**DOKUMEN TEKNIS**

**PERANGKAT LUNAK**

**Sistem Penilaian Esai Otomatis (SPESO)**



Dipersiapkan oleh:

Asep Mulyadi

1203148

**PROGRAM STUDI ILMU KOMPUTER**

**DEPARTEMEN PENDIDIKAN ILMU KOMPUTER**

**FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PENDIDIKAN INDONESIA**

**2016**

# 

# DAFTAR ISI

[DAFTAR ISI 1](#_Toc453535418)

[DAFTAR GAMBAR 3](#_Toc453535419)

[DAFTAR TABEL 4](#_Toc453535420)

[1 PENDAHULUAN 6](#_Toc453535421)

[1.1 Tujuan Penulisan Dokumen 6](#_Toc453535422)

[1.2 Lingkup Masalah 6](#_Toc453535423)

[1.3 Definisi, Istilah dan Singkatan 7](#_Toc453535424)

[1.4 Aturan Penomoran 7](#_Toc453535425)

[1.5 Referensi 8](#_Toc453535426)

[1.6 Deskripsi Umum Dokumen 8](#_Toc453535427)

[2 KEBUTUHAN PERANGKAT LUNAK 10](#_Toc453535428)

[2.1 Deskripsi Umum Sistem 10](#_Toc453535429)

[2.2 Kebutuhan Sistem 10](#_Toc453535430)

[2.3 Fungsi Utama Perangkat Lunak 11](#_Toc453535431)

[2.3.1 Kebutuhan Fungsional 11](#_Toc453535432)

[2.3.2 Kebutuhan Non Fungsional 12](#_Toc453535433)

[2.3.3 Kebutuhan Informasi 13](#_Toc453535434)

[2.4 Karakteristik Pengguna 13](#_Toc453535435)

[2.5 Kebutuhan Antarmuka Eksternal 14](#_Toc453535436)

[2.6 Antarmuka Perangkat Lunak 14](#_Toc453535437)

[2.7 Batasan Sistem 14](#_Toc453535438)

[3 MODEL DESKRIPSI KEBUTUHAN PERANGKAT LUNAK 16](#_Toc453535439)

[3.1 Model Sistem 16](#_Toc453535440)

[3.1.1 *Context Diagram* (Diagram Konteks) 16](#_Toc453535441)

[3.1.2 Data Flow Diagram (Diagram Aliran Data) 16](#_Toc453535442)

[3.1.3 Data Dictionary (Kamus Data) 20](#_Toc453535443)

[3.1.4 Struktur Data 20](#_Toc453535444)

[3.1.5 *Process Spesification* (Spesifikasi Proses) 20](#_Toc453535445)

[3.2 Model Data 25](#_Toc453535446)

[4 RINGKASAN KEBUTUHAN 27](#_Toc453535447)

[4.1 Ringkasan Kebutuhan Fungsional 27](#_Toc453535448)

[4.2 Ringkasan Kebutuhan Non Fungsional 27](#_Toc453535449)

[4.3 Ringkasan Kebutuhan Informasi 27](#_Toc453535450)

[5 MODEL PERANCANGAN 28](#_Toc453535451)

[5.1 Perancangan Data 28](#_Toc453535452)

[5.2 Perancangan Antarmuka Pengguna (*User Interface*) 28](#_Toc453535453)

[6 IMPLEMENTASI 30](#_Toc453535454)

[6.1 Impelemntasi Basis Data 30](#_Toc453535455)

[6.2 Impelemntasi Modul Program 30](#_Toc453535456)

[6.3 Impelemntasi Antarmuka 32](#_Toc453535457)

[7 PENGUJIAN PERANGKAT LUNAK 33](#_Toc453535458)

[7.1 Lingkungan Pengujian 33](#_Toc453535459)

[7.2 Pelaksanaan Pengujian 33](#_Toc453535460)

[7.3 Hasil Pengujian *Black Box* 34](#_Toc453535461)

# DAFTAR GAMBAR

[Gambar 3.1 Context Diagram 16](#_Toc453536032)

[Gambar 3.2 Data Flow Diagram SPESO Level 1 17](#_Toc453536033)

[Gambar 3.3 Data Flow Diagram SPESO Level 2 Proses 1.1 18](#_Toc453536034)

[Gambar 3.4 Data Flow Diagram SPESO Level 2 Proses 1.2 19](#_Toc453536035)

[Gambar 3.5 Model Data SPESO 26](#_Toc453536036)

[Gambar 5.1 Physical Data Model (PDM) 28](#_Toc453535950)

[Gambar 5. 2 Desain Antarmuka User 29](#_Toc453535951)

[Gambar 6.1 Antarmuka Halam User 32](#_Toc453535984)

# DAFTAR TABEL

[Tabel 2.1 Kebutuhan Fungsional Sistem 11](#_Toc453536064)

[Tabel 2.2 Kebutuhan Non Fungsional Sistem 12](#_Toc453536065)

[Tabel 2.3 Kebutuhan Informasi Sistem 13](#_Toc453536066)

[Tabel 2.4 Kebutuhan Antarmuka Eksternal Sistem 14](#_Toc453536067)

[Tabel 2.5 Antarmuka Perangkat Lunak Sistem 14](#_Toc453536068)

[Tabel 3.1 Kamus Data Kata Dasar 20](#_Toc453536069)

[Tabel 3.2 Data Kata Dasar 20](#_Toc453536070)

[Tabel 3.3 Spesifikasi Proses 1.1 Preprocessing 20](#_Toc453536071)

[Tabel 3.4 Spesifikasi Proses 1.2 Perhitungan NGram 21](#_Toc453536072)

[Tabel 3.5 Spesifikasi Proses 1.3 Similarity 22](#_Toc453536073)

[Tabel 3.6 Spesifikasi Proses 1.1.1 Case Folding 23](#_Toc453536074)

[Tabel 3.7 Spesifikasi Proses 1.1.2 Stopwords 23](#_Toc453536075)

[Tabel 3.8 Spesifikasi Proses 1.1.3 Stemming 23](#_Toc453536076)

[Tabel 3.9 Spesifikasi Proses 1.2.1 Text NGram 24](#_Toc453536077)

[Tabel 3.10 Spesifikasi Proses 1.2.2 Vektor NGram 24](#_Toc453536078)

[Tabel 4.1 Ringkasan Kebutuhan Fungsional 27](#_Toc453536079)

[Tabel 4.2 Ringkasan Kebutuhan Non Fungsional 27](#_Toc453536080)

[Tabel 4.3 Ringkasan Kebutuhan Informasi 27](#_Toc453536081)

[Tabel 6.1 Tabel Kata Dasar 30](#_Toc453536115)

[Tabel 6.2 Impelemntasi Modul Program 30](#_Toc453536116)

[Tabel 7.1 Pelaksanaan Pengujian 33](#_Toc453536121)

[Tabel 7.2 Hasil Pengujian SPESO-F-001 34](#_Toc453536122)

[Tabel 7.3 Hasil Pengujian SPESO-F-002 35](#_Toc453536123)

[Tabel 7.4 Hasil Pengujian SPESO-F-003 35](#_Toc453536124)

[Tabel 7.5 Hasil Pengujian SPESO-F-004 36](#_Toc453536125)

# 1 PENDAHULUAN

## 1.1 Tujuan Penulisan Dokumen

Dokumen teknis ini merupakan dokumen yang berisi tentang penjelasan perangkat lunak pada penelitian yang berjudul “Sistem Penilaian Jawaban Esai Otomatis Menggunakan Kombinasi Metode *Term Frequency* dan N-Gram”. Tujuan dibuatnya dokumen ini adalah:

1. Menjelaskan tahapan-tahapan pembuatan perangkat lunak
2. Menjelaskan fungsi-fungsi yang terdapat pada perangkat lunak
3. Sebagai pedoman untuk pengembangan perangkat lunak selanjutnya

Target pembaca dokumen ini adalah pihak-pihak yang akan menggunakan aplikasi ini, sehingga dalam penggunaannya nanti dapat dilakukan sebagaimana mestinya. Selain itu dokumen ini dijadikan sebagai landasan pengembangan dan perawatan Sistem Penilaian Esai Otomatis (SPESO) baik bagi tim pengembang maupun bagi *stake holder*.

## 1.2 Lingkup Masalah

Permasalahan yang akan diselesaikan oleh aplikasi ini adalah bagaimana melakukan penilaian jawaban esai secara otomatis yang menghasilkan daftar nilai dengan kunci jawaban dan jawaban siswa yang diberikan user.

## 1.3 Definisi, Istilah dan Singkatan

Untuk membaca dokumen teknis ini, diperlukan untuk menyamakan persepsi mengenai definisi, istilah dan singkatan yang ada. Berikut definisi, istilah dan definisi yang dapat dilihat pada tabel 1.1.

**Tabel 1.1 Definisi, Istilah dan Singkatan**

|  |  |
| --- | --- |
| **Istilah/Singkatan** | **Definisi** |
| SPESO | Nama dari perangkat lunak pada penelitian ini |
| *Term Frequency* (TF) | Frekuensi kemunculan kata pada suatu dokumen |
| *Vector Space Model* (VSM) | Representasi kumpulan dokumen sebagai vektor dalam sebuah ruang vektor |
| *Stopwrods* | Penghapusan kata yang tidak penting |
| *Stemming* | Pemotongan imbuhan kata |
| *Cosine Similarity* | Menghitung kemiripan antara dua item dari sudut kosinus yang terbuat diantara kedua item |
| *Jaccard Similarity* | Metode yang dipakai untuk menghitung *similarity* antara dua vektor |

Dapat dilihat pada tabel 1.1 bahwa terdapat 5 daftar istilah atau singkatan yang terdapat pada dokumen ini.

## 1.4 Aturan Penomoran

Pembahasan akan dijelaskan dengan hirarki bab, sub-sub, sub-sub-bab, dan seterusnya. Aturan dari penomoran tersebut adalah <no bab>.<no sub-sub>.<no sub-sub bab>.<dst> <judul penomoran>. Untuk poin terurut, maka ditulis dengan urutan abjad dilanjutkan dengan urutan angka. Untuk poin tidak terurut, ditulis dengan symbol *bullets*. Penomoran untuk menuliskan kebutuhan perangkat lunak ditulis dengan <nama SKPL>.<nama fungsi>.<no urut>. Contoh dari penomoran kebutuhan fungsional yaitu “SPESO-F-001”, dimana SPESO merupakan nama sistem, F merupakan nama kebutuhan, dan 001 merupakan no urut kebutuhan fungsional.

## 1.5 Referensi

Dokumen teknis ini disusun berdasarkan standar:

1. Sistematika pada dokumen teknis ini adalah sistematika yang digunakan oleh program Ilmu Komputer Universitas Indonesia.
2. Pressman, Roger S. 2001. Software Engineering: A Practitioner’s Approach 5th ed. McGraw-Hill Book Co.

## 1.6 Deskripsi Umum Dokumen

Dokumen ini terdiri dari 7 bab, berikut penjelasan mengenai tiap bab.

1. BAB 1 PENDAHULUAN

Bab ini berisi tentang tujuan penulisan dokumen, lingkup masalah, definisi istilah dan singkatan, aturan penomoran, referensi pembuatan dokumen teknis dan deskripsi umum dokumen.

1. BAB 2 KEBUTUHAN PERANGKAT LUNAK

Bab ini berisi tentang deskripsi umum sistem, kebutuhan sistem, fungsi utama perangkat lunak (kebutuhan fungsional, kebutuhan non fungsional, kebutuhan informasi), karakteristik pengguna, kebutuhan antarmuka eksternal (antarmuka pengguna, antarmuka perangkat lunak), antarmuka perangkat lunak, dan batasan sistem.

1. BAB 3 MODEL DESKRIPSI KEBUTUHAN PERANGKAT LUNAK

Bab ini berisi uraian-uraian kebutuhan perangkat lunak secara lebih rinci. Deskripsi dari kebutuhan perangkat lunak dipresentasikan dalam bentuk model yang digambar dengan bentuk diagram.

1. BAB 4 RINGKASAN KEBUTUHAN

Bab ini menjelaskan hasil perancangan perangkat lunak, mencakup ringkasan kebutuhan fungsional, ringkasan kebutuhan non fungsional, dan ringkasan kebutuhan informasi.

1. BAB 5 PERANCANGAN PERANGKAT LUNAK

Bab ini menjelaskan hasil perancangan data dan antarmuka yang digunakan.

1. BAB 6 IMPLEMENTASI

Bab ini berisi tentang implementasi data berupa implementasi basis data, implementasi modul program, dan implementasi antarmuka.

1. BAB 7 PENGUJIAN PERANGKAT LUNAK

Bab ini berisi tentang lingkungan pengujian, pelaksanaan pengujian, dan hasil pengujian *black box*.

# 2 KEBUTUHAN PERANGKAT LUNAK

## 2.1 Deskripsi Umum Sistem

SPESO merupakan sebuah sistem penilaian jawaban esai otomatis berbasis dekstop yang terhubung dengan MySQL. Dalam sistem ini, pengguna dapat melakukan penilaian jawaban esai secara otomatis dengan mengupload data soal dan data jawaban siswa.

Tahap awal kerja sistem ini adalah pengguna memilih jenis N-Gram, memasukkan jumlah soal dan jumlah siswa, dan memilik persamaan similarity yang akan digunakan. Ketika proses sedang berlangsung, sistem akan melakukan *preprocessing* terhadap data soal dan jawaban siswa. Hasil dari preprocessing tersebut akan diolah menggunakan perhitungan N-Gram sehingga menghasilkan vektor kunci jawaban dan vektor jawaban siswa. Setelah data terbentuk menjadi vektor, maka data tersebut akan dihitung menggunakan *similarity* untuk memperoleh nilai. Nilai tersebut didapat dari perhitungan kemiripan antara vektor kunci jawaban dengan vektor jawaban siswa.

## 2.2 Kebutuhan Sistem

Sistem yang dikembangkan memiliki tingkatan aplikasi dari sisi layanan dan pengelolaan data sebagai berikut:

1. Data Soal

Data ini merupakan kumpulan soal, kunci jawaban dan bobot tiap soal yang digunakan untuk proses penilaian.

1. Data Jawaban Siswa

Data ini merupakan kumpulan jawaban siswa dan nilai dari guru, di mana data tersebut digunakan sebagai pembanding dengan kunci jawaban.

1. Data *Stopwords*

Data ini merupakan list *stopwords* yang akan digunakan ketika *preprocessing*.

1. Data Kata Dasar

Daat ini merupakan kumpulan kata dasar yang akan digunakan ketika proses *stemming*.

## 2.3 Fungsi Utama Perangkat Lunak

Dalam pembuatan dokumen teknis SPESO ini terdapat beberapa fungsi utama perangkat lunak meliputi kebutuhan fungsional, non fungsional dan kebutuhan informasi. Setiap fungsi diberikan kode sesuai dengan aturan penomoran yang telah dijelaskan sebelumnya. Penjelasan fungsi-fungsi utama perangkat lunak adalah sebagai berikut:

### 2.3.1 Kebutuhan Fungsional

**Tabel 2.1 Kebutuhan Fungsional Sistem**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kode** | **Nama Fungsi** | **Deskripsi** |
| 1 | SPESO-F-001 | Import data soal | Import data soal tes. Eksternal file berformat Microsoft Excel (.csv) 2003 |
| 2 | SPESO-F-002 | Import data jawaban siswa | Import data jawaban siswa dalam tes. Eksternal file berformat Microsoft Excel (.csv) 2003 |
| 3 | SPESO-F-003 | Penentuan nilai | Proses pemeriksaan dan pencocokan jawaban siswa dengan kunci jawaban yang menghasilkan nilai dari masing-masing jawaban tersebut |
| 4 | SPESO-F-004 | Eksport nilai | Mengeksport nilai ke dalam file. Eksternal file berformat Microsoft Excel (.xls) 2003 |

### 2.3.2 Kebutuhan Non Fungsional

**Tabel 2.2 Kebutuhan Non Fungsional Sistem**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kode** | **Aspek** | **Deskripsi** |
| 1 | SPESO-NF-001 | *Availability* | Sistem tersedia untuk pengguna selama 24 jam dalam sehari, 7 hari dalam seminggu |
| 2 | SPESO-NF-002 | *Ergonomy* | Sistem harus memiliki tampilan yang user friendly sehingga dalam penggunaannya nanti memberikan kenyamanan dan tidak membuat bingung |
| 3 | SPESO-NF-003 | *Portability* | Sistem harus mampu berjalan pada berbagai sistem operasi yang mendung MySQL |
| 4 | SPESO-NF-004 | *Response time* | Sistem mampu memberikan respon dengan cepat berdasarkan input pengguna |

### 2.3.3 Kebutuhan Informasi

Data-data yang diperlukan untuk membangun perangkat lunak dijelaskan pada Tabel 2.3.

**Tabel 2.3 Kebutuhan Informasi Sistem**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Kode** | **Informasi** | **Tujuan** | **Frekuensi** | **Format** |
| 1 | SPESO-I-001 | Data soal | Menyajikan informasi data soal dan kunci jawaban untuk suatu penilaian | 1 | Layar |
| 2 | SPESO-I-002 | Data jawaban siswa | Menyajikan data jawaban siswa untuk satu soal dalam suatu penilaian | 1 | Layar |
| 3 | SPESO-I-003 | Data nilai jawaban siswa | Menyajikan nilai yang dihasilkan sistem untuk setiap peserta di dalam suatu penilaian | 1 | Layar |

## 2.4 Karakteristik Pengguna

Karakteristik pengguna dalam sistem SPESO adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| Kualifikasi | : | Memahami pengoperasian komputer, memahami dan mampu mengoperasikan fungsi-fungsi aplikasi |
| Tugas | : | Sebagai pengguna aplikasi SPESO |
| Hak Akses | : | Memasukkan data soal dan jawaban siswa, melihat hasil penilaian |

## 2.5 Kebutuhan Antarmuka Eksternal

Kebutuhan antarmuka eksternal pada aplikasi SPESO agar dapat digunakan oleh pengguna, bisa dilihat pada Tabel 2.4.

**Tabel 2.4 Kebutuhan Antarmuka Eksternal Sistem**

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat** | **Kegunaan** |
| 1 | Mouse | Untuk mengarahkan kursor |
| 2 | Layar monitor | Untuk menampilkan antarmuka aplikasi |
| 3 | Keyboard | Untuk menginputkan data-data yang dibutuhkan pada aplikasi |

## 2.6 Antarmuka Perangkat Lunak

Kebutuhan antarmuka perangkat lunak pada aplikasi SPESO dapat dilihat pada Tabel 2.5.

**Tabel 2.5 Antarmuka Perangkat Lunak Sistem**

|  |  |  |
| --- | --- | --- |
| **No** | **Nama** | **Fungsi** |
| 1 | Windows | Sebagai sistem operasi tempat dijalankannya perangkat lunak |
| 2 | Web Server XAMPP ver. 3.2.1 | Sebagai server lokal |

## 2.7 Batasan Sistem

Aplikasi penilaian esai otomatis (SPESO) merupakan sebuah sistem yang berdiri sendiri (*stand-alone*). SPESO tidak terhubung atau tergantung dengan sistem lainnya. Batasan-batasan dalam pengembangan perangkat lunak ini adalah sebagai berikut:

1. Sistem ini tidak dibuat khusus untuk di akses melalui smartphone maupun perangkat mobile sejenis
2. Sistem ini hanya dapat digunakan dalam penilaian jawaban esai berbahasa Indonesia

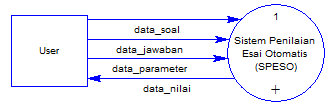
# 3 MODEL DESKRIPSI KEBUTUHAN PERANGKAT LUNAK

## 3.1 Model Sistem

Pemodelan kebutuhan sistem dibuat berdasarkan hasil analisis. Pemodelan dilakukan dengan menggunakan *Context Diagram*, *Data Flow Diagram*, *Data Dictionary*, dan *Process Specification*.

### 3.1.1 *Context Diagram* (Diagram Konteks)

Pengguna yang akan menggunakan aplikasi SPESO adalah *user*. *User* akan memilih jenis N-Gram, memasukkan data soal, memasukkan data jawaban siswa, jumlah soal, jumlah siswa, dan memilih similarity. *Context Diagram* SPESO dapat dilihat pada Gambar 3.1.

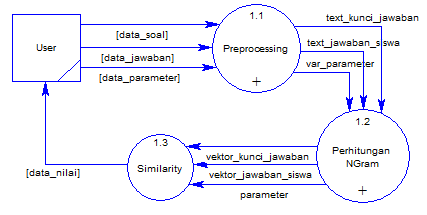


**Gambar 3.1 Context Diagram**

### 3.1.2 Data Flow Diagram (Diagram Aliran Data)

Diagram aliran data dari SPESO adalah sebagai berikut:

1. **Diagram Aliran Data SPESO Level 1**



**Gambar 3.2 Data Flow Diagram SPESO Level 1**

Berdasarkan pada Gambar 3.2, dapat dilihat bahwa terdapat tiga proses turunan yang dapat dijelaskan sebagai berikut.

**Proses 1.1 *Preprocessing***

Merupakan proses awal sebelum dalam mengolah data input sebelum memasuki proses tahapan utama, metode yang digunakan. Dalam proses ini terdapat tiga proses, yaitu *case folding*, *stopwords*, dan *stemming* yang akan dijelaskan pada level selanjutnya.

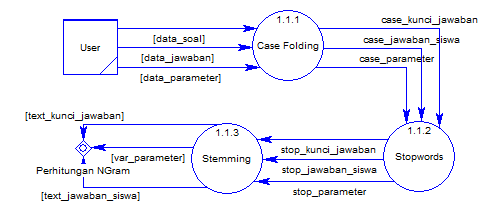
**Proses 1.2 Perhitungan N-Gram**

Merupakan proses utama dalam sistem SPESO, karena pada proses ini metode kombinasi *term frequency* dan N-Gram akan diimplementasikan. Output dari proses ini adalah berupa vektor kunci jawaban dan jawaban siswa.

**Proses 1.3 *Similarity***

Merupakan proses akhir dari sistem SPESO, di mana vektor-vektor yang dihasilkan dari proses perhitungan N-Gram akan dihitung untuk mengetahui tingkat kemiripan yang akan dibandingkan dengan nilai dari guru.

1. **Diagram Aliran Data SPESO Level 2 Proses 1.1**



**Gambar 3.3 Data Flow Diagram SPESO Level 2 Proses 1.1**

Dapat dilihat pada Gambar 3.3 bahwa terdapat tiga proses turunan, yaitu:

**Proses 1.1.1 *Case Folding***

Dalam proses ini, data kunci jawaban dan jawaban siswa akan diproses untuk pengubahan dari huruf besar ke huruf kecil.

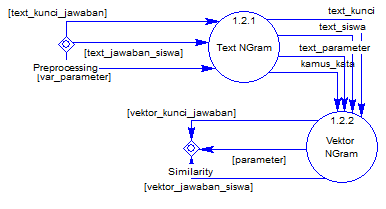
**Proses 1.1.2 *Stopwords***

Dalam proses ini akan dilakukan penghapusan kata yang tidak penting, berdasarkan data pada *list stopwords*. Jika kata yang terdapat pada list stopwords ada pada kalimat kunci jawaban dan jawaban siswa, maka kata tersebut akan dihapus dari kalimat tersebut.

**Proses 1.1.3 *Stemming***

Dalam proses ini, kata yang memiliki imbuhan akan dihapus baik itu prefiks, suffiks, maupun konfiks. Dari proses ini akan menghasilkan kata-kata dasar, kata yang tidak mengandung imbuhan.

1. **Diagram Aliran Data SPESO Level 2 Proses 1.2**



**Gambar 3.4 Data Flow Diagram SPESO Level 2 Proses 1.2**

Dapat dilihat pada Gambar 3.4 bahwa terdapat dua proses turunan, yaitu:

**Proses 1.2.1 Text NGram**

Dalam proses ini, data hasil dari preprocessing akan dibentuk menjadi kamus kata. Kamus kata yang dibentuk berdasarkan jenis N-Gram dari kunci jawaban dan jawaban siswa.

**Proses 1.2.2 Vektor NGram**

Dalam proses ini akan dibentuk vektor-vektor kunci jawaban dan jawaban siswa berdasarkan kamus kata.

### 3.1.3 Data Dictionary (Kamus Data)

**Tabel 3.1 Kamus Data Kata Dasar**

|  |  |
| --- | --- |
| Nama | Data kata dasar |
| Alias | Tidak ada |
| Digunakan | Sebagai input  Proses 1.1.3 pada DFD level 2, *stemming*  Sebagai output  Proses 1.1.3 pada DFD level 2, *stemming* |
| Deskripsi | Pada proses *stemming*, kata yang masuk akan dicek apakah kata tersebut merupakan kata dasar atau tidak. Jika tidak, maka akan dilakukan proses *stemming*. |

### 3.1.4 Struktur Data

**Tabel 3.2 Data Kata Dasar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Item Data** | **Tipe** | **Ukuran** | **Keterangan** |
| 1 | id\_katadasar | int | 10 | id\_katadasar berupa auto\_increment |
| 2 | katadasar | varchar | 70 | kata dasar |
| 3 | Tipe\_katadasar | varchar | 25 | tipe kata apakah verba, nomina, adjektiva, dll |

### 3.1.5 *Process Spesification* (Spesifikasi Proses)

**Tabel 3.3 Spesifikasi Proses 1.1 Preprocessing**

|  |  |
| --- | --- |
| Nomor | 1.1 |
| Nama | *Preprocessing* |
| Input | data\_soal, data\_jawaban, dan data\_parameter |
| Output | text\_kunci\_jawaban, teks\_jawaban\_siswa, dan var\_parameter |
| Logika | **Input** (data\_soal, data\_jawaban, data\_parameter)  data\_soal = lower(data\_soal)  data\_jawaban = lower(data\_jawaban)  stopword\_list = open(stopwords\_list.txt)  sintaks\_html\_list = open(sintaks\_html.txt)  kunci = stopword(data\_soal, stopword\_list)  siswa = stopword(data\_soal, stopword\_list)  kunci = stopword(data\_soal, sintaks\_html\_list)  siswa = stopword(data\_soal, sintaks\_html\_list)  text\_kunci\_jawaban = stemming(kunci)  text\_jawaban\_siswa = stemming(siswa)  **Output** (text\_kunci\_jawaban, text\_jawaban\_siswa, var\_parameter) |

**Tabel 3.4 Spesifikasi Proses 1.2 Perhitungan NGram**

|  |  |
| --- | --- |
| Nomor | 1.2 |
| Nama | Perhitungan NGram |
| Input | text\_kunci\_jawaban, teks\_jawaban\_siswa, dan var\_parameter |
| Output | vektor\_kunci\_jawaban, vektor\_jawaban\_siswa, dan parameter |
| Logika | **Input** (text\_kunci\_jawaban, teks\_jawaban\_siswa, var\_parameter  **If** var\_parameter[banyakGram] = 1  out\_unigram = teksUnigram(teks\_kunci\_jawaban,  teks jawaban\_siswa)  vektorKunciUnigram = vektorUnigram  (out\_unigram, text\_kunci\_jawaban)  vektorSiswaUnigram = vektorUnigram  (out\_unigram, text\_jawaban\_siswa)  **Else If** var\_parameter[banyakGram] = 2  out\_bigram = teksBigram(teks\_kunci\_jawaban,  teks jawaban\_siswa)  vektorKunciBigram = vektorBigram  (out\_bigram, text\_kunci\_jawaban)  vektorSiswaBigram = vektorBigram  (out\_bigram, text\_jawaban\_siswa)  **Else If** var\_parameter[banyakGram] = 3  out\_unigram\_bigram = teksUnigramBigram  (teks\_kunci\_jawaban, teks jawaban\_siswa)  vektorKunciUnigramBigram =  vektorUnigramBigram(out\_unigram\_bigram,  text\_kunci\_jawaban)  vektorSiswaUnigramBigram =  vektorUnigramBigram (out\_unigram\_bigram,  text\_jawaban\_siswa)  **End**  **Output** (vektor\_kunci\_jawaban, vektor\_jawaban\_siswa, dan parameter) |

**Tabel 3.5 Spesifikasi Proses 1.3 Similarity**

|  |  |
| --- | --- |
| Nomor | 1.3 |
| Nama | Similarity |
| Input | vektor\_kunci\_jawaban, vektor\_jawaban\_siswa, dan parameter |
| Output | data\_nilai |
| Logika | **Input** (vektor\_kunci\_jawaban, vektor\_jawaban\_siswa, dan parameter)  **If** var\_parameter[banyakGram] = 1  **If** var\_parameter[similarity] = cosineSimilarity  nilaiUnigram = cosineSimilarity  (vektor\_kunci\_jawaban, vektor\_jawaban\_siswa)  **Else If** var\_parameter[similarity] =  jaccardSimilarity  nilaiUnigram = jaccardSimilarity  (vektor\_kunci\_jawaban, vektor\_jawaban\_siswa)  **End**  data\_nilai = nilaiUnigram  **Else If** var\_parameter[banyakGram] = 2  **If** var\_parameter[similarity] = cosineSimilarity  nilaiBigram = cosineSimilarity  (vektor\_kunci\_jawaban, vektor\_jawaban\_siswa)  **Else If** var\_parameter[similarity] =  jaccardSimilarity  nilaiBigram = jaccardSimilarity  (vektor\_kunci\_jawaban, vektor\_jawaban\_siswa)  **End**  data\_nilai = nilaiBigram  **Else If** var\_parameter[banyakGram] = 3  **If** var\_parameter[similarity] = cosineSimilarity  nilaiUnigramBigram = cosineSimilarity  (vektor\_kunci\_jawaban, vektor\_jawaban\_siswa)  **Else If** var\_parameter[similarity] =  jaccardSimilarity  nilaiUnigramBigram = jaccardSimilarity  (vektor\_kunci\_jawaban, vektor\_jawaban\_siswa)  **End**  data\_nilai = nilaiUnigramBigram  **End**  **Output** (data\_nilai) |

**Tabel 3.6 Spesifikasi Proses 1.1.1 Case Folding**

|  |  |
| --- | --- |
| Nomor | 1.1.1 |
| Nama | *Case Folding* |
| Input | data\_soal, data\_jawaban, dan data\_parameter |
| Output | case\_kunci\_jawaban, case\_jawaban\_siswa, dan case\_parameter |
| Logika | **Input** (data\_soal, data\_jawaban, data\_parameter)  case\_kunci\_jawaban = lower(data\_soal)  case\_jawaban\_siswa = lower(data\_jawaban)  **Output** (case\_kunci\_jawaban, case\_jawaban\_siswa, dan case\_parameter) |

**Tabel 3.7 Spesifikasi Proses 1.1.2 Stopwords**

|  |  |
| --- | --- |
| Nomor | 1.1.2 |
| Nama | *Stopwords* |
| Input | case\_kunci\_jawaban, case\_jawaban\_siswa, dan case\_parameter |
| Output | stop\_kunci\_jawaban, stop\_jawaban\_siswa, dan stop\_parameter |
| Logika | **Input** (case\_kunci\_jawaban, case\_jawaban\_siswa, dan case\_parameter)  stopword\_list = open(stopwords\_list.txt)  sintaks\_html\_list = open(sintaks\_html.txt)  kunci = stopword(case\_kunci\_jawaban, stopword\_list)  siswa = stopword(case\_jawaban\_siswa, stopword\_list)  stop\_kunci\_jawaban = stopword(case\_kunci\_jawaban, sintaks\_html\_list)  stop\_jawaban\_siswa = stopword case\_jawaban\_siswa, sintaks\_html\_list)  **Output** (stop\_kunci\_jawaban, stop\_jawaban\_siswa, dan stop\_parameter) |

**Tabel 3.8 Spesifikasi Proses 1.1.3 Stemming**

|  |  |
| --- | --- |
| Nomor | 1.1.3 |
| Nama | *Stemming* |
| Input | stop\_kunci\_jawaban, stop\_jawaban\_siswa, dan stop\_parameter |
| Output | text\_kunci\_jawaban, teks\_jawaban\_siswa, dan var\_parameter |
| Logika | **Input** (stop\_kunci\_jawaban, stop\_jawaban\_siswa, dan stop\_parameter)  text\_kunci\_jawaban = stemming(stop\_kunci\_jawaban)  text\_jawaban\_siswa = stemming(, stop\_jawaban\_siswa)  **Output** (text\_kunci\_jawaban, teks\_jawaban\_siswa, dan var\_parameter) |

**Tabel 3.9 Spesifikasi Proses 1.2.1 Text NGram**

|  |  |
| --- | --- |
| Nomor | 1.2.1 |
| Nama | *Text* NGram |
| Input | text\_kunci\_jawaban, teks\_jawaban\_siswa, dan var\_parameter |
| Output | text\_kunci, text\_siswa, text\_parameter, dan kamus\_kata |
| Logika | **Input** (text\_kunci\_jawaban, teks\_jawaban\_siswa,  **If** var\_parameter[banyakGram] = 1  out\_unigram = teksUnigram(teks\_kunci\_jawaban,  teks jawaban\_siswa)  kamus\_kata = out\_unigram  **Else If** var\_parameter[banyakGram] = 2  out\_bigram = teksBigram(teks\_kunci\_jawaban,  teks jawaban\_siswa)  kamus\_kata = out\_bigram  **Else If** var\_parameter[banyakGram] = 3  out\_unigram\_bigram = teksUnigramBigram  (teks\_kunci\_jawaban, teks jawaban\_siswa)  kamus\_kata = out\_unigram\_bigram  **End**  **Output** (text\_kunci, text\_siswa, text\_parameter, dan kamus\_kata) |

**Tabel 3.10 Spesifikasi Proses 1.2.2 Vektor NGram**

|  |  |
| --- | --- |
| Nomor | 1.2.2 |
| Nama | Vektor NGram |
| Input | text\_kunci, text\_siswa, text\_parameter, dan kamus\_kata |
| Output | vektor\_kunci\_jawaban, vektor\_jawaban\_siswa, dan parameter |
| Logika | **Input** (text\_kunci, text\_siswa, text\_parameter, dan kamus\_kata)  **If** var\_parameter[banyakGram] = 1  vektorKunciUnigram = vektorUnigram  (kamus\_kata, text\_kunci\_jawaban)  vektorSiswaUnigram = vektorUnigram  (kamus\_kata, text\_jawaban\_siswa)  **Else If** var\_parameter[banyakGram] = 2  vektorKunciBigram = vektorBigram  (kamus\_kata, text\_kunci\_jawaban)  vektorSiswaBigram = vektorBigram  (kamus\_kata, text\_jawaban\_siswa)  **Else If** var\_parameter[banyakGram] = 3  vektorKunciUnigramBigram =  vektorUnigramBigram(kamus\_kata,  text\_kunci\_jawaban)  vektorSiswaUnigramBigram =  vektorUnigramBigram (kamus\_kata,  text\_jawaban\_siswa)  **End**  **Output** (vektor\_kunci\_jawaban, vektor\_jawaban\_siswa, dan parameter) |

## 3.2 Model Data

Model data berfungsi untuk mendiskripsikan objek data yang akan diproses dan menjelaskan semua atribut yang menggambarkan objek data tersebut. Model data juga juga berfungsi untuk menggambarkan kardinalitas relasi antara objek data yang satu dengan yang lainnya. Berikut pemodelan data dari SPESO yang dapat dilihat pada Gambar



**Gambar 3.5 Model Data SPESO**

# 4 RINGKASAN KEBUTUHAN

Kebutuhan sistem merupakan hal-hal yang harus dipenuhi, kebutuhan ini selanjutnya akan menjadi arahan untuk tahapan pengujian, semua kebutuhan sistem yang diuji harus dapat diterima agar dapat dibuktikan bahwa sistem yang dibangun sudah memenuhi kebutuhan. Ringkasan semua item yang sudah diidentifikasi di awal adalah sebagai berikut.

## 4.1 Ringkasan Kebutuhan Fungsional

**Tabel 4.1 Ringkasan Kebutuhan Fungsional**

|  |  |  |
| --- | --- | --- |
| **No** | **Kode Fungsi** | **Nama Fungsi** |
| 1 | SPESO-F-001 | Import data soal |
| 2 | SPESO-F-002 | Import data jawaban siswa |
| 3 | SPESO-F-003 | Penentuan nilai |
| 4 | SPESO-F-004 | Eksport nilai |

## 4.2 Ringkasan Kebutuhan Non Fungsional

**Tabel 4.2 Ringkasan Kebutuhan Non Fungsional**

|  |  |  |
| --- | --- | --- |
| **No** | **Kode Fungsi** | **Nama Fungsi** |
| 1 | SPESO-NF-001 | *Availability* |
| 2 | SPESO-NF-002 | *Ergonomy* |
| 3 | SPESO-NF-003 | *Portability* |
| 4 | SPESO-NF-004 | *Response time* |

## 4.3 Ringkasan Kebutuhan Informasi

**Tabel 4.3 Ringkasan Kebutuhan Informasi**

|  |  |  |
| --- | --- | --- |
| **No** | **Kode Fungsi** | **Nama Fungsi** |
| 1 | SPESO-I-001 | Data soal |
| 2 | SPESO-I-002 | Data jawaban siswa |
| 3 | SPESO-I-003 | Data nilai jawaban siswa |

# 5 MODEL PERANCANGAN

## 5.1 Perancangan Data

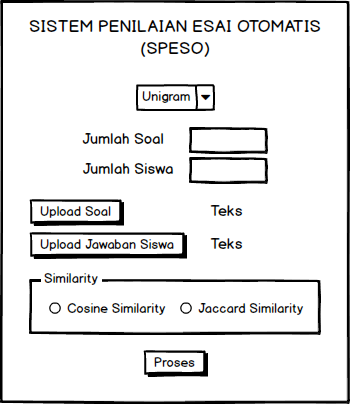
Perancangan basis data diawali dengan membuat Conceptual Data Model (CDM) yang telah dijelaskan pada bab tiga. Langkah selanjutnya adalah membuat model fisik atau disebut dengan Physical Data Model (PDM). Berikut PDM dari SPESO yang dapat dilihat pada gambar



**Gambar 5.1 Physical Data Model (PDM)**

## 5.2 Perancangan Antarmuka Pengguna (*User Interface*)

Gambar 5.2 menunjukkan desain antarmuka untuk halaman user. Halaman ini menampilkan segala kebutuhan sistem dalam melakukan penilaian jawaban esai.



**Gambar 5. 2 Desain Antarmuka User**

# 6 IMPLEMENTASI

## 6.1 Impelemntasi Basis Data

Basis data yang telah dirancang adalah sebagai berikut:

*Database name* : skripsi

*User* : root

*Password* : -

**Tabel 6.1 Tabel Kata Dasar**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama Tabel  Fungsi | | tb\_katadasar  Menyimpan kata dasar | | | |
| **No.** | **Column** | **Type** | **Values** | **Default** | **Action** |
| 1 | Id\_katadasar | Integer | 10 | none | PK, AI |
| 2 | katadasar | varchar | 70 | none |  |
| 3 | Tipe\_katadasar | varchar | 25 | none |  |

## 6.2 Impelemntasi Modul Program

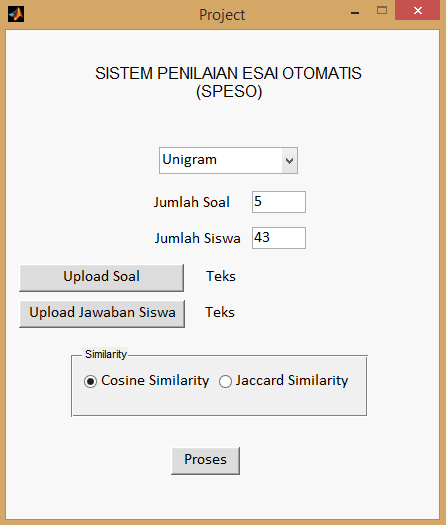
Hasil implementasi modul program dijelaskan dalam Tabel 6.2.

**Tabel 6.2 Impelemntasi Modul Program**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Modul Program** | **Hasil Implementasi File Fisik** | **Keterangan** |
| 1 | GUI | Project.fig | Tampilan sistem |
| 2 | Fungsi GUI | Project.m | Fungsi dari setiap aksi yang ada pada GUI |
| 3 | Proses | Proses.m | Proses yang dilakukan oleh sistem |
| 4 | *Stopword* | Stopword.m | Modul yang berfungsi untuk *stopword* |
| 5 | *Stemming* | Stemming.m | Modul yang berfungsi untuk *stemming* |
| 6 | Teks Unigram | teksUnigram.m | Modul yang berfungsi untuk membentuk kamus kata unigram |
| 7 | Teks Bigram | teksBigram.m | Modul yang berfungsi untuk membentuk kamus kata bigram |
| 8 | Teks Unigram Bigram | teksUnigramBigram.m | Modul yang berfungsi untuk membentuk kamus kata unigram+bigram |
| 9 | Vektor Unigram | vectorUnigram.m | Modul yang berfungsi untuk membentuk vektor unigram kunci jawaban dan jawaban siswa |
| 10 | Vektor Bigram | vectorBigram.m | Modul yang berfungsi untuk membentuk vektor bigram kunci jawaban dan jawaban siswa |
| 11 | Vektor Unigram Bigram | vectorUnigramBigram.m | Modul yang berfungsi untuk membentuk vektor unigram+bigram kunci jawaban dan jawaban siswa |
| 12 | *Cosine Similarity* | cosineSimilarity.m | Modul yang berfungsi untuk menghitung *similarity* dengan *cosine similarity* |
| 13 | *Jaccard Similarity* | jaccardSimilarity.m | Modul yang berfungsi untuk menghitung *similarity* dengan *jaccard similarity* |

## 6.3 Impelemntasi Antarmuka

Berikut implementasi dari halaman user yang dapat dilihat pada gambar 6.1.



**Gambar 6.1 Antarmuka Halam User**

# 7 PENGUJIAN PERANGKAT LUNAK

## 7.1 Lingkungan Pengujian

Lingkungan *hardware* yang digunakan dalam proses pengujian sistem adalah sebagai berikut:

*Processor* : Core i5 2.50 GHz

RAM : 4 GB

*Harddisk* : 500 GB

Untuk lingkungan *software* yang digunakan dalam proses pengujian sistem adalah sebagai berikut:

OS : Windows 8.1 Professional 64 bit

*Server* : XAMPP versi 3.2.1

*Tools* : Matlab 2013

## 7.2 Pelaksanaan Pengujian

**Tabel 7.1 Pelaksanaan Pengujian**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bentuk Pengujian** | **Metode Pengujian** | **Data Uji yang Digunakan** | **Keterangan** | **Hasil Akhir** |
| Pengujian implementasi kombinasi metode *term frequency* dan N-Gram | *Black Box* | Data kunci jawaban dan jawaban siswa | Menguji *return* dari kombinasi metode *term frequency* dan N-Gram | Kombinasi metode *term frequency* dan N-Gram memberikan hasil angka berupa vektor dari kunci jawaban dan jawaban siswa |
| Pengujian similarity | *Black Box* | Vektor kunci jawaban dan jawaban siswa | Menguji nilai yang didapat dari proses *similarity* | Hasil dari fungsi *similarity* memberikan nilai *similarity* antara kunci jawaban dan jawaban siswa |

## 7.3 Hasil Pengujian *Black Box*

Pengujian dilakukan berdasarkan fitur-fitur yang dimiliki dan menggunakan metode pengujian blackbox. Metode blackbox dilakukan dengan memperhatikan hasil output dari serangkaian uji coba. Uji coba dilakukan oleh peneliti sendiri. Hasil pengujian pada aplikasi SPESO adalah sebagai berikut:

**Tabel 7.2 Hasil Pengujian SPESO-F-001**

|  |  |
| --- | --- |
| Kode Kebutuhan yang Diuji | SPESO-F-001 |
| Deskripsi Pengujian | Menguji import data soal |
| Input Data | Data soal |
| Tahapan Skenario Pengujian | 1. Klik data soal 2. Pilih file data soal |
| Skenario Alternatif | Belum memilih data soal |
| Hasil yang Diharapkan | 1. Sistem membaca file data soal 2. Sistem mengambil data soal 3. Jika skenario alternatif dijalankan, menampilkan pesan error |
| Hasil Pengujian | 1. Sistem membaca file data soal 2. Sistem mengambil data soal 3. Jika skenario alternatif dijalankan, menampilkan pesan error |
| Keterangan | Berhasil |

**Tabel 7.3 Hasil Pengujian SPESO-F-002**

|  |  |
| --- | --- |
| Kode Kebutuhan yang Diuji | SPESO-F-002 |
| Deskripsi Pengujian | Menguji import data jawaban siswa |
| Input Data | Data soal |
| Tahapan Skenario Pengujian | 1. Klik data jawaban siswa 2. Pilih file data jawaban siswa |
| Skenario Alternatif | Belum memilih data jawaban siswa |
| Hasil yang Diharapkan | 1. Sistem membaca file data jawaban siswa 2. Sistem mengambil data jawaban siswa 3. Jika skenario alternatif dijalankan, menampilkan pesan error |
| Hasil Pengujian | 1. Sistem membaca file data jawaban siswa 2. Sistem mengambil data jawaban siswa 3. Jika skenario alternatif dijalankan, menampilkan pesan error |
| Keterangan | Berhasil |

**Tabel 7.4 Hasil Pengujian SPESO-F-003**

|  |  |
| --- | --- |
| Kode Kebutuhan yang Diuji | SPESO-F-003 |
| Deskripsi Pengujian | Menguji penentuan nilai |
| Input Data | Data soal dan jawaban siswa |
| Tahapan Skenario Pengujian | 1. Pilih jenis N-Gram 2. Input jumlah soal 3. Input jumlah siswa 4. Pilih file soal 5. Pilih file jawaban siswa 6. Pilih jenis *similarity* 7. Klik proses 8. Muncul nilai *similarity* |
| Skenario Alternatif | Data belum lengkap |
| Hasil yang Diharapkan | 1. Sistem membaca kebutuhan proses 2. Sistem melakukan proses berdasarkan data 3. Sistem menampilkan nilai *similarity* 4. Jika skenario alternatif dijalankan, menampilkan pesan *error* |
| Hasil Pengujian | 1. Sistem membaca kebutuhan proses 2. Sistem melakukan proses berdasarkan data 3. Sistem menampilkan nilai *similarity* 4. Jika skenario alternatif dijalankan, menampilkan pesan *error* |
| Keterangan | Berhasil |

**Tabel 7.5 Hasil Pengujian SPESO-F-004**

|  |  |
| --- | --- |
| Kode Kebutuhan yang Diuji | SPESO-F-004 |
| Deskripsi Pengujian | Menguji export nilai |
| Input Data | Data soal dan jawaban siswa |
| Tahapan Skenario Pengujian | 1. Pilih jenis N-Gram 2. Input jumlah soal 3. Input jumlah siswa 4. Pilih file soal 5. Pilih file jawaban siswa 6. Pilih jenis *similarity* 7. Klik proses 8. Muncul nilai *similarity* 9. Eksport nilai ke excel |
| Skenario Alternatif | Data belum lengkap |
| Hasil yang Diharapkan | 1. Sistem membaca kebutuhan proses 2. Sistem melakukan proses berdasarkan data 3. Sistem menampilkan nilai *similarity* 4. Sistem mengeksport hasil penilaian ke file excel 5. Jika skenario alternatif dijalankan, menampilkan pesan *error* |
| Hasil Pengujian | 1. Sistem membaca kebutuhan proses 2. Sistem melakukan proses berdasarkan data 3. Sistem menampilkan nilai *similarity* 4. Sistem mengeksport hasil penilaian ke file excel 5. Jika skenario alternatif dijalankan, menampilkan pesan *error* |
| Keterangan | Berhasil |