



Implementasi Algoritma Rabin-Karp Untuk Pendeteksian Plagiarisme Pada File Dokumen Berupa Text Berbasis Web

Aldian, Mubarak*

Fakultas Teknik, Teknik Informatika, Universitas Pamulang, Tangerang

Jl. Surya Kencana No. 1 Tangerang, Indonesia

Email: ¹aldianws@gmail.com, ²dosen02685@unpam.ac.id

Submitted: 15/03/2022; Accepted: 30/03/2022; Published: 30/04/2022

Abstrak-Plagiarisme selalu menjadi sorotan, tak terkecuali pada sektor akademis baik dari tingkat sekolah hingga jenjang perguruan tinggi. Oleh karena itu pencegahan dan pendeteksian dini perlu dilakukan untuk meminimalisir tindakan plagiarisme ini. Ada beberapa metode yang dapat diimplementasikan salah satunya menggunakan algoritma Rabin-Karp. Algoritma Rabin-Karp merupakan salah satu dari algoritma pencocokan string yang dapat digunakan untuk mengukur tingkat kemiripan teks. Skripsi ini bertujuan untuk merancang dan membangun sebuah aplikasi dengan menggunakan algoritma Rabin-Karp untuk menemukan persentase kemiripan pada dua file dokumen berupa teks yang diuji. Menurut hasil pengujian yang dilakukan oleh [1] antara dokumen asli dan dokumen yang diuji dari hasil pengujian 10 dokumen teks dengan menggunakan algoritma Rabin-Karp, menghasilkan tingkat akurasi yang terbesar yaitu 47.58%. Sedangkan tingkat akurasi yang terkecil yaitu 19.28%. Sedangkan hasil analisis yang penulis lakukan dari 10 dokumen yang diuji dengan nilai k-gram 1, telah didapatkan hasil persentase kemiripan atau similarity terbesar yaitu 57.14% dan yang terkecil sebesar 28.57%. Dimana jika nilai similaritynya <30% termasuk kedalam plagiarisme ringan, 30%-70% plagiarisme sedang dan >70% plagiarisme besar.

Keywords: Plagiarisme; Text Documents; Rabin-Karp Algorithm; Similarity

Abstrak-Plagiarism is always in the spotlight, not least in the academic sector both from school to college level. Therefore, prevention and early detection need to be done to minimize this plagiarism action. There are several methods that can be implemented, one of which uses the Rabin-Karp algorithm. Rabin-Karp algorithm is one of the string matching, algorithms that can be used to measure the level of similarity of text. The thesis aims to design and build an application using the Rabin-Karp algorithm to find the percentage of similarities in two document files in the form of tested text. According to the results of tests conducted by (Priambodo, 2018) between the original document and the document tested from the test results of 10 text documents using the Rabin-Karp algorithm, resulting in the largest accuracy rate of 47.58%. While the smallest accuracy rate is 19.28%. While the results of the analysis conducted from 10 documents tested with a value of k-gram 1, have obtained the largest percentage of similarity, which is 57.14% and the smallest at 28.57%. Where if the similarity value <30% is included in mild plagiarism, 30%-70% moderate plagiarism and >70% large plagiarism.

Kata Kunci: Plagiarism; Dokumen text; Algoritma Rabin-Karp; Similarity

1. PENDAHULUAN

Perkembangan teknologi informasi pada saat ini berkembang begitu pesat, dengan adanya teknologi dapat membantu atau meringankan pekerjaan manusia. Selain itu berkembangnya suatu teknologi tentu ada dampak positif dan negatifnya, salah satu contoh dampak negatifnya adalah orang-orang cenderung ingin sesuatu yang instan, bentuknya yaitu seperti plagiat. Plagiat atau plagiarisme adalah pengambilan karangan, pendapat, dan sebagainya dari oranglain dan menjadikannya seakan-akan karangan/pendapat sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri [1].

Plagiarisme selalu menjadi sorotan, tak terkecuali pada sektor akademis baik dari tingkat sekolah hingga jenjang perguruan tinggi. Tindakan ini sangat tidak mencerminkan seorang yang terpelajar, selain mematican kreativitas juga kebebasan berinovasi menjadi sempit karena kurang menghargai karya orang lain. Selain itu tindakan plagiarisme tidak hanya merugikan orang lain tapi juga merugikan diri sendiri, karena hal tersebut dapat menimbulkan sikap malas dan tak mau berfikir. Oleh karena itu pencegahan dan pendeteksian dini perlu dilakukan untuk meminimalisir tindakan plagiarisme ini. Adapun pelaku palgiarisme dapat terkena sanksi administratif seperti pencepotan gelar dan jabatan juga dapat terkena sanksi pidana atas pelanggaran hak cipta, dapat dijerat Pasal 72 ayat (1) UUHC dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp5.000.000.000,00 (lima miliar rupiah) [2]

Deteksi plagiarisme, duplikasi dokumen dan pencocokan string telah banyak dilakukan pada penelitian-penelitian sebelumnya. Algoritma yang dapat dipergunakan diantaranya *Knuth-Moris-Pratt*, *Brute Force*, *Rabin-Krap*, *Boyer-Moore* dan lain-lain. Namun sebagian besar tanpa menggunakan *preprocessing* sehingga hal tersebut dapat berpengaruh pada hasil *similarity*.

Pada penelitian ini peneliti akan mengimplementasikan algoritma *Rabin-Karp* melalui metode pencocokan *string*. Algoritma *Rabin-Karp* merupakan suatu algoritma pencarian *string* yang ditemukan oleh Michael Rabin dan Richard Karp, algoritma ini menggunakan *hashing* untuk menemukan sebuah *substring* dalam sebuah teks. *Hashing* adalah metode yang menggunakan fungsi *hash* untuk mengubah suatu jenis data menjadi beberapa bilangan bulat sederhana [3]. Deteksi plagiarisme menggunakan konsep *similarity* adalah



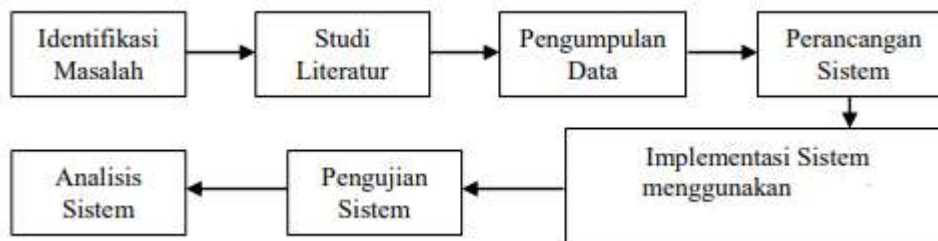
salah satu cara untuk mendeteksi *copy paste* plagiarisme dan *disguised* plagiarisme. Menggunakan algoritma *Rabin-Karp* yang menerapkan metode *fingerprinting* hanya melakukan pencocokan pola *string*. Pendeteksian plagiarisme ini tidak memperhatikan adanya sumber rujukan.

Penelitian-penelitian mengenai pendeteksian plagiarisme yang pertama [4] dengan judul “Perancangan Sistem Pendeteksian Plagiarisme Terhadap Topik Penelitian Menggunakan Metode K-Means Clustering Dan Model Bayesian”. Hasil dari penelitian ini adalah sistem pendeteksian plagiarisme dengan metode K-Means yang dapat mempartisi data kedalam cluster sehingga data yang memiliki karakteristik sama dikelompokkan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik berbeda di kelompokkan ke dalam cluster lain yang kemudian dicari nilai terdekat dari kemiripan antar dokumen. Penelitian lainnya dilakukan oleh [5] dengan judul “Pendeteksian Plagiarisme Menggunakan Algoritma Rabin-Karp Dengan Metode Rolling Hash”. Hasil dari penelitian ini adalah pendeteksian plagiarisme menggunakan algoritma Rabin-Karp dengan metode rolling hash 7 dari hasil pengujian 30 dokumen teks menghasilkan tingkat akurasi yang terbesar yaitu 47.58%. Penelitian lainnya dilakukan oleh [6] dengan judul “Aplikasi Pendeteksian Dugaan Awal Plagiarisme Pada Tugas Siswa Dan Mahasiswa Berdasarkan Kemiripan Isi Teks Menggunakan Algoritma Levenshtein Distance”. Hasil dari penelitian ini dapat ditarik kesimpulan bahwa setelah melalui tahapan preprocessing, algoritma levenshtein distance dapat diimplementasikan pada aplikasi untuk menghitung persentase kemiripan isi teks antar dokumen tugas siswa/mahasiswa, kelemahan dari algoritma levenshtein distance pada aplikasi di penelitian ini adalah tidak dapat mendeteksi plagiarisme ide secara akurat.

Web atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau bergerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman [8]. Web adalah sekumpulan halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet [10]

2. METODOLOGI PENELITIAN

Bab metodologi ini akan dibahas mengenai gambaran umum perancangan sistem pendeteksian plagiarisme pada file dokumen menggunakan algoritma *Rabin Karp*. Secara umum algoritma untuk mendeteksi kesamaan dokumen atau plagiarisme memiliki langkah pemrosesan yang sama untuk dokumen teks. Tahapan tersebut meliputi entry processing dan exit. Mulai dari mengimport dokumen teks yang akan diperiksa kemudian dokumen tersebut akan diproses melalui langkah-langkah *preprocessing* dan *hashing* kemudian menghitung kemiripan dokumen [9] Setelah diproses aplikasi menghasilkan informasi tentang seberapa tingkat *similarity* pada dokumen. Pada tahap ini analisis algoritma dilakukan. Dengan menentukan nilai *k-gram* dan basis bilangan prima. Kemudian hasil *hash* awal dan *hashing* dibandingkan untuk menemukan *hash* yang sama. Jika ditemukan *hash* yang sama maka dari hasil *hash* yang sama tersebut dihitung persentase kemiripan.



Gambar 1. Prosedur Penelitian

Penelitian dimulai dari identifikasi masalah dengan menentukan pertanyaan penelitian. Tahap selanjutnya yaitu studi literatur yaitu mengumpulkan teori-teori yang mendukung penelitian, seperti melakukan proses Rabin Karp. Dilanjutkan pada tahap pengumpulan data, dalam penelitian ini data yang dibutuhkan adalah jurnal untuk dijadikan dokumen repository. Tahap selanjutnya adalah perancangan sistem yaitu untuk memahami alur sistem yang akan dibuat dan mengimplementasikan pada pengambilan konten di setiap jurnal serta metode Rabin-Karp untuk menghitung kemiripan teks dalam proses deteksi plagiarisme. Algoritma Rabin-Karp adalah salah satu algoritma pencarian string dikembangkan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 yang menggunakan fungsi hashing untuk menemukan pattern di dalam string teks[7]

3. HASIL DAN PEMBAHASAN

Implementasi merupakan kelanjutan dari kegiatan perancangan sistem dan dapat dilihat sebagai upaya untuk merealisasikan sistem yang telah dirancang. Pada tahap ini, proses yang diimplementasikan adalah implementasi *hardware* dan *software* untuk mengimplementasikan dan menguji sistem yang dirancang. Hasil dari fase ini adalah sistem pemrosesan data dan informasi yang berfungsi dengan baik.



1. Spesifikasi Perangkat Keras Yang Digunakan

Adapun perangkat keras yang digunakan pada penelitian ini adalah *Personal Computer* atau Laptop yang digunakan dengan spesifikasi yaitu Layar: 14.0” (16:9) *LED-backlit FHD* (1366X768), *OS: Windows 10 Home*, *Prosesor: Intel Core i5 8250U*, *RAM: 4 GB DDR4*.

2. Spesifikasi Perangkat Lunak Yang Digunakan

Selain kebutuhan *hardware* (perangkat keras), diperlukan pula *software* (perangkat lunak) yang harus dipenuhi dalam penelitian ini sehingga *software* tersebut dapat sesuai dengan maksud dan tujuan dalam penelitian. Adapun *software* yang dibutuhkan adalah sebagai berikut: *Windows 10*, *MySQL*, *Xampp*, *Sublime Text*, *Web Browser*, *Microsoft Office*.

3. Sistem Interface



Gambar 2. Tampilan Halaman Login

Halaman login merupakan halaman pertama saat aplikasi ini dijalankan. Pada halaman ini, pengguna dapat memasukkan nama pengguna dan kata sandi. Kemudian langkah selanjutnya adalah klik tombol Sign in. Jika pengguna benar memasukkan username dan password maka sistem akan menampilkan halaman dashboard, namun jika salah sistem akan menampilkan pesan error berupa “username/password tidak sesuai”



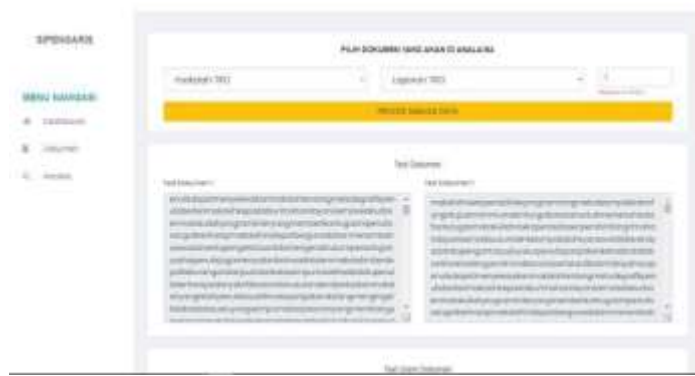
Gambar 3. Tampilan Halaman Dashboard

Halaman dashboard merupakan halaman yang muncul setelah admin berhasil melakukan proses login. Pada halaman ini juga terdapat info terkait algoritma Rabin-Karp.



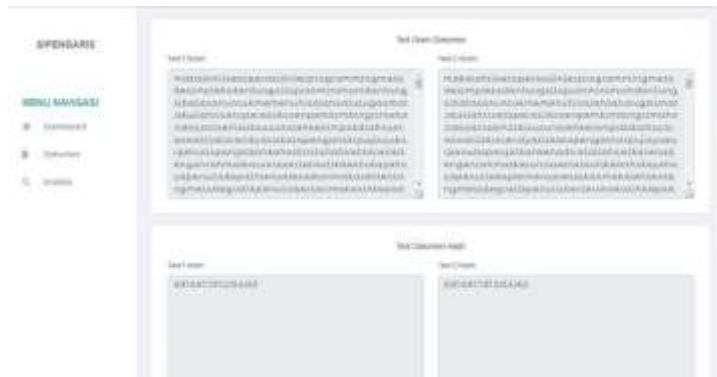
Gambar 4. Tampilan halaman kelola dokumen

Halaman kelola dokumen merupakan halaman yang digunakan admin untuk mengelola data dokumen seperti menambah, mengubah, menghapus juga mencari data dokumen pada sistem pendeteksian plagiarisme pada file dokumen text menggunakan algoritma Rabin-Karp ini.



Gambar 5. Tampilan Halaman Analisa Dokumen

Halaman analisa dokumen merupakan halaman yang digunakan admin untuk menganalisa dokumen, dimana admin menginputkan dokumen yang akan dianalisa dan juga menginputkan berapa k-gram yang akan dianalisa kemudian sistem akan melakukan proses preprocessing dan juga hasing menggunakan algoritma Rabin-Karp.



Gambar 6. Tampilan Halaman Hasil Analisa

Tabel kedua adalah hasil proses hashing dari algoritma Rabin-Karp tersebut.



Gambar 7. Tampilan halaman hasil analisa lanjut

Setelah analisa selesai sistem akan menampilkan hasil berupa presentase similarity tersebut yang mana admin dapat mengetahui seberapa mirip dokumen yang dianalisa dengan database dokumen yang sudah ada dalam sistem ini.

3.1 Pengujian

1. Black Box Testing

Pada tahap ini dilakukan pengujian terhadap sistem yang sudah dibuat. Dengan uji *black box* ini diharapkan jika terdapat kesalahan atau kekurangan pada aplikasi, peneliti dapat segera menemukan hal tersebut secepatnya.

Tabel 1. Pengujian black box testing

Skenario Pengujian	Kasus Pengujian	Hal yang Diharapkan	Hasil Pengujian	Kesimpulan
Login	Input, Username dan Password	Sistem dapat melakukan login	Sesuai	Berhasil
Kelola	Tambah, ubah,	Data dokumen dapat ditampilkan dan	Sesuai	Berhasil



Skenario Pengujian	Kasus Pengujian	Hal yang Diharapkan	Hasil Pengujian	Kesimpulan
dokumen	hapus, dan cari dokumen	diperbaharui dengan cara ditambah, diubah dan dihapus, selain itu juga dapat mencari data dokumen		
Analisa dokumen	Input dokumen yang akan dianalisa dan juga K-Gram	Dapat menampilkan hasil analisa dengan menggunakan algoritma Rabin Krapp	Sesuai	Berhasil

4. KESIMPULAN

Berdasarkan analisis, desain/perancangan dan implementasi pada sistem pendeteksian plagiarisme pada file dokumen text menggunakan algoritma Rabin-Karp dapat disimpulkan Telah dibuat suatu sistem berbasis web yang dapat digunakan untuk pendeteksian plagiarisme pada file dokumen berupa text dengan menggunakan algoritma Rabin-Karp. Telah didapatkan hasil analisa perbandingan persentase kemiripan (similarity) antara dokumen asli dan dokumen yang diuji dari hasil pengujian 10 dokumen teks dengan menggunakan algoritma Rabin-Karp, menghasilkan tingkat akurasi yang terbesar yaitu 47.58%. Sedangkan tingkat akurasi yang terkecil yaitu 19.28% (Priambodo, 2018). Sedangkan hasil analisis yang penulis lakukan dari 10 dokumen yang diuji dengan nilai k-gram 1, telah didapatkan hasil persentase kemiripan atau similarity terbesar yaitu 57.14% dan yang terkecil sebesar 28.57%. Dimana jika nilai similarity <30% termasuk kedalam plagiarisme ringan, 30%-70% plagiarisme sedang dan >70% plagiarisme besar.

REFERENCES

- [1] Bansal, "An Elementary Algorithm For Pattern Matching," *International Jurnal*, vol. IV, 2018:1782.
- [2] Triyono. "Teknik Perancangan Arsitektur dan Sistem Informasi". Jumantaka. vol. IV, 2018:45
- [3] T. d. Safitri, "Perancangan Sistem Informasi Absensi Guru dan Staff Padasmik Pacakarya Tangerang Berbasis Web," *Jurnal Sensi*, vol. Iv, pp. 53-67, 2018:156.
- [4] F. d. Hayaty, "Implementasi Algoritma Rabin-Karp Untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa., , .," *Juita*, vol. p. VII, 2019:25.
- [5] Lopes, "Penegakan Hukum Terhadap Planggaran Hak Cipta di Bidang Musik dan Lagu," *Lex Privatum*, vol. III, 2013.
- [6] Priambodo, "Pendeteksi Plagiarisme Menggunakan Algoritma Rabin-Karp Dengan Metode Rolling Hash," *Jurnal Informatika Universitas Pamulang*, vol. III, pp. 39-45, 2018:39.
- [7] Basyian. dkk, "Perancangan sistem Pendeteksi Plagiarisme Terhadap Topik Penelitian Menggunakan Metode K-Means Clustering dan Model Bayesian," *Seminar Nasional Teknologi Informasi dan Multimedia*, vol. II, pp. 209-222, 2018.
- [8] N. d. Anna, "Rancang Bangun Sistem Informasi Pendeteksi Plagiarime pada dokumen Skripsi," *Evolusi* , vol. VII, p. 6, 2018.
- [9] A. d. Bangsa, "Aplikasi Pendeteksi Dugaan Awal Plagiarisme Pada Tugas Siswa dan Mahasiswa Berdasarkan Kemiripan Isi Teks Mengunakan Algoritma Levenshtein Distance," *Jurnal Pinter*, vol. IV, pp. 25-33, 2017.
- [10] Z. d. Azhari, "Web Bassed Aplications In Calculation Of Family Heritage (Science Of Faroidh)," *Sistem Informasi*, vol. I, pp. 50-60, 2020.