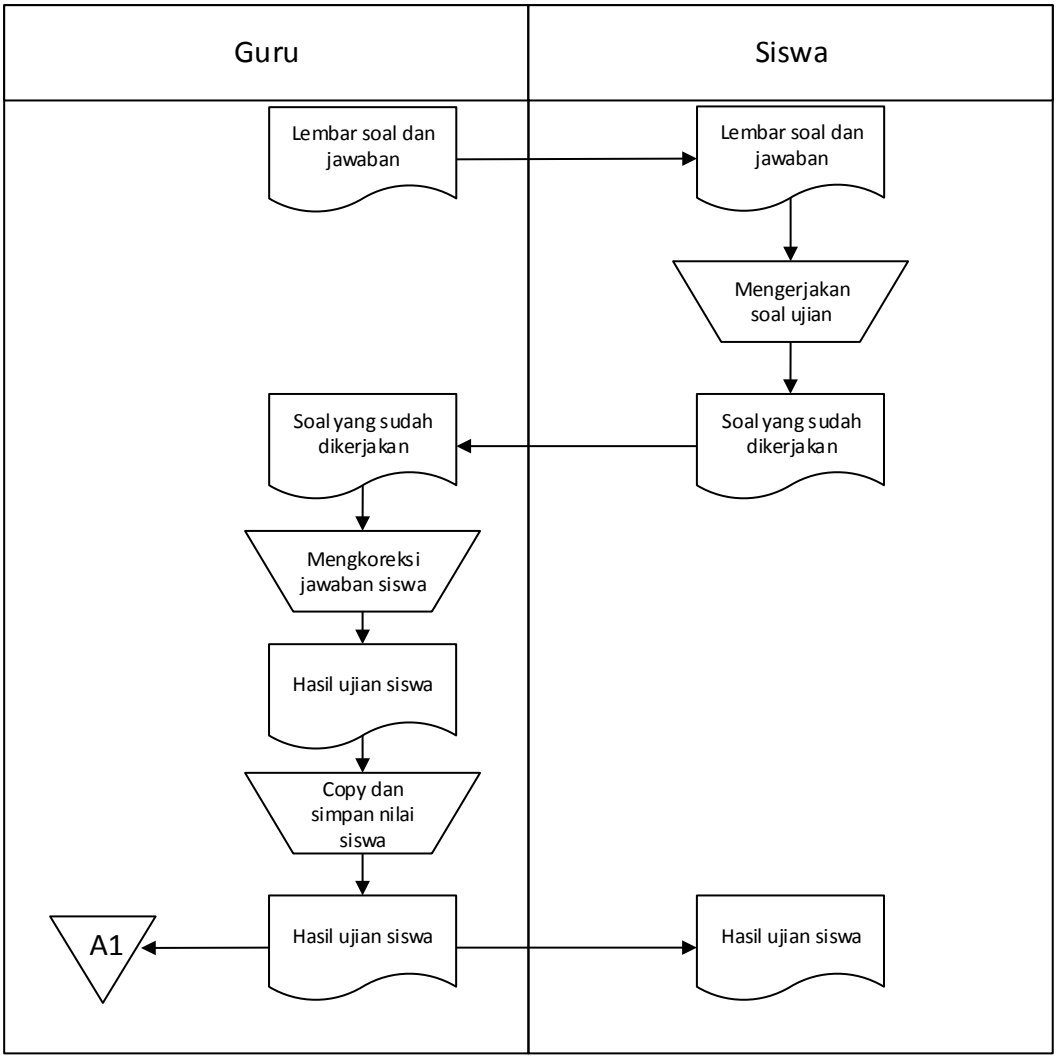
1. Belum adanya pengkoreksian ujian esai secara otomatis oleh sistem LMS yang sudah ada.
2. Pengkoreksian soal esai dilakukan secara manual oleh guru yang membutuhkan waktu yang tidak sebentar.

3.1.2 Analisis Sistem yang Sedang Berjalan

Berdasarkan hasil penelitian yang dilakukan di SMK N 1 Bulakamba maka dapat digambarkan prosedur sistem ujian esai yang sedang berjalan kedalam bentuk *flowmap* diagram sebagai berikut:

1. Guru memberikan soal ujian kepada siswa
2. Siswa mulai mengerjakan soal ujian
3. Siswa mengumpulkan kembali jawaban yang sudah terisi
4. Guru melakukan koreksi secara manual oleh siswa
5. Hasil koreksi guru dimasukan kedalam data nilai

6. Guru membagikan lembar jawaban yang sudah dikoreksi
7. Siswa menerima nilai hasil ujian



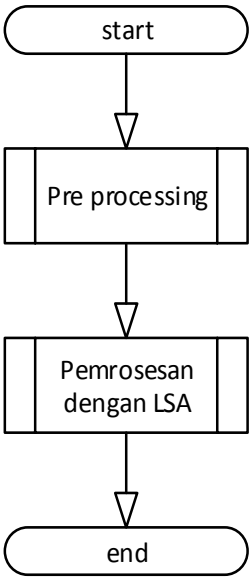
Gambar 3. 1 Flowmap Sistem yang Sedang Berjalan

Keterangan

A1 = Arsip hasil ujian siswa

3.2 Analisis Metode

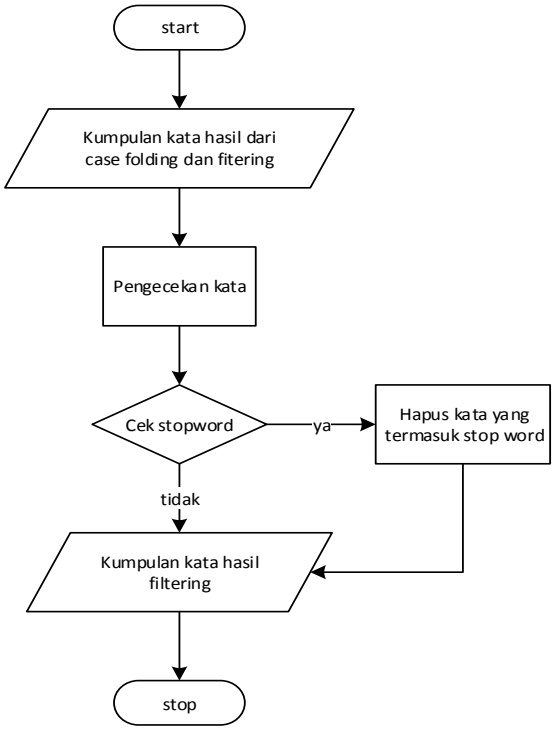
Sistem yang akan dikembangkan dalam penelitian ini adalah sebuah sistem koreksi soal esai otomatis dengan metode *Latent Semantyc Analysis*. Pada prosesnya sebelum sistem melakukan perhitungan dengan metode LSA ada proses proses yang dilakukan terlebih dahulu. Pengembangan koreksi esai ini terdiri dari dua subproses yaitu proses *preprocessing* dan proses LSA itu sendiri. Proses *preprocessing* terdiri dari *case folding*, *tokenizing*, *filtering*, dan *stemming*. Berikut adalah *flowchart* dari masing masing proses.



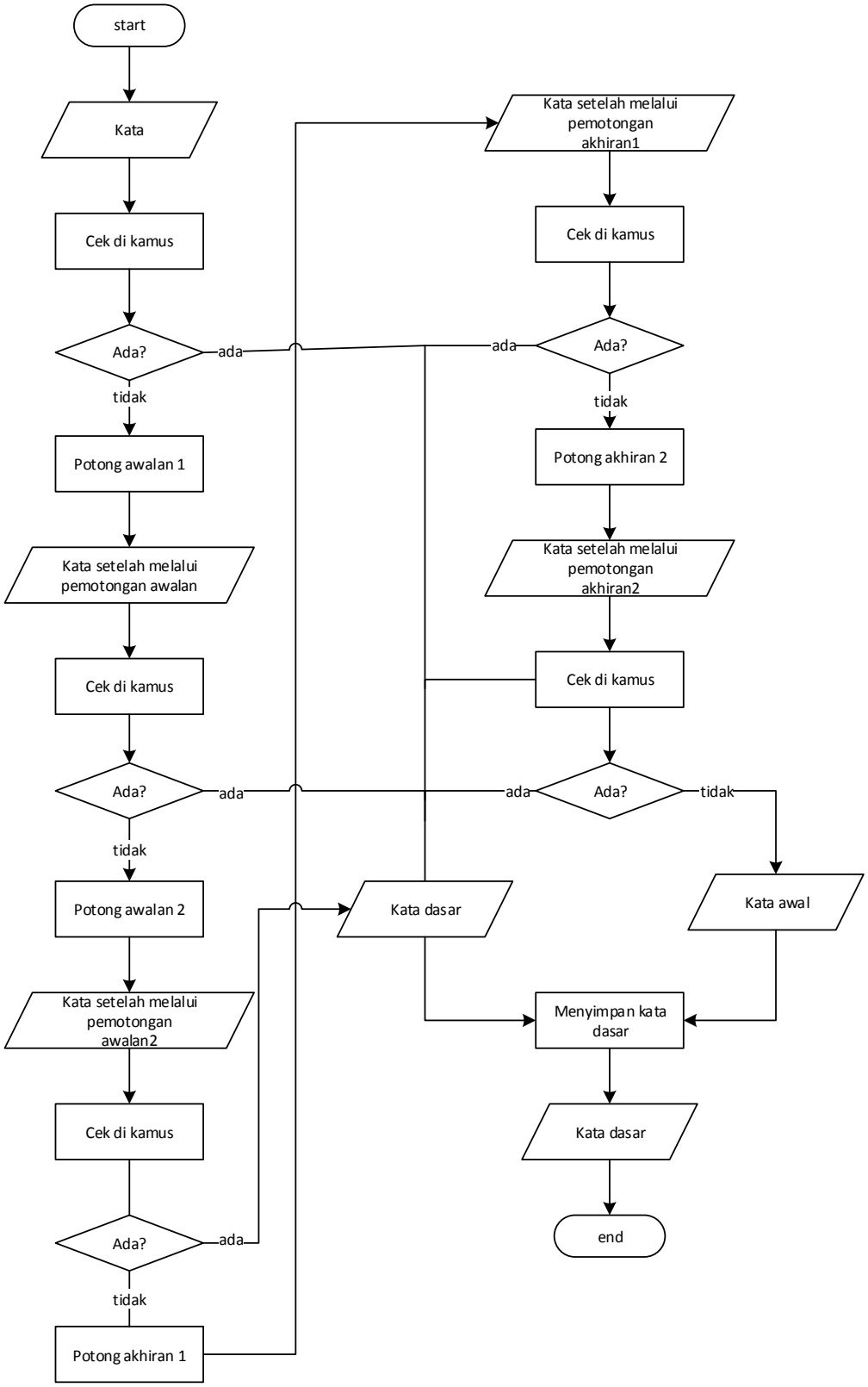
Gambar 3. 2 Flowchart Sistem Koreksi Esai Otomatis



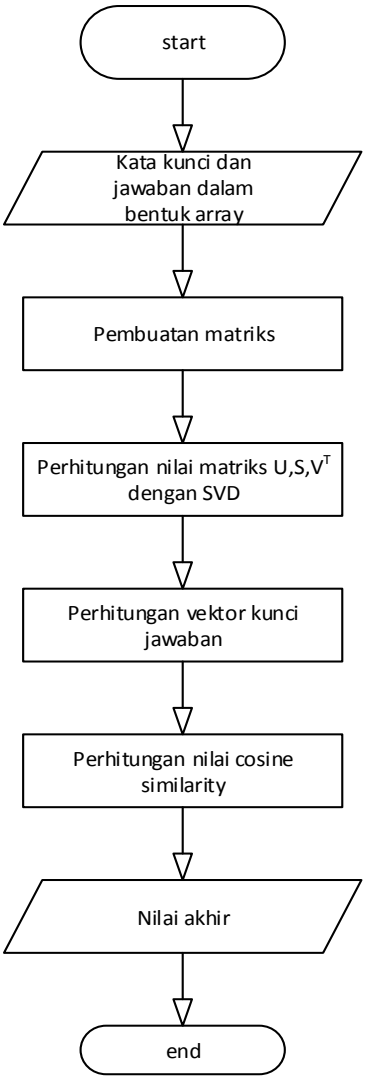
Gambar 3. 3 Flowchart case folding dan tokenizing



Gambar 3. 4 Flowchart proses Filtering



Gambar 3. 5 Flowchart proses Stemming



Gambar 3. 6 Flowchart proses LSA

Untuk lebih memudahkan dalam pemahaman dalam prosesnya diberikan sebuah contoh pencocokan antara kunci jawaban dan jawaban siswa. Dalam contoh berikut adalah salah satu *quiz* yang diadakan dalam suatu mata pelajaran yang berjumlah 2 soal esai. Jika nilai sempurna dalam *quiz* ini adalah 10 maka setiap soal memiliki bobot 5 dalam penyusunan nilai tersebut.

Untuk perhitungan kali ini menggunakan 1 dari 2 soal tersebut untuk memudahkan dalam pemahaman berikut adalah contoh soal, kunci, dan jawaban dari 3 siswa.

Pertanyaan :

"Apa yang dimaksud dengan konduktor???"

Kunci jawaban :

" Bahan atau Zat yang dapat menghantarkan aliran listrik dengan baik"

Jawaban siswa :

1. " bahan yang mampu mengalirkan listrik dengan baik"
2. " zat yang dapat mengalirkan arus listrik"
3. " bahan yang dapat menghantarkan arus listrik"

3.2.1 Analisis *Preprocessing*

Tahapan *preprocessing* dilakukan untuk mengekstraksi data kunci dan jawaban siswa. Hasil dari tahapan ini adalah kumpulan term yang akan digunakan untuk proses perhitungan LSA. Berikut adalah analisis dari tiap tahapan dalam *preprocessing*.

3.2.1.1 Case Folding dan Tokenizing

Tahapan pertama dalam sub proses *preprocessing* adalah *case folding* dan *tokenizing*. Pada tahap ini kalimat jawaban dari siswa dan guru akan diproses untuk menghasilkan token-token (*term*) yang sudah dalam bentuk huruf kecil semua. Berikut adalah gambar *flowchart* dari proses *case folding* dan *tokenizing*.

Tahapan *case folding* adalah tahapan untuk merubah seluruh huruf dari kalimat jawaban dan kunci jawaban menjadi huruf kecil semua. Sedangkan tahapan *tokenizing* adalah tahapan untuk merubah kalimat jawaban dan kunci menjadi perkata (*term*).

Hasil dari proses *case folding* dan t*okenizing* pada contoh soal dan jawaban diatas dapat dilihat pada tabel berikut.

Tabel 3. 1 Hasil proses case folding

No	Kalimat awal	Setelah proses *case folding*
1	Bahan atau Zat yang dapat menghantarkan aliran listrik dengan baik	bahan atau zat yang dapat menghantarkan aliran listrik dengan baik
2	Bahan yang mampu mengalirkan listrik dengan baik	bahan yang mampu mengalirkan listrik dengan baik
3	Zat yang dapat mengalirkan arus listrik	zat yang dapat mengalirkan arus listrik
4	Bahan yang dapat menghantarkan arus listrik	bahan yang dapat menghantarkan arus listrik

Tabel 3. 2 Hasil proses Tokenizing

No	Kalimat awal	Setelah proses *tokenizing*		
1	bahan atau zat yang dapat menghantarkan aliran listrik dengan baik	bahan	atau	zat
		yang	dapat	menghantarkan
		aliran	listrik	dengan
		baik		
2	bahan yang mampu mengalirkan listrik dengan baik	bahan	yang	mampu
		mengalirkan	listrik	dengan
		baik		

3	zat yang dapat mengalirkan arus listrik	zat	yang	dapat
		mengalirkan	arus	listrik
4	bahan yang dapat menghanatarkan arus listrik	bahan	yang	dapat
		menghantarkan	arus	listrik

3.2.1.2 *Filtering*

Tahap *filtering* merupakan tahap penyaringan kata-kata hasil dari proses c*ase folding* dan *tokenizing*. Pada tahap ini terdapat beberapa kriteria yang akan diambil untuk menjadi *term*, diantaranya adalah.

1. Bukan termasuk kedalam *stopword*
2. Mempunyai makna atau arti yang terdaftar dalam KBBI.

Selain kata penghubung, *stopword* juga terdiri dari beberapa kata keterangan atau yang lainnya. Contoh dari *stopword* list dalam bahasa indonesia antara lain.

1. Kata penghubung : sesudah, selesai, sebelum
2. Kata tugas : bagi, dari, dengan, pada
3. Kata keterangan : sangat, hanya, lebih
4. Kata bilangan : beberapa, banyak, sedikit
5. Kata ganti : kami, mereka , kita,
6. Dan masih banyak lainnya.

Berikut adalah tabel hasil dari proses *Filtering*

Tabel 3. 3 Hasil proses Filtering

No	**Kata awal**			**Kata hasil *filtering***		
1	bahan	atau	zat	bahan	zat	dapat
	yang	dapat	menghantarkan	menghantarkan	aliran	listrik
	aliran	listrik	dengan	baik		
	baik					

2	bahan	yang	mampu		bahan	mampu	mengalirkan	
	mengalirkan	listrik	dengan		listrik	baik		
	baik							
3	zat	yang	dapat		zat	dapat	mengalirkan	
	mengalirkan	arus	listrik		arus	listrik		
4	bahan	yang	dapat		bahan	dapat	menghantarkan	
	menghantarkan	arus	listrik		arus	listrik		

Keterangan :

= kata atau term yang akan dihilangkan dalam proses *filtering*

3.2.1.3 *Stemming*

Stemming diperlukan untuk mengurangi jumlah penyimpanan *term* pada *database*, karena banyak kata dasar yang jika ditambahkan imbuhan akan menjadi kata yang berbeda. Selain itu juga untuk memudahkan dalam pencocokan antar *term* yang akan dilakukan pada proses berikutnya. Contoh *term* "hantar" dalam kata "menghantarkan" merupakan dua *term* yang berbeda. Jika tidak dilakukan tahap *stemming* maka pencocokan kata tersebut akan memiliki nilai yang berbeda. Contoh hasil proses *stemming* pada *term* yang terdapat pada kunci jawaban dapat dilihat pada tabel 3.4.

Tabel 3. 4 Contoh hasil stemming pada kunci jawaban

No	**Kata awal**				**Kata hasil *filtering***			
1	bahan	zat	dapat		bahan	zat	dapat	
	menghantarkan	aliran	listrik		hantar	alir	listrik	
	baik				baik			
2	bahan	mampu	mengalirkan		bahan	mampu	alir	
	listrik	baik			listrik	baik		

3	zat	dapat	mengalirkan	zat	dapat	alir
	arus	listrik		arus	listrik	
4	bahan	dapat	menghantarkan	bahan	dapat	hantar
	arus	listrik		arus	listrik	

Keterangan

huruf = kata imbuhan yang mengalami proses *stemming* dan dihilangkan

3.2.2 Analisis metode LSA

Tahapan yang terdapat dalam proses LSA terdiri dari empat tahap dimana masing masing tahap ada proses perhitungan matematisnya. Proses ini adalah lanjutan proses *preprocessing* yang telah dilakukan sebelumnya untuk mendapatkan nilai kemiripan antara jawaban dan kunci jawaban.

3.2.2.1 Pembentukan matriks

Tahapan pembuatan matriks adalah tahapan awal dalam proses LSA yang mana merupakan tahapan lanjutan dari *preprocessing* terakhir yang terdapat pada tabel 3.4. Sebelum membuat matriks hal yang harus diperhatikan adalah mencari *queries* yang merupakan elemen *row* pada matriks yang akan dibangun. Caranya adalah dengan menggabungkan *array (term-term)* dari semua jawaban dan kunci jawaban yang ada.

Tabel 3. 5 Pembentukan nilai Queries

Kumpulan term dari dokumen	Kumpulan term yang sudah disorting	Queries
bahan zat dapat hantar alis listrik baik bahan mampu alir listrik baik zat dapat alir arus listrik	alir alir arus bahan bahan baik baik dapat dapat hantar hantar listrik	alir arus bahan baik dapat hantar listrik mampu zat

bahan dapat hantar arus listrik	listrik listrik mampu zat zat	

Sehingga didapatkan *term* dari setiap *query* jawaban dan kunci jawaban serta *queries* yang dapat dilihat pada tabel 3.6.

Tabel 3. 6 Tabel term yang mewakili

Dokumen	Term yang mewakili dokumen
Q	bahan zat dapat hantar alir listrik baik
D1	bahan mampu alir listrik baik
D2	zat dapat alir arus listrik
D3	bahan dapat hantar arus listrik
Queries	alir arus bahan baik dapat hantar listrik mampu zat

Setelah kita dapatkan *term* yang mewakili dokumen serta *term queries* langkah selanjutnya adalah dengan membentuk matriks dengan ukuran *m x n* dimana *m* merupakan jumlah *term* dari *queries* dan *n* adalah jumlah *query* kunci dan jawaban. Ketika kerangka dari matriks sudah didapatkan selanjutnya adalah mengisi matriks tersebut dengan nilai frequensi kesamaan antara *term* pada masing masing *query* dengan *term* pada *queries.* Sebagai contoh pada matriks (1,1) dalam tabel 3.6 adalah pencocokan *term* "alir" yang ada pada *query* kunci jawaban dengan *queries* sehingga didapatkan nilai 1 karena pada *query* kunci jawaban hanya terdapat 1 *term* "alir". Proses pencocokan antara *term* pada *query* dan *term* pada *queries* dilanjutkan sampai seluruh *term* pada *query* sudah dilakukan pemeriksaan kesamaan. Hasil dari pembentukan matriks dapat dilihat pada tabel 3.7 berikut.

Tabel 3. 7 Tabel Term Frequensi

Term queries	Q	D1	D2	D3
Alir	1	1	1	0
Arus	0	0	1	1
Bahan	1	1	0	1
Baik	1	1	0	0
Dapat	1	0	1	1
Hantar	1	0	0	1
Listrik	1	1	1	1
Mampu	0	1	0	0
Zat	1	0	1	0

3.2.2.2 Perhitungan nilai matriks U, S, V^T

Setelah terbentuk sebuah matriks berukuran *m* x *n* maka langkah selanjutnya adalah dengan mencari nilai matriks U, S, V^T dengan metode SVD (*Singular Value Decomposition*). Untuk mendapatkan matriks-matriks berikut melalui beberapa perhitungan yang cukup kompleks mulai dari melakukan transpose matriks, mencari nilai eigen dan mencari vektor eigennya. Namun dalam implementasi program, proses perhitungan dapat menggunakan library dari JAMA yang telah menyediakan fungsi untuk melakukan perhitungan matriks sebagai pengganti MATLAB sehingga didapatkan matriks U, S, V^T.

Dari tabel 3.6 kita dapatkan sebuah matriks A dengan dimensi 9x4 yang akan dilakukan proses SVD untuk mencari nilai U, S, V^T . langkah pertama untuk mencari matriks tersebut adalah dengan mencari matriks A^T terlebih dahulu.

$$A \quad \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

$$A^T \quad \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Langkah berikutnya adalah dengan mencari nilai $A*A^T$ dan A^T*A sehingga didapatkan matriks berikut:

$$A^T* A \quad \begin{bmatrix} 7 & 4 & 4 & 4 \\ 4 & 5 & 2 & 2 \\ 4 & 2 & 5 & 3 \\ 4 & 2 & 3 & 5 \end{bmatrix}$$

$$A*A^T \quad \begin{bmatrix} 3 & 1 & 2 & 2 & 2 & 1 & 3 & 1 & 2 \\ 1 & 2 & 1 & 0 & 2 & 1 & 2 & 0 & 1 \\ 2 & 1 & 3 & 2 & 2 & 2 & 3 & 1 & 1 \\ 2 & 0 & 2 & 2 & 1 & 1 & 2 & 1 & 1 \\ 2 & 2 & 2 & 1 & 3 & 2 & 3 & 0 & 2 \\ 1 & 1 & 2 & 1 & 2 & 2 & 2 & 0 & 1 \\ 3 & 2 & 3 & 2 & 3 & 2 & 4 & 1 & 2 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 & 2 & 1 & 2 & 0 & 2 \end{bmatrix}$$

Setelah didapatkan hasil perkalian antara matriks A dan transpose dari matriks A diatas maka langkah selanjutnya adalah mencari nilai eigen dengan cara mengurangi matriks tersebut dengan *lambda* dikalikan dengan matriks identitas. Yang pertama adalah mencari nilai eigen dari matriks A^T*A.

Nilai eigen = $(A^T * A) - \lambda I$

$$\begin{bmatrix} 7 & 4 & 4 & 4 \\ 4 & 5 & 2 & 2 \\ 4 & 2 & 5 & 3 \\ 4 & 2 & 3 & 5 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix} = \begin{bmatrix} 7 - \lambda & 4 & 4 & 4 \\ 4 & 5 - \lambda & 2 & 2 \\ 4 & 2 & 5 - \lambda & 3 \\ 4 & 2 & 3 & 5 - \lambda \end{bmatrix}$$

Dari perhitungan diatas dilanjutkan dengan mencari persamaan polynomial (banyak suku) dari matriks diatas. Caranya adalah dengan mengalikan setiap elemen dikurangi elemen lain dari sisi sebelahnya. Sehingga didapakan persamaan berikut :

$$\lambda^4 - 21\lambda^3 + 300\,\lambda^2 - 858\,\lambda + 1375 = 0$$

Setelah didapatkan sebuah persamaan *polynomial* seperti diatas langkah selanjutnya adalah mencari akar dari persamaan tersebut sehingga didapatkan

λ_1 = 15,400 λ_2 = 1,235

λ_3 = 3.366 λ_4 = 2,000

Sedangkan untuk matriks A*A^T dengan melakukan proses yang sama didapatkan nilai eigen sebagai berikut.

λ_1 = 15,400 λ_2 = 3,366

λ_3 = 2,000 λ_4 = 1,235 $\lambda_5, \lambda_6, \lambda_7, \lambda_8, \lambda_9$ = 0,000

Setelah didapatkan nilai eigen dari masing masing persamaan diatas langkah selanjutnya adalah dengan memasukan nilai λ pada persamaan awal yang kemudian dinormalisai sehingga didapatkan matriks matriks berikut. Untuk vector eigen dari matriks A^T* A akan menjadi nilai V yang kemudian akan di transpose menjadi V^T. Dan vektor nilai eigen dari A* A^T akan menjadi nilai matriks U. Sedangkan matriks S didapatkan dari diagonal akar nilai eigen yang sudah diurutkan dari yang terbesar dan bukan nol.

Matriks U

$$\begin{bmatrix} 0.386 & -0.248 & -0.5 & -0.025 \\ 0.233 & 0.483 & 0 & -0.533 \\ 0.386 & -0.248 & 0.5 & -0.025 \\ 0.269 & -0.489 & 0 & 0.251 \\ 0.395 & 0.413 & 0 & 0.131 \\ 0.279 & 0.171 & 0.5 & 0.408 \\ 0.502 & -0.006 & 0 & -0.302 \\ 0.107 & -0.419 & 0 & -0.433 \\ 0.279 & 0.171 & -0.5 & 0.408 \end{bmatrix}$$

Matriks S

$$\begin{bmatrix} 3.924 & 0 & 0 & 0 \\ 0 & 1.835 & 0 & 0 \\ 0 & 0 & 1.414 & 0 \\ 0 & 0 & 0 & 1.111 \end{bmatrix}$$

Matriks V^T

$$\begin{bmatrix} 0.636 & 0.421 & 0.457 & 0.457 \\ 0.129 & 0.769 & -0.443 & -0.443 \\ 0 & 0 & -0.707 & -0.707 \\ 0.761 & -0.482 & -0.307 & -0.307 \end{bmatrix}$$

Setelah didapatkan ketiga matriks tersebut langkah selanjutnya adalah dengan melakukan penyederhanaan kolom-kolom matriks. Penyederhanaan dilakukan pada matriks U dengan penyederhanaan pada jumlah kolomnya, matriks S pada baris dan kolomnya. Langkah penyederhanaan ini dilakukan sesuai dengan nilai dimensi dari matriks tersebut. sehingga didapatkan matriks U dan S dengan dimensi baru sebagai berikut.

Matriks U

0.386	-0.248
0.233	0.483
0.386	-0.248
0.269	-0.489
0.395	0.413
0.279	0.171
0.502	-0.006
0.107	-0.419
0.279	0.171

Matriks S

3.924	0
0	1.835

3.2.2.3 Perhitungan vektor kunci jawaban

Pencarian matriks vektor Q dan tiap-tiap D dilakukan untuk melakukan proses perhitungan *cosine similarity* di tahap berikutnya. Pada langkahnya tahap ini melakukan perkalian matriks dengan himpunan *term* frekuensi masing masing *query* yang di*transpos*e terhadap matriks U yang kemudian dikalikan kembali dengan invers dari matriks S (S^{-1}). perhitungan berikut adalah contoh dari perhitungan vektor kunci jawaban (Q).

$$\bar{q} = Q^T \,.\, U \,.\, S^{-1} \qquad \text{. . . . . } \textbf{persamaan (3.1)}$$

$$\bar{q} = \quad | 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 1 | \begin{vmatrix} 0.386 & -0.248 \\ 0.233 & 0.483 \\ 0.389 & -0.248 \\ 0.269 & -0.489 \\ 0.395 & 0.413 \\ 0.279 & 0.171 \\ 0.502 & -0.006 \\ 0.107 & -0.419 \\ 0.279 & 0.171 \end{vmatrix} \begin{vmatrix} 0.255 & 0.00 \\ 0.000 & 0.545 \end{vmatrix}$$

$\bar{q}$ = | 0.637 -0.129 |

Jadi hasil perhitungan matriks diatas didapatkan sebuah vektor dari *query* jawaban (Q) adalah $\bar{q}$ = | 0.617 -0.135 | . dengan melakukan perhitungan yang sama maka masing masing jawaban akan didapatkan nilai vektor sebagai berikut :

Nilai vector Q = | 0.637 -0.129 |
Nilai vector D1 = | 0.407 -0.813 |
Nilai vector D2 = | 0.444 0.470 |
Nilai vector D3 = | 0.445 0.470 |

3.2.2.4 Perhitungan nilai cosine similarity

Perhitungan nilai *cosine similarity* dilakukan guna mendapatkan nilai atau tingkat kemiripan antar dua buah vektor. Dalam hal ini vektor yang akan dilakukan perhitungan adalah vektor antara kunci jawaban dan masing masing jawaban siswa. Perhitungan ini dilakukan dengan menggunakan rumus pada persamaan 2.6 .

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

$$D1 = \frac{(Q_1 . D_{11}) + (Q_2 . D_{12})}{\sqrt{(Q_1)^2 + (D_{11})^2}\ x\ \sqrt{(Q_2)^2 + (D_{12})^2}}$$

$$\text{D1} = \frac{(0{,}617)(0{,}407)+(-0.135)(-0.813)}{\sqrt{(0{,}617)^2+(0{,}407)^2}\ \sqrt{(-0.135)^2+(-0.813)^2}} = 0.644$$

Sehingga didapatkan nilai *cosine similarity* dari D1 sebesar 0.644 . Untuk nilai yang lain juga dilakukan dengan proses yang sama sampai didapatkan masing masing nilai *cosine similarity* sebagai berikut :

$$\text{D2} = \frac{(0{,}617)(0{,}444)+(-0.135)(0{,}470)}{\sqrt{(0{,}617)^2+(0{,}444)^2}\ \sqrt{(-0.135)^2+(0{,}470)^2}} = 0.566$$

$$\text{D3} = \frac{(0{,}617)(0{,}445)+(-0.135)(0.470)}{\sqrt{(0{,}617)^2+(0{,}445)^2}\ \sqrt{(-0.135)^2+(0.470)^2}} = 0.566$$

Setelah didapatkan nilai *similarity* dari masing masing jawaban maka tahap selanjutnya adalah perkalian dengan bobot dari setiap soal yang diujikan. Dalam soal ini memiliki bobot 5 dari 10 point yang ada dalam *quiz* dari dua soal, sehingga ketika jawaban dari siswa memiliki kesamaan sempurna dengan kunci yang diberikan akan memiliki 5 point pada soal tersebut. Jadi nilai rekomendasi yang didapatkan oleh guru dari fungsionalitas koreksi soal pada contoh soal model ini adalah sebagai berikut :

Nilai rekomendasi = nilai *cosine similarity* * bobot tiap soal

D1 = 0.644 * 5 = 3.2

D2 = 0.566 * 5 = 2.62

D3 = 0.566 * 5 = 2.62

3.3 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan *non-fungsional* merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan dari suatu sistem. Spesifikasi ini juga termasuk elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai sistem dapat diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan oleh sistem, keluaran yang akan dihasilkan oleh sistem dan proses yang dibutuhkan untuk mengolah masukan hingga menghasilkan keluaran yang diinginkan.

3.3.1 Analisis Perangkat Lunak (*Software*)

Analisis perangkat lunak adalah penjelasan tentang kebutuhan-kebutuhan yang diperlukan untuk dapat menjalankan aplikasi koreksi soal esai otomatis ini. Adapun kebutuhan perangkat lunak merupakan faktor yang cukup penting dan sangat dibutuhkan untuk menunjang pembangunan aplikasi ini sendiri maka dari itu harus dicocokan antara fakta dan kebutuhan dari system yang akan dikembangkan.

3.3.1.1 Fakta Lapangan

Perangkat lunak (*software*) yang digunakan di lingkungan SMK N 1 Bulakamba dapat dilihat pada tabel berikut

Tabel 3. 8 Analisis spesifikasi perangkat lunak di lapangan

Sistem operasi	Windows XP
Web browser	*Internet Explorer, Google Chrome, Mozila Firefox*

3.3.1.2 Kebutuhan Sistem

Kebutuhan perangkat lunak (*software*) untuk menjalankan aplikasi ini dapat dilihat pada tabel berikut.

Tabel 3. 9 Analisis spesifikasi perangkat lunak untuk kebutuhan sistem

Sistem operasi	Windows XP
Web browser	*Internet Explorer, Google Chrome, Mozila Firefox*

Dari analisis perangkat lunak yang dilakukan di lingkungan SMK N 1 Bulakamba dan kebutuhan yang diperlukan maka perangkat lunak yang ada di lingkungan SMK N 1 Bulakamba telah mencukupi.

3.3.2 Analisis Perangkat Keras (Hardware)

Perangkat keras merupakan faktor penunjang agar aplikasi yang akan dibangun berjalan dengan baik. Untuk itu dilakukan peninjauan secara langsung terhadap spesifikasi perangkat keras yang ada di lingkungan SMK N 1 Bulakamba. Berikut adalah spesifikasi perangkat keras yang telah digunakan oleh pihak sekolah.

3.3.2.1 Fakta Lapangan

Perangkat keras (*hardware*) komputer yang ada di SMK N 1 Bulakamba spesifikasinya terdapat pada tabel berikut.

Tabel 3. 10 Analisis spesifikasi perangkat keras di lapangan

Perangkat	Spesifikasi
Processor	Intel Dual core 1,8Ghz
RAM (memory)	1024Mb
Space Hardisk	512 Gb
VGA	128 Gb
LAN Card	10/100 Mbps

Koneksi internet	128 kbps

3.3.2.2 Kebutuhan Sistem

Kebutuhan perangkat keras (*hardware*) untuk penggunaan aplikasi yang akan dibangun dapat dilihat pada tabel berikut.

Tabel 3. 11 Analisis kebutuhan perangkat keras untuk kebutuhan sistem

Perangkat	Spesifikasi
Processor	Intel/AMD Pentium 4 1,6 Ghz
RAM (memory)	512 Mb
Space Hardisk	128 Gb
VGA	128 Gb
LAN Card	10/100 Mbps
Koneksi internet	Minimal 64 kbps

Dari hasil pengamatan perangkat keras yang ada di lingkungan SMK N 1 Bulakamba sudah dapat memenuhi standar kebutuhan minimum spesifikasi perangkat keras yang dibutuhkan untuk menjalankan aplikasi ini.

3.3.3 Analisis Pengguna (*User*) sistem

Analisis pengguna merincikan siapa saja pengguna dari sistem yang berjalan pada saat ini yang nantinya akan di spesifikasikan siapa saja yang masuk kedalam sistem yang akan dibangun.

3.3.3.1 Fakta Lapangan

Gambaran umum pengguna yang sekarang ada di SMK N 1 Bulakamba dapat dilihat pada tabel berikut :

Tabel 3. 12 Analisis pengguna di lapangan

No	Pengguna	karakteristik

1	Guru	Umur	25 s/d 50 tahun
		Pendidikan	D4 – S2
		Ketrampilan	Bisa mengoperasikan komputer dengan sistem operasi Microsoft Windows XP
		Pengalaman	Dapat menggunakan browser serta menguasai dasar dasar internet
		Tanggung jawab	Memberikan ujian , mengisi data nilai dan memberikan pengumuman pada siswa

3.3.3.2 Kebutuhan Sistem

Agar memaksimalkan penggunaan sistem yang akan dibangun, maka dijelaskan pula spesifikasi dan hak akses bagi setiap user yang akan menggunakan sistem ini. Pengguna aplikasi akan dibagi menjadi tiga kategori yaitu admin, guru, dan siswa. Adapun kebutuhan pengetahuan bagi setiap pengguna dijelaskan lebih detail dalam tabel berikut :

Tabel 3. 13 Analisis kebutuhan sistem pengguna

No	Pengguna	karakteristik	
1	Guru	Hak akses	Melakukan proses koreksi soal *quiz* esai yang sudah dilakukan oleh siswa
		Pendidikan	D4 – S2
		Ketrampilan	Bisa mengoperasikan komputer dengan sistem

			operasi Microsoft Windows XP
		Tanggung jawab	Memberikan ujian , mengisi data nilai dan memberikan pengumuman pada siswa

Setelah dilakukan analisis terhadap seluruh pengguna sistem LMS di SMK N 1 Bulakamba maka dapat disimpulkan bahwa karakteristik pengguna yang ada di lingkungan SMK N 1 Bulakamba memenuhi persyaratan untuk menjalankan aplikasi yang akan dibangun.

3.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan analisis untuk menggambarkan sistem yang akan dibangun . Analisis kebutuhan fungsional dimodelkan dengan menggunakan UML, tahap-tahap pemodelannya antara lain adalah *use case* diagram, *use case scenario, activity diagram, class diagram* dan *sequence diagram*.

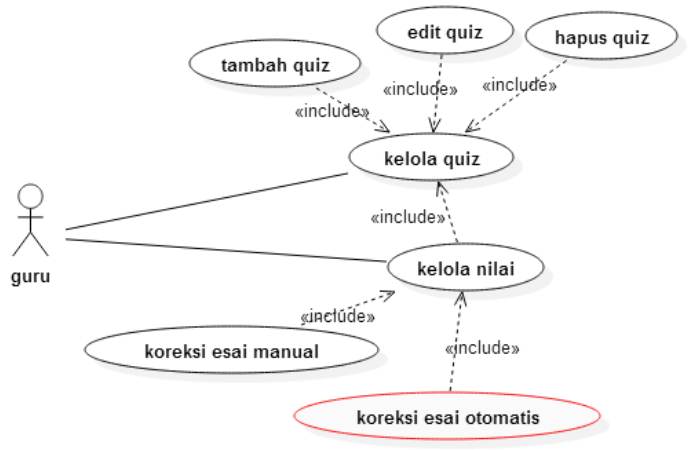
3.4.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) suatu sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Berikut adalah *use case* diagram LMS SMK Negeri 1 bulakamba yang sudah ada.



Gambar 3. 7 Use Case LMS SMK N 1 Bulakamba

Dan berikut adalah *use case* pada bagian yang akan dikembangkan dalam LMS SMK N 1 Bulakamba. Yang akan dikembangkan adalah fungsionalitas koreksi (penilaian) *quiz* esai otomatis.



Gambar 3. 8 Use case pengembangan koreksi esai otomatis

Keterangan :

 = adalah *use case* yang akan dikembangkan

3.4.1.1 Definisi Aktor

Definisi aktor berfungsi untuk menjelaskan aktor yang terdapat pada *use case* diagram. Deskripsi pendefinisian aktor pada pengembangan LMS SMK N 1 Bulakamba dapat dilihat pada tabel berikut

Tabel 3. 14 Definisi aktor

No	Aktor	Deskripsi
1	Guru	Orang yang melakukan koreksi ujian esai secara manual dan otomatis.

3.4.1.2 Definisi *Use Case*

Definisi *use case* berfungsi untuk menjelaskan proses yang terdapat pada *use case* diagram. Deskripsi pendefinisian *use case* pengembangan pada LMS SMK Negeri 1 Bulakamba dapat dilihat pada tabel 3.14:

Tabel 3. 15 Definisi Use Case

No	Use Case	Deskripsi
1	koreksi esai otomatis	Proses koreksi hasil *quiz* esai yang dijalankan oleh guru untuk mendapatkan nilai pertimbangan dalam memberikan penilaian pada siswa

3.4.1.3 *Use Case Scenario*

Use case scenario merupakan deskripsi alur kejadian yang menjelaskan urutan interaksi antara aktor dengan *use case* tersebut. Berdasrkan *use case diagram* maka *use case scenario* untuk koreksi esai otomatis ini adalah sebagai berikut. *Use case scenario* koreksi esai otomatis menjelaskan tentang proses koreksi esai otomatis sehingga bisa didapatkan nilai rekomendasi untuk penilaian guru.

Tabel 3. 16 Use Case Scenario Koreksi Esai Otomatis

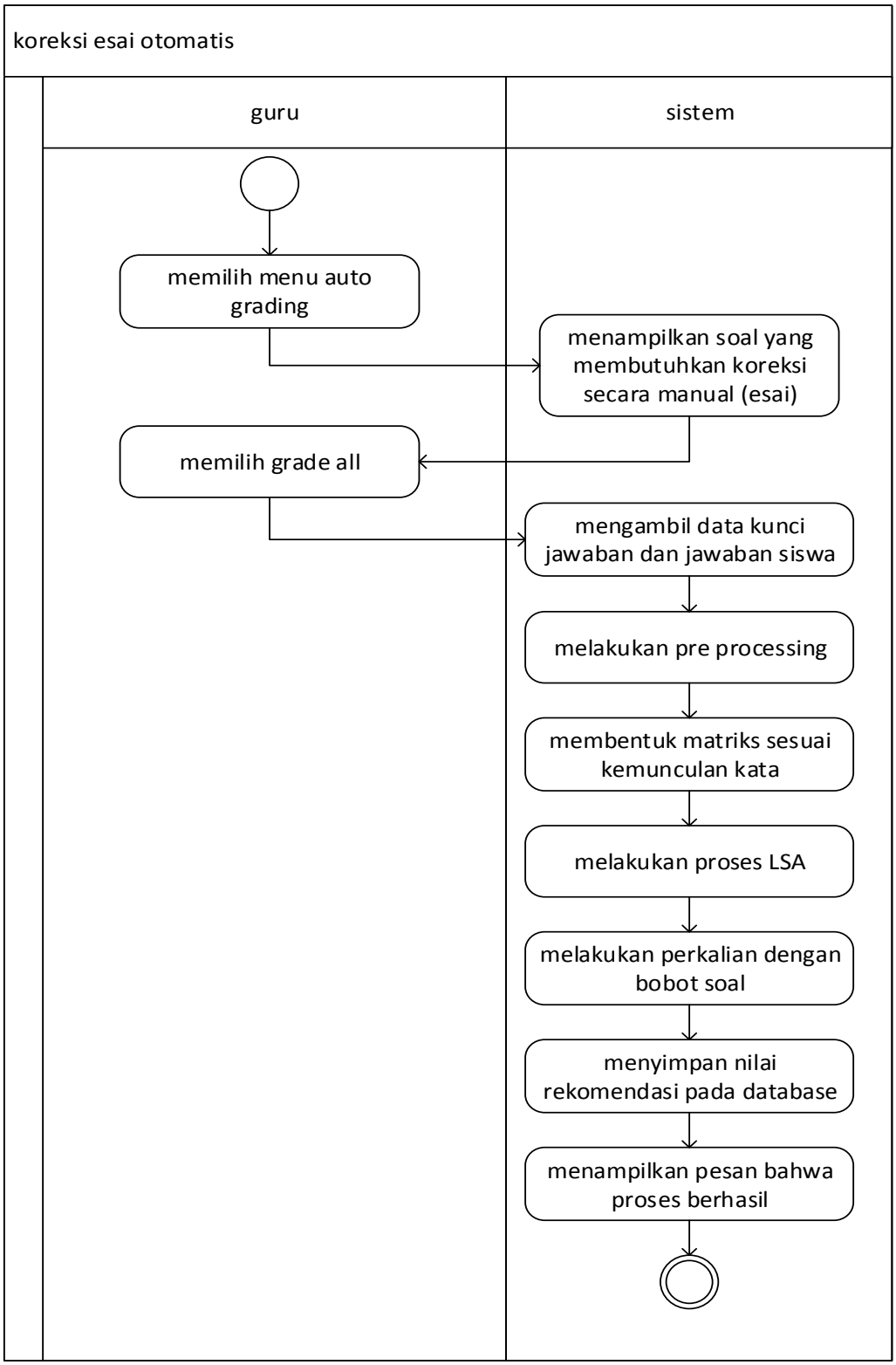
Identifikasi	
Nama	Koreksi esai otomatis
Tujuan	Mendapatkan nilai dari *quiz* esai yang dilakukan oleh siswa
Deskripsi	Proses yang dilakukan oleh guru untuk mendapatkan nilai *quiz* esai siswa.
Aktor	guru
Skenario Utama	
Kondisi awal	Tampil halaman *quiz*
Skenario Normal	
Aksi Aktor	Reaksi Sistem
a. Memilih *auto grading*	

	b. Menampilkan pertanyaan yang membutuhkan koreksi secara manual (esai)
c. Memilih *grade all*	
	d. Mengambil data kunci jawaban dan jawaban siswa *e.* Melakukan proses *pre processing (case folding, tokenizing, filtering, stemming)* sampai didapatkan kata dasar dari masing masing jawaban dan kunci jawaban f. Membentuk matriks sesuai frekuensi kemunculan kata dari jawaban dan kunci jawaban yang sudah dalam g. Melakukan proses perhitungan SVD untuk mencari nilai matriks U,S dan V h. Mencari nilai vektor tiap tiap jawaban dan kunci jawaban i. Mencari nilai *cosine similarity* tiap-tiap jawaban j. Melakukan perkalian dengan nilai bobot setiap soal k. Menyimpan nilai rekomendasi pada database

	l. Menampilkan pesan bahwa proses telah berhasil
Kondisi Akhir	Sistem menghasilkan nilai rekomendasi dari ujian yang telah dilakukan oleh siswa. Dan melakukan penyimpanan nilai tersebut dalam *database*.

3.4.2 Activity Diagram

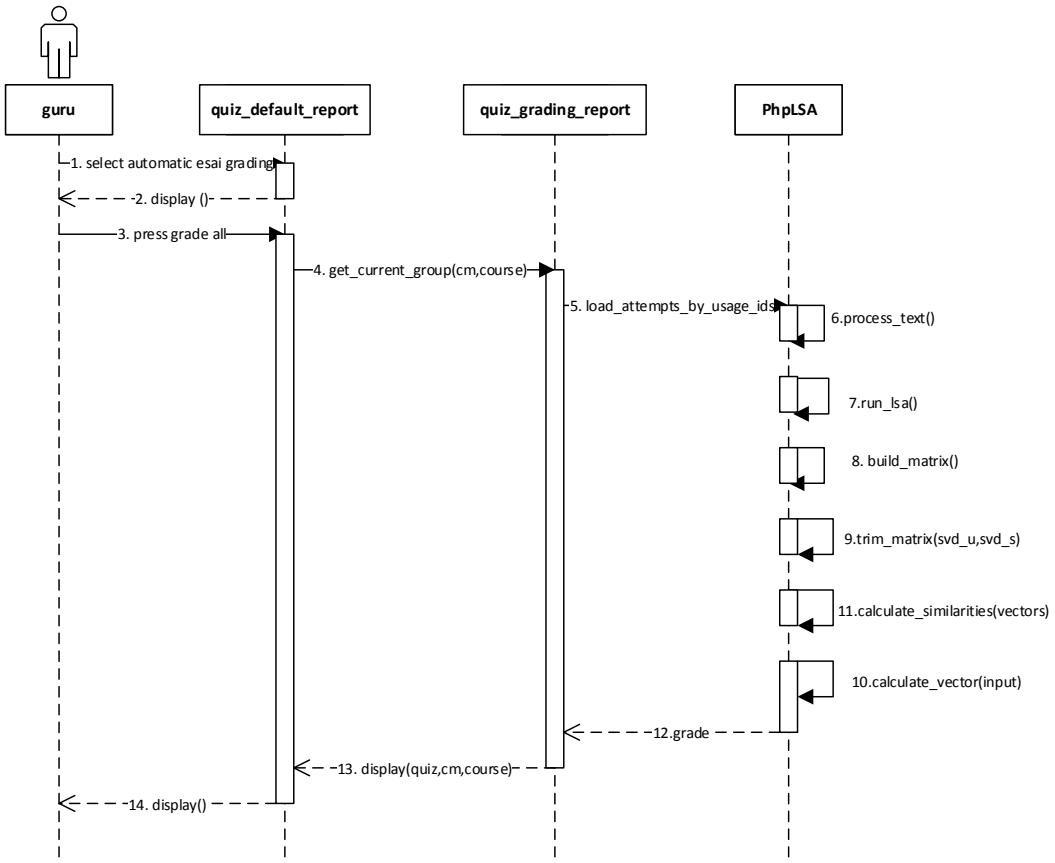
Diagram aktivitas atau *activity diagram* merupakan sebuah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah *activity diagram* pada koreksi esai otomatis dalam pengembangan LMS SMK Negeri 1 Bulakamba :



Gambar 3. 9Activity diagram koreksi esai otomatis

3.4.3 Sequence Diagram

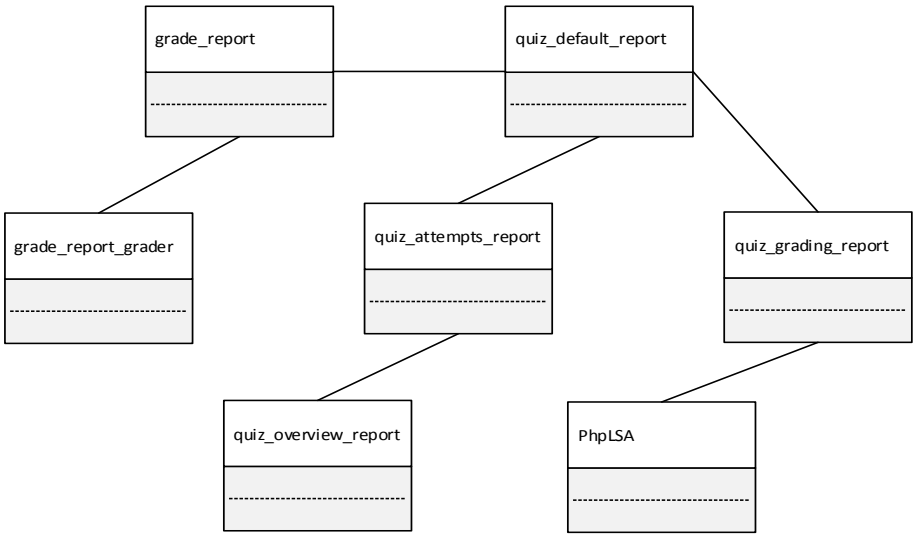
Sequence diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan diagram *use case*. *Sequence diagram* memperlihatkan tahap demi tahap apa yang sebenarnya terjadi untuk menghasilkan sesuatu didalam *use case*. Berikut adalah *sequence diagram*yang terdapat dalam pengembangan LMS SMK N 1 Bulakamba :



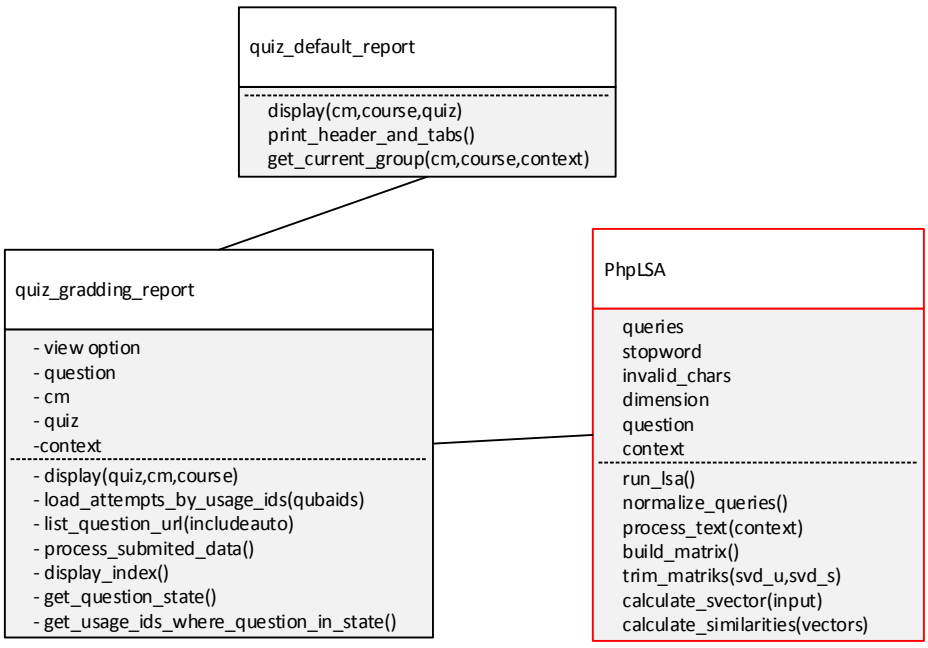
Gambar 3. 10 Sequence diagram koreksi esai otomatis

3.4.4 Class Diagram

Class diagram memperlihatkan hubungan antar *class* dan penjelasan detail tiap-tiap *class* didalam model desain dari suatu system. *Class* diagram pada pengembangan LMS SMK N 1 Bulakamba akan dijelaskan dalam gambar berikut:



Gambar 3. 11 Class diagram grade pada moodle



Gambar 3. 12 Class diagram pengembangan koreksi esai otomatis

Pengembangan koreksi esai otomatis pada LMS yang sudah berjalan hanya menambahkan satu *class* yaitu *class* PhpLSA. Pada tabel 3.17 adalah keterangan dari *class* yang akan ditambahkan dalam sistem.

Tabel 3. 17 Keterangan *class diagram* PhpLSA

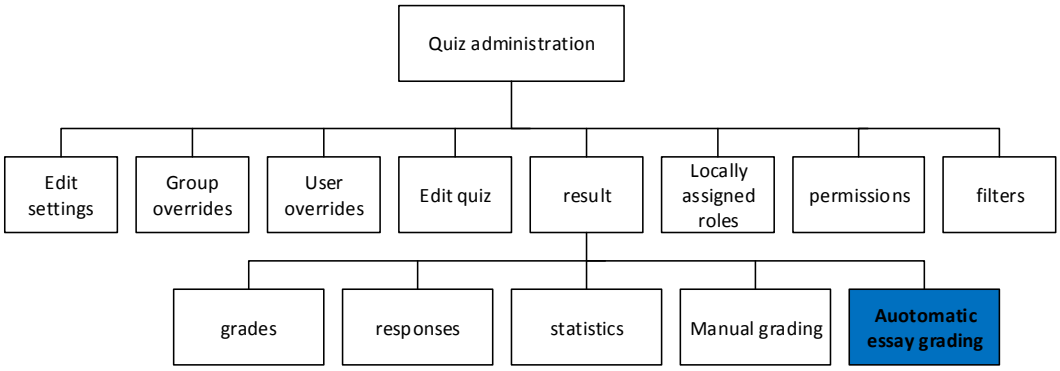
Nama Class	PhpLSA	
Jenis	Nama	Keterangan
Atribut	queries	Text
	stopwords	Text
	invalid_chars	Text
	dimension	Number
	question	Text
	context	Text
Methode	run_lsa( )	Methode yang digunakan untuk menjalankan proses LSA secara keseluruhan
	process_text( )	Method yang digunakan untuk melakukan preprocessing
	build_matrix( )	Method yang digunakan untuk membuat matriks
	trim_matriks( )	Method yang digunakan untuk melakukan proses perhitungan matriks S,U,V
	calculate_vector( )	Methode yang dugunakan untuk melakukan proses perhitungan vektor
	calculate_similarities()	Methode yang digunakan untuk mencari perhitungan nilai kemiripan (cosine similarity)

3.5 Perancangan Sistem

Perancangan sistem merupakan salah satu langkah dalam proses pengembangan suatu perangkat lunak yang dilakukan setelah tahapan analisis. Berikut adalah perancangan sistem pengembangan LMS SMK N 1 Bulakamba :

3.5.1 Perancangan Struktur Menu

Perancangan struktur menu dilakukan guna memberikan gambaran fungsi-fungsi yang akan digunakan dalam pengembangan LMS SMK N 1 Bulakamba. Dan berikut adalah struktur menu *quiz administration* guru yang akan dikembangkan.



Gambar 3. 13 Struktur menu guru

Keterangan :

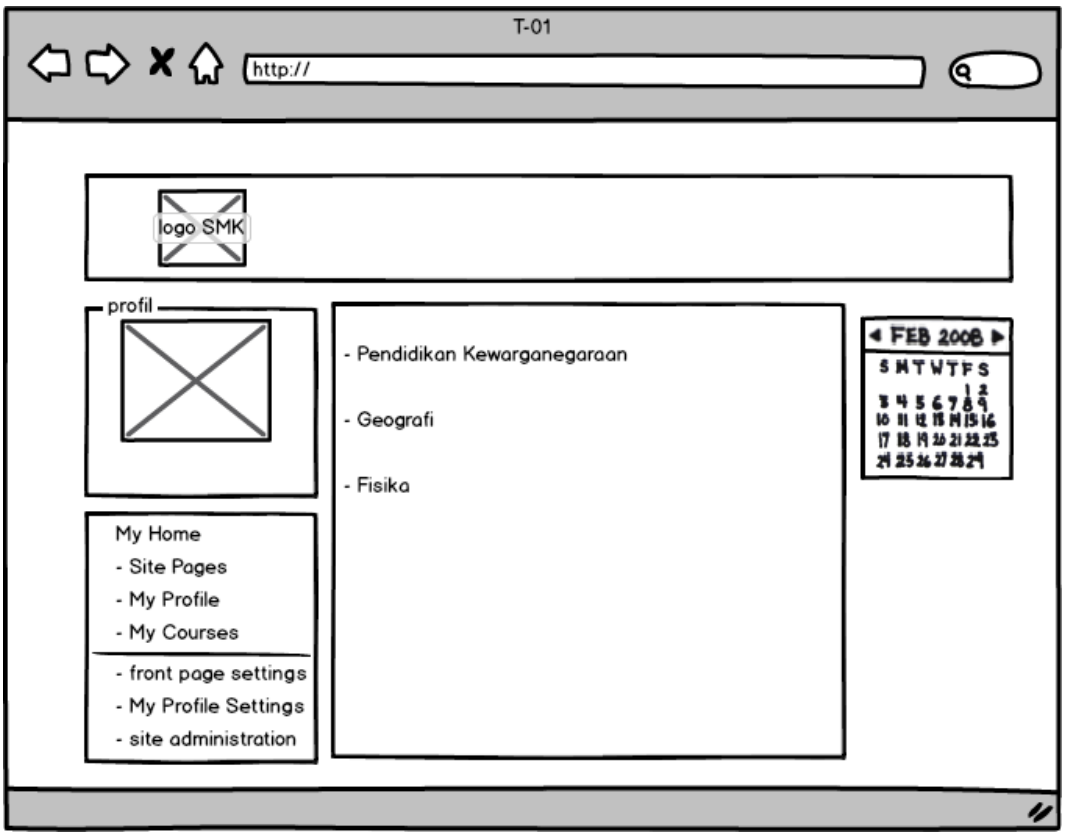
= Penambahan fungsionalitas

3.5.2 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dikembangkan sehingga akan mempermudah dalam pembangunannya.

1. Tampilan halaman awal guru

   Tampilan ini adalah halaman awal ketika guru berhasil login sesuai dengan akunnya. Menampilkan mata pelajaran / kursus yang tersedia dalam sistem.



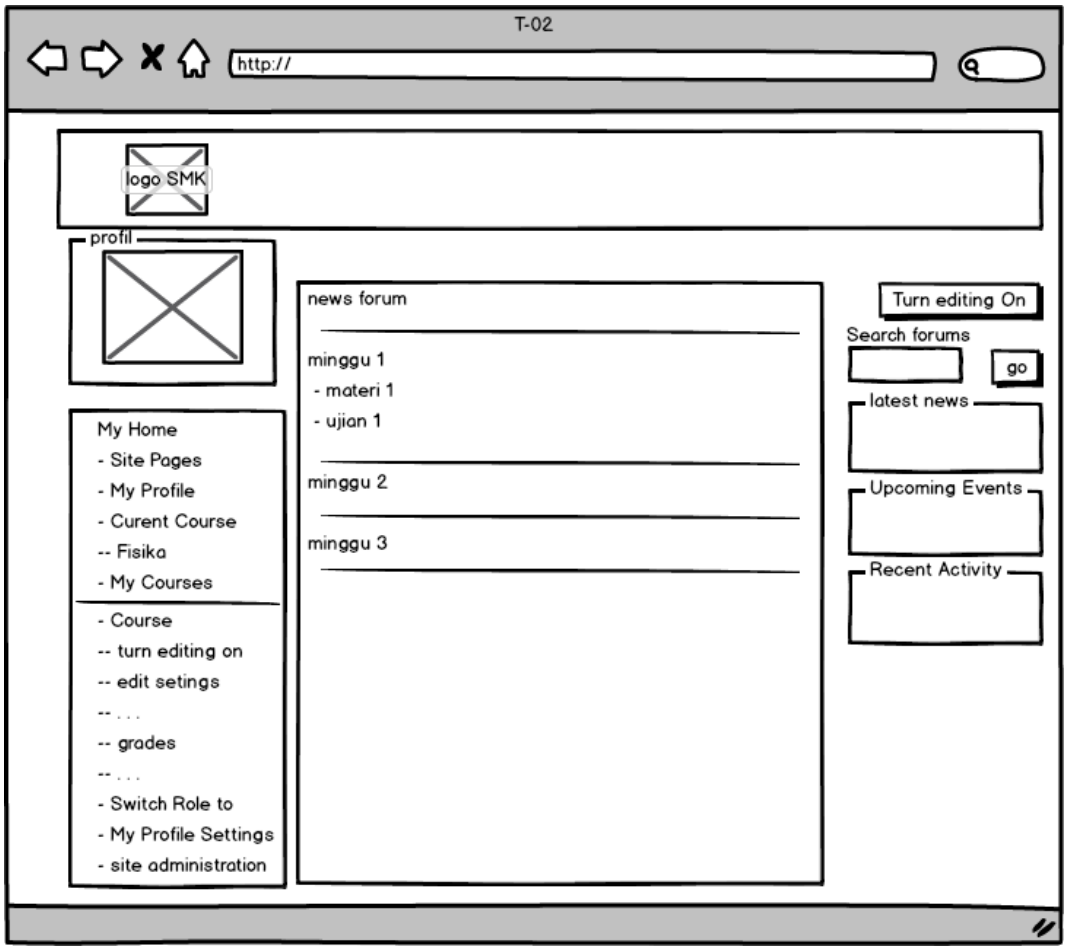
Gambar 3. 14 Tampilan halaman awal guru

Keterangan :

- Klik fisika menuju ke tampilan T-02
- Klik *my home* tetap berada di tampilan T-01
- Ukuran layar = 1366 x 768
- Huruf *times new roman*
- Warna = putih, orange

2. Tampilan halaman pilih pelajaran

   Tapilan ini adalah tampilan setelah user memilih salah satu pelajaran yang tersedia dan yang dia kelola. Dalam tampilan ini menampilkan aktivitas perminggu.



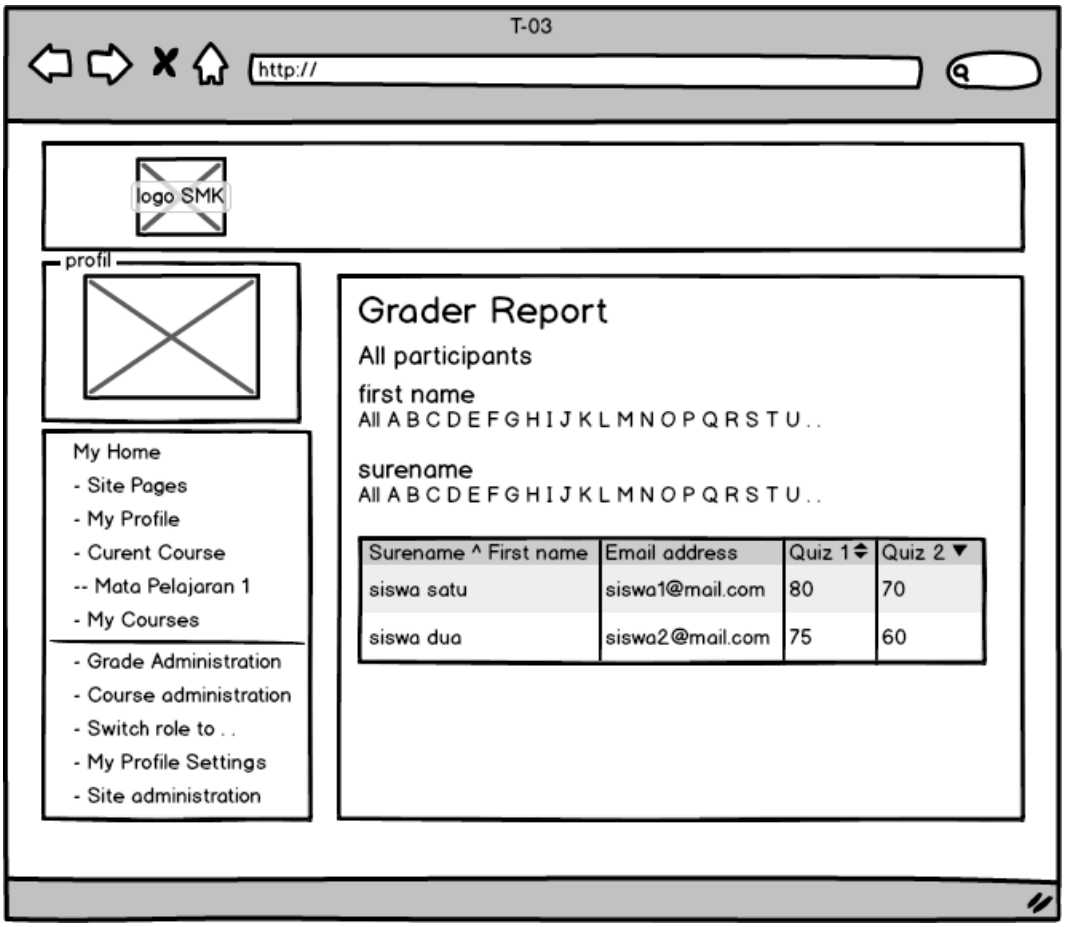
Gambar 3. 15 Tampilan halaman pilih pelajaran

Keterangan :

- Klik *my home* kembali ke tampilan T-01
- Klik *grades* menuju ke tampilan T-03
- Ukuran layar = 1366 x 768
- Huruf *times new roman*
- Warna = putih, orange

3. Tampilan halaman pilih nilai (*grade)*

   Tampilan ini adalah tampilan tentang nilai yang ada pada satu pelajaran tertentu dan mencakup seluruh siswa yang mengambil mata pelajaran tersebut.



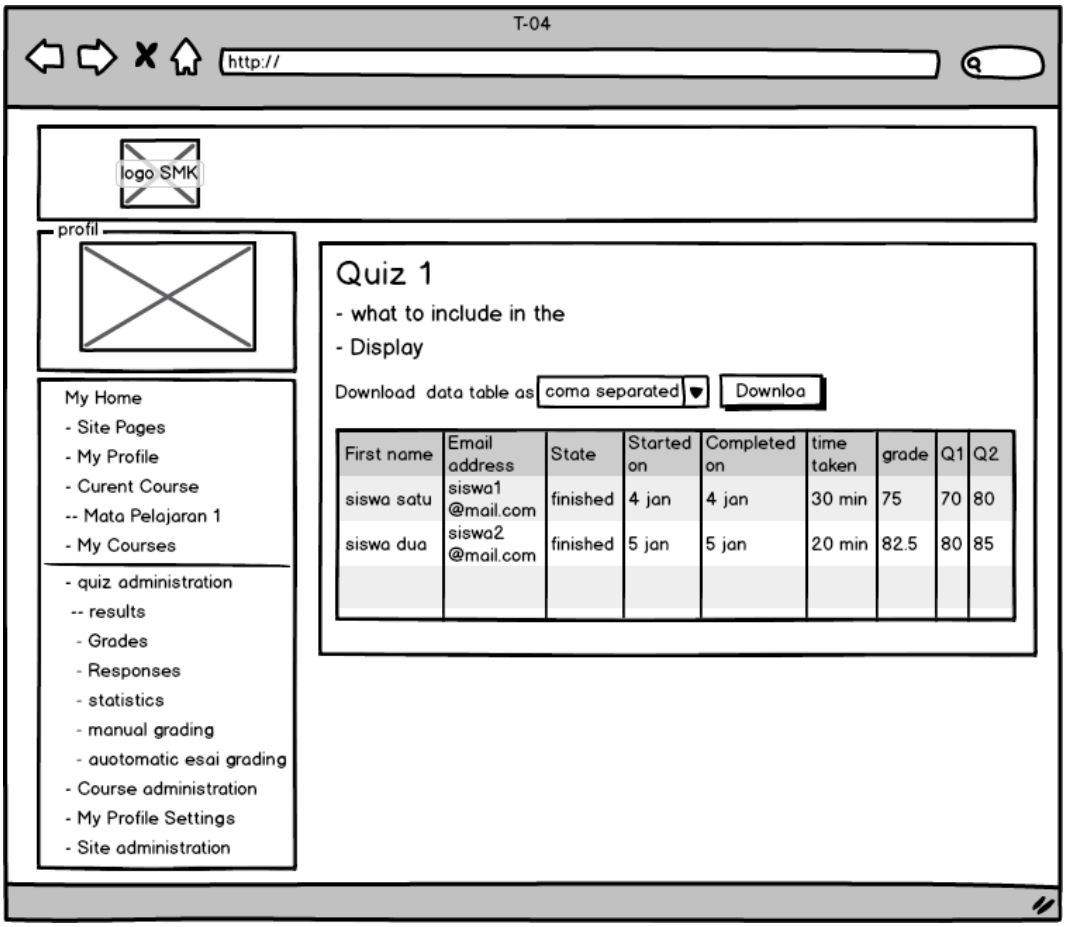
Gambar 3. 16 Tampilan halaman pilih nilai

Keterangan :

- Klik *my home* kembali di tampilan T-01
- Klik *Quiz* 1 masuk ke tampilan T-04
- Ukuran layar = 1366 x 768
- Huruf = Times New Roman
- Warna = putih, orange

4. Tampilan halaman pilih nilai *quiz* (*grade quiz*)

   Tampilan ini adalah tampilan tentang nilai dari suatu *quiz* yang telah diadakan dalam aktivitas pembelajaran. Mencantumkan hasil seluruh siswa dan memberikan tampilan nilai per soal.



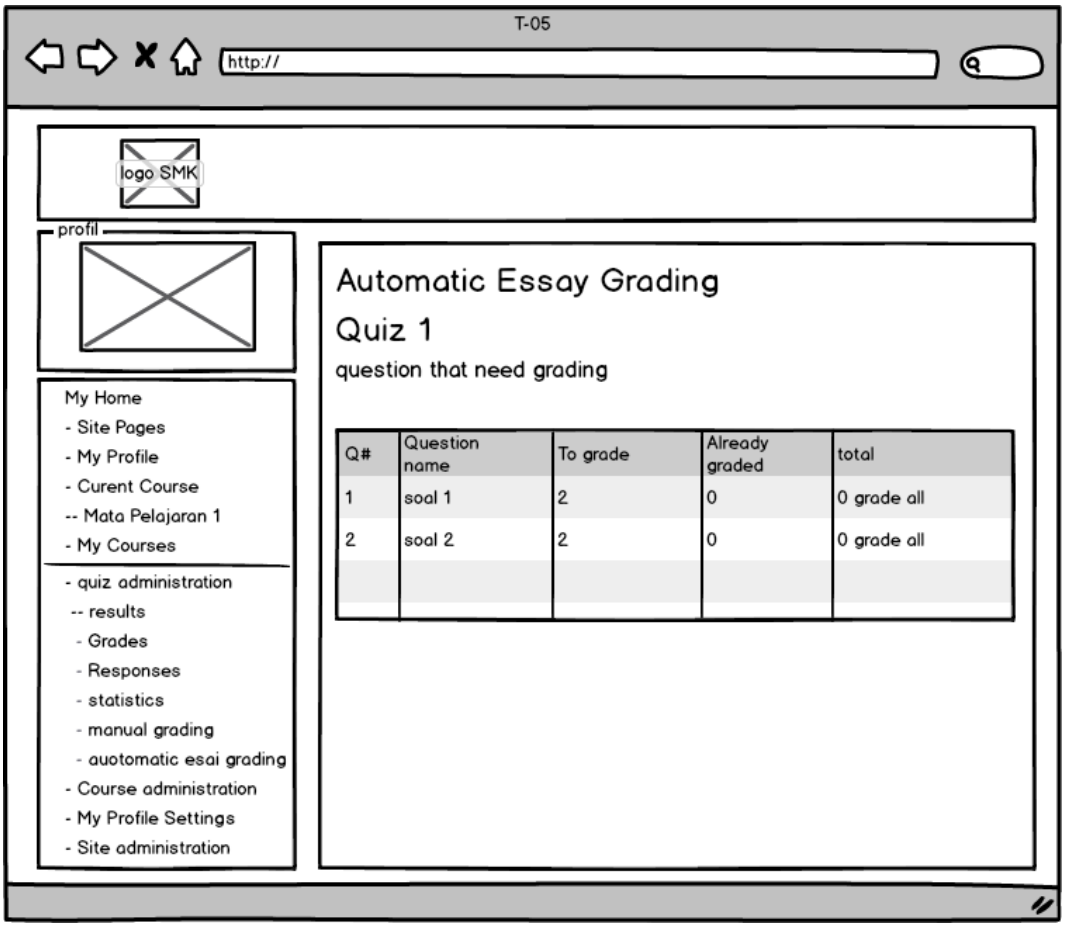
Gambar 3. 17 Tampilan halaman pilih nilai quiz

Keterangan :

- Klik *my home* tetap berada di tampilan T-01
- Klik *automatic essay grading* menuju ke tampilan T-05
- Ukuran layar = 1366 x 768
- Huruf = *Times New Roman*
- Warna = putih, orange

5. Tampilan halaman *automatic essay grading*

   Tampilan ini adalah tampilan halaman yang digunakan untuk melakukan koreksi esai otomatis. Dimana guru dapat melakukan proses koreksi persoal ataupun dalam satu kali klik seluruh soal esai dilakukan koreksi secara otomatis.



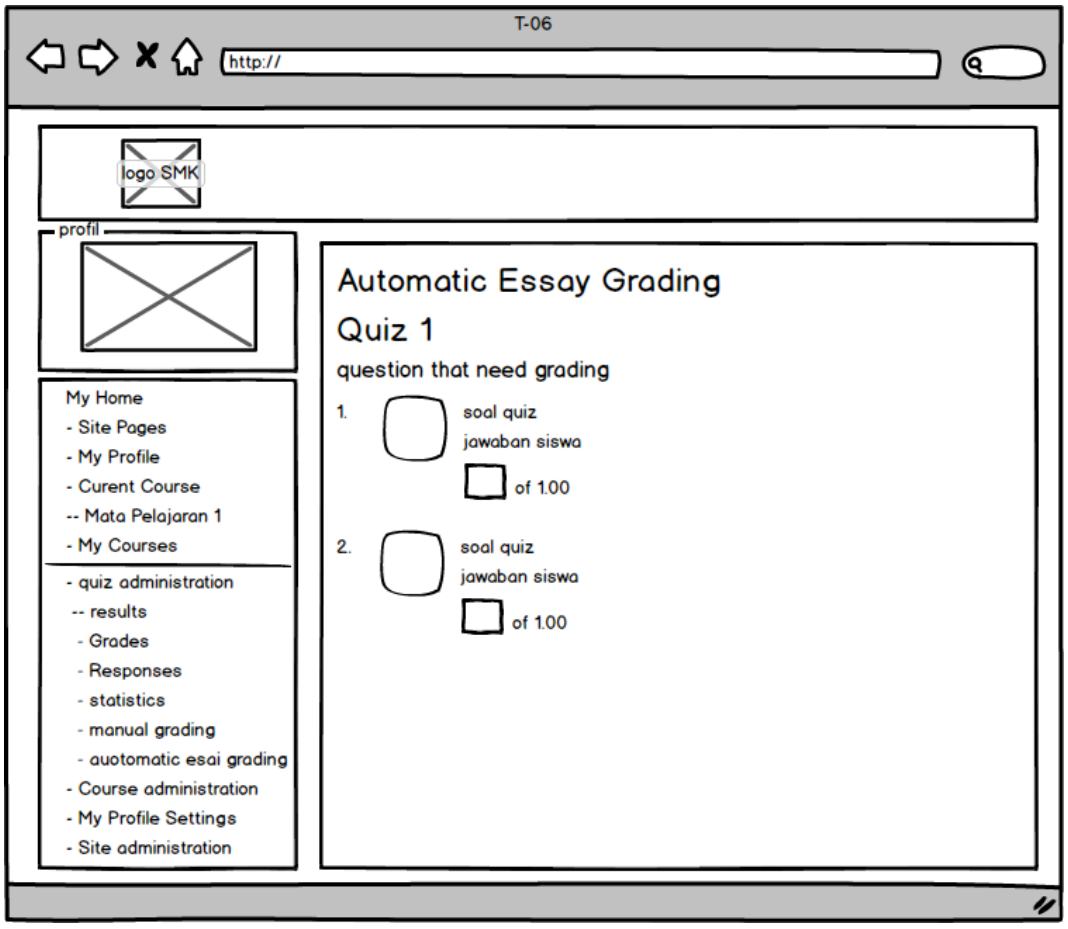
Gambar 3. 18 Tampilan halaman automatic essay grading

Keterangan :

- Klik *my home* kembali di tampilan T-01
- Klik *grade all* masuk ke tampilan T-06
- Ukuran layar = 1366 x 768
- Huruf = *Times New Roman*
- Warna = putih, orange

6. Tampilan halaman hasil *automatic essay grading*

   Tampilan ini adalah tampilan halaman yang digunakan untuk melakukan koreksi esai otomatis. Dimana guru dapat melakukan proses koreksi persoal ataupun dalam satu kali klik seluruh soal esai dilakukan koreksi secara otomatis.



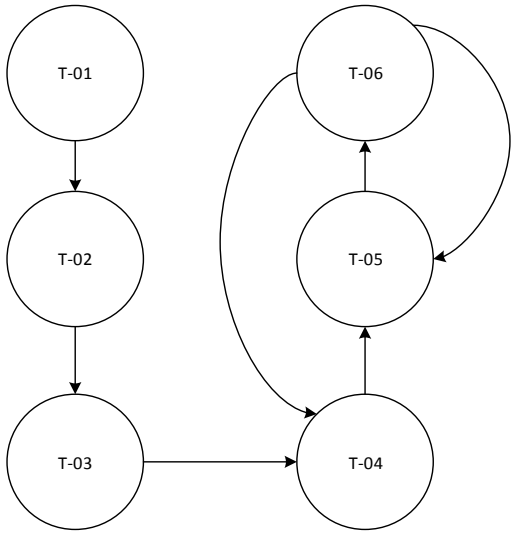
Gambar 3. 19 Tampilan halaman hasil automatic essay grading

Keterangan :

- Klik *my home* kembali di tampilan T-01
- Klik *grade all* masuk ke tampilan T-06
- Ukuran layar = 1366 x 768
- Huruf = *Times New Roman*
- Warna = putih, orange

3.5.3 Jaringan semantik

Struktur aplikasi dibuat secara moduler yaitu program dipecah menjadi modul atau *class* kecil yang mudah dibuat , mudah dites dan mudah dimodifikasi. Dan berikut adalah jaringan semantik yang menunjukan keterkaitan antar bagian



Gambar 3. 20 Jaringan semantik koreksi esai otomatis