BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Implementasi merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis. Implementasi Aplikasi Pendeteksi Plagiarisme berbasis Web dilakukan dengan menggunakan PHP, HTML, Javascript, framework Codeignitier, dan MySQL.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk mengimplementasikan sistem ini dijelaskan pada tabel 4.1.

Tabel 4.1 Perangkat Keras yang digunakan

No.	*Perangkat* Keras	Spesifikasi Minimum	Eksistensi	Keterangan
1	Processor	Intel Pentium	Intel Core 2 duo	Memenuhi
2	RAM	512 MB	1 GB	Memenuhi
3	Hardisk	40 GB	128 GB	Memenuhi

4.1.2 Implementasi Perangkat Lunak

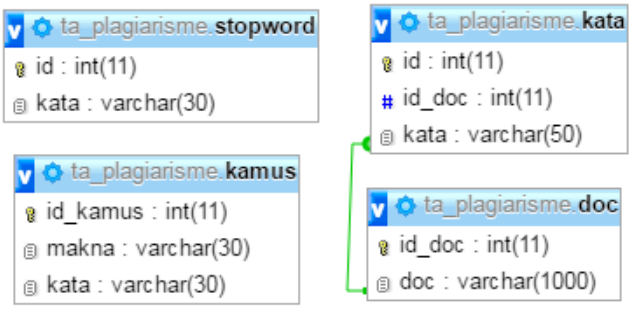
Perangkat Lunak yang digunakan dalam mengimplementasikan sistem ini dijelaskan pada tabel 4.2.

Tabel 4.2 Perangkat Lunak yang digunakan

No.	Perangkat Lunak	Keterangan
1	Windows 7	Sistem Operasi
2	XAMPP/WAMP	*Web Server*
3	MySQL	DBMS
4	Mozilla Firefox, Google Chrome, Internet Explorer	*Web Browser*

4.1.3 Implementasi Basis Data (Sintak SQL)

Implemantasi Basis Data merupakan suatu tahapan dalma proses perancangan basis data. Tahap ini merupakan implementasi dari hasil pemodelan logical dan fisikal.



1 Nama Basis Data : ta_plagiarisme

2 Tabel User

Tabel 4.3 Tabel User

Field	Type
id_user	int(11) NOT NULL
username	varchar(30) NOT NULL
password	varchar(30) NOT NULL
level	enum('Administrator','Pengguna') NOT NULL

Indexes	Columns	Index Type
PRIMARY	id_user	Unique

```
CREATE TABLE `tb_user` (
  `id_user` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(30) NOT NULL,
  `password` varchar(30) NOT NULL,
  `level` enum('Administrator','Pengguna') NOT NULL,
  PRIMARY KEY (`id_user`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1
```

3 Tabel Stopword

Tabel 4.4 Tabel Stopword

	Field	Type
🔑	id	int(11) NOT NULL
	kata	varchar(30) NOT NULL

	Indexes	Columns	Index Type
🔑	PRIMARY	id	Unique

```
CREATE TABLE `stopword` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `kata` varchar(30) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1310 DEFAULT CHARSET=latin1
```

4 Tabel Doc

Tabel 4.5 Tabel Doc

	Field	Type
🔑	id_doc	int(11) NOT NULL
	judul	varchar(1000) NOT NULL
	doc	varchar(1000) NOT NULL
	judul_2	varchar(1000) NOT NULL
	doc_2	varchar(1000) NOT NULL

	Indexes	Columns	Index Type
🔑	PRIMARY	id_doc	Unique

```
CREATE TABLE `doc` (
  `id_doc` int(11) NOT NULL AUTO_INCREMENT,
  `doc` varchar(1000) NOT NULL,
  PRIMARY KEY (`id_doc`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1
```

5 Tabel Kata

	Field	Type
🔑	id	int(11) NOT NULL
	id_doc	int(11) NOT NULL
	kata	varchar(50) NOT NULL

	Indexes	Columns	Index Type
🔑	PRIMARY	id	Unique

```
CREATE TABLE `kata` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_doc` int(11) NOT NULL,
  `kata` varchar(50) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id_doc` (`id_doc`),
  CONSTRAINT `kata_ibfk_1` FOREIGN KEY (`id_doc`) REFERENCES `doc` (`id_doc`)
) ENGINE=InnoDB AUTO_INCREMENT=1191 DEFAULT CHARSET=latin1
```

6 Tabel Kamus

	Field	Type
🔑	id_kamus	int(11) NOT NULL
	makna	varchar(30) NOT NULL
	kata	varchar(30) NOT NULL

	Indexes	Columns	Index Type
🔑	PRIMARY	id_kamus	Unique

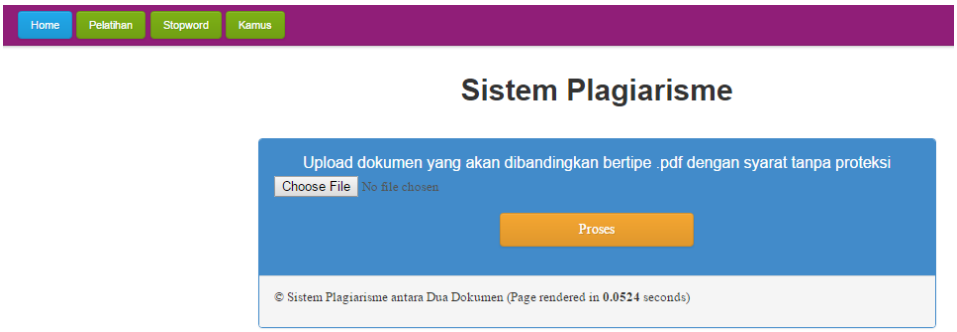
```
CREATE TABLE `kamus` (
  `id_kamus` int(11) NOT NULL AUTO_INCREMENT,
  `makna` varchar(30) NOT NULL,
  `kata` varchar(30) NOT NULL,
  PRIMARY KEY (`id_kamus`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1
```

4.1.4 Implementasi Antarmuka

Berikut ini merupakan implementasi dari setiap halaman Aplikasi Pendeteksi Plagiarisme.

4.1.4.1 Halaman Utama

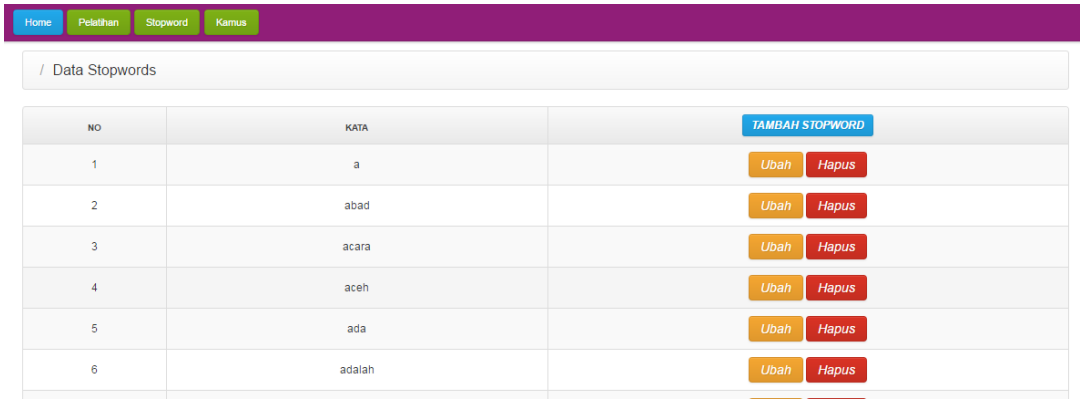
Implementasi halaman utama pada pengguna Aplikasi dapat dilihat pada gambar di bawah ini.



Gambar 4.1 Halaman Utama

4.1.4.2 Halaman Stopword

Implementasi halaman pengolahan stopword dapat dilihat pada gambar di bawah ini.



Gambar 4.2 Halaman Stopword

Pada Halaman Stopword, Admin bisa melakukan pengolahan data yang ada pada stopword.

4.1.4.3 Halaman Tabel Proses

Implementasi halaman tabel pemrosesan data dapat dilihat pada gambar di bawah ini.

HASIL TF-IDF

NO	KATA PENTING	DU	D0	D1	D2	DF	IDF	W_D0	W_D1	W_D2
1	abstrak	1	1	1	0	2	0.239	0.239	0.239	0
2	problema	1	1	1	0	2	0.239	0.239	0.239	0
3	plagiarisme	1	1	1	0	2	0.239	0.239	0.239	0
4	klasik	1	1	1	0	2	0.239	0.239	0.239	0
5	malas	1	1	1	0	2	0.239	0.239	0.239	0
6	berkreatifitas	1	1	1	0	2	0.239	0.239	0.239	0
7	berfikir	1	1	1	0	2	0.239	0.239	0.239	0
8	penelitian	0	1	1	0	2	0.239	0.239	0.239	0
9	diperoleh	0	1	1	0	2	0.239	0.239	0.239	0
10	penambahan	0	1	1	0	2	0.239	0.239	0.239	0
11	stemming	0	1	1	0	2	0.239	0.239	0.239	0
12	preprocessing	0	1	1	0	2	0.239	0.239	0.239	0
13	sistem	0	1	1	0	2	0.239	0.239	0.239	0

HASIL VSM

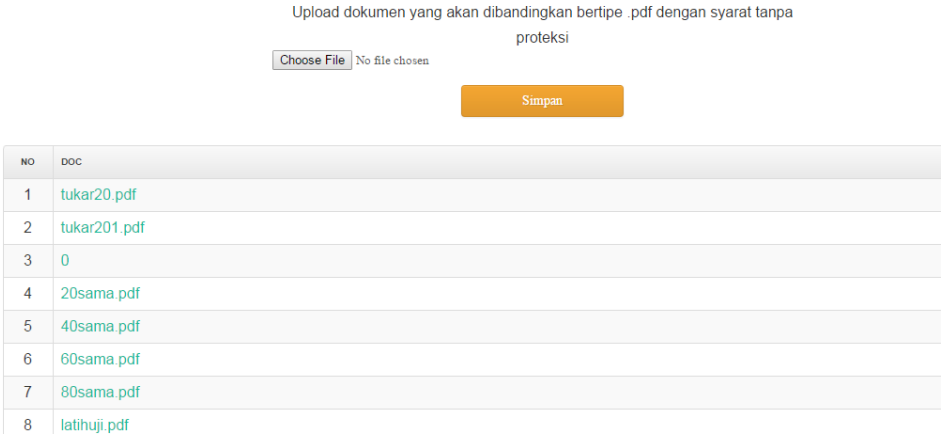
NO	KATA PENTING	DU	D0	D1	D2	DU*D0	DU*D1	DU*D2
1	abstrak	1	0.057	0.057	0	0.057	0.057	0
2	problema	1	0.057	0.057	0	0.057	0.057	0
3	plagiarisme	1	0.057	0.057	0	0.057	0.057	0
4	klasik	1	0.057	0.057	0	0.057	0.057	0
5	malas	1	0.057	0.057	0	0.057	0.057	0
6	berkreatifitas	1	0.057	0.057	0	0.057	0.057	0
7	berfikir	1	0.057	0.057	0	0.057	0.057	0
8	penelitian	0	0.057	0.057	0	0	0	0
9	diperoleh	0	0.057	0.057	0	0	0	0
10	penambahan	0	0.057	0.057	0	0	0	0
11	stemming	0	0.057	0.057	0	0	0	0
12	preprocessing	0	0.057	0.057	0	0	0	0
13	sistem	0	0.057	0.057	0	0	0	0

Gambar 4.3 Halaman Tabel Proses

Pada Halaman Tabel Proses, Pengguna Aplikasi bisa melihat pemrosesan yang dilakukan sistem terhadap dokumen yang telah diinputkan.

4.1.4.4 Halaman Data Pelatihan

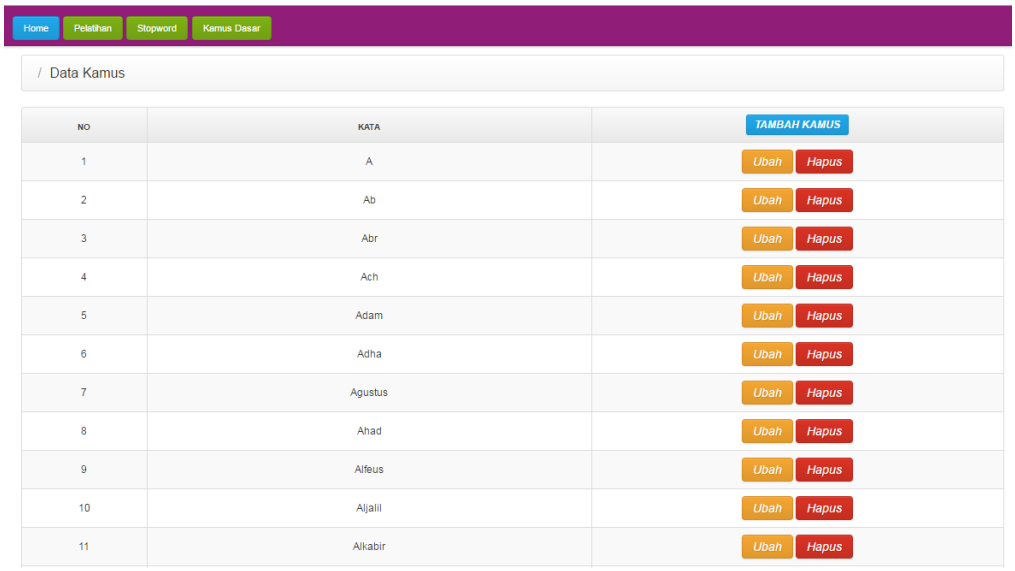
Implementasi halaman data pelatihan dapat dilihat pada gambar di bawah ini.



Gambar 4.4 Halaman Pelatihan

4.1.4.5 Halaman Kamus Dasar

Implementasi halaman data pelatihan dapat dilihat pada gambar di bawah ini.



4.2 Pengujian

Pengujian merupakan tahapan penting dalam pembangunan perangkat lunak. Pengujian dilakukan untuk mengetahui kelayakan suatu perangkat lunak sehingga perangkat lunak tersebut dapat berjalan sesuai dengan yang diharapkan dan mempunyai kualitas yang handal.

4.2.1 Rencana Pengujian

Dalam melakukan rencana pengujian pembangunan aplikasi ini menggunakan metode *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak yang dibuat. Daftar pengujian yang dilakukan dapat dilihat pada tabel berikut.

Daftar Pengujian

Item Uji	Detail Pengujian	Jenis Pengujian
Dokumen	Input data dokumen	*Black box*
Pengolahan data stopword	Tambah data stopword	
	Ubah data stopword	
	Hapus data stopword	
Pengolahan data kamus	Tambah data stopword	
	Ubah data stopword	
	Hapus data stopword	
Dokumen Latih	Input data latih	

Selain itu dilakukan pula pengujian terhadap tingkat similaritas antar dokumen. Pengujian similarity dan waktu ekseskusi ialah membandingkan hasil persentase similarity dan lama waktu antara sistem dengan *stemming*. Dokumen latih merupakan dokumen uji yang telah diberikan delapan perlakuan yaitu : 100% sama, 80% sama, 60% sama, 40% sama, 20 % sama, tukar 20%, tukar 60%, dan ubah 10%. Perlakuan tersebut merupakan pemotongan kata sebesar persentase yang disebutkan terhadap dokumen uji. Daftar pengujian yang dilakukan dapat dilihat sebagai berikut.

1. 100% sama artinya tidak mengubah sedikitpun kata yang ada dalam dokumen asli.

ABSTRAK

Problema *plagiarisme* merupakan salah satu persoalan klasik dalam dunia pendidikan sekarang ini karena seseorang bisa dengan mudah mendapatkan informasi secara mudah. Tindakan plagiarisme tersebut dapat membuat seseorang menjadi malas berkreatifitas karena tidak mau berfikir. Untuk memperkecil tindakan plagiarisme, diperlukan suatu sistem yang mampu mendeteksi plagiarisme tanpa mengabaikan perubahan posisi kalimat, pemotongan kata ataupun perubahan kata aktif menjadi pasif atau sebaliknya. Penelitian ini membuat sisitem pendeteksi plagiarisme suatu dokumen berupa *paper* Bahasa Indonesia menggunakan metode Vector Space Model (VSM). Metode VSM digunakan untuk mengukur tingkat kemiripan antara suatu dokumen dengan suatu *query*. Tujuan dari penelitian ini adalah untuk mengetahui tingkat kemiripan atau plagiarisme suatu dokumen dengan penggunaan *stemming* pada proses *preprocessing*. Pada proses *preprocessing* terdapat penambahan *stemming* yang bertujuan untuk menghilangkan imbuhan kata yang terdapat pada dokumen. *Similarity* antar dokumen didasarkan pada kecocokan berdasarkan *query*. Hasil penelitian, diperoleh bahwa dengan penambahan *stemming* dalam *preprocessing* membantu sistem dalam pengecekan *plagiarisme* dan mempermudah perhitungan nilai similaritas. Dengan hasil penelitian tersebut, maka penulis menyarankan bagi penelitian selanjutnya untuk meneliti mengenai penggunaan *stemming* lebih lagi pada tindak plagiarisme.

2. 80% sama artinya memiliki kemiripan dengan melakukan pemotongan dari dokumen asli secara *random.*

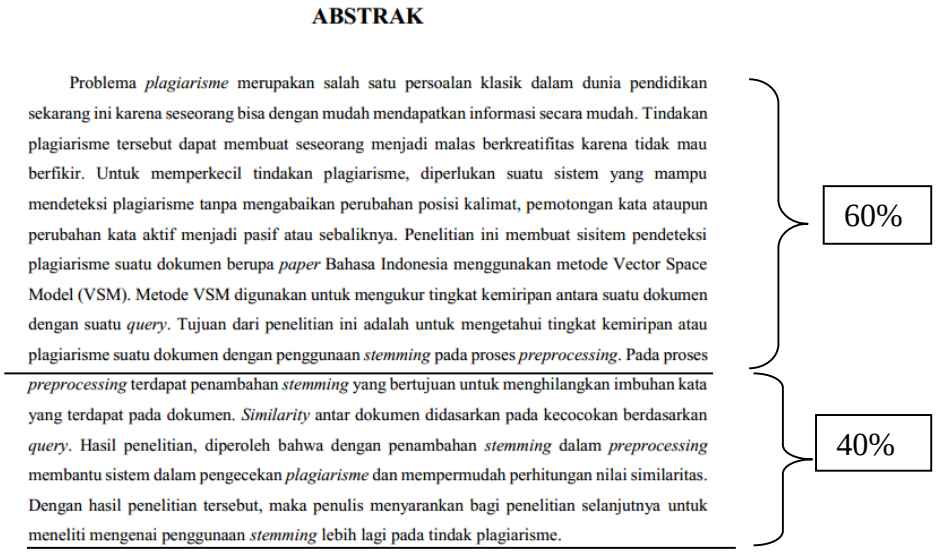
ABSTRAK

Problema *plagiarisme* merupakan salah satu persoalan klasik dalam dunia pendidikan sekarang ini karena seseorang bisa dengan mudah mendapatkan informasi secara mudah. Tindakan plagiarisme tersebut dapat membuat seseorang menjadi malas berkreatifitas karena tidak mau berfikir. Untuk memperkecil tindakan plagiarisme, diperlukan suatu sistem yang mampu mendeteksi plagiarisme tanpa mengabaikan perubahan posisi kalimat, pemotongan kata ataupun perubahan kata aktif menjadi pasif atau sebaliknya. Penelitian ini membuat sisitem pendeteksi plagiarisme suatu dokumen berupa *paper* Bahasa Indonesia menggunakan metode Vector Space Model (VSM). Metode VSM digunakan untuk mengukur tingkat kemiripan antara suatu dokumen dengan suatu *query*. Tujuan dari penelitian ini adalah untuk mengetahui tingkat kemiripan atau plagiarisme suatu dokumen dengan penggunaan *stemming* pada proses *preprocessing*. Pada proses *preprocessing* terdapat penambahan *stemming* yang bertujuan untuk menghilangkan imbuhan kata yang terdapat pada dokumen. *Similarity* antar dokumen didasarkan pada kecocokan berdasarkan *query*. Hasil penelitian, diperoleh bahwa dengan penambahan *stemming* dalam *preprocessing* membantu sistem dalam pengecekan *plagiarisme* dan mempermudah perhitungan nilai similaritas. Dengan hasil penelitian tersebut, maka penulis menyarankan bagi penelitian selanjutnya untuk meneliti mengenai penggunaan *stemming* lebih lagi pada tindak plagiarisme.

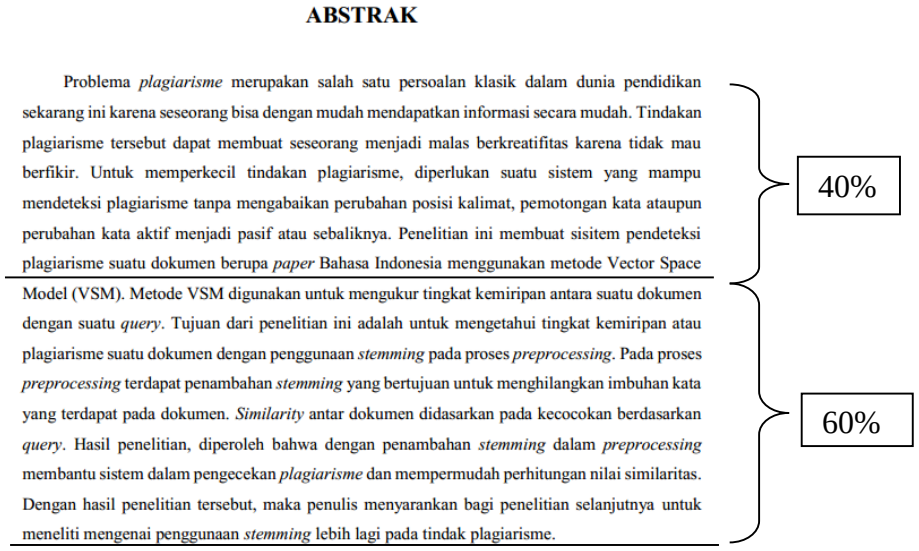
80%

20%

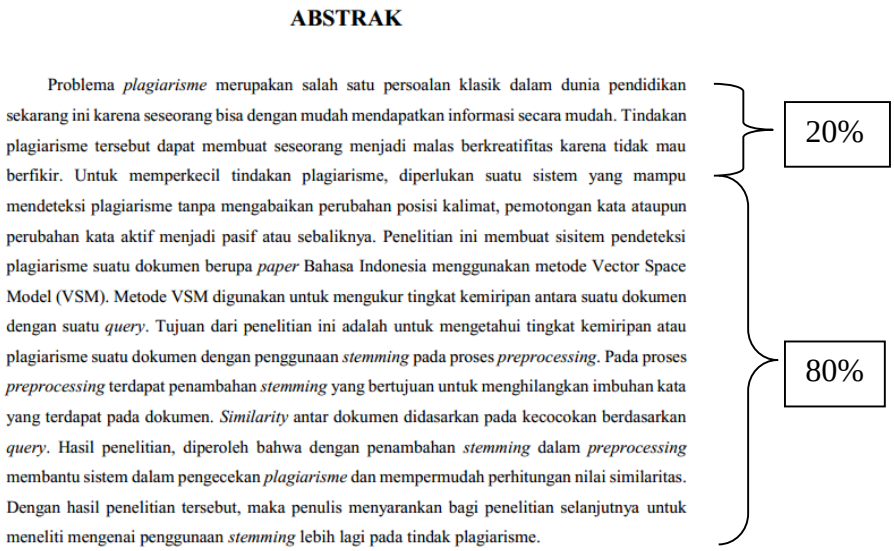
3. 60% sama artinya memiliki kemiripan dengan melakukan pemotongan dari dokumen asli secara *random*.



4. 40% sama artinya memiliki kemiripan dengan melakukan pemotongan dari dokumen asli secara *random*.



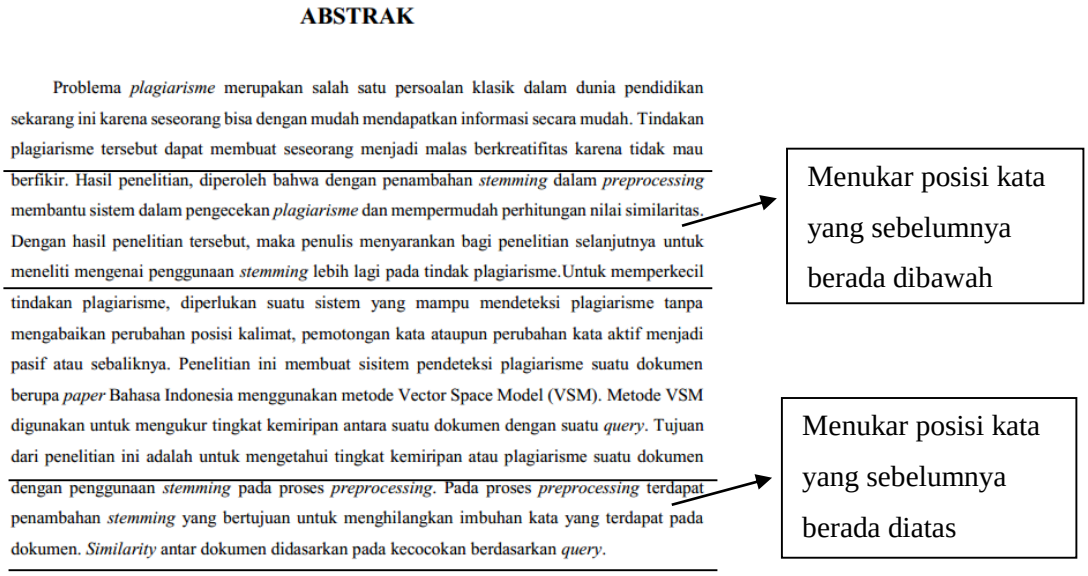
5. 20% sama artinya memiliki kemiripan dengan melakukan pemotongan dari dokumen asli secara *random*.



ABSTRAK

Problema *plagiarisme* merupakan salah satu persoalan klasik dalam dunia pendidikan sekarang ini karena seseorang bisa dengan mudah mendapatkan informasi secara mudah. Tindakan plagiarisme tersebut dapat membuat seseorang menjadi malas berkreatifitas karena tidak mau berfikir. Untuk memperkecil tindakan plagiarisme, diperlukan suatu sistem yang mampu mendeteksi plagiarisme tanpa mengabaikan perubahan posisi kalimat, pemotongan kata ataupun perubahan kata aktif menjadi pasif atau sebaliknya. Penelitian ini membuat sisitem pendeteksi plagiarisme suatu dokumen berupa *paper* Bahasa Indonesia menggunakan metode Vector Space Model (VSM). Metode VSM digunakan untuk mengukur tingkat kemiripan antara suatu dokumen dengan suatu *query*. Tujuan dari penelitian ini adalah untuk mengetahui tingkat kemiripan atau plagiarisme suatu dokumen dengan penggunaan *stemming* pada proses *preprocessing*. Pada proses *preprocessing* terdapat penambahan *stemming* yang bertujuan untuk menghilangkan imbuhan kata yang terdapat pada dokumen. *Similarity* antar dokumen didasarkan pada kecocokan berdasarkan *query*. Hasil penelitian, diperoleh bahwa dengan penambahan *stemming* dalam *preprocessing* membantu sistem dalam pengecekan *plagiarisme* dan mempermudah perhitungan nilai similaritas. Dengan hasil penelitian tersebut, maka penulis menyarankan bagi penelitian selanjutnya untuk meneliti mengenai penggunaan *stemming* lebih lagi pada tindak plagiarisme.

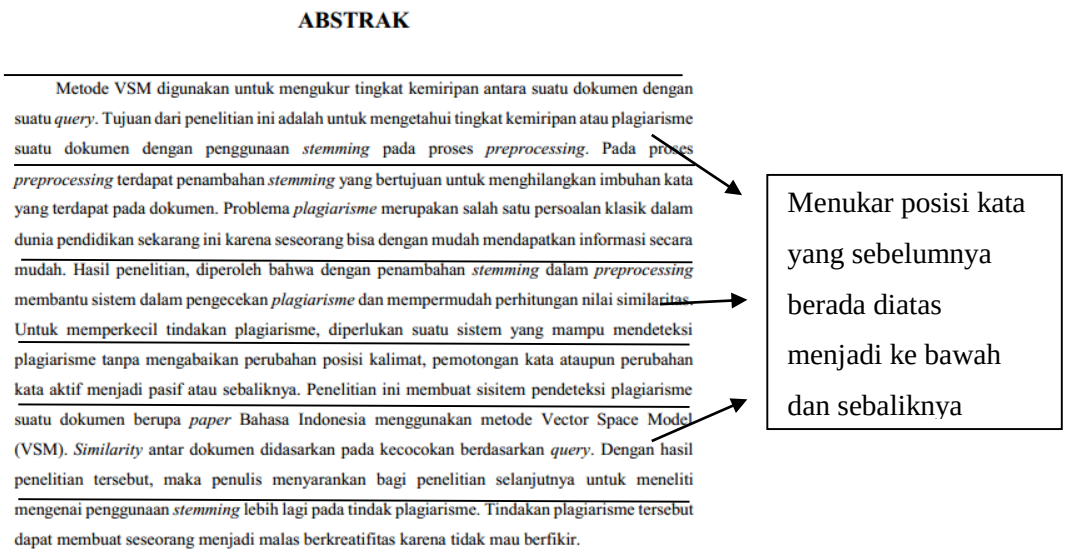
6. Tukar 20% artinya dengan melakukan penukaran posisi kata dari dokumen asli secara *random.*



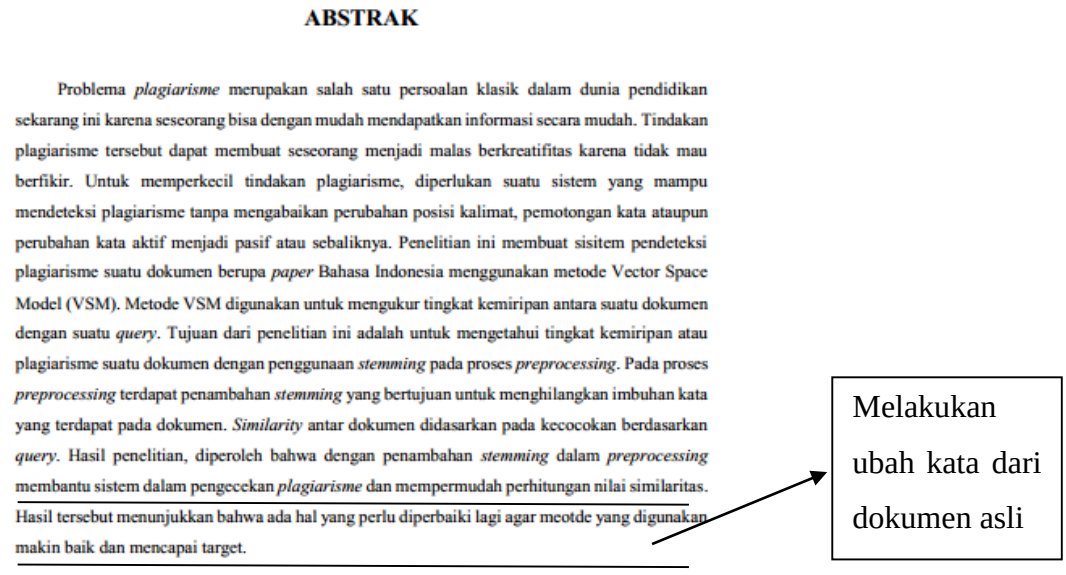
ABSTRAK

Problema *plagiarisme* merupakan salah satu persoalan klasik dalam dunia pendidikan sekarang ini karena seseorang bisa dengan mudah mendapatkan informasi secara mudah. Tindakan plagiarisme tersebut dapat membuat seseorang menjadi malas berkreatifitas karena tidak mau berfikir. Hasil penelitian, diperoleh bahwa dengan penambahan *stemming* dalam *preprocessing* membantu sistem dalam pengecekan *plagiarisme* dan mempermudah perhitungan nilai similaritas. Dengan hasil penelitian tersebut, maka penulis menyarankan bagi penelitian selanjutnya untuk meneliti mengenai penggunaan *stemming* lebih lagi pada tindak plagiarisme.Untuk memperkecil tindakan plagiarisme, diperlukan suatu sistem yang mampu mendeteksi plagiarisme tanpa mengabaikan perubahan posisi kalimat, pemotongan kata ataupun perubahan kata aktif menjadi pasif atau sebaliknya. Penelitian ini membuat sisitem pendeteksi plagiarisme suatu dokumen berupa *paper* Bahasa Indonesia menggunakan metode Vector Space Model (VSM). Metode VSM digunakan untuk mengukur tingkat kemiripan antara suatu dokumen dengan suatu *query*. Tujuan dari penelitian ini adalah untuk mengetahui tingkat kemiripan atau plagiarisme suatu dokumen dengan penggunaan *stemming* pada proses *preprocessing*. Pada proses *preprocessing* terdapat penambahan *stemming* yang bertujuan untuk menghilangkan imbuhan kata yang terdapat pada dokumen. *Similarity* antar dokumen didasarkan pada kecocokan berdasarkan *query*.

7. Tukar 60% artinya dengan melakukan penukaran posisi kata secara *random.*



8. Ubah 10% artinya dengan melakukan ubah kata secara *random*.



4.2.2 Kasus dan Hasil Pengujian

Kasus dan hasil pengujian dibuat untuk mengetahui apakah Aplikasi Pendeteksi Plagiarisme dapat berjalan dengan baik atau tidak. Berdasarkan rencana pengujian, maka dapat dilakukan pengujian adalah sebagai berikut:

4.2.2.1 Pengujian Proses Dokumen

Dalam pengujian ini dilakukan pengujian terhadap inputan file dokumen yang dimasukkan ke dalam aplikasi.

Tabel 4.6 Pengujian Dokumen

Kasus dan hasil uji (data benar)			
Data masukkan	Yang diharapkan	Pengamatan	Kesimpulan
- klik tombol Choose File dan masukan file lalu klik tombol proses	File inputan berupa dokumen dengan format .pdf dan di simpan dalam tabel Doc.	File inputan berupa dokumen dengan format .pdf dan di simpan dalam tabel Doc.	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
- klik tombol Choose File dan masukan file lalu klik tombol proses	Akan terjadi error karena data dokumen tidak sesuai.	Akan terjadi error karena data dokumen tidak sesuai.	[√] Diterima [ ] Ditolak

4.2.2.2 Pengujian Proses Pengolahan Data Stopword

Dalam pengujian ini dilakukan pengujian terhadap halaman stopword dimana pengguna akan melakukan pengolahan data di dalamnya.

Tabel 4.7 Pengujian Tambah Data Stopword

Kasus dan hasil uji (data benar)			
Data masukkan	Yang diharapkan	Pengamatan	Kesimpulan
- Kata : saya	Data tersebut di simpan ke dalam tabel stopword.	Data tersebut di simpan ke dalam tabel stopword.	[√] Diterima [ ] Ditolak

Tabel 4.8 Pengujian Ubah Data Stopword

Kasus dan hasil uji (data benar)			
Data masukkan	Yang diharapkan	Pengamatan	Kesimpulan
- Memilih data yang akan diubah	Data tersebut berhasil diubah dan di simpan ke dalam tabel stopword.	Data tersebut berhasil diubah dan di simpan ke dalam tabel stopword.	[√] Diterima [ ] Ditolak

Tabel 4.9 Pengujian Hapus Data Stopword

Kasus dan hasil uji (data benar)

Data masukkan	Yang diharapkan	Pengamatan	Kesimpulan
- Memilih data yang akan dihapus	Menampilkan konfirmasi hapus data	Menampilkan konfirmasi hapus data	[√] Diterima [ ] Ditolak

4.2.2.3 Pengujian Proses Pengolahan Data Stopword

Dalam pengujian ini dilakukan pengujian terhadap halaman kamus dasar dimana pengguna akan melakukan pengolahan data di dalamnya.

Tabel 4.10 Pengujian Tambah Data Kamus

Kasus dan hasil uji (data benar)			
Data masukkan	Yang diharapkan	Pengamatan	Kesimpulan
- Kata : saya	Data tersebut di simpan ke dalam tabel kamus.	Data tersebut di simpan ke dalam tabel kamus.	[√] Diterima [ ] Ditolak

Tabel 4.11 Pengujian Ubah Data Kamus

Kasus dan hasil uji (data benar)			
Data masukkan	Yang diharapkan	Pengamatan	Kesimpulan
- Memilih data yang akan diubah	Data tersebut berhasil diubah dan di simpan ke dalam tabel kamus.	Data tersebut berhasil diubah dan di simpan ke dalam tabel kamus.	[√] Diterima [ ] Ditolak

Tabel 4.12 Pengujian Hapus Data Kamus

Kasus dan hasil uji (data benar)			
Data masukkan	Yang diharapkan	Pengamatan	Kesimpulan
- Memilih data yang akan dihapus	Menampilkan konfirmasi hapus data	Menampilkan konfirmasi hapus data	[√] Diterima [ ] Ditolak

4.2.2.4 Pengujian Proses Dokumen Latih

Dalam pengujian ini dilakukan pengujian terhadap halaman inputan dokumen latih dimana pengguna akan melakukan pengolahan data di dalamnya.

Tabel 4.13 Pengujian Data Latih

Kasus dan hasil uji (data benar)			
Data masukkan	Yang diharapkan	Pengamatan	Kesimpulan
- klik tombol Choose File dan masukan file lalu klik tombol simpan	File inputan berupa dokumen dengan format .pdf dan di simpan dalam tabel Doc.	File inputan berupa dokumen dengan format .pdf dan di simpan dalam tabel Doc.	[√] Diterima [ ] Ditolak
Kasus dan hasil uji (data salah)			
- klik tombol Choose File dan masukan file lalu klik tombol simpan	Akan terjadi error karena data dokumen tidak sesuai.	Akan terjadi error karena data dokumen tidak sesuai.	[√] Diterima [ ] Ditolak

4.2.3 Kesimpulan Hasil Pengujian Black Box

Berdasarkan hasil pengujian dengan kasus uji di atas, dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

4.2.4 Pengujian Penghitungan Tingkat Similaritas

Pada pengujian tingkat similaritas akan dihitung nilai similaritas dari dokumen yang diupload oleh dokumen secara lengkap dapat dilihat pada halaman lampiran A-1. Dari dokumen tersebut dapat diambil hasil sebagai berikut. pengguna dan nantinya akan dibandingkan dengan dokumen latih yang ada dalam database. Setelah hasil similarity didapatkan selanjutnya akan diurutkan berdasarkan tingkat similaritas dokumen dari yang terbesar ke terkecil. Daftar dari

- Total Dokumen : 3
- Nilai Similarity antara dokumen uji dengan dokumen latih.

Dokumen Uji : jurnal_13423.pdf

a. Dokumen Latih 1 : 21tudesmanJurnal1.pdf
b. Dokumen Latih 2 : DR00098201306_FIN_(1).pdf
c. Dokumen Latih 3 : SNETE-2013.pdf

- Nilai Persentase Similarity = (Similaritas x 100 %)
- Nilai Persentase Similarity :
  a. Dokumen Latih 1 : 0,156 x 100 % = 15,6 %
  b. Dokumen Latih 2 : 0,146 x 100 % = 14,6 %
  c. Dokumen Latih 3 : 0,152 x 100 % = 15,2 %

Setelah dilakukan pengujian, nilai similaritas yang mendekati dengan dokumen uji adalah sokumen latih 1 sebesar 15,6 %.

4.2.5 Hasil Pengujian *Similarity* dan Waktu Eksekusi

Hasil pengujian dibawah ini adalah hasil yang sudah dilakukan dengan melakukan tahapan seperti pada rencana pengujian tingkat similaritas.

Tabel 4.14 Hasil Pengujian *Similarity* dan Waktu Eksekusi

Dokumen	Rata-rata Persentase Similarity (%)	Rata-rata waktu (s)
100% sama	13,8	0,6233
80% sama	11,7	0,6249
60 % sama	10,5	0,7305
40% sama	6,4	0,6268
20% sama	2,7	0,5532
Tukar 20 %	11,8	0,7474
Tukar 60 %	10,8	0,9765
Ubah 10 %	8,9	0,9117

Pada pengujian yang sudah dilakukan menunjukkan bahwa hasil persentase tidak persis sebesar pemotongan kata yang sudah dilakukan, hal ini dikarenakan dokumen mengalami proses pemotongan kata secara *random* , dalam kasus ini tidak hanya kata penting saja yang terpotong melainkan terdapat kemungkinan kata yang dihilangkan merupakan kata *stopword.*

4.2.6 Analisis Hasil Pengujian

Berikut adalah analisis dari pengujian tingkat similaritas yang sudah dilakukan terhadap dokumen latih yang ada dalam database.

1. Banyaknya dokumen dalam database akan mempengaruhi seberapa banyak perbandingan antara dokumen uji dan dokumen latih.
2. Banyak kata penting pada dokumen berpengaruh pada hasil similaritas.
3. Sistem deteksi plagiarisme ini memproses dokumen dengan dokumen yang sudah terintegrasi sebelumnya.
4. Tingkat similaritas antara dokumen uji dengan dokumen latih yang tertera mendapatkan hasil persentase yang tidak terlalu jauh berbeda.
5. Hasil pengujian menunjukkan bahwa rata-rata dari persentase similarity yaitu sebesar 9,575 %.