DETEKSI PLAGIASI DOKUMEN SKRIPSI MAHASISWA MENGGUNAKAN METODE N-GRAMS DAN WINNOWING

Fitri Ratna Ning Wulan
Fakultas Sains dan Teknologi, Program Studi Sistem Informasi
Universitas Islam Negeri Sunan Ampel Surabaya
Email: fitriratna96@gmail.com

Anang Kunaefi
Fakultas Sains dan Teknologi, Program Studi Sistem Informasi
Universitas Islam Negeri Sunan Ampel Surabaya
Email: akunaefi@uinsby.ac.id

Andhy Permadi
Fakultas Sains dan Teknologi, Program Studi Sistem Informasi
Universitas Islam Negeri Sunan Ampel Surabaya
Email: andhy@uinsby.ac.id

ABSTRAK

Salah satu tantangan dalam bidang akademik adalah mencegah maraknya aktivitas plagiarisme. Salah satu cara yang bisa dilakukan adalah dengan melakukan deteksi dini plagiasi terhadap karya mahasiswa terutama skripsi. Penerapan deteksi indikasi plagiarisme menggunakan metode *n-grams* dan *winnowing* merupakan tujuan dari penelitian ini, dan menemukan nominal *n* yang efektif. Kata yang ada dalam dokumen skripsi direpresentasikan dalam bentuk *hash*, lalu dilakukan seleksi menggunakan algoritma winnowing untuk menentukan *fingerprint* dari dokumen yang akan disimpan dalam basis data. Pengujian dilakukan dengan menggunakan sampel dokumen skripsi mahasiswa. Hasil akhir dari penelitian menunjukkan bahwa sistem dapat mendeteksi adanya plagiasi menurut kesamaan kata secara konsisten berdasarkan beberapa skenario pengujian yang dijalankan, nominsl *n* yang paling efektif dalam deteksi kesamaan kata yaitu *n* = 7 dengan prosentase 3,07% berdasarkan selisih antara pengujian menggunakan sistem dan pengujian manual.

Kata kunci: plagiarisme*; hash; n-grams; fingerprint; winnowing*.

ABSTRACT

One of the challenges in the academic field is to prevent the rampant plagiarism activity. One way that can be done is to make early detection of plagiarism on the work of students, especially thesis. Implementation of the detection for plagiarism using n-grams and winnowing method are the aims of this study, and finding the most effective n-grams value. The word in the thesis document is represented in the form of hash, then some of them were selected using the winnowing algorithm to determine the fingerprint of the document to be stored in the database. The test was carried out using a thesis document. The final results of this study indicate that the system is able to detect the presence of plagiarism based on similarity of words based on several test scenarios that are run, and the most effective n value used to detect document similarity is n = 7 with the percentage difference (accuracy) between system testing and testing the manual is 3.07%.

Keywords: *plagiarism; hash; n-grams; fingerprint; winnowing*.

1. PENDAHULUAN

Pada lingkup pendidikan plagiarisme marak dilakukan mahasiswa, diantaranya *copy paste* yang dilakukan saat pengerjaan tugas baik berasal dari internet maupun hasil karya orang lain. Hal tersebut juga memicu tindakan plagiarisme pada penulisan skripsi mahasiswa. Pendeteksian kesamaan kata pada dokumen merupakan salah satu langkah untuk mencegah plagiarisme, tetapi membutuhkan waktu lama ketika pengecekan dengan cara manual. Deteksi kesamaan kata dapat dilakukan menggunakan algoritma yang memperhatikan tinggi akurasi sistem saat mendeteksi (1). Kecepatan dan efisiensi waktu merupakan

kelebihan dari penggunaan sistem deteksi plagiarisme dalam membantu pengecekan dokumen skripsi. Deteksi kesamaan kata mempunyai beberapa kriteria, yaitu : tanda *spasi* dan huruf kapital tidak berpengaruh, menghilangkan kata yang tidak relevan, tidak terpengaruh pada letak kata (2).

Ada beberapa algoritma yang digunakan dalam mendeteksi kemiripan kata, diantaranya algoritma *n-grams* yang menggunakan nilai *n*. Karakter yang ada pada teks digunakan untuk mengetahui tingkat kemiripan dengan panjang sesuai *n*, posisi karakter yang berbeda tidak menjadi hambatan dalam mendeteksi kemiripan kata (3). Kesalahan penulisan yang tidak berpengaruh pada pendeteksian menjadi keunggulan dari algoritma *n-grams* (1). Hal yang berpengaruh pada proses deteksi kesamaan yaitu panjang *n*, sehingga belum didapatkan nilai *n* yang paling efektif dalam deteksi kesamaan kata.

Langkah selanjutnya setelah menggunakan algoritma *n-grams* dapat menggunakan algoritma *winnowing*, algoritma yang digunakan dalam mengecek kesamaan kata atau *document fingerprinting* (4). *Fingerprint* kata digunakan saat mengecek kesamaan kata. Penelitian sebelumnya yang dilakukan oleh Setiawan juga menggunakan algoritma *winnowing* dalam mendeteksi plagiarisme berdasarkan judul dan abstrak skripsi pada STMIK Budidarma (5).

Penelitian yang dilakukan oleh Dillak menggunakan metode *n-grams* dan *winnowing* untuk mendeteksi plagiarisme serta perhitungan kesamaan dengan rumus *cosine similarity* untuk mendapatkan prosentase kesamaan. Hasil penelitian tersebut sistem yang dibangun menggunakan bahasa pemrograman *Java* mampu mendeteksi kemiripan dengan cara membandingkan satu file uji dan satu file pembanding (1).

Penelitian yang dilakukan oleh Lisangan menggunakan algoritma *n-grams* dengan n = 3,4,5,6, dan 7 dalam mendeteksi plagiarisme pada tugas mahasiswa, pada penelitian tersebut membandingkan satu file uji dengan banyak file pembanding dengan menghitung prosentase kemiripan menggunakan perhitungan *Sorensen Dice Coefficient*. Pada penelitian tersebut proses deteksi dilakukan dimulai dari menghapus tanda baca dan spasi pada teks kemudian merubah teks menjadi rangkaian *n-grams* tanpa dilakukan proses *stemming*. Hasil penelitian tersebut yaitu mendapatkan rata-rata selisih relevansi setiap *n* (3).

Penelitian yang dilakukan oleh Purwitasari membahas tentang deteksi kalimat yang sama. Penelitian tersebut menggunakan algoritma *n-grams* dan *winnowing*. Hasil penelitian dengan sistem yang dibangun menggunakan bahasa pemrograman *Java*, sistem mampu mendeteksi persamaan kalimat dengan indikator nilai *n*, nilai b (bilangan prima), nilai w dan *threshold*. Pengujian dilakukan secara *one to one*, sehingga mempengaruhi waktu eksekusi (6).

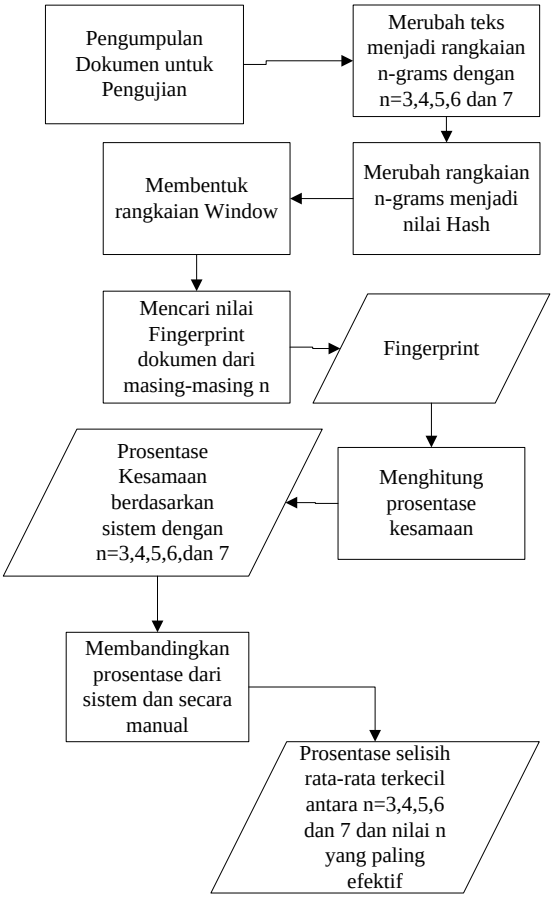
Penelitian yang dilakukan oleh Wibowo membahas tentang penerapan algoritma *winnowing* untuk mendeteksi kemiripan teks pada tugas akhir mahasiswa dengan objek dokumen tugas akhir mahasiswa Univeristas Dian Nuswantoro. Penelitian tersebut menggunakan metode *n-grams, winnowing* dan *jaccard coefficient* untuk mendapatkan prosentase kemiripan teks. Hasil penelitian tersebut yaitu sistem yang dibangun mampu mendeteksi kemiripan teks dengan nilai *n=6* dan nilai *n* dan *w* yang berbeda, menunjukkan bahwa semakin besar nilai *n* maka semakin rendah tingkat kemiripan teks, sebaliknya semakin kecil nilai *n* maka semakin tinggi tingkat kemiripan teks, tetapi hal tersebut tidak membuktikan bahwa hasil yang didapatkan akurat (4).

Penelitian selanjutnya yang dilakukan oleh Setiawan membahas tentang implementasi algoritma *winnowing* dalam mendeteksi kemiripan judul skripsi pada STMIK Budidarma. Penelitian tersebut menggunakan metode *n-grams, winnowing* dan *jaccard* coefficient untuk mendeteksi kemiripan judul dengan objek uji judul dan abstrak skripsi mahasiswa yang akan mengajukan judul skripsi. Pengujian pada penelitian tersebut membandingkan judul dan abstrak skripsi dengan judul dan abstrak skripsi yang sudah ada. Hasil penelitian tersebut didapatkan prosentase kemiripan judul dan abstrak skripsi dengan nilai *n* yang sudah ditentukan (5).

Adanya sistem yang menerapkan algoritma *n-grams* dan *winnowing* untuk pendeteksian kemiripan kata diharapkan bisa didapatkan nilai *n* paling efektif dalam deteksi kesamaan kata dalam dokumen.

2. METODOLOGI PENELITIAN

Penelitian ini mempunyai alur untuk menyelesaikan penelitian, seperti gambar 1 sebagai berikut:



Gambar 1. Tahapan Penelitian

2.1 Pengumpulan Dokumen Pengujian

Pengumpulan dokumen digunakan sebagai dokumen pengujian sistem yang dibuat. Dokumen pengujian sistem deteksi indikasi plagiarisme menggunakan dokumen skripsi mahasiswa sebanyak 37 dokumen.

2.2 Olah Data Dokumen

Olah data digunakan dalam mengolah permasalahan yang ada dan diimplementasikan pada sistem, Pengimplementasian proses pengolahan data menggunakan bahasa pemrograman PHP. Tahapan dalam deteksi indikasi plagiarisme pada penelitian ini sebagai berikut:

a. Preprocessing
   Proses ini diawali dengan merubah isi teks menjadi huruf kecil lalu proses *tokenizing* dengan merubah rangkaian kalimat menjadi rangkaian kata (7). Kemudian menghapus kata yang tidak relevan diantaranya dan, di, dan sebagainya atau disebut *stopword removal* (8). Selanjutnya *stemming* bahasa Indonesia untuk mendapatkan kata dasar. Teknik *stemming* yang digunakan yaitu teknik Nazief dan Adriani, dengan tingkat kebenaran 93% dibandingkan teknik *stemming* bahasa Indonesia dari Vega, Arifin&Setiono (9). Kata dasar tersebut didapatkan dengan menghilangkan imbuhan yang ada, kemudian digunakan dalam membentuk rangkaian *n-grams*.
b. Analisa Algoritma N-Grams
   Algoritma *n*-grams merupakan salah satu algoritma yang digunakan untuk memisahkan teks menjadi rangkaian kata dengan panjang atau *n* yang ditentukan (10). Panjang *n* dimulai dari *n=1* sampai tak terhinga. Contoh rangkaian *n-grams* dari kata "makan minum", dengan *n=4* akan menjadi "maka, akan, kanm, anmi, nmin, minu, inum".
   Rangkaian *n-grams* dibentuk dengan diawali dari menghilangkan *spasi* pada rangkaian kata dasar, kemudian pembentukan *n-grams* menurut panjang *n*. Sistem pada penelitian ini menggunakan *n=3, n=4, n=5, n=6, n=7* untuk mendapatkan nominal *n* yang paling efektif digunakan dalam pengecekan indikasi plagiarisme.

c. Perhitungan *Hashing*
Hashing merupakan suatu metode yang dilkukan untuk merubah nilai *n-grams* menjadi nilai angka yang digunakan dalam algoritma *winnowing.* Rumus yang digunakan dalam perhitungan *hashing* pada penelitian ini adalah *Rolling Hash*. Berikut rumus dari *rolling hash* (5) seperti rumus 1 berikut:

$$H_{(C_{1\ldots}C_n)} = C_1 * b^{(n-1)} + C_2 * b^{(n-2)} + \cdots + C_{(n-1)} * b^{(n)} + C_n \quad (1)$$

Berdasarkan rumus 1, diketahui c yaitu bilangan ASCII dari karakter, b = bilangan prima yang ditentukan, dan n = panjang *n-grams.* Untuk rumus kedua digunakan pada *n-grams* kedua dan seterusnya, dikarenakan tidak memerlukan penghitungan karakter pertama. Berikut rumus kedua *roliing hash*, seperti pada rumus 2 berikut:

$$H_{(C_{2\ldots}C_{n+1})} = (H_{(C_{1\ldots}C_n)} - C_1 * b^{(n-1)}) * b + C_{(n+1)} \quad (2)$$

Nilai bilangan prima dalam rumus *rolling hash* yaitu b=3, mengambil nominal ASCII dari karakter (c). Contoh hasil nilai *hash* dari rangkaian *n-grams* seperti berikut: [4234, 3983, 4201, 4041, 4376, 4335, 4285]

d. Menentukan Fingerprint
Proses menentukan *fingerprint* menggunakan algoritma *winnowing* diawali dengan merubah nilai *hash* menjadi rangkaian *hash* berdasarkan nilai *window (w)*. Selanjutnya dipilih nilai *hash* terkecil dari setiap *window*. Selanjutnya dilakukan pengecekan apabila ada nilai *hash* yang sama maka hanya di ambil salah satu. Penelitian ini menggunakan nilai *w = 5*, contoh rangkaian *window* seperti berikut :

[4234 3983 4201 4041 4376] [3983 4201 4041 4376 4335] [4201 4041 4376 4335 4285]

Kemudian mencari nilai *fingerprint*, nilai *fingerprint* dokumen digunakan untuk membandingkan dengan dokumen lain sehingga didapatkan hasil kemiripan dokumen. *Hash* terkecil dari setiap *window* diambil sebagai *fingerprint* dokumen. Contoh mendapatkan *fingerprint* dalam *window*, seperti berikut :

[4234 **3983** 4201 4041 4376] [3983 4201 4041 4376 4335] [4201 **4041** 4376 4335 4285]

Berdasarkan hasil diatas, diketahui nilai *fingerprint* yaitu :[3983 4041]

e. Menghitung Kesamaan Kata
Jaccard Coefficient merupakan salah satu metode yang digunakan untuk menghitung kesamaan data (11). Berikut rumus dari metode *jaccard coefficient*:

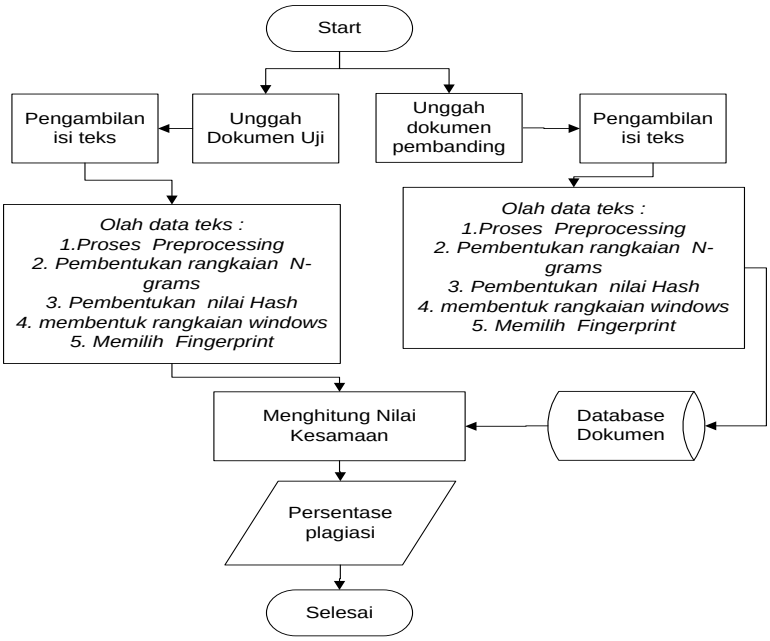
$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \quad (3)$$

Berdasarkan pada rumus 3 dapat dijelaskan bahwa $S_1 \cap S_2$ merupakan jumlah nilai yang sama antara S1 dan S2. Sedangkan $S_1 \cup S_2$ merupakan gabungan keduanya. Hasil dari rumus *Jaccard Coefficient* didapatkan prosentase kesamaan kata.

3. HASIL DAN PEMBAHASAN

3.1 Arsitektur Sistem

Secara umum arsitektur sistem dalam penelitian ini terdiri dari beberapa proses, seperti yang sudah digambarkan pada gambar 2 berikut:



Gambar 2. Arsitektur Sistem

Berdasarkan alur proses sistem pada Gambar 2 diawali dengan unggah file dokumen untuk pembanding lalu didapatkan isi teks, proses preprocessing yaitu dilakukan tokenizing (mengubah kalimat menjadi rangkaian kata), menghilangkan stopword atau menghapus kata tidak relevan kemudian stemming untuk mendapatkan kata dasar, selanjutnya pembentukan n-grams dari hasil preprocessing, mendapatkan nilai hash dari hasil proses sebelumnya, membentuk windowing dan mencari nilai fingerprint lalu disimpan dalam basis data dan digunakan untuk membandingkan dengan hasil dokumen uji.

Langkah selanjutnya mengunggah dokumen uji, dan melakukan proses preprocessing sampai mendapatkan nilai fingerprint. kemudian dilakukan pengecekan dengan membandingkan fingerprint dokumen pembanding dan uji untuk mendapatkan prosentase kesamaan kata antar dokumen.

3.2 Kriteria Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan pada penelitian ini dalam mengembangkan sistem, seperti berikut :

a. XAMPP 5.6.11 2013
b. Notepad++
c. HeidiSQL
d. Google Chrome

3.3 Desain Basis Data (Database)

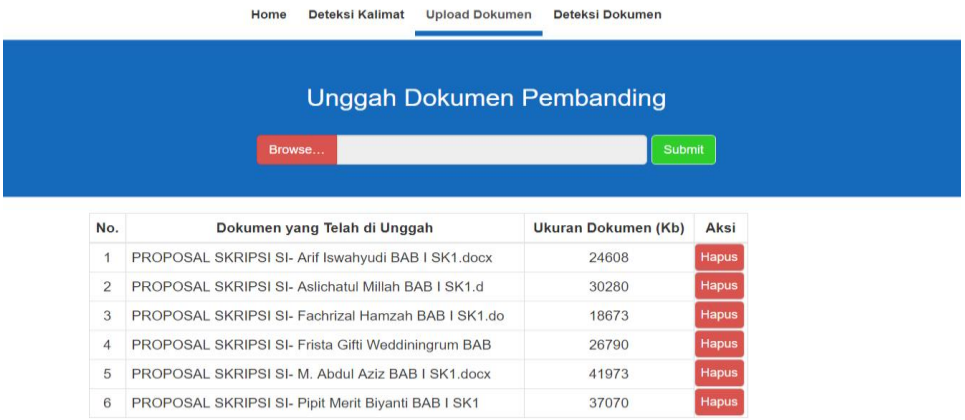
Database digunakan untuk menyimpan hasil *fingerprint* dokumen pembanding pada sistem ini. Basis data yang digunakan yaitu DBMS *mysql*. Nilai *fingerprint* yang disimpan digunakan dalam perulangan pengecekan. Berikut atribut *database* dalam sistem pada tabel berikut:

Tabel 1. Atribut *database*

Nomor	*Atribut*	*Data Type*
1	id_dokumen (Primary Key)	int(11) Auto_Increment
2	nama_dokumen	varchar(50)
3	tipe_dokumen	varchar(50)
4	ukuran_dokumen	11
5	fingerprint_dokumen	varchar(10000)
6	tanggal_unggah	timestamp

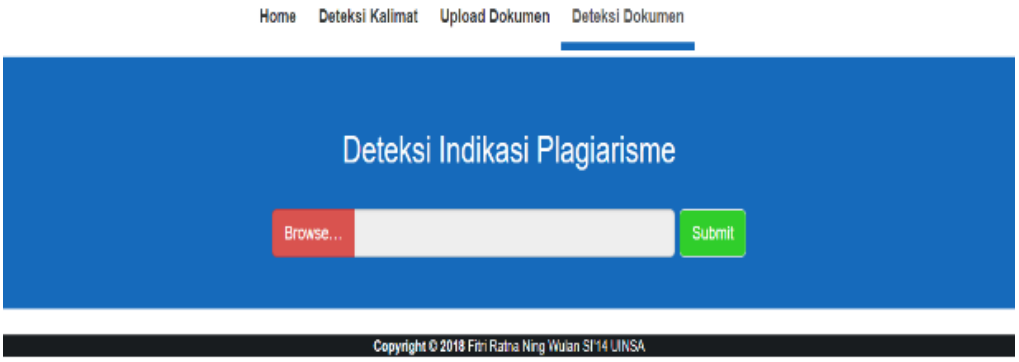
3.4 Tampilan User Interfaces

Sistem pada penelitian ini menggunakan tampilan *interface* sehingga memudahkan saat menggunakan fungsi sistem. Sistem ini mempunyai beberapa menu, yang pertama adalah halaman menu beranda. Menu kedua yaitu menu Unggah Dokumen yang terdapat form unggah file dokumen untuk pembanding. Hasil pengunggahan yaitu didapatkan nilai *fingerprint* dan disimpan di *database*. Berikut tampilan menu Unggah Dokumen seperti Gambar 3 berikut:



Gambar 3. Halaman Upload Dokumen Pembanding

Menu ketiga yaitu menu Deteksi Dokumen, menampilkan form untuk unggah dokumen uji, untuk mendapatkan prosentase kesamaan kata. Berikut tampilan menu deteksi dokumen seperti Gambar 4 berikut :



Gambar 4. Halaman Deteksi Kesamaan Kata

Selanjutnya pada halaman deteksi kesamaan kata sistem akan memproses pengecekan untuk mendapatkan hasil prosentase tingkat kesamaan antar dokumen.

3.5 Pengujian

Pengujian pada penelitian ini menggunakan skenario yang sebelumnya sudah dibuat. Berikut skenario digunakan diantaranya:

a. Skenario 1 (pengujian untuk dokumen yang tidak sama)
   Skenario 1 diawali dengan mengambil dokumen uji dan pembanding. Proses pengujian dengan mengambil 1 paragraf teks dokumen pembanding dan dimasukkan ke dokumen uji, untuk membuktikan sistem dapat melakukan deteksi plagiasi berdasarkan kondisi memiliki sedikit kesamaan antara 2 dokumen. Berikut tabel pengujian secara manual pada tabel 2:

Tabel 2. Pengujian Manual Skenario 1

Dokumen ke-	*kata*	*Persentase Manual*
1	69 dari 897	7,69%
2	62 dari 897	6,91%
3	93 dari 897	10,37%
4	52 dari 897	5,80%
5	53 dari 897	5,91%
6	111 dari 897	12,37%
7	77 dari 897	8,58%
8	56 dari 897	6,24%
9	53 dari 897	5,91%
10	109 dari 897	12,15%
11	42 dari 897	4,68%
12	35 dari 897	3,90%

Dokumen uji untuk skenario 1 menggunakan 1 dokumen, untuk dokumen pembanding menggunakan 12 dokumen. Berikut hasil pengujian skenario 1 untuk *n*=3, *n*=4, *n*=5, *n*=6 dan *n* =7 seperti pada tabel berikut:

Tabel 3. Skenario 1 n=3

Dokumen ke-	*Prosentase Sistem*	*Prosentase Manual*	*Selisih*
1	43,79%	7,69%	36,10%
2	46,98%	6,91%	40,07%
3	54,92%	10,37%	44,55%
4	53,47%	5,80%	47,67%
5	65,40%	5,91%	59,49%
6	54,80%	12,37%	42,43%
7	47,77%	8,58%	39,19%
8	55,20%	6,24%	48,96%
9	61,25%	5,91%	55,34%
10	58,68%	12,15%	46,53%
11	55,65%	4,68%	50,97%
12	54,06%	3,90%	50,16%
Total			561,46%
Rata-Rata Selisih			**46,79%**

Tabel 3 menggambarkan hasil pengujian pada skenario 1 dengan *n*=3, ketika semua selisih dijumlah didapatkan hasil 561,46% lalu dibagi dengan jumlah dokumen yaitu 12, maka didapatkan rata-rata selisih 46,79%. Untuk *n*=4 dijelaskan pada tabel 4 berikut:

Tabel 4. Skenario 1 n=4

Dokumen ke-	*Prosentase Sistem*	*Prosentase Manual*	*Selisih*
1	40,29%	7,69%	32,60%
2	45,50%	6,91%	38,59%
3	49,80%	10,37%	39,43%
4	48,53%	5,80%	42,73%
5	53,75%	5,91%	47,84%
6	49,85%	12,37%	37,48%
7	49,04%	8,58%	40,46%
8	51,29%	6,24%	45,05%
9	53,83%	5,91%	47,92%
10	57,14%	12,15%	44,99%
11	53,09%	4,68%	48,41%
12	49,17%	3,90%	45,27%
Total			510,77%
Rata-Rata Selisih			**42,56%**

Tabel 4 menggambarkan hasil pengujian pada skenario 1 dengan *n*=4, ketika semua selisih dijumlah didapatkan hasil 510,77% lalu dibagi dengan jumlah dokumen yaitu 12, maka didapatkan rata-rata selisih 42,56%. Untuk *n*=5 dijelaskan pada tabel 5 berikut:

Tabel 5. Skenario 1 n=5

Dokumen ke-	*Prosentase Sistem*	*Prosentase Manual*	*Selisih*
1	30,22%	7,69%	22,53%
2	32,30%	6,91%	25,39%
3	35,55%	10,37%	25,18%
4	31,36%	5,80%	25,56%
5	34,10%	5,91%	28,19%
6	35,40%	12,37%	23,03%
7	38,45%	8,58%	29,87%
8	35,27%	6,24%	29,03%
9	35,18%	5,91%	29,27%
10	37,17%	12,15%	25,02%
11	38,62%	4,68%	33,94%
12	34,21%	3,90%	30,31%
Total			327,32%
Rata-Rata Selisih			**27,28%**

Tabel 5 menggambarkan hasil pengujian pada skenario 1 dengan *n*=5, ketika semua selisih dijumlah didapatkan hasil 327,32% lalu dibagi dengan jumlah dokumen yaitu 12, maka didapatkan rata-rata selisih 27,28%. Untuk *n*=6 dijelaskan pada tabel 6 berikut:

Tabel 6. Skenario 1 n=6

Dokumen ke-	*Prosentase Sistem*	*Prosentase Manual*	*Selisih*
1	16,39%	7,69%	8,70%
2	17,91%	6,91%	11,00%
3	18,77%	10,37%	8,40%
4	17,36%	5,80%	11,56%
5	17,03%	5,91%	11,12%
6	20,10%	12,37%	7,73%
7	20,51%	8,58%	11,93%
8	17,88%	6,24%	11,64%
9	19,49%	5,91%	13,58%
10	21,46%	12,15%	9,31%
11	19,51%	4,68%	14,83%
12	16,52%	3,90%	12,62%
Total			132,42%
Rata-Rata Selisih			**11,04%**

Tabel 6 menggambarkan hasil pengujian pada skenario 1 dengan *n*=6, ketika semua selisih dijumlah didapatkan hasil 132,42% lalu dibagi dengan jumlah dokumen yaitu 12, maka didapatkan rata-rata selisih 11,04%. Untuk *n*=7 dijelaskan pada tabel 7 berikut:

Tabel 7. Skenario 1 n=7

Dokumen ke-	*Prosentase Sistem*	*Prosentase Manual*	*Selisih*
1	9,31%	7,69%	1,62%
2	8,54%	6,91%	1,63%
3	12,17%	10,37%	1,80%
4	9,79%	5,80%	3,99%
5	10,10%	5,91%	4,19%
6	13,40%	12,37%	1,03%
7	11,70%	8,58%	3,12%
8	8,79%	6,24%	2,55%
9	10,33%	5,91%	4,42%
10	12,83%	12,15%	0,68%
11	10,58%	4,68%	5,90%
12	9,86%	3,90%	5,96%
Total			36,89%
Selisih			**3,07%**

Tabel 7 menggambarkan hasil pengujian pada skenario 1 dengan *n*=7, ketika semua selisih dijumlah didapatkan hasil 36,89% lalu dibagi dengan jumlah dokumen yaitu 12, maka didapatkan rata-rata selisih 3,07%.

Setelah dilakukan pengujian baik secara sistem maupun manual pada dokumen seperti pada tabel pengujian dengan *n*=3 - *n*=7 diatas, didapatkan hasil nilai *n* terbaik dalam mendeteksi kesamaan kata antar dokumen yaitu *n*=7 dengan prosentase rata-rata selisih 3,07%.

b. Skenario 2 (Pengujian untuk dokumen 100% sama)

Skenario 2 diawali dengan mengambil dokumen uji dan pembanding. Proses pengujian dengan menyalin semua isi dari salah satu dokumen pembanding menjadi dokumen uji.

Dokumen *testing* untuk skenario 2 menggunakan 1 dokumen skripsi, untuk dokumen pembanding menggunakan 11 dokumen. Setelah dilakukan pengujian baik secara sistem maupun manual pada dokumen skripsi, didapatkan hasil bahwa semua pengujian dari nilai *n*=3 sampai *n*=7 mampu mendeteksi kesamaan 100%. Seperti dijelaskan pada gambar 8 berikut:

Tabel 8. Pengujian skenario 2 n=3, n=4, n=5, n=6 dan n=7

Dokumen ke-	*n=3*	*n=4*	*n=5*	*n=6*	*n=7*
1	60,67%	45,27%	23,82%	11,59%	4,62%
2	58,87%	42,86%	24,19%	10,52%	5,01%
3	61,60%	39,78%	23,77%	11,02%	4,88%
4	54,20%	42,63%	25,30%	12,11%	5,88%
5	**100,00%**	**100,00%**	**100,00%**	**100,00%**	**100,00%**
6	54,60%	42,98%	25,21%	11,51%	4,92%
7	57,34%	44,16%	25,72%	11,80%	5,65%
8	56,91%	44,77%	24,07%	11,14%	5,22%
9	40,55%	36,10%	20,87%	10,43%	5,30%
10	58,22%	43,94%	24,40%	10,68%	4,54%
11	52,04%	37,13%	20,17%	9,50%	4,51%

Berdasarkan pemaparan pada tabel 8, dapat disimpulkan bahwa sistem mampu mendeteksi 100% kesamaan dengan *n*=3 sampai *n*=7.

c. Skenario 3 (Pengujian untuk dokumen sebagian sama)

Pengujian skenario 3 bertujuan untuk membuktikan sistem dapat melakukan deteksi indikasi plagiasi dengan sebagian sama. Pengujian skenario 3 diawali dengan mengambil beberapa paragraf teks dari dokumen pembanding dan meletakkannya di dokumen uji. Berikut tabel pengujian secara manual pada tabel 9:

Tabel 9. Pengujian Manual skenario 3

Dokumen Ke-	*kata*	*Persentase Manual*
1	105 dr 1653	6,35%
2	69 dr 1653	4,17%
3	298 dr 1653	18,03%
4	251 dr 1653	15,18%
5	179 dr 1653	10,83%
6	84 dr 1653	5,08%
7	263 dr 1653	15,91%
8	95 dr 1653	5,74%
9	40 dr 1653	2,42%
10	31 dr 1653	1,88%
11	40 dr 1653	2,42%
12	60 dr 1653	3,63%

Dokumen uji untuk skenario 3 menggunakan 1 dokumen, untuk dokumen pembanding menggunakan 12 dokumen. Berikut hasil pengujian skenario 3 *n*=3 – *n*=7 secara sistem dan manual dan hasil selisih keduanya untuk menentukan nominal *n* paling efektif yang digunakan dalam mendeteki plagiasi seperti pada tabel berikut:

Tabel 10. Pengujian Skenario 3 n=3

Dokumen Ke-	*Prosentase Sistem*	*Prosentase Manual*	*Selisih*
1	52,55%	6,35%	46,20%
2	55,88%	4,17%	51,71%
3	62,95%	18,03%	44,92%
4	52,48%	15,18%	37,30%
5	60,12%	10,83%	49,29%
6	57,10%	5,08%	52,02%
7	59,93%	15,91%	44,02%
8	52,95%	5,74%	47,21%
9	54,27%	2,42%	51,85%
10	53,06%	1,88%	51,18%
11	53,99%	2,42%	51,57%
12	57,60%	3,63%	53,97%
	Total		**581,24%**
	Selisih		**48,44%**

Tabel 10 menggambarkan hasil pengujian pada skenario 3 dengan *n*=3, ketika semua selisih dijumlah didapatkan hasil 581,24% lalu dibagi dengan jumlah dokumen yaitu 12, maka didapatkan rata-rata selisih 48,44%. Untuk *n*=4 dijelaskan pada tabel 11 berikut:

Tabel 11. Pengujian Skenario 3 n=4

Dokumen Ke-	*Prosentase Sistem*	*Prosentase Manual*	*Selisih*
1	40,09%	6,35%	33,74%
2	46,86%	4,17%	42,69%
3	59,05%	18,03%	41,02%
4	45,65%	15,18%	30,47%
5	55,54%	10,83%	44,71%
6	50,54%	5,08%	45,46%
7	51,90%	15,91%	35,99%
8	52,18%	5,74%	46,44%
9	52,65%	2,42%	50,23%
10	44,02%	1,88%	42,14%
11	51,19%	2,42%	48,77%
12	50,76%	3,63%	47,13%
	Total		508,79%
	Selisih		**42,40%**

Tabel 11 menggambarkan hasil pengujian pada skenario 3 dengan *n*=4, ketika semua selisih dijumlah didapatkan hasil 508,79% lalu dibagi dengan jumlah dokumen yaitu 12, maka didapatkan rata-rata selisih 42,40%. Untuk *n*=5 dijelaskan pada tabel 12 berikut:

Tabel 12. Pengujian Skenario 3 n=5

Dokumen Ke-	*Prosentase Sistem*	*Prosentase Manual*	*Selisih*
1	28,07%	6,35%	21,72%
2	33,33%	4,17%	29,16%
3	49,43%	18,03%	31,40%
4	34,16%	15,18%	18,98%
5	42,86%	10,83%	32,03%
6	36,73%	5,08%	31,65%
7	38,12%	15,91%	22,21%
8	40,67%	5,74%	34,93%
9	39,29%	2,42%	36,87%
10	32,72%	1,88%	30,84%
11	37,14%	2,42%	34,72%
12	38,51%	3,63%	34,88%
	Total		359,39%
	Selisih		**29,95%**

Tabel 12 menggambarkan hasil pengujian pada skenario 3 dengan *n*=5, ketika semua selisih dijumlah didapatkan hasil 359,39% lalu dibagi dengan jumlah dokumen yaitu 12, maka didapatkan rata-rata selisih 29,95%. Untuk *n*=6 dijelaskan pada tabel 13 berikut:

Tabel 13. Pengujian Skenario 3 n=6

Dokumen Ke-	*Prosentase Sistem*	*Prosentase Manual*	*Selisih*
1	16,35%	6,35%	10,00%
2	16,24%	4,17%	12,07%
3	32,62%	18,03%	14,59%
4	23,85%	15,18%	8,67%
5	27,38%	10,83%	16,55%
6	22,54%	5,08%	17,46%
7	26,14%	15,91%	10,23%
8	22,10%	5,74%	16,36%
9	22,18%	2,42%	19,76%
10	17,72%	1,88%	15,84%
11	20,46%	2,42%	18,04%
12	19,89%	3,63%	16,26%
	Total		175,83%
	Selisih		**14,65%**

Tabel 13 menggambarkan hasil pengujian pada skenario 3 dengan *n*=6, ketika semua selisih dijumlah didapatkan hasil 175,83% lalu dibagi dengan jumlah dokumen yaitu 12, maka didapatkan rata-rata selisih 14,65%. Untuk *n*=7 dijelaskan pada tabel 14 berikut:

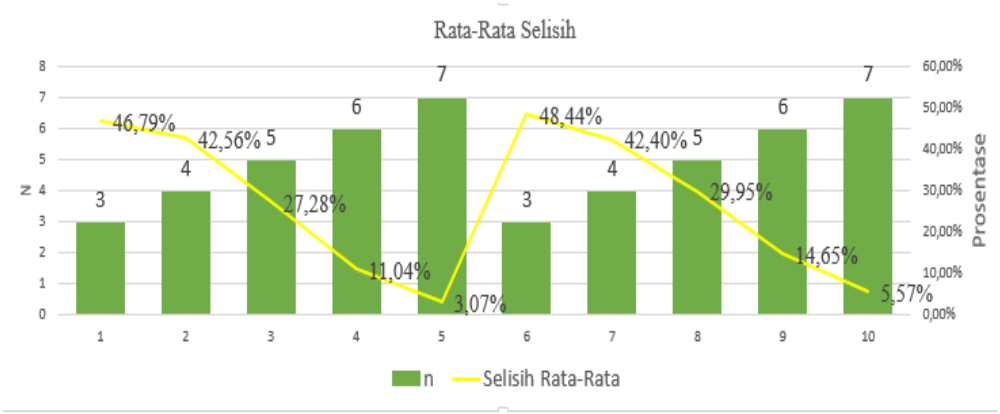
Tabel 14. Pengujian Skenario 3 n=7

Dokumen Ke-	*Prosentase Sistem*	*Prosentase Manual*	*Selisih*
1	10,57%	6,35%	4,22%
2	8,71%	4,17%	4,54%
3	24,48%	18,03%	6,45%
4	17,43%	15,18%	2,25%
5	17,59%	10,83%	6,76%
6	13,07%	5,08%	7,99%
7	19,70%	15,91%	3,79%
8	11,71%	5,74%	5,97%
9	9,76%	2,42%	7,34%
10	8,37%	1,88%	6,49%
11	9,25%	2,42%	6,83%
12	7,87%	3,63%	4,24%
	Total		66,87%
	Selisih		**5,57%**

Tabel 14 menggambarkan hasil pengujian pada skenario 3 dengan *n*=7, ketika semua selisih dijumlah didapatkan hasil 66,87% lalu dibagi dengan jumlah dokumen yaitu 12, maka didapatkan rata-rata selisih 5,57%.

Setelah dilakukan pengujian skenario 3 dengan *n*=3 sampai *n*=7 pada tabel 10 - tabel 14, didapatkan hasil nilai *n* terbaik yang digunakan dalam mendeteksi kesamaan kata antar dokumen adalah *n* = 7 dan prosentase rata-rata selisih (akurasi) 5,57%.

Berdasarkan pengujian yang telah dilakukan, diambil prosentase selisih rata-rata terkecil dari setiap skenario dan didapatkan nilai *n* berdasarkan nilai selisih rata-rata terkecil, nilai *n* tersebut merupakan yang paling efektif ketika digunakan dalam proses deteksi plagiasi. Pada gambar grafik berikut menunjukkan prosentase rata-rata terkecil dari setiap skenario, seperti berikut :



Gambar 5. Grafik Hasil Rata-Rata Selisih

Grafik pada Gambar 5 menggambarkan *n* = 7 merupakam nilai yang paling efektif pada skenario 1 dan 3. Untuk rata-rata selisih pada skenario 1 dengan prosentase 3,07% dan skenario 3 dengan rata-rata selisih prosentase 5,57%. Berdasarkan hasil pengujian dapat disimpulkan semakin kecil nilai *n* maka semakin tinggi prosentase kesamaan, sebaliknya semakin besar nilai *n* maka semakin rendah prosentase kesamaan. Tingkat prosentase dan waktu eksekusi juga dipengaruhi oleh jumlah kata.

4. KESIMPULAN

Berdasarkan hasil pembahasan pada bab-bab sebelumnya maka dapat diambil kesimpulan bahwa:

a) Pengolahan teks saat deteksi dapat menggunakan metode *n-grams* dan *winnowing*.
b) Berdasarkan pengujian membuktikan bahwa *n* = 7 yang paling efektif digunakan dalam pendeteksian plagiasi menurut kesamaan setiap kata..
c) Hasil pengujian dapat membuktikan tingkat prosentase dan waktu eksekusi dipengaruhi oleh jumlah kata. Besar *n* juga berpengaruh pada prosentase kesamaan.

DAFTAR PUSTAKA

[1] Dillak RY, Laumal F, Kadja LJ, S S. Sistem Deteksi Dini Plagiarisme Tugas Akhir Mahasiswa Menggunakan Algoritma N-Grams dan Winnowing. J Ilm. 2016;2.
[2] Schleimer S, Wilkerson DS, Aiken A, Berkeley UC. Winnowing : Local Algorithms for Document Fingerprinting.
[3] Lisangan EA. Implementasi n-gram Technique dalam Deteksi Plagiarisme pada Tugas Mahasiswa. 2015;(SEPTEMBER 2013).
[4] Wibowo RK, Hastuti K. Penerapan Algoritma Winnowing Untuk. TechnoCOM. 2016;15(4):303–11.
[5] Setiawan A. Implementasi Algoritma Winnowing Untuk. Pelita Inform Budi Darma. 2017;(1011340):134–8.
[6] Purwitasari D, Kusmawan PY, dkk. Deteksi Keberadaan Kalimat Sama Sebagai Indikasi Penjiplakan Dengan Algoritma Hashing Berbasis N-Gram. 2011;6(1):37–44.
[7] Kao A, Poteet SR. Natural language processing and text mining. Natural Language Processing and Text Mining. 2007. 1-265 p.
[8] Indurkhya N, Damerau FJ. Handbook of Natural Language Processing Second Edition. New York: CRC Press; 2010.
[9] Asian J, Williams HE, Tahaghoghi SMM. Stemming Indonesian. 2005;38.
[10] Hornik K, Mair P, Rauch J, Geiger W, Buchta C. The textcat Package for n -Gram Based Text. 2013;52(6).
[11] Aggarwal CC. Data Mining: The Textbook. New York: Springer; 2015.