

# Aplikasi Keamanan Berkas dengan Enkripsi AES dan Biometrik Sidik Jari Berbasis Android

## *Android File Security Application with AES Encryption and Fingerprint Authentication*

Habib Husain Amirullah<sup>1\*</sup>  
Ade Eviyanti<sup>2</sup>  
Sumarno<sup>3</sup>

<sup>1,2,3</sup>Informatika, Fakultas Sains dan Teknologi, Universitas Muhammadiyah Sidoarjo, Jl. Mojopahit No.666 B, Sidowayah, Celep, Kec. Sidoarjo, Kabupaten Sidoarjo, Jawa Timur, 61215, Indonesia  
<sup>1</sup>201080200028@umsida.ac.id, <sup>2</sup>adeeviyanti@umsida.ac.id, <sup>3</sup>sumarno@umsida.ac.id

**\*Penulis Korespondensi:**  
Habib Husain Amirullah  
201080200028@umsida.ac.id

### **Abstrak**

*Keamanan data pada era digital menjadi hal yang sangat penting. Pencurian data dan dokumen digital terus terjadi, dengan biaya rata-rata akibat data breach mencapai \$3.86 juta pada tahun 2018. Untuk menghadapi tantangan ini, kriptografi, terutama enkripsi, menjadi elemen kunci dalam menjaga kerahasiaan dan integritas data. Advanced Encryption Standard (AES) telah menjadi standar utama di seluruh dunia untuk menjaga kerahasiaan data dengan mengubah data menjadi bentuk yang sangat sulit dipecahkan tanpa kunci yang sesuai. Namun, keamanan AES masih sangat tergantung pada kekuatan kunci yang digunakan, menghadirkan risiko kunci yang lemah dan kemungkinan kelalaian. Penelitian ini berusaha mengatasi permasalahan tersebut dengan menggabungkan AES-128 sebagai algoritma enkripsi utama dan autentikasi berbasis sidik jari untuk meningkatkan keamanan akses. Penggunaan verifikasi biometrik sidik jari memberikan lapisan keamanan tambahan yang lebih canggih dan mudah digunakan. Aplikasi diuji menggunakan Automated Testing, yang merupakan metode untuk melakukan pengujian pada sebuah sistem dengan serangkaian skrip. Hasil pengujian menunjukkan bahwa dengan menggabungkan teknologi enkripsi terkini dan metode autentikasi biometrik, penelitian ini berhasil mengembangkan aplikasi yang mampu melakukan enkripsi menggunakan algoritma AES dan mengintegrasikannya dengan BiometricPrompt. Hasilnya adalah peningkatan keamanan data pada era digital.*

**Kata Kunci:** AES; android; biometric; kriptografi

### **Abstract**

*Data security in the current digital era has become critically important. Data and digital document thefts continue to occur, with an average cost of \$3.86 million due to data breaches in 2018. To tackle this challenge, cryptography, particularly encryption, has become a key element in maintaining data confidentiality and integrity. The Advanced Encryption Standard (AES) has become the global standard for preserving data confidentiality by transforming data into a form that is difficult to decipher without the correct key. However, AES security relies heavily on the strength of the key used, posing risks of weak keys and potential negligence. This research aims to address these issues by combining AES-128 as the encryption algorithm and fingerprint-based authentication to enhance security access. The use of fingerprint biometric verification provides a user-friendly layer of security. The application was tested using Automated Testing, which is a method for testing a system using a series of scripts. The test results demonstrate that by combining the latest encryption technology and biometric authentication, this research successfully developed an application capable of encrypting data using the AES algorithm and integrating it with BiometricPrompt. The outcome is an improved level of data security in this digital era.*

**Keywords:** AES; android; biometrics; cryptography

## 1. Pendahuluan

Pada era digital saat ini, keamanan data menjadi sebuah prioritas utama. Namun, hingga sekarang masih sering terjadi kasus pencurian data dan dokumen dalam bentuk digital. Tercatat rata-rata biaya atau kerugian dari *data breach* dalam laporan tahun 2018, mencapai \$3.86 juta [1]. Tantangan yang harus dihadapi sekarang adalah bagaimana cara untuk mengamankan data dan mencegah akses yang tidak sah pada dokumen digital.

Kriptografi adalah suatu metode pengkodean yang digunakan untuk menjaga kerahasiaan dan integritas dokumen, serta meningkatkan tingkat keamanan agar hanya pihak yang berwenang yang dapat mengaksesnya [2]. Enkripsi sendiri salah satu metode kriptografi yang menjadi jawaban atas pertanyaan, “*How can we prevent observers from understanding our conversations?*” Dengan menerapkan enkripsi pada pesan ataupun dokumen, informasi yang disimpan kemudian diubah menjadi bentuk yang tidak dapat dipahami oleh orang-orang yang mencoba untuk memantau atau mengakses data tersebut [3], [4], [5].

AES (*Advanced Encryption Standard*) adalah standar enkripsi terkini yang didesain dan dikembangkan oleh Joan Daemen dan Vincent Rijmen bersama NIST (*National Institute of Standards and Technology*) sebagai standar enkripsi yang kemudian diadopsi oleh pemerintah Amerika Serikat [6]. AES telah menjadi pilihan utama dalam mengamankan informasi rahasia di seluruh dunia karena memiliki tingkat keamanan untuk melakukan pertukaran informasi yang cukup baik [7]. Secara umum cara kerja algoritma AES adalah mengubah *plaintext* menjadi *ciphertext* yang merupakan bentuk yang sangat sulit dipecahkan dan tidak dapat dimengerti oleh pihak yang tidak berwenang [8]. *Ciphertext* adalah hasil dari proses enkripsi dan hanya dapat dibaca kembali menjadi *plaintext* dengan menggunakan kunci dekripsi yang tepat [2].

Pada penelitian yang dilakukan Setyaningsih (2020), dibuat sebuah aplikasi berbasis Android untuk melakukan enkripsi file [9]. File dienkripsi menggunakan algoritma AES dengan tambahan opsi untuk memilih jumlah bit dari kunci, yang membuat 3 macam enkripsi AES yakni AES-128, AES-192, dan AES-256. Hal tersebut memberikan fleksibilitas kepada pengguna untuk memilih opsi keamanan atau kecepatan proses enkripsi. Semakin besar jumlah bit kunci yang digunakan, semakin tinggi tingkat keamanan yang dapat dihasilkan [9]. Sementara itu pada penelitian lainnya, dikembangkan juga aplikasi enkripsi-dekripsi dokumen menggunakan algoritma AES pada perangkat Desktop [7].

Penggunaan algoritma enkripsi AES sebagai lapisan keamanan utama telah menjadi standar yang lazim digunakan di dunia industri, namun penggunaan AES masih tergantung pada keamanan kunci yang digunakan [10]. Hal tersebut bisa memunculkan masalah keamanan baru, seperti risiko kunci yang lemah dan mudah ditebak, seperti saat menggunakan teknik *brute-force*, serta kemungkinan kelalaian yang menyebabkan pengguna lupa kunci yang digunakan untuk melakukan enkripsi [11]. Oleh karena itu, kehadiran metode autentikasi yang lebih canggih, aman dan mudah digunakan menjadi penting dalam mengamankan akses ke data yang dienkripsi [12].

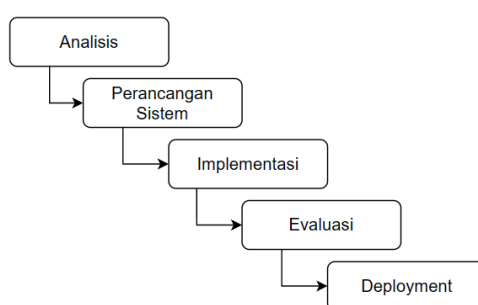
Salah satu solusi yang muncul adalah penggunaan verifikasi biometrik, dengan sidik jari sebagai bentuk autentikasi pengguna. Dengan mendaftarkan sidik jari sebagai kunci akses, pengguna dapat membuka enkripsi data dengan mudah dan cepat, tanpa perlu mengingat atau memasukkan kunci yang rentan terhadap risiko keamanan. Selain itu, sidik jari bersifat unik untuk setiap orang karena tidak ada dua orang yang ditemukan memiliki sidik jari yang sama, dan polanya tidak pernah berubah seumur hidup seseorang [13].

Dengan memahami risiko keamanan yang ada dan mengimplementasikan teknologi terbaru, penelitian dengan judul “Aplikasi Keamanan Berkas dengan Enkripsi AES dan Biometrik Sidik Jari Berbasis Android” diharapkan dapat memberikan kontribusi positif dalam meningkatkan keselamatan dan keamanan data di era digital saat ini. Aplikasi dikembangkan menggunakan algoritma enkripsi AES128 yang telah dikenal luas karena keefektifannya dalam mengamankan

data [2]. Verifikasi pengguna menggunakan sidik jari dilakukan pada saat aplikasi dimulai atau dijalankan oleh pengguna. Langkah tersebut bertujuan untuk memberikan lapisan keamanan tambahan yang mencegah akses yang tidak sah ke dalam aplikasi itu sendiri, menjaga bahwa hanya pemilik sah yang dapat membuka dan mengakses data yang dienkripsi [13].

## 2. Metode Penelitian

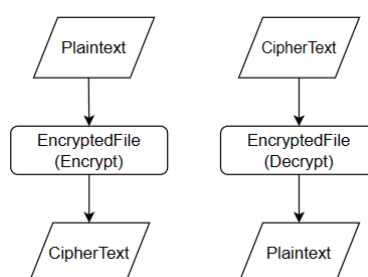
Metode yang digunakan dalam penelitian untuk pengembangan Aplikasi Keamanan Berkas dengan Enkripsi AES dan Biometrik Sidik Jari adalah metode Waterfall. Metode Waterfall adalah model pengembangan perangkat lunak yang paling tua dan sering digunakan dalam beberapa proyek pemerintahan karena memberikan gambaran yang lebih baik pada tahap awal pengembangan [14]. Model Waterfall yang digunakan diilustrasikan pada Gambar 1 di bawah:



**Gambar 1.** Metode Waterfall

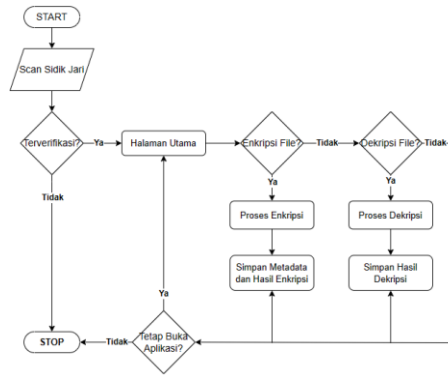
Studi literatur dimulai dengan melakukan pencarian artikel jurnal terkait dengan pengembangan Aplikasi Keamanan Berkas dengan Enkripsi AES dan Biometrik Sidik Jari melalui Google Scholar. Tahap berikutnya melibatkan riset dan identifikasi cakupan area penelitian yang dieksplorasi.

Pada dasarnya, semua jenis ber-kas, seperti gambar atau dokumen, memiliki bentuk yang disebut *plaintext* yang berisi informasi dan isi terkait berkas tersebut. *Plaintext* inilah yang kemudian dienkripsi menjadi *ciphertext* yang merupakan bentuk yang tidak dapat dimengerti tanpa kunci enkripsi yang sesuai. Proses enkripsi dan dekripsi yang dilakukan diilustrasikan pada Gambar 2 di bawah:



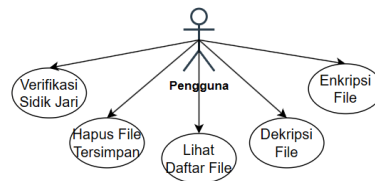
**Gambar 2.** Diagram Sistem Enkripsi dan Dekripsi

Aplikasi Keamanan Berkas dengan Enkripsi AES dan Biometrik Sidik Jari dirancang untuk memiliki *flow* seperti yang diilustrasikan pada Gambar 3 di bawah:



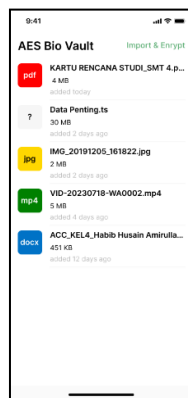
Gambar 3. Flowchart Aplikasi

Selain *flowchart*, *Use Case Diagram* juga digunakan dalam pengembangan aplikasi untuk menggambarkan interaksi antara pengguna dan sistem. *Use Case Diagram* yang digunakan diilustrasikan pada Gambar 4 di bawah:



Gambar 4. Diagram Use Case

Selain itu, komponen penting lain yang perlu dipertimbangkan dalam tahap perancangan sistem adalah desain antarmuka pengguna, atau yang dikenal dengan istilah *user interface* (UI). Antarmuka pengguna merupakan mekanisme komunikasi antara pengguna dan sistem dalam suatu program, termasuk aplikasi web, perangkat seluler, dan perangkat lunak [15]. Antar muka aplikasi yang digunakan diilustrasikan pada Gambar 5 di bawah:



Gambar 5. Antar Muka Aplikasi

Tahap berikutnya adalah implementasi, di mana perancangan konsep diwujudkan menjadi bentuk nyata melalui pengembangan aplikasi menggunakan Android Studio 2022.3.1 (Giraffe). Android Studio adalah *Integrated Development Environment* (IDE) yang kuat, kaya fitur, dan mudah digunakan secara intuitif [16]. Pada tahap implementasi ini, komponen-komponen yang telah dirancang sebelumnya diubah menjadi kode-kode program yang dapat dijalankan oleh perangkat Android. Aplikasi dirancang dengan memanfaatkan teknologi terbaru dalam pengembangan Android, yaitu Jetpack Compose. Dengan Jetpack Compose, pengembang dapat

merancang antarmuka pengguna dengan lebih intuitif dan efisien, menggantikan paradigma tradisional XML yang biasa digunakan.

Dalam penelitian ini, Android telah menyediakan kelas-kelas yang dibutuhkan untuk melakukan prosedur enkripsi dan dekripsi file serta verifikasi sidik jari. Kelas `BiometricPrompt` digunakan untuk menampilkan pesan yang mengarahkan pengguna untuk memindai sidik jari yang telah terdaftar pada sensor sidik jari. Di sisi lain, kelas `EncryptedFile` digunakan untuk proses enkripsi dan dekripsi file. Dokumentasi kedua kelas tersebut dapat ditemukan pada laman resmi Android Developers.

Kelas yang digunakan untuk melakukan verifikasi sidik jari dikenal sebagai `BiometricPrompt` dan dapat ditemukan dalam paket `android.hardware.biometrics`. Sayangnya, kelas `BiometricPrompt` baru diperkenalkan pada API Level 28. Untuk mengatasi keterbatasan ini, digunakan versi Jetpack dari kelas ini, yaitu yang terdapat dalam paket `androidx.biometric`, yang memungkinkan pengembang untuk mendukung verifikasi sidik jari pada berbagai perangkat dengan API Level yang lebih rendah dan memberikan pengalaman yang lebih konsisten bagi pengguna.

Sedangkan kelas `EncryptedFile`, yang juga merupakan bagian dari Android Jetpack, terdapat dalam paket `androidx.security.crypto`. Namun sayangnya, kelas `EncryptedFile` hanya mendukung algoritma AES-256, sehingga diperlukan beberapa modifikasi pada *source code* yang dapat diakses melalui GitHub. Modifikasi yang dilakukan adalah mengubah panjang kunci bawaan yang di-generate, yang sebelumnya 256-bit menjadi 128-bit.

Untuk menyimpan metadata dari dokumen yang telah dienkripsi, digunakan Jetpack Room yang merupakan sebuah library yang terintegrasi dengan Android Jetpack dan didesain khusus untuk memudahkan pengelolaan basis data SQLite dalam aplikasi. Room menyediakan antarmuka atau abstraksi tingkat tinggi untuk *database* SQLite [17]. Room bekerja dengan prinsip-prinsip DAO (*Data Access Object*) yang memungkinkan untuk membuat metode abstrak untuk operasi dasar seperti *insert*, *update*, *delete*, dan *query*, tanpa harus menulis SQL *query* secara langsung.

Dalam tahap evaluasi ini, kinerja dan keamanan Aplikasi Keamanan Berkas berbasis Android dengan Enkripsi AES dan Verifikasi Sidik Jari diperiksa secara menyeluruh. Evaluasi mencakup beberapa aspek penting untuk memastikan bahwa aplikasi memenuhi tujuannya dan memenuhi harapan pengguna. Salah satunya adalah pengujian fungsionalitas, yang mencakup skenario penggunaan berbeda untuk memverifikasi fungsi aplikasi. Pengujian fungsionalitas berfokus pada pengujian berbagai fitur dan integrasi tampilan dalam aplikasi.

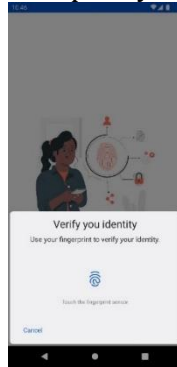
Evaluasi mencakup pengujian kinerja untuk memastikan responsivitas aplikasi dan kesesuaian terhadap *flow* yang telah ditetapkan. Evaluasi kinerja dilakukan menggunakan *Automated Testing*, yang terdiri dari dua jenis tes: *Unit Test*, yang memastikan fungsionalitas dasar seperti menampilkan berkas tersimpan berjalan dengan benar, serta *Instrumentation Test* (UI Test), yang menguji interaksi antara aplikasi dan antarmuka pengguna (UI) serta integrasi komponen aplikasi dengan sistem Android [18].

Tahap terakhir, yaitu tahap deployment yang dilakukan setelah aplikasi berhasil melewati tahap evaluasi dengan hasil yang memuaskan. Pada tahap ini, Aplikasi Keamanan Berkas dengan Enkripsi AES dan Biometrik Sidik Jari Berbasis Android disiapkan untuk dirilis dan digunakan oleh pengguna. Aplikasi yang telah siap dapat diunggah ke platform resmi PlayStore, namun juga tersedia opsi lain untuk publikasi melalui Github Release, terutama apabila dana atau biaya rilis di PlayStore menjadi perhatian utama.

### 3. Hasil

Setelah menyelesaikan beberapa tahapan sebelumnya, langkah berikutnya adalah menganalisis hasil program aplikasi sesuai dengan desain yang telah disusun. Pada Halaman Login, pengguna

diminta untuk memindai sidik jari menggunakan sensor yang tersedia pada perangkat Android. Tampilan pada Halaman Login dapat dilihat seperti yang ditunjukkan pada Gambar 6 di bawah:



**Gambar 6.** Halaman Login

Apabila pengguna tidak terverifikasi melalui pemindaian sidik jari, muncul pesan pemberitahuan bahwa verifikasi sidik jari tidak berhasil dan pengguna tidak diizinkan untuk mengakses aplikasi seperti yang diilustrasikan pada Gambar 7



**Gambar 7.** Pesan Gagal Memverifikasi Sidik Jari

Pada halaman ini, pengguna yang berhasil terverifikasi melalui pemindaian sidik jari kemudian diarahkan ke halaman utama aplikasi yang menampilkan daftar berkas yang telah dienkripsi. Tampilan pada Halaman Utama dapat dilihat dalam Gambar 8 di bawah:

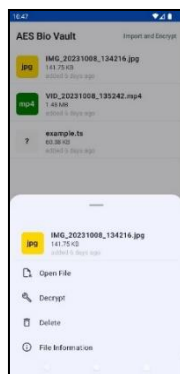


**Gambar 8.** Halaman Utama

Pada bagian kanan atas halaman utama, terdapat tombol untuk menambahkan berkas kedalam aplikasi untuk dilakukan proses enkripsi. Sebelum proses enkripsi dilakukan, berkas di-filter berdasarkan kapasitasnya, hal tersebut dilakukan untuk mencegah *OutOfMemoryError*

dikarenakan kapasitas berkas yang terlalu besar. Apabila berkas memiliki kapasitas kurang dari 128 MB, maka proses enkripsi dijalankan secara otomatis tanpa harus melakukan input kunci enkripsi.

Selain itu, item yang terdapat pada daftar berkas juga memiliki aksi untuk membuka Bottom Sheet yang dijalankan apabila salah satu item tersebut ditekan. Tampilan dari Bottom Sheet ditunjukkan pada Gambar 9 di bawah:



Gambar 9. Tampilan Bottom Sheet

Beberapa aksi yang dapat dilakukan pada sebuah berkas yang telah tersimpan dan terenkripsi di dalam aplikasi. *Open File* adalah membuka berkas yang telah di dekripsi dengan aplikasi bawaan yang sesuai. Contohnya adalah berkas gambar yang dibuka menggunakan galeri. Apabila tidak ada aplikasi yang sesuai untuk membuka suatu berkas, maka ditampilkan pilihan untuk mencari aplikasi di PlayStore. *Decrypt* untuk melakukan proses dekripsi. *Delete* untuk menghapus berkas dari penyimpanan internal aplikasi. *File Information* untuk menampilkan informasi yang lebih detail terkait berkas yang dipilih, termasuk lokasi berkas yang telah didekripsi.

Setelah sebuah berkas berhasil didekripsi, aplikasi kemudian menunggu 10 menit sebelum menghapus berkas yang telah didekripsi secara otomatis, hal tersebut dimaksudkan untuk menjaga berkas agar selalu dalam keadaan terenkripsi. Tampilan setelah sebuah berkas berhasil didekripsi diilustrasikan pada Gambar 10.



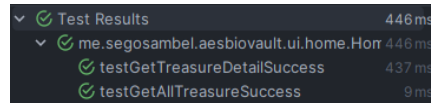
Gambar 10. Tampilan Dekripsi Berkas

#### 4. Pembahasan

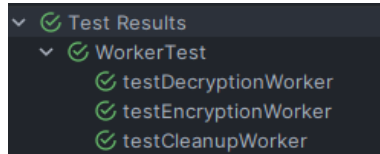
Pada tahap pengujian ini, digunakan *Automated Testing* yang terdiri dari 2 jenis tes yakni *unit test* dan *instrumentation test*. Skrip *Automated Testing* dibagi menjadi 3 bagian, 1 *Unit Test* dan 2 *Instrumentation Test*. *Unit Test* digunakan untuk menguji komponen-komponen perangkat lunak secara terisolasi. Sementara *Instrumentation Test* digunakan untuk menguji interaksi antara



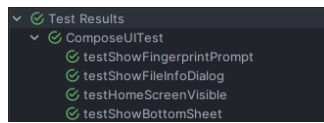
komponen-komponen perangkat lunak dalam konteks yang lebih luas. Hasil dari tes tersebut adalah sebagai berikut:



Gambar 11. Hasil Unit Test



Gambar 12. Hasil Instrumentation Test



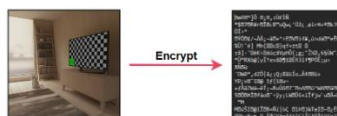
Gambar 13. Hasil Instrumentation Test

Tabel 1. Hasil Pengujian

Jenis Pengujian	Tes yang Dilakukan	Nama Prosedur Tes	Hasil
Unit Test	Mendapatkan Semua Data Berkas yang Tersimpan	<i>testGetAllTreasureSuccess</i>	Berhasil
Unit Test	Mendapatkan Detail Berkas yang Tersimpan	<i>testGetTreasureDetailSuccess</i>	Berhasil
Instrumentation Test	Melakukan Enkripsi Berkas	<i>testEncryptionWorker</i>	Berhasil
Instrumentation Test	Melakukan Dekripsi Berkas	<i>testDecryptionWorker</i>	Berhasil
Instrumentation Test	Menghapus Berkas Tersimpan	<i>testCleanupWorker</i>	Berhasil
Instrumentation Test	Menampilkan Halaman Login	<i>testShowFingerprintPrompt</i>	Berhasil
Instrumentation Test	Menampilkan Halaman Utama	<i>testHomeScreenVisible</i>	Berhasil
Instrumentation Test	Menampilkan BottomSheet	<i>testShowBottomSheet</i>	Berhasil
Instrumentation Test	Menampilkan Informasi File	<i>testShowFileInfoDialog</i>	Berhasil

Data pada Gambar 11-13 dan Tabel 1 menunjukkan hasil pengujian terhadap beberapa skenario yang telah ditulis pada skrip *Automated Testing*. Berdasarkan hasil pengujian, dapat disimpulkan bahwa penelitian ini telah berhasil menguji berbagai skenario dengan baik, menunjukkan tingkat keandalan yang tinggi dalam melakukan tugas enkripsi dan dekripsi. Hasil yang konsisten dan akurat dari pengujian menunjukkan bahwa implementasi enkripsi dan dekripsi dalam aplikasi yang dikembangkan dapat diandalkan dan efektif dalam menjaga keamanan data.

Enkripsi dan Dekripsi dilakukan menggunakan aplikasi yang telah dibuat. Pada Gambar 14, diilustrasikan bentuk berkas sebelum dan sesudah di-Enkripsi.



Gambar 14. Berkas Gambar setelah di-Ekripsi



Enkripsi di atas melibatkan *plaintext* dari Gambar yang kemudian di-proses oleh kelas EncryptedFile untuk menjadi berkas yang berisi *ciphertext*. Sedangkan pada Gambar 15, diilustrasikan proses sebaliknya yaitu Dekripsi.



**Gambar 15.** Berkas Gambar setelah di-Dekripsi

## 5. Penutup

Hasil penelitian pada Aplikasi Keamanan Berkas dengan Enkripsi AES dan Biometrik Sidik Jari berbasis Android memberikan beberapa kesimpulan penting. Pertama, aplikasi berhasil mengintegrasikan enkripsi AES-128 dengan verifikasi sidik jari, menciptakan lapisan keamanan tambahan yang canggih dan mudah digunakan. Kedua, pengujian fungsionalitas dan kinerja telah menunjukkan bahwa aplikasi berfungsi sesuai harapan. Proses enkripsi dan dekripsi berkas berjalan dengan baik, sementara aplikasi tetap responsif dan efisien. Kesimpulan dari penelitian ini adalah bahwa Aplikasi Keamanan Berkas Dengan Enkripsi AES dan Biometrik Sidik Jari berbasis Android adalah sebuah solusi yang efektif untuk meningkatkan keamanan data di era digital saat ini. Aplikasi berhasil dibangun dan diuji dengan baik serta mampu memberikan lapisan keamanan tambahan yang canggih dan mudah digunakan. Dengan demikian, aplikasi yang dikembangkan memiliki potensi untuk digunakan secara luas untuk meningkatkan keamanan data pada perangkat Android. Aplikasi yang dibuat merupakan salah satu langkah dalam menjaga kerahasiaan dan integritas data di perangkat Android, sehingga pengguna dapat mendapatkan rasa aman dalam mengakses dan menyimpan berkas penting mereka. Namun, sebagai saran untuk penelitian mendatang agar mempertimbangkan penggunaan AES-256 sebagai algoritma enkripsi utama, dikarenakan algoritma AES-256 telah didukung oleh kelas EncryptedFile secara default yang dapat memudahkan dalam proses pengembangan.

## Referensi

- [1] E. M. De Los Reyes, A. M. Sison, and R. P. Medina, "File encryption based on reduced-round AES with revised round keys and key schedule," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 897–905, 2019, doi: 10.11591/ijeecs.v16.i2.pp897-905.
- [2] B. E. Widodo and A. S. Purnomo, "Implementasi Advanced Encryption Standard Pada Enkripsi Dan Dekripsi Dokumen Rahasia Ditintelkam Polda Diy," *Jurnal Teknik Informatika (Jutif)*, vol. 1, no. 2, pp. 69–77, 2020, doi: 10.20884/1.jutif.2020.1.2.21.
- [3] D. Wong, *Real-World Cryptography*. Manning Publications, 2021.
- [4] Binantara Parmadi, "Implementasi Algoritma Kriptografi Elgamal pada Data Text," vol. 5, no. 1, 2017, [Online]. Available: <http://jurnal.stiki.ac.id/J-INTECH/article/view/26>
- [5] P. L. T. Irawan, "Implementasi Teknik Kriptografi Stream Cipher Salsa20 Untuk Pengamanan Basis Data," *Smatika Jurnal*, vol. 5, no. 02, pp. 88–92, 2015, [Online]. Available: <http://jurnal.stiki.ac.id/SMATIKA/article/view/97>
- [6] D. Selent, "Advanced Encryption Standard," *Rivier Academic Journal*, vol. 6, no. 2, 2010.
- [7] A. Prameshwari and N. P. Sastra, "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen," *Eksplora Informatika*, vol. 8, no. 1, p. 52, 2018, doi: 10.30864/eksplora.v8i1.139.
- [8] K. Muttaqin and J. Rahmadoni, "Analysis and Design of File Security System Aes (Advanced Encryption Standard) Cryptography Based," *Journal of Applied Engineering and Technological Science*, vol. 1, no. 2, pp. 113–123, 2020, doi: 10.37385/jaets.v1i2.78.

- [9] E. Setyaningsih, "Keamanan file dokumen menggunakan algoritme Advanced Encryption Standard pada aplikasi berbasis Android," *Jnanaloka*, pp. 11–23, Apr. 2020, doi: 10.36802/jnanaloka.2020.v1-no1-13.
- [10] A. I. Suranta and D. V. S. Y. Sakti, "Penerapan Algoritma Aes (Advance Encryption Standart) 128 Untuk Enkripsi Dokumen Di Pt. Gunung Geulis Elok Abadi," *Skanika*, vol. 5, no. 1, pp. 1–10, 2022, doi: 10.36080/skanika.v5i1.2118.
- [11] Y. H. Jo, S. Y. Jeon, J. H. Im, and M. K. Lee, "Security analysis and improvement of fingerprint authentication for smartphones," *Mobile Information Systems*, vol. 2016, no. Krait 400, 2016, doi: 10.1155/2016/8973828.
- [12] S. Iqbal *et al.*, "A novel mobile wallet model for elderly using fingerprint as authentication factor," *IEEE Access*, vol. 8, pp. 177405–177423, 2020, doi: 10.1109/ACCESS.2020.3025429.
- [13] M. Fahmi Shamsudin and N. Hidayah Ab Rahman, "An Authentication of Carpooling Apps Using OTP and Fingerprint," *Applied Information Technology And Computer Science*, vol. 2, no. 1, pp. 1–10, 2021.
- [14] B. Acharya and K. Sahu, "Software Development Life Cycle Models: A Review Paper," *International Journal of Advanced Research in Engineering and Technology*, vol. 11, no. 12, pp. 169–176, 2020, doi: 10.34218/IJARET.11.12.2020.019.
- [15] H. Himawan and M. Yanu F, *Interface USER EXPERIENCE*. 2020.
- [16] N. Smyth, *Android Studio 3.0 Development Essentials Android 8 Edition*. Payload Media, 2017.
- [17] R. B. D. Putra, E. S. Budi, and A. R. Kadafi, "Perbandingan Antara SQLite, Room, dan RBDLiTe Dalam Pembuatan Basis Data pada Aplikasi Android," *JURIKOM (Jurnal Riset Komputer)*, vol. 7, no. 3, p. 376, 2020, doi: 10.30865/jurikom.v7i3.2161.
- [18] P. Kong, L. Li, J. Gao, K. Liu, T. F. Bissyandé, and J. Klein, "Automated testing of Android apps: A systematic literature review," *IEEE Trans Reliab*, vol. 68, no. 1, pp. 45–66, 2019, doi: 10.1109/TR.2018.2865733.