Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

SISTEM DETEKSI PLAGIASI MENGGUNAKAN ALGORITME FREQUENCY BASED HASHING-S PADA FILE PDF

Thamrin Auliya[1], Cosmas Eko Suharyanto[2]

[1]Mahasiswa Program Studi Teknik Informatika, Universitas Putera Batam
[2]Dosen Program Studi Teknik Informatika, Universitas Putera Batam
email:pb180210117@upbatam.ac.id

ABSTRACT

In today's digital computing era, there has been a very high growth of digital data production. Quoted from the forbes.com page, there are 2.5 quintillion bytes of data created every day. One of the digital data is student assignments collected through Google Classroom media. From the student data collected, many assignments indicated plagiarism with other student assignments. If the analysis is done manually, it will take a long time and be very tiring. For efficient use of time and resources, a screening process is needed that has the ability to calculate the plagiarism level of each student's assignment. The Approximate Matching method is the method most often used to find similarities between the data being compared by setting a similarity score. In this study, the Approximate Matching algorithm used is the Frequency Based Hashing-S algorithm. The advantages of this algorithm are that it is safe against active attacks and has a 98% accuracy rate for compressed data formats. This application is useful for lecturers who will check a lot of student assignments. With this application, lecturers only need to check student assignments that are not plagiarized, thereby reducing the number of student assignments that will be checked manually.

Keywords: Frequency Based Hahsing-S, Plagiarism System, Data Filtering System

PENDAHULUAN

Di era digital sekarang ini, telah terjadi lonjakan produksi data digital yang sangat tinggi. Berdasarkan data dari laman forbes.com dalam survei pada tahun 2018, terdapat 2.5 quintillion bytes data yang diproduksi setiap hari dan produksi data digital tersebut akan terus bertambah setiap tahun (Marr, 2018). Dizaman pandemi COVID-19 sekarang ini, banyak kampus yang menyelenggarakan proses belajar mengajar online atau daring menggunakan platform Google Classroom. Tugas mahasiswa berformat PDF akan dikumpulkan pada platform Google Classroom. Namun setelah tugas mahasiswa dikumpulkan dan diperiksa oleh dosen, ternyata banyak tugas yang plagiasi. Dimana, jika tugas mahasiswa tersebut diperiksa secara manual satu persatu maka akan membutuhkan waktu yang lama dan melelahkan. Oleh sebab itu dibutuhkan sebuah sistem yang mampu menyaring tugas mahasiswa yang plagiasi dan tidak plagiasi. Sehingga dosen hanya perlu memeriksa tugas siswa yang plagiasi saja.

Teknik yang biasanya digunakan untuk proses penyaringan adalah teknik Approximate Matching. Approximate Matching merupakan sebuah teknik yang digunakan untuk menemukan kesamaan di antara beberapa file berdasarkan skor kesamaan. Beberapa algoritme yang biasa diguanakan adalah Sd-Hash dan Ssdeep. Namun algoitme ini telah terbukti rentan terhadap serangan aktif (Chang,

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

et al., 2015) (Roussev, 2011). Sehingga dibutuhkan algoritme baru yang aman terhadap serangan aktif dan memiliki akurasi yang tinggi untuk menghitung nilai kesamaan diantara beberapa *file*. Algoritme Frequency Based Hashing-S atau FBHash merupakan algortime baru yang dipublikasikan pada tahun 2019. Pada algoritme FBHash, setiap *byte* dari *chunk* berkontribusi pada skor akhir dan pengaruhnya terhadap skor akhir tergantung pada pentingnya suatu dokumen (Chang, et al., 2019). Untuk membuat skor kesamaan benar benar rendah atau mendekati nol hampir setiap *chunk* harus di modifikasi. Secara keamanan, algoritme FbHash lebih baik dari algoritme SdHash dan Ssdeep, oleh sebab itu penulis memilih untuk menggunakan algoritme FbHash ini. Algoritme FbHash dibagi menjadi dua versi, yaitu versi FbHash-B untuk mengukur kesamaan ditingkat *byte* seperti format text dan FbHash-S untuk mengukur *syntactic matching* dan menggunakan informasi internal dokumen untuk mengukur kesamaan. Algoritme ini digunakan untuk format *file* terkompresi seperti PDF. Karena object penelitian dari penulis adalah *file* yang terkompresi dalam format PDF, maka penulis menggunakan algoritme FbHash-S.

KAJIAN TEORI

2.1 Dasar Teori

2.2.1 Kriptograpi

Kriptografi adalah teknik untuk mencapai kerahasiaan pesan (Muhammed & Varol, 2019). Kriptografi secara bahasa memiliki arti tulisan rahasia, diambil dari bahasa yunani yaitu cryptós (rahasia) dan gráphein (tulisan). Proses melakukan penulisan pesan rahasia dan memecahkan pesan rahasia dinamakan kriptologi yang diperlakukan sebagai bidang studi yang terpisah. Kriptografi memiliki banyak kegunaan seperti enkripsi, dekripsi, otentikasi, tanda tangan digital, hashing, dan sebagainya (Rao, et al., 2019).

2.2.2 Hasing

Hashing adalah teknik yang menerima pesan panjang variabel sebagai Input dan menghasilkan panjang tetap dan *string* yang unik (Rao, et al., 2019). Pada mekanisme *hashing*, fungsi *hash* digunakan untuk menghasilkan nilai *hash*. Nilai *hash* akhir pada skema *hashing* disebut *digest.* Pada algoritma Frequency Based Hashing-S, nilai *digest* akhir digunakan untuk menghitung nilai Cosine Similarity yang merupakan nilai tingkat kesamaan/plagiasi.

2.2.3 Approximate Matching

Approximate Matching merupakan salah satu algoritma *integrity* yang digunakan untuk mengurangi jumlah data yang harus diperiksa oleh investigator secara manual dengan menemukan kesamaan (Lillis, et al., 2018). Teknik Approximate Matching dapat dikategorikan ke dalam tiga kategori berikut: Bytewise Matching, Syntactic Matching dan Semantic Matching.

2.2.4 Optical Character Recognatition

Optical Character Recognition (OCR) adalah proses yang memungkinkan sistem tanpa campur tangan manusia mengidentifikasi skrip atau abjad yang ditulis ke dalam komunikasi verbal pengguna (Sahu & Sonkusare, 2017). Optical Character Recognition atau OCR digunakan untuk mengekstract teks yang

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

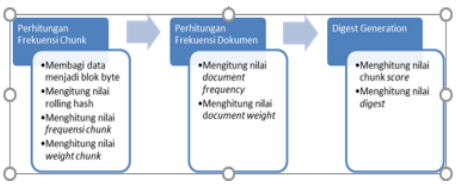
berada dalam *file* PDF. Karena Bahasa yang digunakan adalah Python, maka untuk proses OCR dilakukan menggunakan *library* PyPDF2. Kekurangan utama menggunakan *library* ini adalah skema pengkodean. Dokumen PDF menggunakan penyandian termasuk UTF-8, ASCII, Unicode, dll. Jadi, mengonversi PDF ke teks dapat mengakibatkan hilangnya data karena skema penyandian tersebut.

2.2.5 Cosine Similarity

Cosine similarity merupakan salah satu metode yang dapat digunakan untuk menghitung tingkat kesamaan antara 2 buah objek yang mempunyai kemiripan kata, dihitung menggunakan membandingkan nilai vektornya (Resta, et al., 2021). Dalam penelitian ini, cosine similarity digunakan untuk menghitung nilai similarity dengan membandingkan digest vector D1 (digest 1) dan digest ve

2.2.6 Frequency Based Hashing

Algoritme Frequency Based Hashing atau FbHash mengadopsi Term Frequency Inverse Document Frequency (TF-IDF) untuk menemukan dokumen yang memiliki kesamaan. Cara kerja algoritme ini dibagi menjadi 3 langkah yang ditunjukkan pada Gambar 2.1 (Chang, et al., 2019).

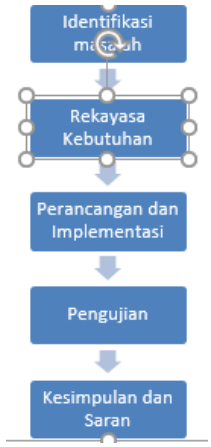


Gambar 1.Tahapan Algoritme Frequency Based Hashing-S

METODE PENELITIAN

3.1 Desain Penelitian

Tahapan yang digunakan dalam penelitian sistem deteksi plagiasi menggunakan algoritme Frequency Based Hashing-S pada file PDF. Tahapan-tahapannya dapat dilihat pada Gambar 3.1.



Gambar 1.Diagram Alir Penelitian

3.2 Perancangan Sistem

Perancangan sistem merupakan sebuah cara yang dilakukan untuk menggambarkan langkah langkah strategis yang diambil dengan memberikan gambaran umum sistem yang dibangun demi tercapainya tujuan penelitian.. Perancangan sistem dibagi menjadi tiga tahap yaitu perancangan data dan UML, perancangan database dan perancangan antar muka.

3.3 Perancangan UML

3.3.1 Diagram Alir OCR pyPDF2

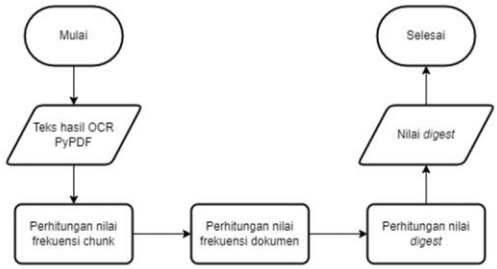
OCR PyPDF2 merupakan OCR yang paling umum dan paling sering digunakan

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

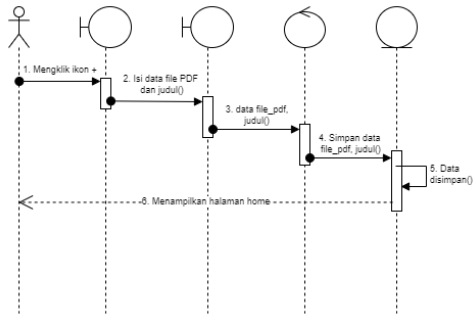
ISSN (Online) 2715-6265

sekarang ini untuk mengekstract teks dari *file* pdf. Berikut diagram alir OCR PyPDF2 pengumpulan data dan analisis dari hasil pengolahan data.



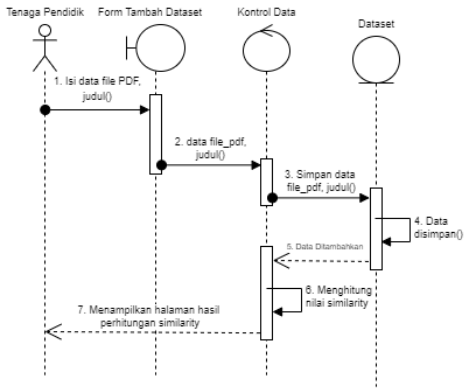
Gambar 2.Diagram Alir OCR PyPDF2

3.3.2 Sequence Diagram Unggah Dataset



Gambar 3. Sequence Diagram Unggah Dataset

3.3.3 Sequence Diagram Perhitungan Nilai Similarity



3.4 Perancangan Database

Struktur data dari database dalam penelitian ini dibagi menjadi tiga, yaitu kosin_proses_dataset, kosin_proses_datatest dan juga kosin_proses_tf. Struktur data kosin_proses_dataset dapat dilihat pada tabel 3.3, Struktur data kosin_proses_datatest dapat dilihat pada tabel 3.4 dan struktur data kosin_proses_tf dapat dilihat pada tabel 3.4.

Tabel 3. 3 Tabel Struktur kosin_proses_dataset

No	Tipe Data	Nama Variabel
1	int(11)	id
2	longtext	dataset_judul
3	Varchar(100)	dataset_flie

Tabel 3. 4 Tabel Struktur kosin_proses_datatest

No	Tipe Data	Nama Variabel
1	int(11)	id
2	longtext	datates_judul
3	Varchar(100)	datates_flie

Tabel 3. 5 Tabel Struktur kosin_proses_tf

No	Tipe Data	Nama Variabel
1	Int(11)	Id
2	Varchar(225)	kata
3	Int(11)	frequency
4	Int(11)	dataset_id
5	longtext	dataset_judul

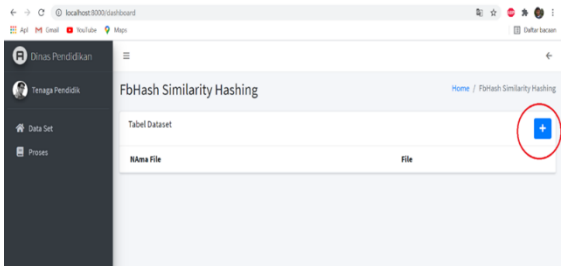
Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

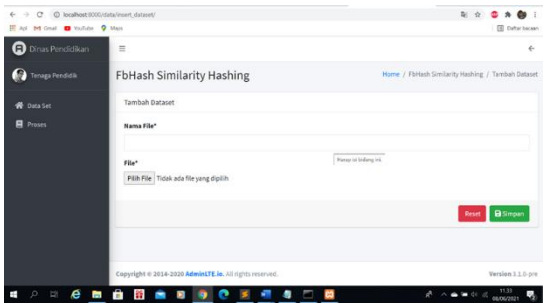
3.5 Perancangan Antarmuka

3.5.1 Tampilan Home



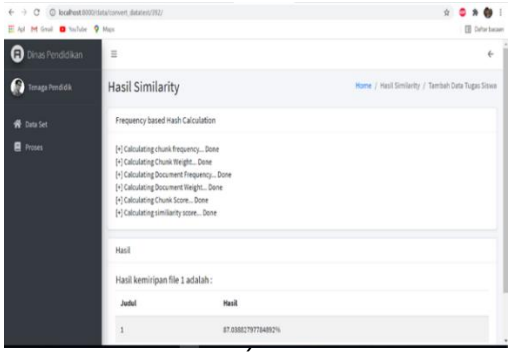
Gambar 1. Tampilan Home

3.5.2 Tampilan Tambah Dataset



Gambar 2. Tampilan Tambah Dataset

3.5.2 Tampilan Hasil Perhitungan NIlai Similarity



HASIL DAN PEMBAHASAN

4.1 Hasil

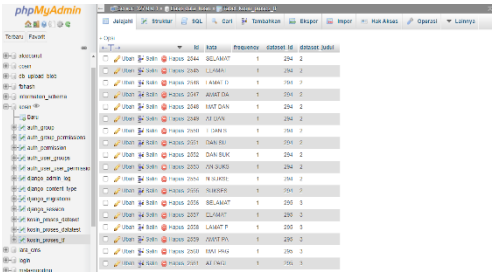
4.1.1 Web Service Dapat Melakukan Proses Tambah Dataset

Pengujian ini bertujuan untuk mengetahui apakah web service dapat menambahkan dataset ke dalam tabel kosin_proses_dataset dengan melakukan proses unggah dataset dan mampu melakukan update data chunk dari setiap data yang diunggah ke dalam tabel kosin_proses_tf.

Hasil dari pengujian web service dapat melakukan proses penambahan dataset bisa dilihat pada Gambar 4.1.

Gambar 4. 1 Tampilan tabel kosin_proses_dataset

Hasil dari pengujian aplikasi web service dapat melakukan update data chunk bisa dilihat pada Gambar 4.2.



Gambar 4. 2 Tampilan tabel kosin_proses_tf

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

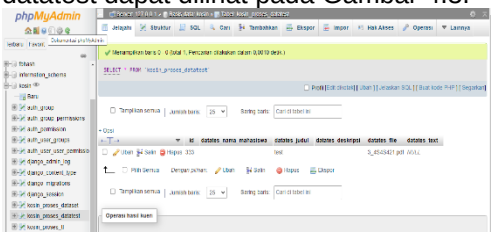
Jurnal Comasie

ISSN (Online) 2715-6265

4.1.2 Web Service Dapat Melakukan Proses Tambah Datatest

Pengujian ini bertujuan untuk mengetahui apakah web service dapat menambahkan datatest ke dalam tabel kosin_proses_datatest dengan melakukan proses perhitungan nilai kesamaan

Hasil dari pengujian aplikasi web service dapat melakukan proses penambahan datatest dapat dilihat pada Gambar 4.3.

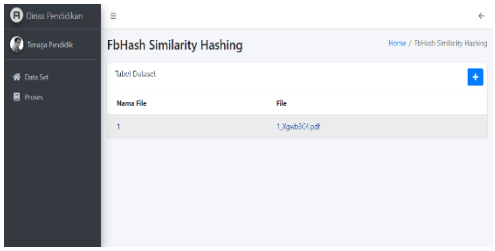


Gambar 4. 3 Tampilan tabel kosin_proses_datatest

4.1.3 Web Service Dapat Menampilkan Dataset

Pengujian ini bertujuan untuk mengetahui apakah web service dapat menampilkan dataset pada halaman /dashboard/ sesuai dengan data yang ada pada tabel kosin_proses_dataset.

Hasil dari pengujian aplikasi web dapat menampilkan dataset dapat dilihat pada Gambar 4.4.

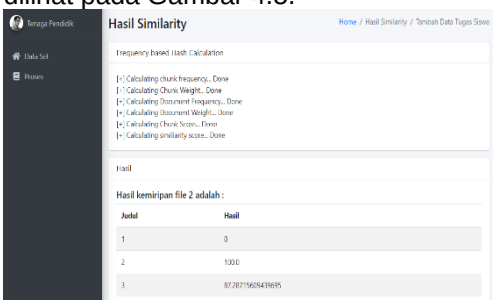


Gambar 4. 4 Tampilan halaman dashboard

4.1.4 Web Service Dapat Menampilkan Nilai Skor Kesamaan

Pengujian ini bertujuan untuk mengetahui apakah aplikasi web dapat menampilkan nilai skor kesamaan pada halaman /convert_datatest/[id]/ sesuai dengan hasil proses perhitungan nilai kesamaan.

Hasil dari pengujian aplikasi web dapat menampilkan nilai skor kesamaan dapat dilihat pada Gambar 4.5.



Gambar 4. 5 Tampilan halaman form hasil perhitungan similarity

4.1.5 Pengujian Kinerja Sistem

Pengujian kinerja sistem dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan oleh aplikasi web untuk menjalankan sistem, menjalankan proses unggah dataset, menjalankan proses unggah datatest, dan menjalankan proses perhitungan skor kesamaan. Tabel 1, menunjukkan hasil pengujian kinerja sistem

Tabel 1. Hasil Pengujian Kinerja Sistem

Fungsi	Hasil
Waktu yang dibutuhkan untuk menjalakan sistem	• 0.358424441019694 Detik
Waktu yang dibutuhkan untuk melakukan proses unggah dataset	• 21 KB (15 chunk): 4.115970675150553 • 7 KB (16 chunk): 4.184040228525798 • 7 KB (24 chunk): 4.644028798739115

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

	• 7 KB (21 chunk): 4.522922380765279 • 7 KB (25 chunk): 4.824703296025594 • 7 KB (25 chunk): 4.901762747764588 • 7 KB (21 chunk): 4.2346729437510175 Rata – rata : 4.48972872438885
Waktu yang dibutuhkan untuk melakukan proses unggah datatest	• 21 KB (15 chunk): 2.773295593261719 • 7 KB (16 chunk): 2.813503400484721 • 7 KB (24 chunk): 3.0274290800094605 • 7 KB (21 chunk): 2.9403717517852783 • 7 KB (25 chunk): 3.0519304434458414 • 7 KB (25 chunk): 3.1677523295084637 • 7 KB (21 chunk): 2.953723088900248 Rata – rata: 2.9611436696279614
Waktu yang dibutuhkan untuk melakukan proses perhitungan skor kesamaan	• 1 Data: 3.0162589152654014 • 2 Data: 3.252214272816976 • 3 Data: 3.2970595757166543 • 4 Data: 3.3498037815093995 • 5 Data: 3.4013511419296263 • 6 Data: 3.4487096071243286 • 7 Data: 3.4859561363856

4.1.6 Pengujian Akurasi Pehitungan Similarity

Pengujian akurasi perhitungan similarity dilakukan untuk mengetahui akurasi dari perhitungan similarity yang telah dilakukan. Pengujian ini dilakukan menggunakan metode confusion matrix. Proses perhitungan confusion matrik ditunjukkan pada Tabel 2.

Tabel 2. Proses *Confusion Matriks*

Judul	GS(F1,F2)		AM(F1,F2)			Keterangan
	Nilai	Perhitungan	Nilai	t	Perhitungan	
P2k.unkris.ac.id.pdf	2	GS ≥ 1	89	16	AM ≥ t	TP
Kompas.com.pdf	4	GS ≥ 1	88	16	AM ≥ t	TP
Kepri.bpk.go.id.pdf	2	GS ≥ 1	90	16	AM ≥ t	TP
Jdih.batam.go.id.pdf	5	GS ≥ 1	89	16	AM ≥ t	TP
Bpbatam.go.id.pdf	0	GS < 1	91	16	AM ≥ t	FP
Batam.terkini.id.pdf	100	GS ≥ 1	100	16	AM ≥ t	TP
Batam.suara.com.pdf	100	GS ≥ 1	99	16	AM ≥ t	TP

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

Sehingga, hasil perhitunga confusion matrik ditunjukkan pada Tabel 3.

Tabel 3. *Confusion Matriks*

	Aktual plagiasi	Aktual tidak plagiasi
Di prediksi plagiasi	TP = 6	FP = 1
Di prediksi tidak plagiasi	FN = 0	TN = 0

4.1.7 Pengujian Integritas Data

Pengujian ini dilakukan dengan mengimplementasikan metode collision attack yang bertujuan untuk mengetahui apakah dua buah chunk dapat memiliki nilai Rolling hash yang sama. Pengujian ini dikatakan berhasil jika tidak ada rolling hash yang sama dari dua chunk yang berbeda.

4.2 Pembahasan

4.2.1 Web Service Dapat Melakukan Proses Tambah Dataset

Setelah program web service dijalankan dan pengujian dilakukan, diperoleh hasil sistem dapat melakukan proses penambahan dataset.

4.2.2 Web Service Dapat Melakukan Proses Tambah Datatest

Setelah program web service dijalankan dan pengujian dilakukan, diperoleh hasil sistem dapat melakukan proses penambahan datatest

4.2.3 Web Service Dapat Menampilkan Dataset

Setelah program web service dijalankan dan pengujian dilakukan, diperoleh hasil sistem dapat menampilkan dataset.

4.2.4 Web Service Dapat Menampilkan Nilai Skor Kesamaan

Setelah program web service dijalankan dan pengujian dilakukan, diperoleh hasil sistem dapat menampilkan nilai similarity.

4.2.5 Pengujian Kinerja Sistem

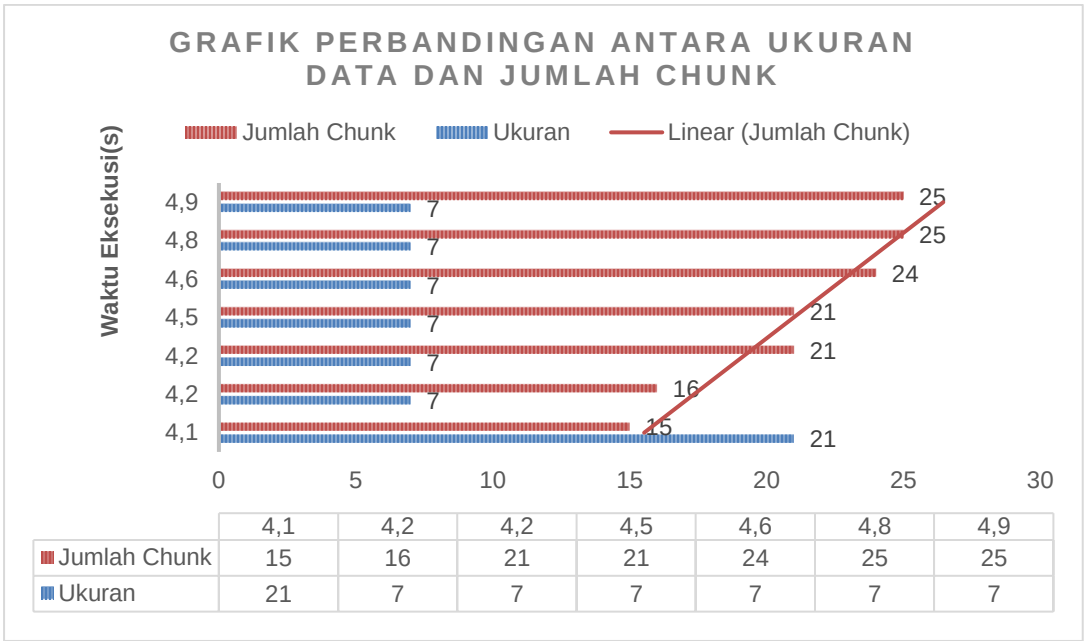
Grafik berikut menggambarkan perbandingan rata rata kecepatan waktu eksekusi program pada proses unggah dataset berdasarkan ukuran datanya dan banyak chunknya.

Tabel 4. Grafik perbandingan rata rata waktu eksekusi program berdasarkan ukuran dan jumlah chunk pada ungggah dataset

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

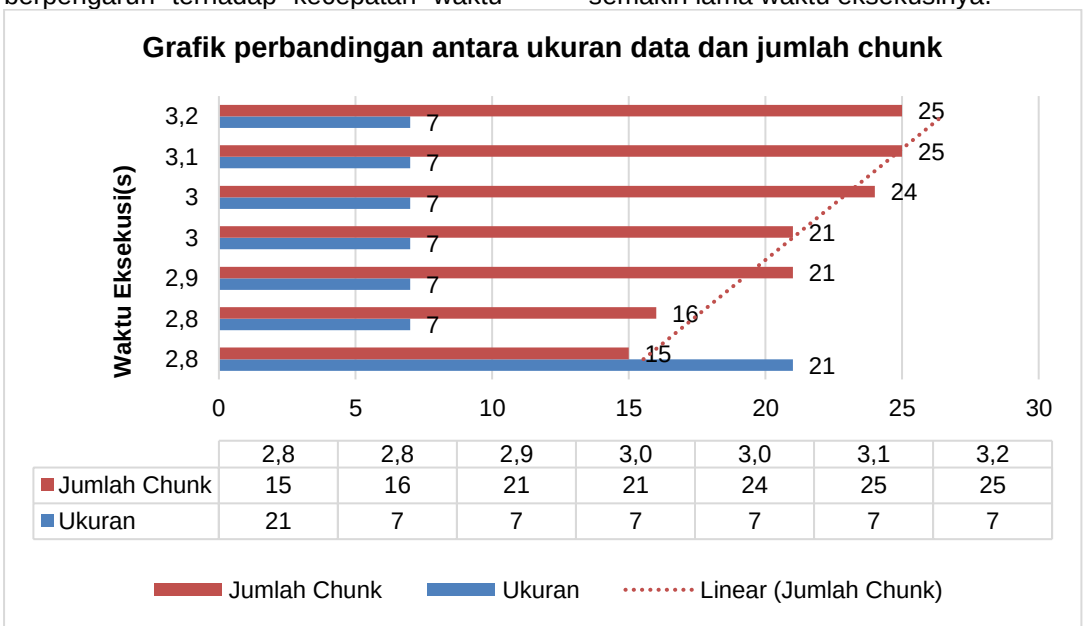
Jurnal Comasie

ISSN (Online) 2715-6265



	4,1	4,2	4,2	4,5	4,6	4,8	4,9
Jumlah Chunk	15	16	21	21	24	25	25
Ukuran	21	7	7	7	7	7	7

Dari grafik diatas dapat diambil kesimpulan bahwa ukuran data tidak berpengaruh terhadap kecepatan waktu eksekusi, sedangkan untuk jumah chunk, semakin banyak jumlah chunknya maka semakin lama waktu eksekusinya.



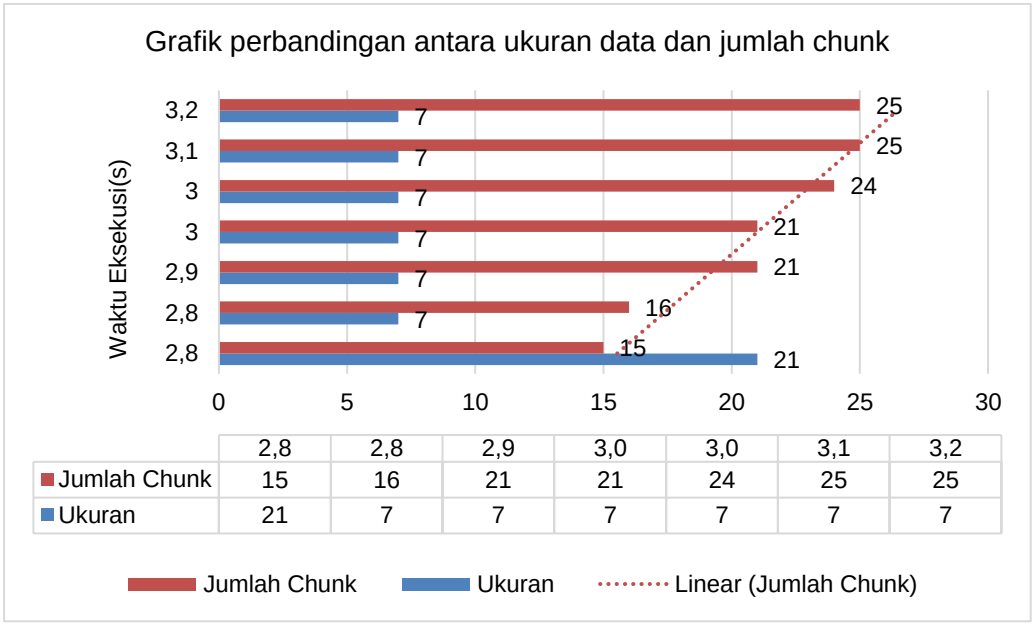
	2,8	2,8	2,9	3,0	3,0	3,1	3,2
Jumlah Chunk	15	16	21	21	24	25	25
Ukuran	21	7	7	7	7	7	7

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

Dari grafik diatas dapat diambil kesimpulan bahwa ukuran data tidak berpengaruh terhadap kecepatan waktu eksekusi, sedangkan untuk jumah chunk, semakin banyak jumlah chunknya maka semakin lama waktu eksekusinya. Perbedaan waktu proses unggah dataset dan proses unggah datatest terletak pada jumlah perintahnya. Dimana untuk proses unggah dataset terdapat 2 perintah, yaitu mengunggah data ke tabel kosin_proses_dataset dab menggunggah chunk ke tabel kosin_proses_tf. Sementara untuk proses unggah datatest hanya terdapat 1, yaitu menguggah data ke tabel kosin_proses_datatest. Sehingga rata rata waktu proses unggah datatest lebih cepat dari proses unggah dataset. Grafik berikut menggambarkan rata rata kecepatan waktu eksekusi program pada proses perhitungan nilai skor kesamaan berdasarkan jumlah data dataset.



	2,8	2,8	2,9	3,0	3,0	3,1	3,2
Jumlah Chunk	15	16	21	21	24	25	25
Ukuran	21	7	7	7	7	7	7

Dari grafik diatas dapat disimpulkan bahwa semakin banyak jumlah dataset maka waktu proses perhitungan nilai similarity akan semakin lama. Begitu juga sebaliknya

4.2.6 Pengujian Akurasi Pehitungan Similarity

Setelah program dijalankan dan pengujian dilakukan sesuai dengan skenario, diperoleh akurasi perhitungan nilai similarity. Tabel 7. menunjukkan hasil pengujian akurasi perhitungan nilai similarity.

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

Tabel 7. Hasil Pengujian Akurasi Perhitungan Nilai Kesamaans

Fungsi	Hasil
Precission	86 %
Recall	100 %
F-Score	92 %

Berdasarkan hasil pengujian akurasi perhitungan similarity menggungakan metode confusion matriks tersebut, menunjukkan bahwa implementasi algoritme Frequency Based Hashing-S menggunakan metode Approximate Matching pada file PDF memiliki akurasi yang tinggi

4.2.7 Pengujian Integritas Data

Setelah program dijalankan dan pengujian dilakukan sesuai dengan skenario, diperoleh hasil yang menunjukkan algoritme FbHash-S ini aman terhadap serangan aktif.

SIMPULAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang dilakukan didapatkan kesimpulan sebagai berikut:

1 Sistem deteksi plagiasi menggunakan algoritme Frequency Based Hashing-S pada file PDF berhasil dibangun sesuai dengan perancangan dan berjalan dengan baik sesuai dengan skenario dan kebutuhan yang sudah ditetapkan.

2 Berdasarkan pengujian kinerja sistem, program yang dibuat sudah efisien dengan rata-rata waktu eksekusi sebanyak 0,36 detik. Untuk proses unggah dataset dan proses unggah datatest, didapat kesimpulan bahwa semakin banyak jumlah chunk maka waktu eksekusi semakin lama begitu juga sebaliknya. Proses unggah dataset lebih lama dengan rata-rata waktu eksekusi 4,5 detik dibandingkan dengan proses unggah datatest yang memiliki rata-rata waktu eksekusi 3.0 detik. Hal ini dikarenakan pada proses unggah dataset terdapat dua perintah yaitu mengguggah file ke tabel kosin_proses_dataset dan menggunggah chunk ke tabel kosin_proses_tf. Sementara pada proses unggah datatest hanya terdapat satu perintah, yaitu mengunggah file ke tabel kosin_proses_datatest. Untuk jumlah file yang berada di dataset, didapatkan kesimpulan bahwa semakin banyak jumlah file didataset maka waktu eksekusi akan semakin lama begitu juga sebaliknya.

3 Permasalahan tentang penyaringan file PDF berdasarkan tingkat kesamaan/plagiasi dapat teratasi dengan mengimplementasikan algoritme Frequency Based Hashing-S. Pengimplementasian algoritme Frequency Based Hashing-S pada file PDF telah terbukti memiliki akurasi yang tinggi berdasarkan hasil pengujian menggunakan metode Confusion Matrix, dengan hasil F-Score 92%, Recall 100% dan Precission 86%

4 Permasalahan algoritme Approximate Matching sekarang yang rentan terhadap serangan aktif dapat teratasi dengan pengimplementasian

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

algoritme Frequency Based Hashing-S. Hal ini dapat dibuktikan berdasarkan hasil pengujian integritas data menggunakan metode Collision Attack. Sehingga data dapat terjamin integritasnya tanpa diubah oleh pihak yang tidak bertanggung jawab.

5.2 Saran

Dari kesimpulan yang telah dipaparkan, Saran yang dapat peneliti berikan adalah agar mengimplementasikan Optical Character Recognition yang lebih mendetail. Hambatan pada penelitian ini adalah skema pengkodean menggunakan libarary PyPDF sebagai OCR masih kurang memuaskan untuk mengenali objek yang terdapat pada file PDF. Dokumen PDF menggunakan penyandian seperti UTF-8, ASCII, Unicode, dll. Mengekstrak teks yang berada pada PDF dapat mengakibatkan hilangnya Sebagian data karena skema pengkodean tersebut. Oleh sebab itu, dengan menggunakan OCR yang lebih mendetail menggunakan Deep Learning seperti penggunaan algoritme Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), dan Artificial Neural Network (ANN) diharapkan mampu meningkatkan akurasi perhitungan similarity.

DAFTAR PUSTAKA

Ahmad, T. et al., 2019. Model-based testing using UML activity diagrams: A systematic mapping study. Computer Science Review, Volume 33, pp. 98-112.

Al-Fedaghi, S., 2017. Diagramming the Class Diagram: Toward a Unified Modeling Methodology. International Journal of Computer Science and Information Security, Volume 15.

Alvin, C., Peterson, B. & Mukhopadhyay, S., 2021. Static generation of UML sequence diagrams. International Journal on Software Tools for Technology Transfer, 23(1), pp. 31-53.

Association, P., 2018. https://www.pdfa.org/. [Online]
Available at: https://www.pdfa.org/wp-content/uploads/2018/06/1330_Johnson.pdf
[Accessed 17 February 2022].

Bulla, C. et al., 2017. My Campus Android Application. International Journal of Engineering Science (IJESC), 7(2).

Chang, D. et al., 2019. FbHash: A New Similarity Hashing Scheme for Digital Forensics. ELSEVIER, Volume 29, pp. S113-S123.

Chang, D., Sanadhya, S., Singh, M. & Verma, R., 2015. A collision attack on sdhash similarity hashing. s.l., s.n., p. 36e46.

Harbour, N., 2002. Dcfldd. Department of Defense Computer Forensics Lab (DCFL).

Kornblum, J., 2006. Identifying almost identical files using context triggered piecewise hashing. Digital Investigation, Volume 3, pp. 91-97.

Kulkarni, R. N. & Srinivasa, C. K., 2019. Ameliorated Methodology to Represent UML use Case Diagram into Table Format. International Journal of Engineering and Advanced Technology (IJEAT), 9(1).

Kurniawan, T. A., 2018. Pemodelan Use Case (UML): Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 5(1).

Lillis, D., Breitinger, F. & Scanlon, M., 2018. Hierarchical Bloom Filter Trees

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

for Approximate Matching. Journal of Digital Forensics, Security and Law, 03, 13(1), pp. 81-96.

Loki, G. & Gal, P., 2018. JavaScript Guidelines for JavaScript Programmers - A Comprehensive Guide for Performance Critical JS Programs. s.l., 13th International Conference on Software Technologies.

Marr, B., 2018. forbes.com. [Online] Available at: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#3977f0b60ba9 [Accessed 29 September 2020].

Meydiansyah, D. Y., 2021. Fenomena Perilaku Menyontek Pada Pelajar Masa Kini Ditinjau Dari Kepercayaan DIri, Efikasi Diri, dan Prokrastinasi. Consilia: Jurnal Ilmiah BK, 4(2), pp. 245-253.

Muhammed, A. & Varol, N., 2019. A Review Paper on Cryptography. s.l., s.n., pp. 1-6.

Ndia, J., Muketha, G. & Omieno, K., 2019. A Survey of Cascading Style Sheets Complexity Metrics. International Journal of Software Engineering & Applications, 10(3).

Nitnaware, R., 2019. Basic Fundamental of Python Programming Language and The Bright Future. A Peer-Reviewed Journal About, Volume VIII, pp. 71-76.

Pratiwi, N. A., 2020. Rancang Bangun Sistem Informasi Kegiatan dan Prestasi Berbasis Website Pada Program Studi Kimia Universitas Cokroaminoto Palopo. Skripsi. Universitas Cokroaminoto. Palopo.

Purwasih, R., Putri, Y. P. & Prima, A., 2017. Rancang Bangun Kamus Minang-Inggris Menggunakan Bahasa Pemrograman Hypertext Processor (PHP). Jurnal Penelitian Pendidikan Bahasa dan Sastra Indonesia, 3(2).

Rao, P. V. et al., 2019. Improve the Integrity of Data Using Hashing Algorithms. International Journal of Innovative Technology and Exploring Engineering, 8(7).

Resta, O. A., Aditya, A. & Purwiantono, F. E., 2021. Plagiarism Detection in Students' Theses Using The Cosine Similarity Method. Sinkron : Jurnal dan Penelitian Teknik Informatika, 5(2).

Roussev, V., 2011. An evaluation of forensic similarity hashes. Digital Investigation, Volume 8, pp. S34-S41.

Sahu, N. & Sonkusare, M., 2017. A Study on Optical Character Recognition Techniques. The International Journal of Computational Science, Information Technology and Control Engineering (IJCSITCE), 4(1).

Sharma, A., 2018. Introduction to HTML (Hyper Text Markup Language) - A Review Paper. International Journal of Science and Research (IJSR), Volume 7.

Standsyah, R. E. & Restu, I. S., 2017. implementasi PhpMyAdmin Pada Rancangan Sistem Pengadministrasian. Unisda Journal of Mathematics and Computer Science, 3(2).

Wati, I. M., Soebagjo, A. S. & Justitia, D., 2012. Self-Regulated Learning Siswa Yang Menyontek (Survey Pada Siswa Kelas X DI SMA N 52 Jakarta Utara Tahun Ajaran 2010/2011). Insight: Jurnal Bimbingan dan Konseling, 1(2).

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

	Biodata Penulis Pertama Thamrin Auliya, adalah mahasiswa Prodi Teknik Informatika
	Biodata Penulis Kedua Cosmas Eko Suharyanto, S.Kom., M.MSI. adalah dosen Prodi Teknik Informatika