DETEKSI SIMILARITAS ARTIKEL ILMIAH DENGAN TEKNIK PENCOCOKAN STRING BOYER MOORE

Amardeep
Program Magister Manajemen Sistem Informasi, Universitas Gunadarma
lights.immortal@gmail.com

Abstrak

Tindakan plagiarisme sering terjadi khususnya pada proses penulisan baik dalam bentuk artikel ilmiah maupun jurnal. Salah satu kontrol yang dapat dilakukan untuk meminimalisir adanya tindakan plagiarisme adalah dengan melakukan perbandingan kemiripan dokumen dengan menghitung tingkat similaritas. Pada penelitian akan dilakukan analisis terhadap penggunaan algoritma Boyer-Moore dengan teknik String Matching pada dokumen berbentuk jurnal ilmiah. Penelitian ini menggunakan teknik crawling dengan memanfaatkan library beautiful soup dari Python pada mesin pencari Google untuk membandingkan dokumen uji berbentuk jurnal ilmiah dengan hasil penelusuran oleh Google agar perbandingan dokumen dapat diperluas sehingga akurasi kemiripan dokumen dapat bertambah. Penelitian ini melakukan pengujian kemiripan dokumen pada jurnal bahasa Indonesia dan bahasa Inggris dalam sebuah jurnal ilmiah dimana proses stemming untuk kedua bahasa dilakukan secara terpisah. Pada deteksi kalimat berbahasa Indonesia, proses stemming dilakukan menggunakan stemming Nazief-Adriani dan pada stemming kalimat berbahasa Inggris digunakan algoritma Porter. Hasil analisis pencocokan string dengan algoritma Boyer-Moore pada proses bigram dapat memisahkan kata menjadi 2 kelompok kata yang disusun dalam 1 list pada setiap kalimat dan hasil pencariannya telah berhasil dilakukan, skor dan tingkat kemiripan dokumen melalu teknik crawling berhasil menghitung persentase kemiripan sebuah artikel ilmiah. Hasil penelitian ini diharapkan dapat menentukan tingkat similaritas dari dua buah dokumen, sehingga dapat meminimalisir tingkat plagiarisme khususnya pada dokumen berbentuk jurnal ilmiah.

***Kata kunci**: Boyer-Moore, Crawling, Kalimat, Tokenization*

Abstract

Plagiarism often occurs, especially in the writing process, both in the form of scientific articles and journals. One of the controls that can be done to minimize the existence of plagiarism is by comparing the similarity of documents and calculating the level of similarity. This study will analyze the use of the Boyer-Moore algorithm with the String Matching technique on documents as in scientific journals. This study uses a crawling technique by utilizing the library beautiful soup from python through Google search engine to compare test documents in the form of scientific journals with search results by Google so the document comparisons can be expanded and the accuracy of document similarities can increase. This study examined the similarity of documents in Indonesian and English journals in form of a scientific journal where the stemming process for both languages was carried out separately. In the detection of Indonesian sentences, the stemming process is carried out using Nazief-Adriani stemming meanwhile the stemming for English sentences is carried out using Porter's algorithm. The results of string matching analysis with the Boyer-Moore algorithm in the Bigram process can separate words into two groups that arranged in one list for each sentence and the search results have been successfully done, the score and level of document similarity through crawling techniques succeed in calculating the percentage of similarity to a scientific article. The results of this study are expected to determine the level of similarity of the two documents, thus it can minimize the plagiarism especially for scientific journals.

***Keywords**: Boyer-Moore, Crawling, Sentence, Tokenization*

PENDAHULUAN

Tindakan plagiarisme sering terjadi khususnya pada proses penulisan baik dalam bentuk artikel ilmiah maupun jurnal. Salah satu kontrol yang dapat dilakukan untuk meminimalisir adanya tindakan plagiarisme adalah dengan melakukan perbandingan kemiripan dokumen dengan menghitung tingkat similaritas. Beberapa penelitian terkait pendeteksian plagiarisme telah dilakukan peneliti terdahulu.

Penelitian [1] melakukan perbandingan kemiripan dokumen pada beberapa teks menggunakan *Turbo Boyer-Moore* dengan metode *string-matching*. Penelitian ini tidak menggunakan teknik *stemming* pada tahapan *preprocessing* maka kata-kata yang diuji belum diubah menjadi kata dasar. Hasil yang didapat pada penelitian ini yaitu bahwa tahap *preprocessing* menentukan kecepatan proses perhitungan skor dengan algoritma *Turbo Boyer-Moore*. Penelitian [2] melakukan perbandingan dokumen untuk mendapatkan nilai *similarity* menggunakan teknik *n-gram*. Peneliti melakukan teknik *n-gram* dalam tingkat *string* jadi teks yang diuji akan dipisah menjadi bagian-bagian *string*. Penelitian [3] melakukan penelitian untuk menemukan metode yang paling sesuai dalam melakukan pengukuran kesamaan string. Peneliti menggunakan GEOnet *Names Server* dengan 21 dataset toponim dari 11 negara yang dilatinkan. Penelitian ini melakukan kajian terhadap metode pengukuran kesamaan string seperti teknik *n-grams* dengan algoritma *Skip-grams*, *Smith-Waterman* dan LCS (L=3), *bag distance* dan NCD. Hasil yang didapat pada penelitian ini yaitu algoritma *Skip-grams* menjadi algoritma pilihan terbaik. Penelitian [4] melakukan perbandingan *code files* antara *file* sumber dengan *file* yang "dicurigai" hasil plagiarisme. Peneliti melakukan 2 tahap yaitu memilih *file* sumber dari dataset kemudian membandingkan *file* tersebut dengan *file* Top "K" dari dataset yang ada dan pada tahap selanjutnya peneliti melakukan penilaian menggunakan algoritma *greedy string tilling*. Peneliti mendapatkan hasil bahwa semakin tinggi nilai "K" maka semakin banyak perbandingan yang harus dilakukan. Peneltian [5] melakukan perbandingan algoritma *cosine similarity measure*, algoritma *fingerprint* dan *winnowing* untuk menemukan kesamaan antara dua dokumen. Algoritma *cosine* mengukur dua vektor dimensi yang mewakili tiap-tiap dokumen. Algoritma ini menghasilkan nilai lebih tinggi daripada estimasi kesamaan. Algoritma kedua yaitu algoritma *fingerprint* menggunakan fungsi *hash* untuk membandingkan dokumen setelah sebelumnya diubah menjadi rangkaian kata menggunakan teknik *n-gram*. Waktu yang dibutuhkan oleh algoritma ini ditentukan oleh ukuran dokumen. Algoritma terakhir adalah algoritma *winnowing* yang membandingkan nilai *hash* menggunakan persamaan *Dice Coefficient*. Performa terbaik untuk algoritma ini ditentukan oleh pemilihan teknik dan parameter yang sesuai. Penelitian [6]

melakukan perbandingan dokumen untuk mendapatkan nilai *similarity* menggunakan algoritma *Rabin-Karp*. Algoritma ini membandingkan *string* yang ditelah diubah menjadi angka menggunakan teknik *hashing*. Selain menggunakn fungsi *hashing* penelitian ini juga menggunakan teknik *stemming* dalam prosesnya. Algoritma ini sangat baik digunakan untuk pencocokan kata yang memiliki banyak pola. Algoritma *Rabin-Karp* membutuhkan angka prima yang banyak untuk menghindari mendapatkan nilai hash yang sama untuk beberapa kata. Penelitian [7] melakukan pendeteksian *file* duplikat untuk desktop dengan metode *string matching*. Penelitian ini mengusulkan sebuah algoritma baru yaitu *Word to Word COMparison* (W2COM), selain itu penelitian ini membandingkan algoritma W2COM dengan algoritma *Boyer Moore Horspool* dan algoritma *Knuth Morris Pratt*. Algoritma *Boyer Moore Horspool* digunakan untuk pencarian substring pada file besar, menggunakan pola pencarian string dari kanan ke kiri. Algoritma *Boyer Moore Horspool* dapat digunakan untuk huruf kecil dan multi pola. Algoritma *Knuth Morris Pratt* adalah algoritma yang dirancang untuk untuk pencarian *string* yang linear dengan menganalisa hasil algoritma sederhana yang variabelnya ditentukan dari awal. Algoritma *Knuth Morris Pratt* hanya bekerja pada huruf kecil dan sebuah poli, sedangkan algoritma W2COM dapat digunakan untuk huruf kapital dan multi pola.

Pada penelitian akan dilakukan analisis terhadap penggunaan algoritma Boyer-Moore [8] dengan teknik *String Matching* pada dokumen berbentuk jurnal ilmiah. Penelitian ini menggunakan teknik *crawling* [9] dengan memanfaatkan *library* beautiful soup dari python pada mesin pencari Google untuk membandingkan dokumen uji berbentuk jurnal ilmiah dengan hasil penelusuran oleh Google agar perbandingan dokumen dapat diperluas sehingga akurasi kemiripan dokumen dapat bertambah. Penelitian ini melakukan pengujian kemiripan dokumen pada jurnal bahasa Indonesia dan bahasa Inggris dalam sebuah jurnal ilmiah dimana proses stemming untuk kedua bahasa dilakukan secara terpisah. Pada deteksi kalimat berbahasa indonesia, proses *stemming* [10] dilakukan menggunakan stemming Nazief-Adriani dan pada stemming kalimat berbahasa inggris digunakan algoritma Porter [11]. Hasil penelitian ini diharapkan dapat menentukan tingkat similaritas dari dua buah dokumen, sehingga dapat meminimalisir tingkat 129lagiarism khususnya pada dokumen berbentuk jurnal ilmiah.

METODE PENELITIAN

Metode pada penelitian ini terdiri dari beberapa tahapan proses. Tahapan-tahapan dari proses tersebut bertujuan untuk membandingkan dokumen/teks input dengan teks yang didapat dari hasil pencarian pada

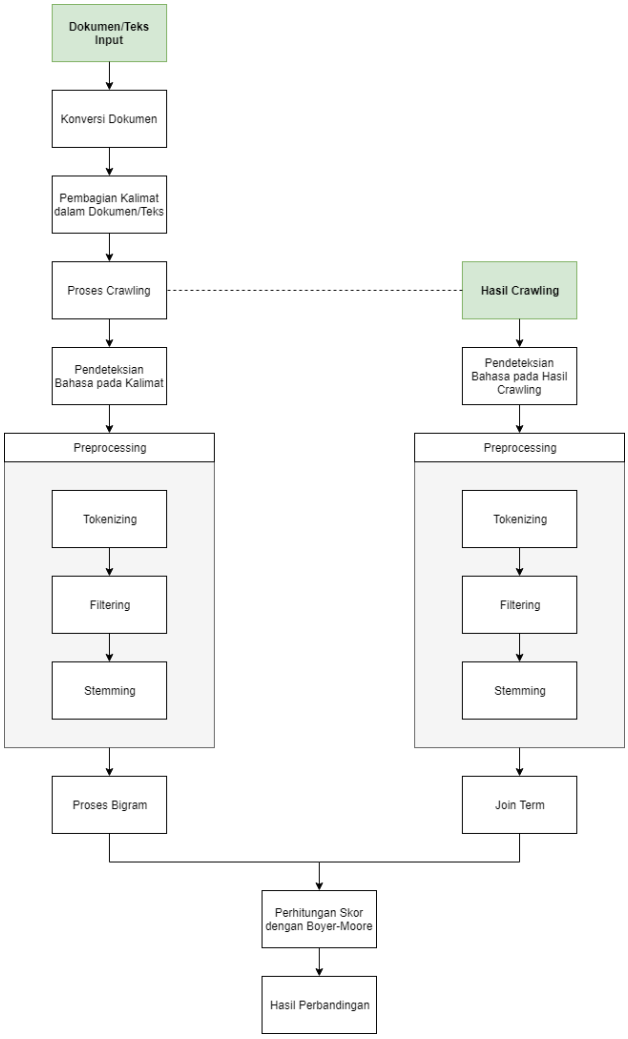
mesin pencari Google. Gambar 1. merupakan tahapan-tahapan pada penelitian ini.

Analisis similaritas dokumen jurnal ilmiah dilakukan dengan penghitungan skor kesamaan dokumen/teks menggunakan algoritma Boyer-Moore menggunakan metode *string-matching*. Pada proses *stemming* berbahasa Inggris dilakukan menggunakan algoritma Porter dan pada proses *stemming* berbahasa Indonesia dilakukan menggunakan algoritma Nazief-Adriani.

A. Data Penelitian

Pada penelitian ini digunakan dokumen sejumlah 300 hasil penelitian ilmiah di bidang pangan dan maritim yang berisi judul, abstrak, kata kunci, topik, dan kelas (PDII LIPI, 2019). Dokumen sumber ini didapat dari PDII-LIPI sebagai sampel dari data asli yang digunakan nanti.

Pada penelitian juga digunakan dokumen yang memiliki dua bahasa di dalamnya yaitu bahasa Indonesia dan bahasa Inggris. Data ini digunakan untuk menghitung akurasi dari penggunaan algoritma Boyer-Moore agar proses pendeteksian bahasa dapat berjalan dengan baik.



Gambar 1. Metode Penelitian

B. Normalisasi Data

Proses normalisasi data input dilakukan dengan tahapan konversi dokumen dan pembagian kalimat. Konversi dokumen dibutuhkan agar tahapan preprocessing dapat dilakukan karena preprocessing membutuhkan plain text sebagai input. Hasil dari konversi dokumen akan berbentuk plain text. Pembagian kalimat ditujukan untuk membagi kalimat atau memisahkan tiap kalimat dengan batasan tanda baca titik (.) dari hasil konversi dokumen yang sudah dilakukan sebelumnya dengan memanfaatkan penggunaan Natural Language Toolkit dari Python menggunakan fungsi sent_tokenize().

C. Proses Crawling Teks

Proses ini menggunakan kalimat hasil pembagian sebelumnya untuk dicari pada mesin pencari Google. Setiap kalimat akan dicari dan hasil pencarian akan digunakan sebagai pembanding dari kalimat utama untuk proses perhitungan skor. Hasil akhir yang diambil adalah 5 pencarian teratas oleh mesin pencari Google. Potongan teks dari halaman hasil pencarian Google dilakukan menggunakan library beautifulsoup.

D. Tahap Pra Pengolahan

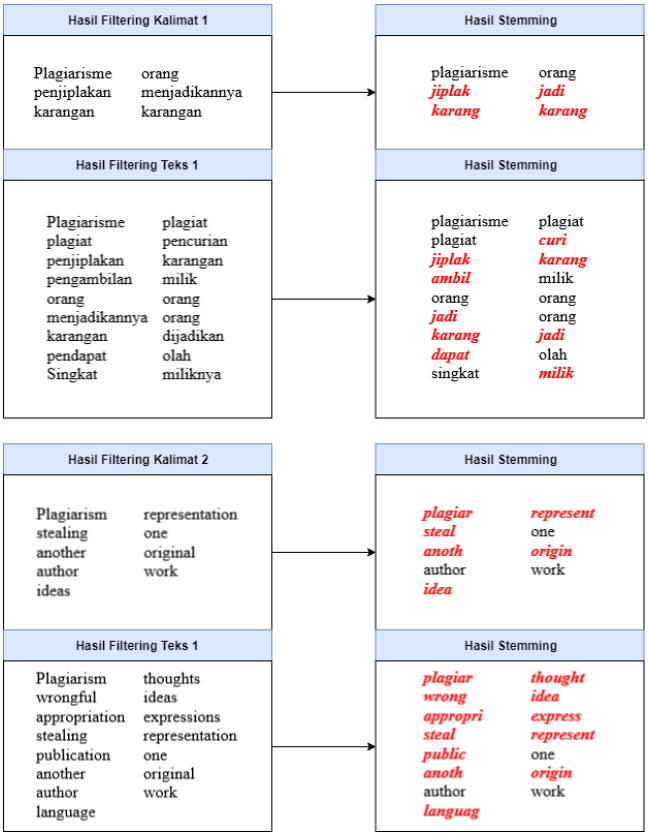
Hasil dari tahap ini digunakan untuk menghitung skor dari perbandingan kalimat pada dokumen utama dan teks hasil crawling. Tahapan ini melibatkan 3 proses yaitu tokenizing, filtering, dan stemming.

1. Tokenizing: Proses ini memisahkan kata berdasarkan spasi yang ada pada suatu teks. Proses ini akan menghasilkan term atau urutan kata yang tersusun dalam suatu list.
2. Filtering: Proses ini bertujuan untuk menghilangkan kata yang kurang penting atau kurang mempunyai makna yang jelas seperti kata sambung dan lainnya.
3. Stemming: Proses ini mengubah setiap kata menjadi kata dasarnya. Stemming berfungsi untuk membuat perhitungan skor menjadi lebih optimal karena kata dalam masing-masing teks sudah dijadikan kata dasar. Algoritma *stemming* yang digunakan pada penelitian ini adalah algoritma Porter untuk teks berbahasa Inggris dan algoritma Nazief-Adriani untuk teks berbahasa Indonesia. Hasil dari proses ini dapat dilihat pada Gambar 2.

E. Analisa Penghitungan Skor Similaritas

Penghitungan similaritas dari dokumen jurnal ilmiah dilakukan melalui tahapan proses menggunakan hasil pada proses stemming yaitu proses bigram, join term lalu dilakukan perhitungan skor similaritas nya.

Proses bigram akan menghasilkan list kata yang akan digunakan untuk perhitungan skor dengan membandingkan setiap kumpulan kata yang ada di list dengan hasil crawling yang sudah menjalani tahap preprocessing, penulis menggunakan n=2. Join term dilakukan untuk menggabungkan tiap kata yang ada pada list hasil stemming untuk membentuk sebuah kalimat kembali. Hasil dari proses ini akan digunakan pada proses perhitungan skor



Gambar 2. Hasil *Stemming*

Proses perhitungan Skor Similaritas memiliki tujuan untuk mencari nilai kemiripan dari dokumen yang sudah diinput dengan hasil crawling. Teks yang dibandingkan yaitu masing-masing kalimat dari dokumen input yang sudah melalu proses n-gram dengan teks hasil crawling dari kalimat tersebut yang sudah digabungkan kembali setelah tahap preprocessing. Tahap ini menggunakan algoritma Boyer-Moore untuk menghitung nilai similaritas dokumen.

Analisis menggunakan Boyer-Moore dilakukan dengan tahapan berikut :

1) Pada langkah awal perbandingan akan berjalan dari karakter terakhir sampai awal pada kedua teks. Saat karakter pertama dibandingkan dapat dilihat bahwa i tidak sama dengan m maka cari terlebih dahulu apakah huruf m ada pada teks pattern setelah huruf i.
2) Karakter m tidak ditemukan maka geser semua teks pattern sampai melewati karakter m di teks sentence.
3) Karakter teks pattern pada langkah 2 dimulai pada huruf e di teks sentence. Lalu bandingkan lagi karakter terakhir dari kedua teks. Terlihat bahwa spasi tidak sama dengan i namun spasi ditemukan setelah huruf i di teks pattern maka geser teks pattern dengan menyamakan posisi spasi pada kedua teks.
4) Pada langkah 3 terlihat bahwa posisi spasi sama pada kedua teks. Lalu mulai

lagi perbandingan karakter dari posisi terakhir ke awal. Terlihat bahwa karakter terakhir tidak sama antara kedua teks. Geser teks pattern ke posisi setelah huruf l pada teks sentence karena huruf l tidak ditemukan pada teks pattern.

5) Teks pattern pada langkah 4 dimulai setelah huruf l pada teks sentence karena langkah sebelumnya. Karakter terakhir pada kedua teks tidak sama namun huruf o pada teks sentence ditemukan pada teks pattern setelah huruf I maka geser teks pattern dengan menyesuaikan huruf o pada kedua teks.
6) Langkah 5 dimulai dengan membandingkan karakter terakhir sampai awal pada kedua teks. Ditemukan bahwa kedua teks sama dan proses sudah selesai.
7) Setelah proses perbandingan teks dengan algoritma Boyer-Moore tersebut selesai maka hal yang sama dilakukan pada teks pattern lainnya dari hasil bigram kalimat 1 dan 2.

Setelah proses perbandingan teks pada hasil bigram kalimat selesai maka hal selanjutnya yang dilakukan adalah melakukan perhitungan skor yang ditentukan dari banyaknya perbandingan teks yang telah ditemukan dibagi dengan jumlah teks yang dibandingkan atau jumlah teks hasil bigram (persamaan 1).

$$Skor\ Perbandingan\ Teks = \frac{\sum Teks\ Sama}{\sum Teks\ Dibandingkan} \times 100 \quad (1)$$

HASIL DAN PEMBAHASAN

Dokumen uji yang digunakan merupakan dokumen sumber berisi 100 buah data penelitian di bidang pangan dan maritim yang berasal dari PDII-LIPI. Dokumen input yang diuji berjumlah 1 buah dokumen yang berasal dari dokumen sumber. Setelah menentukan dokumen sumber dan dokumen uji, peneliti melakukan pemisahan teks menggunakan library dari Python yaitu *Sentence Tokenizer*. Tahapan selanjutnya adalah melakukan *crawling* teks pada mesin pencarian untuk setiap kalimat yang telah dibagi. Peneliti juga melakukan pendeteksian bahasa untuk menentukan *stopword* dan algoritma *stemming* yang akan digunakan pada tahap *preprocessing*. Pada tahap *preprocessing* dilakukan 3 hal, yaitu *tokenizing, filtering*, dan *stemming*. *Tokenizing* merupakan pemisahan kalimat menjadi kata per kata, seperti dapat dilihat pada Tabel 1. Kata-kata tersebut lalu disaring untuk tidak memunculkan kata yang tidak memiliki makna dan kata yang berupa angka. Tahap selanjutnya dilakukan proses stemming yaitu menghilangkan imbuhan pada setiap kata.

Tabel 1. Hasil Tokenizing Kalimat

Kalimat	Hasil
1	Proses, pencarian, merupakan, salah, satu, kegiatan, penting, dalam, pemrosesan, data
2	Proses, ini, dapat, menghabiskan, waktu, dalam, ruang, pencarian, yang, besar, sehingga, diperlukan, suatu, teknik, pencarian, yang, efisien

Tabel 2. Hasil *Stemming* Kalimat

Kalimat	Hasil
1	('proses', 'cari'), ('cari', 'salah'), ('salah', 'giat'), ('giat', 'pemrosesan'), ('pemrosesan', 'data')
2	('proses', 'habis'), ('habis', 'ruang'), ('ruang', 'cari'), ('cari', 'teknik'), ('teknik', 'cari'), ('cari', 'efisien')

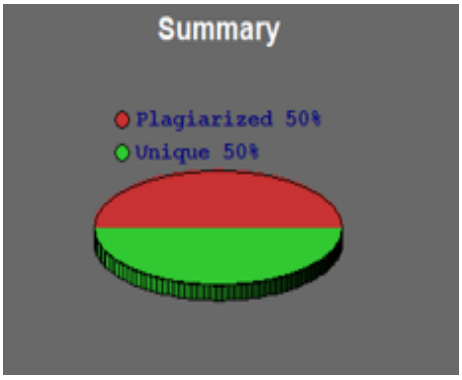
Peneliti melakukan proses *bigram* untuk membagi kata hasil *stemming* menjadi kelompok kata dan tiap kelompok terdiri dari 2 kata seperti dapat dilihat pada Tabel 2. Selanjutnya dilakukan proses *join term* untuk menggabungkan kembali hasil *preprocessing* menjadi teks seutuhnya dari hasil pencarian kalimat yang akan digunakan sebagai pembanding.

Setelah semua proses dilalui, saatnya untuk melakukan penghitungan skor, pada proses ini yang dilakukan adalah membandingkan hasil proses bigram dari tiap kalimat pada dokumen dengan seluruh hasil pencarian kalimat tersebut pada mesin pencari google yang sudah melewati proses join term. Proses pembandingan ini menggunakan algoritma Boyer-Moore dengan metode *string matching*. Setelah mendapatkan skor dari tiap perbandingan kalimat dan hasil pencariannya selanjutnya yaitu menghitung skor total dengan memilih terlebih dahulu skor terbesar (skor yang diberi warna hijau pada tabel) dari tiap hasil pencarian lalu menambahkan tiap hasil perbandingan kalimat dan dibagi dengan jumlah kalimat seperti pada perhitungan berikut.

$$Skor\ Total = \frac{20 + 100 + 100 + 100 + 100 + 40 + 10 + 30 + 0 + 0}{10}$$

$$= 50\ \%$$

Tahapan ini menghasilkan skor kemiripan dokumen sebanyak 50%. Berdasarkan skor yang didapat dapat disimpulkan bahwa dokumen yang diuji termasuk ke dalam kategori plagiarisme sedang menurut klasifikasi oleh Sudigdo Sastroasmoro. Persentase tingkat similaritas yang dilakukan dapat dilihat pada Gambar 3.



Gambar 3. Grafik Tingkat Similaritas

KESIMPULAN

Proses stemming pada dokumen jurnal ilmiah untuk kalimat/teks berbahasa inggris dan kalimat/teks berbahasa Indonesia menghasilkan kata dasar. Proses mengubah kata menjadi kata dasar menggunakan algoritma Porter dan Nazief-Adriani pada setiap kalimat dan hasil pencariannya telah berhasil dilakukan.

Hasil analisis pencocokan string dengan algoritma Boyer-Moore pada proses bigram dapat memisahkan kata menjadi 2 kelompok kata yang disusun dalam 1 list pada setiap kalimat, skor dan tingkat kemiripan dokumen melalu teknik *crawling* berhasil menghitung persentase kemiripan sebuah artikel ilmiah. Pengembangan lebih lanjut yang dapat dilakukan untuk penelitian selanjutnya antara lain dengan melakukan analisis terhadap algoritma yang dapat mendeteksi urutan atau posisi kata, sinonim dari kata maupun parafrase dan metode lainnya agar skor dan tingkat kemiripan dapat memiliki tingkat akurasi yang lebih baik.

DAFTAR PUSTAKA

[1] P. I. Goni, “Penerapan algoritma Turbo Boyer-Moore untuk pendeteksian kemiripan dokumen teks berbasis web,” Skripsi, Universitas Kristen Satya Wacana, Salatiga, Indonesia, 2013.

[2] E. A. Lisangan, “Implementasi n-Gram Technique dalam deteksi plagiarisme pada tugas mahasiswa,” *Jurnal Tematika*, vol. 1, no. 2, Sep., hal. 24-30, 2013.

[3] G. Recchia dan M. Max Louwerse, “A Comparison of string similarity measures for toponym matching,” Dalam Proceedings of The First ACM SIGSPATIAL International Workshop on Computational Models of Place, 2013, hal. 54-61.

[4] O. Ajmal, M. M. S. Missen, T. Hashmat, M. Moosa, dan T. Ali, “EPlag: A two layer source code plagiarism detection system,” *Journal of Information Security Research*, vol. 5, no. 3, Sep., hal. 107-114, 2014.

[5] K. T. Tung, N. D. Hung, dan L. T. M. Hanh, "A Comparison of algorithms used to measure the similarity between two documents," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 4, no. 4, hal. 1117-1121, 2015.

[6] R. E. Putri dan A. Siahaan, "Examination of document similarity using Rabin-Karp algorithm," *International Journal of Recent Trends in Engineering & Research*, vol. 3, no. 8, Agu., hal. 196-201, 2017

[7] S. Vijayarani dan M. Muthulakshmi, "An efficient string matching technique for desktop search to detect duplicate files," *International Journal of Information Technology and Computer Science*. vol. 9, no. 7, Jul., hal. 69-76, 2017.

[8] E. Rahmanita, "Pencarian string menggunakan algoritma Boyer-Moore pada dokumen," *Jurnal NERO*, vol. 1, no. 1, hal. 15-26, 2014.

[9] Y. Patil dan S.Patil, "Review of web crawlers with specification and working," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 1, Jan., hal. 220-223, 2016

[10] A. Firdaus, Ernawati, dan A. Vatresia, "Aplikasi pendeteksi kemiripan pada dokumen teks menggunakan algoritma Nazief & Adriani dan metode Cosine Similarity," *Jurnal Teknologi Informasi*, vol. 10, no. 1, Apr. hal. 96–109, 2014.

[11] L. Agusta, "Perbandingan algoritma S*temming Porter* dengan algoritma Nazief & Adriani untuk *stemming* dokumen teks bahasa Indonesia," Konferensi Nasional Sistem dan Informatika 2009, Bali, Indonesia, 2009.