

## **BAB II**

### **KAJIAN PUSTAKA**

Kajian pustaka berisi tentang usaha penulis untuk menggali berbagai informasi berkaitan dengan penelitian yang bersumber dari jurnal ilmiah, laporan penelitian, artikel ilmiah, jurnal survei dan sumber lainnya. Kajian pustaka ini juga untuk menjelaskan tentang variabel-variabel yang menjadi topik dalam penelitian.

#### **2.1 Dasar Teori**

##### **2.2.1. Kriptografi**

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. Hashing**

*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: Byte-wise Matching, Syntactic Matching dan Semantic Matching.

Byte-wise matching dilakukan dengan mengukur kemiripan objek digital pada level *byte* tanpa mempertimbangkan struktur internal objek data. Syntactic matching dilakukan dengan mengukur kemiripan berdasarkan pada struktur internal objek data. Sementara semantic matching mengukur kesamaan berdasarkan atribut kontekstual dari objek digital.

### **2.2.4. Optical Character Recognition**

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

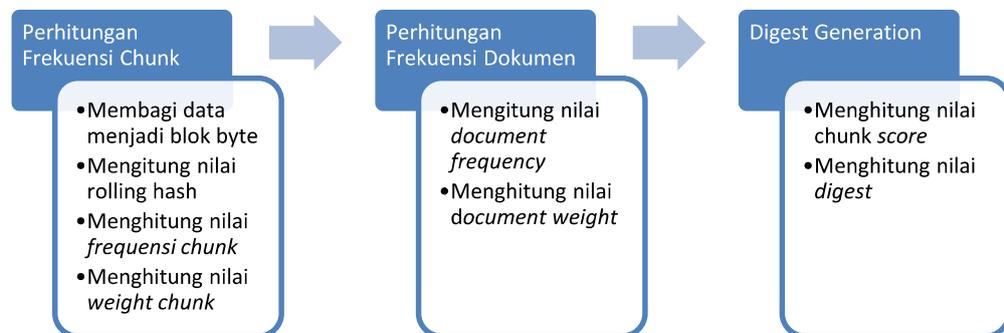
& Sonkusare, 2017). Optical Character Recognition atau OCR digunakan untuk mengekstrak teks yang berada dalam *file* PDF. Karena Bahasa yang digunakan adalah Python, maka untuk proses OCR dilakukan menggunakan *library* PyPDF2. Kekurangan utama menggunakan *library* ini adalah skema pengkodean. Dokumen 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 vector D2 (digest 2).

#### **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 2. 1 Tahapan Algoritme Frequency Based Hashing-S**

### 2.2.5.1. Perhitungan Frekuensi *Chunk*

Pada langkah ini, pertama membagi dokumen menjadi blok *byte* tertentu. Setiap blok disebut sebagai *chunk*. Tujuannya adalah untuk menghitung berapa kali setiap *chunk* muncul dalam dokumen yang diberikan, yaitu menghitung frekuensi *chunk*.

- Misalkan  $D = B_0^D, B_1^D, B_2^D, \dots, B_{l-1}^D$  menjadi dokumen dengan panjang  $l$  byte, di mana  $B_i^D$  menunjukkan byte  $i^{th}$  dari dokumen  $D$ . Sebuah *chunk* adalah urutan  $k$  byte berurutan dari  $D$ , di mana:

$$ch_0^D = B_0^D, B_1^D, B_2^D, \dots, B_{k-2}^D, B_{k-1}^D \quad (2.1)$$

$$ch_1^D = B_1^D, B_2^D, B_3^D, \dots, B_{k-1}^D, B_k^D \quad (2.2)$$

$$ch_2^D = B_2^D, B_3^D, B_4^D, \dots, B_k^D, B_{k+1}^D$$

(2.3)

$$ch_i^D = B_i^D, B_{i+1}^D, B_{i+2}^D, \dots, B_{i+k-2}^D, B_{i+k-1}^D$$

(2.4)

$$ch_{i-k}^D = B_{i-k}^D, B_{i-k+1}^D, B_{i-k+2}^D, \dots, B_{i-2}^D, B_{i-1}^D$$

(2.5)

2. Untuk menghitung frekuensi dari setiap *chunk* yang teridentifikasi dalam dokumen, teknik *Rolling hash* digunakan. *Rolling hash* adalah fungsi *hash* non-kriptografik yang memungkinkan komputasi *hash* yang cepat untuk setiap *chunk* yang berurutan.

$$\text{Rolling hash } (ch_i) = B_i^D a^{k-1} + B_{i+1}^D a^{k-2} + B_{i+2}^D a^{k-3} + \dots + B_{i+k-1}^D a^0 \text{ modulus } n$$

(2.6)

3. Setelah nilai *rolling hash* dari sebuah *chunk* dihitung, frekuensi setiap *chunk* akan dihitung dengan berapa kali nilai *rolling hash* muncul. Nilai *rolling hash* di simpan di dalam tabel hash. Untuk menjamin bahwa setiap *chunk* unik mendapatkan nilai *rolling hash* unik, yaitu tidak ada nilai yang sama, nilai  $n$  diambil sebagai bilangan prima yang lebih besar dari

$$2^{56} \text{ (karena, } 256 * 256^{k-1} = 2^{56} \text{).} \quad (2.7)$$

4. Berdasarkan frekuensi *chunk*, bobot *chunk* akan dihitung untuk setiap *chunk*, menggunakan rumus berikut:

$$\text{Ch\_wght}_{\text{ch}} = 1 + \log_{10}(\text{chf}_{\text{ch}}^d) \quad (2.8)$$

Jadi, semakin tinggi frekuensi *chunk*, semakin tinggi bobot *chunk* dan sebaliknya.

### 2.2.5.2. Perhitungan Frekuensi Dokumen

Frekuensi dokumen dari sebuah *chunk* adalah jumlah dokumen yang berisi *chunk* tersebut. Untuk menghitung frekuensi dokumen, nilai  $N$  ditetapkan 1000. Frekuensi dokumen dari *chunk*  $ch$  disebut sebagai  $\text{df}_{\text{ch}}$ . Frekuensi dokumen dihitung dengan langkah sebagai berikut:

1. Identifikasi *chunk* setiap dokumen dalam dataset. Kemudian menghitung nilai *rolling hash* dari setiap *chunk*. Setelah itu membuat tabel *hash* di mana indeks tabel *hash* menunjukkan nilai *Rolling hash* dari *chunk* dan nilai tabel *hash* menunjukkan frekuensi dokumen dari *chunk* yang sesuai. Setiap *chunk* unik dari setiap dokumen akan meningkatkan nilai tabel *hash* yang di indeks oleh *Rolling hash* sebesar 1.
2. Berdasarkan frekuensi dokumen, bobot dokumen akan dihitung untuk setiap *chunk* yang menunjukkan keinformatifan atau keunikan *chunk*. Di lambangkan sebagai  $\text{doc\_wght}_{\text{ch}}$  dan  $\text{doc\_wght}_{\text{ch}}$  dihitung sebagai berikut:

Jika  $df_{ch} > 0$  : maka  $doc\_wght_{ch} = \log_{10}(1000/df_{ch})$

(2.9)

Sebaliknya:  $doc\_wght_{ch} = 1$

(2.10)

### 2.2.5.3. Perhitungan Nilai Digest

Untuk menghitung nilai *digest*, dilakukan beberapa langkah seperti berikut:

1. Setelah memiliki nilai bobot *chunk* dan bobot dokumen dari setiap *chunk* dalam dokumen *D*, skor *chunk* (dilambangkan sebagai  $w_{ch_i}^d$ )

kemudian dihitung sebagai berikut:

$$w_{ch_i}^d = ch\_wght_{ch_i}^d * doc\_wght_{ch_i}$$

(2.11)

Skor *chunk* ini akan digunakan untuk menghitung kesamaan antara dua dokumen.

2. Digest FbHash akhir dari dokumen *D* dapat direpresentasikan sebagai *n* elemen panjang vektor di mana indeks vektor mewakili nilai  $w_{ch_i}^d$ .

## **2.2 Variabel (Indikator Masalah)**

### **2.2.1 Pemeriksaan Tugas Siswa Secara Manual Kurang Efektif**

Kepercayaan diri adalah salah satu sifat manusia yang unik dan berharga. Setiap orang memiliki tingkat kepercayaan diri yang berbeda-beda. Efikasi Diri (Self Efficacy) adalah salah satu aspek pengetahuan tentang diri yang mempengaruhi kehidupan manusia sehari-hari (Meydiansyah, 2021). Efikasi diri memiliki keterkaitan dengan perilaku menyontek. Perilaku mencontek merupakan salah satu dampak kurangnya percaya diri. Menyontek merupakan sebuah tindakan tidak jujur atau tidak fair dalam rangka memenangkan atau meraih keuntungan.

Pada penelitian yang dilakukan oleh Wati dkk (Wati, et al., 2012) pada siswa kelas X SMA Negeri 52 Jakarta Utara tahun ajaran 2010/2011, dari 153 siswa didapatkan hasil 5,2% selalu menyontek, 37,9% sering menyontek, 52,6% jarang menyontek dan 0,7% tidak pernah menyontek. Melakukan perilaku mencontek berarti bahwa tugas yang dikumpul akan sama atau persis sama dengan tugas siswa yang lainnya. Jika dilakukan pemeriksaan semua tugas siswa secara manual akan melelahkan dan kurang efisien karena dari semua tugas siswa tersebut mungkin terdapat tugas siswa yang plagiasi. Dengan melakukan penyaringan tugas yang tidak plagiasi diantara semua tugas siswa, maka akan mengurangi tugas

yang harus diperiksa secara manual. Sehingga akan mengurangi waktu yang dibutuhkan untuk mengoreksi atau memeriksa tugas siswa.

### **2.2.2 Metode Pencocokan Fungsi Hash Tradisional Memiliki Keterbatasan**

Proses pemfilteran yang digunakan dalam mengekstraksi data biasanya menggunakan Fast Hashing Based Algorithms. Dimana, *file* besar dilewatkan melalui fungsi *hash* untuk menghasilkan keluaran *hash* yang disebut sidik jari digital atau *fingerprint*. Sidik jari dari *file* datatest kemudian dicocokkan dengan kumpulan data referensi dari dataset untuk mengekstrak *file* yang tidak berhubungan. Proses pemfilteran dapat dilakukan dengan *Black Listing* atau *White Listing*. *Black Listing* merupakan proses penyaringan data dengan mencocokkan *file* datatest dengan kumpulan dataset. *File* yang dihasilkan setelah proses ini adalah *file* yang perlu diperiksa secara manual atau tidak plagiasi. Di sisi lain, *White Listing* merupakan proses pemfilteran dengan mencocokkan *file* datatest dengan kumpulan *file* dataset. Berkas yang lolos proses ini tidak perlu diperiksa secara manual atau plagiasi.

Pencocokan berbasis fungsi *hash* tradisional (kriptografik) mengalami keterbatasan bahwa jenis pemfilteran ini hanya menunjukkan salinan persis dari *file* lain. Hal ini disebabkan oleh fakta bahwa bahkan sedikit perubahan dalam konten *file* menghasilkan keluaran *hash* yang

sama sekali tidak terkait dan tampak acak sedangkan persyaratan dalam praktiknya skenario sering menemukan *file* serupa.

Approximate Matching adalah teknik umum untuk menemukan kesamaan di antara *file* yang diberikan, biasanya dengan menetapkan 'skor kesamaan'. Teknik ini digunakan saat ini oleh penyelidik forensik digital.

### **2.2.3 Algoritme Approximate Matching Rentan Terhadap Serangan Aktif**

#### **2.2.3.1 Dcfldd**

Pendekatan Approximate Matching pertama untuk forensik digital diusulkan oleh Nicolas Harbour pada tahun 2002 yang disebut dcfldd. dcfldd adalah skema *hashing* berbasis blok (Harbour, 2002). Dalam skema ini, setiap *file* dibagi menjadi blok ukuran tetap dan keluaran *hash* dihasilkan untuk setiap blok. Intisari terakhir adalah rangkaian dari semua *hash* blok.

#### **2.2.3.2 Ssdeep**

Perbaikan atas dcfldd diusulkan oleh Kornblum dan diberi nama "Context Triggered Piecewise Hashing" (CTPH). Skema CTPH didasarkan pada algoritma deteksi email yang disebut spamsum, yang diusulkan oleh Andrew Tridgell. Alih-alih mencirikan blok ukuran tetap, CTPH membagi data dalam blok ukuran variabel dan kemudian setiap blok di-*hash* menggunakan fungsi *hash* (nonkriptografik) yang disebut

*hash* FNV (Kornblum, 2006). Skema ini terbukti dapat mendeteksi *file* serupa dengan lebih akurat dibandingkan dengan skema pemblokiran *hashing*. Alat yang mengimplementasikan CTPH dikenal sebagai *ssdeep*, yang juga merupakan nama yang biasa dirujuk untuk skema *hashing* itu sendiri. Namun, Breitinger menyajikan analisis menyeluruh tentang *ssdeep* dan menunjukkan bahwa *ssdeep* tidak menahan musuh aktif untuk menghindari black listing.

### **2.2.3.3 SdHash**

Roussev dkk. (Roussev, 2010) mengusulkan skema baru yang disebut *sdhash* pada tahun 2010. Ide utama dalam skema *sdhash* adalah untuk menghasilkan hash akhir hanya dengan menggunakan fitur dokumen yang tidak mungkin secara statistik. Keamanan terperinci dan analisis implementasi *sdhash* disajikan oleh Breitinger. Pekerjaan ini mengungkap masalah implementasi dan keamanan dan menunjukkan bahwa dimungkinkan untuk mengalahkan skor kesamaan dengan merusak file tertentu tanpa mengubah perilaku persepsi file ini (misalnya, file gambar terlihat hampir sama meskipun ada gangguan). Klaim sekali lagi diverifikasi oleh Chang yang menghadirkan serangan di mana musuh dapat menyesatkan penyidik dengan beberapa file serupa.

#### **2.2.3.4 BbHash dan Mrsh-v2**

Dua skema baru yang dikenal sebagai bbHash dan mrsh-telah diusulkan oleh Breitinger pada tahun 2012. Namun, karena kompleksitas waktu proses yang tinggi, bbHash tidak dapat digunakan secara praktis.

#### **2.2.3.5 MvHash-B**

Breitinger mengusulkan skema lain pada tahun 2013 yang disebut mvHash-B. Skema ini bekerja dalam tiga fase: pertama mengompresi data input menggunakan voting mayoritas, kemudian melakukan pengkodean run-length dan terakhir menyimpan sidik jari ke dalam filter Bloom. 'B' di mvHash-B menunjukkan representasi filter bloom dari intisari kesamaan. Dalam hal kinerja, mvHash-B adalah salah satu skema yang paling efisien di antara semua skema yang ada dengan kompleksitas waktu proses terendah dan ukuran ringkasan yang kecil. Analisis menyeluruh mvHash-B disajikan oleh Chang yang mengungkap kelemahan dari skema mvHash-B dan menunjukkan bahwa mvHash-B tidak menahan musuh aktif terhadap blsklist dan juga mengusulkan perbaikan pada desain mvHash-B untuk mengurangi kelemahan tersebut.

### **2.3 Perangkat Lunak Pendukung**

Perangkat lunak pendukung merupakan software yang digunakan dalam membangun sistem. Adapaun beberapa perangkat lunak yang digunakan pada penelitian ini yakni:

### **2.3.1 XAMPP**

XAMPP dikembangkan oleh Apache Friends berupa sebuah paket solusi cross- platform yang dapat diakses secara gratis oleh siapa saja. Paket solusi cross-platform ini terdiri atas Apache HTTP Server, database MariaDB, dan interpreters sebagai script yang ditulis dalam bahasa pemrograman PHP dan Perl (Bulla, et al., 2017). XAMPP adalah singkatan dari Cross Platform (X), Apache (A), MariaDB (M), PHP (P), dan Perl (P). Ini adalah distribusi Apache yang sederhana dan ringan yang membuatnya sangat mudah bagi pengembang untuk membuat server web lokal untuk tujuan pengujian dan penerapan.

#### **2.3.1.1 PHP: Hypertext Preprocessor(PHP)**

PHP (Hypertext PreProcessor) adalah bahasa pemrograman yang bisa digunakan untuk mengolah data dari server agar dapat ditampilkan pada website. PHP (Hypertext Preprocessor) yang merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis (Purwasih, et al., 2017). Aplikasi aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di server. PHP dapat digunakan dengan gratis (free) dan bersifat Open Source. Dari beberapa kelebihan pada bahasa pemrograman PHP, hal ini mendukung pengguna dalam memasukkan fitur pengolahan data secara cepat.

### **2.3.1.2 HTML**

HTML adalah singkatan dari Hyper Text Markup Language dan digunakan untuk menggambarkan struktur halaman web. HTML adalah Bahasa standar untuk Markup Language (Sharma, 2018). HTML ini biasanya digunakan untuk mengembangkan halaman Web. Dalam HTML digunakan berbagai tag seperti "heading", "paragraph", "table", dan sebagainya.

### **2.3.1.3 CSS (Cascading Style Sheet)**

CSS merupakan singkatan dari Cascading Style Sheets. CSS adalah bahasa style sheet berbasis web yang digunakan untuk presentasi dokumen web (Ndia, et al., 2019). CSS telah berkembang dari CSS1 ke CSS3. CSS ini merupakan bagian integral dari aplikasi berbasis web yang tujuannya adalah untuk memisahkan konten dari presentasi. Pada dasarnya, bahasa CSS memungkinkan untuk menata halaman web pada tema seperti penggunaan warna, font, dan tata letak.

### **2.3.1.4 Javascript**

JavaScript adalah bahasa untuk mengembangkan situs web dinamis. JavaScript masih merupakan pilihan paling populer untuk pemrograman web saat ini (Loki & Gal, 2018). Karena JavaScript adalah inti dari web interaktif kontemporer dimana setiap orang menggunakan jejaring sosial, layanan pemesanan makanan online, atau internet banking,

menggunakan setidaknya satu mesin yang terpasang di browser mereka. Namun web bukan satu-satunya kasus penggunaan JavaScript, ada skenario komunikasi *machine-to-machine* dengan perangkat tertanam yang menjalankan mesin mandiri.

### **2.3.1.5 Python**

Python adalah bahasa pemrograman berorientasi objek tingkat tinggi yang kuat yang dibuat oleh Guido van Rossum. Python menggunakan bahasa pemrograman yang memiliki sintaks tertata rapi dan alat yang kuat untuk menyelesaikan tugas apa pun. Selain itu python memiliki sintaks yang sangat dekat dengan pemikiran matematika sederhana. Python dipilih sebagai bahasa pemrograman utama untuk mahasiswa baru di sebagian besar universitas terkemuka. Python adalah bahasa yang cocok untuk pembelajaran dan pemrograman dunia nyata (Nitnaware, 2019).

### **2.3.1.6 MySQL dan phpMyAdmin**

MySQL adalah RDBMS yang didistribusikan secara gratis dibawah lisensi GNU General Public License, dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. Sebagai database server, MySQL memiliki kelebihan dalam query datanya dibandingkan dengan database server lain. Hal ini dibuktikan pada query yang dilakukan oleh single user,

dimana kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase. PhpMyAdmin merupakan sebuah aplikasi / perangkat lunak bebas (opensource) dalam bahasa pemrograman PHP dan digunakan untuk menangani administrasi database MySQL, baik melalui jaringan lokal maupun internet. phpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (fields), relasi (relations), indeks, pengguna (users), perijinan (permissions), dan lain-lain). Perbedaan phpMyAdmin dengan MySQL terletak pada fungsi. PhpMyAdmin merupakan alat untuk memudahkan dalam mengoperasikan database MySQL, sedangkan MySQL adalah database tempat penyimpanan data, phpmyadmin sendiri digunakan sebagai alat untuk mengolah / mengatur data pada MySQL (Standisyah & Restu, 2017).

### **2.3.2 Sublime**

Sublime text adalah text editor berbasis Python, sebuah text editor yang elegan, kaya fitur, cross platform, mudah dan simple yang cukup terkenal dikalangan developer (pengembang) dan desainer. Sublime Text 3 juga digunakan sebagai editor dari Bahasa pemrograman PHP dalam melakukan pengelolaan konten di dalam aplikasi server. Sublime text adalah aplikasi editor yang sangat powerful yang dapat meningkatkan

proktivitas dan mengembangkan kualitas kode yang tinggi dan dapat berjalan di berbagai platform sistem operasi dengan menggunakan teknologi Python API (Pratiwi, 2020).

### **2.3.3 Penelitian Terdahulu**

Berikut ini beberapa penelitian terdahulu mengenai sistem pencocokan yang menjadi referensi dalam penulisan laporan penelitian ini:

1. (Chang, D. et al. 2015) A Collision Attack on Sdhash, In: Proceedings of 10th Intl. Conference on Systematic Approaches to Digital Forensic Engineering, pp. 36e46. Skor kesamaan hashing pada SdHash memiliki kemungkinan untuk diserang karena seluruh konten data tidak berkontribusi pada perhitungan digest akhir, hanya beberapa potongan data yang berkontribusi pada perhitungan digest akhir. Oleh sebab itu dibutuhkan algoritme baru yang aman terhadap serangan aktif.
2. (Roussev 2011) An Evaluation of Forensic Similarity Hashes, Digital Investigation, Volume 8, pp. S34-S41. Ssdeep sangat bergantung pada keberadaan sejumlah besar data yang berkesinambungan, begitu berada di luar zona nyaman, skor

kesamaan dan kemampuan korelasi turun dengan cepat, ditambah lagi tidak adanya hubungan yang mudah dikenali antara tingkat level kesamaan dan skor kesamaan yang membuat penyaringan dan penentuan prioritas bermasalah. Oleh sebab itu dibutuhkan algoritme baru yang aman terhadap serangan aktif

3. (Chang ,D. et al. 2019) FbHash: A New Similarity Hashing Scheme for Digital Forensics, ELSEVIER, Volume 29, pp. S113-S123. Jurnal tersebut memperkenalkan algoritme hashing baru untuk melakukan penyortiran data berdasarkan tingkat kesamaannya. Hasil dari penelitian ini menunjukkan bahwa skema FbHash ini memiliki tingkat akurasi 28% lebih tinggi daripada skema lain untuk format data tidak terkompresi (misalnya, data teks) dan tingkat akurasi 50% lebih tinggi untuk format data terkompresi (mis., Docx dll.). Skema ini juga mampu menghubungkan sebuah fragmen sekecil 1% ke data sumber dengan tingkat deteksi 100% dan mampu mendeteksi kesamaan sekecil 1% antara dua dokumen dengan skor kesamaan yang sesuai. Namun penelitian ini hanya mengimplementasikan skema FbHash pada data berformat text dan docx.

Sementara untuk PDF belum ada penelitian lebih lanjut.

4. Menurut (Rao et al., 2019), dengan judul “Improve the integrity of data using hashing algorithms” ISSN : 2278-3075. penemuan teknologi baru komunikasi informasi sangat cepat dan pengiriman cepatinformasi bahkan untuk jarak jauh menjadi mudah di dunia. Namun, komunikasi online ini telah menimbulkan banyak kerentanan, seperti keandalan dan integritas data. Ada banyak sandi kriptografi, tetapi mereka rentan terhadap berbagai jenis serangan. Buku putih ini berfokus pada berbagai jenis serangan dan bagaimana mereka membahayakan kepercayaan dan integritas data Anda. Algoritma yang berbeda digunakan untuk memastikan keandalan dan integritas data dipertimbangkan, dan studi perbandingan algoritma yang berbeda dan kelemahannya juga dilakukan. Fokusnya adalah pada teknologi hashing.
5. Menurut (Sahu & Sonkusare, 2017), dengan judul “A Study on Optical Character Recognition Techniques” Optical Character Recognition (OCR) merupakan proses yang bisa memungkinkan sistem tanpa campur tangan

manusia. Karakter Optik identifikasi bisa berkembang menjadi individu dari aplikasi pengetahuan yang telah berkembang pesat pada bidang deteksi pola dan kecerdasan buatan. Dalam survei kami, kami mempelajari berbagai teknik OCR. Identifikasi atau klasifikasi karakter optik (OCR) dan Pengenalan Karakter Magnetik (MCR) biasanya digunakan untuk pengenalan pola atau huruf. Secara umum abjadnya adalah dalam berbagai macam gambar piksel serta dapat berupa tulisan tangan atau cap, dari seri, bentuk apa pun atau arah dll. Atau di MCR, huruf dicap dengan tinta magnetik dan mesin belajarmeng kategorikan alfabet berdasarkan medan magnet eksklusif yang dibentuk oleh setiap alfabet.