

ANALISIS DAN IMPLEMENTASI ALGORITMA RABIN-KARP DALAM ALAT BANTU PENDETEKSIAN PLAGIARISME TEKS

Gita Bahana Firdaus¹, Z.k. Abdurahman Baizal², Rimba Widhiana Ciptasari³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Plagiarisme teks semakin mudah dilakukan. Untuk mengatasinya, setiap teks perlu diperiksa apakah hasil plagiarisme atau bukan. Akan tetapi pemeriksaan ini memerlukan waktu yang tidak sedikit jika teks yang harus diperiksa jumlahnya banyak. Untuk itu perlu sebuah sistem yang dapat menemukan kemiripankemiripan diantara teks dan membantu manusia dalam mendeteksi plagiarisme teks ini.

Sistem ini dapat dibangun dengan mengimplimentasikan algoritma Rabin- Karp. Untuk mendeteksi adanya kemiripan, dua teks dibandingkan dengan menganggapnya sebagai string yang hanya terdiri dari 26 huruf. Setiap substring dari kedua teks dibandingkan dengan terlebih dahulu dihitung nilai hash substring-substring tersebut.

Dengan algoritma ini, sistem dapat menemukan kata-kata yang benar-benar sama atau yang berubah bentuk dari kata benda menjadi kata kerja dan sebagainya dalam dua teks yang memiliki kemiripan. Walaupun tidak dapat menemukan perubahan kata menjadi kata lain yang bermakna sama, kemiripan yang ditemukan sistem hampir serupa dengan kemiripan yang ditemukan manusia. Sehingga dengan kemiripan ini, pendeteksiian plagiarisme teks dapat dilakukan lebih cepat.

Kata Kunci : algoritma Rabin-Karp, string, substring, nilai hash

Abstract

Nowadays, text plagiarism become easier. To solve this problem, each text have to be checked whether the text made with plagiarism. This is a time consuming job if the amount of text is abundance. But this time consumption problem can be solved using a system that can find any similarity between texts, so the output of the system can be used in plagiarism detection.

This system can built by implementing Rabin-Karp algorithm. To find any similarity, two texts compared each others by consider each text as a string containing only 26 letters. The hash value of every substring from each text compared to know whether there are any words similarity.

By using this algorithm, the sistem can find any words that absolutely same and any words that change into others form (from noun to verb, adjective, and etc.) between two texts that containing similarity. Although the system can't find the change of vocabulary, the similarity that found by system is almost similar with the human done. Therefore, with this similarity, text plagiarism detection can be done faster.

Keywords : Rabin-Karp algorithm, string, substring, hash value

1. Pendahuluan

1.1 Latar belakang

Semakin berkembangnya teknologi komputerisasi, semakin mudah seseorang menyelesaikan pekerjaannya atau menghasilkan sesuatu. Dengan menggunakan komputer, seseorang dapat menciptakan sebuah hasil karya berdasarkan referensi-referensi dari internet atau yang telah tersimpan dalam media penyimpanan komputer tersebut. Namun kemudahan ini memicu tindakan yang seharusnya dihindari menjadi semakin mudah dilakukan, yaitu tindakan plagiarisme.

Plagiarisme adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya, dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri[1]. Plagiarisme banyak terjadi dalam pembuatan teks, baik berupa tugas-tugas akademis atau artikel-artikel yang dipublikasikan dalam internet. Plagiarisme seperti ini semakin mudah dilakukan dengan adanya fasilitas *copy-and-paste* dalam komputer.

Untuk mencegah plagiarisme teks, banyak upaya yang dilakukan, salah satunya dengan memeriksa hasil karya seseorang secara manual, sebelum dinilai atau dipublikasikan. Jika sebuah teks dideteksi sebagai hasil plagiat, bukan hanya teks tersebut tidak akan dinilai dan dipublikasikan, tetapi pembuat teks juga dihukum sesuai ketentuan-ketentuan yang berlaku. Hal ini dilakukan agar timbul efek jera pada diri pelaku dan untuk mencegah orang lain melakukan tindakan yang sama.

Akan tetapi pemeriksaan teks secara manual untuk mendeteksi plagiarisme bisa menjadi pekerjaan yang sulit jika terdapat banyak teks yang harus diperiksa. Dibutuhkan waktu yang cukup lama untuk melakukan pemeriksaan seluruh teks tersebut. Oleh karena itu beberapa lembaga mencoba mengembangkan sebuah sistem yang dapat mendeteksi plagiarisme secara otomatis.

Namun sistem yang dapat mendeteksi plagiarisme seperti ini cukup sulit untuk direalisasikan. Hal ini karena adanya masalah-masalah hukum yang berkaitan dengan *fair use*, yang membolehkan penggunaan bahan-bahan yang telah dilindungi hak cipta dalam karya penulis lain dengan syarat-syarat tertentu [2].

Berdasarkan alasan tersebut, sistem seperti ini lebih tepat dianggap sebagai alat bantu pendeteksian plagiarisme. Disebut alat bantu karena sistem ini hanya mendeteksi kemiripan-kemiripan antara teks yang dibandingkan. Sedangkan pengkategorian plagiarisme dilakukan secara manual berdasarkan kemiripan-kemiripan tersebut.

Tugas akhir ini menganalisa pembuatan sistem yang berupa alat bantu pendeteksian plagiarisme teks dengan menerapkan algoritma Rabin-Karp. Algoritma ini adalah algoritma yang dapat menemukan posisi beberapa *string* berukuran sama dalam *string* yang lebih besar. Karena teks juga dapat dianggap sebagai *string*, maka algoritma ini dapat diterapkan sebagai bagian dari alat bantu pendeteksian plagiarisme teks. Dengan penerapan algoritma ini, diharapkan sistem dapat menemukan kemiripan-kemiripan dalam teks, sehingga mempermudah pendeteksian plagiarisme secara manual.

1.2 Perumusan masalah

Permasalahan yang muncul dan akan dianalisa dalam tugas akhir ini adalah sebagai berikut:

1. Masukan algoritma Rabin-Karp adalah beberapa *string* (himpunan *string*) berukuran sama dan sebuah *string* lain. Bagaimana membuat sebuah teks menjadi sebuah himpunan *string* berukuran sama sehingga dapat dijadikan sebagai masukan algoritma Rabin-Karp.
2. Tidak ada ketentuan khusus untuk ukuran *string* dalam himpunan *string*. Bagaimana menentukan ukuran *string* yang baik, karena berdasarkan analisa awal, algoritma Rabin-Karp mungkin menghasilkan keluaran yang berbeda untuk ukuran *string* yang berbeda.
3. Keluaran algoritma Rabin-Karp adalah posisi-posisi setiap himpunan *string* dalam *string* lain. Bagaimana mengolah keluaran ini untuk dapat menemukan kemiripan antara teks yang dibandingkan.
4. Berdasarkan hasil analisa awal, terdapat beberapa perubahan dalam teks yang umumnya dilakukan pelaku plagiarisme (dijelaskan dalam bab 2). Bagaimana sistem menemukan perubahan tersebut dalam teks sehingga dapat menentukan apakah sebuah teks memiliki kemiripan dengan teks lain.

Adapun yang menjadi batasan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Sistem hanya difokuskan untuk menemukan kemiripan dalam teks-teks yang ditulis dalam bahasa Indonesia.
2. Masukan sistem adalah dua buah teks yang berupa *file* berekstensi *‘.txt’*.
3. Sistem tidak memeriksa kesalahan penulisan kata dalam teks.
4. Sistem tidak memeriksa ada tidaknya *stopword* dalam teks. *Stopword* tetap diikutsertakan dalam penentuan kemiripan.
5. Selain algoritma Rabin-Karp dan metode-metode yang mendukung penggunaannya (diantaranya *hashing*, *sorting*, dan *searching*), serta metode yang dibuat untuk mengolah keluaran algoritma menjadi keluaran sistem, dalam sistem ini tidak diterapkan metode-metode spesifik lain yang mungkin dapat digunakan dalam menemukan kemiripan diantara teks.

1.3 Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini adalah:

1. Menganalisa pengaruh penggunaan ukuran *string* yang berbeda terhadap keluaran sistem, dan menentukan ukuran *string* yang menghasilkan keluaran sistem yang optimal.
2. Menentukan perubahan-perubahan apa saja (poin 4 perumusan masalah) yang dapat ditemukan sistem sehingga dapat menentukan apakah dua teks yang dibandingkan memiliki kemiripan.
3. Menentukan batas tingkat kemiripan yang menjadi acuan sistem dalam menentukan kemiripan (dijelaskan dalam bab 3), sehingga dengan batas tingkat kemiripan ini sistem dapat digunakan dalam berbagai kondisi kemiripan dalam teks.
4. Menganalisa apakah kemiripan yang ditemukan sistem serupa dengan hasil pemeriksaan manual.

1.4 Metodologi penyelesaian masalah

Metodologi penyelesaian masalah pada tugas akhir ini adalah sebagai berikut:

1. Studi Literatur
Menghimpun dan mempelajari teori-teori mengenai plagiarisme, seperti apa pada umumnya teks hasil plagiarisme, bagaimana mendeteksinya, dan bagaimana cara kerja algoritma Rabin-Karp.
2. Perancangan sistem
Merancang cara kerja sistem memeriksa teks dan menemukan kemiripan dengan teks lain, dan bagaimana sistem menyajikan hasil pemeriksaan tersebut.
3. Implementasi sistem
Membangun sistem sesuai hasil perancangan agar dapat melakukan pengujian dan menganalisa hasil pengujian tersebut.
4. Pengujian dan penganalisaan hasil pengujian
Pengujian dilakukan dengan memeriksa apakah dua teks memiliki kemiripan dengan menggunakan sistem dan secara manual. Hasil pengujian tersebut kemudia dianalisa agar dapat ditarik kesimpulan dari hasil pengujian sistem .
5. Penyimpulan hasil penelitian
Kesimpulan diambil dari analisis hasil pengujian. Dengan kesimpulan ini, tujuan-tujuan dari penganalisaan dan pengimplementasian algoritma Rabin-Karp dalam alat bantu pendeteksian plagiarisme teks dapat tercapai.
6. Penyusunan laporan
Laporan digunakan untuk memaparkan analisis, hasil-hasil pengujian, serta kesimpulan yang telah diambil dari analisis dan hasil-hasil pengujian tersebut.

5. Penutup

5.1 Kesimpulan

Dari hasil analisis dan pengimplementasian algoritma Rabin-Karp pada aplikasi alat bantu pendeteksian plagiarisme, dapat diambil beberapa kesimpulan sebagai berikut:

1. Sebagai algoritma *string searching*, Algoritma Rabin-Karp mudah diimplementasikan dalam alat bantu pendeteksian plagiarisme karena dapat menentukan sendiri *string-string* yang digunakan dalam menemukan kemiripan antara dua buah teks.
2. Ukuran *substring* yang digunakan mempengaruhi keluaran algoritma Rabin-Karp dan mempengaruhi keluaran alat bantu secara keseluruhan. Hal ini karena algoritma tidak dapat menemukan kata-kata yang sama dan berdiri sendiri dalam dua teks yang dibandingkan, yang jumlah hurufnya lebih sedikit dari ukuran *substring*.
3. Untuk menemukan kemiripan antara dua teks dalam bahasa Indonesia, ukuran *substring* tiga adalah ukuran yang paling baik karena membuat alat bantu menghasilkan persentase kemiripan lebih besar dibandingkan ukuran *substring* yang lebih besar.
4. Dengan algoritma Rabin-Karp ini, alat bantu hanya dapat menemukan kata-kata yang benar-benar sama dan kata-kata yang hanya mengalami perubahan bentuk diantara dua teks.
5. Alat bantu ini tidak dapat menemukan kemiripan antara dua teks yang sebenarnya salah satunya merupakan hasil plagiarisme, jika pelaku plagiat merubah seluruh kata dengan kata lain sehingga kemiripannya tidak kentara.

5.2 Saran pengembangan

1. Penggunaan kamus kata pada tugas akhir ini membuat kamus kata tersebut akan menjadi *file* yang berukuran sangat besar karena setiap kata menyimpan kata lain, bukan posisi kata lain dalam kamus. Oleh karena itu perlu dirancang penyimpanan kata dalam kamus kata yang lebih efisien.
2. Selain penggunaan kamus kata ini, dengan memperhitungkan *stopword* dalam teks, keluaran sistem akan lebih membantu manusia dalam mendeteksi plagiarisme.

Referensi

- [1] _____, Copyright Law of the United States of America, <http://www.copyright.gov/title17/92chap1.html#107>, diunduh pada tanggal 2 Maret 2009.
- [2] _____,TA, Skripsi atau Plagiat, 2008, http://www.akmi-baturaja.ac.id/index.php?option=com_content&task=view&id=83&Itemid=14, diunduh pada 13 Februari 2009.
- [3] _____, "Turnitin" Mengincar Plagiator
- [4] Andres, Nicolas, Christopher, Hadi Saloko, 2008, "Penelaahan Algoritma Rabin-Karp dan Perbandingan Prosesnya dengan Algoritma Knut-Morris-Pratt", Laboratorium Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika, Institut Teknologi Bandung.
- [5] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 2001, "*Introduction to Algorithms*", Second Edition, MIT Press and McGraw-Hill.
- [6] Clough, paul, 2000, "Plagiarism in natural and programming languages: an overview of current tools and technologies", Department of Computer Science University of Sheffield.
- [7] Hexam, Irving, Academic Plagiarism Defined, 2005, <http://www.ucalgary.ca/~hexham/study/plag.html>, diunduh pada tanggal 13 Februari 2009.
- [8] Kamus Besar Bahasa Indonesia, 1997: 775
- [9] Rabin, Michael O. dan Richard M. Karp, 1987, "*Efficient Randomized Pattern-matching Algorithms*", International Business Machines Corporation.
- [10] Salmela, Leena, J. Tarhio, J. Kytöjoki, 2006, "*Multi-pattern String Matching With Q-grams*", ACM Journal of Experimental Algorithmics.
- [11] Utorodewo, Felicia, dkk. 2007, "Bahasa Indonesia: Sebuah Pengantar Penulisan Ilmiah", Jakarta: Lembaga Penerbit FEUI.

Telkom
University