

SISTEM PENDETEKSI PENULISAN KATA PADA DOKUMEN BERBASIS WEB

¹ Andika Saputra, ² Dwi Sakethi & ³Yohana Tri Utami

^{1,2,3}Jurusan Ilmu Komputer FMIPA Universitas Lampung

Jalan Prof. Sumantri Brojonegoro No. 1 Bandar Lampung 35145

¹andika.saputra1544@students.unila.ac.id, ²dwijim@fmipa.unila.ac.id, ³yohana.utami@fmipa.unila.ac.id

Abstract— *The process of detecting writing errors manually takes a lot of time and requires a source of reference to prove the truth of writing words. The efficiency of time required if done manually certainly will not be optimal, so we need a word writing error detection utility that is expected to make it easier to check errors in a document of type word (.docx). In the process of developing this utility through several stages of the process, namely Cleansing, Case Folding, Tokenization, Stemming and Dictionary Lookuup. The result of the docx document being processed is influenced by the completeness of the dictionary used as a reference for word correction and the type of word spelling errors.*

Keywords: *Case Folding; Cleansing; Stemming; Tokenization; Utility;*

1. PENDAHULUAN

Menulis merupakan salah satu keterampilan dan kebiasaan manusia dalam kehidupan sehari-hari bahkan dalam dunia pendidikan. Menulis juga salah satu media komunikasi dalam bentuk teks (pesan) dengan menggunakan bahasa serta penulisan yang baik. Bahasa menjadi perihal penting untuk menyampaikan pesan, bahasa akan dituangkan dalam sebuah tulisan yang biasa disebut sebuah kata dan kalimat. Kalimat akan menjadi penting karena, difungsikan sebagai media untuk menyampaikan pesan. Kalimat yang salah akan menimbulkan makna, arti dan pemahaman berbeda terhadap penerima atau pembaca kalimat [1].

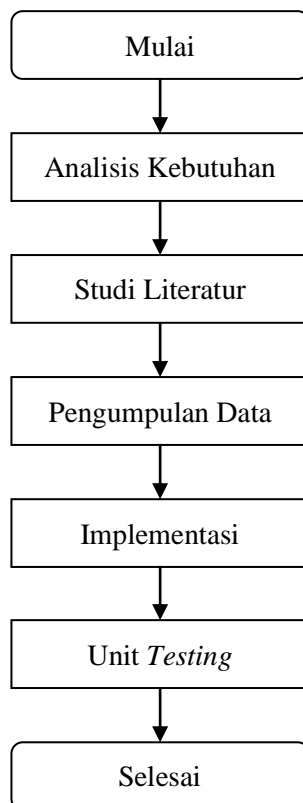
Kalimat maupun tulisan dapat dikatakan layak apabila telah memenuhi Pedoman Umum Ejaan Bahasa Indonesia (PUEBI) dan Kamus Besar Bahasa Indonesia (KBBI). Banyak ditemukan kesalahan penulisan (*typographical error*) yaitu salah dalam pengetikan karena menekan tombol pada *keyboard*, jarak antara tombol-tombol yang berdekatan dan kecepatan perpindahan jari yang tidak seimbang untuk merangkai kata. Akhirnya membuat seseorang salah paham dengan maksud kalimat yang dihasilkan dalam suatu dokumen seperti karya ilmiah dan skripsi[2].

Proses deteksi kesalahan pengetikan dengan cara manual akan menghabiskan banyak waktu dan membutuhkan suatu sumber pasti sebagai acuan bahwa kata tersebut memang salah dalam proses penulisannya. Sistem untuk identifikasi kesalahan pengetikan ini merupakan sistem berbasis web yang dapat digunakan oleh masyarakat luas, sehingga jika terdapat kata yang tidak sesuai dengan kamus dapat terdeteksi sebagai kata salah dan sistem akan menampilkan semua kata dengan diberi keterangan benar dan salah ketik [3]. Berdasarkan hal tersebut, diperlukan sistem deteksi untuk melakukan identifikasi kesalahan pengetikan pada teks dan melakukan proses koreksi terhadap kesalahan pengetikan [4].

Pada penelitian Rahmad (2019), dimana pada penelitian tersebut dilakukan pengembangan *utility* pendeteksi kesalahan penulisan kata pada kata dasar dan dokumen bertipe word (.doc). Pada penelitian ini akan dilakukan pengembangan pendeteksi kesalahan penulisan kata dasar dan kata berimbuhan (di-) dilakukan pada dokumen bertipe word (.docx).

2. METODOLOGI PENELITIAN

Prosedur penelitian yang dilakukan pada penelitian ini terdiri dari beberapa tahap yang dapat dilihat pada Gambar 1.



Gambar 1 Tahapan Penelitian.

Berdasarkan pada Gambar 1 berikut ini penjelasan dari beberapa tahapan yang akan dilakukan pada saat penelitian yaitu:

1. Analisis kebutuhan, dalam tahapan ini merupakan tahapan awal peneliti untuk menganalisis kebutuhan yang nantinya dibutuhkan dalam penelitian.
2. Studi literatur, dalam tahapan ini yaitu sumber pengumpulan data seperti jurnal, buku, internet.
3. Pengumpulan data, dalam tahapan ini dilakukan pengumpulan data untuk penelitian.
4. Implementasi, pada tahap ini data atau informasi kebutuhan penelitian yang telah di peroleh diterapkan pada pengembangan utility sistem.
5. *Unit Testing* merupakan tahapan yang dilakukan setelah implementasi selesai dilakukan dan pada tahapan ini dilakukan *testing* dari fungsi-fungsi dalam *utility*.

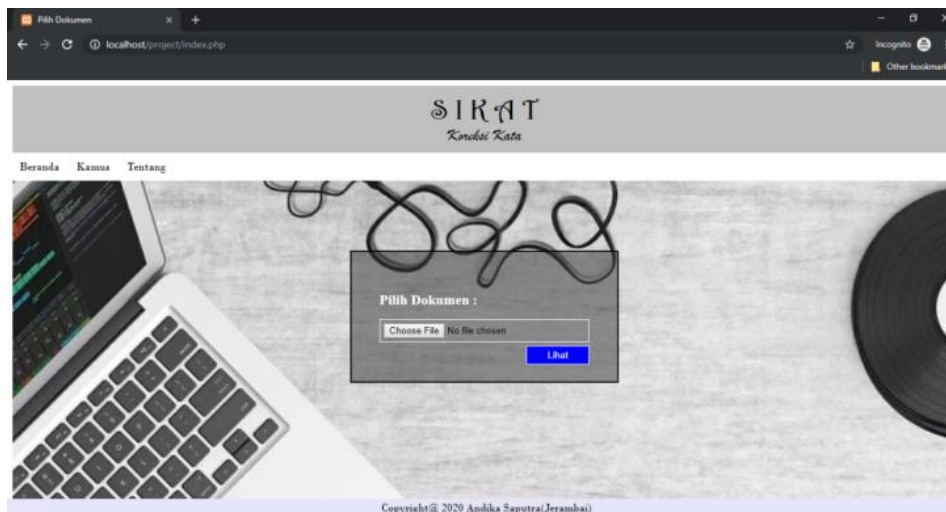
3. HASIL DAN PEMBAHASAN

3.1 Implementasi Tampilan Antarmuka

Antarmuka merupakan serangkaian tampilan grafis yang dapat dimengerti oleh pengguna komputer dan diprogram sedemikian rupa sehingga dapat terbaca oleh sistem operasi komputer dan beroperasi sebagaimana mestinya[5]. Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun, berikut ini adalah implementasi antarmuka *utility* deteksi penulisan kata dasar dan kata berimbuhan di- berdasarkan kamus pada dokumen bertipe word (.docx) yang diberi nama “SIKAT” (Koreksi Kata), berikut ini merupakan implementasi tampilan antarmuka yang dijelaskan secara terperinci.

3.1.1 Halaman Utama

Pada tampilan halaman utama ini terdapat 3 menu bar yaitu beranda, kamus dan tentang. Halaman utama ini merupakan tampilan awal *utility*, dimana terdapat sebuah *form upload* yang berfungsi untuk mengunggah dokumen bertipe word (.docx) yang akan diproses untuk mendeteksi penulisan kata berdasarkan kamus kata dasar dan kata berimbuhan di- untuk lebih jelasnya dapat dilihat pada Gambar 2.

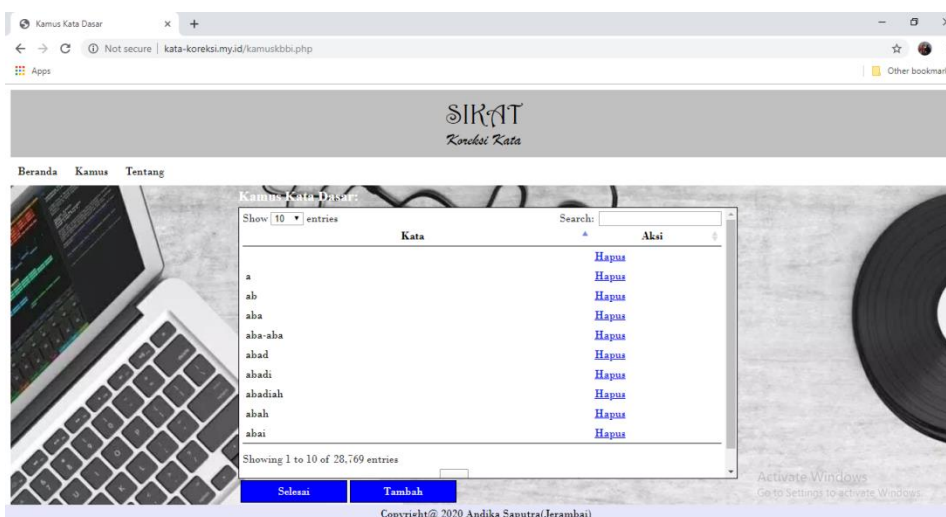


Gambar 2 Halaman Utama.

3.1.2 Halaman Kamus

Pada halaman kamus menampilkan kamus dalam bentuk tabel, dimana terdapat dua kolom yaitu kata dan aksi. Halaman kamus terdiri menjadi dua bagian yaitu halaman kamus kata dasar dan kamus keterangan tempat.

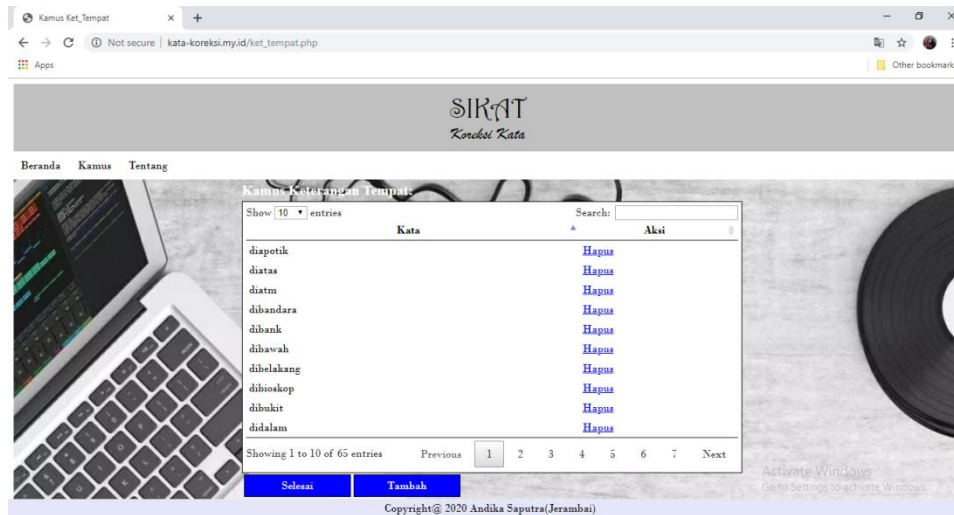
Pertama kamus kata dasar pada kamus kata dasar terdapat dua kolom yaitu kata dan aksi, pada kolom kata menampilkan kata dasar dan pada kolom aksi terdapat *button* (hapus) untuk menghapus kata dasar yang dianggap tidak sesuai, halaman kamus kata dasar ini dapat dilihat pada Gambar 3.



Gambar 3 Halaman Kamus Kata Dasar.

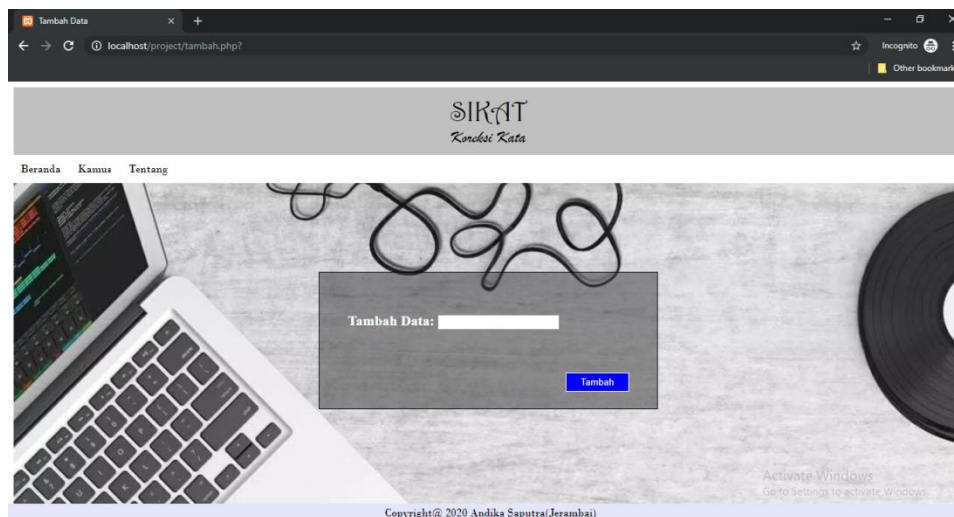
Halaman berikutnya yaitu halaman kamus keterangan tempat dimana terdapat dua kolom yaitu kata dan aksi sama halnya dengan kamus kata dasar kolom kata untuk menampilkan kata keterangan tempat dan

aksi untuk *button* (hapus) yang digunakan untuk menghapus kata yang tidak sesuai. Halaman kamus keterangan tempat ini dapat dilihat pada Gambar 4.



Gambar 4 Halaman Kamus Keterangan Tempat.

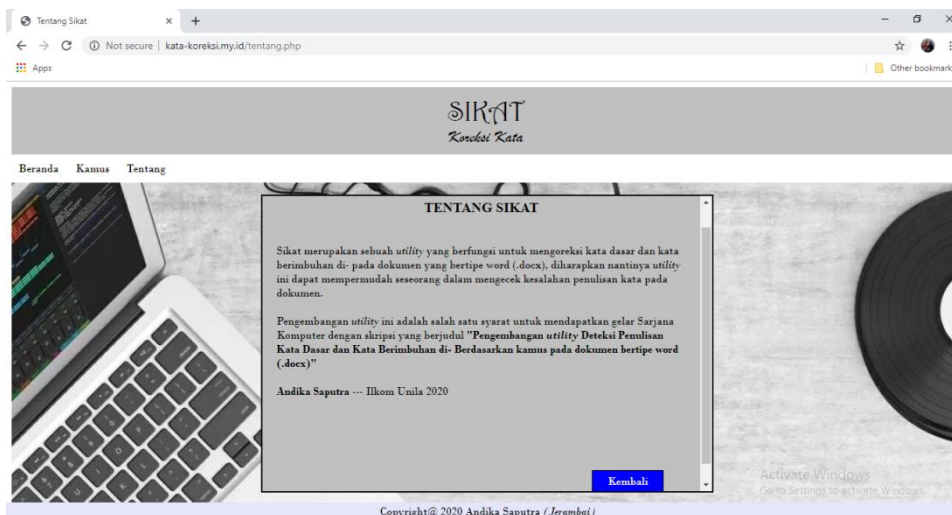
Halaman tambah data kamus berfungsi untuk menambahkan kata yang belum ada di dalam kamus, pada halaman ini terdapat *form* untuk memasukkan data yang akan ditambahkan, sesudah data dimasukkan pada *form* yang tersedia selanjutnya pilih *button* tambah maka data akan tersimpan pada *database* kamus. Halaman ini dapat dilihat pada Gambar 5.



Gambar 5 Halaman Tambah Data Kamus.

3.1.3 Halaman Tentang

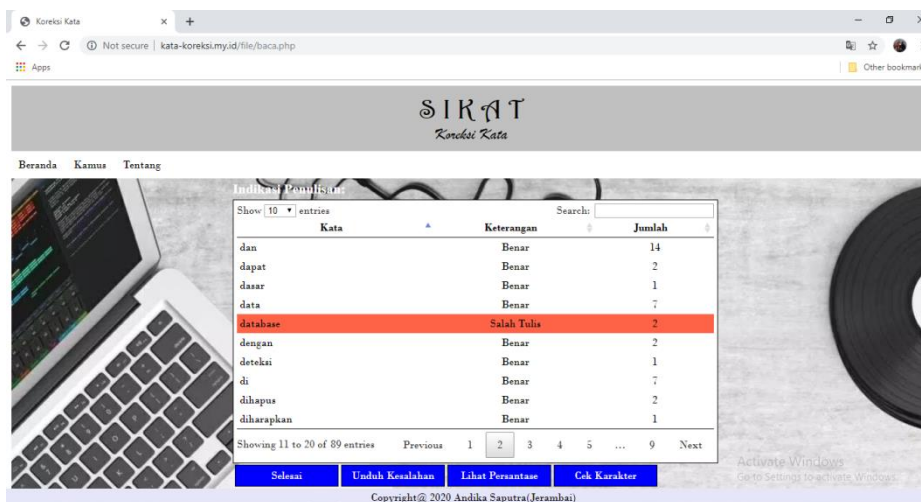
Pada halaman tentang menampilkan informasi singkat mengenai *utility*. tampilan halaman ini dapat dilihat pada Gambar 6.



Gambar 6 Halaman Tentang

3.1.4 Halaman Koreksi Penulisan

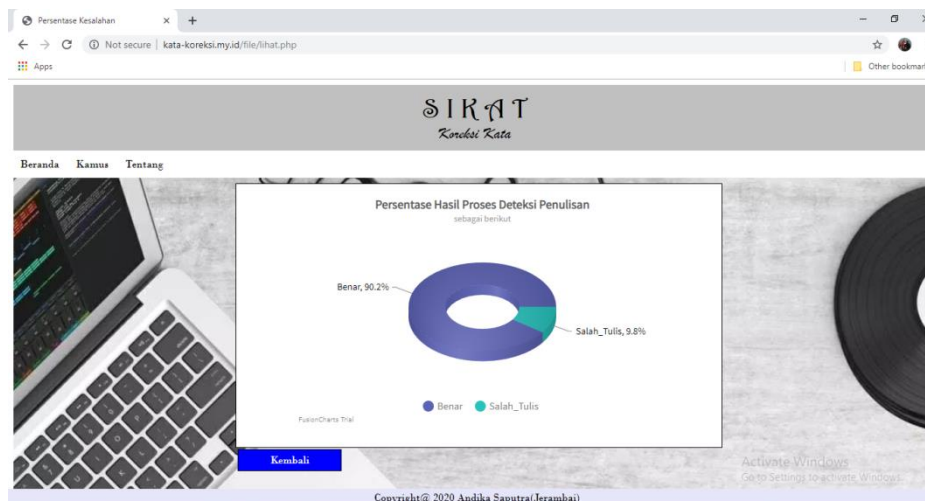
Pada halaman koreksi penulisan menampilkan hasil dokumen yang telah dikoreksi penulisannya dan data disajikan dalam tabel dimana tabel terdiri dari kolom kata yang menampilkan kata yang ada pada dokumen, kolom keterangan yang menunjukkan kata yang ada benar atau salah tulis, dan kolom keterangan yang menampilkan jumlah perkata yang ada dalam dokumen, untuk tampilan halaman ini dapat dilihat pada Gambar 7.



Gambar 7 Halaman Koreksi Penulisan

3.1.5 Halaman Persentase Penulisan

Pada halaman persentase penulisan menampilkan hasil koreksi kata dalam bentuk grafik dimana dibagi menjadi dua aspek yaitu kata benar dan kata yang salah tulis, grafik yang disajikan dapat dilihat pada Gambar 8.



Gambar 8 Halaman Persentase Penulisan

3.2 Hasil Pengujian *Utility*

Pada proses pengujian *utility* ini dilakukan dengan dua tahap yaitu pengujian fungsional dan pengujian waktu pemrosesan. Berikut penjelasan dan hasil pengujian dari masing-masing tahap tersebut.

3.2.1 Pengujian Fungsional

Pengujian fungsional dilakukan dengan menggunakan metode *black box*. *Black box testing* merupakan pengujian suatu aplikasi yang membahas sisi luar suatu aplikasi perangkat lunak, dimana mulai dari tampilan hingga akses *input-an* (masukan), dalam strategi *testing*[6] pengujian fungsional ini melibatkan beberapa mahasiswa yang dilakukan dengan menggunakan kuesioner. Berikut adalah hasil dari pengujian tersebut yang disajikan dalam beberapa macam *test scenario*.

Tabel 1 Test Scenario: Halaman Utama

Kasus uji data benar	
1. Fungsi	Halaman untuk <i>upload</i> dokumen
Skenario Pengujian	Memilih dokumen bertipe word (.docx) untuk di <i>upload</i>
Yang diharapkan	Dokumen di <i>upload</i> dan diproses untuk mengecek penulisan kata
Pengamatan	Dokumen ter <i>upload</i> dan diproses
Kesimpulan	Diterima
Kasus uji data salah	
2. Fungsi	Halaman untuk <i>upload</i> dokumen
Skenario Pengujian	Memilih dokumen yang bertipe selain word (.docx) untuk di <i>upload</i>
Yang diharapkan	Dokumen tidak ter <i>upload</i> dan tidak di proses, kembali ke halaman utama
Pengamatan	Dokumen tidak ter <i>upload</i> dan diproses, kembali kehalaman utama
Kesimpulan	Diterima

Tabel 2 Test Scenario: Halaman Tambah Data

Kasus uji data benar		
1.	Fungsi	Menambahkan data ke dalam kamus
	Skenario Pengujian	Menginputkan kata baru ke dalam <i>form</i> kemudian di tambahkan ke kamus
	Yang diharapkan	Kata di tambah dan disimpan ke dalam <i>database</i> kamus
	Pengamatan	Kata yang diinputkan di <i>form</i> ditambah dan disimpan dalam <i>database</i> kamus
	Kesimpulan	Berhasil
Kasus uji data salah		
2.	Fungsi	Menambahkan data ke dalam kamus
	Skenario Pengujian	Menginputkan kata yang sudah ada ke dalam <i>form</i> untuk di tambahkan ke <i>database</i> kamus
	Yang diharapkan	Kata tidak ditambahkan dan muncul notifikasi kata sudah tersedia
	Pengamatan	Kata tidak ditambahkan dan muncul notifikasi kata sudah tersedia
	Kesimpulan	Berhasil

Tabel 3 Test Scenario: Halaman Koreksi Penulisan

Kasus uji data benar		
1.	Fungsi	Halaman untuk menampilkan hasil data yang telah diproses
	Skenario Pengujian	Data word bertipe (.docx) diproses
	Yang diharapkan	Halaman dapat menampilkan data dalam bentuk tabel,dimana dalam tabel terdapat kata, jumlah kata, dan keterangan kata tersebut
	Pengamatan	Halaman menampilkan data dalam bentuk tabel,dimana dalam tabel terdapat kata, jumlah kata, dan keterangan kata tersebut
	Kesimpulan	Berhasil
Kasus uji data salah		
2.	Fungsi	Halaman untuk menampilkan hasil data yang telah diproses
	Skenario Pengujian	Data word bertipe (.docx) diproses
	Yang diharapkan	Halaman dapat menampilkan data dalam bentuk tabel,dimana dalam tabel terdapat kata, jumlah kata, dan keterangan kata tersebut
	Pengamatan	Halaman menampilkan data dalam bentuk tabel,dimana dalam tabel terdapat kata, jumlah kata, dan keterangan kata tersebut
	Kesimpulan	Berhasil

Tabel 4 Test Scenario: Halaman Persentase Penulisan

Kasus uji data benar		
1.	Fungsi	Menampilkan hasil pendeteksi kata dalam grafik persentase
	Skenario Pengujian	Data word (.docx) yang telah diperoleh hasil pendeteksiannya diproses untuk ditampilkan dalam bentuk grafik persentase
	Yang diharapkan	Halaman dapat menampilkan hasil pendeteksi kata dalam bentuk grafik persentase sesuai dengan kata benar dan kata salah ketik
	Pengamatan	Halaman menampilkan hasil pendeteksi kata dalam bentuk grafik persentase sesuai dengan kata benar dan kata salah ketik
	Kesimpulan	Berhasil

3.2.2 Pengujian Waktu Pemrosesan

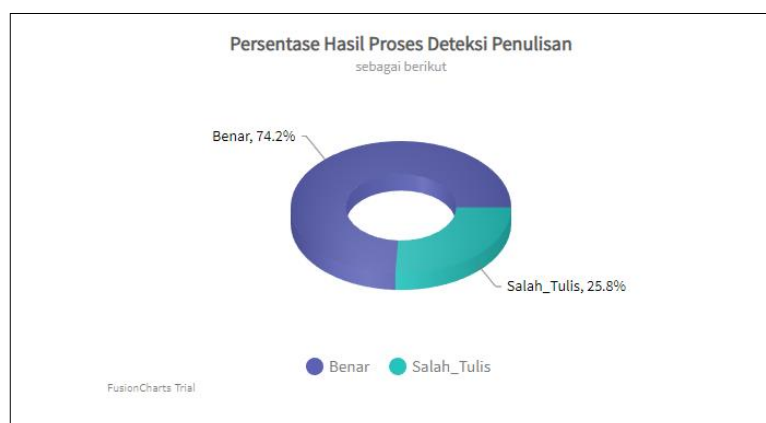
Evaluasi waktu pemrosesan dilakukan untuk mengetahui waktu yang dibutuhkan *utility* untuk memproses sebuah dokumen word (.docx), evaluasi ini menggunakan dua dokumen skripsi bertipe word (.docx) dengan masing-masing percobaan sebanyak lima kali pengulangan. Hasil dari evaluasi ini disajikan pada Tabel 5.

Tabel 5 Evaluasi Kinerja Waktu Pemrosesan.

Data	Jumlah String (JS)	Waktu Pemrosesan (WP) dalam satuan detik (s)					Rata-rata WP (RWP)	Jumlah Kata Perdetik (JS/RWP)
		1	2	3	4	5		
Skripsi 1	4217	195	194	197	195	200	196,2	21,49
Skripsi 2	5874	267	261	265	265	263	264,2	22,23

Hasil evaluasi kinerja yang disajikan pada Tabel 5 dapat disimpulkan bahwa setiap satu detik kata yang dapat diproses *utility* kurang lebih 21 kata.

Untuk grafik persentase skripsi 1 dan 2 dapat dilihat pada Gambar 9 dan 10.



Gambar 9 Grafik persentase Skripsi 1.

Pada skripsi 1 diperoleh persentase penulisan benar sebesar 74,2% (3129 kata), dan salah ketik 25,8% (1088 kata).



Gambar 10 Grafik persentase Skripsi 2.

Pada skripsi 2 diperoleh persentase penulisan benar sebesar 63.8% (3748 kata), dan salah ketik 36,2% (2126 kata).

4. KESIMPULAN

Dari hasil penelitian yang dilakukan kesimpulan yang didapat adalah Pengembangan *utility* pendeteksi penulisan kata dasar dan kata berimbuhan di- berdasarkan kamus pada dokumen bertipe word (.docx) telah diimplementasikan, dari hasil evaluasi kinerja waktu pemrosesan, rata-rata kata yang diproses dalam satu detiknya lebih kurang 21 kata dan hasil koreksi kesalahan kata dipengaruhi oleh kelengkapan kamus yang digunakan sebagai acuan koreksi kata dan tipe kesalahan ejaan kata.

DAFTAR PUSTAKA

- [1] U. I. Budiman, "Sistem Pengolahan Kalimat Bahasa Indonesia Menjadi Kata Dasar dengan Metode Stemmer Berbasis Pemrograman Hypertext Preprocessor," *J. Trisula LP2M Undar*, vol. 1, pp. 501–507, 2017.
- [2] R. P. Dewi and Y. Miftahudin, "Penerapan Algoritma Lemmatization pada Dokumen Bahasa Indonesia," *J. MND*, vol. 3, pp. 47–56, 2018.
- [3] R. Kusumaningrum and Y. Rochmawati, "Studi Perbandingan Algoritma Pencarian String dalam Metode Approximate String Matching untuk Identifikasi Kesalahan Pengetikan Teks," *J. Buana Inform.*, vol. 7, pp. 125–134, 2016.
- [4] R. S. Perdana, I. Cholissodin, and I. A. Fahma, "Identifikasi Kesalahan Penulisan kata (Typographical Error) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan Levenshtein Distance," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, pp. 53–62, 2018.
- [5] R. Darmawan, "Pengalaman, Usability, dan Antarmuka Grafis: Sebuah Penelusuran Teoritis," *ITB J. Vis. Art Des.*, 2013, doi: 10.5614/itbj.vad.2013.4.2.1.
- [6] T. Hidayat and M. Muttaqin, "Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis," *J. Tek. Inform. UNIS JUTIS*, 2018.