Jurnal Informatika Universitas Pamulang **ISSN: 2541-1004**
Penerbit: Program Studi Teknik Informatika Universitas Pamulang **e-ISSN: 2622-4615**
Vol. 6, No. 4, Desember 2021 (880-886) **10.32493/informatika.v6i4.16074**

Pengaruh Stemming Nazief & Adriani terhadap Performa Algoritma Rabin-Karp dalam Mendeteksi Kemiripan Teks

Muhamad Arief Yulianto[1], Nurhasanah[2]

[1,2]Program Studi Teknik Informatika, Universitas Pamulang, Jl. Puspitek, Buaran, Kec. Pamulang, Kota Tangerang Selatan, Banten, Indonesia, 15310
e-mail: [1]dosen02547@unpam.ac.id, [2]dosen01123@unpam.ac.id

Submitted Date: December 20th, 2021 Reviewed Date: January 08th, 2022
Revised Date: January 15th, 2022 Accepted Date: January 31st, 2022

Abstract

One of the information retrieval methods which be able to search root word of each word in document is stemming. Stemming process is done by eliminate prefixes, infixes, suffixes or confixes. Vega, Tala, Arifin and Setiono, Nazief and Adriani and Tala stemming are kind of Indonesian Language stemming. The method that is able to trace each character in sequence character is fingerprinting. Rabin-Karp algorithm is one of the fingerprinting method algorithms. This algorithm implement has function to process matching text/string, so it is really suitable to implement of text/string similarity detection. Researcher will analyze the effect of Nazief and Adriani stemming method to algorithm of Rabin-Karp performance to identify similarity of text/string. The researcher implemented datasets such as titles, keywords, introductions or abstracts from The Pamulang Informatics Engineering Journal which we had changed the wording. The result of the experiment data which has changed word order randomly that used stemming method has decreased 0.76% than without implemented stemming method. Furthermore, the experiment data which has been changed sentence order randomly has decreased 0.04% too.

Keywords: Analyse; Effect; Rabin-Karp; Similarity; stemming

Abstrak

Salah satu kaidah pada pencarian informasi kembali atau *retrieval of information* memiliki manfaat dalam mencari bentuk kata dasar dari setiap kata pada dokumen disebut sebagai *stemming*. Proses *stemming* dilakukan dengan menghapus imbuhan berupa *prefix* (awalan), *infixes* (sisipan), *suffixes* (akhiran) maupun *confixes* (kombinasi awalan dan akhiran). Terdapat beberapa macam *stemming* dalam Bahasa Indonesia yang bisa digunakan diantaranya yaitu *stemming* Vega, *stemming* Tala, *stemming* Nazief Adriani, *stemming* Arifin Setiono dan *stemming* Porter. *Fingerprinting* merupakan salah satu metode yang mampu menelusuri setiap karakter pada deret karakter. Salah satu algoritma pada metode *fingerprinting* yaitu algoritma Rabin-Karp. Algoritma ini menerapkan fungsi *hash* dalam proses string/teks *matching*, sehingga sangat cocok diterapkan pada pendeteksian kemiripaan string/teks. Dalam penelitian ini, peneliti ingin menganalisa terhadap pengaruh metode *stemming* Nazief Adriani terhadap kinerja algoritma Rabin-Karp dalam mendeteksi tingkat kemiripan teks. Adapun teks yang kami maksud berupa judul, abstrak, kata kunci dan pendahuluan dari Jurnal Teknik Informatika Pamulang yang sudah kami ubah susunan katanya. Hasil pengujian data yang mengalami perubahan susunan kata secara acak yang mengimplementasikan metode *stemming* mengalami rata – rata penurunan sebesar 0,76% dibandingkan tanpa mengimplementasikan metode *stemming*. Seirama dengan pengujian data yang mengalami perubahan susunan kalimat secara acak juga mengalami rata – rata penurunan sebesar 0,04%.

Kata Kunci: Analisa; Kemiripan; Pengaruh; Rabin-Karp, *Stemming*

1. Pendahuluan

Data merupakan sebuah sumber informasi awal yang bernilai dan bermanfaat untuk semua orang jika dikembangkan (Yulianingsih, 2017). Data merupakan sebuah asset bernilai dan sebuah informasi digital yang dikirmkan melalui sinyal

digital yang harus dilindungi. Terdapat dua bentuk jenis data yaitu tertulis dan suara, hal ini memungkinkan untuk dimanipulasi ataupun digandakan tanpa sepengatahuan pemiliknya (Putra & Sujaini, 2016).

Pencarian string/text dalam bidang ilmu komputer sudah sangat familiar khususnya di dalam dunia pendidikan. Hal yang sering terjadi, ketika pelajar maupun mahasiswa melakukan proses pembuatan makalah, laporan, tugas akhir, dan lain-lain, dengan melakukan pencarian literatur yang memanfaatkan pencarian string/teks (dalam *search engine* ataupun fasilitas search pada piranti lunak yang dipakai). *Retrieval of information* atau pencarian informasi kembali sering digunakan pada *search engine* di dalam proses berselancar di dunia maya melalui website. Hal ini diimplementasikan dengan tujuan mendapatkan informasi akurat dan sesuai (Ruban et al., 2015).

Proses pencarian yang sering digunakan oleh *search engine* atau mesin pencarian melalui *text queries* yang diketikkan oleh pengguna yang dicocokkan dengan dokumen terkait (Verdaningroem & Saifu-din, 2018). Penjiplakan merupakan proses mencontoh hasil karya orang lain tanpa disertai dengan kutipan atau ijin dari sumbernya(Christina et al., 2018).

Sebuah proses pembentukan kata dasar dari berbagai macam bentuk kata disebut sebagai proses *stemming.* merupakan suatu proses untuk menemukan kata dasar dari sebuah kata. Metode yang digunakan dalam membentuk kata dasar dapat menggunakan kamus atau *dictionary based* dan tanpa kamus atau *purely rule based*. Stemming vega, stemming porter, stemming arifin dan setiono maupun stemming nazief dan adriani merupakan beberapa teknik yang digunakan pada stemming Bahasa Indonesia(Simarangkir, 2017).

Terdapat banyak tantangan dalam proses pencarian kata dasar pada Bahasa Indonesia. Dikarenakan dalam kata bahasa indonesa terdapat berbagai macam kata imbuhan berupa awalan, akhiran, awalan dan akhiran mapun sisipan. Selain imbuhan tersebut dalam bahasa Indonesia juga terdapat kata yang mengalami perulangan. Selain proses penghapusan imbuhan, stemming juga melakukan proses pengelolaan kata kunci atau *keyword* utuh seperti mengilangkan imbuhan seperti "meng", "me" mapun "di". Proses tersebut juga biasa disebut dengan proses *stem*. (Hapsari & Santoso, 2015).

Penelitian mengenai teknik *stemming* melalui "perpaduan dua algoritma yaitu *Winnowing* dan *Enhaned Config Stripping* mengindikasikan kinerja belum memuaskan", hal ini diungkapan oleh Sagala. Penelitian lainnya mengungkapkan "penggunaan teknik *stemming* mampu meningkatkan proses kerja sistem tetapi belum signifikan", hal ini diungkapkan oleh Alfikri, dkk. Sedangkan proses stemming pada algoritma Winnowing yang telah dilakukan oleh Hargyo mengungkapkan "adanya kecenderungan penurunan prosentase capaian similaritas, tetapi meningkatkan waktu proses 30% (Nugroho, 2017).

Mardiana, dkk mengungkapkan hal yang berbeda yaitu "berdasarkan F1-s*core* yang digunakan untuk mengimbangi angka presisi dan recall, deteksi yang menerapkan *stemming* dan *stopword* removal memiliki hasil yang lebih baik dalam mendeteksi kemiripan antar teks dengan rata-rata 42%. Hal ini lebih tinggi dibandingkan dengan pendeteksian kemiripan dengan hanya menggunakan proses *stemming* (31%) atau yang dilakukan tanpa menggunakan teks praproses (34%) saat mengaplikasikan bigram" (Mardiana et al., 2016).

Pemrosesan string/teks. Algoritma pada *text mining* memiliki berapa metode yang mampu mendeteksi kemiripan string/teks pada suatu dokumen disebut sebagai *text mining*. Terdapat algoritma untuk melakukan proses *string matching pattern* ini, diantaranya adalah w*innowing, rabin karp, boyer moore, brute force* dan sebagainya(Wicaksono & Suyanto, 2012).

Proses perbandingan nilai *hash* suatu kata(m) dengan bagian kata/*substring*(n) pada teks merupakan proses kerja dari algoritma rabin karp yang dinyaatakan dalam bentuk sebuah fungsi yang bernama *hash function.* Setiap nilai *hash* yang dicari jika tidak ditemukan pada bagian kata / *substring* maka proses pencarian akan terus dilakukan melalui pergeseran bagian kata/*substring* ke kanan sebanyak (n-m) kali. Kinerja dalam algoritma ini sangat berpengaruh terhadap efisiensi penghitungan *hash value* berdasarkan pergeseran *substring.* (Purba & Situmorang, 2017).

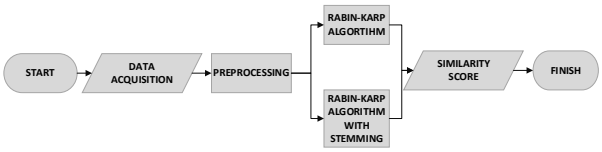
Peneliti juga sudah melakukan riset sebelumya tentang deteksi kemiripan teks bahasa Indonesia menggunakan penggabungan antara *Rabin Karp algorithm* dengan *Jaro Winkler algorithm*. Penelitian tersebut menunjukkan bahwa implementasi algoritma Jaro-Winkler pada Rabin Karp mampu meningkatkan prosentase pendeteksian kesamaan *string*/kata Bahasa Indonesia(Yulianto & Nurhasanah, 2021).

Terdapat beberapa ciri pada diantaranya teks yang tidak terstruktur, ukuran/volume data yang tinggi, terpada data yang tidak penting/*noise* pada

proses *text mining*. Oleh karena itu secara umum tahapan yang perlu dilalui pada proses *text mining* yaitu *preprocessing.* Adapun proses *preprocessing* diantaranya yaitu proses *case folding*, proses tokenisasi, proses filterisasi dan proses *stem/stemming*(Putra & Sujaini, 2016). Berdasarkan penjelasan sebelumnya, maka dari itu dalam penelitian ini, peneliti bermaksud menganalisa terhadap pengaruh metode *stemming* terhadap kinerja algoritma Rabin-Karp dalam mendeteksi tingkat kemiripan teks.

2. Metode

Tahapan pelaksanaan penelitian dapat dilihat berdasarkan gambar flowchart di bawah ini:



Gambar 1. *Flowchart* Alur Pengembagan Sistem

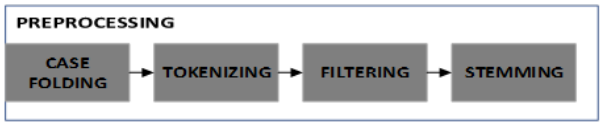
Penjelasan setiap proses *flowchart* adalah sebagai berikut :

2.1. *Data Acquisition*

Data penelitian bersumber dari data judul, abstrak, kata kunci dan pendahuluan. Data ini diperoleh dari jurnal Teknik Informatika Universitas Pamulang.

2.2. *Preprocessing*

Flowchart di bawah ini merupakan tahapan proses *preprocessing.*

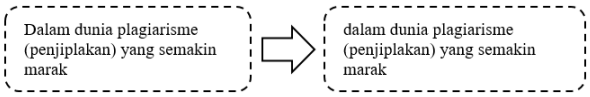


Gambar 2. *Preprocessing*

Penjelasan setiap langkah yang ada pada *flowchart* pada gambar 2 adalah sebagai berikut :

a. Proses *Case Folding*

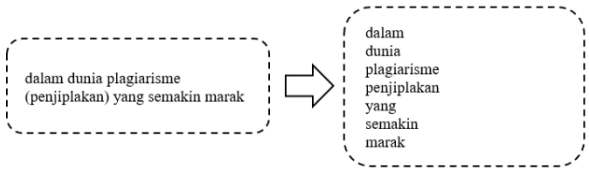
Proses konversi setiap karakter atau huruf kapital atau *uppercase* pada text menjadi *lowercase* atau huruf kecil semua (Hidayatullah, 2015). Di bawah ini contoh proses *case folding*:



Gambar 3. *Case Folding*

b. Proses *Tokenizing*

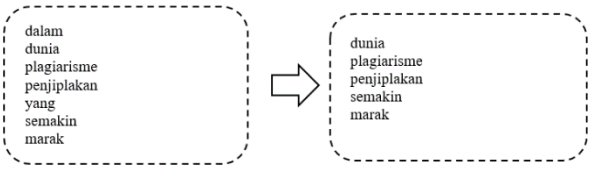
Proses pemecahan *string* atau kalimat menjadi tiap kata penyusunnya serta menghilangnya karakter delimiter (Putra & Sujaini, 2016). Di bawah ini contoh proses *tokenizing*:



Gambar 4. Tokenisasi

c. Filtering

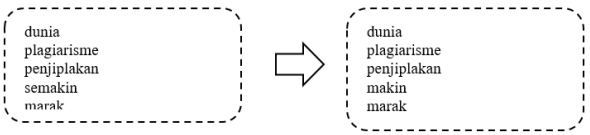
Proses pemilihan kata yang penting dengan menghilangkan kata yang kurang bermakna misalkan konjungsi dan kata ganti (Bhosale & Vankudre, 2017). Berikut ini contoh proses *filtering*:



Gambar 5. *Filtering*

d. Stemming

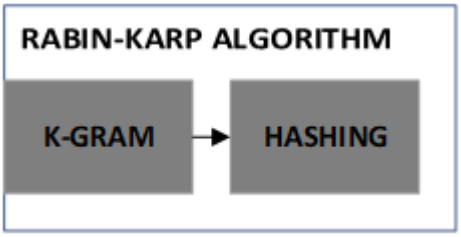
Proses menemukan kata dasar yang digunakan untuk menentukan fitur di dalam teks (Prihatini et al., 2017). *Stemming* Nazief dan Adrian menjadi pilihan dalam penelitian ini untuk menentukan kata dasar. Berikut ini contoh hasil stemmingnya:



Gambar 6. *Proses Stemming*

2.3. *Rabin-Karp Algorithm*

Tahapan proses *Rabin Karp algorithm* dijelaskan berdasarkan *flowchart* di bawah ini:



Gambar 7. Algoritma Rabin-Karp

Penjelasan setiap langkah yang ada pada *flowchart* pada Gambar 7 adalah sebagai berikut:

a. KGram

Pembentukan model kata melalui pemotongan kata sejumlah k secara berulang dimulai dari kata pertama hingga kata terakhir. (Rahmaddeni et al., 2018). Berikut ini contoh penggunaan k-gram dengan nilai k=3 dari teks "duniaplagiarismepenjiplakanmakin
marak" adalah: [dun, uni, nia, iap, apl, pla, lag, agi, gia, iar, ari, ris, ism, sme, mep, epe, pen, enj, nji, jip, ipl, pla, lak, aka, kan, anm, nma, mak, aki, kin, inm, nma, mar, ara, rak].

b. Hashing

Merupakan proses perubahan suatu kata/*string* menjadi angka bernilai special yang memiliki panjang tetap yang berfungsi sebagai fitur suatu kata/*string*. Fungsi yang menghasilkan nilai unik tersebut disebut sebagai fungsi *hash* dan hasilnya sebagai nilai *hash*. Adapun persamaan fungsi *hash* sebagai berikut:

$$H(c_1 \dots c_k) = \left(c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \cdots + c_{(k-1)} * b^{(k)} + c_k\right) mod\ q$$

Keterangan:
H $= substring$
c $= nilai\ ASCII\ setiap\ karakter$
b $= konstanta\ bilangan\ prima$
q $= modulo\ bilangan\ prima$

Berikut ini contoh hasil perhitungan hash dari hasil k-gram sebelumnya dengan nilai parameter b = 397 dan q = 1007 adalah: [580, 543, 973, 262, 63, 177, 797, 515, 375, 264, 854, 38, 356, 115, 897, 110, 432, 328, 371, 934, 171, 177, 801, 81, 287, 277, 547, 311, 89, 442, 385, 547, 318, 846, 882].

2.4. *Similarity Score*

Perhitungan presentase tingkat kemiripan teks pada penelitian ini dengan menggunakan persamaan *Coefficient of Dice's Similarity*. Adapun proses perhintungan melalui perhitungan total KGram semua teks yang akan dilakukan pengujian. Berikuti ini merupakan persamaan untuk menghasilkan nilai tersebut:

$$S = \frac{2C}{A + B}$$

Di mana:
S $= Nilai\ Kemiripan$
C $= Jumlah\ K - Gram\ Sama$
A, B = Total KGram Setiap Kata/String

3. Hasil dan Pembahasan

3.1. Hasil

Proses pengujian pengaruh metode *stemming* terhadap algoritma Rabin-Karp dalam mendeteksi kemiripan teks bahasa Indonesia dilakukan dengan menggunakan data yang berasal dari Jurnal Teknik Informatika Universitas Pamulang. Proses pengujian tingkat kemiripan data dilakukan melalui tiga jenis proses pengujian yaitu dengan melakukan perubahan susunan kata, kalimat dan jurnal yang memiliki topik pembahasan yang mirip.

Pada proses pengujian perubahan susunan kata hanya dilakukan pada bagian judul, abstrak dan kata kunci jurnal. Sedangkan pengujian perubahan susunan kalimat dan topik yang mirip dilakuan pada bagian abstrak dan pendahuluan jurnal. Masing-masing nilai parameter K-Gram, Basis dan Modulo pada Rabin-Karp yang digunakan dalam penelitian ini adalah 3, 397 dan 1007. Proses pengujian pengaruh metode stemming terhadap algoritma Rabin-Karp dimulai dari preprocessing.

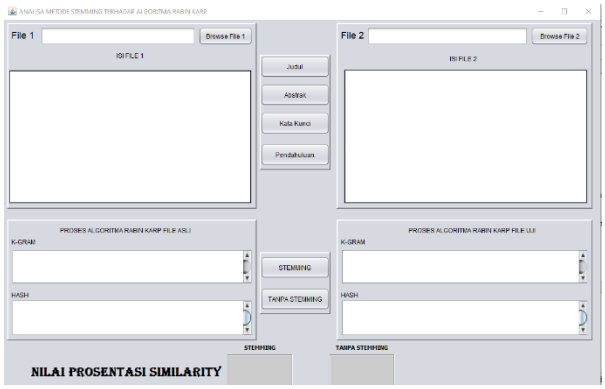
Dalam proses *preprocessing* ini data ada yang mengaami proses stemming dan ada yang tidak dilakukan proses stemming. Setelah proses *preprocessing* selesai dilanjutkan dengan pembentukan data K-Gram. Kemudian dilanjutkan dengan perhitungan nilai hash masing-masing K-Gram dan duplikat hash dengan algoritma Rabin-Karp. Diakhiri dengan presentase kemirpian teks dengan menggunakan persamaan *Dice's Similarity Coefficient*.

Semua proses tersebut telah diimplementasikan dalam bentuk prototipe aplikasi desktop dengan menggunakan bahasa pemrograman java. Adapun tampilan dan tahapan penggunaan

prototipe aplikasi tersebut dapat dilihat sebagai berikut:

a. Tampilan Utama

Setelah program dijalankan maka akan tampil halaman utama prototipe aplikasi.

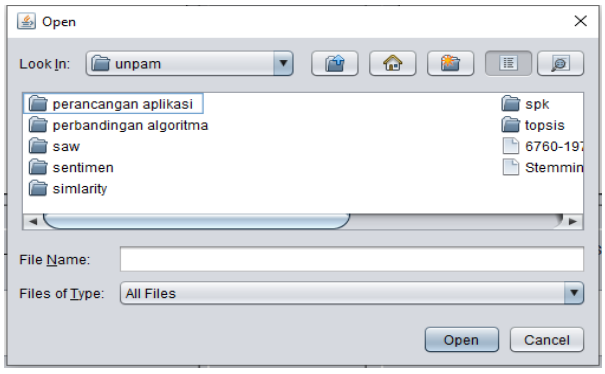


Gambar 8. Halaman Utama Prototipe Aplikasi

User dapat mencari file jurnal yang ada di local komputer dengan mengklik tombol *Browse File*.

b. Tampilan *Browse File*

Adapun tampilan proses pencarian filenya sebagai berikut:

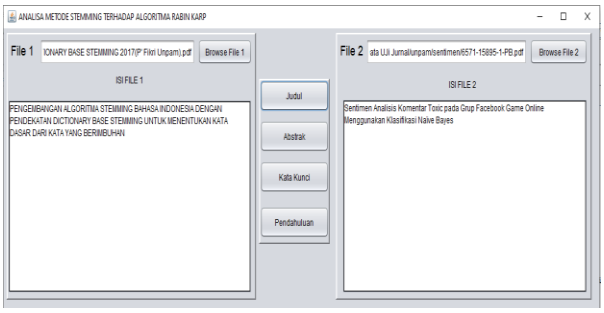


Gambar 9. *Browsing File*

Langkah yang sama dilakukan untuk mencari file kedua yang dijadikan sebagai file uji. Jika kedua teks field sudah berisi alamat file, langkah selanjutnya yaitu melakukan pemilihan bagian jurnal yang akan dilakukan pengujian kemiripan.

c. Tampilan Bagian Teks Jurnal

Di bawah ini merupakan tampilan judul file jurnal yang telah dipilih melalui *browse file* dengan cara mengklik tombol Judul:

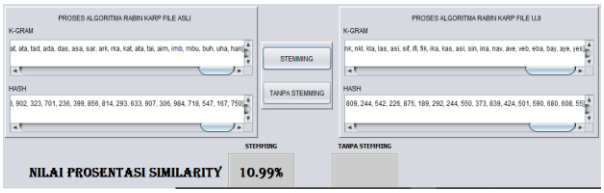


Gambar 10. Teks Judul Jurnal

Untuk menampilkan bagian teks lain seperti Abstrak, Kata Kunci dan Pendahuluan bisa dilakukan dengan cara mengklik salah satu tombol yang dimaksud. Proses perhitungan kemiripannya dilakukan dengan klik tombol *Stemming* ataupun tombol Tanpa *Stemming*

d. Tampilan Hasil Presentase Kemiripan Teks

Berikut ini merupakan tampilan hasil prosentase tingkat kemiripan bagian teks jurnal setelah kita mengklik tombol *Stemming*:



Gambar 11. Tampilan Hasil Presentase Kemiripan Teks

Setelah mengklik tombol *Stemming* maka akan ditampilkan data K-Gram dan *Hash* dari teks bagian jurnal yang sudah dipilih. Ditampilkan juga nilai hasil prosentase kemiripan teksnya. Langkah yang sama dilakukan untuk mencari nilai prosentasi kemiripan teks tanpa menggunakan proses *stemming* yaitu dengan mengklik tombol Tanpa *Stemming*.

3.2. Pembahasan

Adapun hasil proses pengujian tingkat kemiripan data dilakukan melalui dua jenis proses pengujian yaitu dengan melakukan perubahan susunan kata dan perubahan susunan kalimat.

a. Perubahan Susunan Kata

Pada proses pengujian tingkat kemiripan data yang pertama dilakukan dengan membandingkan kedua data yang sama yaitu berupa judul, abstrak dan kata kunci, namun susunan kata mengalami perubahan secara acak. Adapun data dan hasil pengujiannya di bawah ini:

Tabel 1. Hasil Uji Coba Data Berdasarkan Perubahan Susunan Kata

No	Jenis Data Uji	*Stemming*	Tanpa *Stemming*
1	Rata – Rata Hasil Pengujian Judul Jurnal	84,14%	84,75%
2	Rata – Rata Hasil Pengujian Abstrak Jurnal	90,06%	91,60%
3	Rata – Rata Hasil Pengujian Kata Kunci Jurnal	74,73%	74,85%

Berdasarkan dari hasil pengujian di atas menunjukkan bahwa pengujian implementasi metode *stemming* Nazief&Adriani pada algoritma Rabin-Karp cenderung memiliki prosentasi kemiripan lebih rendah dibanding tanpa mengimplementasikan metode *stemming*.

b. Perubahan Susunan Kalimat

Pada proses pengujian tingkat kemiripan data yang kedua dilakukan dengan membandingkan kedua data yang sama yaitu berupa abstrak dan pendahuluan, namun susunan kalimat mengalami perubahan secara acak. Berikut ini hasil pengujiannya :

Tabel 2. Uji Coba Data Berdasarkan Perubahan Susunan Kalimat

No	Jenis Data Uji	*Stemming*	Tanpa *Stemming*
1	Rata – Rata Hasil Pengujian Abstrak Jurnal	90,50%	90,50%
2	Rata – Rata Hasil Pengujian Pendahuluan Jurnal	90,69%	99,70%

Tidak berbeda jauh dengan pengujian sebelumnya, hasil pengujian perubahan susunan kalimat pada abstrak dan pendahuluan juga menunjukkan bahwa pengujian implementasi metode stemming Nazief&Adriani pada algoritma Rabin-Karp cenderung memiliki prosentasi kemiripan lebih rendah dibanding tanpa mengimplementasikan metode stemming. Meskipun memang ada yang memiliki tingkat prosentase kemiripan yang sama juga.

4. Kesimpulan

Hasil pengujian di atas menunjukkan bahwa implementasi metode stemming Nazief&Adrian pada algoritma Rabin-Karp mempengaruhi tingkat prosentase kemiripan teks. Dalam penelitian ini hasil implementasi metode *stemming* mengalami rata – rata penurunan sebesar 0,76% dibandingkan tanpa menggunakan metode *stemming* pada pengujian data yang mengalami perubahan susunan kata secara acak. Seirama dengan pengujian data yang mengalami perubahan susunan kalimat secara acak juga mengalami rata – rata penurunan sebesar 0.04%.

Meskipun hasil penurunan tingkat prosentasenya tidak terlalu signifikan hal tersebut menunjukkan bahwa adanya pengaruh terhadap implementasi algoritma stemming yang diperkenalkan oleh Nazief danAdriani pada metode Rabin Karp dalam mendeteksi tingkat kesamaan teks. Hal tersebut terjadi dikarenakan semakin banyaknya hash yang terbentuk dan duplikat *hash* yang muncul dikarenakan tidak adanya proses penghilangan imbuhan kata atau afiks meliputi prefiks/ awalan kata, infiks/sisipan kata, sufiks/akhiran kata, maupun konfiks/ gabungan awalan dan akhiran kata.

Daftar Pustaka

Bhosale, M. V., & Vankudre, A. A. (2017). Detection of Real-Time Traffic through Twitter Stream Analysis. *International Research Journal of Advanced Engineering and Science*, *2*(2), 124–126.

Christina, S., Oktaviyani, E. D., & Famungkas, B. (2018). Mendeteksi Plagiarism Pada Dokumen Proposal Skripsi Menggunakan Algoritma Jaro Winkler Distance. *Jurnal Saintekom*, *8*(2), 143–153. https://doi.org/https://doi.org/10.33020/saintekom.v8i2.68

Hapsari, R. K., & Santoso, Y. J. (2015). Stemming Artikel Berbahasa Indonesia Dengan Pendekatan Confix-Stripping. *Prosiding Seminar Nasional Manajemen Teknologi*

XXII, 1–8.

Hidayatullah, A. F. (2015). The Influence of Stemming on Indonesian Tweet Sentiment Analysis. *Proceeding of the Electrical Engineering Computer Science and Informatics*, *Vol 2*, 127–132. https://doi.org/http://dx.doi.org/10.11591/eecsi.v2.791

Mardiana, T., Adji, T. B., & Hidayah, I. (2016). Stemming Influence on Similarity Detection of Abstract Written in Indonesia. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *14*(1), 219–227. https://doi.org/http://dx.doi.org/10.12928/telkomnika.v14i1.1926

Nugroho, H. T. (2017). Pengaruh Algoritma Stemming Nazief-Adriani Terhadap Kinerja Algoritma Winnowing Untuk Mendeteksi Plagiarisme Bahasa Indonesia. *Ultima Computing : Jurnal Sistem Komputer*, *9*(1), 36–40. https://doi.org/https://doi.org/10.31937/sk.v9i1.572

Prihatini, P. M., Putra, I. D., Giriantari, I., & Sudarma, M. (2017). Stemming Algorithm for Indonesian Digital News Text Processing. *International Journal of Engineering and Emerging Technology*, *2*(2), 1–7.

Purba, A. H., & Situmorang, Z. (2017). Analisis Perbandingan Algoritma Rabin-Karp Dan Levenshtein Distance Dalam Menghitung Kemiripan Teks. *Jurnal Teknik Informatika UNIKA Santo Thomas*, *2*(2), 24–32. https://doi.org/https://doi.org/10.17605/jti.v2i2.187

Putra, D. A., & Sujaini, H. (2016). Implementasi Algoritma Rabin-Karp untuk Membantu Pendeteksian Plagiat pada Karya Ilmiah. *JUSTIN (Jurnal Sistem Dan Teknologi Informasi)*, *4*(1), 66–74.

Rahmaddeni, Sazali, D., & Agustin. (2018). Sistem Pendeteksi Tingkat Kesamaan Teks pada Pengusulan Proposal Penelitian Internal Menggunakan Algoritma Rabin-Karp. *SATIN - Sains Dan Teknologi Informasi*, *4*(2), 84–92. https://doi.org/https://doi.org/10.33372/stn.v4i2.415

Ruban, S. S., Serrao, S. B., & Harshitha, L. V. (2015). A Study and Analysis of Information Retrieval Models. *International Journal of Innovative Research in Computer and Communication Engineering*, *3*(7), 230–236.

Simarangkir, M. S. H. (2017). Studi Perbandingan Algoritma - Algoritma Stemming untuk Dokumen Teks Bahasa Indonesia. *Jurnal Inkofar*, *1*(1), 40–46.

Verdaningroem, N. J. M., & Saifudin, A. (2018). Penerapan Kamus Dasar pada Algoritma Porter untuk Mengurangi Kesalahan Stemming Bahasa Indonesia. *Jurnal Teknologi*, *10*(2), 103–112. https://doi.org/https://doi.org/10.24853/jurtek.10.2.103-112

Wicaksono, Y. A., & Suyanto. (2012). *Analisis dan Implementasi Algoritma Rabin-Karp dan Algoritma Stemming Nazief-Adriani pada Sistem Pendeteksi Plagiat Dokumen Teks Berbahasa Indonesia*. Universitas Telkom.

Yulianingsih. (2017). Implementasi Algoritma Jaro-Winkler dan Levenstein Distance dalam Pencarian Data pada Database. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, *2*(1), 18–27. https://doi.org/https://doi.org/10.30998/string.v2i1.1720

Yulianto, M. A., & Nurhasanah, N. (2021). The Hybrid of Jaro-Winkler and Rabin-Karp Algorithm in Detecting Indonesian Text Similarity. *Jurnal Online Informatika*, *6*(1), 88. https://doi.org/10.15575/join.v6i1.640