Analisis Kemiripan Dokumen Tesis Menggunakan Algoritma Rabin-Karp Dan Dice Coefficient Similarity

Document Similarity Analysis of Student Thesis using Rabin-Karp Algorithm and Dice Coefficient Similarity

Agus Santoso[1], Achmad Solichin[2]
[1,2]Universitas Budi Luhur, Jakarta
E-mail: [1]agussantoso552@gmail.com, [2]achmad.solichin@budiluhur.ac.id

Abstrak

Kemiripan dokumen (*document similarity*) merupakan salah satu topik penelitian yang populer. Pada konteks akademis, tingkat kemiripan dokumen sering digunakan sebagai ukuran indikasi plagiasi karya ilmiah. Selain itu, tingkat kemiripan dokumen juga bermanfaat bagi akademisi dalam menemukan publikasi ilmiah yang selaras dengan topik penelitian tertentu. Kontribusi utama dari penelitian ini adalah melakukan analisis kemiripan dokumen tesis mahasiswa pada suatu program studi. Hasil analisis kemiripan dapat menjadi ukuran tingkat indikasi plagiasi dokumen tesis di program studi. Untuk menghasilkan kemiripan dokumen teks digunakan algoritma *Rabin-Karp* dan metode *Dice Coefficient Similarity*. Sebagai data uji, digunakan kumpulan dokumen tesis dari 4 (empat) program studi di Universitas Budi Luhur. Hasil pengujian analisis kemiripan dokumen tesis untuk program studi S2 Ilmu Komputer sebesar 20,95%, S2 Ilmu Komunikasi sebesar 21,07%, S2 Akuntansi sebesar 26,63%, dan S2 Manajemen sebesar 27,9%. Selain itu, untuk mengukur akurasi metode yang diusulkan dilakukan perbandingan hasil kemiripan dokumen dengan perangkat lunak *CheckPlagiarism* dan menghasilkan tingkat akurasi sebesar 94,7%. Hasil tersebut menunjukkan bahwa metode yang diusulkan mampu menghasilkan tingkat similaritas dokumen dengan baik.

Kata kunci: Kemiripan, Dokumen, Tesis, Rabin-Karp, Dice Coefficient Similarity

Abstract

Document similarity is one of the popular research topics. In an academic context, the similarity of documents is often used as an indication of plagiarism in scientific work. In addition, the similarity of documents is also useful for academics in finding scientific publications that are in line with certain research topics. The main contribution of this research is to analyze the document similarity of student thesis in a study program. The results of the similarity analysis can be a measure of the level of indication of plagiarism in thesis documents in the study program. To produce the similarity of text documents, the Rabin-Karp algorithm and the Dice Coefficient Similarity method are used. As test data, a collection of thesis documents from four study programs at Universitas Budi Luhur was used. The test results of the analysis of the similarity of the thesis documents for the Master of Computer Science study program are 20.95%, Master of Communication Science 21.07%, Master of Accounting 26.63%, and Master of Management 27.9%. In addition, to measure the accuracy of the proposed method, a comparison of the results of the document similarity with the CheckPlagiarism software was carried out and resulted in an accuracy rate of 94.7%. These results indicate that the proposed method is able to produce a good level of document similarity.

Keywords: Similarity, Document, Thesis, Rabin-Karp, Dice Coefficient Similarity

1. PENDAHULUAN

Kemiripan dokumen *(document similarity)* merupakan salah satu topik penelitian yang populer. Pada konteks akademis, tingkat kemiripan dokumen sering digunakan sebagai ukuran indikasi plagiasi karya ilmiah. Berdasarkan Permendiknas, (Pencegahan dan Penanggulangan Plagiarisme di Perguruan Tinggi, Nomor 7, Pasal 1 ayat 1 2010) Plagiarisme adalah tindakan atau perilaku dengan sengaja atau tidak sengaja dalam mendapatkan kredit ataupun nilai pada karya ilmiah, baik mengutip sebagian maupun seluruh karya ilmiah orang lain, tanpa menyebutkan sumbernya dengan baik dan benar. [1]. Sedangkan pelaku plagiarisme disebut dengan Plagiator. Selain itu, tingkat kemiripan dokumen juga bermanfaat bagi akademisi dalam menemukan publikasi ilmiah yang selaras dengan topik penelitian tertentu. Khususnya pada Universitas Budi Luhur yang belum mempunyai sistem analisis kemiripan dokumen tesis. Berdasarkan latar belakang tersebut, diperlukan sistem yang dapat menganalisis tingkat kemiripan antar dokumen tesis. Metode yang diusulkan untuk menghasilkan kemiripan dokumen teks yaitu algoritma *Rabin-Karp* dan *Dice Coefficient Similarity*.

Beberapa penelitian sebelumnya yang berkaitan dengan algoritma untuk menentukan tingkat kemiripan dokumen, diantaranya penelitian "Perbandingan Algoritma *Boyer-Moore* dan *Brute Force* pada Pencarian Kamus Besar Bahasa Indonesia Berbasis Android"[2]. Algoritma *Brute Force* mempunyai keunggulan terhadap pencarian string satu pola (*single pattern*), tetapi masih ada kelemahan terhadap pencarian string beberapa pola (*multi pattern*). Algoritma *Boyer-Moore* secara umum baik dalam menggeser dua karakter terakhir, namun buruk dalam menggeser karakter sebelumnya [3]. Algoritma *Rabin-Karp* dapat mengatasi kendala algoritma *Brute Force* dalam pencarian string beberapa pola, ini tepat untuk mendeteksi kemiripan dokumen yang memerlukan beberapa perbandingan pola. Menurut Wicaksono untuk mengembangkan sistem pendeteksian kemiripan dokumen membutuhkan algoritma yang baik untuk bermacam jenis pola pencocokan string. Salah satu algoritma yang cocok untuk kendala pencocokan pola *multi string* adalah algoritma *Rabin-Karp*. Kelebihan dari algoritma *Rabin-Karp* dibandingkan dengan algoritma pencocokan string lainnya adalah dapat mencari pola dari beberapa string atau beberapa pola [4].

Leonardo dan Hansun [4] menggunakan algoritma *Rabin-Karp* dan *Jaro-Winkler Distance* untuk membandingkan antara dokumen uji dan dokumen pembanding, dan didapatkan algoritma *Rabin-Karp* lebih efektif daripada algoritma *Jaro-Winkler Distance*, algoritma *Rabin-Karp* mempunyai nilai rata-rata kemiripan 51% dan algoritma *Jaro-Winkler Distance* sebesar 35%. Dan dari segi waktu proses algoritma *Rabin-Karp* lebih cepat, algoritma *Rabin-Karp* mempunyai rata-rata 0,594 menit, sedangkan algoritma *Jaro-Winkler Distance* rata-ratanya mencapai 0,992 menit. Priambodo [5] menerapkan algoritma *Rabin-Karp* dengan *Rolling Hash* pada 30 dokumen teks dan mencapai akurasi tertinggi 47.58%. Purba dan Situmorang [6] menggunakan *Rabin-Karp* Dan *Levenshtein Distance*, dan didapatkan *Rabin-Karp* mendapatkan persentase kemiripan lebih tinggi dalam dokumen yang diuji dibandingkan dengan *Levenshtein Distance*. Algoritma *Rabin-Karp* mencapai persentase 79,9%, sedangkan *Levenshtein Distance* sebesar 66,7%. Filcha dan Hayati [7] menggunakan algoritma Rabin-Karp untuk pendeteksi plagiarisme pada dokumen tugas Mahasiswa, Hasil perhitungan akurasi dengan *confusion matrix* pada sistem mencapai 90% yang diperoleh dari 20 perbandingan dokumen tugas mahasiswa. Algoritma yang digunakan pada sistem tidak memiliki perbedaan persentase saat urutan perbandingan diubah. Pamungkas dan Fitrianingsih [8] menggunakan algoritma *Rabin Karp* untuk deteksi similaritas dokumen ilmiah. Penelitian ini berhasil membentuk aplikasi pendeteksi similaritas tulisan ilmiah menggunakan sumber hasil pencarian mesin pencari google.

Setelah penulis melakukan studi literatur, algoritma yang akan digunakan oleh penulis yaitu algoritma *Rabin-Karp* untuk menganalisis kemiripan antar dokumen dan metode *Dice Coefficient Similarity* untuk mengukur tingkat kemiripan.

Kontribusi utama dari penelitian ini adalah melakukan analisis kemiripan dokumen tesis mahasiswa pada suatu program studi. Hasil analisis kemiripan dapat menjadi ukuran tingkat indikasi plagiasi dokumen tesis di program studi. Sebagai data uji, digunakan kumpulan dokumen

tesis dari 4 (empat) program studi di Universitas Budi Luhur.

2. METODE PENELITIAN

2.1 Metode Penelitian

Metode penelitian menggunakan algoritma *Rabin-Karp* dan metode *Dice Coefficient Similarity*, algoritma *Rabin-Karp* untuk menganalisis kemiripan dokumen sedangkan metode *Dice Coefficient Similarity* untuk mengukur tingkat kemiripan.

a. Algoritma Rabin-Karp

Algoritma *Rabin-Karp* adalah algoritma pencarian *string* yang dikembangkan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987. Algoritma *Rabin-Karp* menggunakan fungsi hash untuk menemukan pola berdasarkan string teks. Algoritma *Rabin-Karp* memiliki karakteristik *K-Gram* dan langkah *hashing*. Sebelum mengimplementasikan algoritma *Rabin-Karp*, terlebih dahulu harus dilakukan langkah *preprocessing* teks [7]. Tahap *preprocessing* bertujuan untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut [9]. Dan berikut adalah *pseudocode* dari algoritma *Rabin-Karp*.

Tabel 1 *Pseudocode* algoritma *Rabin-Karp*

```
Fungsi
RabinKarp (input s: s[1..m], teks: string[1..n])  boolean
{Pencarian string s pada string teks dengan algoritma Rabin-Karp}
Deklarasi
i : integer
ditemukan = boolean
Algoritma
ditemukan  ← false
hs ← hash(s[1..m])
hsub ← hash(teks[1..i+m-1])
  for i  ←  1 to n do
              if hsub = hs then
                 if teks[i..i+m-1] = s then
                   ditemukan ← true
              else
                 hsub  hash(teks[i+1..i+m])
  endfor
return ditemukan
```

Berikut langkah-langkah yang dilalui pada algoritma *Rabin-Karp*.

1) K-Gram

K-Gram adalah rangkaian token yang panjangnya k. Metode *K-Gram* mengambil potongan karakter huruf dengan jumlah nilai k dari sebuah teks yang dibaca secara berurutan dari awal teks sampai akhir teks sumber. Contoh K-Gram dengan nilai k = 5 dapat dilihat pada Tabel 2.

Tabel 2 Contoh *K-Gram*

Kalimat	:	mobil barang mewah
Text Preprocessing	:	mobilbarangmewah
K-Gram{3}		mob, obi, bil, ilb, lba, bar, ara, ran, ang, ngm, gme, mew, ewa, wah

2) Hashing

Hashing merupakan proses untuk mentransformasi string menjadi suatu nilai yang unik (nilai *hash*) dengan panjang tertentu yang berfungsi sebagai

penanda string tersebut [10]. Proses konversi ke *hash* menggunakan *rolling hash*. Nilai hash sering digambarkan dengan *fingerprint*, yang merupakan string pendek dari huruf dan angka yang muncul secara acak (data biner ditulis dalam heksadesimal). Nilai *hash* tersebut merupakan masukkan algoritma *Rabin-Karp*. Dan berikut adalah persamaan *rolling hash* [7].

$$H\ (c1 \ldots ck) = c1 * b^{(k-1)} + c2 * b^{(k-2)} + .. + ck * b^{(k-k)} \quad (1)$$

Keterangan :
c : nilai ASCII per karakter
b : bilangan prima konstanta
k : jumlah karakter

Tabel 3 Tabel ASCII

Karakter	Nilai ANSI ASCII (Desimal)	Karakter	Nilai ANSI ASCII (Desimal)
a	97	n	110
b	98	o	111
c	99	p	112
d	100	q	113
e	101	r	114
f	102	s	115
g	103	t	116
h	104	u	117
i	105	v	118
j	106	w	119
k	107	x	120
l	108	y	121
m	109	z	122

Tabel 4 Contoh *Rolling Hash*

Contoh Karakter	*Rolling Hash*
mob	Diketahui: m = 109, o = 111, b = 98 H = c1*b^(k-1)+c2*b^(k-2)+..+ck*b^(k-k) H = 109*7^(3-1)+111*7^(3-2)+98*7^(3-3) H = 5341 + 777 + 98 H = 6216

b. Dice Coefficient Similarity

Dice Coefficient Similarity adalah metode pengukuran untuk melakukan perhitungan nilai kesamaan terhadap pendekatan *K-Gram*. Persamaan *Dice Coefficient Similarity* dapat dilihat pada Persamaan 2 [7].

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

Diketahui persamaan 2, S adalah nilai kemiripan atau *similarity*, A dan B adalah jumlah dari *fingerprint hash* teks 1 dan *fingerprint hash* teks 2. C adalah jumlah dari *fingerprint hash* A dan B. *Fingerprint hash* adalah *hash* yang unik dan tidak terduplikasi (*non duplicated*) [7].

2.2 Data Penelitian

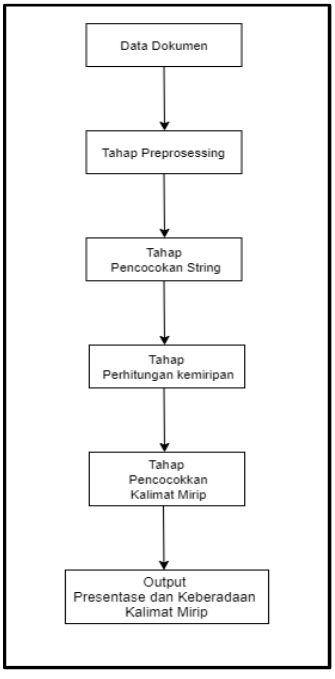
Data yang digunakan sebagai sampel penelitian adalah dokumen abstrak Bahasa Indonesia dari tesis mahasiswa S2 Universitas Budi Luhur dengan ekstensi pdf. File – file terdiri dari 100 *dataset* tesis Tahun 2018/2019 dan Tahun 2019/2020.

Tabel 5 Dokumen Sampel *Dataset*

No.	Program Studi	Jumlah *Dataset*
1.	Magister Manajemen (MM)	40
2.	Magister Ilmu Komputer (MKOM)	39
3.	Magister Akuntansi (MAKSI)	11
4.	Magister Ilmu Komunikasi (MIKOM)	10

3. HASIL DAN PEMBAHASAN

3.1 Analisis Kemiripan Dokumen



Gambar 1 Tahapan Sistem Analisis Kemiripan Dokumen

Tahapan proses sistem analisis kemiripan dokumen teks sebagai berikut:

a. Data Dokumen

Pada tahap ini dilakukan proses pengumpulan data dokumen abstrak dari tesis mahasiswa Universitas Budi Luhur. Dimana data dokumen ini ada yang dijadikan sebagai dataset dan ada juga yang menjadi data tes. Dataset ini berasal dari dokumen abstrak tesis mahasiswa yang sudah di publish di perpustakaan Universitas Budi Luhur. Sedangkan data tes berasal dari dokumen abstrak mahasiswa yang akan menjalankan tesis, sehingga mahasiswa bisa melakukan pengecekan *similarity* di dokumen abstraknya.

b. Tahap *Preprocessing*

Setelah dataset dan data tes dimasukkan, selanjutnya melakukan tahap pra proses (*text preprocessing*). *Text preprocessing* merupakan tahap awal memproses isi dokumen atau teks. Tahap ini terdiri dari proses *Case Folding, Tokenizing, Punctuation Removal, Stopword Removal, Space Removal*:

1) *Case Folding*
Proses ini melakukan pengubahan masukan teks menjadi huruf kecil.

2) *Tokenizing*
Proses ini melakukan pemisahan kata pada kalimat berdasarkan susunan kata. Hasil dari pemisahan kata disebut juga dengan token.

3) *Punctuation Removal*
Proses ini melakukan penghapusan karakter – karakter unik/tanda baca seperti karakter tanda titik, tanda seru, tanda tanya, tanda koma dan lain-lain.

4) *Stopword Removal*
Proses ini melakukan penghapusan kata yang tidak relevan dalam teks atau kata-kata yang tidak deskriptif. Contoh *stopword* adalah “yang”, “dan”, “di”, “dari” dan sebagainya.

5) *Space Removal*
Proses ini melakukan penghapusan spasi pada tiap kata.

c. Tahap Pencocokkan String

Setelah tahap *preprocessing* selesai, selanjutnya melakukan tahap pencocokan string yaitu parsing *K-Gram*, *Hashing* dan pencocokkan string dengan algoritma *Rabin-Karp.*

d. Tahap Perhitungan Kemiripan

Tahapan ini melakukan perhitungan hasil kemiripan kata pada data tes dengan dataset. Nilai kemiripan dihitung berdasarkan pada banyaknya jumlah *hash* yang telah diproses dengan Algoritma *Rabin-Karp* dan jumlah *hash* pada data tes. Hasil perhitungan kemiripan ini berasal dari hasil data tes yang menghasilkan nilai persentase kemiripan dokumen. Tahapan perhitungan kemiripan menggunakan metode *Dice Coefficient Similarity*.

e. Tahap Pencocokan Kalimat Mirip

Pada tahapan ini dilakukan pengambilan data dari dataset dan data tes. Dengan memotong paragraf menjadi kalimat-kalimat. Kemudian potongan kalimat tersebut dipotong kembali menjadi kata-kata. Kemudian, kata pada data tes ini akan dicocokkan dengan kata pada dokumen dataset. Kemudian tahapan selanjutnya akan menampilkan kata yang sama antara kata pada data tes dan dataset.

f. Output Persentase dan keberadaan kalimat mirip

Hasil output sistem kemiripan kata dihasilkan dari tahap perhitungan kemiripan dan tahap pencocokkan kata yang sama antara kata pada data tes dan dataset. Tahap perhitungan kemiripan menghasilkan nilai persentase data tes dan waktu proses, sedangkan tahap pencocokan kata yang sama menghasilkan keberadaan kata sama antara kata pada data tes dan dataset. Setiap kata yang sama akan ditandai dengan warna berbeda.

Berikut ini contoh tahapan proses sistem analisis kemiripan dokumen teks:

a. Dokumen Teks

Tabel 6 Dokumen Teks

Dokumen Pertama	Dokumen Kedua
Dalam dunia teknologi informasi, perkembangan teknologi informasi selalu mengalami kemajuan dan pembaharuan.	Dalam dunia teknologi informasi, teknologi informasi selalu berkembang dan mengalami pembaharuan.

b. Tahap *Preprocessing*

Tabel 7 *Text Preprocessing*

Dokumen	*Case Folding*	*Tokenizing*	*Punctuation Removal*	*Stopword Removal*	*Space Removal*
Dokumen Pertama	dalam dunia teknologi informasi, perkembangan teknologi informasi selalu mengalami kemajuan dan pembaharuan.	dalam dunia teknologi informasi , perkembangan teknologi informasi selalu mengalami kemajuan dan pembaharuan .	dalam dunia teknologi informasi perkembangan teknologi informasi selalu mengalami kemajuan dan pembaharuan	dunia teknologi informasi perkembangan teknologi informasi mengalami kemajuan pembaharuan	duniateknologi informasiperkembangantekno logiinformasi mengalamike majuanpembah aruan
Dokumen Kedua	dalam dunia teknologi informasi, teknologi informasi selalu berkembang dan mengalami pembaharuan.	dalam dunia teknologi informasi , teknologi informasi selalu berkembang dan mengalami pembaharuan .	dalam dunia teknologi informasi teknologi informasi selalu berkembang dan mengalami pembaharuan	dunia teknologi informasi teknologi informasi berkembang mengalami pembaharuan	duniateknologi informasitekno logiinformasib erkembangme ngalamipemba haruan

Tabel 8 Hasil *Text Preprocessing*

Teks Pertama	duniateknologiinformasiperkembanganteknologiinformasimengalamikemajuanpembaharuan
Teks Kedua	duniateknologiinformasiteknologiinformasiberkembangmengalamipembaharuan

c. K-Gram

Tabel 9 *K-Gram*

	K-Gram (3)
Teks Pertama	{dun}, {uni}, {nia}, {iat}, {ate}, {tek}, {ekn}, {kno}, {nol}, {olo}, {log}, {ogi}, {gii}, {iin}, {inf}, {nfo}, {for}, {orm}, {rma}, {mas}, {asi}, {sip}, {ipe}, {per}, {erk}, {rke}, {kem}, {emb}, {mba}, {ban}, {ang}, {nga}, {gan}, {ant}, {nte}, {tek}, {ekn}, {kno}, {nol}, {olo}, {log}, {ogi}, {gii}, {iin}, {inf}, {nfo}, {for}, {orm}, {rma}, {mas}, {asi}, {sis}, {ise}, {sel}, {ela}, {lal}, {alu}, {lum}, {ume}, {men}, {eng}, {nga}, {gal}, {ala}, {lam}, {ami}, {mik}, {ike}, {kem}, {ema}, {maj}, {aju}, {jua}, {uan}, {anp}, {npe}, {pem}, {emb}, {mba}, {bah}, {aha}, {har}, {aru}, {rua}, {uan},
Teks Kedua	{dun}, {uni}, {nia}, {iat}, {ate}, {tek}, {ekn}, {kno}, {nol}, {olo}, {log}, {ogi}, {gii}, {iin}, {inf}, {nfo}, {for}, {orm}, {rma}, {mas}, {asi}, {sit}, {ite}, {tek}, {ekn}, {kno}, {nol}, {olo}, {log}, {ogi}, {gii}, {iin}, {inf}, {nfo}, {for}, {orm}, {rma}, {mas}, {asi}, {sis}, {ise}, {sel}, {ela}, {lal}, {alu}, {lub}, {ube}, {ber}, {erk}, {rke}, {kem}, {emb}, {mba}, {ban}, {ang}, {ngd}, {gda}, {dan}, {anm}, {nme}, {men}, {eng}, {nga}, {gal}, {ala}, {lam}, {ami}, {mip}, {ipe}, {pem}, {emb}, {mba}, {bah}, {aha}, {har}, {aru}, {rua}, {uan},

d. Hashing

Tabel 10 *K-Gram*

	Hash
Teks Pertama	{1361}, {1488}, {1402}, {1352}, {1322}, {1454}, {1340}, {1404}, {1431}, {1434}, {1408}, {1413}, {1347}, {1370}, {1377}, {1407}, {1365}, {1450}, {1450}, {1387}, {1323}, {1462}, {1382}, {1425}, {1358}, {1448}, {1375}, {1334}, {1372}, {1283}, {1306}, {1396}, {1328}, {1319}, {1439}, {1454}, {1340}, {1404}, {1431}, {1434}, {1408}, {1413}, {1347}, {1370}, {1377}, {1407}, {1365}, {1450}, {1450}, {1387}, {1323}, {1465}, {1391}, {1446}, {1330}, {1371}, {1314}, {1432}, {1481}, {1394}, {1342}, {1396}, {1326}, {1294}, {1372}, {1305}, {1403}, {1367}, {1375}, {1333}, {1378}, {1308}, {1402}, {1454}, {1315}, {1427}, {1420}, {1334}, {1372}, {1277}, {1282}, {1341}, {1332}, {1474}, {1454},
Teks Kedua	{1361}, {1488}, {1402}, {1352}, {1322}, {1454}, {1340}, {1404}, {1431}, {1434}, {1408}, {1413}, {1347}, {1370}, {1377}, {1407}, {1365}, {1450}, {1450}, {1387}, {1323}, {1466}, {1394}, {1454}, {1340}, {1404}, {1431}, {1434}, {1408}, {1413}, {1347}, {1370}, {1377}, {1407}, {1365}, {1450}, {1450}, {1387}, {1323}, {1465}, {1391}, {1446}, {1330}, {1371}, {1314}, {1421}, {1448}, {1299}, {1358}, {1448}, {1375}, {1334}, {1372}, {1283}, {1306}, {1399}, {1324}, {1301}, {1312}, {1418}, {1394}, {1342}, {1396}, {1326}, {1294}, {1372}, {1305}, {1408}, {1382}, {1420}, {1334}, {1372}, {1277}, {1282}, {1341}, {1332}, {1474}, {1454},

e. Dice Coefficient Similarity

Tabel 11 *Fingerprint*

	Fingerprint
Teks Pertama	{1361}, {1488}, {1402}, {1352}, {1322}, {1454}, {1340}, {1404}, {1431}, {1434}, {1408}, {1413}, {1347}, {1370}, {1377}, {1407}, {1365}, {1450}, {1450}, {1387}, {1323}, {1462}, {1382}, {1425}, {1358}, {1448}, {1375}, {1334}, {1372}, {1283}, {1306}, {1396}, {1328}, {1319}, {1439}, {1465}, {1391}, {1446}, {1330}, {1371}, {1314}, {1432}, {1481}, {1394}, {1342}, {1326}, {1294}, {1372}, {1305}, {1403}, {1367}, {1333}, {1378}, {1308}, {1402}, {1454}, {1315}, {1427}, {1420}, {1277}, {1282}, {1341}, {1332}, {1474}, (Keterangan : Jumlah 64)
Teks Kedua	{1361}, {1488}, {1402}, {1352}, {1322}, {1454}, {1340}, {1404}, {1431}, {1434}, {1408}, {1413}, {1347}, {1370}, {1377}, {1407}, {1365}, {1450}, {1450}, {1387}, {1323}, {1466}, {1394}, {1465}, {1391}, {1446}, {1330}, {1371}, {1314}, {1421}, {1448}, {1299}, {1358}, {1448}, {1375}, {1334}, {1372}, {1283}, {1306}, {1399}, {1324}, {1301}, {1312}, {1418}, {1394}, {1342}, {1396}, {1326}, {1294}, {1372}, {1305}, {1408}, {1382}, {1420}, {1277}, {1282}, {1341}, {1332}, {1474}, {1454}, (Keterangan : Jumlah 60)

Jumlah *Fingerprint* yang sama dari Teks Pertama dan Teks Kedua	{1361}, {1488}, {1402}, {1352}, {1322}, {1454}, {1340}, {1404}, {1431}, {1434}, {1408}, {1413}, {1347}, {1370}, {1377}, {1407}, {1365}, {1450}, {1450}, {1387}, {1323}, {1394}, {1342}, {1326}, {1294}, {1372}, {1305} {1465}, {1391}, {1446}, {1330}, {1371}, {1314}, {1448}, {1375}, {1334}, {1372}, {1283}, {1306}, {1396}, {1394}, {1402}, {1454}, {1420}, {1277}, {1282}, {1341}, {1332}, {1474}, (Keterangan : Jumlah 49)

Berikut perhitungan kemiripan (*similarity*) teks pertama dan kedua dengan persamaan:

S = ((2*C)/(A+B)) * 100

Diketahui pada Tabel 11 Fingerprint :
A = Jumlah Fingerprint Teks Pertama = 64
B = Jumlah Fingerprint Teks Kedua = 60
C = Jumlah fingerprint yang sama dari Teks Pertama dan Teks Kedua = 49

Sehingga,
S = ((2*49)/(64+60)) * 100
S = 79.03%

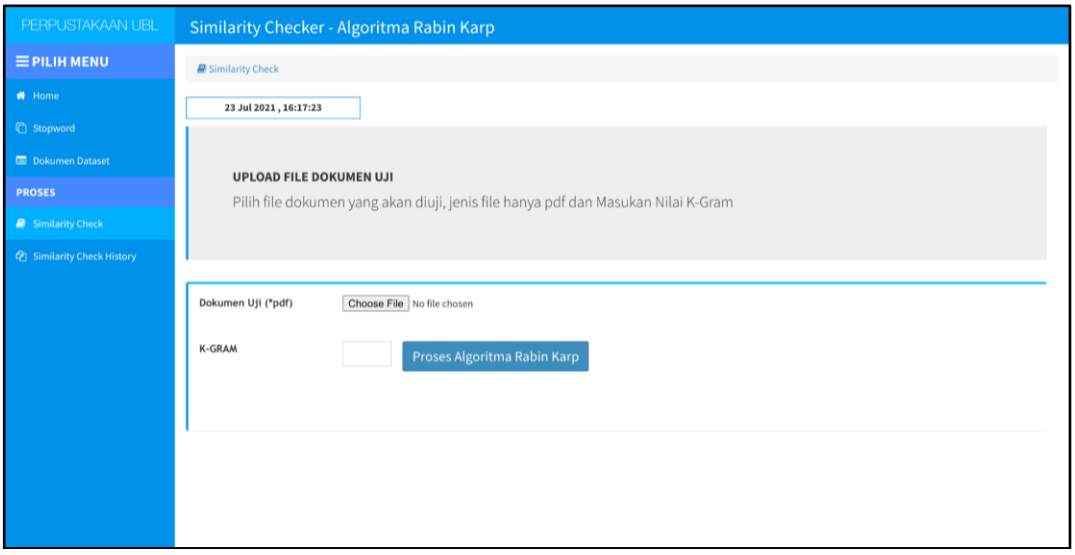
Berdasarkan perhitungan di atas, dapat diketahui persentase kemiripan antar dokumen pertama dan kedua adalah 79.03%.

3.2 System Prototype

Tahap ini menjelaskan tentang *prototype* dari sistem yang diimplementasikan.

a. *Similarity Check Prototype*

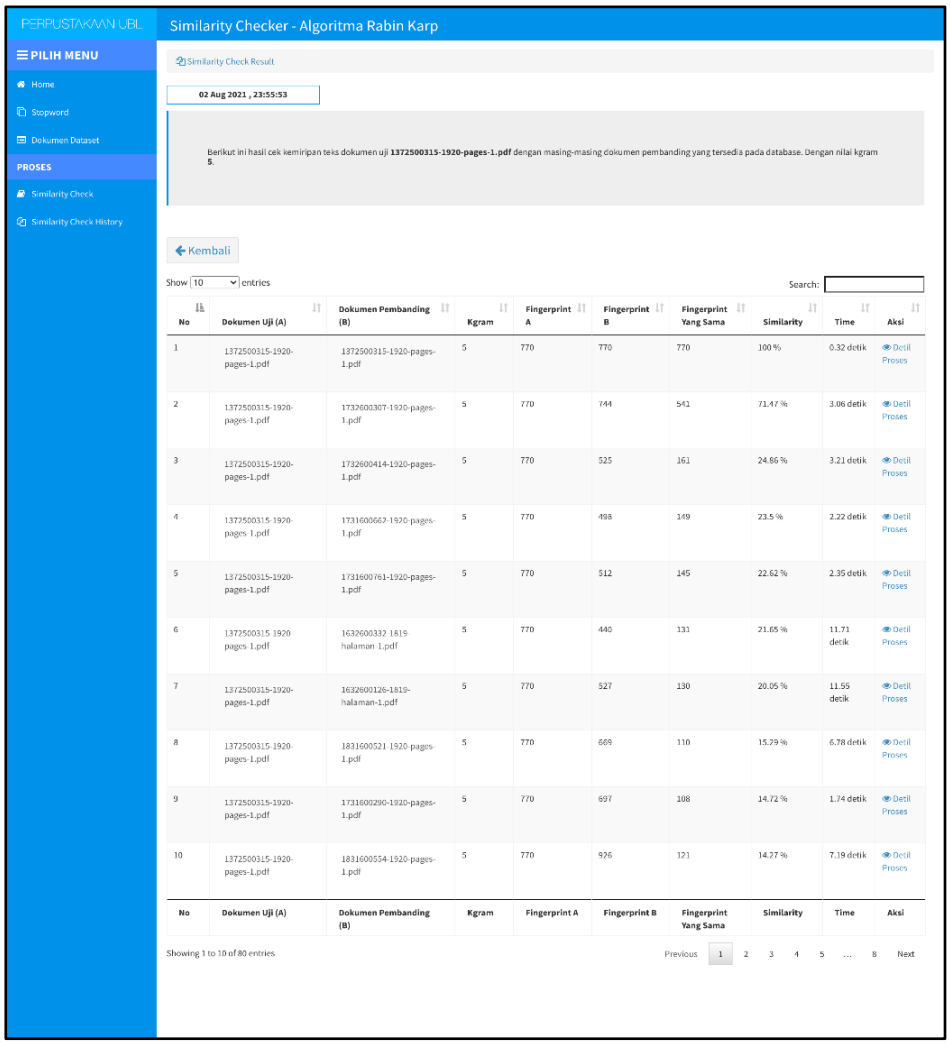
Pada Gambar 2 merupakan *Similarity Check Prototype* terdapat proses untuk mengunggah dokumen dan dokumen yang diunggah berupa dokumen abstrak dalam format pdf.



Gambar 2 *Similarity Check Prototype*

b. *Similarity Check Result Prototype*

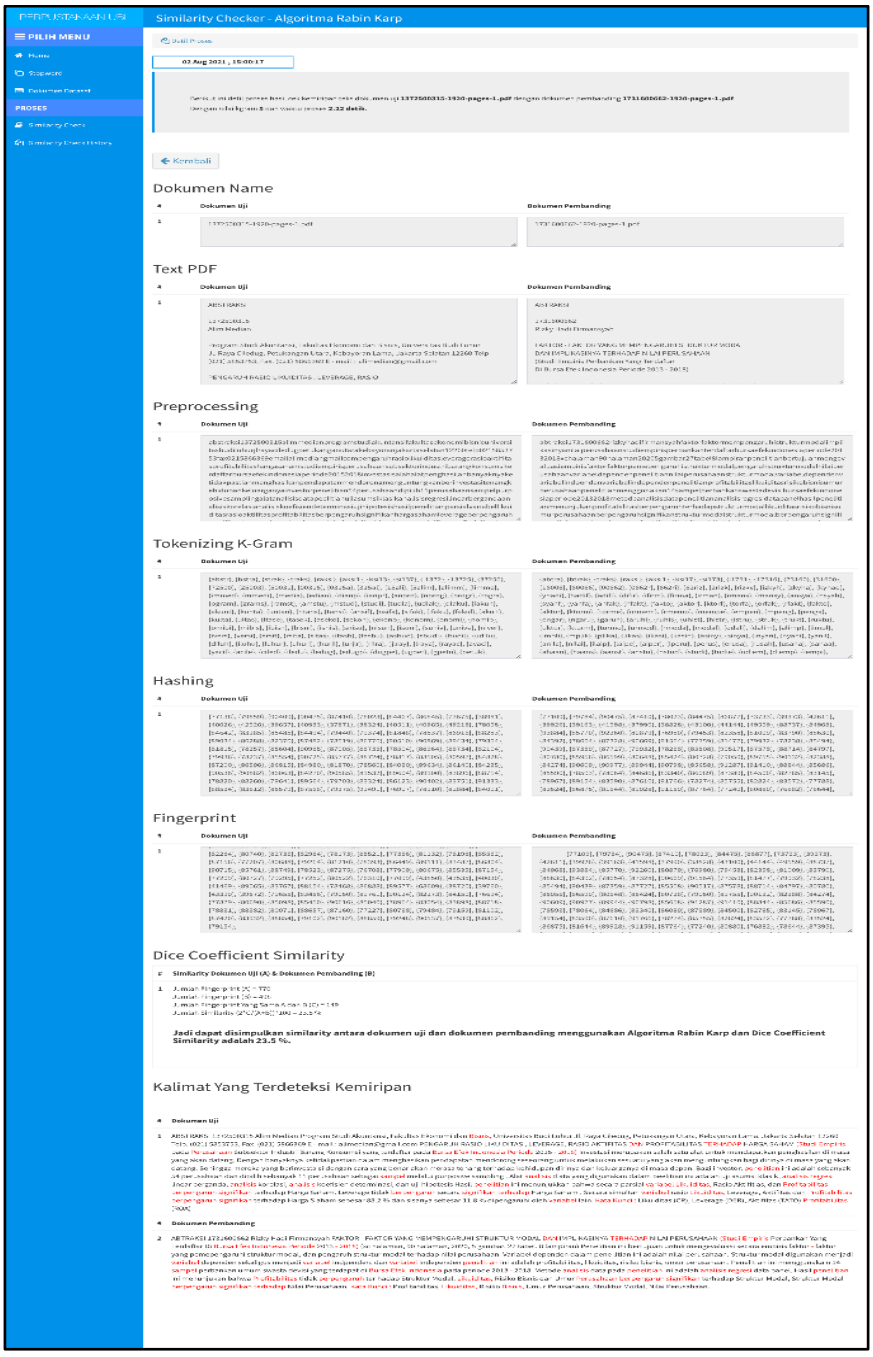
Pada Gambar 3 merupakan *Similarity Check Result Prototype*, pada tampilan ini berisi hasil perbandingan satu dokumen data tes abstrak yang sebelumnya diunggah dengan beberapa dokumen *dataset* sebagai pembanding yang tersedia pada sistem.



Gambar 3 *Similarity Check Result Prototype*

c. *Similarity Check Result Detail Prototype*

Pada Gambar 4 merupakan *Similarity Check Result Detail Prototype*, pada tampilan ini hasil perbandingan satu dokumen data tes abstrak yang sebelumnya diunggah dengan masing-masing dokumen *dataset* sebagai pembanding kemiripan dokumen. Hasil berupa nilai persentase kemiripan dan berisi kalimat yang terdeteksi kemiripannya.



Gambar 4 *Similarity Check Result Detail Prototype*

3.3 Pengujian

Pengujian adalah suatu keharusan dalam pengembangan sistem dengan tujuan untuk mengevaluasi, menganalisis, dan menemukan tingkat akurasi atau kesamaan hasil yang dicapai oleh sistem yang dirancang. Pengujian sistem peneliti dilakukan menggunakan beberapa cara diantaranya:

a. Pengujian mencari akurasi sistem peneliti dengan pembanding sistem *Check Plagiarism*

Pengujian menggunakan 20 dokumen abstrak program studi MKOM sebagai *dataset* yang semua data dokumennya berbeda-beda, dan saling dibandingkan masing-masing dari 20 dokumen tersebut. Pada pengujian ini dilakukan pada sistem peneliti dan sistem *check plagiarism*. Pada pengujian sistem peneliti ini digunakan nilai *k-gram* 5.

Pada pengujian ini bertujuan untuk mencari akurasi sistem peneliti dengan menghitung rata-rata selisih persentase similaritas dari sistem peneliti dengan sistem *check plagiarism*.

Pada Tabel 12 merupakan tabel hasil uji coba 20 *dataset* pada sistem *check plagiarism* dan berisi nilai persentase similaritas antar dokumen dalam satuan persen (%). Pada Tabel 13 merupakan tabel hasil uji coba 20 *dataset* pada sistem peneliti dan berisi nilai persentase similaritas antar dokumen dalam satuan persen (%). Pada Tabel 14 merupakan tabel berisi selisih nilai persentase similaritas masing-masing hasil uji coba 20 *dataset* pada sistem *check plagiarism* dan sistem peneliti dalam satuan persen (%). Kemudian dihitung rata-rata selisih dari total selisih dibagi total data perbandingan. Kemudian 100 dikurangi rata-rata selisih, 100 - 5,3 = 94,7 %. Dan dapat disimpulkan akurasi sistem *check plagiarism* dan sistem peneliti sebesar 94,7 %.

Tabel 12 Uji Coba 20 *Dataset* pada sistem *check plagiarism* (Dalam %)

No.	NIM	A	B	C	D	E	… T
1	141160xx95 (A)	100	1	1	2	1	….
2	151160xx93 (B)	1	100	2	2	4	….
3	161160xx48 (C)	1	2	100	3	3	….
4	161160xx63 (D)	2	2	3	100	2	….
5	161160xx11 (E)	1	4	3	2	100	….
….20	…. (T)	….	….	….	….	….	….

Tabel 13 Uji Coba 20 *Dataset* pada sistem peneliti (Dalam %)

No.	NIM	A	B	C	D	E	… T
1	141160xx95 (A)	100	5,88	3,59	8,2	6,79	….
2	151160xx93 (B)	5,88	100	7,47	8,89	12,96	….
3	161160xx48 (C)	3,59	7,47	100	9,81	7,48	….
4	161160xx63 (D)	8,2	8,89	9,81	100	10,35	….
5	161160xx11 (E)	6,79	12,96	7,48	10,35	100	….
….20	…. (T)	….	….	….	….	….	….

Tabel 14 Selisih uji coba 20 *Dataset* pada sistem *check plagiarism* dan sistem peneliti (Dalam %)

No.	NIM	A	B	C	D	E	… T
1	141160xx95 (A)	0	4,88	2,59	6,2	5,79	….
2	151160xx93 (B)	4,88	0	5,47	6,89	8,96	….
3	161160xx48 (C)	2,59	5,47	0	6,81	4,48	….
4	161160xx63 (D)	6,2	6,89	6,81	0	8,35	….

5	161160xx11 (E)	5,79	8,96	4,48	8,35	0	….
….20	…. (T)	….	….	….	….	….	….
	Total	2121,34	2121,34				
	Jumlah Data	400					
	Rata-Rata	5,30335	5,3				

b. Pengujian mencari rata-rata waktu proses sistem peneliti

Pengujian menggunakan 20 dokumen abstrak program studi MKOM sebagai *dataset* yang semua data dokumennya berbeda-beda, dan saling dibandingkan masing-masing dari 20 dokumen tersebut. Pada pengujian ini dilakukan pada sistem peneliti. Pada pengujian sistem peneliti ini digunakan nilai k-gram 5. Pada pengujian ini bertujuan untuk mencari rata-rata waktu proses sistem peneliti. Hasil pengujian dapat dilihat Pada Tabel 15. Pada Tabel 15 merupakan tabel berisi waktu proses pada sistem peneliti dalam satuan detik. Kemudian dihitung rata-rata dari total waktu proses dibagi total data perbandingan. Dan dapat disimpulkan rata-rata waktu proses per dokumen pada sistem peneliti sebesar 0,16 detik.

Tabel 15 Uji Coba 20 *Dataset* pada sistem peneliti (Dalam detik)

No.	NIM	A	B	C	D	E	… T
1	141160xx95 (A)	0,51	0,42	0,36	0,42	0,5	….
2	151160xx93 (B)	0,12	0,13	0,15	0,13	0,14	….
3	161160xx48 (C)	0,09	0,08	0,08	0,08	0,09	….
4	161160xx63 (D)	0,15	0,16	0,15	0,14	0,17	….
5	161160xx11 (E)	0,17	0,17	0,16	0,16	0,18	….
….20	…. (T)	….	….	….	….	….	….
	Total	63,61					
	Jumlah Data	400					
	Rata-Rata	0,1590	0,16				

c. Analisis Similaritas Dokumen Tesis per Program Studi

Pengujian menggunakan dokumen abstrak *dataset* tiap program studi Magister Ilmu Komputer (MKOM), Magister Ilmu Komunikasi (MIKOM), Magister Akuntansi (MAKSI) dan Magister Manajemen (MM) sebagai dataset yang semua data dokumennya berbeda-beda, dan saling dibandingkan masing-masing dari dokumen tersebut. Pada pengujian ini dilakukan pada sistem peneliti. Pada pengujian sistem peneliti ini digunakan nilai k-gram 5. Pada pengujian ini bertujuan untuk mencari rata-rata similaritas masing-masing program studi.

Pada Tabel 16 merupakan tabel berisi nilai persentase similaritas 39 dataset program studi MKOM pada sistem peneliti dalam satuan persen (%). Kemudian dihitung rata-rata dari total nilai persentase similaritas dibagi total data perbandingan. Dan dapat disimpulkan rata-rata nilai persentase similaritas pada dokumen 39 dataset program studi

MKOM sebesar 20,95 %. Pada Tabel 17 merupakan tabel berisi nilai persentase similaritas 10 dataset program studi MIKOM pada sistem peneliti dalam satuan persen (%). Kemudian dihitung rata-rata dari total nilai persentase similaritas dibagi total data perbandingan. Dan dapat disimpulkan rata-rata nilai persentase similaritas pada dokumen 10 dataset program studi MIKOM sebesar 21,07 %. Pada Tabel 18 merupakan tabel berisi nilai persentase similaritas 11 dataset program studi MAKSI pada sistem peneliti dalam satuan persen (%). Kemudian dihitung rata-rata dari total nilai persentase similaritas dibagi total data perbandingan. Dan dapat disimpulkan rata-rata nilai persentase similaritas pada dokumen 11 dataset program studi MAKSI sebesar 26,63 %. Pada Tabel 19 merupakan tabel berisi nilai persentase similaritas 40 dataset program studi MM pada sistem peneliti dalam satuan persen (%). Kemudian dihitung rata-rata dari total nilai persentase similaritas dibagi total data perbandingan. Dan dapat disimpulkan rata-rata nilai persentase similaritas pada dokumen 40 dataset program studi MM sebesar 27,9 %.

Tabel 16 Uji Coba 39 *Dataset* Program Studi MKOM

No.	NIM	A	B	C	D	E	… AM
1	141160xx95 (A)	100	8,58	5,15	5,88	3,59	….
2	151160xx63 (B)	8,58	100	7,75	7,29	6,89	….
3	151160xx86 (C)	5,15	7,75	100	4,65	4,96	….
4	151160xx93 (D)	5,88	7,29	4,65	100	7,47	….
5	161160xx48 (E)	3,59	6,89	4,96	7,47	100	….
….39	…. (AM)	….	….	….	….	….	….
	Total	15954					
	Jumlah Data	1521					
	Rata-Rata	20,95					

Tabel 17 Uji Coba 10 *Dataset* Program Studi MIKOM

No.	NIM	A	B	C	D	E	… J
1	157160xx11 (A)	100	7,92	16,7	14,07	7,96	….
2	157160xx77 (B)	7,92	100	8,18	8,06	9,14	….
3	157160xx06 (C)	16,7	8,18	100	16,97	10,33	….
4	157160xx48 (D)	14,07	8,06	16,97	100	7,1	….
5	167160xx36 (E)	7,96	9,14	10,33	7,1	100	….
….10	…. (J)	….	….	….	….	….	….
	Total	2106,52					
	Jumlah Data	100					
	Rata-Rata	21,065	21,07				

Tabel 18 Uji Coba 11 *Dataset* Program Studi MAKSI

No.	NIM	A	B	C	D	E	… K
1	153260xx28 (A)	100	13,13	15,32	12,36	29,89	….
2	153260xx68 (B)	13,13	100	15,89	10,13	14,6	….
3	153260xx09 (C)	15,32	15,89	100	8,22	13,97	….
4	163260xx00 (D)	12,36	10,13	8,22	100	13,17	….
5	163260xx26 (E)	29,89	14,6	13,97	13,17	100	….
….11	…. (K)	….	….	….	….	….	….
	Total	3222,52					
	Jumlah Data	121					
	Rata-Rata	26,632	26,63				

Tabel 19 Uji Coba 40 *Dataset* Program Studi MM

No.	NIM	A	B	C	D	E	… AN
1	153160xx55 (A)	100	15,72	12,25	13,62	8,3	….
2	153160xx39 (B)	15,72	100	12,84	12,63	11,64	….
3	153160xx20 (C)	12,25	12,84	100	14,74	12,15	….
4	153160xx33 (D)	13,62	12,63	14,74	100	11,06	….
5	163160xx83 (E)	8,3	11,64	12,15	11,06	100	….
….40	…. (AN)	….	….	….	….	….	….
	Total	22319,48					
	Jumlah Data	1600					
	Rata-Rata	27,89	27,9				

4. KESIMPULAN DAN SARAN

Berdasarkan dari penelitian dan pengujian yang telah dilakukan pada sistem Analisis Kemiripan Dokumen Tesis menggunakan algoritma *Rabin-Karp* dan *Dice Coefficient Similarity* pada dokumen abstrak tesis mahasiswa Universitas Budi Luhur, maka dapat disimpulkan diantaranya:

a. Sistem dapat mendeteksi tingkat kemiripan sebuah dokumen, dengan data penelitian berupa dokumen abstrak tesis mahasiswa Universitas Budi Luhur.
b. Penggunaan metode *K-Gram* dan Dice Coefficient Similarity terhadap algoritma Rabin Karp cukup baik dengan menghasilkan persentase kemiripan mencapai 100%.
c. Nilai k-gram mempengaruhi persentase kemiripan, penggunaan k-gram yang tepat sangat diperlukan.
d. Nilai k-gram terlalu kecil dapat mengakibatkan nilai persentase kemiripan yang besar,

begitupun sebaliknya.

e. Pada pengujian mencari akurasi sistem peneliti dengan pembanding software Check-Plagiarism menghasilkan akurasi sebesar 94,7%. Sedangkan pengujian mencari rata-rata waktu proses sistem peneliti menghasilkan rata-rata waktu proses sebesar 0,16 detik. Kemudian analisis similaritas dokumen tesis per program studi menghasilkan rata-rata nilai persentase similaritas pada dokumen 39 dataset program studi MKOM sebesar 20,95 %, kemudian pada dokumen 10 dataset program studi MIKOM sebesar 21,07%, kemudian pada dokumen 11 dataset program studi MAKSI sebesar 26,63%, dan kemudian pada dokumen 40 dataset program studi MM sebesar 27,9%.

Saran dari peneliti sebagai pengembangan lebih lanjut untuk sistem ini agar diimplementasikan dengan fungsionalitas yang lebih baik, yaitu pengembangan sistem dapat dikombinasikan dengan metode lain misalnya algoritma *Winnowing* atau *Stemming Najief Andriani* ataupun yang lainnya, agar lebih baik hasil yang didapat dengan melihat perbandingan hasil akurasi nilai similaritas pada metode lain tersebut. Serta dapat melakukan pendeteksian similaritas satu penuh file dokumen tesis, bukan hanya bagian abstrak.

DAFTAR PUSTAKA

[1] Alamsyah, N., 2017, Perbandingan Algoritma Winnowing Dengan Algoritma Rabin Karp Untuk Mendeteksi Plagiarisme Pada Kemiripan Teks Judul Skripsi, *Technologia: Jurnal Ilmiah*, No.3, Vol.8, 124.

[2] Irawan, C., dan Pratama, M. R., 2021, Perbandingan Algoritma Boyer-Moore dan Brute Force pada Pencarian Kamus Besar Bahasa Indonesia Berbasis Android, *BIOS : Jurnal Teknologi Informasi dan Rekayasa Komputer*, No.2, Vol.1, 54–60.

[3] Singla, N., dan Garg, D., 2012, String Matching Algorithms and their Applicability in various Applications, vol 6, 218–222.

[4] Leonardo, B,. dan Hansun, S., 2017, Text documents plagiarism detection using Rabin-Karp dan Jaro-Winkler distance algorithms, *Indonesian Journal of Electrical Engineering and Computer Science*, No.2, Vol.5, 462–471.

[5] Priambodo, J., 2018, Pendeteksian Plagiarisme Menggunakan Algoritma Rabin-Karp, *Jurnal Informatika Universitas Pamulang*, No.1, Vol.3, 39–45.

[6] Purba, A. H., dan Situmorang, Z, 2017, Analisis Perbandingan Algoritma Rabin-Karp Dan Levenshtein Distance Dalam Menghitung Kemiripan Teks, *Jurnal Teknik Informatika Unika St. Thomas (JTIUST)*, Vol.2, 24–32.

[7] Filcha, A., dan Hayaty, M., 2019, Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa, *JUITA : Jurnal Informatika*, No.1, Vol.7, 25.

[8] Pamungkas, H. Y., dan Fitrianingsih, 2019, Deteksi Similaritas Dokumen Ilmiah Menggunakan Algoritma Rabin-Karp, *Jurnal Ilmiah Informatika Komputer*, No.3, Vol.24, 209–219.

[9] Putra P. N., dan Sularno, S., 2019, Penerapan Algoritma Rabin-Karp Dengan Pendekatan Synonym Recognition Sebagai Antisipasi Plagiarisme Pada Penulisan Skripsi, *Jurnal Teknologi Dan Sistem Informasi Bisnis*, No.2, Vol.1, 48–58.

[10] Rahmaddeni, Didik, S., dan Agustin, 2018, Sistem Pendeteksi Tingkat Kesamaan Teks pada Pengusulan Proposal, *SATIN - Sains dan Teknologi Informasi*, No.2, Vol.4, 84–91.