

DETEKSI *PLAGIARISME* PADA FILE DOKUMEN BERDASARKAN TINGKAT KESAMAAN DENGAN MENGUNAKAN METODE ALGORITMA *RABIN-KARP* BERBASIS *WEB*

Hardison^{1*}, Maulana Ardhiyansyah¹

¹Fakultas Teknik, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46,
Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: ^{1*}hardisonpradana2@gmail.com, ²maulanaunpam2012@gmail.com

(* : coressponding author)

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 (Priambodo, 2018) antara dokumen asli dan dokumen yang diuji dari hasil pengujian 10 dokumen teks dengan menggunakan algoritma RabinKarp, 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.

Kata Kunci: Plagiarisme, Dokumen text, Algoritma Rabin-Karp, Similarity

Abstract–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.

Keywords: Plagiarism, Text Documents, Rabin-Karp Algorithm, 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 orang lain dan menjadikannya seakan-akan karangan/pendapat sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri (Filcha & Hayaty, 2019).

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 mematikan 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.

Deteksi plagiarisme, duplikasi dokumen dan pencocokan string telah banyak dilakukan pada penelitian-penelitian sebelumnya. Algoritma yang dapat dipergunakan diantaranya Knuth-Morris-Pratt, Brute Force, 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 RabinKarp 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 (Priambodo, 2018).

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. Dengan mempertimbangkan pemaparan yang telah dituliskan pada paragraf-paragraf sebelumnya, dengan ini penulis mengambil judul “**Deteksi Plagiarisme Pada File Dokumen Berdasarkan Tingkat Kesamaan Dengan Menggunakan Metode Algoritma Rabin-Karp Berbasis Web**”.

2. METODOLOGI PENELITIAN

2.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan adalah metode studi pustaka dan Studi Literatur.

- a. Studi Pustaka (*Library Research*), dilakukan untuk memperoleh data sekunder, dengan melakukan penelaahan teori-teori yang berkaitan dengan topik penelitian yang berasal dari sumber-sumber penelitian kepustakaan yaitu buku, jurnal, hasil-hasil penelitian terdahulu yang telah dipublikasikan (tesis dan disertasi) dan sumber lainnya (internet, surat kabar dan lain-lain) yang sesuai dengan topik penelitian.
- b. Studi literatur dilakukan untuk mempelajari sistem informasi dan metode apa yang bisa digunakan dalam pendeteksian plagiarisme dari berbagai media seperti jurnal, buku dan sumber lainnya.

2.2 Metode Pengembangan Sistem

Model Waterfall adalah “model menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, dan pengujian” (Anton, Astuti, & Risaldi, 2020).

2.3 Algoritma Rabin-karp

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 (Priambodo, 2018:39). 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 (Bansal, 2018:1782).

Berdasarkan definisi di atas dapat disimpulkan bahwa Algoritma RabinKarp merupakan algoritma yang dikemukakan oleh Michael Rabin dan Richard yang menggunakan hashing untuk menemukan pattern dalam sebuah teks.

2.4 Prinsip Kerja Algoritma Rabin-Karp

Rabin Karp merepresentasikan setiap karakter ke dalam bentuk desimal digit (digit radix-d) $\Sigma = \{0, 1, 2, 3, \dots, d\}$, dimana $d = |\Sigma|$. Sehingga didapat masukan string k berturut-turut sebagai perwakilan panjang k desimal. Karakter string 31415 sesuai dengan jumlah desimal 31,415. Kemudian pola p di-hash menjadi nilai desimal dan string direpresentasikan dengan penjumlahan digit-digit angka menggunakan aturan Horner's, misal.

3. ANALISA DAN PEMBAHASAN

3.1 Analisa Deteksi Dokumen

Pada bab ini akan dibahas mengenai gambaran umum perancangan sistem pendeteksi 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. Setelah diproses aplikasi menghasilkan informasi tentang seberapa tingkat similarity pada dokumen.

3.2 Analisa Algoritma Rabin-Karp

Pada tahap ini analisis algoritma dilakukan. Dengan menentukan nilai kgram 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. Langkah-langkah algoritma Rabin Karp diimplementasikan dalam perancangan sistem sebagai berikut:

- a. Menghilangkan tanda baca.
- b. Membagi teks dalam bentuk k-gram, dimana nilai k adalah sebagai nilai parameter yang dipilih.
- c. Hitung nilai hash pada setiap k-gram.
- d. Pilih nilai hash yang sama.
- e. Hitung presentase kemiripan dua dokumen.

3.3 Analisa Kebutuhan

Penelitian ini akan menghasilkan suatu aplikasi website yang digunakan untuk menganalisa dokumen dalam hal kemiripan.

- a. Sistem akan terhubung ke web yang akan menampilkan data hasil Analisa.
- b. Sistem terhubung kepada server databases melalui koneksi internet dari perangkat pengguna.

3.4 Gambaran Umum Sistem

Gambaran umum sistem yang dikembangkan mempunyai 4 komponen yaitu pengguna, sistem, server dan database. Secara garis besar gambaran umum sistem deteksi kemiripan kata pada dua dokumen menggunakan Algoritma Rabin-Karp.

Tahapan proses sistem deteksi kemiripan kata yaitu sebagai berikut:

1. Input Dua Dokumen

Memasukkan dua dokumen, yaitu dokumen asli dan dokumen uji. Dokumen asli sebagai pola (pattern) atau dokumen dasar yang digunakan sebagai acuan kemiripan dokumen uji. Dokumen uji sebagai teks atau dokumen yang akan diuji kemiripannya berdasarkan dokumen asli. Dokumen yang dimasukkan berekstensi (*. docx). Pembacaan isi dokumen yang diproses terdiri dari huruf dan angka.

| Dokumen | Isi Dokumen |
|--------------|--|
| Dokumen asli | Algoritma yang digunakan adalah Rabin-Karp. Hasil persentase dan keberadaan kalimat. |
| Dokumen uji | Hasil persentase dan keberadaan kalimat. |

Gambar 1. Dokumen Yang Di Input

2. Tahap Praproses (Text Preprocessing)

Setelah dokumen asli dan dokumen uji dimasukkan, maka akan dilakukan tahap praproses (text preprocessing). Text preprocessing merupakan tahap awal memproses isi dokumen. Tahap ini terdiri dari proses filtering, whitespace insensitivity, dan parsing k gram:

a. Filtering

Proses ini menghilangkan kata-kata yang tidak penting (stopwords) pada isi dokumen.

| Dokumen | Isi Dokumen |
|--------------|--|
| Dokumen asli | Algoritma yang digunakan adalah Rabin-Karp. Hasil persentase dan keberadaan kalimat. |
| Dokumen uji | Hasil persentase dan keberadaan kalimat. |

Gambar 2. Hasil *Filtering*

b. Whitespace Insensitivity

Proses ini menghapus tanda baca, spasi dan mengubah menjadi huruf kecil (lowercase) pada isi dokumen.

| Dokumen | Isi Dokumen |
|--------------|---|
| Dokumen asli | algoritmarabinkarhasilpersentasekeberadaankalimat |
| Dokumen uji | hasilpersentasekeberadaankalimat |

Gambar 3. Hasil *Whitaspace Insensitivity*

c. Parsing K-gram

Proses ini memecah isi dokumen yang sudah melalui proses filtering dan whitespace insensitivity. Nilai k yang ditentukan 5-gram, Dokumen hasil Whitespace insensitivity dipecah menjadi 5 ukuran kata atau parse.

| No | Term | No | Term | No | Term |
|----|-------|----|-------|----|-------|
| 1 | algor | 17 | rphas | 33 | ekebe |
| 2 | lgori | 18 | phasi | 34 | keber |
| 3 | gorit | 19 | hasil | 35 | ebara |
| 4 | oritm | 20 | asilp | 36 | berad |
| 5 | ritma | 21 | silpe | 37 | erada |
| 6 | itmar | 22 | ilper | 38 | radaa |
| 7 | tmara | 23 | lpers | 39 | adaan |
| 8 | marab | 24 | perse | 40 | daank |
| 9 | arabi | 25 | ersen | 41 | aanka |
| 10 | rabin | 26 | rsent | 42 | ankal |
| 11 | abink | 27 | senta | 43 | nkali |
| 12 | binka | 28 | entas | 44 | kalim |
| 13 | inkar | 29 | ntase | 45 | alima |
| 14 | nkarp | 30 | tasek | 46 | limat |
| 15 | karph | 31 | aseke | | |
| 16 | arpha | 32 | sekeb | | |

Gambar 4. Hasil *Parsing K-gram*

4. IMPLEMENTASI

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.

a. 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

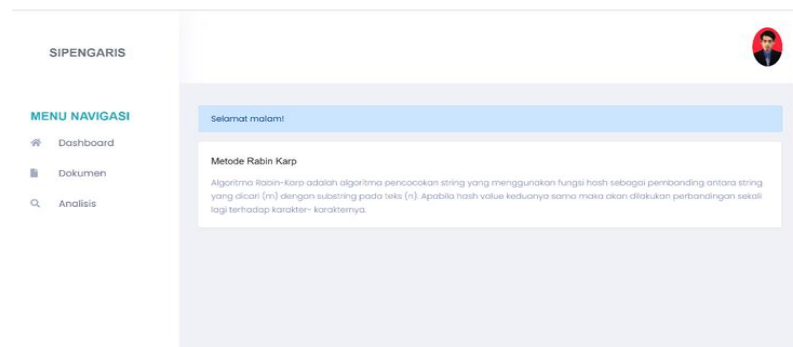
“username/password tidak sesuai” Form halaman login terlihat seperti pada gambar



Gambar 5. Tampilan Halaman *Login*

b. Tampilan Halaman *Dashboard*

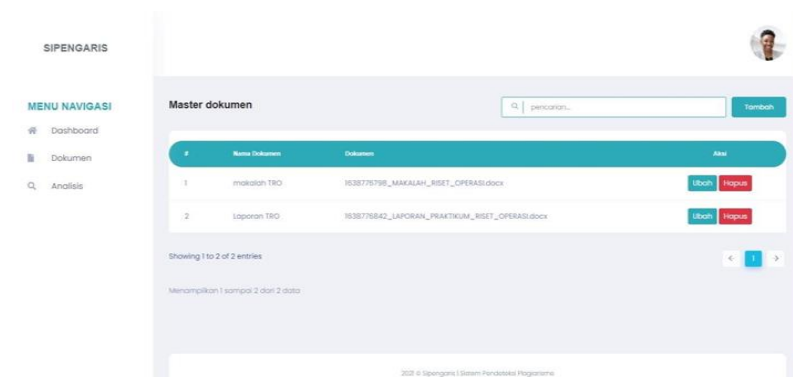
Halaman dashboard merupakan halaman yang muncul setelah admin berhasil melakukan proses login. Pada halaman ini juga terdapat info terkait algoritma Rabin-Karp. Halaman dashboard admin terlihat seperti pada gambar



Gambar 6. Tampilan Halaman *Dashboard*.

c. Tampilan Halaman *Kelola Dokumen*

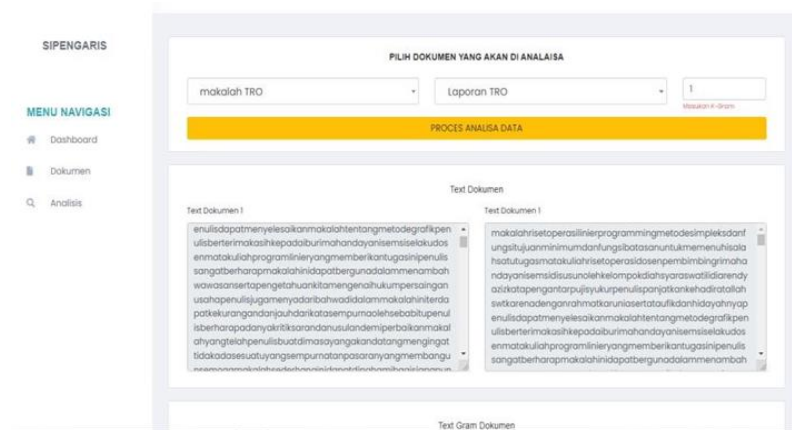
Halaman kelola dokumen merupakan halaman yang digunakan admin untuk mengelola data dokumen seperti manambah, mengubah, menghapus juga mencari data dokumen pada sistem pendeteksian plagiarisme pada file dokumen text menggunakan algoritma Rabin-Karp ini. Halaman kelola judul terlihat seperti pada gambar



Gambar 7. Tampilan Halaman *Kelola Dokumen*

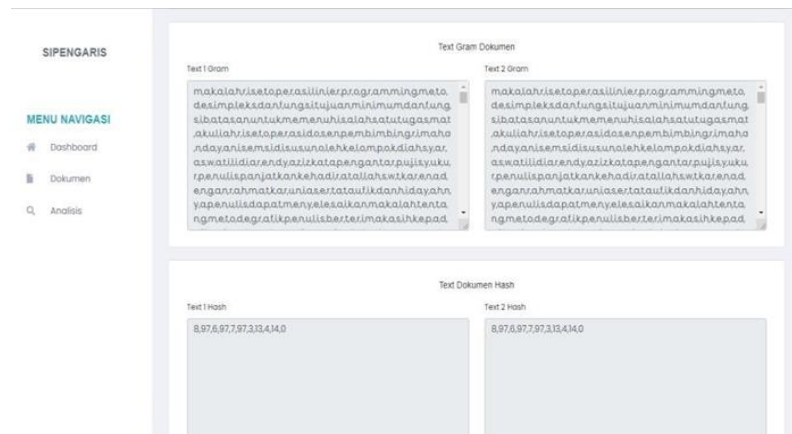
d. Tampilan Halaman Analisa Dokumen

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.



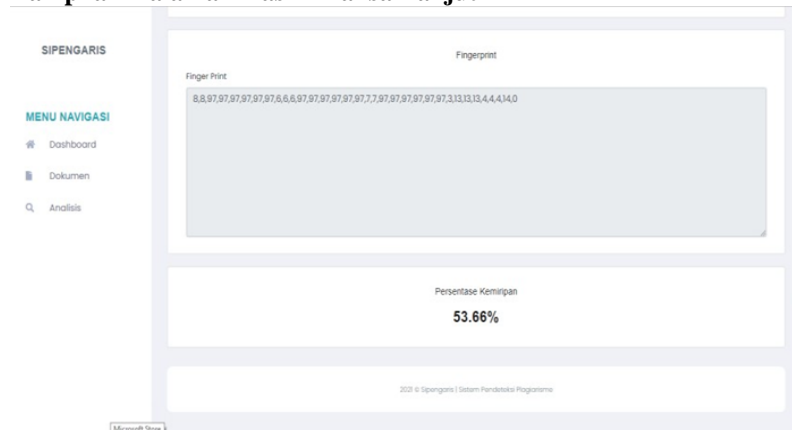
Gambar 8. Tampilan Halaman Analisa Dokumen

e. Tampilan Halaman Hasil Analisa



Gambar 9. Tampilan Halaman Hasil Analisa

f. Tampilan Halaman Hasil Analisa Lanjut



Gambar 10. Tampilan Halaman Hasil Analisa Lanjut

5. KESIMPULAN

Berdasarkan analisis, desain/perancangan dan implementasi pada sistem pendeteksian plagiarisme pada file dokumen text menggunakan algoritma Rabin-Karp dapat disimpulkan bahwa:

- a. Telah dibuat suatu sistem berbasis web yang dapat digunakan untuk pendeteksian plagiarisme pada file dokumen berupa text dengan menggunakan algoritma Rabin-Karp.
- b. Telah didapatkan hasil analisa perbandingan persentase kemiripan (similarity) antara dokumen asli dan dokumen yang diuji dari hasil pengujian 10 dokumen teks dengan menggunakan algoritma RabinKarp, 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

- Anton, Astuti , P., & Risaldi, M. A. (2020). PERANCANGAN SISTEM INFORMASI MENGGUNAKAN METODE WATERFALL UNTUK PENGAJUAN CUTI DAN PERJALANAN DINAS PADA PT.IGTAX EKUSERU INDONESIA. *JURNAL BUFFER INFORMATIKA*.
- Filcha, A., & Hayaty, M. (2019). Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa. *Implementasi Algoritma Rabin-Karp ...* , 26.
- Priambodo, J. (2018). Pendektesian Plagiarisme menggunakan Algoritma RabinKarp dengan Metode Rolling Hash. *Jurnal Informatika Universitas Pamulang*, 39-45.
- Puspita, W. (2018). *Manajemen Konflik: Suatu Pendekatan Psikologi, Komunikasi, dan Pendidikan*. Deepublish.
- Ramadhon, R. N. (2021, November 9). *Pengertian Algoritma*. Retrieved from Direktorat Sistem Informasi dan Teknologi: <https://unida.ac.id/teknologi/artikel/pengertian-algoritma.html>
- Sucipto. (2017). Perancangan Active Database System pada Sistem Informasi Pelayanan Harga Pasar. *Jurnal INTENSIF*, 35.
- Sudarmanto, E. S. (2021). *Manajemen Konflik*. Yayasan Kita Menulis .
- Sunarto, K. (2005). *Pengantar Sosiologi*. Universitas Indonesia Publishing.