3rd Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)
30 Agustus 2023 – Jakarta, Indonesia
Volume 2, Nomor 2, September 2023 - ISSN 2962-8628 (*online*)

PENERAPAN ALGORITME RSA DAN HUFFMAN ENCODE UNTUK PENGAMANAN FILE PADA SMP NEGERI 16 JAKARTA

Achmad Sultan Wijaya[1*], Painem[2*]

[1,2] Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia

Email: [1*]1911502274@student.budiluhur.ac.id, [2]painem@budiluhur.ac.id
(* : corresponding author)

Abstrak- Keamanan informasi merupakan hal yang sangat penting, terutama dengan perkembangan teknologi *software* dan penggunaan *internet* yang semakin luas. Keamanan informasi berkaitan erat dengan upaya mencegah pencurian data atau informasi oleh pihak yang tidak bertanggung jawab, yang dapat mengakses atau mengubah informasi tersebut. Tujuan dari pengamanan informasi adalah melindungi informasi agar tidak dapat diakses oleh orang yang tidak berwenang. Dalam konteks Sekolah SMP 16 Negeri Jakarta, pernah terjadi kebocoran data pada *file* soal ulangan akhir sekolah. Kebocoran ini terjadi karena data masih disimpan secara lokal pada komputer dan flashdisk. Oleh karena itu, dalam penelitian ini, dilakukan pengimplementasian kriptografi menggunakan metode *Rivest Shamir Adleman* (RSA) dan *Huffman Encode* untuk meningkatkan keamanan data dalam file dokumen. Metode *Rivest Shamir Adleman* adalah algoritme kriptografi asimetris yang menggunakan sepasang kunci publik dan kunci pribadi. Metode ini memiliki keamanan yang tinggi karena kesulitan dalam memfaktorkan bilangan prima. Selain itu, algoritme *Huffman* digunakan sebagai algoritme pengkodean yang memampukan kompresi data dengan mengkodekan karakter yang sering muncul menggunakan rangkaian bit yang lebih pendek. Dengan mengimplementasikan aplikasi kriptografi menggunakan metode *Rivest Shamir Adleman*, data-data penting dapat diamankan secara maksimal dengan teknik enkripsi pada *file* dokumen. Penelitian ini membuktikan bahwa metode *Rivest Shamir Adleman* dan *Huffman Encode* efektif dalam menjaga keamanan data dalam *file* dokumen. Hasil penelitian menunjukkan bahwa dengan menerapkan metode *Rivest Shamir Adleman* (RSA) dan *Huffman Encode*, keamanan *file* di SMP 16 Negeri Jakarta berhasil ditingkatkan dan proses enkripsi memiliki efektivitas yang baik. Waktu rata-rata proses enkripsi adalah 7413,1 milidetik, sedangkan waktu rata-rata proses dekripsi adalah 7162,7 milidetik. Ukuran *file* rata-rata setelah proses enkripsi adalah 10072400 byte, sedangkan ukuran rata-rata setelah proses dekripsi adalah 551100 byte. Selain itu, metode *Huffman Encode* menghasilkan waktu rata-rata proses enkripsi sebesar 79 milidetik dan waktu rata-rata proses dekripsi sebesar 310,8 milidetik. Ukuran *file* rata-rata setelah proses enkripsi adalah 3357700 byte, sedangkan ukuran rata-rata setelah proses dekripsi adalah 551100 byte.

Kata Kunci: Keamanan data, Kriptografi, *Rivest Shamir Adleman*(RSA), *Huffman Encode.*

IMPLEMENTATION OF RSA AND HUFFMAN ENCODE ALGORITHMS FOR FILE SECURITY AT SMP NEGERI 16 JAKARTA

Abstract- *Information security is a crucial aspect, especially given the widespread development of software technology and internet usage. Information security closely relates to efforts aimed at preventing data or information theft by unauthorized parties with the ability to access or alter such information. The objective of information security is to safeguard information from unauthorized access. In the context of SMP 16 Negeri Jakarta School, a data leak occurred in 2021 regarding the final exam question files. This leak transpired because the data was still stored locally on computers and flash drives. Therefore, this research implements cryptography using the Rivest-Shamir-Adleman (RSA) method and Huffman Encoding to enhance data security in document files. The Rivest-Shamir-Adleman method is an asymmetric cryptographic algorithm that uses a pair of public and private keys. This method offers high security due to the difficulty in factoring prime numbers. Additionally, the Huffman algorithm is employed as a coding algorithm that enables data compression by encoding frequently occurring characters with shorter bit sequences. By implementing a cryptography application using the Rivest-Shamir-Adleman method, critical data can be maximally secured through encryption techniques in document files. This research demonstrates the effectiveness of the Rivest-Shamir-Adleman and Huffman Encoding methods in preserving data security within document files. The results show that by applying the Rivest-Shamir-Adleman (RSA) and Huffman Encoding methods, file security at SMP 16 Negeri Jakarta has been successfully improved, and the encryption process is highly efficient. The average encryption process time is 7,413.1 milliseconds, while the average decryption process time is 7,162.7 milliseconds. The average file size after the encryption process is 10,072,400 bytes, while the average size after the decryption process is 551,100 bytes. Additionally, the Huffman Encoding method yields an average encryption process time of 79 milliseconds and an average decryption process time of 310.8 milliseconds. The average file size after the encryption process is 3,357,700 bytes, while the average size after the decryption process is 551,100 bytes.*

Keywords*: Data security,Cryptography,Rivest Shamir Adleman(RSA),Huffman Encode.*

1. PENDAHULUAN

Keamanan informasi menjadi salah satu hal yang sangat penting, terutama dengan pesatnya perkembangan teknologi *software* dan semakin meluasnya penggunaan internet. Upaya untuk mencegah pencurian data atau informasi oleh pihak yang tidak bertanggung jawab serta menghindari akses atau perubahan informasi tersebut. Tujuan utama dari pengamanan informasi adalah melindungi informasi agar tidak dapat diakses oleh orang yang tidak berwenang. Dalam mengamankan komputer, terdapat banyak cara yang dapat digunakan, salah satunya adalah dengan memanfaatkan teknik kriptografi[1].

Masalah keamanan data seringkali diabaikan atau kurang mendapat perhatian yang cukup dari perusahaan dan instansi. Bahkan, keamanan data seringkali menjadi prioritas terakhir dalam menyimpan dan menukarkan informasi. Di SMP 16 Negeri Jakarta, keamanan data pada *file* dokumen belum ada implementasi yang memadai. Data masih disimpan di local komputer dan flashdisk, sehingga menghadirkan potensi besar untuk pencurian data dan kehilangan data. Kasus pencurian data berupa *file* ujian sekolah akhir sebelumnya telah terjadi di sekolah tersebut.

Oleh karena itu, langkah yang perlu diambil adalah dengan mengenkripsi *file* dokumen. Penerapan kriptografi dengan metode *Rivest Shamir Adlman* (RSA) dan *Huffman Encode* menjadi solusi yang dipertimbangkan dalam penelitian ini. Dengan demikian, diharapkan data dapat lebih aman dan terlindungi dari ancaman keamanan, serta memberikan perlindungan yang lebih baik terhadap informasi yang dimiliki sekolah.

Kriptografi, yang berasal dari bahasa Yunani "*crypto*" yang berarti rahasia dan "*graphy*" yang berarti tulisan, secara umum dapat diartikan sebagai seni menyimpan pesan dengan cara yang rahasia. Pada dasarnya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Pada awal sejarahnya, setiap orang memiliki cara unik untuk menjaga kerahasiaan pesan, sehingga kriptografi menjadi suatu seni[2].

Pada tahun 1977, algoritme RSA ditemukan oleh Ron Rivest, Adi Shamir, dan Len Adleman dari MIT (*Massachusetts Institute of Technology*). Algoritme ini telah menjadi penemuan revolusioner dalam bidang kriptografi kunci publik dan hingga saat ini tetap digunakan karena kunci yang digunakan memiliki panjang yang besar serta penerapannya terus disempurnakan [3].

Sementara itu, algoritme *Huffman encode* merupakan salah satu algoritme kompresi *lossless* tertua yang ditemukan oleh David Huffman pada tahun 1952. Konsep dasar dari algoritme Huffman adalah menciptakan kode biner yang lebih singkat untuk setiap karakter dengan kemunculan yang lebih tinggi, sementara karakter dengan frekuensi kemunculan yang lebih rendah diberikan kode biner yang lebih panjang [4].

Dalam penelitian terdahulu, banyak penelitian telah membahas tentang kriptografi rsa dalam konteks keamanan data. Penelitian pertama adalah "Implementasi Algoritma Kriptografi Rivest Shamir Adlemen untuk Mengamankan Data Ijazah pada SMK Swasta Prama Artha Kab. Simalungun"[5]. Selanjutnya, penelitian kedua adalah "Implementasi Algoritma Kriptografi RSA (*Rivest Shamir Adleman*) untuk Keamanan Data Rekam Medis Pasien"[1]. Selanjutnya, Penelitian ketiga adalah "Implementasi Kriptografi Pada Aplikasi Memo Berbasis Android Menggunakan Algoritma RSA"[6]. Selanjutnya, Penelitian Keempat adalah "Implementasi Keamanan Pesan Teks Menggunakan Kriptografi Algoritma Rsa Dengan Metode Waterfall Berbasis Java"[7], [8]. Terakhir, Penelitian Kelima adalah "Implementasi Kriptografi Rsa Untuk Peningkatan Keamanan Database *E-Commerce*"[8]. Pada lima penelitian sebelumnya, digunakan satu algoritma yaitu algoritma RSA, tetapi dengan tujuan yang berbeda untuk mengamankan data. Perbedaan pada penelitian terbaru adalah penggunaan dua algoritma, yakni algoritma RSA dan *Huffman Encode*. Dengan menggunakan dua algoritma ini mempunyai keunggulan pada efisiensi kompresi *file* dengan *Huffman encode* yang dapat mengurangi ukuran *file* dan waktu proses sebelum di enkripsi menggunakan RSA.

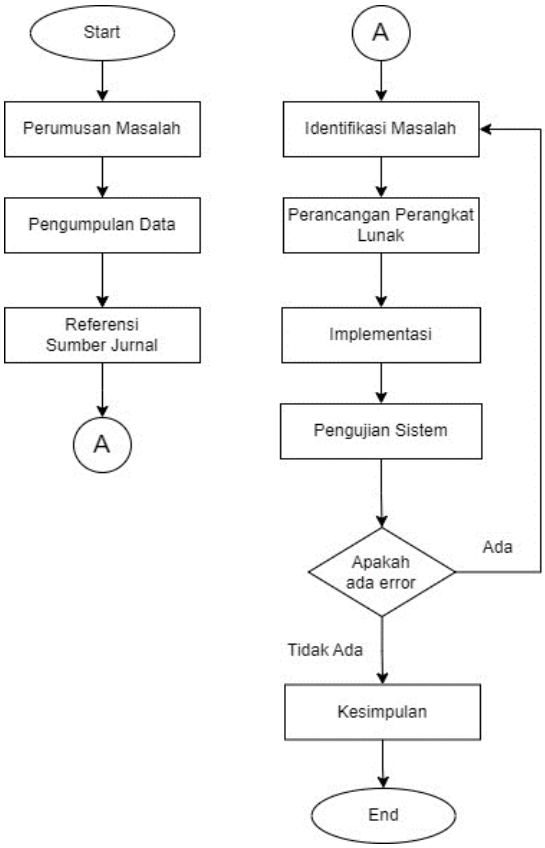
2. METODE PENELITIAN

2.1 Data Penelitian

Data penelitian berupa laporan keuangan SMP Negeri 16 dikumpulkan secara langsung. Data penting ini memerlukan keamanan untuk mencegah kebocoran dan peretasan, yang saat ini belum optimal. Faktor penyimpanan data di *folder* komputer atau *flash drive* serta faktor manusia berpengaruh pada kerahasiaan data. Oleh karena itu, SMP Negeri 16 membuat program untuk meningkatkan keamanan data melalui proses enkripsi dan dekripsi.

2.2 Tahap Penelitian

Pada gambar 1 menggambarkan langkah awal dan akhir yang akan diambil dalam penerapan metode penelitian ini.



Gambar 1. Tahapan Metode Penelitian

Berikut akan menjelaskan secara detil tentang tahapan dari metode penelitian yang terdapat pada gambar 1 di atas :

a. Perumusan Masalah
Pada tahapan perumusan masalah, dilakukan penentuan masalah yang akan diselesaikan dalam penelitian ini, yaitu membangun sistem untuk pengamanan data sekolah, khususnya laporan anggaran keuangan BOP dan BOS di SMP Negeri 16 Jakarta. Sistem ini akan menggunakan metode kriptografi *Rivest Shamir Adleman* (RSA) dan algoritme *Huffman encode*.

b. Pengumpulan Data
Tahap pengumpulan data dilakukan melalui dua proses, yaitu wawancara dan observasi.
1. Wawancara dilakukan dengan bertanya langsung kepada pihak terkait untuk mendapatkan informasi tentang aplikasi dan keamanan yang ada.
2. Observasi dilakukan di SMP Negeri 16 Jakarta untuk memperoleh informasi yang lebih faktual mengenai data dan informasi yang dibutuhkan dalam penelitian.

c. Referensi Sumber Jurnal
Pada tahap referensi sumber jurnal digunakan untuk melakukan *review* terhadap jurnal-jurnal sebelumnya yang relevan. Hal ini bertujuan untuk mendapatkan referensi yang kuat dalam menentukan metode yang tepat untuk menyelesaikan permasalahan yang akan diteliti.

d. Indentifikasi Masalah
Setelah dilakukan pengumpulan data, tahap identifikasi masalah dilakukan dengan melakukan analisis data dan analisis penerapan algoritme.
1. Analisis Data
   a) Mengumpulkan berkas yang diperlukan guna mendapatkan informasi yang dibutuhkan dalam proses perancangan program
   b) Mengelompokkan *file* berdasarkan jenisnya.
   c) Melakukan enkripsi *file* untuk menetapkan langkah-langkah yang perlu diambil dalam membangun aplikasi yang memiliki tampilan yang baik dan mudah dipahami.
2. Analisis Penerapan Algoritme

a) Menentukan bilangan prima yang akan digunakan sebagai pasangan kunci publik dan kunci privat RSA untuk proses enkripsi dan dekripsi *file*.
b) Proses kompresi file dilakukan dengan menggunakan algoritme *Huffman Encode*. Proses ini bertujuan untuk mengurangi ukuran file dan meningkatkan efisiensi penyimpanan.
c) Proses enkripsi *file* dilakukan menggunakan kunci enkripsi publik RSA. *File* yang telah dikompresi akan mengalami transformasi menjadi *ciphertext*, yang merupakan representasi terenkripsi dari file asli.
d) Proses yang berikutnya adalah dekripsi *ciphertext*. Proses ini menggunakan kunci dekripsi privat RSA untuk mengembalikan *ciphertext* ke bentuk plainteks sebelum dikompresi dengan algoritme *Huffman Encode*.
e) Setelah proses dekripsi selesai, langkah selanjutnya adalah melakukan dekompresi pada *plainteks* yang dihasilkan. Menggunakan algoritme *Huffman Decode*, plainteks dikembalikan ke bentuk semula sebelum dikompresi.
f) Akhirnya, diperoleh hasil dekripsi berupa file yang merupakan representasi yang dapat dibaca dan dipahami dari data yang telah dilindungi sebelumnya.

e. Perancangan Perangkat Lunak
Tahap perancangan perangkat lunak dilakukan untuk merancang sistem sesuai dengan hasil analisis, khususnya dalam perancangan enkripsi dan dekripsi. Integrasi dengan komponen lain juga dilakukan dalam tahapan ini.

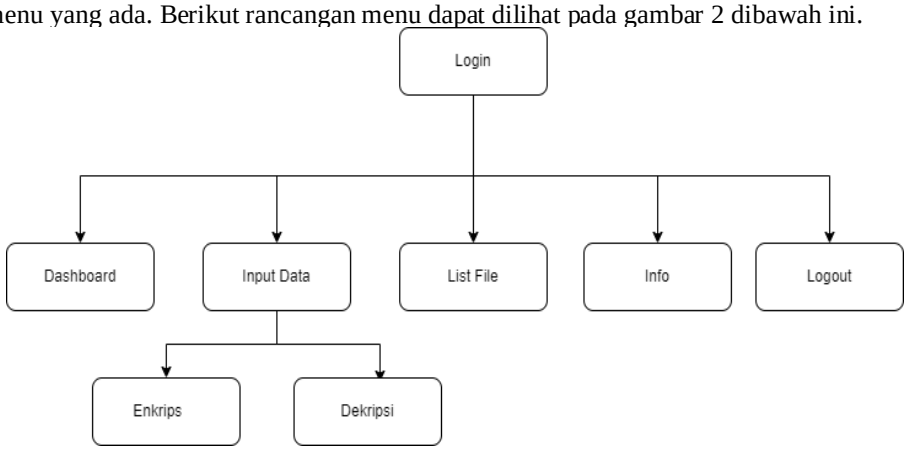
f. Implementasi
Proses implementasi dilakukan dengan mengimplementasikan modul-modul yang telah dirancang dalam bahasa pemrograman PHP dan DBMS PHPMyAdmin.

g. Pengujian Sistem
Tahap pengujian sistem dilakukan dengan menggunakan metode *blackbox* untuk memastikan kesesuaian sistem dengan perancangan yang telah dibuat sebelumnya.

h. Kesimpulan
Pada tahap akhir, dapat disimpulkan bahwa penerapan metode kriptografi RSA dan *Huffman Encode* dalam mengamankan *file* pada data penting di SMP Negeri 16 Jakarta berfungsi dengan baik. Metode ini melindungi kerahasiaan dan integritas data dengan menggunakan kunci publik dan kunci pribadi RSA, serta mengompresi *file* dengan algoritme *Huffman*.

2.3 Rancangan Menu

Rancangan menu merupakan tahapan yang menunjukkan bagaimana antarmuka pengguna aplikasi akan disusun. Ini mencakup serangkaian langkah atau tahapan yang akan digunakan oleh pengguna saat menggunakan aplikasi tersebut. Rancangan menu mencakup fitur-fitur menu yang tersedia, mulai dari menu *login* hingga menu utama dan sub menu yang ada. Berikut rancangan menu dapat dilihat pada gambar 2 dibawah ini.

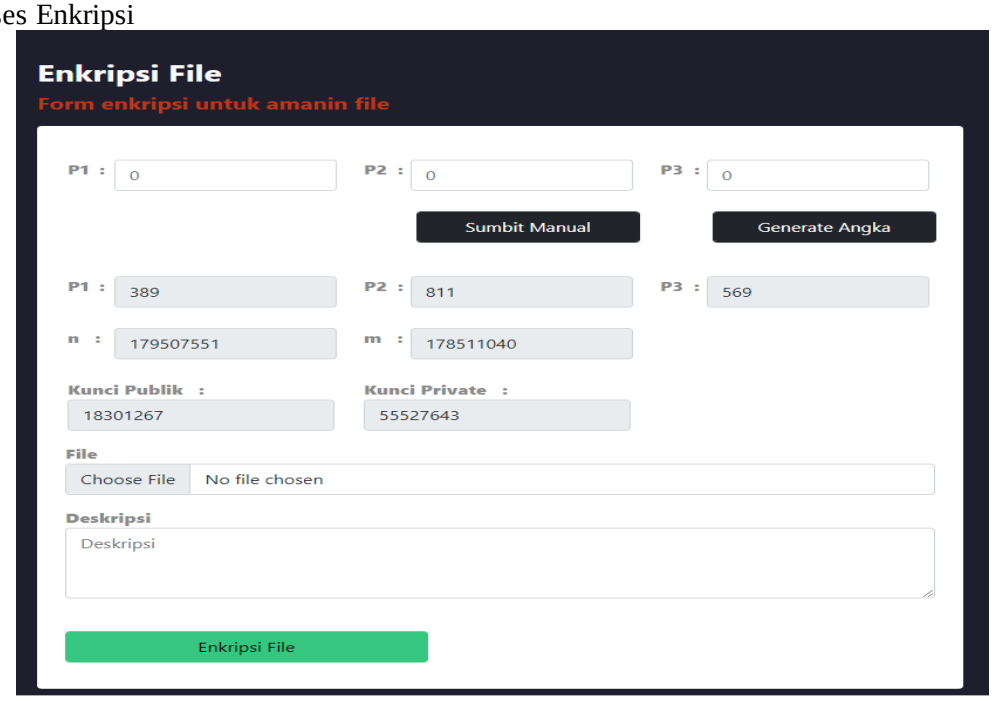


Gambar 2. Rancangan Menu

3. HASIL DAN PEMBAHASAN

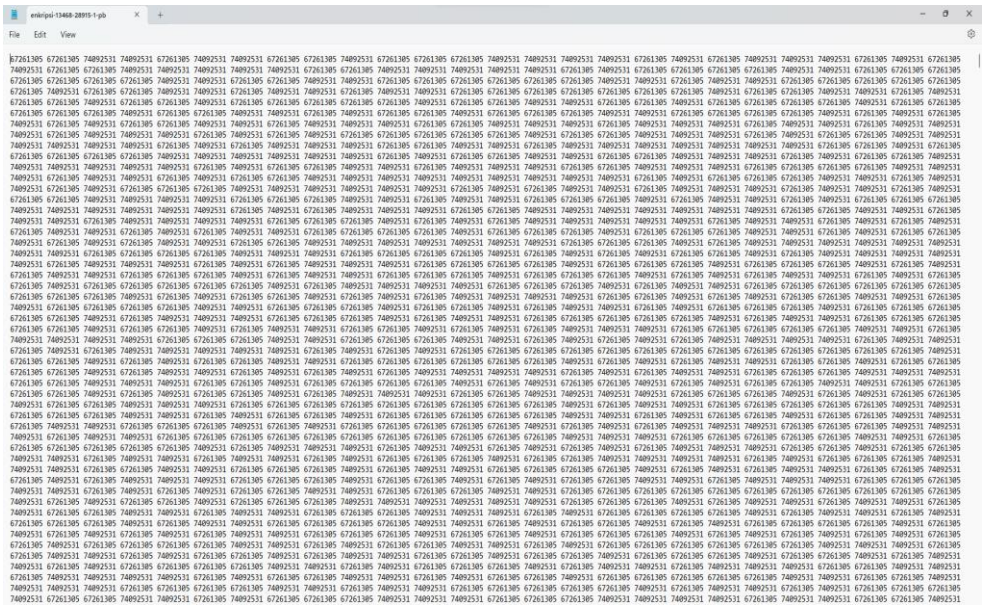
3.1 Implementasi Metode

a. Proses Enkripsi



Gambar 3. Proses Ekripsi Untuk Pembangkitan Kunci

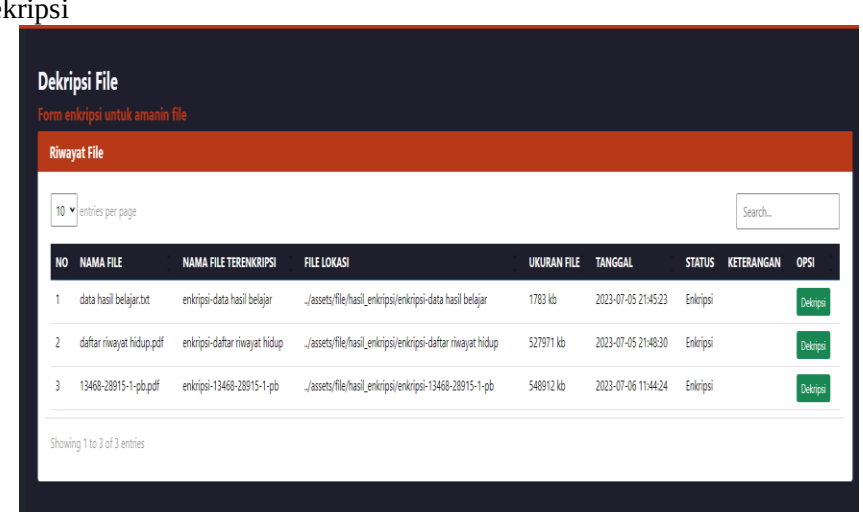
Pada gambar 3 ini proses enkipsi dengan metode algoritme *Rivest Shamir Adleman*(RSA) untuk pembangkitan kunci agar mendapatkan publik dan kunci pribadi. Pada gambar diatas mendapatkan nilai n yaitu 179507551, nilai m 178511040, nilai kunci publik 18301267 dan kunci *private* 55527643.



Gambar 4. Hasil File Yang Dienkripsi

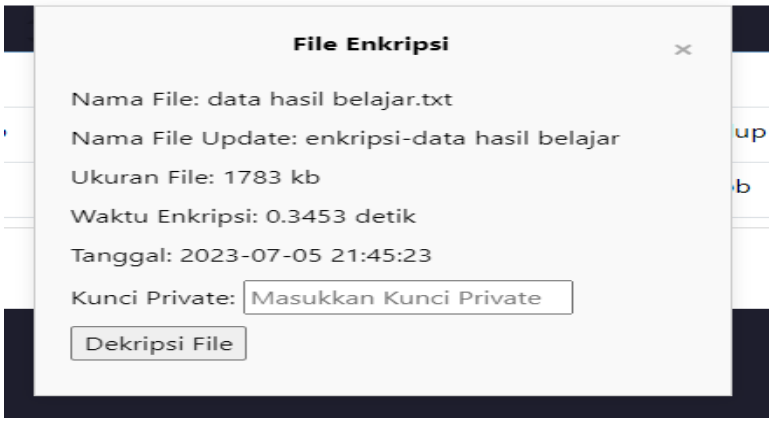
Pada gambar 4 ini adalah tampilan *file* yang telah terenkripsi, sebelum *output* tampilan *file* seperti ini yaitu, dengan proses metode RSA dan *Huffman encode*. Pada *file* yang sudah terenkripsi menampilkan angka – angka acak yang sulit dipahamin.

b. Proses Dekripsi



Gambar 5. Proses Pemilihan Yang Ingin Didekripsi

Pada Gambar 5 ini adalah proses pemilihan file yang ingin dekripsi. Tiap file dekripsi yang dipilih memiliki kunci dekripsi masing – masing.



Gambar 6. Proses Input Kunci Private

Pada gambar 6 ini adalah proses dekripsi untuk memasukan kunci *private*. Apabila memasukan kunci *private* salah maka proses dekripsi akan gagal.

Gambar 7. Hasil File Yang Sudah Didekripsi

Pada gambar 7 ini adalah tampilan hasil *file* yang sudah terdekripsi. sebelum *output* tampilan *file* seperti ini yaitu, dengan proses metode *Huffman encode* dan RSA. Apabila memasukan kunci *private* tidak benar maka hasil file seperti ini tidak akan diproses atau ditampilkan.

3.2 Algoritme Enkripsi RSA dan Huffman Encode

a. Algoritma Enkripsi RSA

Algoritme pada gambar 8 dibawah ini, menjelaskan mengenai proses metode pada enkripsi RSA. Pada enkripsi algoritma RSA untuk mengekripsi pada *file* asli untuk mengamankan data.

```
Start
Input "bilangan prima & generate key" then
generateRandomNumber(100,1000)
   Isprime
   calculateModulusAndTotient
   euclideanGCD
   Euclid
   calculatePublicKey
   calculatePrivateKey
 if input Pilih then
 rsaEncrypt
 Return
```

Gambar 8. Algoritme Enkripsi RSA

b. Algoritma Enkripsi *Huffman Encode*

Algoritme pada gambar 9 dibawah ini, menjelaskan mengenai proses metode pada enkripsi *Huffman Encode*. Pada enkripsi algoritma *Huffman Encode* untuk kompresikan pada *file* setelah dienkripsi RSA.

```
Start
if input Pilih then
  hitungFrekuensi
  bangunPohonHuffman
  generateHuffmanCodes
  huffmanEncode
  generateHuffmanCodes
Output code huffman
Return
```

Gambar 9. Algoritme Enkripsi Huffman Encode

c. Algoritma Dekripsi RSA

Algoritme pada gambar 10 dibawah ini, menjelaskan mengenai proses metode pada Dekripsi RSA. Pada dekripsi algoritma RSA untuk mendekripsikan pada file enkripsi.

```
Start
Input "Key private" then
Input "ChipperFile" then
  rsaDecrypt
Return
```

Gambar 10. Algoritme Dekripsi RSA

d. Algoritma Dekripsi *Huffman Encode*

Algoritme pada gambar 10 dibawah ini, menjelaskan mengenai proses metode pada Dekripsi *Huffman Encode*. Pada dekripsi algoritma *Huffman Encode* untuk mendekripsikan pada file enkripsi.

```
Start
Input "Key private" then
Input "ChipperFile" then
  hitungFrekuensi
  bangunPohonHuffman
  huffmanDecode
Return
```

Gambar 11 Algoritme Dekripsi Huffman Encode

3.3 Rancangan Aplikasi

Pada table pengujian adalah hasil dari proses enkripsi file yang dikerjakan oleh sistem dengan menggunakan aplikasi. Jumlah pengujian file memiliki berjumlah 10 file dan file tersebut dari 2018 - 2022. Untuk pengujiannya dengan input tiga bilangan prima yang sama dan kunci publik atau pribadi yang berbeda.

3.4 Hasil Pengujian

Pada table pengujian adalah hasil dari proses enkripsi file yang dikerjakan oleh sistem dengan menggunakan aplikasi. Jumlah pengujian file memiliki berjumlah 10 file dan file tersebut dari 2018 - 2022. Untuk pengujiannya dengan input tiga bilangan prima yang sama dan kunci publik atau pribadi yang berbeda.

a. Tabel Hasil Enkripsi Pada Algoritme Rivest Shamir Adleman dan Huffman Encode

Dibawah ini adalah sebuah tabel hasil *file* Dekripsi menggunakan algoritme *Rivest Shamir Adleman* dan *Huffman Encode*. Pada hasil pengujian enkripsi file nomor satu, telah berhasil melakukan pengujian. Pengujian ini menghasilkan ukuran file enkripsi yaitu 6952000 *byte* dari *file* asli yaitu 551000 *byte*, untuk durasi waktu enkripsi yaitu 5947 milidetik dan perubahaan nama *file* asli yaitu enkripsi-e-rkas – cetak rkas 2020-1.rda dari nama file *asli* yaitu e-RKAS - Cetak RKAS 2020-1.docx. Maka kesepuluh percobaan ini, berhasil dilakukan enkripsi terhadap *file* asli, menghasilkan ukuran file dan durasi enkripsi pada semua file.

Tabel 1. Hasil Pengujian Enkripsi File

NO	Nama File Awal	Ukuran File	Nama File hasil Enkripsi	Ukuran File Setelah Enkripsi	Durasi Enkripsi	Keterangan
1	e-RKAS - Cetak RKAS 2020-1.docx	551000 byte	enkripsi-e-rkas - cetak rkas 2020-1.rda	6952000 byte	5947 milidetik	Berhasil
2	e-RKAS - Cetak RKAS 2020-2.xlsx	37000 byte	enkripsi-e-rkas - cetak rkas 2020-2.rda	870000 byte	537 milidetik	Berhasil
3	e-RKAS - Cetak RKAS 2020-3.pdf	1325000 byte	enkripsi-e-rkas - cetak rkas 2020-3.rda	21996000 byte	19509 milidetik	Berhasil
4	e-RKAS - Cetak RKAS 2021-4.docx	99000 byte	enkripsi-e-rkas - cetak rkas 2021-4.rda	2355000 byte	2029 milidetik	Berhasil
5	e-RKAS - Cetak RKAS 2021-5.xlsx	40000 byte	enkripsi-e-rkas - cetak rkas 2021-5.rda	938000 byte	717 milidetik	Berhasil
6	e-RKAS - Cetak RKAS 2021-6.pdf	1442000 byte	enkripsi-e-rkas - cetak rkas 2021-6.rda	24083000 byte	14725 milidetik	Berhasil
7	e-RKAS - Cetak RKAS 2022-7.pdf	671000 byte	enkripsi-e-rkas - cetak rkas 2022-7.rda	11384000 byte	7090 milidetik	Berhasil
8	e-RKAS - Cetak RKAS 2022-8.excel	24000 byte	enkripsi-e-rkas - cetak rkas 2022-8.rda	543000 byte	336 milidetik	Berhasil
9	e-RKAS - Cetak RKAS 2019-9.pdf	716000 byte	enkripsi-e-rkas - cetak rkas 2019-9.rda	17111000 byte	14254 milidetik	Berhasil
10	e-RKAS -	606000	enkripsi-e-rkas -	14492000	8987	Berhasil

	Cetak RKAS 2018-10.pdf	byte	cetak rkas 2018-10.rda	byte	milidetik
Rata – rata		551100 byte		10072400 byte	7413,1 milidetik

b. Tabel Hasil Dekripsi Pada Algoritme Rivest Shamir Adleman dan Huffman Encode

Dibawah ini adalah sebuah table hasil file Dekripsi menggunakan algoritme *Rivest Shamir Adleman* dan *Huffman Encode*. Pada hasil pengujian Dekripsi file nomor satu, telah berhasil melakukan pengujian. Pengujian ini menghasilkan ukuran *file* dekripsi yaitu 551000 *byte* dari *file* enkripsi yaitu 6952000 *byte*, untuk durasi waktu dekripsi yaitu 5829 milidetik dan perubahaan nama *file* asli yaitu dekripsi-e-rkas - cetak rkas 2020-1. docx dari nama *file* enkripsi yaitu enkripsi-e-rkas - cetak rkas 2020-1.rda Maka kesepuluh percobaan ini, berhasil dilakukan enkripsi terhadap *file* asli, menghasilkan ukuran file dan durasi enkripsi pada semua file.

Tabel 2. Hasil Pengujian Dekripsi File

NO	Nama File Awal	Ukuran File	Nama File hasil Dekripsi	Ukuran File Setalah Dekripsi	Durasi Dekripsi	Keterangan
1	enkripsi-e-rkas - cetak rkas 2020-1.rda	6952000 byte	dekripsi-e-rkas - cetak rkas 2020-1. docx	551000 byte	5829 milidetik	Berhasil
2	enkripsi-e-rkas - cetak rkas 2020-2.rda	870000 byte	dekripsi-e-rkas - cetak rkas 2020-2.xlsx	37000 byte	517 milidetik	Berhasil
3	enkripsi-e-rkas - cetak rkas 2020-3.rda	21996000 byte	dekripsi-e-rkas - cetak rkas 2020-3.pdf	1325000 byte	18278 milidetik	Berhasil
4	enkripsi-e-rkas - cetak rkas 2021-4.rda	2355000 byte	dekripsi-e-rkas - cetak rkas 2021-4.docx	99000 byte	1947 milidetik	Berhasil
5	enkripsi-e-rkas - cetak rkas 2021-5.rda	938000 byte	dekripsi-e-rkas - cetak rkas 2021-5.xlsx	40000 byte	696 milidetik	Berhasil
6	enkripsi-e-rkas - cetak rkas 2021-6.rda	24083000 byte	dekripsi-e-rkas - cetak rkas 2021-6.pdf	1442000 byte	14390 milidetik	Berhasil
7	enkripsi-e-rkas - cetak rkas 2022-7.rda	11384000 byte	dekripsi-e-rkas - cetak rkas 2022-7.pdf	671000 byte	6802 milidetik	Berhasil
8	enkripsi-e-rkas - cetak rkas 2022-8.rda	543000 byte	dekripsi-e-rkas - cetak rkas 2022-8.pdf	24000 byte	325 milidetik	Berhasil
9	enkripsi-e-rkas - cetak rkas 2019-9.rda	17111000 byte	dekripsi-e-rkas - cetak rkas 2019-9.pdf	716000 byte	14254 milidetik	Berhasil
10	enkripsi-e-rkas - cetak rkas 2018-10.rda	14492000 byte	dekripsi-e-rkas - cetak rkas 2018-10.pdf	606000 byte	8589 milidetik	Berhasil
	Rata – rata	10072400 byte		551100 byte	7162,7 milidetik	

4. KESIMPULAN

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa di SMP Negeri 16 Jakarta dapat diimplementasikan aplikasi berbasis web untuk kriptografi pengamanan file dengan format .doc, .xlsx, dan .pdf menggunakan algoritma *Rivest Shamir Adleman*(RSA). Aplikasi ini akan membantu sekolah dalam mengamankan file-file berisi data keuangan. Proses enkripsi dan dekripsi berjalan tanpa mengubah ukuran file asli. Selain itu, sistem kriptografi berbasis website ini diharapkan dapat melakukan pengamanan pada berbagai format file sehingga data terhindar dari penyalahgunaan. Dan hasil pengujian dari 10 file pada proses enkripsi yaitu rata – rata ukuran file sebesar 10072400 byte dari file asli yaitu 551100 byte dan rata – rata waktu enkripsi yaitu 7413,1 milidetik, untuk proses dekripsi yaitu rata – rata ukuran file sebesar 551100 byte dari file enkripsi yaitu 10072400 byte dan rata – rata waktu dekripsi yaitu 7162,7 milidetik.

DAFTAR PUSTAKA

[1] Sutejo, "IMPLEMENTASI ALGORITMA KRIPTOGRAFI RSA (RIVEST SHAMIR ADLEMAN) UNTUK KEAMANAN DATA REKAM MEDIS PASIEN IMPLEMENTATION OF RSA CRYPTOGRAPHY ALGORITHM (RIVEST SHAMIR ADLEMAN) FOR PATIENT MEDICAL RECORD DATA SECURITY," *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 4, no. 1, 2021.

[2] R. Lestari, R. Buaton, and I. Gultom, "Penerapan Algoritma OTP dan Algoritma RSA CRT dalam Pengamanan Citra," *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, vol. 4, no. 2, pp. 180–190, 2021, [Online]. Available: https://ojs.trigunadharma.ac.id/index.php/jsk/index

[3] S. Suhandinata, R. A. Rizal, D. O. Wijaya, P. Warren, and S. Srinjiwi, "ANALISIS PERFORMA KRIPTOGRAFI HYBRID ALGORITMA BLOWFISH DAN ALGORITMA RSA," *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, vol. 6, no. 1, pp. 1–10, Dec. 2019, doi: 10.33330/jurteksi.v6i1.395.

[4] Rosanny N Sihombing, "Implementasi Algoritma Multi Group Huffman Dalam Kompresi File Teks Dokumen," *Buletin Ilmiah Informatika Teknologi*, vol. 1, no. 3, pp. 76–82, 2023, [Online]. Available: https://ejurnal.amikstiekomsu.ac.id/index.php/BIIT

[5] W. Wahyudi, D. Hartama, I. O. Kirana, S. Sumarno, and I. Gunawan, "Implementasi Algoritma Kriptografi Rivest Shamir Adlemen untuk Mengamankan Data Ijazah pada SMK Swasta Prama Artha Kab. Simalungun," *Jurnal Ilmu Komputer dan Informatika*, vol. 2, no. 1, pp. 57–66, Jan. 2022, doi: 10.54082/jiki.19.

[6] Giri Adi Nuryanto and Hari Murt, *IMPLEMENTASI KRIPTOGRAFI PADA APLIKASI MEMO BERBASIS ANDROID MENGGUNAKAN ALGORITMA RSA*. 2019.

[7] R. Firmansyah and A. A. Permana, "IMPLEMENTASI KEAMANAN PESAN TEKS MENGGUNAKAN KRIPTOGRAFI ALGORITMA RSA DENGAN METODE WATERFALL BERBASIS JAVA," 2019.

[8] S. Rahmadhiyanti, "IMPLEMENTASI KRIPTOGRAFI RSA UNTUK PENINGKATAN KEAMANAN DATABASE E-COMMERCE," 2019.

[9] I. Listiani *et al.*, "PERANCANGAN KEAMANAN DATA PASIEN DI KLINIK KECANTIKAN RATU BEAUTY STUDIO MENGGUNAKAN METODE KRIPTOGRAFI RSA," 2022.

[10] Jafarudin Firdaus, Rini Marwati, and Sumanang Muhtar Gozali, "PENYANDIAN PESAN MENGGUNAKAN KOMBINASI ALGORITMA RSA YANG DITINGKATKAN DAN ALGORITMA ELGAMAL," *Departemen Pendidikan Matematika FPMIPA UPI*, vol. 6, no. 1, pp. 23–32, 2018.