

APLIKASI PENDETEKSI PLAGIARISME TUGAS DAN MAKALAH PADA SEKOLAH MENGGUNAKAN ALGORITMA RABIN KARP

Plagiarisme Detection Applications For Tasks and Problems in School Using Rabin Karp Algorithm

Danny Steveson¹⁾, Halim Agung, hagung@bundamulia.ac.id ²⁾, Fendra Mulia³⁾
¹⁾²⁾³⁾Teknik Informatika / Fakultas Teknologi dan Desain, Universitas Bunda Mulia

ABSTRACT

Plagiarism is a very frequent problem in all aspects of one occurring in school. There is often plagiarism on the content of the papers or assignments collected by the students. This is to support the decreasing creativity of students in giving ideas and personal opinions on the task given. To answer the above problems then this research using Rabin-Karp algorithm. Rabin-Karp algorithm is a string search algorithm that uses hashing to find one of a series of string patterns in text. Using this application, the user can compare document 1 with another document, which gives results in sentence similarity, then spelled out per word, followed by per hashing and is calculated from the average number of percentages. The test in this research is done by taking samples 50 times and in comparison between percentage with Rabin Karp algorithm and percentage with manual taking. Testing is done by comparing one document with another document. Based on the result of the research, it can be concluded by using Rabin Karp Algorithm, which can be implemented in plagiarism application evidenced by the test using 50 test samples with 43 samples of success of 14.22%.

Keywords: document, Rabin Karp Algorithm, Dice Sorensen Index, Plagiarism, sentence, word

ABSTRAK

Plagiarisme merupakan permasalahan yang sangat sering terjadi di semua aspek salah satunya terjadi di sekolah. Seringkali terjadi plagiat terhadap konten karya tulis atau tugas yang dikumpulkan oleh siswa. Hal ini menjadi penunjang berkurangnya kreatifitas siswa dalam memberikan ide dan pendapat pribadi pada tugas yang diberikan. Untuk menjawab permasalahan diatas maka penelitian ini menggunakan algoritma Rabin-Karp. Algoritma Rabin-Karp adalah algoritma pencarian string yang menggunakan hashing untuk menemukan salah satu dari serangkaian pola string dalam teks. Dengan menggunakan aplikasi ini, pengguna bisa membandingkan dokumen 1 dengan dokumen yang lain, yang memberikan hasil berupa kesamaan kalimat, kemudian diuraikan dengan per kata, dilanjutkan dengan per hashing dan dihitung dari jumlah rata-rata persentase. Pengujian pada penelitian ini dilakukan dengan pengambilan sampel sebanyak 50 kali dan di bandingkan antara persentase dengan algoritma Rabin Karp dan persentase dengan pengambilan manual. Pengujian dilakukan dengan membandingkan dokumen satu dengan dokumen yang lain. Berdasarkan hasil penelitian, dapat ditarik kesimpulan dengan menggunakan Algoritma Rabin Karp, yang mana dapat diimplementasikan pada aplikasi plagiarisme dibuktikan dengan pengujian menggunakan sampel uji sebanyak 50 kali dengan 43 sampel keberhasilan sebesar 14,22%.

Kata Kunci: dokumen, Algoritma Rabin Karp, Dice Sorensen Index, Plagiarisme, kalimat, kata

PENDAHULUAN

Plagiarisme ada beberapa cara yang dilakukan untuk mengatasi permasalahan ini, yaitu dengan mencegah dan mendeteksi. Mencegah berarti menjaga atau menghalangi agar plagiarisme tidak dilakukan. Mendeteksi berarti melakukan

usaha untuk menentukan tindakan plagiat yang telah dilakukan.

Namun di sisi lain, seringkali terjadi plagiat terhadap konten karya tulis atau bahkan tugas yang di kumpul. Hal ini berujung pada kurangnya kreatifitas siswa dalam menuangkan ide dan pendapat pribadi ke dalam konten yang ingin

disampaikan. Juga permasalahan seringkali terjadi dengan duplikasi hak cipta. Hal ini berdampak pada sisi siswa, dimana nanti ke depan dalam menuangkan pendapatnya akan sulit terutama ketika dimintai pendapat pribadi. Penelitian ini tentunya tidak sendiri. Dengan melihat penelitian pada sebelumnya didapatkan beberapa data dan teori yang mendukung jalannya riset ini. Berikut disampaikan penelitian terdahulu seperti Pembuatan Sistem Penilaian Otomatis Pada Jawaban Ujian Berbentuk Esai Menggunakan Metode *Rabin Karp* [1] yang mendeteksi ujian dari siswa antara jawaban siswa satu dengan siswa yang lain dengan hasil akhir berupa kemunculan merah dan persentase yang berlaku, Pendeteksian Plagiarisme Teks [2] dengan menggunakan algoritma *Smith-Waterman* untuk mendeteksi kesamaan dokumen dan data *noise* untuk menghitung bobot hasil perbandingan, Pembuatan Sistem Deteksi Plagiarisme Dokumen Teks dengan Menggunakan Algoritma *Rabin Karp* [3] yang menghasilkan *K-Gram* untuk mendeteksi kesamaan dokumen dan cara perhitungan lebih menggunakan *K-Gram* serta persentase *error* (toleransi kesalahan), Aplikasi anti plagiarisme dengan algoritma *Karp-Rabin* pada penulisan ilmiah Universitas Gunadarma [4] menunjukkan bahwa algoritma *Rabin-Karp* bertujuan untuk mencari presentase kesamaan dua *file* yang ada.

METODE PENELITIAN

Plagiarisme

Menurut Sastroasmoro [5], plagiarisme adalah penggunaan ide, pikiran, data, kalimat orang lain seolah-olah sebagai miliknya tanpa menyebutkan sumbernya. Plagiarisme merupakan salah satu pelanggaran serius yang bersifat universal terhadap etika akademis (*scientific misconduct*). Tindakan plagiarisme ini merupakan sebuah tindakan tercela, karena merupakan ketidakjujuran alias kebohongan. Perkembangan pesat pada dunia teknologi dan informasi memudahkan kita untuk memperoleh berbagai macam informasi. Internet sangat berguna sebagai

sumber informasi dalam mencari berita, serta mencari referensi untuk membantu pembuatan tugas dan karya tulis. Namun, hal tersebut terkadang dimanfaatkan oleh sebagian orang untuk kepentingannya sendiri, seperti tindakan plagiarisme dalam pengerjaan tugas dan karya tulis. Secara garis besar, plagiarisme dapat dikelompokkan menjadi 2, yaitu plagiarisme ide dan plagiarisme kata demi kata (*word for word plagiarizing*). Derajat plagiarisme yang paling berat adalah *word for word plagiarizing* yakni pencurian kata demi kata, yang dapat mencakup kalimat, paragraf, atau seluruh tulisan [5]. Sedangkan, dalam karya tulis ilmiah, plagiarisme ide sering dihubungkan dengan laporan hasil penelitian replikatif, yaitu penelitian yang secara garis besar mengulang penelitian orang lain, dengan maksud untuk menambah data, menguji hasil hipotesis [5]. Ada beberapa definisi tentang pengertian plagiarisme menurut para ahli, diantaranya:

1. Plagiarisme adalah berbuat sesuatu seolah-olah karya orang lain tersebut adalah karya sendiri dan mengakui hasil karya tersebut [6].
2. Pengertian plagiarisme adalah bentuk penyalahgunaan hak kekayaan intelektual milik orang lain, yang mana karya tersebut direpresentasikan dan diakui secara tidak sah sebagai hasil karya pribadi [7].

Preprocessing

Pengertian *Preprocessing* [8] adalah proses perubahan bentuk data yang terstruktur sembarang menjadi data yang terstruktur sesuai kebutuhan untuk proses dalam text mining. Tahap *preprocessing* terdiri dari *case folding*, *tokenizing*, *filtering*, dan *stemming*. Penelitian ini menggunakan tahap *case folding* hingga *stemming*. *Case Folding* adalah tahap mengubah semua huruf dalam dokumen menjadi huruf kecil. Selain itu, karakter non-huruf akan dihilangkan. *Tokenizing* adalah tahap pemecahan kalimat berdasarkan tipe kata yang menyusunnya. *Filtering* adalah tahap mengambil kata-kata penting dari hasil tahap *tokenizing*. *Filtering* dapat dilakukan dengan

menghilangkan *stoplist/stopword* (kata-kata yang tidak deskriptif, seperti kata “yang” dan “dari”). *Stemming* adalah tahap transformasi suatu kata menjadi kata dasarnya (*root word*) dengan menggunakan aturan-aturan tertentu.

Algoritma Rabin Karp

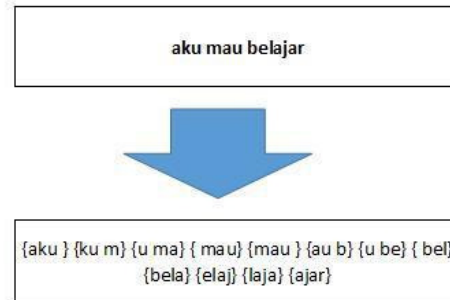
Algoritma *Karp-Rabin* [9] adalah algoritma pencarian *string* yang menggunakan *hashing* untuk menemukan salah satu dari serangkaian pola *string* dalam teks. Untuk teks panjang n dan p pola gabungan panjang m , rata-rata dan kasus terbaik waktu berjalan adalah $(n + m)$ di ruang $O(p)$, tapi kali terburuk adalah $O(nm)$. Sebaliknya algoritma pencocokan *string Aho-Corasick* memiliki *asymptotic* terburuk waktu kompleksitas $O(n + m)$ di ruang $O(m)$. sebuah aplikasi praktis dan algoritma ini mendeteksi plagiarisme. Mengingat bahan sumber, algoritma cepat dapat mencari melalui kertas untuk contoh kalimat dari bahan sumber, mengabaikan rincian seperti kasus dan tanda baca. Karena kelimpahan *string* dicari, algoritma pencarian tunggal *string* tidak praktis. Algoritma *Rabin-Karp* adalah salah satu algoritma yang dapat digunakan untuk mencari dimana sebuah *string* (dalam kasus ini dinamakan sebagai pola) apakah ditemukan di dalam kumpulan *string* lain dengan ukuran yang lebih besar. Contoh yang dibahas kali ini adalah mengenai pencarian kata dari sebuah *input* teks. Untuk mempercepat pencarian pola dalam kalimat, *Rabin-Karp* menggunakan Metode *hash*. Metode ini mengubah setiap *string* menjadi angka, yang dinamakan nilai *hash*. Apabila 2 *string* adalah sama persis, maka nilai *hash*nya juga akan sama, sehingga pencarian *string* dapat diturunkan dengan cara menghitung nilai *hash* dari pola, dan kemudian melakukan pencarian pola dengan nilai *hash* yang sama pada data *input*. Langkah-langkah algoritma *rabin karp* adalah sebagai berikut :

1. Menghilangkan tanda baca dan mengubah ke teks sumber dan kata yang ingin dicari menjadi kata-kata tanpa huruf kapital yang dapat dilihat pada gambar 1.



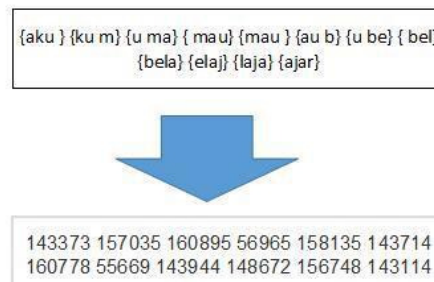
Gambar 1 Proses Algoritma Rabin Karp Pertama

2. Membagi teks ke dalam gram-gram yang ditentukan nilai *k-gram* nya yang dapat dilihat pada gambar 2.



Gambar 2 Proses Algoritma Rabin Karp kedua

3. Mencari nilai *hash* dengan fungsi *rolling hash* dari tiap gram yang terbentuk yang dapat dilihat pada gambar 3.



Gambar 3 Proses Algoritma Rabin Karp ketiga

4. Mencari nilai *hash* yang sama diantara kedua teks.
5. Menentukan persamaan 2 buah teks dengan persamaan *Sorensen Dice Coefficient*. Mengukur kemiripan dan

jarak antara dua entitas. Dalam perhitungan nilai *similarity* yang dapat dilihat pada rumus (1).

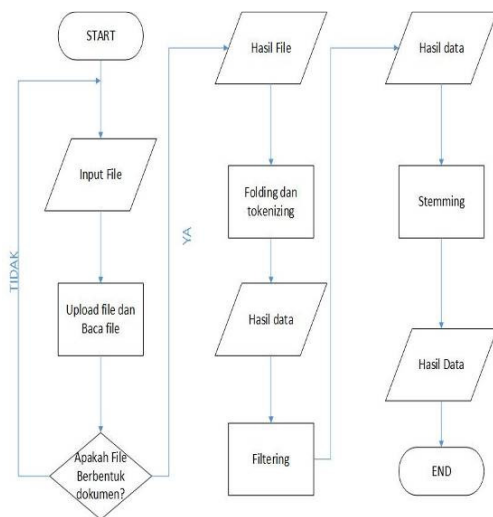
$$S = \frac{K * C}{(A + B)} \times 100\% \dots\dots\dots (1)$$

Metode Sorensen Dice Coefficient

Metode *sorensen dice coefficient* [10] atau nama lainnya *Dice Similarity Coefficient*. Untuk menghitung kemiripan dapat digunakan *Dice Similarity Coefficients* dengan cara menghitung jumlah *K-Gram* yang digunakan pada kedua dokumen yang diuji. Nilai kemiripan tersebut dapat dihitung dengan rumus: dimana S adalah nilai kemiripan, C adalah jumlah *K-Gram* yang sama dan A serta B masing-masing adalah jumlah *K-Gram* dari masing-masing string yang diujikan. Hasil yang didapat adalah bilangan dari 0 sampai 100 dimana bilangan tersebut adalah persentase kemiripan 2 *string* yang diujikan.

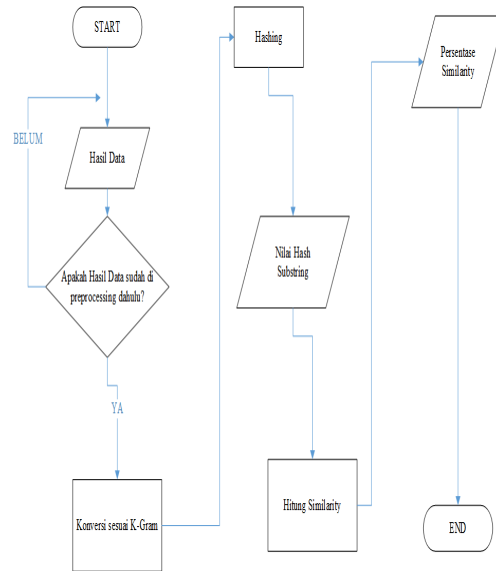
Flowchart Algoritma

Perancangan *flowchart* yang menjelaskan cara kerja programnya dapat dilihat pada gambar 4.

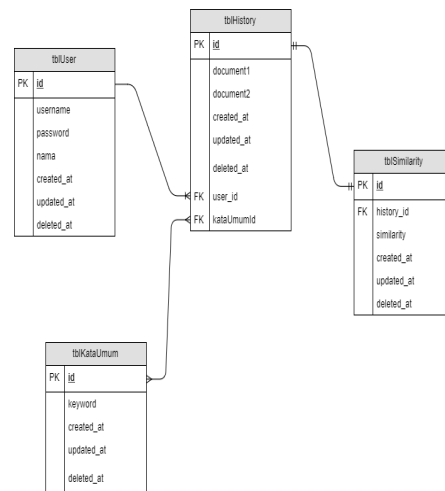


Gambar 4 Flowchart Aplikasi

Kemudian pada proses *flowchart* algoritma *Rabin Karp* dijelaskan secara detail nya pada gambar 5.



Gambar 5 Flowchart Algoritma

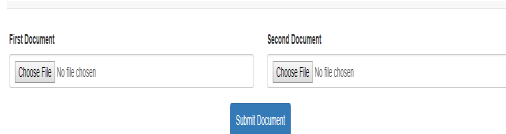


Gambar 6 Entity Relationship Diagram

Entity Relationship Diagram (ERD)

Gambar 6 menjelaskan tentang keterkaitan tabel yang di buat dalam aplikasi ini. Tabel yang dipakai terdiri dari tabel *user*, tabel *history*, tabel kata umum, dan tabel *similarity*. Untuk tabel *user* ke tabel *history* menggunakan *rule one to many*. Sedangkan untuk tabel kata umum dengan tabel *history* menggunakan *rule many to many*, dan yang terakhir tabel

history dengan tabel similarity menggunakan rule one to one.



Gambar 7 Tampilan Awal Aplikasi

HASIL DAN PEMBAHASAN

Pada penelitian yang dilakukan pada aplikasi plagiarisme, mendapatkan hasil desain dan tampilan sesuai dengan kebutuhan. Terlihat pada gambar 7 merupakan tampilan awal pada aplikasi pendeteksian plagiarisme.

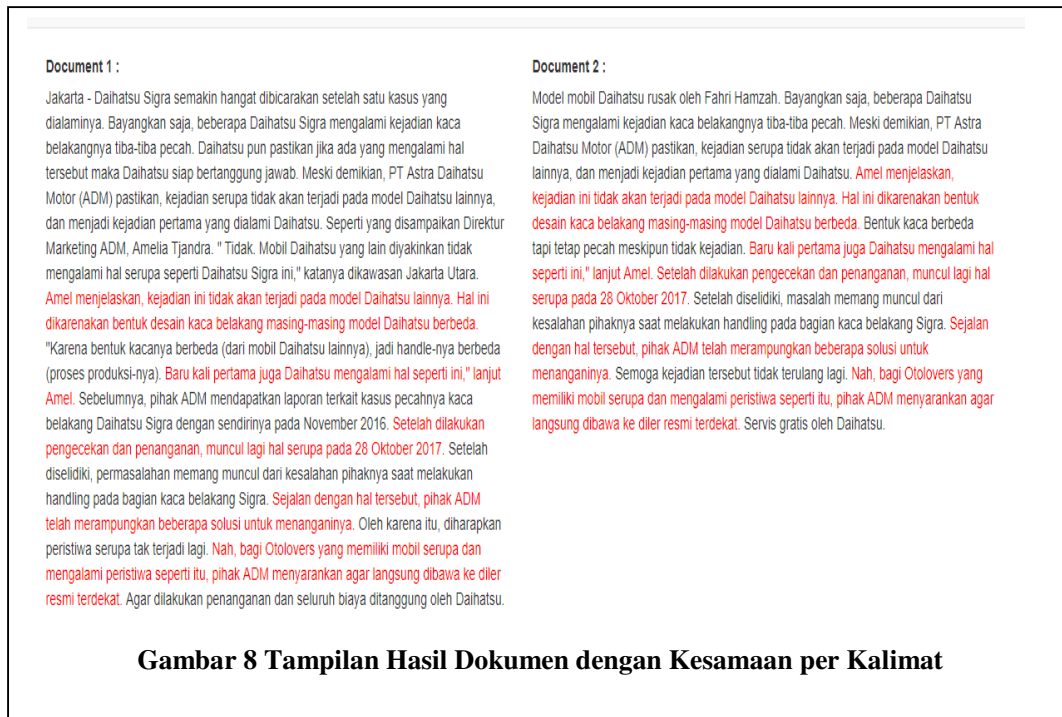
Gambar 7 merupakan tampilan tempat user memasukkan dokumen yang akan dibandingkan baik dokumen sumber maupun dokumen yang dibandingkan. Setelah dokumen – dokumen tersebut dimasukkan maka dokumen tersebut akan langsung memasuki tahap pemilihan dengan melakukan sortir kalimat mana saja yang sama yang terdapat pada dokumen

sumber dan diberi warna merah. Seperti yang terdapat pada gambar 8.

Kemudian dokumen tersebut melalui tahap hashing sesuai dengan proses algoritma rabin-karp yang hasilnya berupa angka hash. hasil hash dari 2 data yang sudah di proses, dan dapat ditemukan persamaan dari 2 data. hasil persentase hasil hash 2 data yang sudah di submit, dan persamaan yang ada menjadi persentase dari Sorensen Index Similiarity yang dapat dilihat pada gambar 9.

Adapun berikut tabel dari hasil 3 dari 50 pengujian yang menunjukkan hasil pengujian antara manual dengan pengujian berdasarkan sistem dengan algoritma Rabin Karp dapat dilihat pada tabel 1.

Berdasarkan hasil pengujian tersebut, dapat ditemukan rata-rata persentase plagiarisme dengan menggunakan algoritma Rabin Karp mendapatkan hasil sebesar 14,21% dan persentase pendeteksian plagiarisme secara manual diperoleh sebesar 13,22%. Adanya perbedaan persentase yang terjadi menghasilkan angka sebesar 0,99%.









Gambar 8 Tampilan Hasil Dokumen dengan Kesamaan per Kalimat

Hasil Sorensen Index Similarity
 3.8461538461538 %

Gambar 9 Hasil Persentase Kesamaan

Tabel 1 Tabel Pengujian Perbandingan antara Perhitungan Menggunakan Algoritma dan Perhitungan Pencarian Manual

NO.	DOC 1	DOC 2	PERSENTASE BY ALGORITHM	PERSENTASE (MANUAL)
1.			7.967	6.192%
2.			1.689%	1.146%
3.			10.526%	9.762%

KESIMPULAN

Berdasarkan hasil penelitian mengenai “Aplikasi Pendeteksi Plagiarisme Tugas dan Makalah Pada Sekolah menggunakan Algoritma *Rabin Karp*” yang di lakukan melalui observasi, wawancara, dan studi dokumenter. Algoritma *Rabin Karp* dapat diimplementasikan pada aplikasi plagiarisme dibuktikan dengan pengujian menggunakan sampel uji sebanyak 50 kali dengan 43 sampel keberhasilan sebesar 14,22%.

DAFTAR PUSTAKA

[1] Lesmana, DI. (2012), Pembuatan Sistem Penilaian Otomatis Pada Jawaban Ujian Berbentuk Esai Menggunakan Metode Rabin Karp, Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim, Malang.

[2] Novanta, A. (2009), Pendeteksian Plagiarisme Teks, PP. 11-26, Universitas Sumatra Selatan, Medan.

[3] Nugroho, E. (2011), Pembuatan Sistem Deteksi Plagiarisme Dokumen Teks dengan Menggunakan Algoritma Rabin Karp, Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Brawijaya Malang, Malang.

[4] Mutiara, dkk. (2014), Aplikasi anti plagiatisme dengan algoritma Karp-Rabin pada penulisan ilmiah Universitas Gunadarma, Skripsi Program Studi Teknik Informatika.

[5] Sastroasmoro, Sudigdo. (2010). Beberapa Catatan tentang Plagiarisme. Departemen Ilmu Kesehatan Anak, Fakultas Kedokteran Universitas Indonesia. Jakarta.

[6] Novanta, A. (2009), Pendeteksian Plagiarisme Teks, PP. 11-26, Universitas Sumatra Selatan, Medan.

[7] Sulianta, F. (2008). Konten Internet, Elexmedia Komputindo, Jakarta. ISBN : 979271788

[8] Feldman, R & Sanger, J. (2007). The Text Mining Handbook : Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press : New York.

[9] Benny, R.Joshi. (2010). Simulated Annealing untuk mencari rute jalan

- terpendek dengan biaya minimal.
Jurnal ilmu komputer 1(2): 234-137
- [10] Nugroho, E. (2011), Pembuatan Sistem Deteksi Plagiarisme Dokumen Teks dengan Menggunakan Algoritma

Rabin Karp, Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Brawijaya Malang, Malang.