Analisa Perbandingan Jenis N-Gram Dalam Penentuan Similarity Text pada Deteksi Plagiat

Comparative Analysis N-Gram Types in Determining Similarity on Plagiarism Detections

Zudha Pratama*[1], Ema Utami[2], M. Rudyanto Arief[3]
[1,2,3]Magister Teknik Informatika Universitas Amikom Yogyakart
E-mail: ***[1]zudhapratama@gmail.com, [2]ema.u@amikom.ac.id, [3]rudy@amikom.ac.id [2]email**

Abstrak

Dampak.akses informasi yang mudah membuat tindakan plagiasi makin marak. Tindakan tersebut dapat dicegah dengan menggunakan sistem deteksi plagiat. Sistem tersebut dapat dibangun dengan menggunakan konsep similarity dengan algoritma rabin-karp sebagai string matchingnya dan n-gram sebagai metode parsingnya. Penelitian terdahulu menggunakan kedua algoritma tersebut menunjukkan hasil sistem yang cukup baik untuk deteksi plagiat. Kemudian hasil penelitian dari luar negeri ada yang melakukan hal serupa mengenai deteksi plagiat serta menghasilkan penemuan baru misalnya cross-language similarity. Selain itu ada temuan fakta-fakta baru mengenai deteksi plagiat dengan berbagai cara pengujian serta penggabungan berbagai metode yang sudah ada untuk perbaikan hasil deteksi. Sedangkan tujuan kami pada penelitian ini adalah membandingkan metode parsing untuk mengetahui metode parsing yang mana yang dapat memberikan hasil paling cepat dan masih dalam nilai akurasi yang wajar. Kami sebagai kontrol ukuran akurasi kami menggunakan plagiarism checker x free. Kami menggunakan aplikasi tersebut untuk menentukan akurasi instrumen uji kami menggunakan selisih similarity aplikasi ini dengan instrumen uji kami. Hasilnya kami menemukan fakta jika n-gram word memiliki akurasi yang paling optimal dibanding n-gram yang lain dan masih relatif paling cepat dibanding lainnya.

Kata Kunci *— perbandingan, ngram, similarity text, deteksi plagiat*

Abstract

The impact of easy information access makes plagiarism more and more prevalent. Such action can be prevented by using a plagiarism detection system. The system can be constructed using the concept of similarity with the rabin-karp algorithm as its matching string and n-gram as its parsing method. Earlier studies using both algorithms show good system results for plagiarism detection. Then the results of research from abroad have done the same about the detection of plagiarism and produce new inventions such as cross-language similarity. In addition, there are new facts about plagiarism detection by various testing methods and incorporating existing methods for improving the detection. While our goal in this study is to compare the method of parsing to find out which parsing method that can provide the fastest results and still in a reasonable accuracy value. We measure our accuracy as accurate using plagiarism checker x free. We use the application to determine the accuracy of our test instruments using the similarity difference of this application with our test instruments. We found that n-gram word has the most optimal accuracy compared to other n-grams and is still relatively fastest compared to others.

Keywords *— comparison, ngram, similarity text, plagiarism detection*

Dikirim: 18 Januari 2018; Direvisi: 22 Januari 2018 Diterima: 1 Februari 2018

1. PENDAHULUAN

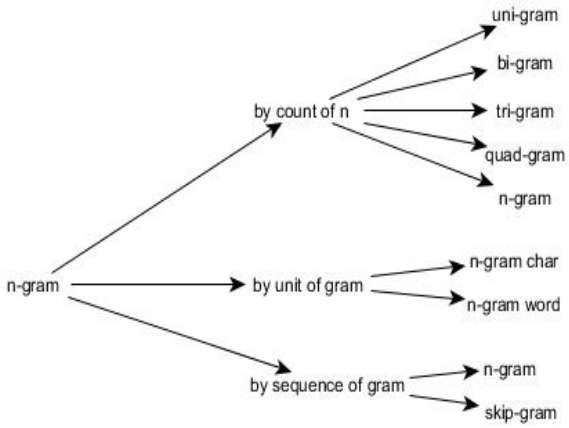
Kemudahan setiap orang untuk mengakses informasi pada internet saat ini, berdampak negatif pada tindakan penyalah gunaan dan pelanggaran hak cipta. Berkaitan dengan ketentuan plagiasi yang telah tertulis pada Permendiknas No.17 pasal 1 maka, pengutipan karya tulis tanpa disertai sumber yang tepat dapat dikategorikan tindakan plagiat. Salah satu metode untuk mencegah dan mendeteksi plagiat adalah menggunakan konsep similarity dengan string matching. String matching yang cukup terkenal untuk copy-paste plagiarism yaitu dengan algoritma Rabin-Karp. N-gram parsing merupakan urutan pemenggalan (kata atau karakter) sepanjang nilai n. Kemudian dalam n-gram sendiri terdapat beberapa varian jenis. Untuk mendapatkan alternatif terbaik dari penggunaan n-gram peneliti ingin membandingkannya.

Sebelum melakukan perbandingan kami melakukan studi literatur dan menemukan beberapa penelitian terkait dengan deteksi plagiat dengan Rabin-Karp [1][2][3]. Minaei dalam penelitiannya mempresentasikan sebuah metode untuk mengidentifikasi bagian-bagian serupa dari dua dokumen. Dengan membuat n-gram dari setiap dokumen dan dengan membandingkan n-gram dari dua dokumen untuk mengidentifikasi kemiripannya [4]. Ehsan mengusulkan memperkenalkan pendekatan baru untuk menilai cross-language similarity antara dua teks untuk mendeteksi plagiat dengan pendekatan vector-based retrieval framework dan analisa kesamaan yang berdasar pada dynamic text alignment pada beberapa tingkatan granularitas: kalimat, EDU (elementary discourse units), dan n-gram [5]. Nguyen mengusulkan metode deteksi plagiarisme bahasa Vietnam dengan menggabungkan empat Metode: substring n-gram, LCS, CS dan Fuzzy based. Model dari penggabungan keempat metode tersebut mampu mendeteksi tindakan plagiat dalam berbagai tingkatan plagiasi [6]. Kuta melakukan penelitian peningkatan deteksi plagiarisme intrinsic dengan meningkatkan kinerja metode profil n-gram karakter yang merupakan usulan dari penelitian Stamatatos. Peningkatan dilakukan dengan mengatur parameter dan modifikasi baru dengan rangkaian fitur yang banyak [7]. Bensalem mengenalkan metode pendeteksian plagiarisme intrinsik bahasa baru yang berbasis pada representasi teks baru dalam kelas n-gram / pengklasifikasian kemunculan n-gram. Sebagai contoh tingkat kelas kemunculan yang paling sering muncul, kelas kemunculan paling sering dan kelas kemunculan menengah [8]. Palkovskii menggabungkan semua hasil penelitian sebelumnya dari penelitian PAN12 dan PAN13 dan memperbaiki metode pendeteksian plagiarisme yang dikembangkan sebelumnya, dengan bantuan: n-gram kontekstual, n-gram konteks sekitar, n-gram berbasis entitas, dan lain-lain [9].

N-gram adalah model probabilistik yang awalnya dirancang oleh ahli matematika dari Rusia pada awal abad ke-20 dan kemudian dikembangkan untuk memprediksi item berikutnya dalam urutan item. Item bisa berupa huruf / karakter, kata, atau yang lain sesuai dengan aplikasi. Salah satunya, model n-gram yang berbasis kata digunakan untuk memprediksi kata berikutnya dalam urutan kata tertentu. Dalam arti bahwa sebuah n-gram hanyalah sebuah wadah kumpulan kata dengan masing-masing memiliki panjang n kata [10].

Kita dapat menggunakan model n-gram untuk memperkirakan probabilitas kata terakhir dari n-gram dari kata-kata sebelumnya, dan juga untuk menentukan probabilitas dalam seluruh rangkaian. N-gram digunakan untuk urutan kata itu sendiri atau model prediktif yang menugaskan sebuah probabilitas.

Berdasarkan dari jumlah potongan gram substring, n-gram dibedakan menjadi uni-gram, bi-gram, tri-gram, quad-gram, 5-gram dan seterusnya sejumlah nilai n dalam n-gram. Jika berdasarkandari satuan unit yang diambil dalam proses parsing, n-gram dapat dibedakan menjadin-gram karakter dan n-gram kata. Serta yang terakhir jika berdasarkan pengambilan urutan parsingterdapat n-gram itu sendiri dengan pengambilan substring secara urut dan untuk pengambilan substring yang tidak urut atau lompat-lompat (skip) disebut skip-gram [11]. Berikut Gambar 1 hasil studi literatur mengenai klasifikasi jenis n-gram.



Gambar 1. Jenis N-gram

Berikut contoh pemenggalan kalimat untuk sebagian jenis n-gram diatas, jika input berupa : "saya senang meneliti dan mengembangkan metode n-gram". Maka:

1. Uni-gram char: {'s','a','y','a',' ','s','e','n','a','n','g',…,'r','a','m'}
2. Bi-gram char:{'sa','ay','ya','a ',' s','se','en','na','an','ng',…,'gr','ra','am'}
3. 6-gram char: {'saya s','aya se','ya sen','a sena',…,' n-gra','n-gram'}
4. Uni-gram word: {'saya','senang','meneliti','dan',…,'metode','n-gram'}
5. Bi-gram word: {'saya senang','senang meneliti', 'dan mengembangkan','mengembangkan metode','metode n-gram'}
6. 1-Skip-bi-gram: {'saya senang','saya meneliti','senang meneliti','senang dan', 'meneliti dan' , 'dan mengembangkan', 'dan metode', 'mengembangkan metode', 'mengembangkan n-gram', 'metode n-gram'}.

Dalam contoh di atas terlihat skip-gram memiliki pola pengambilan kata yang berbeda, kata pertama muncul dua kali. Dalam tersebut, dua kali lipat lebih banyak 1-skip-bi-gram dihasilkan daripada bi-gram. Hal itu karena jika nilai k dari k-skip-n-gram adalah 1 maka himpunannya mencakup 1-skip dan 0-skip (n-gram yang terbentuk dari kata-kata yang berdekatan) [11]. Sesuai persamaan 1, sebagai berikut:

$$\{w_{i_1}, w_{i_1}, \dots w_{i_1} \mid \sum_{j=1}^{n} i_j - i_{j-1} < k\} \quad (1)$$

Keterangan:
w : kata
k : jarak *skip*
n : jumlah *gram*

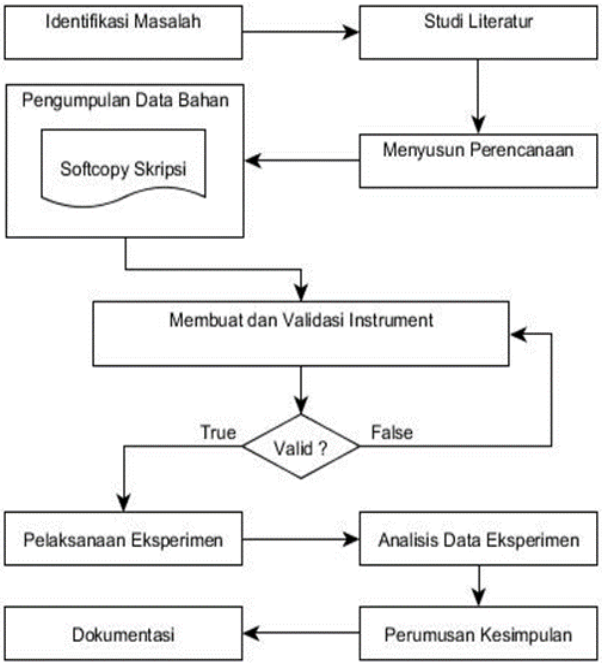
Algoritma Rabin-Karp ditemukan oleh Michael O. Rabin dan Richard M. Karp. Algoritma ini menggunakan metode hash dalam mencari suatu kata. Teori ini jarang digunakan untuk mencari kata tunggal, namun cukup penting dan sangat efektif bila digunakan untuk pencarian jamak[12].

Rabin Karp merepresentasikan setiap karakter ke dalam bentuk desimal digit (digit radix-d) ∑ = {0, 1, 2, 3, …, d}, dimana d = |∑|. Sehingga didapat masukan string berturut-turut sebagai perwakilan panjang n desimal. Karakter string 31415 sesuai dengan jumlah desimal 31,415. Kemudian pola p dihash menjadi nilai desimal dan string direpresentasikan dengan penjumlahan digit-digit angka menggunakan aturan Horner's.

Plagiarism Checker X yang merupakan produk dari anak perusahaan RealKit Technologies. Dapat digunakan untuk menganalisis teks untuk plagiarisme dengan mencari secara online untuk ungkapan yang identik dan indikator penyalinan lainnya.Hal ini juga dapat membandingkan teks secara berdampingan untuk persamaan menggunakan fitur side-by-side comparison. Opsi berikutnya memungkinkan anda membandingkan satu teks dengan banyak teks,

banyak ke satu, atau menjalankan perbandingan silang menggunakan fitur Bulk Search. Fitur yang akan kami gunakan adalah side-by-side comparison, karena kami akan menggunakannya sebagai alat control ukuran prosentase kemiripan yang akan dihasilkan oleh instrumen uji kami.

2. METODE PENELITIAN



Gambar 2. Alur Penelitian

Alur penelitian terlihat pada Gambar 2 yang terdiri dari 9 tahap. Tahap pertama adalah identifikasi masalah, peneliti melakukan kajian masalah yang diambil untuk menentukan spesifikasi dan batasan dari objek masalah yang akan diteliti. Hal ini dilakukan agar penelitian fokus pada permasalahan utama yang diteliti dan tujuan penelitian menjadi lebih jelas.

Tahap kedua adalah studi literatur, peneliti mengumpulkan data yang berhubungan dengan topik penelitian yang dilakukan dari berbagai media untuk menambah pengetahuan peneliti tentang riset-riset yang pernah dilakukan oleh peneliti sebelumnya. Pengetahuan ini tidak hanya berupa pemahaman terhadap riset-riset tersebut, tetapi juga hasil yang terbentuk antar riset-riset tadi.

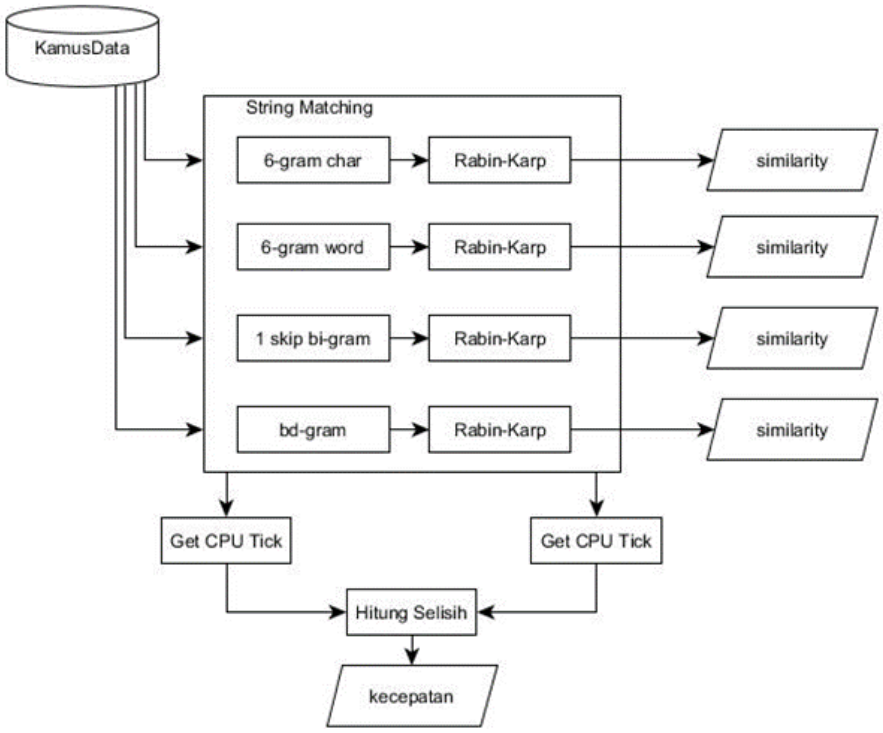
Tahap ketiga adalah menyusun perencanaan, peneliti menentukan langkah-langkah yang harus diambil untuk mencapai tujuan penelitian. Identifikasi variabel luar, menentukan cara kontrol, memilih metode penelitian yang tepat, mengidentifikasi prosedur pengumpulan data, dan rancangan prosedur dalam eksperimen.

Tahap keempat adalah pengumpulan data bahan yang berupa softcopy skripsi. Softcopy skripsi diambil dari petugas perpustakaan STMIK KADIRI dengan sebelumya melakukan permohonan secara tertulis kepada ketua STMIK KADIRI. Dari softcopy skripsi yang diambil hanya bab II saja yang digunakan sebagai bahan eksperimen. Karena bab II berisi teori yang biasanya mahasiswa melakukan copy-paste dari skripsi yang sudah ada tanpa melakukan penulisan ulang dengan gaya bahasanya sendiri.

Tahap kelima adalah membuat instrument eksperimen. Implementasi algoritma sesuai literatur yang didapat dari tahap studi literatur sebelumnya. Semua subproses tersebut diimplementasikan pada beberapa fungsi dalam sebuah aplikasi. Implementasi menggunakan bahasa pemrograman Delphi. Kemudian melakukan validasi instrument. Validasi dilakukan agar instrument menghasilkan nilai yang terukur sesuai dengan alat lain yang sejenis dan sudah

digunakan secara global dan teruji. Proses ini melibatkan aplikasi Plagiarism Checker X. Jika hasil instrument tidak sesuai maka akan kembali ke proses desain fungsi string matching lagi hingga mendapatkan instrumen yang memiliki hasil yang sesuai atau valid. Validasi menggunakan dokumen rekaan yang dibuat mirip 50 %. Fungsi dianggap valid jika selisih prosentase yang dihasilkan tidak terlalu jauh.

Tahap keenam adalah pelaksanaan eksperimen, semua bahan uji softcopy skripsi bab III di masukkan ke instrumen / aplikasi uji dan dicatat hasilnya. Pencatatan pada tiap tipe n-gram meliputi nama file uji, prosentase ketepatan dan kecepatan proses tiap jenis n-gram. prosentase ketepatan diperoleh dari selisih prosentase similarity yang dihasilkan Plagiarism Checker X dengan prosentase hasil penghitungan similarity pada masing-masing jenis n-gram. Prosentase similarity untuk tiap jenis n-gram itu sendiri diperoleh dari jumlah substringyangsama dibanding dengan jumlah total kedua substringdariseluruhisi dokumen. Sedangkan kecepatan proses diperoleh dari selisih waktu yang mulai dicatat instrument saat tombol proses ditekan hingga hasil similarity ditampilkan. Waktu tersebut diperoleh dari CPU Tick, yakni waktu yang berjalan dalam processor computer. Berikut skema teknis pengujian disajikan pada Gambar 3.



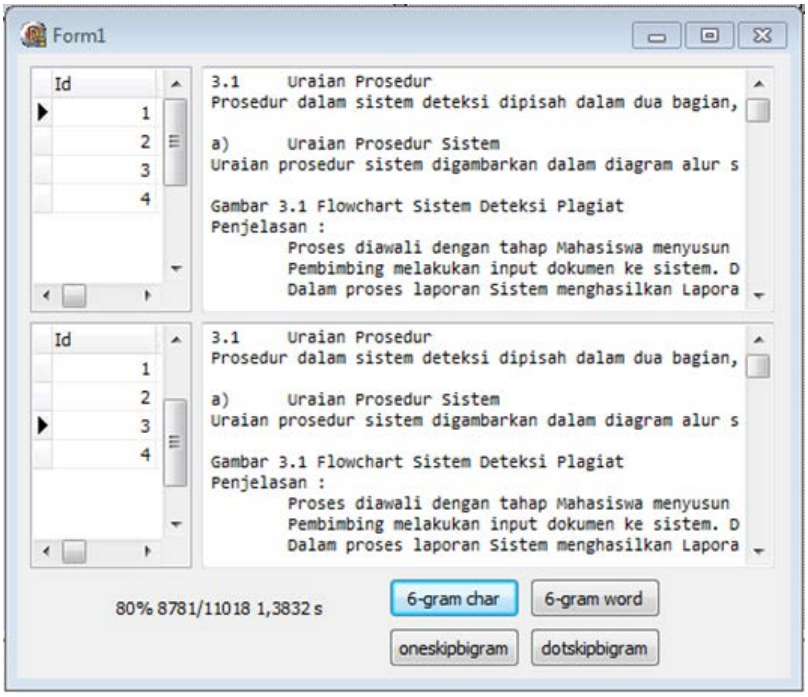
Gambar 3. Alur Teknis Eksperimen

Skenario terdiri dari 24 kali tahap pembandingan dokumen dengan instrumen dan Plagiarism Checker X Free. Untuk masing-masing metode n-gram sebanyak 6 kali dan setiap pembandingan pada instrumen juga disertai pembandingan dengan Plagiarism Checker X Free.
Tahap ketujuh adalah analisis data eksperimen, data hasil pencatatan eksperimen dibandingkan dan dihitung selisih nilai antar tipe n-gram. Nilai yang dijadikan pembanding adalah selisih prosentase kemiripan antara instrumen dan Plagiarism Checker X Free dan waktu proses deteksi dari tiap metode parsing.

Tahap kedelapan adalah perumusan kesimpulan, dari proses analisis data yang menyajikan tabulasi perbandingan hasil eksperimen ditariklah sebuah kesimpulan. Rumusan kesimpulan dikaitkan dengan rumusan masalah agar dapat menjawab pertanyaan yang ada dalam rumusan masalah.

Tahap terakhir adalah dokumentasi, walau ini dijelaskan pada bagian terakhir namun proses ini bukan proses yang dimulai paling akhir. Dokumentasi dilakukan disetiap akhir tiap tahapan atau proses. Kemudian setelah analisa selesai baru dilakukan penggabungan hasil dokumentasi untuk penyusunan laporan hasil penelitian tesis.

3. HASIL DAN PEMBAHASAN



Gambar. 4. Tampilan Instrumen Uji

Gambar 4 menunjukkan tampilan instrumen uji yang dimana disetiap tombolnya terdapat fungsi hasil implementasi algoritma parsing dari beberapa tipe n-gram dan dengan rabin-karp sebagai algoritma string matching nya. Hasil berada disisi sebelah kiri tombol dengan menampilkan prosentase kemiripan, jumlah kejadian perbandingan yang sama dengan total perbandingan, dan waktu proses dalam satuan second. Dokumen yang dibandingkan dipilih sesuai pilihan yang ada di sisi sebelah kiri. Pada bagian ini ada dua tempat input pilihan dokumen. Bagian atas untuk dokumen asal (tercurigai) dan bagian bawah merupakan dokumen target. Saat pengguna memilih dokumen uji dengan id 1 maka isi dokumen akan tampil disebelah kanan.

Proses pengujian yang kamilakukan adalah memilih dokumen asal, kemudian memilih dokumen target. Dokumen akan muncul langsung pada kota sebelah kanan. Sehingga proses load dari database diluar dari proses perbandingan, waktu proses yang dibandingkan nanti tidak termasuk waktu untuk proses load dari database. Setelah kedua dokumen di load ke instrumen, maka berikutnya kami menekan tombol pertama untuk 6-gram char dan catat nilai prosentase similarity dan waktu proses yang nampak pada sebelah kiri. Proses tersebut kami lakukan berulang pada tombol berikutnya (6-gram word, oneskipbigram dan dotskipbigram). Berikut tabel 1 yang merupakan hasil pengujian terhadap keempat jenis n-gram.

Tabel 1. Hasil Pengujian Jenis N-Gram

No	Jenis N-gram	Dokumen		Ketepatan (%)			Kecepatan (s)
		Sumber	Target	Similarity Instrumen	Similarity PCX Free	Selisih Similarity	
1	n-gram char	Dok 1 (8 hal)	Dok 2 (12 hal)	47	24	23	1,34
2	n-gram word	Dok 1 (8 hal)	Dok 2 (12 hal)	21	24	3	0,41
3	one-skip-gram	Dok 1 (8 hal)	Dok 2 (12 hal)	20	24	4	0,52
4	dot-skip-gram	Dok 1 (8 hal)	Dok 2 (12 hal)	18	24	6	0,35
5	n-gram char	Dok 1 (8 hal)	Dok 3 (14 hal)	80	70	10	1,25
6	n-gram word	Dok 1 (8 hal)	Dok 3 (14 hal)	68	70	2	0,34
7	one-skip-gram	Dok 1 (8 hal)	Dok 3 (14 hal)	53	70	27	0,43
8	dot-skip-gram	Dok 1 (8 hal)	Dok 3 (14 hal)	52	70	28	0,33
9	n-gram char	Dok 1 (8 hal)	Dok 4 (15 hal)	69	69	0	1,26
10	n-gram word	Dok 1 (8 hal)	Dok 4 (15 hal)	55	69	14	0,34
11	one-skip-gram	Dok 1 (8 hal)	Dok 4 (15 hal)	41	69	28	0,36
12	dot-skip-gram	Dok 1 (8 hal)	Dok 4 (15 hal)	38	69	31	0,38
13	n-gram char	Dok 2 (12 hal)	Dok 3 (14 hal)	91	86	5	1,88
14	n-gram word	Dok 2 (12 hal)	Dok 3 (14 hal)	86	86	0	0,54
15	one-skip-gram	Dok 2 (12 hal)	Dok 3 (14 hal)	62	86	24	0,59
16	dot-skip-gram	Dok 2 (12 hal)	Dok 3 (14 hal)	60	86	26	0,55
17	n-gram char	Dok 2 (12 hal)	Dok 4 (15 hal)	61	46	15	1,68
18	n-gram word	Dok 2 (12 hal)	Dok 4 (15 hal)	45	46	1	0,47
19	one-skip-gram	Dok 2 (12 hal)	Dok 4 (15 hal)	39	46	7	0,51
20	dot-skip-gram	Dok 2 (12 hal)	Dok 4 (15 hal)	40	46	6	0,44
21	n-gram char	Dok 3 (14 hal)	Dok 4 (15 hal)	49	36	13	2,00
22	n-gram word	Dok 3 (14 hal)	Dok 4 (15 hal)	27	36	9	0,65
23	one-skip-gram	Dok 3 (14 hal)	Dok 4 (15 hal)	24	36	12	0,62
24	dot-skip-gram	Dok 3 (14 hal)	Dok 4 (15 hal)	23	36	13	0,53

Setelah hasil pengujian didapatkan maka proses selanjutnya adalah melakukan analisa perbandingan terhadap masing-masing jenis n-gram. Dari tabel 1 kemudian disusun ulang sehingga tampak seperti pada Tabel 2. Tabel 2 tersebut menyajikan perbandingan ketepatan hasil penghitungan similarity dari instrumen uji. Ketepatan merupakan selisih prosentase similarity dari instrumen dengan similarity dari software Plagiarism Checker X Free. Sehingga definisi tepat dalam ketepatan dalam perbandingan ini adalah nilai similarity hasil dari instrumen yang bernilai paling kecil (selisih paling kecil di antara keempat fungsi n-gram).

Tabel 2. Hasil Pengujian Jenis N-Gram

No	Pengujian Dokumen	Ketepatan - Selisih Similarity (%)			
		n-gram char	n-gram word	one-skip-gram	dot-skip-gram
1	Dok1→Dok2	23	3	4	6
2	Dok1→Dok3	10	2	27	28
3	Dok1→Dok4	0	14	28	31
4	Dok2→Dok3	5	0	24	26
5	Dok2→Dok4	15	1	7	6
6	Dok3→Dok4	13	9	12	13

Selain membandingkan ketepatan kami juga melakukan perbandingan kecepatan. Kecepatan diperoleh dari penghitungan selisih cpu tick yang didapatkan sebelum fungsi dijalankan dan saat fungsi berhasil mendapatkan nilai atau selesai melakukan eksekusi perintah. Penghitungan dimulai saat tombol pada instrumen di tekan. Berikut Tabel 3 yang menyajikan perbandingan kecepatan eksekusi fungsi ngram.

Tabel 3. Hasil Pengujian Jenis N-Gram

No	Pengujian Dokumen	Kecepatan (s)			
		n-gram char	n-gram word	one-skip-gram	dot-skip-gram
1	Dok1→Dok2	1,34	0,41	0,52	0,35
2	Dok1→Dok3	1,25	0,34	0,43	0,33
3	Dok1→Dok4	1,26	0,34	0,36	0,38
4	Dok2→Dok3	1,88	0,54	0,59	0,55
5	Dok2→Dok4	1,68	0,47	0,51	0,44
6	Dok3→Dok4	2,00	0,65	0,62	0,53

4. KESIMPULAN

Kesimpulan yang dapat diambil dari fakta hasil pengujian yang telah dilakukan adalah:

1. Pengujian ketepatan hasil similarity dari keempat tipe ngram menunjukan jika n-gram word memiliki ketepatan akurasi yang paling baik, selisih similarity lebih dari 10% hanya satu kali dari enam kali pengujian.
2. Pengujian kecepatan waktu proses dari keempat tipe ngram menunjukan jika dot-skip-gram yang paling cepat, setelah itu baru n-gram word yang paling cepat berikutnya. Perbedaan keduanya tidak terlalu signifikan.
3. N-gram word merupakan jenis n-gram yang paling sesuai untuk digunakan karena paling akurat diantara lainnya dan dengan kecepatan yang tidak terlau jauh dari yang paling cepat menurut pengujian diatas.

5. SARAN

Berikut ini beberapa saran yang dirumuskan bagi peneliti selanjutnya agar dapat ditanggapi atau dapat dilakukan tindak lanjut dari hasil penelitian ini:

1. Perlu dilakukan penelitian lanjut mengenai pembandingan dengan menggunakan bahasa pemrograman yang lain, agar diketahui apakah perbandingan kecepatan proses memiliki nilai yang sama, serta seberapa pengaruh kemampuan bahasa pemrograman mempengaruhi hasilnya tersebut.
2. Dapat dilakukan pengkajian ulang terhadap masing-masing fungsi apakah sudah sesuai dengan aturan tahapan/algoritma parsing tiap jenis n-gram.
3. Bisa dilakukan pengujian dengan dokumen yang lebih besar lagi (misal: satu dokumen penuh). Apakah perbandingannya akan sama atau berbeda agar dapat diketahui penggunaan tipe n-gram yang sesuai untuk panjang dokumen yang berbeda.
4. Menggunakan jenis n-gram atau teknik lain untuk dilakukan perbandingan agar proses deteksi plagiat bisa lebih baik lagi.

DAFTAR PUSTAKA

[1] Nugroho, E., 2011, Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp, *Skripsi*, Program Studi Ilmu Komputer, Universitas Brawijaya, Malang.

[2] Novian, D., Abdillah, T., Tuloli, M. S., Yassin, R. M. T., Aplikasi Pendeteksian Plagiat Pada Karya Ilmiah Menggunakan Algoritma Rabin-Karp, *Laporan Penelitian Pengembangan Fakultas Dan Keilmuan Dana BOPTN Tahun Anggaran 2012*, Pengembangan Fakultas Dan Keilmuan, Universitas Negeri Gorontalo, Gorontalo.

[3] Salmuasih, 2013, Perancangan Sistem Deteksi Plagiat pada Dokumen Teks dengan Konsep Similarity Menggunakan Algoritma Rabin-Karp, *Skripsi*, Jurusan Teknik Informatika, STMIK Amikom Yogyakarta, Yogyakarta.

[4] Minaei, B., Niknam, M., 2016, An N-Gram Based Method For Nearly Copy Detection In Plagiarism Systems, *Proceedings of the 8th annual meeting of the Forum for Information Retrieval Evaluation*, Kolkata, India, 7-10 Desember 2016.

[5] Ehsan, N., Tompa, F, W., Shakery, A., 2016, Using a Dictionary and n-gram Alignment to Improve Fine-grained Cross-Language Plagiarism Detection,*Proceedings of the 2016 ACM Symposium on Document Engineering*, Vienna, Austria, 13-16 September 2016.

[6] Nguyen, L. T., Toan, N. X., Dien, D., 2016, Vietnamese plagiarism detection method. *Proceedings of the Seventh Symposium on Information and Communication Technology*, Ho Chi Minh City, Viet Nam, 8-9 Desember 2016.

[7] Kuta, M., Kitowski, J., 2014, Optimisation of Character n-gram Profiles Method for Intrinsic Plagiarism Detection, *Proceedings of 13th ICAISC: Artificial Intelligence and Soft Computing*, Zakopane, Poland, 1-5 Juni 2016.

[8] Bensalem, I., Rosso, P., Chikhi, S., 2014, Intrinsic Plagiarism Detection using N-gram Classes,*Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (EMNLP), Doha, Qatar, 25-29 Oktober 2014.

[9] Palkovskii, Y., Belov, A., 2014, Developing High-Resolution Universal Multi-Type N-Gram Plagiarism Detector. *CLEF 2014: Conference and Labs of the Evaluation Forum Information Access Evaluation meets Ultilinguality, Multimodality, and Interaction*, Sheffield, United Kingdom, 15-18 September 2014.

[10] Sugianto, S. A., Liliana., Rostianingsih, S., 2013. Pembuatan Aplikasi Predictive Text Menggunakan Metode N-Gram-Based. *Jurnal Infra Universitas Kristen Petra*, Vol. 11, No. 2, Hal. 119-124.

[11] Guthrie, D., Allison, B., Liu, W., Guthrie, L., Wilks, Y., 2006, A Closer Look at Skip-gram Modelling, *https://homepages.inf.ed.ac.uk/ballison/pdf/lrec_skipgrams.pdf*, didownload Sabtu, 13 Nopember 2017.

[12] Atmopawiro, A., 2006, Pengkajian dan Analisis Tiga Algoritma Efisien RabinKarp, Knuth-Morris-Pratt dan Boyer-Moore dalam Pencarian Pola dalam Suatu Teks, *Skripsi*, Program Studi Teknik Informatika, Institut Teknologi Bandung, Bandung.