https://jurnal.pustakagalerimandiri.co.id/index.php/pustakadata
DOI: https://doi.org/10.55382/jurnalpustakadata.v5i2.1373

Jurnal Pustaka Data

Pusat Akses Kajian Database, Analisa Teknologi, dan Arsitektur Komputer

Vol. 5 No. 2 (2025) 295 – 302
E ISSN: 2809-7483

Pengembangan Aplikasi *Notes* Terenkripsi di Android Menggunakan *Python* dan *Library Cryptography*

Nidia Enjelita Saragih[1], Robiatul Adawiyah[2], Ermayanti Astuti[3]
Sistem Informasi, Teknik dan Ilmu Komputer, Universitas Pembinaan Masyarakat Indonesia
[1]nidia.1924@gmail.com, [2]robiatulbintisyarifuddin@gmail.com, [3]ermaemma0216@gmail.com

Abstract

In the era of rapid technological advancement, information can be easily accessed, including personal data. Therefore, an effective data protection mechanism is required, especially for mobile devices used by individuals on a daily basis. One of the applications that requires such protection is the Notes application, as it is often used to store personal information that is vulnerable to unauthorized access. This research aims to develop an encrypted notes application for Android using the Python programming language with the assistance of the Cryptography library. The method applied is symmetric encryption using the Advanced Encryption Standard (AES) algorithm to secure user note contents. The application was developed using the Kivy framework, allowing it to run on Android devices. The testing results show that the encryption and decryption processes work properly and remain stable. Based on performance testing with data sizes ranging from 1 KB to 1 MB, the application achieved an average encryption time of 0.0009 seconds and an average decryption time of 0.00025 seconds. The processing time increases linearly with the data size but remains within a highly efficient range for mobile applications. Furthermore, stored notes cannot be accessed without the correct encryption key. Thus, this study demonstrates that Python can be effectively utilized in developing mobile applications that prioritize data security, particularly for protecting personal information on Android devices.

Keywords: Cryptography, Python,Android, AES, Notes Application

Abstrak

Di era perkembangan pesat teknologi seperti saat ini, informasi bisa dengan mudah diakses. Tak terkecuali data yang bersifat pribadi. Karena itu dibutuhkan sebuah mekanisme pengamanan data pribadi, utamanya pada perangkat mobile yang digunakan setiap orang. Salah satu aplikasi yang membutuhkan mekanisme pengamanan ini adalah aplikasi catatan (notes). Sebab seringkali notes digunakan sebagai alat untuk menyimpan informasi-informasi pribadi yang rentan terhadap akses tidak sah. Penelitian ini bertujuan untuk mengembangkan aplikasi catatan terenkripsi berbasis Android menggunakan bahasa pemrograman Python dengan bantuan library Cryptography. Adapun metode yang digunakan adalah enkripsi simetris menggunakan algoritma Advanced Encryption Standard (AES) untuk mengamankan isi catatan pengguna. Aplikasi dikembangkan menggunakan framework Kivy sehingga dapat dijalankan di Android. Hasil pengujian menunjukkan bahwa proses enkripsi dan dekripsi berjalan dengan baik dan stabil. Berdasarkan uji performa terhadap data berukuran 1 KB hingga 1 MB, diperoleh rata-rata waktu enkripsi sebesar 0,0009 detik dan rata-rata waktu dekripsi sebesar 0,00025 detik**.** Waktu pemrosesan meningkat secara linear terhadap ukuran data, tetapi tetap dalam kategori sangat efisien untuk aplikasi mobile. Selain itu, catatan yang disimpan tidak dapat diakses tanpa kunci enkripsi yang sesuai. Dengan demikian, penelitian ini membuktikan bahwa Python dapat digunakan secara efektif dalam pengembangan aplikasi mobile yang mengutamakan keamanan data, serta menjadi alternatif fleksibel terhadap bahasa pemrograman konvensional seperti Java dan Kotlin.

Kata kunci: Kriptografi, Python, Android, AES, Aplikasi Notes

Submitted : 17-10-2025 | Reviewed : 24-10-2025 | Accepted : 26-11-2025

1. Pendahuluan

Di era perkembangan teknologi seperti saat ini, penyebaran data dan informasi terjadi dengan sangat cepat dan luas. Yang membawa banyak kemudahan bagi kehidupan manusia untuk belajar, berinteraksi dengan banyak orang, dan juga memunculkan aneka ragam pekerjaan yang belum pernah ada sebelumnya. Berbagai platform media sosial muncul dan menawarkan banyak pilihan bagi pengguannya, baik untuk sekedar menjadikan sebagai alat menikmati pertemanan dunia maya, mendapatkan akses informasi, atau bahkan menjadikan media sosial sebagai sarana memperoleh penghasilan.

Akan tetapi, segala bentuk kemudahan yang ditawarkan ini justru bisa membawa bahaya tersendiri. Dimana kemudahan akses data dan informasi juga terjadi pada data yang bersifat pribadi. Dampaknya data tersebut bisa diakses untuk diubah, dihapus, atau disalahgunakan oleh pihak yang tidak berwenang [1].

Karena itu, dibutuhkan sebuah mekanisme pengamanan data, utamanya pada data yang bersifat pribadi atau hal-hal menyangkut finansial, yang rentan diakses dan disalahgunakan orang lain. Kriptografi adalah solusi atas persoalan ini [2].

Dengan kriptografi, data yang bersifat rahasia akan diubah ke dalam bentuk lain sehingga tidak mudah diakses oleh siapapun kecuali pihak yang memiliki otoritas. Proses pengubahan ini disebut sebagai penyandian(enkripsi). Yakni proses mengubah plainteks(pesan asli) yang dapat dibaca dan dipahami maknanya, menjadi cipherteks(pesan terkode atau acak) yang tidak lagi bisa dipahami maknanya dengan menggunakan sebuah key (kunci) [3].

Advance Encryption Standar (AES) adalah salah satu algoritma kriptografi yang masih cukup tangguh untuk digunakan dalam mengamankan data. Algoritma AES dikenal memiliki ketahanan tinggi terhadap serangan brute force karena struktur kunci dan jumlah putarannya yang kompleks [4].

Smartphone merupakan bentuk dari perkembangan teknologi yang paling dekat dengan kehidupan masyarakat dewasa ini. Banyak pekerjaan yang awalnya dikerjakan menggunakan komputer, mulai tergantikan dengan teknologi smartphone. Perusahaan maupun lembaga-lembaga resmi turut pula menjadikan smartphone sebagai alat untuk meningkatkan efektivitas komunikasi dan kinerja karyawan. Hampir semua perusahaan atau lembaga memiliki sistem informasi tersendiri yang bisa diakses dengan menggunakan smartphone oleh seluruh karyawan dengan menggunakan username dan password dengan kriteria-kriteria tertentu. Belum lagi jika perusahaan itu memiliki beberapa sistem informasi untuk tujuan-tujuan berbeda yang mengharuskan karyawan memiliki data privat berupa nama dan username yang berbeda pula.

Ditambah dengan aplikasi-aplikasi perbankan mobile yang juga menawarkan kemudahan transaksi dengan menggunakan smartphone. Akhirnya, pengguna smartphone dituntut untuk mampu mengingat setiap data pribadi utama berupa nama dan username setiap akun pada aplikasi perbankan maupun sistem informasi perusahaan atau yang lainnya.

Salah satu aplikasi android yang bisa dimanfaatkan untuk menyimpan data-data privat ini adalah notes. Notes juga seringkali dimanfaatkan untuk menyimpan informasi-informasi pribadi yang akan berbahaya jika diakses oleh orang lain. Karena itu, dibutuhkan sebuah mekanisme pengamanan pesan pada notes android yang membuat pesan tidak mudah diakses oleh orang lain.

Penelitian ini secara khusus bertujuan mengimplementasikan pengamanan pesan dengan algoritma AES pada aplikasi notes terenkripsi yang berjalan pada smartphone bersistem operasi android. Pengembangan aplikasi dilakukan menggunakan python.

Penelitian sebelumnya sebagian besar dilakukan menggunakan bahasa Java atau Kotlin, karena keduanya merupakan bahasa utama dalam pengembangan aplikasi Android. Namun, penelitian ini berbeda karena menggunakan Python-Kivy, yang lebih fleksibel, lintas platform, dan mudah dalam integrasi dengan pustaka kriptografi modern. Framework Kivy juga memungkinkan aplikasi Python dikompilasi menjadi *Android Package (APK)* tanpa kehilangan performa secara signifikan.

Pemilihan Python dalam penelitian ini didasarkan pada beberapa pertimbangan ilmiah:

1. Python memiliki sintaks yang sederhana dan mudah dipelajari, sehingga mempercepat proses pengembangan dan pengujian aplikasi [5].
2. Library Cryptography pada Python mendukung implementasi AES yang sudah teruji dan sesuai dengan standar keamanan industri (*industry-grade implementation*).
3. Python memiliki dukungan komunitas yang besar serta kompatibilitas tinggi dengan berbagai sistem operasi, termasuk Android dan Windows.

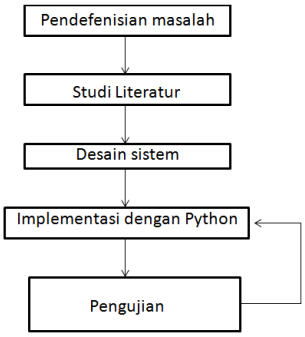
Submitted : 17-10-2025 | Reviewed : 24-10-2025 | Accepted : 26-11-2025

4. Penggunaan framework Kivy menjadikan Python mampu membuat antarmuka grafis (GUI) yang responsif di perangkat mobile, menjadikannya alternatif kuat terhadap Java dan Kotlin.

Dengan demikian, penelitian ini tidak hanya mengimplementasikan enkripsi AES pada aplikasi Notes, tetapi juga menunjukkan potensi Python sebagai bahasa pemrograman yang efisien dan aman untuk pengembangan aplikasi Android.

2. Metode Penelitian

Metodologi penelitian merupakan penjelasan atas prosedur dan tahapan kegiatan yang dijalankan peneliti untuk mencapai tujuan atau merancang solusi atas persoalan yang telah terdefenisi. Seperti yang telah dijelaskan sebelumnya, bahwa penelitian ini bertujuan merancang sebuah aplikasi notes terenkripsi pada android dengan menggunakan algoritma AES dan diimplementasikan menggunakan bahasa pemrograman Python. Adapun yang menjadi tahapan dalam pelaksanaan penelitian ini, bisa dilihat dari gambar 1.



Gambar 1. Tahapan metodologi penelitian

2.1. Pendefinisian Masalah
Dalam tahapan awal ini, peneliti berusaha menangkap persoalan yang terjadi dalam keseharian masyarakat pengguna android, utamanya dalam upaya penyimpanan data rahasia. Persoalan ini kemudian coba dirumuskan dan dicari solusinya dengan melakukan studi literatur yang menjadi tahapan berikutnya.

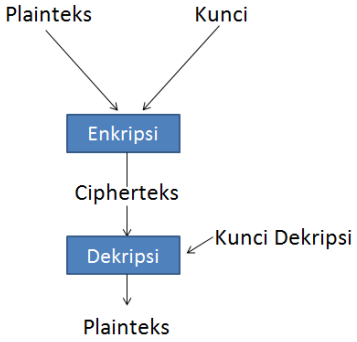
2.2. Studi Literatur
Tahapan ini dilakukan dengan pengkajian mendalam terhadap beberapa penelitian sebelumnya yang terkait dengan topik yang sedang diangkat. Seperti penelitian yang dilakukan oleh Tasya D. A dan Yuli A. yang menunjukkan bahwa penggunaan algoritma kriptografi AES mampu diterapkan dengan baik pada data perusahaan dan mampu meningkatkan keamanan. File dokumen berhasil dienkripsi dan tidak dapat dibuka kecuali dengan menggunakan kunci dan token yang sesuai [6].

Penelitian yang dilakukan oleh Muhammad Ryan A. dan Pristi S, dengan menggunakan algoritma AES telah menghasilkan peningkatan keamanan pada sistem E-marketplace. Kunci yang panjang dan banyaknya proses putaran membuat algoritma ini efektif untuk digunakan mengamankan data transaksi pembayaran dari pengguna [7].

Maulyanda, dkk pada tahun 2021 juga telah membuktikan bahwa penggunaan algoritma AES bersama Triple Key, dan Base64 telah berhasil mengamankan data berupa e-KTP sehingga tidak bisa diakses oleh orang yang tidak berhak [8].

Setelah melakukan pengkajian terhadap penelitian sebelumnya, tahapan ini dilanjutkan dengan mengumpulkan teori-teori dasar yang berkenaan dengan tema yang hendak diteliti. Teori-teori tersebut dijelaskan dalam beberapa sub-bab berikut.

2.2.1 Kriptografi
Kriptografi adalah ilmu yang mempelajari mengenai teknik-teknik matematika yang berkaitan dengan aspek keamanan informasi data atau keamanan pesan dengan menggunakan dua proses dasar kriptografi yaitu enkripsi dan dekripsi [9].



Gambar 2. Proses enkripsi dan dekripsi

Dari gambar 1, terlihat jelasn bahwa terdapat dua kunci yang digunakan dalam proses enkripsi dan dekripsi. Kesamaan dan perbedaan pada kedua kunci tersebut membuat algoritma kriptografi terbagi menjadi dua jenis, yakni algoritma kriptografi simetri dan asimetri.

Algoritma kriptografi simetris menggunakan kunci yang sama dalam proses enkripsi dan dekripsi. Sedangkan algoritma asimetris menggunakan kunci yang berbeda.

2.2.2 Advance Encyption Standard(AES)
AES merupakan algoritma kriptografi simetris yang bekerja pada block cipher.[10]Berdasarkan kunci yang digunakan, dikenal beberapa variasi dari algoritma AES. Yakni AES 128 , AES 192, dan AES 256. Setiap variasi menggunakan panjang kunci dan jumlah putaran yang berbeda. Perbedaan ketiganya dapat dilihat pada tabel 1.

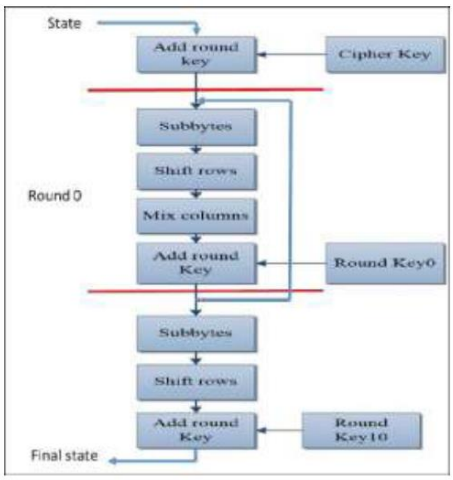
Tabel 1. Variansi pada AES

Varian AES	Panjang Kunci(NK Words)	Ukuran Blok(Nb words)	Jumlah Putaran(Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Perbedaan jumlah putaran pada masing-masing variansi tersebut menambah tingkat kesulitan dalam memecahkan kode yang dihasilkan. Terdapat beberapa transformasi yang mesti dilakukan pada setiap putaran, yaitu :

a. Substitusi byte. Dilakukan dengan menggunakan table S-Box dalam tujuan meningkatkan kekacauan.
b. Shift Rows. Yakni mengubah matriks data. Dilakukan untuk meningkatkan difusi.
c. Mix Coloums. Yaitu operasi matriks yang dilakukan untuk menyebarkan perubahan ke setiap kolom.
d. Add Round Key. Yakni menggabungkan data dengan kunci enkripsi yang telah diperluas dari kunci awal [11].

Gambar 3 akan memudahkan memahami proses yang terjadi selama enkripsi.



Gambar 3. Proses Enkripsi pada AES-128

Setelah cipherteks dihasilkan dari proses di atas, proses dekripsi dimulai dengan menggunakan Round-Key yang diperoleh dalam proses ekspansi kunci. Dilanjutkan dengan proses inversi *shift rows* dan inversi *sub bytes* yang merupakan kebalikan dari proses enkripsi sebelumnya. Operasi Add Round Key dilakukan sebanyak jumlah putaran yang sesuai dengan variansi yang digunakan. Penyebaran perubahan di antara kolom matriks blok data dilakukan dengan tahapan inversi mix coloumns. Kesemua tahapan ini dilakukan secara berulang sesuai jumlah putaran yang dibutuhkan sesuai penjelasan pada tabel sebelumnya sehingga menghasilkan plainteks yang sama tepat dengan yang diinput sebelumnya.

Walaupun AES-256 secara teoritis tahan terhadap serangan brute-force, aspek implementasi di tingkat aplikasi tetap berpotensi menimbulkan kerentanan. Seperti yang ditunjukkan dalam penelitian Razaghpanah et al. [12], kesalahan konfigurasi kriptografi dan penggunaan TLS yang tidak tepat pada aplikasi Android dapat membuka peluang serangan meskipun algoritma yang digunakan tergolong aman. Oleh karena itu, pengujian keamanan aplikasi perlu mencakup verifikasi penerapan algoritma dan protokol enkripsi agar tidak terjadi *crypto misuse*.

2.2.3 Android

Android merupakan sistem operasi yang berjalan khusus pada perangkat mobile sehingga memungkinkan pengguna melakukan berbagai aktivitas berbantuan piranti cerdas dengan mobilitas tinggi. Sistem platform terbuka yang diterapkan dalam sistem operasi ini memungkinkan pengembang menciptakan aplikasi mereka sendiri. Ditambah dengan dukungan komunitas Open Source dunia yang membuat perkembangan sistem operasi ini semakin cepat [13].

2.2.4 Notes

Notes adalah aplikasi yang dikenal luas penggunaannya pada android. Digunakan untuk menyimpan berbagai informasi yang sering dibutuhkan namun rentan terlupakan. Notes mengadaptasi fungsi buku catatan kecil yang biasa digunakan orang dalam mencatat berbagai hal penting.

2.2.5 Python

Salah satu bahasa pemrograman yang paling mudah dipelajari adalah Python.Sering digunakan dalam pengembangan perangkat lunak di berbagai bidang, termasuk kriptografi. Pyton menyediakan berbagai pustaka yang memudahkan pengguna dalam melakukan berbagai fungsi komputasi numerik [14].

Penelitian ini menggunakan bahasa Python karena memiliki pustaka yang kuat untuk komputasi dan kriptografi. Aplikasi dikembangkan dalam lingkungan berikut:

1. Python versi 3.10
2. Kivy versi 2.2 untuk antarmuka grafis (GUI) dan kompatibilitas Android
3. Library Cryptography versi 41.0 untuk implementasi algoritma AES
4. Sistem operasi pengembangan: Windows 11 (64-bit)
5. Database: SQLite sebagai media penyimpanan lokal terenkripsi

Dengan kombinasi tersebut, aplikasi dapat dijalankan baik di lingkungan desktop maupun dikompilasi menjadi file APK untuk sistem Android.

2.3. Rencana Pengujian Sistem (Test Plan)
Pengujian dilakukan untuk memastikan keandalan, fungsionalitas, dan keamanan aplikasi. Rencana pengujian meliputi:

2.3.1 Pengujian Fungsional
1. Uji Enkripsi–Dekripsi: Memastikan data yang dienkripsi dapat didekripsi kembali dengan hasil identik terhadap plainteks asli.
2. Uji Kesalahan Kunci: Memastikan proses dekripsi gagal jika kunci yang dimasukkan berbeda dari kunci enkripsi.
3. Uji Input Pengguna: Memastikan sistem menolak input kosong atau karakter yang tidak valid.
4. Uji Penyimpanan Database: Memverifikasi bahwa data yang tersimpan di database berada dalam bentuk ciphertext.

2.3.2 Pengujian Keamanan
1. Uji Brute-Force Resistance: Menganalisis tingkat ketahanan terhadap serangan *brute-force* berdasarkan panjang kunci AES-256 (2^{256} kemungkinan kunci).
2. Uji Integritas Data: Memastikan perubahan kecil pada ciphertext menyebabkan hasil dekripsi yang gagal atau tidak valid.
3. Uji Nonce dan Salt: Memastikan hasil ciphertext berbeda untuk data yang sama apabila nonce dan salt berbeda.
4. Uji Autentikasi Tag: Mengevaluasi validitas *authentication tag* untuk mencegah modifikasi ciphertext oleh pihak ketiga.
5. Uji Efisiensi Waktu: Mengukur waktu proses enkripsi dan dekripsi berdasarkan variasi ukuran data (1 KB hingga 1 MB).

Melalui tahapan metodologi yang sistematis, mulai dari pendefinisian masalah, studi literatur, hingga uji keamanan, penelitian ini memastikan bahwa aplikasi *Secure Notes* tidak hanya berfungsi dengan baik, tetapi juga memenuhi standar yang dibutuhkan untuk penggunaan pada platform Android.

3. Hasil dan Pembahasan

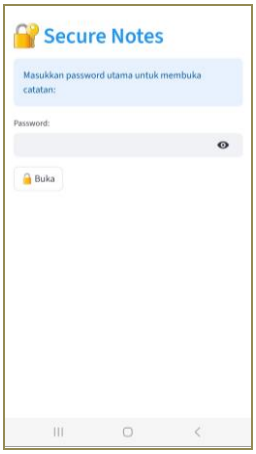
Aplikasi *Secure Notes* yang dikembangkan dalam penelitian ini dibangun menggunakan bahasa pemrograman Python dengan framework Kivy untuk antarmuka grafis (*Graphical User Interface*) yang kompatibel dengan sistem operasi Android. Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi utama aplikasi berjalan dengan baik, serta untuk mengukur efisiensi algoritma AES (Advanced Encryption Standard) yang digunakan dalam proses enkripsi dan dekripsi.

Pengujian dilakukan berdasarkan rencana pengujian yang dijelaskan pada bagian 2.3, meliputi pengujian fungsional, keamanan, dan efisiensi waktu eksekusi.

Adapun skenario pengujian terhadap aplikasi dilakukan dengan tiga langkah :

1. Memastikan semua fungsi berjalan tanpa error.
2. Memastikan bahwa jika kunci yang diinputkan tidak sama dengan kunci enkripsi, maka dekripsi tidak bisa dilakukan.
3. Membandingkan hasil dekripsi dengan plainteks asli, yang membuktikan bahwa aplikasi berhasil memgubah plainteks menjadi cipherteks dengan proses enkripsi dan dan mengembalikannya ke bentuk semula dengan proses dekripsi.

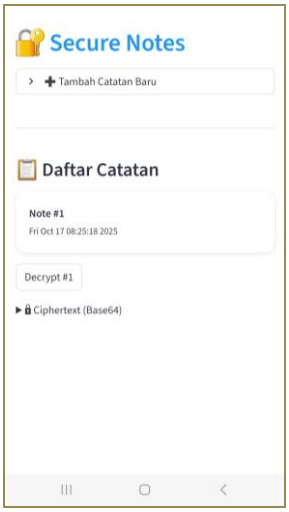
Karena itu, proses pengujian akan dijelaskan bersamaan dengan penjelasan interface aplikasi berikut ini.



Gambar 4. Interface awal aplikasi

Pada halaman depan aplikasi seperti gambar 4, pengguna akan diminta memasukkan “Password” yakni nilai kunci enkripsi yang sudah ditentukan di awal. Sebagai algoritma simetris, kunci enkripsi pada AES juga berfungsi sebagai kunci dekripsi.

Jika kunci yang dimasukkan tepat, maka pengguna akan diarahkan pada halaman berikutnya, seperti gambar 5.



Gambar 5. Interface daftar catatan

Submitted : 17-10-2025 | Reviewed : 24-10-2025 | Accepted : 26-11-2025

Pengguna bisa mengklik tombol tambah catatan untuk membuat catatan baru. Di dalamnya pengguna memasukkan judul dan isi catatan untuk dienkripsikan dengan algoritma AES. Proses ini bisa dilihat dalam gambar 6.

Gambar 6. Menu Tambah Catatan Baru

Ketika pengguna mengklik tombol simpan, maka judul maupun isi catatan akan secara otomatis tersimpan dalam bentuk cipherteks.

Hasil penyimpanan catatan dapat dilihat pada bagian Daftar Catatan yang terletak di bawah. Seperti gambar 7

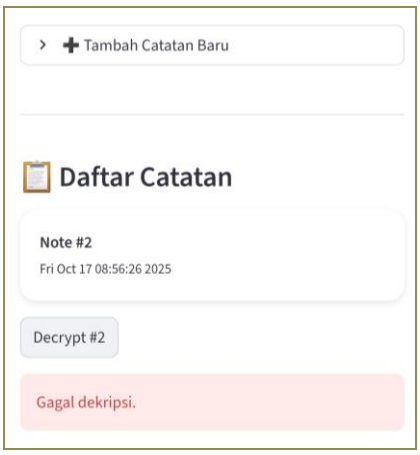
Gambar 7. Tampilan Hasil Enkripsi Catatan

Untuk bisa melihat isi catatan, pengguna mesti terlebih dahulu mengklik tombol Decrypt. Pesan tersimpan dalam bentuk Cipherteks akan didekripsi dan kembali ke bentuk semula. Perbandingan catatan sebelum dekripsi dan setelah dekripsi bisa dilihat pada gambar berikut.

Gambar 8. Perbandingan Hasil Dekripsi Catatan dengan Catatan Asli.

Pada gambar 8 tampak bahwa catatan yang telah didekripsikan kembali ke bentuk catatan asli sebelum dienkripsi, jika menggunakan kunci yang benar.

Sedangkan, jika kunci yang dimasukkan salah atau tidak tepat, maka aplikasi tidak bisa melakukan proses dekripsi. Hal ini dibuktikan pada gambar 9.



Gambar 9. Interface Aplikasi Jika Kunci yang Dimasukkan Salah.

Dengan demikian, bisa disimpulkan bahwa seluruh aplikasi telah berjalan sesuai harapan. Skenario pengujian juga telah dilaksanakan sebagaimana yang dijelaskan sebelumnya.

Dari aplikasi yang telah berjalan, dilakukan beberapa kali percobaan pada ukuran data yang berbeda dan diperoleh data waktu eksekusi seperti tabel 2.

Tabel 2. Pengujian Waktu Eksekusi

Percobaan	Ukuran (KB)	Waktu Enkripsi (detik)	Waktu Dekripsi (detik)
1	1	0.000069	0.000014
2	10	0.000071	0.000012
3	50	0.000102	0.000034
4	100	0.000107	0.000043
5	250	0.000241	0.000098
6	500	0.000347	0.000225
7	1000	0.001480	0.000382

Berdasarkan tabel pengujian waktu eksekusi, proses enkripsi dan dekripsi menggunakan algoritma AES (Advanced Encryption Standard) menunjukkan hasil yang sangat efisien. Waktu eksekusi enkripsi dan dekripsi meningkat secara proporsional terhadap

ukuran data yang diuji, namun kenaikannya bersifat linear dan relatif kecil**.**

Pada data berukuran 1 KB, waktu enkripsi hanya membutuhkan sekitar 0.000069 detik dan dekripsi sekitar 0.000014 detik. Sementara pada data berukuran 1 MB (1000 KB)**,** waktu enkripsi hanya naik menjadi 0.001480 detik, dan dekripsi menjadi 0.000382 detik. Artinya, peningkatan ukuran data sebesar 1000 kali hanya meningkatkan waktu proses sekitar 20 kali, yang masih dalam batas efisiensi sangat tinggi.

Hal ini menunjukkan bahwa algoritma AES yang digunakan mampu memproses data berukuran besar dengan overhead komputasi yang minimal, sehingga sangat sesuai diterapkan pada aplikasi mobile berbasis Python seperti *Secure Notes* ini. Selain itu, hasil pengujian juga membuktikan bahwa proses dekripsi cenderung sedikit lebih cepat dibandingkan enkripsi, yang umum terjadi karena mekanisme enkripsi melibatkan proses pembangkitan tag autentikasi tambahan.

Algoritma AES (Advanced Encryption Standard) menggunakan panjang kunci 256 bit (AES-256) pada aplikasi ini. Panjang kunci sebesar itu menghasilkan 2^{256} kemungkinan kombinasi kunci, atau sekitar $1{,}15 \times 10^{77}$ kombinasi. Bahkan jika sebuah sistem mampu mencoba 1 miliar kunci per detik, diperlukan waktu lebih dari $3{,}67 \times 10^{60}$ tahun untuk menebak satu kunci yang benar melalui *brute-force attack*. Oleh karena itu, AES-256 dianggap praktis tidak dapat dipecahkan dengan teknologi komputasi saat ini**.**

Selain itu, penggunaan nonce (number used once) dan salt yang dihasilkan secara acak pada setiap proses enkripsi membuat hasil ciphertext selalu berbeda walaupun plainteks dan kunci yang digunakan sama, sehingga serangan berbasis pola atau *dictionary attack* menjadi tidak efektif.

Dengan kombinasi tersebut, sistem ini telah memenuhi aspek kerahasiaan (confidentiality)**,** integritas (integrity), dan keaslian data (authenticity) sesuai prinsip dasar keamanan informasi.

Dalam pengimplementasian algoritma AES dalam Phyton-Kivy pada penelitian ini, dapat disimpulkan beberapa kelebihan dari penggunaan Phyton-Kivy, yaitu :

a. Portabilitas tinggi: Kivy memungkinkan aplikasi Python dijalankan di berbagai platform (Windows, Linux, dan Android) tanpa perubahan kode yang signifikan.
b. Antarmuka modern: Kivy menyediakan dukungan GUI yang interaktif dan mendukung *touchscreen*, sesuai kebutuhan perangkat Android.
c. Integrasi cepat dengan library kriptografi: Modul *cryptography* Python dapat digunakan langsung dalam Kivy tanpa perlu pustaka tambahan kompleks.
d. Waktu pengembangan singkat: Python memiliki sintaks sederhana dan pustaka luas yang mempercepat pengembangan aplikasi prototipe.

Adapun keterbatasan penggunaan Phyton-Kivy yakni:

a. Kinerja lebih lambat dibandingkan bahasa native (Java/Kotlin): Karena Python bersifat interpreted, waktu eksekusi terutama untuk proses komputasi besar sedikit lebih tinggi dibandingkan implementasi di bahasa kompilasi langsung.
b. Kebutuhan memori lebih besar: Aplikasi Kivy umumnya memerlukan RAM lebih tinggi dibandingkan aplikasi Android native.
c. Ukuran file aplikasi lebih besar: File APK hasil kompilasi Python–Kivy memiliki ukuran yang lebih besar akibat dependensi runtime Python.
d. Integrasi dengan fitur sistem Android masih terbatas, misalnya akses sensor atau latar belakang sistem memerlukan penyesuaian tambahan.

4. Kesimpulan

Aplikasi Notes terenkripsi berhasil dikembangkan menggunakan bahasa pemrograman Python dengan memanfaatkan library Cryptography sebagai inti sistem keamanan. Aplikasi ini mampu melakukan proses enkripsi dan dekripsi catatan (notes) secara otomatis berdasarkan kunci yang diturunkan dari password pengguna.

Implementasi algoritma AES (Advanced Encryption Standard) memberikan jaminan kerahasiaan (confidentiality) dan integritas data (integrity), karena setiap ciphertext disertai authentication tag yang mencegah modifikasi data secara tidak sah.

Rencana pengujian yang diterapkan berhasil membuktikan bahwa aplikasi bekerja sesuai standar keamanan dan efisiensi yang dirancang.

Aplikasi belum diuji terhadap serangan *brute-force* atau *side-channel attack* yang berpotensi mengeksploitasi kelemahan pada sistem keamanan fisik maupun perilaku enkripsi.

Penerapan autentikasi biometrik dan pengembangan sistem multi-user(misalnya sidik jari atau pengenalan wajah) bisa dilakukan dalam penelitian selanjutnya guna memperkuat lapisan keamanan akses terhadap data terenkripsi.

Ucapan Terimakasih

Penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada Universitas Pembinaan Masyarakat Indonesia (UPMI) atas dukungan, fasilitas, dan kesempatan yang telah diberikan

sehingga penelitian ini dapat terlaksana dengan baik. Dukungan institusional dari UPMI, baik dalam bentuk sarana, bimbingan akademik, maupun lingkungan penelitian yang kondusif, telah memberikan kontribusi yang sangat berarti terhadap penyelesaian penelitian ini.

Daftar Rujukan

[1] Hidayatulloh, N. W., Tahir, M., Amalia, H., Basyar, N. A., Prianggara, A. F., & Yasin, M. (2023). Mengenal Advance Encryption Standard (AES) sebagai algoritma kriptografi dalam mengamankan data. *Digital Transformation Technology (Digitech)*, 3(1), 40–45. https://doi.org/10.47709/digitech.v3i1.2293

[2] Latip, P. N. (2025). Implementasi algoritma kriptografi AES dalam pengamanan file teks. *Jurnal Riset Sistem Informasi*, 2(3), 1–4. https://doi.org/10.69714/k6pr0s45

[3] Fitriani, D. E., Zulatifa, N., Anggraini, D. P., & Saputro, I. A. (2024). Analisis implementasi enkripsi dan dekripsi menggunakan algoritma Advanced Encryption Standard (AES) pada Java NetBeans. *Seminar Nasional Amikom Surakarta (SEMNASA)*, e-ISSN 3031-5581

[4] Andriyanto, M. R., & Sukmasetya, P. (2022). Penerapan algoritma Advanced Encryption Standard (AES) untuk keamanan data transaksi pada sistem e-marketplace. *Journal of Computer System and Informatics (JoSYC)*, 4(1), 179–187. https://doi.org/10.47065/josyc.v4i1.245

[5] Fajar, M., Wahid, A., Dirawan, G. D., Wahid, M. S. N., & Risal, A. A. N. (2024). Python dan kriptografi: Edukasi dan pengabdian untuk masa depan yang aman. *Jurnal Kreativa: Kemitraan Responsif untuk Aksi Inovatif dan Pengabdian Masyarakat*, 1(2), 150–152. http://journal.lontaradigitech.com/KREATIV

[6] Wardhani, T. D. A. P., & Asriningtias, Y. (2023). Implementasi algoritma AES-256 dalam perancangan aplikasi pengamanan dokumen digital perusahaan berbasis Android. *Journal of Information Technology and Computer Science (INTECOMS)*, 6(2), 1289–1293

[7] Maulyanda, M., Pane, S. F., & Rolly, R. (2022). Cryptography: Perancangan middleware web service encryptor menggunakan triple key MD5, Base64, dan AES. *Jurnal Informatika dan Rekayasa Perangkat Lunak (JIRPL)*, 3(2), 112–119

[8] Damayanti, N. R. (2021). Design aplikasi catatan daily berbasis Android menggunakan metode Waterfall. *Jurnal Teknologi dan Sistem Informasi (JTSI)*, 2(3), 45–52.

[9] Amalya, N., Silalahi, S. M. S., Nasution, D. F., Sari, M., & Gunawan, I. (2023). Kriptografi dan penerapannya dalam sistem keamanan data. *Jurnal Media Informatika (JUMIN)*, 4(2), 90–93

[10] Ramadhan, A. A. I., Rivanti, E. Z., & Zulva, R. S. (2023). Implementasi kriptografi AES menggunakan bahasa Java programming: Meningkatkan keamanan data melalui enkripsi & dekripsi yang kuat. *TRIPLE A: Jurnal Pendidikan Teknologi Informasi*, 2(1), 20–26.

[11] Purwanti, D. S. (2022). Perancangan penerapan algoritma kriptografi AES-256 untuk keamanan database aplikasi manajemen siswa. *Jurnal Teknologi dan Sistem Informasi (JTSI)*, 3(1), 34–40.

[12] A. Razaghpanah, A. Akhavan Niaki, N. Vallina-Rodríguez, S. Sundaresan, J. Amann, and P. Gill, "Studying TLS usage in Android apps," in *Proceedings of CoNEXT '17*, Incheon, Korea, Dec. 2017, 13 pages. doi: 10.1145/3143361.3143400.

[13] Kharisma, R. S., & Rachman, M. A. F. (2022). Pembuatan aplikasi notes menggunakan algoritma kriptografi Polyalphabetic Substitution Cipher kombinasi kode ASCII dan operasi XOR berbasis Android. *Jurnal Teknologi Informasi dan Komputer (JTIK)*, 8(2), 120–128

[14] Surbakti, N. M., Angelyca, A., Talia, A., Perangin-Angin, C. B., & Olivia, D. (2023). Penggunaan bahasa pemrograman Python dalam pembelajaran kalkulus fungsi dua variabel. *Jurnal Pendidikan Matematika dan Sains Terapan*, 7(1), 14–20.