Implementasi Secure Multi-Party Computation Menggunakan Metode Shamir Secret Sharing pada Pengamanan Dokumen Digital Rahasia

Willy Sudiarto Raharjo[1], Antonius Rachmat C.[2], Pedro Nadirio A.[3]

***Abstrak*—Keamanan data merupakan aspek penting yang perlu diperhatikan dalam proses penyimpanan data. Pada beberapa situasi, terdapat data rahasia yang hanya boleh diakses apabila terdapat sejumlah pihak yang memiliki hak akses telah memberikan kontribusinya untuk mengakses data rahasia tersebut. Pada publikasi ini, kami mengimplementasikan sebuah sistem pengamanan data berbasis *Multi-Party Computation* dengan memanfaatkan skema *Shamir Secret Sharing*. Sistem yang dibangun telah mampu membagi data rahasia menjadi N potongan dan hanya setelah N potongan tersebut digabungkan kembali, data asli bisa didapatkan kembali. Kami juga mampu mengatasi sebuah persoalan integritas data yang belum dapat diatasi oleh *Shamir Secret Sharing* yaitu dengan memanfaatkan Hashed Key.**

***Kata Kunci*: Dokumen Digital Rahasia, Keamanan Data, Skema Shamir Secret Sharing, Multi-Party Computation.**

***Abstract*— Data security is an important aspect that needs to be considered in data storing process. In some cases, there are data that can only accessed if the number of valid participants gave their contributions to access that data. In this paper, we implement a data security system based on Multi-Party Computation that utilize Shamir Secret Sharing Scheme. The proposed system is able to divide the secret data into N pieces and only after joining all N pieces together, the original data can be retrieved. We also solved a data integrity issue that can't be solved by Shamir Secret Sharing by using Hashed Key.**

Keywords: Digital Private Document, Data Security, Shamir Secret Sharing Scheme, Multi-party Computation.

I. PENDAHULUAN

Keamanan sebuah data yang sifatnya rahasia (*confidential*) merupakan suatu hal yang penting dan harus menjadi pertimbangan pada era digital saat ini. Setiap pengguna yang memiliki data rahasia ingin mendapatkan jaminan kepastian bahwa data mereka tidak dapat diakses atau bahkan diubah seenaknya oleh pihak tertentu dengan tujuan yang tidak baik. Setiap data seharusnya hanya dapat diakses oleh pemilik data tersebut. Pada beberapa kasus, misalnya dalam kasus surat wasiat, terdapat data rahasia yang hanya dapat diakses apabila terdapat sejumlah pihak yang memiliki hak akses telah memberikan kontribusinya untuk mengakses rahasia tersebut. Apabila pihak yang akan mengakses sebuah rahasia berjumlah kurang dari nilai yang telah tentukan, maka rahasia tersebut tidak dapat diakses. Proses pengaksesan data secara berkelompok tersebut tidak dapat dilakukan oleh sembarang pihak yang tidak memiliki hak akses atau mencoba mengakses data rahasia dengan cara yang tidak semestinya.

Melihat fakta bahwa pengamanan data menjadi sebuah hal yang penting, muncul sebuah gagasan untuk membuat sebuah sistem pengamanan untuk melindungi data tersebut dari pihak yang tidak bertanggung jawab. Data rahasia dalam bentuk digital akan diamankan dengan metode yang terdapat pada bidang ilmu kriptografi sehingga pada akhirnya data tersebut tidak dapat diakses secara sembarangan. Setiap data rahasia yang akan diamankan akan dipecah menjadi beberapa bagian secara mandiri dimana setiap bagian akan diberikan kepada pihak yang memiliki hak akses terhadap rahasia tersebut. Untuk membentuk kembali informasi rahasia yang telah diamankan, diperlukan kontribusi setiap pihak yang memiliki akses dan memiliki bagian dari rahasia sesuai dengan yang telah dibagikan. Informasi rahasia hanya dapat terbentuk kembali apabila terdapat bagian rahasia yang valid dalam jumlah tertentu serta setiap bagian sudah terverifikasi dengan tepat.

Penelitian ini berfokus pada perancangan sebuah sistem pengamanan yang mengkombinasikan teknik kriptografi dan protokol *secure multi-party computation*. Kriptografi digunakan untuk melakukan enkripsi pada data dimana algoritma yang akan digunakan adalah algoritma AES (*Advanced Encryption Standard*) yang masih tergolong aman sampai dengan hari ini. Protokol *secure multi-party computation* digunakan untuk membagi data yang sudah terenkripsi tersebut menjadi beberapa bagian dan didistribusikan kepada pemilik akses. Algoritma *Shamir Secret Sharing Scheme* dipilih karena memiliki cara kerja yang sesuai dan dapat diimplementasikan untuk kasus pengamanan dokumen rahasia digital yang dapat dibuka apabila terdapat sejumlah pihak pemilik hak akses. Selain itu, algoritma tersebut memiliki dasar yang kuat serta menyediakan kerangka yang baik pula dalam pengaplikasiannya. Tingkat keamanan dari algoritma Shamir Secret Sharing Scheme juga terbukti aman [1]. Kombinasi skema Shamir Secret Sharing dengan AES juga telah terbukti memberikan hasil yang maksimal [2].

[1] *Dosen Tetap, Informatika Universitas Kristen Duta Wacana, Jl. Dr. Wahidin Sudirohusodo 5-25 Yogyakarta Indonesia (tlp: 0274-563929; e-mail: willysr@ti.ukdw.ac.id)*

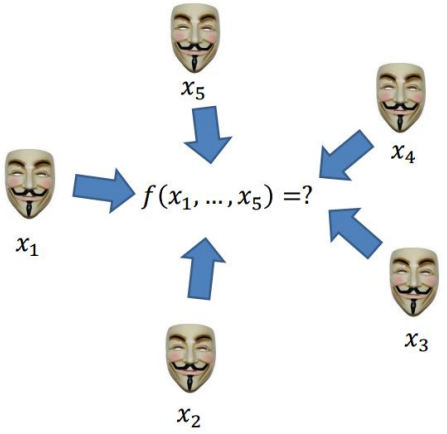
[2] *Dosen Tetap, Informatika Universitas Kristen Duta Wacana, Jl. Dr. Wahidin Sudirohusodo 5-25 Yogyakarta Indonesia (tlp: 0274-563929; e-mail: anton@ti.ukdw.ac.id)*

[3] *Mahasiswa, Informatika Universitas Kristen Duta Wacana, e-mail: pedro.nadirio@ti.ukdw.ac.id*

II. TINJAUAN PUSTAKA

A. *Secure Multi-party Computation*

Secure Multi-party Computation adalah salah satu bagian dari kriptografi yang bertujuan untuk menentukan sebuah cara untuk menggabungkan *input* dari setiap anggota pada suatu grup untuk mendapatkan sebuah informasi, akan tetapi *input* yang diberikan bersifat rahasia dan hanya diketahui oleh pemilik *input* tersebut. Ilustrasi dari *secure multi-party computation* diperlihatkan pada *Gambar 1*.



Gambar 1 Ilustrasi Secure Multi-party Computation

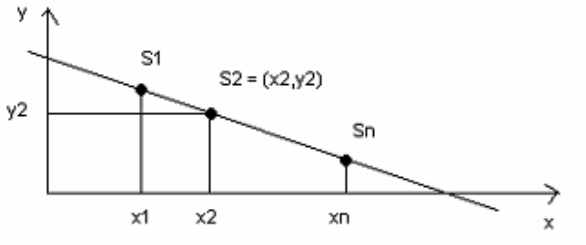
Pada kasus diatas, terdapat lima partisipan $P_1,\ldots,P_5$ yang masing-masing memiliki potongan kunci $X_1,..,X_5$. Kelima partisipan tersebut sudah sepakat dengan sebuah fungsi *f* yang menerima sejumlah *n* input. Tujuan akhirnya adalah untuk menghitung sebuah nilai $y = f(x_1, \ldots, x_n)$ dengan memastikan bahwa dua kondisi selalu terpenuhi, yaitu *correctness* dan *privacy* [3]. Kondisi *correctness* memastikan bahwa nilai *y* merupakan nilai hasil komputasi yang benar, sedangkan *privacy* menjamin bahwa hanya informasi baru yang dipublikasikan dan bukan setiap *input* yang masuk. Input yang diberikan tetap bersifat *private*, sehingga hanya pemilik *input* tersebut yang mengetahuinya.

B. *Shamir Secret Sharing*

Shamir's secret sharing scheme ditemukan oleh Adi Shamir pada tahun 1979 [4]. Pada protokol ini, kunci rahasia akan dibagi sebanyak jumlah partisipan di dalam sebuah kelompok dan dikirimkan ke masing-masing partisipan. Dalam protokol ini terdapat sebuah nilai *threshold* yaitu jumlah *share key* yang dibutuhkan untuk membentuk kembali kunci rahasia dimana nilai *threshold* selalu lebih kecil dari jumlah *share key*.

Skema ini didasarkan atas sebuah fakta yang sangat terkenal dalam ilmu matematika yaitu suatu himpunan *n* buah titik akan mendefinisikan sebuah kurva unik dengan derajat n-1. Skema ini bekerja sebagai berikut: untuk membagi sebuah secret *S* antara *n* pihak sehingga minimal terdapat *m* pihak dari *n* pihak tersebut bisa membentuk kembali secret *S*, maka bentuklah sebuah kurva yang melalui titik (0,S) dengan derajat m-1. Melalui jalur rahasia, setiap pihak akan menerima hasil pembagian dari dari secret *S* berupa sebuah koordinat titik yang dilalui kurva tersebut, dimana masing-masing pihak harus menerima koordinat titik yang berbeda. Untuk nilai m = 2, kurva yang dibentuk seperti Gambar 2.

Shamir Secret Sharing bisa diterapkan pada banyak situasi, seperti halnya pada penerapan watermark pada gambar digital [5], RFID [6], dan E-Voting [7].



Gambar 2 Skema Kurva Polinom Shamir's Scheme dengan Threshold = 2

C. *Lagrange Basis Polynomial*

Lagrange polynomial digunakan untuk mencari interpolasi polinomial. Semisal terdapat satu pasang titik X_j dan Y_j, maka *Lagrange polynomial* didapat dari polinomial dengan sudut terkecil dari tiap point X_j yang diasumsikan sesuai dengan Y_j. Interpolasi polinomial dari sudut terkecil bersifat unik, oleh karena itu lebih tepat dikatakan "the Lagrange form" daripada "the Lagrange interpolation polynomial", karena polinomial yang sama dapat diselesaikan dengan berbagai metode. Meskipun nama Lagrange polynomial diambil dari Joseph Louis Lagrange, yang mempublikasikan pada tahun 1795, tetapi metode ini pertama kali ditemukan pada tahun 1779 oleh Edward Waring [8].

Misalkan terdapat potongan kunci $X_1,..,X_n$, maka nilai f(x) dapat dicari dengan persamaan (1):

$$\mathrm{f(x)} = \sum_{i=1}^{n} f(i) * L_i(X) \qquad (1)$$

dimana L_i merupakan lagrange polynomial yang dapat dihitung menggunakan persamaan (2):

$$\mathrm{L_i(X)} = \frac{\pi_{j \neq i} \left(x - x_j\right)}{\pi_{j \neq i} \left(x_i - x_j\right)} \qquad (2)$$

Sebagai contoh misalkan informasi rahasia s = 1234 dan jumlah partisipan = 6, maka jumlah minimal partisipan yang diperlukan untuk merekonstruksi informasi rahasia s sebanyak 3 peserta [9].

Langkah berikutnya adalah memilih a_1 = 166 dan a_2 = 94, sehingga didapatkan nilai $f(x) = 1234 + 166x + 94x^2$. Lakukan komputasi dan distribusi pada 6 titik, yaitu (1, 1494); (2, 1942); (3, 2578); (4, 3402); (5, 4414); (6, 5614). Untuk bisa merekonstruksi pesan rahasia s kembali, maka cukup diperlukan 3 titik dari 6 titik yang tersedia.

Misalkan
$(x_0, y_0) = (2, 1942)$
$(x_1, y_1) = (4, 3402)$
$(x_2, y_2) = (5, 4414)$

Maka didapatkan

$$\ell_0 = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2}$$

$$= \frac{x - 4}{2 - 4} \cdot \frac{x - 5}{2 - 5}$$

$$= \frac{1}{6} x^2 - 1\frac{1}{2} x + 3\frac{1}{3}$$

$$\ell_1 = \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2}$$

$$= \frac{x - 2}{4 - 2} \cdot \frac{x - 5}{4 - 5}$$

$$= -\frac{1}{2} x^2 + 3\frac{1}{2} x - 5$$

$$\ell_2 = \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1}$$

$$= \frac{x - 2}{5 - 2} \cdot \frac{x - 4}{5 - 4}$$

$$= \frac{1}{3} x^2 - 2x + 2\frac{2}{3}$$

$$f(x) = 1942 \cdot \left(\frac{1}{6} x^2 - 1\frac{1}{2} x + 3\frac{1}{3}\right) + 3402 \cdot \left(-\frac{1}{2} x^2 + 3\frac{1}{2} x - 5\right) + 4414 \cdot \left(\frac{1}{3} x^2 - 2x + 2\frac{2}{3}\right) = 1234 + 166x + 94x^2$$

D. Horner's Method

Dalam dunia matematika, *Horner's method* (atau biasa dikenal dengan *Horner Scheme* atau *Horner rule*) merupakan sebuah cara efisien untuk menghitung nilai dari suku banyak dan turunan dari titik yang diberikan. *Horner's method* dilakukan dengan cara menghilangkan eksponensial untuk menyatakan suku banyak f(x). Dengan menghilangkan eksponensial, dapat mengurangi perhitungan yang diulang sehingga membuatnya menjadi lebih sederhana. Metode ini diberi nama dari seorang matematikawan Inggris bernama William George Horner [10]. Sebagai contoh, diberikan persamaan suku banyak:

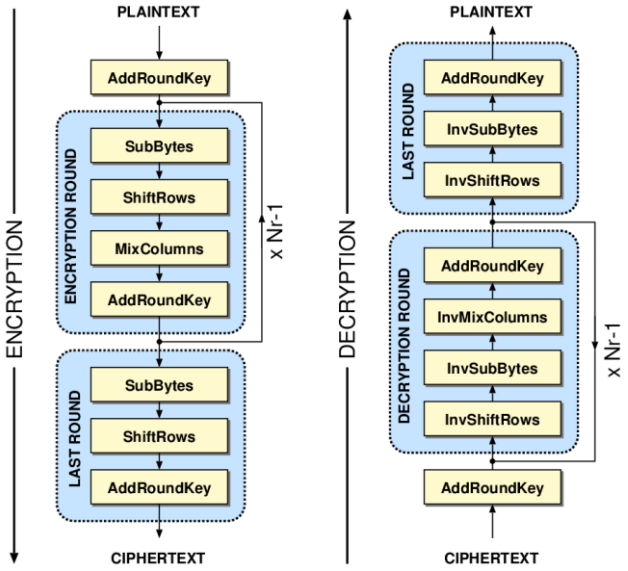
$2x^3 – 6x^2 + 2x – 1$.

Persamaan suku banyak tersebut dapat dihitung dengan cara $((2x – 6)x + 2)x – 1$. Ide yang menjadi dasar adalah menginisialisasi hasil sebagai koefisien dari xn dimana dalam kasus ini bernilai 2, kemudian mengulang menambahkan hasil dengan nilai x dan koefisien berikutnya. Dari persamaan tersebut didapat hasil yang bernilai 5. Algoritma tersebut memiliki kompleksitas O(n).

E. Advanced Encryption Standard

Advanced Encryption Standard (AES) merupakan salah satu model enkripsi menggunakan kunci simetris yang diadopsi oleh pemerintah Amerika Serikat melalui standar FIPS 197 [11]. Standar ini terdiri atas 3 blok cipher, yaitu AES-128, AES-192, dan AES-256. Masing-masing cipher memiliki ukuran blok sebesar 128-bit, dengan ukuran kunci masing-masing 128, 192, dan 256 bit.

AES memiliki empat operasi dalam setiap proses enkripsi maupun dekripsi, yaitu Sub Bytes, Shift Rows, Mix Columns, dan Add Round Key seperti terlihat pada Gambar 3.



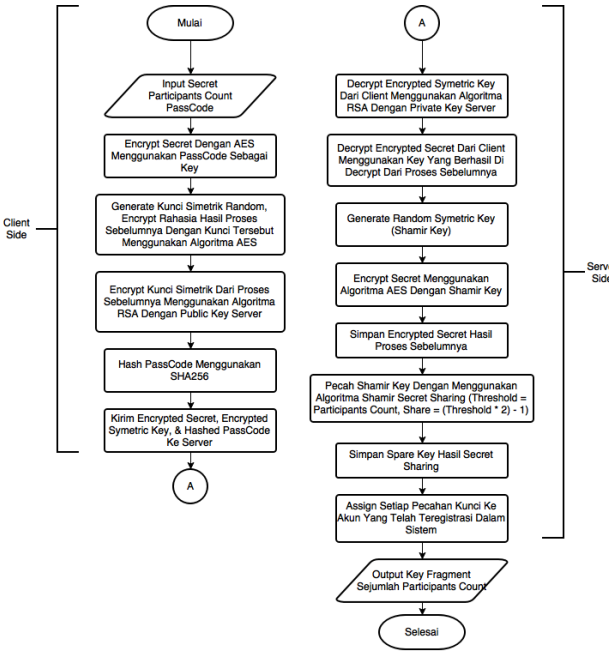
Gambar 3 Skema AES [12]

III. HASIL DAN PEMBAHASAN

Pada sistem pengamanan dokumen rahasia digital, data akan diamankan terlebih dahulu menggunakan algoritma AES dengan kunci yang dihasilkan oleh sistem dengan memanfaatkan pustaka yang dikembangkan oleh Chris Veness [13]. Kunci yang digunakan pada proses pengamanan rahasia tersebut kemudian dipecah menggunakan algoritma *Shamir Secret Sharing*. Alur kerja sistem secara lebih lengkap terlihat seperti pada Gambar 4 untuk proses pengamanan dan Gambar 5 untuk proses pembentukan kembali Dokumen Rahasia Digital. Gambar *6* menggambarkan alur untuk pembuatan rahasia baru, sedangkan Gambar 7 menggambarkan alur untuk pembentukan kembali rahasia.

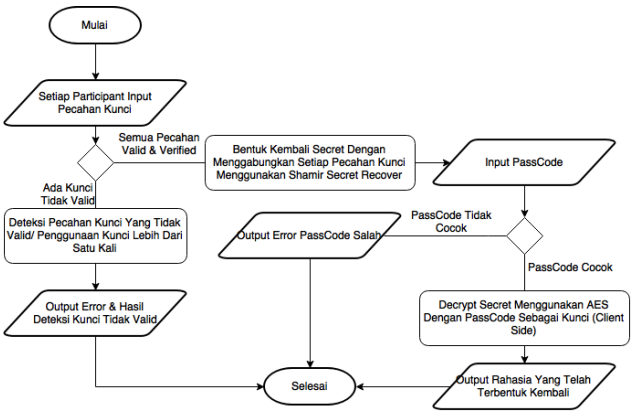
Input PassCode yang terdapat pada alur kerja sistem digunakan untuk menambah tingkat keamanan sebuah

dokumen rahasia. Sebelum dikirim ke server, dokumen rahasia diamankan dengan melakukan enkripsi AES menggunakan PassCode pada sisi *client*. Pihak *server* sekalipun tidak dapat membuka dokumen rahasia yang telah diamankan, tanpa adanya nilai PassCode tersebut. Dokumen rahasia dapat dibuka kembali setelah terbentuk kembali dengan sempurna dan melalui proses dekripsi akhir dengan PassCode yang dimiliki pengguna pada sisi *client*. Tanpa adanya PassCode, rahasia tidak dapat terbuka kembali sehingga membuat segala tindakan curang tanpa adanya PassCode tidak dapat dilakukan.

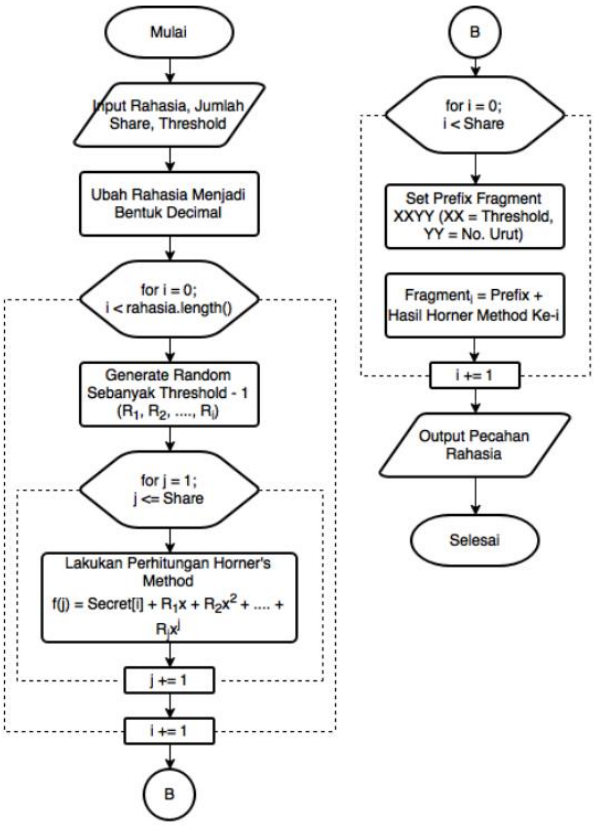


Gambar 4 Flowchart Proses Pengamanan Dokumen Rahasia Digital

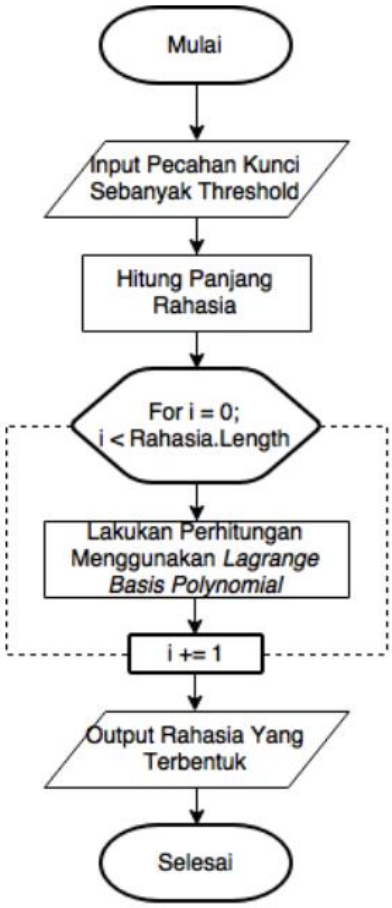
PassCode juga digunakan sebagai data yang diperlukan utnuk menggunakan fitur *Request Key*. Fitur tersebut digunakan apabila terdapat satu atau lebih pecahan kunci valid yang hilang, sehingga sebuah dokumen rahasia tetap dapat terbentuk kembali. Sistem akan menyimpan pecahan kunci cadangan di dalam *server* sejumlah N-1 (N adalah jumlah partisipan). Karena kunci cadangan yang disimpan hanya berjumlah partisipan dikurang satu, maka setidaknya masih dibutuhkan satu buah kunci valid yang dapat digunakan untuk membuka rahasia selain dari kunci cadangan.



Gambar 5 Flowchart Proses Pembentukan Kembali Dokumen Rahasia Digital



Gambar 6 Flowchart Algoritma Pembuatan Rahasia Baru



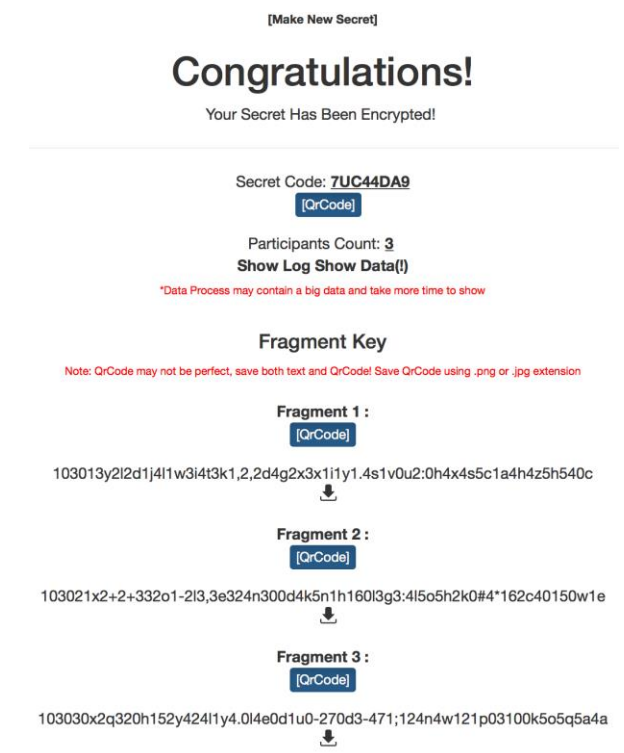
Gambar 7 Flowchart Algoritma Pembentukan Kembali Rahasia

Sistem pengamanan dokumen rahasia digital dapat terlihat pada Gambar 8. Pada tampilan ini, pengguna dapat membuat dan mengamankan sebuah dokumen rahasia baru. Terdapat beberapa input yang harus dimasukan antara lain jenis rahasia, jumlah partisipan, PassCode, serta dokumen rahasia yang akan diamankan. Apabila proses pengamanan berhasil maka akan terlihat seperti pada Gambar 9, dan sistem akan memberi pecahan-pecahan kunci sejumlah partisipan yang digunakan untuk membuka kembali dokumen rahasia.

Gambar 8 Halaman Pembuatan Dokumen Rahasia Digital Baru

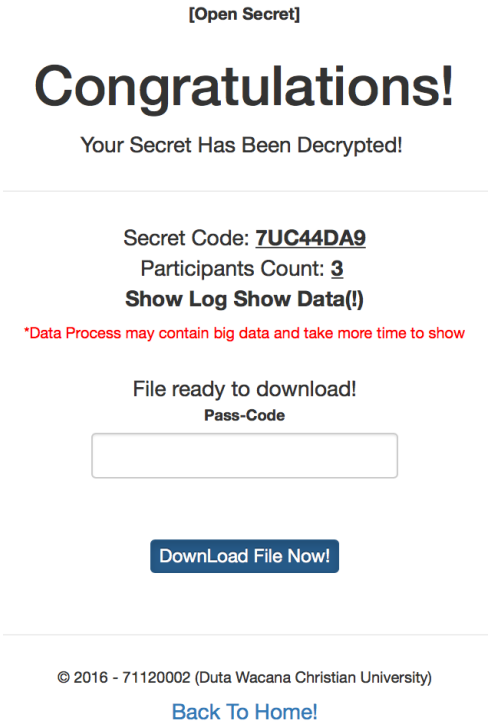
Untuk membuka kembali sebuah dokumen rahasia, diperlukan *input* setiap pecahan (fragmen) kunci valid seperti terlihat pada Gambar 10. Untuk mempermudah pendistribusian informasi pecahan kunci dan pembacaan kembali saat proses restorasi dokumen rahasia digunakan QR Code dengan menggunakan pustaka JavaScript QRCode reader [14]. Apabila proses pembentukan kembali rahasia berhasil, maka akan muncul tampilan seperti pada Gambar 11. Untuk dapat membuka kembali rahasia, diperlukan input PassCode sebagai kunci dari proses dekripsi akhir di sisi *client*. Apabila terdapat *input* kunci tidak valid atau PassCode tidak valid, maka sistem akan memberi pesan kesalahan dan mengidentifikasi tindakan curang tersebut sehingga dokumen rahasia tidak dapat terbentuk kembali. Apabila terdapat potongan yang tidak valid atau sistem gagal melakukan validasi terhadap setiap potongan informasi maupun terhadap kunci yang dipalsukan atau yang digunakan lebih dari satu kali, maka sistem juga akan menampilkan pesan kesalahan (Gambar 12, Gambar 13, Gambar 14, dan Gambar 15).

Sistem yang dibangun dapat diakses melalui alamat URL: https://vhost.ti.ukdw.ac.id/~pedro/. Beberapa pustaka lain yang digunakan pada penelitian ini adalah CryptoJS for Key Derivation [15], File Saver [16], RSA JsEncrypt [17], dan SHA256 Hash [18].

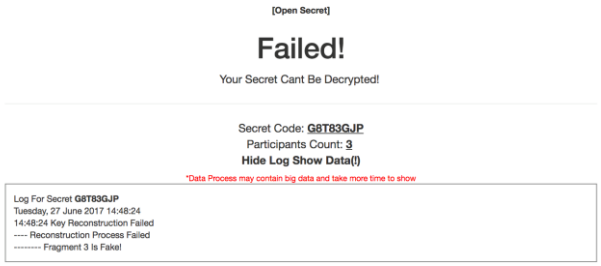


Gambar 9 Proses Pemecahan Data Rahasia

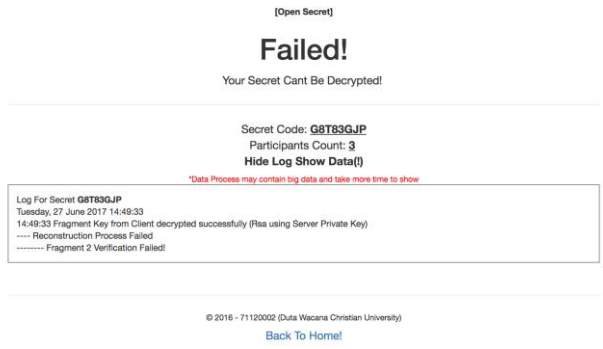
Gambar 10 Halaman Penggabungan Fragmen



Gambar 11 Proses Penggabungan Berhasil



Gambar 12 Salah satu potongan rahasia tidak valid



Gambar 13 Percobaan Penggabungan Gagal

Gambar 14 Mendeteksi Kunci Palsu

Gambar 15 Mendeteksi Penggunaan Kunci Berulang Kali

Untuk menguji keamanan dari sistem yang dibangun, maka akan dilakukan proses pengujian sistem. Pengujian yang akan dilakukan meliputi hal-hal yang dapat terjadi pada saat pembentukan kembali dokumen rahasia. Hal-hal tersebut antara lain:

- Apakah sistem dapat membentuk kembali rahasia apabila setiap pecahan kunci merupakan kunci yang valid.
- Apakah sistem dapat mencegah tindak kecurangan yang dilakukan dengan cara mengubah pecahan kunci sehingga menjadi tidak valid atau penggunaan pecahan kunci yang sama sebanyak lebih dari satu kali.
- Apakah sistem dapat mendeteksi pihak yang melakukan kecurangan dengan cara menentukan pecahan kunci yang tidak valid bila terjadi tindak kecurangan.
- Apakah sistem dapat mencegah kecurangan yang dilakukan dengan cara membuka kembali rahasia tanpa menggunakan PassCode yang valid.

Tabel 1 Hasil Analisis Sistem

Input Kunci	**Expected Output**	**Output**
Setiap Kunci Valid	Rahasia Terbentuk Kemballi	Rahasia Dapat Terbentuk
Kunci Valid Yang Dipotong Pada Beberapa Bagian	Kunci Menjadi Tidak Valid	Sistem Mendeteksi Sebagai Kunci Tidak Valid
Kunci Valid Yang Ditambah Nilainya Dengan Beberapa Karakter	Kunci Menjadi Tidak Valid	Sistem Mendeteksi Sebagai Kunci Tidak Valid
Kunci Valid Yang Diubah Nilainya Satu Karakter Atau Lebih	Kunci Menjadi Tidak Valid	Sistem Mendeteksi Sebagai Kunci Tidak Valid
Input Kunci Acak	Kunci Tidak Valid	Sistem Mendeteksi Sebagai Kunci Tidak Valid
Input Kunci Cadangan Yang Didapat Dari Server	Kunci Valid	Sistem Mendeteksi Kunci Sebagai Kunci Valid
Input Kunci Valid Namun Berasal Dari Secret Lain	Kunci Tidak Valid	Sistem Mendeteksi Sebagai Kunci Tidak Valid Karena Tidak Cocok
Input Kunci Valid Dengan Password Yang Tidak Cocok	Verifikasi Gagal	Sistem Mendeteksi Verifikasi Input Kunci Dengan *Password* Yang Tidak Sesuai

Pengujian dilakukan dengan cara menjalankan sebuah skenario yang dapat mencakup semua pengujian di atas terhadap 20 dokumen yang berbeda. Skenario yang akan dijalankan dimulai dengan proses pembuatan dokumen rahasia baru yang dilanjutkan dengan proses percobaan pembentukan kembali dokumen rahasia dengan berbagai kemungkinan yang dapat mencakup seluruh tujuan pengujian.

Dari hasil pengujian yang dilakukan, sistem telah berhasil mengamankan dokumen rahasia serta mendeteksi kunci mana yang merupakan kunci tidak valid. Hasil pengujian yang dilakukan dapat terlihat pada Tabel 1. Dari hasil yang didapat dari proses pengujian sistem, diketahui bahwa sistem dapat menangani tindak kecurangan yang dilakukan pada setiap skenario. Sistem juga dapat melakukan proses pendeteksian kunci yang tidak valid. Pendeteksian kunci yang tidak valid tersebut dilakukan dengan cara menyimpan pecahan kunci valid cadangan di dalam *server*, dan melakukan pengujian terhadap input

pecahan kunci dari pengguna menggunakan pecahan kunci cadangan. Pecahan kunci cadangan yang disimpan di dalam *server* berjumlah partisipan dikurang satu sehingga *server* sekalipun tidak dapat membuka rahasia hanya dengan kunci cadangan tersebut demi menjaga keamanan dokumen rahasia. Setiap pecahan kunci yang disimpan di dalam *server* merupakan pecahan yang berbeda dengan yang diberikan kepada pengguna.

Selain pendeteksian kunci tidak valid, terdapat hal yang tidak dapat ditangani algoritma *Shamir Secret Sharing Scheme*, yaitu pembentukan sebuah dokumen rahasia dengan kunci yang tidak valid akan tetapi memiliki panjang kunci yang sama dengan kunci valid tetap dapat dilakukan walau tidak menghasilkan rahasia yang sempurna. Hal ini disebut *bottom* yang sering disimbolkan dengan $\perp$. *Bottom* digunakan untuk menjamin *ciphertext integrity* pada sistem *authenticated encryption* [19] [20]. Untuk dapat membedakan dengan hasil yang bersifat *random* dari hasil percobaan kunci yang tidak valid, maka kunci disimpan dalam bentuk *hashed key*. Ketika sebuah dokumen rahasia terbentuk kembali, maka informasi rahasia yang terbentuk tersebut dibandingkan dengan *hashed key* yang disimpan oleh *server*. Penyimpanan *hashed key* ini tidak membuat *server* dapat membuka kembali informasi kunci. Alternatif lain yang dapat digunakan untuk menyelesaikan persoalan ini adalah dengan menggunakan konsep fair reconstruction [21] [22] [23] [24], dan [25].

IV. KESIMPULAN

Berdasarkan hasil implementasi sistem dan analisis yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Dokumen rahasia yang dimasukkan pengguna berhasil diamankan dengan metode *Shamir Secret Sharing Scheme*. Dokumen rahasia hanya dapat terbentuk kembali apabila terdapat setiap pecahan kunci yang valid. Terbukti bahwa dengan adanya sebuah kunci saja yang bernilai tidak valid, dokumen rahasia tidak dapat terbentuk kembali.
2. Sistem telah berhasil mendeteksi tindak kecurangan pemalsuan pecahan kunci untuk membuka sebuah dokumen rahasia. Sistem menggunakan pecahan kunci cadangan untuk melakukan proses deteksi kecurangan tersebut.
3. Pemalsuan pecahan kunci dengan cara mengubah sebagian nilai dari kunci tanpa mengubah ukuran kunci akan tetap menghasilkan sebuah dokumen rahasia walau tidak sempurna atau tidak bermakna apapun. Hal ini tidak dapat diatasi oleh algoritma *Shamir Secret Sharing Scheme* karena rahasia tetap dapat terbentuk walau tidak sesuai dengan dokumen rahasia awal dan tidak memberi dampak kesalahan. Untuk mengatasi hal ini, sistem menyimpan *hashed key* untuk dibandingkan dengan rahasia yang berhasil dibentuk. Terbukti sistem dapat mendeteksi apabila terdapat indikasi pemalsuan kunci.
4. Untuk lebih meningkatkan keamanan sebuah rahasia, maka pada alur proses pengamanan diperlukan sebuah PassCode. Dengan adanya pengamanan tambahan dengan PassCode ini, maka hanya pihak yang memiliki PassCode yang dapat membuka dokumen rahasia. Sistem sekalipun tidak dapat mencoba membuka dokumen rahasia tanpa adanya PassCode yang sesuai.
5. Apabila terdapat pecahan kunci valid yang hilang maka rahasia tidak dapat lagi terbentuk. Hal ini telah dapat diatasi dengan cara menyediakan fitur *Request Key* yang digunakan untuk meminta kunci cadangan yang disimpan sistem. Terbukti kunci cadangan yang disimpan, dapat digunakan untuk pembentukan kembali rahasia.

Untuk pengembangan sistem lebih lanjut, dapat dilakukan dengan cara mengembangkan sistem pengamanan dokumen rahasia digital pada *platform* lain seperti pada *smartphone Android* atau *iOS*. Hal tersebut dapat dilakukan karena sistem dokumen rahasia yang diperuntukkan kepada sejumlah pihak cocok diimplementasikan pada *mobile device* karena memberi kemudahan akses lebih baik bagi penggunanya. Terdapat hal yang perlu diperhatikan untuk pengembangan sistem pada *mobile device*, yaitu bagaimana cara mengintegrasikan setiap proses yang terdapat pada sistem seperti proses pengumpulan kembali pecahan kunci sehingga nyaman dan mudah digunakan oleh penggunanya.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Fakultas Teknologi Informasi UKDW yang telah mendukung penelitian ini dengan menyediakan infrastruktur untuk melakukan pengujian secara online.

DAFTAR PUSTAKA

[1] D. Bogdanov, "Foundations and properties of Shamir's secret sharing scheme" dalam Research Seminar in Cryptography, Mei 2007.
[2] L. Goubin & A. Martinelli, "Protecting AES with Shamir's Secret Sharing Scheme", Workshop on Cryptographic Hardware and Embedded Systems, Nara, Jepang September, 2011.
[3] R. Cramer & I. Bjerre Damgard, "Secure Multiparty Computation" dalam "Secure Multiparty Computation and Secret Sharing", edisi ke-1, Inggris, 2015, bab 1, halaman 6.
[4] A. Shamir, "How to share a secret", Communications ACM, Vol 22, Issue 11, Halaman 612-613, November 1979.
[5] G. Gnaneshwar, "Color Image Integrity Verification Using Shamir's Secret Sharing Scheme", International Journal of Innovative Research & Development, halaman 20-24, Juni, 2014.
[6] A. Al-Adhami, M. Ambroze, & I. Stenget, "A 256 bit implementation of ECC-RFID based system using Shamir Secret sharing scheme and Keccak hash function", Ubiquitous and Future Networks (ICUFN), Ninth International Conference, 4-7 Juli 2017.
[7] B. V.P, D. G. Nair, A. Sreekumar, "Secret Sharing Homomorphism and Secure E-voting", Februari 2016.
[8] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing. *Proceedings of the IEEE",* halaman 319-342. Agustus 2002.
[9] V. Relan, "Secret Sharing", September, 2009.
[10] W. G. Horner, "A New Method of Solving Numerical Equations of All Orders, by Continuous Approximation", Philosophical

Transactions of the Royal Society of London, Vol. 109, halaman 308-335, 1819.
[11] Specification for the ADVANCED ENCRYPTION STANDARD (AES), National Institute of Standards and Technology, Springfield, VA, 2001.
[12] F. Kagan Gürkaynak, "GALS System Design: Side Channel Attack Secure Cryptographic Accelerators", disertasi Ph.D, Dept. of Information Technology and Electrical Engineering, ETH Zurich, Zürich, Switzerland, 2006.
[13] C. Veness, "AES Implementation in PHP" [online], tersedia: http://www.movable-type.co.uk/scripts/aes-php.html.
[14] L. Laszlo, "Javascript QRCode scanner" [online], tersedia: https://github.com/LazarSoft/jsqrcode
[15] J. Mott, "crypto-js" [online], tersedia: https://code.google.com/archive/p/crypto-js/
[16] E. Grey, "FileSaver.js" [online], tersedia: https://github.com/eligrey/FileSaver.js/
[17] T. Tidwell, "jsencrypt" [online], tersedia: https://github.com/travist/jsencrypt
[18] A. Martin & P. Johnston, "Javascript SHA-256" [online], tersedia: http://www.webtoolkit.info/javascript_sha256.html#.WqT14OeYPDd
[19] D. Boneh, "Authenticated encryption" [online], tersedia: https://crypto.stanford.edu/~dabo/courses/OnlineCrypto/slides/07-authenc-v2-annotated.pdf
[20] P. Rogaway, "The Evolution of Authenticated Encryption", Workshop on Real-World Cryptography, California, Amerika Serikat, Januari, 2013.
[21] M. Tompa & H. Woll, "How to share a secret with cheaters", Journal of Cryptology, halaman 133-138, 1988.
[22] C.-S. Laih & Y-C. Lee, "V-fairness (t,n) secret sharing scheme", IEEE Proceedings – Computer and Digital Technique Juli 1997.
[23] W. K. Moses & C. Pandu Rangan, "Rational Secret Sharing over an Asynchronous Broadcast Channel with Information Theoretic Security", International Journal of Network Security & Its Application, November 2011.
[24] Y. Tian, J. Ma, C. Peng, & J. Zhu, "Secret Sharing Scheme with Fairness", Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference, November 2011.
[25] L. Harn, C. Lin, Y. Li, "Fair Secret reconstruction in (t,n) secret sharing", Journal of Information Security and Applications, Juli 2015.