PERBANDINGAN PENGUNAAN STEMMING PADA DETEKSI KEMIRIPAN DOKUMEN MENGGUNAKAN METODE RABIN KARP DAN JACCARD SIMILARITY

Adji Sukmana[1)], Kusrini[2)], Andi Sunyoto[3)]

[1,2,3)] Magister Teknik Informatika, Universitas AMIKOM Yogyakarta
Jl. Ring Road Utar, Condong Catur, Depok, Sleman, Yogyakarta 55281
Email : adji.sukmana@gmail.com[1)], kusrini@amikom.ac.id [2)], andi@amikom.ac.id[3)]

Abstrak

Kemiripan Dokumen dapat digunakan dalam mencari informasi yang sama pada dua atau lebih dokumen. Kemampuan dalam mencari kemiripan ini dapat diimplementasikan dalam pendeteksi plagiarisme pada jurnal maupun karya ilmiah.

Praktek tindakan plagiarisme sering terjadi di dunia akademis, baik plagiarisme dalam hal penyelesaian tugas maupun penyusunan karya ilmiah. Untuk meminimalkan tindakan plagiarisme, diperlukan suatu sistem untuk menilai atau mengukur seberapa banyak kemiripan dalam sebuah dokumen. Algoritma Rabin Karp adalah salah satu metode document fingerprinting yang digunakan untuk mendeteksi kemiripan antar teks dokumen dengan menggunakan teknik hashing, sedangkan untuk menggambarkan tingkat kemiripan antara dokumen dapat diukur dengan menggunakan Jaccard Similarity.

Penelitian ini membandingkan antara Rabin karp murni dan rabin karp dengan penambahan stemming nazief adriani. Dengan menggunakan dokumen uji yang terdiri dari 100 kata, 75 kata, 50 kata dan 25 kata dapat disimpulkan bahwa stemming nazief adriani dapat mempercepat waktu eksekusi rabin karp lebih cepat dengan hasil similarity yang hampir sama.

Kata kunci: *Rabin Karp, Jaccard Coeficient, Stemming.*

1. Pendahuluan

Kemiripan dokumen (document similarity) dapat digunakan sebagai alat pencarian informasi lain yang sejenis, sehingga dapat mempersingkat waktu. Kemampuan pencarian kemiripan dokumen biasanya diimplementasikan pada sebuah artikel berita dan jurnal. Salah satu contoh pemanfaatan dari kemiripan dokumen dapat digunakan sebagai deteksi plagiarism.

Maraknya tindakan plagiarisme di dunia akademis, baik plagiarisme dalam hal penyelesaian tugas maupun penyusunan karya ilmiah dapat menjadi sorotan penting yang harus diminimalisir. Oleh karena itu, banyak penelitian yang membahas mengenai sistem pendeteksi plagiarisme. Menurut Stein dkk [1], salah metode untuk mendeteksi plagiarisme dapat menggunakan metode dokumen fingerprinting yang hanya melakukan pencocokan pola string, pendeteksian plagiat ini tidak memperhatikan adanya penulisan sumber rujukan. Algoritma yang digunakan yaitu algoritma Rabin Karp dengan perhitungan jacard similarity.

Duplikasi dokumen, deteksi plagiat, dan pencocokan string telah banyak dibahas pada penelitian-penelitian sebelumnya. Algoritma yang digunakan diantaranya Winnowing, Smith Waterman, Boyer Moore, dan Rabin Karp [2][3]. Namun sebagian besar tanpa menggunakan preprocessing, sehingga berpengaruh pada akurasi similarity. Pendeteksian plagiat menggunakan konsep similarity atau kemiripan dokumen merupakan salah satu cara untuk mendeteksi copy&paste plagiarism dan disguised plagiarism. Menggunakan algoritma Rabin Karp yang menerapkan metode fingerprinting yang hanya melakukan pencocokan pola string, pendeteksian plagiat ini tidak memperhatikan adanya penulisan sumber rujukan. Pada sistem deteksi ini akan diaplikasikan text mining untuk tahap preprocessing dan algoritma Rabin Karp untuk string matching. Algoritma Rabin Karp adalah algoritma multiple pattern search yang sangat efisien untuk mencari string dengan pola banyak [3].

Banyak yang membahas mengenai algoritma rabin karp dengan menggunakan stemming, tetapi tidak melakukan perbandingan apakah stemming tersebut mempengaruhi algoritma rabin karp atau tidak. Penelitian ini melakukan perbandingan dengan menghitung kecepatan dan juga hasil similarity dari algoritma rabin karp tanpa menggunakan stemming dan dengan menggunakan stemming. Stemming yang digunakan disini adalah stemming nazief adriani yang memiliki akurasi lebih baik daripada stemming lainnya [4]. Proses stemming menggunakan teknik indexing sehingga tidak berpengaruh terhadap database.

Tinjauan Pustaka

Penelitian yang dilakukan oleh Satia dan Saerul [5] melakukan analisa kemiripan dalam sebuah dokumen sepeti tugas akhir dan makalah ilmiah dengan menerapkan Algoritma Rabin Karp dalam mendeteksi plagiarisme karena algoritma ini terbukti efektif untuk membandingkan patern-patern yang ada pada sebuah

essai dengan menggunakan fungsi hashing yang dapat menemukan bentuk bentuk / pola dalam teks, untuk lebih meningkatkan keakuratan dalam proses penemuan pola pada sebuah teks kemudian digunakan algoritma Steeming Najief Andriani yang dapat menemukan kata-kata yang setara yang memiliki persamaan kata dasar yang sama. Tetapi kelemahan pada penelitian ini tidak melakukan analisis perbandingan yang dilihat dari segi waktu eksekusi.

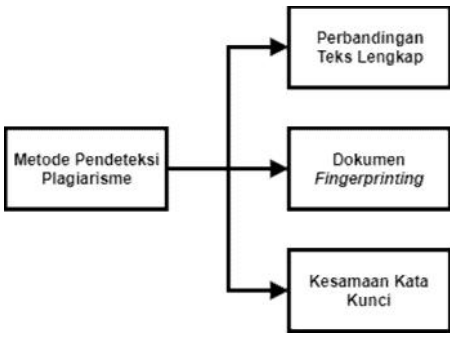
Penelitian oleh Hafiz [6] menggabungkan stemming Adriani & Nazief dan Algoritma Similarity yang digunakan untuk pengecekan judul dan abstraksi thesis, apakah judul dengan tema tersebut sudah pernah diajukan atau belum. Stemming berfungsi untuk mengumpulkan index judul dan abstraksi thesis sebagai database sehingga dapat dilakukan pengecekan dengan menggunakan algoritma similarity.

Penelitian Jayanta dkk [7] menghasilkan penggunaan algoritma Rabin-Karp menghasilkan ukuran self-plagiarism yang tinggi pada pengukuran yang menggunakan bagian "pendahuluan". Jumlah kata sangat mempengaruhi hasil pengukuran dengan algoritma Rabin-Karp.

Landasan Teori

Kemiripan dokumen (document similarity) dapat digunakan sebagai alat pencarian informasi lain yang sejenis, sehingga dapat mempersingkat waktu. Kemampuan pencarian kemiripan dokumen biasanya diimplementasikan pada sebuah artikel berita dan jurnal [8].

Metode Pendeteksi Plagiarisme di bagi menjadi tiga bagian yaitu metode perbandingan teks lengkap, metode dokumen fingerprinting, dan metode kesamaan kata kunci [1] yang dapat dilihat pada gambar 3.



Gambar 1. *Metode pendeteksi plagiarisme*

Berikut ini penjelasan dari masing-masing metode dan algoritma pendeteksi plagiarisme. Ketiga metode tersebut adalah :

1. Perbandingan Teks Lengkap
   Metode ini di terapkan dengan membandingkan semua isi dokumen. Dapat diterapkan untuk dokumen yang besar. Pendekatan ini membutuhkan waktu yang lama tetapi cukup efektif, karena kumpulan dokumen yang diperbandingkan adalah dokumen yang di simpan pada penyimpanan lokal. Metode perbandingan teks lengkap tidak dapat diterapkan untuk kumpulan dokumen yang tidak terdapat pada dokumen lokal. Algoritma yang digunakan pada metode ini adalah algoritma brute force, algoritma edit distance, algoritma boyer moore dan algoritma lavenshtein distance Brute Force, Edit Distance, dan Smith.
2. Dokumen Fingerprinting
   Dokumen fingerprinting merupakan metode yang digunakan untuk mendeteksi keakuratan salinan antar dokumen, baik semua teks yang terdapat di dalam dokumen atau hanya sebagian teks saja. Prinsip kerja dari metode dokumen fingerprinting ini adalah dengan menggunakan teknik hashing. Teknik hashing adalah sebuah fungsi yang mengkonversi setiap string menjadi bilangan. Algoritma pada pendekatan ini adalah algoritma Rabin Karp, Winnowing dan Manber.
3. Kesamaan Kata Kunci.
   Prinsip dari metode ini adalah mengekstrak kata kunci dari dokumen dan kemudian dibandingkan dengan kata kunci pada dokumen yang lain. Pendekatan yang digunakan pada metode ini adalah teknik dot.

Case folding dan Tokenizing Case folding merupakan sebuah langkah yang merubah huruf bentuk huruf yang semula UPPER CASE ke dalam bentuk lowercase sedangkan proses tokenizing adalah proses pemisahan kalimat kedalam bentuk kata [9]

Stemming Stemming merupakan suatu proses yang terdapat dalam sistem IR yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (root word) dengan menggunakan aturan-aturan tertentu. Sebagai contoh, kata bersama, kebersamaan, menyamai, akan distem ke root word nya yaitu "sama" [10].

Algoritma Rabin-Karp diciptakan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 yang menggunakan fungsi hashing untuk menemukan pattern di dalam string teks[11]. Algoritma Rabin-Karp merupakan versi awal dari fingerprinting dengan menggunakan metode k-gram yang diciptakan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987[12]. Pada dasarnya, Algoritma Rabin Karp menghitung nilai hash untuk pattern dan setiap k-gram dari teks yang akan dibandingkan. Jika nilai hash tidak sama, maka akan dihitung nilai hash untuk k-gram selanjutnya secara berurutan. Jika nilai hash sama, maka dilakukan perbandingan antara pattern dan k-gram.

Berikut ini merupakan langkah-langkah dalam mencocokan kata dengan algoritma Rabin-Karp [13] :

1. Hilangkan tanda baca dan ubah teks sumber menjadi kata-kata tanpa huruf kapital.
2. Tentukan panjang dari teks sumber yang ingin dicari (n) dan kata yang ingin dicari (m)
3. Mencari nilai hash dari teks sumber dan kata yang ingin dicari menggunakan fungsi hash yang telah ditentukan
4. Lakukan iterasi dari indeks i=0 sampai i=n-m+1 untuk membandingkan nilai hash dari kata yang ingin dicari dengan nilai hash dari teks sumber pada indeks i sampai dengan i+m-1. Jika nilai hash sama, maka akan diperiksa lebih lanjut apakah kata yang dicari sama dengan bagian teks dari sumber pada indeks i sampai dengan i+m-1. Jika sama, maka telah ditemukan kata yang cocok. Jika tidak maka dilanjutkan dengan membandingkan nilai hash dari kata yang dicari dengan nilai hash teks sumber pada indeks berikutnya.

Rabin Karp merepresentasikan setiap karakter ke dalam bentuk desimal digit (digit radix-d) $\Sigma = \{0, 1, 2, 3, \cdots, d\}$, dimana $d = |\Sigma|$. Sehingga didapat masukan string k berturut - turut sebagai perwakilan panjang k desimal. Karakter string 31415 sesuai dengan jumlah desimal 31,415. Kemudian pola p di-hash menjadi nilai desimal dan string direpresentasikan dengan penjumlahan digit-digit angka menggunakan aturan Horner's, misal (Cormen, 2001):

{ A, B, C, ..., Z } → { 0, 1, 2, ..., 26 }
AJI → 1 + 18 + 8 = 28
FANI → 13 + 20 + 17 +8 = 56

Untuk pola yang panjang dan teks yang besar, algoritma ini menggunakan operasi mod, setelah dikenai operasi mod q, nilainya akan menjadi lebih kecil dari q. Rumus matematis dari algoritma rabin karp yang ditunjukkan pada persamaan 1 :

$$T_{s+1} = (d\,(t_s - T[s+1]\,h) + T[s+m+1]) \bmod q \quad \text{.....(1)}$$

Dimana

s	nilai desimal dengan panjang m dari substring T [s + 1 .. s + m], untuk s = 0, 1, ..., n - m
ts+1	nilai desimal selanjutnya yang dihitung dari ts
d	radix desimal (bilangan basis 10)
h	d m-1
n	panjang teks
m	panjang pola
q	nilai modulo

Teknik Rolling Hash pada awalnya digunakan pada algoritma Rabin-Karp. Setiap karakter di dalam dokumen teks diubah (encode) menjadi nilai array bilangan bulat, sehingga nilai masukan yang awalnya berupa karakter menjadi fungsi hash berupa angka. Perhitungan operasi modulo digunakan agar tidak mempersulit sistem menghitung dalam jumlah banyak, selama nilai modulo yang digunakan tidak terlalu besar pula.

Persamaan teknik rolling hash [14] ditunjukkan pada persamaan 2 :

$$h(k) = (k[0]\,b^{L-1} + k[1]\,b^{L-2} + k[2]\,b^{L-3} + k[3]\,b^{L-4} + \ldots + k[L-1]\,b^{0}) \bmod m \quad \text{.....(2)}$$

Untuk menghitung hash lanjutan ditunjukkan pada persamaan 3.

$$h(S_{i+1}) = b(h(S_i) - b^{L-1} S[i]) + S[i+L] \bmod m \quad \text{.....(3)}$$

Dimana

b	Nilai bilangan basis (10)
k	Nilai ASCII karakter
h(k)	Nilai hash
m	Nilai bilangan prima (10007)
L	Banyaknya karakter yang di hashing
S(i)	Nilai hash awal
S(i+1)	Nilai hash berikutnya

Text mining adalah salah satu bidang khusus dari data mining.[15] mendefinisikan text mining sebagai suatu proses menggali informasi dimana seorang user berinteraksi dengan sekumpulan dokumen menggunakan tools analisis yang merupakan komponen-komponen dalam data mining. Tujuan dari text mining adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen.

Dalam memberikan solusi, text mining mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti Data mining, Information Retrieval (IR), Statistic and Mathematic, Machine Learning, Linguistic, Natural Languange Processing (NLP), dan Visualization. Kegiatan riset untuk text mining antara lain ekstraksi dan penyimpanan teks, preprocessing akan konten teks, pengumpulan data statistik dan indexing, dan analisa konten [16]. Tahapan dalam text mining meliputi tokenizing, filtering, stemming, tagging dan analyzing [17].

Stemming merupakan bagian dari proses Information Retrieval (IR), yang mengubah beberapa kata ke bentuk kata dasarnya sebelum dilakukan pengindeksan. Contoh, kata dibaca, membaca, pembaca, akan diubah ke kata dasarnya, yaitu “baca” [18].
Stemming yang akan digunakan dalam sistem pendeteksi plagiarisme ialah steming najif andriani yang akan digunakan untuk preprossing teks sebelum kemudian di cari kemiripan kata algoritma najif andriani memiliki

kelebihan dari segi prosentasi keakuratan (presisi) lebih besar dibanding dengan algoritma stemming porter [4]. Tahapan pada algoritma najief dan andriani :

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tesebut adalah root word. Maka algoritma berhenti.
2. Inflection Suffixes(“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa particles (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possesive Pronouns(“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus Derivation Suffixes(“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
    a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
    b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus Derivation Prefix. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
    a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
    b. For i = 1 to 3, tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Melakukan Recoding.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai root word. Proses selesai.

2. Pembahasan

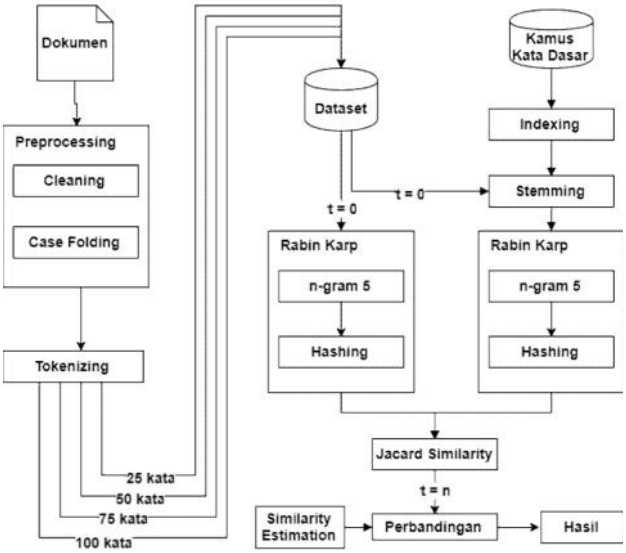
Pada penelitian ini menerapkan metode fingerprinting dengan algoritma yang dipakai adalah Rabin Karp dan Jaccard similarity. Sebelum melakukan penelitian harus dipersiapkan terlebih dahulu rancangan arsitektur untuk keperluan pengujian.

Rancangan arsitektur basis data yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 1.

Gambar 2. *Arsitektur basis data*

Rancangan basisdata pada gambar 1 diatas terdiri dari 2 buah tabel yaitu tabel dataset dan tabel dictionary. Tabel dataset digunakan untuk menyimpan dokumen uji yang berupa kalimat dengan kondisi tertentu yang akan di jelaskan pada arsitektur sistem. Tabel dictionary berisi kamus kata dasar yang didapatkan dari penelitian Nazief Adriani yang digunakan untuk proses stemming.

Dibawah ini adalah rancangan keseluruhan sistem yang ditunjukkan pada gambar 2.



Gambar 3. *Arsitektur sistem*

Dokumen uji sebelum masuk ke database dilakukan tahapan preprocessing dan tokenizing untuk keperluan pengujian. Pada tahapan preprocessing terdapat proses cleaning dan case folding, yang berarti dokumen akan dihapus karakter selain huruf angka dan spasi, kemudian mengubah huruf besar menjadi huruf kecil.

Untuk menentukan model terbaik untuk mendeteksi plagiarisme berdasarkan pengukuran kemiripan dokumen, dilakukan beberapa alur proses dan skenario uji. Dokumen teks yang akan diuji dalam kasus ini dirujuk dari penelitian yang dilakukan oleh Salmuasih [3] yang terbagi menjadi empat macam dokumen teks dengan beberapa dimodifikasi. Sehingga pada tahapan toknizing di dihasilkan file uji yang berupa empat buah dokumen yang masing-masing telah dilakukan penghapusan sebanyak 25%. Berikut adalah informasi dokumen yang akan digunakan untuk pengujian yang ditunjukkan pada Tabel 1.

Tabel 1. *Tabel Dokumen Uji*

Dokumen	Kata
Dokumen 1	100
Dokumen 2	75
Dokumen 3	50
Dokumen 4	25

Dari tabel 1 diatas dapat disimpulkan bahwa perbandingan similarity estimation masing – masing dokumen ditunjukkan pada tabel 2.

Tabel 2. *Tabel Similarity Estimation*

Dokumen	Dokumen	Similarity Estimation
Dokumen 1	Dokumen 1	100%
Dokumen 1	Dokumen 2	75%
Dokumen 1	Dokumen 3	50%
Dokumen 1	Dokumen 4	25%

Setelah dataset terisi maka langkah selanjutnya yaitu dilakukan 2 pengujian, yaitu dataset terhadap algoritma Rabin Karp tanpa stemming dan dengan stemming. Rabin Karp disini menggunakan gram karakter yang berjumlah 5. Stemming yang digunakan yaitu stemming Nazief Adriani yang menggunakan database sebagai kamus kata dasar.

Pada penelitian ini dilakukan tahapan indexing kamus kata dasar dari database sehingga kamus tersebut disimpan kedalam array kemudian dilakukan flip array. Dengan begitu index array akan bertukar yang semula index menjadi value dan sebaliknya. Proses ini penting dilakukan karena jika tanpa menggunakan flip array maka pencarian akan lebih lama.

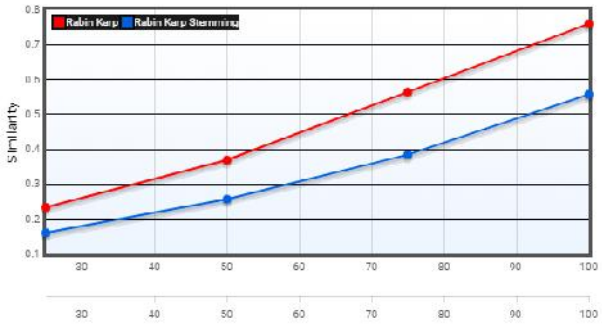
Selanjutnya yaitu menghitung waktu eksekusi dan similarity yang dihasilkan kedua pengujian. Waktu eksekusi dihitung dari setelah data disambil dari database dan berakhir ketika perhitungan similarity selesai. Perhitungan similarity disini menggunakan Jaccard Similarity.

Berikut hasil dari pengujian yang dilakukan yang ditunjukkan pada tabel 3 dan tabel 4.

Tabel 3. *Tabel Perbandingan Waktu Eksekusi*

Dokumen	Similarity Estimation	Time Second (s)	
		Rabin Karp	Rabin Karp Stemming
Dokumen 1	100	0.7594	0.5574
Dokumen 2	75	0.5643	0.3845
Dokumen 3	50	0.3694	0.2573
Dokumen 4	25	0.2338	0.1610

Tabel 3 diatas menunjukkan bahwa algoritma rabin karp dengan menggunakan stemming lebih cepat dibandingkan rabin karp tanpa stemming. Untuk lebih jelasnya tabel 3 diatas disajikan dalam bentuk grafik yang ditunjukkan pada gambar 3.



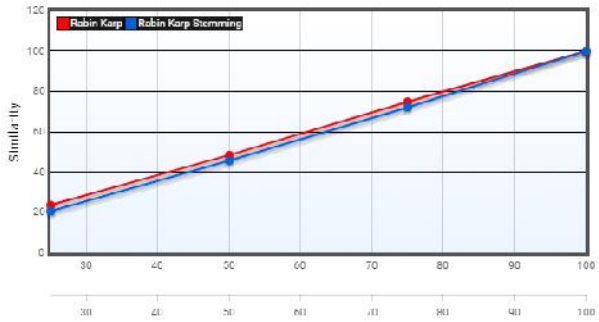
Gambar 4. *Grafik Perbandingan Waktu Eksekusi*

Selain dari segi waktu, dilakukan pengujian juga pada segi similarity, yang ditunjukkan pada tabel .

Tabel 4. *Tabel Perbandingan Similarity*

Dokumen	Similarity Estimation	Similarity Percent (%)	
		Rabin Karp	Rabin Karp Stemming
Dokumen 1	100	100	100
Dokumen 2	75	74.87	72.09
Dokumen 3	50	48.39	45.69
Dokumen 4	25	23.60	20.59

Tabel 4 diatas menunjukkan bahwa algoritma rabin karp dengan menggunakan stemming dan rabin karp tanpa stemming memiliki similarity yang hampir sama. Untuk lebih jelasnya tabel 4 diatas disajikan dalam bentuk grafik yang ditunjukkan pada gambar 4.



Gambar 5. *Grafik Perbandingan Similarity*

3. Kesimpulan

Dari hasil pengujian dan analisis maka dapat disimpulkan beberapa hal sebagai berikut: Algoritma Rabin Karp dengan menggunakan stemming Nazief Adriani dapat mempercepat waktu eksekusi dengan hasil similarity yang hampir sama.

Daftar Pustaka

[1] B. Stein, s. Meyer zu eissen, near similiarity search and plagiarismanalysis, 29th annual conference of the german classification society(gfk1), magdeburg, isdn 1431-8814, pp. 430-437, 2006.
[2] Obed kharisman, budi susanto, dan sri suwarno, implementasi algoritmawinnowing untuk mendeteksi kemiripan pada dokumenteks,informatika; vol. 9, n0. 1, april 2013, pp 73-81.
[3] Salmuasih dan sunyoto, a., 2013, implementasi algoritma rabin karp untukpendeteksian plagiat dokumen teks menggunakan konsep similarity, proc. Ofseminar nasional aplikasi teknologi informasi, yogyakarta, 15 juni 2013, 23-28.
[4] Agusta, ledy, 2009, perbandingan algoritma stemming porter denganalgoritma nazief & adriani untuk stemming dokumen teksbahasa indonesia, konferensi nasional sistem dan informatika 2009; bali, november 14, 2009.
[5] Suhada, satia dan saeful bahri, 2017, implementasi algoritma rabin karp dan stemming najief andriani untuk deteksi plagiarisme dokumen, swabumi, vol.5 maret 2017, pp. 84-89
[6] Pramudita, hafiz ridha. 2014. Penerapan algoritma stemming nazief & adriani dan similarity pada penerimaan judul thesis. Jurnal ilmiah dasi vol. 15 no. 04 desember 201, hlm 15 – 19.
[7] Jayanta, halim mahfud dan titin pramiyati. 2017. Analisis pengukuran self plagiarism menggunakan algoritma rabin-karp dan jaro-winkler distance dengan stemming tala. Seminar nasional teknologi informasi dan multimedia 2017.
[8] Sugiyamta, 2015. Sistem deteksi kemiripan dokumen dengan algoritma cosine similarity dan single pass clustering, dinamika informatika – vol.7 no. 2
[9] N. Made, a. Lestari, i. K. Gede, d. Putra, a. A.ketut, and a. Cahyawan, “personality typesclassification for indonesian text in partnerssearching website using naïve bayesmethods,” vol. 10, no. 1, pp. 1–8, 2013.
[10] L. Agusta, u. Kristen, and s. Wacana,“perbandingan algoritma stemming porter dengan algoritma nazief & adriani untuk stemming dokumen teksbahasa indonesia,” pp. 196–201,2009.
[11] B. Gipp and n. Meuschke,(2011)."citation pattern matching algorithms for citation-based plagiarism detection: greedy citation tiling, citation chunking and longest common citation sequence," proceedings of the 11th acm symposium on document engineering (doceng2011), pp. 249—258.
[12] Schleimer, s., wilkerson, d.s., aiken, a.,(2003). Winnowing: local algorithms for document fingerprinting, proc. Of the 2003 acm sigmod international conference on management of data, new york, 9-12 juni 2003, 76–85. 2003.
[13] Firdaus, hari b.(2003). Deteksi plagiat dokumen menggunakan algoritma rabin-karp. Jurnal ilmu komputer dan teknologi informasi, vol. Iii, no. 2.
[14] T. H. Cormen, c. E. Leiserson, r. L. Rivest and c. Stein, introduction to algorithms, usa: mit press, 2001
[15] R. Feldman and j. Sanger, the text mining handbook: advancedapproaches in analyzing unstructured data, cambridge: cambridgeuniversity press, 2007.
[16] C. Triawati, "metode pembobotan statistical concept based untukklastering dan kategorisasi dokumen berbahasa indonesia," institutteknologi telkom, bandung, 2009.
[17] R. J. Mooney, "cs 391l: machine learning text categorization,"university of texas, austin, 2006.
[18] Peng, f., ahmed, n., li, x., & lu, y. (2007), context sensitive stemming for web search. Domain specific nlp. Sunnyvale, california

Biodata Penulis

Adji Sukmana, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Informasi Sekolah Tinggi Manajemen Informatika dan Komputer Amikom Yogyakarta, lulus tahun 2016. Saat ini menempuh program pasca sarjana Teknik Informatika di Universitas AMIKOM Yogyakarta.

Kusrini, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Ilmu Komputer Universitas Gadjah Mada Yogyakarta, lulus tahun 2002. Memperoleh gelar (M.Kom) pada Jurusan Ilmu Komputer Universitas Gadjah Mada Yogyakarta di tahun 2006 dan memperoleh gelar Doktor pada jurusan Ilmu Komputer Universitas Gadjah Mada pada tahun 2010. Sejak 2003, bekerja sebagai Dosen Tetap Universitas Amikom Yogyakarta. Saat ini menjabat sebagai Direktur Program Pascasarjana dan Ketua Program Studi S2 Teknik Informatika.

Andi Sunyoto, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Informasi STMIK AMIKOM Yogyakarta, lulus tahun 2005. Program Pasca Sarjana (S2) Fakultas MIPA Jurusan Ilmu Komputer, Universitas Gadjah Mada Yogyakarta, lulus tahun 2007. Saat ini menjadi Dosen di STMIK AMIKOM Yogyakarta.