

# Implementasi Algoritma Rabin-Karp pada Pendeteksian Plagiarisme

Ari Kurniawan Saputra <sup>1\*</sup>, Robby Yuli Endra <sup>1</sup>, Fenty Arani <sup>2</sup>, Tia Tanjung <sup>2</sup>, Agustian Prakarsya <sup>3</sup>

<sup>1</sup> Informatika, Fakultas Ilmu Komputer, Universitas Bandar Lampung, Bandar Lampung

<sup>2</sup> Sistem Informasi, Fakultas Ilmu Komputer, Universitas Bandar Lampung, Bandar Lampung

<sup>3</sup> Sistem Informasi, Fakultas Ilmu Komputer, Institut Teknologi dan Bisnis Lembah Dempo, Sumatera Selatan Indonesia

\* [ari.kurniawan@ubl.ac.id](mailto:ari.kurniawan@ubl.ac.id), [robby.yuliendra@ubl.ac.id](mailto:robby.yuliendra@ubl.ac.id), [fenty.arani@ubl.ac.id](mailto:fenty.arani@ubl.ac.id), [tia.tanjung@ubl.ac.id](mailto:tia.tanjung@ubl.ac.id), [prakarsya@lembahdempo.ac.id](mailto:prakarsya@lembahdempo.ac.id)

**ABSTRACT** – Plagiarism is a crime and a scourge of science. To avoid plagiarism in scientific articles, as in the case of this research, string-matching methods can be used. This study aims to implement the Rabin-Karp Algorithm in detecting plagiarism in scientific writing based on the level of text similarity. The Rabin-Karp algorithm was chosen for this research problem because previous studies revealed that the Rabin-Karp premise is to separate the hash value of the input string from the text substring. Assuming they are the same, the character check is performed one more time, and if not, moves the substring aside. The main part of this computation exhibit is successfully calculating the hash of the substring when applied. This research is quantitative. The stages of this research flow were carried out by testing the implementation of the Rabin-Karp algorithm. Based on the calculation above, the percentage of similarity between Test Sentence 1 and Test Sentence 2 is 77.96%. Referring to previous studies, the Winnowing algorithm was found to be better at detecting text similarities than the Rabin-Karp algorithm. This is shown in the results of the similarity detection test of 30 paper documents as test data with the results of the average percentage value. Rabin-Karp Algorithm 41.41% and Winnowing Algorithm 35.15%. This study shows that the Rabin-Karp Algorithm does not work optimally in detecting text similarity, so further research needs additional methods to calculate a good level of similarity to optimize the performance of the Rabin-Karp Algorithm.

**Keywords:** Algorithm; Plagiarism; Rabin-Karp.

**ABSTRAK** – Plagiarisme adalah tindakan kejahatan dan momok ilmu pengetahuan. Untuk menghindari plagiarisme dalam artikel ilmiah, seperti pada masalah penelitian ini, dapat digunakan metode string matching atau pencocokan string. Penelitian ini bertujuan untuk melakukan implementasi Algoritma Rabin-Karp pada pendeteksian plagiarisme penulisan ilmiah berdasarkan tingkat kemiripan teks. Algoritma Rabin-Karp dipilih dalam permasalahan penelitian ini karena pada penelitian-penelitian sebelumnya mengungkapkan bahwa premis Rabin-Karp adalah memisahkan nilai hash dari input string dengan substring teks. Dengan asumsi keduanya sama, pemeriksaan karakter dilakukan sekali lagi, dan jika tidak, pindahkan substring ke samping. Bagian utama dari pameran perhitungan ini adalah berhasil menghitung hash dari substring saat diterapkan. Penelitian ini bersifat kuantitatif. Tahapan alur penelitian ini dilakukan dengan menguji implementasi algoritma Rabin-Karp. Berdasarkan perhitungan diatas, dapat diketahui persentase kemiripan antar Kalimat Uji 1 dan Kalimat Uji 2 adalah 77,96%. Mengacu pada penelitian sebelumnya, algoritma Winnowing ditemukan lebih baik dalam mendeteksi kemiripan teks dibandingkan algoritma Rabin-Karp. Hal ini ditunjukkan pada hasil pengujian pendeteksian kemiripan 30 dokumen paper sebagai data uji dengan hasil nilai rata-rata persentase. Algoritma Rabin-Karp 41,41% dan Algoritma Winnowing 35,15%. Penelitian ini menunjukkan bahwa Algoritma Rabin-Karp tidak bekerja secara optimal dalam mendeteksi kemiripan teks, sehingga penelitian selanjutnya perlu adanya metode tambahan untuk menghitung tingkat kemiripan teks (Similarity) yang baik untuk mengoptimalkan kinerja Algoritma Rabin-Karp.

**Kata Kunci:** Algoritma; Plagiarisme; Rabin-Karp.

## 1. PENDAHULUAN

Plagiarisme adalah tindakan kejahatan dan momok ilmu pengetahuan [1]. Plagiarisme dalam penulisan ilmiah dianggap ilegal karena melibatkan pembajakan karya orang lain. Salah satu faktor penyebab plagiarisme adalah keinginan komunitas ilmiah untuk publikasi cepat [2].

Untuk menghindari plagiarisme dalam artikel ilmiah, seperti pada masalah penelitian ini, dapat digunakan metode *string matching* atau pencocokan *string* [3][4][5]. *String matching* adalah metode untuk menentukan akurasi atau output dari satu atau lebih pola teks tertentu [6].

Penelitian ini bertujuan untuk melakukan implementasi Algoritma Rabin-Karp pada pendeteksian



plagiarisme penulisan ilmiah berdasarkan tingkat kemiripan teks. Algoritma Rabin-Karp dipilih dalam permasalahan penelitian ini karena pada penelitian yang dilakukan oleh Jody, J., A.T. Wibowo, dan A. Arifianto pada tahun 2015 dengan judul “Analisis Dan Implementasi Algoritma *Winnowing* Dengan *Synonym Recognition* Pada Deteksi Plagiarisme Untuk Dokumen Teks Berbahasa Indonesia” mengungkapkan bahwa premis Rabin-Karp adalah memisahkan nilai *hash* dari *input string* dengan *substring* teks. Dengan asumsi keduanya sama, pemeriksaan karakter dilakukan sekali lagi, dan jika tidak, pindahkan *substring* ke samping. Bagian utama dari pameran perhitungan ini adalah berhasil menghitung *hash* dari *substring* saat diterapkan [7].

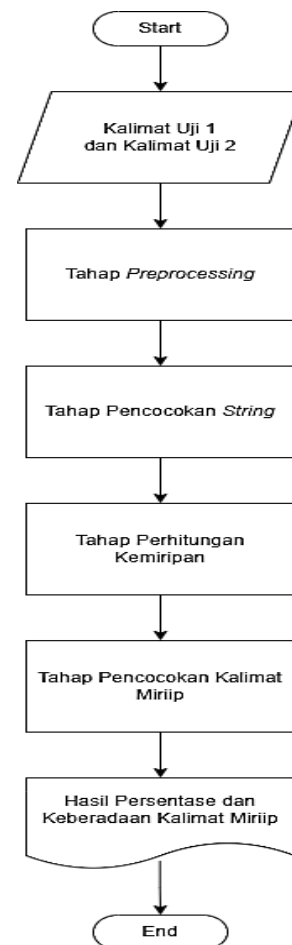
Berdasarkan penelitian terdahulu menyatakan bahwa algoritma Rabin-Karp dapat digunakan untuk mendeteksi plagiarisme dalam *file* dokumen dalam format teks. Hasil analisis yang membandingkan persentase kemiripan dari pengujian sepuluh dokumen teks dengan algoritma Rabin-Karp menghasilkan dokumen asli dan dokumen yang diuji dengan akurasi tertinggi sebesar 47.58 persen. Ketepatan yang paling rendah adalah 19.28%. Analisis terhadap sepuluh dokumen dengan nilai k-gram 1 menghasilkan persentase kemiripan tertinggi dan terendah masing-masing sebesar 57.14% dan 28.57%. Sedangkan nilai kemiripan 30% meliputi plagiarisme *minor*, plagiarisme sedang 30%-70%, dan plagiarisme *mayor* lebih dari 70% [8][9].

Beberapa penelitian terdahulu terkait penambahan atau penggabungan metode ataupun algoritma sebagai upaya untuk meningkatkan hasil akurasi algoritma Rabin-Karp terdapat pada penelitian [5] yang menyatakan bahwa algoritma Rabin-Karp dengan menggunakan metode deteksi sinonim lebih akurat menentukan skor persentase kesamaan dibandingkan dengan algoritma Rabin-Karp tanpa deteksi sinonim. Karena meskipun kata-kata dalam dokumen uji diubah dengan cara ini, kata-kata tersebut memiliki arti yang sama dan dikenali oleh sistem ini. Penelitian yang dilakukan oleh [10] menyatakan bahwa penerapan algoritma Rabin-Karp dan metode *Cosine Similarity* memproses perbandingan plagiarisme dengan jumlah karakter sebanyak 42432 memakan waktu sebanyak 12.49-16.17 Detik, waktu bertambah berdasarkan banyaknya jumlah karakter dalam dokumen. Penelitian yang dilakukan oleh [11] menyatakan bahwa hasil penerapan TF-IDF untuk mengelompokkan frase pencarian dan menghitung derajat kesamaan menggunakan algoritma Rabin-Karp mencatat peningkatan rata-rata sebesar 11.81%. Ini terjadi karena frase akar dikelompokkan dan TF-IDF menghitung frekuensi indeks untuk membuat konversi kata dasar menjadi fungsi *hash* lebih mudah dan lebih akurat. Penelitian selanjutnya dilakukan oleh [12] yang menyatakan bahwa terjadi peningkatan presentasi akurasi algoritma Rabin-Karp pada pengujian kemiripan kata sebesar 20.06% melalui implementasi algoritma Jaro-Winkler pada Rabin-Karp. Namun, tingkat persentase pada pengujian kemiripan kalimat dan abstrak jurnal mengalami penurunan sebesar 15.20%.

Hal ini menunjukkan bahwa algoritma Jaro-Winkler dapat meningkatkan akurasi pendeteksian kesamaan teks ketika diterapkan pada Rabin-Karp.

## 2. METODOLOGI

Penelitian ini bersifat kuantitatif. Penelitian kuantitatif adalah penelitian yang mencari pemecahan masalah dengan cara mengukur variabel-variabel tertentu secara cermat untuk menarik generalisasi yang dapat diterapkan pada setiap waktu, tempat, atau jenis data, khususnya data kuantitatif [13]. Pemeriksaan kuantitatif dipandang sebagai eksplorasi murni yang dapat dipahami dengan angka yang tepat. Penelitian kuantitatif memiliki contoh penalaran yang mendalam, menyiratkan bahwa pemeriksaan dimulai dengan hipotesis keseluruhan, dan kemudian penelitian diselesaikan untuk menguji hipotesis untuk mendapatkan tujuan yang jelas. Dengan demikian, proses penelitian kuantitatif dimulai dengan mengenali isu-isu yang memerlukan pemeriksaan hipotesis untuk mencapai penentuan melalui penetapan pilihan untuk mengakui atau menolak spekulasi penelitian [14]. Tahapan alur penelitian ini dilakukan dengan menguji implementasi algoritma Rabin-Karp yang digambarkan pada Gambar 1.



Gambar 1. Tahapan Analisis Kemiripan Kalimat [15]

### 3. HASIL DAN PEMBAHASAN

Berikut adalah tahapan untuk mengimplementasikan algoritma Rabin-Karp pada kesamaan kalimat sebagai tolak ukur deteksi plagiarisme.

Kalimat Uji (1) – merupakan kalimat atau teks yang digunakan untuk menguji implementasi algoritma Rabin-Karp. Pengumpulan kalimat uji ini merupakan tahapan awal dalam pengujian implementasi algoritma Rabin-Karp. Dalam penelitian ini menggunakan 2 kalimat uji yang terdiri dari Kalimat Uji 1 yang dijadikan sebagai *dataset* dan Kalimat Uji 2 yang dijadikan sebagai *data test*.

Tahap *preprocessing* (2) – adalah langkah yang mendasari perlakuan terhadap arsip atau konten teks.

Tahap ini terdiri dari beberapa proses berikut. Tahap *Preprocessing* ini terdapat pada Tabel 1.

- a. Kalimat Uji  
Proses pengumpulan dan akuisisi Kalimat Uji.
- b. *Case Folding*  
Proses mengubah teks informasi menjadi huruf kecil.
- c. *Tokenizing*  
Proses memisahkan kata-kata dalam kalimat sesuai dengan urutannya.
- d. *Removal of Punctuation Marks*  
Proses menghilangkan salah satu karakter atau tanda aksentuasi, misalnya, tanda titik, tanda seru, tanda tanya, koma, dan sebagainya.
- e. *Removal of Stopwords*  
Proses menghilangkan kata-kata yang tidak penting dalam teks atau kata-kata yang tidak deskriptif, seperti kata "yang", "dan", "di", "dari", dan sebagainya.
- f. *Removal of Space*  
Proses menghilangkan spasi dari setiap kata.

Kalimat Uji 1

Universitas Bandar Lampung, merupakan universitas swasta terbaik Peringkat 1 di lingkungan LLDIKTI Wilayah II.

Kalimat Uji 2

Universitas ABC, merupakan universitas negeri terbaik Peringkat 1 di lingkungan LLDIKTI Wilayah II.

**Tabel 1.** Tahap *Preprocessing*

Kalimat Uji	<i>Case Folding</i>	<i>Tokenizing</i>	<i>Removal of Punctuation Marks</i>	<i>Removal of Stopwords</i>	<i>Removal of Space</i>
Universitas Bandar Lampung, merupakan universitas swasta terbaik Peringkat 1 di lingkungan LLDIKTI Wilayah II.	universitas bandar, lampung merupakan universitas swasta terbaik peringkat 1 di lingkungan lldikti wilayah ii.	universitas bandar lampung merupakan universitas swasta terbaik peringkat 1 di lingkungan lldikti wilayah ii	universitas bandar lampung merupakan universitas swasta terbaik peringkat 1 di lingkungan lldikti wilayah ii	universitas bandar lampung merupakan universitas swasta terbaik peringkat 1 lingkungan lldikti wilayah ii	universitasbanda rlampung merupakanunive rsitas swasta terbaikperingkat 1 lingkungan lldiktiwilayahii
Universitas ABC, merupakan universitas negeri terbaik Peringkat 1 di lingkungan LLDIKTI Wilayah II.	universitas abc, merupakan universitas negeri terbaik peringkat 1 di lingkungan lldikti wilayah ii.	universitas abc , merupakan universitas negeri terbaik peringkat 1 di lingkungan lldikti wilayah ii	universitas abc merupakan universitas negeri terbaik peringkat 1 di lingkungan lldikti wilayah ii	universitas abc merupakan universitas negeri terbaik peringkat 1 lingkungan lldikti wilayah ii	universitasabcm erupakan universitasnegeri terbaik peringkat1lingku nganlldikti wilayahii

Tahapan pencocokan *String K-Gram* (3) – terdapat pada Tabel 2. *K-Gram*. *K-Gram* adalah rangkaian kata dengan panjang *K* (karakter). *K-Gram* digunakan untuk mengambil potongan-potongan karakter huruf sejumlah *K* dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari kalimat.

Hasil *K-Gram* dalam algoritma *Rabin-Karp* adalah serangkaian *K* karakter berurutan yang dihasilkan dari pemisahan teks menjadi *K-Gram*. Setiap *K-Gram* kemudian digunakan untuk menghitung nilai *hash* dan membandingkannya dengan nilai *hash* dari pola yang dicari [5].

**Tabel 2. K-Gram**

Kalimat Uji	<i>K-Gram</i> (3)
Kalimat Uji 1	{uni},{niv},{ive},{ver},{ers},{rsi},{sit},{ita},{tas},{asb},{sba},{ban},{and},{nda},{dar},{arl},{rla},{lam},{amp},{mpu},{pun},{ung},{ngm},{gme},{mer},{eru},{rup},{upa},{pak},{aka},{kan},{anu},{nun},{uni},{niv},{ive},{ver},{ers},{rsi},{sit},{ita},{tas},{ass},{ssw},{swa},{was},{ast},{sta},{tat},{ate},{ter},{erb},{rba},{bai},{aik},{ikp},{kpe},{per},{eri},{rin},{ing},{ngk},{gka},{kat},{at1},{t1d},{1di},{dil},{ili},{lin},{ing},{ngk},{gku},{kun},{ung},{nga},{gan},{anl},{nll},{lld},{ldi},{dik},{ikt},{kti},{tiw},{iwi},{wil},{ila},{lay},{aya},{yah},{ahi},{hii}
Kalimat Uji 2	{uni},{niv},{ive},{ver},{ers},{rsi},{sit},{ita},{tas},{asa},{sab},{abc},{bcm},{cme},{mer},{eru},{rup},{upa},{pak},{aka},{kan},{anu},{nun},{uni},{niv},{ive},{ver},{ers},{rsi},{sit},{ita},{tas},{asn},{sne},{neg},{ege},{ger},{eri},{rit},{ite},{ter},{erb},{rba},{bai},{aik},{ikp},{kpe},{per},{eri},{rin},{ing},{ngk},{gka},{kat},{at1},{t1d},{1di},{dil},{ili},{lin},{ing},{ngk},{gku},{kun},{ung},{nga},{gan},{anl},{nll},{lld},{ldi},{dik},{ikt},{kti},{tiw},{iwi},{wil},{ila},{lay},{aya},{yah},{ahi},{hii}

**Tabel 3. Hashing**

Kalimat Uji	<i>Hash</i>
Kalimat Uji 1	{4235},{4036},{4007},{4188},{3895},{4210},{4157},{3979},{4100},{3771},{4049},{3640},{3725},{3944},{3698},{3781},{4124},{3888},{3739},{4079},{4190},{4231},{3998},{3867},{3963},{3899},{4244},{4239},{3984},{3689},{3865},{3759},{4140},{4235},{4036},{4007},{4188},{3895},{4210},{4157},{3979},{4100},{3805},{4263},{4259},{4175},{3807},{4229},{4102},{3787},{4138},{3861},{4024},{3630},{3689},{3919},{3997},{4038},{3875},{4120},{3931},{3994},{3839},{3877},{3683},{3590},{2435},{3766},{3915},{3970},{3931},{3994},{3879},{4065},{4231},{3974},{3765},{3741},{4046},{3980},{3910},{3764},{3927},{4045},{4188},{4025},{4241},{3899},{3912},{3829},{4203},{3675},{3860}
Kalimat Uji 2	{4235},{4036},{4007},{4188},{3895},{4210},{4157},{3979},{4100},{3769},{4041},{3603},{3658},{3767},{3963},{3899},{4244},{4239},{3984},{3689},{3865},{3759},{4140},{4235},{4036},{4007},{4188},{3895},{4210},{4157},{3979},{4100},{3795},{4177},{3966},{3757},{3813},{3875},{4132},{3987},{4138},{3861},{4024},{3630},{3689},{3919},{3997},{4038},{3875},{4120},{3931},{3994},{3839},{3877},{3683},{3590},{2435},{3766},{3915},{3970},{3931},{3994},{3879},{4065},{4231},{3974},{3765},{3741},{4046},{3980},{3910},{3764},{3927},{4045},{4188},{4025},{4241},{3899},{3912},{3829},{4203},{3675},{3860}

Tahapan pencocokan *string hashing* (4) – terdapat pada Tabel 3. *Hashing* merupakan tahapan untuk mengubah *string* menjadi nilai unik dengan panjang tertentu (*fixed length*) yang bertindak seperti penanda *string*. Fungsi *hash* adalah cara untuk membuat *Fingerprint* dari berbagai data masukan. *Hash* akan mengganti atau mengubah data untuk membuat *Fingerprint* [5].

Tahap Perhitungan Kemiripan (5) – *Fingerprint* berfungsi sebagai representasi singkat dari pola yang digunakan untuk membandingkan dengan nilai *hash K-Gram* dalam teks. Pada Tabel 4. *Fingerprint* menunjukkan nilai hash yang mirip antara Kalimat Uji 1 sebagai *dataset* dan Kalimat Uji 2 sebagai data *test* terdapat 69 kemiripan data *hash*. Langkah ini melakukan perhitungan terhadap hasil kemiripan kata pada data *test*

dengan *dataset*. Nilai kemiripan akan dihitung berdasarkan jumlah *hash* yang telah diproses oleh algoritma *Rabin-Karp* dan jumlah *hash* pada data *test*. Hasil perhitungan kemiripan ini berasal dari hasil data *test* yang menghasilkan nilai persentase kemiripan kalimat.

Pencocokan kalimat mirip (6) – merupakan langkah terakhir. Pada langkah ini pengumpulan data dilakukan dari *dataset* dan data *test*. Dengan membagi bagian menjadi kalimat. Kemudian fragmen kalimat dipotong menjadi kata-kata. Selanjutnya kata-kata dari data *test* ini akan dicocokkan dengan kata-kata yang ada dalam dokumen kumpulan data tersebut. Selanjutnya akan menampilkan kata-kata yang mirip antara kata-kata pada data *test* dengan *dataset*.



Tabel 4. *Fingerprint*

Kalimat Uji	<i>Fingerprint</i>
Kalimat Uji 1	{4235}, {4036}, {4007}, {4188}, {3895}, {4210}, {4157}, {3979}, {4100}, {3771}, {4049}, {3640}, {3725}, {3944}, {3698}, {3781}, {4124}, {3888}, {3739}, {4079}, {4190}, {4231}, {3998}, {3867}, {3963}, {3899}, {4244}, {4239}, {3984}, {3689}, {3865}, {3759}, {4140}, {4235}, {4036}, {4007}, {4188}, {3895}, {4210}, {4157}, {3979}, {4100}, {3805}, {4263}, {4259}, {4175}, {3807}, {4229}, {4102}, {3787}, {4138}, {3861}, {4024}, {3630}, {3689}, {3919}, {3997}, {4038}, {3875}, {4120}, {3931}, {3994}, {3839}, {3877}, {3683}, {3590}, {2435}, {3766}, {3915}, {3970}, {3931}, {3994}, {3879}, {4065}, {4231}, {3974}, {3765}, {3741}, {4046}, {3980}, {3910}, {3764}, {3927}, {4045}, {4188}, {4025}, {4241}, {3899}, {3912}, {3829}, {4203}, {3675}, {3860}
	<b>(Keterangan : Jumlah 94)</b>
Kalimat Uji 2	{4235}, {4036}, {4007}, {4188}, {3895}, {4210}, {4157}, {3979}, {4100}, {3769}, {4041}, {3603}, {3658}, {3767}, {3963}, {3899}, {4244}, {4239}, {3984}, {3689}, {3865}, {3759}, {4140}, {4235}, {4036}, {4007}, {4188}, {3895}, {4210}, {4157}, {3979}, {4100}, {3795}, {4177}, {3966}, {3757}, {3813}, {3875}, {4132}, {3987}, {4138}, {3861}, {4024}, {3630}, {3689}, {3919}, {3997}, {4038}, {3875}, {4120}, {3931}, {3994}, {3839}, {3877}, {3683}, {3590}, {2435}, {3766}, {3915}, {3970}, {3931}, {3994}, {3879}, {4065}, {4231}, {3974}, {3765}, {3741}, {4046}, {3980}, {3910}, {3764}, {3927}, {4045}, {4188}, {4025}, {4241}, {3899}, {3912}, {3829}, {4203}, {3675}, {3860}
	<b>(Keterangan : Jumlah 83)</b>
Jumlah <i>Fingerprint</i> yang sama dari Kalimat Uji 1 dan Kalimat Uji 2	{4235}, {4036}, {4007}, {4188}, {3895}, {4210}, {4157}, {3979}, {4100}, {3899}, {4244}, {4239}, {3984}, {3689}, {3865}, {3759}, {4140}, {4235}, {4036}, {4007}, {4188}, {3895}, {4210}, {4157}, {3979}, {4100}, {4138}, {3861}, {4024}, {3630}, {3689}, {3919}, {3997}, {4038}, {3875}, {4120}, {3931}, {3994}, {3839}, {3877}, {3683}, {3590}, {2435}, {3766}, {3915}, {3970}, {3931}, {3994}, {3879}, {4065}, {4231}, {3974}, {3765}, {3741}, {4046}, {3980}, {3910}, {3764}, {3927}, {4045}, {4188}, {4025}, {4241}, {3899}, {3912}, {3829}, {4203}, {3675}, {3860}
	<b>(Keterangan : Jumlah 69)</b>

Langkah-langkah menghitung kemiripan kalimat menggunakan metode *Dice Coefficient Similarity*. Tahapan perhitungan kemiripan antara Kalimat Uji 1 dan Kalimat Uji 2 menggunakan metode *Dice Coefficient Similarity* dengan Rumus 1:

$$S = ((2 \times C) / (A + B)) \times 100 \quad (\text{Rumus 1})$$

Berdasarkan Tabel 4, maka:

A = Jumlah *Fingerprint* Kalimat Uji 1 = 94;

B = Jumlah *Fingerprint* Kalimat Uji 2 = 83;

C = Jumlah *Fingerprint* yang sama dari Kalimat Uji 1 dan Kalimat Uji 2 = 69;

sehingga,

$$S = ((2 \times 69) / (94 + 83)) \times 100$$

$$S = 77,96\%$$

Berdasarkan perhitungan Rumus 1 dapat diketahui persentase kemiripan antar Kalimat Uji 1 dan Kalimat Uji 2 adalah 77.96%. Mengacu pada penelitian sebelumnya, algoritma *Winnowing* ditemukan lebih baik dalam mendeteksi kemiripan teks dibandingkan algoritma *Rabin-Karp*. Hal ini ditunjukkan pada hasil pengujian pendeteksian kemiripan 30 dokumen *paper* sebagai data uji dengan hasil nilai rata-rata persentase Algoritma *Rabin-Karp* 41.41% dan Algoritma *Winnowing* 35.15% [16].

#### 4. KESIMPULAN

Penelitian ini menunjukkan bahwa Algoritma *Rabin-Karp* tidak bekerja secara optimal dalam mendeteksi kemiripan teks., sehingga penelitian selanjutnya perlu adanya metode tambahan untuk menghitung tingkat kemiripan teks (*Similarity*) yang baik untuk mengoptimalkan kinerja Algoritma *Rabin-Karp*.

#### DAFTAR PUSTAKA

- [1] M. A. Shadiqi, "Memahami dan Mencegah Perilaku Plagiarisme dalam Menulis Karya Ilmiah," *Bul. Psikol.*, vol. 27, no. 1, p. 30, Jun. 2019, doi: 10.22146/buletinsikologi.43058.
- [2] A. K. Saputra, E. Erlangga, and T. Tanjung, "Analisis Kinerja Algoritma *Winnowing* pada Pendeteksian Plagiarisme," *Explor. J. Sist. Inf. dan Telemat.*, vol. 13, no. 2, p. 125, Dec. 2022, doi: 10.36448/jsit.v13i2.2845.
- [3] A. Filcha and M. Hayaty, "Implementasi Algoritma *Rabin-Karp* untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa," *JUITA J. Inform.*, vol. 7, no. 1, p. 25, May 2019, doi: 10.30595/juita.v7i1.4063.
- [4] A. Sunyoto and T. Informatika, "Implementasi Algoritma *Rabin Karp* untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep *Similarity*," pp. 23–28, 2013.





- [5] N. Prima Putra and S. Sularno, "Penerapan Algoritma Rabin-Karp Dengan Pendekatan Synonym Recognition Sebagai Antisipasi Plagiarisme Pada Penulisan Skripsi," *J. Teknol. Dan Sist. Inf. Bisnis*, Vol. 1, No. 2, Pp. 48–58, Jul. 2019, Doi: 10.47233/Jteksis.V1i2.52.
- [6] M. O. Pohan And R. D. Sianturi, "Penerapan Algoritma Rabin-Karp Pada Pencarian Sinonim Kata," *J. Informatics Manag. Inf. Technol.*, Vol. 2, No. 1, Pp. 12–17, 2022.
- [7] S. L. B. Ginting, Y. R. Ginting, S. Sutono, And W. A. Sirait, "Aplikasi Deteksi Kemiripan Kata Menggunakan Algoritma Rabin-Karp," *J. Teknol. Dan Inf.*, Vol. 12, No. 2, Pp. 162–175, 2022, Doi: 10.34010/Jati.V12i2.6947.
- [8] A. Mubarak, "Implementasi Algoritma Rabin-Karp Untuk Pendeteksianplagiarisme Pada File Dokumen Berupa Text Berbasis Web," *J. Inf. Syst. Res.*, Vol. 3, No. 3, Pp. 150–154, 2022, Doi: 10.47065/Josh.V3i3.1404.
- [9] M. A. Hardison, "Deteksi Plagiarisme Pada File Dokumen Berdasarkan Tingkat Kesamaan Dengan Menggunakan Metode Algoritma Rabin-Karp Berbasis Web," *Oktal J. Ilmu Komput. Dan Sci.*, Vol. 2, No. 3, Pp. 760–766, 2023, [Online]. Available: <https://journal.Mediapublikasi.Id/Index.Php/Oktal/Article/View/952/1050>
- [10] E. Siswanto And Y. Ceng Giap, "Implementasi Algoritma Rabin-Karp dan Cosine Similarity untuk Pendeteksi Plagiarisme Pada Dokumen," *J. Algor.*, Vol. 1, No. 2, Pp. 16–22, 2020, [Online]. Available: <https://jurnal.Buddhidharma.Ac.Id/Index.Php/Algor/Indeks>
- [11] S. Bahri And R. Wajhillah, "Optimalisasi Algoritma Rabin Karp Menggunakan Tf-Idf Dalam Pencocokan Text Pada Penilaian Ujian Essay Otomatis," *Infotekjar J. Nas. Inform. Dan Teknol. Jar.*, Vol. 4, No. 2, Pp. 292–295, 2020.
- [12] M. A. Yulianto And N. Nurhasanah, "The Hybrid Of Jaro-Winkler And Rabin-Karp Algorithm In Detecting Indonesian Text Similarity," *J. Online Inform.*, Vol. 6, No. 1, P. 88, Jun. 2021, Doi: 10.15575/Join.V6i1.640.
- [13] M. Arsyam And M. Yusuf Tahir, "Ragam Jenis Penelitian Dan Perspektif," *Al-Ubudiyah J. Pendidik. Dan Stud. Islam*, Vol. 2, No. 1, Pp. 37–47, Jun. 2021, Doi: 10.55623/Au.V2i1.17.
- [14] A. A. G. Muhammad Darwin, Marianne Reynelda Mamondol, Salman Alparis Sormin, Yuliana Nurhayati, Hardi Tambunan, Diana Sylvia, I Made Dwi Mertha Adnyana, Budi Prasetyo, Pasionista Vianitati, "Metode Penelitian Pendekatan Kuantitatif," T. S. Tambunan, Ed. Kota Bandung: Media Sains Indonesia, 2021, P. 178. [Online]. Available: <https://store.Medsan.Co.Id/Detail/978-623-629-056-9-Metode-Penelitian-Pendekatan-Kuantitatif>
- [15] A. Santoso And A. Solichin, "Analisis Kemiripan Dokumen Tesis Menggunakan Algoritma Rabin-Karp Dan Dice Coefficient Similarity," *Techno.Com*, Vol. 22, No. 1, Pp. 12–27, Feb. 2023, Doi: 10.33633/Tc.V22i1.7110.
- [16] A. K. Saputra, K. Muludi, And T. Thamrin, "Comparative Analysis Between Rabin Karp Algorithm, Winnowing, And Turnitin Applications For Detecting Plagiarized Words," *Proceeding 6th Icitb 2020* –, No. December, Pp. 40–49, 2020, [Online]. Available: <https://jurnal.Darmajaya.Ac.Id/Index.Php/Icitb/Article/View/2505>.

