

DETEKSI PLAGIARISME PADA DOKUMEN SKRIPSI BERDASARKAN TINGKAT KESAMAAN DENGAN MENGGUNAKAN METODE LONGEST COMMON SUBSEQUENCE

Imam Nawawi¹, Putra Prima Arhandi², Faisal Rahutomo³

^{1,2,3} Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang
Malang, Indonesia

e-mail: nawawi932@gmail.com¹; putraprima@polinema.ac.id²; faisalrahutomo@polinema.ac.id³

Abstrak

Plagiarisme adalah tindakan menyalin, mengambil karangan atau pendapat orang lain tanpa adanya izin tertulis dan menjadikannya seolah-olah pendapatnya sendiri. Hal ini masih menjadi fenomena yang sering terjadi pada instansi akademik atau non akademik. Namun pada jurusan Teknologi Informasi Politeknik Negeri Malang belum ada aplikasi yang dapat digunakan untuk mendeteksi plagiarisme. Berdasarkan permasalahan diatas, maka dibuatlah aplikasi deteksi plagiarisme pada dokumen Tugas Akhir / Skripsi yang bernama *Document plagiarism Detection (Doristec)* dengan menggunakan metode *Longest Common Subsequence (LCS)* dengan membuat modifikasi untuk mencapai hasil yang sesuai dengan perancangan. Aplikasi ini dibuat dengan tujuan untuk mengetahui seberapa besar tingkat plagiarisme pada dokumen Tugas Akhir / Skripsi yang nantinya akan diketahui oleh mahasiswa dan panitia Laporan Akhir dan Skripsi. Berdasarkan penelitian yang telah dilakukan, Metode *Longest Common Subsequence* dapat digunakan untuk deteksi plagiarisme dengan perbandingan dua atau lebih dokumen pembanding, dapat menguji lebih dari satu kalimat dan lebih dari satu kandidat pembanding, serta hasil pengujian akurasi yang dilakukan mendapatkan hasil akurasi yang tinggi. Hal ini dapat menjadi alternatif bagi mahasiswa jurusan Teknologi Informasi Politeknik Negeri Malang dalam melakukan pengujian terhadap penelitiannya dan bagi panitia Laporan Akhir dan Skripsi dapat melakukan monitoring Laporan Akhir dan Skripsi.

Kata kunci: Kemiripan, Plagiarisme, Longest Common Subsequence

Abstract

Plagiarism is the act of copying, taking essays or opinions of others without any written permission and making it as if his own opinion. This is still a phenomenon that often occurs in academic or non-academic institutions. However, in the Information Technology Department, Malang State Polytechnic there are no applications that can be used to detect plagiarism. Based on the above problems, a plagiarism detection application was developed for the Final Project / Thesis document it is called Document plagiarism Detection (Doristec) using the Longest Common Subsequence (LCS) method with modifications to achieve results that are in accordance with the design. This application aims at finding out the level of plagiarism in the Final Project / Thesis document which later will be known by students and the Final Report and Thesis committee. Based on research that has been done, The Longest Common Subsequence method can be used to detect plagiarism by comparing two or more comparative documents, can test more than one sentence and more than one comparative candidate, and the results of accuracy testing performed to get high accuracy results. This can be an alternative for students majoring in Information Technology at the State Polytechnic of Malang in testing their research and for the Final Report and Thesis committee can monitor the Final Report and Thesis.

Keywords : Similarities, Plagiarism, Longest Common Subsequence

PENDAHULUAN

Tugas akhir atau skripsi merupakan penelitian yang harus dilakukan oleh mahasiswa sebagai syarat kelulusan. Tuntutan bagi setiap mahasiswa untuk membuat dan menyelesaikan tugas akhir atau skripsi sebagai bukti dari hasil belajar selama di perguruan tinggi. Membuat laporan tugas akhir atau skripsi adalah langkah yang harus dilakukan oleh mahasiswa untuk mendokumentasikan penelitiannya. Membaca tugas akhir atau skripsi terdahulu diperlukan sebagai referensi bagi mahasiswa dan gambaran dari laporan penelitian. Dalam penulisan laporan penelitian tidak semua mahasiswa mampu menulis berdasarkan argumennya sendiri melainkan dapat mengambil dari laporan penelitian terdahulu, ketika mengambil dari penelitian terdahulu namun tidak dilakukan modifikasi maka akan menimbulkan adanya pelanggaran yaitu plagiarisme. Situasi seperti ini dapat menghambat perkembangan dari aktivitas kreatif mahasiswa maka diperlukan aplikasi yang dapat mendeteksi adanya plagiarisme pada laporan tugas akhir atau skripsi.

Berdasarkan sumber yang penulis amati dalam [1] yang dilakukan oleh Alamsyah, Nur membahas mengenai deteksi plagiarisme pada judul skripsi, metode yang digunakan adalah algoritma Winnowing. Pada penelitian tersebut, aplikasi yang dibangun dapat memberikan informasi berupa hasil plagiarisme pada judul skripsi. Namun, pengujian yang dapat dilakukan pada penelitian tersebut, terbatas pada teks judul skripsi.

Penelitian yang dilakukan oleh Widiastuti, Inta pada tahun 2014 yang berjudul "Aplikasi Pendeteksi Kemiripan Dokumen Menggunakan Algoritma Rabin Karp" [2]. Dalam penelitiannya mengenai deteksi kemiripan antara dua dokumen dengan dilakukan perbandingan pattern pada dokumen teks, metode yang digunakan untuk mendeteksi kemiripan dokumen adalah Algoritma Rabin Karp. Penelitian ini berhasil menciptakan aplikasi deteksi plagiarisme dokumen yang bekerja

dengan baik serta memperlihatkan hasil similarity-nya. Namun, pada penelitian ini hanya dapat melakukan pengujian antara dua dokumen.

Dikarenakan pada Jurusan Teknologi Informasi (JTI) belum ada aplikasi deteksi plagiarisme untuk laporan tugas akhir atau skripsi, oleh karena itu dalam skripsi ini penulis membuat sistem deteksi plagiarisme dokumen skripsi menggunakan metode *Longest Common Subsequence* (LCS). Proses deteksi menggunakan perbandingan pada dokumen yang di upload mahasiswa dengan dataset yang ada pada database. Dengan adanya aplikasi ini, diharapkan dapat membantu untuk mendeteksi adanya plagiarisme dokumen tugas akhir atau skripsi.

Dari penelitian-penelitian terdahulu yang penulis amati dan penulis pelajari, penulis akan melakukan penelitian menggunakan metode *Longest Common Subsequence* yang diharapkan dapat mendapatkan sistem yang lebih kompleks untuk menyempurnakan sistem dari penelitian terdahulu.

Berdasarkan latar belakang masalah yang diuraikan, maka dapat dirumuskan masalah sebagai berikut :

1. Bagaimana mendeteksi plagiarisme pada dokumen digital menggunakan metode *Longest Common Subsequence*?
2. Bagaimana menciptakan aplikasi yang dapat mendeteksi adanya plagiarisme dengan tingkat akurasi yang tinggi?

LANDASAN TEORI

Pada landasan teori ini akan diuraikan kajian pustaka dan dasar teori yang mendukung penelitian ini. Dasar teori tersebut diperoleh dari beberapa referensi yang relevan dengan topik yang diangkat dalam penelitian ini. Dalam bab ini akan dijelaskan mengenai kajian pustaka dan metode *Longest Common Subsequence* (LCS) yang merupakan topik utama diadakannya penelitian ini.

A. Metode Longest Common Subsequence

Longest Common Subsequences merupakan masalah dalam proses mencari sebuah *subsequence* terpanjang dari beberapa *sequence* (biasanya hanya terdiri dari 2 buah *sequence*), dimana sebuah *subsequence* tersebut merupakan sekumpulan kata dari *sequence* yang memiliki urutan kemunculan yang sama seperti yang dikatakan oleh [7]. *Subsequence* dari sebuah string A adalah sekumpulan kata yang ada pada A dengan urutan kemunculan kata yang sama.

Dalam rangkaian Z yang merupakan *subsequence* dari X (sebagai *sequence*) dimana $X = \langle x_1, x_2, \dots, x_m \rangle$, jika terdapat urutan menaik $\langle i_1, i_2, \dots, i_n \rangle$ yang merupakan indeks X untuk semua $j=1, 2, \dots, k$, yang memenuhi $x_{i_j} = z_j$. misal pada common subsequences dari 2 rangkaian string1 = tadi saya sedang makan dikantin sekolah pada jam istirahat dan string2 = saya sedang mengerjakan tugas seni budaya pada jam istirahat, maka saya sedang, dan pada jam istirahat adalah common subsequences dari string1 dan string2. Longest Common Subsequences

adalah common subsequences dari rangkaian hasil yang paling panjang pada 2 rangkaian string (string1 dan string2), sehingga Longest Common Subsequences yang terpilih adalah pada jam istirahat.

Mendapatkan Panjang LCS

Dalam menjabarkan proses pengukuran panjang akan dijelaskan sebagai berikut : Misal m adalah panjang string1, n adalah panjang string2, dan tab adalah matriks berukuran m x n, dan tab[i, j] adalah panjang LCS dari subsequences pertama yang terdiri dari i kata pada string1 dengan subsequences kedua yang terdiri dari j kata pada string2, maka untuk $1 \leq i \leq m$ dan $1 \leq j \leq n$, sesuai dengan fungsi rekursif diatas, maka :

$$tab[i, j] = \begin{cases} tab[i-1, j-1] + 1, & s1[i] = s2[j] \\ \max(tab[i-1, j], tab[i, j-1]), & \text{otherwise} \end{cases} \quad (1)$$

Teks yang dibentuk yaitu string1 = tadi saya sedang makan dikantin sekolah pada jam istirahat dan string2 = saya sedang mengerjakan tugas seni budaya pada jam istirahat, tabel matrik yang terbentuk berukuran 9 x 7 yang isinya sebagai berikut

Tabel 1. Hasil Kalkulasi LCS

		1	2	3	4	5	6	7	8	9
		tadi	saya	sedang	makan	dikantin	sekolah	pada	jam	istirahat
1	saya	0	1	1	1	1	1	1	1	1
2	sedang	0	1	2	2	2	2	2	2	2
3	mengerjakan	0	1	2	2	2	2	2	2	2
4	tugas	0	1	2	2	2	2	2	2	2
5	pada	0	1	2	2	2	2	3	3	3
6	jam	0	1	2	2	2	2	3	4	4
7	istirahat	0	1	2	2	2	2	3	4	5

Dan sesuai dengan arti notasi dari setiap $tab[i, j]$ yang berisi pada tabel 1 bahwa LCS dari string1 dan string2 adalah $tab[m, n]$ atau dalam kasus ini adalah ditemukan 5 kata.

Sedangkan untuk mendapatkan LCS yang dimaksud, kita bisa merunut balik pada tabel 1 yang dimulai dari $tab[m, n]$. Dalam runut balik ini, yang kita lakukan sebenarnya hanyalah merunut balik pilihan yang dilakukan pada saat kalkulasi $tab[i, j]$.

Mendapatkan LCS

Tabel 2. Hasil Penurutan Balik Untuk Mendapatkan LCS

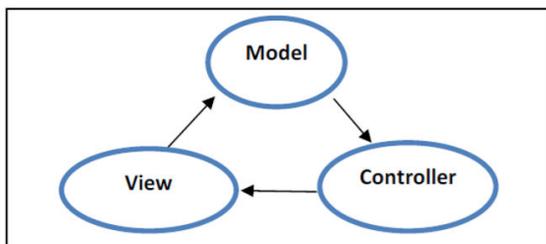
		1	2	3	4	5	6	7	8	9
		tadi	saya	sedang	makan	dikantin	sekolah	pada	jam	istirahat
1	Saya	0	1	1	1	1	1	1	1	1
2	Sedang	0	1	2	2	2	2	2	2	2
3	Mengerjakan	0	1	2	2	2	2	2	2	2
4	Tugas	0	1	2	2	2	2	2	2	2
5	Pada	0	1	2	2	2	2	3	3	3
6	Jam	0	1	2	2	2	2	3	4	4
7	Istirahat	0	1	2	2	2	2	3	4	5

Untuk setiap $1 \leq i \leq m$ dan $1 \leq j \leq n$, jika $string1[i]$ sama dengan $string2[j]$, maka kata itu pasti terdapat pada LCS. Selain itu, cek, dari mana mendapatkan LCS saat itu, lakukan pemilihan yang sama. Lakukan hal tersebut dimulai dari $i = m$ dan $j = n$. Setelah hal itu dilakukan maka akan terbentuk tabel di bawah ini. Di mana cell yang dihitamkan menandakan kata yang masuk ke dalam LCS.

Berdasarkan pada tabel 2 dapat dilihat bahwa hasil LCS yang ditemukan adalah saya sedang, pada jam istirahat.

B. Model View Controller (MVC)

Menurut [17] *Model View Controller* (MVC) merupakan *framework* yang menerapkan model MVC, model ini diterapkan guna dapat memberi dampak kedisiplinan dalam menuliskan kode program serta mempermudah dalam pengembangan aplikasi karena telah terstruktur dengan baik, model MVC dibagi menjadi 3 bagian yaitu : *model*, *view*, dan *controller* seperti pada gambar 1.



Gambar 1. Model View Controller

Bagian-bagian tersebut memiliki arti sebagai berikut:

Model

Model merupakan bagian yang berinteraksi dengan *database*.

View

View merupakan pages yang tampil di halaman web.

Controller

Controller akan mengatur interaksi antar pages/ mengontrol hubungan view dan model.

METODE

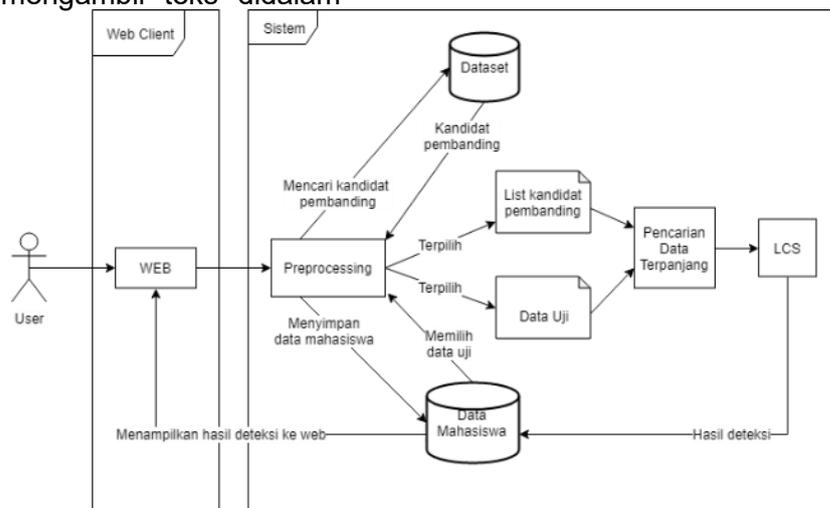
A. Metode Pengambilan Data

Metode pengambilan data yang dilakukan untuk melaksanakan penelitian ini yaitu mencari referensi berupa penelitian terdahulu dan jurnal. penelitian yang berjudul Aplikasi Pendeteksi Kemiripan Dokumen Menggunakan Algoritma Rabin Karp tersebut penulis gunakan sebagai referensi utama. Setelah referensi berhasil dipelajari kemudian mengumpulkan dataset dari dokumen – dokumen tugas akhir dan skripsi terdahulu, dataset tersebut berupa dokumen word atau pdf skripsi mahasiswa, dari dokumen tersebut akan dilakukan proses konversi dengan mengambil teks dan menghilangkan semua elemen lain seperti gambar dan simbol – simbol, kemudian teks tersebut di simpan kedalam database sebagai dataset, sehingga penulis memutuskan untuk menggunakan metode Longest Common Subsequences (LCS) sebagai metode untuk mencari kesamaan dari dokumen yang di uji. Metode tersebut diambil dari referensi jurnal yang berjudul Plagiarism detection using document similarity based on distributed representation.

B. Metode Pengolahan Data

Data yang didapat dari data tugas akhir dan skripsi pada Jurusan Teknologi Informasi merupakan data yang diambil secara manual dari website digilib.jti.polinema.ac.id. data tersebut akan dijadikan sebagai dataset sistem, data tersebut perlu dilakukan pengolahan agar mempermudah dalam proses deteksi oleh sistem, pengolahan yang dilakukan yaitu ketika proses upload data tugas akhir dan skripsi akan dilakukan convert dari dokumen pdf/docx/doc menjadi teks, hal ini dilakukan untuk mengambil teks didalam

dokumen, sehingga gambar dan objek-objek lain tidak diambil, setelah berhasil mengambil keseluruhan teks kemudian data disimpan ke database, proses convert ini juga berlaku ketika proses upload dokumen yang akan diuji, proses ini disebut dengan preprocessing. Ada juga yang terdapat dalam proses preprocessing selain convert dokumen adalah filter data-data pada dataset berdasarkan kata kunci yang akan menjadi kandidat pembanding. Tahapan tersebut akan digambarkan pada arsitektur sistem, seperti pada gambar 2.



Gambar 2. Arsitektur Sistem

Convert Dokumen Menjadi Teks

Convert dokumen bertujuan untuk mempermudah dalam penyimpanan ke database dan meminimalisir penggunaan logika yang terlalu banyak, sehingga proses ini dapat mempersingkat waktu karna ketika proses deteksi dijalankan tidak perlu convert dokumen berulang-ulang.

Filter Kandidat Pembanding

Filter Kandidat Pembanding merupakan sebuah fungsi yang terdapat didalam sistem yang bertujuan untuk mencari data-data didalam dataset, pencarian dilakukan berdasarkan kata kunci yang sesuai dengan data uji. Sehingga data uji nantinya dilakukan proses perbandingan dengan kandidat pembanding saja, karna data yang tidak memiliki kata kunci yang sama tidak akan diambil menjadi kandidat pembanding.

Proses ini memiliki dampak yang besar karna sistem tidak perlu melakukan perbandingan data uji dengan data-data pada dataset, serta dapat mempersingkat waktu yang dibutuhkan dalam proses deteksi menggunakan LCS.

Pencarian Dokumen Terpanjang

Pencarian dokumen terpanjang merupakan proses yang dilakukan sebelum menjalankan LCS, karena data parameter yang dibutuhkan LCS, parameter 1 harus lebih panjang dari parameter 2, sehingga dibuat sebuah fungsi pencarian data terpanjang untuk mempersiapkan variabel yang akan dimasukkan pada parameter LCS. Sehingga sistem ini benar-benar berjalan secara dinamis.

Jika $X > Y$		Jika $Y > X$
Maka		Maka
$S1 = X$	Atau	$S1 = Y$
$S2 = Y$		$S2 = X$

Keterangan :

S1 = variabel yang menyimpan yang panjang
S2 = variabel yang menyimpan lebih pendek
X = data uji
Y = kandidat pembandingan

Longest Common Subsequences (LCS)

Longest Common Subsequences merupakan cara yang digunakan untuk menghitung dan mencari setiap rangkaian kata dalam deteksi plagiarisme. Pencarian ini dilakukan dengan cara membuat tabel matriks yang menyimpan hasil penghitungan LCS, matrik tersebut akan dibuat dengan rumus :

$$tab[i, j] = \begin{cases} tab[i-1, j-1] + 1, & s1[i] = s2[j] \\ \max(tab[i-1, j], tab[i, j-1]) \end{cases} \quad (2)$$

Keterangan :

tab[i, j] = panjang LCS yang ditemukan
S1 = data uji
S2 = kandidat pembandingan

Karena pada dasar metode LCS hanya mengambil 1 hasil LCS dengan rangkaian kata terpanjang didalam 1 dokumen, maka dibuat sebuah kondisi perulangan untuk menemukan lebih dari 1 rangkaian kata yang plagiat. Kondisi pada perulangan ini dibatasi dengan jumlah kata hasil LCS lebih dari 2 kata, dalam kondisi ini diambil berdasarkan struktur kalimat terbaik adalah kalimat yang memiliki lebih dari 2 rangkaian kata. Dengan adanya kondisi ini akan

membuat sistem dapat melakukan pencarian secara berulang-ulang dalam 1 dokumen pembandingan, sehingga dalam 1 dokumen pembandingan sistem dapat mencari dan mendapatkan lebih dari 1 hasil LCS. Selain itu metode juga dibuat secara dinamis dengan dibuat sebuah kondisi untuk mengetahui jumlah data kandidat pembandingan, sehingga sistem dapat berjalan secara berulang-ulang sebanyak data kandidat pembandingan.

HASIL DAN PEMBAHASAN

Pengujian Metode merupakan pengujian yang dilakukan pada aplikasi dengan terfokus pada metode yang diterapkan pada aplikasi, pengujian ini dilakukan untuk mengetahui berjalannya metode LCS dapat berfungsi dengan baik. Pengujian manual dengan metode LCS akan menggunakan tiga studi kasus dan metode pengujian *recall precision* yang diberikan. Berikut ini pengujian menggunakan tiga studi kasus yang telah disediakan berupa tahapan sebagai berikut :

A. Kasus Pertama

Pengujian pada kasus pertama merupakan pengujian yang dilakukan dengan memberikan dokumen yang sama antara kandidat pembandingan dan data uji, berikut ditunjukkan pada tabel 1.

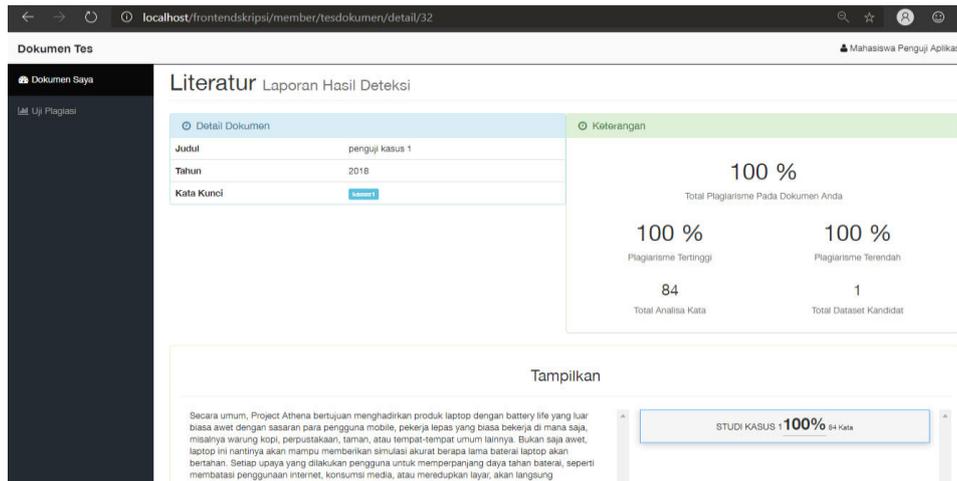
Dari hasil pengujian pada studi kasus pertama menggunakan *recall precision*, hasil perhitungan *recall* memiliki tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi sebesar 100% ,sedangkan perhitungan *precision* didapatkan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem sebesar 100%.

Tabel 1. Perhitungan Manual Pada Studi Kasus Pertama

No	Kategori Dokumen	Jumlah Kata	Jumlah kata ditemukan kata	Jumlah kalimat ditemukan
1	Dokumen Uji	84 Kata	-	-
2	Kandidat pembandingan	84 Kata	84 kata	1

$$Recall = \frac{1}{1} \times 100 = 100\%$$

$$Precision = \frac{1}{1} \times 100 = 100\%$$



Gambar 3. Perhitungan Kasus Pertama Pada Sistem

B. Kasus Kedua

Pengujian pada kasus kedua merupakan pengujian yang dilakukan dengan memberikan dokumen data uji dengan memiliki beberapa kemiripan

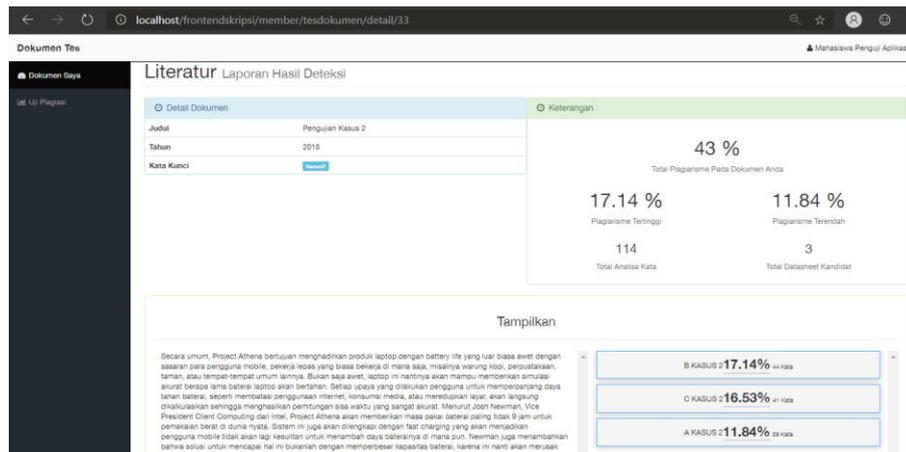
kalimat pada tiga kandidat pembanding yang diujikan, berikut ditunjukkan pada tabel 2.

Tabel 2. Perhitungan Manual Pada Studi Kasus Kedua

No	Kategori Dokumen	Jumlah Kata	Jumlah ditemukan kata	Jumlah kalimat ditemukan
1	Dokumen Uji	262 Kata	-	-
2	Kandidat pembanding 1	134 Kata	29 kata	1
3	Kandidat pembanding 2	197 kata	44 kata	1
4	Kandidat pembanding 3	156 kata	41 kata	1

$$Recall = \frac{3}{3} \times 100 = 100\%$$

$$Precision = \frac{3}{3} \times 100 = 100\%$$



Gambar 4. Perhitungan Kasus Kedua Pada Sistem

Dari hasil pengujian pada studi kasus kedua menggunakan recall precision, hasil perhitungan recall memiliki tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi sebesar 100% ,sedangkan perhitungan precision didapatkan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem sebesar 100%.

C. Kasus Ketiga

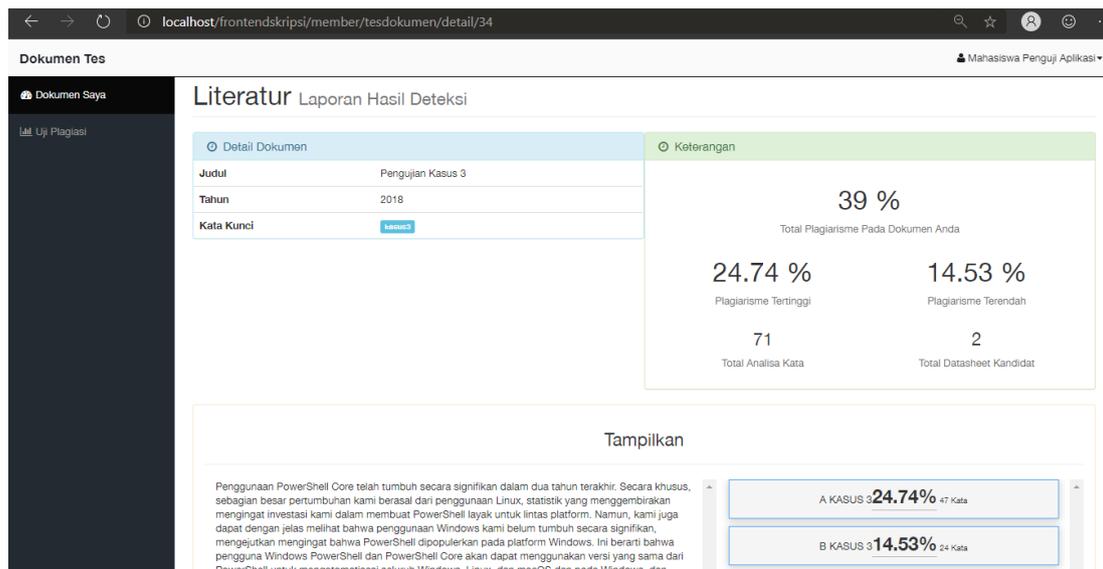
Pengujian pada kasus ketiga merupakan pengujian yang dilakukan dengan memberikan dokumen data uji

dengan memiliki beberapa kemiripan kalimat pada dua kandidat pbanding, berikut ditunjukkan pada tabel 3.

Dari hasil pengujian pada studi kasus ketiga menggunakan recall precision, hasil perhitungan recall memiliki tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi sebesar 100% ,sedangkan perhitungan precision didapatkan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem sebesar 100%.

Tabel 3. Perhitungan Manual Pada Studi Kasus Ketiga

No	Kategori Dokumen	Jumlah Kata	Jumlah kata disetiap LCS	Jumlah kalimat ditemukan	Total Kata
1	Dokumen Uji	183 Kata	-	-	-
2	Kandidat pbanding 1	198 Kata	23 kata, 18 kata, 6 kata	3	47 kata
3	Kandidat pbanding 2	180 kata	9 kata, 6 kata, 5 kata, 4 kata	4	24 kata



Gambar 5. Perhitungan Kasus Ketiga Pada Sistem

SIMPULAN

Adapun kesimpulan yang dapat diambil dari hasil penelitian yang dilakukan mengenai deteksi plagiarisme pada dokumen skripsi berdasarkan tingkat kesamaan dengan menggunakan *longest common subsequence* yaitu metode *Longest Common Subsequence* dapat digunakan untuk deteksi plagiarisme dengan perbandingan dua atau lebih dokumen pembanding, aplikasi yang dibuat telah berhasil menerapkan metode *Longest Common Subsequence* untuk menguji lebih dari satu kalimat dan lebih dari satu kandidat pembanding serta hasil pengujian yang telah dilakukan dengan tiga jenis setudi kasus sesuai dengan hasil akurasi yang tinggi. Saran yang dapat diberikan dari hasil penelitian untuk pengembangan sistem ini ke depannya yaitu dengan menambah *library* lain seperti memperluas *dataset* diluar jurusan Teknologi Informasi Politeknik Negeri Malang, *crawling* data dari internet. Selain itu sistem dapat dikombinasikan dengan algoritma lain untuk mendapatkan tingkat akurasi yang lebih maksimal.

REFERENSI

- [1] Alamsyah, Nur, "Deteksi Plagiarisme Tingkat Kemiripan Judul Skripsi Dengan Algoritma Winnowing,"

- Technologia, Vol.8, No.4, pp.205-213, Desember 2017.
- [2] Widiastuti, Inta, Aplikasi Pendeteksi Kemiripan Dokumen Menggunakan Algoritma Rabin Karp, Malang : Politeknik Negeri Malang (2014).
- [3] Pasma Cadea Mikha, Pengembangan Sistem Pendeteksi Kemiripan Karya Pada Inaicta 2013, Malang : Politeknik Negeri Malang (2014).
- [4] Kharismadita Paratisa, Implementasi Tokenizing Plus Pada Sistem Pendeteksi Kemiripan Jurnal Skripsi Di Program Studi Teknik Informatika Politeknik Negeri Malang, Malang : Politeknik Negeri Malang (2015).
- [5] Roshinta Trisna Ali, Analisis Skema-Skema Kemiripan Vektor Pada Sistem Penilaian Essay Online, Malang : Politeknik Negeri Malang (2016).
- [6] Nurhadi Riza A, Pengembangan Data Uji Sistem Komputasi Kemiripan Teks Secara Semantik Berbahasa Indonesia, Malang : Politeknik Negeri Malang (2016).
- [7] Kensuke Baba, Tetsuya Nakatoh, Toshiro Minami "Plagiarism detection using document similarity based on distributed representation" 8th International Conference on Advances in Information Technology, IAIT2016, 19-22 December 2016, Macau, China, doi:

- <https://doi.org/10.1016/j.procs.2017.06.038>.
- [8] Sidik., Betha, Framework Codeigniter, Bandung : Infomatika (2012).
- [9] Asad Abdi, Norisma Idris, Rasim M. Alguliyev, Ramiz M. Aliguliyev “*Plagiarism detection using linguistic knowledge*” Expert Systems With Applications (2015), doi: <https://doi.org/10.1016/j.eswa.2015.07.048>.
- [10] Col Jyotindu Debnath “*Plagiarism: A silent epidemic in scientific writing – Reasons, recognition and remedies*” *Medical Journal Armed Forces India*. (2016), doi: <https://doi.org/10.1016/j.mjafi.2016.03.010>.
- [11] L.K. Burdine, BA, M.B. de Castro Maymone, MD, DSc, N.A. Vashi, MD “*Text recycling: Self-plagiarism in scientific writing*” *International Journal of Women's Dermatology* (2018), doi: <https://doi.org/10.1016/j.ijwd.2018.10.002>.
- [12] Alwi, dkk, Tata Bahasa Baku Bahasa Indonesia, cet. 8, Jakarta: Pusat Bahasa dan Balai Pustaka, 2010.
- [13] Pratama, Rahmatullah Aditya, “Belajar UML - Use Case Diagram,” 21 02 2019. [Online]. Available: <https://www.codepolitan.com/mengenal-uml-diagram-use-case>.
- [14] Aini, Bariroh Isriya Nur, Identifikasi Kemiripan Judul Tugas Akhir PSTI Dan PSMI Di Politeknik Negeri Malang, Malang : Politeknik Negeri Malang (2014).
- [15] Basuki, Alwan Pribadi, “Menguasai Codeigniter Kasus Membangun Aplikasi Perpustakaan”, Yogyakarta, Lokomedia, 2016.
- [16] Galuh, Kresna, “Cara Menggunakan Bootstrap 3 untuk Membuat Web” 21 12 2015. [Online]. Available : <https://www.codepolitan.com/tutorial/cara-menggunakan-bootstrap-3-untuk-membuat-web>
- [17] Solikin, I, Perancangan Sistem Infomasi Penjualan Berbasis Framework Model View Controller (MVC) Pada PT Thamrin Brother Cabang Oki, Jurnal Media Informatika dan Komputer, (2014), 4 (1), pp. 1-13.