

IMPLEMENTASI DETEKSI PLAGIARISME MENGGUNAKAN METODE *N-GRAM* DAN *JACCARD SIMILARITY* TERHADAP ALGORITMA *WINNOWER*

Sunardi¹, Anton Yudhana², dan Iif Alfiatul Mukaromah^{3*)}

Program Studi Teknik Elektro, Universitas Ahmad Dahlan^{1, 2}.

Program Studi Magister Teknik Informatika, Universitas Ahmad Dahlan^{3*)}

Kampus 3 UAD Jl. Prof Soepomo, Janturan, Yogyakarta, 55164, Indonesia.^{1, 2, 3}

E-Mail: Sunardi@mti.uad.ac.id¹, eyudhana@ee.uad.ac.id², iifam1604@gmail.com^{3*)}

Abstrak

Fenomena plagiarisme dalam dunia akademik bukanlah hal yang baru di Indonesia, perguruan tinggi di Indonesia sudah cukup lama dicemari oleh tindak plagiarisme. Dengan didukungnya kemajuan teknologi yang semakin canggih, salah satunya jaringan internet yang mendunia, memudahkan, cepat, praktis dan murah. Sehingga banyak orang lebih memilih menggunakan internet untuk menyelesaikan tugasnya seperti membuat karya ilmiah dengan cara *copy-paste* karya orang lain dari internet tanpa menyebutkan sumber yang digunakan untuk menyelesaikan tugasnya. Hal tersebut bisa diatasi secara dini menggunakan metode *n-gram* untuk mengambil potongan karakter huruf pada dokumen yang akan dicocokkan dan *Jaccard similarity* untuk menghitung berapa persen kesamaan pada sebuah dokumen yang diproses menggunakan algoritma *winnower*. Algoritma *winnower* merupakan algoritma yang digunakan untuk mencocokkan karakter huruf atau angka pada dua dokumen dengan teknik *hashing*.

Kata kunci: N-gram, Jaccard Similarity, Algoritma Winnower.

Abstract

The phenomenon of plagiarism in the academic world is not new in Indonesia, universities in Indonesia have long been contaminated by the act of plagiarism. With the advancement of technological advancements that increasingly sophisticated, one of the Internet network worldwide, easy, fast, practical and inexpensive. So many people prefer to use the internet to accomplish tasks such as making scientific work by copy-pasting someone else's work from the internet without mentioning the source used to complete the task. This can be resolved early using the *N-gram* method to extract the chunks of the letters on the matching document and *Jaccard similarity* to calculate what percentage of similarity in a document is processed using *winnower* algorithm. The *winnower* algorithm is an algorithm used to match letters or numbers on two documents by hashing technique.

Keywords: N-gram, Jaccard Similarity, Algoritma Winnower.

1. Pendahuluan

Plagiarisme merupakan tindakan kejahatan yang sudah tidak asing lagi dalam dunia akademik di Indonesia. Tindakan plagiarisme dalam kamus Inggris *The Oxford Advanced Learner's Dictionary* mendefinisikan plagiarisme sebagai *'to take somebody's else idea or words and use them as if they were one's own'*[1]. Plagiarisme merupakan perbuatan yang tidak baik disebabkan mengambil karya (ide, gagasan, dan pikiran) orang lain tanpa izin, tanpa menyatakan karya yang diambilnya merupakan karya orang lain/tidak menyantumkan sumbernya, dan mengakui bahwa karya itu miliknya. Tindakan tersebut sama seperti tindakan mencuri.

Tindakan plagiarisme disuburkan oleh perkembangan internet sebagai teknologi yang semakin canggih, kemudahan mendapatkan suatu informasi, penggunaan yang cepat dan murah sehingga manusia lebih memilih mencari bahan atau referensi tugasnya dalam internet dibandingkan harus membeli buku. Akan tetapi banyak orang yang menyalah gunakan internet untuk hal yang negatif salah satunya yaitu mengambil karya orang lain dari internet tanpa menyatakan sumbernya[2].

Dampak buruk dari tindakan plagiarisme dapat mengakibatkan kemunduran suatu bangsa, karena tidak ada keinginan untuk maju atau memunculkan hal-hal yang baru, bahkan tindakan plagiarisme juga merupakan

bentuk pelanggaran ketentuan dan peraturan pemerintah[1]. Untuk itu perlu adanya alat yang dapat meminimalisir atau mencegah agar tindakan mencuri karya orang lain itu tidak terulang kembali, khususnya mencegah kejahatan dalam dunia akademik.

Beberapa penelitian sebelumnya telah membuat pendeteksian plagiarisme menggunakan beberapa algoritma yang mempunyai fungsi sama sebagai *document fingerprint*, tetapi permasalahan yang sering terjadi pada hasil atau tingkat keakurasian [2]. Penggunaan model Bayesian juga telah digunakan[3], namun tidak sepenuhnya dapat mendeteksi kemiripan. *N-gram* merupakan algoritma yang akan digunakan untuk mengambil potongan huruf sejumlah *n* dan mempunyai pengaruh yang tinggi terhadap hasil *similarity*[4]. Algoritma yang akan digunakan untuk mengetahui tingkat kemiripan teks atau dokumen yaitu *Jaccard Similarity* atau sering disebut *Jaccard Coefficient*.

Jaccard Similarity merupakan algoritma yang berfungsi untuk membandingkan antar dua dokumen dengan menghitung kemiripan atau perbedaan dari dokumen tersebut[5]. Untuk menjalankan *n-gram* dan *Jaccard Similarity* diperlukan algoritma yang berfungsi sebagai *document fingerprint* dan algoritma yang akan digunakan untuk mendukung tersebut yaitu algoritma *winnowing*. Algoritma *winnowing* merupakan algoritma yang dapat digunakan untuk membantu mendeteksi kemiripan kata atau dokumen yang akan digunakan sebagai suatu cara untuk mendeteksi adanya suatu tindak plagiarisme[6]. Diharapkan dengan adanya alat pendeteksi plagiarisme menggunakan metode *n-gram* dan *Jaccard similarity* terhadap algoritma *winnowing* dapat memberikan keakurasian yang tepat dan mengatasi tindakan suatu plagiarisme yang terjadi di dunia akademik maupun dunia non akademik.

2. Metode

2.1. N-Gram

N-gram merupakan metode yang dilakukan dengan mengambil rangkaian *substring* dari *string* sejumlah *n* (rangkain token sepanjang *n*)[7]. Metode *n-gram* sering digunakan pada teknik analisis statistik dan juga bahasa[8]. Dalam *game n-gram* digunakan untuk *action prediction* dengan mengambil rangkain aksi *player* sejumlah yang ditentukan dalam aksi *player* secara berkelanjutan[8]. Akan tetapi *n-gram* paling banyak digunakan dalam teks *mining* (pengolahan kata) dan pengolahan bahasa. Dalam mendeteksi plagiarisme, *n-gram* digunakan untuk mengambil potongan-potongan karakter huruf atau pemisahan *string* sepanjang *n* dari sebuah kata atau dokumen secara berkelanjutan (kontinuitas) hingga bergeser sesuai dengan *offset* yang diberikan atau akhir dari suatu kata atau dokumen[7][9].

```
private function n_gram($word, $n) {
    $ngrams = array();
    $length = strlen($word);
    for($i = 0; $i < $length; $i++) {
        if($i > ($n - 2)) {
            $ng = '';
            for($j = $i - $n + 1; $j >= 0; $j--) {
                $ng .= $word[$i - $j];
            }
            $ngrams[] = $ng;
        }
    }
    return $ngrams;
}
```

Gambar 1. Program *N-gram*

Gambar 1 merupakan program PHP dari *n-gram* dimana dokumen atau sekumpulan kata akan diproses dan akan dibentuk sebanyak *n-gram* atau memisahkan *string* sepanjang *n* yang akan dihitung pergeserannya secara terus menerus ke depan sejumlah nilai *n* sampai akhir dokumen. Sebagai contoh *n-gram* dari kata “Magister Teknik Informatika” dengan nilai *N=4* maka (magi agis gist iste ster tert erte rtek tekni knik niki ikin kinf info nfor form orma rmat mati atik tika).

N-gram dalam deteksi plagiarisme sangat mempengaruhi tingkat akurasi atau tingkat *similarity* dari sebuah dokumen yang dibandingkan, maka teknik *n-gram* dipadukan dengan pendekatan statistika untuk memperoleh *similarity* dari antar dokumen seperti *Simple Matching*, *Cosine Similarity*, *Jaccard Similarity* dan *Dice Coefficient*. Ada 2 (dua) teknik *n-gram* yaitu membagi *string* menjadi suatu set *substring* dengan panjang *n* (*overlapping n-gram*) dan mengecek untuk membentuk *substring* yang memiliki struktur yang sama[9].

2.2. Jaccard Similarity

Jaccard Similarity atau *Jaccard Coefficient* merupakan algoritma yang fungsinya untuk membandingkan dua *sample* yaitu dokumen yang satu dengan yang lainnya berdasarkan kata yang dimilikinya. *Jaccard similarity* biasanya digunakan untuk membandingkan dokumen dan menghitung nilai kemiripan (*similarity*) dari dua buah objek atau dokumen[5][10][11]. *Jaccard similarity* dapat dirumuskan sebagai berikut:

$$Similarity(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad (1)$$

Dimana:

X = Dokumen 1
Y = Dokumen 2

Rumus 1 merupakan dari rumus *Jaccard Similarity* atau *Jaccard Coefficient* yang digunakan untuk mencari persamaan dan perbedaan pada dua *sample*. sebagai contoh diketahui X “Magister Teknik Informatika

Yogyakarta”, dan Y “Magister Teknik Sipil Yogyakarta”. Maka akan menghasilkan nilainya:

$$\text{Similarity}(X, Y) = \frac{|\text{Magister Teknik Yogyakarta}|}{|\text{Magister Teknik Informatika Sipil Yogyakarta}|}$$

$$\text{Similarity}(X, Y) = \frac{3}{5} = 0,6 \times 100\% = 60\%$$

Dari dua contoh *sample* di atas setelah dihitung kesamaannya menggunakan *Jaccard similarity*, bahwa kedua *sample* tersebut memiliki kesamaan atau kemiripan sebesar 60%. Di bawah ini merupakan program dari *Jaccard similarity* atau *Jaccard Coefficient*.

```
private function
jaccard_coefficient($
fingerprint1, $fingerprint2){
$arr_intersect =
array_intersect( $
fingerprint1, $fingerprint2
);
$arr_union = array_merge( $
fingerprint1, $fingerprint2
);

$count_intersect_fingers =
count($arr_intersect);
$count_union_fingers = count( $
arr_union );

$coefficient = $
count_intersect_fingers /
(count_union_fingers - $
count_intersect_fingers
);
```

Gambar 2. Potongan Program dari *Jaccard Similarity* atau *Jaccard Coefficient*

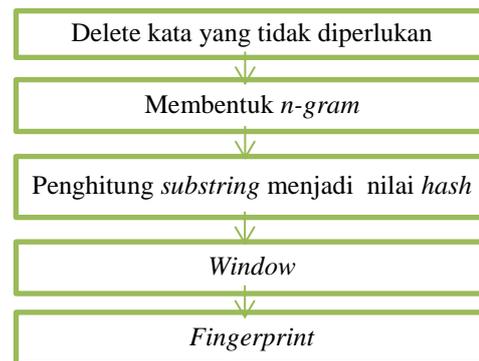
Gambar 2 menunjukkan potongan program yang akan digunakan untuk perhitungan derajat kesamaan atau untuk mengecek suatu kemiripan pada dua buah *sample* atau dokumen menggunakan *Jaccard Similarity* atau *Jaccard Coefficient*.

Ukuran tingkat kesamaan [12]:

- 0% : dua dokumen tidak memiliki kesamaan
- <15% : memiliki sedikit kesamaan
- 15-50% : termasuk dalam katagori plagiarisme sedang
- 50% : mendeteksi adanya plagiarisme
- 100% : dokumen tersebut plagiarisme

2.3. Algoritma *Winnowing*

Algoritma *winnowing* merupakan salah satu algoritma yang berfungsi sebagai *document fingerprint* atau algoritma yang digunakan untuk mendeteksi tindakan plagiarisme dengan menggunakan teknik *hashing*. *Input* dari algoritma *winnowing* berupa dokumen teks, dan akan menghasilkan keluaran berupa kumpulan nilai *hash* yang terbentuk dari peritungan ASCII pada setiap karakter. Dan nilai-nilai *hash* yang akan digunakan sebagai *fingerprint* untuk mendeteksi adanya suatu tindakan plagiarisme [13][14]. Gambar 3 merupakan konsep algoritma *winnowing*.



Gambar 3. Konsep Algoritma *Winnowing*

Konsep dari algoritma *winnowing* yang ditujukan pada gambar 3 yaitu menghapus karakter yang tidak relevan, membentuk *n-gram* dengan panjang *n*, menghitung nilai *hash*, membentuk nilai *window*, dan memilih nilai *hash* sebagai *document fingerprint*[6]. Nilai *hash* didapatkan dari rumus *Rolling hash*.

Rolling hash adalah metode *hashing* yang digunakan untuk mencari nilai *hash* dari rangkaian *grams* yang telah terbentuk dan memberikan kemampuan untuk menghitung nilai tanpa mengulangi seluruh *string*. Nilai *hash* merupakan nilai numerik yang dibentuk dari kode ASCII[15].

Berikut rumus *rolling hash*

$$H(C_1..C_l) = C_1 \cdot b^{(l-1)} + C_2 \cdot b^{(l-2)} + \dots + C_{(l-1)} \cdot b + C_l \quad (2)$$

$$H(c_2 \dots c_{l+1}) = (H(c_1 \dots c_l) - c_1 \cdot b^{(l-1)}) \cdot b + c^{(l-1)} \quad (3)$$

Dimana:

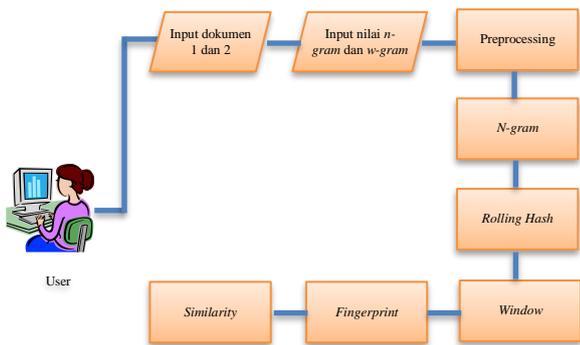
- $H(C_1..C_l)$ = nilai *hash*
- C_l = nilai ASCII karakter ke -*l* pada *string*
- l = panjang *string*
- b = nilai basis *hash*

Perhitungan awal pada rangkaian *n-gram* paling awal dihitung menggunakan rumus nomer 2, dan rangkaian *n-gram* berikutnya sampai rangkaian *gram* terakhir di hitung menggunakan rumus nomer 3, sehingga proses akan jauh lebih cepat karna tidak menghitung lagi dari awal.

Algoritma *winnowing* merupakan eksitensi dari algoritma *Rabin-karp*, proses yang dilakukan hampir sama hanya pada algoritma *winnowing* ditambahkan dengan konsep *window*.

2.4. Perancangan Sistem

Perancangan Sistem yang akan digunakan untuk membuat aplikasi deteksi plagiarisme menggunakan metode *n-gram* dan *Jaccard similarity* atau *Jaccard coefficient* terhadap algoritma *winnowing*.



Gambar 4. Tahapan Algoritma W-innowing

Gambar 4 merupakan tahapan algoritma *w-innowing*, pada awal antarmuka *user* diperlukan untuk memasukan dokumen atau teks yang akan digunakan sebagai pembandingan dan dokumen yang akan dibandingkan, kemudian *user* diminta untuk menginputkan nilai *n-gram* dan *w-gram* yang akan digunakan untuk memproses dari algoritma *w-innowing*. Pada tahap selanjutnya yaitu pemisahan karakter yang tidak relevan atau sering disebut dengan tahap *preprocessing*, pada tahap ini akan dihilangkan karakter yang tidak diperlukan seperti spasi, tanda baca (*Tokenizing*) dan merubah huruf kapital menjadi huruf kecil (*case Folding*). Selanjutnya sistem akan membentuk nilai *n-gram* sepanjang *n*, potongan karakter yang diambil sepanjang *n* akan dihitung untuk dicari nilai *hash* dengan menggunakan rumus *rolling hash*. Nilai-nilai *hash* yang sudah terbentuk, maka tahap selanjutnya yaitu tahap algoritma *w-innowing* (pembentukan *window* atau *w-gram*) yang akan digunakan untuk mencari atau memilih *fingerprint*. Tahap terakhir yaitu penggunaan *Jaccard similarity* digunakan untuk membandingkan dari kedua dokumen atau *sample*.

3. Hasil dan Analisa

3.1. Uji Coba

Hasil uji coba dari pendeteksi plagiarisme menggunakan metode *n-gram* dan *Jaccard Similarity* atau *Jaccard Coefficient* terhadap algoritma *w-innowing* akan di tampilkan pada tabel 1 dengan menggunakan *n-gram* dan *w-gram* yang berbeda-beda.

Tabel 1 menjelaskan pengaruh dari penggunaan *n-gram* dan *w-gram* terhadap hasil *similarity*. Hasil pengujian terhadap dua *sample* atau dokumen pada tabel 1 dengan nilai *n-gram* dan *w-gram* yang berbeda memiliki drajat kesamaan yang berbeda-beda. Semakin kecil nilai *n-gram* dan *w-gram* maka drajat kesamaan atau *similarity* terhadap dua dokumen mempunyai drajat kesamaan yang tinggi, sedangkan jika semakin besar nilai *n-gram* dan *w-gram* maka drajat kesamaan pada dua dokumen itu memiliki drajat yang rendah, jika nilai *n-gram* dan *w-gram* sama. Pada tabel 1 membuktikan *n-gram* dan *w-gram* dengan nilai 3 menghasilkan *similarity* 35%,

sedangkan *n-gram* dan *w-gram* dengan nilai 6 menghasilkan *similarity* yang rendah yaitu 18,18%. Uji coba pada tabel 1 nomer 2 dan 3 menghasilkan drajat kesamaan atau *similarity* yang jauh lebih tinggi dari pada pengujian pada nomer 1. Penginputan nilai yang berbeda pada nilai *n-gram* dan *w-gram* dapat menghasilkan nilai *similarity* yang tinggi. Semakin kecil nilai *n-gram* dan *w-gram* maka akan semakin sering potongan itu akan dicocokkan dan sering ditemukan, jika sebaliknya maka akan semakin jarang data itu dicocokkan atau di temukan.

Table 1. Uji Coba Pendeteksi Plagiarisme Menggunakan *N-gram* dan *W-gram* yang berbeda Terhadap Hasil *Similarity* dengan Metode *Jaccard Similarity*

No	Input Dokumen		<i>N-gram</i>	<i>W-gram</i>	<i>Similarity</i>
	Kalimat 1	Kalimat 2			
1.	Magister Teknik Elektro	Magister Teknik Informatika	3	3	35,71%
2.	Magister Teknik Elektro	Magister Teknik Informatika	5	3	41,67%
3.	Magister Teknik Elektro	Magister Teknik Informatika	4	5	45,45%
4.	Magister Teknik Elektro	Magister Teknik Informatika	4	4	41,67%
5.	Magister Teknik Elektro	Magister Teknik Informatika	6	6	18,18%

3.2. Implementasi

Berikut adalah implementasi deteksi plagiarisme pada dunia tulis-menulis dengan menggunakan metode *n-gram* dan *Jaccard Similarity* atau *Jaccard Coefficient* terhadap algoritma *w-innowing*. Gambar 5 merupakan proses yang digunakan *user* untuk menginputkan dokumen yang akan dibandingkan.

Gambar 5. Input Dokumen

Gambar 5 menunjukan antarmuka pertama dari algoritma *w-innowing* yang akan digunakan untuk mendeteksi plagiarisme, *user* bisa memasukan dokumen pembandingan dan dokumen yang akan dibandingkan. Selanjutnya pada Gambar 6 merupakan *input* dari *n-gram*, *w-gram* dan bilangan prima.

Gambar 6. Input *N-gram*, *W-gram* dan Bilangan Prima

Gambar 6 menunjukkan sebelum algoritma *winnowing* melakukan proses, *user* terlebih dahulu harus menentukan dan menginputkan *n-gram*, *w-gram* dan bilangan prima yang akan menentukan hasil akhir.

N Gram 1.	Rolling Hash
indo ndon done ones nesi esia siai lain aind inda ndah	Rolling Hash 1. 3182 3224 3130 3290 3238 3150 3278 3096 3032 3154 3156
N Gram 2.	Rolling Hash 2.
indo ndon done ones nesi esia siai lain aind inda ndah	3182 3224 3130 3290 3238 3150 3278 3096 3032 3154 3156

Windows
Windows 1. W-1 : {3182 3224 3130 3290} W-2 : {3224 3130 3290 3238} W-3 : {3130 3290 3238 3150} W-4 : {3290 3238 3150 3278} W-5 : {3238 3150 3278 3096} W-6 : {3150 3278 3096 3032} W-7 : {3278 3096 3032 3154} W-8 : {3096 3032 3154 3156}
Windows 2. W-1 : {3182 3224 3130 3290} W-2 : {3224 3130 3290 3238} W-3 : {3130 3290 3238 3150} W-4 : {3290 3238 3150 3278} W-5 : {3238 3150 3278 3096} W-6 : {3150 3278 3096 3032} W-7 : {3278 3096 3032 3154} W-8 : {3096 3032 3154 3156}

Gambar 7. Proses pembentukan *N-gram*, *Rolling Hash*, dan Pembentukan *Window*

Fingerprints Kalimat 1
1464 1464 1464 1438 1416 1416 1416 1396
Fingerprints Kalimat 2
1464 1464 1464 1438 1416 1416 1416 1396
Jumlah Fingerprints kalimat 1 = 8
Jumlah Fingerprints kalimat 2 = 8
Union (Gabungan) Fingerprints 1 dan 2 = 16
Intersection (Fingerprints yang sama) = 8
(Union - Intersection) = 8
Prosentase Plagiarisme
Koefisien Jaccard = (Intersection / (Union-Intersection)) * 100
(8/8) * 100 = 100 %

Gambar 8. Hasil akhir dari algoritma *winnowing* menggunakan *Jaccard Similarity*

Setelah ditentukan nilai *n-gram* dan *w-gram* maka baru bisa dilakukan proses dari algoritma *winnowing* untuk mendeteksi plagiarisme. Gambar 5 sistem sudah melakukan proses pengecekan, sistem pertama akan menghilangkan karakter yang tidak relevan yaitu *tokenizing* (menghilangkan spasi dan tanda baca) dan *cas folding* (merubah huruf kapital menjadi huruf kecil). setelah itu sistem baru akan membentuk *n-gram* sepanjang *n*, kemudian melakukan teknik *hashing* dan pembentukan *window* atau *w-gram* untuk menghasilkan *fingerprint* dari setiap dokumen dimana akan diambil nilai *hash* yang paling kecil untuk dijadikan *fingerprint*, jika terdapat dua atau lebih nilai *hash* yang paling kecil maka ambillah nilai *hash* yang paling kanan untuk dijadikan *fingerprint*.

Hasil akhir dari pendeteksi plagiarisme pada gambar 8 mampu menghasilkan persentasi sampai 100% dengan menggunakan *Jaccard Similarity* dari dokumen atau sample pada Gambar 6 dengan nilai *n-gram* 3 dan nilai *w-gram* 5. Jika dibandingkan dengan metode *k-gram* yang digunakan pada [16] yang memperoleh kemiripan 83%, hasil ini cukup andal. Dengan demikian, metode *Jaccard Similarity* memiliki prospek untuk digunakan dalam deteksi plagiat.

4. Kesimpulan

Dari hasil penelitian di atas dapat dipahami *n-gram* sangat mempengaruhi hasil dari *similarity*, penggunaan *n-gram* yang tepat sangat diperlukan. Tingkat drajat kesamaan pada dua dokumen/sample akan menghasilkan nilai *similarity* yang berbeda-beda jika nilai *n-gram* dan *w-gram* yang dimasukan itu berbeda beda nilainya. Jika nilai yang dimasukan kecil maka hasil drajat kesamaan memiliki hasil tinggi. Jika nilai *n-gram* dan *w-gram* yang dimasukan semakin besar maka nilai *similarity* atau drajat kesamaan antar dokumen akan memiliki hasil rendah. Nilai *n-gram* yang besar dapat menghasilkan *similarity* yang tinggi jika nilai *w-gram* atau *window* yang dimasukan bernilai kecil dan begitu juga sebaliknya. *Jaccard similarity* yang digunakan untuk menghasilkan tingkat kesamaan antar dokumen juga cukup baik karna koefisien ini sederhana dengan mencari item yang sama dari dua dokumen kemudian dibagi dengan total item kedua dari penggabungan dua dokumen. Pendeteksi plagiarisme menggunakan metode *n-gram* dan *jaccard Similarity* terhadap algoritma *winnowing* cukup baik di gunakan untuk membandingkan kesamaan antara dua dokumen dan cukup baik digunakan untuk meminimalisir tindakan plagiarisme. Penelitian selanjutnya diharapkan dapat mendeteksi plagiarisme yang *copy-paste* dari karya ilmiah yang berbahasa inggris yang diterjemahkan ke dalam Bahasa Indonesia.

Referensi

- [1]. Adik W. Mencegah dan Menanggulangi Plagiarisme di Dunia Pendidikan. *Kemas J. Kesehat. Masy. Nas.* 2012; 6(5):195-200.
- [2]. Iif A. M, Sunardi, Anton Y. "Perancangan Aplikasi Deteksi Plagiarisme Karya Ilmiah Menggunakan Algoritma *Winnowing*," in Seminar Nasional Serba Informatika. Samarinda. 2017;
- [3]. Danang W Wicaksono, Mohammad Isa Irawan, dan Alvida Mustika Rukmi. Sistem Deteksi Kemiripan Antar Dokumen Teks Menggunakan Model Bayesian Pada Term Latent Semantic Analysis (LSA). *Jurnal Sains dan Seni ITS.* 2014; 2(3);A40-A45.
- [4]. Reynald K. W, Khafiizh H. Penerapan Algoritma *Winnowing* Untuk Mendeteksi Kemiripan Teks pada Tugas Akhir Manusia. *Techno.COM.* 2016; 15(4): 303-311.
- [5]. Komang R. Simple Query Suggestion untuk Pencarian Artikel menggunakan *Jaccard Similarity*. *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi.* 2017; 3(1): 30-34.
- [6]. Agus R, S. Implementasi Algoritma *Winnowing* untuk Deteksi Kemiripan Text. *Pelita Informatika Budi Darma.* 2015; IX(1).
- [7]. Eko N. Perancangan Sistem Deteksi Plagiarisme Dokumen Teks dengan Menggunakan Algoritma Rabin-Karp. Sekripsi. Malang. Universitas Braijaya. 2011;
- [8]. Aisy M.R, Dyah A.I, Faisal R. Rancang Bangun Game Fighting Pewayangan Bahuraksa Arena. *Jurnal Infrmatika Polinema.* 39-44.

- [9]. Erick A.L. Implementasi N-gram Technique dalam Deteksi Plagiarisme pada Tugas Mahasiswa. *Jurnal Tematika*. 2013; 1(2): 24-30.
- [10]. Muhammad F, Imam M.I.S, Dedy K. Sistem Rekomendasi Hasil Pencarian Artikel menggunakan Metode Jaccard's Coefficient. *Jurnal Transistor Elektro dan Informatika (TRANSSISTOR EI)*. 2017; 2(1):1-14.
- [11]. Sugiyanto, Bayu S, Aris S. Analisa Performa Metode Cosine dan Jaccard pada Pengujian Kesamaan Dokumen. *Jurnal Masyarakat Informatika*. 5(10):1-8.
- [12]. Anton Y, Abdul D.D. Implementation of Pattern Matching Algorithm for Portable Document Format. *Int. J. Adv. Comput. Sci. Appl.* 2017; 8(11):509-512.
- [13]. Ana K, Wayan S.W. *Perbandingan Pendekatan Deteksi Plagiarism Dokumen dalam Bahasa Inggris*. Proceeding, Seminar Ilmiah Nasional Komputer dan Intelijen (KOMMIT 2008). Depok. 2008; 284-291.
- [14]. Muhammad R. Rancang Bangun Aplikasi Pendeteksi Penjiplakan Dokumen menggunakan Algoritma Biword Winnowing. Sekripsi. Pekanbaru Riau. Universitas Islam Negeri SLTAN Syarif Khasin; 2013;
- [15]. Jody, Agung T. W, Anditya A. Analisis dan Implementasi Algoritma Winnowing dengan Synonym Recognition pada Deteksi Plagiarisme untuk Dokumen Teks Berbahasa Indonesia. *e-Proceeding Eng.* 2015; 2(3):7674-768.
- [16]. Mudafiq R Pratama, EB Cahyono, GI Marthasari, "Aplikasi Pendeteksi Duplikasi Dokumen Teks Bahasa Indonesia Menggunakan Algoritma Winnowing Dengan Metode K-Gram Dan Synonym Recognition." *Jurusan Teknik Informatika Universitas Muhammadiyah Malang*, 2012.