Jurnal Kajian Ilmiah Teknologi Informasi dan Komputer
ISSN 2962-9055 (Media Online)
Vol 3, No 1, November 2024
Hal 1-8
https://journal.grahamitra.id/index.php/jutik

Perancangan Aplikasi Pembangkit Identitas File Dokumen Menerapkan Algoritma Fungsi HASH MD

Suriani Ndruru

Program Studi Teknik Informatika, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Budi Darma,
Jalan Sisingamanganraja No. 338, Medan, Sumatera Utara, Indonesia
Email: surianinduru732@gmail.com

Abstrak

Saat ini sudah berada dalam era dunia digital. Akan tetapi sekarang banyak industri yang sudah beralih teknologi digital seperti foto, video, musik dan lain sebagainya. Tentunya perubahan ini membawa banyak manfaat untuk kita dalam hal kecepatan akses data. Kelebihan lain dari teknologi komputer adalah dalam hal menyalin dan membangkitkan identitas file digital dimana hasilnya akan sama nilai identitasnya. Sehingga jika ingin membedakan satu file dokumen dengan yang lainnya akan sangat sulit karena harus mengecek seluruh isi file dokumen tersebut. Dalam kriptografi terdapat fungsi hash yang merupakan fungsi satu arah yang dapat membangkitkan identitas sebuah file. Dimana jika file dokumen itu isinya sama maka akan menghasilkan nilai hash yang sama. Hal ini dapat digunakan untuk mengindentifikasi file dokumen yang sama. Dalam fungsi hash terdapat metode MD2 ataupun perancangan aplikasi pembangkit identitas file dokumen menerapkan algoritma fungsi hash MD2 yang merupakan salah satu algoritma hashing yang sering digunakan untuk mengenkripsi data dengan lebar 20 byte atau 160 bit. Dengan menerapkan algoritma ini pada aplikasi dokumen identitas maka hasil dari pencarian file dokumen tersebut dikelompokkan berdasarkan nilai hash yang sama dan pengguna dapat memudahkan dan mempercepat untuk menghapus file dokumen yang identitas sama tanpa harus membuka dan membaca isi file tersebut.

Kata Kunci : Identitas; Perancangan; Algoritma MD2; Dokumen

Abstract

Currently in the digital world era. However, now many industries have switched to digital technology such as photos, videos, music and so on. Of course, this change brings many benefits to us in terms of data access speed. Another advantage of computer technology is in terms of copying and generating digital file identities where the results will have the same identity value. So if you want to distinguish one document file from another, it will be very difficult because you have to check the entire contents of the document file. In cryptography there is a hash function which is a one-way function that can generate the identity of a file. Where if the document file has the same contents, it will produce the same hash value. This can be used to identify the same document file. In the hash function there is an MD2 method or the design of a document file identity generator application applies the MD2 hash function algorithm which is one of the hashing algorithms that is often used to encrypt data with a width of 20 bytes or 160 bits. By applying this algorithm to the identity document application, the results of the document file search are grouped based on the same hash value and users can easily and quickly delete document files with the same identity without having to open and read the contents of the file.

Keywords: Identity; Design; MD2 Algorithm; Document

1. PENDAHULUAN

Format-format dokumen yang selama ini digunakan untuk media pengolah kata pada perangkat lunak komputer, dimana file yang berformat dokumen ini pada umumnya apabila dibuka dengan aplikasi tertentu memiliki fungsi untuk menulis, edit tulisan, atau sekedar membaca saja. Misalnya pada perangkat lunak Microsoft Word dapat membuka dan menyimpan file berformat doc dan docx. File dokumen di identitaskan karena file dokumen sangat banyak digunakan saat ini dan sangat penting untuk merepresentasikan isi file. Memilih MD2 karena algoritma ini memiliki tingkat kompleksitas yang rendah sehingga mempercepat proses komputasi.

Perkembangan teknologi dalam era digital saat ini menyebabkan banyak industri yang sudah beralih teknologi digital seperti foto. Video, musik, dan lain sebagainya. Tentunya perubahan ini membawa banyak manfaat untuk kita dalam hal kecepatan dan akses data. Kelebihan lain dari teknologi komputer adalah dalam hal membangkitkan suatu identitas dari data yang disimpan didalam sistem berkas yang dapat diakses dan diatur oleh pengguna. Sebuah berkas atau file mempunyai nama yang unik dalam suatu direktori atau folder dimana file itu berada. Setiap software atau program pasti memiliki jenis atau tipe file tersendiri. Sebagai contoh, misalnya program photoshop secara default filenya berjenis PPT, MS Word berjenis DOC dan lain sebagainya. Format atau jenis file pada program-program tersebut umumnya dikembangkan oleh perusahaan yang menciptakan program yang bersangkutan. Meskipun tidak bisa digeneralisasi, namun jenis file yang dikembangkan oleh perusahaan-perusahaan penghasil software umumnya bertujuan untuk memberikan identitas pada dokumen yang dihasilkan atas penggunaan software tersebut.

Jenis file tertentu umumnya hanya bisa dibuka menggunakan program yang menghasilkan jenis file itu. Namun kini hal tersebut tidaklah berlaku mutlak, karena telah banyak beredar program yang dapat digunakan untuk membuka atau mengakses file dengan jenis tertentu. Setiap aplikasi menyimpan file dengan extensi yang berbeda atau bahkan

sama jika mengikuti standar. Pengertian extensi file atau file extension adalah suffix atau akhiran dari nama file yang biasanya dipisahkan dengan period atau dot atau titik, yang menjelaskan jenis dan tipe file dari dokumen tersebut. Tujuan memberikan suffix atau extensi didalam komputer untuk memudahkan kita dalam mengenali jenis file dan untuk menentukan program yang tepat untuk mengedit file. Seperti mengenali dan membedakan jenis dan tipe dari setiap dokumen,dengan file yang memiliki akhiran, suffix, extensi.mp3 dengan cepat kita simpulkan bahwa file tersebut yaitu file musik. File extensi juga digunakan oleh sistem sebagai pengenal jenis file dan ditentukan aplikasi untuk mengeditnya dengan tepat.

Fungsi hashing kriptografi memiliki banyak kegunaan seperti digital tanda tangan, otentikasi pesan, pembuatan checksum, digital sidik jari dan mengamankan kata sandi dalam database. Ada beberapa algoritma yang sedang diteliti adalah terbukti lemah dan mudah pecah. Perbaikan selanjutnya memiliki dan dibuat untuk yang lebih baru untuk memastikan keamanan yang lebih tinggi. Kebanyakan aspek umum dari algoritma hashing menjadi fokus membantu memahami keuntungan dan kerugian hashing. Hashing digunakan untuk mewakili file digital, pesan atau entitas apapun kedalamnya string karakter yang lebih pendek, panjang tetap dan unik dalam file cara perhitungan hash untuk entitas digital akan selalu menjadi sama dan tidak mungkin untuk mengambil digital asli entitas dan string hash-nya. Fungsi hash kriptografi memetakan string (pesan) hampir panjang acak ke-string dengan panjang tetap dan pendek, biasanya suatu tempat antara 128 dan 512 bit. Banyak istilah berbeda telah digunakan untuk string keluaran. Diantaranya adalah hash, nilai hash, dan pesannya intisari. Fungsi hash diharapkan sangan efisien.

MD2 (Message Digest 2) merupakan algoritma hashing kriptografi dipublikasikan ditahun 1989 dan digunakan untuk menghasilkan intisari pesan 128 bit dengan menggunakan fungsi kompresi 18 putaran. MD2 adalah diketahui rentan terhadap serangan preimage kompleksitas waktu setara dengan 2104 aplikasi fungsi kompres[1]. Karenanya dalam kata-kata penulis MD2, MD2 tidak dapat lagi dianggap sebagai fungsi hash satu arah yang aman.

Berdasarkan penelitian yang dilakukan oleh Cendikia pada tahun 2018 yang berjudul "Digital Signature Dengan Algoritma Sha-1 Dan Rsa Sebagai Autentikasi". Disimpulkan bahwa hasil algoritma hashing dapat digunakan dengan baik untuk membuat tanda tangan digital ataupun digital signature pada file pdf [2]. Aplikasi ini juga dapat digunakan untuk autentikasi pada Surat Keterangan Pendamping Ijazah (SKPI) yang dapat diandalkan untuk mencegah perbuatan memanipulasi dan memodifikasi data file Surat Keterangan Pendamping Ijazah tersebut. Dengan menggunakan aplikasi ini maka pihak ataupun instansi terkait tidak perlu merasa khawatir terhadap tindakan memanipulasi atau memodifikasi data dari file Surat Keterangan Pendamping Ijazah tersebut.

Berdasarkan penelitian yang dilakukan Prosiding Seminar Nasional Riset Information Science (SENARIS) pada tahun 2019 yang berjudul "Analisa Algoritma Sha-256 Untuk Mendeteksi Orisinalitas Citra Digital". Disimpulkan bahwa hasil algoritma SHA-256 dapat digunakan untuk membangkitkan identitas file citra digital yang digunakan untuk mendeteksi orisinalitas citra hasil pemindaian file ijazah ataupun transkrip nilai [3]. Algoritma SHA-256 ini dapat mendeteksi perubahan yang sangat kecil pada citra digital, bahkan perubahan yang terjadi hanya satu piksel saja maka akan menghasilkan perbedaan yang sangat signifikan pada nilai hashnya. Nilai hash ini dapat dimanfaatkan sebagai identitas dari file citra digital karena bersifat unik dan tidak memungkinkan file citra digital yang berbeda memiliki nilai hash yang sama.

Berdasarkan penelitian yang dilakukan oleh Budi K. Hutasuhut, Syahril Efendi, dan Zakarias Situmorang pada tahun 2019 yang berjudul "Digital Signature Untuk Menjaga Keaslian Data Dengan Algoritma Md5 Dan Algoritma Rsa". Disimpulkan bahwa hasil nilai hash dapat digunakan untuk memberikan keamanan terhadap keamanan atau orisinalitas dari suatu data sehingga orang yang menerima data tersebut dapat terhindar dari data yang sudah dimodifikasi ataupun data yang sudah dimanipulasi[4]. Dengan perubahan sedikit saja pada suatu data maka akan mengubah nilai hash yang dihasilkan secara signifikan. Digital signature dapat dibangkitkan dengan kombinasi algoritma MD5 dan algoritma RSA, dengan kombinasi kedua algoritma tersebut maka akan memberikan tingkat keamanan yang sangat tinggi.

Berdasarkan penelitian yang dilakukan oleh Sandeep K V, Dr.Sayed Abdulhayan pada tahun 2020 yang berjudul "Implementation Of Data Integrity Using Md5 Dan Md2 Algorithms In Iot Devices". Disimpulkan bahwa hasil integritas data sangat penting untuk memastikan bahwa keaslian sebuah data dapat diketahui. Beberapa algoritma dapat digunakan untuk menjaga integritas data seperti MD2, MD5, AES, DES, tetapi penggunaan MD2 dan MD5 lebih mudah untuk menghasilkan nilai hash [5]. Algoritma ini digunakan untuk membangkitkan identitas sebuah file, kemudian file tersebut ditransmisikan melalui jaringan, jika nilai hash yang dihasilkan dari file setelah ditransmisikan sama maka dapat disimpulkan bahwa file tersebut tidak rusak. Jika nilai hasnya berbeda maka dapat disimpulkan bahwa file tersebut telah rusak atau dirusak oleh penyusup selama proses transmisi file.

Berdasarkan penelitian yang dilakukan oleh Winda Wulan Sari pada tahun 2021 yang berjudul "Implementasi Metode Md2 Untuk Otentikasi Hasil Scan Citra Ijazah". Disimpulkan bahwa autentikasi hasil pemindaian citra ijazah dapat dilakukan menggunakan algoritma MD2 dan menghasilkan sebuah tanda tangan digital atau digital signature dalam bentuk nilai hash [6]. Dengan menggunakan algoritma ini maka didapatkan nilai hash, di mana jika terjadi perubahan terhadap nilai piksel pada citra tersebut maka akan menghasilkan nilai hash yang berbeda pula. Dengan membandingkan kedua nilai hash tersebut maka dapat diketahui apakah citra sudah dimodifikasi atau belum.

Dari penjabaran di atas, penulis telah mengumpulkan beberapa jurnal dari berbagai sumber yang berkaitan dengan permasalahan yang sedang penulis bahas. Oleh sebab itu penelitian ini bertujuan memudahkan dan mempercepat untuk menghapus file dokumen yang identitas sama tanpa harus membuka dan membaca isi file tersebut.

Jurnal Kajian Ilmiah Teknologi Informasi dan Komputer
ISSN 2962-9055 (Media Online)
Vol 3, No 1, November 2024
Hal 1-8
https://journal.grahamitra.id/index.php/jutik

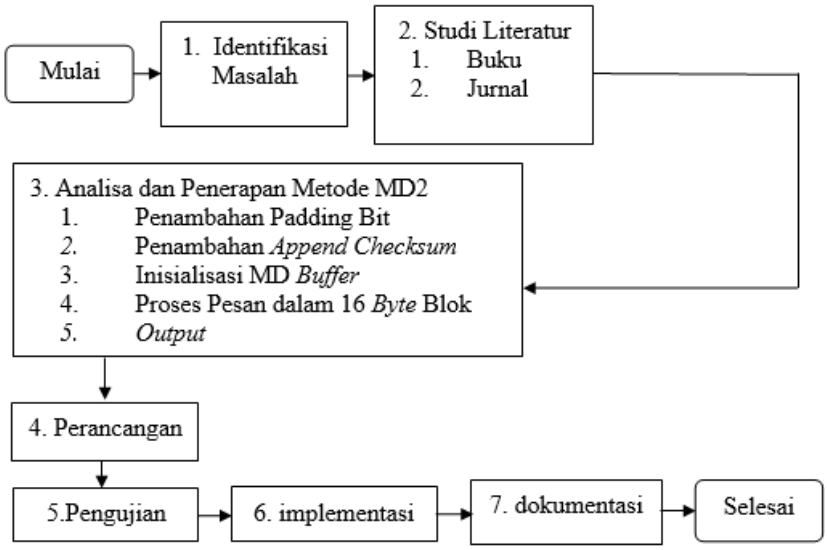
2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Metodologi penelitian langkah untuk melakukan sesuatu atau sistem panduan untuk memecahkan persoalan dalam menyelesaikan penelitian. Langkah ini sangat berguna untuk melakukan dan dapat mempermudah penelitian, sebagai penulis tentunya akan menganalisa topik permasalahan yang akan diteliti. Untuk menganalisa penelitian akan dijelaskan sebagaimana proses pengambilan pengumpulan data-data yang diperlukannya untuk penyelesain penelitian. Penulis mencari jurnal-jurnal dan buku pendukung untuk mengumpulkan dan mendapatkan data-data yang tepat.

Untuk menyelesaikan masalah dalam dokumen identitas file ini penulis akan melakukan perancangan suatu aplikasi yang berbasis komputisasi yaitu perancangan aplikasi pembangkit identitas file dokumen dengan menggunakan bahasa pemrograman Microsoft Visual Basic dengan menerapkan metode algoritma MD2. Penulis mengharapkan dapat memberikan hasil yang maksimal dalam perancangan aplikasi pembangkit identitas file dokumen menerapkan algoritma fungsi hash MD2.

Data yang penulis peroleh dari jurnal-jurnal atau buku-buku berupa data kriteria dan alternatif yang telah diposting atau ditulis oleh penulis lain. Setelah semua data yang diperoleh sudah tersedia, maka tahap selanjutnya penulis akan melakukan studi literatur, guna untuk mendapatkan referensi untuk mendukung keberlangsungan penellitian ini. Berikut ini merupakan tahapan sederhana dalam proses pengumpulan data hingga menyelesaikan masalah ini.. Adapun tahapan penelitian yang di gunakan oleh penulis dalam penelitian ini dapat dilihat pada gambar 1 di bawah ini :



Gambar 1. Tahapan Penelitian

Adapun keterangan dari tahapan penelitian yang terdapat pada Gambar 1 di atas adalah sebagai berikut:

1. Tahap Identifikasi Masalah
   Tahap ini merupakan suatu proses dalam menguraikan atau mencari solusi dengan permasahan dalam identitas file dokumen.
2. Tahap Studi Literatur
   Tahapan ini dilakukan sebagai sumber referensi terhadap objek yang diteliti, dengan membaca buku dan jurnal membantu penulis untuk lebih mudah dalam menyelesaikankan penelitian ini.
3. Tahap Analisa Dan Penerapan Metode Algoritma MD2
   Pada tahap analisa ini merupakan salah satu proses yang penulis lakukan untuk mengetahui apa yang menjadi sumber masalah identitas file dokumen. Sehingga memberikan hasil yang baik untuk menyelesaikan masalah yang ada, setelah menganalisa permasalahan, maka penulis menerapkan metode algoritma MD2 pada identitas file, penerapan metode ini ditunjukkan untuk menghitung nilai dari setiap alternatif dan kriteria.
4. Tahap Perancangan
   Pada tahap ini, penulis akan mempresentasikan gambaran mengenai pembangkit identitas file dokumen yang akan penulis usulkan, tahapan perancangan pembangkit yang menggunakan metode algoritma MD2 dengan bahasa pemrograman visual basic 2008
5. Tahap Pengujian
   Tahap ini dilakukan untuk menguji hasil yang sudah dikerjakan penulis dengan cara mendesain program.
6. Tahap Implementasi
   Tahapan ini merupakan tahap untuk mengimplementasikan hasil perancangan pembangkit yang sudah dibuat untuk melihat apakah penerapan metode algoritma MD2 dapat menyelesaikan masalah.
7. Tahap Dokumentasi

Tahap dokumentasi ini dilakukan untuk persediaan dokumen jika suatu saat diperlukan

2.2 Identitas file

File dokumen merupakan file teks yang tersusun atas rangkaian berisi teks. Jenis-jenis file yang termasuk dalam kategori ini umunya berisi rangkaian karakter tanpa informasi format visual. Konten file ini biasanya merupakan catatan atau daftar personal, artikel, buku dan lain sebagainya. File teks mirip dengan file yang dihasilkan oleh program pengolah kata yang konten utamanya bersifat tekstual [7].

Identitas file merupakan sebuah nilai yang menggambarkan isi dari file tersebut sehingga nilai dari identitas file ini sangat tergantung dari isinya. Syarat dari identitas file yang baik adalah bersifat unik, yaitu tidak memungkinkan ada nilai identitas yang sama dengan isi file yang berbeda. Dengan adanya sifat yang unik ini maka dapat dijadikan solusi untuk mencari file yang memiliki isi yang sama tanpa harus membuka isi dari file tersebut [8].

2.3 Fungsi Hash

Fungsi hash adalah fungsi matematis yang mengubah nilai input numerik menjadi nilai numerik yang terkompresi yeng bertujuan mengkompresi nilai numerik yang dinputkan. Hashing ini digunakan untuk mewakili file digital, pesan atau entitas apapun ke dalamnya string karakter yang lebih pendek, panjang tetap dan unik dalam file cara perhitungan hash untuk entitas digital akan selalu menjadi sama dan tidak mungkin untuk mengambil digital asli entitas dari string hash-nya. Fungsi hash kriptografi memetakan string (pesan) hampir panjang acak ke string dengan panjang tetap dan pendek, biasanya suatu tempat antara 128 dan 512 bit [9]. Fungsi hash menerima masukan string yang sembarang panjangnya lalu mentransformasikannya menjadi keluaran string memiliki panjang tetap yang berukuran biasanya lebih kecil dari ukuran string semula. Banyak istilah berbeda telah digunakan untuk string keluaran. Diantaranya adalah hash, nilai hash dan pesannya intisari. Fungsi hash diharapkan sangat efisien. Berbeda aplikasi mengharapkan properti yang berbeda dari fungsi hash, tetapi beberapa properti selalu diharapkan.

2.4 Algoritma MD2

Message Digest 2 dirancang pertama kali ditahun 1989 dan dirancang ke komputer berbasis 8-bit. Tentang detail MD2 bisa dilihat di RFC 1319. Meskipun banyak memiliki flaw, bahkan mencapai sekarang masih dipakai MD2 sebagaimana infrastruktur untuk sertifikat RSA. MD2 mengubahkan pesan string dalam kode heksadesimal 32 bit. Secara fisik, dengan bekerja MD2 mengkompresi 128 hash value dari sembarang pesan kedalam blok-blok yang berukuran masing-masing 128 bit (16 byte) kemudian sebuah checksum ditambahkan. Dengan kalkulasi yang sebenarnya, menggunakan sebesar blok 48 byte dan tebal 256 byte yang secara tidak langsung dihasilkan. Apabila semua pesan dipanjangkan dari blok yang telah diproses, pertama fragmen dari blok 48 byte menjadi pesan dari hash value [11].

Di bawah ini akan dijelaskan 4 tahap proses untuk menghasilkan message digest untuk algoritma MD2.

1. Memasukkan Padding byte
   Pesan akan ditambahkan melalui proses padding sehingga panjang pesan tersebut kongruen dengan 0 modulo 16. Maka, pesan akan diperluas sehingga panjangnya merupakan kelipatan dari 16 byte. Proses padding ini selalu dilakukan meskipun panjang pesan awal sebelum dilakukan padding sudah kongruen dengan 0 modulo 16. Padding dilakukan dengan mengikuti : “i” byte dari nilai “i” akan ditambahkan pada pesan sehingga panjang pesan kongruen dengan 0 modulo 16. Maka ukuran padding byte paling sedikit 1 byte sampai 16 byte. Pada tahap ini pesan hasil padding memiliki panjang pesan kelipatan 16 byte. Kita bagi pesan menjadi M[0 . . . N-1] dimana N merupakan kelipatan 16.
2. Memasukkan Checksum
   Sebanyak 16 byte checksum dari pesan akan ditambahkan pada hasil dari tahap sebelumnya. Pada langkah ini digunakan sebuah 256-byte yang dibangkitkan secara acak yang dibuat dengan nilai digit dari pi. Jika S[i] menotasikan untuk elemen ke-i pada tabel.
3. Inisialisasi Penyangga MD
   Sebuah 48-byte penyangga X digunakan untuk menghasilkan message digest, nilai penyangga diinisialisasi dengan nol.
4. Proses pesan dalam blok 16-byte
   Langkah ini menggunakan angka hasil pembangkitan sebanyak 256-byte yang sama yang dihasilkan pada proses 2.

3. HASIL DAN PEMBAHASAN

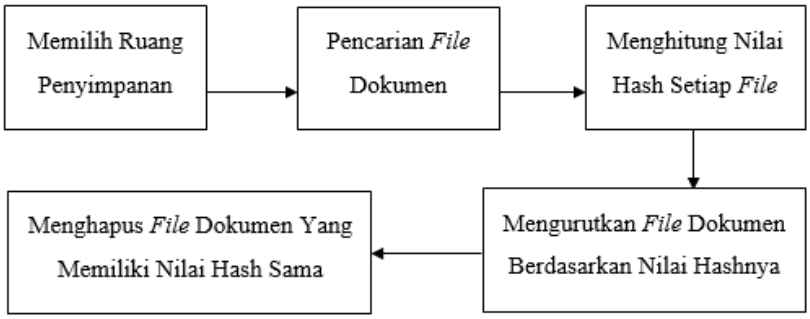
Masalah yang dihadapi oleh pengguna dalam memanajemen file untuk mengoptimalkan ruang penyimpanan adalah kesulitan untuk membedakan antara file yang satu dengan file yang lainnya. Terlebih kesulitan untuk membedakan file dokumen yang satu dengan file dokumen yang lainnya. Pengguna harus membaca semua isi file dokumen tersebut dan membedakan dengan file yang lainnya. Hal ini tentunya sangat tidak memungkinkan, apalagi jika file dokumen tersebut memiliki jumlah halaman yang banyak dan terdapat banyak file dokumen pada media penyimpanan tersebut. Cara

Jurnal Kajian Ilmiah Teknologi Informasi dan Komputer
ISSN 2962-9055 (Media Online)
Vol 3, No 1, November 2024
Hal 1-8
https://journal.grahamitra.id/index.php/jutik

dengan melihat ini dari file dokumen satu persatu tidak mungkin dilakukan, oleh karena itu maka setiap file dokumen perlu diberikan identitas.

Dengan memanfaatkan identitas dari file dokumen tersebut maka pengguna dapat dengan mudah menghapus file dokumen yang ganda atau duplikat. Untuk membangkitkan identitas dari file dokumen ini maka digunakanlah algoritma MD2. Algoritma MD2 ini akan mengubah isi dari file dokumen tersebut menjadi sebuah nilai yang disebut hash value, dimana nilai ini merupakan representasi dari isi file dokumen. Sehingga jika terdapat file dokumen yang memiliki hash value sama dengan file dokumen lainnya, maka dapat dipastikan bahwa file dokumen tersebut merupakan identitas file yang sama.

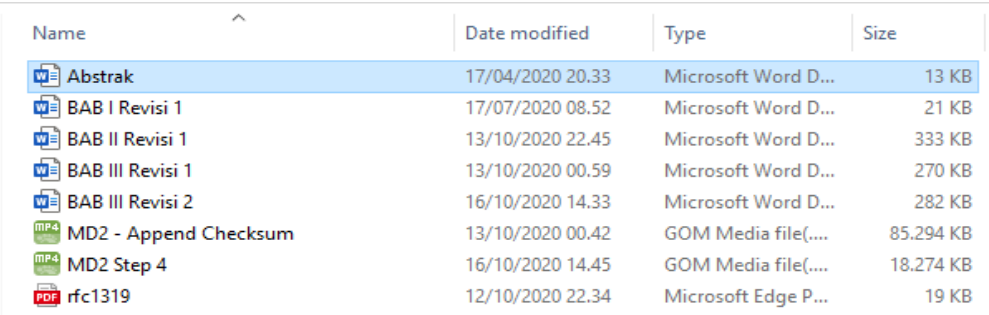
Langkah pertama yang wajib ditempuh adalah melakukan proses pencarian file dokumen pada sebuah media penyimpanan. Setelah semua file dokumen yang ada pada media penyimpanan tersebut ditemukan maka langkah selanjutnya adalah membangkitkan identitas dari setiap file dokumen tersebut. Setelah semua file dokumen memiliki identitas yang dibangkitkan menggunakan algoritma MD2, maka langkah selanjutnya adalah mengurutkan file dokumen tersebut berdasarkan nilai identitasnya. Setelah semua file dokumen tersebut terurut maka pengguna dapat dengan mudah menghapus file dokumen yang memiliki nilai identitas yang sama dan hanya menyisakan satu file dokumen saja. Untuk lebih jelasnya alur dari aplikasi pembangkit identitas file menerapkan algoritma MD2 ini dapat dilihat pada gambar 2. berikut ini:



Gambar 2. Alur Proses Pembangkit Identitas File

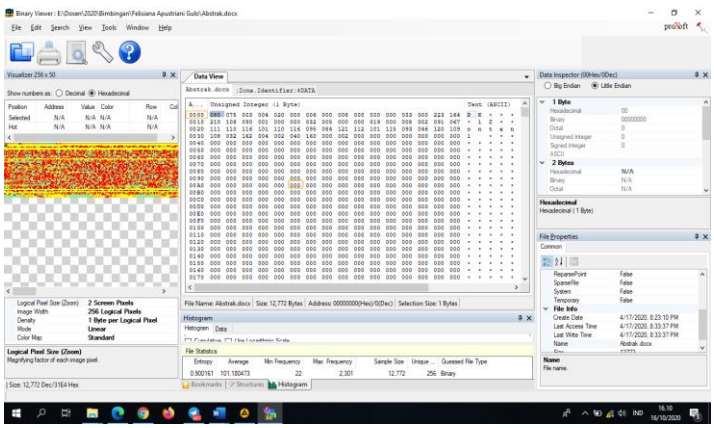
3.1 Penerapan Metode MD2

Metode MD2 memiliki 5 langkah untuk menghasilkan nilai hash. Berikut ini adalah langkah-langkah perhitungan metode MD2 dalam membangkitkan identitas file dokumen berdasarkan sampel data yaitu:



Name	Date modified	Type	Size
Abstrak	17/04/2020 20.33	Microsoft Word D...	13 KB
BAB I Revisi 1	17/07/2020 08.52	Microsoft Word D...	21 KB
BAB II Revisi 1	13/10/2020 22.45	Microsoft Word D...	333 KB
BAB III Revisi 1	13/10/2020 00.59	Microsoft Word D...	270 KB
BAB III Revisi 2	16/10/2020 14.33	Microsoft Word D...	282 KB
MD2 - Append Checksum	13/10/2020 00.42	GOM Media file(....	85.294 KB
MD2 Step 4	16/10/2020 14.45	GOM Media file(....	18.274 KB
rfc1319	12/10/2020 22.34	Microsoft Edge P...	19 KB

Gambar 3. File Dokumen Berekstensi Docx



Gambar 4. File Dokumen Dalam Binary Viewer

Jurnal Kajian Ilmiah Teknologi Informasi dan Komputer
ISSN 2962-9055 (Media Online)
Vol 3, No 1, November 2024
Hal 1-8
https://journal.grahamitra.id/index.php/jutik

Dengan menggunakan bantuan aplikasi binary viewer tersebut maka dapat dilihat nilai integer dari file dokumen tersebut. Untuk mempermudah proses perhitungan secara manual maka diambil sampel dari mulai address 0000 sebanyak 12 byte. Maka data yang didapat disajikan dalam tabel 1. sebagai berikut:

Tabel 1. Data Sample

M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
80	75	3	4	20	0	6	0	8	0	0	0

File dokumen berukuran 150 KB, diambil sampel datanya sebanyak 12 byte, yang dimulai dari address 0000, sehingga dalam penelitian ini sampel data yang akan dianalisa adalah sebanyak 12 byte. Sampel data inilah yang akan dibangkitkan nilai hashnya menggunakan algoritma MD2.

a. Menambahkan Padding Bytes

Penambahan padding bytes dilakukan jika panjang pesan tidak sampai 16 bytes. Pesan akan ditambahkan padding bytes hingga panjang pesan kongruen dengan 0 modulus 16. Artinya pesan tersebut diperpanjang hingga panjang pesan merupakan kelipatan dari 16. Penambahan pesan dilakukan dengan memberikan nilai ke i pada posisi ke i. Berdasarkan sampel data yang ada pada tabel 1 maka akan ditambahkan padding byte dari M12 sampai M15. Maka dapat dilihat pada tabel 2. dibawah ini:

Tabel 2. Penambahan *Padding Bytes*

M12	M13	M14	M15
12	13	14	15

b. Append Checksum

Langkah selanjutnya adalah penambahan Checksum, permutasi acak merupakan susunan yang berbeda satu sama lain yang terbetuk dari sebagian atau seluruh objek. Langkah ini menggunakan permutasi acak 256 byte yang di bangun dari digit pi.

Tabel 3. Permutasi Acak 256 Byte

Indeks	0	1	2	3	4	5	6	7
Nilai pi	41	46	67	201	162	216	124	1
Indeks	8	9	10	11	12	13	14	15
Nilai pi	61	54	84	161	236	240	6	19
….	**….**	**….**	**….**	**….**	**….**	**….**	**….**	**….**
Indeks	248	249	250	251	252	253	254	255
Nilai pi	219	153	141	51	159	17	131	20

Untuk melakukan penambahan *Checksum* di lakukan sebanyak 16 kali sebagai berikut:

Iterasi 0 : I=0, J=0, L=0

Set C to M [I * 16+J]	Set C[J] to S [C xor L]	Set L to C [J]
Set C to M [0 * 16+0]	Set C[0] to S [80 xor 0]	Set L to C [0]
Set C to M [0]	Set C[0] to S [80]	Set L to 128
Set C to 80	Set C[0] to 128	

Iterasi 1 : I=0, J=1, L=128

Set C to M [I * 16+J]	Set C[J] to S [C xor L]	Set L to C [J]
Set C to M [0 * 16+1]	Set C[1] to S [75 xor 128]	Set L to C [1]
Set C to M [1]	Set C[1] to S [203]	Set L to 8
Set C to 75	Set C[1] to 8	

……..

Iterasi 15 : I=0, J=15, L=168

Set C to M [I * 16+J]	Set C[J] to S [C xor L]	Set L to C [J]
Set C to M [0 * 16+15]	Set C[15] to S [15 xor 168]	Set L to C [15]
Set C to M [15]	Set C[15] to S [167]	Set L to 58
Set C to 15	Set C[15] to 58	

Berikut adalah hasil dari perhitungan 16 iterasi yang disajikan dalam tabel 4 berikut ini :

Tabel 4. Hasil Penambahan Checksum

Jurnal Kajian Ilmiah Teknologi Informasi dan Komputer
ISSN 2962-9055 (Media Online)
Vol 3, No 1, November 2024
Hal 1-8
https://journal.grahamitra.id/index.php/jutik

M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
80	75	3	4	20	0	6	0	8	0	0	0	12	13	14	15
M16	M17	M18	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31
128	8	161	81	178	13	161	71	33	155	105	94	93	128	168	58

c. Inisialisasi MD Buffer

Sebuah Buffer atau penyangga X yang berukuran 48 Bytes digunakan untuk menghitung message digest. Buffer atau penyangga ini diinisialisasikan atau diberi nilai awal dengan nilai 0. Untuk hasil inisialisasi MD buffer dapat dilihat pada tabel 5. berikut ini:

Tabel 5. Inisialisasi MD Buffer

X0	X1	X2	X3	X4	X5	X6	X7
0	0	0	0	0	0	0	0
X8	X9	X10	X11	X12	X13	X14	X15
0	0	0	0	0	0	0	0
X16	X17	X18	X19	X20	X21	X22	X23
0	0	0	0	0	0	0	0
X24	X25	X26	X27	X28	X29	X30	X31
0	0	0	0	0	0	0	0
X32	X33	X34	X35	X36	X37	X38	X39
0	0	0	0	0	0	0	0
X40	X41	X42	X43	X44	X45	X46	X47
0	0	0	0	0	0	0	0

d. Memproses Pesan dalam 16 Bytes Blok

Untuk memproses pesan dalam 16 blok, maka lakukan subsitusi dari blok i ke blok x dengan jumlah iterasi sebanyak 16 kali, dapat dilihat pada perhitungan dibawah ini:

Iterasi 0 : I=0, J=0

Set X [16+J] to M [I * 16+J]	Set X [32+J] to (X [16+J] xor X[J])
Set X [16+0] to M [0 * 16+0]	Set X [32+0] to (X [16+0] xor X[0])
Set X[16] to M [0]	Set X [32] to (X [16] xor X[0])
Set X[16] to 80	Set X [32] to (80 xor 0)
	Set X [32] to 80

Iterasi 1 : I=0, J=1

Set X [16+J] to M [I * 16+J]	Set X [32+J] to (X [16+J] xor X[J])
Set X [16+1] to M [0 * 16+1]	Set X [32+1] to (X [16+1] xor X[1])
Set X[17] to M [1]	Set X [33] to (X [17] xor X[1])
Set X[17] to 75	Set X [33] to (75 xor 0)
	Set X [33] to 75

……

Iterasi 15 : I=0, J=15

Set X [16+J] to M [I * 16+J]	Set X [32+J] to (X [16+J] xor X[J])
Set X [16+15] to M [0 * 16+15]	Set X [32+15] to (X [16+15] xor X[15])
Set X[31] to M [15]	Set X [47] to (X [31] xor X[15])
Set X[31] to 15	Set X [47] to (15 xor 0)
	Set X [47] to 15

Tabel 6. Hasil Proses 16 *Bytes Word*

X0	X1	X2	X3	X4	X5	X6	X7
0	0	0	0	0	0	0	0
X8	X9	X10	X11	X12	X13	X14	X15
0	0	0	0	0	0	0	0
X16	X17	X18	X19	X20	X21	X22	X23
80	75	3	4	20	0	6	0
X24	X25	X26	X27	X28	X29	X30	X31
8	0	0	0	12	13	14	15
X32	X33	X34	X35	X36	X37	X38	X39
80	75	3	4	20	0	6	0
X40	X41	X42	X43	X44	X45	X46	X47
8	0	0	0	12	13	14	15

Maka hasil outputnya

Tabel 7. *Output*

X0	X1	X2	X3	X4	X5	X6	X7
41	66	121	252	159	15	19	243
X8	X9	X10	X11	X12	X13	X14	X15
180	133	113	184	211	139	101	165
X16	X17	X18	X19	X20	X21	X22	X23
1	101	166	171	218	213	101	165
X24	X25	X26	X27	X28	X29	X30	X31
89	62	252	159	3	196	28	179
X32	X33	X34	X35	X36	X37	X38	X39
210	222	31	206	218	214	101	165
X40	X41	X42	X43	X44	X45	X46	X47
89	62	238	75	133	124	146	42

Output dari algoritma MD2 di ambil dari X0 sampai X15 yaitu 41 66 121 252 159 15 19 243 180 133 113 184 211 139 101 165 dalam bilangan desimal, jika diubah kedalam bilangan heksadesimal maka hasilnya adalah sebagai berikut: 294279FC9F0F13F3B48571B8D38B65A5. *Output* atau nilai *hash* yang dihasilkan oleh algoritma MD2 berukuran 128 bit yang terdiri dari 32 bilangan heksadesimal. Nilai *hash* ini sangat tergantung dari isi *file* dokumen, jika terjadi perubahan sedikit saja pada isi *file* dokumen tersebut, maka akan menghasilkan perubahan yang sangat signifikan pada nilai *hash*nya. Tidak memungkinkan isi dua buah *file* dokumen yang berbeda tetapi memiliki nilai *hash* yang sama, sehingga nilai *hash* dari algoritma MD2 dapat digunakan sebagai identitas *file* dokumen. Nilai inilah yang menjadi representasi isi dari *file* dokumen sehingga jika ada dua atau lebih *fille* dokumen yang memiliki nilai *hash* yang sama mengindikasikan bahwa *file* dokumen tersebut memiliki isi yang sama.

4. KESIMPULAN

Adapun kesimpulan yang dapat penulis uraikan dari hasil penelitian ini dimana Pencarian file dokumen yang identitas sama didalam sebuah media penyimpanan dapat di lakukan dengan memberikan identitas dari setiap file dokumen menggunakan fungsi hash sehingga tidak perlu membuka dan membaca isi file dokumen satu persatu. Algoritma fungsi hash MD2 dapat digunakan untuk merepresentasikan isi dari file dokumen sehingga dapat digunakan untuk membandingkan file dokumen yang identitas sama. *Output* dari algoritma MD2 di ambil dari X0 sampai X15 yaitu 41 66 121 252 159 15 19 243 180 133 113 184 211 139 101 165 dalam bilangan desimal, jika diubah kedalam bilangan heksadesimal maka hasilnya adalah sebagai berikut: 294279FC9F0F13F3B48571B8D38B65A5. *Output* atau nilai *hash* yang dihasilkan oleh algoritma MD2 berukuran 128 bit yang terdiri dari 32 bilangan heksadesimal. Nilai *hash* ini sangat tergantung dari isi *file* dokumen, jika terjadi perubahan sedikit saja pada isi *file* dokumen tersebut, maka akan menghasilkan perubahan yang sangat signifikan pada nilai *hash*nya

REFERENCES

[1] K. V Sandeep S. Abdulhayan, Implementation of Data Integrity using MD5 and MD2 Algorithms in IoT Devices, 2020.
[2] Sugiyatno P. D. Atika, Digital Signature Dengan Algoritma Sha-1 Dan Rsa Sebagai Autentikasi 2018.
[3] I. Saputra S. D. Nasution, Analisa Algoritma SHA-256 Untuk Mendeteksi Orisinalitas Citra Digital September, 2019
[4] B. K. Hutasuhut, S. Efendi Z. Situmorang Digital Signature untuk Menjaga Keaslian Data dengan Algoritma MD5 dan Algoritma RSA, 2019.
[5] S. (Vivesvaraya T. U. KVS. (Vivesvaraya T. U. Abdulhayan, Implementation of Data Integrity using MD5 and MD2 2020.
[6] Wi. (Universitas B. D. Wulan Sari, Implementasi Metode Base64 Untuk Otentikasi Hasil Scan Citra Ijazah, 2021.
[7] RusdiantA. Qashlim,「Implementasi Algoritma Md5 Untuk Keamanan Dokumen, J. Ilm. Ilmu Komput., 2016.
[8] S. J. Patil, N. P. Jagtap, S. H. Rajput R. B. Sangore,A Duplicate File Finder System.
[9] M. Huda, keamanan informasi. Surabaya: CV. Garuda Mas Sejahtera, 2020.
[10] D. A. Wijaya, Mengenal Bitcoin dan Cryptocurrency. Puspantara, 2016.
[11] H. Mukhtar, Kriptografi Untuk Keamanan Data. Deepublish, 2018.
[12] F. Muller, International Conference on the Theory and Application of Cryptology and Information Security, 2004
[13] A. Nugroho, Unified Modeling Language (UML), , 2009.
[14] K. Darmayuda,Pemrograman Aplikasi Database dengan Microsoft Visual Basic. Net 2008」, 2008.
[15] W. W. Sari, Implementasi Metode MD2 Untuk Otentikasi Hasil Scan Citra Ijazah」, Resolusi Rekayasa Tek. Inform. dan Inf., 2021.
[16] D. P. Precilia A. Izzuddin, Aplikasi Tanda Tangan Digital (Digital Signature) Menggunakan Algoritma Message Digest 5 (MD5), 2016.