Analisa Metode Kriptografi Modern Advance Encryption Standar (AES) 128 Bit dalam Mengenkripsi dan Mendekripsi File Dokumen Digital

Sekar Putri Ananda, Saepul Lukman dan Irfan

STMIK Jakarta STI&K
Jl. BRI Radio Dalam Kebayoran Baru Jakarta Selatan
E-mail: sekar.anananda@gmail.com, fulman2012@gmail.com*, irfansasa357@gmail.com

Abstrak

Pada tahun 2021 ini, pemerintah Indonesia telah menerapkan berbagai kebijakan yang mewajibkan masyarakat untuk bekerja dari rumah sebagai upaya menekan jumlah kasus COVID-19. Dengan demikian, pengiriman berbagai dokumen pun harus melalui media elektronik. Data yang diunggah ke Internet biasanya berupa data teks biasa. Hal ini sangat berbahaya jika ada pihak lain yang dapat mencegat informasi tersebut dan dengan mudah mencegatnya. Baca isi dokumen. Salah satu metode keamanan yang dapat Anda gunakan Perlindungan informasi, atau prosedur enkripsi. Mirip dengan tujuan dari penelitian ini, yaitu merancang dan membangun aplikasi menggunakan kriptografi tingkat lanjut. Standar Enkripsi 128-bit (AES-128) dengan kunci simetris untuk melakukan enkripsi Dekripsi berbagai jenis file dokumen digital.Berdasarkan hasil ujicoba dengan menggunakan metode *Blackbox*, diketahui bahwa aplikasi web yang dibuat telah berhasil dibangun dan diberi nama Documentcryption. Dokumen digital berhasil dienkripsi dan didekripsi kembali. Berdasarkan hasil uji coba juga dapat diketahui bahwa kecepatan proses enkripsi dan dekripsi dipengarui oleh besar ukuran dokumen yang diproses. Semakin kecil ukurannya, maka semakin cepat pula waktu proses yang dibutuhkan. Tentu saja aplikasi ini untuk membantu pengguna menjaga keandalan dokumen mereka apa yang disampaikan dapat dengan selamat sampai di tempat tujuan dan tidak dapat diketahui atau dimanipulasi oleh pihak yang tidak bertanggung jawab

Kata kunci : AES128, Kriptografi, Dokumen Digital, Enkripsi, Dekripsi, PHP

Pendahuluan

Pada awal tahun 2020 Virus Corona mulai menjadi pendemi global dan menjadi masalah kesehatan di beberapa negara. Salah satu cara pencegahan di Indonesia yaitu dengan Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM). PPKM mewajibkan untuk menerapkan 100 persen *Work From Home* (WFH). Dengan demikian, pengiriman berbagai dokumen pun harus melalui media elektronik, seperti *smartphone,* laptop, atau internet.

Kerahasiaan adalah aspek keamanan informasi yang mencegah informasi yang disimpan disalahgunakan oleh orang yang tidak berwenang. Perlindungan data informasional dapat dilakukan dengan mengenkripsi data menggunakan algoritma yang telah ditentukan sebelumnya. Di sini, proses enkripsi diartikan sebagai proses perubahan dari pesan asli (*plaintext*) menjadi pesan terproteksi (dalam hal ini pesan terenkripsi (*ciphertext*)), pesan menjadi pesan terenkripsi yang dilindungi. format disebut: Dekripsi. [1].

Pesatnya perkembangan teknologi informasi dan komunikasi dapat mempermudah aktivitas pertukaran informasi. Permasalahannya adalah data-data yang diunggah ke internet, pada umumnya merupakan data plaintext. Hal ini sangat beresiko saat ada pihak lain yang dapat menyadap informasi dan dapat dengan mudah membaca isi dokumen tersebut. Salah satu cara pengamanan yang dapat digunakan untuk mengamankan informasi yaitu, metode kriptografi. Atas dasar hal tersebut, tugas akhir ini akan membahas mengenai pembuatan aplikasi yang dapat mengamankan suatu dokumen digital dengan mengimplementasikan algoritma kriptografi AES (*Advanced Encryption Standard*) dengan menggunakan kunci simetri. Dimana metode AES adalah algoritma chipher blok yang menggunakan teknik substitusi, pemutasian dan sejumlah putaran pada setiap blok yang akan di enkripsi dan deskripsi. Pada proses pembuatan aplikasi enkripsi dekripsi ini menggunakan software Microsoft Visual Studio dengan Bahasa Pemrograman PHP dan MySql sebagai *Database Manage-*

DOI : http://dx.doi.org/10.32409/jikstik.21.3.2973, *) Penulis Korespondensi

ment System (DBMS).

Masalah yang dibahas dalam Penelitian ini adalah bagaimana membuat aplikasi enkripsi dan dekripsi yang dapat mengamankan suatu dokumen digital dengan menggunakan algoritma kriptografi AES-128 (*Advanced Encryption Standard* 128 bit) dengan menggunakan kunci simetri, yang diimplementasikan dengan menggunakan bahasa pemrograman PHP. Adapun batasan masalah yang diangkat pada penelitian ini adalah memfokuskan untuk mengenkripsi dokumen digital dengan tipe file yang memiliki ekstensi PDF, DOCX, PPTX, XLSX, dan TXT yang memiliki ukuran maksimal sebesar 3 MB.

Tujuan dari penelitian ini adalah untuk merancang dan membuat aplikasi dengan mengimplemetasikan algoritma kriptografi AES-128 dengan kunci simetri untuk melakukan enkripsi dan dekripsi pada berbagai jenis file dokumen digital. Aplikasi ini diharapkan dapat membantu dan memudahkan para pengguna untuk menjaga keaslian dari suatu dokumen.

Kriptografi

Kata kriptografi (*Cryptography*) berasal dari kata Yunani, yaitu kata *Cryptos* yang berarti tersembunyi dan *Graphein* berarti menulis. Kriptografi dapat didefinisikan sebagai ilmu pengetahuan Atau seni mengeksplorasi bagaimana data diubah menjadi format tertentu yang sulit dipahami [2] .

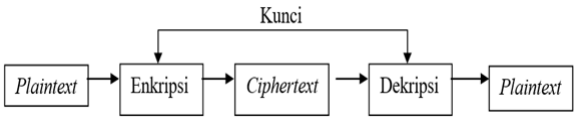
Kriptografi itu sendiri adalah ilmu sekaligus seni keamanan data, dan bersifat spesifik karena bertujuan untuk menimbulkan kerancuan atau kerancuan dengan mengubah teks biasa (*plaintext*) menjadi teks rahasia (*ciphertext)* yang tidak dapat dibaca. Enkripsi memiliki proses penyandian yang dapat mengubah teks atau data (*plain text*) menjadi teks rahasia (*ciphertext)* dan sebaliknya, dari teks rahasia (*ciphertext)* menjadi teks atau data (*plain text*), dikembalikan. Enkripsi menggunakan algoritma (enkripsi) dan kunci (*keys*) untuk mengenkripsi dan mendekripsi data. Kriptografi adalah fungsi matematika yang digunakan untuk enkripsi dan dekripsi, dan kuncinya adalah urutan bit yang diperlukan untuk mengenkripsi dan mendekripsi data. [3].

Kriptografi menggunakan suatu algoritma (*cipher*) dan kunci *(key*) untuk mengenkripsi dan mendekripsi data. *Cipher* adalah fungsi matematika yang digunakan untuk mengenkripsi dan mendekripsi, sedangkan kunci merupakan sederetan *bit* yang diperlukan untuk mengenkripsi dan mendekripsi data. Secara sederhana tahapan tersebut digambarkan pada Gambar 1.

Algoritma Kriptografi

Kriptografi memiliki berbagai macam algoritma yang banyak digunakan untuk melindungi informasi. Algoritma kriptografi terbagi menjadi dua jenis: algoritma kriptografi tradisional dan modern. Saat beroperasi, algoritma kriptografi tradisional bekerja dalam mode karakter, sedangkan algoritma kriptografi modern bekerja dalam mode bit[4]. Kriptografi klasik merupakan kriptografi yang sudah dikembangkan bahkan sejak belum ada komputer. Beberapa metode kriptografi klasik adalah *substitution cipher* (Teknik Subtitusi) dan *transposition cipher* (Teknik Transposisi / Permutasi).

Algoritma enkripsi modern adalah peningkatan yang terkait dengan enkripsi tradisional. Contoh enkripsi modern termasuk MD5, RC4, dan AES. Algoritma ini memiliki tingkat kesulitan yang kompleks sehingga sangat sulit bagi seorang *cryptanalyst* untuk memecahkan sebuah *ciphertext* tanpa mengetahui kuncinya. Ada tiga jenis kunci dalam enkripsi modern: simetris, asimetris, dan hibrida.



Gambar 1: Enkripsi dan Dekripsi Sederhana

Advance Encryption Standard (AES)

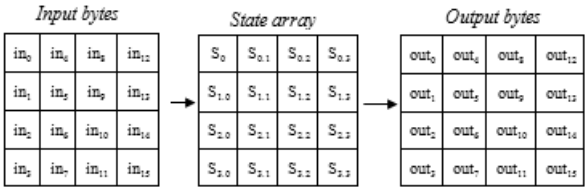
Sebuah algoritma kriptografi yang disebut Rijndael, yang dirancang oleh Vincent Rijmen dan John Daemen dari Belgia, sebagai juara pada *Cryptographic Algorithm Contest* sebagai pengganti DES. Algoritma *Rijndael* telah dikenal sebagai *Advanced Encryption Standard* (AES). Setelah beberapa proses standarisasi oleh NIST, Rijndael kemudian diadopsi sebagai standar algoritma enkripsi resmi pada 22 Mei 2002 [5]. *Advanced Encryption Standard* (AES) termasuk dalam enkripsi kunci simetris dan algoritma *block cipher*. Algoritma ini menggunakan kunci yang sama untuk enkripsi dan dekripsi [6]. Input dan output dari algoritma AES terdiri dari urutan data 128-bit. Urutan data yang dibentuk oleh sekelompok 128 bit juga disebut blok data atau *Plaintext*. *Plaintext* dienkripsi dalam menjadi *ciphertext*. Setiap putaran, algoritma AES menggunakan kunci yang berbeda. Kunci untuk setiap putaran disebut *round key*. Tabel 1 menunjukkan jumlah putaran (Nr) yang perlu dilaksanakan. dengan panjang kunci apa pun [7].

Tabel 1: *Jumlah round/putaran (Nr)*

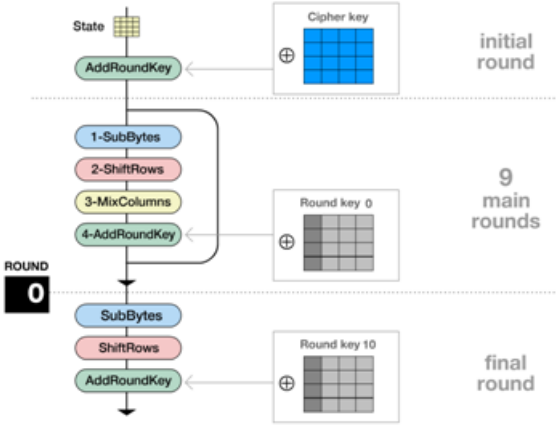
Tipe AES	Jumlah Key (Nk)	Besar Blok (Nb)	Jumlah Round (Nr
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Tabel 1 menunjukkan jenis algoritma AES dengan panjang kunci yang berbeda, panjang blok, dan jumlah putaran. Penelitian ini menggunakan AES-128 bit dengan 10 putaran enkripsi.

Sebelum memulai proses enkripsi atau dekripsi, ada proses yang disebut *state*. Keadaan AES ini merupakan implementasi dari operasi AES yang dilakukan pada *array byte* dua dimensi. Ukuran pada *state* adalah konfigurasi *NROWS XNCOLS*. Pada awal proses enkripsi, data pesan terlebih dahulu diubah ke dalam format *heksadesimal* kemudian dimasukkan ke dalam matriks 4x4. Matriks tersebut berisi 1 *byte* (8 bit) dalam format in0, in2, in3, in4, in5, in6. , In7, in8, in9, in10, in11, in12, in13, in14, in15 disalin ke *array state*. *State* ini nantinya akan berperan dalam operasi enkripsi dan dekripsi. Kemudian *output* ditempatkan di *array out*. Gambar 2 menunjukkan proses *input byte* , *state array* , dan *output byte* .



Gambar 2: Proses *Input Bytes, State Array,* dan *Output Bytes*



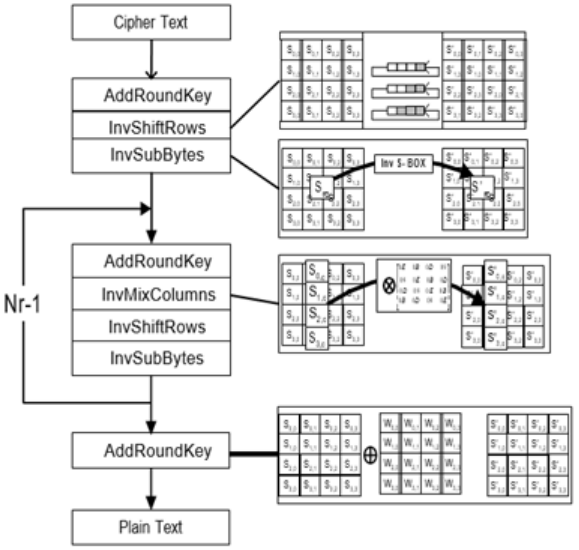
Gambar 3: Proses enkripsi dengan algoritma AES-128

Proses Enkripsi AES

Pada awal proses enkripsi, input yang disalin ke *state* mengalami konversi *byte AddRoundKey*. kemudian *State* secara berulang sebanyak Nr kali mengalami transformasi *SubBytes, ShiftRows, MixColumns,* dan *AddRoundKey*. Proses algoritma AES ini disebut *round function*. *round* terakhir sedikit berbeda dengan babak sebelumnya dimana keadaan babak terakhir belum mengalami *transformasi MixColumns* [8][9]. Gambar 3 menunjukkan diagram proses enkripsi awal menggunakan algoritma AES-128.

Proses Dekripsi AES

Proses dekripsi berlawanan dengan proses enkripsi. Transformasi *bytes* yang digunakan dalam proses dekripsi yaitu *AddRoundKey, InvShiftRows, InvSubBytes*, dan *InvMixColumns*. Pada proses dekripsi, untuk iterasi pertama dilakukan transformasi *AddRoundKey, Inverse ShiftRows*, dan *Inverse SubBytes*. *Chiper text* akan melakukan transformasi *AddRoundKey*. Pada Gambar 4 menunjukkan tahapan proses dekripsi algoritma AES [8][9].



Gambar 4: Proses dekkripsi dengan algoritma AES-128

Metode

Metode perancangan sistem menggunakan *System Developmet Life Cycle* (SDLC). Berikut tahapan sistem SDLC :

1. Tahap Perencanaan Pada tahap ini dilakukan identifikasi sistem dan prosedur yang ingin dibuat
2. Analisis masalah Mempelajari data yang diperoleh dari sistem yang sedang beroperasi, kemudian melakukan analisa permasalahan yang terjadi
3. Perancangan Aplikasi Pada proses ini dimulai dengan mendesain tampilan didalam komputer sebagai gambaran awal yang memperlihatkan tampilan program.
4. Pembuatan Aplikasi Di tahap ini desain program di buat menggunakan software Mi-

crosoft Visual Studio dengan Bahasa Pemrograman PHP dan menggunakan MySql sebagai Database Management System (DBMS).

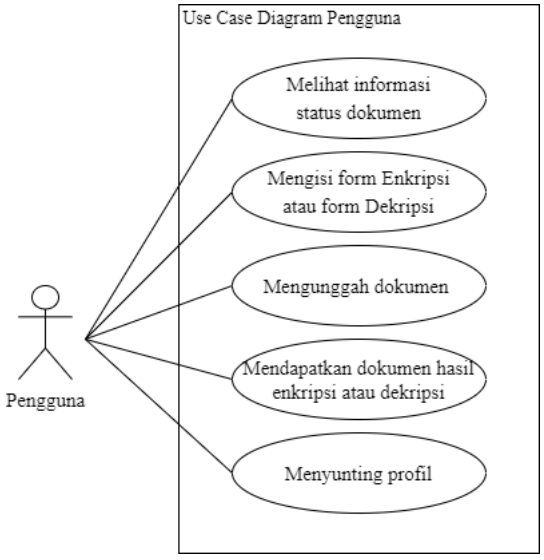
5. Uji Coba Pada tahap ini aplikasi akan dicoba dengan menggunakan beberapa file dokumen digital.

Pembuatan Aplikasi

Dalam proses pembuatan aplikasi website enkripsi dokumen digital ini, dirancang sebuah program interaktif dengan menggunakan software Microsoft Visual Studio dengan Bahasa Pemrograman PHP dan menggunakan MySql sebagai *Database Management System* (DBMS).

Use Case Diagram Pengguna

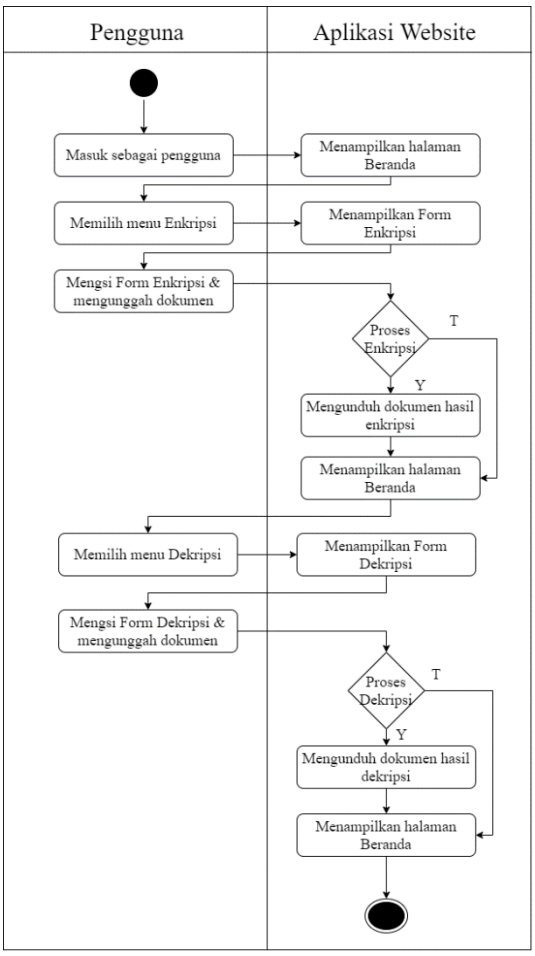
Use case diagram digunakan untuk mencari tahu interaksi antara sistem dan pengguna. Adapun struktur use case diagram pengguna yang menjelaskan interaksi antara pengguna dengan fungsionalitas system dari aplikasi ini dapat dilihat pada Gambar 5.



Gambar 5: Use Case Diagram Pengguna.

Activity Diagram Pengguna

Activity diagram pengguna ini digunakan untuk menggambarkan aktivitas dan tindakan terstruktur antara interaksi pengguna dengan system. Adapun activity diagram dari aplikasi ini dapat dilihat pada Gambar 6.



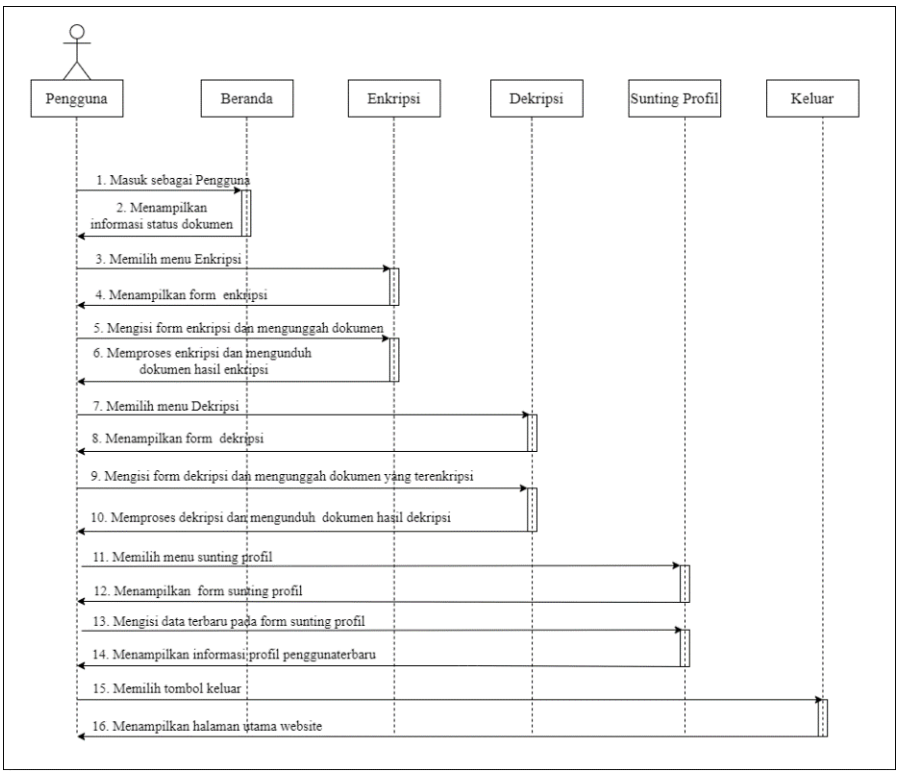
Gambar 6: Activity Diagram Pengguna.

Sequence Diagram Pengguna

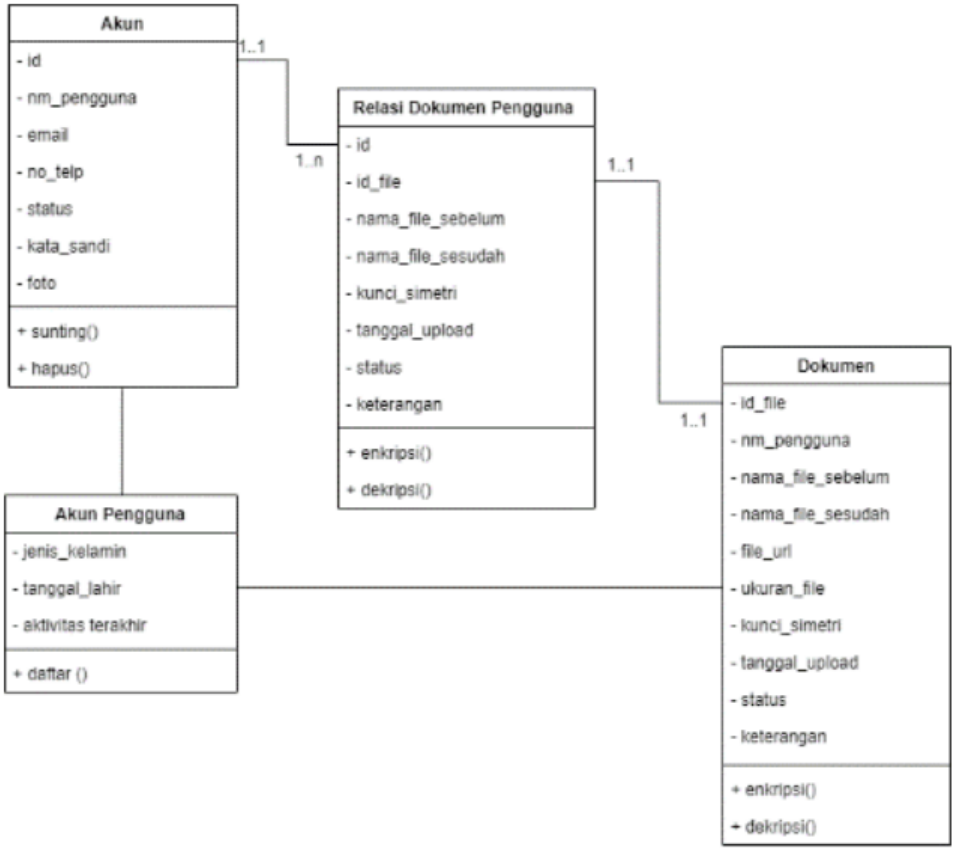
Sequence diagram pengguna menjelaskan interaksi yang dapat dilakukan oleh pengguna terhadap sistem. Adapun gambar sequence diagram pengguna dapat dilihat pada Gambar 7.

Class Diagram

Class diagram digunakan untuk memperlihatkan struktur database yang ada pada sistem. Class diagram yang terdapat pada website Documencryption dapat dilihat pada Gambar 8. Pada gambar 8 terdapat *Class Diagram* yang digunakan dalam pembuatan *Website* Documentcryption. Class dokumen merupakan komposisi dari subclass akun pengguna. Karena untuk melakukan sebuah enkripsi atau dekripsi dokumen, pengguna harus memiliki akun terlebih dahulu. Sehingga, jika *subclass* akun pengguna ditiadakan, maka *class* dokumen juga tidak akan terbentuk. Sedangkan *Class* Relasi Dokumen Pengguna merupakan class yang merelasikan antara *class* akun dengan *class* dokumen. Dimana setiap satu akun dapat melakukan banyak proses enkripsi dan dekripsi, dan setiap dokumen yang diproses hanya dapat memiliki satu id_file saja.



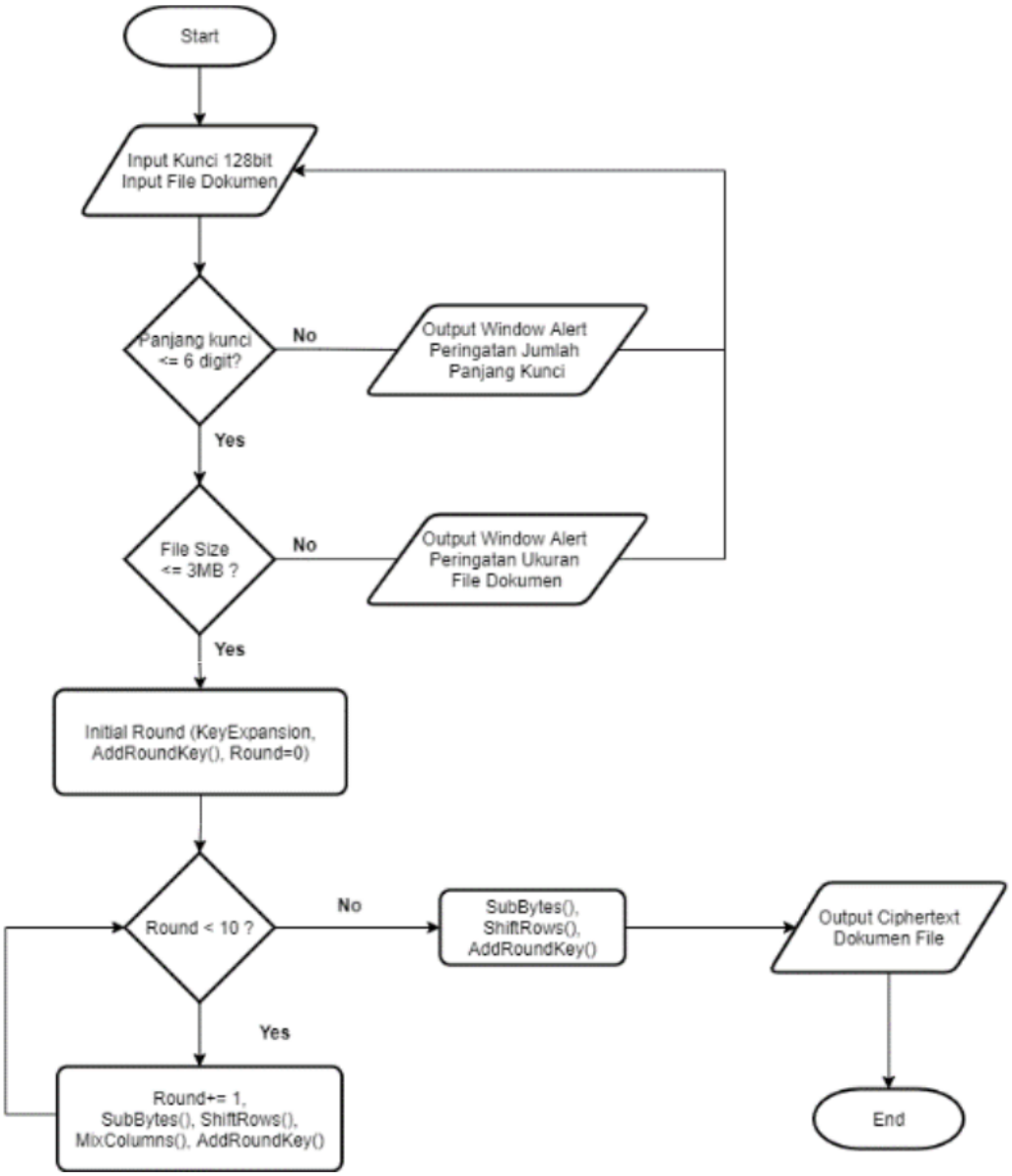
Gambar 7: Sequence Diagram Pengguna



Gambar 8: Class Diagram

Perancangan Algoritma Program

Perancangan algoritma program aplikasi, akan digambarkan dengan menggunakan flowchart. Bagan ini menjelaskan urut-urutan langkah dan proses prancangan sistem. Selain itu, *flowchart* juga memiliki fungsi memudahkan proses pengecekan terhadap sistem yang akan dibuat. Berikut merupakan *flowchart* dari sistem yang dibangun. Gambar.9 dan Gambar.10 menampilkan *flowchart* Enkripsi dan Dekripsi dari Sistem.

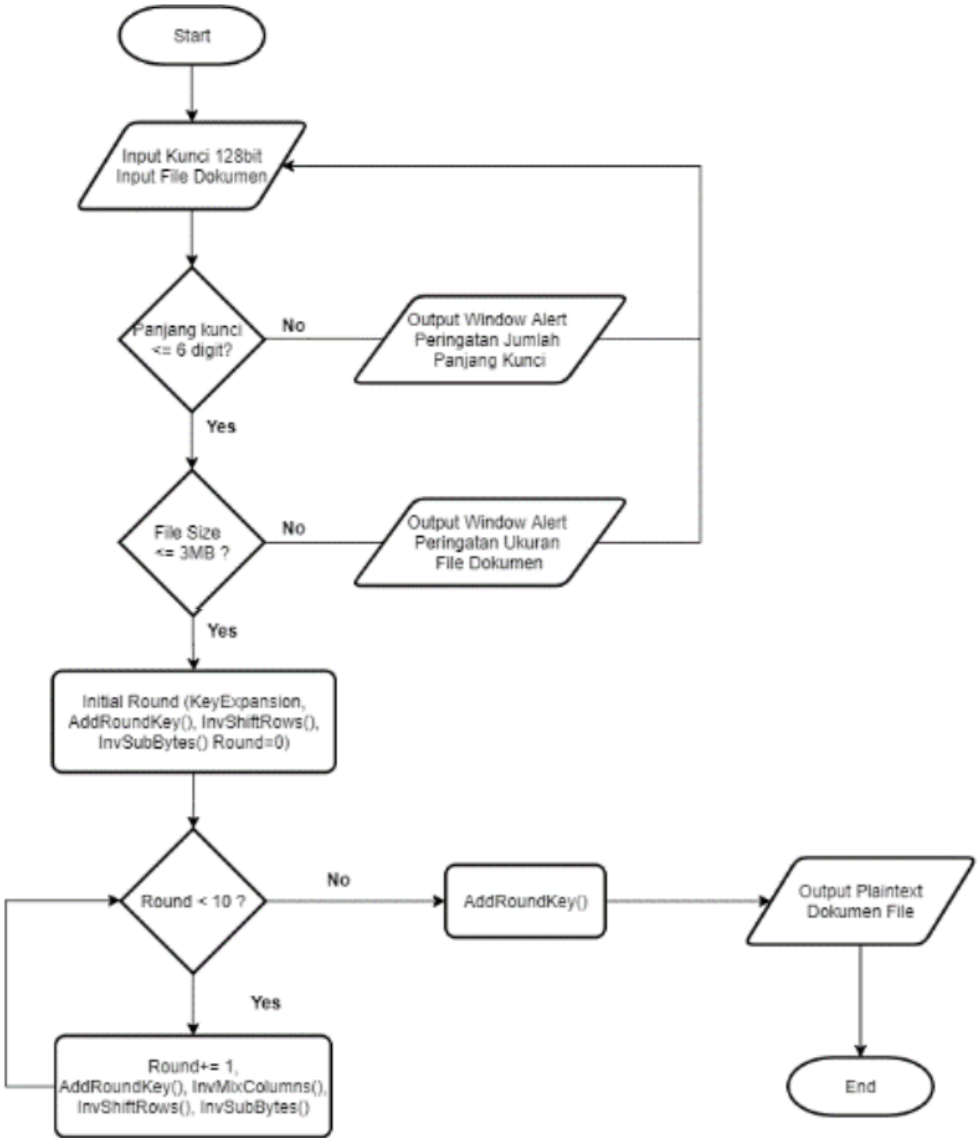


Gambar 9: Flowchart Enkripsi AES 128 bit

Implementasi dan Ujicoba

Tampilan Halaman Utama

Ketika pengguna membuka website Documencryption, halaman pertama yang akan ditampilkan yaitu halaman utama. Pada halaman ini, pengguna baru dapat langsung mendaftarkan diri untuk membuat akun baru dengan mengisi form pendaftaran yang tersedia. Pada halaman ini, pengguna yang sudah memiliki akun juga dapat masuk ke aplikasi website dengan mengisi Nama Pengguna dan Kata sandi untuk dapat melakukan enkripsi ataupun dekripsi dokumen. Tampilan halaman utama website Documencryption dapat dilihat pada Gambar 11.



Gambar 10: Flowchart dekripsi AES 128 bit

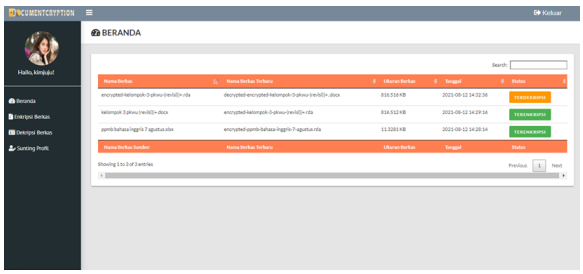
Gambar 11: Tampilan Halaman Utama.

Tampilan Halaman Beranda

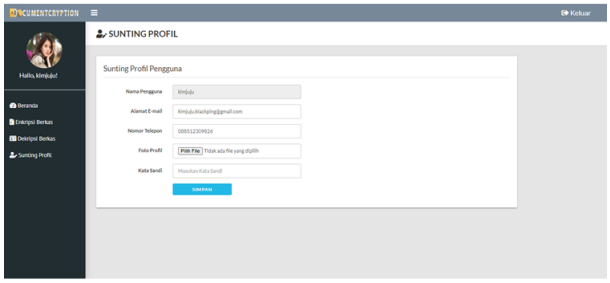
Setelah berhasil masuk sebagai pengguna, halaman web yang akan pertama kali ditampilkan adalah halaman beranda. Pada halaman ini, terdapat tabel yang berisikan daftar dokumen milik pengguna, yang sebelumya pernah diproses oleh website documencryption. Informasi yang ditampilkan pada tabel ini antara lain yaitu Nama Berkas, Nama Berkas Terbaru, Ukuran Berkas, Tanggal, dan Status. Tampilan halaman beranda pengguna dapat dilihat pada Gambar12.

Tampilan Halaman Enkripsi

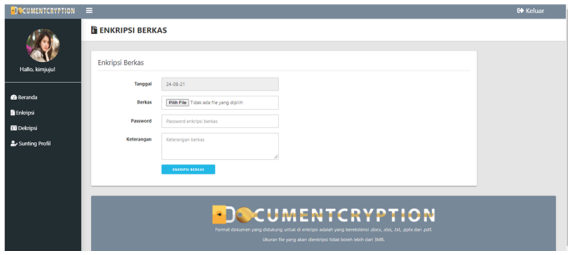
Pada Halaman Enkripsi, pengguna dapat melakukan enkripsi dokumen digital dengan cara mengisi form enkripsi dan mengunggah dokumen dari penyimpanan internal serta memasukan kata sandi sebagai kunci simetri yang digunakan untuk mengenkripsi dokumen. Tampilan halaman enkripsi dokumen ditujukan pada Gambar 13.



Gambar 12: Tampilan Halaman Beranda



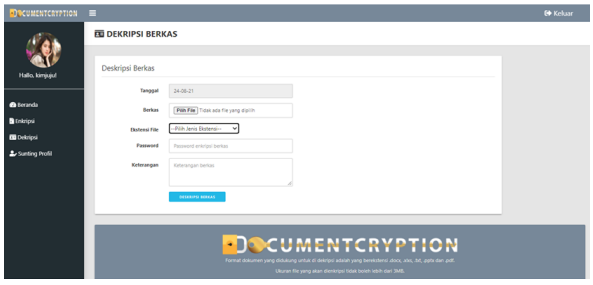
Gambar 15: Tampilan Halaman Sunting Profil Pengguna.



Gambar 13: Tampilan Halaman Enkripsi

Tampilan Halaman Dekripsi

Pada Halaman Dekripsi, pengguna dapat melakukan dekripsi dokumen digital dengan cara mengisi form enkripsi dan mengunggah dokumen digital yang telah terenkripsi dari penyimpanan internal serta memasukan kata sandi sebagai kunci simetri yang digunakan untuk mendekripsi dokumen. Tampilan halaman dekripsi dokumen dapat dilihat pada Gambar 14.



Gambar 14: Tampilan Halaman Deskripsi

Tampilan Halaman Sunting Profil

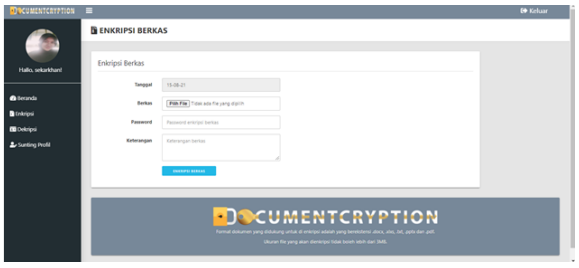
Pada halaman Sunting Profil, pengguna dapat memperbarui informasi data diri. Selain itu, ada halaman ini pengguna juga dapat mengganti foto profil dengan cara mengunggah foto dari penyimpanan internal. Tampilan halaman sunting profil pengguna, dapat dilihat pada Gambar 15.

Uji Coba Sistem

Pengujian sistem adalah proses menjalankan sistem perangkat lunak untuk menentukan apakah sistem memenuhi spesifikasi sistem dan berjalan di lingkungan yang diinginkan. Pengujian dilakukan dengan menguji setiap proses dan kesalahan yang mungkin terjadi pada setiap proses. Sistem pengujian yang digunakan adalah *black box*.[10] Pengujian kotak hitam menguji perangkat lunak terhadap spesifikasi fungsional tanpa menguji desain atau kode program. Tujuan pengujian adalah untuk menentukan apakah kemampuan input dan output perangkat lunak memenuhi spesifikasi yang dibutuhkan.

Pengujian Proses Enkripsi

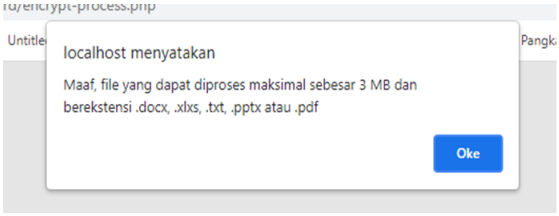
Melalui halaman Menu Enkripsi, pengguna pertama dapat melakukan enkripsi dokumen digital dengan cara mengisi form enkripsi dan mengunggah dokumen yang akan diproses dari penyimpanan internal. Pada kesempatan kali ini, dipilih dokumen yang berekstensi .pdf sebagai contoh uji coba proses enkripsi. Adapun proses pengisian form enkripsi dapat dilihat pada Gambar 16.



Gambar 16: Proses pengisian form enkripsi

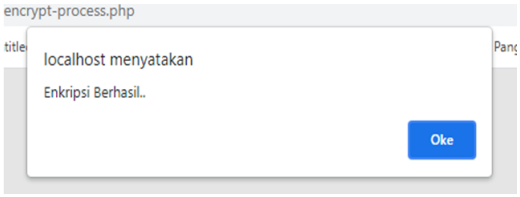
Pada form enkripsi yang tertera pada menu enkripsi berkas, pengguna dapat mengunggah dokumen yang akan di enkripsi, memasukan *password* sebagai kunci simetri, dan memasukan keterangan sebagai catatatan pribadi pengguna.

Jenis dokumen yang dapat diproses yaitu dokumen yang memiliki ukuran maksimal sebesar 3MB dan berekstensi .docx, .xlxs, .pptx, .txt, dan .pdf. Jika dokumen yang akan diproses memiliki ekstensi yang lain atau berukuran lebih besar dari 3MB, maka system akan menampilkan *Window Alert* yang menginformasikan jenis-jenis file dan besar ukuran file yang dapat diproses. Adapun *window Alert* yang akan ditampilkan dapat dilihat pada Gambar 17.



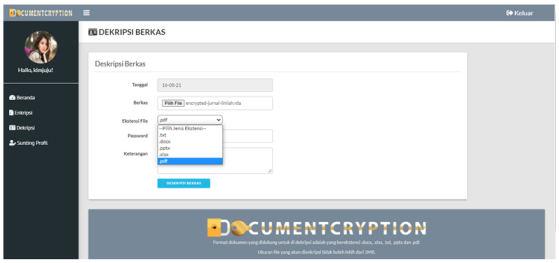
Gambar 17: *Window alert* jika dokumen tidak sesuai

Jika dokumen yang akan diproses sudah sesuai, maka proses enkripsi akan segera dimulai. Setelah system berhasil membaca isi dokumen, Semua karakter yang terdapat pada dokumen akan dikonversi menjadi blok bilangan hexadesimal berukuran 128 bit yang kemudian akan diubah kedalam bentuk matriks dua dimensi yang berukuran 4x4 bernama matriks *state*. Sebelum masuk kedalam putaran yang pertama, matriks state terlebih dahulu di XOR kan dengan kunci putaran yang ke-0 (rk0) atau *Cipher key*. *Cipher key* merupakan kunci eksternal yang diberikan oleh pengguna yang panjangnya juga 128 bit. Operasi XOR antara state dengan kunci putaran yang ke-0 dinyatakan dalam prosedur *AddRoundKey*. Setelah itu matriks *state* akan melewati 10 putaran. Pada putaran yang pertama sampai putaran yang kesembilan, matriks *state* akan mengalami empat transformasi, yaitu Transformasi *SubBytes*, Transformasi *ShiftRows*, Transformasi *MixColumns*, dan Transformasi *AddRoundKey,* yaitu XOR antara matriks *state* dengan kunci putaran yang ke-r. sedangkan pada putaran yang terakhir, hanya ada tiga transformasi, yaitu Transformasi *SubBytes*, Transformasi *ShiftRows*, dan Transformasi *AddRoundKey*, yaitu XOR antara matriks state dengan kunci putaran yang terakhir. Selanjutnya, matriks state akan akan diterjemahkan ke dalam kode ASCII untuk mendapatkan karakter hasil enkripsinya. Setelah berhasil melalui proses enkripsi, maka sistem akan menampilkan *window alert* yang menginformasikan bahwa proses enkripsi telah berhasil dilakukan dan dokumen yang telah terenkripsi akan secara otomatis tersimpan pada penyimpanan internal. Adapun *window alert* yang akan ditampilkan dapat dilihat pada Gambar18.



Gambar 18: *Window alert* proses enkripsi berhasil

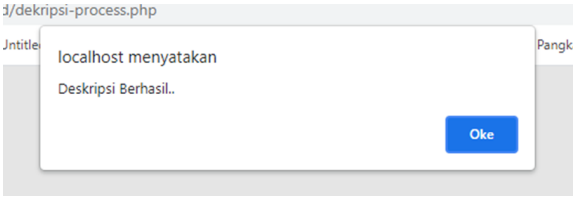
Pengujian Proses Dekripsi Dekripsi merupakan kebalikan dari proses enkripsi. Melalui halaman Menu Dekripsi, pengguna kedua dapat melakukan dekripsi dokumen digital dengan cara mengisi *form* dekripsi dan mengunggah dokumen yang terenkripsi dari penyimpanan internal. Setelah itu, pengguna harus memilih ekstensi file yang sesuai agar dokumen yang terenkripsi berhasil didekripsi. Selain itu, pengguna juga harus mengetahui kata sandi enkripsi, karena kata sandi tersebut merupakan kunci simetri yang akan digunakan untuk mendekripsi dokumen. Adapun proses pengisian *form* dekripsi dapat dilihat pada Gambar 19.



Gambar 19: . Proses pengisian *form* Deskripsi

Pada *form* dekripsi yang tertera pada menu dekripsi, pengguna dapat mengunggah dokumen yang akan di dekripsi, memilih jenis ekstensi file yang sesuai, memasukan password sebagai kunci simetri, dan memasukan keterangan sebagai catatatan pribadi. Setelah pengguna menekan tombol *ENKRIPSI BERKAS,* maka proses enkripsi akan segera dimulai. Sama seperti proses enkripsi, semua karakter yang terdapat pada dokumen akan dikonversi menjadi blok bilangan hexadecimal berukuran 128 bit yang kemudian akan diubah menjadi matriks *state*. Selanjutnya matriks ini akan mengalami empat transformasi pada setiap putaran, yaitu Transformasi *AddRoundKey,* Transformasi *InvShitftRows*, Transformasi *InvSubBytes*, dan Transformasi *InvMixColumns*. Sedangkan untuk putaran terakhir hanya ada tiga transformasi yaitu Transformasi *InvShitftRows*, Transformasi *InvSubBytes*, dan Transformasi *AddRoundKey*. Jika dokumen yang akan diproses berhasil di proses, maka system akan menampilkan *window alert* yang menginformasikan bahwa proses dekripsi telah berhasil dilakukan dan dokumen yang telah

terdekripsi akan secara otomatis tersimpan pada penyimpanan internal. Adapun *window alert* yang akan ditampilkan dapat dilihat pada Gambar 20.



Gambar 20: *Window alert* proses dekripsi berhasil

Hasil Uji Coba

Hasil uji coba yang didapatkan melalui penilitian ini berupa hasil dari proses enkripsi dan dekripsi dokumen digital. Adapun hasil uji coba tersebut adalah sebagai berikut :

Proses Enkripsi

Berkas yang berhasil dienkripsi, akan tersimpan secara otomatis pada penyimpanan internal dengan format ekstensi baru, yaitu .rda. dokumen yang telah terenkripsi dapat dibuka dengan menggunakan aplikasi notepad. Adapun hasil dari enkripsi file pdf yang telah dienkripsi sebelumnya, dapat dilihat pada Gambar 21.

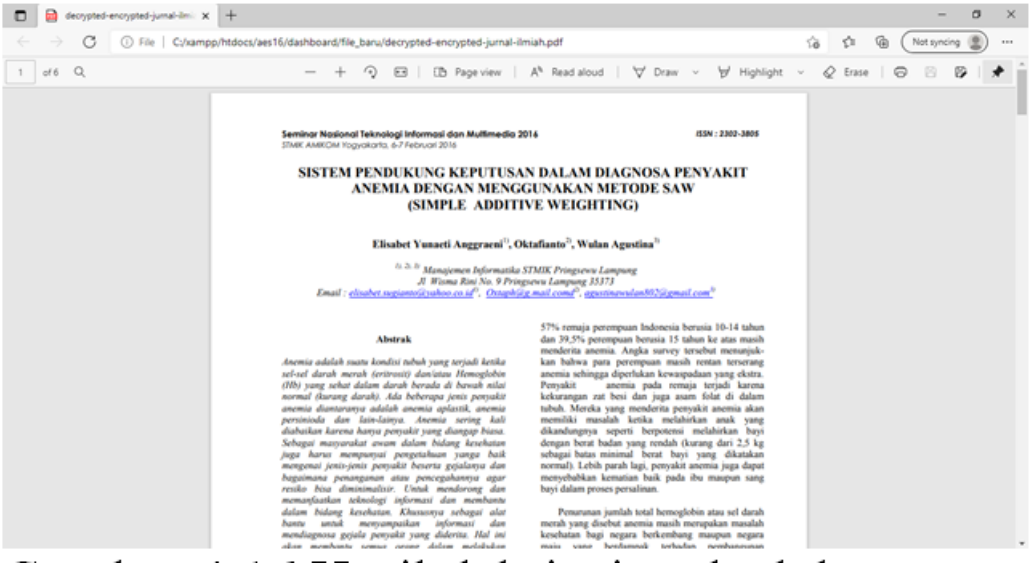
Gambar 21: Hasil enkripsi pada dokumen berekstensi .pdf

Gambar 21 menunjukan hasil dari enkripsi dokumen Jurnal Ilmiah.pdf yang sebelumnya berukuran 480 KB menjadi dokumen dengan nama file encrypted-jurnal-ilmiah.rda yang berukuran 480 KB. Pada gambar 21 juga menunjukan bahwa proses enkripsi telah berhasil dilakukan, sehingga file dokumen yang terenkripsi menampilkan deretan ciphertext yang tidak dapat dimengerti dan menjadi lebih aman untuk dikirim kepada penerima dokumen, jika pengiriman dilakukan melalui jaringan internet.

Proses Dekripsi

Setelah berhasil melakukan proses dekripsi, berkas hasil dekripsi juga akan tersimpan secara otomatis pada penyimpanan internal dengan format ekstensi yang telah dipilih, sesuai dengan ekstensi dokumen aslinya. Adapun hasil dari enkripsi file pdf yang telah dienkripsi sebelumnya, dapat dilihat pada Gambar 22.

Gambar 22 menunjukan hasil dari dekripsi dokumen encrypted-jurnal-ilmiah.rda yang sebelumnya berukuran 480 KB menjadi dokumen dengan nama file decrypted-encrypted-jurnal-ilmiah.pdf yang berukuran 480 KB. Gambar 22 juga menunjukan bahwa proses dekripsi telah berhasil dilakukan, sehingga file dokumen yang terenkripsi dapat kembali menunjukan deretan *plaintext* yang dapat dibaca oleh penerima dokumen.



Gambar 22: Hasil dekripsi pada dokumen terenkripsi.

Penutup

Berdasarkan hasil uji coba pada pengujian *Blackbox* yang telah dilakukan, dapat diketahui bahwa aplikasi web yang dibuat dengan menggunakan metode kriptografi AES-128 dengan kunci simetri untuk melakukan enkripsi dan dekripsi berbagai jenis file dokumen digital telah berhasil dibangun dan diberi nama Documentcryption. Dokumen digital berhasil dienkripsi oleh pengguna pertama dan berhasil didekripsi kembali oleh pengguna kedua. Dengan menggunakan website Documentcryption, pengguna pertama dan pengguna kedua harus saling berkomunikasi mengenai kunci simetri dan ekstensi dokumen yang akan diproses. Ukuran file dokumen asli, ukuran file dokumen yang terenkripsi, dan ukuran file dokumen yang didekripsi, tidak mengalami perubahan ukuran yang signifikan. Berdasarkan hasil uji coba juga dapat diketahui bahwa kecepatan proses enkripsi dan dekripsi dipengarui oleh besar ukuran dokumen yang diproses. Semakin kecil ukurannya, maka semakin cepat pula waktu proses yang dibutuhkan. Selain itu, dapat diketahui bahwa browser Google Chrome dapat melakukan proses lebih cepat daripada browser Microsoft Edge, sehingga pengguna disarankan menggunakan Google Chrom untuk mengakses website Documentcryption.

Daftar Pustaka

[1] Asri Prameshwari dan Nyoman Putra Sastra, "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen", Jurnal Eksplora Informatika, Vol 8, No 1, 2018.

[2] M. Sigit Prasetyo, "Implementasi Algoritma Advance Encryption Standard (AES) Rijndael Untuk Proteksi File Audio", Skripsi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, Medan, 2016.

[3] Susanto dan Andri Anto Tri Susilo, "Penerapan Algoritma Asimetris RSA Untuk Keamanan Data Pada Aplikasi Penjualan CV. Sinergi Komputer Lubuk Linggau Berbasis WEB", Jurnal Simetris, Vol.9 No.2, Program Studi Teknik Informatika, STMIK MUSIRAWAS, Lubuk Linggau, Sumatra Utara, 2018.

[4] Yusfrizal, "Rancang Bangun Aplikasi Kriptografi Pada Teks Menggunakan Metode Reverse Chiper dan RSA Berbasis Android", Jurnal Teknik Informatika Kaputama (JTIK), Vol. 3, No. 2, Universitas Potensi Utama, Medan, Juli 2019.

[5] Gilang Gumira P. U. K., Ernawati, dan Aan Erlanshari, "Implementasi Metode Advanced Encryption Standard (AES) dan Message Digest 5(MD5) Pada Enkripsi Dokumen (Studi Kasus LPSE UNIB)", Jurnal Rekursif, Vol.4 No.3, Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu, 2016.

[6] Yulius Rio Pujianto, "Perancangan dan Implementasi Aplikasi Kriptografi Algoritma AES-128 Pada File Dokumen", Skripsi, Fakultas Teknologi Informasi Universitas Kristen SatyaWacana, Salatiga, 2016.

[7] Anggraeni Eka Putri, Aghistina Kartikadewi, Lina Audia Abdul Rosyid, "Implementasi Kriptografi Dengan Algoritma Advanced Encryption Standard (AES) 128 bit dan Steganografi Menggunakan Metode End of

File (EOF) Berbasis Java Desktop Pada DinasPendidikan Kabupaten Tangerang", Applied Information Systems and Management (AISM), Volume 3, Universitas Budi Luhur, 2020.

[8] Intan Fitriani, "Implementasi Algoritma Advanced Encryption Standard (AES) Pada Layanan SMS Desa", JISKA (Jurnal Informatika Sunan Kalijaga), 5(3):153, DOI:10.14421/jiska.2020.53-032020, 2020.

[9] S. Widyastuti, W. Ariandi & V. Sulistiono, "Implementasi Kriptografi AES Dalam Penganaman Data Seleksi Peserta JAMKESMAS. Jurnal Ilmiah Intech : Information Technology Journal of UMUS, 1(02), 13–22, https://doi.org/10.46772/intech.v1i02.66, 2019.

[10] Fadhila Cahya Ningrum, Dandi Suherman, Sita Aryanti, Handika Arngga Prasetya, dan Aries Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions", Jurnal Informatika Universitas Pamulang, Vol.04, No.04, Teknik Informatika, Universitas Pamulang, Banten, 2019.