



Uji Plagiarism pada Tugas Mahasiswa Menggunakan Algoritma Winnowing

Moh Sadly Ramli¹, Sugiarto Cokrowibowo², Muh Fahmi Rustan³

^{1,2,3}Informatika, Teknik, Universitas Sulawesi Barat

¹muhsadly11@gmail.com*, ²sugiarto.cokrowibowo@unsulbar.ac.id, ³muhfahmi@unsulbar.ac.id

Abstract

Plagiarism is the act of taking or plagiarizing the work, ideas or ideas of others either intentionally or unintentionally and claiming to be one's own work without mentioning the source or author. Often students plagiarize assignments given by the lecturer so that sometimes students just copy other people's assignments to complete the assignment. So that this can lead to dependence on other people in doing assignments and not being able to independently carry out assignments given by the lecturer. Currently, many plagiarism detection systems have been created to help reduce the level of plagiarism, one of which is the winnowing algorithm. In this study the authors used the winnowing algorithm to detect plagiarism in student assignments, namely programming source code, from the results of research conducted on 10 student assignments using the winnowing algorithm produced various similarity values as a percentage of similarity between two students tasks compared. With an average value of the overall similarity of the 10 tasks, namely 75.12%.

Keywords: winnowing algorithm, plagiarism, similarity, student assignments

Abstrak

Plagiarism adalah sebuah tindakan mengambil atau menjiplak karya, gagasan atau ide-ide orang lain baik sengaja maupun tidak sengaja dan mengklaim sebagai karya sendiri tanpa menyebutkan sumber atau penulisnya. Seringkali mahasiswa melakukan plagiat pada tugas yang diberikan oleh dosen sehingga terkadang mahasiswa hanya menyalin tugas orang lain untuk menyelesaikan tugas tersebut. Sehingga hal ini bisa menyebabkan adanya ketergantungan kepada orang lain dalam mengerjakan tugas dan tidak dapat secara mandiri mengerjakan tugas yang diberikan oleh dosen. Sekarang telah banyak sistem pendeteksi plagiarisme yang dibuat untuk membantu mengurangi tingkat plagiarisme salah satunya yaitu algoritma winnowing. Pada penelitian ini penulis menggunakan algoritma winnowing untuk mendeteksi plagiarisme pada tugas mahasiswa yaitu source code pemrograman, dari hasil penelitian yang telah dilakukan terhadap 10 tugas mahasiswa menggunakan algoritma winnowing menghasilkan nilai similarity yang beragam sebagai persentase kesamaan antara 2 tugas yang dibandingkan. Dengan rata-rata nilai similarity keseluruhan dari 10 tugas yaitu 75,12 %.

Kata kunci: algoritma winnowing, plagiarisme, similarity, tugas mahasiswa

1. Pendahuluan

Saat ini perkembangan teknologi sangatlah pesat dan kini telah mencakup beberapa bidang, salah satunya yaitu bidang pendidikan. Dengan perkembangannya, metode pembelajaran kini beralih menjadi berbasis teknologi dengan berbagai platform yang tersedia. Apalagi ditengah pandemi COVID-19 saat ini tentunya sangat bermanfaat bagi bidang pendidikan dikarenakan adanya himbauan dari pemerintah untuk tetap belajar dari rumah. Namun tak sedikit pula mahasiswa menyalahgunakan teknologi informasi ketika sedang mengerjakan tugas yang diberikan oleh dosen, salah satunya yaitu tindakan *plagiarism*.

Plagiarism adalah sebuah tindakan mengambil atau menjiplak karya, gagasan atau ide-ide orang lain baik sengaja maupun tidak sengaja dan mengklaim sebagai

karya sendiri tanpa menyebutkan sumber atau penulisnya [1]

Seringkali mahasiswa melakukan plagiat pada tugas yang diberikan oleh dosen sehingga terkadang mahasiswa hanya menyalin tugas orang lain untuk menyelesaikan tugas tersebut. Sehingga hal ini bisa menyebabkan adanya ketergantungan kepada orang lain dalam mengerjakan tugas dan tidak dapat secara mandiri mengerjakan tugas yang diberikan oleh dosen. Pada penelitian ini tugas yang akan di uji plagiatnya adalah tugas source code pemrograman dengan bahasa pemrograman java.

Kita tahu sendiri bahwa di bidang pemrograman tidak mempunyai pedoman ataupun standar kutipan layaknya teks esai atau karya tulis ilmiah untuk penyebutan



Lisensi

Lisensi Internasional Creative Commons Attribution-ShareAlike 4.0.

sumbernya namun hal ini akan tetap menjadi sebuah masalah pelanggaran hak cipta.

Salah satu metode dalam mendeteksi plagiarisme yaitu metode dokumen fingerprinting dimana metode ini menelusuri setiap karakter untuk mengkonversinya menjadi sebuah bilangan yang disebut teknik hashing yang kemudian akan menghasilkan nilai fingerprint sebagai acuan dalam menghitung kesamaan kata [2].

Algoritma winnowing adalah salah satu algoritma yang dapat melakukan metode dokumen fingerprinting untuk mendeteksi *plagiarism*. Pendeteksian *plagiarism* source code pemrograman menggunakan algoritma winnowing memiliki tahapan yaitu pertama input teks, Pada tahap ini akan di input 2 file teks yang ingin di uji tingkat plagiasinya. Tahap selanjutnya yaitu *preprocessing*, tahap ini seluruh teks akan melalui proses manipulasi teks dimana seluruh karakter yang tidak relevan seperti simbol dan spasi akan dihapus lalu seluruh huruf kapital akan diubah menjadi huruf kecil. Tahapan selanjutnya adalah pembentukan rangkaian k-gram, yaitu tahapan akan dibentuk rangkaian karakter dengan panjang k dari hasil preprocessing sebelumnya. Panjang k disini berupa parameter yang di input atau ditentukan oleh user [3]. Selanjutnya adalah tahapan perhitungan nilai hash, tahap ini yaitu menghitung nilai hash dari pembentukan rangkaian k-gram untuk mengubah karakter menjadi sebuah kode atau nilai yang kemudian akan menjadi penanda dalam mendapat informasi [4]. Tahap selanjutnya adalah pembentukan *window*, tahap ini nilai hash yang telah diperoleh akan dibagi ke dalam *window* yang berukuran *w*. Dimana *w* adalah nilai parameter yang di input atau ditentukan oleh user [5]. Selanjutnya adalah tahapan pemilihan nilai *fingerprint*, pada tahapan ini akan dipilih nilai hash terkecil dari setiap *window* untuk dijadikan nilai *fingerprint*, jika terdapat nilai hash yang sama maka akan diambil nilai hash yang paling kanan untuk dijadikan nilai *fingerprint* [4]. Tahap terakhir adalah persamaan *jaccard coefficient*, yaitu Nilai fingerprint yang diperoleh akan digunakan dalam persamaan *jaccard coefficient* untuk mengukur persentase kesamaan teks.

Dalam penelitian ini penulis memaparkan beberapa penelitian terdahulu yang digunakan sebagai ajuan dari penelitian ini yaitu memberikan rekomendasi untuk menggunakan algoritma winnowing ataupun algoritma lain yang memiliki metode sama dalam mendeteksi plagiat diantaranya yaitu : *automatic essay grading system for japanese language examination using winnowing algorithm* [5], *detecting documents plagiarism using winnowing algorithm and k-gram method* [6], *rabin-karp and winnowing algorithm for statistics of text document plagiarism detection* [7], deteksi kemiripan dokumen bahasa indonesia menggunakan algoritma smithwaterman dan algoritma nazief & andriani [8], *the implementation of winnowing algorithm for plagiarism detection in moodle-based e-learning* [4], analisis perbandingan algoritma rabin-karp

dan levenshtein distance dalam menghitung kemiripan teks [9], aplikasi pendeteksi dugaan awal plagiarisme pada tugas siswa dan mahasiswa berdasarkan kemiripan isi teks menggunakan algoritma levenshtein distance [10].

Dari penelitian-penelitian yang dilakukan sebelumnya tentunya memiliki perbedaan dengan penelitian ini yaitu pada objek yang di teliti. Dari permasalahan yang ada tujuan yang ingin dicapai pada penelitian ini yaitu untuk mengetahui hasil uji plagiarisme pada tugas mahasiswa dalam hal ini adalah source code pemrograman menggunakan algoritma winnowing. Manfaat dari penelitian ini yaitu diharapkan dapat memudahkan dalam mendeteksi plagiat pada tugas mahasiswa.

2. Metode Penelitian

Metode penelitian berisi langkah-langkah yang akan dilakukan dalam penelitian ini yang terdiri dari pengumpulan data dan menghitung *similarity* menggunakan algoritma winnowing

2.1 Pengumpulan Data

Pengumpulan data merupakan proses untuk mengumpulkan data yang berguna untuk penelitian yang dilakukan. Metode pengumpulan data yang digunakan dalam penelitian ini yaitu studi dokumen. Studi dokumen dilakukan untuk mendapatkan data dari berbagai dokumen dalam hal ini tugas mahasiswa untuk kemudian dilakukan analisis terhadap data tersebut serta hasil dari penelitian yang dilakukan sebelumnya.

2.2 Menghitung *Similarity* Menggunakan Algoritma Winnowing

Algoritma winnowing digunakan untuk mendapatkan nilai *similarity* pada tugas mahasiswa. Tahapannya yaitu sebagai berikut:

2.2.1 Input Teks

Tahap ini akan di input teks yang ingin diuji tingkat plagiasinya. Dimana teksnya adalah *source code* pemrograman java seperti Gambar 1.

```
int n = sc.nextInt();
System.out.println("Output: ");
```

Gambar 1. Contoh Input Teks

2.2.2 *Preprocessing*

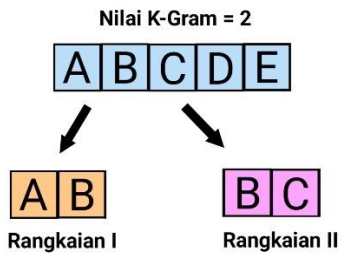
Tahapan ini inputan teks akan melalui manipulasi teks yaitu mengubah menjadi huruf kecil, menghapus simbol-simbol dan karakter spasi seperti Gambar 2.

```
intnscnextintssystemoutprintlnoutput
```

Gambar 2. Tahapan *Preprocessing*

2.2.3 Pembentukan Rangkaian K-Gram

Tahapan ini akan dibentuk rangkaian karakter sesuai dengan panjang *k* secara berkelanjutan.



Gambar 3. Contoh Tahapan Rangkaian K-Gram

Pada gambar 3 dapat dilihat contoh tahapan untuk membuat rangkaian k-gram dengan nilai k-gram = 2 dimana rangkaian 1 yaitu diambil dari karakter pertama hingga k lalu rangkaian selanjutnya diambil dari karakter kedua hingga k dan seterusnya. Untuk hasil rangkaian k-gram dari hasil *preprocessing* diatas yaitu seperti Gambar 4.

in nt tn ns sc cn ne ex xt ti in nt ts sy ys st te em mo
 ou ut tp pr ri in nt tl ln no ou ut tp pu ut

Gambar 4. Hasil Rangkaian K-Gram

2.2.4 Perhitungan Nilai Hash

Tahapan ini seluruh rangkaian yang telah dibentuk akan diubah menjadi nilai atau kode yang menjadi penanda dari rangkaian karakter tersebut. Nilai hash dihasilkan dari fungsi hash yang mana fungsi tersebut dapat dilihat pada persamaan 1.

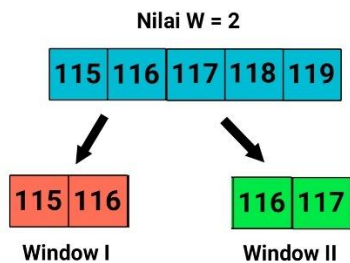
$$H = c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_k * b^{(k-k)} \quad (1)$$

Dengan H adalah nilai hash, c adalah nilai ascii karakter, b adalah bilangan prima, dan k adalah banyak karakter [2].

Dapat dilihat pada persamaan (1) terdapat suatu keterangan yaitu b yang merupakan bilangan prima. b disini merupakan nilai parameter yang bebas dan ditentukan oleh user.

2.2.5 Pembentukan Window

Pada proses ini nilai hash yang telah diperoleh akan dibagi ke dalam *window* yang berukuran w.



Gambar 5. Contoh Pembentukan *window*

Pada gambar 5 dapat dilihat contoh pembentukan *window* dengan nilai w = 2, dimana di setiap *window* terdapat dua nilai hash. *Window* 1 dimulai dari nilai hash

yang pertama hingga w, *window* 2 dimulai dari nilai hash yang kedua hingga w dan seterusnya. Untuk hasil pembentukan *window* dapat dilihat pada gambar 6.

320 336	336 342	342 335	335 329
329 308	308 321	321 322	322 356
356 337	337 320	320 336	336 347
347 351	351 357	357 346	346 333
333 311	311 329	329 339	339 350
350 344	344 338	338 333	333 320
320 336	336 340	340 326	326 331
331 339	339 350	350 344	344 341
341 350			

Gambar 6. Hasil Pembentukan *Window*

Dapat dilihat pada gambar 6 hasil pembentukan *window* dengan total *window* yang terbentuk sebanyak 33.

2.2.6 Pemilihan Nilai *Fingerprint*

Tahapan ini akan dipilih nilai hash terkecil dari setiap *window* untuk dijadikan nilai *fingerprint*, jika terdapat nilai hash yang sama maka akan diambil nilai hash yang paling kanan untuk dijadikan nilai *fingerprint*. Untuk nilai *fingerprint* yang telah dipilih adalah Gambar 7.

320, 336, 335, 329, 308, 321, 322, 337, 347, 351,
 346, 333, 311, 339, 344, 338, 326, 331, 341

Gambar 7. Nilai *Fingerprint* yang dipilih

2.2.7 Persamaan *Jaccard Coefficient*

Untuk mengukur nilai kesamaan atau *similarity* digunakan persamaan *jaccard coefisient*. Untuk pesamaannya dapat dilihat pada persamaan 2.

$$Similarity(x, y) = \frac{|x \cap y|}{|x \cup y|} \times 100 \% \quad (2)$$

Dengan x adalah nilai *fingerprint* teks 1, y adalah nilai *fingerprint* teks 2, $x \cap y$ adalah jumlah *fingerprint* yang sama dan $x \cup y$ adalah total seluruh *fingerprint* [11].

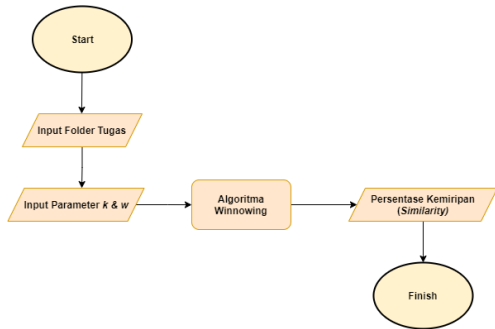
Dapat dilihat pada persamaan (2) untuk mencari *similarity* antara 2 teks maka terlebih dahulu akan dicari jumlah *fingerprint* yang sama antara teks 1 dan teks 2, lalu dimasukkan ke dalam persamaan (2).

3. Hasil dan Pembahasan

Pada tahap ini menjelaskan tentang mengenai rancangan dari sistem serta menjelaskan tentang implementasi yang ada.

3.1 Rancangan Sistem

Penjelasan mengenai rancangan sistem algoritma winnowing guna mendeteksi *plagiarism* pada tugas mahasiswa dapat dilihat pada gambar berikut.



Gambar 8. Rancangan Sistem

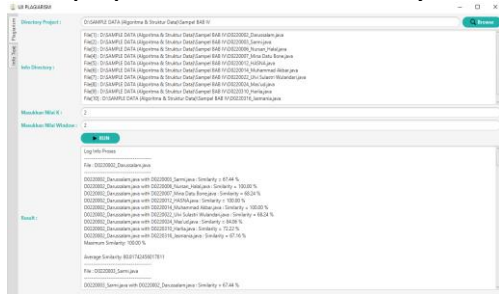
Dari gambar 8. Dapat dilihat bahwa sistem dimulai dari penginputan folder tugas dimana isi dari folder tersebut merupakan file tugas yang akan diuji tingkat plagiasinya, ekstensi dari tugas yang diuji yaitu (.java). Setelah folder di input maka file yang berada di folder tersebut akan dibaca oleh sistem, kemudian akan di input parameter k dan w yang ditentukan oleh user lalu selanjutnya akan di proses pada algoritma winnowing untuk menghasilkan persentase kemiripan (*similarity*) dari setiap tugas atau file yang diuji.

3.2 Implementasi

Tahap implementasi merupakan tahap untuk menjelaskan implementasi dari tahap analisa yang dilakukan. Pengimplementasian penelitian ini menggunakan *javaFx* untuk membuat tampilan sistem.

a. Antarmuka form utama

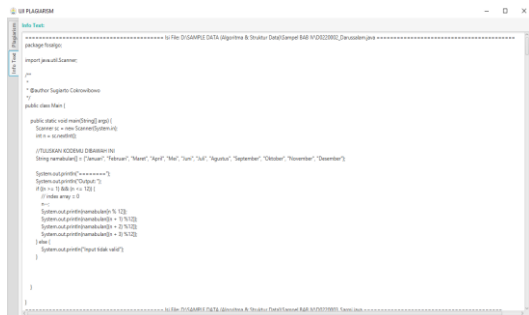
Antarmuka form utama berisi tampilan tambah folder, input parameter dan hasil *similarity*.



Gambar 9. Antarmuka Form Utama

b. Antarmuka info text

Antarmuka info text berisi tampilan text dari file yang ditambahkan pada form utama.



Gambar 10. Antarmuka Info Text

3.3 Algoritma Winnowing

Pada pengujian ini diuji beberapa nilai parameter untuk mengetahui tingkat akurasi nilai *similarity* yang dihasilkan. Untuk sampel tugas yang diuji pada pengujian ini yaitu tugas Hasna dan Ulil Amri. Nilai parameter yang akan diuji yaitu nilai k dan nilai w yang akan diuraikan pada tabel berikut.

Tabel 1. Hasil Pengujian Parameter Nilai k dan Nilai w

No	K-Gram (k)	Window (w)	Similarity
1.	2	2	61,29 %
	2	4	52,87 %
	2	6	41,33 %
	2	8	37,87 %
2	4	2	37,16 %
	4	4	29,43 %
	4	6	23,88 %
	4	8	20,37 %
3	6	2	23,41 %
	6	4	19,26 %
	6	6	16,03 %
	6	8	15,50 %
4	8	2	13,74 %
	8	4	11,47 %
	8	6	8,92 %
	8	8	8,05 %

Dari tabel 1 dapat dilihat kombinasi parameter antara k dan w bahwa semakin besar kombinasi parameter tersebut maka akan kecil nilai *similarity* yang dihasilkan begitupun sebaliknya. Dengan demikian untuk pemilihan nilai parameter pada pengujian ini digunakan kombinasi nilai $k = 2$ dan $w = 2$ karena semakin kecil nilai kombinasi parameter akan semakin banyak potongan kata yang bisa diperoleh untuk meningkatkan peluang menemukan kesamaan antara dua sampel yang dibandingkan.

Untuk parameter nilai b akan diuji pada tabel tersendiri karena pengaruh terhadap *similarity* tidak terlalu signifikan. Untuk pengujian nilai b dapat dilihat pada tabel berikut.

Tabel 2. Hasil Pengujian Parameter nilai b

Nilai Bilangan Prima (b)	Similarity
2	61,29 %
3	61,11 %
4	65,74 %
5	61,01 %
6	61,41 %
7	62,77 %

Dapat dilihat pada tabel 2 pengujian terhadap parameter nilai b dengan nilai $k = 2$ dan nilai $w = 2$. Perbedaan nilai parameter b tidak memberikan pengaruh yang signifikan terhadap nilai *similarity* yang dihasilkan.

3.3.1 Hasil Uji Coba Pengujian Pada Tugas Mahasiswa

Pada pengujian ini akan diambil secara random sampel sebanyak 10 tugas mahasiswa prodi teknik informatika Universitas Sulawesi Barat angkatan 2020 berupa *source code* pemrograman yaitu mata kuliah Algoritma

dan Struktur Data yang kemudian akan menghasilkan nilai *similarity* antara tugas yang dibandingkan. Pengujian tugas ini yaitu akan diuji sampel per sampel dimana sampel 1 dengan 9 sampel lainnya, sampel 2 dengan 9 sampel lainnya dan seterusnya. Untuk rata-rata *similarity* dari hasil pengujian tugas mahasiswa ini dapat dilihat pada tabel berikut.

Tabel 3. Hasil Pengujian Tugas Mahasiswa

No	Pengujian	Rata- Rata Similarity
1	Sampel 1	80,81 %
2	Sampel 2	80,81 %
3	Sampel 3	72,47 %
4	Sampel 4	80,81 %
5	Sampel 5	80,81 %
6	Sampel 6	73,22 %
7	Sampel 7	67,56 %
8	Sampel 8	64,02 %
9	Sampel 9	73,22 %
10	Sampel 10	77,51 %

3.3.2 Pengujian *Black Box*

Pengujian *black box* merupakan pengujian yang dilakukan dengan mengamati hasil eksekusi dari perangkat lunak. Hasil pengujian melakukan pengujian *black box* dapat dilihat pada tabel berikut.

Tabel 2. Hasil Pengujian *Black Box*

No	Model Uji	Keberhasilan
1.	Menambahkan folder tugas	100%
2.	Tampilkan isi folder yang ditambahkan	100%
3.	Input Parameter	100%
4.	Run aplikasi	100%
5.	Tampilkan Hasil <i>similarity</i>	100%
6.	Tampilkan isi text dari file yang dipilih	100%

4. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan dapat disimpulkan bahwa algoritma *Winnowing* dapat digunakan untuk menguji plagiat pada tugas source code pemrograman dengan pemilihan parameter *k-gram*, bilangan prima dan *window* dengan nilai rata-rata *similarity* keseluruhan dari 10 sampel yang diuji yaitu 75,12 %. Pemilihan nilai parameter *k* dan *w* sangat berpengaruh terhadap nilai *similarity* yang dihasilkan

yaitu semakin besar nilai *k* dan *w* maka semakin kecil nilai *similarity* yang dihasilkan dan semakin kecil nilai *k* dan *w* maka semakin besar juga nilai *similarity* yang dihasilkan. Sehingga digunakan nilai *k* = 2 untuk meningkatkan peluang dalam menemukan kesamaan antara kedua sampel yang dibandingkan mengingat algoritma *winnowing* ini bekerja dengan cara melakukan pengecekan di setiap code yang diuji.

Daftar Rujukan

- [1] A. Abdi, N. Idris, R. M. Alguliyev, and R. M. Aliguliyev, "PDLK: Plagiarism detection using linguistic knowledge," *Expert Syst. Appl.*, 2015, doi: 10.1016/j.eswa.2015.07.048.
- [2] N. Alamsyah, "Perbandingan Algoritma *Winnowing* Dengan Algoritma Rabin Karp Untuk Mendeteksi Plagiarisme Pada Kemiripan Teks Judul Skripsi," *Technol. J. Ilm.*, 2017, doi: 10.31602/tji.v8i3.1116.
- [3] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local Algorithms for Document Fingerprinting," 2003.
- [4] E. G. Hasan, A. Wicaksana, and S. Hansun, "The Implementation of *Winnowing* Algorithm for Plagiarism Detection in Moodle-based E-learning," 2018, doi: 10.1109/ICIS.2018.8466429.
- [5] A. Agung Putri Ratna, D. Lalita Luhurkinanti, I. Ibrahim, D. Husna, and P. Dewi Purnamasari, "Automatic Essay Grading System for Japanese Language Examination Using *Winnowing* Algorithm," 2018, doi: 10.1109/ISEMANTIC.2018.8549789.
- [6] R. Sutoyo *et al.*, "Detecting documents plagiarism using *winnowing* algorithm and *k-gram* method," 2018, doi: 10.1109/CYBERNETICSCOM.2017.8311686.
- [7] D. Leman, M. Rahman, F. Ikorasaki, B. S. Riza, and M. B. Akbbar, "Rabin Karp and *Winnowing* Algorithm for Statistics of Text Document Plagiarism Detection," 2019, doi: 10.1109/CITSM47753.2019.8965422.
- [8] B. Sari and Y. Sibaroni, "Deteksi Kemiripan Dokumen Bahasa," vol. 4, pp. 87–98, 2019, doi: 10.21108/indojc.2019.4.3.365.
- [9] 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.
- [10] H. Ajie and A. Surya Bangsa, "Aplikasi Pendeteksi Dugaan Awal Plagiarisme Pada Tugas Siswa Dan Mahasiswa Berdasarkan Kemiripan Isi Teks Menggunakan Algoritma *Levenshtein Distance*," *PINTER J. Pendidik. Tek. Inform. dan Komput.*, 2017, doi: 10.21009/pinter.1.1.4.
- [11] S. Sunardi, A. Yudhana, and I. A. Mukaromah, "Implementasi Deteksi Plagiarisme Menggunakan Metode *N-Gram* Dan *Jaccard Similarity* Terhadap Algoritma *Winnowing*," *Transmisi*, 2018, doi: 10.14710/transmisi.20.3.105-110.