

PERANCANGAN DAN PENERAPAN ALGORITMA RIZKI TANJUNG 24 (RTG24) UNTUK KOMPARASI KATA PADA FILE TEXT

Rizki Tanjung, Haruno Sajati, Dwi Nugraheny
Jurusan Teknik Informatika
SekolahTinggi Teknologi Adisutjipto
informatika@stta.ac.id

ABSTRACT

Plagiarism is the act of taking essay or work of others, and recognize it as his own work. Plagiarism of the text is very common and difficult to avoid. Therefore, many created a system that can assist in plagiarism detection text document. To make the detection of plagiarism of text documents at its core is to perform string matching. This makes the emergence of the idea to build an algorithm that will be implemented in RTG24 Comparison file.txt applications. Document to be compared must be a file. Txt or plaintext, and every word contained in the document must be in the dictionary of Indonesian. RTG24 algorithm works by determining the number of same or similar words in any text between the two documents. In the process RTG24 algorithm has several stages: parsing, filtering, stemming and comparison. Parsing stage is the stage where every sentence in the document will be broken down into basic words, filtering step is cleaning the particles are not important. The next stage, stemming is the stage where every word searchable basic word or root word, this is done to simplify and facilitate comparison between the two documents. Right after through the process of parsing, filtering, and stemming, then the document should be inserted into the array for the comparison or the comparison between the two documents. So it can be determined the percentage of similarity between the two documents.

Keywords: *Algorithms RTG24, Komparasi File.TXT, Similarity*

1. Pendahuluan

Plagiat atau plagiarisme menjadi sebuah masalah yang sering terjadi akhir-akhir ini, khususnya di dunia akademik, dan mulai menjadi masalah yang serius karena berkaitan dengan hak kekayaan intelektual (HAKI), Menurut kamus besar bahasa Indonesia (KBBI), plagiat adalah pengambilan karangan atau pendapat orang lain dan menjadikannya seolah-olah karangan atau pendapat sendiri, sedangkan tindakannya disebut plagiarisme. Dengan perkembangan teknologi saat ini, sangat mudah untuk seseorang melakukan praktik plagiat terhadap sebuah file atau dokumen orang lain yang sudah ada, sehingga file tersebut mempunyai nilai kemiripan dengan file yang sudah ada sebelumnya. Oleh karena itu, salah satu cara yang bias dilakukan untuk mengetahui seberapa besar kemiripan dari dua file tersebut adalah dengan cara membandingkan kata demi kata dari kedua file tersebut.

2. Metodologi

2.1 Tinjauan Pustaka

Berdasarkan jurnal yang berjudul Perancangan Sistem Deteksi Plagiarisme Dokumen Teks dengan menggunakan algoritma rabin-karp, mengemukakan similarity (Eko Nugroho, 2012) atau similaritas merupakan tingkat perbandingan persentase kemiripan antar dokumen yang diuji. Algoritma Rabin-Karp adalah algoritma pencocokan string yang menggunakan fungsi hash sebagai pembanding antara string yang dicari (m) dengan substring pada text (n).

2.2 Algoritma RTG24

Algoritma RTG24 merupakan algoritma baru yang diciptakan untuk mencari tingkat kemiripan file dengan cara membandingkan setiap kata yang terdapat dalam kedua file text. Ada beberapa langkah penting dari algoritma RTG24 yaitu parsing, filtering, stemming dan comparing.

2.3 Algoritma Narasi

Langkah-langkah untuk mendapatkan persentase kemiripan antara dua file adalah sebagai berikut :

1. Mulai
2. Parsing setiap kata pada text
3. Hitung kata pada kedua *file*
4. *Filtering* atau proses pembersihan tanda baca
5. Stemming atau mencari kata dasar pada setiap kata
6. Insert kata ke dalam *database*
7. Jumlah kata pada *file text* yang lebih banyak yang akan menjadi *file* pembanding.
8. *Comparing* atau proses perbandingan antara dua *file*.
9. Hitung persentase kemiripan (hitung/pembanding) x 100 %
10. Tampilkan hasil
11. Selesai.

2.4 Pseudocode

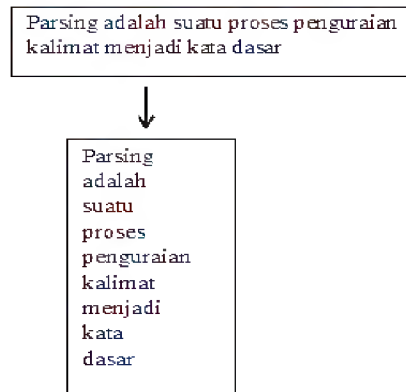
Adapun pada algoritma RTG24 memiliki beberapa proses yang dituangkan dalam Pseudocode sebagai berikut :

1. `$txt_file = file_get_contents('upload/fileA.txt');`
2. `Parsing` ← `$rows = explode(" ", $txt_file);`
3. Hitung kata ← `$jumlah_kata = str_word_count($txt_file);`
4. `Filtering` ← `$kata = preg_replace("/[^a-zA-Z0-9]+/", "", $rows[$i]);`
5. `Stemming`
6. `Comparing` ← `$persentase = round((($hitung / $total_kt) * 100), 2);`
7. Echo "PersentaseKemiripan";

2.5 Parsing

Parsing adalah suatu proses penguraian kalimat menjadi kata. Proses parsing sangat diperlukan untuk perbandingan antara kedua *file*, karena pada proses perbandingan

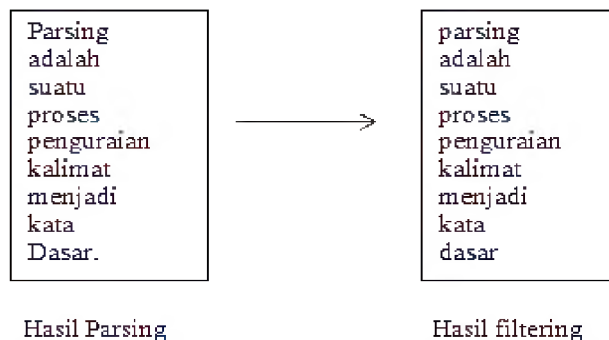
nantinya akan membandingkan setiap kata dasar yang ada dari kedua *file* yang akan dibandingkan.



Gambar 1. Contoh *parsing*

2.6 Proses *Filtering*

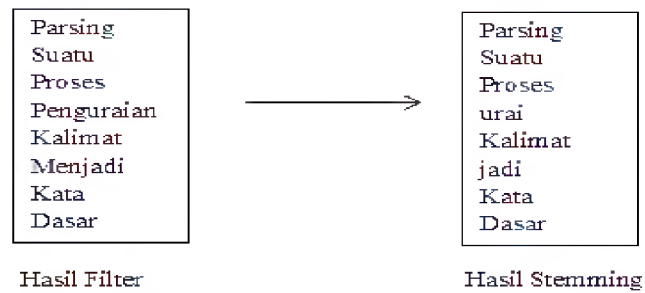
Proses *filtering* adalah proses penghilangan partikel-partikel kata yang tidak penting seperti konjungsi dan menghilangkan tanda baca sehingga didapatkan kata yang penting untuk perbandingan kedua *file*.



Gambar 2. Contoh *Filtering*

2.7 Proses *Stemming*

Stemming adalah tahap mencari *root* kata dari tiap kata hasil filtering. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Setiap kata yang ada di dalam *file* akan dibandingkan dengan tabel kata dasar yang terdapat di dalam *database*, apabila kata tersebut sama seperti kata yang ada di dalam tabel kata dasar yang terdapat di dalam *data base*, maka kata tersebut sudah termasuk kata dasar.



Gambar 3. Contoh *Stemming*

2.8 Proses *Comparing*

Tahap komparasi adalah tahap perbandingan antara kata yang sudah bersih dari tanda baca, kata yang akan dibandingkan di ambil database *file upload*, dengan kata lain kata pembanding dan kata yang akan dibandingkan sama-sama sudah di inputkan terlebih dahulu ke dalam database, sehingga proses perbandingan dilakukan antar tabel. Setiap kata yang ditemukan kesamaannya antara *file A* dan *file B* akan dihitung, sedangkan kata yang tidak sama akan diabaikan. Untuk mencari persentase kemiripan digunakan rumus perhitungan sebagai berikut:

$$\text{Kemiripan file} = \frac{\text{Jumlah Kata yang sama}}{\text{Jumlah Kata pembanding}} \times 100 \%$$

2.9 Uji Algoritma

Algoritma yang baik haruslah melewati proses pengujian lebih dahulu, sehingga dapat diketahui *performance* dari suatu algoritma. Terdapat dua *file* yang akan dibandingkan, isi dari kata kedua *file* tersebut sama banyaknya. Isi kedua *file* tersebut misalnya :

A : Rafael Memukul Bola!.

B : Rafael Mengambil Bola

1. *Parsing*

Setiap kata yang terdapat dalam *file text* akan dipecah menjadi per-kata.

- | | |
|-------------|---------------|
| A1. Rafael | B1. Rafael |
| A2. Memukul | B2. Mengambil |
| A3. Bola!. | B3. Bola |

2. *Filtering*

Setiap kata diubah menjadi *lower case*, kemudian semua tanda baca dihilangkan.

- | | |
|-------------|---------------|
| A1. rafael | B1. rafael |
| A2. memukul | B2. mengambil |
| A3. bola | B3. Bola |

3. *Stemming*

Proses *stemming* adalah proses mencari atau mengembalikan bentuk kata ke dalam kata dasar.

- | | |
|-------------|-------------|
| A.1. rafael | B.1. rafael |
| A.2. pukul | B.2. ambil |
| A.3. bola | B.3. bola |

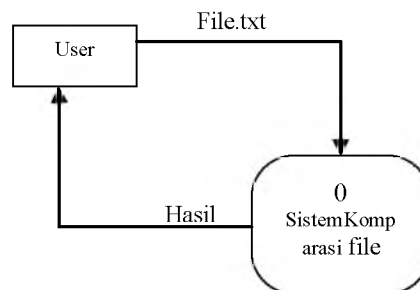
4. Comparing

Setelah melewati proses *parsing*, *filtering*, dan *stemming*, maka proses selanjutnya adalah *comparing*. Pada proses *comparing*, kedua *file* tersebut akan dihitung berapa jumlah kata dasarnya. Dengan demikian dapat dicari persentasenya dengan rumus :

$$\begin{aligned} \text{Kemiripan file} &= \frac{\text{Jumlah Kata yang sama}}{\text{Jumlah Kata pembanding}} \times 100 \% \\ \text{Kemiripan file} &= \frac{2}{3} \times 100 \% \\ &= 66.67 \% \end{aligned}$$

2.10 Diagram Konteks

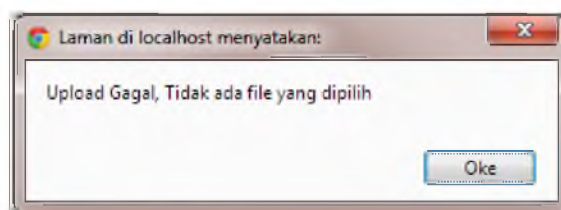
Diagram ini merupakan diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram ini menjelaskan seluruh *input* ke system atau *output* system sehingga dapat memberikan gambaran tentang keseluruhan sistem.



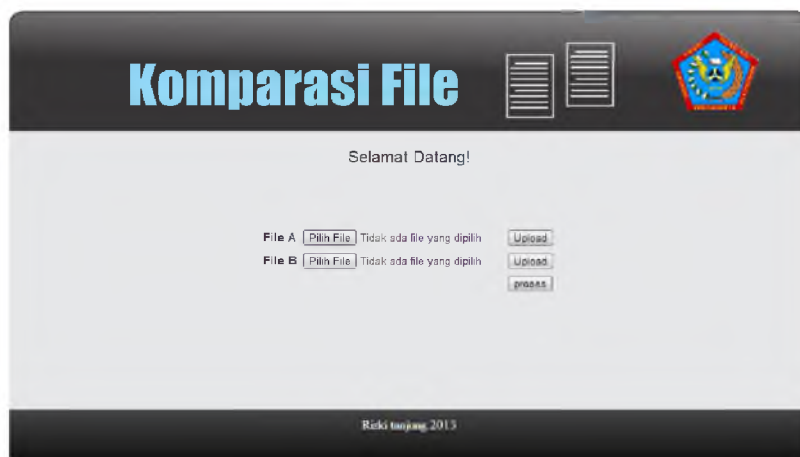
Gambar 4 Diagram Konteks

3 Hasil dan Pembahasan

Pada tampilan utama terdapat dua *uploader* yang berfungsi untuk memilih *file* dan mengunggah *file* yang berekstensi txt atau plaint text akan dibandingkan. Namun jika tidak ada satupun *file* yang dipilih tetapi user sudah menekan *button* upload pada uploader maka akan muncul dialog seperti pada gambar 5.



Gambar 5. Dialog box



Gambar 6. Tampilan Utama Komparasi File



Gambar 7. Tampilan Hasil persentase

Pada pengujian Algoritma RTG24 akan dilakukan beberapa kali percobaan dengan cara melakukan proses perbandingan *file text* sebanyak 8 kali percobaan dengan panjang kalimat yang berbeda-beda pada setiap percobaan.

Tabel 1. Hasil uji perbandingan

No	File A	File B	Persentase kemiripan (%)	Waktu (s)
1	1 kata	1 kata	100	0,021
2	3 kata	3 kata	66,66	0,023
3	5 kata	5 kata	63,64	40
4	8 kata	8 kata	47,37	58
5	25 kata	25 kata	40	65
6	50 kata	50 kata	100	80
7	100 kata	102 kata	28,33	120
8	200 kata	200 kata	5	240

4 Kesimpulan dan Saran

Setelah dilakukan pengujian terhadap aplikasi yang dibuat, maka dapat diambil kesimpulan sebagai berikut :

1. Algoritma RTG24 dapat mencari nilai persentase kemiripan pada *file.txt*.
2. Penggunaan aplikasi Komparasi *File* dapat membantu pengguna untuk menentukan seberapa besar nilai plagiarism antara *file a* dan *file b*.
3. Semakin banyak kata yang terdapat dalam dokumen yang akan dibandingkan semakin lama pula waktu proses upload.

Saran yang dapat diberikan agar aplikasi ini dapat dikembangkan antara lain :

1. Aplikasi yang dihasilkan dapat mencapai waktu yang lebih cepat dan lebih akurat
2. Dapat menyimpan kosa kata yang terdapat pada *file* yang telah diupload, sehingga dapat menjadi pembaharuan kosa kata.

Daftar Pustaka

- Raharjo, Budi. 2011 *Belajar Otodidak Pemrograman WEB dengan PHP + Oracle*. Bandung Informatika
- Stein, B., and Meyer, S. zuEissen. 2006. *Near Similarity Search and Plagiarismm Analysis*. Germany.
- Hartono, Jogiyanto. 2005. *Analisis dan Desain System Informasi: Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis, Edisi ketiga*. Yogyakarta : Andi
- Nugroho, Eko. 2011. *Perancangan Sistem Deteksi Plagiarisme Dokumen Teks dengan Menggunakan Algoritma Rabin Karp*. Program Studi Ilmu Komputer, Universitas Brawijaya Malang.
- Kamus Besar Bahasa Indonesia, 1998.
- Rinaldi Munir, MT, 2004. *Strategi Algoritmik: Seri Diktat Kuliah*, Departemen Teknik Informatika Institut Teknologi Bandung.
- Surahman, Ade Mirza. 2013. *Perancangan Sistem Penentuan Similarity Kode Program Pada Bahasa C Dan Pascal Dengan Menggunakan Algoritma Rabin-Karp*. Program Studi Teknik Informatika, Universitas Tanjungpura.

