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

Wahyu Hidayat*[1], Ema Utami [2], Anggit Dwi Hartanto [3]
[1,2,3]Magister Teknik Informatika Universitas Amikom Yogyakarta
E-mail: ***[1]wahyu.1181@students.amikom.ac.id, [2]ema.u@amikom.ac.id, [3]anggit@amikom.ac.id**

Abstrak

Pengidentifikasian terkait plagiarisme terhadap dokumen berbahasa Indonesia telah dilakukan di penelitian terkait, untuk pendeteksi tingkat kesamaan dokumen. Dalam penelitian tersebut telah digunakan algoritma pendeteksi kesamaan dokumen dengan metode fingerprint sseperti Algoritma Winnowing. Algoritma Winnowing memiliki perbedaan pada penggunaan parameter seperti ada yang menggunakan k-gram dan n-gram. Dari perbedaan parameter tersebut dilakukan penelitian performa dari perbandingan penggunaan parameter yang berbeda pada pemotongan string pada tahap algoritma Winnowing sehingga dapat diketahui parameter yang mempunyai tingkat performa yang paling baik. Hasil penelitian pada k-gram memiliki tingkat nilai similarity yang tinggi namun ketika nilai jumlah k semakin besar akan mengurangi tingkat nilai similarit dengan rata-rata hasil pada k = 2 sebesar 0.5299, k = 3 sebesar 0.1689, k = 5 sebesar 0.0283 dan k = 7 sebesar 0.0095. Penerapan pemotongan string n-gram pada unigram memiliki rata-rata tingkat similarity sebesar 0.0683, bigram 0.003, pada trigram dan four-gram sebesar 0.000. Pada perbandingan kecepatan pemrosesan waktu k-gram dan n-gram tidak terlihat perbedaan yang signifikan dan keduanya mendominasi selama 6 detik.

Kata Kunci—*Algoritma Winnowing, Jaccard Similarity, Fingerprint, K-gram, N-gram*

Abstract

Identification related to plagiarism of Indonesian language documents has been carried out in related research, such as for the purpose of detecting the level of similarity documents. In this research, algorithm similarity detection algorithms have been used, especially with the fingerprint method wich Winnowing algorithm. Winnowing algorithm using parameters such as those using k-gram and n-gram. From these different parameters, a study of the performance of the comparison the use of different parameters in the string cutting at the Winnowing algorithm stage can be found out which parameter has the best level of performance. The results of research on k-gram have a high level of similarity value, but when the value of the number of k gets bigger it will reduce the level of similarity values with an average result at k = 2 of 0.5299, k = 3 of 0.1689, k = 5 of 0.0283 and k = 7 in the amount of 0.0095. The application of cutting n-gram strings on unigram has an average similarity level of 0.0683, bigram 0.003, on trigrams and four-grams of 0.000. In the comparison of the processing speed of k-gram and n-gram time, there was no significant difference, and both dominated for 6 seconds.

Keywords*— Winnowing algorithm, Jaccard Similarity, Fingerprint, K-gram, N-gram*

1. PENDAHULUAN

Plagiarisme yang dilakukan dalam penulisan bukan merupakan hal baru yang terjadi khususnya pada dokumen berbahasa Indonesia [1]. Penulis sering mengutarakan sebuah kata yang mirip dengan apa yang telah di tulis oleh orang lain, meskipun telah menambah kata sambung atau merubah letak penulisan tetap di identifikasi sebagai plagiarisme karena penulis tidak memparafrasa ulang terkait dengan makna yang sama namun dengan penulisan yang berbeda. Permasalahan tingkat kepentingan plagiarisme tergantung terhadap jenis dokumen yang diteliti, namun pada dasarnya melakukan plagiarisme merupakan pelanggaran etika. Selain itu melakukan plagiarisme merupakan hal yang melanggar hukum terkait dengan kerugian yang didampakkannya, karena perbuatan tersebut merupakan perbuatan yang tidak terpuji [2].

Tentang penambahan kata sambung dan kata imbuhan pada dokumen yang mirip jika di lihat secara manual memiliki penempatan kata yang berbeda seperti pembuatan kata tersebut ke kata langsung dan tidak langsung serta pengimbuhan kata sehingga seolah-olah penulisan dokumen tersebut berbeda. Namun hal tersebut dapat diteliti dengan menggunakan *text preprocessing* karena terdapat proses yang dapat menghilangkan kata sambung serta dapat merubah kata berimbuhan menjadi kata baku dalam prosesnya. Maka dari itu *text preprocessing* tersebut dapat mentransformasikan sebuah dokumen yang semulanya memiliki susunan kata menjadi sekumpulan kata tanpa mengurangi makna [3]. Algoritma Nazief & Adriani merupakan algoritma stemming yang digunakan untuk mengolah kata berimbuhan menjadi kata baku berdasarkan aturan pada algoritma tersebut [4]. Penerapan algoritma Nazief & Adriani bertujuan untuk mengidentifikasi kata dasar berdasarkan aturan [5], aturan tersebut didasarkan pada aturan morfologi bahasa Indonesia yang mengidentifikasi kata berimbuhan kedalam aturannya kemudian dicocokkan dengan kamus Bahasa Indonesia [6].

Algoritma Winnowing merupakan algoritma pendeteksi tingkat kesamaan dokumen, salah satu langkah dari algoritma winnowing yaitu mentransformasikan sebuah kata menjadi sebanyak ukuran k-gram ataupun n-gram, jika menggunakan k-gram didasarkan pada sekumpulan per karakter yang dipotong berdasarkan jumlah *k* [7] namun terdapat pemotongan berdasarkan pemenggalan kata dengan metode memotong berdasarkan unigram, bigram, trigram, four-gram [8]. Setelah pemotongan berdasarkan jumlah k-gram ataupun n-gram selanjutnya pembuatan hashing pada setiap pemotongan tersebut menggunakan rolling hash [9]. Dari hasil rolling hash tersebut di kelompokkan menjadi beberapa *window* dengan anggota *window* yang ditentukan, setelah dikelompokkan maka akan diidentifikasi nilai terkecil dari setiap *window* yang akan menjadi nilai *fingerprint*. Berdasarkan nilai terkecil atau *fingerprint* pada setiap *window* kemudian dicocokkan antar *fingerprint* dokumen lainnya, dari jumlah *fingerprint* yang sama dan akumulasi jumlah *fingerprint* di kedua dokumen yang dibandingkan akan di hitung dengan persamaan Jaccard Similarity [10] sehingga akan ditemukan nilai kesamaan antar dokumen tersebut.

Dalam menyelesaikan permasalahan pendeteksi kesamaan dokumen [11] melakukan penelitian tentang mengukur tingkat kemiripan dokumen pada pengajuan proposal menggunakan algoritma Winnowing dengan biword. Pemakaian biword didasarkan pada pemotongan kata yang di potong berdasarkan per dua kata yang mempunyai tujuan menjaga arti kata pada dokumen yang di proses. Pada penelitian tersebut menunjukkan hasil basis dan *window* terbaik yaitu basis 4 dan *window* 2 pada dataset penelitian sedangkan pada dataset pengabdian menghasilkan basis 2 dan *window* 1. Dalam penelitian yang dilakukan terdapat perbandingan dari hasil checker X yang dibandingkan dengan penerapan algoritma biword Winnowing yang menunjukkan hasil presentase rata-rata selisih 0.635% pada dataset penelitian sedangkan untuk dataset pengabdian memiliki rata-rata 0.586%.

Penelitian terkait penggunaan biword pada penerapan algoritma Winnowing [12] yang membandingkan penerapan biword dan triword serta membandingkan algoritma *fingerprint similarity* dengan metode Winnowing dan Manber. Dataset yang digunakan menggunakan 10 dokumen abtrak yang berbeda, pada penerapan algoritma Winnowing menggunakan variasi jumlah *window* yang ditetapkan pada satu angka yaitu dengan nilai 4 untuk percobaan

menggunakan biword dan triword. Pada perbandingan penerapan biword dan triword dalam algoritma Winnowing memiliki hasil nilai rata-rata *similarity* 94% untuk penerapan biword dan 91.22% untuk peneraan triword sedangkan dalam perbandingan algoritma Winnowing dan algoritma Manber menghasilkan kesimpulan bahwa algoritma Winnowing memiliki hasil tingkat kemiripan 94% dan algoritma Manber 90.56%. Jadi dalam penerapan perbandingan yang dilakukan, algoritma Winnowing dengan metode biword memiliki hasil yang paling baik dibandingkan dalam penerapan algoritma Winnowing dengan triword dan penerapan algoritma Manber.

Penerapan algoritma Winnowing lainnya yaitu [13] yang menerapkan pemotongan gram dengan jumlah gram 3, pada penelitian ini memotong berdasarkan huruf dengan pemotongan k-gram atau bukan berdasarkan kata. Pada pemakaian jumlah *window* menggunakan nilai 2, pada pemilihan *fingerprint* dilakukan dengan memilih nilai terkecil pada setiap *window* namun jika terdapat nilai terkecil yang sama pada *window* berikutnya atau duplikasi nilai *fingerprint* maka hanya dipilih satu dari nilai yang sama tersebut. Pada penerapan algoritma Winnowing tersebut digunakan untuk memberikan rekomendasi pemilihan dosen pembimbing berdasarkan kata kunci yang dibandingkan. Namun pada penelitian tersebut tidak dilakukan perbandingan hasil tentang pemilihan nilai k-gram ataupun jumlah *window*. Pada penerapan sistem yang dibuat mampu menunjukkan persentase rekomendasi pemilihan dosen pembimbing dengan hasil untuk Saiful Bukhori memiliki nilai 16.67 % dan Slamin memiliki nilai 1.18 %.

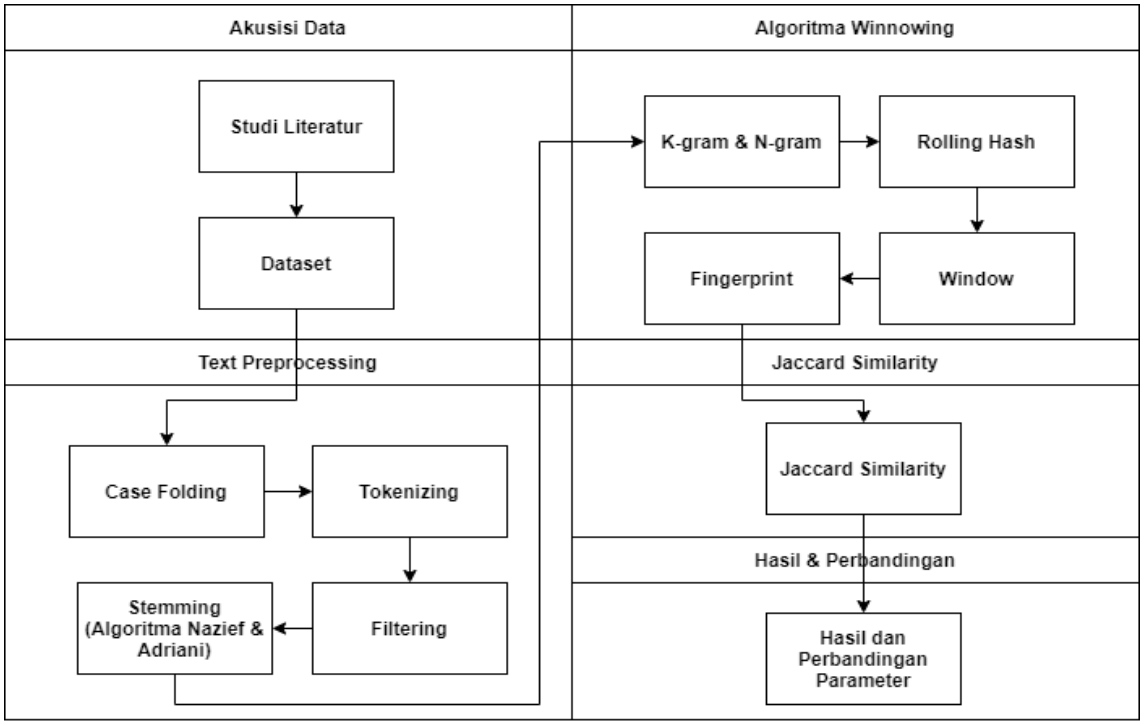
Pendeteksi kemiripan dokumen teks lainnya [14] menggunakan pemotongan berdasarkan karakter dengan jumlah *k* pada pemotongan gram terhadap algoritma Winnowing. Pada penelitian tersebut pada penentuan *fingerprint* menggunakan pemilihan nilai *window* terkecil namun jika ada nilai yang sama maka akan dipilih nilai paling kanan dari *window* tersebut. Pada pencocokan yang dilakukan mengakumulasinya dengan Jaccard Similarity dengan menghitung berdasasarkan *fingerprint* pada kedua dokumen yang dibandingkan. Dalam pengujian yang dilakukan menggunakan nilai variasi k-gram dan w-gram sebanyak 6, 12, dan 24 serta diterapkan pada dataset yang berupa 2 file tugas akhir. Hasil rata-rata pada penerapan variasi k-gram dan w-gram menunjukkan bahwa penerapan nilai w-gram 6 menghasilkan nilai rata-rata tingkat *similarity* 12.40%, nilai w-gram 12 menghasilkan 9.11% dan nilai w-gram 24 menghasilkan nilai 8.42 %.

Pendeteksian plagiarisme lainnya diteliti oleh [15] dalam penelitian tersebut melakukan pengujian terhadap variasi nilai gram, *window* dan basis terhadap algoritma Winnowing. Dalam penerapannya menggunakan dataset yang berjumlah 5, dengan keterangan bahwa dokumen tersebut merupakan dokumen milik mahasiswa pada mata kuliah Metode Penelitian di STMIK Akba Makassar. Selanjutnya dalam penerapannya membandingkan kemungkinan perbandingan antara dokumen satu dengan yang lainnya, hasil dari penelitian tersebut menghasilkan pemilihan nilai gram 8, *window* 6 dan basis 23 yang diterapkan pada dataset yang diteliti karena pada pengujian sebelumnya ketika menggunakan nilai basis 2 hasil dari tingkat presentase tidak stabil.

Berdasarkan perbedaan pemakaian langkah dan parameter dari penelitian sebelumnya maka akan diteliti mengenai perbandingan pemakaian parameter pada penerapan k-gram dengan nilai bilangan prima 2, 3, 5 dan 7 serta menerapkan n-gram dengan pemotongan per-kata pada unigram, bigram, trigram dan four-gram sehingga dapat diketahui pemilihan parameter yang paling baik berdasarkan nilai tingkat kesamaannya dalam mendeteksi kesamaan dokumen.

2. METODE PENELITIAN

Dalam melakukan penelitian telah dibuatkan struktur penelitian supaya penelitian yang dilakukan dapat terarah sesuai tujuan. Berikut alur penelitian diuraikan pada Gambar 1.



Gambar 1. Alur Penelitian

2.1. Akusisi Data

Pada penelitian ini dilakukan studi literatur untuk mengetahui topik penelitian dan pembuatan perumusan masalah. Setelah itu menggunakan dataset publik pada dataset dokumen Bahasa Indonesia.

2.2. Text Preprocessing

Pada pemrosesan teks terdapat beberapa langkah yaitu case folding untuk merubah semua huruf menjadi lowercase, tokenizing untuk membuat potongan kata berdasarkan spasi, filtering untuk membuang kata yang kurang penting, dan stemming untuk merubah kata menjadi kata baku [16]. Pada tahap stemming diterapkan algoritma Nazief & Adriani dengan memiliki 6 langkah dalam penerapan aturan yaitu (1) mengecek kata terhadap kamus kata dasar, jika ditemukan maka proses berhenti, (2) Menghapus sufiks infleksi, sufiks infleksi tidak memengaruhi ejaan kata yang dilampirkan, dan sufiks infleksi ganda selalu muncul berurutan. (3) Menghapus semua sufiks derivatif {"-i," "-kan," dan "-an"}. (4) Menghapus semua awalan turunan {"be-," "di-," "kem," "me-," "pe-," "se," dan "te-"}. (5) Pengkodean ulang kata jika terjadi pemotongan yang tidak valid. (6) Jika semua langkah tidak berhasil, akan mengembalikan kata asli [17].

2.3. Algoritma Winnowing

Setelah membersihkan teks pada tahap *preprocessing*, selanjutnya memproses teks tersebut pada algoritma Winnowing sebagai berikut [18]:

1. Memotong Substring
   Terdapat metode dalam memotong sebuah teks, yaitu k-gram dengan teknik pemotongan perkarakter dan n-gram dengan implementasi pemotongan berdasarkan perkata.
2. Rolling Hash
   Berdasarkan perpotongan substring yang dilakukan, semua substring yang terbentuk akan diproses pada proses hashing dengan teknik Rolling Hash. Berikut persamaan 1 (Rolling Hash).

$$H_{(c1\ldots ck)} = c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \cdots + c_{(k-1)} * b^k + c_k \quad (1)$$

Keterangan:
c: nilai ascii karakter
b: basis (bilangan prima)
k: banyak karakter

3. Window
   Setelah dilakukan pembuatan hash pada setiap potongan substring, selanjutnya mengelompokkan hash tersebut kedalam window dengan jumlah anggota setiap window yang ditentukan dengan nilai window.
4. Fingerprint
   Selanjutnya, mencari nilai fingerprint yaitu berdasarkan nilai hash terkecil dari setiap window, jika ada nilai terkecil yang sama maka diambil salah satu nilai yang paling kanan.

2.4. Jaccard Similarity

Setelah menentukan nilai *fingerprint* selanjutnya mencocokkan *fingerprint* yang sama, dan hasilnya diakumulasi dengan persamaan 2, Jaccard Similarity sebagai berikut [19]:

$$JS(X,Y) = \frac{|X \cap Y|}{|X|+|Y|-|X \cap Y|} \quad (2)$$

Keterangan:
X: *fingerprint* dokumen 1
Y: *fingerprint* dokumen 2

2.5. Hasil & Perbandingan

Berdasarkan penerapan algoritma stemming Nazief & Adriani, algoritma Winnowing serta Jaccard Similarity akan dilakukan perbandingan pada pemilihan parameter terbaik pada pemotongan sub*string* menggunakan k-gram dan n-gram pada langkah algoritma Winnowing.

3. HASIL DAN PEMBAHASAN

3.1. Akusisi Data

Dataset yang digunakan pada penelitian ini merupakan dataset publik yang berasal dari Github https://github.com/anggamaulana/Document*Similarity* yang merupakan dokumen abstrak dari penelitian terkait sebanyak 5 dokumen.

3.2. Text Preprocessing

Setelah melakukan akusisi data selanjutnya data tersebut diproses pada tahap *text preprocessing*. Dari tahap case folding untuk merubah huruf menjadi lowercase, tokenizing untuk memisah huruf berdasarkan spasi dan filtering untuk menghapus kata yang kurang penting, dan langkah terakhir dari *text preprocessing* yaitu stemming untuk merubah kata menjadi kata dasar, pada penelitian ini menggunakan penerapan algoritma stemming Nazief & Adriani pada tahap stemming, hasil dari stemming ditampilkan pada Tabel 1.

Tabel 1. Hasil Stemming

id	Stemming
1	["modern","nakal","remaja","masyarakat","masyarakat","daerah","kota","nakal", ... "akurasi"]
2	["masalah","pasca","panen","buah","belimbing","produksi","skala","industri", ... "belimbing"]
...	...
5	["tempat","praktek","kerja","lapang","pkl","mahasiswa","maksimal","mampu", ... "selatan"]

3.3. Algoritma Winnowing.

Tahap pertama dari algoritma Winnowing yaitu melakukan pemotongan dari *string*, pada penelitian ini dilakukan 2 tipe pemotongan yaitu menggunakan k-gram dan n-gram. K-gram menggunakan angka 2, 3, 5, dan 7 sedangkan pada n-gram menggunakan nilai unigram, bigram, trigram dan four-gram. Hasil dari pemotongan kata menggunakan k-gram berdasarkan per-huruf ditampilkan pada Tabel 2.

Tabel 2. Hasil Pemotongan *String* dengan K-Gram

id	k-gram			
	***k* = 2**	***k* = 3**	***k* = 5**	***k* = 7**
1	"mo", "od", "de", "er", "rn", "nn", ... "si"	"mod", "ode", "der", "ern", "rnn", "nna", ... "asi"	"moder", "odern", "dernn", "ernna", ... "urasi"	"modernn", "odernna", ... "akurasi"
2	"ma", "as", "sa", "al", "la", "ah", ... "ng"	"mas", "asa", "sal", "ala", "lah", "ahp", ... "ing"	"masal", "asala", "salah", "alahp", ... "mbing"	"masalah", "asalahp", ... "limbing"
..	...	...	...	...
5	"te", "em", "mp", "pa", "at", "tp", ... "an"	"tem", "emp", "mpa", "pat", "atp", "tpr", ... "tan"	"tempa", "empat", "mpatp", "patpr", ... "latan"	"tempatp", "empatpr", ... "selatan"

Selain menggunakan k-gram, dilakukan pemotongan *string* menggunakan n-gram dengan unigram, bigram, trigram dan fougram yang memotong *string* berdasarkan per-kata. Hasil dari n-gram ditampilkan pada Tabel 3.

Tabel 3. Hasil Pemotongan *String* dengan N-Gram

id	n-gram			
	unigram	**bigram**	**trigram**	**four-gram**
1	"modern", "nakal", "remaja", ... "akurasi"	"modern nakal", "nakal remaja", ... "tingkat akurasi"	"modern nakal remaja", "nakal remaja masyarakat", ... "hasil tingkat akurasi"	"modern nakal remaja masyarakat", ... "video hasil tingkat akurasi"
2	"masalah", "pasca", "panen", "buah", ... "belimbing"	"masalah pasca", "pasca panen", ... "buah belimbing"	"masalah pasca panen", "pasca panen buah", ... "citra buah belimbing"	"masalah pasca panen buah", ... "fitur citra buah belimbing"
..	...	...	...	...
5	"tempat", "praktek", "kerja", ... "selatan"	"tempat praktek", "praktek kerja", "kerja lapang", ... "aceh selatan"	"tempat praktek kerja", "praktek kerja lapang", "kerja lapang pkl", ... "politeknik aceh selatan"	"tempat praktek kerja lapang", ... "komputer politeknik aceh selatan"

Setelah pemotongan *string* dengan k-gram dan n-gram selanjutnya dilakukan pengubahan *string* menjadi sebuah hash menggunakan rolling hash dengan basis nilai bilangan prima 11. Berikut contoh perhitungan rolling hash dengan masukan *string* "mo" pada persamaan 3.

$$H_{(mo)} = 109 * 11^{(2-1)} + 111 * 11^{(2-2)} \quad (3)$$
$$H_{(mo)} = 1310$$

Berikut ditampilkan hasil rolling hash pada potongan k-gram yang sebelumnya telah diketahui, hasil dari perhitungan rolling hash ditampilkan pada Tabel 4.

Tabel 4. Hasil Rolling Hash pada K-Gram.

id	k-gram			
	***k* = 2**	***k* = 3**	***k* = 5**	***k* = 7**
1	1310, 1321, 1201 ... 1370	14510, 14632, 13325 ... 13107	1756935, 1771836, 1613645 ... 1877838	212590455, 214393463, 195252219 ... 190951712
2	1296, 1182, 1362 ... 1313	14371, 13099, 15090 ... 14018	1740066, 1586264, 1827061 ... 1740325	210549157, 191939200, 221075710 ... 209979268
..	...	...	...	...
5	1377, 1220, 1311 ... 1177	15256, 13532, 14518 ... 15213	1847305, 1638555, 1758066 ... 1725548	223525293, 198266501, 212727337 ... 221721214

Pemotongan yang dihasilkan dari n-gram dilakukan juga perhitungan dengan rolling hash, hasil rolling hash pada n-gram ditampilkan pada Tabel 5.

Tabel 5. Hasil Rolling Hash pada N-Gram

id	n-gram			
	unigram	**bigram**	**trigram**	**four-gram**
1	19326395, 1753739, 19996634 ... 190951712	34237894559966, 34175488468955, 5.7052741634162e+18 ... 4.8053509399811e+16	6.671997059701e+20, 9.7506657090256e+24, 1.6277813028581e+30 ... 7.6596037971803e+22	1.903598627415e+32, 2.7819787230115e+36, 3.1720852599398e+37 ... 1.5309568521711e+29
2	210549157, 1784000, 1783430 ... 23185353877	3.7300068206171e+14, 3160471761062, 287223799208 ... 3.7808932740694e+15	6.6079346131393e+20, 5.0899713759141e+17, 7.4498456356689e+21 ... 6.7028325227805e+21	1.0642144773807e+26, 1.3202074878402e+28, 1.7566346314932e+31 ... 1.2199887882438e+28
..	...	...	...	...
5	20320471, 218352494, 1716075 ... 221721214	4.3558742668949e+15, 3.8682476949284e+14, 33441522639584 ... 30504328435867	7.7166969721346e+21, 7.5381204301427e+21, 4.8961733296621e+17 ... 1.0018315799464e+25	1.5037659345117e+29, 1.1036562121772e+26, 1.2699412657154e+28 ... 2.2665573573324e+34

Setelah dilakukan perhitungan rolling hash pada k-gram dan n-gram selanjutnya dilakukan pembuatan *window* menggunakan ukuran *window* sebanyak 2. Hasil dari pembuatan *window* dari hash yang didapatkan di k-gram ditampilkan pada Tabel 6.

Tabel 6. *Window* pada Rolling Hash K-Gram

id	k-gram			
	***k* = 2**	***k* = 3**	***k* = 5**	***k* = 7**
1	{1310, 1321}, {1321, 1201}, {1201, 1225} ... {1182, 1370}	{14510, 14632}, {14632, 13325}, {13325, 13585} ... {14976, 13107}	{1756935, 1771836}, {1771836, 1613645}, {1613645, 1645092} ... {1737290, 1877838}	{212590455, 214393463}, {214393463, 195252219}, {195252219, 199057406} ... {222860313, 190951712}
2	{1296, 1182}, {1182, 1362}, {1362, 1175} ... {1265, 1313}	{14371, 13099}, {13099, 15090}, {15090, 13022} ... {13123, 14018}	{1740066, 1586264}, {1586264, 1827061}, {1827061, 1576918} ... {1695507, 1740325}	{210549157, 191939200}, {191939200, 221075710}, {221075710, 190808260} ... {198016676, 209979268}
..	...	...	...	...
5	{1377, 1220}, {1220, 1311}, {1311, 1329} ... {1373, 1177}	{15256, 13532}, {13532, 14518}, {14518, 14735} ... {13110, 15213}	{1847305, 1638555}, {1638555, 1758066}, {1758066, 1784281} ... {1635599, 1725548}	{223525293, 198266501}, {198266501, 212727337}, {212727337, 215899175} ... {204398808, 221721214}

Berikut hasil dari pembuatan *window* dari perhitungan rolling hash pada n-gram, hasil *window* ditampilkan pada Tabel 7.

Tabel 7. *Window* pada Rolling Hash N-Gram

id	n-gram			
	unigram	**bigram**	**trigram**	**four-gram**
1	{19326395, 1753739}, {1753739, 19996634}, {19996634, 280281927580} ... {224173164, 190951712}	{34237894559966, 34175488468955}, {34175488468955, 5.7052741634162e+18}, {5.7052741634162e+18, 7.9967705001031e+22} ... {3.5732617008670e+14, 4.8053509399811e+16}	{6.671997059701e+20, 9.7506657090256e+24}, {9.7506657090256e+24, 1.6277813028581e+30}, {1.6277813028581e+30, 1.5583443418327e+30} ... {7.1420267032046e+20, 7.6596037971803e+22}	{1.903598627415e+32, 2.7819787230115e+36}, {2.7819787230115e+36, 3.1720852599398e+37}, {3.1720852599398e+37, 2.5097291459649e+35} ... {1.0794979859123e+27, 1.5309568521711e+29}
2	{210549157, 1784000}, {1784000, 1783430}, {1783430, 145766} ... {145766, 23185353877}	{3.7300068206171e+14, 3160471761062}, {3160471761062, 287223799208}, {287223799208, 3.7808932740694e+15} ... {258423209929, 3.7808932740694e+15}	{6.6079346131393e+20, 5.0899713759141e+17}, {5.0899713759141e+17, 7.4498456356689e+21}, {7.4498456356689e+21, 8.9151485655093e+24} ... {4.7035849048666e+17, 6.7028325227805e+21}	{1.0642144773807e+26, 1.3202074878402e+28}, {1.3202074878402e+28, 1.7566346314932e+31}, {1.7566346314932e+31, 1.5793729507862e+31} ... {9.0420826076092e+23, 1.2199887882438e+28}
..	...	...	...	...
5	{20320471, 218352494}, {218352494, 1716075}, {1716075, 18975807} ... {142301, 221721214}	{4.3558742668949e+15, 3.8682476949284e+14}, {3.8682476949284e+14, 33441522639584}, {33441522639584, 277824847716} ... {4.6736182577216e+16, 30504328435867}	{7.7166969721346e+21, 7.5381204301427e+21}, {7.5381204301427e+21, 4.8961733296621e+17}, {4.8961733296621e+17, 7.206061040019e+21} ... {1.0573657348642e+26, 1.0018315799464e+25}	{1.5037659345117e+29, 1.1036562121772e+26}, {1.1036562121772e+26, 1.2699412657154e+28}, {1.2699412657154e+28, 1.6991514990518e+31} ... {2.2013576196466e+33, 2.2665573573324e+34}

Selanjutnya setelah di bentuk *window* maka akan dipilih *fingerprint* dari nilai terkecil pada setiap *window*, berdasarkan literatur yang ada nilai *fingerprint* apabila menemukan nilai terkecil yang sama maka akan diambil salah satu. Hasil dari *fingerprint* pada rolling hash k-gram ditampilkan pada Tabel 8.

Tabel 8. Fingerprint pada Rolling Hash K-Gram

id	k-gram			
	***k* = 2**	***k* = 3**	***k* = 5**	***k* = 7**
1	1310, 1201, 1225 ... 1294	14510, 13325, 13585 ... 14976	1756935, 1613645, 1645092 ... 1737290	212590455, 195252219, 199057406 ... 190951712
2	1182, 1175, 1171 ... 1353	13099, 13022, 12993 ... 12932	1586264, 1576918, 1573335 ... 1565943	191939200, 190808260, 190374721 ... 189480282
..	...	...	...	...
5	1220, 1311, 1183 ... 1339	13532, 14518, 13125 ... 13587	1638555, 1758066, 1589476 ... 1645356	198266501, 212727337, 192327889 ... 199089347

Berikut hasil dari *fingerprint* yang didapatkan dari *window* yang terbentuk pada rolling hash n-gram, hasil *fingerprint* ditampilkan pada Tabel 9.

Tabel 9. Fingerprint pada Rolling Hash N-Gram

id	n-gram			
	unigram	**bigram**	**trigram**	**four-gram**
1	1753739, 19996634, 280281927580 ... 190951712	34175488468955, 5.7052741634162e+18, 5.4619018510354e+18 ... 3.5732617008670e+14	6.671997059701e+20, 9.7506657090256e+24, 1.5583443418327e+30 ... 7.1420267032046e+20	1.903598627415e+32, 2.7819787230115e+36, 2.5097291459649e+35 ... 1.0794979859123e+27
2	1784000, 1783430, 145766 ... 1788924	3160471761062, 287223799208, 3.7808932740694e+15 ... 258423209929	5.0899713759141e+17, 7.4498456356689e+21, 8.9151485655093e+24 ... 4.7035849048666e+17	1.0642144773807e+26, 1.3202074878402e+28, 1.5793729507862e+31 ... 9.0420826076092e+23
..	...	...	...	...
5	20320471, 1716075, 14837 ... 2301135907	3.8682476949284e+14, 33441522639584, 277824847716 ... 4.7907955632875e+16	7.5381204301427e+21, 4.8961733296621e+17, 7.206061040019e+21 ... 1.0573657348642e+26	1.1036562121772e+26, 1.2699412657154e+28, 1.6079668527932e+30 ... 2.2013576196466e+33

3.4. Jaccard Similarity

Berdasarkan hasil *fingerprint* pada setiap dokumen, selanjutnya dihitung kesamaan antara dokumen satu dengan yang lainnya, hasil tingkat kesamaan dihitung menggunakan Jaccard Similarity dengan perhitungan misalnya dokumen 1 terhadap dokumen 2 pada k-gram 2, diketahui jumlah *fingerprint* dokumen 1 sebanyak 111 dan *fingerprint* dokumen 2 sebanyak 87 dan *fingerprint* yang sama antara dua dokumen tersebut adalah 62 sehingga perhitungannya pada persamaan 4 sebagai berikut.

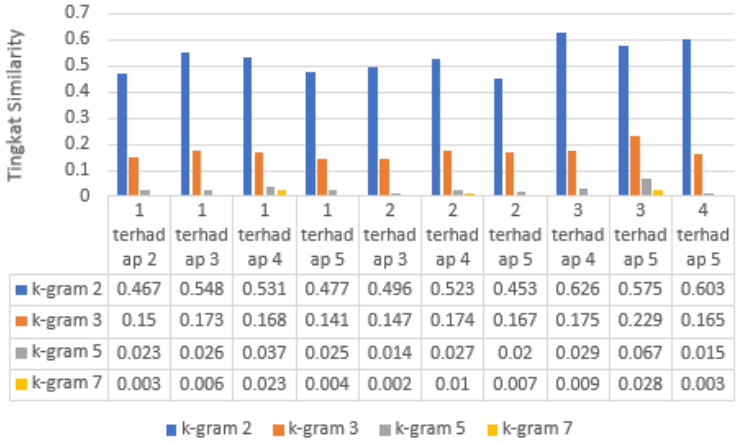
$$JS(X,Y) = \frac{|X \cap Y|}{|X|+|Y|-|X \cap Y|} \quad (4)$$

$$JS(X,Y) = \frac{62}{198-62}$$

$$JS(X,Y) = 0.456$$

3.5. Hasil & Perbandingan

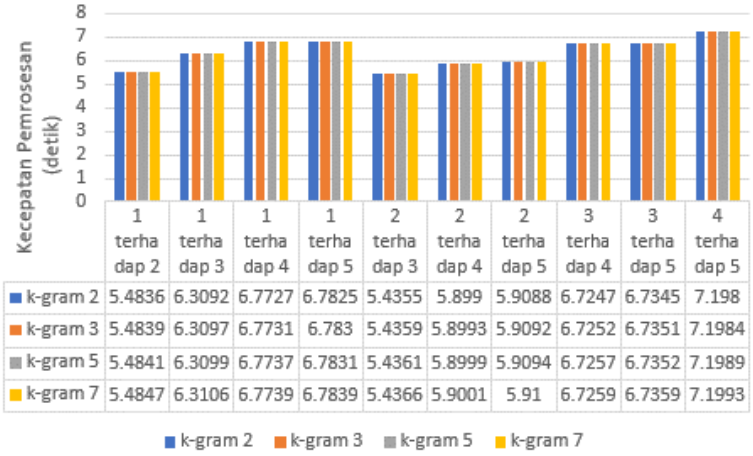
Berdasarkan hasil dari tingkat *similarity* yang dihitung menggunakan Jaccard Similarity dapat ditampilkan perbandingan dari nilai *similarity* berdasarkan perbedaan nilai parameter pada k-gram, perbandingan tingkat *similarity* ditampilkan pada Gambar 2.



	1 terhadap 2	1 terhadap 3	1 terhadap 4	1 terhadap 5	2 terhadap 3	2 terhadap 4	2 terhadap 5	3 terhadap 4	3 terhadap 5	4 terhadap 5
k-gram 2	0.467	0.548	0.531	0.477	0.496	0.523	0.453	0.626	0.575	0.603
k-gram 3	0.15	0.173	0.168	0.141	0.147	0.174	0.167	0.175	0.229	0.165
k-gram 5	0.023	0.026	0.037	0.025	0.014	0.027	0.02	0.029	0.067	0.015
k-gram 7	0.003	0.006	0.023	0.004	0.002	0.01	0.007	0.009	0.028	0.003

Gambar 2. Perbandingan Tingkat *Similarity* Berdasarkan Nilai K-Gram

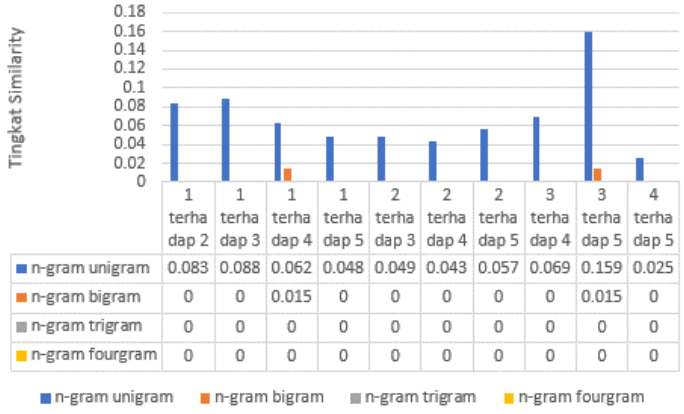
Selain hasil *similarity*, didapatkan hasil waktu kecepatan pemrosesan dari tahap *text preprocessing*, algoritma Winnowing dan menghitung Jaccard Similarity terhadap dokumen yang dibandingkan, hasil kecepatan pemrosesan ditampilkan pada Gambar 3.



	1 terhadap 2	1 terhadap 3	1 terhadap 4	1 terhadap 5	2 terhadap 3	2 terhadap 4	2 terhadap 5	3 terhadap 4	3 terhadap 5	4 terhadap 5
k-gram 2	5.4836	6.3092	6.7727	6.7825	5.4355	5.899	5.9088	6.7247	6.7345	7.198
k-gram 3	5.4839	6.3097	6.7731	6.783	5.4359	5.8993	5.9092	6.7252	6.7351	7.1984
k-gram 5	5.4841	6.3099	6.7737	6.7831	5.4361	5.8999	5.9094	6.7257	6.7352	7.1989
k-gram 7	5.4847	6.3106	6.7739	6.7839	5.4366	5.9001	5.91	6.7259	6.7359	7.1993

Gambar 3. Perbandingan Kecepatan Pemrosesan Berdasarkan Nilai K-Gram

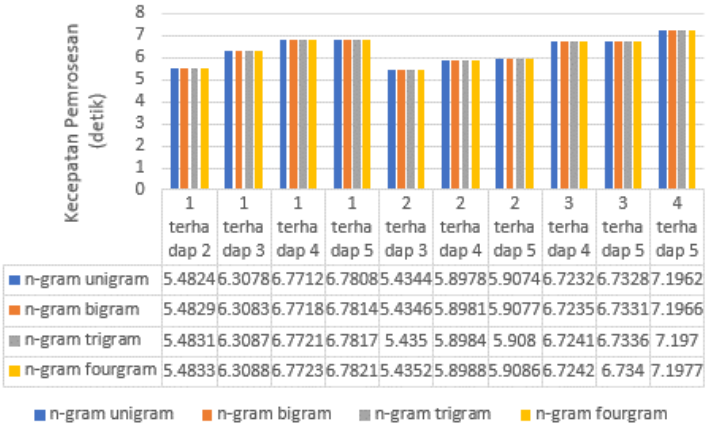
Berikut ditampilkan hasil tingkat *similarity* terhadap dokumen yang dibandingkan berdasarkan nilai dari n-gram, hasil perbandingan tingkat *similarity* ditampilkan pada Gambar 4.



	1 terhadap 2	1 terhadap 3	1 terhadap 4	1 terhadap 5	2 terhadap 3	2 terhadap 4	2 terhadap 5	3 terhadap 4	3 terhadap 5	4 terhadap 5
n-gram unigram	0.083	0.088	0.062	0.048	0.049	0.043	0.057	0.069	0.159	0.025
n-gram bigram	0	0	0.015	0	0	0	0	0	0.015	0
n-gram trigram	0	0	0	0	0	0	0	0	0	0
n-gram fourgram	0	0	0	0	0	0	0	0	0	0

Gambar 4. Perbandingan Tingkat *Similarity* Berdasarkan Nilai N-Gram

Perhitungan dari kecepatan waktu pemrosesan terhadap pemrosesan variasi n-gram ditampilkan pada Gambar 5.



	1 terhadap 2	1 terhadap 3	1 terhadap 4	1 terhadap 5	2 terhadap 3	2 terhadap 4	2 terhadap 5	3 terhadap 4	3 terhadap 5	4 terhadap 5
n-gram unigram	5.4824	6.3078	6.7712	6.7808	5.4344	5.8978	5.9074	6.7232	6.7328	7.1962
n-gram bigram	5.4829	6.3083	6.7718	6.7814	5.4346	5.8981	5.9077	6.7235	6.7331	7.1966
n-gram trigram	5.4831	6.3087	6.7721	6.7817	5.435	5.8984	5.908	6.7241	6.7336	7.197
n-gram fourgram	5.4833	6.3088	6.7723	6.7821	5.4352	5.8988	5.9086	6.7242	6.734	7.1977

Gambar 5. Perbandingan Kecepatan Pemrosesan Berdasarkan Nilai N-Gram

Penelitian penerapan algoritma Winnowing sebelumnya [13] tidak menggunakan variasi penentuan jumlah pemotongan karakter atau kata, berdasarkan hasil pada penelitian sebelumnya terdapat hasil tingkat *similarity* yang mendominasi pada 0% sehingga perlu dilakukan pemilihan jumlah *k* atau *n* terbaik untuk mengetahui penggunaan parameter yang optimal. Penelitian terkait [15] menggunakan nilai *fingerprint* dari nilai terkecil pada setiap *window* tanpa memilih nilai *fingerprint* yang sama serta menggunakan nilai pemotongan string k-gram berdasarkan huruf dengan melakukan pengujian menggunakan nilai *k* 5 dan 8 hasilnya pada nilai *k* = 5 menunjukkan hasil yang tidak relevan pada perbandingan kesamaan dokumen karena pada beberapa pengujian hasilnya melebihi 100% karena memperoleh *fingerprint* yang sama sangat tinggi dan tidak menerapkan perhitungan Jaccard Similarity dan berbeda pada penerapan *k* = 8 dengan hasil nilai tengah 8.79% pada beberapa pengujian yang dilakukan. Perbandingan terhadap penelitian tersebut, dengan pengujian yang dilakukan pada penelitian ini dengan variasi nilai *k* hasilnya tetap di jarak 0.0 sampai 1 atau 0% sampai 100% walaupun menggunakan nilai *k* kecil.

Setelah didapatkan hasil *similarity* dapat diketahui bahwa penggunaan nilai *k* pada k-gram dengan nilai kecil akan menaikkan nilai tingkat *similarity* dan semakin besar nilai *k* pada k-gram akan memperkecil nilai *similarity* serta didapatkan rata-rata nilai tingkat *similarity* berdarkan nilai pengujian terhadap nilai k, pada *k* = 2 sebesar 0.5299, *k* = 3 sebesar 0.1689, *k* = 5 sebesar 0.0283 dan *k* = 7 sebesar 0.0095. Hasil pada penerapan n-gram dengan pemotongan unigram mendominasi hasil nilai dibawah 0.10 atau 10% namun ketika diterapkan dengan bigram, trigram dan four-gram mendominasi nilai tingkat *similarity* 0 atau 0% dengan detail nilai rata-rata pada setiap pengujian yaitu pada unigram sebesar 0.0683, bigram sebesar 0.003, trigram sebesar 0.000 dan four-gram sebesar 0.000.

Berdasarkan kecepatan pemrosesan waktu pada k-gram tidak terlihat perbedaan yang signifikan terhadap pada penerapan n-gram, dengan nilai rata-rata waktu pemrosesan pada *k* = 2 memiliki 6.32485 detik, *k* = 3 memiliki 6.32528 detik, *k* = 5 memiliki 6.3256 detik dan *k* = 7 memiliki 6.32609 detik. Sedangkan pada n-gram didapatkan rata-rata waktu pemrosesan pada unigram memiliki 6.3234 detik, bigram memiliki 6.3238 detik, trigram memiliki 6.32417 detik dan four-gram memiliki 6.3245 detik.

4. KESIMPULAN

Penerapan algoritma Winnowing dengan teknik pemotongan *string* menggunakan k-gram memiliki tingkat nilai *similarity* yang tinggi namun ketika nilai jumlah *k* semakin besar akan mengurangi tingkat nilai *similarity* dengan hasil rata-rata pada $k = 2$ sebesar 0.5299, $k = 3$ sebesar 0.1689, $k = 5$ sebesar 0.0283 dan $k = 7$ sebesar 0.0095. Penerapan pemotongan *string* berdasarkan kata menggunakan n-gram pada penerapan unigram memiliki rata-rata tingkat *similarity* sebesar 0.0683 sedangkan bigram memiliki tingkat nilai *similarity* 0.003, serta pada trigram dan four-gram memiliki tingkat kesamaan sebesar 0.000. Pada perbandingan kecepatan pemrosesan waktu pada k-gram dan n-gram tidak terlihat perbedaan yang signifikan dan keduanya mendominasi pada kecepatan waktu selama 6 detik.

5. SARAN

Berdasarkan penelitian yang dilakukan, belum dilakukan tentang perbandingan dari perbedaan penerapan dari nilai *window*. Sehingga dapat dilakukan penelitian lebih lanjut tentang penggunaan nilai perbedaan tersebut untuk mengetahui hasil perbandingannya.

DAFTAR PUSTAKA

[1] Sunardi., Yudhana, A., Mukaromah, I. A., 2018, Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan Jaccard Similarity Terhadap Algoritma Winnowing, *Jurnal Transmisi*, No. 3, Vol. 20, Hal. 105

[2] ALAMSYAH, N., 2017, Perbandingan Algoritma Winnowing Dengan Algoritma Rabin Karp Untuk Mendeteksi Plagiarisme Pada Kemiripan Teks Judul Skripsi, *Technologia Jurnal Ilmiah*, No. 3, Vol. 8, Hal. 124

[3] Harjanta, J., Tri, A., 2015, Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining, *Jurnal Informatika Upgris*, No. 1, Vol. 1

[4] Hariri, F. R., Utami, E., Amborowati, A., 2015, Learning Vector Quantization untuk Klasifikasi Abstrak Tesis, *Citec Journal*, No. 2, Vol. 2.

[5] Hariri, F. R., Pamungkas, D. P., 2016, Self Organizing Map-Neural Network untuk Pengelompokan Abstrak, *Citec Journal*, No. 2, Vol. 3

[6] Wirayasa, I. P. M., Wirawan, I. M. A., Pradnyana, A., 2019, Algoritma Bastal: Adaptasi Algoritma Nazief & Adriani Untuk Stemming Teks Bahasa Bali, *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, No. 1, Vol. 8

[7] Pamungkas, H. Y., Fitrianingsih., 2019, Deteksi Similaritas Dokumen Ilmiah Menggunakan Algoritma Rabin-Karp, *Jurnal Ilmiah Informatika Komputer*, No. 3, Vol. 24, Hal. 209–219

[8] Negoro, W. A., Amalia, F., Santoso, E., 2019, Pengembangan Aplikasi Resep Masakan dengan Rekomendasi berdasarkan Bahan-Bahan Makanan Berbasis Web, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* No. 9, Vol. 3, Hal. 9212–9221

[9] Khidfi, M. N., Isnawaty., Sari, J. Y., 2018, Rancang bangun aplikasi pendeteksian kesamaan pada dokumen teks menggunakan algoritma Enhaced Confix Stripping dan Algoritma Winnowing, SemanTIK Univ Haluoelo, No. 2, Vol. 4, no. 2, Hal. 1–10

[10] Sukmana, A., Kusrini., Sunyoto, A., 2018, Perbandingan Pengunaan Stemming Pada Deteksi Kemiripan Dokumen Menggunakan Metode Rabin Karp Dan Jaccard Similarity, *Seminar Nasional Teknologi Informasi dan Multimedia 2018*, Yogyakarta, 2 Februari

[11] Maskur., Putra, D. Q., Hayatin, N., 2019, Deteksi Kemiripan Dokumen Proposal Penelitian Dan Pengabdian Menggunakan Algoritma Biword Winnowing, *Jurnal Informatika Polinema*, No. 3, Vol. 6, Hal. 43–48

[12] Faisal, M., Nugroho, F., El Sulthan, M. M., Amini, F., Hariyadi, M. A., Sedayu, A., 2020, Plagiarism detection using manber and winnowing algorithm, International Journal of Advanced Science and Technology, No. 6, Vol. 29 Special Issue, Hal. 2130–2136.

[13] Jarwati., Prihandoko, A. C., Yulia R, W. E., 2017, Penerapan Algoritma Winnowing pada Sistem Rekomendasi Penetuan Dosen Pembimbing Skripsi (Studi Kasus: Prodi Sistem Informasi Universitas Jember Jember), Berkala SAINSTEK Jurnal, No. 1, Vol. 5, Hal. 11–20.

[14] Wibowo, R. K., Hastuti, K., 2016, Penerapan Algoritma Winnowing Untuk Mendeteksi Kemiripan Teks pada Tugas Akhir Mahasiswa, Techno.com, No. 4, vol. 15, Hal. 303–311.

[15] Ilham., Pasnur., 2017, Penerapan Algoritma Winnowing Untuk Mendeteksi Kemiripan Pada Karya Tulis Mahasiswa, Jurnal Teknologi Informasi dan Komunikasi, No. 2, Vol. 7, Hal. 131–136

[16] Prasidhatama, A., Suryaningrum, K. M., 2018, Perbandingan Algoritma Nazief & Adriani Dengan Algoritma Idris Untuk Pencarian Kata Dasar, Jurnal Teknologi dan Manajemen Informatika, No. 1, Vol. 4, Hal. 1–4

[17] Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M. M., Williams, H. E., 2007, Stemming Indonesian: A confix-stripping approach, ACM Transactions on Asian Language Information Processing, No. 4, Vol. 6.

[18] Sanjaya, S., Absyar, E. A., 2015, Pengelompokan Dokumen Menggunakan Winnowing Fingerprint dengan Metode K-Nearest Neighbour, Jurnal CoreIT, No. 2, Vol. 1, Hal. 50–56

[19] Hasanah, U., Mutiara, D. A., 2019, Perbandingan metode cosine similarity dan Jaccard Similarity untuk penilaian otomatis jawaban pendek, Seminar Nasional SENSITIf 2019, Makassar, 16 - 17 Desember