

PENERAPAN ALGORITMA RABIN-KARP DENGAN PENDEKATAN SYNONYM RECOGNITION SEBAGAI ANTISIPASI PLAGIARISME PADA PENULISAN SKRIPSI

Noviardi Prima Putra ¹⁾, Sularno ²⁾

¹Vokasi Universitas Baiturrahmah

Email : noviardi@unbrah.ac.id

²Prodi Sistem Informasi Universitas Dharma Andalas

Email : soelarno@unidha.ac.id

Abstrak - Plagiarisme dalam penulisan karangan ilmiah saat ini sangat mudah dilakukan. Untuk mengatasinya, setiap karya ilmiah tersebut perlu diperiksa apakah hasil plagiarisme atau bukan dengan memeriksa setiap jengkal teks yang ada. Sistem ini dibangun dengan mengimplementasikan algoritma Rabin-Karp. Untuk mendeteksi adanya kemiripan, dua teks dibandingkan dengan mentransformasikan ke dalam bentuk deretan angka yang mengacu pada tabel ASCII, disebut juga dengan proses *hashing*. Algoritma ini dikombinasikan dengan metode *Synonym Recognition* sehingga dapat menemukan kata-kata yang benar-benar sama atau kata yang sudah dirubah kebentuk kata yang bermakna sama.

Keywords : Plagiarisme, Algoritma *Rabin-Karp*, Metode *Synonym Recognition*

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi yang sangat pesat saat ini, khususnya di bidang teknologi dan internet, banyak dampak positif yang dirasakan dari kemajuan teknologi ini. Namun tidak sedikit pula dampak negatif yang hampir tidak bisa dihindari. Salah satunya adalah tindakan Plagiarisme.

Kebiasaan ini tidak hanya terjadi di kalangan mahasiswa tingkat sarjana saja, di kalangan para akademisi senior yang seharusnya terhormat karena derajat keilmuannya, kebiasaan copy-paste ini seolah-olah lumrah dan dapat dimaklumi.

Dengan memanfaatkan metode pencocokan string pada sebuah dokumen, dapat dikembangkan sebuah aplikasi pendeteksi plagiarisme. Algoritma yang dipakai untuk pendeteksian plagiarisme ada bermacam – macam, antara lain algoritma *Winnowing*, *Boyer-Moore*, *Brute Force*, *Knuth-Morris-Pratt*, *Rabin Karp*, *Smith-Waterman* dan lain-lain.

1.2 Perumusan Masalah

Rumusan masalah pada penelitian ini adalah :

1. Bagaimana membuat sebuah sistem yang dapat melakukan pendeteksian plagiarisme terhadap sebuah skripsi?
2. Bagaimana menentukan sebuah skripsi termasuk dalam hasil plagiarisme atau bukan?
3. Bagaimana sistem yang diterapkan mengenali kata-kata yang memiliki kesamaan makna (*Synonym Recognition*)?
4. Bagaimana menerapkan algoritma *Rabin-karp* yang akan digunakan dalam pengembangan aplikasi pendeteksian plagiat?

1.3 Hipotesis

Hipotesis atau sering disebut juga dengan dugaan awal terhadap penelitian

yang akan dibuat. Berdasarkan perumusan masalah di atas, maka diperoleh hipotesis yang nantinya dapat mengatasi masalah dalam plagiarisme :

1. Sebagai alat bantu dalam menentukan sebuah skripsi adalah hasil dari tindakan plagiarisme;
2. Dapat menentukan persentase kemiripan antara skripsi yang diuji dengan skripsi yang dijadikan sebagai bahan perbandingan oleh sistem ini;
3. Sebagai bahan referensi untuk penelitian selanjutnya.

LANDASAN TEORI

2.1.1 Algoritma Rabin-Karp

Algoritma Rabin Karp adalah algoritma pencocokan string yang menggunakan fungsi hash sebagai pembanding antara string yang dicari (m) dengan substring pada teks (n). Apabila hash value keduanya sama maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya (Firdaus, 2008 dalam Noprisson et al., 2013).

Algoritma Rabin Karp ditemukan oleh Michael O. Rabin dan Richard M. Karp. Teori ini jarang digunakan untuk mencari kata tunggal, namun cukup penting dan sangat efektif bila digunakan untuk pencarian jamak (Atmopawiro, 2006 dalam Salmuasih dan Sunyoto, 2013).

2.1.2 Prinsip Kerja Algoritma Rabin Karp

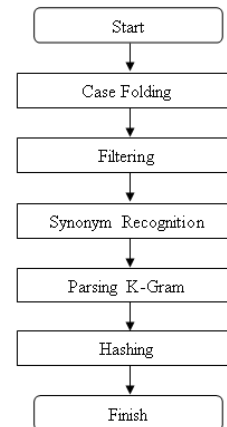
Rabin Karp merepresentasikan setiap karakter ke dalam bentuk desimal digit (digit radix-d) $\Sigma = \{0, 1, 2, 3, \dots, d\}$, dimana $d = |\Sigma|$. Sehingga didapat masukan string k berturut-turut sebagai perwakilan panjang k desimal. Karakter string 31415 sesuai dengan jumlah desimal 31,415. Kemudian pola p di-hash menjadi nilai desimal dan

string direpresentasikan dengan penjumlahan digit-digit angka menggunakan aturan Horner's, misal (Elchison, 2012 dalam Salmuasih dan Sunyoto, 2013) :

- $$\{ A, B, C, \dots, Z \} \square \{ 0, 1, 2, \dots, 26 \}$$
- BAN : $1 + 0 + 13 = 14$
 - CARD : $2 + 0 + 17 + 3 = 22$

2.1.3 Preprocessing

Tahap ini melakukan analisis semantik (kebenaran arti) dan sintaktik (kebenaran susunan) teks. Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Operasi yang dapat dilakukan pada tahap ini meliputi part-of-speech (PoS) tagging, menghasilkan parse tree untuk tiap-tiap kalimat, dan pembersihan teks. (Nugroho, 2012 dalam Surahman, 2013).



Gambar 2.1 Alur Preprocessing

2.1.4 Case Folding

Case folding adalah merupakan tahapan yang mengubah semua huruf dalam dokumen atau isi konten yang terdapat pada suatu web menjadi huruf kecil. Hanya huruf „a“ sampai dengan „z“ yang diterima. Karakter selain huruf dihilangkan dan akan dilakukan proses penghapusan. Pada penelitian ini case folding yang diterima adalah huruf “a”

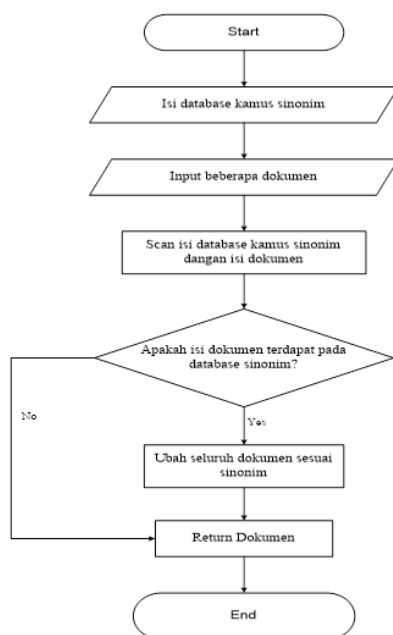
sampai dengan “z”,selain itu seperti tanda hubung atau tanda baca akan dihapus dan dihilangkan (Putra, 2013).

2.1.5 Filtering

Filetring adalah tahap mengambil kata – kata penting dari hasil token. Bisa menggunakan algoritma stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata penting). Stoplist / stopword adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words. Contoh stopwords adalah “yang”, “dan”, “di”, “dari” dan seterusnya (Langgeni, et al., 2010).

2.1.6 Synonym Recognition

Synonym Recognition atau pengenalan sinonim adalah pendeteksian plagiarisme melalui pendekatan sinonim. Dalam hal ini dokumen satu dibandingkan dengan dokumen lain dengan mendeteksi kata-kata yang mengandung sinonim sehingga tingkat kesamaan dapat dikatakan lebih akurat.



Gambar 2.2 Synonym Recognition Flowchart

Pada proses pencocokan sinonim, kaitannya antara isi dokumen dengan kamus sinonim yang terdapat pada database. Jika kata tidak tersedia pada kamus sinonim, maka tidak akan melakukan proses synonym recognition. Seluruh isi dokumen akan discan dan dicocokkan dengan kata yang ada dalam kamus, kemudian akan diubah berdasarkan isi kamus sinonim tersebut (Pratama et al., 2013).

2.1.7 Parsing K-grams

K-grams adalah rangkaian terms dengan panjang K. Kebanyakan yang digunakan sebagai terms adalah kata. K-gram merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode K-grams ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen.

Berikut ini adalah contoh K-grams dengan k=5 :

Teks : “A do run run run ,a do run run” kemudian dilakukan penghilangan spasi “adorunrunrunadorunrun” sehingga dihasilkan rangkaian 5-grams yang diturunkan dari teks : “adoru dorun orunr runru unrun nrunr runru unrun nruna runad unado nador adoru dorun orunr runru unrun” (Schleimer, et al., 2003 dalam Noprisson, et al., 2013).

2.1.8 Hashing

Hashing adalah suatu cara untuk mentransformasikan sebuah string menjadi suatu nilai yang unik dengan panjang tertentu (fixed-length) yang berfungsi sebagai penanda string tersebut. Hash

function atau fungsi hash adalah suatu cara menciptakan fingerprint dari berbagai data masukan. Hash function akan mengganti atau akan mentranspose-kan data tersebut untuk menciptakan fingerprint yang biasa disebut hash value. Contoh sederhana hashing adalah :

Fidau, Hari
Munir, Rinaldi
Rabin, Michael
Karp, Richard
Menjadi :

7864 = Firdaus, Hari
9802 = Munir, Rinaldi
1990 = Rabin, Michael
8822 = Karp, Richard

Nilai hash pada umumnya digambarkan sebagai fingerprint yaitu suatu string pendek yang terdiri atas huruf dan angka yang terlihat acak (**Firdaus, 2008 dalam Noprisson, et al., 2013**).

2.1.9 Rolling Hash

Algoritma Rabin-Karp menggunakan fungsi hash yang disebut dengan rolling hash untuk menentukan apakah kata-kata yang dicocokkan sama. Rolling hash adalah sebuah fungsi hash yang input-nya dikelompokkan ke dalam suatu blok yang digerakkan melewati input secara keseluruhan. Beberapa fungsi hash memungkinkan rolling hash untuk dikomputasi dengan cepat. Nilai hash yang baru dapat dengan cepat dihitung dari nilai hash yang lama dengan cara menghilangkan nilai lama dari kelompok hash dan menambahkan nilai baru ke dalam kelompok tersebut (**Surahman, 2013**).

Kunci dari performa algoritma Rabin-Karp adalah komputasi yang efektif dari nilai hash dari substring-substring yang berurutan pada teks. Algoritma Rabin-Karp melakukan perhitungan nilai hash dengan memperlakukan setiap substring sebagai

sebuah angka dengan basis tertentu, di mana basis yang digunakan pada umumnya merupakan bilangan satu digit atau lebih.

Nilai basis yang besar (lebih dari 40) akan memberikan hasil INF (Infinity), pada proses pembentukan nilai hash. Untuk mengatasi masalah tersebut, sistem diberi batasan untuk menggunakan nilai basis 2-30 (**Kharisman, et al., 2013**).

2.1.10 Dokumen Fingerprinting

Dalam penentuan nilai similarity dua buah dokumen, diburuhkan sebuah teknik pengumpulan nilai hash yang bernilai sama pada dua buah dokumen yang dibandingkan.

Menurut (**Saul Schleimer, et al., 2003**), untuk efisiensi, hanya sebuah subset dari hasil hash yang harus dipertahankan sebagai dokumen fingerprint. Berikut ini contoh fingerprint dengan nilai k-grams 5 :

1. Teks Sampel :
A do run run run, a do run run
2. Teks yang sudah diproses :
Adorunrunrunadorunrun
3. Proses teks dengan k-gram 5 :
adoru dorun orunr runru unrun
nrunr runru unrun nruna runad
unado nador adoru dorun orunr
runru unrun
4. Teks setelah di hash :
77 72 42 17 98 50 17 98 8 88 67 39
77 72 42 17 98
5. Hasil Fingerprint
72 88 72

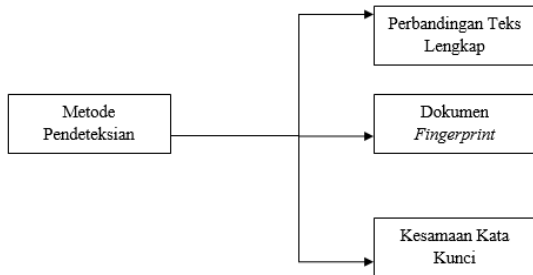
2.2 Plagiarisme

Plagiarisme adalah penjiplakan atau pengambilan karangan pendapat orang lain dan menjadikannya seolah-olah karangan sendiri (**Sugono, et al., 1997 dalam Salmuasih dan Sunyoto, 2013**).

2.2.1 Metode Pendeteksi Plagiarisme

Metode pendeteksi plagiarisme dibagi menjadi tiga bagian yaitu metode

perbandingan teks lengkap, metode dokumen fingerprinting, dan metode kesamaan kunci. Metode pendeteksi plagiarisme dapat dilihat pada gambar 2.3 (Stein, et al., 2006 dalam Surahman, 2013) :



Gambar 2.3 Metode Pendeteksi Plagiarisme

2.3 Similarity

Konsep similarity sudah menjadi isu yang sangat penting di hampir setiap bidang ilmu pengetahuan. Tiga macam teknik yang dibangun untuk menentukan nilai similarity (kemiripan) dokumen (Zaka, 2009 dalam Salmuasih dan Sunyoto, 2013) :

a. Distance-based similarity measure

Distance-based similarity measure mengukur tingkat kesamaan dua buah objek dari segi jarak geometris dari variabel-variabel yang tercakup di dalam kedua objek tersebut. Metode *Distance-based similarity* ini meliputi *Minkowski Distance*, *Manhattan/City block distance*, *Euclidean distance*, *Jaccard Distance*, *Dice's Coefficient*, *Cosine similarity*, *Levenshtein Distance*, *Hamming Distance* dan *Soundex distance*.

b. Feature-based similarity measure

Feature-based similarity measure melakukan penghitungan tingkat kemiripan dengan merepresentasikan objek ke dalam bentuk *feature-feature* yang ingin diperbandingkan. *Feature-based similarity measure* banyak digunakan dalam melakukan

pengklasifikasian atau *pattern matching* untuk gambar dan teks.

c. Probabilistic-based similarity measure

Probabilistic-based similarity measure menghitung tingkat kemiripan dua objek dengan merepresentasikan dua set objek yang dibandingkan dalam bentuk *probability*. *Kullback Leibler Distance* dan *Posterior Probability* termasuk dalam metode ini.

2.3.1 Pengukuran Nilai Similarity

Mengukur *similarity* (kemiripan) dan jarak antara dua entitas informasi adalah syarat inti pada semua kasus penemuan informasi, seperti pada *Information Retrieval* dan *Data Mining* yang kemudian dikembangkan dalam bentuk aplikasi, salah satunya adalah sistem deteksi plagiat. Adapun pengaplikasian *Dice's Similarity Coefficient* dalam perhitungan nilai *similarity* yang menggunakan pendekatan k-gram adalah sebagai berikut (Kosinov, 2001 dalam Salmuasih dan Sunyoto, 2013) :

$$S = \frac{2 * C}{(A + B)} \times 100$$

Dimana S adalah nilai similarity, A dan B adalah jumlah dari kumpulan k-grams dalam teks 1 dan teks 2. C adalah jumlah dari k-grams yang sama dari teks yang dibandingkan.

Untuk menentukan jenis plagiarisme antara dokumen yang diuji ada 5 jenis penilaian persentase (Mutiarra dan Agustina, 2008) :

1. 0% : Hasil uji 0% berarti kedua dokumen tersebut benar-benar berbeda baik dari segi isi dan kalimat secara keseluruhan;
2. < 15%: Hasil uji 15% berarti kedua dokumen tersebut hanya mempunyai sedikit kesamaan;

3. 15-50%: Hasil uji 15-50% berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang;
4. >50%: Hasil uji lebih dari 50% berarti dapat dikatakan bahwa dokumen tersebut mendekati plagiarisme;
5. 100%: Hasil uji 100% menandakan bahwa dokumen tersebut adalah plagiat karena dari awal sampai akhir mempunyai isi yg sama persis.

METODE PENELITIAN

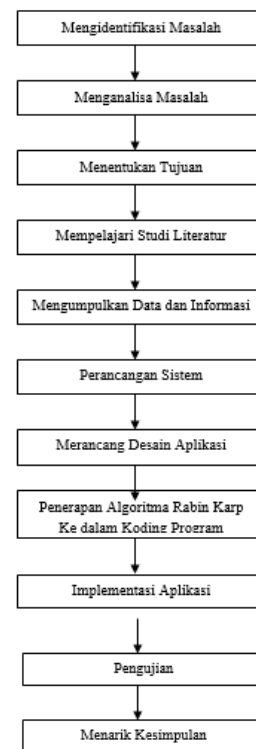
3.1 Pendahuluan

Penelitian ini adalah salah satu jenis penelitian yang hasilnya diharapkan dapat langsung diterapkan untuk memecahkan permasalahan yang dihadapi. Penelitian ini juga bertujuan untuk merancang dan membangun sebuah sistem berbasis web sebagai media untuk menyimpan dan menginformasikan hasil skripsi atau karangan ilmiah yang dilengkapi dengan fasilitas rekomendasi dokumen teks dengan memanfaatkan algoritma *Rabin-Karp*.

Penelitian merupakan rangkaian kegiatan ilmiah dalam rangka pemecahan suatu permasalahan. Fungsi penelitian adalah mencari penjelasan dan jawaban terhadap permasalahan serta memberikan alternatif bagi kemungkinan yang dapat digunakan untuk pemecahan masalah.

3.1.1 Kerangka Kerja

Kerangka kerja diperlukan dalam acuan langkah-langkah untuk mengerjakan suatu penelitian secara terstruktur dengan membuat sebuah tahapan metode penelitian sehingga hasil yang dicapai menjadi lebih maksimal.



Gambar 3.1 Kerangka Kerja Penelitian

ANALISA DAN HASIL

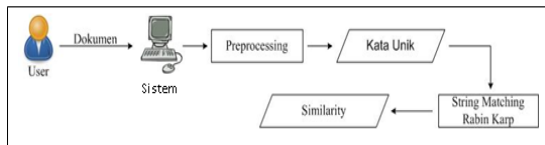
4.1 Analisa Sistem

Pada analisa sisten ini, pengujian dua dokumen dengan hasil persentase 0% berarti kedua dokumen benar-benar berbeda baik dari segi isi maupun kalimat keseluruhan, <15% berarti kedua dokumen tersebut hanya mempunyai sedikit kesamaan, 15-50% menandakan bahwa kedua dokumen tersebut termasuk plagiat tingkat sedang, >50% berarti kedua dokumen tersebut mendekati plagiarisme dan 100% menandakan bahwa kedua dokumen mempunyai isi yang sama. Namun, adakalanya nilai persentase yang dihasilkan lebih dari 100%.

4.2 Perancangan Sistem

Cara kerja sistem pendeteksi plagiarisme ini pertama kali user akan melihat daftar dokumen asli dan dokumen

uji pada tampilan awal aplikasi. Melihat apakah pada daftar tersebut sudah terdapat file dokumen yang hendak diuji. Bila belum ada, user akan menambahkan file dokumen baik itu dokumen asli maupun dokumen yang akan diuji. Setelah itu, sistem akan melakukan proses analisa kemiripan (similarity) dan waktu prosesnya dengan menggunakan algoritma Rabin-Karp yang telah diterapkan di dalam aplikasi.



Gambar 4.1 Skema Aliran Data

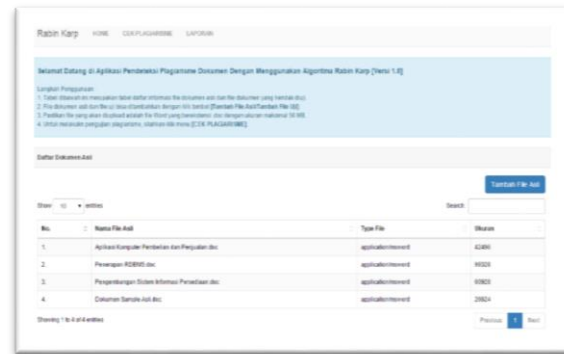
4.3 Perancangan Proses

Perancangan aplikasi yang dibuat adalah berupa sistem untuk mendeteksi plagiarisme suatu dokumen. Inputan pada aplikasi ini berupa file berekstensi .doc. Sedangkan pada PHP, file teks yang bisa dibaca hanyalah file dengan ekstensi .txt. Sehingga untuk mengatasi masalah tersebut dibutuhkan sebuah fungsi PHP untuk mengekstrak file tersebut kedalam bentuk .txt. Selanjutnya aplikasi akan memproses kedua dokumen tersebut guna mengevaluasi berapakah similarity antara dokumen tersebut.

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

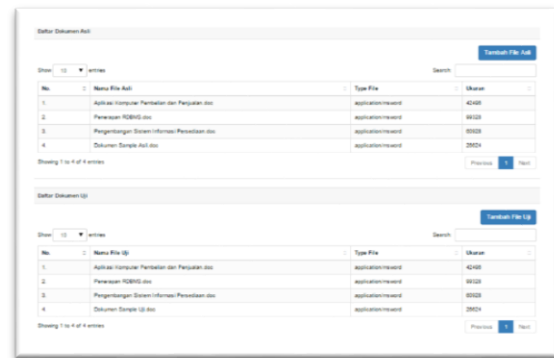
Sistem bisa diakses langsung oleh user dengan menggunakan browser dengan jenis dan versi apapun.



Gambar 5.1 Perancangan Antarmuka 5.2 Pengujian Sistem

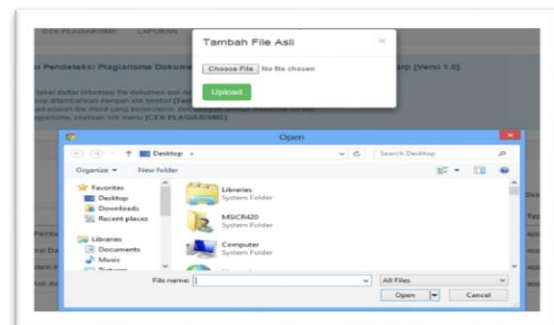
Adapun tahap-tahap penggunaan aplikasi yang sudah dirancang adalah sebagai berikut :

1. Menginputkan file dokumen yang akan diuji



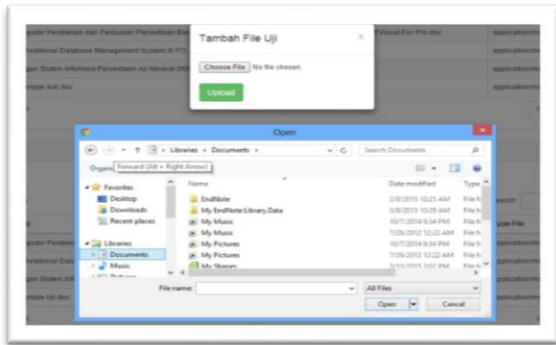
Gambar 5.2 Daftar File Pada Database

2. Input File Asli

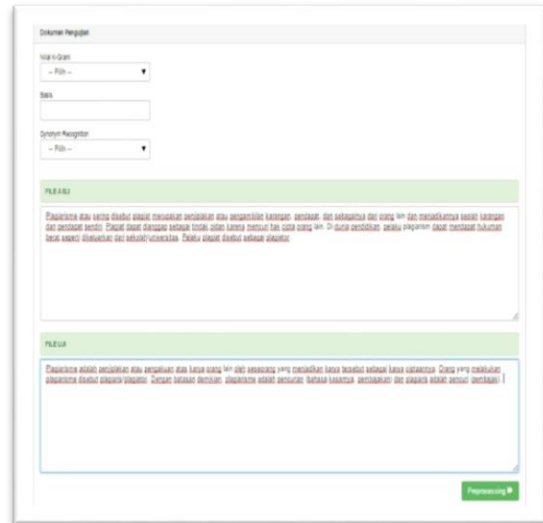


Gambar 5.3 Form Input File Asli

3. Input File Uji



Gambar 5.4 Form Input File Uji



Gambar 5.7 File Asli dan File Uji Setelah Dipilih

4. Source Code Ekstarksi File Doc

```

class DocxConversion{
    private $filename;

    public function __construct($filePath) {
        $this->filename = $filePath;
    }

    private function read_docx() {
        $filename = fopen($this->filename, "r");
        $line = fread($filename, filesize($this->filename));
        $lines = explode(chr(0x0D), $line);
        $outtext = "";
        foreach($lines as $thisline)
        {
            $spos = strpos($thisline, chr(0x0B));
            if (($spos !== FALSE) || (strlen($thisline) == 0))
            {
                $outtext .= $thisline . " ";
            }
        }
        $outtext = preg_replace("/[^\p{L}\p{N}]/", "", $outtext);
        return $outtext;
    }
}
    
```

Gambar 5.5 Source Code Ekstraksi File Doc

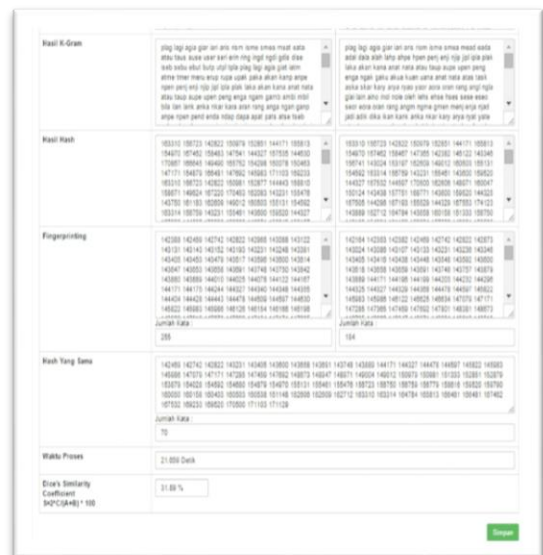
5. Pengecekan Plagiarisme



Gambar 5.6 Pilih File Asli dan File Uji

6. File Asli dan File Uji Setelah di Pilih

7. Hasil Similarity dengan Algoritma Rabin-Karp



Gambar 5.8 Hasil Penghitungan Similarity dengan Algoritma Rabin-Karp

8. Laporan Hasil Pengujian

No	Nama File (A)	Tipe File (A)	Ukuran File (A)	Nama File (B)	Tipe File (B)	Ukuran File (B)	K-gram	Basis	Jst Hash (A)	Jst Hash (B)	Hash Sama (C)	Simsim	Yield	Proses (%)	Similarity (%)
1.	Dokumen Sampel A0.doc	application/msword	28Kb	Dokumen Sampel B0.doc	application/msword	28Kb	1	11	21	20	20	Total	18,273	97,50	
2.	Dokumen Sampel A1.doc	application/msword	28Kb	Dokumen Sampel B1.doc	application/msword	28Kb	1	15	21	20	20	Total	17,880	97,50	
3.	Dokumen Sampel A2.doc	application/msword	28Kb	Dokumen Sampel B2.doc	application/msword	28Kb	1	17	21	20	20	Total	17,920	97,50	
4.	Dokumen Sampel A3.doc	application/msword	28Kb	Dokumen Sampel B3.doc	application/msword	28Kb	1	11	21	20	20	Ya	18,030	97,50	
5.	Dokumen Sampel A4.doc	application/msword	28Kb	Dokumen Sampel B4.doc	application/msword	28Kb	1	15	21	20	20	Ya	18,141	97,50	
6.	Dokumen Sampel A5.doc	application/msword	28Kb	Dokumen Sampel B5.doc	application/msword	28Kb	1	17	21	20	20	Ya	18,714	97,50	
7.	Dokumen Sampel A6.doc	application/msword	28Kb	Dokumen Sampel B6.doc	application/msword	28Kb	2	11	105	30	72	Total	15,09	73,85	

Gambar 5.9 Hasil Pengujian dengan Sistem

akurat bila dibandingkan dengan algoritma Rabin-Karp tanpa Synonym Recognition. Karena walaupun kata pada dokumen uji diganti sedemikian rupa namun memiliki makna yang sama akan terdeteksi oleh sistem ini.

4. Hasil dari Synonym Recognition sangat bergantung pada banyaknya sinonim yang diinputkan ke dalam kamus sinonim pada database.
5. Nilai K-gram yang lebih kecil akan menghasilkan akurasi nilai similarity yang lebih baik dibandingkan dengan nilai K-gram yang lebih besar.
6. Untuk nilai basis, tidak semua angka bisa digunakan. Karena di beberapa kasus nilai basis yang salah akan mengakibatkan nilai hash yang dihasilkan akan sama dengan nilai hash lain yang memiliki kata berbeda.
7. Dokumen uji dinyatakan terduplikasi apabila nilai persentase Similarity nya diatas 50%.

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil analisa dan penerapan yang sudah dilakukan pada bab sebelumnya, maka dapat disimpulkan bahwa :

1. Telah dibuat suatu sistem berbasis web yang dapat digunakan untuk pendeteksian plagiarisme terhadap suatu dokumen teks dengan menggunakan algoritma Rabin-Karp.
2. Tindakan plagiarisme dapat dilakukan dengan mengubah beberapa bagian bahkan mungkin secara keseluruhan dengan cara mengubah kata-kata dengan sinonim daripada kata-kata tersebut.
3. Algoritma Rabin-Karp yang menggunakan metode Synonym Recognition memiliki akurasi penentuan nilai persentase Similarity yang lebih

6.2 Saran

Adapun saran yang diberikan oleh penulis untuk pengembangan sistem pendeteksi plagiat ini adalah :

1. Aplikasi ini dapat dikembangkan dengan pendeteksi kalimat aktif dan pasif, karena tindak penjiplakan juga sering kali mengubah kalimat aktif menjadi pasif atau sebaliknya.
2. Penelitian selanjutnya disarankan untuk mencoba menggabungkan metode-metode lain agar nilai hash yang dihasilkan bisa lebih kecil dan unik agar terhindar dari bentrokan nilai hasing itu sendiri.

DAFTAR PUSTAKA

Atmopawiro, A., 2006. Pengkajian dan Analisis Tiga Algoritma Efisien

- Rabin-Karp, Knuth Morris-Pratt, dan Boyer-Moore dalam Pencarian Pola dalam Suatu Teks, Bandung: STEI ITB.
- Firdaus, H. B. 2008. Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp.
- Gipp, B. and Meuschke, N. 2011. Citation pattern matching algorithms for citation-based plagiarism detection: greedy citation tiling, citation chunking and longest common citation sequence. Proceedings of the 11th ACM Symposium on Document Engineering (DocEng2011), pp. 249--258.
- Hamza, Sahrial, et al. 2013. Sistem Koreksi Soal Essay Otomatis Dengan Menggunakan Metode Rabin Karp. Jurnal EECCIS. Vol. 7. No. 2
- Kharisman, Obed, et al., 2013. Implementasi Algoritma Winnowing Untuk Mendeteksi Kemiripan Pada Dokumen Teks. Informatika. Vol. 9. No. 1.
- Kosinov, S. 2001. Evaluation of n-grams conflation approach in text-based information retrieval. Canada: Computing Science Department. 136-142.
- Langgeni, D.P., et al. 2010. Clustering Artikel Berita Berbahasa Indonesia Menggunakan Unsupervised Feature Selection. Yogyakarta: Seminar Nasional Informatika. (22 Mei 2010)
- Mutiara, A.B., Agustina, Sinta. 2008. Aplikasi Anti Plagiatisme Dengan Algoritma Karp-Rabin Pada Penulisan Ilmiah Universitas Gunadarma. Depok: Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Gunadarma.
- Noprisson, H. et al. 2013. Implementasi Algoritma Rabin-Karp Untuk Menentukan Keterkaitan Antar Publikasi Penelitian Dosen Tahun 2013. Bengkulu: Jurnal Teknologi Informasi. Vol. 9, No. 2.
- Nugroho, Eko. 2011. Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin Karp. Malang: Program Studi Ilmu Komputer Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Univesitas Brawijaya.
- Pratama, M.R., et al. 2013. Aplikasi Pendeteksi Duplikasi Dokumen Teks Bahasa Indonesia Menggunakan Algoritma Winnowing Dengan Metode K-Gram Dan Synonym Recognition. Malang: Jurusan Teknik Informatika Universitas Muhammadiyah Malang.
- Putra, I.M.B.D., 2013. Case Folding Pada Konten Web Dengan PHP. Bandung: Jurusan Teknologi Informasi Fakultas Teknik Universitas Udayana. Merpati Vol. 1, No. 3.
- Salmuasih dan Sunyoto. 2013. Implementasi Algoritma Rabin Karp untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep Similarity. Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi. (15 Juni 2013)
- Schleimer, Saul., Daniel S. Wilkerson, and Aiken Alex, "Winnowing: Local Algorithms for Document Fingerprinting," in SIGMOD, San Diego, CA, 2003, p. 2003.
- Stein, B., and Meyer, S. zu Eissen. 2006. Near Similarity Search and Plagiarism Analysis. Germany.

DOI: <https://doi.org/10.47233/jteksis.v1i2.52>

Februari, 8, 2013.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.825&rep=rep&type=pdf>

Sugono, D and dkk. 1997. Kamus Besar Bahasa Indonesia (KBBI), <http://pusatbahasa.kemdiknas.go.id/kbbi>, 1997.

Surahman, A.M., 2013. Perancangan Sistem Penentuan Similarity Kode Program Pada Bahasa C dan Pascal Dengan Menggunakan Algoritma Rabin-Karp. Tanjungpura: Program Studi Teknik Informatika Fakultas Teknik Universitas Tanjungpura.

Thomas H. Cormen, et al. 2009. Introduction to Algorithms. 3rd edition. MIT Press.

Zaka, B. 2009. Theory and Applications of Similarity Detection Techniques. Graz University of Technology.