PENERAPAN ALGORITMA RABIN KARP UNTUK MEDETEKSI KEMIRIPAN DUA DOKUMEN TEKS

[1]Agus Suparwanta, [2]Riyadi J. Iskandar,[3]Soebandi,
[1, 2]Teknik Informatika, STMIK Widya Dharma, Pontianak
[3]Sistem Informasi, STMIK Widya Dharma, Pontianak
[1]agus_po@yahoo.co.id, [2]riyadijiskandar@gmail.com,[3]soebandi@gmail.com

Abstract

Plagiarism is the act of taking essay or opinion of others without citing sources or quotations from the original work. Basically, the matching words is used to find the similarity of two documents. In this essay, Rabin Karp algorithm was used to transform words into hash values to be able to do a comparison or matching. Also, to find the value of similarity, dice similarity coefficient was used.The author used causal relationships (experimental) as the study design which means the author conducted experiments and tested applications which have been made and studied the literature related to the design of similarity of two text documents detector. The author used analysis techniques, Unified Modeling Language (UML) to describe the work flow detection system resemblance to text documents. In designing the application, the author used Visual Basic .NET. This research was conducted to produce a system that can be used as a reference for detecting similarities of text documents. However, the level of plagiarism can be adjusted with the consumer view. The whole process can be concluded that the similarity detection system two text documents can display the percentage of similarity in the two text documents. The suggestion is addressed to the reader to develop a similarity detection system of two text documents with Rabin Karp algorithm to get better and become a reference for determining plagiarism.

***Keywords**: Application, menu,expert,disease,digestion.*

Abstrak

Plagiat adalah tindakan mengambil karangan atau pendapat orang lain tanpa menyebutkan sumber atau kutipan dari karya aslinya. Pada dasarnya untuk menemukan kemiripan dua dokumen teks yaitu dengan pencocokan kata. Dalam penelitian ini menggunakan algoritma Rabin Karp yaitu dengan mengubah kata menjadi nilai hash untuk dapat melakukan perbandingan atau pencocokan. Untuk menemukan nilai kemiripan digunakan dice similarity coefisien.Penulis menggunakan desain penelitian hubungan kausal (eksperimental) yaitu penulis melakukan percobaan dan pengujian terhadap aplikasi yang dibuat dan dengan cara mempelajari literatur-literatur yang berhubungan dengan materi perancangan deteksi kemiripan dua dokumen teks. Penulis menggunakan teknik analisis Unified Modelling Language (UML) untuk menggambarkan alur kerja sistem deteksi kemiripan dokumen teks. Dalam merancang aplikasi tersebut penulis menggunakan bahasa pemrograman Visual Basic .NET.Penelitian ini dilakukan untuk menghasilkan sistem yang dapat dijadikan acuan untuk mendeteksi kemiripan dokumen teks. Namun, tingkat tindakan plagiat dapat disesuaikan dengan pandangan penggunannya.Dari keseluruhan proses penelitian dapat disimpulkan bahwa sistem deteksi kemiripan dua dokumen teks dapat menampilkan persentase kemiripan pada dua dokumen teks. Adapun saran yang ditujukan kepada pembaca adalah untuk mengembangkan sistem dekteksi kemiripan dua dokumen teks dengan algoritma Rabin Karp menjadi lebih baik dan menjadi acuan untuk menentukan tindakan plagiat..

Kata Kunci: Plagiat, text mining, algoritma rabin karp, dice similarity coefisien.

1. PENDAHULIAN

Teknologi informasi pada saat ini berkembang begitu pesat. Karena teknologi adalah fasilitas yang membantu pekerjaan manusia. Perkembangan teknologi informasi memberikan dampak postif dan negatif. Dampak positifnya membantu meringankan pekerjaan dan dampak negatifnya orang cendrung ingin proses yang instan, salah satu bentuknya yaitu plagiat. Plagiat merupakan tindakan yang merugikan karena mengambil karya orang lain tanpa sepengetahuan atau mencantumkan sumber dari pembuatnya. Kegiatan plagiat sangat meresahkan bagi yang membuat karya orisinil. Dikarenakan plagiat juga dapat diartikan penjilplakan karya orang lain seolah-olah menjadi karya sendiri. Ada begitu banyak tindakan yang berhubungan dengan plagiat salah satunya yaitu pelanggaran dengan hak cipta berupa dokumen. Selain merugikan bagi pembuat aslinya juga merugikan bagi orang yang melakukan kegiatan plagiat. Kebebasan berinovasi menjadi sempit dan lunturnya moral yang kurang menghargai hasil karya orang lain. Maka dari itu Pencegahan merupakan suatu cara untuk mengurangi *Plagiarisme.*

Metode untuk menentukan kemiripan dokumen adalah dengan cara pencocokan string. Banyak algoritma yang dapat digunakan antara lain *Rabin Karp, Boyer Moore, Knut Morris Path* dan lain-lain. Sangat kurang efisien bila pecocokan *string* tidak dapat melakukan pecocokan *String* pada *Multi Patren.* Namun, ini tergantung pada kondisi yang sedang dihadapi. Biasanya pada penelitian lain telah dilakukan beberapa modifikasi pada algoritma dan menambahkan metode lain untuk dapat mengatasi kasus yang sedang dihadapi.

Pada deteksi kemiripan dokumen ini menggunakan konsep *Text Similarity* , *Text Mining*, dan algoritma *Rabin Karp* untuk *String Matching*. Pada *Text Similarity* adalah proses untuk menentukan kemiripan antara kedua dokumen yaitu menggunakan *Dice Similarity Coefisien*. Sedangkan *Text Ming* adalah untuk mencari kata-kata untuk mewakili isi dokumen. Namun pada *Text Mining* tidak menggunakan *stemming* yaitu menemukan kata dasar pada kata yang berimbuhan, ini dikarenakan pada kata berimbuhan juga memiliki arti yang berbeda dengan kata yang tidak memiliki imbuhan. Sedangkan *Algoritma Rabin Karp* adalah *algoritma Multiple Pattern Search* yang sangat efisien untuk mencari *String* dengan pola banyak. *Algoritma Rabin Karp* memiliki metode pencocokan dengan metode membanding nilai karakter yang sudah di *Hash*.

Berdasarkan urain di atas ingin membantu membuat perancangan deteksi kemiripan dokumen teks dengan *Algoritma Rabin Karp*.

2. METODE PENELITIAN

2.1. Bentuk penelitian dan teknik pengumpulan data yang digunakan adalah:

2.2. Rancangan Penelitian

Rancangan penelitian yang digunakan adalah Desain Penelitian Hubungan Kausal (Eksperimental) yaitu penulis melakukan percobaan dan pengujian terhadap aplikasi yang dibuat dan dengan cara mempelajari literatur-literatur yang berhubungan dengan materi Perancangan Deteksi Kemiripan Dokumen Teks Dengan Algoritma Rabin Karp.

2.3. Studi Kepustakaan

Studi ini dilakukan dengan cara mencari sekaligus mempelajari beberapa literatur dan artikel mengenai steganografi sebagai acuan dalam perencanaan dan pembuatan penelitian ini.

2.4. Teknik Analisis Data

Teknik analisis sistem yang digunakan penulis dalam menganalisis adalah Unified Modeling Language (UML) yang digunakan untuk menggambarkan alur kerja dari aplikasi perancangan *game* pertualangan.

2.5. Teknik Perancangan Sistem

Dalam perancangan aplikasi steganografi, penulis akan menggunakan bahasa pemrograman *Microsoft Visual Studio* 2010.

2.6. Landasan Teori

a. Plagiat

Plagiarisme adalah tindakan menyerahkan (*submitting*) atau menyajikan (*prensenting*) ide atau kata/kalimat orang lain tanpa menyebut sumbernya.[1] Plagiarisme didefinisikan sebagai tindakan mengambil, mengumpulkan atau menyampaikan pemikiran, tulisan atau hasil karya orang lain selayaknya itu adalah hasil karya diri sendiri tanpa persetujuan dari pemilik hasil karya tersebut.[2]

b. Algoritma Rabin Karp

Rabin Karp ditemukan oleh Michael O. Rabin an Richar M. Karp. Algoritma ini menggunakan metode *Hash* dalam mencari kata. Teori ini jarang digunakan untuk mencari kata tunggal, namun cukup penting dan sangat efektif bila digunakan untuk pencarian jamak.[3] Pada dasarnya, *Algoritma Rabin-Karp* akan membandingkan nilai *Hash* dari *String* masukan dan substring pada teks. Apabila sama, maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya. Apabila tidak sama, maka Substring akan bergeser ke kanan. Kunci utama performa algoritma ini adalah perhitungan yang efisien terhadapat nilai *Hash Substring* pada saat penggeseran dilakukan.Secara garis besar, *algoritma Rabin-Karp* dapat dijelaskan dengan Pseudocode berikut:[4]

c. Hashing

Hashing adalah suatu cara untuk mentransformasi sebuah *String* menjadi suatu nilai yang unik dengan panjang tertentu (*Fixed-Length*) yang berfungsi sebagai penanda string tersebut. Fungsi untuk menghasilkan nilai ini disebut *fungsi Hash*, sedangkan nilai yang dihasilkan disebut nilai *Hash*.[4] Fungsi *Hash* sering disebut dengan fungsi *Hash* satu arah *(one-way function), Message Digest, Fingerprin*t, fungsi kompresi dan *message authentication code(MAC),* merupakan suatu fungsi matematika yang mengambil masukan panjang variabel dan mengubahnya ke dalam urutan *Biner* dengan panjang yang tetap.[5]

d. K-Gram

K-Gram adalah rangkaian terms dengan panjang K. Kebanyakan yang digunakan sebagai *terms* adalah kata. *K-Gram* merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode *K-Gram* ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen.[2]

Dalam Markov Model nilai *K-Gram* yang sering digunakan yaitu, *2-gram (bigram)*, *3-gram (trigram)* dan seterusnya disebut *K-Gram* (*4-gram*, *5-gram* dan seterusnya). Dalam *natural language processing*, penggunaan *K-Gram* (atau lebih dikenal dengan *n gram*), proses *parsing token (tokenisasi)* lebih sering menggunakan *3-gram* dan *4-gram*, sedangkan *2-gram* digunakan dalam *parsing sentence*, misal dalam *partof-speech (POS)*. Penggunaan *2-gram* dalam *tokenisasi* akan menyebabkan tingkat perbandingan antar karakter akan semakin besar. Contohnya pada kata "makan" dan "mana" yang merupakan dua kata yang sama sekali berbeda. Dengan menggunakan metode bigram dalam mencari similarity, hasil dari bigram tersebut yaitu kata "makan" akan menghasilkan bigram ma, ak, ka, an serta kata "mana" akan menghasilkan bigram ma, an, na. Dengan demikian, akan terdapat kesamaan dalam pengecekannya *similarity*. Namun jika menggunakan *3-gram* ("makan" = mak, aka, kan dan "mana" = man, ana) atau 4-gram ("makan" = maka, akan, dan "mana" = mana) akan mengecilkan kemungkinan terjadinya kesamaan pada kata yang strukturnya berbeda.[2]

e. Unified Modeling Language(UML)

UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek.[6] *Unified Modeling Language* (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak.[7]

f. SQL Server Management Studio

SQL *(Structured Query Language)* pada dasarnya adalah bahasa komputer standar yang ditetapkan untuk mengakses dan memanipulasi sistem *database*.[8] SQL Server 2008 adalah RDBMS (*Relational Database Management System*) yang di-*Develop* oleh *Microsoft*, yang digunakan untuk menyimpan dan mengolah *database*.[9]

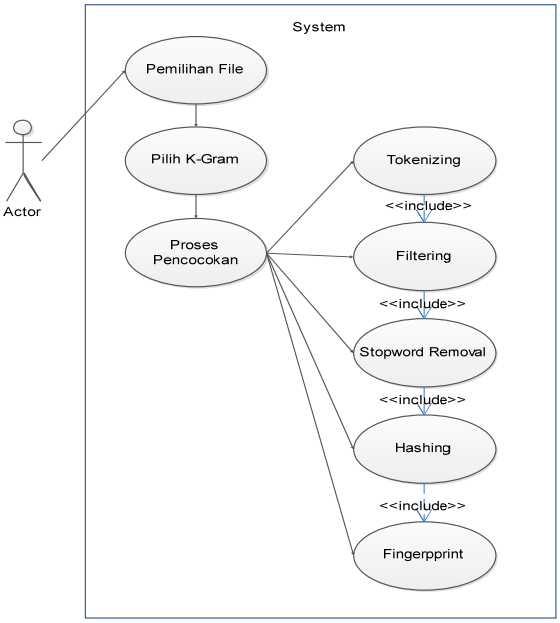
g. Microsoft Visual Studio

Microsoft *Visual Basic (VB. NET)* merupakan bahasa pemograman modern yang dibangun dengan implementasi *Micorosoft .NET Framework*, sebagai pengembangan dari pemograman *Visual Basic* klasik.[10] Visual Basic .NET adalah Visual Basic yang direkayasa kembali untuk digunakan pada *platform .NET* sehingga aplikasi yang dibuat menggunakan *Visual Basic .NET* dapat berjalan pada sistem computer apapun dan dapat mengambil data dari *server* dengan tipe apapun asalkan terinstal *.NET Framework*.[11]

3. HASIL DAN PEMBAHASAN

3.1. Diagram Use Case

Diagram *Use Case* menampilkan interaksi pengguna dan cara kerja deteksi kemiripan dua dokumen teks dengan Algoritma Rabin Karp.

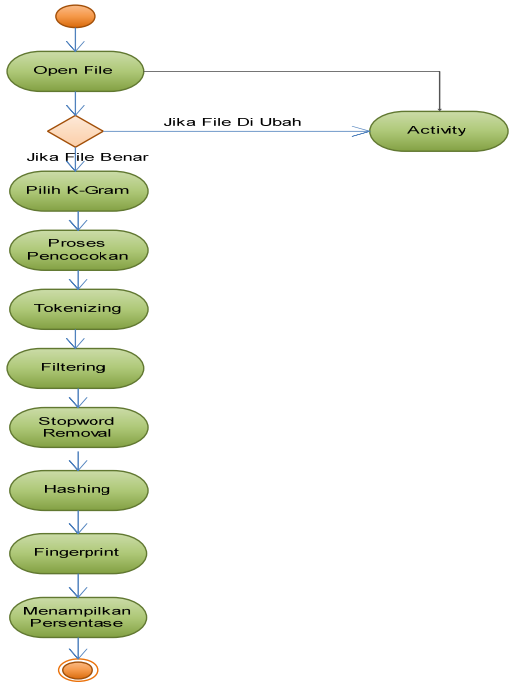


Gambar 1. Use Case Diagram

Pada diagram use case aktor atau pengguna akan berinterkaksi dengan aplikasi dengan melakukan pemilihan file setelah itu memilih k-gram yang digunakan dan aplikasi akan mulai melakukan proses pencocokan.

3.2. Diagram Aktivitas

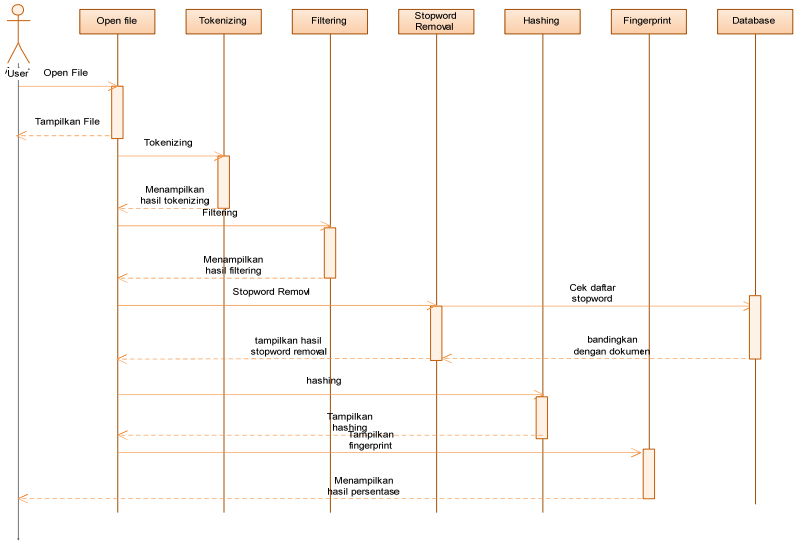
Diagram aktivitas menggambarkan prosedur yang terjadi di dalam deteksi kemiripan dua dokumen teks dengan Algoritma Rabin Karp. Berikut ini adalah diagram aktivitas:



Gambar 2. Diagram Aktivitas

Pada diagram aktivitas deteksi kemiripan dua dokumen teks, pengguna membuka *File 1* dan *File 2* hingga menampilkan dokumen satu dan dua. Selanjutnya jika pengguna ingin menggandi dokumen yang lain maka pengguna menekan tombol hapus dan kembali ke *Open File 1* dan *Open File 2*. Setelah itu pengguna memilih *K-gram* untuk selanjutnya menekan tombol proses. Pada proses, akan melalui tahap *Tokenizing*, *Filtering*, *Stopword Removal*, *Hashing*, dan *Fingerprint* untuk dapat menampilkan hasil persentase kemiripan antara kedua dokumen.

3.3. Diagram Sekuensial



Gambar 3.Diagram Sekuensial

3.4. Implementasi Kemiripan Dua Dokumen Teks

Implementasi merupakan tahap menggunakan dan sekaligus pengujian pada yang telah diperoleh dari menganalisis dan perancangan. Pada bagian ini akan mengimplementasikan dari hasil rancangan hingga menjadi aplikasi Deteksi Kemiripan Dua Dokumen Teks Dengan Alogritma Rabin Karp. Pada tahap ini akan dibahas tampilan dan spesifikasi sitem yang digunakan.

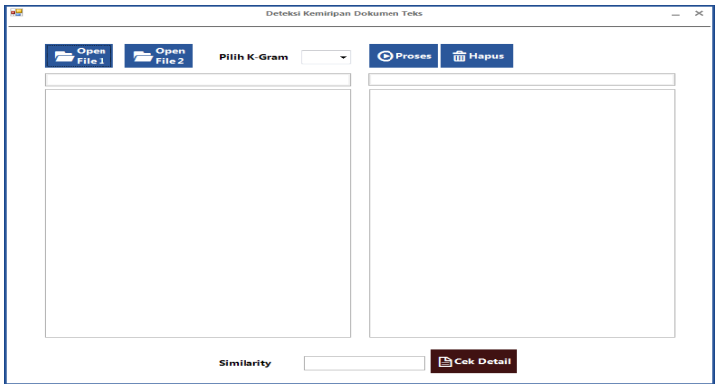
3.4.1. TampilanMenu Utama

Pada tampilan menu utama terdapat pilihan jumlah file yang akan di input dan tombol tentang.

Gambar 4. Tampilan Menu Utama

Pada tampilan menu utama terdapat pilihan untuk memilih jumlah File yang akan di bandingkan dan tombol tentang untuk menampilkan halaman tentang serta tombol keluar untuk menutup aplikasi deteksi kemiripan dokumen teks.

3.4.2. Tampilan Input Dua File



Gambar 5. Tampilan Input Dua File

Halaman *Input* dua *File* berfungsi untuk menampilkan dua dokumen teks untuk dapat dibandingkan dan dapat menemukan kemiripan dua dokumen teks. Halama *Input* dua *File* terdiri dari:

a. *Open File* 1 adalah untuk menampilkan dokumen teks pertama.
b. Open File 2 adalah untuk menampilkan dokumen tkes kedua.
c. Proses adalah untuk memproses kedua dokumen teks agar dapat menemukan kemiripan pada kedua dokumen teks.
d. Hapus adalah untuk menghapus keseluruhan teks pada halaman Input dua File.
e. *Cek Detail* adalah untuk menampilkan detail serta menampilkan proses pada deteksi kemiripan dokumen teks.

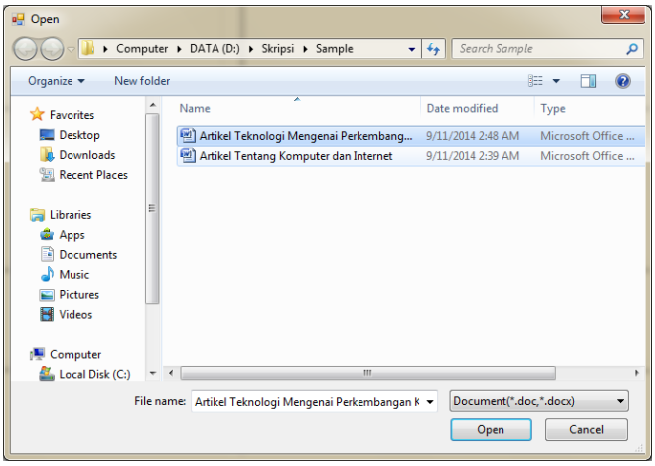
Pada halaman ini juga dapat menginput langsung tanpa perlu menggunakan tombol *Open File*.

3.4.3. Tampilan Input Langsung

Gambar 6.Tampilan Input Langsung

Pada halaman *Input* dua *File*, pengguna dapat melakukan *Input* secara langsung. Fungsi dari *Input* secara langsung agar dapat pengguna dapat memasukan *File* selain file berekstensi *.Doc* atau *.Docx*.

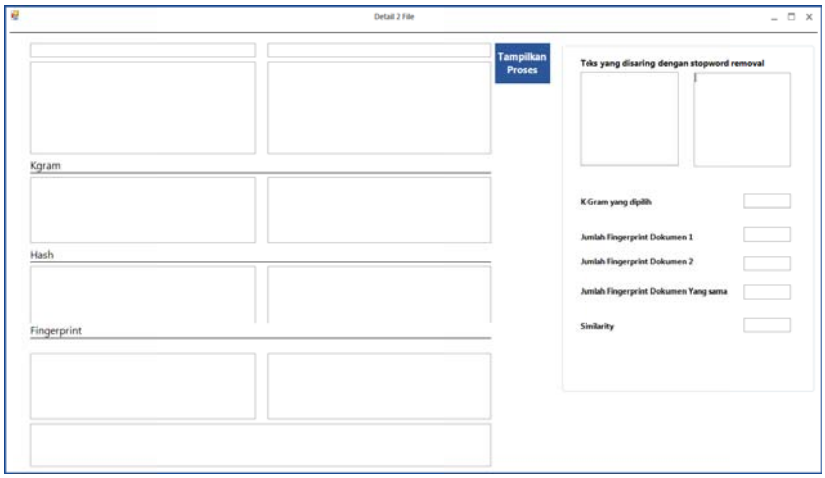
3.4.4. Tampilan Open File



Gambar 7.Tampilan Input Langsung

Tampilan dari tombol *Open File* 1 menampilkan *Open File Dialog* dari *Microsoft Visual Basic .Net* 2010 untuk memilih *File* yang akan ditampilkan atau dibandingkan.

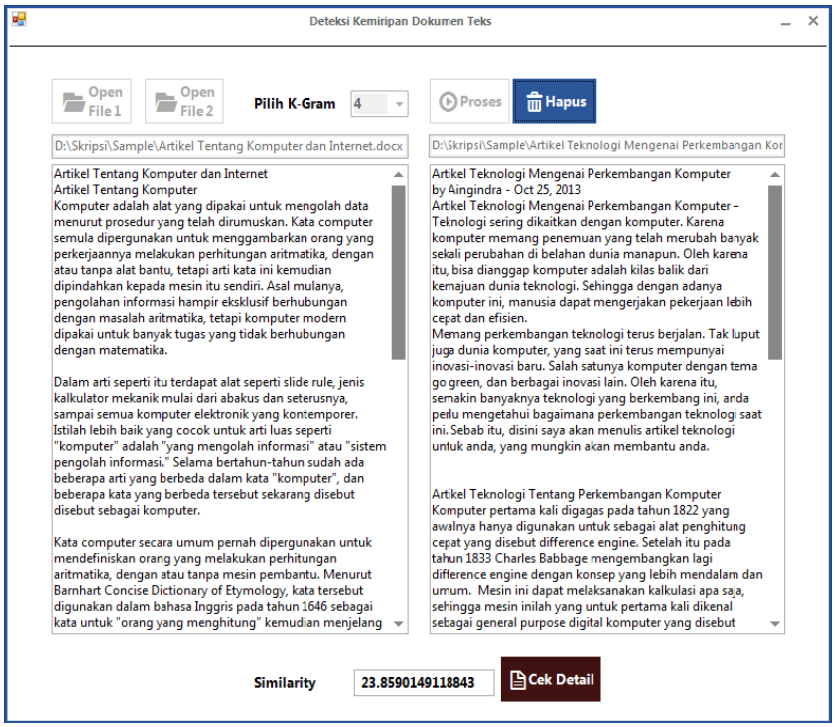
3.4.5. Tampilan Detail



Gambar 8.Tampilan Detail Dua File

Halama detail input dua file menampilkan proses deteksi kemiripan dokumen teks. Proses yang ditampilkan adalah dokumen kedua teks serta *K-gram*, *Hashing*, *Fingerprint*, dan *Similarity*.

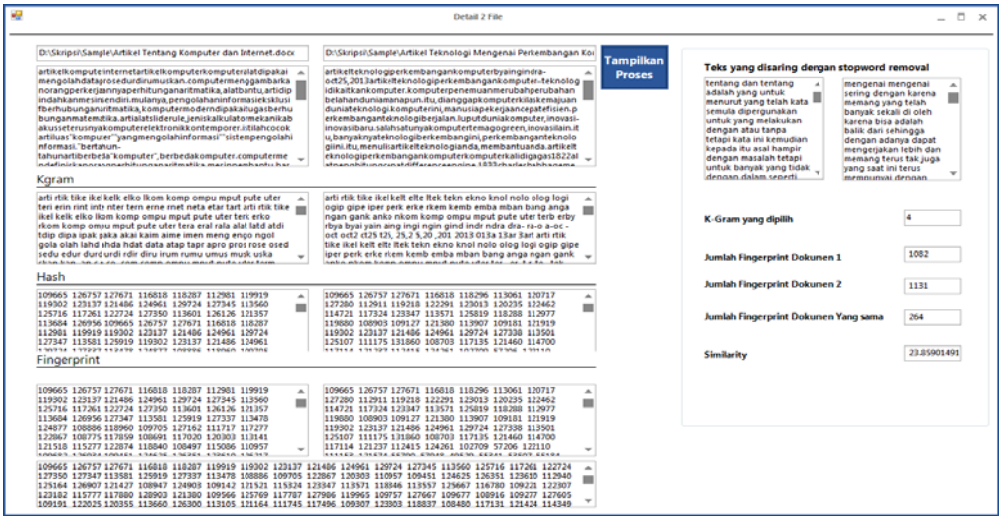
3.4.6. Implementasi Input Dua File



Gambar 10. Tampilan Implementasi Input Dua File

Pengguna mengakses deteksi kemiripan dokumen dengan melalui beberapa tahapan. Tahapan pertama pengguna dapat memilih untuk menginput jumlah *File* yang akan dibandingkan. Setelah memilih salah satu dari memilih jumlah *Input File* maka tahapan ke dua pengguna melakukan *Input File* menggunakan tombol "*Open File*" atau menginput secara langsung. Tahapan ke tiga pengguna memilih *K-gram*. Tahapan ke empat pengguna menekan tombol proses maka hasil dapat ditampilkan. Untuk melihat proses pada dektsi kemiripan dokumen teks pengguna dapat menekan tombol cek detail.

3.4.7. Detail Implementasi Input Dua File



Gambar 11. Tampilan Detail Implementasi Input Dua File

Sebelum aplikasi melakukan proses membandingakan kedua file terlebih dahulu aplikasi melakukan proses text mining yaitu meliputi *Tokenizing, Filtering* dan *Stopword Removal*. Pada proses *Tokenizing*, sebelumnya dilakukan penghapusan tanda baca agar data dapat diproses selanjutnya yaitu *Stopword Removal*. Sedangkan pada *Tokenizing* yaitu pemotongan pada *String Input* dengan acuan spasi hingga menjadi kata perkata. Selanjutnya pada *Stopword* yaitu bertujuan untuk memisahkan teks yang diperlukan dengan teks yang tidak deduktif.

4. KESIMPULAN

Berdasarkan analisi dan pengujian pada sistem deteksi kemiripan dua dokumen teks dengan menggunakan *Algoritma Rabin Karp* maka dapat disimpulkan bahwa:

a. Berdasarkan hasil pengujian, sistem dapat menentukan kemiripan dari dokumen teks yang diuji.
b. Tingkat persentase kemiripan dua dokumen teks dipengaruhi nilai dari k-gram yang telah ditentukan pengguna. Semakin besar nilai k-gram digunakan maka semakin kecil persentase kemiripan dua dokumen teks.
c. Panjang karakter pada dokumen teks yang diuji akan berbengaruh pada waktu proses yaitu cepat atau lambatnya sistem menampilkan persentase kemiripan pada dokumen yang diuji.

5. SARAN

Diharapkan dengan perancangan sistem ini dapat menjadi bahan acuan untuk menentukan tindakan plagiat pada dokumen teks dan disesuikan dengan pandangan pengguna. Jadi dengan adanya sistem ini dapat mencegah terjadinya tindakan palgiat. Adapun saran untuk pengembangan sistem ini yaitu:

a. Sistem dapat menambahkan berbagai jenis file extensi dokumen yang berbeda seperti txt, pdf dan berbagai file extensi dokumen lainnya.
b. Sistem dapat mempersingkat atau mempercepat waktu proses untuk menampilkan kemiripan dokumen.
c. Diharapkan pada proses text mining menggunakan stemming karena stemming dapat menemukan kata dasar pada sebuah kata sehingga dapat memaksimalkan untuk perbandingan pada teks yang akan diuji.
d. Diharapkan dapat membandingkan lebih dari tiga file dengan metode yang sesuai dengan konsep untuk menentukan persentase kemiripan dokumen teks.

UCAPAN TERIMA KASIH

Dalam penulisan penelitian ini, penulis telah banyak mendapat bantuan berupa bimbingan, petunjuk, saran maupun dorongan moril dari berbagai pihak, maka pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada seluruh sivitas akademika sekolah tinggi manajemen informatika dan komputer widya dharma Pontianak.

DAFTAR PUSTAKA

[1] Sastoasmoro, Sudigdo. (Agustus 2007). "*Beberapa Catatan tentang Plagiarisme.*" Majalah Kedokteran Indonesia. Vol. 57, no 8: hal. 239-244.

[2] Surahman, Ade Mirza. (Maret 2013). "Perancangan Sistem Penentuan Similarity Kode Program Pada Bahasa C dan Pascal Dengan Menggunakan Algoritma Rabin-Karp." Jurnal Sistem dan Teknologi Informasi (JustIN) UNTAN. Vol.1, no.1.

[3] Atmopawiro, Alsasian. (2006). "Pengkajian dan Analisis Tiga Algoritma Efisien Rabin-Karp, Knuth-Morris-Pratt, dan Boyer-Moore Dalam Pencarian Pola Dalam Suatu Teks." Program Studi Teknik Informatika, Institut Teknologi Bandung.

[4] Prasetiya, Arif. (2010). "Penerapan Algoritma Boyer-Moore dan Algoritma Rabin-Karp dalam Mendeteksi Aksi Plagiatisme." Progrem Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika. Bandung.

[5] Ariyus, Dony. (2008). Pengantar Ilmu Kriptografi: Teori Analisis dan Implementasi. Andi. Yogyakarta.

[6] Nugroho, Adi. (2011). Perancangan dan Implementasi Sistem Basis Data. Andi. Yogyakarkta.

[7] Sugiarti, Yuni. (2013). Analisis dan Perancangan Unified Modeling Languege (UML). Graha Ilmu. Yogyakarta.

[8] Utami, Ema dan Sukrisno. (2008). Mengoptimalkan Query Pada Sql Server. Andi. Yogyakarta.

[9] Cybertron Solution dan SmitDev Community. (2010). Aplikasi Database dengan Visual Basic 2008 dan SQL Server 2008. Elex Media Komputindo. Jakarta.

[10] Sibero, Alexander F. K. (2010). Dasar-dasar VB .NET. Mediakom. Yogyakarta.

[11] Hidayahtullah, Priyanto. (2010). Visual Basic. NET: Membuat Aplikasi Database dan Program Kreatif. Informatika. Bandung.