EXPLORE
Jurnal Sistem Informasi dan Telematika
(Telekomunikasi, Multimedia dan Informatika)

Analisis Kinerja Algoritma Winnowing pada Pendeteksian Plagiarisme

Ari Kurniawan Saputra [1], Erlangga [2]*, Tia Tanjung [2]
[1] Informatika, Fakultas Ilmu Komputer Universitas Bandar Lampung
[2] Sistem Informasi, Fakultas Ilmu Komputer Universitas Bandar Lampung
Bandar Lampung, Indonesia
ari.kurniawan@ubl.ac.id, *)erlangga@ubl.ac.id, tia.tanjung@ubl.ac.id

ABSTRACT – The problem of plagiarism at this time can be solved. This research was carried out by analyzing the performance of the Winnowing Algorithm on plagiarism detection. The choice of the Winnowing Algorithm is because by calculating the hash value of k-grams, the rolling hash function is used to find the hash value, after which a window is formed from the hash value. In each window, the smallest hash value is selected. When more than one hash has the lowest value, the rightmost hash value is selected. Then all selected hash values are stored to be used as document fingerprints. This fingerprint serves as a basis for comparing the similarity of embedded text. The results of the Winnowing Algorithm performance analysis research show that this algorithm is quite good at detecting text similarity or plagiarism with the result of the smallest percentage value of 10 tests shown in the 10th test with n-gram values n = 10, window w = 3, time process sec = 0.0094 with a result of 47% text similarity. This result is better than the results of the similarity of the Rabin Karp Algorithm from the results of previous studies.

Keywords: Document Fingerprints, Plagiarism, Similarity, Turnitin, Winnowing Algorithm.

ABSTRAK – Masalah plagiarisme di saat ini dapat diselesaikan. Penelitian ini dilakukan analisis kinerja Algoritma Winnowing pada pendeteksian plagiarisme. Pemilihan Algoritma Winnowing karena dengan menghitung nilai hash dari k-gram, fungsi hash bergulir dipakai mencari nilai hash, setelah itu sebuah window dibentuk dari nilai hash. Di tiap window, nilai hash terkecil dipilih. Ketika lebih dari satu hash nilai terendah, nilai hash paling kanan dipilih. Kemudian semua nilai hash yang dipilih disimpan untuk digunakan sebagai fingerprint dokumen. Fingerprint ini berfungsi sebagai dasar untuk membandingkan kesamaan teks yang disematkan. Hasil penelitian analisis kinerja Algoritma Winnowing ini menunjukan bahwa algoritma ini cukup baik dalam pendeteksian kemiripan teks atau plagiarisme dengan hasil nilai persentase terkecil dari 10 kali pengujian yang ditunjukan pada pengujian ke-10 dengan nilai n-gram n=10, window w=3, waktu proses sec=0.0094 dengan hasil kemiripan teks 47 %. Hasil ini lebih baik dari hasil kemiripan Algoritma Rabin Karp dari hasil penelitian sebelumnya.

Kata Kunci: Algoritma Winnowing, Fingerprint Dokumen, Plagiasi, Similaritas, Turnitin.

1. Pendahuluan

Masalah plagiarisme pada saat ini dapat dilakukan secara praktis dan lebih mudah. Praktik plagiat ini sudah sering terjadi khususnya di kalangan akademisi. Salah satu penyebab plagiarisme adalah tekanan untuk mempublikasikan dengan cepat di komunitas ilmiah. Konsekuensinya, mereka terdorong untuk menjiplak karena mereka kurang memiliki motivasi untuk menciptakan karya mereka sendiri, keterampilan untuk mengevaluasi sumber secara kritis, pengetahuan tentang kapan dan bagaimana mengutip, dan kepedulian pembimbing atau instruktur yang minim [1].

Jumlah teks plagiasi terdapat 3 kategori: berat, sedang, ringan Plagiarisme ringan memiliki rasio atau persentase ter plagiasi kurang dari 30%, sedang 30% -70%, dan berat memiliki rasio atau persentase teks yang dicuri lebih besar dari 70% [2].

Tindakan plagiarisme dalam karya seperti penulisan ilmiah ini sudah dikatakan melanggar hukum karena sudah mencuri hak cipta karya orang lain. Untuk mencegah tindakan plagiarisme dalam hal penulisan ilmiah seperti ini dapat dilakukan dengan cara mendeteksi setiap kata atau kalimat di dalamnya memakai algoritma *matching string* atau kata pada tulisan.

Penelitian ini dilakukan analisis kinerja Algoritma Winnowing pada pendeteksian plagiarisme. Winnowing dipilih karena menggunakan *rolling hash function* untuk menghitung nilai *hash* dari setiap k-gram dan kemudian menggunakan nilai tersebut untuk membentuk *window*.

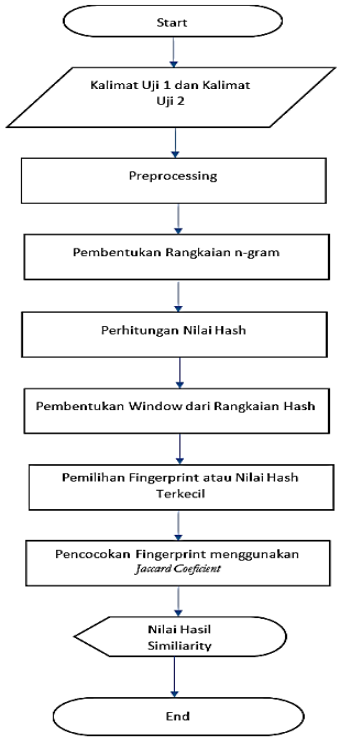
EXPLORE: ISSN: 2087-2062, Online ISSN: 2686-181X / DOI: http://dx.doi.org/10.36448/jsit.v13i2.2845

Identifikasi nilai *hash* serendah mungkin untuk setiap interval waktu. Jika lebih dari satu hash memiliki nilai terkecil, yang paling kanan dipilih. Setelah dokumen di-*hash*, sidik jarinya dapat diambil dan digunakan di masa mendatang. Saat menentukan seberapa mirip dua teks, sidik jari ini akan berfungsi sebagai standar. [3].

Berdasarkan penelitian terdahulu menyatakan bahwa Algoritma Winnowing mendapatkan hasil terbaik pada uji coba dengan nilai n-gram (n) = 3 dan window (w) = 3, sehingga dapat disimpulkan Algoritma Winnowing memiliki persentase akurasi yang baik [4][5]. Pada penelitian perbandingan antara algoritma Rabin-Karp dengan Algoritma Winnowing lebih unggul dibandingkan dengan Algoritma Rabin-Karp dari kedua variabel yang menjadi landasan perbandingan karena bukan hanya nilai *hash* saja yang dibandingkan, namun nilai *hash* yang ada akan dikonversikan terlebih dulu kedalam *window* baru sehingga dapat diketahui nilai *hash* yang sama [6][7][8][9].

2. Metodologi

Penelitian ini bersifat kualitatif. Tujuan dari metode penelitian kualitatif adalah untuk mendapatkan pemahaman menyeluruh tentang suatu topik dari pada hanya mengungkap yang dangkal. Karena metode kualitatif menganggap sifat suatu pertanyaan akan berbeda dengan sifat pertanyaan lainnya, maka metode penelitian ini lebih menyukai penggunaan teknik analisis mendalam, seperti mempelajari pertanyaan kasus per kasus [10]. Tahap penelitian alur sistem ini dilakukan dengan menguji antara Algoritma Winnowing dan Aplikasi Turnitin yang digambarkan pada Gambar 1.



Gambar 1. *Flowchart* Alur Pengujian

A. Pengumpulan Data

Pengumpulan data dilakukan dengan cara observasi dan studi pustaka. Observasi dengan membuat *dataset* atau data pelatihan yang digunakan sebagai data perbandingan pada proses pendeteksian plagiarisme. Saat melakukan tinjauan pustaka, praktik yang umum dilakukan adalah penggunaan kutipan langsung dan catatan rinci yang diambil dari bahan pustaka pendukung dan relevan secara ekstensif.

B. Metode

Dalam penelitian ini dilakukan analisis kinerja dari Algoritma Winnowing dan menggunakan Aplikasi Turnitin sebagai tolak ukur dalam pendeteksian plagiarisme. Untuk melakukan tahap pengukuran kinerja antara Algoritma Winnowing dan Aplikasi Turnitin sebagai tolak ukur pendeteksian plagiarisme tersebut, penelitian ini menggunakan metode perbandingan nilai rata-rata uji hipotesis t-*test*. Masing-masing pendekatan ini diuraikan sebagai berikut:

1. Algoritma Winnowing

Winnowing [3], sebuah algoritma untuk sidik jari dokumen. Menggunakan sidik jari dokumen, Anda dapat memeriksa apakah dua dokumen atau dua salinan dari teks yang sama akurat. Teknik sidik jari yang dijelaskan di sini menggunakan algoritma *hashing*. Fungsi *hash* menghitung pengidentifikasian unik untuk *string* teks apa pun.

Nilai *hash* ditentukan untuk setiap k-gram oleh algoritma Winnowing, yang dalam hal ini menggunakan fungsi *hash* bergulir. Nilai *hash* yang dihasilkan digunakan untuk membuat jendela. Nilai *hash* terkecil dipilih di setiap interval waktu. Nilai *hash* paling kanan dipilih jika ada beberapa nilai *hash* dengan nilai terendah yang sama. Sebagai sidik jari dokumen, hash yang dipilih kemudian disimpan.

File teks berfungsi sebagai input untuk sidik jari dokumen. Produk akhirnya adalah sidik jari, yang merupakan kumpulan *hash*. Sidik jari ini kemudian digunakan untuk menentukan seberapa mirip teks yang disematkan satu sama lain. Parameter algoritma untuk mendeteksi plagiarisme harus:

a. *Whitespace Insensitivity,* yaitu spasi, huruf kapital, tanda baca, tidak boleh mempengaruhi perataan teks *file.*
b. *Noise Suppression*, yaitu pengurangan noise, atau mengabaikan file teks yang berisi kata yang terlampau kecil atau tidak relevan dibandingkan dengan kata lebih sering dipakai.
c. *Position Independence*, yaitu mencocokkan file teks tidak boleh bergantung pada urutan kata di dalam file tersebut, sebuah fitur yang dikenal sebagai kemandirian posisi, yang memastikan bahwa dokumen dengan kata dalam urutan tampilan yang berbeda akan tetap cocok.

Algoritma Winnowing memenuhi kebutuhan ini dengan mem-filter karakter non-alphanumeric seperti

EXPLORE: ISSN: 2087-2062, Online ISSN: 2686-181X / DOI: http://dx.doi.org/10.36448/jsit.v13i2.2845

tanda baca dan spasi sebelum melanjutkan pemrosesan. Algoritma Winnowing diimplementasikan dengan langkah-langkah berikut:

1. Penghapusan Karakter yang tidak dipakai
   Penghilangan tanda baca, spasi dan simbol “, =, #, %, &, (,), -, _, $, @, !, /,”

   Contoh 1:

Aplikasi Deteksi Source Code C++

   dimodifikasi menjadi

aplikasideteksisourcecodec

2. Pembentukan Rangkaian n-*gram*.
   Dari teks di atas yang sudah dibersihkan dengan ukuran k, k = 7, kita dapat membentuk deret n-gram dengan menghilangkan karakter yang tidak diperlukan (gram terbaik penelitian sebelumnya) Gunakan “*licasid ikaside kasidet acidite Ideteks Detect Eteksis Eksiso Eksisou Ksisour Csisour Isourcec Sidetek ourcod urcod rcecode cecodec*”.

3. Perhitungan Fungsi *Hash* untuk tiap n-*gram*
   Tugas dari *Hash* untuk menentukan nilai *hash* dari setiap gram dalam n-gram. Untuk Algoritma Winnowing, *rolling hash* adalah fungsi yang dipakai untuk menghitung nilai *hash* gram yang berurutan. Dengan menggunakan *rolling hash*, sebuah *string* dapat diubah menjadi nilai unik dengan panjang tetap yang dapat digunakan sebagai *token string*. Fungsi semacam ini dikenal sebagai *fungsi hash*, dan nilai yang dikembalikan dikenal sebagai nilai *hash*; Rumus (1) ascii karakter (c), basis (b), dan jumlah karakter (k).

$$H(ck) = c1 * b(k-1) + c2 * b\,(k-2) + ck * b\,(k-k) \qquad (1)$$

   Hasil *rolling hash* kalimat Contoh 1:

26194 27766 27060 26674 26700 26210 27802 26394 26118 26866 28098 26734 27840 28486 27734 28786 28354 28492 27234 25482

4. Pembentukan *Window* dari Nilai *Hash*
   Nilai hash dari *window* ukuran *w* = 9 yaitu:

W-1: {26194 27766 27060 26674 26700 26210 27802 26394 26118} W-2: {27766 27060 26674 26700 26210 27802 26394 26118 26866} W-3: {27060 26674 26700 26210 27802 26394 26118 26866 28098} W-4: {26674 26700 26210 27802 26394 26118 26866 28098 26734} W-5: {26700 26210 27802 26394 26118 26866 28098 26734 27840} W-6: {26210 27802 26394 26118 26866 28098 26734 27840 28486} W-7: {27802 26394 26118 26866 28098 26734 27840 28486 27734} W-8: {26394 26118 26866 28098 26734 27840 28486 27734 28786} W-9: {26118 26866 28098 26734 27840 28486 27734 28786 28354} W-10: {26866 28098 26734 27840 28486 27734 28786 28354 28492} W-11: {28098 26734 27840 28486 27734 28786 28354 28492 27234} W-12: {26734 27840 28486 27734 28786 28354 28492 27234 25482}

5. Pemilihan *Fingerprint* dari Setiap *Window*
   Langkah terakhir adalah melihat nilai terkecil di tiap *window* sebagai *fingerprint*, nilai *fingerprint*-nya:

26118 26118 26118 26118 26118 26118 26118 26118 26118 26734 26734 25482

6. Persamaan *Jaccard Coefficient*
   Adalah persamaan Koefisien *Jaccard*, nilai sidik jari yang dibentuk oleh Algoritma Winnowing digunakan sebagai ukuran persentase kesamaan teks. Seberapa mirip dua set kata hash dapat ditentukan dengan menggunakan Persamaan Koefisien *Jaccard*. Rumus (2) digunakan untuk menghitung koefisien *Jaccard*.

$$Similiarity = \frac{\text{Jumlah\_fingerprint\_sama}}{Total\_seluruh\_fingerprint} X\ 100 \qquad (2)$$

3. Hasil Dan Pembahasan

A. Implementasi

Berikut merupakan tahapan implementasi Algoritma Winnowing yang dijabarkan pada pengujian berikut.

Kalimat uji 1

ANALISA PERBANDINGAN ALGORITMA RABIN KARP, ALGORITMA WINNOWING DAN APLIKASI TURNITIN PADA PENDETEKSIAN PLAGIARISME

Kalimat uji 2

ANALISA PERBANDINGAN ALGORITMA RABIN KARP, ALGORITMA TF-IDF PADA PENDETEKSIAN PLAGIARISME

a. Pembuangan Karakter yang Tidak Relevan.
   Tahapan ini akan melakukan penghapusan tanda baca, spasi, symbol “, =, #, %, &, (,), -, _, $, @, !, /,” dan merubah semua teks menjadi *lowercase* atau huruf kecil seperti hasil berikut ini.

Kalimat uji 1

analisaperbandinganalgoritmarabinkarpalgoritmawinnowingdanaplikasiturnitinpadapendeteksianplagiarisme

Kalimat uji 2

analisaperbandinganalgoritmarabinkarpalgoritmatfidfpada pendeteksianplagiarisme

b. Pembentukan Rangkaian *n-gram*
   Tahapan ini akan melakukan pembentukan *n-gram* sebagai kalimat uji, jumlah data pengelompokan *n-gram* dimulai dari 2 sampai 10, berikut pengujian menggunakan nilai *n-gram* = 2.

EXPLORE: ISSN: 2087-2062, Online ISSN: 2686-181X / DOI: http://dx.doi.org/10.36448/jsit.v13i2.2845

Kalimat uji 1

an na al li is sa ap pe er rb ba an nd di in ng ga an na al lg go or ri it tm ma ar ra ab bi in nk ka ar rp pa al lg go or ri it tm ma aw wi in nn no ow wi in ng gd da an na ap pl li ik ka as si it tu ur rn ni it ti in np pa ad da ap pe en nd de et te ek ks si ia an np pl la ag gi ia ar ri is sm me

Kalimat uji 2

an na al li is sa ap pe er rb ba an nd di in ng ga an na al lg go or ri it tm ma ar ra ab bi in nk ka ar rp pa al lg go or ri it tm ma at tf fi id df fp pa ad da ap pe en nd de et te ek ks si ia an np pl la ag gi ia ar ri is sm me

c. Perhitungan Fungsi *Hash* untuk tiap *n-gram*

Tahapan ini merumuskan perhitungan fungsi *hash,* masing-masing *n-gram* dibuat *rolling hash,* berikut pengujian perhitungan *hash n-gram* pada kalimat uji 1 dan 2 pada kata "analisa" dengan nilai basis (b) = 3 dan nilai *n-gram* = 7.

$$H(analisis) = Ascii(a) * 3^6 + Ascii(n) * 3^5 + Ascii(a) * 3^4 + Asii(l) * 3^3 + Ascii(i) * 3^2 + Ascii(s) * 3^1 + Ascii(a) * 3^0$$

$$= 97 * 729 + 110 * 243 + 97 * 81 + 108 * 27 + 105 * 9 + 115 * 3 + 97 * 1$$

Dari hasil perhitungan tersebut maka didapatkan nilai *hash* berikut:

70713 26730 7857 2916 945 345 97

d. Pembentukan *Window* dari Nilai *Hash*

Tahapan ini melakukan pengelompokan (*Windowing)* untuk setiap hasil *hash*, langkah tersebut sama seperti dengan *n-gram*. Hasil rangkaian *window* pada kalimat uji 1 dan kalimat uji 2 ditunjukan pada Tabel 1.

Tabel 1. Tabel Rangkaian *Window*

No	Kalimat Uji 1 dan 2 pada teks "analisa"
1	{ 70713 26730 7857 }
2	{ 7857 26730 2916 }
3	{ 2916 7857 945 }
4	{ 945 2916 345 }
5	{ 345 945 97 }

e. Pemilihan *Fingerprint* dari Setiap *Window*

Tahapan ini melakukan pengambilan nilai *hash* terkecil dari rangkaian *Window* yang disebut dengan *Fingerprint*. Berikut proses dari *Fingerprint:*

[7857, 1] [2916, 2] [945, 3] [345, 4] [97, 5]

f. Persamaan *Jaccard Coefficient*

Tahapan ini melakukan perhitungan persentase kemiripan yang dilakukan menggunakan persamaan *(2).*

$$Similiarity = \frac{5}{5} X\ 100 = 100\%$$

B. Pengujian

Tabel 2 merupakan hasil uji coba pendeteksian kemiripan teks pada Algoritma Winnowing untuk mendapatkan persentase kemiripan teks pada Kalimat Uji 1 dan Kalimat Uji 2.

Tabel 2. Tabel Uji Persentase Kemiripan Teks Algoritma Winnowing

Jumlah Uji Coba	*n-gram (n)*	*Window (w)*	*Waktu Proses (sec)*	Kemiripan *(%)*
1	2	3	0.0080	64.5
	2	5	0.0934	63.1
	2	7	0.0092	56.2
	2	9	0.0104	57.1
2	3	3	0.0122	60.0
	3	5	0.0096	60.0
	3	7	0.0088	59.0
	3	9	0.0135	55.5
3	4	3	0.0081	58.6
	4	5	0.0085	59.3
	4	7	0.0083	58.3
	4	9	0.0101	55.5
4	5	3	0.0084	56.2
	5	5	0.0088	54.5
	5	7	0.0093	58.3
	5	9	0.0093	55.5
5	6	3	0.0090	56.2
	6	5	0.0084	54.5
	6	7	0.0100	58.3
	6	9	0.0091	55.5
6	7	3	0.0110	55.3
	7	5	0.0081	54.5
	7	7	0.0083	58.3
	7	9	0.0082	55.5

EXPLORE: ISSN: 2087-2062, Online ISSN: 2686-181X / DOI: http://dx.doi.org/10.36448/jsit.v13i2.2845

Jumlah Uji Coba	*n-gram (n)*	*Window (w)*	*Waktu Proses (sec)*	Kemiripan *(%)*
7	8	3	0.0082	53.0
	8	5	0.0085	51.4
	8	7	0.0088	53.8
	8	9	0.0092	55.5
8	9	3	0.0757	53.0
	9	5	0.0082	51.4
	9	7	0.0086	53.8
	9	9	0.0094	55.5
9	10	3	0.0094	47.0
	10	5	0.0090	47.2
	10	7	0.0089	48.1
	10	9	0.0089	47.3

Dari tabel pengujian pada Tabel 2, uji coba pendeteksian kemiripan teks Algoritma Winnowing dilakukan sebanyak 9 kali dengan nilai *n-gram* awal *n*=2 dan nilai *n-gram* akhir *n*=10 serta waktu proses yang berbeda-beda setiap *n-gram*, dengan penggunaan nilai Window 3,5,7,9. Dari hasil uji coba didapatkan nilai *n-gram* dan *window* terbaiknya, berikut hasil nilai *n-gram* dan *window* terbaik; Tabel 3.

Tabel 3. Tabel Hasil Nilai *n-gram* dan *Window* Terbaik

Jumlah Uji Coba	*n-gram (n)*	*Window (w)*	Waktu Proses *(sec)*	Kemiripan *(%)*
1	2	7	0.0092	56.2
2	3	9	0.0135	55.5
3	4	9	0.0101	55.5
4	5	5	0.0088	54.5
5	6	5	0.0084	54.5
6	7	5	0.0081	54.5
7	8	5	0.0085	51.4
8	9	5	0.0082	51.4
9	10	3	0.0094	47.0

Dari Tabel 3 persentase nilai terbaik atau terkecil ditunjukan pada pengujian ke-10, n-gram n=10, window w=3, waktu proses sec=0.0094, serta hasil kemiripan teks 47%. Merujuk pada penelitian yang telah dilakukan sebelumnya [11] dinyatakan bahwa Algoritma Winnowing lebih baik karena memperlihatkan persentase yang lebih kecil dan proses lebih cepat dengan menggunakan data pengujian sebanyak 30 paper, n-*gram*=9 dan *window*=3, waktu 0.0257.

4. Kesimpulan

Berdasarkan hasil penelitian ini menunjukan bahwa Algoritma Winnowing cukup baik dalam pendeteksian kemiripan teks atau plagiarisme yang ditunjukan pada Tabel 3 diatas dengan hasil nilai persentase terkecil dari 10 kali pengujian yang ditunjukan pada pengujian ke 10 dengan nilai *n-gram* *n*=10, *window w*=3, waktu proses *sec*=0.0094, serta hasil kemiripan teks 47 %.

5. Daftar Pustaka

[1] lib.ugm.ac.id, "Panduan Anti Plagiarism," *Perpustakaan Universitas Gadjah Mada*, 2014. https://lib.ugm.ac.id/?page_id=327 (accessed Nov. 14, 2022).

[2] A. H. Purba and Z. Situmorang, "Analisis Perbandingan Algoritma Rabin-Karp dan Levenshtein Distance dalam Menghitung Kemiripan Teks," *J. Tek. Inform. Unika St. Thomas,* vol. 02, pp. 24–32, 2017.

[3] N. Alamsyah, "Perbandingan Algoritma Winnowing dengan Algoritma Rabin Karp untuk Mendeteksi Plagiarisme pada Kemiripan Teks Judul Skripsi," *Technol. J. Ilm.*, vol. 8, no. 3, pp. 124–134, 2017, doi: 10.31602/tji.v8i3.1116.

[4] I. B. K. S. Arnawa, "Implementasi Algoritma *Winnowing* dalam Mendeteksi Plagiarisme pada Tugas Mahasiswa," vol. 10, pp. 220–230, 2022.

[5] W. Hidayat, E. Utami, and A. D. Hartanto, "*Pemilihan* Parameter Terbaik pada Algoritma Winnowing dalam Mendeteksi Tingkat Kesamaan Dokumen Bahasa Indonesia Selection of the Best Parameters in the Winnowing Algorithm in Detecting the Level of Similarity in Indonesian Documents," *Citec J.*, vol. 7, no. 2, 2020.

[6] S. Sugiono, H. Herwin, H. Hamdani, and E. Erlin, "Aplikasi Pendeteksi Tingkat Kesamaan Dokumen Teks: Algoritma Rabin Karp Vs. Winnowing," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 9, no. 1, pp. 82–93, 2018, doi: 10.31849/digitalzone.v9i1.1242.

[7] G. Hizkia, O. Mangundap, H. Sujaini, and H. S. Pratiwi, "Implementasi Algoritma Winnowing pada Aplikasi," vol. 8, no. 1, pp. 147–153, 2022.
[8] A. M. A. K. Parewe, A. Aman, and D. P. M. Dewang, "Perbandingan Algoritma Winnowing dan Algoritma Manber dalam Mendeteksi Berita Hoax di Media Sosial," *Semin. Nas. Teknol. Inf. dan Komput.*, pp. 41–46, 2021.
[9] I. Bagus and K. Surya, "Komparasi Algoritma Winnowing dan Rabin," pp. 345–351.
[10] gurupendidikan, "Metode Penelitian Kualitatif," 2022. https://www.gurupendidikan.co.id/metode-penelitian-kualitatif/ (accessed Nov. 14, 2022).
[11] 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.

EXPLORE: ISSN: 2087-2062, Online ISSN: 2686-181X / DOI: http://dx.doi.org/10.36448/jsit.v13i2.2845