Aplikasi Pendeteksi Plagiarisme Dokumen Karya Ilmiah Menggunakan Algoritma Rabin Karp

Wisnu Prabowo

Sistem Informasi – Universitas Nasional

pwisnu.93@gmail.com

Abstrak -- Penggunaan internet dalam pencarian referensi karya ilmah sudah menjadi hal umum di lingkungan akademisi. Dalam pembuatan karya ilmiah sangat mungkin mengambil kata atau kalimat sebagai kutipan pada jurnal ataupun buku. Banyaknya kertersediaan dokumen karya ilmiah yang terdapat di internet menyebabkan pencarian dokumen referensi menjadi lebih praktis, namun hal tersebut juga meningkatkan potensi terjadinya plagiat. Dengan demikian teknik text mining dapat menjadi pendukung untuk melakukan pendeteksian dokumen karya ilmiah dengan tools untuk keakuratan hasil. Algoritma rabin karp cocok untuk membantu text mining dalam pendeteksian dokumen.

Kata kunci :Plagiat, text mining, karya ilmiah, rabin karp, internet

Abstract *- Internet use in search of reference of scientific work has become common in the academic environment. In making scientific work it is possible to take a word or phrase as a quote in a journal or book. The large number of scientific paperwork available on the internet causes the searching of reference documents to be more practical, but it also increases the potential for plagiarism. Thus text mining techniques can be a support for the detection of scientific paper documents with tools for the accuracy of the results. Karp's rabbin algorithm is suitable for text mining in document detection.*

Keywords: Plagiarism, text mining, scientific papers, rabin karp, internet

I. PENDAHULUAN

Dalam dunia pendidikan sangat rawan terjadinya plagiarisme dalam pembuatan karya ilmiah. Plagiarisme merupakan praktik menyalin ide kreatif karya oranglain [1]. karya ilmiah sebenarnya bentuk dari hak kekayaan intelektual manusia yang perlu dilindungi. Upaya untuk meminimalisir dan mengurangi praktik plagiarisme dalam membuat karya ilmiah, dapat di lakukan pendeteksian awal dengan menggunakan sistem untuk mempermudah pencegahan terjadinya plagiarisme. Teknik text mining dapat dipergunakan sebagai alat bantu untuk melakukan analisis adanya unsur plagiat dalam sebuah karya ilmiah. Dalam hal ini text mining dapat di kombinasikan dengan *algoritma rabin karp* karena algoritma ini menunjang untuk melakuakan pengecekan *mulitiple pattern matching* [2], dengan memanfaatkan fungsi hashing untuk memberi nilai sebagai landasan dasar mencari kesamaan pola dalam dokumen karya ilmiah [3]. Pada [4] di tunjukkan bahwa algoritma rabin karp pembobotan karakter dengan fungsi hash lebih unggul dari pada algoritma Jaro-winkler distance yang melakukan pendeteksian similarity dengan mengukur jarak dua kata.

Membaca dokumen karya ilmiah merupakan hal dasar untuk mendeteksi plagiat. Dalam upaya mengurangi kegiatan plagiat setiap karya ilmiah harus dilakukan pengecekan secara teliti menganalisis isi dari dokumen. Seorang verifikator karya ilmiah ketika melakukan pedeteksian harus membaca secara berulang untuk mencari kesamaan isi dokumen. Untuk menghemat waktu dan ke akuratan informasi di perlukan alat pendeteksi karya ilmiah agar bisa melakukan penecekan secara otomatis hanya dengan input beberapa parameter pendukung. Pada [4, 5, 6] saat pendeteksi dokumen hanya mencakup input manual text dan dokumen txt dalam pengujiannya sebelumnya data *string* yang digunakan file uji.txt dengan jumlah kata 58 [1]. Pada dasarnya karya ilmiah yang telah di publikasi berbentuk dokumen pdf dan jumlah *string* berkisar 2000 sampai 4000 kata dengan demikian penulis mencoba untuk mengembangkan penelitian dengan melakukan pengujian menggunakan dokumen karya ilmiah.

Penelitian ini mencakup pengujian menggunakan dokumen karya ilmiah dengan format doc dan pdf berbahasa indonesia. Aplikasi ini di bangun dengan platform berbasis web agar dapat di akses secara global dan tidak harus melakukan instalasi software. Sebagai pendukung keakuratan dan kecepatan informasi pendeteksian dan menghitung similiarity penulis memilih algoritma rabin karp untuk membantu analisis, karena algoritma ini sangant efektif untuk melakukan pencarian multi pattern [2]. Dengan kombinasi basis dan modulo untuk menentukan nilai hash dari kedua dokumen, diharapkan bisa meningkatkan akurasi kemiripan [7].

Tujuan penelitian ini untuk melakukan pengujian bahwa *algoritma rabin karp* dapat melakukan analisis data string multi pattern. Selain itu penelitian ini dapat di manfaatkan untuk mengetahui plagiarisme dengan cepat dan efisien dalam pendeteksian dokumen karya ilmiah mengguanakan format doc, pdf berbasis web.

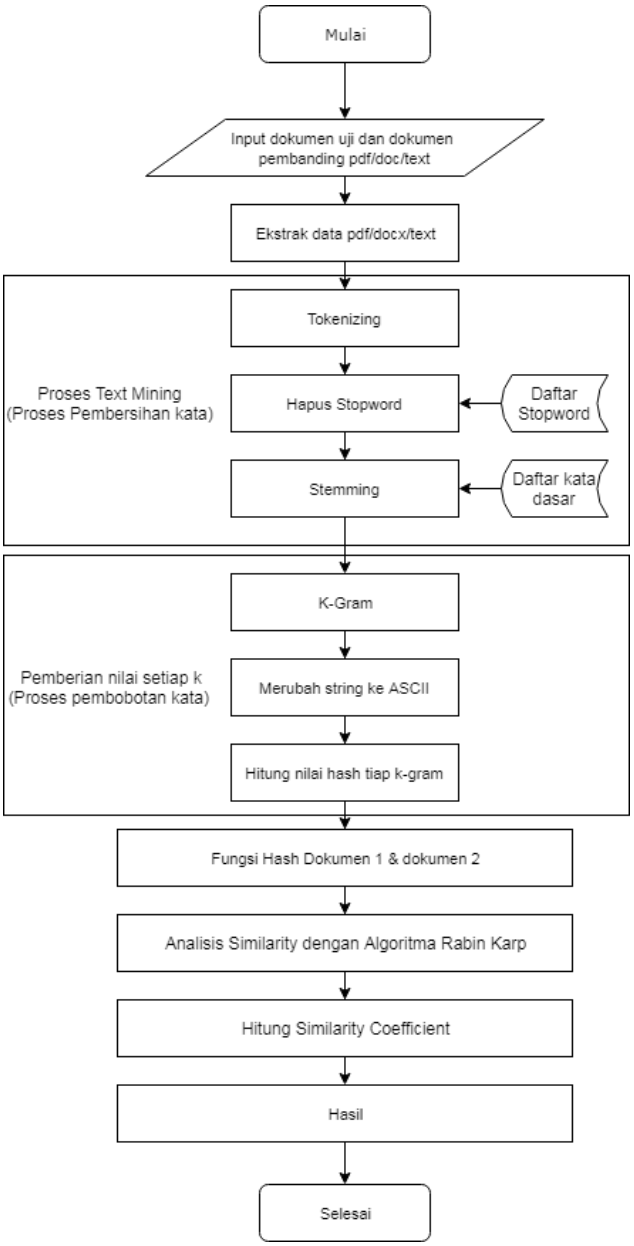
II. METODE PENELITIAN

Dalam penelitian ini penulis melakukan pendeteksian dokumen karya ilmiah menggunakan *algoritma rabin karp*, dengan data pengujian dokumen karya ilmiah dengan format pdf. Mentranformasikan dokumen pdf kedalam bentuk text dilakukan untuk memudahkan

pembersihan *string* yang tidak diperlukan sebelum dilakukan analisis dan menghitung similarity.

Secara umum Text Mining memiliki beberapa tahapan proses untuk mendukung tercapainya hasil analisis informasi yang maksimal. Ada dua tahapan dalam analisis text mining pertama tahap preprosesing pembersihan data mencakup case folding, tokenizing, stopword removal, Stemming [6].

Pendeteksian dokumen karya ilmiah teknik text mining di tambahkan Algoritma Rabin Karp untuk melengkapi tahapan analisis. Algoritma Rabin Karp memiliki beberapa tahapan untuk melakukan analisis setelah tahapan text mining selesai dilakukan antara lain K-Gram, Hashing dan menghitung Similarity.



Gambar 1. Flowchart Umum Sistem

Input dokumen uji dan dokumen pembanding, sebagai langkah awal pendeteksian kesamaan dokumen, syarat minimal harus ada dua dokumen untuk dilakukan proses pendeteksian. Format dokumen yang di gunakan adalah pdf. Dokumen yang di lakukan pendeteksian harus setara atau dengan tema yang sama.

Proses ekstraks dokumen dari pdf menjadi text memanfaatkan library pdfparser kemudian di simpan kedalam database, ekstraksi dokumen berfungsi untuk mengeluarkan data agar dapat di lakukan pembersihan karakter yang tidak berpengaruh terhadap hasil analisis.

Tokenizing proses pemotongan string masukan berdasarkan tiap kata menyusunnya [8]. Di dalam proses ini terdapat proses eliminasi tanda baca karena di anggap tidak penting dan tidak berpengaruh dalam analisis seperti titik, koma, titik dua, tanda petik. Selain itu didalam tahapan ini ada proses case folding merubah huruf kapital menjadi huruf kecil karena nilai angka huruf kapital dan huruf kecil berbeda maka dilakukan case folding, agar nantinya akan di lakukan analisis mengeluarkan nilai yang setara [9].

Proses Filtering, proses ini menghilangkan kata - kata tidak penting atau disebut juga stoplist proses ini termasuk dalam pembersihan data. Dalam tahapan ini data yang telah selesai proses tokenizing di hilangkan kata kata tidak penting *(remove stopword)* seperti kata dan, di, ke, dari, yang, dengan, dalam proses ini juga akan di sediakan dataset kata-kata tidak penting bisa di lakukan manual setting kata apa saja yang akan di hilangkan sesuai user inginkan.

Stamming, Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama menggunakan data stopword yang sudah di siapkan menggunakan library sastrawi.

K-Gram proses membagi panjang string menjadi bagian bagian sejumlah (k) dalam bentuk array berfungsi untuk membangkitkan kata atau karakter, dalam hal ini untuk nilai K-Gram antara satu dengan yang lainnya harus sama. Seperti kata belajar menggunakan k=4 menjadi berikut bela, elaj, laja, ajar. Semakin kecil nilai (k) akan semakin akurat dalam analisis selanjutnya K-Gram akan membentuk nilai hash untuk memberikan ID pada setiap potongan kata [2].

Mempercepat pengujian kesetaraan pola ke substring dalam teks dengan menggunakan fungsi hash. Fungsi hash adalah fungsi yang mengubah setiap string menjadi nilai numerik [10]. Hashing adalah cara untuk merubah potongan string yang telah di bagi sesuai panjang k-gram kedalam nilai hash. Untuk mendapat kan nilai hash setiap karakter harus di rubah mnggunakan ASCII (American Standard Code for Information Interchange) karena dalam proses tokenizing terdapat proses merubah huruf kapital menjadi kecil maka nilai decimal ASCII dari 97-122. Contoh nilai a = 97, b = 98, c= 99 dan seterusnya. Berikut persamaan untuk mendapatkan nilai hash atau di sebut juga roling hash.

$$H= C1_1 * b^{k-1} + C_2 * b^{k-2} * \ldots + C_{k-1} * b + C_k \quad (1)$$

Keterangan:

C : nilai ascii karakter

b : basis (bilangan prima)

k : banyak karakter

Analisis dengan Algoritma Rabin-Karp untuk pencocokan string yang menggunakan fungsi hash sebagai pembanding antara string yang dicari dengan substring pada teks. Apabila hash value keduanya sama maka akan dilakukan perbandingan sekali lagi terhadap karakter karakternya. Apabila hasil keduanya tidak sama, maka substring akan bergeser ke kanan [10]. Untuk pola data teks yang panjang algoritma Rabin karp menggunakan operasi *mod* nilainya akan menjadi lebih kecil dan tidak merubah hasil akhir fungsi *mod* untuk mempercepat algoritma dalam memproses data teks yang besar. Contoh menggunakan nilai ASCII untuk menghitung nilai "aku" dengan *mod* 13.

- aku = 97+107+117=321
- 321 *mod* 13 = 9

Untuk menghitung nilai *similarity coefficient* di gunakan untuk melakukan perhitungan kemiripan dengan rumus.

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

Keterangan:

S : Nilai Similarity

A : Jumlah Hash dari text 1

B : Jumlah Hash dari text 2

C : Jumlah Hash yang sama dari text 1 dan 2

III. HASIL DAN PEMBAHASAN

Dalam tahapan ini penulis akan menguraikan tahap text mining untuk melakukan Analisis pendeteksian dua dokumen karya ilmuah. Dari pemrosesan awal penyiapan dokumen sampai menentukan nilai similarity.

Sebelum dilakukan pendeteksian, peneliti menyiapkan dokumen pdf atau doc yang akan di gunakan. Agar data dapat di olah dokumen pdf atau doc di lakukan proses parsing atau ekstrak data ke bentuk teks.

Table 1 Hasil proses parsing dokumen

Id	Judul	Kata	Size (kb)
1	Text_pdf_1	2594	309
2	Text_pdf_2	3670	752

Setelah proses parsing selesai proses selanjutnya tahapan tokenizing, pembersihan tanda baca dan kata atau kalimat tidak perlu *(removal stopword)* yang tidak mempegaruhi tahapan analisis.

Tabel 2 Parsing, tokenizing, stopword, stemming

Id	Judul	Stopword	Kata
1	Text_pdf_1	Di, ke, dari, dengan, pendahuluan, daftar pustaka, metode penelitian	2489
2	Text_pdf_2	Di, ke, dari, dengan, pendahuluan, daftar pustaka, metode penelitian	3461

Tabel 2 merupakan hasil dari tokenizing dan stopword, stemming dalam proses ini tokenizing menghapus karakater selain huruf dan angka yang dilakukan langsung oleh sistem. Untuk stopword dalam sistem ini user harus input kata atau kalimat yang tidak masuk dalam konten untuk di analisis seperti kata pendahuluan, metode penelitian termasuk ke dalam daftar stopword. Penulis mengambil potongan kalimat dari dua dokumen untuk perbandingan. Kalimat pertama **"plagiarisme adalah penjiplakan atau pengambilan karangan, pendapat orang lain dan menjadikannya seolah-olah karangan sendiri."** Kalimat dua **"Plagiat adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah-olah karangan dan pendapat sendiri."**

Tabel 3 k-gram dan nilai hash kalimat pertama

Kalimat pertama	
K-Gram	Hash
[plagi][lagia][agiar][giari]	[5262][2846][57]
[iaris][arism][risme][ismea]	[9719][300][4884]
[smead][meada][eadal]	[2770][3577][894]
[adala][dalah][alahj][lahji]	[2016][9996][5207]
[ahjip][hjipl][jipla][iplak]	[6326][5813][2986]
[plaka][lakat][akata][katau]	[1595][6626][5348]
[ataua][tauam][auamb]	[9436][5298][3261]
[uambi][ambil][mbilk]	[4605][9724][6588]
[bilka][ilkar][lkara][karan]	[1508][7832][5181]
[arang][rangp][angpe]	[7278][9096][7809]
[ngpen][gpend][penda]	[3998][5522][9475]
[endap][ndapa][dapat]	[3855][1469][9284]
[apato][pator][atora][toran]	[1144][9464][7484]
[orang][rangl][angla][nglai]	[7289][5451][6822]
[glain][laind][ainda][indan]	[1267][3024][8249]
[ndanj][danja][anjad][njadi]	[9773][8689][1465]
[jadio][adiol][diola][iolah]	[9236][611][3611]
[olahk][lahka][ahkar][hkara]	[3146][3340][5809]
[karan][arang][rangs][angse]	[5438][6660][9481]
[ngsen][gsend][sendi][endir]	[3306][3236][6219]
[ndiri]	[7444][8102][641]
	[2989][1630][7000]
	[9475][3855][1472]
	[9317][1507][3450]
	[1380][7379][6449]

Tabel di atas merupakan tabel kgram dan nilai hash kalimat pertama. Dalam pengujian ini penulis menggunakan kgram = 5 dengan basis = 11 dan mod 10007.

Contoh menghitung nilai hash frase ‘plagi’ pada kedua kalimat.

$$\text{Hash}=[112 * 11^4] + [108 * 11^3] + [97 * 11^2] + [103 * 11^1] + [105 * 11\text{^}0] \bmod 10007$$

$$\text{Hash}= 1639792 + 143748 + 11737 + 1133 + 105$$

$$\text{Hash}= 1796515 \bmod 10007$$

$$\text{Hash}= 5262$$

Tabel 4 Hasil K-gram dan nilai hash kalimat kedua

Kalimat kedua	
K-Gram	Hash
[plagi][lagia][agiat][giata]	[5262][2846][59]
[iatad][atada][tadal][adala]	[9733][439][6401]
[dalah][alahj][lahji][ahjip]	[9457][5207][6326]
[hjipl][jipla][iplak][plaka]	[5813][2986][1595]
[lakat][akata][katau][ataua]	[6626][5348][9436]
[tauam][auamb][uambi]	[5298][3261][4605]
[ambil][mbilk][bilka][ilkar]	[9724][6588][1508]
[lkara][karan][arang][rangp]	[7832][5181][7278]
[angpe][ngpen][gpend]	[9096][7809][3998]
[penda][endap][ndapa]	[5522][9475][3855]
[dapat][apatd][patda][atdan]	[1469][9284][1144]
[tdans][danse][anseb][nseba]	[9464][7484][7289]
[sebag][ebaga][bagai][agaio]	[5451][6822][1256]
[gaior][aiora][ioran][orang][	[2886][6744][3230]
rangl][angla][nglai][glain]	[6763][605][5732]
[laind][ainda][indan][ndanj]	[9900][1012][6461]
[danja][anjad][njadi][jadio]	[9181][15][3615]
[adiol][diola][iolah][olahk]	[8834][8689][1465]
[lahka][ahkar][hkara][karan]	[9236][611][3611]
[arang][rangd][angda]	[3146][3340][5809]
[ngdan][gdanp][danpe]	[5438][6660][9481]
[anpen][npend][penda]	[3306][3236][6219]
[endap][ndapa][dapat][apats]	[7444][8102][641]
[patse][atsen][tsend][sendi]	[2989][1630][7000]
[endir][ndiri]	[9475][3855][1457]
	[9148][9655][3027]
	[6730][254][1874]
	[7484][7289][5451]
	[6822][1271][3055]
	[8603][3650][1380]
	[7379][6449]

Jumlah nilai hash yang sama dari kedua potongan kaliamat sebesar 66 kemudian jumlah kgram kalimat pertama 75 dan kalimat ke dua 92 jadi nilai similaritynya.

$$S = \frac{2*66}{75+92} x\ 100$$

S = 79.04 %

Pengujian similarity di atas dengan mengambil sampel kalimat dari masing – masing dokumen yang telah dilakukan proses preprosesing. Dalam merubah dan menghitung nilai hash kedua sampel kalimat di atas menggunakan fungsi hash sebagai berikut kode programnya.

```
function rollingHash($string) {
 $basis = 11;
 $pjgString = strlen($string);
 $hash = 0;
 for ($i = 0; $i < $pjgString; $i++) {
   $ascii = ord($string[$i]);
   $hash+=$ascii*pow($basis,$pjgString -
   ($i + 1));
 }
 return $hash % 10007;}
```

Tabel 5 Hasil pengujian similarity dua dokumen

id	kgram	Modulo	Similarity (%)
1	10	10007	61.42
2	20	10007	18.90
3	30	10007	20.40
4	40	10007	20.40
5	50	10007	25.51

Dari tabel di atas penulis melakukan pengujian similarity keseluruhan isi dokumen dengan beberapa kgram yang berbeda. Hasil pengujian menunjukkan bahwa semakin kecil kgram, untuk data string yang besar hasil similarity semakin besar, karena setiap menemukan pola yang sama pada isi dokumen akan menyimpan secara berulang dan menyebabkan jumlah pola yang sama bertambah. Misalkan kata **“plagiarism”** ketika dalam satu kalimat terdapat 2 kata “plagiarism” maka akan di anggap 4 yang sama, menjadikan jumlah pola similarity akan bertambah.

Tabel 6 Hasil Pengujian kecepatan analsis dengan modulo

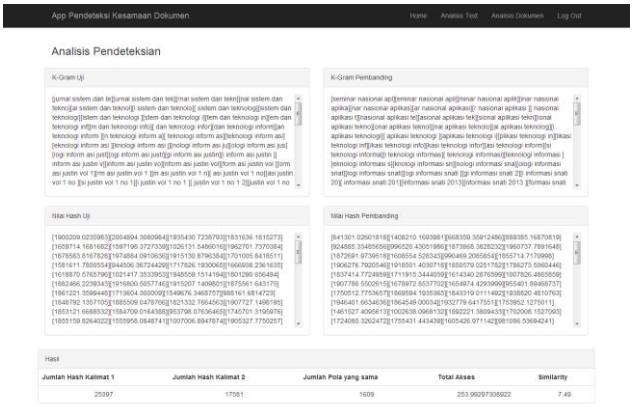
Id	Kgram	Modulo	Basis	Similarity	Waktu
1	20	100007	3	17.66	255
2	20	10007	3	17.97	84
3	20	1007	3	103.69	20

Dalam table di atas nilai basis kecil di gunakan untuk memperkecil nilai hash yang di hasilkan dan nilai modulo semakin kecil waktu yang di butuhkan semankin cepat dalam penggunaan data yang besar nilai modulo dan basis mempengaruhi hasil.

Tabel 7 Hasil pengujian dengan remainder terhadap nilai similarity dan waktu

Id	Kgram	Remainder	Basis	Similarity	Waktu
1	20	100007	3	7.49	253
2	20	10007	3	7.49	263
3	20	1007	3	7.49	261

Nilai remainder merupakan nilai yang digunakan untuk membagi nilai hash guna mendapatkan nilai hash yang unik. Hasil pengujian dari tabel di atas menunjukkan nilai similarity konsisten dengan menggunakan reminder dibandingkan menggunakan modulo. Penggunaan reminder tidak mempengaruhi kecepatan waktu analisis.



Gambar 1 Testing analisis plagiat dokumen

Hasil pengujian analisis plagiat dokumen pada gambar 1 menunjukkan penguraian kalimat menggunakan kgram dan hasil nilai hash pada setiap kgram dengan hasil similarity 7.49% dari keseluruhan isi dokumen.

IV. KESIMPULAN DAN SARAN

Dilakukan proses ekstrak dokumen pdf ke dalam bentuk text dapat memudahkan proses tokenizing, remove stopword, stemming dan fungsi algoritma rabin karp dapat berjalan dengan baik. Banyaknya data dalam jurnal mempengaruhi waktu proses. Penggunaan kgram untuk jumlah data teks yang besar lebih baik menggunakan kgram lebih dari 20 kareakter dengan basis kecil untuk menunjang validnya hasil similarity. Dengan demikian algoritma rabin karp akan efektif di gunakan dengan mengkombinasikan nilai basis dan remainder untuk mendapatkan hasil yang maksimal.

Untuk penelitian selanjutnya agar proses parsing di perhatikan lebih detail karena ada beberapa isi dokumen, hasil parsing terlihat tidak beraturan dan tidak bisa di stemming dengan maksimal kemudian di tambah proses stopword yang lebih detail untuk mengurangi waktu proses dan di tambah proses dalam dokumen pembanding untuk menghilangkan beberapa frase atau kata yang berulang.

DAFTAR PUSTAKA

[1] Salmuasih, Andi S. 2013. Implementasi Algoritma Rabin Karp untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep Similarity. ISSN: 1907 – 5022.

[2] Singla. Nimisha, Deepak Garg. 2012. String Matching Algoritms and their Applicability in various Aplications. IJSCE Vol. 1 Issue. 6. ISSN: 2231-2307.

[3] Saini. Anu, Ankita Bahl, MitaliSingh, Supria Kumari. 2016. Plagiarism Checker: Text Mining. Vol. 134-No.3

[4] Jayanta, Halim M, Titin P. 2017. Analisis Pengukuran Self Plagiarism Menggunakan Algoritma Rabin-Karp Dan Jaro-Winkler Distance Dengan Stemming Tala. ISSN 2302-3805.

[5] Suparwanto. Agus, Riyadi J. Iskandar, Soebandi. Penerapan Algoritma Rabin Karp Untuk Mendeteksi Kemiripan Dua Dokumen Text. Jurnal InTekSis Vol. 4 No. 1.

[6] Purba Andry Hery, Zakarias Situmorang, 2017. Analisis Perbandingan Algoritma Rabin Karp Dan Levenshtein Distance Dalam menghitung Kemiripan Text. Vol. 2, No. 2, ISSN: 2548-1916.

[7] Siahaan, Andyansyah Putera Utama. 2017. Analisis K-Gram, Basis dan Modulo Rabin Karp Sebagai Penentu Akurasi Presentase Kemiripan Dokumen. SENASPRO.

[8] Ariyani,N.H., Sutardi Ramadan,R, 2016. Aplikasi pendeteksi kemiripan teks Dokumen Menggunakan Metode Levenshtein Distance, Program Studi Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari, Vol. 2, No. 1, pp 279-286, ISSN: 2502-8928.

[9] Putra Doddi Aria, Herry Sujaini, Helen Sasty P. 2015 Implementasi Algoritma Rabin-Karp untuk Membantu Pendeteksian Plagiat pada Karya Ilmiah. Vol. 1, No. 1,

[10] Astuti. Yuli, Sumarni Adi. Information Retrieval Mendeteksi Konten Anarkis Pada Web Keagamaan Menggunakan Algoritma Rabin Karp. Jurnal Ilmiah DASI Vol. 17 No. 4 hlm 56-62. ISSN: 1411 – 3201.

[11] Putra, D.A., Sijaini, H dan Pratiwi,H.S., 2015. Implementasi Algoritma Rabin Karp untuk membantu pendeteksian Plagiat pada Karya Ilmiah, Program Teknik Informatika, Fakultas Teknik, Universitas Tanjungpura, Vol. 1, No. 1.

[12] Talib. Ramzan, Muhammad Kashif Hanify, Shaeela Ayeshaz, and Fakeeha Fatimax. 2016. Text Mining: Techniques, Applications and Issues.(IJACSA) International Journal of Advanced Computer Science and Applications.Vol. 7 No. 11.

[13] Putri. Ranti Eka, Andysah Putera Utama Siahaan., 2017. Examination of Document Similarity Using Rabin-Karp Algorithm. International Journal of Recent Trends in Engineering & Research (IJRTER). Vol: 03, Issue: 08, ISSN: 2455-1457.

[14] A. P. U. Siahaan, Mesran, R. Rahim and D. Siregar, 2017 "K-Gram As A Determinant Of Plagiarism Level In Rabin-Karp Algorithm," International Journal of Scientific & Technology Research, vol.6, no. 7, pp. 350-353.