Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 1, September 2020
e-ISSN 2685-998X
DOI 10.30865/json.v2i1.2451

Perancangan Aplikasi Deteksi Kemiripan Dokumen Teks Menggunakan Algoritma Shingling

Irwan Saputra Simanullang

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: irwansaputra719@gmail.com
Submitted: **02/09/2020**; Accepted: **27/09/2020**; Published: **30/09/2020**

Abstrak–Deteksi kemiripan dokumen teks diperlukan untuk menghindari penumpukan informasi pada suatu database atau file sistem, salah satu cara untuk mengetahui plagiasi antara dua dokumen dengan deteksi kemiripan dua dokumen yang akan dibandingkan untuk mendeteksi plagiasi, Algoritma Shingling merupakan algoritma yang digunakan untuk proses pencarian near-duplicate document. Algoritma Shingling diimplementasikan untuk mendeteksi kemiripan pada dokumen teks. Kemudahan dalam menyalin informasi dari satu dokumen ke dokumen yang lain merupakan salah satu efek dari kemajuan teknologi informasi. Dan hal ini mengarah kepada plagiasi yang tidak di inginkan, sehingga proses deteksi dokumen dapat diminimalisasi dengan baik. Pada penelitian ini Hasil pengujian menunjukkan bahwa aplikasi yang dibuat, dapat mendeteksi kemiripan dokumen teks yang telah melalui berbagai manipulasi yaitu scaling (perbesar/perkecil), rotasi, cropping (potong sebagian), dan memanipulasi dokumen.

Kata Kunci: Shingling, Deteksi, Kemiripan, Dokumen, Teks

Abstract–Detection of similarity in text documents is needed to avoid the accumulation of information in a database or file system, one way to find out plagiarism between two documents by detecting the similarity of two documents to be compared to detect plagiarism, the Shingling Algorithm is an algorithm used for near-duplicate document search processes. Shingling's algorithm is implemented to detect similarities in text documents. The ease of copying information from one document to another is one of the effects of advances in information technology. And this leads to unwanted plagiarism, so that the document detection process can be minimized properly. In this study, the test results show that the application created can detect the similarity of text documents that have gone through various manipulations, namely scaling (enlarge / reduce), rotation, cropping (partially cut), and manipulating documents.

Keywords: Shingling, Detection, Similarity, Document, Text

1. PENDAHULUAN

Saat ini sistem komputer memegang peranan yang sangat penting dalam segala bidang. Dimana komputer dapat melakukan pekerjaan yang sangat cepat, teliti, dan efesien untuk pekerjaan yang berulang-ulang. Sehingga perangkat ini banyak digunakan oleh perguruan tinggi. Untuk mengatasi banyaknya plagiarism dalam dunia pendidikan, maka dibutuhkan suatu aplikasi dari komputer.

Kemiripan dokumen merupakan permasalahan yang tidak hanya melanggar hak cipta atau kepemilikan, melainkan mencontoh atau meniru karya orang lain. Kemiripan dokumen tersebut hanya tinggal melakukan *Copy-paste-modify* pada sebagian isi dokumen, bahkan seluruh isi dokumen. Contohnya dokumen A dan dokumen B adalah dokumen yang kemiripannya menyamai dokumen A, dari perbandingan dokumen tersebut maka sangat sulit untuk mendeteksi kemiripan dokumen hanya memodalkan pengelihatan mata seseorang, maka dokumen yang dibandingkan disederhanakan dalam bentuk set, melalui proses *shingling*[1].

Banyak institusi dalam dunia pendidikan menerapkan sanksi akademis terhadap pelaku plagiat untuk mengurangi plagiarisme. Yang menjadi permasalahannya adalah bagaimana cara untuk mengetahui apakah seorang mahasiswa melakukan kemiripan dokumen atau tidak dalam membuat suatu karya tulis. Untuk mengetahuinya perlu dilakukan pengecekan secara teliti terhadap hasil tulisan mahasiswa tersebut kemudian dibandingkan dengan hasil tulisan mahasiswa yang lainnya. Tetapi usaha tersebut akan memerlukan waktu yang lama dan ketelitian yang sangat tinggi, jika pembandingan tersebut dilakukan secara manual. Oleh karena itu diperlukan suatu aplikasi pendeteksian kemiripan pada dokumen teks yang dilakukan secara terkomputerisasi.

Algoritma *Shingling* merupakan algoritma yang ditemukan oleh Andrei Broder[2], yang digunakan untuk proses pencarian *near-duplicate* document. Algoritma *shingling* bekerja dengan cara memotong-motong teks menjadi kumpulan kata, dan membangkitkan suatu nilai unik (*fingerprint*) untuk tiap kumpulan kata. Algoritma *hash* dapat digunakan untuk membangkitkan *fingerprint*[3].

Pada dasarnya salah satu sifat manusia suka meniru dan menginginkan sesuatu kemudahan. Sifat tersebut akan memicu tindakan negatif apabila dilatarbelakangi oleh motivasi untuk berbuat curang dan rendahnya kemampuan masyarakat berkreasi dan berinovasi menciptakan suatu karya yang original. Dalam hal ini tindakan negatif yang dimaksud adalah kemiripan dokumen (plagiarisme).

Fenomena kemiripan dokumen yang lebih spesifik sering terjadi di dunia akademis. Hal ini dikarenakan kegiatan tulis-menulis sering dilakukan oleh mahasiswa untuk menyelesaikan tugas kuliah. Praktik menduplikasikan beberapa bagian atau keseluruhan tulisan milik orang lain tanpa mencantumkan sumbernya secara teliti dan lengkap merupakan hal yang sering ditemui dalam penulisan laporan, tugas, makalah ataupun skripsi mahasiswa.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 1, September 2020
e-ISSN 2685-998X
DOI 10.30865/json.v2i1.2451

Pada penelitian yang ditulis oleh Muhammad Yusuf Adiansyah yang berjudul Perancangan dan Implementasi Aplikasi Deteksi Kemiripan Dokumen Menggunakan Algoritma Shingling dan MD5 Fingerprint menghasilkan kesimpulan bahwa deteksi kemiripan dokumen dapat dilakukan dengan menggunakan algoritma *shingling* dan diperkuat dengan algoritma MD5 untuk membangkitkan nilai *fingerprint*, sehingga terhindar dari *collision*[4].

Penelitian yang ditulis oleh Andry Vegard Sariwating dengan judul Perancangan dan Implementasi Aplikasi Deteksi Kemiripan Citra Digital Menggunakan Algoritma Shingling dan Redundant Pixel Removal menghasilkan kesimpulan bahwa Deteksi kemiripan gambar dapat dilakukan dengan membandingkan nilai-nilai piksel gambar tersebut, dengan membentuk shingle-shingle dan Untuk memperkecil jumlah piksel dan shingleyang dibandingkan, maka dilakukan penghapusan piksel-piksel yang redudan. Piksel redundan adalah piksel yang bernilai sama dengan piksel sebelumnya[5].

Ada dua cara untuk mengatasi permasalahan kemiripan dokumen, yaitu dengan mencegah dan mendeteksi. Mencegah berarti menjaga atau menghalangi agar plagiarisme tidak dilakukan. Usaha seperti ini harus dilakukan sedini mungkin terutama pada sistem pendidikan dan moral masyarakat. Mendeteksi berarti melakukan usaha untuk menemukan tindakan kemiripan yang telah dilakukan.

2. METODE PENELITIAN

2.1 Kemiripan Dokumen

Kemiripan dokumen merupakan permasalahan yang tidak hanya melanggar hak cipta atau kepemilikan. Apabila dipandang dari sisi para pembaca, Kemiripan dokumen juga merupakan tindakan yang membohongi dan menimbulkan kesalahpahaman mengenai orisinalitas dari penulis yang sebenarnya. Para siswa dan mahasiswa atau peneliti diperbolehkan untuk menciptakan suatu karya baru yang timbul dari pengembangan ide orang lain. Tetapi pemanfaatan ide orang lain tanpa membubuhkan pernyataan sumber atau keterangan yang menyatakan pengakuan bahwa karya tersebut berasal dari pengembangan ide orang lain, hal ini merupakan tindakan yang tidak dapat diterima[6].

2.2 Text Mining

Teks mining, yang juga disebut sebagai Teks Data *Mining* (TDM) atau *Knowledg*e *Discovery in Text* (KDT), secara umum mengacu pada proses ekstraksi informasi dari dokumen-dokumen teks tak terstruktur (*unstructured*). Teks mining dapat didefinisikan sebagai penemuan informasi baru dan tidak diketahui sebelumnya oleh komputer, dengan secara otomatis mengekstrak informasi dari sumber -sumber teks tak terstruktur yang berbeda. Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber[10].

2.3 String Matching

String Matching atau pencocokan string adalah suatu metode yang digunakan untuk menemukan suatu keakuratan/hasil dari satu atau beberapa pola teks yang diberikan. *String Matching* merupakan pokok bahasan yang penting dalam ilmu computer karena teks merupakan bentuk utama dari pertukaran informasi antar manusia, misalnya pada *literature*, karya ilmiah, halaman web dan sebagainya.

2.4 Algoritma Shingling

Algoritma ini bekerja dengan cara membuat sebuah shingle yang berisi beberapa kata dengan jumlah yang tetap. Angka yang menentukan jumlah kata dalam satu *shingle* ini disebut ***gram***. Pada tiap *shingle* dibangkitkan nilai *fingerprint*. Proses *shingling* ini akan menghasilkan himpunan yang berisi sejumlah *fingerprint*. Himpunan ini kemudian dibandingkan dengan himpunan yang dihasilkan dari dokumen kedua. Nilai kemiripan diperoleh dengan cara membagi jumlah fingerprint yang sama (*intersection*) dari dua dokumen dengan jumlah *fingerprint* gabungan (*union*). Proses perhitungan tersebut didefinisikan sebagai berikut[11]:

$$r(A,B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|} \quad (1)$$

Persamaan 1 Rumus Nilai Kemiripan Algoritma Shingling dilakukan melalui beberapa langkah berikut:

1. Hilangkan tanda baca pada dokumen.
2. Dimulai dengan kata pertama, buat satu shingle berisi kata pertamatersebut sampai 3 kata berikutnya.
3. Pindah ke kata kedua, buat shingle berisi kata kedua dan 3 kataberikutnya.
4. Lakukan pembentukan shingle sampai dengan 4 kata terakhir daridokumen tersebut.
5. Untuk tiap shingle, bangkitkan nilai fingerprint.
6. Lakukan langkah 1 sampai dengan 5 untuk dokumen kedua.
7. Gunakan rumus nilai kemiripan dokumen Persamaan 1 untuk menghitung nilai kemiripan dokumen.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 1, September 2020
e-ISSN 2685-998X
DOI 10.30865/json.v2i1.2451

3. HASIL DAN PEMBAHASAN

Dalam merancang sebuah sistem khususnya sistem yang berbasis aplikasi perlu dilakukan analisa. Analisa berguna untuk meminimalisir terjadinya kesalahan pada saat mendeteksi kemiripan dokumen teks. Analisa merupakan upaya untuk melakukan pemahaman tertentu terhadap sesuatu masalah yang dilakukan dalam pengkajian. Dalam mendeteksi suatu kemiripan dokumen teks mutlak dilakukan penelitian dan penganalisaan tentang dokumen teks yang akan dideteksi, berikut beberapa analisa yang dilakukan untuk mendeteksi tepi menggunakan metode algoritma *Shingling*.

Dalam menganalisa cara kerja algoritma tersebut maka perlu adanya sebuah aplikasi yang akan mengetahui bagaimana proses yang dihasilkan dari masing-masing algoritma tersebut dalam melakukan pendeteksian, aplikasi yang dirancang ini yaitu aplikasi yang berbasis dekstop. tools yang digunakan untuk merancang aplikasi analisa pendeteksian tersebut.

Manfaat yang diperoleh dalam menganalisa pendeteksian algortima tersebut adalah dapat dengan mudah memahami bagaimana proses dokumen teks yang dilakukan dan bagaimana kecepatan yang dilakukan dari algoritma tersebut dalam melakukan pendeteksian.

3.1 Penerapan Algoritma Shingling

Dalam penerapan algoritma shingling untuk mendeteksi kemiripan dokumen teks, langkah pertama dilakukan dengan membaca isi dokumen teks tersebut. Dokumen teks yang digunakan dalam penelitian ini yaitu dokumen dengan eksteksi dan isi dokumen teks tersebut yang nantinya akan di dideteksi dan disimpan menjadi dua dokumen dan akan dibandingkan hasilnya seberapa besar kemiripan dari dua dokumen tersebut.

Dalam pelaksanaan Shingling, ada beberapa metode yang digunakan yaitu metode deskriptif, evaluatif dan eksperimental. Metode penelitian deskriptif digunakan dalam penelitian awal untuk menghimpun data tentang kondisi yang ada. Metode evaluatif digunakan untuk mengevaluasi proses ujicoba pengembangan suatu dokumen. Dan metode eksperimen digunakan untuk menguji keampuhan dari deteksi yang dihasilkan.

Tahap Deskriptif: dilakukan pengumpulan data dengan memperhatikan kebutuhan deteksi plagiasi dokumen dan algoritma yang dapat digunakan untuk melakukan deteksi kemiripan dokumen; Tahap Evaluatif: dilakukan evaluasi proses ujicoba perancangan aplikasi yang meliputi perancangan proses aplikasi dan perancangan antarmuka. Perancangan proses dibuat dengan menggunakan Use-Case dan Activity Diagram; Tahap Eksperimen: dokumen yang dihasilkan pada tahap sebelumnya kemudian diuji dan dilakukan analisa berdasarkan hasil pengujian tersebut, untuk mengetahui apakah aplikasi yang dihasilkan, telah memenuhi tujuan dan kebutuhan.

Proses dimulai dengan memecah dokumen menjadi kumpulan kata-kata. Pada proses ini semua tanda baca dan karakter non-alphanumeric diabaikan. Selanjutnya adalah membentuk shingle- shingle, tiap shingle berisi berberapa kata. Untuk tiap shingle dihitung nilainya. Angka kemiripan diperoleh dengan cara menghitung jumlah shingle nya yang sama dari dua dokumen yang dibandingkan, kemudian angka tersebut dibagi dengan gabungan teks kedua dokumen.

Dokumen teks diperoleh dengan cara membandingkan nilai dokumen teks A dan dokumen teks B, namun hanya nilai unik saja, sehingga tidak ada nilai dokumen yang muncul lebih dari satu kali.

Adapun isi dari Dokumen A adalah sebagai berikut :

Dokumen A
Ayah kita Pejuang Putra
Kita Pejuang Putra sejati
Pejuang Putra sejati Putra
Putra sejati Putra Indonesia
Sejati Putra Indonesia Harum
Putra Indonesia Harum namanya

Adapun isi dari Dokumen B adalah sebagai berikut :

Dokumen B
Ayah kita Pejuang Putra
Kita Pejuang Putra sejati
Pejuang Putra sejati Putra
Putra sejati Putra Nusantara
Sejati Putra Nusantara Wangi
Putra Nusantara Wangi namanya

Dimana dokumen teks yang dihasilkan akan dibandingkan seberapa mirip dokumen yang akan dibandingkan. Nilai dari dokumen A dan B akan dibandingkan di bawah ini.

Dokumen A	Dokumen B
Ayah kita Pejuang Putra	Ayah kita Pejuang Putra

Tabel 1. Dokumen A

A y a h	k i t a	P e j u a n g	P u t r a

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 1, September 2020
e-ISSN 2685-998X
DOI 10.30865/json.v2i1.2451

Tabel 2. Dokumen B

A y a h k i t a P e j u a n g P u t r a

Hasil dari tabel dokumen Ayah kita Pejuang Putra bahwa perbandingan kemiripan Dokumen a dan b sangat Cocok, jika dinilai dari perumusan maka nilai yang dihasilkan dari perbandingan tersebut adalah 100%. Tidak memiliki cacat pada dokumen yang dibandingkan, maka dokumen teks ini di nyatakan 100% Plagiarisme.

Tabel 3. Dokumen A

P u t r a s e j a t i P u t r a I n d o n e s i a

Tabel 4. Dokumen B

P u t r a s e j a t i P u t r a N u s a n t a r a

Dari hasil tabel yang di atas dokumen a dan dokumen b memiliki perbedaan yaitu Indonesia dan Nusantara jika kata demi kata di masukan maka nilai yang akan akan muncul akan berbeda,
Nilai Dokumen A 100%
Nilai Dokumen B 100%

Jika digabungkan nilai dari dokumen A dan dokumen B maka nilai 100%. Dan hasil kemiripan dari dokumen A dan B adalah = 72%

$r(A,B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|}$

$r(100.100) = \frac{16 \cap 9}{16 \cup 9}$

$r(25.25) = \frac{25*4}{25*4} = \frac{Dok\ A}{Dok\ B} = \frac{100}{100} = 1$

r = 64 - 36
r = 28% Hasil Perbedaan Dokumen.
r = 72% Hasil Persamaan dari Dokumen.

Untuk mencapai tujuan perancangan sistem, maka perlu dilakukan pengujian.Pengujian dilakukan dengan menggunakan beberapa dokumen txt dan doc.

Tahap 1. Menggunakan dokumen txt.

- *File* 1 dan 2 menggunakan dokumen txt yang berisi 100 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 3 dan 4 menggunakan dokumen txt yang berisi 500 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 5 dan 6 menggunakan dokumen txt yang berisi 1000 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 7 dan 8 menggunakan dokumen txt yang berisi 2000 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 9 dan 10 menggunakan dokumen txt yang berisi 5000 kata. Dicatat prosentase kecocokan dan waktu proses.

Tahap 2. Menggunakan dokumen doc.

- *File* 1 dan 2 menggunakan dokumen doc yang berisi 100 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 3 dan 4 menggunakan dokumen doc yang berisi 500 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 5 dan 6 menggunakan dokumen doc yang berisi 1000 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 7 dan 8 menggunakan dokumen doc yang berisi 2000 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 9 dan 10 menggunakan dokumen doc yang berisi 5000 kata. Dicatat prosentase kecocokan dan waktu proses.

Tahap 3. Menggunakan dokumen doc yang memuat gambar dan *hyperlink*.

- *File* 1 dan 2 menggunakan dokumen doc yang berisi 100 kata, 10 gambar dan 10 *hyperlink*. Dicatat prosentase kecocokan dan waktu proses.
- *File* 3 dan 4 menggunakan dokumen doc yang berisi 100 kata, 20 gambar dan 20 *hyperlink*. Dicatat

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 1, September 2020
e-ISSN 2685-998X
DOI 10.30865/json.v2i1.2451

prosentase kecocokan dan waktu proses.

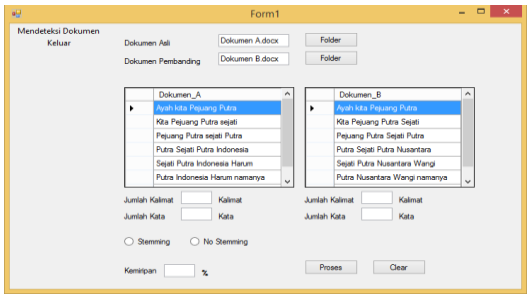
- *File* 5 dan 6 menggunakan dokumen doc yang berisi 100 kata, 30 gambar dan 30 *hyperlink*. Dicatat prosentase kecocokan dan waktu proses.
- *File* 7 dan 8 menggunakan dokumen doc yang berisi 100 kata, 40 gambar dan 40 *hyperlink*. Dicatat prosentase kecocokan dan waktu proses.
- *File* 9 dan 10 menggunakan dokumen doc yang berisi 100 kata, 50 gambar dan 50 *hyperlink*. Dicatat prosentase kecocokan dan waktu proses.

Tahap 4. Menggunakan dokumen *doc* yang memiliki jumlah kata sama namun memiliki isi yang berbeda.

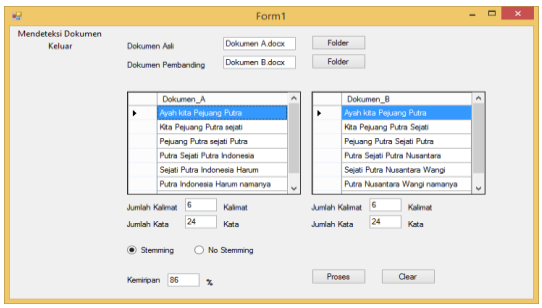
- *File* 1 dan 2 menggunakan dokumen *doc* yang berisi 100 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 3 dan 4 menggunakan dokumen *doc* yang berisi 1000 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 5 dan 6 menggunakan dokumen *doc* yang berisi 10000 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 7 dan 8 menggunakan dokumen *doc* yang berisi 100000 kata. Dicatat prosentase kecocokan dan waktu proses.
- *File* 9 dan 10 menggunakan dokumen *doc* yang berisi 1000000 kata. Dicatat prosentase kecocokan dan waktu proses.

3.2 Implementasi

Berikut adalah merupakan tampilan dari aplikasi form proses hasil dari implementasi sistem yang telah dibuat:



Gambar 1. Tampilan Form Proses



Gambar 2. Tampilan Form Output

4. KESIMPULAN

Berdasarkan penelitian ini maka kesimpulan yang di peroleh sebagai berikut :

1. Proses mendeteksi kemiripan dokumen teks yang menggunakan algoritma Shingling pada dokumen teks tersebut.
2. Perancangan aplikasi deteksi kemiripan dokumen teks menggunakan *Microsoft Visual Basic* 2008 dan menerapkan metode algoritma Shingling.
3. Metode algoritma Shingling harus diterapkan dalam program *Microsoft Visual Basic* 2008 agar programnya berjalan sesuai dengan perancangan aplikasi deteksi kemiripan dokumen teks.

REFERENCES

[1] A. Ilmiah, "Perancangan dan Implementasi Aplikasi Deteksi Kemiripan Citra Digital Menggunakan Algoritma Shingling dan Redundant Pixel Removal Perancangan dan Implementasi Aplikasi Deteksi Kemiripan Citra Digital Menggunakan Algoritma Shingling dan Redundant Pixel Rem," no. 672010147, 2016.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 1, September 2020
e-ISSN 2685-998X
DOI 10.30865/json.v2i1.2451

[2] D. Kepada, F. T. Informasi, U. Memperoleh, and G. Sarjana, "Perancangan dan Implementasi Aplikasi Deteksi Kemiripan Dokumen Menggunakan Algoritma Shingling," 2014.
[3] A. G. N. K. Walia, "A Review. International Journal of Engineering Development and Research," no. Cryptography Algorithms, 2014.
[4] A. Kadir, Algoritma & Pemrograman Menggunakan C & C++. yogyakarta: andi, 2012.
[5] P. L. Montanari, D. & Puglisi, "In Multidisciplinary Research and Practice for Information Systems," Near Duplic. Doc. Detect. large Inf. flows, vol. Informatio, 2012.