

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Dewasa ini, informasi yang kita lihat di internet khususnya artikel ilmiah sering kali mempunyai struktur kata atau penulisan yang mirip dengan artikel ilmiah yang lain. Sebagai contoh adalah artikel-artikel ilmiah yang sering kita jumpai di internet. Hal ini menyebabkan rancunya sumber informasi sehingga pembaca bingung akan sumber informasi asal, begitupun dengan para penulis artikel ilmiah yang menginginkan karya tulisnya agar tidak mirip dengan karya tulis lain sehingga penulis artikel ilmiah tidak melakukan tindakan plagiarisme dengan tidak disengaja. Untuk mengetahui sebuah artikel mirip dengan artikel yang lain maka dibutuhkan sebuah metode *stemming* yang dapat memangkas kata-kata menjadi kata dasarnya untuk menemukan kemiripan kata pada artikel yang dicocokkan dan untuk mengetahui seberapa mirip artikel yang dicocokkan maka dibutuhkan suatu persamaan untuk menghitung bobot kemiripan (*similarity*).

Stemming merupakan proses pemangkasan suatu kata untuk menemukan kata dasarnya (*root*). Tujuan dari proses *stemming* adalah untuk menghilangkan kata-kata imbuhan (*affixes*) baik itu imbuhan awalan (*prefix*) maupun imbuhan akhiran (*suffix*). *Stemming* sangat berguna untuk proses *indexing* maupun *searching* di dalam *Information Retrieval* (Temu Kembali Informasi). Algoritma *stemming* bahasa Indonesia yang cukup populer adalah algoritma Porter dan algoritma Nazief dan Adriani. Algoritma Porter dalam proses *stripping* tidak menggunakan kamus kata dasar, berbeda dengan algoritma Nazief dan Adriani yang dalam proses *stripping*-nya membutuhkan kamus kata dasar. Pada sebuah penelitian yang membandingkan kedua algoritma tersebut [1] disebutkan bahwa dari total 2.132 kata yang diuji, tingkat kesalahan *stemmer* Nazief dan Adriani hanya 5%, sangat jauh jika dibandingkan dengan Porter yaitu 25%. Akan tetapi *stemmer* Nazief dan Adriani membutuhkan waktu proses hampir dua kali lipat dibandingkan dengan Porter. Dari 2.132 kata yang di *stem* menggunakan algoritma Nazief dan Adriani menghasilkan 2.031 kata dasar yang benar dan 101 kata (5%) yang mengalami kesalahan *stemming*, terdiri dari 23 *understemming*, 27 *overstemming* dan 51 *unchange*. Sedangkan algoritma Porter menghasilkan kata dasar yang benar sebanyak 1.168 kata dan kesalahan *stemming* sebanyak 445 kata (21%), terdiri dari 67 *understemming*, 374 *overstemming*

dan 4 *unchange*. Berdasarkan hasil yang didapat pada kedua *stemmer* yang diujikan, dapat dihitung tingkat akurasi dari kedua *stemmer* seperti ditunjukkan pada tabel 1.1.

Tabel 1.1 Perbandingan Hasil Stemming Porter dan Nazief & Adriani

Algoritma	Akurasi	Waktu Proses
Porter	79,13%	12.3822753429
Nazief dan Adriani	95,26%	22.1668348312

Dari hasil yang tampak pada tabel 1.1, dapat disimpulkan bahwa algoritma Nazief & Adriani memberikan hasil yang lebih baik untuk *stemming* pada dokumen berbahasa Indonesia dalam hal akurasi, namun dalam waktu proses algoritma Nazief dan Adriani membutuhkan waktu proses hampir dua kali lipat lebih lama dibandingkan algoritma Porter. Untuk mengurangi waktu proses *stemming*, dibutuhkan sebuah metode *filtering* yang terdapat dalam *text preprocessing*.

Text preprocessing merupakan tahapan seleksi data yang akan diproses pada setiap dokumen agar data yang akan diolah menjadi lebih terstruktur. *Filtering stopwords* merupakan suatu tahap yang terdapat pada *text preprocessing*, proses yang dilakukan adalah membuang semua kata-kata tidak penting atau umum yang terdapat dalam teks. Sehingga dapat mengurangi waktu proses *stemming* algoritma Nazief dan Adriani.

Sedangkan untuk mengetahui tingkat kemiripan artikel, menggunakan metode *Dice Similarity Coefficient*. metode *Dice Similarity Coefficient* atau dikenal dengan nama *Sorensen-Dice Coefficient* merupakan metode yang digunakan dalam perhitungan kemiripan dokumen dengan membandingkan dua sampel teks yang berbeda. Jika nilai bobot yang dihasilkan bernilai 0 maka kedua teks yang dibandingkan tidak memiliki kemiripan sama sekali, dan jika nilai bobot yang dihasilkan mendekati 1 maka kedua teks yang dibandingkan semakin mirip.

Berdasarkan pemaparan dan tingkat akurasi *stemming* algoritma Nazief dan Adriani pada tabel 1.1 di atas, maka penulis mengusulkan judul “Penerapan Metode *Dice Similarity Coefficient* Untuk Mendeteksi Kemiripan Artikel Berbasis Desktop”.

1.2 Rumusan Masalah

Perumusan masalah berdasarkan latar belakang di atas adalah sebagai berikut:

1. Bagaimana menerapkan metode *Dice Similarity Coefficient* pada aplikasi berbasis desktop untuk mengetahui persentase tingkat kemiripan artikel ilmiah sehingga dapat digunakan oleh penulis artikel dalam membantu menjauhkan penulis artikel dari tindakan plagiarisme.
2. Apakah metode *filtering stopword* dapat mempengaruhi bobot kemiripan yang dihasilkan perhitungan metode *Dice Similarity Coefficient*?

1.3 Batasan Masalah

Batasan masalah digunakan untuk menghindari pelebaran pokok masalah maka beberapa batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Metode yang digunakan pada penelitian ini adalah metode *stemming* Nazief dan Adriani.
2. Metode yang digunakan untuk menemukan kemiripan adalah metode *Dice Similarity Coefficient*.
3. Penelitian berfokus pada aplikasi berbasis desktop menggunakan *software* Microsoft Visual Studio dan SQLiteStudio.
4. Artikel yang dibandingkan adalah artikel ilmiah berbahasa Indonesia.
5. Artikel ilmiah yang diinputkan berupa file berformat PDF.
6. Maksimal kata pada artikel yang diinputkan berjumlah 600 kata.

1.4 Tujuan Penelitian

Adapun tujuan penelitian pada kali ini adalah sebagai berikut:

1. Melakukan implemmentasi metode *Dice Similarity Coefficient* pada aplikasi berbasis desktop untuk mengetahui bobot kemiripan artikel ilmiah sehingga dapat membantu penulis artikel ilmiah agar terhindar dari tindakan plagiarisme.
2. Mengetahui pengaruh metode *filtering stopword* terhadap bobot kemiripan yang dihasilkan perhitungan metode *Dice Similarity Coefficient*.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian yang dilakukan adalah sebagai berikut:

1. Membantu penulis artikel ilmiah dalam mendeteksi kemiripan isi teks pada artikel yang ditulis dengan artikel serupa.
2. Sebagai dasar penelitian lain yang menggunakan metode *stemming* khususnya metode Nazief dan Adriani.
3. Hasil penelitian diharapkan dapat digunakan sebagai stimulus bagi peneliti yang tertarik dalam bidang *information retrieval*.

1.6 Metode Penelitian

Penelitian ini menggunakan metode penelitian Research and Development (penelitian dan pengembangan). Metode ini adalah metode penelitian yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut. Langkah-langkah yang dilakukan peneliti adalah sebagai berikut:

1.6.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini meliputi beberapa tahap yaitu:

1. Studi Literatur

Studi Literatur adalah metode pengumpulan data dengan cara mengumpulkan data dengan mengumpulkan literatur, jurnal, paper, bacaan-bacaan, artikel yang berkaitan dengan penelitian yang dilakukan.

2. Studi Pustaka

Mempelajari data-data dan arsip yang sudah ada sebagai acuan atau bahan referensi dalam memperoleh data informasi yang dibutuhkan.

1.6.2 Metode Analisis

Metode analisis yang digunakan dalam penelitian ini yaitu analisis kebutuhan sistem yang meliputi kebutuhan fungsional dan non-fungsional dan analisis kelayakan sistem yang meliputi kelayakan teknologi, operasional dan hukum.

1.6.3 Metode Perancangan

Metode perancangan yang dilakukan yaitu membuat prototype terlebih dahulu dengan melakukan simulasi desain kebutuhan sistem seperti interface dan flowchart.

1.6.4 Metode Pengembangan

Pada tahap metode pengembangan akan mulai membuat kode program berdasarkan desain yang telah dibuat sebelumnya.

1.6.5 Metode Testing

Pengujian yang dilakukan pada aplikasi ini yaitu pengujian akurasi sistem menggunakan metode *black box* dan *white-box* untuk mengetahui berapa banyak data yang sesuai antara hasil berdasarkan perhitungan manual dengan hasil perhitungan oleh sistem.

1.7 Sistematika Penulisan

Penulisan skripsi ini ditulis secara sistematis yang terdiri dari beberapa sub bab. Berikut uraian sistematika penulisan:

1.7.1 BAB I : Pendahuluan

Bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penulisan.

1.7.2 BAB II : Landasan Teori

Pada bab ini dijelaskan tentang teori, definisi-definisi, atau model yang berkaitan dengan ilmu dan masalah yang sedang diteliti.

1.7.3 BAB III : Anallsa dan Perancangan Sistem

Bab ini menjelaskan tentang analisis permasalahan dalam pembuatan aplikasi perangkat lunak serta menjelaskan tentang rancangan struktur program serta merancang interface dari perangkat lunak yang akan dibuat.

1.7.4 BAB IV : Implementasi dan Pembahasan

Bab ini menjelaskan tentang pengimplementasian algoritma Nazief dan Adriani pada aplikasi berbasis desktop serta pengujian aplikasi menggunakan metode *white-box* dan *black-box*.

1.7.5 BAB V : Penutup

Bab ini merupakan bab terakhir yang berisi kesimpulan yang diperoleh dari perumusan masalah yang telah disampaikan, serta saran yang membangun untuk pengembangan diri.