

Plagiarism Checker pada Sistem Manajemen Data Tugas Akhir

Made Hanindia Prami Swari¹⁾, Chrystia Aji Putra²⁾, I Putu Susila Handika³⁾

¹⁾²⁾ Program Studi Informatika, Fakultas Ilmu Kompuer UPN “Veteran” Jawa Timur

Jl. Rungkut Madya No.1, Surabaya

¹⁾madehanindia.fik@upnjatim.ac.id

²⁾ajiputra@upnjatim.ac.id

³⁾ Program Studi Teknik Informatika, STMIK STIKOM Indonesia

Jl. Tukad Pakerisan No. 97, Denpasar

³⁾ susila.handika@stiki-indonesia.ac.id

Abstrak

Plagiarisme merupakan isu yang santer berkembang khususnya di Perguruan Tinggi. Perpustakaan di Fasilkom UPN “Veteran” Jawa Timur telah memiliki sistem pencatatan skripsi mahasiswa yang bernama E-Read, seringkali mahasiswa yang akan menempuh skripsi akan mencari ide topik dan literatur pada perpustakaan. Untuk meminimalkan kemungkinan terjadinya plagiasi maka sistem E-Read dilengkapi dengan fitur *Plagiarism Checker* yang dibangun pada penelitian ini. *Plagiarism Checker* dibuat melalui tahapan pengembangan terstruktur menggunakan metode Waterfall yang terdiri dari analisis kebutuhan, perancangan, implementasi, dan pengujian sistem. Algoritma Jaro Winkler dipilih untuk mendeteksi similaritas dokumen abstrak calon skripsi dengan abstrak-abstrak yang telah tersimpan pada basis data E-Read karena berdasarkan penelitian terdahulu dinyatakan memiliki akurasi yang baik dan waktu komputasi yang cepat. Berdasarkan pengujian terhadap akurasi sistem, maka didapatkan *Plagiarism Checker* yang dibangun memiliki akurasi 40%, presisi 33,3%, recall 100%, dan F-Measure 35%. Selain itu berdasarkan analisis terhadap proses komputasi algoritma Jaro Winkler maka didapatkan bahwa algoritma ini berfokus pada pencarian kesamaan karakter sesuai *range* uji yang ditentukan, hal ini berarti bahwa algoritma ini belum memiliki mekanisme pengecekan makna kata yang seharusnya diperlukan dalam sistem pengecekan plagiasi dokumen. Melalui penelitian ini, mahasiswa dapat melakukan pengecekan awal abstrak dari calon skripsinya untuk meminimalkan terjadinya plagiasi.

Kata kunci: *Plagiarism Checker*, Jaro Winkler, Akurasi, Komputasi

Abstract

Plagiarism is a growing problem, especially in universities. Fasilkom UPN "Veteran" Jawa Timur library already has a management system of undergraduate-thesis document called E-Read, and students who generally participate in essays will search the library for topics and documents. To minimize the possibility of Plagiarism, E-Read system is equipped with the Plagiarism check feature built on this research. The Plagiarism Checker is completed through a structured development phase using the Waterfall method, which includes requirements analysis, design, implementation, and system testing. Jaro Winkler algorithm was chosen to detect the similarity between the job candidate's summary document and the summary stored in the E-Read database, since according to previous research, it has good precision and fast calculation time. Based on the test of the accuracy, it is found that the accuracy rate of the Plagiarism Checker constructed in this study is 40%, precision rate is 33.3%, recall value of 100%, and F-Measure is 35% compared with expert opinion. In addition, based on the analysis results of the Jaro Winkler algorithm calculation process, it is found that the algorithm focuses on finding similar characters according to the specified test range, which means that the algorithm does not have the meaning of the word that should be required in the document Plagiarism system. Through this research, students can conduct preliminary verification of the abstracts of their undergraduate-thesis candidates to minimize the occurrence of Plagiarism.

Keywords: *Plagiarm Checker*, Jaro Winkler, Accuracy, Computing

1. PENDAHULUAN

Kasus Plagiarisme merupakan isu yang selalu santer beredar, terutama di kalangan akademisi. Tindakan meniru karya ilmiah penulis lain tanpa mencantumkan sumber rujukan atau dengan kata lain mengakui karya tersebut merupakan buah pemikiran sendiri telah berlangsung lama di berbagai perguruan tinggi baik negeri maupun swasta di Indonesia [1]. Plagiarisme adalah tindakan melanggar hukum dan kode etik, hal ini tentunya menjadi momok dalam perkembangan ilmu pengetahuan. Pelaku plagiarisme bertindak dengan mengambil dan menuliskan karya penulis lain pada karya yang diakui sebagai miliknya tanpa menuliskan asal dari tulisan yang ditulis tersebut [2].

Plagiarism merupakan hal yang serius, dimana *Plagiarism* bukan saja berasal dari faktor kesengajaan, namun juga dari faktor ketidaksengajaan. Sangat mungkin calon penulis memiliki suatu ide tertentu (misalnya pembuatan sebuah sistem yang dapat digunakan di UPN), namun karena tidak membaca atau kesulitan mencari referensi maka yang bersangkutan tidak tahu bahwa calon karyanya sudah pernah dibuat oleh kakak kelasnya [3]. Lebih celaka lagi jika dosen pembimbing juga tidak mengetahui hal tersebut, maka hal ini dapat membuat baik dosen ataupun mahasiswa tersebut mendapat masalah kedepannya seperti beberapa kasus pencabutan gelar yang pernah terjadi di Indonesia beberapa waktu lalu. Berdasarkan hasil analisis masalah yang dilakukan maka penelitian ini dibuat dengan tujuan untuk membantu memecahkan masalah dari sisi metode melalui implementasi sistem deteksi Plagiarisme yang akan ditanam pada sistem Electronic Research Data Management (E-READ). Melalui sistem deteksi ini diharapkan mahasiswa ataupun calon dosen pembimbing dapat mengetahui apakah usulan calon proposal skripsi yang akan diajukan oleh mahasiswa terindikasi Plagiarisme dengan karya-karya skripsi yang telah dilakukan oleh senior-seniornya.

Sistem *Plagiarism* bekerja dengan membandingkan dokumen calon usulan proposal skripsi mahasiswa dengan abstrak dari seluruh karya skripsi koleksi perpustakaan Fakultas Ilmu Komputer UPN "Veteran" Jawa Timur. Implementasi sistem deteksi Plagiarisme ini membutuhkan sebuah algoritma pengecekan kemiripan antar dokumen yang mumpuni. Algoritma pengecekan kemiripan Jaro Winkler dipilih untuk diterapkan pada sistem deteksi ini karena berdasarkan beberapa penelitian yang telah dilakukan oleh peneliti sebelumnya seperti yang dilakukan oleh Bahri pada tahun 2017 [4], penelitian ini bertujuan membuat sebuah sistem deteksi *Plagiarism* antar dokumen menggunakan dua buah algoritma, yakni algoritma Jaro -Winkler Distance dan Algoritma Rabin-Karp untuk kemudian dianalisis performa kedua algoritma tersebut dari sisi kecepatan serta akurasi deteksi. Setelah penelitian dilakukan, maka dinyatakan bahwa Algoritma Jaro -Winkler distance menghasilkan performa yang lebih unggul dari sisi waktu pemrosesan, sedangkan Algoritma Rabin-Karp memberikan nilai ketepatan lebih baik. Penelitian terkait pedeteksian dokumen proposal skripsi pernah dilakukan oleh Sherly Christina dkk [5] setahun setelah penelitian yang dilakukan Bahri. Perbedaan penelitian ini dengan penelitian yang dikembangkan Sherly dkk tersebut adalah pada penelitian yang dilakukan Sherly dkk, sistem dibangun hanya untuk melakukan deteksi plagiasi antara proposal yang akan diajukan oleh mahasiswa yang dibandingkan dengan dataset yang telah disimpan pada *database*, sedangkan sistem yang akan dikembangkan peneliti adalah dengan menanamkan algoritma deteksi plagiasi ini pada sistem E-READ serta dilakukan pengujian akurasi sistem deteksi plagiasi menggunakan beberapa treshold. Penelitian Sherly, dkk menghasilkan bahwa algoritma yang diterapkan pada penelitian tersebut berhasil menemukan 80% data yang relevan, hal ini berarti bahwa aplikasi yang mereka bangun dapat memberikan kontribusi untuk mendeteksi tindakan plagiasi pada dokumen proposal skripsi mahasiswa dengan baik. Penelitian selanjutnya pernah dilakukan oleh Michael Julian Tannga, dkk pada tahun 2017 [6] tentang analisis perbandingan algoritma levenshtein distance dan Jaro winkler dalam studi kasus pembuatan aplikasi deteksi Plagiasi dan menghasilkan bahwa rata-rata nilai similaritas yang dihasilkan oleh algoritma Jaro Winkler cukup besar, yakni 80.92 % dan algoritma Levenshtein Distance rata-rata menghasilkan nilai similaritas sebesar 49.43%. Untuk rata-rata waktu proses eksekusi komputasi masing-masing algoritma menghasilkan nilai 0.054 detik untuk algoritma Jaro-Winkler, waktu eksekusi komputasi yang dibutuhkan oleh algoritma Levenshtein Distance menghasilkan waktu yang lebih lama, yaitu 0.138 detik. Penelitian yang bertujuan untuk membandingkan performa beberapa algoritma pencarian kemiripan kata juga pernah dilakukan oleh Yeny Rochmawati, dkk [7] pada tahun 2016 Algoritma yang dibandingkan pada penelitian ini adalah Levenshtein Distance, Hamming Distance, Damerau Levenshtein Distance dan Jaro Winkler Distance dan pengujian performa dilakukan menggunakan penilaian Mean Average Precision (MAP). Setelah dilakukan penelitian, maka dapat disimpulkan bahwa algoritma Jaro Winkler memiliki nilai MAP 0,87 dan dapat dikatakan merupakan algoritma terbaik.

Studi kasus yang sedikit berbeda dilakukan pada penelitian Imants Zarembo, dkk [8] pada tahun 2015. Tujuan dari penelitian ini adalah untuk mengevaluasi beberapa pendekatan pencocokan ontologi yang dapat digunakan dalam domain administrasi pertanahan. Untuk mendukung penelitian dikembangkan prototipe alat pencocokan ontologi menggunakan algoritma terpilih. Algoritma yang dievaluasi pada penelitian ini adalah Levensthein distance, Jaro -Winkler distance, Monge Elkan, dan Longest Common Substring. Berdasarkan penelitian yang telah dilakukan peneliti maka dapat dinyatakan bahwa hasil pencocokan ontologi paling baik dihasilkan dengan menggunakan algoritma Jaro -Winkler. Penelitian terakhir yang dirujuk adalah penelitian yang dilakukan oleh Taoxin Peng, dkk [9] pada tahun 2012. Peneliti melakukan perbandingan beberapa algoritma untuk menemukan kata pada 42 dataset yang ditentukan. Terdapat 5 algoritma pencocokan kata yang dibandingkan yakni Levensthein, Smith-Waterman, Jaro, Jaro-Winkler, dan Q-Gram dan dinyatakan bahwa Algoritma Jaro-Winkler dan Jaro secara umum menghasilkan performa terbaik dibandingkan algoritma lainnya terutama pada dataset yang memiliki tingkat eror cukup tinggi. Studi kasus pada penelitian yang dilakukan Zarembo, dkk serta Taoxin Peng, dkk memiliki perbedaan dengan penelitian ini, namun dalam implementasinya sama-sama melakukan pengecekan kesamaan string.

Berdasarkan seluruh studi literatur pada berbagai studi kasus tersebut maka dapat disimpulkan bahwa algoritma Jaro Winkler memiliki performa yang baik dari sisi kecepatan waktu proses maupun keakuratan hasil pengecekan kemiripan antar dokumen. Melalui implementasi algoritma ini diharapkan dapat dibangun sebuah fitur deteksi Plagiarisme pada sistem manajemen data skripsi (E-Read) yang telah dibangun sebelumnya. Sistem deteksi yang dibangun pada penelitian ini diharapkan memiliki performa baik dalam melakukan pendeteksian kemiripan antar dokumen calon skripsi mahasiswa dengan skripsi-skripsi koleksi Perpustakaan Fakultas Ilmu Komputer UPN "Veteran" Jawa Timur sehingga dapat menjadi benteng awal dalam kemungkinan akan dilakukannya tindakan plagiasi oleh mahasiswa baik secara sengaja ataupun tidak sengaja.

2. TINJAUAN PUSTAKA

Terdapat beberapa teori yang digunakan dalam penyelesaian penelitian ini, diantaranya terkait teori perancangan sistem dengan menggunakan model terstruktur menggunakan Conceptual Data Model (CDM), Physical Data Model (PDM), serta Data Flow Diagram (DFD), implementasi sistem dengan menggunakan pemrograman PHP dan basis data MySQL, serta terkait algoritma Jaro -Winkler serta pengujian sistem menggunakan confusion matrix menggunakan presisi, recall, dan akurasi. Pada bagian ini juga dijelaskan terkait algoritma Jaro -Winkler serta teori pengujian sistem.

2.1 Plagiarisme

Plagiarisme secara umum merupakan tindakan baik dilakukan dengan sengaja ataupun tidak untuk mendapatkan atau mencoba memperoleh nilai untuk suatu karya ilmiah, dengan cara mengutip sebagian atau seluruh karya ilmiah orang lain, tanpa mencantumkan sumber kutipan secara lengkap [10]. Berdasarkan cara penerapannya, etrdapat beberapa jenis plagiarisme [11], yaitu:

1. *Word Switch Plagiarism* dilakukan dengan menuliskan baik kalimat utuh atau penggalan kalimat ataupun paragraph dari penulis lain tanpa mengganti susunan kalimat tersebut serta tidak mencantumkan asal rujukan tersebut.
2. *Style Plagiarism* merupakan tindakan mengutip dengan *style* bahasa yang identik dengan pembuat tulisan yang asli, meskipun kalimat yang diambil sudah dirubah menggunakan kata-kata lain. Tindakan ini termasuk tindakan plagiasi apabila tidak mencantumkan sumber rujukan.
3. *Metaphor Plagiarism* adalah tindakan mengambil atau mencontek karya dari penulis lain yang bertujuan untuk memberi dan atau memperjelas makna dan arti pada tulisan yang dibuat. Tindakan ini boleh dilakukan asalkan mencantumkan asal rujukan.
4. *Idea Plagiarism* dilakukan dengan mengambil ide yang telah dibuat oleh penulis lain untuk melakukan penyelesaian terhadap suatu masalah yang akan dipecahkan namun dirujuk tanpa menuliskan asal sumber informasi tersebut.
5. *Self Plagiarism* merupakan tindakan plagiasi yang dilakukan dengan mengutip sebagian atau seluruh isi karyanya sendiri dengan tujuan untuk dikirimkan ke sejumlah jurnal tanpa menjelaskan bahwa karya tersebut merupakan karya sendiri yang pernah diterbitkan pada jurnal lain sebelumnya.

2.2 Algoritma Jaro-Winkler

Algoritma Jaro-Winkler merupakan salah satu algoritma populer yang digunakan untuk mengukur kesamaan antara dua string, penerapan algoritma ini sering digunakan pada kasus pendeteksian duplikasi string dalam dokumen. Semakin tinggi Jaro Winkler Distance untuk dua string maka semakin mirip dengan string tersebut. Nilai normalnya ialah nol menandakan tidak ada kesamaan dan dkk yang menandakan adanya kesamaan [12] atau dengan kata lain Semakin tinggi nilai Jaro -Winkler distance untuk dua string, semakin mirip kedua string tersebut. Skor normalnya adalah nilai 0 menandakan tidak ada kesamaan, dan 1 adalah sama persis. Dasar dari algoritma ini memiliki tiga bagian [13] :

1. Menghitung panjang string
2. Menemukan jumlah karakter yang sama di dalam dua string
3. Menemukan jumlah transposisi

Pada algoritma Jaro digunakan rumus untuk menghitung jarak (d_j) antara dua string yaitu S_1 dan S_2 tercantum pada persamaan (1).

$$d_j = \frac{1}{3} \times \left(\frac{m}{s_1} \right) + \left(\frac{m}{s_2} \right) + \left(\frac{m-t}{m} \right) \quad (1)$$

Dimana :

m = jumlah karakter yang sama persis

$|S_1|$ = panjang string 1

$|S_2|$ = panjang string 2

T = jumlah transposisi

Jarak teoritis dua buah karakter yang dikatakan sama dapat dibenarkan jika tidak melebihi batas yang tercantum pada persamaan (2).

$$\left(\frac{\max(|S_1|, |S_2|)}{s} \right) < -1 \quad (2)$$

Jaro Winkler Distance menggunakan prefix scale (p) yang memberikan tingkat penilaian yang lebih, dan prefix length (l) yang menyatakan panjang awalan yaitu panjang karakter yang sama dari string yang dibandingkan sampai ditemukannya ketidaksamaan, maka Jaro Winkler distancenya (d_w) dapat dihitung sesuai Persamaan (3).

$$d_w = d_j + \left(l_p(1 - d_w) \right) \quad (3)$$

Dimana:

d_j = Jaro Distance untuk string s_1 dan s_2

l = panjang prefiks umum diawal string nilai maksimalnya 4 karakter (panjang karakter yang sama sebelum ditemukan ketidaksamaan max 4).

p = konstanta scalling factor. Nilai standar untuk konstanta ini menurut Winkler adalah 0,1.

2.3 Confusion Matrix

Untuk melakukan pengujian sistem, maka dilakukan pencocokan antara hasil plagiasi yang dilakukan oleh manusia (ahli) terhadap hasil yang dikeluarkan. Perhitungan nilai akurasi dapat dilihat pada persamaan (4), persamaan (5) merupakan persamaan untuk menghitung nilai presisi, sedangkan recall dihitung dengan menggunakan persamaan (6), dan persamaan (7) merupakan persamaan untuk menghitung nilai f-measures [14].

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (4)$$

$$Presisi = \frac{TP}{TP+FP} \times 100\% \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (6)$$

$$F - Measure = 2 \times \frac{Presisi \times recall}{Presisi+recall} \times 100\% \quad (7)$$

Dimana :

TP = True Positif, dalam artian hasil deteksi menyatakan plagiat, dan hasil aktual juga menyatakan plagiat

FP = False Positif, dalam artian hasil deteksi menyatakan plagiat, dan hasil aktual menyatakan tidak plagiat

FN = False Negatif, dalam artian hasil deteksi menyatakan tidak plagiat, namun hasil aktual menyatakan plagiat
 TN = True Negatif, dalam artian hasil deteksi menyatakan tidak plagiat, dan hasil aktual juga menyatakan tidak plagiat

3. METODE PENELITIAN

Plagiarism Checker menggunakan metode Jaro Winkler ini dibuat dengan pendekatan terstruktur menggunakan metode waterfall. Modul ini ditambahkan pada sistem E-Read yang telah ada. Gambar 1 merupakan langkah-langkah penelitian yang dilakukan.



Gambar 1. Langkah penelitian

Terdapat 5 langkah penelitian yang dilakukan seperti yang terlihat pada Gambar 1, penelitian dimulai dengan melakukan studi literatur terutama terkait teori Jaro Winkler. Penelitian dilakukan dengan pengumpulan softcopy dokumen abstrak skripsi. Langkah ketiga yakni perancangan sistem menghasilkan diagram dan desain yang akan membantu dalam tahap implementasi sistem yakni berupa diagram conceptual data model (CDM), physical data model (PDM), data flow diagram (DFD), serta rancangan antarmuka dengan mockup. Setelah perancangan sistem selesai dibuat, maka langkah selanjutnya adalah implementasi sistem yang pada akhir langkah ini akan menghasilkan *Plagiarism Checker* pada sistem E-Read. Penelitian ditutup dengan pengujian akurasi sistem menggunakan Confusion Matrix yang terdiri dari perhitungan akurasi, presisi, recall, dan f-measures terhadap sistem *Plagiarism Checker* yang telah dibangun sebagai hasil penelitian ini.

3.1 Studi Literatur dan Analisis Kebutuhan

Tahapan ini dimulai dengan melakukan observasi dan pengamatan terkait kekhawatiran mahasiswa dan calon dosen pembimbing skripsi untuk mengetahui apakah topik skripsi yang akan diajukan pernah dibuat oleh peneliti sebelumnya, terutama peneliti dari kalangan internal Fakultas Ilmu Komputer UPN “Veteran” Jawa Timur. Berdasarkan hal tersebut maka diruskan kebutuhan-kebutuhan pengguna dalam melakukan pengecekan awal terhadap dokumen skripsi yang telah tersimpan di perpustakaan Fakultas Ilmu Komputer. Studi literatur kemudian dilakukan untuk memilih algoritma yang tepat untuk diterapkan pada sistem *Plagiarism Checker* yang dibuat serta pengumpulan teori terkait algoritma Jaro Winkler untuk dapat diimplementasikan serta teori pengujian sistem untuk dapat dilakukan pengujian terhadap hasil penelitian yang telah dilakukan.

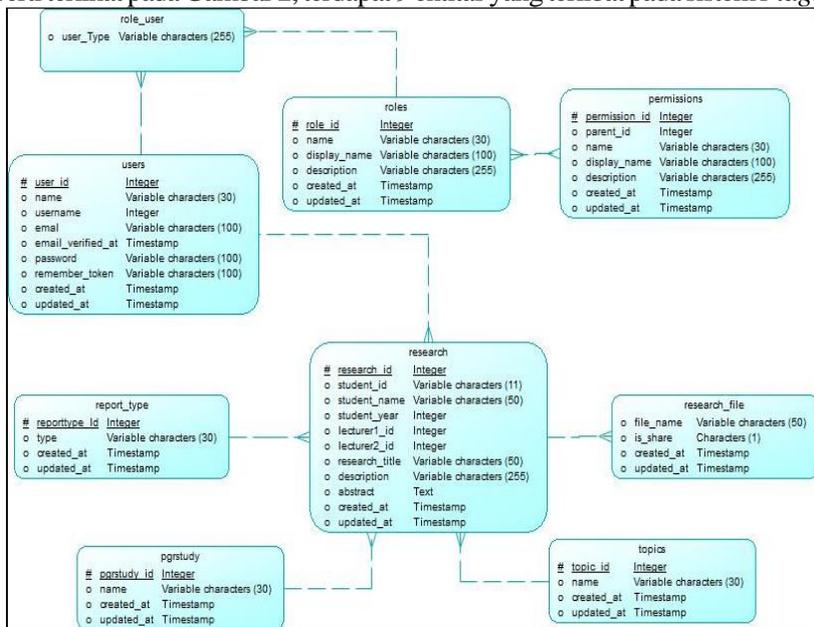
3.2 Pengumpulan Data

Data primer yang perlu dikumpulkan dalam melakukan penelitian ini adalah berupa *softcopy* abstrak skripsi yang akan disimpan pada basis data sistem *Plagiarism Checker*. Data abstrak skripsi didapatkan dari alumni yang telah mengumpulkan berkas skripsinya dalam bentuk CD di perpustakaan Fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur. Data ini yang akan menjadi dasar pengecekan apakah calon skripsi yang akan diajukan oleh mahasiswa terindikasi plagiat atau tidak. Data ini juga diperlukan untuk pengujian akurasi sistem untuk menentukan kelayakan penerapan algoritma dalam sistem *Plagiarism Checker* yang dibuat. Data abstrak skripsi sebelumnya telah tersimpan pada sistem E-Read, namun untuk pengujian sistem,

dilakukan penambahan koleksi agar proses komputasi perbandingan dokumen calon skripsi mahasiswa dapat dibandingkan dengan lebih banyak dokumen yang tersimpan pada basis data E-Read.

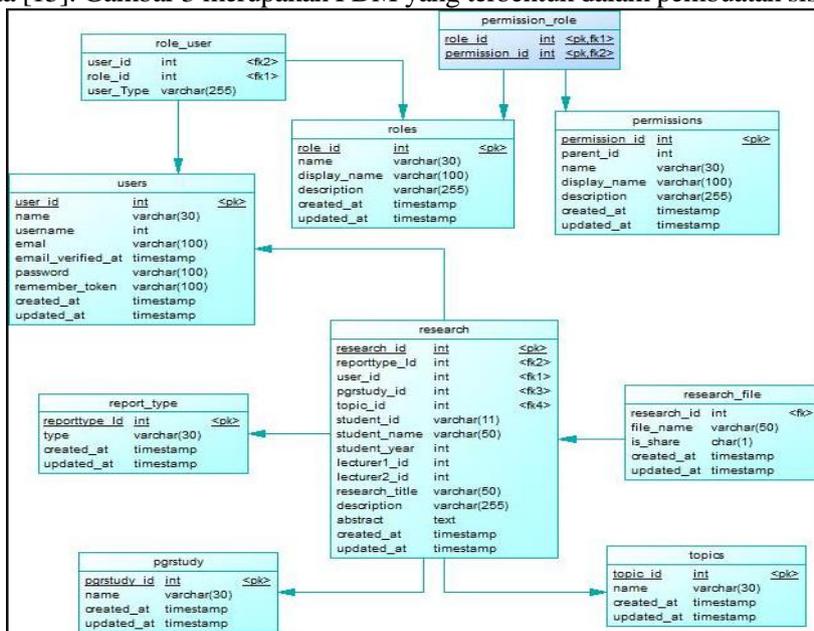
3.3 Perancangan Sistem

Sebelum sistem diimplementasikan dalam pembuatan perangkat lunak, maka terlebih dahulu dilakukan perancangan sistem untuk memudahkan dalam membaca alur kerja sistem sehingga nantinya tercipta hasil akhir sistem yang sesuai dengan harapan pengguna dan pengembang. Perancangan sistem pada penelitian ini menggunakan Conceptual Data Model (CDM), Physical Data Model (PDM), Data Flow Diagram (DFD). Conceptual Data Model yang digunakan pada penelitian ini dapat dilihat pada Gambar 2. Seperti terlihat pada Gambar 2, terdapat 9 entitas yang terlibat pada sistem *Plagiarism Checker*.



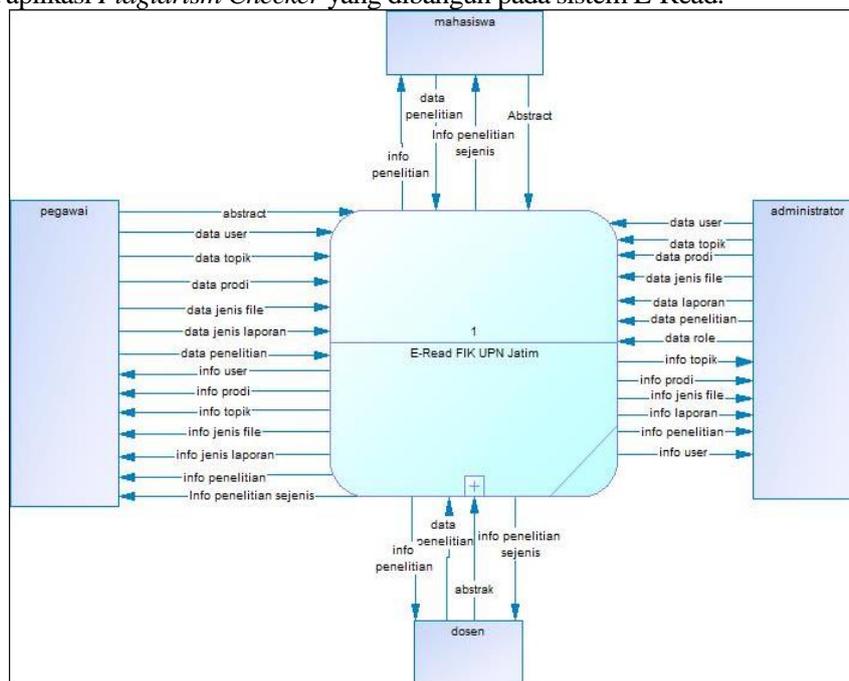
Gambar 2. Conceptual Data Model

Setelah CDM terbentuk, maka dapat dibentuk PDM yang nantinya akan diturunkan ke dalam relasi tabel sesuai kebutuhan sistem. PDM merepresentasikan struktur tabel dan kolom yang sebenarnya pada basis data [15]. Gambar 3 merupakan PDM yang terbentuk dalam pembuatan sistem.



Gambar 3. Physical Data Model

Pada pembentukan diagram PDM, Atribut kunci entitas yang memiliki hubungan 1 ke m akan diturunkan ke entitas yang memiliki relasi dengan derajat lebih tinggi (m). Sebagai contoh pada relasi antar atribut `report_type` ke atribut `research` yang memiliki relasi 1 to m, maka atribut `reporttype_id` pada entitas `report_type` akan dirunkan menjadi kunci tamu pada entitas `research`. Pada PDM juga terlihat ada penambahan entitas baru sebagai dampak dari relasi m to n pada entitas `roles` dan `permissions` yang berisi atribut kunci dari kedua entitas tersebut. Untuk menggambarkan keseluruhan proses yang berjalan pada sistem beserta storage yang terpakai, maka dibentuklah data flow diagram (DFD). DFD merupakan sebuah tools yang digunakan untuk menggambarkan hasil dari analisis kebutuhan dan perancangan spesifikasi sistem dalam bentuk gambar (*graphical view*) [16]. DFD yang dibentuk terdiri dari diagram konteks, DFD Level 0, dan DFD Level 1. Adapun diagram context yang digunakan pada penelitian ini dapat dilihat pada Gambar 4, dimana entitas dosen dan mahasiswa bisa mengakses informasi penelitian sejenis pada aplikasi *Plagiarism Checker* yang dibangun pada sistem E-Read.



Gambar 4. Context Diagram

3.4 Implementasi Sistem

Setelah proses perancangan sistem selesai dilakukan, maka langkah berikutnya adalah melakukan implementasi hasil rancangan yang telah dibuat tersebut kedalam barisan kode program. Implementasi dari sistem dilakukan menggunakan bahasa pemrograman PHP dan basis data MySQL. Bahasa pemrograman PHP digunakan karena memiliki banyak keunggulan, diantaranya bersifat free, memiliki kecepatan akses yang tinggi, dan didukung oleh banyak sistem basis data, salah satunya MySQL [17].

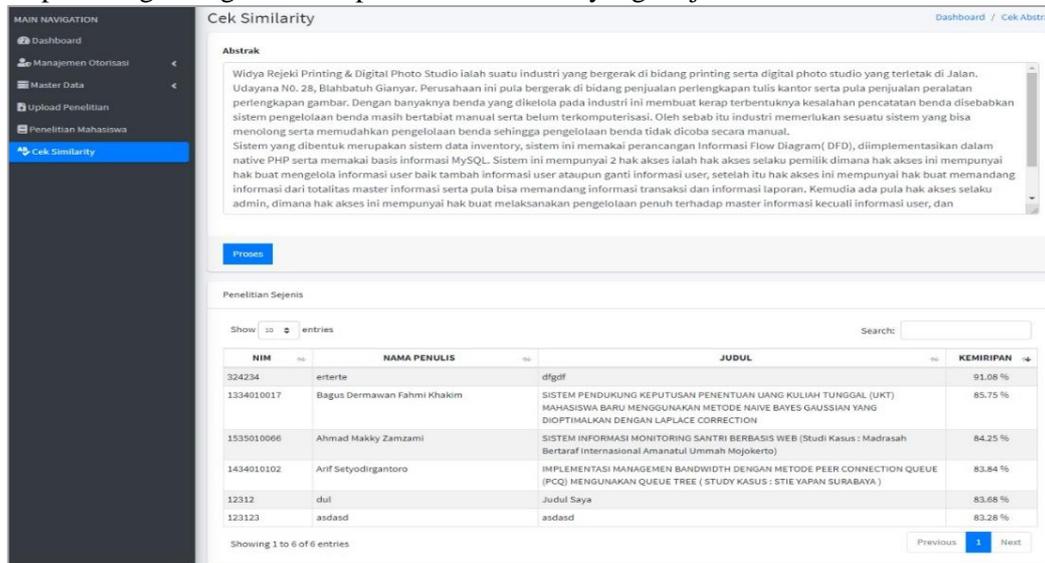
3.5 Pengujian Sistem

Langkah terakhir yang dilakukan pada penelitian ini adalah dengan melakukan pengujian sistem. Pengujian yang dilakukan ditekankan pada akurasi sistem dalam mendeteksi kemungkinan plagiasi antar dokumen abstrak yang tersimpan pada sistem E-Read dengan dokumen abstrak calon skripsi yang akan diajukan mahasiswa. Pengujian dilakukan menggunakan confusion matrix, sehingga nantinya dapat direkomendasikan kehandalan algoritma Jaro Winkler dalam melakukan deteksi kemiripan dokumen pada kasus abstrak skripsi.

4. PEMBAHASAN

Diagram-diagram yang telah dihasilkan pada tahap perancangan kemudian diimplementasikan pada basis data serta kode program untuk menghasilkan sebuah sistem deteksi plagiasi pada

sistem E-Read yang telah dibangun pada penelitian sebelumnya. Gambar 5 merupakan hasil pengecekan plagiasi yang dapat dilakukan oleh sistem menggunakan algoritma Jaro -Winkler dalam perhitungan tingkat kemiripan antar dokumen yang diuji.



Gambar 5. Hasil Pengecekan Kesamaan Dokumen

Seperti yang terlihat pada Gambar 5, proses pengecekan kemiripan dokumen dapat dilakukan dengan memasukkan teks abstrak yang akan diuji, maka setelah tombol Proses ditekan, sistem akan menampilkan dokumen-dokumen abstrak yang memiliki nilai kemiripan tertinggi dengan dokumen uji sesuai treshold yang ditentukan pengguna.

4.1 Pengujian Sistem

Untuk mengukur akurasi sistem, maka digunakan metode confusion matrix, dimana hasil similaritas sistem dibandingkan dengan pendapat ahli, yakni pendapat dosen di lingkungan program studi Informatika yang melakukan penilaian secara manual terhadap abstrak skripsi yang dibandingkan sesuai data pengujian yang tercantum pada Tabel 1. Peneliti melakukan pengujian terhadap 10 dokumen skripsi yang terdapat pada sistem E-Read dengan treshold sebesar 0,7. Hasil pengujian tingkat kemiripan dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian Kemiripan Dokumen

No	Dokumen Uji	Dokumen Hasil Similaritas Sistem	Nilai	Sistem	Penguji (Human)
1	16340100xx	14340101xx	70,97	Mirip	Tidak
2	16340100xx	14340101xx	73,96	Mirip	Mirip
3	15340100xx	15350100xx	70,97	Mirip	Tidak
4	15340101xx	16340100xx	73,52	Mirip	Mirip
5	16340100xx	15340100xx	72,41	Mirip	Tidak
6	16340100xx	16340100xx	85,09	Mirip	Mirip
7	16340100xx	15340101xx	85,34	Mirip	Mirip
8	16340100xx	15350100xx	52,62	Tidak	Tidak
9	170810100xx	16340100xx	76,01	Mirip	Mirip
10	170810101xx	14340101xx	76,55	Mirip	Mirip

Berdasarkan perbandingan hasil similaritas sistem dengan pendapat ahli (human) maka didapatkan hasil pengujian sebagai berikut:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% = \frac{6+1}{10} \times 100\% = 70\%$$

$$Presisi = \frac{TP}{TP+FP} \times 100\% = \frac{6}{6+2} \times 100\% = 75\%$$

$$Recall = \frac{TP}{TP+FN} \times 100\% = \frac{6}{6} \times 100\% = 100\%$$

TP = True Positif, dalam artian hasil deteksi menyatakan plagiat, dan hasil aktual juga menyatakan plagiat

FP = False Positif, dalam artian hasil deteksi menyatakan plagiat, dan hasil aktual menyatakan tidak plagiat

FN = False Negatif, dalam artian hasil deteksi menyatakan tidak plagiat, namun hasil aktual menyatakan plagiat

TN = True Negatif, dalam artian hasil deteksi menyatakan tidak plagiat, dan hasil aktual juga menyatakan tidak plagiat

4.2 Analisis Hasil Similaritas dengan Jaro Winkler

Selain pengujian menggunakan threshold sebesar 0,7 seperti hasil pada Tabel 1, pengujian juga dilakukan terhadap 10 dokumen yang dibandingkan dengan 80 dokumen abstrak mahasiswa yang tersimpan pada basis data E-Read dengan hasil bahwa 9 dokumen abstrak yang memiliki nilai similaritas tertinggi lebih dari 0,7. Hal ini berarti algoritma Jaro Winkler relatif menghasilkan nilai similaritas yang tinggi. Berdasarkan hasil analisis terhadap proses komputasi algoritma Jaro Winkler, didapatkan kesimpulan bahwa algoritma ini bekerja dengan membandingkan satu persatu karakter dari index 1 sampai n dengan karakter pada dokumen uji sesuai *range* uji yang ditentukan seperti ilustrasi yang tampak pada Tabel 2.

Index 1	Index 2	Index 3	Index 4	Index 5	Index 6	Index 7	Index 8
L	A	K	S	H	I	T	A
L	A	K	S	I	T	H	A

Pertama-tama, akan dicari *range* uji sebagai patokan untuk mencari kesamaan karakter yang diuji sebagai berikut :

$$\text{Perhitungan Range Uji} = (\text{Banyaknya Jumlah Huruf/Kata} / 2) - 1 = (8/2) - 1 = 3$$

Berdasarkan perhitungan *range* uji, maka akan dicari kesamaan karakter pada 3 huruf dibelakang dan 3 huruf didepannya. Sebagai contoh, misalkan yang dicek adalah karakter L (karakter pertama) pada string 1, maka komputasi dilakukan dengan mencari 3 huruf di kiri karakter pertama (L) dan 3 huruf di kanan karakter pertama (L) pada string 2. Jika ditemukan huruf yang sama dalam *range* tersebut, maka dinyatakan ditemukan kesamaan huruf dan secara komputasi nilai *m* yang akan digunakan pada persamaan (1) akan bertambah, sebagai akibatnya maka nilai similaritas akan cenderung meningkat. Dengan kata lain dapat disimpulkan bahwa semakin tinggi *range* uji yang digunakan, maka semakin tinggi pula kemungkinan ditemukannya huruf yang sama yang berakibat pada cenderung meningkatnya nilai similaritas yang dihasilkan sistem.

5. KESIMPULAN

Berdasarkan seluruh rangkaian penelitian yang telah dilakukan maka dapat disimpulkan bahwa telah berhasil dibangun sebuah *Plagiarism Checker* yang ditanamkan pada sistem manajemen perpustakaan (E-Read). Pengujian akurasi yang dilakukan menunjukkan nilai 40% jika dibandingkan dengan pendapat ahli (human), nilai presisi 33,3%, recall 100%, dan F-Measure sebesar 35%. Berdasarkan hasil analisis terhadap proses komputasi algoritma Jaro Winkler maka didapatkan hasil bahwa algoritma ini berfokus pada pencarian kesamaan karakter sesuai *range* uji yang ditentukan, hal ini berarti bahwa algoritma ini belum memiliki mekanisme pengecekan makna kata yang seharusnya diperlukan dalam sistem pengecekan plagiasi dokumen.

DAFTAR PUSTAKA

- [1] M. Zalnur, "Plagiarisme di kalangan mahasiswa dalam membuat tugas-tugas perkuliahan

- pada Fakultas Tarbiyah IAIN Imam Bonjol Padang,” *J. Al-Ta’lim*, vol. 1, no. 1, 2012.
- [2] M. A. Shadiqi, “Memahami dan Mencegah Perilaku Plagiarisme dalam Menulis Karya Ilmiah,” *Bul. Psikol.*, vol. 27, no. 1, 2019.
 - [3] M. H. P. Swari, C. A. Putra, and I. P. S. Handika, “Sistem Manajemen Data Skripsi (Studi Kasus: Perpustakaan Fakultas Ilmu Komputer Upn” Veteran” Jawa Timur),” *J. Pendidik. Teknol. dan Kejuru.*, vol. 17, no. 2, pp. 198–209, 2020.
 - [4] S. Bahri, “Perbandingan Algoritma Jaro-Winkler Distance Dengan Algoritma Rabin-Karp Dalam Mendeteksi Kasus Plagiarism Pada Dokumen,” Universitas Maritim Raja Ali Haji, 2017.
 - [5] S. Christina, E. D. Oktaviyani, and B. Famungkas, “Mendeteksi Plagiarism Pada Dokumen Proposal Skripsi Menggunakan Algoritma Jaro Winkler Distance,” *Saintekom*, vol. 8, no. 2, 2018.
 - [6] M. J. Tangga, S. Rahman, and Hasniati, “Analisis Perbandingan Algoritma Levenshtein Distance Dan Jaro Winkler Untuk Aplikasi Deteksi Plagiarisme Dokumen Teks,” *J. Technol. Res. Inf. Syst. Eng.*, vol. 4, no. 1, 2017.
 - [7] Y. Rochmawati and R. Kusumaningrum, “Studi Perbandingan Algoritma Pencarian String dalam Metode Approximate String Matching untuk Identifikasi Kesalahan Pengetikan Teks,” *Buana Inform.*, vol. 7, no. 2, 2016.
 - [8] I. Zarembo, A. Teilansa, A. Rausisb, and J. Bulsb, “Assessment of Name Based Algorithms for Land Administration Ontology Matching,” in *ICTE in Regional Development*, 2015, pp. 53–61.
 - [9] T. Peng, L. Li, and J. Kennedy, “A comparison of techniques for name matching,” *GSTF J. Comput.*, vol. 2, no. 1, 2012.
 - [10] Menteri Pendidikan Nasional, “Peraturan Menteri Pendidikan Nasional No. 17 Tahun 2010 Tentang Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi,” Jakarta, 2010.
 - [11] A. Wibowo, “Mencegah dan Menanggulangi Plagiarisme di Dunia Pendidikan,” *Kesmas Natl. Public Heal. J.*, vol. 6, no. 5, pp. 195–200, 2012.
 - [12] A. Kornain, F. Yansen, and T. Tinaliah, “Penerapan Algoritma Jaro-Winkler Distance untuk Sistem Pendeteksi Plagiarisme pada Dokumen Teks Berbahasa Indonesia,” *J. Ilmu Komput. dan Teknol. Inf.*, pp. 62–70, 2014.
 - [13] T. Willyanto, “Sistem Deteksi Plagiarisme Dokumen Teks Menggunakan Algoritma Jaro-Winkler Distance,” Universitas Nusantara PGRI Kediri, 2015.
 - [14] Ž. Vujovi, “A case study of the application of WEKA software to solve the problem of liver in amation,” pp. 1–18.
 - [15] M. West, *Developing High Quality Data Models*. Massachusetts: Morgan Kaufmann Publisher, 2011.
 - [16] R. Ibrahim and S. Y. Yen, “Formalization of the Data Flow Diagram Rules for Consistency Check,” *Int. J. Softw. Eng. Appl.*, vol. 1, no. 4, pp. 95–111, 2010, doi: 10.5121/ijsea.2010.1406.
 - [17] K. S. Haryana, “Pengembangan Perangkat Lunak dengan Menggunakan PHP,” *J. Comput. Bisnis*, vol. 2, no. 1, pp. 14–21, 2015.

Biodata Penulis

Made Hanindia Prami Swari, lahir di Denpasar, 5 Februari 1989 menempuh pendidikan S1 di Program Studi Teknik Informatika Fakultas MIPA Universitas Udayana di Denpasar, Bali pada tahun 2011, dan menamatkan pendidikan Magister Ilmu Komputer di UGM Yogyakarta pada tahun 2014. Saat ini penulis tercatat sebagai dosen di Program Studi Informatika, Fakultas Ilmu Komputer UPN “Veteran” Jawa Timur

Chrystia Aji Putra, lahir di Bogor, 8 Oktober 1986, menempuh program S1 di Prodi Teknik Informatika UPN "Veteran" Jawa Timur pada tahun 2008, dan menamatkan pendidikan Magister Teknik di Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2015. Saat ini penulis tercatat sebagai dosen di Program Studi Informatika, Fakultas Ilmu Komputer UPN “Veteran” Jawa Timur

I Putu Susila Handika, lahir di Tabanan, 5 Juli 1990. Lulusan Sarjana Program Studi Ilmu Komputer Fakultas MIPA Universitas Brawijaya di Malang pada tahun 2012, dan menamatkan pendidikan Magister Sistem Informasi dan Komputer Fakultas Teknik Universitas Udayana pada tahun 2016. Pekerjaan penulis saat ini adalah sebagai Dosen di Program Studi Teknik Informatika STMIK STIKOM Indonesia.