

## **APLIKASI PENDETEKSI PLAGIARISME BERDASARKAN ISI TEKS DOKUMEN MENGGUNAKAN METODE *LEVENSHTEIN DISTANCE***

**SHELLA BATRISYIA LUPITASARI**

Teknik Informatika, Fakultas Teknik  
Universitas Maarif Hasyim Latif, Sidoarjo, Indonesia  
e-mail : shella-batrisyia-lupitasari@student.umaha.ac.id

### **ABSTRAK**

Hadirnya teknologi informasi berkembang serta memiliki berdampak negatif yaitu plagiarisme termasuk isi teks dokumen . Plagiarisme adalah tindakan menirukan atau menjiplakan karya dan sumber orang lain dan mengakui sebagai karya sendiri. Oleh karena itu pendeteksian plagiarisme perlu dilakukan untuk mengurangi penjiplakan terhadap hasil karya orang lain. Tujuan ini menggunakan metode *levenshtein distance* sehingga dapat digunakan untuk membantu menentukan tindakan plagiarisme. Tipe format isi dokumen yang diuji adalah *.docx* dan *.txt* isi dokumen hanya menggunakan bahasa Indonesia. Tahapan dalam sistem adalah *preprocessing* yang terdiri dari *case folding*, *tokenizing*, *filtering*, *stemming*, dan *sorting*. Setelah proses *preprocessing* maka tahap selanjutnya adalah perhitungan menggunakan rumus metode *Levenshtein Distance* dan rumus pengukuran nilai *similarity* sehingga mendapatkan nilai presentase kemiripan antara isi teks dokumen. Pada pengujian menggunakan isi dokumen yang asli yaitu data dokumen berplagiat dengan metode *Levenshtein Distance* menghasilkan nilai *similarity* yang tinggi yaitu di atas 77% sampai 100% untuk isi dokumen yang tingkat kemiripannya tinggi. Sedangkan untuk isi dokumen dengan tingkat kemiripan yang rendah atau tidak berplagiat maka menghasilkan nilai *similarity* di bawah 40%.

**Kata kunci** : Dokumen, Java, *Levenshtein Distance*, Plagiarisme, Preprocessing

### **PENDAHULUAN**

Pada zaman sekarang teknologi sudah sangat berkembang dan sangat maju dan pesat. Salah satunya adalah pesatnya perkembangan teknologi informasi di dunia computer apalagi pemanfaatan teknologi informasi yang di dunia digital telah menjadi kebutuhan dalam era masa kini. Komponen dunia digital salah satunya adalah isi teks dokumen. plagiat sering terjadi dalam dunia akademik, baik tingkat umum . Tindakan plagiat yang dilakukan oleh kalangan umum ini yang tidak kreatif dan pemikiran hasil karya sendiri. Demi menyelesaikan secara cepat, dikalangan umum dapat melakukan teknik *copy-paste* lalu merubahkan kalimat tanpa perlu mempelajari materi terlebih dahulu. Kadangkala tindak menirukan ini dimodifikasi dengan mengganti kata-kata yang mengandung persamaan kata-kata (sinonim), dengan maksud persamaan kata-kata agar terlihat berbeda dari pekerjaan teman lainnya. Hal semacam ini dapat menimbulkan masalah terhadap evaluasi hasil belajar umum seperti siswa atau mahasiswa.

Solusi untuk mendeteksi kemiripan isi teks dokumen agar dapat membantu dalam proses pengerjaan untuk mendeteksi perbandingan dua buah dokumen dengan dalam jumlah yang sangat banyak dengan efektif dan efisien. Salah satu metode yang tepat dalam melakukan deteksi

kemiripan isi teks dokumen adalah dengan melakukan perhitungan dengan metode *Levenshtein Distance*. *Levenshtein Distance* memperhatikan tiga operasi dalam menentukan jarak *diff*.

### **METODE PENELITIAN**

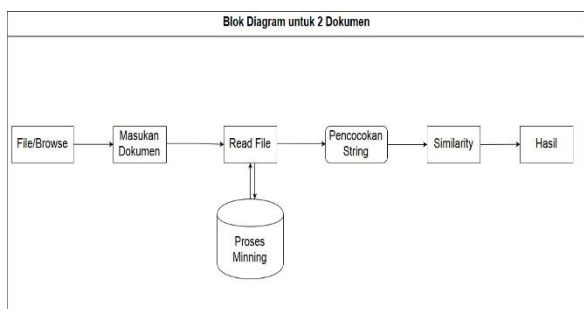
Sistem pendeteksi plagiarisme yang akan dibangun merupakan sistem yang akan mendeteksi tingkat kesamaan dua dokumen teks adalah sistem yang dilakukan secara otomatis yang menggunakan sistem offline dan hanya untuk mem. Aplikasi pendeteksi plagiarisme berdasarkan isi teks dokumen yang diusulkan ini menggunakan metode *Leveinstein Distance*. Data yang digunakan dalam penelitian ini adalah dokumen yang berformat *.docx*, dan *.txt* Dokumen tersebut akan menjadi bahan untuk membandingkan kemiripan antar dokumen.

### **Diagram Blok Sistem**

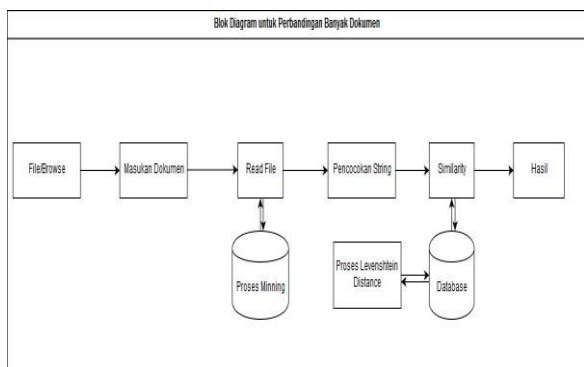
Proses perbandingan 2 dokumen dapat dijelaskan sebagai berikut :

1. pilih browse pada file laptop lalu pilih salah satu dokumen
2. lalu masukkan data isi teks dokumen dengan format Txt, dan Docx untuk memproseskan teks dokumen.

3. Kemudian sistem read file akan membaca file dokumen yang kita buat melalui dengan proses minning.
4. Sedangkan pada data isi teks dokumen dilakukan dengan proses perhitungan pencocokan string menggunakan metode Levenshtein Distance.
5. Dan melakukan proses similarity sehingga akan keluar hasil presentase nilai kemiripan isi dokumen dan waktu proses



Gambar 1. Blok Diagram 2 Dokumen



Gambar 2. Blok Diagram Perbandingan Banyak Dokumen

Proses perbandingan banyak dokumen sebagai berikut :

1. Pilih browse pada tempat file lalu pilih salah satu dokumen yang kita akan berplagiat
2. Lalu masukkan data isi teks yang berformat Docx dan Txt untuk memproseskan teks dokumen.
3. Kemudian sistem read file akan membaca file isi teks dokumen melalui dengan proses minning.
4. Sedangkan data isi teks dokumen dilakukan dengan proses perhitungan pencocokan string menggunakan metode levenshtein distance.
5. Dan melakukan proses similarity dapat memasukan dokumen yang akan dimasukan kedalam database, sebelum memasukan isi teks dokumen maka terlebih dahulu dilakukan deteksi dengan dokumen yang terdapat dalam database, apabila terdapat dokumen yang plagiat besar maka dokumen secara otomatis tidak akan masuk kedalam database.
6. Sehingga akan keluar hasil presentase dari nilai similarity kesamaan isi teks dokumen dan waktu proses.

Dan sebagaimana yang diilustrasikan pada gambar di atas, sistem yang akan dibangun dibagi menjadi dua, yakni blok perbandingan 2 dokumen dan blok perbandingan banyak dokumen. Blok perbandingan 2 dokumen merupakan blok sistem akan membaca dan akan menampilkan jumlah kalimat dan kata sedangkan blok perbandingan banyak dokumen merupakan blok dapat memasukan dokumen kedalam *database* tetapi sebelum memasukan dokumen kedalam *database* maka dilakukan perbandingan terlebih dahulu dengan seluruh dokumen yang ada dalam *database*.

## HASIL DAN PEMBAHASAN

Hasil dari penelitian ini adalah Aplikasi Pendeteksi Plagiarisme Berdasarkan isi Teks Dokumen Menggunakan Metode *Levenshtein Distance*. Ada beberapa *screen* yang ada di aplikasi ini adalah Form Beranda Halaman, Form Halaman Perbandingan 2 Dokumen, Form Halaman Perbandingan Banyak Dokumen, Form Halaman Bantuan.

### a. Form Beranda Halaman

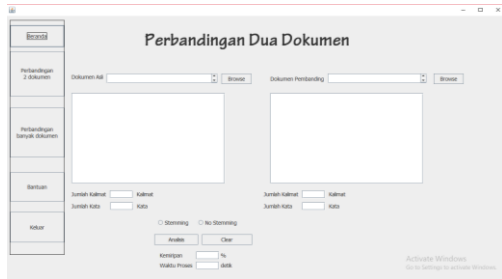


Gambar 3. Form Halaman Beranda

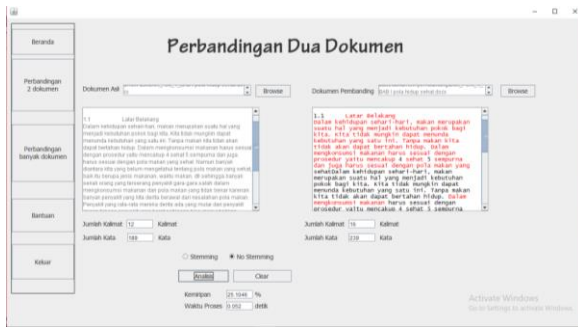
Beranda Gambar3 merupakan tampilan awal yang muncul pertama ketika menjalankan program pada tampilan ini. Terdapat tombol browse yang memudahkan user yaitu tombol *Browse* untuk mencari alamat lokasi file dokumen asli dan file dokumen yang akan dibandingkan dan dapat memilih menu perbandingan dua dokumen, perbandingan banyak dokumen, dan tombol bantuan.

### b. Form Halaman Perbandingan 2 Dokumen

Form halaman perbandingan dua dokumen di Gambar 4 jika pengguna pilih tombol tab menu yang berisi "Perbandingan Dua Dokumen" kemudian pengguna pilih tombol browse untuk menampilkan tempat file kemudian pilih Stemming atau No stemming kemudian pilih tombol analisis, muncul hasil proses melalui Kemiripan dan Waktu Proses seperti hasil Gambar 5.

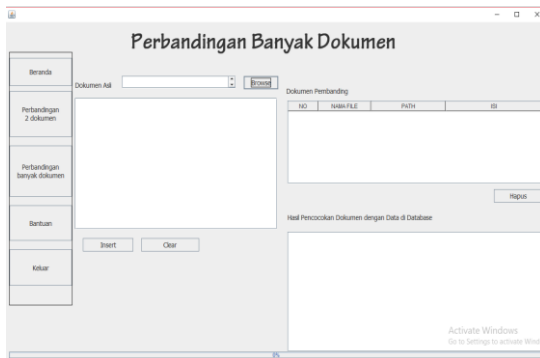


Gambar 4. Form Halaman Perbandingan 2 Dokumen



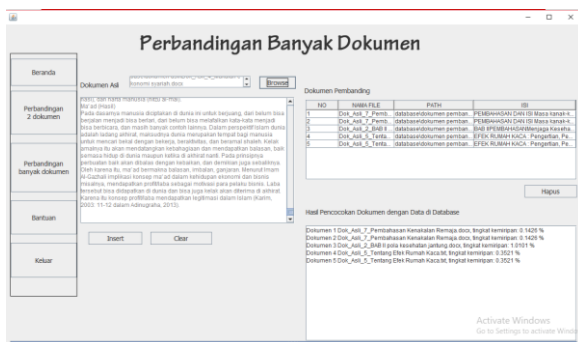
Gambar 5. Hasil Perbandingan Dua Dokumen

**c. Form Halaman Perbandingan Banyak Dokumen**



Gambar 6. Form Halaman Perbandingan Banyak Dokumen

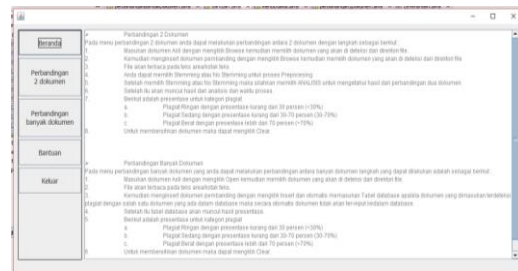
Form halaman perbandingan banyak dokumen di Gambar 6 jika pengguna pilih tombol tab menu yang berisi “Perbandingan Banyak Dokumen” kemudian pengguna pilih tombol browse untuk menampilkan tempat file kemudian pilih tombol insert lalu dokumen akan memasukan tabel “dokumen pembanding” seperti hasil Gambar 7.



Gambar 7. Hasil Perbandingan Banyak Dokumen

Hasil Tabel 1 dari Perbandingan Banyak Dokumen pengguna dapat melakukan perbandingan dengan banyak dokumen yang ada dalam database. Proses perbandingan sama dengan proses perbandingan dua dokumen dalam menu yang sebelumnya. Pengguna dapat memasukan dokumen untuk dimasukan kedalam database. Tahap pertama adalah membaca file yang akan dimasukan kemudian setelah dokumen terbaca kedalam teks area maka akan dilakukan insert dokumen yang dimasukan terdeteksi plagiat dengan salah satu dokumen yang ada dalam database maka secara otomatis dokumen tidak akan ter-input kedalam database. Begitu sebaliknya apabila dokumen yang dimasukan tidak terdeteksi plagiat dengan salah satu dokumen yang ada dalam database maka dokumen akan secara otomatis masuk kedalam database. Jika penyimpanan file Database terotomatis penyimpan di Mysql.

**d. Form Halaman Bantuan**



Gambar 8. Form Halaman Bantuan

Pada menu ini berisi petunjuk bantuan dalam menjalankan proses perbandingan. Pada tampilan ini dijelaskan langkah-langkah pengoperasian sistem sehingga pengguna tidak kesulitan untuk mengoperasikan sistem tersebut

**Pengujian Sistem**

Pada tahap ini dilakukan proses pengujian terhadap suatu sistem yang dibangun. Pengujian sistem adalah analisis kemampuan aplikasi berdasarkan somilaritas, waktu dan tipe-tipe file. Pengujian pertama dilakukan pada dokumen dengan kemungkinan kemiripan yang besar, sedangkan pengujian kedua dilakukan pada dokumen yang dengan kemungkinan kemiripan yang kecil.

Tabel 1. Hasil Perbandingan Dokumen Yang Tidak Mirip

No	Dokumen Asli	Dokumen Pembanding	Stemming		No Stemming	
			Kemiripan (%)	Waktu (Detik)	Kemiripan (%)	Waktu (Detik)
1	Dok_Aslis_8	Dok_Pem_3	0,1328	1,003	0,3131	0,191
2	Dok_Aslis_3	Dok_Pem_5	0,2558	0,251	0,6452	0,042
3	Dok_Aslis_1	Dok_Pem_2	0,2294	0,405	0,2102	0,076
4	Dok_Aslis_4	Dok_Pem_6	0,0000	0,835	1,1111	0,102

Hasil perbandingan dokumen Tabel 1 menunjukkan bahwa dokumen dengan kemungkinan kemiripan yang kecil menghasilkan nilai similaritas yang rendah yaitu di bawah 40%. pengujian dokumen di atas menunjukkan bahwa yang paling mempengaruhi lama tidaknya perhitungan nilai *similarity* bukan ditentukan oleh jenis dokumen maupun ukuran dokumen

## PENUTUP

Aplikasi ini menggunakan proses *Preprocessing* dan proses pengukuran nilai *similarity* untuk mendapatkan nilai kemiripan. Pada pengujian menggunakan data real yaitu data dokumen berplagiat yang diambil dari artikel/berita lewat internet, algoritma Levenshtein Distance menghasilkan nilai *similarity* yang tinggi yaitu di atas 85 % sampai 100 % untuk dokumen yang tingkat kemiripannya tinggi. Sedangkan untuk dokumen dengan tingkat kemiripan yang rendah atau tidak berplagiat maka menghasilkan nilai *similarity* di bawah 40%.

Dalam pembuatan Aplikasi Pendeteksi Plagiarisme Berdasarkan Isi Teks Dokumen Menggunakan Metode *Levenshtein Distance* ini masih banyak terdapat kekurangan dan jauh dari kata sempurna. Untuk itu masih perlu dilakukan sebuah penyempurnaan. Berikut beberapa saran untuk pengembangan lebih lanjut dari aplikasi ini Perlu ditambahkan proses untuk mendeteksi padanan kata yaitu kata-kata yang berbeda namun memiliki makna yang sama, Perlu ditambahkan sebuah proses yang bisa mendeteksi adanya kesalahan ejaan karena kesalahan ejaan ini sangat mempengaruhi proses filtering dan stemming sehingga dapat mengurangi nilai *similarity*, Dalam program yang selanjutnya dapat dikembangkan

untuk proses yang juga bisa membaca dokumen berbentuk .xlsx dan .Pdf maupun dalam berbentuk program aplikasi Android.

## DAFTAR PUSTAKA

- [1] U. Sutisna and J. Adisantoso, "Koreksi Ejaan Query Bahasa Indonesia Menggunakan Algoritme Damerau Levenshtein," *J. Ilm. Ilmu Komput.*, vol. 8, no. 2, 2010.
- [2] M. R. Pratama, C. Eko Budi, and M. Gita Indah, "Aplikasi Pendeteksi Duplikasi Dokumen Teks Bahasa Indonesia Menggunakan Algoritma Winnowing Dengan Metode K-Gram Dan Synonym Recognition," *Tek. Inform. UM Malang*, 2012.
- [3] M. F. Azhri, "Aplikasi Pendeteksi Kemiripan Isi Teks Tugas Siswa Menggunakan Algoritma Levenshtein Distance," Universitas Nusantara PGRI, 2019.
- [4] S. Sastroasmoro, "Beberapa Catatan tentang Plagiarisme," *Maj. Kedokt. Indones.*, vol. 57, no. 8, pp. 239-244, 2007.
- [5] E. Waridah, "Pedoman Umum Ejaan Bahasa Indonesia," 2016.
- [6] B. P. Pratama and S. A. Pamungkas, "Analisis kinerja algoritma levenshtein distance dalam mendeteksi kemiripan dokumen teks," *LOGIK@*, vol. 6, no. 2, pp. 131-143, 2016.
- [7] H. Ajie and A. S. Bangsa, "Aplikasi Pendeteksi Dugaan Awal Plagiarisme Pada Tugas Siswa Dan Mahasiswa Berdasarkan Kemiripan Isi Teks Menggunakan Algoritma Levenshtein Distance," *PINTER J. Pendidik. Tek. Inform. dan Komput.*, vol. 1, no. 1, pp. 23-30, 2017.