PENAMBAHAN ASPEK INTEGRITAS INFORMASI PADA MODEL ENKRIPSI *LIGHT WEIGHT PARALLEL ENCRYPTION WITH DIGIT ARITHMETIC OF COVERTEXT* MENGGUNAKAN ALGORITMA *SHA-1*

Dameria Naomi Simatupang [1], Eka Ardhianto [2]
[1] Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank
[2] Program Studi Teknologi Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank
Jalan Tri Lomba Juang No.1 Semarang, Indonesia
damerianaomisimatupang@mhs.unisbank.ac.id

ABSTRAK

Keamanan informasi menjadi aspek penting dalam era digital, terutama dalam perlindungan dokumen elektronik. Penelitian ini bertujuan untuk meningkatkan integritas dokumen dengan mengembangkan model enkripsi *Light Weight Parallel Encryption with Digit Arithmetic of Covertext* (LWPDAC) menggunakan algoritma SHA-1. Model sebelumnya, seperti *New PDAC* dan *Parallel Encryption with Covertext* (PECT), masih memiliki kelemahan dalam aspek integritas data. Studi ini mengusulkan skema integrasi SHA-1 pada LWPDAC untuk meningkatkan ketahanan terhadap serangan, *Collisions Resistance*, serta memastikan integritas pada informasi. Metode penelitian yang digunakan meliputi studi literatur, perancangan model, implementasi, dan testing, serta evaluasi berbasis entropi dan pengujian *collision resistance*. Hasil pengujian menunjukkan bahwa nilai entropi meningkat menjadi 3,9391 yang lebih baik dari sebelumnya yaitu 3,84817, dengan peningkatan keamanan sebesar 61,08%. Sementara uji signifikansi menggunakan *Mann-Whitney* membuktikan bahwa integrasi SHA-1 secara signifikan memperbaiki keamanan data. Dengan demikian skema usulan ini memberikan kontribusi dalam pengembangan sistem enkripsi yang lebih aman bagi dokumen digital.

Kata kunci : *Keamanan data, enkripsi, dekripsi, SHA-1, integritas dokumen, PDAC*

1. PENDAHULUAN

Data sangat penting bagi banyak bisnis [1]. Data yang bersifat rahasia, dan data pribadi harus dijaga supaya tidak disalahgunakan [2].

Karena sifat data yang rahasia, diperlukan mekanisme khusus untuk mengamankannya. Badan Siber dan Sandi Negara (BSSN) mencatat 207 kasus pembobolan data di Indonesia pada tahun 2023, dengan 55% dari kasus tersebut terjadi pada administrasi pemerintahan, sektor keuangan, transportasi, teknologi informasi, dan lainnya juga terdampak, meskipun dengan jumlah yang lebih kecil [3].

Meskipun awalnya didefinisikan sebagai ilmu tentang menyembunyikan pesan, kriptografi saat ini didefinisikan sebagai ilmu yang mempelajari teknik matematika yang digunakan untuk menjaga keamanan informasi [4].

Namun, menurut [5], kriptografi digunakan untuk memastikan bahwa data yang dikirim dari satu tempat ke tempat lain tetap aman. Keamanan data menjadi sangat penting dalam menjaga kerahasiaan dan integritas informasi, terutama dalam era digital saat ini. Pendekatan yang tepat diperlukan untuk menjaga data sensitif dari akses yang tidak sah, manipulasi, atau penyalahgunaan.

Dalam penelitian ini, fungsi *hash SHA-1* digunakan untuk menjaga integritas informasi. Salah satu cara untuk mendapatkan jaminan integritas informasi adalah menggunakan fungsi *hash* [6].

Nilai *hash* dari setiap dokumen digital akan menjadi berbeda, jadi nilai *hash* dari satu dokumen digital juga akan berubah jika dokumen digital lainnya diubah [6].

Fungsi hash akan menghasilkan sebuah nilai untuk mengidentifikasi integritas informasi sebagai message digest, atau juga dikenal sebagai "sidik pesan".

Salah satu fungsi *hash* yang bekerja dalam satu arah adalah algoritma *Secure Hash Algorithm (SHA)*. Algoritma *SHA-1* bekerja dengan menerima *input* berupa pesan berukuran acak dan menghasilkan pesan terbesar dengan panjang 160 bit [6].

Penelitian terdahulu tentang PDAC sudah melakukan perkembangan yang signifikan sejak pertama kali diperkenalkan. Pada tahun 2013, model PDAC telah dikembangkan sebagai metode enkripsi yang berbasis steganografi yang mengintegrasikan informasi ke dalam bentuk *covertext* untuk meningkatkan keamanan data [7].

Model ini menyesuaikan melalui model ECR yang berfokus pada pengacakan karakter untuk menjaga kerahasiaan informasi. Pada tahun 2015, telah diperkenalkan model *New* PDAC yang yang dioptimalkan untuk keamanan data *cloud*. Model ini telah menggabungkan teknik enkripsi dan steganografi untuk melindungi data kemudian meningkatkan privasi dan memperkuat keamanan penyimpanan di *platform cloud* [8].

Kemudian, pada tahun 2020, telah dikembangkan PECT untuk memperkenalkan penggunaan *covertext* secara *parallel* dan untuk meningkatkan efisiensi proses enkripsi dengan memungkinkan enkripsi simultan, membuatnya lebih ideal untuk aplikasi yang memerlukan keamanan tinggi [9]

Pada tahun 2023, telah diperkenalkan *fuzzy logic*. Model ini mengintegrasikan logika *fuzzy* untuk menyesuaikan parameter enkripsi secara dinamis berdasarkan kondisi data dan tingkat keamanan yang diperlukan. Dengan pendekatan ini, efisiensi enkripsi dan kerahasiaan data ditingkatkan [10].

Selanjutnya pada tahun 2024, telah mengembangkan LWPDAC. Model ini dirancang sebagai versi ringan dari PDAC yang mengurangi konsumsi sumberdata komputasi dan membuatnya cocok untuk perangkat dengan keterbatasan memori dan daya pemrosesan. Juga sudah mengeksplorasi mekanisme baru dalam PDAC untuk meningkatkan efisiensi dan keamanan. Dengan memperkenalkan Teknik enkripsi yang lebih kompleks, penelitian ini bertujuan untuk mengurangi waktu pemrosesan [11].

Berdasarkan berbagai model enkripsi yang telah dikembangkan, algoritma LWPDAC dipilih karena keunggulannya dalam memberikan skema keamanan yang lebih efisien dengan ukuran *chipertext* yang lebih kecil dibandingkan dengan model lainnya. LWPDAC telah menawarkan efisiensi komputasi yang lebih baik, dapat diterapkan pada perangkat dengan keterbatasan sumber daya. Selain itu, algoritma ini mampu menjaga tingkat privasi yang tinggi yang mempertimbangkan kebutuhan akan enkripsi yang efisien dan aman diberbagai platform, algorima ini menjadi pilihan yang tepat untuk penelitian ini.

Algoritma *Light Weight Parallel Encryption with Digit Arithmetic and Covertext* (LWPDAC) adalah pengembangan algoritma *Parallel Encryption with Digit Arithmetic and Covertext* (PDAC). Algorima LWPDAC saat ini telah memberikan skema keamaman yang baik, dan capaian ukuran *cipherteks* yang lebih kecil. Meskipun LWPDAC saat ini telah memberikan jaminan keamanan informasi yang baik, namun aspek integritas informasi masih belum tertuang dalam skema pengamanannya.

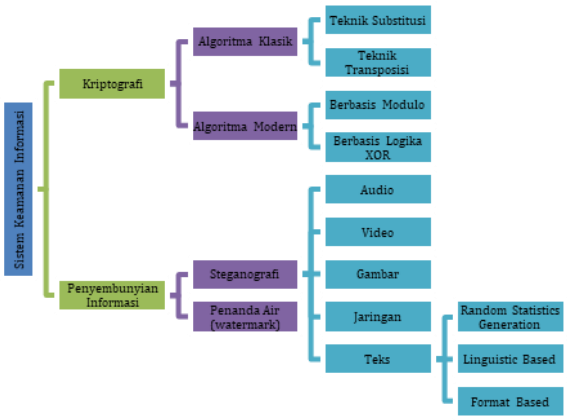
Penelitian ini bertujuan untuk memberikan jaminan integritas informasi dengan menggunakan algoritma *hash SHA-1* melalui pengembangan skema proses pada algoritma *LWPDAC*. Sebagai metrik pengukuran digunakan entropi, collision resistance test, dan perhitungan signifikansi hasil yang diperoleh.

2. TINJAUAN PUSTAKA

2.1. Tinjauan Umum Sistem Keamanan Informasi

Steganografi dan kriptografi berbeda. Bahasa Yunani "steganografi" berasal dari dua kata, *"steganos"* yang berarti tersembunyi dan *"graphien"* yang berarti tulisan. Herodotus, seorang sejarahwan Yunani kuno, menggunakan lilin sebagai penutup untuk melapisi pesan tertulis [12].

Ilmu Sistem keamanan informasi dibagi dalan dua cabang, yaitu: Kriptografi dan Penyembunyian Informasi. Dalam bidang Penyembunyian Informasi, terdapat teknik steganografi, dan Penanda air, seperti terlhat pada gambar 1. Secara teknis steganografi terbagi menjadi empat kategori: steganografi gambar, teks, steganografi suara, dan steganografi video, yang bergantung pada sampul media yang digunakan untuk menyematkan pesan [13]; [14].



Gambar 1. Bidang dan Sub Bidang dalam Sistem Keamanan Informasi [14][15][16]

Berdasarkan gambar 1, kriptografi dibagi menjadi dua kelompok berdasarkan alat yang digunakan: kriptografi klasik dan kriptografi modern [14].

Kriptografi klasik menggunakan alat manual seperti pensil dan kertas, dan objeknya adalah huruf dan angka. Kriptografi modern menggunakan teknik transposisi dan substitusi. *Cipher Caesar*, *Cipher Affine*, *Playfair*, *Vigenere*, dan *Enigma* adalah beberapa contohnya. Kriptografi kontemporer memanfaatkan komputer dalam prosesnya. Karakter dalam rangkaian bit biner digunakan. Kriptografi kontemporer menggunakan dua proses: berbasis *XOR* dan berbasis *modulo*. Beberapa algoritma berbasis *modulo* adalah *RSA*, *RC4*, *Elgamal*, *Knapsack*, dan *Rabin-Miller*. Algoritma berdasarkan *XOR* adalah *A5*, *DES*, *Triple DES*, *GOST*, *RC5*, *AES*, dan *Rijndael*.

SHA-1 adalah fungsi *hash* satu arah yang biasanya disebut sebagai *message digest* atau *digital fingerprint* [17].
Fungsi ini mengubah input dengan panjang variabel menjadi nilai *hash* tetap sepanjang 160 bit. Oleh karena itu, *SHA-1* sangat penting untuk memastikan integritas data dan autentikasi data dengan mendeteksi perubahan pada dokumen digital.

2.2. Tinjauan Umum Aspek-Aspek Keamanan Informasi

Performa sistem kriptografi dan steganografi dapat diukur dari beberapa aspek. Kriptografi dan steganografi memiliki kesamaan aspek yang dipandang sebagai aspek umum, yaitu: aspek kerahasiaan. Aspek lain yang melekat pada kriptografi dan steganografi merupakan aspek yang dipandang sebagai aspek unik [18].

Tabel 1 memperlihatkan aspek-aspek yang terdapat pada kriptografi dan steganografi.

Tabel 1.Aspek Aspek Keamanan Informasi [19]

Aspek	Kriptografi	Steganografi
Aspek Umum	Kerahasiaan (*confidentiality*)	Kerahasiaan (*confidentiality*)
Aspek Unik	Otentikasi Integritas	*Impercibility Capacity Robustness*

Kriptografi menawarkan beberapa layanan keamanan, seperti kerahasiaan, otentikasi, dan integritas data, sebagai bagian dari keamanan informasi [18].

Keamanan informasi dapat diterapkan pada informasi sensitif menggunakan elemen seperti kerahasiaan data, otentikasi pengguna yang valid, dan integritas data [20].

Karena penyusup dapat mencegat pesan, pengirim data dan hanya penerima yang dituju harus memahami konteks pesan yang sebenarnya. Di sini, pesan teks biasa harus dienkripsi sehingga penyusup tidak dapat mendekripsi pesan yang disadap. Otentikasi dan validasi adalah istilah yang biasanya disebut sebagai kerahasiaan data atau kerahasiaan data, yang berkaitan dengan keinginan pengirim dan penerima untuk memverifikasi identitas satu sama lain melalui saluran komunikasi [20].

Kerahasiaan, impercibility, kapasitas, dan kekuatan adalah elemen steganografi yang paling penting. Kerahasiaan data dalam setganografi menunjukkan betapa sulitnya mengetahui keberadaan pesan tersembunyi. Karena tingkat kualitas steganografi diuji dengan menggunakan PSNR, percepatan sangat penting. Kapasitas adalah jumlah informasi atau semua data yang dapat disembunyikan di media penutup. Ketahanan adalah ketahanan teknik steganografi terhadap perubahan atau kerusakan media stego [21].

2.3. *Entropy*

Shannon adalah orang pertama yang mengusulkan entropi sebagai ukuran klasik ketidakpastian [22].

Kriptografi yang kuat, yang bergantung pada entropi yang tinggi, dan tidak dapat diprediksi, diperlukan untuk mengamankan informasi didalam perjalanannya baik melewati internet atau jaraingan yang lain [23].

Nilai entropi yang tinggi menunjukkan keacakan yang sebenarnya dalam teori informasi [23].

Dalam teori kriptografi, "entropi informasi" adalah ukuran jumlah informasi yang tidak teratur yang terkandung dalam sebuah pesan. Untuk menunjukkan tingkat keacakan informasi, entropi digunakan dalam satuan [24].

Histogram digunakan dalam analisis kriptografi untuk menghitung jumlah karakter yang ada dalam informasi dan memperkirakan kemungkinan setiap karakter akan didistribusikan [25].

Nilai entropi ideal untuk data terenkripsi adalah 8, dengan jumlah anggota semesta dalam 256 karater ASCII. Penjumlahan semua distribusi karakter probabilitas yang mungkin adalah konsep utama dibalik entropi [25].

Nilai entropi yang mendekati 8 menunjukkan bahwa sistem enkripsi yang dirancang dengan baik dan informasinya harus aman dari penyusup [25];[26].

Pada penelitian ini menggunakan nilai entropi optimal 96 karakter ASCII pada nilai entropi 6,58.

Nilai entropi informasi terenkripsi H(m) dapat dihitung dengan persamaan (2.23) [4];[24];[25]**.** Nilai keseluruhan data ditunjukkan oleh $P(m_i)$, dan nilai probabilitas simbol (karakter) m_i ditunjukkan oleh $P(m_i)$.

$$H(m) = \sum_{i=0}^{n-1} \quad P(m_i) log_2 \frac{1}{P(m_i)} \qquad (1)$$

2.4. Perkembangan Model Enkripsi PDAC

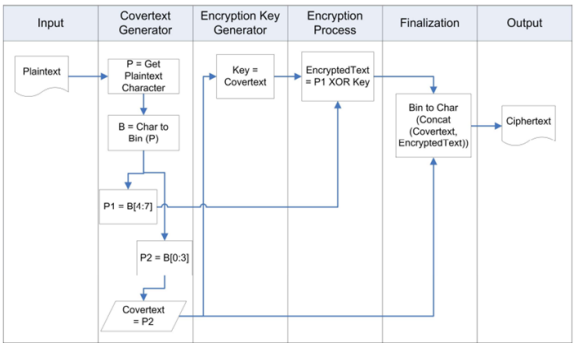
PDAC pertama kali muncul pada tahun 2013. Model pengamanan dokumen PDAC menggabungkan steganografi berbasis format teks dengan enkripsi berbasis logika XOR [27].

Evolusi dalam PDAC menjadi *New PDAC* pada tahun 2015 [8].

Pada tahun 2020, PDAC berubah menjadi *Enkripsi Parallel* dengan *Covertext (PECT)* [28]. Saat ini PDAC telah dikembangkan menjadi LWPDAC.

Metode PDAC yang baru diusulkan, *LWPDAC (Light Weight of Parallel Encryption With Digit Arithmetic Of Covertext Encryption)*, akan dirancang pada tahun 2024 [11].

Metode ini menghasilkan chipertext yang lebih ringan. Dengan demikian, model LWPDAC akan dirancang untuk mencapai rasio penyimpanan dan pengiriman yang lebih rendah. Gambar 2 adalah proses yang digunakan pada model LWPDAC.

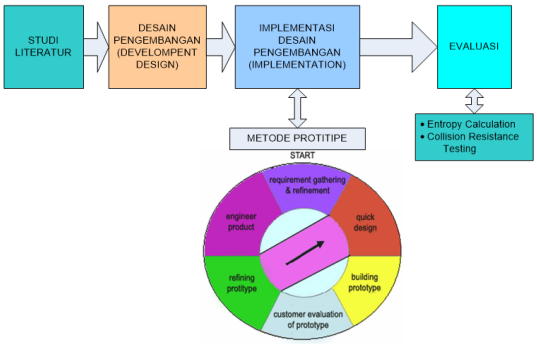


Gambar 2. *Light Weight PDAC* [11]

3. ANALISA DAN RANCANGAN SISTEM

3.1. Kerangka Penelitian

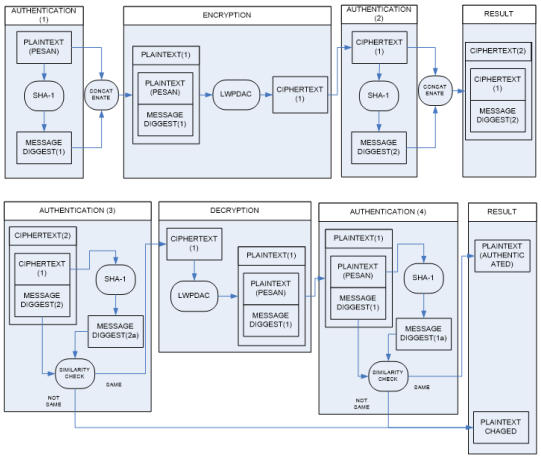
Tujuan dari penelitian ini adalah untuk mengembangkan algoritma PDAC berat ringan melalui empat tahap: penelitian literatur, desain pengembangan, implementasi menggunakan metode *prototyping*, dan evaluasi kinerja berdasarkan entropi dan uji ketahanan cipherteks menggunakan *Collision Resistance Test*.



Gambar 3. Metode Penelitian

Proses metode penelitian dalam skripsi ini dijelaskan sebagai berikut, seperti yang ditunjukkan pada gambar 3, yaitu:

a. Studi Literatur
   Tahap pertama penelitian ini berfokus pada konsep dasar kriptografi, khususnya algoritma berat ringan yang efektif dalam penggunaan sumber daya.
b. Desain Pengembangan
   Pada tahap ini, struktur algoritma ditentukan dengan mengacu pada prinsip dasar algoritma kriptografi yang ringan tetapi kuat secara keamanan. Desain pengembangan juga mempertimbangkan penggunaan blok kriptografi yang efisien untuk menjaga keseimbangan antara performa dan keamanan. Gambar 4 menunjukkan desain pengembangan yang diusulkan sebagai pemecahan masalah untuk penelitian ini.



Gambar 4. Skema Desain Penelitian

c. Implementasi
   Algoritma PDAC berat ringan diterapkan dengan metode prototyping, yang memungkinkan pengembang dan pelanggan berkomunikasi melalui protokol pengembangan perangkat lunak. Ini adalah proses yang terdiri dari beberapa tahap:
   - Mengumpulkan kebutuhan pelanggan tentang produk.
   - Desain cepat, yang berarti membuat sketsa awal sesuai dengan kebutuhan.
   - Pembuatan prototipe sebagai versi awal produk untuk menguji konsep utama.
   - Evaluasi prototipe oleh pelanggan untuk memberikan umpan balik.
   - Penyempurnaan prototipe dengan memperbaiki fitur sesuai dengan umpan balik.
   - Pengembangan produk akhir berdasarkan prototipe yang telah diperbaiki.

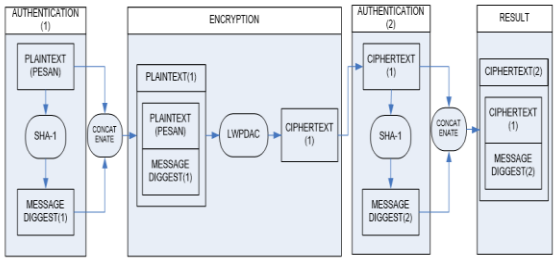
   Dengan menggunakan proses ini, pengembang dapat membangun algoritma secara bertahap, menguji fungsionalitas dasar, dan memperbaikinya secara berulang. Versi awal algoritma dibuat dengan menggunakan bahasa pemrograman seperti *PHP* dan *JavaScript*. Agar algoritma berkembang secara optimal sebelum rilis final, pengujian di setiap tahap sangat penting.
d. Evaluasi
   Evaluasi dalam hasil penelitian ini diguanakn pengukuran entropi, dan pengujian Collision Resistance Test. Secara lebih detil sebagai berikut:
   - Pengukuran entropi
     Alat yang digunakan untuk menghitung tingkat keacakan (*randomness*) dari kode autentikasi yang dibuat. Sulit bagi penyerang untuk menebak pola atau nilai algoritma PDAC dengan entropi yang lebih tinggi. Perhitungan entropi dilakukan pada Alamat website: *https://planetcalc.com/2476/*.
   - Uji *Collision Resistance Testing*
     Pengujian ini menguji kemampuan untuk mencegah dua input menghasilkan output yang sama. Kekuatan algoritma terhadap potensi serangan dinilai melalui simulasi serangan dan analisis hasil. Pengujian ini dilakukan menggunakan website *https://shattered.io/*.

3.2. Desain Penelitian

Desain penelitian ini mempunyai dua proses yaitu , proses enkripsi dan dekripsi. Pada gambar 5 menjelaskan proses enkripsi yang dilakukan dalam beberapa tahap utama yaitu:

a. Verifikasi: Pesan awal diproses dengan algoritma *SHA-1* untuk menghasilkan *Message Digest* (1), yang berfungsi untuk memastikan integritas pesan dan menggabungkannya dengan *plaintext*. *SHA-1* yang dihunakan ialah algoritma *hash* kriptografi yang menggunakan nilai *hash* yaitu 160 bit yang mempunyai 40 karakter.
b. Enkripsi: Algoritma LWPDAC mengenkripsi hasil penggabungan setelah tahap autentikasi, menghasilkan *Ciphertext* (1) sebagai output awal.
c. Validasi: *Ciphertext* (1) diproses kembali dengan *SHA-1* untuk membuat *Message Digest* (2). Data kemudian divalidasi dengan menggabungkan *Message Digest* (2).
d. Hasil Akhir: *Ciphertext* (2) dibuat dengan menggabungkan *Ciphertext* (1) dan *Message Digest* (2). Metode ini meningkatkan keamanan

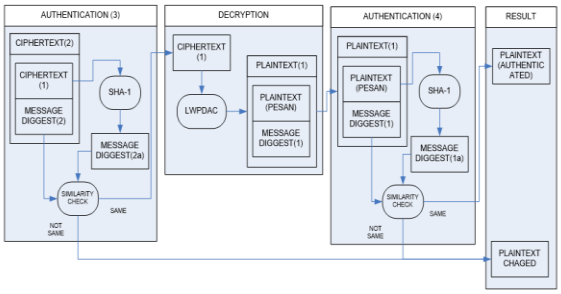
data dengan menggabungkan enkripsi dan autentikasi berbasis *hash.*



Gambar 5. Prosedur Enkripsi

Selanjutnya pada gambar 6 akan menjelaskan tentang proses dekripsi terdiri dari beberapa langkah:

a. Autentikasi: *Message Digest* (2) dibuat dengan memproses *ciphertext* (1) dengan *SHA-1*. *SHA-1* yang dikerjakan ialah algoritma *hash* kriptografi yang menggunakan nilai *hash* yaitu 160 bit yang mempunyai 40 karakter. *Ciphertext* (1) dan hasilnya dibandingkan. Jika tidak sama, autentikasi gagal (*plaintext* dianggap berubah), dan proses dilanjutkan jika semuanya sama.
b. Dekripsi: Algoritma LWPDAC digunakan untuk mendekripsi *chiphertext* (1) dan menghasilkan *plaintext* (1).
c. Validasi: Untuk menghasilkan *Message Digest* (1a), *Plaintext* (1) diproses kembali dengan *SHA-1*. *Plaintext* dianggap autentik jika hasilnya sama; jika berbeda, dianggap telah diubah.
d. Hasil Akhir: *Plaintext* dianggap autentik jika semua proses autentikasi berhasil. Jika ada perbedaan, *plaintext* dianggap tidak valid.



Gambar 6. Prosedur Dekripsi

3.3. Dataset Pengujian

Pengujian ini menggunakan file Kaggle https://www.kaggle.com/datasets/mitusha/email-data set/data , yang berisi 5.846 email dan berukuran 3.44 MB. Tujuan dari file ini adalah untuk menguji seberapa tangguh usulan metode yang baru dalam menjaga keamanan dan integritas data.

Pada dataset ini dilakukan dengan berbagai ukuran dataset untuk menilai kecepatan proses, efisiensi algoritma, dan akurasi hasil dekripsi. Dataset ini juga termasuk metadata *email*, termasuk isi dan kategorinya; metadata ini sering digunakan dalam analisis teks, klasifikasi *email*, dan analisis pola komunikasi.

4. IMPLEMENTASI DAN PEMBAHASAN

4.1. Sampel Data Input

Ada 60 sampel data dari dataset yang diambil secara acak. Kemudian, algoritma LWPDAC dan *SHA-1* digunakan untuk mengenkripsi sampel. Sampel yang digunakan antara 1900 karakter sampai dengan 5000 karakter. *SHA-1* dengan panjang *Message Digest* 40 karakter.

4.2. Algoritma Light Weight PDAC dan SHA-1

Algoritma ini memiliki dua fungsi utama yaitu proses enkripsi dan dekripsi yang berfokus pada peningkatan integritas dokumen menggunakan SHA-1. Fungsi enkripsi melibatkan penggunaan tombol untuk mengubah *plaintext*, yang dapat dibaca, menjadi *chipertext*, yang tidak dapat dibaca. Tombol tersebut memiliki 96 karakter yang secara otomatis dibuat oleh kode *JavaScript*. Potongan *sourcecode* dapat dilihat pada tabel 3.

Tabel 3. *Sourcecode* Enkripsi

Source Code 1

```
$fileContent =
file_get_contents($_FILES["fileInput"]["tmp_name"]);

// Lakukan enkripsi
$encryptedText = "";
$m1Array = array();
$m2Array = array();
$eArray = array();
$m3Array = array();
$convertTextArray = array();
$encryptedBitsArray = array();
$m2dArray = array();
$plainTextArray = array();
$decryptedTextArray = array(); // Gunakan array untuk
menyimpan karakter terdekripsi
$lines = explode("\n", $fileContent); // Pisahkan teks
menjadi baris-baris

foreach ($lines as $line) {
// Lakukan enkripsi untuk setiap baris
$encryptedLine = "";
for ($i = 0; $i < strlen($line); $i++) {
$char = $line[$i];
if (preg_match('/[\r0-9A-Za-z
    • .,!@#$%^&*()_+="|-`{}\[\]:;<>,?~\\/-]/i', $char)) {
 $binaryChar = str_pad(decbin(ord($char)), 8, "0",
STR_PAD_LEFT);

$m1 = $binaryChar[0] . $binaryChar[2] . $binaryChar[4] .
$binaryChar[6];
$m2 = $binaryChar[1] . $binaryChar[3] . $binaryChar[5] .
$binaryChar[7];
$encryptedChar = "";
```

Pada tabel 3 diatas telah menjelaskan proses enkripsi yang diawali dengan membaca *file* yang diunggah pada pengguna menggunakan *file_get_content ()*, lalu memisahkannya berdasarkan baris menggunakan *explode ()*. Kemudian, *Variabel* dan *array* seperti *$m1Array*, *$m2Array*, dan *$encryptedBitsArray* diinisialisasi untuk menyimpan hasil dari berbagai tahapan enkripsi. Setiap karakter dalam baris diverifikasi melalui *regex* menggunakan *preg_match ()* untuk memastikan karakter yang

diizinkan, seperti huruf, angka, dan symbol tertentu. Karakter yang valid lalu dikonversi menjadi kode ASCII menggunakan *ord ()* dan diubah menjadi representasi biner sepanjang 8 bit dengan *decbin ()* dan *str_pad ()*. Hasil biner tersebut selanjutnya dipisah menjadi dua bagian: *$m1*, yang berisi bit genap (0,2,4,6), dan *$m2* yang berisi bit ganjil (1,3,5,7). Proses ini memungkinkan menghasilkan *chipertext* yang lebih aman, dan disimpan dalam *variabel $encryptedText*. Selanjutnya, fungsi Dekripsi akan digunakan untuk mengubah *chipertext*, yang tidak dapat dibaca, menjadi *plaintext*, yang dapat dibaca, dengan menggunakan *key* yang telah dibuat sebelumnya. Dapat dilihat pada *sourcecode* tabel 2 dibawah ini.

Tabel 2. *Sourcecode* dekripsi

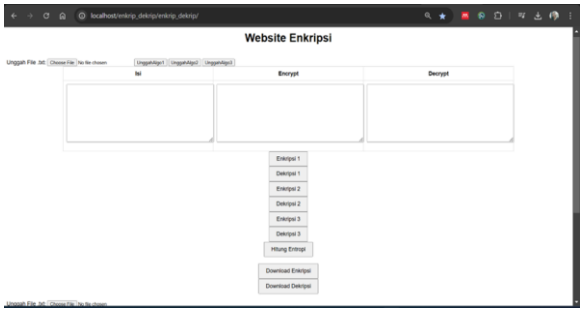
Source Code 2

```
for ($j = 0; $j < strlen($m1); $j++) {
$bit1 = $m1[$j];
$bit2 = $m2[$j];
$result = ($bit1 === $bit2) ? "0" : "1";
$encryptedChar .= $result;
}
$m3 = $m1 . $encryptedChar;
$m1Array[] = $m1;
$m2Array[] = $m2;
$eArray[] = $encryptedChar;
$m3Array[] = $m3;
$convertTextArray[] = $m1;
$encryptedBitsArray[] = $encryptedChar;
$m2d = "";
for ($k = 0; $k < strlen($m1); $k++) {
$bit1 = $m1[$k];
$bit2 = $encryptedChar[$k];
$result = ($bit1 === $bit2) ? "0" : "1";
$m2d .= $result;
}
$m2dArray[] = $m2d;
// Konstruksi plainText
$plainText = "";
$maxLength = min(strlen($m1), strlen($m2));
for ($k = 0; $k < $maxLength; $k++) {
$plainText .= $m1[$k] . $m2[$k];
}
$plainTextArray[] = $plainText;
// Dekripsi karakter
$decryptedChar = chr(bindec($plainText));
// Tambahkan karakter terdekripsi ke dalam array
$decryptedTextArray[] = $decryptedChar;
$encryptedLine .= chr(bindec($m3));
} else {
$encryptedLine .= $char;
}
}
// Gabungkan baris terenkripsi menjadi teks terenkripsi
dengan karakter baris baru
$encryptedText .= $encryptedLine . "\n";
}
```

Pada tabel 2 diatas telah menjelaskan proses dekripsi pada algoritma ini. Proses ini diawali dengan membandingkan setiap bit dari *$m1* dan *$m2* menggunakan operasi XOR. Jika kedua bit sama maka hasilnya 0 dan jika berbeda maka hasilnya 1. Hasil perbandingan ini disimpan dalam variabel *$encryptedChar*. Lalu, Proses XOR dilakukan Kembali menggunakan bit dari *$m1* dan *$encryptedChar* untuk merekonstruksi *$m2* yang asli. Bila hasil XOR sama dengan bit di *$m1* maka hasilnya 0 dan jika berbeda hasilnya 1. Hasil dekripsi ini akan disimpan pada *$m2d*. Kemudian, *$m1* dan *$m2d* digabungkan secara berurutan untuk membentuk Kembali representasi biner 8 bit yang sesuai dengan karakter asli. Konversi biner ke ASCII dilakukan dengan bindec () untuk mengubah biner menjadi decimal dan chr () untuk mengubah decimal menjadi karakter. Hasil dekripsi akhirnya disimpan dalam array *$decryptedTextArray*.

4.3. Implementasi *Interface*

Pada tampilan ini, akan ditampilkan bentuk website enkripsi dan dekripsi di mana pengguna menggunakan file teks untuk memasukkan data. Kemudian, mereka akan masuk ke kolom isi dan memilih algoritma *Light Weight PDAC 3* pada tombol enkripsi 3 untuk masuk ke kolom enkripsi. Kemudian, akan terlihat hasil deskripsi chipertext dengan mengklik tombol dekripsi 3. Tampilan antarmuka yang dienkripsi dan didekripsi dapat dilihat pada gambar 7 dan gambar 8.



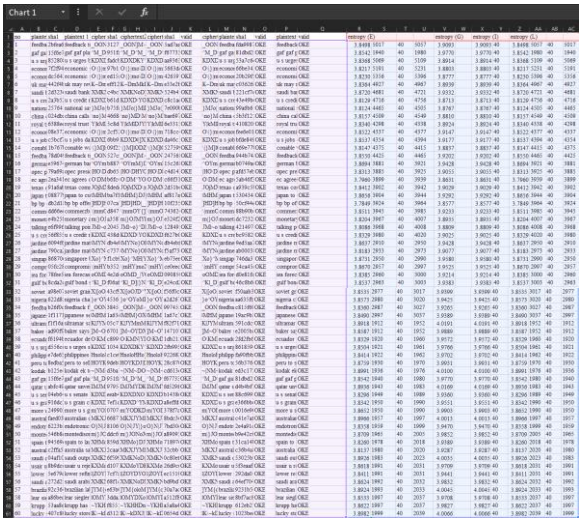
Gambar 7. Tampilan file melakukan enkripsi dan dekripsi.

Gambar 8. Tampilan Enkripsi ke Dekripsi

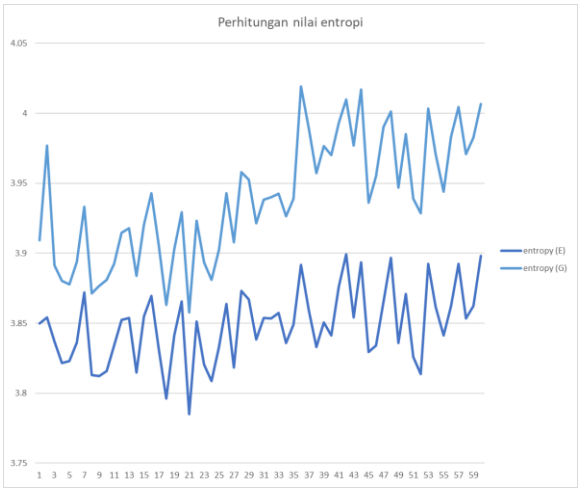
4.4. Hasil dan Pembahasan

Pada bagian ini akan dibahas hasil yang diperoleh dalam percobaan pengamanan informasi menggunakan usulan skema LWPDAC dan SHA-1 dengan menggunakan entropi 96 karakter dengan nilai *entropy* dengan nilai maksimum 6,58. Jika nilai *entropy* menghasilkan nilai lebih tinggi, maka dapat disimpulkan bahwa tingkat keamanan akan semakin lebih baik. Pengujian dilakukan pada tiap *chipertext* yang diperoleh. Kemudian digunakan algoritma *Hash Generator SHA-1* untuk menghasilkan *message digest* dengan panjang message digest 40 karakter, yang digabungkan dengan plainteks untuk menghasilkan

plaintext2. Proses enkripsi menggunakan LWPDAC untuk menghasilkan *chipertext1*. Ciphertext1 kemudian dihitung message digest menggunakan *SHA-1*, dan selanjutnya dilakukan concatenate pada *chipertext1* dan *SHA-1* untuk menghasilkan *chipertext2*. Kemudian dilakukan *Collision Resistenace Test* pada hasil yang diperoleh baik sebelum maupun sesudah enkripsi. Pengujian ini menggunakan 60 sample yang berbeda. Gambar 9 memperlihatkan file pengujian.



Gambar 9. Tampilan 60 sampel pada dataset



Gambar 10. Perbandingann nilai entropi hasil Pengujian.

Pada gambar 10, terlihat hasil grafik tersebut dijelaskan bahwa nilai entropi yang dihasilkan pada 60 data sample, yang menunjukkan bahwa terdapat peningkatan. Setelah enkripsi pertama dengan *SHA-1*, yang disebut *chipertext1*, kemudian dilakukan perhitungan *mean* (rata-rata) entropi sebesar 3,9391 yang lebih baik dari sebelumnya yaitu 3,84817.

Nilai maksimum, dengan nilai *entropy Chipertext1* sebesar 3.8991 dan nilai *entropy Chipertext2* sebesar 4.0191. Perhitungan *entropy* 96 karakter pada sample tersebut dilakukan hasil pengujian yang lebih baik dari 59,26% yaitu sebesar 61,08%.

Perhitungan untuk 96 karakter dengan nilai maksimum 6,58 telah dilakukan dengan menggunakan rumus (1). Perhitungan peningkatan dari pengurangan hasil perhitungan *entropy* 96 karakter *chipertext1* dan *chipertext2* yang menghasilkan total peningkatan sebesar 1,82%.

Berdasarkan perhitungan statistik dilakukan dengan menggunakan alamat *website https://www.socscistatistics.com/tests/mannwhitney/default2.aspx*. Fokusnya adalah untuk mengetahui apakah ada perbedaan yang signifikan antara dua kelompok data yang diuji—dalam hal ini, *entropy ciphertext 1* dan *ciphertext 2*. Dengan kata lain, dapat disimpulkan bahwa algoritma yang digunakan mempengaruhi hasil enkripsi secara signifikan jika hasil uji menunjukkan nilai p yang lebih kecil dari tingkat signifikansi (dengan nilai $\alpha = 0.05$). Dengan kata lain, jika hasilnya signifikan, algoritma tersebut dapat dianggap efektif dalam meningkatkan keamanan *ciphertext* melalui optimasi *entropy*. Seperti yang terlihat pada tabel 4.

Tabel 4. Pengujian signifikan *Mann-Whitney u Test Calculator*

Uji Statistik	Nilai
Nilai U	107.5
Z-score	-8.8807
P-value	< 0.00001
Signifikansi (p < 0.05)	Signifikan

Pada tabel 4 diatas merupakan hasil pengujian yang sudah di lakukan. Hasil pengujian memperlihatkan Nilai U adalah 107.5 dengan menunjukkan bahwa distribusi antara dua kelompok berbeda tetapi tidak ekstrim, nilai Z adalah -8.8807, yang menunjukkan bahwa *sample* 1 memiliki peringkat lebih rendah dari *sample* 2, dan nilai *P-Value* adalah lebih rendah dari 0.00001, yang menunjukkan bahwa hasil sangat signifikan, artinya ada perbedaan yang signifikan antara kedua sampel dikarekanakan p <0.05.

5. KESIMPULAN DAN SARAN

Setelah melakukan penelitian, dapat mengenkripsi dan mendekripsi website LWPDAC menggunakan algoritma *SHA-1* dapat disimpulkan bahwa Algoritma usulan dengan skema LWPDAC SHA-1 memiliki tingkat keamanan yang lebih baik daripada algoritma LWPDAC standar. Penelitian ini memperlihatkan bahwa penggabungan LWPDAC dengan SHA-1 dapat meningkatkan aspek *integrity* data melalui proses autentikasi sebelum dan sesudah enkripsi. Pengujian menggunakan metode *entropy* dan *collision resistance test* pada 60 sampel yang diambil secarak acak dan menghasilkan peningkatan nilai entropi sebesar 1,82% dan menunjukkan perbedaan signifikan dengan nilai P-Value < 0.00001.

Sebagai saran pengembangan diharapkan bahwa penelitian selanjutnya akan mengembangkan model enkripsi dan mencakup aspek keamanan informasi yang diperlukan untuk melengkapi elemen keamanan informasi, seperti aspek *non-repudiation*. Juga, melakukan perbaikan desain agar tampilan dan sistem yang dihasilkan lebih baik dan optimal dibandingkan penelitian ini.

DAFTAR PUSTAKA

[1] I. Gunawan, "Peningkatan Pengamanan Data File Menggunakan Algoritma Kriptografi AES Dari Serangan Brute Force," *TECHSI - Jurnal Teknik Informatika*, vol. 13, no. 1, p. 14, Apr. 2021, doi: 10.29103/techsi.v13i1.2395.

[2] N. D. Girsang, H. Siagian, M. H. Santoso, A. Wahyudi, and B. A. Sitorus, *Kombinasi Algoritma Kriptografi Transposisi Rail Fence Cipher dan Route Cipher*, vol. 2.

[3] "BSSN Catat Ada 207 Pencurian Data, Pemerintah Paling Banyak - Teknologi Katadata.co.id." Accessed: Oct. 20, 2024. [Online]. Available: https://katadata.co.id/digital/teknologi/6537ae3a29314/bssn-catat-ada-207-pencurian-data-pemerintah-paling-banyak#google_vignette

[4] R. Thabit, N. Izura Udzir, S. Md Yasin, A. Asmawi, and A. Gutub, "CSNTSteg: COLOR SPACING NORMALIZATION TEXT STEGANOGRAPHY MODEL to IMPROVE CAPACITY and INVISIBILITY of HIDDEN DATA", doi: 10.1109/ACCESS.Doi.

[5] A. Eka Putri, A. Kartika Dewi, and L. A. Abdul Rosyid, "Implementasi_Kriptografi_dengan_Algoritm," *Applied Information Systems and Management (AISM)*, vol. 03, pp. 1–9, 2020.

[6] R. Pamungkas, F. Wahyu, and Z. Zaney, "Penerapan Hashing SHA1 dan Algoritma Asimetris RSA untuk Keamanan Data pada Sistem Informasi berbasis Web Implementation Of Hashing SHA1 And RSA Asymetric Algorithm For Data Security In Web-Based Information Systems," *Research : Journal of Computer*, vol. 4, no. 1, pp. 84–89, 2021.

[7] S. Kataria, K. Singh, T. Kumar, and M. S. Nehra,*IEEE Second International Conference on Image Information Processing : 9-11 December, Jaypee University of Information Technology, Waknaghat, Shimla, H.P., India*. IEEE.

[8] M. Gaur and M. Sharma, "International Journal on Recent and Innovation Trends in Computing and Communication A New PDAC (Parallel Encryption with Digit Arithmetic of Cover Text) Based Text Steganography Approach for Cloud Data Security," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 3, pp. 1–9, [Online]. Available: http://www.ijritcc.org

[9] S. Panwar, M. Kumar, and S. Sharma, *4th International Conference on Internet of Things and Connected Technologies (ICIoTCT)*, vol. 1122. in Advances in Intelligent Systems and Computing, vol. 1122. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-39875-0.

[10] E. Ardhianto, Y. Heryadi, L. A. Wülandhari, and W. Budiharto, "Covertext generation using fuzzy logic approach in parallel encryption with digit arithmetic of covertext to improve information confidentiality," *International Journal of Innovative Computing, Information and Control*, vol. 19, no. 4, pp. 1311–1321, Aug. 2023, doi: 10.24507/ijicic.19.04.1311.

[11] W. Tri Handoko, E. Ardhianto, H. Murti, and R. S. Redjeki, "A LIGHT WEIGHT OF PARALLEL ENCRYPTION WITH DIGIT ARITHMETIC OF COVERTEXT ENCRYPTION MODEL," *J Theor Appl Inf Technol*, vol. 31, p. 24, 2023, [Online]. Available: www.jatit.org

[12] "Sliced images and encryption techniques in steganography using multi threading for fast retrieval." Accessed: Feb. 21, 2025. [Online]. Available: https://www.researchgate.net/publication/319096757_Sliced_images_and_encryption_techniques_in_steganography_using_multi_threading_for_fast_retrieval

[13] M. Y. Elmahi, T. M. Wahby, M. H. Sayed, M. Y. Elmahi, T. M. Wahbi, and M. H. Sayed, "Text Steganography Using Compression and Random Number Generators,". [Online]. Available: www.ijcat.com

[14] M. A. Majeed, R. Sulaiman, Z. Shukur, and M. K. Hasan, "A review on text steganography techniques," *Mathematics*, vol. 9, no. 21, Nov. 2021, doi: 10.3390/MATH9212829.

[15] T. Mahjabin, A. Olteanu, Y. Xiao, W. Han, T. Li, and W. Sun, "Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000. A Survey on DNA-Based Cryptography and Steganography", doi: 10.1109/ACCESS.DOI.

[16] Y. Wu and X. Dai, "Encryption of accounting data using DES algorithm in computing environment," *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 4, pp. 5085–5095, 2020, doi: 10.3233/JIFS-179994.

[17] R. M. Nasution, "Implementasi Metode Secure Hash Algorithm (SHA-1) Untuk Mendeteksi Orisinalitas File Audio," *Bulletin of Computer Science Research*, vol. 2, no. 3, pp. 73–84, Aug. 2022, doi: 10.47065/bulletincsr.v2i3.140.

[18] L. N. Degambur, S. Armoogum, and S. Pudaruth, "A Study of Security Impacts and Cryptographic Techniques in Cloud-based e-Learning Technologies," *International Journal of Advanced Computer Science and Applications*,

vol. 13, no. 1, pp. 58–66, 2022, doi: 10.14569/IJACSA.2022.0130108.

[19] P. Parmar and D. Sanghani, "Enhancing the Security of Confidential Data Using Video Steganography," pp. 203–210, 2021, doi: 10.1007/978-981-33-6691-6_23.

[20] "(PDF) A Novel approach towards Implicit Authentication System by using Multi- share visual key Cryptography Mechanism." Accessed: Feb. 21, 2025. [Online]. Available: https://www.researchgate.net/publication/353466822_A_Novel_approach_towards_Implicit_Authentication_System_by_using_Multi-_share_visual_key_Cryptography_Mechanism

[21] D. Shehzad and T. Dag, "LSB image steganography based on blocks matrix determinant method," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 7, pp. 3778–3793, Jul. doi: 10.3837/TIIS.2019.07.024.

[22] H. Delfs and H. Knebl, "Probabilistic Algorithms," *Information Security and Cryptography*, vol. 2, pp. 135–145, doi: 10.1007/3-540-49244-5_5.

[23] A. Vassilev and R. Staples, "Entropy as a Service: Unlocking Cryptography's Full Potential," *Computer (Long Beach Calif)*, vol. 49, no. 9, pp. 98–102, Sep. doi: 10.1109/MC.2016.275.

[24] A. Shukla and S. Kumar, "Analysis of secure watermarking based on DWT-SVD technique for piracy," *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016*, pp. 1110–1115, Jan. doi: 10.1109/CCAA.2016.7813882.

[25] A. Gutub and B. O. Al-Roithy, "Varying PRNG to improve image cryptography implementation," *Journal of Engineering Research (Kuwait)*, vol. 9, no. 3, pp. 153–183, Sep. 2021, doi: 10.36909/JER.V9I3A.10111.

[26] S. Tariq, M. Khan, A. Alghafis, and M. Amin, "A novel hybrid encryption scheme based on chaotic Lorenz system and logarithmic key generation," *Multimed Tools Appl*, vol. 79, no. 31–32, pp. 23507–23529, Aug. 2020, doi: 10.1007/S11042-020-09134-8.

[27] S. Kataria, B. Singh, T. Kumar, and H. S. Shekhawat, "PDAC (Parallel Encryption with Digit Arithmetic of Cover Text) Based Text Steganography.

[28] S. Panwar, M. Kumar, and S. Sharma, "Text Steganography Based on Parallel Encryption Using Cover Text (PECT)," in *Springer Link*, 2020, pp. 303–313. doi: 10.1007/978-3-030-39875-0_32.