JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 1, Februari 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i1.1927
Hal 1-6

Deteksi Kemiripan Abstraksi Tugas Akhir Diploma Informatika Universitas AMIKOM Yogyakarta dengan Algoritma Rabin Karp

Siti Fatonah, Arifiyanto Hadinegoro, Anggit Dwi Hartanto

Ilmu Komputer, Informatika, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia
Email: [1] siti.fatonah@students.amikom.ac.id , [2] arifiyanto@amikom.ac.id, [3] anggit@amikom.ac.id

Submitted 09-01-2020; Accepted 15-01-2020; Published 15-02-2020

Abstrak

Praktek plagiarisme cukup sering terjadi baik di kalangan publik maupun akademis. Plagiarisme merupakan tindakan pelecehan, pencurian, perampasan, penerbitan, pernyatan atau menyatakan sebagai milik sendiri sebuah pikiran, ide tulisan atau ciptaan yang sebenarnya milik orang lain. Plagiat menjadi masalah yanag cukup signifikan dalam segi akademisi. Biasanya proses pengecekannya masih manual dan memakan waktu lama. Oleh karena itu, dibutuhkan aplikasi yang membantu proses pengecekan plagiat secara lebih efisien. Degan mencari kata dasar dalam abstraksi tugas akhir dengan menggunakan metode steamming, setelah itu dibentuk gram dan dicari nilai hash dengan algoritma Rabin Karp. Nilai hash tugas akhir yang akan diuji dibandingakan dengan nilai hash tugas akhir yang ada di dalam database. Dengan metode tersebut, diharapkan dapat mengetahui seberapa tingkat plagiat abstraksi tugas akhir mahasiswa diploma dengan mekanisme pengujian presentase kemiripan teks (similarity). Hasil perbandingan akan ditampilkan dalam bentuk persen, disimpan dalma databse, dan akan ditampilakan dari sistem dengan nilai persentase similarity terbesar sampai terkecil.

Kata Kunci: Plagiarisme, Algoritma rabin karp, Stemming, Abstraksi, Tugas akhir

Abstract

The practice of plagiarism in writing is quite common, both public and academic. Plagiarism is an act of abuse, theft, deprivation, publishing, statement, or declaring itself as a thought, idea, writing, or creation that actually belongs to someone else. For manual checking, whether a work is categorized as a work of plagiarism or a book is definitely a long time. Therefore, an application is needed that can detect the level of originality of a work more effectively, efficiently, and accountably. The application process is to find the basic words in the final assignment abstraction with the steaming method being tested. After obtaining the basic words of each word, formed gram and searched for hash values using the Rabin Karp algorithm. The final assignment abstraction hash value will be tested compared to the Final Task abstraction value in the database. It is expected to find out how much the level of plagiarism abstraction with the final task testing mechanism looking for presentation of final assignment similarity abstraction value compared to the final assignment abortion of the database and the comparison results in percent form, stored in the database and displayed in the system sequentially from the total presentation similarity of the biggest to the smallest.

Keywords: Plagiarism, Similarity, Rabin Karp Algorithm, Stemming, Abstraction, Final assignment.

1. PENDAHULUAN

Sekarang ini pencarian data dan informasi melalui internet menjadi semakin mudah dengan perkembangan teknologi informasi dan komunikasi yang semakin pesat. Perkembangan teknologi sekarang ini menimbulkan dampak positif dan negatif, misalnya penjiplakan atau plagiarisme terhadap karya orang lain.. Plagiarisme atau sering juga disebut plagiat adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri [1]. Plagiat dapat dianggap sebagai pencurian terhadap hak cipta orang lain dan dapat ditindak pidana. Plagiarisme sering dijumpai dalam sektor akademis maupun non-akademis. Dalam sektor akademis, plagiarisme dianggap sebagai tindak pidana serius karena dianggap pengambilan, gagasan ,ide atau pendapat orang lain. Plagiarime tidak sengaja juga bisa terjadi jika dalam pembuatan karya tulis lupa untuk menuliskan sumber pustaka dengan lengkap dan cermat. Plagiarisme belum begitu diketahui dan dipahami khusunya pada kalangan mahasiswa sehingga tingkat kejadiannya plagiarism ini masih cukup tinggi dan sulit dipantau. Di dunia akademik khususnya di Universitas masih besar sekali kemungkinan terjadi plagiarisme. Contohya dalam penulisan tugas akhir yang sering kali terjadi plagiarisme antar tugas akhir. Tindakan plagiarisme dalam instansi, sektor akademis secara perlahan dapat dicegah dan dihilangkan dengan melakukan pendeteksian plagiat secara manual ataupun dengan memaanfaatkan metode string. Namun pendeteksian secara manual memiliki masalah yang cukup besar yaitu tidak memungkinkan melakukan pendeteksian dokumen dengan membandingkan dengan dokumen lain dalam jumlah ratusan bahkan ribuan. Dengan demikian melakukan pendeteksian secara manual sangat tidak efektif. Metode kedua yaitu dengan melakukan pembandingan antara sumber dokumen asli atau yang sering disebut pencocokan string yang kemudian dapat dikembangakan menjadi sebuah aplikasi pendeteksi plagiarisme. Algoritma Rabin Karp yang dapat melakukan proses preprocessing terlebih dahulu. Algoritma tersebut dipilih karena Rabin Karp merupakan algoritma multiple pattern search yang efektif untuk pencarian string dengan pola yang banyak sehingga waktu dan keakuratan pencarian string menjadi lebih baik. Dan untuk mengetahui tingkat akurasi kemiripan teks antar abstraksi tugas akhir.

Pada penelitian yang dilakukan Sonawane Kiran Shivaji dan Prabhudeva S pada tahun 2017 yang berjudul Plagiarism Detection by using Karp-Rabin and String Matching Algorithm Together membahas mengenai deteksi plagiat menggunakan RabinKarp dan String Matching, yaitu membandingkan inputan berupa dokumen kemudian di bandingkan dengan yang ada di database. Pada penelitian ini terdapat masalah dengan algoritma Rabin-Karp yaitu, jika nilai hash sama namun untuk string yang berbeda dalam dokumen, hal ini dapat memberikan kebingungan pada sistem.. Dari uji coba pada penelitian tersebut dapat memberikan nilai hasil nilai presisi hingga 85% serta mampu meminimalkan persentase kegagalan sekitar 10%[2].

JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 1, Februari 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i1.1927
Hal 1-6

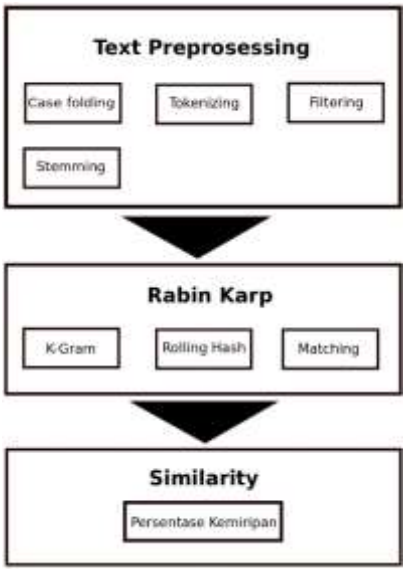
Pada penelitian yang dilakukan oleh Andysah Putera Utama Siahaan dkk tahun 2017 yang berjudul K-Gram As A Determinant Of Plagiarism Level In Rabin-Karp Algorithm membahas mengenai penentuan jumlah K-Gram dalam proses algoritma Rabin-Karp. Dalam algoritma Rabin-Karp penentuan tingkat plagiarisme didasarkan pada kesamaan nilai hash pada kedua dokumen. Setiap kata dalam dokumen akan membentuk K-Gram dengan panjang tertentu. Dari penelitian ini, uji coba nilai K-Gramnya adalah 10 dan 5. Pada K-Gram 10 mendapatkan hasil tingkat plagiarisme 1,439%, sementra untuk nilai K-Gram 5 mendapat hasil tingkat plagiarisme 17,449%. Kesimpulan dari penelitian ini adalah dalam pemilihan nilai K-Gram sangat mempengaruhi persentase tingkat plagiarisme kebenaran. Untuk nilai K-Gram lebih tinggi cenderung memiliki tingkat kesamaan yang rendah, sementara K-Gram yang rendah lebih meningkatkan presentase kesamaan. Penentuan K-Gram membutuhkan pertimbangan yang benar sehingga nilainya akan lebih tepat[3] .

Pada penelitian yang dilakukan oleh PM Prihatini dkk pada tahun 2017 yang berjudul Stemming Algorithm for Indonesian Digital News Text Processing membahas mengenai Stemming algoritma, Proses untuk menemukan kata dasar dari kata asli yang muncul dalam teks disebut Stemming. Nilai dari algoritma Stemming Nazief-Adriani adalah 0,9976(~100%) untuk Precision, 0,8966(90%) untuk Recall, dan 0,9444(94%) untuk F-Measure. Jadi, kelengkapan kata dalam kamus kata dasar dan ketepatan dan kelengkapan aturan memainkan peran penting dalam keberhasilan algoritma Stemming[4].

2. METODE PENELITIAN

2.1 *Dataset*

Data yang digunakan pada penelitian ini merupakan data berupa teks abstraksi tugas akhir yang diperoleh dari repository universitas amikom yogyakarta. Alur proses pendeteksian plagiarisme dalam penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Diagram Proses Pendeteksian Plagiarisme

2.2 Text Preprocessing

Text preprocessing adalah proses pengolahan data mentah menjadi data yang siap untuk diproses seperti menghilangkan karakter dan simbol tanda baca yang tidak dibutuhkan, membuang kata-kata yang tidak dibutuhkan

2.3 Algoritma Rabin-Karp

Algoritma Rabin-Karp dikembangkan oleh dua peneliti yaitu, Michael O.Rabin dan Richard M.Karp tahun 1987[5]. Algoritma Rabin-Karp adalah algoritma pencocokan string yang menggunakan fungsi hash sebagai pembanding antara string yang dicari (m) dengan substring pada teks (n). Nilai hash yang dihitung secara efisien dapat mempengaruhi kinerja dari algoritma ini[6]. Pada umumnya, nilai hash digambarkan sebagai suatu string terpendek yang terdiri atas angka dan huruf yang terlihat acak yaitu data biner dalam bentuk heksadesimal, gambaran tersebut dikenal dengan istilah fingerprint[7]. Rumus Hashing dalam algoritma Rabin-Karp adalah sebagaimana yang ditunjukan pada persamaan (1).

$$h(s) \leftarrow (s[i] * b\,(n-1) + s[i+1] * b\,(n-2) + \ldots + s[i+n-1]) mod q \qquad (1)$$

Dimana :
s = panjang string yang dicari
i = array ke-i
b = basis

JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 1, Februari 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i1.1927
Hal 1-6

n = jumlah panjang string yang dicari
q = modulo

2.4 Pengukuran Nilai Similarity

Nilai similarity antara dokumen uji terhadap dokumen asli dapat dihitung setelah mendapatkan jumlah substring pada dokumen uji yang match atau cocok dengan substring pada dokumen asli. Mencari nilai similarity dengan cara mengelompokkan hasil terms dari K-grams yang sama. Kemudian untuk menghitung similarity dari kumpulan kata tersebut maka digunakan *Dice's Similarity Coefficient* untuk pasangan kata yang digunakan. Rumus yang digunakan untuk mencari nilai similarity dapat dilihat pada persamaan (2).

$$S = \frac{2 * C}{(A + B)} \qquad (2)$$

Dimana
S = nilai similarity
A,B = jumlah dari kumpulan K-Grams dalam teks 1 dan teks 2
C = jumlah hashing yang telah dibandingkan

3. ANALISA DAN PEMBAHASAN

Dokumen yang digunakan untuk melakukan pengujian adalah dokumen input yang berupa abstrak tugas akhir dan dibandingkan dengan 5 dokumen yang ada dalam database.

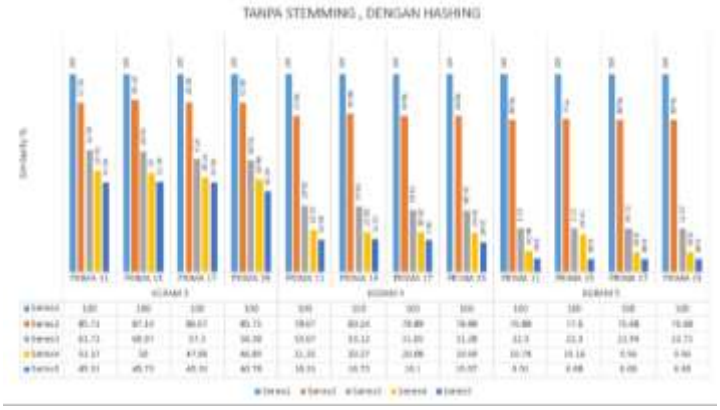
3.1 Uji Coba Nilai K-Gram dan Nilai Basis

Pengujian dilakukan dengan menggunakan nilai K-Gram 3,4,5 dengan nilai basis 11,13,17,23 dengan proses menggunakan Stemming dan Rolling Hash dan tanpa melewati proses Stemming dan Rolling Hash. Hasil penguiian dapat dilihat pada gambar 2,3,4 dan 5 sebagai berikut :

Gambar 2. Diagram Proses Tanpa Stemming dan Tanpa Hashing

Gambar 3. Diagram Proses Stemming dan Hashing

JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 1, Februari 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i1.1927
Hal 1-6



Gambar 4. Diagram Proses Tanpa Stemming dan Hashing

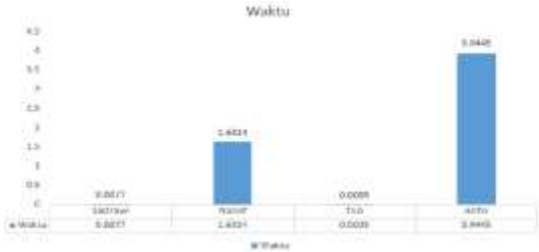
Gambar 5. Diagram Proses Stemming dan Tanpa Hashing

Dari proses uji akurasi abstraksi yang ditunjukkan pada gambar 2,3,4,dan 5 dapat ditarik kesimpulan sebagai berikut:

a. Dari hasil percobaan diketahui bahwa perubahan yang terjadi pada nilai KGRAM akan mempengaruhi hasil pengecekan. Semakin kecil nilai K-GRAM maka nilai hasil pengecekan akan semakin tinggi.
b. Dihilangkannya proses stemming akan membuat hasil pengecekan semakin tinggi, hal ini terjadi karena banyak imbuhan dan akhiran kata yang ikut andil dalam menambah nilai similarity. Oleh karena itu pemilihan algoritma stemmer yang akan dipakai akan sangat mempengaruhi hasil dan akurasi proses pengecekan rabinkarp.
c. Penggunaan hashing pada proses rabinkarp mempengaruhi hasil similarity. Jika nilai prima yang kita terapkan terlalu kecil maka nilai hasil pengecekan akan cenderung besar dan prosesnya akan lebih cepat, namun apabila terlalu besar maka nilai hasil pengecekan akan cenderung kecil dan prosesnya akan lama. Nilai ideal dari prima yang diterapkan sebaiknya tidak terlalu tinggi dan tidak terlalu rendah.
d. Abstrak dengan tingkat plagiarisme yang tinggi tidak akan banyak berubah nilai similaritynya ketika K-GRAM dan PRIMA nya diubah, baik itu menggunakan stemmer atau tidak dan melalui proses hashing atau tidak.
e. Abstrak dengan tingkat plagiarisme yang rendah akan sangat variatif nilai similaritynya ketika K-GRAM dan PRIMA nya diubah, baik itu menggunakan stemmer atau tidak dan melalui proses hashing atau tidak.
f. Nilai K-GRAM yang ideal adalah tidak terlalu tinggi atau tidak terlalu rendah, untuk nilai PRIMA 11,13,17, dan 23 nilai K-GRAM yang optimal berada di angka 5. Karena di K-GRAM 5 ini hasilnya akan mulai terlihat stabil.

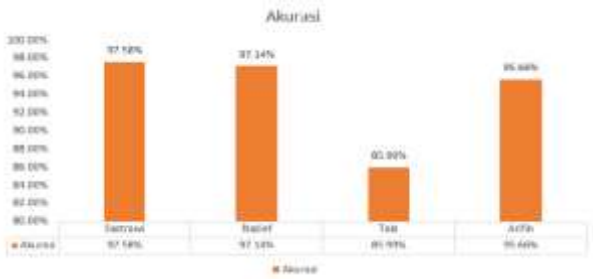
3.2. Uji Coba Keakuratan Library Stemmer

Perbandingan waktu dan akurasi dari tiap-tiap stemmer dengan menggunakan data uji yang sama dapat dilihat pada gambar berikut:



Gambar 6. Diagram Perbedaan Waktu Uji Coba Keakuratan library Stemmer

JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 1, Februari 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i1.1927
Hal 1-6



Gambar 7. Diagram Perbedaan Akurasi Uji Coba Keakuratan library

Proses pengujian stemmer dilakukan dengan membandingkan abstrak yang sudah di stem oleh masing-masing algoritma dengan abstrak yang di stem secara manual berdasarkan KBBI. Dari hasil pengujian yang ditunjukkan oleh gambar 6 dan 7 diatas didapat kesimpulan sebagai berikut:

a. Algoritma dengan proses paling cepat adalah Sastrawi dan Tala, sedangkan yang paling lama adalah Nazief-Andriani dan Arifin-Setiono.
b. Algoritma dengan hasil akurasi paling tinggi adalah Sastrawi dan Nazief Andriani. Kedua algoritma ini terbukti akurat karena mampu mengolah kata berimbuhan dan berakhiran dengan variasi yang bermacam-macam menjadi kata dasar yang cocok dengan KBBI serta mempunyai tingkat kesalahan yang kecil.
c. Tingkat akurasi Algoritma Nazief-Andriani dan Arifin sangat dipengaruhi oleh jumlah data kata dasar di database yang dipakai untuk pengecekan. Semakin banyak data di databasenya maka hasil akan semakin akurat namun proses stemming akan semakin lama.
d. Algoritma Arifin dan Tala mempunyai tingkat kesalahan yang cukup tinggi dimana kedua algoritma ini beberapa kali terbukti menghilangkan huruf-huruf penting pada kata dasar. Hal ini dapat mempengaruhi hasil pengecekan, walaupun tidak signifikan tapi integritas datanya menjadi rendah sehingga kurang bisa dipertanggung jawabkan.
e. Algoritma sastrawi selain melakukan stemming ternyata juga melakukan proses mengubahan case dari kapital menjadi lowercase. Hal ini sangat bagus mengingat kita juga akan melakukan rolling hash pada tiap potongan kata supaya proses pengecekan akan lebih cepat, dan lebih akurat.
f. Dari hasil uji coba ini, didapat kesimpulan bahwa algoritma yang paling cocok untuk digunakan adalah sastrawi karena akurasinya yang tinggi dan waktu stemming yang cukup cepat.

4. IMPLEMENTASI

Berikut merupakan hasil dari implementasi tapi aplikasi pengecekan plagiarism detection.

Gambar 8. Implementasi Halaman Dashboard

Gambar 9. Implementasi Halaman Result

JURIKOM (Jurnal Riset Komputer), Vol. 7 No. 1, Februari 2020
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i1.1927
Hal 1-6

Gambar 10. Implementasi Halaman Detail Result

5. KESIMPULAN

Berdasarkan pembahasan diatas maka penulis dapat mengambil kesimpulan bahwa aplikasi ini dapat digunakan untuk mendeteksi plagiarism dengan baik dengan menggunakan metode Rabin-Karp. Proses Stemming memiliki pengaruh yang signifikan terhadap nilai similarity dimana nilai similarity akan bernilai lebih rendah pada proses yang menggunakan stemming apabila dibandingkan dengan yang tidak. Dari dua proses stemming dengan stemmer yang berbeda, stemmer nazief andriani dinilai lebih akurat daripada stemmer sastrawi apabila jumlah data pada database katadasar diperluas. Penulis juga dapat menyimpulkan bahwa jika nilai K-gram semakin kecil maka tingkat kesamaan dokumen akan semakin besar dan sebaliknya. Besar kecilnya nilai basis bilangan memiliki pengaruh terhadap kesamaan pada dokumen yang diuji namun tidak signifikan. Untuk penelitian selanjutnya diharapkan aplikasi dapat dikembangakan lebih lanjut sehingga sistem tidak hanya dapat mendeteksi plagiarisme abstraksi tugas akhir saja tetapi juga dapat mendeteksi plagiarisme pada file tugas akhir dan disarankan juga untuk mencoba algoritma stemming kata yang lain untuk perbandingan.

REFERENCES

[1] Departemen Pendidikan dan Kebudayaan RI, “Kamus Besar Bahasa Indonesia,” 2013. [Online]. Available: https://kbbi.web.id/plagiarisme.
[2] S. KiranShivaji and P. S, “Plagiarism Detection by using Karp-Rabin and String Matching Algorithm Together,” *Int. J. Comput. Appl.*, vol. 115, no. 23, pp. 37–41, 2015, doi: 10.5120/20294-2734.
[3] S. Andysah Putera Utama, M. Mesran, R. Robbi, and S. Dodi, “K-Gram As A Determinant Of Plagiarism Level In Rabin-Karp Algorithm,” *Int. J. Sci. Technol. Res.*, vol. 6, no. 07, pp. 350–353, 2017.
[4] P. M. Prihatini, I. K. G. D. Putra, I. A. D. Giriantari, and M. Sudarma, “Stemming Algorithm for Indonesian Digital News Text Processing,” *Int. J. Eng. Emerg. Technol.*, vol. 2, no. 2, pp. 1–7, 2017.
[5] K. T. Tung, N. D. Hung, L. Thi, and M. Hanh, “A Comparison of Algorithms used to measure the Similarity between two documents,” *Int. J. Adv. Res. Comput. Eng. Technol.* , no. April 2016, 2015.
[6] N. ALAMSYAH, “Perbandingan Algoritma Winnowing Dengan Algoritma Rabin Karp Untuk Mendeteksi Plagiarisme Pada Kemiripan Teks Judul Skripsi,” *Technol. J. Ilm.*, vol. 8, no. 3, p. 124, 2017, doi: 10.31602/tji.v8i3.1116.
[7] A. H. Purba and Z. Situmorang, “Analisis Perbandingan Algoritma Rabin-Karp Dan Levenshtein Distance Dalam Menghitung Kemiripan Teks,” *J. Tek. Inform. Unika St. Thomas*, vol. 02, pp. 24–32, 2017.